Impact of COVID-19 on maritime traffic and vessel-related emissions

Master's Thesis

Facultat de Nàutica de Barcelona
Universitat Politècnica de Catalunya

Developed by:

Javier Nieto Guarasa

Supervised by:

Anna Mujal i Colilles, PhD

Marcel·la Castells i Sanabra, PhD

Master's degree in Naval Architecture and Ocean Engineering

Barcelona, 2 September 2020

Department of Nautical Science and Engineering

(This page intentionally left blank)

Acknowledgement

As this period of my life comes to an end, I would like to express my sincere gratitude to every single person who ever believed in me, showed interest in me and followed me as I kept going onwards to fulfill my dreams.

I would like to show my thanks to lecturers, thesis directors and mentors Anna Mujal-Colilles, PhD and Marcel·la Castells-Sanabra, PhD. To Anna, especially because she showed me the purpose of education and helped me understand the hidden beauty in engineering problems, and to Marcel·la, because she lighted up the spark that made me rethink my career moving from nautical science towards naval architecture.

Lastly, I would like to remember my family, every good friend and colleague, for every happy moment, every single conversation and every single smile that kept me going through the time.

As this period of my life comes to an end, I would like to thank everyone who ever crossed my life and showed me that learning is a never ending process and « that the day I stop learning is the day I start dying ».

To my cousin, Jose Durán

« El miedo es el alimento del fracaso »

Resumen

El nuevo virus SARS-CoV-2 fue detectado en humanos por primera vez en Diciembre de 2019. Más de medio año después, el virus se ha transmitido por todo el mundo causando la peor pandemia mundial en más de un siglo. Sus impactos llegan más allá de la enfermedad, ya que la situación de inestabilidad socioeconómica ha resultado en la llamada Gran Recesión del Coronavirus, afectando también a la industria marítima.

Este trabajo de fin de máster estudia el impacto de la COVID-19 en el tráfico marítimo y sus emisiones en un rango de 30 millas náuticas alrededor del puerto de Barcelona. El estudio emplea datos obtenidos a tiempo real, desde Marzo a Julio de 2020, a través de un receptor AIS localizado en la Facultat de Nàutica de Barcelona (Universitat Politècnica de Catalunya). Las emisiones se calcularon a través del algoritmo STEAM v.2 desarrollado por Jalkanen et al. en 2012.

Los resultados muestran que durante el confinamiento, el tráfico de buques aumentó en un +1,8% sobre la media global, a pesar de una reducción del -8,8% en las escalas en el puerto de Barcelona. Esto unido a una caída en las velocidades medias de los buques hace suponer un cambio en la forma de operar de éstos derivado de las incertidumbres de la pandemia.

En cuanto a las emisiones, éstas también fueron superiores a la media. Destaca que los buques de pasaje supusieron casi el 45% de las emisiones mientras que sólo representaron un 17,2% del total de buques. Además, se constató que el tráfico marítimo tiene un importante impacto en las variaciones diarias de los principales gases contaminantes en la ciudad, especialmente en lo que respecta a emisiones de SO_2 y NO_X.

Palabras clave

AIS, Barcelona, contaminación atmosférica, COVID-19, emisiones y tráfico marítimo.

Resum

El nou virus SARS-CoV-2 va ser detectat en humans per primer cop el mes de Desembre de 2019. Mig any després, el virus s'ha transmès per tot el món causant la pitjor pandèmia mundial en més d'un segle. Els seus impactes van més enllà de la malaltia, ja que la situació d'inestabilitat socioeconòmica ha resultat en l'anomenada Gran Recessió del Coronavirus, afectant també la indústria marítima.

Aquest treball de fi de màster estudia l'impacte de la COVID-19 en el trànsit marítim i les seves emissions en un rang de 30 milles nàutiques al voltant de la ciutat de Barcelona. L'estudi fa servir dades obtingudes a temps real, des del Març fins el Juliol de l'any 2020, a través d'un receptor d'AIS localitzat a la Facultat de Nàutica de Barcelona (Universitat Politècnica de Catalunya). Les emissions es van calcular per mitjà de l'algoritme STEAM v.2 desenvolupat per Jalkanen et. l'any 2012.

Els resultats mostren que durant el confinament, el trànsit de vaixells va augmentar en un 1,8% per sobre de la mitjana global, tot i una reducció del -8,8% en el nombre d'escales al port de Barcelona. Això juntament amb una caiguda de les velocitats mitjanes dels vaixells ens indica que van canviar la seva forma operativa degut a les incerteses de la pandèmia.

Pel que fa a les emissions, aquestes també van augmentar per sobre de la mitjana. Destaquen els vaixells de passatge, que van ser responsables de fins el 45% de les emissions però representaven només el 17,2% de tots els vaixells. A més, es va detectar que el trànsit marítim té una estreta relació amb les variacions diàries dels principals gasos contaminants de la ciutat, amb especial èmfasi en les emissions de SO_2 i NO_X.

Paraules clau

AIS, Barcelona, contaminació atmosfèrica, COVID-19, emissions i trànsit marítim.

Abstract

The novel SARS-CoV-2 virus was first found in humans somewhere in December 2019. More than half a year later, the virus has spread all around the world causing the worst global pandemic in over a century. Its impacts have gone well beyond the disease, as the unstable socioeconomic situation resulted in the dubbed Coronavirus Great Recession, reaching the maritime industry, as well.

This Master's thesis assesses the impact of COVID-19 on maritime traffic and the related emissions within a 30 nautical mile range around Barcelona. The study uses real-time AIS-acquired data, from March to July 2020, through a receiving unit located at the Barcelona School of Nautical Studies (UPC-BarcelonaTECH). Emissions were computed following the STEAM v.2 model developed by Jalkanen et al. in 2012.

Results found that during the strictest lockdown days, vessel traffic increased by +1.8% in the area, in spite of a reduction in -8.8% in the number of calls at Barcelona. This together with a reduction in the average speeds of vessels can be explained due to changes in the way vessels operated owing to the ongoing uncertainties.

Concerning emissions, values were also above the average. Passenger vessels were responsible for up to 45% of total emissions, whereas they represented just 17.2% of total traffic. Moreover, maritime traffic was found to have an important impact in daily variations of major air pollutants within the city, especially with regards to SO_2 and NO_X emissions.

Key words

Air pollution, AIS, Barcelona, COVID-19, emissions and maritime traffic.

Table of contents

List of acronyms

AIS	: Automatic Identification System
CDC	: Center for Disease Control and Prevention
CLIA	: Cruise Line International Association
CO	: Monoxide of Carbon
CO_2	: Dioxide of Carbon
COG	: Course Over Ground
COVID-19	: Coronavirus 2019 Disease
DTE	: Data Terminal Equipment
ECA	: Emission Control Area
EEDI	: Energy Efficiency Design Index
EEOI	: Energy Efficiency Operational Indicator
EF	: Emission Factor
EL	: Engine Load
EMSA	: European Maritime Safety Agency
EPFS	: Electronic Position Fixing System
ETA	: Estimated Time of Arrival
EU	: European Union
EUMRV	: European Union Monitoring, Reporting and Verification Program
GDP	: Gross Domestic Product
GHG	: Green House effect Gas
GLONASS	: Global Navigation Satellite System
GPS	: Global Positioning System
HDG	: Heading
HF	: High Frequency
HFO	: Heavy Fuel Oil
HSE	: High-Speed Engines
IAPP	: International Air Pollution Prevention
IEC	: International Electrotechnical Commision
IECC	: International Energy Efficiency Certificate

IMO : International Maritime Organization

ITU : International Telecommunications Union

LNG : Liquified Natural Gas

LSHFO **:** Low-Sulfur Heavy Fuel Oil

MARPOL : International Convention for the Prevention of Pollution from Ships

MCR : Maximum Continuous Rating

MGO : Marine Gas Oil

MMSI : Maritime Mobile Service Identity

MSE : Medium-Speed Engine

N_2 : Nitrogen

NaN : Not a Number

NO_X : Oxides of Nitrogen

NO : Monoxide of Nitrogen

NO_2 : Dioxide of Nitrogen

NUC : Not Under Command

O_2 : Oxygen

O_3 : Ozone

OPEC : Organization of Petroleum Exporting Countries

PM : Particulate Matter

PM_{10} : Particulate Matter of less than 10µm in radius

$PM_{2.5}$: Particulate Matter of less than 2.5µm in radius

ROT : Rate Of Turn

S : Sulfur

SAR : Search And Rescue

SARS-CoV-2 : Severe Acute Respiratory Syndrome Coronavirus 2

SEEMP : Shipboard Energy Efficiency Management Plan

SFC : Specific Fuel Consumption

SFOC : Specific Fuel Oil Consumption

SO_X : Oxides of Sulfur

SO_2 : Dioxide of Sulfur

SO_3 : Trioxide of Sulfur

SOG : Speed Over Ground

SOLAS : International Convention for the Safety of Life at Sea

SOTDMA : Self-Organized Time Division Multiple Access

SSE : Slow-Speed Engines

STEAM : Ship Traffic Emission Assessment Model

ULCC : Ultra Large Crude Carrier

TEU : Twenty-feet Equivalent Unit

UNCATD : United Nations Conference on Trade and Development

UNESCO : United Nations Educational, Scientific and Cultural Organization

UTC : Universal Time Coordinated

VHF : Very High Frequency

VLCC : Very Large Crude Carrier

VOC : Volatile Organic Compound

VW : Volkswagen

WHO : World Health Organization

List of figures

List of tables

(This page intentionally left blank)

Chapter 1. Introduction and objectives

1.1. Introduction

The novel Coronavirus Disease 2019 was first reported symptomatically somewhere between December 1 and December 8, 2019 in the landlocked Chinese city of Wuhan, Hubei (1)(2). Human-to-human transmission was first confirmed by WHO authorities by January 20, 2020 (3). Its rapid spread across the world, led to the global pandemic declaration by March 11, 2020 (3); with subsequent disruptions in world industry, trade and economy (4).

Table 1. Top 5 countries by number of confirmed COVID-19 cases as of July 31, 2020 - SOURCE: JHU

Country	Infected	Deaths
United States of America	4,770,379	157,424
Brazil	2,733,677	94,130
India	1,695,988	36,511
Russian Federation	856,264	14,207
South Africa	516,862	8,539

As seen in Table 1, most affected countries, in terms of total number of infected, include the United States, Brazil, India, Russia and South Africa, among others (5). As of July 31, 2020 a total of 18M people have been infected and up to 689k people have perished because of the infection, with a global average death-to-case rate estimated to be 3.8% (5).

Amidst the ongoing[1] COVID-19 global pandemic, 188 out of 251 countries and territories worldwide have declared some sort of curfew (5), lockdown or travel restrictions in order to prevent any further spread of the virus (6).

At that time, intercity wheeled and rail traffic went down by 90% in major EU cities (7). Worldwide air traffic shrank by 65% (7). Millions of schoolers worldwide were left without face-to-face lessons for the remaining academic year[2]. Unemployment grew up by 10% to 15% (6)(4)(8) in many developed nations. Oil barrel prices traded at a minimum for the first time since early 2000s (9). Although effect on the maritime world was not as tremendous, vessels did slow down their pace for a while, as uncertainties kept growing (10)(11).

According to latest UNCATD reports, international shipping moves around 80% of global goods worldwide (12). On daily basis, million tons of food, clothing, energies and commodities are shipped across the oceans. Supply chains are the foundation of many developed economies, which depend on international trade as a source of commerce and income. Whereas developing powers require customers for all their manufactured goods. Owing to these reasons, cross-border goods exchange was deemed essential to overcome the crisis (10)(11), as vital medical supplies, food and other manufactured goods were still required.

However, as national lockdowns are being constantly extended and economies begin to contract, shipping starts suffering as well (10)(11). Passenger cruise vessels were the very first to lay up, given their inherent high risk. Afterwards, passenger ferries moved onto essential rolled cargo shipments, and tankers began trading as mere drifting storage units, waiting for higher oil prices to resume operations. Slow steaming was also implemented among containerized fleet and larger vessels were taken out of trade.

Barcelona, as EU largest cruise port and Spain's third-largest cargo port, handled around 3M cruise passengers and 67.7M tons of cargo in 2019[3]. Spain has been one of the most hit countries by the COVID-19 crisis, with a total number of confirmed cases rising up to 288,522 and 28,455 officially declared deaths[4] (80). National lockdown entered in force in Spain on March 16, 2020[5] at midnight, and extended until June 22, 2020[6], with a nationwide home-quarantine week running from April 6

---

[1] As of July 31, 2020.
[2] 290M students globally according to UNESCO (Azoulay, 2020).
[3] As per 2019 Traffic Statistics Report by the Port Office Statistics Service.
[4] As of July 31, 2020. Official data retrieved from Carlos III Health Institute, appointed by the Spanish Ministry of Health to track the evolution of the disease in Spain. More info at: https://cnecovid.isciii.es/covid19/
[5] Royal Decree RD 463/2020, dated March 14.
[6] Royal Decree RD 555/2020, dated June 5.

to April 13[7]. Air traffic was reduced by 90%, wheeled traffic went down to a residual 20% and passenger traffic by sea was completely disrupted (13). Meanwhile, concentration of major pollutants went down only by 40% to 80% (13)(14). Bearing in mind the massive reduction of industrial activities, this may indicate that maritime might actually be pouring much more pollutants into the atmosphere.

In this scenario, the opportunity to assess and contrast with real data how the pandemic impacted maritime traffic and air quality within Barcelona came across. Official reports approved by the IMO acknowledge that shipping is responsible for up to 4% (15) of world's carbon footprint. However, these values may hide a different reality.

1.1.1. Impact of COVID-19 on worldwide shipping

International shipping is an essential vector within global economy (9). They are both so closely related that macroeconomic indicators actually play a major role in global maritime traffic trends (8).

Given this exceptional global situation, national lockdowns and curfews were established all around the globe (15). Economic activities were reduced to minimum essential services and gradually resumed as the pandemic curves flattened. The impact of lockdowns and mandatory quarantines resulted in negative consequences in the socio-economic field, far beyond the effects related to the spread of the disease, leading to the so-called Coronavirus Great Recession (2)(16)(17).

Impact on economic activities began as early as February 24, 2020; when first community infections were reported outside of China (2). On February 28, 2020 global stock exchange markets suffered the largest single-week fallout since the 2008 recession (17). This was followed by the March 2020 stock market crack, which has been labelled as the trigger of the forthcoming global recession (17). The April 2020 fuel oil price collapse followed shortly after (17). Tourism and hospitality have been by far the most affected industries, with a forecasted drop of 20% to 30% for the whole year (18) (19).

All in all, the worldwide impact on economy has been estimated at -0.9% of global GDP (2), bottoming at -2.4% and -3.5% GDP in the United States and the European Union respectively (17). Only in the first quarter, a real -5.0% GDP global drop was scored (17), affecting mostly developed economies.

Tankers and passenger vessels were the very firsts to be hit by the crisis (7)(8). As national lockdowns were constantly extended, some container operators downgraded selected liner services. The -40% demand drop in the car manufacturing industry, resulted also in a reduction of

[7] Royal Decree RD 476/2020, dated March 27.

capacity by car carrier operators (8). So far, commodities were the only sector able to weather the crisis with minor impact (8). As of July 31, 2020; the industry seems to be back in a slow growing path (20), meanwhile passenger traffic is still mostly limited to ferry crossings (20) as cruise tourism is still banned in major destinations across the globe.

Cargo vessels

Among all ship types, cargo vessels of any kind, except for car carriers, did see a moderate reduction in global traffic (16). In Europe, single digit drops, except during April, were common among major ports through the first semester. A weaker demand in Europe, together with the early stages of COVID-19 in China and later global crisis explain this situation (16). As of August 2020, early signs of recovery within global cargo traffic are already visible in major ports (10).

Table 2. Changes in the number of ship calls for cargo vessels in EU ports in 2020 - SOURCE: EMSA
Values are changes compared to 2019 monthly average number of calls

Ship type	March	April	May	June	July	TOTAL
Bulk carrier	-8%	-18%	-6%	-3%	-5%	-6%
Container ship	-6%	-11%	-10%	-9%	-6%	-9%
General cargo	-11%	-12%	-10%	-9%	-8%	-9%
Ro/Ro	-18%	-14%	-15%	-12%	-6%	-11%
Car carrier	-66%	-69%	-49%	-31%	-27%	-43%

As seen in Table 2, bulk carriers have managed to weather the effects of COVID-19 crisis much better than other traffics. In fact, commodities usually behave in a steadier trend than other cargoes (16). All in all, reductions have been reported mostly in terms of coal and ore carriers (10), as the shutdown of heavily industrial economies, like China or India, reduced also the importation of such goods.

Container ships, general cargo and Ro/Ro shipments have been following very similar trends along the crisis. These traffics are heavily dependent on the demand of developed economies, where manufactured goods are shipped from Asia (16). They are actually among the less liquid within the shipping business, as they are closely related to economic activities (17).

A special consideration for car carriers, as demand for such cargo is related to the performance of the industry (17). Major car manufacturers saw in early 2020 a drastic reduction in local and global markets, a situation that became worse as the pandemic developed (18). As economies reopened, a slight growth became effective within the sector, but still well below previous years.

Tanker vessels

Tankers were affected at a very first stage of the global pandemic (19)(20). As early as mid-February, the demand of crude oil in the Chinese giant went down from an average daily of 3.4b tons in 2019 to virtually none (19). Chartering rates for VLCC went down by 20% in over a month, driven by a reduced manufacturing capacity in China (20). By the beginning of March, the ongoing Saudi Arabia – Russia oil price war resulted in the walking out of the Russian Federation from the OPEC agreement, which drove fuel prices down by 65%, reaching negative values by April 20, 2020 (19).

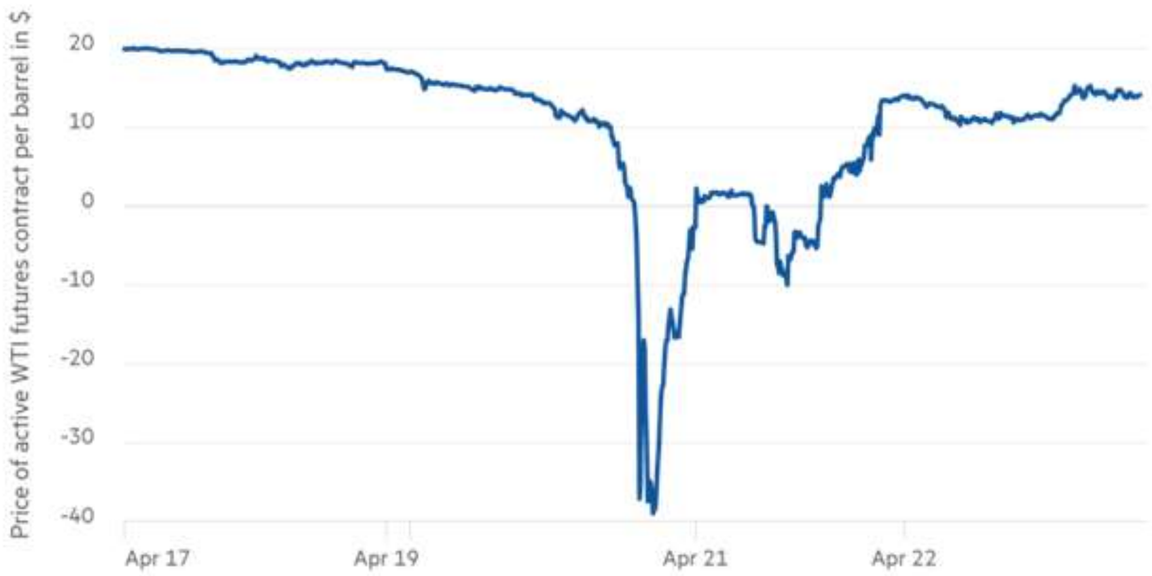

Figure 1. West Texas Intermediate price evolution (April 2020) - SOURCE: Financial Times

Given the shortage in global fuel demand and the impossibility to completely shut down fuel extraction, larger vessels turned into oil floating storage units, drifting at sea waiting for oil buyers (19). As of April 20, the holding capacity was not enough to sustain the continuous oil extraction, thus resulting in negative prices, as shown in Figure 1. In fact, the oil demand dropped by -70% during the month of April and many shore-side terminals and storing facilities reported to be over their limit by that time (20).

Table 3. Changes in the number of ship calls for tanker vessels in EU ports in 2020 - SOURCE: EMSA Values are changes compared to 2019 monthly average number of calls

Ship type	March	April	May	June	July	TOTAL
Chemical tanker	+31%	+10%	+26%	+16%	+2%	+15%
LNG tanker	-16%	-24%	-18%	-8%	-10%	-14%
Oil tanker	-6%	-3%	-3%	-4%	-4%	-4%

In Table 3, as countries lifted home-quarantines and travel restrictions, the demand of fuel and petrol derivatives began to increase again.

Experts have reported the COVID-19 crisis as the best example of how influent the Chinese economy is in the shipping industry (21). In fact, as the pandemic was developing in China in early January 2020, world shipping already started seeing the very first signs of contraction. The importance of China as a maritime traffic driver has actually two main sources (17):

i. Demand of raw materials drives bulk carrier and tanker traffic; and
ii. Exportation of manufactured goods from China drives the container and general cargo markets.

Passenger vessels

Passenger traffic was the most affected sector within the maritime business (20). After a month of confinement in Europe, major ferry operators reported a drop of -40% in their businesses (10)(16), leading to many of their vessels to be transitioned into warm layup, waiting for travel restrictions to be lifted (16). Ferry traffic slightly recovered pre-COVID-19 traffic figures upon reopening. Cruise passenger vessels are left out of any recovery, as travel restrictions still apply to this kind of tourism in many nations (10).

Table 4. Changes in the number of ship calls for passenger vessels in EU ports in 2020 - SOURCE: EMSA Values are changes compared to 2019 monthly average number of calls

Ship type	March	April	May	June	July	TOTAL
Cruise ships	-95%	-95%	-94%	-93%	-92%	-94%
Ro/Pax	-37%	-33%	-30%	-10%	+1%	-17%

As shown in Table 4, the cruise industry has been the most badly affected, with a full fleet layup starting mid-March 2020, and still in force as of July 31, 2020 (16). Many cruise operators decided also to cancel sailings through summer 2021[8], owing to extremely pessimistic forecasted bookings

[8] As of August 1, 2020; only the following cruise lines are operating at a reduced capacity: Hurtigruten (Svalbard roundtrips on the *MS Roald Amudsen* and the *MS Finnmarken*), Sea Dream Cruises (Norwegian fjords roundtrips for the local market on the *MS Sea Dream I and MS Sea Dream II*), Ponant (full fleet trading in Normandy and Côte d'Azur for the local market), Paul Gaugin Cruises (French Polynesia roundtrips for the local market), Dream Cruise Lines (Taiwan roundtrips for the local market on the *MS Explorer* Dream) and TUI

and rise in cruise travel bans in many ports (10). Owners have also delayed the delivery of new vessels and/or cancelled several orders[9] and sold older fleet members for scrap[10], so as to adjust capacity to future demand.

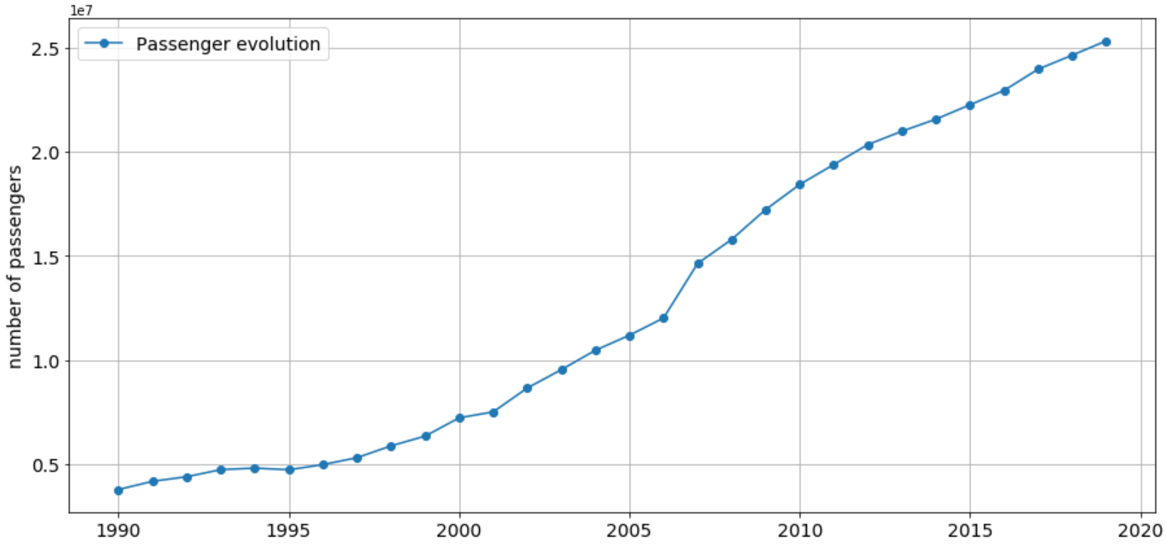

Figure 2. Worlwide cruise passengers evolution - SOURCE: Cruise Market Watch

Figure 2 shows that prior to the COVID-19 pandemic, cruise ship passengers were growing at a steady rate since 1990. With an expected number of cruisers estimated at around 30M in 2020, major cruise lines warned that it might take until 2023 for them to recover from the current crisis (22).

Given that several passenger cruise vessels have been involved in major COVID-19 outbreaks (10), they have been subject to continuous mass media coverage. This resulted in increased attention by port communities, seeking to ban or reduce cruise ship arrivals, as they may help spread the virus.

Cruises (cruises to nowhere out of Germany for the local market on the *Mein Schiff 1* and *Mein Schiff 2*). Costa Crociere, MSC Cruises and Aida Cruises plan to resume operations with two vessels each starting August 15, 2020. All major American brands and the rest of the operators have delayed the gradual reintroduction of cruises to October, 2020; except for Princess Cruises and Crystal Cruises that paused all operations through December, 2020 and Celestyal Cruises that decided not to resume operations until March, 2021.

[9] As of August 1, 2020 a total of 9 deliveries have been delayed to 2021: *MS Mardi Gras* (Carnival Cruise Lines), *MS Silver Moon* (Silversea), *MS Odyssey of the Seas* (Royal Caribbean International), *MSC Virtuosa* (MSC Cruises), *MS Evrima* (Ritz Carlton Cruises), *MS Iona* (P&O Cruises), *MS Enchanted Princess* (Princess Cruises), *MS Costa Firenze* (Costa Crociere) and *MS Crystal Endavour* (Crystal Cruises).

[10] As of August 1, 2020 a total of 31 cruise vessels have been sold for scrap or their future is uncertain.

The resuming of operations by end of July was heavily controversial, as COVID-19 cases have been reported in 5 out of 8 cruise operators already serving guests or planning to do so in the near future. This led several countries to reintroduce temporary bans on ruise ship operations, fearing new COVID-19 outbreaks related to cruise passengers (23).

1.1.2. Impact of COVID-19 in the Port of Barcelona

The Port of Barcelona is a major economic asset for the city. The 2000-year-old facility is the main getaway for 18% of Spanish GDP[11] and a massive inter- and cross-border hinterland, reaching all neighboring Spanish regions and Southern France. Its success has been driven partly thanks to a diversified traffic, heavy international promotion campaigns, its location in the Gibraltar-Italy passage, and its intermodal connectivity (24).

The Port of Barcelona has been constantly breaking all traffic records within the last decade. It ranks among top 10 cruise ports worldwide (24). It is the largest cruise port outside of the Americas, and number one in Europe, with more than 3M passengers solely in 2019, and a continuous year-over-year growth rate of 3% to 5%[12]. Moreover, cargo is also an important asset for the Port of Barcelona, as it ranks third within Spain (10)(24), with 67.7M tons of cargo, 34M TEUs and 800,000 cars in 2019[13]. All in all, the port contributes with 9,300M€ or 5.7% of the Catalan GDP (24).

As of July 2020, official sources from the Port of Barcelona reported a -18.8% fallout in terms of total cargo, when compared to 2019 figures[12] (81). Additionally, a -58.4% and -78.3% reductions have been noted for both passenger ferry and cruise vessel calls, owing to the ongoing COVID-19 global pandemic and travel restrictions (18).

[11] Official Catalan GDP for FY2019, as reported by the Catalan Institute of Statistics.
Available at: https://www.idescat.cat/pub/?id=aec&n=356&lang=en

[12] As per 2019 Traffic Statistics Report by the Port Office Statistics Service.

Vessel traffic trends

As of December 2019, the Port of Barcelona received a total of 8,901 vessels, down from a total of 9,038 vessels accumulated in 2018 (81). However, this value is consistent with a steady trend on a year-over-year number of vessels, which usually peaks at around 9,000 (27).

The total number of calls follows a traditional season-influenced pattern (27), where a basis of cargo – mostly container ships, Ro/Ro and few bulk carriers and tankers – represents a continuous steady foundation throughout the year; and passenger traffic – cruise ships and additional Ro/Pax ferry traffic – helps boost the total number of yearly calls between March and November.

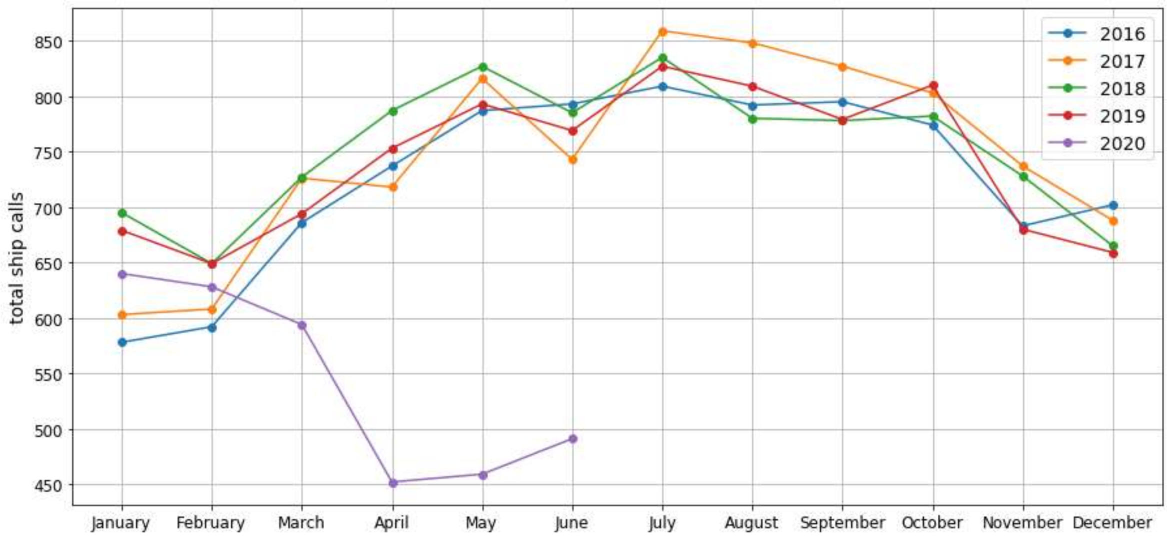

Figure 3. 5-year overview of ship calls at the Port of Barcelona – SOURCE: Port of Barcelona

Over the last 5 years, as seen in Figure 3, July has always been the peak month, with a maximum of 859 vessels solely in 2017; whereas, February tends to be the month with the minimum number of vessels, scoring a lowest in 2016 with only 592 vessels.

Continuous growth rates of +3% have been experienced over the last 10 years in terms of total vessel calls. However, in 2019 an averaged -1.5% reduction was reported, driven mostly by the latest worldwide economic contraction trends, specially affecting European economies (25).

As for 2020, official figures showed a negative trend in the first quarter (81), similar to the pre-2018 levels, continuing the previous 2019 prospections. On monthly basis, January saw -5.7% less calls, February -4.5% less calls and March up to -7.9% less calls than in 2019. These negative trends were initially driven due to the reduced capacity of Chinese exportations given the early stages of the COVID-19 crisis, a weaker automotive industry in Europe and the ongoing economic contraction within the EU (26). The introduction of lockdown measures as the pandemic hit Europe, resulted in a massive fall down in terms of total vessels, with -16.6% less calls in April, -22.3% less calls in May and -24.8% less calls in June. June also saw a timid recovery, driven by an increase in the number of calls of container ships, tankers and Ro/Pax ferries. However, figures were far from previous years as the boost from cruise ships was missing.

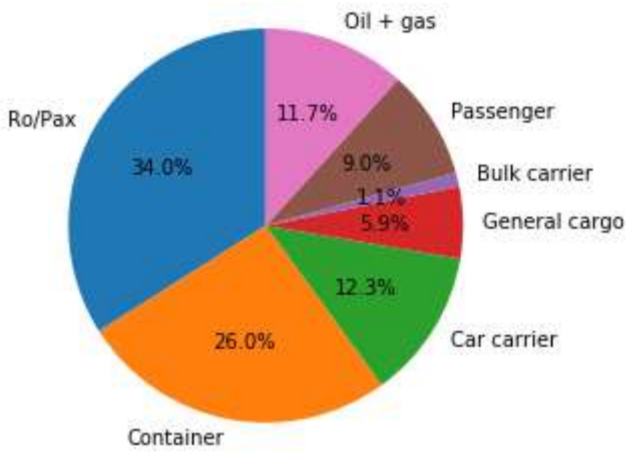

Figure 4. 2019 distribution of calls by type of ship – SOURCE: Port of Barcelona

In Figure 4, Ro/Pax ferries dominate the scene, representing up to 34.0% of total vessel or 3,025 vessel calls in 2019. Container vessels represented up to 26.0% or 2,313 vessel calls in 2019. Pure car-carriers and Ro/Ro vessels accounted for up to 12.3%, whereas tankers represented 11.7% of total calls. Passenger cruise vessels contributed with 9.0% to the total calls in 2019. All of these values are consistent with the latest trends (81).

Cargo and tanker vessels

The Port of Barcelona has a total of 24 cargo terminals, as stated in Table 5, of which 2 are devoted to containerized cargo, 7 to commodities in bulk, 10 to oil and gas, 2 to manufactured vehicles, 2 to wheeled cargo (Ro/Ro) and 1 to general cargo. All in all, pure cargo traffic represented up to 57% of total vessel calls in 2019, as shown in Figure 4.

Table 5. Cargo terminals at the Port of Barcelona - SOURCE: Port of Barcelona

Terminal	Dock	Cargo / Tanker
Terminal APM	Moll Sud	CONTAINER
Terminal BEST	Moll Prat	CONTAINER
Terminal Port Nou	Moll Adossat	GENERAL
Terminal Autoterminal	Moll Sud	WHEELED
Terminal SETRAM	Moll Sud	WHEELED
Terminal Enagás	Moll Energia	OIL & GAS
Terminal TEPSA	Moll Energia	OIL & GAS
Terminal Terquimsa	Moll Barcelona	OIL & GAS
Terminal DECAL	Moll Energia	OIL & GAS
Terminal Koalagás	Moll Energia	OIL & GAS
Terminal Meroil	Moll Ponent	OIL & GAS
Terminal TRADEBE	Moll Delta	OIL & GAS
Terminal Quimidroga	Moll Delta	OIL & GAS
Terminal CLH	Moll Delta	OIL & GAS
Terminal Internacional BCN	Moll Álvarez de la Campa	BULK
Terminal Portcemen	Moll Contradic	BULK
Terminal Cargill	Moll Álvarez de la Campa	BULK
Terminal Bunge Ibérica	Moll Contradic	BULK
Terminal ERGRANSA	Moll Príncep d'Espanya	BULK
Terminal TRAMER	Moll Contradic	BULK

Most of the terminals are grouped by areas and type of cargo, as shown in Figure 5. All of them are in Barcelona proper, except for the newest container terminal, which is located in El Prat de Llobregat, a municipality in Greater Barcelona.

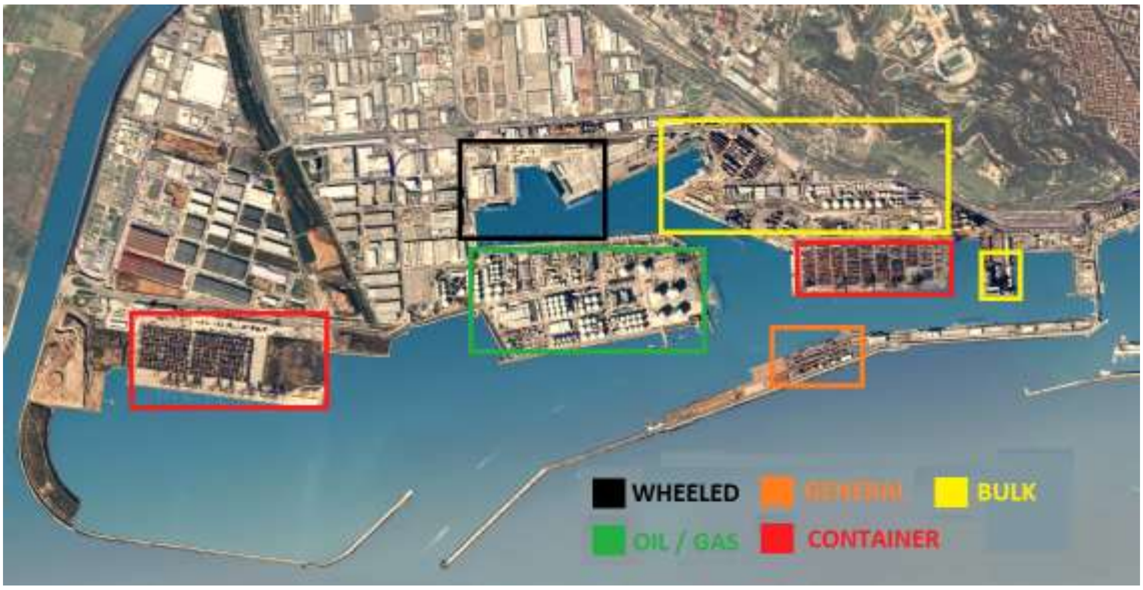

Figure 5. Location of cargo terminals in Barcelona

Considering only pure cargo vessels – excluding tankers –, an average number of 330 to 350 calls have been reported on monthly basis, as in Figure 6, for the last 5 years. This value is consistent and follows a steady trend along the year, as it represents the foundation of monthly calls at the Port of Barcelona.

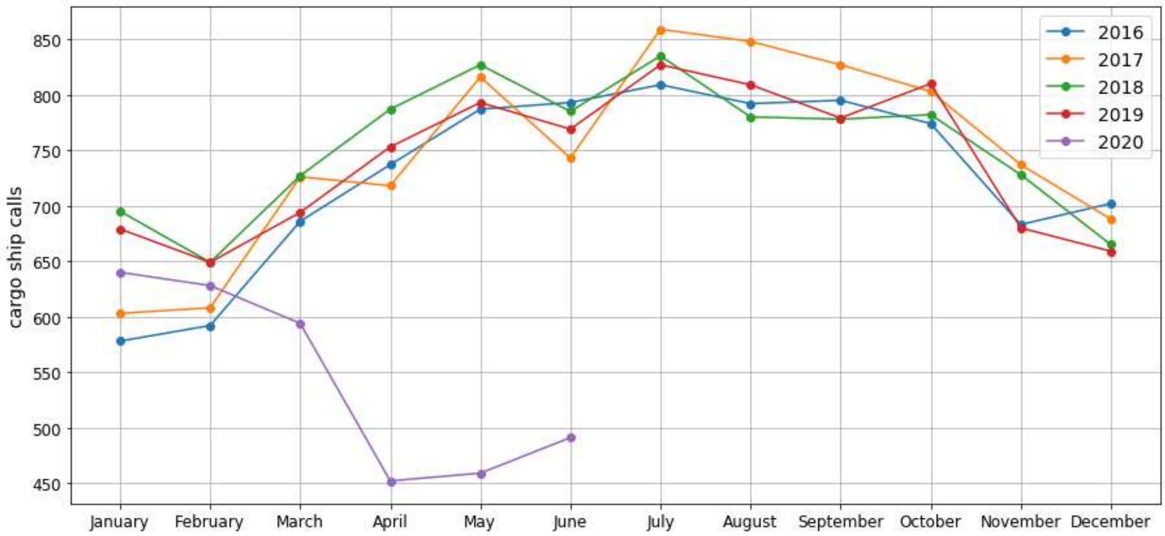

Figure 6. 5-year overview of cargo ship calls at the Port of Barcelona – SOURCE: Port of Barcelona

A massive drop was recorded once the lockdown was declared in Spain, as the number of calls fell well below 300 for the forthcoming months. April was by far the worst month, driven mostly by a -50% drop in the number of car carrier calls and -15% in all other traffics, except for bulk carriers, which kept similar figures as in previous years (81). All in all, the number of calls dropped by -25% the early stages of the COVID-19 crisis.

As for tankers, an average of 70 to 90 calls per month have been the norm during the last years, as per Figure 7. Similar values have been reported for the first semester of 2020, more in the lower range, especially for the second half.

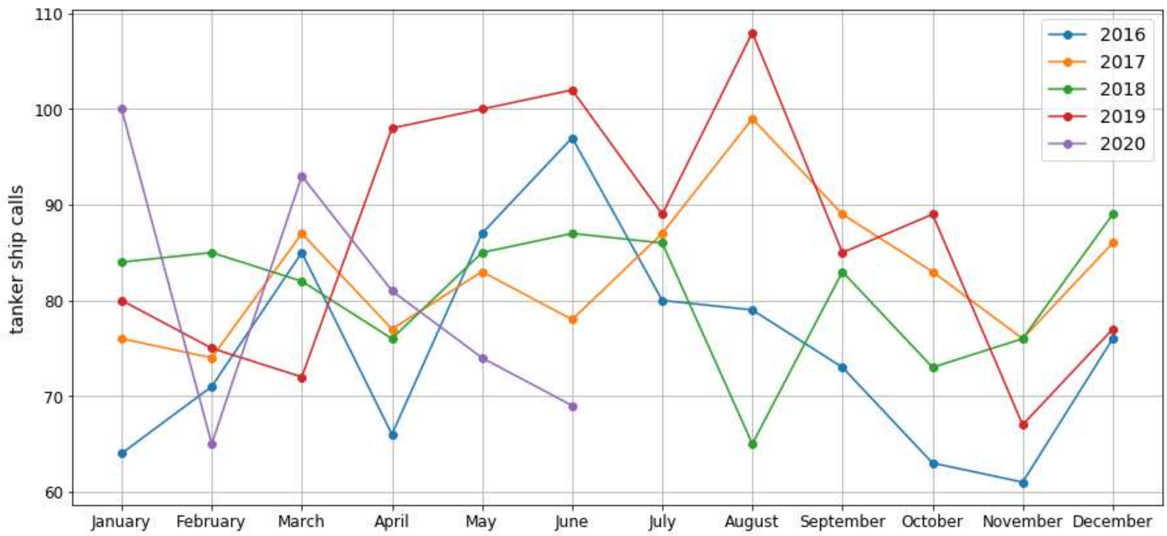

Figure 7. 5-year overview of tanker ship calls at the Port of Barcelona – SOURCE: Port of Barcelona

Initially, as the lockdown measures were established across Europe, most traders decided to offload all oil and gas and store it in shore facilities (19), this explains the peak related to March, in which the number of calls was +14% higher than the average. However, as demand dropped, the number of tanker calls reduced at a steady state. All in all, the number of calls dropped by -6% during the COVID-19 crisis.

In a quarterly results call on July 24, 2020; Ms Mercè Conesa, president of the Port of Barcelona, noted the following impact of COVID-19 in the port business over the first semester of 2020 (18):

i. Container movements went down by -20.5% - or -1.3M containers –;
ii. Manufactured cars were reduced by -47.6% - or -218,000 vehicles –;
iii. Fuel and LNG went down by -23.2%; and
iv. Ro/Ro cargo went down by -19.4%.

Both traffic and cargo movements for all types, except for car carriers, saw early improvement stages by end-June (18), as the national lockdown came to an end in Spain and the economy began a slow growing process. Manufactured vehicles remained on lower digits, compared to previous years as the car industry is submerged in its own crisis in Europe (18), especially affecting Catalonia.

Passenger vessels

The Port of Barcelona is well known internationally as a major cruise homeport, ranking among top 10 worldwide. During 2019, a total of 4,628,562 passengers transited through the port terminals, of which 3,137,918 correspond to cruise passengers and 1,490,644 to ferry passengers[13] (81). In this scenario, Barcelona ranks as Europe's largest cruise port and Spain's third busiest port in terms of total passengers[14].

The Barcelona cruise boom has been subject to international analysis, given that the port managed to get 10 times more passengers in less than 15 years of cruise industry (27). The success of Barcelona is explained thanks to an effective port-city integration and heavy international promotion campaigns (27).

In terms of vessel movements, pure passenger vessels represented around 9.0% of total traffic, whereas Ro/Pax ferries accounted for up to 34.0%, as seen in Figure 4. A total of 11 passenger terminals are operative, of which 8 are devoted to passenger cruise vessels and 3 to Ro/Pax ferries. Passenger terminals are located along *Moll Adossat* (Terminals A to E), *Moll Barcelona* (Terminals WTC-N, WTC-E and WTC-S), *Moll Drassanes* (Terminal *Drassanes*), *Moll Sant Bertran* (Terminal *Ferry Barcelona*) and *Moll Ponent* (Terminal *Grimaldi*).

Table 6. Passenger terminals at the Port of Barcelona - SOURCE: Port of Barcelona

Terminal	Dock	Turnaround
Terminal A	Moll Adossat	4500
Terminal B	Moll Adossat	4500
Terminal C	Moll Adossat	3800
Terminal D "Palacruceros"	Moll Adossat	4500
Terminal E "Hèlix"	Moll Adossat	4500
Terminal North	Moll Barcelona	800

[13] As per 2019 Traffic Statistics Report by the Port Office Statistics Service.
[14] Behind Palma and Algeciras, which served 5.7M and 5.5M passengers in 2019.

Terminal South	Moll Barcelona	1400
Terminal East	Moll Barcelona	1400
Terminal Drassanes	Moll Drassanes	500
Terminal Ferry Barcelona	Moll St. Bertran	200
Terminal Grimaldi	Moll Ponent	3000

With a total turnaround capacity of 29,100 passengers simultaneously, as stated in Table 6, the Port of Barcelona is as a primary hub for major passenger operators in the Western Mediterranean. Most of the terminals are located within a walkable distance from the city center, less than 2km, as noted in Figure 8.

Figure 8. Location of passenger terminals in Barcelona - SOURCE: Port of Barcelona

With an average of 250 monthly calls during low season and 350 vessel calls during peak months, passenger vessels calls are a major asset for the Port of Barcelona, as they boost the total number of operations from March to November every year.

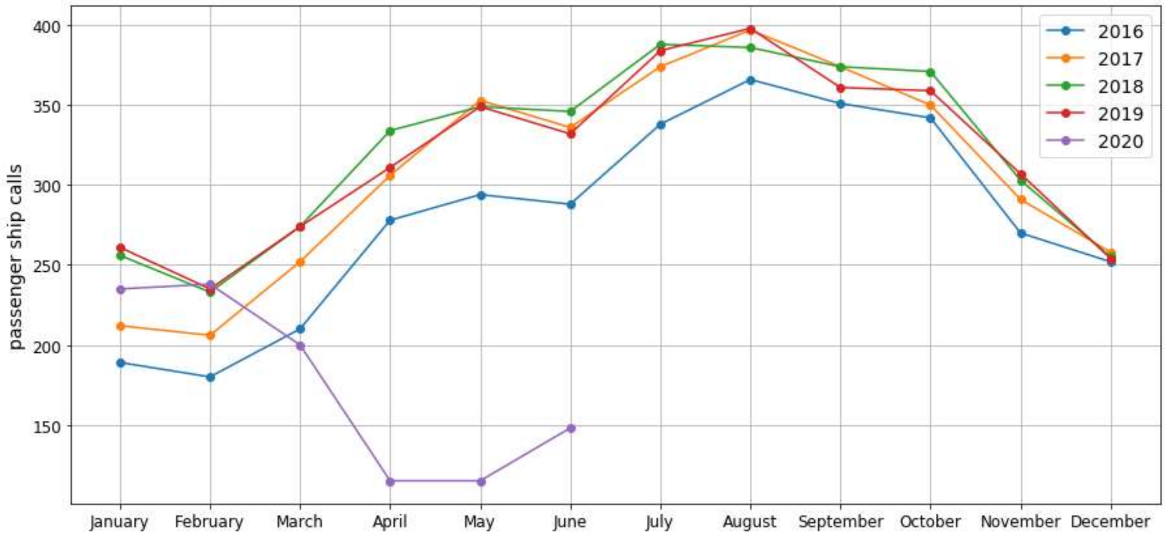

Figure 9. 5-year overview of passenger ship calls at the Port of Barcelona – SOURCE: Port of Barcelona

Figure 9 explains the first semester results, as cruise operations have been virtually non-existent since March 15, 2020; when cruise vessels were banned from entering Spanish ports except for emergency[15]. In 2019, the Port of Barcelona received a total of 830 cruise ship calls, expecting to break the count again in 2020 (81).

Regarding ferry operations, although they resumed by mid-June 2020, the total number of calls was still -45% lower than in the same period of 2019. With a monthly average of 230 to 250 ferry calls, current data peaked at 150 in June 2020, well below previous years.

Similarly, as for cargo operations, the port executive noted the following impact of COVID-19 in the passenger business over the first semester of 2020 (18):

i. Passengers by ferry went down by -61%; and
ii. Passengers by cruise ship were reduced by -84%.

Ferry passengers saw an important boost as travel restrictions were lifted within and towards Spain, with special increase in the operations towards the Balearic Islands and Italy. However, cruise passenger operations, a critical business for Barcelona, did not see any improvement at all, as cruise operations attended by the port are virtually zero (18) since March 15, 2020; except for passenger and crew repatriation operations. Cruise passenger operations are not expected to return to previous values at least until 2023 (18).

[15] According to Orders PCM/216/2020, of March 12; TMA/286/2020, of March 25; and TMA/330/220, of April 8; cruise vessel calls were still banned from entering Spanish waters as of August 1, 2020.

What to expect in the near future?

In terms of cargo and tanker operations, the number of calls is expected to slowly increase as the economy reopens and recovers previous values. The shipping industry is closely tied to major macroeconomic indicators, thus as economy keeps growing, the shipping business does so. As for passenger operations, the bell curve-like trend will not be expected at least until 2021, as cruise operations are way far from being resumed at its fullest, whereas ferry traffic is not enough to sustain the importance of passenger movements in the Port of Barcelona.

Based on the current economic perspectives, cargo and tanker operations are expected to slowly grow during the forthcoming months, with early recovering signs by 2021 and full consolidation by 2023 (18). Most analysts believe a V- (more optimistic) or U-shape (less optimistic) rebound of cargo transshipments, following the economy trends (28). W- (pessimistic) and L-shape (more pessimistic) rebounds could be possible if major developed economies and China are not able to control secondary pandemic outbreaks, so-called waves (28).

Manufactured vehicles are a special exception, as Japanese manufacturer Nissan Motor Co. Ltd., announced in May 2020 the closure of its Barcelona facilities, where 87.5% of all manufactured vehicles were for exportation (29), mostly ferried by sea. Several car carrier companies currently associated with Nissan Motor Co. Ltd., will likely cancel all their calls at the Port of Barcelona. Pre-COVID-19 car carrier traffics are thus unlikely in the near future.

Although travel restrictions have been mostly lifted within Europe and major ports are accepting cruise ships again as of August 1, 2020; most cruise operators are still delaying their resuming of operations well beyond fall 2020. Some companies extended their pause citing global travel restrictions and difficult for international passengers to reach embarkation ports, while others argue that operation cannot be run safely given the current status of the pandemic across the globe. As soon as operations were restarted, several companies have reported COVID-19 cases either among crew members or guests, something that might become the new reality on board cruise ships. In fact, as with the rest of the tourism and hospitality industry, cruise ships have been the most affected sector within the maritime industry. Several experts have already forecasted that cruise operations might not even resume until mid-2021, and only at half gas (10).

All in all, a facility like the Port of Barcelona, which has an extremely high dependence on passenger vessels has been hardly hit by the pandemic. The executive board expects to recover and consolidate pre-COVID-19 traffic rates by 2023 (18), with an ambitious program to enhance the port hinterland, its connectivity and services offered to passenger vessels.

1.2. Objectives

The main purpose of this thesis is to **study the impact that the novel SARS-CoV-2 virus had in the maritime traffic in Barcelona, and the effect of these changes in major emissions related to the shipping industry** within the city. Minor goals include:

i. Reviewing the current legal framework in terms of air pollution prevention;
ii. Understanding the post-COVID-19 maritime traffic trends within Barcelona;
iii. Generating a model able to estimate emissions when enhanced technical data is not available; and
iv. Analyzing the impact of ship-related emissions on air quality in Barcelona.

The idea came up amidst the early stages of the Coronavirus Great Recession, in early April 2020; as a sizeable number of vessels transitioned into warm and cold layup. It was deemed necessary to study the real immediate impact of COVID-19 on maritime traffic and make the most of the situation by producing updated and accurate data on ship-related emissions and their impact, which is always a topical issue, within industry experts.

Research was developed following the IMRaD approach – Introduction, Method, Research and Discussion – applied to analyze AIS-acquired data on maritime traffic, and to assess emissions through the STEAM v.2 emission algorithm. It focused on the Port of Barcelona, and within a range of 30 nautical miles during a timespan from March to July 2020, both included.

This master's thesis is organized in 6 chapters. Chapter 1 covers an introductory background and the thesis objectives, analyzing the socioeconomic impact in the shipping industry behind the COVID-19 pandemic. Chapter 2 revises the Automatic Identification System, marine power, associated emissions and the legal framework covering them, both internationally and locally. Chapter 3 disserts on the methodology followed to assess the impact on maritime traffic and generate the emission inventory. Chapter 4 and 5 introduce the results and discussion related to the changes in maritime traffic and related emissions, respectively. Finally, Chapter 6 is a closing for the thesis, summarizing the main points and results. The thesis also includes 4 annexes, with all code used to develop the project and more detailed data on the obtained results.

The study was developed from April to August 2020, with a continuous work performed day-by-day, as data was being readily available. All code for data analysis was finalized by end-July 2020, whereas the required algorithms for emission assessment were finished by August 2020.

Chapter 2. Background

This chapter introduces the state-of-the-art of the Automatic Identification System, vessel-related air pollution and air quality in Barcelona.

All data in the project was acquired through an AIS receiver, therefore a comprehensive introduction on the system, its technical particulars and messages is provided.

Regarding vessel-related air pollution, a brief discussion on shipboard power demands and marine engines is given, together with an enumeration of major air pollutants resulting from combustion and the legal framework covering them, both internationally and locally in Barcelona.

Eventually, some words on air quality and COVID-19 related changes closes the chapter.

2.1. The Automatic Identification System

The Automatic Identification System, in short AIS, is a ship-based VHF-borne tracking system which automatically broadcasts and receives both technical and voyage particulars about the ownship and all other surrounding vessels within VHF range (60)(61).

Adopted by the International Maritime Organization through SOLAS Chapter V, regulation 19, the carriage of an AIS transceiver unit has been compulsory since December 31, 2004 for all vessels falling into any of the following categories (60):

i. Ships of 300GT and upwards engaged in international traffic;
ii. Cargo ships of 500GT and upwards engaged in domestic traffic; and
iii. All passenger ships.

AIS operates on two dedicated VHF channels, the so-called AIS Ch.1 – VHF Ch.87b or 161.975MHz – and AIS Ch.2 – VHF Ch.88b or 162.025MHz –, where these frequencies are not available, transceivers are able to switch to the locally allocated frequencies (62).

The IMO performance standard requires a minimum of 2000 time slots to be sent per minute. Current ITU standards divide a minute into 2250 time slots of 26.67ms each and a transmission speed of 9.6kbps, which means a total of 256 bits sent per slot, enough to cover a full dynamic AIS transmission message (62)(63).

The AIS broadcast mode is based on the SOTDMA access scheme, short for self-organized time division multiple access. This allows the network to continuously work on an overloaded mode by 400% to 500% and still provide nearly 100% error-free messages for stations in the 8 nautical miles (= 14.8km) to 10 nautical miles (= 18.5km) range (61)(62)(63). Every AIS transmission, as explained in Figure 10, also includes a time slot reservation for the next message to be broadcasted. If the system is further overloaded, a drop-out filter acquires only information from stations closer to the receiving unit.

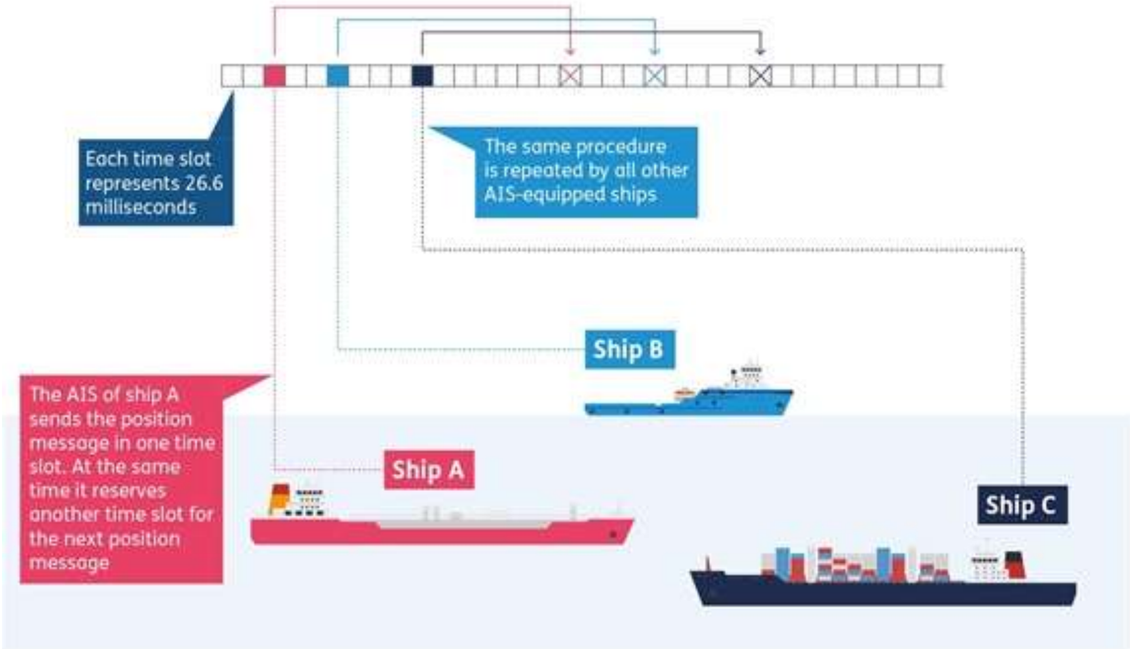

Figure 10. AIS message time slot management - SOURCE: Maritime Intelligence

Two different AIS systems have been developed so far, i.e. AIS Class A and AIS Class B (62). Information and functionalities vary between both protocols. For instance, Class B systems operate at reduced reception-transmission rates using free time slots when available. Assembly Resolution A.1106(29) adopted on December 2, 2015 requires all SOLAS[16] vessels carrying AIS units to install only Class A transceivers, whereas non-SOLAS vessels may use the Class B counterpart.

Information encrypted within Class A AIS messages is classified into static, dynamic, voyage-related and safety-related data (63).

Static information includes a series of non-user interchangeable data to be set up upon installation.

i. Station MMSI;
ii. Station callsign and name;
iii. IMO number;
iv. Length and beam (overall values);
v. Type of ship; and
vi. Location of electronic position fixing system (EPFS) antenna.

[16] For SOLAS vessel the following shall be understood: vessel of 300GT and upwards engaged in international travel, 500GT and upwards engaged in domestic travel or any size carrying more than 12 passengers.

Dynamic information includes mostly data automatically acquired from sensors. Some of this data might not be available if sensors are not connected to the AIS transceiving unit or they are not available on board.

i. Ship's position with accuracy indication and integrity status;
ii. Position time stamp in UTC;
iii. Course over ground (COG);
iv. Speed over ground (SOG);
v. Heading (HDG);
vi. Navigational status ; and
vii. Rate of turn (ROT).

Voyage-related information includes data that must be entered manually by the user and is related to the specific voyage undertook by the vessel.

i. Ship’s draught;
ii. Hazardous cargo type;
iii. Destination and ETA; and
iv. Route plan (waypoints).

Safety-related information includes short free-format safety messages, to be manually entered by the user and addressed to a specific station and to all stations within range, when broadcasted by coastal stations.

Class A AIS data is broadcasted on different time ranges, depending on the type of information, the vessel status and her speed (63). In general terms, static and voyage-related information is broadcasted every 6 minutes or when parameters have changed and safety-related messages are broadcasted upon request. For dynamic information different broadcasting times apply based on a series of conditions as stated in Table 7.

Table 7. AIS broadcasting frequency based on dynamic condition - SOURCE: ITU

Ship dynamic condition	Broadcasting time
At anchor or moored and with SOG less than 3 knots	3 minutes
At anchor or moored and with SOG more than 3 knots	10 seconds
SOG 0 – 14 knots without changing COG	10 seconds
SOG 0 – 14 knots and changing COG	3 1/3 seconds
SOG 14 – 23 knots without changing COG	6 seconds
SOG 14 – 23 knots and changing COG	2 seconds
SOG more than 23 knots without changing COG	2 seconds
SOG more than 23 knots and changing COG	2 seconds

2.1.1. How AIS data is sent?

AIS messages are broadcasted in 256-bit packets following the ISO/IEC 13239:2002 standard transmission package (63) as specified in Table 8. It contains a start buffer (23 bits), a training sequence (24 bits), a start flag (8 bits), the message data (168 bits), the frame-check sequence (16 bits), the end flag (8 bits) and the end buffer (9 bits), totaling 256 bits[17].

Table 8. Data sequence contained in a generic AIS message - SOURCE: ITU

Data sequence	Bits	Description
Start buffer[18]	23	6 different bit sequences opening the time slot
Training sequence	24	Alternating sequence of binary digits (010101…)
Start flag[19]	8	01111110
Data	168	Message proper
Frame check[20]	16	Codified polynomial as per ISO/IEC 13239:20002 standard
End flag[21]	8	01111110
End buffer[22]	9	3 different bit sequences closing the time slot
TOTAL	256	

As the transmission speed is 9.6kps, a total of 26.67ms are required for a full AIS message to be transmitted (63).

[17] Sample of AIS message: !AIVDM,1,1,,A,H3GQ1u@ltu8TpN0<Tp<v2222220,2*202248
[18] Including the ramp-up bit sequence (8 bits)
[19] Announces the upcoming AIS message.
[20] Used for data message error-checking.
[21] Similar to the *start flag* sequence. It announces the ending of the AIS message.
[22] Including the ramp-down bit sequence (3 bits).

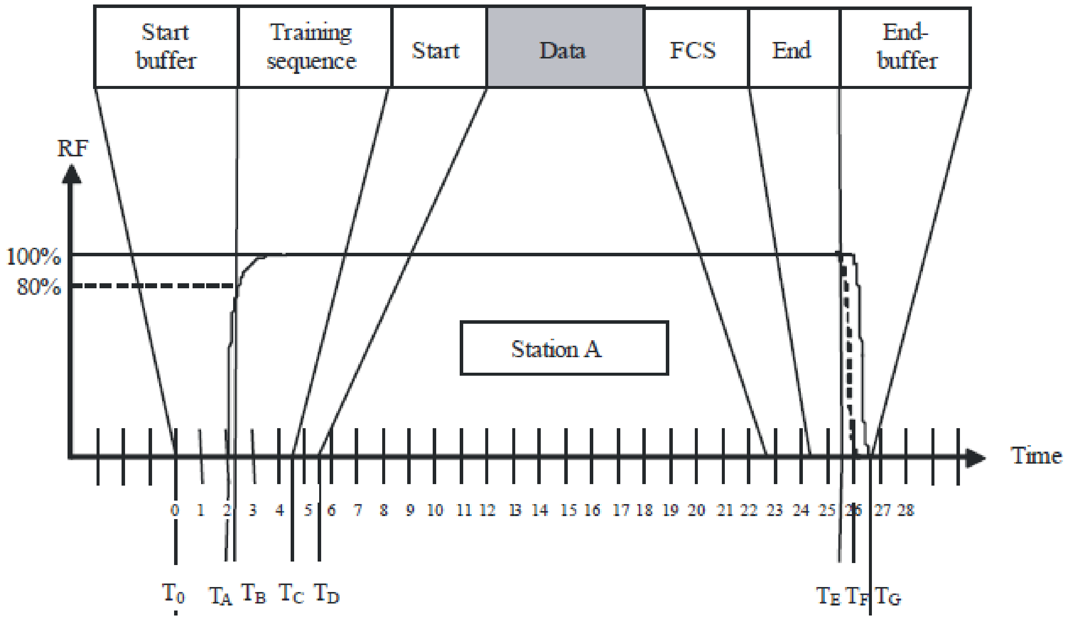

Figure 11. Dataset time allocation - SOURCE: ITU

Figure 11 shows the time allocated to each of the message data sequences. In short, the start buffer finishes at time T_B = 2396µs, the training sequence finishes at T_C = 4896µs, the start flag finishes at T_D = 5729µs, the data proper, FCS and end flag finish at T_E = 25729µs and the end buffer finishes at T_F = 26042µs. The total time slot finishes at T_G = 26667µs, which marks the beginning of the next message.

2.1.2. AIS messages

AIS message types are classified into 28 different messages IDs (63) based on the transmitted data, priority, access scheme and whether they are transmitted by a mobile (ship) or a static (shore) station.

For the sake of the project, AIS messages classified with IDs 1, 2, 3 (positions reports) and 5 (static and voyage data) were used. Table 9 lists their particulars.

Table 9. Description of AIS messages 1, 2, 3 and 5 - SOURCE: ITU

Message ID	Name	Description	Priority[23]
1	Position report	Class A scheduled report	1
2	Position report	Class A assigned scheduled report	1
3	Position report	Class A special report upon interrogation	1
5	Static and voyage-related	Class A scheduled report	4

Messages 1, 2 and 3 are position reports that contain essentially the same information. They only differ in the reason why the message has been sent (63). These messages are periodically being broadcasted only by mobile stations equipped with Class A AIS transceivers and include the information given in Table 10. As previously discussed in section 2.1 and Table 7 these messages are broadcasted every 2 seconds to 3 minutes, based on the dynamic condition of the vessel.

Table 10. AIS message 1, 2 and 3 information – SOURCE: ITU

Parameter	Bits	Description
Message ID	6	Message identifier, either 1, 2 or 3
Repeat indicator	2	May take any value from 0 to 3, where: 0 = default; 3 = do not repeat anymore.
User ID	30	Unique MMSI
Navigational status	4	May take any value from 0 to 15, where: 0 = underway using engine; 1 = at anchor; 2 = not under command; 3 = restricted maneuverability; 4 = constrained by her draught; 5 = moored;

[23] AIS messages are transmitted in 4 different priority categories, i.e. highest priority (level 1), highest service priority (level 2), interrogation (level 3) and lowest priority (level 4). Highest priorities are given to position reports and safety messages, which might be critical links for navigation; whereas, the lowest priority is assigned to all other messages.

		6 = aground; 7 = engaged in fishing; 8 = underway sailing; 9 = not in use[24]; 10 = not in use; 11 = power-driven vessel towing astern (regional); 12 = power-driven vessel towing ahead (regional); 13 = not in use; 14 = AIS-SART, MOB-AIS or EPIRB-AIS; 15 = undefined.
Rate of turn[25]	8	May take any value from -128 to +126, where: 0 to +126 = turning right at up to 708º/min; 0 to -126 = turning left at up to 708 º/min; +127 = turning right at more than 5º per 30s; -127 = turning left at more than 5º per 30s; -128 = ROT information not available (default).
Speed over ground[26]	10	May take any value from 0 to 1023, where: 1022 = 102.2 knots or higher; 1023 = speed not available.
Position accuracy	1	May take any value from 0 to 1, where: 0 = low accuracy (>10m); 1 = high accuracy (<10m).
Longitude[27]	28	May take any value from -180º to +180º, where: 0 to +180 = Easterly longitudes; 0 to -180 = Westerly longitudes; 181 = longitude not available (default)

[24] Numbers not currently in use are reserve for future SOLAS amendments.
[25] Given in ROTAIS, where ROTAIS = 4.733sqrt(ROT) in degrees per minute. Information provided from the ROT onboard sensor or a ROT calculator inputting data from COG.
[26] Given in 1/10 of a knot.
[27] Given in 1/10,000 of a minute.

Latitude[23]	27	May take any value from -90º to +90º, where: 0 to +90 = Northerly latitudes; 0 to -90 = Southerly latitudes; 91 = latitude not available (default)
Course over ground[28]	12	May take any value from 0 to 4095, where: 0 – 3599 = valid COG; 3600 = COG not available (default); 3601 – 4095 = not in use
True heading	9	May take any value from 0 to 511, where: 0 – 359 = valid headings; 360 – 510 = not in use; 511 = heading not available (default)
Time stamp[29]	6	May take any value from 0 to 63, where: 0 – 59 = valid reporting second; 60 = time stamp not available; 61 = positioning system in manual input mode; 62 = positioning system in dead reckoning mode; 63 = positioning system inoperative.
Special maneuver	2	May take any value from 0 to 2, where: 0 = not available (default); 1 = not engaged in special maneuver; 2 = engaged in special maneuver.
Spare	3	Not in use, set to 0
RAIM-flag[30]	1	May take any value from 0 to 1, where: 0 = RAIM not in use (default); 1 = RAIM in use.

[28] Given in 1/10 of a degree.
[29] UTC second when the report was generated. Further information is given in section 3.2.1
[30] Receiver autonomous integrity monitoring for electronic position fixing. Further information is given in section 3.2.1

Communication state	19	Data message containing SOTDMA and ITDMA protocols.
TOTAL	168	

Message 5 contains static and voyage-related data of a vessel or SAR aircraft equipped with a Class A AIS transceiver (63). As stated in section 2.1, these messages are broadcasted every 6 minutes and must be immediately transmitted when one of the vessel parameters changes. All the information contained in message 5 is listed in Table 11.

Table 11. AIS message 5 information - SOURCE: ITU

Parameter	**Bits**	**Description**
Message ID	6	Message identifier, either 1, 2 or 3
Repeat indicator[31]	2	May take any value from 0 to 3, where: 0 = default; 3 = do not repeat anymore.
User ID	30	Unique MMSI
AIS version	2	May take any value from 0 to 3, where: 0 = compliant with ITU-RM.1371-1; 1 = compliant with ITU-RM.1371-3; 2 = compliant with ITU-RM.1371-5; 3 = not in use.
IMO number	30	May take any value from 0 to 1073741823, where: 0 = IMO number not available (default); 0000000001 – 0000999999 = not in use; 0001000000 – 0009999999 = valid IMO number; 0010000000 – 1073741823 = official Flag State number.
Callsign	42	7x6 bit ASCII characters, where: @@@@@@@ = callsign not available (default)
Name	120	20x6 bit ASCII characters, where: @@@@@@@@@@@@@@@@@@@@ = not available

[31] Repeater indicator may be 0 = default, 3 = do not repeat anymore.

Type of ship and cargo[32]	8	May take any value from 0 to 255, where: 0 = type of ship not available (default); 10 – 19 = not in use; 20 – 28 = wing in ground (WIG) aircraft; 29 = search and rescue (SAR) aircraft; 30 = fishing; 31 – 32 = tugboat; 33 = dredger; 34 = dive vessel; 35 = military vessel; 36 = sailing vessel; 37 = pleasure craft; 38 – 39 = not in use; 40 – 49 = high speed craft (HSC); 50 = pilot boat; 51 = SAR vessel; 52 = tugboat; 53 = port tender; 54 = anti-pollution craft; 55 = law enforcement boat; 56 – 57 = local boat; 58 = medical transport; 59 = special craft; 60 – 69 = passenger vessel[33];

[32] For cargo vessels and tankers, second digits from 1 to 4 provide specific information on hazardous cargo carried on board, i.e. 1 = major hazard (cat. A), 2 = hazard (cat. B), 3 = minor hazard (cat. C) and 4 = recognizable hazard (cat. D).

[33] Includes accommodation barges and platforms, cruise vessels, floating hotels, houseboats, passenger tenders, Ro/Pax ferries and in general terms all floating crafts carrying more than 12 passengers. A cargo vessel carrying more than 12 passenger is to be considered a passenger vessel as per SOLAS requirements.

		70 – 79 = cargo vessel[34]; 80 – 89 = tanker vessel[35]; 90 – 99 = other; 100 – 199 = navigation aids and marks (including VTS); 200 – 255 = not in use.
Overall dimensions[36]	30	4 distances (A, B, C and D) are given, where: A = forward length; 0 – 511; 511 = 511m or longer; B = aft length; 0 – 511; 511 = 511m or longer; C = port beam; 0 – 61; 61 = 61m or longer; D = starboard beam; 0 – 61; 61 = 61m or longer.
Type of EPFD	4	May take any value from 0 to 15, where: 0 = type not available (default); 1 = GPS; 2 = GLONASS; 3 = GPS + GLONASS combined; 4 = Loran-C; 5 = Chayka; 6 = integrated navigation system; 7 = surveyed; 8 = Galileo; 9 – 14 = not in use; 15 = internal GNSS.
ETA[37]	20	Bits 19 – 16: month; 1 – 12; 0 = not available (default); Bits 15 – 11: day; 1 – 31; 0 = not available (default);

[34] Includes livestock carriers, dry bulk carriers (including mineral and cement), general cargo vessels, heavy lift vessels, container ships, Ro/Ro cargo vessels, car carriers, reefer vessels, cargo barges and in general all cargo vessels not carrying liquid cargo in tanks, and carrying less than 12 passengers.
[35] Includes asphalt tankers, chemical tankers, crude oil tankers, fruit juice and wine tankers, bunker and water barges, oil product carriers, LPG and LNG carriers, and in general all vessels carrying cargo in tanks.
[36] Given in terms of forward/aft and port/starboard distances from the point of reference (antenna). It includes also the calculated length overall (LOA = A+B) and the beam (B = C+D).
[37] Given in MMDDHHMM UTC format.

		Bits 10 – 6: hour; 0 – 23; 0 = not available (default); Bits 5 – 0: minute; 0 – 59; 60 = not available (default).
Draught[38]	8	May take any value from 0 to 255, where: 0 = draught not available (default); 255 = draught 25.5m or higher.
Destination[39]	120	20x6 bit ASCII characters, where: @@@@@@@@@@@@@@@@@@@@ = not available
DTE[40]	1	May take any value from 0 to 1, where: 0 = DTE available; 1 = DTE not available.
Spare	1	Not in use, set to 0
TOTAL	424	

Given that a full message 5 occupies more than the 168 available bits for data transfer, these messages have to be broadcasted in separated time slots (61)(63).

2.2. Vessels and air pollution

Maritime traffic is an important source of different air polluting substances, resulting from onboard combustion and energy transformation processes, mostly for propulsion and shipboard energy production (30)(31). Although emission per ship are not high, collectively vessel-generated emissions might be quite significant in heavy traffic areas, leading to issues in terms of air quality.

Smoke from vessels' funnels have been estimated to affect health and contribute to increased mortality within coastal communities. Several other problems, i.e. acidification, forest damage or corrosion have been also reported (31).

Among all the emissions related to vessels, oxides of sulfur and nitrogen do particularly have an important impact, as recent studies have shown that shore-related emissions of these pollutants

[38] In accordance with IMO Resolution A.851. Given in 1/10 of a meter. Maximum static draught to be set up during installation.
[39] IMO Resolution A.1106(29) recommend the use of UN LOCODE list as per ISO 3166 standard.
[40] Data terminal equipment. End instrument that converts data into signal and opposite.

have been reducing during the last years, while ship-related ones have been constantly increasing (32). Moreover, ship-related GHG emissions have also been growing in the last 10 years (32).

Emissions are mostly related to shipboard power, type of engine and fuel quality (69). In this section, first, a revision on marine engines for power production is discussed. Secondly, major ship-related air pollutants and their impacts are listed to eventually discuss the legal framework covering them and the air quality in Barcelona.

2.2.1. Marine power and engines

Shipboard power

Shipboard power is mostly required in order to provide the following services (34)(35)(36):

i. Propulsion power; and
ii. Auxiliary (or service) power.

On the one hand, propulsion power is required to move the vessel across the waters. Propulsion plants are made up of a prime mover (engine) and a propeller system, turning the heat energy into motion (35). Internal combustion engines, in different configurations, are the most popular source of shipboard propulsion (36). Slower, less maneuverable merchant vessels tend to use directly coupled engine-to-shaft power plants, powered by medium- and slow-speed diesel engines, whereas faster and highly maneuverable vessels tend to use geared- or diesel-electric power plants, in which medium- and high speed power plants are used (35).

On the other hand, auxiliary power is required for shipboard electricity and onboard services (35). The auxiliary power system is composed of auxiliary engines coupled to generators, power-takeoff shaft generators and boilers (35). The electric power demand on vessels is highly variable, depending on their main trade (35). Whereas passenger vessels have high hotel power demands, and require larger auxiliary engines or shaft generators; cargo vessels have lower demands that may be covered with smaller generators (36). In all cases, auxiliary engines tend to be mostly medium- to high-speed diesel engines (34)(37). Regarding boilers, they are required to produce steam for cargo and fuel lines heating, and hot water for shipboard services (37). Lately, turbogenerators have been installed to reduce the number of boilers and reuse the residual heating from the engine exhaust (35).

Marine engines

Internal combustion engines dominate the scene as a main source of marine power (35)(38). As of 2020, around 99% of worldwide fleet is powered by some source of internal combustion engine configuration, whereas only 1% rely on steam turbines and a residual 0.1% of global fleet is powered by gas turbines (36).

Traditionally, fuel-fired steam turbines were the most popular source of shipboard power (35). However, the rise in fuel costs in the mid-1900s triggered their fast demise, being rapidly replaced with more efficient internal combustion engines (35). Currently, the use of steam turbines have been relegated to gas boil-off fueled turbines on LNG tankers and nuclear-powered aircraft carriers and submarines (36).

Gas turbines were installed in the early 2000s mostly in high-speed war ships and some cruise ships trading in environmentally-sensitive areas, given their lower emission rates (35). Unfortunately, they proved to be extremely unpopular due to their low efficiency and high operational costs, and were mostly replaced with diesel engines by mid-2010s (35). Recently, the turbogenerator, a spinoff of traditional gas turbines, has been used on ships equipped with combined cycle power plants, so as to increase their efficiency by recycling waste heat from internal combustion engines.

HFO-fueled internal combustion diesel engines have been the most popular marine engine within the maritime industry for the last half century (35)(36)(38). Their versatility, efficiency and cost-effective performance have been among the main reasons for their popularity. Current stricter fuel quality regulations lead the transition towards LNG-fueled engines and dual LNG/MGO-fueled engines (36).

Table 12 lists the most common fitted engines per ship type, as of 2010.

Table 12. Most common fitted engines (2010) - SOURCE: EMEP / EEA 2019

Type of ship	Diesel engines			Steam turbines	Gas turbines
	SSE	MSE	HSE		
Tankers	74.95%	23.64%	1.27%	0.14%	0.00%
Bulk carriers	92.00%	7.92%	0.08%	0.00%	0.00%
Container vessels	94.21%	5.67%	0.11%	0.00%	0.00%
General cargo vessels	44.95%	50.19%	4.75%	0.00%	0.10%
Ro/Ro vessels	20.26%	69.68%	7.80%	0.00%	2.27%
Passenger vessels	3.81%	82.66%	5.44%	0.00%	8.08%
Fishing vessels	0.00%	88.24%	11.76%	0.00%	0.00%
Tugboats	0.00%	46.13%	53.58%	0.00%	0.28%
Others	30.62%	49.17%	3.34%	0.00%	0.58%

On the one hand, pure merchant vessels tend to be slower and heavier, thus they require heavier powers to thrust their hulls through the waters, and slow-speed engines are the best option (36). On the other hand, faster and more maneuverable vessels, such as Ro/Ro, passenger vessels, general cargo vessels and fishing boats, are fitted mostly with medium-speed engines (36). High-speed engines are only common in tugboats, given their high maneuverability and smaller size. Gas turbines show some small presence in passenger and Ro/Ro vessels, whereas steam turbines are fitted residually only on tankers (36).

Out of a global merchant fleet of 53,000 vessels, only 117 were powered by LNG in 2019 (36). However, this figure is scheduled to increase as a total of 111 LNG-powered vessels were on order by the same time. Heavy fuel oil and more distillated marine diesel and gas oil still dominate the scene.

Table 13 presents pre-2020 most common fuels by ship type.

Table 13. Pre-2020 most common fuels - SOURCE: EMEP / EEA 2019

Type of ship	HFO	MDO/MGO
Tankers	95.44%	4.56%
Bulk carriers	98.94%	1.06%
Container vessels	98.63%	1.37%
General cargo vessels	86.86%	13.14%
Ro/Ro vessels	82.13%	17.87%
Passenger vessels	85.85%	14.15%
Fishing vessels	3.82%	96.18%
Tugboats	6.93%a	93.07%
Others	52.93%	47.07%

Diesel engines

Most of world's fleet relies on reciprocating diesel engines as main source of onboard power and propulsion (36). Marine engines can be classified in terms of cycles, i.e. 2-stroke or 4-stroke engines, and in terms of speed, i.e. slow-, medium- and high-speed engines (34). However, the most common configurations are either 2-stroke slow-speed crosshead engines, used on larger yet slower vessels, or 4-stroke medium-speed trunk engines, used on faster and more maneuverable vessels (38).

Following the traditional crankshaft speed description, the following may be noticed on vessels:

i. Slow-speed engines represent around 18% of total diesel engines fitted on board (36). The marine counterpart consists mostly of 4 to 12 cylinders, commonly working on the 80 to 140 rpm range (35). Given their low revolutions, they are only used for propulsion directly coupled to the propeller shaft on board large and slow merchant vessels (35). Their installed powers are the highest among the industry, so their emissions do; given their slow crankshaft speed (34).
ii. Medium-speed engines represent around 55% of total diesel engines fitted on board (36). They consists mostly of 12 to 20-cylinder engines working on the 300 to 900rpm range. Given their higher speed, they can both be used for propulsion, when geared to the propeller shaft, and to generate shipboard power (35). Owing to the same reasoning, medium-speed engines pollute less than their slower counterpart (34).
iii. High-speed engines represent around 27% of total diesel engines fitted on board (36). They run at speeds higher than 1,000 rpm and are smaller than their slow- and medium-speed counterparts (35). Despite their popularity, they are mostly used only on smaller crafts and as auxiliary engines (34).

Steam turbines

Steam turbines are rotatory heat engines that transform the thermal energy contained in pressurized heated steam into mechanical energy (35). Steam turbines were once the most popular means of propulsion on ships. However, their lower efficiency, higher operating costs and handling against diesel engines resulted in their disappearance by 1970s (36).

Steam turbines conform a much larger system, which includes a fuel-fed boiler that heats water into steam. The emissions arising from them are actually related to the fuel burnt in the boiler. Although the use of steam turbines within the maritime industry is currently residual, boilers are still deployed on board for auxiliary services (35).

Steam turbines have been traditionally used in vessels requiring high propulsive powers. Given that they achieve better efficiencies at rotatory speeds higher than 1,000rpm, direct coupled ship propulsion is unfeasible and geared drive is always required (35). Hence, modern vessels fitted with steam turbines, rely usually on turbo-electric propulsion. Currently, steam turbines are installed on LNG and ULCC tankers, powerful icebreakers, submarines and navy vessels.

Gas turbines

Gas turbines are rotatory internal combustion engines (38), which working fluid is air. They consist of a single unit comprising a gas compressor, a combustor and a rotatory downstream turbine. Marine gas turbines work in power ranges of 4 to 30MW, which makes them suitable for medium-sized high-speed vessels, with high power density demands, such as navy ships (35). However, they have lower efficiencies, higher fuel consumptions and require highly refined fuels, which result in expensive operating costs. For this reason, their use within the marine industry is merely residual.

Several cruise ships built in the late 1990s and early 2000s were fitted with gas turbines in order to comply with strict rules regarding air emissions (36). However, their high operational costs and the further development of more environmentally friendly diesel engines resulted in their replacement by late 2010s.

2.2.2. Marine exhaust gases

Marine engines produce, to a different extent, exhaust gases composed mostly of water vapor (H_2O), carbon dioxide (CO_2), nitrogen gas (N_2), oxygen gas (O_2), carbon monoxide (CO), oxides of sulfur (SO_X) and oxides of nitrogen (NO_X) and particulate matter (PM), among others (39)(40).

Out of these flue gases, water vapor, N_2 and O_2 in normal concentrations are not toxic to human being (41). CO_2 is also not toxic, however it has a powerful greenhouse effect, with a negative impact on human life (42). The remaining three, SO_X, NO_X and PM are noxious gases, which have been proved to increase mortality and respiratory illnesses among population (41).

The composition of exhaust gases is variable in terms of engine phase, whether transitory- or steady-state; engine characteristics and type of fuel (43). Modern marine engines are built complying with the latest international standards and regulations, in terms of air pollutant emissions. However, the fuel quality burnt on board is still subject to controversy, as its poor refining level is partly responsible for the success of global shipping (37).

Carbon dioxide (CO_2)

CO_2 is along water vapor the largest effluent resulting from combustion (39). The amount of CO_2 produced is a ratio of the amount of burnt fuel, therefore it depends on the engine power demand and its efficiency (43).

CO_2 is labelled as a major green-house effect gas (GHG); given that it represents up to 82% of total GHG gases in the atmosphere (42). The main issues related to an increase in GHG concentration include an increase in global atmospheric temperature, higher sea water level, ocean acidification and carbon cycle alterations (41).

It is mostly generated in industrial processes, energy production, and transportation, which represented up to 57.9% and 21.9% in EU-27 during 2018 (44). International shipping has been estimated to produce around 2.2% of total CO_2 worldwide (45). However, different sources cite that shipping may represent 4% to 6% (15). Several studies have found that slow steaming might reduce significantly carbon dioxide emissions from shipping (45). For instance, a -10% reduction in speed in the worldwide fleet, may cut shipping carbon footprint by -20%, from current values.

Nitrogen and oxygen gas (N_2 / O_2)

Both N_2 and O_2 are found naturally as free gases within the engine intake air (39). Originally, N_2 represents up to 78% of total intake air and except of residual reactions with O_2 and sulfur to form NO_X and nitrogen sulfur, N_2 is mostly found unreacted within the exhaust gases (39). Furnace temperature and excess intake air are among the reasons that trigger the reaction of N_2 with other present gases (40). O_2 represents up to 21% of total intake air. Contrary to N_2, it mostly reacts with fuel during combustion (39). Therefore, free O_2 traces (> 1%) are only found as a result of excess air intake (43).

Excessive amounts of free N_2 and O_2 traces in exhaust gases result from improper combustion due to excess air intake. Out of the combustion chamber, free N_2 may react with free O_2 and water vapor producing oxides of nitrogen, which in recombination with oxygen may produce ozone (O_3) (43).

Carbon monoxide (CO)

CO results from incomplete combustion of fuel. This is due to excess air intake, inappropriate furnace temperature or poor engine maintenance (39). Generally, it is a result of improper engine management (37).

CO is a GHG with milder effects on the environment if compared to carbon dioxide or water vapor (41). However, it is extremely harmful for human being as people exposed to it for prolonged period of times might experience shortened breath (41).

Oxides of sulfur (SO_X)

SO_X are generated during fossil-fuel combustion due to the sulfur traces found in poor refined fuels, like coal or heavy fuel oils (39). During combustion, sulfur reacts with O_2 to generate mostly dioxide of sulfur (SO_2) and, to a lesser extent, trioxide of sulfur (SO_3) (40). In fact, SO_2 may represent up to 97% of total SO_X compared to a residual 3% of SO_3 (43).

Major issues related to SO_X include an increase in breath-related illnesses, unstable crop health, damage on building materials and acid rain, leading to ocean and ground acidification (41). The effects of SO_X can be noticed in many heavily congested metropolis in form of chemical smog (43).

Recent studies have estimated that the shipping industry contributes with 13% of total SO_X emissions worldwide (15)(46), more than China and US mainland together. With the latest regulations dealing with stricter sulfur contents in fuel, SO_X arising from marine activates are estimated to reduce to 5% to 6% of global emissions (46).

Oxides of nitrogen (NO_X)

Fossil fuel combustion is responsible for about 90% of total atmospheric NO_X (43). They are found in the atmosphere as monoxide of nitrogen (NO) and dioxide of nitrogen (NO_2), in variable ratios based on solar radiation, atmospheric temperature and ground-level O_3. In general, NO represents 90% of total NO_X, whereas NO_2 may represent up to 10% (43).

During combustion, NO_X are produced through two different processes, namely thermal-induced reactions and fuel-induced reactions (39)(40). On the one hand, thermal NO_X are generated due to the recombination of excess N_2 gas and O_2 gas in intake air due to high furnace temperature. On the other hand, fuel-related NO_X are generated due to traces of N_2 present in the fuel. As the formation of NO_X is related to the residence time of the burnt gas at high temperature in the combustion chambers, low speed engines produce more NO_X than medium- and high-speed engines (35).

High concentration of NO_X may increase the rate of lung-related illnesses (41), due to their recombination with atmospheric O_2, resulting in ground-level O_3. They may further react, generating acids of nitrogen, which are responsible for ocean acidification (41). Shipping has been estimated to be responsible for 15% of worldwide emissions of NO_X (15), as much as India and US mainland together.

Particulate matter (PM)

PM is a complex mixture of both organic and inorganic compounds, resulting from different sources (43). Major sources include incomplete combustion, unburnt traces of lubricant oil and fuel, thermal splitting of hydrocarbon, ashes and sulfates, among others (39). PM within a diameter range between 2.5µm and 10µm conforms the so-called inhalable particles group (41).

Two mechanisms are responsible for the formation of PM during combustion, namely nuclei- and accumulation-mode particles (43). Nuclei-mode particles are formed through condensation of flue gases, as exhaust temperature goes down and it mixes with free atmospheric air; this includes sulfates and free unburnt hydrocarbons. Accumulation-mode particles are formed in the combustion chamber, by combination of carbon and other solids present in the fuel; this forms the infamous black carbon or snoot, typically flowing out from vessel funnels (40).

Inhalable PM is carcinogenic (41). PM emissions related to ships have been found to be responsible of 60,000 excess breath-related deaths worldwide (41).

2.2.3. Legal framework

The legal framework covering international shipping emissions consists of three main pillars:

i. MARPOL Annex VI (51);
ii. EU Sulfur Directives (48)(49); and
iii. EU Monitoring, Reporting and Verification Program (EUMRV) (50).

Latest revisions of the aforementioned rules entered into force on January 1, 2020; having special impact on SO_X and carbon footprint programs (49)(50)(51).

MARPOL Annex VI, approved in 1997 by the Commission and entering into force in 2005, focuses on the prevention of air pollution from shipping (51). It includes 25 regulations, of which the following deal with emissions from major air pollutants:

i. NO_X, chapter 2 - regulation 13;
ii. SO_X and PM, chapter 2 - regulation 14;
iii. Volatile organic compounds (VOC), chapter 2 - regulation 15;
iv. Fumes from incineration, chapter 2 - regulation 15; and
v. CO_2, chapter 3 - regulations 19 to 23.

Fulfillment of MARPOL Annex VI is mandatory for all vessels of more than 400GT, which are subject to inspection against all existing regulations leading to a 5-year valid IAPP certificate (51). A second certificate, the IEEC, is also required for vessels of more than 400GT engaged in international voyages, proving the fulfillment of Chapter 3 regulations (51).

Complementing the existing MARPOL Annex VI regulations, the EU Sulfur Directives and the EUMRV, create a regulatory framework within the EU (49)(50). They seek to reduce SO_X emissions (49) and limit the shipping industry carbon footprint (50). They work on a similar scope as MARPOL, implementing a minimum fuel quality level with regards to sulfur content and a reporting system in order to create a proper ship-based CO_2 inventory database. The EU Sulfur Directives apply to all EU-flagged fleet and to all vessels trading within EU territorial waters (49), whereas the EUMRV applies to EU-flagged vessels and those trading within EU territorial waters over 5,000GT (50).

Carbon dioxide (CO_2)

Both MARPOL Annex VI and EUMRV regulations seek to reduce the carbon footprint of the shipping industry through different programs (50)(51). For instance, MARPOL Annex VI provides a series of indexes to monitor the vessel carbon footprint, namely the EEDI and the EEOI; and tools to implement shipboard energy management plans, the SEEMP. Meanwhile, the EUMRV is a mandatory reporting system, aiming to create an accurate inventory of vessel emissions (50). Certification is required against compliance of MARPOL CO_2 regulations, the IAPP certificate; and EUMRV; the EUMRV Document of Compliance.

The goal of the EEDI program is to implement a quantitative index, allowing vessels to calculate their efficiency in terms of carbon footprint, aiming to reduce their impact by adjusting the minimum required EEDI threshold every 5 years (52). The required EEDI index is dependent on the vessel deadweight tonnage and type, whereas the attained EEDI depends on the installed power. As of January 1, 2020 and for a period of 5 years, the 2nd tier, corresponding to a reduction factor of 20% is in force (52). Around 85% of worldwide fleet is subject to the EEDI program (52). An optional index based on the attained EEDI value, the EEOI, is fitted for the remaining 15% of global fleet, so as to provide a tool to control and manage the whole shipping industry carbon footprint (52).

In order to help vessels meet IMO requirements in terms of carbon footprint, the SEEMP was implemented for all existing vessels on January 1, 2013 (51). It aims to establish a series of shipboard procedures leading to increased vessel efficiencies and reduced CO_2 emissions. The plan is subject to constant revision and includes several measures, i.e. enhanced voyage planning and weather routing, optimized shaft power, optimized trim, improved autopilot usage, etc. (54) Several studies concluded that proper SEEMP use helped cut CO_2 emissions by -34% since entering into force (55).

Finally, the EUMRV consists of an approved reporting system, through which EU-flagged vessels and those trading within EU ports are required to notify their sailed miles, time, power and fuel consumption per voyage (50). Based on approved SFOC and emission factors certified by partnering classification societies, the EUMRV program supervisors are able to calculate the vessel emissions, produce inventories and monitor their carbon footprint.

Oxides of sulfur (SO_X)

Given that fuel sulfur content is proportional to the emitted oxides of sulfur (39)(51), the legal basis is based on regulating the fuel quality or fuel refining level, i.e. its sulfur content (m/m)[41]. MARPOL Annex VI provided the basics to an enhanced regulatory framework (51) in force within the EU until January 1, 2020; when both regulations came to a similar restrictive level (51)(49).

The following are a series of emission control areas (ECAs), established by MARPOL Annex VI, where reduced sulfur contents are enforced (51):

i. The Baltic Sea area;
ii. The North Sea area;
iii. The North American area; and
iv. The US Caribbean Sea area.

As seen in Table 14, current strictest regulations entered into force in January 1, 2020; heavily limiting sulfur content in fuels outside of ECAs.

Table 14. IMO requirements on SOx emissions – SOURCE: MARPOL A-VI

Date	Outside ECAs	ECAs
	Sulfur content (m/m)	Sulfur content (m/m)
Before July 1, 2010	4.50%	1.50%
On or after July 1, 2010 but before January 1, 2012	4.50%	1.00%
On or after January 1, 2012 but before 2015.	3.50%	1.00%
On or after 2015 but before January 1, 2020	3.50%	0.10%[42]
On or after January 1, 2020	0.50%	0.10%

Within the EU, the 2005-approved Directive 2005/33/EC and the 2012-approved Directive 2012/33/EU create the legal basis regulating fuel quality for EU-flagged vessels and vessels trading within EU ports (48)(49). Jointly, they are referred to as the Sulfur Directives.

[41] Acronym for mass/mass.
[42] However, vessels built on or before August, 1 2011 and operating within the North American or the US Caribbean Sea are not required to fulfill these requirements until January 1, 2020.

Table 15. EU sulfur content requirements for marine oils - SOURCE: Directive 2012/33/EU

When?	From 2015	From 2020
Within SECA's[43]	0.10% (m/m)	0.10% (m/m)
Outside SECA's[44]	3.50% (m/m)	0.50% (m/m)
Special req. for pax. vessels[45]	1.50% (m/m)	0.50% (m/m)
Berthed[46]	0.10% (m/m)	0.10% (m/m)

As part of the Horizon 2020 program, in Table 15, the latest tier entered into force on January 1, 2020; harmonizing both the EU and worldwide fuels in terms of sulfur content. The most restrictive fuel quality levels forced the transition towards new LNG- or MGO-fueled engines or the installation of scrubbers in order to comply with the sulfur content requirements. With this program, shipping-related SO_X emissions are expected to reduce by up to -77% (46).

Oxides of nitrogen (NO_X)

Regulations covering NO_X emissions are based on installed power (39)(51). All vessels with a power output of more than 130kW may comply with them (51). The regulatory framework is organized in 3 tiers based on the construction date and crank-shaft revolutions, i.e. covering slow-, medium- and high-speed engines.

As seen in Table 16, the current strictest tier, Tier III, is expected to be fully in force globally by January 1, 2021 as it currently applies only to the US Caribbean Sea emission control area.

[43] Applies from January 1, 2015. Previously, 1.00% (m/m).
[44] First tier applies from June 18, 2014. Reduced from January 1, 2020.
[45] Applies to passenger vessels deployed on scheduled routes from/to or within EU Member States, except when they are engaged in navigation within SECA's. This does not apply to cruise r vessels (Rodrigo J., 2011)
[46] Does not apply to vessels berthed less than 2 hours or those using shore-based power.

Table 16. IMO requirements on NOx emissions - SOURCE: MARPOL A-VI

Crank-shaft revolutions (rpm)	TIER I Before January 1, 2011	TIER II Before January 1, 2016	TIER III After January 1, 2016[47]
	NO_X emissions (g/kWh)	NO_X emissions (g/kWh)	NO_X emissions (g/kWh)
n < 130	17.0	14.4	3.4
130 ≤ n < 2000	$45 \cdot n^{(-0.2)}$	$44 \cdot n^{(-0.23)}$	$9 \cdot n^{(-0.2)}$
n ≥ 2000	9.8	7.7	2.0

2.3. Air quality in Barcelona

Greater Barcelona and the city proper rank among the top most polluted areas within Spain and the EU, in terms of NO_X and CO_2 (56). The traditional industrial core of Greater Barcelona and its surroundings together with the geographical and weather features of the Catalan coast explain this singularity (24)(56). Several 10-year plans have been developed by the City Council, aiming to reduce the city carbon footprint and improve its air quality (56). Despite all efforts, the city still ranks among the most polluted metropolis in continental Europe. In fact, for the last 10 years, as shown in Figure 12, Barcelona exceeded the average annual maximum allowed concentration of NO_X in all but one of the existing air quality monitoring stations (24)(57).

Several reports have indicated that in terms of NO_X, wheeled traffic may represent up to 65.6%, background dispersed emissions may contribute with up to 21.8%, commercial and housing may account for up to 8.6%, industry for up to 4.8%, shipping and port services up to 2.1% and aviation up to 0.1% (56).

In terms of CO_2, Barcelona is estimated to emit up to 3.7M tons of CO_2 per year; or 2.28 tons per capita (24). By sectors, wheeled traffic represents ca. 27.4% of total emissions, commercial services account for ca. 20.6%, housing for up to 20.4%, waste treatment for up to 10.7% and industry up to 7.7% (56).

[47] Applying only to vessels engaged in navigation within or to/from the US Caribbean and North American ECA's. It does not apply to recreational crafts of 24m or less (L) and vessels built before January 1, 2021 of less than 500GT and 24m or more (L).

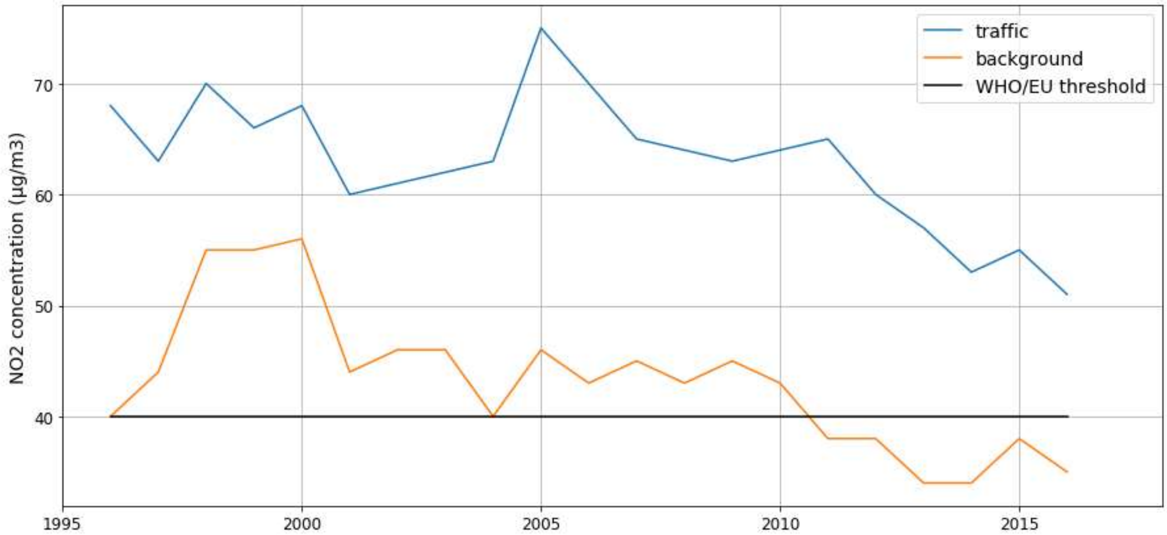

Figure 12. 20-year average NO_2 levels within Barcelona – SOURCE: Barcelona City Council

Regarding particulate matter, as in Figure 13, values have been reduced during the last 10 years. Background dispersion emissions may represent up to 71% of total PM, wheeled traffic generates around 20.8%, commercial and housing account for up to 6.4%, port activities represent 1.5% and industry accounts for a residual 0.3% (24)(56).

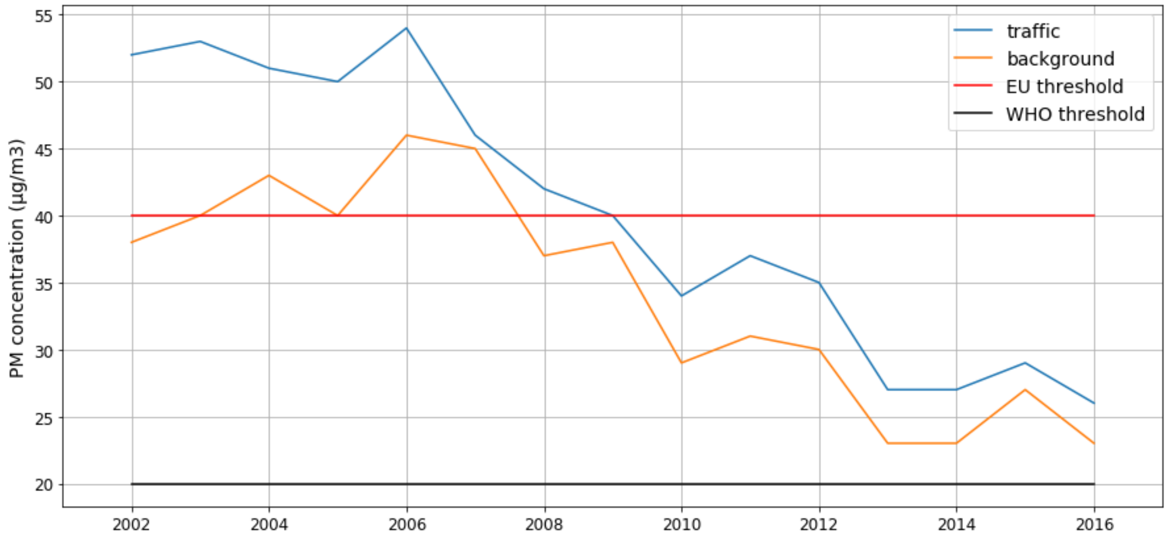

Figure 13. 15-year average PM_{10} levels within Barcelona – SOURCE: Barcelona City Council

SO_X emissions are relatively low, as common sources are not present within the city limits (13). Port activities have been recently assessed to be a major contributor of this pollutant, with special emphasis on cruise ships and passenger ferries which have been recently identified as a major source (58).

The port itself is responsible for around 7.60% of total NO_X and 1.50% of PM_{10} within the city (24)(56). Road traffic is actually the major pollutant related to port emissions, contributing with 89% of NO_X levels and 77% of PM_{10} emissions (56). Regarding pure vessel emissions, Ro/Ro, container and cruise vessels are major sources of air pollutants. Ro/Ro vessels account for up to 2.00% NO_X and 0.48% PM_{10}, container vessels represent 1.70% NO_X and 0.38% PM_{10} and passenger cruise vessels may account for up to 1.20% of NO_X and 0.23% of PM_{10} levels (24)(56).

2.3.1. Air quality changes related to COVID-19

The confinement measures adopted in Barcelona during the early COVID-19 outbreak resulted in a reduction in the overall concentration of major air pollutants (13)(59). The most notorious reduction was recorded in NO_X , as shown in Figure 14, and CO levels, which dropped by -70% and -50% (59), respectively during strictest lockdown days. However, these reductions have merely been temporary, thus a real improvement of air quality is unlikely. In fact, post-COVID-19 measures could even result in a short-term increase in air pollution (59), as there is a strong negative causality relation between crisis management and environmental policies.

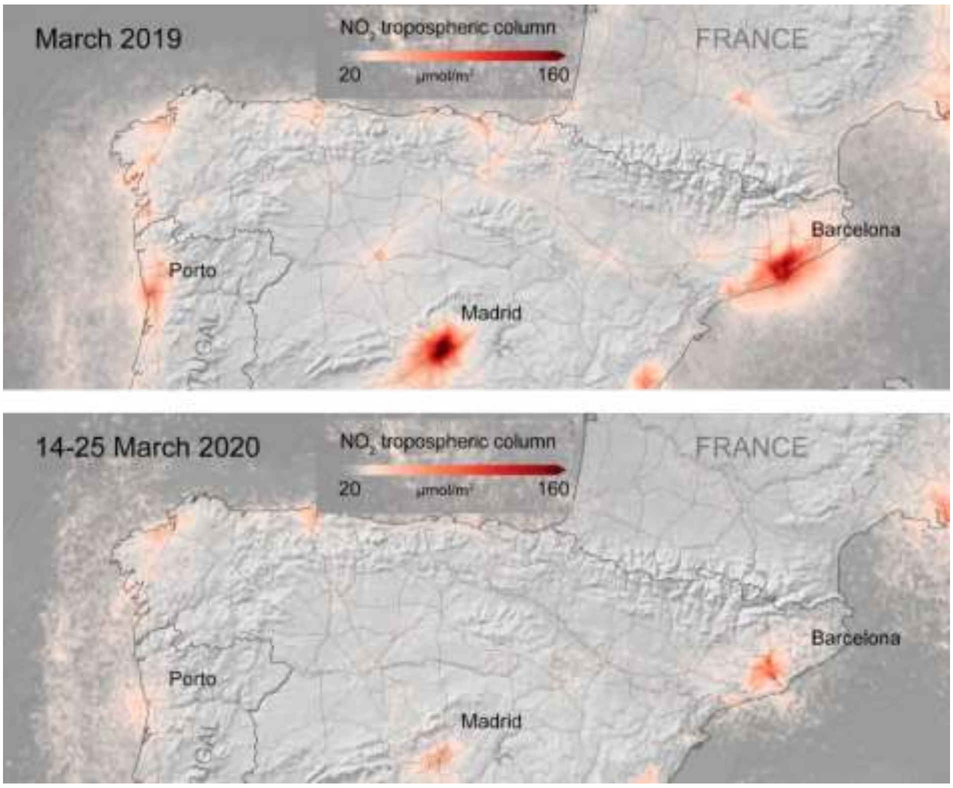

Figure 14. Satellite NO_2 tropospheric column concentration in northern Spain – SOURCE: ESA

An early study conducted during the first two lockdown weeks in Barcelona (13) recorded -28% and -31% reductions in background and traffic-related PM_{10}; and -47% and -51.4% reductions in background and traffic-related NO_2, as stated in Table 17. However, minimum improvement was reported for SO_2 emissions. For a long period of time, Barcelona have been reporting PM_{10} concentration values well below the WHO air quality threshold of 20µg/m^3 (13).

Table 17. Mean concentration variation of major pollutants from February 16 to March 30 – SOURCE: IDAEA-CSIC

Air pollutant	Before lockdown	During lockdown	Variation	
			µg/m^3	%
Urban background				
PM_{10}	22.4	16.2	-6.2	-27.8
NO_2	30.0	15.9	-14.1	-47.0
SO_2	1.2	1.0	-0.2	-0.2
Traffic-related				
PM_{10}	29.2	20.2	-9.1	-31.0
NO_2	42.4	20.6	-21.8	-51.4
SO_2	2.5	2.6	0.1	+1.8

Reductions in NO_2 were explained mostly due to the reduced wheeled traffic activity within the city limits and Greater Barcelona (13). Whereas, lower PM_{10} levels were strongly related to less road traffic and power generation, due to reduced industrial activity (13). Theoretically, a more aggressive PM_{10} reduction, similar to NO_2 levels; was expected (13). However, meteorological conditions[48] within the early lockdown days explain the given values and indicate that PM_{10} levels in the city are heavily influenced by regional-background origin, mostly related to air mass transportation.

As previously discussed, SO_2 levels are relatively low in Barcelona (13), compared to other major European cities. Therefore, the minimum variation was mostly related to changes in maritime traffic within the region (13).

[48] Weather plays an important role in atmospheric pollution dispersion. During the period of time between March 14 and March 30, an episode of Saharan dust affected the area of Barcelona.

(This page intentionally left blank)

Chapter 3. Methods

This chapter focuses on the methodologies followed to study the evolution of maritime traffic and port calls, and assess the emissions from vessels through an emission inventory due to the impact of SARS-CoV-2 virus.

The maritime traffic and port calls evolution was assessed through further filtering of AIS datasets, presenting the information from different point of views, so as to analyze all possible scenarios. Extensive information on how data was filtered is further given within the following lines.

Emissions were computed for every ship following the renowned STEAM v.2 algorithm, developed by Jalkanen et al. in 2009, using real technical data from the IHS database. As lacking a thorough database with vessel technical information was a major setback at the beginning, this chapter also includes a brief description of a secondary model to calculate emissions through equations adapted to compute vessel installed powers based only on AIS-provided information.

3.1. Maritime and port traffic analysis

Once filtered out and processed, all available AIS messages were further analyzed through a series of filters coded in the free license *Python* language. It was selected over other options given its free access, general purpose construction and clear structure, which allowed a logic and accurate data assessing. The following *Python* libraries were required: *pandas* for data analysis*, numpy* for matrix and arrays analysis and *matplotlib* for plots, *folium* for live maps, among others.

The following codes were developed and are available in Annex A1 for further reference:

i. AISdata.py – reads and plots all vessels in range –;
ii. AISdata_map.py – reads and plots in a semi-live map all vessels in range –;
iii. AISdata_status.py – reads and assesses the change of status for all vessels in range –;
iv. AISspeed.py – reads and assesses the change in speed for all vessels in range –;
v. AIScalls.py – reads all vessels and analyses the number of calls in Barcelona –; and
vi. AISdraft.py – reads and assesses the change in draft for all vessels in range.

3.1.1. How AIS data is obtained?

Initially, all acquired AIS data was filtered out so as to be presented in a readable and easy-to-access manner. For the project, data was already treated and ready to be further filtered and assessed. However, within the following lines a brief description is given on some important steps and how data was initially presented.

Data in bits was decoded and filtered out, replacing missing information with NaN data, based on the parameters stated in section 2.1.2, such as incorrect position, driven from GPS errors and incorrect reported speeds.

The time assigned to each dataset corresponded to the receiving time at the AIS station located at the Barcelona School of Nautical Studies, stamped with the UTC second included in the message for dynamic messages. This means that the processed message actually contains only the UTC second at which it was broadcasted, the so-called time stamp, as seen in section 2.1.2 in Table 10. Out of all received reports, a 16% reception rate error was common, as the time stamp second differed greatly from the UTC reception second.

Regarding position, accuracy was checked through the receiver autonomous integrity monitoring for electronic position fixing, in short RAIM. Out of all received reports, a 37% of them had an accuracy of less than 10m. However, for the sake of the project, this error is more than acceptable.

AIS messages 1, 2 and 3 were afterwards stamped together in a single comma separated value (.csv) file, with all the initially filtered information. This file is ready to be imported into *python* as a dataframe. Figure 15 shows a sample of Class A messages 1, 2 and 3 for March 2020; containing the reception date (UTC), second sent, station MMSI, status, rate of turn, speed, longitude, latitude, course over ground and heading.

	A
1	date,second_sent,mmsi,status,turn,speed,lon,lat,course,heading
2	20200301000000,58,247243600,0,0.0,0.0,2.17512,41.363820000000004,33.0,33
3	20200301000000,0,224713000,5,0.0,0.0,2.18275,41.370979999999996,329.6,30
4	20200301000001,55,209293000,5,0.0,0.0,2.17313,41.35071,100.7,26
5	20200301000002,1,224878000,5,-128.0,36.0,2.17696,41.36235,122.3,511
6	20200301000002,0,229866000,0,0.0,78.0,2.17758,41.30095,339.9,333
7	20200301000002,1,225394000,15,-128.0,58.0,2.1663799999999998,41.330729999999996,8.7,511
8	20200301000002,59,224026000,15,0.0,0.0,2.15171,41.34588,135.7,2
9	20200301000002,2,224324620,7,-128.0,0.0,2.18459,41.37323,112.8,511
10	20200301000003,1,225423000,0,0.0,180.0,2.25558,41.21787,170.9,170
11	20200301000003,2,211382280,0,0.0,77.0,2.43803,41.371390000000005,61.1,59

Figure 15. Example of class A AIS message 1, 2 and 3 dataset

AIS messages 5 were presented in a text (.txt) file, with all the initially filtered information. This file can be easily accessed through *python* as a dataframe. Figure 16 is a sample of Class A message 5 for March 2020; containing the reception date (UTC), message ID, station MMSI, station name, forward length from the antenna, aft length from the antenna, port breadth from the antenna, starboard breadth from the antenna, draught and destination.

```
20200301000019,5,229561000,IMERA                , 80,114.0, 29.0, 11.0, 10.0,  9.0,ESBCN
20200301000021,5,319080700,AZAMANTA             , 37, 27.0, 28.0,  5.0,  4.0,  3.0,.
20200301000021,5,224368920,PILAR Y MARIA        , 30, 16.0,  4.0,  3.0,  3.0,  0.0,
20200301000021,5,229561000,IMERA                , 80,114.0, 29.0, 11.0, 10.0,  9.0,ESBCN
20200301000023,5,319134300,VOLPINI 2            , 37, 20.0, 38.0,  4.0,  6.0,  3.5,BARCELONA
20200301000036,5,224799000,GARCIA DEL CID       , 33, 10.0, 20.0,  6.0, 12.0,  4.8,BARCELONA
20200301000036,5,224327160,GREENOIL             , 80, 67.0, 10.0, 10.0,  8.0,  4.2,BARCELONA
20200301000046,5,319055000,AURORA               , 37, 30.0, 20.0,  5.0,  5.0,  3.2,BARCELONA
20200301000054,5,224713000,SARMIENTO DE GAMBOA ,  3, 18.0, 53.0, 11.0,  5.0,  7.0,BARCELONA
20200301000113,5,319094900,DILBAR               , 37, 58.0, 98.0, 12.0, 12.0,  6.0,SEA TRIALS
20200301000133,5,374859000,MSC MAYA             , 71,145.0,251.0, 36.0, 23.0, 13.1,BARCELONA
20200301000157,5,255804280,HARBOUR FEATURE      , 89,  1.0,144.0, 12.0, 11.0,  6.2,ESBCN
20200301000159,5,210688000,AMAZONITH            , 89, 73.0, 20.0,  6.0,  8.0,  5.3,ES BCN
20200301000159,5,255804280,HARBOUR FEATURE      , 89,  1.0,144.0, 12.0, 11.0,  6.2,ESBCN
20200301000202,5,224324620,CABRERA DOS          , 30,  4.0, 16.0,  4.0,  2.0,  4.0,
```

Figure 16. Example of class A AIS message 5 dataset

3.1.2. Position filtering

VHF waves propagation

As AIS messages are transmitted through VHF electromagnetic waves, its actual range is limited in terms of frequency, power and height of the transceiver unit (61). As defined by the ITU, VHF ranges from 30MHz to 300MHz, with the maritime VHF band assigned to the range between 156MHz and 174MHz (61), both included. Transmitting power is limited to 25W, thus maximum ranges of 54[49] nautical miles (= 100km) in normal conditions are common. Other official bodies state maximum ranges of 25[50] nautical miles (= 46km) to 40[51] nautical miles (= 74km), depending on the expected accuracy and if retransmission by other stations is available.

VHF radio waves propagate mostly through line-of-sight or direct waves (61)(64). This means that they travel in straight paths, not following the Earth’s curvature. Therefore, their maximum range is limited by the actual location of transmitting and receiving antennas, which must be in sight, to guarantee communication. Ground-bounce and skywave – ionospheric – propagation might be possible for the lower frequencies, closer to the HF upper range, as well. In some occasions, VHF waves can also travel longer distances through tropospheric ducting related to temperature gradients in the atmosphere (64).

---

49 As per the International Telecommunication Union.
50 As per the United States Coast Guard. Based on ship-to-ship maximum accuracy in high seas.
51 As per the European Space Union. Based on maximum accuracy through repeaters.

Maximum line-of-sight distance can be easily computed through equation (eq.1) below, when transmitting and receiving antenna heights are known:

$$LOS = 2.22 \cdot \left(\sqrt{h_{Tx}} + \sqrt{h_{Rx}}\right) \quad \text{(eq.1)}$$

Where:

h_{Tx} : Height (m) of the transmitting antenna above sea level; and

h_{Rx} : Height (m) of the receiving antenna above sea level.

Considering that the height of the antenna at the Barcelona School of Nautical Studies is around 17m above sea level; and that average height of the antenna on a container ship of around 85 – 90m, about 30 nautical miles (= 55.6km) is a good range to consider for the sake of the study and guarantee stable reception. However, weather permitting, maximum range rises up to 120 nautical miles (= 222.2km).

Initial position filtering

Once knowing the maximum accuracy range based on the line-of-sight propagation, the full dataset was filtered out to consider only the stations within a reliable range. The filtering process was based on the simple equation for the circle, as shown below (eq.2).

Coordinates only located within a range equal or less to the circle radius were acquired, the rest being dropped.

$$(x_i - x_0)^2 + (y_i - y_0)^2 \le r^2 \quad \text{(eq.2)}$$

Where:

x_i : Lateral planar coordinate of any given point;

x_0 : Lateral planar coordinate of the reference point;

y_i : Vertical planar coordinate of any given point;

y_0 : Vertical planar coordinate of the reference point; and

r : Planar circle radius.

In terms of coordinates, as distances were well below 200 nautical miles (= 370.4km), accuracy was deemed to be of limited importance. Therefore, the equirectangular projection approximation through the Pythagoras' theorem (65) was used, when applying coordinates to equation (eq.2).

The forward projection method, which converts spherical coordinates into planar coordinates, presuming the Earth as a perfect sphere was used (eq.3).

$$x = R \cdot (\lambda_i - \lambda_0) \cdot \cos \varphi_m$$
$$y = R \cdot (\varphi_i - \varphi_0) \tag{eq.3}$$

Where:

R	: Radius of earth;
λ_i	: Longitude of any given point;
λ_0	: Latitude of the reference point;
φ_m	: Half of the sum of the latitudes of the reference and any given point;
φ_i	: Latitude of any given point; and
φ_0	: Latitude of the reference point.

Applying the previous to equation (eq.2), and considering the maximum accuracy range, as discussed before, the position filter was as follows. Equation (eq.4) was applied as a Boolean algorithm to the dataset, discarding all FALSE results.

$$R^2 \cdot \left[\left[(\lambda_i - \lambda_{FNB}) \cdot \cos\left(\frac{\varphi_i + \varphi_{FNB}}{2}\right) \right]^2 + (\varphi_i - \varphi_{FNB})^2 \right] \leq r^2 \tag{eq.4}$$

Where:

R	: Radius of earth, taken as R = 3440.6 nautical miles (= 6371.0km);
λ_i	: Longitude of any given station, as given in the AIS dynamic message;
λ_{FNB}	: Longitude of the receiving station, taken at λ_{FNB} = 002º11.1'E;
φ_i	: Latitude of any given station, as given in the AIS dynamic message;
φ_{FNB}	: Longitude of the receiving station, taken at λ_{FNB} = 41º22.9'N; and
r	: Selected range, taken as r = 30 nautical miles (= 55.6km).

Figure 17 shows a circle covering the range of study of 30 nautical miles (= 55.6km) centered at the Barcelona School of Nautical Studies.

Figure 17. Initial 30nm (= 55.6km) radius circle position filter

Call filtering

Several methods could be used in order to filter out the number of calls per ship. Out of simplicity, a secondary filter based on equation (4) applied as a Boolean algorithm was used to establish whether the station was located inside, TRUE, or outside, FALSE, of the harbor limits.

A circle centered in position[52] λ = 002º05.5'E and φ = 41º21.2'N and with a radius r = 4.4 nautical miles (= 8.1km) was considered, as it represented a circle tangent to both harbor entrances. Afterwards, each MMSI was grouped in sequences of inside (TRUE) and outside (FALSE), with a port call being considered as the change between conditions outside to inside, performed at a time when the condition changed.

So as to avoid false calls introduced by vessels already inside the harbor limits when the time started counting on March 1, 2020 at 00:00 UTC, all the vessels in port at that time were dropped from the dataset.

[52] More precise position is: λ = 2.092140º and φ = 41.353021º.

Figure 18 shows the boundaries of the call filter algorithm. As seen, part of the anchorage is located within the boundaries. Therefore, vessels with status *At Anchor* were also dropped from the selection so as to make sure that false calls were not being considered.

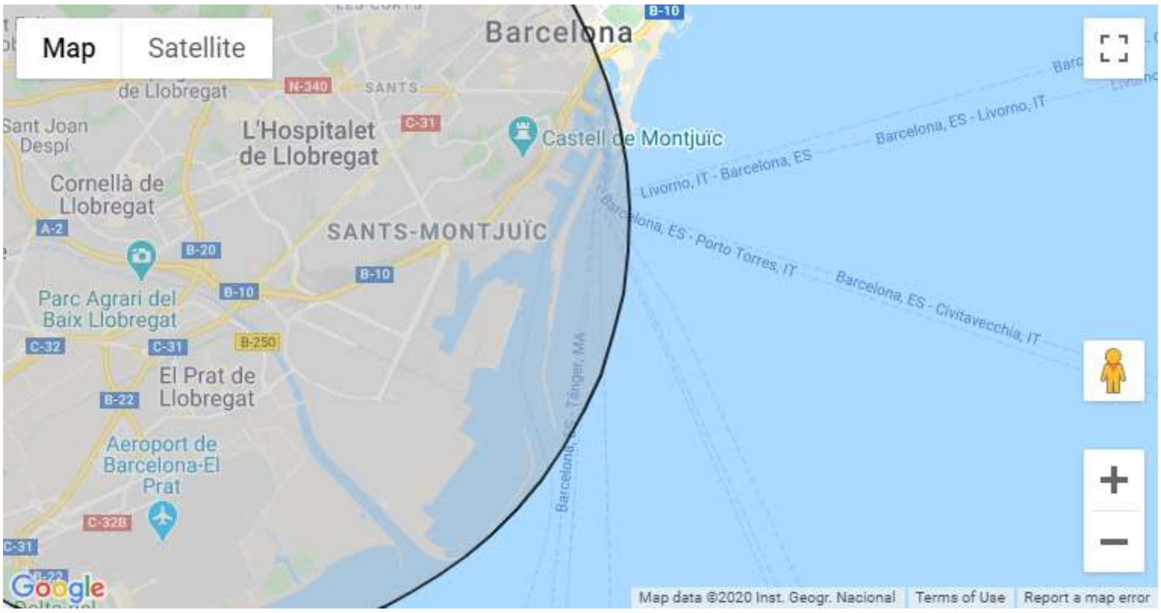

Figure 18. Secondary 4.4nm (= 8.1km) radius circle call filter

Calibration

The call filter was calibrated based on data provided by the Port of Barcelona (81) and the IHS Seaweb database[53], so as to assess the error related to this AIS-based method. Table 18 shows the reported values from the three different methods, and the relative error based on the AIS filter.

Table 18. Reported calls at Barcelona from March to June 2020
Values in brackets are the relative errors calculated over the AIS call filtre values

Month	AIS call filter	Port of Barcelona	IHS Seaweb
March	614	594 (ϵ 3.4%)	636 (ϵ 3.5%)
April	450	452 (ϵ 0.4%)	536 (ϵ 16.1%)
May	455	459 (ϵ 0.9%)	520 (ϵ 12.5%)
June	526	491 (ϵ 7.1%)	561 (ϵ 6.2%)

[53] As per IHS Seaweb Movements: Arrivals and Departures. Retrieved on August 3, 2020.

The IHS database reported an average of +10.2% more calls than the AIS-based filter; whereas, the Port of Barcelona reported an average of -2.4% less. This led to average relative errors of 9.6% and 3.0%, when comparing the AIS filter to the IHS dataset and the Port of Barcelona statistics service, respectively.

Table 19 shows that for cargo vessels, the IHS database reported an average of -1.0% less vessels than the AIS-based filter; whereas the Port of Barcelona counted an average of -6.1% less vessels. This lead to average relative errors of 2.4% and 6.5%, respectively when comparing the AIS filter to the IHS database and the Port of Barcelona official statistics.

Table 19. Reported cargo vessel calls at Barcelona from March to June 2020
Values in brackets are the relative errors calculated over the AIS call filtre values

Month	AIS call filter	Port of Barcelona	IHS Seaweb
March	315	301 (ϵ 4.7%)	312 (ϵ 1.0%)
April	276	256 (ϵ 7.8%)	287 (ϵ 3.8%)
May	287	270 (ϵ 6.3%)	296 (ϵ 3.0%)
June	294	274 (ϵ 7.3%)	289 (ϵ 1.7%)

Regarding tankers, as stated in Table 20, the IHS database reported an average of +57.0% more vessels than the AIS-based filter; whereas the Port of Barcelona reported an average of -0.3% less vessels. This difference is related to the fact that the movement of bunker barges within the port is counted as separated calls by IHS. When considering AIS calls, relative errors of 36.3% and 2.9% have to be taken into account compared to the IHS and Port of Barcelona statistics, although the upper limit might not be realistic owing to the previously stated reasons.

Table 20. Reported tanker vessel calls at Barcelona from March to June 2020
Values in brackets are the relative errors calculated over the AIS call filtre values

Month	AIS call filter	Port of Barcelona	IHS Seaweb
March	90	93 (ϵ 3.2%)	135 (ϵ 33.3%)
April	81	81 (ϵ 0.0%)	132 (ϵ 38.6%)
May	72	74 (ϵ 2.7%)	108 (ϵ 33.3%)
June	73	69 (ϵ 5.8%)	121 (ϵ 39.7%)

In terms of passenger vessels, as seen in Table 21, the IHS database counted up an average of +5.0% more vessels than the AIS-based filter; whereas the Port of Barcelona reported an average of +4.3% more vessel. The differences arise from the fact that both the harbor authority and IHS count berth-to-berth movements within the port and that the AIS dataset contains also messages from local excursion boats[54], which are not considered in the official statistics. When considering AIS calls, relative errors of 11.5% and 11.6% have to be taken into account, respectively.

Table 21. Reported passenger vessel calls at Barcelona from March to June 2020
Values in brackets are the relative errors calculated over the AIS call filtre values

Month	AIS call filter	Port of Barcelona	IHS Seaweb
March	209	200 (ϵ 3.2%)	203 (ϵ 3.0%)
April	93	115 (ϵ 19.1%)	117 (ϵ 20.5%)
May	96	115 (ϵ 16.5%)	116 (ϵ 17.2%)
June	159	148 (ϵ 7.4%)	151 (ϵ 5.3%)

The IHS database has to be assessed with care, as it considers berth-to-berth and bunker barge movements as independent calls. That is the reason why a lower relative error is obtained when comparing the AIS data to the official statistics reported by the harbor office. The discrepancy in terms of cargo vessels is mostly related to the fact that a sizeable minority of vessels anchored very close to the breakwater, thus the filter considered them as a call when they moved in and out.

Therefore, in order to increase the accuracy of the algorithm, the following was applied:

1. Drop out all calls which status was *At Anchor*;
2. Reduce to a single call, all calls that were repeated within the same time; and
3. Drop out all calls corresponding to the local excursion boats[55].

All in all, the AIS filter is able to recognize the number of calls with an overall accuracy of 97.1%, which was deemed to be sufficient for the scope of the project.

3.1.3. Ship-specific filters

Merchant fleet filtering

Out of the whole dataset, only merchant vessels were considered in this project. This filtering was performed based on two main arguments. On the one side, so as to guarantee that access to proper database containing ship-specific technical information was readily available and with the required

[54] Known locally as *Las Golondrinas*, Spanish for barn swallow bird.
[55] Corresponding to the following MMSIs: 224022660 (MS *Trimar*), 224022650 (MS *Omnibus*) and 224334000 (*MS Jolly Roger*).

accuracy levels. On the other side, given the limited computational capacity so as to avoid excessive inaccurate or imprecise data related to missing AIS reports common in selected types of ship.

Class A messages and IDs 1, 2, 3 and 5 were only considered. Afterwards, they were further filtered out, dropping all datasets which type of ship in message 5 was other than any value within the range 60 to 89, both included. The range was further cut into 3 different sets, assigning the category *passenger vessel* to the first set, *cargo vessel* to the second set and *tanker vessel* to the third set, following Table 11.

As shown in Table 22, all of these three categories are further subdivided into specific types of ship.

Table 22. AIS merchant fleet category subdivision labelling - SOURCE: IHS Fairplay

Passenger **(60 – 69)**	**Cargo** **(70 – 79)**	**Tanker** **(80 – 89)**
Passengers Ship	Livestock Carrier	Asphalt/Bitumen Tanker
Inland Passengers Ship	Bulk Carrier	Chemical Tanker
Inland Ferry	Ore Carrier	Crude Oil Tanker
Floating Hotel	General Cargo	Inland Tanker
Ferry	Wood Chips Carrier	Fruit Juice Tanker
Ro-Ro/Passenger Ship	Container Ship	Bunkering Tanker
Accommodation Ship	Ro/Ro Cargo	Wine Tanker
Accommodation Barge	Reefer	Oil Products Tanker
Accommodation Jack Up	Heavy Load Carrier	Oil/Chemical Tanker
Accommodation Vessel	Barge	Water Tanker
Passengers Landing Craft	Ro-Ro/Container Carrier	Tank Barge
Houseboat	Inland Cargo	Edible Oil Tanker
Accommodation Platform	Cement Carrier	Lpg/Chemical Tanker
Air Cushion Passenger Ship	Vegetable/Animal Oil Tanker	Shuttle Tanker
	OBO Carrier	CO_2 Tanker
	Vehicles Carrier	
	Inland Ro-Ro Cargo Ship	
	Rail/Vehicles Carrier	
	Pallet Carrier	
	Cargo Barge	
	Hopper Barge	
	Deck Cargo Ship	
	Aggregates Carrier	
	Limestone Carrier	
	Self Discharging Bulk Carrier	
	Deck Cargo Pontoon	
	Bulk Carrier With Vehicle Deck	
	Pipe Carrier	
	Cement Barge	

	Stone Carrier Bulk Storage Barge Aggregates Barge Timber Carrier Bulker Trans Shipment Barge Powder Carrier Cabu Carrier Vehicle Carrier	

Status filtering

Messages 1, 2 and 3 contain information about the vessel navigational status, as defined in Table 10. Prior to re-labelling the dataset status based on the provided numerical information, all vessels with status other than 0 (= *Underway using Engine*), 1 (= *At Anchor*), 2 (= *Not Under Command*) and 5 (= *Moored*) were dropped out.

The process of status filtering was deemed heavily important, in order to plot vessels in a live map and calculate emissions. This was due to the fact that depending on the vessel status, messages were only broadcasted at longer intervals, which required the repetition of datasets in order to obtain stable data.

3.2. Emission inventory

Emissions related to the maritime traffic during the 5-months period were assessed through the STEAM v.2 algorithm. They were computed independently for each vessel based on available technical data from the IHS database. A secondary method to generate the emission inventory, based on modelled installed power is also provided in section 3.2.2.

Although several models (66)(67)(68) have been proved to generate accurate emission inventories, this method was selected over the others owing to its accuracy, simplicity and compatibility with AIS data (66)(69)(70)(71).

The Ship Traffic Emission Assessment Model, in short STEAM; is an algorithm developed by Jalkanen et al., in 2009 to study the emissions within the Baltic Sea (69). The preliminary model included a methodology to be used to assess the fuel consumption and emissions of CO_2, SO_2 and NO_X using extensive AIS data and a comprehensive database, with all the required technical information on the considered vessels. The model was further enhanced in 2012, in the so-called STEAM v.2, to include PM and CO emissions and to calibrate fuel consumptions and instantaneous power (70). The latest available version, the STEAM v.3 corrects previous uncertainties related to missing AIS data and can be applied over a global scale (71).

The following codes were developed to assess the emissions. They can be found in Annex A2 for further reference:

i. AISemissions_db.py – generates an emission inventory through the STEAM v.2 algorithm, based on a comprehensive technical database –;
ii. AISemissions_math.py – generates an emission inventory through a modified version of the STEAM v.2 algorithm, based on a mathematical model to estimate installed power –;
iii. AISemissions_map.py – plots in a semi-live map all vessels in range, their fuel consumption and emissions per minute –; and
iv. AISemissions_heatmap.py – plots in a semi-live heatmap the concentration of higher pollutants above the average –.

3.2.1. The STEAM algorithm

Emission models related to shipping within a spatial region typically consider separate stages or phases per vessel (36)(70), i.e. cruising, maneuvering and hoteling. This allows for a better approach, as power demands are different depending on the phase (68).

For the sake of the project, as seen in (eq.5), this classification was further expanded with a fourth phase, at anchor, in order to better discretize the impact of anchored vessels.

$$E_T = E_{cruising} + E_{at\ anchor} + E_{maneuvering} + E_{hoteling} \qquad \text{(eq.5)}$$

Where:

E_T	: Total emissions related to a ship;
$E_{cruising}$	: Emissions related to the cruising stage;
$E_{at\ anchor}$	: Emissions related to the period at anchor;
$E_{maneuvering}$	: Emissions related to the maneuvering stage; and
$E_{hoteling}$	: Emissions related to the hoteling stage.

The stages were filtered out based on speed, position and rate-of-turn data. For instance, vessels in the cruising phase were all those with speeds above 1.5 knots and out of port premises. Vessels at anchor and adrift were considered a single group, made up of all those vessels with speeds below 1.5 knots outside of harbor limits. Maneuvering vessels were those with speeds higher than 0.5

knots and inside of port premises, meanwhile hoteling vessels where those with speeds below 0.5 knots and inside the harbor.

Traditional models typically calculate emissions based on the installed engine power, engine load in the given phase and pollutant emission factors (36). As AIS data contained information on vessel speed and position, an enhanced methodology was considered, in which the engine power and engine load were merged into the instantaneous power (66)(69), as seen in (eq.6).

$$\begin{aligned} E_{T,i} &= \sum_p \left[\Delta t_{1,2} \cdot \sum_e \left(P_e \cdot EL_e \cdot EF_{e,i,m,p,v} \right) \right] \cdot 10^{-6} \\ &= \sum_p \left[\Delta t_{1,2} \cdot \sum_e \left(P_{1,2} \cdot EF_{e,i,m,p,v} \right) \right] \cdot 10^{-6} \end{aligned} \quad \text{(eq.6)}$$

Where:

E_T	: Total emissions related to a ship (tons);
$\Delta t_{1,2}$	: Time difference between consecutive waypoints (hours);
P	: Engine power (kW);
$P_{1,2}$	: Instantaneous engine power between consecutive waypoints (kW);
EL	: Engine load factor, related to the MCR (%);
EF	: Emission factor (g/kWh);
e	: Type of engine, either main engine or auxiliary engine;
i	: Polluting substance;
m	: Type of fuel, either light sulfur heavy fuel oil (LSHFO), marine gasoil (MGO) or liquefied natural gas (LNG);
p	: Vessel phase, either cruising, at anchor, maneuvering or hoteling; and
v	: Engine revolutions (rpm), only for NO_X emissions.

Power estimation

Generic vessel power can be calculated based on the Propeller Law (66), as shown in (eq.7), which states that it is k times the cube of the vessel speed.

$$P = k \cdot v^3 \quad \text{(eq.7)}$$

Where:

P : Generic vessel power (kW);

k : Power to speed constant (kW·s/m); and

v : Vessel speed (m/s).

Vessels are designed at a service speed. However, sea and engine margins are always observed when dimensioning the vessel power plant (69). Therefore, the actual installed power is slightly higher than the real required power.

Let the maximum engine load be the ratio between the service and the maximum installed power, as in (eq.8).

$$EL_{max} = \frac{P_{service}}{P_{installed}} \quad \text{(eq.8)}$$

Where:

EL_{max} : Maximum engine load factor, related to the MCR (%);

$P_{service}$: Service power related to the service speed (kW); and

$P_{nstalled}$: Maximum installed power (kW).

Combining (eq.7) and (eq.8), constant k can be easily computed through (eq.9), when the installed power and service speed are both known. For the emission inventory, values were obtained straight from the IHS vessel database. Regarding the maximum engine load, literature research showed that several values are recommended (66)(67)(69), nonetheless, the most common EL_{max} = 80% was selected over the others (66).

$$P_{service} = k \cdot {v_{service}}^3 = EL_{max} \cdot P_{nstalled} \rightarrow k = \frac{EL_{max} \cdot P_{installed}}{{v_{service}}^3} \quad \text{(eq.9)}$$

Where:

$P_{service}$: Service power related to the service speed (kW); and

$v_{service}$: Vessel service speed (m/s).

All in all, combining (eq.7) and (eq.9), the instantaneous consumed power was computed based on the vessel actual speed, provided in AIS and its unique constant k, as seen in (eq.10). In some studies (66), calculated distance-over-time speeds are preferred to the one given in AIS messages. However, given the reduced size of the location and based on literature research (68)(72), AIS speeds were considered this time.

$$P_{1,2} = k \cdot {v_{1,2}}^3 \qquad \text{(eq.10)}$$

Where:

$P_{1,2}$: Instantaneous engine power between consecutive waypoints (kW);

k : Power to speed constant (kW·s/m), unique to every vessel; and

$v_{1,2}$: Vessel current speed (m/s), as provided in AIS messages.

This method was selected over general engine load factors for the cruising and maneuvering stages, as vessel operability is unique to its owners and crew (37). Therefore, using real-time data increased the reliability of the obtained results. For the hoteling and anchored phases, auxiliary engine power was preferred over main engine output. However, when auxiliary engines were not deployed, a 20% and 10% load factors were respectively considered for main engines (27)(69).

Emission factors

In this study, emission factors were computed independently for each air pollutant. Three different methodologies were used, depending on the air pollutant nature, namely fuel-related (CO_2 and SO_2), engine-related (NO_X) and special considerations for PM.

For fuel-related emissions, based on the engine instantaneous specific fuel consumption and the chemistry of the type of fuel burnt in each phase, the emission factor was easily calculated (69). Information on the type of fuel was readily available in the IHS database for all the considered vessels, either low sulfur heavy fuel oil (LSHFO), marine gasoil (MGO) or liquefied natural gas (LNG). While at anchor, maneuvering and hoteling, MGO and LNG were preferred over LSHFO (27)(37).

In general terms, equation (eq.11) shows that the emission factor is the product of the molar mass of that substance and the number of moles per energy resulting from the reaction.

$$EF = m \cdot n \qquad \text{(eq.11)}$$

Where:

EF : Emission factor (g/kWh);

m : Molar mass (g/mol); and

n : Moles per energy unit (mol/kWh).

For CO_2, (eq.12) shows that the ratio between carbon radicals and combined CO_2 is 1:1, therefore 1 mole of CO_2 is obtained per every mole of C in the reaction.

$$C + O_2 \rightarrow CO_2 \qquad \text{(eq.12)}$$

Hence, the emission factor was computed through (eq.13) based on the instantaneous specific fuel consumption, the molar masses of CO_2 and C, and the carbon content in the fuel, as given in Table 23.

$$EF_{CO_2} = \frac{SFC_{1,2} \cdot \%C}{m_C} \cdot m_{CO_2} \qquad \text{(eq.13)}$$

Where:

EF_{CO_2} : Emission factor for CO_2 (g/kWh);

$SFC_{1,2}$: Instantaneous specific fuel consumption (g/kWh);

$\%C$: Carbon content in fuel (%), taken as given in Table 23;

m_C : Molar mass of carbon (g/mol), taken at m_C = 12.01 g/mol; and

m_{CO_2} : Molar mass of CO_2 (g/mol), taken at m_{CO_2} = 44.0886 g/mol.

Table 23. Carbon content per fuel type – SOURCE: ISO 8217

Type of fuel	Carbon content % (m/m)
LSHFO	86.0%
MGO	87.5%
LNG	75.0%

For SO_2, (eq.14) shows that the ratio between sulfur radicals and combined SO_2 is 1:1, therefore 1 mole of SO_2 is obtained per every mole of sulfur in the reaction.

$$S + O_2 \rightarrow SO_2 \qquad \text{(eq.14)}$$

Eventually, the emission factor was computed through (eq.15), considering the sulfur contents provided in Table 24.

$$EF_{SO_2} = \frac{SFC_{1,2} \cdot \%S}{m_S} \cdot m_{SO_2} \qquad \text{(eq.15)}$$

Where:

EF_{CO_2} : Emission factor for CO_2 (g/kWh);

$SFC_{1,2}$: Instantaneous specific fuel consumption (g/kWh);

$\%S$: Sulfur content in fuel (%), taken as given in Table 24;

m_S : Molar mass of sulfur (g/mol), taken at m_S = 32.0655 g/mol; and

m_{SO_2} : Molar mass of SO_2 (g/mol), taken at m_{SO_2} = 64.06436 g/mol.

Table 24. Sulfur content per fuel type – SOURCE: ISO 8217

Type of fuel	Sulfur content % (m/m)
LSHFO	0.5%
MGO	0.5%
LNG	0.004%

For engine-related emissions, NO_X in this case, the emission factors were the ones recommended as maximum by the IMO (69), given in section 2.2.3 in Table 16. As these values were year- and engine revolution-dependent, the required information was gathered from the IHS database for all the vessels. However, as in previous cases, it was not always readily available, specifically engine revolutions. Through literature research, 500rpm (medium-speed engine) and engines built before 2011, thus Tier I; were found as the most accurate values (68)(70), when data was not available.

For PM, the method developed by Jalkanen et al. in 2012 (70) was deemed accurate and easy to implement, as in (eq.16). PM consists mostly of traces of elementary carbon (EC), organic carbon (OC), ashes, sulfate (SO_4) and water vapor (H_2O).

$$EF_{PM} = SFC_{rel} \cdot \left(EF_{SO_4} + EF_{H_2O} + EF_{EC} + EF_{ash} + EF_{OC} \cdot OC_{EL}\right) \qquad \text{(eq.16)}$$

Where:

EF_{PM} : Emission factor for PM (g/kWh);

SFC_{rel} : Relative specific fuel consumption (g/kWh), as calculated through (13);

EF_{SO_4} : Emission factor for SO_4 (g/kWh), taken at $EF_{SO_4} = 0.312 \cdot \%S$ g/kWh;

EF_{H_2O} : Emission factor for H_2O (g/kWh), taken at $EF_{H_2O} = 0.244 \cdot \%S$ g/kWh;

EF_{EC} : Emission factor for EC (g/kWh), taken at EF_{EC} = 0.08g/kWh;

EF_{ash} : Emission factor for ashes (g/kWh), taken at EF_{ash} = 0.06g/kWh;

EF_{OC} : Emission factor for OC (g/kWh), taken at EF_{OC} = 0.20g/kWh;

OC_{EL} : Component of OC (dimensionless), calculated through (eq.17).

$$OC_{EL} = \begin{cases} 3.333 & for\ EL_{1,2} < 15\% \\ \dfrac{1024}{1 + 47000 \cdot e^{-32547 \cdot EL_{1,2}}} & for\ EL_{1,2} \geq 15\% \end{cases} \qquad \text{(eq.17)}$$

Where:

OC_{EL} : Component of OC based on engine load (dimensionless); and

$EL_{1,2}$: Instantaneous engine load factor (%), taken as $EL_{1,2} = EL_{max} \cdot \left(\frac{v_{1,2}}{v_{service}}\right)^3$.

Fuel consumption

Fuel consumption is a good indicator of sustainability and traffic within the shipping industry (66). Moreover, several methods (36) rely on this value to generate emission inventories and assess the impact of the shipping industry within their boundaries.

Actual fuel consumption was easily computed through (eq.18) based on the instantaneous power, calculated as in (eq.10) and the specific fuel consumption, which changed based on the type of engine, fuel and phase.

$$FC_T = \sum_p \left[\Delta t_{1,2} \cdot \sum_e \left(P_{1,2} \cdot SFC_{1,2\ e,m,p} \right) \right] \cdot 10^{-6} \qquad \text{(eq.18)}$$

Where:

FC_T : Total fuel consumption (tons);

$\Delta t_{1,2}$: Time difference between consecutive waypoints (hours);

$P_{1,2}$: Instantaneous engine power between consecutive waypoints (kW);

$SFC_{1,2}$: Instantaneous specific fuel consumption (g/kWh);

e : Type of engine, either main engine or auxiliary engine;

m : Type of fuel, either light sulfur heavy fuel oil (LSHFO), marine gasoil (MGO) or liquefied natural gas (LNG); and

p : Vessel phase, either cruising, at anchor, maneuvering or hoteling.

In fact, the methodology to estimate specific fuel consumption was based on the parabolic curves developed by Jalkanen et al., in 2012 for the STEAM v.2 (70), in which the instantaneous specific fuel consumption is computed through a base and a relative value, as in (eq.19).

$$SFC_{1,2} = SFC_{base} \cdot SFC_{rel} \qquad \text{(eq.19)}$$

Where:

$SFC_{1,2}$: Instantaneous specific fuel consumption (g/kWh);

SFC_{base} : Base specific fuel consumption (g/kWh), mostly in the IHS database; and

SFC_{rel} : Relative specific fuel consumption (g/kWh), based on the engine load.

Base specific fuel consumption depended much on the type of engine, either main or auxiliary engine; and the type of fuel (70). IHS included in its database information on the base specific fuel consumption of selected vessels. However, as not all the considered vessels had this information readily available, the model was enhanced with the values provided in Table 25.

Table 25. Base specific fuel consumptions for different engines – SOURCE: Jalkanen et al., 2012

Type of engine	SFC_{base} (g/kWh)
Main Engine	200
Auxiliary Engine	220

The relative component changed based on the instantaneous engine load factor (70). Equation (eq.20) was developed by Jalkanen et al. in 2012 from the assessment of specific fuel consumption curves from several engine manufacturers.

$$SFC_{rel} = 0.455 \cdot {EL_{1,2}}^2 - 0.71 \cdot EL_{1,2} + 1.28 \qquad \text{(eq.20)}$$

Where:

SFC_{rel} : Relative specific fuel consumption (g/kWh); and

$EL_{1,2}$: Instantaneous engine load factor (%), taken as $EL_{1,2} = EL_{max} \cdot \left(\frac{v_{1,2}}{v_{service}}\right)^3$

Combining equations (eq.19) and (eq.20), the resultant instantaneous specific fuel consumption was as in (eq.21).

$$SFC_{1,2} = SFC_{base} \cdot \left(0.455 \cdot {EL_{1,2}}^2 - 0.71 \cdot EL_{1,2} + 1.28\right) \qquad \text{(eq.21)}$$

Where:

$SFC_{1,2}$: Instantaneous specific fuel consumption (g/kWh);

SFC_{base} : Base specific fuel consumption (g/kWh), mostly in the IHS database; and

$EL_{1,2}$: Instantaneous engine load factor (%), taken as $EL_{1,2} = EL_{max} \cdot \left(\frac{v_{1,2}}{v_{service}}\right)^3$

Special consideration for auxiliary engines

Whenever available, auxiliary engines were also considered as part of shipboard power. For the sake of simplicity, all of them were considered to be medium-speed engines (70), running at 500rpm and burning MGO (36), with a specific fuel consumption of 220g/kWh (70).

Engine configurations are extremely flexible depending on the installed power and the needs of the vessel crew (37). Therefore, as stated in Table 26, auxiliary engines were the main source of power while at anchor and berthed. However, if vessels did not have auxiliary installed power as per the IHS database, main engine power with reduced fixed load factors was considered, as previously discussed.

Table 26. Engine configuration per stage – SOURCE: J. Jalkanen et al., 2012

Stage	Engine configuration		
	Cargo	**Tankers**	**Passenger**
Cruising	Main + Auxiliary	Main + Auxiliary	Main + Auxiliary
At Anchor	Auxiliary	Auxiliary	Auxiliary
Maneuvering	Main + Auxiliary	Main + Auxiliary	Main + Auxiliary
Hoteling	Auxiliary	Auxiliary	Auxiliary

Regarding load factors, different values could be considered (68)(70)(72). However, based on literature research, values given in Table 27 seemed to be the most common in studies (68) with similar regional scope.

Table 27. Auxiliary engine load factor – SOURCE: X. Sun et al., 2018

Stage	Auxiliary engine load factor		
	Cargo	**Tankers**	**Passenger**
Cruising	60%	60%	80%
At Anchor	40%	70%	70%
Maneuvering	60%	70%	80%
Hoteling	40%	70%	70%

Finally, for NO_X emissions, auxiliary engines were considered to follow Tier I as per section 2.2.3, Table 16.

3.2.2. Modeling installed power

As comprehensive databases containing all required technical information about vessels are not always available, it was deemed necessary to generate a mathematical model, able to estimate the emissions from vessels around Barcelona with limited data access.

The main goal was to develop a system which based solely on available AIS data and average service speeds, auxiliary engine power, engine revolutions and specific fuel consumptions, would be able to predict the instantaneous power and calculate the emissions with an acceptable accuracy. Given the limitations of the model, global accuracy was preferred over detailed results.

Mathematical model

The database upon which the mathematical model was created consisted on 896 vessels, of which 812 vessels represented the 70% of the total number of unique merchant vessels which were in AIS-range from March to July, 2020 and 84 were passenger cruise vessels introduced on purpose in the model. Cropping above 70% of the database did not significantly change the results, therefore it was deemed enough to obtain a mathematical relation.

A total of 528 cargo vessels, 256 tanker vessels and 112 passenger vessels were considered. As per the passenger vessels, the extra 84 cruise ships were introduced so as to create a more generic equation which could be used in the future to estimate emissions. For each type of ship a 25% was devoted as a test set to analyze over-fitting, in case similar particulars were reported by sister-ships.

Table 28 shows the model upper and lower limits, as taken from the database.

Table 28. Lower and upper límits per type of ship for which the mathematical model is valid

Item	Cargo		Tankers		Passenger	
	Lower	Upper	Lower	Upper	Lower	Upper
Length (m)	66.0	440.0	72.1	400.0	87.0	362.0
Breadth (m)	10.0	61.5	12.2	61.5	13.0	72.0
Installed power (kW)	598	75600	749	97020	590	92400
Service speed (knots)	7.0	26.3	5.0	26.1	11.0	25.0
Engine speed (rpm)	69	3494	68	1500	104	765

Cepowski suggests that the installed power on cargo vessels and tankers can be estimated through different vessel dimensions, such as deadweight and service speed (73). For passenger vessels, the

EMEP/EEA algorithm suggests a gross tonnage-based model (36). However, all of these values are not provided by default in AIS messages, as seen in section 2.1.2, Table 10 and Table 11.

As vessel power is also related to their main dimensions (73), a non-linear iterative numerical regression was applied over the installed power and different ship dimensions provided in AIS messages. As proposed by Cepowski in several studies (73)(79), the aim of the method was to find through non-linear numerical iterations the highest coefficient of determination (R^2) and the lowest root-mean squared error (RMSE) for 1D (installed power v. length) or 2D (installed power v. length and breadth). Although 3D combinations (installed power v. length, breadth and draft) were initially considered, they were discarded given that AIS-provided draft is a voyage-related changing data.

Results for the model and fittings are given in section 5.1.3.

Applying the mathematical model to compute emissions

Once knowing the vessel installed power, the calculation process was in as much as described in section 3.2.1. However, several considerations were to be taken into account as some information was also missing.

For instance, the considered service speed was the computed average from the same 896-vessel database, as shown in Table 29.

Table 29. Average service speed per type of ship

Type of ship	Average service speed (knots)
Cargo	19.0 (= 9.77m/s)
Tankers	14.5 (= 7.46m/s)
Passenger	22.5 (= 11.56m/s)

Regarding engine revolutions, the computed average from the 896-vessel database was also considered, as seen in Table 30. As 75% of the vessels were built before 2011, Tier I was preferred for NO_X emissions, based on section 2.2.3 Table 16.

Table 30. Average engine revolution per type of ship

Type of ship	Aerage engine revolutions (rpm)
Cargo	350
Tankers	400
Passenger	325

As auxiliary engine power was a common unknown, values provided by Jalkanen et al., 2009 and revalidated in 2012 were considered, as shown in Table 31. Similar load factors and engine configuration were taken into account, as in section 3.2.1 in Table 26 and Table 27; taking into account, that full auxiliary engine configurations applied for all vessels while at anchor and moored. As in the previous case, auxiliary engines were considered to be medium-speed engines, running at 500rpm and burning MGO.

Table 31. Average auxiliary engine installed power per ship – SOURCE: Jalkanen et al., 2012

Type of ship	Auxiliary engine installed power (kW)
Cargo	1000
Tankers	1000
Passenger	1250

Main engines were considered to burn only LSHFO during the cruising stage, and MGO elsewhere. Regarding, specific fuel consumptions, a higher value, as given in Table 32, was selected for main engines in this case, as the average consumption of the 896-vessel database was slightly above 200g/kWh.

Table 32. Base specific fuel consumptions for different engines

Type of engine	SFC_{base} (g/kWh)
Main Engine	220
Auxiliary Engine	220

Fuel consumption was selected as the major indicator to assess the model error, as most inventories calculate emissions straight from consumption (36)(67).

Results for the fuel consumption and emissions are presented in section 5.1.3.

Chapter 4. Maritime and port traffic analysis

This chapter presents an analysis of the evolution of maritime traffic, number of port calls and overall characteristics of vessels trading within the area of Barcelona during a period of 5 months, spanning from March to July 2020.

The study is based on the data related to merchant vessels acquired through an AIS receiver located at the Barcelona School of Nautical Studies (UPC-BarcelonaTECH) and a range of 30 nautical miles.

Within the following lines, the database is further transformed so as to represent different dimensions and parameters, giving some light to the impact that the pandemic might have had in the shipping industry in the region. In order to do so, answers to the following questions were provided through data assessment.

1. Has there been a reduction in the number of vessels transiting in the area?
2. Has the number of vessels *At Anchor, Not Under Command* or *Moored* increased?
3. Has the average speed of vessels reduced?
4. Has the number of port calls changed in the Port of Barcelona?

4.1. Results

In the period running from March 1 through and including July 31, 2020; 1160 different vessels were reported within a range of 30 nautical miles. A total of 11,860,409 AIS messages were processed in the given time.

4.1.1. Vessels in range

An average of 32 vessels could be found every hour within the studied range. Maximum number of vessels in range was 53 (+68.5% above average), corresponding to July 22, 2020 at midnight UTC and July 30, 2020 at 19:00 UTC. Minimum number of vessels was 15 (-52.3% below average), corresponding to July 8, 2020 for a period of time between 13:00 UTC and 17:00 UTC.

Table 33. Monthly minimum, average and maximum number of vessels in range
Values in brackets are the difference over the 5-monthly average

Month	Minimum	Average	Maximum
March	18 (-42.8%)	32 (-0.4%)	45 (+43.0%)
April	20 (-36.4%)	34 (+8.1%)	52 (+65.3%)
May	18 (-42.8%)	31 (-1.1%)	48 (+52.6%)
June	16 (-49.1%)	30 (-5.0%)	48 (+52.6%)
July	15 (-52.3%)	31 (-2.4%)	53 (+68.5%)

As seen in Table 33, April was the month with the largest number of vessels transiting the region, with an average of 34 vessels per hour (+8.1% above the average), whereas June and July were the months with the least number of vessels, with averages of 30 (-5.0% below the average) and 31 (-2.4% below average), respectively. The largest difference between the minimum and maximum number of vessels was reported for the month of July, whereas March was the month with the minimum difference between both values.

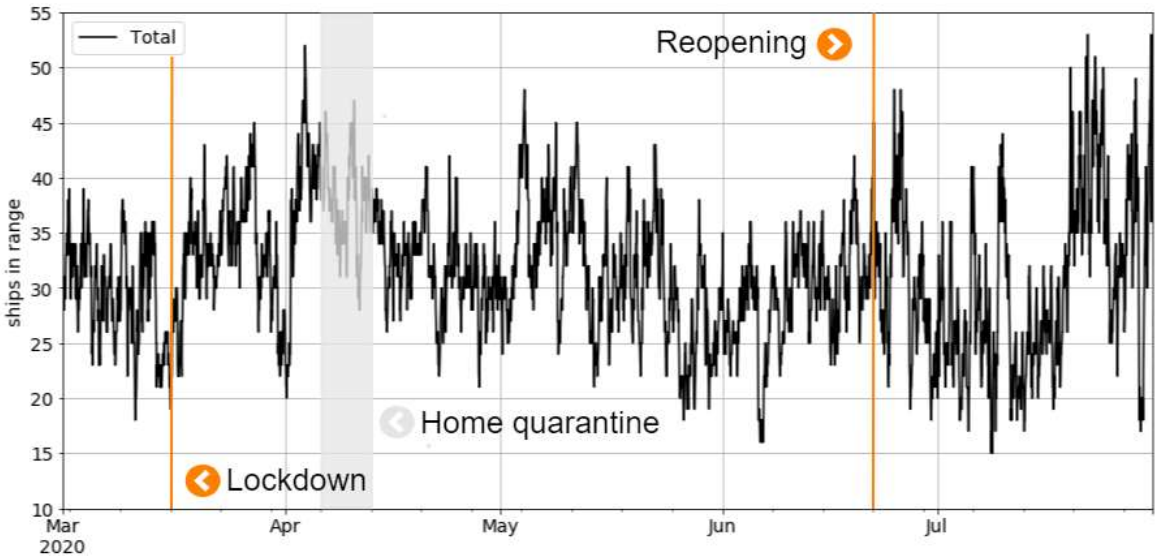

Figure 19. Hourly count of ships in range from March 1 to July 31, 2020

Figure 19 shows a minimum on March 16, when the lockdown entered into force, followed by a bulk mass of vessels that grew up until the very last days of March. After scoring a low on April 1, this mass grew up again to achieve maximum values by April 7, beginning a slight reduction over the time.

The number of vessels in the region increased during the lockdown, especially during the first two weeks of April, which included the home-quarantine week. Daily differences, however decreased during the same period of time, as the plot shows a more stable tendency on 24-hours outlooks.

Table 34. Minimum, average and maximum number of vessels per period
Values in brackets are the difference over the 5-monthly average

Period	Minimum	Average	Maximum
Pre-lockdown[56]	18 (-42.8%)	30 (-6.0%)	39 (+23.9%)
Lockdown[57]	16 (-49.1%)	32 (+1.8%)	52 (+65.3%)
Home-quarantine[58]	28 (-11.0%)	38 (+20.3%)	47 (+49.4%)
Post-lockdown[59]	15 (-52.3%)	31 (-2.3%)	53 (+68.5%)

Considering the strictest lockdown period running from March 16 to June 22, the minimum and maximum number of vessels in range corresponded both to post-lockdown dates, as previously discussed. However, as seen in Table 34, the highest averages were scored during lockdown period, with an average of 32 (+1.8% above average) and peaking at an average of 38 (+20.3% above average) during the home-quarantine period. The lowest difference between the minimum and the maximum number of vessels was also during the home-quarantine period.

As daily differences went down and the average number of vessels went up, **it can be concluded that the number of vessels in the area was significantly increased during the early stages of lockdown, including the home-quarantine week**. Morever, as seen in Figure 19, **these vessels also reamained within the area for a longer time**.

This can be explained through different theories, either the number of vessels at anchor, drifting for orders or moored in the Port of Barcelona increased; the overall speed of the vessels decreased due to ordered slow-steaming[60] practices on board; or the number of vessels returning to their homeport increased as global economy was being shut down. These theories are further discussed in sections 4.1.2, 4.1.3 and 4.1.4.

---

[56] From March 1 to March 15 at 23:00.7
[57] From March 16 at 00:00 to June 22 at 00:00.
[58] From April 6 at 00:00 to April 13 at 00:00.
[59] From June 22 at 01:00 to July 31 at 23:00.
[60] Sailing at reduced speed.

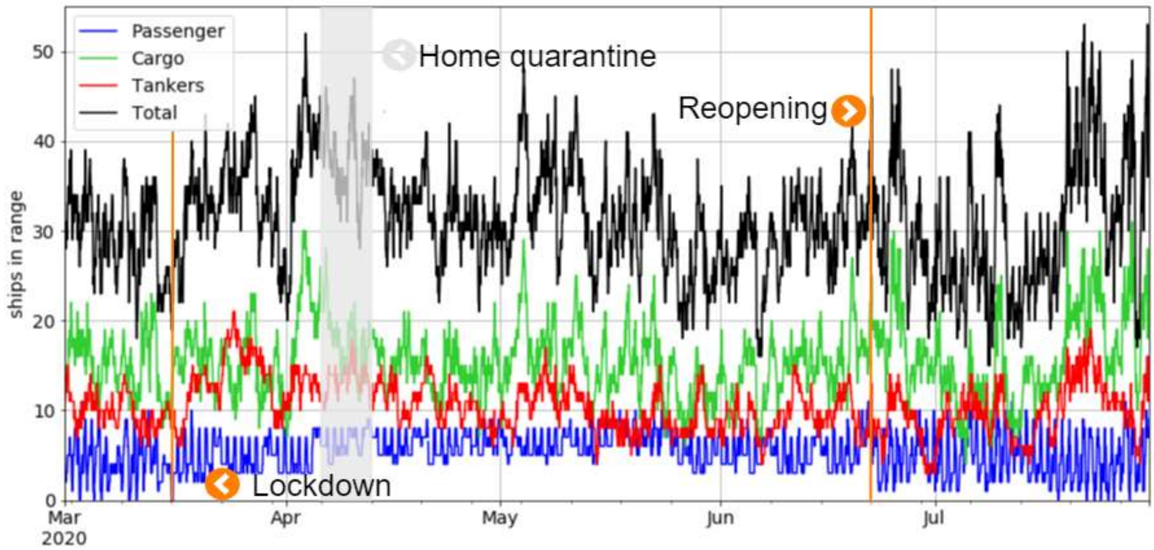

Figure 20. Hourly count of ships in range, divided by type of ship, from March 1 to July 31, 2020

In terms of type of vessel, different trends can be observed over Figure 20. Both cargo and tanker traffic had a high impact on the daily curve of vessels in the area. Worth focusing on passenger vessels traffic, as the lockdown effect can be noticed quite well. Daily changes were actually reduced to minimum and a constant base of 4 vessels was reported in range if compared to pre- and post-lockdown times. These were actually the laid up units in port owing to travel restrictions.

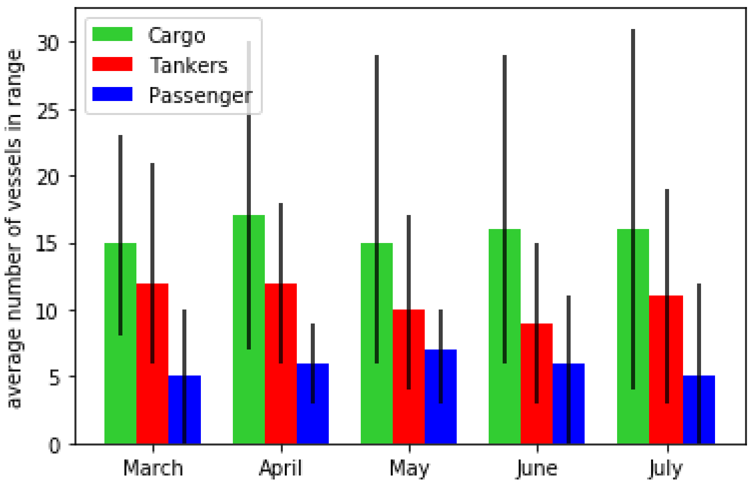

Figure 21. Average number of vessels in range per type of ship, including maximum and minimum values More detailed values available in Annex A3: Table A 1, Table A 3 and Table A 5

In Figure 21, 16 cargo vessels (50% of all vessels), 11 tanker vessels (34.4% of all vessels) and 6 passenger vessels (18.6% of all vessels) were reported on average every hour during 5 months.

The maximum number of cargo vessels was 31 (+98.9% above average), corresponding to July 28, 2020 at 17:00 UTC and July 30, 2020 at 19:00 UTC. Minimum number of cargo vessels was 4 (-74.3% below average), corresponding to July 5, 2020 for at 08:00 UTC. April was the month with the largest number of cargo vessels in the region, with 17 vessels (+4.6% above the average), whereas June and March were the months with the least number of cargo vessels, with averages of 15 (-3.4% below the average), respectively. All in all, average differences were not that much significant.

Regarding tankers, the maximum number was 21 (+100.7% above average), corresponding to March 24, 2020 at 17:00 UTC. Minimum number of tankers was 3 (-71.3% below average) corresponding to June 29 and June 30, 2020. Both March and April were the months with the largest number of tankers in the area, with 12 vessels (+11.6% above average); whereas June was the month with the lowest number, scoring 9 vessels on average (-14.2% below average).

Concerning passenger vessels, the maximum was 12 (+121.7% above average), corresponding to May 7, 2020 between 17:00 UTC and 19:00 UTC. Minimum number was 0, which was scored several times upon resumption of operations. Under normal conditions, the minimum value shall be close to 0 for all the months, as both passenger cruise and ferry vessels do not tend to stay overnight at Barcelona, or within the area. May was the month with the largest number of passenger vessels in the area, peaking at 7 (+18.0% above average); whereas July was the month with the lowest number, socring 5 vessels (-13.2% below average). Comparing this data with Figure 20, the base of passenger vessels in the range was steady at 4 vessels, upon declaring lockdown, as several units were laid up in Barcelona or left adrift as operations were forlough. Limited variation upon arrival and departure explains the inter-daily changes. As the reopening and the new normal were in force, this base completely disapperared.

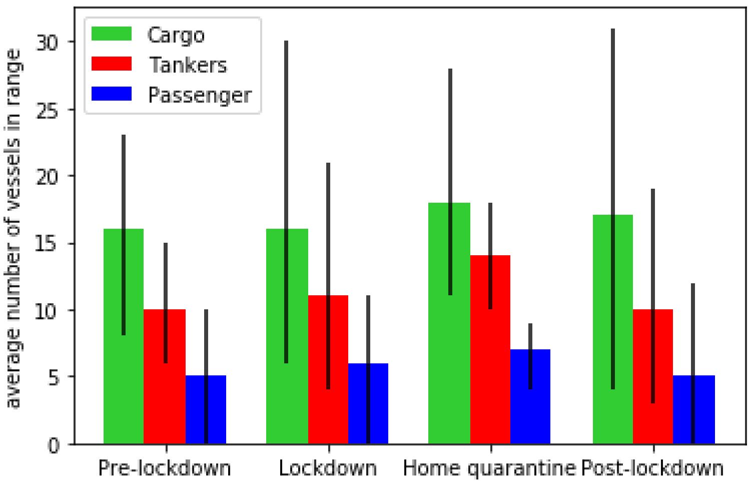

Figure 22. Average number of vessels in range per type of ship and period, including maximum and minimum values
More detailed values available in Annex A3: Table A 2, Table A 4 and Table A 6

By period, Figure 22 shows that minimum and maximum number of cargo vessels in range corresponded again to post-lockdown dates. The highest average corresponded to the home-quarantine period, with an average of 18 vessels (+15.5% above average). In this case, this value was kept above the average also during the post-lockdown scenario. The lowest difference between the minimum and the maximum number of cargo vessels was also during the home-quarantine period.

Regarding tankers, contratry to previous cases, the minimum and maximum values corresponded to the lockdown period, as in Figure 22. The highest average of tankers was found during the home-quarantine time, peaking at 14 vessels (+27.1% above average). This timespan also recorded the least difference between the minimum and maximum number of vessels in the region. They saw also an important increase during the home-quarantine period, which was even higher than the average reported for cargo and for the absolute number of vessels. In fact, the increase was driven by massive offloading of fuel and gas into shore facilities during early lockdown stages.

As with cargo vessels, the minimum and maximum values of passenger vessels corresponded to the post-lockdown period, as seen in Figure 22. The higher average was found during the home-quarantine and lockdown periods, socring at an average of 7 (+21% above average) and 6 (+7.5%), respectively. The lowest difference between maximum and minimum was also reported during the home-quarantine period. The increase in the number of passenger vessels during the home-quarantine and lockdown periods corresponded fully to the fact that most of the vessels were laid up in Barcelona owing to travel restrictions. In fact, the minimum of 4 vessels scored during home-quarantine relates exactly to the laid up ferries moored in the Port of Barcelona. The continuity found in the maximum number of vessels is related to the fact that regular passenger lines were the only ones trading, as cruise vessels were banned at the time.

4.1.2. Vessel status

As discussed in section 3.1.3, only vessel statuses *Underway Using Engine, At Anchor, Not Under Command* and *Moored* were considered in the study. An increased number of overall vessels staying at anchor or adrift could arise from a reduced traffic overall, as vessels had to wait for orders.

As seen in Figure 23, the status *Moored* was the most common, with an average of 56.0%, compared to *Underway* and *At Anchor,* averaging at 28.0% and 15.7%, respectively. The number of vessels *Not Under Command* was merely residual, scoring an average of <0.1%. However, it was more commonly seen during the lockdown period, especially during the last two weeks of March and the first two weeks of May. Only during the first week of April, the number of vessels *Underway* grew up considerably, scoring as the main status for almost 2 days.

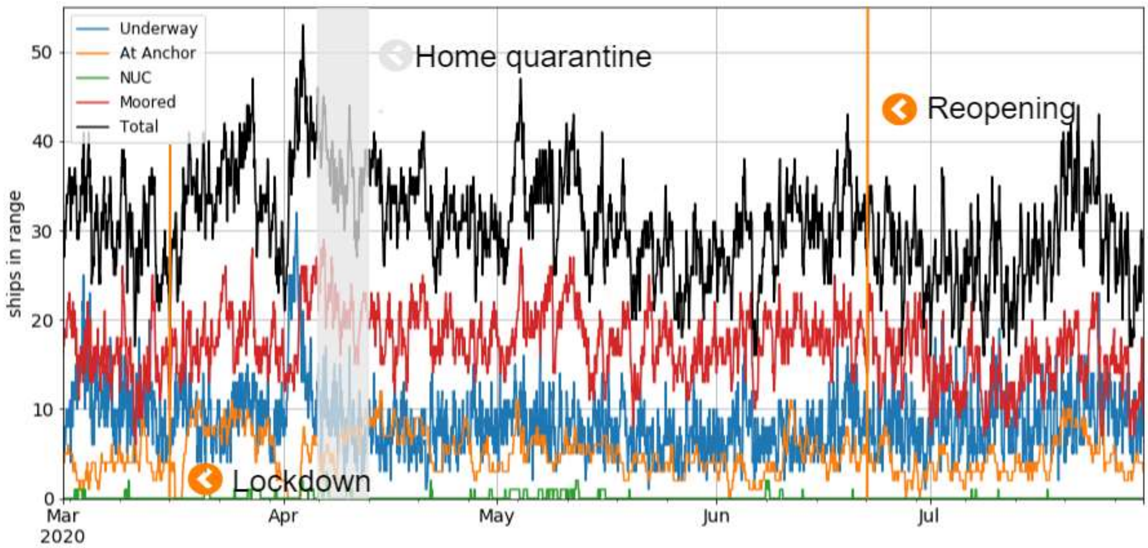

Figure 23. Hourly status of ships in range from March 1 to July 31, 2020

In Figure 24, April, May and June saw a reduction in the number of vessels with status *Underway* and *At Anchor,* while increasing the number of vessels *Moored,* which peaked at 60.6% in June. July was the month with largest number of vessels *Underway,* scoring a 33.0%, while the number of *Moored* vessels also scored a minimum at 51.0%.

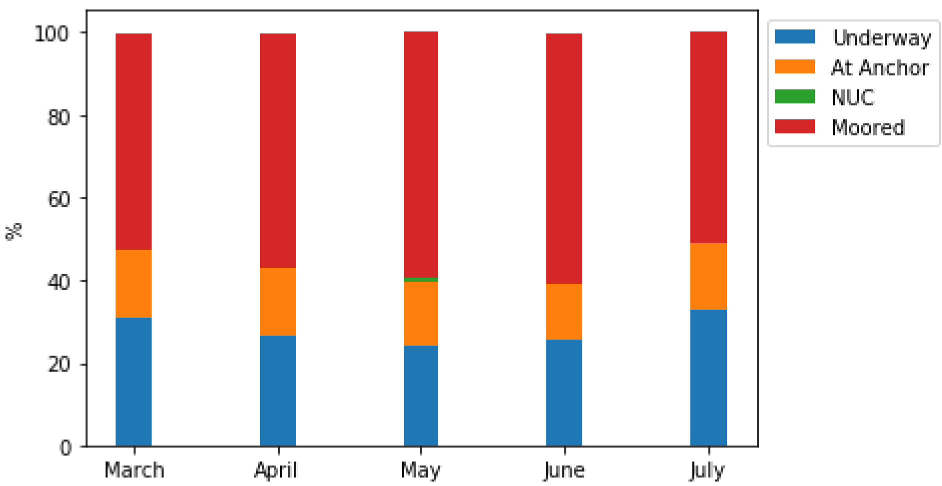

Figure 24. Distribution of status (%) for all vessels in range
More detailed values available in Annex A3: Table A 7

As per Figure 25, the home-quarantine period saw the maximum number of *Moored, At Anchor* and *Not Under Command* vessels, with averages of 58.5%, 17.3% and 0.01%; whereas the number of vessels *Underway* scored the lowest at 23.3%. Pre- and post-lockdown values were consistent and very similar.

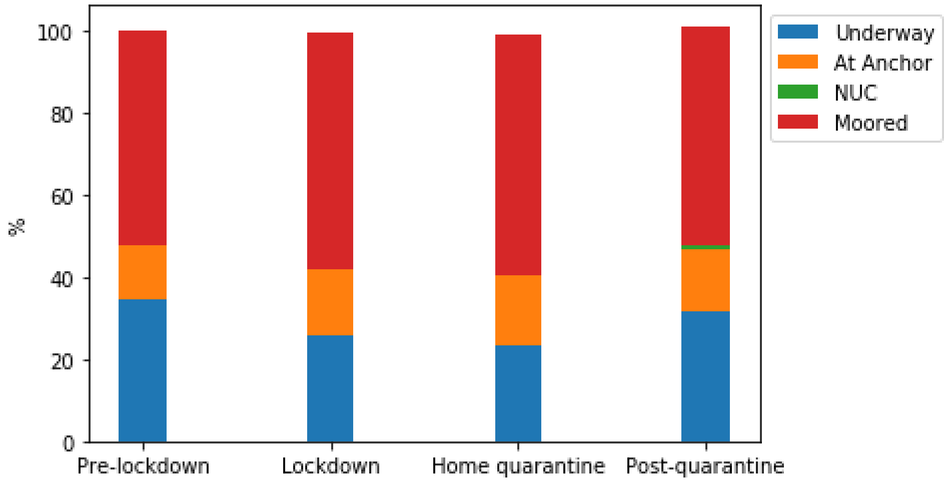

Figure 25. Distribution of status (%) for all vessels in range per period
More detailed values available in Annex A3: Table A 8

As discussed in section 4.1.1, the increase in number of vessels was partly related to an increased number of vessels with static or quasi-static positions, as statuses *At Anchor, N.U.C* and *Moored* did increase during the lockdown period, with special emphasis during the home-quarantine period. In fact, **it is not that more vessels arrived but that vessels did not leave, increasing their time in port or in the anchorage area.**

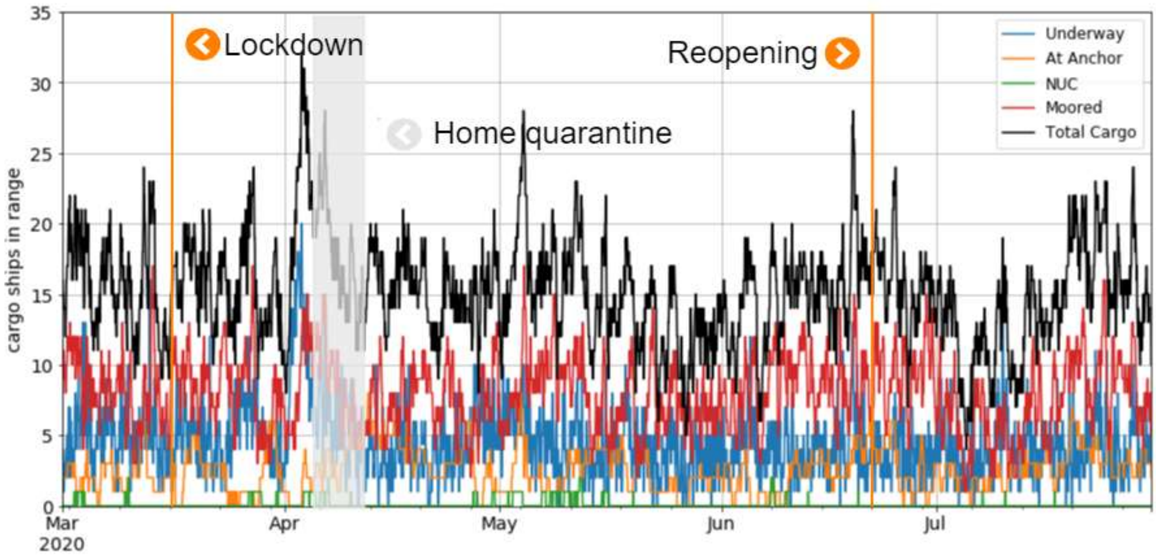

Figure 26. Hourly status of cargo ships in range from March 1 to July 31, 2020

Regarding cargo vessels, as seen in Figure 26, values were consistent and similar to the generic averages. *Moored* vessels led also with an average of 52.6%, followed by *Underway* and *At Anchor* vessels, scoring 29.8% and 17.1%, respectively. Interestingly, the increase in number of *Underway* vessels in April was traced back to an increased in the number of cargo vessels reporting that status, which overtook the status *Moored* as the main one.

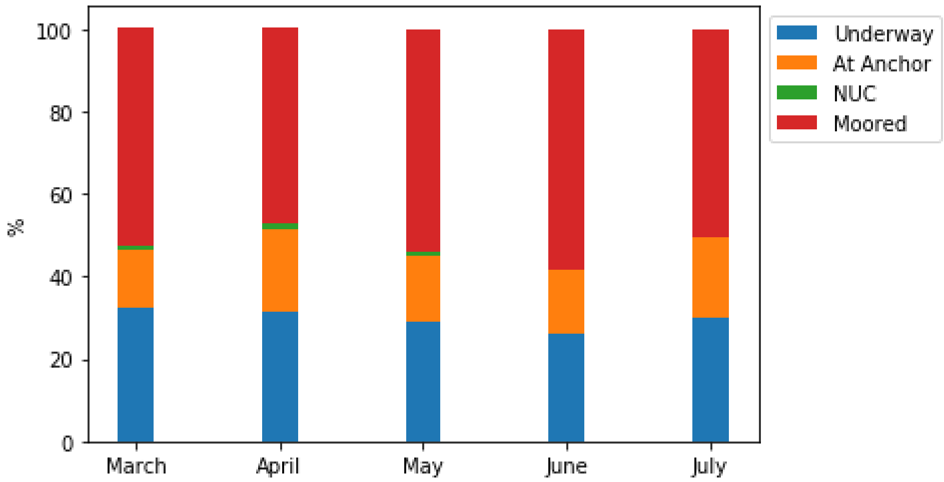

Figure 27. Distribution of status (%) for cargo vessels in range
More detailed values available in Annex A3: Table A 9

As shown in Figure 27, a remarkably increase in the number of *At Anchor* vessels was scored in April, peaking at 20.0%; together with the lowest number of *Moored* vessels, with an average of 47.9%. The number of *Underway* vessels followed a steady trend over the time.

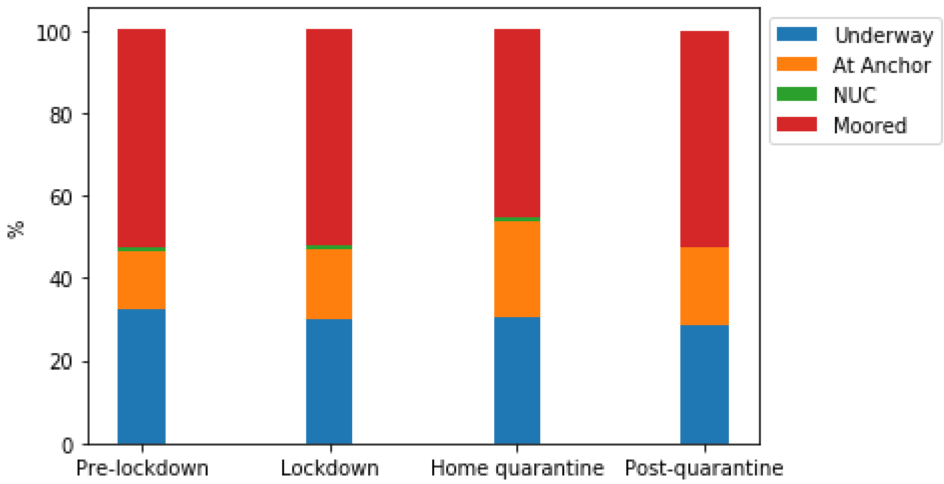

Figure 28. Distribution of status (%) for cargo vessels in range per period
More detailed results available in Annex A3: Table A 10

In Figure 28, the April values were mostly affected by the home-quarantine period, in which the number of *Moored* vessels scored down low at 45.3%, with an increase on the number of units *At Anchor,* which scored a high 23.3%.

In general terms, cargo vessels followed a constant and consistent trend all over the time. Except for the unique situation during the home-quarantine period, in which the number of vessels *At Anchor* increased considerably, to a reduced number of moored vessels. This is explained due to limited working capacity in the port facilities at that time. Worth noting that the home-quarantine period also matched with Holy Week in Spain.

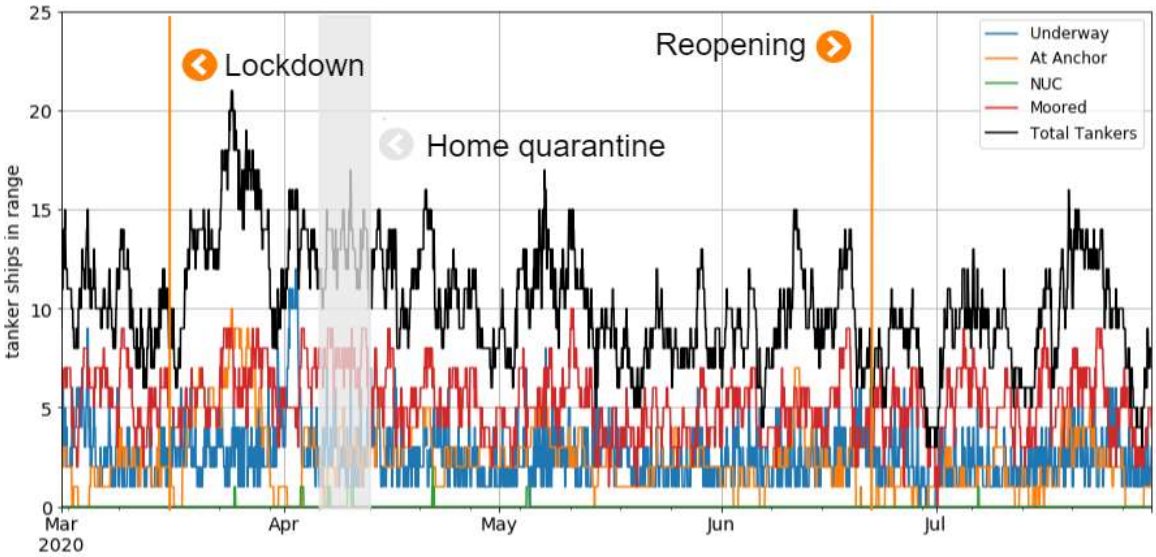

Figure 29. Hourly status of tanker ships in range from March 1 to July 31, 2020

Tanker vessels also followed similar trends to the generic patterns, as per Figure 29. *Moored* vessels were the most common at 53.3%, followed by *Underway* and *At Anchor* vessels which scored similarly at 25.4% and 21.1%, respectively. Worth noting that given the characteristics of cargo offloading operations, and berth availability in Barcelona, it is common for tanker vessels to wait in the anchorage area.

All four statuses followed steady trends during the studied time, as seen in Figure 30. Interestingly, the number of vessels *At Anchor* only reduced in June and July, to an increase in the number of *Moored* vessels. This is a result of the reduced number of vessels in the area that also decongested the port.

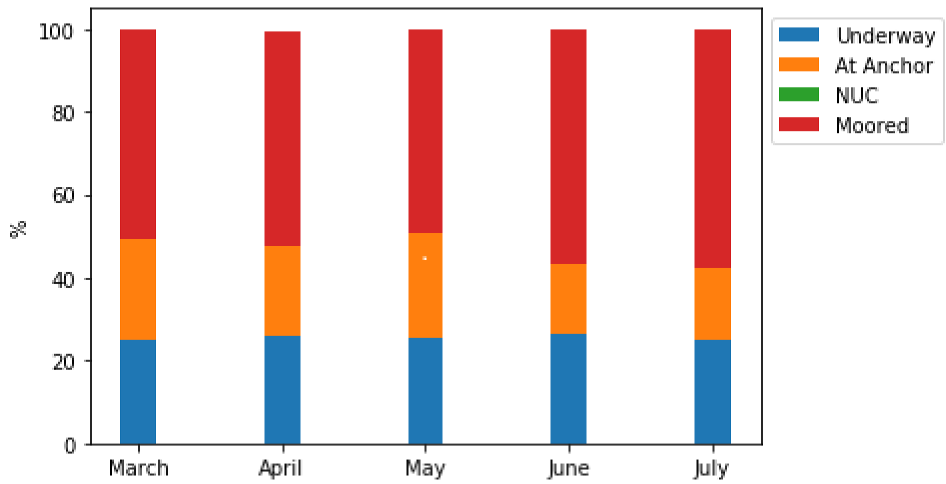

Figure 30. Distribution of status (%) for tanker vessels in range
More detailed values available in Annex A3: Table A 11

The lockdown and home-quarantine periods saw slightly increases in the number of vessels *At Anchor,* which scored at 19.1% and 23.9%, respectively, as shown in Figure 31. These results are consistent with the previously discussed increase in number of vessels by 3.2% and 27.1%, respectively; as berthing capacity in the Port of Barcelona is limited for tankers.

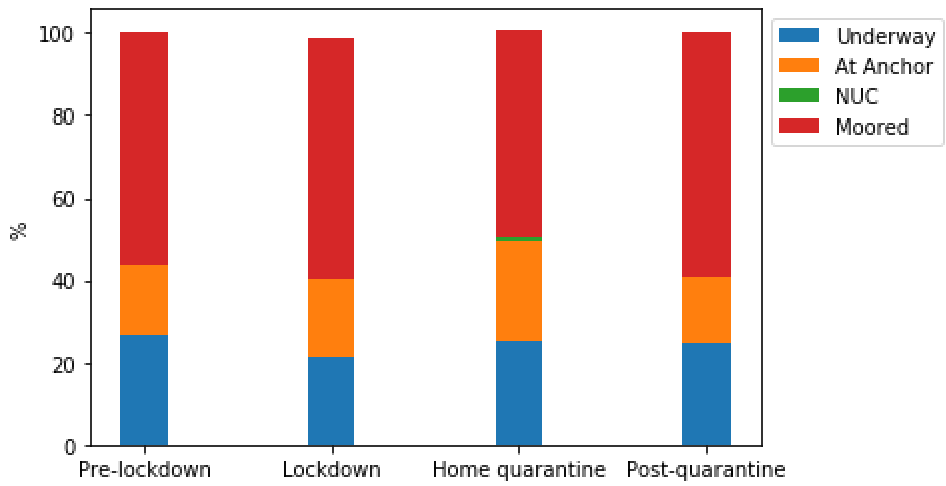

Figure 31. Distribution of status (%) for tanker vessels in range per period
More detailed results available in Annex A3: Table A 12

All in all, tanker vessels did not seem to be so badly affected, as their percentages stayed mostly consistent along the time, with small increases in *At Anchor* and *Moored* statuses. Special consideration, though, for the home-quarantine period, when as with cargo vessels, the number of *At Anchor* units was also increased to a reduced number of *Moored* vessels.

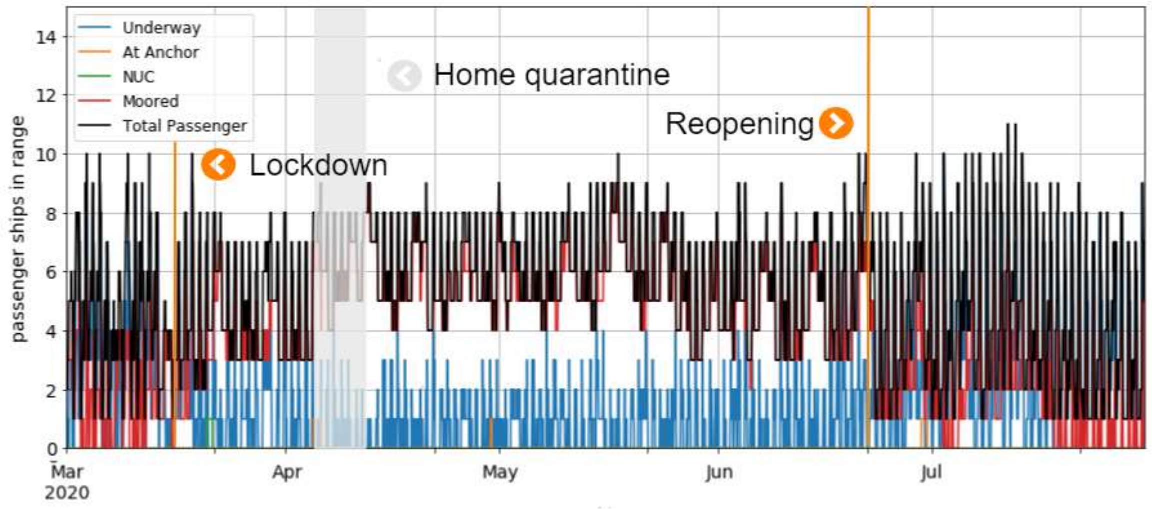

Figure 32. Hourly status of passenger ships in range from March 1 to July 31, 2020

Passenger vessels followed a unique trend compared to cargo and tanker vessels, as seen in Figure 32. In this case, both *Moored* and *Underway* ruled as main status, with 70.9% and 28.7% respectively. The other two statuses were merely residual, scoring 0.3% for *At Anchor* and < 0.01% for *Not Under Command*.

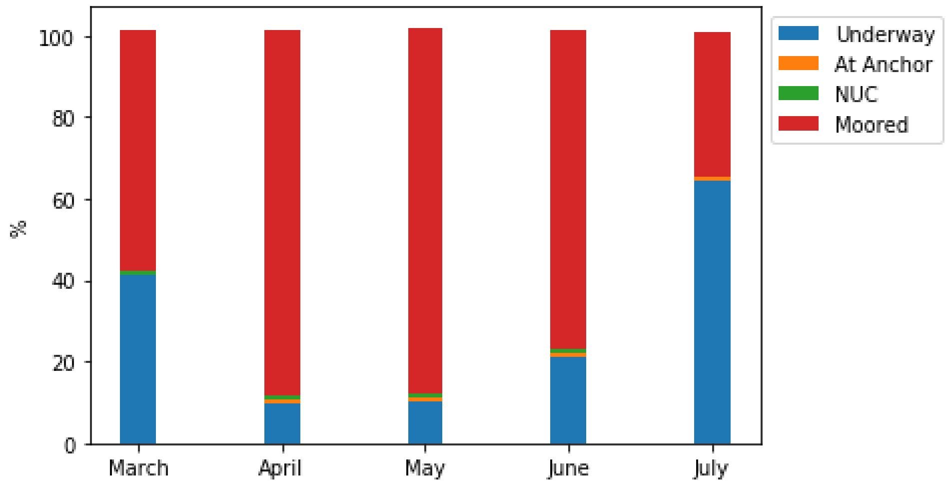

Figure 33. Distribution of status (%) for passenger vessels in range
More detailed values available in Annex A3: Table A 13

The lockdown effect can be clearly noticed on monthly basis in Figure 33. As the number of vessels *Underway* scored down low in April, May and June; while the status *Moored* dominated the scene. Only July saw a more common distribution, in which barely 1/3 of the time, vessels were berthed and 2/3 vessels were making her way.

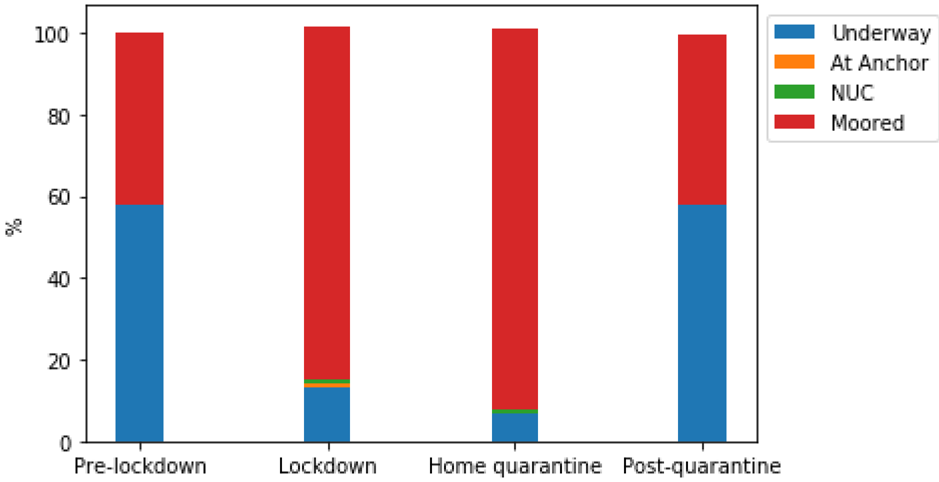

Figure 34. Distribution of status (%) for passenger vessels in range per period
More detailed values available in Annex A3: Table A 14

In Figure 34, the status *Underway* went down to 13.0% and 6.6%, respectively during the whole lockdown and during the home-quarantine periods. Moreover, the status *Moored* saw increases to 86.7% and 93.5% respectively for the same period of time.

Passenger vessels showed the greatest variability in status along the lockdown. In fact, the increase in 7.5% and 21.0% in the number of vessels respectively for the lockdown and home-quarantine periods can be traced back to the condition upon which they stayed longer alongside.

4.1.3. Vessel speeds

Vessel speed is a good indicator of the overall maritime economy. When freight rates are high, so does the demand for vessels too (17). Thus, operators request their crew to operate at higher speeds. However, when the business is down low, operators order reduced speeds to save fuel and reduce operating costs.

For this analysis vessels with status *At Anchor* or located within port premises were dropped out of the dataset, so as to better show the real impact over trading speeds.

As shown in Figure 35, vessels reported an average speed of 10.47 knots. Upon declaring lockdown, the average speed of vessels went down during the last two weeks of March and the whole month of April, to a slightly recover by May and a return to higher, yet more common values by July.

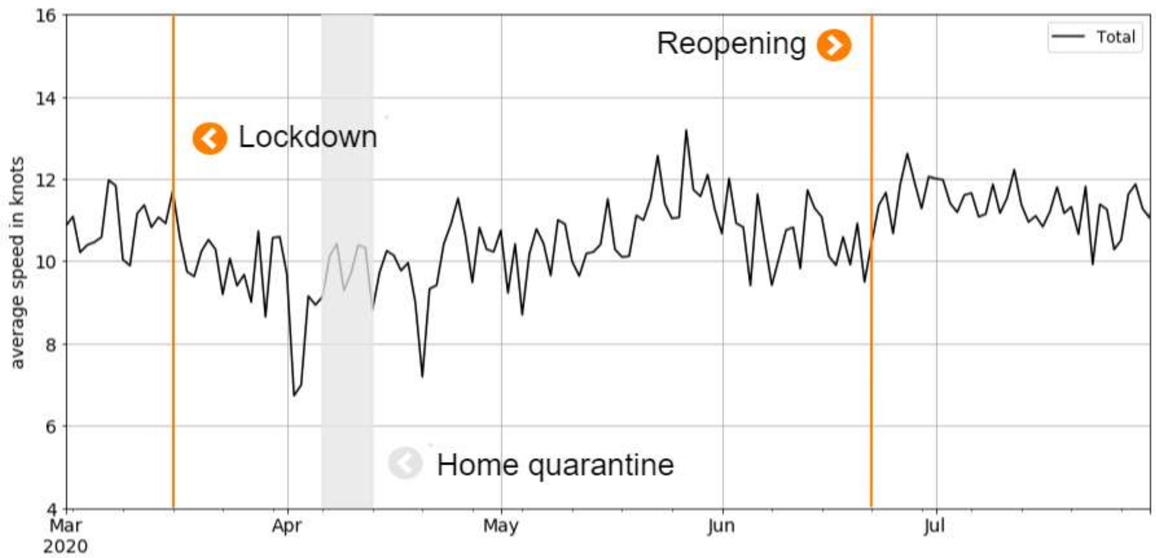

Figure 35. Average speed for all vessels from March 1 to July 31, 2020

In Figure 36, cargo vessels reported an average speed of 10.85 knots (+3.6% above average), tanker vessels reported an average of 8.43 knots (-19.5% below average) and passenger vessels reported an average of 11.28 knots (+7.7% above average).

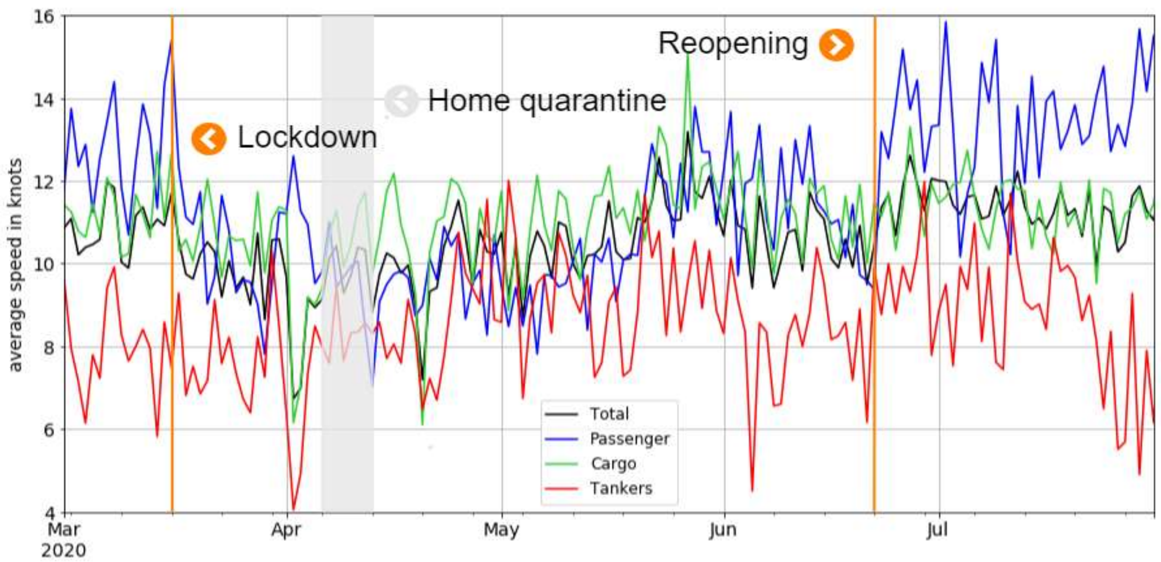

Figure 36. Average speed for all vessels by type of ship from March 1 to July 31, 2020

Cargo vessels followed the general trend in terms of average speed. Regarding tanker vessels, values changed quite abruptly, as after the lockdown, extremely low speeds, below 6 knots were reported several times. Passenger vessels behaved on a different way, as their average speed, usually higher, was kept below common values.

Table 35. Average speeds per month
Values in brackets aret he difference over the 5-monthly average for each traffic

Month	Cargo	Tankers	Passenger	Total
March	10.72 (-1.2%)	7.73 (-8.2%)	11.53 (+2.2%)	10.28 (-1.6%)
April	10.18 (-6.1%)	7.98 (-5.3%)	9.74 (-13.7%)	9.51 (-9.0%)
May	11.13 (+2.6%)	9.38 (+11.3%)	10.26 (-9.0%)	10.59 (+1.3%)
June	11.11 (+2.5%)	8.51 (+1.0%)	11.85 (+5.0%)	10.75 (+2.9%)
July	11.09 (+2.3%)	8.54 (+1.3%)	13.03 (+15.5%)	11.12 (+6.4%)

As stated in Table 35, April saw the most important reduction in average speeds for all traffics, with a reported average of 9.51 knots (-9.0% below average). May saw the recovery of speeds, which scored at higher-than-average values onwards; except for passenger vessels which still recorded average lower figures.

Table 36. Average speeds per period
Values in brackets aret he difference over the 5-monthly average for each traffic

Month	Cargo	Tankers	Passenger	Total
Pre-lockdown	10.83 (-0.1%)	7.81 (-7.3%)	12.50 (+10.9%)	10.62 (+1.6%)
Lockdown	10.76 (-0.8%)	8.41 (-0.2%)	10.41 (-7.7%)	10.16 (-2.8%)
Home-quarantine	10.49 (-3.3%)	8.20 (-2.7%)	9.68 (-14.8%)	9.73 (-6.9%)
Post-lockdown	11.11 (+2.5%)	8.73 (+3.5%)	13.01 (+15.3%)	11.16 (+6.8%)

As in Table 36, the lockdown period saw reduction in the average speed of all traffics, with special emphasis during the home-quarantine period. By far, passenger vessels were the most affected during the lockdown, scoring an average reduction of -7.7% below average, further down to -14.8% during the home-quarantine period. However, they also saw the highest increase in average speed as reopening went effective.

It is certain that slow steaming could be better analyzed if whole trips instead of traffic in a single region were to be considered. However, it can be concluded that this practice was not observed within the vessels in range, *a priori.*

Figures show that during late March and the whole of April, **cargo, tankers and passenger vessels reduced their average speed by -6.1%, -5.3% and -13.7%, respectively. This is consistent with the uncertainties related to the economic situation at that time**. Worth noting that passenger vessels maintained reduced speeds during the whole lockdown period and did not show a recovery well into the reopening. In fact, this speed reduction is also related to an increase in number of vessels reported during these periods, as they were relatively slower, thus staying longer within the area.

4.1.4. Port calls in Barcelona

A total of 2623 ship calls were reported in the Port of Barcelona during the period going from March 1 to July 31, 2020. Of which, 1483 (56.5%) corresponded to cargo vessels; 393 (15.0%) corresponded to tanker vessels and 747 (28.5%) corresponded to passenger vessels.

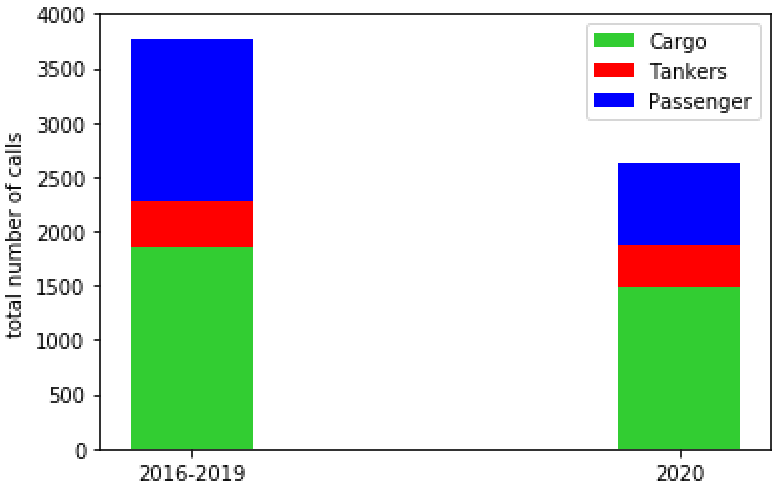

Figure 37. Distribution of number of calls by type of ship from March to July, average 2016 – 2019 (left) and 2020 AIS-based calls (right).
More detailed values available in Annex A3: Table A 15

As seen in Figure 37, the distribution of port calls by ship type changed dramatically compared to the last 4 years, with special emphasis on passenger vessels. **Total number of port calls went down by -30.5% in the same period of time**. Passenger vessels scored down low by -50.2%, whereas cargo vessels lost up to -19.8%. Tanker vessels were the least affected, losing only -7.8% to previous years.

Table 37. Top 5 vessels by number of calls over the 5-month period

Ship	Number of calls
Abel Matutes	111
Tenacia	109
Ciudad de Mahón	98
Cruise Barcelona	64
Cruise Roma	54

Table 37 lists the top 5 vessels by number of calls within the studied 5-month period. As expected, all of them were Ro/Pax ferries on scheduled routes towards the Balearic Islands, Sardinia and Italy.

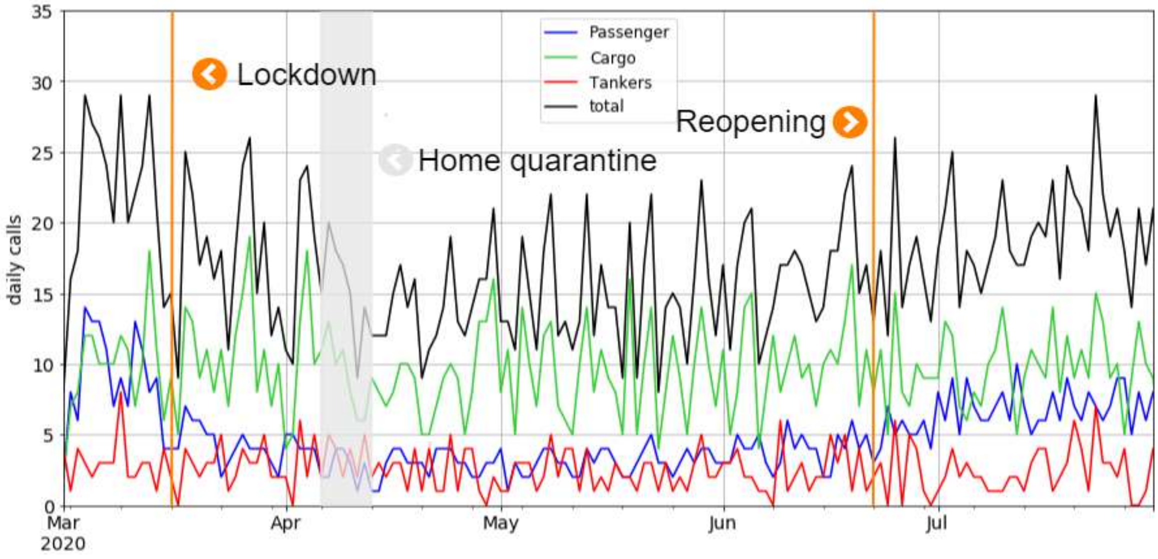

Figure 38. Daily calls in Barcelona from March 1 to July 31, 2020

In Figure 38, the number of calls scored a minimum on Sundays and maximum values in the middle of the week. Cargo and passenger calls were tied up as the main sources of port calls in Barcelona all the way until the lockdown entered into force, when passenger calls fell well below 5 per day. As discussed in section 4.1.1, the bulk of passenger vessels boosting total annual values was non-existent for the ongoing season.

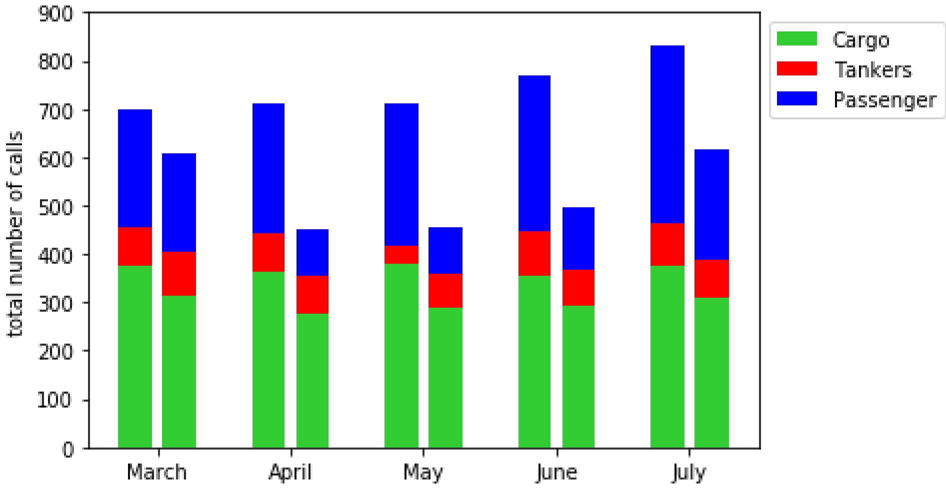

Figure 39. Distribution of number of calls by type of ship and month from March to July, average 2016 – 2019 (left) and 2020 AIS-based calls (right)
More detailed values available in Annex A3: Table A 16

When compared to average calls in the last 5 years, as seen in Figure 39, the number of calls reportedly reduced in all 5 months for all three traffics, except for tankers during the months of March and April. May saw the largest reduction in number of calls, scoring down low by -24.6% for cargo vessels, -18.9% for tankers and -67.1% for passenger vessels, leading to an average of -40.3% less calls when compared to previous years. July showed early signs of recovery. However, values were still well below expected, especially for passenger ships.

Table 38. Average daily calls per month by ship type in 2020
Values in brackets are the difference over the 5-monthly daily average for each traffic

Month	Cargo	Tankers	Passenger	Total
March	11 (+4.9%)	3 (+13.0%)	7 (+34.6%)	20 (+14.5%)
April	10 (-5.1%)	3 (+5.1%)	3 (-36.3%)	15 (-12.4%)
May	10 (-4.5%)	3 (-9.6%)	3 (-36.4%)	15 (-14.3%)
June	10 (+1.1%)	3 (-5.3%)	4 (-12.3%)	17 (-3.7%)
July	10 (+3.5%)	3 (-3.3%)	8 (+50.5%)	20 (+15.8%)

Previous values by daily rates can be seen in Table 38. An average of 18 calls were reported on daily basis during the given period, of which 10 were cargo vessels, 3 were tankers and 5 were passenger vessels. The minimum number of calls was scored in May, with only 15 calls per day on average. Cargo vessels scored a low in number of calls in April, with an average of 9 calls per day; whereas for tankers it was in May, with an average of 3 calls per day. For passenger vessels, the minimum values were scored both in April and May, when only 3 passenger ferries operated on regular basis.

Table 39. Average daily calls per period by ship type in 2020
Values in brackets are the difference over the 5-monthly daily average for each traffic

Month	Cargo	Tankers	Passenger	Total
Pre-lockdown	10 (+0.4%)	3 (+19.5%)	9 (+84.9%)	22 (+27.3%)
Lockdown	10 (-0.7%)	3 (-1.1%)	4 (-28.9%)	16 (-8.8%)
Home-quarantine	9 (-4.6%)	3 (+26.5%)	3 (-48.6%)	15 (-12.4%)
Post-lockdown	10 (+0.8%)	2 (-5.0%)	7 (+38.1%)	19 (+10.5%)

The lockdown saw a reduction in the number of port calls in all three traffics, as shown in Table 39. The most badly affected were passenger vessels, which dropped to 4 calls per day, and went further down to less than 3 calls during the home-quarantine week. Pre-lockdown values were not totally recovered once the reopening was announced and were far from common figures for the season. Regarding tankers, it is worth noting the increase experienced during the home-quarantine period, which saw values quite similar to pre-lockdown. Only cargo vessels seem to have weathered somehow the situation, maintaining similar average number of calls.

4.2. Discussion

Along the lockdown period, as seen in Table 40, **the average number of vessels in the 30 nautical mile range was +1.8% above the inter-annual average**. This was driven mostly by an increased number of passenger vessels at +7.5% above average; and tanker vessels at +3.2% above average. On the other side, cargo vessels saw a slight reduction, by -1.2% below average.

For all of types of ships, **the number of vessels reporting statuses *Moored* or *At Anchor* was well above the inter-annual average**, whereas the vessels *Underway* went down, with special emphasis on passenger vessels.

Table 40. Vessel characteristics during the lockdown
Variations are based on 5-monthly averages

Type of ship	Vessels	Moored / At Anchor	Underway	Speed	Calls
Cargo	-1.2%	73.9% (↑)	25.6% (↓)	-0.8%	-0.7%
Tankers	+3.2%	77.7% (↑)	21.1% (↓)	-0.2%	-1.1%
Passenger	+7.5%	93.5% (↑)	6.6% (↓)	-7.7%	-28.9%
Total	+1.8%	65.3% (↑)	34.4% (↓)	-2.8%	-8.8%

Speeds also saw a reduction, scoring low **by -2.8%** below average, **driven by lower recorded speeds on passenger vessels**, which went down by -7.7% below average. The reductions in cargo and tanker vessels were not so remarkable.

So did the number of calls in general, scoring down **by -8.8%** below average, **influenced only by the number of passenger vessels which were well below the average** in 2020 and well below the accumulated over the past 5 years, as shown in Figure 37 and Figure 39.

Table 41. Vessel characteristics during home-quarantine
Variations are based on 5-monthly averages

Type of ship	Vessels	Moored / At Anchor	Underway	Speed	Calls
Cargo	+15.5%	68.6% (↓/↑)	30.6% (=)	-3.3%	-4.6%
Tankers	+27.1%	74.4% (↓/↑)	25.4% (=)	-2.7%	+26.5%
Passenger	+21.0%	93.5% (↑)	6.6% (↓)	-14.8%	-48.6%
Total	+20.3%	75.8% (↓/↑)	23.3% (↓)	-6.9%	-12.4%

Focusing on the home-quarantine period, the previously mentioned **increase in the number of vessels in range was even higher,** scoring **at +20.3%** above average, as seen in Table 41. In this case, **all three traffics showed important increases**.

The number of vessels reporting status *Underway* was very similar to the average, whereas **the number of berthed vessels went down to an increased number of vessels *At Anchor***, related to a reduced port activity at that time.

All the traffics saw a greater reduction in speed, at -6.9% below average; which bottomed again for passenger vessels, loosing up to -14.8% below average.

The number of calls also reduced by -12.4% below average, both over the studied period and the accumulated 5 years values, with a **major downfall by -48.6% in passenger vessel calls** to increased +26.5% tanker vessel calls.

Under these conditions, seems clear that the increase in number of vessels was driven by changes in the way vessels operated. In fact, it is not that more vessels were trading in the area, but that normal traffic stayed over longer periods of time within it.

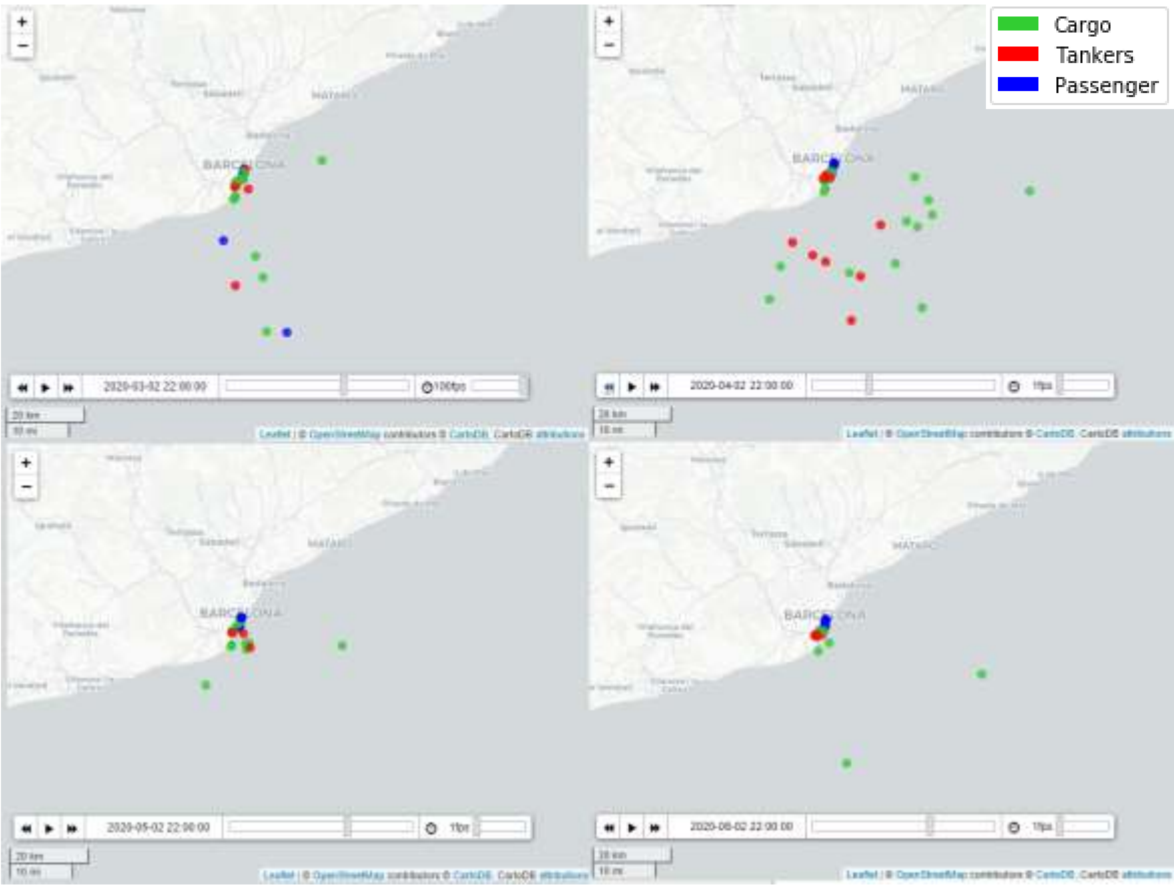

Figure 40. Vessels around Barcelona on March 2 (upper left), April 2 (upper right), May 2 (lower left) and June 2 (lower right)

Supporting the main points, Figure 40 shows the evolution of number of vessels on the 2nd day of March, April, May and June at 22:00 UTC. On March 2, a common mixture of cargo, tanker and passenger vessels could be seen approaching the Port of Barcelona, average speeds were 10.2 knots for cargo vessels, 9.2 knots for tankers and 13.2 knots for passenger vessels, all of them above the 5-monthly average. On April 2, the number of vessels increased dramatically, and their average speeds went down to 5.9 knots for cargo vessels, 3.7 knots for tankers and 12.6 knots for passenger vessels, well below the average. Both May 2 and June 2 showed a very similar trend, with a reduced number vessels in the vicinity and average speeds of 8.4 and 10.0 knots for cargo vessels, 8.8 and 9.5 knots for tankers and 11.4 and 13.5 knots for passenger vessels, respectively.

Figure 41. Vessels in Barcelona on March 2 (upper left), April 2 (upper right), May 2 (lower left) and June 2 (lower right)

For the same period of time, Figure 41 shows the number of vessels berthed at Barcelona. In general terms, May 2 showed a larger number of vessels than April 2, May 2 and June 2. This was especially seen in the container ship terminals and the passenger terminals, where only berthed Ro/Pax ferries and two cruise ships disembarking stranded crew members could be spotted. Interestingly, the number of tankers berthed in Barcelona hit a peak on June 2, compared to the other 3 days.

Not all traffics weathered the pandemic the same way. In fact, two major responses can be seen. One concerning cargo and tanker vessels and the other one regarding passenger vessels.

During the very early stages of the lockdown, including the home-quarantine period, uncertainty arising from the ongoing situation drove cargo and tanker vessels to adjust their speeds and reduce the overall number of calls. Several vessels transitioned into short layup, waiting for orders regarding their operation. This explains the increase in the number of vessels reporting *At Anchor* status and the fact that most vessels with status *Underway* reduced their speeds, to minimum required for steering. In fact, on April 2, plenty of vessels were occupying the area around Barcelona harbor, while port facilities were actually operating below their maximum capacity. By May, both traffics saw a general revamp as the general pandemic situation in Europe saw prominent figures. Several companies adjusted their capacities to meet the existing demand, and as uncertainty diluted, maritime traffic returned to a more normal condition.

Regarding passenger ship traffics flows, trends are extremely dependent on travel restrictions. In fact, recovery is limited for passenger ferry traffics and null for cruise vessels. By June, terminals in Barcelona should have been full of cruise vessels, whereas current traffic is still non-existent. Their vulnerability to the pandemic, together with the negative public opinion related to the cruise business have had a massive impact on them.

In short, the pandemic has driven vessels to change the way they are being operated, and adjustments were necessary in all traffics so as to transition the new normal towards the industry.

Chapter 5. Emissions inventory

This chapter presents the fuel consumption and emissions inventory of major air pollutants, namely CO_2, SO_2, NO_X and PM; related to the maritime traffic within the area of Barcelona for a period of 5 months. It aims to assess the real impact that COVID-19 had in the overall emission figures, and finding a qualitatively correlation between air quality and ship-related emissions.

The chapter also introduces the results and calibration of the mathematical model discussed in section 3.2.2, to be used when enhanced technical data is limited or not available.

The discussion aims to provide answers to the following questions:

1. Has there been a variation in fuel consumption related to changes in vessel operation?
2. Have emissions been lower during the most restrictive lockdown days?
3. What is the relation between vessel traffic and emissions?
4. How important are ship-related emissions to air quality in Barcelona?

Air quality was assessed through SO_2, NO_X and PM average concentration values (in $\mu g/m^3$) provided by the Meteorological Service of Catalonia in selected locations across downtown Barcelona, i.e. Palau Reial, Eixample, Poblenou and Ciutadella. Unfortunately, not all the stations had accurate recording for all pollutants.

Worth noting that air pollutant concentrations are loosely related to emissions, as dispersion is heavily dependent on weather and atmospheric conditions (43). Therefore, the assessment of air quality v. emission ratios was done mostly based on daily trend variations.

5.1. Results

Values presented are the total computed through the STEAM v.2 method as stated in section 3.2.1 for every vessel and every stage, considering the technical particulars of each unit, obtained from the IHS database. Table 42 shows the totaling values of fuel consumption and emissions.

Table 42. Total values of fuel consumption and emissions over 5-months period

Item	Total value (tons)
Fuel consumption	40421
CO_2	108603
SO_2	403
NO_X	2309
PM	81

Table 43. Top 5 vessels by FC and emissions over the 5-month period

Ship	FC (tons)	CO_2 (tons)	SO_2 (tons)	NO_X (tons)	PM (tons)
Volcán de Tinamar	5223	16766	52	217	9.1
Cruise Barcelona	2041	6660	21	94	3.6
Cruise Roma	1912	6430	20	90	3.4
Tenacia	1649	5266	17	83	2.9
Nápoles	1328	3658	< 1	68	2.3

Table 43 shows that the top 5 most polluting vessels were in fact Ro/Pax ferries on regular schedule. MS *Volcán de Tinamar* topped the list with extremely high values. The 2nd and 3rd position were for the Italian sisterships, MS *Cruise Barcelona* and MS *Cruise Roma*[61]. Interestingly, the 5th position, allocated to the MS *Nápoles,* shows very low SO_2 emissions, thanks to her LNG-fueled engine.

Getting back to Table 37 in section 4.1.1, only 3 out of 5 vessels are found in Table 43, and the order differs greatly. Both MS *Volcán de Tinamar* and MS *Nápoles* were actually laid up for most of the lockdown period in Barcelona, thus explaining higher values. Meanwhile, MS *Cruise Barcelona* and MS *Cruise Roma* with half the number of calls than MS *Tenacia* showed higher values, owing to the fact both are actually the largest Ro/Pax ferries in the world, with extremely high power demands.

5.1.1. Fuel consumption

Total computed fuel consumption for the 5-month period was 40421 tons of fuel, which led to an average of 263 tons of fuel per day.

Table 44. Computed fuel consumption (tons) on monthly basis

Month	Fuel consumption (tons)
March	8242
April	8048
May	8882
June	7330
July	7779

[61] Results for MS *Cruise Roma* might actually be different, as the vessel power demands while in port are supplied by a set of lithium batteries. Thus, without emissions.

Table 44 shows that June was the month with the lowest fuel consumption, with a daily average of 244 tons of fuel per day. May peaked as the month with the largest fuel consumption, with a total of 8882 tons of fuel and a daily average of 287 tons per day.

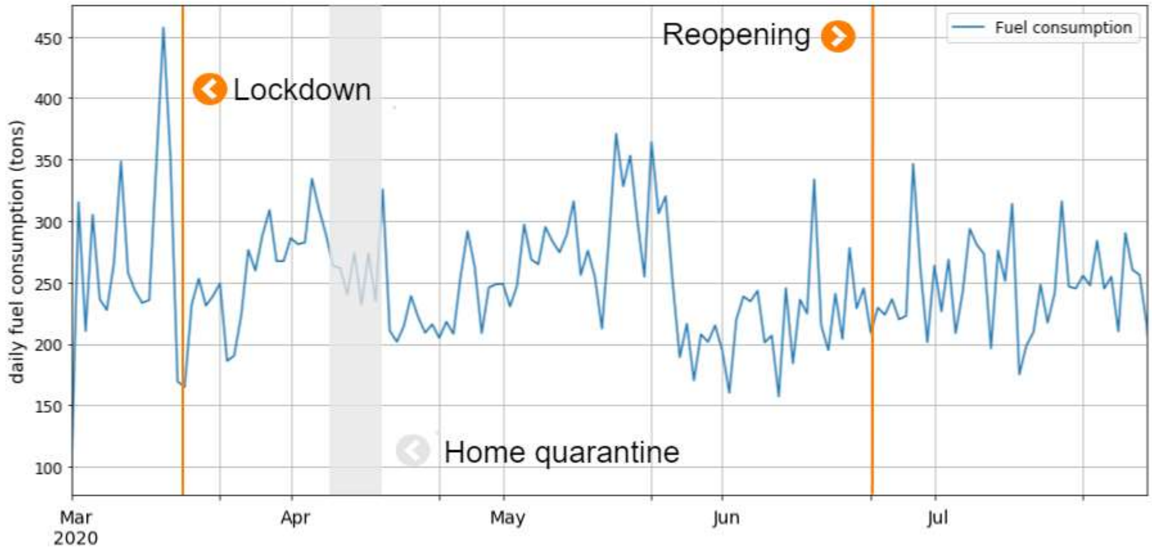

Figure 42. Daily changes in fuel consumption from March 1 to July 31, 2020

Figure 42 shows higher daily consumptions in average during the pre-lockdown period. Interestingly, the highest and lowest values were both consecutive, on March 14 and March 16, when the lockdown entered in force. Peaks mostly concentrated at the beginning of every month, whereas lower consumptions were reported by the end.

Table 45. Computed average fuel consumption (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Month	Fuel consumption (tons)
Pre-lockdown	277 (+5.2%)
Lockdown	266 (+1.1%)
Home-quarantine	275 (+4.6%)
Post-lockdown	251 (-4.6%)

As seen in Table 45, values were 5% above the average for the pre-lockdown and during the home-quarantine periods. However, they went down as the lockdown was subsequently extended and reached below the average levels in the post-lockdown period.

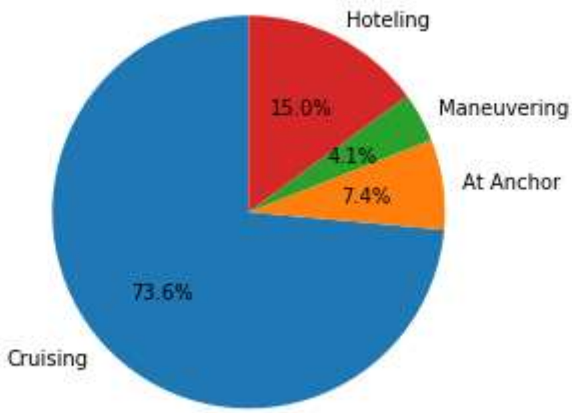

Figure 43. Distribution of fuel consumption per stage

The cruising stage represented up to 74% of total fuel consumption, compared to 15% related to the hoteling phase, 7.4% to the at anchor phase and a residual 4% related to the maneuvering stage, as shown in Figure 43.

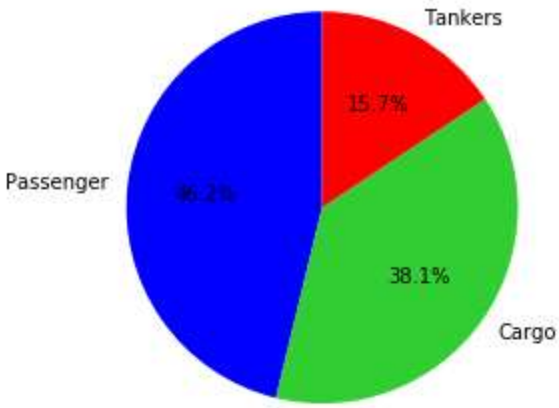

Figure 44. Distribution of fuel consumption per type of ship

Per ship, as seen in Figure 44, passenger vessels accounted for up to 46.2% of total fuel consumption, compared to a 38.1% related to cargo vessels and 15.7% related to tanker vessels. **Although, passenger vessels were only the second largest group both in number of calls and number of vessels in range, even falling third during the most restrictive lockdown days, they ranked as the highest in term of fuel consumption**. In fact, passenger vessels have overall higher installed power on board than their cargo counterparts, given their higher power demand to keep shipboard services running for guests.

5.1.2. Emissions

Carbon dioxide (CO_2)

A computed total of 108603 tons of CO_2 were poured onto the atmosphere as a result of maritime traffic during the 5-month period. This led to an average daily emission rate of 888 tons per day.

Table 46. Computed CO_2 emissions (tons) on monthly basis

Month	CO_2 (tons)
March	27716
April	27329
May	29828
June	25184
July	26262

In Table 46, May peaked as the month with the highest emissions of CO_2, totaling 29828 tons. On the other side, June was the month with the lowest total emissions, totaling 25184 tons. These values are consistent with fuel consumption, as discussed in section 5.1.1, given that CO_2 emissions are fuel-related, thus a lower consumption led to a lower number of emissions, as well.

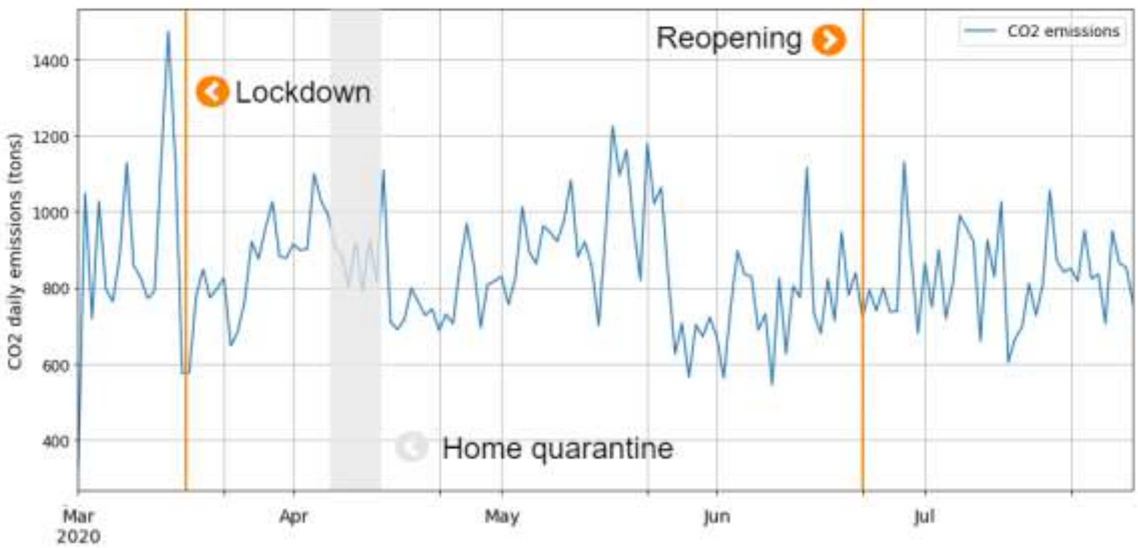

Figure 45. Daily changes in CO_2 emissions from March 1 to July 31, 2020

Figure 45 shows as expected that CO_2 emission trends matched fuel consumption. Worth noting the maximum and minimum values recorded consecutively between March 14 and March 16, related to the uncertainty of the newly imposed lockdown.

Table 47. Computed average CO_2 emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	CO_2 (tons)
Pre-lockdown	921 (+3.7%)
Lockdown	904 (+1.8%)
Home-quarantine	949 (+6.9%)
Post-lockdown	847 (-4.6%)

In Table 47, lockdown values were slightly below the pre-lockdown daily emission rates. As with fuel consumption, CO_2 rates increased slightly during the very first lockdown days and reduced as time went by. Post-lockdown values were lower than pre-lockdown and lockdown emissions, owing mostly to reduced traffic in the area if compared to the other periods.

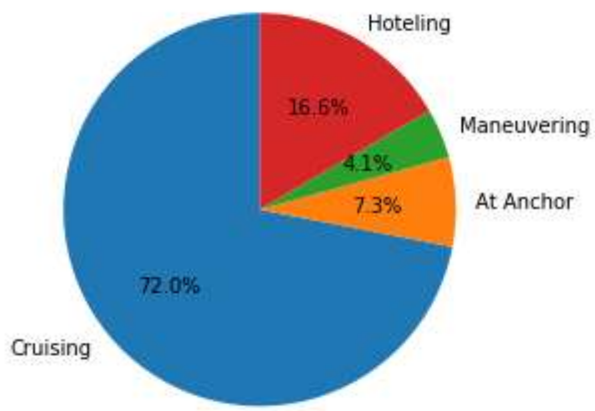

Figure 46. Distribution of CO_2 emissions per stage

As seen in Figure 46, CO_2 emission rates were mostly related to the cruising stage; scoring a 72.0%. The hoteling stage represented 16.6%, higher by a point than fuel consumption, owing to the fact that carbon content is slightly higher in MGO than in LSHFO, which was used in the hoteling and cruising phases respectively.

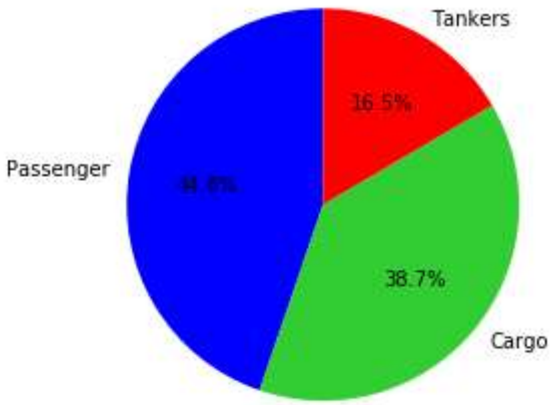

Figure 47. Distribution of CO_2 emissions per type of ship

Figure 47 shows a point higher contribution of tanker vessel to a point lower of passenger vessels in CO_2 emissions, when compared to fuel consumption.

Oxides of sulfur (SO_2)

The 5-month period resulted in a computed total of 403 tons of SO_2, related to maritime traffic. This led to an average rate of 2.6 tons per day. As SO_2 is a fuel-related emission, similar trends as in CO_2 and fuel consumption are expected.

Table 48. Computed SO_2 emissions (tons) on monthly basis

Month	SO_2 (tons)
March	81
April	80
May	89
June	75
July	78

Table 48 shows that May peaked as the month with highest SO_2 emissions, totaling 89 tons. As in previous cases, June scored down to 75 tons, computed through both methods.

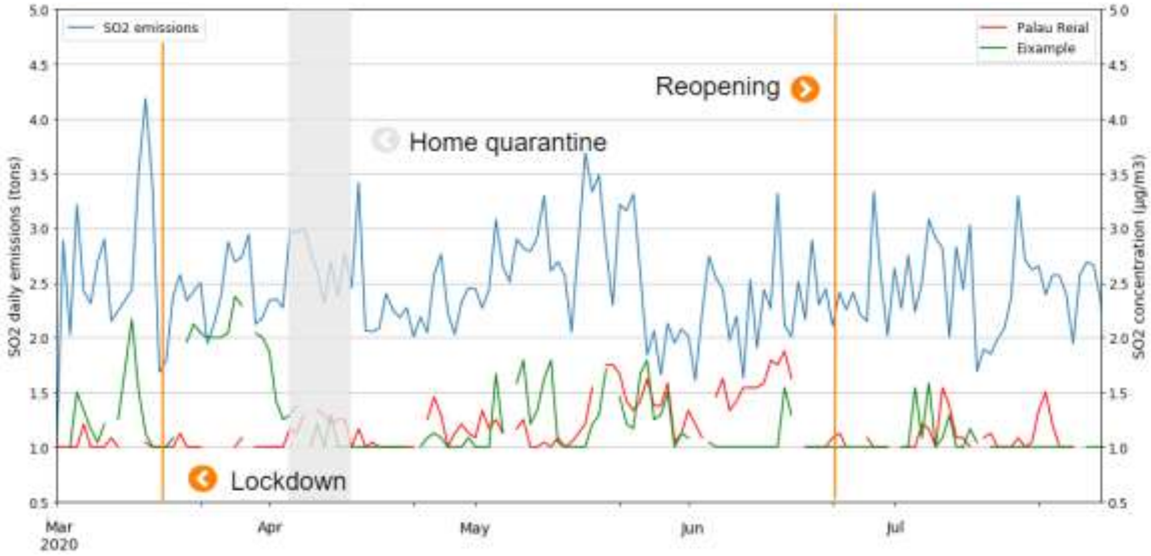

Figure 48. Daily changes in SO_2 emissions and concentrations from March 1 to July 31, 2020

Figure 48 shows the same trend as with fuel consumption and CO_2 emissions. Daily rates were compared to SO_2 concentration in two locations across the city (Palau Reial and Eixample), which led to quite interesting results. In fact, peaks in SO_2 concentration followed mostly the same trend as SO_2 emissions from shipping, with similar variations between days as with the computed emissions.

In light of the plot and given that Barcelona is a low SO_2-concentration city, seems clear that ship emissions as far as 30 nautical miles from the city do have an impact on air quality in the city.

Table 49. Computed average SO_2 emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	SO_2 (tons)
Pre-lockdown	2.7 (+3.9%)
Lockdown	2.7 (+3.8%)
Home-quarantine	2.8 (+7.7%)
Post-lockdown	2.5 (-3.8%)

Table 49 shows that pre-lockdown and lockdown daily rates were quite similar. However, bearing in mind that the home-quarantine values were above the average, higher variability in emissions within the lockdown period explains these figures. Post-lockdown values felt by -3.8% owing to an overall reduction in the number of vessels in the area.

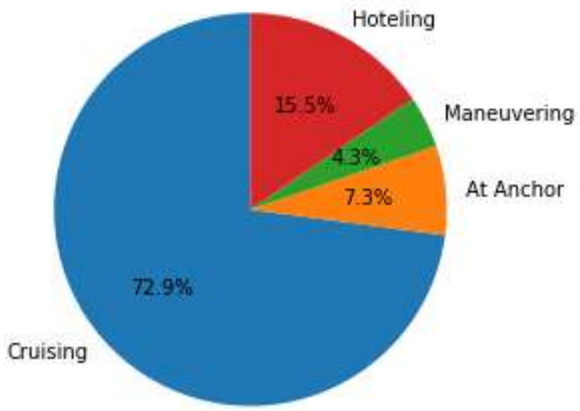

Figure 49. Distribution of SO_2 emissions per stage

Figure 49 shows little changes in terms of contribution per phase to total SO_2 emissions when compared to fuel consumption. This is partly due to the fact that both LSHFO and MGO have by default similar sulfur contents. All in all, the cruising stage represented 72.9% of total SO_2 emissions, compared to 15.5% related to the hoteling stage, 7.3% to anchored vessels and 4.3% related to the maneuvering phase.

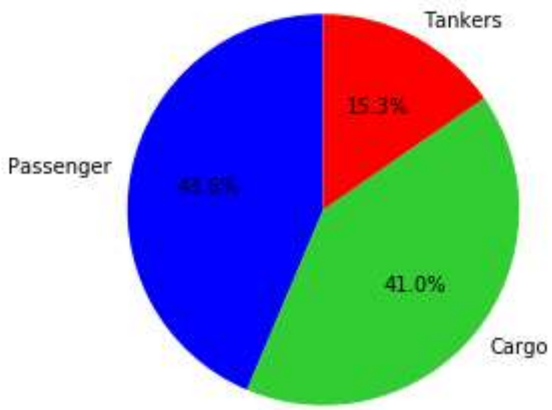

Figure 50. Distribution of SO_2 emissions per type of ship

As seen in Figure 50, the distribution by type of ship was quite similar to the other fuel-related emissions and consumption. Worth noting that the slight reduction in tanker and passenger vessel emissions were related to the fact that some of them are powered by LNG; with residual sulfur content, leading to an overall reduction of these types of vessel in the total contribution to SO_2.

Oxides of nitrogen (NO_X)

A total of 2309 tons of NO_X were estimated to result from maritime traffic within the 5-month period. This accounted for an average daily rate of 15.1 tons per day.

Table 50. Computed NO_X emissions (tons) on monthly basis

Month	NO_X (tons)
March	470
April	466
May	509
June	413
July	451

As in Table 50, May and June recorded the highest and lowest number of NO_X emissions; totaling 509 tons of NO_X in May; and 413 tons of NO_X in June.

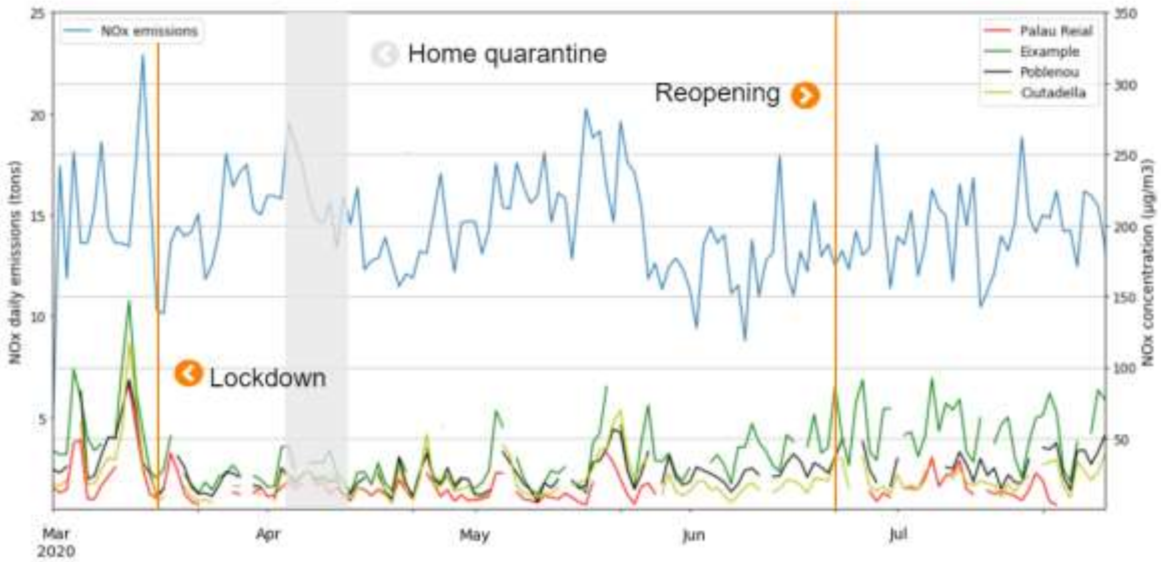

Figure 51. Daily changes in NO_X emissions and concentrations from March 1 to July 31, 2020

Figure 51 shows a similar trend than fuel consumption, yet different in the magnitude of peaks, owing to the fact that NO_X emissions are not fuel- but engine-related. Note that, for this scenario values were compared to the NO_X concentrations in Palau Reial, Eixample, Poblenou and Ciutadella. NO_X concentrations went clearly down upon declaring the lockdown, and did not increase again until late May. However, most of the peaks and their relative change followed a very similar trend than the actual emissions related to the maritime traffic. A clear example is the period running from March 14 to March 16, where a maximum peak and a down low valley were both scored at the same time.

Wheeled traffic and airport-related emissions have an important contribution in total NO_X concentrations across the city, however given their reduced activity during the strictest lockdown days, it is feasible to say that the lower base was mostly related to the 30 nautical miles maritime traffic.

Table 51. Computed average NO_X emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	NO_X (tons)
Pre-lockdown	15.1 (-0.2%)
Lockdown	15.3 (+1.3%)
Home-quarantine	16.2 (+7.3%)
Post-lockdown	14.5 (-4.0%)

Table 51 shows that lockdown NO_X emissions were slightly above the average, with special emphasis on the home-quarantine period, when they were +7.3% above. Pre-lockdown values were not recovered upon reopening.

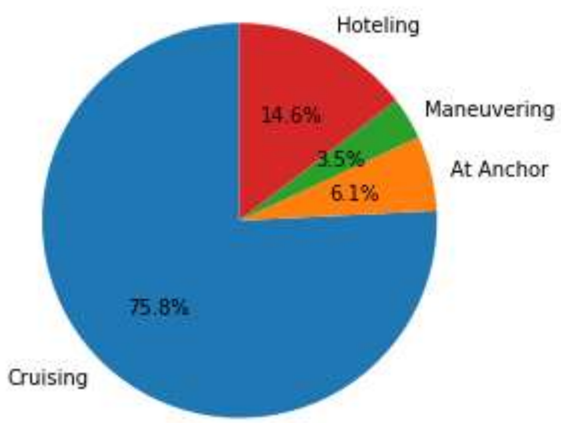

Figure 52. Distribution of NO_X emissions per stage

NO_X emissions are heavily dependent on the engine characteristics and its revolutions. This is the main reason why in Figure 52, the cruising-related emissions were higher than in previous cases, contributing with 75.8%; compared to 14.6% during the hoteling phase, 6.1% while at anchor and

3.5% during maneuvering. The difference might actually be impacted by the engine configuration introduced in the model, as preference was given to auxiliary engines with higher revolutions and lower emission factors, during the hoteling phase over main engines operating at its fullest during the cruising stage.

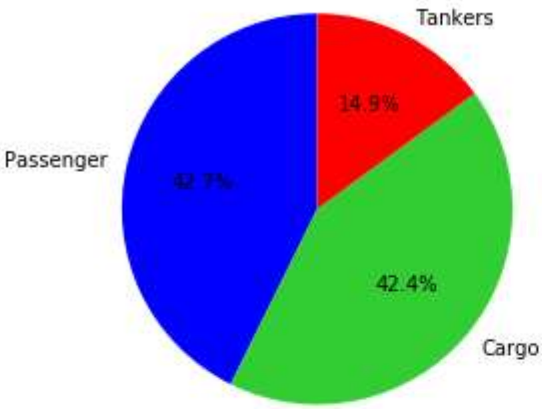

Figure 53. Distribution of NO_X emissions per type of ship

A similar reasoning applies in Figure 53, as cargo and tanker vessels saw an increased contribution to NO_X emissions compared to passenger vessels in previous cases. This arises from the fact that passenger vessels are more modern, thus falling in Tier II regulations, and operate mostly with medium-speed engines, with lower NO_X emission factors than their counterpart.

Particulate matter (PM)

A computed total of 81 tons of PM resulted from maritime traffic within the 5-month period. This led to an average rate of 0.52 tons per day.

Table 52. Computed PM emissions (tons) on monthly basis

Month	PM (tons)
March	16.4
April	16.2
May	17.8
June	14.5
July	15.7

Table 52 shows that May and June ranked again as the months with the highest total of PM emissions, with 17.8 tons in May and 14.5 tons.

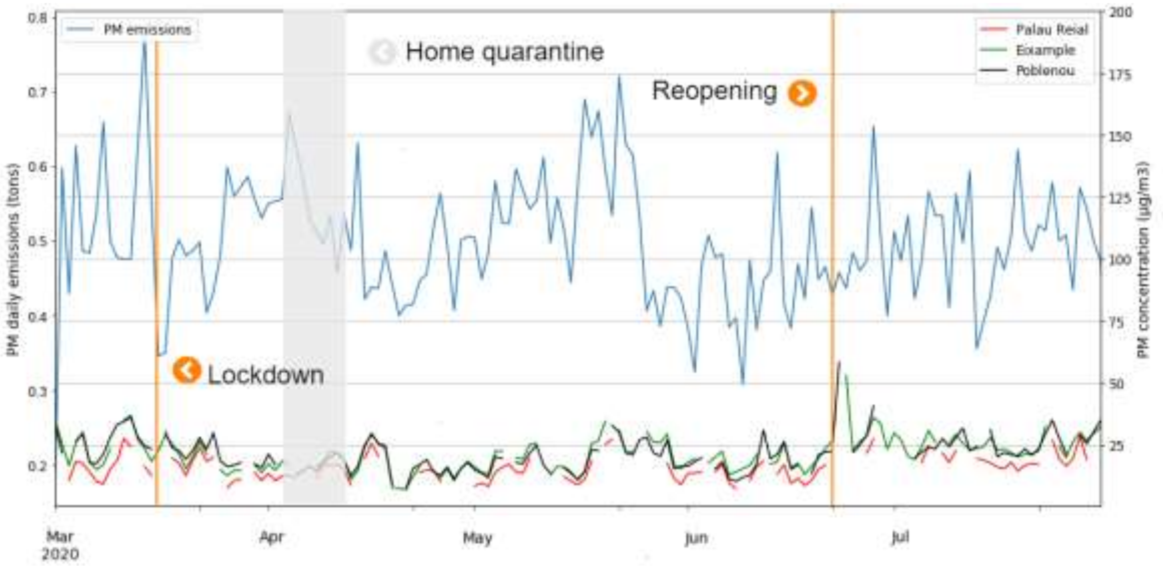

Figure 54. Daily changes in PM emissions and concentrations from March 1 to July 31, 2020

In Figure 54, slightly different values were reported during the lockdown period, with special emphasis in the home-quarantine period. Variability within days was mostly noticed during the most restrictive days as well. In this scenario, PM emissions were compared to PM concentrations in Poblenou and Eixample. **Although correlation is not as clear as with SO_2 and NO_X, still most of the peaks matched with increased emissions arising from the maritime traffic.** In general, lower concentration values were common during the lockdown period, which leads again to conclude that **30 nautical miles traffic has an influence over real-time air quality in terms of PM.**

Table 53. Computed average NO_X emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	PM (tons)
Pre-lockdown	0.53 (+1.9%)
Lockdown	0.53 (+1.9%)
Home-quarantine	0.55 (+5.8%)
Post-lockdown	0.51 (-1.9%)

Table 53 shows that lockdown values were slightly above the 5-month average, with special emphasis during the home-quarantine period, when a +5.8% above the average was recorded for PM emissions. Post-lockdown values felt down below the average as in previous cases.

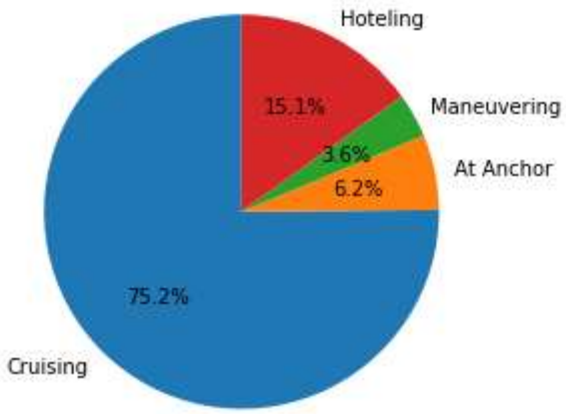

Figure 55. Distribution of PM emissions per stage

Figure 55 shows a very similar trend in distribution per phase compared to NO_X emissions. In fact, PM emissions are both fuel- and engine-related, thus explaining this characteristic distribution. The cruising stage dominated again with 75.2% contribution, compared to 15.1% related to the hoteling phase, 6.2% related to anchored vessels and 3.6% to vessels in maneuvering stage.

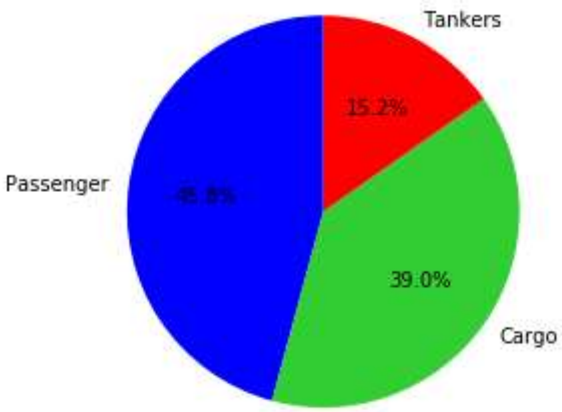

Figure 56. Distribution of PM emissions per type of ship

As in Figure 56, distribution by type of ship is quite similar to fuel-related emissions, with passenger leading the rank, accounting for 45.8% of total emissions, 39.0% related to cargo vessels and 15.2% related to tankers.

5.1.3. Modelled installed power

After assessing fuel consumption and emissions through the generic database model, this thesis also aimed to provide a mathematical equation able to estimate the installed power based solely on AIS data, when comprehensive ship technical particulars are not available.

Equations and fitting

Separate equations and fittings are provided for each of the studied ships: cargo, tankers and passenger. The installed power on cargo vessels is dependent on ship length and breadth, whereas length is the only variable for tankers and passengers.

Cargo vessels

The fitting of the 528-cargo vessel database installed power v. length and breadth resulted in equation (eq.22), with a $R^2 = 0.92$, adjusted $R^2 = 0.92$ and RMSE = 5369. The model upper and lower limitations were listed in section 3.2.2, Table 28.

$$P_{cargo}(B, L) = -1203 - 0.000077091 \cdot B^{4.9} + 0.03408829 \cdot L^{2.5} \qquad \text{(eq.22)}$$

Where:

P_{cargo} : Installed engine power (kW);

B : Vessel breadth (m); and

L : Vessel length (m).

As shown in Figure 57, the model predicted with high accuracy low installed powers, whereas the error incremented as the real installed power did.

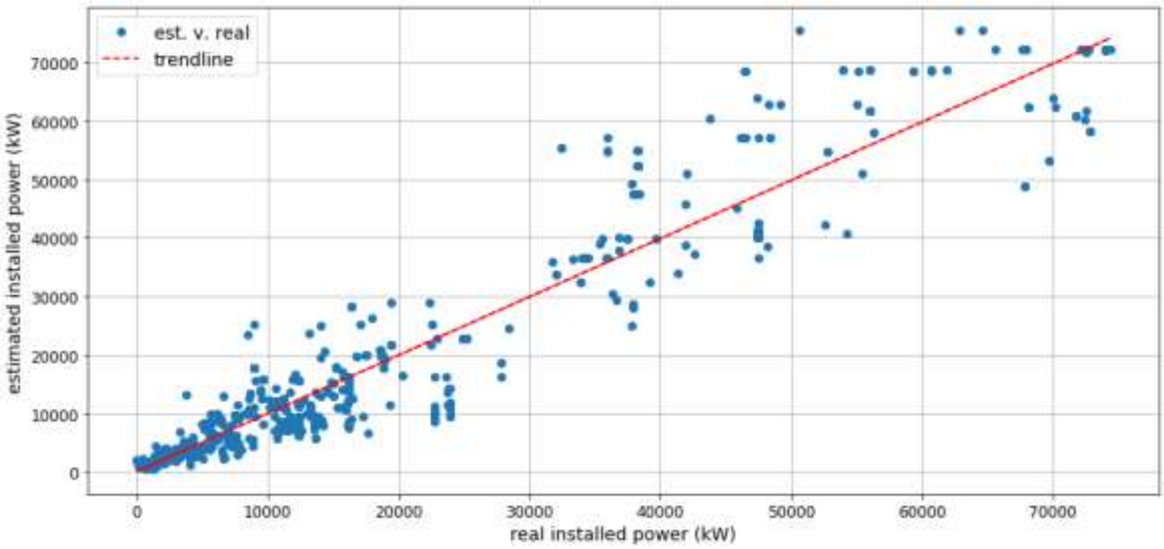

Figure 57. Estimated v. real installed power (kW) for cargo vessels

Tanker vessels

The fitting of the 528-cargo vessel database installed power v. length resulted in equation (eq.23), with a R^2 = 0.90, adjusted R^2 = 0.90 and RMSE = 5756. The resulting equation was expanded so as to increase accuracy from R^2 = 0.85 to the current value, also achieving a lower RMSE from previous 6744. The model upper and lower limitations were listed in section 3.2.2, Table 28.

$$P_{tankers}\,(L) = 3671.10566147 + 0.36347426 \cdot L^{2.15} - 7{,}5869 \cdot 10^{-38} \cdot L^{16} - 0.18208 \cdot \ln(L)^{7} \quad \text{(eq.23)}$$

Where:

$P_{tankers}$: Installed engine power (kW); and

L : Vessel length (m).

In Figure 58, as in the previous case, the fitting is more accurate for low installed-power vessels rather than for vessels with higher outputs. This is also expected, as the number of tanker and cargo vessels with high outputs are limited compared to the amount of small vessels (79).

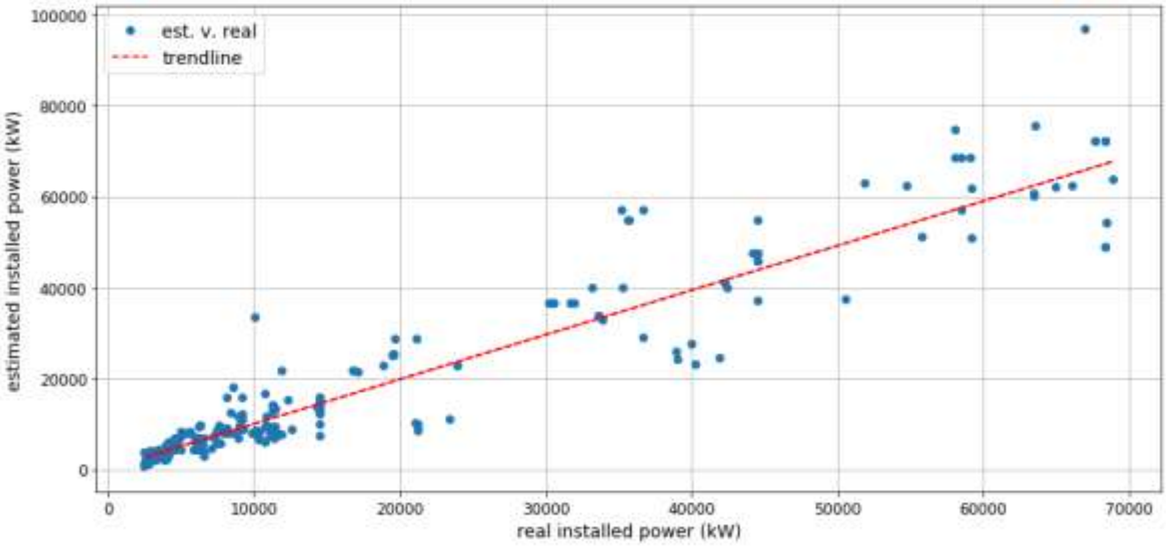

Figure 58. Estimated v. real installed power (kW) for tanker vessels

Passenger vessels

The fitting of the 112-passenger vessel database installed power v. length resulted in equation (eq.24), with a R^2 = 0.91, adjusted R^2 = 0.92 and RMSE = 6452. The model upper and lower limitations were listed in section 3.2.2, Table 28.

$$P_{passenger}\,(L) = -5353.6 + 1.65640834 \cdot L^{1.85} \qquad \text{(eq.24)}$$

Where:

$P_{passenger}$: Installed engine power (kW); and

L : Vessel length (m).

As seen in Figure 59, the estimated power on passenger vessels showed a higher deviation for lower values than the others. This value was expected, as the installed power on passenger vessels is mostly dependent on the number of passengers, cabins and services available for guests (70). However, the fitting was still considered acceptable, given the statistical results.

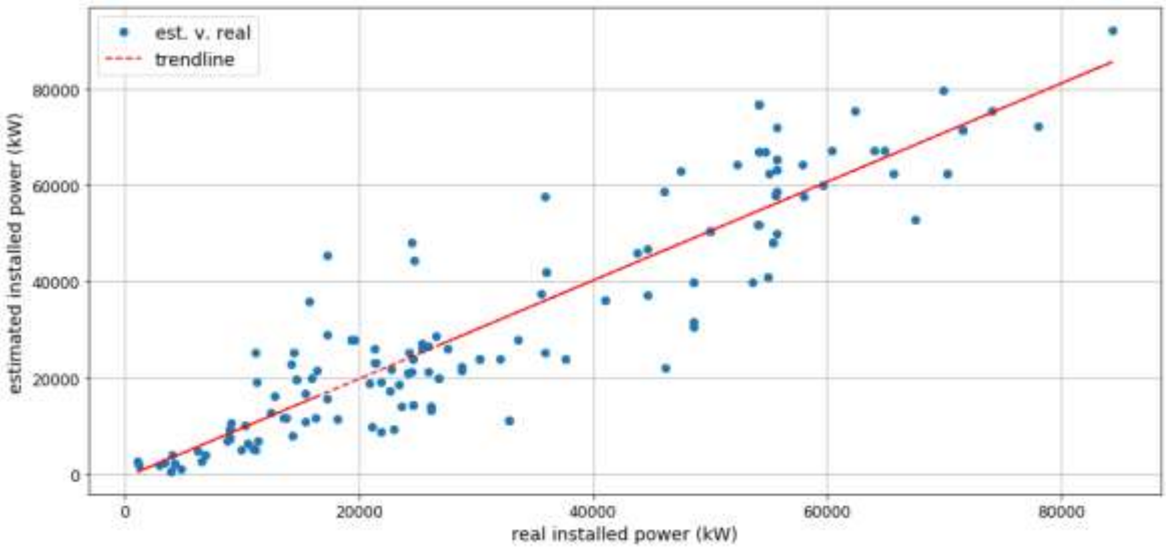

Figure 59. Estimated v. real installed power (kW) for passenger vessels

Calibration

As fuel consumption is a major parameter used in most inventories to calculate emissions (36)(67), it was selected as an indicator to calibrate the mathematical model, bearing in mind the special considerations in section 3.2.2, when generating emission inventories. As the purpose of the model was to obtain a generic approach, global results were preferred over more detailed ones.

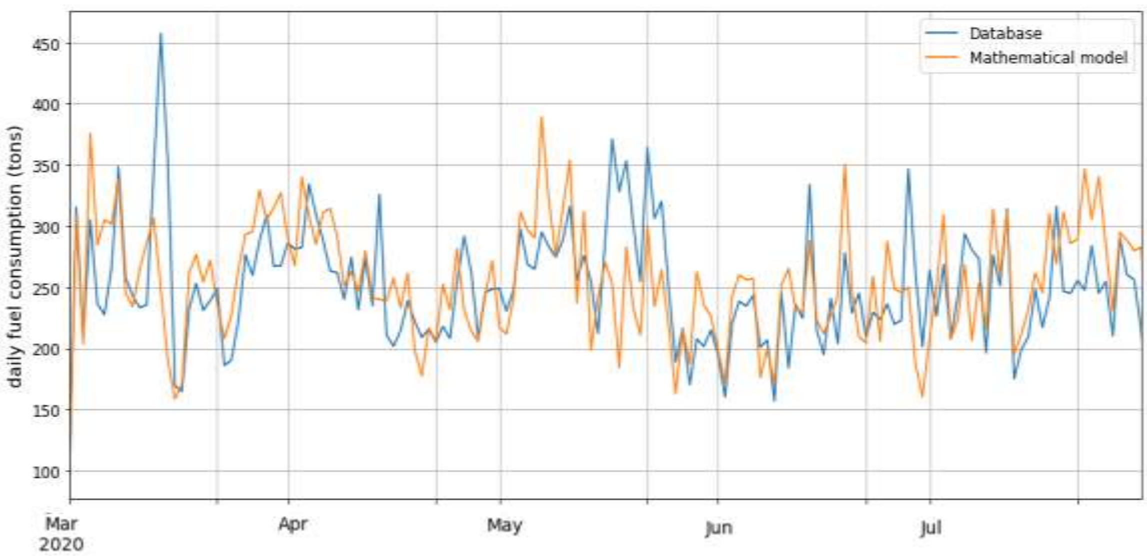

Figure 60. Daily fuel consumption over 5-months period

The database model returned a total fuel consumption of 40421 tons, compared to a 38604 tons computed through the mathematical model, which means a relative error of 4.5%.

Figure 60 shows the daily evolution of fuel consumption, comparing both values. In general terms, both of them followed very similar trends, with daily averages of 263 tons through the database and 251 tons through the mathematical model. Difference in the peaks can be explained through different engine configurations related to the model limitations, as auxiliary engines, which have higher consumptions, have been given preference over main engines during the hoteling and at anchor stages.

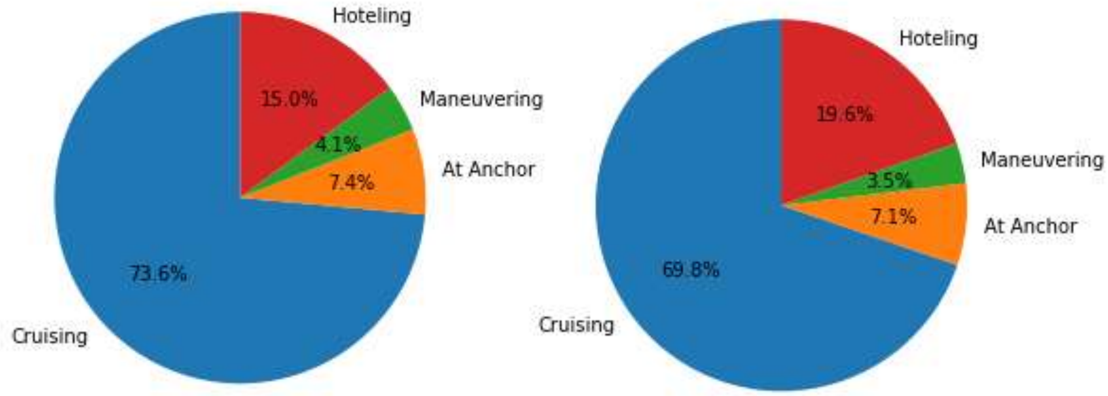

Figure 61. Distribution of fuel consumption per stage, computed through the database (left) and through the mathematical model (right)

In Figure 61, slightly different distributions per stage are obtained. The mathematical model tends to overestimate hoteling emissions, to underestimated values during cruising and maneuvering stages. This is related to the model limitation in terms of auxiliary engine power, as the database model neglects them when they are not available.

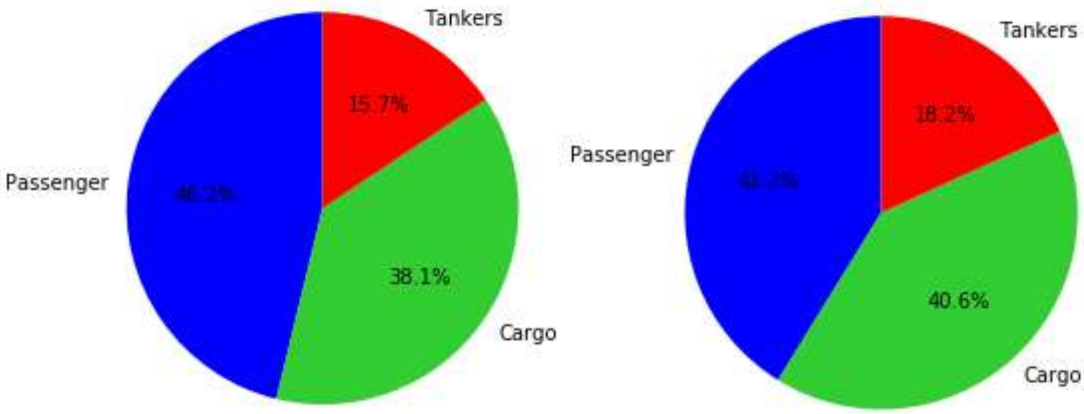

Figure 62. Distribution of fuel consumption per type of ship, computed through the database (left) and through the mathematical model (right)

Figure 62 shows also a discrepancy between both models, as passenger-related consumption is underestimated through the mathematical model, to higher values related to cargo and tanker vessels. Again, these differences are mostly related to the engine configuration issues previously reported.

Table 54. Total air pollutant emissions over 5-months period
Values in brackets are the relative errors related to the database model

Pollutant	Database model (tons)	Mathematical model (tons)
CO_2 (tons)	108603	105055 (ϵ 3.3%)
SO_2 (tons)	403	417 (ϵ 3.5%)
NO_X (tons)	2309	2068 (ϵ 11.7%)
PM (tons)	81	71 (ϵ 14.1%)

In terms of emissions, as shown in Table 54, and SO_2 attained relative errors well below 5% comparing the mathematical model to the database one. CO_2 emissions followed exactly the same path as the fuel consumption, with a similar reported relative error. Higher SO_2 emissions, in spite of lower fuel consumption through the mathematical model are driven by the fact that LNG-powered vessels, with lower SO_2 emission factors were neglected from the model.

However, relative errors slightly over 10% were obtained for NO_X and PM emissions, both of them reporting lower values through the mathematical model. This was driven by several reasons. On the one side, NO_X levels are heavily dependent on engine speed, therefore even if averaged engine revolutions were taken into account, different emission factors apply for lower- and upper-end speeds, which were not considered in the model. On the other side, PM levels were lower due to the model limitations when applying engine configurations, as all berthed and anchored vessels were considered to use only auxiliary engines in the mathematical-based model, whereas in the database model, main engine power was used when auxiliary engines were not available.

The mathematical model can therefore be used to generate low-detailed global emission inventories, when enhanced data on vessel particulars might not be available, and **aiming to find general emission and fuel consumption magnitudes rather than specific figures**. It can also be used to discuss the relative impact of each of the stages and type of ship to total fuel consumption and generic emission values.

For further guidance, results of sections 5.1.1 and 5.1.2 comparing those calculated with data straight from the database and through the mathematical model are presented in Annex A4.

5.2. Discussion

Worth noting that maritime traffic within 30 nautical miles range during 5 months, poured on average the same CO_2 as 27,151 VW Polo covering 25,000km per year or the same NO_X as 144,313 VW Polo covering the same distance per year. Figures are quite high, given the limited spatial and temporal area covered, which reinforces the idea that **vessels are extremely polluting machines**.

Table 55. Fuel consumption, CO_2 and SO_2 emissions change during lockdown and home-quarantine

Period	Fuel consumption	CO_2 emissions	SO_2 emissions
Lockdown	+1.1%	+1.8%	+3.8%
Home-quarantine	+4.6%	+6.9%	+7.7%

Along the lockdown days, **fuel consumption and emissions did increase above averaged values** for the whole 5-month period, with special emphasis during the home-quarantine time. In general terms, Table 55 shows that fuel consumption went up by +1.1% during the lockdown, resulting in average higher emissions of CO_2 by +1.8% and SO_2 by +3.8%. During the home-quarantine period, the increase in fuel consumption was even higher, scoring +4.6% above the average, resulting in +6.9% and +7.7% above the average emissions of CO_2 and SO_2.

Recovering Table 40 and Table 41 in section 4.2, vessels in the 30 nautical miles range went up by +1.8% and +20.3% above the average during lockdown and home-quarantine periods, respectively. Although there is a relationship between an increased fuel consumption and emissions and an increased number of vessels in vicinity, several other factors play an important role in final values. A clear example is the mismatch in higher number of vessels and the associated fuel consumption and emissions during the home-quarantine period. The issue can be traced back to Table 41, as the average speed of the vessels went down by -6.9%, so did the status *Underway,* to an increased number of vessels *At Anchor* and *Moored.* Hence**, the change in operating mode also had an impact on global values, as emissions during the home-quarantine did not increase in as much as the number of vessels in the area.**

Table 56. NO_X and PM emissions change during lockdown and home-quarantine

Period	NO_X emissions	PM emissions
Lockdown	+1.2%	+1.9%
Home-quarantine	+7.3%	+5.8%

Moving onto non-fuel-related emissions, the increase during lockdown was again related to a higher number of vessels in vicinity, as shown in Table 56. As with fuel-related emissions, similar reasoning applies to home-quarantine figures, given that the increase in emissions was only 25% of the increase in number of vessels. Nonetheless, the different operating mode at that time also resulted in lower-than-expected emissions. In fact, NO_X emissions were slightly lower for auxiliary engines than main engines, given that they mostly fall into Tier II, whereas PM emissions were lower in average due to lower engine load factors, arising from reduced speeds.

There is a straight correlation between vessels operating mode and their overall impact on air quality, as vessels trading at reduced pace are in fact more environmentally friendly.

Table 57. Average distribution of fuel consumption and emissions per type of ship

Item	Cargo	Tankers	Passenger
Fuel consumption	38.1%	15.7%	46.2%
CO_2 emissions	38.7%	16.5%	44.2%
SO_2 emissions	41.0%	15.3%	43.6%
NO_X emissions	42.2%	14.9%	42.7%
PM emissions	39.0%	15.2%	45.8%

Table 57 lists an interesting fact, as the contribution of each type of ship did not match at all the distribution of vessels in the 30 nautical miles range during the 5 month period. Surprisingly, passenger vessels, which only represented an average of 17.2% peaked in fuel consumption and in all of the emissions, with a contribution well above 40%. On the other side, cargo vessels represented almost half of the total traffic, and accounted for around 40% of total emissions and fuel consumption. On the lower end, tankers represented 33.2% of total traffic but accounted for around 15% of emissions and fuel consumption. This scenario is clearly related to the fact that passenger vessels, operate at high loads constantly and trade at very high speeds, as seen in Table 35 and Table 36. Their shipboard services have high power demand which explains their higher installed power. It is then clear that, even during reduced trades, they are a major pollutant in the vicinity of Barcelona.

Figure 48, Figure 51 and Figure 54 reveal that the concentration of SO_2, NO_X and PM follow quite similar trends as the maritime traffic. This impact can be seen all along the 5-month period, with special emphasis on SO_2 and NO_X concentrations during the lockdown period. The reduced industrial activity and wheeled traffic at the time leads to the theory that those values were mostly related to maritime activity. It also allows to easily picture that not only vessels berthed and at anchor around Barcelona have an impact on the city air quality, but also traffic as far as 30 nautical miles from the city.

In fact, recovering Figure 14 and Table 17 in section 2.3.1 compared to Figure 51 in section 5.2, a reduction in NO_X emission levels in the last week of March resulted in overall reduced NO_2 concentrations at sea.

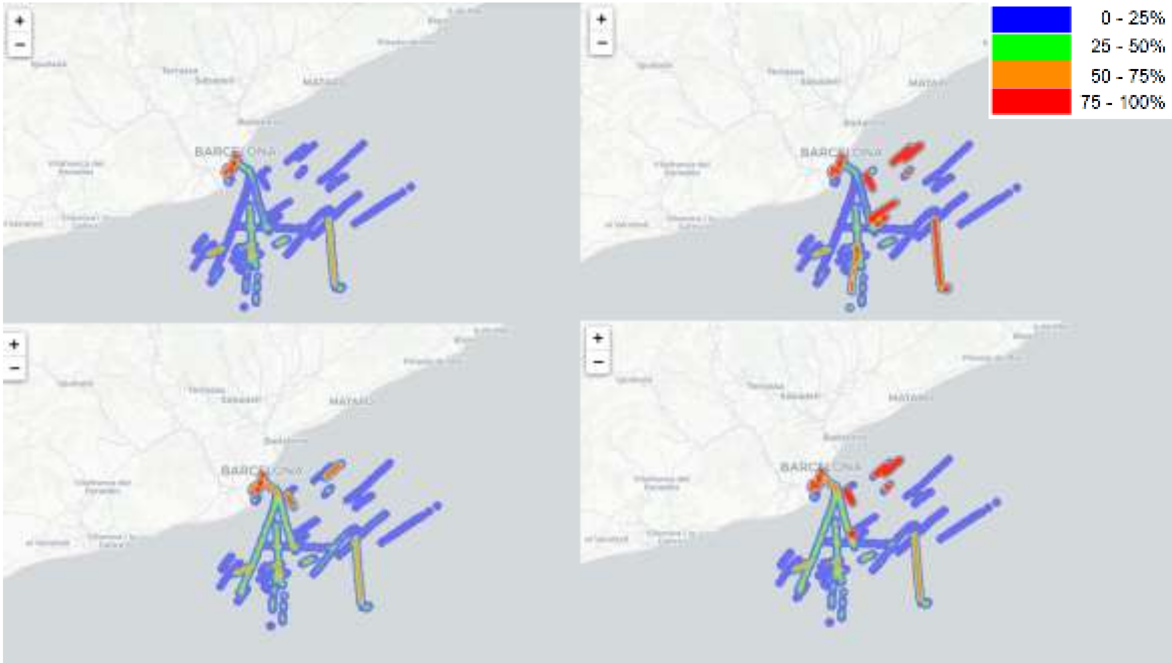

Figure 63. Emissions over 1-hour period on April 2 at 19:00 UTC, of CO_2 (upper left), SO_2 (upper right), NO_X (lower left) and PM (lower left); concentration values are based on daily averages

Colors represent the distribution of minimum and maximum emissions per minute, where blue is 0 to 25% (minimum) and red is 75 to 100% (maximum)

Supporting this theory, Figure 63, shows the emissions over 1-hour period on April 2, 2020 at 19:00 UTC; which was previously discussed as one of the days with the largest number of vessels in the vicinity. Note that emissions of all four pollutants are quite high.

Nonetheless, Figure 64, shows the emissions over a 1-hour period on March 14, 2020 at 19:00 UTC, which peaked as top day in terms of emissions of all four pollutants. Note that actually, even if less vessels were in the vicinity at that time, emissions per ship were higher, as vessels operated at averaged higher speeds.

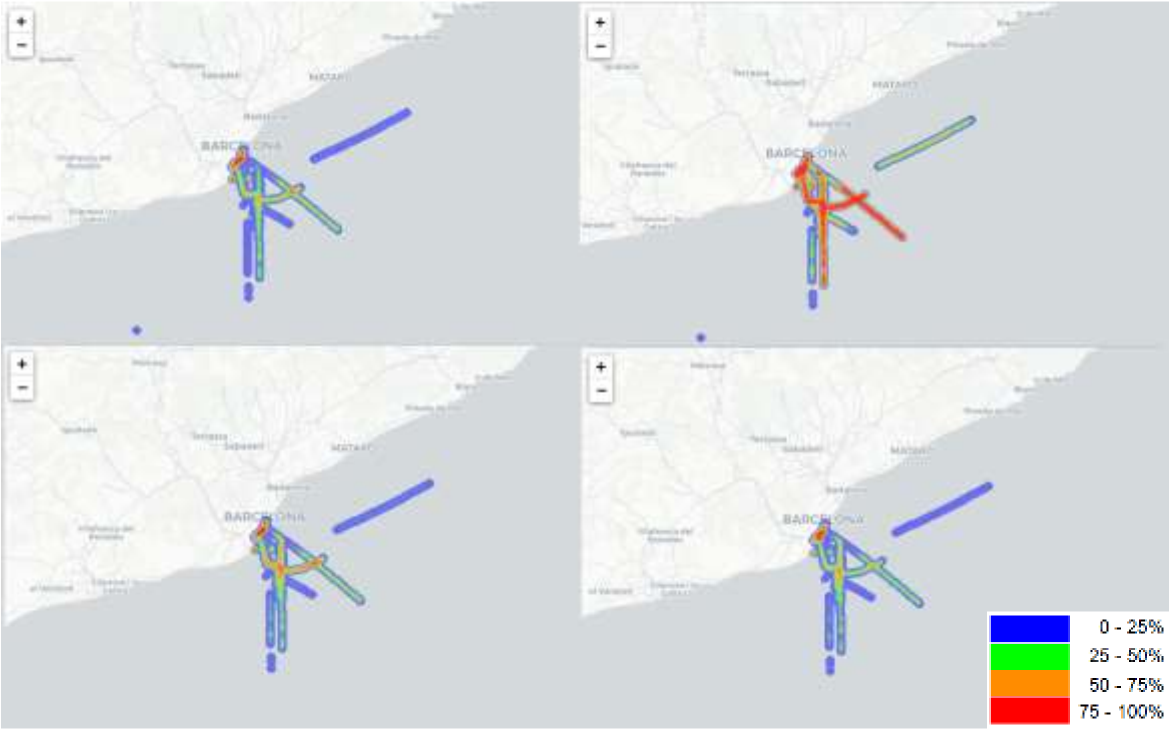

Figure 64. Emissions over 1-hour period on March 14 at 19:00 UTC, of CO_2 (upper left), SO_2 (upper right), NO_X (lower left) and PM (lower left); concentration values are based on daily averages

Colors represent the distribution of minimum and maximum emissions per minute, where blue is 0 to 25% (minimum) and red is 75 to 100% (maximum)

Concluding, emissions in general were above the 5-month average during the pandemic period, following a similar trend than the changes in traffic within the 30 nautical miles range. However, **changes in the way vessels operated, drove emissions not to grow further than the variation in the number of vessels, with special emphasis during early lockdown days, given the uncertainties at the time.** This standard is in accordance with the fact that a greener shipping can only be accomplished by reducing average vessel speeds and of course rethinking the way **passenger vessels** operate, as they **were responsible for almost half of the emissions when they just represented 17.2% of the traffic in the area.**

It is certain that **air quality in Barcelona is heavily influenced by maritime traffic, especially in terms of SO_2 and NO_X,** which accounted responsible for almost all of the pollutant concentration peak during the pandemic, as daily changes followed exactly the same trends as daily emissions from vessels during the lockdown days, with special emphasis on SO_2 concentrations which even in the post-lockdown time are heavily influenced by maritime traffic.

Chapter 6. Conclusions

The **COVID-19 pandemic did have an impact on maritime traffic and related emissions, due to changes in the way vessels trade**. Pre- and post-lockdown values, in both traffic and emissions, show a clear mismatch, as recovery from the initial downfall was not the same for all three traffics.

Regarding maritime traffic in the 30 nautical miles range, **the pandemic brought a slight increase of +1.8% in total number of reported vessels** in the area, with special emphasis during the early lockdown days and the home-quarantine period. However, **this change did not match an increase in the number of calls in Barcelona, which went down by -8.8%, but a change in the way vessels operated; owing to reduced average speeds by -2.8% and increased number of vessels *At Anchor*, *Moored* and *Not Under Command*.** It is not that more vessels were reported, but the ones that were already in range stayed over for longer periods.

Cargo and tanker vessels did manage to weather the situation by adjusting capacity to real-time demand, and show early signs of recovery as of July 31, 2020. However, **passenger vessels succumbed badly to the travel restrictions related to the pandemic**, and although ferry traffic was resumed upon reopening was in force, the ongoing uncertainties related to a still spreading virus do not forecast a smooth second semester for the business.

Concerning **fuel consumption, fuel-related and engine-related emission values were on average higher during the lockdown owing to an increased number of vessels in the area**, as fuel consumption increased by +1.1%; CO_2, NO_X and PM by +1.8%, +1.3% and +1.9%, respectively; and SO_2 by +3.8%. **This slightly higher increase in SO_2 was related to a lower consumption of LNG, driven by higher use of auxiliary engines while moored and anchored**. This result actually matches findings in Table 17, as SO_2 did increase during the pandemic. Worth noting that the increase in fuel consumption and emissions during early lockdown and home-quarantine was well below the increase in number of vessels, driven by reduced speeds and increased number of moored and anchored vessels. Yet another point supporting the fact that changes in the operation of vessels were in force at the time.

The **distribution of types of ship in range did not match the distribution of fuel consumption and emissions**. In fact, **passenger vessels were surprisingly responsible for more than 40% of the consumption and the emissions, but represented only 17.2%**. The reason is higher installed outputs arising from higher power demands to sustain shipboard services and higher-than-average trading speeds. This raises the current ongoing discussion of sustainability of passenger business other than ferry crossings.

With reference to the **mathematical model** developed to estimate installed power based solely on AIS data, it **is quite a useful tool to be used when access to comprehensive technical databases is limited**, as was initially the case in this project. Further work is required to refine the algorithm and correct the issues regarding NO_X and PM emissions. However, it is a feasible method to be employed to craft generic and low-detailed emission inventories.

About the relationship between emissions and air quality in Barcelona, concluding results have been found for SO_2, NO_X and PM concentrations, as **the magnitude of their peaks and valleys are consistent with emissions resulting from shipping activities within the 30nm range**. In fact, **the relation between SO_2 and NO_X ship-related emissions and their concentrations across the city is a critical problem** which shall be further assessed through proper atmospheric dispersion modelling, as in light of the results quantitative values are necessary.

This leads to raise concern over the fact that **an ECA might be established in the Mediterranean Sea and that NO_X stricter tiers shall also be enforced in Europe**, as air pollution is a major issue, to be promptly addressed real measures.

Final thoughts are for ship operators and their crews, as it has been proved that **reducing operating speeds might be crucial to improve vessel performance in terms of fuel consumption and emissions**. Vessels are provided with several state-of-the-art tools to improve and optimize fuel consumption, which are not always being properly used. It is no secret within the industry that in order to comply with line schedules, Masters order full ahead speeds to later reduce power well below optimum engine loads, leading to higher fuel consumptions per trip as economical speeds are never observed. Figures are overwhelming, as a single minute of shipping in the 30nm range means 3 VW Polo running 25,000km in terms of CO_2 and 40 VW Polo running 25,000km in terms of NO_X. **Lower speeds mean lower consumption and emissions, and contribute to a greener world. It is necessary to rethink that reducing the global pace in terms of trade and economy, might contribute to a more sustainable living.**

Further work in the scope of the matter is required, as this project was just a preliminary analysis of both the COVID-19 effects over maritime traffic and the related emissions. Long-term effects of the pandemic shall be assessed more in deep, with all regards. It is also a breakthrough to rethink our world and what society wants to leave for future generations.

References

1. World Health Organization (WHO). Novel Coronavirus - China. In: *World Health Organization*. [online]. 2020. [retrieved on: 2 june 2020]. Available at: https://www.who.int/csr/don/12-january-2020-novel-coronavirus-china/en/

2. World Health Organization (WHO). Novel SARS-CoV-2 outbreak. In: *World Health Organization* [online]. 2020. [retrieved on: 3 june 2020]. Available at: https://www.who.int/westernpacific/emergencies/covid-19.

3. World Health Organization (WHO). WHO Director-General's opening remarks at the media briefing on COVID-19. In: *WHO Director General's speeches*. 2020, no. 3, p. 4.

4. Board Editorial. Here Comes the Coronavirus Pandemics. In: *The New York Times* [online]. 2020. [retrieved on: 14 june 2020]. Available at: https://www.nytimes.com/2020/02/29/opinion/sunday/corona-virus-usa.html.

5. John Hopkins University (JHU). COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU). In *ArcGIS* [online]. 2020. [retrieved on: 1 august 2020]. Available at: https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6.

6. Gita, G. The Great Lockdown: Worst Economic Downturn Since the Great Depression. In: *International Monetary Fund Blog* [online]. 2020. [retrieved on: 4 july 2020]. Available at: https://blogs.imf.org/2020/04/14/the-great-lockdown-worst-economic-downturn-since-the-great-depression/#:~:text=The%20Great%20Lockdown%3A%20Worst%20Economic%20Downturn%20Since%20the%20Great%20Depression,-April%2014%2C%202020&text=As%20countries%20implement%20necessary%20quarantines,put%20in%20a%20Great%20Lockdown.

7. European Transport Safety Council (ETSC). COVID-19: Huge drop in traffic in Europe, but impact on road deaths unclear. In: *ETSC Journal* [online]. 2020. [retrieved on: 7 july 2020]. Available at: https://etsc.eu/covid-19-huge-drop-in-traffic-in-europe-but-impact-on-road-deaths-unclear/.

8. Editorial. Coronavirus unemployment at nearly 15% is still shy of the record high reached during the Great Depression. In: *The Conversation* [online]. [retrieved on: 20 june 2020]. Available at: https://theconversation.com/coronavirus-unemployment-at-nearly-15-is-still-shy-of-the-record-high-reached-during-the-great-depression-138237.

9. Tagliapietra, S. COVID-19 is causing the collapse of oil markets: when will they recover? In: *Bruegel* [online]. 2020. [retrieved on: 9 june 2020]. Available at: https://www.bruegel.org/2020/04/covid-19-is-causing-the-collapse-of-oil-markets-when-will-they-recover/.

10. Union for the Mediterranean. Impacts of the COVID-19 pandemic on Ports and Maritime Transport in the Mediterranean Region. In: UFM Secretariat Reports [online]. 2020. [retrieved on: 30 june 2020]. Available at: https://ufmsecretariat.org/impacts-covid-ports-maritime-transport-mediterranean/.

11. Kituyi, M. Coronavirus: Ships moving, ports open and cross-border trade flowing. In: *UNCATD News* [online]. 2020. [retrieved on: 15 june 2020]. Available at: https://unctad.org/en/pages/newsdetails.aspx?OriginalVersionID=2311.

12. United Nations Conference on Trade and Development (UNCATD). *Review of Maritime Transport 2019*. Geneva, Switzerland: 2019. ISBN 9789211129588.

13. Tobías, A. et al. Changes in air quality during the lockdown in Barcelona (Spain) one month into the SARS-CoV-2 epidemic. In: *Science of the Total Environment*. Elsevier B.V., 2020, vol. 726, p. 138540. ISSN 18791026. DOI 10.1016/j.scitotenv.2020.138540.

14. Presidency of Spain. La Moncloa. Mejora sin precedentes en los niveles de calidad del aire de las ciudades españolas debido a la limitación de desplazamientos y de actividad. In: La Moncloa Servicio de Prensa [online]. [retrieved on: 19 july 2020]. Available at: https://www.lamoncloa.gob.es/serviciosdeprensa/notasprensa/transicion-ecologica/Paginas/2020/170720-calidad_aire.aspx.

15. European Union. Climate Action. Reducing emissions from the shipping sector. In: EU Climate Action [online]. [retrieved on: 5 june 2020]. Available at: https://ec.europa.eu/clima/policies/transport/shipping_en.

16. European Maritime Safety Agency (EMSA). COVID-19: Coronavirus Outbreak – Impact on Shipping. 2020, no. 7.

17. Instituto Marítimo Español. *Economía del sector marítimo*. 4. Madrid, Spain: Fondo de Ingeniería Naval - Instituto Marítimo Español. 2016. ISBN 9788481987904.

18. Port de Barcelona. Presentació de resultats 1er semestre 2020. Barcelona, Spain: Port de Barcelona. 2020.

19. Organization of Petroleum Exporter Countries (OPEC). OPEC Monthly Oil Market Report. Vienna, Austria: OPEC Monthly Reports. 2020. ISSN 0003-7028. DOI 10.1366/000370205775142601.

20. Editorial. The impact of Covid-19 on global shipping. In: Ship Technology [online]. [retrieved on: 4 june 2020]. Available at: https://www.ship-technology.com/features/impact-of-covid-19-on-shipping/.

21. Editorial. COVID-19: Impact on Global Shipping and China's Economy. In: *The Diplomant* [online]. [retrieved on: 24 july 2020]. Available at: https://thediplomat.com/2020/03/covid-19-impact-on-global-shipping-and-chinas-economy/.

22. Editorial. Carnival Corporation to Emerge Leaner and Stronger. In: Cruise Industry News [online]. [retrieved on: 4 august 2020]. Available at: https://www.cruiseindustrynews.com/cruise-news/23239-carnival-corporation-to-emerge-leaner-and-stronger.html.

23. Editorial. Coronavirus: Norway bans passengers from large cruise ships. In: *The Sydney Morning Herald* [online]. [retrieved on: 4 august 2020]. Available at: https://www.smh.com.au/world/europe/norway-bans-passengers-from-large-cruise-ships-spain-hails-covid-app-20200804-p55i7p.html.

24. Gonzalez-Aregall, M. y Bergqvist, R. *Green port initiatives for a more sustainable port-city interaction: The case study of Barcelona*. 2020. ISBN 9780128191347. DOI 10.1016/b978-0-12-819134-7.00007-1.

25. Editorial. Germany and Italy recessions as European economy slumpsr. In: Business Insider [online]. [retrieved on: 1 july 2020]. Available at: https://www.businessinsider.com/germany-and-italy-recessions-as-european-economy-slumps-2019-1?IR=T.

26. Organisation for Economic Co-operation and Development (OECD). GDP Growth - First quarter of 2020. In: OECD Newsroom [online]. [retrieved on: 3 july 2020]. Available at: http://www.oecd.org/newsroom/gdp-growth-first-quarter-2020-oecd.htm.

27. Nieto Guarasa, J. Study on the impact of cruise ships calling at Barcelona in the city air quality. Barcelona, Spain: Barcelona School of Nautical Studies - UPC BarcelonaTECH. 2017

28. Editorial. The Shape of the COVID-19 Recession. In: Forbes [online]. [retrieved on: 7 july 2020]. Available at: https://www.forbes.com/advisor/investing/covid-19-coronavirus-recession-shape/.

29. Editorial. El Gobierno ha planteado a Nissan que posponga el cierre de sus plantas de Barcelona. In: *La Vanguardia* [online]. [retrieved on: 24 july 2020]. Available at: https://www.lavanguardia.com/economia/20200708/482187408443/gobierno-nissan-cierre-planta-barcelona-2022.html.

30. International Maritime Organisation (IMO). Air Pollution. In: *IMO Our Work* [online]. [retrieved on: 5 july 2020]. Available at: http://www.imo.org/en/OurWork/environment/pollutionprevention/airpollution/pages/air-pollution.aspx.

31. European Maritime Safety Agency (EMSA). Air Emissions. In: *EMSA Maritime Pollution* [online]. [retrieved on: 24 june 2020]. Available at: http://www.emsa.europa.eu/main/air-pollution.html.

32. European Maritime Safety Agency (EMSA). Air pollution. In: *EMSA Maritime Pollution* [online]. [retrieved on: 24 june 2020]. Available at: http://www.emsa.europa.eu/main/air-pollution/air-pollution.html.

33. Mjelde, A. et al. Environmental accounting for Arctic shipping - A framework building on ship tracking data from satellites. In: *Marine Pollution Bulletin*. Elsevier Ltd, 2014, vol. 87, no. 1, p. 22-28. ISSN 18793363. DOI 10.1016/j.marpolbul.2014.07.013..

34. Knak, C. *Diesel motor ships engines and machinery text*. 1. Copenhagen, Denmark: GAD Publishers, 1979. ISBN 8712467758.

35. Woud, H.K. y Stapersma, D. *Design of propulsion and electric power generation systems*. 13. London, United Kingdom: IMarEST, 2019. ISBN 9781856098496.

36. European Environmental Agency. International maritime and inland navigation, national navigation, national fishing, recreational boats. In: *EMEP / EEA air pollutant emission inventory guidebook 2019*. 1. Luxembourg: Publication office of the European Union, 2019, p. 1-37. ISBN 9789292138066.

37. Ford, M. *A Master's Guide To Using Fuel Oil Onboard Ships*. London, United Kingdom: The Standard Club, 2012.

38. Salazar, F. *Internal Combustion Engines*. South Bend, United States: University of Notre Dame, 1998.

39. Wright, A. *Exhaust emissions from combustion machinery*. 5. London, United Kingdom: Institute of Marine Engineers, 2000. ISBN 1902536177.

40. Kristensen, H.O. Energy demand and exhaust Gas emissions of marine engines. in: *Clean Shipping Currents*. 2015, vol. 1, no. 2014, p. 18-26.

41. Hajjar, Y. Gaseous pollutatns formation and their harmful effects on health and environment. En: *Innovative Energy Policies*. 2011, vol. 1, no. 1, p. 31-39.

42. United States of America. Environment Protection Agency. Overview of Greenhouse Gas (GHG) Emissions. In: EPA Online [online]. [retrieved on: 15 july 2020]. Available at: https://www.epa.gov/ghgemissions/overview-greenhouse-gases#carbon-dioxide.

43. Flagan, R.C. and Seinfeld, J.H. *Fundamentals of air pollution engineering*. Egleswood Cliff (NJ), United States of America: Prentice Hall Inc, 1988. ISBN 0133325377.

44. European Union. European Environment Agency. Total greenhouse gas emission trends and projections in Europe. In: European Environmental Agency Data [online]. [retrieved on: 13 july 2020]. Available at: https://www.eea.europa.eu/data-and-maps/indicators/greenhouse-gas-emission-trends-6/assessment-3.

45. Pinella Corbacho, F. *Transporte marítimo y medio ambiente: un binomio reconciliable*. 1. Cadiz, Spain: Universidad de Cádiz - Servicio de Publicaciones, 1996. ISBN 8477862885.

46. International Maritime Organisation (IMO). Sulphur 2020 – cutting sulphur oxide emissions. In: IMO Media Centre [online]. [retrieved on: 24 june 2020]. Available at: http://www.imo.org/en/MediaCentre/HotTopics/Pages/Sulphur-2020.aspx.

47. European Parliament and Council of Europe. Directive 2008/50/EC, of 21 May 2008 *on ambient air quality and cleaner air for Europe*.

48. European Parliament and Council of Europe. Directive 2005/33/EC, of 6 July 2005 *as regards the sulphur content of marine fuels.*

49. European Parliament and Council of Europe. Directive 2012/33/EU, of 21 November 2012 *as regards the sulphur content of marine fuels.*

50. European Parliament and Council of Europe. Regulation 2015/757/EU, of 29 April 2015 *on the monitoring, reporting and verification of carbon dioxide emissions from maritime transport, and amending Directive 2009/16/E*C.

51. International Maritime Organisation (IMO). *International Convention for the Prevention of Pollution from Ships, 1973 as modified by the Protocol of 1978 .* London, United Kingdom. 2012

52. International Maritime Organization (IMO). Maritime Environmental Protection Committee. *MEPC.212(63). 2012 Guidelines on the method of calculation of the attained energy efficiency ship design index (EEDI) for new ships*. London, United Kingdom. 2012

53. International Maritime Organization (IMO). Maritime Environmental Protection Committee. *MEPC1/Circ.684. Guidelines for Voluntary use of the Ship Energy Efficiency Operational Indicator (EEOI).* London, United Kingdom. 2009.

54. International Maritime Organisation (IMO). Maritime Environmental Protection Committee. *MEPC.282(70). Guidelines for the development of a ship energy efficiency management plan (SEEMP).* London, United Kingdom. 2016

55. International Maritime Organization (IMO). Study shows significant reductions in CO2 emissions from ships from IMO measures. In: IMO Press Briefings [online]. 2014. Available at: http://www.imo.org/en/ModerateCentre/PressBriefings/Pages/57-EEDIstudy.aspx#.Vjtkf_kvfDc.

56. Ajuntament de Barcelona. The energy, climate change and air quality plan of Barcelona (PECQ 2011-2020). Barcelona, Spain: Servei de Publicació de l'Ajuntament de Barcelona. 2011, p. 1-178.

57. Consorci Sanitari de Barcelona. Informe d'avaluació de la qualitat de l'aire a la ciutat de Barcelona. In: *Butlletí de Salut Pública* 2013, no. 2, p. 206.

58. De Melo Rodríguez, G. et al. Evaluating air emission inventories and indicators from ferry vessels at ports. En: *Safety of Sea Transportation - Proceedings of the International Conference on Marine Navigation and Safety of Sea Transportation, TRANSNAV 2017*. 2017, p. 111-118. DOI 10.1201/9781315099088-19.

59. Querol, X. Are the Reductions in Air Pollution Triggered by the COVID-19 Epidemic Having Health Effects? In: *ISGLOBAL* [online]. [retrieved on: 8 july 2020]. Available at: https://www.isglobal.org/en/healthisglobal/-/custom-blog-portlet/-sera-relevante-para-nuestra-salud-la-disminucion-de-la-contaminacion-atmosferica-durante-la-epidemia-de-la-covid-19-/5083982/11101.

60. Intenrational Maritime Organisation (IMO). Automatic Identification Systems (AIS). In: IMO Our Work [online]. [retrieved on: 17 july 2020]. Available at: http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx.

61. Del Fante Serra, J. *GMDSS - Manual del Operador General*. 3. Barcelona, Spain: DelfCasini, 2016.

62. International Maritime Organisation (IMO). A.1106(29) Guidelines for the onboard operational use of ship borne Automatic Identification Systems (AIS). London, United Kingdom. IMO Guidelines. 2015.

63. International Telecommunication Union. Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile band (Recommendation ITU-R M.1371-4). In: *International Journal of Rock Mechanics and Mining Science*. 2005, vol. 42, no. 4, p. 481-507.

64. Hudson D. et al. Telecommunications Engineer's Reference Book. Oxford, United Kingdom: Butterworth-Heinemann. 1993. ISBN 9780750611626. DOI 10.1016/B978-0-7506-1162-6.50005-8

65. Calculate distance and bearing between two points using haversine formula in JavaScript. In: Movable Type Maths [online]. [retrieved on: 17 july 2020]. Available at: https://www.movable-type.co.uk/scripts/latlong.html.

66. Goldsworthy, L. and Goldsworthy, B. Modelling of ship engine exhaust emissions in ports and extensive coastal waters based on terrestrial AIS data - An Australian case study. In: *Environmental Modelling and Software*. Elsevier Ltd, 2015, vol. 63, p. 45-60. ISSN 13648152. DOI 10.1016/j.envsoft.2014.09.009.

67. Coello, J. et al. An AIS-based approach to calculate atmospheric emissions from the UK fishing fleet. In: *Atmospheric Environment*. Elsevier Ltd, 2015, vol. 114, p. 1-7. ISSN 18732844. DOI 10.1016/j.atmosenv.2015.05.011.

68. Sun, X. et al. Estimation of vessel emissions inventory in Qingdao port based on big data analysis. En: *Symmetry*. 2018, vol. 10, no. 10, p. 1-11. ISSN 20738994. DOI 10.3390/sym10100452.

69. Jalkanen, J.-P. et al. A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area. In: *Atmospheric Chemistry and Physics Discussions*. 2009, vol. 9, no. 4, p. 15339-15373. ISSN 1680-7375. DOI 10.5194/acpd-9-15339-2009.

70. Jalkanen, J.P. et al. Extension of an assessment model of ship traffic exhaust emissions for particulate matter and carbon monoxide. In: *Atmospheric Chemistry and Physics*. 2012, vol. 12, no. 5, p. 2641-2659. ISSN 16807316. DOI 10.5194/acp-12-2641-2012.

71. Johansson, L., Jalkanen, J.P. and Kukkonen, J. Global assessment of shipping emissions in 2015 on a high spatial and temporal resolution. In: *Atmospheric Environment*. Elsevier Ltd, 2017, vol. 167, p. 403-415. ISSN 18732844. DOI 10.1016/j.atmosenv.2017.08.042.

72. Liu, T.K., Chen, Y.S. y Chen, Y.T. Utilization of vessel automatic identification system (AIS) to estimate the emission of air pollutant from merchant vessels in the port of kaohsiung. In: *Aerosol and Air Quality Research*. 2019, vol. 19, no. 10, p. 2341-2351. ISSN 20711409. DOI 10.4209/aaqr.2019.07.0355.

73. Cepowski, T. Regression Formulas for the Estimation of Engine Total Power for Tankers, Container Ships and Bulk Carriers on the Basis of Cargo Capacity and Design Speed. In: *Polish Maritime Research*. 2019, vol. 26, no. 1, p. 82-94. ISSN 20837429. DOI 10.2478/pomr-2019-0010.

74. International Maritime Organisation (IMO). *International Convention on the Safety of Life at Sea.* London, United Kingdom. 2012

75. Saputra, H. et al. Estimation of exhaust ship emission from marine traffic in the straits of Singapore and batam waterways using automatic identification system (AIS) data. In: *Jurnal Teknologi*. 2015, vol. 77, no. 23, p. 47-53. ISSN 01279696. DOI 10.11113/jt.v77.6687.

76. Alessandrini, A. et al. Estimation of shipping emissions using vessel long range identification and tracking data. En: *Journal of Maps*. Taylor & Francis, 2017, vol. 13, no. 2, p. 946-954. ISSN 17445647. DOI 10.1080/17445647.2017.1411842.

77. Zhang, Y. et al. Inland port vessel emissions inventory based on Ship Traffic Emission Assessment Model-Automatic Identification System. In: *Advances in Mechanical Engineering*. 2017, vol. 9, no. 7, p. 1-9. ISSN 16878140. DOI 10.1177/1687814017712878.

78. Perez, H.M. et al. Automatic Identification Systems (AIS) Data Use in Marine Vessel Emission Estimation. In: *Time*. 2009, no. 4, p. 1-25.

79. Abramowski, T., Cepowski, T. y Zvolenský, P. Determination of Regression Formulas for Key Design Characteristics of Container Ships at Preliminary Design Stage. In: *New Trends in Production Engineering*. 2018, vol. 1, no. 1, p. 247-257. DOI 10.2478/ntpe-2018-0031.

80. Instituto de Salud Carlos III. Distribución geográfica del COVID-19 en España. In: Centro Nacional de Epidemiología [online]. 2020. [retrieved on: 31 july 2020]. Available at: https://cnecovid.isciii.es/covid19/

81. Port de Barcelona. Estadístiques de tràfic del Port de Barcelona: dades acumulades a desembre de 2019. Barcelona, Spain: 2019.

Annex A1. Code for maritime and port traffic analysis

This section contains all code used in the maritime and port traffic analysis section. The code is set to be run using the *Jupyter Notebook* (*Anaconda* 3) with the following libraries:

i. *Pandas* for data analysis;
ii. *Numpy* for matrix and array analsysis;
iii. *Math* for performing mathematical operations;
iv. *Os* for directory changes;
v. *Matplotlib* for data plotting;
vi. *Datetime* for time arrangement; and
vii. *Folium* for live maps.

The below notebooks are enclosed within this annex:

i. AISdata.py – reads and plots all vessels in range –;
ii. AISdata_map.py – reads and plots in a semi-live map all vessels in range –;
iii. AISdata_status.py – reads and assesses the change of status for all vessels in range –;
iv. AIScalls.py – reads all vessels and analyses the number of calls in Barcelona –;
v. AISspeed.py – reads and assesses the change in speed for all vessels in range –; and
vi. AISdraft.py – reads and assesses the change in draft for all vessels in range.

The code is set to be run, with decoded AIS messages 1, 2, 3 and 5 of Class A. All steps are further detailed within the code.

Prior to using the code, a notebook named *modMeu.py* has to be created by the reader introducing the code given in section A.1.1., which includes the 30nm and port limit filter functions.

A1.1. Code for modMeu.py

```
#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def join_DynStat(dyn, stat):

    res = pd.merge(dyn, stat, on='mmsi', how='inner', suffixes=('_m123',
'_m5'))

    return res


#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def apb_lim(data,r):        #filters lat-lon boundaries of the Port of
Barcelona based on circle equation r in nautical miles

    R = 3440.06             # radius of the earth in nautical miles

    fnaut = np.radians([2.184643,41.38247])  #longitude, latitude facultat
de nàutica in radians

    data            =            data.loc[((R**2*(((np.radians(data.lon)-
fnaut[0])*np.cos((np.radians(data.lat)+fnaut[1])/2))**2+(np.radians(data
.lat)- fnaut[1])**2))<= (r**2))]

    return data


#%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

def inport(lat,lon):

    R = 3440.06

    r = 4

    b = np.radians([2.092140,41.353021])

    value                  =                  ((R**2*(((np.radians(lon)-
b[0])*np.cos((np.radians(lat)+b[1])/2))**2+(np.radians(lat)-b[1])**2)))

    value = value <= r**2

    return value
```

A1.2. Code for AISdata.py

AIS vessels in range

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
July 14, 2020

```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway','At anchor','NUC','Restricted maneuv.','Constr
ained','Moored','Aground','Sailing']
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter
    t = pd.read_csv('ClassA_clean.csv', sep = ",", usecols = ['date','mm
si','lat','lon','status'])
    t = apb_lim(t, r)
    t = t.drop(columns = ['lat','lon'])
    m = np.unique(t.mmsi)
```

```
    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname','shiptype','to_
bow','to_stern','to_port', 'to_starboard','draught','destination']
    aux = aux.drop(columns = ['type','IMO','shipname','to_bow','to_stern
','to_port','to_starboard','draught', 'destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    d = np.unique(aux.date)

    # Dataframe appending
    s = s.append(aux, ignore_index = True)
    del(m, t, d, aux)

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date']=s.date.dt.round('1h')
s['group']=pd.cut(s.shiptype, 3, right=False, labels=labelship)
sf = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(sf))
sf.head()
```

Hourly count of ships in range

Total

```
# Hourly count of ships in range (plot)
## This section shows the total number of vessels in range per hour alon
g a 5-month time period

fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date']).count()['shiptype'].plot(ax=ax, color = 'k').legend
(['Total'], fontsize = 14)
plt.ylabel('ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
```

```
ax.set_ylim([10,55])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

Per shiptype

```
# Hourly count of ships in range by shiptype (plot)
## This section shows the total number of vessels grouped by shiptype in
range per hour along a 5-month time period

fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','group']).count().fillna(0)['shiptype'].unstack().plo
t(ax=ax, color = ['#0000FF','#32CD32','#FF0000'])
sf.groupby(['date']).count()['shiptype'].plot(ax=ax, color= 'k').legend(
labelship+['Total'], fontsize = 14)
plt.ylabel('ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,55])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

A1.3. Code for AISdata_map.py

AIS vessel map

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
July 25, 2020

```
import pandas as pd
import numpy as np
import datetime as dt
import folium
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates
from folium.plugins import TimestampedGeoJson

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelColor = ['#0000FF','#32CD32','#FF0000']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverabili
ty', 'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']

# Map boundaries function
## This function generates an OpenStreetMap Map centered at the Port de
Barcelona

def generateBaseMap(default_location=[41.382472, 2.205039], default_zoom
_start=12):
    base_map = folium.Map(location=default_location, control_scale=True,
zoom_start=default_zoom_start,tiles='cartodbpositron', width=640, height
=480)
    return base_map

# Live map function
## This function transforms the database into points to be plotted in a
dynamic map

def create_geojson_features(s):
```

```
    features = []

    for _, row in s.iterrows():
        feature = {
            'type': 'Feature',
            'geometry': {
                'type':'Point',
                'coordinates':[row['lon'],row['lat']]
            },
            'properties': {
                'time': pd.to_datetime(row['date']).__str__(),
                'popup':'name: '+row['Name'].__str__()+'<br>'+'speed: '+
row['speed'].__str__()+' knots'+'<br>'+'status: '+row['status'].__str__(
),
                'style': {'color' : ''},
                'icon': 'circle',
                'iconstyle':{
                    'fillColor': row['fillColor'],
                    'fillOpacity': 0.8,
                    'radius': 5
                }
            }
        }
        features.append(feature)
    return features
```

Data loading and filtering

Data loading

```
# Data Loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter
    t = pd.read_csv('ClassA_clean.csv', sep = ",", usecols = ['date','mm
si','lat','lon','status','speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    m = np.unique(t.mmsi)

    # Static data Loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname','shiptype','to_
```

```
bow','to_stern','to_port','to_starboard','draught','destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
,'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

print("data length:", len(s))
s.head()

## This section reads the vessel technical database and assigns the name
to each vessel

p = pd.read_excel('Particulars.xlsx')
p = p.drop(columns = ['ENG_KW','GT', 'Fuel','SFC','Disp','Built','AUX_KW
','service_speed','rpm'])
s = s.merge(p, how = 'left', on = ['IMO','IMO'])
del(p)
s = s.drop(columns = ['IMO'])

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1min basis

s.date = pd.to_datetime(s.date, format = '%Y%m%d%H%M%S')
s['date'] = s['date'].dt.round('1min')
s['month'] = s.date.apply(lambda x: x.month)
s['day'] = s.date.apply(lambda x: x.day)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset based on the co
rresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s['fillColor'] = pd.cut(s.shiptype, 3, right=False, labels = labelColor)
s.status.loc[s.status == 0] = labelstatus[0]
s.status.loc[s.status == 1] = labelstatus[1]
s.status.loc[s.status == 2] = labelstatus[2]
s.status.loc[s.status == 3] = labelstatus[3]
s.status.loc[s.status == 4] = labelstatus[4]
s.status.loc[s.status == 5] = labelstatus[5]
s.status.loc[s.status == 6] = labelstatus[6]
s.status.loc[s.status == 8] = labelstatus[7]
s.status.loc[s.status == 10] = labelstatus[8]
s.status.loc[s.status == 11] = labelstatus[9]
s.status.loc[s.status == 15] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

First dataset rearrangement

```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## This overcomes the fact that these vessels, due to their status, only
transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minute
s=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

Second dataset rearrangement

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 2nd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot
```

```
dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

Third dataset rearrangement

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 3rd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli3_df))
dupli3_df.head()
```

Data appending

```
# Final data appending
## This section appends the rearranged dynamic data for vessels with sta
tus "Moored" and "At Anchor" with all other vessels
## The final data length is much larger than the initial one, as it guar
antees a dataset available per ship every 1min

s = pd.concat([s,dupli3_df], ignore_index = True)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
del(dupli3_df)

print("data length:", len(s))
s.head()
```

Live map

```
# Select month (in numbers) and range (in numbers) - maximum 5 days

month = 3
start_day = 1
end_day = 3
```

Live map

```
# Live map
## This section generates the live map plotting all vessels, their info
and postn. in 1min time intervals

s_copy = s.loc[(s.day >= start_day) & (s.day <= end_day) & (s.month == m
onth)]
start_geojson = create_geojson_features(s_copy)
base_map = generateBaseMap()
TimestampedGeoJson(start_geojson, period = 'PT1M', add_last_point=True,
duration = 'PT59S', transition_time = 0.0000001, max_speed = 100, auto_p
lay = True).add_to(base_map)
base_map
```

A1.4. Code for AISdata_status.py

AIS status

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
June 28, 2020

```
import pandas as pd
import numpy as np
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat
```

```
folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006', 'C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Moored']
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status'])
    t= apb_lim(t, r)
    t = t.drop(columns = ['lat', 'lon'])
    m = np.unique(t.mmsi)
```

```
    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname','shiptype','to_
bow','to_stern','to_port','to_starboard','draught','destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
,'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m, t, aux)

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1h')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset and drops
status other than "Underway", "At Anchor", "NUC" or "Moored"

s['group'] = pd.cut(s.shiptype, 3, right=False, labels=labelship)
s = s.loc[(s.status != 3) & (s.status != 4) & (s.status != 6) &
(s.status != 7) & (s.status != 8) & (s.status != 15)]
s.status.loc[s.status == 5] = 3
s['vsl_status'] = pd.cut(s.status, 4, right=False, labels=labelstatus)

print("data length:", len(s))
s.head()
```

Hourly status of ships in range

Total

```
# Hourly status of ships in range
## This section shows the hourly evolution of status of vessels in range
along a 5-month time period

fig, ax = plt.subplots(figsize=(15,7))
s.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).p
lot(ax=ax)
s.groupby(['date']).size().plot(ax=ax, color=
'k').legend(labelstatus+['Total'], fontsize = 12)
plt.ylabel('ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,55])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

Cargo

```
# Hourly status of cargo ships in range
## This section shows the hourly evolution of status of cargo vessels in
range along a 5-month time period

sf = s.loc[s.group == 'Cargo']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color=
'k').legend(labelstatus+['Total Cargo'], fontsize = 12)
plt.ylabel('cargo ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,35])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

Tankers

```
# Hourly status of tanker ships in range
## This section shows the hourly evolution of status of tanker vessels
in range along a 5-month time period

sf = s.loc[s.group == 'Tankers']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color=
'k').legend(labelstatus+['Total Tankers'], fontsize = 12)
plt.ylabel('tanker ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,25])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

Passenger

```
# Hourly status of passenger ships in range
## This section shows the hourly evolution of status of passenger
vessels in range along a 5-month time period

sf = s.loc[s.group == 'Passenger']
fig, ax = plt.subplots(figsize=(15,7))
sf.groupby(['date','vsl_status']).count()['status'].unstack().fillna(0).
plot(ax=ax)
sf.groupby(['date']).size().plot(ax=ax, color=
'k').legend(labelstatus+['Total Passenger'], fontsize = 12)
plt.ylabel('passenger ships in range', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,15])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

A1.5. Code for AIScalls.py

AIS calls filter

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
July 18, 2020

```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
months = ['March', 'April', 'May', 'June', 'July']
```

Data loading and filtering

Data loading

```
# Data Loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('ClassA_clean.csv', sep = ",", usecols = ['date','mm
si','lat','lon','status'])
    t = apb_lim(t,r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
/F) value to the question "Is the vessel in port?"
```

```
    t = t.drop(columns = ['lat','lon'])
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname','shiptype','to_
bow','to_stern','to_port','to_starboard','draught','destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
,'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m, t, aux)

print("data length:", len(s))
s.head()

## This section reads the vessel technical database and assigns the name
to each vessel

p = pd.read_excel('Particulars.xlsx')
p = p.drop(columns = ['ENG_KW','GT', 'Fuel','SFC','Disp','Built','AUX_KW
','service_speed','rpm'])
s = s.merge(p, how = 'left', on = ['IMO','IMO'])
del(p)
s = s.drop(columns = ['IMO'])

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1h')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
```

```
print("data length:", len(s))
s.head()
```

Vessel cals

Call filter

```
# Call filter
## This section groups data into bunches of in-port and out-port dataset
s per vessel
## A vessel call is considered as a bool value change from in-port = Fal
se to in-port = True

df = s
df['mmsi2'] = df.mmsi
df = df.groupby(['mmsi2'])
m = np.unique(s.mmsi)
calls = pd.DataFrame()
for i in range(0,len(df)):
    k = df.get_group(m[i])
    k['groupno'] = k.inport.diff().cumsum().fillna(0)
    result = k.groupby(['groupno']).agg(['first'])
    result.inport = result.inport.astype(int)
    result = result.loc[(result.inport['first'] == 1)]
    calls = calls.append(result, ignore_index = True)
    del(k, result)

del(df, m, i)
calls = calls.stack().reset_index().drop(columns = ['level_0', 'level_1'
, 'inport'])

print("data length:", len(calls))
calls.head()
```

Call refining

```
# Call refining
## This section drops all calls dated 2020-03-01 00:00:00, so as to prev
ent previous days calls from being counted for March 1

calls = calls.loc[calls.date != '2020-03-01 00:00:00']
calls = calls.loc[calls.mmsi != 224022660]
calls = calls.loc[calls.mmsi != 224022650]
calls = calls.loc[calls.mmsi != 224334000]
calls['date'] = calls.date.dt.round('1d')
calls = calls.loc[calls.status != 1]

print("data length:", len(calls))
calls.head()
```

Daily calls

Figures - total

```
# Total figures by day
## This section groups and counts the number of calls per day along a 5-
month period of time

byday = calls.groupby(['date']).size().reset_index(name = 'number_of_cal
ls')

print("data length:", len(byday))
byday.head()
```

Plot - total

```
# Total figures by day (plot)
## This section shows the evolution in the number of calls per day along
a 5-month period of time

fig, ax = plt.subplots(figsize=(15,7))
calls.groupby(['date']).count()['shiptype'].plot(ax=ax, color = 'k').leg
end(['Total'], fontsize = 12)
plt.ylabel('daily calls', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([5,35])
ax.tick_params(labelsize=14)
plt.grid()
plt.show()
```

Figures - per shiptype

```
# Total figures by day and shiptype
## This section groups and counts the number of calls per day and shipty
pe along a 5-month period of time

calls['group'] = pd.cut(calls.shiptype, 3, right=False, labels=labelship
)
bydayngroup = calls.groupby(['date','group']).size().to_frame('number_of
_calls')
bydayngroup.head()
```

Plot - per shiptype

```
# Total figures by day and shiptype (plot)
## This section shows the evolution in the number of calls per day and s
hiptype along a 5-month period of time

calls['group'] = pd.cut(calls.shiptype, 3, right=False, labels=labelship
```

```
)
fig, ax = plt.subplots(figsize=(15,7))
calls.groupby(['date','group']).count().fillna(0)['shiptype'].unstack().
plot(ax=ax, color = ['#0000FF','#32CD32','#FF0000'])
calls.groupby(['date']).count()['shiptype'].plot(ax=ax, color= 'k').lege
nd(labelship+['total'], fontsize = 12)
plt.ylabel('daily calls', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([0,35])
ax.tick_params(labelsize=14)
plt.grid()
```

Calls per ship

```
# Total calls by ship
## This section groups and counts the number of calls per ship along a 5
-month period of time

byship = calls.groupby(['Name']).size().reset_index(name = 'number_of_ca
lls')
byship = byship.sort_values(by = ['number_of_calls'], ascending = False
)

print("total number of ships:", len(byship))
byship.head()
```

Calls per shiptype

Figures - total

```
# Total calls by shiptype
## This section groups and counts the number of calls per shiptype along
a 5-month period of time

bygroup = calls.groupby(['group']).size().reset_index(name = 'number_of_
calls')
bygroup
```

Pie chart - total

```
# Total calls by shiptype (plot)
## This section shows in a pie chart the number of calls per shiptype al
ong a 5-month period of time

fig, ax = plt.subplots()
sizes = [bygroup.iloc[0]['number_of_calls'], bygroup.iloc[1]['number_of_
calls'], bygroup.iloc[2]['number_of_calls']]
ax.pie(sizes, labels=labelship, autopct='%1.1f%%', shadow=False, startan
gle=90, colors = ['#0000FF','#32CD32','#FF0000'])
```

```
ax.axis('equal')
plt.title('March - July 2020 calls by shiptype')
plt.show()
```

Figures - per month

```
# Total calls by month and shiptype
## This section groups and counts the number of calls per shiptype and m
onth along a 5-month period of time

calls['month'] = calls.date.apply(lambda x: x.month)
calls.month = pd.cut(calls.month, 5, right=False, labels=months)
calls = calls.drop(columns = ['date'])
bymonth = calls.groupby(['month','group']).size().reset_index(name = 'nu
mber_of_calls')
bymonth.head()
```

Pie chart - per month

```
# Total calls by month and shiptype (plot)
## This section shows in a pie chart the number of calls per shiptype an
d month along a 5-month period of time

calls.groupby(['month','group']).size().unstack(level = 0).plot.pie(subp
lots = True, startangle = 90,
figsize = (25,20), autopct='%1.1f%%', colors = ['#0000FF','#32CD32','#FF
0000'])
plt.legend(loc = 'best')
plt.show()
```

A1.6. Code for AISspeed.py

AIS speed analysis

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
July 14, 2020

```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006', 'C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status','speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
/F) value to the question "Is the vessel in port?"
```

```
    t = t.drop(columns = ['lat', 'lon'])
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date', 'type', 'mmsi','IMO','shipname','shiptype','t
o_bow','to_stern','to_port','to_starboard','draught','destination']
    aux=aux.drop(columns = ['date','IMO','type','shipname','to_bow','to_
stern','to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m, t, aux)

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1d basis

s.date = pd.to_datetime(s['date'], format='%Y%m%d%H%M%S')
s['date']=s.date.dt.round('1d')

print("data length:", len(s))
s.head()
```

In-port and status filtering

```
# In-port and status filtering
## This section filters and reutrns data located outside of port and wit
h status other than "At Anchor"

s.inport = s.inport.astype(int)
s = s.loc[s.inport == 0]
s = s.loc[s.status != 1]
```

```
print("data length:", len(s))
s.head()
```

Vessel speed

Average speed per vessel

```
# Average speed per vessel
## This section groups vessels and returns their mmsi and speeds

s = s.drop(columns = ['inport', 'status'])
s = s.groupby(['date','mmsi']).mean().reset_index()
s['group'] = pd.cut(s.shiptype, 3, right = False, labels = labelship)
s = s.drop(columns = ['mmsi', 'shiptype'])

print("data length:", len(s))
s.head()
```

Average speed per day

```
# Average speed per vessel
## This section groups vessels by date and returns the total average spe
ed per day

total = s.groupby(['date'])['speed'].mean().reset_index(name = 'avg_spee
d')
total
```

Plot - Average speed per day

```
# Average speed per vessel - Plot
## This section shows the daily avarage speed variation per day along a
5-month time period

fig, ax = plt.subplots(figsize = (15, 7))
s.groupby(['date'])['speed'].mean().plot(ax = ax, color = 'k').legend(['
Total'], fontsize = 12)
plt.ylabel('average speed in knots', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([4,16])
ax.tick_params(labelsize=14)
plt.grid()
```

Average speed per shiptype

```
# Average speed per shiptype
## This section groups all vessels by date and shiptype and returns the
average daily speed

bygroup = s.groupby(['date','group'])['speed'].mean().to_frame('avg_spee
```

```
d')
bygroup
```

Plot - Average speed per shiptype

```
# Average speed per shiptype - Plot
## This section shows the avarage speed variation per day and shiptype a
long a 5-month time period

fig, ax = plt.subplots(figsize = (15,7))
s.groupby(['date'])['speed'].mean().plot(ax=ax, color = 'k')
s.groupby(['date','group'])['speed'].mean().unstack().plot(ax=ax, color
= ['#0000FF','#32CD32','#FF0000']).legend(['Total']+labelship, fontsize
= 12)
plt.ylabel('average speed in knots', fontsize = 14)
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
ax.set_ylim([4,16])
ax.tick_params(labelsize=14)
plt.grid()
```

A1.7. Code for AISdraft.py

AIS draft study

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
July 18, 2020

```
import pandas as pd
import numpy as np
import math
import os
import random

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from sklearn.utils import shuffle
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006']
labelship = ['Passenger','Cargo','Tankers']
```

Data loading and filtering

Data loading

```
# Data Loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 50 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter
    t = pd.read_csv('ClassA_clean.csv', sep = ",", usecols = ['date', 'm
msi', 'lat', 'lon', 'status'])
    t = apb_lim(t, r)
    t = t.drop(columns = ['lat','lon','status'])
```

```
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns = ['date','type','mmsi','IMO','shipname','shiptype','to_
bow','to_stern','to_port','to_starboard','draught','destination']
    aux=aux.drop(columns = ['date','IMO','type','shipname','to_bow','to_
stern','to_port','to_starboard', 'destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset = 'mmsi', keep = "first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi', 'mmsi'])
    s = s.append(t, ignore_index = True)
    del(m, t, aux)

print("data length:", len(s))
s.head()
```

Randomization and cropping

```
# Data randomly shuffling and cropping
## This section randomly shuffles all vessels (mmsi) and crops the datab
ase to a 70% - significant data -

m = np.unique(s.mmsi)
m = shuffle(m)
m = m[0:math.ceil(0.7*len(m))]
s = s[s.mmsi.isin(m)]

print("data length:", len(m))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format='%Y%m%d%H%M%S')
s['date']=s.date.dt.round('1h')
s=s.drop_duplicates(['date','mmsi'],keep='first')
s['date']=s.date.dt.round('1d')
```

```
s['group'] = pd.cut(s.shiptype, 3, right=False, labels=labelship)
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

Draft analysis

Total - plot

```
# Draught variation plot
## This section shows the evolution of draught per vessel and time along
a 5-month time period

fig, ax = plt.subplots(figsize=(15,7))
pax = s.loc[s.group == 'Passenger']
pax.set_index('date', inplace = True)
pax.groupby(['mmsi'])['draught'].plot(ax = ax, color = '#0000FF')
cargo = s.loc[s.group == 'Cargo']
cargo.set_index('date', inplace = True)
cargo.groupby(['mmsi'])['draught'].plot(ax = ax, color = '#32CD32')
tanker = s.loc[s.group == 'Tankers']
tanker.set_index('date', inplace = True)
tanker.groupby(['mmsi'])['draught'].plot(ax = ax, color = '#FF0000')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-06-30'))
ax.legend(labelship)
ax.set_ylim([3,15])
plt.ylabel('draught')
plt.title('Draught variation')
plt.grid()
```

Cargo - histogram

```
# Draught variation histogram - Cargo
## This section shows an histogram of draught per vessel in percentage a
long a 5-month time period

e = cargo.reset_index()
e = e.groupby(['mmsi'])['draught'].agg(['min','max']).reset_index()
e['rel'] = abs(((e['min'] - e['max'])/e['max']))
e.dropna(subset = ['rel'], inplace=True)
plt.hist(e.rel, color='#32CD32', label='Cargo', histtype = 'bar')
plt.legend()
plt.xlabel('relative draught change')
plt.ylabel('number of ships')
plt.title('draught variation histogram')
plt.show()
print("vessels not changing draught:",(len(e.loc[e.rel == 0])/len(e))*10
0,"%")
print("maximum draught variation:", max(e.rel)*100,"%")
print("average draught variation:", np.mean(e.rel)*100,"%")
```

Cargo - pie chart

```
# Draught variation pie chart - Cargo
## This section shows a pie chart of draught per vessel in percentage al
ong a 5-month time period

e.rel = round(e.rel,1)
e.groupby(['rel']).size().plot.pie(startangle = 90, autopct='%1.1f%%')
plt.legend(loc='best', bbox_to_anchor=(0.7, 0.5, 0.7, 0.5))
plt.title('draught relative variation')
plt.ylabel('Cargo')
plt.show()
```

Tankers - histogram

```
# Draught variation histogram - Tankers
## This section shows an histogram of draught per vessel in percentage a
long a 5-month time period

b = tanker.reset_index()
b = b.groupby(['mmsi'])['draught'].agg(['min','max']).reset_index()
b['rel'] = abs(((b['min'] - b['max'])/b['max']))
b.dropna(subset = ['rel'], inplace=True)
(len(b.loc[b.rel == 0])/len(b))*100
plt.hist(b.rel, color='#FF0000', label='Tankers', histtype = 'bar')
plt.legend()
plt.xlabel('relative draught change')
plt.ylabel('number of ships')
plt.title('draught variation histogram')
plt.show()
print("% of vessels not changing draught:",(len(b.loc[b.rel == 0])/len(b
))*100,"%")
print("maximum draught variation:", max(b.rel)*100,"%")
print("average draught variation:", np.mean(b.rel)*100,"%")
```

Tankers - pie chart

```
# Draught variation pie chart - Tankers
## This section shows a pie chart of draught per vessel in percentage al
ong a 5-month time period

b.rel = round(b.rel,1)
b.groupby(['rel']).size().plot.pie(startangle = 90, autopct='%1.1f%%')
plt.legend(loc='best', bbox_to_anchor=(0.7, 0.5, 0.7, 0.5))
plt.title('draught relative variation')
plt.ylabel('Tankers')
plt.show()
```

Passenger - histrogram

```
# Draught variation histogram - Passenger
## This section shows an histogram of draught per vessel in percentage a
```

```
long a 5-month time period

a = pax.reset_index()
a = a.groupby(['mmsi'])['draught'].agg(['min','max']).reset_index()
a['rel'] = abs(((a['min'] - a['max'])/a['max']))
a.dropna(subset = ['rel'], inplace=True)
plt.hist(a.rel, color='#0000FF', label='Passenger', histtype = 'bar')
plt.legend()
plt.xlabel('relative draught change')
plt.ylabel('number of ships')
plt.title('draught variation histogram')
plt.show()
print("vessels not changing draught:",(len(a.loc[a.rel == 0])/len(a))*10
0,"%")
print("maximum draught variation:", max(a.rel)*100,"%")
print("average draught variation:", np.mean(a.rel)*100,"%")
```

Passenger - pie chart

```
# Draught variation pie chart - Passenger
## This section shows a pie chart of draught per vessel in percentage al
ong a 5-month time period

a.rel = round(a.rel,1)
a.groupby(['rel']).size().plot.pie(startangle = 90, autopct='%1.1f%%')
plt.legend(loc='best', bbox_to_anchor=(0.7, 0.5, 0.7, 0.5))
plt.title('draught relative variation')
plt.ylabel('Passenger')
plt.show()
```

(This page intentionally left blank)

Annex A2. Code for emission inventory

This section contains all code used in the emission inventory section. The code is set to be run using the *Jupyter Notebook* (*Anaconda* 3) with the following libraries:

i. *Pandas* for data analysis;
ii. *Numpy* for matrix and array analsysis;
iii. *Math* for performing mathematical operations;
iv. *Os* for directory changes;
v. *Matplotlib* for data plotting;
vi. *Datetime* for time arrangement; and
vii. *Folium* for live maps.

The below notebooks are enclosed within this annex:

i. AISemissions_db.py – generates an emission inventory through the STEAM v.2 algorithm, based on a comprehensive technical database –;
ii. AISemissions_math.py – generates an emission inventory through a modified version of the STEAM v.2 algorithm, based on a mathematical model to estimate installed power –;
iii. AISemissions_map.py – plots in a semi-live map all vessels in range, their fuel consumption and emissions per minute –; and
iv. AISemissions_heatmap.py – plots in a semi-live heatmap the concentration of higher pollutants above the average –.

The code is set to be run, with decoded AIS messages 1, 2, 3 and 5 of Class A. All steps are further detailed within the code.

A2.1. Code for AISemissions_db.py

AIS emissions - method 1

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
August 18, 2020

```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverabili
ty', 'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']

# Input data for emission calculation

# Engine data
EL = 0.80                # Average design engine load on merchant ships (
Jalkanen et al, 2012)
SFOC = 200               # Average specific fuel oil consumption (g/kWh)
(Jalkanen et al, 2009)
SFOC_AE = 220            # Average specific fuel oil consumption of aux.
engines (g/kWh) (Jalkanen et al, 2012)
rpm = 500                # Average working revolutions on medium speed en
gines (Jalkanen et al, 2009)

# Fuel qualities - These values are maximum as per ISO 8217 standards /
chemistry of natural gas
SC_fuel = 0.5            # Sulfur content of Light Fuel Oil (%)
CC_fuel = 86             # Carbon content of Light Fuel Oil as per ISO 82
17 (%)
SC_diesel = 0.5          # Sulfur content of Marine Gasoil (%)
CC_diesel = 87.5         # Carbon content of Marine Gasoil as per ISO 821
```

```
(%)
SC_lng = 4e-3            # Sulfur content of LNG (%)
CC_lng = 75              # Carbon content of LNG (%)

# Element properties
m_S = 32.0655            # Molar mass of sulfur (g/mol)
m_SO2 = 64.06436         # Molar mass of sulfur dioxide (g/mol)
m_C = 12.01              # Molar mass of carbon (g/mol)
m_CO2 = 44.0886          # Molar mass of carbon dioxide (g/mol)

# Data for PM calculation
ef_ec = 0.08             # Emission factor for elementary carbon (g/kWh)
ef_oc = 0.2              # Emission factor for organic carbon (g/kWh)
ef_ash = 0.06            # Emission factor for ashes (g/kWh)
oc_el = 1.025            # Organic carbon related to engine load (dimensi
onless)
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status', 'speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
/F) value to the question "Is the vessel in po
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname', 'shiptype','to_
bow','to_stern', 'to_port','to_starboard','draught', 'destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
, 'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]
```

```
    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

s.speed.loc[s.speed > 40] = 0
s = s.loc[s.IMO != 0]
print("data length:", len(s))
s.head()

p = pd.read_excel('Particulars.xlsx')
p.SFC.loc[p.SFC == 0] = SFOC
p['ef_NOx'] = (44*rpm**-0.23)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm < 130)] = 17
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 130) & (p.rpm < 2000)] = (45*p
.rpm**-0.2)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 2000)] = 9.8
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm < 130)] = 14
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm >= 2000)] = 7.7
p = p.drop(columns = ['Built', 'rpm'])
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1min')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset based on the co
rresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s.status.loc[s.status == 0] = labelstatus[0]
s.status.loc[s.status == 1] = labelstatus[1]
s.status.loc[s.status == 2] = labelstatus[2]
```

```
s.status.loc[s.status == 3] = labelstatus[3]
s.status.loc[s.status == 4] = labelstatus[4]
s.status.loc[s.status == 5] = labelstatus[5]
s.status.loc[s.status == 6] = labelstatus[6]
s.status.loc[s.status == 8] = labelstatus[7]
s.status.loc[s.status == 10] = labelstatus[8]
s.status.loc[s.status == 11] = labelstatus[9]
s.status.loc[s.status == 15] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

First dataset rearrangement

```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## This overcomes the fact that these vessels, due to their status, only
transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minute
s=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

Second dataset rearrangement

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 2nd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

Third dataset rearrangement

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 3rd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli3_df))
dupli3_df.head()
```

Data appending

```
# Final data appending
## This section appends the rearranged dynamic data for vessels with sta
tus "Moored" and "At Anchor" with all other vessels
## The final data length is much larger than the initial one, as it guar
antees a dataset available per ship every 1min

s = pd.concat([s,dupli3_df], ignore_index = True)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
del(dupli3_df)

print("data length:", len(s))
s.head()
```

Emission inventory

Phase filtering

```
# Phase filtering
## This section separates the database into the 4 stages: "Cruising", "A
t Anchor", "Maneuvering" and "Hoteling"

s['AE'] = 0.6
s.inport = s.inport.astype(int)
s['SC'] = SC_fuel/100
s['CC'] = CC_fuel/100
sf_in = s.loc[s.inport == 1]
sf_hotelling = sf_in.loc[sf_in.speed <= 0.5]
sf_maneuvering = sf_in.loc[sf_in.speed > 0.5]
```

```
sf_out = s.loc[s.inport == 0]
sf_anchor = sf_out.loc[sf_out.speed <= 1.5]
sf_cruising = sf_out.loc[sf_out.speed > 1.5]
```

Crusing emissions

```
# Cruising emissions
## This section computes the emissions of vessels in the "cruising" stag
e per minute
## Main Engine loads are computed through the Propeller Law, whereas 60%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_cruising = sf_cruising.merge(p, how = 'left', on = ['IMO','IMO'])
sf_cruising = sf_cruising.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_cruising.SC.loc[sf_cruising.Fuel == 'LNG'] = SC_lng/100
sf_cruising.CC.loc[sf_cruising.Fuel == 'LNG'] = CC_lng/100
sf_cruising.AE.loc[sf_cruising.group == 'Passenger'] = 0.8

sf_cruising['k'] = EL*sf_cruising.ENG_KW/((sf_cruising.service_speed*185
2/3600)**3)
sf_cruising['trans_KW'] = sf_cruising.k*(sf_cruising.speed*1852/3600)**3
sf_cruising['SFOC'] = sf_cruising.SFC*(0.455*(EL*(sf_cruising.speed/sf_c
ruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.se
rvice_speed)**3)+1.28)
sf_cruising['SFOC_AE'] = SFOC_AE*(0.455*(sf_cruising.AE)**2-0.17*(sf_cru
ising.AE)+1.28)
sf_cruising['FC'] = (sf_cruising.trans_KW*sf_cruising.SFOC*(1/60) + sf_c
ruising.AE*sf_cruising.AUX_KW*sf_cruising.SFOC_AE*(1/60))*1e-6
sf_cruising['SO2'] = ((sf_cruising.SFOC*sf_cruising.SC/m_S)*m_SO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*SC_diesel/10
0/m_S)*m_SO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['CO2'] = ((sf_cruising.SFOC*sf_cruising.CC/m_C)*m_CO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*CC_diesel/10
0/m_C)*m_CO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['NOx'] = (sf_cruising.ef_NOx*sf_cruising.trans_KW*(1/60) + s
f_cruising.AE*(45*rpm**-0.2)*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['PM'] = (sf_cruising.trans_KW*(0.455*(EL*(sf_cruising.speed/
sf_cruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruisin
g.service_speed)**3)+1.28)*((0.312*sf_cruising.SC)+(0.244*sf_cruising.SC
)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_cruising.AE*sf_cruising.AUX_KW*(
0.455*sf_cruising.AE**2-0.17*sf_cruising.AE+1.28)*((0.312*SC_diesel/100)
+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_cruising = sf_cruising.drop(columns = ['inport', 'AE', 'Fuel', 'SC',
'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'k'
, 'trans_KW', 'ef_NOx'])
```

Maneuvering emissions

```
# Maneuvering emissions
## This section computes the emissions of vessels in the "maneuvering" s
```

```
tage per minute
## Main Engine loads are computed through the Propeller Law, whereas 70%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## All engines are considered to burn MGO

sf_maneuvering = sf_maneuvering.merge(p, how = 'left', on = ['IMO','IMO'
])
sf_maneuvering = sf_maneuvering.drop(columns = ['IMO', 'mmsi', 'GT', 'Di
sp'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_maneuvering.AE.loc[sf_maneuvering.group == 'Passenger'] = 0.8
sf_maneuvering.AE.loc[sf_maneuvering.group != 'Passenger'] = 0.7

sf_maneuvering['k'] = EL*sf_maneuvering.ENG_KW/((sf_maneuvering.service_
speed*1852/3600)**3)
sf_maneuvering['trans_KW'] = sf_maneuvering.k*(sf_maneuvering.speed*1852
/3600)**3
sf_maneuvering['SFOC'] = sf_maneuvering.SFC*(0.455*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed
/sf_maneuvering.service_speed)**3)+1.28)
sf_maneuvering['SFOC_AE'] = SFOC_AE*(0.455*(sf_maneuvering.AE)**2-0.17*(
sf_maneuvering.AE)+1.28)
sf_maneuvering['FC'] = (sf_maneuvering.trans_KW*sf_maneuvering.SFOC*(1/6
0) + sf_maneuvering.SFOC_AE*sf_maneuvering.AE*sf_maneuvering.AUX_KW*(1/6
0))*1e-6
sf_maneuvering['SO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.SC/m_S)*m_S
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*SC_diesel/100/m_S)*m_SO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['CO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.CC/m_C)*m_C
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*CC_diesel/100/m_C)*m_CO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['NOx'] = (sf_maneuvering.ef_NOx*sf_maneuvering.trans_KW*(
1/60) + sf_maneuvering.AE*(45*rpm**-0.2)*sf_maneuvering.AUX_KW*(1/60))*1
e-6
sf_maneuvering['PM'] = (sf_maneuvering.trans_KW*(0.455*(EL*(sf_maneuveri
ng.speed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)+1.28)*((0.312*sf_maneuvering.SC)+
(0.244*sf_maneuvering.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_maneuver
ing.AE*sf_maneuvering.AUX_KW*(0.455*sf_maneuvering.AE**2-0.17*sf_maneuve
ring.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+e
f_ec+ef_ash)*(1/60))*1e-6
sf_maneuvering = sf_maneuvering.drop(columns = ['inport', 'AE', 'Fuel',
'SC', 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed
', 'k', 'trans_KW', 'ef_NOx'])
```

At Anchor emissions

```
# At Anchor emissions
## This section computes the emissions of vessels in the "at anchor" sta
```

```
ge per minute
## Main Engine loads are estimated at 10% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_anchor = sf_anchor.merge(p, how = 'left', on = ['IMO','IMO'])
sf_anchor = sf_anchor.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_anchor.SC.loc[sf_anchor.Fuel == 'LNG'] = SC_lng/100
sf_anchor.CC.loc[sf_anchor.Fuel == 'LNG'] = CC_lng/100
sf_anchor.AE.loc[sf_anchor.group == 'Passenger'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Tankers'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Cargo'] = 0.4
sf_anchor.ENG_KW.loc[sf_anchor.AUX_KW != 0 ] = 0

sf_anchor['SFOC'] = sf_anchor.SFC*(0.455*(EL*0.1)**2-0.17*(EL*0.1)+1.28)
sf_anchor['SFOC_AE'] = SFOC_AE*(0.455*(sf_anchor.AE)**2-0.17*(sf_anchor.
AE)+1.28)
sf_anchor['FC'] = (0.1*sf_anchor.ENG_KW*sf_anchor.SFOC*(1/60) + sf_ancho
r.AE*sf_anchor.AUX_KW*sf_anchor.SFOC_AE*(1/60))*1e-6
sf_anchor['SO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.SC/m_S)*m_SO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*SC_diesel/100/m_S)*m_S
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['CO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.CC/m_C)*m_CO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*CC_diesel/100/m_C)*m_C
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['NOx'] = (0.1*sf_anchor.ef_NOx*sf_anchor.ENG_KW*(1/60) + sf_an
chor.AE*(45*rpm**-0.2)*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['PM'] = (0.1*sf_anchor.ENG_KW*(0.455*(0.1)**2-0.17*(0.1)+1.28)
*((0.312*sf_anchor.SC)+(0.244*sf_anchor.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1
/60) + sf_anchor.AE*sf_anchor.AUX_KW*(0.455*sf_anchor.AE**2-0.17*sf_anch
or.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_
ec+ef_ash)*(1/60))*1e-6
sf_anchor = sf_anchor.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC'
, 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'ef_NOx
'])
```

Hotelling emissions

```
# Hoteling emissions
## This section computes the emissions of vessels in the "hoteling" stag
e per minute
## Main Engine loads are estimated at 20% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn either MGO or LNG, whereas all au
xiliary engines burn MGO

sf_hotelling = sf_hotelling.merge(p, how = 'left', on = ['IMO','IMO'])
sf_hotelling = sf_hotelling.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp']
)
```

```
sf_hotelling.SC.loc[sf_hotelling.Fuel == 'LNG'] = SC_lng/100
sf_hotelling.CC.loc[sf_hotelling.Fuel == 'LNG'] = CC_lng/100
sf_hotelling.SC.loc[sf_hotelling.Fuel != 'LNG'] = SC_diesel/100
sf_hotelling.CC.loc[sf_hotelling.Fuel != 'LNG'] = CC_diesel/100
sf_hotelling.AE.loc[sf_hotelling.group == 'Passenger'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Tankers'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Cargo'] = 0.4
sf_hotelling.ENG_KW.loc[sf_hotelling.AUX_KW != 0 ] = 0

sf_hotelling['SFOC'] = sf_hotelling.SFC*(0.455*(EL*0.2)**2-0.17*(EL*0.2)
+1.28)
sf_hotelling['SFOC_AE'] = SFOC_AE*(0.455*(sf_hotelling.AE)**2-0.17*(sf_h
otelling.AE)+1.28)
sf_hotelling['FC'] = (0.2*sf_hotelling.ENG_KW*sf_hotelling.SFOC*(1/60) +
sf_hotelling.AE*sf_hotelling.AUX_KW*SFOC*(1/60))*1e-6
sf_hotelling['SO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.SC/m_S)*m_SO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*SC_d
iesel/100/m_S)*m_SO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['CO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.CC/m_C)*m_CO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*CC_d
iesel/100/m_C)*m_CO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['NOx'] = (0.2*sf_hotelling.ef_NOx*sf_hotelling.ENG_KW*(1/60
) + sf_hotelling.AE*(45*rpm**-0.2)*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['PM'] = (0.2*sf_hotelling.ENG_KW*(0.455*(0.2)**2-0.17*(0.2)
+1.28)*((0.312*sf_hotelling.SC)+(0.244*sf_hotelling.SC)+ef_oc*oc_el+ef_e
c+ef_ash)*(1/60) + sf_hotelling.AE*sf_hotelling.AUX_KW*(0.455*sf_hotelli
ng.AE**2-0.17*sf_hotelling.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_die
sel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_hotelling = sf_hotelling.drop(columns = ['inport', 'AE', 'Fuel', 'SC'
, 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', '
ef_NOx'])
```

Data appending

```
# Emission data appending
## This section appends all 4 inventories per stage into a single consol
idated one

e = sf_cruising.append(sf_maneuvering, ignore_index = True)
e = e.append(sf_anchor, ignore_index = True)
e = e.append(sf_hotelling, ignore_index = True)

print("data length:", len(e))
e.head()
```

Results

Emission - Total

```
# Fuel consumption and emissions
## This section computes the total fuel consumption and emissions (tons)

e['date'] = e.date.dt.round('1d')
ef = e.drop(columns = ['date','status','speed','lat','lon','group','Name
'])
ef.sum(axis = 0)
```

Emission - Day

```
# Fuel consumption and emissions per day
## This section computes the fuel consumption and emissions (tons) per d
ay

e.groupby(['date'])['FC','SO2','CO2','NOx','PM'].sum()
```

Emission - Shiptype

```
# Fuel consumption and emissions per shiptype
## This section computes the fuel consumption and emissions (tons) per s
hiptype

e.groupby(['group'])['FC','SO2','CO2','NOx','PM'].sum()
```

Emission - Ship

```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per s
hip

e.groupby(['Name'])['FC','SO2','CO2','NOx','PM'].sum().sort_values(by =
['FC'], ascending = False )
```

Plot

```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per s
hip

fig, ax = plt.subplots(figsize=(15,7))
e.groupby(['date'])['FC'].sum().plot(ax=ax).legend(['Total'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
plt.title('Hourly count of ships in range')
plt.grid()
plt.show()
```

A2.2. Code for AISemissions_math.py

AIS emissions - method 2

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
August 19, 2020

```
import pandas as pd
import numpy as np
import math
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverabili
ty', 'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']

# Input data for emission calculation

# Engine data
EL = 0.80               # Average design engine load on merchant ships (
Jalkanen et al, 2012)
SFOC = 220              # Average specific fuel oil consumption (g/kWh)
(Jalkanen et al, 2009)
SFOC_AE = 220           # Average specific fuel oil consumption of aux.
engines (g/kWh) (Jalkanen et al, 2012)
rpm = 500               # Average working revolutions on medium speed en
gines (Jalkanen et al, 2009)

# Fuel qualities - These values are maximum as per ISO 8217 standards /
chemistry of natural gas
SC_fuel = 0.5           # Sulfur content of Light Fuel Oil (%)
CC_fuel = 86            # Carbon content of Light Fuel Oil as per ISO 82
17 (%)
SC_diesel = 0.5         # Sulfur content of Marine Gasoil (%)
CC_diesel = 87.5        # Carbon content of Marine Gasoil as per ISO 821
```

```
7 (%)
SC_lng = 4e-3           # Sulfur content of LNG (%)
CC_lng = 75             # Carbon content of LNG (%)

# Element properties
m_S = 32.0655           # Molar mass of sulfur (g/mol)
m_SO2 = 64.06436        # Molar mass of sulfur dioxide (g/mol)
m_C = 12.01             # Molar mass of carbon (g/mol)
m_CO2 = 44.0886         # Molar mass of carbon dioxide (g/mol)

# Data for PM calculation
ef_ec = 0.08            # Emission factor for elementary carbon (g/kWh)
ef_oc = 0.2             # Emission factor for organic carbon (g/kWh)
ef_ash = 0.06           # Emission factor for ashes (g/kWh)
oc_el = 1.025           # Organic carbon related to engine load (dimensi
onless)
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status', 'speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
/F) value to the question "Is the vessel in po
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname', 'shiptype','to_
bow','to_stern', 'to_port','to_starboard','draught', 'destination']
    aux=aux.drop(columns = ['date','type','shipname','draught','destinat
ion'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]
```

```
    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

s.speed.loc[s.speed > 40] = 0
s = s.loc[s.IMO != 0]
s['length'] = s.to_bow + s.to_stern
s['breadth'] = s.to_port + s.to_starboard
s = s.loc[s.length != 0]
s = s.loc[s.breadth != 0]
s = s.drop(columns = ['to_bow', 'to_stern', 'to_port', 'to_starboard'])

print("data length:", len(s))
s.head()
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1min')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset based on the co
rresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s.status.loc[s.status == 0] = labelstatus[0]
s.status.loc[s.status == 1] = labelstatus[1]
s.status.loc[s.status == 2] = labelstatus[2]
s.status.loc[s.status == 3] = labelstatus[3]
s.status.loc[s.status == 4] = labelstatus[4]
s.status.loc[s.status == 5] = labelstatus[5]
s.status.loc[s.status == 6] = labelstatus[6]
s.status.loc[s.status == 8] = labelstatus[7]
```

```
s.status.loc[s.status == 10] = labelstatus[8]
s.status.loc[s.status == 11] = labelstatus[9]
s.status.loc[s.status == 15] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

First dataset rearrangement

```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## This overcomes the fact that these vessels, due to their status, only
transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minute
s=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

Second dataset rearrangement

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 2nd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

Third dataset rearrangement

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 3rd rearrangement generates datasets that might not be available an
```

```
d stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli3_df))
dupli3_df.head()
```

Data appending

```
# Final data appending
## This section appends the rearranged dynamic data for vessels with sta
tus "Moored" and "At Anchor" with all other vessels
## The final data length is much larger than the initial one, as it guar
antees a dataset available per ship every 1min

s = pd.concat([s,dupli3_df], ignore_index = True)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
del(dupli3_df)

print("data length:", len(s))
s.head()
```

Emission inventory

Estimating installed power

```
# Inventory input
## This section assigns installed power based on the modelled equation
## It also inputs the auxiliary engine power, service speed and engine r
evolutions

s['ENG_KW'] = -1203 -0.000077091*(s.breadth**4.9) + 0.03408829*(s.length
**2.5)
s.ENG_KW.loc[s.group == 'Passenger'] = -5353.6 + 1.65640834*(s.length**1
.85)
s.ENG_KW.loc[s.group == 'Tankers'] = 3671.10566147 + 0.36347426*(s.lengt
h**2.15) - 7.5869e-38*(s.length**16) - 0.18208*((np.log(s.length))**7)
s['AUX_KW'] = 1000
s.AUX_KW.loc[s.group == 'Tankers'] = 1000
s.AUX_KW.loc[s.group == 'Passenger'] = 1250
```

```
s['service_speed'] = 19
s.service_speed.loc[s.group == 'Passenger'] = 22.5
s.service_speed.loc[s.group == 'Tankers'] = 14.5

s['rpm'] = 350
s.rpm.loc[s.group == 'Passenger'] = 400
s.rpm.loc[s.group == 'Tankers'] = 325
```

Phase filtering

```
# Phase filtering
## This section separates the database into the 4 stages: "Cruising", "A
t Anchor", "Maneuvering" and "Hoteling"

s['AE'] = 0.6
s['SFC'] = SFOC
s.inport = s.inport.astype(int)
s['SC'] = SC_fuel/100
s['CC'] = CC_fuel/100
sf_in = s.loc[s.inport == 1]
sf_hotelling = sf_in.loc[sf_in.speed <= 0.5]
sf_maneuvering = sf_in.loc[sf_in.speed > 0.5]
sf_out = s.loc[s.inport == 0]
sf_anchor = sf_out.loc[sf_out.speed <= 1.5]
sf_cruising = sf_out.loc[sf_out.speed > 1.5]
```

Cruising emissions

```
# Cruising emissions
## This section computes the emissions of vessels in the "cruising" stag
e per minute
## Main Engine loads are computed through the Propeller Law, whereas 60%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## Main engines are considered to burn LSHFO, whereas all auxiliary engi
nes burn MGO

sf_cruising = sf_cruising.drop(columns = ['IMO', 'mmsi'])
sf_cruising.AE.loc[sf_cruising.group == 'Passenger'] = 0.8

sf_cruising['k'] = EL*sf_cruising.ENG_KW/((sf_cruising.service_speed*185
2/3600)**3)
sf_cruising['trans_KW'] = sf_cruising.k*(sf_cruising.speed*1852/3600)**3
sf_cruising['SFOC'] = sf_cruising.SFC*(0.455*(EL*(sf_cruising.speed/sf_c
ruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.se
rvice_speed)**3)+1.28)
sf_cruising['SFOC_AE'] = SFOC_AE*(0.455*(sf_cruising.AE)**2-0.17*(sf_cru
ising.AE)+1.28)
sf_cruising['FC'] = (sf_cruising.trans_KW*sf_cruising.SFOC*(1/60) + sf_c
ruising.AE*sf_cruising.AUX_KW*sf_cruising.SFOC_AE*(1/60))*1e-6
sf_cruising['SO2'] = ((sf_cruising.SFOC*sf_cruising.SC/m_S)*m_SO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*SC_diesel/10
0/m_S)*m_SO2*sf_cruising.AUX_KW*(1/60))*1e-6
```

```
sf_cruising['CO2'] = ((sf_cruising.SFOC*sf_cruising.CC/m_C)*m_CO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*CC_diesel/10
0/m_C)*m_CO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['NOx'] = ((45*sf_cruising.rpm**-0.2)*sf_cruising.trans_KW*(1
/60) + sf_cruising.AE*(45*rpm**-0.2)*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['PM'] = (sf_cruising.trans_KW*(0.455*(EL*(sf_cruising.speed/
sf_cruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruisin
g.service_speed)**3)+1.28)*((0.312*sf_cruising.SC)+(0.244*sf_cruising.SC
)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_cruising.AE*sf_cruising.AUX_KW*(
0.455*sf_cruising.AE**2-0.17*sf_cruising.AE+1.28)*((0.312*SC_diesel/100)
+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_cruising = sf_cruising.drop(columns = ['inport', 'rpm', 'AE', 'breadt
h', 'length', 'SFC', 'SFOC', 'SFOC_AE', 'SC', 'CC', 'ENG_KW', 'AUX_KW',
'service_speed', 'k', 'trans_KW'])
```

Maneuvering emissions

```
# Maneuvering emissions
## This section computes the emissions of vessels in the "maneuvering" s
tage per minute
## Main Engine loads are computed through the Propeller Law, whereas 70%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## All engines are considered to burn MGO

sf_maneuvering = sf_maneuvering.drop(columns = ['IMO', 'mmsi'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_maneuvering.AE.loc[sf_maneuvering.group == 'Passenger'] = 0.8
sf_maneuvering.AE.loc[sf_maneuvering.group != 'Passenger'] = 0.7

sf_maneuvering['k'] = EL*sf_maneuvering.ENG_KW/((sf_maneuvering.service_
speed*1852/3600)**3)
sf_maneuvering['trans_KW'] = sf_maneuvering.k*(sf_maneuvering.speed*1852
/3600)**3
sf_maneuvering['SFOC'] = sf_maneuvering.SFC*(0.455*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed
/sf_maneuvering.service_speed)**3)+1.28)
sf_maneuvering['SFOC_AE'] = SFOC_AE*(0.455*(sf_maneuvering.AE)**2-0.17*(
sf_maneuvering.AE)+1.28)
sf_maneuvering['FC'] = (sf_maneuvering.trans_KW*sf_maneuvering.SFOC*(1/6
0) + sf_maneuvering.SFOC_AE*sf_maneuvering.AE*sf_maneuvering.AUX_KW*(1/6
0))*1e-6
sf_maneuvering['SO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.SC/m_S)*m_S
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*SC_diesel/100/m_S)*m_SO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['CO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.CC/m_C)*m_C
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*CC_diesel/100/m_C)*m_CO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['NOx'] = ((45*sf_maneuvering.rpm**-0.2)*sf_maneuvering.tr
ans_KW*(1/60) + sf_maneuvering.AE*(45*rpm**-0.2)*sf_maneuvering.AUX_KW*(
```

```
1/60))*1e-6
sf_maneuvering['PM'] = (sf_maneuvering.trans_KW*(0.455*(EL*(sf_maneuveri
ng.speed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)+1.28)*((0.312*sf_maneuvering.SC)+
(0.244*sf_maneuvering.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_maneuver
ing.AE*sf_maneuvering.AUX_KW*(0.455*sf_maneuvering.AE**2-0.17*sf_maneuve
ring.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+e
f_ec+ef_ash)*(1/60))*1e-6
sf_maneuvering = sf_maneuvering.drop(columns = ['inport', 'rpm', 'AE', '
breadth', 'length', 'SC', 'SFOC_AE', 'SFOC','SFC', 'CC', 'ENG_KW', 'AUX_
KW', 'service_speed', 'k', 'trans_KW'])
```

At Anchor emissions

```
# At Anchor emissions
## This section computes the emissions of vessels in the "at anchor" sta
ge per minute
## All vessels are considered to use only auxiliary power, with 70% load
on passenger and
## tanker vessels, and 40% assigned to those on cargo vessels
## They are all considered to burn MGO

sf_anchor = sf_anchor.drop(columns = ['IMO', 'mmsi'])
sf_anchor.AE.loc[sf_anchor.group == 'Passenger'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Tankers'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Cargo'] = 0.4

sf_anchor['SFOC_AE'] = SFOC_AE*(0.455*(sf_anchor.AE)**2-0.17*(sf_anchor.
AE)+1.28)
sf_anchor['FC'] = (sf_anchor.AE*sf_anchor.AUX_KW*sf_anchor.SFOC_AE*(1/60
))*1e-6
sf_anchor['SO2'] = (sf_anchor.AE*(sf_anchor.SFOC_AE*SC_diesel/100/m_S)*m
_SO2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['CO2'] = (sf_anchor.AE*(sf_anchor.SFOC_AE*CC_diesel/100/m_C)*m
_CO2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['NOx'] = (sf_anchor.AE*sf_anchor.AUX_KW*(45*rpm**-0.2)*(1/60))
*1e-6
sf_anchor['PM'] = (sf_anchor.AE*sf_anchor.AUX_KW*(0.455*sf_anchor.AE**2-
0.17*sf_anchor.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_
oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_anchor = sf_anchor.drop(columns = ['inport', 'rpm', 'AE', 'length', '
breadth', 'SC', 'SFC', 'CC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_spe
ed'])
```

Hotelling emissions

```
# Hoteling emissions
## This section computes the emissions of vessels in the "hoteling" stag
e per minute
## All vessels are considered to use only auxiliary power, with 70% load
on passenger and
## tanker vessels, and 40% assigned to those on cargo vessels
## They are all considered to burn MGO
```

```
sf_hotelling = sf_hotelling.drop(columns = ['IMO', 'mmsi'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_hotelling.AE.loc[sf_hotelling.group == 'Passenger'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Tankers'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Cargo'] = 0.4

sf_hotelling['SFOC_AE'] = SFOC_AE*(0.455*(sf_hotelling.AE)**2-0.17*(sf_h
otelling.AE)+1.28)
sf_hotelling['FC'] = (sf_hotelling.AE*sf_hotelling.AUX_KW*SFOC*(1/60))*1
e-6
sf_hotelling['SO2'] = (sf_hotelling.AE*(sf_hotelling.SFOC_AE*SC_diesel/1
00/m_S)*m_SO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['CO2'] = (sf_hotelling.AE*(sf_hotelling.SFOC_AE*CC_diesel/1
00/m_C)*m_CO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['NOx'] = (sf_hotelling.AE*sf_hotelling.AUX_KW*(45*rpm**-0.2
)*(1/60))*1e-6
sf_hotelling['PM'] = (sf_hotelling.AE*sf_hotelling.AUX_KW*(0.455*sf_hote
lling.AE**2-0.17*sf_hotelling.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_
diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_hotelling = sf_hotelling.drop(columns = ['inport', 'rpm', 'AE', 'leng
th', 'breadth', 'SC', 'SFC', 'CC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'servi
ce_speed'])
```

Data appending

```
# Emission data appending
## This section appends all 4 inventories per stage into a single consol
idated one

e = sf_cruising.append(sf_maneuvering, ignore_index = True)
e = e.append(sf_anchor, ignore_index = True)
e = e.append(sf_hotelling, ignore_index = True)

print("data length:", len(e))
e.head()
```

Results

Emission - Total

```
# Fuel consumption and emissions
## This section computes the total fuel consumption and emissions (tons)

e['date'] = e.date.dt.round('1d')
ef = e.drop(columns = ['date','status','speed','lat','lon','group'])
ef.sum(axis = 0)
```

Emission - Day

```
# Fuel consumption and emissions per day
## This section computes the fuel consumption and emissions (tons) per d
ay

e.groupby(['date'])['FC','SO2','CO2','NOx','PM'].sum()
```

Emission - Shiptype

```
# Fuel consumption and emissions per shiptype
## This section computes the fuel consumption and emissions (tons) per s
hiptype

e.groupby(['group'])['FC','SO2','CO2','NOx','PM'].sum()
```

Plot

```
# Fuel consumption and emissions per ship
## This section computes the fuel consumption and emissions (tons) per s
hip

fig, ax = plt.subplots(figsize=(15,7))
e.groupby(['date'])['FC'].sum().plot(ax=ax).legend(['Total'])
plt.ylabel('ships in range')
ax.set_xlim(pd.Timestamp('2020-03-01'), pd.Timestamp('2020-07-31'))
plt.title('Hourly count of ships in range')
plt.grid()
plt.show()
```

A2.3. Code for AISemissions_map.py

AIS emissions - emission map

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
August 23, 2020

```
import pandas as pd
import numpy as np
import datetime as dt
import folium
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates
from folium.plugins import TimestampedGeoJson

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelColor = ['#0000FF','#32CD32','#FF0000']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverabili
ty', 'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']

# Map boundaries function
## This function generates an OpenStreetMap Map centered at the Port de
Barcelona

def generateBaseMap(default_location=[41.382472, 2.205039], default_zoom
_start=12):
    base_map = folium.Map(location=default_location, control_scale=True,
zoom_start=default_zoom_start,tiles='cartodbpositron', width=640, height
=480)
    return base_map

# Live map function
## This function transforms the database into points to be plotted in a
dynamic map

def create_geojson_features(s):
```

```
    features = []

    for _, row in s.iterrows():
        feature = {
            'type': 'Feature',
            'geometry': {
                'type':'Point',
                'coordinates':[row['lon'],row['lat']]
            },
            'properties': {
                'time': pd.to_datetime(row['date']).__str__(),
                'popup':'name: '+row['Name'].__str__()+'<br>'+'speed: '+
row['speed'].__str__()+' knots'+'<br>'+'status: '+row['status'].__str__(
)+'<br>'+'FC: '+row['FC'].__str__()+' kg'+'<br>'+'SO2: '+row['SO2'].__st
r__()+' kg'+'<br>'+'CO2: '+row['CO2'].__str__()+' kg'+'<br>'+'NOx: '+row
['NOx'].__str__()+' kg'+'<br>'+'PM: '+row['PM'].__str__()+' kg',
                'style': {'color' : ''},
                'icon': 'circle',
                'iconstyle':{
                    'fillColor': row['fillColor'],
                    'fillOpacity': 0.8,
                    'radius': 5
                }
            }
        }
        features.append(feature)
    return features

# Input data for emission calculation

# Engine data
EL = 0.80               # Average design engine load on merchant ships (
Jalkanen et al, 2012)
SFOC = 200              # Average specific fuel oil consumption (g/kWh)
(Jalkanen et al, 2009)
SFOC_AE = 220
rpm = 500               # Average working revolutions on medium speed en
gines (Jalkanen et al, 2009)

# Fuel qualities - These values are maximum as per ISO 8217 standards /
chemistry of natural gas
SC_fuel = 0.5           # Sulfur content of Light Fuel Oil (%)
CC_fuel = 86            # Carbon content of Light Fuel Oil as per ISO 82
17 (%)
SC_diesel = 0.5         # Sulfur content of Marine Gasoil (%)
CC_diesel = 87.5        # Carbon content of Marine Gasoil as per ISO 821
7 (%)
SC_lng = 4e-3           # Sulfur content of LNG (%)
CC_lng = 75             # Carbon content of LNG (%)

# Element properties
m_S = 32.0655           # Molar mass of sulfur (g/mol)
```

```
m_SO2 = 64.06436        # Molar mass of sulfur dioxide (g/mol)
m_C = 12.01             # Molar mass of carbon (g/mol)
m_CO2 = 44.0886         # Molar mass of carbon dioxide (g/mol)

# Data for PM calculation
ef_ec = 0.08            # Emission factor for elementary carbon (g/kWh)
ef_oc = 0.2             # Emission factor for organic carbon (g/kWh)
ef_ash = 0.06           # Emission factor for ashes (g/kWh)
oc_el = 1.025           # Organic carbon related to engine load (dimensi
onless)
```

Data loading and filtering

Data loading

```
# Data Loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status', 'speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
/F) value to the question "Is the vessel in po
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname', 'shiptype','to_
bow','to_stern', 'to_port','to_starboard','draught', 'destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
, 'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
```

```
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

s.speed.loc[s.speed > 40] = 0
s = s.loc[s.IMO != 0]
print("data length:", len(s))
s.head()

p = pd.read_excel('Particulars.xlsx')
p.SFC.loc[p.SFC == 0] = SFOC
p['ef_NOx'] = (44*rpm**-0.23)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm < 130)] = 17
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 130) & (p.rpm < 2000)] = (45*p
.rpm**-0.2)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 2000)] = 9.8
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm < 130)] = 14
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm >= 2000)] = 7.7
p = p.drop(columns = ['Built', 'rpm'])
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1min')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')

print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset based on the co
rresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s.status.loc[s.status == 0] = labelstatus[0]
s.status.loc[s.status == 1] = labelstatus[1]
s.status.loc[s.status == 2] = labelstatus[2]
s.status.loc[s.status == 3] = labelstatus[3]
s.status.loc[s.status == 4] = labelstatus[4]
s.status.loc[s.status == 5] = labelstatus[5]
s.status.loc[s.status == 6] = labelstatus[6]
s.status.loc[s.status == 8] = labelstatus[7]
s.status.loc[s.status == 10] = labelstatus[8]
```

```
s.status.loc[s.status == 11] = labelstatus[9]
s.status.loc[s.status == 15] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

First dataset rearrangement

```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## This overcomes the fact that these vessels, due to their status, only
transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minute
s=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

Second dataset rearrangement

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 2nd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

Third dataset rearrangement

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
```

```
## A 3rd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli3_df))
dupli3_df.head()
```

Data appending

```
# Final data appending
## This section appends the rearranged dynamic data for vessels with sta
tus "Moored" and "At Anchor" with all other vessels
## The final data length is much larger than the initial one, as it guar
antees a dataset available per ship every 1min

s = pd.concat([s,dupli3_df], ignore_index = True)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
del(dupli3_df)

print("data length:", len(s))
s.head()
```

Emission inventory

Phase filtering

```
# Phase filtering
## This section separates the database into the 4 stages: "Cruising", "A
t Anchor", "Maneuvering" and "Hoteling"

s['AE'] = 0.6
s.inport = s.inport.astype(int)
s['SC'] = SC_fuel/100
s['CC'] = CC_fuel/100
sf_in = s.loc[s.inport == 1]
sf_hotelling = sf_in.loc[sf_in.speed <= 0.5]
sf_maneuvering = sf_in.loc[sf_in.speed > 0.5]
sf_out = s.loc[s.inport == 0]
sf_anchor = sf_out.loc[sf_out.speed <= 1.5]
sf_cruising = sf_out.loc[sf_out.speed > 1.5]
```

Crusing emissions

```
# Cruising emissions
## This section computes the emissions of vessels in the "cruising" stag
e per minute
## Main Engine loads are computed through the Propeller Law, whereas 60%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_cruising = sf_cruising.merge(p, how = 'left', on = ['IMO','IMO'])
sf_cruising = sf_cruising.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_cruising.SC.loc[sf_cruising.Fuel == 'LNG'] = SC_lng/100
sf_cruising.CC.loc[sf_cruising.Fuel == 'LNG'] = CC_lng/100
sf_cruising.AE.loc[sf_cruising.group == 'Passenger'] = 0.8

sf_cruising['k'] = EL*sf_cruising.ENG_KW/((sf_cruising.service_speed*185
2/3600)**3)
sf_cruising['trans_KW'] = sf_cruising.k*(sf_cruising.speed*1852/3600)**3
sf_cruising['SFOC'] = sf_cruising.SFC*(0.455*(EL*(sf_cruising.speed/sf_c
ruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.se
rvice_speed)**3)+1.28)
sf_cruising['SFOC_AE'] = SFOC_AE*(0.455*(sf_cruising.AE)**2-0.17*(sf_cru
ising.AE)+1.28)
sf_cruising['FC'] = (sf_cruising.trans_KW*sf_cruising.SFOC*(1/60) + sf_c
ruising.AE*sf_cruising.AUX_KW*sf_cruising.SFOC_AE*(1/60))*1e-6
sf_cruising['SO2'] = ((sf_cruising.SFOC*sf_cruising.SC/m_S)*m_SO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*SC_diesel/10
0/m_S)*m_SO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['CO2'] = ((sf_cruising.SFOC*sf_cruising.CC/m_C)*m_CO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*CC_diesel/10
0/m_C)*m_CO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['NOx'] = (sf_cruising.ef_NOx*sf_cruising.trans_KW*(1/60) + s
f_cruising.AE*(45*rpm**-0.2)*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['PM'] = (sf_cruising.trans_KW*(0.455*(EL*(sf_cruising.speed/
sf_cruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruisin
g.service_speed)**3)+1.28)*((0.312*sf_cruising.SC)+(0.244*sf_cruising.SC
)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_cruising.AE*sf_cruising.AUX_KW*(
0.455*sf_cruising.AE**2-0.17*sf_cruising.AE+1.28)*((0.312*SC_diesel/100)
+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_cruising = sf_cruising.drop(columns = ['inport', 'AE', 'Fuel', 'SC',
'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'k'
, 'trans_KW', 'ef_NOx'])
```

Maneuvering emissions

```
# Maneuvering emissions
## This section computes the emissions of vessels in the "maneuvering" s
tage per minute
## Main Engine loads are computed through the Propeller Law, whereas 70%
```

```
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## All engines are considered to burn MGO

sf_maneuvering = sf_maneuvering.merge(p, how = 'left', on = ['IMO','IMO'
])
sf_maneuvering = sf_maneuvering.drop(columns = ['IMO', 'mmsi', 'GT', 'Di
sp'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_maneuvering.AE.loc[sf_maneuvering.group == 'Passenger'] = 0.8
sf_maneuvering.AE.loc[sf_maneuvering.group != 'Passenger'] = 0.7

sf_maneuvering['k'] = EL*sf_maneuvering.ENG_KW/((sf_maneuvering.service_
speed*1852/3600)**3)
sf_maneuvering['trans_KW'] = sf_maneuvering.k*(sf_maneuvering.speed*1852
/3600)**3
sf_maneuvering['SFOC'] = sf_maneuvering.SFC*(0.455*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed
/sf_maneuvering.service_speed)**3)+1.28)
sf_maneuvering['SFOC_AE'] = SFOC_AE*(0.455*(sf_maneuvering.AE)**2-0.17*(
sf_maneuvering.AE)+1.28)
sf_maneuvering['FC'] = (sf_maneuvering.trans_KW*sf_maneuvering.SFOC*(1/6
0) + sf_maneuvering.SFOC_AE*sf_maneuvering.AE*sf_maneuvering.AUX_KW*(1/6
0))*1e-6
sf_maneuvering['SO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.SC/m_S)*m_S
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*SC_diesel/100/m_S)*m_SO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['CO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.CC/m_C)*m_C
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*CC_diesel/100/m_C)*m_CO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['NOx'] = (sf_maneuvering.ef_NOx*sf_maneuvering.trans_KW*(
1/60) + sf_maneuvering.AE*(45*rpm**-0.2)*sf_maneuvering.AUX_KW*(1/60))*1
e-6
sf_maneuvering['PM'] = (sf_maneuvering.trans_KW*(0.455*(EL*(sf_maneuveri
ng.speed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)+1.28)*((0.312*sf_maneuvering.SC)+
(0.244*sf_maneuvering.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_maneuver
ing.AE*sf_maneuvering.AUX_KW*(0.455*sf_maneuvering.AE**2-0.17*sf_maneuve
ring.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+e
f_ec+ef_ash)*(1/60))*1e-6
sf_maneuvering = sf_maneuvering.drop(columns = ['inport', 'AE', 'Fuel',
'SC', 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed
', 'k', 'trans_KW', 'ef_NOx'])
```

At Anchor emissions

```
# At Anchor emissions
## This section computes the emissions of vessels in the "at anchor" sta
ge per minute
## Main Engine loads are estimated at 10% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
```

```
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_anchor = sf_anchor.merge(p, how = 'left', on = ['IMO','IMO'])
sf_anchor = sf_anchor.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_anchor.SC.loc[sf_anchor.Fuel == 'LNG'] = SC_lng/100
sf_anchor.CC.loc[sf_anchor.Fuel == 'LNG'] = CC_lng/100
sf_anchor.AE.loc[sf_anchor.group == 'Passenger'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Tankers'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Cargo'] = 0.4
sf_anchor.ENG_KW.loc[sf_anchor.AUX_KW != 0 ] = 0

sf_anchor['SFOC'] = sf_anchor.SFC*(0.455*(EL*0.1)**2-0.17*(EL*0.1)+1.28)
sf_anchor['SFOC_AE'] = SFOC_AE*(0.455*(sf_anchor.AE)**2-0.17*(sf_anchor.
AE)+1.28)
sf_anchor['FC'] = (0.1*sf_anchor.ENG_KW*sf_anchor.SFOC*(1/60) + sf_ancho
r.AE*sf_anchor.AUX_KW*sf_anchor.SFOC_AE*(1/60))*1e-6
sf_anchor['SO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.SC/m_S)*m_SO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*SC_diesel/100/m_S)*m_S
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['CO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.CC/m_C)*m_CO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*CC_diesel/100/m_C)*m_C
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['NOx'] = (0.1*sf_anchor.ef_NOx*sf_anchor.ENG_KW*(1/60) + sf_an
chor.AE*(45*rpm**-0.2)*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['PM'] = (0.1*sf_anchor.ENG_KW*(0.455*(0.1)**2-0.17*(0.1)+1.28)
*((0.312*sf_anchor.SC)+(0.244*sf_anchor.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1
/60) + sf_anchor.AE*sf_anchor.AUX_KW*(0.455*sf_anchor.AE**2-0.17*sf_anch
or.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_
ec+ef_ash)*(1/60))*1e-6
sf_anchor = sf_anchor.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC'
, 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'ef_NOx
'])
```

Hotelling emissions

```
# Hoteling emissions
## This section computes the emissions of vessels in the "hoteling" stag
e per minute
## Main Engine loads are estimated at 20% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn either MGO or LNG, whereas all au
xiliary engines burn MGO

sf_hotelling = sf_hotelling.merge(p, how = 'left', on = ['IMO','IMO'])
sf_hotelling = sf_hotelling.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp']
)
sf_hotelling.SC.loc[sf_hotelling.Fuel == 'LNG'] = SC_lng/100
sf_hotelling.CC.loc[sf_hotelling.Fuel == 'LNG'] = CC_lng/100
```

```
sf_hotelling.SC.loc[sf_hotelling.Fuel != 'LNG'] = SC_diesel/100
sf_hotelling.CC.loc[sf_hotelling.Fuel != 'LNG'] = CC_diesel/100
sf_hotelling.AE.loc[sf_hotelling.group == 'Passenger'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Tankers'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Cargo'] = 0.4
sf_hotelling.ENG_KW.loc[sf_hotelling.AUX_KW != 0 ] = 0

sf_hotelling['SFOC'] = sf_hotelling.SFC*(0.455*(EL*0.2)**2-0.17*(EL*0.2)
+1.28)
sf_hotelling['SFOC_AE'] = SFOC_AE*(0.455*(sf_hotelling.AE)**2-0.17*(sf_h
otelling.AE)+1.28)
sf_hotelling['FC'] = (0.2*sf_hotelling.ENG_KW*sf_hotelling.SFOC*(1/60) +
sf_hotelling.AE*sf_hotelling.AUX_KW*SFOC*(1/60))*1e-6
sf_hotelling['SO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.SC/m_S)*m_SO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*SC_d
iesel/100/m_S)*m_SO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['CO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.CC/m_C)*m_CO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*CC_d
iesel/100/m_C)*m_CO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['NOx'] = (0.2*sf_hotelling.ef_NOx*sf_hotelling.ENG_KW*(1/60
) + sf_hotelling.AE*(45*rpm**-0.2)*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['PM'] = (0.2*sf_hotelling.ENG_KW*(0.455*(0.2)**2-0.17*(0.2)
+1.28)*((0.312*sf_hotelling.SC)+(0.244*sf_hotelling.SC)+ef_oc*oc_el+ef_e
c+ef_ash)*(1/60) + sf_hotelling.AE*sf_hotelling.AUX_KW*(0.455*sf_hotelli
ng.AE**2-0.17*sf_hotelling.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_die
sel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_hotelling = sf_hotelling.drop(columns = ['inport', 'AE', 'Fuel', 'SC'
, 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', '
ef_NOx'])
```

Data appending

```
# Emission data appending
## This section appends all 4 inventories per stage into a single consol
idated one

e = sf_cruising.append(sf_maneuvering, ignore_index = True)
e = e.append(sf_anchor, ignore_index = True)
e = e.append(sf_hotelling, ignore_index = True)

print("data length:", len(e))
e.head()

e.FC = (e.FC*1000).round(1)
e.SO2 = (e.SO2*1000).round(2)
e.CO2 = (e.CO2*1000).round(1)
e.NOx = (e.NOx*1000).round(2)
e.PM = (e.PM*1000).round(3)
```

Results

```
# Data modification
## This section modifies data to be presented in a plot

e['fillColor'] = labelColor[0]
e.fillColor.loc[e.group == 'Tankers'] = labelColor[2]
e.fillColor.loc[e.group == 'Cargo'] = labelColor[1]
e['month'] = e.date.apply(lambda x: x.month)
e['day'] = e.date.apply(lambda x: x.day)

# Select month (in numbers) and range (in numbers) - maximum 5 days

month = 4
start_day = 1
end_day = 3
```

Plot

```
# Live map
## This section generates the live map plotting all vessels, their info
and postn. in 1min time intervals

s_copy = e.loc[(e.day >= start_day) & (e.day <= end_day) & (e.month ==
month)]
start_geojson = create_geojson_features(s_copy)
base_map = generateBaseMap()
TimestampedGeoJson(start_geojson, period = 'PT1M', add_last_point=True,
duration = 'PT59S', transition_time = 0.0000001, max_speed = 100,
auto_play = True).add_to(base_map)
base_map
```

A2.4. Code for AISemissions_heatmap.py

AIS emissions - emission map

Developed by: Javier Nieto-Guarasa
Supervised by: Anna Mujal-Colilles, PhD
Polytechnic University of Catalonia
August 23, 2020

```
import pandas as pd
import numpy as np
import datetime as dt
import folium
import os

import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import matplotlib.dates as mdates

from matplotlib.ticker import (MultipleLocator, FormatStrFormatter, Auto
MinorLocator)
from matplotlib import ticker, cm
from modMeu import apb_lim, join_DynStat, inport
from folium.plugins import HeatMap
from folium.plugins import HeatMapWithTime

folder = ['C:/Users/nieto/202003','C:/Users/nieto/202004','C:/Users/niet
o/202005','C:/Users/nieto/202006','C:/Users/nieto/202007']
labelship = ['Passenger','Cargo','Tankers']
labelstatus = ['Underway', 'At Anchor', 'NUC', 'Restricted Maneuverabili
ty', 'Constrained by her draught', 'Moored',
               'Aground', 'Sailing', 'Error', 'Towing', 'Undefined']

# Map boundaries function
## This function generates an OpenStreetMap Map centered at the Port de
Barcelona

def generateBaseMap(default_location=[41.382472, 2.205039], default_zoom
_start=12):
    base_map = folium.Map(location=default_location, control_scale=True,
zoom_start=default_zoom_start,tiles='cartodbpositron', width=640, height
=480)
    return base_map

# Input data for emission calculation

# Engine data
EL = 0.80               # Average design engine load on merchant ships (
Jalkanen et al, 2012)
SFOC = 200              # Average specific fuel oil consumption (g/kWh)
```

```
(Jalkanen et al, 2009)
SFOC_AE = 220
rpm = 500               # Average working revolutions on medium speed en
gines (Jalkanen et al, 2009)

# Fuel qualities - These values are maximum as per ISO 8217 standards /
chemistry of natural gas
SC_fuel = 0.5           # Sulfur content of Light Fuel Oil (%)
CC_fuel = 86            # Carbon content of Light Fuel Oil as per ISO 82
17 (%)
SC_diesel = 0.5         # Sulfur content of Marine Gasoil (%)
CC_diesel = 87.5        # Carbon content of Marine Gasoil as per ISO 821
7 (%)
SC_lng = 4e-3           # Sulfur content of LNG (%)
CC_lng = 75             # Carbon content of LNG (%)

# Element properties
m_S = 32.0655           # Molar mass of sulfur (g/mol)
m_SO2 = 64.06436        # Molar mass of sulfur dioxide (g/mol)
m_C = 12.01             # Molar mass of carbon (g/mol)
m_CO2 = 44.0886         # Molar mass of carbon dioxide (g/mol)

# Data for PM calculation
ef_ec = 0.08            # Emission factor for elementary carbon (g/kWh)
ef_oc = 0.2             # Emission factor for organic carbon (g/kWh)
ef_ash = 0.06           # Emission factor for ashes (g/kWh)
oc_el = 1.025           # Organic carbon related to engine load (dimensi
onless)
```

Data loading and filtering

Data loading

```
# Data loading
## This section reads all available static and dynamic AIS data and conv
erts them into a workable pandas DataFrame

r = 30 # Enter range radius in nautical miles (1nm = 1852m)

s = pd.DataFrame()
for fold in folder:
    os.chdir(fold)

    # Dynamic data range filter + Inport function
    t = pd.read_csv('ClassA_clean.csv', sep=",", usecols = ['date','mmsi
','lat','lon','status', 'speed'])
    t.speed = t.speed/10
    t = apb_lim(t, r)
    t['inport'] = inport(t.lat, t.lon) # This function returns a bool (T
```

```
/F) value to the question "Is the vessel in po
    m = np.unique(t.mmsi)

    # Static data loading
    aux = pd.read_csv('llista_arx_5.txt', sep = ",")
    aux.columns =['date','type','mmsi','IMO','shipname', 'shiptype','to_
bow','to_stern', 'to_port','to_starboard','draught', 'destination']
    aux=aux.drop(columns = ['date','type','shipname','to_bow','to_stern'
, 'to_port','to_starboard','draught','destination'])

    # Merchant fleet filter
    aux = aux[(aux.shiptype < 90)]
    aux = aux[(aux.shiptype > 59)]

    # Data range crosscheck
    aux = aux[aux.mmsi.isin(m)]
    aux = aux.drop_duplicates(subset='mmsi',keep="first")
    m = np.unique(aux.mmsi)
    t = t[t.mmsi.isin(m)]

    # Dataframe appending
    t = t.merge(aux, how = 'left', on = ['mmsi','mmsi'])
    s = s.append(t, ignore_index = True)
    del(m,t,aux)

s.speed.loc[s.speed > 40] = 0
s = s.loc[s.IMO != 0]
print("data length:", len(s))
s.head()

p = pd.read_excel('Particulars.xlsx')
p.SFC.loc[p.SFC == 0] = SFOC
p['ef_NOx'] = (44*rpm**-0.23)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm < 130)] = 17
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 130) & (p.rpm < 2000)] = (45*p
.rpm**-0.2)
p.ef_NOx.loc[(p.Built < 2011) & (p.rpm >= 2000)] = 9.8
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm < 130)] = 14
p.ef_NOx.loc[(p.Built >= 2011) & (p.rpm >= 2000)] = 7.7
p = p.drop(columns = ['Built', 'rpm'])
```

Time filtering

```
# Time filtering
## This section formats data columns into yyyy-mm-dd hh:mm:ss and drops
duplicated values based on dt = 1h basis

s.date = pd.to_datetime(s['date'], format = '%Y%m%d%H%M%S')
s['date'] = s.date.dt.round('1min')
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
```

```
print("data length:", len(s))
s.head()
```

Status filtering

```
# Status filtering
## This section assigns group and status to each dataset based on the co
rresponding numerical value

s['group'] = pd.cut(s.shiptype, 3, right=False, labels = labelship)
s.status.loc[s.status == 0] = labelstatus[0]
s.status.loc[s.status == 1] = labelstatus[1]
s.status.loc[s.status == 2] = labelstatus[2]
s.status.loc[s.status == 3] = labelstatus[3]
s.status.loc[s.status == 4] = labelstatus[4]
s.status.loc[s.status == 5] = labelstatus[5]
s.status.loc[s.status == 6] = labelstatus[6]
s.status.loc[s.status == 8] = labelstatus[7]
s.status.loc[s.status == 10] = labelstatus[8]
s.status.loc[s.status == 11] = labelstatus[9]
s.status.loc[s.status == 15] = labelstatus[10]
s = s.drop(columns = ['shiptype'])

print("data length:", len(s))
s.head()
```

First dataset rearrangement

```
# Dataset rearrangement (1st)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## This overcomes the fact that these vessels, due to their status, only
transmit dynamic data every 3 minutes

df = s.loc[(s.status == 'Moored') | (s.status== 'At Anchor')]
df = df.loc[df.speed < 3]
dupli_df = pd.concat([df]*3, ignore_index=True)
l_col_datetime = dupli_df.select_dtypes('datetime').columns
len_df = len(df)
dupli_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minute
s=1)
dupli_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli_df))
dupli_df.head()
```

Second dataset rearrangement

```
# Dataset rearrangement (2nd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
```

```
## A 2nd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli2_df = pd.concat([dupli_df]*3, ignore_index=True)
l_col_datetime = dupli2_df.select_dtypes('datetime').columns
len_df = len(dupli_df)
del(dupli_df)
dupli2_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli2_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli2_df))
dupli2_df.head()
```

Third dataset rearrangement

```
# Dataset rearrangement (3rd)
## This section copies all datasets with status "Moored" and "At Anchor"
with speeds < 3 knots and offsets the time by 1min
## A 3rd rearrangement generates datasets that might not be available an
d stabilizes the dynamic plot

dupli3_df = pd.concat([dupli2_df]*3, ignore_index=True)
l_col_datetime = dupli3_df.select_dtypes('datetime').columns
len_df = len(dupli2_df)
del(dupli2_df)
dupli3_df.loc[len_df:2*len_df-1, l_col_datetime] +=  pd.DateOffset(minut
es=1)
dupli3_df.loc[2*len_df:, l_col_datetime] +=  pd.DateOffset(minutes=2)

print("data length:", len(dupli3_df))
dupli3_df.head()
```

Data appending

```
# Final data appending
## This section appends the rearranged dynamic data for vessels with sta
tus "Moored" and "At Anchor" with all other vessels
## The final data length is much larger than the initial one, as it guar
antees a dataset available per ship every 1min

s = pd.concat([s,dupli3_df], ignore_index = True)
s = s.drop_duplicates(['date', 'mmsi'], keep = 'first')
del(dupli3_df)

print("data length:", len(s))
s.head()
```

Emission inventory

Phase filtering

```
# Phase filtering
## This section separates the database into the 4 stages: "Cruising", "A
t Anchor", "Maneuvering" and "Hoteling"

s['AE'] = 0.6
s.inport = s.inport.astype(int)
s['SC'] = SC_fuel/100
s['CC'] = CC_fuel/100
sf_in = s.loc[s.inport == 1]
sf_hotelling = sf_in.loc[sf_in.speed <= 0.5]
sf_maneuvering = sf_in.loc[sf_in.speed > 0.5]
sf_out = s.loc[s.inport == 0]
sf_anchor = sf_out.loc[sf_out.speed <= 1.5]
sf_cruising = sf_out.loc[sf_out.speed > 1.5]
```

Crusing emissions

```
# Cruising emissions
## This section computes the emissions of vessels in the "cruising" stag
e per minute
## Main Engine loads are computed through the Propeller Law, whereas 60%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_cruising = sf_cruising.merge(p, how = 'left', on = ['IMO','IMO'])
sf_cruising = sf_cruising.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_cruising.SC.loc[sf_cruising.Fuel == 'LNG'] = SC_lng/100
sf_cruising.CC.loc[sf_cruising.Fuel == 'LNG'] = CC_lng/100
sf_cruising.AE.loc[sf_cruising.group == 'Passenger'] = 0.8

sf_cruising['k'] = EL*sf_cruising.ENG_KW/((sf_cruising.service_speed*185
2/3600)**3)
sf_cruising['trans_KW'] = sf_cruising.k*(sf_cruising.speed*1852/3600)**3
sf_cruising['SFOC'] = sf_cruising.SFC*(0.455*(EL*(sf_cruising.speed/sf_c
ruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruising.se
rvice_speed)**3)+1.28)
sf_cruising['SFOC_AE'] = SFOC_AE*(0.455*(sf_cruising.AE)**2-0.17*(sf_cru
ising.AE)+1.28)
sf_cruising['FC'] = (sf_cruising.trans_KW*sf_cruising.SFOC*(1/60) + sf_c
ruising.AE*sf_cruising.AUX_KW*sf_cruising.SFOC_AE*(1/60))*1e-6
sf_cruising['SO2'] = ((sf_cruising.SFOC*sf_cruising.SC/m_S)*m_SO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*SC_diesel/10
0/m_S)*m_SO2*sf_cruising.AUX_KW*(1/60))*1e-6
```

```
sf_cruising['CO2'] = ((sf_cruising.SFOC*sf_cruising.CC/m_C)*m_CO2*sf_cru
ising.trans_KW*(1/60) + sf_cruising.AE*(sf_cruising.SFOC_AE*CC_diesel/10
0/m_C)*m_CO2*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['NOx'] = (sf_cruising.ef_NOx*sf_cruising.trans_KW*(1/60) + s
f_cruising.AE*(45*rpm**-0.2)*sf_cruising.AUX_KW*(1/60))*1e-6
sf_cruising['PM'] = (sf_cruising.trans_KW*(0.455*(EL*(sf_cruising.speed/
sf_cruising.service_speed)**3)**2-0.17*(EL*(sf_cruising.speed/sf_cruisin
g.service_speed)**3)+1.28)*((0.312*sf_cruising.SC)+(0.244*sf_cruising.SC
)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_cruising.AE*sf_cruising.AUX_KW*(
0.455*sf_cruising.AE**2-0.17*sf_cruising.AE+1.28)*((0.312*SC_diesel/100)
+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_cruising = sf_cruising.drop(columns = ['inport', 'AE', 'Fuel', 'SC',
'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'k'
, 'trans_KW', 'ef_NOx'])
```

Maneuvering emissions

```
# Maneuvering emissions
## This section computes the emissions of vessels in the "maneuvering" s
tage per minute
## Main Engine loads are computed through the Propeller Law, whereas 70%
is assigned to auxiliary engines on cargo and tanker
## vessels, and 80% is assigned to those on passenger vessels
## All engines are considered to burn MGO

sf_maneuvering = sf_maneuvering.merge(p, how = 'left', on = ['IMO','IMO'
])
sf_maneuvering = sf_maneuvering.drop(columns = ['IMO', 'mmsi', 'GT', 'Di
sp'])
sf_maneuvering.SC = SC_diesel/100
sf_maneuvering.CC = CC_diesel/100
sf_maneuvering.AE.loc[sf_maneuvering.group == 'Passenger'] = 0.8
sf_maneuvering.AE.loc[sf_maneuvering.group != 'Passenger'] = 0.7

sf_maneuvering['k'] = EL*sf_maneuvering.ENG_KW/((sf_maneuvering.service_
speed*1852/3600)**3)
sf_maneuvering['trans_KW'] = sf_maneuvering.k*(sf_maneuvering.speed*1852
/3600)**3
sf_maneuvering['SFOC'] = sf_maneuvering.SFC*(0.455*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.speed
/sf_maneuvering.service_speed)**3)+1.28)
sf_maneuvering['SFOC_AE'] = SFOC_AE*(0.455*(sf_maneuvering.AE)**2-0.17*(
sf_maneuvering.AE)+1.28)
sf_maneuvering['FC'] = (sf_maneuvering.trans_KW*sf_maneuvering.SFOC*(1/6
0) + sf_maneuvering.SFOC_AE*sf_maneuvering.AE*sf_maneuvering.AUX_KW*(1/6
0))*1e-6
sf_maneuvering['SO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.SC/m_S)*m_S
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*SC_diesel/100/m_S)*m_SO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
sf_maneuvering['CO2'] = ((sf_maneuvering.SFOC*sf_maneuvering.CC/m_C)*m_C
O2*sf_maneuvering.trans_KW*(1/60) + sf_maneuvering.AE*(sf_maneuvering.SF
OC_AE*CC_diesel/100/m_C)*m_CO2*sf_maneuvering.AUX_KW*(1/60))*1e-6
```

```
sf_maneuvering['NOx'] = (sf_maneuvering.ef_NOx*sf_maneuvering.trans_KW*(
1/60) + sf_maneuvering.AE*(45*rpm**-0.2)*sf_maneuvering.AUX_KW*(1/60))*1
e-6
sf_maneuvering['PM'] = (sf_maneuvering.trans_KW*(0.455*(EL*(sf_maneuveri
ng.speed/sf_maneuvering.service_speed)**3)**2-0.17*(EL*(sf_maneuvering.s
peed/sf_maneuvering.service_speed)**3)+1.28)*((0.312*sf_maneuvering.SC)+
(0.244*sf_maneuvering.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60) + sf_maneuver
ing.AE*sf_maneuvering.AUX_KW*(0.455*sf_maneuvering.AE**2-0.17*sf_maneuve
ring.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+e
f_ec+ef_ash)*(1/60))*1e-6
sf_maneuvering = sf_maneuvering.drop(columns = ['inport', 'AE', 'Fuel',
'SC', 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed
', 'k', 'trans_KW', 'ef_NOx'])
```

At Anchor emissions

```
# At Anchor emissions
## This section computes the emissions of vessels in the "at anchor" sta
ge per minute
## Main Engine loads are estimated at 10% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn their main fuel, whereas all auxi
liary engines burn MGO

sf_anchor = sf_anchor.merge(p, how = 'left', on = ['IMO','IMO'])
sf_anchor = sf_anchor.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp'])
sf_anchor.SC.loc[sf_anchor.Fuel == 'LNG'] = SC_lng/100
sf_anchor.CC.loc[sf_anchor.Fuel == 'LNG'] = CC_lng/100
sf_anchor.AE.loc[sf_anchor.group == 'Passenger'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Tankers'] = 0.7
sf_anchor.AE.loc[sf_anchor.group == 'Cargo'] = 0.4
sf_anchor.ENG_KW.loc[sf_anchor.AUX_KW != 0 ] = 0

sf_anchor['SFOC'] = sf_anchor.SFC*(0.455*(EL*0.1)**2-0.17*(EL*0.1)+1.28)
sf_anchor['SFOC_AE'] = SFOC_AE*(0.455*(sf_anchor.AE)**2-0.17*(sf_anchor.
AE)+1.28)
sf_anchor['FC'] = (0.1*sf_anchor.ENG_KW*sf_anchor.SFOC*(1/60) + sf_ancho
r.AE*sf_anchor.AUX_KW*sf_anchor.SFOC_AE*(1/60))*1e-6
sf_anchor['SO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.SC/m_S)*m_SO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*SC_diesel/100/m_S)*m_S
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['CO2'] = (0.1*(sf_anchor.SFOC*sf_anchor.CC/m_C)*m_CO2*sf_ancho
r.ENG_KW*(1/60) + sf_anchor.AE*(sf_anchor.SFOC_AE*CC_diesel/100/m_C)*m_C
O2*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['NOx'] = (0.1*sf_anchor.ef_NOx*sf_anchor.ENG_KW*(1/60) + sf_an
chor.AE*(45*rpm**-0.2)*sf_anchor.AUX_KW*(1/60))*1e-6
sf_anchor['PM'] = (0.1*sf_anchor.ENG_KW*(0.455*(0.1)**2-0.17*(0.1)+1.28)
*((0.312*sf_anchor.SC)+(0.244*sf_anchor.SC)+ef_oc*oc_el+ef_ec+ef_ash)*(1
/60) + sf_anchor.AE*sf_anchor.AUX_KW*(0.455*sf_anchor.AE**2-0.17*sf_anch
or.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_diesel/100)+ef_oc*oc_el+ef_
```

```
ec+ef_ash)*(1/60))*1e-6
sf_anchor = sf_anchor.drop(columns = ['inport', 'AE', 'Fuel', 'SC', 'CC'
, 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', 'ef_NOx
'])
```

Hotelling emissions

```
# Hoteling emissions
## This section computes the emissions of vessels in the "hoteling" stag
e per minute
## Main Engine loads are estimated at 20% for all vessels, whereas 70% i
s assigned to auxiliary engines on passenger and
## tanker vessels, and 40% is assigned to those on cargo vessels
## Main engines are considered to burn either MGO or LNG, whereas all au
xiliary engines burn MGO

sf_hotelling = sf_hotelling.merge(p, how = 'left', on = ['IMO','IMO'])
sf_hotelling = sf_hotelling.drop(columns = ['IMO', 'mmsi', 'GT', 'Disp']
)
sf_hotelling.SC.loc[sf_hotelling.Fuel == 'LNG'] = SC_lng/100
sf_hotelling.CC.loc[sf_hotelling.Fuel == 'LNG'] = CC_lng/100
sf_hotelling.SC.loc[sf_hotelling.Fuel != 'LNG'] = SC_diesel/100
sf_hotelling.CC.loc[sf_hotelling.Fuel != 'LNG'] = CC_diesel/100
sf_hotelling.AE.loc[sf_hotelling.group == 'Passenger'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Tankers'] = 0.7
sf_hotelling.AE.loc[sf_hotelling.group == 'Cargo'] = 0.4
sf_hotelling.ENG_KW.loc[sf_hotelling.AUX_KW != 0 ] = 0

sf_hotelling['SFOC'] = sf_hotelling.SFC*(0.455*(EL*0.2)**2-0.17*(EL*0.2)
+1.28)
sf_hotelling['SFOC_AE'] = SFOC_AE*(0.455*(sf_hotelling.AE)**2-0.17*(sf_h
otelling.AE)+1.28)
sf_hotelling['FC'] = (0.2*sf_hotelling.ENG_KW*sf_hotelling.SFOC*(1/60) +
sf_hotelling.AE*sf_hotelling.AUX_KW*SFOC*(1/60))*1e-6
sf_hotelling['SO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.SC/m_S)*m_SO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*SC_d
iesel/100/m_S)*m_SO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['CO2'] = (0.2*(sf_hotelling.SFOC*sf_hotelling.CC/m_C)*m_CO2
*sf_hotelling.ENG_KW*(1/60) + sf_hotelling.AE*(sf_hotelling.SFOC_AE*CC_d
iesel/100/m_C)*m_CO2*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['NOx'] = (0.2*sf_hotelling.ef_NOx*sf_hotelling.ENG_KW*(1/60
) + sf_hotelling.AE*(45*rpm**-0.2)*sf_hotelling.AUX_KW*(1/60))*1e-6
sf_hotelling['PM'] = (0.2*sf_hotelling.ENG_KW*(0.455*(0.2)**2-0.17*(0.2)
+1.28)*((0.312*sf_hotelling.SC)+(0.244*sf_hotelling.SC)+ef_oc*oc_el+ef_e
c+ef_ash)*(1/60) + sf_hotelling.AE*sf_hotelling.AUX_KW*(0.455*sf_hotelli
ng.AE**2-0.17*sf_hotelling.AE+1.28)*((0.312*SC_diesel/100)+(0.244*SC_die
sel/100)+ef_oc*oc_el+ef_ec+ef_ash)*(1/60))*1e-6
sf_hotelling = sf_hotelling.drop(columns = ['inport', 'AE', 'Fuel', 'SC'
, 'CC', 'SFC', 'SFOC', 'SFOC_AE', 'ENG_KW', 'AUX_KW', 'service_speed', '
ef_NOx'])
```

Data appending

```
# Emission data appending
## This section appends all 4 inventories per stage into a single consol
idated one

e = sf_cruising.append(sf_maneuvering, ignore_index = True)
e = e.append(sf_anchor, ignore_index = True)
e = e.append(sf_hotelling, ignore_index = True)

print("data length:", len(e))
e.head()
```

```
e.FC = (e.FC*1000).round(1)
e.SO2 = (e.SO2*1000).round(2)
e.CO2 = (e.CO2*1000).round(1)
e.NOx = (e.NOx*1000).round(2)
e.PM = (e.PM*1000).round(3)
```

Emission - Day

```
# Data modification
## This section modifies data to be presented in a plot

e['month'] = e.date.apply(lambda x: x.month)
e['week'] = e.date.apply(lambda x: x.week)
e['day'] = e.date.apply(lambda x: x.day)
e['hour'] = e.date.apply(lambda x: x.hour)
```

```
# Select month (in numbers) and range (in numbers) - maximum 5 days

month = 7
start_day = 15
end_day = 15
```

Plot

```
# Live heatmap
## This section generates a live heatmap representing the percentage of
emissions over the average per hour

s_copy = e.loc[(e.day >= start_day) & (e.day <= end_day) & (e.month == m
onth)]
s_hour_list = []
for hour in s_copy.hour.sort_values().unique():
    s_hour_list.append(s_copy.loc[s_copy.hour == hour, ['lat', 'lon', 'P
M']].groupby(['lat', 'lon']).sum().reset_index().values.tolist())
base_map = generateBaseMap(default_zoom_start=11)
HeatMapWithTime(s_hour_list, radius=5, gradient={0.2: 'blue', 0.4: 'lime
', 0.6: 'orange', 1: 'red'}, min_opacity=0.5, max_opacity=0.8, use_local
_extrema=True).add_to(base_map)
base_map
```

Annex A3. Maritime and port traffic analysis tables

This annex presents more detailed data listed in tables on the bar plots shown in sections 4.1.1, 4.1.2 and 4.1.4.

A3.1. Vessels in range

Cargo vessels in range for the 5-month period:

Table A 1. Monthly minimum, average and maximum number of cargo vessels in range
Values in brackets are the difference over the 5-monthly average of cargo vessels

Month	Minimum	Average	Maximum
March	8 (-48.7%)	15.07 (-3.4%)	23 (+47.5%)
April	7 (-55.1%)	16.31 (+4.6%)	30 (+92.4%)
May	6 (-61.5%)	15.06 (-3.4%)	29 (+86.0%)
June	6 (-61.5%)	15.62 (-0.2%)	30 (+92.4%)
July	4 (-74.3%)	15.74 (-0.9%)	31 (+98.9%)

Table A 2. Minimum, average and maximum number of cargo vessels per period
Values in brackets are the difference over the 5-monthly average of cargo vessels

Period	Minimum	Average	Maximum
Pre-lockdown	8 (-48.7%)	15.15 (-2.8%)	23 (+47.5%)
Lockdown	6 (-61.5%)	15.42 (-1.2%)	30 (+92.4%)
Home-quarantine	11 (-29.4%)	18.01 (+15.5%)	28 (+79.6%)
Post-lockdown	4 (-74.3%)	16.15 (+3.6%)	31 (+98.8%)

Tanker vessels in range for the 5-month period:

Table A 3. Monthly minimum, average and maximum number of tanker vessels in range
Values in brackets are the difference over the 5-monthly average of tanker vessels

Month	Minimum	Average	Maximum
March	6 (-42.6%)	11.67 (+11.6%)	21 (+100.7%)
April	6 (-42.6%)	11.64 (+11.3%)	18 (+72.1%)
May	4 (-61.8%)	9.67 (-7.6%)	17 (+62.5%)
June	3 (-71.3%)	8.98 (-14.2%)	15 (+43.4%)
July	3 (-71.3%)	10.27 (-1.8%)	19 (+81.6%)

Table A 4. Minimum, average and maximum number of tanker vessels per period
Values in brackets are the difference over the 5-monthly average of tanker vessels

Period	Minimum	Average	Maximum
Pre-lockdown	6 (-42.6%)	9.95 (-4.9%)	15 (+43.3%)
Lockdown	4 (-61.7%)	10.80 (+3.2%)	21 (+100.7%)
Home-quarantine	10 (-4.4%)	13.30 (+27.1%)	18 (+72.1%)
Post-lockdown	3 (-71.3%)	9.81 (-6.2%)	19 (+81.6%)

Passenger vessels in range for the 5-month period:

Table A 5. Monthly minimum, average and maximum number of passenger vessels in range
Values in brackets are the difference over the 5-monthly average of passenger vessels

Month	Minimum	Average	Maximum
March	0 (-100%)	4.59 (-15.2%)	10 (+84.7%)
April	3 (-44.6%)	6.06 (+11.9%)	9 (+66.3%)
May	3 (-44.6%)	6.39 (+18.0%)	10 (+84.7%)
June	0 (-100%)	5.85 (+8.1%)	11 (+103.2%)
July	0 (-100%)	4.70 (-13.2%)	12 (+121.7%)

Table A 6. Minimum, average and maximum number of passenger vessels per period
Values in brackets are the difference over the 5-monthly average of passenger vessels

Period	Minimum	Average	Maximum
Pre-lockdown	0 (-100%)	4.48 (-17.2%)	10 (+84.7%)
Lockdown	0 (-100%)	5.82 (+7.5%)	11 (+103.2%)
Home-quarantine	4 (-26.1%)	6.55 (+21.0%)	9 (+66.3%)
Post-lockdown	0 (-100%)	4.78 (-11.7%)	12 (+121.7%)

A3.2. Vessel status

Status of all vessels in range for the 5-month period:

Table A 7. Distribution of status for all vessels in range

Month	Underway	At Anchor	N.U.C.	Moored
March	30.81%	16.39%	< 0.01%	52.41%
April	26.40%	16.72%	< 0.01%	56.44%
May	24.22%	15.55%	0.01%	59.56%
June	25.64%	13.59%	< 0.01%	60.60%
July	32.96%	15.99%	< 0.01%	51.01%

Table A 8. Distribution of status for all vessels in range per period

Period	Underway	At Anchor	N.U.C.	Moored
Pre-lockdown	34.43%	13.06%	< 0.01%	52.21%
Lockdown	25.57%	16.26%	< 0.01%	57.70%
Home-quarantine	23.30%	17.31%	0.01%	58.47%
Post-lockdown	31.56%	15.34%	< 0.01%	53.07%

Status of cargo vessels in range for the 5-month period:

Table A 9. Distribution of status for cargo vessels in range

Month	Underway	At Anchor	N.U.C.	Moored
March	32.17%	14.13%	0.01%	53.08%
April	31.53%	20.03%	0.01%	47.91%
May	29.02%	15.90%	0.01%	53.84%
June	26.26%	15.56%	< 0.01%	57.88%
July	29.79%	19.59%	< 0.01%	50.57%

Table A 10. Distribution of status for cargo vessels in range per period

Period	Underway	At Anchor	N.U.C.	Moored
Pre-lockdown	32.50%	14.06%	0.01%	52.92%
Lockdown	30.05%	16.71%	0.01%	52.47%
Home-quarantine	30.55%	23.30%	0.01%	45.27%
Post-lockdown	28.29%	19.09%	< 0.01%	52.58%

Status of tanker vessels in range for the 5-month period:

Table A 11. Distribution of status for tanker vessels in range

Month	Underway	At Anchor	N.U.C.	Moored
March	24.99%	24.16%	< 0.01%	50.82%
April	25.94%	21.78%	< 0.01%	51.85%
May	25.33%	25.05%	< 0.01%	49.47%
June	26.53%	16.79%	< 0.01%	56.67%
July	24.98%	17.34%	< 0.01%	57.64%

Table A 12. Distribution of status for tanker vessels in range per period

Period	Underway	At Anchor	N.U.C.	Moored
Pre-lockdown	26.82%	17.02%	0.00%	56.16%
Lockdown	21.13%	19.05%	< 0.01%	58.64%
Home-quarantine	25.44%	23.94%	< 0.01%	50.43%
Post-lockdown	24.81%	15.82%	< 0.01%	59.35%

Status of passenger vessels in range for the 5-month period:

Table A 13. Distribution of status for passenger vessels in range

Month	Underway	At Anchor	N.U.C.	Moored
March	40.22%	< 0.01%	0.01%	59.19%
April	9.64%	0.47%	0.01%	89.89%
May	10.32%	0.01%	0.01%	89.68%
June	20.88%	0.77%	0.01%	78.35%
July	64.51%	0.00%	0.00%	35.49%

Table A 14. Distribution of status for passenger vessels in range per period

Period	Underway	At Anchor	N.U.C.	Moored
Pre-lockdown	57.92%	0.00%	0.00%	42.08%
Lockdown	12.98%	0.15%	0.00%	86.68%
Home-quarantine	6.55%	0.00%	0.00%	93.45%
Post-lockdown	58.01%	0.61%	0.00%	41.38%

A3.3. Port calls in Barcelona

Number of calls at Barcelona for the 5-month period:

Table A 15. Comparison of calls at Barcelona from March to July over the last 5 years

Period	Cargo	Tankers	Passenger	Total
March to July 2016 – 2019	1849	426	1499	3774
March to July 2020	1483	393	747	2623

Table A 16. Monthly calls by ship type in 2020
Values in brackets are the difference over the 2016-2019 average for the same months

Month	**Cargo**	**Tankers**	**Passenger**	**Total**
March	315 (-15.8%)	90 (+10.4%)	203 (-16.1%)	608 (-12.9%)
April	276 (-23.8%)	81 (+2.2%)	93 (-65.4%)	450 (-36.7%)
May	287 (-24.6%)	72 (-18.9%)	96 (-67.1%)	455 (-40.3%)
June	294 (-17.4%)	73 (-19.8%)	128 (-60.7%)	495 (-35.9%)
July	311 (-17.3%)	77 (-9.9%)	227 (-38.8%)	615 (-26.1%)

Annex A4. Mathematical model calibration tables and plots

This annex presents more detailed data listed in tables and plots on the results obtained through the mathematical model for section 5.1.1 and 5.1.2. For the reader's guidance, the colors corresponding to the database and mathematical model are switched in the plots here presented.

A4.1. Fuel consumption

Plot and tables with detailed data on the fuel consumption computed through the database and mathematical model:

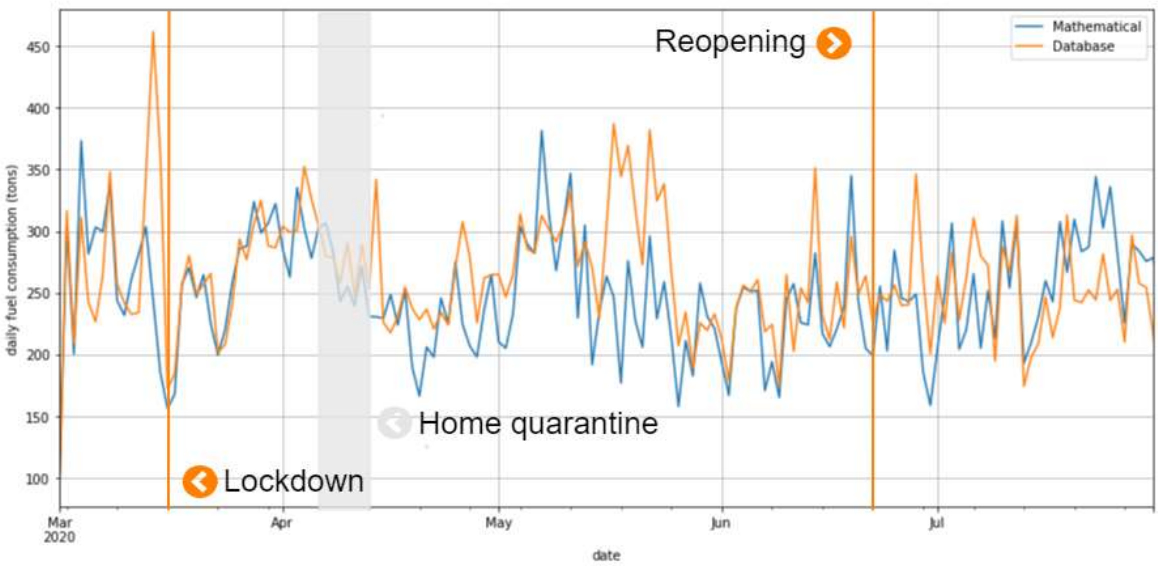

Figure A 1. Daily changes in fuel consumption from March 1 to July 31, 2020

Table A 17. Computed fuel consumption (tons) on monthly basis

Month	Database model (tons)	Mathematical model (tons)
March	8242	8071
April	8048	7457
May	8882	8223
June	7330	6851

July	7779	7797

Table A 18. Computed average fuel consumption (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	Database model	Mathematical model
Pre-lockdown	277 (+5.2%)	264 (+5.2%)
Lockdown	266 (+1.1%)	256 (+2.0%)
Home-quarantine	275 (+4.6%)	267 (+6.4%)
Post-lockdown	251 (-4.6%)	246 (-2.0%)

A4.2. Emissions

Carbon dioxide

Plot and tables with detailed data on the CO_2 emissions computed through the database and mathematical model:

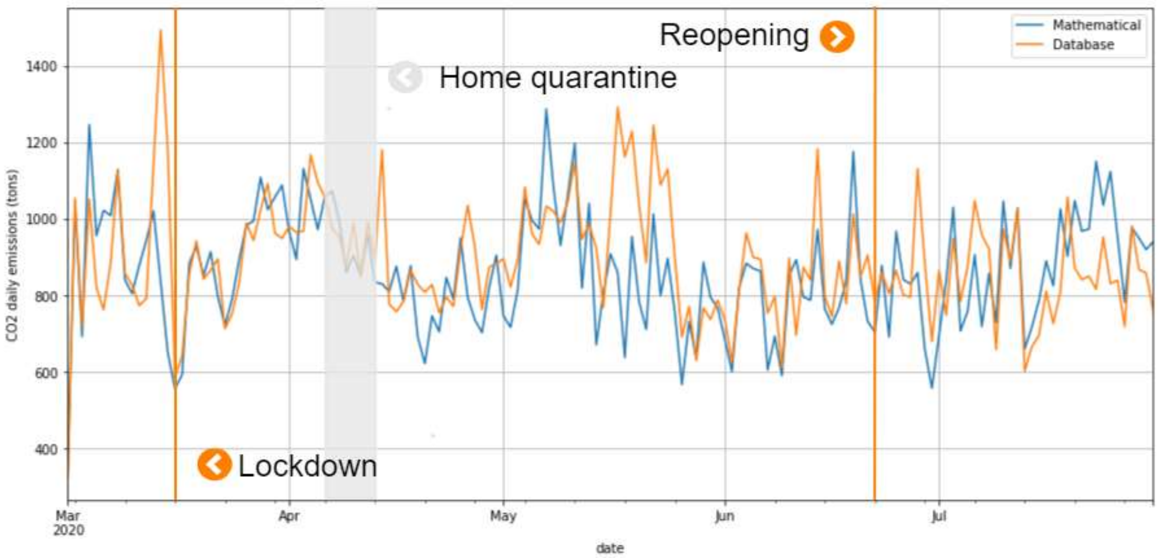

Figure A 2. Daily changes in CO2 emissions from March 1 to July 31, 2020

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Nàutica de Barcelona

Table A 19. Computed CO2 emissions (tons) on monthly basis

Month	Database model (tons)	Mathematical model (tons)
March	27716	27712
April	27329	26197
May	29828	27908
June	25184	23881
July	26262	27069

Table A 20. Computed average CO_2 emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	Database model (tons)	Mathematical model (tons)
Pre-lockdown	921 (+3.7%)	897 (+4.1%)
Lockdown	904 (+1.8%)	861 (-0.1%)
Home-quarantine	949 (+6.9%)	947 (+9.9%)
Post-lockdown	847 (-4.6%)	873 (+1.3%)

Oxides of sulfur

Plot and tables with detailed data on the SO_2 emissions computed through the database and mathematical model:

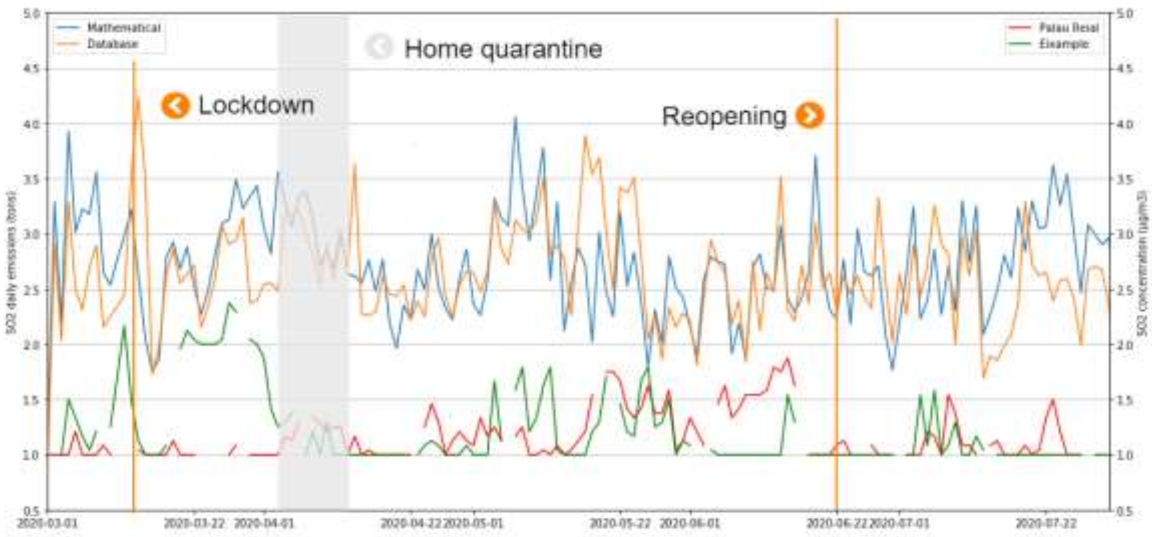

Figure A 3. Daily changes in SO2 emissions and concentrations from March 1 to July 31, 2020

Table A 21. Computed SO2 emissions (tons) on monthly basis

Month	Database model (tons)	Mathematical model (tons)
March	81	87
April	80	82
May	89	88
June	75	75
July	78	85

Table A 22. Computed average SO_2 emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	Database model (tons)	Mathematical model (tons)
Pre-lockdown	2.7 (+3.9%)	2.8 (+3.7%)
Lockdown	2.7 (+3.8%)	2.7 (+3.6%)
Home-quarantine	2.8 (+7.7%)	3.0 (+11.1%)
Post-lockdown	2.5 (-3.8%)	2.7 (-3.3%)

Oxides of nitrogen

Plot and tables with detailed data on the NO_X emissions computed through the database and mathematical model:

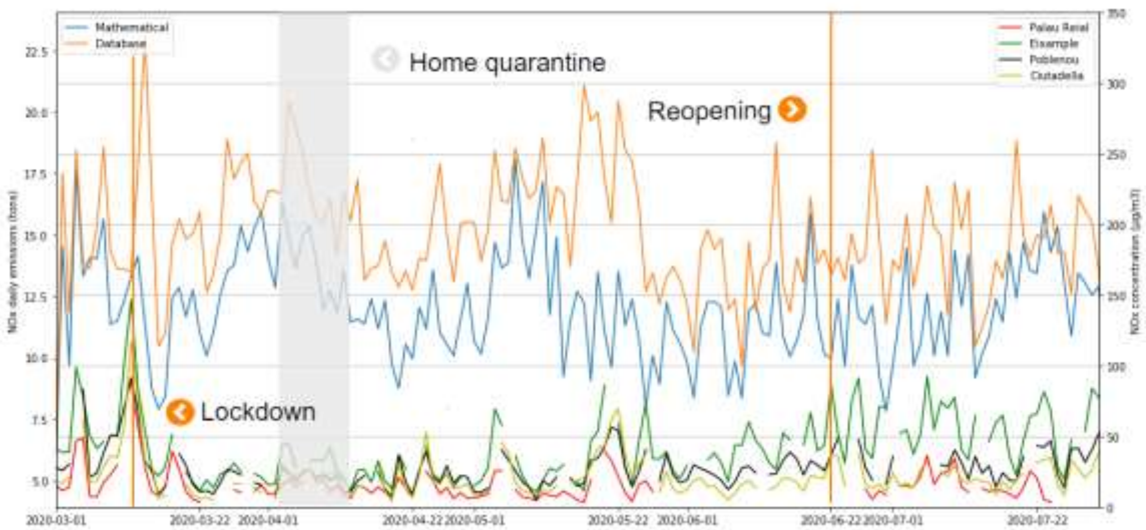

Figure A 4. Daily changes in NOX emissions and concentrations from March 1 to July 31, 2020

Table A 23. Computed NO_X emissions (tons) on monthly basis

Month	Database model (tons)	Mathematical model (tons)
March	470	421
April	466	417
May	509	456
June	413	370
July	451	404

Table A 24. Computed average NO_X emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	Database model (tons)	Mathematical model (tons)
Pre-lockdown	15.1 (-0.2%)	14.1 (+4.4%)
Lockdown	15.3 (+1.3%)	13.7 (+1.5%)
Home-quarantine	16.2 (+7.3%)	14.2 (+5.2%)
Post-lockdown	14.5 (-4.0%)	13.2 (-2.2%)

Particulate matter

Plot and tables with detailed data on the PM emissions computed through the database and mathematical model:

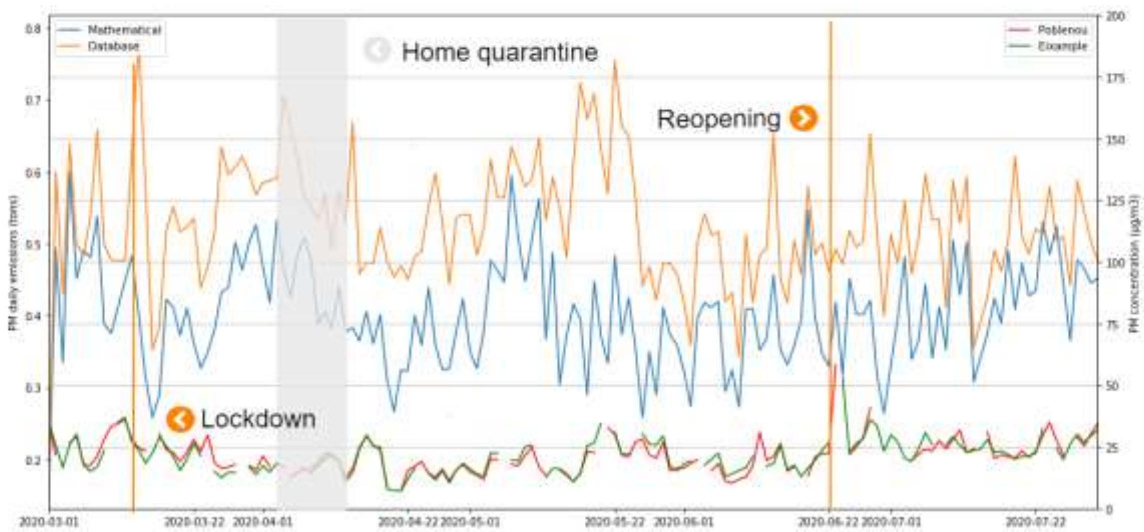

Figure A 5. Daily changes in PM emissions and concentrations from March 1 to July 31, 2020

Table A 25. Computed PM emissions (tons) on monthly basis

Month	Database model (tons)	Mathematical model (tons)
March	16.4	14.4
April	16.2	14.3
May	17.8	15.7
June	14.5	12.7
July	15.7	13.8

Table A 26. Computed average PM emissions (tons) per day and period
Values in brackets are the difference over the 5-monthly daily average

Period	Database model (tons)	Mathematical model (tons)
Pre-lockdown	0.53 (+1.9%)	0.46 (-2.2%)
Lockdown	0.53 (+1.9%)	0.47 (+0.4%)
Home-quarantine	0.55 (+5.8%)	0.48 (+2.1%)
Post-lockdown	0.51 (-1.9%)	0.46 (-2.5%)