Annex 1
R code

The logistic regression model has been created with the help of the Package NNET.

- **Description:** Software for feed-forward neural networks with a single hidden layer, and for multinomial log-linear models.
- **Last Version:** 7.3-12
- **Date:** 2016-02-02

In this annex, the code that was created for building the model, processing the data and analysing the results is included with explanatory comments for each section:

1.1 Logistic Regression code for the Point-by-Point Analysis

```
#===================INITIAL PACKAGE INSTALLATION======================
library(nnet)
library(data.table)
library(plyr)
library(xlsx)

#===================SET UP OF VARIABLES================================
#Upload of the dataframe obtained from the QGIS and pre-processing of the data

setwd("~/Desktop/Teruel/R_calculus")
X = read.table(file="dataframe.csv",sep=";",dec=",",
               header=TRUE,strip.white=TRUE)
#Upload also a guided dataframe that will be used in the future to store the results
x_1 = read.table(file="template_map.csv",sep=";",dec=",",
               header=TRUE,strip.white=TRUE)
#Measure of the number of rows and the names of the variables
colnames(X)
nrow(X)
TotalPointID <- length(unique(X$POINTID))


#==================MULTINOMIAL LOG MODEL (Category)==================
# Section containing the creation of the Multinomial Log Model

# In the "for" loop, the data is separated between those points valids for
# analysis and those that are rejected (points that have one category in all the
measurements)
Valid <- 0
```

```
Rejected <- 0
fitlist <- list()
p <- 1

for (i in 1:3928) {
  x.sub <- subset(X, POINTID==i)
  if (length(unique(x.sub$Category))>1) {
    Valid <-rbind.data.frame(Valid, x.sub)
    fitlist[[p]] <- multinom(Category ~ HS_TR + SLR, x.sub, maxit=3000)
    p <- p + 1
} else if (length(unique(x.sub$Category))==1) {
    Rejected <-rbind.data.frame(Rejected, x.sub)
}
}
Valid <- Valid[-c(1), ]
Rejected <- Rejected[-c(1), ]

#The total of points valid for the multinomial regression are obtained counting the
#total amount of Points of the dataframe "Valid" - 1 because of the initial 0

Valid_Points <- length(unique(Valid$POINTID))
Non_Valid_Points <- length(unique(Rejected$POINTID))
ListValid_Points <- as.data.frame(unique(Valid$POINTID))
names(ListValid_Points)[1]<-paste("POINTID")

#===================PREDICTIONS=======================================
#Definition of 5 categories of changes in Sea Lelvel Rise (SLR) and Wave Return Perio
(Hs_Tr):

d2<-data.frame("HS_TR"=c(10,25,50,75,100),"SLR"=c(0,0,0,0.6,0.6))

vallist_Pred <- list()
for (m in 1:(p-1)) {
  vallist_Pred[[m]] <-(predict(fitlist[[m]],d2))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_c <- as.data.frame(vallist_Pred)

#Since the initial dataframe considers each point in a row, the results are transposed
results_r <- t(results_c)
results_r  <- cbind(ListValid_Points, results_r )
names(results_r)[2]<-paste("PREDICTION1")
names(results_r)[3]<-paste("PREDICTION2")
names(results_r)[4]<-paste("PREDICTION3")
names(results_r)[5]<-paste("PREDICTION4")
names(results_r)[6]<-paste("PREDICTION5")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Finalist <- merge(x_1, results_r, by="POINTID", all=TRUE)

#Exporting the dataframe obtaied to post-process and analyse the results in QGIS
```

```
write.xlsx(Finalist, file = "Finalist.xlsx")

#===================COMPARISON OF RESULTS==================================
# Creation of the Histograms of the resulting categories for the prediction of 5
scenarios

par(mgp=c(2,1,0))
plot(results_r$PREDICTION1, main="Histogram for HS_TR:10 and SLR:0", xlab="Category",
ylab="Frequency",
     ylim=c(0,3000))
box()
plot(results_r$PREDICTION2, main="Histogram for HS_TR:25 and SLR:0", xlab="Category",
ylab="Frequency",
     ylim=c(0,3000))
box()
plot(results_r$PREDICTION3, main="Histogram for HS_TR:50 and SLR:0", xlab="Category",
ylab="Frequency",
     ylim=c(0,3000))
box()
plot(results_r$PREDICTION4, main="Histogram for HS_TR:75 and SLR:0,6", xlab="Category",
ylab="Frequency",
     ylim=c(0,3000))
box()
plot(results_r$PREDICTION5, main="Histogram for HS_TR:100 and SLR:0,6", xlab="Category",
ylab="Frequency",
     ylim=c(0,3000))
box()

#===================MULTINOMIAL PROBABILITIES==============================
# Section dealing with the calculation of the expected probabilities of categories

              #------------Define Probability of Prediction 1------------:

d3<-data.frame("HS_TR"=c(10),"SLR"=c(0))
vallist_Prob1 <- list()
for (m in 1:(p-1)) {
  vallist_Prob1[[m]] <-(predict(fitlist[[m]],d3, type="probs"))
}

Table1 <-data.frame("POINTID"=c(1,2,3,4))
Prob_Pred1 <-data.frame("POINTID"=c(1,2,3,4))
for (m in 1:(p-1)){
J1 <- as.data.frame(vallist_Prob1[[m]])
Tables1 <- merge(as.data.frame(Table1), as.data.frame(J1), by='row.names', all=TRUE)
Tables1 <- Tables1[ ,-c(1:2)]
Prob_Pred1 <-cbind.data.frame(Prob_Pred1,Tables1 )
}
Prob_Pred1 <- Prob_Pred1[ ,-c(1)]
Prob_Pred1 <- t(Prob_Pred1)
Prob_Pred1 <- cbind(Prediction=results_r$PREDICTION1, Prob_Pred1 )
Prob_Pred1 <- cbind(POINTID=ListValid_Points , Prob_Pred1 )

write.xlsx(Prob_Pred1, file = "Prob_Pred1.xlsx")
```

```
#---------------------------------------------------------------#

   #------------Define Probability of Prediction 2------------:

d4<-data.frame("HS_TR"=c(25),"SLR"=c(0))
vallist_Prob2 <- list()
for (m in 1:(p-1)) {
  vallist_Prob2[[m]] <-(predict(fitlist[[m]],d4, type="probs"))
}

Table2 <-data.frame("POINTID"=c(1,2,3,4))
Prob_Pred2 <-data.frame("POINTID"=c(1,2,3,4))
for (m in 1:(p-1)){
  J2 <- as.data.frame(vallist_Prob2[[m]])
  Tables2 <- merge(as.data.frame(Table2), as.data.frame(J2), by='row.names', all=TRUE)
  Tables2 <- Tables2[ ,-c(1:2)]
  Prob_Pred2 <-cbind.data.frame(Prob_Pred2,Tables2 )
}
Prob_Pred2 <- Prob_Pred2[ ,-c(1)]
Prob_Pred2 <- t(Prob_Pred2)
Prob_Pred2 <- cbind(Prediction=results_r$PREDICTION2, Prob_Pred2 )
Prob_Pred2 <- cbind(POINTID=ListValid_Points , Prob_Pred2 )

write.xlsx(Prob_Pred2, file = "Prob_Pred2.xlsx")

#---------------------------------------------------------------#

   #------------Define Probability of Prediction 3------------:

d5<-data.frame("HS_TR"=c(50),"SLR"=c(0))
vallist_Prob3 <- list()
for (m in 1:(p-1)) {
  vallist_Prob3[[m]] <-(predict(fitlist[[m]],d5, type="probs"))
}

Table3 <-data.frame("POINTID"=c(1,2,3,4))
Prob_Pred3 <-data.frame("POINTID"=c(1,2,3,4))
for (m in 1:(p-1)){
  J3 <- as.data.frame(vallist_Prob1[[m]])
  Tables3 <- merge(as.data.frame(Table3), as.data.frame(J3), by='row.names', all=TRUE)
  Tables3 <- Tables3[ ,-c(1:2)]
  Prob_Pred3 <-cbind.data.frame(Prob_Pred3,Tables3 )
}
Prob_Pred3 <- Prob_Pred3[ ,-c(1)]
Prob_Pred3 <- t(Prob_Pred3)
Prob_Pred3 <- cbind(Prediction=results_r$PREDICTION3, Prob_Pred3 )
Prob_Pred3 <- cbind(POINTID=ListValid_Points , Prob_Pred3 )

write.xlsx(Prob_Pred3, file = "Prob_Pred3.xlsx")

#---------------------------------------------------------------#

   #------------Define Probability of Prediction 4------------:
```

```
d6<-data.frame("HS_TR"=c(75),"SLR"=c(0.6))
vallist_Prob4 <- list()
for (m in 1:(p-1)) {
  vallist_Prob4[[m]] <-(predict(fitlist[[m]],d6, type="probs"))
}

Table4 <-data.frame("POINTID"=c(1,2,3,4))
Prob_Pred4 <-data.frame("POINTID"=c(1,2,3,4))
for (m in 1:(p-1)){
  J4 <- as.data.frame(vallist_Prob4[[m]])
  Tables4 <- merge(as.data.frame(Table4), as.data.frame(J4), by='row.names', all=TRUE)
  Tables4 <- Tables4[ ,-c(1:2)]
  Prob_Pred4 <-cbind.data.frame(Prob_Pred4,Tables4 )
}
Prob_Pred4 <- Prob_Pred4[ ,-c(1)]
Prob_Pred4 <- t(Prob_Pred4)
Prob_Pred4 <- cbind(Prediction=results_r$PREDICTION4, Prob_Pred4 )
Prob_Pred4 <- cbind(POINTID=ListValid_Points , Prob_Pred4 )

write.xlsx(Prob_Pred4, file = "Prob_Pred4.xlsx")

          #-------------------------------------------------------------------#

             #-------------Define Probability of Prediction 5-------------:

d7<-data.frame("HS_TR"=c(100),"SLR"=c(0.6))
vallist_Prob5 <- list()
for (m in 1:(p-1)) {
  vallist_Prob5[[m]] <-(predict(fitlist[[m]],d7, type="probs"))
}

Table5 <-data.frame("POINTID"=c(1,2,3,4))
Prob_Pred5 <-data.frame("POINTID"=c(1,2,3,4))
for (m in 1:(p-1)){
  J5 <- as.data.frame(vallist_Prob5[[m]])
  Tables5 <- merge(as.data.frame(Table5), as.data.frame(J5), by='row.names', all=TRUE)
  Tables5 <- Tables5[ ,-c(1:2)]
  Prob_Pred5 <-cbind.data.frame(Prob_Pred5,Tables5 )
}
Prob_Pred5 <- Prob_Pred5[ ,-c(1)]
Prob_Pred5 <- t(Prob_Pred5)
Prob_Pred5 <- cbind(Prediction=results_r$PREDICTION5, Prob_Pred5 )
Prob_Pred5 <- cbind(POINTID=ListValid_Points , Prob_Pred5 )

write.xlsx(Prob_Pred5, file = "Prob_Pred5.xlsx")


#========PROBABILITY OF CATEGORY DEPENDANT ON THE PREDICTION=============
# Storage of the probability results calculations as a dataframe

prob_1 <- as.data.frame(colMeans(Prob_Pred1[,c(3:6)], na.rm=TRUE))
prob_2 <- as.data.frame(colMeans(Prob_Pred2[,c(3:6)], na.rm=TRUE))
prob_3 <- as.data.frame(colMeans(Prob_Pred3[,c(3:6)], na.rm=TRUE))
prob_4 <- as.data.frame(colMeans(Prob_Pred4[,c(3:6)], na.rm=TRUE))
```

```
prob_5 <- as.data.frame(colMeans(Prob_Pred5[,c(3:6)], na.rm=TRUE))

# Merge all the results together
prob_df <- join_all(list(prob_1,prob_2,prob_3,prob_4,prob_5))

#===================SENSITIVITY ANALYSIS==================================
# Application of the model with the same procedure as before but with more measurements

d8<-
data.frame("HS_TR"=c(10,10,10,10,10,10,10,10,10),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens[[m]] <-(predict(fitlist[[m]],d8))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_sensibility <- as.data.frame(vallist_Pred_Sens)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility <- t(results_sensibility)
results_sensibility  <- cbind(ListValid_Points, results_sensibility )
names(results_sensibility)[2]<-paste("SLR_0.05")
names(results_sensibility)[3]<-paste("SLR_0.10")
names(results_sensibility)[4]<-paste("SLR_0.15")
names(results_sensibility)[5]<-paste("SLR_0.20")
names(results_sensibility)[6]<-paste("SLR_0.25")
names(results_sensibility)[7]<-paste("SLR_0.30")
names(results_sensibility)[8]<-paste("SLR_0.35")
names(results_sensibility)[9]<-paste("SLR_0.40")
names(results_sensibility)[10]<-paste("SLR_0.50")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Sensibility_Analysis <- merge(x_1, results_sensibility, by="POINTID", all=TRUE)

#Exporting the dataframe obtaied to post-process and analyse the results in QGIS
write.xlsx(Sensibility_Analysis, file = "Sensibility_Analysis.xlsx")

#================== SENSITIVITY ANALYSIS (Tr_Hs:15) ==================

d9<-
data.frame("HS_TR"=c(15,15,15,15,15,15,15,15,15),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens_15 <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens_15[[m]] <-(predict(fitlist[[m]],d9))
}

#By doing this, the results of the 5 predictions are obtained in columns.
```

```
results_sensibility_15 <- as.data.frame(vallist_Pred_Sens_15)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility_15 <- t(results_sensibility_15)
results_sensibility_15  <- cbind(ListValid_Points, results_sensibility_15 )
names(results_sensibility_15)[2]<-paste("SLR_0.05")
names(results_sensibility_15)[3]<-paste("SLR_0.10")
names(results_sensibility_15)[4]<-paste("SLR_0.15")
names(results_sensibility_15)[5]<-paste("SLR_0.20")
names(results_sensibility_15)[6]<-paste("SLR_0.25")
names(results_sensibility_15)[7]<-paste("SLR_0.30")
names(results_sensibility_15)[8]<-paste("SLR_0.35")
names(results_sensibility_15)[9]<-paste("SLR_0.40")
names(results_sensibility_15)[10]<-paste("SLR_0.50")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Sensibility_Analysis_15 <- merge(x_1, results_sensibility_15, by="POINTID", all=TRUE)
write.xlsx(Sensibility_Analysis_15, file = "Sensibility_Analysis_15.xlsx")

#================== SENSITIVITY ANALYSIS (Tr_Hs:20) ==================

d10<-
data.frame("HS_TR"=c(20,20,20,20,20,20,20,20,20),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens_20 <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens_20[[m]] <-(predict(fitlist[[m]],d10))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_sensibility_20 <- as.data.frame(vallist_Pred_Sens_20)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility_20 <- t(results_sensibility_20)
results_sensibility_20  <- cbind(ListValid_Points, results_sensibility_20 )
names(results_sensibility_20)[2]<-paste("SLR_0.05")
names(results_sensibility_20)[3]<-paste("SLR_0.10")
names(results_sensibility_20)[4]<-paste("SLR_0.15")
names(results_sensibility_20)[5]<-paste("SLR_0.20")
names(results_sensibility_20)[6]<-paste("SLR_0.25")
names(results_sensibility_20)[7]<-paste("SLR_0.30")
names(results_sensibility_20)[8]<-paste("SLR_0.35")
names(results_sensibility_20)[9]<-paste("SLR_0.40")
names(results_sensibility_20)[10]<-paste("SLR_0.50")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Sensibility_Analysis_20 <- merge(x_1, results_sensibility_20, by="POINTID", all=TRUE)
```

```
write.xlsx(Sensibility_Analysis_20, file = "Sensibility_Analysis_20.xlsx")

#================== SENSITIVITY ANALYSIS (Tr_Hs:50) ==================

d11<-
data.frame("HS_TR"=c(50,50,50,50,50,50,50,50,50),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens_50 <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens_50[[m]] <-(predict(fitlist[[m]],d11))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_sensibility_50 <- as.data.frame(vallist_Pred_Sens_50)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility_50 <- t(results_sensibility_50)
results_sensibility_50  <- cbind(ListValid_Points, results_sensibility_50 )
names(results_sensibility_50)[2]<-paste("SLR_0.05")
names(results_sensibility_50)[3]<-paste("SLR_0.10")
names(results_sensibility_50)[4]<-paste("SLR_0.15")
names(results_sensibility_50)[5]<-paste("SLR_0.20")
names(results_sensibility_50)[6]<-paste("SLR_0.25")
names(results_sensibility_50)[7]<-paste("SLR_0.30")
names(results_sensibility_50)[8]<-paste("SLR_0.35")
names(results_sensibility_50)[9]<-paste("SLR_0.40")
names(results_sensibility_50)[10]<-paste("SLR_0.50")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Sensibility_Analysis_50 <- merge(x_1, results_sensibility_50, by="POINTID", all=TRUE)
write.xlsx(Sensibility_Analysis_50, file = "Sensibility_Analysis_50.xlsx")

#================== SENSITIVITY ANALYSIS (Tr_Hs:75) ==================

d12<-
data.frame("HS_TR"=c(75,75,75,75,75,75,75,75,75),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens_75 <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens_75[[m]] <-(predict(fitlist[[m]],d12))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_sensibility_75 <- as.data.frame(vallist_Pred_Sens_75)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility_75 <- t(results_sensibility_75)
results_sensibility_75  <- cbind(ListValid_Points, results_sensibility_75 )
```

```
names(results_sensibility_75)[2]<-paste("SLR_0.05")
names(results_sensibility_75)[3]<-paste("SLR_0.10")
names(results_sensibility_75)[4]<-paste("SLR_0.15")
names(results_sensibility_75)[5]<-paste("SLR_0.20")
names(results_sensibility_75)[6]<-paste("SLR_0.25")
names(results_sensibility_75)[7]<-paste("SLR_0.30")
names(results_sensibility_75)[8]<-paste("SLR_0.35")
names(results_sensibility_75)[9]<-paste("SLR_0.40")
names(results_sensibility_75)[10]<-paste("SLR_0.50")

#Merge the results with the template map (to store the results for both the "Valid"
points
# and include a "NULL" to the points that were "Rejected")
Sensibility_Analysis_75 <- merge(x_1, results_sensibility_75, by="POINTID", all=TRUE)
write.xlsx(Sensibility_Analysis_75, file = "Sensibility_Analysis_75.xlsx")

#================== SENSITIVITY ANALYSIS (Tr_Hs:25) ==================

d13<-
data.frame("HS_TR"=c(25,25,25,25,25,25,25,25,25),"SLR"=c(0.05,0.10,0.15,0.20,0.25,0.30,0.
35,0.40,0.50))

vallist_Pred_Sens_25 <- list()
for (m in 1:(p-1)) {
  vallist_Pred_Sens_25[[m]] <-(predict(fitlist[[m]],d13))
}

#By doing this, the results of the 5 predictions are obtained in columns.
results_sensibility_25 <- as.data.frame(vallist_Pred_Sens_25)

#Since the initial datafram considers each point in a row, we will transpose the results
matric
results_sensibility_25 <- t(results_sensibility_25)
results_sensibility_25  <- cbind(ListValid_Points, results_sensibility_25 )
names(results_sensibility_25)[2]<-paste("SLR_0.05")
names(results_sensibility_25)[3]<-paste("SLR_0.10")
names(results_sensibility_25)[4]<-paste("SLR_0.15")
names(results_sensibility_25)[5]<-paste("SLR_0.20")
names(results_sensibility_25)[6]<-paste("SLR_0.25")
names(results_sensibility_25)[7]<-paste("SLR_0.30")
names(results_sensibility_25)[8]<-paste("SLR_0.35")
names(results_sensibility_25)[9]<-paste("SLR_0.40")
names(results_sensibility_25)[10]<-paste("SLR_0.50")

#Merge the results with a template map (important colums only the POSITION X-Y and the
POINT ID to
#know the location of X and Y for the QGIS). The SLR and HS_TR will be deleted
afterwards.
Sensibility_Analysis_25 <- merge(x_1, results_sensibility_25, by="POINTID", all=TRUE)
write.xlsx(Sensibility_Analysis_25, file = "Sensibility_Analysis_25.xlsx")
```

END