Title: Simulation of the dynamics of a drug in farm animals

Author: Alberto Guillén Jiménez

Director: Pau Fonseca i Casas

Departament: Department of Statistics and Operations Research

Academic period: 2014/15
Simulation of the dynamics of a drug in farm animals

Alberto Guillén Jiménez

Director: Pau Fonseca i Casas
Summary

This study aims to analyze the behavior of a drug in farm animals. The study is part of a research project conducted by researchers at the Veterinarian Faculty of the Universitat Autonoma of Barcelona (UAB), whose purpose is to estimate how much drug would be present in the blood of the animals when a given amount of drug is added into the food.

A statistical analysis of the feeding patterns of the animals is performed in order to gain insight into the behavior of the animals, based on experimental obtained in pig farms located in Catalonia. The data was collected by automatic feeding devices that register the times the animals enter and leave the feeding trough, by means of an electronic chip attached to them, and that also register the weight of the food available in the feeding trough at those particular times. From the difference between the amounts of food present in the feeding trough at the times of the entry and the exit of the animal, the animal’s intake of food can be inferred. Different amounts of drug in the food, layouts for a farm or distributions of animals with different characteristics can then by analyzed by means of simulation to adapt more intelligently to the needs of an efficient and healthy farming, without needing to test empirically the different designs.

Several factors are involved in the process that results in varying levels of drug in the animals’ metabolism over time: the amount of drug provided in the food, the feeding mechanism, the competition among individuals, the animal’s distinctive behavior, the animal’s physiology and the absorption characteristics of the drug are some of these factors. The competition among animals in the boxes is the main factor that causes uneven and irregular consumption of food, and therefore drug, across animals and over time. The animals’ feeding patterns and the absorption of the drug in the body of the animals can be re-created in a simulation environment by means of a multi-agent system.

The data used in this study contained noise and errors. After addressing this problem, some empirical evidence was assessed from the data. For example, the circadian rhythm of the animals, the different types of feeding habits across animals and the empirical segmentation and distribution of those feeding patterns. Theoretical assumptions on the data did, in some occasions, fail under empirical testing. This occurred either because the data don’t support a priori beliefs on the expected behavior of the animals, for example the relation between animal weights and feeding behavior; or because the signals available in the data are limited and it is not possible to infer some information with enough accuracy from the experiment, as is for instance the case of establishing the hierarchical dominance of the individuals from their indirectly observed interactions, which results in information that has no strong relation with their feeding habits that would be expected otherwise.

A logical simulation model was established and updated on an ongoing basis, as new information was available, and the stochastic variables in the model were modelled statistically. The results from the simulation were not known at the time of writing this document, but several conclusions could be drawn at the time. Most important of all was the difficulty to obtain the intended quantitative relationship between observable biometric information – of which only the animal weights was available – and behavioural characteristics, in which case the conclusions from this simulation model could be easily extrapolated to other experimental conditions based on this observable biometric information. Other, more behavioural-based approaches might be attempted for this type of simulation model, as is done here, assuming that the observed animal behavior remains homogenous and pervasive under changing environmental conditions. Also in this study, possible improvements are suggested to introduce further complexity in future developments of the project.
The data cleansing and statistical analysis was performed in the R language. The graphs in this document are also produced with this statistical programming language and the code used can be found in the appendix. The implementation of the simulation model was automated by the software SDLPS, developed by Pau Fonseca i Casas at the Universitat Politecnica de Catalunya. With this software, the logic of the model can be represented in SDL diagrams in Microsoft Visio and C code is automatically generated, compiled and verified by the software. The SDL diagrams are available in the appendix. The code implementing the variables that are modeled statistically has been written in C and is available in the present document.

**Keywords:** Agent-based, simulation, stochastic systems  
**MSC2000:** 92-02
Abstract

The dynamics of a drug in the body of a farm animal is an important piece of information in order to determine, among other things, if the amount of drug (for example antibiotics) that they are supplying the animals are sufficient in order to make the effect they need to do. To develop the model we have data related to the intake of food of each animal allowing the creation of a simulation model to analyze this dynamics. The results allow us to adapt in a more intelligent way the amount of medication, reverting in economic savings and an increase in food safety.

Keywords: Agent-based, simulation, stochastic systems
MSC2000: 92-02
Index

Summary .......................................................................................................................... 1
Abstract ............................................................................................................................ 3
Index ................................................................................................................................. 4
Preface ............................................................................................................................. 5
Introduction ...................................................................................................................... 6
Part I – Data analysis ...................................................................................................... 7
  Description of the data ................................................................................................. 8
  Data cleansing .............................................................................................................. 13
  Data transformations ................................................................................................. 19
  Exploratory analysis ................................................................................................. 24
Part II – Simulation ........................................................................................................ 36
  Model ............................................................................................................................ 37
  Implementation ............................................................................................................ 40
  Distribution of the random variables ....................................................................... 42
  Implementation of the random variables .................................................................... 65
Part III – Conclusions .................................................................................................... 69
  Main findings from this study ................................................................................... 70
  Suggested extensions for the study ............................................................................ 71
Appendix ........................................................................................................................ 73
  A) R code .................................................................................................................... 74
  B) SDL diagrams ....................................................................................................... 88
Bibliography .................................................................................................................... 95
Preface

This work was made as a final project for the master degree in statistical science and operations research as of the plan of 1999. It aims to apply the statistical techniques and skills acquired during the realization of this degree, and it is measured on an expected workload of 15 ECTS.

This study aims to simulate the behavior of a drug in farm animals, for which limited data is available. The drug is provided in the food for the animals, but only experimental data related to food intake over time for a set of pigs in a farm is available for the design of the simulation model. The main purpose of this study is to increase the understanding of the process of intake and absorption of drug in farm animals and to establish a framework that could be used in future applications addressing how to design farms in a more intelligent way or how to supply the necessary amount of drug that is efficiently protecting all the animals from disease while costs are minimized. This study is part of a research project conducted by researchers at the Veterinarian Faculty of the Universitat Autonoma of Barcelona (UAB) and that is expected to be published in several scientific journals.

The data handling, statistical analysis, modeling and simulation carried out in this study cover all the processes involved in the dynamics of a drug in farm animals, from its intake to its absorption for given configurations of a farm and animal characteristics, but they focus to a higher extent, due to the need to limit the workload of this study, on an important part of the process: the varying food intake of the animals, for which data is provided. It doesn’t aim to model the process of drug absorption in the presence of disease or other circumstances; nor does it address the optimization of the required supply of drug.

The basis of the correct understanding of the behavior of the animals and the possibility to simulate the intake and absorption dynamics of the drug in the animals has large industrial benefits for the safety of the product, a more intelligent management of farms and cost reduction. The present work aims at understanding and modeling the inherent behavior of pigs on an agent-based approach to simulate the dynamics of drug intake and absorption in farm animals.
Introduction

This study aims to analyze the behavior of a drug in farm animals, for which limited data is available. The purpose of the investigation is to estimate how much drug would be present in the blood of the different animals over time when a given amount of drug is added into the food. Since experimental data is scarce and costly and industry regulations limit the possibility of direct experimentation on animals, a simulation study is hereby conducted to analyze the dynamics of ingestion and absorption of the drug by the animals over time. Different amounts of drug in the food, layouts for a farm or distributions of animals with different characteristics can then be analyzed by means of simulation to adapt more intelligently to the needs of an efficient and healthy farming, without having to test empirically the different designs.

Several factors are involved in the process that results in varying levels of drug in the animals’ metabolism over time: the amount of drug provided in the food, the feeding mechanism, the competition among individuals, the animal’s distinctive behavior, the animal’s physiology and the absorption characteristics of the drug are some of these factors. The competition among animals in the boxes is the main factor that causes uneven and irregular consumption of food, and therefore drug, across animals and over time. The animals’ feeding patterns and the absorption of the drug in the body of the animals can be re-created in a simulation environment through a multi-agent system.

Several academic studies address exclusively the general absorption dynamics of a drug, the physiological characteristics of different races of pigs or the hierarchical dominance structures among different animal species. Fernández Capo (2002), presented in his PhD dissertation a survey of what factors drive the feeding behavior of a group of pigs sharing space in a farm. However, no study is known to us on what the dynamics of drug absorption in a pig farm are. The choice of a simulation approach to analyze the animals’ feeding behavior and the absorption of drug in their bodies allows us to obtain fast valuable data which would be otherwise costly to collect in an experimental set-up and which can be used to replicate the expected dynamics of the absorption of the drug in a group of pigs adapted to different types of farm configuration and supplied with different quantities of drug.

We base our study on limited experimental data: weights and intake. In the case of the animal weights, the data was collected between January and March 2008; and in the case of feeding data, the data was collected from December 2008 through April 2009. The farms are located in Catalonia and the data was collected by automatic feeding devices that register the times the animals enter and leave the feeding trough by means of an electronic chip attached to them, and that also register the weight of the food available in the feeding trough at those particular times. From the difference between the amounts of food present in the feeding trough at the times of entry and exit, the animal’s intake can be inferred. In this experiment, groups of 11 to 21 animals shared space in 24 isolated boxes and in each box only one animal could be in the feeding trough at a time.

In order to determine the most adequate simulation model, several challenges are faced: the animals’ behavior and interaction with other animals must be inferred from the data, and the relevant variables that determine their feeding patterns must be defined and modeled; no less challenging is the need to structure and cleanse the data, as it contains a fair amount of noise and error.

The structure of the document is as follows: first the data is presented and the empirical evidence on the pigs’ behavior is assessed; then, variables that determine the feeding habits of the animals are modelled statistically and integrated into a simple multi-agent model. Main conclusions and suggested extensions for study are presented at the end of the document.
Part I – Data analysis

The experimental data has been obtained from automated feeding devices placed in several farms, and thus, manual control over the quality of the data has been limited. The instrumental measurements don’t capture, for example, information related to addition of food in the interval of time when some animal is occupying the trough or any other phenomenon rather than the time and amount of food available at each event triggered by the move of the animal. In addition, the erratic movement of the animals at the proximity of the device may yield noisy signals when the animal only intends to enter the feeding trough once, and fights between animals for the occupation of the space might not be available in the observed data, since only the animal that occupies the space is being registered by the device.

The data from the experiment has two characteristics that condition the analysis: on the one hand, the high granularity of the registration of events by the automated device provides long time series of discrete events in the order of seconds and extending throughout several months, and the number of boxes and animals is reasonably high, so that sample size is not a significant problem; on the other hand, only data related to these events is available and no other information is directly available; for example, the hierarchy established between animals shortly after these are introduced into the box would be better observed from the fights between animals (Fernández Capo, 2002) by human observation. These fights may not happen only in the moments when two animals want to occupy simultaneously the feeding trough. Observation of group hierarchy is therefore difficult.

In order to extract signal from the noise and obtain meaningful information, the data must be filtered and cleansed and variables must be derived from this minimal but extensive dataset, in order to identify and quantify the factors that drive the feeding behaviour of the pigs. Empirical evidence will determine how the system could be modelled and to what extent the characteristics of the system can be generalized and extrapolated to other environments.
**Description of the data**

The data available from the experiment consists of:

- A time series dataset on food intakes by animal, which includes: (1) the times of entry and exit by the animal, (2) the animal’s id, registered from the electronic chips attached to the animals, when they enter and leave the feeding trough, and (3) the quantity of food available at the times when the animals enter and exit, which is used to infer the intake of the animal during that period.

- A dataset with the weights of some of the animals at three different measurement dates, which precede the time period when the data on intakes was collected.

The animals have each an electronic chip with an identification number and belong almost uniquely to a box where they share space with other animals. The exception are 15 animals which were relocated between boxes during the experiment. This relocation involved only one change of box in all 15 cases. There are 24 boxes and 13 animals on average in each box. There is a total of 291 animals. Also, the chips of some animals were replaced during the experiment, but this information is known and can be corrected in the data.

The data on food intakes was collected between 2008-12-28 and 2009-04-19, with several boxes missing complete information during the period. The data are discrete and sampled at irregular points in time whenever the events of entry and exit occurred.

Below are some statistics by box on the dataset with consumptions, showing the number of individuals, the number of registered events ("occupancy events") and the time period when data is available.

<table>
<thead>
<tr>
<th>box</th>
<th>nr_ids</th>
<th>nr_records</th>
<th>start</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14</td>
<td>17306</td>
<td>2009-01-07</td>
<td>23:57:33</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>15001</td>
<td>2009-01-07</td>
<td>23:55:58</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>17149</td>
<td>2009-01-07</td>
<td>23:00:45</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>16519</td>
<td>2009-01-07</td>
<td>22:56:01</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>13142</td>
<td>2009-01-05</td>
<td>21:46:59</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>22929</td>
<td>2009-01-08</td>
<td>20:06:20</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>16282</td>
<td>2009-01-08</td>
<td>20:05:16</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>15951</td>
<td>2009-01-08</td>
<td>21:33:05</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>15213</td>
<td>2009-01-08</td>
<td>00:12:24</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>19551</td>
<td>2009-01-08</td>
<td>23:53:29</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>18551</td>
<td>2008-12-28</td>
<td>01:00:28</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>19865</td>
<td>2008-12-28</td>
<td>10:49:49</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>23301</td>
<td>2008-12-28</td>
<td>21:31:52</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>27342</td>
<td>2008-12-28</td>
<td>22:38:02</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>20236</td>
<td>2008-12-28</td>
<td>00:02:25</td>
</tr>
<tr>
<td>17</td>
<td>12</td>
<td>20304</td>
<td>2008-12-28</td>
<td>00:04:47</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>20304</td>
<td>2008-12-28</td>
<td>16:28:06</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>22341</td>
<td>2008-12-28</td>
<td>23:01:54</td>
</tr>
<tr>
<td>20</td>
<td>11</td>
<td>24333</td>
<td>2008-12-28</td>
<td>23:02:24</td>
</tr>
<tr>
<td>21</td>
<td>13</td>
<td>24333</td>
<td>2008-12-28</td>
<td>11:10:26</td>
</tr>
<tr>
<td>22</td>
<td>11</td>
<td>18478</td>
<td>2008-12-28</td>
<td>09:59:15</td>
</tr>
<tr>
<td>23</td>
<td>13</td>
<td>19393</td>
<td>2008-12-28</td>
<td>16:12:35</td>
</tr>
<tr>
<td>24</td>
<td>12</td>
<td>23814</td>
<td>2008-12-28</td>
<td>07:01:33</td>
</tr>
<tr>
<td>25</td>
<td>21</td>
<td>22755</td>
<td>2009-01-07</td>
<td>25:52:10</td>
</tr>
<tr>
<td>26</td>
<td>13</td>
<td>20585</td>
<td>2009-01-07</td>
<td>00:10:23</td>
</tr>
</tbody>
</table>

No more than one individual is ever present in the time series between each entry time and exit time (i.e., there are no overlaps in time between occupancy events in a box), indicating that only one animal can occupy the feeding space at a time.

Following is a sample of the dataset on consumption.
In figure Figure 1 a short time interval for one box is visualized. In this figure, an event is defined as the moment an animal enters or leaves the trough, meaning that an occupancy event is composed of to such events; they are denoted as “in” and “out” and each colour corresponds to an animal. Each entry is denoted by a circle, whereas each exit is denoted by a triangle. The difference in the amount of food in the trough between entry and exit is the estimated intake. We notice that some individuals don’t occupy the trough continuously and return irregularly to eat, and that there is a linear decay pattern in the amount of food until the trough is refilled, which means that the animals eat at fairly constant pace while they occupy the trough.

<table>
<thead>
<tr>
<th>id</th>
<th>box</th>
<th>time.in</th>
<th>time.out</th>
<th>duration</th>
<th>food.in</th>
<th>food.out</th>
<th>consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81791999</td>
<td>2009-01-07 12:23:39</td>
<td>2009-01-07 12:23:42</td>
<td>3</td>
<td>10.45</td>
<td>10.44</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>81791999</td>
<td>2009-01-07 12:38:43</td>
<td>2009-01-07 12:38:44</td>
<td>1</td>
<td>10.45</td>
<td>10.44</td>
<td>0.01</td>
</tr>
<tr>
<td>3</td>
<td>81791678</td>
<td>2009-01-07 13:03:09</td>
<td>2009-01-07 13:03:09</td>
<td>18</td>
<td>10.44</td>
<td>10.46</td>
<td>-0.02</td>
</tr>
<tr>
<td>5</td>
<td>81791999</td>
<td>2009-01-07 13:45:11</td>
<td>2009-01-07 13:45:11</td>
<td>58</td>
<td>10.45</td>
<td>10.45</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Figure 1 Time series of intakes based on the amount of food available in the trough

The dataset on weights contains the weights of some of the animals at three different points in time: 16.01.2008, 27.02.2008 and 12.03.2008.
The period when the animals were weighed precedes the period when the animals’ feeding information was obtained, thus the information on weights over time has limited relevance for the analysis, which is aggravated by the fact that several boxes miss totally or partially weight information for their pigs.

Several boxes have no weight data available or they miss data for some individuals or for some measurement dates. The statistics on data availability by box can be seen below.

<table>
<thead>
<tr>
<th>box</th>
<th>has.weight</th>
<th>has.no.weight</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>11</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>21</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>22</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>24</td>
<td>11</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>25</td>
<td>6</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>
It is worth noticing that a few animals have been relocated between boxes over time or have been introduced or extracted at different dates. This fact can be potentially problematic, as the boxes don’t have a stable composition of individuals, but it also gives the possibility to study animals in different environments. In total 15 animals were relocated one time during the experiment.
Data cleansing

The following issues were encountered in the data collected from the experiment:

- Some animals lost the chip identifier and it was replaced by a new chip, which is known at each replacement but was not originally included in the data on intakes.
- The time registration is not consistent:
  o The inexisten date 2014-02-29 appears in the dataset
  o The time period between 2:00 and 3:00 AM on 2009-02-28 corresponds to the change of hour in the CET time saving calendar and should not have appeared in the data.
  o Animals entering the feeding trough before midnight and leaving it after midnight have a one day lag error at the time of exit.
  o Data is missing on some dates for some boxes
- Some intakes and occupancy times are not consistent with the food weights and times registered.
- Some intakes are negative or too large, and some food weights are clear outliers in the data due to errors.
- Some weight data is incomplete.
- Some lines of data are duplicated.

In addition, the data presents irregularities, which might be due to instrumental errors. An example is given in figure 3 below, where the constant decay pattern previously shown in figure 1 is now uneven, seemingly caused by instrumental errors.

Figure 3 Errors in the intake dataset

The raw data from the experiment contains some inferred intakes that are negative, as can be seen in figure 4, which shows the distribution of the intake of the animals at each time they are by the feeding device.
The possible causes for these negative intakes observed are either unknown or impossible to filter in practice, if they are systematic; for example, in the case of the interference of the process of refilling of food with the moments the animal is occupying the trough, it was not possible to model and estimate the process of refilling of food, in order to filter out its effects from the data. The chosen solution to handle these outliers was simply to remove these observations by setting a lower bound for the intakes at zero and an upper bound at 2kg. The resulting distribution for the intake data after trimming it can be seen in Figure 5.

In the following chart, we can see that sometimes the feeding trough is empty. This could introduce errors in the estimation of the feeding patterns, since the animals can’t feed due to this exogenous reason.
Fortunately, the times when the feeding trough is empty are very few, and thus there is not a strong necessity to introduce in the simulation model the possibility of lack of food in the trough. This also justifies the simplifying assumption taken in the simulation model of not modeling the process to load the food.

Only one date in the period presents a significant problem of lack of food in the troughs in a number of boxes, as can be seen in Figure 7. It can also be noticed that some boxes seem to have common processes to refill food, since they experience food scarcity at exactly the same times.
Figure 7. Simultaneous lack of food in several boxes.

No action was taken in this case, as the deletion of data would have affected the estimation of feeding patterns to a higher degree than the effects of the empty feeding troughs have during this short time interval.

Another problem is observed in the box 26: there is no activity until Jan 15. The data during that period was ignored (removed) to avoid the spurious estimation of null intakes (Figure 8).
Other food weights in the data seem erroneous, as it is illustrated in the following cases:

Figure 8 Inactive period in the dataset
However, the variable of interest for the analysis is the difference between the initial and final amounts of food (intakes) at each event, and these eventual drops in the registered food weights cause some erroneously large estimations of intakes. This problem was previously resolved by removing large intakes over 2kg from the dataset.

In the dataset containing animal weights, some animals were missing complete or partial information. In the case of partially missing information, linear regression estimated on the complete information set was used to extrapolate to the animals with missing data the average increase in weight (expressed as percentage of the previous weight) from one date of measurement to the next.
Data transformations

In order to extract meaningful information from the dataset, some variables must be derived from the intakes of food and times of occupancy of the trough by each animal. Qualitatively, it has been determined from the inspection of the dataset, input and expertise from the Veterinarian Faculty at the Universitat Autonoma de Barcelona and similar findings in related literature (Fernández Capo, 2002), that the factors most relevant to the feeding patterns are: (i) the fight for occupancy between individuals (hierarchical dominance), (ii) the hunger that the animals seem to experience and (iii) the circadian nature of the animals (double-peak feeding pattern during the day, as shown later in the next section).

The following variables were calculated from the time series on intakes:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Descriptor</th>
<th>Calculation level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hunger</td>
<td>Intake speed: intake (as % of animal weight) by unit of time</td>
<td>Each event</td>
</tr>
<tr>
<td></td>
<td>Exponentially increasing function of accumulated hunger on time</td>
<td>Each animal and event</td>
</tr>
<tr>
<td>Dominance</td>
<td>Average daily occupancy time</td>
<td>Each animal</td>
</tr>
<tr>
<td></td>
<td>Median occupancy time</td>
<td>Each animal</td>
</tr>
<tr>
<td></td>
<td>Average occupancy time with low intake speed</td>
<td>Each animal</td>
</tr>
<tr>
<td></td>
<td>% interactions (fights for occupancy) won</td>
<td>Animal pairs and &quot;fight events&quot;</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
<td>Each animal</td>
</tr>
<tr>
<td>Occupancy time</td>
<td>Occupancy time</td>
<td>Each event</td>
</tr>
<tr>
<td>Come back time</td>
<td>Time to come back to eat</td>
<td>Each animal and event</td>
</tr>
</tbody>
</table>

Two of the variables listed above are complex to define and will be explained next. These are the hunger level, considered as a state variable that decreases with food intakes and increases over time, and the animal dominance estimated from seemingly animal interactions indirectly observed in the data.

The hunger level could be estimated by the speed at which the animal feeds while occupying the trough, standardized to a common size measure such as the percentage of the intake respect to the animal’s weight, or it could be indirectly measured as a state variable that would change based on the previously cumulated hunger, that would decrease at each intake and that would increase over time at some rate, preferably bounded at some level.

Expressing the hunger state level as its reciprocal – the satiation level– the basic dynamics of the variable, in the absence of intakes, can be formulated as:

\[
satiation(i, j_i) = satiation(i, j_{i-1}) \exp\{\alpha (t_i - t_{j_i})\}
\]

where

- \(i\): animal ID
- \(j_i\): event nr. in the series of events for the animal \(i\)
- \(\alpha < 0\): rate of satiation decrease by unit of time (sec)

The decrease factor \(\alpha\) is roughly estimated as 1e-4 based on the time a dominant individual would need to get hungry again, which is around 8 hours. The time is expressed in seconds. The hunger variable is then the opposite: 1 minus satiation.

Further elements are introduced in the calculation of the hunger variable: it is bound between 0
and 1 and the intake is subtracted from the cumulative satiation level by standardizing it to each animal’s average intake during 8 hours, scaled by 1.5 to allow for the possibility of occasional higher-than-average intakes. In this way, the satiation measure so obtained is expressed in the same order of units as the intakes, and it could be roughly interpreted as the amount of food left in the animal’s organism. The hunger level is calculated for each animal at each time – entries and exits.

The R implementation for the calculation of the hunger level is the following:

```r
library(plyr)
data$hunger.out <- data$hunger.in <- NA
for (i in unique(data$id)) {
  tmp.id <- arrange(data[data$id == i,], time.in)
  prev.hunger.out <- 1
  for (t in tmp.id$time.in) {
    tmp.t <- tmp.id[tmp.id$time.in == t]
    hunger.in <- 1 - min(1-max(prev.hunger.out, 0), 1)*exp(-
      tmp.t$time.between.individual.exit.and.entry/10000)
    hunger.out <- max(min(1-(1-hunger.in)*exp(-tmp.t$duration/10000) -
      tmp.t$intake/(1.5*tmp.t$avg.daily.intake/3), 1), 0)
    data[data$id == i & data$time.in == t, "hunger.in"] <- hunger.in
    data[data$id == i & data$time.in == t, "hunger.out"] <- hunger.out
    prev.hunger.out <- hunger.out
  }
}
data <- ddply(data, .(id), function(x) data.frame(x, next.return=c(x$time.between.individual.exit.and.entry[-1], NA)))
data$hunger.in.decile <- as.character(data$hunger.in%/%0.1)
data[data$hunger.in.decile=="10","hunger.in.decile"] <- "9"
data$hunger.out.decile <- as.character(data$hunger.out%/%0.1)
data[data$hunger.out.decile=="10","hunger.out.decile"] <- "9"
```

Some results from the hunger variable can be seen in Figure 10 and Figure 11. The hunger state is only shown at the events of entry (figure 10) or exit (figure 11) and is represented in the y-axis. It ranges below 0 and 1 by construction. The subsequent intake after entering the trough (figure 10) or the previous intake before leaving the trough (figure 11) are represented by the size of the dots in the graphs. All pigs in the graphs belong to the same box. In both graphs we can see cases of pigs that eat frequently and in small amounts –this are assumed to be the weaker animals- and pigs eating less frequently, staying longer time and eating high intakes – the dominant individuals. By the level of the hunger state, we can see that the dominant animals enter the trough only after a long period and when they are sufficiently hungry; and they leave often satiated after an uninterrupted stay.
Pigs are known to have strict hierarchical orders that get established after a few days of interaction (Fernández Capo, 2002). Here we assume, as is common practice in biology, that the hierarchical relations are linear and transitive, although not deterministic: some animal interactions might be occasionally resolved in favor of the weakest animal.

The definition of an interaction has been defined here as the event in which an animal is replaced by another one in the trough in a time lapse of less than 5 seconds and the same animal returns to the trough in less than 5 minutes from that moment, indicating that it intended to feed for a longer period but it was probably pushed away by another animal. Other animals might
enter the trough in the meantime. In this definition of interaction event, a loss has been assigned to the animal that was occupying the feeding trough, and a win to the replacing animal. No draw is possible in this definition.

There is, however, a caveat in this definition: interactions for dominant animals will be biased, since possibly many of their won interactions with other weaker animals will not be observable in the data, as the weaker animal will almost never push them out of the trough.

A measure suggested by Neumann et al. (2011) is the application of the Elo rating to animal hierarchies. The Elo measure, named after its creator, the mathematician Arpad Elo, is popularly known for its application to the calculation of chess ratings. It is a Bayesian measure that measures relative dominance and incorporates new information on a continuous basis. It depends on the previous Elo ratings of the players and it changes their ratings in an asymmetric fashion, decreasing the stronger player’s rating to a higher degree the weaker his opponent is, and vice versa. In order to compute the measure we need to define what an animal interaction event will be and how its result will be resolved. Once defined, the Elo score of each pig can be calculated dynamically, since the ratings can change over time until they eventually converge towards a stable distribution of ratings. However, the final score over the entire period has been used here, since the measure might take some time to converge (if ever) and in the case of non-dominant pigs, it is somewhat unstable. This calculation is implemented in R in the package EloRating by Christof Neumann & Lars Kulik.

Using the Elo measure, the probabilities of win for a given pair of animals are given by their ratings through the following formula:

$$\text{Win}(elo_1, elo_2) \sim \text{Bernoulli}(p)$$

where

$$p = F^{-1} \left( \frac{elo_1 - elo_2}{200 \sqrt{2}} \right)$$

and F is the cumulative Gaussian distribution function with mean 0 and variance 1.

The Elo measure is centered at 1000. As an example, the probability of win for an animal with an Elo of 1200 against another animal with an Elo of 1000 is 76%. In figure 12 an example of calculation of Elo score dynamically over time is shown for all the animals in one particular box.
Stability measures are also available from the calculation. They produce a statistic that measures how fast the convergence towards stability is achieved. The S index, for example, is a measure ranging from 0 (unstable) to 1 (perfect stability) which is based on changes in ranks (Neumann et al., 2011).

As an alternative to the Elo score, using the sample interaction matrix (percentage of wins for each animal in rows against other animals in columns), the equivalent David’s score, which is a measure for hierarchical dominance broadly used in biostatistics can be also inferred.

These measures are ready implemented in the R package EloRatings. For example, the S index:

```r
> stab.elo(res)
[1] 0.553
```

Or the equivalent David’s scores:

```r
> head(DS(createMatrix(res)))
   ID      DS normDS
1 22112629 32.92309 142.1155
2 96658712 31.75664 142.1115
3 20096216 30.36404 142.1065
4 13099764 29.70662 142.1042
5 96658707 28.65816 142.1005
6 96658698 28.40487 142.0997
```
**Exploratory analysis**

Once extra variables have been generated and the raw data has been cleansed, we are in position to draw some observations on the feeding pattern of the animals. In Figure 13, a data sample for the quantity (kg) of food in the feeding trough is shown. We can observe how some individuals alternate the occupancy of the feeding trough. The visual representation of the data on intakes offers a first insight into the feeding habits of the animal which is pervasive across all boxes and over time: they tend to eat continuously throughout the day, but there are some periods of reduced activity, between midnight and sunrise; and some individuals show a more repetitive pattern to enter the trough, whereas others enter fewer times and stay longer periods.

![Box 20 on 28.12.2009](image)

**Figure 13. One-day sample of underlying consumption data**

The amount of food decreases at a fairly constant pace until the feeding trough is refilled with more food, added in fixed quantities. Figure 14 shows an example of a period when the amount of food decreases at a very constant pace, regardless of the animal occupying the trough. The feeding speed is very homogeneous across all animals and nearly at a rate of 1 kg of food per hour.
Figure 14. Feeding pattern with constant consumption speed.

By looking at some chosen individuals, we can see that some seemingly dominant animals eat fewer times and often at similar hours, avoiding the night hours (figure 15) and some animals eat more often and at early hours (figure 16).

Figure 15 Feeding patterns of a dominant pig
There are some periods during the day when there is no much activity in the feeding trough; usually between midnight and 6:00, corresponding to the sleeping habits of the animals. Otherwise, the feeding patterns of the pigs seem well spread over the day. We can also find that the feeding activity peaks at two times during the day in all boxes. Figure 17 shows the average occupancy in each box during the measurement period. In one particular box the feeding trough is occupied full time between 17:00 and 18:00. These feeding pattern could reflect a healthy feeding behavior of the pigs, since few animals are forced to eat during the sleep hours due to excessive competition, and they follow normal daily patterns which coincide roughly with sunrise and sunset hours. This circadian characteristic of the pigs should also be obtained from the multi-agent simulation model if this is well constructed. Figure 18 shows that also the amount of intake peaks following a bimodal distribution during the day. This pattern is pervasive across boxes.
Figure 17. Average occupancy by hour of the day

Figure 18. Average intake by hour and by box

In figures 19 and 20 we can see that there is no clear pattern for the relation between the only biometric variable available from the experiments —the animal weight—and the feeding habits. They show the intake and occupancy time for those pigs that have available data on weight,
grouped by box. There is no clear relation between animal size and feeding dominance. Some heavy animals feed the least and for shorter time.

Figure 19 Average daily intake by pig vs weight

Figure 20 Average daily occupancy by pig vs weight
In figure 21 we can see that the weight of the animals is not a good predictor for the average feeding speed of the animal.

![Hunger (intake as % of weight / occupancy) by pig](image)

**Figure 21 Hunger, as feeding speed, vs weight**

Some animals eat more regularly and in lower amounts. This is however not predicted by the biometric variable—weight—as would be suspected. This is shown in an example in figure 22, where the daily distribution of intake for all animals in one box is shown. Some weak animals eat irregularly during the day whereas other weak animals tend to eat following a more regular schedule.
In figure 23 we can see an example on how some dominant animals tend to concentrate their occupancy during the two main meal times, but their domination of the trough doesn’t always follow closely their weight. Some heavy animals occupy shortly the trough.

Figure 22 Average hourly intakes by pig vs weight

Figure 23 Average occupancy schedule by pig vs weight
However, there is a variable that presents some predicted power for the time distributions when the animal comes back to the trough or occupies the trough. This is the state variable hunger, previously described. This variable, however, is not biometric and it only helps simulate the different time distributions for the dominant pigs as a function of the hunger level at the time of leaving the trough. In figure 24 we can see that in the case of dominant (long occupancy) animals, the level of hunger determines the shape of the distribution of time to come back to eat, whereas in the case of weak animals it doesn’t modify the distribution.

![Time to return as a function of hunger for a dominant pig (blue) and a subordinate pig (red)](image)

**Figure 24** hunger level vs time to come back to the trough for a dominant and a weak animal

In the rest of this chapter, some other relations are visualized to show how the only variable on feeding behavior that is significantly predicted by the weight of the animals is the amount of food intake, but this is not such case when the intake is standardized by the weight of the animal.
Part II – Simulation
An agent-based simulation model can be used to replicate the feeding behavior of the animals under the presence of competition and other factors. The varying level of drug in their bodies, which depends on the food intake at different times, the amount of drug supplied in the food and the characteristics of the absorption process of the drug are implemented in the simulation.

In order to replicate the dynamics of the observed feeding patterns of the pigs, we need to know what variables are driving their behavior. These variables have been explored in the previous section. The feeding patterns will be a consequence of the type of feeding space, the number of animals sharing the space, their characteristics (e.g., race, weight, gender, age ...) and the hierarchical dominance that is established among them. Ideally, the simulation model should be capable of predicting the feeding behavior of a group of animals based on some observable animal characteristics, preferably biometric, and of extrapolating the results of the experiment out-of-sample to other farm configurations and environments. However, in the previous section it has been observed a weak empirical relation between the only biometric information available, the weight, and the feeding patterns.

Another alternative approach is to formulate a factor model in which non-observable factors drive the feeding patterns, and these unobservable factors can be stochastically attributed to a group of animals once its probabilistic distribution is known and we assume the hypothesis of invariance of this distribution under changes in the configuration of the experiment. This hypothesis has, unfortunately, limited testability with the available experimental data. Also, the few animals that are relocated to a different box could be used to assess if there is statistical evidence of any change in habits given the new composition of the group to which they are allocated. However, there were no significant differences in the composition of the groups of animals across boxes in this experiment so the change of box provided little evidence of for changing conditions.

Also, as observed from the previous section, the high level of noise in the data and dispersion of the feeding and return times make it difficult to formulate a full parametric model in which the distribution of the events is easily and robustly parameterized on some driving factors. Therefore, we have chosen to adopt a semi-parametric formulation in which the times of occupancy and times to return are modelled by parametric distribution functions, but in which the unobservable factors that define the type of behavior of the animals and the dependence of these random times to the hunger level are introduced into the model as discretizations of the sample set of animals. Discretization is feasible with this data since the sample size is high.

The discretization of the behavior of the pigs into types of behavior is done based on characteristics of the random times observed for each kind of pig. Dominant pigs tend to occupy the trough in an uninterrupted fashion and they return to feed after long lapses of time, whereas the subordinated animals enter and exit the trough repeatedly and stay shorter time. There is a relation in the random times at which the animal stay and return.

A logical model will reflect how the agents (pigs) compete in their attempts to occupy the feeding trough when they wish to do so, in order to feed, and the sampled stochastic variables and agent interactions will determine the output of the simulation. Competition is reflected both in the observed alternating occupancy in their fight for the space and in the length of the times different animals stay and return. The model should include both effects and reflect the rank hierarchies when the animals attempt to occupy the trough but other animals are too. This is the stage of the process when interactions between agents happen. Otherwise, the simulation could be applied on a single-animal basis.
The model chosen for the simulation reflects the experiment in its full scope; every event registered in the data, each time animals enter and leave the trough, is simulated. Two main alternative approaches could have been considered:

1. The data could be segmented and aggregated
   Meals are aggregated whenever the animal occupancies are clustered in time. Competition could then be reflected in its effects on intake, since animals that are continuously entering and leaving spend less time feeding during the period. An advantage would be that hunger is more clearly determined after the entire meal and noise in the data would be reduced by aggregation into lower frequency samples.

2. The full process is captured to its highest detail level
   Repeated entries are modelled to its full extent in the distribution of times. One main advantage of this approach is that occupancy times don’t overlap and there is no need to introduce theoretical assumptions in the model, but simply describing and emulating the observed process.

The latter approach was chosen.

Basically, the model contains the following elements:

- **Feeding trough**
  Food is always available and only one animal can occupy the trough at a time.

- **Animals**
  Each has assigned a type of behavior.
  Hunger levels vary over time and with intakes.
  Hierarchical ranks are reflected when the animal finds the feeding trough occupied.

- **Queue**
  Several animal might attempt to feed while there is still some animal occupying the space. Empirical evidence provided by subject matter experts indicates that the pigs wait their turn to eat, and the eventual availability of the trough during their waiting time, makes the simulation model amenable to this approach, since in the opposite case, if the animals came back after a new time lapse when the trough was occupied, it would not be observable in the data and therefore a longer time lapse would be estimated.
  Higher ranked animals might obtain priority to enter first the trough, although this not deterministic.
  Animals already occupying the trough are not let be kicked away from new animals; their degree of dominance will be already reflected in their probabilistic distributions of the occupancy time.

- **Occupancy time**
  The time an animal stays in the feeding trough. It will depend on the type of behavior of the animal and the level of hunger at that moment.

- **Return time**
  The time the animal takes to return to the feeding trough. It will depend on the type of behavior of the animal and the level of hunger at the moment when the animal left the trough the last time.

- **Intake**
  The intake of food during the occupancy of the trough can be decomposed into occupancy time and feeding rate, which is assumed constant, although, in a more
complex model, it could be modelled stochastically and depending on the type of behavior and hunger level.
**Implementation**

To specify the system that will be simulated, we use a formal representation language that will help in the process of understanding and implementing the simulation.

The formal specification of the problem yields five advantages (Fonseca i Casas, 2014): (1) the formalization is more precise than the textual representation, (2) the formalisms represent the dynamic relationships between the different elements of the relevant processes in a detailed form, (3) a deeper understanding of the model behavior helps in the implementation process and grants independence to choose the optimal tool to perform the implementation, (4) the process can improve communication between different personnel involved in model construction, and (5) the formalization of the model can be considered a product itself. In addition, prediction is not the only purpose of a model. Other benefits of the construction of the model are possible, including the explanation of how the system works or the discovery of new problems.

Specification and Description Language (SDL) is an object-oriented, formal language defined by The International Telecommunications Union-Telecommunications Standardization Sector (Telecommunication standardization sector of ITU 1999; formerly Comité Consultatif International Télégraphique et Téléphonique [CCITT]), in recommendation Z.100. The language is intended for the specification of complex, event-driven, realtime, and interactive applications involving many concurrent activities that communicate using discrete signals (Reed 2000). SDL uses four levels in order to describe model behaviors: system, blocks, processes, and procedures, as shown in Figure 25. An example of the application of SDL on Multiagent Systems (MAS) is provided in Fonseca i Casas (2014).

![Figure 25 Structural view of an SDL system](image)

The model is implemented using the software SDLPS, developed at the Polytechnic University of Catalonia by Pau Fonseca I Casas. The SDL diagrams are drawn in Microsoft Visio and rendered automatically into C code by means of a plug-in for Microsoft Visio. Only externally defined functions need to be added into the simulation program. These will be:

- The random sampling of behavioural types and dominance ranks, according to its empirical distribution in the data
- The hunger state variable
- The sampling of random occupancy time
- The random resolution of conflicts in the queue at the feeding trough, according to probabilities defined by the dominance ranks
- The sampling of random time of return to the feeding trough
The SDL diagrams for the simulation model developed in Microsoft Visio can be found in the appendix.

In the following sections, we address the estimation and implementation of the stochastic variables in the simulation.
**Distribution of the random variables**

The observed random time variables of occupancy and return are a result of the animal’s endogeneous dynamics (hunger, sleep, weight...) and the exogeneous interactions with other animals while competing for food. Hence, two approaches can be taken to model stochastic variables that are affected by interactions with other animals:

1. The variable’s probability distribution can be modelled generically for all animals, free from interaction effects, by using the most dominant pigs as the sample set, since they are least affected by interactions. The effective specific distribution for each animal will be only a consequence of eventual interactions.

2. The interactions can be modeled directly, embedded into the distribution of the variables for each behavioural type of pig. Dominance ranks are then only expressed explicitly in the model by the position achieved in the waiting queue.

The advantage of the first approach is that the resulting effects will be a consequence of the configuration of the group and the farm’s layout: it can potentially simulate the feeding patterns of any group of animals, given their dominance scores and number of feeding troughs in the box. This is the “structural” approach. The second approach has the advantage that each animal is modelled directly on the resulting patterns from their current environment. This is the “reduced-form” approach, using the terminology that is popularly employed in econometrics.

The first approach has the difficulty of having to estimate distributions clean from interaction effects and, although these could be estimated from the most dominant animals, it also assumes that the established hierarchy will be fairly linear based on the Elo ratings of the group, but the resulting hierarchical ranks could have non-linear effects, where only a few individuals are accepted as the dominant individuals in every group, even though several animals had similar dominance scores in the context of other more heterogeneous groups. This clustering behavior is difficult to reflect in a structural form.

The chosen approach in this study is the second, where variable distributions are estimated on each individual, including the interaction effects in the experiment. The dominance ranks of the pigs will be already related to their behavioural types but they will also be reflected, most explicitly, in the queing at the feeding trough, where the most dominant animals will tend to obtain a better position in the simulation model.

A disadvantage of the second, chosen approach is that there is the potential problem of double-counting the effects of interactions: if these are already reflected in the behavioural time distributions, the effects from the interactions in the waiting queue could be significant and affect again the resulting time distributions. This should be corrected eventually by recalibrating the probability distributions to achieve expected results.

In the next pages, the random time variables of the system -occupancy time and return time- are estimated for each animal, and behavioural clusters on these two interrelated variables are determined. First the clusters are determined based on the estimated parameters of each animal-specific distributions. Once the behavioural clusters of animals are determined, the data for all animals belonging to a same cluster and with a same hunger level are pooled into a common dataset for each combination of cluster and hunger level, and a cluster-and-hunger-specific distribution is then fitted. The hunger level is discretized by taking deciles on the range of values of the hunger variable, which ranges from 0% to 100%. Again, notice that hunger level is not included in the first stage when the clusters are defined based on the animal-specific distributions. This is so because some animals may experience certain hungry levels more often
and this effect should be included in the distributions in order to determine behavioural types properly.

Next, univariate probability distribution are fitted to the occupancy time and to the return time of each animal: one (marginal) distribution is fitted independently for each animal. The multivariate dependence between both distributions for an animal will be addressed when the animal types are allocated to clusters based on the estimated parameters for these two distributions. In order to fit a time distribution for each animal, regardless of the hunger levels at each moment, we will consider the gamma distribution, the Weibull distribution, the log-normal distribution and the exponential distribution. All of them are common distributions used to model longitudinal data. The occupancy time distribution for all the animals shown in figure 16 contains heterogeneity in terms of different animal behaviours and different states of hunger, so we would expect a mixed distribution for this aggregate sample. In this case a Weibull distribution would fit better the distributional mixture. However, fitting the distribution to each animal, the Gamma distribution seems to be a good robust fit. The Gamma distribution is also a sensible choice, since it is the theoretical distribution that appears in the waiting times between events in a Poisson process.

In figure 26 we can see the generic shape for the occupancy time distribution.

![Figure 26: Generic occupancy time distribution for all animals (Gaussian kernel interpolation)](image)

Figure 26 shows a sample of animal-specific distributions for all animals in same a box. We can see in figure 27 that there are clear differences among the individual behaviours in occupancy times.
Figure 27 Animal-specific occupancy time distributions
In figure 28 we can see the same heterogeneity in the return time distribution.

**Figure 28 Animal-specific return time distributions**

For each animal, a Gamma distribution for the occupancy time and a Gamma distribution for return time are estimated. An example is illustrated here for one animal, showing goodness of fit tests and charts to visualize the fits. In the cases of the Weibull and Exponential distributions, the distributions are fitted by the maximum log-likelihood; in the cases of the Gamma and Log-normal distributions, the method of moments is used. This is because some methods fail numerically in those distributions in some particular cases. Other methods, such quantile matching or maximization of goodness of fit statistics are possible.
Figure 29 empirical return time distribution (Gaussian kernel interpolation)

Goodnes-of-fit statistics for the return time distribution of a particular pig:

**Goodness-of-fit statistics**

<table>
<thead>
<tr>
<th></th>
<th>Exponential</th>
<th>Weibull</th>
<th>gamma</th>
<th>lognormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kolmogorov-Smirnov</td>
<td>0.4105674</td>
<td>0.1910207</td>
<td>0.165436</td>
<td>0.1117062</td>
</tr>
<tr>
<td>Cramer-von Mises</td>
<td>123.7760930</td>
<td>11.6978281</td>
<td>16.5737716</td>
<td>11.4404447</td>
</tr>
<tr>
<td>Anderson-Darling</td>
<td>1576.7084232</td>
<td>71.6579233</td>
<td>62.9480778</td>
<td>71.4021014</td>
</tr>
</tbody>
</table>

**Goodness-of-fit criteria**

<table>
<thead>
<tr>
<th></th>
<th>Exponential</th>
<th>Weibull</th>
<th>gamma</th>
<th>lognormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aikake's Information</td>
<td>43570.12</td>
<td>39925.21</td>
<td>40059.15</td>
<td>39763.01</td>
</tr>
<tr>
<td>Bayesian Information</td>
<td>43875.88</td>
<td>39936.73</td>
<td>40070.67</td>
<td>39774.54</td>
</tr>
</tbody>
</table>

Empirical and theoretical CDFs
The estimated parameters for a Gamma distribution for the return time are, for this individual:

```
Fitting of the distribution ' gamma ' by matching moments
Parameters :
  estimate
    shape 3.313021e-01
    rate 8.552908e-05
Loglikelihood: -20027.57  AIC: 40059.15  BIC: 40070.67
```

The same estimation is done for the occupancy time distribution. This particular pig presents a slightly bi-modal empirical distribution.
Goodnes-of-fit statistics for the return time distribution of this particular pig:

**Goodness-of-fit statistics**

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Exponential</th>
<th>Weibull</th>
<th>Gamma</th>
<th>Lognormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kolmogorov-Smirnov statistic</td>
<td>0.2456988</td>
<td>0.116186</td>
<td>0.119439</td>
<td>0.09367758</td>
</tr>
<tr>
<td>Cramer-von Mises statistic</td>
<td>43.3522122</td>
<td>6.311559</td>
<td>7.4085969</td>
<td>2.95873641</td>
</tr>
<tr>
<td>Anderson-Darling statistic</td>
<td>247.5230681</td>
<td>41.886969</td>
<td>51.5337609</td>
<td>18.92763243</td>
</tr>
</tbody>
</table>

**Goodness-of-fit criteria**

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Exponential</th>
<th>Weibull</th>
<th>Gamma</th>
<th>Lognormal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aikake's Information Criterion</td>
<td>10009.06</td>
<td>9478.758</td>
<td>9662.183</td>
<td>9236.929</td>
</tr>
<tr>
<td>Bayesian Information Criterion</td>
<td>10014.82</td>
<td>9487.285</td>
<td>9673.679</td>
<td>9247.955</td>
</tr>
</tbody>
</table>
The estimated parameters for a Gamma distribution for the return time are, for this individual:
Having estimated the parameters of the two distributions for every animal in the dataset, we proceed to determine clusters on those four parameters. We will use the K-means algorithm. The suggested number of clusters is 8, based on the BIC criterium.

However, using 8 clusters, some sample datasets for some particular combinations of cluster and hunger decile are too thin for estimation of the time variables. Therefore the final chosen number of clusters is 7.

In the next figures we can see the Gamma distributions fitted on the pooled data of the pigs belonging to the same cluster. Behavioural differences are apparent and the transition of dominant through subordinate behavior is smooth rather than clustered around certain polarized behaviours. This is the reason why 7 clusters were needed to describe the spectrum of behavior of the dataset.
The next figures show the distributional parameters from the animal-specific time distributions identified by estimated cluster. We can see that rate and shape parameters of a same type of distribution are highly correlated, and that the shape parameters across distributions are also highly correlated; the rate parameters across distributions show a lower degree of dependence. We can also see that the shape parameters are the most dominant to determine the clusters. This can be also seen in the biplot of the four parameters in figure 36, built on the two first principal components.
Figure 32 Shape and rate parameters of the time to come back to eat, coloured by cluster

Figure 33 Shape and rate parameters of the occupancy time, coloured by cluster
Figure 34 Shape parameters of both time distributions, coloured by cluster

Figure 35 Rate parameters of both time distributions, coloured by cluster
The clusters will be sampled in the simulation based on a multinomial distribution with probabilities of occurrence for each cluster estimated by the empirical frequency of animals in each cluster.

\[
\begin{array}{c|c}
\text{cluster} & \text{count}\mid \text{id} \\
1 & 42 \\
2 & 41 \\
3 & 23 \\
4 & 64 \\
5 & 22 \\
6 & 34 \\
7 & 11 \\
8 & 53 \\
\end{array}
\]

Once the clusters, or behavioural types, are defined, we proceed to estimate the Gamma distributions for each combination of cluster and hunger level. In figure 37 we can see that the mean and variance of the distributions depend on the hunger level, but this dependence is not the same for all types of pigs; dominant pigs have a higher dependence on the hunger level, and the attribute of dominance is a nearly continuous one, as we have seen by the need to use 7 types.

Figure 36 Biplot of the four parameters (first two principal components) coloured by cluster
Figure 37 Different effect of hunger level on the distribution of times for different animal types

In figures 38 though 41 we can see the estimated parameters for the occupancy and return times for each type and discretized hunger level.

Figure 38 Shape parameter for the time to come back to eat by hunger level
Figure 39 Rate parameter for the time to come back to eat by hunger level

Figure 40 Shape parameter for the occupancy time by hunger level
Figure 41  Rate parameter for the occupancy time by hunger level

In figures 42 and 43 we can see how the distributions vary for a certain dominant animal and a certain subordinate animal correspondingly. They vary across hunger levels in both animals but to a higher extent in the case of the dominant animal.

Figure 42 Time spent in the trough vs hunger for a certain (dominant) type
Dominance is another factor affecting the feeding dynamics, and is already reflected in the time variables. As seen in a previous section, an especial event was defined to quantify the only probable fights for occupancy that could be observable in the experimental data. We obtained Elo ratings for each pig by using the interaction matrix from these events between pairs of animals. In order to introduce the distribution of dominance ratings in the simulation and we introduced the behavioural types, we need to asses not only the distribution of ratings but also their dependence to the behavioural types, since they should be sampled jointly. However, the next figures demonstrate that there is a very weak relation between both.
Dominance (Elo) vs Weight

correlation = 0.039

Dominance (Elo) vs TimeToEat shape parameter

correlation = 0.1
Dominance (Elo) vs TimeToEat rate parameter
correlation = 0.14

Dominance (Elo) vs TimeToHungry shape parameter
correlation = 0.154
A linear of dominance ratings on the distributional parameters used to compute clusters yields a very weak explanatory model, although most parameters seem to be statistically significant to help determine the dominance rating.

\[ \text{lm(formula = elo ~ id.stats$TimeToHungry.shape + id.stats$TimeToHungry.rate +} \]
\[ \text{id.stats$TimeToEat.shape + id.stats$TimeToEat.rate, data = id.stats)}\]

**Residuals:**

<table>
<thead>
<tr>
<th>Min</th>
<th>1Q</th>
<th>Median</th>
<th>3Q</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>-305.26</td>
<td>-84.04</td>
<td>-9.79</td>
<td>89.94</td>
<td>474.49</td>
</tr>
</tbody>
</table>

**Coefficients:**

|                  | Estimate | Std. Error | t value | Pr(>|t|) |
|------------------|----------|------------|---------|----------|
| (Intercept)      | 915.66   | 28.81      | 31.788  | < 2e-16 *** |
| id.stats$TimeToHungry.shape | 184.10 | 63.54 | 2.897 | 0.00406 ** |
| id.stats$TimeToHungry.rate   | -194512.49 | 492728.25 | -0.395 | 0.69331 |
| id.stats$TimeToEat.shape     | -112.08 | 98.50 | -1.142 | 0.25371 |
| id.stats$TimeToEat.rate      | 392.01 | 123.15 | 3.183  | 0.00162 ** |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**Residual standard error:** 124.8 on 281 degrees of freedom

(5 observations deleted due to missingness)

**Multiple R-squared:** 0.07706,  **Adjusted R-squared:** 0.06392
**F-statistic:** 5.865 on 4 and 281 DF,  **p-value:** 0.0001513

Hence, the dominance scores estimated in this way are not tightly related to the behavioural types. This could be due to the fact that the resolution of conflicts between competing animals in the queue is not observable in the data. Furthermore, the hunger level doesn’t explain the estimated dominance variable either, whereas it did for the time variables; there is no significant difference in the hunger of the animal kicked-out respect to the hunger of the animal entering beyond that difference that would be expected by the fact that the animal leaving has been eating already. Therefore, the hunger variable doesn’t help in explaining the results between competing pigs.
Distribution of hunger differences in 'kicked-out' situations

Hunger of animal entering minus hunger of animal leaving to return later

Density

Hunger of animal entering minus hunger of animal leaving to return later
Implementation of the random variables

Following, we present the C code used in the simulation model to introduce the stochastic variables described in the previous sections.

```c
#include <stdio.h>
#include <gsl/gsl_cdf.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>

gsl_rng * r; /* global generator */
const gsl_rng_type *T; /* type of generator */

unsigned int get_type(void)
{
    /* random type assigned to 1 of 7 possible clusters with probabilities p[]
    const size_t K = 7; /* number of animal types */
    const double p[] = {0.100000000, 0.193103448, 0.113793103, 0.037931034,
                        0.244827586, 0.082758621, 0.227586207}; /* frequency of each animal type */
    unsigned int n[K]; /* array of size N to store sampled animal types */
    int i;
    unsigned int type = 0;

    // 1 sample is placed in 1 of K clusters with probabilities p. The index
    // of n where the sample is placed is the type.
    gsl_ran_multinomial (r, K, 1, p, n);
    for (i = 0; i < K; i++) if (n[i] == 1) type = i + 1;

    return type;
}

float get_time_to_hungry(unsigned int Type, unsigned int HungerLevel)
{
    /* the distribution gamma has two parameters: shape and rate
     * each pair of parameters is defined by the type of pig (7 types) and
     * the level of hunger (integers from 1 to 10)
     * the resulting time is expressed in seconds

    float shape_parameters[7][10] = {
        [0.268719303297549, 0.261313551727635, 0.273516174212435, 0.26780004143624, 0.248374152364477, 0.2363043683003, 0.218784674636291, 0.20695001571734, 0.194208746139443, 0.172343137511315],
        [1.00596637217462, 0.920466848380145, 0.93185514801416, 0.9163538740743, 0.798598093868517, 0.762092512953533, 0.725744574953859, 0.631178675957013, 0.52750513781094, 0.460981696793619],
        [1.07185142529571, 1.00569917677071, 1.0036839019413, 1.0284807726819, 0.924665326206152, 0.907083956444375, 0.795488134950116, 0.749093358675891, 0.561928684036188, 0.502700900742333],
        [0.28734152364477, 0.2363043683003, 0.218784674636291, 0.20695001571734, 0.194208746139443, 0.172343137511315],
        [1.00596637217462, 0.920466848380145, 0.93185514801416, 0.9163538740743, 0.798598093868517, 0.762092512953533, 0.725744574953859, 0.631178675957013, 0.52750513781094, 0.460981696793619],
        [1.07185142529571, 1.00569917677071, 1.0036839019413, 1.0284807726819, 0.924665326206152, 0.907083956444375, 0.795488134950116, 0.749093358675891, 0.561928684036188, 0.502700900742333],
        [0.28734152364477, 0.2363043683003, 0.218784674636291, 0.20695001571734, 0.194208746139443, 0.172343137511315]
    };
```
float rate_parameters[7][10] = {
  {5.05693267610817e-05, 5.64785346766507e-05, 5.254482357097e-05, 0.8379895898544241, 0.363028788721838, 0.33284305830516, 0.00000000000000, 0.00000000000000, 0.00000000000000, 0.00000000000000},
  {0.289224646430536e-05, 0.50481112724243e-05, 0.57625611298875e-05, 0.50000000000000, 0.50000000000000, 0.50000000000000, 0.50000000000000, 0.50000000000000, 0.50000000000000, 0.50000000000000},
  {0.8294654632, 0.000170819815243023, 0.000195117444263913, 0.00018020610181429, 0.00015923720212558, 0.000126744958355121, 0.54792092666645e-05, 0.73639854190838e-05, 0.8.87512606982244e-05, 0.9.16997640152736e-05, 0.164914409293819e-05, 0.5379461779409e-05, 5.1299937486304e-05, 5.00000000000000, 0.00013064192748445, 0.000165354319229846, 0.000174065714983894, 0.00017969666991, 0.00017226162211529, 0.000191241200416147, 0.000172798396118915, 0.00015623086632451, 0.000148285072586612, 0.00017177830482724, 0.174908593574e-05, 0.7.0209391928591e-05, 0.5.19910575082611e-05, 0.7.8873214683864e-05, 0.21379756384277e-05, 0.5.555347556675e-05, 0.8.70562658468133e-05, 0.8.41915179027925e-05, 0.8.12956727910829e-05, 0.7.86497970726603e-05},
  
  return gsl_ran_gamma (r, shape_parameters[Type-1][HungerLevel-1], 1/rate_parameters[Type-1][HungerLevel-1]);
}

double getWinProb (double dominance1, double dominance2) {
  double Z_score = (dominance1 - dominance2)/(200*sqrt(2));
  double WinProb = gsl_cdf_ugaussian_P (Z_score);
  return WinProb;
}

unsigned int getWin (double WinProb) {
  unsigned int Win = gsl_ran_bernoulli (r, WinProb);
  return Win;
}

float getTimeToEat (unsigned int Type, unsigned int HungerLevel) {
  /*
  
  float shape_parameters[7][10] = {
    {0.451530494807732, 0.399148663426249, 0.440521092073778, 0.4681227820016, 0.407161166743996, 0.44180960047273, 0.422248440108888, 0.387799797036698, 0.421006166467, 0.499246391365266},
    {0.289003167946789, 0.49575429942826, 0.529892659094522, 0.6258296569276, 0.30, 0.719901882135411, 0.762276765268647, 0.858616027857665, 0.89323853018502, 1.0428617717133, 1.22176824124506},
    {0.69610982130580, 0.877596657921258, 0.735683750400652, 0.8059598407295, 0.48, 0.91129442695906, 0.982127182751769, 1.0902264034718, 1.17176400287011, 1.34297179908789, 1.69496675853974},
  };
}
```c
float rate_parameters[7][10] = {
    {0.00759546150958754, 0.00587936094515589, 0.00636048660890055, 0.003241157, 0.00456966441712287, 0.0038958701715291, 0.003241157, 0.00456966441712287, 0.0038958701715291, 0.003241157},
    {0.00320500844430315, 0.00463601989522801, 0.00465965567360178, 0.004968382345493368, 0.00421049672505808, 0.00395212493503214, 0.003543210793617, 0.00322751907292607, 0.00293600454933668, 0.002629361307234},
    {0.00692388234545206, 0.00511704670433432, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234},
    {0.0167849919633696, 0.00511704670433432, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234, 0.0033068325493744, 0.002629361307234},
    {0.0041964875381143, 0.00571847521609829, 0.0045898983211389, 0.0040394507350843, 0.00353951671926916, 0.0030385428374061, 0.00353951671926916, 0.0030385428374061, 0.00353951671926916, 0.0030385428374061},
    {0.00386262659402242, 0.00222752291565697, 0.0042905791933367, 0.0033874917589886031129, 0.00255380375955098, 0.0036262659402242, 0.00222752291565697, 0.0042905791933367, 0.0033874917589886031129, 0.00255380375955098},
    {0.002857759886031129, 0.00255380375955098, 0.002857759886031129, 0.00255380375955098, 0.002857759886031129, 0.00255380375955098, 0.002857759886031129, 0.00255380375955098, 0.002857759886031129, 0.00255380375955098},
};

return gsl_ran_gamma (r, shape_parameters[Type-1][HungerLevel-1], 1/rate_parameters[Type-1][HungerLevel-1]);

float getFeedingSpeed (int Type, int HungerLevel) // TO DO (asumo que comen a ritmo constante de 3 kg por hora)
{
    // Expressed as kg/sec
    return 3/60/60;
}

float getIntake (float TimeToEat, float FeedingSpeed)
{
    return TimeToEat*FeedingSpeed;
}
```
Part III – Conclusions

The simulation study was continued after the writing of the present document. However, the methods of analysis and main observations have been already described here. Further improvements to the simulation model can be easily introduced within the current framework. Especially, the simulation of the absorption of drug in the body of the animals can be studied in higher detail with help of an appropriate experimental dataset in which to base some empirical evidence on which to simulate the absorption process.

Some possible applications from this simulation will be the possibility to optimize the amount of drug given in the food, observing how the levels of drug are in bodies of the weaker and more dominant pigs over time.

The results from this study will be shared with other researchers and further developments can be achieved with the help of new empirical data and observation of animals in different environments.
Main findings from this study

Some empirical evidence has been assessed during this study. This has been already described throughout this document, but we will mention again the difference in feeding patterns across the spectrum of dominant to weaker animals and the double-peak observed in the feeding habits during the day as the most relevant properties of the data. The characterization of the animals among dominant (long occupancies and long periods to come back to eat) and subordinated (repetitive feeding habits) is not as clear as we supposed previously to the study. There is a continuous spectrum of characterization among this two types. The model, therefore, had to adapt to this diversity of behavior, since it was not possible to characterize dominance as a clearly binomial variable.

The impossibility to relate feeding behaviour to biometric data is also a major result, since a priori hypotheses on the expected results from the statistical analysis would have predicted the weight or age of the animals to be predicting factors of the feeding habits. However, the lack of information or the weak relation between this observed variables doesn’t allow for a more transparent and simple characterization of the animals.


**Suggested extensions for the study**

Apart from the already intended extensions of this research project, such as the analysis of the absorption of the drug in the body of animals or the optimization of drug supply, it can be suggested after this study to include in the simulation model the following elements which were not initially considered in a simple version of the simulation model:

- **Sleeping patterns**

  One possible way to enforce the generation of the two natural peaks in feeding activity observed in the data and that correspond roughly to the sunrise and the sunset is to introduce during the hours of sleep an inhibitor of the hunger parameter; or, alternatively, to introduce an indicator variable in the estimation of the distribution of time to return back to the feeding trough which took binary values depending on the time of the day (day/night). In this way only the most hungry animals would present more often feeding activity at night, and the entire group would show probably lower feeding intensity during the night and higher intensity at breakfast and dinner times, corresponding to the habits of the dominant animals, who eat a fairly regularly spaced time intervals.

- **Stochastic feeding rate**

  The feeding rate (feeding speed by unit of time) has been seen to be moderately constant, and the resulting intake has been simplified to be proportional to the occupancy time. However, once the model has been successfully validated, further detail can be introduced in the model by allowing the feeding speed vary stochastically and probably letting the probability distribution depend on the type of animal and hunger level by means of discretization. However, this new variable could not be used to determine the types of animals by adding it to the four parameters of the two time distributions of feeding habits, since the curse of dimensionality might limit the accuracy of the estimation of clusters in the data.

- **Analysis of the hierarchical structures and relation to biometric characteristics**

  Although the data in this experiment doesn’t support strongly any theoretical relation between animal weights and dominance or feeding habits, further analysis could done in the patterns from more or less dominant animals and their distribution in the boxes to assess whether some limited number of dominant animals is allowed in each group or if, on the contrary, this behavior is not relative to the group of animals sharing the space and many such animals would preserve their behavior and concentration of this kind would result in continuous fights for occupancy. The data of the experiment, however, doesn’t allow for much analysis on hierarchical structures since only the resulting feeding habits are registered by the devices.

- **Introduction of shrunk dominance scores in the queuing process**

  The Elo ratings were introduced as an attempt to quantify the different dominance ranks when individuals appear to be fighting for occupancy. Apart from in the feeding time distributions, the dominance of the animals is supposed to be reflected in the interactions between animals when these arrive at the feeding trough at similar times. The simulation model in this study contains an element representing the queue of pigs waiting to enter the trough and, according to field experts, the animals might compete to enter the trough, but the animals occupying the trough are not usually kicked out from it. The stochastic positioning in this queue might be dependent on the dominance scores
of the animals present in it. However, we have observed previously in this study that the Elo ratings showed low predictive ability for the feeding time distributions, and this is counterintuitive and should put this indirectly obtained information on dominance hierarchies into question. Arguments have been given previously pointing at the inherent weaknesses to obtain this information from the available data. Therefore, another more simple and robust proxy for dominance, such as the median occupancy time by each animal, could be used. A possibility to preserve some of the information captured by the Elo ratings from the fight interactions between animals as defined in this study is to use an statistical shrinkage of the Elo rating estimates such that the estimated Elo rating is shrunk towards the rank implied by the median occupancy times, to correct their aforementioned bias in this way by a more robust measure, while retaining some of the information captured in that signal.
Appendix
A) \textit{R} code

\# 0-- LOAD DATA AND LIBRARIES

\begin{verbatim}
library(fitdistrplus)
library(reshape)
library(plyr)
library(ggplot2)
library(sqldf)
library(chron)
library(reshape2)
library(moments)
data <- read.csv("Consumptions.csv", stringsAsFactors=FALSE)
weight.data <- read.csv("Weights.csv", stringsAsFactors=FALSE)
\end{verbatim}

\# 1- DATA CLEANSING AND STRUCTURING

\# some lines are duplicated:
tmp <- data
names(tmp) <- c("Fecha", "X", "xip", "box", "pes_i", "pes_f", "consum", "hora_i", "hora_f", "temps")
unique(sqldf(\texttt{select Fecha, X, xip, box, pes_i, pes_f, consum, hora_i, hora_f, temps from tmp group by Fecha, X, xip, box, pes_i, pes_f, consum, hora_i, hora_f, temps having count(*) > 1 order by Fecha, box\}))
rm(tmp)
data <- unique(data)

\# pivot the weights dataset
weights <- melt(weight.data[,c("X1.16.2008","X2.27.2008","X3.12.2008")], id.var=c("Xip","BOX"))
names(weights) <- c("id","box","date","weight")
weights$date <- as.character(weights$date)
weights$id <- as.character(weights$id)
weights[weights$date == "X1.16.2008", "date"] <- as.Date("2008-01-16")
weights[weights$date == "X2.27.2008", "date"] <- as.Date("2008-02-27")
weights[weights$date == "X3.12.2008", "date"] <- as.Date("2008-03-12")
weights$date <- as.Date(as.numeric(weights$date), origin="1970-01-01")
ggplot(weights, aes(date, weight*1e-3, group=id, colour=box)) + geom_line() + labs(x="",y="weight (kg)") + scale_colour_brewer(palette="Set3") + ggtitle("Weight measurements during the year 2008 (linearly interpolated)") + geom_vline(xintercept=as.numeric(as.Date(c("2008-01-16","2008-02-27","2008-03-12"))),linetype = "longdash")
ggplot(weights, aes(date, weight*1e-3, group=id)) + geom_point(colour="dark blue", size=1.5) + geom_line(colour="dark blue") + facet_wrap(~box) + labs(x="",y="weight (kg)") + geom_vline(xintercept=as.numeric(as.Date(c("2008-01-16","2008-02-27","2008-03-12"))),linetype = "longdash") + ggtitle("Weight measurements in the year 2008 (linearly interpolated)")

\# There is an incorrect date in the consumption data: 2009-02-29.
\# Changed here to 2009-03-01 and the subsequent delayed increased by one day.
data.tmp <- subset(data, Fecha != "29/02/2009")
data.tmp$Fecha <- as.Date(data.tmp$Fecha, "%m/%d/%Y", tz="CET")
data.tmp$Fecha <- as.numeric(data.tmp$Fecha)
data.tmp[data.tmp$Fecha > 14303,"Fecha"] <- data.tmp[data.tmp$Fecha > 14303,"Fecha"] + 1
data.insert <- subset(data, Fecha == "29/02/2009")
data.insert$Fecha <- 14304
data <- rbind(data.tmp, data.insert)

74
# There is an incorrect hour 2009-03-28 2:00- 3:00 (time-saving calendar)
# Deleted from the dataset here.
data <-
  subset(data, 
    !as.Date(Fecha, origin = "1970-01-01") == as.Date("2009-03-29") &
    (strptime(hora.i, "%H:%M:%S") >= strptime("2:00:00", "%H:%M:%S")
    & strptime(hora.i, "%H:%M:%S") < strptime("3:00:00", "%H:%M:%S")
    |
    (strptime(data$hora.f, "%H:%M:%S") >= strptime("2:00:00", "%H:%M:%S")
    & strptime(data$hora.f, "%H:%M:%S") < strptime("3:00:00", "%H:%M:%S")
    )
    )
  )

# "Wide-format" dataset
data <-
  with(data, data.frame(
    id = as.character(xip),
    box = as.character(box),
    time.in = as.POSIXct(paste(as.Date(Fecha, origin = "1970-01-01"), hora.i)),
    time.out = as.POSIXct(paste(as.Date(Fecha, origin = "1970-01-01"), hora.f)),
    duration = temps,
    food.in = pes.i,
    food.out = pes.f,
    intake = consum,
    stringsAsFactors=FALSE
  )
data$box <- factor(data$box, levels = as.character(1:30))

# When the entry time is before midnight and the exit time after midnight,
# the hour is not increased in the dataset
#data[data$time.in > data$time.out, "time.out"] <-
#  data[data$time.in > data$time.out, "time.out"] + 24*60*60
# the entry hour is decreased by one day
data[data$time.in > data$time.out, "time.in"] <-
  data[data$time.in > data$time.out, "time.in"] - 24*60*60

# Some durations are wrong: they are re-calculated here:
data$duration <- as.numeric(difftime(data$time.out, data$time.in, units="secs"))

# Some intake data or food weights are not consistent either:
data$intake <- data$food.in - data$food.out

# sometimes the food in the feeding trough has 0 or negative weight. Remove those cases (the pigs would seem to be not hungry)
tmp <- subset(data, food.in < 0 | food.out < 0)
tmp$date <- as.Date(tmp$time.in)
unique(tmp[,c("box","date")])
rm(tmp)
ggplot(subset(data, box == "2"), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + labs(x="", y="food") + ggtitle("Food in box 2")

ggplot(subset(data, box == "26"), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + labs(x="", y="food") + ggtitle("Food in box 26")

ggplot(subset(data, as.Date(time.in) >= as.Date("2009-01-18") & as.Date(time.in) <= as.Date("2009-01-20")), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + facet_wrap(~box) + labs(x="", y="food") + ggtitle("Food in each box from Jan 18 through Jan 20") + theme(axis.text.x=element_text(size=6, angle=45))

ggplot(subset(data, box %in% character(3:10) & as.Date(time.in) == as.Date("2009-01-18")), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + facet_wrap(~box) + labs(x="", y="food") + ggtitle("Food in each box from Jan 18 through Jan 20")

ggplot(subset(data, box == "11" & as.Date(time.in) == as.Date("2009-01-25")), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + facet_wrap(~box) + labs(x="", y="food") + ggtitle("Food in box 11 on January 25")

ggplot(subset(data, box == "3" & as.Date(time.in) == as.Date("2009-04-07")), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + labs(x="", y="food") + ggtitle("Food in box 3 on April 7")

ggplot(subset(data, box %in% c(3,24) & as.Date(time.in) == as.Date("2009-04-11")), aes(time.in, food.in)) + geom_point(size=1, colour="dark blue") + facet_wrap(~box) + labs(x="", y="food") + ggtitle("Food in boxes 3 and 24 on April 11")
data <- subset(data, !(box == 26 & time.in < as.POSIXct("2009-01-14 12:00:0")))
data <- subset(data, food.in > 0.5)

# remove outliers

ggplot(data, aes(intake)) + geom_density() + ggtitle("Intake distribution during registered events\nin the raw dataset")
data <- subset(data, intake > 0 & intake < 2)

# Re-order the data
data <- data[order(data$id, data$time.in),]

# Some chip identifiers were replaced. Use the new ones.

chip.mapping <-
  data.frame(
    box=c(14, 14, 15, 17, 18, 23, 23, 23, 24),
    old.chip=as.character(c(18288051, 17695378, 107244487, 16845074, NA, 18495272, NA, NA, 16793661, 107244540)),
    new.chip=as.character(c(107244099, 107244146, 18495325, 107244671, 18495776, 107244617, 20105505, 17998379, 107244352, 18495737)),
    stringsAsFactors=FALSE
  )
data <- merge(data, chip.mapping[,c("old.chip", "new.chip")], by.x="id", by.y="old.chip", all.x=TRUE)
data$new.chip <- NULL
weights <- merge(weights, chip.mapping[, c("old.chip","new.chip")], by.x="id", by.y="old.chip", all.x=TRUE)
weights$new.chip <- NULL
weights$new.chip <- weights[!is.na(weights$new.chip),"id"]
data$new.chip <- data$new.chip[!is.na(data$new.chip),"new.chip"]
weights$new.chip <- NULL

# some descriptive statistics (need to rename some columns with dots in the names)
tmp <- data
tmp[names(tmp)="time.in"] <- "time_in"
tmp[names(tmp)="time.out"] <- "time_out"
stats <- sqldf(
"select
  box,
  count(distinct id) as nr_ids, count(*) as nr_records,
  min(time_in) as start, max(time_out) as end
from tmp
  group by box
  order by box;"
)
rm(tmp)
stats$start <- as.POSIXct(stats$start, origin = "1970-01-01")
stats$end <- as.POSIXct(stats$end, origin = "1970-01-01")
stats <- arrange(stats, box)

# some animals belong to more than one box:
stats <- sqldf("select id, count(distinct box) as nr_boxes from data group by id having count(distinct box) > 1")

# create some variables that will be useful to analyse the data
calculate.box.variables <- function(x)
  with(arrange(x,time.in),data.frame(
    time.between.entries = c(NA,diff(time.in)),
    time.between.exits = c(NA,diff(time.out)),
    time.between.exit.and.entry = c(as.POSIXct(NA),time.in[-1]) - c(as.POSIXct(NA),time.out[-nrow(x)]),
    food.diff.between.entries = c(NA,diff(food.in)),
    food.diff.between.exits = c(NA,diff(food.out)),
    food.diff.between.exit.and.entry = c(NA,food.in[-1]) - c(NA,food.out[-nrow(x)]),
    prev.food.in = c(NA,food.in[-nrow(x)]),
    prev.food.out = c(NA,food.out[-nrow(x)])
  ))
data <- ddply(data, .(box), calculate.box.variables)

# create variable for each individual
calculate.individual.variables <- function(x)
  with(arrange(x,time.in),data.frame(
    time.between.individual.entries = c(NA,diff(time.in)),
    time.between.individual.exits = c(NA,diff(time.out)),
    time.between.individual.exit.and.entry = as.numeric(c(as.POSIXct(NA),time.in[-1]) - c(as.POSIXct(NA),time.out[-nrow(x)]))
  ))
data <- ddply(data, .(id), calculate.individual.variables)
subset(data, time.between.individual.exit.and.entry < 0) # three lines are wrong (they correspond to entries before midnight and exits after midnight on 2009-03-28)
```r
# impute missing weights
tmp <- subset(weights, !is.na(weight))
ids.missing.dates <- sqldf("select id from tmp group by id having count(*) < 3 order by id")[,1]
subset(weights, id %in% ids.missing.dates)

weight.regr.data <- cast(weights, box+id~date, value="weight")
w2.on.w1 <- lm(w2 ~ w1, data=weight.regr.data)$coeff
w3.on.w2 <- lm(w3 ~ w2, data=weight.regr.data)$coeff
weight.regr.data[is.na(weight.regr.data$w2),"w2"] <-
  w2.on.w1[1] + weight.regr.data[is.na(weight.regr.data$w2),"w1"]*w2.on.w1[2]
weight.regr.data[is.na(weight.regr.data$w3),"w3"] <-
  w3.on.w2[1] + weight.regr.data[is.na(weight.regr.data$w3),"w2"]*w3.on.w2[2]

# attach the weights dataset to the consumption dataset
# (the last measurement only, since they previous to the consumption dataset)
data <- merge(data, data.frame(id=weight.regr.data$id, weight=weight.regr.data$w3), all.x=TRUE, all.y=FALSE)

# count how many pigs have weight data in each box
( no.weight.ids <- setdiff(unique(data$id),unique(weights$id)) )
( weight.ids <- intersect(unique(data$id),unique(weights$id)) )
( no.data.ids <- setdiff(unique(weights$id),unique(data$id)) )
count.weight.data <- unique(data[data$id %in% weight.ids,c("box","id"))
count.weight.data$has.weight <- 1
count.weight.data$has.no.weight <- 0
count.weight.data$total <- 1
count.no.weight.data <- unique(data[data$id %in% no.weight.ids,c("box","id"))
count.no.weight.data$has.weight <- 0
count.no.weight.data$has.no.weight <- 1
count.no.weight.data$total <- 1
count.weight.data.by.box <- rbind(count.weight.data, count.no.weight.data)
aggregate(~box, count.weight.data.by.box[-2], sum)

# create extra time columns (days, hours, times - in addition to the original datetimes)
data$date.in <- as.Date(data$time.in, tz="CET")
data$date.out <- as.Date(data$time.out, tz="CET")
data$hour.in <- as.numeric(substr(data,"time.in",12,13))
data$hour.out <- as.numeric(substr(data,"time.out",12,13))
data$daytime.in <- as.POSIXct(paste(2000-01-01",substr(data,"time.in",12,19)))
data$daytime.out <- as.POSIXct(paste(2000-01-01",substr(data,"time.out",12,19)))
data$month.in <- substr(data$daytime.in,1,7)

# "long-format" dataset
data.long <-
  with(data,
  rbind(data.frame(
    id=id,
    box=box,
  )
  )
```

event="in",
time=time.in,
food=food.in,
time.lapse.last.same.event=time.between.entries,
time.lapse=time.between.exit.and.entry,
food.change.last.same.event=food.diff.between.entries,
food.change=food.diff.between.exit.and.entry,
prev.food.same.event=prev.food.in,
prev.food=prev.food.out
}
data.frame(
   id=id,
   box=box,
   event="out",
time=time.out,
food=food.out,
time.lapse.last.same.event=time.between.exits,
time.lapse=duration,
food.change.last.same.event=food.diff.between.exits,
food.change=intake,
prev.food.same.event=prev.food.out,
prev.food=food.in)
)
data.long <- arrange(data.long, box, time, desc(event))
ggplot(subset(data.long, box == 19 & as.POSIXct("2009-02-01 13:00:00") < time & time <
as.POSIXct("2009-02-01 14:00:00")), aes(time, food, colour=id, shape=event)) + geom_point() + xlab(""

ggplot(subset(data.long, box == 19 & as.POSIXct("2009-02-01 17:00:00") < time & time <
as.POSIXct("2009-02-01 18:00:00")), aes(time, food, colour=id, shape=event)) + geom_point() + xlab(""

# look at box 19 (it has weight data available for all the 14 pigs)
period.id.stats <-
ddply(
data,
   .(box, id, weight),
   function(x)
data.frame(
total.intake=sum(x$intake),
total.occupancy=sum(x$duration)/60/60,
avg.intake=sum(x$intake)/length(unique(x$date.in)),
avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
hunger=sum(x$intake)/sum(x$duration)*60
))

# A) statistics by animal during the period
period.id.stats$id <- factor(as.character(period.id.stats$id),
   levels=unique(as.character(period.id.stats$[order(period.id.stats$total.intake,"id")]))
ggplot(period.id.stats, aes(id, total.intake, fill=weight)) + geom_bar(stat="identity") + facet_wrap(~box, scales="free_x") +
xlab("") + ylab("Kg") + ggttitle("Total intake by pig") +
theme(axis.text.x=element_text(angle=90,size=6))
period.id.stats$id <- factor(as.character(period.id.stats$id),
   levels=unique(as.character(period.id.stats$[order(period.id.stats$avg.intake,"id")]))
ggplot(period.id.stats, aes(id, avg.intake, fill=weight)) + geom_bar(stat="identity") +
facet_wrap(~box, scales="free_x") +
xlab("") + ylab("Kg") + ggttitle("Average daily intake by pig") +
theme(axis.text.x=element_text(angle=90,size=6))
```
period.id.stats$id <- factor(as.character(period.id.stats$id),
  levels=unique(as.character(period.id.stats$order(period.id.stats$savg.intake/period.id.stats$weight),"id")))
ggplot(period.id.stats, aes(id, avg.intake/weight*1000*100, fill=weight)) + geom_bar(stat="identity") + facet_wrap(~box, scales="free_x") + xlab("") + ylab("") + ggtitle("Average daily intake (as % of weight) by pig") + theme(axis.text.x=element_text(angle=90, size=6))

period.id.stats$id <- factor(as.character(period.id.stats$id),
  levels=unique(as.character(period.id.stats$order(period.id.stats$total.occupancy),"id")))
ggplot(period.id.stats, aes(id, total.occupancy, fill=weight)) + geom_bar(stat="identity") + facet_wrap(~box, scales="free_x") + xlab("") + ylab("Hours") + ggtitle("Total occupancy by pig") + theme(axis.text.x=element_text(angle=90, size=6))

period.id.stats$id <- factor(as.character(period.id.stats$id),
  levels=unique(as.character(period.id.stats$order(period.id.stats$avg.occupancy),"id")))
ggplot(period.id.stats, aes(id, avg.occupancy, fill=weight)) + geom_bar(stat="identity") + facet_wrap(~box, scales="free_x") + xlab("") + ylab("Minutes") + ggtitle("Daily average occupancy by pig") + theme(axis.text.x=element_text(angle=90, size=6))

period.id.stats$id <- factor(as.character(period.id.stats$id),
  levels=unique(as.character(period.id.stats$order(period.id.stats$hunger),"id")))
ggplot(period.id.stats, aes(id, hunger*60/weight*1000*100, fill=weight)) + geom_bar(stat="identity") + facet_wrap(~box, scales="free_x") + xlab("") + ylab("%/min") + ggtitle("Hunger (intake as % of weight / occupancy) by pig") + theme(axis.text.x=element_text(angle=90, size=6))

period.id.stats$id <- factor(as.character(period.id.stats$id),
  levels=unique(as.character(period.id.stats$order(period.id.stats$savg.occupancy),"id")))
ggplot(period.id.stats, aes(avg.intake/weight*1000*100, hunger/weight)) + geom_point(size=1) + ggtitle("Intake as % of weight vs Hunger as % of weight") + geom_smooth() + xlab("Daily intake as % of weight") + ylab("Hunger (as % of weight / min)")

# B) intraday patterns in the consumption of all pigs
intraday.all.stats <- 
  ddply(
```

```r
# A) intraday patterns for each box
intraday.all.stats <- ddply(data, 
  .(box, hour.in), 
  function(x) 
    data.frame(
      total.intake=sum(x$intake),
      total.occupancy=sum(x$duration)/60/60,
      avg.intake=sum(x$intake)/length(unique(x$date.in)),
      avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
      hunger=sum(x$intake)/sum(x$duration)*60,
      nr.competitors=length(unique(x$id))
    ))

ggplot(intraday.all.stats,aes(hour.in, total.intake)) + geom_bar(stat="identity", fill="dark blue") + facet_grid( box ~ . ) + xlab("hour of the day") + ylab("total intake (kg)") + ggtitle("Hourly total consumption by box")

ggplot(intraday.all.stats,aes(hour.in, total.intake, colour=box)) + geom_line() + xlab("hour of the day") + ylab("total intake (kg)") + ggtitle("Hourly total consumption by box")

intraday.id.stats <- ddply(data, 
  .(box, hour.in, total.intake), 
  function(x) 
    data.frame(
      total.intake=sum(x$intake),
      total.occupancy=sum(x$duration)/60/60,
      avg.intake=sum(x$intake)/length(unique(x$date.in)),
      avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
      hunger=sum(x$intake)/sum(x$duration)*60,
      nr.competitors=length(unique(x$id))
    ))

# C) intraday patterns for each pig
intraday.id.stats <- ddply(data, 
  .(box, id, weight, hour.in), 
  function(x) 
    data.frame(
      total.intake=sum(x$intake),
      total.occupancy=sum(x$duration)/60/60,
      avg.intake=sum(x$intake)/length(unique(x$date.in)),
      avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
      hunger=sum(x$intake)/sum(x$duration)*60,
      nr.competitors=length(unique(x$id))
    ))

ggplot(subset(intraday.id.stats, box==19), aes(hour.in, avg.intake, group=weight,colour=weight)) + geom_line() + xlab("hour of the day") + ylab("intake (kg)") + ggtitle("Hourly average intakes by box")
```

81
animal in box 19")

```r
ggplot(subset(intraday.id.stats, box==19), aes(hour.in, avg.intake/weight*1000*100, colour=id)) +
  geom_line() + xlab("hour of the day") + ylab("intake as % of weight") + ggtitle("Hourly average
intakes by animal in box 19")
ggplot(subset(intraday.id.stats, box==19), aes(hour.in, avg.intake/weight*1000*100,
group=weight, colour=weight)) +
  geom_line() + xlab("hour of the day") + ylab("intake as % of weight") + ggtitle("Hourly average
intakes by animal in box 19")
ggplot(subset(intraday.id.stats, box==19), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
19")
ggplot(subset(intraday.id.stats, box==13), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
13")
ggplot(subset(intraday.id.stats, box==14), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
14")
ggplot(subset(intraday.id.stats, box==15), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
15")
ggplot(subset(intraday.id.stats, box==16), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
16")
ggplot(subset(intraday.id.stats, box==17), aes(hour.in, avg.intake, fill=weight)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ggtitle("Hourly average intakes by animal in box
17")
ggplot(subset(intraday.id.stats, box==19), aes(hour.in, avg.occupancy, fill=weight)) +
  geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ylab("occupancy (min)") + ggtitle("Hourly average
occupancy by animal in box 19")
ggplot(subset(intraday.id.stats, box==19), aes(hour.in, hunger, fill=box)) + geom_bar(stat="identity") +
  facet_grid( id ~ . ) + xlab("hour of the day") + ylab("hunger kg/min") + ggtitle("Hourly average
hunger by animal in box 19")
```

# C b) intraday patterns for a pig across several days (faceted) to see if it is stable

```r
intraday.week.id.stats <-
  ddply(
    subset(data, id="20111915"),
    .(box, id, date.in, weight, hour.in),
    function(x)
      data.frame(
        total.intake=sum(x$intake),
        total.occupancy=sum(x$duration)/60/60,
        avg.intake=sum(x$intake)/length(unique(x$date.in)),
        avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
        hunger=sum(x$intake)/sum(x$duration)*60
      ))
```
intraday.all.stats.by.month <-
  ddply(
    data,
    list(box, month.in, hour.in),
    function(x)
      data.frame(
        total.intake=sum(x$intake),
        total.occupancy=sum(x$duration)/60/60,
        avg.intake=sum(x$intake)/length(unique(x$date.in)),
        avg.occupancy=sum(x$duration)/length(unique(x$date.in))/60,
        hunger=sum(x$intake)/sum(x$duration)*60,
        nr.competitors=length(unique(x$id))
      )
  )

ggplot(intraday.all.stats.by.month,aes(hour.in, avg.occupancy)) + geom_bar(stat="identity", fill="dark blue") + facet_grid( month.in ~ . ) +
  xlab("hour of the day") + ylab("average occupancy (min)") + ggtitle("Hourly time occupancy by month")

ggplot(subset(intraday.all.stats.by.month, box%in%c(1,2,3,4,5)),aes(hour.in, avg.occupancy)) + geom_bar(stat="identity", fill="dark blue") + facet_grid( month.in ~ box ) +
  xlab("hour of the day") + ylab("average occupancy (min)") + ggtitle("Hourly time occupancy by box and month")

# DISTRIBUTIONS

# Distribution of time to the next visit: GetTimeToHungry(int type)
# to classify clusters of individuals (types) I will use raw statistics (non-parametric distributions) and see what distribution family each cluster follows
# with scale

calc.id.stats <-
  function(x)
  {
    TimeToHungry <- coef(fitdist(as.numeric(na.omit(x$next.return)), "gamma", method="mme"))
    TimeToEat <- coef(fitdist(x$duration, "gamma", method="mme"))
    data.frame(
      avg.nr.daily.events=nrow(x)/length(unique(x$date.in)),
      avg.daily.stay=sum(x$duration)/length(unique(x$date.in)),
      mean.stay=mean(x$duration),
      median.stay=median(x$duration),
      mad.stay=mad(x$duration),
      variance.stay=var(x$duration),
      skewness.stay=skewness(x$duration),
      kurtosis.stay=kurtosis(x$duration),
      avg.daily.intake=sum(x$intake)/length(unique(x$date.in)),
      mean.comeback=mean(x$time.between.individual.exit.and.entry),
      median.comeback=median(x$time.between.individual.exit.and.entry),
      mad.comeback=mad(x$time.between.individual.exit.and.entry),
      variance.comeback=var(x$time.between.individual.exit.and.entry),
      skewness.comeback=skewness(x$time.between.individual.exit.and.entry),
      kurtosis.comeback=kurtosis(x$time.between.individual.exit.and.entry),
      corr.comeback.stay=cor(x$time.between.individual.exit.and.entry, x$duration),
      avg.daily.stay.time=sum(x$duration),
      avg.intake.speed=60*60*sum(x$intake)/sum(x$duration),
    )
  }
max.intake = max(x$intake),
max.intake.pct.weight = max(x$intake)/(x$weight[1]/1000),
TimeToHungry.shape = TimeToHungry[1],
TimeToHungry.scale = 1/TimeToHungry[2],
TimeToEat.shape = TimeToEat[1],
TimeToEat.scale = 1/TimeToEat[2]

}

id.stats <- ddply(data, .(id, weight), calc.id.stats)
ggplot(id.stats, aes(TimeToHungry.shape)) + geom_density()
ggplot(id.stats, aes(TimeToHungry.scale)) + geom_density()
ggplot(id.stats, aes(TimeToEat.shape)) + geom_density()
ggplot(id.stats, aes(TimeToEat.scale)) + geom_density()
ggplot(id.stats, aes(TimeToEat.shape, TimeToEat.scale)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToHungry.scale)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.shape)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.scale, TimeToEat.shape)) + geom_point()
biplot(princomp(na.omit(id.stats[,c("TimeToHungry.shape","TimeToHungry.scale","TimeToEat.shape","TimeToEat.scale")])))

library(adegenet)
grp <- find.clusters(na.omit(subset(id.stats, !is.na(TimeToHungry.scale))[c("TimeToHungry.shape","TimeToHungry.scale","TimeToEat.shape","TimeToEat.scale")]))
dapc1 <- dapc(na.omit(id.stats[,c("TimeToHungry.shape","TimeToHungry.scale","TimeToEat.shape","TimeToEat.scale")]), grp$grp)
scatter(dapc1)
id.stats$cluster <- NA
id.stats[[is.na(id.stats$TimeToHungry.scale),"cluster"]]<-
kmmeans(subset(id.stats, !is.na(TimeToHungry.scale))[c("TimeToHungry.shape","TimeToHungry.scale","TimeToEat.shape","TimeToEat.scale")], 8)$cluster
id.stats$cluster <- as.character(id.stats$cluster)
ggplot(id.stats, aes(TimeToEat.shape, TimeToEat.scale, colour=cluster)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToHungry.scale, colour=as.character(cluster)))+ geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.shape, colour=cluster)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.scale, TimeToEat.scale, colour=cluster)) + geom_point()
  # with rate
calc.id.stats <- function(x)
  {
    TimeToHungry <- coef(fitdist(as.numeric(na.omit(x$next.return)), "gamma", methodology="mme")
    TimeToEat <- coef(fitdist(x$duration, "gamma", method="mme")
    data.frame(
      avg.nr.daily.events=nrow(x)/length(unique(x$date.in)),
      avg.daily.stay=sum(x$duration)/length(unique(x$date.in)),
      mean.stay=mean(x$duration),
      median.stay=median(x$duration),
      mad.stay=mad(x$duration),
      variance.stay=var(x$duration),
      skewness.stay=skewness(x$duration),
      kurtosis.stay=kurtosis(x$duration),
      avg.daily.intake=sum(x$intake)/length(unique(x$date.in)),
      mean.comeback=mean(x$time.between.individual.exit.and.entry),
      median.comeback=median(x$time.between.individual.exit.and.entry),
      mad.comeback=mad(x$time.between.individual.exit.and.entry),
...
variance.comback=var(x$time.between.individual.exit.and.entry),
skewness.comback=skewness(x$time.between.individual.exit.and.entry),
kurtosis.comback=kurtosis(x$time.between.individual.exit.and.entry),
corr.comback.stay=cor(x$time.between.individual.exit.and.entry, x$duration),
avg.daily.stay.time=sum(x$duration),
max.intake=max(x$intake),
max.intake.pct.weight=max(x$intake)/(x$weight[1]/1000),
TimeToHungry.shape=TimeToHungry[1],
TimeToHungry.rate=TimeToHungry[2],
TimeToEat.shape=TimeToEat[1],
TimeToEat.rate=TimeToEat[2])

id.stats <- ddply(data, .(id, weight), calc.id.stats)
ggplot(id.stats, aes(TimeToHungry.shape)) + geom_density()
ggplot(id.stats, aes(TimeToHungry.rate)) + geom_density()
ggplot(subset(id.stats, TimeToHungry.rate < 0.001), aes(TimeToHungry.rate)) + geom_density()
id.stats[id.stats$TimeToHungry.rate > 0.001, "TimeToHungry.rate"] <- NA
ggplot(id.stats, aes(TimeToEat.shape)) + geom_density()
ggplot(id.stats, aes(TimeToEat.rate)) + geom_density()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.shape)) + geom_density()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.rate)) + geom_density()
biplot(princomp(na.omit(id.stats[,c("TimeToHungry.shape","TimeToHungry.rate","TimeToEat.shape","TimeToEat.rate")])))
library(adegenet)
grp <- find.clusters(na.omit(subset(id.stats, !is.na(TimeToHungry.rate))[,c("TimeToHungry.shape","TimeToHungry.rate","TimeToEat.shape","TimeToEat.rate")]), 7)
dapc1 <- dapc(na.omit(id.stats[,c("TimeToHungry.shape","TimeToHungry.rate","TimeToEat.shape","TimeToEat.rate")]), grp$grp)
scatter(dapc1)
id.stats$cluster <- NA
id.stats[[is.na(id.stats$TimeToHungry.rate), "cluster"]]
  kmeans(subset(id.stats, is.na(TimeToHungry.rate))[,c("TimeToHungry.shape","TimeToHungry.rate","TimeToEat.shape","TimeToEat.rate")], 7)$cluster
id.stats$cluster <- as.character(id.stats$cluster)
ggplot(id.stats, aes(TimeToEat.shape, TimeToEat.rate, colour=cluster)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToHungry.rate, colour=as.character(cluster))) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.shape, colour=cluster)) + geom_point()
ggplot(id.stats, aes(TimeToHungry.shape, TimeToEat.shape, colour=cluster)) + geom_point()
tmp < id.stats; sqldf("select cluster, count(id) from tmp group by cluster") # using 8 clusters yields very
thin clusters not suitable for estimation of time distributions

PCbiplot <- function(PC, clusters, x="PC1", y="PC2", colour="cluster", colors=c('black', 'black', 'red', 'red'))
{
  # PC being a prcomp object
data <- data.frame(obsnames=row.names(PC$x), PC$x, cluster=clusters)
plot <- ggplot(data, aes(PC1, PC2, colour=cluster)) + geom_text(alpha=1, size=3, aes(label=obsnames))
  + scale_colour_brewer(palette="Set3")
plot <- plot + geom_hline(aes(0), size=.2) + geom_vline(aes(0), size=.2, color=colors[2])
datapc <- data.frame(varnames=rownames(PC$rotation), PC$rotation)
mult <- min(
  {max(data[,y]) - min(data[,y])/(max(datapc[,y])-min(datapc[,y]))},

85
\[ \frac{\max(\text{data}[x]) - \min(\text{data}[x])}{\max(\text{datapc}[x]) - \min(\text{datapc}[x])} \]

datapc <- transform(datapc,
  \[ v1 = 0.7 \times \text{mult} \times (\text{get}(x)), \]
  \[ v2 = 0.7 \times \text{mult} \times (\text{get}(y)) \]
)

plot <- plot + coord_equal() + geom_text(data=datapc, aes(x=v1, y=v2, label=varnames), size = 5, vjust=1, color=colors[3])
plot <- plot + geom_segment(data=datapc, aes(x=0, y=0, xend=v1, yend=v2), alpha=0.75, color=colors[4])
plot

fit <- prcomp(na.omit(id.stats[,c("TimeToHungry.shape","TimeToHungry.rate","TimeToEat.shape","TimeToEat.rate")]), scale=T)
PCbiplot(fit, id.stats[!is.na(id.stats$TimeToHungry.rate), "cluster"], colors=c("black", "black", "red", "yellow"))

data <- merge(data, id.stats[,c("id", "avg.daily.intake", "cluster")], by="id")
data$hunger.out <- data$hunger.in <- NA # data <- data.bkp; data <- subset(data, id %in% unique(data$id)[60:63])
for (i in unique(data$id))
{
  tmp.id <- arrange(data[data$id == i,], time.in)
  for (t in tmp.id$time.in)
  {
    tmp.t <- tmp.id[tmp.id$time.in == t]
    hunger.in <- 1 - min(1 - max(prev.hunger.out, 0), 1)*exp(-
    tmp.t$time.between.individual.exit.and.entry/10000)
    hunger.out <- \min\left(1 - \text{hunger.in}\right) \times \exp\left(-\frac{\text{duration}}{10000}\right) - \frac{\text{intake}}{1.5 \times \text{avg.daily.intake}/3}, 0)\right)
    data[data$id == i & data$time.in == t, "hunger.in"] <- hunger.in
    data[data$id == i & data$time.in == t, "hunger.out"] <- hunger.out
    prev.hunger.out <- hunger.out
  }
}
data <- ddply(data, .(id), function(x) data.frame(x, next.return=x$Time.between.individual.exit.and.entry[-1], NA))
data$hunger.in.decile <- as.character(data$hunger.in%/%0.1)
data[data$hunger.in.decile=="10","hunger.in.decile"] <- "9"
data$hunger.out.decile <- as.character(data$hunger.out%/%0.1)
data[data$hunger.out.decile=="10","hunger.out.decile"] <- "9"

calc.type.TimeToEat.parameters <- function(x)
{
  TimeToEat <- coef(fitdist(x$duration, "gamma", method="mme"))
data.frame(
    TimeToEat.shape=TimeToEat[1],
    TimeToEat.rate=TimeToEat[2]
  )
}
calc.type.TimeToHungry.parameters <- function(x)
{
  TimeToHungry <- coef(fitdist(as.numeric(na.omit(x$next.return)), "gamma",}
method="mme")
    data.frame(
        TimeToHungry.shape=TimeToHungry[1],
        TimeToHungry.rate=TimeToHungry[2]
    )
}
tmp <- data; tmp$hunger <- tmp$hunger.in.decile; sqldf("select cluster, hunger, count(*) from tmp group by cluster, hunger"); rm(tmp)
tmp <- data; tmp$hunger <- tmp$hunger.out.decile; sqldf("select cluster, hunger, count(*) from tmp group by cluster, hunger"); rm(tmp)
time.to.eat.params <- ddply(subset(data, !is.na(cluster)), .(cluster, hunger.in.decile),
calc.type.TimeToEat.parameters) # problem: too thin cluster 4 and low hunger level
time.to.hungry.params <- ddply(subset(data, !is.na(cluster)), .(cluster, hunger.out.decile),
calc.type.TimeToHungry.parameters) # problem: too thin cluster 4 and low hunger level
ggplot(data, aes(duration/60)) + geom_density() + xlim(0,30) + xlab("TimeToEat (minutes)")
ggplot(data, aes(duration/60, colour=cluster)) + geom_density() + xlim(0,30) + xlab("TimeToEat (minutes)")
ggplot(subset(data, cluster %in% c(7)), aes(duration/60, colour=hunger.in.decile)) + geom_density() +
    xlim(0,30) + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(subset(data, cluster %in% c(3)), aes(duration/60, colour=hunger.in.decile)) + geom_density() +
    xlim(0,30) + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(subset(data, cluster %in% c(6)), aes(duration/60, colour=hunger.in.decile)) + geom_density() +
    xlim(0,30) + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(data, aes(time.between.individual.exit.and.entry/60/60, colour=cluster)) + geom_density() +
    xlab("TimeTohunger (hours)") + xlim(0,7)
ggplot(subset(data, cluster %in% c(7)), aes(time.between.individual.exit.and.entry/60/60,
    colour=hunger.out.decile)) + geom_density() + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(subset(data, cluster %in% c(3)), aes(time.between.individual.exit.and.entry/60/60,
    colour=hunger.out.decile)) + geom_density() + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(subset(data, cluster %in% c(6)), aes(time.between.individual.exit.and.entry/60/60,
    colour=hunger.out.decile)) + geom_density() + xlab("TimeToEat (minutes)") + scale_colour_discrete(name="Hunger\nLevel")
ggplot(time.to.eat.params, aes(hunger.in.decile, TimeToEat.shape, colour=cluster)) + geom_line() +
    scale_colour_discrete(name="Type") + xlab("Hunger Level")
ggplot(time.to.eat.params, aes(hunger.in.decile, TimeToEat.rate, colour=cluster)) + geom_line() +
    scale_colour_discrete(name="Type") + xlab("Hunger Level")
ggplot(time.to.hungry.params, aes(hunger.out.decile, TimeToHungry.shape, colour=cluster)) + geom_line() +
    scale_colour_discrete(name="Type") + xlab("Hunger Level")
ggplot(time.to.hungry.params, aes(hunger.out.decile, TimeToHungry.rate, colour=cluster)) + geom_line() +
    scale_colour_discrete(name="Type") + xlab("Hunger Level")
**B) SDL diagrams**

The following diagrams are constructed in Microsoft Visio and implement the simulation expressed in SDL language. A plug-in for SDLPS in Visio generates and verifies automatically the code. The system is divided in blocks and processed.

**B.1 The system**

![Diagram of the system](image)
B.2 Block BPorc:

The Process PPorc has three states:

IDLE:

WAITING:
**B.3 Block: Feeding trough**

The process Feeding trough has two states:

**FREE**

**process** PFeedingTrough

DCL

```c
int freeSlots = 10;
double foodQe=0;
int dominance=0;
int queueNum=0;
char * Pid;
```

FREE

FeedingRequest(dominance)

```c
freeSlots = freeSlots -1;
```

FreeFeeding to SENDER

FREE > 0

true

FREE

false

FULL

**FULL**
B4 Process: PFarm

PROCESS PFarm

DCL
int Pporec_number=10;

Pporec_number=0
  true
  Pporec
  Pporec_number-1
  IDLE

IDLE

* IDLE

IDLE
B.5 Medicine ingestion:

```plaintext
procedure PRMedicineIngestion( float A0, float Ka, float C0, float Ke, float t, float C)

C = C0*(e^{(-Ke*t)})-A0*(e^{(-Ka*t)});
```

94
Bibliography

