Pododermatitis in Swiss group housing: A longitudinal cluster study

---

Master Thesis in Biostatistics (STA495)

by

Audrey YEO Te-ying
15-347-602

supervised by

Assoc. Prof. Reinhard Furrer

Zurich, August 2020

Pododermatitis in Swiss group housing: A longitudinal cluster study

Audrey YEO Te-ying

Version 25 August 2020

Contents

Preface

For those I have to thank;

It was the novel that got me to the finish line, rather than the lecture.

Audrey Yeo Te-ying
August 2020

Abstract

Pododermatitis is a debilitating skin disease in many rodents, birds and rabbits. In rabbits, the worsening of the condition sometimes lead to premature slaughter by farmers. Nevertheless the welfare of rabbits are in question, and risk factors such as weight, age, claw length, have been found to impact the progress of this disease through a multivariate analysis by Ruchti *et al.* (2019). A new finding from their research is that disease may improve and worsen over time. A dataset has been collected within one year about the progression of pododermatitis in "does", which are female rabbits. This dataset is contained disease scores and covariates over 13 time points. The goal of this longitudinal study is to find out more about the progression of pododermatitis in the Swiss group housing systems. Furthermore, to identify risk factor associated with the occurrence of pododermatitis in these particular housing systems.

This interdisciplinary thesis will also involve assessing the state of the art clustering of longitudinal data by means of the `kml`, which will also be featured.

Audrey Yeo Te-ying
August 2020

Chapter 1

Introduction

Primary clinical assessment of rodents, birds and rabbits often begin with assessment of the organism's morphology. A common dermatological problem among rabbits are pressure sores on hocks and feet, otherwise known as pododermatitis (Mancinelli *et al.*, 2014). This skin disease starts with a reddish area, loss of fur, progressing to breakage of the does' skin barrier integrity, chronic granulomatous and ulcerative dermatitis (Ruchti *et al.*, 2019). Clinicians approximate several factors that contribute to this painful condition, such as age, claw length and environmental factors such as temperature and humidity (Drescher and Schlender-Bobbis, 1996, Martorell, 2014, Rommers and Meierhod, 1996 and Rosell and De la Fuente, 2009). Pododermatitis has several secondary effects including poor pedal function and severe loss of quality of life for the animal resulting in general poor welfare, sometimes relieved by premature slaughter (Seaman *et al.*, 2013). Categorically, the risk factors of this disease include

- anatomical factors: claw length
- physiological factors: age, parity, body weight, reproductive state, hybrid
- environmental factors: cleanliness, moisture of paws, temperature and humidity

While genetic factors predetermine anatomical and physiological factors, housing factors need to be considered. Some literature about the disease of pododermatitis is based on European housing styles with mesh floors, as cited in Ruchti *et al.* (2019). In Switzerland, group housing of rabbits occur in pens with litter and plastic slats that allows positive social contacts (Seaman *et al.*, 2008). The precedent study by Ruchti *et al.* (2018) reported a range of disease scores in these types of housing and an incidence rate between four and 49 % in their cross sectional analysis in Switzerland. An example of a group housing pen is seen in Figure 1.1. A multivariable analysis, with implementation of the additive Bayesian network or `abn` package (Kratzer *et al.*, 2019) has subsequently been performed and potential risk factors and their associations were visualised in a directed acyclic graph (dag) seen in Figure 1.2 taken from the Ruchti *et al.* (2019) study. This study noted positive effects of temperature, humidity and weight whereas negative effects of age (through weight) in mean pododermatitis scores and cleanliness. Multinomial effects from claw length were found to influence scores (Ruchti *et al.*, 2018). Age appears to play an indirect effect on weight and hybrid which directly affects the heel's disease severity. In addition, claw length has a multinomial effect on age and score. I use the data from the precedent study to understand the longitudinal influence of these factors in a clustered setting, thus my study aims to

- understand which factors influence the healing process over thirteen time points and furthermore,
- apply longitudinal cluster analysis in order to understand evolution of clusters (disease states) and time dependences.

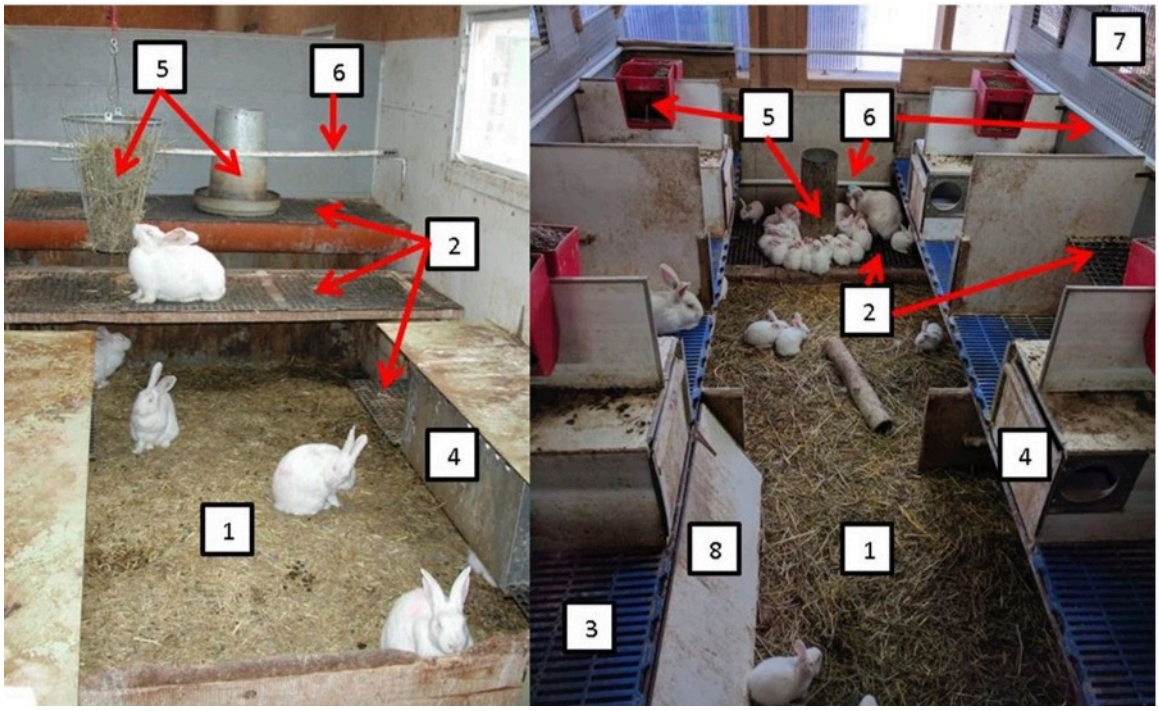

Figure 1.1: Example of group housing systems in a pen with plastic flooring and elevated areas taken from Ruchti *et al.* (2019).

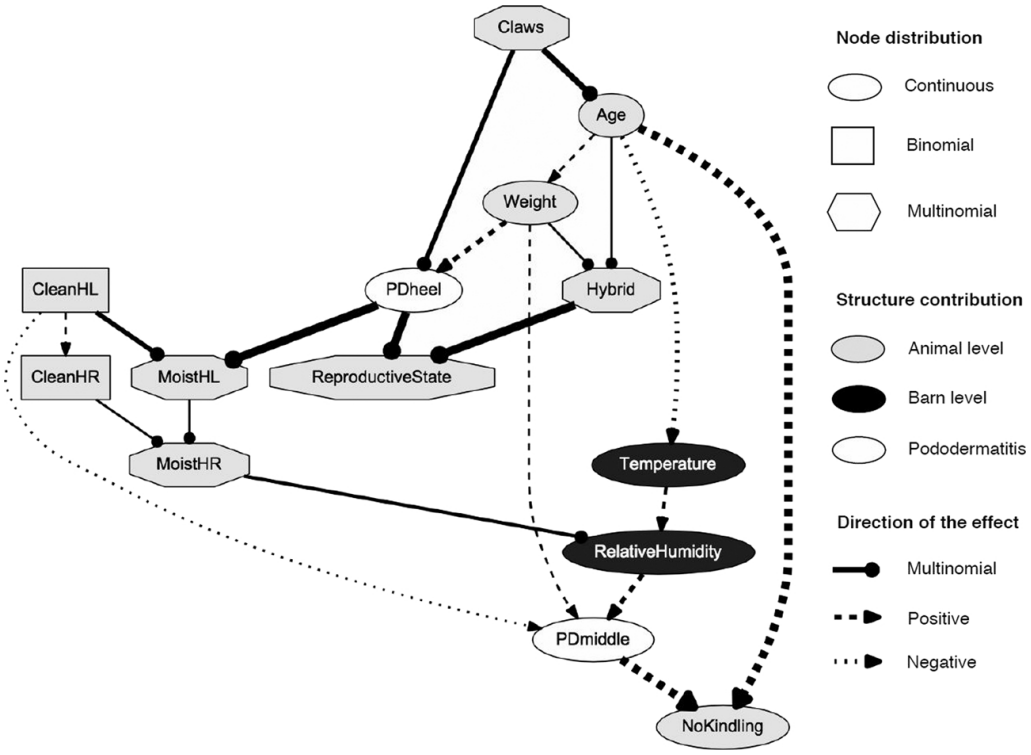

Figure 1.2: Directed acyclic graph (dag) from `abn` taken from Ruchti *et al.* (2018).

1.1 Statistical considerations

As mentioned, given that data is available across time points, it is possible that out of these four unique areas (barn 1, barn 2, barn 3 and barn 4), meaningful cohort affects can be found with respect to mean score and mean score trajectory over time. Several `R` packages to clustering longitudinal are available including `longclust`, a method based on a mixture of multivariate t or Gaussian distributions, model-based clustering in `mclust`, classification of trajectories based on selected factors in `traj` and k-means design to cluster `kml`. Amongst these approaches, this study will use the k-means approach.

1.2 K-means longitudinal cluster analysis

The idea behind k-means clustering is to use a centroid based clustering methods to spherically include data points around a single mean representing k units or observations where k is the number of optimal clusters. K-means clustering was implemented to determine the optimal number of k clusters or partitions using an available `R` package called `kml` of version 2.4.1. The `kml` approach is a hill-climbing alogrithm which always verges towards an optimum.

The features of `kml` include (Genolini and Falissard, 2010) :

- observations are not required to be based on a parametric distribution. This could be advantages if no prior information is available,
- since it requires iterations to optimise the means per cluster, it is likely to be robust with regards to numerical convergence,
- no assumption on the shape of the trajectory is assumed as partitioning the centroid is exploratory until the centroid is optimised,
- the formation of k-means clusters in a longitudinal context is independant from time scaling.

1.3 Mixed model approaches to optimally partitioned clusters in longitudinal setting

In classical regression analysis, observations are assumed to be independant from one another. Since subjects of these observations occur in a several time points in a longitudinal setting, an adjustment of dependant observations need to be considered. One such solution is to treat their baseline values as different between subjects, such that they are random and their intercepts differ (Twisk, 2013). This is in contrast to fixed effects where the variance is shared across all subjects (Twisk, 2013).

In a longitudinal setting, a mixed model can study the cohort and age (or time) effects and thus is an effective means of studying change (Diggle *et al.*, 2013). The setting of this longitudinal study requires optimal partitions to assess the effect of cohort and these partitions will be created by `kml` implementation. Thereafter, the time component can be assessed with other important covariates in a mixed model analysis. Since the same individuals are evaluated over time (but not for all time points), the assumption that individual observations are independant to one another will not hold. Thus each individual will have a random intercept as they share the same baseline values. For that same reason, age and weight variables were grouped per individual and was treated as a random variable.

1.4 Data Collection

The following information about how data was collected is cited from the precedent study (Ruchti *et al.*, 2019).

Data was collected via several visits between July 20th, 2016 and June 30th, 2017 on three commercial Swiss rabbit farms with group housing of breeding does (www.schweizerkaninchen.ch). These farms had animal friendly housing label BTS (https://www.kontrolldienst-sts-ch/html/index.php/de/coop-nts-kanichen). Visits were made once every four weeks for a period of two days, and thirteen visits were made. In each farm, 67 does were initially caught in a stratified manner. At least one doe per pen was chosen, bar the rabbits which appear in moribound state which were subsequently reported to the farmer. Scoring is performed on the middle ("mid-paw") and mean heel palmar or plantar surface by manual palpatation (without gloves) by one person and the doe was immediately return to its respective pen after evaluation. A headlamp was used to control for consistent lighting. Environmental factors were scored by a randomisation of pen number per farm.

1.5 Data Preparation

One ear tag was found to have no input values for the concerned scores. This row was removed from the dataset. There were six ear tags that were the same in barn 1 and barn 2 which belong to separate farm locations, and this was renamed such that each ear tag in the study was unique and its individual trajectory can be clarified. There were six different categories for claw length, for simplicity, this was recorded by the predecessor as "normal" or "too long". Months were refactored in ordered visits 1 to 13. All other ordinal variables were refactored such as reproductive state. The F1 hybrid was replaced with its alternative name, Hylamax. The score of pododermatitis were from zero to ten, and rescaled from one to six, thus often scores are represented with two decimal places and three significant figures. Consistent with the predecessor, I created a new variable called "meanPDheel" as a mean value of the left and right disease scores of each of heel and mid-paw. The choice of heel or mid-paw scores as dependant variable was based on the higher number of complete cases, found to be in the mean heel score data.

Chapter 2

Materials and Methods

2.1 Description of variables

To perform the analysis of risk factor on pododermatitis, nine variables were evaluated where four were continuous and the other five had their respective factor levels. These included claw length, relative humidity, temperature, age, weight, cleanliness, moisture and barn areas. Environmental factors affect mid-paw scores and not mean heel scores as seen in the directed acyclic graph (Ruchti *et al.*, 2019) however they were nontheless included to measure a possible effect.

2.1.1 Scoring, moisture of paws and claw length

Pododermatitis is localized in the heel and mid-paw plantar region of hind paws which are scored on a visual analogue scale (Drescher and Schlender-Bobbis, 1996) as seen in Figure 2.1 and

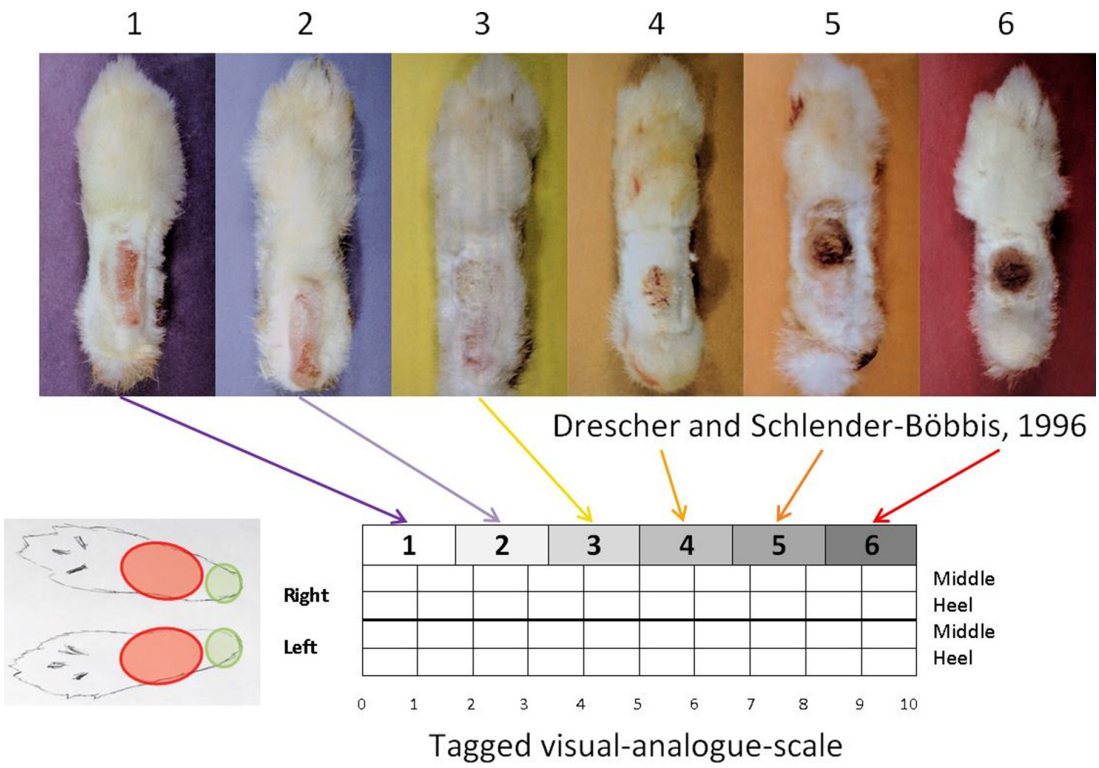

Figure 2.1: Visual analogue scale of pododermatitis taken from Ruchti *et al.* (2019). The disease were originally scored between 1 to 10 as seen at the inferior side of the above table. The location of scoring is shown in the bottom-left corner.

Description of the 6 stages from Drescher and Schlender-Böbbis (1996) used for the pododermatitis-scoring of the hind legs.

Severity score	Description
0	Healthy paw with normal amount of fur
1	Reddened skin, hypotrichosis or alopecia
2	Low-grade hyperkeratosis, hypotrichosis or alopecia
3	Hyperkeratosis, alopecia, scaling
4	Hyperkeratosis, alopecia, scabs from clear wound secretion, beginning ulceration
5	Hyperkeratosis, alopecia, scabs from bloody wound secretion, ulceration
6	Hyperkeratosis, alopecia, crusts from bloody wound secretion, deep ulceration, degeneration of the surrounding tissue

Figure 2.2: Descriptives of pododermatitis score by visual analogue scale in Figure 2.1 by Drescher and Schlender-Bobbis (1996) taken from Ruchti *et al.* (2019).

described by Figure 2.2. Severe pain is understood to be experienced at score 4 and progressively toward score 6 (Ruchti *et al.*, 2019). The paws' moisture is assessed with three ordinal levels: dry, moist and wet on the front and hind limbs bilaterally. Claw length was measured bilaterally on the hind paws only and categorised as "normal" or "too long".

2.1.2 Temperature, relative humidity, age, weight and hybrid

Temperature was recorded in degrees celcius and relative humidity in percentages. The does were of three different breeds; Hycole, Hyla and the interbreed to both, Hylamax. The distribution is seen in Table A.1 and A.3 in the Appendix section. Most does from barn 1 belong to the Hyla breed. Hycole is highly represented in barn 4 and almost evenly distributed in barns 2 and 3 which are exclusively in farm 13. Age is recorded in months and weight in kilograms (kg) to the nearest second decimal place.

2.2 Statistical analysis

Data was read from an excel file provided by the authors of the precedent study, by using `readxl` (Wickham and Bryan, 2019). Statistical analyses were performed by R version 3.5.2 (2018-12-20) (R Core Team, 2018). The data transformation to wide form was done by functions by `tidyverse` (Wickham, 2017) and `base R`. Missing values were replaced by means through function `imputation` by package `longitudinalData` (**?**). Visualisations were performed by package `ggplot2` (Wickham, 2016).

A mosaic plot is a graphical summary of the conditional distributions in a contingency table and graphically plots two or more qualitative variables. Mosaic plots were created with the package `ggmosaic` (Jeppson *et al.*, 2018).

An exploratory data analysis was performed on risk factors guided by the precedent study (Ruchti *et al.*, 2018) and their mean, median and standard deviation were stratified by area and by visit. Trends of continuous variables were graphed with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). A linear mixed model was performed using the `lmer` (Bates *et al.* (2015) and Kuznetsova *et al.* (2017)) and the covariates included were risk factors indicated by the precedent study. This model treated individual rabbit (variable ear tag) and mid foot scores as random variables. The mixed model analysis also included a random intercept on age per ear tag. This mixed model was applied to the unclustered population followed by the two and four clusters created by `kml`.

After `kml` partitioning was performed, generalised logistic regression was compared of the two and four parition case to understand if these partitions are different with respect to variables associated with pododermatitis scores.

2.3 Initialisation of kml implementation

The following order of events were required prior to kml implementation:

1. Creating a wide form data frame with each time point as a column.
2. Maintaining only unique identification of does with the column as time points. This is called a `traj` object. Ensuring this is of class `matrix`. Missing values are created as not all does are evaluated at every time point in the study.
3. Imputation using the `trajMean` option, where missing values are replaced by mean of an individual does' trajectory.
4. Inputting (2) into `cld` or `clusteringLongdata` command to create a `cld` object.
5. The `cld` objet is ready for `kml` implementation.

2.4 Implementation of kml

The following order of events were required for kml implementation:

1. The `cld` object was used in the `kml` implementation using `fastkml` mode, a `C`-programmed computation. The option "nbCluster" was given values of 2 and 4 to obtain optimal 2 and 4 partitions. No other options were chosen.
2. Ensuring the wide form data frame is of class `data frame`, we create a column where the `kml` implementation will allocate a cluster category (e.g. A, B, C, D for four partitians) per individual doe or (per row).
3. `getCluster` command was implemented to (1) and provided the input for the afore mentioned column created. At this point, `kml` partitioning is complete.

The `kml` implementation required creation of `clusterLongdata` objects, which involved a wide-form class `matrix` input with time as column variables. As the individual does' trajectories are not all of time points 1 to 13, creating wide-form inputs revealed missing values which were imputated using the `trajmean` from the `kml` package (Genolini and Falissard, 2010). In theory, `trajmean` replaces the missing x_{ij} observation by the average of the values of that individual's trajectory, for example, by the mean of $\boldsymbol{x}_{.j}$. The observation x being the score per ith time point of individual j.

The number of clusters were determined to be two and four and no random starting value was initialised on the mean scores of rabbits' left and right heels. In the `kml`, I chose two and four optimal partitions using the `nbClusters` option. I specified the default number of redrawings to find optimal partitions, and this was 20 as set by the package itself, (Genolini and Falissard, 2010). The Euclidean distance between individuals was default and used in this implementation.

The partitioned or clustered data set by `kml` implementation shall be referred to as either two and four partitians.

This study used two implementations of `kml`, however once the trajectories were visualised in the slow setting, the `fastkml` was used to generate k-means partitions. The `slowkml` is programmed by R and graphically displays the partioning process. The `fastkml` implementation is optimised in C which is approximately 25 times faster than `slowkml` (Genolini and Falissard, 2010). After the number of centroids per time point were randomly assigned by the implementation. The Euclidian distance were maximized between these points. This occured for each time point simultaneously. The `kml` chooses the partitian by maximising the determinant of the matrixes between time points. The trajectory is formed when the algorithm has reached maximum iterations of 20 which is a setting within the `kml` package (Genolini and Falissard, 2010).

Clusters when implemented are temporarily saved in the cluster object, where clusters can be obtained at the completion of this partitioning, even if halted midway. From these clusters, I described the mean, standard deviation and sample size per cluster. I compared these two and four partitians in a concordance matrix to each barn area as well.

To improve the understanding of `kml` partitions, visual inspections were performed on risk factors identified by the predecent study; age, weight, area, claw length (categorical), moisture (categorical), cleanliness (categorical), reproductive state (categorical), relative humidity and temperature.

2.5 Robustness of `kml`

To ensure that the `kml` used for the two and four partitians were robust, that is, it achieves a consistent numerical convergence, I randomly created systematic random deletion of data and compared the patterns of trajectory to each magnitude of random deletion and to the scenario where no loss of data were observed.

Random deletions of magnitude 1%, 5%, 10% and 20% were implemented at two different stages prior to visual inspections and analysis.

When the random deletions were implemented on the original data set, `kml` algorithm was performed thereafter. This analysis is referred to as as “pre imputation”. When the random deletions were implemented on the original data set, the `kml` algorithm was performed thereafter. This analysis is referred to as as “post imputation”.

For both cases, this meant that some trajectories were removed and or entire individual rabbits were removed, as such entire trajectories were no longer present.

Chapter 3

Results

This chapter will include results of exploratory data analysis (EDA) of data prior and after `kml` implementation. Following this, it will include the performance results of the partitions for pre and post imputation robusticity checks.

3.1 EDA and pre-`kml` results

The total number of observations were 2612 rabbits in the study population where each unique farm 11, 13 and 17 respectively contributed to evaluations of 200 does for visit 1 and 201 for each of visits 2 to 13 inclusively. There were 343 does in the entire study population and they are not each evaluated at each time point, and they "appear" at different time points in the study.

Table 3.1: Sample size in four distinct areas (barns) over thirteen visits totalling to 2612. Each row represents counts of rabbits evaluated per visit for a specific barn, totalling to 871 for barns 1 and 4 each. Barns 2 and 3 belong to the same farm (Farm 13).

Visit	1	2	3	4	5	6	7	8	9	10	11	12	13	Total
Barn 1	67	67	67	67	67	67	67	67	67	67	67	67	67	871
Barn 2	34	34	34	34	34	34	34	34	34	34	34	34	34	442
Barn 3	32	33	33	33	33	33	33	33	33	33	33	33	33	428
Barn 4	67	67	67	67	67	67	67	67	67	67	67	67	67	871
Total	200	201	201	201	201	201	201	201	201	201	201	201	201	2612

3.1.1 Scoring of Pododermatitis

The mean of the heel of left and right foot was used instead of median, and heel scores were preferred over mid foot scores due to higher counts of complete cases (2600 and 2603). Furthermore, the correlation is 0.14 via Kendall's rank correlation (see Appendix in Figure A.1) between heel and mid paw and trends show that as mean heel score increase, mean mid-paw score increases in general but not in a strictly linear manner as seen also in Figure 3.1. There is a significant evidence of association between mid-paw and mean heel scores, as seen in the linear model output Figure A.5.

Mean heel scores across 13 visits is 3.9 with standard deviation 0.33. Mean scores stratified by area show that some mean heel scores improve over time and have overlap of the standard error, in Figure 3.2 These scores also show peaks within the winter months for barns 1, 3 and 4. When looking merely at the summer to autumn months, scores tend to decrease for all barns between visit 1 to 4, and for barns 1 and 3, scores increase from visit 4 reach a peak between

Figure 3.1: Trend of mean mid-paw and heel score with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016).

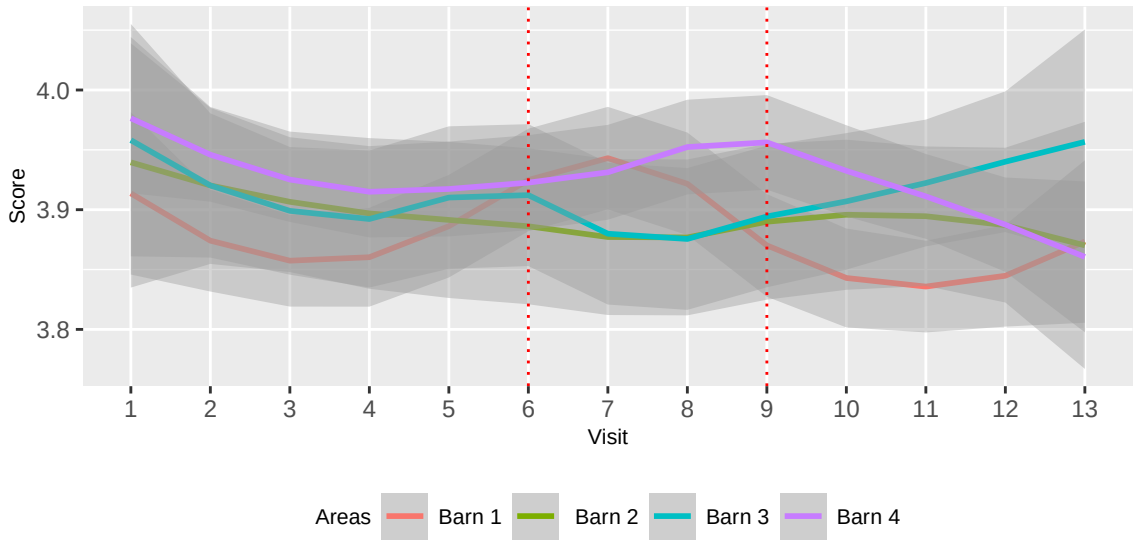

Figure 3.2: Mean bilateral heel scores across 13 visits for all areas with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

visit 6 and 9 and the scores in these barns decrease during winter. When visit 1 is compared to the visit 13, there is an overall improvement of pododermatitis scores across most barns.

3.1.2 Reproductive state, age and weight

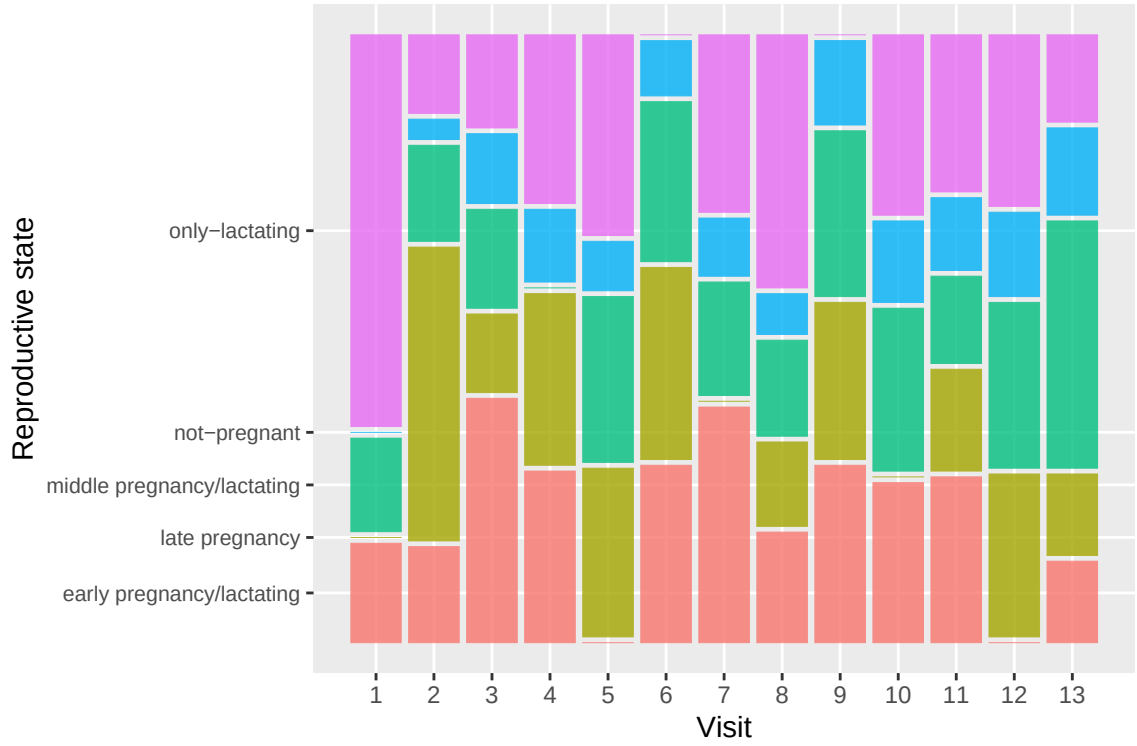

Figure 3.3: Reproductive stats across all areas or barns. On the x-axis are each visit where visit 6 to 9 inclusive represents the winter months. The width of each rectangle represents the proportion within each barn. On the x-axis, the length of each rectangle represents the proportion in each reproductive state.

I note that most rabbits are lactating or pregnant across visits in Figure 3.3 and across barns Figure 3.4. Does were between three to 43 months of age, with a mean of 16.56 months. The median age was 15 months. The does had a mean weight of 5.12 kg, with minimum and maximum of 2.73 kg and 9.09 kg respectively. The median weight was 5.11 kg. The parameters stratified by barns are seen in Table 3.2. When age was compared across time, there is a non-linear trend with visits. This is likely due to new entries of does into the study, thus the age is not consistent across all time points. Barn 3's does seem to have a higher range of age in mean, followed by barn 1, barn 2 and barn 4, especially at the first six months of the study as seen in Figure 3.5. When weight is compared across visits, barn 2 seems to have the highest weight in mean across time. There is also a considerable peak reached during the winter months across all areas for weight. The standard error bounds approximately represent 100 g to 300 g across all areas, upon visual inspection of Figure 3.6.

Table 3.2: Stratified mean and standard deviation of age (in weeks) and weight (in kg) including population sample per barn area. The stratification with youngest mean age is barn 4. The stratification with the highest weight is barn 3.

Area	n	Age: mean (SD)	Weight: mean (SD)
Barn 1	871	17.6 (8.2)	5.15 (0.55)
Barn 2	442	16.6 (6.8)	5.15 (0.54)
Barn 3	442	18.3 (8.1)	5.29 (0.52)
Barn 4	871	14.6 (6.4)	5.15 (0.53)

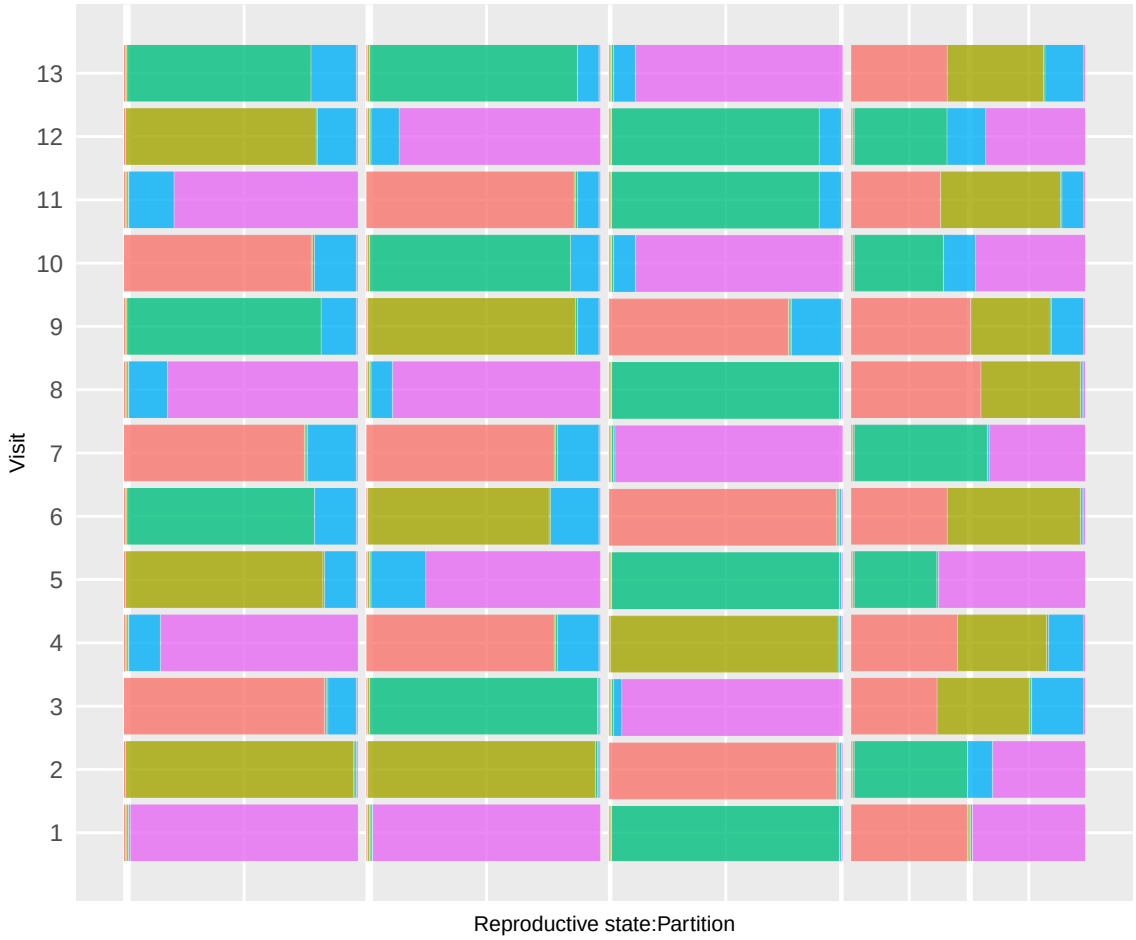

Figure 3.4: Reproductive stats across visits on y-axis and area from barn 1 to 4 from left to right on the x-axis. On the x-axis, the length of each coloured rectangle represents the proportion within each respective barn. On the y-axis, the height of each rectangle represents the proportion in each visit. The categories for reproductive state and their colours in brackets are as follows : Late pregnancy (khaki green), middly pregnancy (aqua green), not pregnant (blue), early pregnancy (salmon), only lactating (fuschia), not pregnant (clear).

3.1.3 Temperature and relative humidity

The temperature and relative humidity stratified across areas is visualised in Figure 3.7 and Figure 3.8. The relative humidity of barn 4 has an inverse trends to all other barns and this area was reported to have the highest altitude Ruchti *et al.* (2019). The peaks of the lowest and highest peak of relative humidity during the winter months has an approximate difference of 20%. All other temperature and relative humidity follow similar trends amongst other areas.

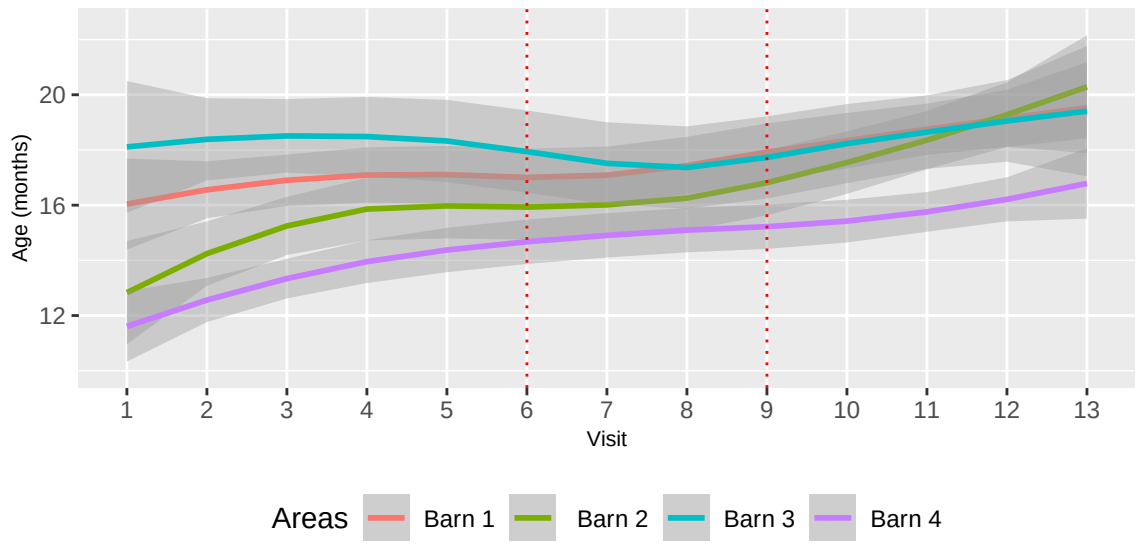

Figure 3.5: Age (in months) trends across visits stratified by area with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

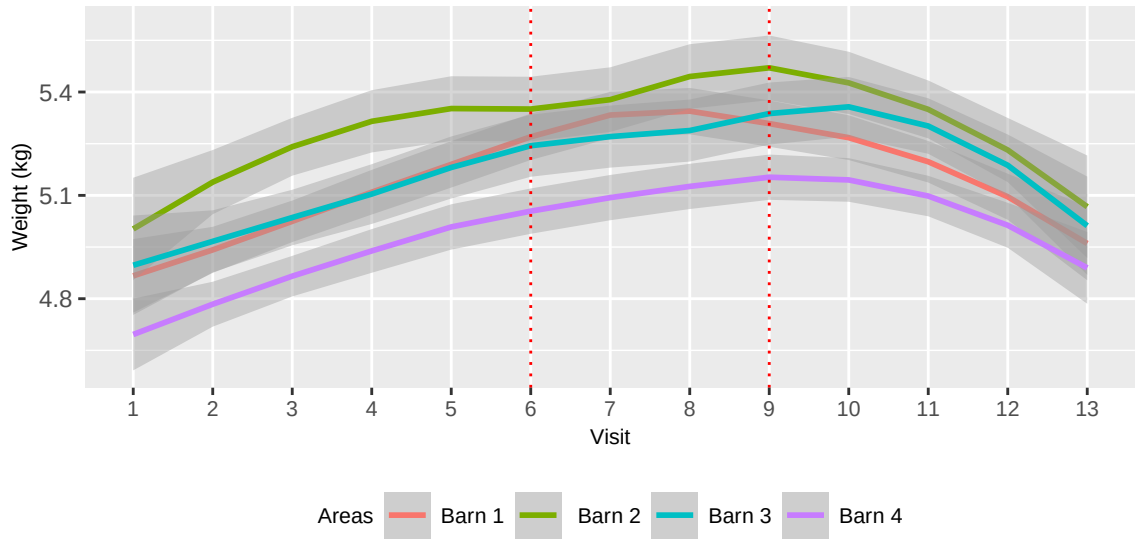

Figure 3.6: Weight (kg) trends across visits stratified by areas with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

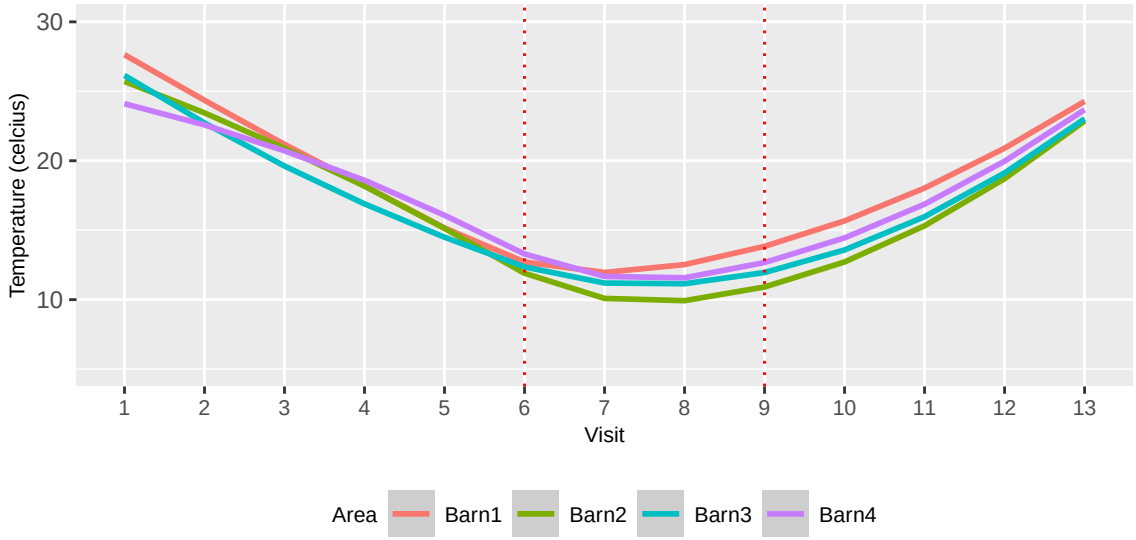

Figure 3.7: Temperature trends across visits stratified by areas with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

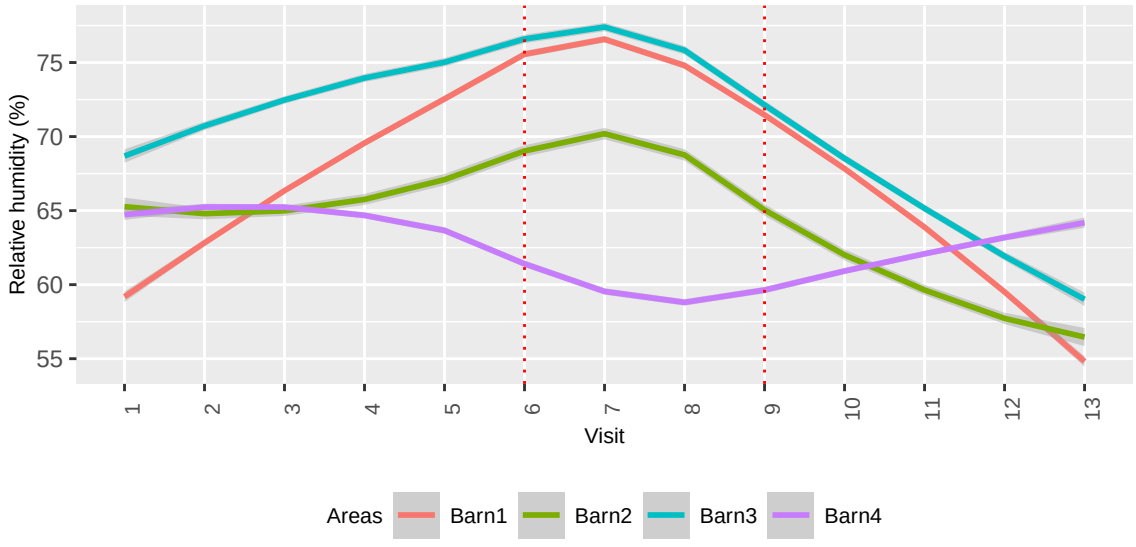

Figure 3.8: Relative humidity trends across visits stratified by areas with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

3.1.4 Cleanliness and moisture of paws

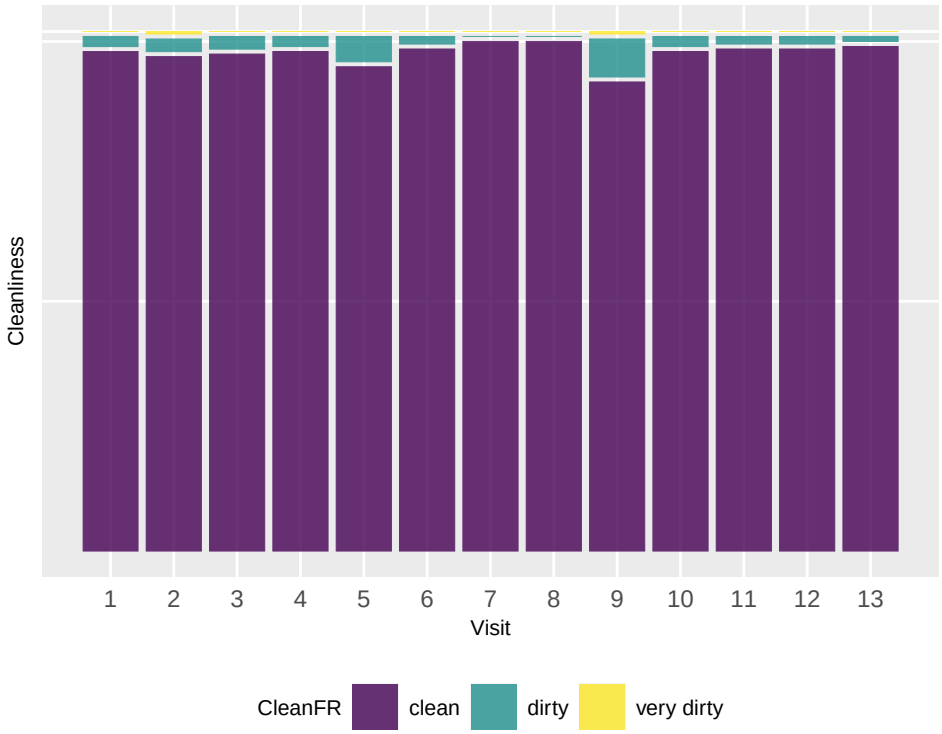

Figure 3.9: Cleanliness of rabbit front paws (FR = Front Paw) across visits. The x-axis represents visits from 1 to 13. On the x-axis, each width is the proportion of observations. On the y-axis are the categories of cleanlinessindicated by the legend. On the y-axis, each of its height represents the proportion of that category of each visit.

Most paws when assessed were clean in the right front paws as seen in Figure 3.9. Most rabbits across visits had dry paws as well, and for hind left paws, there is a winter trend in barn 1 where paws are mostly moist in visit 7 and 8. Specific winter trend on other areas on the moisture of the paws are not apparent. Some hind right paws were wet throughout the year but in small proportions per barn and visit as seen in Figure 3.10.

3.1.5 Claw length

With respect of claw length, there is a higher proportion of normal length of nails than "too long" across all areas and within stratified areas, especially during the winter months.

3.2 Initialisation and implementation of `kml`

During the initialisation process, the long form data frame where each row was a single observation, was converted to a 343 by 13 wide-form data frame as there were 343 individual does across 13 visits. The wide form data frame now had each row represent the trajectory of each individual doe. As not all individual does are present at each of the time points, 1872 values were missing and imputated prior to `kml` implementation. Sample sizes per partition decreases from A to D for the `kml` implementation for four partitions, whilst sample sizes are similar for both groups A and B in the `kml` implementation for two partitions as seen in Table 3.3. Sample sizes decreases progressively with increase in partitions. The graphical illustration of the two partition process is seen in Figure 3.12, provided by the slow `kml` command. The left side of this illustration is the

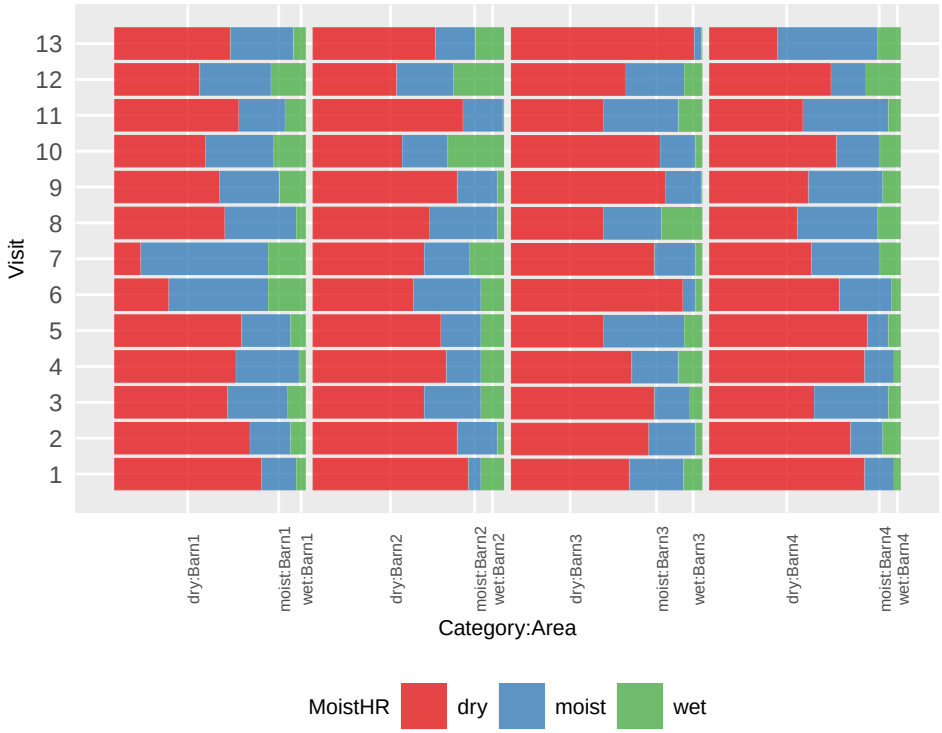

Figure 3.10: MoistHR represents the ordinal variable of moisture of hind right paw of the doe. The y-axis represents each visit in descending order from top to bottom. On the y-axis, the height of each rectangle represents the proportion of does in that specific visit or time point. On the x-axis from left to right, represents barns 1 to 4. The width of the x-axis represent the proportion of a particular ordinal category.

Table 3.3: Mean and standard deviation score for two and four partitions including sample sizes. Since `kml` partitioning required full trajectories, each implementation required thirteen time points per individual (unique) doe or ear tag. The number of observations thus increased from 2612 of the original dataset.

Partitions	n	Mean (SD)
A of 4 partitions	2951	4.0 (0.2)
B of 4 partitions	1183	3.7 (0.4)
C of 4 partitions	312	4.0 (0.2)
D of 4 partitions	26	3.8 (0.3)
A of 2 partitions	3341	3.5 (0.4)
B of 2 partitions	1131	5.0 (0.0)

quality criterion indicating the best number of partitions according to the algorithm. Among all the partitions, the one indicated with a black dot is displayed on the right with “2” indicated

Table 3.4: Concordance of two and four partition scenarios

	A	B	C	D
A	0.88	0.11	0.00	0.01
B	0.00	0.72	0.28	0.00

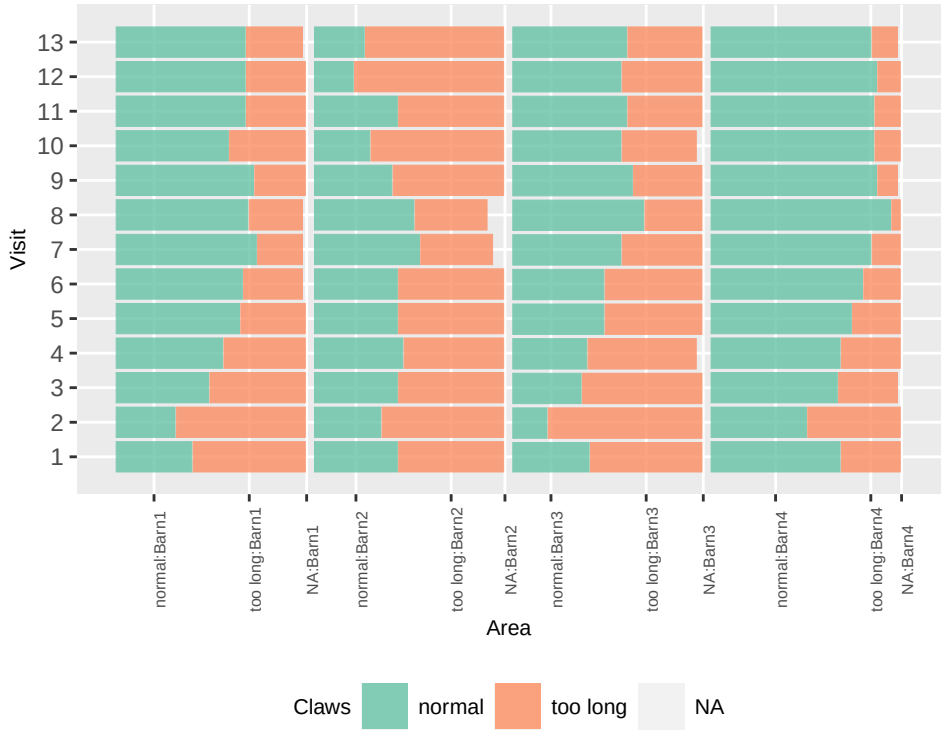

Figure 3.11: Claw length categories across four areas for each visit from barn 1 to 4 from left to right. The y-axis represents each visit in descending order from top to bottom. On the y-axis, the height of each rectangle represent the proportion of does in that specific visit or time point. On the x-axis, each length represents the proportion of a particular category of each respective barn.

above and "4" as a curve inferior to it. The partitioned data after `kml` implemention in two and four partitions were compared in Table 3.4. The paritions A and B proposed by two partition is most similar to partition A and B of four partitions.

3.2.1 Generalized linear models

Generalized linear models of binomial distribution assumption was used and showed that there is evidence of significant difference between the partitions within each `kml` implementation. For more detailed output and for the four `kml` partition summary output, refer to Appendix section on Figure A.6 and Figure A.11.

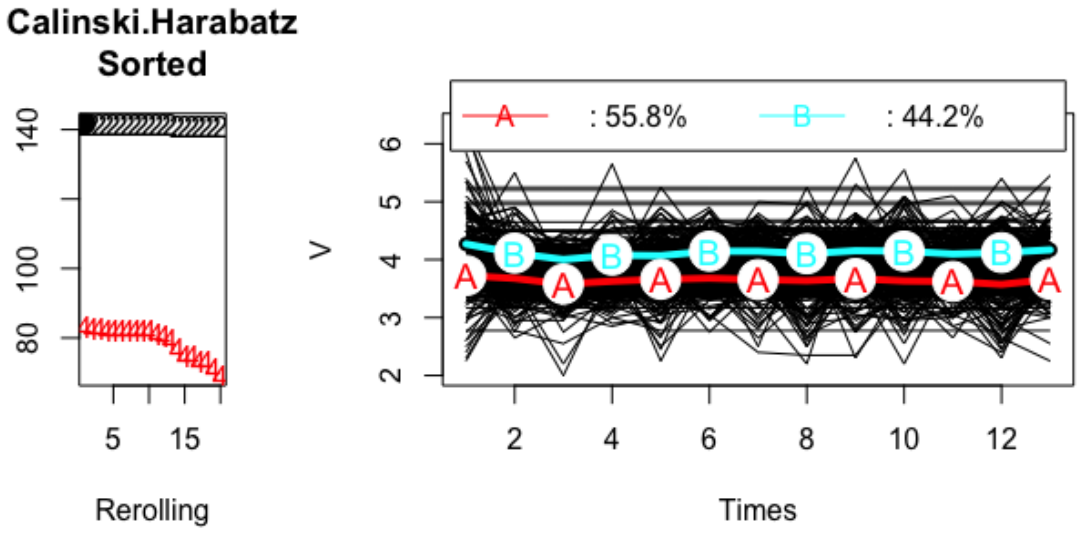

Figure 3.12: Graphical display of the four partioning process with quality criterion on the left, distribution of trajectories on top and trajectories in process of partitioning on the bottom right. The left side of the figure is the score of quality criterion. Where the dot is placed indicates the number of clusters (seen here "2") that is shown on the right side of this figure. The lower curve in read indicates the four parition scenario where seen closely is reresented by the character "4" on the curve.

.

3.3 EDA of `kml` partitions

3.3.1 Temperature and relative humidity

The trajectories of temperature and relative humidity follow a similar trend in the two and four partitions. The effect of barn 4's relative humidity is not apparent after partitioning as observed in Figure 3.14.

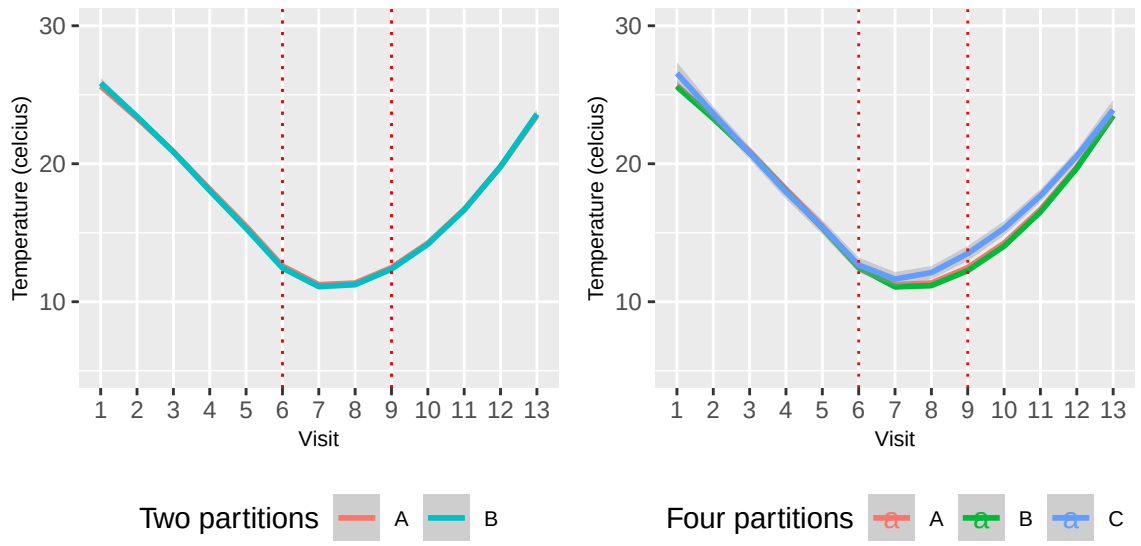

Figure 3.13: Temperature across visit for two partitions in a with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

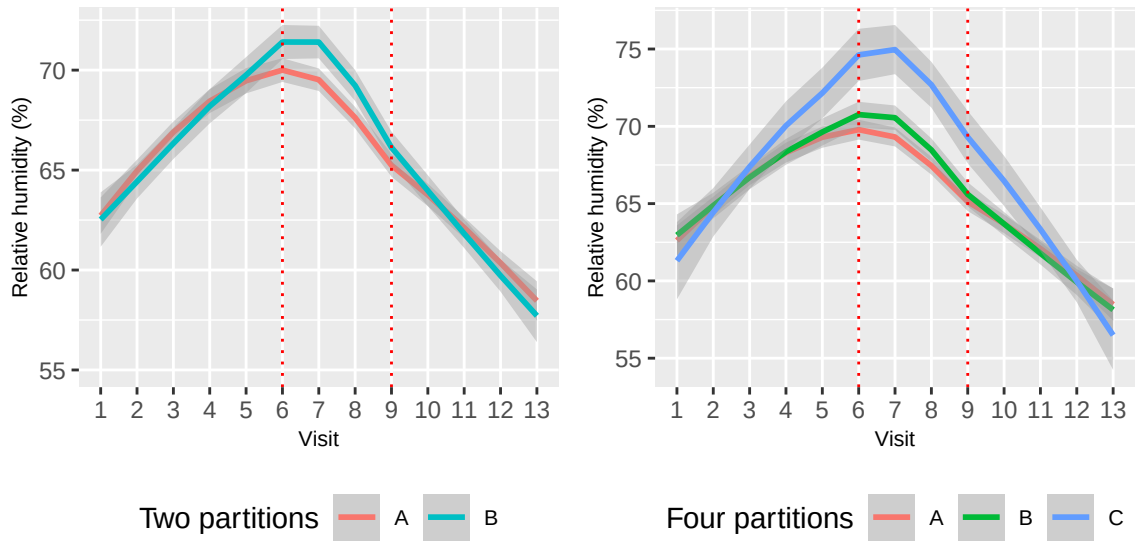

Figure 3.14: Relative humidity of two partitions across visits with `loess` smoothing with banding of its time-point standard error, based on t-approximatio (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

3.3.2 Scores of Pododermatitis

For the two partition implementation, the mean heel scores across all visits had a mean of 3.9 and a standard deviation of 0.33. For the four partition implementation, the mean heel scores overall had a mean of 3.91 and a standard deviation of 0.32. There is a clear separation between the two partitions in Figure 3.15, below and above score four. For the four partitions, there is mostly a clear separation between trajectories of A to C as seen in the Figure 3.16. There is mostly no peak during the winters for the two and four partition case, except for partition C in the latter case. Scores in all partitions tend to be higher in the winter months.

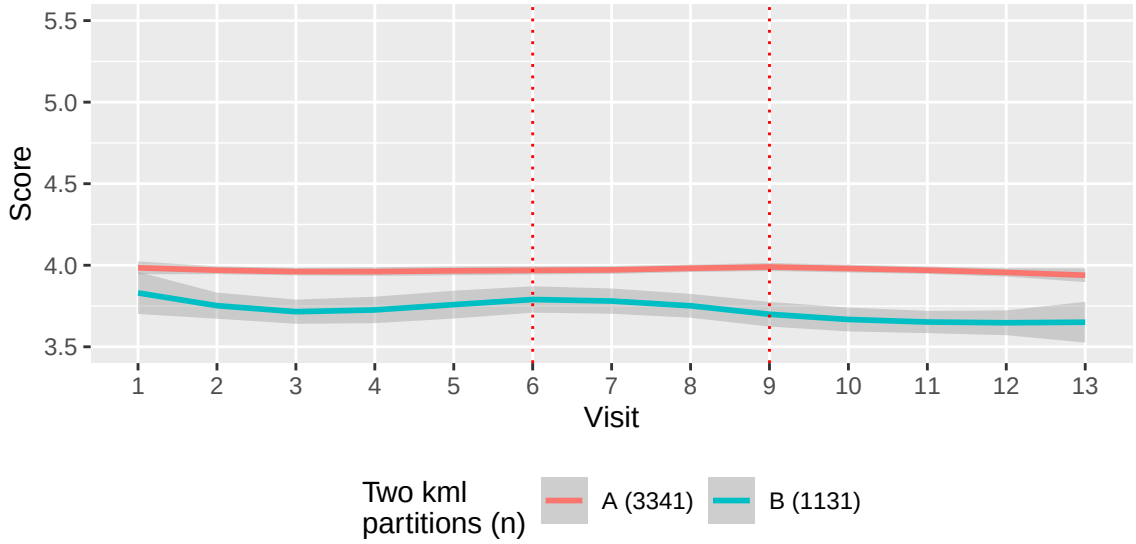

Figure 3.15: Mean bilateral heel scores across 13 visits for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

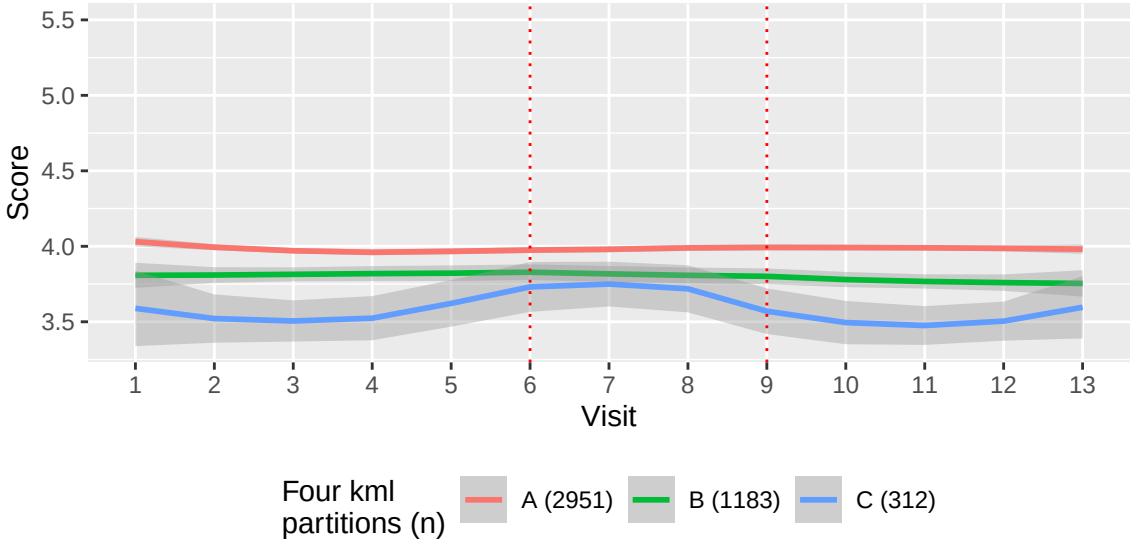

Figure 3.16: Mean bilateral heel scores across 13 visits for four partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation a (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season. Partition D has a small sample count and thus is not represented here.

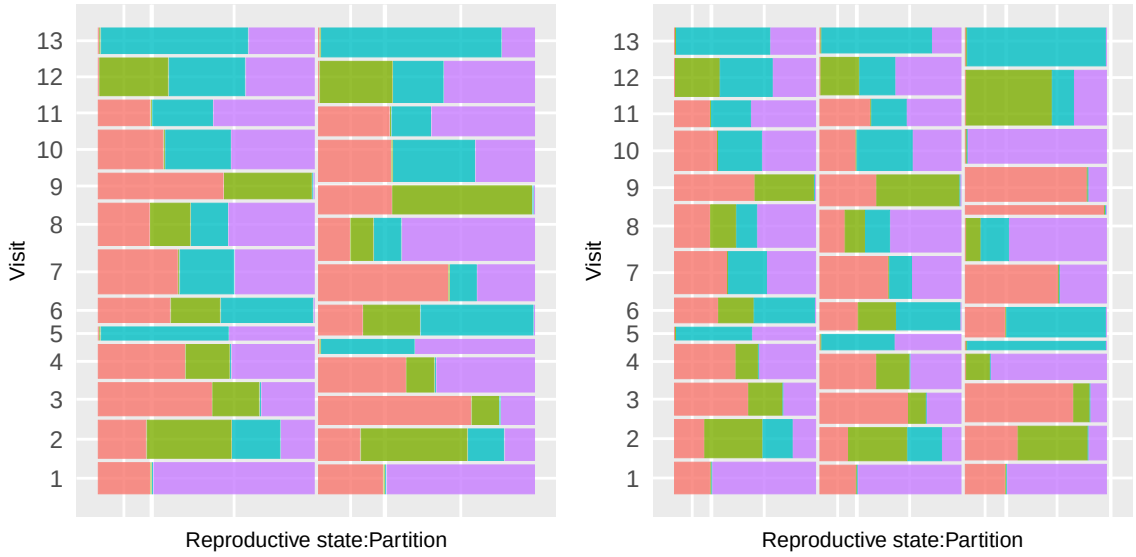

Figure 3.17: Reproductive states across two partitions and four partitions from A on the origin axis. On the x-axis are respectively partition A and B (from left to right) for the two partition case on the left mosaic plot, and partitions A to C (from left to right) on the right mosaic plot. On the y-axis of each mosaic plot, the height of each rectangle represents the proportion in each reproductive state. On the x-axis on each mosaic plot, each length represents the proportion per category. The categories for reproductive state and their colours in brackets are as follows : Late pregnancy (khaki green), middly pregnancy (aqua blue), early pregnancy (salmon), only lactating (fuschia). In the four parition data set, partition D has a small sample count and thus is not represented here.

3.3.3 Reproductive state, age and weight

I note that most rabbits are lactating or pregnant in any given time and farm areas in the partitions, seen in Figure 3.17. This is consistent with results prior to `kml` implementation, for example in Figure 3.4 in the Appendix section.

In both the two and four partitions, age does not increase in a strict linear manner with time. The trends are similar to the area by stratification analysis. The standard errors increase for partition C in the four partition case.

Weight increases for the two and four partitions with time. There is a visible peak on visual inspection during the winter months. For two partitions, partitian B is superior in weight than A however given the standard error shaded, there is an overlap. The standard error across time is relatively stable in across all visits as observed in Figure 3.18. For the four parition case, weights increase with time. The standard error for C is larger and overlaps the other trajectories for most of the time points.

3.3.4 Cleanliness of paws

Most claws at all time points are categorised as clean as seen in Figure 3.20 for both two and four partitions. The other appendages are recorded in the Appendix in Figure A.2 and Figure A.3.

3.3.5 Claw length

When comparing the effect of claw length across optimal partitions in the four partitions, frequency of normal length claws increase over time whilst “too long” claws decreased at the end of

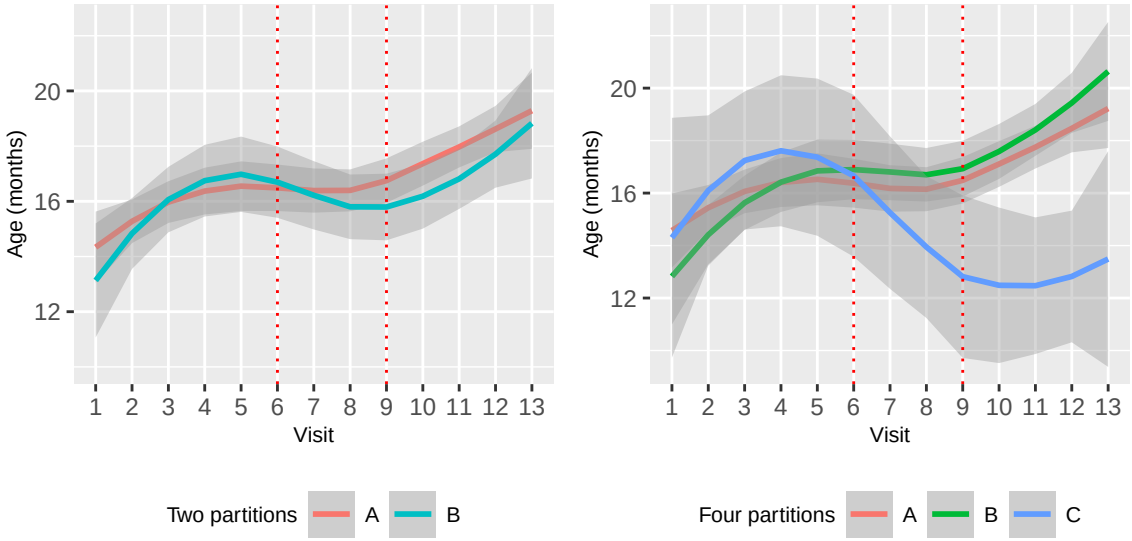

Figure 3.18: Mean age of two and four partitians across visits with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

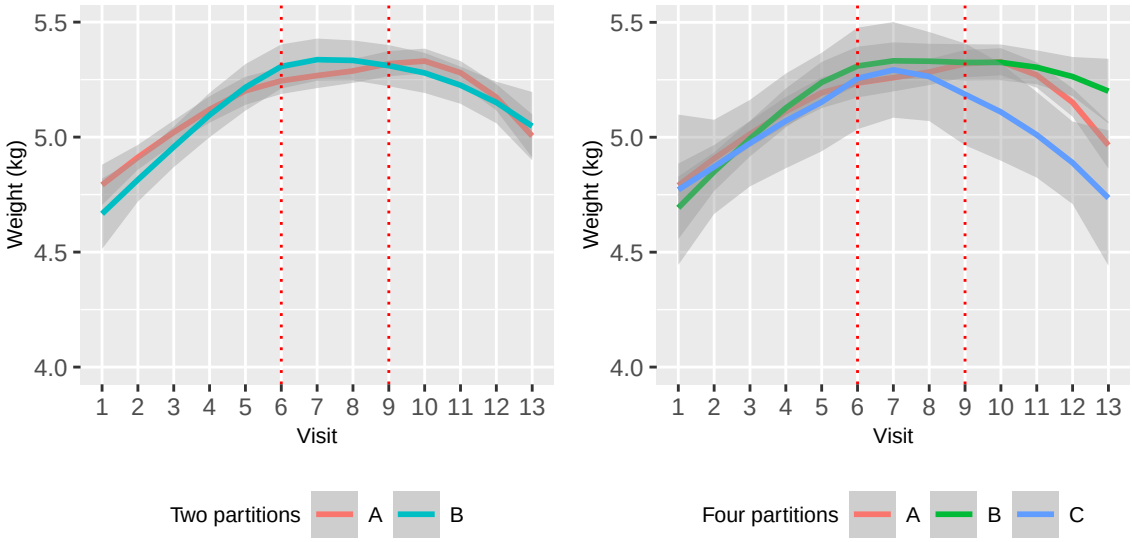

Figure 3.19: Mean weight of two and four partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season. In the four partition case, partition D has small sample counts and thus is not shown here.

the study as seen in (Figure 3.21).

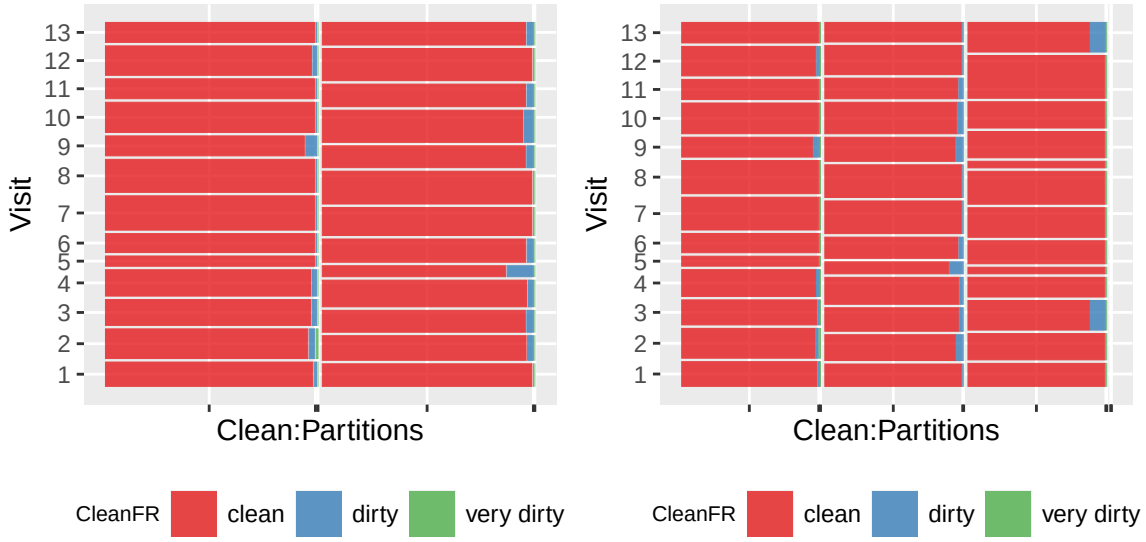

Figure 3.20: CleanFR represents the ordinal variable of cleanliness of front right paw of the doe. On the x-axis are respectively partition A and B (from left to right) for the two partition case on the left mosaic plot, and partitions A to C (from left to right) on the right mosaic plot, where the width of each rectangle represents the proportion within each barn. On the y-axis, the height of each rectangle represents the proportion in each cleanliness state. In the four parition data set, partition D has a small sample count and thus is not represented here.

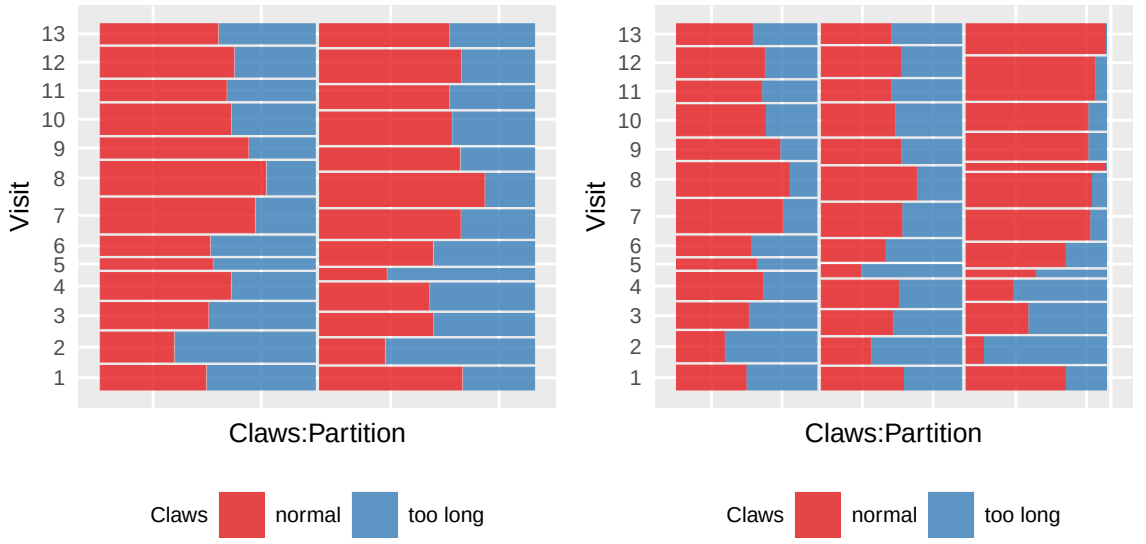

Figure 3.21: Claw length across two partitions and four partitions from A on the origin axis. On the x-axis are respectively partition A and B (from left to right) for the two partition case on the left mosaic plot, and partitions A to C (from left to right) on the right mosaic plot, where the width of each rectangle represents the proportion within each barn. On the y-axis, the height of each rectangle represents the proportion in each category. In the four parition data set, partition D has a small sample count and thus is not represented here.

3.4 Mixed models of non-partitions and partitions

Table 3.6: With two and four partitions, the mixed model estimates of variables with evidence of significant effect with random effects of area, ear tag and age per ear tag, weight per ear tag.

Two partition	Estimate	Std.Error	CI	*pvalue*
visit	-0.005	0.002	-0.001,-0.009	0.019

Four partition	Estimate	Std.Error	CI	*pvalue*
visit	-0.005	0.002	-0.001,-0.009	0.025

The mixed model results overall showed that the variable visit had evidence of significance effect to mean scores when the two and four partitions were included in the model, seen in Table 3.6.

3.5 Robustness of `kml` via two experiments

Pre imputation sample sizes are expected to be equal across random deletions of original data. This is because deletion occured prior to `kml` implementation was deletion of random observations. Since each row represented results of visits per ear tag, those missing scores per visits were later replaced by imputation. Unequal sample sizes as in for 20% random deletion of data is due that when all visits within the same ear tag were removed and cannot be imputated due to an entire loss of the same ear tag. The post imputation sample sizes are expected to be lower than the pre imputation as observations were deleted from already imputated data.

Table 3.8: Pre imputation sample sizes per cluster which is expected to be equal to imputated data set without deletion. The deletion for pre imputation is done prior to imputation and also prior to `kml` implementation for four partitions. Since `kml` partitioning required full trajectories, each implementation required thirteen time points per individual (unique) doe or ear tag.

% loss	A	B	C	D	Total
0 %	3042	1092	312	26	4472
1 %	2795	1183	468	26	4472
5 %	2860	1261	325	26	4472
10 %	2782	1144	494	39	4459
20 %	2873	1183	299	39	4394

Table 3.10: Post imputation sample size per cluster which is expected to be lower than data with imputation as deletion was performed after imputation and `kml` implementation for four partitions.

% loss	A	B	C	D	Total
0 %	2951	1183	312	26	4472
1 %	2920	1174	308	25	4427
5 %	2809	1121	295	23	4248
10 %	2668	1056	277	24	4025
20 %	2341	964	255	18	3578

3.5.1 Pre imputation robustness checks

The comparisons for each `kml` partition was made for systematic random deletion of data prior to imputation. Each graphic indicates the sample size or number of observations for the cluster. Deviations are marginal to the 0 % curve, for example in Figures 3.22, in Figure 3.23 and Figure 3.24. When all the partitions are observed collectively, the largest deviation is seen in the 20 % random deletion (see Figure 3.25) when compared to 1 % and 5 %. For parition C, the marked deviation in the 5 % curve from the 0 % curve when compared with higher volumes of deletion, for example in Figure 3.25 could be a result of the random deletion of whole rabbits, i.e. all entries of the same ear tag.

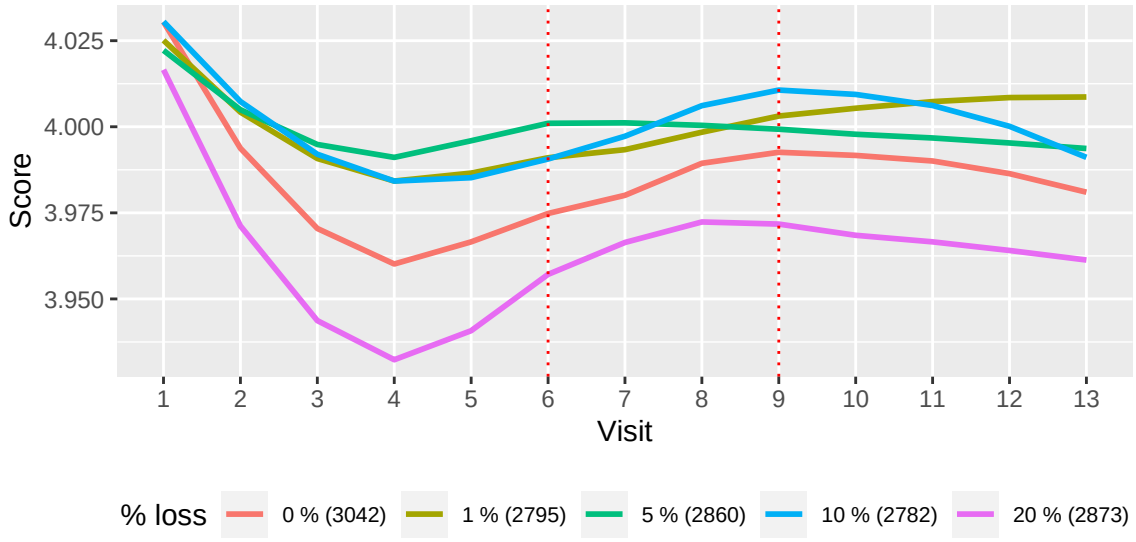

Figure 3.22: Comparison of pre imputation optimal partition A from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

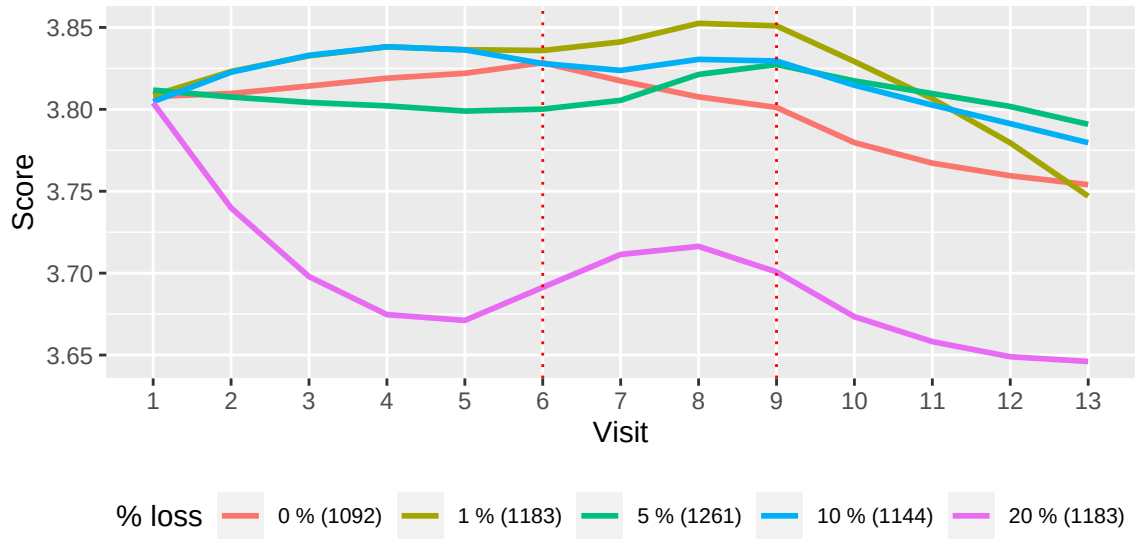

Figure 3.23: Comparison of pre imputation optimal partition B from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

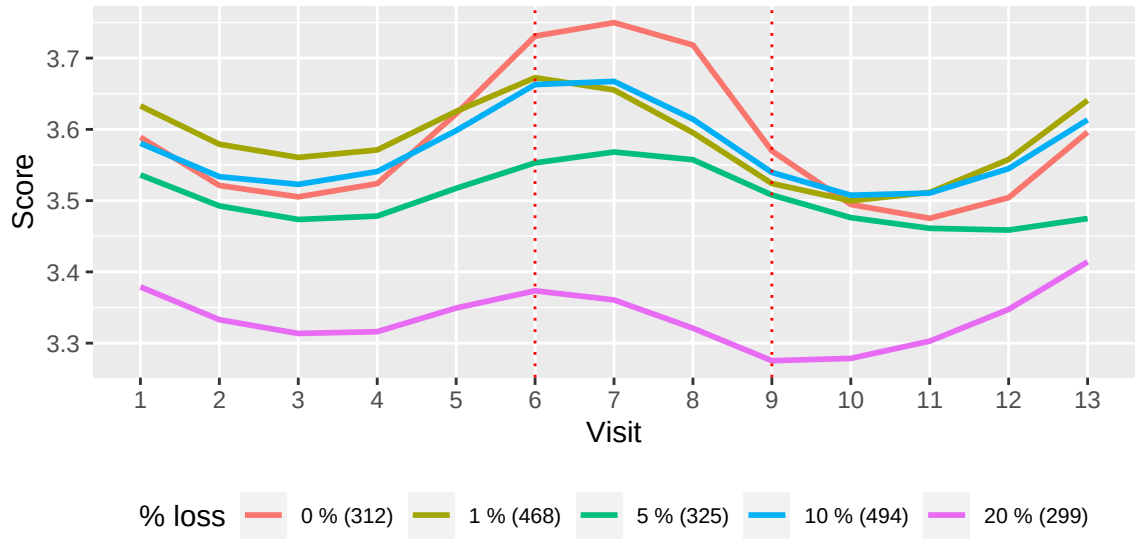

Figure 3.24: Comparison of pre imputation optimal partition C from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

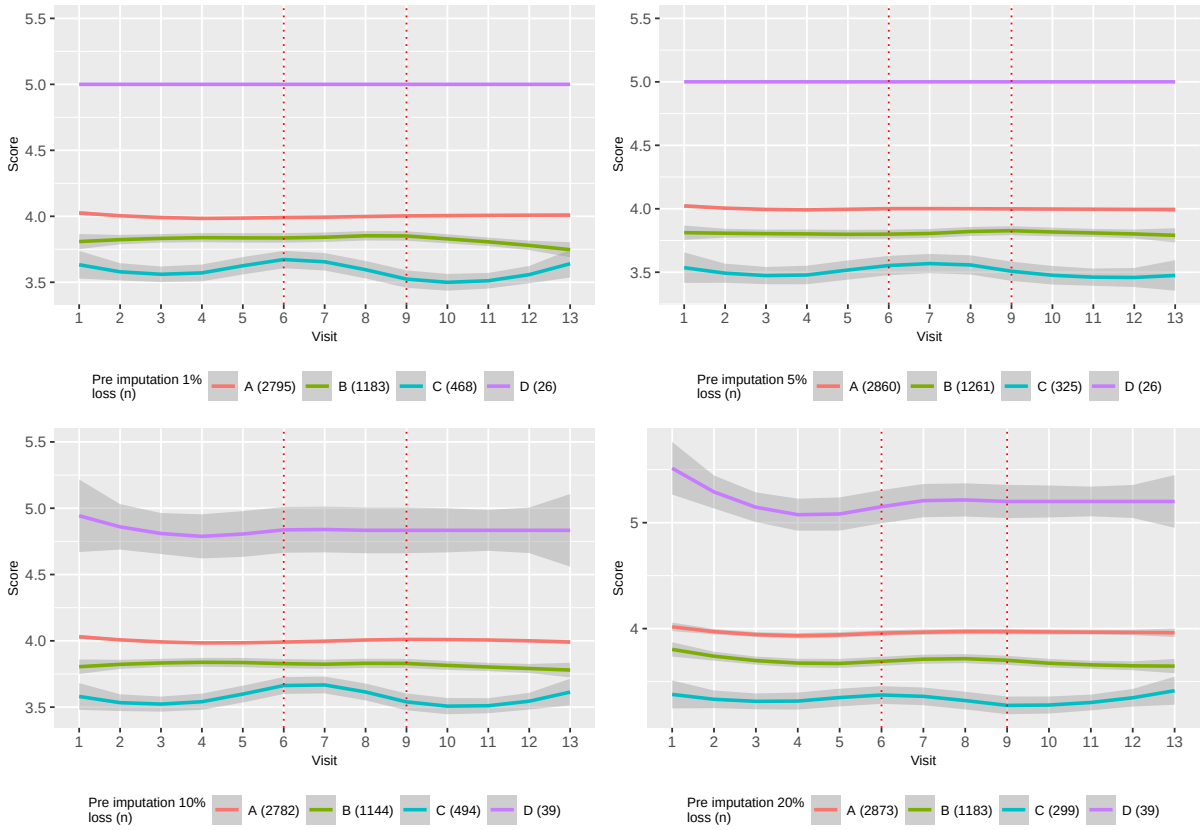

Figure 3.25: Trajectory of scores with random deletion for four partitions (pre-imputation) with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016).Indicated between vertical red lines are the northern hemisphere winter season.

3.5.2 Post imputation robustness checks

Each graphic indicates the sample size or number of observations for the cluster. Deviations are marginal to the 0 % curve, for example in Figures 3.29. When all the partitions are observed collectively similar to the pre imputation analysis, the largest deviation is also seen in the 20 % random deletion when compared to other random deletion.

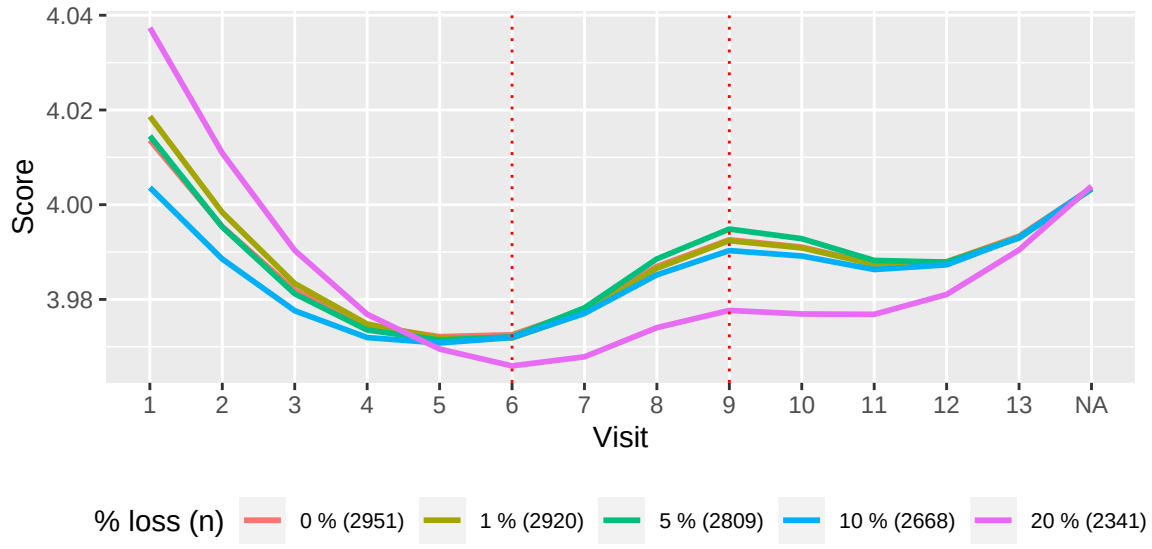

Figure 3.26: Comparison of post imputation optimal partition A from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

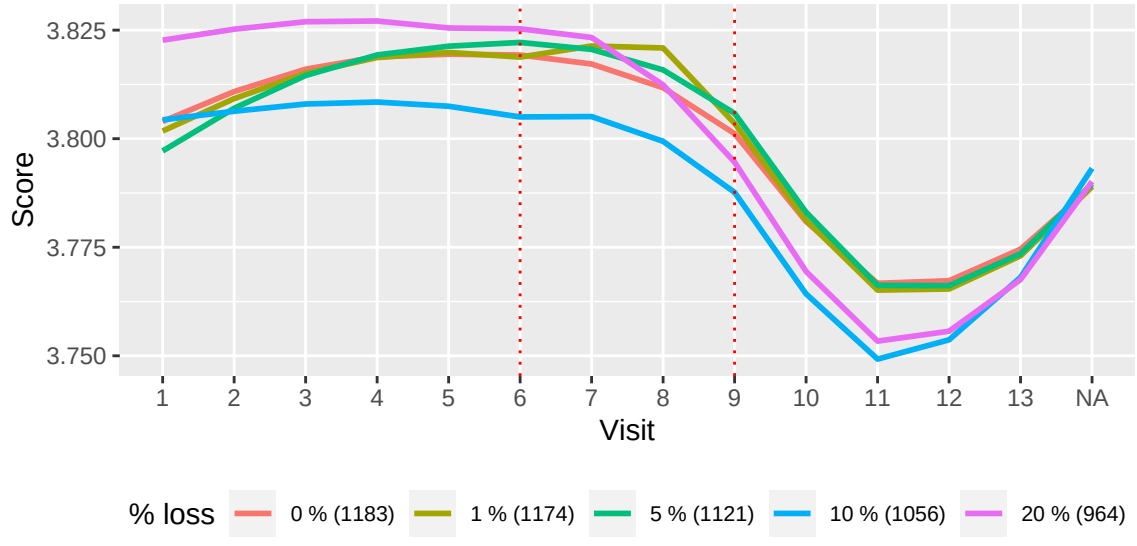

Figure 3.27: Comparison of post imputation optimal partition B from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

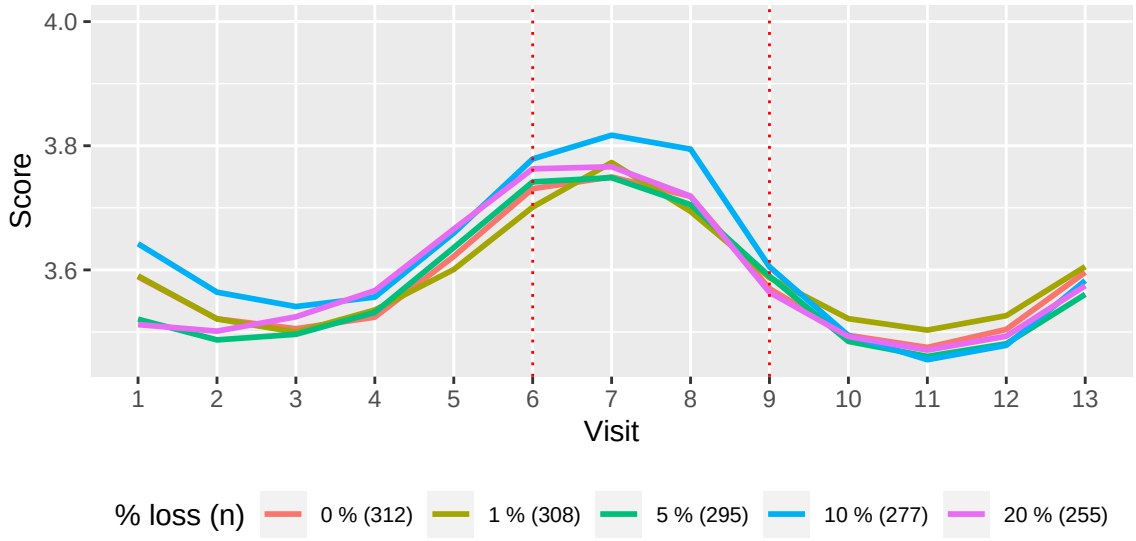

Figure 3.28: Comparison of post imputation optimal partition C from random deletion of data for two partitions with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season.

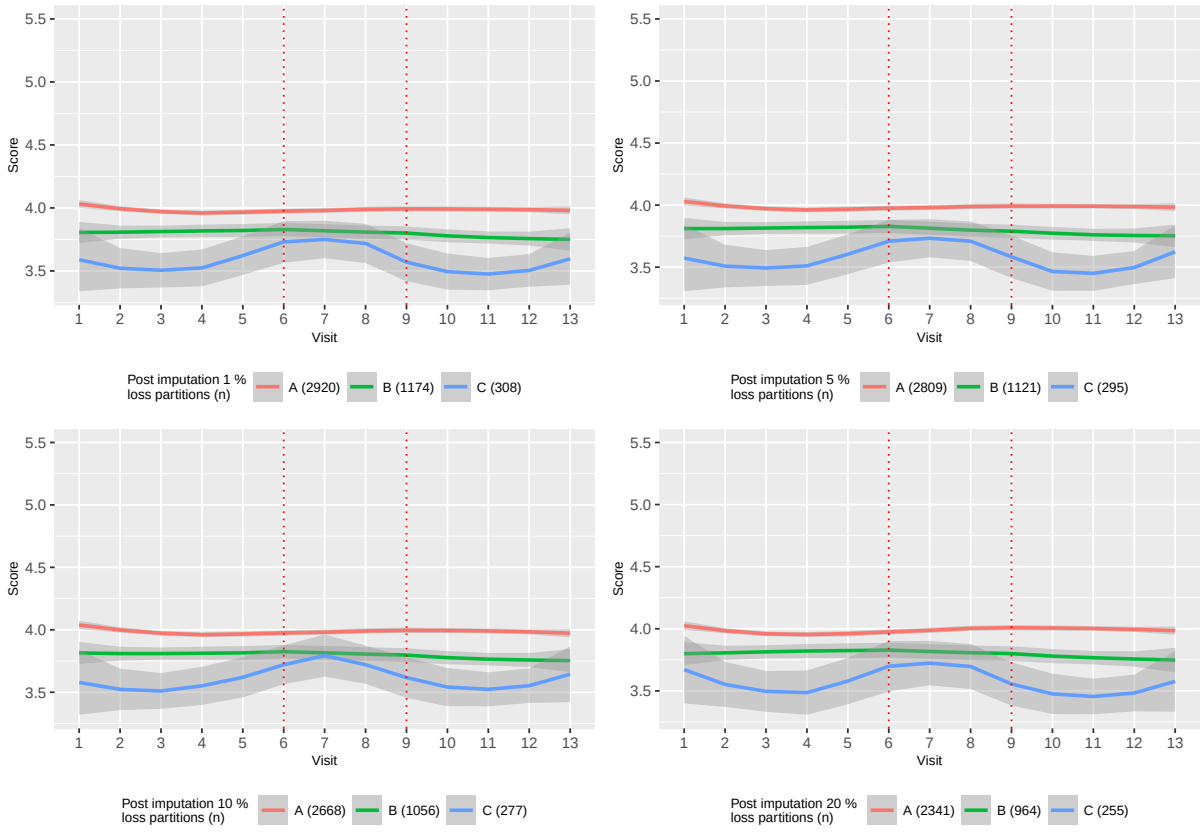

Figure 3.29: Trajectory of scores with random deletion for for four partitions (post imputation) with `loess` smoothing with banding of its time-point standard error, based on t-approximation (Wickham, 2016). Indicated between vertical red lines are the northern hemisphere winter season. Low partition D counts are not reflected.

Chapter 4

Discussion and Outlook

The analysis of disease progression of pododermatitis over time is a curious example of longitudinal cluster analysis. To understand the risk factors in a clustered setting, I performed the analysis by stratification by area and by two and four `kml` partitioning. This is followed by a mixed model analysis. A check of robusticity of `kml` was performed after systematic random deletion of data, where the deletion was performed prior and post imputation.

The correlation between pododermatitis of the two regions of the plantar surface of the does' hind paws is low. In addition, since factors affect mid-paw scores are not all the same to ones affecting heel scores, I did not take the mean of the scores for both regions and chose to focus on one region. In addition, a naive linear model summary output show that there is evidence of significant association between the two scores. The risk factors for both outcome variables could be more common than suggested by the directed acyclic graph.

The concordance between the four partition and four areas were not marked for any confident conclusions, thus I do not confirm any homogeneity between the areas and partitions. The concordance between the two and four partitions were neither marked.

A note about the mixed model choice of variables. The choice of ear tag, age (age by ear tag), weight by ear tag and area as random variables is from the belief that these may have individual variations and merit a respective intercept.

4.1 Stratification by area

Winter months show elevated scores for most barns. The decrease of scores seem to be greater in the first six months, followed by the second six months which suggests that factors observed within certain visits could have a progressive affect on the does later on. The direct associations of environmental factors to mean heel scores are difficult to confirm as well. Elevated scores are also observed with lower temperatures and often, highest relative humidity, except for barn 4 for the latter factor. Most does are in the pregnancy or lactation state and possess clean paws which means that these scores change while other factors are relatively constant. Claw lengths however, vary across visits and longer claws observed in summer months preceding the winter months could be investigated further to explain elevated scores are stratified by area.

Age and weight stratified by area have a non-linear increase with time due to new entries of does. Since non-moribound does were chosen, we cannot know if the does that do not reappear in following time points within the study were moribound looking but some positive selection bias cannot be ruled out due to the debilitating effects of pododermatitis to the welfare of rabbits. Age, through weight has a positive impact on mean heel scores (Ruchti *et al.*, 2019) and it can be seen that whilst age increases with subsequent visits, scores still improved between certain time points (see Figure 3.2). Weight increases during the winter months, a factor found to be impactful by the precedent study (Ruchti *et al.*, 2019). However, scores show a mixed trend after visit 6 to 13 even though weight decreases between visit 1 and 13. The trend of claw

length appears to be similar across most barns, with normal length claws moreso from winter on. The directed acyclic graph taken from Ruchti *et al.* (2019) suggests that claw length has a positive impact on mean heel scores but not on mean mid-paw scores. This could suggest more investigations of biomechanical factors on this disease could be meaningful. Furthermore, mean heel scores do increase during winter, when normal length claws are in higher proportions. Since pododermatitis takes time to develop, some variables may be a factor that progresses the disease over time but not immediately observed in the summer months. Furthermore, longer claws may reflect older age of rabbits which is also higher during the winter months.

4.2 Partitions from `kml` implementations and mixed model

Whilst the visual inspection of scores show the improvement of disease status per barn area when visit 1 is compared to visit 13, the same inspection in the two and four `kml` partitions show only marginal. The downside of the four partition case is that sample counts decrease with the more latter partitions (in this case partition D) and standard errors increase as a result. Furthermore, in both two and four partitions, there is no marked peak during the winter months as is observed in some barns in the stratification by area analysis. What is helpful is the separation seen in the two partition as one trajectory is above score four and the other below as it is known that score four and above is a disease state associated with pain. Mixed model analysis show that there are no variables that show a significant impact on scores, neither do any partition.

Currently generalised linear model assessment showed that groups have evidence of difference with respect of mean heel scores between partitians, thus `kml` was successful in creating groups that have evidence of significant difference. In the two and four partition case, only the visit variable had evidence of significant negative impact on the mean scores. This was estimate was -0.005 and -0.005 respectively. In the both cases, the estimate is below 1. Consultations with clinicians could aid in understanding the contextual value of this estimate. With respect to reproductive state, since only few does were not-pregnant in the two and four partitions across visits and most paws were clean, to control for weight and behavioural repertoires related to pregnancy and thereafter on welfare, further studies can include some of these important observations to control for hormonal influences in socialisation behaviours and the interaction of does in group housing. Of the time factor, since this disease develops over time, and has higher proportion of "too long" claws during the summer months across all partitions, a similar interpretation could be held from the one of scores stratified by area. The higher proportion of "too long" scores reflected during summer, suggests that claw length may be a time dependant factor, although this is not a statistically significant result according to the mixed modeling.

It is important to note that whilst we are able to analyse scores of full trajectories in the `kml` partitions due to imputation, the facility did not impute all other covariates per time point. This meant that partitioned data set had the same number of covariate observations with the partitioned data sets. This does not strengthen the mixed model analysis, however if future studies can recruit the same rabbit in all or most time points, there could be more covariates that could be used in a mixed model analysis.

4.3 The robusticity of `kml` implementation

Even though up to half the data was imputated, the performance of `kml` could be assessed through systematic random deletions of observations. I performed the systematic random deletion prior and after imputation of data and `kml` partitioning in view that it could aid future studies decide if sampling every fifth observation could still reached a similar, clinically acceptable result, for example. A note about sample size, for the pre-imputation experiment, sample sizes are stable due to imputation and the loss of observations are possible from deletion of entire ear tags that do not undergo imputation. However, for the post-imputation experiment, the deletion of data

was implemented after imputation and subsequent kml partitioning which is why the sample sizes are smaller than the former experiment.

It is important to recall that for the kml facility, full trajectories are required which is role of imputation. Thus I would recommend robusticity checks where imputation required is minimal.

With systematic random deletion on the current rate of imputation, it appears that kml performs similarly when trajectories are removed at magnitude 1-10%. For example, shifts in trajectories are marginal and do not change the disease states of pododermatitis. This is no surprise as almost half of the data was imputated with the trajectory mean. As such I am cautious to use the paritioned dataset to make confident inferences about the risk factors influencing neither partitions nor trajectories. The kml suffers from some drawbacks as cited as well in Genolini and Falissard (2010). For instance, there are no formal tests to evaluate the validity of the partitions. Furthermore, the number of clusters needs to be determined a priori, and the starting condition determined at random. In the optimization step, scores converges to a local maximum however one cannot be sure that this is a local or global maximum, therefore, whether the best partition has been found. To this point, if the number of redrawings (iterations) needed until the optimum is reached exceeds their maximum of 20 (Genolini and Falissard, 2010), this could create uncertainty in whether a global maximum and or the best partitians are created. The quality criterion which is not extensively examined in this study could be used to compare the quality of partitions without nominating number of clusters to aid with this uncertainty. I would recommend a coupling of this and designing a study with fuller trajectories. Further outlook to this study could also include male rabbits and assess their progression of pododermatitis and their reproductive state to control for mating potential as at least on buck is expected in each pen evaluated (Ruchti *et al.*, 2019). This would enable a possible comparison of weight and reproductive state, amongst all other factors except number of kindlings. The socialisation of rabbits in group housing could be studied with other variables not used in this study, such as proportion of wound bites and descriptives on the pododermatitis. It is also a valid question of repeating the entire analysis with mean mid-paw scores, as other factors associated on heel scores are not associated on mid-paw scores, according to the dag. Generalized linear models can still assess the between kml partitians difference to check for evidence for statistically different partitions. The convenience of imputation of missing data to create trajectories through k-means clustering, followed by statistical analysis on the partitioned data, may also suffer from the lack of known time-dependant covariates. Different ways of replacing missing values could also be examined. The kml facility also provides other manipulations and options on the algorithm which could be tested for its performance in pododermatitis scoring.

Chapter 5

Conclusions

A common lifestyle related disease among rabbits are pressure sores on hocks and feets, otherwise known as pododermatitis Mancinelli *et al.* (2014). As clinicians approximate several factors that contribute to this condition, this study performed analysis on whether disease scores improve or worsen over time, and if risk factors identified from a precedent study can explain this (Ruchti *et al.*, 2019). A k-means clustering method from `R` package `kml` was applied to a data set to create two and four kml clusters or partitions. Mixed modeling on these scenarios showed that some risk factors previously identified by clinicians had evidence of significant impact such as time. The `kml` method was able to show clear separation of trajectories, albeit from imputated data, which, with fuller original trajectories, may be why it could be useful for further research in pododermatitis, including the fate of bucks versus does.

Appendix A

Appendix

A.1 Descriptive statistics

Table A.1: Breeds of does per barn and their percentages across all areas. These are counts across all time points.

Area	n	percent
Barn 1	871	0.33
Barn 2	442	0.17
Barn 3	428	0.16
Barn 4	871	0.33

Table A.3: Breeds of does per barn for each barn. Hycole is represented across al barns. Hyla is only found in Barn 1 and Hylamax is disproportionately represented in barn 1 to 3.

	Barn 1	Barn 2	Barn 3	Barn 4
Hycole	5	436	386	871
Hyla	798	0	0	0
Hylamax	68	6	42	0

```
##
##  Kendall's rank correlation tau
##
## data:  df$meanPDheel and df$meanPDmid
## z = 7.4998, p-value = 6.392e-14
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## 0.140527
```

Figure A.1: (Naive)Assumption check of trend of mean mid-paw and heel score with summary statistics. Observations are naively assumed to be independant, only the association between the two scores are assessed here.

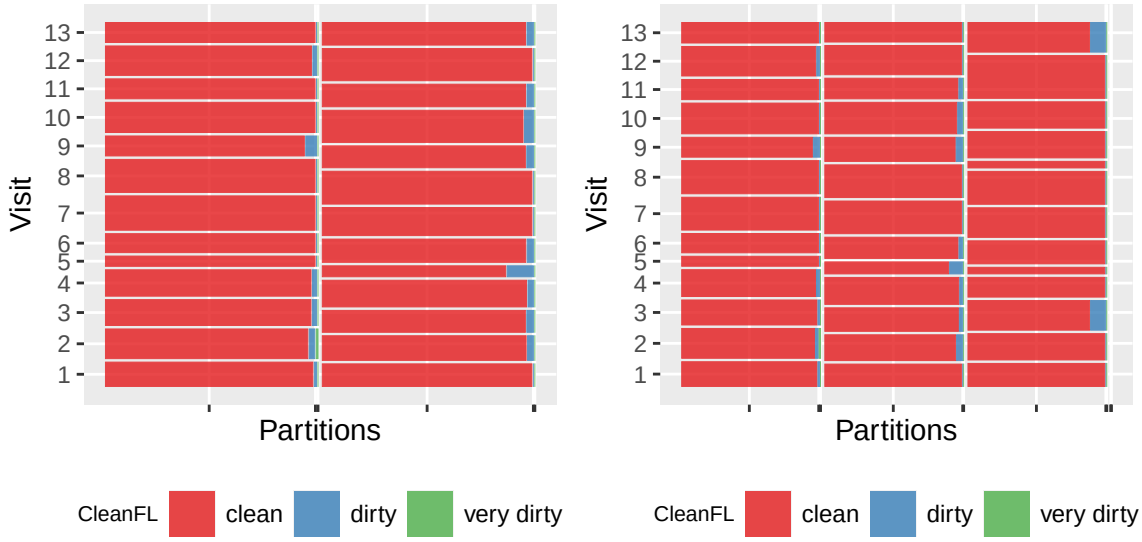

Figure A.2: CleanFL represents the ordinal variable of cleanliness of front left paw of the doe. The y-axis represents each visit in descending order from top to bottom. On the x-axis from left to right, represents two partitions of the `kml` implementation. The height of each rectangle and y-axes represent the proportion of does in that specific visit or time point. The width of the x-axis represent the proportion of a particular ordinal category it represents. Barn 1 and barn 2 is left and right respectively.

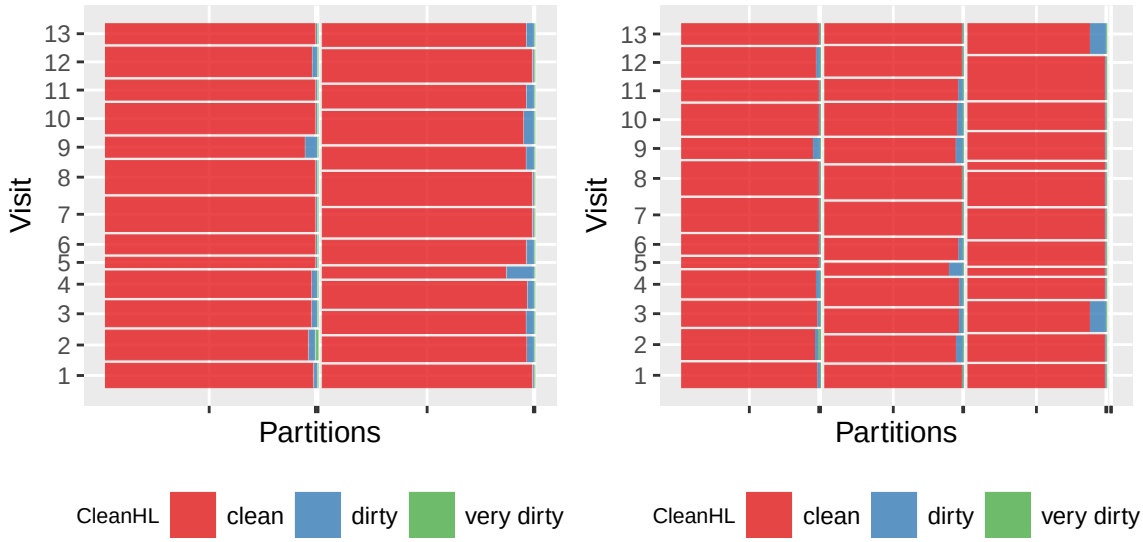

Figure A.3: CleanHL represents the ordinal variable of cleanliness of front left paw of the doe. The y-axis represents each visit in descending order from top to bottom. On the x-axis from left to right, represents two partitions of the `kml` implementation. The height of each rectangle and y-axes represent the proportion of does in that specific visit or time point. The width of the x-axis represent the proportion of a particular ordinal category it represents. Barn 1 and barn 2 is left and right respectively.

```
##
## Call:
## lm(formula = meanPDmid ~ meanPDheel, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.201  0.089  0.089  0.089  2.380
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.74933    0.16663  16.500  < 2e-16 ***
## meanPDheel   0.29041    0.04257   6.822 1.11e-11 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7178 on 2593 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.01763,Adjusted R-squared:  0.01726
## F-statistic: 46.55 on 1 and 2593 DF,  p-value: 1.11e-11
```

Figure A.4: (Naive)Assumption check of trend of mean mid-paw and heel score with summary statistics. Observations are naively assumed to be independant, only the association between the two scores are assessed here.

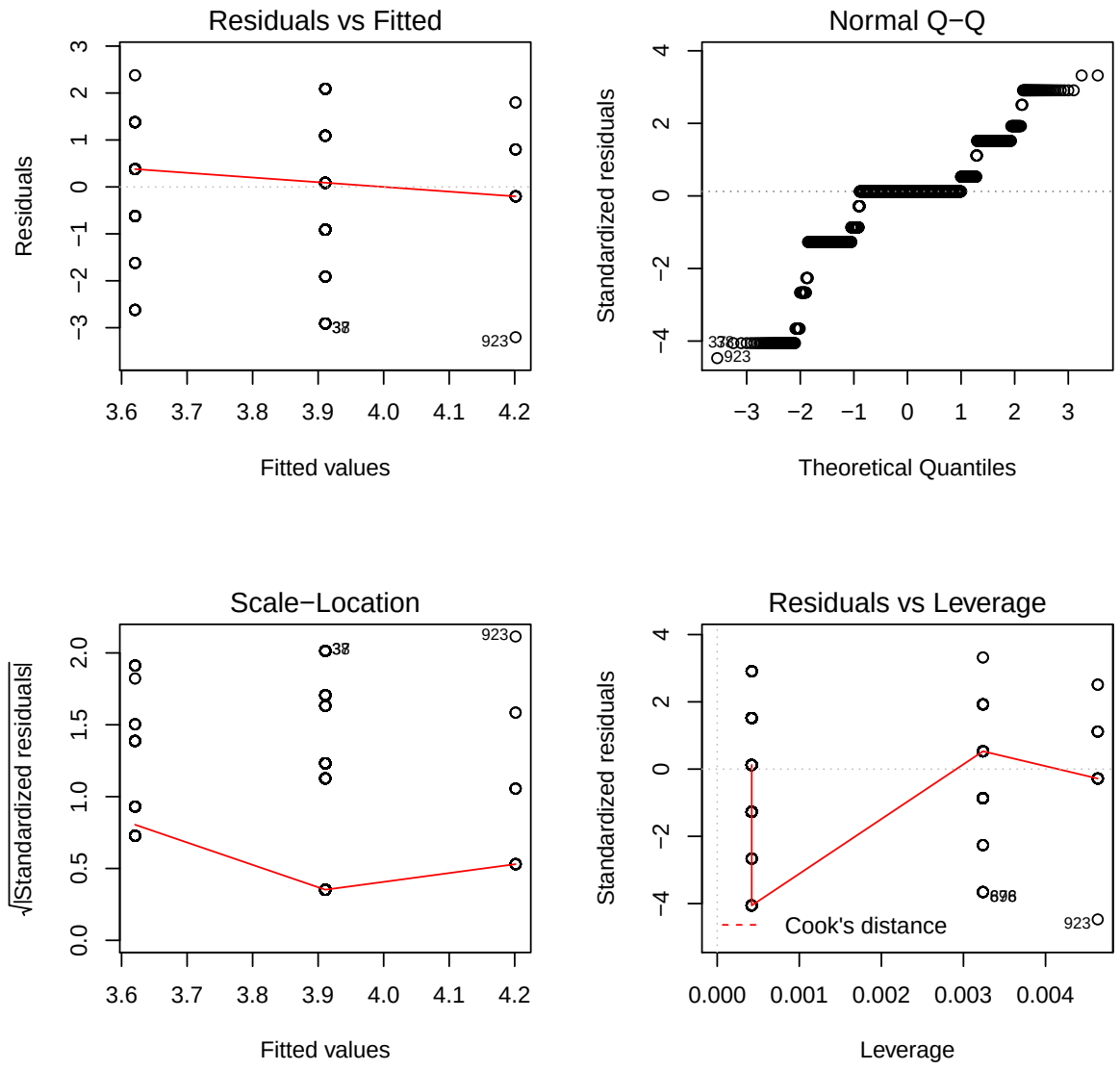

Figure A.5: (Naive) Assumption check of trend of mean mid-paw and heel score with summary statistics. Observations are naively assumed to be independant, only the association between the two scores are assessed here.

```
mod = summary(glm(kmlclusters2 ~ Claws + Hybrid + CleanFL + MoistFL + meanPDheel.x, data = df_longdf, family = "binomial"))
mod$coefficients
##                   Estimate   Std. Error      z value
## (Intercept)    3.24103192  76.24922833   0.04250577
## Clawstoo long  0.07111463   0.09754745   0.72902602
## HybridHyla     0.34937848   0.10009272   3.49054832
## HybridHylamax -0.35399303   0.26018447  -1.36054636
## CleanFL.L     -8.03213918 161.74469914  -0.04965937
## CleanFL.Q     -4.75328920  93.38370025  -0.05090063
## MoistFLmoist  -0.03449773   0.12586959  -0.27407513
## MoistFLwet    -0.24277966   0.25085664  -0.96780240
## meanPDheel.x  -2.08365623   0.13872317 -15.02024659
##                     Pr(>|z|)
## (Intercept)   9.660955e-01
## Clawstoo long 4.659857e-01
## HybridHyla    4.820304e-04
## HybridHylamax 1.736571e-01
## CleanFL.L     9.603938e-01
## CleanFL.Q     9.594047e-01
## MoistFLmoist  7.840269e-01
## MoistFLwet    3.331431e-01
## meanPDheel.x  5.410606e-51
```

Figure A.6: Generalized linear regression output of two partition setting to show that two partitions have evidence of significant difference between come covariates and their levels.

```
mod = summary(glm(kmlclusters4 ~ Claws + Hybrid + CleanFL + MoistFL + meanPDheel.x, data = df_longdf, family = "binomial"))
mod$coefficients
##                   Estimate   Std. Error      z value
## (Intercept)    4.30772539  76.49017665   0.05631737
## Clawstoo long  0.24530059   0.08913864   2.75189951
## HybridHyla     0.13663598   0.09341486   1.46267927
## HybridHylamax -0.59521497   0.23815192  -2.49930794
## CleanFL.L     -8.38437360 162.25443473  -0.05167423
## CleanFL.Q     -5.25812778  93.67795076  -0.05612983
## MoistFLmoist   0.06122815   0.11500718   0.53238549
## MoistFLwet    -0.21195912   0.23225640  -0.91260829
## meanPDheel.x  -2.23223756   0.15939504 -14.00443571
##                     Pr(>|z|)
## (Intercept)   9.550890e-01
## Clawstoo long 5.925070e-03
## HybridHyla    1.435552e-01
## HybridHylamax 1.244361e-02
## CleanFL.L     9.587883e-01
## CleanFL.Q     9.552384e-01
## MoistFLmoist  5.944590e-01
## MoistFLwet    3.614486e-01
## meanPDheel.x  1.464382e-44
```

Figure A.7: Generalized linear regression output of four partition setting to show that four partitions have evidence of significant difference between come covariates and their levels

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters2 %in% c("A", "B"),]))
mixed1$coefficients
##                         Estimate  Std. Error        df
## (Intercept)        3.9374105192 0.277338451 2466.6994
## as.numeric(visit) -0.0049661147 0.002217069 2013.8832
## Clawstoo long      0.0050730512 0.016286780 1628.1534
## Weight             0.0257910517 0.014298759  627.7528
## Age                0.0002043626 0.001197596  269.7793
## Temperature       -0.0014360646 0.001692276 2540.1026
## RelativeHumidity  -0.0017497656 0.001538211 2502.2453
## HybridHyla         0.0034592042 0.064444992  351.2812
## HybridHylamax      0.0484678834 0.052371509  284.7041
##                        t value     Pr(>|t|)
## (Intercept)       14.19713174 4.933018e-44
## as.numeric(visit) -2.23994620 2.520312e-02
## Clawstoo long      0.31148276 7.554735e-01
## Weight             1.80372653 7.175347e-02
## Age                0.17064412 8.646315e-01
## Temperature       -0.84859958 3.961841e-01
## RelativeHumidity  -1.13753293 2.554245e-01
## HybridHyla         0.05367685 9.572231e-01
## HybridHylamax      0.92546280 3.555090e-01
```

Figure A.8: Linear mixed model output of two partition setting to show that visit has an evidence of significant impact to mean pododermatitis scores.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters2 %in% c("A"),]))
mixed1$coefficients
##                         Estimate   Std. Error          df
## (Intercept)        4.1512060598 0.1110948131   55.672026
## as.numeric(visit) -0.0029491070 0.0018069365 1009.855050
## Clawstoo long     -0.0163125057 0.0127862755  672.525656
## Weight             0.0028937806 0.0111082305 1803.770472
## Age                0.0006099366 0.0008151875 1804.782946
## Temperature        0.0002328175 0.0014049015  843.881223
## RelativeHumidity  -0.0028185939 0.0011576148   33.577940
## HybridHyla        -0.0082303734 0.0147749423    1.580092
## HybridHylamax      0.0410535960 0.0275904396  101.469154
##                        t value     Pr(>|t|)
## (Intercept)       37.3663355 4.084136e-41
## as.numeric(visit) -1.6321033 1.029695e-01
## Clawstoo long     -1.2757824 2.024729e-01
## Weight             0.2605078 7.945018e-01
## Age                0.7482163 4.544272e-01
## Temperature        0.1657181 8.684185e-01
## RelativeHumidity  -2.4348288 2.036992e-02
## HybridHyla        -0.5570494 6.461077e-01
## HybridHylamax      1.4879646 1.398615e-01
```

Figure A.9: Linear mixed model output of partition A in two partition case.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters2 %in% c( "B"),]))
mixed1$coefficients
##                        Estimate  Std. Error         df
## (Intercept)        3.6869146085 0.372389815 111.594397
## as.numeric(visit) -0.0071329777 0.005653565 642.526431
## Clawstoo long      0.0459540622 0.038342604 169.520918
## Weight             0.0408197328 0.033208646 132.946582
## Age               -0.0029796581 0.002679121 356.537374
## Temperature       -0.0057540862 0.004664583 526.881774
## RelativeHumidity   0.0002623515 0.003890591 110.291741
## HybridHyla        -0.0177310405 0.042218631   1.025566
## HybridHylamax     -0.0805728016 0.107179427 122.657047
##                       t value     Pr(>|t|)
## (Intercept)         9.9006859 5.773575e-17
## as.numeric(visit) -1.2616778 2.075226e-01
## Clawstoo long      1.1985118 2.323907e-01
## Weight             1.2291899 2.211717e-01
## Age               -1.1121776 2.668112e-01
## Temperature       -1.2335694 2.179134e-01
## RelativeHumidity   0.0674323 9.463596e-01
## HybridHyla        -0.4199814 7.454912e-01
## HybridHylamax     -0.7517562 4.536378e-01
```

Figure A.10: Linear mixed model output of partition B in two partition case.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters4 %in% c("A", "B", "C", "D"),]))
mixed1$coefficients
##                        Estimate  Std. Error        df
## (Intercept)        3.9374105192 0.277338451 2466.6994
## as.numeric(visit) -0.0049661147 0.002217069 2013.8832
## Clawstoo long      0.0050730512 0.016286780 1628.1534
## Weight             0.0257910517 0.014298759  627.7528
## Age                0.0002043626 0.001197596  269.7793
## Temperature       -0.0014360646 0.001692276 2540.1026
## RelativeHumidity  -0.0017497656 0.001538211 2502.2453
## HybridHyla         0.0034592042 0.064444992  351.2812
## HybridHylamax      0.0484678834 0.052371509  284.7041
##                       t value     Pr(>|t|)
## (Intercept)       14.19713174 4.933018e-44
## as.numeric(visit) -2.23994620 2.520312e-02
## Clawstoo long      0.31148276 7.554735e-01
## Weight             1.80372653 7.175347e-02
## Age                0.17064412 8.646315e-01
## Temperature       -0.84859958 3.961841e-01
## RelativeHumidity  -1.13753293 2.554245e-01
## HybridHyla         0.05367685 9.572231e-01
## HybridHylamax      0.92546280 3.555090e-01
```

Figure A.11: Linear mixed model output of four partition setting to show that visit has an evidence of significant impact to mean pododermatitis scores.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters4 %in% c("A"),]))
mixed1$coefficients
##                         Estimate  Std. Error       df
## (Intercept)        4.0977071850 0.103638371 1549.707
## as.numeric(visit) -0.0018288015 0.001681331 1591.141
## Clawstoo long     -0.0123959328 0.012152111 1588.047
## Weight             0.0001484346 0.010427897 1271.325
## Age                0.0008552174 0.000757249 1544.679
## Temperature        0.0002475980 0.001327328 1598.353
## RelativeHumidity  -0.0017499473 0.001078047 1593.131
## HybridHyla        -0.0121448506 0.012510441 1561.622
## HybridHylamax      0.0358235769 0.024758394 1550.798
##                        t value      Pr(>|t|)
## (Intercept)       39.53851397 5.369481e-237
## as.numeric(visit) -1.08771047  2.768877e-01
## Clawstoo long     -1.02006418  3.078534e-01
## Weight             0.01423438  9.886452e-01
## Age                1.12937401  2.589154e-01
## Temperature        0.18653877  8.520460e-01
## RelativeHumidity  -1.62325668  1.047324e-01
## HybridHyla        -0.97077715  3.318096e-01
## HybridHylamax      1.44692651  1.481196e-01
```

Figure A.12: Linear mixed model output of partition A in two partition case.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters4 %in% c("B"),]))
mixed1$coefficients
##                         Estimate  Std. Error         df
## (Intercept)        3.839339542 0.311281960 182.289919
## as.numeric(visit) -0.006094980 0.004920964 758.115024
## Clawstoo long      0.054531971 0.032007510 633.725649
## Weight             0.039574295 0.028418514 322.974176
## Age               -0.003329421 0.002362849 720.542889
## Temperature       -0.003793008 0.003725849 719.968768
## RelativeHumidity  -0.001648212 0.003307794 160.432662
## HybridHyla         0.007871768 0.060843668   2.149003
## HybridHylamax     -0.043480173 0.105389443 146.086007
##                        t value     Pr(>|t|)
## (Intercept)       12.3339610 8.322302e-26
## as.numeric(visit) -1.2385743 2.158865e-01
## Clawstoo long      1.7037243 8.892285e-02
## Weight             1.3925533 1.647128e-01
## Age               -1.4090704 1.592458e-01
## Temperature       -1.0180252 3.090079e-01
## RelativeHumidity  -0.4982813 6.189682e-01
## HybridHyla         0.1293770 9.081635e-01
## HybridHylamax     -0.4125667 6.805290e-01
```

Figure A.13: Linear mixed model output of partition B in two partition case.

```
mixed1 <- summary(firstmixedmod <-
                    lmer(meanPDheel.y ~ as.numeric(visit) + Claws + Weight +
                           Age + Temperature + RelativeHumidity + (1|EarTag) +
                           Hybrid + (1|Area) + (Age|EarTag) + (Weight|EarTag),
                         data = df_longdf[df_longdf$kmlclusters4 %in% c("C"),]))
mixed1$coefficients
##                        Estimate  Std. Error          df
## (Intercept)        2.809809649 1.074119498  31.2056359
## as.numeric(visit) -0.001275224 0.014884920 136.7356932
## Clawstoo long     -0.099059779 0.129873898 122.8369484
## Weight             0.011094186 0.104072071   7.1489901
## Age               -0.002046401 0.006478125   7.6501992
## Temperature        0.003098816 0.014771543 116.7053875
## RelativeHumidity   0.010803956 0.011181253  61.6127080
## HybridHyla        -0.019889946 0.099526736   0.1040884
## HybridHylamax     -0.045333026 0.238173051   3.1224181
##                        t value   Pr(>|t|)
## (Intercept)        2.61591904 0.01359207
## as.numeric(visit) -0.08567221 0.93185231
## Clawstoo long     -0.76273817 0.44708181
## Weight             0.10660099 0.91803544
## Age               -0.31589410 0.76052834
## Temperature        0.20978280 0.83420298
## RelativeHumidity   0.96625630 0.33769350
## HybridHyla        -0.19984525 0.94339718
## HybridHylamax     -0.19033650 0.86075255
```

Figure A.14: Linear mixed model output of partition C in two partition case.

A.2 R code

```
load("../Data/data.Rdat")
# changes will be made to GK's file and saved as "df.Rda"
# the following code will save it with changes for analysis.
# loading packages
df$WAPP_cont <- as.numeric(df$WAPP_cont)
df$Area <- NA
df$Area <- ifelse(df$Barn == "1" & df$Farm == "11", "Barn1",
                  ifelse(df$Barn == "1" & df$Farm == "13", "Barn2",
                         ifelse(df$Barn == "2" & df$Farm == "13", "Barn3",
                                ifelse(df$Barn == "1" &
                                                              df$Farm == "17", "Barn4", NA)) ))
df$Area = factor(df$Area)
######## var matching #####
df$EarTag <- as.numeric(df$EarTag)
which(df$EarTag[df$Area == "Barn1"] %in% df$EarTag[df$Area == "Barn2"])
#[1] 526 540 607 674 741 808 to rename as they are duplications
#none:
which(df$EarTag[df$Area == "Barn1"] %in% df$EarTag[df$Area == "Barn3"])
which(df$EarTag[df$Area == "Barn1"] %in% df$EarTag[df$Area == "Barn4"])
which(df$EarTag[df$Area == "Barn2"] %in% df$EarTag[df$Area == "Barn3"])
which(df$EarTag[df$Area == "Barn2"] %in% df$EarTag[df$Area == "Barn4"])
which(df$EarTag[df$Area == "Barn3"] %in% df$EarTag[df$Area == "Barn4"])

sort(unique(df$EarTag[df$Area == "Barn1"]))
sort(unique(df$EarTag[df$Area == "Barn2"]))
sort(unique(df$EarTag[df$Area == "Barn3"]))
sort(unique(df$EarTag[df$Area == "Barn4"]))

# sort
sort(unique(df$EarTag))
# EarTag 1 to 6 are available labels to reassign to

# Reassign duplicate to new labels
"1" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "526"]
"2" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "540"]
"3" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "607"]
"4" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "674"]
"5" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "741"]
"6" -> df$EarTag[df$Area == "Barn1" & df$EarTag == "808"]

#check
which(df$EarTag[df$Area == "Barn1"] %in% df$EarTag[df$Area == "Barn3"])

df[df$EarTag == "1",]
df[df$EarTag == "2",]

######## remove missing values #####
dim(df)
df <- df[!df$EarTag == "11353",]
######## var meanPDheel : Mean scores from L and R limbs
df$meanPDheel <- rowMeans(df[c('PDHLH', 'PDHRH')], na.rm=TRUE)
df$meanPDmid <- rowMeans(df[c('PDHLM', 'PDHRM')], na.rm=TRUE)
df$Hybrid[df$Hybrid == "F1"] <- "Hylamax"
######## var Claw length #####
#recoding Claw length:
df$Claws[df$Claws=="1 torn out HR, rest normal"]<-"normal"
df$Claws[df$Claws=="1 torn out HR, rest too long"]<-"too long"
df$Claws[df$Claws=="1 torn out VL, rest normal"]<-"normal"
df$Claws[df$Claws=="mormal"]<-"normal"
df$Claws[df$Claws=="normal & torn out 1xvl"]<-"normal"
df$Claws[df$Claws=="too lang"]<-"too long"
df$Claws[df$Claws=="too long, one torn out HL"]<-"too long"
df$Claws[df$Claws=="too long, one torn out HR"]<-"too long"
df$Claws[df$Claws=="torn out (2xhl)/too long"]<-"too long"
df$ReproductiveState[df$ReproductiveState=="only-lacting"]<-
       "only-lactating"
######## var mmyy #####
df$mmyy[df$visit == "1"] <- "July2016"
df$mmyy[df$visit == "2"] <- "August2016"
df$mmyy[df$visit == "3"] <- "September2016"
df$mmyy[df$visit == "4"] <- "October2016"
df$mmyy[df$visit == "5"] <- "November2016"
df$mmyy[df$visit == "6"] <- "December2016"
df$mmyy[df$visit == "7"] <- "January2017"
df$mmyy[df$visit == "8"] <- "February2017"
df$mmyy[df$visit == "9"] <- "March2017"
df$mmyy[df$visit == "10"] <- "April2017"
df$mmyy[df$visit == "11"] <- "May2017"
df$mmyy[df$visit == "12"] <- "June2017"
df$mmyy[df$visit == "13"] <- "endJune2017"
df$mmyy <- factor(df$mmyy,
      levels = c("July2016", "August2016", "September2016", "October2016",
      "November2016", "December2016", "January2017",
        "February2017", "March2017","April2017",
      "May2017", "June2017", "endJune2017"),
ordered = TRUE)
df$visit <- factor(df$visit, levels = c("1", "2", "3", "4",
                                        "5", "6", "7", "8",
        "9", "10", "11", "12",
                                        "13"), ordered = TRUE)
######## var cat #####
df$cat <- NA
df$cat <- ifelse(df$Area == "Barn1", "A",
                 ifelse(df$Area == "Barn2", "B",
```

```
                      ifelse(df$Area == "Barn3", "C",
                             ifelse(df$Area == "Barn4", "D", NA)) ))
######## var MdAge, MdWeight, mdScore, (MEDIANS) #####
df %>%
        group_by(visit, cat) %>%
        mutate(mdAge = median(Age)) -> df
df %>%
        group_by(visit, cat) %>%
        mutate(mdWeight = median(Weight)) -> df
df %>%
        group_by(visit, cat) %>%
        mutate(mdScore = median(meanPDheel)) -> df
######## var Clean #####
df$CleanFR <- factor(df$CleanFR, levels =
                                                                    c("clean", "dirty", "very dirty"),
                                                                ordered = TRUE )
df$CleanFL <- factor(df$CleanFR, levels =
                                                                    c("clean", "dirty", "very dirty"),
                                                                ordered = TRUE )
df$CleanHL <- factor(df$CleanFR, levels =
                                                                    c("clean", "dirty", "very dirty"),
                                                                ordered = TRUE )
df$CleanHR <- factor(df$CleanFR, levels =
                                                                    c("clean", "dirty", "very dirty"),
                                                                ordered = TRUE )
# save
save(df, file = "../Data/df.Rda")
```

```
load("../Data/df.Rda")
library(kml)
library(tidyverse)
library(ggplot2)
library(janitor)
dev.off()
######## Cld object needed for kml #####
shortdf <- df[, c(3, 40, 37, 38, 41)]
#str(shortdf)
spread(shortdf, mmyy, meanPDheel) -> widedf0
#spread(df, mmyy, meanPDheel) -> testwide
head(widedf0)
sum(is.na(widedf0))
#class(widedf0)
#names(widedf0) #is data.frame
widedf1 <- as.matrix(widedf0[, 4:16])
#class(widedf1)
#head(widedf1)
sum(is.nan(widedf1))
widedf1[is.nan(widedf1)] <- NA
widedf1 <- imputation(widedf1, "trajMean")
# I do not think its appropriate to impute actually
# head(widedf1)
# colnames(widedf1)
sum(is.na(widedf1)) # 13
class(widedf1)
# create cld for matrix object pour widedf1 which has been imputated
mycld <- clusterLongData(widedf1, timeInData = 1:13)
#this likes matrices, the vignette said both
save(mycld, file = "../Data/cldSDQ.Rdata")

######### nbCluster = 2 #####
# slow kml not needed
# kml(mycld, toPlot = "both") # runs well, straight from the paper
#choice(mycld)
#plotAllCriterion(mycld) # works # show case
# from now on we use fast kml,
kml(mycld, nbClusters = 2, parAlgo = parALGO(distance = function(x, y)
  +    cor(x, y), saveFreq = 10)) #fastkml

choice(mycld)
plotAllCriterion(mycld) # works # show case

######## nbCluster = 4 #####
#kml(mycld, nbClusters = 4, toPlot = "both")  # slow kml
kml(mycld, nbClusters = 4, parAlgo = parALGO(distance = function(x, y)
  +    cor(x, y), saveFreq = 10)) #fastkml
choice(mycld)
plotAllCriterion(mycld)

######## df = df_longdf : creating df for nbClusters = 2 clusters
str(widedf1)
widedf1 <- as.data.frame(widedf1)
#widedf1$cluster <- widedf0$cluster
widedf1$EarTag <- widedf0$EarTag
widedf1$Area <- widedf0$Area
widedf1$kmlclusters2 <- getClusters(mycld, 2,
                                    asInteger = FALSE)

######## creating df for nbClusters = 4 clusters #####
widedf1$kmlclusters4 <- getClusters(mycld, 4,
                                    asInteger = FALSE)
save(widedf1, file = "../Data/widedf1.Rda")

######## Long form df from kml partitions: for glm and lmer models #####
#df_long <- gather(widedf0, "mmyy",
```

```
# "meanPDheel", -EarTag, -Area, -cluster, -kmlclusters2, -kmlclusters4)
gather(widedf1, mmyy, meanPDheel, -EarTag,
       -Area, -kmlclusters4, -kmlclusters2) -> testlong

testlong$mmyy <- factor(testlong$mmyy,
                        c("July2016", "August2016",
                              "September2016", "October2016",
                          "November2016", "December2016",
                              "January2017", "February2017",
                           "March2017", "April2017",
                              "May2017", "June2017", "endJune2017"),
                        ordered = TRUE)
names(testlong)
#df_longdf0 <- full_join(testlong, df, by = "EarTag")
df_longdf <- full_join(df, testlong, by =
                         c("EarTag" = "EarTag", "mmyy" = "mmyy",
                              "Area" = "Area"))

# first save but see second save below
save(df_longdf, file = "../Data/df_longdf.Rda")
# [1] "ID"                "Pen"
# [3] "EarTag"            "Hybrid"
# [5] "Age"               "NoKindlings"
# [7] "Weight"            "TimeD"
# [9] "CleanFR"           "CleanFL"
# [11] "CleanHR"           "CleanHL"
# [13] "MoistFR"           "MoistFL"
# [15] "MoistHR"           "MoistHL"
# [17] "PDFR"              "PDFL"
# [19] "PDHRM"             "PDHRH"
# [21] "PDHLM"             "PDHLH"
# [23] "Claws"             "WoundBite"
# [25] "WoundFeetHR"       "WoundFeetHL"
# [27] "RemarksDoe"        "visit"
# [29] "ID.Farm"           "Barn"
# [31] "Farm"              "ReproductiveState"
# [33] "WAPP"              "WAPP_cont"
# [35] "Temperature"       "RelativeHumidity"
# [37] "Area"              "meanPDheel.x"
# [39] "meanPDmid"         "mmyy"
# [41] "cat"               "mdAge"
# [43] "mdWeight"          "mdScore"
# [45] "kmlclusters2"      "kmlclusters4"
# [47] "meanPDheel.y"
# remove unimputated data
df_longdf <- df_longdf[, -c(38)]
names(df_longdf)
"meanPDheel" -> colnames(df_longdf)[47]
# final save for df_longdf
save(df_longdf, file = "../Data/df_longdf.Rdat")


######## concordance with areas and with each other #####
# 2 kml
df_longdf %>%
  tabyl(Area, kmlclusters2) %>%
  adorn_percentages() %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> area2kml
save(area2kml, file = "../Data/area2kml.Rda")
# 4 kml
df_longdf %>%
  tabyl(Area, kmlclusters4)  %>%
  adorn_percentages() %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame()  -> area4kml
save(area4kml, file = "../Data/area4kml.Rda")

# concordance of 2kml vs 4kml

df_longdf %>%
  tabyl(kmlclusters2, kmlclusters4) %>%
  adorn_percentages() %>%
  adorn_rounding(2) %>%
  as.data.frame()  -> conkml

colnames(conkml) <- c( " ", "A", "B", "C", "D")
save(conkml, file = "../Data/conkml.Rda")

######## sample size per cluster #####
# load("../Data/df_longdf.Rda")
df_longdf %>%
  tabyl(kmlclusters2) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> twokmlcounts
colnames(twokmlcounts) <- c("kml clusters", "n", "(%)")
save(twokmlcounts, file = "../Data/twokmlcounts.Rda")
df_longdf %>%
  tabyl(kmlclusters4) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> fourkmlcounts
colnames(fourkmlcounts) <- c("kml clusters", "n", "(%)")
save(fourkmlcounts, file = "../Data/fourkmlcounts.Rda")
###### score vs kml

a = paste( "A", " (", fourkmlcounts$n[1], ")", sep = "")
```

```
b = paste( "B", " (", fourkmlcounts$n[2], ")", sep = "")
c = paste( "C", " (", fourkmlcounts$n[3], ")", sep = "")
d = paste( "D", " (", fourkmlcounts$n[4], ")", sep = "")
ggplot(df_longdf,
       aes(x = visit, y = meanPDheel.y,
           colour = kmlclusters4, group = kmlclusters4)) +
  geom_smooth(method = "loess") +
  scale_colour_discrete(name = "Four kml\npartitians (n)",
                        labels=c(a, b, c, d)) +
  labs(x = "Visit", y = "Score") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "Visit",
                                                        limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

load("../Data/twokmlcounts.Rda")
a = paste( "A", " (", twokmlcounts$n[1], ")", sep = "")
b = paste( "B", " (", twokmlcounts$n[2], ")", sep = "")
ggplot(df_longdf,
       aes(x = visit, y = meanPDheel.y,
               colour = kmlclusters2, group = kmlclusters2)) +
  geom_smooth(method = "loess") +
  labs(x = "Visit", y = "Score") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
          expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_colour_discrete(name = "Two kml\npartitians (n)",
                        labels=c(a, b)) +
  scale_x_discrete(name = "Visit",
                                                        limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))



####### Descriptive stats tables #####
# mean score
# kmlclusters2
load("../Data/df_longdf.Rda")
mean(df_longdf[df_longdf$kmlclusters2 == "B",]$meanPDheel.y, na.rm = TRUE)
mean(df_longdf[df_longdf$kmlclusters2 == "A",]$meanPDheel.y, na.rm = TRUE)

#kmlclusters4
mean(df_longdf[df_longdf$kmlclusters4 == "A",]$meanPDheel.y, na.rm = TRUE)
sd(df_longdf[df_longdf$kmlclusters4 == "A",]$meanPDheel.y, na.rm = TRUE)
mean(df_longdf[df_longdf$kmlclusters4 == "B",]$meanPDheel.y, na.rm = TRUE)
mean(df_longdf[df_longdf$kmlclusters4 == "C",]$meanPDheel.y, na.rm = TRUE)
mean(df_longdf[df_longdf$kmlclusters4 == "D",]$meanPDheel.y, na.rm = TRUE)

# table for score
l = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters2 == "A",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters2 == "A",]$meanPDheel.y,
                 na.rm = TRUE))
m = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters2 == "B",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters2 == "B",]$meanPDheel.y,
                 na.rm = TRUE))

n = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters4 == "A",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters4 == "A",]$meanPDheel.y,
                 na.rm = TRUE))
p = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters4 == "B",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters4 == "B",]$meanPDheel.y,
                 na.rm = TRUE))
q = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters4 == "C",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters4 == "C",]$meanPDheel.y,
                 na.rm = TRUE))
r = sprintf("%.1f (%.1f)",
            mean(df_longdf[df_longdf$kmlclusters4 == "D",]$meanPDheel.y,
                         na.rm = TRUE),
            sd(df_longdf[df_longdf$kmlclusters4 == "D",]$meanPDheel.y,
                 na.rm = TRUE))
s = c("A of 4 partitions", "B of 4 partitions", "C of 4 partitions",
                       "D of 4 partitions",
      "A of 2 partitions", "B of two partitions" )
s = factor(s, level = c("A of 4 partitions", "B of 4 partitions",
                                                                    "C of 4 partitions",
                        "D of 4 partitions", "A of 2 partitions",
                                                                    "B of two partitions" ),
           ordered = TRUE)
```

```
idx = c(1, 3, 2, 4) # sorting as in publication
tab_kmlarea = array(NA, dim=c(3,6))
#df$Area = factor(df$Area)
rownames(tab_kmlarea) = c("  Partitions", "  n", " Mean (SD)")
tab_kmlarea
tab_kmlarea[1,] = s
tab_kmlarea[2,] = c(dim(df_longdf[df_longdf$kmlclusters4 == "A",])[1],
                    dim(df_longdf[df_longdf$kmlclusters4 == "B",])[1],
                    dim(df_longdf[df_longdf$kmlclusters4 == "C",])[1],
                    dim(df_longdf[df_longdf$kmlclusters4 == "D",])[1],
                    dim(df_longdf[df_longdf$kmlclusters2 == "A",])[1],
                    dim(df_longdf[df_longdf$kmlclusters2 == "B",])[1])
tab_kmlarea[1,] = c("A of 4 partitions", "B of 4 partitions",
                    "C of 4 partitions", "D of 4 partitions",
                    "A of 2 partitions", "B of 2 partitions" )
tab_kmlarea[3,] = c(l, m, n, p , q, r)
tab_kmlarea <- as.data.frame(tab_kmlarea)
t(tab_kmlarea) -> kml24
kml24
save(kml24, file = "../Data/kml24.Rda")

######## kmlcluster4 #####

##### Claws
ggplot(df_longdf) +
  geom_mosaic(aes(x = product(Claws, visit),
                  fill = Claws , conds = product(kmlclusters2))) +
  theme_minimal() +
  labs(x = "Claw length:Partitian", y = "Visit") +
  scale_fill_brewer(palette = "Set2") +
  theme(
    legend.position = "bottom",
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    plot.title = element_text(size = 12),
    axis.title = element_text(size = 10),
    axis.text.x.bottom = element_text(size = 6, angle = 90))

ggplot(df_longdf) +
  geom_mosaic(aes(x = product(Claws, visit),
                  fill = Claws , conds = product(kmlclusters4))) +
  theme_minimal() +
  labs(x = "Claw length:Partitian", y = "Visit") +
  scale_fill_brewer(palette = "Set2") +
  theme(
    legend.position = "bottom",
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    plot.title = element_text(size = 12),
    axis.title = element_text(size = 10),
    axis.text.x.bottom = element_text(size = 6, angle = 90))

##### weight

ggplot(df_longdf,
       aes(x = df_longdf$visit, y = df_longdf$Weight,
           colour = kmlclusters2, group = kmlclusters2)) + theme_gray() +
  geom_smooth(method = "loess") +
  labs(x = "Visit", y = "Weight") +
  scale_x_discrete(name = "Visit",
                                                          limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  labs(colour = "Two Partitians") +
  expand_limits(x=c(1,13), y=c(4.5, 5.5)) +
  ylab("Weight (kg)") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 10))


ggplot(df_longdf,
       aes(x = df_longdf$visit, y = df_longdf$Weight,
           colour = kmlclusters4, group = kmlclusters4)) + theme_gray() +
  geom_smooth(method = "loess") +
  labs(x = "Visit", y = "Weight (kg)") +
  scale_x_discrete(name = "Visit",
                                                          limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  labs(colour = "Four Partitians") +
  expand_limits(x=c(1,13), y=c(4.5, 6.5)) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 10))

###### age
ggplot(df_longdf, aes(x = df_longdf$visit,
                      y = df_longdf$Age,
                                                  colour = kmlclusters2, group = kmlclusters2)) +
```

```
  geom_smooth(method = "loess") +
  scale_x_discrete(name = "Visit",
                                                    limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  labs(colour = "Two partitians") +
  expand_limits(x=c(1,13), y=c(10, 22.5)) +
  ylab("Age (months)") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 10))

ggplot(df_longdf, aes(x = df_longdf$visit,
                      y = df_longdf$Age,
                                                                colour = kmlclusters4, group = kmlclusters4)) +
  geom_smooth(method = "loess") +
  scale_x_discrete(name = "Visit",
                                                    limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  labs(colour = "Two partitians") +
  expand_limits(x=c(1,13), y=c(10, 22.5)) +
  ylab("Age (months)") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 10))

###### temp
ggplot(df_longdf,
       aes(x = visit, y = Temperature,
           colour = kmlclusters2, group = kmlclusters2)) +
  geom_smooth(method = "loess") +
  labs(x = "visit", y = "Temperature") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  geom_text(aes(x = 7.5, y=18, label = "")) +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  expand_limits(x=c(1,13), y=c(10, 27)) +
  scale_x_discrete(name = "visit", limits =
                     factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  scale_colour_discrete(name = "Two partitions")

ggplot(df_longdf,
       aes(x = visit, y = Temperature,
           colour = kmlclusters4, group = kmlclusters4)) +
  geom_smooth(method = "loess") +
  labs(x = "visit", y = "Temperature") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  geom_text(aes(x = 7.5, y=18, label = "")) +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  expand_limits(x=c(1,13), y=c(10, 27)) +
  scale_x_discrete(name = "visit", limits =
                     factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  scale_colour_discrete(name = "Four partitions")


### rh
ggplot(df_longdf,
       aes(x = visit, y = RelativeHumidity,
                colour = kmlclusters2, group = kmlclusters2)) +
  geom_smooth(method = "loess") +
  theme_gray() +
  labs(x = "visit", y = "Relative humidity (%)") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  expand_limits(x=c(1,13), y=c(55, 72)) +
  scale_color_discrete(name = "Two partitians")

ggplot(df_longdf,
       aes(x = visit, y = RelativeHumidity,
                colour = kmlclusters4, group = kmlclusters4)) +
  geom_smooth(method = "loess") +
  labs(x = "visit", y = "relative humidity (%)") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom",
        legend.title = element_text(size = 10),
        legend.text = element_text(size = 8),
        plot.title = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  expand_limits(x=c(1,13), y=c(55, 70))  +
  labs(colour = "4 kml clusters") +
```

```
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13))) +
  scale_color_discrete(name = "Four partitians")
######## kmlclusters2 #####
df_longdf %>%
  group_by(visit, kmlclusters2) %>%
  filter(kmlclusters2 == "A") %>%
  tabyl(visit, Claws) %>%
  adorn_percentages() %>%
  adorn_rounding(digits = 2) %>%
  mutate(kmlclusters2 = rep("A", 13)) -> kml2clawsA
df_longdf %>%
  group_by(visit, kmlclusters2) %>%
  filter(kmlclusters2 == "B") %>%
  tabyl(visit, Claws) %>%
  adorn_percentages() %>%
  adorn_rounding(digits = 2) %>%
  mutate(kmlclusters2 = rep("B", 13)) -> kml2clawsB

bind_rows(kml2clawsA, kml2clawsB) -> df_kml2claws
save(df_kml2claws, file = "../Data/df_kml2claws.Rda")

#concordance matrix of Farm area and kml clusters
load("../Data/df_longdf.Rda")

table(df_longdf$Area, df_longdf$kmlclusters4) -> area4kml

conareakml4 <- round(prop.table(area4kml, margin = 1),
                     digits = 2) # how much do they agree with uniqe Farms
save(con_areakml4, file = "../Data/con_areakml4.Rda")

areakml2 <- table(df_longdf$Area, df_longdf$kmlclusters2)
con_areakml2 <- round(prop.table(areakml2, margin = 1), digits = 2)
con_areakml2 <- con_areakml2[,-c(3)]
save(con_areakml2, file = "../Data/con_areakml2.Rda")


areakml4 <- table(df_longdf$Area, df_longdf$kmlclusters4)
con_areakml4 <- round(prop.table(areakml4, margin = 1), digits = 2)
save(con_areakml4, file = "../Data/con_areakml2.Rda")
#
###### Generalised linear models to compare kml partitions#####
kml2diff <- summary(glm(kmlclusters2 ~ Claws + Hybrid +
                          CleanFL + MoistFL +
         meanPDheel.x, data = df_longdf, family = "binomial"))
varkml2 <- kml2diff$coefficients

kml2difftab = data.frame(
  matrix(vector(), 3, 5,
         dimnames=list(c(),
                       c("Variable", "Estimate", "Std. Error", "CI", "P value"))),
                        stringsAsFactors=F)
kml2difftab[,1] = c("Claws too long", "Hybrid : Hyla", "Mean heel scores")
kml2difftab[1,2] <- varkml2[2,1]
kml2difftab[2,2] <- varkml2[3,1]
kml2difftab[3,2] <- varkml2[9,1]
#Std error
kml2difftab[1,3] <- varkml2[2,2]
kml2difftab[2,3] <- varkml2[3,2]
kml2difftab[3,3] <- varkml2[9,2]
#CI
kml2difftab[1,4] <-  paste(
  varkml2[1,2] + 1.96*varkml2[2,2],
  varkml2[1,2] - 1.96*varkml2[2,2])
kml2difftab[2,4] <-  paste(
  varkml2[1,3] + 1.96*varkml2[3,2],
  varkml2[1,3] - 1.96*varkml2[3,2])
kml2difftab[3,4] <-  paste(
  varkml2[1,4] + 1.96*varkml2[9,2],
  varkml2[1,4] - 1.96*varkml2[9,2])
# P value
kml2difftab[1,5] <- paste("p", "0.0001", sep = "<")
kml2difftab[2,5] <- paste("p", "0.0001", sep = "<")
kml2difftab[3,5] <- paste("p", "0.0001", sep = "<")



save(kml2difftab, file = "../Data/kml2difftab.Rda")
```

```
####### simulation on loss of data #####

set.seed(1989)
obs <- dim(df)[1]
percentloss <- c(0.01, 0.05, 0.10, 0.20)
sizeofloss <- round(obs*percentloss, 0)
sizeofloss

# remove "percentloss" % of random rows of data
onepercentloss <- sample(1:obs, sizeofloss[1], replace = FALSE)
fivepercentloss <- sample(1:obs, sizeofloss[2], replace = FALSE)
tenpercentloss <- sample(1:obs, sizeofloss[3], replace = FALSE)
twentypercentloss <- sample(1:obs, sizeofloss[4], replace = FALSE)
#class(onepercentloss)
```

```
# remove 1% of random rows
df_onepercentloss <- df[-onepercentloss,]
dim(df_onepercentloss) #2586 x 44, 1% of data loss

# remove 5% of random rows
df_fivepercentloss <- df[-fivepercentloss,]
#dim(df_fivepercentloss) #2481 x 44, 5% of data loss

# remove 10% of random rows
df_tenpercentloss <- df[-tenpercentloss,]
#dim(df_tenpercentloss) #2351 x 44, 10% of data loss

# remove 20% of random rows
df_twentypercentloss <- df[-twentypercentloss,]
#dim(df_twentypercentloss) #2091 x 44, 20% of data loss

####### Random losses of data (as stipulated above) #####

####### one percent loss kml and sample size #####
# first create wide form
df_onepercentloss0 <- df_onepercentloss[,c(3, 37,28, 41, 38)]
#sum(is.na(df_onepercentloss0)) # 12
#dim(df_onepercentloss0) #2586 x 5
df_onepercentloss0$visit <-
  factor(df_onepercentloss0$visit, levels = c("1", "2","3", "4","5", "6",
                                              "7", "8","9", "10",
                                              "11", "12","13"),
                                   order = TRUE)

df_onepercentloss_wide1 <-spread(df_onepercentloss0, visit, meanPDheel)
onepercentloss_wide2 <- as.matrix(df_onepercentloss_wide1[,4:16])
#sum(is.na(onepercentloss_wide2)) # is 1898
#sum(is.nan(onepercentloss_wide2)) # is 12
is.nan(onepercentloss_wide2) -> "NA"
opkml <- imputation(as.matrix(
       df_onepercentloss_wide1[,4:16]), "trajMean")
#sum(is.na(opkml)) #0
save(opkml, file = "../Data/opkml.Rda")
opcld <- cld(traj = opkml, timeInData = 1:13)
save(opcld, file = "../Data/opcld.Rda")

# implement kml on 1 % loss
load("../Data/opcld.Rda")
#kml(opcld, 4, toPlot = "both") # slow kml
kml(opcld, nbClusters = 4, parAlgo = parALGO(distance = function(x, y)
  +    cor(x, y), saveFreq = 10)) # fast kml
opkml <- data.frame(opkml)
opkml$kmlclusters4op <- getClusters(opcld, 4, asInteger = FALSE)
#likes matrix class
#View(opkml)

op <- gather(opkml, visit, score, -kmlclusters4op)
1 -> op$visit[op$visit == "X1"]
2 -> op$visit[op$visit == "X2"]
3 -> op$visit[op$visit == "X3"]
4 -> op$visit[op$visit == "X4"]
5 -> op$visit[op$visit == "X5"]
6 -> op$visit[op$visit == "X6"]
7 -> op$visit[op$visit == "X7"]
8 -> op$visit[op$visit == "X8"]
9 -> op$visit[op$visit == "X9"]
10 -> op$visit[op$visit == "X10"]
11 -> op$visit[op$visit == "X11"]
12 -> op$visit[op$visit == "X12"]
13 -> op$visit[op$visit == "X13"]
op$visit <- factor(op$visit,
                       levels = c("1", "2", "3", "4", "5", "6",
                       "7", "8", "9", "10","11", "12","13"),
                       ordered = TRUE)
# save op df and create plot (for full data go to kml4score in ch03)
save(op, file = "../Data/op.Rda")

# sample size
# for zero percent loss see "sample size per cluster" in this file

op %>%
  tabyl(kmlclusters4op) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> opkml4counts
colnames(opkml4counts) <- c("kml clusters", "n", "(%)")
save(opkml4counts, file = "../Data/opkml4counts.Rda")

a = paste( "A", " (", opkml4counts$n[1], ")", sep = "")
b = paste( "B", " (", opkml4counts$n[2], ")", sep = "")
c = paste( "C", " (", opkml4counts$n[3], ")", sep = "")
d = paste( "D", " (", opkml4counts$n[4], ")", sep = "")
ggplot(op, aes(x = visit, y = score,
                                                    group = kmlclusters4op, colour = kmlclusters4op)) +
  geom_smooth(method = "loess") +
  scale_colour_discrete(name = "% loss (n)", breaks = c("A", "B", "C", "D"),
                        labels = c(a, b, c, d)) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted")  +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))
```

```
######### five percent loss kml #####
# first create wide form
df_fivepercentloss0 <- df_fivepercentloss[,c(3, 37,28, 41, 38)]
#df_fivepercentloss0 <- df_fivepercentloss[,c(3,40, 37,39, 41)]
#sum(is.na(df_fivepercentloss0)) #13
#dim(df_fivepercentloss0) #2481 x 5
df_fivepercentloss0$visit <- factor(df_fivepercentloss0$visit,
                                                                    levels = c("1", "2","3",
                                                                    "4","5", "6", "7", "8", "9", "10",
                                                                    "11", "12","13"), ordered = TRUE)


df_fivepercentloss_wide1 <-spread(df_fivepercentloss0, visit, meanPDheel)
fivepercentloss_wide2 <- as.matrix(df_fivepercentloss_wide1[,4:16])
#sum(is.na(fivepercentloss_wide2)) # is 2016
#sum(is.nan(fivepercentloss_wide2)) #12
fpkml <- imputation(as.matrix(df_fivepercentloss_wide1[,4:16]), "trajMean")
#fpkml <- fpkml[-61, ]
# fpkml <- data.frame(fpkml) # at fpcld stage, idALL duplicated apparently
sum(is.na(fpkml)) #0
fpcld <- clusterLongData(traj = fpkml, timeInData = 1:13)
save(fpcld, file = "../Data/fpcld.Rda")
#kml(fpcld, 4, toPlot = "both") # slow kml
kml(fpcld, nbClusters = 4, parAlgo = parALGO(distance = function(x, y)
  +    cor(x, y), saveFreq = 10)) # fast kml
#onepercentloss_wide3 <- as.data.frame(onepercentloss_wide2[-61,])
dim(fpkml)
fpkml <- data.frame(fpkml)
fpkml$kmlclusters4fp <- getClusters(fpcld, nbCluster = 4,
                        asInteger = FALSE)

fp <- gather(fpkml, visit, score, -kmlclusters4fp)
1 -> fp$visit[fp$visit == "X1"]
2 -> fp$visit[fp$visit == "X2"]
3 -> fp$visit[fp$visit == "X3"]
4 -> fp$visit[fp$visit == "X4"]
5 -> fp$visit[fp$visit == "X5"]
6 -> fp$visit[fp$visit == "X6"]
7 -> fp$visit[fp$visit == "X7"]
8 -> fp$visit[fp$visit == "X8"]
9 -> fp$visit[fp$visit == "X9"]
10 -> fp$visit[fp$visit == "X10"]
11 -> fp$visit[fp$visit == "X11"]
12 -> fp$visit[fp$visit == "X12"]
13 -> fp$visit[fp$visit == "X13"]
fp$visit <- factor(fp$visit, levels = c("1", "2", "3", "4",
                                                                    "5", "6", "7", "8",
    "9", "10","11", "12","13"), order = TRUE)
# save df
save(fp, file = "../Data/fp.Rda")

# sample size
fp %>%
  tabyl(kmlclusters4fp) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> fpkml4counts
colnames(fpkml4counts) <- c("kml clusters", "n", "(%)")
save(fpkml4counts, file = "../Data/fpkml4counts.Rda")

# plot
a = paste( "A", " (", fpkml4counts$n[1], ")", sep = "")
b = paste( "B", " (", fpkml4counts$n[2], ")", sep = "")
c = paste( "C", " (", fpkml4counts$n[3], ")", sep = "")
d = paste( "D", " (", fpkml4counts$n[4], ")", sep = "")

ggplot(fp, aes(x = visit, y = score,
               group = kmlclusters4fp, colour = kmlclusters4fp)) +
  geom_smooth(method = "loess") +
  scale_colour_discrete(name = "% loss (n)", breaks = c("A", "B", "C", "D"),
                        labels=c(a, b, c, d)) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######## ten percent loss kml #####
# ten create wide form
#scene 1 use visit as factor for ten percent loss
df_tenpercentloss0 <- df_tenpercentloss[,c(3, 37,28, 41, 38)]
#sum(is.na(df_tenpercentloss0)) #9
#dim(df_tenpercentloss0) #2351 x 5
df_tenpercentloss0$visit <-
  factor(df_tenpercentloss0$visit, levels = c("1", "2","3",
                                    "4", "5", "6", "7", "8","9", "10",
                                      "11", "12","13"),
                                    order = TRUE)

df_tenpercentloss_wide1 <-spread(df_tenpercentloss0, visit, meanPDheel)
tenpercentloss_wide2 <- as.matrix(df_tenpercentloss_wide1[,4:16])
#sum(is.na(tenpercentloss_wide2)) # is 2104
#sum(is.nan(tenpercentloss_wide2)) # is 9
tpkml <- imputation(as.matrix(df_tenpercentloss_wide1[,4:16]), "trajMean")
#tpkml <- tpkml[-61, ]
#tpkml <- data.frame(tpkml) # at fpcld stage, idALL duplicated apparently
#sum(is.na(tpkml)) #0
class(tpkml)
```

```
tpcld <- clusterLongData(traj = tpkml, timeInData = 1:13)
save(tpcld, file = "../Data/tpcld.Rda")
#kml(tpcld, 4, toPlot = "both") # slow kml
kml(tpcld, nbClusters = 4, parAlgo = parALGO(distance = function(x, y)
  +    cor(x, y), saveFreq = 10)) # fast kml
#onepercentloss_wide3 <- as.data.frame(onepercentloss_wide2[-61,])
tpkml <- data.frame(tpkml)
tpkml$kmlclusters4tp <- getClusters(tpcld, nbCluster = 4,
                                                                    asInteger = FALSE)
# likes data.frame class
#View(tpkml)

tp <- gather(tpkml, visit, score, -kmlclusters4tp)
1 -> tp$visit[tp$visit == "X1"]
2 -> tp$visit[tp$visit == "X2"]
3 -> tp$visit[tp$visit == "X3"]
4 -> tp$visit[tp$visit == "X4"]
5 -> tp$visit[tp$visit == "X5"]
6 -> tp$visit[tp$visit == "X6"]
7 -> tp$visit[tp$visit == "X7"]
8 -> tp$visit[tp$visit == "X8"]
9 -> tp$visit[tp$visit == "X9"]
10 -> tp$visit[tp$visit == "X10"]
11 -> tp$visit[tp$visit == "X11"]
12 -> tp$visit[tp$visit == "X12"]
13 -> tp$visit[tp$visit == "X13"]
tp$visit <- factor(tp$visit, levels = c("1", "2", "3",
                "4", "5", "6", "7", "8",
   "9", "10","11", "12","13"),
                                                                    ordered = TRUE)
# save df
save(tp, file = "../Data/tp.Rda")

# sample size
tp %>%
  tabyl(kmlclusters4tp) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> tpkml4counts
colnames(tpkml4counts) <- c("kml clusters", "n", "(%)")
save(tpkml4counts, file = "../Data/tpkml4counts.Rda")

# plot
a = paste( "A", " (", tpkml4counts$n[1], ")", sep = "")
b = paste( "B", " (", tpkml4counts$n[2], ")", sep = "")
c = paste( "C", " (", tpkml4counts$n[3], ")", sep = "")
d = paste( "D", " (", tpkml4counts$n[4], ")", sep = "")

ggplot(tp, aes(x = visit, y = score,
               group = kmlclusters4tp, colour = kmlclusters4tp)) +
  geom_smooth(method = "loess") +
  scale_colour_discrete(name = "% loss", breaks = c("A", "B", "C", "D"),
                        labels=c(a, b, c, d)) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######## twenty percent loss kml #####
# twenty create wide form
# df_twentypercentloss0 <- df_twentypercentloss[,c(3, 37,28, 41, 38)]
# #df_fivepercentloss0 <- df_fivepercentloss[,c(3,40, 37,39, 41)]
#sum(is.na(df_twentypercentloss0)) #8
#dim(df_twentypercentloss0) #2091 x 5
df_twentypercentloss0$visit <-
  factor(df_twentypercentloss0$visit,
         levels = c("1", "2", "3", "4", "5", "6", "7", "8","9", "10",
                    "11", "12","13"), ordered = TRUE)

df_twentypercentloss_wide1 <-spread(
        df_twentypercentloss0, visit, meanPDheel)
twentypercentloss_wide2 <- as.matrix(df_twentypercentloss_wide1[,4:16])
#sum(is.na(twentypercentloss_wide2)) # is 2311
#sum(is.nan(twentypercentloss_wide2)) # is 8
twentypkml <- imputation(as.matrix(df_twentypercentloss_wide1[,4:16]),
                                                                                    "trajMean")

#View(twentypkml)
#sum(is.na(twentypkml)) #0
twentypcld <- cld(traj = twentypkml, timeInData = 1:13)
save(twentypcld, file = "../Data/tpcld.Rda")
#kml(twentypcld, 4, toPlot = "both") # slow kml
kml(twentypcld, 4, parAlgo = parALGO(distance = function(x, y) #fast kml
  +    cor(x, y), saveFreq = 10)) # fast kml
twentypkml <- data.frame(twentypkml)
twentypkml$kmlclusters4twentyp <- getClusters(
        twentypcld, nbCluster = 4, asInteger = FALSE)
#View(twentypkml)
#twentypkml <- data.frame(twentypkml)
twentyp <- gather(twentypkml, visit, score, -kmlclusters4twentyp)
1 -> twentyp$visit[twentyp$visit == "X1"]
2 -> twentyp$visit[twentyp$visit == "X2"]
3 -> twentyp$visit[twentyp$visit == "X3"]
4 -> twentyp$visit[twentyp$visit == "X4"]
5 -> twentyp$visit[twentyp$visit == "X5"]
6 -> twentyp$visit[twentyp$visit == "X6"]
7 -> twentyp$visit[twentyp$visit == "X7"]
```

```
8 -> twentyp$visit[twentyp$visit == "X8"]
9 -> twentyp$visit[twentyp$visit == "X9"]
10 -> twentyp$visit[twentyp$visit == "X10"]
11 -> twentyp$visit[twentyp$visit == "X11"]
12 -> twentyp$visit[twentyp$visit == "X12"]
13 -> twentyp$visit[twentyp$visit == "X13"]
twentyp$visit <- factor(twentyp$visit,
                levels = c("1", "2", "3", "4", "5", "6", "7",
                "8", "9", "10","11", "12","13"),
                            ordered = TRUE)
twentyp$score <- as.numeric(twentyp$score)
# save df
save(twentyp, file = "../Data/twentyp.Rda")

# sample size
twentyp %>%
  tabyl(kmlclusters4twentyp) %>%
  adorn_rounding(digits = 2) %>%
  as.data.frame() -> twentypkml4counts
colnames(twentypkml4counts) <- c("kml clusters", "n", "(%)")
save(twentypkml4counts, file = "../Data/twentypkml4counts.Rda")

# plot
a = paste( "A", " (", twentypkml4counts$n[1], ")", sep = "")
b = paste( "B", " (", twentypkml4counts$n[2], ")", sep = "")
c = paste( "C", " (", twentypkml4counts$n[3], ")", sep = "")
d = paste( "D", " (", twentypkml4counts$n[4], ")", sep = "")
ggplot(twentyp, aes(x = visit, y = score,
                    group = kmlclusters4twentyp,
                                                                    colour = kmlclusters4twentyp)) +
  geom_smooth(method = "loess") +
  scale_colour_discrete(name = "% loss", breaks = c("A", "B", "C", "D"),
                        labels=c(a, b, c, d)) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))
######## Means of scores per percentloss data set to generate mse

# table of means for lm model
df_longdf %>%
  select(kmlclusters4, visit, meanPDheel.y) %>%
  group_by_at(vars(kmlclusters4, visit)) %>%
  summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
  mutate_if(is.numeric, round, digits=1) %>%
  as.data.frame() -> zerop_means
colnames(zerop_means)[3] <-  "zp_m"
colnames(zerop_means)[4] <-  "zp_sd"

op %>%
  group_by_at(vars(kmlclusters4op, visit)) %>%
  summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
  mutate_if(is.numeric, round, digits=1) %>%
  as.data.frame() -> op_means
colnames(op_means)[3] <-  "op_m"
colnames(op_means)[4] <-  "op_sd"

fp %>%
  group_by_at(vars(kmlclusters4fp, visit)) %>%
  summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
  mutate_if(is.numeric, round, digits=1) %>%
  as.data.frame() -> fp_means
colnames(fp_means)[3] <-  "fp_m"
colnames(fp_means)[4] <-  "fp_sd"

tp %>%
  group_by_at(vars(kmlclusters4tp, visit)) %>%
  summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
  mutate_if(is.numeric, round, digits=1) %>%
  as.data.frame() -> tp_means
colnames(tp_means)[3] <-  "tp_m"
colnames(tp_means)[4] <-  "tp_sd"

twentyp %>%
  group_by_at(vars(kmlclusters4twentyp, visit)) %>%
  summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
  mutate_if(is.numeric, round, digits=1) %>%
  as.data.frame() -> twentyp_means
colnames(twentyp_means)[3] <-  "twentyp_m"
colnames(twentyp_means)[4] <-  "twentyp_sd"

######## lm models and their residuals for loss of data #####
zeropmod = lm(zp_m ~ visit + kmlclusters4, data = zerop_means)
plot(zeropmod)
mean(zeropmod$residuals^2) # [1]  0.02424556

fpmod = lm(fp_m ~ visit + kmlclusters4fp, data = fp_means)
plot(fpmod)
mean(fpmod$residuals^2) # [1] 0.005465976

tpmod = lm(tp_m ~ visit + kmlclusters4tp, data = tp_means)
plot(tpmod)
mean(tpmod$residuals^2) # [1] 0.004252959

twentypmod = lm(twentyp_m ~ visit +
```

```
                                                                    kmlclusters4twentyp, data = twentyp_means)
plot(tpmod)
mean(tpmod$residuals^2) # [1] 0.004252959

######### Score trajectory for percent loss and no loss #####

load("../Data/op.Rda")
load("../Data/fp.Rda")
load("../Data/tp.Rda")
load("../Data/twentyp.Rda")

op_long <- gather(op, kmlclusters4op, value, -score, -visit)
fp_long <- gather(fp, kmlclusters4fp, value, -score, -visit)
tp_long <- gather(tp, kmlclusters4tp, value, -score, -visit)
twentyp_long <- gather(twentyp, kmlclusters4twentyp, value, -score, -visit)

# reshape data
# for data frame op

op_long$loss <- rep("one percent loss", dim(op_long)[1])
op_long <- op_long[, -3]
head(op_long)

# for data frame fp
fp_long$loss <- rep("five percent loss", dim(fp_long)[1])
fp_long <- fp_long[, -3]
head(fp_long)

# for data frame tp
tp_long$loss <- rep("ten percent loss", dim(tp_long)[1])
tp_long <- tp_long[, -3]
head(tp_long)

# for data frame twentyp
twentyp_long$loss <- rep("twenty percent loss", dim(twentyp_long)[1])
twentyp_long <- twentyp_long[, -3]
head(twentyp_long)

#for data frame no loss

no_loss <- df_longdf[,c(28, 46, 45)]
no_loss$loss <- rep("zero percent loss", dim(df_longdf)[1])
colnames(no_loss) <- c("visit", "score", "value", "loss")

loss_long <- rbind(op_long, fp_long, tp_long, twentyp_long, no_loss)
loss_long$loss <-
        factor(loss_long$loss, c("zero percent loss", "one percent loss",
        "five percent loss",
"ten percent loss", "twenty percent loss"),
                                ordered = TRUE)
save(loss_long, file = "../Data/loss_long.Rda")

######## Score in Loss plots #####
a = paste( "0 %", " (", tab_loss$A[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss$A[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss$A[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss$A[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss$A[5], ")", sep = "")
# loss plot for kml cluster A
ggplot(loss_long[loss_long$value == "A",],
                        aes(x = visit, y = as.numeric(score), group = loss,
                      colour = loss)) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  scale_colour_discrete(name = "% loss (n)",
                        labels = c(a,b,c,d, e)) +
  labs(y = "score") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.7, 4)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

# loss plot for kml cluster B
a = paste( "0 %", " (", tab_loss$B[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss$B[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss$B[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss$B[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss$B[5], ")", sep = "")
ggplot(loss_long[loss_long$value == "B",],
                        aes(x = visit, y = as.numeric(score), group = loss,
                                            colour = loss)) +
  geom_smooth(method = "loess", se = FALSE) +
  scale_colour_discrete(name = "% loss (n)",
                        labels = c(a,b,c,d,e)) +
  labs(y = "score") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  expand_limits(x=c(1,13), y=c(3.7, 4)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

# loss plot for kml cluster C
a = paste( "0 %", " (", tab_loss$C[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss$C[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss$C[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss$C[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss$C[5], ")", sep = "")
```

```
ggplot(loss_long[loss_long$value == "C",],
                        aes(x = visit, y = as.numeric(score), group = loss,
                                              colour = loss)) +
  geom_smooth(method = "loess", se = FALSE) +
  scale_colour_discrete(name = "% loss (n)",
                        labels = c(a,b,c,d,e)) +
  labs(y = "score") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

# loss plot for kml cluster D
a = paste( "0 %", " (", tab_loss$D[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss$D[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss$D[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss$D[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss$D[5], ")", sep = "")
ggplot(loss_long[loss_long$value == "D",],
                        aes(x = visit, y = as.numeric(score), group = loss,
                                              colour = loss)) +
  geom_smooth(method = "loess", se = FALSE) +
  scale_colour_discrete(name = "% loss (n)",
                        labels = c(a,b,c,d,e)) +
  labs(y = "score") +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

### end of loss plots and loss ###
```

```
####### simulation on loss of data #####

set.seed(1964) # Michelle Obama's birth year
obs <- dim(df_longdf)[1]
percentloss <- c(0.01, 0.05, 0.10, 0.20)
sizeofloss <- round(obs*percentloss, 0)
sizeofloss

# remove "percentloss" % of random rows of data
onepercentloss <- sample(1:obs, sizeofloss[1], replace = FALSE)
fivepercentloss <- sample(1:obs, sizeofloss[2], replace = FALSE)
tenpercentloss <- sample(1:obs, sizeofloss[3], replace = FALSE)
twentypercentloss <- sample(1:obs, sizeofloss[4], replace = FALSE)
#class(onepercentloss)

# remove 1% of random rows
df_onepercentloss0 <- df_longdf[-onepercentloss,]
#dim(df_onepercentloss0) #4446 x 47, 1% of data loss

# remove 5% of random rows
df_fivepercentloss0 <- df_longdf[-fivepercentloss,]
#dim(df_fivepercentloss0) #4341 x 47, 5% of data loss

# remove 10% of random rows
df_tenpercentloss0 <- df_longdf[-tenpercentloss,]
#dim(df_tenpercentloss0) #4211 x 47, 10% of data loss

# remove 20% of random rows
df_twentypercentloss0 <- df_longdf[-twentypercentloss,]
#dim(df_twentypercentloss0) #3950 x 47, 20% of data loss

save(df_onepercentloss0, file = "../Data/df_onepercentloss0.Rda")
save(df_onepercentloss0, file = "../Data/df_fivepercentloss0.Rda")
save(df_onepercentloss0, file = "../Data/df_tenpercentloss0.Rda")
save(df_onepercentloss0, file = "../Data/df_twentypercentloss0.Rda")


####### Random losses of data (as stipulated above) #####

df_onepercentloss0 %>%
tabyl(kmlclusters4) %>%
adorn_rounding(digits = 2) %>%
as.data.frame() %>%
adorn_totals()-> opkml4counts0
colnames(opkml4counts0) <- c("kml clusters", "n", "(%)")
save(opkml4counts0, file = "../Data/opkml4counts0.Rda")

a = paste( "A", " (", opkml4counts0$n[1], ")", sep = "")
b = paste( "B", " (", opkml4counts0$n[2], ")", sep = "")
c = paste( "C", " (", opkml4counts0$n[3], ")", sep = "")
d = paste( "D", " (", opkml4counts0$n[4], ")", sep = "")
ggplot(df_onepercentloss0, aes(x = visit, y = meanPDheel.y,
                               group = kmlclusters4, colour = kmlclusters4)) +
geom_smooth(method = "loess") +
scale_colour_discrete(name = "% loss" (n), breaks = c("A", "B", "C", "D"),
labels = c(a, b, c, d)) +
geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted")  +
labs( y = "score") +
theme(legend.position = "bottom") +
expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
```

```
scale_x_discrete(name = "visit",
 limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######### five percent loss kml #####

# sample size
df_fivepercentloss0 %>%
tabyl(kmlclusters4) %>%
adorn_rounding(digits = 2) %>%
as.data.frame() %>%
adorn_totals() -> fpkml4counts0
colnames(fpkml4counts0) <- c("kml clusters", "n", "(%)")
save(fpkml4counts0, file = "../Data/fpkml4counts0.Rda")

# plot
a = paste( "A", " (", fpkml4counts0$n[1], ")", sep = "")
b = paste( "B", " (", fpkml4counts0$n[2], ")", sep = "")
c = paste( "C", " (", fpkml4counts0$n[3], ")", sep = "")
d = paste( "D", " (", fpkml4counts0$n[4], ")", sep = "")

ggplot(df_fivepercentloss0, aes(x = visit, y = meanPDheel.y,
                                group = kmlclusters4, colour = kmlclusters4)) +
geom_smooth(method = "loess") +
scale_colour_discrete(name = "Four partitions (n)",
breaks = c("A", "B", "C", "D"),
labels=c(a, b, c, d)) +
  labs( y = "Score") +
geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
theme(legend.position = "bottom",
      axis.title = element_text(size = 12),
      legend.title = element_text(size = 8),
      legend.text = element_text(size = 8)) +
expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
scale_x_discrete(name = "Visit",
 limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######## ten percent loss kml #####

# sample size
df_tenpercentloss0 %>%
tabyl(kmlclusters4) %>%
adorn_rounding(digits = 2) %>%
as.data.frame() %>%
adorn_totals()-> tpkml4counts0
colnames(tpkml4counts0) <- c("kml clusters", "n", "(%)")
save(tpkml4counts0, file = "../Data/tpkml4counts0.Rda")

# plot
a = paste( "A", " (", tpkml4counts0$n[1], ")", sep = "")
b = paste( "B", " (", tpkml4counts0$n[2], ")", sep = "")
c = paste( "C", " (", tpkml4counts0$n[3], ")", sep = "")
d = paste( "D", " (", tpkml4counts0$n[4], ")", sep = "")
ggplot(df_tenpercentloss0, aes(x = visit, y = meanPDheel.y,

                               group = kmlclusters4, colour = kmlclusters4)) +
geom_smooth(method = "loess") +
scale_colour_discrete(name = "Four partitions (n)",
breaks = c("A", "B", "C", "D"),
labels=c(a, b, c, d)) +
  labs(y = "Score") +
geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
theme(legend.position = "bottom",
      axis.title = element_text(size = 12),
      legend.title = element_text(size = 8),
      legend.text = element_text(size = 8)) +
expand_limits(x=c(1,13), y=c(3.5, 5.5)) +
scale_x_discrete(name = "Visit",
 limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######## twenty percent loss kml #####

# sample size
df_twentypercentloss0 %>%
tabyl(kmlclusters4) %>%
adorn_rounding(digits = 2) %>%
as.data.frame() -> twentypkml4counts0
colnames(twentypkml4counts0) <- c("kml clusters", "n", "(%)")
save(twentypkml4counts0, file = "../Data/twentypkml4counts0.Rda")

# plot
a = paste( "A", " (", twentypkml4counts0$n[1], ")", sep = "")
b = paste( "B", " (", twentypkml4counts0$n[2], ")", sep = "")
c = paste( "C", " (", twentypkml4counts0$n[3], ")", sep = "")
d = paste( "D", " (", twentypkml4counts0$n[4], ")", sep = "")
ggplot(df_twentypercentloss0, aes(x = visit, y = meanPDheel.y,
                                  group = kmlclusters4, colour = kmlclusters4)) +
geom_smooth(method = "loess") +
scale_colour_discrete(name = "Four partitions (n)",
breaks = c("A", "B", "C", "D"),
labels=c(a, b, c, d)) +
geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
theme(legend.position = "bottom",
      axis.title = element_text(size = 12),
      legend.title = element_text(size = 8),
      legend.text = element_text(size = 8)) +
  labs(y = "Score") +
```

```
expand_limits(x=c(1,13), y=c(3.5, 5)) +
scale_x_discrete(name = "Visit",
 limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

######## Means of scores per percentloss data set

# table of means for lm model
df_longdf %>%
select(kmlclusters4, visit, meanPDheel.y) %>%
group_by_at(vars(kmlclusters4, visit)) %>%
summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
mutate_if(is.numeric, round, digits=1) %>%
as.data.frame() -> zerop_means0
colnames(zerop_means0)[3] <-  "zp_m"
colnames(zerop_means0)[4] <-  "zp_sd"

df_onepercentloss0 %>%
group_by_at(vars(kmlclusters4, visit)) %>%
summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
mutate_if(is.numeric, round, digits=1) %>%
as.data.frame() -> op_means0
colnames(op_means0)[17] <-  "op_m"
colnames(op_means0)[37] <-  "op_sd"

df_fivepercentloss0 %>%
group_by_at(vars(kmlclusters4, visit)) %>%
summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
mutate_if(is.numeric, round, digits=1) %>%
as.data.frame() -> fp_means0
colnames(fp_means0)[17] <-  "fp_m"
colnames(fp_means0)[37] <-  "fp_sd"

df_tenpercentloss0 %>%
group_by_at(vars(kmlclusters4, visit)) %>%
summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
mutate_if(is.numeric, round, digits=1) %>%
as.data.frame() -> tp_means0
colnames(tp_means0)[17] <-  "tp_m"
colnames(tp_means0)[37] <-  "tp_sd"

df_twentypercentloss0 %>%
group_by_at(vars(kmlclusters4, visit)) %>%
summarise_if(is.numeric, c(mean, sd), na.rm=TRUE) %>%
mutate_if(is.numeric, round, digits=1) %>%
as.data.frame() -> twentyp_means0
colnames(twentyp_means0)[3] <-  "twentyp_m"
colnames(twentyp_means0)[4] <-  "twentyp_sd"


######## lm models and their residuals for loss of data #####
zeropmod0 = lm(zp_m ~ visit + kmlclusters4, data = zerop_means)
plot(zeropmod0)
mean(zeropmod0$residuals^2) # [1]  0.02424556

fpmod0 = lm(fp_m ~ visit + kmlclusters4, data = fp_means0)
plot(fpmod0)
mean(fpmod$residuals0^2) # [1] 0.005465976 (old) vs 0.005465976

tpmod0 = lm(tp_m ~ visit + kmlclusters4, data = tp_means0)
plot(tpmod0)
mean(tpmod$residuals^2) # [1] 0.004252959 vs 28.50422

twentypmod0 = lm(twentyp_m ~ visit + kmlclusters4, data = twentyp_means0)
plot(tpmod0)
mean(tpmod$residuals^2) # [1] 0.004252959 (old) vs 28.50422

####### zeroer 0% loss #####


full_join(df_longdf, zerop_means0,
          by = c("visit" = "visit",
                 "kmlclusters4" = "kmlclusters4"), copy = FALSE) -> zeroer0
zeroer0$mse_zp0 <- ((zeroer0$meanPDheel.x - zeroer0$zp_m)^2) / dim(zeroer)[1]
zeroer0$variable0 <- rep("0 %", dim(zeroer)[1])
# gather zeroer
names(zeroer0)
zeroer0 <- zeroer0[c(46, 28, 47, 48, 49, 50, 51)]
names(zeroer0)
#zero0 <- zero0[,c( 46, 28, 38, 62, 82, 88, 89)]

######## loner 1 % loss #####
full_join(df_onepercentloss0, op_means0,
          by = c("visit" = "visit",
            "kmlclusters4" = "kmlclusters4"), copy = FALSE) -> loner0
dim(loner0)
loner0$mse_op0 <- ((loner0$meanPDheel.y - loner0$op_m)^2) / dim(loner0)[1]
loner0$variable0 <- rep("1 %", dim(loner0)[1])
loner0 <- loner0[,c( 46, 28, 38, 62, 82, 88, 89)]

######### fiver 5 % loss #####
full_join(df_fivepercentloss0, fp_means0,
          by = c("visit" = "visit",
                 "kmlclusters4" = "kmlclusters4"), copy = FALSE) -> fiver0
dim(fiver0)
fiver0$mse_fp0 <- ((fiver0$meanPDheel.y - fiver0$fp_m)^2) / dim(fiver0)[1]
#View(fiver0)
```

```
fiver0$variable0 <- rep("5 %", dim(fiver0)[1])
names(fiver0)
fiver0 <- fiver0[,c( 46, 28, 38, 62, 82, 88, 89)]

######### tenner 10 % loss#####
full_join(df_tenpercentloss0, tp_means0,
          by = c("visit" = "visit",
                 "kmlclusters4" = "kmlclusters4"), copy = FALSE) -> tenner0
dim(tenner0)
tenner0$mse_tp0 <- ((tenner0$meanPDheel.x - tenner0$tp_m)^2) / dim(tenner0)[1]
tenner0$variable0 <- rep("10 %", dim(tenner0)[1])
#View(tenner)
names(tenner0)
tenner0 <- tenner0[,c( 46, 28, 38, 62, 82, 88, 89)]
names(tenner0)
#c("kml clusters", "visit", "score",
"mean per loss", "sd per loss", "mse per loss", "variable")

######### twentier 20 % loss #####

full_join(df_twentypercentloss0, twentyp_means0,
          by = c("visit" = "visit",
                 "kmlclusters4" = "kmlclusters4"), copy = FALSE) -> twentier0
dim(twentier0)
twentier0$mse_twenty0 <-
  ((twentier0$meanPDheel.y - twentier0$twentyp_m)^2) / dim(twentier0)[1]
twentier0$variable0 <- rep("20 %", dim(twentier0)[1])
#View(twentier0)
names(twentier0)
twentier0 <- twentier0[,c( 46, 28, 38, 48, 49, 88, 89)]

######## rbind

colnames(zeroer0) <- c("kml clusters",
                       "visit", "score", "mean per loss",
 "sd per loss", "mse per loss", "variable")

colnames(loner0) <- c("kml clusters",
                      "visit", "score", "mean per loss",
"sd per loss", "mse per loss", "variable")

colnames(fiver0) <- c("kml clusters",
                      "visit", "score", "mean per loss",
"sd per loss", "mse per loss", "variable")

colnames(tenner0) <- c("kml clusters",
                       "visit", "score", "mean per loss",
 "sd per loss", "mse per loss", "variable")

colnames(twentier0) <- c("kml clusters",
                         "visit", "score", "mean per loss",
 "sd per loss", "mse per loss", "variable")

names(zeroer0)
dim(zeroer0)
names(loner0)
dim(loner0)
names(fiver0)
names(tenner0)
names(twentier0)

er_long0 <- data.frame(rbind(loner0, zeroer0, fiver0, tenner0, twentier0))

#er_long0 <- na.omit(er_long0)
save(er_long0, file = "../Data/er_long0.Rda")

er_long0 %>%
  tabyl(variable, kml.clusters) %>%
  as.data.frame() %>%
  adorn_totals(where = "col")-> tab_loss0
tab_loss0
tab_loss0 <- tab_loss0[-6, ]
save(tab_loss0, file = "../Data/tab_loss0.Rda")

######### Score trajectory for percent loss and no loss #####

load("../Data/op.Rda")
load("../Data/fp.Rda")
load("../Data/tp.Rda")
load("../Data/twentyp.Rda")

op_long <- gather(op, kmlclusters4op, value, -score, -visit)
fp_long <- gather(fp, kmlclusters4fp, value, -score, -visit)
tp_long <- gather(tp, kmlclusters4tp, value, -score, -visit)
twentyp_long <- gather(twentyp, kmlclusters4twentyp, value, -score, -visit)

# reshape data
# for data frame op

df_onepercentloss0$loss <- rep("1 % loss", dim(df_onepercentloss0)[1])
#head(df_onepercentloss0)

# for data frame fp
df_fivepercentloss0$loss <- rep("5 % loss", dim(df_fivepercentloss0)[1])
#head(df_fivepercentloss0)
```

```
# for data frame tp
df_tenpercentloss0$loss <- rep("10 % loss", dim(df_tenpercentloss0)[1])
#head(df_tenpercentloss0)

# for data frame twentyp
df_twentypercentloss0$loss <- rep("20 % loss", dim(df_twentypercentloss0)[1])
#head(df_twentypercentloss0)


# long form
loss_long0 <- data.frame(rbind(df_longdf, df_onepercentloss0, df_fivepercentloss0,
                               df_tenpercentloss0, df_twentypercentloss0))


loss_long0$loss <- factor(loss_long0$loss, c("0 % loss", "1 % loss",
 "5 % loss", "10 % loss",
 "20 % loss"), order = TRUE)

save(loss_long0, file = "../Data/loss_long0.Rda")

######## Score in Loss plots #####

# loss plot for kml cluster A
a = paste( "0 %", " (", tab_loss0$A[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss0$A[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss0$A[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss0$A[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss0$A[5], ")", sep = "")
ggplot(er_long0[er_long0$kml.clusters == "A",],
       aes(x = visit, y = score, group = variable, colour = variable)) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  scale_colour_discrete(name = "% loss (n)", labels = c(a,b,c,d, e)) +
  theme(legend.position = "bottom")

# loss plot for kml cluster B
a = paste( "0 %", " (", tab_loss0$B[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss0$B[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss0$B[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss0$B[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss0$B[5], ")", sep = "")
ggplot(er_long0[er_long0$kml.clusters == "B",],
       aes(x = visit, y = score, group = variable, colour = variable)) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  scale_colour_discrete(name = "% loss (n)", labels = c(a,b,c,d, e)) +
  theme(legend.position = "bottom")


# loss plot for kml cluster C
a = paste( "0 %", " (", tab_loss0$C[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss0$C[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss0$C[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss0$C[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss0$C[5], ")", sep = "")
ggplot(er_long0[er_long0$kml.clusters == "C",],
       aes(x = visit, y = score, group = variable, colour = variable)) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  scale_colour_discrete(name = "% loss (n)", labels = c(a,b,c,d, e)) +
  theme(legend.position = "bottom")

# loss plot for kml cluster D

a = paste( "0 %", " (", tab_loss0$D[1], ")", sep = "")
b = paste( "1 %", " (", tab_loss0$D[2], ")", sep = "")
c = paste( "5 %", " (", tab_loss0$D[3], ")", sep = "")
d = paste( "10 %", " (", tab_loss0$D[4], ")", sep = "")
e = paste( "20 %", " (", tab_loss0$D[5], ")", sep = "")
ggplot(er_long0[er_long0$kml.clusters == "D",],
       aes(x = visit, y = score, group = variable,
                                                colour = variable)) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_vline(xintercept = c(6, 9), colour = "red", linetype = "dotted") +
  theme(legend.position = "bottom") +
  expand_limits(x=c(1,13), y=c(3.7, 4)) +
  scale_colour_discrete(name = "% loss (n)", labels = c(a,b,c,d, e)) +
  scale_x_discrete(name = "visit",
                   limits = factor(c(1,2,3,4,5,6,7,8,9,10,11,12,13)))

### end of loss plots and loss ###
```

Bibliography

Bates, D., Machler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, **67**, 1–48. 6

Diggle, P., Heagerty, P., Liang, K.-Y., and Zeger, S. (2013). *Analysis of longitudinal data*. Oxford University Press. 3

Drescher, B. and Schlender-Bobbis, I. (1996). Guide to skin diseases in rabbits. *In Practice*, **41**, 488–497. 1, 5, 6

Genolini, C. and Falissard, B. (2010). Kml: A package to cluster longitudinal data. *Computer Methods and Programs in Biomedicine*, **104**, e112 – e121. 3, 7, 33

Jeppson, H., Hofmann, H., and Cook, D. (2018). *ggmosaic: Mosaic Plots in the 'ggplot2' framework*. R package version 0.2.0. 6

Kratzer, G., Pittavino, M., Comin, A., Lewis, F., and Furrer, R. (2019). Additive Bayesian Network Modelling with the R package abn. *Additive Bayesian Network Modelling*, **1**, 1–38. 1

Kuznetsova, A., Brockhoff, P., and Christensen, R. (2017). *lmerTest Package: Tests in Linear Mixed Effects Models*. 6

Mancinelli, E., Keeble, E., Richardson, J., and Hedley, J. (2014). Husbandry risk factors associated with hock pododermatitis in *uk* pet rabbits (oryctolagus cuniculus). *Veterinary Record*, **174**, 429–429. 1, 35

Martorell, J. (2014). Scoring pododermatitis in pet rabbits. *Vetenariy Recreation*, **174**, 427–428. 1

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 6

Rommers, J. and Meierhod, R. (1996). The effect of different floor types on footpad injuries of rabbit does. 6*th World Rabbit Congress Toulouse 2*, **2**, 431–436. 1

Rosell, J. and De la Fuente, L. (2009). Effect of footrests onthe incidence of ulcerative pododermatitis in domestic rabbit does. *Animal Welfare*, **18**, 199–204. 1

Ruchti, S., Kratzer, G., Furrer, R., Hartnack, S., Wurbel, H., and Gebhardt-Henrich, S. (2019). Progression and risk factors of pododermatitis in part-time group housed rabbit does in Switzerland. *Preventive Veterinary Medicine*, **166**, 56–64. 1, 2, 4, 5, 6, 12, 31, 32, 33, 35

Ruchti, S., Meier, A., Wurbel, H., and Kratzer, G. (2018). Pododermatitis in group housed rabbit does in Switzerland - prevalence, severity and risk factors. *Preventative Veterinary Medicine*, **158**, 114–121. 1, 2, 6

Seaman, S. C., Waran, N. K., Mason, G., and D′Eath, R. B. (2008). Animal economics: assessing the motivation of female laboratory rabbits to reach a platform, social contact and food. *Animal Behaviour*, **75**, 31–42. 1

Seaman, S. C., Waran, N. K., Mason, G., and D′Eath, R. B. (2013). Development of a pododermatitis score in breeding does using clustering methods. *Animal*, **158**, 114–121. 1

Twisk, J. (2013). *Applied Longitudinal Data Analysis for Epidemiology*. Cambridge University Press, Cambridge. 3

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. 6, 10, 13, 14, 19, 20, 21, 23, 26, 27, 28, 29, 30

Wickham, H. (2017). *tidyverse: Easily Install and Load the Tidyverse*. R package version 1.2.1. 6

Wickham, H. and Bryan, J. (2019). *readxl: Read Excel Files*. R package version 1.3.0. 6