

Characterizing dominance in free stall housed dairy cattle based on competitive behavior at the water trough

Helen Chen, Annie Wang, Vivian Zhu

April 15, 2023

Summary

The research aims to investigate the social hierarchy of individuals within a group of dairy cattle based on competitive behavior at the water trough. The objectives of the study are to examine whether the data collected from water bins are as reliable as data from feed bins in measuring the dominance hierarchy of cows; to check whether the temperature and humidity affect the dominance hierarchy measured from the data at water bins. By conducting the Elo rating algorithm, Spearman's rank correlation test, VAH algorithm (an iterative undersampling algorithm introduced in section 3.3), and two-sample t-test, we discover that there is a moderate correlation between the social hierarchy of cows estimated at water bins and the one estimated at feed bins, suggesting there is insufficient evidence to say the replacements recorded at water bins may be reliable in predicting cows' dominance hierarchy. Moreover, there is some evidence that suggests that THI affects the dominance measure at water bins.

1. Introduction

The purpose of this report is to investigate how the involvement of dairy cows in agonistic interactions (replacements) among feed bins and water bins is associated with their social hierarchy in the herd. The motivation behind this study is to improve herd management strategy and animal welfare through the study of animal social behavior. With the help of automated measures of agonistic interactions (replacements), each cow was assigned a rank within the herd based on its performance in the replacements. Previous studies have shown that the social hierarchy of cows can be captured by measures of how cows engage in replacements among the feed bins. However, little is known about the reliability of estimating the social dominance hierarchy based on cows' behaviors among the water bins.

This study aims to check whether the data recorded from the water bin is as reliable as the data recorded from the feed bins. By the influence of temperature and humidity, cattle tend

to compete more often for water due to thirst during summer compared to winter. Therefore, the second objective of this study is to check whether the temperature and humidity affect the dominance hierarchy measured from the data at water bins. The study examines two hypotheses: (1) the data collected from water bins are as reliable as data from feed bins in measuring the dominance hierarchy of cows, and (2) when THI reaches a threshold of 65, the competition for water is more likely to increase, which affects the dominance measure. To investigate the two hypotheses, we performed a series of statistical analysis tasks. We used the Elo rating algorithm to calculate cow’s dominance scores and Spearman’s rank correlation test for comparing the similarities between ranks in different groups. Then we used two-sample t-tests to investigate if the difference between two correlation coefficients is statistically significant; Furthermore, we integrated the VAH algorithm which is an under-sampling algorithm to account for inconsistencies introduced through random sampling.

The following introduces the structure of the remainder of this report. Section 2 describes the data set and the data clustering process. Section 3 introduces statistical methods (i.e., Elo rating algorithm, Spearman’s rank correlation test, VAH algorithm, and two-sample t-test) and applies these methods to test the two hypotheses. Section 4 presents the results obtained from performing the statistical analysis. Lastly, sections 5 and 6 discuss the conclusions and limitations of this study.

2. Data Description

2.1 The Data

Electronic feed bin and water bin data for agonistic interactions (replacements) were recorded from July 2020 to May 2021 among the population of 48 Holstein dairy cows in a dynamic group. The dynamic group worked as the following: once every 2 weeks, 5 cows would be taken out of the dynamic group and replaced by 5 new cows, which formed multiple stable groups. Moreover, only the healthy cows would be kept in the group and cows could leave and return later. In this study, there were 159 cows in total. In each replacement, the actor cow takes the bin that the reactor cow had originally been at. The IDs of the cows involved in replacements were recorded by electronic bins using the ear tag radiofrequency identification (ID). Temperature humidity index (THI) was calculated based on the air temperature and humidity index, which was recorded using HOBO temperature humidity loggers (Onset Computer Corporation, USA) every 5 minutes. For each hour, the average, minimum, and maximum THI were computed respectively to provide a better understanding of how THI varied. This was an observational study since no treatment was involved to affect the outcome. In total, 180905 replacements were observed with 171622 of them at feed bins and 9283 at water bins. Additionally, no missing data were reported. A detailed summary of the original data is shown in Table 1.

Variable	Description
date	A timestamp when the agonistic interaction (replacement) happened
hour	An hourly timestamp in the range of 0 to 24 when the agonistic interaction (replacement) happened
Bin	The ID for the bin (feed bins have IDs in the range from 1 to 30 and water bins have IDs in the range from 101 to 105)
Bin_type	Bin type can be either feed or water
Reactor_cow	The ID for the cow that was involved in the replacement. Specifically, this is the cow originally at the bin and later displaced.
Actor_cow	The ID for the cow that initiated the replacement.
Bout_interval	The amount of time taken for the replacement
THI_mean	The average temperature humidity index in the given hour
THI_min	The minimum temperature humidity index in the given hour
THI_max	The maximum temperature humidity index in the given hour

Table 1: Variables

2.2 Data Clustering

We divided data into two groups according to their bin type (either feed bin or water bin) in order to address the first hypothesis that replacements at the water bins are as reliable as replacements at the feed bins in measuring the dominance hierarchy. We further divided the two groups into four subgroups based on the THI of replacements in order to examine the second hypothesis which states that when THI reaches a high threshold (65), the competition for water is more likely to increase, which affects the dominance measure. The Elo rating algorithm requires the number of replacements to be at least 15-20 times the number of cows in the group. There were 159 cows observed in the data set and with the help of relevant

materials posted by the University of Minnesota (2020), we set the THI threshold to be at 65. Table 2 below shows the number of replacements in each group under different THI thresholds.

Bin Type	THI Level	Number of Replacements
Water	high THI($\text{THI_MAX} \geq 65$)	2595
Water	low THI($\text{THI_MAX} < 65$)	6688
Feed	high THI($\text{THI_MAX} \geq 65$)	44992
Feed	low THI($\text{THI_MAX} < 65$)	126630

Table 2: Number of replacements in each group based on Bin Type and different THI thresholds

3. Statistical Methods

3.1 Elo rating algorithm

We used the Elo rating algorithm to calculate the dominance scores for cows. Dominance scores are utilized to capture cows’ dominant behavior and are later used to establish the dominance rank for Spearman’s rank correlation test. We utilized the Elo rating algorithm (using the R package EloRating) in our analysis to assign cows’ dominance scores (Elo ratings) based on the chronological order of agonistic interactions (replacements). Each cow began with a predetermined score of 1000 points. Following each replacement event, the actor cow gained points and the reactor cow lost points. The maximum number of points that were gained or lost at each replacement is a function of k . We set k to 20 to account for the disproportionally large effects of single replacements. For example, if one cow (actor) replaced another cow (reactor), this one replacement would cause at most 20 points to be gained and lost. The exact number of points is proportional to the difference between the actor and the reactor cows’ scores prior to the interaction. For instance, if the two cows had the same Elo rating prior to the replacement, then the actor cow gained 10 points and the reactor lost 10 points. If the actor cow had higher Elo ratings than the reactor cow prior to the replacement, the actor cow gained 0-9 points, and the reactor cow lost 0-9 points. Contrary, If the reactor cow had higher Elo ratings than the actor cow prior to the replacement, both gained/lost 11-20 points, respectively.

3.2 Spearman’s rank correlation test

Spearman’s rank correlation test compares the ranking between two variables for every subject and analyzes the level of association between them. Spearman’s rank correlation test does not require assumptions about the distribution of variables. Instead, it requires the assumption that data must be at least ordinal and the scores on one variable must be monotonically related to the other variable, which is suitable for the given dataset. The interest of our analysis is about the dominance hierarchy of cows, which can be obtained by ranking individual cows based on their final Elo rating scores produced in the Elo rating algorithm. Using Spearman’s rank correlation test is appropriate since both hypotheses examine the correlation of dominance hierarchy between groups. The relationship between the Elo-ratings from the water bins and the feed bins was first examined by Spearman’s rank correlation test to investigate the first objective. In a similar manner, we performed the tests for comparisons between the 4 groups (Water Bin & high THI, Feed Bin & high THI, Water Bin & low THI, and Feed Bin & low THI) to explore the second objective. Table 3 below presents the set of Spearman’s rank correlation tests performed to examine the two hypotheses.

	Group 1	Group 2
Test 1	Feed bins	Water bins
Test 2	Feed bins & high THI	Water bins & high THI
Test 3	Feed bins & low THI	Water bins & low THI
Test 4	Water bins & high THI	Water bins & low THI
Test 5	Feed bins & High THI	Feed bins & low THI

Table 3: Pairwise comparison tests

3.3 VAH algorithm

VAH algorithm is an iterative under-sampling approach designed by us to account for inconsistencies introduced through random sampling. As introduced in section 3.1, the maximum number of points that were gained or lost at each replacement is a function of k. According to previous studies (Foris et al., 2021), we planned to use a fixed k value of 20. Due to the fact that we have much more replacements at feed bins in comparison to the ones at water bins, we can predict a huge gap in the sample size after dividing the groups based on bin types and THI threshold. Therefore, using the same k across all groups would result in distinct dominance score ranges that may not be appropriate to compare against each other. Consequently, we designed the VAH algorithm to solve this problem and it’s as follows.

1. For test 1 in Table 3, we sampled observations based on the smaller group size (water bins group).
2. For the remaining 4 tests, since we were doing pairwise comparisons between four subgroups (mentioned in Table 3), the smallest group size was used for sampling, which is the one for the water bins & high THI group.
3. Then we calculated Elo rating scores and ran the Spearman's rank correlation test for each comparison group.
4. Given the fact that there's a huge difference in the number of observations in each group, the result obtained from sampling a subset of them and feeding it into the Elo rating algorithm may be fluctuating due to chance. Therefore, we repeated the same process 50 times and used the 50 correlation coefficients for further analysis.

3.4 Two-sample t-test

The two-sample t-test is to test if the difference between two sets of correlation coefficients is statistically significant. For each test listed in Table 3, The VAH algorithm was executed and calculated 50 correlation coefficients. We first used a two-sample t-test to compare the correlation coefficients obtained from the VAH algorithm for test 2 and test 3 to investigate whether having THI level as a factor affects the similarities of dominance ranks captured at feed bins and water bins. In a similar manner, we compared the correlation coefficients obtained from the VAH algorithm for test 4 and test 5 to examine if changing the THI level impacted the dominance measured at water bins significantly more than the one measured at feed bins. The results of these two comparisons are shown in the results section.

4. Results

We performed 5 comparison tests in total. Within each test we compared between two groups and the VAH algorithm was executed to produce 50 correlation coefficients. The average of the correlation coefficients (ρ) was obtained for each of the comparison tests. The summary table below lists the results produced by the VAH algorithm. We used the result of Test 1 to investigate the strength of association between the dominance measures at feed bins and water bins in general. We used Tests 2-5 to investigate whether THI level has an impact on the dominance measures at feed and water bins. These analyses were done in R with a significance set to $P < .05$.

	Group 1	Group 2	Average correlation coefficient ($\overline{\rho}$)
Test 1	Feed bins	Water bins	.59

	Group 1	Group 2	Average correlation coefficient (\overline{rho})
Test 2	Feed bins & high THI	Water bins & high THI	.52
Test 3	Feed bins & low THI	Water bins & low THI	.58
Test 4	Water bins & high THI	Water bins & low THI	.55
Test 5	Feed bins & High THI	Feed bins & low THI	.83

Table 4: Spearman’s rank correlation test results on 5 different comparisons using VAH algorithm.

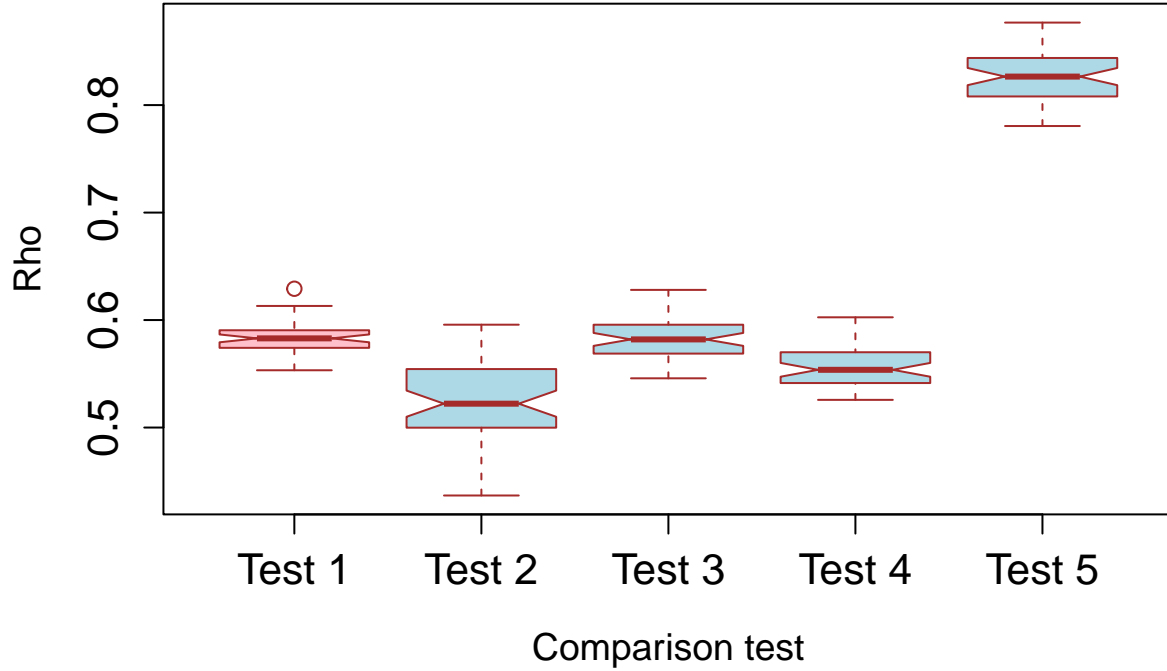
We calculated correlation coefficients (ρ) in Test 1 which provided us with the strength of the relationship between the two dominance measures at feed bins and water bins. The results of Test 1 indicate that there is a significant positive association between the two groups ($p < .05$ for all 50 trials). The box plot in Figure 1 (first from the left) shows the distributions of the 50 correlation coefficients as the result of the VAH algorithm. The strength of the correlation is moderate between water bins and feed bins. Thus, the relationship between dominance scores estimated from feed bins and the ones estimated from water bins is not strong.

The results of the two Spearman correlation tests (Tests 2 & 3) show a significantly positive association between dominance scores estimated from feed bins and water bins (Both tests: $p < .05$ for all 50 trials of sampling analysis). The correlation in dominance measures between water and feed bins is slightly stronger under low THI levels. The two box plots in Figure 1 (second and third from the left) below show the distributions of the two test results and illustrate that a High THI level significantly weakens the correlation between dominance measures at the two bin types. The result of two sample t-tests ($t = -9.48$; $p < .05$) indicates that under different THI levels (low/high THI), there is a significant difference in the associations of dominance hierarchy captured at feed bins and water bins.

In order to further determine the THI impact on the dominance hierarchy measured at water bins, we compared it to the impact of THI on the dominance hierarchy measured at feed bins. The results of the two Spearman correlation tests (Tests 4 & 5) indicate a significantly positive association between dominance scores estimated under high THI levels and the ones estimated under low THI levels (Both tests: $p < .05$ for all 50 trials of sampling analysis). The two box plots in Figure 1 (fourth and fifth from the left) below show the distributions of the two test results and reveal that the dominance hierarchy measured at feed bins under different THI levels tends to be more similar and consistent. In contrast, the dominance

measurements at the water bins seem to be less consistent under different THI levels. The result from the two-sample t-test ($t = -61.46$; $p < .05$) shows the difference in the correlations between Tests 4 and 5 is statistically significant, which substantiates the second hypothesis that THI levels affect the structure of dominance hierarchy captured at water bins.

Correlation coefficients (rho) calculated from the 5 comparison test



5. Conclusions

In this report, we discussed how we utilized the Elo rating algorithm to calculate the cows' dominance scores. In addition, we integrated a new iterative approach (i.e., VAH algorithm) to perform Spearman's rank correlation with additional results from two-sample t-tests to investigate the two research hypotheses. Since this is an observational study, we can only refer to correlation rather than causation. The two dominance hierarchies finalized from the water and feed bins only have a moderate correlation. Therefore, we could not confidently say the use of replacements at the water bins is as reliable as the use of replacements at the feed bins for accessing cows' dominance hierarchy. From the results of Spearman's rank correlation test and two sample t-tests, the THI levels have a more notable influence on the dominance scores measured at water bins. Thus the structure of the dominance hierarchy at water bins would be affected by THI levels.

6. Limitation

6.1 The order of replacements

A problem that we encountered is that more than one replacement could happen for an individual cow during a given hour. Since we only had the hourly timestamp, it was not possible to determine the relative order. Imagine if a cow won four replacements in a row and lost one shortly after, its winning probability before the fifth replacement would be relatively high, resulting in a great loss of scores in the final round. However, if the order was “win, lose, win, win, win”, the final dominance score might be distinct from the previous scenario. The way we investigated this problem was to randomly shuffle all rows, calculate their corresponding dominance scores and compare the result from Spearman’s ranking correlation at the end. The result is as follows:

Trial	Correlation
Trial 1	0.6316266
Trial 2	0.6400618
Trial 3	0.635646

Table 5: Correlation between rankings at feed and water bins

From the three trials, the correlations are very close to each other, suggesting that the order for this data set doesn’t have a huge influence on the cows’ dominance hierarchy. Therefore, we proceed without further modification.

7. Reference

EloRating - a brief tutorial - cloud.r-project.org. (n.d.). Retrieved February 26, 2023, from https://cloud.r-project.org/web/packages/EloRating/vignettes/EloRating_tutorial.pdf

Neumann, Christof, et al. “Assessing Dominance Hierarchies: Validation and Advantages of Progressive Evaluation with ELO-Rating.” *Animal Behaviour*, vol. 82, no. 4, 2011, pp. 911–921., <https://doi.org/10.1016/j.anbehav.2011.07.016>.

Armstrong, J. (n.d.). Heat stress in dairy cattle. UMN Extension. Retrieved February 25, 2023, from <https://extension.umn.edu/dairy-milking-cows/heat-stress-dairy-cattle>

8. Appendix

Loading libraries

Calculating Elo Score

The following code segment provided by the client will calculate the elo ratings of cows.

```
# setting up
all_replacement_THI <- read.csv('../data/replacement_with_THI.csv')
water <- all_replacement_THI %>% filter(Bin_type == 'Water')
feed <- all_replacement_THI %>% filter(Bin_type == 'Feed')
THI_threshold <- 65
water_high <-
  all_replacement_THI %>% filter(Bin_type == 'Water', THI_max>=THI_threshold)
water_low <-
  all_replacement_THI %>% filter(Bin_type == 'Water', THI_max<THI_threshold)
feed_high <-
  all_replacement_THI %>% filter(Bin_type == 'Feed', THI_max>=THI_threshold)
feed_low <-
  all_replacement_THI %>% filter(Bin_type == 'Feed', THI_max<THI_threshold)

water_cows <- unique(water$Reactor_cow,water$Actor_cow)
feed_cows <- unique(feed$Reactor_cow,feed$Actor_cow)
intersection <- Reduce(intersect, list(water_cows,feed_cows))

# filtering
water_high_cows <- unique(water_high$Reactor_cow,water_high$Actor_cow)
water_low_cows <- unique(water_low$Reactor_cow,water_low$Actor_cow)
feed_high_cows <- unique(feed_high$Reactor_cow,feed_high$Actor_cow)
feed_low_cows <-unique(feed_low$Reactor_cow,feed_low$Actor_cow)
sub_intersection <- Reduce(intersect, list(water_high_cows,
                                           water_low_cows, feed_high_cows,
                                           feed_low_cows))

water <- water %>%
  filter(Actor_cow %in% intersection & Reactor_cow %in% intersection)

feed <- feed %>%
  filter(Actor_cow %in% intersection & Reactor_cow %in% intersection)

water_high <- water_high %>%
```

```

    filter(Actor_cow %in% sub_intersection & Reactor_cow %in% sub_intersection)

water_low <- water_low %>%
  filter(Actor_cow %in% sub_intersection & Reactor_cow %in% sub_intersection)

feed_high <- feed_high %>%
  filter(Actor_cow %in% sub_intersection & Reactor_cow %in% sub_intersection)

feed_low <- feed_low %>%
  filter(Actor_cow %in% sub_intersection & Reactor_cow %in% sub_intersection)

```

Sampling replacements

Repeat the same process (num) times

```

water_feed_cor = 0;
water_feed_high_cor = 0;
water_feed_low_cor = 0;
water_high_low_cor = 0;
feed_high_low_cor = 0;

sig_water_feed_cor = 0;
sig_water_feed_high_cor = 0;
sig_water_feed_low_cor = 0;
sig_water_high_low_cor = 0;
sig_feed_high_low_cor = 0;

num <- 10
for (i in 1:num) {
  # -----* Test 1 *-----
  # ----- feed bins overall -----
  sub_repl <- feed[sample(1:nrow(water)),]
  sub_repl$index <- seq(1, nrow(sub_repl))
  ## Order replacements
  elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                  "Reactor_cow", "hour",
                                                  "Bin", "date")]
  colnames(elo_repl_list)=c("winner", "loser", "time", "bin", "date")
  elo_repl_list$index <- seq(1, nrow(elo_repl_list))
  #Calculate the ELO scores
  elo_res_feed=elo.seq(winner=as.character(elo_repl_list$winner),
                      loser=as.character(elo_repl_list$loser),

```

```

                                Date=elo_repl_list$date,
                                k=20,
                                #presence=presence_comb,
                                runcheck = F,
                                progressbar=T)

# ----- water bins overall -----
sub_repl <- water
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                "Reactor_cow","hour",
                                                "Bin", "date")]
colnames(elo_repl_list)=c("winner","loser","time","bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_water = elo.seq(winner=as.character(elo_repl_list$winner),
                        loser=as.character(elo_repl_list$loser),
                        Date=elo_repl_list$date,
                        k=20,
                        #presence=presence_comb,
                        runcheck = F,
                        progressbar=T)

# ----- calculate feed vs water correlation -----
last_elo_water <- extract_elo(elo_res_water)
last_elo_feed <- extract_elo(elo_res_feed)
cows <- Reduce(intersect, list(names(last_elo_water),names(last_elo_feed)))
sorted_last_elo_feed <- last_elo_feed[sort(cows)]
sorted_last_elo_water <- last_elo_water[sort(cows)]
water_feed_test <- cor.test(x=unname(sorted_last_elo_water),
                            y=unname(sorted_last_elo_feed),
                            method = 'spearman')
water_feed_cor <- water_feed_cor + water_feed_test$estimate
sig_water_feed_cor <- sig_water_feed_cor + (water_feed_test$p.value < 0.05)

# -----* Test 2 *-----
# ----- feed bins high THI -----
sub_repl <- feed_high[sample(1:nrow(water_high)),]
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                "Reactor_cow","hour",
                                                "Bin", "date")]

```

```

colnames(elo_repl_list)=c("winner","loser","time","bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_feed_high_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                             loser=as.character(elo_repl_list$loser),
                             Date=elo_repl_list$date,
                             k=20,
                             #presence=presence_comb,
                             runcheck = F,
                             progressbar=T)

# ----- water bins high THI -----
sub_repl <- water_high
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),
                          c("Actor_cow", "Reactor_cow",
                            "hour", "Bin", "date")]
colnames(elo_repl_list)=c("winner","loser","time","bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_water_high_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                              loser=as.character(elo_repl_list$loser),
                              Date=elo_repl_list$date,
                              k=20,
                              #presence=presence_comb,
                              runcheck = F,
                              progressbar=T)

# -----calculate feed high THI vs water high THI correlation --
last_elo_water_high_THI <- extract_elo(elo_res_water_high_THI)
last_elo_feed_high_THI <- extract_elo(elo_res_feed_high_THI)
high_THI_cows <-
  Reduce(intersect, list(names(last_elo_water_high_THI),
                        names(last_elo_feed_high_THI)))
sorted_last_elo_feed_high_THI <-
  last_elo_feed_high_THI[sort(high_THI_cows)]
sorted_last_elo_water_high_THI <-
  last_elo_water_high_THI[sort(high_THI_cows)]
water_feed_high_test <- cor.test(x=unname(sorted_last_elo_feed_high_THI),
                                y=unname(sorted_last_elo_water_high_THI),
                                method = 'spearman')
water_feed_high_cor <- water_feed_high_cor + water_feed_high_test$estimate
sig_water_feed_high_cor <- sig_water_feed_high_cor +

```

```

(water_feed_high_test$p.value < 0.05)

# -----* Test 3 *-----
# ----- feed bins low THI -----
sub_repl <- feed_low[sample(1:nrow(water_low)),]
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                "Reactor_cow", "hour",
                                                "Bin", "date")]
colnames(elo_repl_list)=c("winner", "loser", "time", "bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_feed_low_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                             loser=as.character(elo_repl_list$loser),
                             Date=elo_repl_list$date,
                             k=20,
                             #presence=presence_comb,
                             runcheck = F,
                             progressbar=T)

# ----- water bins low THI -----
sub_repl <- water_low
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),
                          c("Actor_cow", "Reactor_cow",
                            "hour", "Bin", "date")]
colnames(elo_repl_list)=c("winner", "loser", "time", "bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_water_low_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                              loser=as.character(elo_repl_list$loser),
                              Date=elo_repl_list$date,
                              k=20,
                              #presence=presence_comb,
                              runcheck = F,
                              progressbar=T)

# ----- calculate feed low THI vs water low THI correlation -----
last_elo_water_low_THI <- extract_elo(elo_res_water_low_THI)
last_elo_feed_low_THI <- extract_elo(elo_res_feed_low_THI)
low_THI_cows <- Reduce(intersect, list(names(last_elo_water_low_THI),

```

```

names(last_elo_feed_low_THI)))
sorted_last_elo_feed_low_THI <- last_elo_feed_low_THI[sort(low_THI_cows)]
sorted_last_elo_water_low_THI <- last_elo_water_low_THI[sort(low_THI_cows)]
water_feed_low_test <- cor.test(x=unname(sorted_last_elo_feed_low_THI),
                                y=unname(sorted_last_elo_water_low_THI),
                                method = 'spearman')
water_feed_low_cor <- water_feed_low_cor + water_feed_low_test$estimate
sig_water_feed_low_cor <- sig_water_feed_low_cor +
  (water_feed_low_test$p.value < 0.05)

# -----* Test 4 *-----
# ----- water bins high THI -----
sub_repl <- water_high
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                "Reactor_cow", "hour",
                                                "Bin", "date")]
colnames(elo_repl_list)=c("winner", "loser", "time", "bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_water_high_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                               loser=as.character(elo_repl_list$loser),
                               Date=elo_repl_list$date,
                               k=20,
                               #presence=presence_comb,
                               runcheck = F,
                               progressBar=T)

# ----- water bins low THI -----
sub_repl <- water_low[sample(1:nrow(water_high)),]
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),c("Actor_cow",
                                                "Reactor_cow", "hour",
                                                "Bin", "date")]
colnames(elo_repl_list)=c("winner", "loser", "time", "bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_water_low_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                              loser=as.character(elo_repl_list$loser),
                              Date=elo_repl_list$date,
                              k=20,
                              #presence=presence_comb,

```



```

runcheck = F,
progressbar=T)

# ----- calculate water high THI vs low THI correlation -----
last_elo_water_low_THI <- extract_elo(elo_res_water_low_THI)
last_elo_water_high_THI <- extract_elo(elo_res_water_high_THI)
water_cows <- Reduce(intersect,
                      list(names(last_elo_water_low_THI),
                           names(last_elo_water_high_THI)))
water_high_low_test <-
  cor.test(x=unname(last_elo_water_low_THI[sort(water_cows)]),
           y=unname(last_elo_water_high_THI[sort(water_cows)]),
           method = 'spearman')
water_high_low_cor <- water_high_low_cor + water_high_low_test$estimate
sig_water_high_low_cor <- sig_water_high_low_cor +
  (water_high_low_test$p.value < 0.05)

# -----* Test 5 *-----
# ----- feed bins high THI -----
sub_repl <- feed_high
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),
                          c("Actor_cow", "Reactor_cow",
                            "hour", "Bin", "date")]
colnames(elo_repl_list)=c("winner","loser","time","bin", "date")
elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_feed_high_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                              loser=as.character(elo_repl_list$loser),
                              Date=elo_repl_list$date,
                              k=20,
                              #presence=presence_comb,
                              runcheck = F,
                              progressbar=T)

# ----- feed bins low THI -----
sub_repl <- feed_low[sample(1:nrow(feed_high)),]
sub_repl$index <- seq(1, nrow(sub_repl))
## Order replacements
elo_repl_list <- sub_repl[order(sub_repl$hour),
                          c("Actor_cow", "Reactor_cow",
                            "hour", "Bin", "date")]
colnames(elo_repl_list)=c("winner","loser","time","bin", "date")

```

```

elo_repl_list$index <- seq(1, nrow(elo_repl_list))
#Calculate the ELO scores
elo_res_feed_low_THI=elo.seq(winner=as.character(elo_repl_list$winner),
                             loser=as.character(elo_repl_list$loser),
                             Date=elo_repl_list$date,
                             k=20,
                             #presence=presence_comb,
                             runcheck = F,
                             progressBar=T)

# ----- calculate feed high THI vs low THI correlation -----
last_elo_feed_low_THI <- extract_elo(elo_res_feed_low_THI)
last_elo_feed_high_THI <- extract_elo(elo_res_feed_high_THI)
feed_cows <- Reduce(intersect,
                    list(names(last_elo_feed_low_THI),
                         names(last_elo_feed_high_THI)))
feed_high_low_test <-
  cor.test(x=unname(last_elo_feed_low_THI[sort(feed_cows)]),
           y=unname(
             last_elo_feed_high_THI[sort(feed_cows)]),
           method = 'spearman')
feed_high_low_cor <- feed_high_low_cor + feed_high_low_test$estimate
sig_feed_high_low_cor <- sig_feed_high_low_cor
+ (feed_high_low_test$p.value < 0.05)
}

# compute average correlation
feed_water_cor <- water_feed_cor/num
final_water_feed_high_cor <- water_feed_high_cor/num
final_water_feed_low_cor <- water_feed_low_cor/num
final_water_high_low_cor <- water_high_low_cor/num
final_feed_high_low_cor <- feed_high_low_cor/num

sig_water_feed_cor
sig_water_feed_high_cor
sig_water_feed_low_cor
sig_water_high_low_cor
sig_feed_high_low_cor

```

Calculate fisher's z transformation score

```

# -----Test 2 final_water_feed_high_cor & 3 final_water_feed_low_cor -----
library(DescTools)
# ----- COMPARING BETWEEN TEST 2 RHO VS. TEST 3 RHO -----
n1 <- length(sorted_last_elo_feed_high_THI) +
  length(sorted_last_elo_water_high_THI)
n2 <- length(sorted_last_elo_feed_low_THI) +
  length(sorted_last_elo_water_low_THI)
z_final_water_feed_high_cor <- FisherZ(final_water_feed_high_cor)
z_final_water_feed_low_cor <- FisherZ(final_water_feed_low_cor)
zobs_test23 <-
  (z_final_water_feed_high_cor-z_final_water_feed_low_cor)/
  sqrt(1/(n1-3)+1/(n2-3))
pval_test23 <- 2 * pnorm(-abs(zobs_test23))
pval_test23

# -----Test 4 final_water_high_low_cor & 5 final_feed_high_low_cor -----

# ----- COMPARING BETWEEN TEST 4 RHO VS. TEST 5 RHO -----
n1<- length(last_elo_water_low_THI[sort(water_cows)]) +
  length(last_elo_water_high_THI[sort(water_cows)])
n2 <- length(last_elo_feed_low_THI[sort(feed_cows)]) +
  length(last_elo_feed_high_THI[sort(feed_cows)])
z_final_water_high_low_cor <- FisherZ(final_water_high_low_cor)
z_final_feed_high_low_cor <- FisherZ(final_feed_high_low_cor)
zobs_test45 <-
  (z_final_water_high_low_cor-z_final_feed_high_low_cor)/
  sqrt(1/(n1-3)+1/(n2-3))
pval_test45 <- 2 * pnorm(-abs(zobs_test45))
pval_test45

```

Optimize k

```

elo_res_water <- readRDS('../data/elo_res_water.rds')
elo_res_feed <- readRDS('../data/elo_res_feed.rds')
elo_res_water_low_THI <- readRDS('../data/elo_res_water_low_THI.rds')
elo_res_water_high_THI <- readRDS('../data/elo_res_water_high_THI.rds')
elo_res_feed_low_THI <- readRDS('../data/elo_res_feed_low_THI.rds')
elo_res_feed_high_THI <- readRDS('../data/elo_res_feed_high_THI.rds')

#optimize k
ores_water <- optimizek(elo_res_water, krange = c(1, 100), resolution = 100)

```

```

ores_water$best

ores_feed <- optimizek(elo_res_feed, krange = c(1, 100), resolution = 100)
ores_feed$best

ores_water_low_THI <-
  optimizek(elo_res_water_low_THI, krange = c(1, 100), resolution = 100)
ores_water_low_THI$best
ores_water_high_THI <-
  optimizek(elo_res_water_high_THI, krange = c(1, 100), resolution = 100)
ores_water_high_THI$best
ores_feed_low_THI <-
  optimizek(elo_res_feed_low_THI, krange = c(1, 100), resolution = 100)
ores_feed_low_THI$best
ores_feed_high_THI <-
  optimizek(elo_res_feed_high_THI, krange = c(1, 100), resolution = 100)
ores_feed_high_THI$best

```

Figure 4.1 code:

```

df_f_w <- data.frame(x = rank(unname(sorted_last_elo_feed), ties.method =
                           'average'),
                    y = rank(unname(sorted_last_elo_water),
                           ties.method = 'average'))
sp_df_f_w <- ggscatter(df_f_w, x = 'x', y = 'y',
  add = "reg.line", # Add regression line
  add.params = list(color = "blue", fill = "lightgray"),
  conf.int = TRUE, # Add confidence interval
  cor.coeff.args = list(method = "spearman", label.sep = "\n"),
  xlab = "Dominance ranks of cows in Feed Bin group",
  ylab = "Dominance ranks of cows in Water Bin group"
) + labs(caption = "Figure 4.1:
  Correlation between the
  dominance ranks at Feed Bin and Water Bin (Overall)")
sp_df_f_w

```

Figure 4.2 code:

```

df_fh_wh <- data.frame(x = rank(unname(sorted_last_elo_feed_high_THI),
                              ties.method = 'average'),

```

```

        y = rank(unname(sorted_last_elo_water_high_THI),
                  ties.method = 'average'))
sp_df_fh_wh <- ggscatter(df_fh_wh, x = 'x', y = 'y',
  add = "reg.line", # Add regression line
  add.params = list(color = "blue", fill = "lightgray"),
  conf.int = TRUE, # Add confidence interval
  cor.coeff.args = list(method = "spearman"),
  xlab = "Dominance ranks of cows in Feed Bin High THI group",
  ylab = "Dominance ranks of cows in Water Bin High THI group"
) +
  labs(caption = "Figure 4.2: Correlation between the dominance ranks at
    Feed Bin and Water Bin under High THI")

```

Figure 4.3 code:

```

df_fl_wl <- data.frame(x = rank(unname(sorted_last_elo_feed_low_THI),
  ties.method = 'average'),
  y = rank(unname(sorted_last_elo_water_low_THI),
    ties.method = 'average'))

sp_df_fl_wl <- ggscatter(df_fl_wl, x = 'x', y = 'y',
  add = "reg.line", # Add regression line
  add.params = list(color = "blue", fill = "lightgray"),
  conf.int = TRUE, # Add confidence interval
  cor.coeff.args = list(method = "spearman", label.sep = "\n"),
  xlab = "Dominance ranks of cows in Feed Bin Low THI group",
  ylab = "Dominance ranks of cows in Water Bin Low THI group"
) + labs(caption = "Figure 4.3: Correlation between the dominance
  ranks at Feed Bin and Water Bin under Low THI")

```

Figure 4.4 code:

```

df_wh_wl <- data.frame(x = rank(unname(
  last_elo_water_high_THI[sort(water_cows)]),
  ties.method = 'average'),
  y = rank(unname(
    last_elo_water_low_THI[sort(water_cows)]),
    ties.method = 'average'))

sp_df_wh_wl <- ggscatter(df_wh_wl, x = 'x', y = 'y',
  add = "reg.line", # Add regression line

```

```

add.params = list(color = "blue", fill = "lightgray"),
conf.int = TRUE, # Add confidence interval
cor.coeff.args = list(method = "spearman", label.sep = "\n"),
xlab = "Dominance ranks of cows in Water Bin High THI group",
ylab = "Dominance ranks of cows in Water Bin Low THI group"
) + labs(caption = "Figure 4.4: Correlation between the dominance ranks
          at Water Bin under High THI vs under Low THI")

```

Figure 4.5 code:

```

df_fh_fl <- data.frame(x = rank(unname(
  last_elo_feed_high_THI[sort(feed_cows)]),
                        ties.method = 'average'),
                      y = rank(unname(
                        last_elo_feed_low_THI[sort(feed_cows)]),
                        ties.method = 'average'))
sp_df_fh_fl <- ggscatter(df_fh_fl, x = 'x', y = 'y',
  add = "reg.line", # Add regression line
  add.params = list(color = "blue", fill = "lightgray"),
  conf.int = TRUE, # Add confidence interval
  cor.coeff.args = list(method = "spearman", label.sep = "\n"),
  xlab = "Dominance ranks of cows in Feed Bin High THI group",
  ylab = "Dominance ranks of cows in Feed Bin Low THI group"
) + labs(caption = "Figure 4.5: Correlation between the dominance ranks
          at Feed Bin under High THI vs under Low THI")

```