

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

Helen Chen, Annie Wang, Vivian Zhu

March 26, 2023

Summary

The social hierarchy of individuals is one of the most important concepts in the study of animal social behavior. In this study, we investigate the social hierarchy among a dynamic group of 48 dairy cows as an approach to improve herd management and animal welfare. 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 rankings were based on data recorded at feed bins because of the abundance of data. However, little is known about the reliability of estimating the social dominance hierarchy based on cows' behaviors around water bins. This research study aims to check whether the data recorded from the water bin is as reliable as the data recorded from the feed bin. Also, the environment, especially the temperature and humidity, will affect the actions of cattle. 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 data from water bins. By conducting the Elo rating algorithm, Spearman's rank correlation and VAH algorithm, we discover that rankings computed based on water bins' data have only moderate correlation with the rankings 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, THI actually seems to affect the dominance measure at water bins.

1. Introduction

This study aims to investigate how the involvement of dairy cows in agonistic interactions (replacements) among feed bins and water bins are associated with their social hierarchy in the herd. 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, the competition for water is more likely to increase, which affects the dominance measure. This report will mention the data description, statistical methods

used to test the two hypotheses, test results, conclusions and further discussions about the tests.

2. Data Description and Collection

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 dataset, there are 159 cows in total.

In each replacement, the actor cow came and took the bin that the reactor cow is originally at. The IDs of cows involved in replacements were recorded by electronic bins using cows' ear tag radiofrequency identification (ID). Temperature humidity index (THI) was calculated based on the air temperature and humidity index, which are recorded using HOBO temperature humidity loggers (Onset Computer Corporation, USA) every 5 minutes. For each hour, the average, minimum, and maximum THI was computed to provide a better understanding of how THI varied.

This is an observational study since no treatment is 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.

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

Variable	Description
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 2.1: Variables

2.1 THI Threshold Adjustment

Although our client’s project proposal suggested a THI level threshold of 75, we made some adjustments because of the limit of the data size. The fact is that the number of replacements needs to be at least 15-20 times the number of cows in the group for the Elo rating algorithm to work. Consequently, given the fact that there were 159 cows observed in the data set and with the help of relevant materials posted by the University of Minnesota (2020), we finalized the THI threshold at 65.

Group	Threshold = 75	Threshold = 65
Feed bins & high THI(THI_MAX \geq Threshold)	6303	44992
Feed bins & low THI(THI_MAX<Threshold)	165319	126630
Water bins & high THI(THI_MAX \geq Threshold)	403	2595
Water bins & low THI(THI_MAX<Threshold)	8880	6688

Table 2.1.1 Number of replacements in each group under different THI thresholds

3. Statistical Methods

3.1 Dominance Scores Calculation (Elo rating)

Dominance scores are used to capture individual cows’ dominant behavior and are later used to establish the dominance rank for the Spearman’s rank correlation test. We utilized the Elo rating algorithm (using the R package EloRating) in our analysis to assign individual cow’s 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 be 20 to account for having disproportionally large effects of single replacements. For example, if one cow (actor) replaced another cow (reactor), this one replacement will 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, accordingly, 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

Spearman's rank correlation compares the ranking between two variables for every subject and analyzes the level of association between them. Spearman's rank correlation does not require assumptions about the distribution of variables. The assumption of this test is 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.

3.3 Fisher's z-test

Fisher's z-test first transforms the correlation coefficient to z scores. Then it computes the p-value to test if the difference between two correlation coefficients ρ_1 and ρ_2 are statistically significant.

3.4 First Hypothesis

For the first comparison test (first row in table 3.5.1), we divided data into two groups according to their bin type (either feed bin or water bin) in order to address the first hypothesis (i.e. replacements at the water bins are as reliable as replacements at the feed bins in measuring the dominance hierarchy). For each group, we calculated the Elo rating score and ranked them from the highest to lowest to satisfy the assumption of Spearman's rank correlation. Then, we formed a null hypothesis: there is no correlation between the two ranking datasets, indicating a strong dissimilarity between ranks. The Spearman's rank correlation was calculated and the p-value was used to decide whether to reject the null hypothesis or not.

3.5 Second Hypothesis

For the remaining comparison tests (second row to fifth row in table 3.5.1), we further divided the two groups into four subgroups based on the THI of replacements in order to examine the second hypothesis (i.e. when THI reaches a high threshold (65), the competition for water is more likely to increase, which affects the dominance measure). As shown in table 3.5.1, we compared the feed bins & high THI data with water bins & high THI data to form the second comparison test. The third comparison test was for the two subgroups under the low THI level. Comparison test four and five compared the high THI and low THI data at water bins and feed bins. After dividing the dataset, we calculated the Elo rating score for each group and ranked them from the highest to lowest. For each comparison test, we did a Spearman’s rank correlation and tested the same null hypothesis: there is no correlation between the two ranking datasets. Then, we calculated Spearman’s rank correlation and used the p-value to decide whether to reject the null hypothesis or not.

	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 3.5.1: Pairwise comparison tests

3.6 VAH algorithm

As introduced in the Elo rating algorithm section, the maximum number of points that were gained or lost at each replacement is a function of k . According to previous studies in this field (cite the paper), 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 the first comparison group, we sampled observations based on the size of the water bins group.

2. For the remaining ones, since we're doing pairwise comparisons between four subgroups, the smallest group size, which is the one for the water bins & high THI group, was used for sampling.
3. Then we calculate Elo rating scores and run 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 ten times and used the average correlation for later analysis.

4. Results

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. Furthermore, we used Test 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 (ρ)
Test 1	Feed bins	Water bins	.59
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 4.1: Spearman's rank correlation test result 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 was a significant positive association between the two groups, ($\rho_{\text{bar}} = .59$, $p < .05$ for all 10 trials of sampling analysis). The strength of the correlation is moderate between the two groups. Thus, the relationship between dominance scores estimated from feed bins and the ones estimated from water bins is not strong.

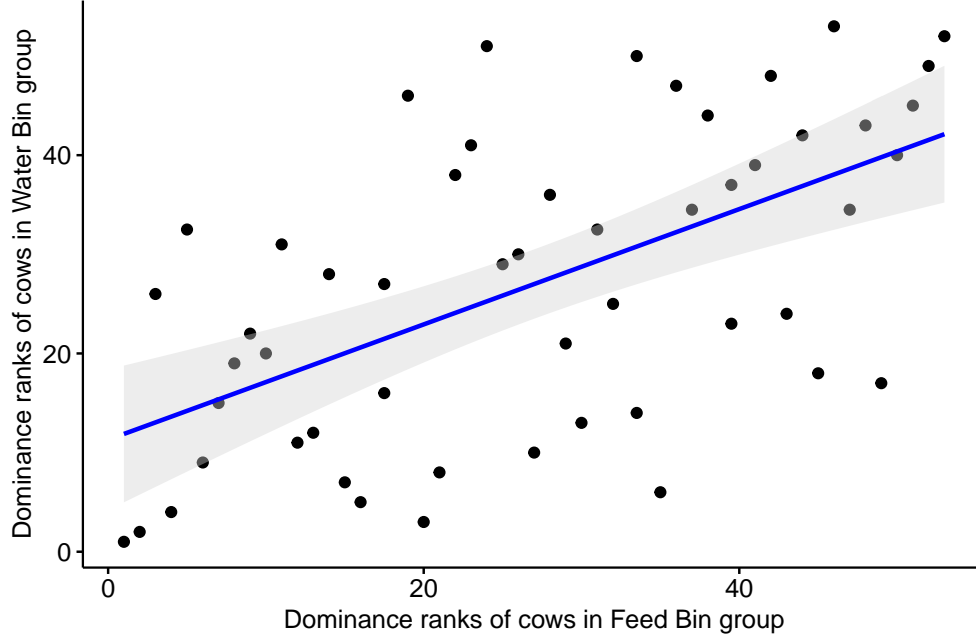


Figure 4.1: Correlation between the dominance ranks at Feed Bin and Water Bin (Overall)

The results of the two Spearman correlation tests (Test 2 & 3) show the association between dominance scores estimated from feed bins and water bins is significantly positive, (Test 2: $\rho_{\text{bar}} = .52$; Test 3: $\rho_{\text{bar}} = .58$; Both tests: $p < .05$ for all 10 trials of sampling analysis). It is key to note that the association in dominance measures between water and feed bins is slightly stronger under low THI levels. The result of Fisher's z-test ($p = .517 < .05$) demonstrates that there is insufficient evidence suggesting the two correlations are significantly different. Thus, the association between dominance scores estimated from feed bins and water bins is still not strong regardless of the THI levels.

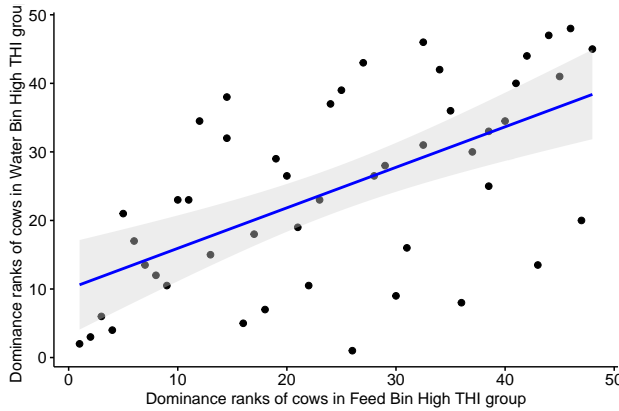


Figure 4.2: Correlation between the dominance ranks at Feed Bin and Water Bin under High THI

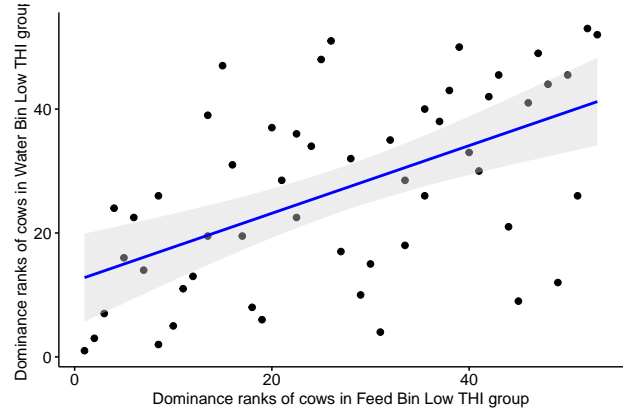


Figure 4.3: Correlation between the dominance ranks at Feed Bin and Water Bin under Low THI

We wanted to determine the impact of THI on the dominance hierarchy at water bins compared to the impact of THI on the dominance hierarchy measured at feed bins. From the results of the two Spearman correlation tests (Test 4 & 5), both tests indicate a significantly positive association between dominance scores estimated under high THI levels and the ones estimated under low THI levels (Test 4: $\rho_{\text{bar}} = .55$; Test 5: $\rho_{\text{bar}} = .83$; Both tests: $p < .05$ for all 10 trails of sampling analysis). The dominance scores estimated at feed bins have a much stronger correlation, meaning the feed bin's dominance score measured under different THI levels tends to be more similar. In another word, dominance measurements at the water bin seem to be less consistent under high and low THI levels. The result from Fisher's z-test ($p < .05$) shows the difference in the correlations between Tests 4 and 5 is statistically significant, which further consolidates the point that THI levels have a greater impact on dominance measures at water bins.

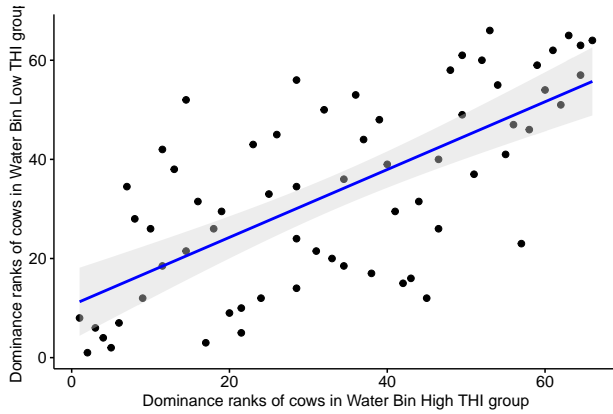


Figure 4.4: Correlation between the dominance ranks at Water Bin under High THI vs under Low THI

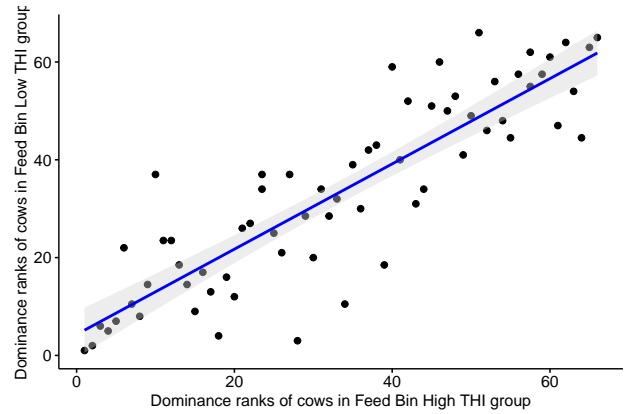


Figure 4.5: Correlation between the dominance ranks at Feed Bin under High THI vs under Low THI

5. Conclusions

In this report, we discussed how we utilized the Elo rating algorithm to calculate cow's dominance scores. In addition, we integrated a new iterative approach (i.e., VAH algorithm) to perform Spearman's rank correlation with additional results from Fisher's z-test 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 Fisher's z-test, the THI levels do have a more notable influence on the dominance scores measured at water bins. Thus the results from the test do suggest the client's second hypothesis that the structure of the dominance hierarchy at water bins will be affected by THI levels. As briefly mentioned in the previous section, we lowered the THI threshold to 65, given the current outcome of the analysis, we could anticipate similar discoveries when the THI threshold increases.

6. Further Discussion

6.1 Optimize k

Before designing the VAH algorithm for this study, we actually adopted a different approach to balance out the sample size for each group and the maximum number of points that can be transferred in each replacement. Again, the latter value depends on the parameter k in the Elo rating algorithm, which can be manually set. Since dividing the dataset using both bin types and THI threshold led us to several groups of different sizes, using the same k across all comparison groups would widen the range of final dominance scores for groups with relatively large sample sizes, resulting in potential bias and inconsistency when calculating Elo rating scores. To avoid this issue, our goal is to optimize k for each group to account for the sample size difference. The rationale for this approach is to maximize the log-likelihood of winning probabilities of the actor cows. There are the results:

Group	optimized k
Feed bins	6
Water bins	20
Feed bins & high THI	8
Feed bins & low THI	8
Water bins & high THI	31
Water bins & low THI	23

Table 6.1.1 Optimized k values for each subgroup

However, after discussing with our client, we revised the algorithm to the present version, which was discussed in the methods section, due to the fact that the maximum log-likelihood approach needs further justification and we are constrained in time.

6.2 The order of replacements

Another problem that we encountered is that there can be more than one replacement happened for an individual cow during a given hour. Since we only had the hourly timestamp, it was not possible to determine the relative order. The reason why this matter was that, 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 6.2.1: 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 cows' dominance hierarchy. Therefore, we proceeded 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_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")

```