DAR F23 Hockey Analytics

Hockey Analytics

Jeff Jung

2023-11-16

Contents

Weekly Work Summary

NOTE: Follow an outline format; use bullets to express individual points.

- RCS ID: jungj6
- Project Name: Hockey Analytics
- Summary of work since last week
  - Cluster analysis of continuous and categorical variables using pheatmap
  - Improving categorization of categorical variables by looking at density plots
  - Omitting outliers
  - Heatmap of clusters and outcomes
- NEW: Summary of github issues added and worked
  - No issues
- Summary of github commits
  - Added assignment06 to dar-jungj6
- List of presentations, papers, or other outputs
- List of references (if necessary)
- Indicate any use of group shared code base
- Indicate which parts of your described work were done by you or as part of joint efforts
- **Required:** Provide illustrating figures and/or tables

Personal Contribution

- Clearly defined, unique contribution(s) done by you: code, ideas, writing...
- Include github issues you've addressed

- Improved categorization
- Omitting outliers

Analysis: pheatmap of clusters

Question being asked

What do the correlations between clusters of continuous variables and cluters of categorical variables look like?

Data Preparation

I took the original shots data and categorized shots data and droppd variables that won't be used for cluster analysis. Then I converted numerical variables in categorical data into factors. Finally, I used kmeans for cluster method and put them into a matrix for pheatmap.

```
# Include all data processing code (if necessary), clearly commented

# Load required library
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

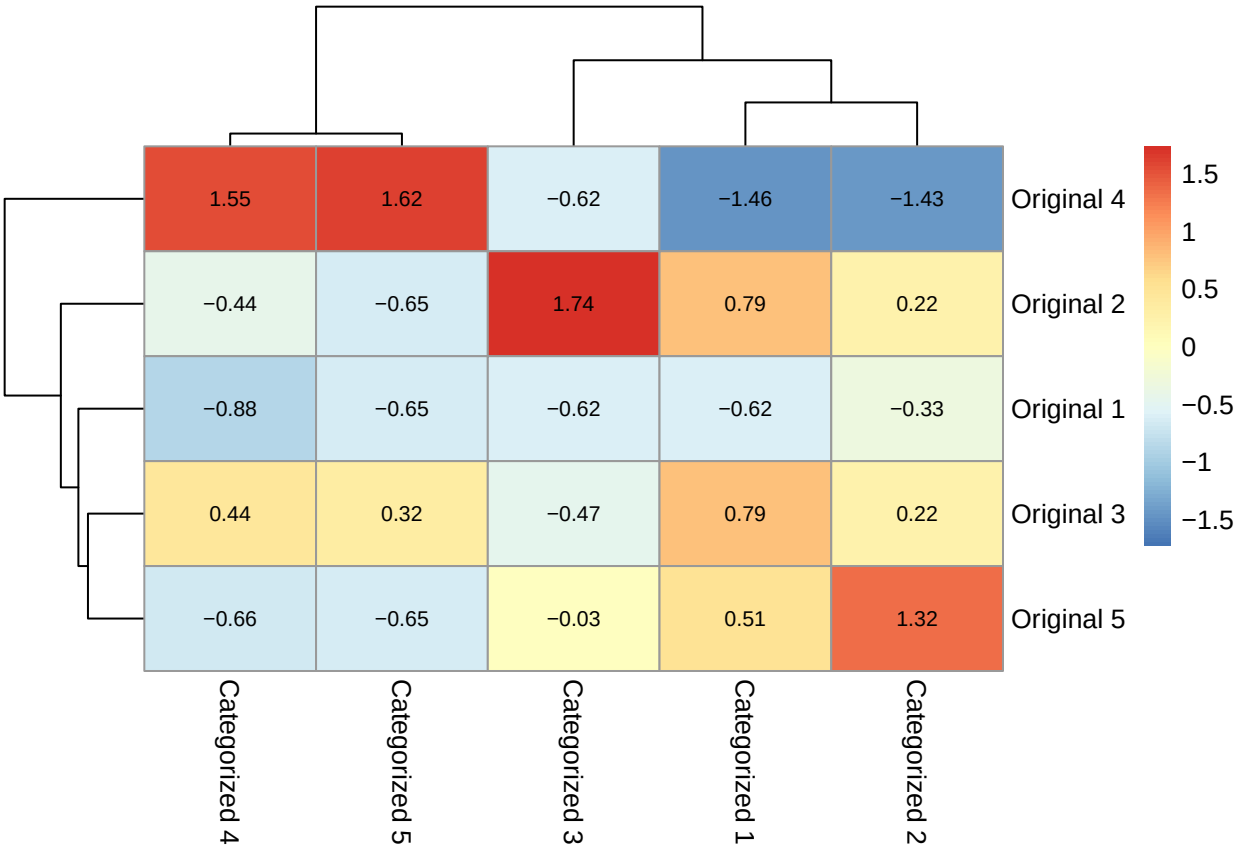
```
library(pheatmap)
library(dplyr)

# Load datasets
shots <- readRDS('shots_stats_goal.df.Rds')
catshots <- readRDS('categorized_shots_stats_goal.df.Rds')

# Add a binary goal variable
shots <- shots %>%
  mutate(goal_binary = as.integer(outcomes.goal == 2))
shots$goal_binary <- as.factor(shots$goal_binary)

# Dropping some variables necessary so that the data can be used for kmeans cluster without categorical
catshots <- catshots %>%
    select(- puckDist, - puckAngle, - puckSpeed, - shooterSpeed, - goalieDist, - goalieAngle, - posTime
org_shots <- shots %>%
    select(- closestDef, - shotOutcome, - outcomes.goal)

# Convert categorized variable as factors
catshots[sapply(catshots, is.numeric)] <- lapply(catshots[sapply(catshots, is.numeric)], as.factor)

# Making models using k means
org_model <- kmeans(org_shots, centers = 5)
cat_model <- kmeans(catshots, centers = 5)
```

```r
# Get cluster assignments
org_cluster <- org_model$cluster
cat_cluster <- cat_model$cluster

# Create a confusion matrix
conf_mat <- table(org_cluster, cat_cluster)
```

Analysis: Methods and results

The confusion matrix is prepared from the data preparation. I just need to put it in the pheatmap function. I also named the columns and rows for classification.

```r
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.

# Modify row and column names
rownames(conf_mat) <- paste("Original", 1:5)
colnames(conf_mat) <- paste("Categorized", 1:5)

# Plot the confusion matrix as a heatmap
pheatmap(conf_mat,
         scale = "column",
         display_numbers = TRUE,
         number_color = "black",
         fontsize_number = 8)
```



Discussion of results

The higher the number is or the darker the cell is, there is a higher correlation between the two clusters. We can see that of categorized cluster has a highly correlated cluster to the original cluster especially in one cell, which is exactly what we want from cluster analysis.

Analysis: Omitting outliers and improving categorization

Question being asked

Some data might not be reliable due to outliers, and putting every data into the thirds might not work due to the nature of distribution of particular feature. So I am trying to improve the model by omitting the outliers and changing the number of categories for each variable by looking at the density plots.

Data Preparation

I loaded the original shots data and added binary goals variable to plot the density plots. Then I dropped some variables that I will not use for cluster analysis. Then I made a function that will detect outliers and remove them. I then applied this function to variables that will be used for cluster analysis.

```
# Include all data processing code (if necessary), clearly commented

# load datasets
shots <- readRDS('shots_stats_goal.df.Rds')

# Creates a variable goal_binary that contains binary values 0 for save 1 for goal
shots <- shots %>%
  mutate(goal_binary = as.integer(outcomes.goal == 2))
```

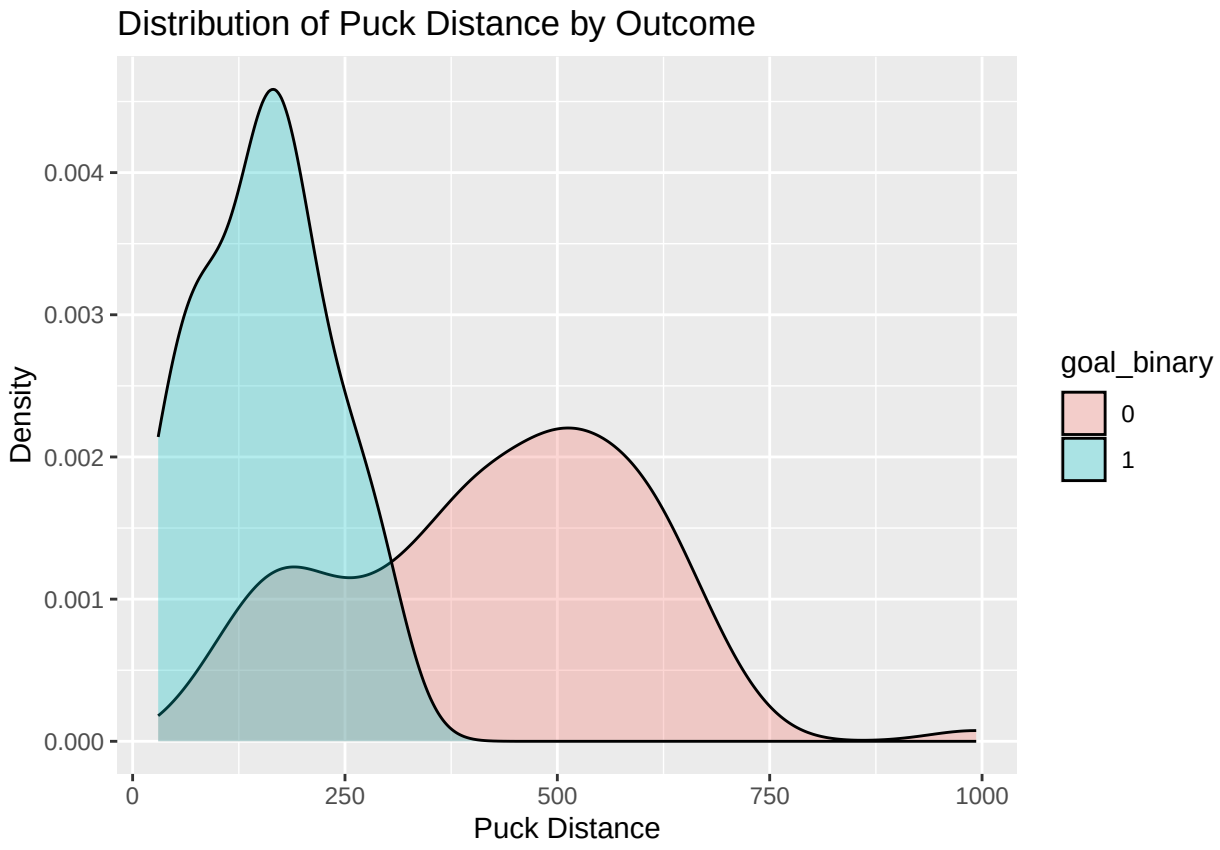
```r
# A dataset that will be used for the analysis 4
shots_4 <- shots

# Drop not needed variables
shots <- shots %>%
    select(- closestDef, - shotOutcome, - outcomes.goal, - NumOffense, -NumDefense, - rightHanded)

# create detect outlier function
detect_outlier <- function(x) {

    # calculate first quantile
    Quantile1 <- quantile(x, probs=.25)

    # calculate third quantile
    Quantile3 <- quantile(x, probs=.75)

    # calculate inter quartile range
    IQR = Quantile3-Quantile1

    # return true or false
    x > Quantile3 + (IQR*1.5) | x < Quantile1 - (IQR*1.5)
}

# create remove outlier function
remove_outlier <- function(dataframe,
                           columns=names(dataframe)) {

    # for loop to traverse in columns vector
    for (col in columns) {

        # remove observation if it satisfies outlier function
        dataframe <- dataframe[!detect_outlier(dataframe[[col]]), ]
    }
}

# Make an another dataset without outliers
clean_shots <- shots
clean_shots$goal_binary <- as.factor(clean_shots$goal_binary)

# Select the columns where you want to remove outliers
remove_outlier(clean_shots, c('puckDist', 'puckAngle', 'shooterSpeed','goalieDist','goalieAngle', 'posT
```

Analysis: Methods and Results

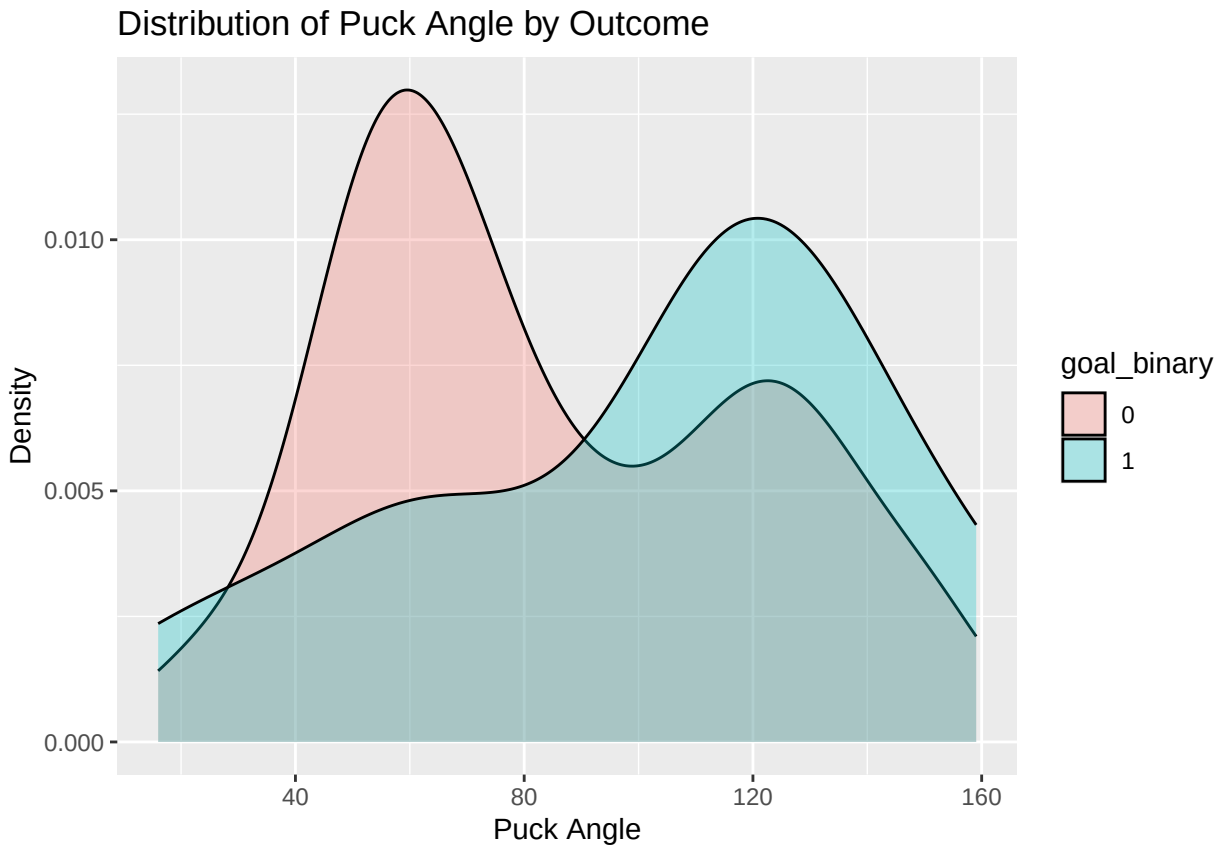
I examined density plots to find out how many categories I would need for each continuous variables.

```r
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.
```

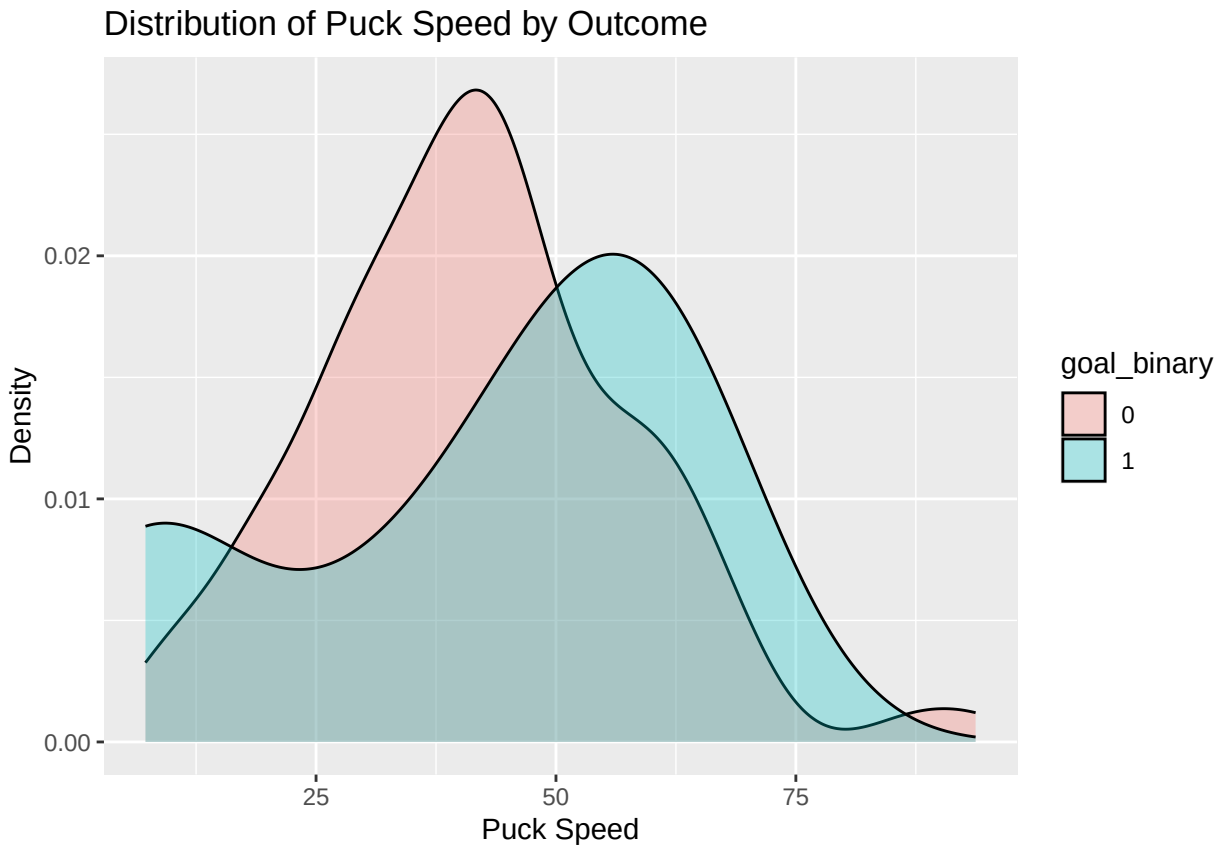
```r
# Plot the data without outliers
ggplot(clean_shots, aes(x = puckDist, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Puck Distance by Outcome",
       x = "Puck Distance",
       y = "Density")
```



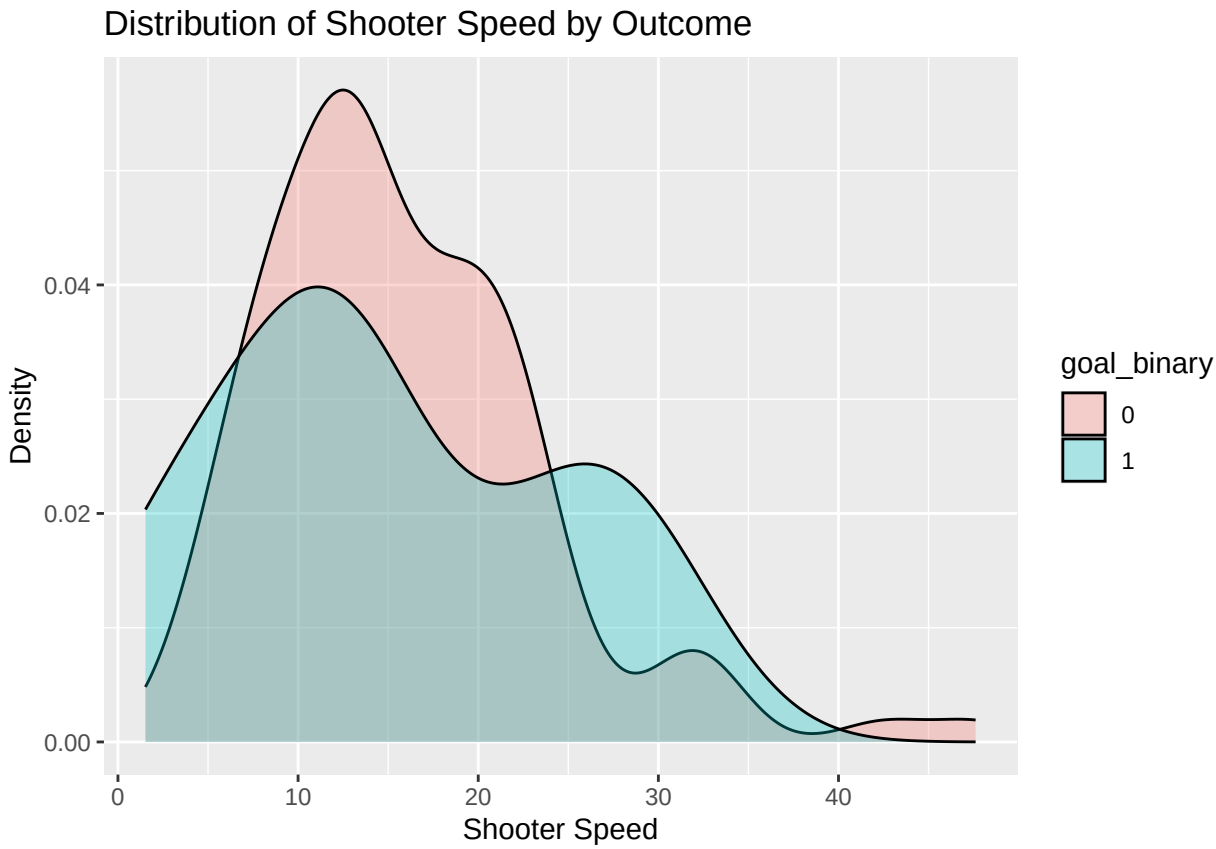

```r
ggplot(clean_shots, aes(x = puckAngle, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Puck Angle by Outcome",
       x = "Puck Angle",
       y = "Density")
```



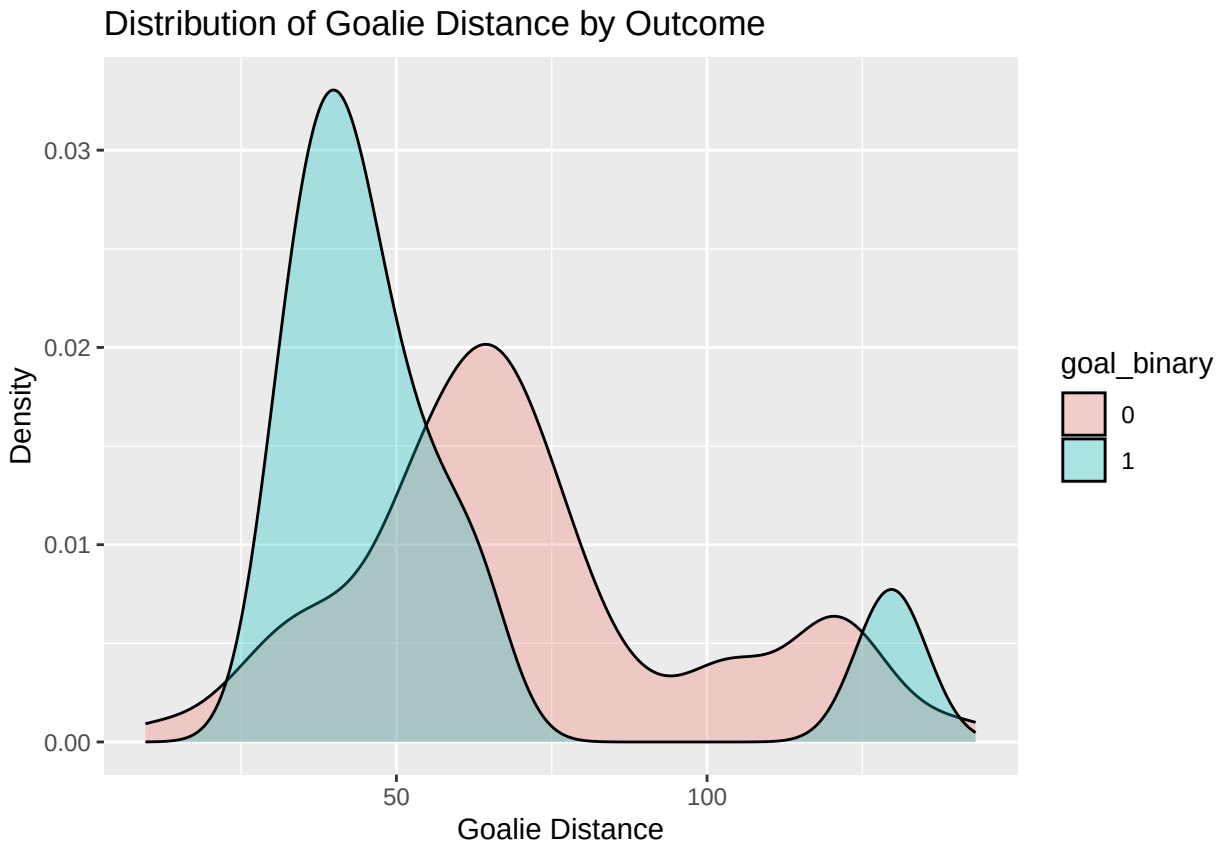

```
ggplot(clean_shots, aes(x = puckSpeed, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Puck Speed by Outcome",
       x = "Puck Speed",
       y = "Density")
```



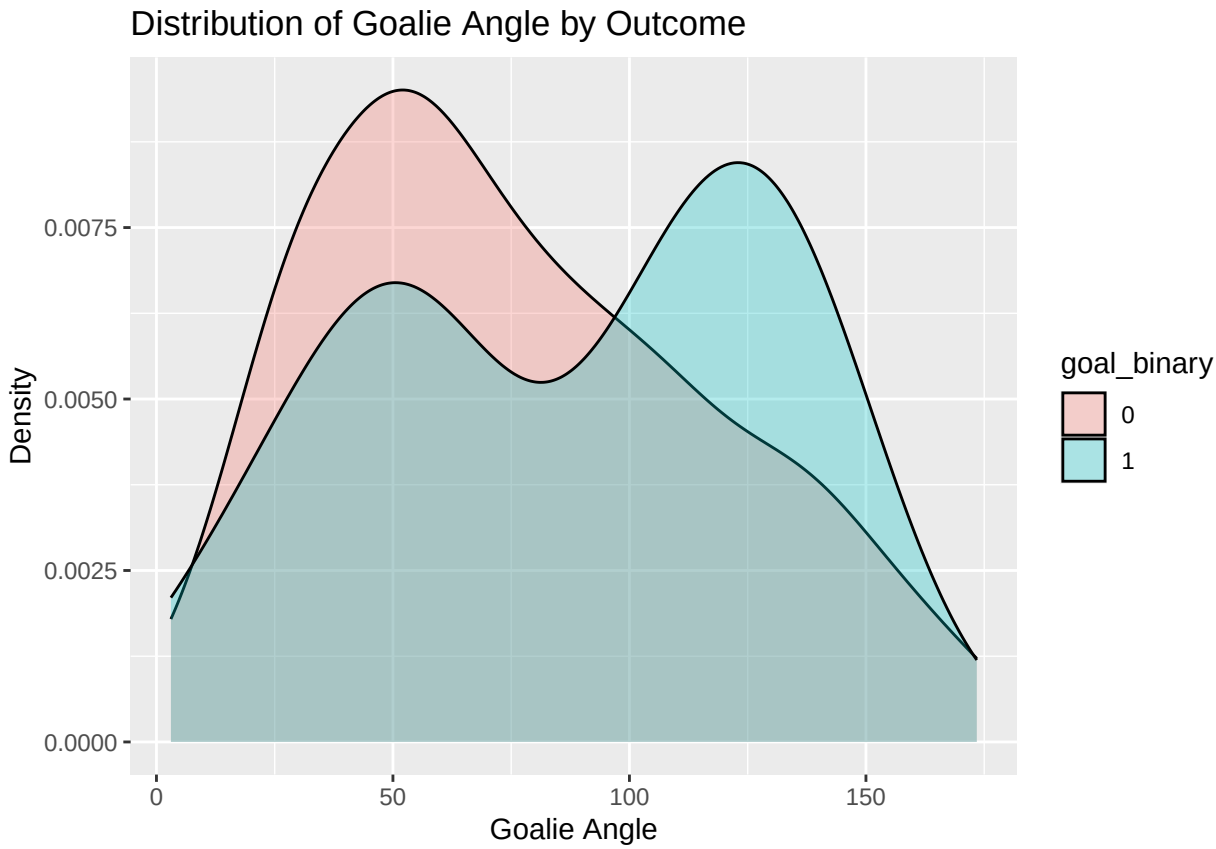

```
ggplot(clean_shots, aes(x = shooterSpeed, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Shooter Speed by Outcome",
       x = "Shooter Speed",
       y = "Density")
```



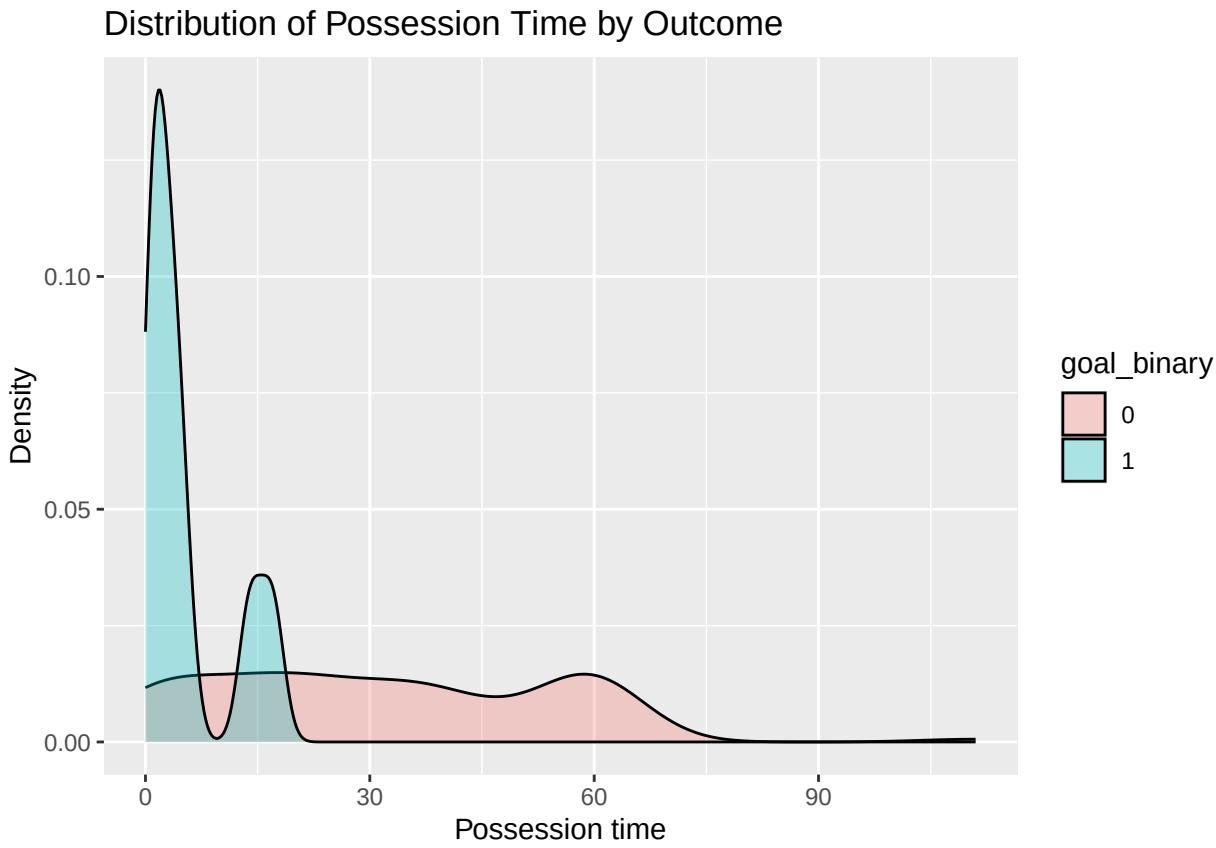

```
ggplot(clean_shots, aes(x = goalieDist, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Goalie Distance by Outcome",
       x = "Goalie Distance",
       y = "Density")
```



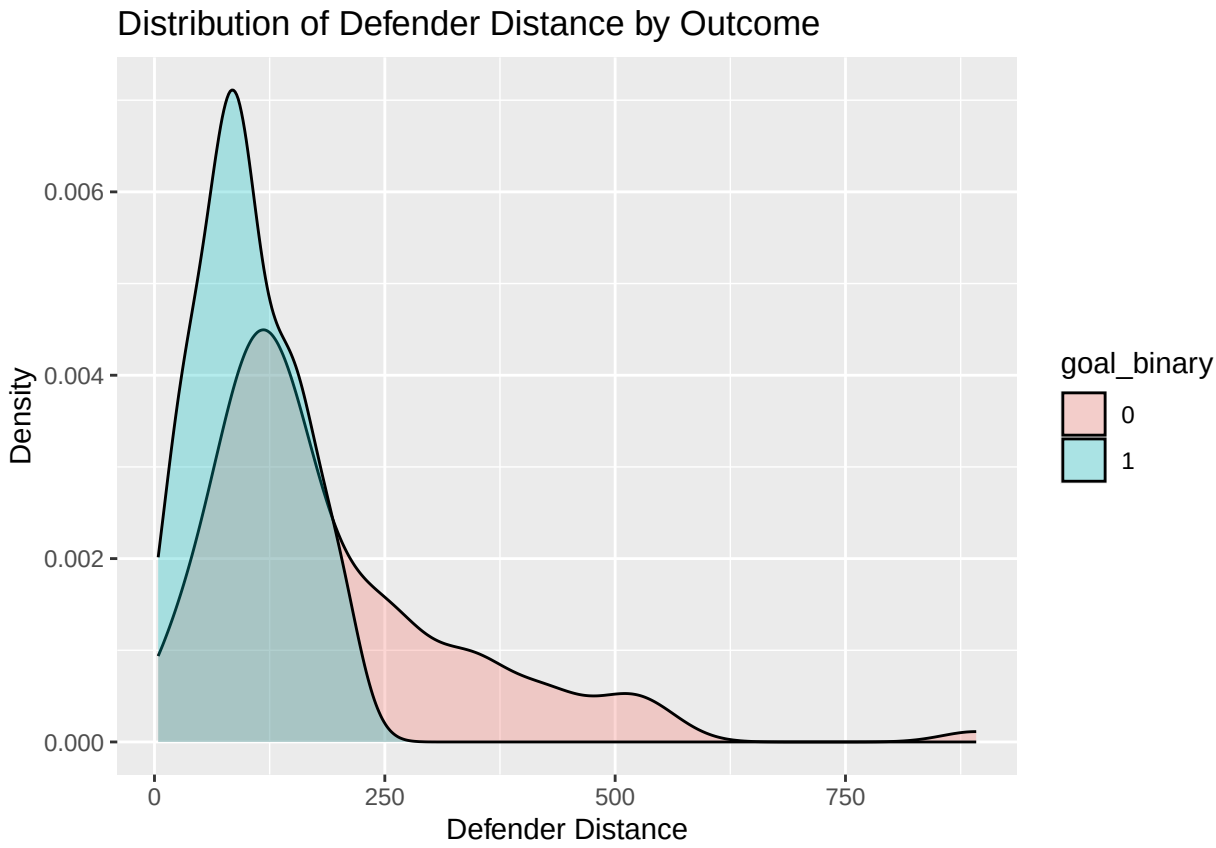

```
ggplot(clean_shots, aes(x = goalieAngle, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Goalie Angle by Outcome",
       x = "Goalie Angle",
       y = "Density")
```



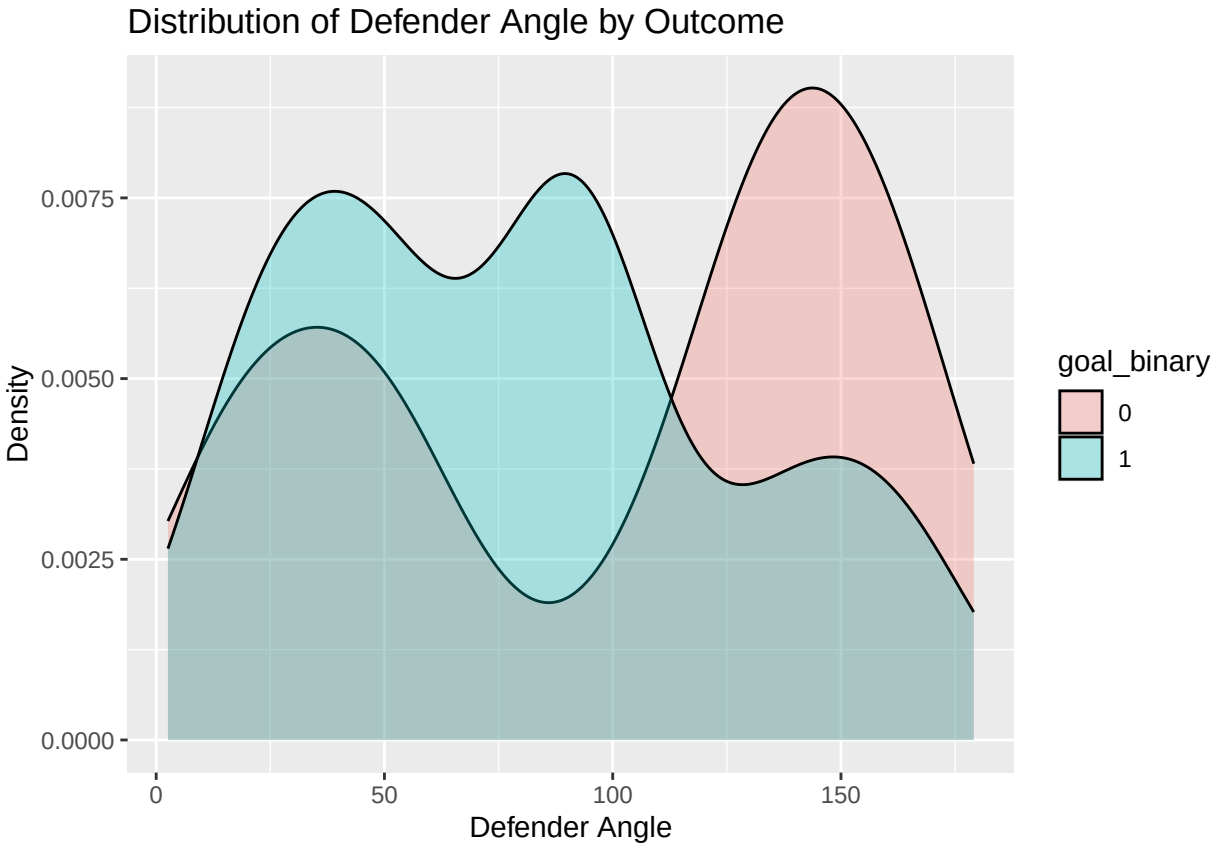

```
ggplot(clean_shots, aes(x = posTime, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Possession Time by Outcome",
       x = "Possession time",
       y = "Density")
```




```
ggplot(clean_shots, aes(x = defDist, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Defender Distance by Outcome",
       x = "Defender Distance",
       y = "Density")
```




```
ggplot(clean_shots, aes(x = defAngle, fill = goal_binary)) +
  geom_density(alpha = 0.3) +
  labs(title = "Distribution of Defender Angle by Outcome",
       x = "Defender Angle",
       y = "Density")
```



Discussion of results

After removing outliers, we can analyze the density plots without outliers that will could potentially mislead the density plots. Now, we can look at the density plots and determine how many categories we will need for each data based on the shape of the density plots. For puck distance density plot, we can see that there are two areas where most of the points are concentrated at. So we can use two categories for this variable. Similarly, for puck angle, we can use 2 categories followed by 3 for puckspeed, 3 for shooter speed, 3 for goalie distance, 2 for goalie angle, 2 for possession time, 2 for defender distance, and 3 for defender angle.

Analysis: Recategorization and plotting the heatmap again with better data

Question being asked

With the analysis done in the last section, I am able to better categorize the continuous variables. I manually selected fixed points for categorization and made new categorical variables with different number of categories for each variable depending on the shape of the density plot. Then I dropped continuous variables from the data that contains both categorical and continuous variables. Also, categorical data were converted to factors as they should be. Then I found clusters for continuous and categorical data using kmeans and put them into a matrix for pheatmap.

Data Preparation

This time, I will repeat what I did for the first analysis but without the categorical data that existed in the original data. Omit outliers and manual catagorization by examining density plots of continuous variables.

```
# Include all data processing code (if necessary), clearly commented

# Make an another dataset for categorical data
```

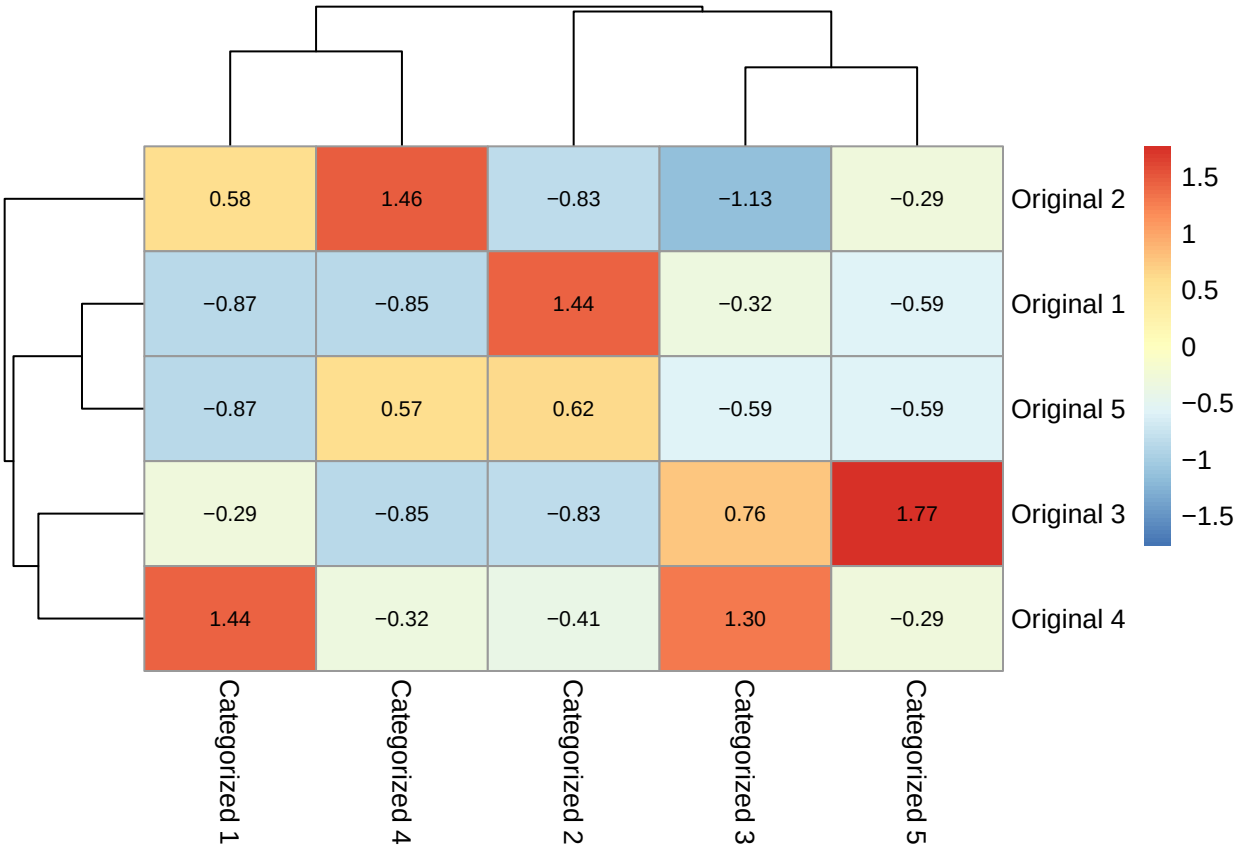
```r
clean_catshots <- clean_shots

# Manual selection for categories based on the shape of density plots above
puckDist_q <- c(350)
puckAngle_q <- c(90)
puckSpeed_q <- c(20, 46)
shooterSpeed_q <- c(16, 24)
goalieDist_q <- c(60, 74)
goalieAngle_q <- c(90)
posTime_q <- c(22)
defDist_q <- c(250)
defAngle_q <- c(60, 120)

# Create categorical variables
clean_catshots <- clean_catshots %>%
  mutate(puckSpeedCategory = case_when(
    puckSpeed <= puckSpeed_q[1] ~ 0,
    puckSpeed <= puckSpeed_q[2] ~ 1,
    TRUE ~ 2
  ))
clean_catshots <- clean_catshots %>%
  mutate(puckAngleCategory = case_when(
    puckAngle <= puckAngle_q[1] ~ 0,
    TRUE ~ 1
  ))
clean_catshots <- clean_catshots %>%
  mutate(puckDistCategory = case_when(
    puckDist <= puckDist_q[1] ~ 0,
    TRUE ~ 1
  ))
clean_catshots <- clean_catshots %>%
  mutate(posTimeCategory = case_when(
    posTime <= posTime_q[1] ~ 0,
    TRUE ~ 1
  ))
clean_catshots <- clean_catshots %>%
  mutate(goalieDistCategory = case_when(
    goalieDist <= goalieDist_q[1] ~ 0,
    goalieDist <= goalieDist_q[2] ~ 1,
    TRUE ~ 2
  ))
clean_catshots <- clean_catshots %>%
  mutate(shooterSpeedCategory = case_when(
    shooterSpeed <= shooterSpeed_q[1] ~ 0,
    shooterSpeed <= shooterSpeed_q[2] ~ 1,
    TRUE ~ 2
  ))
clean_catshots <- clean_catshots %>%
  mutate(goalieAngleCategory = case_when(
    goalieAngle <= goalieAngle_q[1] ~ 0,
    TRUE ~ 1
  ))
clean_catshots <- clean_catshots %>%
```

```
  mutate(defDistCategory = case_when(
    defDist <= defDist_q[1] ~ 0,
    TRUE ~ 1
  ))
clean_catshots <- clean_catshots %>%
  mutate(defAngleCategory = case_when(
    defAngle <= defAngle_q[1] ~ 0,
    defAngle <= defAngle_q[2] ~ 1,
    TRUE ~ 2
  ))

# Drop continous variables for categorical dataset
clean_catshots <- clean_catshots %>%
    select(- puckDist, - puckAngle, - puckSpeed, - shooterSpeed, - goalieDist, - goalieAngle, - posTime

# Convert categorized variable as factors
clean_catshots[sapply(clean_catshots, is.numeric)] <- lapply(clean_catshots[sapply(clean_catshots, is.nu

# Making models using k means
org_model <- kmeans(clean_shots, centers = 5)
cat_model <- kmeans(clean_catshots, centers = 5)

# Get cluster assignments
org_cluster <- org_model$cluster
cat_cluster <- cat_model$cluster

# Create a confusion matrix
conf_mat <- table(org_cluster, cat_cluster)
```

Analysis methods used

The matrix is prepared from the data preparation section. I named rows and columns and plotted the matrix in heatmap.

```
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.

# Modify row and column names
rownames(conf_mat) <- paste("Original", 1:5)
colnames(conf_mat) <- paste("Categorized", 1:5)

# Plot the confusion matrix as a heatmap
pheatmap(conf_mat,
         scale = "column",
         display_numbers = TRUE,
         number_color = "black",
         fontsize_number = 8)
```



Discussion of results

The result is somewhat similar to the heatmap in analysis 1, and the meanings of the color and the number are the same. The difference is not apparent in the heatmap, but this gives more accurate correlations because outliers have been omitted and better categorization was used. Overall, most clusters have roughly one correlating cluster to each other, which means that this heatmap is statistically significant.

Analysis: 4 types of goals vs clusters

Question being asked

Now that we have analyzed how two types of clusters are correlated to each other, I wanted to find out how the each original cluster (continuous) is correlated with the each of the outcomes.

Data Preparation

The data preparation is very simple. We have created the clusters with the original shots data, so I just took the same clusters and put it into a matrix with the outcomes of the goals.

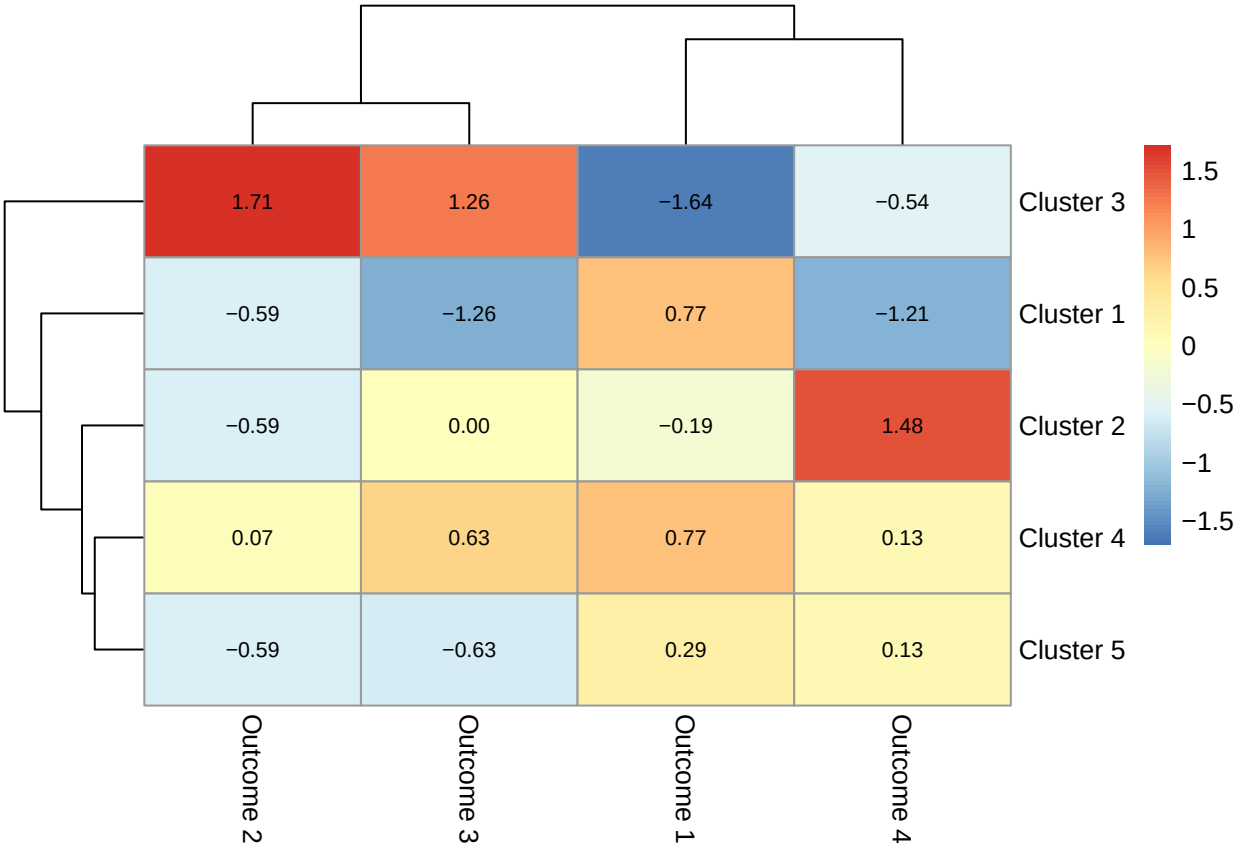
```
# Make a matrix with the clusters from the orirignal data and 4 outcomes of shots
conf_mat <- table(org_cluster, shots_4$outcomes.goal)
```

Analysis methods used

I used pheatmap for plotting the heatmap. Also, I labeled the rows and columns.

```
# Plot the confusion matrix as a heatmap
rownames(conf_mat) <- paste("Cluster", 1:5)
colnames(conf_mat) <- paste("Outcome", 1:4)
```

```
pheatmap(conf_mat,
         scale = "column",
         display_numbers = TRUE,
         number_color = "black",
         fontsize_number = 8)
```



Discussion of results

If the cell where each outcome is dark red or the number in the cell is high, that means that outcome is likely in that cluster. Here, outcome 1 is defender block, outcome 2 is goal, outcome 3 is save, and outcome 4 is miss. So, by looking at each cell, we can see how each cluster is correlated with each of the outcome.

Summary and next steps

I will work with Liebin to find a way to incorporate my heatmap with Liebin's app. Also, I will talk to Professor Bennett on Monday to figure out if there is anything else I need to work on.