STAT 847: Final Project

DUE: Friday April 19, 2024 by 11:59pm Eastern

```
library(plyr)
library(ggplot2)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(rpart)
library(rpart.plot)
library(FactoMineR)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rsample)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(RColorBrewer)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(leaps)
library(png)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library("factoextra")
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
dat = read.csv("./Big-Data-Cup-2024-main/BDC_2024_Womens_Data.csv")
dat$Period <- as.character(dat$Period)
```

1) **MUST BE INCLUDED** Describe and justify two different topics or approaches you might want to consider for this dataset and task. You don't have to use these tasks in the actual analysis.

In this project, I'm aiming to use the following techniques to do the analysis:

Regression Analysis: Use regression to predict game outcomes. player/team performance, game conditions based on the information provided in the data set, such as events, goals, clock, period and details.

Classification Algorithms: Use logistic regression, decision trees, random forests, classification trees to classify game outcomes. We can also classify event based on the coordinates informations.

Clustering: Use K-means clustering can help identify similar players based on their performance.

Time Series Analysis: As in the data set there're some columns includes time aspects, Time series techniques can be helpful to analyze trends and make predictions.

Two major topics:

Topic 1: Individual player performance In this exploration of individual player performance, we delve into the intricate details of each player's contributions throughout the game. The analysis needs to focus on every event in which they participated. We need to focus on the frequency of each player's involvement in various event types, ranging from shots and plays to takeaways, puck recoveries, dump-ins, dump-outs, zone entries, faceoffs, and penalties. By dissecting their participation in these events, we gain information about how much they contribute to the game. Also, we can develop performance metrics for each event type to accurately assess each player's impact. Metrics can be shoting efficiency, which calculates the ratio of successful shots to total shots, providing a measure of offensive prowess. Also, we calculate defensive Impact by evaluating the frequency of takeaways and puck recoveries, shedding light on a player's defensive prowess and ability to disrupt the opponent's game plan. Through this meticulous analysis and the development of specific metrics, we aim to provide a comprehensive evaluation of each player's performance. By quantifying their contributions across various facets of the game, we gain deeper insights into their strengths, weaknesses, and overall impact on the team's performance. Furthermore, to improve our analysis, we can generate visual charts.

To approach Topic 1, we will need to make some modifications to the data set to gather information from it, So the data set will focus more on players' performance but not the each event, We will first generate some subsets which will group the data set by Players, Event, Date, Period, calculate the number of each events that each players involved in and average coordinates of this player when event happens:

```
player_dat <- dat %>% group_by(Player, Date, Event, Team, Period) %>%
  summarise(avg_x = mean(X.Coordinate), avg_y = mean(Y.Coordinate), num_event = n())
```

```
## `summarise()` has grouped output by 'Player', 'Date', 'Event', 'Team'. You can
## override using the `.groups` argument.
```

```
# head(player_dat)

# regular time
player_dat_reg <- filter(player_dat, Period != 4)


# specific Offense Performance

player_dat_off <-  dat %>%
  filter(Event %in% c("Goal", "Incomplete Play", "Play", "Shot", "Zone Entry"))%>%
  group_by(Player, Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Player', 'Team', 'Date', 'Event',
## 'Period', 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(player_dat_off)


# specific shoting Performance

player_dat_shot <-  dat %>%
  filter(Event %in% c("Goal", "Shot"))%>%
  group_by(Player, Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Player', 'Team', 'Date', 'Event',
## 'Period', 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(player_dat_shot)


# specific Passing Performance

player_dat_pass <-  dat %>%
  filter(Event %in% c("Incomplete Play", "Play"))%>%
  group_by(Player, Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Player', 'Team', 'Date', 'Event',
## 'Period', 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(player_dat_pass)


# Defense Performance

player_dat_def <- dat %>%
  anti_join(player_dat_off)%>%
  group_by(Player, Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## Joining with `by = join_by(Date, Period, Team, Player, Event, X.Coordinate,
## Y.Coordinate)`
## `summarise()` has grouped output by 'Player', 'Team', 'Date', 'Event',
## 'Period', 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(player_dat_def)
```

Task 2: Team strategy In this topic, we will focus on analyzing team performance involving different aspects of the game, including offensive and defensive actions, according to the dataset, the analysis is based on: We will start by tracking the number of shots attempted by each team, focusing on shots on goal, missed shots, and blocked shots. This data will provide valuable insights into the team's offensive strategies and their ability to create scoring opportunities. Additionally, analyzing shot locations will allow us to identify high-danger areas where the team is most effective at generating scoring chances. Moving on to passing, we will monitor successful pass attempts and puck movement to assess the team's ability to maintain possession and orchestrate offensive plays. By examining the types of passes and their accuracy, we can gain a deeper understanding of the team's passing efficiency and playmaking capabilities. In terms of puck recoveries, we will evaluate how quickly the team regains possession after a turnover. This metric will shed light on the team's resilience and determination in winning back the puck, as well as their ability to transition from defence to offence effectively. Furthermore, we will keep track of takeaways to measure the team's defensive pressure and their ability to disrupt the opponent's plays.

To approach Topic 2, we will need to make some modifications to the data set to gather information from it, So the data set will focus more on Teams' performance but not the each event, We will first generate some subsets which will group the data set by Teams (Home/Away/Team), Event, Date, Period, calculate the number of each events that each Team involved in and average coordinates of this player when event happens:

```
team_dat <- dat %>% group_by(Team, Date, Event, Period) %>% summarise(avg_x = mean(X.Coordinate), avg_y = mean(Y.Coordinate), num_event = n())
```

```
## `summarise()` has grouped output by 'Team', 'Date', 'Event'. You can override
## using the `.groups` argument.
```

```
# regular time overall
team_dat_reg <- filter(team_dat, Period != 4)


# specific Offense Performance

team_dat_off <-  dat %>%
  filter(Event %in% c("Goal", "Incomplete Play", "Play", "Shot", "Zone Entry"))%>%
  group_by(Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Team', 'Date', 'Event', 'Period',
## 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(team_dat_off)

# specific shoting Performance

team_dat_shot <-  dat %>%
  filter(Event %in% c("Goal", "Shot"))%>%
  group_by(Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Team', 'Date', 'Event', 'Period',
## 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(team_dat_shot)


# specific Passing Performance

team_dat_pass <-  dat %>%
  filter(Event %in% c("Incomplete Play", "Play"))%>%
  group_by(Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>%
  summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Team', 'Date', 'Event', 'Period',
## 'X.Coordinate'. You can override using the `.groups` argument.
```

```
# head(team_dat_pass)



# Defense Performance

team_dat_def <- dat %>%
  anti_join(team_dat_off)%>% group_by(Team, Date, Event, Period, X.Coordinate, Y.Coordinate) %>% summarise(num_event = n())
```

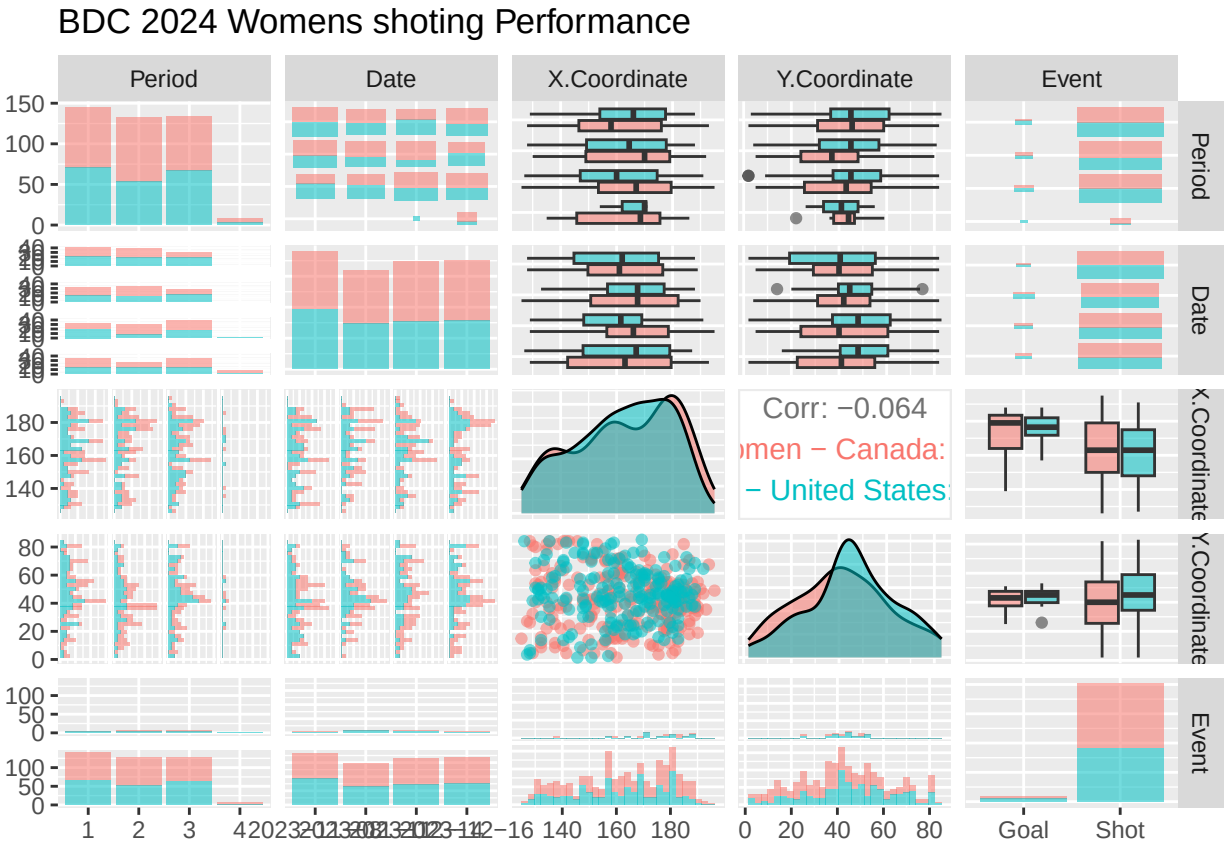
```
## Joining with `by = join_by(Date, Period, Team, Event, X.Coordinate,
## Y.Coordinate)`
## `summarise()` has grouped output by 'Team', 'Date', 'Event', 'Period',
## 'X.Coordinate'. You can override using the `.groups` argument.
```

The subsets computed above may used along with the main data set to do the analysis.

2) **MUST BE INCLUDED** Give a ggpairs plot of what you think are the six most important variables. At least one must be categorical, and one continuous. Explain your choice of variables and the trends between them.

```
pm <- ggpairs(team_dat_shot, columns = c("Period", "Date", "X.Coordinate", "Y.Coordinate","Event"),  mapping=aes(color = Team, alpha=0.1),
              cardinality_threshold = NULL,
  title = "BDC 2024 Womens shoting Performance")
pm
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In question 2, We want to use ggpairs plot to analyze the shoting performance of both teams (Team USA and Team Canada) in all four games. So we will focus on either Goal: Successful Shot attempts (goal) and Shot: Unsuccessful shot attempts (block, miss or save). As mentioned in Q1, our goal is to track the number of shots attempted by each team to show their ability to create scoring opportunities and the ability to shot (finishing). In the meantime, it also can show the opponent's goalkeeper's goalkeeping ability. We also will be able to identify the high-danger areas where both teams are most effective at creating goal opportunities.

Now let's talk about the six most important variables I will use to generate such a ggpairs plot:

X Coordinate / Y Coordinate (Continuous): The X/Y coordinate provides important information about their location (Event location) relative to offensive, defensive, or neutral zones. In this case, we will focus on the offensive. Analyzing this variable enhances our understanding of team attacking/ shoting strategies and gameplay tactics. Also, we will be able to know the area where they prefer to take a shot and may create goal opportunities.

Period (Continuous): The period explains the current period when events occur. It offers insights into game progression over time. Understanding event distribution across periods helps assess team performance and strategic adjustments throughout the game.

Event (Categorical): The event describes the type of event taking place during the game, providing a detailed overview of the current key moment. Analyzing event types facilitates the identification of patterns, trends, and critical moments in the match. We will be focusing on either goals or misses.

Team (Categorical): We need to focus on the two teams, Team USA and Team Canada, this variable enables analysis of attacking/ shoting strategies used by each team. By analyzing team-specific trends and behaviours, we will be able to gain deeper insights into their performance and competitive strategies.

Date (Categorical): There are 4 games in total in the series. Which is the following:
2023-11-08, Home Team: USA, Away Team: Canada
2023-11-11, Home Team: USA, Away Team: Canada
2023-12-14, Home Team: Canada, Away Team: USA
2023-12-16, Home Team: Canada, Away Team: USA

By using the Date variables, we will be able to see the difference in performance and attacking/ shoting strategies both teams use when they are at a home game or an away game.

Let's take a look at the gg pairs plot in order to get a better understanding of the performance of Team Canada and Team USA.

We differentiated each team by colour, with pink representing Canada and blue representing the USA. Each row in the data set corresponds to either a shot or a miss. They also indicate the number of attempts (goals or misses) each team made.

Plot (1,1) shows the number of attempts per period. It's obvious that Team Canada made more shoting attempts than Team USA across all four periods. Especially leading in periods 2 and 4 (extra time).

Plot (2,2) shows the number of attempts per game (Date). We observe that in away games (Canada's away game), Team Canada and Team USA created a similar amount of opportunities. However, in home games, Team Canada demonstrated a clear advantage in creating shoting opportunities.

Plot (1,2) and (2,1) explain attempts per period per game. Notably, Team Canada had a higher number of shoting attempts in period 2. Team USA only showed stronger offensive willingness in one or two periods in all four games. Overall, the ggpairs suggest that Team Canada maintains a more aggressive offensive strategy compared to Team USA.

A similar situation is shown in plot (5,5). It illustrates the number of goals and shots created by both teams in all games. However, although Team Canada had a higher number of attempts (shoting opportunities), Team USA still scored more goals. This can indicate a potential weakness in Team Canada's finishing abilities. The players waste a huge amount of chances.

Further details are provided in plot (5,1). It shows the number of goals and shots per period. Team Canada had a commendable performance in most of period 2. It matches the goal rate of Team USA in period 2. However, in periods 1, 3, and extra time, Team USA took the lead on the goals.

Plot (5,2) reconfirms Team Canada's advantage in creating shoting opportunities. But it also highlights a concerning trend: although Team Canada always has more shoting opportunities in nearly every game, Team Canada failed to secure any victories, ending the series with 3 losses and 1 draw. This suggests that Team USA possesses a stronger overall performance in the series. It is also possible that Canada's strategy is to play more offensively and shot whenever they can. But it doesn't have any effective effect. More aggressive can cause the opponent to have a chance to make a quick counterattack.

Beyond goal-scoring statistics, we need to also analyze the spatial distribution of attempts:

For Team USA:

X-axis analysis in column 3 indicates a preferred shoting area (global maximum) just below 180 in plots (3,3) and (5,3). Similarly, the majority of goals fall within the range of 170-190. Which is the area pretty close to the goal gate.
Y-axis plots in column 4 have a peak around 45, with corresponding goals clustered between 35-55 in plots (4,4) and (5,4). Which is in the middle y-axis, which means facing the goal gate directly.

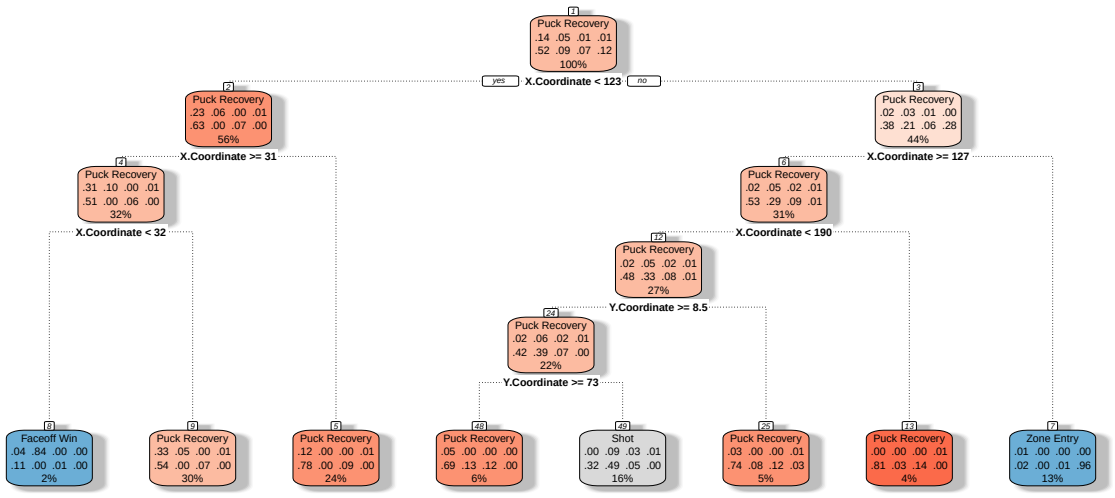
For Team Canada:

X-axis analysis in column 3 highlights multiple local maximums and a global maximum at 180, with additional peaks at 140 and 160, indicating preferred shoting positions. But most of the goals are still around 180.
The maximum of Y-axes in column 4 is also around 40 again, facing the goal gate directly. But it has a much-flattened graph.
According to this result, we can also think maybe team Canada is trying to score at a tricky angle, but barely succeeds.

In Q6, we will also having a ggplot which is about the offense performance of both team, we will be taking about their game strategies combine with ggplot graph and ggpairs.

3) **MUST BE INCLUDED** Build a classification tree of one of the six variables from the last part as a function of the other five, and any other explanatory variables you think are necessary. Show code, explain reasoning, and show the tree as a simple (ugly) plot. Show the confusion matrix. Give two example predictions and follow them down the tree.

```
# split data

dat_no_pass <- filter(dat, Event != "Play" & Event != "Incomplete Play")
dat_split <- initial_split(dat_no_pass, prop = 0.8)
dat_train <- training(dat_split)
dat_test <- testing(dat_split)

fit = rpart(Event~Period + Date+Y.Coordinate+X.Coordinate + Team, method="class", data=dat_train)


# Use pruning
pfit<- prune(fit, cp= fit$cptable[which.min(fit$cptable[,"xerror"]),"CP"])
fancyRpartPlot(pfit, main="Classification tree of Event")
```




```
pred <- predict(pfit, dat_test, "class")

# Make the confusion matrix
confusion <- table(pred, dat_test$Event)
print(confusion)
```

```
##
## pred            Dump In/Out Faceoff Win Goal Penalty Taken Puck Recovery Shot
##   Dump In/Out             0           0    0             0             0    0
##   Faceoff Win             1          15    0             0             3    0
##   Goal                    0           0    0             0             0    0
##   Penalty Taken           0           0    0             0             0    0
##   Puck Recovery         113          12    0            11           416    8
##   Shot                    2          18    1             0            38   76
##   Takeaway                0           0    0             0             0    0
##   Zone Entry              0           0    0             0             2    0
##
## pred            Takeaway Zone Entry
##   Dump In/Out          0          0
##   Faceoff Win          0          0
##   Goal                 0          0
##   Penalty Taken        0          0
##   Puck Recovery       43          1
##   Shot                 4          1
##   Takeaway             0          0
##   Zone Entry           0        106
```

```
# Print accuracy
accuracy <- sum(diag(confusion)) / sum(confusion)
print(accuracy)
```

```
## [1] 0.7037887
```

```
# Exmaple predicions

# 1
pred1 = predict(pfit, dat_test[1,], "class")

# 2
pred2 = predict(pfit, dat_test[2,], "class")
print("--Example 1--")
```

```
## [1] "--Example 1--"
```

```
print(dat_test[1, ])
```

```
##         Date              Home.Team      Away.Team Period Clock
## 1 2023-11-08 Women - United States Women - Canada      1 19:45
```

```
##   Home.Team.Skaters Away.Team.Skaters Home.Team.Goals Away.Team.Goals
## 1                 5                 5               0               0
##             Team      Player         Event X.Coordinate Y.Coordinate Detail.1
## 1 Women - Canada Renata Fast Puck Recovery            9            9
##   Detail.2 Detail.3 Detail.4 Player.2 X.Coordinate.2 Y.Coordinate.2
## 1                                                NA             NA
```

```
print(pred1)
```

```
##            1
## Puck Recovery
## 8 Levels: Dump In/Out Faceoff Win Goal Penalty Taken Puck Recovery ... Zone Entry
```

```
print("--Example 2--")
```

```
## [1] "--Example 2--"
```

```
print(dat_test[2, ])
```

```
##         Date             Home.Team      Away.Team Period Clock
## 2 2023-11-08 Women - United States Women - Canada      1 19:39
##   Home.Team.Skaters Away.Team.Skaters Home.Team.Goals Away.Team.Goals
## 2                 5                 5               0               0
##             Team              Player         Event X.Coordinate Y.Coordinate
## 2 Women - Canada Marie-Philip Poulin Puck Recovery           64           40
##   Detail.1 Detail.2 Detail.3 Detail.4 Player.2 X.Coordinate.2 Y.Coordinate.2
## 2                                                         NA             NA
```

```
print(pred2)
```

```
##            2
## Puck Recovery
## 8 Levels: Dump In/Out Faceoff Win Goal Penalty Taken Puck Recovery ... Zone Entry
```

We aim to construct a classification tree to classify one of the six variables from the previous question. We will choose the Event. Building such a classification tree assists us in classifying the potential attributes of each event.

We exclude passing actions from classification due to their high frequency, location-free, and potential similarity with unsuccessful passing. This exclusion prevents confusion in the classification tree and ensures distinct attributes for each event category.

In further analysis, we can build a classification tree to classify the passing location by teams, which allows us to find out if there's a specific location where one of the teams likes to do the passing. But in this question, We will focus solely on classifying non-passing events in order to build a more general tree.

From the classification tree, it's evident that only the X-coordinate and Y-coordinate are used. This could be due to a few reasons:

Let's consider other variables:

Date / Period: These variables represents the date and period of the game, it's not a signiciant aid for classification since all events can happens in all period/ games. The events listed all have high frequency and they are most likely going to happens anytime during the game.

Team: Any team can trigger any event. Therefore, this variable does not offer substantial value for classification purposes.

Hence, these variables are unlikely to significantly aid in the construction of our classification tree.

4) **MUST BE INCLUDED** Build another model using one of the continuous variables from your six most important. This time use your model selection and dimension reduction tools, and include at least one non-linear term.

```
# prepare data
model_dat <- dat %>% group_by(Team,Period, X.Coordinate, Event) %>% summarise(num_event = n())
```

```
## `summarise()` has grouped output by 'Team', 'Period', 'X.Coordinate'. You can
## override using the `.groups` argument.
```

```
dat_split <- initial_split(model_dat, prop = 0.8)
dat_train <- training(dat_split)
dat_test <- testing(dat_split)
```

```
lm_model <- lm(dat_train$num_event~ I(X.Coordinate^2) +., data =dat_train)

# Use Stepwise regression
step <- step(lm_model, direction="both")
```

```
## Start:  AIC=4505.98
## dat_train$num_event ~ I(X.Coordinate^2) + (Team + Period + X.Coordinate +
##     Event)
##
##                     Df Sum of Sq   RSS    AIC
## - Team               1       0.2 13896 4504.0
## <none>                           13896 4506.0
## - Period             3     330.1 14226 4565.3
## - I(X.Coordinate^2)  1     607.9 14504 4623.0
## - X.Coordinate       1     773.9 14670 4654.7
## - Event              9    8803.4 22699 5852.7
##
## Step:  AIC=4504.02
## dat_train$num_event ~ I(X.Coordinate^2) + Period + X.Coordinate +
##     Event
##
##                     Df Sum of Sq   RSS    AIC
## <none>                           13896 4504.0
## + Team               1       0.2 13896 4506.0
## - Period             3     330.0 14226 4563.3
## - I(X.Coordinate^2)  1     608.0 14504 4621.1
## - X.Coordinate       1     774.2 14670 4652.8
## - Event              9    8803.3 22699 5850.8
```

```
summary(step)
```

```
##
## Call:
## lm(formula = dat_train$num_event ~ I(X.Coordinate^2) + Period +
##     X.Coordinate + Event, data = dat_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4927  -0.9233  -0.1913   0.5373  31.5250
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.039e+00  1.826e-01  16.646  < 2e-16 ***
## I(X.Coordinate^2)     1.581e-04  1.437e-05  11.001  < 2e-16 ***
## Period2               2.313e-02  1.050e-01   0.220   0.8256
## Period3              -1.003e-02  1.060e-01  -0.095   0.9246
## Period4              -2.048e+00  2.604e-01  -7.865 5.25e-15 ***
## X.Coordinate         -3.601e-02  2.901e-03 -12.414  < 2e-16 ***
## EventFaceoff Win      6.219e+00  4.578e-01  13.584  < 2e-16 ***
## EventGoal            -3.974e-01  5.819e-01  -0.683   0.4947
## EventIncomplete Play -2.182e-01  1.693e-01  -1.289   0.1976
## EventPenalty Taken   -7.318e-01  4.436e-01  -1.650   0.0991 .
## EventPlay             8.956e-01  1.513e-01   5.918 3.65e-09 ***
## EventPuck Recovery    8.187e-01  1.526e-01   5.366 8.73e-08 ***
## EventShot             2.494e-01  2.149e-01   1.161   0.2458
## EventTakeaway        -5.294e-01  2.112e-01  -2.507   0.0122 *
## EventZone Entry       1.848e+01  4.881e-01  37.850  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.241 on 2766 degrees of freedom
## Multiple R-squared:  0.4003, Adjusted R-squared:  0.3973
## F-statistic: 131.9 on 14 and 2766 DF,  p-value: < 2.2e-16
```

```
# mean squared error
test <- data.frame(predict(step, dat_test), actual = dat_test$num_event)
mean((test$actual - test$pred)^2)
```

```
## [1] 3.035467
```

```
# Use best subsets regression with the Adjusted R-squared criterion

regsubsetsObj <- regsubsets(dat_train$num_event~ I(X.Coordinate^2) + ., data=dat_train, really.big=TRUE)
print(summary(regsubsetsObj)$adjr2)
```

```
## [1] 0.2874915 0.3215921 0.3324492 0.3559632 0.3670528 0.3815303 0.3955162
## [8] 0.3968622
```

```r
coef_list = coef(regsubsetsObj,8)
print(coef_list)
```

```
##        (Intercept)  I(X.Coordinate^2)            Period4       X.Coordinate
##       2.7999055877       0.0001523115      -2.0655980621      -0.0350225213
##   EventFaceoff Win          EventPlay EventPuck Recovery          EventShot
##       6.4366905185       1.1198904027       1.0437527167       0.4875967598
##    EventZone Entry
##      18.6871943671
```

```r
regsubsets_formula = "dat_train$num_event~"

coef_list = coef_list[-1]
for (name in names(coef_list)) {
  value <- coef_list[[name]]
  if (value != 0){
    name <- ifelse(grepl("^Period", name), "Period", name)
    name <- ifelse(grepl("^Event", name), "Event", name)
    regsubsets_formula = paste(regsubsets_formula, name)
    regsubsets_formula = paste(regsubsets_formula, "+")
  }
}
regsubsets_formula <- substring(regsubsets_formula, 1, nchar(regsubsets_formula) - 1)

# print(regsubsets_formula)
best_sub_model = lm(as.formula(regsubsets_formula), data = dat_train)

summary(best_sub_model)
```

```
##
## Call:
## lm(formula = as.formula(regsubsets_formula), data = dat_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4927  -0.9233  -0.1913   0.5373  31.5250
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.039e+00  1.826e-01  16.646  < 2e-16 ***
## I(X.Coordinate^2)      1.581e-04  1.437e-05  11.001  < 2e-16 ***
## Period2                2.313e-02  1.050e-01   0.220   0.8256
## Period3               -1.003e-02  1.060e-01  -0.095   0.9246
## Period4               -2.048e+00  2.604e-01  -7.865 5.25e-15 ***
## X.Coordinate          -3.601e-02  2.901e-03 -12.414  < 2e-16 ***
## EventFaceoff Win       6.219e+00  4.578e-01  13.584  < 2e-16 ***
## EventGoal             -3.974e-01  5.819e-01  -0.683   0.4947
## EventIncomplete Play  -2.182e-01  1.693e-01  -1.289   0.1976
## EventPenalty Taken    -7.318e-01  4.436e-01  -1.650   0.0991 .
## EventPlay              8.956e-01  1.513e-01   5.918 3.65e-09 ***
## EventPuck Recovery     8.187e-01  1.526e-01   5.366 8.73e-08 ***
## EventShot              2.494e-01  2.149e-01   1.161   0.2458
## EventTakeaway         -5.294e-01  2.112e-01  -2.507   0.0122 *
## EventZone Entry        1.848e+01  4.881e-01  37.850  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.241 on 2766 degrees of freedom
## Multiple R-squared:  0.4003, Adjusted R-squared:  0.3973
## F-statistic: 131.9 on 14 and 2766 DF,  p-value: < 2.2e-16
```

```r
# mean squared error
test <- data.frame(predict(best_sub_model, dat_test), actual = dat_test$num_event)
mean((test$actual - test$pred)^2)
```

```
## [1] 3.035467
```

```r
# Use Dimension reduction tool
famd <- FAMD(dat_train,ncp=3, graph=FALSE)

# Extract the components
reduced <- as.data.frame(famd$ind$coord)

# Build a linear model
famd_model <- lm(dat_train$num_event~ I(Dim.1^2) + ., data = reduced )

# Print the summary of the linear model
summary(famd_model)
```

```
##
## Call:
## lm(formula = dat_train$num_event ~ I(Dim.1^2) + ., data = reduced)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0736  -0.4746  -0.1164   0.3304  10.0339
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  2.137579   0.024547  87.082  < 2e-16 ***
## I(Dim.1^2)   0.023061   0.003109   7.417 1.58e-13 ***
## Dim.1        1.782341   0.038862  45.863  < 2e-16 ***
## Dim.2        0.011092   0.021460   0.517    0.605
## Dim.3       -0.250660   0.023583 -10.629  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.268 on 2776 degrees of freedom
## Multiple R-squared:  0.8075, Adjusted R-squared:  0.8072
## F-statistic:  2911 on 4 and 2776 DF,  p-value: < 2.2e-16
```

```r
# Use Dimension reduction tool
famd_test <- FAMD(dat_test,ncp=3, graph=FALSE)

# Extract the components
reduced_test <- as.data.frame(famd_test$ind$coord)

# mean squared error
test <- data.frame(predict(famd_model, reduced_test), actual = dat_test$num_event)
mean((test$actual - test$pred)^2)
```

```
## [1] 1.586788
```

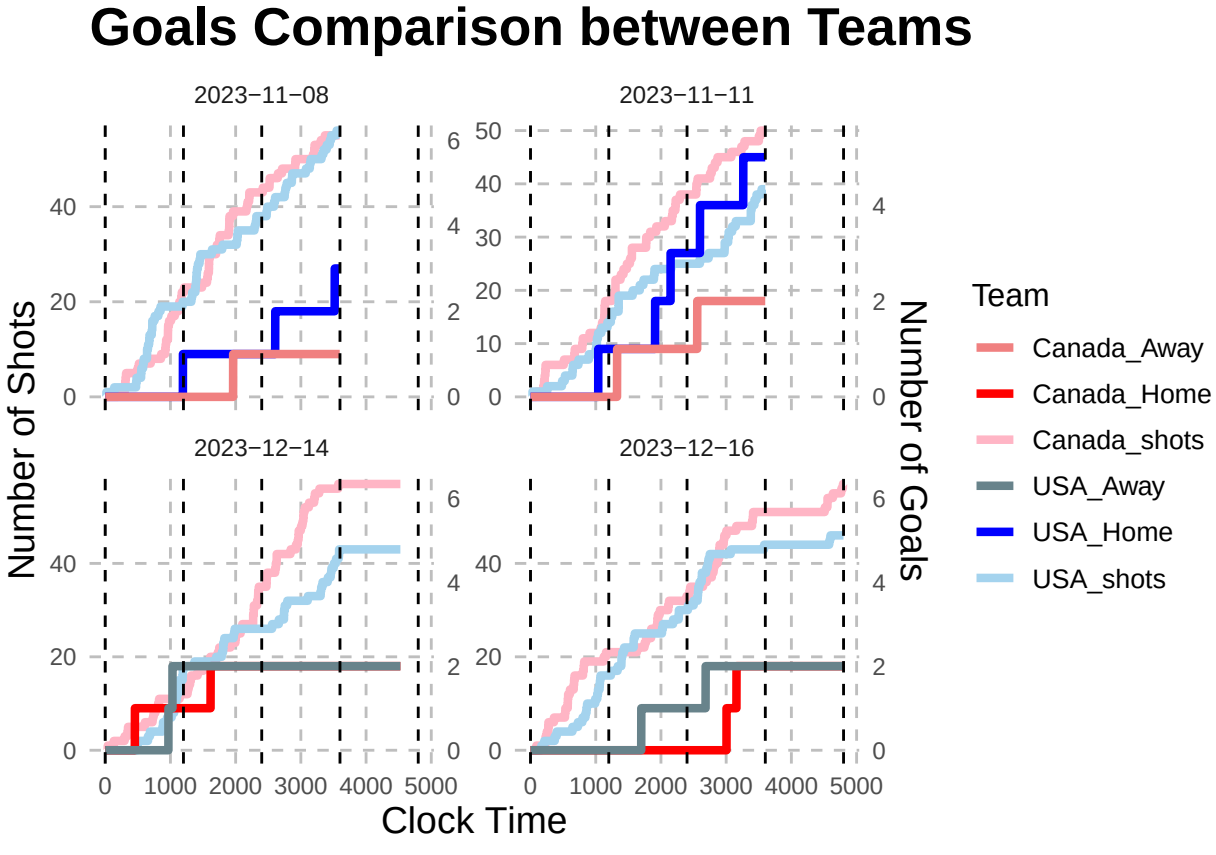
In this question, we are constructing a model using one of the continuous variables among my six most important ones, which is X-Coordinate. The model predicts the number of events at a given Period, Team, and X-coordinate. We build three models using stepwise regression, best-subset, and Factor Analysis of Mixed Data. By calculating mean square error on the test set, we find that the RAMD model has the smallest error, and it's simpler with reduced dimensionality to 3. Using this model, we can predict and learn about each team's strategy. As the model learns each team's habits, we'll know what events are likely to happen across all x-axes for each team and the estimated number of events in specific locations during specific periods. This is valuable for estimating each team's performance and strategies.

6) **OPTIONAL: PICK 2 OF 4** Build a visually impressive ggplot to show the relationship between at least three variables.

```
dat6 = dat

time_parts <- strsplit(dat6$Clock, ":")




clock_to_sec <- function(clock) {
  clock_parts <- strsplit(clock, ":")
  minutes <- as.numeric(clock_parts[[1]][1])
  seconds <- as.numeric(clock_parts[[1]][2])
  return(1200 - (minutes * 60 + seconds))
}

# Change the clock to second and start from 00:00 instead of 20:00
dat6$Clock_sec <- as.numeric(sapply(dat6$Clock, clock_to_sec) + (as.numeric(dat6$Period) - 1) * 1200 )


# Make a column tracking the shoting event
make_cont <- function(shot_count, team, game_date) {
  curr <- 0
  curr_Date = "2023-11-08"
  for (i in 1:length(shot_count)) {
    if (shot_count[i] != -1) {
      curr <- shot_count[i]
      if (game_date[i] == "2023-11-11"){
        if (team == "Women - Canada"){
          curr <- curr - 55
        }else{curr <- curr - 56}
      }
      if (game_date[i] == "2023-12-14"){
        if (team == "Women - Canada"){
          curr <- curr - 105
        }else{curr <- curr - 95}
      }
      if (game_date[i] == "2023-12-16"){
        if (team == "Women - Canada"){
          curr <- curr - 162
        }else{curr <- curr - 138}
      }
      shot_count[i] <- curr
      curr_Date = game_date[i]
    } else {
      if (game_date[i] != curr_Date){
        shot_count[i] <- 0}
      else{shot_count[i] <- curr}

    }
  }

  return(shot_count)
}


# cumsum the shots for both team
dat6 <- dat6 %>%
  arrange(Date, Clock_sec) %>%
  mutate(Shoot_Count_ca = ifelse(Team == "Women - Canada" & Event == "Shot", cumsum(Event == "Shot" & Team == "Women - Canada"), -1))


dat6 <- dat6 %>%
  arrange(Date, Clock_sec) %>%
  mutate(Shoot_Count_us = ifelse(Team == "Women - United States" & Event == "Shot", cumsum(Event == "Shot" & Team == "Women - United States"), -1))

dat6$Shoot_Count_ca = make_cont(dat6$Shoot_Count_ca, "Women - Canada",dat6$Date)
dat6$Shoot_Count_us = make_cont(dat6$Shoot_Count_us,"Women - United States",dat6$Date)

ggplot(dat6, aes(x = Clock_sec)) +
  geom_line(aes(y = Shoot_Count_ca, color = "Canada_shots"), size = 1.5) +
  geom_line(aes(y = Shoot_Count_us, color = "USA_shots"), size = 1.5) +

   geom_line(aes(y = Home.Team.Goals * 9, color = ifelse(Home.Team == "Women - Canada", "Canada_Home", "USA_Home")), size = 1.5) +

  geom_line(aes(y = Away.Team.Goals * 9, color = ifelse(Away.Team == "Women - Canada", "Canada_Away", "USA_Away")), size = 1.5) +

    scale_y_continuous(

    name = "Number of Shots",

    sec.axis = sec_axis(~./9, name="Number of Goals")
  ) +


  facet_wrap(~ Date, nrow = 2, scales = 'free_y') +
  xlim(0,4800)+
```

```
  scale_color_manual(values = c(Canada_Home = "red", Canada_Away = "lightcoral",
                                USA_Home = "blue", USA_Away = "lightblue4",
                                Canada_shots = "pink1", USA_shots = "lightskyblue2",
                                Other = "black")) +
labs(title = "Goals Comparison between Teams", x = "Clock Time", y = "Number of Goals", color = "Team") +
  geom_vline(xintercept = c(0, 1200, 2400, 3600, 4800), linetype = "dashed", color = "black")+
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20, face = "bold"),
    axis.title = element_text(size = 14),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    panel.grid.minor = element_blank(),
    panel.grid.major = element_line(color = "gray", linetype = "dashed")
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The plot illustrates the correlation among Clock Time, Period, Date, Home Team, Away Team, Home Team Goals, and Away Team Goals.Also with the number of shots both team have.

Clock Time is derived from the clock and period columns, where the clock counts down while the clock time counts up. Each period spans 20 minutes, equivalent to 1200 seconds. Thus, the current clock time is calculated as 1200 multiplied by the period number minus one, plus the clock time counting up. The dotted line denotes the period divisions. The labels distinguish between the Team USA and Team Canada, either it's home or away, and the amount of attempts both teams do in order to get goal. Left y-axis stands for number of shots each team made, right y-axis stands for number of goals each team have by making these amount of shots.

From the graph, we obtained similar results as ggpairs, revealing that although Team Canada has a significant advantage in shot quantity, it doesn't translate into goals. In the Dec.16 game, despite Canada's efforts to score in the second and third periods, numerous attempts only led to Team USA extending their lead.

We can also analyze both teams' attacking strategies using this plot.

For Team USA:
In their Home Game, the shot count rises steadily when the score is close. Once they lead by a lot, they become more aggressive. Particularly in game 2, with a substantial lead, attempts increase notably in period 3.
In their Away Game, shots increase steadily but decrease toward the end. This can be because they are prioritizing defence over offence, regardless of the score. They prefer not to lose than win the game.

Team USA maintains stable strategic plans, efficiently converting goals. Although we didn't analyze the defence part yet, we should be able to know that they excel in defence and counterattacks against the aggressive Team Canada.

For Team Canada:
With a more offensive approach, Team Canada persistently seeks goals regardless of the period or score. Their strategy is pure attack, considering it the best form of defence. Most players are good at creating shooting opportunities and exhibit high energy.

Although we lack additional plots to analyze defence, we can speculate on Team Canada's losses. They may lack strong defensive players, relying too heavily on offence. Additionally, their wide shooting zone suggests a strategy akin to gegenpressing in soccer, prioritizing attack over defence, leaving them vulnerable to counterattacks.

8) **OPTIONAL: PICK 2 OF 4** Discuss briefly any ethical concerns like residual disclosure that might arise from the use of your data set, possibly in combination with some additional data outside your dataset.

My dataset containing detailed event data for women's hockey, it will normally raise ethical concerns, especially regarding personal privacy, and information disclosure. Let's combine several points covered in the lecture and list some ethical considerations.

Privacy is the most important topic we need to be aware of when dealing with statistical activities involving personal information. Thus, privacy and security considerations are key and mandatory. The dataset includes information about specific players, teams, and their performance during games, potentially giving the information to do the identification of individuals when combined with other events. For instance, strange player performance or unusual behaviour on the court could be cross-referenced with external news, compromising their privacy and potentially harming the players.

Moreover, since the dataset encompasses details about players, teams, and their in-game actions or strategies, it's crucial to ensure whether the players and teams consent to collecting and using their data for research purposes. Without proper consent, utilizing the data would undoubtedly raise ethical concerns regarding respect for individuals' rights. And also may harm teams and player, and waste their work on preparing strategies. This is very disrespectful.

When conducting statistical activities, it's necessary to consider all potential risks to the well-being of individuals or specific groups. Analysis of the data may reveal biases in the game, such as biased officiating, disparities in player treatment, or discriminatory practices, which occur frequently across all sports. It can be the referee on the court, it can be fans in the stadium. Being aware of these biases is essential to ensure that our analysis does not cause harm.

Statistical activities aimed at benefiting society must be transparent about data sources, usage, and confidentiality measures. Respect for data ownership rights and proper attribution to the source are imperative. Using data without permission or failing to acknowledge dataset contributions is unacceptable and disrespectful.

Conducting analysis or drawing conclusions without considering broader contexts can lead to misinterpretations. We must be meticulous, striving for transparency and ethical integrity in our analysis and reporting endeavors.