Project1

Ryan Kyaw, Makenna Drye, Mark Ramos

April 15, 2019

Introduction

The Crazy-8 Data Set is a compilation of data from 100 games of Crazy-8 played by three students. The game Crazy-8 comes from the iOS app, GamePigeon, and this app allows people to play games like UNO, Connect4, and 8-ball using iMessage. In particular, Crazy-8 is similar to the game of UNO, and this game is one of the most popular games in GamePigeon.

Data was collected manually by the three students playing Crazy-8, and no outside sources influenced this data set. Before each game, the quality of day, whether the subjects were in class, and the outside temperature was recorded. Throughout the entire game, each player recorded individual statistics about the type of cards held and the number of turns. The variables about the type of cards held will be explained further in the data dictionary.

Data Dictionary

- Winner: Categorical, 3 levels. The winner of each individual game was recorded. The levels of the variable is “Makenna”, “Ryan”, and “Mark”, as these are the three individuals participating in the data collection.
- Date: Categorical, 3 levels. The date of each game played was recorded, and the levels correspond to the month and day each game was played.
- Day.Rating.Mark, Day.Rating,Makenna, Day.Rating.Ryan: Numerical, postive integer from [0, 10]. Data on how well each subject’s day was recorded, and to make this quantitative, each player used a number from 1 to 10 to describe how good or bad his day was going.
- In.Class: Categorical, 2 levels. Whether the game was played while at least one member of the study was in class was recorded.
- Color.Change.Beg.Mark, Color.Change.Beg.Makenna, Color.Change.Beg.Ryan: Numerical, positive integer from [0, 7]. Color change cards are defined in this study as change color cards or draw-4 change color cards. These variables record the amount of color change cards that each player started a particular game with.
- Day.Temp: Numerical, positive integer. The outside temperature of each game played was recorded.
- One.Card.Mark, One.Card.Makenna, One.Card.Ryan: Numerical, positive integer. The number of times each individual was down to one card during a particular game was recorded.
- Turns: Numerical, positive integer. The number of total turns during each game played. Throughout each game, every player recorded the number of turns they had during the game. After the game, these values were added up to come up the total number of turns for the game.
- Cards.Drawn.Mark, Cards.Drawn.Makenna, Cards.Drawn.Ryan: Numerical, positive integer. THe number of cards that each individual drew throughout an individual game. Cards drawn is defined as the number of cards drawn because of a draw-card played in the previous turn or drawing a card because of an inability to play any card in the player’s hand.

- Special.Beg.Mark, Special.Beg.Makenna, Special.Beg.Ryan: Numerical, positive integer from [0, 7]. Special cards are defined as the draw-2 cards, skip cards, and reverse cards. This variable is the number of special cards that each individual holds at the beginning of a game.
- Change.Color.Drawn.Mark, Change.Color.Drawn.Makenna, Change.Color.Drawn.Ryan: Numerical, positive integer. This variable records the number of change color cards drawn by each player throughout a particular game.
- Special.Drawn.Mark, Special.Drawn.Makenna, Special.Drawn.Ryan: Numerical, positive integer. This variable records the number of special cards drawn by each player throughout a particular game.
- Max.Cards.Mark, Max.Cards.Makenna, Max.Cards.Ryan: Numerical, positive integer. This variable records the maximum amount of cards that each player has during a game.
- Wild.Beg.Mark, Wild.Beg.Makenna, Wild.Beg.Ryan: Numerical, positive integer. This variable combines the number of special and color change cards that each player starts with.
- Wild.Drawn.Mark, Wild.Drawn.Makenna, Wild.Drawn.Ryan: Numerical, positive integer. This variable combines the number of special and color change cards that each player draw throughout the game.
- gamelength: categorical, 4 levels. This variable defined games of 0 to 20 turns as short, 21 to 40 turns as medium, 41 to 60 turns as long, and 61 turns and above as very long games.

Preparing/Cleaning the Data

Loading up the tidyverse library and our dataset

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------- tidyver
se 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
## -- Conflicts ------------------------------------------------------------- tidyverse_con
flicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df <- read.csv("Project 1 Data - Sheet1.csv")
```

Before the experiment began, we thought that the variables describing how our days were going, whether we were in class or not, and the outside temperature would convey some interesting, unintuitive information about the winner of a game of Crazy-8s. However, after performing the experiment, not any clear relationships with these variables stand out. Also, we were only able to play 3 games during class time. Hence, we can remove the Day.Rating, In.Class, and Day.Temp variables from the data set.

```
df$Day.Rating.Makenna <- NULL
df$Day.Rating.Mark <- NULL
df$Day.Rating.Ryan <- NULL
df$In.Class <- NULL
df$Day.Temp <- NULL
```

Let us create a variable that combines the special and color change cards. Cards that are either special or a color change will be defined as “wild” cards.

```
df$Wild.Beg.Mark <- df$Color.Change.Beg.Mark + df$Special.Beg.Mark
df$Wild.Beg.Makenna <- df$Color.Change.Beg.Makenna + df$Special.Beg.Makenna
df$Wild.Beg.Ryan <- df$Color.Change.Beg.Ryan + df$Special.Beg.Ryan
df$Wild.Drawn.Mark <- df$Change.Color.Drawn.Mark + df$Special.Drawn.Mark
df$Wild.Drawn.Makenna <- df$Change.Color.Drawn.Makenna + df$Special.Drawn.Makenna
df$Wild.Drawn.Ryan <- df$Change.Color.Drawn.Ryan + df$Special.Drawn.Ryan
```

We will create separate data sets that filters out the data for each player in the study.

```
df_Ryan <- df[c(1:2, 5, 8:9, 12, 15, 18, 21, 24, 27, 30)]
df_Mark <- df[c(1:3, 6, 9:10, 13, 16, 19, 22, 25, 28)]
df_Makenna <- df[c(1:2, 4, 7, 9, 11, 14, 17, 20, 23, 26, 29)]
```

Let us create a categorical variable that labels games as short (turns - [0,20]), medium (turns - [21, 40]), long (turns - [41, 60]), and very long (turns > 60).

```
df$gamelength <- cut(df$Turns, c(min(df$Turns)-1, 20, 40, 60, max(df$Turns)+1), labels = c("shor
t", "medium", "long", "verylong"))
```

Let us create a total cards drawn variable among the players.

```
df$total.cards.drawn <- df$Cards.Drawn.Makenna + df$Cards.Drawn.Mark + df$ Cards.Drawn.Ryan
```

Exploratory Data Analysis

To begin, let us find out how many wins each player had over the course of our study.

Number of Ryan wins

```
length(which(df$Winner == "Ryan"))
```

```
## [1] 24
```

Number of Mark wins

```
length(which(df$Winner == "Mark"))
```

```
## [1] 19
```

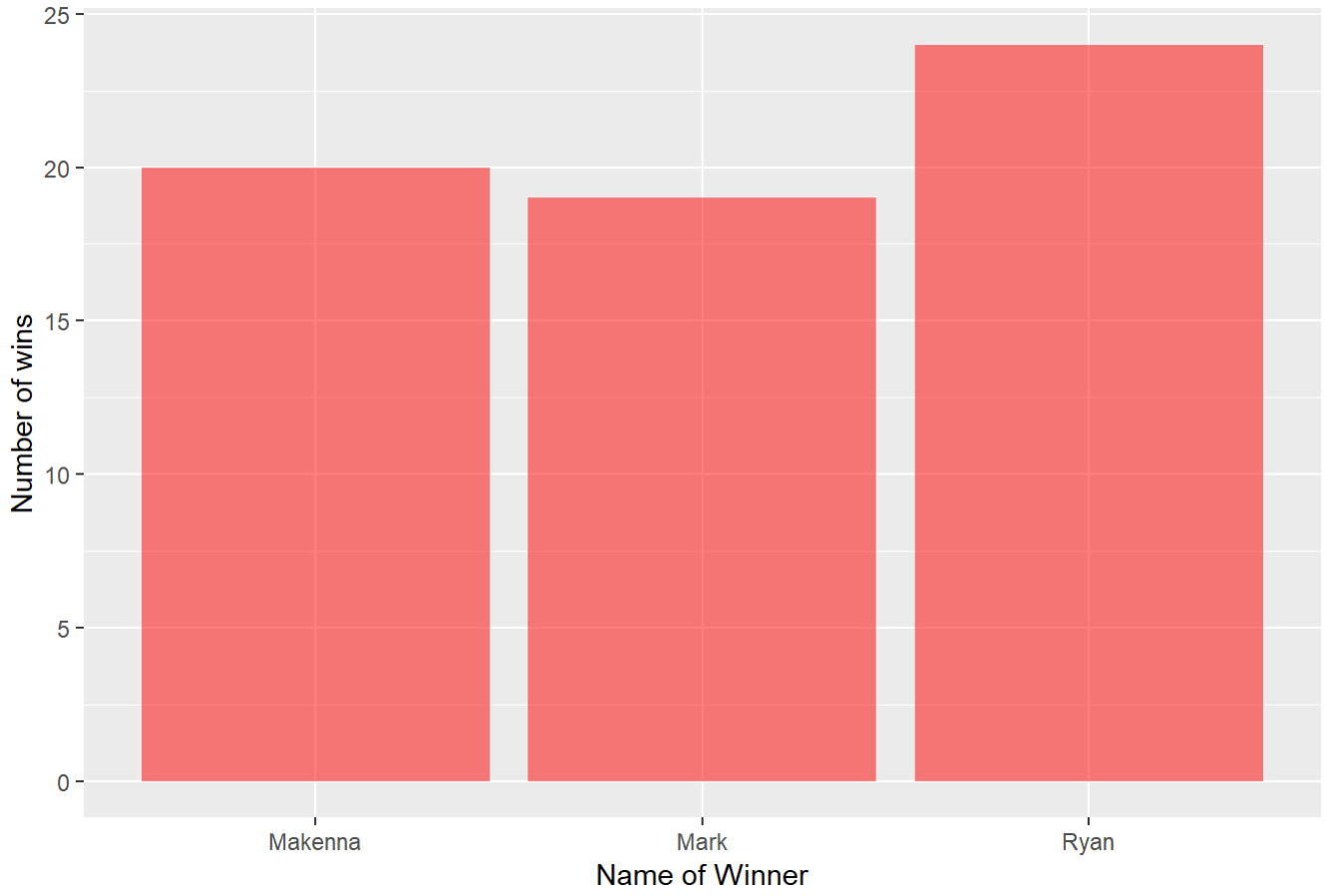
Number of Makenna wins

```
length(which(df$Winner == "Makenna"))
```

```
## [1] 20
```

```
ggplot(df) +
  geom_bar(aes(x = Winner), fill = "red", alpha = 0.5) +
  ggtitle("Figure 1: The amount of wins each player had") +
  xlab("Name of Winner")+
  ylab("Number of wins")
```

Figure 1: The amount of wins each player had



Ryan had the most wins at 24, Makenna had the second most wins at 20, and Mark finished with 19 wins.

Now, let us analyze the number of wins each player had based on game length.

The “short” games (0 to 20 turns)

```
short.win <- df$Winner[grep("short", df$gamelength)]
length(which(short.win == "Makenna"))
```

```
## [1] 7
```

```
length(which(short.win == "Ryan"))
```

```
## [1] 8
```

```
length(which(short.win == "Mark"))
```

```
## [1] 7
```

The “medium” games (21 to 40 turns)

```
medium.win <- df$Winner[grep("medium", df$gamelength)]
length(which(medium.win == "Makenna"))
```

```
## [1] 8
```

```
length(which(medium.win == "Mark"))
```

```
## [1] 5
```

```
length(which(medium.win == "Ryan"))
```

```
## [1] 9
```

The “long” games (41 to 60 turns)

```
long.win <- df$Winner[grep("long", df$gamelength)]
length(which(long.win == "Makenna"))
```

```
## [1] 5
```

```
length(which(long.win == "Mark"))
```

```
## [1] 7
```

```
length(which(long.win == "Ryan"))
```

```
## [1] 7
```

The “very long” games (61 turns and higher)

```
verylong.win <- df$Winner[grep("verylong", df$gamelength)]
length(which(verylong.win == "Makenna"))
```

```
## [1] 3
```

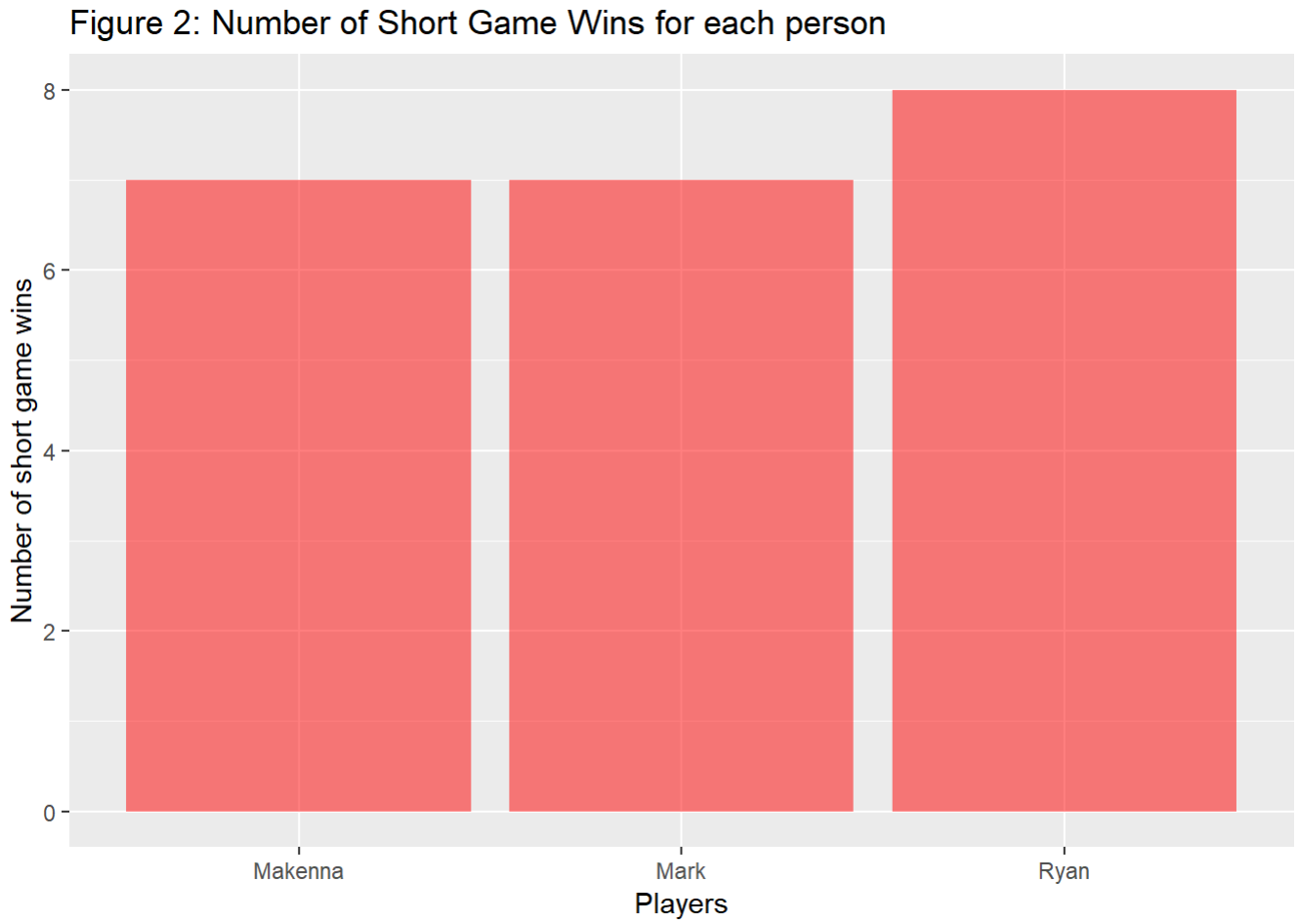
```
length(which(verylong.win == "Mark"))
```

```
## [1] 2
```

```
length(which(verylong.win == "Ryan"))
```

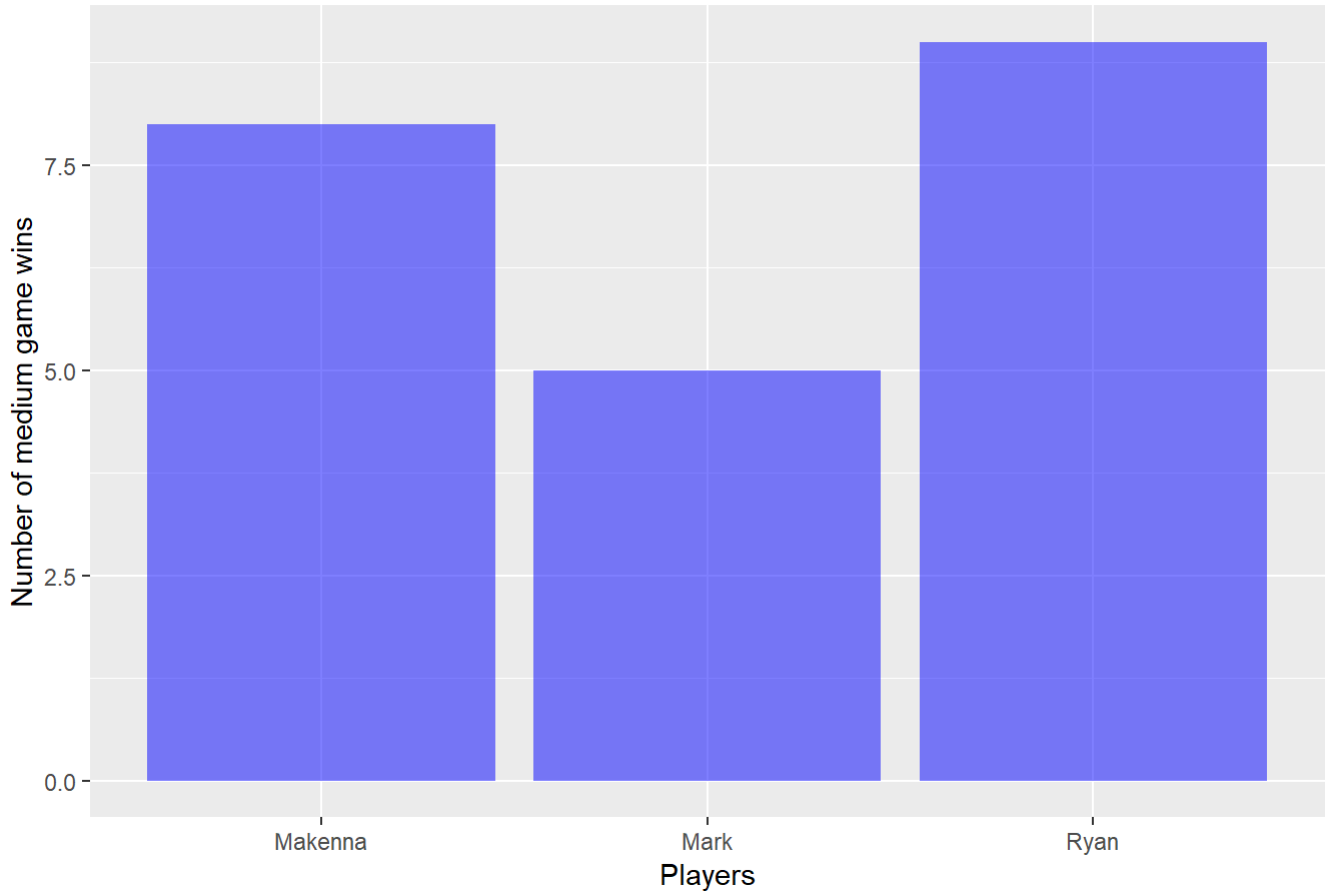
```
## [1] 3
```

```
ggplot()+
  geom_bar(aes(x = short.win), fill = "red", alpha = 0.5) +
  ggtitle("Figure 2: Number of Short Game Wins for each person") +
  xlab("Players") +
  ylab("Number of short game wins")
```



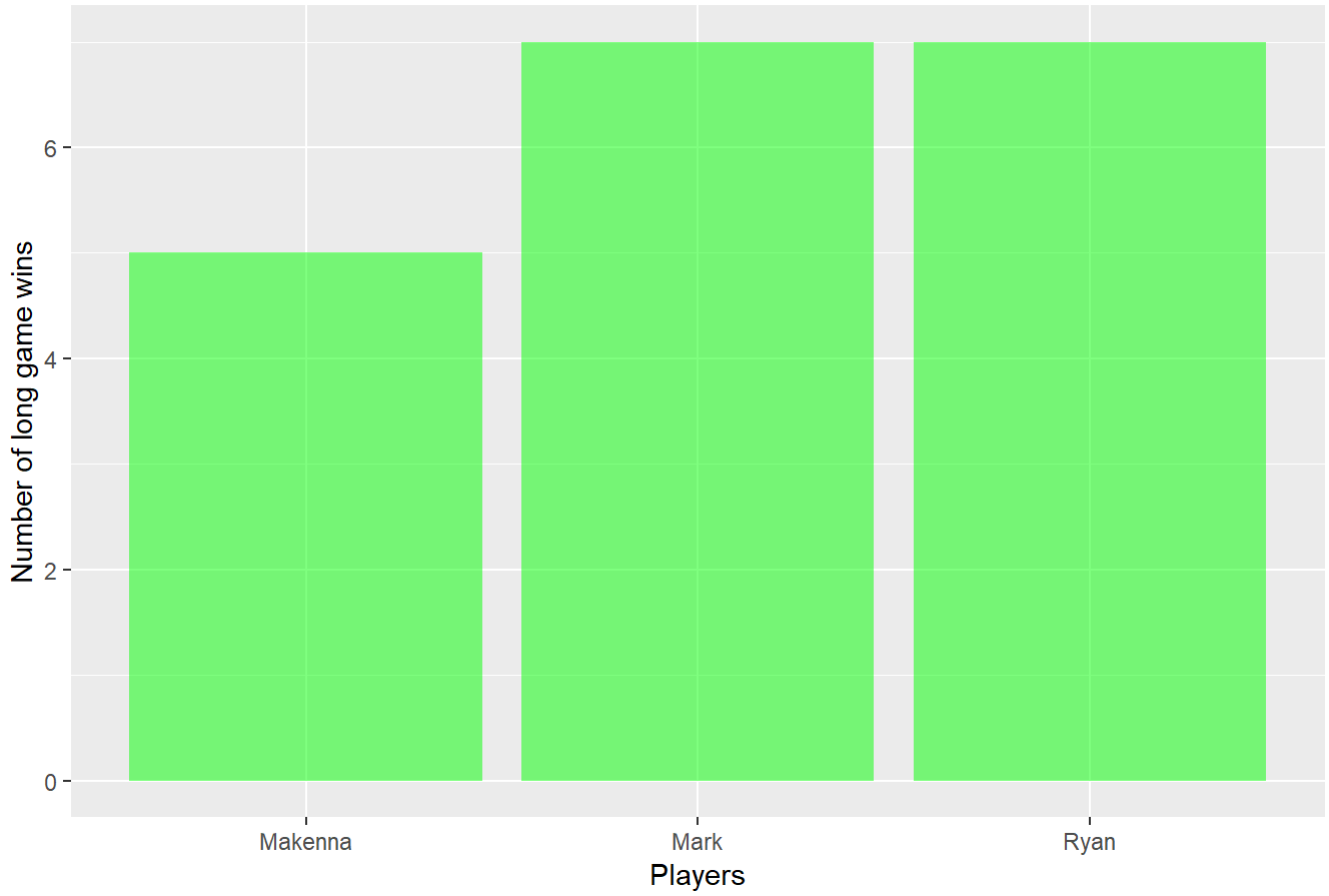

```
ggplot()+
  geom_bar(aes(x = medium.win), fill = "blue", alpha = 0.5) +
  ggtitle("Figure 3: Number of Medium Game Wins for each person") +
  xlab("Players") +
  ylab("Number of medium game wins")
```

Figure 3: Number of Medium Game Wins for each person



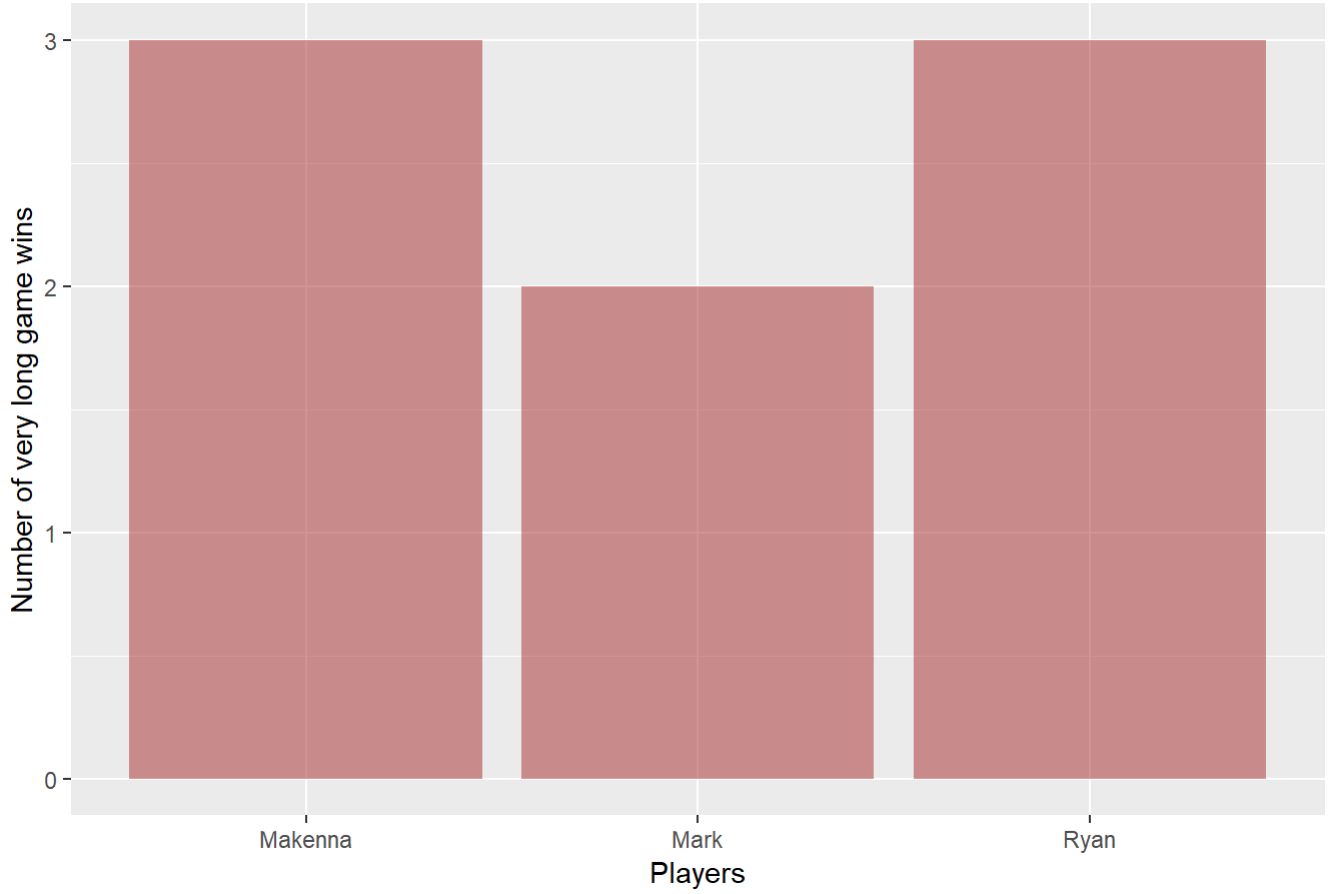

```
ggplot()+
  geom_bar(aes(x = long.win), fill = "green", alpha = 0.5) +
  ggtitle("Figure 4: Number of Long Game Wins for each person") +
  xlab("Players") +
  ylab("Number of long game wins")
```

Figure 4: Number of Long Game Wins for each person




```
ggplot()+
  geom_bar(aes(x = verylong.win), fill = "brown", alpha = 0.5) +
  ggtitle("Figure 5: Number of Very Long Game Wins for each person") +
  xlab("Players") +
  ylab("Number of very long game wins")
```

Figure 5: Number of Very Long Game Wins for each person

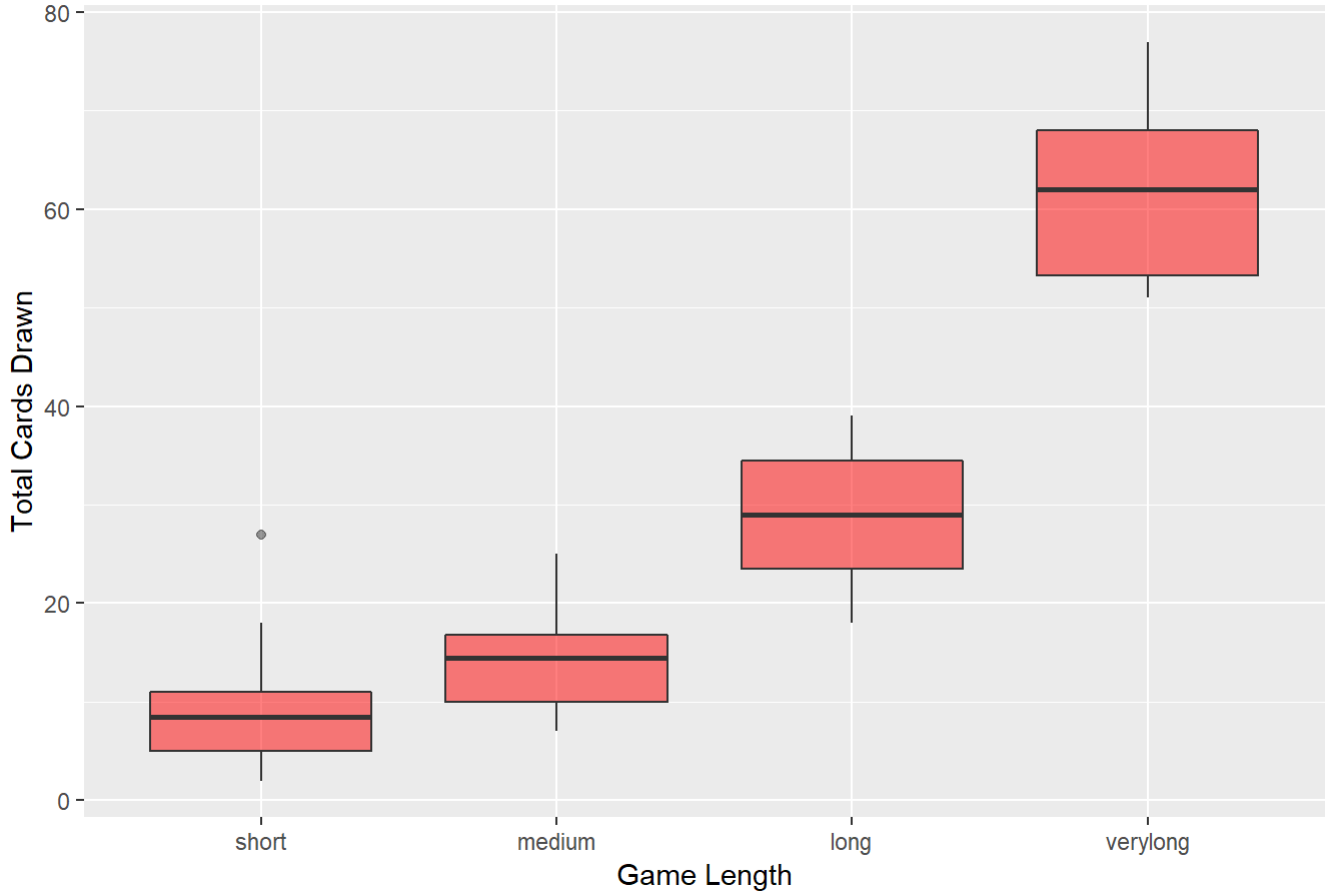


Looking at the numbers and the bar graphs, there seemed to be no one player that stood out as being dominant given a specific game length. If there were more observations to this data, then maybe a pattern would emerge. However, the luck and randomness that defines Crazy-8s and UNO seems to explain the lack of someone dominating a certain game length.

Comparing the amount of cards drawn to game length

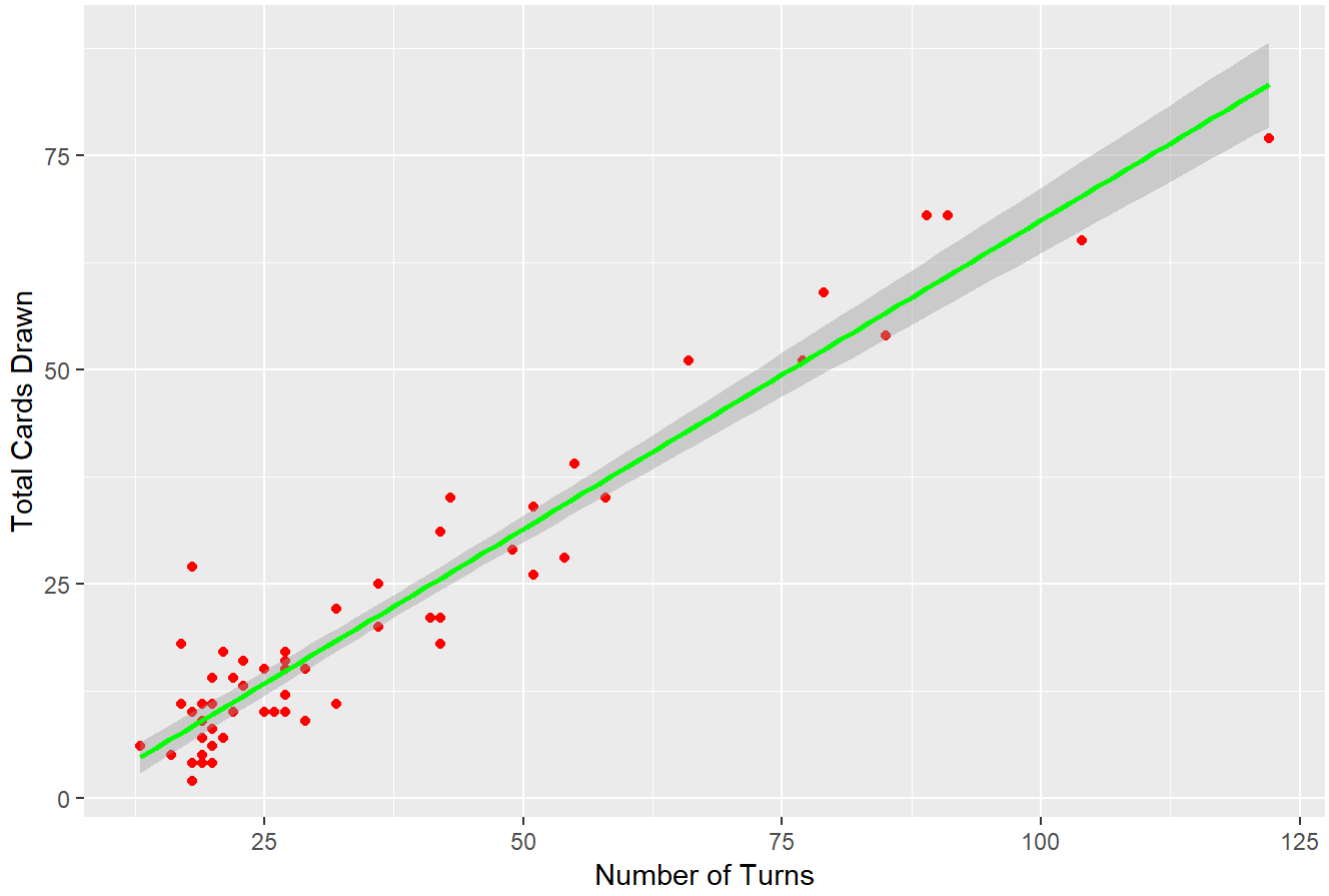
```
ggplot(data = df)+
  geom_boxplot(aes(x = gamelength, y = total.cards.drawn), fill = "red", alpha = 0.5) +
  ggtitle("Figure 6: Game Length vs Total Cards Drawn") +
  xlab("Game Length") +
  ylab("Total Cards Drawn")
```

Figure 6: Game Length vs Total Cards Drawn




```
ggplot(data = df)+
  geom_point(aes(x = Turns, y = total.cards.drawn), color = "red") +
  ggtitle("Figure 7: Total Cards Drawn vs Number of Turns") +
  xlab("Number of Turns")+
  ylab("Total Cards Drawn") +
  geom_smooth(aes(x = Turns, y = total.cards.drawn), color = "green", method = "lm")
```

Figure 7: Total Cards Drawn vs Number of Turns



As expected, there appears to be a positive relationship between game length and total cards drawn. Let us create a linear regression model for the number of turns versus the total.cards drawn to confirm this.

```
lm.fit <- lm(Turns ~ total.cards.drawn, data = df)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Turns ~ total.cards.drawn, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.2096  -4.5353   0.4359   4.5201  15.1272
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.83150    1.34613   6.561 1.31e-08 ***
## total.cards.drawn  1.27326    0.04888  26.051  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.964 on 61 degrees of freedom
## Multiple R-squared:  0.9175, Adjusted R-squared:  0.9162
## F-statistic: 678.6 on 1 and 61 DF,  p-value: < 2.2e-16
```

The p-value associated with the total cards drawn variable is much less than 0.05. Furthermore, the correlation coefficient is very close to 1. Therefore, we can reject our null hypothesis, and say there is a relationship between the number of turns and total cards drawn. In turn, we confirm the positive correlation between number of turns and total cards drawn.

Let us now assess how the wild cards held at the beginning of a game correlates to winning or losing.

Here, we will extract the number of wild cards that Makenna, Mark, and Ryan had at the beginning of games they won. We will also extract the number of wild cards that Makenna, Mark, and Ryan had at the beginning of games they lost.

```
Wild.Beg.Makenna.Wins <- df$Wild.Beg.Makenna[grep("Makenna", df$Winner)]
Wild.Beg.Mark.Wins <- df$Wild.Beg.Mark[grep("Mark", df$Winner)]
Wild.Beg.Ryan.Wins <- df$Wild.Beg.Ryan[grep("Ryan", df$Winner)]
Wild.Beg.Makenna.Lose <- df$Wild.Beg.Makenna[df$Winner != "Makenna"]
Wild.Beg.Mark.Lose <- df$Wild.Beg.Mark[df$Winner != "Mark"]
Wild.Beg.Ryan.Lose <- df$Wild.Beg.Ryan[df$Winner != "Ryan"]
```

Now, let us average out the number of wild cards that the winner of a Crazy-8 game had at the beginning of the game.

```
sum1 <- mean(Wild.Beg.Makenna.Wins)*20
sum2 <- mean(Wild.Beg.Mark.Wins)*19
sum3 <- mean(Wild.Beg.Ryan.Wins)*24
avg.Wild.Beg.Winner <- (sum1 + sum2 + sum3)/63
avg.Wild.Beg.Winner
```

```
## [1] 2.396825
```

Next, we will average out the number of wild cards that the loser of a Crazy-8 game had at the beginning of the game.

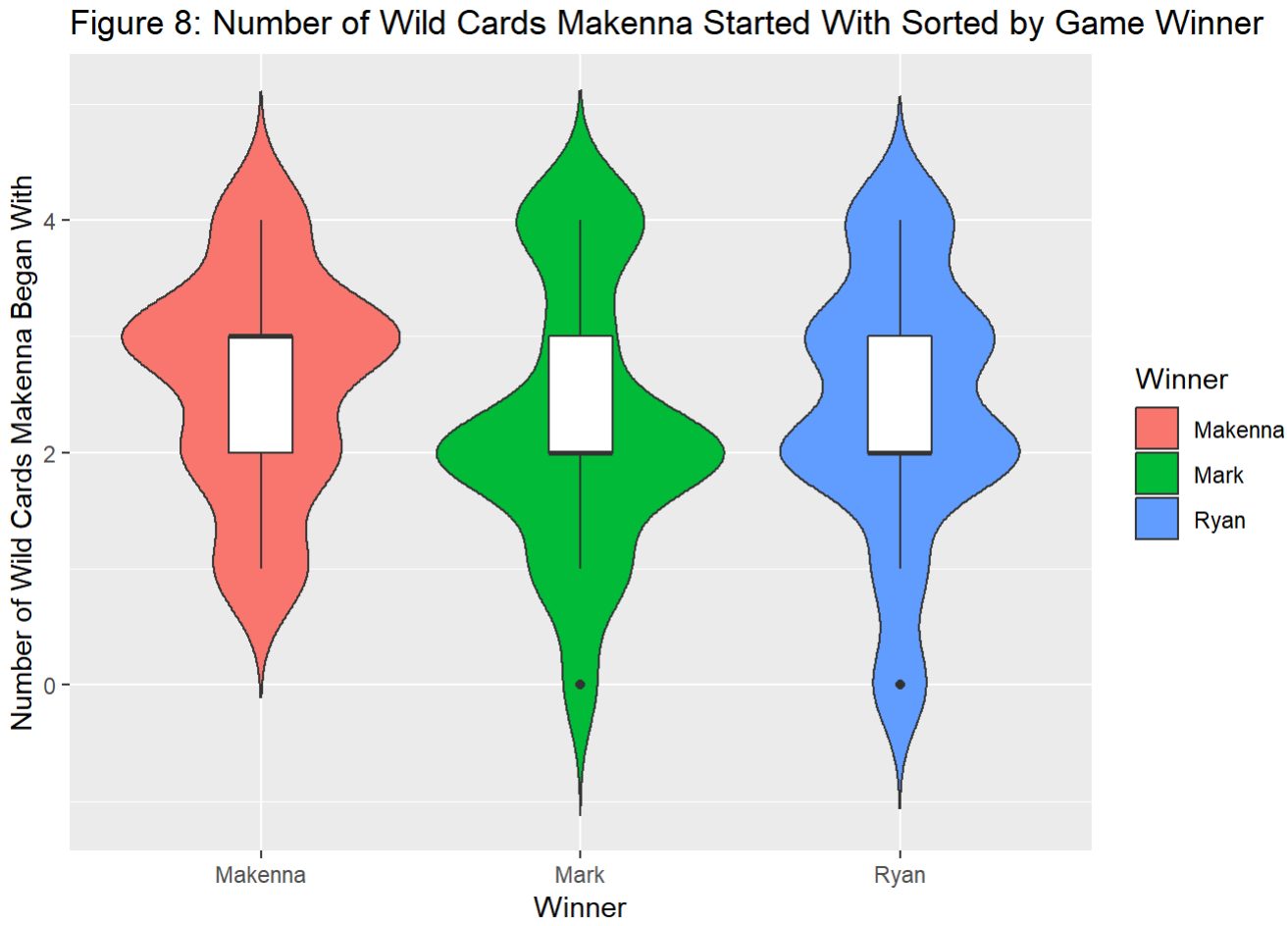
```
sum1 <- mean(Wild.Beg.Makenna.Lose)*43
sum2 <- mean(Wild.Beg.Mark.Lose)*44
sum3 <- mean(Wild.Beg.Ryan.Lose)*39
avg.Wild.Beg.Loser <- (sum1 + sum2 + sum3)/(43+44+39)
avg.Wild.Beg.Loser
```

```
## [1] 2.198413
```

The numbers show that the winner of a Crazy-8 game started out with only 0.2 more Wild Cards on average compared to the losers of the game. This is a very small difference, and this can be interpreted to believe that the number of special and change color cards that one starts with does not matter at all.

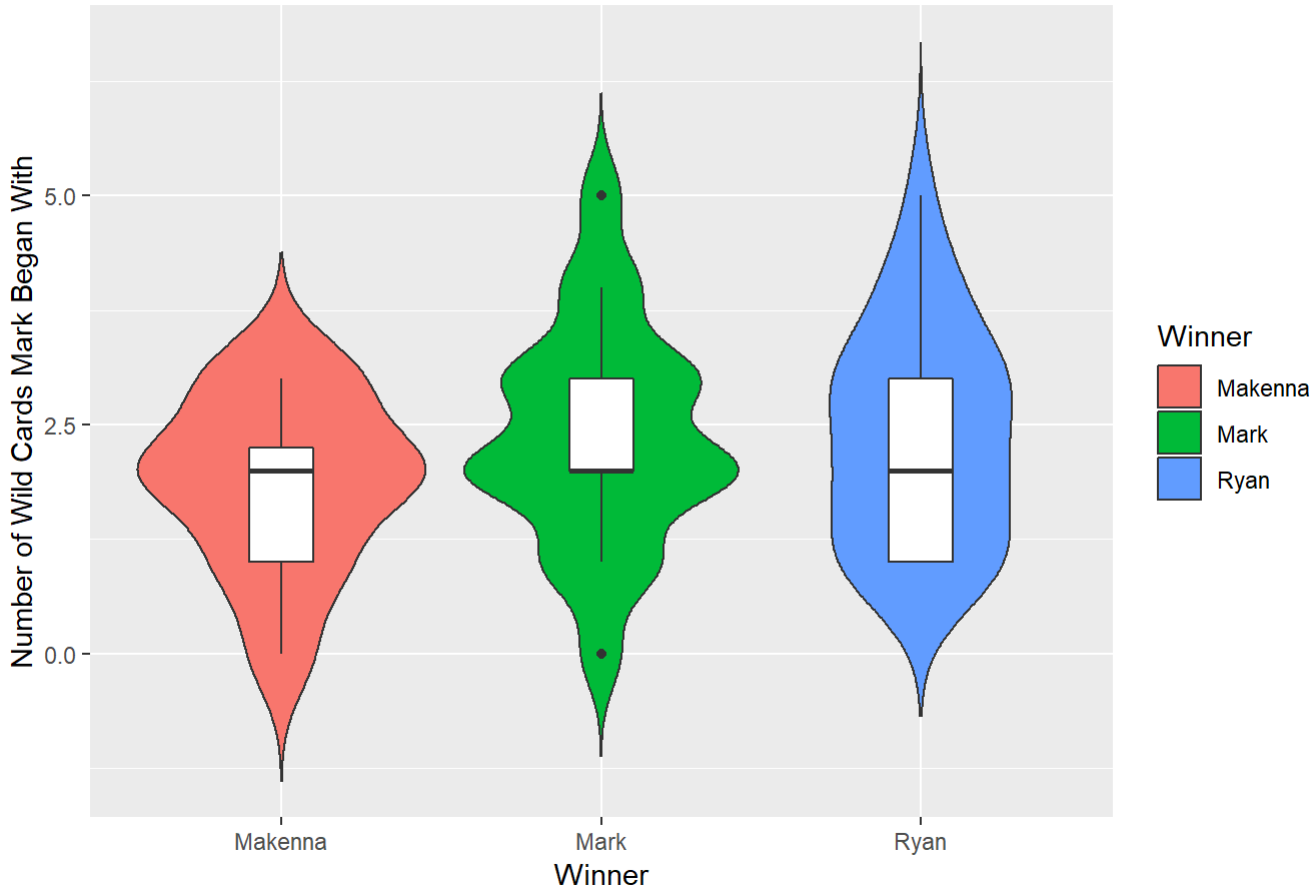
In fact, looking at the graphs below, they show that there is little to no difference to the Wild cards that a player starts with in relation to whether that player wins or not.

```
ggplot(data = df, aes(x = Winner, y = Wild.Beg.Makenna, fill = Winner))+
  geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.2, fill = "white")+
  ggtitle("Figure 8: Number of Wild Cards Makenna Started With Sorted by Game Winner")+
  xlab("Winner")+
  ylab("Number of Wild Cards Makenna Began With")
```



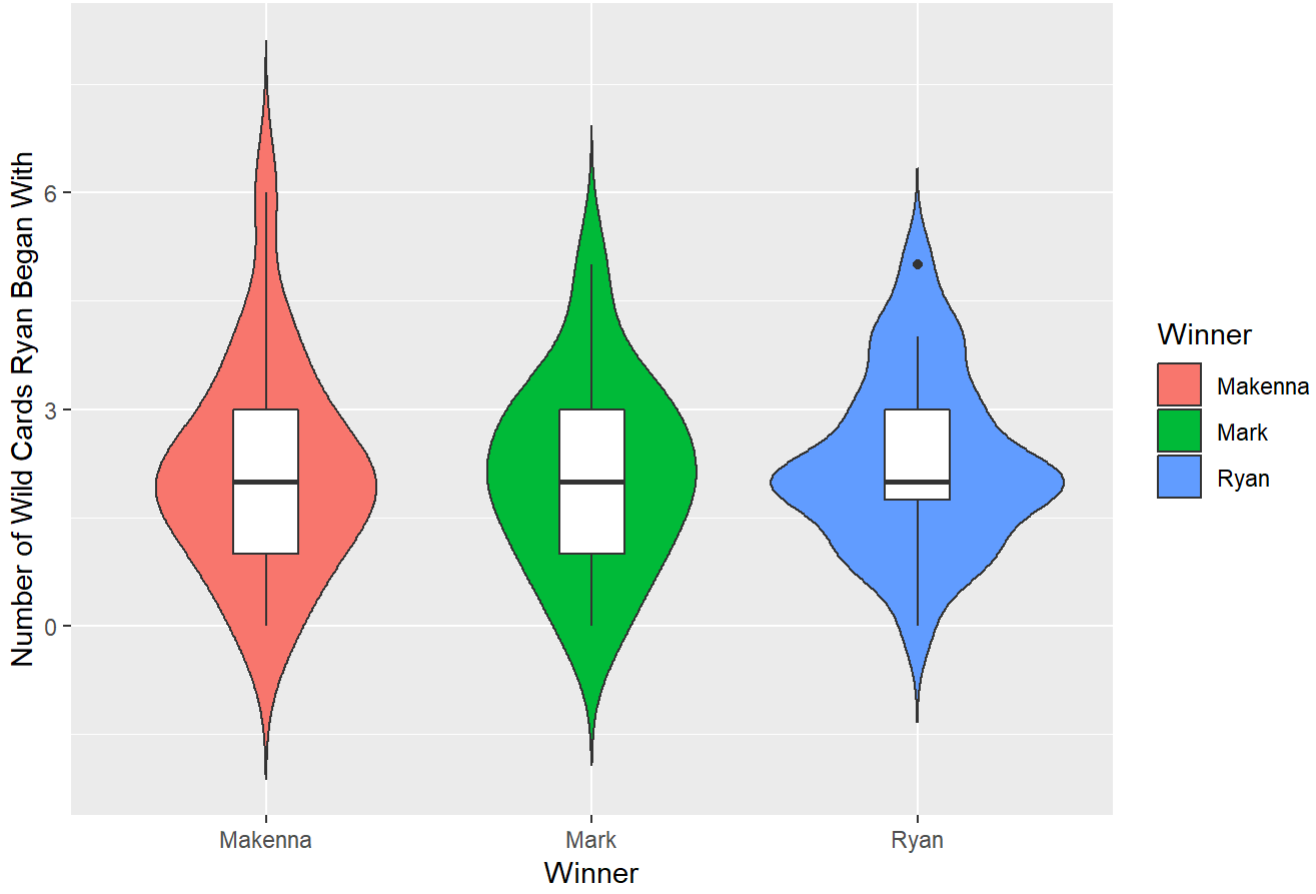

```
ggplot(data = df, aes(x = Winner, y = Wild.Beg.Mark, fill = Winner)) +
  geom_violin(trim = FALSE)+
  geom_boxplot(width = 0.2, fill = "white") +
  ggtitle("Figure 9: Number of Wild Cards Mark Started With Sorted by Game Winner") +
  xlab("Winner") +
  ylab("Number of Wild Cards Mark Began With")
```

Figure 9: Number of Wild Cards Mark Started With Sorted by Game Winner




```
ggplot(data = df, aes(x = Winner, y = Wild.Beg.Ryan, fill = Winner)) +
  geom_violin(trim = FALSE)+
  geom_boxplot(width = 0.2, fill = "white") +
  ggtitle("Figure 10: Number of Wild Cards Ryan Started With Sorted by Game Winner") +
  xlab("Winner")+
  ylab("Number of Wild Cards Ryan Began With")
```

Figure 10: Number of Wild Cards Ryan Started With Sorted by Game Winner




```
summary(aov(Wild.Beg.Makenna ~ Winner, data = df))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Winner       2   1.16  0.5803   0.497  0.611
## Residuals   60  70.11  1.1685
```

We will now use the bootstrapping method to test the null hypothesis of there being no difference in the wild card that the winner and losers of each game started with.

```
pop1 <- df$Wild.Beg.Makenna[df$Winner == "Makenna"]
pop2 <- df$Wild.Beg.Makenna[df$Winner == "Mark"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.2986
```

```
pop1 <- df$Wild.Beg.Mark[df$Winner == "Mark"]
pop2 <- df$Wild.Beg.Mark[df$Winner == "Ryan"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.9843
```

```
pop1 <- df$Wild.Beg.Ryan[df$Winner == "Ryan"]
pop2 <- df$Wild.Beg.Ryan[df$Winner == "Makenna"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.7993
```

Based on these p values that are greater than 0.05, we fail to reject the null hypothesis, and there appears to be no relationship between wild cards started with and winning Crazy-8 games. This appears to emphasize the luck and randomness that is associated with the game of Crazy-8s.

Next, we will analyze how the number of wild cards drawn by each player correlates to winning or losing.

Let us first extract the number of wild cards that Makenna, Mark, and Ryan drew during games they won and lost.

```
Wild.Drawn.Makenna.Wins <- df$Wild.Drawn.Makenna[grep("Makenna", df$Winner)]
Wild.Drawn.Mark.Wins <- df$Wild.Drawn.Mark[grep("Mark", df$Winner)]
Wild.Drawn.Ryan.Wins <- df$Wild.Drawn.Ryan[grep("Ryan", df$Winner)]
Wild.Drawn.Makenna.Lose <- df$Wild.Drawn.Makenna[df$Winner != "Makenna"]
Wild.Drawn.Mark.Lose <- df$Wild.Drawn.Mark[df$Winner != "Mark"]
Wild.Drawn.Ryan.Lose <- df$Wild.Drawn.Ryan[df$Winner != "Ryan"]
```

Now, let us figure out what the average number of wild cards drawn by the winner of a game.

```
sum1 <- mean(Wild.Drawn.Makenna.Wins)*20
sum2 <- mean(Wild.Drawn.Mark.Wins)*19
sum3 <- mean(Wild.Drawn.Ryan.Wins)*24
avg.Wild.Drawn.Winner <- (sum1 + sum2 + sum3)/63
avg.Wild.Drawn.Winner
```

```
## [1] 1.873016
```

Next, we can figure out the average number of wild cards drawn by the losers of a game.

```
sum1 <- mean(Wild.Drawn.Makenna.Lose)*43
sum2 <- mean(Wild.Drawn.Mark.Lose)*44
sum3 <- mean(Wild.Drawn.Ryan.Lose)*39
avg.Wild.Drawn.Loser <- (sum1 + sum2 + sum3)/(43+44+39)
avg.Wild.Drawn.Loser
```

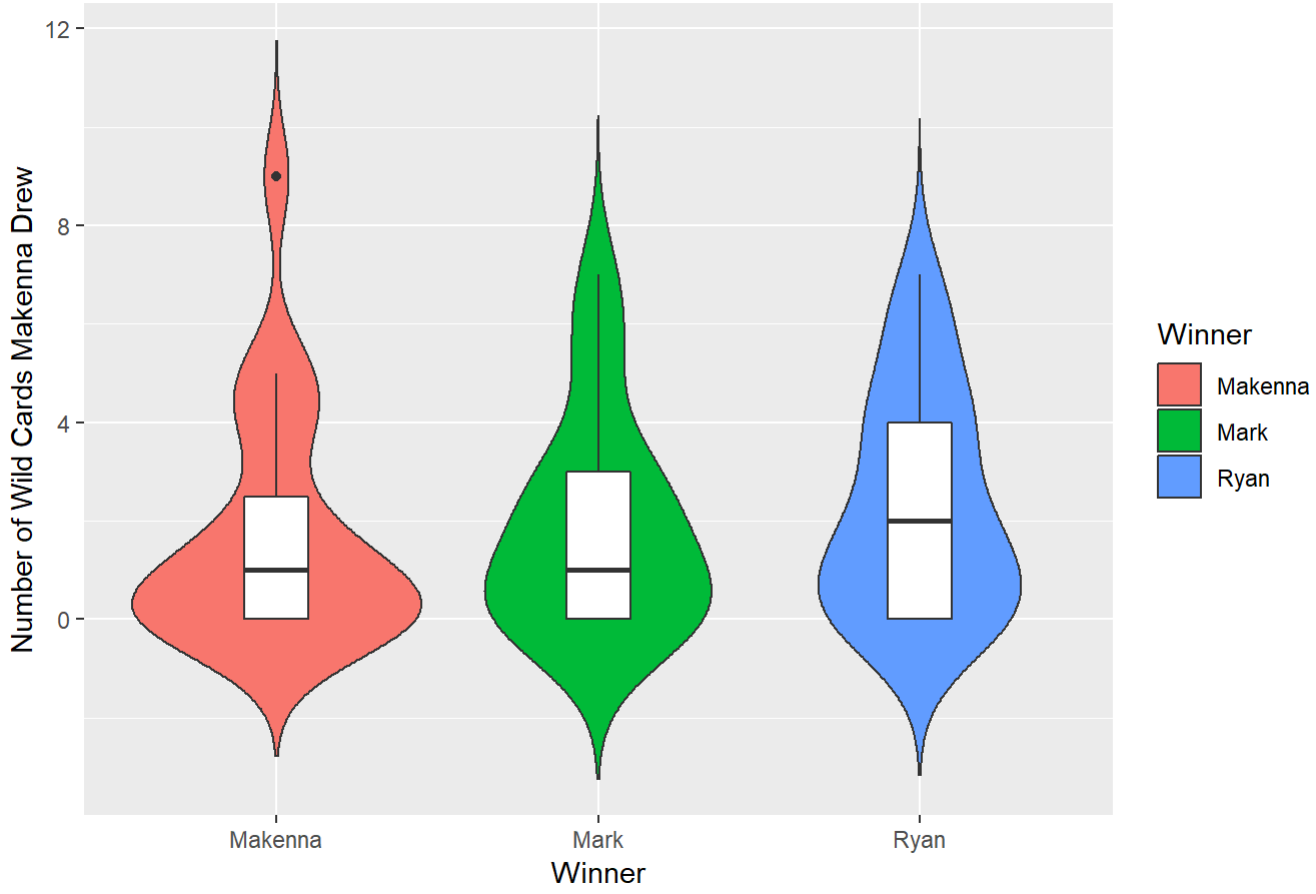
```
## [1] 2.222222
```

Based on the information present in our study, we can see that the losers draw more special and color change cards compared to the winner of a game. Despite the power of these wild cards, the name of the game is to get rid of all your cards, and the more cards drawn, the farther away the player goes from winning. However, this difference between Wild Cards drawn by the winner and the losers is very small (difference = 0.4), and this appears to highlight the luck and randomness that is associated with the game of Crazy-8.

In addition, the graphs below appear to show no relationship between the number of wild cards drawn based on whether the player won or lost the game.

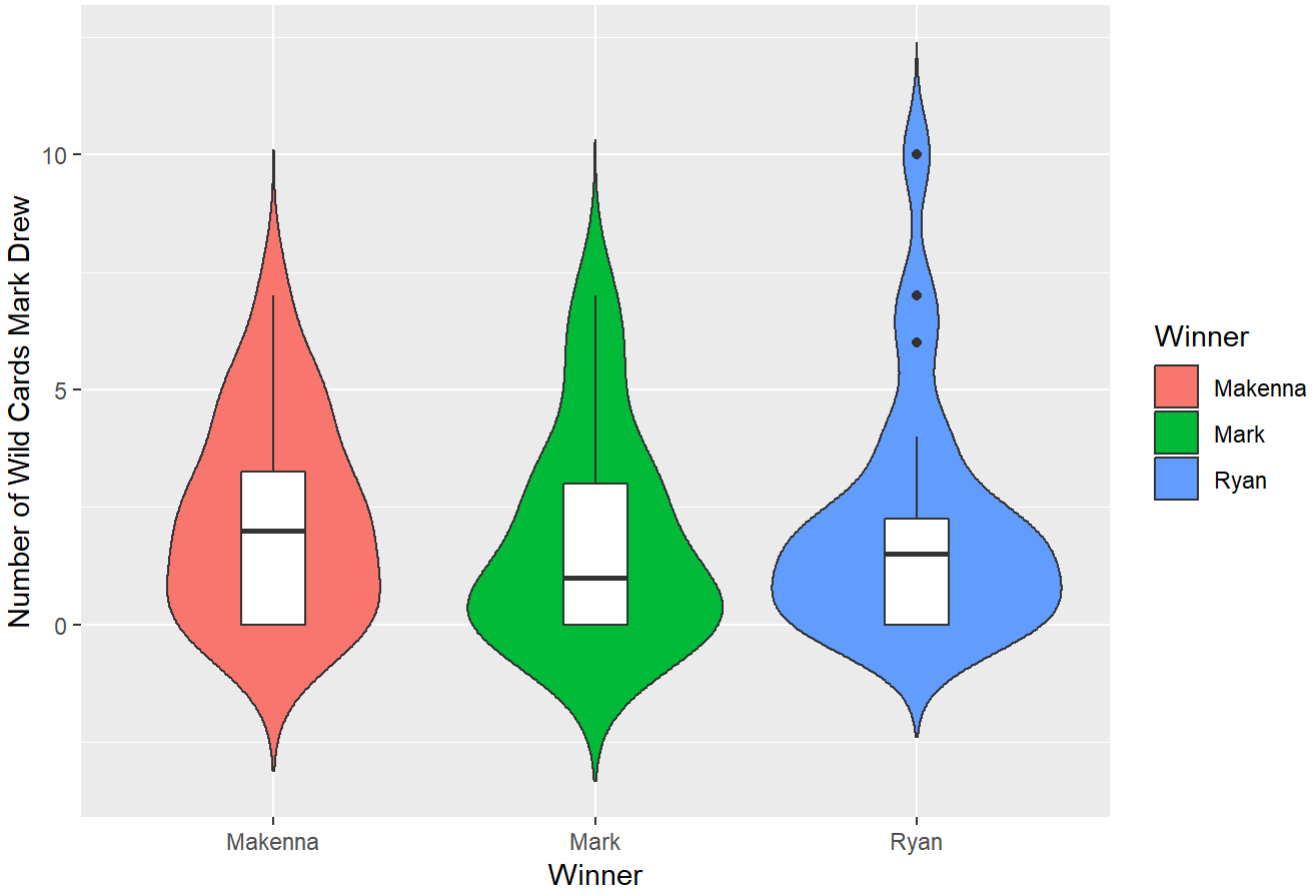
```
ggplot(data = df, aes(x = Winner, y = Wild.Drawn.Makenna, fill = Winner))+
  geom_violin(trim=FALSE)+
  geom_boxplot(width = 0.2, fill = "white")+
  ggtitle("Figure 11: Number of Wild Cards Makenna Drew Sorted by Game Winner")+
  xlab("Winner")+
  ylab("Number of Wild Cards Makenna Drew")
```

Figure 11: Number of Wild Cards Makenna Drew Sorted by Game Winner



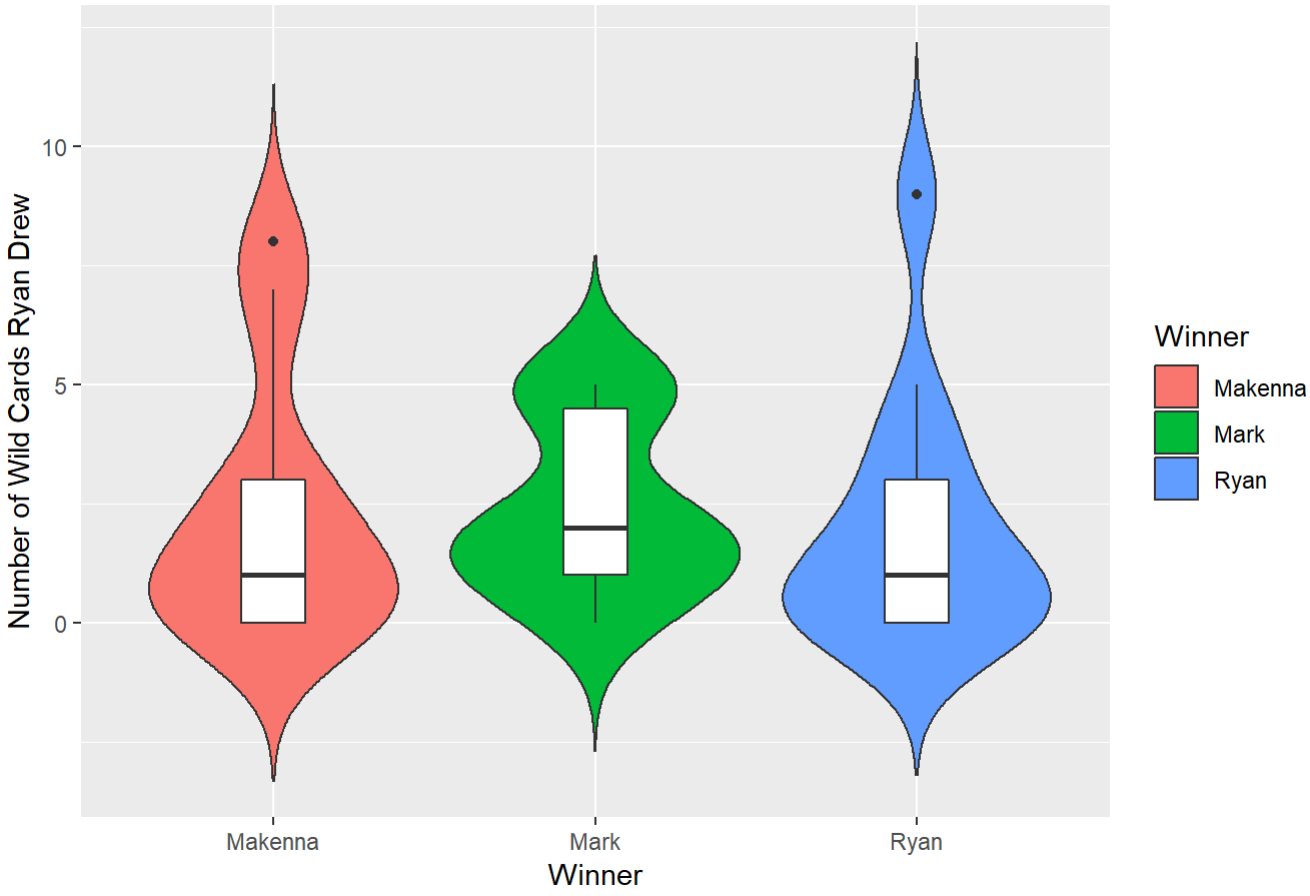

```
ggplot(data = df, aes(x = Winner, y = Wild.Drawn.Mark, fill = Winner)) +
  geom_violin(trim = FALSE)+
  geom_boxplot(width = 0.2, fill = "white") +
  ggtitle("Figure 12: Number of Wild Cards Mark Drew Sorted by Game Winner") +
  xlab("Winner") +
  ylab("Number of Wild Cards Mark Drew")
```

Figure 12: Number of Wild Cards Mark Drew Sorted by Game Winner




```
ggplot(data = df, aes(x = Winner, y = Wild.Drawn.Ryan, fill = Winner)) +
  geom_violin(trim = FALSE)+
  geom_boxplot(width = 0.2, fill = "white") +
  ggtitle("Figure 13: Number of Wild Cards Ryan Drew Sorted by Game Winner") +
  xlab("Winner")+
  ylab("Number of Wild Cards Ryan Drew")
```

Figure 13: Number of Wild Cards Ryan Drew Sorted by Game Winner



We will once again use the bootstrap method to confrim the relationships seen in the graph.

```
pop1 <- df$Wild.Drawn.Makenna[df$Winner == "Makenna"]
pop2 <- df$Wild.Drawn.Makenna[df$Winner == "Mark"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.8409
```

```
pop1 <- df$Wild.Drawn.Mark[df$Winner == "Mark"]
pop2 <- df$Wild.Drawn.Mark[df$Winner == "Ryan"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.6934
```

```
pop1 <- df$Wild.Drawn.Ryan[df$Winner == "Ryan"]
pop2 <- df$Wild.Drawn.Ryan[df$Winner == "Makenna"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

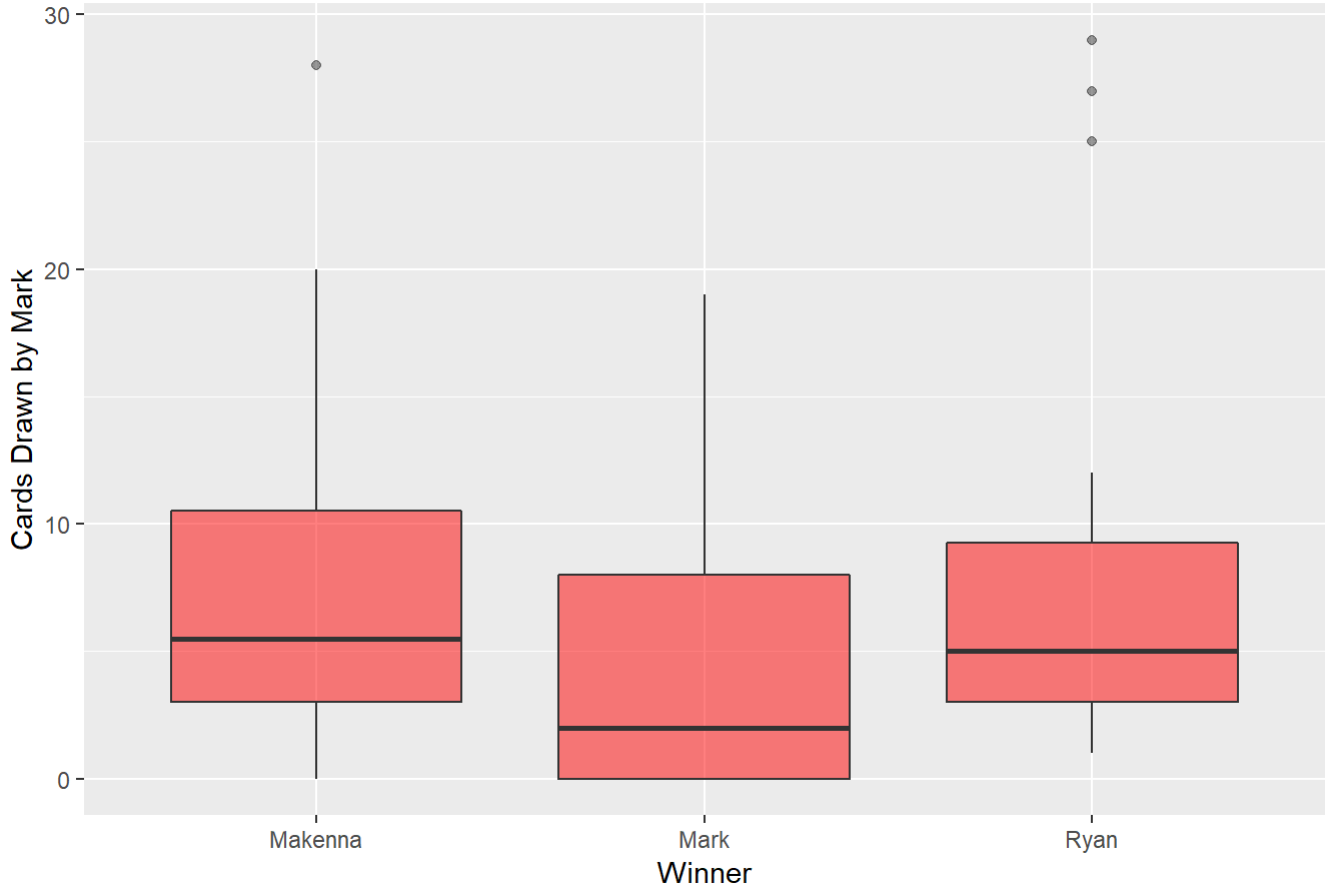
```
## [1] 0.6564
```

The high p-values resulting from the bootstrap say that we fail to reject the null hypothesis that there is no difference in the wild cards drawn among the winner and losers of a game. Hence, we can confirm the relationships seen in the graphs and say that the number of wild cards drawn is almost insignificant to winning or losing a game of Crazy-8s.

In addition, we will now assess the correlation between total cards drawn and winning or losing games.

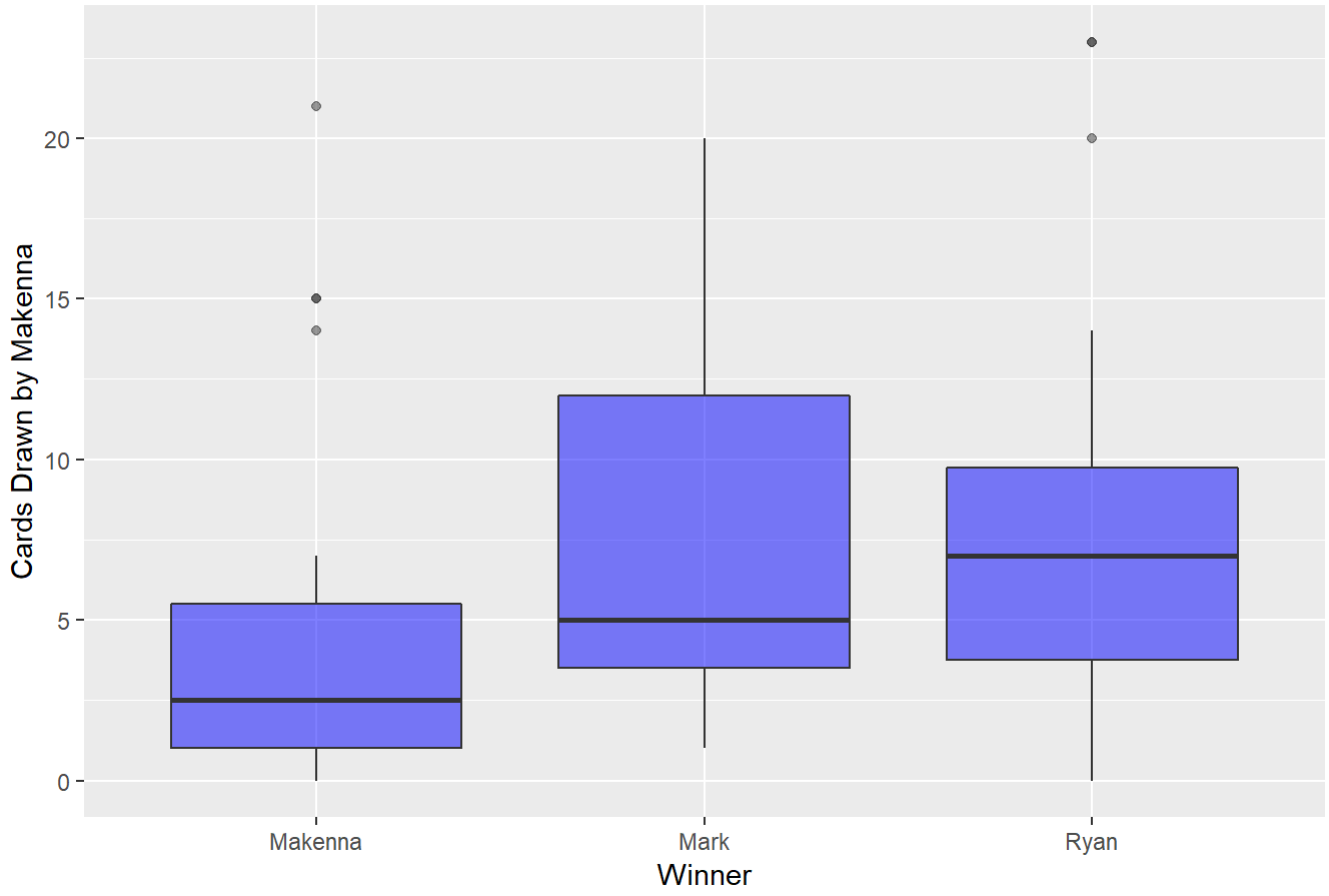
```
ggplot(data = df) +
  geom_boxplot(aes(x = Winner, y = Cards.Drawn.Mark), fill = "red", alpha = 0.5) +
  ggtitle("Figure 14: Cards Drawn by Mark sorted by Game Winner") +
  xlab("Winner") +
  ylab("Cards Drawn by Mark")
```

Figure 14: Cards Drawn by Mark sorted by Game Winner



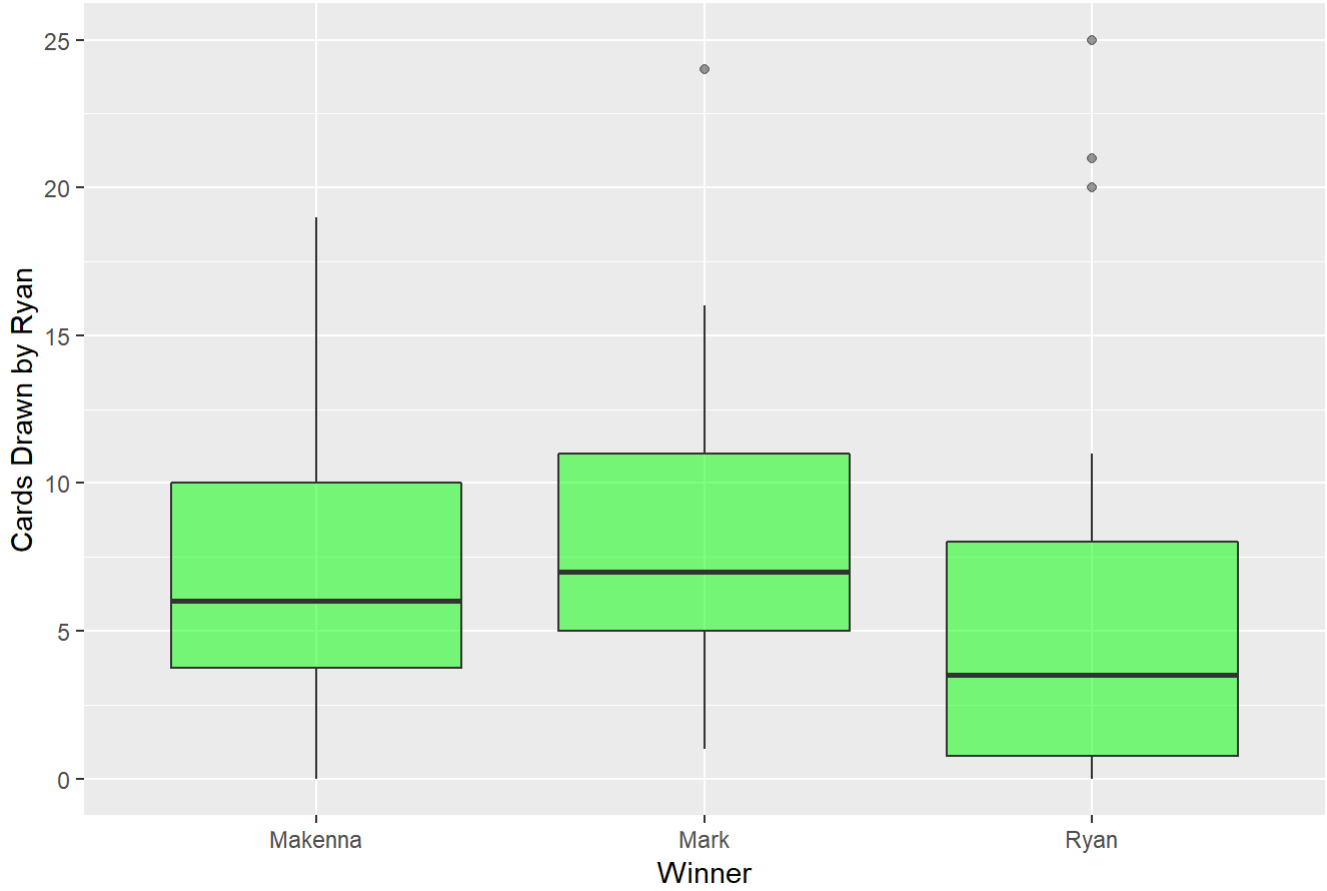

```
ggplot(data = df) +
  geom_boxplot(aes(x = Winner, y = Cards.Drawn.Makenna), fill = "blue", alpha = 0.5) +
  ggtitle("Figure 15: Cards Drawn by Makenna sorted by Game Winner") +
  xlab("Winner") +
  ylab("Cards Drawn by Makenna")
```

Figure 15: Cards Drawn by Makenna sorted by Game Winner




```
ggplot(data = df) +
  geom_boxplot(aes(x = Winner, y = Cards.Drawn.Ryan), fill = "green", alpha = 0.5) +
  ggtitle("Figure 16: Cards Drawn by Ryan sorted by Game Winner") +
  xlab("Winner") +
  ylab("Cards Drawn by Ryan")
```

Figure 16: Cards Drawn by Ryan sorted by Game Winner




```
summary(aov(Cards.Drawn.Mark ~ Winner, data = df))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Winner       2  137.6   68.80   1.394  0.256
## Residuals   60 2962.3   49.37
```

The graphs appear to show that the winner of the games drew less cards overall than the other two players. We will now use the bootstrapping method to confirm whether this difference in cards drawn by the winner and losers is significant.

```
pop1 <- df$Cards.Drawn.Makenna[df$Winner == "Makenna"]
pop2 <- df$Cards.Drawn.Makenna[df$Winner == "Mark"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.1439
```

```
pop1 <- df$Cards.Drawn.Mark[df$Winner == "Mark"]
pop2 <- df$Cards.Drawn.Mark[df$Winner == "Ryan"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.0997
```

```
pop1 <- df$Cards.Drawn.Ryan[df$Winner == "Ryan"]
pop2 <- df$Cards.Drawn.Ryan[df$Winner == "Makenna"]
sample.statistic <- abs(mean(pop1) - mean(pop2))
experiment <- Vectorize(function(x=1) {
  mu1 <- mean(sample(pop1,length(pop1),replace = TRUE))
  mu2 <- mean(sample(pop2, length(pop2),replace = TRUE))
  mu1-mu2
  })
resample.data <- experiment(1:10000)
adjusted.resample.data <- resample.data - mean(resample.data)
pvalue <- length(which(abs(adjusted.resample.data)>sample.statistic))/length(adjusted.resample.d
ata)
pvalue
```

```
## [1] 0.3797
```

Althought these p-values are low, they are still greater than 0.05. Hence, we fail to reject the null hypothesis that there is no difference in total cards drawn between the winners and losers of the game. However, we cannot confirm that our null hypothesis is true, and the boxplots show a small relationship between total cards drawn and winning or losing games. The difference is not significant, but there appears to be a relationship between drawing less cards leads to winning more games.

Conclusion

Ultimately, our data exploration appears to confirm that the game of Crazy-8s is very much luck-based and random. There appears to be a lack of independent variables that can predict who wins and loses a game of Crazy-8. Although a small correlation can be seen with drawing less cards leading to winning more games, this relationship does not appear to be significant. As expected, winning and losing a game like Crazy-8s or UNO is very much based on how lucky an individual gets. Especially with a game like Crazy-8s that eliminates the human element of shuffling cards, things are completely random and winning is based on luck.

```
df <- df %>% mutate(id = row_number())
traindf <- df %>% sample_frac(0.5)
testdf <- df %>% anti_join(traindf,by='id')
write.csv(testdf,"test.csv",row.names = FALSE)
rm(testdf)
traindf$id <- NULL
```

```
newtraindf <- traindf %>%
  select(Cards.Drawn.Mark, Cards.Drawn.Makenna, Cards.Drawn.Ryan, Wild.Beg.Mark, Wild.Beg.Makenn
a, Wild.Beg.Ryan, Winner)
```

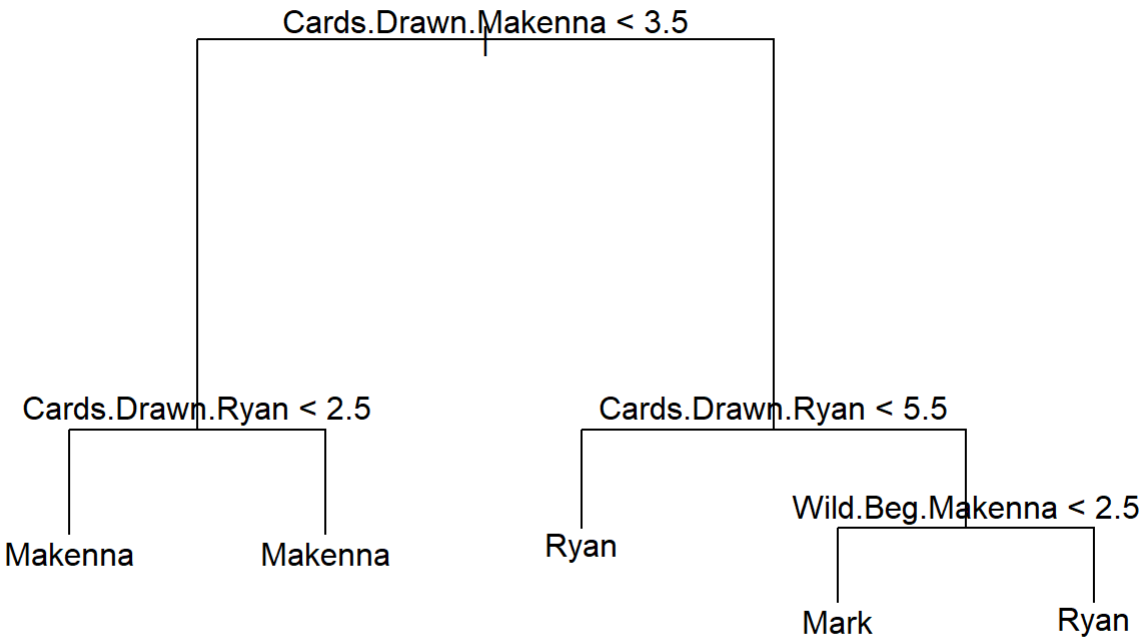
```
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.5.3
```

```
mymodel <- tree(Winner ~ ., data=newtraindf)
summary(mymodel)
```

```
##
## Classification tree:
## tree(formula = Winner ~ ., data = newtraindf)
## Variables actually used in tree construction:
## [1] "Cards.Drawn.Makenna" "Cards.Drawn.Ryan"    "Wild.Beg.Makenna"
## Number of terminal nodes:  5
## Residual mean deviance:  1.354 = 36.55 / 27
## Misclassification error rate: 0.25 = 8 / 32
```

```
plot(mymodel)
text(mymodel,pretty=0)
```




```
newtraindf$pred <- predict(mymodel,type="class")
table(newtraindf$Winner,newtraindf$Winner)
```

```
##
##           Makenna Mark Ryan
##   Makenna      10    0    0
##   Mark          0    9    0
##   Ryan          0    0   13
```

```
testd4 <- read.csv("test.csv")
testd4$id <- NULL
```

```
newtestd4 <- testd4 %>%
  select(Cards.Drawn.Mark, Cards.Drawn.Makenna, Cards.Drawn.Ryan, Wild.Beg.Mark, Wild.Beg.Makenn
a, Wild.Beg.Ryan, Winner)
```

```
newtestd4$pred <- predict(mymodel, newtestd4, type="class")
table(newtestd4$pred,newtestd4$Winner)
```

```
##
##           Makenna Mark Ryan
##   Makenna       5    3    5
##   Mark          3    3    1
##   Ryan          2    4    5
```