Final Project: Chess Dataset Analysis

Isabella Sturm

3/4/2021

Load the data

```
# Load libraries
library(ggplot2)
```

```
## Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
## when loading 'dplyr'
```

```
library(rsample)

# Load the data
setwd('/Users/isabellasturm/Documents/DataScience/DSC520Winter2020/ChessFinal/')
full_chess_df <- read.csv('games.csv')
```

Clean the data

```
# Clean the data
# Remove unneeded columns
chess_df <- subset(full_chess_df, select=c(id, created_at, last_move_at, turns,
                                           victory_status, winner, white_rating,
                                           black_rating, moves, opening_eco,
                                           opening_name, opening_ply))

# Set factors: victory_status
chess_df$victory_status <- as.factor(chess_df$victory_status)
# Set winner to numeric: white = 1, black = 0, draw = -1
chess_df$winner_num[chess_df$winner == "white"] <- 1
chess_df$winner_num[chess_df$winner == 'black'] <- 0
chess_df$winner_num[chess_df$winner == 'draw'] <- -1
chess_df$winner_num <- as.numeric(chess_df$winner_num)
# Set dates: created_at, last_move_at
#chess_df$created_at <- as.Date.numeric(chess_df$created_at, origin='1970-01-01')
#chess_df$last_move_at<- as.Date.numeric(chess_df$last_move_at, origin = '1970-01-01')
# Create new column: game_length, rating_diff
chess_df$game_length <- (chess_df$last_move_at - chess_df$created_at) / 1000 / 60 # Time diff in minute
chess_df$rating_diff <- chess_df$white_rating - chess_df$black_rating # negative if black is rated high

chess_df <- chess_df[chess_df$game_length < (2.5e+04/60),] # Remove the potential outliers to see the p


# Split the dataset for later
```

```r
sample <- initial_split(chess_df, prop=.70)
train <- training(sample)
test <- testing(sample)
```

Data Summary

```r
head(chess_df)
```

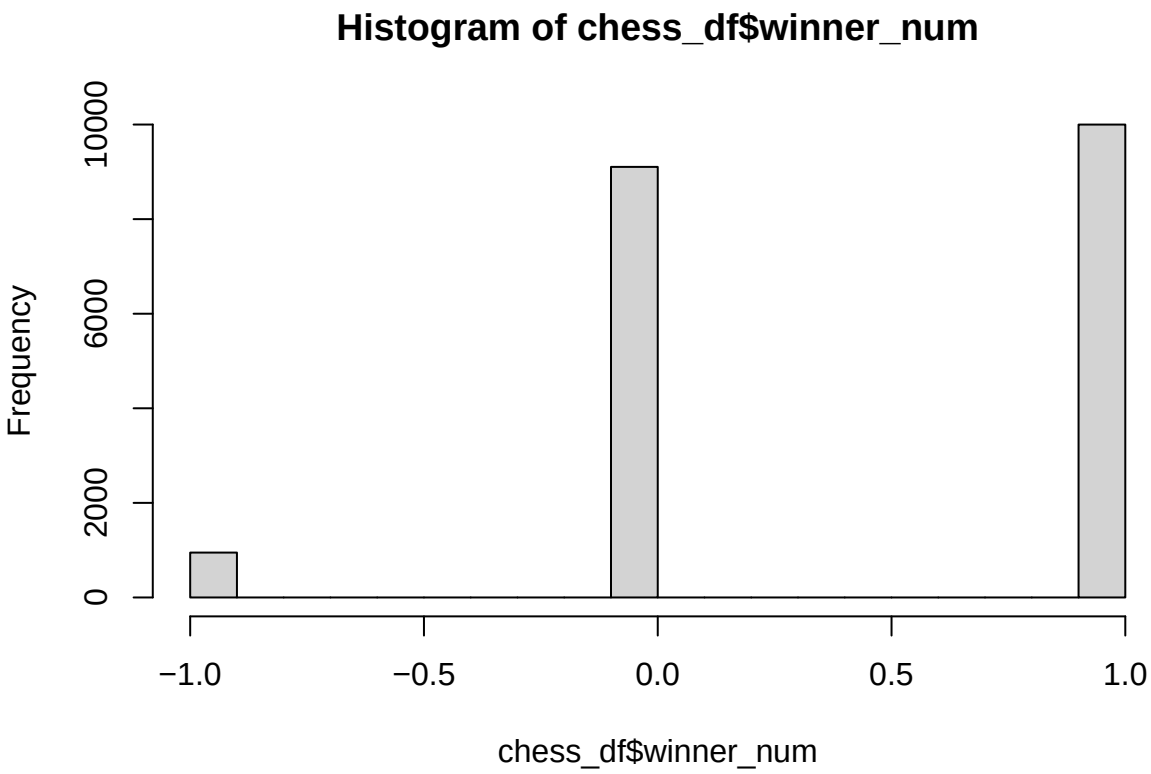
```
##         id  created_at last_move_at turns victory_status winner white_rating
## 1 TZJHLljE 1.50421e+12  1.50421e+12    13      outoftime  white         1500
## 2 l1NXvwaE 1.50413e+12  1.50413e+12    16         resign  black         1322
## 3 mIICvQHh 1.50413e+12  1.50413e+12    61           mate  white         1496
## 4 kWKvrqYL 1.50411e+12  1.50411e+12    61           mate  white         1439
## 5 9tXo1AUZ 1.50403e+12  1.50403e+12    95           mate  white         1523
## 6 MsoDV9wj 1.50424e+12  1.50424e+12     5           draw   draw         1250
##   black_rating
## 1         1191
## 2         1261
## 3         1500
## 4         1454
## 5         1469
## 6         1002
##
## 1
## 2
## 3
## 4
## 5 e4 e5 Nf3 d6 d4 Nc6 d5 Nb4 a3 Na6 Nc3 Be7 b4 Nf6 Bg5 O-O b5 Nc5 Bxf6 Bxf6 Bd3 Qd7 O-O Nxd3 Qxd3 c6
## 6
##   opening_eco                          opening_name opening_ply winner_num
## 1         D10       Slav Defense: Exchange Variation           5          1
## 2         B00 Nimzowitsch Defense: Kennedy Variation           4          0
## 3         C20  King's Pawn Game: Leonardis Variation           3          1
## 4         D02 Queen's Pawn Game: Zukertort Variation           3          1
## 5         C41                       Philidor Defense           5          1
## 6         B27   Sicilian Defense: Mongoose Variation           4         -1
##   game_length rating_diff
## 1           0         309
## 2           0          61
## 3           0          -4
## 4           0         -15
## 5           0          54
## 6           0         248
```

```r
summary(chess_df)
```

```
##       id              created_at         last_move_at           turns
##  Length:20054       Min.   :1.377e+12   Min.   :1.377e+12   Min.   :  1.00
##  Class :character   1st Qu.:1.478e+12   1st Qu.:1.478e+12   1st Qu.: 37.00
##  Mode  :character   Median :1.496e+12   Median :1.496e+12   Median : 55.00
##                     Mean   :1.484e+12   Mean   :1.484e+12   Mean   : 60.47
##                     3rd Qu.:1.503e+12   3rd Qu.:1.503e+12   3rd Qu.: 79.00
##                     Max.   :1.504e+12   Max.   :1.504e+12   Max.   :349.00
```

```
##    victory_status     winner           white_rating   black_rating
##  draw     :  905   Length:20054       Min.   : 784   Min.   : 789
##  mate     : 6325   Class :character   1st Qu.:1398   1st Qu.:1391
##  outoftime: 1678   Mode  :character   Median :1567   Median :1562
##  resign   :11146                      Mean   :1597   Mean   :1589
##                                       3rd Qu.:1793   3rd Qu.:1784
##                                       Max.   :2700   Max.   :2723
##     moves           opening_eco        opening_name        opening_ply
##  Length:20054       Length:20054       Length:20054       Min.   : 1.000
##  Class :character   Class :character   Class :character   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Mode  :character   Median : 4.000
##                                                           Mean   : 4.817
##                                                           3rd Qu.: 6.000
##                                                           Max.   :28.000
##    winner_num       game_length       rating_diff
##  Min.   :-1.0000   Min.   :  0.000   Min.   :-1605.000
##  1st Qu.: 0.0000   1st Qu.:  0.000   1st Qu.: -108.000
##  Median : 0.0000   Median :  4.007   Median :    3.000
##  Mean   : 0.4513   Mean   : 13.785   Mean   :    7.797
##  3rd Qu.: 1.0000   3rd Qu.: 13.209   3rd Qu.:  122.000
##  Max.   : 1.0000   Max.   :355.027   Max.   : 1499.000
```

```r
hist(chess_df$winner_num)
```



Descriptive Stats

```r
# Mean, median of the white player ratings
mean(chess_df$white_rating)
```

```
## [1] 1596.676
```

```r
median(chess_df$white_rating)
```

```
## [1] 1567
```

```r
# Mean of the black player ratings
mean(chess_df$black_rating)
```

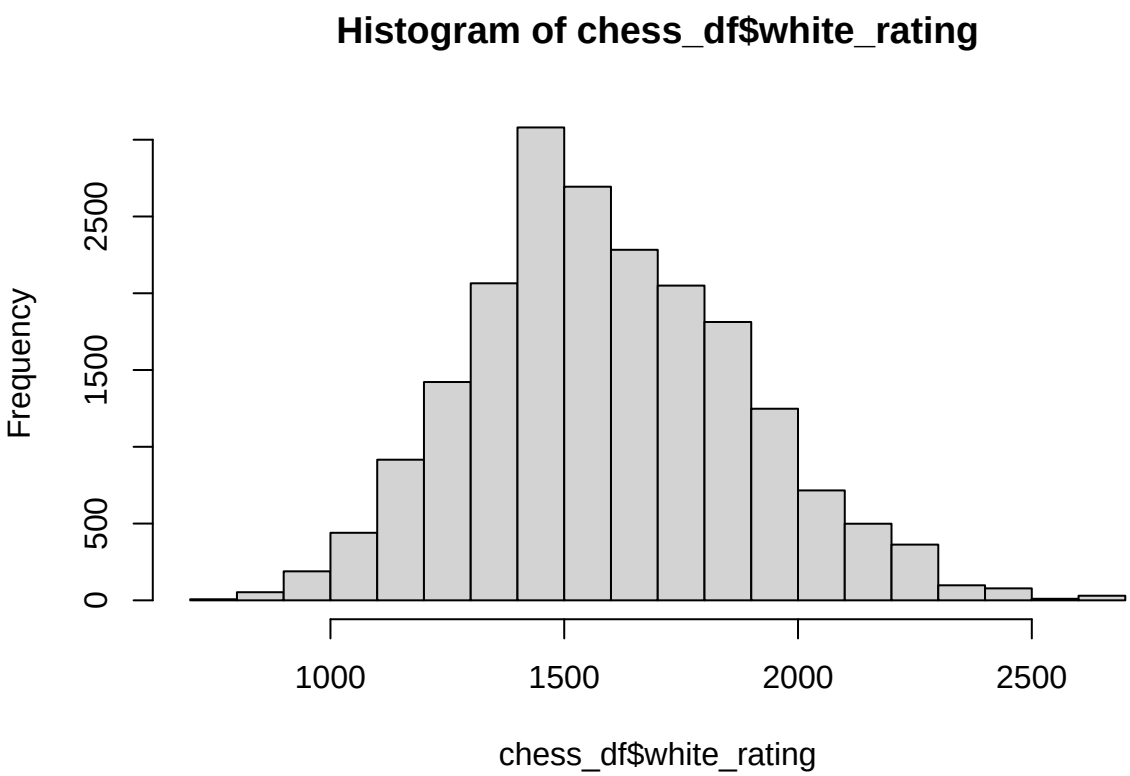
```
## [1] 1588.879
```

```r
median(chess_df$black_rating)
```

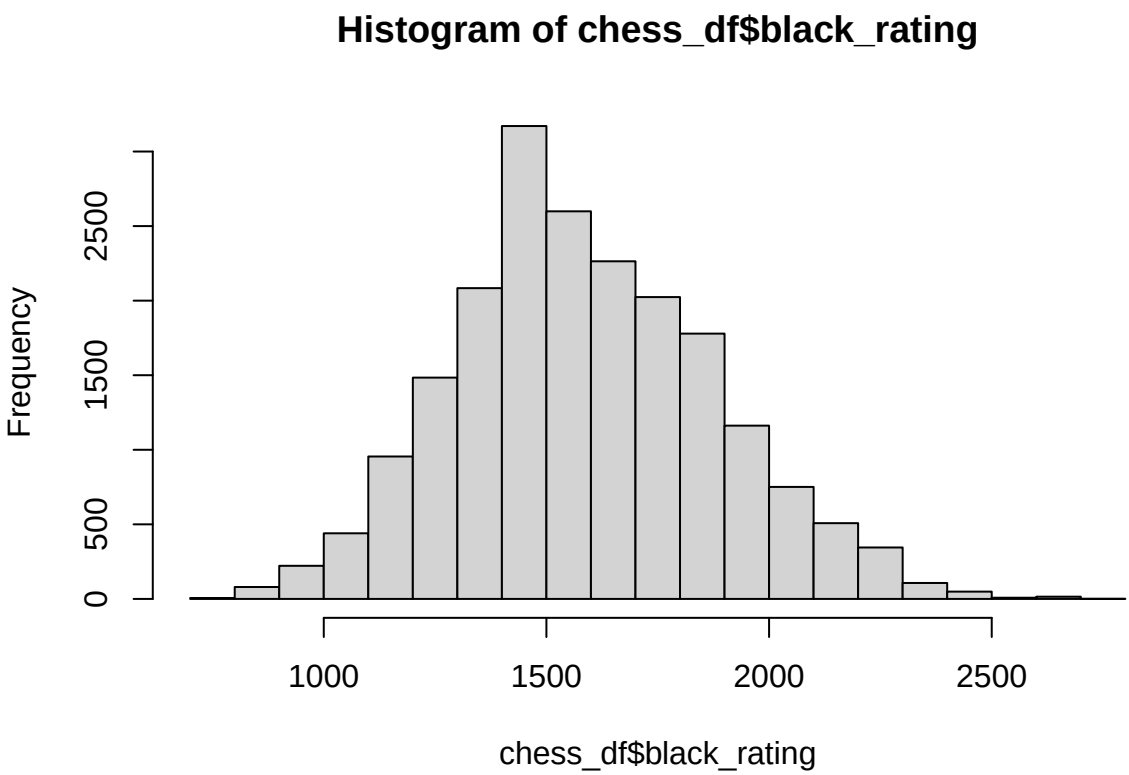
```
## [1] 1562
```

Distributions, Plotting

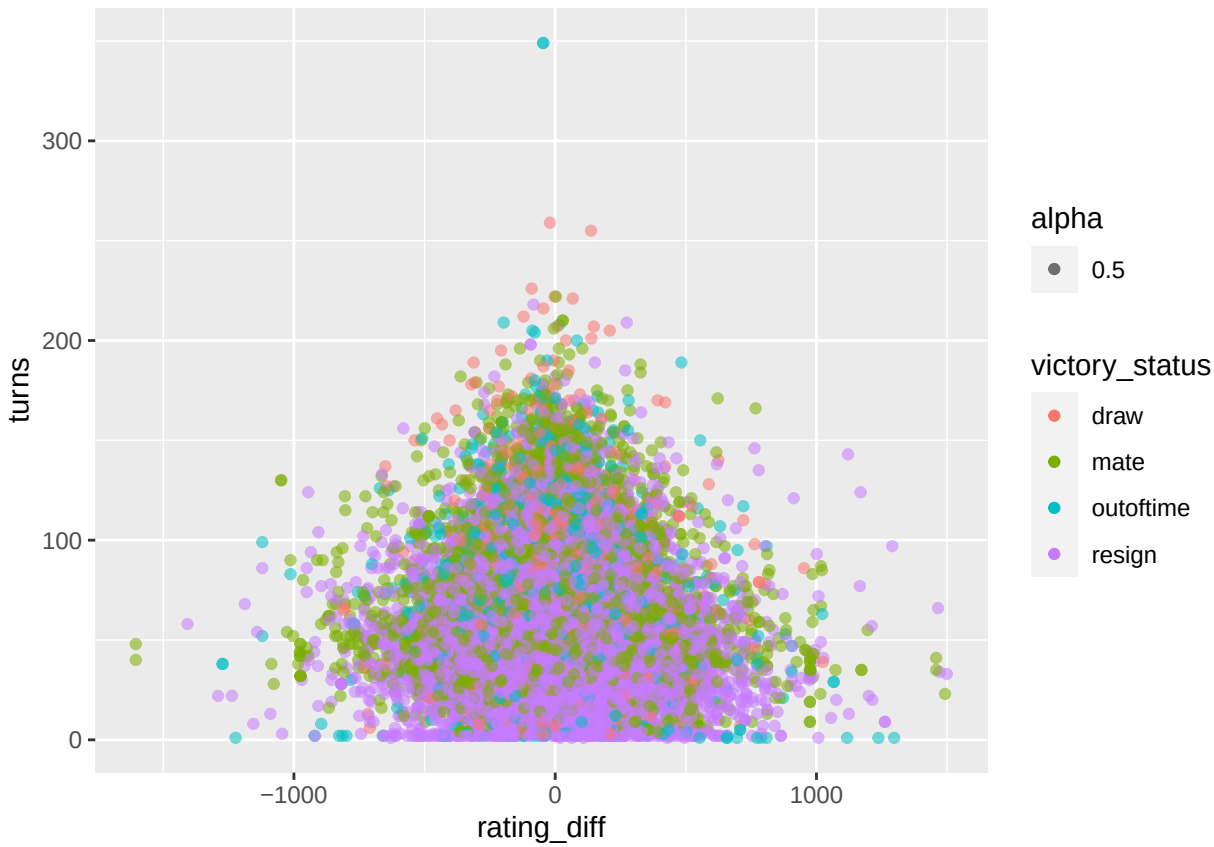
```r
# Frequency of player ratings
hist(chess_df$white_rating)
```



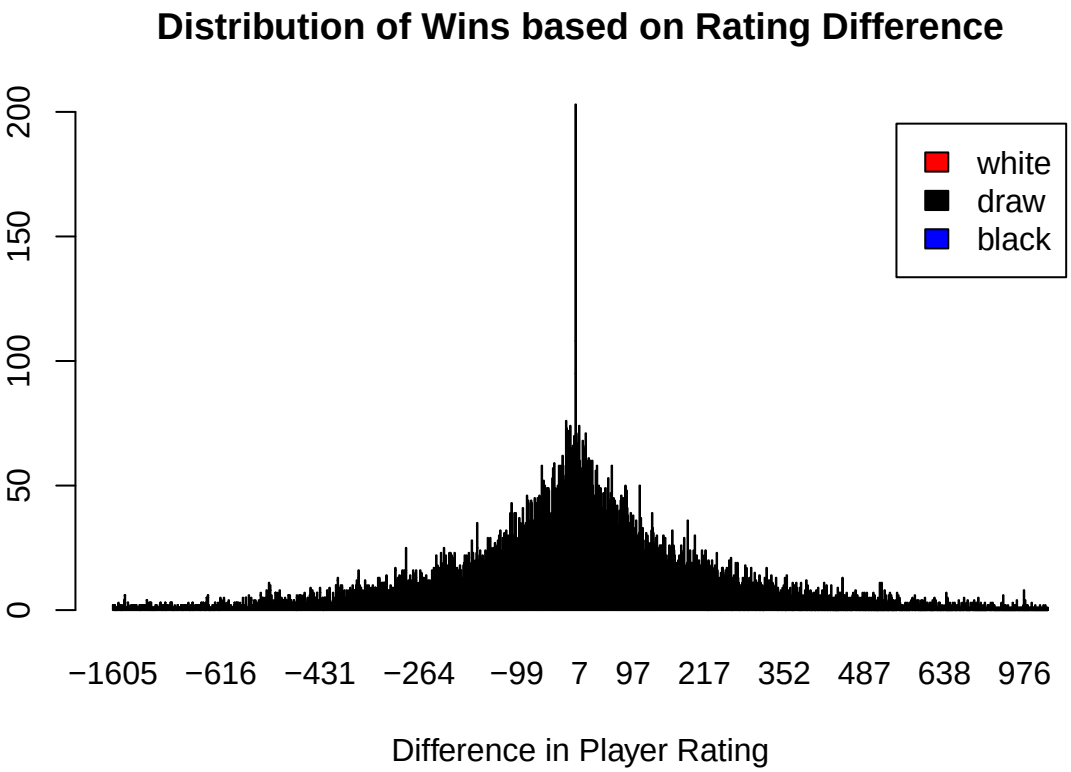

```r
hist(chess_df$black_rating)
```



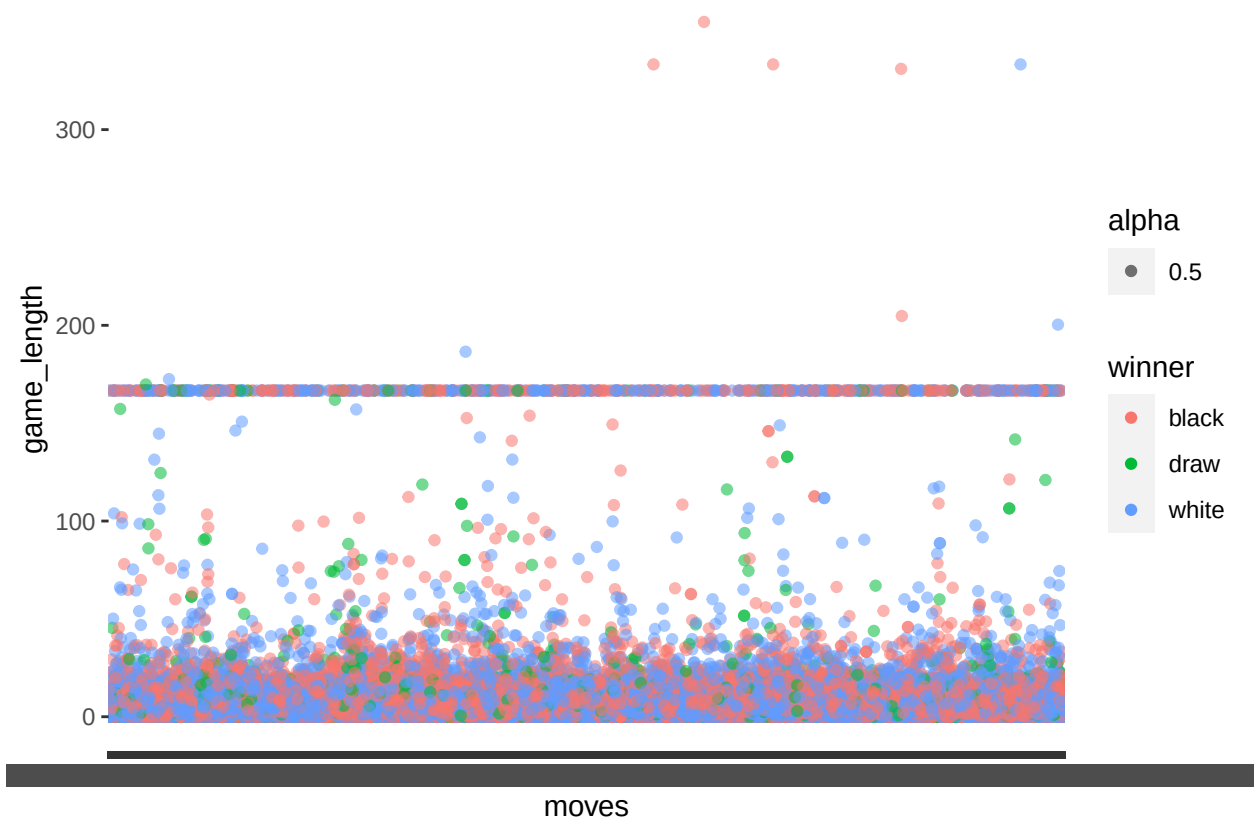

```r
# Scatterplot of player rating difference vs number of moves, color
ggplot(data=chess_df, aes(x=rating_diff, y=turns, color=victory_status, alpha=0.5)) +
        geom_point()
```



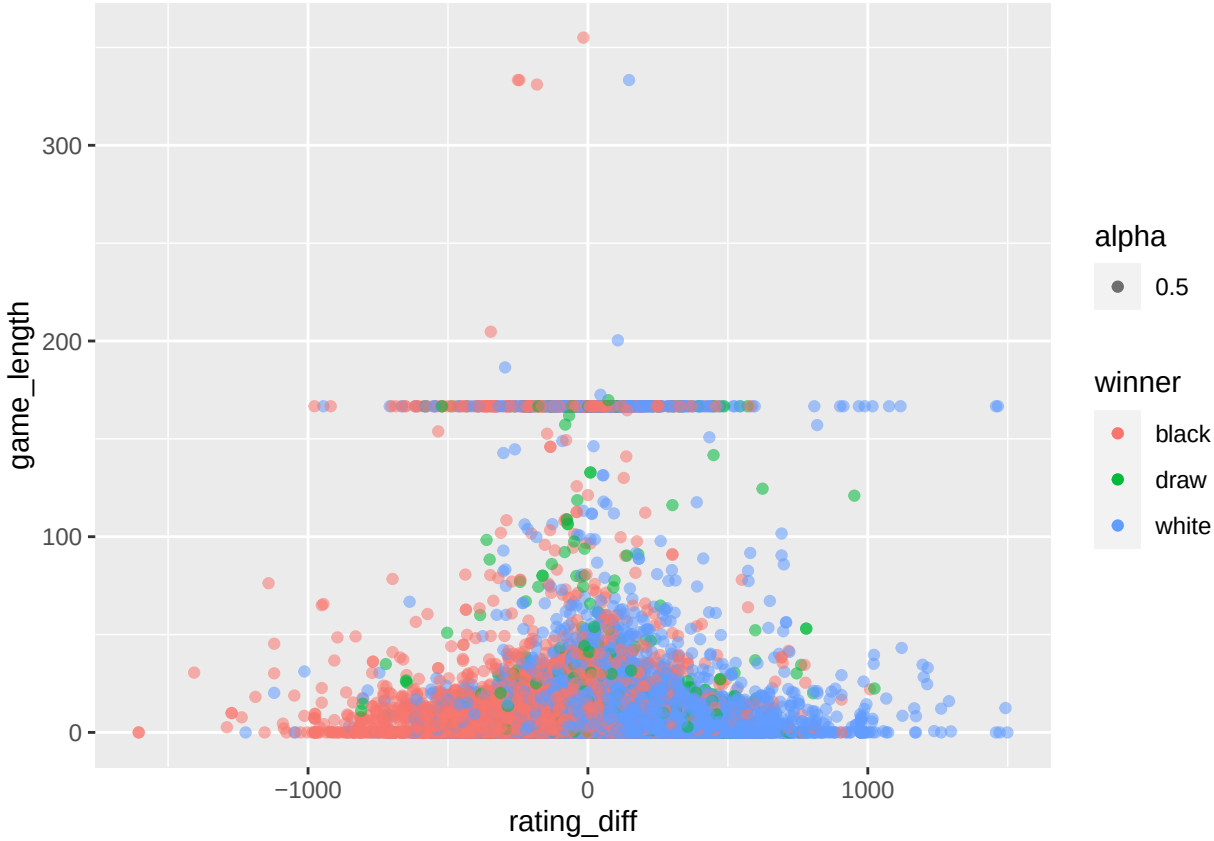

```r
# Barplot of Winner based on Difference in Player Rating
win_counts = table(chess_df$winner, chess_df$rating_diff)
barplot(win_counts, main="Distribution of Wins based on Rating Difference",
        xlab="Difference in Player Rating", col=c("blue", "black", "red"),
        legend=rownames(win_counts))
```




```r
# Plot the number of turns to game duration
ggplot(data=chess_df, aes(x=moves, y=game_length, color=winner, alpha=0.5)) +
  geom_point()
```



```r
# Plot the player rating diff to game duration
ggplot(data=chess_df, aes(x=rating_diff, y=game_length, color=winner, alpha=0.5)) +
  geom_point()
```



Correlation

```r
# Find correlation between game length and game duration
cor(x=chess_df$rating_diff, y=chess_df$winner_num)
```

```
## [1] 0.2972719
```

```r
# Find covariance between rating difference and winner
cov(x=chess_df$rating_diff, y=chess_df$winner_num)
```

```
## [1] 43.3153
```

```r
# Find correlation between rating difference and winner
#cor(x=chess_df$rating_diff, y=chess_df$game_length)

# Find covariance between difference in player rating and game length (duration)
#cov(x=chess_df$rating_diff, y=chess_df$game_length)
```

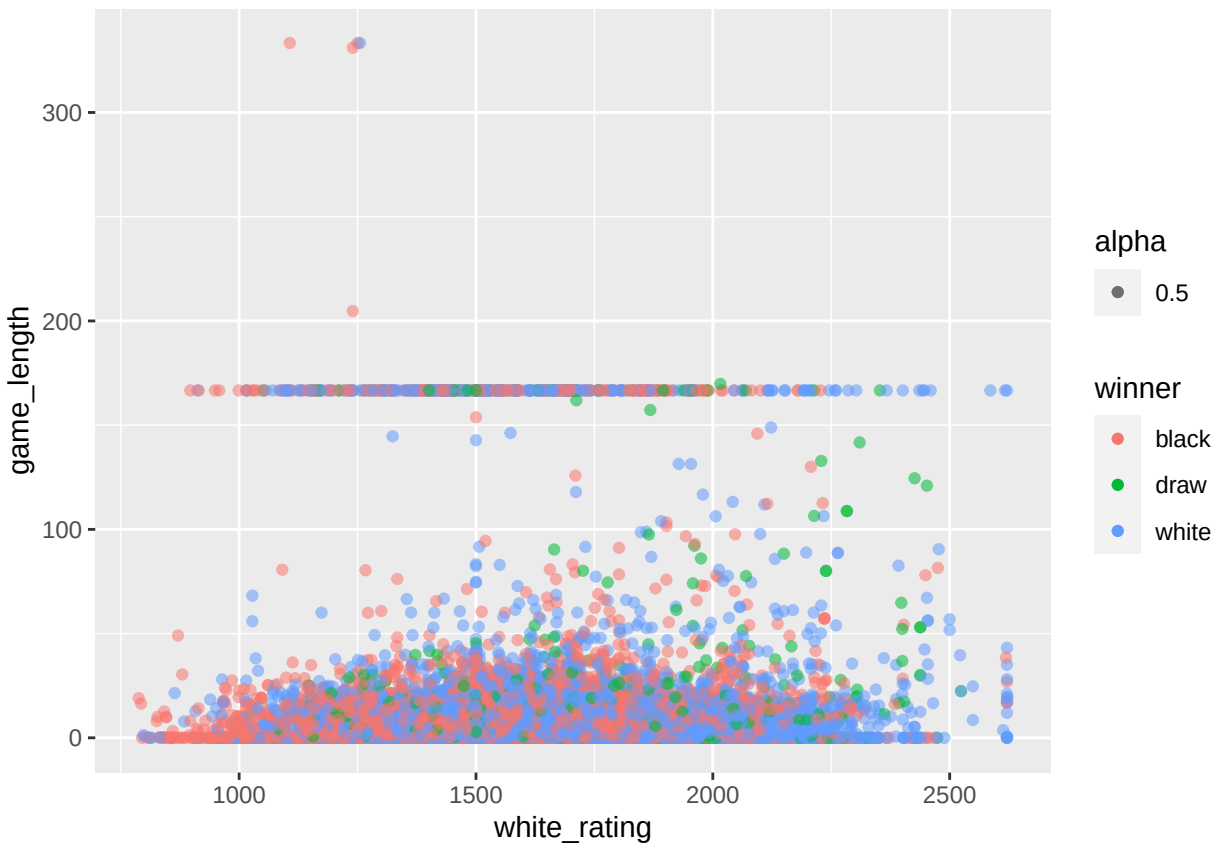
Regression (Linear/Multiple/Logical?)

```r
# Find linear regression model for the chess data
# Winner based on rating and number of turns
```

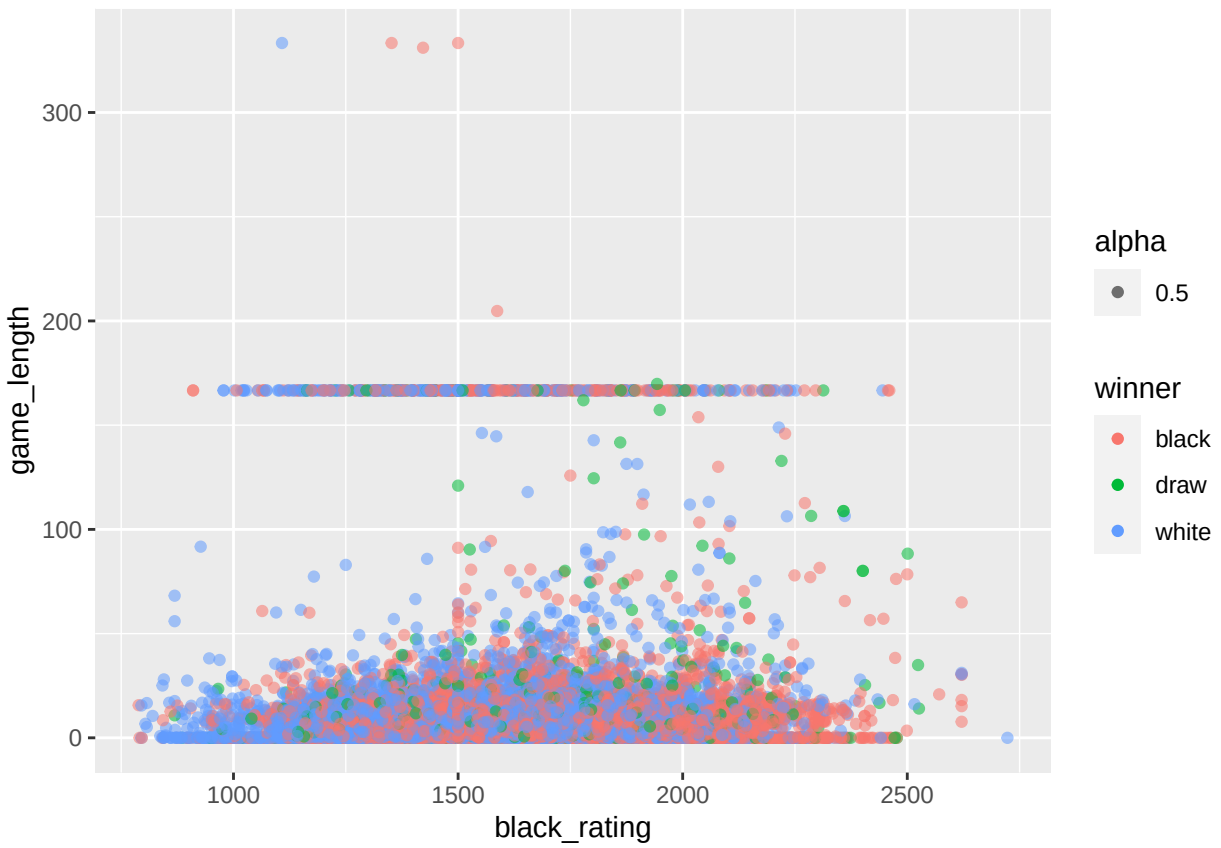
```r
chess_lm = lm(winner_num ~ white_rating + black_rating + turns, data=train)
# Return the summary
summary(chess_lm)
```

```
##
## Call:
## lm(formula = winner_num ~ white_rating + black_rating + turns,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16245 -0.41704  0.00569  0.50434  1.35214
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.511e-01  2.919e-02   22.30   <2e-16 ***
## white_rating  6.567e-04  2.091e-05   31.40   <2e-16 ***
## black_rating -7.124e-04  2.105e-05  -33.84   <2e-16 ***
## turns        -1.988e-03  1.421e-04  -13.99   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5574 on 14034 degrees of freedom
## Multiple R-squared:  0.1008, Adjusted R-squared:  0.1006
## F-statistic: 524.3 on 3 and 14034 DF,  p-value: < 2.2e-16
```

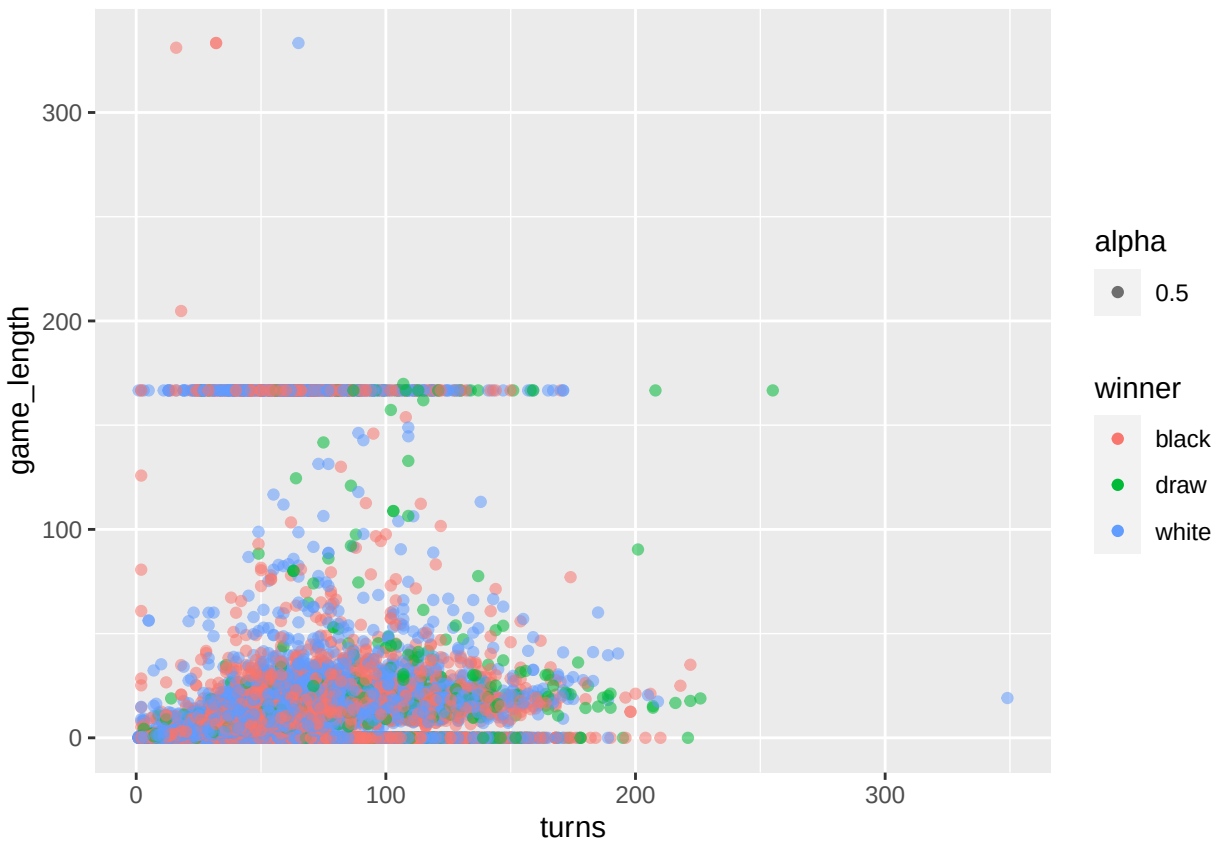
```r
# Plot the data with regression line
ggplot(data=train, aes(x=white_rating, y = game_length, color=winner, alpha=0.5)) +
  geom_point()
```



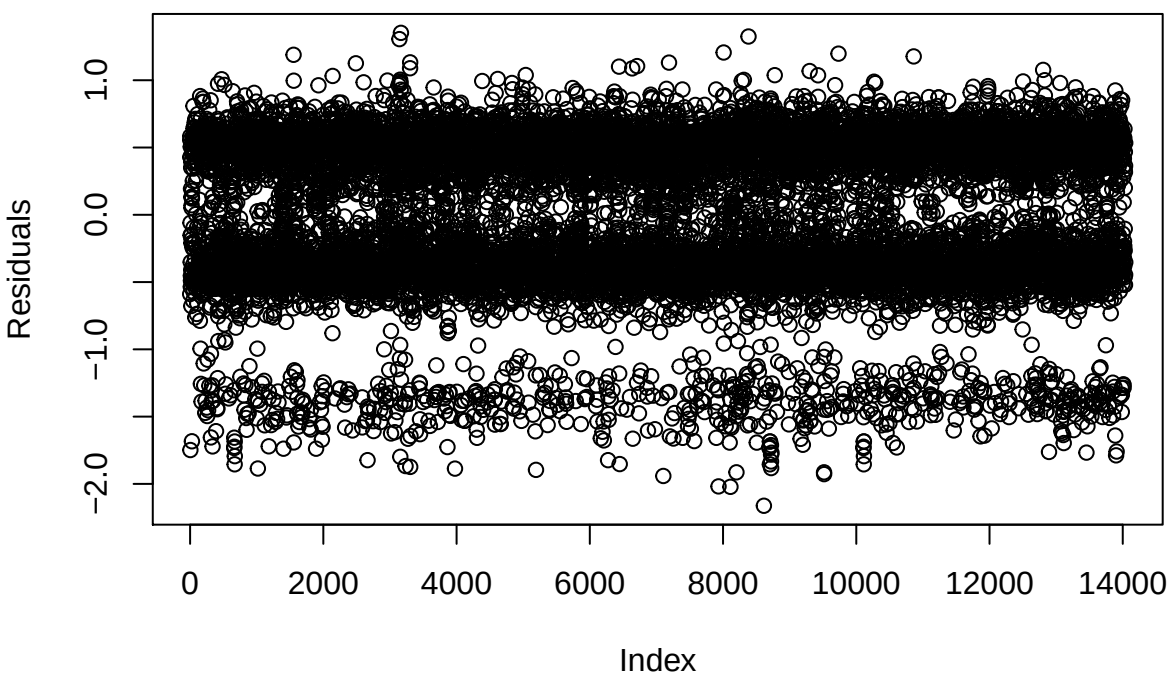

```
ggplot(data=train, aes(x=black_rating, y = game_length, color=winner, alpha=0.5)) +
  geom_point()
```



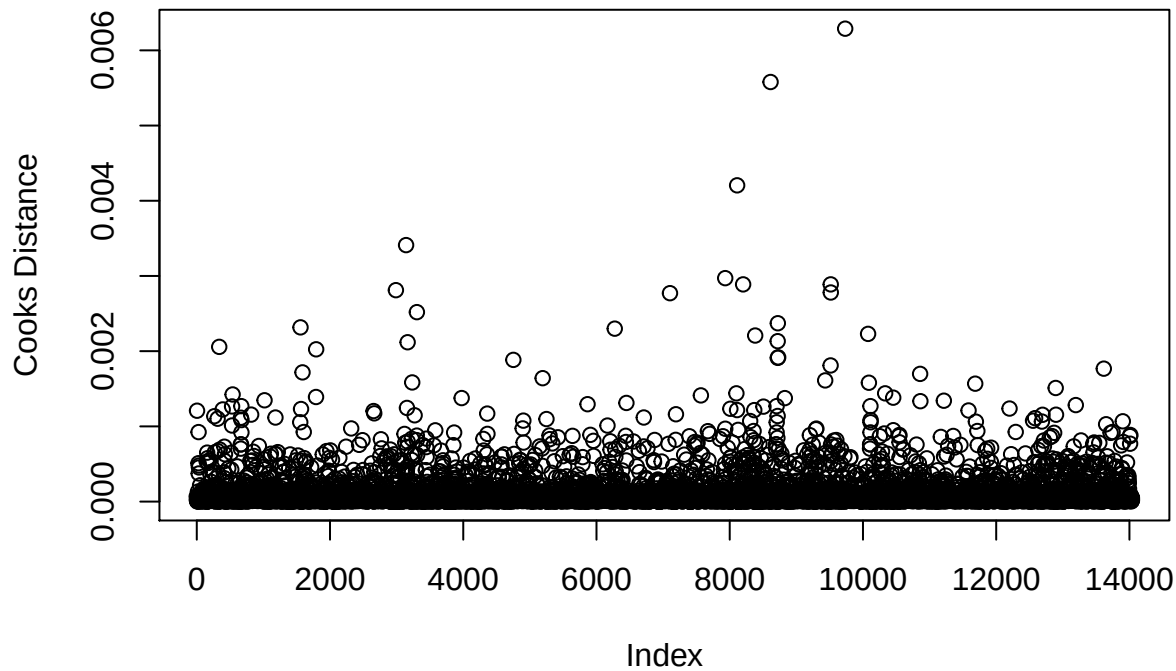

```
ggplot(data=train, aes(x=turns, y=game_length, color=winner, alpha=0.5)) +
  geom_point()
```




```r
# Plot the residuals
plot(chess_lm$residuals, ylab="Residuals")
```




```r
# Plot cooks distance
plot(cooks.distance(chess_lm), ylab="Cooks Distance")
```




```r
# Predict and check accuracy of model
predictions <- predict(chess_lm, test, type="response")
confmat <- table(Actual_Value=test$winner_num,Predicted_Value=predictions > 0.5)
# Validate accuracy
(confmat[[1,1]]) / sum(confmat)
```

```
## [1] 0.0334109
```