Stat3355Project

Nitish Ghosh

3/27/2021

Cleaning Data

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(RColorBrewer)
chess <- read.csv("chess.csv", header = TRUE)

# Deleting the unnecessary variables
chess$created_at <- NULL
chess$last_move_at <- NULL

# New column for difference in rating of both players
diff_rating = NULL
i <- 1
while(i <= length(chess$id)) {
  diff_rating[i] <- abs(chess$white_rating[i] - chess$black_rating[i])
  i <- i + 1
}
chess <- cbind(chess, diff_rating)

# New column for avg rating of both players
avg_rating = NULL
i <- 1
while(i <= length(chess$id)) {
  avg_rating[i] <- (chess$white_rating[i] + chess$black_rating[i]) / 2
  i <- i + 1
}
chess <- cbind(chess, avg_rating)

# Factorizing game lengths based on quartiles
gamelength = NULL
summary(chess$turns)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   37.00   55.00   60.47   79.00  349.00
```

```
gamelength <- cut(chess$turns, c(0, 37, 55, 79, 349), labels = c("Quick", "Normal", "Long", "Time-Consu
chess <- cbind(chess, gamelength)

# Factorizing ratings based on quartiles / chess wiki
white_rating_factor = NULL
black_rating_factor = NULL
all_rating <- c(chess$white_rating, chess$black_rating)
summary(all_rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     784    1394    1564    1593    1788    2723
```

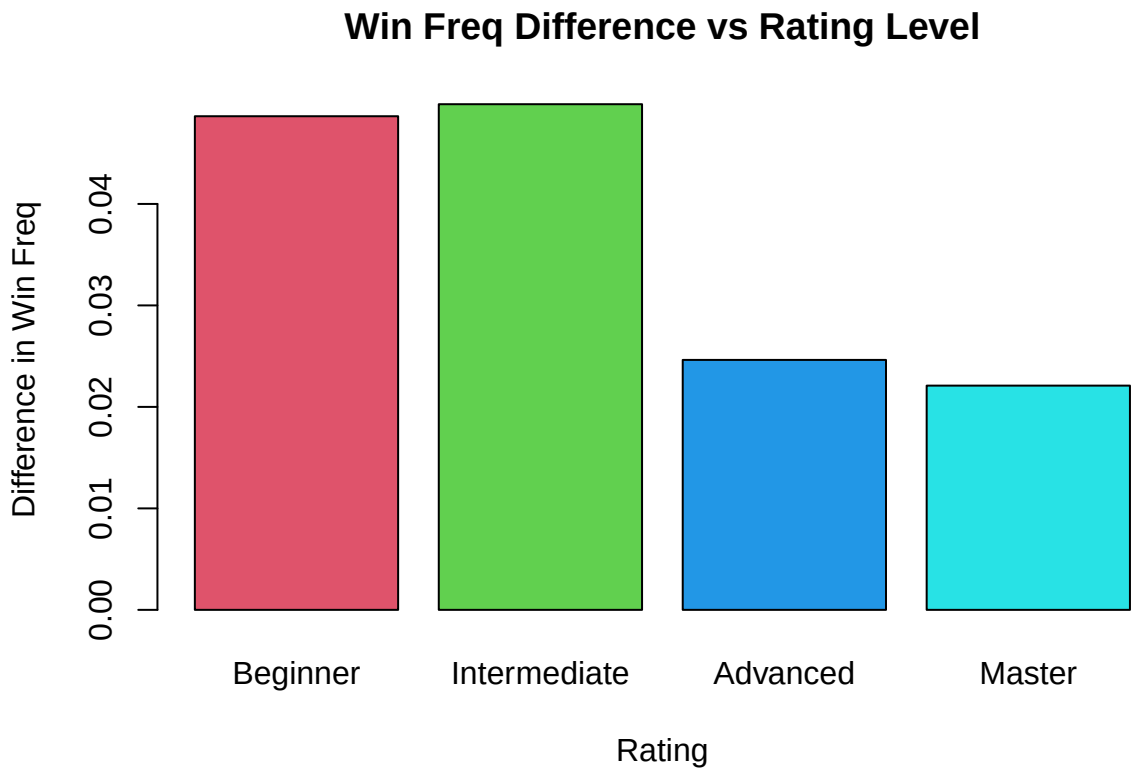
```
white_rating_factor <- cut(chess$white_rating, c(783, 1394, 1788, 2200, 2723), labels = c("Beginner", "
black_rating_factor <- cut(chess$black_rating, c(783, 1394, 1788, 2200, 2723), labels = c("Beginner", "
chess <- cbind(chess, white_rating_factor, black_rating_factor)
avg_rating_factor <- cut(chess$avg_rating, c(783, 1394, 1788, 2200, 2723), labels = c("Beginner", "Inte
chess <- cbind(chess, avg_rating_factor)

# Factorizing rating difference based on quartiles
diff_rating_factor = NULL
summary(chess$diff_rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    45.0   115.0   173.1   241.0  1605.0
```

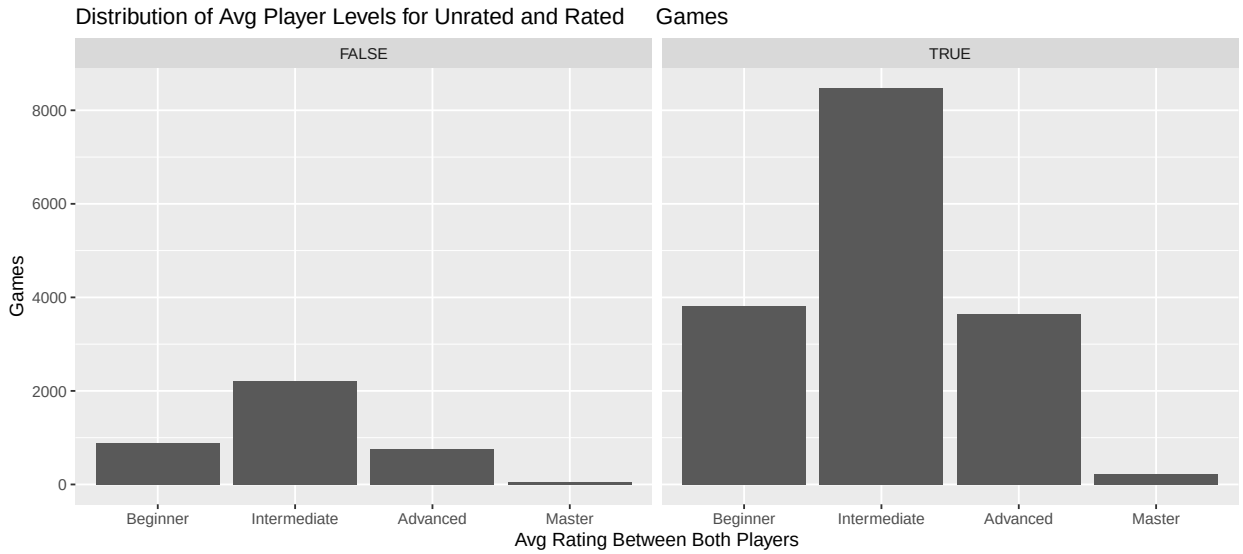
```
diff_rating_factor <- cut(chess$diff_rating, c(-1, 45, 115, 241, 1605), labels = c("Small", "Moderate",
chess <- cbind(chess, diff_rating_factor)

# change victory status to a factor variable.
chess$victory_status <- as.factor(chess$victory_status)
chess$victory_status <- factor(chess$victory_status, levels <- c("draw", "mate", "outoftime", "resign"))
```

```
# Finding win frequencies for each of the colors for each rating level
white_beginner <- which(chess$white_rating_factor %in% ("Beginner"))
white_beginner_win <- which(chess$winner %in% ("white") & chess$white_rating_factor %in% ("Beginner"))
white_beginner_win_freq <- length(white_beginner_win) / length(white_beginner)
white_intermediate <- which(chess$white_rating_factor %in% ("Intermediate"))
white_intermediate_win <- which(chess$winner %in% ("white") & chess$white_rating_factor %in% ("Intermedi
white_intermediate_win_freq <- length(white_intermediate_win) / length(white_intermediate)
white_advanced <- which(chess$white_rating_factor %in% ("Advanced"))
white_advanced_win <- which(chess$winner %in% ("white") & chess$white_rating_factor %in% ("Advanced"))
white_advanced_win_freq <- length(white_advanced_win) / length(white_advanced)
white_master <- which(chess$white_rating_factor %in% ("Master"))
white_master_win <- which(chess$winner %in% ("white") & chess$white_rating_factor %in% ("Master"))
white_master_win_freq <- length(white_master_win) / length(white_master)

black_beginner <- which(chess$black_rating_factor %in% ("Beginner"))
black_beginner_win <- which(chess$winner %in% ("black") & chess$black_rating_factor %in% ("Beginner"))
black_beginner_win_freq <- length(black_beginner_win) / length(black_beginner)
black_intermediate <- which(chess$black_rating_factor %in% ("Intermediate"))
black_intermediate_win <- which(chess$winner %in% ("black") & chess$black_rating_factor %in% ("Intermedi
black_intermediate_win_freq <- length(black_intermediate_win) / length(black_intermediate)
```

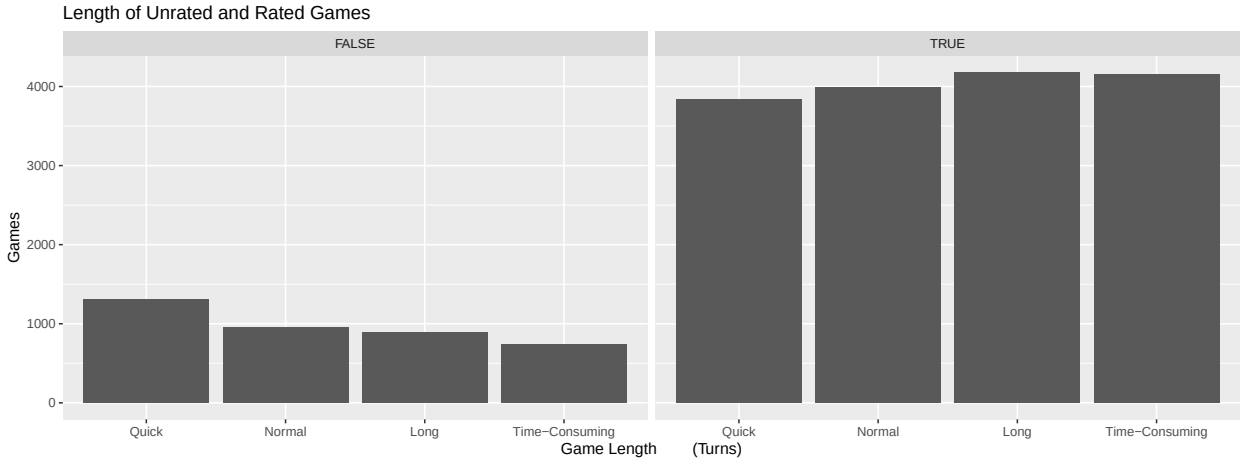
```r
black_advanced <- which(chess$black_rating_factor %in% ("Advanced"))
black_advanced_win <- which(chess$winner %in% ("black") & chess$black_rating_factor %in% ("Advanced"))
black_advanced_win_freq <- length(black_advanced_win) / length(black_advanced)
black_master <- which(chess$black_rating_factor %in% ("Master"))
black_master_win <- which(chess$winner %in% ("black") & chess$black_rating_factor %in% ("Master"))
black_master_win_freq <- length(black_master_win) / length(black_master)

# Vector to store the proportion difference at each level
win_freq_diff_rating <- c(white_beginner_win_freq - black_beginner_win_freq, white_intermediate_win_fre
names(win_freq_diff_rating) <- c("Beginner", "Intermediate", "Advanced", "Master")

barplot(win_freq_diff_rating, main = "Win Freq Difference vs Rating Level", xlab = "Rating", ylab = "Di
```



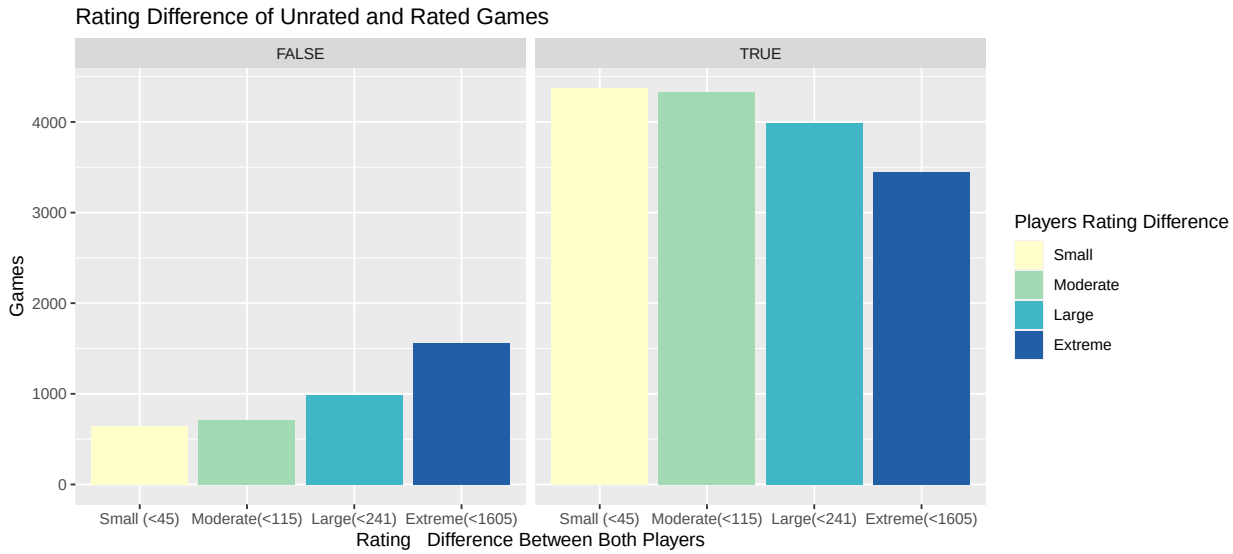

```r
# Barplot of expertise of rated and unrated
ggplot(chess) +
  geom_bar(mapping = aes(x = avg_rating_factor)) +
  facet_wrap(~rated) +
  labs(title = "Distribution of Avg Player Levels for Unrated and Rated     Games", x = "Avg Rating Betw
```



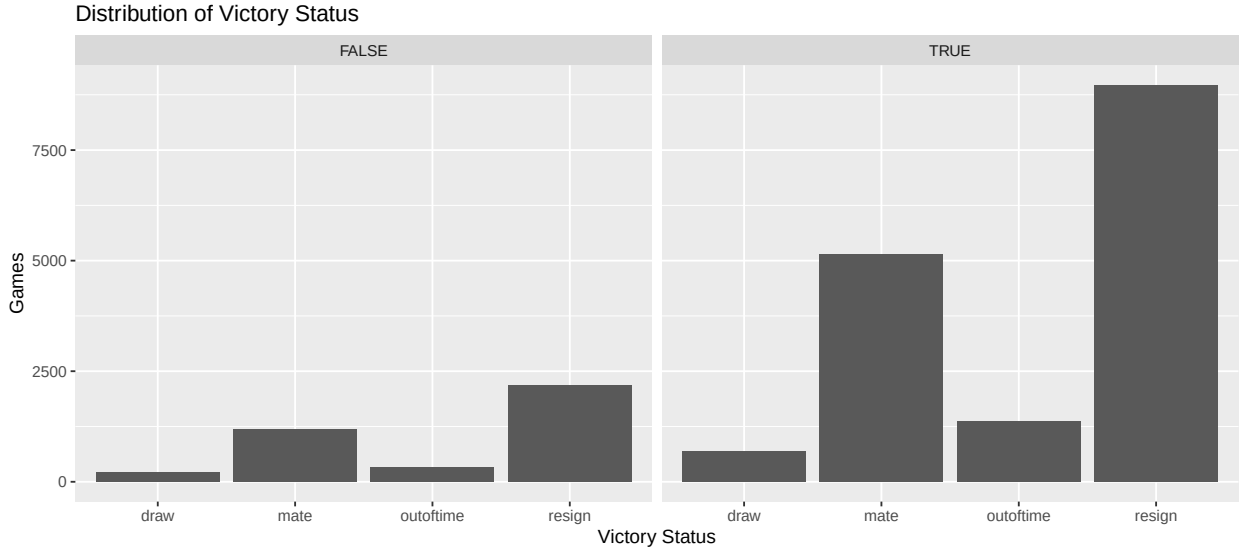

```
# Barplot of gamelengths of rated and unrated
ggplot(chess) +
  geom_bar(mapping = aes(x = gamelength)) +
  facet_wrap(~rated) +
  labs(title = "Length of Unrated and Rated Games", x = "Game Length        (Turns)", y = "Games")
```



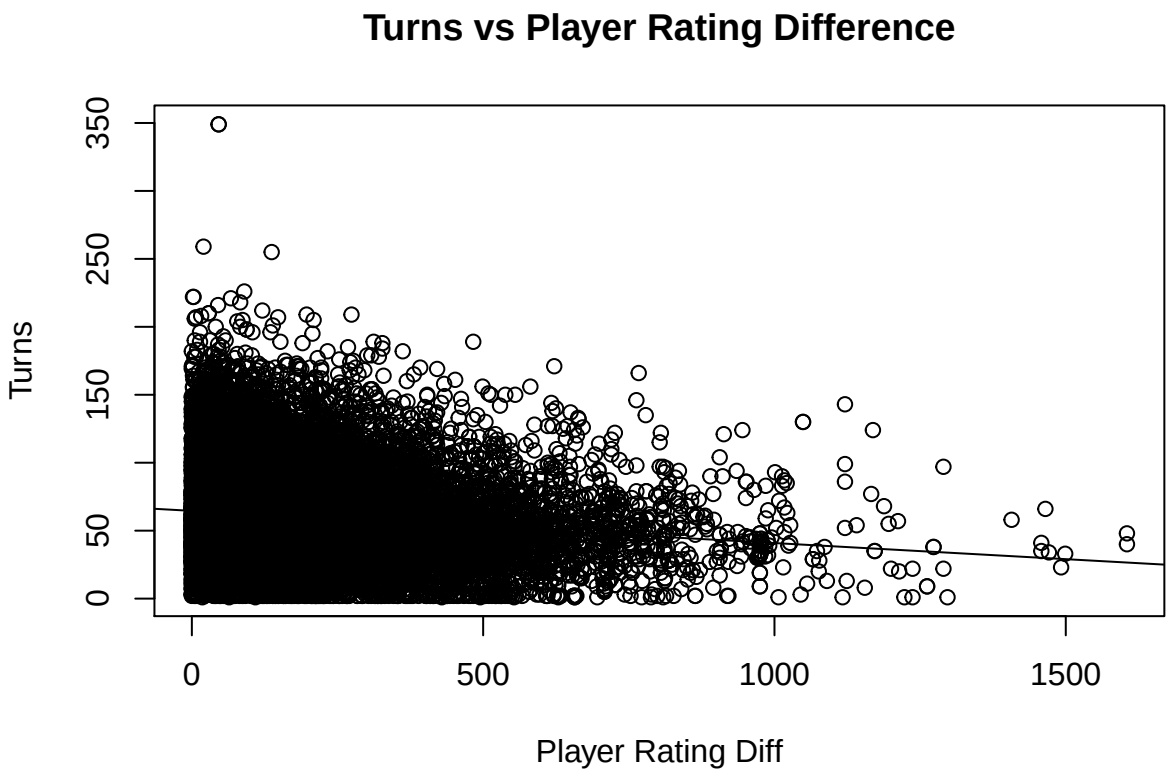

```
# Barplot of rating difference of rated and unrated
ggplot(chess) +
  geom_bar(mapping = aes(x = cut(diff_rating, c(-1, 45, 115, 241, 1605),
  labels = c("Small (<45)", "Moderate(<115)", "Large(<241)",                "Extreme(<1605)")), fill = c
  facet_wrap(~rated) +
  scale_fill_brewer(palette = "YlGnBu", type = c(5)) +
  labs(title = "Rating Difference of Unrated and Rated Games", x = "Rating   Difference Between Both Pla
```




```
# Barplot of victory status of rated and unrated
ggplot(chess) +
  geom_bar(mapping = aes(x = victory_status)) +
  facet_wrap(~rated) +
  scale_fill_brewer(palette = "YlGnBu", type = c(5)) +
  labs(title = "Distribution of Victory Status", x = "Victory Status", y =   "Games")
```




```
# Scatterplot of gamelength vs rating difference
# Find coefficients
plot(chess$diff_rating, chess$turns, main = "Turns vs Player Rating Difference", xlab = "Player Rating
m <- lm(chess$turns ~ chess$diff_rating, data = chess)
abline(m)
```




```
a = coef(m)[1]
a
```

```
## (Intercept)
##    64.56858
```

```
b = coef(m)[2] # Slope
b
```

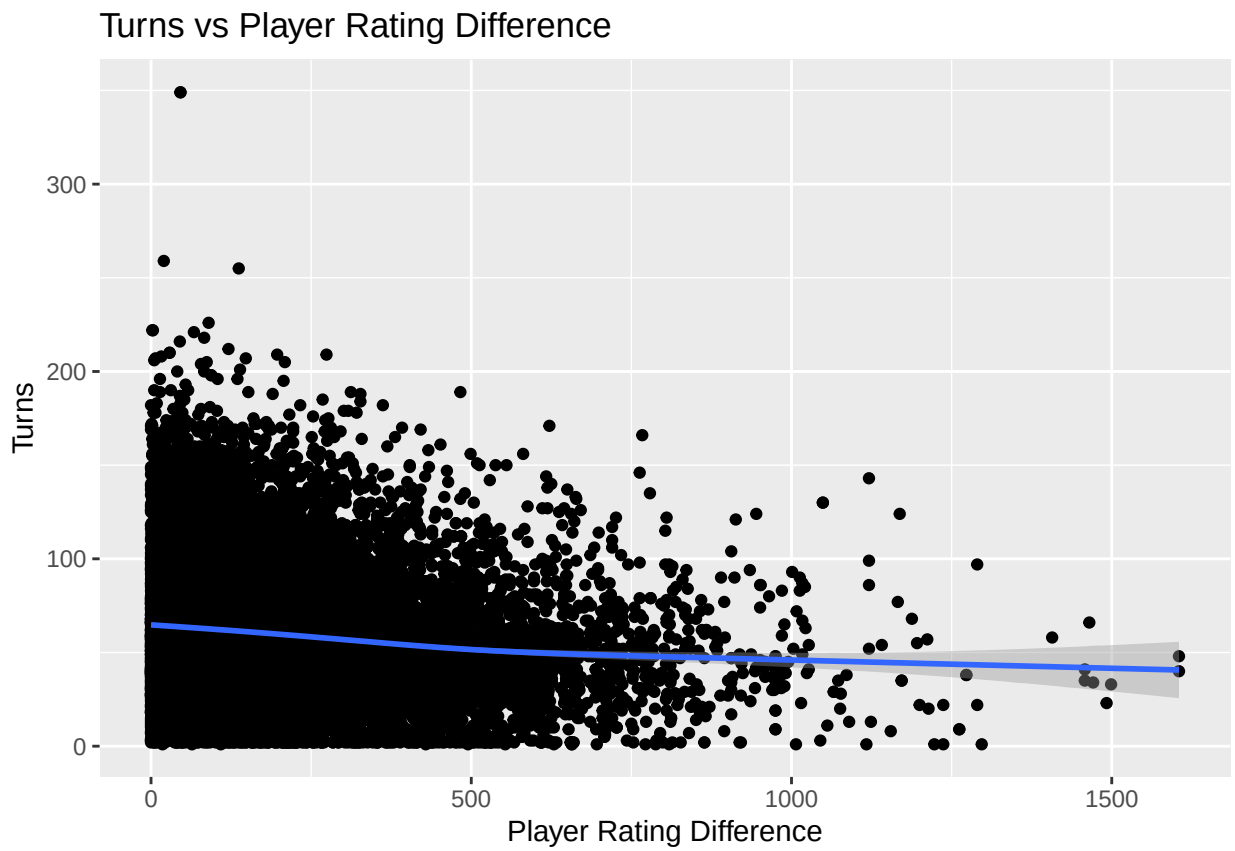
```
## chess$diff_rating
##        -0.0237018
```

```
cor(chess$diff_rating, chess$turns)
```

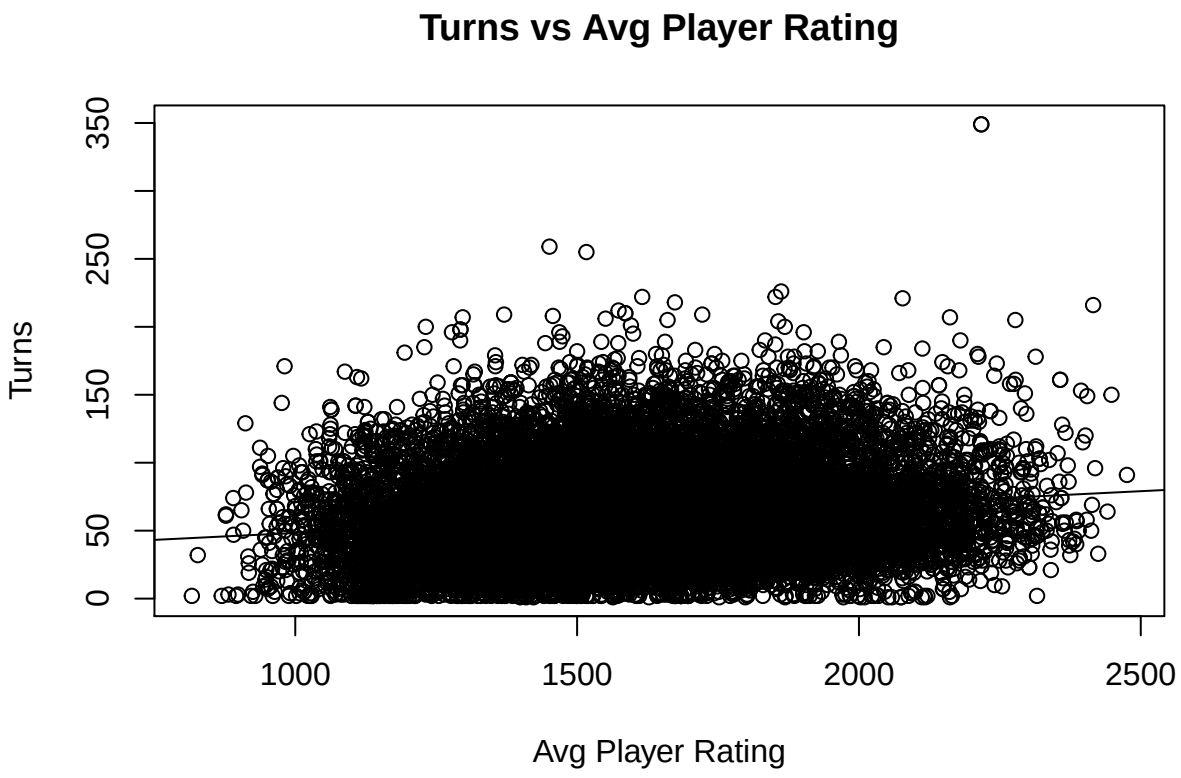
```
## [1] -0.1265309
```

```
# Actual plot that was used
ggplot(chess) +
geom_point(mapping = aes(x = diff_rating, y = turns)) + geom_smooth(mapping = aes(x = diff_rating, y = 
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
# Scatterplot of gamelength vs player expertise
# Find coefficients
plot(chess$avg_rating, chess$turns, main = "Turns vs Avg Player Rating", xlab = "Avg Player Rating", yl
m <- lm(chess$turns ~ chess$avg_rating, data = chess)
abline(m)
```




```
a = coef(m)[1]
a
```

```
## (Intercept)
##    27.85196
```

```
b = coef(m)[2] # Slope
b
```

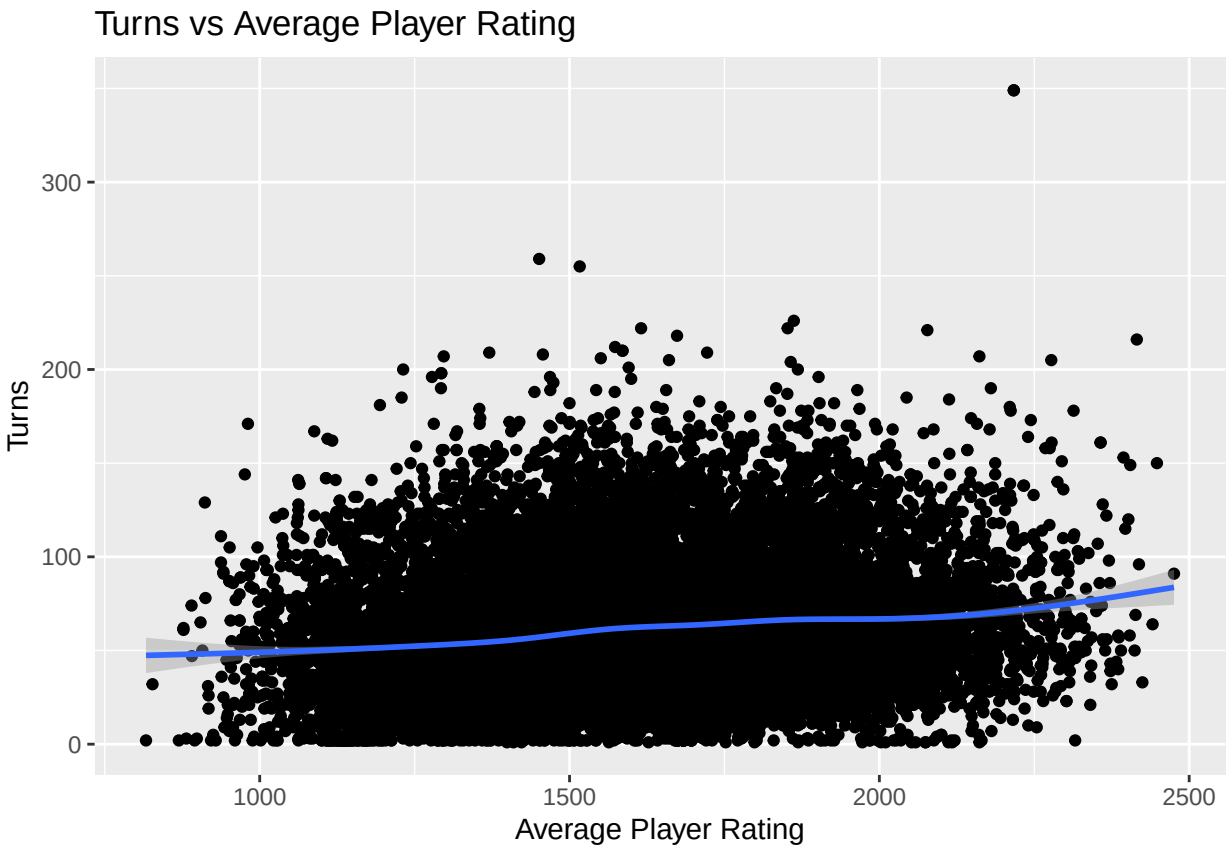
```
## chess$avg_rating
##       0.02047679
```

```
cor(chess$avg_rating, chess$turns)
```

```
## [1] 0.1605261
```

```
# Actual plot that was used
ggplot(chess) +
geom_point(mapping = aes(x = avg_rating, y = turns)) + geom_smooth(mapping = aes(x = avg_rating, y = tu
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


Turns vs Average Player Rating
Turns
300
200
100
0
1000
1500
2000
2500
Average Player Rating