Poker

Bas Groeneveld

October 9, 2016

Our data set contains information about Texas Holdem poker, from a Pokerstars account (self). With 884 games, there is plenty of data to analyze. There are a few things we'd like to know up front:

- How much Net Profit we're making (on average), per game type / speed / poker tag, and what's our return on investment **[Chunk 4: Plot 1]**
- Whether date / time has a significant influence on our results **[Chunk 4: Plot 2,3]**
- How closely related the net winnings and the expected value are (this is basically our 'luck factor') **[Chunk 4: Plot 4]**
- At what spots we usually finish - this can be a good indication to determine whether we 'bubble' too often, or not often enough **[Chunk 4: Plot 5]**
- How much rake we pay on average, and if rake is consistant with total buyins **[Chunk 4: Plot 6]**
- How much hourly profit we make per hand, and if there are serious problem hands we need to improve on. **[Chunk 4: Plot 7]**
- Dig deeper into correlation between variables **[Chunk 5: Corrplot 1]**
- Lastly, we look at some more complicated in-game stats **[Chunk 6]**

Let's start by making our data suitable for analysis.

```
# Numerics & integers
poker$Total.Buy.In <- as.numeric(sub("\\$", "", poker$Total.Buy.In))
poker$Rake <- as.numeric(sub("\\$", "", poker$Rake))
poker$Rebuy <- as.numeric(sub("\\$", "", poker$Rebuy))
poker$Winnings <- as.numeric(sub("\\$", "", poker$Winnings))
poker$Net.Winnings...USD <- as.numeric(sub("\\$", "", poker$Net.Winnings...USD))
poker$X.EV <- as.numeric(sub("\\$", "", poker$X.EV))

# Convert tourney time to minutes
poker$Tourney.Time.Played <- as.numeric(sub("m$", "", poker$Tourney.Time.Played))

# Factors
poker$Speed <- as.factor(poker$Speed)
poker$Table.size <- as.factor(poker$Table.size)
poker$Tag <- as.factor(poker$Tag)

# Date & time
glct <- Sys.getlocale("LC_TIME")
poker$Start.Time <- as.Date(poker$Start.Time, "%m/%d")

# Data frame
poker <- as.data.frame(poker)

# Targeted subsetting & filtering
poker <- subset.data.frame(poker, poker$Tag != "Spin and Go")
poker <- subset.data.frame(poker, poker$Tag != "18-Person")
```

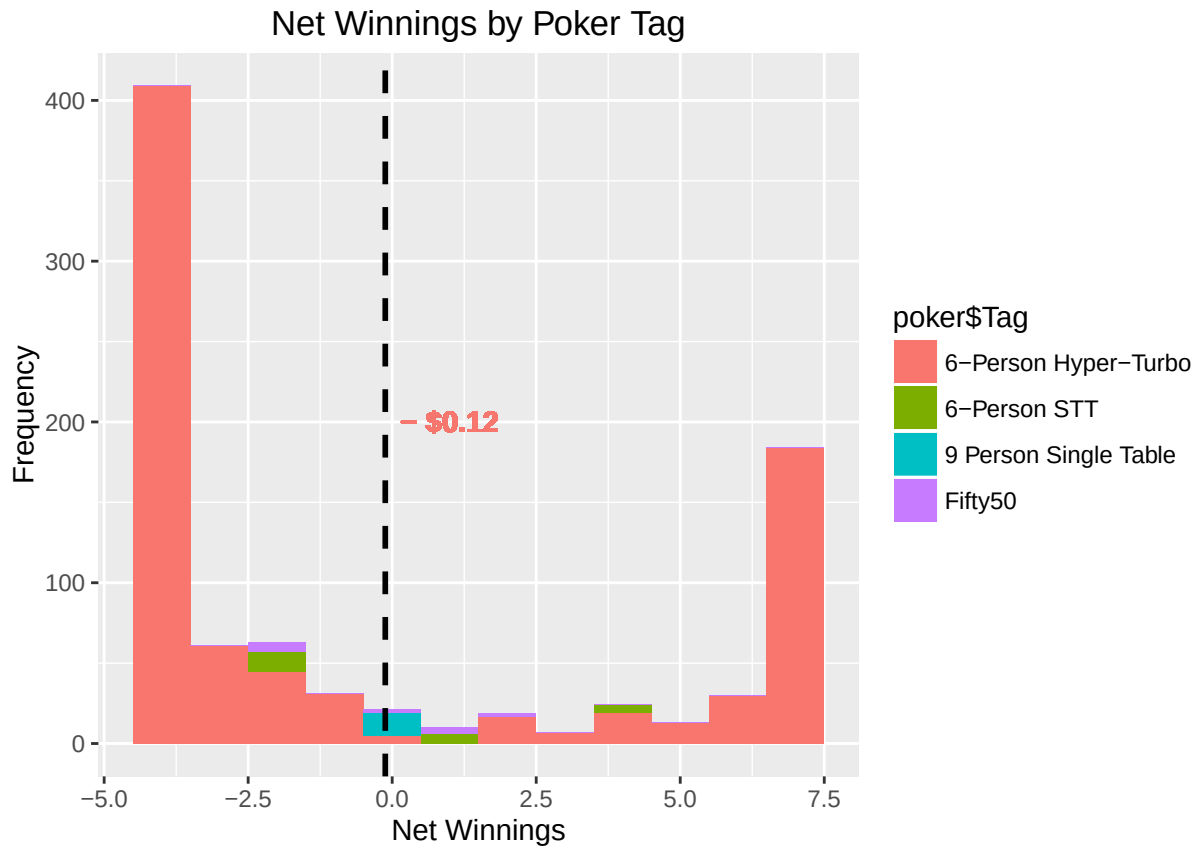
```
poker <- poker[-5]
poker <- na.omit(poker)
```

There is a lot to analyze in the world of poker. Let's take a look at both hands and additional computed values.

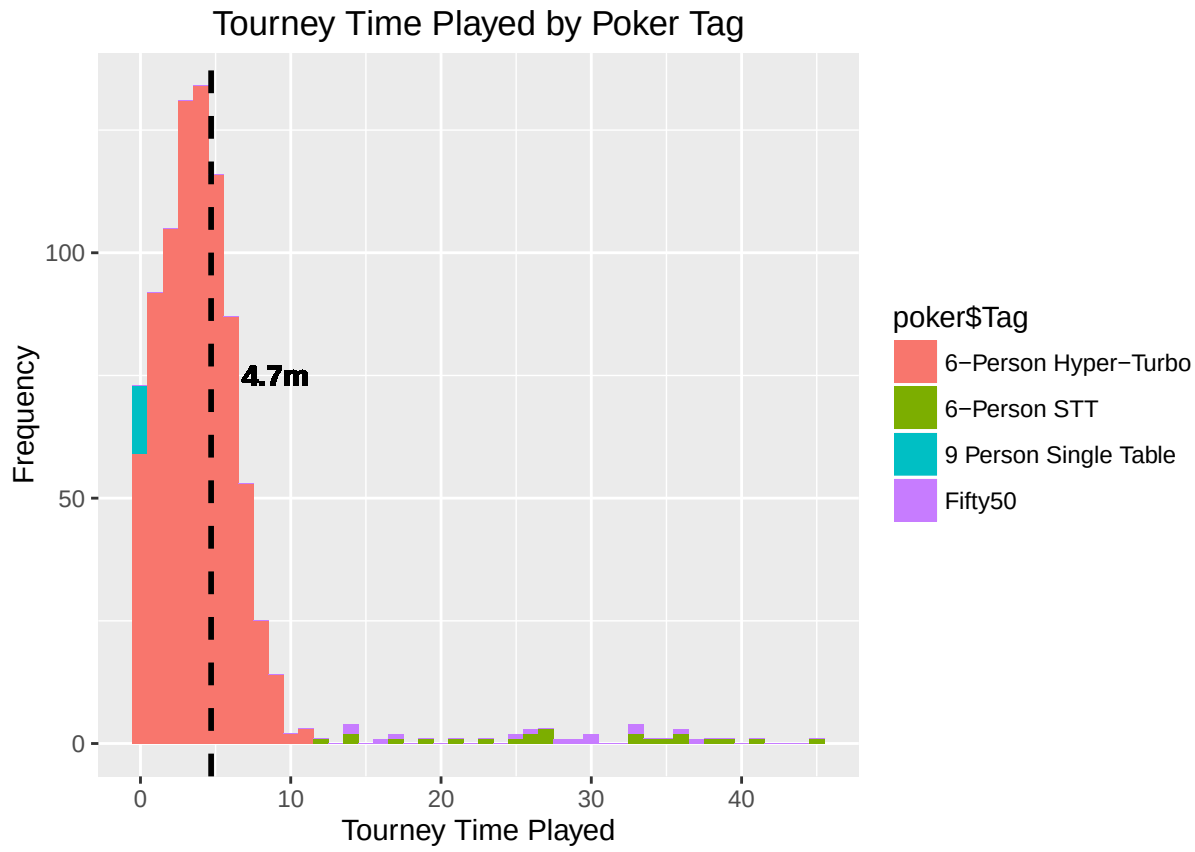
```
# Adding columns
poker$ProfitPerHour <- poker$Net.Winnings...USD/(poker$Tourney.Time.Played/60)
poker$EquityDiff <- poker$Net.Winnings...USD - poker$X.EV

# Creating hand combinations with random sampling of suits
# and ranks

Suits <- as.list(c("Clubs", "Diamonds", "Hearts", "Spades"))
Ranks <- as.list(c(2:10, "Jack", "King", "Queen", "Ace"))

poker$Suits <- sample(Suits, 872, replace = TRUE)
poker$Ranks <- sample(Ranks, 872, replace = TRUE)
poker$Merged <- paste(poker$Ranks, "of", poker$Suits)
poker$Merged <- as.factor(poker$Merged)

# Making a seperate data frame for each suit

pokerClubs <- subset(poker, grepl("Clubs", poker$Merged))
pokerClubs <- as.data.frame(pokerClubs)

pokerDiamonds <- subset(poker, grepl("Diamonds", poker$Merged))
pokerDiamonds <- as.data.frame(pokerDiamonds)

pokerHearts <- subset(poker, grepl("Hearts", poker$Merged))
pokerHearts <- as.data.frame(pokerHearts)

pokerSpades <- subset(poker, grepl("Spades", poker$Merged))
pokerSpades <- as.data.frame(pokerSpades)
```

Now, we're ready to start plotting - both qplots and basic R plots

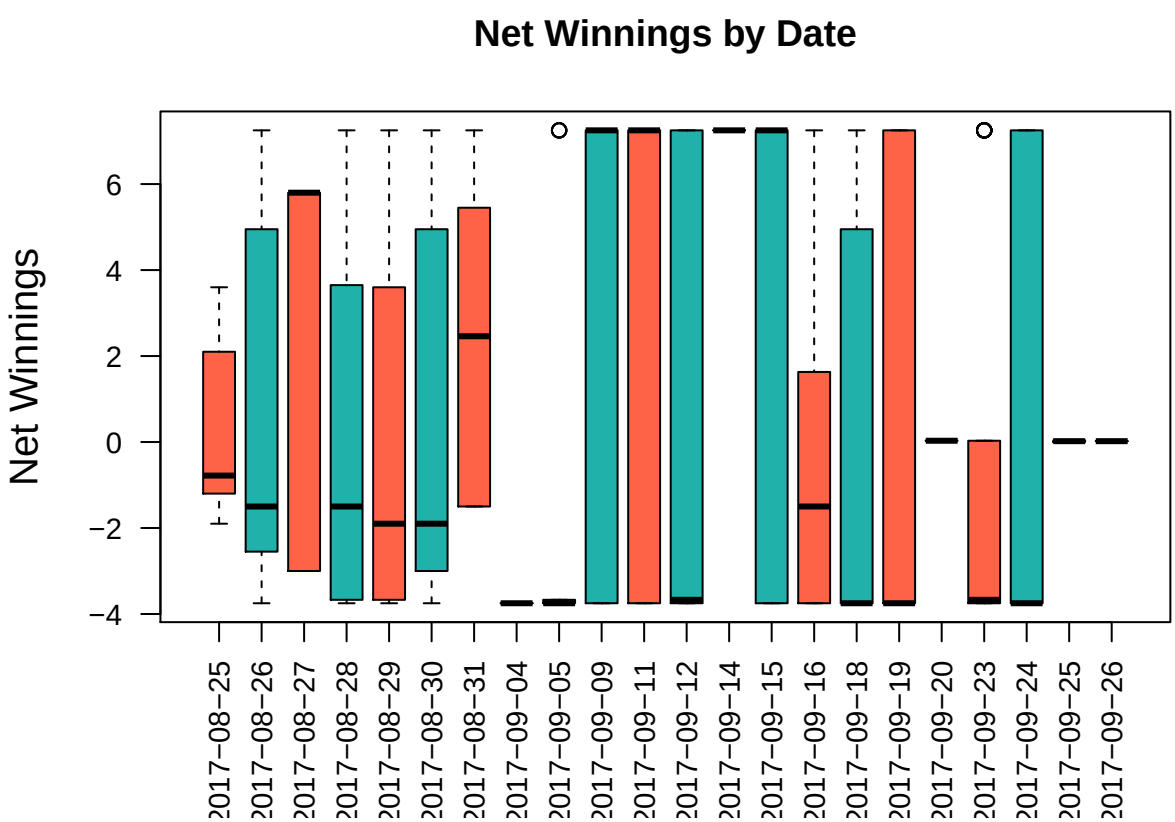
```
# Plot1: We'd like to see what Net Winnings we're getting for
# each Poker Tag

plot1 <- qplot(poker$Net.Winnings...USD, data = poker, geom = "histogram",
    fill = poker$Tag, binwidth = 1, xlab = "Net Winnings", ylab = "Frequency") +
    labs(title = "Net Winnings by Poker Tag") + geom_vline(xintercept = (mean(poker$Net.Winnings...USD)),
    colour = "black", linetype = "dashed", size = 1) + geom_text(aes(1,
    200, label = "- $0.12", fontface = "bold", colour = "red")) +
    guides(colour = FALSE)
plot1
```



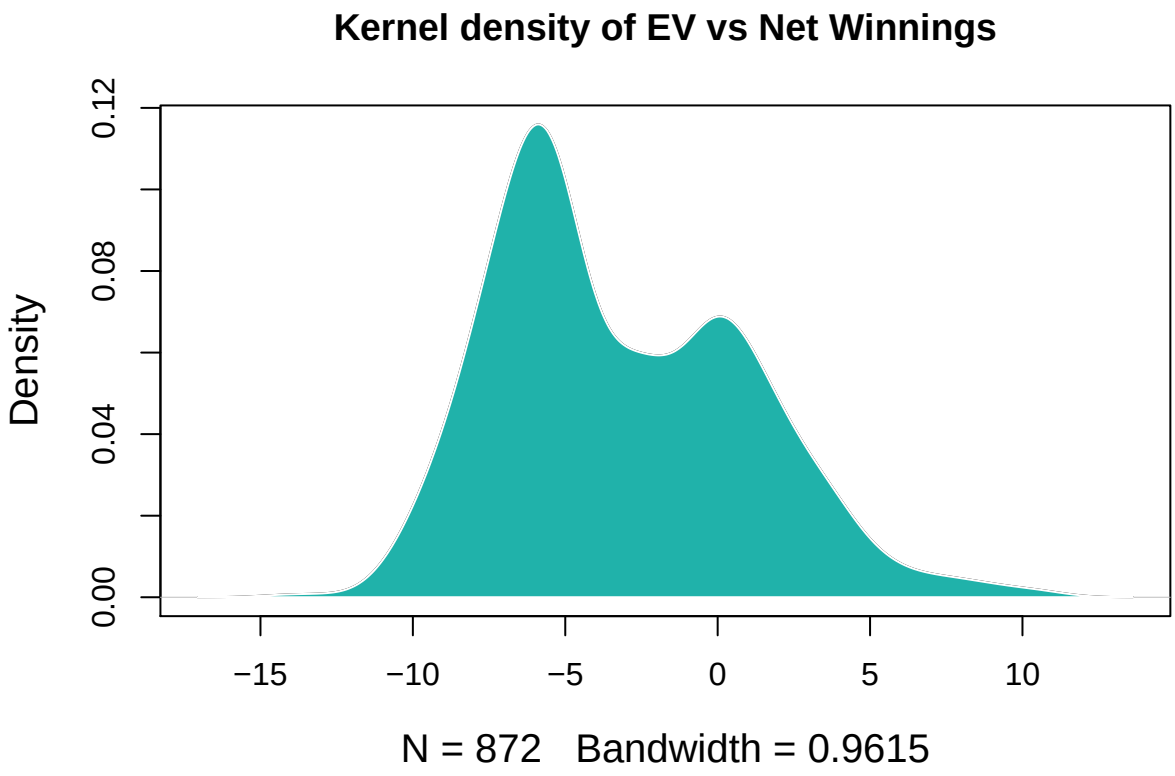

```
# Subsequently, let's get the RoI while we're at it

PokerBuyInSum <- sum(poker$Total.Buy.In)
PokerNetSum <- sum(poker$Net.Winnings...USD)
PokerRoI <- (PokerNetSum/PokerBuyInSum) * 100

# Plot2: We'd like to see if time is a factor that influences
# the result of each Poker Tag.

plot2 <- qplot(poker$Tourney.Time.Played, data = poker, geom = "histogram",
    fill = poker$Tag, binwidth = 1, xlab = "Tourney Time Played",
    ylab = "Frequency") + labs(title = "Tourney Time Played by Poker Tag") +
    geom_vline(xintercept = (mean(poker$Tourney.Time.Played)),
        colour = "black", linetype = "dashed", size = 1) + geom_text(aes(9,
    75, label = "4.7m", fontface = "bold", colour2 = "black")) +
    guides(colour = FALSE)
plot2
```




```
# Additional calculations

PokerHours <- sum(poker$Tourney.Time.Played/60)
PokerHourlyMeanLoss <- mean(poker$Net.Winnings...USD) * 872/60

# Plot3: We created a boxplot to see on what days we were
# more succesful and how big the differences were during the
# day (or 'swings' in poker)

plot3 <- boxplot(poker$Net.Winnings...USD ~ poker$Start.Time,
    data = poker, main = "Net Winnings by Date", ylab = "Net Winnings",
    col = (c("tomato1", "lightseagreen")), boxwex = 0.75, las = 2,
    cex.axis = 0.875, cex.lab = 1.25)
```

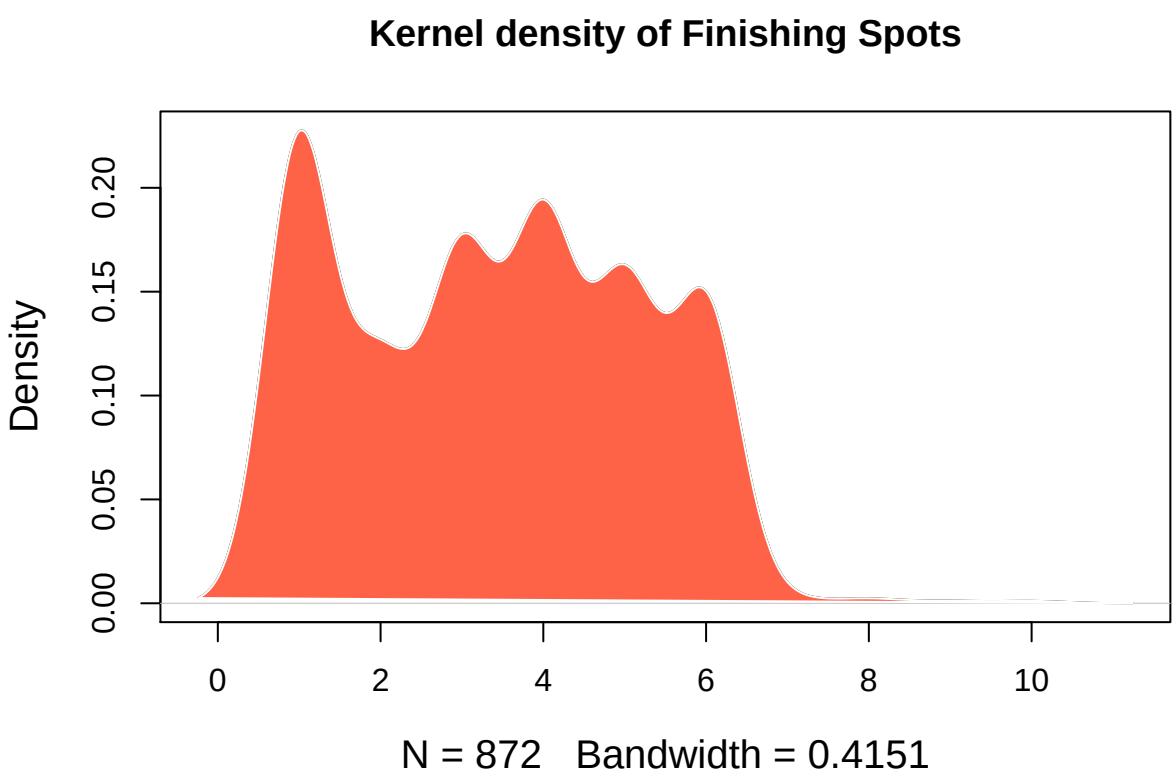



```
# Plot4: We created a density plot in which kernel density
# estimates are displayed for EV vs Net Winnings.

plot4 <- density(poker$EquityDiff)
plot(plot4, main = "Kernel density of EV vs Net Winnings", ylab = "Density",
    cex.lab = 1.25)
polygon(plot4, col = "lightseagreen", border = "white")
```



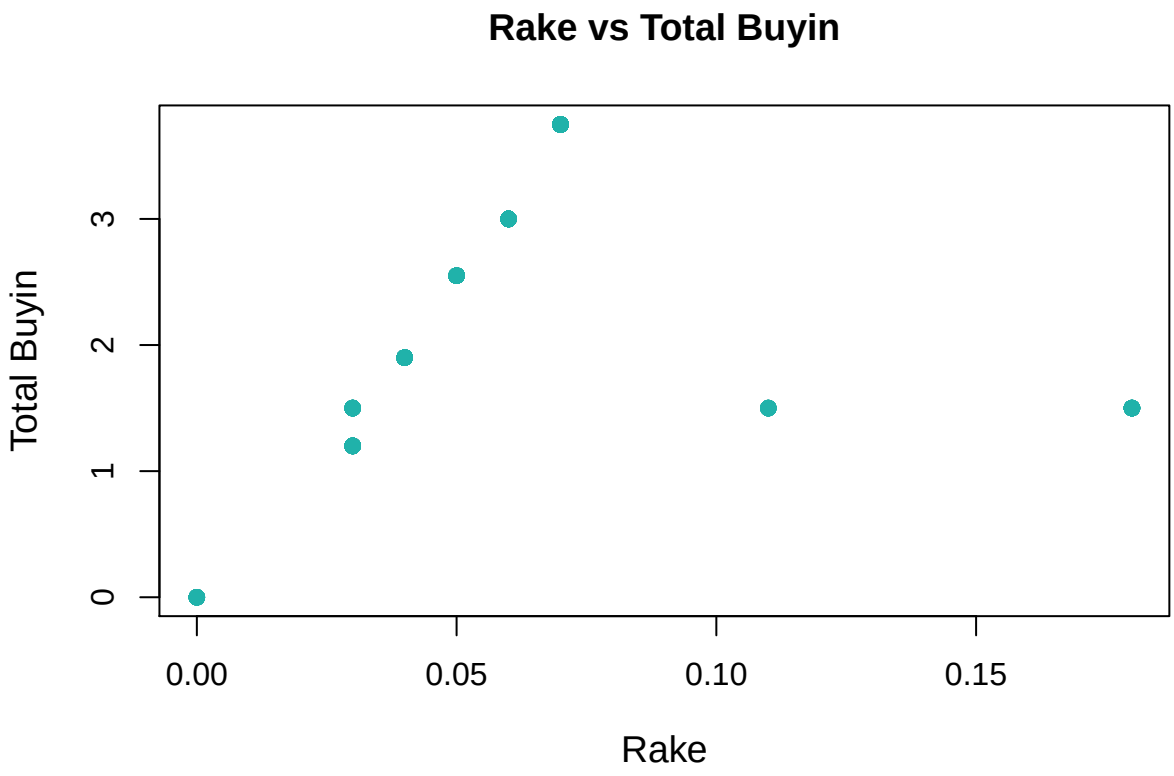
The model clearly shows that EquityDiff has a higher density for negative values, which translates into a higher expected value than our actual Net Winnings.

```
# Plot5: We created a density plot in which kernel density
# estimates are displayed for finishing spots.

plot5 <- density(poker$Finish)
plot(plot5, main = "Kernel density of Finishing Spots", ylab = "Density",
    cex.lab = 1.25)
polygon(plot5, col = "tomato1", border = "white")
```



Our finishing spot is also good, but our tag indicates we're mostly playing 6-max games, and therefor this is to be expected.

```
# Plotting rake against total buyin

plot6 <- plot(poker$Rake, poker$Total.Buy.In, main = "Rake vs Total Buyin",
    xlab = "Rake", ylab = "Total Buyin", cex.lab = 1.2, pch = 19,
    col = "lightseagreen")
```

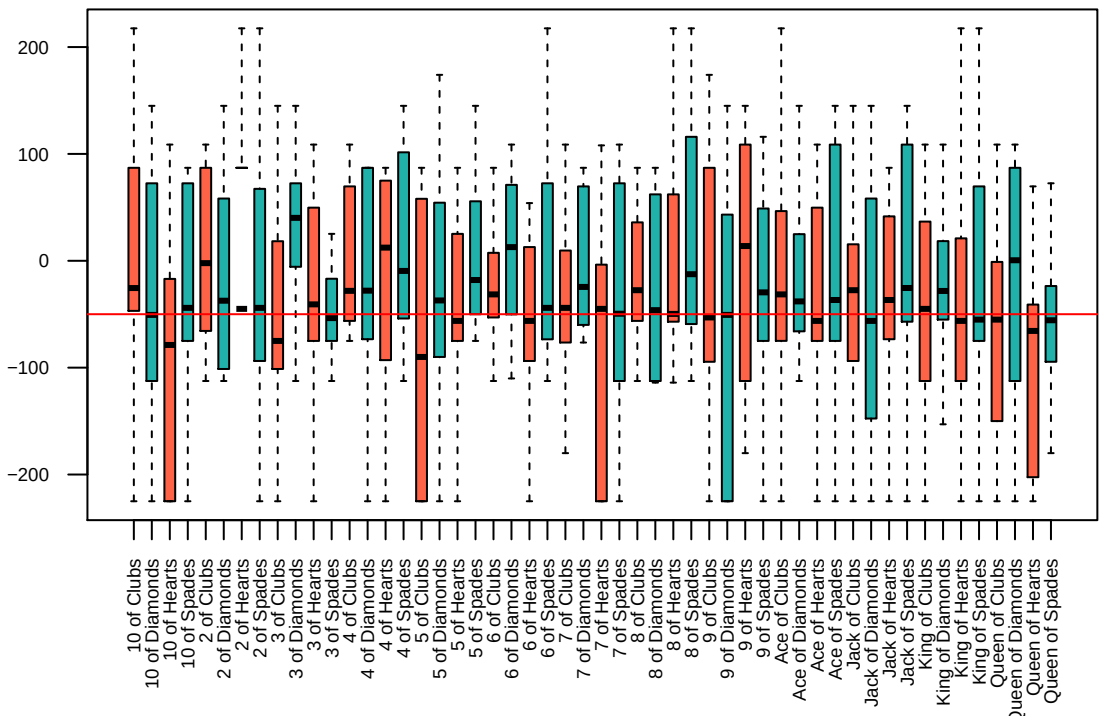



```
# Additional calculations

TotalRake <- sum(poker$Rake)
```

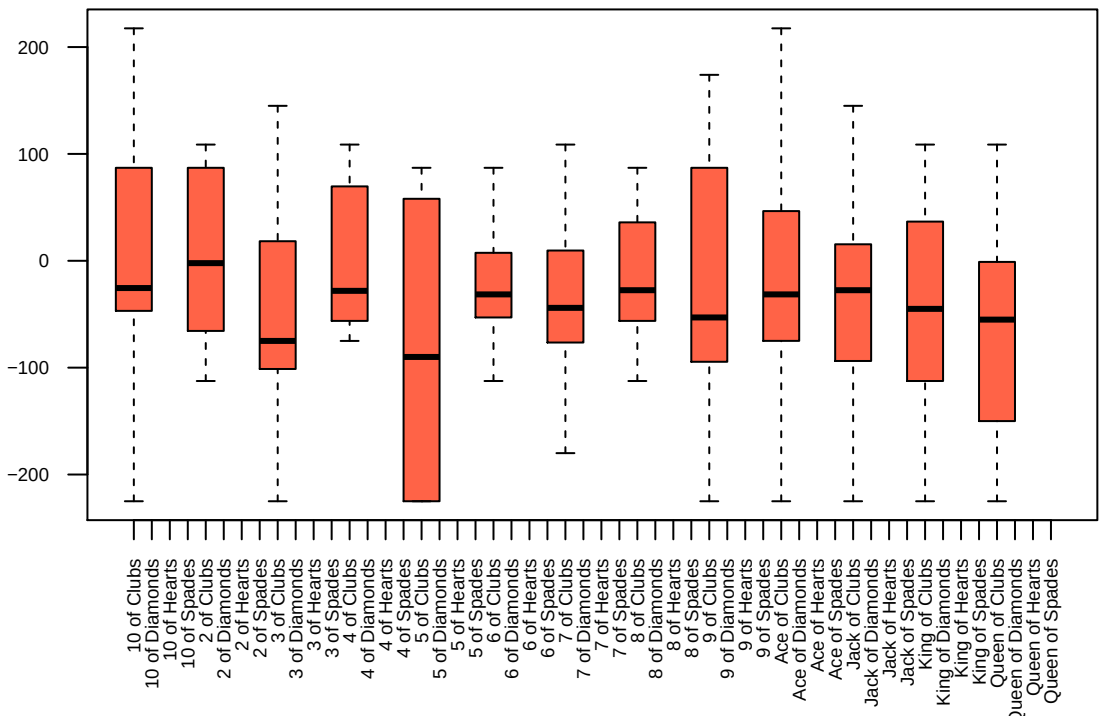
Rake didn't seem important at first, but with the TotalRake being $57.11, it makes up for a total of 54,5% percent of our total loss.

Plot7: Looking at all hands and the profit we make per hour overflows our plot window, but it should provide a good indication as to our true problem hands. Note that the hands were sampled and the results of this plot are therefor less valuable, but it's a nice experiment anyway

```
plot7 <- plot(poker$Merged, poker$ProfitPerHour, main = "Hourly profit per hand",
    col = (c("tomato1", "lightseagreen")), boxwex = 0.6, las = 2,
    cex.axis = 0.6, cex.lab = 1.25, outline = FALSE)
abline(h = -50, col = "red")
```




```
plot8 <- plot(pokerClubs$Merged, pokerClubs$ProfitPerHour, main = "Hourly profit per hand: Clubs",
    col = (c("tomato1", "lightseagreen")), boxwex = 2, las = 2,
    cex.axis = 0.6, cex.lab = 1.25, outline = FALSE)
```




```
# For some reason, the subplot on Suits doesn't work. The
# histograms are correct, but the x-axis labels still display
# Hearts, Spades and Clubs. Looking at the data frame
# 'PokerClubs', these values are non-present. I cannot tell
# why - when I outut plot8, the $names still list these
# values. Perhaps $Merged is taking values from Poker rather
# than pokerClubs? Anyhow, the idea is to at least reduce the
# $Merged values to be interpretable and more readable -
# ideally we output four different plots in one plot window,
# but we choose not to here due to our unexplainable problem.
```

Our problem hands are:

```
ProblemHands <- c("10 of Diamonds", "2 of Spades", "3 of Diamonds",
    "3 of Hearts", "5 of Clubs", "6 of Diamonds", "7 of Diamonds",
    "8 of Hearts", "9 of Hearts", "9 of Spades", "Ace of Clubs",
    "Ace of Spades", "King of Diamonds", "King of Hearts")

ProblemLines <- writeLines(ProblemHands)
```

```
## 10 of Diamonds
## 2 of Spades
## 3 of Diamonds
## 3 of Hearts
## 5 of Clubs
## 6 of Diamonds
## 7 of Diamonds
```

```
## 8 of Hearts
## 9 of Hearts
## 9 of Spades
## Ace of Clubs
## Ace of Spades
## King of Diamonds
## King of Hearts
```

We now have an outputted list of our problem hands, but no criteria other than a low mean Net Winnnings to draw conclusions on. We could do better by providing hand stats with them (i.e. Hands on the board, Position, Cbet succes, VPIP and PFR) but since we sampled hands, the result of that analysis would be too far off reality.

Let's also take a closer look at correlation,

To see what we should really be focusing on. The above plots may have answered some very important questions, but don't necessarily focus on everything we need for a complete analysis.

```
# Correlations we assume are relevant - this isn't really a
# good method, so we'll be looking at a different method
# aswell. We only select the initial values (excluding
# HourlyProfit, Hand-related variables and EquityDiff)

cor1 <- cor(poker$Winnings, poker$Net.Winnings...USD)
cor2 <- cor(poker$Winnings, poker$X.EV)
cor3 <- cor(poker$Net.Winnings...USD, poker$X.EV)
cor4 <- cor(poker$Size, poker$Tourney.Time.Played)

# Ordering

cordouble <- cbind(cor1, cor2, cor3, cor4)
rownames(cordouble) <- "Correlation"
ranks <- order(cordouble, decreasing = FALSE)
cordouble[, ranks]
```

```
##      cor4      cor3      cor2      cor1
## 0.3334195 0.4220069 0.4956214 0.9773668
```

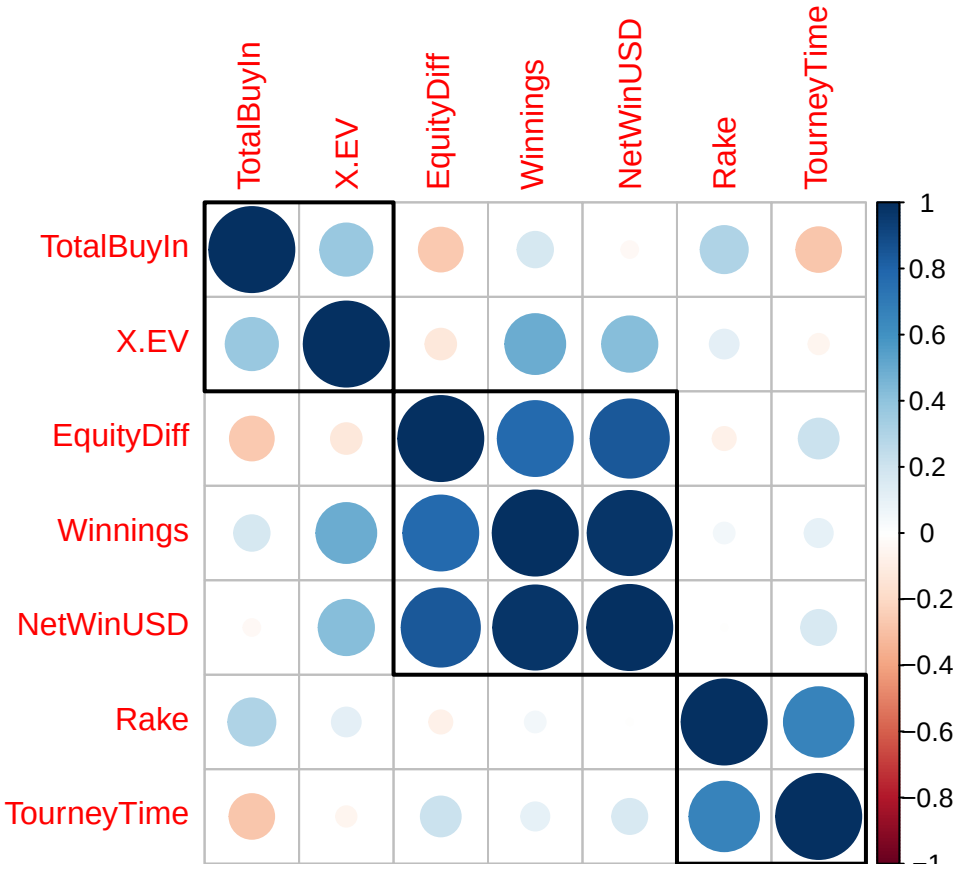
```
# Messages

writeLines("Best assumed correlation: Winnings vs Net winnings")
```

```
## Best assumed correlation: Winnings vs Net winnings
```

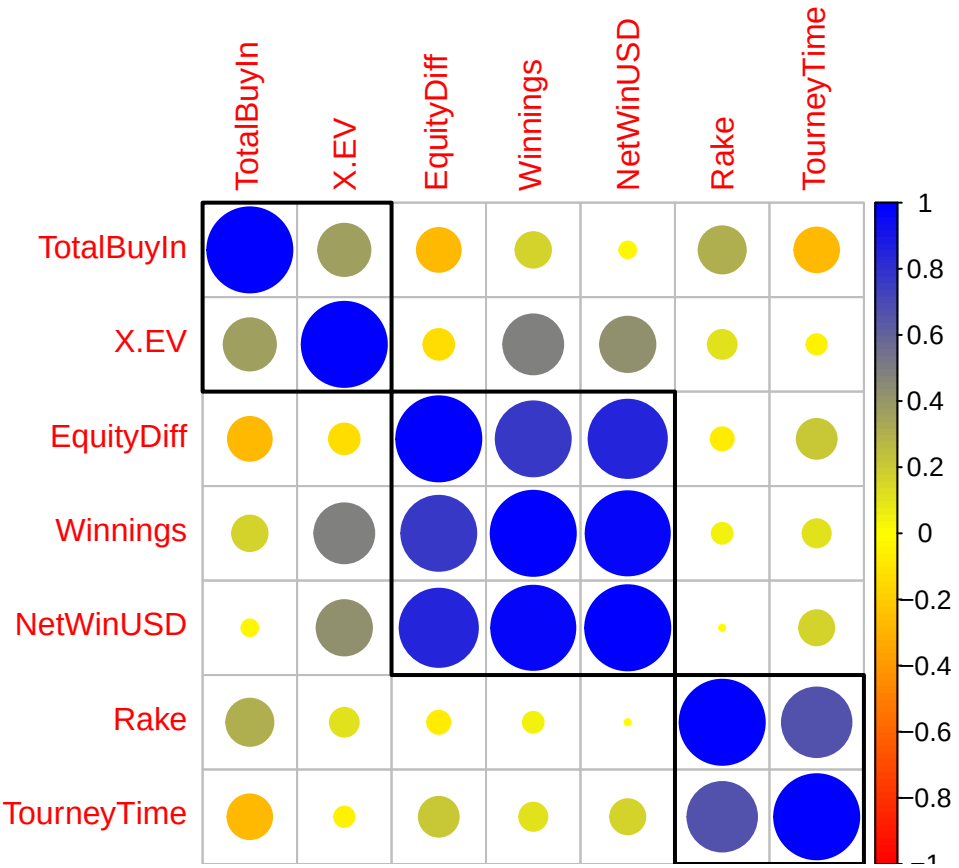
Let's look at a more automated method. Suggest we'd like to create a matrix in which each, or at least the most important correlations, are shown graphically, rather than in a vector-like setting. We installed the 'corrplot' package for this specific purpose. This time, we put every numeric value (except HourlyProfits, because it has many Inf values) in a data frame

```
# pokerStruc <- str(poker)

PokerNumerics <- data.frame(poker$Total.Buy.In, poker$Rake, poker$Winnings,
    poker$Net.Winnings...USD, poker$X.EV, poker$Tourney.Time.Played,
    poker$EquityDiff)
```

```r
# Changing the names for prettier labelling

names(PokerNumerics) <- c("TotalBuyIn", "Rake", "Winnings", "NetWinUSD",
    "X.EV", "TourneyTime", "EquityDiff")

# Correlation plot

Col1 <- colorRampPalette(colors = c("red", "yellow", "blue"))
Corrplot1 <- corrplot(cor(PokerNumerics), method = "circle",
    order = "hclust", addrect = 3)
```

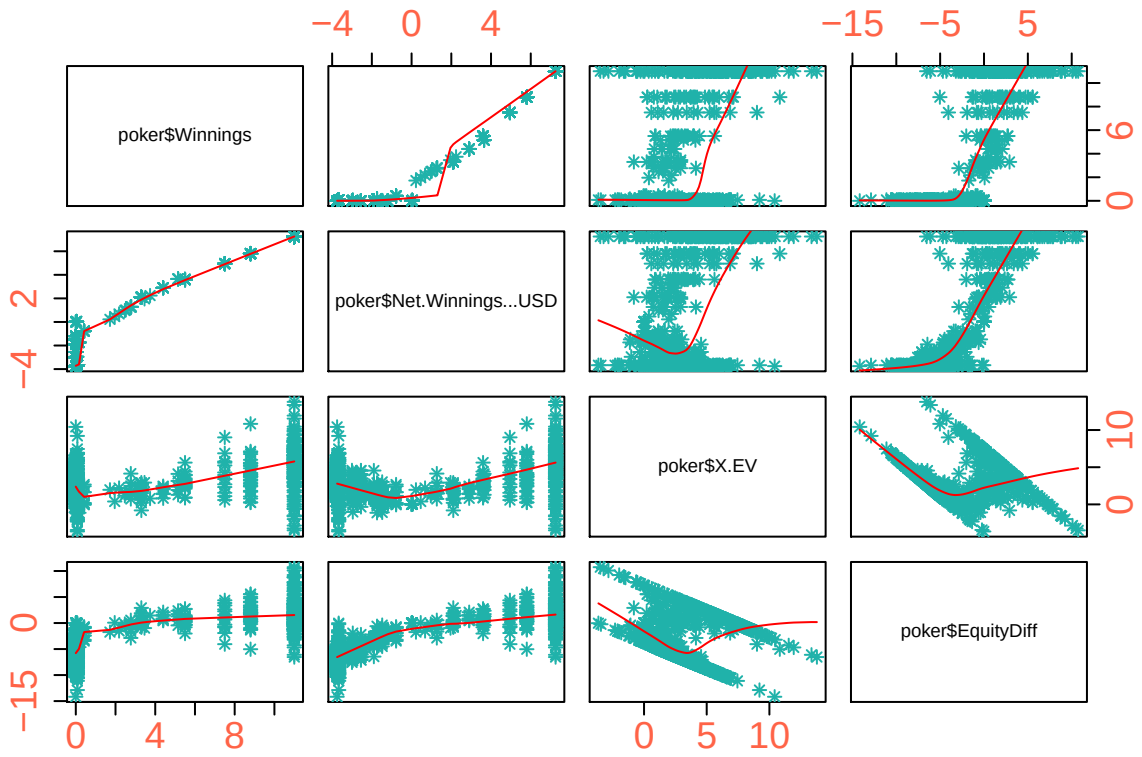



```r
# This one is prettier!

Corrplot1 <- corrplot(cor(PokerNumerics), order = "hclust", addrect = 3,
    col = Col1(100))
```



#Now that we have a better idea of the strongly correlated variables, we can 'select' more valuable scatterplots to visualize in a matrix. We can use pairs() for this.

```
Scatterplot1 <- pairs(~poker$Winnings + poker$Net.Winnings...USD +
    poker$X.EV + poker$EquityDiff, data = PokerNumerics, lower.panel = panel.smooth,
    upper.panel = panel.smooth, pch = 8, col = "lightseagreen",
    col.axis = "tomato1", cex.axis = 1.75)
```



The scatterplots aren't as straightforward as we might have imagined: they're more confusing than the plots before, and cluttering our thoughts. They do support the correlations, just shown in a different way. But do they provide actionable insights? No, unfortunately just statistical insights. #Let's finally take a look at another .csv-file containing more information on our in-game stats. As a result of our basic analysis, we concluded that providing hand stats would actually give **plot6** some meaning. We could have binded these stats to the poker table before, but we purposefully didn't because that would broaden the scope of our analysis a bit too much.

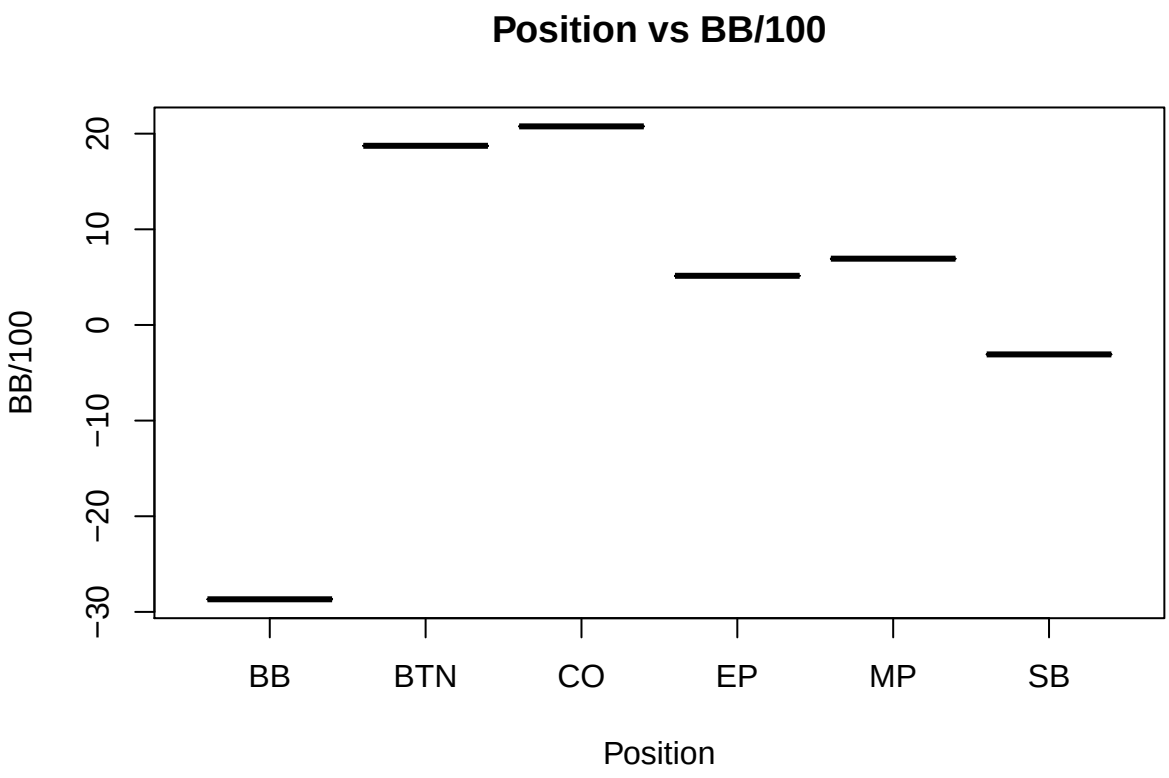
```
# sessionInfo()
perform <- read.csv("Reported.csv", stringsAsFactors = FALSE)
str(perform)
```

```
## 'data.frame':    6 obs. of  11 variables:
##  $ Position     : chr  "BTN" "CO" "MP" "EP" ...
##  $ Total.Hands  : int  3206 2142 1463 832 3405 3375
##  $ Net.Won.Chips: chr  "na" "na" "na" "na" ...
##  $ bb.100       : num  18.73 20.76 6.93 5.14 -28.68 ...
##  $ VPIP         : num  31.1 19.8 12.8 11.4 27.8 ...
##  $ PFR          : num  32 20.2 12.1 10.8 17.6 ...
##  $ X3Bet        : num  12.27 8.08 1.14 4.35 21.41 ...
##  $ WTSD.        : num  94.3 95.7 92.9 92.2 79.7 ...
##  $ W.SD.        : num  47.1 54.7 42.3 44.7 53.4 ...
##  $ Agg          : num  2.54 2.25 4.5 2 3.88 4.81
##  $ Agg.         : num  26.8 37.5 47.4 40 31.6 ...
```

It looks like our total hands got aggregated per position. This isn't necessarily bad (it justifies a big enough sample to be representative) but does question whether we can adequately draw conclusions from a 6x11

data frame. Net.Won.Chips. only contains 'NA' values, so let's get rid of that straight away.

```
perform <- perform[-3]
perform$Position <- as.factor(perform$Position)
plot8 <- plot(perform$Position, perform$bb.100, xlab = "Position",
    ylab = "BB/100", main = "Position vs BB/100")
```

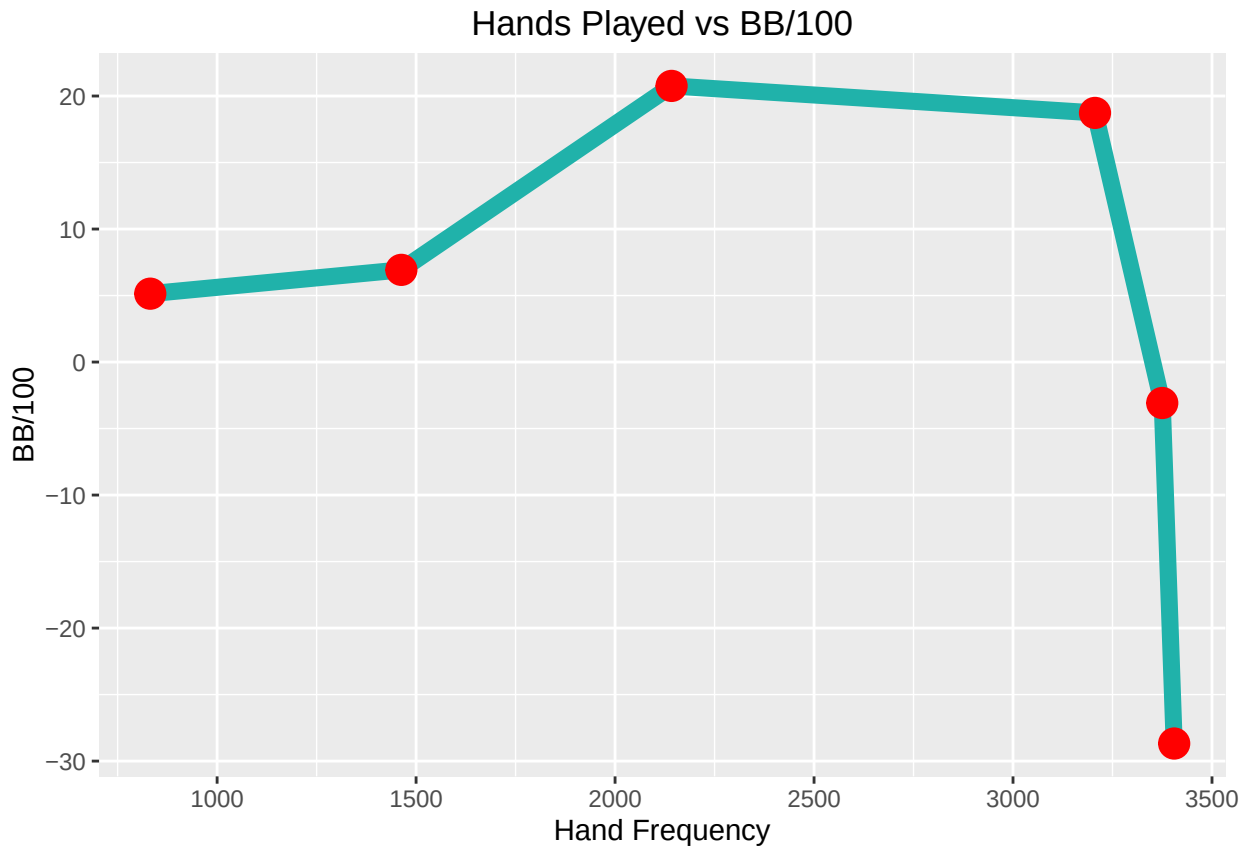


From a first basic plot, we can see that we're only making losses on the BB (=Big Blind). For further analysis, we could look at our problem hands on the big blind, but we'd have to bind the two data frames together and dig deeper. This is probably the most valuable result we've come across though.

Plot 3 and plot 7/8 were the graphs that dissapointed badly.

We tried to show our Net Winnings through dates, but the boxplot was a poor graphical representation. Let's give it another try by making a line plot which compares total hands and bb/100

```
plot9 <- ggplot(perform, aes(perform$Total.Hands, perform$bb.100,
    group = 1)) + geom_line(linetype = "solid", color = "lightseagreen",
    size = 3) + geom_point(color = "red", size = 5) + scale_x_continuous("Hand Frequency") +
    scale_y_continuous("BB/100") + ggtitle("Hands Played vs BB/100")
plot9
```



We can conclude this gives us a much better result. There is a clear downfall in BB/100 in the end. It's time to make a summary of our results.

We mostly play Hyper-Turbo poker, which can be seen in our average tourney time played of just 4.7 minutes. With 68 hours of poker played, our hourly average profit is -$1.74. Our net winnings are between -5 and 7.5, where the last plots shows a clear downfall in the end. However, our expected value is much higher than that, as seen in the kernel density plot. This makes it probable that we were simply 'out of luck'. Our finishing spot is also good, but our tag indicates we're mostly playing 6-max games, and therefor this is to be expected. Rake didn't seem important at first, but with the TotalRake being $57.11, it makes up for a total of 54,5% percent of our total loss.

The correlations along with the scatterplots are a nice addition, but don't provide that much extra information on what we already concluded in the previous plots. This is mostly a different graphical representation, and the reader can decide the value of this.

As to our hands, we have defined a set of problem hands to investigate on the big blind. We combined the information from two different data sets for this.

All in all, the conclusion is that we need to review our hands on the big blind in Hyper-Turbo, to determine whether there can be improvements through in-game stats and if we should keep playing at all or simply try a different variant.