STAC58 Group Project

Rebecca Han

2025-03-14

- pre-flop
- flop
- turn
- river

This data was collected from the University of Alberta's Computer Poker Research Group: https://poker.cs.ualberta.ca/irc__poker__database.html.

Interpreations of the column names came from https://github.com/allenfrostline/PokerHandsDataset/blob/master/src/extract.py

"Winning" in this first data analysis section just means that player hand is stronger than opponent hand in a 4-player game. - odds of winning given pre-flop - pairs, suited non-pairs, and off-suited non-pairs are the buckets - however, for my analysis, I included suited as well since it means a higher chance of getting a flush later on - so we have Off-Suited Pair / Suited Non-Pair / Suited Pair / Off-Suited Non-Pair - odds of improving hand at flop / winning - odds of improving hand at turn / winning - odds of improving hand at river / winning describe how we can bucket the data, leads into next part.

Pre-Flop

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

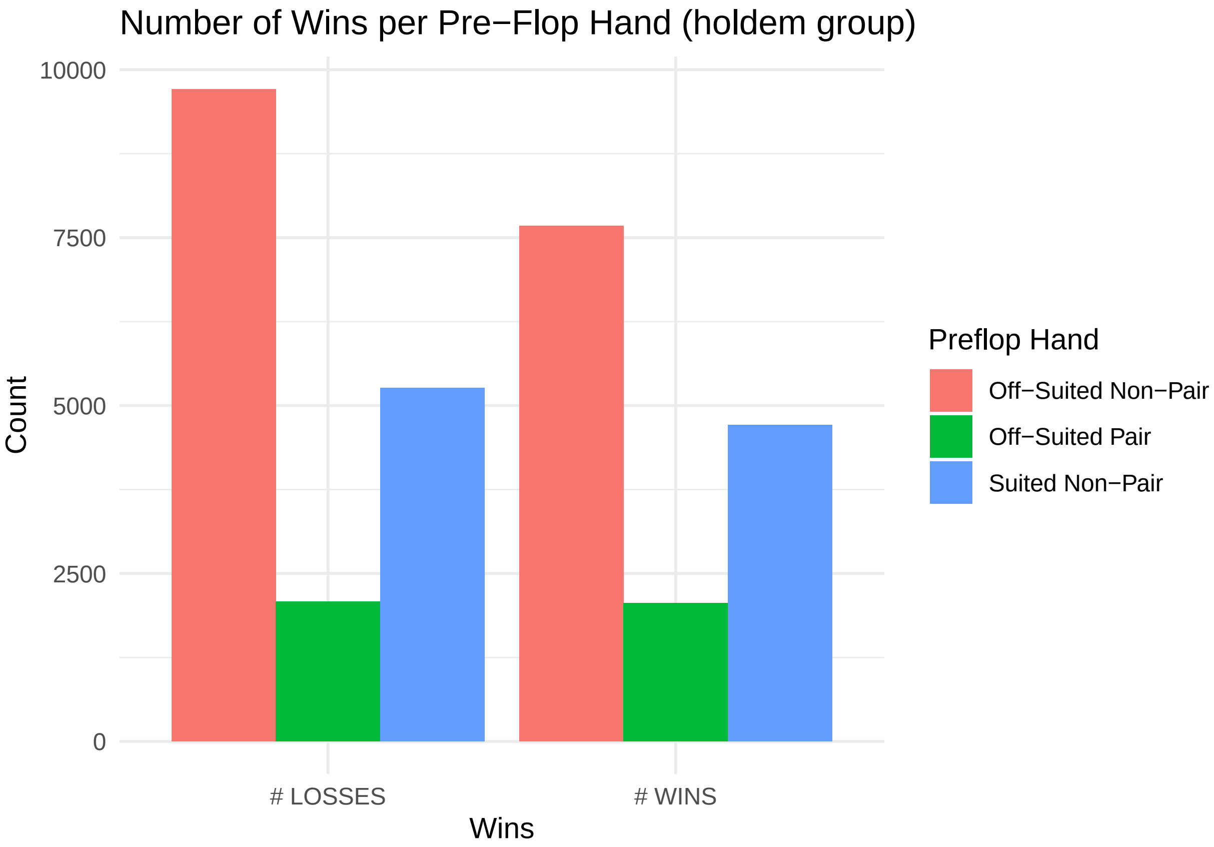
```r
library(ggplot2)

# Get list of files
filenames <- list.files("data/holdem")

# Create full file paths
full_paths <- paste("data/holdem", filenames, sep="/")

# Read files with read.table()
all_data <- lapply(full_paths, function(x) {
  read.table(x, header = FALSE, fill = TRUE, stringsAsFactors = FALSE)
})

# Find the maximum number of columns across all files
```

```r
max_cols <- max(sapply(all_data, ncol))

# Convert all columns to character type to ensure consistency
all_data_char <- lapply(all_data, function(df) {
  # Convert all columns to character
  for(i in 1:ncol(df)) {
    df[,i] <- as.character(df[,i])
  }
  return(df)
})

# Combine all dataframes
games1 <- bind_rows(all_data_char)

games_with_preflop_1 <- games1 %>%
  filter(V8 != "-") %>%
  filter(!is.na(V12), !is.null(V12), V12 != "") %>%
  mutate(rank12 = substr(V12, 1, 1)) %>%
  mutate(rank13 = substr(V13, 1, 1)) %>%
  mutate(suit12 = substr(V12, 2, 2)) %>%
  mutate(suit13 = substr(V13, 2, 2)) %>%
  mutate(preflop = case_when(
    rank12 == rank13 & suit12 != suit13 ~ "Off-Suited Pair",
    rank12 != rank13 & suit12 == suit13 ~ "Suited Non-Pair",
    rank12 != rank13 & suit12 != suit13 ~ "Off-Suited Non-Pair"
  )) %>%
  mutate(wins = case_when(
    V11 > 0 ~ "# WINS",
    TRUE ~ "# LOSSES"
  )) %>%
  select(wins, preflop)
```

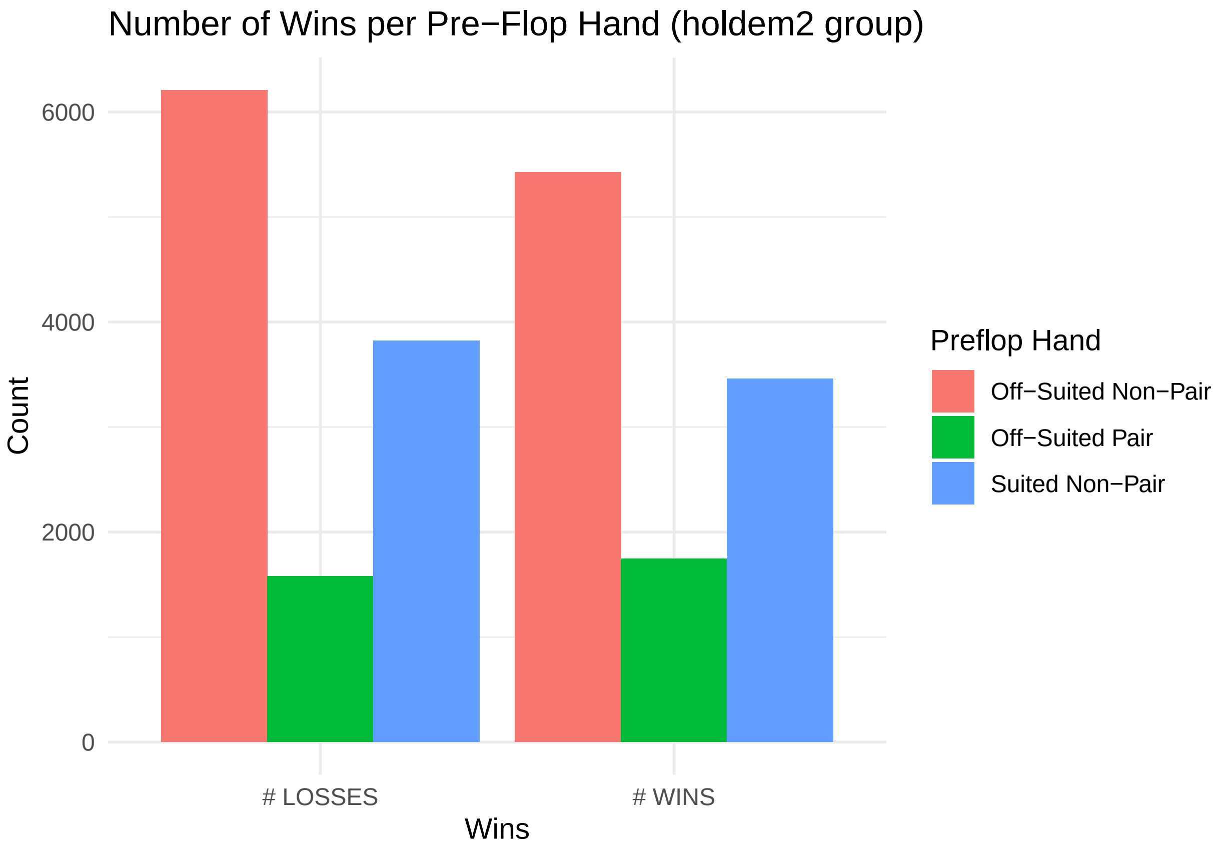
```r
# Get list of files
filenames <- list.files("data/holdem2")

# Create full file paths
full_paths <- paste("data/holdem2", filenames, sep="/")

# Read files with read.table()
all_data <- lapply(full_paths, function(x) {
  read.table(x, header = FALSE, fill = TRUE, stringsAsFactors = FALSE)
})

# Find the maximum number of columns across all files
max_cols <- max(sapply(all_data, ncol))

# Convert all columns to character type to ensure consistency
all_data_char <- lapply(all_data, function(df) {
  # Convert all columns to character
  for(i in 1:ncol(df)) {
    df[,i] <- as.character(df[,i])
  }
  return(df)
})

# Combine all dataframes
games2 <- bind_rows(all_data_char)

games_with_preflop_2 <- games2 %>%
  filter(V8 != "-") %>%
  filter(!is.na(V12), !is.null(V12), V12 != "") %>%
  mutate(rank12 = substr(V12, 1, 1)) %>%
  mutate(rank13 = substr(V13, 1, 1)) %>%
  mutate(suit12 = substr(V12, 2, 2)) %>%
  mutate(suit13 = substr(V13, 2, 2)) %>%
  mutate(preflop = case_when(
    rank12 == rank13 & suit12 != suit13 ~ "Off-Suited Pair",
    rank12 != rank13 & suit12 == suit13 ~ "Suited Non-Pair",
    rank12 != rank13 & suit12 != suit13 ~ "Off-Suited Non-Pair"
  )) %>%
  mutate(wins = case_when(
    V11 > 0 ~ "# WINS",
    TRUE ~ "# LOSSES"
  )) %>%
  select(wins, preflop)
```

```r
# Get list of files
filenames <- list.files("data/holdem3")

# Create full file paths
full_paths <- paste("data/holdem3", filenames, sep="/")

# Read files with read.table()
all_data <- lapply(full_paths, function(x) {
  read.table(x, header = FALSE, fill = TRUE, stringsAsFactors = FALSE)
})

# Find the maximum number of columns across all files
max_cols <- max(sapply(all_data, ncol))

# Convert all columns to character type to ensure consistency
all_data_char <- lapply(all_data, function(df) {
  # Convert all columns to character
  for(i in 1:ncol(df)) {
    df[,i] <- as.character(df[,i])
  }
  return(df)
})

# Combine all dataframes
games3 <- bind_rows(all_data_char)

games_with_preflop_3 <- games3 %>%
  filter(V8 != "-") %>%
  filter(!is.na(V12), !is.null(V12), V12 != "") %>%
  mutate(rank12 = substr(V12, 1, 1)) %>%
  mutate(rank13 = substr(V13, 1, 1)) %>%
  mutate(suit12 = substr(V12, 2, 2)) %>%
  mutate(suit13 = substr(V13, 2, 2)) %>%
  mutate(preflop = case_when(
    rank12 == rank13 & suit12 != suit13 ~ "Off-Suited Pair",
    rank12 != rank13 & suit12 == suit13 ~ "Suited Non-Pair",
    rank12 != rank13 & suit12 != suit13 ~ "Off-Suited Non-Pair"
  )) %>%
  mutate(wins = case_when(
    V11 > 0 ~ "# WINS",
    TRUE ~ "# LOSSES"
  )) %>%
  select(wins, preflop)
```

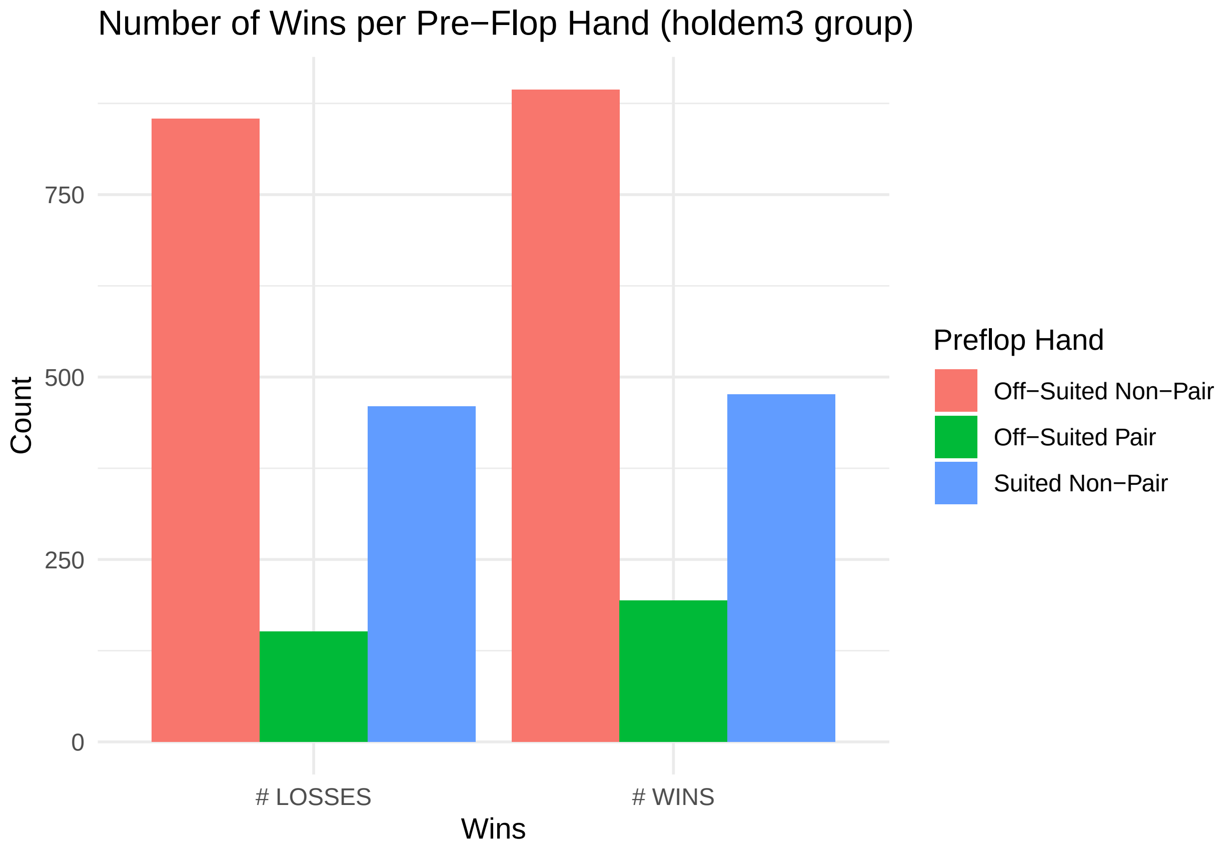
```
par(mfrow = c(1, 3))
ggplot(games_with_preflop_1, aes(x = wins, fill = preflop)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Number of Wins per Pre-Flop Hand (holdem group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



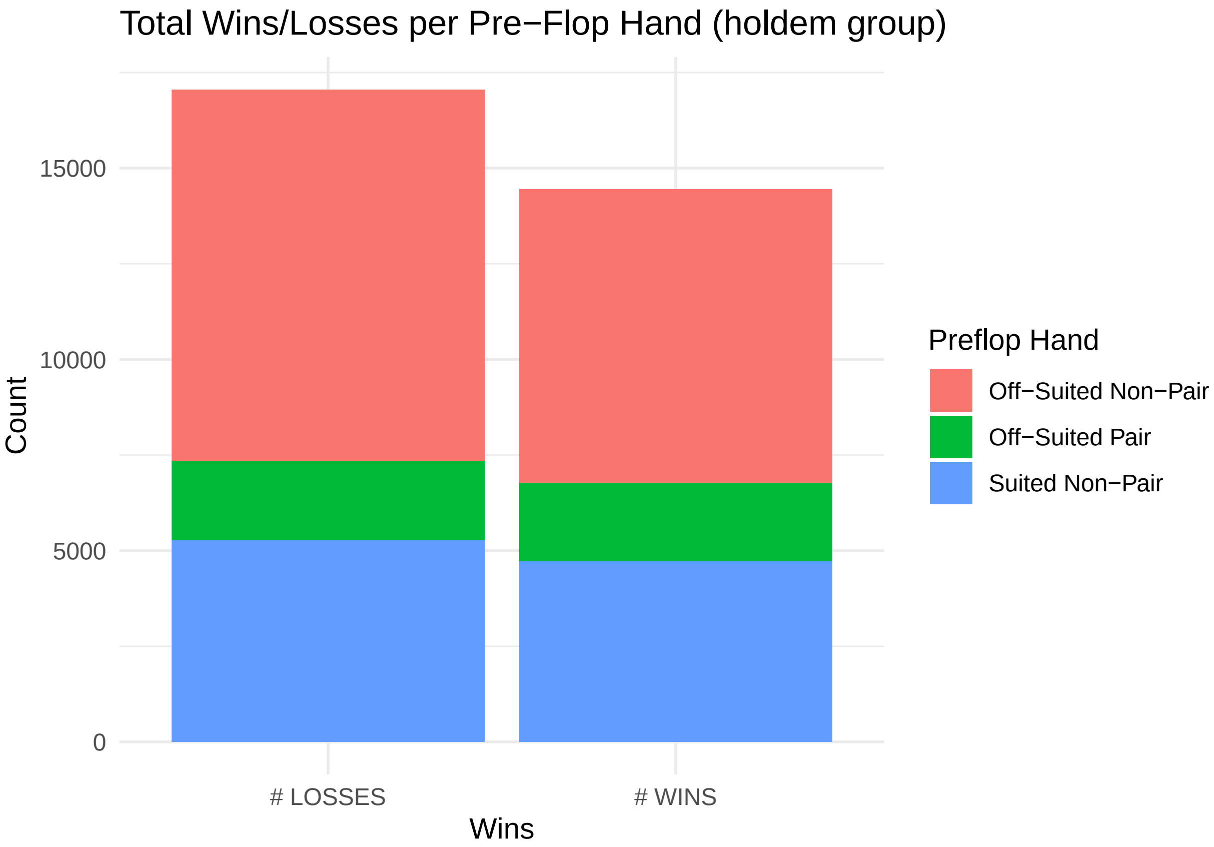
```r
ggplot(games_with_preflop_2, aes(x = wins, fill = preflop)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Number of Wins per Pre-Flop Hand (holdem2 group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



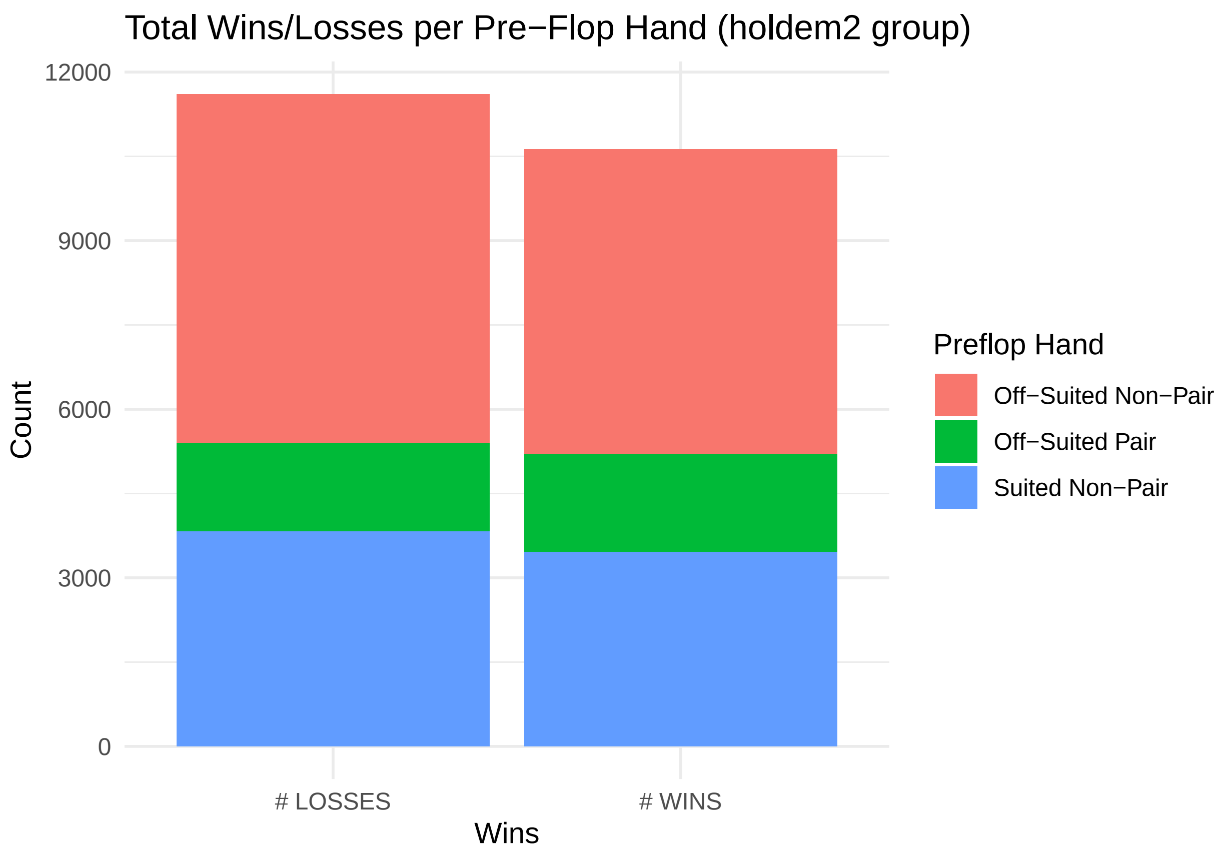
```r
ggplot(games_with_preflop_3, aes(x = wins, fill = preflop)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Number of Wins per Pre-Flop Hand (holdem3 group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



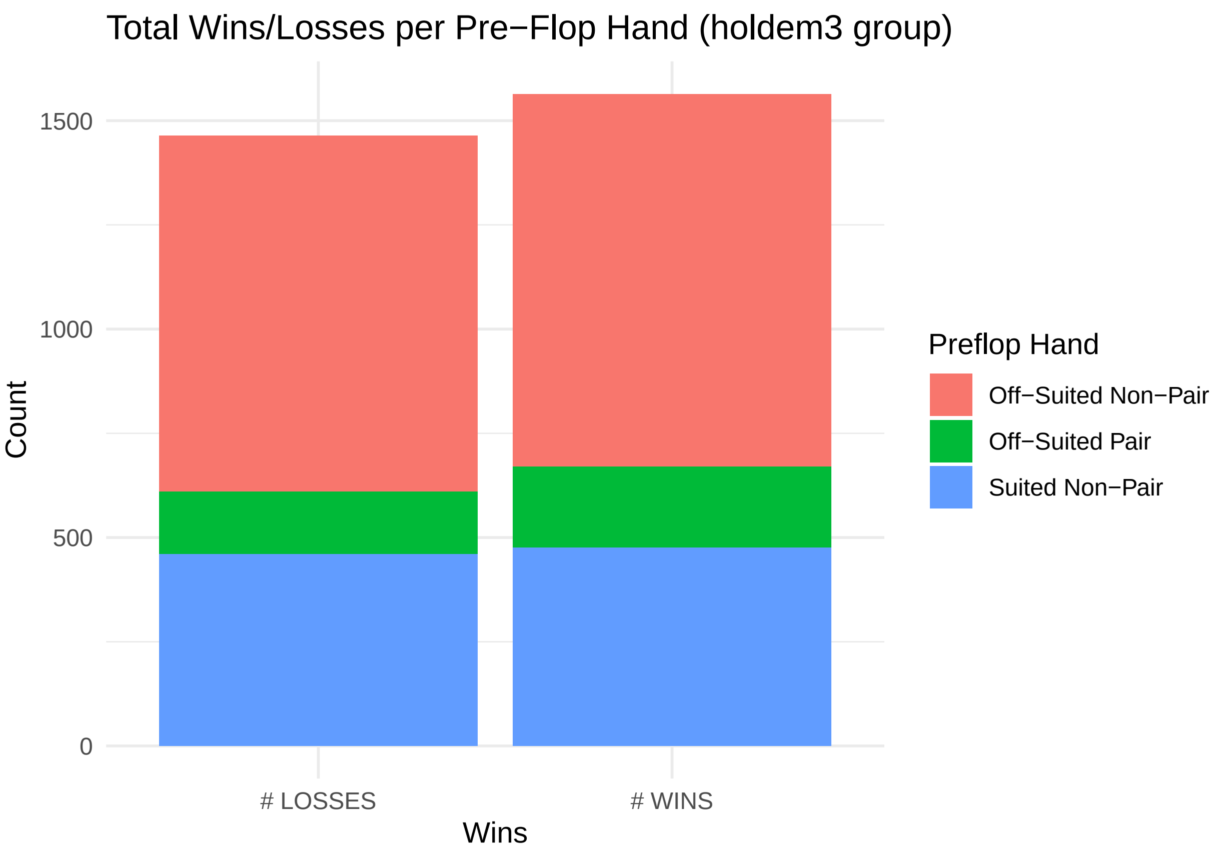
```
par(mfrow = c(1, 3))
ggplot(games_with_preflop_1, aes(x = wins, fill = preflop)) +
  geom_bar(position = "stack") +
  labs(
    title = "Total Wins/Losses per Pre-Flop Hand (holdem group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



```
ggplot(games_with_preflop_2, aes(x = wins, fill = preflop)) +
  geom_bar(position = "stack") +
  labs(
    title = "Total Wins/Losses per Pre-Flop Hand (holdem2 group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



```
ggplot(games_with_preflop_3, aes(x = wins, fill = preflop)) +
  geom_bar(position = "stack") +
  labs(
    title = "Total Wins/Losses per Pre-Flop Hand (holdem3 group)",
    x = "Wins",
    y = "Count",
    fill = "Preflop Hand"
  ) +
  theme_minimal()
```



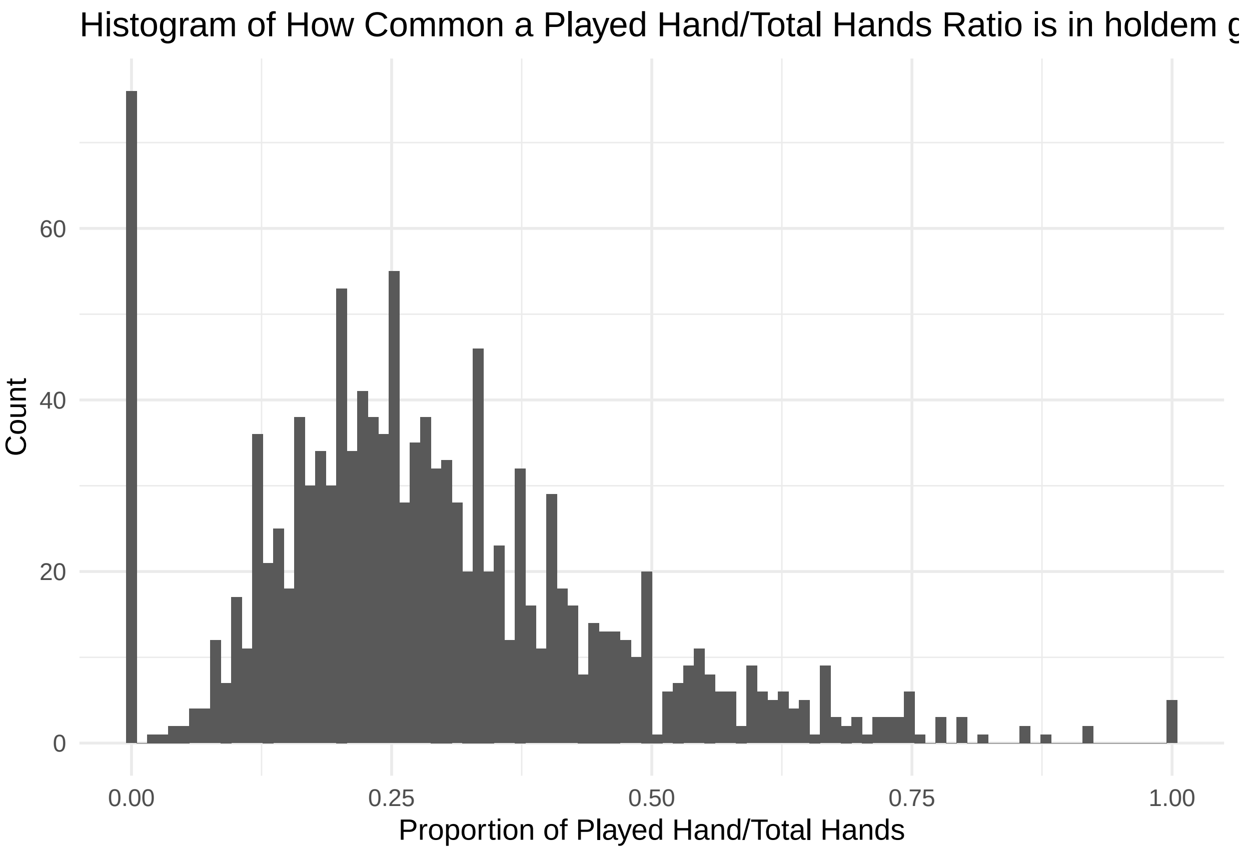
not including folds. these are comparing odds of winning in a showdown ONLY

- some missing values where the dealt cards were missing from the data despite the player not folding
- this doesn't tell us how much money was won with each win, nor how much money was lost in each loss.
- wanted to compare proportions
- doesn't include rounds where player won by getting default (everyone else folds)
- In least experienced groups, players seem to have more reliance on the dealt hand to win (wins/losses are least consistent in the worst dealt hand, the off-suited non-pair). Meanwhile, the most experienced group seems to know how to handle the off-suited non-pair hands, whether it's by bluffing or finding a good final hand with a straight. This could also imply that more experienced players know when to fold to minimize their losses
- The rarest starting hand, the pair, seems to generally allow players to win more than they lose, which means that the starting hand *can* help a player win a showdown, but as seen in the difference between newer players and experienced players, good poker players will also be able to make the most out of a bad starting hand.

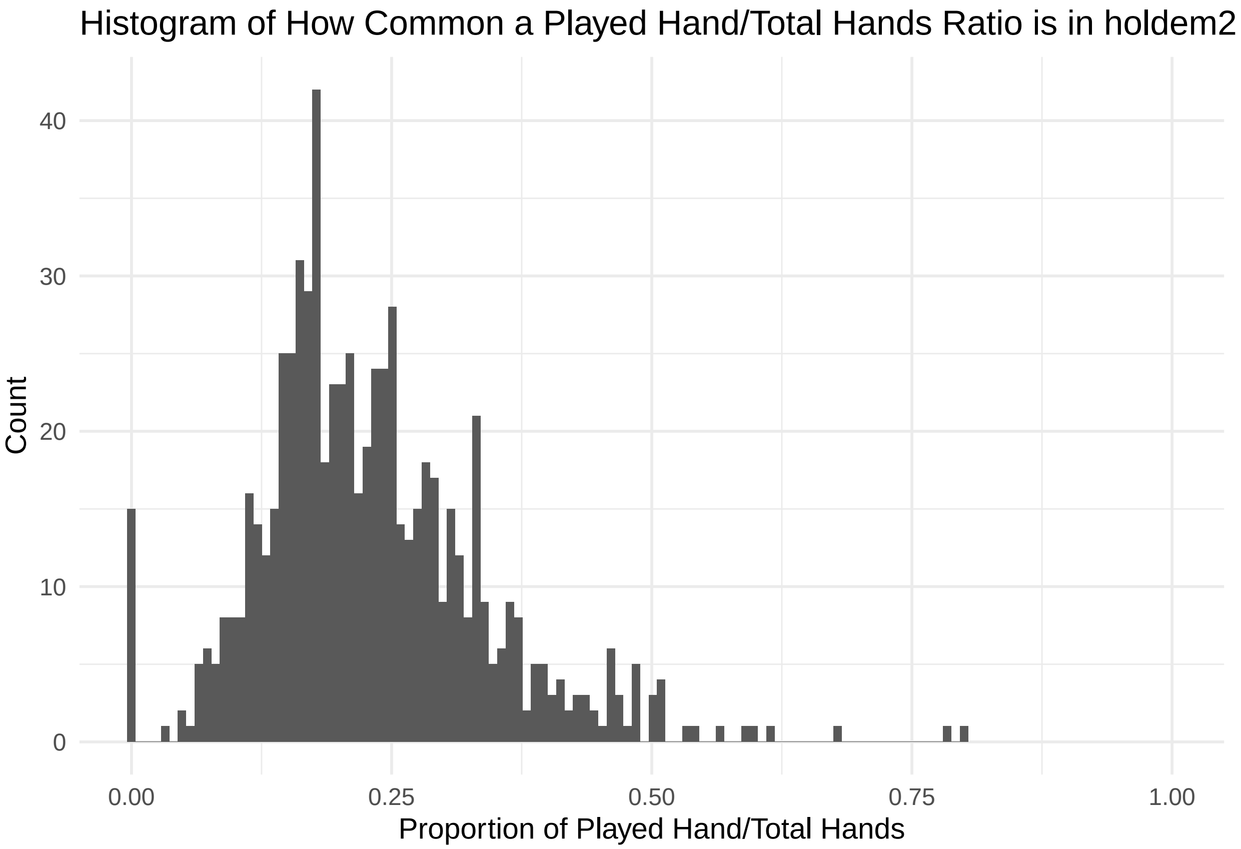
Frequency of Play

```
# Get list of files
filenames <- list.files("data/holdemtest")
```

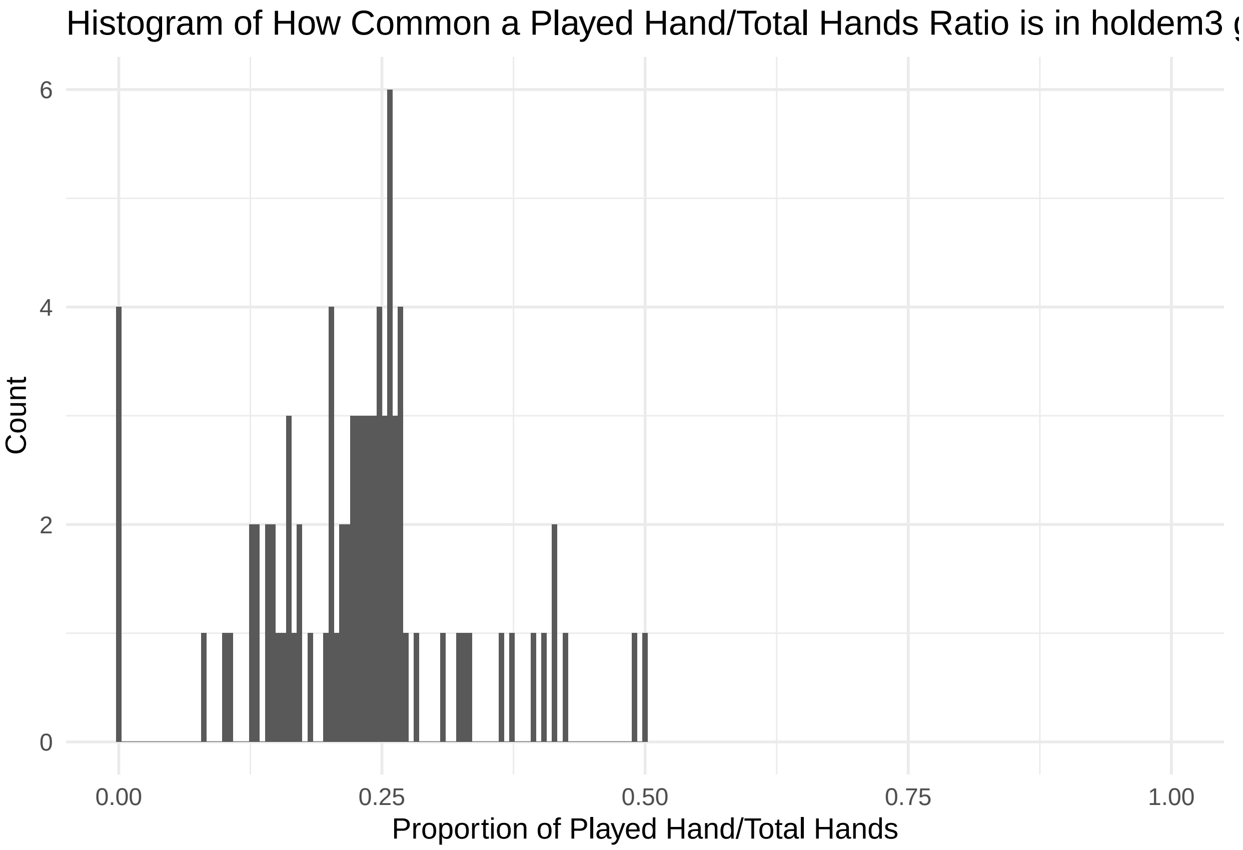
```r
# Create full file paths
full_paths <- paste("data/holdemtest", filenames, sep="/")

# Read files with read.table()
all_data <- lapply(full_paths, function(x) {
  read.table(x, header = FALSE, fill = TRUE, stringsAsFactors = FALSE)
})

# Find the maximum number of columns across all files
max_cols <- max(sapply(all_data, ncol))

# Convert all columns to character type to ensure consistency
all_data_char <- lapply(all_data, function(df) {
  # Convert all columns to character
  for(i in 1:ncol(df)) {
    df[,i] <- as.character(df[,i])
  }
  return(df)
})

# Combine all dataframes
gamestest <- bind_rows(all_data_char)

games_fold_bet_1 <- games3 %>%
  filter(!is.na(V3), !is.null(V3), V3 != "") %>%
  mutate(turn_folded = case_when(
    V5 == "-" ~ "Pre-Flop",
    V6 == "-" ~ "Flop",
    V7 == "-" ~ "Turn",
    V8 == "-" ~ "River",
    TRUE ~ "Didn't Fold"
  )) %>%
  mutate(bet = as.numeric(V10)) %>%
  select(V5, V6, V7, V8, bet, turn_folded)

games_played_hands_1 <- games1 %>%
  filter(!is.na(V3), !is.null(V3), V3 != "") %>%
  mutate(player = as.character(V1)) %>%
  group_by(player)%>%
  mutate(did_fold = case_when(
    V8 == "-" ~ 1,
    TRUE ~ 0
  )) %>%
  summarize(
    N = n(),
    num_folded = sum(did_fold),
    num_played = N-num_folded,
    proportion_played = num_played/(num_played+num_folded),
    start_bankroll = as.numeric(first(V9[which.min(as.numeric(V2))])),
    end_bankroll = as.numeric(first(V9[which.max(as.numeric(V2))])),
    got_money = case_when(
      start_bankroll-end_bankroll >= 0 ~ 1,
      TRUE ~ 0
    )
  )

games_played_hands_2 <- games2 %>%
  filter(!is.na(V3), !is.null(V3), V3 != "") %>%
  mutate(player = as.character(V1)) %>%
  group_by(player)%>%
  mutate(did_fold = case_when(
    V8 == "-" ~ 1,
    TRUE ~ 0
  )) %>%
  summarize(
    N = n(),
    num_folded = sum(did_fold),
    num_played = N-num_folded,
    proportion_played = num_played/(num_played+num_folded),
    start_bankroll = as.numeric(first(V9[which.min(as.numeric(V2))])),
    end_bankroll = as.numeric(first(V9[which.max(as.numeric(V2))])),
    got_money = case_when(
      start_bankroll-end_bankroll >= 0 ~ 1,
      TRUE ~ 0
    )
  )

games_played_hands_3 <- games3 %>%
  filter(!is.na(V3), !is.null(V3), V3 != "") %>%
  mutate(player = as.character(V1)) %>%
  group_by(player)%>%
  mutate(did_fold = case_when(
    V8 == "-" ~ 1,
    TRUE ~ 0
  )) %>%
  summarize(
    N = n(),
    num_folded = sum(did_fold),
    num_played = N-num_folded,
    proportion_played = num_played/(num_played+num_folded),
    start_bankroll = as.numeric(first(V9[which.min(as.numeric(V2))])),
    end_bankroll = as.numeric(first(V9[which.max(as.numeric(V2))])),
    got_money = case_when(
      start_bankroll-end_bankroll >= 0 ~ 1,
      TRUE ~ 0
    )
  )

#ggplot(data=games_fold_bet_1, aes(x=turn_folded, y=bet)) +
#  geom_boxplot(na.rm=T) +
#  coord_cartesian(ylim = c(0, 150))

ggplot(games_played_hands_1, aes(x=proportion_played))+
  geom_histogram(bins=100)+
  coord_cartesian(xlim = c(0, 1))+
  labs(
    title = "Histogram of How Common a Played Hand/Total Hands Ratio is in holdem group",
    x = "Proportion of Played Hand/Total Hands",
    y = "Count"
  ) +
  theme_minimal()
```



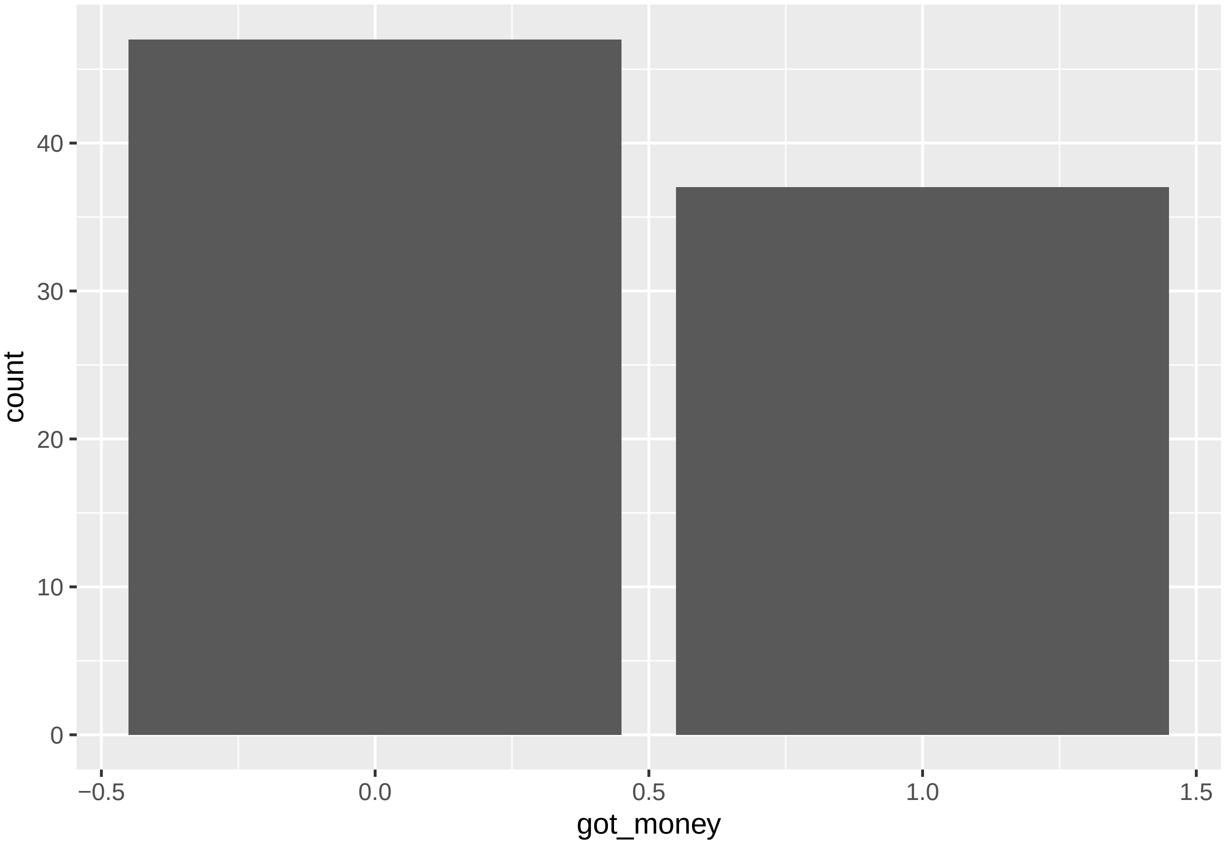
```
ggplot(games_played_hands_2, aes(x=proportion_played))+
  geom_histogram(bins=100)+
  coord_cartesian(xlim = c(0, 1))+
  labs(
    title = "Histogram of How Common a Played Hand/Total Hands Ratio is in holdem2 group",
    x = "Proportion of Played Hand/Total Hands",
    y = "Count"
  ) +
  theme_minimal()
```



```r
ggplot(games_played_hands_3, aes(x=proportion_played))+
  geom_histogram(bins=100)+
  coord_cartesian(xlim = c(0, 1))+
  labs(
    title = "Histogram of How Common a Played Hand/Total Hands Ratio is in holdem3 group",
    x = "Proportion of Played Hand/Total Hands",
    y = "Count"
  ) +
  theme_minimal()
```



```
ggplot(games_played_hands_3, aes(x = got_money)) +
  geom_bar()
```



- As experience increases, the desire to play hands tend to decrease. It seems that across the board, you shouldn't be playing much more than one hand per every four or five rounds on average.

It is clear from the preliminary data analysis that there are notable difference in the playstyles of experienced players and newer players. However, our analysis is currently too limited to make any actual conclusions about how we can reliably win at poker. Just refraining from playing a hand or making sure we don't get overconfident is good advice, but doesn't give as much information as we want. We need some way for all of our variables to be incorporated into the final decision. To do this, we use Bayesian inference to improve on our findings and to make better use of this data.

- is my starting hand playable?
- is this pot worth the risk?
- when should I fold?
- are my opponents bluffing?
- opponents will also consider all this…

TODO: - Add titles to plots - introduce buckets before first chart