

# 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](https://poker.cs.ualberta.ca/irc_poker_database.html).

Interpretations 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

```
library(tidyverse)

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

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

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
```

```

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)

```

```

# 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)

```

```

# 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)) %>%

```

```

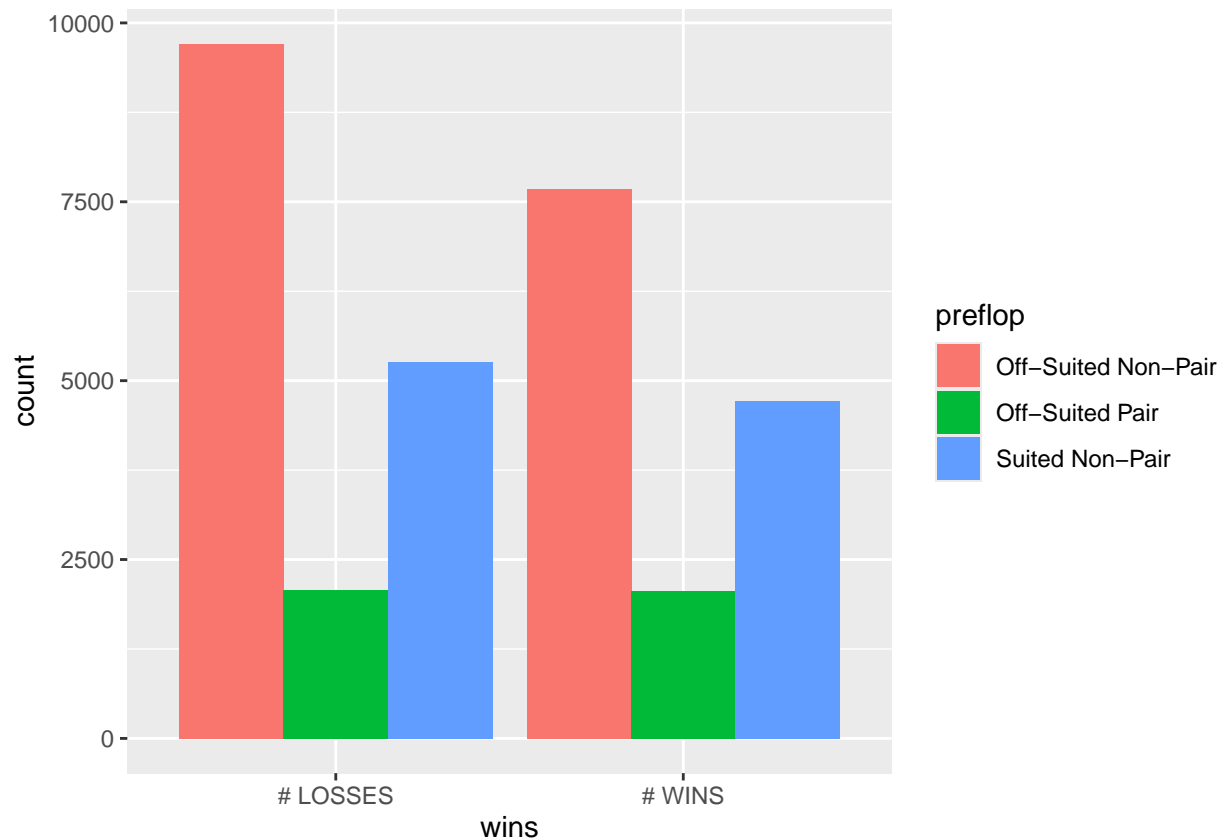
mutate(rank13 = substr(V13, 1, 1)) %>%
mutate(suit12 = substr(V12, 2, 2)) %>%
mutate(suit13 = substr(V13, 2, 2)) %>%
mutate(preflop = case_when(
  rank12 == rank13 & suit12 != suit13 ~ "Off-Suited Pair",
  rank12 != rank13 & suit12 == suit13 ~ "Suited Non-Pair",
  rank12 != rank13 & suit12 != suit13 ~ "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")

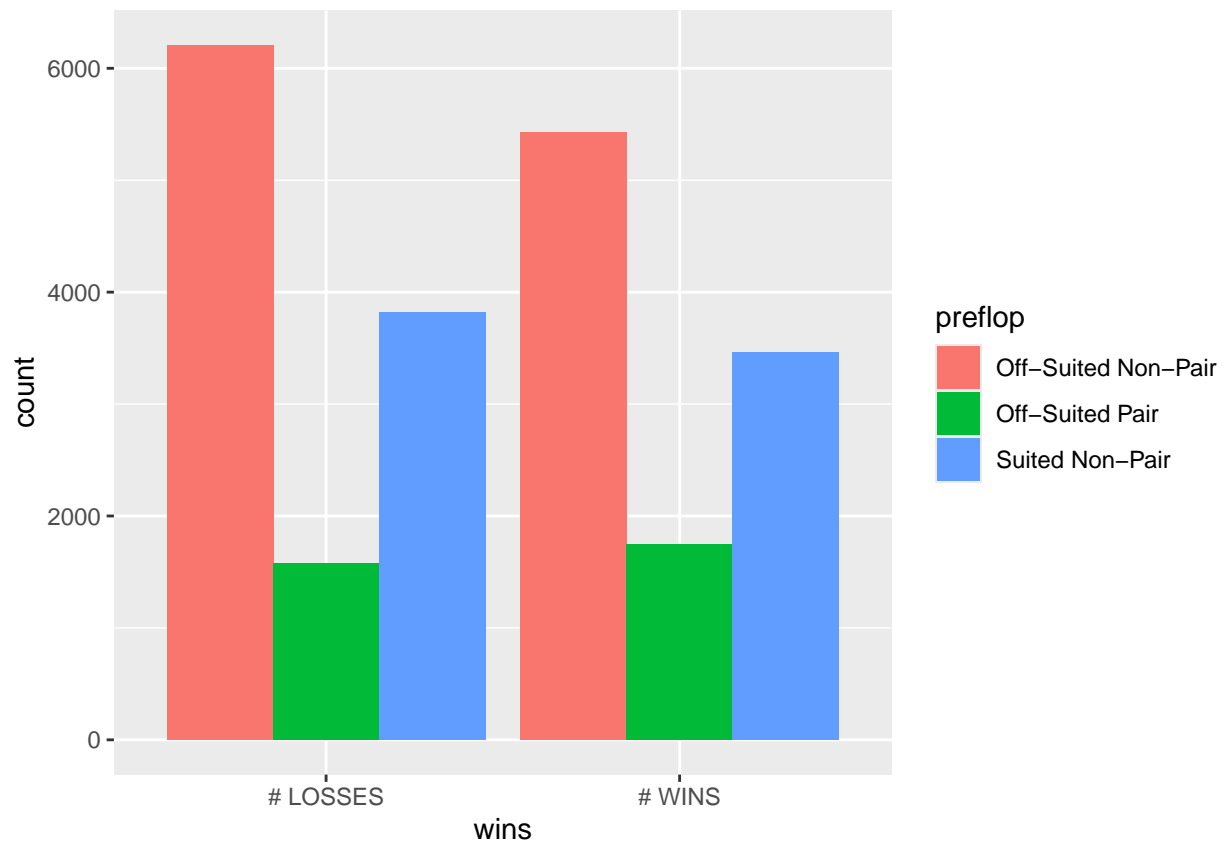
```



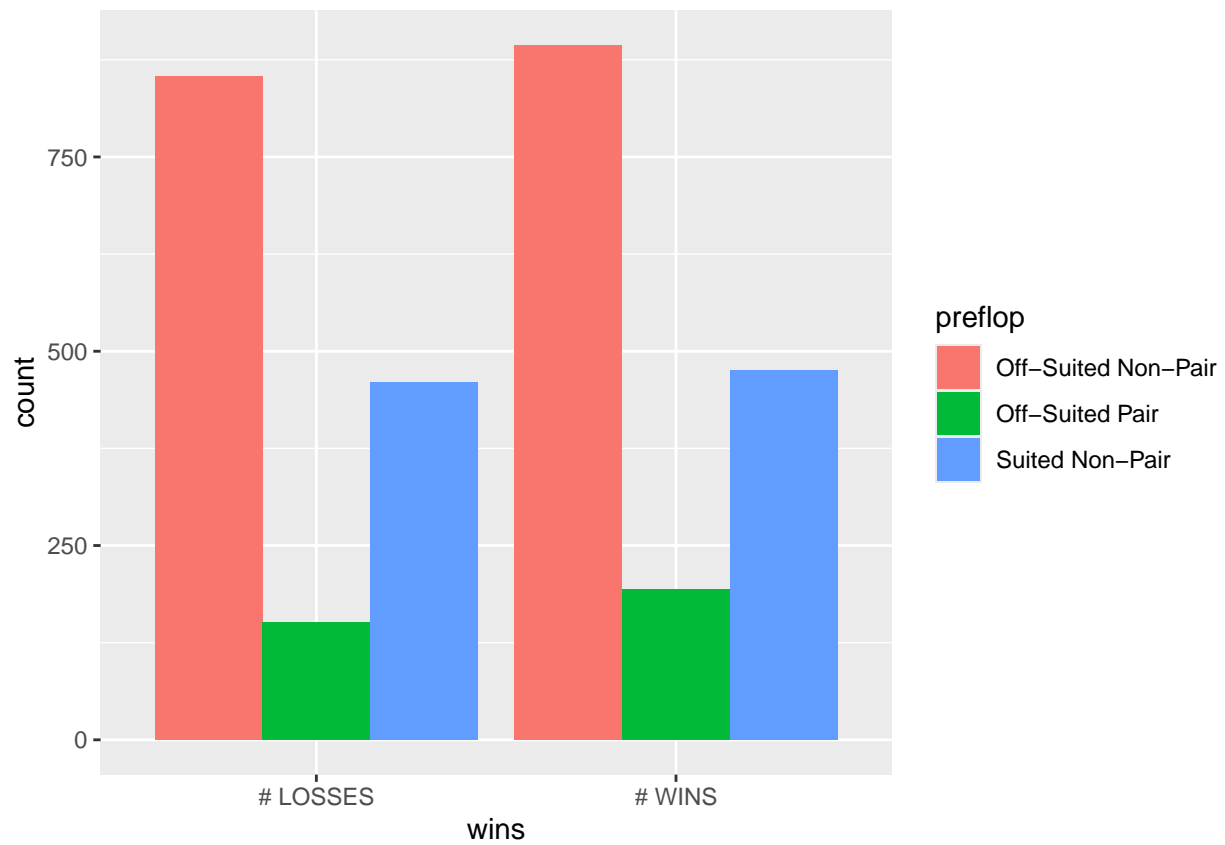
```

ggplot(games_with_preflop_2, aes(x = wins, fill = preflop)) +
  geom_bar(position = "dodge")

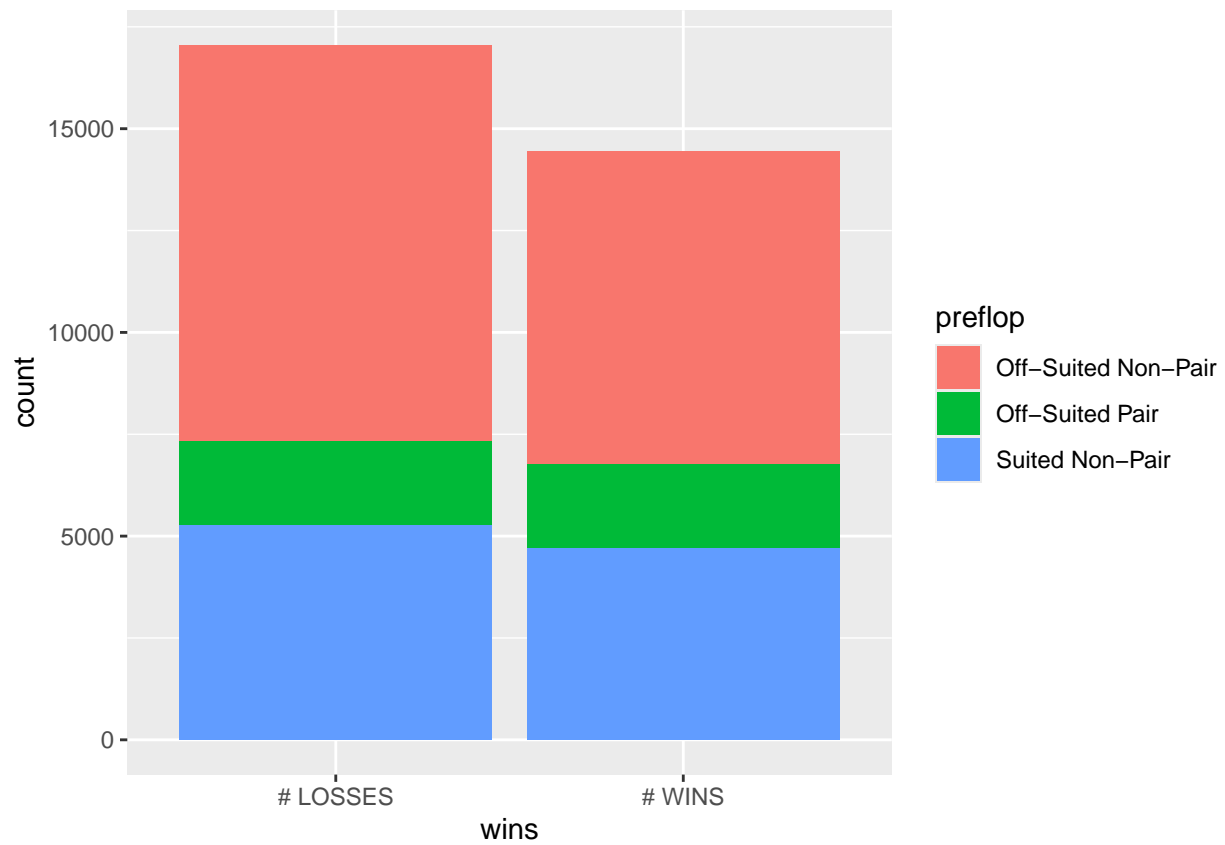
```



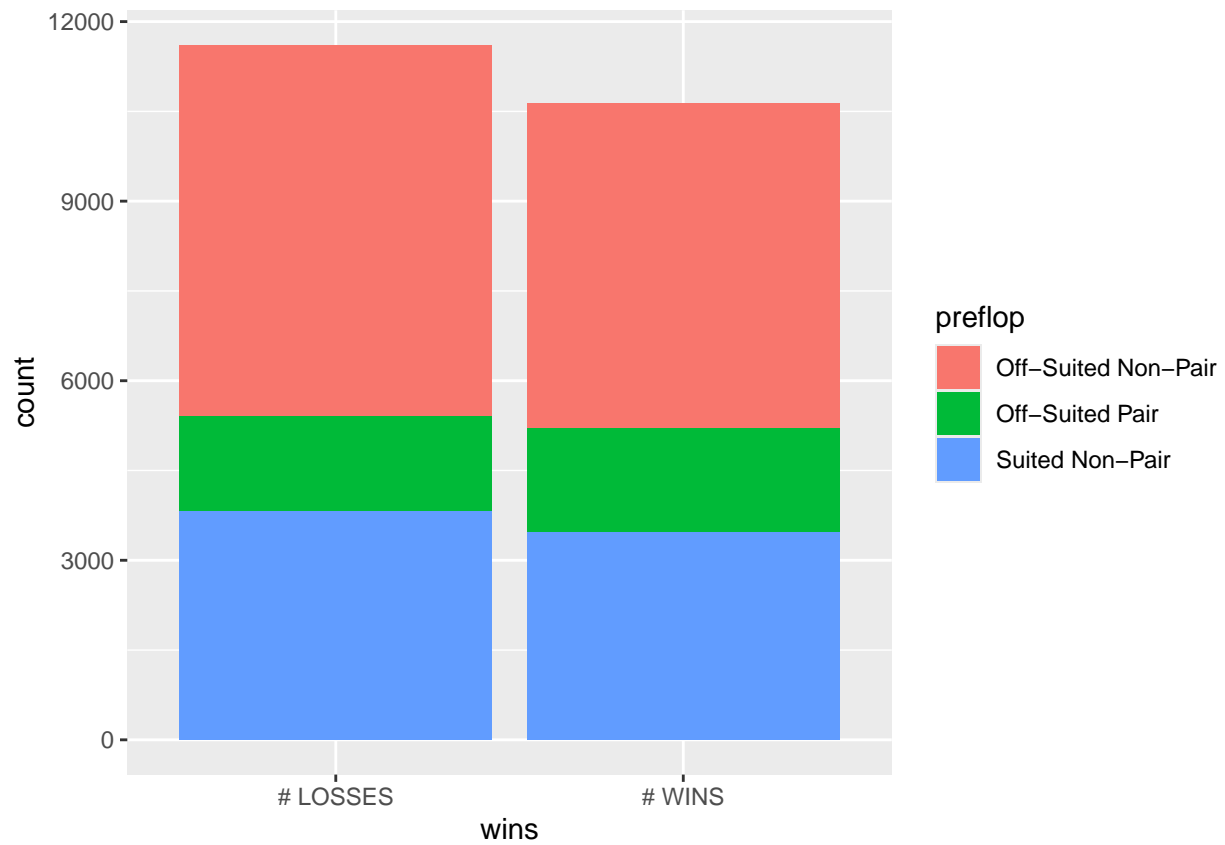
```
ggplot(games_with_preflop_3, aes(x = wins, fill = preflop)) +  
  geom_bar(position = "dodge")
```



```
par(mfrow = c(1, 3))
ggplot(games_with_preflop_1, aes(x = wins, fill = preflop)) +
  geom_bar(position = "stack")
```

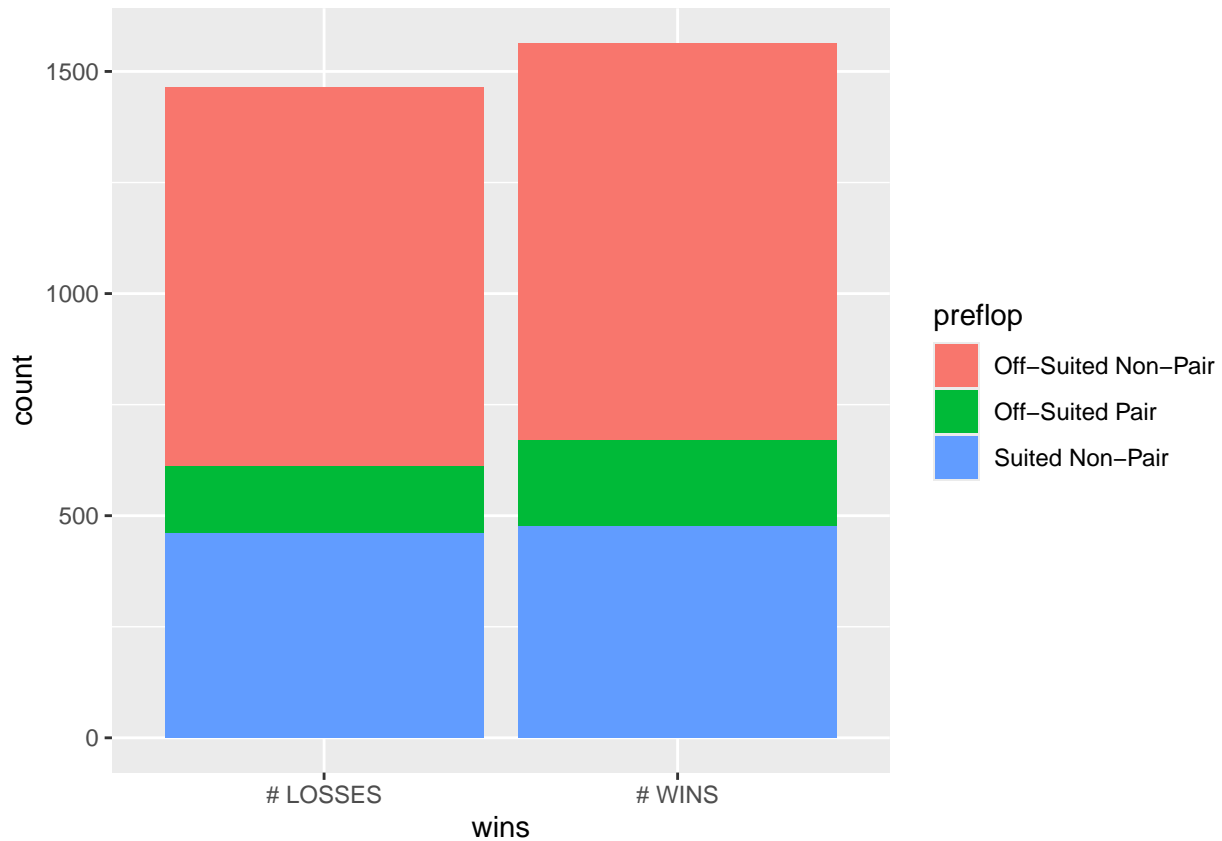


```
ggplot(games_with_preflop_2, aes(x = wins, fill = preflop)) +  
  geom_bar(position = "stack")
```



```
ggplot(games_with_preflop_3, aes(x = wins, fill = preflop)) +  
  geom_bar(position = "stack")
```





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 - 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.