Pset6

Problem 1

We read in the data from the csv file and then we fill use dplyr data pipping to select the variables. Then we use the group_by() and summarise() functions to perform the necessary analysis.

```
data <- read.csv("PitchFX_Filtered.csv")
```

```
#Only select important variables.
df <- data %>% select(type, true_strike, inning, Month, inning_side, count)
```

Problems 1a-d

```
df %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
##    % CldS S  % CldB B  % S CldS  % B CldB
## 1 0.7738335 0.9460259 0.8816607 0.8894956
```

Problem 1e - Two-Strike Counts Breakdown

```
#1e - Two Strikes
twostrikes <- levels(df$count)[c(3,6,9,12)]
df[df$count %in% twostrikes,] %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
##    % CldS S % CldB B % S CldS % B CldB
## 1 0.6778235 0.968025 0.754562 0.953955
```

Problem 1e - Three-Ball Counts Breakdown

```
#1e - Three Balls
threeBs <- levels(df$count)[c(10:12)]
df[df$count %in% threeBs,] %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
##    % CldS S % CldB B % S CldS  % B CldB
## 1 0.7733631 0.945856 0.906968 0.8594439
```

Problem 1f - Overall Count Breakdown

```
#1f
df %>%
  group_by(count) %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
## # A tibble: 12 x 5
##    count `% CldS S` `% CldB B` `% S CldS` `% B CldB`
##    <fct>      <dbl>      <dbl>      <dbl>      <dbl>
##  1 0-0        0.806      0.929      0.908      0.845
##  2 0-1        0.755      0.946      0.803      0.929
##  3 0-2        0.691      0.971      0.702      0.970
##  4 1-0        0.760      0.939      0.903      0.840
##  5 1-1        0.729      0.949      0.835      0.909
##  6 1-2        0.675      0.970      0.738      0.960
##  7 2-0        0.753      0.942      0.920      0.810
##  8 2-1        0.713      0.948      0.857      0.884
##  9 2-2        0.668      0.966      0.782      0.941
## 10 3-0        0.827      0.928      0.955      0.742
## 11 3-1        0.741      0.939      0.889      0.846
## 12 3-2        0.684      0.956      0.805      0.920
```

Problem 1g - Inning Breakdown

```
#1g
df %>%
  group_by(inning) %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
## # A tibble: 19 x 5
##    inning `% CldS S` `% CldB B` `% S CldS` `% B CldB`
##     <int>      <dbl>      <dbl>      <dbl>      <dbl>
##  1      1      0.776      0.944      0.884      0.885
##  2      2      0.783      0.943      0.883      0.888
##  3      3      0.781      0.946      0.888      0.888
##  4      4      0.767      0.947      0.879      0.890
##  5      5      0.772      0.947      0.883      0.889
##  6      6      0.767      0.949      0.880      0.893
##  7      7      0.773      0.947      0.880      0.893
##  8      8      0.775      0.945      0.876      0.893
##  9      9      0.769      0.946      0.884      0.884
## 10     10      0.772      0.943      0.865      0.898
## 11     11      0.752      0.947      0.867      0.894
## 12     12      0.777      0.947      0.876      0.898
## 13     13      0.753      0.934      0.837      0.894
## 14     14      0.762      0.929      0.842      0.886
## 15     15      0.731      0.950      0.891      0.865
## 16     16      0.762      0.942      0.877      0.879
## 17     17      0.741      0.959      0.870      0.910
## 18     18      0.824      1.00       1.00       0.922
## 19     19      0.800      0.833      0.571      0.938
```

Problem 1h - Month/Season Breakdown

```
#1h
df$MonthNew <- df$Month
```

```
df$MonthNew[df$Month == 2] <- "First Month"
df$MonthNew[df$Month == 10] <- "Last Month"
df$MonthNew[df$MonthNew != "First Month" & df$MonthNew != "Last Month"] <- "Rest of Season"

df %>%
  group_by(MonthNew) %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
## # A tibble: 3 x 5
##   MonthNew       `% CldS S` `% CldB B` `% S CldS` `% B CldB`
##   <chr>               <dbl>      <dbl>      <dbl>      <dbl>
## 1 First Month         0.802      0.946      0.869      0.915
## 2 Last Month          0.762      0.949      0.884      0.888
## 3 Rest of Season      0.774      0.946      0.882      0.889
```

Problem 1i - Side of the Inning Breakdown

Note: bottom means it was the bottom of the inning so this is home batters, top means top of the inning which indicates away batters

```
#1i
df %>%
  group_by(inning_side) %>%
  summarise("% CldS S" = table(type,true_strike)[4]/sum(type == "S"),
         "% CldB B" = table(type,true_strike)[1]/sum(type == "B"),
         "% S CldS" = table(type,true_strike)[4]/sum(true_strike == 1),
         "% B CldB" = table(type,true_strike)[1]/sum(true_strike == 0))
```

```
## # A tibble: 2 x 5
##   inning_side `% CldS S` `% CldB B` `% S CldS` `% B CldB`
##   <fct>            <dbl>      <dbl>      <dbl>      <dbl>
## 1 bottom           0.776      0.946      0.880      0.892
## 2 top              0.772      0.946      0.883      0.887
```

Comments:

Looking at part 1a), the umpires call a strike 78% of the time when it should be called one. From 1c, we see that it is a strike 88% of the time that they call it a strike. Similarly for balls, its called a ball 95% of the time it is actually a ball and is a ball 89% of the time it is called a ball. This shows us that umpires are not perfect, and could be subject to several factors: the movement of the pitch may make it more difficult to call a strike if it just passes the edge of the zone, the strike zone is smaller so its less likely to get very clear strike calls, or the catcher can frame a pitch or drop a good pitch, which could influence the umpires decision.

When looking at two strike counts, both the percentage of called strikes that are actually strikes and the percentage of strikes that are properly called strikes decrease significantly. Here we see the umpire subject to the omission bias, as he is less willing to make a terminal call. The called strike is less likely to be a strike (1a), and are more likely to miscall a strike as a ball (1c). The answer from 1a suggests that perhaps when a pitcher makes a good pitch that “hits their spot” but is not actually a strike, the umpire rewards the quality pitch with a strike call. This supports the idea of the pitcher and catcher being able to extend the zone and frame slightly.

For three ball counts, the percentage of called balls that are actually balls remain relatively similar compared to the overall percentage. This illustrates that umpires are unwilling to make an egregiously wrong call even with the omission bias. However, percentage of actual balls that are called balls drops slightly, illustrating that if the pitch is close they are more likely to call this actual ball a strike. This supports the presence of an omission bias.

An important note is the difference in magnitude between the two-strike and three-ball counts. This could be because the umpire views it has a bigger action to call someone out than to put a runner on base, so is more willing to make the correct call in three ball counts.

Umpire performance does not change for home vs. away batters.

By month of season, there might be a very small decrease in accuracy of umpires as the games become slightly more impactful for the overall season (ie later in the season). The umps could also be getting fatigued.

There are slight fluctuations by inning, but they are all within ± 1.5% of the average. There is not a clear enough trend to draw any significant conclusions.

##Problem 2

```
df2 <- df
```

```
#Indicator if the umpire made a missed call on a pitch.
df2$MissedCall <- ifelse((df2$true_strike == 0 & df2$type == "S") |
                           (df2$true_strike == 1 & df2$type == "B"),1,0)

#Indicator on if the previous pitch was a missed call and this pitch is a missed call.
df2$NextPitchMissedCall <- ifelse(lag(df2$MissedCall,1) == 1 &
                                    ((df2$true_strike == 0 & df2$type == "S") |
                                    (df2$true_strike == 1 & df2$type == "B")),1,0)
```

2a

```
#Percentage of total pitches miss called.
perc_totalmissed <- 1-sum(df2$MissedCall)/nrow(df2)
print(paste("Percentage of total pitches called correctly:",perc_totalmissed))
```

```
## [1] "Percentage of total pitches called correctly: 0.887144093261126"
```

```
#Percentage of next pitch missed call.
perc_nextmissed <- 1-sum(df2$NextPitchMissedCall)/sum(df2$MissedCall)
print(paste("Percentage of next pitch called correctly:",perc_nextmissed))
```

```
## [1] "Percentage of next pitch called correctly: 0.886623499232332"
```

They appear roughly equally likely to call the next pitch correctly whether or not the previous pitch was a missed call.

2b

```
#Two indicators for if a true strike/ball was called ball/strike.
df2$BcallS <- ifelse((df2$true_strike == 0 & df2$type == "S"),1,0)
df2$ScallB <- ifelse((df2$true_strike == 1 & df2$type == "B"),1,0)
```

```
df2$ReversedError <- ifelse(((df2$NextPitchMissedCall == 1) & (df2$BcallS == lag(df2$ScallB,1))) |
                              ((df2$NextPitchMissedCall == 1) & (df2$ScallB == lag(df2$BcallS,1))),1,0)
```

```
#Percentage of total missed calls they followed with a reverse missed call.
perc_reverror <- sum(df2$ReversedError)/sum(df2$MissedCall)
print(paste("Percentage of missed calls that are a reverse missed call:",perc_reverror))
```

```
## [1] "Percentage of missed calls that are a reverse missed call: 0.0501988308554972"
```

```r
#Percentage of next pitch missed calls that are a reverse missed call.
perc_reverror2 <- sum(df2$ReversedError)/sum(df2$NextPitchMissedCall)
print(paste("Percentage of next pitch missed calls that are a reverse missed call:",perc_reverror2))
```

```
## [1] "Percentage of next pitch missed calls that are a reverse missed call: 0.44276221717554"
```

They are less likely to make an error on the next called pitch but in the opposite direction than they are to just make another missed call (44.3% < 50%).

2c We make a new dataframe that only includes counts with 2 strikes or 3 balls (where a missed call could result in a terminal call).

```r
terminalcounts <- levels(df2$count)[c(3,6,9:12)]
df2_terminal <- df2[df2$count %in% terminalcounts,]
```

```r
#Percentage of total pitches miss called.
perc_totalmissed <- 1-sum(df2_terminal$MissedCall)/nrow(df2_terminal)
print(paste("Percentage of total pitches called correctly:",perc_totalmissed))
```

```
## [1] "Percentage of total pitches called correctly: 0.920469427047195"
```

```r
#Percentage of next pitch missed call.
perc_nextmissed <- 1-sum(df2_terminal$NextPitchMissedCall)/sum(df2_terminal$MissedCall)
print(paste("Percentage of next pitch called correctly:",perc_nextmissed))
```

```
## [1] "Percentage of next pitch called correctly: 0.892939681072336"
```

```r
#Percentage of total missed calls they followed with a reverse missed call.
perc_reverror <- sum(df2_terminal$ReversedError)/sum(df2_terminal$MissedCall)
print(paste("Percentage of missed calls that are a reverse missed call:",perc_reverror))
```

```
## [1] "Percentage of missed calls that are a reverse missed call: 0.0488213542870349"
```

```r
#Percentage of next pitch missed calls that are a reverse missed call.
perc_reverror2 <- sum(df2_terminal$ReversedError)/sum(df2_terminal$NextPitchMissedCall)
print(paste("Percentage of next pitch missed calls that are a reverse missed call:",perc_reverror2))
```

```
## [1] "Percentage of next pitch missed calls that are a reverse missed call: 0.456017269293038"
```

We see that the percentage of missed calls that are a reverse missed call decreases in terminal counts (5.0% to 4.9%). We also see that the percentage of next pitch missed calls that are a reverse missed call increases (44.3% to 45.6%). We also see that overall, umpires are less likely to make a correct call following a missed call. This can suggest that umpires do not want to compound their previous mistake into a terminal decision. Additionally, they may be making more reverse missed calls to make ammend for a missed call in a potentially terminal count.

2d If it is later in the game, the game is close, or there are runners on base; we might see that the umpire is more concious of the impact of their call and could be more likely to make a reverse missed call if it will prevent a terminal decision (i.e. omission bias might be greater in more impactful situations). Additionally, in unimportant situations (top of an inning, beginning of the game, nobody on base), umpires may be less likely to concsiouly reverse missed calls because the moment is not impactful. Thus, the umpire is under less pressure and scrutiny and a correctly terminal call will not be as impactful.