Data project 2: Game developer A/B testing

Your name

```
## # A tibble: 90,189 x 5
##       id group games_played retained_1 retained_7
##    <dbl> <chr>        <dbl> <lgl>      <lgl>
##  1   116 A                3 FALSE      FALSE
##  2   337 A               38 TRUE       FALSE
##  3   377 B              165 TRUE       FALSE
##  4   483 B                1 FALSE      FALSE
##  5   488 B              179 TRUE       TRUE
##  6   540 B              187 TRUE       TRUE
##  7  1066 A                0 FALSE      FALSE
##  8  1444 B                2 FALSE      FALSE
##  9  1574 B              108 TRUE       TRUE
## 10  1587 B              153 TRUE       FALSE
## # ... with 90,179 more rows
```

Problem 1: Exploring the data

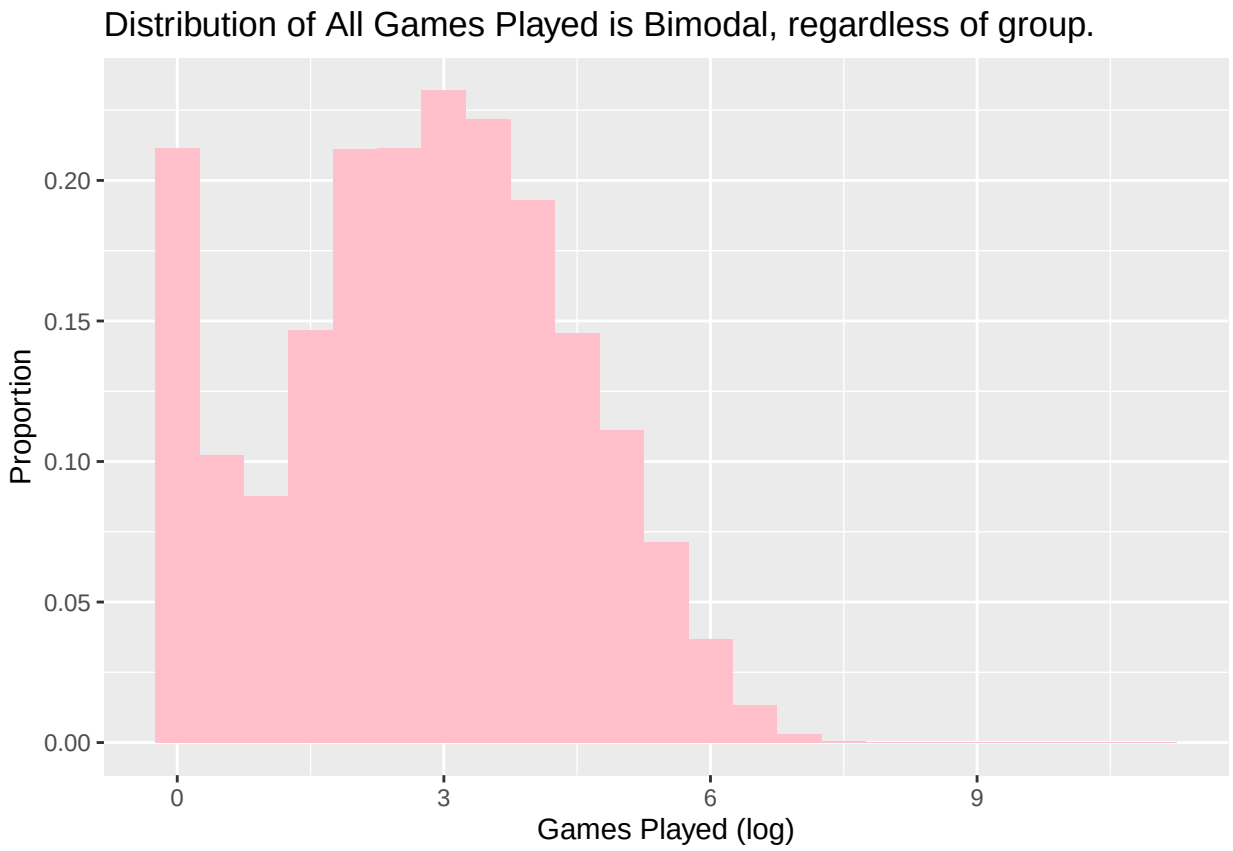
Part a

Since the original graph had some extreme outliers (max games played was about 50k), it made our data very skewed and not that informative, so we did a logarithmic transformation on the data to gain more insight. As we can see below, this data is definitely bimodal, with a good amount of players (over 20% not playing anymore than a couple games).

```
game_data$log_games = log(game_data$games_played)
game_data[game_data < 0] = 0
game_data
```

```
## # A tibble: 90,189 x 6
##       id group games_played retained_1 retained_7 log_games
##    <dbl> <chr>        <dbl> <lgl>      <lgl>          <dbl>
##  1   116 A                3 FALSE      FALSE          1.10
##  2   337 A               38 TRUE       FALSE          3.64
##  3   377 B              165 TRUE       FALSE          5.11
##  4   483 B                1 FALSE      FALSE          0
##  5   488 B              179 TRUE       TRUE           5.19
##  6   540 B              187 TRUE       TRUE           5.23
##  7  1066 A                0 FALSE      FALSE          0
##  8  1444 B                2 FALSE      FALSE          0.693
##  9  1574 B              108 TRUE       TRUE           4.68
## 10  1587 B              153 TRUE       FALSE          5.03
## # ... with 90,179 more rows
```

```
ggplot(game_data, aes(x=log_games)) +
  geom_histogram(aes(y = ..density..), binwidth = .5, , fill = "pink")+
  ylab("Proportion") +
  xlab("Games Played (log)") +
  labs(title = "Distribution of All Games Played is Bimodal, regardless of group.")
```

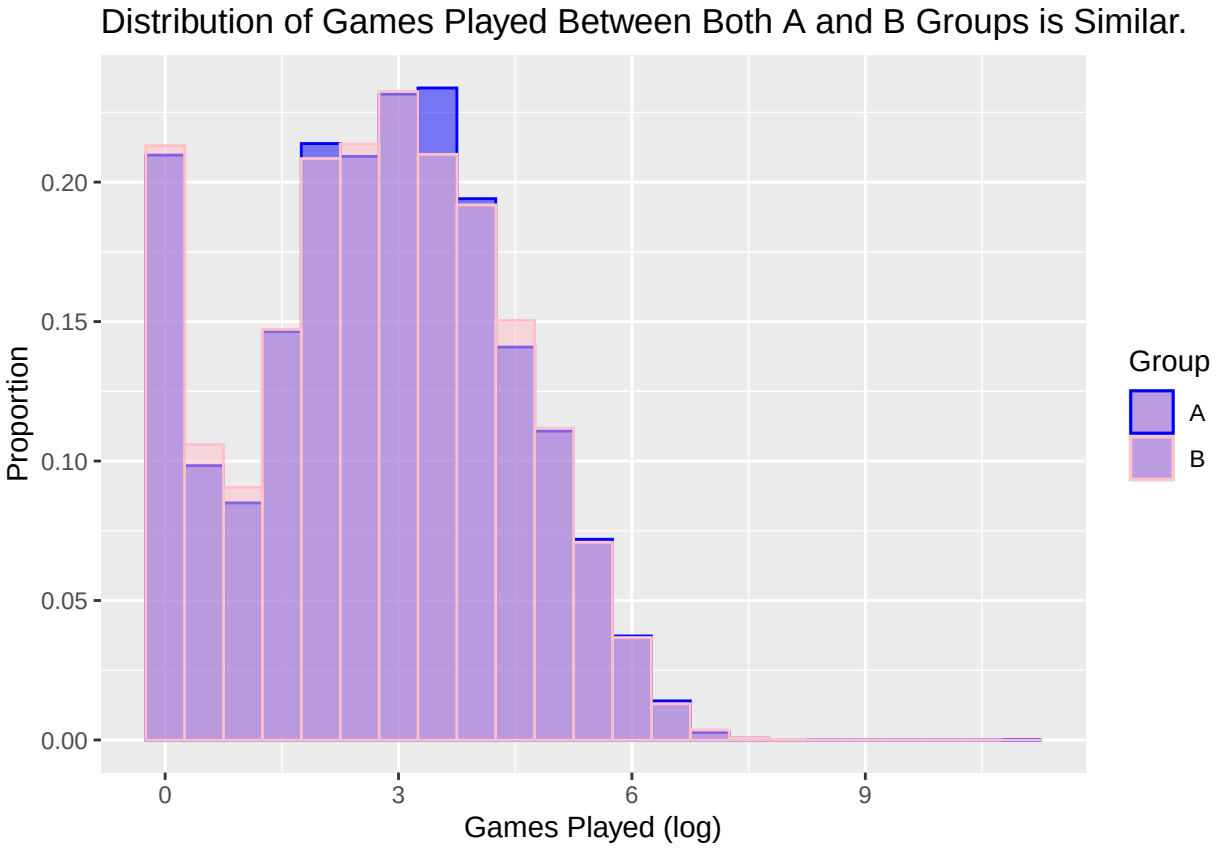


Part b

Since both distributions of how much of each game was played visually looks similar, I would not recommend one game version over the other, and would use tools such as hypothesis testing to determine if there is enough of a difference to determine which is more successful.

```
data_A = filter(game_data, group == "A")
data_B = filter(game_data, group == "B")
colors <- c("A" = "blue", "B" = "pink")

ggplot(data_A, aes(x=log_games)) +
  geom_histogram(aes(y = ..density.., color='A'), alpha=0.5, binwidth = .5, fill = 'blue') +
  geom_histogram(data = data_B, aes(y = ..density.., x = log_games, color='B'), binwidth = .5, alpha=0.5
  ylab("Proportion") +
  xlab("Games Played (log)") +
  labs(title = "Distribution of Games Played Between Both A and B Groups is Similar.") +
  scale_color_manual(name='Group',
                     breaks=c('A', 'B'),
                     values=c('A'='blue', 'B'='pink'))
```



Part c

```
A_1 = filter(data_A, retained_1)
A_7 = filter(data_A, retained_7)
B_1 = filter(data_B, retained_1)
B_7 = filter(data_B, retained_7)
print("Prop of users retained after one day in group A:")
```

```
## [1] "Prop of users retained after one day in group A:"
```

```
a1p = mean(data_A$retained_1)
a1p
```

```
## [1] 0.4481879
```

```
print("Prop of users retained after seven days in group A:")
```

```
## [1] "Prop of users retained after seven days in group A:"
```

```
a7p = mean(data_A$retained_7)
a7p
```

```
## [1] 0.1902013
```

```
print("Prop of users retained after one day in group B:")
```

```
## [1] "Prop of users retained after one day in group B:"
```

```
b1p = mean(data_B$retained_1)
b1p
```

```
## [1] 0.4422827
```

```
print("Prop of users retained after seven days in group B:")
```

```
## [1] "Prop of users retained after seven days in group B:"
```

```
b7p = mean(data_B$retained_7)
b7p
```

```
## [1] 0.182
```

Problem 2: Bootstrapped estimates

Part a

The 95% confidence interval done theoretically for the mean amount of games played is [50.59947, 53.14545].

```
#reference: ci.R in course files
alpha_2a = .05
mean_2a = mean(game_data$games_played)
sd_2a = sd(game_data$games_played)
n = nrow(game_data)
t = qt(1 - alpha_2a/2, n-1)
theoretical_ci = c(mean_2a - t * sd_2a / sqrt(n),
                   mean_2a + t * sd_2a / sqrt(n))
theoretical_ci
```

```
## [1] 50.59947 53.14545
```

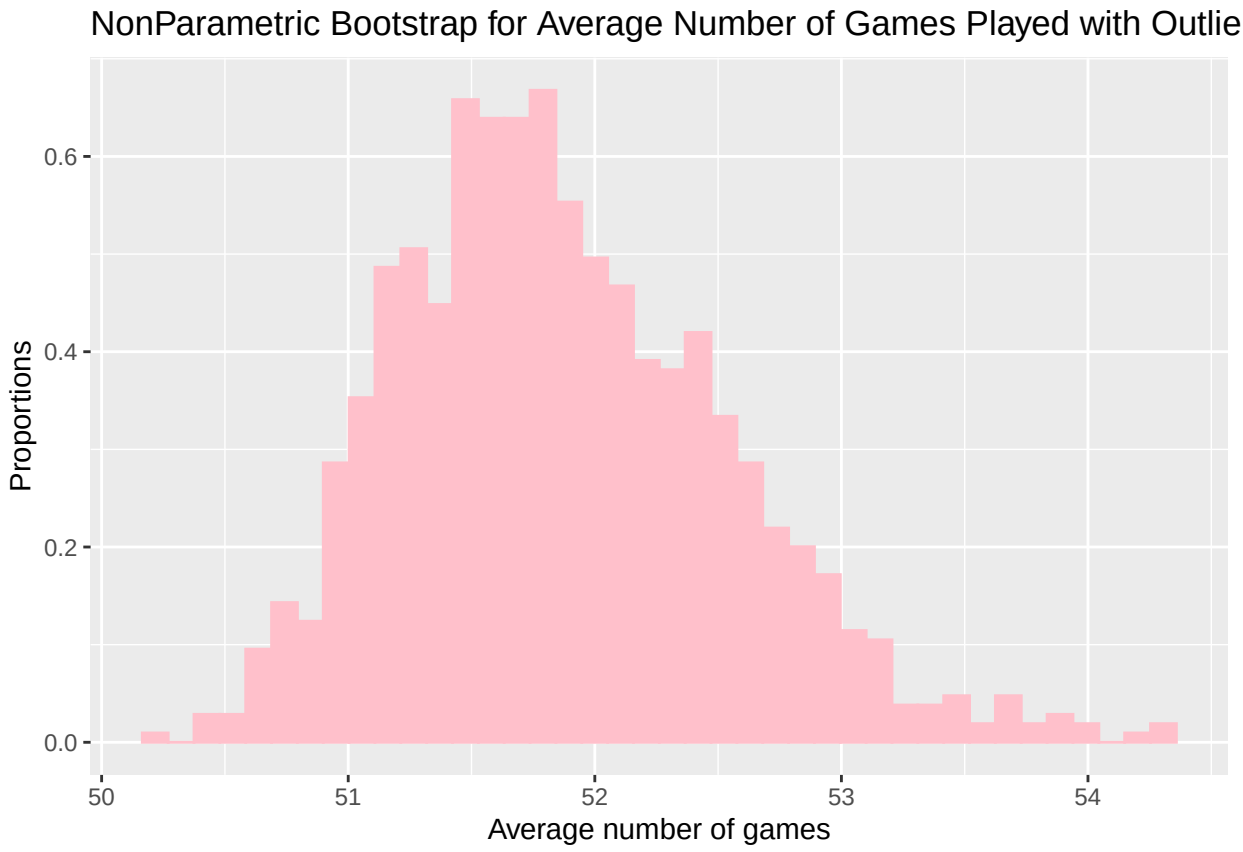
```
game_data_f = filter(game_data, games_played < 40000)
alpha_2a = .05
mean_2a = mean(game_data_f$games_played)
sd_2a = sd(game_data_f$games_played)
n = nrow(game_data)
t = qt(1 - alpha_2a/2, n-1)
theoretical_ci = c(mean_2a - t * sd_2a / sqrt(n),
                   mean_2a + t * sd_2a / sqrt(n))
theoretical_ci
```

```
## [1] 50.65010 51.99041
```

Part b

Regardless if outlier of about 50k games played is included or not, it seems that there is no determinable difference between the two CI of both groupings. The confidence intervals that do not include the outlier are within .01 games of each other. The bootstrapped confidence intervals that included the outlier is a little tighter than the theoretical confidence interval that included the outlier, but only by about 1.1 games, which is pretty significant.

```
#reference: bootstrap_mean.R in course files
set.seed(15)
np_2b = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2b = sample(game_data$games_played, length(game_data$games_played), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2b = mean(b_2b))
})
#plot histogram of values
ggplot(np_2b, aes(x=b_2b, y = ..density..,)) +
  geom_histogram(colour="pink", fill="pink", bins = 40) +
  xlab("Average number of games") +
  ylab("Proportions") +
  labs(title = "NonParametric Bootstrap for Average Number of Games Played with Outlier")
```

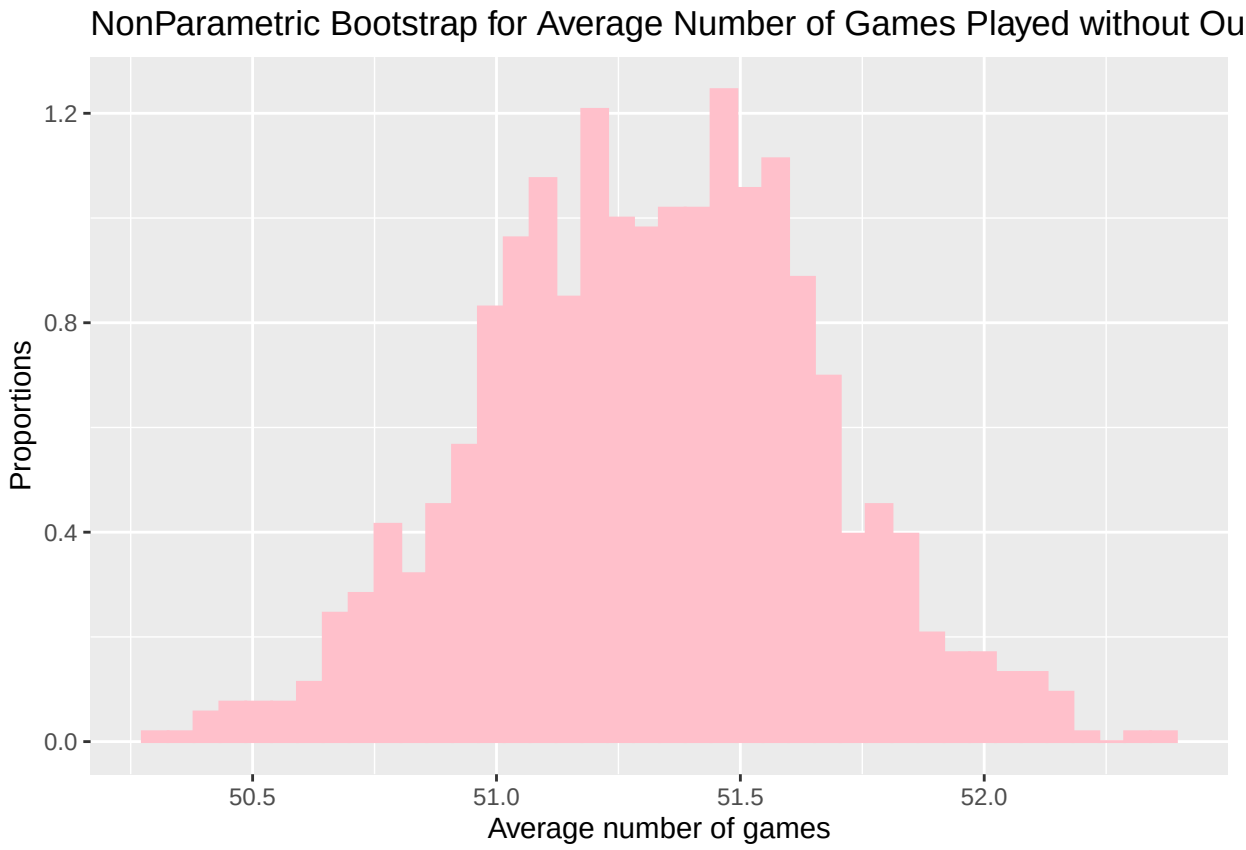



```
z = 1.98
mean_2b = mean(np_2b$b_2b)
sd_2b = sd(np_2b$b_2b)
```

```
ci_2b = c(mean_2b - z * sd_2b ,
                    mean_2a + z * sd_2b )
ci_2b
```

```
## [1] 50.55305 52.63155
```

```
game_data_f = filter(game_data, games_played < 40000)
np_2b = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2b = sample(game_data_f$games_played, length(game_data_f$games_played), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2b = mean(b_2b))
})
#plot histogram of values
ggplot(np_2b, aes(x=b_2b, y = ..density..,)) +
  geom_histogram(colour="pink", fill="pink", bins = 40) +
  xlab("Average number of games") +
  ylab("Proportions") +
  labs(title = "NonParametric Bootstrap for Average Number of Games Played without Outlier")
```



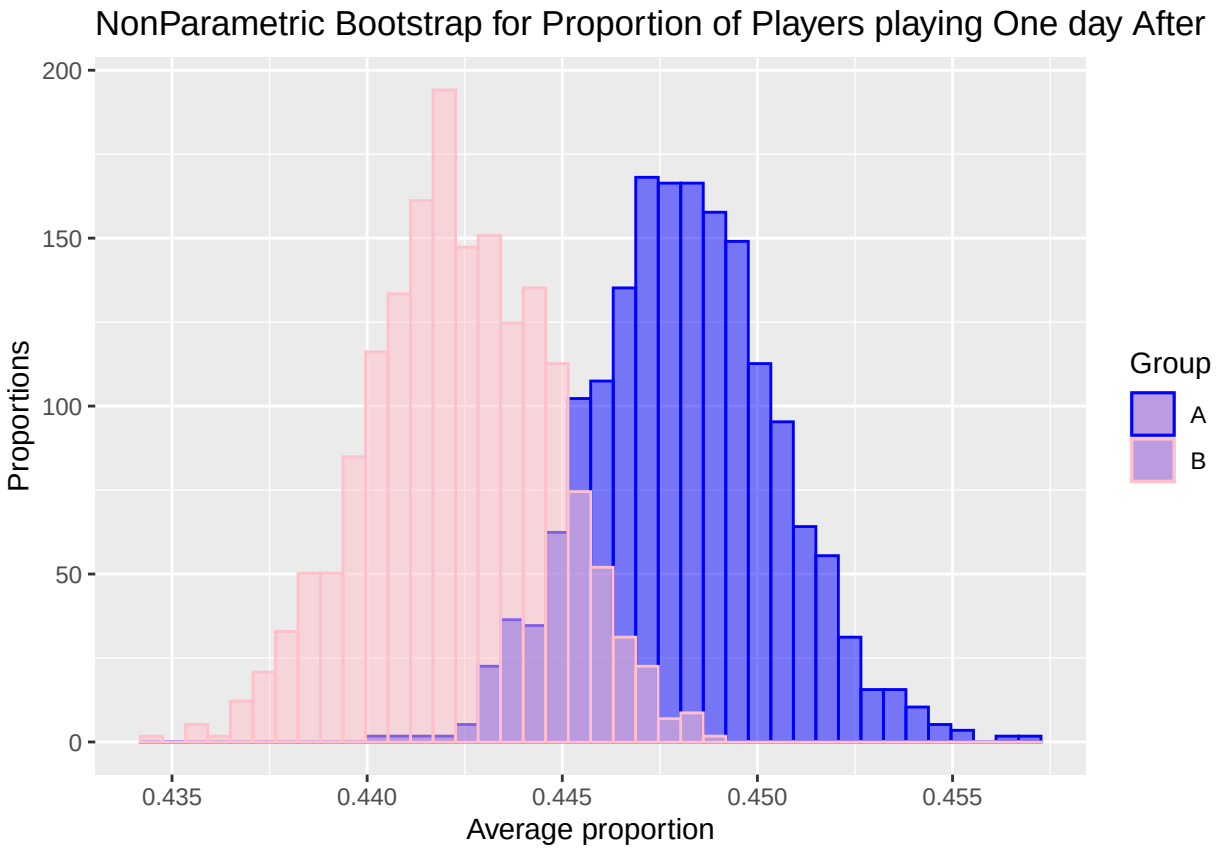

```
z = 1.98
mean_2b = mean(np_2b$b_2b)
sd_2b = sd(np_2b$b_2b)
ci_2b = c(mean_2b - z * sd_2b ,
                    mean_2a + z * sd_2b )
ci_2b
```

```
## [1] 50.64168 51.99637
```

Part c

Based on this graph, I would recommend to go with version A if we based our data off the first day retention rates, visually.

```
#reference: bootstrap_mean.R in course files
np_2c = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2c = sample(data_A$retained_1, length(data_A$retained_1), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2c = mean(b_2c))
})
np_2cb = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2cb = sample(data_B$retained_1, length(data_B$retained_1), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2cb = mean(b_2cb))
})
#plot histogram of values
ggplot(np_2c, aes(x=b_2c, y = ..density.., color = 'A')) +
  geom_histogram(fill="blue", bins = 40, alpha = .5) +
  geom_histogram(data = np_2cb, aes(x=b_2cb, color = 'B'), fill="pink", bins = 40, alpha = .5) +
  xlab("Average proportion") +
  ylab("Proportions") +
  labs(title = "NonParametric Bootstrap for Proportion of Players playing One day After") +
  scale_color_manual(name='Group',
                     breaks=c('A', 'B'),
                     values=c('A'='blue', 'B'='pink'))
```

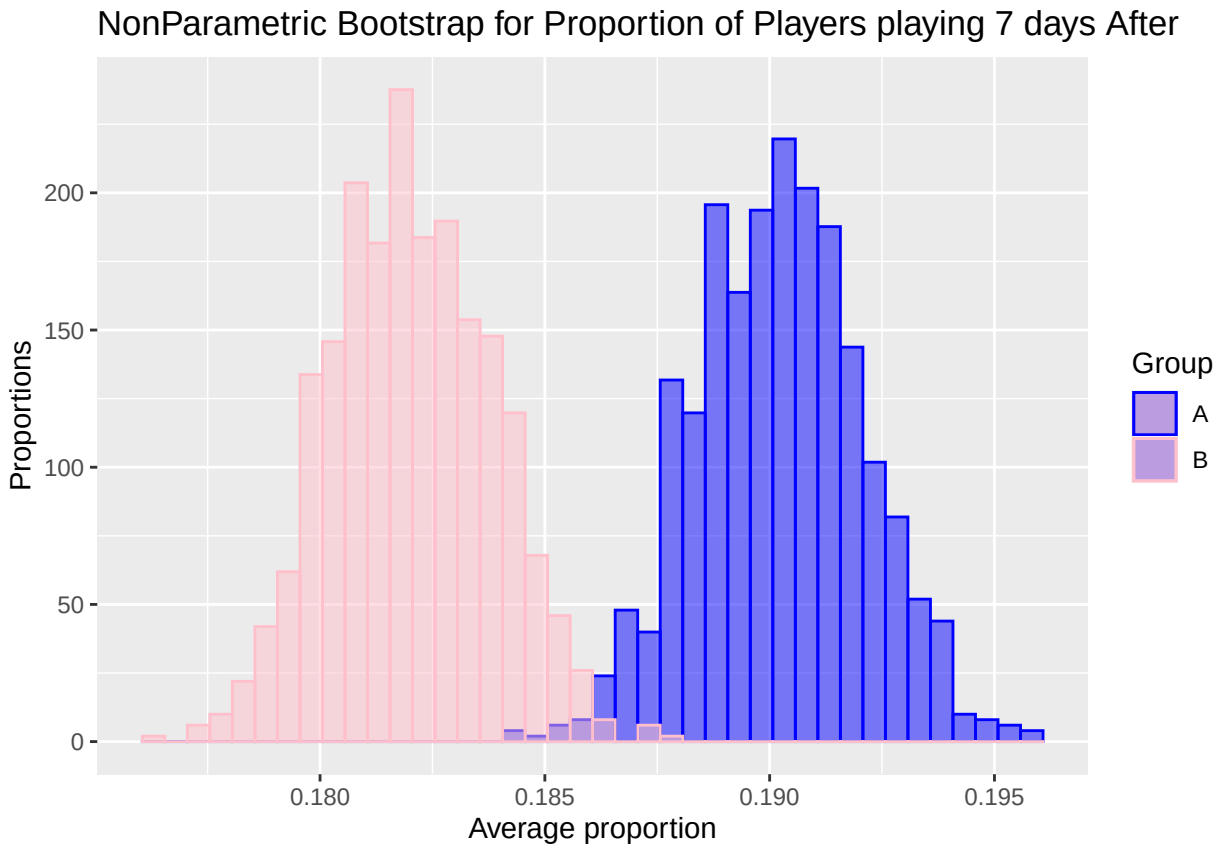


Part d

Judging off our histogram, I would reccommend version A, as it has a higher retension rate, visually.

```
#reference: bootstrap_mean.R in course files
np_2d = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2d = sample(data_A$retained_7, length(data_A$retained_7), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2d = mean(b_2d))
})
np_2db = map_df(1:1000, function(i) {
# sample from the data WITH replacement
b_2db = sample(data_B$retained_7, length(data_B$retained_7), replace = TRUE)
# compute the sample mean of the bootstrap sample
data.frame(b_2db = mean(b_2db))
})
#plot histogram of values
ggplot(np_2d, aes(x=b_2d, y = ..density.., color = 'A')) +
  geom_histogram(fill="blue", bins = 40, alpha = .5) +
  geom_histogram(data = np_2db, aes(x=b_2db, color = 'B'), fill="pink", bins = 40, alpha = .5) +
  xlab("Average proportion") +
  ylab("Proportions") +
  labs(title = "NonParametric Bootstrap for Proportion of Players playing 7 days After") +
  scale_color_manual(name='Group',
```

```
                    breaks=c('A', 'B'),
                    values=c('A'='blue', 'B'='pink'))
```



Problem 3: Hypothesis tests

For this part forward, I remove an outlier, which can durastically change our results if we didn't. ### Part a

```
game_data = game_data_f
data_A = filter(game_data, group == "A")
data_B = filter(game_data, group == "B")
#source: t_test.R in course materials
set.seed(10)
# H0: mu = 54
# H1: mu < 54

mu_0 = 54
sd = sd(game_data$games_played)
n = nrow(game_data)
x_bar = mean(game_data$games_played)

# to test that the mean is less than 54
t <- (x_bar - mu_0) / (sd / sqrt(n))
t
```

```
## [1] -7.837378
```

```
pt(t, df = n-1)
```

```
## [1] 2.325224e-15
```

```
#checking
t.test(game_data$games_played, mu = mu_0, alternative = "less")
```

```
##
##  One Sample t-test
##
## data:  game_data$games_played
## t = -7.8374, df = 90187, p-value = 2.325e-15
## alternative hypothesis: true mean is less than 54
## 95 percent confidence interval:
##      -Inf 51.88267
## sample estimates:
## mean of x
##  51.32025
```

We choose to reject the null (co-workers hypotheis) and accept the alternate hypothesis (boss's hypothesis) at a 5% level.

Part b

The average gameplay data gives us an oppurtunity to capture a possible range of values that the population may hold the true value, however due to the hypothesis test, we can see clearly that the chances of 54 being the true average and not lower, is extremely low.

Part c

I chose to use Welch's test for two Sample T-Test since our samples both have unknown standard deviation. I did not use pooled variance estimator because our samples are very large as well as no indication that the variations are the same, and thus my reasoning. The answer does remain consistent with what I observed in 1b, as the graphs are overlapping eachother fairly well and follow similar looking distributions. We continue to accept our null hypothesis, which is that the average games played between group A and B are the same at a 5% level.

```
#refernce: two_sided_test.R from course files
#H_0 mu_A = mu_B
#H_A mu_A != mu_B
#data
n <- nrow(data_A)
m <- nrow(data_B)
xbar <- mean(data_A$games_played)
xbar
```

```
## [1] 51.34211
```

```r
ybar <- mean(data_B$games_played)
ybar
```

```
## [1] 51.29878
```

```r
s1 <- sd(data_A$games_played)
s1
```

```
## [1] 102.0576
```

```r
s2 <- sd(data_B$games_played)
s2
```

```
## [1] 103.2944
```

```r
#df and t stats
df <- (s1^2/n + s2^2/m)^2 / (s1^4 / (n^2 * (n-1)) + s2^4 / (m^2 * (m-1)))
df
```

```
## [1] 90183.3
```

```r
t <- (xbar - ybar) / (sqrt(s1^2/n + s2^2 / m))
t
```

```
## [1] 0.06337426
```

```r
#pvalue
print("P-value:")
```

```
## [1] "P-value:"
```

```r
pval <- 2 * (1 - pt(t, df))
pval
```

```
## [1] 0.9494686
```

```r
t.test(data_A$games_played, data_B$games_played, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  data_A$games_played and data_B$games_played
## t = 0.063374, df = 90183, p-value = 0.9495
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.296897  1.383568
## sample estimates:
## mean of x mean of y
##  51.34211  51.29878
```

Part d

```
#H_0 p_A = p_B
#H_A p_A != p_B

p_A = mean(data_A$retained_1)
p_B = mean(data_B$retained_1)
p_hat = mean(game_data$retained_1)
n <- nrow(data_A)
m <- nrow(data_B)

z = (p_A-p_B-0)/(sqrt((p_hat*(1-p_hat))*((1/n)+(1/m))))
z
```

```
## [1] 1.787104
```

```
p_val = 2*(1-pnorm(z))
p_val
```

```
## [1] 0.07392076
```

We choose to keep the null hypothesis, which is that their proportions are the same at a 5% level.

Part e

```
#H_0 p_A = p_B
#H_A p_A != p_B

p_A = mean(data_A$retained_7)
p_B = mean(data_B$retained_7)
p_hat = mean(game_data$retained_7)
n <- nrow(data_A)
m <- nrow(data_B)

z = (p_A-p_B-0)/(sqrt((p_hat*(1-p_hat))*((1/n)+(1/m))))
z
```

```
## [1] 3.15741
```

```
p_val = 2*(1-pnorm(z))
p_val
```

```
## [1] 0.001591773
```

We reject the null and accept the alternate hypothesis at a 5% level.

Part f

I personally think that the company should stick with the original grouping, which is option A due to the retention rates being generally higher over the two week period. The longer the game is being played, the more income is made (especially if there are ads!)

Fun fact–

I did a calculation of how much time per day the outlier did... and well... check the math out ;-; hope theyre doing okay and eventually touch grass.

```
#in 2 weeks, assuming they sleep 8 hours a night and wake up to ONLY play this game:
time = max(game_data$games_played) / 14 #convert to games per days
time = time / 16 #convert to games per hours, assuming 8 are spent sleeping
time = 60 / time # convert to how many minutes per game.
time
```

```
## [1] 4.539007
```