A/B Testing of Cookies Cats: Determining the Better Level for Player Retention

Mariam Badmus

06/11/2022

Contents

1. Introduction **3**

2. About the data **3**

3. Analysing data **3**

1. Import required packages 3
2. Read the data 4
3. Look at the data, check for duplicates, blank rows and NA values 4
4. Check the numbers of players for each day 5
5. Use boxplot to identify outliers 5
6. Apply summary statistics and group by version 6
7. Remove outliers, visualise the new data without outliers 6
8. Recalculate the summary statistics and standard deviation after removing the outliers . . 7

4. A/B TEsting **7**

1. Step 1: Using histogram to determine if our data has normal distribution 7
2. Step 2: Using levene test to determine if there is homogeneity among the control and treatment groups 11
3. Step 3: Applying Mann Whitney U test because both groups are not normal distribution . 11

5. Calculating retention in day one and day 7 between gate_30 and gate 40 **12**

6. Data Insight and Conclusion **13**

1. Introduction

Cookie Cats is a hugely popular mobile puzzle game developed by Tactile Entertainment. It's a classic "connect three" style puzzle game where the player must connect tiles of the same color in order to clear the board and win the level.

As players progress through the game they will encounter gates that force them to wait some time before they can progress or make an in-app purchase. In this project, we will analyze the result of an A/B test where the first gate in Cookie Cats was moved from level 30 to level 40. In particular, we will analyze the impact on player retention and game rounds.

For this A/B hypothesis testing, level 30 (gate_30) will be the control group because the in app purchases start at gate_30,and level 40 (gate_40) will be the test group. The project goals are:

2. About the data

The data is downloaded from Kaggle, it is in csv format and organised in a long format.The data is originally from datacamp but downloaded from Kaggle. Using ROCCC (Reliable, Original, Comprehensive, Current, and Cited):

Reliability: This data is from a Datacamp project which serves as the only reference to the dataset, and access is by premium users only. there is a link of the dataset on Kaggle

Original: Without access it the data camp premium account, it is unclear if it is original dataset

Comprehensive: for the purposes of the project tasks and deliverables, the data is sufficiently comprehensive.

Current: the data has been available for the past three years on Kaggle

Cited: the citation goes back to Datacamp which remains inaccessible without a premium acount to determine if datacamp references the original data and if the data is publicly available and if it was available under any licence.

3. Analysing data

1. Import required packages

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.2.2
```

```
library(stats)
```

2. Read the data

```
cookies_cats <- read.csv("C:/Users/kike_/OneDrive/Documents/cookies_cats/cookie_cats.csv")
```

3. Look at the data, check for duplicates, blank rows and NA values

Userid: unique id for each player version: players are randomly asssinged to either gate_30 or gate_40. the number fo gate is where players are offered gate_30 is the first gate where players are offered in-app purchases. for gate_30, the first gate is at level 30, for gate_40, the fist gate is at level 40 sum_gamerounds: total sum of gamerounds for each player between day 1 and day 7 retention_1: did the player play a day after game installation? If yes, the column records True, if no, it records False retention_7: did the player play seven days after game installation? If yes, the column records True, if no, it records False

Some players might play after day 1 and day 7 (True - True) Some players might play after day 1 but not day 7 (True - False) Some players might not play on after day 1 but play by day 7 (False - True) Some players might not play after day 1 and day 7 (False - False)

```
glimpse(cookies_cats)
```

```
## Rows: 90,189
## Columns: 5
## $ userid         <int> 116, 337, 377, 483, 488, 540, 1066, 1444, 1574, 1587, 1~
## $ version        <chr> "gate_30", "gate_30", "gate_40", "gate_40", "gate_40", ~
## $ sum_gamerounds <int> 3, 38, 165, 1, 179, 187, 0, 2, 108, 153, 3, 0, 30, 39, ~
## $ retention_1    <lgl> FALSE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, FALSE, TRU~
## $ retention_7    <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, FALSE, T~
```

```
sum(duplicated(cookies_cats))
```

```
## [1] 0
```

```
sum(is.na(cookies_cats))
```

```
## [1] 0
```

4. Check the numbers of players for each day

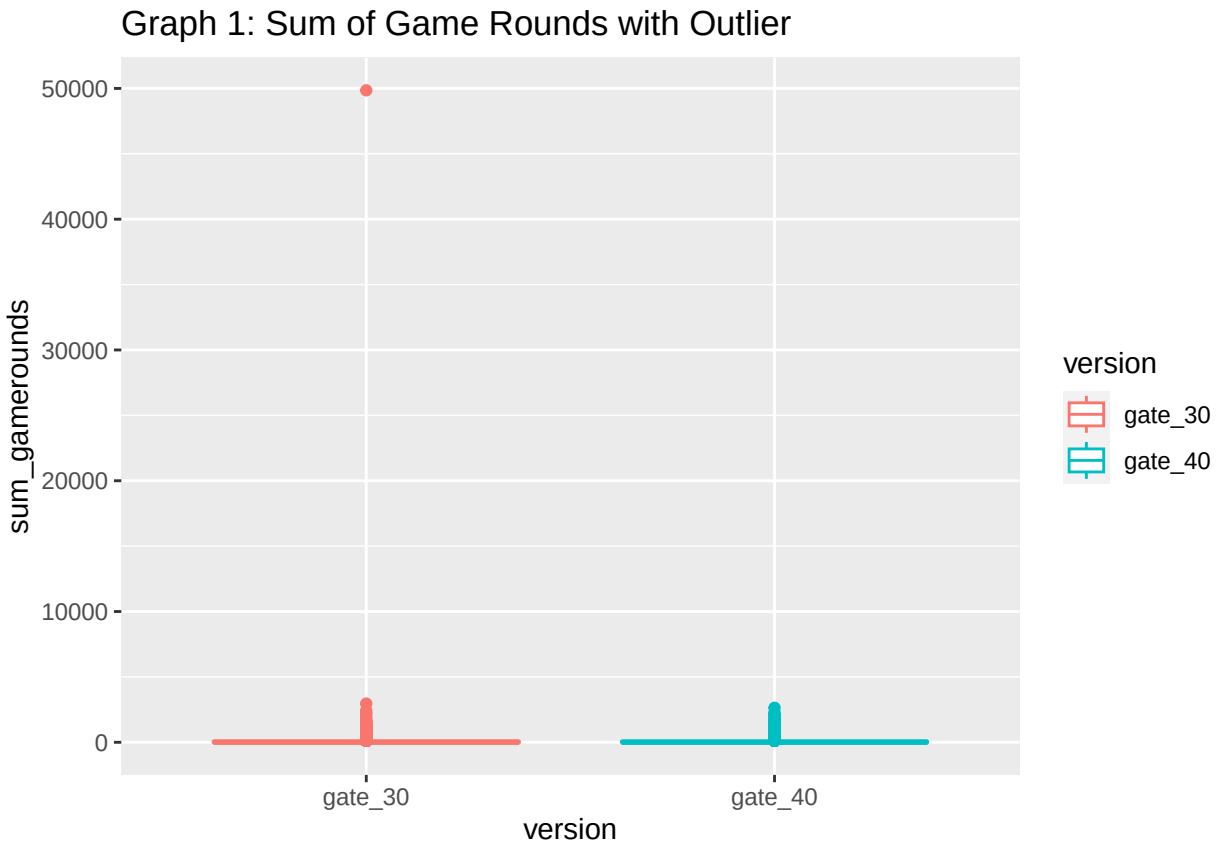
```
aggregate(cookies_cats$userid ~ cookies_cats$version, FUN = length)
```

```
##   cookies_cats$version cookies_cats$userid
## 1              gate_30               44700
## 2              gate_40               45489
```

5. Use boxplot to identify outliers

The outlier is the lone dot is closer to 50000 and it looks too far off from the other values

```
ggplot(cookies_cats, (aes(x = version, y = sum_gamerounds, color = version)))+
  geom_boxplot()+
  labs(title = "Graph 1: Sum of Game Rounds with Outlier")
```



6. Apply summary statistics and group by version

```
tapply(cookies_cats$sum_gamerounds, cookies_cats$version, summary)
```

```
## $gate_30
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00     5.00    17.00    52.46    50.00 49854.00
##
## $gate_40
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     5.0    16.0    51.3    52.0  2640.0
```

```
aggregate(cookies_cats$sum_gamerounds ~ cookies_cats$version, FUN = sd)
```

```
##   cookies_cats$version cookies_cats$sum_gamerounds
## 1              gate_30                    256.7164
## 2              gate_40                    103.2944
```

7. Remove outliers, visualise the new data without outliers

```
subset(cookies_cats, sum_gamerounds == max(sum_gamerounds))
```

```
##        userid version sum_gamerounds retention_1 retention_7
## 57703 6390605 gate_30          49854       FALSE        TRUE
```

```
cookies_cats_rv <- cookies_cats[!(row.names(cookies_cats) %in%  c(57703)),]
```

8. Recalculate the summary statistics and standard deviation after removing the outliers

```
tapply(cookies_cats_rv$sum_gamerounds, cookies_cats_rv$version, summary)
```

```
## $gate_30
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.00   17.00   51.34   50.00 2961.00
##
## $gate_40
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     5.0    16.0    51.3    52.0  2640.0
```

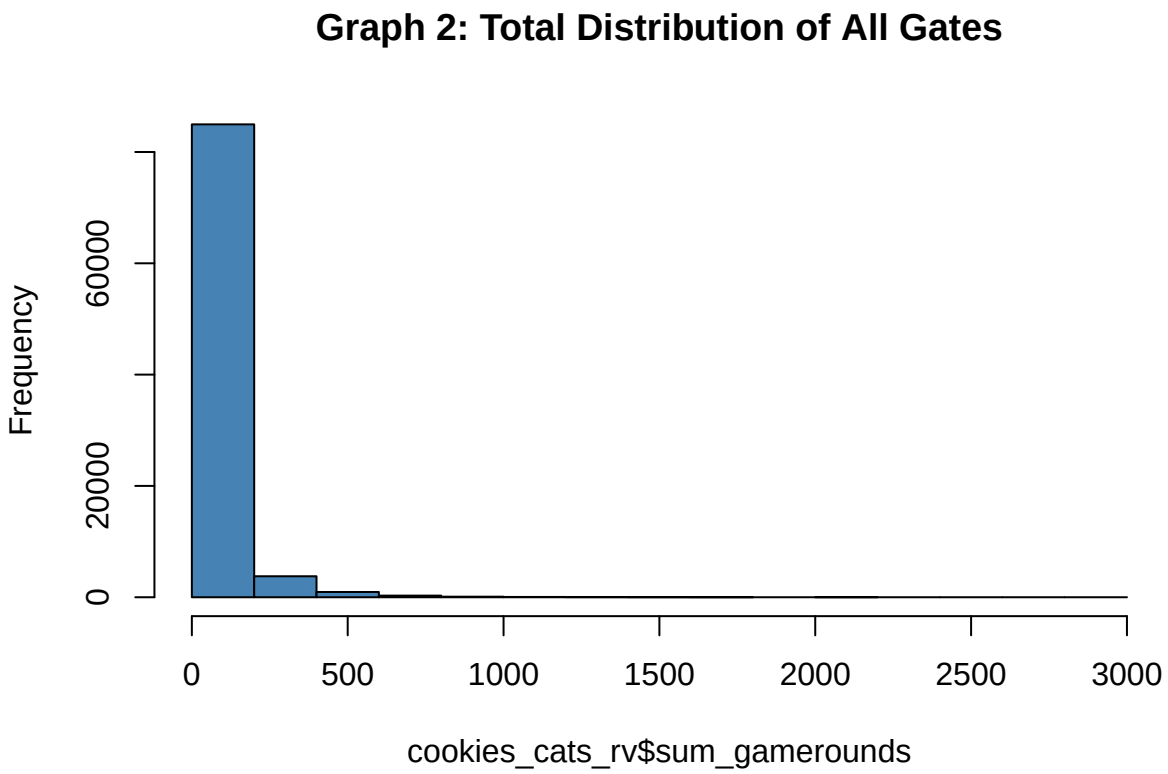
```
aggregate(cookies_cats_rv$sum_gamerounds ~ cookies_cats_rv$version, FUN = sd)
```

```
##   cookies_cats_rv$version cookies_cats_rv$sum_gamerounds
## 1                 gate_30                       102.0576
## 2                 gate_40                       103.2944
```

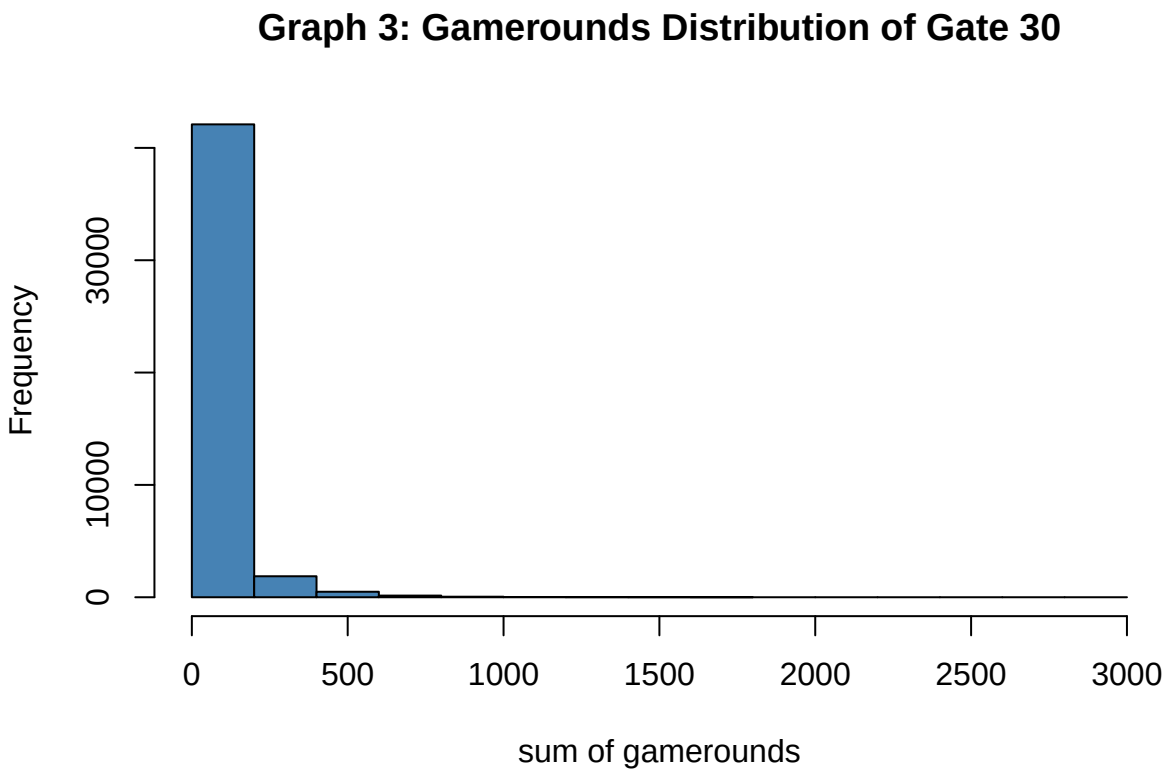
4. A/B TEsting

1. Step 1: Using histogram to determine if our data has normal distribution

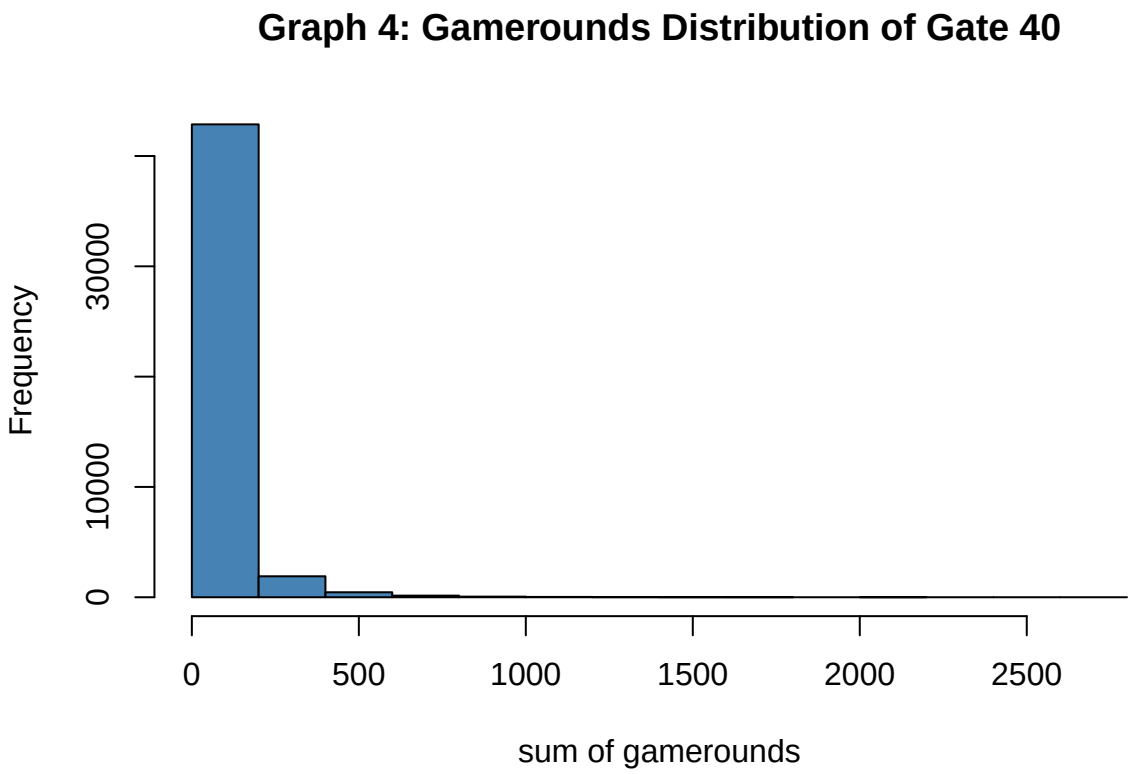
```
hist(cookies_cats_rv$sum_gamerounds, main = paste("Graph 2: Total Distribution of All Gates"), col = 's
```



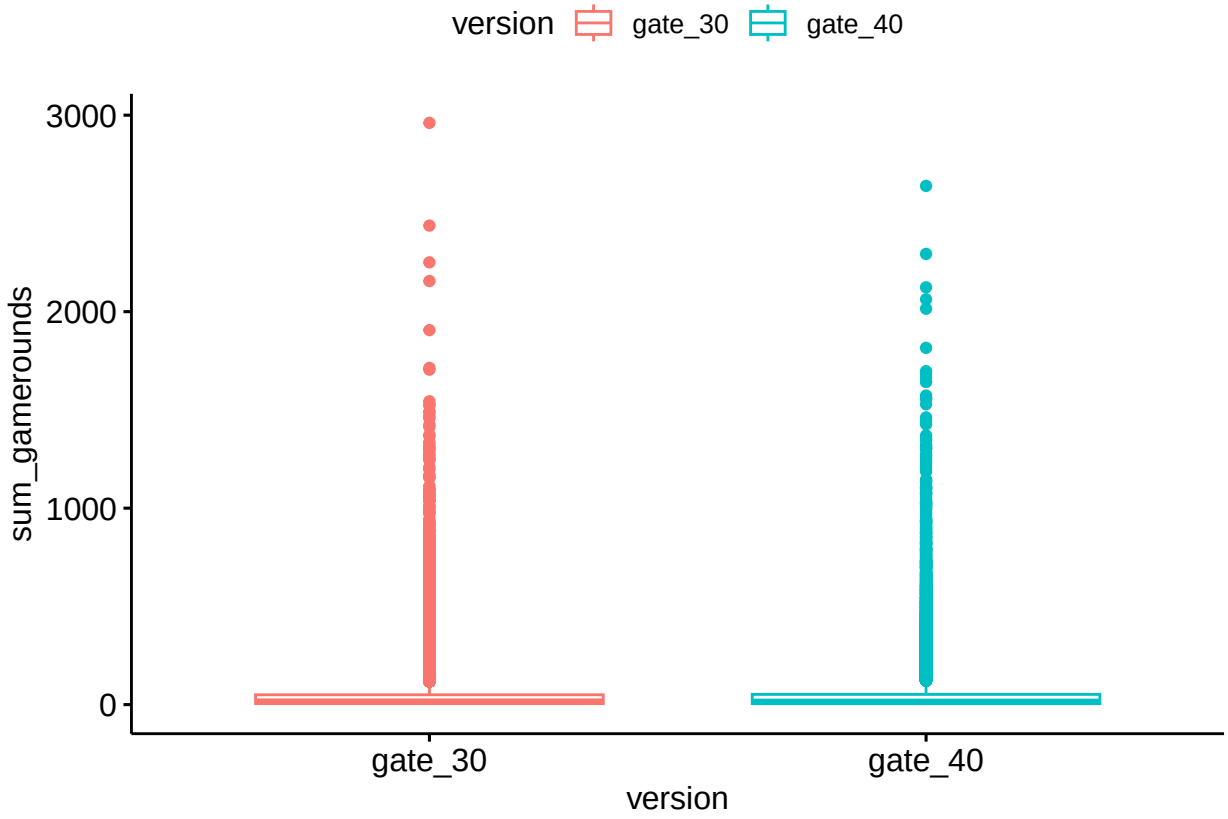

```
hist(cookies_cats_rv$sum_gamerounds[cookies_cats_rv$version == 'gate_30'],  xlab = "sum of gamerounds",
```




```
hist(cookies_cats_rv$sum_gamerounds[cookies_cats_rv$version == 'gate_40'], xlab = "sum of gamerounds", m
```




```
ggboxplot(cookies_cats_rv, x = 'version', y = 'sum_gamerounds', color = 'version')
```



2. Step 2: Using levene test to determine if there is homogeneity among the control and treatment groups

Null hypothesis = assumption of that both groups have equal variance Alternative hypothesis = assumption of that both groups do ot have equal variance

From the Levene test result below, we accept the assumption that the null hypothesis that the groups have equal variance because the p-value is greater than 0.05 at 0.784

```
leveneTest(sum_gamerounds ~ version, data = cookies_cats_rv)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     1  0.0751  0.784
##       90186
```

3. Step 3: Applying Mann Whitney U test because both groups are not normal distribution

Null hypothesis: The groups are similar Alternative hypothesis:The groups are not equal to each other

```
wilcox.test (sum_gamerounds ~ version, data = cookies_cats_rv)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sum_gamerounds by version
## W = 1024285762, p-value = 0.05089
## alternative hypothesis: true location shift is not equal to 0
```

5. Calculating retention in day one and day 7 between gate_30 and gate 40

```
cookies_cats_rv %>%
  group_by(version, retention_1, retention_7) %>%
  summarise(count = n())
```

```
## `summarise()` has grouped output by 'version', 'retention_1'. You can override
## using the `.groups` argument.
```

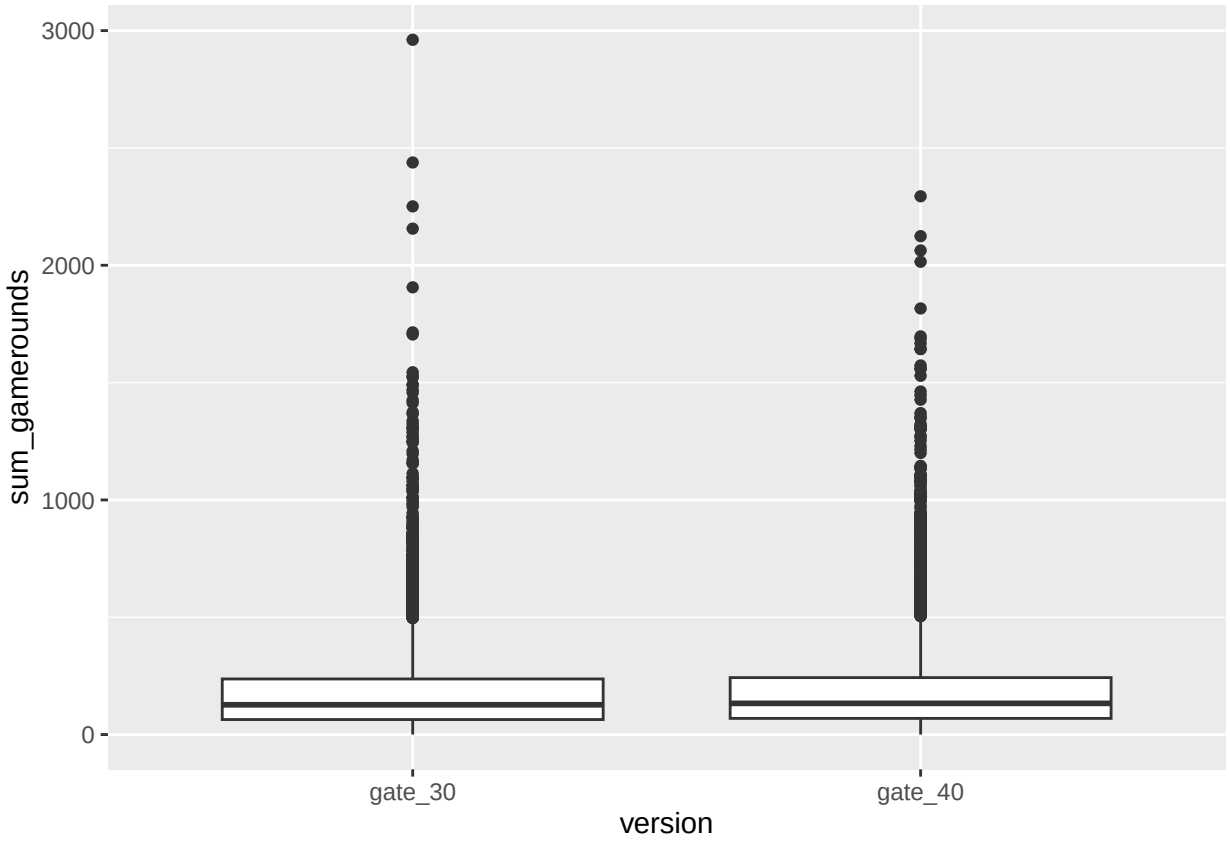
```
## # A tibble: 8 x 4
## # Groups:   version, retention_1 [4]
##   version retention_1 retention_7 count
##   <chr>   <lgl>       <lgl>       <int>
## 1 gate_30 FALSE       FALSE       22840
## 2 gate_30 FALSE       TRUE         1825
## 3 gate_30 TRUE        FALSE       13358
## 4 gate_30 TRUE        TRUE         6676
## 5 gate_40 FALSE       FALSE       23597
## 6 gate_40 FALSE       TRUE         1773
## 7 gate_40 TRUE        FALSE       13613
## 8 gate_40 TRUE        TRUE         6506
```

```
#further analysis of day 1 and day 7 if retention remains true
cookies_cats_rv %>%
  group_by(version, retention_1, retention_7) %>%
  filter(retention_1 == 'TRUE' & retention_7 == 'TRUE') %>%
  summarise(count = n(), median =  median(sum_gamerounds), mean =  mean(sum_gamerounds), sum = sum(sum_g
```

```
## `summarise()` has grouped output by 'version', 'retention_1'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 2 x 7
## # Groups:   version, retention_1 [2]
##   version retention_1 retention_7 count median  mean     sum
##   <chr>   <lgl>       <lgl>       <int>  <dbl> <dbl>   <int>
## 1 gate_30 TRUE        TRUE         6676    127  184. 1227625
## 2 gate_40 TRUE        TRUE         6506    133  190. 1237977
```

```
cookies_cats_rv %>%
  filter(retention_1 == 'TRUE' & retention_7 == 'TRUE') %>%
  ggplot(aes(x = version, y = sum_gamerounds))+
  geom_boxplot()
```



6. Data Insight and Conclusion

1. Player churn rate is very high, it is unclear if that is normal for online games
2. The sample size for gate_40 is 790 more than gate_30. yet gate_30 has higher retention rate than gate_40 for both day 1 and day 7 (True-True)
3. More gate_40 players than gate_30 players failed to return after both day 1 and day 7. A larger sample size for gate_40 does not explain these difference as it is already covered by False-False difference (23, 597 - 22, 840 = 757) (False-False)
4. Despite the higher number of players that continued playing at day 1 and day 7 (True-True), a further analysis shows that gate_40 has a larger median and mean of gamerounds than gate_30. Summing up the gamerounds reveal that gate_40 players played more game rounds than gate_30 which affected the median and the mean.
5. Visualising gamerounds of players that continued playing at day 1 and day 7 (True-True) shows that gate_30 has more extreme sum_gamerounds than gate_40 however they were into enough to affect the mean, median and sumtotal of the sum_gamerounds

6. Where day 1 and day 7 remains true for gate_30 and gate_40, the median is 127 and 133 respectively and the mean is 183.8 and 190 respectively.

7. Application of the Tactile Entertainment metrics will determine which is more important, whether more players at gate_30 or increased gamerounds at gate_40 or a balance of players and increased gamerounds is best effective for retaining players.

8. Grouping the number of players that played around median, level 30 and 40, about half of the total players amass sum_gamerounds of 17 or less (16-17 is the median for gate_30 and gate_40). About a third of the total players played 30 and above sum_gamerounds for both gate groups, and the number of players declined at 40 and above sum_gamerounds.

9. The sampling size is greater than 30 and it does not have a normal distribution. The distribution is a right skewed distribution. It shows that most players in this data have play few gamerounds with the median falling between 16 and 17 for gate_30 and gate_40 and the mean which is more susceptible to extreme values falling between 51.3 for both gates. The groups' have similar large standard deviation at 102 and 103 which means the values of sum_gamerounds are well spread out from the mean at 51.3.

10. Comparing the groups using Mann Whitney U test, The p-value is exactly 0.05 which means that the the groups are similar therefore we can't reject our Null hypothesis. There is no motivation from this data for Tactile Entertainment to move the first gate from gate 30 to gate 40.