GameData Analysis (Kevin Ayala)

Kevin Ayala

3/9/2020

Read in csv. Mini EDA

```
getwd()
```

```
## [1] "/Users/kevinlorenzoayala/Downloads"
```

```
data <- read.csv("/Users/kevinlorenzoayala/downloads/testData (1).csv")
head(data)
```

```
##   GameID SessionDate       UserID                             SessionID
## 1 game_1     1/11/20 57A6F9B13B38 8F865C2D-4C12-4C82-BA63-FAF9542AA45B
## 2 game_1     1/18/20 92BA77C32443 2BD56E28-C8E1-4213-9510-7C1D6E6518C3
## 3 game_1     1/18/20 92BA77C32443 65E2D7E2-8C93-4BEB-BCA9-7E6113C2020A
## 4 game_1     1/18/20 92BA77C32443 444785EF-3267-4E2B-8E19-0430A9A265CE
## 5 game_1     1/19/20 3889FF1BF3FC 4886E62F-5CF8-4279-9464-3A21D2FFF9B9
## 6 game_1     1/19/20 3889FF1BF3FC BF711EFB-516C-41D2-94DE-19B9A3870CF1
##   FirstSessionDate
## 1          1/11/20
## 2          1/18/20
## 3          1/18/20
## 4          1/18/20
## 5          1/19/20
## 6          1/19/20
```

```
#Checking out the data

GameID_count <- data %>% distinct(GameID) %>% count()
GameID_count   #there are a total of 7 different games in the data set
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     7
```

```
First_session_ofgame <- data %>%
  mutate(FirstSessionDate = as.Date(FirstSessionDate, '%m/%d/%y')) %>%
  group_by(GameID) %>%
  summarise(FirstSessionDate = min(FirstSessionDate))

First_session_ofgame #has the first day of game having a user.
```

```
## # A tibble: 7 x 2
##   GameID FirstSessionDate
##   <fct>  <date>
## 1 game_1 2020-01-08
## 2 game_2 2020-01-09
## 3 game_3 2020-01-16
## 4 game_4 2020-01-04
## 5 game_5 2020-01-19
## 6 game_6 2020-01-22
```

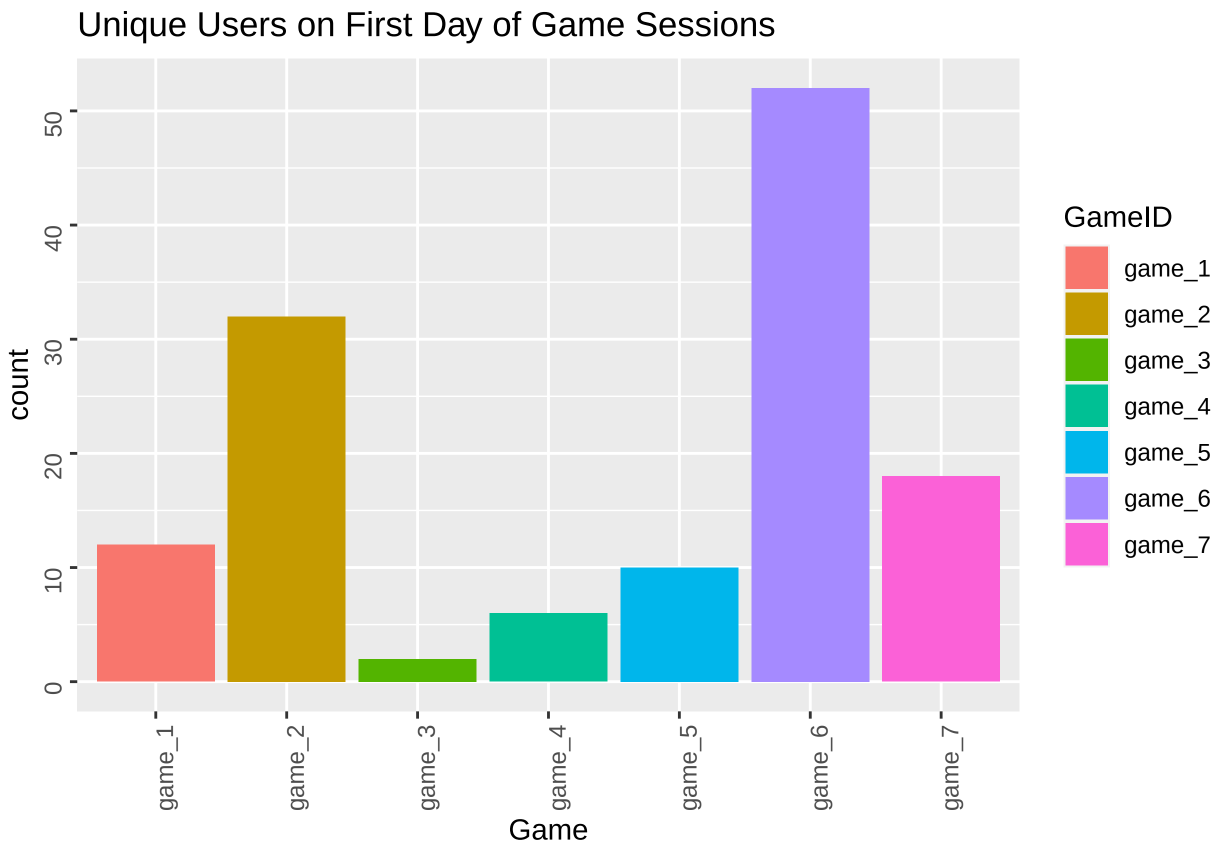
```
## 7 game_7 2019-12-23
```

```r
First_session_ofgame <-First_session_ofgame %>% mutate(NboGamers = NA)

x <- data %>%
  mutate(FirstSessionDate = as.Date(FirstSessionDate, '%m/%d/%y'))

for (i in 1:7){
First_session_ofgame$NboGamers[i] = x %>%
  filter(FirstSessionDate == First_session_ofgame$FirstSessionDate[i]) %>%
  filter(GameID == First_session_ofgame$GameID[i]) %>%
  distinct(UserID) %>%
  count()

z <- as.numeric(unlist(First_session_ofgame$NboGamers))
First_session_ofgame$NboGamers <- z
}
```

```r
First_session_ofgame
```

```
## # A tibble: 7 x 3
##   GameID FirstSessionDate NboGamers
##   <fct>  <date>               <dbl>
## 1 game_1 2020-01-08             12
## 2 game_2 2020-01-09             32
## 3 game_3 2020-01-16              2
## 4 game_4 2020-01-04              6
## 5 game_5 2020-01-19             10
## 6 game_6 2020-01-22             52
## 7 game_7 2019-12-23             18
```

```r
ggplot(aes(x=GameID),data=First_session_ofgame)+xlab("Game")+
theme(axis.text=element_text(angle=90))+geom_bar(aes(weight=NboGamers,fill=GameID))+
ggtitle("Unique Users on First Day of Game Sessions")
```




```
#Game 6 had the strongest number of users on the First Day of user appearing
```

Making an assumption that the data was put in correctly, so that the date 1/8/20 is 1/08/20 and not 1/18/20. And that 20 is 2020

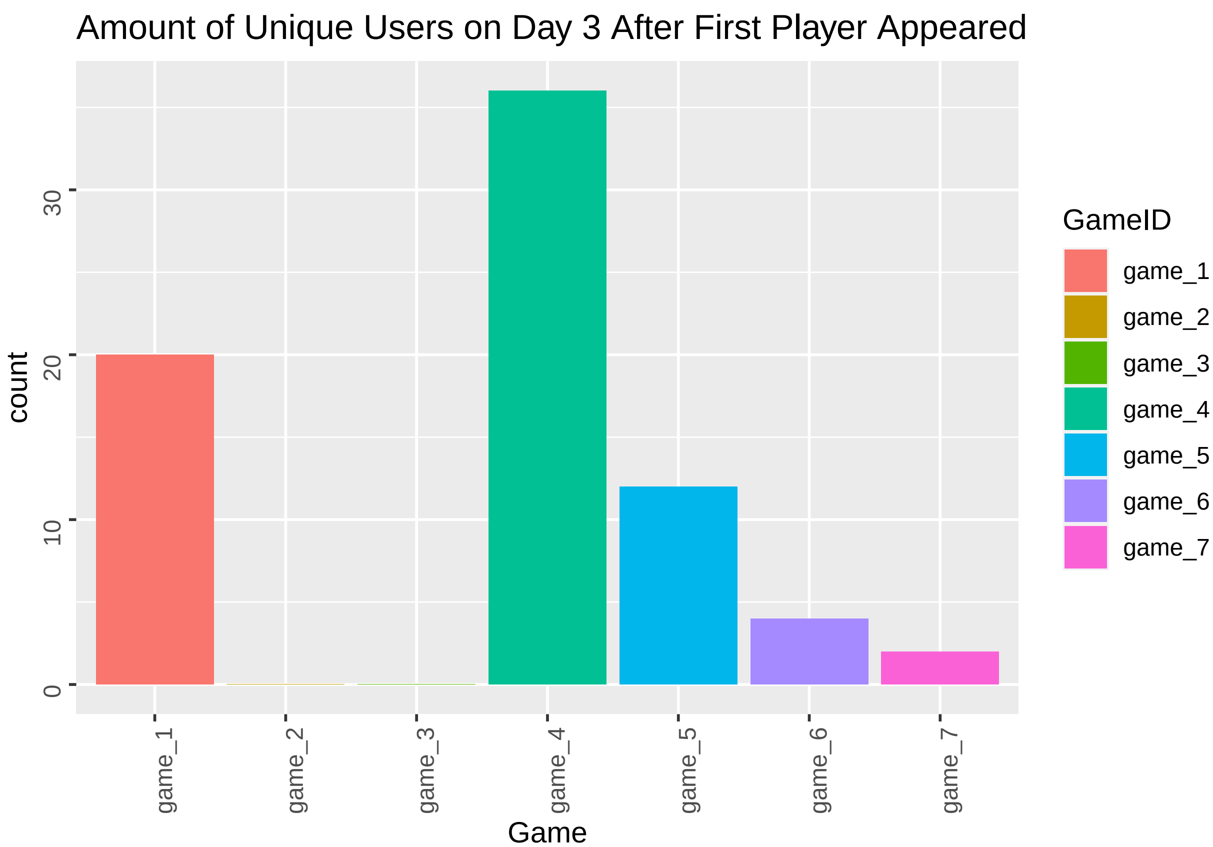
The table shows the earliest date of when a user first appeared in game. As well as the number of unique players. The visualization is a representation of the previous table.

```
First_session_ofgame <- First_session_ofgame %>%
  mutate(ThirdSessionDate = FirstSessionDate + 3)

First_session_ofgame <-First_session_ofgame %>%
  mutate(NboGamers_3 = NA)

for (i in 1:7){
First_session_ofgame$NboGamers_3[i] = x %>%
  filter(FirstSessionDate == First_session_ofgame$ThirdSessionDate[i]) %>%
  filter(GameID == First_session_ofgame$GameID[i]) %>%
  distinct(UserID) %>%
  count()

v <- as.numeric(unlist(First_session_ofgame$NboGamers_3))
First_session_ofgame$NboGamers_3 <- v
}

First_session_ofgame
```

```
## # A tibble: 7 x 5
##   GameID FirstSessionDate NboGamers ThirdSessionDate NboGamers_3
```

```
##   <fct>  <date>               <dbl> <date>                <dbl>
## 1 game_1 2020-01-08              12 2020-01-11               20
## 2 game_2 2020-01-09              32 2020-01-12                0
## 3 game_3 2020-01-16               2 2020-01-19                0
## 4 game_4 2020-01-04               6 2020-01-07               36
## 5 game_5 2020-01-19              10 2020-01-22               12
## 6 game_6 2020-01-22              52 2020-01-25                4
## 7 game_7 2019-12-23              18 2019-12-26                2
```

```r
ggplot(aes(x=GameID),data=First_session_ofgame)+xlab("Game")+
theme(axis.text=element_text(angle=90))+geom_bar(aes(weight=NboGamers_3,fill=GameID))+
ggtitle("Amount of Unique Users on Day 3 After First Player Appeared")
```



Question 1) Which Game has the best Day 1, Day 3 Retention respectively based on the data?

First will be using classic or N day retention. Decided this after reading this article here: https://amplitude.com/blog/n-day-retention-for-mobile-games

I assume that Day 1 is the date a user first appeared in the game. Hence why Day 1 = First Session Date And assume that Day 3 is 2 days after Day 1. And not 3 days after Day 1

```r
x <- x %>%
  mutate(SessionDate = as.Date(SessionDate, '%m/%d/%y'))
#converting date factor, to date format

x <- x %>%
  mutate(Day_1 = FirstSessionDate, Day_3 = FirstSessionDate + 2)
#adding Day 1, and Day 3 to each User from the time they first appeared in the game.
```

```r
day1_data <- x %>% rowwise() %>%
  mutate(match_Day1 = ifelse(between(SessionDate, FirstSessionDate, FirstSessionDate), 1, 0))

day1_return <- day1_data %>%
  group_by(GameID, UserID) %>%
  count(match_Day1)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```r
#filter out users who has count of 2 or more, a count of 1 indicates
#the first time a user first appeared. A count of 2 or indicates that
# a user returned to the app after first appearance.
day1.retention <- day1_return %>% filter(n >= 2, match_Day1 == "1") %>%
  distinct(UserID)

day1.retention <- day1.retention %>% group_by(GameID) %>% count()
colnames(day1.retention)[2] <- "active_users.day1"
day1.retention
```

```
## # A tibble: 7 x 2
## # Groups:   GameID [7]
##   GameID active_users.day1
##   <fct>              <int>
## 1 game_1               126
## 2 game_2                18
## 3 game_3               113
## 4 game_4               184
## 5 game_5                31
## 6 game_6                29
## 7 game_7                16
```

Total number of unique users who returned to the game at least once again on the same day (Day1) after their first log in/play session.

```r
Total_users <- day1_return %>%
  group_by(GameID) %>% distinct(UserID) %>% count()

colnames(Total_users)[2] <- "total_users"
Total_users
```

```
## # A tibble: 7 x 2
## # Groups:   GameID [7]
##   GameID total_users
##   <fct>        <int>
## 1 game_1         212
## 2 game_2          32
## 3 game_3         326
## 4 game_4         524
## 5 game_5         120
## 6 game_6         136
## 7 game_7         104
```

```r
#total users per game.
```

```r
day1.retention <- left_join(day1.retention, Total_users)
```

```
## Joining, by = "GameID"
```

```
day1.retention <- day1.retention %>%
  mutate(day1_retentionrate = active_users.day1/total_users)

day1.retention
```

```
## # A tibble: 7 x 4
## # Groups:   GameID [7]
##   GameID active_users.day1 total_users day1_retentionrate
##   <fct>              <int>       <int>              <dbl>
## 1 game_1               126         212              0.594
## 2 game_2                18          32              0.562
## 3 game_3               113         326              0.347
## 4 game_4               184         524              0.351
## 5 game_5                31         120              0.258
## 6 game_6                29         136              0.213
## 7 game_7                16         104              0.154
```

```
day1.retention %>%
  filter(day1_retentionrate == max(day1.retention$day1_retentionrate))
```

```
## # A tibble: 1 x 4
## # Groups:   GameID [1]
##   GameID active_users.day1 total_users day1_retentionrate
##   <fct>              <int>       <int>              <dbl>
## 1 game_1               126         212              0.594
```

The game with the highest day 1 retention rate is game 1 relative to its user population. However, when looking in the table above, game 4 has the most returned users but with lower retention rate. To get percentages we can multiply by 100.

Thus 59.4% is the highest retention rate for Day 1. Will keep future retention rates in decimal form.

```
day3_data <- x %>% rowwise() %>%
  mutate(match_Day3 = ifelse(between(SessionDate, Day_3, Day_3), 1, 0))
#adding counter, 1 if user played a session exactly 3 days after install.
#keeping between() function as its useful for in the future to
#check retention between specified dates.

day3_return <- day3_data %>%
  group_by(GameID, UserID) %>%   #grouping by GameID, and User ID
  count(match_Day3)              #counting times a user appeared in a game
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
day3.retention <- day3_return %>%
  filter(match_Day3 == '1', n >= 1) %>%
  distinct(UserID)
#filter out users who has a count of 1 and appeared on Day3

day3.retention <- day3.retention %>%
  group_by(GameID) %>% count()  #getting count users of users on Day 3 per game

colnames(day3.retention)[2] <- "active_users.day3"
day3.retention
```

```
## # A tibble: 6 x 2
## # Groups:   GameID [6]
```

```
##   GameID active_users.day3
##   <fct>                <int>
## 1 game_1                  13
## 2 game_3                  20
## 3 game_4                  45
## 4 game_5                   8
## 5 game_6                   9
## 6 game_7                   2
```

```
Total_users <- day1_return %>%
  group_by(GameID) %>%
  distinct(UserID) %>% count()

colnames(Total_users)[2] <- "total_users"
Total_users #total users per game
```

```
## # A tibble: 7 x 2
## # Groups:   GameID [7]
##   GameID total_users
##   <fct>        <int>
## 1 game_1         212
## 2 game_2          32
## 3 game_3         326
## 4 game_4         524
## 5 game_5         120
## 6 game_6         136
## 7 game_7         104
```

```
day3.retention <- left_join(day3.retention, Total_users)
```

```
## Joining, by = "GameID"
```

```
day3.retention <- day3.retention %>%
  mutate(day3_retentionrate = active_users.day3/total_users) #classic retention rate

day3.retention
```

```
## # A tibble: 6 x 4
## # Groups:   GameID [6]
##   GameID active_users.day3 total_users day3_retentionrate
##   <fct>                <int>       <int>              <dbl>
## 1 game_1                  13         212             0.0613
## 2 game_3                  20         326             0.0613
## 3 game_4                  45         524             0.0859
## 4 game_5                   8         120             0.0667
## 5 game_6                   9         136             0.0662
## 6 game_7                   2         104             0.0192
```

```
day3.retention %>% filter(day3_retentionrate == max(day3.retention$day3_retentionrate))
```

```
## # A tibble: 1 x 4
## # Groups:   GameID [1]
##   GameID active_users.day3 total_users day3_retentionrate
##   <fct>                <int>       <int>              <dbl>
## 1 game_4                  45         524             0.0859
```

The game with the highest retention rate on Day 3 is game 4. Retention rate here meaning that Game 4 had the highest rate of users returning to the game on exactly the third day after first appearing in the game. Game 4 also had the highest number of users returning on the 3rd day, this is due to game 4 having a bigger population of players.

Short answer to question 1. Game 1 had biggest retention rate on Day 1 relative to its population. Game 4 also had the biggest retention rate relative to its population on Day 3.

2) For the Game which has the highest number of users, which cohort of users based on date had the best Day 1 and Day 3 overall?

The game with the highest number of Users

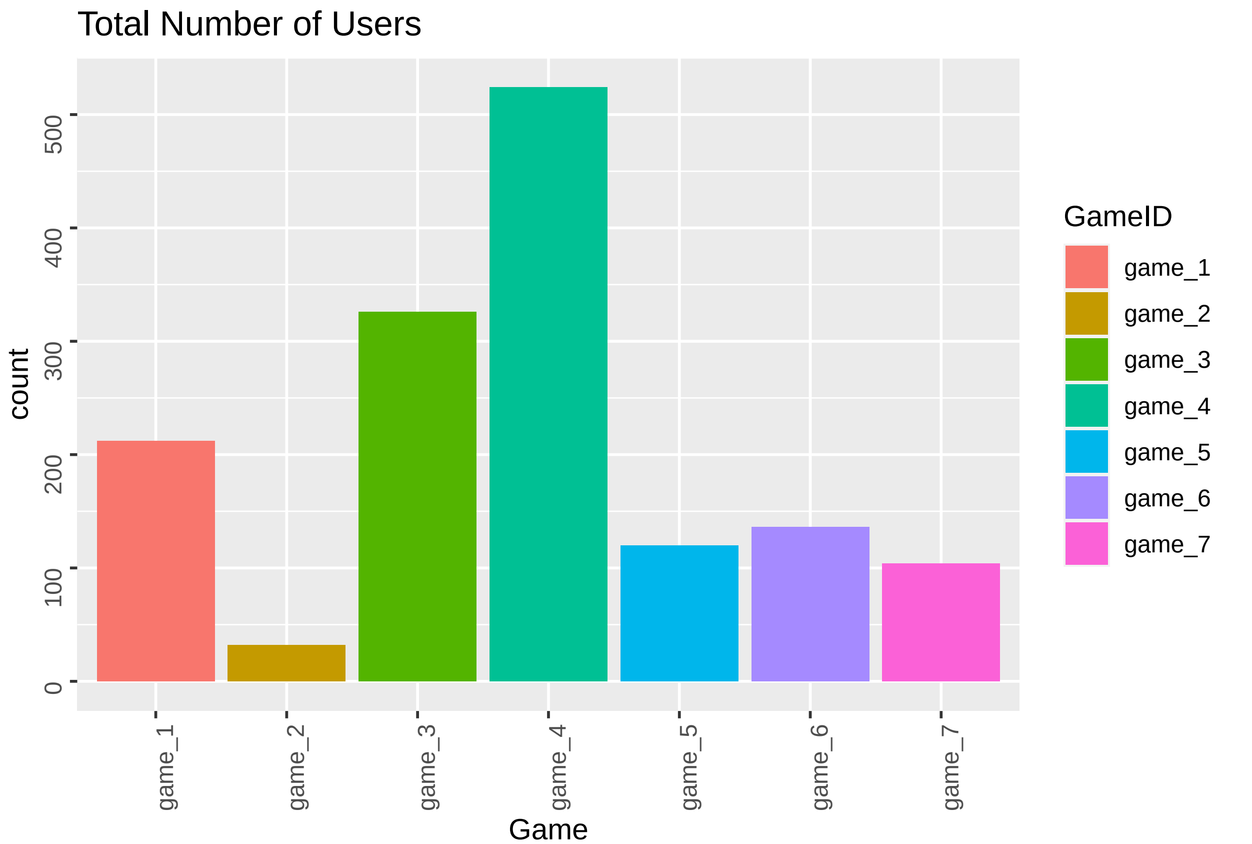
```
highest.user_game <- x %>% group_by(GameID) %>%
  distinct(UserID) %>%
  count() #distinct to get unique values

highest.user_game <- as.data.frame(highest.user_game)
most.activegame <- highest.user_game %>% filter(n == max(n))
#selecting max of unique users, game 4 has the most all time users
most.activegame
```

```
##   GameID   n
## 1 game_4 524
```

There are 524 unique Users in Game 4, the highest.

Number of Users Per Game, Visualized

```
ggplot(aes(x=GameID),data=highest.user_game)+xlab("Game")+
theme(axis.text=element_text(angle=90))+geom_bar(aes(weight=n,fill=GameID))+
ggtitle("Total Number of Users")
```



```r
#confirms table from Part 1
```

Vizualization of unique users per the lifetime of the game, according to the data

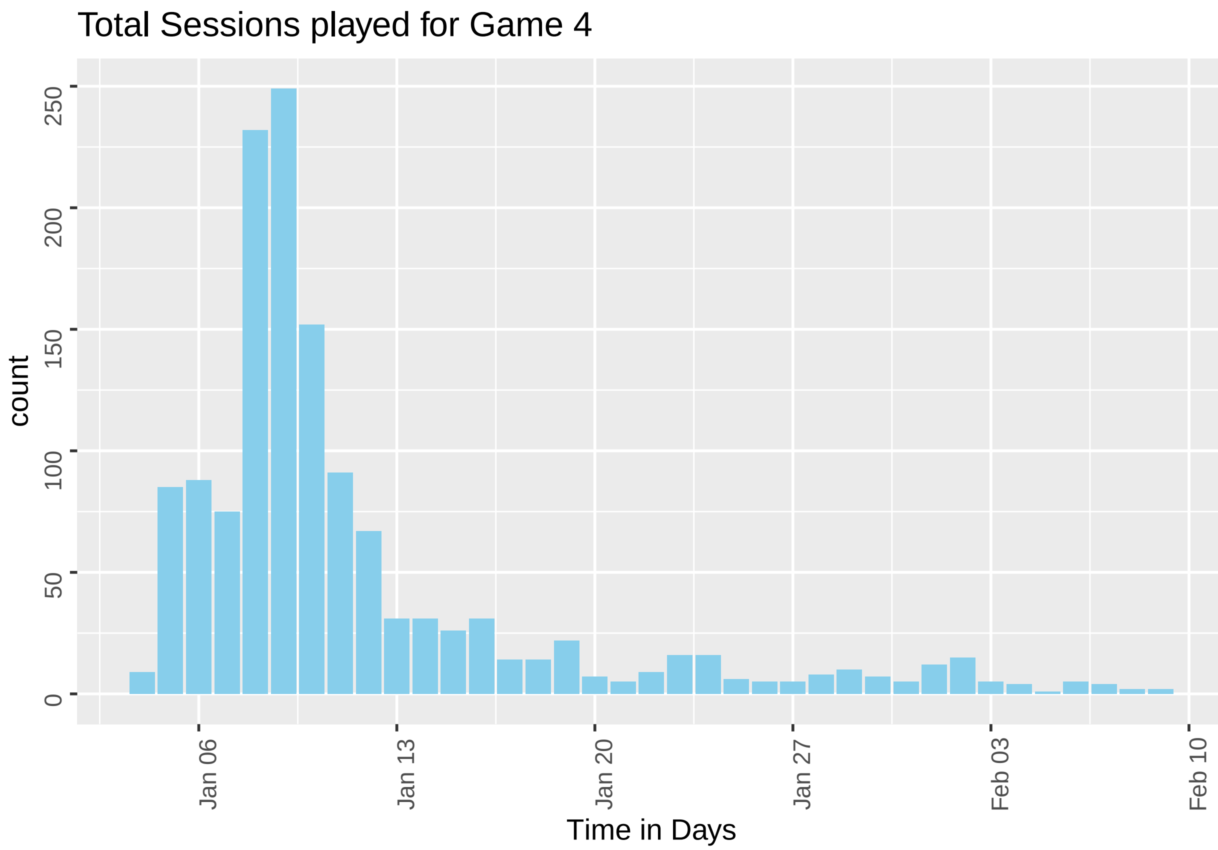
```r
game_4 <- x %>% filter(GameID == 'game_4')
head(game_4)  #subset of Data, game 4
```

```
##   GameID SessionDate       UserID                            SessionID
## 1 game_4  2020-01-10 6354D71B7006 31412F1B-AA97-4695-801E-FC69D3943DF9
## 2 game_4  2020-01-10 6354D71B7006 9066FDAC-8A1F-45BD-8658-72B562086579
## 3 game_4  2020-01-10 6354D71B7006 27D25AFC-6EFB-4C13-9649-554C99BC590A
## 4 game_4  2020-01-10 6354D71B7006 38A2AE12-FF06-4A64-9B8A-C76D2DD9C8B8
## 5 game_4  2020-01-10 6354D71B7006 F9779D8C-6DF9-4787-A5B0-0ED77362750E
## 6 game_4  2020-01-10 6354D71B7006 0D1931AD-AB47-4CC4-81C3-3694E5372641
##   FirstSessionDate      Day_1      Day_3
## 1       2020-01-10 2020-01-10 2020-01-12
## 2       2020-01-10 2020-01-10 2020-01-12
## 3       2020-01-10 2020-01-10 2020-01-12
## 4       2020-01-10 2020-01-10 2020-01-12
## 5       2020-01-10 2020-01-10 2020-01-12
## 6       2020-01-10 2020-01-10 2020-01-12
```

```r
#adding counter, when user was active/in session on Day 1 and if played game on Day 3.
#1 if true
game_4 <- game_4 %>% rowwise() %>%
  mutate(match_Day1 = ifelse(between(SessionDate, Day_1, Day_1), 1, 0))
game_4 <- game_4 %>%
  rowwise() %>% mutate(match_Day3 = ifelse(between(SessionDate, Day_3, Day_3), 1, 0))
head(game_4)
```

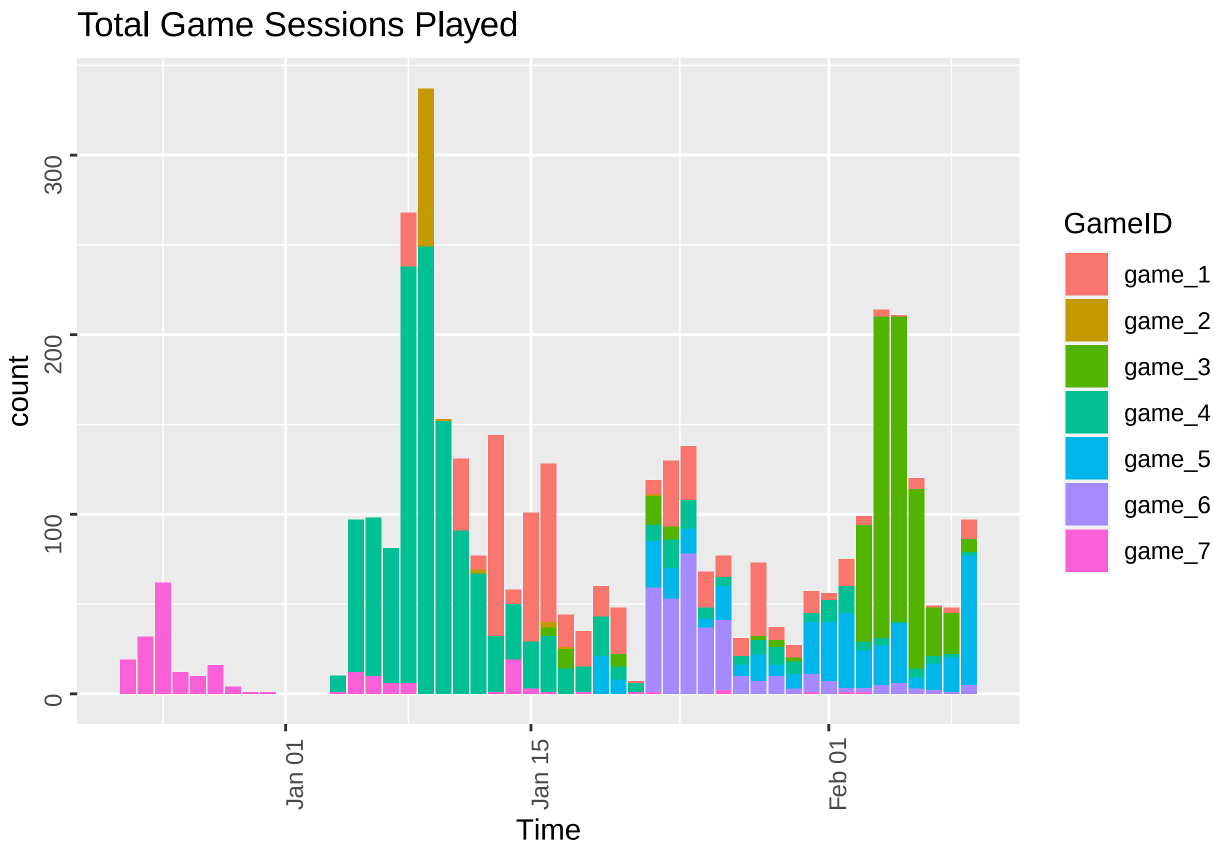
```
## Source: local data frame [6 x 9]
## Groups: <by row>
##
## # A tibble: 6 x 9
##   GameID SessionDate UserID SessionID FirstSessionDate Day_1
##   <fct>  <date>      <fct>  <fct>     <date>           <date>
## 1 game_4 2020-01-10  6354D~ 31412F1B~ 2020-01-10       2020-01-10
## 2 game_4 2020-01-10  6354D~ 9066FDAC~ 2020-01-10       2020-01-10
## 3 game_4 2020-01-10  6354D~ 27D25AFC~ 2020-01-10       2020-01-10
## 4 game_4 2020-01-10  6354D~ 38A2AE12~ 2020-01-10       2020-01-10
## 5 game_4 2020-01-10  6354D~ F9779D8C~ 2020-01-10       2020-01-10
## 6 game_4 2020-01-10  6354D~ 0D1931AD~ 2020-01-10       2020-01-10
## # ... with 3 more variables: Day_3 <date>, match_Day1 <dbl>,
## #   match_Day3 <dbl>
```

```r
ggplot(aes(x=SessionDate),data = game_4)+xlab("Time in Days")+
theme(axis.text=element_text(angle=90))+
  geom_bar(aes(),fill = "skyBlue") +
  ggtitle("Total Sessions played for Game 4")
```



This plot shows the total game 4 sessions played per day. The graph represents a total of new users and old users per day (up until that date) and how many total game sessions happened on that day. Assuming more total sessions played indicates there are more players.

```
ggplot(aes(x=SessionDate, group = GameID),data = x)+xlab("Time")+
theme(axis.text=element_text(angle=90))+
  geom_bar(aes(fill=GameID))+
  ggtitle("Total Game Sessions Played")
```



This graph shows the total sessions played per game per day. January 9, 2020 seems to be the most active day to be gaming in either Game 2 or game 4. It is interesting to note that game 2 was played mostly during January 9, 2020. Important to note that counts per game are added onto each other.

```
#Counting Unique Users who got on the game again on Day 1
day_1_return.users <- game_4 %>%
  group_by(FirstSessionDate, UserID) %>%
  count(match_Day1) %>%
  filter(match_Day1 == '1', n >= 2)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
#we filter with n >= 2 because its Day 1.


#Grouping and counting unique UserID per FirstSessionDate in final output
day_1_return.users <- day_1_return.users %>%
  group_by(FirstSessionDate) %>%
  count()


day_1_return.users <- as.data.frame(day_1_return.users)
colnames(day_1_return.users)[1] <- "Cohort"
colnames(day_1_return.users)[2] <- "frequency"  #renaming n

day_1_return.users %>% filter(frequency == max(frequency))
```

```
##       Cohort frequency
## 1 2020-01-09        57
```

```r
#getting max number of new users/player per date.

head(day_1_return.users)
```

```
##       Cohort frequency
## 1 2020-01-04         1
## 2 2020-01-05        19
## 3 2020-01-06        12
## 4 2020-01-07        12
## 5 2020-01-08        44
## 6 2020-01-09        57
```

The highest number of new users who returned to game 4 on day 1 is 57 on January 09, 2020

```r
#Counting Unique Users who got on the game again on Day 3
day_3_return.users <- game_4 %>%
  group_by(FirstSessionDate, UserID) %>%
  count(match_Day3) %>%
  filter(match_Day3 == '1', n >= 1)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```r
day_3_return.users <- day_3_return.users %>%
  group_by(FirstSessionDate) %>%
  count()

day_3_return.users <- as.data.frame(day_3_return.users)
colnames(day_3_return.users)[1] <- "Cohort"
colnames(day_3_return.users)[2] <- "frequency"
day_3_return.users %>% filter(frequency == max(frequency))
```

```
##       Cohort frequency
## 1 2020-01-10        10
```

```r
day_3_return.users
```

```
##        Cohort frequency
## 1  2020-01-04         1
## 2  2020-01-05         4
## 3  2020-01-06         6
## 4  2020-01-07         1
## 5  2020-01-08         9
## 6  2020-01-09         9
## 7  2020-01-10        10
## 8  2020-01-11         1
## 9  2020-01-12         1
## 10 2020-01-14         1
## 11 2020-01-16         1
## 12 2020-01-23         1
```

The highest number of players/users that returned to the game at exactly 3 days after they first appeared in the game is 10. This is a small number due to dataset size.

As a gamer, it is my assumption that if someone still plays a game after/on 3 days, it is because they enjoy it and are likely to stick with it long term. Since we used exactly three days after and not within 3 days, there is the disadvantage of missing players who hypothetically could not play that specific day in the real world, even though they may turn to be a long time fan of the game and played at day 2 or any day after day 3.

The best cohort for Day 3 on Classical Retention is on January 10, 2020, for Game 4

The best cohort for Day 1 and Day 3 is the cohort who first appeared in the game on January 09, 2020 and January 10, 2020 for Game 4

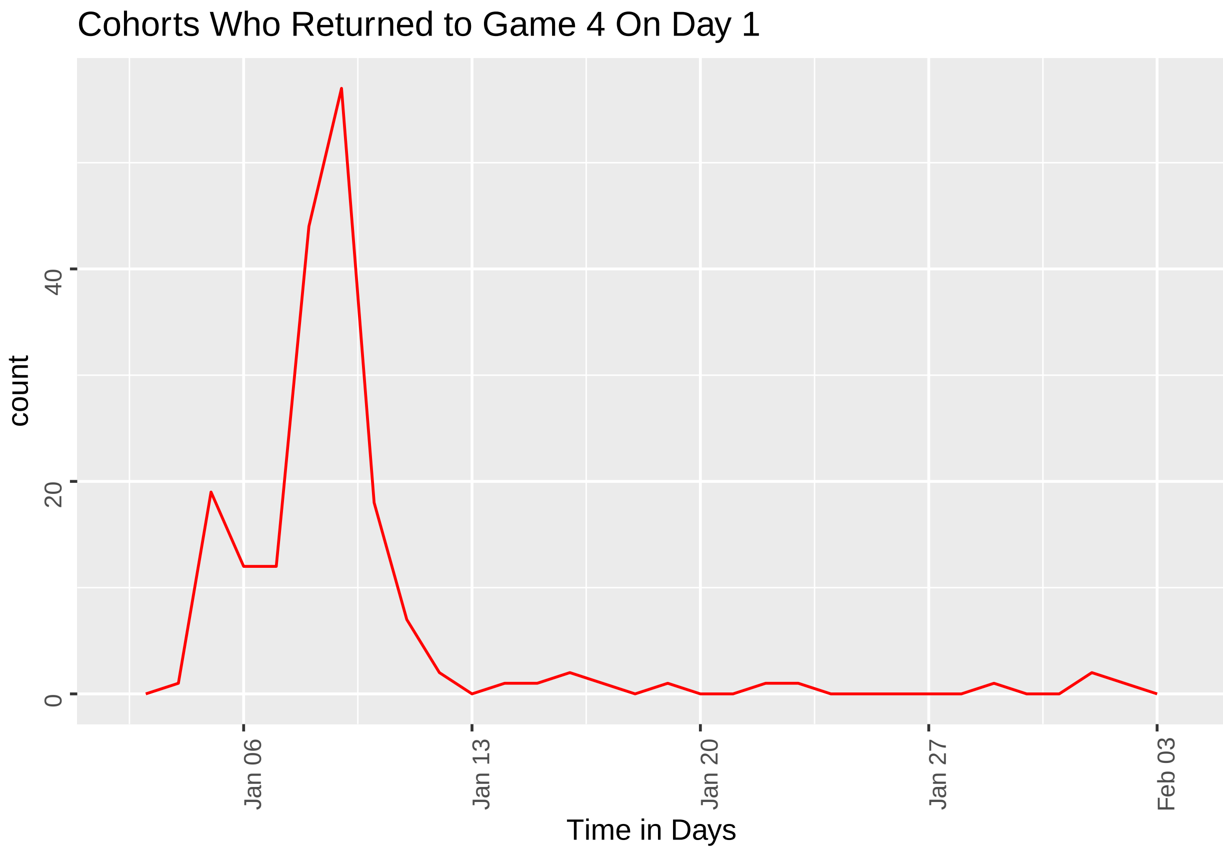
Vizualizing the above results

```
cohort_day1 <- game_4 %>% group_by(FirstSessionDate, UserID) %>%
  count(match_Day1) %>%
  filter(match_Day1 == '1', n >= 2) #data for visualization
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
ggplot(aes(x=FirstSessionDate), data = cohort_day1)+xlab("Time in Days")+
theme(axis.text=element_text(angle=90))+
  geom_freqpoly(aes(), color = "red")+
  ggtitle("Cohorts Who Returned to Game 4 On Day 1")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



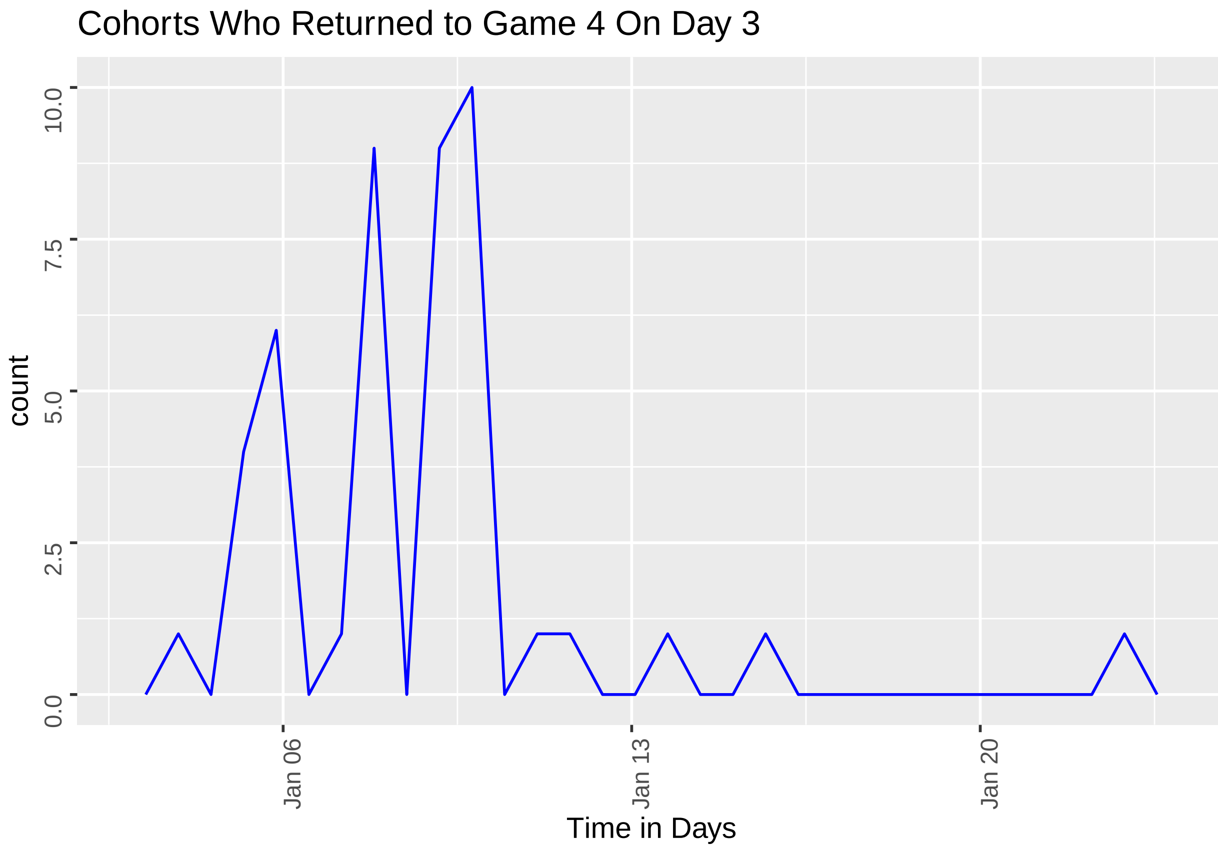

```
cohort_day3 <- game_4 %>%
  group_by(FirstSessionDate, UserID) %>%
  count(match_Day3) %>%
  filter(match_Day3 == '1', n >= 1) #data for vizualization
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
ggplot(aes(x=FirstSessionDate), data = cohort_day3)+xlab("Time in Days")+
theme(axis.text=element_text(angle=90))+
  geom_freqpoly(aes(), color = "blue")+
```

```r
  ggtitle("Cohorts Who Returned to Game 4 On Day 3")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We see that the cohort who did best on Day 1, is the cohort who first appeared in the game on Jan. 09, 2020.

The cohort that did best on Day 3, is the cohort who first appeared in the game on Jan. 10, 2020. Since Jan. 09, 2020 was the most popular day for the game based on the data, I suspect that players who first appeared on Jan. 10, 2020 are players who heard of the games popularity and became active users due to the presented popularity.

I assume that the popularity is defined by the amount of new users per date, and that only frequent users (more than once) contribute to the games success/popularity.

It is frequent users that are the ones who are mostly exposed to advertisments/microtransactions in game, thus from a business perspective, the cohort from Jan 09, 2020 and Jan 10, 2020 are the ones most likely to spend real money in game.