# Final Project

ChenyueLiao

5/2/2020

Part 1. The Trend of Populer Games.

In this part, I collect data from “https://steamcharts.com/ (https://steamcharts.com/)”. There are some basic data of the most popular Steam games on the website.The data on this website is not very comprehensive, so the data can only show the game player’s situation

I planned to use data from “steamspy.com” and “steamdb.info” which can provide more diverse data. However, I found that I can not use “read_html()” to collect these 2 websites’ data. The requests were refused by them. Therefore, some interesting research can not be shown in this project. I choose data from github as substite.

Collect the recent 2 years’ data of Counter-Strike: Global Offensive.

There are tables on the website which contain all the data, so I use “html_nodes(”table“)” and “html_table(fill = TRUE)” to collect them. The data collected from the website is the number of average players and peak players in the past 94 months. I want to get the recent 2 years’ data, so I just intercepted data from row 2 to row 25.

Another question is that the timeline of the data is from near to far, but I want it to be from far to near. To handle this, I added a column which contains its row number. And then I sorted the dataframe in descending order by the row number. After that I deleted this row number column. And I got the 2-year game players’ trend dataframe I wanted.

```
csgo_data <- "https://steamcharts.com/app/730" %>%
  read_html() %>%
  html_nodes("table") %>%
  html_table(fill = TRUE)

csgo2year <- csgo_data[[1]][2:25,]

csgo2year <- csgo2year %>%
  mutate(Number = rownames(csgo2year))

csgo2year$Number <- as.numeric(csgo2year$Number)

csgo2year <- csgo2year %>%
  arrange(desc(Number)) %>%
  select(Month, `Avg. Players`, `Gain`, `% Gain`, `Peak Players`)

head(csgo2year, 10)
```

```
##             Month Avg. Players      Gain  % Gain Peak Players
## 1        May 2018     262170.9 -26905.82  -9.31%       454481
## 2       June 2018     266862.2   4691.36  +1.79%       420261
## 3       July 2018     273307.3   6445.02  +2.42%       426008
## 4     August 2018     283531.3  10224.05  +3.74%       454370
## 5  September 2018     333164.0  49632.68 +17.51%       583029
## 6    October 2018     325907.8  -7256.17  -2.18%       565968
## 7   November 2018     310085.4 -15822.39  -4.85%       546031
## 8   December 2018     395509.3  85423.83 +27.55%       746548
## 9    January 2019     401366.9   5857.61  +1.48%       684511
## 10  February 2019     371359.0 -30007.91  -7.48%       654069
```

Collect the recent 2 years' data of Dota2. The collecting method is the same.

```
##             Month Avg. Players      Gain  % Gain Peak Players
## 1        May 2018     474325.9  43984.93 +10.22%       844713
## 2       June 2018     473900.0   -425.87  -0.09%       796886
## 3       July 2018     441714.3 -32185.65  -6.79%       701582
## 4     August 2018     476101.1  34386.73  +7.78%       829281
## 5  September 2018     466470.7  -9630.34  -2.02%       826166
## 6    October 2018     431173.9 -35296.83  -7.57%       739643
## 7   November 2018     461073.5  29899.57  +6.93%       826053
## 8   December 2018     439367.8 -21705.66  -4.71%       765422
## 9    January 2019     475747.0  36379.17  +8.28%       874888
## 10  February 2019     564909.7  89162.67 +18.74%       964921
```

Collect the recent 2 years' data of Playerunknown's Battlegrounds. The collecting method is the same.

```
##             Month Avg. Players       Gain  % Gain Peak Players
## 1        May 2018     876180.6 -231001.16 -20.86%      2175704
## 2       June 2018     800668.2  -75512.41  -8.62%      1750216
## 3       July 2018     688620.4 -112047.81 -13.99%      1350463
## 4     August 2018     619320.5  -69299.93 -10.06%      1260894
## 5  September 2018     542607.1  -76713.44 -12.39%      1125229
## 6    October 2018     469141.7  -73465.31 -13.54%      1048662
## 7   November 2018     418159.5  -50982.25 -10.87%       895650
## 8   December 2018     473541.3   55381.78 +13.24%      1109766
## 9    January 2019     497803.2   24261.99  +5.12%      1084606
## 10  February 2019     437959.1  -59844.15 -12.02%       931754
```

Collect the recent 2 years' data of Grand Theft Auto 5. The collecting method is the same.

```
##             Month Avg. Players      Gain  % Gain Peak Players
## 1        May 2018     41278.73  -1500.80  -3.51%        82956
## 2       June 2018     48927.49   7648.77 +18.53%       118332
## 3       July 2018     80395.34  31467.85 +64.32%       162021
## 4     August 2018     68312.15 -12083.19 -15.03%       123556
## 5  September 2018     49147.67 -19164.49 -28.05%        97920
## 6    October 2018     43265.97  -5881.70 -11.97%        81337
## 7   November 2018     43009.38   -256.59  -0.59%        81360
## 8   December 2018     55641.33  12631.96 +29.37%       120693
## 9    January 2019     59851.20   4209.87  +7.57%       118210
## 10  February 2019     58124.22  -1726.98  -2.89%       119439
```

Collect the recent 2 years' data of Tom Clancy's Rainbow Six Siege. The collecting method is the same.

```
##             Month Avg. Players      Gain  % Gain Peak Players
## 1        May 2018     63092.81  -1170.62  -1.82%       142362
## 2       June 2018     66376.81   3283.99  +5.21%       141630
## 3       July 2018     68241.00   1864.20  +2.81%       116054
## 4     August 2018     73816.38   5575.38  +8.17%       134159
## 5  September 2018     73676.87   -139.52  -0.19%       143866
## 6    October 2018     62428.64 -11248.23 -15.27%       123794
## 7   November 2018     57902.93  -4525.71  -7.25%       115820
## 8   December 2018     70865.32  12962.39 +22.39%       129558
## 9    January 2019     79175.76   8310.43 +11.73%       137434
## 10  February 2019     72997.66  -6178.10  -7.80%       136018
```

Collect the recent 2 years' data of Ark: Survival Envolved. The collecting method is the same.

```
##             Month Avg. Players      Gain  % Gain Peak Players
## 1        May 2018     31404.34  -3013.94  -8.76%        50021
## 2       June 2018     41794.49  10390.15 +33.09%        79451
## 3       July 2018     45060.20   3265.71  +7.81%        70823
## 4     August 2018     32497.43 -12562.77 -27.88%        50789
## 5  September 2018     28288.19  -4209.24 -12.95%        49072
## 6    October 2018     29337.62   1049.43  +3.71%        49049
## 7   November 2018     51906.05  22568.43 +76.93%       106170
## 8   December 2018     42992.14  -8913.91 -17.17%        76351
## 9    January 2019     36177.70  -6814.44 -15.85%        58388
## 10  February 2019     32446.72  -3730.98 -10.31%        52697
```

Here I needed a table which could show all the six games' average number of players in past 2 years.

First, I used "select(Month, `Avg. Players`)" to get six subtable with column "Month" and "Avg. Players". And then, I changed the second column names of subtables with "gamename_avg" in order to do joins. Finally, I used "left_join" for 5 times to combine the 6 subtables and got the "games_avg_trend" table which would be used to draw the plot.

```
##             Month csgo_avg dota2_avg pubg_avg gta5_avg rainbow6_avg  ark_avg
## 1        May 2018 262170.9  474325.9 876180.6 41278.73     63092.81 31404.34
## 2       June 2018 266862.2  473900.0 800668.2 48927.49     66376.81 41794.49
## 3       July 2018 273307.3  441714.3 688620.4 80395.34     68241.00 45060.20
## 4     August 2018 283531.3  476101.1 619320.5 68312.15     73816.38 32497.43
## 5  September 2018 333164.0  466470.7 542607.1 49147.67     73676.87 28288.19
## 6    October 2018 325907.8  431173.9 469141.7 43265.97     62428.64 29337.62
## 7   November 2018 310085.4  461073.5 418159.5 43009.38     57902.93 51906.05
## 8   December 2018 395509.3  439367.8 473541.3 55641.33     70865.32 42992.14
## 9    January 2019 401366.9  475747.0 497803.2 59851.20     79175.76 36177.70
## 10  February 2019 371359.0  564909.7 437959.1 58124.22     72997.66 32446.72
```

When I tried to draw the plot using “ggplot2” at first, I met a big problem: the order of x axis labels were arranged alphabetically by R. This was definitely not what I wanted.

I thought the problem was caused by factor vectors, because the default level in factor vectors is alphabetical. So I specified the order of levels by “levels = unique(games_avg_trend$Month)”.

```
games_avg_trend$Month <- factor(games_avg_trend$Month, levels = unique(games_avg_trend$M
onth))
```

After I specified the Month factor, I could successfully finish my plot.

Since the values are discrete, I used “geom_point(aes(y = csgo_avg)) + geom_line(aes(y = csgo_avg, group = 1, color =”CS:GO“))” to draw a line chart.

In addition, I used “ggtitle” to add a title, and used “xlab” and “ylab” to add labels on both axes.

Besides, I had to use “scale_x_discrete(labels = abbreviate) +theme(axis.text = element_text(angle = 90))” to adjust the angle of the axis scale because there wasn’t enough space for the characters.

At last, the plot, “The Average Number of Players’ Trends in the Past Two Years” is shown here.

From the plot we can discover that CS:GO, DOTA2 and PUBG are far more popular than the other 3 games.

What is striking is that the players of PUBG is keeping decreasing since May, 2018. The game lost more than two thirds of players in the recent 2 years. The phenomenon is obviously caused by the bad operating strategy of its developer, Bluehole, and another reason is that there are a lot of cheaters in the game which made players having bad game experience.

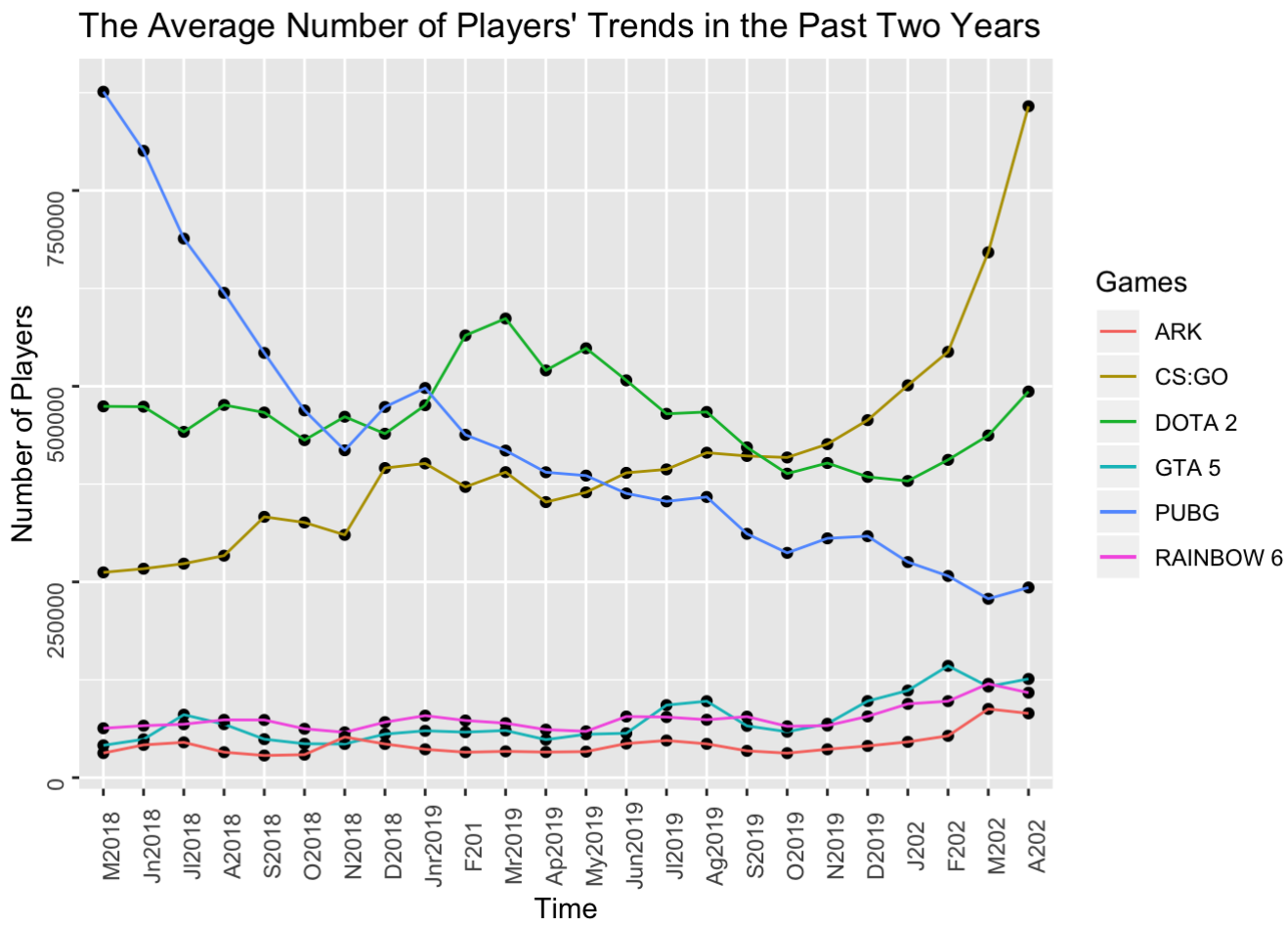
In contrast, the average players of CS:GO is keeping increasing. The number of players in April 2020 is 3 times of the number in May 2018.

Another popular game, Dota 2 has stable performance. The number is around 500k in past 2 years. During March 2019 to August 2019, the game once became the most popular one. I believe it is because the “Autochess” mode which attracted a lot of new players.

Since January 2020, the numbers of players of all games except PUBG have increased significantly. I think this is caused by the COVID-19 quarantine. People have to stay at home, and playing video games is an excellent kind of entertainment.

```
average_plot <- ggplot(data = games_avg_trend, aes(x = Month)) +
  geom_point(aes(y = csgo_avg)) +
  geom_line(aes(y = csgo_avg, group = 1, color = "CS:GO")) +
  geom_point(aes(y = dota2_avg)) +
  geom_line(aes(y = dota2_avg, group = 1, color = "DOTA 2")) +
  geom_point(aes(y = pubg_avg)) +
  geom_line(aes(y = pubg_avg, group = 1, color = "PUBG")) +
  geom_point(aes(y = gta5_avg)) +
  geom_line(aes(y = gta5_avg, group = 1, color = "GTA 5")) +
  geom_point(aes(y = rainbow6_avg)) +
  geom_line(aes(y = rainbow6_avg, group = 1, color = "RAINBOW 6")) +
  geom_point(aes(y = ark_avg)) +
  geom_line(aes(y = ark_avg, group = 1, color = "ARK")) +
  ggtitle("The Average Number of Players' Trends in the Past Two Years") +
  xlab("Time") +
  ylab("Number of Players") +
  scale_color_discrete(name = "Games") +
  scale_x_discrete(labels = abbreviate) +
  theme(axis.text = element_text(angle = 90))

average_plot
```




For this part, I needed a table which could show all the six games' peak number of players in past 2 years.

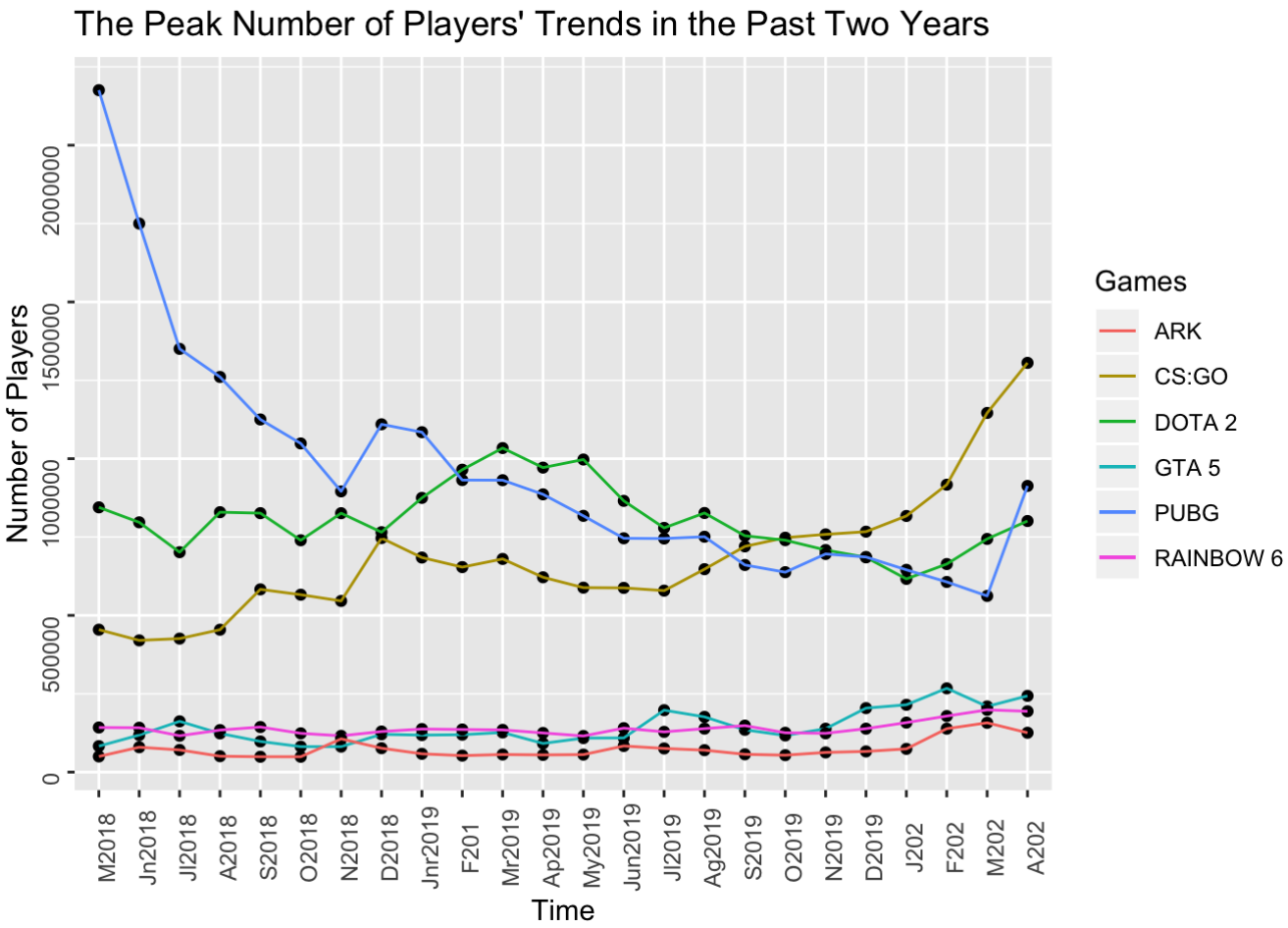
First, I used “select(Month, `Peak Players`)” to get six subtable with column “Month” and “Peak Players”. And then, I changed the second column names of subtables with “gamename_peak” in order to do joins. Finally, I used “left_join” for 5 times to combine the 6 subtables and got the “games_peak_trend” table which would be used to draw the plot.

And of course I used “levels = unique(games_avg_trend$Month)” again to specify the levels’ order in the factor vectors.

```
##             Month csgo_peak dota2_peak pubg_peak gta5_peak rainbow6_peak
## 1        May 2018    454481     844713   2175704     82956         142362
## 2       June 2018    420261     796886   1750216    118332         141630
## 3       July 2018    426008     701582   1350463    162021         116054
## 4     August 2018    454370     829281   1260894    123556         134159
## 5  September 2018    583029     826166   1125229     97920         143866
## 6    October 2018    565968     739643   1048662     81337         123794
## 7   November 2018    546031     826053    895650     81360         115820
## 8   December 2018    746548     765422   1109766    120693         129558
## 9    January 2019    684511     874888   1084606    118210         137434
## 10  February 2019    654069     964921    931754    119439         136018
##    ark_peak
## 1     50021
## 2     79451
## 3     70823
## 4     50789
## 5     49072
## 6     49049
## 7    106170
## 8     76351
## 9     58388
## 10    52697
```

To draw “The Peak Number of Players’ Trends in the Past Two Years” plot, I used the code which is similar to the average number plot. Including “geom_point(aes(y = csgo_peak)) + geom_line(aes(y = csgo_avg, group = 1, color =”CS:GO“))” and “scale_x_discrete(labels = abbreviate) +theme(axis.text = element_text(angle = 90))”.

From the peak plot we can see the great popularity of PUBG before 2019. I went to search PUBG’s peak number of players of all time. It is a huge number that you cannot imagine, 3,236,027. This number is bigger than the sum of all the other games’ players.




Part 2

Here are the steam history statistics from github. The data comes courtesy of Liza Wood via Steam Spy. There is time played, ownership, release date, publishing information, and for some a metascore. And the average and median playtime is over the last two weeks when the data was collected, as such there are many many games where playtime is low or zero. When I analyzed the data, I remove all the 0 value and na value, so there might be some limitation.

```
## # A tibble: 10 x 9
##    number game  release_date price score_rank_user… owners playtime_median
##     <dbl> <chr> <chr>        <chr> <chr>            <chr>  <chr>
##  1      1 Half… Nov 16, 2004 9.99  N/A (N/A/96%)    10,00… 01:50 (01:06)
##  2      3 Coun… Nov 1, 2004  9.99  N/A (N/A/88%)    10,00… 03:56 (02:08)
##  3     21 Coun… Mar 1, 2004  9.99  N/A (N/A/65%)    10,00… 00:10 (00:03)
##  4     47 Half… Nov 1, 2004  4.99  N/A (N/A)        5,000… 00:00 (00:00)
##  5     36 Half… Jun 1, 2004  9.99  N/A (N/A)        2,000… 00:00 (00:00)
##  6     52 CS2D  Dec 24, 2004 Free  N/A (N/A)        1,000… 00:16 (00:10)
##  7      2 Unre… Mar 16, 2004 14.99 N/A (N/A/93%)    500,0… 00:00 (00:00)
##  8      4 DOOM… Aug 3, 2004  4.99  N/A (N/A/87%)    500,0… 00:00 (00:00)
##  9     14 Beyo… Apr 27, 2004 5.99  N/A (N/A/73%)    500,0… 00:00 (00:00)
## 10     40 Hitm… Apr 20, 2004 8.99  N/A (N/A)        500,0… 00:00 (00:00)
## # … with 2 more variables: developer_s <chr>, publisher_s <chr>
```

And there some clean up. Using “as.numeric()” to change data type of “price”, “playtime” and “metascore”. And I deleted the extra characters in the dataframe.

```
## # A tibble: 10 x 10
##    number game  release_date price owners developer publisher average_playtime
##     <dbl> <chr> <chr>        <dbl> <chr>  <chr>     <chr>                 <dbl>
##  1      1 Half… Nov 16, 2004  9.99 10,00… Valve     Valve                   110
##  2      3 Coun… Nov 1, 2004   9.99 10,00… Valve     Valve                   236
##  3     21 Coun… Mar 1, 2004   9.99 10,00… Valve     Valve                    10
##  4     47 Half… Nov 1, 2004   4.99 5,000… Valve     Valve                     0
##  5     36 Half… Jun 1, 2004   9.99 2,000… Valve     Valve                     0
##  6     52 CS2D  Dec 24, 2004 NA    1,000… Unreal S… Unreal S…                16
##  7      2 Unre… Mar 16, 2004 15.0  500,0… Epic Gam… Epic Gam…                 0
##  8      4 DOOM… Aug 3, 2004   4.99 500,0… id Softw… id Softw…                 0
##  9     14 Beyo… Apr 27, 2004  5.99 500,0… Larian S… Larian S…                 0
## 10     40 Hitm… Apr 20, 2004  8.99 500,0… Io-Inter… Io-Inter…                 0
## # … with 2 more variables: median_playtime <dbl>, metascore <dbl>
```

Then, I removed all the 0 value and NA value in the dataframe, in order to let it more convenient to do further research.

```
## # A tibble: 10 x 10
##    number game  release_date price owners developer publisher average_playtime
##     <dbl> <chr> <chr>        <dbl> <chr>  <chr>     <chr>                 <dbl>
##  1      1 Half… Nov 16, 2004  9.99 10,00… Valve     Valve                   110
##  2      3 Coun… Nov 1, 2004   9.99 10,00… Valve     Valve                   236
##  3     21 Coun… Mar 1, 2004   9.99 10,00… Valve     Valve                    10
##  4      2 Gran… Jun 6, 2005  15.0  2,000… Rockstar… Rockstar…               373
##  5      7 STAR… Feb 8, 2005   9.99 2,000… Obsidian… LucasArt…                37
##  6      5 Half… Jun 1, 2006   7.99 5,000… Valve     Valve                    12
##  7     29 Eart… Apr 1, 2006   4.99 200,0… Reality … Topware …               430
##  8      4 Team… Oct 10, 2007 NA    50,00… Valve     Valve                   336
##  9      8 Port… Oct 10, 2007  9.99 10,00… Valve     Valve                    69
## 10      7 Half… Oct 10, 2007  7.99 5,000… Valve     Valve                    15
## # … with 2 more variables: median_playtime <dbl>, metascore <dbl>
```

I'm concentrating on the developer's statistics in this part. So, I use "group_by" and "summarise()" to get the games' average playtime and metascore from different developers.

```
## # A tibble: 10 x 3
##    developer                                    avg_playtime avg_metascore
##    <chr>                                               <dbl>         <dbl>
##  1 11 bit studios                                       47           83.5
##  2 2K Australia, Gearbox Software, Aspyr (Linux)       445           75
##  3 5th Cell Media                                       31           75
##  4 Airship Syndicate                                   135           78
##  5 Amanita Design                                       19           80
##  6 AMPLITUDE Studios                                   200.          78
##  7 anchor Inc.                                        1216.          79.5
##  8 Antimatter Games, Tripwire Interactive               91           81
##  9 Arc System Works                                     60           85
## 10 Arkane Studios                                        1           86
```

For average playtime and score, I decided to analyze them seperately because they are different statisitcal indicators.

I divided the developer statistics dataframe into 2 dataframes, and ordered them in descending sort.

From both 2 dataframes, I chose the top 10 developers. From different perspectives, we got different answers about top developers.

In the playtime table, the top five developers are Blazing Griffin, Clifftop Games, KOEI TECMO GAMES CO., LTD., Gone North Games and Square Enix. In the metascore table, the top five developers are Rockstar North, Neko Climax Studios, Irrational Games, Virtual Programming (Linux), Rockstar Games and Larian Studios.

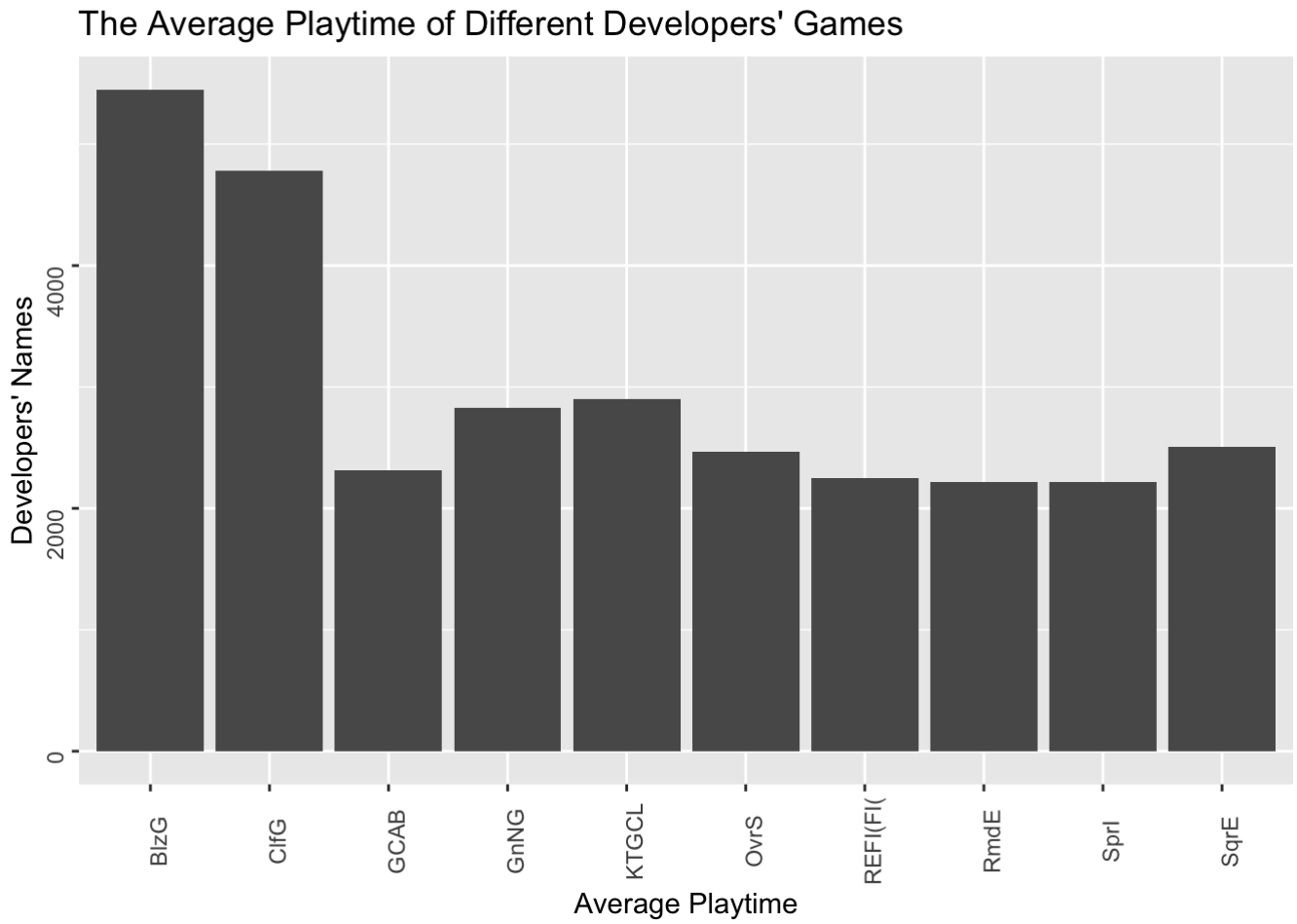
For myself, I prefer the answer from metascore table, because those developers do have a lot of famous and popular games such as Divinity II, GTA 5, Red Dead Redemption II.

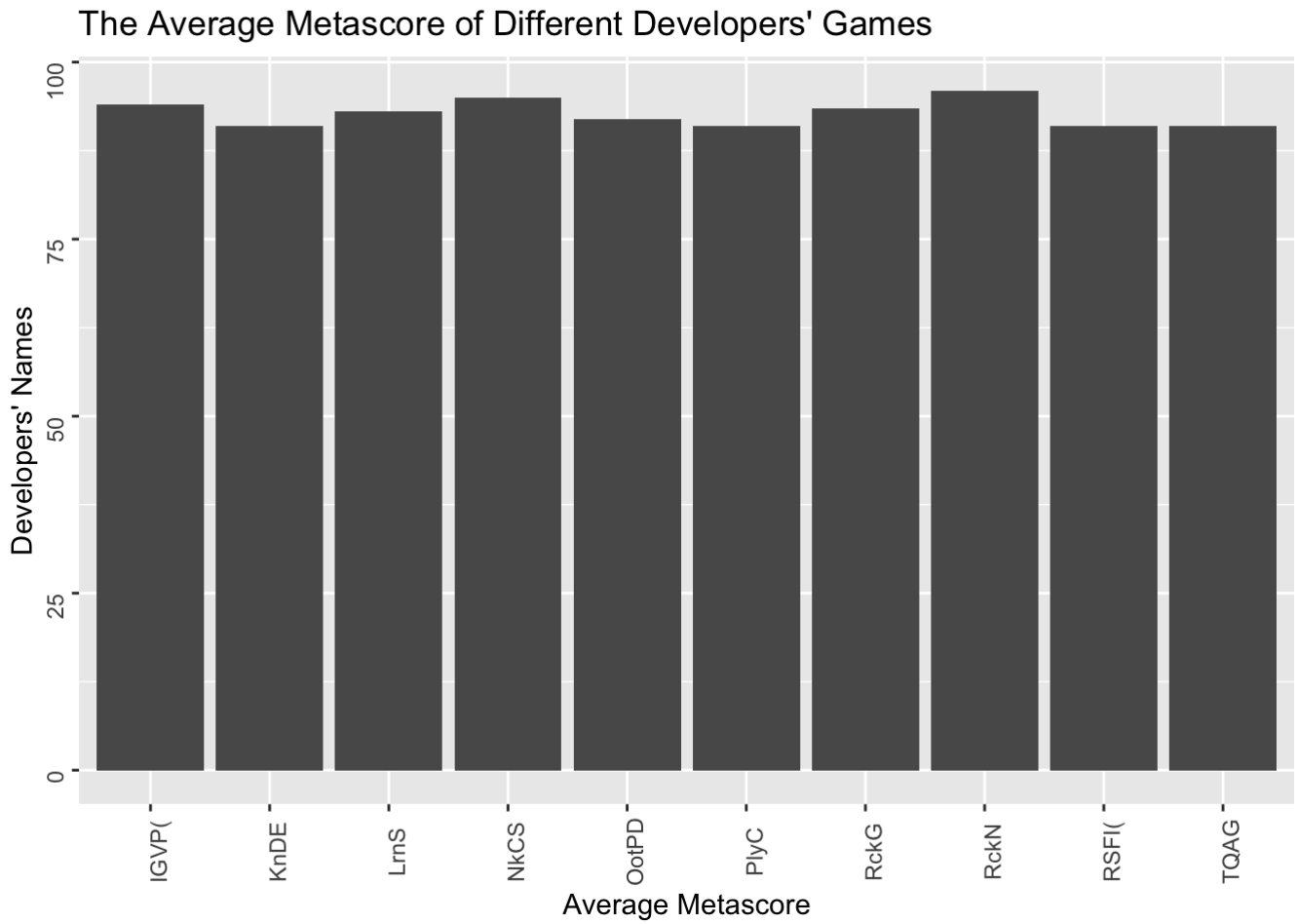
And from the part 1 of this report, we can find that the game, GTA 5 developed by Rockstar Games and Rockstar North is still very popular now. However it is a game developed 7 years ago. Throughout the history of the game, this is very amazing.

```
## # A tibble: 10 x 2
##    developer                                                    avg_playtime
##    <chr>                                                               <dbl>
##  1 Blazing Griffin                                                      5450
##  2 Clifftop Games                                                       4779
##  3 KOEI TECMO GAMES CO., LTD.                                           2904
##  4 Gone North Games                                                     2830
##  5 Square Enix                                                          2503
##  6 Overhype Studios                                                     2464
##  7 Gaming Corps AB                                                      2313
##  8 Relic Entertainment, Feral Interactive (Mac), Feral Interactive…     2248
##  9 Remedy Entertainment                                                 2221
## 10 Sports Interactive                                                   2220
```

```
## # A tibble: 10 x 2
##    developer                                    avg_metascore
##    <chr>                                                <dbl>
##  1 Rockstar North                                        96
##  2 Neko Climax Studios                                   95
##  3 Irrational Games, Virtual Programming (Linux)         94
##  4 Rockstar Games                                        93.5
##  5 Larian Studios                                        93
##  6 Out of the Park Developments                          92
##  7 Konami Digital Entertainment                          91
##  8 Polytron Corporation                                  91
##  9 Rocksteady Studios, Feral Interactive (Mac)           91
## 10 The Quantum Astrophysicists Guild                     91
```

To make the data more intuitive, I drew 2 histgrams using the dataframes. Since it is numeric variable in column “avg_playtime” and “avg_metascore”, I used “geom_bar(stat =”identity“)”.






In this report, I did analysis about the most popular games in recent 2 years and top game developers. I have to admit that there may be some limitation in the answer I got. However, I think it is easy for people who are not so familiar with video games to understand the games and developers on STEAM.

Moreover, I hope that one day, video games will be widely accepted by people of all ages without being unjustly biased. After all, it is a good entertainment.