Assessment Autumn 2025

Ken Lu

2025-05-07

Read dataset

```
rm(list = ls()) # cleaning the environment

PGA <- read.csv("businessesPGA.csv")
PGB <- read.csv("businessesPGB.csv")
review <- read.csv("reviews.csv")
users <- read.csv("users.csv")
```

Install and call the 'Tidyverse' package

```
# install.packages('tidyverse')
library('tidyverse')
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(knitr)

# install.packages("kableExtra") #
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
# for knitting
#install.packages("tinytex")
#tinytex::install_tinytex()
```

Data cleaning and grouping

```r
# Data cleaning and grouping
## Transform data type of member_since variable from chr into date.
users$member_since <- as.Date(users$member_since)

## Cleaning data - review
review <- review %>%
  mutate(user_id = str_trim(user_id)) %>% #remove any empty spaces
  filter(user_id != "") # remove any empty strings

## Cleaning data - users
users <- users %>%
  mutate(user_id = str_trim(user_id)) %>% #remove any empty spaces
  filter(user_id != "") # remove any empty strings

## group review and users into 1 group - maindata
maindata <- review %>%
  left_join(users, by = c("user_id" = "user_id")) %>% #fill more details of users into the reviews data
  drop_na(review_id, business_id, member_since, user_id) %>% # eliminate NA in review_id, business_id,
  select(-review_count, -average_stars) # eliminate unused columns
```

Finding
- First of all, from the review of data, each data is a set of sample which does not completely reflect each other.
- The review of questions shows that the analysis of customers behaviours is the primary objective and the dataset - 'review' is the dataset that has 'user_id', 'review_id' and 'business_id' to connect with other datasets, thus, it is selected as the main dataset to join variables of datasets such as member_since, state, etc).

Question 1:

1.1) Dividing the users into 3 groups: Veteran, Intermediate and New:

```r
# Classify the review from 3 groups of users
## The reviews from Veteran users
reviewOfVeteran <- maindata %>%
  filter(member_since < as.Date('2017-01-01'))

# The reviews from Intermediate users
reviewOfIntermediate <- maindata %>%
  filter(between(member_since, as.Date('2017-01-01'), as.Date('2022-12-31')))

# The reviews from New users
```

```r
reviewOfNew <- maindata %>%
  filter(member_since > as.Date('2022-12-31'))
```

1.2) Calculate the numbers of users, their average review stars and average number of reviews per user in each group.

```r
# The numbers of unique users per group
## group Veteran
Veteran <- reviewOfVeteran %>%
  group_by(user_id) %>%
  summarise(review_count = n_distinct(review_id)) # unique Veteran users from the review data

unique_users_V <- as.numeric(count(Veteran)) # The numbers of Veteran users

## group Intermediate
Intermediate <- reviewOfIntermediate %>%
  group_by(user_id) %>%
  summarise(review_count = n_distinct(review_id)) # unique Intermediate users from the review data

unique_users_I <- as.numeric(count(Intermediate)) # The numbers of Intermediate users

## group New
New <- reviewOfNew %>%
  group_by(user_id) %>%
  summarise(review_count = n_distinct(review_id)) # unique New users from the review data

unique_users_N <- as.numeric(count(New)) # The numbers of New users

# The average review stars per group
## average review stars from Veteran
avg_Review_Stars_V <- round(mean(reviewOfVeteran$stars), 3)

## average review stars from Intermediate
avg_Review_Stars_I <- round(mean(reviewOfIntermediate$stars), 3)

## average review stars from New
avg_Review_Stars_N <- round(mean(reviewOfNew$stars), 3)

# The average number of reviews per user in each group
## average number of reviews per user in Veteran
avg_Review_V <- round(mean(Veteran$review_count), 3)

## average number of reviews per user in Intermediate
avg_Review_I <- round(mean(Intermediate$review_count), 3)

## average number of reviews per user in New
avg_Review_N <- round(mean(New$review_count), 3)
```

```r
# Create table to turn it into a kable
user_group <- c("Veteran","Intermediate","New") # This will becomes the column names for the table
```

```r
value_Users <- c(unique_users_V, unique_users_I, unique_users_N)
value_Stars <- c(avg_Review_Stars_V, avg_Review_Stars_I, avg_Review_Stars_N)
value_Reviews <- c(avg_Review_V, avg_Review_I, avg_Review_N)

table <- data.frame(value_Users, value_Stars, value_Reviews, row.names = user_group)
## Turn the table into a kable
kable(table, caption = "The summary of average figures by groups", digits = 3, col.names = c("The Numbe
  kable_styling(bootstrap_options = c("bordered"), position = "center")
```

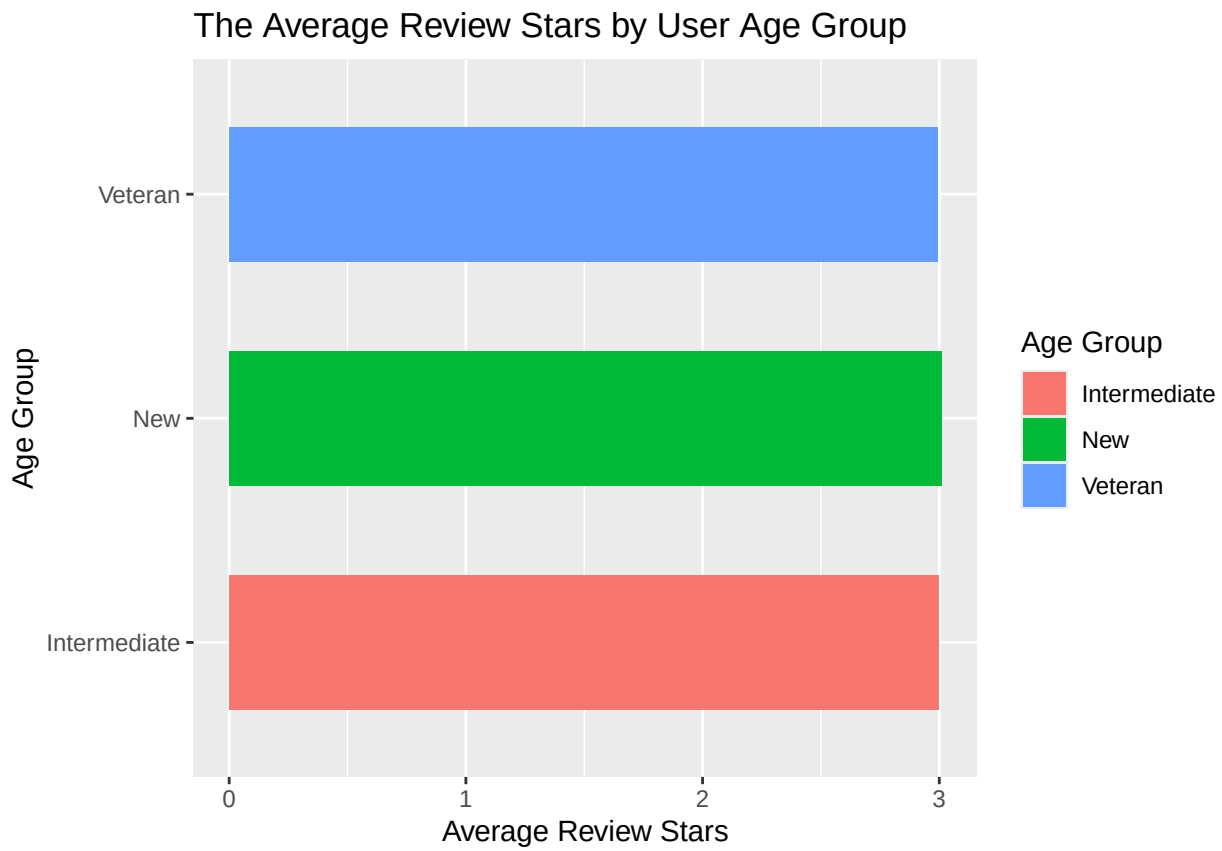
Table 1: The summary of average figures by groups

	The Numbers of Users	Average Review Stars	Average Number of Reviews
Veteran	6518	2.993	4.746
Intermediate	22470	2.999	4.751
New	8311	3.010	4.758

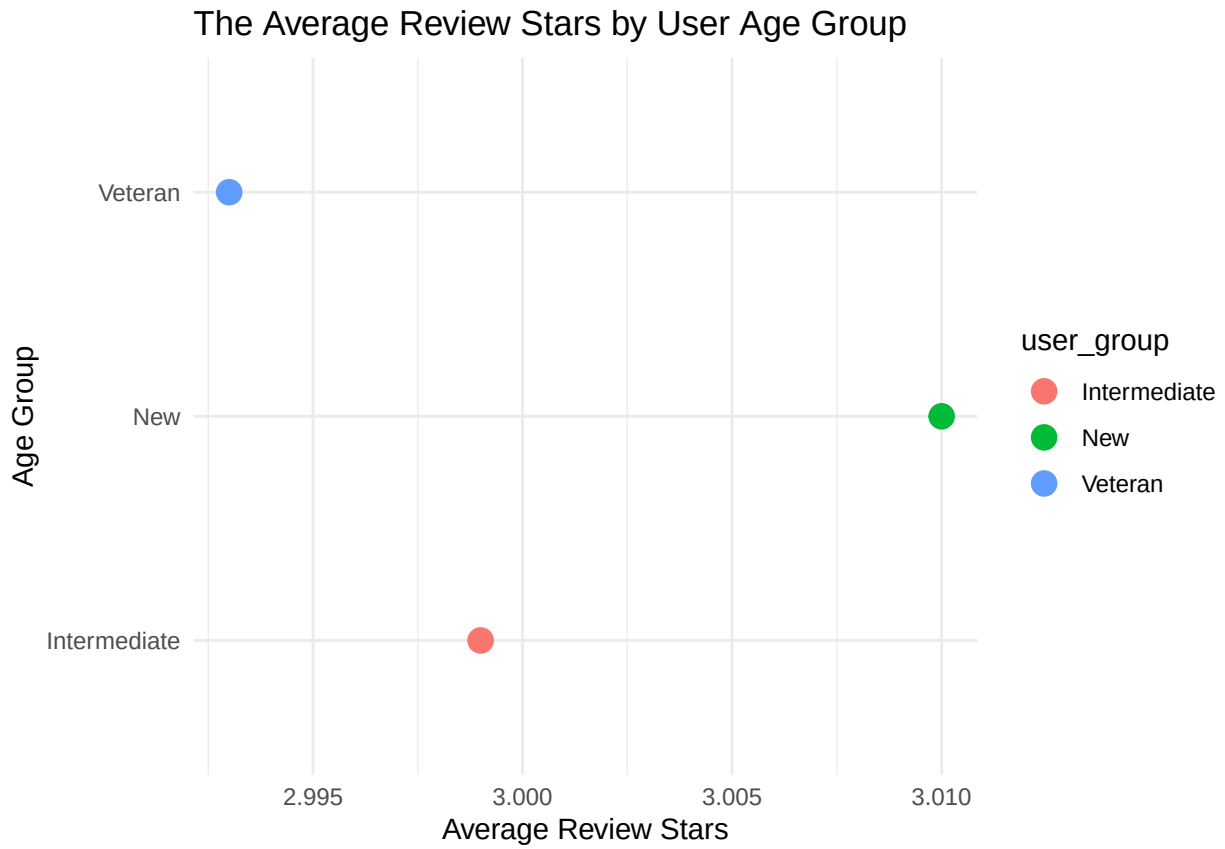
Findings from table - The summary of average figures by groups
- The analysis is mainly based on the review dataset, thus it focuses on the review's perspective to analyse.
- It is obvious to notice that the Intermediate accounts for the majoirty of users interactions in the dataset 'review'.
- However, the average review stars and the average number of reviews per user in each group are similar at 2.99 and 4.7 respectively.
- The scale of the stars rating presents the preference/favor of the users to the business they review. With the average review stars at 2.99 across all groups, it states that, despite the old or new users, they share a common 'normal' favor to the services of businesses they review.
- Moreover, each of user will give around 5 reviews (round up from 4.7) to 5 different businesses.
- From a different perspective, it could say that Intermedian users are the most active reviewers in this review sample. They accounts for the majority of reviews.
- However, it does not mean that the New users (joined since 1/1/2023) are not active. The year when this analysis is conducted is 2025, thus, the New users have been active at most around 3 years.
- Assuming all users are active, those who have been in the online community longer tend to have more comments than newer members. The number of unique Veteran user is less than the number of unique New user, but the average number of comments per New user is higher than the average number of comments per Veteran user. Therefore, the New users are also active.
- Overall, the data shows that the feelings of all identical users in the review sample towards the services they received are at average (normal). However, it also indicates that Intermediate and New users are very active in giving feedback for the businesses they visited.

1.3) Visualise the Average Review Stars by User Age Group. You are required to make sure you handle the NA value in your analysis. Explain your findings.

```r
# Bar chart to visual the overall view of the Average Review Stars by the User Age Group
Figure_1 <- ggplot(data = table, mapping = aes(x = user_group, y = value_Stars, fill = user_group)) + g
Figure_1
```




```
# Using points to indicates the difference between figures
Figure_2 <- ggplot(data = table, mapping = aes(x = user_group, y = value_Stars)) +
  labs(title = "The Average Review Stars by User Age Group", x = "Age Group", y = "Average Review Stars
  coord_flip() +
  theme_minimal()
Figure_2
```



Question 2:

2.1) Calculate the average review star, the number of reviews and the number of unique users. Is there any difference between the 2 datasets?

```
# Clean PGA and PGB datasets
## There are empty space/string in state in PGA and PGB. They need to be removed before continue
PGA <- PGA %>%
  mutate(state = str_trim(state)) %>% #remove any empty spaces
  filter(state != "") %>% # remove any empty strings in state column
  filter(business_id != "") # remove any empty strings in state column

PGB <- PGB %>%
  mutate(state = str_trim(state)) %>% #remove any empty spaces
  filter(state != "") %>% # remove any empty strings
  filter(business_id != "") # remove any empty strings in state column
```

Discussion on the data cleaning process:
- PGA and PGB are two data contain State to answer the question, thus, they are selected. The variables 'state' in both sets have its data type as character, so that is.na() becomes inapplicable to detect and filter empty values. Therefore, str_trim() and empty space as " ", are used to remove empty space detected in two sets.

```r
# Create 2 data from maindata, 1 with only PGA businesses and 1 with only PGB businesses
mainPGA <- maindata %>%
  inner_join(PGA, by = c("business_id" = "business_id")) %>% # join 2 groups using business_id
  select(-c(stars.y,review_count,categories, X, business_group, name.y,city)) # eliminate any unused co

mainPGB <- maindata %>%
  inner_join(PGB, by = c("business_id" = "business_id")) %>% # join 2 groups using business_id
  select(-c(stars.y,review_count,categories, X, business_group, name.y,city)) # eliminate any unused co
```

```r
# calculate figures from mainPGA
calculated_PGA <- mainPGA %>%
  group_by(state) %>%
  summarise("Average Review Star" = round(mean(stars.x),3), # calculate 3 values
            "Number of Reviews" = n_distinct(review_id),
            "Number of Unique Users" = n_distinct(user_id)) %>%
  mutate("The Proportion of Unique Users" = `Number of Unique Users`/sum(`Number of Unique Users`)) %>%
  arrange(state) # reorder the State variable in ascending order
calculated_PGA
```

```
## # A tibble: 51 x 5
##    state `Average Review Star` `Number of Reviews` `Number of Unique Users`
##    <chr>                 <dbl>               <int>                    <int>
##  1 AK                     2.96                1983                     1934
##  2 AL                     3.05                1796                     1759
##  3 AR                     3.03                1815                     1783
##  4 AZ                     3.00                2091                     2041
##  5 CA                     2.96                1962                     1913
##  6 CO                     3.00                1808                     1769
##  7 CT                     3.02                1773                     1737
##  8 DC                     2.98                1834                     1788
##  9 DE                     2.99                1700                     1668
## 10 FL                     2.98                1785                     1743
## # i 41 more rows
## # i 1 more variable: `The Proportion of Unique Users` <dbl>
```

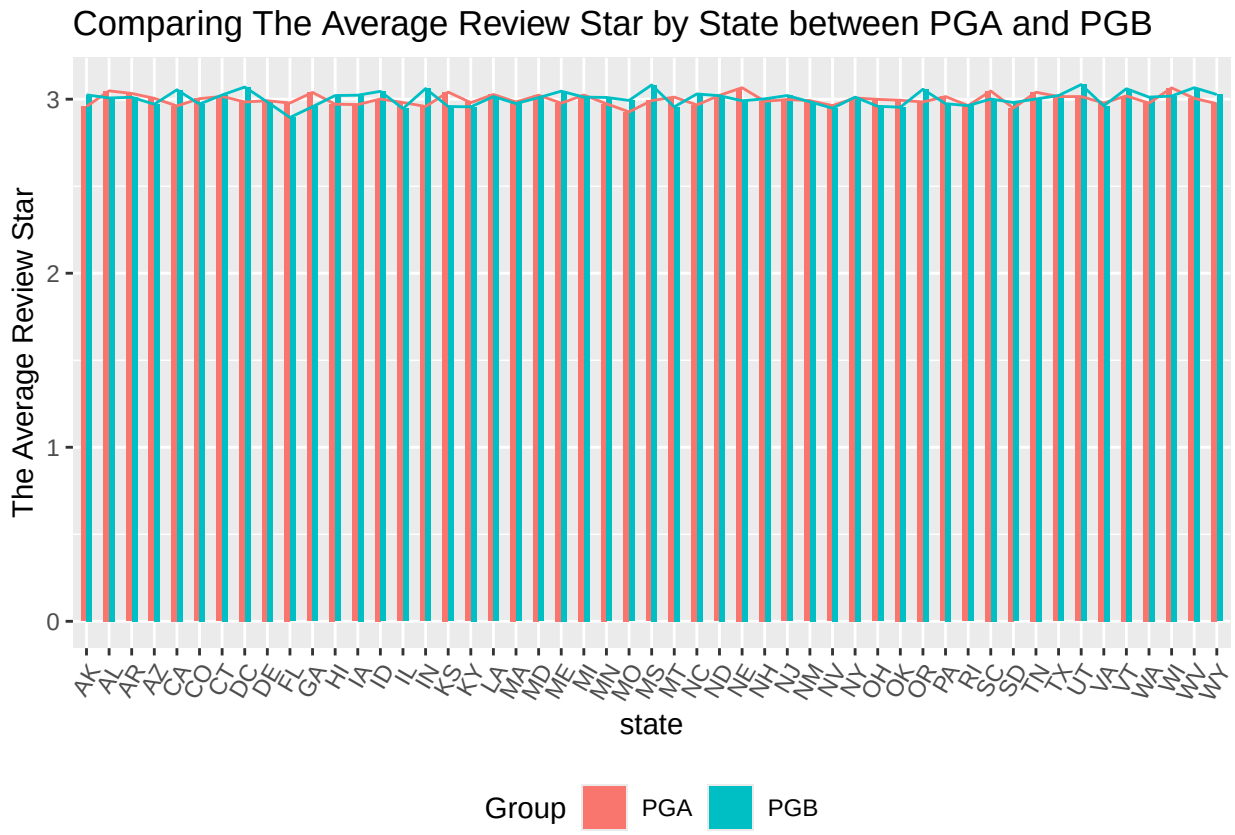
```r
# calculate figures from mainPGB
calculated_PGB <- mainPGB %>%
  group_by(state) %>%
  summarise("Average Review Star" = round(mean(stars.x),3), # calculate 3 values
            "Number of Reviews" = n_distinct(review_id),
            "Number of Unique Users" = n_distinct(user_id)) %>%
  mutate("The Proportion of Unique Users" = `Number of Unique Users`/sum(`Number of Unique Users`)) %>%
  arrange(state) # reorder the State variable in ascending order
calculated_PGB
```

```
## # A tibble: 51 x 5
##    state `Average Review Star` `Number of Reviews` `Number of Unique Users`
##    <chr>                 <dbl>               <int>                    <int>
##  1 AK                     3.02                1180                     1158
##  2 AL                     3.01                1436                     1405
##  3 AR                     3.01                1356                     1331
##  4 AZ                     2.97                1190                     1164
```

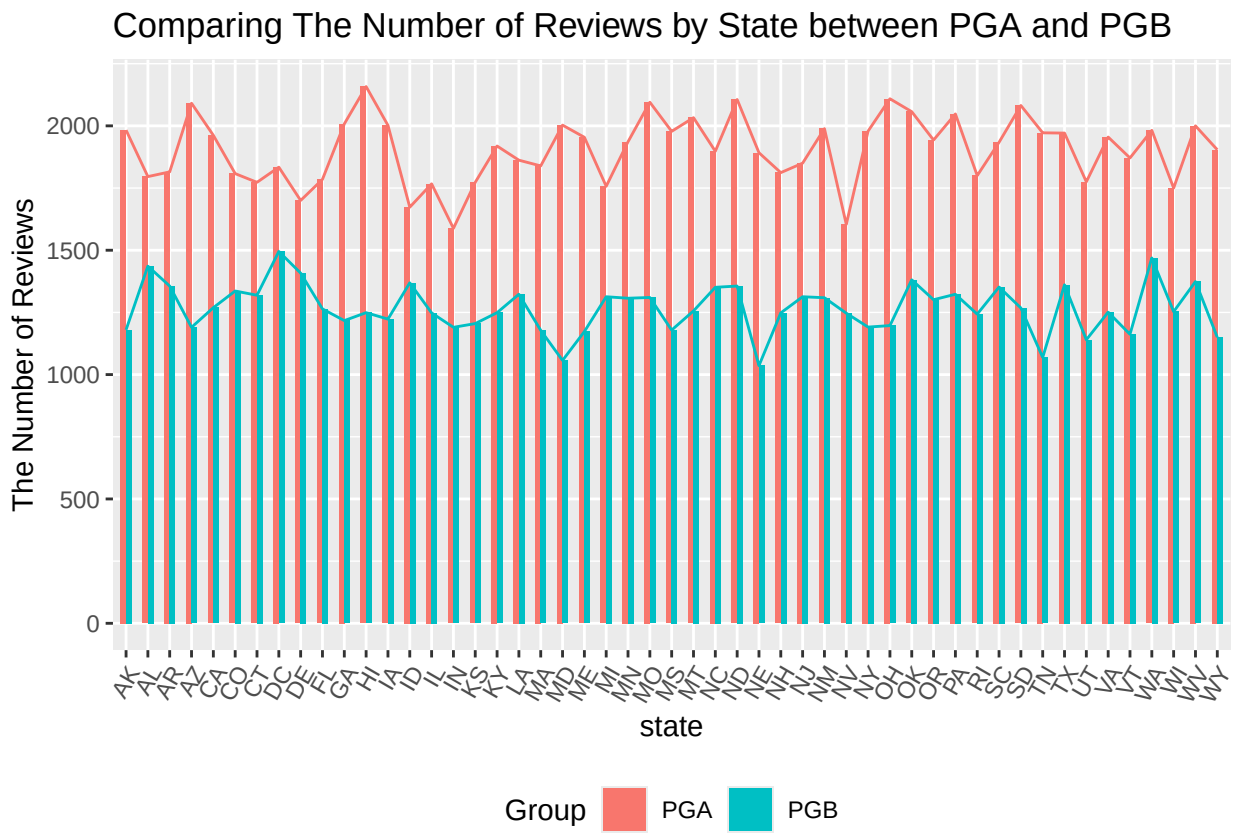
```
##  5 CA                    3.05                1269                    1251
##  6 CO                    2.97                1336                    1307
##  7 CT                    3.02                1319                    1300
##  8 DC                    3.07                1495                    1475
##  9 DE                    2.98                1408                    1383
## 10 FL                    2.90                1264                    1245
## # i 41 more rows
## # i 1 more variable: `The Proportion of Unique Users` <dbl>
```

2.2) Visualise the average review star, the number of reviews and the number of unique users by state

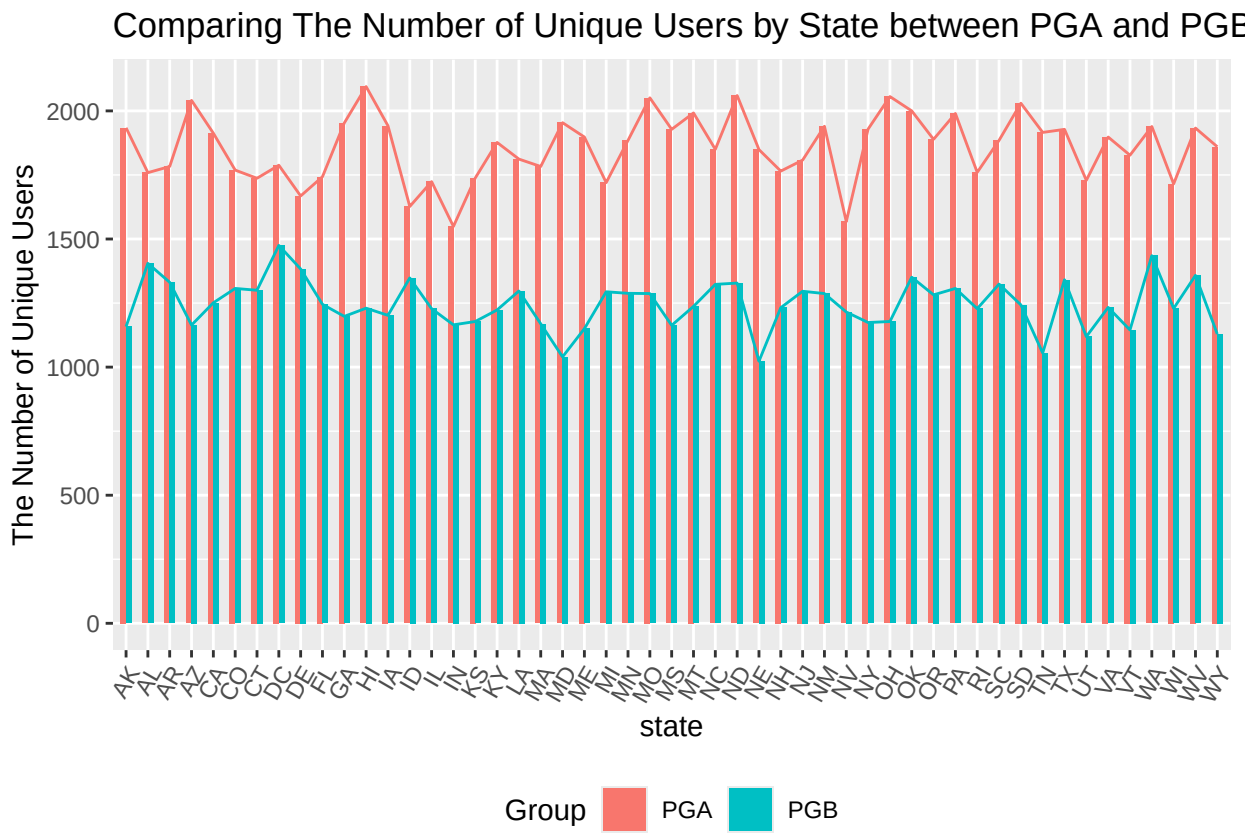
```
# combine 2 groups mainPGA and mainPGB for the comparison and visualisation purposes.
## add a column to know it is from PGA
calculated_PGA <- calculated_PGA %>%
  mutate("Group" = "PGA")

## add a column to know it is from PGB
calculated_PGB <- calculated_PGB %>%
  mutate("Group" = "PGB")

## Then, combine 2 groups so that we have a data set with category in advance
joinPGAPGB <- calculated_PGA %>%
  bind_rows(calculated_PGB)

## Visualise by Average Review Star
ggplot(data = joinPGAPGB, mapping = aes(x = state, y = `Average Review Star`, fill = Group)) + geom_col
  theme(axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "bottom") + # Rotate x-axi
  geom_line(aes(group = Group, color =Group)) +
  labs(y = "The Average Review Star", title = "Comparing The Average Review Star by State between PGA a
```



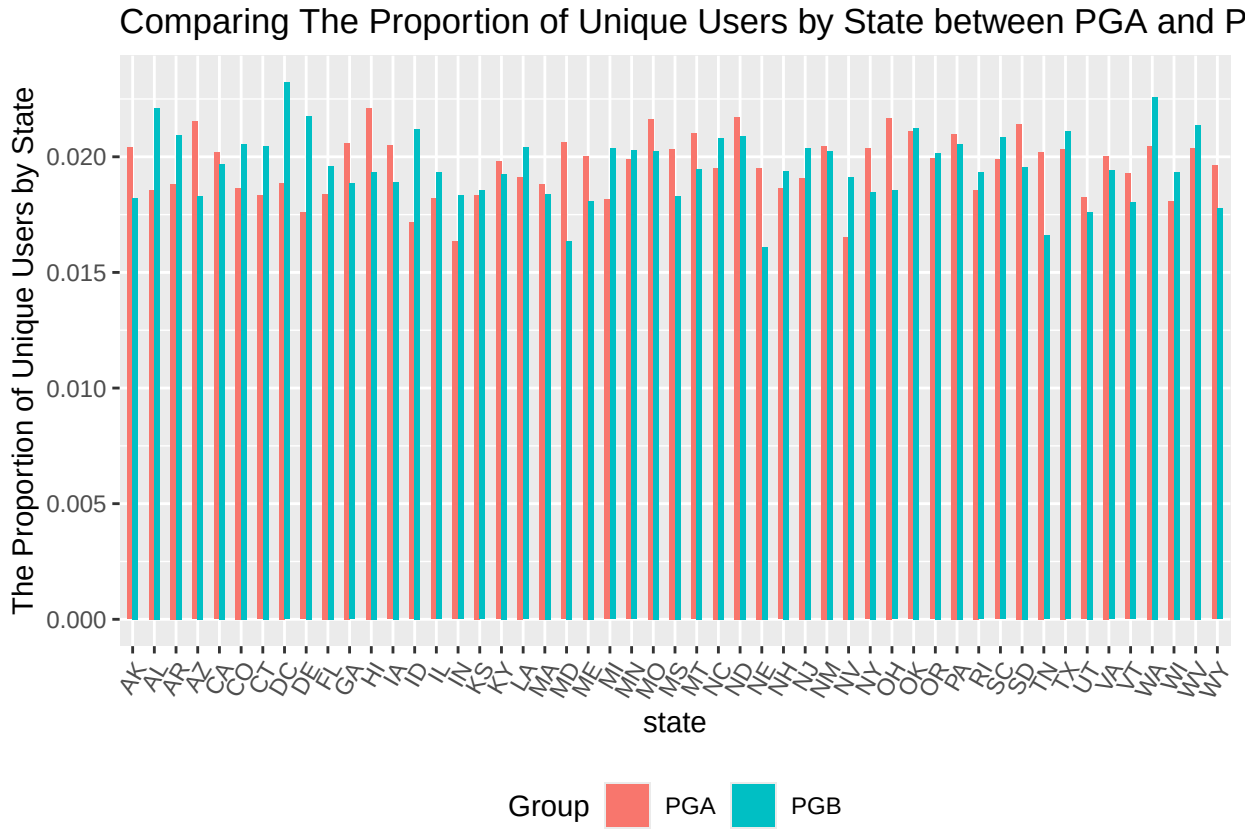

```
## Visualise by Number of Reviews
ggplot(data = joinPGAPGB, mapping = aes(x = state, y = `Number of Reviews`, fill = Group)) + geom_col(p
  theme(axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "bottom") + # Rotate x-axi
  geom_line(aes(group = Group, color =Group)) +
  labs(y = "The Number of Reviews", title = "Comparing The Number of Reviews by State between PGA and P
```




```
## Visualise by Number of Unique Users
ggplot(data = joinPGAPGB, mapping = aes(x = state, y = `Number of Unique Users`, fill = Group)) +
  geom_col(position = "dodge", width = 0.5) +
  geom_line(aes(group = Group, color =Group)) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "bottom") + # Rotate x-axi
  labs(y = "The Number of Unique Users", title = "Comparing The Number of Unique Users by State between
```




```
## Visualise by Proportion of Unique Users by State
ggplot(data = joinPGAPGB, mapping = aes(x = state, y = `The Proportion of Unique Users`, fill = Group))
  geom_col(position = "dodge", width = 0.5) +
  theme(axis.text.x = element_text(angle = 60, hjust = 1), legend.position = "bottom") + # Rotate x-axi
  labs(y = "The Proportion of Unique Users by State", title = "Comparing The Proportion of Unique Users
```



Findings from the visualisations

- From the Average Review Star bar char, it indicates the all users share common feelings about the services they used which is 'normal' (around 3 stars). This is conclusion is consistent with the findings in the question 1. This is expected as they are technically analysed on the same datasets from the question 1.
- The attention is at the number of users per State. Nothing that the variable State is taken from the PGA and PGB dataset, not from the Users dataset. Thus, it does not mean the user is coming from that State. Instead, it should be understood as the number of unique users who visit businesses in that State.
- In 2 groups PGA and PGB, it appears that there are more unique users reviewing the businesses in group PGA than group PGB. Therefore, it is reasonable to see the outnumbers of reviews for business in group PGA and group PGB.
- From this insight, it could state that the businesses in group PGA are attracting more attention of users than the businesses in the other group. However, this conclusion does not deem that businesses in group PGA are having more visits than businesses in group PGB.
- On the other hand, based on the distribution of reviews by state, businesses in DC (in PGB) received the most reviews from unique users, while businesses in HI (in PGA) received the most in their group. This suggests that businesses in DC from PGB and in HI from PGA attracted the most attention within their respective groups.

Question 3:

3.1) Identify the top 10 users by the review count. For those top 10 users, calculate their average review stars.

```
# Identify the top 10 users using maindata
top10 <- maindata %>%
  group_by(user_id) %>%
  summarise(review_count = n_distinct(review_id)) %>% #count the total reviews per unique user
  arrange(desc(review_count)) %>%  #soft review_count from the biggest to the smallers
  slice_head(n = 10) # take the top 10
top10
```

```
## # A tibble: 10 x 2
##    user_id review_count
##    <chr>          <int>
##  1 u_27070           18
##  2 u_11551           15
##  3 u_6766            15
##  4 u_11229           14
##  5 u_14899           14
##  6 u_17629           14
##  7 u_22933           14
##  8 u_27907           14
##  9 u_29224           14
## 10 u_32335           14
```

```
# Use this list to identify their reviews in the maindata
reviews_of_Top10 <- maindata %>%
  semi_join(top10, by = "user_id")
```

```
# For those top 10 users, calculate their average review stars.
avg_Review_top10 <- reviews_of_Top10 %>%
  group_by(user_id) %>%
  summarise("Average Review Stars" = round(mean(stars), 3),
            "Review Count" = n_distinct(review_id)) %>%
  arrange(desc(`Review Count`))

table2 <- kable(avg_Review_top10, caption = "The summary of average figures of top 10: Before and After
  kable_styling(bootstrap_options = c("bordered"), position = "center")
table2
```

Table 2: The summary of average figures of top 10: Before and After 2020

user_id	Average Review Stars	Review Count
u_27070	2.833	18
u_11551	3.267	15
u_6766	3.267	15
u_11229	3.071	14

u_14899	2.571	14
u_17629	2.214	14
u_22933	2.929	14
u_27907	3.429	14
u_29224	3.500	14
u_32335	2.857	14

```
# Further analysis is conducted to identify the pattern of each users' behaviours
by_Star_top10 <- reviews_of_Top10 %>%
  group_by(user_id, stars) %>%
  summarise("Total Reviews per Star per Top10 User" = n())
```

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

```
table3 <- kable(by_Star_top10, caption = "The Number of Reviews per Star Per Top10 User") %>%
  kable_styling(bootstrap_options = c("bordered"), position = "center")
table3
```

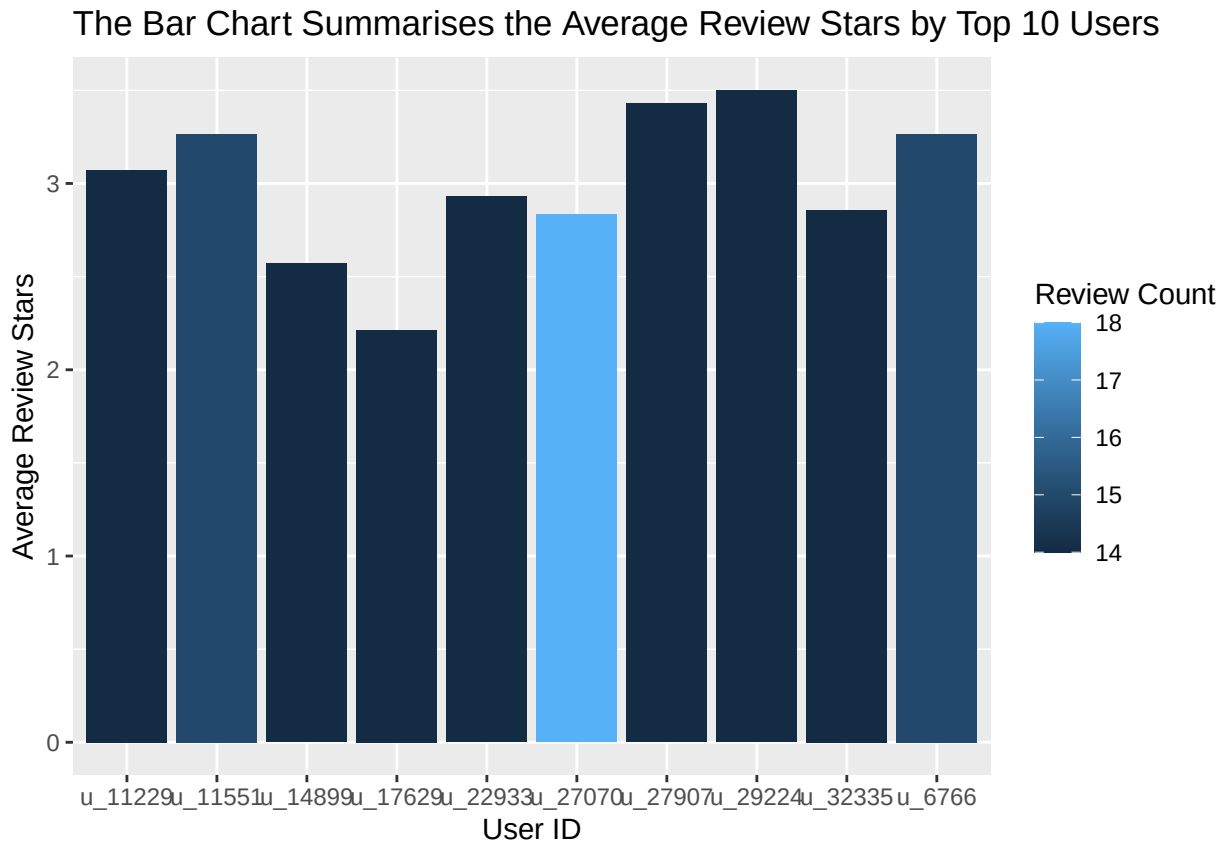
Table 3: The Number of Reviews per Star Per Top10 User

user_id	stars	Total Reviews per Star per Top10 User
u_11229	1	2
u_11229	2	3
u_11229	3	4
u_11229	4	2
u_11229	5	3
u_11551	1	2
u_11551	2	2
u_11551	3	4
u_11551	4	4
u_11551	5	3
u_14899	1	1
u_14899	2	7
u_14899	3	3
u_14899	4	3
u_17629	1	5
u_17629	2	5
u_17629	3	2
u_17629	5	2
u_22933	1	2
u_22933	2	2
u_22933	3	7
u_22933	4	1
u_22933	5	2
u_27070	1	4
u_27070	2	5
u_27070	3	3

u_27070	4	2
u_27070	5	4
u_27907	1	2
u_27907	2	2
u_27907	3	2
u_27907	4	4
u_27907	5	4
u_29224	1	2
u_29224	2	3
u_29224	3	1
u_29224	4	2
u_29224	5	6
u_32335	1	4
u_32335	2	2
u_32335	3	2
u_32335	4	4
u_32335	5	2
u_6766	1	4
u_6766	3	3
u_6766	4	4
u_6766	5	4

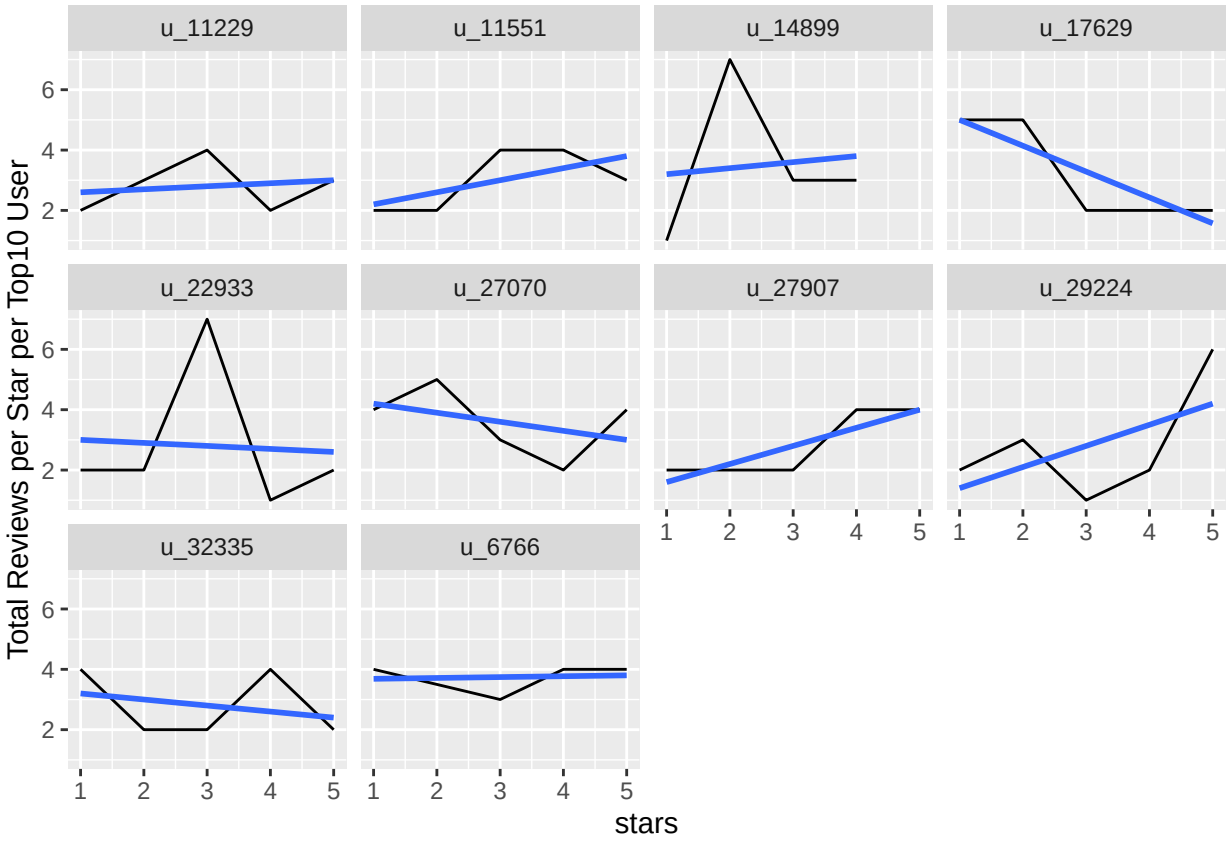
3.2) Visualise their rating distrubtion using ggplot2 - boxplot. Discuss your findings

```
ggplot(data = avg_Review_top10, mapping = aes(x = user_id ,y = `Average Review Stars`, fill = `Review C
  geom_col() +
  labs(x = "User ID", title = "The Bar Chart Summarises the Average Review Stars by Top 10 Users")
```




```
ggplot(data = by_Star_top10, mapping = aes(x = stars, y = `Total Reviews per Star per Top10 User`)) +
  geom_line() +
  geom_smooth(se = FALSE, method = 'lm', span = 1) +
  facet_wrap(~ user_id, nrow = 3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Findings

- The visual bar chart tells that in the top 10 users, the average review stars arranges in between 2 to 3 stars. It means that most of their experiences with at least 14 different businesses were acceptable to normal. This finding is continue to be consistent with two previous studies about the average review stars.
- With a closer look into the pattern of each user in the next visual, the number of upward trend to the right (more good overall experiences - higher stars, less bad overall experiences - lower stars) are not significant. Only the line of user u_29224 shows this trend in the most obvious way. Meanwhile, at least 4 upward trend to the left (less good overall experiences - higher stars, more bad overall experiences - lower stars) are seen (u_17629, u_27070 , u_22933 and u_32335). The rest shows an equal distributions in giving their stars.
- Overall, it appears that in the group of the most active users, they receive more negative experiences than positive experiences.

Question 4:

4.1) Write the code to analyse if there is a major difference between the review behavior of users who joined before and after 2020.For these 2 groups of users, compare their star rating behaviour and the length of the reviews (number of charaters in the review text).

```
# separate the maindata into 2 groups of users
## Reviews Before2020 (including 31.12.2019)
reviews_bf2020 <- maindata %>%
  filter(member_since < as.Date('2020-01-01')) %>%
```

```r
  mutate("Length of text" = nchar(text)) # Count the length of each review

## Reviews After2020 (including 01.01.2020)
reviews_af2020 <- maindata %>%
  filter(member_since >= as.Date('2020-01-01')) %>%
  mutate("Length of text" = nchar(text)) # Count the length of each review
```

```r
# Identify rating behaviour metrics - average review stars, average number of reviews per user, and the
## The Review Count, Average Review Star per User and Average Length of Reviews per user who joined bef
behaviour_bf2020 <- reviews_bf2020 %>%
  group_by(user_id)  %>%
  summarise("Review Count" = n(),
            "Average Review Star per User" = round(mean(stars),3),
            "Average Length of Reviews" = round(mean(`Length of text`),3))

## The Average Number of Reviews, Average Review Star and Average Length of Reviews of the whole group
avg_Figure_bf2020 <- behaviour_bf2020 %>%
  summarise("Average Number of Reviews" = round(mean(`Review Count`),3),
            "Average Review Star" = round(mean(`Average Review Star per User`),3),
            "Average Length of Reviews" = round(mean(`Average Length of Reviews`),3))

## The Review Count, Average Review Star and Average Length of Reviews per user who joined before 2020.
behaviour_af2020 <- reviews_af2020 %>%
  group_by(user_id)  %>%
  summarise("Review Count" = n(), "Average Review Star per User" = round(mean(stars),3), "Average Lengt

## The Average Number of Reviews, Average Review Star and Average Length of Reviews of the whole group
avg_Figure_af2020 <- behaviour_af2020 %>%
  summarise("Average Number of Reviews" = round(mean(`Review Count`),3),
            "Average Review Star" = round(mean(`Average Review Star per User`),3),
            "Average Length of Reviews" = round(mean(`Average Length of Reviews`),3))

table4 <- avg_Figure_bf2020 %>%
  bind_rows(avg_Figure_af2020) %>% #combine 2 dataframes
  mutate(Group = c("Before 2020", "After 2020")) %>% #create a new column to name the group
  relocate(Group) # move the column Group to the most left

kable(table4, caption = "The summary of average figures of 2 groups: Before and After 2020", digits = 3
  kable_styling(bootstrap_options = c("bordered"), position = "center")
```

Table 4: The summary of average figures of 2 groups: Before and After 2020

Group	Average Number of Reviews	Average Review Star	Average Length of Reviews
Before 2020	4.761	3.001	59.141
After 2020	4.743	3.005	59.076

```r
# Analyst the length of reviews by stars by using review_bf2020 and review_af2020 data.
## The Number of Review by Star and The Average Length of Review by Star in Group Before 2020
by_Star_bf2020 <- reviews_bf2020 %>%
  group_by(stars) %>%
```

```
  summarise("The Number of Review" = n(), "The Average Lenght of Review" = round(mean(`Length of text`)
  mutate(Group = "Before 2020") %>%
  relocate(Group)

by_Star_bf2020
```

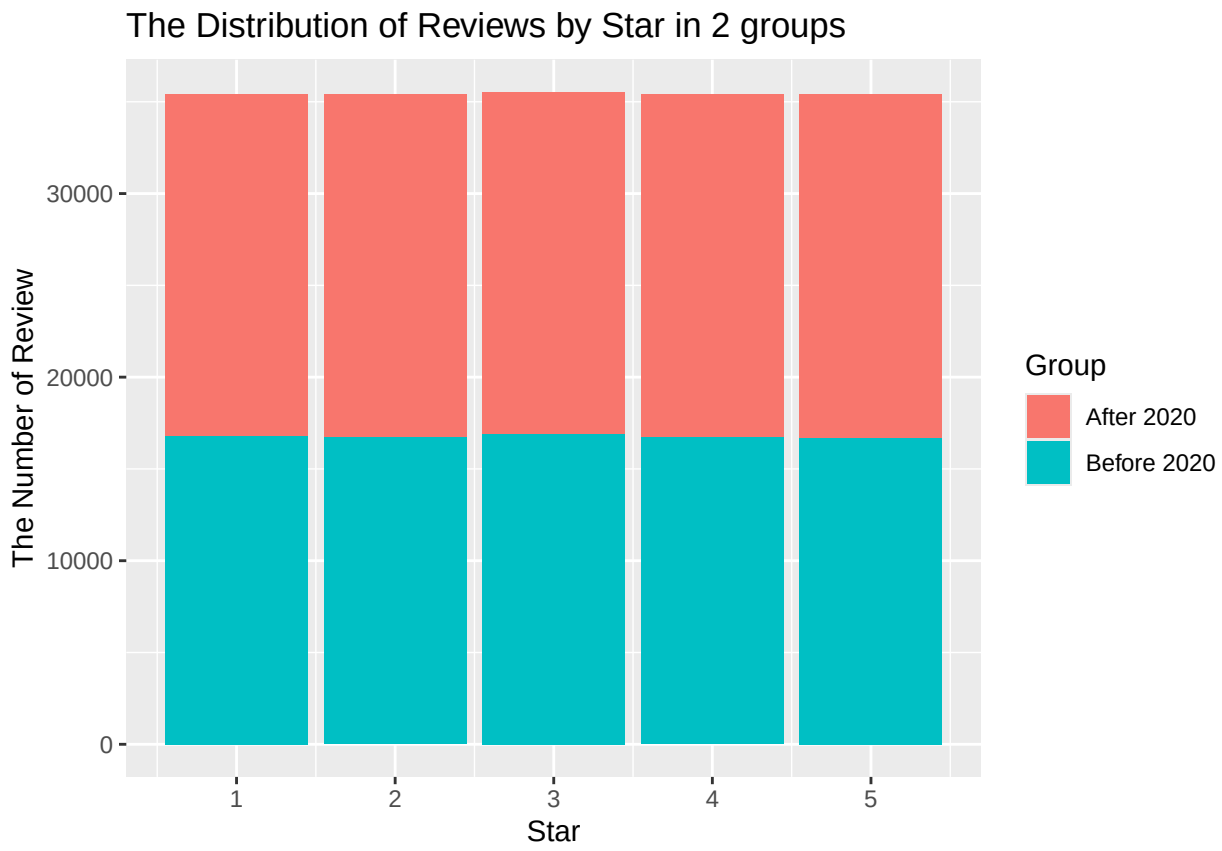
```
## # A tibble: 5 x 4
##   Group       stars `The Number of Review` `The Average Lenght of Review`
##   <chr>       <int>                  <int>                          <dbl>
## 1 Before 2020     1                  16804                           58.9
## 2 Before 2020     2                  16772                           59.0
## 3 Before 2020     3                  16906                           59.4
## 4 Before 2020     4                  16757                           59.1
## 5 Before 2020     5                  16698                           59.1
```

```
## The Number of Review by Star and The Average Length of Review by Star in Group After 2020
by_Star_af2020 <- reviews_af2020 %>%
  group_by(stars) %>%
  summarise("The Number of Review" = n(), "The Average Lenght of Review" = round(mean(`Length of text`)
  mutate(Group = "After 2020") %>%
  relocate(Group)

by_Star_af2020
```

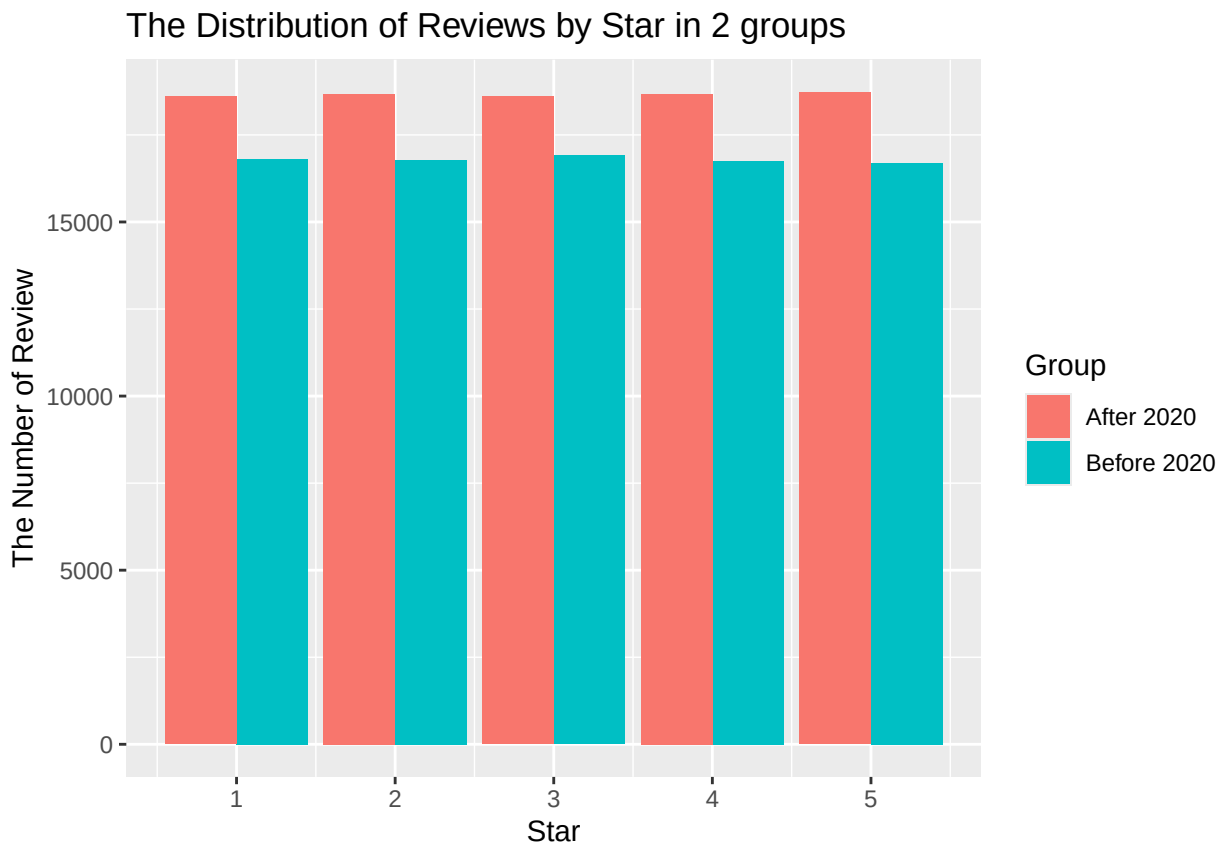
```
## # A tibble: 5 x 4
##   Group      stars `The Number of Review` `The Average Lenght of Review`
##   <chr>      <int>                  <int>                          <dbl>
## 1 After 2020     1                  18605                           59.0
## 2 After 2020     2                  18672                           59.2
## 3 After 2020     3                  18608                           58.6
## 4 After 2020     4                  18678                           59.1
## 5 After 2020     5                  18723                           58.9
```

4.2) Visualise the average review length by the two groups.

```
## Visualise by Star
# Combine 2 groups
by_Star <- by_Star_bf2020 %>%
  bind_rows(by_Star_af2020)

ggplot(data = by_Star, mapping = aes(x = stars, y= `The Number of Review`, fill = Group)) +
  geom_col() +
  labs(title = "The Distribution of Reviews by Star in 2 groups", x = "Star", y = "The Number of Review
```




```
ggplot(data = by_Star, mapping = aes(x = stars, y= `The Number of Review`, fill = Group)) +
  geom_col(position = "dodge") +
  labs(title = "The Distribution of Reviews by Star in 2 groups", x = "Star", y = "The Number of Review
```




```
ggplot(data = by_Star, mapping = aes(x = stars, y= `The Average Lenght of Review`, fill = Group)) +
  geom_col(position = "dodge") +
  labs(title = "The Distribution of Reviews by Star in 2 groups", x = "Star", y = "The Average Length o
```

Findings

- Between 2 groups, the number of reviews by star (from 1 star to 5 stars) in group After 2020 is higher than the figure in group Before 2020. It is an interesting insight to add on with the findings in question 1 because it could potentially indicate that even in the Intermediate group, the Intermediate users who joined after 2020 are more active than Intermediate users who joined before 2020.
- On the other hand, looking from the distribution perspective, the number of reviews in 2 groups are divided fairly equal across the stars (See the Distribution of Reviews by Star in 2 groups).
- Meanwhile, similarly, the average length of each review by star is around 57 to 59 in 2 groups, so that it could not deem that a low star reviews will be longer than a high star reviews or vice versa.

Conclusion - key highlights:

_ It is note that the average review stars across questions (from question 1 to question 4) is around 3 stars. Therefore, it could state that the overall experiences of users in this review sample is fairly 'average'. Moreover, in question 3, when reviewing the behaviours of the top 10 users in this sample, it is noted that they tend to have more negative experiences (more bad reviews - low stars than good reviews - high stars).
- Businesses in group PGA has the potential to increase the interactions of users when it has been proven that they received more total reviews than businesses in group PGB. Given that, businesses in group PGB that are based at DC are a more potential to boost interactions with users.
- In addition, the length of the reviews cannot help in indicating whether a review will receive a low-star review or a high-star review. In question 4, the analysis illustrates that the average length of the review by star are fairly equal.