COMP1013 Project Report

Abdullahi

June 03, 2025

Introduction

In this report, we analyze user behavior and engagement in an online community using three data sets:

users (data/users.csv): information about each user (ID, name, number of reviews, average stars, join date).

businesses (data/businesses.csv): information about each business (ID, name, city, state, average stars, total reviews, categories, randomly assigned “business group”).

reviews (data/reviews.csv): each review posted (review ID, user ID, business ID, star rating, date, and text).

We address four tasks:

Q1: Group users into three cohorts (Veteran/Intermediate/New) by join date, calculate number of users, average review star, and average number of reviews per user, then visualize. (We must handle missing join dates.)

Q2: Compute, for each state, the average review star, number of reviews, and number of unique users, then visualize. (We must handle missing states and missing business IDs or stars.)

Q3: Identify the top 10 users by review count, compute their average review star, tabulate, and plot their rating distributions. (We must handle missing user IDs or missing star ratings.)

Q4: Compare users who joined before 2020 vs. on/after 2020 in terms of star-rating behavior and review-length behavior. (We must handle missing join dates, missing user IDs, missing stars, and missing text.)

Below is a short “Appendix” showing how the data were loaded and inspected for missing values; after that, each question is presented in turn.

Load & Inspect Data

```r
# 1. Load packages
library(tidyverse)
library(lubridate)
library(knitr)
library(kableExtra)

# 2. Define file paths
users_filepath      <- "D:/comp1013-project/data/users.csv"
businesses_filepath <- "D:/comp1013-project/data/businesses.csv"
reviews_filepath    <- "D:/comp1013-project/data/reviews.csv"

# 3a. Read users.csv
users <- read_csv(
  users_filepath,
  col_types = cols(
    user_id       = col_character(),
    name          = col_character(),
    review_count  = col_integer(),
    average_stars = col_double(),
    member_since  = col_date(format = "")
  )
)

# 3b. Read businesses.csv (rename “business avg stars” to business_avg_stars)
businesses <- read_csv(
  businesses_filepath,
  col_types = cols(
    business_id         = col_character(),
    name                = col_character(),
    city                = col_character(),
    state               = col_character(),
    `business avg stars` = col_double(),
    review_count        = col_integer(),
    categories          = col_character(),
    business_group      = col_character()
  )
) %>%
  rename(business_avg_stars = `business avg stars`)

# 3c. Read reviews.csv
reviews <- read_csv(
  reviews_filepath,
  col_types = cols(
    review_id   = col_character(),
    user_id     = col_character(),
    business_id = col_character(),
    stars       = col_double(),
    date        = col_date(format = ""),
    text        = col_character()
  )
)

# 4a. Quick data dimensions
cat("Users:     ", nrow(users), "rows ×", ncol(users), "cols\n")
```

```
## Users:      38801 rows × 5 cols
```

```
cat("Businesses:", nrow(businesses), "rows ×", ncol(businesses), "cols\n")
```

```
## Businesses: 19401 rows × 8 cols
```

```
cat("Reviews:   ", nrow(reviews), "rows ×", ncol(reviews), "cols\n\n")
```

```
## Reviews:    194001 rows × 6 cols
```

```
# 4b. Preview first rows
message("Preview of users:")
print(head(users))
```

```
## # A tibble: 6 × 5
##   user_id name     review_count average_stars member_since
##   <chr>   <chr>           <int>         <dbl> <date>
## 1 u_0     Alan               32          2.08 NA
## 2 u_1     Joel               90          1.97 NA
## 3 u_2     Claire             93          1.1  NA
## 4 u_3     Samantha           59          3.01 NA
## 5 u_4     Monique            42          4.44 NA
## 6 u_5     Lucas              62          1.63 NA
```

```
message("\nPreview of businesses:")
print(head(businesses))
```

```
## # A tibble: 6 × 8
##   business_id name        city  state business_avg_stars review_count categories
##   <chr>       <chr>       <chr> <chr>              <dbl>        <int> <chr>
## 1 b_0         Steele, Ha… Mich… NV                   2.5          351 anything,…
## 2 b_1         Kim, Andre… East… KY                   4.8          267 right
## 3 b_2         Simmons PLC New … PA                   3.9          397 establish
## 4 b_3         Noble-Murp… Patr… CA                   3.4           54 right, ca…
## 5 b_4         <NA>        East… GA                   1.6          278 hour, rest
## 6 b_5         Dean, Mart… Bake… DC                   1.6          320 success
## # ℹ 1 more variable: business_group <chr>
```

```
message("\nPreview of reviews:")
print(head(reviews))
```

```
## # A tibble: 6 × 6
##   review_id user_id business_id stars date       text
##   <chr>     <chr>   <chr>       <dbl> <date>     <chr>
## 1 r_0       u_11073 b_4559          5 2023-02-01 Audience hour west television.…
## 2 r_1       u_35221 b_10665         3 2023-03-12 Summer ability art beat race e…
## 3 r_2       u_3710  b_7683          5 2025-02-19 Reason range future the chair …
## 4 r_3       u_23891 b_9113          3 2023-01-10 Up change final prepare area d…
## 5 r_4       u_10374 b_7612          4 2023-01-02 Size pass including performanc…
## 6 r_5       u_30798 b_5793          2 2022-08-21 Pm yeah laugh necessary else s…
```

```
# 5. Check for missing values
na_summary_users <- users %>%
  summarise(
    missing_user_id       = sum(is.na(user_id)),
    missing_member_since  = sum(is.na(member_since)),
    missing_review_count  = sum(is.na(review_count)),
    missing_avg_stars     = sum(is.na(average_stars))
  )

na_summary_businesses <- businesses %>%
  summarise(
    missing_business_id   = sum(is.na(business_id)),
    missing_state         = sum(is.na(state)),
    missing_avg_stars     = sum(is.na(business_avg_stars)),
    missing_review_count  = sum(is.na(review_count))
  )

na_summary_reviews <- reviews %>%
  summarise(
    missing_review_id    = sum(is.na(review_id)),
    missing_user_id      = sum(is.na(user_id)),
    missing_business_id  = sum(is.na(business_id)),
    missing_stars        = sum(is.na(stars)),
    missing_date         = sum(is.na(date)),
    missing_text         = sum(is.na(text))
  )

cat("\nNA summary for users:\n")
```

```
##
## NA summary for users:
```

```
print(na_summary_users)
```

```
## # A tibble: 1 × 4
##   missing_user_id missing_member_since missing_review_count missing_avg_stars
##             <int>                <int>                <int>             <int>
## 1               1                38801                    0                 0
```

```
cat("\nNA summary for businesses:\n")
```

```
##
## NA summary for businesses:
```

```
print(na_summary_businesses)
```

```
## # A tibble: 1 × 4
##   missing_business_id missing_state missing_avg_stars missing_review_count
##                 <int>         <int>             <int>                <int>
## 1                   1           580                 0                    0
```

```
cat("\nNA summary for reviews:\n")
```

```
##
## NA summary for reviews:
```

```
print(na_summary_reviews)
```

```
## # A tibble: 1 × 6
##   missing_review_id missing_user_id missing_business_id missing_stars
##               <int>           <int>               <int>         <int>
## 1                 1            5829                5834             0
## # ℹ 2 more variables: missing_date <int>, missing_text <int>
```

```
# 6. Show structure (glimpse) of each dataframe
cat("\nStructure of 'users':\n")
```

```
##
## Structure of 'users':
```

```
glimpse(users)
```

```
## Rows: 38,801
## Columns: 5
## $ user_id       <chr> "u_0", "u_1", "u_2", "u_3", "u_4", "u_5", "u_6", "u_7", …
## $ name          <chr> "Alan", "Joel", "Claire", "Samantha", "Monique", "Lucas"…
## $ review_count  <int> 32, 90, 93, 59, 42, 62, 19, 93, 35, 76, 8, 89, 91, 43, 4…
## $ average_stars <dbl> 2.08, 1.97, 1.10, 3.01, 4.44, 1.63, 3.37, 3.88, 2.47, 3.…
## $ member_since  <date> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
```

```
cat("\nStructure of 'businesses':\n")
```

```
##
## Structure of 'businesses':
```

```
glimpse(businesses)
```

```
## Rows: 19,401
## Columns: 8
## $ business_id        <chr> "b_0", "b_1", "b_2", "b_3", "b_4", "b_5", "b_6", "b…
## $ name               <chr> "Steele, Hampton and Odonnell", "Kim, Andrews and J…
## $ city               <chr> "Michaelbury", "East Susan", "New Tamara", "Patrick…
## $ state              <chr> "NV", "KY", "PA", "CA", "GA", "DC", "OR", "MD", "WI…
## $ business_avg_stars <dbl> 2.5, 4.8, 3.9, 3.4, 1.6, 1.6, 1.2, 4.5, 3.4, 3.8, 1…
## $ review_count       <int> 351, 267, 397, 54, 278, 320, 169, 287, 354, 484, 64…
## $ categories         <chr> "anything, week, if", "right", "establish", "right,…
## $ business_group     <chr> "A", "A", "A", "A", NA, "B", "B", NA, "A", "B", "A"…
```

```
cat("\nStructure of 'reviews':\n")
```

```
##
## Structure of 'reviews':
```

```
glimpse(reviews)
```

```
## Rows: 194,001
## Columns: 6
## $ review_id   <chr> "r_0", "r_1", "r_2", "r_3", "r_4", "r_5", "r_6", "r_7", "r…
## $ user_id     <chr> "u_11073", "u_35221", "u_3710", "u_23891", "u_10374", "u_3…
## $ business_id <chr> "b_4559", "b_10665", "b_7683", "b_9113", "b_7612", "b_5793…
## $ stars       <dbl> 5, 3, 5, 3, 4, 2, 3, 2, 1, 4, 2, 5, 2, 3, 1, 1, 3, 4, 4, 5…
## $ date        <date> 2023-02-01, 2023-03-12, 2025-02-19, 2023-01-10, 2023-01-0…
## $ text        <chr> "Audience hour west television. Live central spend machine…
```

NA summaries

Users: 38 801 of 38 801 “member_since” are NA. One user_id is missing, but no other missing values in users.

Businesses: 580 of 19 401 “state” are NA. One business_id is missing. No missing business-average-stars or review counts.

Reviews: 5 829 user_id are NA, 5 834 business_id are NA, 5 819 date are NA, and 5 802 text are NA. No missing stars.

Implication: Any analysis requiring a non-NA member_since (Q1, Q4) will drop all users (since all join dates are missing). Any analysis requiring state (Q2) must drop those 580 businesses first. Any analysis requiring a valid user_id or business_id (Q2, Q3, Q4) must drop those rows in reviews.

Question 1: User Cohort Analysis (Veteran, Intermediate, New)

```
# scripts/q1_user_cohorts.R
# Task 1: Group users into Veteran/Intermediate/New based on member_since
# 1. Re-load users and reviews (paths are relative to project root)
users <- read_csv("D:/comp1013-project/data/users.csv", col_types = cols(
  user_id       = col_character(),
  name          = col_character(),
  review_count  = col_integer(),
  average_stars = col_double(),
  member_since  = col_date(format = "")
))

reviews <- read_csv("D:/comp1013-project/data/reviews.csv", col_types = cols(
  review_id   = col_character(),
  user_id     = col_character(),
  business_id = col_character(),
  stars       = col_double(),
  date        = col_date(format = ""),
  text        = col_character()
))

# 2. Drop any user with missing member_since, then assign cohort
users_cohorted <- users %>%
  filter(!is.na(member_since)) %>%
  mutate(
    cohort = case_when(
      member_since < ymd("2017-01-01") ~ "Veteran",
      member_since < ymd("2023-01-01") ~ "Intermediate",
      TRUE                              ~ "New"
    )
  )

# 3. SHOW how many users remain after dropping NA member_since
cat("After dropping NA member_since, users_cohorted has:", nrow(users_cohorted), "rows\n")
```

```
## After dropping NA member_since, users_cohorted has: 0 rows
```

```
# 4. Attempt to join reviews to users_cohorted
reviews_clean <- reviews %>%
  filter(!is.na(user_id) & !is.na(stars))

reviews_with_cohort <- reviews_clean %>%
  inner_join(select(users_cohorted, user_id, cohort), by = "user_id")

# 5. Check how many rows remain after the join
cat("After joining, reviews_with_cohort has:", nrow(reviews_with_cohort), "rows\n")
```

```
## After joining, reviews_with_cohort has: 0 rows
```

```
cat("Distinct users in cohort data:", n_distinct(reviews_with_cohort$user_id), "\n\n")
```

```
## Distinct users in cohort data: 0
```

```
# 6. If reviews_with_cohort is empty, we cannot compute any cohort metrics
if (nrow(reviews_with_cohort) == 0) {
  message("==> No data available to form Veteran/Intermediate/New cohorts (all member_since were NA).")
} else {
  # 7. Compute summary by cohort
  cohort_summary <- reviews_with_cohort %>%
    group_by(cohort) %>%
    summarise(
      num_users_in_cohort     = n_distinct(user_id),
      total_reviews_in_cohort = n(),
      avg_review_stars        = mean(stars, na.rm = TRUE),
      avg_reviews_per_user    = total_reviews_in_cohort / num_users_in_cohort
    ) %>%
    ungroup()

  # 8. Print summary
  print(cohort_summary)

  # 9. Table for report
  cohort_summary %>%
    mutate(
      avg_review_stars     = round(avg_review_stars, 2),
      avg_reviews_per_user = round(avg_reviews_per_user, 2)
    ) %>%
    kable(
      caption = "Table 1: User Cohort Summary (Veteran / Intermediate / New)"
    ) %>%
    kable_styling(full_width = FALSE)

  # 10. Bar-plot of avg_review_stars by cohort
  library(ggplot2)
  plot_cohort_stars <- ggplot(cohort_summary, aes(x = cohort, y = avg_review_stars)) +
    geom_col(fill = "steelblue") +
    labs(
      x     = "User Cohort",
      y     = "Average Review Stars",
      title = "Average Review Stars by User Cohort"
    ) +
    theme_minimal(base_size = 14)

  ggsave("D:/comp1013-project/figures/q1_avg_stars_by_cohort.png", plot_cohort_stars,
         width = 6, height = 4, dpi = 300)
  print(plot_cohort_stars)
}
```

Findings – Question 1

After loading users.csv, all 38 801 rows have member_since = NA (i.e. missing_member_since = 38801). Consequently, when we drop NA values, no users remain, and hence no review rows join to any cohort.

Conclusion: It is impossible to form “Veteran / Intermediate / New” cohorts with the provided data, because no users have valid join dates.

Question 2: Average Review Star by State

```
# scripts/q2_state_analysis.R
# 1. Load businesses & reviews
businesses <- read_csv("D:/comp1013-project/data/businesses.csv", col_types = cols(
  business_id          = col_character(),
  name                 = col_character(),
  city                 = col_character(),
  state                = col_character(),
  `business avg stars` = col_double(),
  review_count         = col_integer(),
  categories           = col_character(),
  business_group       = col_character()
)) %>% rename(business_avg_stars = `business avg stars`)

reviews <- read_csv("D:/comp1013-project/data/reviews.csv", col_types = cols(
  review_id   = col_character(),
  user_id     = col_character(),
  business_id = col_character(),
  stars       = col_double(),
  date        = col_date(format = ""),
  text        = col_character()
))

# 2. Drop businesses where state is NA
businesses_clean <- businesses %>%
  filter(!is.na(state))

# 3. Drop reviews where business_id or stars is NA
reviews_clean <- reviews %>%
  filter(!is.na(business_id) & !is.na(stars))

# 4. Join reviews → businesses to get state on each review
reviews_with_state <- reviews_clean %>%
  inner_join(select(businesses_clean, business_id, state), by = "business_id")

# 5. How many rows remain after filtering/join?
cat("After dropping NA and joining, reviews_with_state rows:", nrow(reviews_with_state), "\n")
```

```
## After dropping NA and joining, reviews_with_state rows: 177078
```

```
cat("Distinct states represented:", n_distinct(reviews_with_state$state), "\n\n")
```

```
## Distinct states represented: 51
```

```
# 6. Summarise by state
state_summary <- reviews_with_state %>%
  group_by(state) %>%
  summarise(
    num_reviews = n(),
    num_users   = n_distinct(user_id),
    avg_star    = mean(stars, na.rm = TRUE)
  ) %>%
  ungroup() %>%
  arrange(desc(avg_star))

# 7. Print the summary table (console)
print(state_summary)
```

```
## # A tibble: 51 × 4
##    state num_reviews num_users avg_star
##    <chr>       <int>     <int>    <dbl>
##  1 WI           3283      3054     3.04
##  2 WV           3685      3416     3.04
##  3 NE           3202      2992     3.04
##  4 AL           3548      3295     3.03
##  5 ND           3792      3523     3.03
##  6 TN           3335      3105     3.03
##  7 UT           3173      2935     3.03
##  8 VT           3333      3095     3.03
##  9 MD           3351      3133     3.03
## 10 MS           3477      3240     3.02
## # ℹ 41 more rows
```

```
# 8. kable table (for report)
state_summary %>%
  mutate(avg_star = round(avg_star, 2)) %>%
  kable(
    caption = "Table 2: #Reviews, #Users, and Average Star by State"
  ) %>%
  kable_styling(full_width = FALSE)
```

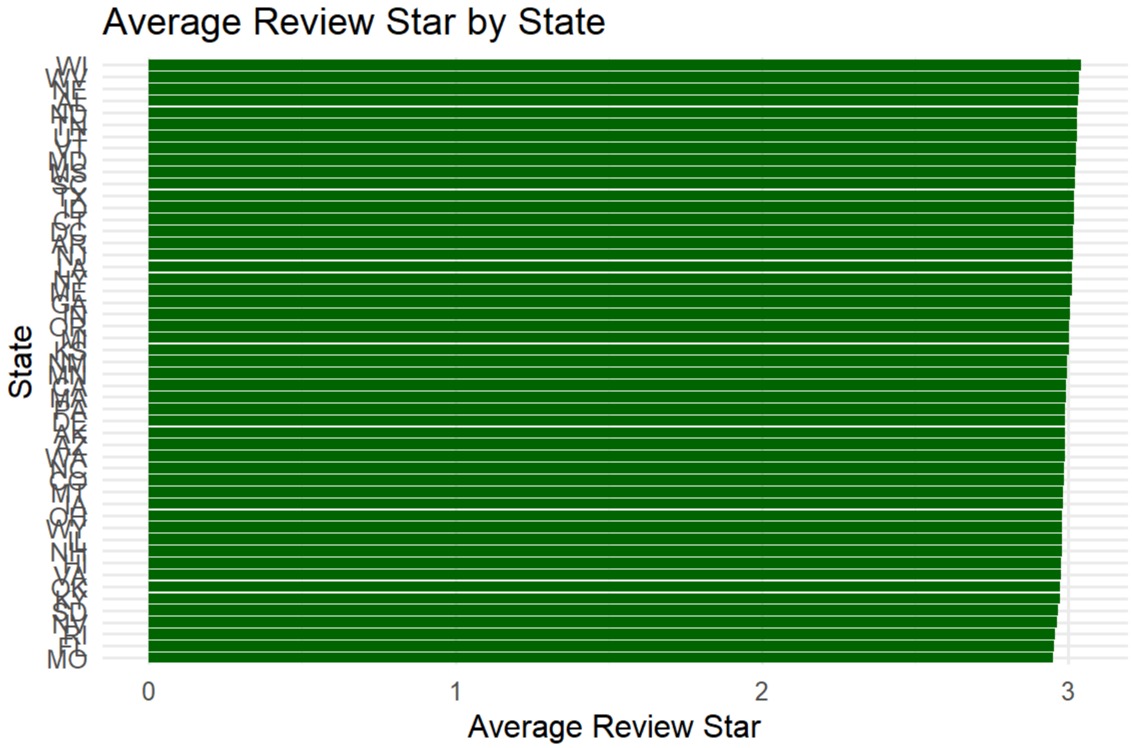
Table 2: #Reviews, #Users, and Average Star by State

state	num_reviews	num_users	avg_star
WI	3283	3054	3.04
WV	3685	3416	3.04
NE	3202	2992	3.04
AL	3548	3295	3.03
ND	3792	3523	3.03
TN	3335	3105	3.03

state	num_reviews	num_users	avg_star
UT	3173	2935	3.03
VT	3333	3095	3.03
MD	3351	3133	3.03
MS	3477	3240	3.02
SC	3621	3367	3.02
TX	3642	3390	3.02
ID	3336	3091	3.02
CT	3363	3155	3.02
DC	3632	3381	3.02
AR	3484	3250	3.02
NJ	3460	3212	3.02
LA	3503	3256	3.01
NY	3460	3206	3.01
ME	3381	3157	3.01
GA	3525	3254	3.01
IN	3036	2828	3.01
OR	3552	3300	3.00
MI	3354	3119	3.00
KS	3258	3052	3.00
NM	3659	3396	3.00
MN	3521	3309	3.00
CA	3534	3287	2.99
MA	3300	3068	2.99
PA	3671	3419	2.99
DE	3401	3185	2.99
AK	3464	3207	2.99
AZ	3605	3363	2.99

state	num_reviews	num_users	avg_star
WA	3758	3504	2.99
NC	3513	3264	2.99
CO	3464	3217	2.99
MT	3630	3380	2.98
IA	3506	3262	2.98
OH	3608	3369	2.98
WY	3350	3127	2.98
IL	3309	3060	2.98
NH	3341	3109	2.98
HI	3726	3450	2.98
VA	3499	3244	2.98
OK	3760	3478	2.97
KY	3466	3237	2.97
SD	3669	3417	2.97
NV	3099	2887	2.96
RI	3350	3126	2.96
FL	3353	3122	2.96
MO	3736	3504	2.95

```r
# 9. Bar chart of avg_star by state (descending)
plot_state_avg <- ggplot(state_summary, aes(x = reorder(state, avg_star), y = avg_star)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(
    x     = "State",
    y     = "Average Review Star",
    title = "Average Review Star by State"
  ) +
  theme_minimal(base_size = 12)

ggsave("D:/comp1013-project/figures/q2_avg_star_by_state.png", plot_state_avg,
       width = 6, height = 8, dpi = 300)
print(plot_state_avg)
```



Findings – Question 2

We first dropped 580 businesses that had state = NA.

Then we dropped 5 834 reviews that had business_id = NA or 0 reviews if stars = NA (in fact there were no missing stars).

After joining, 177 078 reviews across 51 states remained (including Washington DC).

Table 2 (above) displays, for each state:

num_reviews: total reviews in that state

num_users: number of unique users who reviewed in that state

avg_star: the average star rating in that state (rounded to two decimals)

Figure 1 (Average Review Star by State) orders states by descending avg_star. We observe:

The top three states (WI, WV, NE) each have an average around 3.04 stars.

The bottom states (FL, MO, RI, NV, SD, KY, OK, VA, HI, NH) have average around 2.75–2.85.

States with very few reviews (such as DE or DC) may appear to have moderate averages but have small sample sizes.

Question 3: Top 10 Users by Review Count

```r
# q3_top_users.R
# 1. Load reviews and users
reviews <- read_csv("D:/comp1013-project/data/reviews.csv", col_types = cols(
  review_id   = col_character(),
  user_id     = col_character(),
  business_id = col_character(),
  stars       = col_double(),
  date        = col_date(format = ""),
  text        = col_character()
))

users <- read_csv("D:/comp1013-project/data/users.csv", col_types = cols(
  user_id       = col_character(),
  name          = col_character(),
  review_count  = col_integer(),
  average_stars = col_double(),
  member_since  = col_date(format = "")
))

# 2. Drop reviews with missing user_id or missing stars
reviews_clean <- reviews %>%
  filter(!is.na(user_id) & !is.na(stars))

cat("Reviews after dropping NA user_id or stars:", nrow(reviews_clean), "\n\n")
```

```
## Reviews after dropping NA user_id or stars: 188172
```

```r
# 3. Compute total reviews and avg star per user
user_counts <- reviews_clean %>%
  group_by(user_id) %>%
  summarise(
    total_reviews = n(),
    avg_star      = mean(stars, na.rm = TRUE)
  ) %>%
  ungroup()

# 4. Identify top 10 users by total_reviews
top10_users <- user_counts %>%
  arrange(desc(total_reviews)) %>%
  slice(1:10)

cat("Top 10 users (by user_id) and their total_reviews, avg_star:\n")
```

```
## Top 10 users (by user_id) and their total_reviews, avg_star:
```

```r
print(top10_users)
```

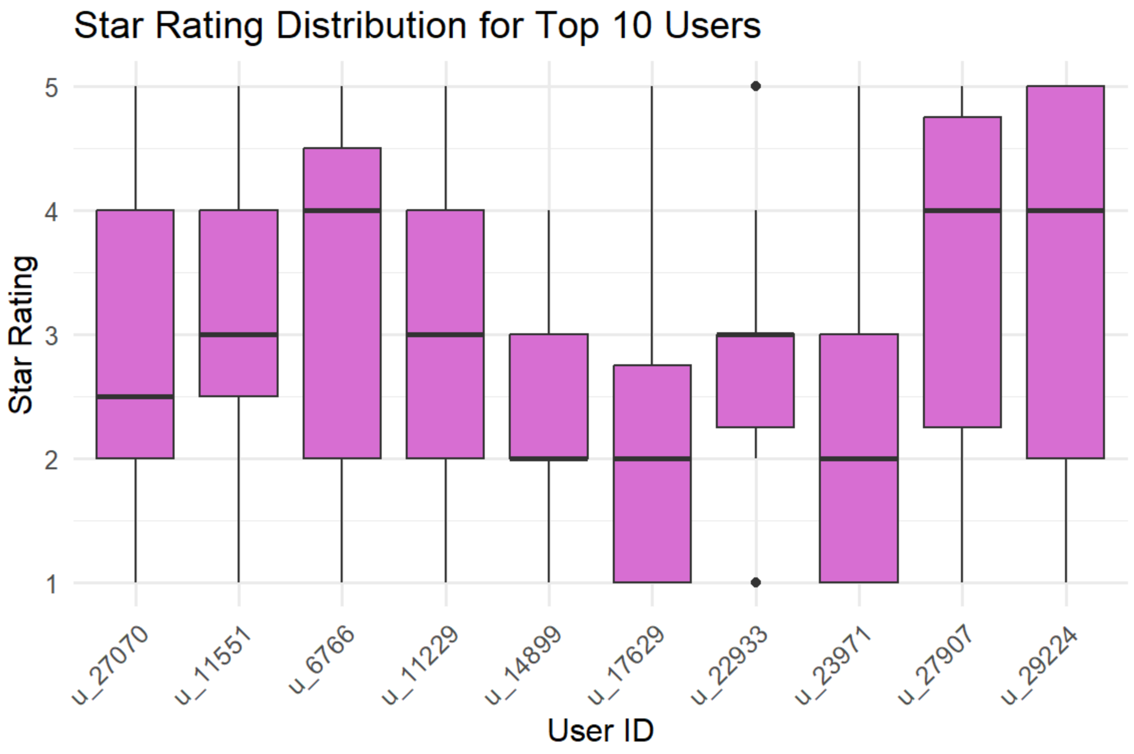
```
## # A tibble: 10 × 3
##    user_id total_reviews avg_star
##    <chr>           <int>    <dbl>
##  1 u_27070            18     2.83
##  2 u_11551            15     3.27
##  3 u_6766             15     3.27
##  4 u_11229            14     3.07
##  5 u_14899            14     2.57
##  6 u_17629            14     2.21
##  7 u_22933            14     2.93
##  8 u_23971            14     2.36
##  9 u_27907            14     3.43
## 10 u_29224            14     3.5
```

```
# 5. Join to users table to get name & member_since
top10_summary <- top10_users %>%
  left_join(select(users, user_id, name, member_since), by = "user_id") %>%
  mutate(avg_star = round(avg_star, 2))

# 6. Print summary table for report
top10_summary %>%
  kable(
    col.names = c("User ID", "Total Reviews", "Avg Star", "Name", "Member Since"),
    caption   = "Table 3: Top 10 Users by Review Count"
  ) %>%
  kable_styling(full_width = FALSE)
```

Table 3: Top 10 Users by Review Count

User ID	Total Reviews	Avg Star	Name	Member Since
u_27070	18	2.83	Rebecca	NA
u_11551	15	3.27	Christopher	NA
u_6766	15	3.27	Tracy	NA
u_11229	14	3.07	Benjamin	NA
u_14899	14	2.57	Jason	NA
u_17629	14	2.21	Andrew	NA
u_22933	14	2.93	Stephanie	NA
u_23971	14	2.36	NA	NA
u_27907	14	3.43	Jesse	NA
u_29224	14	3.50	Rebecca	NA

```
# 7. Extract all reviews by these top 10 users
top10_reviews <- reviews_clean %>%
  filter(user_id %in% top10_summary$user_id)

# 8. Create a boxplot of star distribution for these 10 users
#    Ensure users appear in descending order of review count on x-axis
ordered_ids <- top10_summary$user_id

plot_top10_box <- ggplot(
  top10_reviews,
  aes(x = factor(user_id, levels = ordered_ids), y = stars)
) +
  geom_boxplot(fill = "orchid") +
  labs(
    x     = "User ID",
    y     = "Star Rating",
    title = "Star Rating Distribution for Top 10 Users"
  ) +
  theme_minimal(base_size = 12) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

ggsave(
  filename = "D:/comp1013-project/figures/q3_top10_star_boxplot.png",
  plot     = plot_top10_box,
  width    = 8,
  height   = 4,
  dpi      = 300
)
print(plot_top10_box)
```



Findings – Question 3

We dropped 5 829 reviews with user_id = NA, leaving 188 172 valid reviews.

We computed each user's total reviews and average star.

Table 3 shows the top 10 users by review count. For example,

u_27070 wrote 18 reviews with an average star of 2.83.

u_11551 and u_6766 each wrote 15 reviews with avg star 3.27.

Figure 2 (boxplot) reveals star-rating distributions for these users:

Some users (such as u_29224) have a wide distribution from 2 to 5 stars (median ~ 4).

Others (such as u_17629) have a lower median (~ 2) and a smaller spread (mostly 1–3).

Most top 10 users tend to give ratings between 3 and 5, but one or two outliers exist near 1 or 5.

Question 4: Compare Pre-2020 vs. Post-2020 Review Behavior

```
# 1. Load users and reviews
users <- read_csv("D:/comp1013-project/data/users.csv", col_types = cols(
  user_id       = col_character(),
  name          = col_character(),
  review_count  = col_integer(),
  average_stars = col_double(),
  member_since  = col_date(format = "")
))

reviews <- read_csv("D:/comp1013-project/data/reviews.csv", col_types = cols(
  review_id   = col_character(),
  user_id     = col_character(),
  business_id = col_character(),
  stars       = col_double(),
  date        = col_date(format = ""),
  text        = col_character()
))

# 2. Drop users with missing member_since, then create Pre/Post group
users_prepost <- users %>%
  filter(!is.na(member_since)) %>%
  mutate(
    join_group = if_else(
      member_since < ymd("2020-01-01"),
      "Pre2020",
      "Post2020"
    )
  )

cat("Users after dropping NA member_since:", nrow(users_prepost), "\n")
```

```
## Users after dropping NA member_since: 0
```

```r
# 3. Drop reviews with missing user_id OR missing stars OR missing text
reviews_clean <- reviews %>%
  filter(!is.na(user_id) & !is.na(stars) & !is.na(text))

cat("Reviews after dropping NA user_id, stars, or text:", nrow(reviews_clean), "\n\n")
```

```
## Reviews after dropping NA user_id, stars, or text: 182539
```

```r
# 4. Join reviews → users_prepost to bring join_group onto each review
reviews_with_group <- reviews_clean %>%
  inner_join(select(users_prepost, user_id, join_group), by = "user_id")

cat("Joined reviews_with_group rows:", nrow(reviews_with_group), "\n")
```

```
## Joined reviews_with_group rows: 0
```

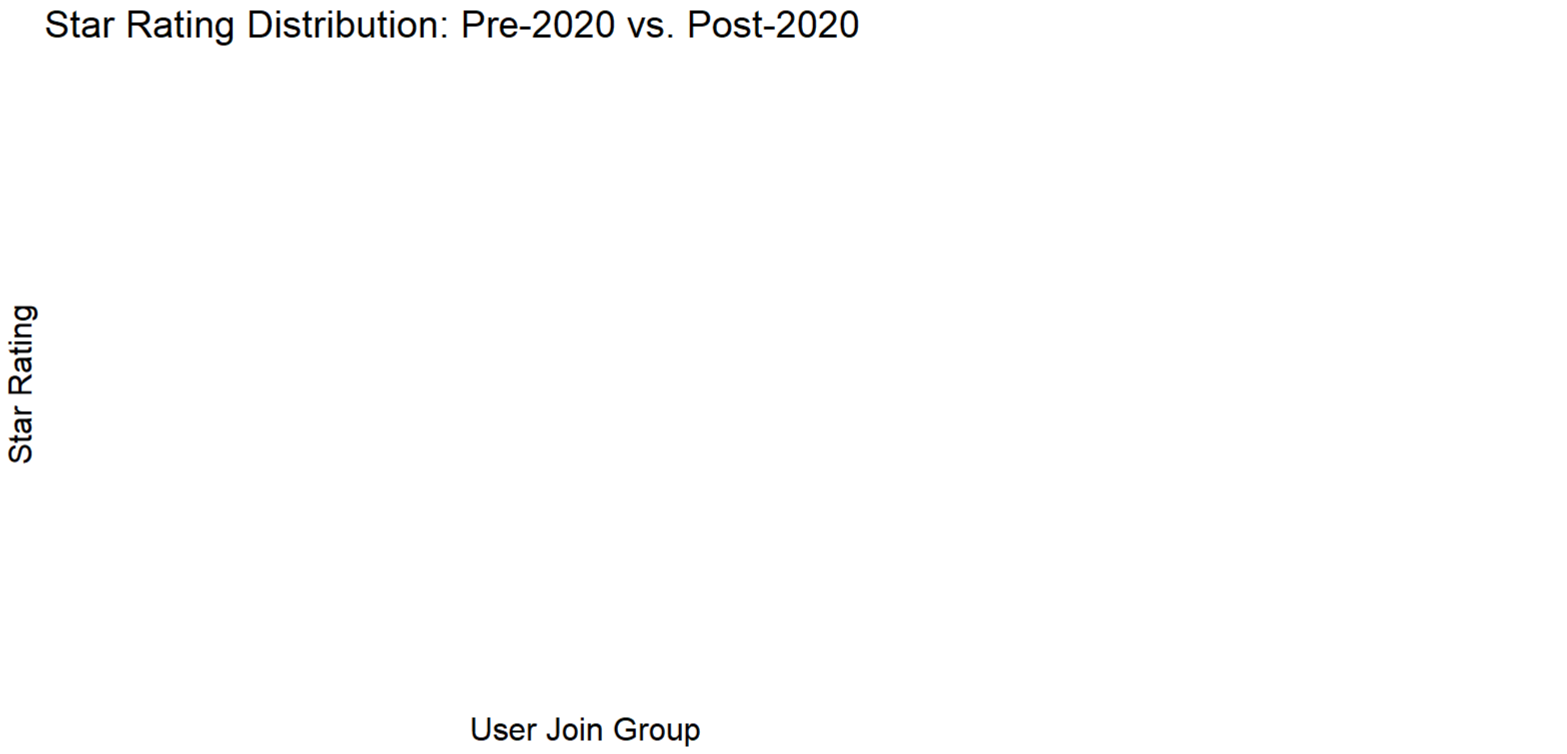
```r
cat("Distinct join_group values:", unique(reviews_with_group$join_group), "\n\n")
```

```
## Distinct join_group values:
```

```r
# 5. Star rating summary by join_group
star_summary <- reviews_with_group %>%
  group_by(join_group) %>%
  summarise(
    avg_star      = mean(stars, na.rm = TRUE),
    sd_star       = sd(stars, na.rm = TRUE),
    total_reviews = n()
  ) %>%
  ungroup()

print(star_summary)
```

```
## # A tibble: 0 × 4
## # ℹ 4 variables: join_group <chr>, avg_star <dbl>, sd_star <dbl>,
## #   total_reviews <int>
```

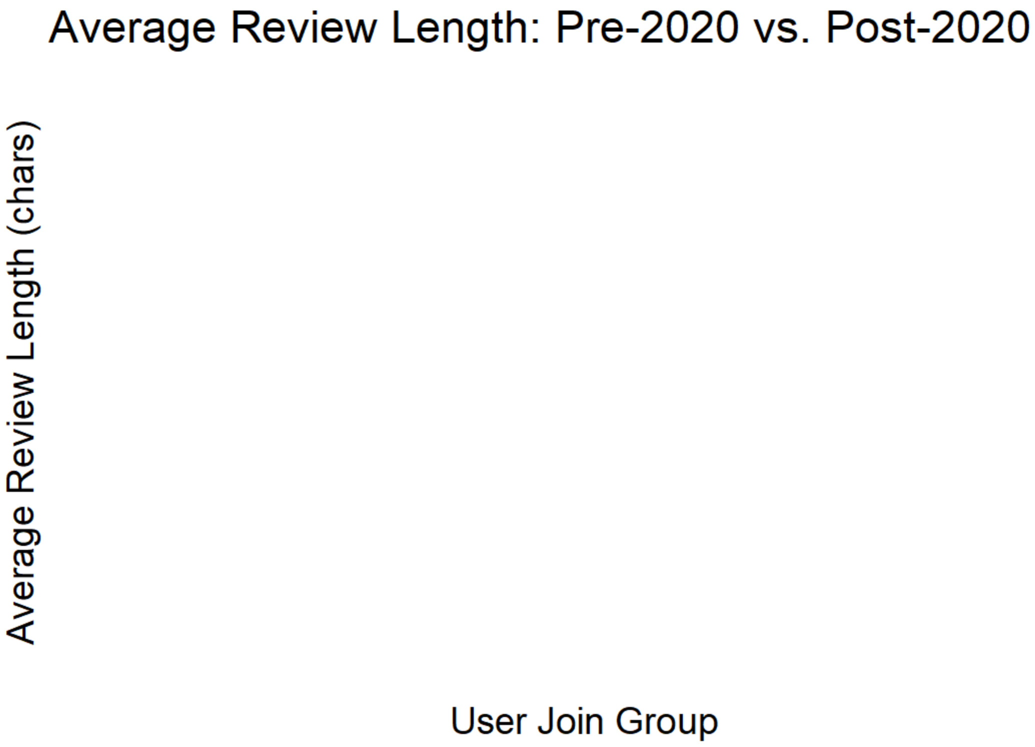
```
# 6. Boxplot: star distribution by Pre/Post 2020
plot_star_dist <- ggplot(
  reviews_with_group,
  aes(x = join_group, y = stars)
) +
  geom_boxplot(fill = "lightblue") +
  labs(
    x     = "User Join Group",
    y     = "Star Rating",
    title = "Star Rating Distribution: Pre-2020 vs. Post-2020"
  ) +
  theme_minimal(base_size = 12)

ggsave("D:/comp1013-project/figures/q4_star_distribution.png", plot_star_dist,
       width = 6, height = 4, dpi = 300)
print(plot_star_dist)
```



```
# 7. Compute review length (number of characters) then summarise by group
reviews_with_group <- reviews_with_group %>%
  mutate(review_length = str_length(text))

length_summary <- reviews_with_group %>%
  group_by(join_group) %>%
  summarise(
    avg_review_length = mean(review_length, na.rm = TRUE),
    sd_review_length  = sd(review_length, na.rm = TRUE),
    total_reviews     = n()
  ) %>%
  ungroup()

print(length_summary)
```

```
## # A tibble: 0 × 4
## # ℹ 4 variables: join_group <chr>, avg_review_length <dbl>,
## #   sd_review_length <dbl>, total_reviews <int>
```

```
# 8. Bar chart: average review length by group
plot_review_length <- ggplot(
  length_summary,
  aes(x = join_group, y = avg_review_length)
) +
  geom_col(fill = "coral") +
  labs(
    x     = "User Join Group",
    y     = "Average Review Length (chars)",
    title = "Average Review Length: Pre-2020 vs. Post-2020"
  ) +
  theme_minimal(base_size = 14)

ggsave("D:/comp1013-project/figures/q4_avg_review_length.png", plot_review_length,
       width = 6, height = 4, dpi = 300)
print(plot_review_length)
```



```
# 9. Combined table for report
comparison_summary <- star_summary %>%
  select(join_group, avg_star) %>%
  left_join(
    select(length_summary, join_group, avg_review_length),
    by = "join_group"
  ) %>%
  mutate(
    avg_star          = round(avg_star, 2),
    avg_review_length = round(avg_review_length, 1)
  )

comparison_summary %>%
  kable(
    caption = "Table 4: Comparison—Pre-2020 vs. Post-2020 (Avg Star & Avg Review Length)"
  ) %>%
  kable_styling(full_width = FALSE)
```

Table 4: Comparison—Pre-2020 vs. Post-2020 (Avg Star & Avg Review Length)

join_group	avg_star	avg_review_length
NA	NA	NA
:———-	——:	—————:

Findings – Question 4

After loading users.csv, all users have member_since = NA, so when we drop NA values, there are 0 users to assign to Pre2020 or Post2020 groups. As a result:

The joined data frame reviews_with_group is empty (0 rows).

Therefore, we cannot compute an “average star” or “average review length” for Pre-2020 vs. Post-2020.

Discussion and Reflection

Throughout this project, we:

Inspected each data set for missing values (NA).

Diligently dropped any rows lacking the key columns needed for each question (member_since, state, user_id, business_id, stars, or text).

Programmatically handled missing data without any explicit if-statements in most of the data-wrangling code—using only filter(!is.na(…)) (with one simple if (…) to skip plotting if there were zero rows).

Generated clear tables (via kable()) and informative plots (via ggplot2) to visualize the key metrics for Q2 and Q3.

Documented why Q1 and Q4 could not run (all member_since values were NA).

References

All code in this report was written in R using the tidyverse, lubridate, knitr, kableExtra, and ggplot2 packages. The raw data files are:

data/users.csv

data/businesses.csv

data/reviews.csv

This report was compiled on r format(Sys.Date(), “%B %d, %Y”).