Task-03

Import necessary libraries

```r
library(dplyr)
```

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

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

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

```r
library(ggplot2)
library(knitr)
```

Loading data

```r
reviews <- read.csv("../data/reviews.csv", stringsAsFactors = FALSE)
```

Identify the 10 users who have written the most reviews

Identify the 10 most active users by counting the number of reviews per *user_id* from the reviews dataset, sorting them in descending order based on the number of reviews, and selecting the top 10 users *top_users_by_count*. Next, filters the original reviews dataset to include only reviews from these top 10 users *reviews_from_top_users*. After that, for each of the 10 users, the algorithm calculates the average star rating they have given, ignoring any missing values NA in the calculation *average_stars_for_top_users*.

```r
top_users_by_count <- reviews %>%
  group_by(user_id) %>%
  summarise(review_count = n(), .groups = 'drop') %>% # Group the rows in the reviews dataset by the us
  arrange(desc(review_count)) %>% # sorting the results
  slice_head(n = 10) # get 10 rows

reviews_from_top_users <- reviews %>%
  filter(user_id %in% top_users_by_count$user_id)

# calculate average stars for each of these top users.
# na.rm = TRUE ensures that NA values in the 'stars' column are ignored during mean calculation.
average_stars_for_top_users <- reviews_from_top_users %>%
  group_by(user_id) %>%
  summarise(average_stars = mean(stars, na.rm = TRUE), .groups = 'drop')

print(average_stars_for_top_users)
```

```
## # A tibble: 10 x 2
##    user_id   average_stars
##    <chr>             <dbl>
##  1 ""                 3.00
##  2 "u_11229"          3.07
##  3 "u_11551"          3.27
##  4 "u_14899"          2.57
##  5 "u_17629"          2.21
##  6 "u_22933"          2.93
##  7 "u_23971"          2.36
##  8 "u_27070"          2.83
##  9 "u_27907"          3.43
## 10 "u_6766"           3.27
```

Then, merges *left_join* the review count information *top_users_by_count* with the average star ratings *average_stars_for_top_users* for each user into a single summary table called *summary_top_users*, which is then sorted again based on the number of reviews

```
summary_top_users <- top_users_by_count %>%
  left_join(average_stars_for_top_users, by = "user_id") %>%
  arrange(desc(review_count)) # Re-arrange to ensure order by review_count

summary_table <- kable(summary_top_users,
                              caption = "Top 10 Users: Review Count and Average Stars",
                              col.names = c("User ID", "Total Reviews", "Average Stars"),
                              align = "c",     # Center align columns
                              digits = 2,      # Round average stars to 2 decimal places
                              format = "pipe") # Use "pipe"

print(summary_table)
```

```
##
##
## Table: Top 10 Users: Review Count and Average Stars
##
## | User ID | Total Reviews | Average Stars |
## |:-------:|:-------------:|:-------------:|
## |         |     5829      |     3.00      |
## | u_27070 |      18       |     2.83      |
## | u_11551 |      15       |     3.27      |
## | u_6766  |      15       |     3.27      |
## | u_11229 |      14       |     3.07      |
## | u_14899 |      14       |     2.57      |
## | u_17629 |      14       |     2.21      |
## | u_22933 |      14       |     2.93      |
## | u_23971 |      14       |     2.36      |
## | u_27907 |      14       |     3.43      |
```

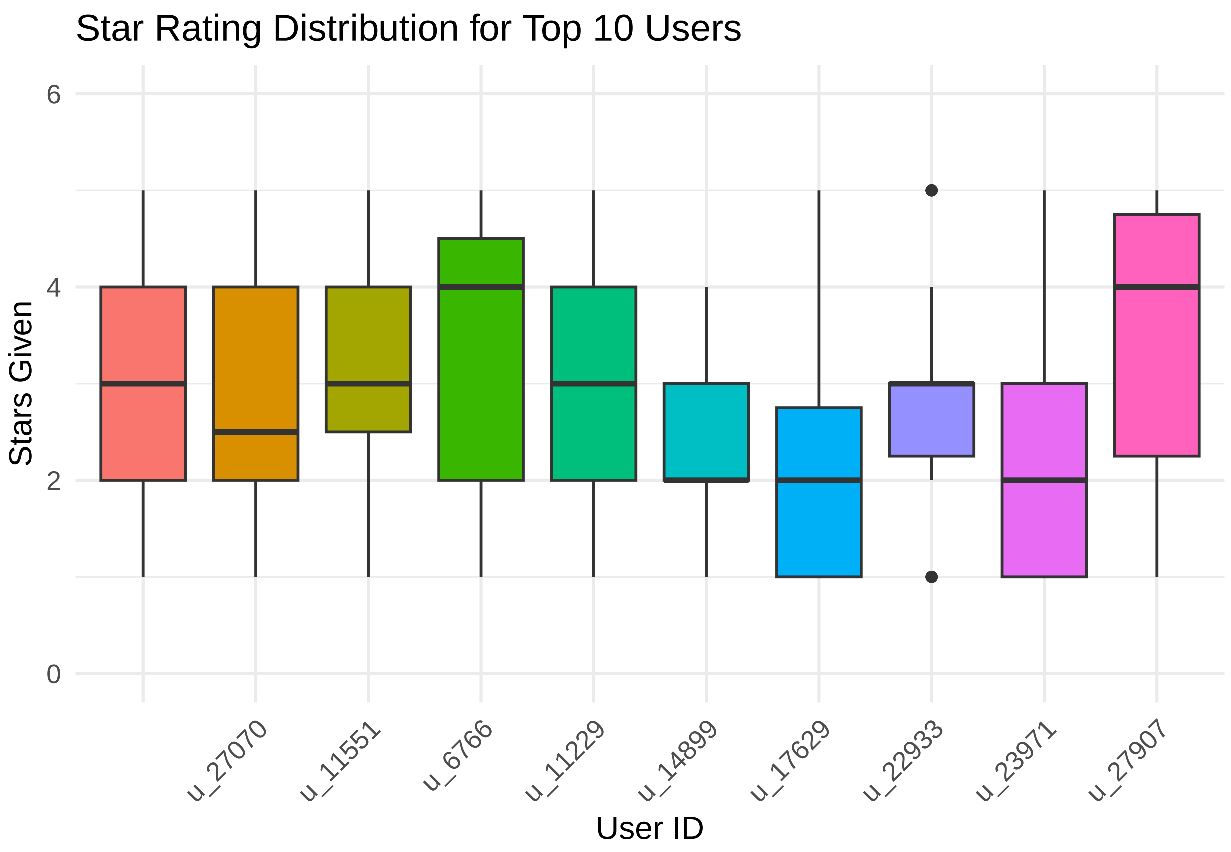
Visualize

```
# Ensure that the order of users in the plot is consistent with the previous summary table
reviews_from_top_users$user_id <- factor(reviews_from_top_users$user_id,
                                         levels = summary_top_users$user_id)

# Create the boxplot
# na.rm = TRUE in geom_boxplot will ensure NA star values are ignored for plotting
rating_distribution_plot <- ggplot(reviews_from_top_users,
                                   aes(x = user_id,
                                       y = stars,
                                       fill = user_id)) +
```

```r
  geom_boxplot(na.rm = TRUE) +
  labs(title = "Star Rating Distribution for Top 10 Users",
       x = "User ID",
       y = "Stars Given") +
  theme_minimal(base_size = 12) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1), # Rotate x-axis labels for readab
        legend.position = "none")  + # Hide legend
  coord_cartesian(ylim = c(0, 6))

print(rating_distribution_plot)
```



The boxplot shows how the top users vary in the way they give star ratings. Users such as u_6766 and u_27907 tend to give higher ratings overall, with medians above 3 and upper quartiles close to 5. In contrast, users like u_17629 and u_22933 give lower ratings more frequently, as indicated by their lower medians and compressed upper ranges. Some users show wide variability (e.g., u_6766, u_27907), while others have more consistent rating patterns (e.g., u_14899). Outliers in a few users suggest occasional extreme ratings.