CHICAGO BOOTH

The University of Chicago Booth School of Business

BUSN 41201 - Big Data - Final Project

PROJECT TITLE

26 May 2024

Yi Cao, Shri Lekkala, Ningxin Zhang

Contents

Note: The full the code used in all the questions can be found in the appendix.

1. Executive Summary

[REDO AFTER WE COMPLETE THE REPORT]

In this report, we present a comprehensive analysis of the "Google Play Store dataset" to gain insights into the characteristics and success factors of mobile applications. By examining various aspects related to app details, including categories, ratings, reviews, sizes, installations, and pricing, we aim to identify patterns and trends that contribute to an app's success on the Google Play Store.

We begin by exploring the general statistics of apps, focusing on the distribution of app categories, ratings, and reviews. This provides a foundational understanding of the data and highlights key areas of interest. Next, we delve into specific analyses to understand the relationship between app size, installs, and pricing, exploring how these factors influence an app's popularity and user engagement.

Our study also includes a sentiment analysis of user reviews, examining the polarity and subjectivity of feedback to understand how user sentiments correlate with app ratings and success. Additionally, we develop predictive models to forecast app ratings based on various features, and we investigate potential causal relationships between app characteristics and their performance metrics.

By leveraging data visualization, feature engineering, and predictive modeling techniques, we aim to provide actionable insights for potential app developers. These insights can help optimize app features, improve user satisfaction, and ultimately enhance the app's visibility and success on the Google Play Store.

2. Introduction

[WE CAN CHANGE THE QUESTIONS, THESE ARE JUST EXAMPLES]

In this paper, we aim to analyze the Google Play Store dataset to gain a comprehensive understanding of the factors that contribute to the success of mobile applications. The dataset includes details of apps such as categories, ratings, reviews, sizes, installations, and pricing, as well as user reviews with sentiment analysis. Our objective is to uncover patterns and trends that can help app developers optimize their offerings and improve user satisfaction.

The Google Play Store dataset, available on Kaggle, consists of two files: `googleplaystore.csv`, which contains detailed information about the apps, and `googleplaystore_user_reviews.csv`, which includes user reviews and sentiment data.

Our analysis will focus on the following research questions:

- **What factors affect the number of installs an app receives?** Specifically, what is the relationship between app size, type (free or paid), price, and the number of installs?
- **What are the key features that influence an app's rating?** How do factors like category, price, and number of reviews contribute to the overall rating of an app?
- **How does user sentiment in reviews correlate with app ratings?**
  Can sentiment analysis of user reviews provide additional insights into user satisfaction and app performance?

We will begin by loading and cleaning the dataset, followed by a thorough exploratory data analysis to uncover initial insights. Subsequently, we will perform detailed analyses to address our research questions, culminating in the development of predictive models and the identification of causal relationships. We will end by making concluding remarks from our research.

3. Dataset

a) Understanding the data

For `googleplaystore.csv` there are the following columns:

- App: Application Name
- Category: Category Type (e.g. Family, Game, Art)
- Rating: User rating review
- Reviews: Number of reviews
- Size: Download size of application
- Installs: Number of user downloads
- Type: Paid or Free
- Price: Price of App
- Content.Rating: Age group that app is targeted at (E.g. Everyone, Teen, Child)
- Genres: Other categories the app belongs to, other than the main category
- Last.Updated: Date when app was last updated
- Current.Ver: Current app version available
- Android.Ver: Minimum required Android version for app

There are a total of 10841 rows (applications).

For `googleplaystore_user_reviews.csv` there are the following columns:

- App: Application Name
- Translated_Review: User review, translated to English
- Sentiment: Positive / Negative / Neutral (Preprocessed)
- Sentiment_Polarity: Sentiment polarity score (Preprocessed)
- Sentiment_Subjectivity: Sentiment subjectivity score (Preprocessed)

This dataset contains the first 100 'most relevant' review for each app, with some prepocessing already done to add the last 3 features.
There are a total of 64295 rows (reviews).

b) Data Cleaning

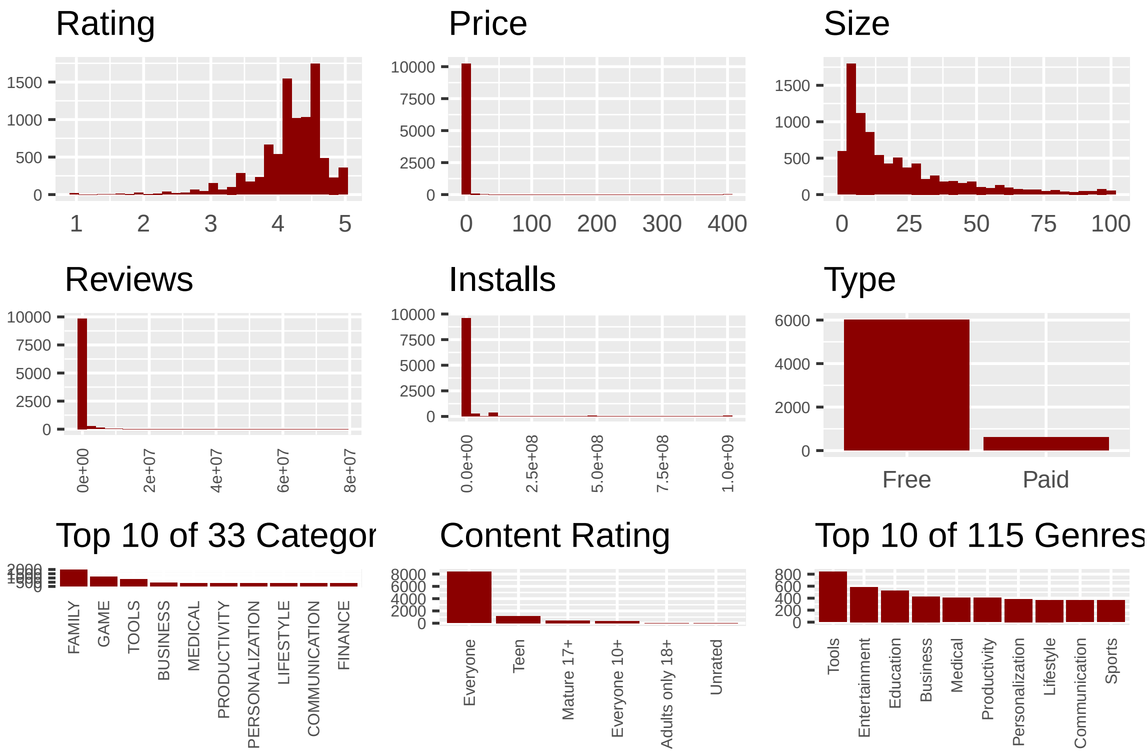
For the `googleplaystore` dataset, we first process the variables by converting columns to the appropriate datatype. For example Installs, Size, Reviews Price, and Android.Ver are converted to numerics, Last.Updated is converted to date. Then we filter out apps with Type 0 or NA, and remove duplicated rows.

After this, we are left with 10356 rows.

With the `googleplaystore_user_reviews` dataset, the variables were already well structured, but we noticed there were many rows with "nan"s. After filtering these out, we were left with 37432 rows.

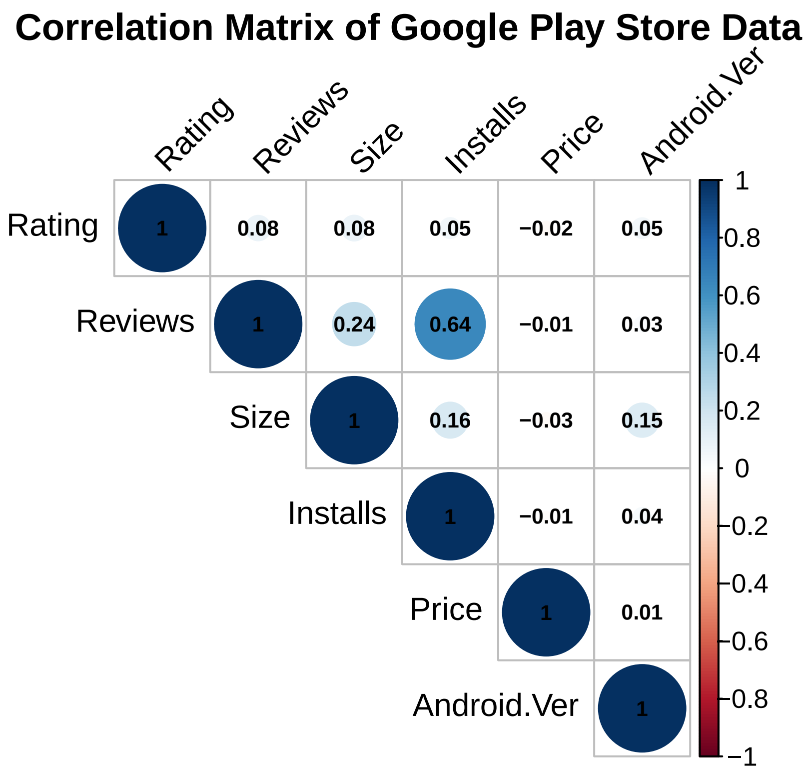
4. Exploratory Analysis

a) Overall Historgram Overview



b) Correlation Matrix

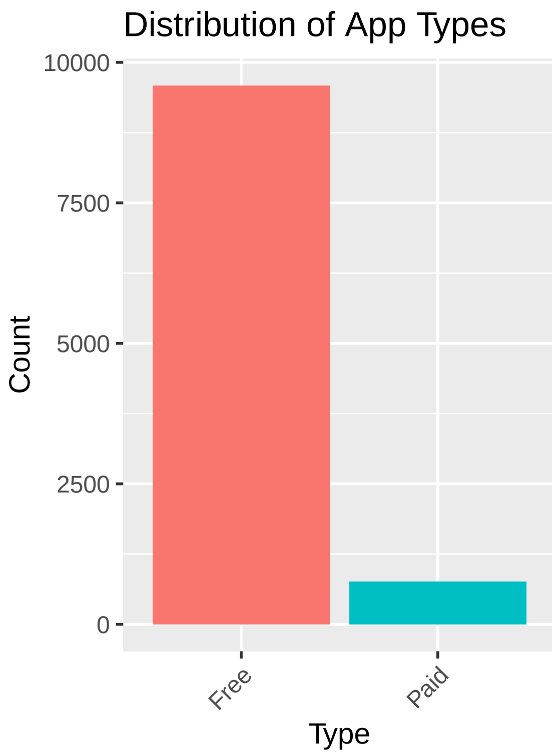
We begin by analysing the correlation matrix of all the numeric variables for googleplaystore:

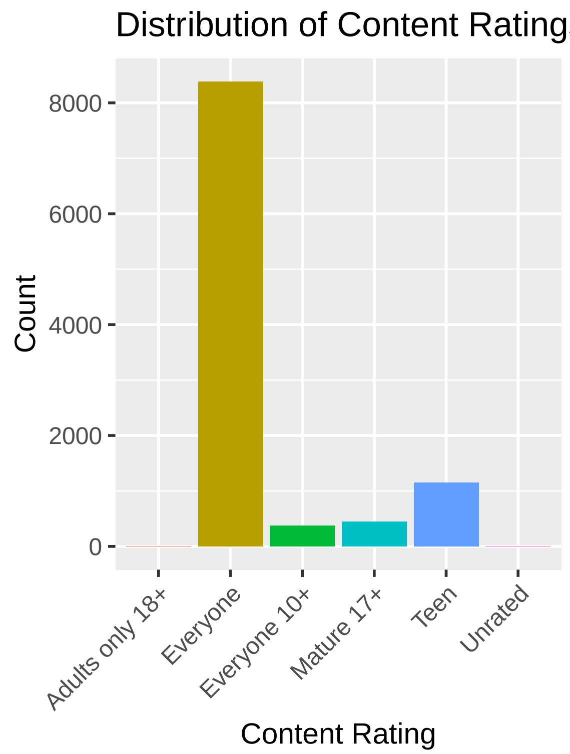


This seems surprising initially as the variables appear to be fairly uncorrelated with eachother, except for the fact that “Installs” and “Reviews” which are highly correlated with a score of 0.64, which would make sense as one would expect a more popular app with a greater number of installs to also have a higher number of reviews. One surprising variable that is somewhat positively correlated with others is “Size”, with small positive correlations with “Reviews” and “Installs”. This might perhaps be due to the fact with apps with a larger download size are more ‘complicated’ and may perform more functions, and thus lead to a greater number of installations and thus reviews too.

c) Categorical Features

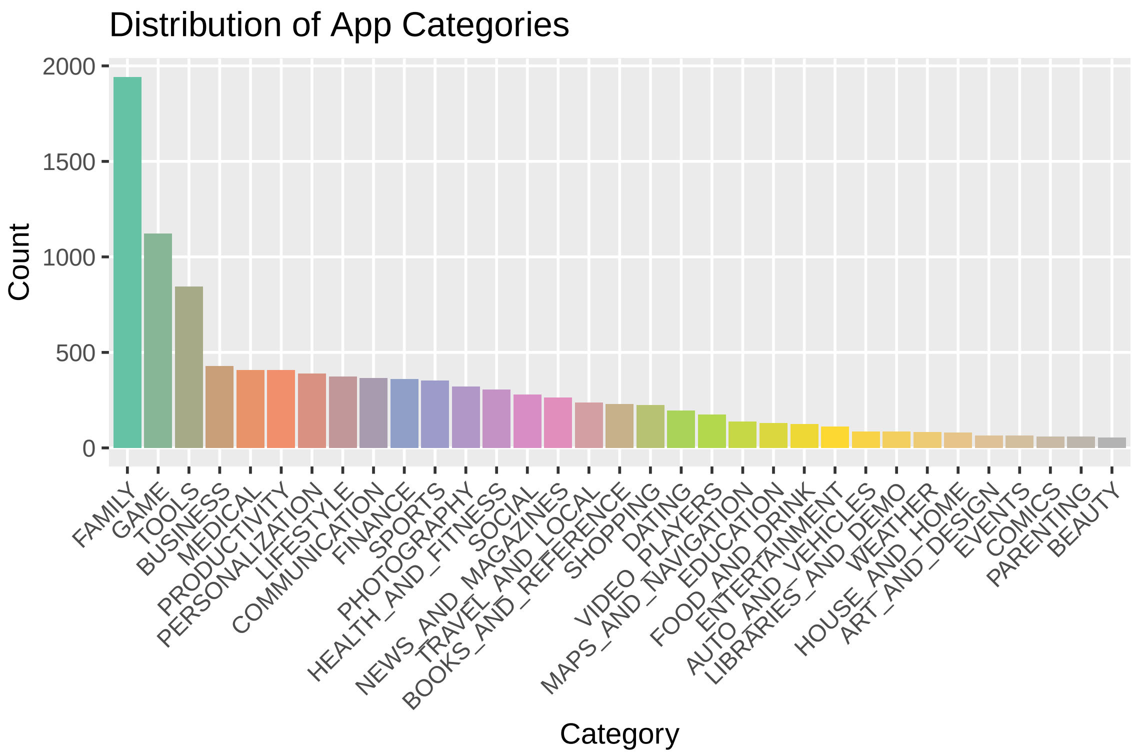
We also look at the distribution of the categorical features in our dataset:

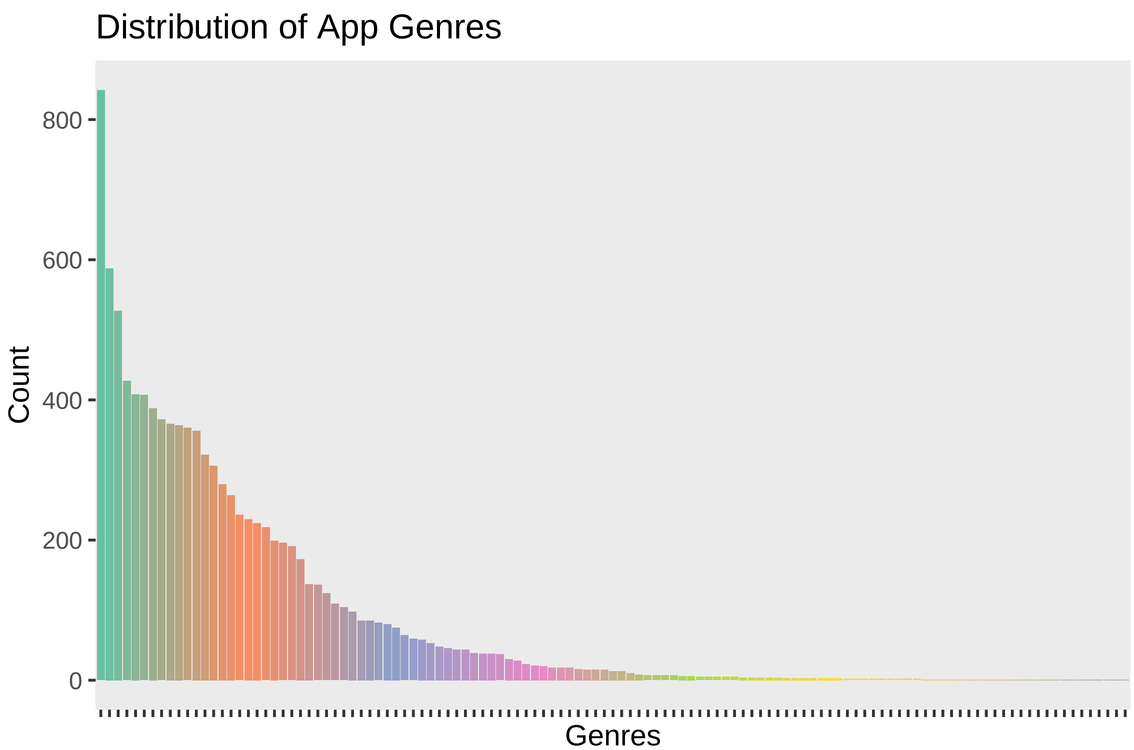




So immediately we observe that there is a much greater proportion of free apps than paid, this aligns with the common “freemium” model where apps are free to download but may offer in-app purchases. This model also lowers the barrier to entry to users.

The content rating distribution shows that the the significant majority of apps are aimed at is “Everyone”. This indicates that most apps are designed to be accessible for a general audience, which makes sense if developers want the largest possible user base for their app.





Next, looking at the distribution of category, sorted by count, we see that distribution is very heavily skewed to the right. In particular the first 3 categories (Family, Game, and Tools) have a very large number of apps, after which the count per category drops and falls slowly for the remaining categories.

Secondly, from the genre distribution (recalling that genres are additional categories that apps can be listed as), we observe the same skewness. However the top 30-40% of genres contain most of the count, whereas afterwards the genres listed have a count of almost 0 which suggests that there are many genres with very few apps, suggesting either niche markets or less popular app types.

Table 1: Top 10 Genres and Categories

Rank	Category	Category_Count	Genre	Genre_Count
1	FAMILY	1942	Tools	842
2	GAME	1121	Entertainment	588
3	TOOLS	843	Education	527
4	BUSINESS	427	Business	427
5	MEDICAL	408	Medical	408
6	PRODUCTIVITY	407	Productivity	407
7	PERSONALIZATION	388	Personalization	388
8	LIFESTYLE	373	Lifestyle	372
9	COMMUNICATION	366	Communication	366
10	FINANCE	360	Sports	364

d)

5. What factors affect the number of installs an app receives?

A. Introduction

B. Analysis

Model 1.

Model 2.

Model 3.

C. Conclusion

6. What are the key features that influence an app's rating?

A. Introduction

B. Analysis

Model 1.

Model 2.

Model 3.

C. Conclusion

7. How does user sentiment in reviews correlate with app ratings?

A. Introduction

B. Analysis

Model 1.

Model 2.

Model 3.

C. Conclusion

8. Conclusion

9. Appendix

```
###########################################
# Setup
###########################################

knitr::opts_chunk$set(
    echo = FALSE,
    fig.height = 4,
    fig.width = 6,
    warning = FALSE,
    cache = TRUE,
    digits = 3,
    width = 48
)

# Required Packages
library(tidyverse)
library(ggplot2)
library(dplyr)
library(corrplot)
library(gridExtra)
library(RColorBrewer)
library(kableExtra)
###########################################
# 3. a) Understanding the datasets
###########################################
# Load the datasets
googleplaystore_raw <- read.csv("data/googleplaystore.csv")
googleplaystore_user_reviews_raw <- read.csv("data/googleplaystore_user_reviews.csv")

# Check the column names
colnames(googleplaystore_raw)
colnames(googleplaystore_user_reviews_raw)

# Check the dimensions
dim(googleplaystore_raw)
dim(googleplaystore_user_reviews_raw)
###########################################
# 3. b) Data Cleaning
###########################################

# Convert the variables to the appropriate data type
googleplaystore <- googleplaystore_raw |>
  mutate(
    # Transform Installs and size to numeric
    Installs = gsub("\\+", "", as.character(Installs)),
    Installs = as.numeric(gsub(",", "", Installs)),
    Size = gsub("M", "", Size),
    # Convert apps with size < 1MB to 0, and transform to numeric
    Size = ifelse(grepl("k", Size), 0, as.numeric(Size)),
```

```
    # Transform reviews to numeric
    Reviews = as.numeric(Reviews),
    # Change currency numeric
    Price = as.numeric(gsub("\\$", "", as.character(Price))),
    # Convert Last.Updated to date
    Last.Updated = mdy(Last.Updated),
    # Change version number to 1 decimal, and add NAs where appropriate
    Android.Ver = gsub("Varies with device", NA, Android.Ver),
    Android.Ver = as.numeric(substr(Android.Ver, start = 1, stop = 3)),
  ) |>
  # Remove apps with Type 0 or NA
  filter(Type %in% c("Free", "Paid")) |>
  # Convert Category, Type, Content.Rating and Genres to factors
  mutate(
    App = as.factor(App),
    Category = as.factor(Category),
    Type = as.factor(Type),
    Content.Rating = as.factor(Content.Rating),
    Genres = as.factor(Genres)
  ) |>
  # Remove duplicate rows
  distinct()

# Remove all rows with nans
googleplaystore_user_reviews <- googleplaystore_user_reviews_raw |>
  filter(Translated_Review != "nan")
common_theme <- theme(
  axis.ticks.x = element_blank(), # Optional: Remove x-axis ticks if not needed
  axis.title.x = element_blank(), # Removes x-axis title for cleaner look
  axis.text.y = element_text(size = 6), # Y-axis text size for uniformity
  axis.title.y = element_blank(), # Removes x-axis title for cleaner look
)

# Determine the top 10 values for categorical data
top_categories <- googleplaystore %>%
  count(Category) %>%
  top_n(10) %>%
  pull(Category)

filtered_google <- googleplaystore %>%
  filter(Category %in% top_categories) %>%
    mutate(Category = factor(Category, levels = names(sort(table(Category), decreasing = TRUE))))

p1 <- ggplot(filtered_google, aes(x = Category)) +
  geom_bar(fill = "darkred") +
  ggtitle("Top 10 of 33 Categories")+
  theme(axis.text.x = element_text(size = 6,angle = 90, hjust = 1, vjust = 0.5))+common_theme
#########
p2 <- ggplot(googleplaystore, aes(x = Rating)) +
  geom_histogram(bins = 30, fill = "darkred") +
  ggtitle("Rating")+common_theme
#########
p3 <- ggplot(googleplaystore, aes(x = Reviews)) +
```

```
  geom_histogram(bins = 30, fill = "darkred") +
  ggtitle("Reviews")+
  theme(axis.text.x = element_text(size = 6,angle = 90, hjust = 1, vjust = 0.5))+common_theme
#########
p4 <- ggplot(googleplaystore, aes(x = Size)) +
  geom_histogram(bins = 30, fill = "darkred") +
  ggtitle("Size")+common_theme
#########
p5 <- ggplot(googleplaystore, aes(x = Installs)) +
  geom_histogram(bins = 30, fill = "darkred") +
  ggtitle("Installs")+
  theme(axis.text.x = element_text(size = 6,angle = 90, hjust = 1, vjust = 0.5))+common_theme
#########
p6 <- ggplot(filtered_google, aes(x = Type)) +
  geom_bar(fill = "darkred") +
  ggtitle("Type")+common_theme
#########
p7 <- ggplot(googleplaystore, aes(x = Price)) +
  geom_histogram(bins = 30, fill = "darkred") +
  ggtitle("Price")+common_theme
#########
filtered_google <- googleplaystore %>%
    mutate(Content.Rating = factor(Content.Rating, levels = names(sort(table(Content.Rating), decreasin
p8 <- ggplot(filtered_google, aes(x = Content.Rating)) +
  geom_bar(fill = "darkred") +
  ggtitle("Content Rating")+
  theme(axis.text.x = element_text(size = 6,angle = 90, hjust = 1, vjust = 0.5))+common_theme
#########
top_genres <- googleplaystore %>%
  count(Genres) %>%
  top_n(10) %>%
  pull(Genres)
filtered_google <- googleplaystore %>%
  filter(Genres %in% top_genres) %>%
  mutate(Genres = factor(Genres, levels = names(sort(table(Genres), decreasing = TRUE))))
p9 <- ggplot(filtered_google, aes(x = Genres)) +
  geom_bar(fill = "darkred") +
  ggtitle("Top 10 of 115 Genres") +
  theme(axis.text.x = element_text(size = 6,angle = 90, hjust = 1, vjust = 0.5))+common_theme
grid.arrange(p2,p7, p4, p3, p5, p6,p1, p8,p9, nrow = 3, ncol = 3, heights = rep(1, 3), widths = rep(1,
###########################################
# 4. b) Correlation Matrix
###########################################
# Select only the numeric columns for the correlation matrix
numeric_columns <- googleplaystore[, sapply(googleplaystore, is.numeric)]

# Compute the correlation matrix
cor_matrix <- cor(numeric_columns, use = "complete.obs")

# Visualize the correlation matrix using a heatmap
corrplot(cor_matrix, type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
```

```
        title = "Correlation Matrix of Google Play Store Data",
        mar = c(0, 0, 1, 0))
##########################################
# 4. b) Categorical Features
##########################################
# Distribution of Types (Free vs. Paid)
p1 <- ggplot(googleplaystore, aes(x = Type, fill = Type)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of App Types", x = "Type", y = "Count") +
  theme(legend.position = "none")

# Distribution of Content Ratings
p2 <- ggplot(googleplaystore, aes(x = `Content.Rating`, fill = `Content.Rating`)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of Content Ratings", x = "Content Rating", y = "Count") +
  theme(legend.position = "none")

# Arrange the plots in a grid
grid.arrange(p1, p2, ncol = 2)
# Count the number of apps in each category
category_counts <- googleplaystore |>
  count(Category) |>
  arrange(desc(n))

# Convert Category to a factor with levels ordered by count
category_counts$Category <- factor(category_counts$Category, levels = category_counts$Category)

# Plot the distribution of app categories sorted by count
p3 = ggplot(category_counts, aes(x = n, y = Category, fill = Category)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of App Categories", x = "Count", y = "Category") +
  theme(legend.position = "none") +
  scale_fill_manual(values = colorRampPalette(brewer.pal(8, "Set2"))(nrow(category_counts))) +
  theme(panel.grid.minor = element_blank()) +
  coord_flip()

# Count the number of apps in each genre
genre_counts <- googleplaystore |>
  count(Genres) |>
  arrange(desc(n))

# Convert Genres to a factor with levels ordered by count
genre_counts$Genres <- factor(genre_counts$Genres, levels = genre_counts$Genres)

# Plot the distribution of app genres sorted by count
p4 = ggplot(genre_counts, aes(x = n, y = Genres, fill = Genres)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Distribution of App Genres", x = "Count", y = "Genres") +
  theme(legend.position = "none") +
```

```
  scale_fill_manual(values = colorRampPalette(brewer.pal(8, "Set2"))(nrow(genre_counts))) +
  theme(panel.grid.minor = element_blank(), panel.grid.major = element_blank(), axis.text.x = element_bl
  coord_flip()

p3
p4
# Combine the dataframes
combined_df <- data.frame(
  Rank = 1:10,
  Category = category_counts[1:10,]$Category,
  Category_Count = category_counts[1:10,]$n,
  Genre = genre_counts[1:10,]$Genres,
  Genre_Count = genre_counts[1:10,]$n
)

# Print the combined dataframe using kable
kable(combined_df, caption = "Top 10 Genres and Categories", align = 'c') %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"))
```