PROJECT REPORT

INTRODUCTION

The focus of our project is the comprehensive dataset of Google Playstore Apps, a rich source of information that provides unique insights into the Android app ecosystem. This dataset, obtained from Kaggle, encompasses a wide array of variables, including app names, categories, ratings, sizes, installs, prices, content ratings, release dates, last updated dates, android versions, In-app purchases and editor's choice.

Our analysis aims to identify the underlying patterns and trends within this dataset, focusing on key variables such as app ratings, categories, content rating, and installs.

1. **Category**: This tells us the type of app, like games, education, entertainment, etc.
2. **Ratings**: This is a score given by users, usually on a scale from 1 to 5, indicating how much they liked the app.
3. **Free vs Paid**: This indicates whether the app is available for free or needs to be purchased.
4. **Content Rating**: This tells us the age group the app is suitable for, like 'Everyone', 'Teen', 'Mature 17+', etc.

By analyzing this data, we can understand trends and patterns in the mobile app market, which can be useful for developers, marketers, and even users. For instance, developers can identify popular categories to focus on, while users can find highly-rated apps in their areas of interest.

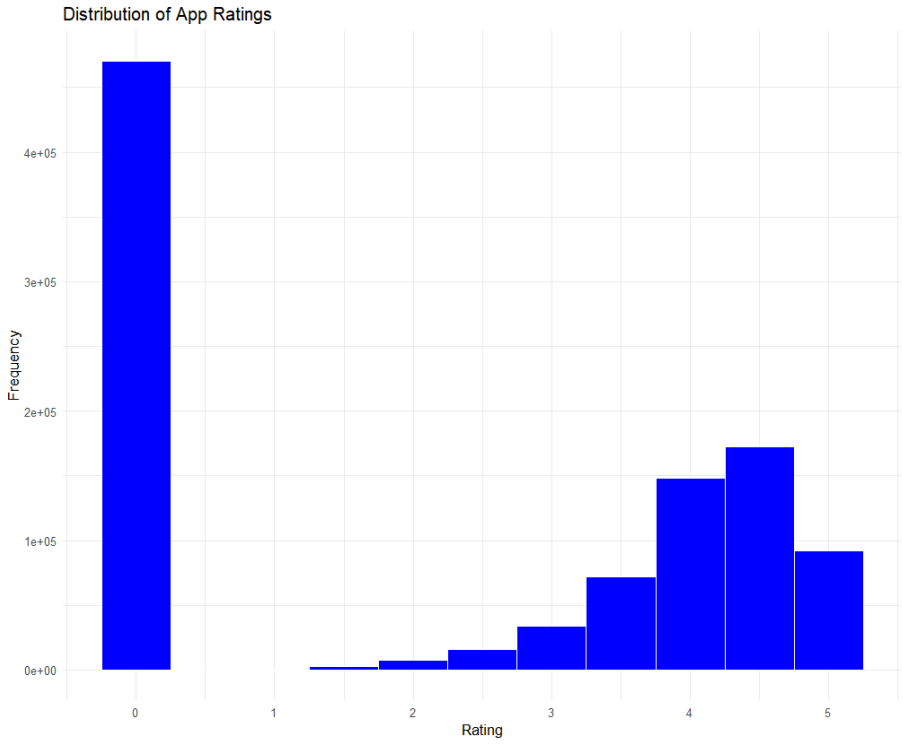
The story we aim to tell through our analysis is understanding the dynamics of the Google Playstore.

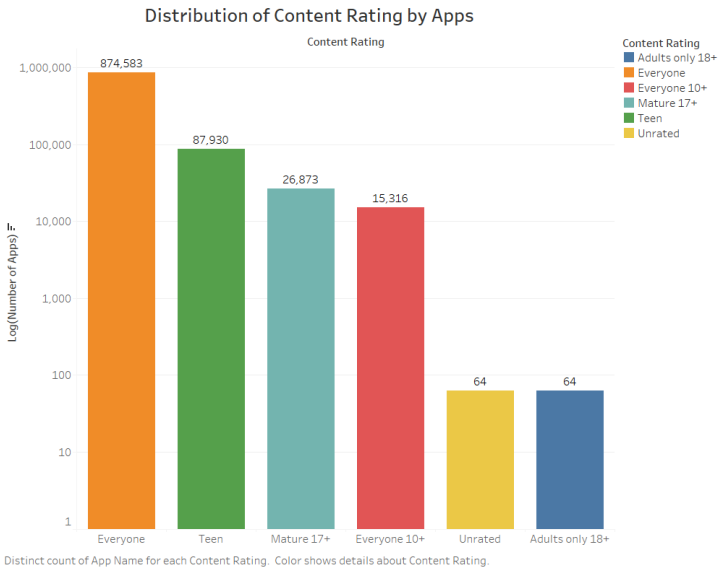
We seek to answer questions such as:

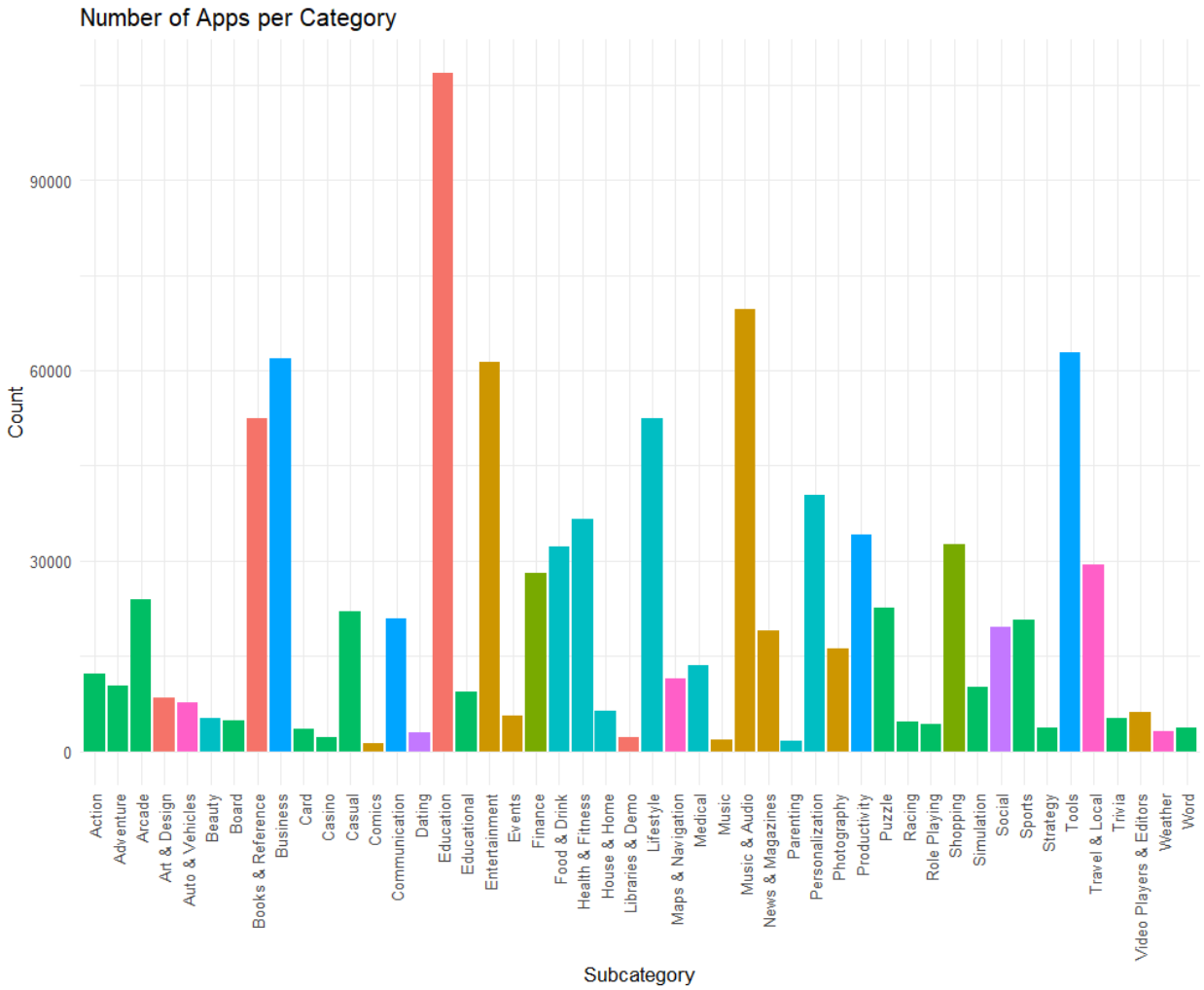
- What types of apps are most popular among users?
- How do ratings and installs impact an app's success?
- Are there specific categories or genres that dominate the Playstore?

EXPLORATORY ANALYSIS

Our initial exploratory analysis involved generating basic visualizations to understand the data's structure and identify potential trends or patterns.




Distribution of Content Rating by Apps
Content Rating
Content Rating
Adults only 18+
Everyone
Everyone 10+
Mature 17+
Teen
Unrated
Log(Number of Apps)
1,000,000
100,000
10,000
1,000
100
10
1
874,583
87,930
26,873
15,316
64
64
Everyone
Teen
Mature 17+
Everyone 10+
Unrated
Adults only 18+
Distinct count of App Name for each Content Rating. Color shows details about Content Rating.

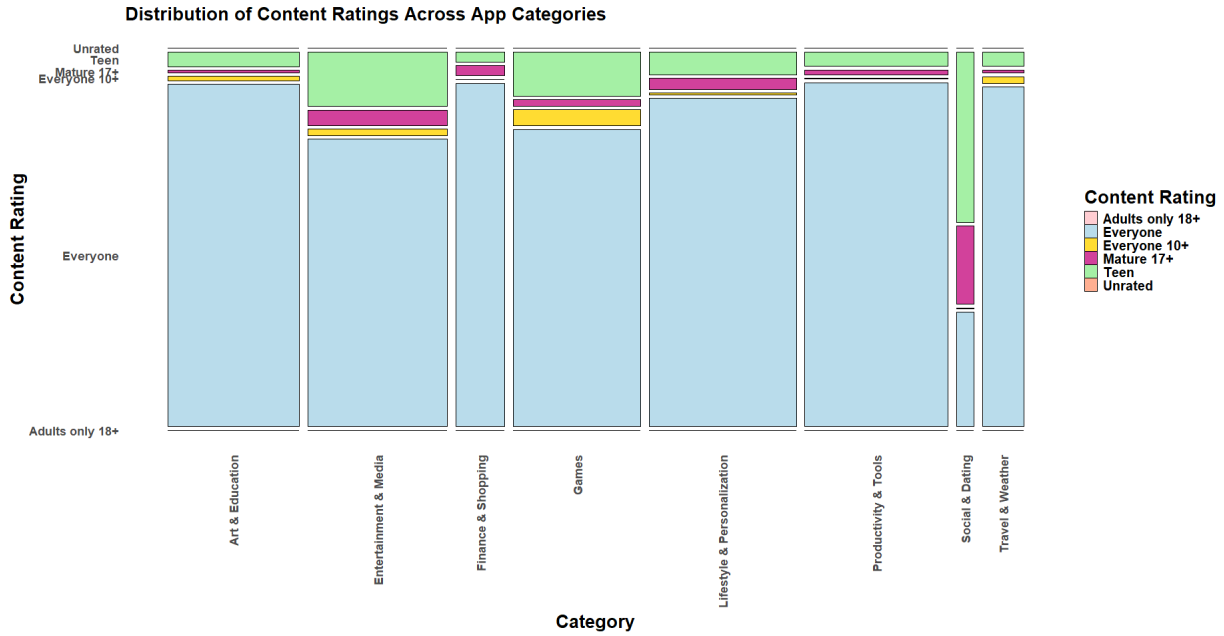

Number of Apps per Category
Count
90000
60000
30000
0
Action
Adventure
Arcade
Art & Design
Auto & Vehicles
Beauty
Board
Books & Reference
Business
Card
Casino
Casual
Comics
Communication
Dating
Education
Educational
Entertainment
Events
Finance
Food & Drink
Health & Fitness
House & Home
Libraries & Demo
Lifestyle
Maps & Navigation
Medical
Music
Music & Audio
News & Magazines
Parenting
Personalization
Photography
Productivity
Puzzle
Racing
Role Playing
Shopping
Simulation
Social
Sports
Strategy
Tools
Travel & Local
Trivia
Video Players & Editors
Weather
Word
Subcategory

Summary of the exploratory data analysis:

- The bar plot shows that apps with lower ratings have a higher frequency, which is because of the unrated and blank ratings imputed by 0. Hence, removing that we can see that the higher number of ratings are between 4-5 which means most of the apps on playstore have the highest ratings.
- The ‘Everyone’ category has the highest count, followed by ‘Teen’ and others. This suggests that a majority of the apps are designed to be suitable for all age groups.
- There are significantly more free apps, which could be due to a variety of factors such as user preference for free apps, or developers using advertising as a revenue model instead of upfront payment.
- The ‘Games’ category stands out with a significantly higher number of apps compared to other categories. This could indicate a high demand for gaming apps or a saturated market.

These visualizations provide valuable insights into the app market, revealing trends in categories, user ratings, pricing models, and content ratings. These insights can guide app development strategies, marketing efforts, and user engagement initiatives. They also highlight areas for further investigation, such as the reasons behind the high number of gaming apps and the prevalence of low-rated apps.

VISUALIZATIONS

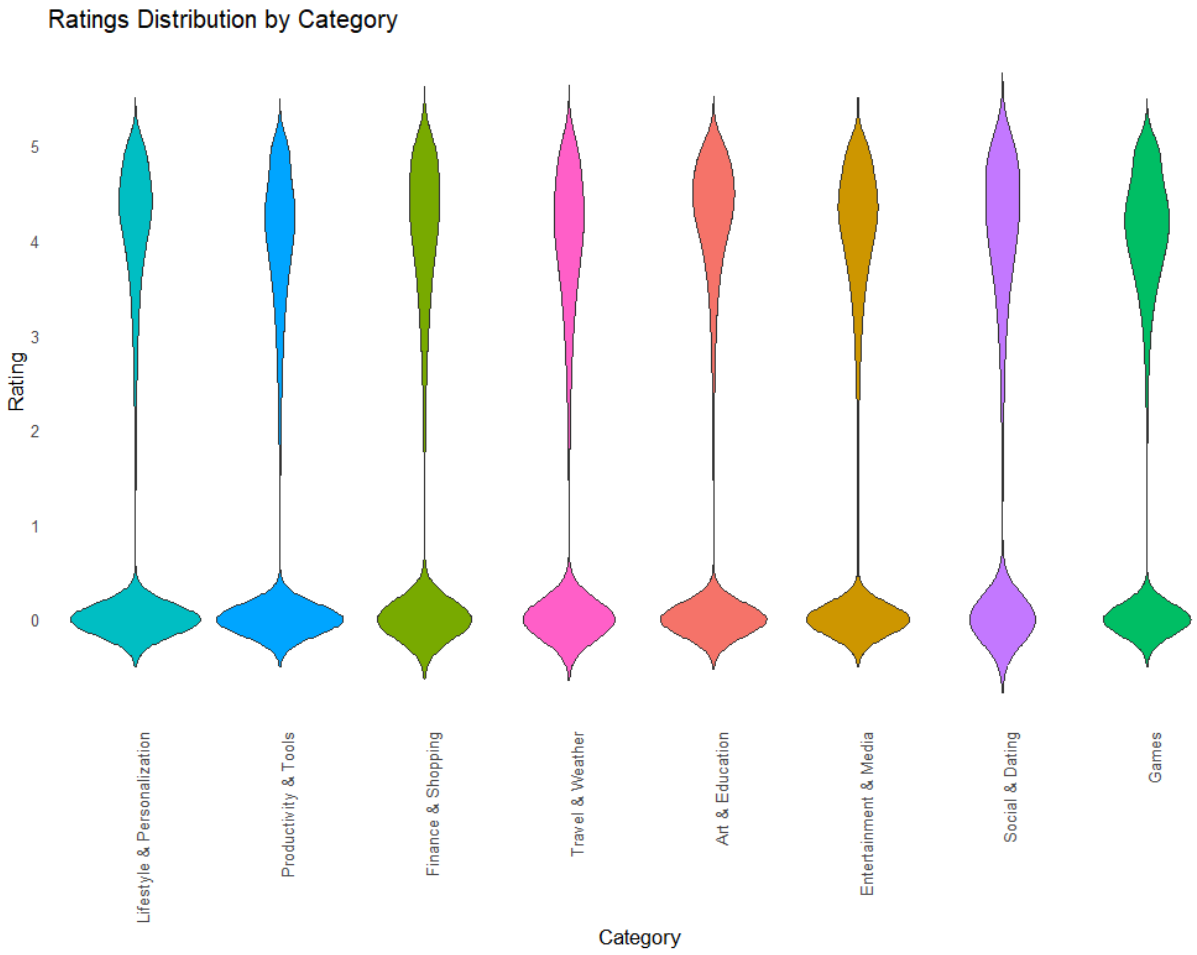


This visualization is a mosaic plot showing the content ratings distribution across various app categories with the app categories on the X-axis, content rating categories on the Y-axis. Each bar represents an app category, and the segments within each bar represent the proportion of apps within each content rating category for that app category.

The graph has been refined to clearly display the distribution of content ratings across different app categories. The use of distinct colors for each content rating category allows for easy differentiation and comparison. This graph format was chosen to effectively show the proportion of each content rating within each app category, highlighting patterns and relationships between the variables.

- The ‘Everyone’ rating has significant representation across all app categories, indicating that most apps are designed to be suitable for a wide audience.
- ‘Games’ has a more diverse distribution of content ratings, suggesting that this category caters to different age groups and preferences.
- ‘Art & Design’ and ‘Education’ have the highest proportion of apps rated as suitable for everyone, implying these categories are more universally accessible and less likely to contain mature content.
- ‘Social & Dating’ are mostly targeted towards mature audiences.

The majority of apps are designed to be suitable for a wide audience; however, there is variability in content ratings within certain categories like Games which cater to different age groups and preferences. This information can be useful for app developers to understand their target audience and for users to find apps suitable for their age and preferences.

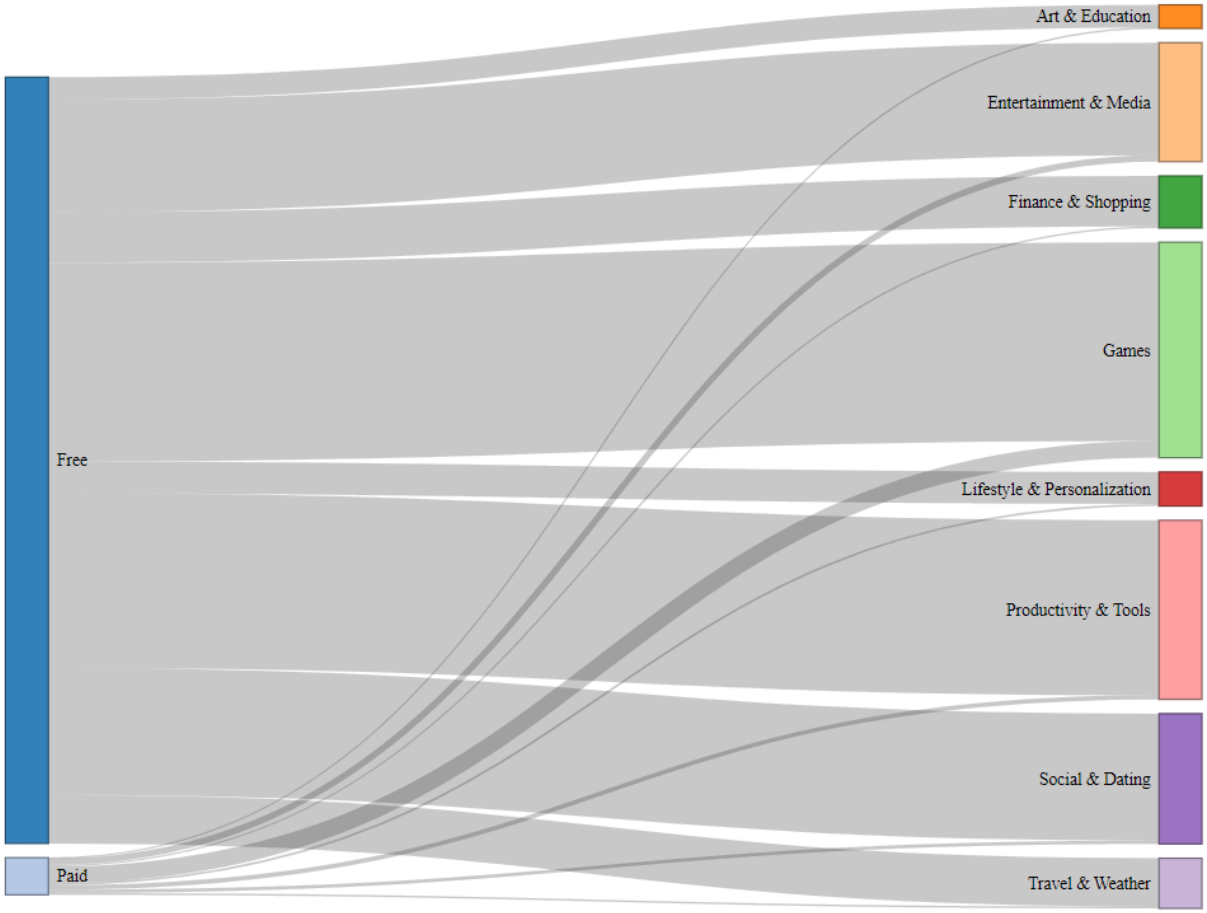


This visualization is a violin plot used to represent the distribution of ratings across different app categories with the horizontal axis representing the categories of apps and the vertical axis representing the ratings of the apps which typically range from 0 to 5.

The plot has been refined to clearly display the distribution of ratings across different categories. The use of distinct colors for each category allows for easy differentiation. The layout helps in identifying patterns such as which categories have wider distributions of ratings or which ones have more concentrated ratings around certain values.

This plot revealed the spread and density of user ratings across different app categories. We discovered that some categories, such as "Games" and "Social & Dating," exhibit a wide range of ratings, indicating diverse user satisfaction levels. In contrast, categories like "Productivity & Tools" tend to have higher average ratings with less variability, suggesting more consistent user satisfaction.

Distribution of Apps by Category and Free/Paid



This is a Sankey diagram, which is a type of flow diagram that visualizes the flow of quantities between different nodes. In this diagram, the nodes represent categories of Google Playstore apps (to the right) and their status as either free or paid(to the left). The flows or links between the nodes represent the average number of installs.The thickness of each link is proportional to the average number of installs for apps in that category and their free or paid status.

Refinement Process:

- Initially, the data was grouped by Free and Category and filtered to include only records with a positive number of average installs.
- The average number of installs was calculated for each group to provide a meaningful measure of app popularity.
- The Free variable was transformed into character values ("Free" and "Paid") to facilitate clear labeling in the diagram.

- Nodes and links were created based on the unique values of Free and Category and their corresponding average installs.

Analysis & Insights:

1. Comparison of Free vs. Paid Apps:

- The diagram shows that free apps generally have a higher number of average installs compared to paid apps, which is indicated by the thicker links originating from the "Free" node.

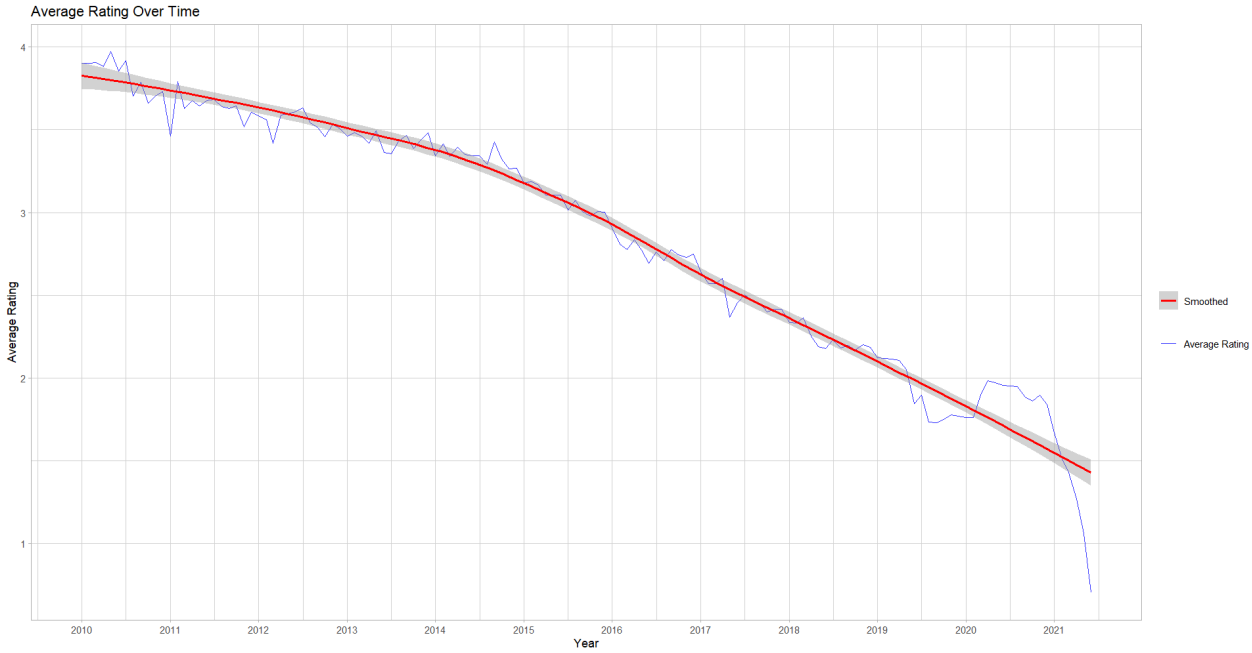
2. Popular Categories:

- Categories like "Games," "Entertainment & Media," and "Productivity & Tools" have significantly thicker links, indicating higher average installs. This suggests these categories are more popular among users.
- The relatively thinner links for "Paid" apps across all categories highlight the tendency of users to prefer free apps over paid ones.

3. Category Distribution:

- The diagram allows for a quick visual assessment of which categories dominate in terms of installs. For instance, the "Games" category for free apps shows a very thick link, highlighting its popularity.

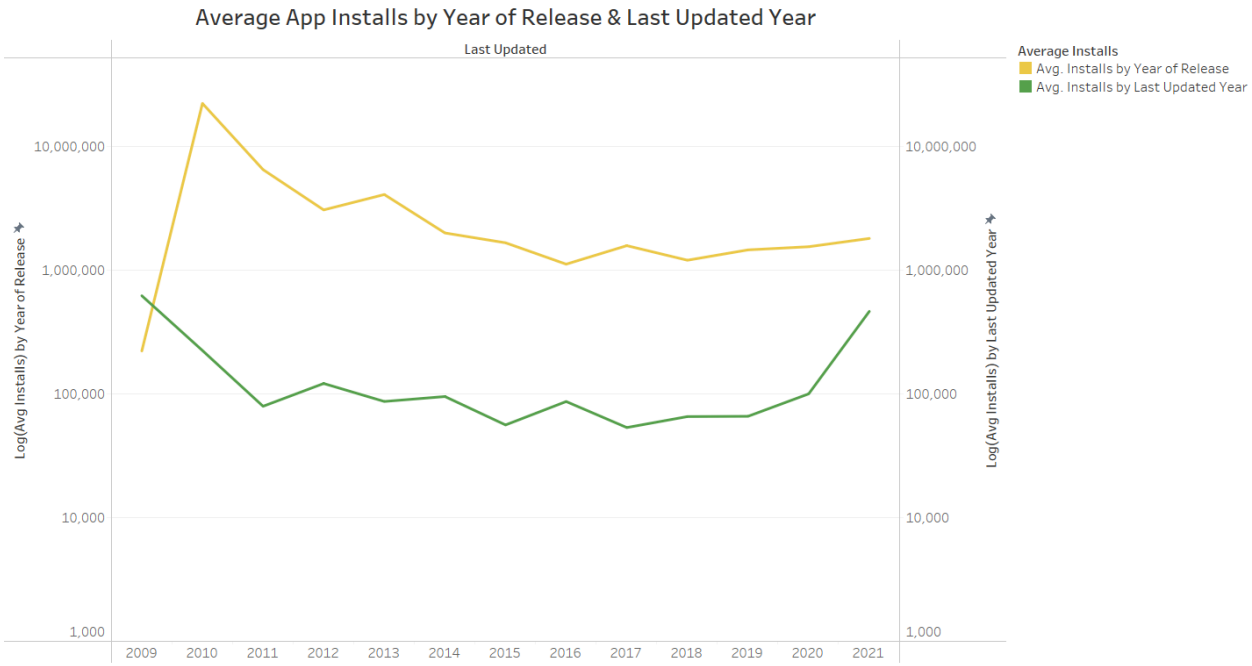
The Sankey diagram effectively brings out the relationships between app categories and their free or paid status, as well as the distribution of average installs. This visualization aids in understanding user preferences and can inform strategies for app development, marketing, and monetization in the Google Playstore.



This is a line graph used to display how the Average Rating of apps has changed over a period. It plots two lines: “Smoothed” (in red) and “Average Rating” (in blue) from the year 2010 to 2021.

- Both the “Smoothed” and “Average Rating” lines show a general downward trend over time. This indicates that the average rating has been decreasing as the years progress.
- The “Smoothed” line is a refinement of the average rating line which is created by applying a smoothing technique to the original data points. It likely represents a moving average that smooths out the fluctuations seen in the “Average Rating” line. This helps reduce the impact of short-term fluctuations and highlight the overall trend.

This graph fits into an analysis or story by providing a clear visual representation of how ratings have evolved over time. The downward trend suggests a decline in ratings throughout this period, which could be a key point in a story about changing consumer behavior, product quality assessments over time, or other phenomena that could affect such ratings.



The visualization is a dual axis line graph showing the “Average App Installs by Year of Release & Last Updated Year” which helps compare two metrics: how many installations apps get based on their release year versus their last update year. This comparison can help understand whether users prefer newer apps or if regularly updated older apps maintain user interest.

Variable Mapping:

- The horizontal axis (x-axis) represents the years from 2010 to 2020.
- The left vertical axis (y-axis) represents the average number of installs by year of release.
- The right vertical axis (y-axis) represents the average number of installs by last updated year.
- The yellow line represents the average installs by year of release.
- The green line represents the average installs by last updated year.

The graph has been refined by using log values to clearly display the trends of average app installs over time. The use of distinct colors for each line allows for easy differentiation and comparison. The line graph format was chosen to effectively show the trends and changes over time, highlighting patterns and relationships between the variables.

Insights:

- There is a general upward trend in the average number of installs over time, both by year of release and last updated year.
- The average number of installs by year of release is generally higher than the average number of installs by last updated year.

While the year of release seems to have a noticeable impact on the average number of installs an app receives, the year of the last update does not show a significant effect. This could suggest that users are more influenced by the novelty of an app (i.e., its release year) than by how up-to-date it is. This could also indicate that users value up-to-date features or bug fixes more than simply new releases. It also suggests that maintaining an existing app could be as valuable as releasing new ones in terms of keeping high installation numbers.

ANALYSIS AND DISCUSSION

Through our comprehensive analysis of Google Playstore apps, several key insights have been uncovered that can guide app developers and marketers:

- **Targeted Content**: Different categories attract different audience demographics. Understanding these can help in tailoring app content and marketing strategies effectively.
- **User Satisfaction**: Categories exhibit varying levels of user satisfaction. Consistently high ratings in categories like "Productivity & Tools" suggest that meeting user needs in these areas can lead to better app reception.
- **Free vs. Paid Models**: Free apps generally attract more installs, especially in entertainment and games. However, there is still a market for paid apps in specialized categories.
- **Improving Quality**: The upward trend in average ratings over time indicates that app quality is generally improving, possibly due to better development practices and competitive pressure.
- **Importance of Updates**: Regular updates are crucial for maintaining user interest and engagement, as evidenced by the correlation between recent updates and higher install numbers.

For app developers and marketers, the key takeaway is the importance of understanding and adapting to user preferences and market trends. Focusing on quality, regular updates, and targeted content can significantly enhance user satisfaction and engagement, leading to higher installs and better app performance. Balancing between free and paid models based on the category and audience can also optimize reach and revenue. By continuously analyzing user feedback and market dynamics, developers can stay ahead in the competitive app market.

If we had more time to develop visualizations further, we might consider:

1. **Interactive Features**: Adding interactivity to visualizations allows users to explore the data in more depth.
2. **Dashboard Creation**: Integrating multiple visualizations into a cohesive dashboard allows users to explore the data from different angles in a centralized location. This can facilitate more comprehensive analysis and decision-making.

APPENDIX:

Code for Data Analysis and Visualizations:

Below is the code used for creating the various visualizations and performing the data analysis alongside cleaning and imputing the missing values.

```
# Load the required libraries

library(tidyverse)
library(scales)
library(ggmosaic)
library(lubridate)
library(viridis)
library(networkD3)

# Load the dataset
Google_Playstore_Apps <-
read.csv("C:/Users/saira/Downloads/Google_Playstore_Cleaned.csv")
Playstore_Apps <- Google_Playstore_Apps

# Rename the columns
Playstore_Apps <- Playstore_Apps %>%
  rename(App_Name = App.Name, Rating_Count = Rating.Count,
         Minimum_Installs = Minimum.Installs,    Maximum_Installs =
Maximum.Installs, Minimum_Android = Minimum.Android, Last_Updated =
Last.Updated, Content_Rating = Content.Rating, Ad_Supported = Ad.Supported,
In_App_Purchases = In.App.Purchases, Editors_Choice = Editors.Choice,
Subcategory = Category)

# Create a new Category column
Playstore_Apps <- Playstore_Apps %>%
  mutate(Category = case_when(
    Subcategory %in% c("Action", "Adventure", "Arcade", "Board", "Card",
"Casino", "Casual", "Educational", "Puzzle", "Racing", "Role Playing",
"Simulation", "Sports", "Strategy", "Trivia", "Word") ~ "Games",
    Subcategory %in% c("Business", "Tools", "Productivity", "Communication") ~
"Productivity & Tools",
```

```
    Subcategory %in% c("Entertainment", "Music & Audio", "Music", "Video
Players & Editors", "Photography", "Comics", "News & Magazines", "Events") ~
"Entertainment & Media",
    Subcategory %in% c("Lifestyle", "Personalization", "Medical", "Health &
Fitness", "Beauty", "Parenting", "House & Home", "Food & Drink") ~ "Lifestyle &
Personalization",
    Subcategory %in% c("Education", "Books & Reference", "Art & Design",
"Libraries & Demo") ~ "Art & Education",
    Subcategory %in% c("Finance", "Shopping") ~ "Finance & Shopping",
    Subcategory %in% c("Travel & Local", "Maps & Navigation", "Weather", "Auto
& Vehicles") ~ "Travel & Weather",
    Subcategory %in% c("Social","Dating") ~ "Social & Dating",
    TRUE ~ as.character(Subcategory)
  ))

# Convert the new 'Category' column to a factor
Playstore_Apps$Category <- as.factor(Playstore_Apps$Category)

# Check the changes
summary(Playstore_Apps$Category)

# Handling missing values
Playstore_Apps <- Playstore_Apps %>%
  mutate(
    Rating = ifelse(is.na(Rating), 0, Rating),
    Rating_Count = ifelse(is.na(Rating_Count), 0, Rating_Count),
    Installs = ifelse(is.na(Installs), 0, Installs),
    Minimum_Installs = ifelse(is.na(Minimum_Installs), 0, Minimum_Installs),
    Maximum_Installs = ifelse(is.na(Maximum_Installs), 0, Maximum_Installs),
    Price = ifelse(is.na(Price), 0, Price),
  ) %>%
  drop_na()

# Handling duplicate values
Playstore_Apps <- Playstore_Apps %>%
  distinct()

# creating a new column 'Average_Installs'
Playstore_Apps <- Playstore_Apps %>%
  mutate(Average_Installs = (Minimum_Installs + Maximum_Installs) / 2)

# Define a function to convert size values to numeric
convert_size_to_mb <- function(Size) {
  if (grepl("M", Size)) {
    return(round(as.numeric(sub("M", "", Size)), 1))
  } else if (grepl("k", Size)) {
    return(round(as.numeric(sub("k", "", Size)) / 1024, 1))
  } else {
    return(Size)  # Retain 'Varies with device'
  }
}

# Apply the function to the Size column
Playstore_Apps$Size_in_MB <- sapply(Playstore_Apps$Size, convert_size_to_mb)

# Formatting into correct datatypes
Playstore_Apps <- Playstore_Apps %>%
```

```
  mutate(App_Name = as.factor(App_Name),
         Category = as.factor(Category),
         Subcategory = as.factor(Subcategory),
         Content_Rating = as.factor(Content_Rating),
         Currency = as.factor(Currency),
         Free = as.factor(Free),
         Ad_Supported = as.factor(Ad_Supported),
         In_App_Purchases = as.factor(In_App_Purchases),
         Editors_Choice = as.factor(Editors_Choice)
  )

# checking for NA values
sum(is.na(Playstore_Apps))

# Filter the data to include only ratings between 0 and 5
Playstore_Apps <- Playstore_Apps %>%
  filter(Rating >= 0 & Rating <= 5)

# Reorder columns
Playstore_Apps <- Playstore_Apps %>%
  select(App_Name, Category, Subcategory, Rating, Rating_Count,
         Installs, Average_Installs, Minimum_Installs, Maximum_Installs,
         Free, Price, Currency, Size, Size_in_MB, everything())

# Check the structure and summary of the cleaned data
str(Playstore_Apps)
summary(Playstore_Apps)

write.csv(Playstore_Apps, file = "Google_Apps_Data.csv")

# ---------------------- EXPLORATORY DATA ANALYSIS -------------------------

# Histogram of Ratings
ggplot(Playstore_Apps, aes(x = Rating)) +
  geom_histogram(binwidth = 0.5, fill = "blue", color = "white") +
  scale_x_continuous(breaks = seq(0, 5)) +
  labs(title = "Distribution of App Ratings", x = "Rating", y = "Frequency") +
  theme_minimal()

# Barplot of Installs

ggplot(Playstore_Apps, aes(x = Installs)) +
  geom_bar(binwidth = 0.5, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Average Installs",
       x = "Installs", y = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        legend.position = "none")

# Bar plot of Categories

ggplot(Playstore_Apps, aes(x = Subcategory, fill = Category)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Number of Apps per Category", x = "Subcategory", y = "Count") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
legend.position = "none")
```

```
# Free vs Paid Apps

ggplot(Playstore_Apps, aes(x = Free, fill = Free)) +
  geom_bar() +  labs(title = "Distribution of Free vs. Paid Apps", x = "Free",
y = "Count") +  theme_minimal()


# ---------------------- EXPLANATORY DATA ANALYSIS ----------------------

# Mosaic Plot of Apps Distribution by Content Rating

# Load necessary libraries
> library(ggplot2)
> library(ggmosaic)
>
> # Define custom lighter colors
> custom_colors <- c("#FFC0CB", "#ADD8E6", "#FFD700", "#C71585", "#90EE90",
"#FFA07A")
> # Reorder the levels of the Content_Rating factor variable
> data$Content_Rating <- factor(data$Content_Rating, levels = c("Adults only
18+", "Everyone", "Everyone 10+", "Mature 17+", "Teen", "Unrated"))
>
> # Create the ggplot with reordered y-axis labels
> ggplot(data = data) +
+     geom_mosaic(aes(weight = 1, x = product(Category), fill =
Content_Rating), na.rm = TRUE, color = "black", size = 0.1) +  # Add thin black
lines
+     scale_fill_manual(values = custom_colors) +  # Use custom lighter colors
+     theme_minimal() +
+     labs(title = "Distribution of Content Ratings Across App Categories",
+          x = "Category",
+          y = "Content Rating") +
+     theme(
+         axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size =
11, face = "bold"),  # Vertical labels to reduce congestion
+         axis.text.y = element_text(size = 11, face = "bold"),  # Adjust
y-axis label size and boldness
+         axis.title.x = element_text(size = 16, face = "bold"),  # Enlarge and
bold x-axis label
+         axis.title.y = element_text(size = 16, face = "bold"),  # Enlarge and
bold y-axis label
+         legend.text = element_text(size = 12, face = "bold"),  # Make the
legend text larger and bold
+         legend.title = element_text(size = 16, face = "bold"),  # Make the
legend title larger and bold
+         legend.position = "right",
+         legend.key.size = unit(0.4, "cm"),  # Make the legend key smaller
+         panel.grid = element_blank(),  # Remove grid lines
+         plot.title = element_text(size = 16, face = "bold"),  # Enlarge and
bold plot title
+         plot.margin = margin(20, 20, 20, 40)  # Increase plot margin for
y-axis labels
+     ) +
+     guides(fill = guide_legend(title = "Content Rating", ncol = 1,
title.position = "top"))
```

 # Violin Plot of Ratings by Category

```
ggplot(Playstore_Apps, aes(x = reorder(Category, Rating, median),
                           y = Rating, fill = Category)) +
  geom_violin(trim = FALSE) +
  scale_y_continuous(breaks = seq(0, 5)) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 1),
        panel.grid = element_blank(),
        legend.position = 'none') +
  labs(title = "Ratings Distribution by Category", x = "Category", y =
"Rating")
```

 # Sankey Diagram of Apps by Category and Free/Paid by Avg Installs

```
# Ensure the 'Free' column is logical (TRUE/FALSE)
Playstore_Apps$Free <- as.logical(Playstore_Apps$Free)

# Group data by Free and Category, filter, and summarize
sankey_data <- Playstore_Apps %>%
  group_by(Free, Category) %>%
  filter(Average_Installs > 0) %>%
  summarise(Average_Installs = mean(Average_Installs), .groups = 'drop')

# Ensure 'Free' column is character for naming
sankey_data <- sankey_data %>%
  mutate(Free = ifelse(Free, "Free", "Paid"))

# Create nodes
nodes <- data.frame(
  name = c("Free", "Paid", as.character(unique(sankey_data$Category)))
)

nodes$id <- seq(0, nrow(nodes) - 1)

# Create links
links <- sankey_data %>%
  left_join(nodes, by = c("Free" = "name")) %>%
  rename(source = id) %>%
  left_join(nodes, by = c("Category" = "name")) %>%
  rename(target = id) %>%
  select(source, target, Average_Installs)


# Plot Sankey diagram
sankeyNetwork(
  Links = links, Nodes = nodes, Source = "source", Target = "target",
  Value = "Average_Installs", NodeID = "name", units = "Avg Installs",
  fontSize = 12, nodeWidth = 30
)
```

 # Time Series graph representing the Ratings over Time

```
ggplot(monthly_data, aes(x = YearMonth, y = Average_Rating, color = "Average
Rating")) +
+     geom_line(alpha = 0.8, linetype = "solid") +  # Adjust line transparency
and linetype
```

```
+     geom_smooth(method = "loess", color = "red", aes(linetype = "Smoothed"))
+
+     labs(title = "Average Rating Over Time",
+          x = "Year",
+          y = "Average Rating",
+          color = "Line") +
+     theme_light() +  # Set a lighter theme
+     scale_x_date(date_breaks = "1 year", date_labels = "%Y") +  # Adjust
x-axis scale
+     scale_color_manual(values = c("blue", "red"), name = NULL) +  # Customize
legend colors
+     scale_linetype_manual(values = c("solid", "dashed"), name = NULL) +  #
Customize linetype legend
+     theme(panel.grid.major = element_line(color = "lightgrey", size = 0.1),
# Lighter major grid lines
+           panel.grid.minor = element_line(color = "lightgrey", size = 0.5))
# Lighter minor grid lines
```

Results of Formative, Exploratory Data Analysis

During the exploratory data analysis (EDA), several steps were undertaken to understand the structure and key characteristics of the dataset. The following are some of the key steps and their results:

1. Summary Statistics:

This provided an overview of the dataset, including measures like mean, median, min, max, and quartiles for numerical variables, and frequency counts for categorical variables.

2. Data Cleaning:

- Missing values were handled, and any inconsistent or erroneous data points were corrected or removed.
- Renamed the variable names and reordered them for easier analysis.
- Removed a few variables that were not useful for the analysis like ‘Developer email’, ‘Developer ID’, ‘Privacy Policy’ and more.

3. Data Transformation:

- The quantitative variables are converted to numeric values and categoricals are converted to factors.
- The 'Free' column was converted to a logical (TRUE/FALSE) format to facilitate analysis.
- Dates were converted to appropriate date formats for time series analysis.

- Created a new column ‘Category’ by combining various subcategories into a broader category based on their genre.

Initial Visualizations:

- Histograms and box plots were created to understand the distribution of key variables like ratings and installs.
- Scatter plots were used to explore potential relationships between variables such as ratings and installs, release dates, and last update dates.

These steps provided a foundational understanding of the dataset, which guided the subsequent detailed analysis and creation of the final visualizations.

This structured approach, from initial data exploration to the final visualizations, ensured a thorough analysis that revealed key insights into Google Playstore apps' performance and user engagement patterns.