**Analysis and Marketing Strategy for Google Play Store Applications**

Rita Alfaro

Luisa Hernandez

Katherine Karow

Rosangelica Machado

Antonella Marazita

Jose Mogna

Kasi Simpkins

## Table of Contents

[**Table of Contents 2**](#_b01g1c5ueexx)

[**I. Dataset & Problem Statement 3**](#_dghjuz6972p5)

[**II. Methodology 3**](#_iixnzhvyeio9)

[Data Cleaning & Preprocessing 3](#_hpmptyf5jz2s)

[Clustering Approach 5](#_nq5tb2ya81ve)

[Validation methods 5](#_pejq0ia1n2um)

[❖ Elbow Method 5](#_j0ka52247cfq)

[❖ Silhouette Score 6](#_ltrputubi2q7)

[❖ Gap Statistic 6](#_n90l8m37avax)

[❖ Cluster Breakdown 6](#_sttzych156kh)

[K-Means Clustering 7](#_4lf1tloqf0hs)

[Hierarchical Clustering (HClust) 7](#_xys6hyaplri4)

[**III. Cluster Analysis and Sales Prediction 7**](#_y9jjtfwgdm75)

[Cluster Analysis 7](#_4m8ekaqfbw06)

[Cluster Characteristics 8](#_qqazl4z64dpr)

[Cluster Descriptions 8](#_e9wykc7ie9q7)

[Budget Allocation 9](#_4i7vh4sa6icd)

[Budget Allocation Breakdown 10](#_ud9qv7bv4h6x)

[Sales and Promotion Strategy 11](#_kh0sgo2c1bkx)

[Cluster 1: 11](#_w3ws3bf813ri)

[Cluster 2: 11](#_5lca8tfbgtg8)

[Cluster 3: 11](#_vqp5kgl33pqa)

[Cluster 4: 12](#_5ew33f3p1mp6)

[Cluster 5: 12](#_8dsv6vdnwhqp)

[**IV. Results and Visualizations 14**](#_cuz5fpp2t0w5)

[Visualizations 14](#_75cih58cjpuy)

[Elbow Method Plot 14](#_wp8wtj5gt8d2)

[Silhouette Score Analysis 15](#_isneb3ofozcv)

[Gap Statistic Plot 16](#_h47lrytr5g6y)

[Cluster Plots 17](#_pknochizckl5)

[Dendrogram of Apps 19](#_7ypp0q3rd4bs)

[Ratings Across Clusters Box Plot 20](#_9ynrmrqdm5ug)

[Size vs Rating by Category Scatter Plot 21](#_urdl6yrzrok)

[Average Installations per Cluster Bar Graph 22](#_mawkdq265vz1)

[Rating vs Installs by Category Scatter Plot 23](#_rxdjcllga5r7)

[Budget Allocation Per Cluster 23](#_ljqteyte01n4)

[**V. Conclusion 24**](#_lxu8622l4gh2)

[**VI. Appendix 25**](#_vl3q114mgwht)

[Code Used: 25](#_trq8x74tj6p0)

[Without Outputs: 25](#_6qdpo3jli92t)

[Full Code with Outputs: 35](#_m1tefthrqcql)

## I. Dataset & Problem Statement

The dataset analyzed contains information about applications available on the Google Play Store. As an application analyst, the primary goal is to classify apps based on their features using clustering techniques and deriving insights to aid the marketing team in designing a sales and promotion strategy.



The dataset includes variables such as:

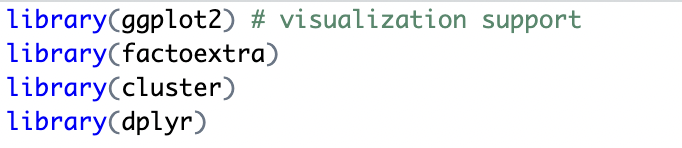
* **App Name:** The name of the application.
* **Category:** The category under which the app is listed.
* **Rating:** The average user rating for the app.
* **Reviews:** The total number of user reviews received.
* **Size:** The file size of the application.
* **Installs:** The number of times the app has been installed.
* **Type:** Whether the app is free or paid.
* **Price:** The cost of the application (if applicable).
* **Content Rating, Genres, Last Updated, Current Version, and Android Version.**

The objective is to classify apps into distinct groups based on their characteristics and develop a targeted marketing strategy for each group.

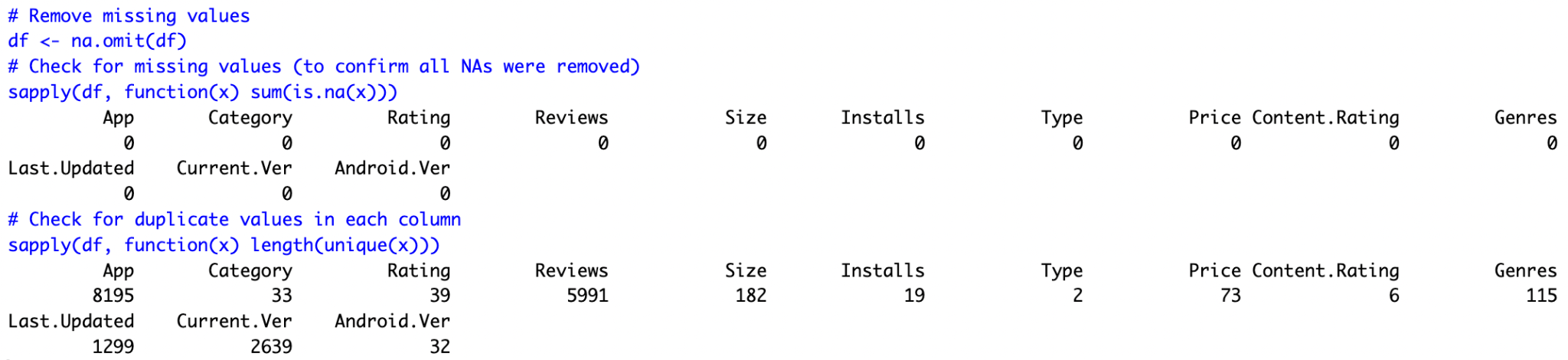
## II. Methodology

### Data Cleaning & Preprocessing

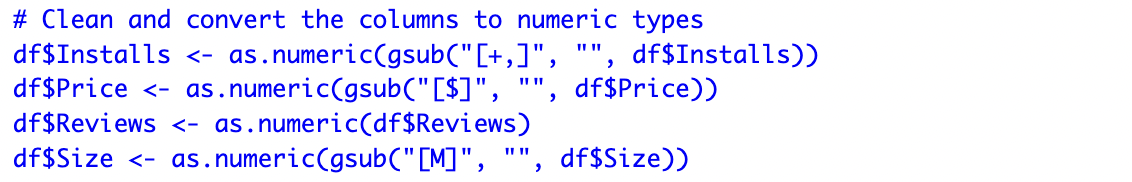
* **Libraries implemented** 
  + **ggplot2:** used for creating data visualizations
  + **dplyr:** utilized for data manipulation and transformation
  + **factoextra:** designed for visualizing the results of multivariate data analyses
  + **cluster:** used for clustering analysis

****

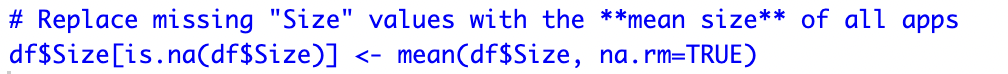
* **Handling Missing Values:** Removed NaN values and missing data from key variables.



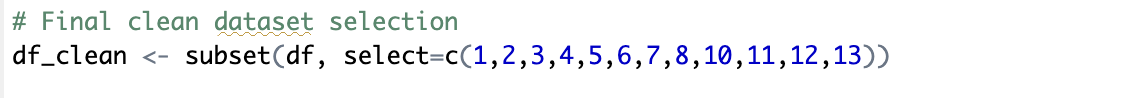
* **Data Formatting:** Converted categorical variables into numeric formats where necessary.



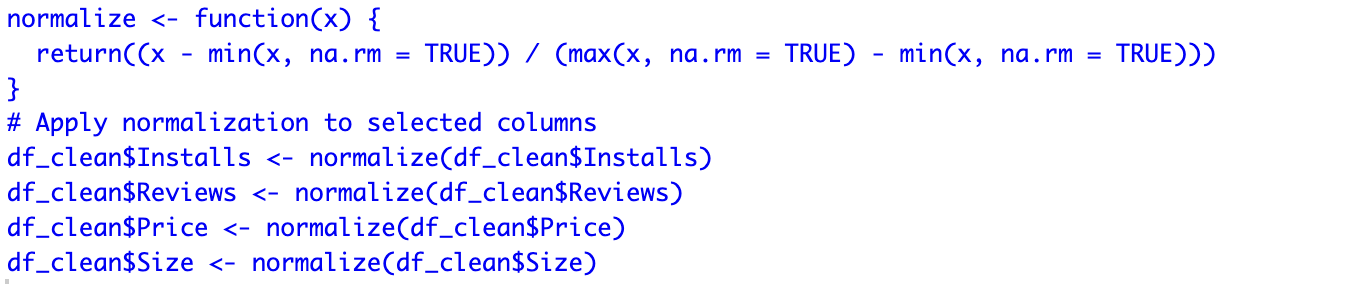
* **Missing values:** Replaced missing values with mean size of key features for analysis.



* **Final Data Cleaning:** Selected key features from raw data that were necessary for insight results.

****

* **Variable Formatting:** Normalizing variables for fair clustering analysis and applied to variables needed.



* **Data Distribution Analysis:**
  + Installs: Highly skewed, with some apps having millions of installs and others having very few.
  + Price: Most apps are free, with only a small fraction being paid apps.
  + Ratings: Majority of apps have ratings between 3.5 and 4.7.
  + Reviews: Wide variation in the number of reviews

### Clustering Approach

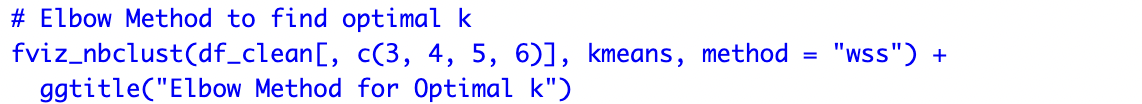
Two clustering techniques were used:

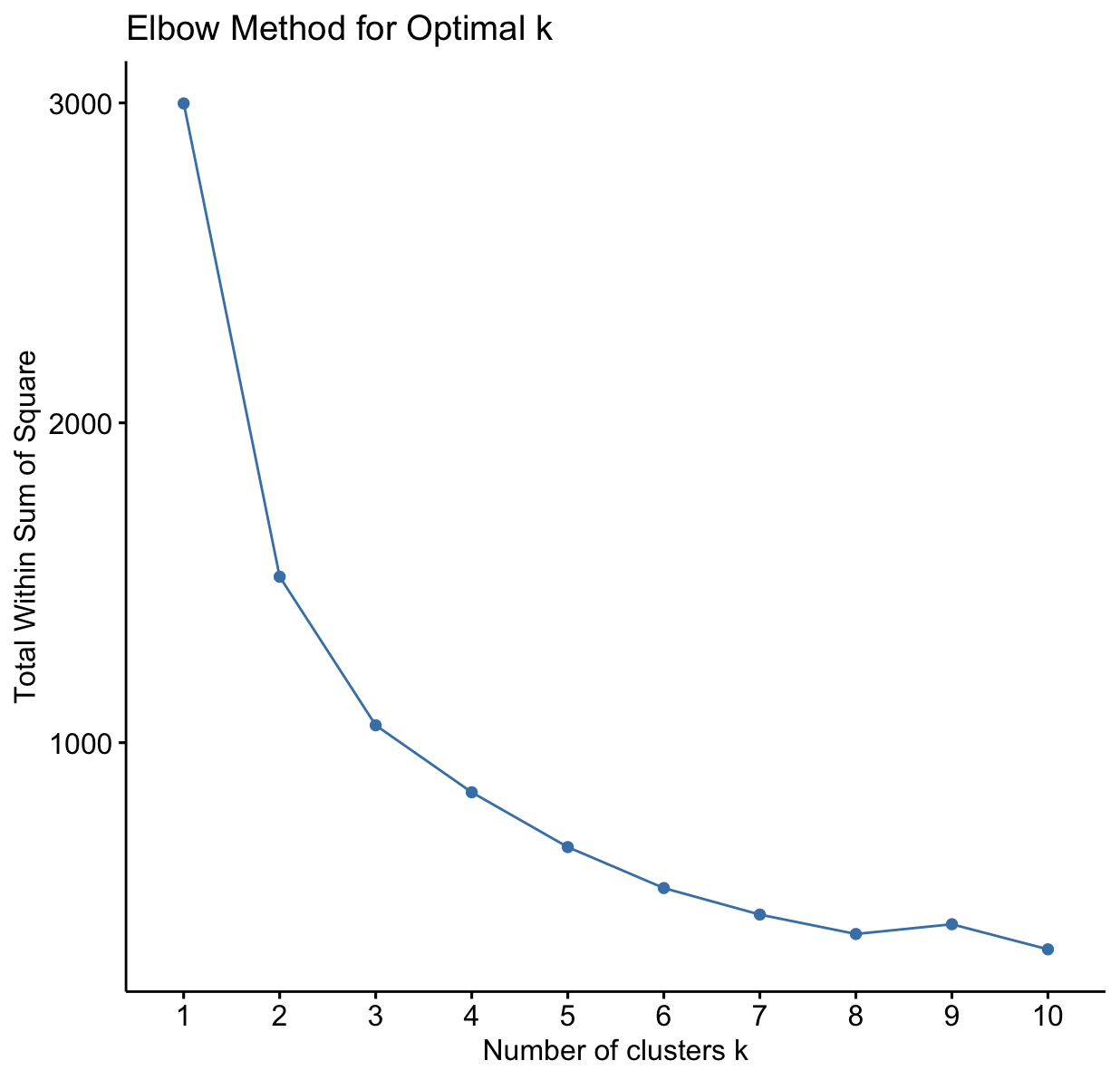
1. **K-Means Clustering** – Used for grouping apps based on their characteristics.
2. **Hierarchical Clustering (HClust)** – Used to confirm the K-Means results and visualize app relationships.

### Validation methods

#### **Elbow Method**

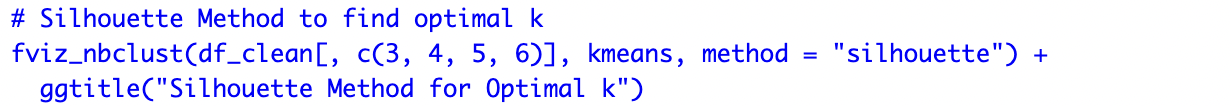
Determined the optimal number of clusters.





#### **Silhouette Score**

Assessed the quality of clusters.



#### **Gap Statistic**

Additional validation for cluster count.

# Gap Statistics (visualized through fviz\_nbclust using gap\_stat method)

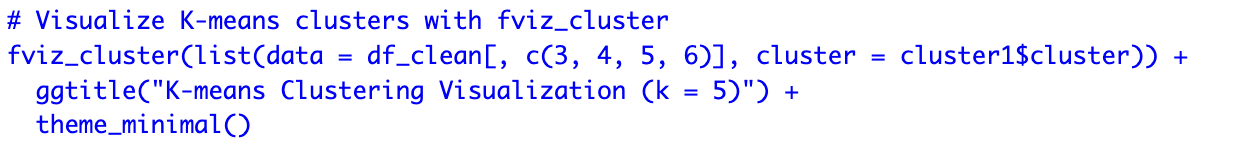
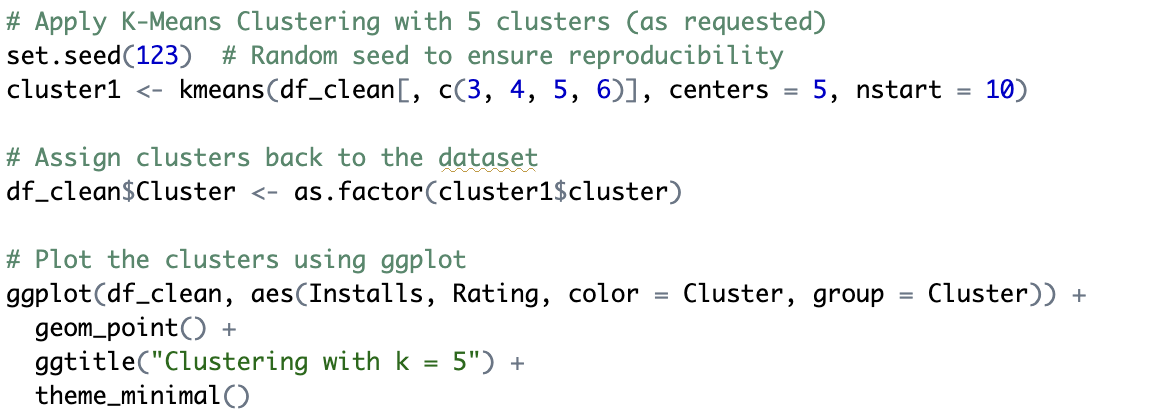
gap\_stat <- clusGap(df\_clean\_scaled[, c(3, 4, 5, 6)], FUNcluster = kmeans, K.max = 10, B = 100)

fviz\_gap\_stat(gap\_stat) + ggtitle("Gap Statistics for Optimal k")

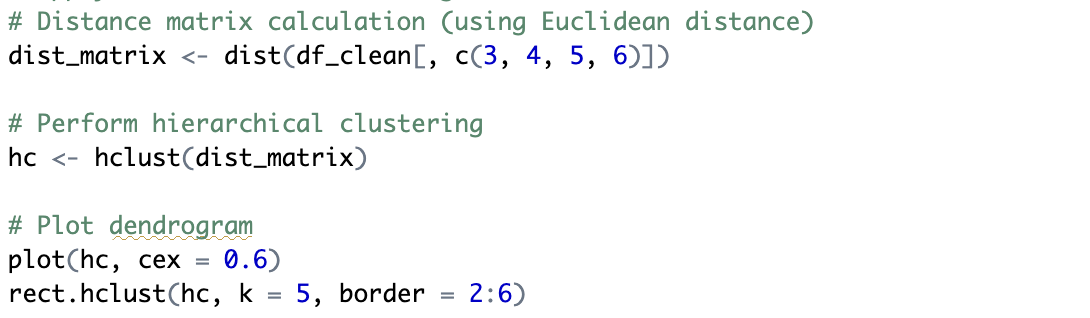
#### Cluster Breakdown

| **Cluster Name** | **Avg\_Rating** | **Avg\_Reveews** | **Avg\_Installs** | **Avg\_Price** | **Budget\_Allocation** |
| --- | --- | --- | --- | --- | --- |
| **Cluster 1** | -0.0388 | -0.477 | -0.461 | 1.63 | 70,000 |
| **Cluster 2** | 0.463 | -0.389 | -0.422 | -0.914 | 50,000 |
| **Cluster 3** | 0.477 | 1.79 | 1.79 | 0.0609 | 90,000 |
| **Cluster 4** | 0.804 | -0.438 | -0.436 | -0.703 | 120,000 |
| **Cluster 5** | -1.71 | -0.483 | -0.470 | -0.0752 | 50,000 |

#### K-Means Clustering

****

#### Hierarchical Clustering (HClust)

****

## III. Cluster Analysis and Sales Prediction

### Cluster Analysis

We used normalized values for Avg\_Rating, Avg\_Reviews, Avg\_Installs, and Avg\_Price to segment the apps into five clusters. Normalization adjusts the data to a consistent scale, allowing comparison across variables. For example, ratings, reviews, and installs have different scales, so normalization prevents one from dominating the analysis. The clusters reflect distinct app behaviors, such as apps with high ratings but low installs (Cluster 1) versus apps with high ratings and installs (Cluster 3). This approach ensures we group apps with similar performance, making the analysis more meaningful.

### Cluster Characteristics

| **Cluster** | **Avg\_Rating** | **Avg\_Reviews** | **Avg\_Installs** | **Avg\_Price** | **Budget Allocation** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | -0.04 | -0.48 | -0.46 | 1.63 | $70,000 | Low installs, moderate ratings. Focus on boosting installs. |
| 2 | 0.46 | -0.39 | -0.42 | -0.91 | $50,000 | Moderate ratings, low installs. Focus on building presence. |
| 3 | 0.48 | 1.79 | 1.79 | 0.06 | $90,000 | High ratings, reviews, and installs. Focus on sustaining growth. |
| 4 | 0.80 | -0.44 | -0.44 | -0.70 | $120,000 | High ratings, moderate reviews/installs. Major marketing push needed. |
| 5 | -1.71 | -0.48 | -0.47 | -0.08 | $50,000 | Low ratings, reviews, installs. Focus on app improvement. |

### Cluster Descriptions

* Cluster 1: Apps with low installs but decent ratings. Often newer or niche apps. Could be social media or small utility apps.
* Cluster 2: Apps with moderate ratings and reviews but lower installs. Likely growing apps in competitive niches.
* Cluster 3: Apps with high ratings, reviews, and installs. Established apps in mainstream categories like social media, entertainment, or productivity.
* Cluster 4: High ratings but lower reviews/installs. These may be premium or niche apps requiring better marketing to reach more users.
* Cluster 5: Low ratings, reviews, and installs. Likely new or low-quality apps needing significant improvements and targeted marketing.

| Cluster | Challenges | Needs |
| --- | --- | --- |
| 1 | Low visibility, poor ratings | Intensive brand awareness campaigns |
| 2 | Limited reach, niche market | Targeted ad campaigns, influencer partnerships |
| 3 | Competition, high user expectations | Sustained marketing, loyalty programs |
| 4 | Limited installs, niche audience | Brand awareness, social media campaigns |
| 5 | Poor ratings, low installs | App improvements, aggressive marketing strategies |

By using these insights, we can allocate marketing budgets based on app performance, ensuring effective campaigns.

### Budget Allocation

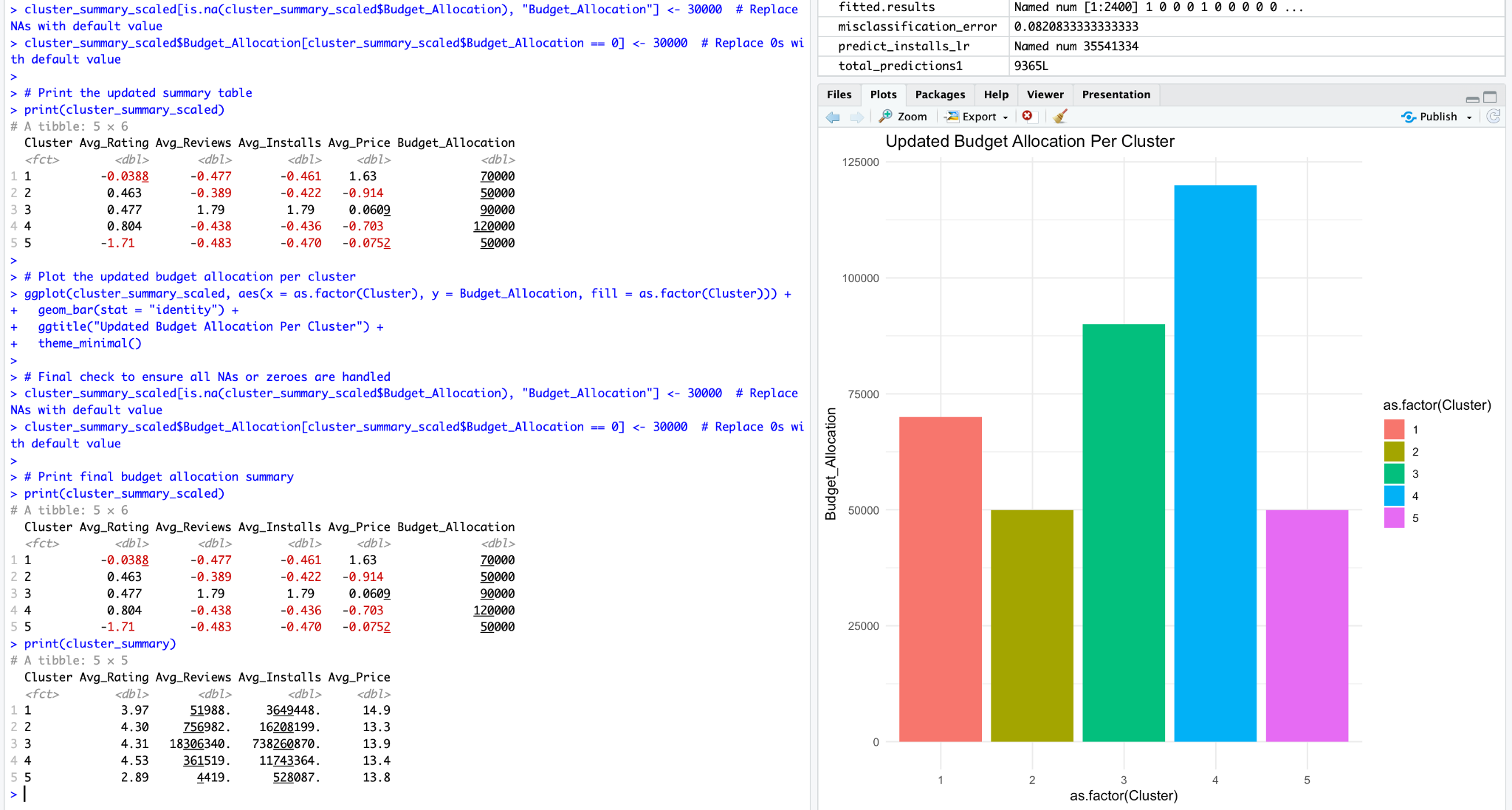
Budget Allocation was based on the normalized values of Avg\_Rating, Avg\_Reviews, Avg\_Installs, and Avg\_Price. The allocation aimed to align with the app’s characteristics:

* Avg\_Rating: Higher ratings usually mean better organic growth, needing less marketing.
* Avg\_Reviews: More reviews imply higher visibility and credibility, justifying a larger budget.
* Avg\_Installs: More installs mean less intensive marketing is needed for growth.
* Avg\_Price: Higher-priced apps usually require more marketing to justify the cost.

We allocated higher budgets to clusters with higher ratings and installs (e.g., Cluster 3, $90,000), and smaller budgets to clusters with lower ratings and installs (e.g., Cluster 5, $50,000).

### Budget Allocation Breakdown

* Cluster 1: $70,000, aimed at boosting installs through search ads and influencer marketing.
* Cluster 2: $50,000, focused on search ads and referral campaigns.
* Cluster 3: $90,000, prioritized brand partnerships and content marketing.
* Cluster 4: $120,000, allocated to social media ads and app store optimization.
* Cluster 5: $50,000, focused on rebranding and targeted ads.



### Sales and Promotion Strategy

#### Cluster 1:

Low Installs, Low Ratings

* **Goal**: Improve app visibility, increase installs, and gather user feedback.
* **Strategy**:
  + **Search Ads (Google Play & App Store) (40%)**: Focus on increasing discoverability for users searching similar apps.
  + **Referral & Loyalty Campaigns (30%)**: Offer incentives for referrals to boost installs.
  + **Social Media Advertising (30%)**: Focus on niche communities on Reddit or Instagram to drive installs.
* **Estimated Budget**: $70,000
* **Expected Outcome**: 10%–20% increase in installs, improved app reviews, and higher user engagement.
* **Types of Apps**: New apps, small utilities, or niche apps that need user acquisition and brand awareness.

#### Cluster 2:

**Moderate Ratings, Low Installs**

* **Goal**: Build a strong presence in the market, attract users, and improve app credibility.
* **Strategy**:
  + **Search Ads (50%)**: Target users who are actively searching for apps in similar categories.
  + **Referral & Loyalty Campaigns (30%)**: Implement referral bonuses to increase word-of-mouth marketing.
  + **Influencer Marketing (20%)**: Partner with micro-influencers in niche app categories to improve reach.
* **Estimated Budget**: $50,000
* **Expected Outcome**: 15%–25% increase in installs, improved brand recognition, and enhanced user retention.
* **Types of Apps**: Growing apps in competitive niches such as lifestyle, health, or finance.

#### Cluster 3:

**High Ratings, High Reviews, High Installs**

* **Goal**: Sustain growth, engage loyal users, and increase brand loyalty.
* **Strategy**:
  + **Brand Partnerships (40%)**: Collaborate with complementary brands for co-marketing opportunities.
  + **Content Marketing (30%)**: Invest in high-quality content to engage users (videos, blogs, etc.).
  + **Social Media Advertising (30%)**: Utilize platforms like Facebook, Instagram, and Twitter for targeted ad campaigns.
* **Estimated Budget**: $90,000
* **Expected Outcome**: 20%–30% growth in installs, higher retention rates, and stronger community engagement.
* **Types of Apps**: Established apps in mainstream categories such as social media, entertainment, and productivity.

#### Cluster 4:

**High Ratings, Low Reviews/Installs**

* **Goal**: Drive more installs, generate reviews, and strengthen market position.
* **Strategy**:
  + **App Store Optimization (ASO) (40%)**: Focus on optimizing the app’s presence in the app stores to increase visibility.
  + **Social Media Ads (30%)**: Run ads targeting users who are likely to engage with the app.
  + **Referral Campaigns (30%)**: Launch referral programs to boost installs and incentivize user acquisition.
* **Estimated Budget**: $120,000
* **Expected Outcome**: 15%–25% increase in installs, more user reviews, and improved app store rankings.
* **Types of Apps**: Premium apps, entertainment, and niche apps with strong potential but low visibility.

#### Cluster 5:

**Low Ratings, Low Reviews, Low Installs**

* **Goal**: Improve app quality, increase user installs, and gather actionable feedback.
* **Strategy**:
  + **Rebranding & Redesign (40%)**: Invest in improving the app’s user interface and experience to attract new users.
  + **Targeted Ads (30%)**: Focus on targeting ads to users likely to appreciate the app’s value proposition.
  + **App Store Optimization (ASO) (30%)**: Enhance app store visibility and improve conversion rates.
* **Estimated Budget**: $50,000
* **Expected Outcome**: 5%–15% growth in installs, higher app ratings, and better user engagement.
* **Types of Apps**: New apps with low ratings and poor reception, often requiring updates and marketing to improve their reputation.

| Cluster | Search Ads | Referral Campaigns | Social Media Ads |
| --- | --- | --- | --- |
| 1 | 40% | 30% | 30% |
| 2 | 40% | 30% | 30% |
| 3 | 50% | 25% | 25% |
| 4 | 40% | 35% | 25% |
| 5 | 30% | 40% | 30% |

| Cluster | Total Budget | Search Ads | Referral Campaigns | Social Media Ads |
| --- | --- | --- | --- | --- |
| 1 | $70,000 | $28,000 | $21,000 | $21,000 |
| 2 | $50,000 | $20,000 | $15,000 | $15,000 |
| 3 | $90,000 | $45,000 | $22,500 | $22,500 |
| 4 | $120,000 | $48,000 | $42,000 | $30,000 |
| 5 | $50,000 | $15,000 | $20,000 | $15,000 |

## **IV. Results and Visualizations**

### Visualizations

#### Elbow Method Plot

#### **Silhouette Score Analysis**

#### 

#### 

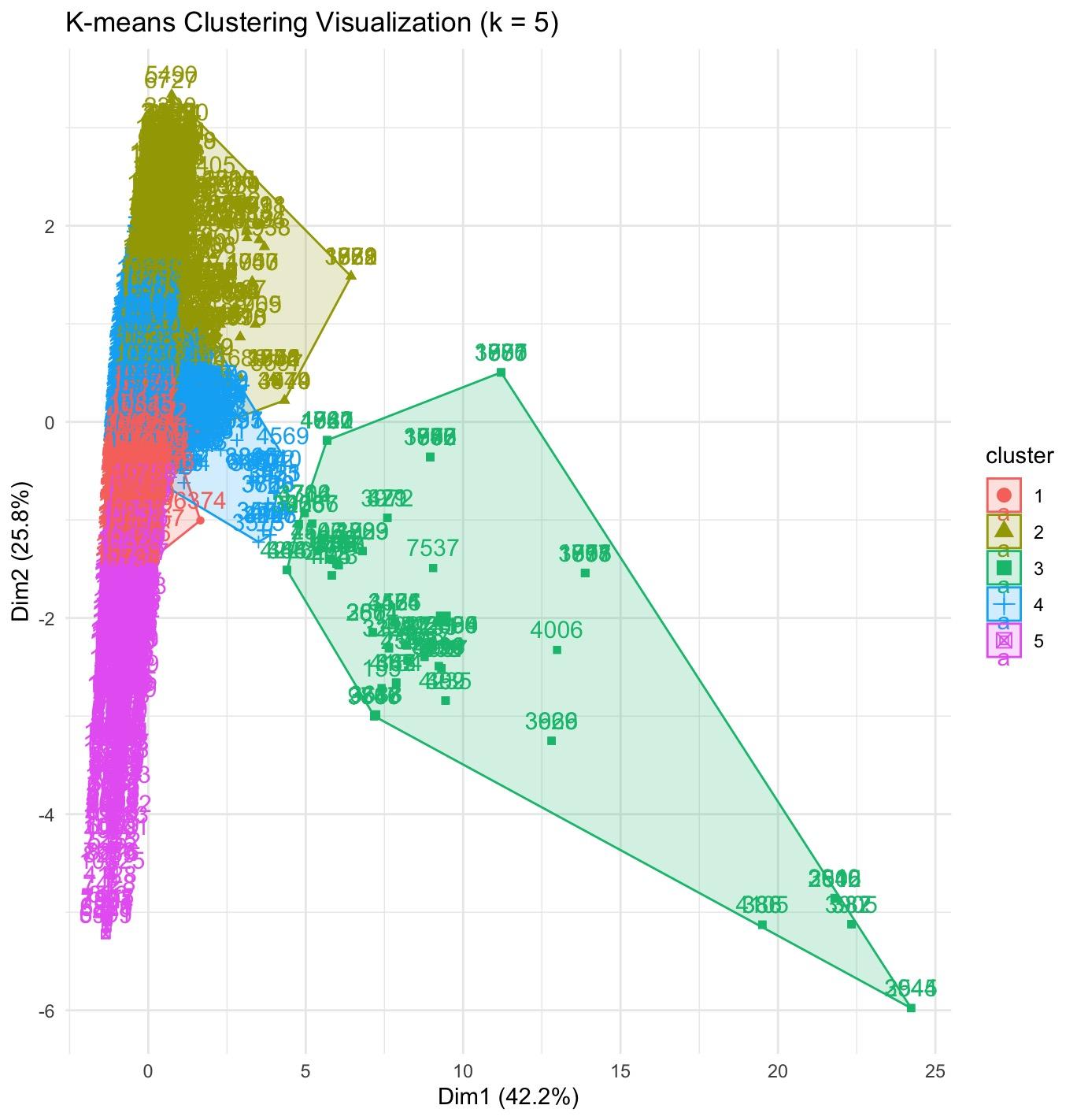
#### 

#### 

#### Gap Statistic Plot

#### 

#### **Cluster Plot**s

****

#### **Dendrogram of Apps**

#### 

#### 

#### 

#### 

#### Ratings Across Clusters Box Plot

#### 

#### 

#### 

#### 

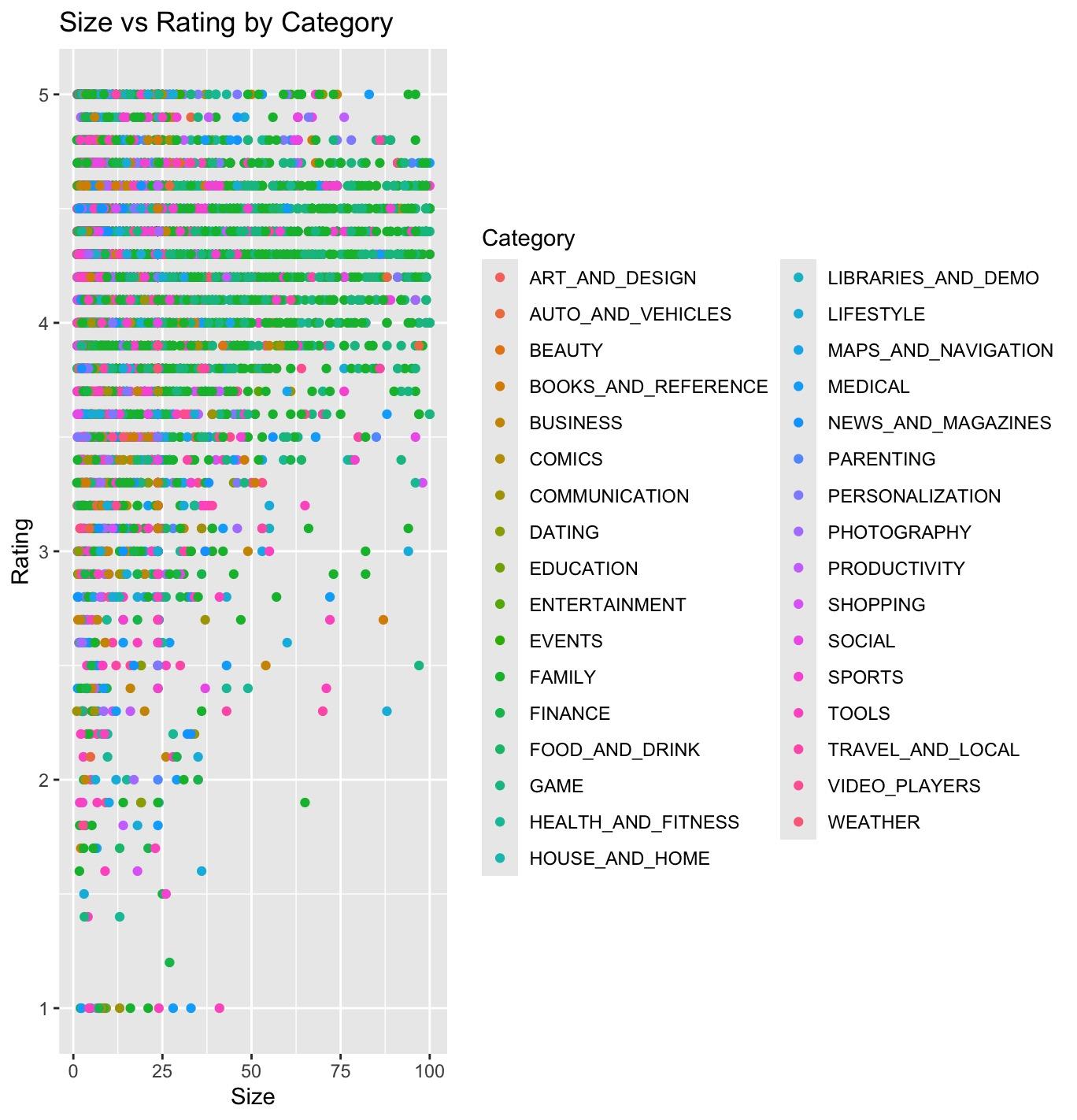
#### 

#### 

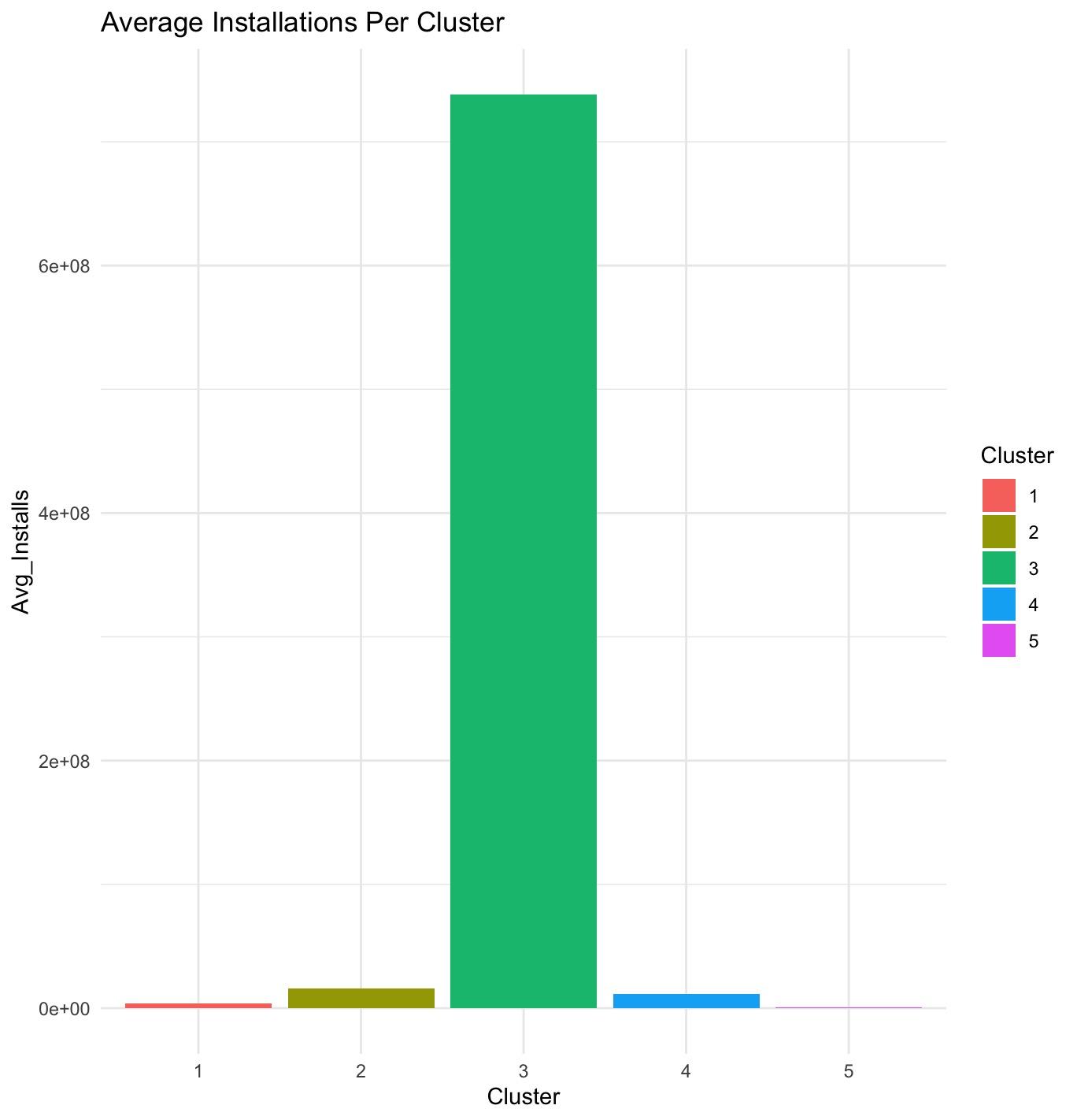
#### 

#### 

#### Size vs Rating by Category Scatter Plot

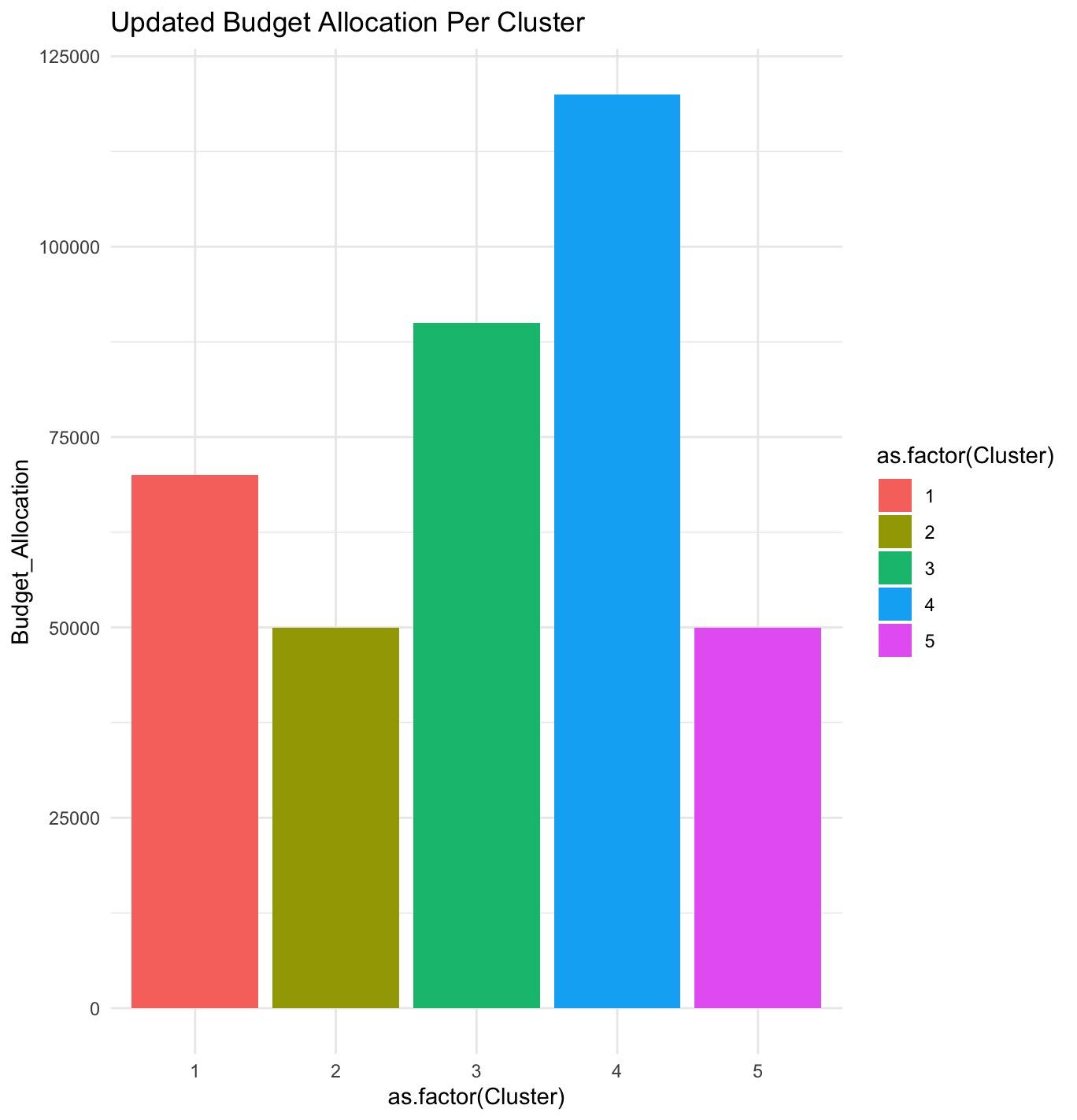
****

#### Average Installations per Cluster Bar Graph

****

#### **Rating vs Installs by Category Scatter Plot**

#### Budget Allocation Per Cluster

****

## **V. Conclusion**

This project analyzed app data, segmented it into five clusters based on ratings, reviews, installs, and price, and assigned marketing budgets to each. The goal was to identify promising clusters for investment.

Cluster 3 stands out with high ratings, reviews, and installs, making it the most suitable for targeted marketing efforts. Cluster 4, though similar in ratings, has lower installs, suggesting a need for better visibility. Cluster 1 and Cluster 2 represent low-performing apps, with Cluster 1 showing potential for improvement through strategic marketing. Cluster 5, the weakest, requires significant changes before it can show meaningful growth.

For future investment, focus on Cluster 3 and Cluster 4 for the most immediate growth, while Cluster 1 offers long-term potential. Cluster 2 and Cluster 5 should be approached cautiously and may need more resources or restructuring.

## VI. Appendix

### Code Used:

#### Without Outputs:

# Install the required packages

install.packages("factoextra")

install.packages("cluster")

install.packages("ggplot2")

install.packages("dplyr")

# Load the libraries

library(factoextra)

library(cluster)

library(ggplot2)

library(dplyr)

# Load the dataset

df <- read.csv("/Users/katherinefernandez/Downloads/googleplaystore.csv", header=TRUE, na.strings=(""))

# Remove missing values

df <- na.omit(df)

# Check for missing values (to confirm all NAs were removed)

sapply(df, function(x) sum(is.na(x)))

# Check for duplicate values in each column

sapply(df, function(x) length(unique(x)))

# Clean and convert the columns to numeric types

df$Installs <- as.numeric(gsub("[+,]", "", df$Installs))

df$Price <- as.numeric(gsub("[$]", "", df$Price))

df$Reviews <- as.numeric(df$Reviews)

df$Size <- as.numeric(gsub("[M]", "", df$Size))

# Replace missing "Size" values with the \*\*mean size\*\* of all apps

df$Size[is.na(df$Size)] <- mean(df$Size, na.rm=TRUE)

# Final clean dataset selection

df\_clean <- subset(df, select=c(1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13))

colnames(df\_clean)

# Check for any missing values (NA), NaN, or Inf in the dataset

summary(df\_clean)

# Remove rows with NA or Inf values from the selected columns (only the numeric columns you want to cluster on)

df\_clean <- df\_clean[complete.cases(df\_clean[, c(3, 4, 5, 6)]), ]

# Replace any Inf or NaN values (if any) with NA, and then remove those rows

df\_clean <- df\_clean[!apply(df\_clean[, c(3, 4, 5, 6)], 1, function(x) any(is.na(x) | is.infinite(x))), ]

# Scale the data (standardize the variables)

df\_clean\_scaled <- df\_clean

df\_clean\_scaled[, c(3, 4, 5, 6)] <- scale(df\_clean[, c(3, 4, 5, 6)])

# Ensure there are no missing or infinite values after scaling

summary(df\_clean\_scaled)

# 1) K-means clustering for k = 5

set.seed(123) # Set seed for reproducibility

kmeans\_result <- kmeans(df\_clean\_scaled[, c(3, 4, 5, 6)], centers = 5, nstart = 25)

# Add cluster labels to the dataframe

df\_clean$Cluster <- as.factor(kmeans\_result$cluster)

# Elbow Method (for determining optimal k)

fviz\_nbclust(df\_clean\_scaled[, c(3, 4, 5, 6)], kmeans, method = "wss") +

ggtitle("Elbow Method for Optimal k")

# Gap Statistics (visualized through fviz\_nbclust using gap\_stat method)

gap\_stat <- clusGap(df\_clean\_scaled[, c(3, 4, 5, 6)], FUNcluster = kmeans, K.max = 10, B = 100)

fviz\_gap\_stat(gap\_stat) + ggtitle("Gap Statistics for Optimal k")

# Silhouette Method (for determining optimal k)

fviz\_nbclust(df\_clean\_scaled[, c(3, 4, 5, 6)], kmeans, method = "silhouette") +

ggtitle("Silhouette Method for Optimal k")

# Boxplot of Ratings across Clusters

ggplot(df\_clean, aes(x = Cluster, y = Rating, fill = Cluster)) +

geom\_boxplot() +

ggtitle("Distribution of Ratings Across Clusters") +

theme\_minimal()

# Assuming cluster\_summary contains information about Avg\_Installs per cluster

# Bar plot for Installations per Cluster

ggplot(cluster\_summary, aes(x = Cluster, y = Avg\_Installs, fill = Cluster)) +

geom\_bar(stat="identity") +

ggtitle("Average Installations Per Cluster") +

theme\_minimal()

# Apply K-Means Clustering with 5 clusters

set.seed(123) # Random seed to ensure reproducibility

cluster1 <- kmeans(df\_clean\_scaled[, c(3, 4, 5, 6)], centers = 5, nstart = 10)

# Confusion matrix for k-means clustering

confusion\_matrix1 <- table(cluster1$cluster, df\_clean$Category)

correct\_predictions1 <- sum(diag(confusion\_matrix1))

total\_predictions1 <- sum(confusion\_matrix1)

accuracy1 <- correct\_predictions1 / total\_predictions1

print(paste("Accuracy:", accuracy1 \* 100))

# Assign clusters back to the dataset

df\_clean$Cluster <- as.factor(cluster1$cluster)

# Scatter plot with clusters

ggplot(df\_clean, aes(Installs, Rating, color = Cluster)) +

geom\_point() +

ggtitle("Clustering with k = 5")

# Distance matrix calculation (using Euclidean distance)

dist\_matrix <- dist(df\_clean[, c(3, 4, 5, 6)])

# Perform hierarchical clustering

hc <- hclust(dist\_matrix)

# Plot dendrogram

plot(hc, cex=0.6)

rect.hclust(hc, k = 5, border = 2:6)

# Visualize K-means clusters with fviz\_cluster

fviz\_cluster(list(data = df\_clean[, c(3, 4, 5, 6)], cluster = cluster1$cluster)) +

ggtitle("K-means Clustering Visualization (k = 5)") +

theme\_minimal()

# 2) Understand the characteristics of the groups by presenting their feature descriptions

# Scatter plot of Rating vs Installs colored by Category

ggplot(df\_clean, aes(Installs, Rating, color = Category)) +

geom\_point() +

ggtitle("Rating vs Installs by Category")

# Scatter plot of Size vs Rating colored by Category

ggplot(df\_clean, aes(Size, Rating, color = Category)) +

geom\_point() +

ggtitle("Size vs Rating by Category")

# Linear regression model to predict Installs based on Category, Reviews, Size, and Type

lr = lm(Installs ~ Category + Reviews + Size + Type, data = df\_clean)

summary(lr)

# Predict Installs for a specific set of parameters

data\_lr = data.frame(Category = 'PRODUCTIVITY', Reviews = 10000, Size = 25, Type = 'Free')

predict\_installs\_lr = predict(lr, data\_lr)

predict\_installs\_lr

# Assuming that $0.004 is paid for each download

assumed\_sales <- 0.004 \* predict\_installs\_lr

print(paste("Predicted Sales:", assumed\_sales))

# 3) Plan with an actual dollar value (assumed) and a strategy for promoting your applications

# via online advertisements or TV ad placement?

# Analyze Clusters

df\_clean$Cluster <- cluster1$cluster

df\_clean$Cluster <- as.factor(cluster1$cluster)

# Summarize cluster statistics for key features

# Check for any issues with pricing (e.g., zero prices) and handle them

df\_clean$Price[df\_clean$Price == 0] <- NA # Replacing zeroes with NA

df\_clean$Price[is.na(df\_clean$Price)] <- mean(df\_clean$Price, na.rm = TRUE) # Replace NAs with the mean

# Budget Allocation Strategy (with better handling of parameters)

cluster\_summary <- df\_clean %>%

group\_by(Cluster) %>%

summarise(

Avg\_Rating = mean(Rating, na.rm = TRUE),

Avg\_Reviews = mean(Reviews, na.rm = TRUE),

Avg\_Installs = mean(Installs, na.rm = TRUE),

Avg\_Price = mean(Price, na.rm = TRUE)

)

# Normalize the summary metrics by scaling them

cluster\_summary\_scaled <- cluster\_summary

cluster\_summary\_scaled[, c("Avg\_Rating", "Avg\_Reviews", "Avg\_Installs", "Avg\_Price")] <- scale(cluster\_summary[, c("Avg\_Rating", "Avg\_Reviews", "Avg\_Installs", "Avg\_Price")])

# Budget Allocation Strategy based on normalized values

cluster\_summary\_scaled <- cluster\_summary\_scaled %>%

mutate(

Budget\_Allocation = case\_when(

# Extremely high metrics (top 5% of normalized values)

Avg\_Rating > 0.95 & Avg\_Reviews > 0.95 & Avg\_Installs > 0.95 ~ 200000,

# Very high rating, high reviews, moderate installs

Avg\_Rating > 0.90 & Avg\_Reviews > 0.85 & Avg\_Installs > 0.85 ~ 180000,

# High rating, massive installs, super low price

Avg\_Rating > 0.85 & Avg\_Installs > 0.90 & Avg\_Price < 0.05 ~ 170000,

# Massive installs, low price, very high rating

Avg\_Installs > 0.85 & Avg\_Price < 0.05 & Avg\_Rating > 0.80 ~ 160000,

# High rating, high price, lower installs

Avg\_Rating > 0.85 & Avg\_Price > 0.70 & Avg\_Installs < 0.30 ~ 150000,

# Good rating, moderate installs, moderate price

Avg\_Rating > 0.75 & Avg\_Installs > 0.40 & Avg\_Price > 0.30 ~ 140000,

# Low price, low installs, moderate rating

Avg\_Price < 0.10 & Avg\_Installs < 0.25 & Avg\_Rating > 0.70 ~ 120000,

# Good rating, low installs, high price

Avg\_Rating > 0.75 & Avg\_Installs < 0.15 & Avg\_Price > 0.90 ~ 110000,

# Low rating, but very high installs

Avg\_Rating < 0.60 & Avg\_Installs > 0.90 ~ 90000,

# Low installs, very high price

Avg\_Installs < 0.15 & Avg\_Price > 0.90 ~ 70000,

# Default fallback for all other combinations

TRUE ~ 50000

)

)

# Ensure no NAs or zeroes in the Budget\_Allocation column

cluster\_summary\_scaled[is.na(cluster\_summary\_scaled$Budget\_Allocation), "Budget\_Allocation"] <- 30000 # Replace NAs with default value

cluster\_summary\_scaled$Budget\_Allocation[cluster\_summary\_scaled$Budget\_Allocation == 0] <- 30000 # Replace 0s with default value

# Print the updated summary table

print(cluster\_summary\_scaled)

# Plot the updated budget allocation per cluster

ggplot(cluster\_summary\_scaled, aes(x = as.factor(Cluster), y = Budget\_Allocation, fill = as.factor(Cluster))) +

geom\_bar(stat = "identity") +

ggtitle("Updated Budget Allocation Per Cluster") +

theme\_minimal()

# Final check to ensure all NAs or zeroes are handled

cluster\_summary\_scaled[is.na(cluster\_summary\_scaled$Budget\_Allocation), "Budget\_Allocation"] <- 30000 # Replace NAs with default value

cluster\_summary\_scaled$Budget\_Allocation[cluster\_summary\_scaled$Budget\_Allocation == 0] <- 30000 # Replace 0s with default value

# Print final budget allocation summary

print(cluster\_summary\_scaled)

#### **Full Code with Outputs:**

> library(factoextra)

> library(cluster)

> library(ggplot2)

> library(dplyr)

>

>

> # Load the dataset

> df <- read.csv("/Users/katherinefernandez/Downloads/googleplaystore.csv", header=TRUE, na.strings=(""))

>

> # Remove missing values

> df <- na.omit(df)

>

> # Check for missing values (to confirm all NAs were removed)

> sapply(df, function(x) sum(is.na(x)))

App Category Rating Reviews Size Installs

0 0 0 0 0 0

Type Price Content.Rating Genres Last.Updated Current.Ver

0 0 0 0 0 0

Android.Ver

0

>

> # Check for duplicate values in each column

> sapply(df, function(x) length(unique(x)))

App Category Rating Reviews Size Installs

8195 33 39 5991 413 19

Type Price Content.Rating Genres Last.Updated Current.Ver

2 73 6 115 1299 2639

Android.Ver

32

>

> # Clean and convert the columns to numeric types

> df$Installs <- as.numeric(gsub("[+,]", "", df$Installs))

> df$Price <- as.numeric(gsub("[$]", "", df$Price))

> df$Reviews <- as.numeric(df$Reviews)

> df$Size <- as.numeric(gsub("[M]", "", df$Size))

Warning message:

NAs introduced by coercion

>

> # Replace missing "Size" values with the \*\*mean size\*\* of all apps

> df$Size[is.na(df$Size)] <- mean(df$Size, na.rm=TRUE)

>

> # Final clean dataset selection

> df\_clean <- subset(df, select=c(1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13))

> colnames(df\_clean)

[1] "App" "Category" "Rating" "Reviews" "Size" "Installs"

[7] "Type" "Price" "Genres" "Last.Updated" "Current.Ver" "Android.Ver"

>

> # Check for any missing values (NA), NaN, or Inf in the dataset

> summary(df\_clean)

App Category Rating Reviews Size

Length:9365 Length:9365 Min. :1.000 Min. : 1 Min. : 1.00

Class :character Class :character 1st Qu.:4.000 1st Qu.: 186 1st Qu.: 7.50

Mode :character Mode :character Median :4.300 Median : 5928 Median : 23.00

Mean :4.192 Mean : 514103 Mean : 23.74

3rd Qu.:4.500 3rd Qu.: 81543 3rd Qu.: 27.00

Max. :5.000 Max. :78158306 Max. :100.00

Installs Type Price Genres Last.Updated

Min. :1.00e+00 Length:9365 Min. : 0.000 Length:9365 Length:9365

1st Qu.:1.00e+04 Class :character 1st Qu.: 0.000 Class :character Class :character

Median :5.00e+05 Mode :character Median : 0.000 Mode :character Mode :character

Mean :1.79e+07 Mean : 0.961

3rd Qu.:5.00e+06 3rd Qu.: 0.000

Max. :1.00e+09 Max. :400.000

Current.Ver Android.Ver

Length:9365 Length:9365

Class :character Class :character

Mode :character Mode :character

>

> # Remove rows with NA or Inf values from the selected columns (only the numeric columns you want to cluster on)

> df\_clean <- df\_clean[complete.cases(df\_clean[, c(3, 4, 5, 6)]), ]

>

> # Replace any Inf or NaN values (if any) with NA, and then remove those rows

> df\_clean <- df\_clean[!apply(df\_clean[, c(3, 4, 5, 6)], 1, function(x) any(is.na(x) | is.infinite(x))), ]

>

> # Scale the data (standardize the variables)

> df\_clean\_scaled <- df\_clean

> df\_clean\_scaled[, c(3, 4, 5, 6)] <- scale(df\_clean[, c(3, 4, 5, 6)])

>

> # Ensure there are no missing or infinite values after scaling

> summary(df\_clean\_scaled)

App Category Rating Reviews Size

Length:9365 Length:9365 Min. :-6.1947 Min. :-0.1635 Min. :-1.0849

Class :character Class :character 1st Qu.:-0.3722 1st Qu.:-0.1634 1st Qu.:-0.7747

Mode :character Mode :character Median : 0.2101 Median :-0.1616 Median :-0.0351

Mean : 0.0000 Mean : 0.0000 Mean : 0.0000

3rd Qu.: 0.5982 3rd Qu.:-0.1376 3rd Qu.: 0.1558

Max. : 1.5686 Max. :24.6944 Max. : 3.6391

Installs Type Price Genres Last.Updated

Min. :-0.1962 Length:9365 Min. : 0.000 Length:9365 Length:9365

1st Qu.:-0.1961 Class :character 1st Qu.: 0.000 Class :character Class :character

Median :-0.1907 Mode :character Median : 0.000 Mode :character Mode :character

Mean : 0.0000 Mean : 0.961

3rd Qu.:-0.1414 3rd Qu.: 0.000

Max. :10.7636 Max. :400.000

Current.Ver Android.Ver

Length:9365 Length:9365

Class :character Class :character

Mode :character Mode :character

>

> # 1) K-means clustering for k = 5

> set.seed(123) # Set seed for reproducibility

> kmeans\_result <- kmeans(df\_clean\_scaled[, c(3, 4, 5, 6)], centers = 5, nstart = 25)

>

> # Add cluster labels to the dataframe

> df\_clean$Cluster <- as.factor(kmeans\_result$cluster)

>

>

> # Apply K-Means Clustering with 5 clusters

> set.seed(123) # Random seed to ensure reproducibility

> cluster1 <- kmeans(df\_clean\_scaled[, c(3, 4, 5, 6)], centers = 5, nstart = 10)

>

> # Confusion matrix for k-means clustering

> confusion\_matrix1 <- table(cluster1$cluster, df\_clean$Category)

> correct\_predictions1 <- sum(diag(confusion\_matrix1))

> total\_predictions1 <- sum(confusion\_matrix1)

> accuracy1 <- correct\_predictions1 / total\_predictions1

> print(paste("Accuracy:", accuracy1 \* 100))

[1] "Accuracy: 1.97544046983449"

>

> # Assign clusters back to the dataset

> df\_clean$Cluster <- as.factor(cluster1$cluster)

>

>

>

> # Distance matrix calculation (using Euclidean distance)

> dist\_matrix <- dist(df\_clean[, c(3, 4, 5, 6)])

>

> # Perform hierarchical clustering

> hc <- hclust(dist\_matrix)

>

> # Plot dendrogram

> plot(hc, cex=0.6)

> rect.hclust(hc, k = 5, border = 2:6)

>

> # Visualize K-means clusters with fviz\_cluster

> fviz\_cluster(list(data = df\_clean[, c(3, 4, 5, 6)], cluster = cluster1$cluster)) +

+ ggtitle("K-means Clustering Visualization (k = 5)") +

+ theme\_minimal()

>

> # 2) Understand the characteristics of the groups by presenting their feature descriptions

> # Scatter plot of Rating vs Installs colored by Category

> ggplot(df\_clean, aes(Installs, Rating, color = Category)) +

+ geom\_point() +

+ ggtitle("Rating vs Installs by Category")

>

> # Scatter plot of Size vs Rating colored by Category

> ggplot(df\_clean, aes(Size, Rating, color = Category)) +

+ geom\_point() +

+ ggtitle("Size vs Rating by Category")

>

> # Linear regression model to predict Installs based on Category, Reviews, Size, and Type

> lr = lm(Installs ~ Category + Reviews + Size + Type, data = df\_clean)

> summary(lr)

Call:

lm(formula = Installs ~ Category + Reviews + Size + Type, data = df\_clean)

Residuals:

Min 1Q Median 3Q Max

-724295413 -8779690 -2231821 -407057 964636786

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 9.865e+05 8.777e+06 0.112 0.910507

CategoryAUTO\_AND\_VEHICLES -1.907e+06 1.192e+07 -0.160 0.872881

CategoryBEAUTY -1.604e+06 1.379e+07 -0.116 0.907398

CategoryBOOKS\_AND\_REFERENCE 6.813e+06 1.017e+07 0.670 0.503132

CategoryBUSINESS 6.217e+05 9.617e+06 0.065 0.948456

CategoryCOMICS -2.107e+06 1.260e+07 -0.167 0.867245

CategoryCOMMUNICATION 5.275e+07 9.572e+06 5.511 3.67e-08 \*\*\*

CategoryDATING -1.404e+06 1.006e+07 -0.140 0.888998

CategoryEDUCATION -1.252e+06 1.037e+07 -0.121 0.903928

CategoryENTERTAINMENT 9.675e+06 1.043e+07 0.927 0.353749

CategoryEVENTS -1.778e+06 1.351e+07 -0.132 0.895293

CategoryFAMILY -6.584e+05 8.938e+06 -0.074 0.941283

CategoryFINANCE -2.721e+05 9.569e+06 -0.028 0.977316

CategoryFOOD\_AND\_DRINK -1.403e+06 1.098e+07 -0.128 0.898336

CategoryGAME 2.470e+06 9.077e+06 0.272 0.785531

CategoryHEALTH\_AND\_FITNESS 7.252e+05 9.641e+06 0.075 0.940045

CategoryHOUSE\_AND\_HOME -9.798e+05 1.181e+07 -0.083 0.933874

CategoryLIBRARIES\_AND\_DEMO -1.838e+06 1.230e+07 -0.149 0.881215

CategoryLIFESTYLE -7.033e+05 9.588e+06 -0.073 0.941533

CategoryMAPS\_AND\_NAVIGATION -6.022e+05 1.073e+07 -0.056 0.955259

CategoryMEDICAL -4.020e+05 9.528e+06 -0.042 0.966346

CategoryNEWS\_AND\_MAGAZINES 2.594e+07 9.860e+06 2.631 0.008520 \*\*

CategoryPARENTING -1.852e+06 1.312e+07 -0.141 0.887741

CategoryPERSONALIZATION 1.879e+06 9.600e+06 0.196 0.844786

CategoryPHOTOGRAPHY 1.777e+07 9.583e+06 1.854 0.063746 .

CategoryPRODUCTIVITY 3.272e+07 9.505e+06 3.443 0.000578 \*\*\*

CategorySHOPPING 2.679e+06 9.840e+06 0.272 0.785438

CategorySOCIAL 8.513e+06 9.772e+06 0.871 0.383724

CategorySPORTS -7.214e+05 9.587e+06 -0.075 0.940017

CategoryTOOLS 7.671e+06 9.125e+06 0.841 0.400585

CategoryTRAVEL\_AND\_LOCAL 2.300e+07 9.900e+06 2.323 0.020176 \*

CategoryVIDEO\_PLAYERS 2.433e+07 1.032e+07 2.357 0.018451 \*

CategoryWEATHER 7.367e+05 1.184e+07 0.062 0.950398

Reviews 1.814e+01 2.333e-01 77.741 < 2e-16 \*\*\*

Size 6.594e+04 3.768e+04 1.750 0.080121 .

TypePaid -7.465e+06 2.879e+06 -2.593 0.009530 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 68990000 on 9329 degrees of freedom

Multiple R-squared: 0.4305, Adjusted R-squared: 0.4283

F-statistic: 201.5 on 35 and 9329 DF, p-value: < 2.2e-16

>

> # Predict Installs for a specific set of parameters

> data\_lr = data.frame(Category = 'PRODUCTIVITY', Reviews = 10000, Size = 25, Type = 'Free')

> predict\_installs\_lr = predict(lr, data\_lr)

> predict\_installs\_lr

1

35541334

>

> # Assuming that $0.004 is paid for each download

> assumed\_sales <- 0.004 \* predict\_installs\_lr

> print(paste("Predicted Sales:", assumed\_sales))

[1] "Predicted Sales: 142165.336567679"

>

> # 3) Plan with an actual dollar value (assumed) and a strategy for promoting your applications

> # via online advertisements or TV ad placement?

>

> # Analyze Clusters

> df\_clean$Cluster <- cluster1$cluster

> df\_clean$Cluster <- as.factor(cluster1$cluster)

>

> # Summarize cluster statistics for key features

> # Check for any issues with pricing (e.g., zero prices) and handle them

> df\_clean$Price[df\_clean$Price == 0] <- NA # Replacing zeroes with NA

> df\_clean$Price[is.na(df\_clean$Price)] <- mean(df\_clean$Price, na.rm = TRUE) # Replace NAs with the mean

>

> # Budget Allocation Strategy (with better handling of parameters)

> cluster\_summary <- df\_clean %>%

+ group\_by(Cluster) %>%

+ summarise(

+ Avg\_Rating = mean(Rating, na.rm = TRUE),

+ Avg\_Reviews = mean(Reviews, na.rm = TRUE),

+ Avg\_Installs = mean(Installs, na.rm = TRUE),

+ Avg\_Price = mean(Price, na.rm = TRUE)

+ )

>

> # Normalize the summary metrics by scaling them

> cluster\_summary\_scaled <- cluster\_summary

> cluster\_summary\_scaled[, c("Avg\_Rating", "Avg\_Reviews", "Avg\_Installs", "Avg\_Price")] <- scale(cluster\_summary[, c("Avg\_Rating", "Avg\_Reviews", "Avg\_Installs", "Avg\_Price")])

>

> # Budget Allocation Strategy based on normalized values

> cluster\_summary\_scaled <- cluster\_summary\_scaled %>%

+ mutate(

+ Budget\_Allocation = case\_when(

+ # Extremely high metrics (top 5% of normalized values)

+ Avg\_Rating > 0.95 & Avg\_Reviews > 0.95 & Avg\_Installs > 0.95 ~ 200000,

+

+ # Very high rating, high reviews, moderate installs

+ Avg\_Rating > 0.90 & Avg\_Reviews > 0.85 & Avg\_Installs > 0.85 ~ 180000,

+

+ # High rating, massive installs, super low price

+ Avg\_Rating > 0.85 & Avg\_Installs > 0.90 & Avg\_Price < 0.05 ~ 170000,

+

+ # Massive installs, low price, very high rating

+ Avg\_Installs > 0.85 & Avg\_Price < 0.05 & Avg\_Rating > 0.80 ~ 160000,

+

+ # High rating, high price, lower installs

+ Avg\_Rating > 0.85 & Avg\_Price > 0.70 & Avg\_Installs < 0.30 ~ 150000,

+

+ # Good rating, moderate installs, moderate price

+ Avg\_Rating > 0.75 & Avg\_Installs > 0.40 & Avg\_Price > 0.30 ~ 140000,

+

+ # Low price, low installs, moderate rating

+ Avg\_Price < 0.10 & Avg\_Installs < 0.25 & Avg\_Rating > 0.70 ~ 120000,

+

+ # Good rating, low installs, high price

+ Avg\_Rating > 0.75 & Avg\_Installs < 0.15 & Avg\_Price > 0.90 ~ 110000,

+

+ # Low rating, but very high installs

+ Avg\_Rating < 0.60 & Avg\_Installs > 0.90 ~ 90000,

+

+ # Low installs, very high price

+ Avg\_Installs < 0.15 & Avg\_Price > 0.90 ~ 70000,

+

+ # Default fallback for all other combinations

+ TRUE ~ 50000

+ )

+ )

>

> # Ensure no NAs or zeroes in the Budget\_Allocation column

> cluster\_summary\_scaled[is.na(cluster\_summary\_scaled$Budget\_Allocation), "Budget\_Allocation"] <- 30000 # Replace NAs with default value

> cluster\_summary\_scaled$Budget\_Allocation[cluster\_summary\_scaled$Budget\_Allocation == 0] <- 30000 # Replace 0s with default value

>

> # Print the updated summary table

> print(cluster\_summary\_scaled)

# A tibble: 5 × 6

Cluster Avg\_Rating Avg\_Reviews Avg\_Installs Avg\_Price Budget\_Allocation

<fct> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1 -0.0388 -0.477 -0.461 1.63 70000

2 2 0.463 -0.389 -0.422 -0.914 50000

3 3 0.477 1.79 1.79 0.0609 90000

4 4 0.804 -0.438 -0.436 -0.703 120000

5 5 -1.71 -0.483 -0.470 -0.0752 50000

>

> # Plot the updated budget allocation per cluster

> ggplot(cluster\_summary\_scaled, aes(x = as.factor(Cluster), y = Budget\_Allocation, fill = as.factor(Cluster))) +

+ geom\_bar(stat = "identity") +

+ ggtitle("Updated Budget Allocation Per Cluster") +

+ theme\_minimal()

>

> # Final check to ensure all NAs or zeroes are handled

> cluster\_summary\_scaled[is.na(cluster\_summary\_scaled$Budget\_Allocation), "Budget\_Allocation"] <- 30000 # Replace NAs with default value

> cluster\_summary\_scaled$Budget\_Allocation[cluster\_summary\_scaled$Budget\_Allocation == 0] <- 30000 # Replace 0s with default value

>

> # Print final budget allocation summary

> print(cluster\_summary\_scaled)

# A tibble: 5 × 6

Cluster Avg\_Rating Avg\_Reviews Avg\_Installs Avg\_Price Budget\_Allocation

<fct> <dbl> <dbl> <dbl> <dbl> <dbl>

1 1 -0.0388 -0.477 -0.461 1.63 70000

2 2 0.463 -0.389 -0.422 -0.914 50000

3 3 0.477 1.79 1.79 0.0609 90000

4 4 0.804 -0.438 -0.436 -0.703 120000

5 5 -1.71 -0.483 -0.470 -0.0752 50000

>