

Unlocking Mobile Application Insights: A Comprehensive Data Analysis

Unset

--Data Cleaning & Data Wrangling:

SELECT

COUNT(*)

FROM

`data-analysis-389112.ecommerce.new_google_app_dat`; --29,456 apps (rows)

--Check if we have duplicate app id's:

SELECT

COUNT(DISTINCT appId)

FROM

`data-analysis-389112.ecommerce.new_google_app_dat`;--Also = 29,456 --> No!

--

SELECT

appId

FROM

`data-analysis-389112.ecommerce.new_google_app_dat`

GROUP BY appId

HAVING COUNT(appId) > 1; --No!

--Data Cleaning CTE:

WITH table_cleaned AS

(SELECT

appId,-- appId: Google Play ID.

developer, -- developer: Developer's name.

developerWebsite, -- developerWebsite: Developer's website.

free, -- free: Indicates whether the app is free or paid. --1 == the app is

free

genre, -- genre: App's genre.

```

cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
*
FROM table_cleaned;

-- Calculate summary statistics (e.g., mean, median, standard deviation) for
'inAppProductPrice' and 'minInstalls.'
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.

```

```

ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice
FROM table_cleaned;

```

```

-----
-----
--Slide 1. Introduction: free Vs paid apps:
-----
-----

```

```

--statistics for free apps:
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.

```

```

developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned

```

```

WHERE free = 1
;

--statistics for paid apps:
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,

```

```

STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE free = 0
;

-- Create a bar chart to visualize the distribution of 'free' (free vs. paid)
apps.
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,

```

```

cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT free,
COUNT(*) AS num_of_apps
FROM table_cleaned
GROUP BY free
ORDER BY num_of_apps DESC
;

-- Create a bar chart to visualize the distribution of 'free' (free vs. paid)
apps. in terms of the average number of installations
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,

```

```

cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT free,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs
FROM table_cleaned
GROUP BY free
;

-- Calculate the average price of in-app products for both free and paid apps.
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT free,

```



```

COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY free
;

```

--Slide 2. Genre Trends and Popularity:

-- Create a bar chart to visualize the distribution of 'genre.'

```

WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.

```

```

cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
COUNT(*) AS num_of_apps
FROM table_cleaned
GROUP BY genre
ORDER BY num_of_apps DESC
;

-- Determine the total number of installs for each genre and subcategory. --
Genre VS Total number of app downloads
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.

```

```

cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY genre
ORDER BY total_num_of_installs DESC
;

```

```

--Genre VS Average Review Score

-- Determine the total number of installs for each genre and subcategory.
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,

```

```

APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY genre
ORDER BY mean_review_score DESC
;

```

--Slide 3. Diving Deeper: Developer Insights:

-- Identify the most prevalent genre.

```

WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,

```

```

summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY genre
ORDER BY total_num_of_installs DESC
; --Business

-- Identify the top 5 developers with the highest number of apps.

WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.

```

```

free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT developer,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY developer

```

```

ORDER BY total_num_of_installs DESC
; --Zoom.us is the top developer!

--These are the most consistent developers, they have achieved over 10,000
downloads while maintaining perfect 5 star reviews:

WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT developer,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,

```



```

APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY developer
HAVING mean_review_score = 5 AND total_num_of_installs >= 10000
ORDER BY total_num_of_installs DESC
; --only 18 developers

--their percentage:
SELECT
(SELECT COUNT(*))
FROM(
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,

```

```

score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT developer,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY developer
HAVING mean_review_score = 5 AND total_num_of_installs >= 10000)) / (SELECT
COUNT(DISTINCT developer)
FROM `data-analysis-389112.ecommerce.new_google_app_dat`) * 100; --0.116%

```

--Slide 4. Diving Deeper: Developer Insights:

```

--Exploring Seasonal trends:
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT EXTRACT(MONTH FROM releaseDate) AS release_month,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,

```

```

ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY release_month
ORDER BY release_month
;

--Effect of In - App advertisement

--On Number of Installs:
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,

```

```

cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT containsAds,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY containsAds
ORDER BY total_num_of_installs DESC
;

```

--Effect on the average Price of in-app products of free app:

```

WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,

```

```

adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT containsAds,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE free = 1
GROUP BY containsAds
; --0.19 for apps with ads compared to 0.31 without: 63.1579% higher for apps
without ads!

--Top 5 genres by average app installs for ad-supported apps:
WITH table_cleaned AS
(SELECT

```

```

appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,

```

```

SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE containsAds = 1
GROUP BY genre
ORDER BY mean_num_of_installs DESC
LIMIT 5;

--Disturbution of Ad - supported Apps across Genres:
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT genre,
SUM(minInstalls) AS total_num_of_installs,

```



```

(COUNT(CASE WHEN containsAds = 1 THEN 1 END) / (COUNT(*))) * 100 AS
percentage_of_ad_supported_apps
FROM table_cleaned
GROUP BY genre
ORDER BY percentage_of_ad_supported_apps DESC, total_num_of_installs DESC
;

```

-- Step 4: Building the Presentation with Common Data Presentation Mistakes to Avoid

-- Include visualizations, insights, and recommendations.

-- Structure the presentation with an introduction, data summary, key visualizations, insights, developer recommendations, and any pertinent additional information.

--Additional queries that were not used in the Presentation:

--statistics for contains ads:

WITH table_cleaned AS

(SELECT

appId, -- appId: Google Play ID.

developer, -- developer: Developer's name.

developerWebsite, -- developerWebsite: Developer's website.

free, -- free: Indicates whether the app is free or paid. --1 == the app is free

genre, -- genre: App's genre.

cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS

inAppProductPrice, -- inAppProductPrice: Price of in-app products.

```

minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT containsAds,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY containsAds
;

--statistics for contains ads with reviews:
WITH table_cleaned AS
(SELECT

```

```

appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT containsAds,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,

```

```

SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE score != 0
GROUP BY containsAds
;

```

```

--
SELECT SUM(minInstalls)
FROM `data-analysis-389112.ecommerce.new_google_app_dat`
WHERE reviews > 0; --906464045

```

```

--
SELECT SUM(minInstalls)
FROM `data-analysis-389112.ecommerce.new_google_app_dat`; --964294777

```

-- Assess the correlation between in-app product prices and the number of installs.

```

WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,

```

```

score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
CORR(inAppProductPrice, minInstalls) AS
corr_between_in_app_product_prices_and_the_number_of_installs
FROM table_cleaned
; --0.018900326443770651

```

```

-- Evaluate the profitability of paid apps based on in-app purchases.
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,

```

```

score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT inAppProductPrice,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE free = 0
GROUP BY inAppProductPrice
ORDER BY inAppProductPrice
;

--
-- Evaluate the profitability of paid apps based on in-app purchases.
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.

```

```

developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT inAppProductPrice,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned

```

```

WHERE free = 0
GROUP BY inAppProductPrice
ORDER BY inAppProductPrice
;

-- Evaluate the profitability of paid apps based on in-app purchases.
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT inAppProductPrice,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,

```



```

APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE free = 1
GROUP BY inAppProductPrice
ORDER BY inAppProductPrice
;
--
WITH table_cleaned AS
(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,

```

```

EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT inAppProductPrice,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY inAppProductPrice
ORDER BY mean_num_of_installs DESC
;

-- Investigate the connection between app pricing and installation numbers.
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.

```

```

minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT price,
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,
ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
GROUP BY price
ORDER BY price
;

--statistics for free reviewed apps:
WITH table_cleaned AS

```

```

(SELECT
appId, -- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice, -- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,
STDDEV(minInstalls) AS standard_deviation_num_of_installs,
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,
ROUND(AVG(price),2) AS mean_price,
STDDEV(price) AS standard_deviation_price,
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,

```

```

ROUND(AVG(score),2) AS mean_review_score,
SUM(minInstalls) AS total_num_of_installs
FROM table_cleaned
WHERE free = 1 AND score != 0
;

--statistics for paid reviewed apps:
WITH table_cleaned AS
(SELECT
appId,-- appId: Google Play ID.
developer, -- developer: Developer's name.
developerWebsite, -- developerWebsite: Developer's website.
free, -- free: Indicates whether the app is free or paid. --1 == the app is
free
genre, -- genre: App's genre.
cast(IFNULL(REGEXP_EXTRACT(inAppProductPrice, r'(\d+)'), '0') as int) AS
inAppProductPrice,-- inAppProductPrice: Price of in-app products.
minInstalls, -- minInstalls: Number of installs.
ratings,
adSupported,
containsAds,
reviews,
score,
summary,
title,
EXTRACT(DATE FROM ParseReleasedDayYear) AS releaseDate,
EXTRACT(DATE FROM dateUpdated) AS updateDate,
cast(ROUND(price,0) as int) AS price,
cast(ROUND(maxprice,0) as int) AS maxprice,
cast(IFNULL(REGEXP_EXTRACT(minprice, r'(\d+)'), '0') as int) AS minprice
FROM `data-analysis-389112.ecommerce.new_google_app_dat`)
SELECT
COUNT(*) AS num_of_apps,
ROUND(AVG(minInstalls),2) AS mean_num_of_installs,

```

```
STDDEV(minInstalls) AS standard_deviation_num_of_installs,  
APPROX_QUANTILES(minInstalls, 2)[OFFSET(1)] AS median_num_of_installs,  
ROUND(AVG(inAppProductPrice),2) AS mean_inAppProductPrice,  
STDDEV(inAppProductPrice) AS standard_deviation_inAppProductPrice,  
APPROX_QUANTILES(inAppProductPrice, 2)[OFFSET(1)] AS median_inAppProductPrice,  
ROUND(AVG(price),2) AS mean_price,  
STDDEV(price) AS standard_deviation_price,  
APPROX_QUANTILES(price, 2)[OFFSET(1)] AS median_price,  
ROUND(AVG(score),2) AS mean_review_score,  
SUM(minInstalls) AS total_num_of_installs  
FROM table_cleaned  
WHERE free = 0 AND score != 0  
;
```