**Unlocking Mobile Application Insights: A Comprehensive Data Analysis**

--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 achived over 10,000 downloads while maintaing 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 advertisment

--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

;

