databricks **Pyspark project**

(https://databricks.com)

Import libraries

```
# Importing necessary libraries and modules
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, IntegerType
from pyspark.sql.functions import *
```

```
# Loading data from CSV file into DataFrame
df = spark.read.load('/FileStore/tables/googleplaystore.csv',format='csv',sep=',',header='true',escape='"',inferschema='true')
```

```
# Counting the number of rows in the DataFrame
df.count()
```

```
Out[56]: 10841
```

```
# Displaying the first row of the DataFrame
df.show(1)
```

```
+--------------------+--------------+------+-------+----+--------+----+-----+--------------+------------+---------------+-----------
+------------+
|                 App|      Category|Rating|Reviews|Size|Installs|Type|Price|Content Rating|      Genres|   Last Updated|Current Ver
| Android Ver|
+--------------------+--------------+------+-------+----+--------+----+-----+--------------+------------+---------------+-----------
+------------+
|Photo Editor & Ca...|ART_AND_DESIGN|   4.1|    159| 19M| 10,000+|Free|    0|      Everyone|Art & Design|January 7, 2018|      1.0.0
|4.0.3 and up|
+--------------------+--------------+------+-------+----+--------+----+-----+--------------+------------+---------------+-----------
+------------+
only showing top 1 row
```

Check.schema

```
# Displaying the schema of the DataFrame
df.printSchema()
```

```
root
 |-- App: string (nullable = true)
 |-- Category: string (nullable = true)
 |-- Rating: double (nullable = true)
 |-- Reviews: string (nullable = true)
 |-- Size: string (nullable = true)
 |-- Installs: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Price: string (nullable = true)
 |-- Content Rating: string (nullable = true)
 |-- Genres: string (nullable = true)
 |-- Last Updated: string (nullable = true)
 |-- Current Ver: string (nullable = true)
 |-- Android Ver: string (nullable = true)
```

data cleaning

```
# Dropping unnecessary columns from the DataFrame
df=df.drop("size","Content Rating","Last Updated","Android ver","Current Ver")
```

```
# Displaying the Second row of the DataFrame
df.show(2)
```

```
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
|                 App|      Category|Rating|Reviews|Installs|Type|Price|              Genres|
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
|Photo Editor & Ca...|ART_AND_DESIGN|   4.1|    159| 10,000+|Free|    0|        Art & Design|
| Coloring book moana|ART_AND_DESIGN|   3.9|    967|500,000+|Free|    0|Art & Design;Pret...|
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
only showing top 2 rows
```

```
# Displaying the schema of the DataFrame
df.printSchema()
```

```
root
 |-- App: string (nullable = true)
 |-- Category: string (nullable = true)
 |-- Rating: double (nullable = true)
 |-- Reviews: string (nullable = true)
 |-- Installs: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Price: string (nullable = true)
 |-- Genres: string (nullable = true)
```

```
from pyspark.sql.functions import regexp_replace, col
from pyspark.sql.types import IntegerType

# Assuming `df` is your DataFrame
df = df.withColumn("Reviews", col("Reviews").cast(IntegerType())) \
    .withColumn("Installs", regexp_replace(col("Installs"), "[^0-9]", "")) \
    .withColumn("Installs", col("Installs").cast(IntegerType())) \
    .withColumn("Price", regexp_replace(col("Price"), "[$]", "")) \
    .withColumn("Price", col("Price").cast(IntegerType()))
```

```
# Displaying the Fifth row of the DataFrame
df.show(5)
```

```
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
|                 App|      Category|Rating|Reviews|Installs|Type|Price|              Genres|
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
|Photo Editor & Ca...|ART_AND_DESIGN|   4.1|    159|   10000|Free|    0|        Art & Design|
| Coloring book moana|ART_AND_DESIGN|   3.9|    967|  500000|Free|    0|Art & Design;Pret...|
|U Launcher Lite –...|ART_AND_DESIGN|   4.7|  87510| 5000000|Free|    0|        Art & Design|
|Sketch - Draw & P...|ART_AND_DESIGN|   4.5| 215644|50000000|Free|    0|        Art & Design|
|Pixel Draw - Numb...|ART_AND_DESIGN|   4.3|    967|  100000|Free|    0|Art & Design;Crea...|
+--------------------+--------------+------+-------+--------+----+-----+--------------------+
only showing top 5 rows
```

```
# Creating a temporary view for DataFrame to run SQL queries
df.createOrReplaceTempView("apps")
```

```
%sql select * from apps
```

Table

	App ▲	Category
1	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
2	Coloring book moana	ART_AND_DESIGN

3	U Launcher Lite – FREE Live Cool Themes, Hide Apps	ART_AND_DESIGN
4	Sketch - Draw & Paint	ART_AND_DESIGN
5	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN
6	Paper flowers instructions	ART_AND_DESIGN

10,000 rows | Truncated data

Top reviews give to the apps

```
%sql select App,sum(Reviews) from apps
group by 1
order by 2 desc
```

Table

	App	sum(Reviews)
1	Instagram	266241989
2	WhatsApp Messenger	207348304
3	Clash of Clans	179558781
4	Messenger – Text and Video Chat for Free	169932272
5	Subway Surfers	166331958
6	Candy Crush Saga	156993136

9,660 rows

Top 10 installs app

```
%sql select App,Type,sum(Installs) from apps
group by 1,2
order by 3 desc
```

Table

	App	Type	sum(Ins
1	Subway Surfers	Free	6000000
2	Instagram	Free	4000000
3	Google Drive	Free	4000000
4	Hangouts	Free	4000000
5	Google Photos	Free	4000000
6	Google News	Free	4000000

9,662 rows

Category wise distribution

```
%sql select category,sum(Installs) from apps
group by 1
order by 2 desc
```

Table

	category	sum(Installs)
1	GAME	35086024415
2	COMMUNICATION	32647276251
3	PRODUCTIVITY	14176091369
4	SOCIAL	14069867902
5	TOOLS	11452771915

6	FAMILY	10258263505
7	PHOTOGRAPHY	10088247655

34 rows

Top paid apps

Table

	App	sum(Price)
1	I'm Rich - Trump Edition	400
2	I am Rich Plus	399
3	I AM RICH PRO PLUS	399
4	I'm Rich/Eu sou Rico/أنا غني/我很有錢	399
5	I Am Rich Premium	399
6	most expensive app (H)	399
7	I Am Rich Pro	399

756 rows