# Day14_06_kaggle_google_eda

October 16, 2023

## 1 Muhammad Nauman Sair


- Kaggle
- Linkedin
- Github
- twitter-X
- Email: mnsair@live.com


### 1.1 Google App Store EDA

#### 1.1.1 To-Dos

1. Download the data Google_Play_Store_apps
2. Extract data from zip file
3. Import the Libraries
4. Read the data into python pandas
5. Crate EDA with ydata_profiling before wrangling
6. Understand the data
7. Assignments in this notebook
8. Crate EDA with ydata_profiling after wrangling

#### 1.1.2 Assignments in this notebook

1. Size Column:
    1. Varies with device to NaN
    2. Convert KBs to MBs
    3. dtype should be numeric or float64
    4. drop the Size column, and create new column name Size_MB
2. Price Column:
    1. drop $ sign
    2. dtype should be numeric or float64
3. Installs Column: bin the data for this column

#### 1.1.3 3. Import the Libraries

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

### 1.1.4 4. Read the data into python pandas

[ ]:
```
df = pd.read_csv('C:/Users/mnsai/Desktop/Python/AI-and-DS/data/googleplaystore.
 ↪csv')
# Please select the path where you unzip and save the data
```

### 1.1.5 5. EDA Ydata Profiling before wrangling

[ ]:
```
# Automated EDA to review and understand the Data before wrandgling
import ydata_profiling as yd
profile = yd.ProfileReport(df)
profile.to_file(output_file='C:/Users/mnsai/Desktop/Python/AI-and-DS/Output/
 ↪06_ydata_kaggle_google.html')
# This will create an HTML file at your selected location, preferred to open in␣
 ↪Google Chrome

# Note: ydata_profiling in not compatible with python 3.11 as of Oct 16,02023
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

### 1.1.6 6. Understand the data

[ ]:
```
# check the data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             10841 non-null  object
 1   Category        10840 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  int64
 4   Size            10841 non-null  object
 5   Installs        10841 non-null  object
 6   Type            10841 non-null  object
 7   Price           10841 non-null  object
 8   Content Rating  10841 non-null  object
 9   Genres          10840 non-null  object
 10  Last Updated    10841 non-null  object
 11  Current Ver     10833 non-null  object
 12  Android Ver     10839 non-null  object
```

```
dtypes: float64(1), int64(1), object(11)
memory usage: 1.1+ MB
```

[ ]:
```
# lets have a look at the data
df.head()
```

[ ]:
```
                                     App        Category  Rating  \
0                         Viber Messenger   COMMUNICATION     4.3
1           imo free video calls and chat   COMMUNICATION     4.3
2  Google Duo - High Quality Video Calls   COMMUNICATION     4.6
3  UC Browser - Fast Download Private & Secure  COMMUNICATION     4.5
4           imo free video calls and chat   COMMUNICATION     4.3

    Reviews                Size      Installs  Type Price Content Rating  \
0  11334799  Varies with device  500,000,000+  Free     0       Everyone
1   4785892                 11M  500,000,000+  Free     0       Everyone
2   2083237  Varies with device  500,000,000+  Free     0       Everyone
3  17712922                 40M  500,000,000+  Free     0           Teen
4   4785988                 11M  500,000,000+  Free     0       Everyone

          Genres Last Updated             Current Ver         Android Ver
0  Communication    18-Jul-18        Varies with device  Varies with device
1  Communication     8-Jun-18          9.8.000000010501          4.0 and up
2  Communication    31-Jul-18  37.1.206017801.DR37_RC14          4.4 and up
3  Communication     2-Aug-18                12.8.5.1121          4.0 and up
4  Communication     8-Jun-18          9.8.000000010501          4.0 and up
```

[ ]:
```
# Randomly pick the data to review it
# This also used to take sample from big data set
df.sample(10)
```

[ ]:
```
                                App            Category  Rating  Reviews  \
2202                          Fancy            SHOPPING     4.2    39735
9367  Nights at Cube Pizzeria 3D - 3              FAMILY     4.0    15875
8073                  HDWallpaper DK     PERSONALIZATION     NaN        0
9234                   AT&T U-verse              FAMILY     3.7    38606
5550             ZOOM Cloud Meetings            BUSINESS     4.4    31614
3235  Opera Browser: Fast and Secure       COMMUNICATION     4.4  2473509
4920                     Ah Ha Block              FAMILY     NaN        2
2050  BetterMe: Weight Loss Workouts  HEALTH_AND_FITNESS     4.2    14709
4877       Android P Style Icon Pack     PERSONALIZATION     5.0        1
6465              Timely Alarm Clock           LIFESTYLE     4.3   258717

                    Size      Installs  Type   Price Content Rating  \
2202                 18M    5,000,000+  Free       0           Teen
9367                 40M    1,000,000+  Free       0           Teen
8073                6.6M           10+  Free       0           Teen
9234                 96M    1,000,000+  Free       0       Everyone
```

```
5550                37M   10,000,000+  Free       0       Everyone
3235  Varies with device  100,000,000+  Free       0       Everyone
4920                16M          100+  Free       0       Everyone
2050                15M    5,000,000+  Free       0       Everyone
4877                60M          100+  Paid  $0.99        Everyone
6465               9.4M   10,000,000+  Free       0       Everyone

                Genres Last Updated       Current Ver         Android Ver
2202          Shopping    25-Jul-18            3.13.09          4.0 and up
9367        Simulation     3-Mar-17                 1.5          4.2 and up
8073   Personalization     5-Oct-17                   1          4.0 and up
9234     Entertainment    13-Jun-18         5.9.0.0031          4.1 and up
5550          Business    20-Jul-18    4.1.28165.0716          4.0 and up
3235     Communication    31-Jul-18  47.1.2249.129326  Varies with device
4920            Casual     9-Jul-17                   1          4.0 and up
2050  Health & Fitness    26-Jul-18               2.8.2          5.0 and up
4877   Personalization    16-Jun-18                   1          4.1 and up
6465         Lifestyle    25-Sep-17               1.3.1        4.0.3 and up
```

[ ]:
```
print(df.describe()) # This will only summarize the numeric data
```

```
             Rating       Reviews
count  9367.000000  1.084100e+04
mean      4.191513  4.441119e+05
std       0.515735  2.927629e+06
min       1.000000  0.000000e+00
25%       4.000000  3.800000e+01
50%       4.300000  2.094000e+03
75%       4.500000  5.476800e+04
max       5.000000  7.815831e+07
```

### 1.1.7 7. Assignments in this notebook

Based on the data info we understand that Size and Price columns should be numeric, instead of str/object.
For number of installs, the data is in thousands and ending with plus (+) sign, we can remvoe the + sign and convert create bins, or just convert the data to numeric as int or float.

#### Assignment 1 - Size Column

1. Varies with device to NaN
2. Convert KBs to MBs
3. dtype should be numeric or float64
4. drop the Size column, and create new column name Size_MB

[ ]:
```
# View the Size column infor
print(df['Size'].info())
print(df['Size'].value_counts())
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 10841 entries, 0 to 10840
Series name: Size
Non-Null Count  Dtype
--------------  -----
10841 non-null  object
dtypes: object(1)
memory usage: 84.8+ KB
None
Size
Varies with device    1695
11M                    198
12M                    196
14M                    194
13M                    191
                      …
173k                     1
597k                     1
809k                     1
411k                     1
885k                     1
Name: count, Length: 461, dtype: int64
```

Assigment 1 - Subpart 1: Convert the 'Varies with device" to NaN

[ ]:
```
df['Size'] = df['Size'].replace('Varies with device', np.nan)
print(df['Size'].value_counts())
print(df['Size'].isnull().sum()) # As you see there were 1695 values are␣
  ↪"Varies with Device", now shown as Null
```

```
Size
11M     198
12M     196
14M     194
13M     191
15M     184
       …
173k      1
597k      1
809k      1
411k      1
885k      1
Name: count, Length: 460, dtype: int64
1695
```

[ ]:
```
df['Size'].value_counts()
```

[ ]:
```
Size
11M     198
12M     196
14M     194
13M     191
15M     184
       …
173k      1
597k      1
809k      1
411k      1
885k      1
Name: count, Length: 460, dtype: int64
```

Assigment 1 -
2. Subpart 2: Convert the 'Varies with device" to NaN
3. Subpart 3: dtype should be numeric or float64

[ ]:
```
def convert_size_to_MB(size_str):
    if 'k' in size_str:
        return float(size_str.replace('k', '')) / 1024  # KB to MB
    elif 'M' in size_str:
        return float(size_str.replace('M', ''))  # MB to MB
    else:
        return float(size_str)
```

Assigment 1 -
4. Subpart 4: Create new column name Size_MB, and Drop the Size column

[ ]:
```
# Add a new Size column in KBs,Apply the conversion function to the 'Size'␣
 ↪column
df['Size_MB'] = df['Size'].astype(str).apply(convert_size_to_MB)

#drop the Size Column
df = df.drop('Size', axis=1)

# Dataframe info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             10841 non-null  object
 1   Category        10840 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  int64
 4   Installs        10841 non-null  object
```

```
 5   Type            10841 non-null  object
 6   Price           10841 non-null  object
 7   Content Rating  10841 non-null  object
 8   Genres          10840 non-null  object
 9   Last Updated    10841 non-null  object
 10  Current Ver     10833 non-null  object
 11  Android Ver     10839 non-null  object
 12  Size_MB         9146 non-null   float64
dtypes: float64(2), int64(1), object(10)
memory usage: 1.1+ MB
```

[ ]:
```
df.tail(10)
```

[ ]:

| | App | Category | Rating | Reviews | \ |
|---|---|---|---|---|---|
| 10831 | Pekalongan CJ | SOCIAL | NaN | 0 | |
| 10832 | CX Network | BUSINESS | NaN | 0 | |
| 10833 | Sweden Newspapers | NEWS_AND_MAGAZINES | NaN | 0 | |
| 10834 | Test Application DT 02 | ART_AND_DESIGN | NaN | 0 | |
| 10835 | EG \| Explore Folegandros | TRAVEL_AND_LOCAL | NaN | 0 | |
| 10836 | EP Cook Book | MEDICAL | NaN | 0 | |
| 10837 | Eu sou Rico | FINANCE | NaN | 0 | |
| 10838 | Eu Sou Rico | FINANCE | NaN | 0 | |
| 10839 | I'm Rich/Eu sou Rico/ / | LIFESTYLE | NaN | 0 | |
| 10840 | Command & Conquer: Rivals | FAMILY | NaN | 0 | |

| | Installs | Type | Price | Content Rating | Genres | Last Updated | \ |
|---|---|---|---|---|---|---|---|
| 10831 | 0+ | Free | 0 | Teen | Social | 21-Jul-18 | |
| 10832 | 0+ | Free | 0 | Everyone | Business | 6-Aug-18 | |
| 10833 | 0+ | Free | 0 | Everyone | News & Magazines | 7-Jul-18 | |
| 10834 | 0+ | Free | 0 | Everyone | Art & Design | 14-Mar-17 | |
| 10835 | 0+ | Paid | $3.99 | Everyone | Travel & Local | 22-Jan-17 | |
| 10836 | 0+ | Paid | $200.00 | Everyone | Medical | 26-Jul-15 | |
| 10837 | 0+ | Paid | $30.99 | Everyone | Finance | 9-Jan-18 | |
| 10838 | 0+ | Paid | $394.99 | Everyone | Finance | 11-Jul-18 | |
| 10839 | 0+ | Paid | $399.99 | Everyone | Lifestyle | 1-Dec-17 | |
| 10840 | 0 | Free | 0 | Everyone 10+ | Strategy | 28-Jun-18 | |

| | Current Ver | Android Ver | Size_MB |
|---|---|---|---|
| 10831 | 0.0.1 | 4.4 and up | 5.9 |
| 10832 | 1.3.1 | 4.1 and up | 10.0 |
| 10833 | 1.1 | 4.4 and up | 2.1 |
| 10834 | 4 | 4.2 and up | 1.2 |
| 10835 | 1.1.1 | 4.1 and up | 56.0 |
| 10836 | 1 | 3.0 and up | 3.2 |
| 10837 | 1 | 4.0 and up | 2.6 |
| 10838 | 1 | 4.0.3 and up | 1.4 |
| 10839 | MONEY | 4.1 and up | 40.0 |

```
10840  Varies with device  Varies with device      NaN
```

**Assignment 2-Price Column:**

1. drop $ sign
2. dtype should be numeric or float64

[ ]:
```
# Remove '$' sign and convert to float
df['Price'] = df['Price'].str.replace('$', '').astype(float)

# Display the DataFrame
df.tail(10)
```

[ ]:
```
                                   App             Category  Rating  Reviews  \
10831                     Pekalongan CJ               SOCIAL     NaN        0
10832                        CX Network             BUSINESS     NaN        0
10833                  Sweden Newspapers  NEWS_AND_MAGAZINES     NaN        0
10834            Test Application DT 02      ART_AND_DESIGN     NaN        0
10835          EG | Explore Folegandros    TRAVEL_AND_LOCAL     NaN        0
10836                       EP Cook Book             MEDICAL     NaN        0
10837                        Eu sou Rico             FINANCE     NaN        0
10838                        Eu Sou Rico             FINANCE     NaN        0
10839  I'm Rich/Eu sou Rico/    /             LIFESTYLE     NaN        0
10840          Command & Conquer: Rivals              FAMILY     NaN        0

      Installs  Type   Price Content Rating            Genres Last Updated  \
10831       0+  Free    0.00           Teen            Social    21-Jul-18
10832       0+  Free    0.00       Everyone          Business     6-Aug-18
10833       0+  Free    0.00       Everyone  News & Magazines     7-Jul-18
10834       0+  Free    0.00       Everyone      Art & Design    14-Mar-17
10835       0+  Paid    3.99       Everyone    Travel & Local    22-Jan-17
10836       0+  Paid  200.00       Everyone           Medical    26-Jul-15
10837       0+  Paid   30.99       Everyone           Finance     9-Jan-18
10838       0+  Paid  394.99       Everyone           Finance    11-Jul-18
10839       0+  Paid  399.99       Everyone         Lifestyle     1-Dec-17
10840        0  Free    0.00   Everyone 10+          Strategy    28-Jun-18

             Current Ver         Android Ver  Size_MB
10831              0.0.1         4.4 and up      5.9
10832              1.3.1         4.1 and up     10.0
10833                1.1         4.4 and up      2.1
10834                  4         4.2 and up      1.2
10835              1.1.1         4.1 and up     56.0
10836                  1         3.0 and up      3.2
10837                  1         4.0 and up      2.6
10838                  1       4.0.3 and up      1.4
10839              MONEY         4.1 and up     40.0
10840  Varies with device  Varies with device      NaN
```

[ ]:
```
# Rerun the df.info to check the executions
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             10841 non-null  object
 1   Category        10840 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  int64
 4   Installs        10841 non-null  object
 5   Type            10841 non-null  object
 6   Price           10841 non-null  float64
 7   Content Rating  10841 non-null  object
 8   Genres          10840 non-null  object
 9   Last Updated    10841 non-null  object
 10  Current Ver     10833 non-null  object
 11  Android Ver     10839 non-null  object
 12  Size_MB         9146 non-null   float64
dtypes: float64(3), int64(1), object(9)
memory usage: 1.1+ MB
```

Its been verified that `Size` column converted to MBs, Varies with device converted to NaN Price column converted to numeric and $ sign dropped

**Assignment - 3. Installs Column: bin the data for this column**

[ ]:
```
# Remove ',' and '+' and convert to integers
df['Installs'] = df['Installs'].str.replace(',', '').str.rstrip('+').astype(int)
```

[ ]:
```
# Understand the Installs Column
df['Installs'].value_counts()
```

[ ]:
```
Installs
1000000       1579
10000000      1252
100000        1169
10000         1054
1000           908
5000000        752
100            719
500000         539
50000          479
5000           477
100000000      409
10             386
500            330
```

```
50000000       289
50             205
5               82
500000000       72
1               67
1000000000      58
0               15
Name: count, dtype: int64
```

[ ]:
```
# binning of age columns into 7 categories
bins = [0,1000,100000,1000000,10000000,1000000000]
labels = ['UnderThousand','In100Ks','In1000ks','InMillions','InBillions']
df['Installs_Group'] = pd.cut(df['Installs'], bins, labels=labels)
# cut the `Installs` data into bins and add a new column name `Installs_Group`␣
 ↪into the data. adding a new column called feature enginerring
df.head()
```

[ ]:
```
                                        App       Category  Rating  \
0                           Viber Messenger  COMMUNICATION     4.3
1              imo free video calls and chat  COMMUNICATION     4.3
2      Google Duo - High Quality Video Calls  COMMUNICATION     4.6
3  UC Browser - Fast Download Private & Secure  COMMUNICATION     4.5
4              imo free video calls and chat  COMMUNICATION     4.3

    Reviews   Installs  Type  Price Content Rating         Genres  \
0  11334799  500000000  Free    0.0       Everyone  Communication
1   4785892  500000000  Free    0.0       Everyone  Communication
2   2083237  500000000  Free    0.0       Everyone  Communication
3  17712922  500000000  Free    0.0           Teen  Communication
4   4785988  500000000  Free    0.0       Everyone  Communication

  Last Updated               Current Ver         Android Ver  Size_MB  \
0    18-Jul-18        Varies with device  Varies with device      NaN
1     8-Jun-18          9.8.000000010501         4.0 and up     11.0
2    31-Jul-18  37.1.206017801.DR37_RC14         4.4 and up      NaN
3     2-Aug-18              12.8.5.1121         4.0 and up     40.0
4     8-Jun-18          9.8.000000010501         4.0 and up     11.0

  Installs_Group
0     InBillions
1     InBillions
2     InBillions
3     InBillions
4     InBillions
```

[ ]:
```
sns.histplot(df['Installs_Group']) #just incasxe if histogram failed to plot in␣
 ↪working notebokk, run the following function.
```

```
plt.savefig('Installs_Group_Hist.png') # This will create and save the␣
  ↪histogram in the same folder
```

[ ]:
```
<Axes: xlabel='Installs_Group', ylabel='Count'>
```

### 1.1.8 8. Crate EDA with ydata_profiling after wrangling

[ ]:
```
import ydata_profiling as yd
profile = yd.ProfileReport(df)
profile.to_file(output_file='C:/Users/mnsai/Desktop/Python/AI-and-DS/Output/
  ↪07_ydata_kaggle_google_CleanData.html')
#An HTML File save at directed location
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

Now one can comapare Two EDAs for better understanding, really helpful.