Google Play Project

May 23, 2019

0.0.1 Analyzing Google Play Store Data

Project Done By : Reem Alashhab

0.0.2 Introduction

Google Play Store is a huge digital media store as it offers music, books, movies, games, and etc. For the Android mobile developers, it might be sometimes a difficult decision to decide on a successful and profitable application to develop in the market. For this reason, I am going to investigate the data set for Google Play Store (Last Updated : 3/2/2019) to answer some important questions. The data set is pulled from Kaggle Website. Kaggle is an online website that acts as a community for data scientists and machine learners. It is owned by Google Company. Kaggle allows users to find and publish data sets through its website online. It also offers data science and machine learning competitions and many other services.

0.0.3 Project Phases

To complete this project successfully and efficiently, I will go through three data wrangling phases before presenting the final visualizations :

- Gathering Data.
- Assessing Data.
- Cleaning Data.

0.0.4 Project Requirements

- Installing Jupyter Notebook.

0.0.5 Table of Content

0.0.6 Data Dictionary

1. App:Application name.
2. Category: Category the app belongs to.
3. Rating: Overall user rating of the app (as when scraped).
4. ReviewsNumber: of user reviews for the app (as when scraped).
5. Size: Size of the app (as when scraped).
6. Installs: Number of user downloads/installs for the app (as when scraped).
7. Type: Paid or Free.
8. Price: Price of the app (as when scraped).
9. Content Rating: Age group the app is targeted at - Children / Mature 21+ / Adult.

Content Rating Categories :
Everyone
Content is generally suitable for all ages. May contain minimal cartoon, fantasy or mild violence and/or infrequent use of mild language.
Everyone 10+
Content is generally suitable for ages 10 and up. May contain more cartoon, fantasy or mild violence, mild language and/or minimal suggestive themes.
Teen
Content is generally suitable for ages 13 and up. May contain violence, suggestive themes, crude humor, minimal blood, simulated gambling and/or infrequent use of strong language.
Mature 17+
Content is generally suitable for ages 17 and up. May contain intense violence, blood and gore, sexual content and/or strong language.

10. Genres : An app can belong to multiple genres (apart from its main category). For eg, a musical family game will belong to Music, Game, Family genres.
11. Last UpdatedDate : when the app was last updated on Play Store (as when scraped).
12. Current Ver: Current version of the app available on Play Store (as when scraped).
13. Android VerMin: required Android version (as when scraped).

0.0.7 Gathering Data

Gathering data is the first step in the data wrangling process. It invloves obtaining the data. In this project, I added the data set of Google Play Store automaticly from the website and then added it to the Jupyter Environment because this data set is updated from time to time.

```
In [5]: # importing some python libraries required
        # through all the data analysis for the selected data set
        import pandas as pd
        import csv
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style('darkgrid')
        %matplotlib inline
```

```
In [194]: # reading the data set file
          google_store = pd.read_csv('googleplaystore.csv')
```

```
In [195]: # checking to see of the file is imported correclty
          # displaying the first four rows of the data set
          google_store.head()

Out[195]:                                                  App        Category  Rating  \
          0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
          1                                Coloring book moana  ART_AND_DESIGN     3.9
          2  U Launcher Lite  FREE Live Cool Themes, Hide ...  ART_AND_DESIGN     4.7
          3                              Sketch - Draw & Paint  ART_AND_DESIGN     4.5
          4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

            Reviews  Size     Installs  Type Price Content Rating  \
          0     159   19M      10,000+  Free     0       Everyone
          1     967   14M     500,000+  Free     0       Everyone
          2   87510  8.7M   5,000,000+  Free     0       Everyone
          3  215644   25M  50,000,000+  Free     0           Teen
          4     967  2.8M     100,000+  Free     0       Everyone

                               Genres      Last Updated         Current Ver  \
          0                  Art & Design   January 7, 2018               1.0.0
          1  Art & Design;Pretend Play  January 15, 2018               2.0.0
          2                  Art & Design    August 1, 2018               1.2.4
          3                  Art & Design      June 8, 2018  Varies with device
          4    Art & Design;Creativity     June 20, 2018                 1.1

              Android Ver
          0  4.0.3 and up
          1  4.0.3 and up
          2  4.0.3 and up
          3    4.2 and up
          4    4.4 and up

In [196]: # displaying the last four rows of the data set
          google_store.tail()

Out[196]:                                                  App             Category  \
          10836                               Sya9a Maroc - FR               FAMILY
          10837               Fr. Mike Schmitz Audio Teachings               FAMILY
          10838                         Parkinson Exercices FR              MEDICAL
          10839                  The SCP Foundation DB fr nn5n  BOOKS_AND_REFERENCE
          10840  iHoroscope - 2018 Daily Horoscope & Astrology            LIFESTYLE

                 Rating Reviews                Size     Installs  Type Price  \
          10836     4.5      38                 53M       5,000+  Free     0
          10837     5.0       4                3.6M         100+  Free     0
          10838     NaN       3                9.5M       1,000+  Free     0
          10839     4.5     114  Varies with device       1,000+  Free     0
          10840     4.5  398307                 19M  10,000,000+  Free     0
```

```
            Content Rating           Genres      Last Updated         Current Ver  \
      10836       Everyone        Education     July 25, 2017                1.48
      10837       Everyone        Education      July 6, 2018                 1.0
      10838       Everyone          Medical  January 20, 2017                 1.0
      10839     Mature 17+  Books & Reference  January 19, 2015  Varies with device
      10840       Everyone        Lifestyle     July 25, 2018  Varies with device

                    Android Ver
      10836          4.1 and up
      10837          4.1 and up
      10838          2.2 and up
      10839  Varies with device
      10840  Varies with device
```

0.0.8 Assessing Data

After gathering data required for this project from kaggle website, the next step will be assessing data. I will inspect the dataset for two things: data quality issues (i.e. content issues) and lack of tidiness (i.e structural issues) before moving into the cleaning phase.

```
In [197]: # checking the data type for each column or variable in the data set
          google_store.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
App                 10841 non-null object
Category            10841 non-null object
Rating              9367 non-null float64
Reviews             10841 non-null object
Size                10841 non-null object
Installs            10841 non-null object
Type                10840 non-null object
Price               10841 non-null object
Content Rating      10840 non-null object
Genres              10841 non-null object
Last Updated        10841 non-null object
Current Ver         10833 non-null object
Android Ver         10838 non-null object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

Observation I can observe that the price for the app is object while it should be float. Also, the Installs column is object while it should be int. In addtion, the Reviews column is object while it should be int.

```
In [198]: # displaying the first four rows of the data set
          google_store.head()
```

```
Out[198]:                                                  App        Category  Rating  \
          0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
          1                                Coloring book moana  ART_AND_DESIGN     3.9
          2  U Launcher Lite  FREE Live Cool Themes, Hide ...  ART_AND_DESIGN     4.7
          3                              Sketch - Draw & Paint  ART_AND_DESIGN     4.5
          4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

            Reviews  Size     Installs  Type Price Content Rating  \
          0     159   19M      10,000+  Free     0       Everyone
          1     967   14M     500,000+  Free     0       Everyone
          2   87510  8.7M   5,000,000+  Free     0       Everyone
          3  215644   25M  50,000,000+  Free     0           Teen
          4     967  2.8M     100,000+  Free     0       Everyone

                                Genres      Last Updated         Current Ver  \
          0                 Art & Design   January 7, 2018               1.0.0
          1  Art & Design;Pretend Play  January 15, 2018               2.0.0
          2                 Art & Design    August 1, 2018               1.2.4
          3                 Art & Design      June 8, 2018  Varies with device
          4    Art & Design;Creativity     June 20, 2018                 1.1

              Android Ver
          0  4.0.3 and up
          1  4.0.3 and up
          2  4.0.3 and up
          3    4.2 and up
          4    4.4 and up
```

```
In [199]: # displaying the last four rows of the data set
          google_store.tail()
```

```
Out[199]:                                                  App             Category  \
          10836                               Sya9a Maroc - FR               FAMILY
          10837               Fr. Mike Schmitz Audio Teachings               FAMILY
          10838                         Parkinson Exercices FR              MEDICAL
          10839                  The SCP Foundation DB fr nn5n  BOOKS_AND_REFERENCE
          10840  iHoroscope - 2018 Daily Horoscope & Astrology            LIFESTYLE

                 Rating Reviews                Size     Installs  Type Price  \
          10836     4.5      38                 53M       5,000+  Free     0
          10837     5.0       4                3.6M         100+  Free     0
          10838     NaN       3                9.5M       1,000+  Free     0
          10839     4.5     114  Varies with device       1,000+  Free     0
          10840     4.5  398307                 19M  10,000,000+  Free     0
```

```
      Content Rating             Genres      Last Updated         Current Ver  \
10836       Everyone          Education     July 25, 2017                1.48
10837       Everyone          Education      July 6, 2018                 1.0
10838       Everyone            Medical  January 20, 2017                 1.0
10839     Mature 17+  Books & Reference  January 19, 2015  Varies with device
10840       Everyone          Lifestyle     July 25, 2018  Varies with device

              Android Ver
10836          4.1 and up
10837          4.1 and up
10838          2.2 and up
10839  Varies with device
10840  Varies with device
```

Observation I can observe that the category column values are capitalized and some contains underscores. These two problems need to be fixed to have a more clear data to visualize. Also, the two columns of Last Updated and Current Ver need to be dropped as I will not need them in my analysis. In addition, the Genres column sometimes contains several values in one row, so I need to split these values to more than one row to count them in the final visualization. The Installs columns has a plus(+) sign besides each number and a comma; they need to be removed , so I can apply mathmatical operations on it. Also, in the price column, I need to remove the dollar sign , so I can apply mathamtical operations on it.

```
In [200]: # checking the google_store data set
          google_store.describe()

Out[200]:             Rating
          count  9367.000000
          mean      4.193338
          std       0.537431
          min       1.000000
          25%       4.000000
          50%       4.300000
          75%       4.500000
          max      19.000000
```

Observation It appears that one of the rows has a rating value of 19 which is an extreme to other values in this column. I will check it further using a query.

```
In [201]: # using query to find the row with the rating value of 19
          google_store.query('Rating == 19.0')

Out[201]:                                            App Category  Rating Reviews  \
          10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0    3.0M

                   Size Installs Type     Price Content Rating             Genres  \
          10472  1,000+     Free    0  Everyone            NaN  February 11, 2018
```

```
      Last Updated Current Ver Android Ver
10472       1.0.19  4.0 and up         NaN
```

Observation It appears that this row was wrongly extracted as the data is misplaced under each column, so it needs to be dropped as it can't be fixed.

```
In [202]: # checking duplicated rows
          google_store[google_store.App.duplicated()]
```

```
Out[202]:                                                  App            Category  \
          229                      Quick PDF Scanner + OCR FREE            BUSINESS
          236                                               Box            BUSINESS
          239                                 Google My Business            BUSINESS
          256                                ZOOM Cloud Meetings            BUSINESS
          261                          join.me - Simple Meetings            BUSINESS
          265                                               Box            BUSINESS
          266                                          Zenefits            BUSINESS
          267                                        Google Ads            BUSINESS
          268                                 Google My Business            BUSINESS
          269                                             Slack            BUSINESS
          270                                FreshBooks Classic            BUSINESS
          271                                      Insightly CRM            BUSINESS
          272        QuickBooks Accounting: Invoicing & Expenses            BUSINESS
          273                      HipChat - Chat Built for Teams            BUSINESS
          274                           Xero Accounting Software            BUSINESS
          275           MailChimp - Email, Marketing Automation            BUSINESS
          276               Crew - Free Messaging and Scheduling            BUSINESS
          277                     Asana: organize team projects            BUSINESS
          278                                  Google Analytics            BUSINESS
          279                                    AdWords Express            BUSINESS
          280                       Accounting App - Zoho Books            BUSINESS
          281                     Invoice & Time Tracking - Zoho            BUSINESS
          282                          join.me - Simple Meetings            BUSINESS
          283      Invoice 2go  Professional Invoices and Estimates           BUSINESS
          284       SignEasy | Sign and Fill PDF and other Documents            BUSINESS
          285                      Quick PDF Scanner + OCR FREE            BUSINESS
          286                          Genius Scan - PDF Scanner            BUSINESS
          287                      Tiny Scanner - PDF Scanner App            BUSINESS
          288                      Fast Scanner : Free PDF Scan            BUSINESS
          289                    Mobile Doc Scanner (MDScan) Lite            BUSINESS
          ...                                               ...                 ...
          9854      Nike Training Club - Workouts & Fitness Plans  HEALTH_AND_FITNESS
          9856                      Hangouts Dialer - Call Phones       COMMUNICATION
          9859                          Offline Maps & Navigation     TRAVEL_AND_LOCAL
          9936                  Strawberry Shortcake Ice Cream Island              FAMILY
          9975                        Home Workout - No Equipment  HEALTH_AND_FITNESS
          9982   Home Security Camera WardenCam - reuse old phones      HOUSE_AND_HOME
          10018                                      Food Network              FAMILY
```

```
10026                             Web Browser for Android      COMMUNICATION
10049         Airway Ex - Intubate. Anesthetize. Train.              MEDICAL
10118                       FilterGrid - Cam&Photo Editor        PHOTOGRAPHY
10170        Messages, Text and Video Chat for Messenger              SOCIAL
10171                                 All Social Networks              SOCIAL
10178                       Premier League - Official App              SPORTS
10186                                    Farm Heroes Saga              FAMILY
10188                                 ESPN Fantasy Sports              SPORTS
10190                                      Fallout Shelter              FAMILY
10200                              Facebook Pages Manager            BUSINESS
10203                                Facebook Ads Manager            BUSINESS
10213  Who Viewed My Facebook Profile - Stalkers Visi...              SOCIAL
10269                                       The 5th Stand              SPORTS
10327                                    Garena Free Fire                GAME
10473                              osmino Wi-Fi: free WiFi               TOOLS
10502                          Fun Kid Racing - Motocross              FAMILY
10646  Podcast App: Free & Offline Podcasts by Player FM  NEWS_AND_MAGAZINES
10647                                  Motorola FM Radio       VIDEO_PLAYERS
10715                                 FarmersOnly Dating              DATING
10720                  Firefox Focus: The privacy browser      COMMUNICATION
10730                                         FP Notebook             MEDICAL
10753                     Slickdeals: Coupons & Shopping            SHOPPING
10768                                                AAFP             MEDICAL

       Rating  Reviews                Size      Installs  Type Price  \
229       4.2    80805  Varies with device    5,000,000+  Free     0
236       4.2   159872  Varies with device   10,000,000+  Free     0
239       4.4    70991  Varies with device    5,000,000+  Free     0
256       4.4    31614                 37M   10,000,000+  Free     0
261       4.0     6989  Varies with device    1,000,000+  Free     0
265       4.2   159872  Varies with device   10,000,000+  Free     0
266       4.2      296                 14M       50,000+  Free     0
267       4.3    29313                 20M    5,000,000+  Free     0
268       4.4    70991  Varies with device    5,000,000+  Free     0
269       4.4    51507  Varies with device    5,000,000+  Free     0
270       4.1     1802                 26M      100,000+  Free     0
271       3.8     1383                 51M      100,000+  Free     0
272       4.3    23175                 41M    1,000,000+  Free     0
273       3.8     5868                 20M      500,000+  Free     0
274       3.5     2111  Varies with device      100,000+  Free     0
275       4.1     5448                 12M      500,000+  Free     0
276       4.6     4159                 48M      500,000+  Free     0
277       4.3    20815                 10M    1,000,000+  Free     0
278       4.5    78662                 22M    1,000,000+  Free     0
279       4.1     7149                 11M    1,000,000+  Free     0
280       4.5     3079                8.5M      100,000+  Free     0
281       4.6     5800                8.6M      100,000+  Free     0
282       4.0     6989  Varies with device    1,000,000+  Free     0
```

```
283        4.2    16422                  28M    1,000,000+  Free     0
284        4.3     8978  Varies with device    1,000,000+  Free     0
285        4.2    80804  Varies with device    5,000,000+  Free     0
286        4.4    42492  Varies with device    1,000,000+  Free     0
287        4.7   286897                  39M   10,000,000+  Free     0
288        4.5   103755                  14M   10,000,000+  Free     0
289        4.2    46505                  19M    1,000,000+  Free     0
...        ...      ...                  ...           ...   ...   ...
9854       4.6   251618                  93M   10,000,000+  Free     0
9856       4.0   122512                  79k   10,000,000+  Free     0
9859       4.7   193364                  33M    5,000,000+  Free     0
9936       4.2    32200                  15M    5,000,000+  Free     0
9975       4.8   432160                  15M   10,000,000+  Free     0
9982       4.3    43847  Varies with device    1,000,000+  Free     0
10018      4.1     7823  Varies with device      500,000+  Free     0
10026      4.1    55110                 4.3M    1,000,000+  Free     0
10049      4.3      123                  86M       10,000+  Free     0
10118      4.6   126338                 9.6M    1,000,000+  Free     0
10170      4.4    49580                 4.0M   10,000,000+  Free     0
10171      4.2    22650                 1.5M    1,000,000+  Free     0
10178      4.3    63782                  24M    5,000,000+  Free     0
10186      4.4  7615646                  71M  100,000,000+  Free     0
10188      4.0   176487                  10M    5,000,000+  Free     0
10190      4.6  2721923                  25M   10,000,000+  Free     0
10200      4.0  1279800  Varies with device   50,000,000+  Free     0
10203      4.1    19051  Varies with device    1,000,000+  Free     0
10213      4.6   273244                 9.9M    5,000,000+  Free     0
10269      4.8     1697                  17M      100,000+  Free     0
10327      4.5  5534114                  53M  100,000,000+  Free     0
10473      4.2   134203                 4.1M   10,000,000+  Free     0
10502      4.1    59768  Varies with device   10,000,000+  Free     0
10646      4.6    66407                  19M    1,000,000+  Free     0
10647      3.9    54815  Varies with device  100,000,000+  Free     0
10715      3.0     1145                 1.4M      100,000+  Free     0
10720      4.4    36981                 4.0M    1,000,000+  Free     0
10730      4.5      410                  60M       50,000+  Free     0
10753      4.5    33599                  12M    1,000,000+  Free     0
10768      3.8       63                  24M       10,000+  Free     0

      Content Rating                     Genres       Last Updated  \
229         Everyone                   Business  February 26, 2018
236         Everyone                   Business      July 31, 2018
239         Everyone                   Business      July 24, 2018
256         Everyone                   Business      July 20, 2018
261         Everyone                   Business      July 16, 2018
265         Everyone                   Business      July 31, 2018
266         Everyone                   Business      June 15, 2018
267         Everyone                   Business      July 30, 2018
```

```
268         Everyone                   Business      July 24, 2018
269         Everyone                   Business     August 2, 2018
270         Everyone                   Business     April 18, 2018
271         Everyone                   Business      July 12, 2018
272         Everyone                   Business      July 13, 2018
273         Everyone                   Business       July 3, 2018
274         Everyone                   Business      July 30, 2018
275         Everyone                   Business      July 25, 2018
276         Everyone                   Business      July 20, 2018
277         Everyone                   Business      July 26, 2018
278         Everyone                   Business  February 13, 2018
279         Everyone                   Business      July 31, 2018
280         Everyone                   Business     August 2, 2018
281         Everyone                   Business     August 2, 2018
282         Everyone                   Business      July 16, 2018
283         Everyone                   Business     August 1, 2018
284         Everyone                   Business      July 25, 2018
285         Everyone                   Business  February 26, 2018
286         Everyone                   Business      July 11, 2018
287         Everyone                   Business       May 30, 2017
288         Everyone                   Business      July 11, 2018
289         Everyone                   Business     August 2, 2018
...              ...                        ...                ...
9854        Everyone           Health & Fitness      July 18, 2018
9856        Everyone              Communication  September 2, 2015
9859        Everyone             Travel & Local       June 7, 2018
9936        Everyone        Casual;Pretend Play   November 6, 2017
9975        Everyone           Health & Fitness      June 28, 2018
9982        Everyone               House & Home       July 6, 2018
10018           Teen              Entertainment      July 27, 2018
10026       Everyone              Communication      June 20, 2018
10049       Everyone                    Medical       June 1, 2018
10118       Everyone                Photography     March 15, 2017
10170       Everyone                     Social       June 4, 2018
10171       Everyone                     Social       May 21, 2018
10178       Everyone                     Sports     August 7, 2018
10186       Everyone                     Casual     August 7, 2018
10188       Everyone                     Sports  November 21, 2017
10190           Teen                 Simulation      June 11, 2018
10200       Everyone                   Business     August 6, 2018
10203       Everyone                   Business     August 1, 2018
10213       Everyone                     Social      June 24, 2018
10269       Everyone                     Sports      July 31, 2018
10327           Teen                     Action     August 3, 2018
10473       Everyone                      Tools     August 7, 2018
10502       Everyone  Racing;Action & Adventure     August 7, 2018
10646           Teen           News & Magazines      July 25, 2018
10647       Everyone    Video Players & Editors        May 2, 2018
```

```
10715     Mature 17+                  Dating  February 25, 2016
10720       Everyone           Communication       July 6, 2018
10730       Everyone                 Medical     March 24, 2018
10753       Everyone                Shopping      July 30, 2018
10768       Everyone                 Medical      June 22, 2018

             Current Ver         Android Ver
229   Varies with device        4.0.3 and up
236   Varies with device  Varies with device
239     2.19.0.204537701          4.4 and up
256       4.1.28165.0716          4.0 and up
261            4.3.0.508          4.4 and up
265   Varies with device  Varies with device
266                3.2.1          4.1 and up
267               1.12.0        4.0.3 and up
268     2.19.0.204537701          4.4 and up
269   Varies with device  Varies with device
270               1.7.14          4.2 and up
271               3.24.1          5.0 and up
272                 18.7          4.1 and up
273             3.19.005          4.1 and up
274   Varies with device  Varies with device
275                4.9.1          5.0 and up
276                6.1.2        4.0.3 and up
277                6.4.4          5.0 and up
278                3.7.5          4.4 and up
279              2.6.158          4.0 and up
280               5.20.7          4.1 and up
281               5.20.7          4.1 and up
282            4.3.0.508          4.4 and up
283              10.46.2          4.1 and up
284   Varies with device  Varies with device
285   Varies with device        4.0.3 and up
286   Varies with device  Varies with device
287                1.2.6          3.0 and up
288                3.9.2          4.1 and up
289               3.4.49          4.1 and up
...                  ...                 ...
9854              5.14.0          5.0 and up
9856       0.1.100944346        4.0.3 and up
9859              17.4.1        4.0.3 and up
9936                 1.2          4.1 and up
9975  Varies with device  Varies with device
9982  Varies with device  Varies with device
10018 Varies with device  Varies with device
10026                2.2          4.0 and up
10049             0.6.88          5.0 and up
10118              2.0.5          4.0 and up
```

```
10170                 1.24          4.1 and up
10171               2.4.12          4.0 and up
10178                1.1.5          4.1 and up
10186                5.2.6          2.3 and up
10188                5.3.0          4.4 and up
10190              1.13.12          4.1 and up
10200   Varies with device  Varies with device
10203         99.0.0.35.75          4.1 and up
10213                4.1.1        4.0.3 and up
10269              1.2.677          4.4 and up
10327               1.21.0        4.0.3 and up
10473              6.06.14          4.4 and up
10502                 3.53          4.2 and up
10646             4.1.0.72          4.0 and up
10647   Varies with device  Varies with device
10715                  2.2          4.0 and up
10720                  5.2          5.0 and up
10730            2.1.0.372          4.4 and up
10753                  3.9          4.4 and up
10768                2.3.1          5.0 and up

[1181 rows x 13 columns]
```

```
In [203]: # counting the number of duplicated rows
          google_store[google_store.App.duplicated()].count()
```

```
Out[203]: App               1181
          Category          1181
          Rating            1170
          Reviews           1181
          Size              1181
          Installs          1181
          Type              1181
          Price             1181
          Content Rating    1181
          Genres            1181
          Last Updated      1181
          Current Ver       1181
          Android Ver       1181
          dtype: int64
```

Observation There are about 1181 duplicated rows in the google play store data set; they need to be dropped immediately.

0.1 Conclusion

The Google Store data set has seven data quality issues and one tidiness issue as the following :

Google Store Data Set :

Quality Issues:

1. The Category column values are capitalized and some contains underscores; they need to be fixed.
2. The Last Updated and Current Ver columns need to be dropped from the data set.
3. One row of data was extracted wrongly and needs to be dropped from the data set.
4. There are 1181 duplicated rows in the Google Play Store data set; they need to be dropped from the data set.
5. The Installs columns has a plus(+) sign besides each number and a comma; they need to be removed and then converte the column type to an int.
6. The price column has a dollar sign; it needs to be removed and then converte the column type to a float.
7. The Reviews column is an object while it should be an int.

Tidiness Issues:

1. The Genres column contains several values in one in one row; they need to splitted to more than one row to be counted.

0.1.1 Cleaning Data

Cleaning data is the third step in data wrangling. I will fix the quality and tidiness issues that were identified in the assessing step using Python codes.

```
In [204]: # creating a copy of the google
          google_store_clean = google_store.copy()
```

Define
The Last Updated and Current Ver columns need to be dropped from the data set. I will use the drop function from Pandas library to drop them.
Code

```
In [205]: # dropping un needed columns from Google Store Data Set
          google_store_clean.drop(["Last Updated","Current Ver"],axis = 1, inplace = True)
```

Test

```
In [206]: # checking if the two columns are dropped from the table
          google_store_clean.head()

Out[206]:                                                 App        Category  Rating  \
          0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
          1                                Coloring book moana  ART_AND_DESIGN     3.9
          2  U Launcher Lite  FREE Live Cool Themes, Hide ...  ART_AND_DESIGN     4.7
          3                              Sketch - Draw & Paint  ART_AND_DESIGN     4.5
          4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3
```

```
  Reviews  Size     Installs  Type Price Content Rating  \
0     159   19M      10,000+  Free     0       Everyone
1     967   14M     500,000+  Free     0       Everyone
2   87510  8.7M   5,000,000+  Free     0       Everyone
3  215644   25M  50,000,000+  Free     0           Teen
4     967  2.8M     100,000+  Free     0       Everyone

                     Genres   Android Ver
0              Art & Design  4.0.3 and up
1  Art & Design;Pretend Play  4.0.3 and up
2              Art & Design  4.0.3 and up
3              Art & Design    4.2 and up
4    Art & Design;Creativity    4.4 and up
```

Define One row of data was extracted wrongly and needs to be dropped from the data set. I will use the drop function from Pandas Library to drop it.
Code

```
In [207]: # dropping the wrongly extracted row
          google_store_clean.drop(google_store_clean.index[[10472]],inplace = True)
```

Test

```
In [208]: # checking that the row is dropped
          google_store_clean.query('Rating == 19.0')

Out[208]: Empty DataFrame
          Columns: [App, Category, Rating, Reviews, Size, Installs, Type, Price, Content Ratin
          Index: []
```

Define
There are 1181 duplicated rows in the Google Play Store data set; they need to be dropped from the data set. I will use the drop_duplicates function from Pandas Library to drop them.
Code

```
In [209]: # dropping duplicates from the data set
          google_store_clean.drop_duplicates(subset = "App", inplace = True)
```

Test

```
In [210]: # counting the number of duplicated rows
          google_store_clean[google_store_clean.App.duplicated()].count()

Out[210]: App                 0
          Category            0
          Rating              0
          Reviews             0
          Size                0
          Installs            0
```

```
Type              0
Price             0
Content Rating    0
Genres            0
Android Ver       0
dtype: int64
```

Define
The Category column values are capitalized and some contains underscores; they need to be fixed. I will use replace function to get rid of undersocres and use lower and title functions to make the category title in a upper case.
Code

```
In [211]: # fixing the category column in Google Store Data Set
          # replacing the underscore with space
          google_store_clean.Category = google_store_clean.Category.str.replace('_', ' ')
          # lower case for the category column values
          google_store_clean.Category = google_store_clean.Category.str.lower()
          # title case for the category column values
          google_store_clean.Category = google_store_clean.Category.str.title()
```

Test

```
In [212]: # checking that changes are made in the category column
          google_store_clean.head()
```

```
Out[212]:                                                 App        Category  Rating  \
          0     Photo Editor & Candy Camera & Grid & ScrapBook  Art And Design     4.1
          1                                Coloring book moana  Art And Design     3.9
          2  U Launcher Lite  FREE Live Cool Themes, Hide ...  Art And Design     4.7
          3                              Sketch - Draw & Paint  Art And Design     4.5
          4              Pixel Draw - Number Art Coloring Book  Art And Design     4.3

            Reviews  Size     Installs  Type Price Content Rating  \
          0     159   19M      10,000+  Free     0       Everyone
          1     967   14M     500,000+  Free     0       Everyone
          2   87510  8.7M   5,000,000+  Free     0       Everyone
          3  215644   25M  50,000,000+  Free     0           Teen
          4     967  2.8M     100,000+  Free     0       Everyone

                               Genres   Android Ver
          0                Art & Design  4.0.3 and up
          1  Art & Design;Pretend Play  4.0.3 and up
          2                Art & Design  4.0.3 and up
          3                Art & Design    4.2 and up
          4    Art & Design;Creativity    4.4 and up
```

Define

The Genres column contains several values in one in one row; they need to splitted to more than one row to be counted. I will use stack,str.split, unstack and reset_index functions to split this column into multiple row.

Code

```
In [213]: # splitting genres column into multiple rows
          google_store_clean = google_store_clean.set_index(['App','Category','Rating','Reviews
```

Test

```
In [214]: # checking that the column is splited
          google_store_clean.query('App == "Coloring book moana" | App == "Pixel Draw - Number

Out[214]:                                 App        Category  Rating Reviews  \
          2711             Coloring book moana  Art And Design     3.9     967
          2712             Coloring book moana  Art And Design     3.9     967
          7308  Pixel Draw - Number Art Coloring Book  Art And Design     4.3     967
          7309  Pixel Draw - Number Art Coloring Book  Art And Design     4.3     967

                Size  Installs  Type Price Content Rating   Android Ver        Genres
          2711   14M  500,000+  Free     0       Everyone  4.0.3 and up  Art & Design
          2712   14M  500,000+  Free     0       Everyone  4.0.3 and up  Pretend Play
          7308  2.8M  100,000+  Free     0       Everyone    4.4 and up  Art & Design
          7309  2.8M  100,000+  Free     0       Everyone    4.4 and up    Creativity
```

Define

The Installs columns has a plus(+) sign besides each number and a comma; they need to be removed and then converte the column type to an int. I will use the replace function to remove the plus sign and astype to convert its value to an int.

Code

```
In [215]: # fixing the Installs column issues
          # removing plus sign and comma and then changing the column type to an int
          google_store_clean.Installs = google_store_clean.Installs.str.replace('+','').str.rep
```

Test

```
In [216]: # checking that the Installs column issues are solved
          google_store_clean.head(20)

Out[216]:                                      App           Category  \
          0         "i DT" Fútbol. Todos Somos Técnicos.             Sports
          1               +Download 4 Instagram Twitter             Social
          2                   - Free Comics - Comic Apps             Comics
          3                                          .R              Tools
          4                                      /u/app      Communication
          5                                      058.ba  News And Magazines
          6                                1. FC Köln App             Sports
          7                        10 Best Foods for You  Health And Fitness
```

```
8                        10 Minutes a Day Times Tables              Family
9      10 WPM Amateur ham radio CW Morse code trainer        Communication
10                            10,000 Quotes DB (Premium)  Books And Reference
11                                  100 Doors of Revenge              Family
12                                       100+ C Programs              Family
13     100000+ Messages - DP, Status, Jokes & GIF 2018           Lifestyle
14                            101 C Programming Problems              Family
15  104 Looking for a job - looking for a job, loo...            Business
16                                                 11st            Shopping
17  12 Step Meditations & Sober Prayers AA NA AL-ANON           Lifestyle
18                                        14thStreetVet             Medical
19                              17th Edition Cable Sizer  Books And Reference

    Rating Reviews  Size  Installs  Type  Price Content Rating   Android Ver  \
0      NaN      27  3.6M       500  Free      0       Everyone    4.1 and up
1      4.5   40467   22M   1000000  Free      0       Everyone    4.1 and up
2      3.5     115  9.1M     10000  Free      0     Mature 17+    5.0 and up
3      4.5     259  203k     10000  Free      0       Everyone    1.5 and up
4      4.7     573   53M     10000  Free      0     Mature 17+    4.1 and up
5      4.4      27   14M       100  Free      0       Everyone    4.2 and up
6      4.6    2019   41M    100000  Free      0       Everyone    4.4 and up
7      4.0    2490  3.8M    500000  Free      0   Everyone 10+  2.3.3 and up
8      4.1     681   48M    100000  Free      0       Everyone    2.2 and up
9      3.5      10  3.8M       100  Paid  $1.49       Everyone    2.1 and up
10     4.1      70  3.5M       500  Paid  $0.99       Everyone    2.1 and up
11     4.1  105766   48M  10000000  Free      0           Teen    4.4 and up
12     4.2      20  1.6M      5000  Free      0       Everyone    4.0 and up
13     3.7     121  3.8M     10000  Free      0     Mature 17+  4.0.3 and up
14     4.6     498  5.0M     50000  Free      0       Everyone  4.0.3 and up
15     4.4   74359   25M   1000000  Free      0       Everyone    4.0 and up
16     3.8   48732   20M  10000000  Free      0       Everyone    4.0 and up
17     4.7     759   15M     50000  Free      0   Everyone 10+    4.1 and up
18     NaN       0   29M         5  Free      0       Everyone  4.0.3 and up
19     4.4      47  1.4M      1000  Paid  $3.08       Everyone    2.2 and up

              Genres
0             Sports
1             Social
2             Comics
3              Tools
4      Communication
5   News & Magazines
6             Sports
7   Health & Fitness
8          Education
9      Communication
10  Books & Reference
11            Puzzle
```

```
        12          Education
        13          Lifestyle
        14          Education
        15           Business
        16           Shopping
        17          Lifestyle
        18            Medical
        19  Books & Reference
```

```
In [217]: # checking the type of Installs column
          google_store_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10052 entries, 0 to 10051
Data columns (total 11 columns):
App                  10052 non-null object
Category             10052 non-null object
Rating               8575 non-null float64
Reviews              10052 non-null object
Size                 10052 non-null object
Installs             10052 non-null int32
Type                 10051 non-null object
Price                10052 non-null object
Content Rating       10052 non-null object
Android Ver          10050 non-null object
Genres               10052 non-null object
dtypes: float64(1), int32(1), object(9)
memory usage: 824.7+ KB
```

Define
The price column has a dollar sign; it needs to be removed and then converte the column type to a float. I will use the replace function to remove the plus sign and astype to convert its value to a float.
Code

```
In [218]: # fixing the Price column issues
          # removing $ sign and changing the column values to float
          google_store_clean.Price = google_store_clean.Price.str.replace('$','').astype(float
```

Test

```
In [219]: # checking that the Price column issues are solved
          google_store_clean.head(20)
```

```
Out[219]:                                               App           Category  \
          0                 "i DT" Fútbol. Todos Somos Técnicos.             Sports
          1                        +Download 4 Instagram Twitter             Social
          2                           - Free Comics - Comic Apps             Comics
```

```
3                                                 .R               Tools
4                                             /u/app       Communication
5                                             058.ba   News And Magazines
6                                     1. FC Köln App              Sports
7                                10 Best Foods for You   Health And Fitness
8                        10 Minutes a Day Times Tables              Family
9      10 WPM Amateur ham radio CW Morse code trainer       Communication
10                        10,000 Quotes DB (Premium)  Books And Reference
11                              100 Doors of Revenge              Family
12                                   100+ C Programs              Family
13     100000+ Messages - DP, Status, Jokes & GIF 2018           Lifestyle
14                        101 C Programming Problems              Family
15  104 Looking for a job - looking for a job, loo...            Business
16                                              11st            Shopping
17  12 Step Meditations & Sober Prayers AA NA AL-ANON           Lifestyle
18                                      14thStreetVet             Medical
19                           17th Edition Cable Sizer  Books And Reference

    Rating Reviews  Size  Installs  Type  Price Content Rating   Android Ver  \
0      NaN      27  3.6M       500  Free   0.00       Everyone    4.1 and up
1      4.5   40467   22M   1000000  Free   0.00       Everyone    4.1 and up
2      3.5     115  9.1M     10000  Free   0.00     Mature 17+    5.0 and up
3      4.5     259  203k     10000  Free   0.00       Everyone    1.5 and up
4      4.7     573   53M     10000  Free   0.00     Mature 17+    4.1 and up
5      4.4      27   14M       100  Free   0.00       Everyone    4.2 and up
6      4.6    2019   41M    100000  Free   0.00       Everyone    4.4 and up
7      4.0    2490  3.8M    500000  Free   0.00   Everyone 10+  2.3.3 and up
8      4.1     681   48M    100000  Free   0.00       Everyone    2.2 and up
9      3.5      10  3.8M       100  Paid   1.49       Everyone    2.1 and up
10     4.1      70  3.5M       500  Paid   0.99       Everyone    2.1 and up
11     4.1  105766   48M  10000000  Free   0.00           Teen    4.4 and up
12     4.2      20  1.6M      5000  Free   0.00       Everyone    4.0 and up
13     3.7     121  3.8M     10000  Free   0.00     Mature 17+  4.0.3 and up
14     4.6     498  5.0M     50000  Free   0.00       Everyone  4.0.3 and up
15     4.4   74359   25M   1000000  Free   0.00       Everyone    4.0 and up
16     3.8   48732   20M  10000000  Free   0.00       Everyone    4.0 and up
17     4.7     759   15M     50000  Free   0.00   Everyone 10+    4.1 and up
18     NaN       0   29M         5  Free   0.00       Everyone  4.0.3 and up
19     4.4      47  1.4M      1000  Paid   3.08       Everyone    2.2 and up

              Genres
0             Sports
1             Social
2             Comics
3              Tools
4      Communication
5    News & Magazines
6             Sports
```

```
7     Health & Fitness
8            Education
9        Communication
10   Books & Reference
11              Puzzle
12           Education
13           Lifestyle
14           Education
15            Business
16            Shopping
17           Lifestyle
18             Medical
19   Books & Reference
```

```
In [220]: # checking the price column type
          google_store_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10052 entries, 0 to 10051
Data columns (total 11 columns):
App               10052 non-null object
Category          10052 non-null object
Rating            8575 non-null float64
Reviews           10052 non-null object
Size              10052 non-null object
Installs          10052 non-null int32
Type              10051 non-null object
Price             10052 non-null float64
Content Rating    10052 non-null object
Android Ver       10050 non-null object
Genres            10052 non-null object
dtypes: float64(2), int32(1), object(8)
memory usage: 824.7+ KB
```

Define
The Reviews column is an object while it should be an int. I will use the astype function to convert it to an int.
Code

```
In [221]: # fixing the Reviews column issues
          # converting Reviews Column to an int
          google_store_clean.Reviews = google_store_clean.Reviews.astype(int)
```

Test

```
In [222]: # checking that Reviews column is converted to the correct data type
          google_store_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10052 entries, 0 to 10051
Data columns (total 11 columns):
App                 10052 non-null object
Category            10052 non-null object
Rating              8575 non-null float64
Reviews             10052 non-null int32
Size                10052 non-null object
Installs            10052 non-null int32
Type                10051 non-null object
Price               10052 non-null float64
Content Rating      10052 non-null object
Android Ver         10050 non-null object
Genres              10052 non-null object
dtypes: float64(2), int32(2), object(7)
memory usage: 785.4+ KB
```

```
In [223]: # creating a new file for the cleaned data set
          with open('google_store_clean.csv', 'w', encoding = 'UTF-8') as file:

              google_store_clean.to_csv(file, index = False)
```

0.1.2 Data Analysis and Visualizations

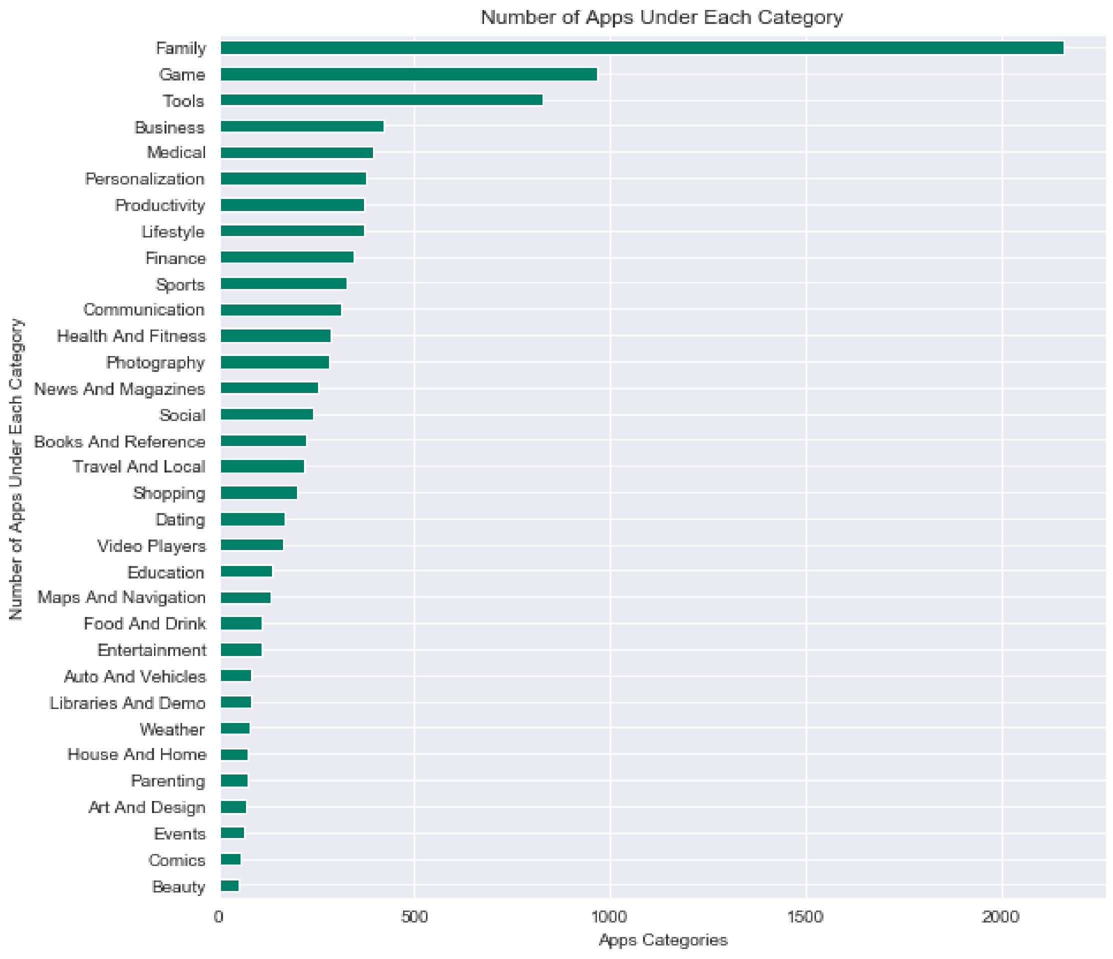
In the last part of this project, I will make sense of all the data that was previously gathered, assessed and cleaned using the graphs and figures by the help of matplotlib and seaborn library.

```
In [6]: # reading clean data set first
        google_store_clean = pd.read_csv('google_store_clean.csv')
```

What are the top 10 categories for Android mobile applications currently ?

```
In [225]: # checking the number of each app category
          app_cats = google_store_clean.groupby('Category')['App'].count().sort_values()
          fig, ax = plt.subplots(figsize=(9,9))
          app_cats.plot(ax=ax,kind="barh",colormap='summer')
          ax.set_xlabel('Apps Categories')
          ax.set_ylabel('Number of Apps Under Each Category')
          ax.set_title('Number of Apps Under Each Category')

Out[225]: Text(0.5, 1.0, 'Number of Apps Under Each Category')
```



The top 10 categories for Android mobile applications currently are :

1. Family.
2. Game.
3. Tools.
4. Business.
5. Medical.
6. Personalization.
7. Productivity.
8. Lifestyle.
9. Finance.
10. Sports.

What is the average rating for each Android App category ?

```
In [226]: # finding the average rating for each Android app category
          google_store_clean.groupby('Category')['Rating'].mean().sort_values()
```

```
Out[226]: Category
          Dating                 3.970149
          Maps And Navigation    4.036441
          Tools                  4.040195
          Video Players          4.044295
          Travel And Local       4.069681
          Lifestyle              4.093046
          Business               4.098479
          Finance                4.115563
          Communication          4.121484
          News And Magazines     4.121569
          Entertainment          4.142857
          House And Home         4.150000
          Photography            4.157414
          Medical                4.166552
          Food And Drink         4.172340
          Libraries And Demo     4.178125
          Productivity           4.183389
          Auto And Vehicles      4.190411
          Comics                 4.192727
          Family                 4.195223
          Sports                 4.216154
          Shopping               4.230000
          Health And Fitness     4.243033
          Weather                4.243056
          Social                 4.247291
          Game                   4.249024
          Parenting              4.273333
          Beauty                 4.278571
          Personalization        4.332215
          Books And Reference    4.344970
          Art And Design         4.356716
          Education              4.378986
          Events                 4.435556
          Name: Rating, dtype: float64
```

It can be concluded that the average rating for each Android app category is somehow similar to the other apps under different categories. However, the Dating Apps got the lowest average rating with a score of 3.97. This could be an opportunity for developers to develop a better featured dating apps.

What are the top ten apps under each category ?

```
In [227]: # list of apps categories
          google_store_clean.Category.unique()
```

```
Out[227]: array(['Sports', 'Social', 'Comics', 'Tools', 'Communication',
                 'News And Magazines', 'Health And Fitness', 'Family',
                 'Books And Reference', 'Lifestyle', 'Business', 'Shopping',
                 'Medical', 'Game', 'Finance', 'Personalization', 'Photography',
                 'Travel And Local', 'Dating', 'Productivity', 'Art And Design',
                 'Food And Drink', 'Video Players', 'House And Home',
                 'Maps And Navigation', 'Entertainment', 'Events', 'Education',
                 'Auto And Vehicles', 'Weather', 'Beauty', 'Libraries And Demo',
                 'Parenting'], dtype=object)
```

In [25]:
```
# excluding apps with no rating and app with less than 1000 review
#(This approach helps me to have more reliable results)
group = google_store_clean.query(' Rating != "NaN" and Reviews >= 1000')
# sorting apps with the highest ratings
group = group.sort_values(by = ['Rating','Installs'],ascending = False)
group = group.groupby('Category')
# replace this category with any category you want to view data !
group.get_group('Tools')
```

Out[25]:
```
                                                    App Category  Rating  \
4572  File Manager by Xiaomi: release file storage s...    Tools     4.8
2359                        Calculator with Percent (Free)    Tools     4.8
2360                                              Calcy IV    Tools     4.8
5684                                  K keyboard - Myanmar    Tools     4.8
5729                        Kernel Manager for Franco Kernel    Tools     4.8
4343                                   FK Crvena Zvezda Izzy    Tools     4.8
9640                                        ZArchiver Donate    Tools     4.8
2647                    Clean Master- Space Cleaner & Antivirus    Tools     4.7
8037  Security Master - Antivirus, VPN, AppLock, Boo...    Tools     4.7
1815                               Brightest Flashlight Free ő    Tools     4.7
5714  Kaspersky Mobile Antivirus: AppLock & Web Secu...    Tools     4.7
4641                                              Flashlight    Tools     4.7
4914                    GO SecurityAntiVirus, AppLock, Booster    Tools     4.7
8261                                             Sound Meter    Tools     4.7
1939                          CALCU Stylish Calculator Free    Tools     4.7
5538   Inf VPN - Global Proxy & Unlimited Free WIFI VPN    Tools     4.7
7367                          Poke Genie - Safe IV Calculator    Tools     4.7
8287  Speed Booster - Ram, Battery & Game Speed Booster    Tools     4.7
8768                                     The Zueira's Voice    Tools     4.7
8890                           TorrDroid - Torrent Downloader    Tools     4.7
9892                                           iSwipe Phone X    Tools     4.7
9098                                                   UniFi    Tools     4.7
9420                              WhatsVPN - Unlimited Free VPN    Tools     4.7
1496                                          Battery HD Pro    Tools     4.7
1882                                             BusyBox Pro    Tools     4.7
532                           Advanced Download Manager Pro    Tools     4.7
3364                                          Digital Falak    Tools     4.7
3966                                   EZ-GUI Ground Station    Tools     4.7
4640                                   FlashLight HD LED Pro    Tools     4.7
886                                   Automagic * Automation    Tools     4.7
```

```
...                                                    ...     ...     ...
8032                             Secret Codes For Android    Tools     3.6
6399                             Metal Detector Pro 2015    Tools     3.6
7519                                        Q-See QT View    Tools     3.6
9946                                                 myQ    Tools     3.6
5083                                Google Korean Input    Tools     3.5
390                                     AT&T FamilyMapő    Tools     3.5
6701                                             My love    Tools     3.5
8190                                     Smart TV Remote    Tools     3.5
1742                                      Bluetooth Pair    Tools     3.5
7727                             Remote CT - Smart Remote    Tools     3.5
9642                                    ZERO Lock Screen    Tools     3.5
794                                            App vault    Tools     3.4
5233                                     HTC Sense Input    Tools     3.4
3124                                              DS get    Tools     3.4
127                           A/C Air Conditioner Remote    Tools     3.4
590                       Air conditioner remote control    Tools     3.4
5846                                        LG AV REMOTE    Tools     3.4
7736     Remote for Sony TV & Sony Blu-Ray Players MyAV    Tools     3.4
7839                                    Ruler(cm, inch)    Tools     3.4
4083                             Emo Ads Blocker Browser    Tools     3.4
7735  Remote for Samsung TV & BluRay Players (Read D...    Tools     3.3
8182                               Smart Air Conditioner    Tools     3.3
9896                                  iWnn IME for Nexus    Tools     3.2
3121                                            DS cloud    Tools     3.2
6684                                           My Telcel    Tools     3.1
2650                                               Clear    Tools     3.1
129                           A/C Universal Remote Control    Tools     3.1
164                                     AC Remote For LG    Tools     3.0
1289                                         BT Notifier    Tools     2.5
797                                 AppFinder by AppTap    Tools     2.0

       Reviews                Size   Installs  Type  Price Content Rating  \
4572    337532                 15M   10000000  Free   0.00       Everyone
2359     48211                7.4M    1000000  Free   0.00       Everyone
2360     36557                 14M    1000000  Free   0.00       Everyone
5684      1955                 14M     100000  Free   0.00       Everyone
5729     12700                 10M     100000  Paid   3.49       Everyone
4343      1456                 15M      50000  Free   0.00       Everyone
9640      1721  Varies with device      10000  Paid   2.50       Everyone
2647  42916526  Varies with device  500000000  Free   0.00       Everyone
8037  24900999  Varies with device  500000000  Free   0.00       Everyone
1815   1335799                3.8M   50000000  Free   0.00       Everyone
5714   2598579                 49M   50000000  Free   0.00       Everyone
4641    115409  Varies with device   10000000  Free   0.00       Everyone
4914   1251479  Varies with device   10000000  Free   0.00       Everyone
8261     88993                1.5M   10000000  Free   0.00       Everyone
1939    152692                 11M    5000000  Free   0.00       Everyone
```

```
5538    61445              7.8M    1000000  Free   0.00       Everyone
7367    92958               36M    1000000  Free   0.00       Everyone
8287    85079              2.8M    1000000  Free   0.00       Everyone
8768   136540              3.1M    1000000  Free   0.00       Everyone
8890    59632               10M    1000000  Free   0.00       Everyone
9892    58366              6.3M    1000000  Free   0.00       Everyone
9098    11018               25M     500000  Free   0.00       Everyone
9420    24985              7.8M     500000  Free   0.00       Everyone
1496    17861  Varies with device   100000  Paid   3.99       Everyone
1882     8114  Varies with device   100000  Paid   2.49       Everyone
532      6505              2.0M      50000  Paid   2.99       Everyone
3364     3408               15M      50000  Free   0.00       Everyone
3966     3696               29M      50000  Free   0.00       Everyone
4640     4928  Varies with device    50000  Paid   2.99       Everyone
886      1947  Varies with device    10000  Paid   3.90       Everyone
...       ...                 ...      ...   ...    ...            ...
8032     6060              592k     500000  Free   0.00       Everyone
6399     1166              350k     100000  Free   0.00       Everyone
7519     7357               19M     100000  Free   0.00       Everyone
9946     3642               29M     100000  Free   0.00       Everyone
5083    74819  Varies with device 100000000 Free   0.00       Everyone
390     21592               25M   10000000  Free   0.00       Everyone
6701   163997               16M   10000000  Free   0.00       Everyone
8190    81747              9.1M   10000000  Free   0.00       Everyone
1742     1960              2.7M    1000000  Free   0.00       Everyone
7727     3988               11M    1000000  Free   0.00       Everyone
9642    75336              544k    1000000  Free   0.00       Everyone
794     25094  Varies with device 10000000  Free   0.00       Everyone
5233    17030  Varies with device 10000000  Free   0.00       Everyone
3124    71852               22M    5000000  Free   0.00       Everyone
127      7816              5.1M    1000000  Free   0.00       Everyone
590     29854              5.0M    1000000  Free   0.00       Everyone
5846     2420              1.8M    1000000  Free   0.00       Everyone
7736     3491              7.3M    1000000  Free   0.00       Everyone
7839     2889              2.5M     500000  Free   0.00       Everyone
4083     1555              4.1M     100000  Free   0.00       Everyone
7735     1988              7.3M     500000  Free   0.00       Everyone
8182     4786               13M     500000  Free   0.00       Everyone
9896     2394  Varies with device  5000000  Free   0.00       Everyone
3121     4908               38M     500000  Free   0.00       Everyone
6684    45838               16M   50000000  Free   0.00       Everyone
2650    24151               15M   10000000  Free   0.00       Everyone
129      3263              3.0M    1000000  Free   0.00       Everyone
164      1298              3.4M     100000  Free   0.00       Everyone
1289     2794              3.8M    1000000  Free   0.00       Everyone
797      2221              4.9M    5000000  Free   0.00       Everyone

             Android Ver Genres
```

```
4572          4.4 and up  Tools
2359          4.1 and up  Tools
2360          5.0 and up  Tools
5684          4.1 and up  Tools
5729          5.0 and up  Tools
4343          4.4 and up  Tools
9640  Varies with device  Tools
2647  Varies with device  Tools
8037  Varies with device  Tools
1815          4.2 and up  Tools
5714          4.1 and up  Tools
4641  Varies with device  Tools
4914  Varies with device  Tools
8261          2.3 and up  Tools
1939          4.1 and up  Tools
5538          4.1 and up  Tools
7367          5.0 and up  Tools
8287          2.3 and up  Tools
8768        4.0.3 and up  Tools
8890          4.1 and up  Tools
9892          4.1 and up  Tools
9098          4.1 and up  Tools
9420          4.1 and up  Tools
1496  Varies with device  Tools
1882  Varies with device  Tools
532           4.0 and up  Tools
3364          4.1 and up  Tools
3966          4.3 and up  Tools
4640  Varies with device  Tools
886   Varies with device  Tools
...                  ...    ...
8032          2.1 and up  Tools
6399          4.0 and up  Tools
7519          5.0 and up  Tools
9946        4.0.3 and up  Tools
5083          7.1 and up  Tools
390         4.0.3 and up  Tools
6701          4.2 and up  Tools
8190          4.1 and up  Tools
1742          4.1 and up  Tools
7727          4.4 and up  Tools
9642          4.0 and up  Tools
794   Varies with device  Tools
5233  Varies with device  Tools
3124          4.1 and up  Tools
127           3.0 and up  Tools
590           2.3 and up  Tools
5846        2.3.3 and up  Tools
```

```
7736         4.3 and up  Tools
7839         2.3 and up  Tools
4083         3.0 and up  Tools
7735         4.3 and up  Tools
8182         2.2 and up  Tools
9896  Varies with device  Tools
3121         4.0 and up  Tools
6684         4.1 and up  Tools
2650         4.1 and up  Tools
129          3.0 and up  Tools
164          4.0 and up  Tools
1289         4.3 and up  Tools
797          5.0 and up  Tools

[361 rows x 11 columns]
```

The above table will help apps developers to view the common features for succuessful apps in the market under each app category.

What are the top 10 apps genres under each app category?

```
In [143]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Art And Design"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[143]: Category        Genres
          Art And Design  Art & Design          64
                          Creativity             5
                          Pretend Play           1
                          Action & Adventure     1
          Name: App, dtype: int64
```

```
In [145]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Sports"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[145]: Category  Genres
          Sports    Sports    325
          Name: App, dtype: int64
```

```
In [146]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Social"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[146]: Category  Genres
          Social    Social    239
          Name: App, dtype: int64
```

```
In [147]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Comics"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[147]: Category  Genres
          Comics    Comics        56
                    Creativity     1
          Name: App, dtype: int64
```

```
In [148]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Tools"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[148]: Category  Genres
          Tools     Tools        827
                    Education      1
          Name: App, dtype: int64
```

```
In [149]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Communication"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[149]: Category       Genres
          Communication  Communication    315
          Name: App, dtype: int64
```

```
In [150]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="News And Magazines"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[150]: Category            Genres
          News And Magazines  News & Magazines    254
          Name: App, dtype: int64
```

```
In [151]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Health And Fitness"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[151]: Category            Genres
          Health And Fitness  Health & Fitness    288
          Name: App, dtype: int64
```

```
In [152]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Family"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[152]: Category  Genres
          Family    Education                  542
                    Entertainment              490
                    Casual                     198
                    Simulation                 194
                    Puzzle                     118
                    Educational                102
                    Role Playing                 100
                    Action & Adventure            89
                    Strategy                      82
                    Pretend Play                  62
                    Brain Games                   57
                    Creativity                    22
                    Board                         19
                    Racing                        17
                    Music & Video                 16
                    Arcade                        15
                    Action                         9
                    Adventure                      6
                    Sports                         4
                    Books & Reference              3
                    Music                          3
                    Health & Fitness               2
                    Card                           2
                    Video Players & Editors        2
                    Trivia                         1
                    Communication                  1
                    Lifestyle                      1
                    Art & Design                   1
                    Music & Audio                  1
          Name: App, dtype: int64
```

```
In [153]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Books And Reference"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[153]: Category             Genres
          Books And Reference  Books & Reference    222
          Name: App, dtype: int64
```

```
In [154]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Lifestyle"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[154]: Category   Genres
          Lifestyle  Lifestyle       369
                     Pretend Play      1
          Name: App, dtype: int64
```

```
In [155]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Business"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[155]: Category  Genres
          Business  Business    420
          Name: App, dtype: int64
```

```
In [156]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Shopping"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[156]: Category  Genres
          Shopping  Shopping    202
          Name: App, dtype: int64

In [157]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Medical"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[157]: Category  Genres
          Medical   Medical    395
          Name: App, dtype: int64

In [158]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Game"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[158]: Category  Genres
          Game      Action                 302
                    Arcade                 184
                    Racing                  91
                    Adventure               74
                    Card                    47
                    Board                   41
                    Casino                  39
                    Trivia                  38
                    Casual                  27
                    Puzzle                  24
                    Word                    23
                    Music                   19
                    Strategy                17
                    Role Playing            15
                    Simulation              12
                    Sports                   6
                    Action & Adventure       6
                    Creativity               1
                    Pretend Play             1
                    Brain Games              1
                    Education                1
          Name: App, dtype: int64

In [159]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Finance"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[159]: Category  Genres
          Finance   Finance    345
          Name: App, dtype: int64
```

```
In [160]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Personalization"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[160]: Category         Genres
          Personalization  Personalization    376
          Name: App, dtype: int64

In [161]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Photography"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[161]: Category     Genres
          Photography  Photography    281
          Name: App, dtype: int64

In [162]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Travel And Local"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[162]: Category          Genres
          Travel And Local  Travel & Local        219
                            Action & Adventure      1
          Name: App, dtype: int64

In [163]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Dating"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[163]: Category  Genres
          Dating    Dating    171
          Name: App, dtype: int64

In [164]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Productivity"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[164]: Category      Genres
          Productivity  Productivity    374
          Name: App, dtype: int64

In [165]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Food And Drink"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[165]: Category        Genres
          Food And Drink  Food & Drink    112
          Name: App, dtype: int64
```

```
In [167]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Video Players"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[167]: Category       Genres
          Video Players  Video Players & Editors    163
                         Music & Video                1
          Name: App, dtype: int64

In [168]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="House And Home"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[168]: Category        Genres
          House And Home  House & Home    74
          Name: App, dtype: int64

In [169]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Maps And Navigation"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[169]: Category             Genres
          Maps And Navigation  Maps & Navigation    131
          Name: App, dtype: int64

In [170]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Entertainment"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[170]: Category       Genres
          Entertainment  Entertainment    102
                         Music & Video      7
                         Brain Games        2
                         Creativity         1
          Name: App, dtype: int64

In [171]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Events"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)

Out[171]: Category  Genres
          Events    Events    64
          Name: App, dtype: int64

In [172]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Education"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[172]: Category   Genres
          Education  Education              127
                     Pretend Play             4
                     Creativity               3
                     Brain Games              3
                     Music & Video            1
                     Action & Adventure       1
          Name: App, dtype: int64
```

```
In [173]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Auto And Vehicles"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[173]: Category           Genres
          Auto And Vehicles  Auto & Vehicles    85
          Name: App, dtype: int64
```

```
In [174]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Weather"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[174]: Category  Genres
          Weather   Weather    79
          Name: App, dtype: int64
```

```
In [175]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Beauty"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[175]: Category  Genres
          Beauty    Beauty    53
          Name: App, dtype: int64
```

```
In [176]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Libraries And Demo"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

```
Out[176]: Category            Genres
          Libraries And Demo  Libraries & Demo    84
          Name: App, dtype: int64
```

```
In [177]: # displaying the number of each genre under each App category
          query = google_store_clean.query('Category =="Parenting"')
          query.groupby(['Category','Genres'])['App'].count().sort_values(ascending = False)
```

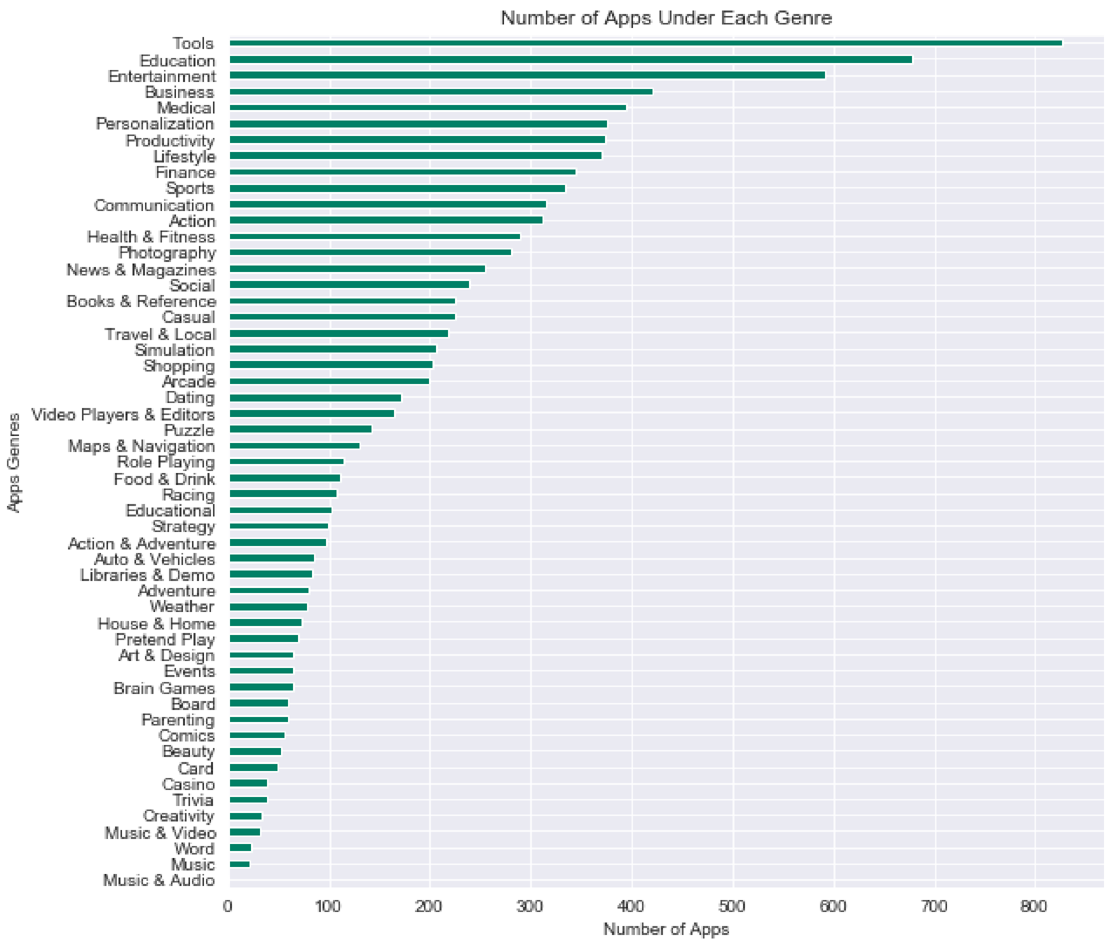
```
Out[177]: Category   Genres
          Parenting  Parenting        60
                     Education         7
                     Music & Video     6
                     Brain Games       1
          Name: App, dtype: int64
```

The apps developers can benefit from the results above to develop an app under a rare genre and category to fufill the market needs.

What are the top 10 app genres ?

```
In [28]: # checking the count of Apps produced genres
         app_cats = google_store_clean.groupby('Genres')['App'].count().sort_values()
         fig, ax = plt.subplots(figsize=(9,9))
         app_cats.plot(ax=ax,kind="barh",colormap='summer')
         ax.set_xlabel('Number of Apps')
         ax.set_ylabel('Apps Genres')
         ax.set_title('Number of Apps Under Each Genre')
```

```
Out[28]: Text(0.5, 1.0, 'Number of Apps Under Each Genre')
```



The top 10 genres for Android mobile applications currently are :

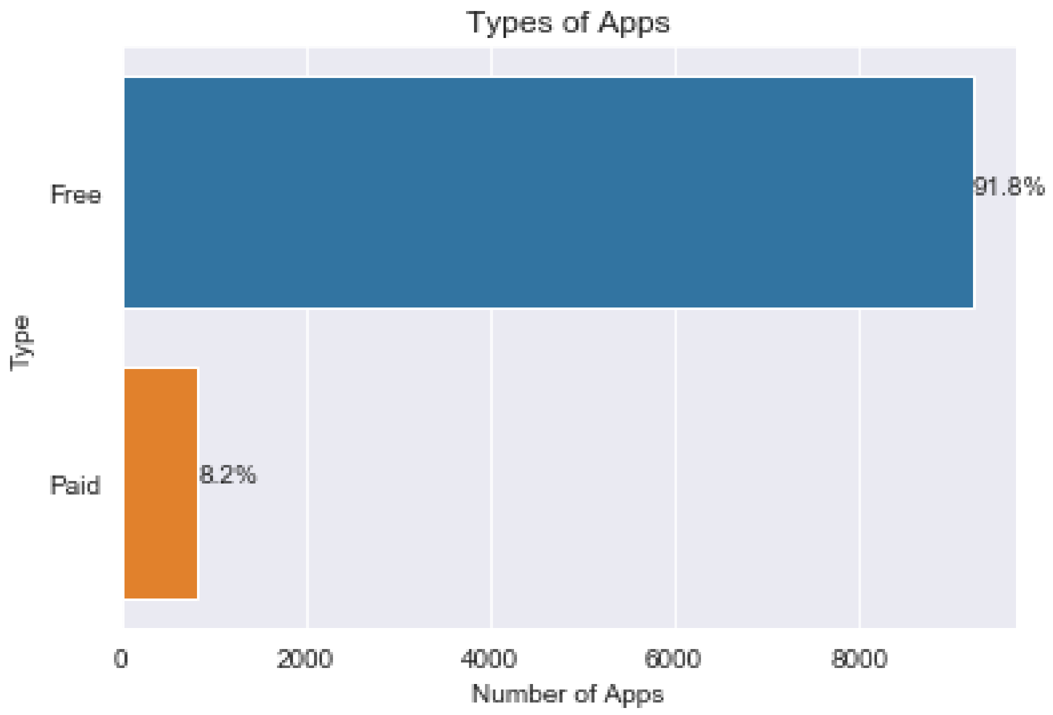
1. Tools.
2. Education.
3. Entertainment.
4. Business.
5. Medical.

6. Personalization.
7. Productivity.
8. Lifestyle.
9. Finance.
10. Sports.

Which is more produced : free apps or paid apps ?

```
In [18]: # drawing the plot
         ax = sns.countplot(y="Type", data=google_store_clean)
         # setting the plot title
         plt.title('Types of Apps')
         # setting the x label title on the plot
         plt.xlabel('Number of Apps')

         # dispalying the precentage for each app type on the plot
         total = len(google_store_clean['Type'])
         for p in ax.patches:
                 percentage = '{:.1f}%'.format(100 * p.get_width()/total)
                 x = p.get_x() + p.get_width() + 0.02
                 y = p.get_y() + p.get_height()/2
                 ax.annotate(percentage, (x, y))
         # showing the plot
         plt.show()
```

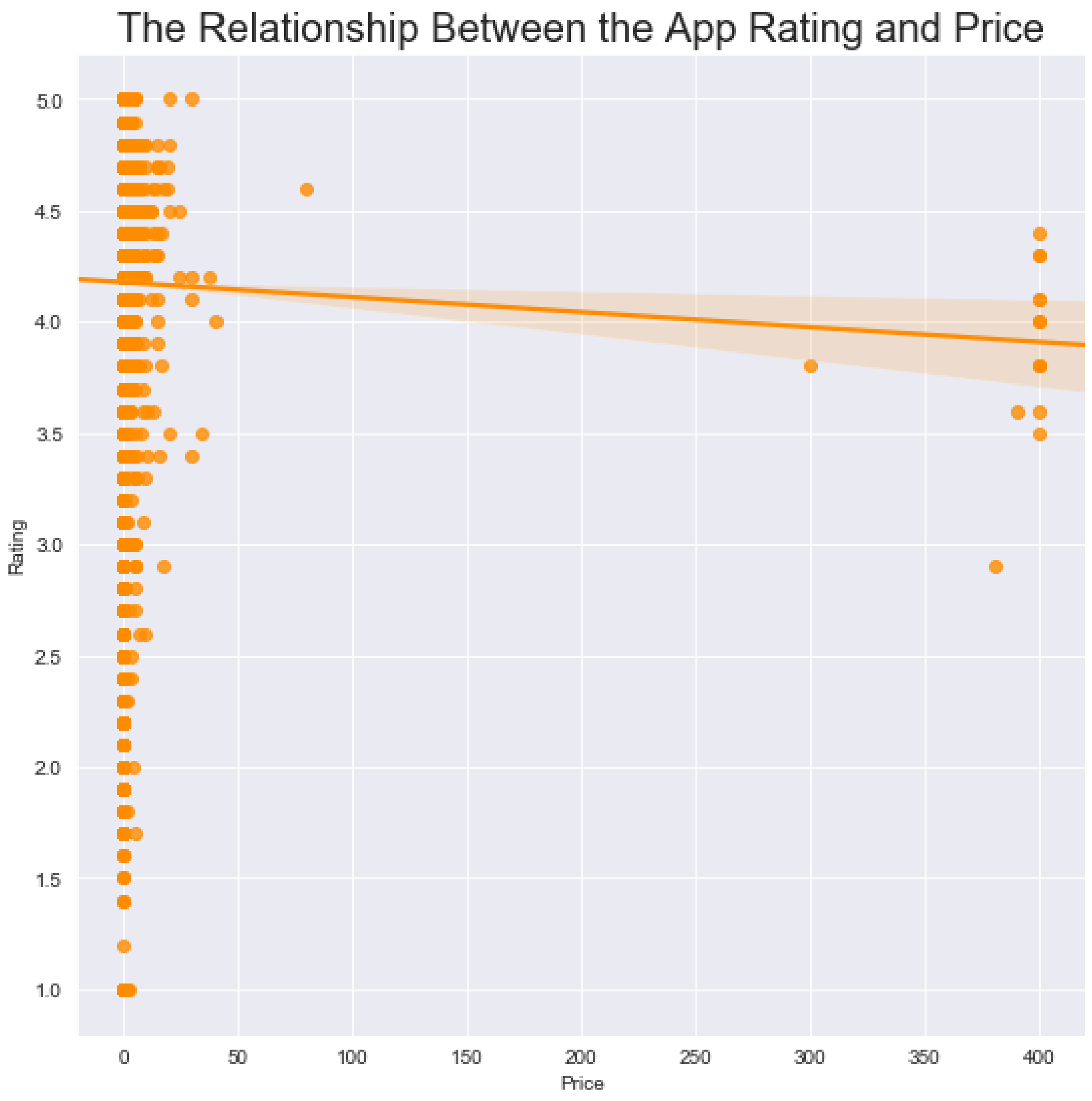


It can be seen from the above graph that free apps are more than paid apps. They account for 91.8% of total apps while the paid apps account for only 8.2%.

Is their a relationship between the price of the app and the number of app installs ?

In [188]:
```
# plotting the scatter plot to show the relationship between the app price and the n
plt.figure(figsize = (9,9))
sns.regplot(x="Price", y="Rating", color = 'darkorange',data=google_store_clean)
plt.title('The Relationship Between the App Rating and Price',size = 20)
```

Out[188]:
```
Text(0.5, 1.0, 'The Relationship Between the App Rating and Price')
```



In [183]:
```
# find the correlation coffient between the two variables
google_store_clean['Price'].corr(google_store_clean['Installs'])
```

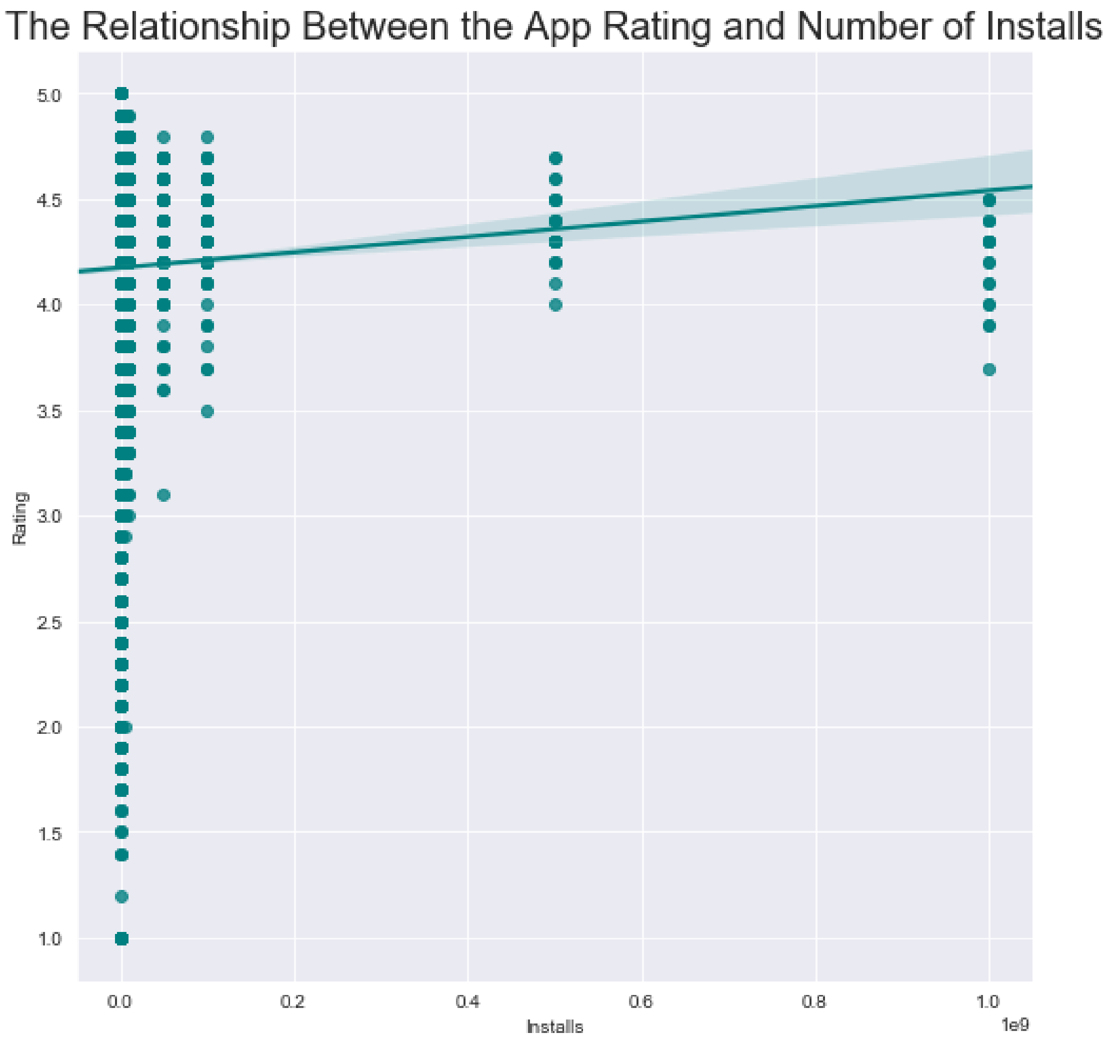
```
Out[183]: -0.009416933693207025
```

There is no apparent relationship between the price of the app and the number of installs. Therefore, I believe that the apps developers must focus on providing well featured apps that benefit users to get a more positive rating and profits.

Is there a relationship between the app rating the number of installs ?

```
In [191]: # plotting the scatter plot to show the relationship between the app price and the n
          plt.figure(figsize = (9,9))
          sns.regplot(x="Installs", y="Rating", color = 'teal',data=google_store_clean)
          plt.title('The Relationship Between the App Rating and Number of Installs',size = 20)
```

```
Out[191]: Text(0.5, 1.0, 'The Relationship Between the App Rating and Number of Installs')
```



```
In [192]: # find the correlation coffient between the two variables
          google_store_clean['Rating'].corr(google_store_clean['Installs'])

Out[192]: 0.03936064045150467
```

It appears that there is no apparent relationship between the app rating and the number of installs of the app itself. Therefore, I believe that the apps developers must focus on providing well featured apps that benefit users to get a more positive rating and profits.

What is the average number of installs for apps that target a specific audience under each app category ?

```
In [56]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Sports"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[56]: Category  Content Rating
         Sports    Teen                 9.754400e+06
                   Everyone 10+         3.651571e+06
                   Everyone             3.087840e+06
                   Mature 17+           2.333333e+06
                   Adults only 18+      1.000000e+06
         Name: Installs, dtype: float64

In [57]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Social"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[57]: Category  Content Rating
         Social    Teen               4.209964e+07
                   Mature 17+         1.565824e+07
                   Everyone           4.158332e+06
                   Everyone 10+       5.050000e+05
         Name: Installs, dtype: float64

In [58]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Comics"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[58]: Category  Content Rating
         Comics    Teen                 1.515000e+06
                   Adults only 18+      5.000000e+05
                   Everyone             4.639296e+05
                   Mature 17+           3.683333e+05
                   Everyone 10+         1.700167e+05
         Name: Installs, dtype: float64
```

```
In [59]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Tools"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[59]: Category  Content Rating
         Tools     Teen              1.200402e+07
                   Everyone          9.697191e+06
                   Unrated           5.000000e+04
                   Mature 17+        2.550000e+03
         Name: Installs, dtype: float64

In [60]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Communication"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[60]: Category       Content Rating
         Communication  Everyone          3.579839e+07
                        Teen              3.091173e+07
                        Mature 17+        2.251375e+07
         Name: Installs, dtype: float64

In [62]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="News And Magazines"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[62]: Category            Content Rating
         News And Magazines  Mature 17+        4.850136e+07
                             Teen              3.916242e+07
                             Everyone 10+      1.291211e+07
                             Everyone          6.311639e+05
         Name: Installs, dtype: float64

In [63]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Health And Fitness"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[63]: Category            Content Rating
         Health And Fitness  Everyone          4.310933e+06
                             Teen              1.428989e+06
                             Mature 17+        6.202000e+05
                             Everyone 10+      2.583517e+05
         Name: Installs, dtype: float64

In [64]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Family"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
Out[64]: Category  Content Rating
         Family    Everyone 10+      5.251043e+06
                   Teen              3.626739e+06
                   Everyone          2.232681e+06
                   Mature 17+        1.278601e+06
                   Unrated           5.000000e+02
         Name: Installs, dtype: float64
```

```
In [65]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Books And Reference"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[65]: Category            Content Rating
         Books And Reference  Teen              8.433029e+07
                              Everyone 10+      2.777500e+06
                              Mature 17+        1.833667e+06
                              Everyone          1.089614e+06
         Name: Installs, dtype: float64
```

```
In [66]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Lifestyle"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[66]: Category   Content Rating
         Lifestyle  Mature 17+        1.251289e+07
                    Everyone 10+      2.040200e+06
                    Everyone          1.145595e+06
                    Teen              3.808100e+05
         Name: Installs, dtype: float64
```

```
In [67]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Business"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[67]: Category  Content Rating
         Business  Everyone          1.721140e+06
                   Teen              7.917692e+03
                   Everyone 10+      1.000000e+02
                   Mature 17+        5.000000e+00
         Name: Installs, dtype: float64
```

```
In [68]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Shopping"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
Out[68]: Category  Content Rating
         Shopping  Teen              2.307593e+07
                   Mature 17+        5.033333e+06
                   Everyone          4.431387e+06
         Name: Installs, dtype: float64
```

```
In [69]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Medical"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[69]: Category  Content Rating
         Medical   Everyone          100541.795756
                   Everyone 10+       21513.750000
                   Teen               12222.000000
                   Mature 17+         11140.000000
         Name: Installs, dtype: float64
```

```
In [70]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Game"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[70]: Category  Content Rating
         Game      Everyone 10+      2.652648e+07
                   Everyone          1.708689e+07
                   Mature 17+        7.871111e+06
                   Teen              7.616992e+06
         Name: Installs, dtype: float64
```

```
In [71]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Finance"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[71]: Category  Content Rating
         Finance   Everyone          1.335990e+06
                   Teen              2.224000e+05
         Name: Installs, dtype: float64
```

```
In [72]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Personalization"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[72]: Category         Content Rating
         Personalization  Teen              6.680516e+06
                          Everyone          3.786358e+06
                          Everyone 10+      2.502000e+06
                          Mature 17+        1.912233e+05
         Name: Installs, dtype: float64
```

```
In [73]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Photography"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[73]: Category     Content Rating
         Photography  Mature 17+        3.700000e+07
                      Everyone          1.633075e+07
                      Teen              1.615070e+07
         Name: Installs, dtype: float64

In [74]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Travel And Local"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[74]: Category          Content Rating
         Travel And Local  Everyone          1.344313e+07
                           Mature 17+        1.000000e+07
                           Teen              3.600008e+06
         Name: Installs, dtype: float64

In [75]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Dating"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[75]: Category  Content Rating
         Dating    Mature 17+        957915.868966
                   Teen              112323.333333
                   Everyone           59846.823529
         Name: Installs, dtype: float64

In [76]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Productivity"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[76]: Category      Content Rating
         Productivity  Everyone          1.593136e+07
                       Teen              1.111347e+06
                       Everyone 10+      5.000000e+03
                       Mature 17+        1.000000e+03
         Name: Installs, dtype: float64

In [77]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Art And Design"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
Out[77]: Category        Content Rating
         Art And Design  Teen              2.000333e+07
                         Everyone          8.304194e+05
                         Everyone 10+      5.000000e+05
         Name: Installs, dtype: float64
```

```
In [78]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Food And Drink"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[78]: Category        Content Rating
         Food And Drink  Teen              2.137576e+06
                         Everyone          1.907727e+06
                         Everyone 10+      5.500000e+04
         Name: Installs, dtype: float64
```

```
In [79]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Video Players"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[79]: Category       Content Rating
         Video Players  Teen              1.160074e+08
                        Everyone          1.073027e+07
                        Everyone 10+      6.700000e+06
                        Mature 17+        3.533333e+06
         Name: Installs, dtype: float64
```

```
In [80]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="House And Home"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[80]: Category        Content Rating
         House And Home  Everyone          1.343090e+06
                         Teen              2.550000e+05
         Name: Installs, dtype: float64
```

```
In [81]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Maps And Navigation"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[81]: Category             Content Rating
         Maps And Navigation  Everyone          3.922613e+06
                              Teen              2.550000e+06
                              Mature 17+        1.000000e+04
                              Everyone 10+      1.000000e+02
         Name: Installs, dtype: float64
```

```
In [82]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Entertainment"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[82]: Category       Content Rating
         Entertainment  Teen              2.795632e+07
                        Everyone          1.243667e+07
                        Everyone 10+      5.500000e+06
                        Mature 17+        3.850000e+06
         Name: Installs, dtype: float64

In [83]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Events"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[83]: Category  Content Rating
         Events    Everyone          278583.981132
                   Teen              138388.750000
                   Everyone 10+       33700.000000
         Name: Installs, dtype: float64

In [84]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Education"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[84]: Category   Content Rating
         Education  Everyone 10+      7.000000e+06
                    Everyone          3.675030e+06
                    Teen              1.000000e+06
                    Mature 17+        6.833333e+05
         Name: Installs, dtype: float64

In [85]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Auto And Vehicles"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi

Out[85]: Category           Content Rating
         Auto And Vehicles  Teen              1.000000e+07
                            Everyone 10+      1.000000e+06
                            Everyone          5.075929e+05
         Name: Installs, dtype: float64

In [86]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Weather"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
Out[86]: Category  Content Rating
         Weather   Everyone 10+      1.000000e+07
                   Everyone          4.660007e+06
                   Mature 17+        1.000000e+06
                   Teen              3.000000e+05
         Name: Installs, dtype: float64
```

```
In [87]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Beauty"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[87]: Category  Content Rating
         Beauty    Everyone          593712.222222
                   Teen               83333.333333
                   Everyone 10+       55000.000000
                   Mature 17+         40000.000000
         Name: Installs, dtype: float64
```

```
In [88]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Libraries And Demo"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[88]: Category            Content Rating
         Libraries And Demo  Everyone          630903.690476
         Name: Installs, dtype: float64
```

```
In [91]: # selecting an app category
         # grouping by category and content rating with the average number of installs and the
         category = google_store_clean.query('Category =="Parenting"')
         category.groupby(['Category','Content Rating'])['Installs'].mean().sort_values(ascendi
```

```
Out[91]: Category   Content Rating
         Parenting  Everyone          586820.972222
                    Mature 17+        100000.000000
                    Teen               50000.000000
         Name: Installs, dtype: float64
```

What is the average rating for apps that target specific audience under each app category ?

```
In [95]: # selecting an app category
         # grouping by category and content rating with the rating average and then sorting ds
         category = google_store_clean.query('Category =="Sports"')
         category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascending
```

```
Out[95]: Category  Content Rating
         Sports    Adults only 18+    4.500000
                   Mature 17+         4.400000
```

```
           Everyone 10+       4.366667
           Everyone           4.211814
           Teen               4.161538
Name: Rating, dtype: float64
```

```
In [96]: # selecting an app category
         # grouping by category and content rating with the rating average and then sorting ds
         category = google_store_clean.query('Category =="Social"')
         category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendin
```

```
Out[96]: Category  Content Rating
         Social    Everyone          4.307042
                   Teen              4.289535
                   Everyone 10+      4.100000
                   Mature 17+        4.075000
         Name: Rating, dtype: float64
```

```
In [97]: # selecting an app category
         # grouping by category and content rating with the rating average and then sorting ds
         category = google_store_clean.query('Category =="Comics"')
         category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendin
```

```
Out[97]: Category  Content Rating
         Comics    Everyone 10+       4.450000
                   Everyone           4.361538
                   Adults only 18+    4.200000
                   Teen               4.036842
                   Mature 17+         3.866667
         Name: Rating, dtype: float64
```

```
In [98]: # selecting an app category
         # grouping by category and content rating with the rating average and then sorting ds
         category = google_store_clean.query('Category =="Tools"')
         category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendin
```

```
Out[98]: Category  Content Rating
         Tools     Teen              4.50000
                   Unrated           4.10000
                   Everyone          4.03736
                   Mature 17+        3.70000
         Name: Rating, dtype: float64
```

```
In [99]: # selecting an app category
         # grouping by category and content rating with the rating average and then sorting ds
         category = google_store_clean.query('Category =="Communication"')
         category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendin
```

```
Out[99]: Category       Content Rating
         Communication  Teen              4.294737
```

```
              Mature 17+        4.285714
              Everyone          4.102174
Name: Rating, dtype: float64
```

```
In [100]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="News And Magazines"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[100]: Category            Content Rating
          News And Magazines  Mature 17+        4.270000
                              Teen              4.234615
                              Everyone          4.120968
                              Everyone 10+      4.022727
          Name: Rating, dtype: float64
```

```
In [101]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Health And Fitness"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[101]: Category            Content Rating
          Health And Fitness  Mature 17+        4.340000
                              Teen              4.325000
                              Everyone          4.240826
                              Everyone 10+      3.980000
          Name: Rating, dtype: float64
```

```
In [102]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Family"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[102]: Category  Content Rating
          Family    Everyone 10+      4.234677
                    Everyone          4.198764
                    Mature 17+        4.193182
                    Teen              4.148869
                    Unrated                NaN
          Name: Rating, dtype: float64
```

```
In [103]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Books And Reference"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[103]: Category             Content Rating
          Books And Reference  Everyone 10+      4.475000
                               Everyone          4.351370
```

```
                     Teen                4.287500
                     Mature 17+          4.166667
Name: Rating, dtype: float64
```

```
In [104]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Lifestyle"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[104]: Category   Content Rating
          Lifestyle  Teen              4.435294
                     Mature 17+        4.255556
                     Everyone          4.070849
                     Everyone 10+      3.840000
          Name: Rating, dtype: float64
```

```
In [105]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Business"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[105]: Category  Content Rating
          Business  Teen              4.300000
                    Everyone          4.095367
                    Everyone 10+           NaN
                    Mature 17+             NaN
          Name: Rating, dtype: float64
```

```
In [106]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Shopping"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[106]: Category  Content Rating
          Shopping  Teen              4.315385
                    Mature 17+        4.266667
                    Everyone          4.214570
          Name: Rating, dtype: float64
```

```
In [107]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Medical"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[107]: Category  Content Rating
          Medical   Teen              4.533333
                    Mature 17+        4.425000
                    Everyone 10+      4.371429
                    Everyone          4.153623
          Name: Rating, dtype: float64
```

```
In [108]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Game"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[108]: Category  Content Rating
          Game      Everyone 10+      4.316822
                    Teen              4.245833
                    Everyone          4.237879
                    Mature 17+        4.230769
          Name: Rating, dtype: float64
```

```
In [109]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Finance"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[109]: Category  Content Rating
          Finance   Everyone          4.116107
                    Teen              4.075000
          Name: Rating, dtype: float64
```

```
In [110]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Personalization"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[110]: Category         Content Rating
          Personalization  Mature 17+        4.500000
                           Everyone 10+      4.380000
                           Teen              4.360870
                           Everyone          4.323664
          Name: Rating, dtype: float64
```

```
In [111]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Photography"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[111]: Category     Content Rating
          Photography  Mature 17+        4.4000
                       Teen              4.2600
                       Everyone          4.1504
          Name: Rating, dtype: float64
```

```
In [112]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Travel And Local"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
Out[112]: Category          Content Rating
          Travel And Local  Mature 17+        4.600000
                            Everyone          4.067033
                            Teen              4.060000
          Name: Rating, dtype: float64
```

```
In [113]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Dating"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[113]: Category  Content Rating
          Dating    Mature 17+        3.973600
                    Teen              3.966667
                    Everyone          3.900000
          Name: Rating, dtype: float64
```

```
In [114]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Productivity"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[114]: Category      Content Rating
          Productivity  Mature 17+        4.600000
                        Everyone          4.184848
                        Teen              4.050000
                        Everyone 10+      3.600000
          Name: Rating, dtype: float64
```

```
In [115]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Art And Design"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[115]: Category        Content Rating
          Art And Design  Everyone 10+      4.700000
                          Teen              4.466667
                          Everyone          4.346032
          Name: Rating, dtype: float64
```

```
In [116]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Food And Drink"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[116]: Category        Content Rating
          Food And Drink  Teen              4.414286
                          Everyone 10+      4.300000
                          Everyone          4.149412
          Name: Rating, dtype: float64
```

```
In [117]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Video Players"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi

Out[117]: Category       Content Rating
          Video Players  Everyone 10+      4.100000
                         Teen              4.057143
                         Everyone          4.047154
                         Mature 17+        3.650000
          Name: Rating, dtype: float64

In [118]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="House And Home"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi

Out[118]: Category        Content Rating
          House And Home  Teen              4.650000
                          Everyone          4.133333
          Name: Rating, dtype: float64

In [119]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Maps And Navigation"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi

Out[119]: Category             Content Rating
          Maps And Navigation  Teen              4.400000
                               Everyone          4.041739
                               Mature 17+        2.700000
                               Everyone 10+           NaN
          Name: Rating, dtype: float64

In [120]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Entertainment"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi

Out[120]: Category       Content Rating
          Entertainment  Mature 17+        4.250000
                         Everyone          4.182222
                         Teen              4.114035
                         Everyone 10+      3.950000
          Name: Rating, dtype: float64

In [121]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Events"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
Out[121]: Category  Content Rating
          Events    Teen              4.542857
                    Everyone 10+      4.500000
                    Everyone          4.411111
          Name: Rating, dtype: float64
```

```
In [122]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Education"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[122]: Category   Content Rating
          Education  Teen              4.800000
                     Everyone 10+      4.500000
                     Everyone          4.377863
                     Mature 17+        4.166667
          Name: Rating, dtype: float64
```

```
In [123]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Auto And Vehicles"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[123]: Category           Content Rating
          Auto And Vehicles  Everyone 10+      4.300000
                             Teen              4.200000
                             Everyone          4.188732
          Name: Rating, dtype: float64
```

```
In [124]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Weather"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[124]: Category  Content Rating
          Weather   Mature 17+        4.700000
                    Teen              4.450000
                    Everyone 10+      4.400000
                    Everyone          4.227941
          Name: Rating, dtype: float64
```

```
In [125]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Beauty"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[125]: Category  Content Rating
          Beauty    Mature 17+        4.500000
                    Everyone          4.287179
```

```
                Teen               4.000000
                Everyone 10+            NaN
         Name: Rating, dtype: float64
```

```
In [126]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Libraries And Demo"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[126]: Category            Content Rating
          Libraries And Demo  Everyone          4.178125
          Name: Rating, dtype: float64
```

```
In [127]: # selecting an app category
          # grouping by category and content rating with the rating average and then sorting d
          category = google_store_clean.query('Category =="Parenting"')
          category.groupby(['Category','Content Rating'])['Rating'].mean().sort_values(ascendi
```

```
Out[127]: Category   Content Rating
          Parenting  Teen              4.700000
                     Mature 17+        4.600000
                     Everyone          4.260345
          Name: Rating, dtype: float64
```

The above results can help developers estimate the expected average rating for their newly developed apps in the market.

0.1.3 Conclusion

The apps development field is very a competitive field. Thus, apps developers need to develop apps that are unique that offer beneficial features for users in a way that fulfill their needs in the best way. App developers must understand their audience very well and be updated to the new trend in the market. The project will help defentiyl apps developers to have some insight before developing any kind of mobile apps.