In [1]:
```
# Import statements
# Import necessary python libraries and packages


# For data analysis & manipulation
import pandas as pd
import numpy as np

# For visualising distributional values
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
# Python version

import sys
print ("The Python version is: {}".format(sys.version))
```

```
The Python version is: 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64
bit (AMD64)]
```

In [3]:
```
# Generating the version of a wide variety of packages/libraries used

pd.__version__
pd.show_versions(as_json=False)
```

```
INSTALLED VERSIONS
------------------
commit: None
python: 3.7.3.final.0
python-bits: 64
OS: Windows
OS-release: 10
machine: AMD64
processor: Intel64 Family 6 Model 78 Stepping 3, GenuineIntel
byteorder: little
LC_ALL: None
LANG: None
LOCALE: None.None

pandas: 0.24.2
pytest: 5.0.1
pip: 19.1.1
setuptools: 41.0.1
Cython: 0.29.12
numpy: 1.16.4
scipy: 1.2.1
pyarrow: None
xarray: None
IPython: 7.6.1
sphinx: 2.1.2
patsy: 0.5.1
dateutil: 2.8.0
pytz: 2019.1
blosc: None
bottleneck: 1.2.1
tables: 3.5.2
numexpr: 2.6.9
feather: None
matplotlib: 3.1.0
openpyxl: 2.6.2
xlrd: 1.2.0
xlwt: 1.3.0
xlsxwriter: 1.1.8
lxml.etree: 4.3.4
bs4: 4.7.1
html5lib: 1.0.1
sqlalchemy: 1.3.5
pymysql: None
psycopg2: None
jinja2: 2.10.1
s3fs: None
fastparquet: None
pandas_gbq: None
pandas_datareader: None
gcsfs: None
```

In [4]:
```
# Assigning the dataset with the name: "app"

app= pd.read_csv(r"C:\Users\Δημητρης\Desktop\other\Strathclyde\master\strathclyde\Lectures\Semester 1\Big Data Technologies_CS982\Assignments\Projects\google-playstore-apps\Google-Playstore-Full.csv", low_memory=False)
```

In [5]:
```
# The type of this dataset is a dataframe
type(app)
```

Out[5]:
```
pandas.core.frame.DataFrame
```

In [6]:
```
# The columns of this dataframe are "series"
type(app["Installs"])
```

Out[6]:
```
pandas.core.series.Series
```

In [7]:
```
# First 5 rows of the dataframe

app.head()
```

Out[7]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Conten Rating
0	DoorDash - Food Delivery	FOOD_AND_DRINK	4.548561573	305034	5,000,000+	Varies with device	0	Everyone
1	TripAdvisor Hotels Flights Restaurants Attract...	TRAVEL_AND_LOCAL	4.400671482	1207922	100,000,000+	Varies with device	0	Everyone
2	Peapod	SHOPPING	3.656329393	1967	100,000+	1.4M	0	Everyone
3	foodpanda - Local Food Delivery	FOOD_AND_DRINK	4.107232571	389154	10,000,000+	16M	0	Everyone
4	My CookBook Pro (Ad Free)	FOOD_AND_DRINK	4.647752285	2291	10,000+	Varies with device	$5.99	Everyone

In [8]:
```
# Getting the last five rows

app.tail()
```

Out[8]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Conten Ratin
267047	Community Healthplex	HEALTH_AND_FITNESS	5	1	100+	4.2M	0	Everyon
267048	Pet ads: Buy & Sell	BUSINESS	2.599999905	5	500+	8.4M	0	Everyon
267049	Collectors Market: Buy & Sell	BUSINESS	3.285714388	7	1,000+	7.9M	0	Everyon
267050	Car Market, Buy & Sell	BUSINESS	5	1	1,000+	8.2M	0	Everyon
267051	Selfie with Ariana Grande	PHOTOGRAPHY	4.611111164	18	1,000+	7.8M	0	Everyon

In [9]:
```
# Getting the number of rows and columns of the dataframe

app.shape
```

Out[9]:
```
(267052, 15)
```

In [10]:
```
# Removing the columns with index position: 11, 12, 13, 14. They do not seem t
o offer any substantial value to the data analysis

app=app.drop("Unnamed: 11", axis=1)
app=app.drop("Unnamed: 12", axis=1)
app=app.drop("Unnamed: 13", axis=1)
app=app.drop("Unnamed: 14", axis=1)
```

In [11]:
```
# Number of rows and columns after removing the useless columns

app.shape
```

Out[11]:
```
(267052, 11)
```

In [12]:
```
# Columns after removing the useless ones

app.columns
```

Out[12]:
```
Index(['App Name', 'Category', 'Rating', 'Reviews', 'Installs', 'Size',
       'Price', 'Content Rating', 'Last Updated', 'Minimum Version',
       'Latest Version'],
      dtype='object')
```

In [13]:
```
# Number of app categories

app["Category"].nunique()
```

Out[13]:
```
67
```

In [14]:
```
# The app categories

app.Category.unique()
```

Out[14]:
```
array(['FOOD_AND_DRINK', 'TRAVEL_AND_LOCAL', 'SHOPPING', 'LIFESTYLE',
       'GAME_ACTION', 'GAME_CASUAL', 'GAME_ROLE_PLAYING', 'GAME_PUZZLE',
       'GAME_RACING', 'GAME_ADVENTURE', 'GAME_ARCADE', 'GAME_STRATEGY',
       'GAME_SPORTS', 'GAME_SIMULATION', 'GAME_MUSIC', 'MUSIC_AND_AUDIO',
       'FINANCE', 'EVENTS', 'ENTERTAINMENT', 'EDUCATION',
       'GAME_EDUCATIONAL', 'BOOKS_AND_REFERENCE', 'NEWS_AND_MAGAZINES',
       'PHOTOGRAPHY', 'VIDEO_PLAYERS', 'GAME_WORD', 'ART_AND_DESIGN',
       'GAME_TRIVIA', 'GAME_BOARD', 'BUSINESS', 'PRODUCTIVITY',
       'COMMUNICATION', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME', 'SOCIAL',
       'BEAUTY', 'GAME_CASINO', 'MAPS_AND_NAVIGATION', 'PERSONALIZATION',
       'GAME_CARD', 'TOOLS', 'SPORTS', 'AUTO_AND_VEHICLES',
       'LIBRARIES_AND_DEMO', 'COMICS', 'PARENTING', 'DATING', 'WEATHER',
       'MEDICAL', ' Podcasts', ')', ' Channel 2 News', nan,
       ' Breaking News', '6', 'Gate ALARM', ' Alfabe ?ren',
       ' T rk Alfabesi', ' not notified you follow -', ' Mexpost)',
       ' Romantic Song Music Love Songs', ' ETEA & MDCAT', ' Tour Guide',
       'TRAVEL', ' Speaker Pro 2019', ' Islamic Name Boy & Girl+Meaning',
       ' Accounting', ' super loud speaker booster'], dtype=object)
```

In [15]:
```
# Viewing the number of classes (gradation) of the number of installations
# There are 38 different classes

app["Installs"].nunique()
```

Out[15]:
```
38
```

In [16]:
```
# The gradation of installations in the dataframe

# There seem to be some input mistakes, such as "EDUCATION", which should not
 belong in this column. They will be edited


app.Installs.unique()
```

Out[16]:
```
array(['5,000,000+', '100,000,000+', '100,000+', '10,000,000+', '10,000+',
       '1,000,000+', '50,000,000+', '500,000+', '50,000+', '5,000+',
       '1,000+', '500,000,000+', '1,000,000,000+', '5,000,000,000+',
       '100+', '500+', '50+', '5+', '10+', '1+', 'EDUCATION', '6',
       '11976', '0+', '20', '156', ' Xmax X', '166', '1', '54', '71',
       '59', '13', '117', '511', '927', '4.823529243', '10'], dtype=object)
```

In [17]:
```
# There are a lot of app sizes

app["Size"].nunique()
```

Out[17]:
```
1248
```

In [18]:
```
# Viewing the content rating; who is permitted to download these apps

# There are some invalid contents. They will be edited

app["Content Rating"].unique()
```

Out[18]:
```
array(['Everyone', 'Teen', 'Everyone 10+', 'Mature 17+',
       'Adults only 18+', '100,000+', 'Unrated', '$0.99', '0', '3702',
       '$2.49', '17M'], dtype=object)
```

In [19]:
```
# the number of categories of the age content rating

len(app["Content Rating"].unique())
```

Out[19]:
```
12
```

In [20]:
```
#current first five rows

app.head()
```

Out[20]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Conten Rating
0	DoorDash - Food Delivery	FOOD_AND_DRINK	4.548561573	305034	5,000,000+	Varies with device	0	Everyone
1	TripAdvisor Hotels Flights Restaurants Attract...	TRAVEL_AND_LOCAL	4.400671482	1207922	100,000,000+	Varies with device	0	Everyone
2	Peapod	SHOPPING	3.656329393	1967	100,000+	1.4M	0	Everyone
3	foodpanda - Local Food Delivery	FOOD_AND_DRINK	4.107232571	389154	10,000,000+	16M	0	Everyone
4	My CookBook Pro (Ad Free)	FOOD_AND_DRINK	4.647752285	2291	10,000+	Varies with device	$5.99	Everyone

In [21]:
```
app.isnull().sum()
```

Out[21]:
```
App Name           1
Category           1
Rating             0
Reviews            1
Installs           0
Size               0
Price              0
Content Rating     0
Last Updated       0
Minimum Version    1
Latest Version     3
dtype: int64
```

In [22]:
```
# There are totally 11 empty data entries which will be dropped
len(app.isnull().sum())
```

Out[22]:
```
11
```

In [23]:
```
# Dropping the entries where there are missing values

app=app.dropna()
```

In [24]:
```
app.isnull().any()

# False for every category means that there are no longer missing values
```

Out[24]:
```
App Name           False
Category           False
Rating             False
Reviews            False
Installs           False
Size               False
Price              False
Content Rating     False
Last Updated       False
Minimum Version    False
Latest Version     False
dtype: bool
```

In [25]:
```
# Ensuring there are no missing values in any column, in any data of every col
umn

app.isnull().any().any()
```

Out[25]:
```
False
```

In [26]:
```
##############################################################################
#########################################
```

In [27]:
```
##############################################################################
#########################################
```

In [28]:
```
##############################################################################
#########################################
```

Cleaning of the Data - Exploring and Managing the Data

In [29]:
```
# Start of cleaning
# There were given some commands to locate any invalid data
# I noticed that are some misplacing, e.g. here, "4" should move to "Rating",
 and "GAME_STRATEGY" should move to "Category"

# Wherever the data are misplaced but valid, the data will be kept and edited
 (correcting the entry positions)
# Wherever the data are misplaced and invalid too (with lot's of mistakes), th
e data will be removed

app[app["Rating"]== "GAME_STRATEGY"]
```

Out[29]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	M
13504	Never have I ever 18+	)	GAME_STRATEGY	4	6	100+	2.4M	$0.99	Mature 17+	D

In [30]:
```
# dropping the invalid entry

app.drop(index=13504, inplace=True)
```

In [31]:
```
# Now the column "Rating" is fixed

app[app["Rating"]== "GAME_STRATEGY"]
```

Out[31]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [32]:
```
# Noticing the same pattern. Wrong entry data in the columns

app[app["Rating"]== "NEWS_AND_MAGAZINES"]
```

Out[32]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Con Ra
23457	Israel News	Channel 2 News	NEWS_AND_MAGAZINES	3.857798815	11976	1,000,000+	Varies with device	
48438	Mojo Times: Bihar Hindi Video News	Breaking News	NEWS_AND_MAGAZINES	4.775640965	156	10,000+	6.9M	

In [33]:
```
# Here the data are misplaced but valid

# I am manually fixing the misplacing data values

app.loc[23457, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "NEW
S_AND_MAGAZINES", "3.857798815", "11976", "1,000,000+", "Varies with device",
"0", "Everyone 10+", "March 16, 2019", "Varies with device", "NaN"
```

In [34]:
```
app.loc[23457]
```

Out[34]:
```
App Name                  Israel News
Category            NEWS_AND_MAGAZINES
Rating                     3.857798815
Reviews                          11976
Installs                     1,000,000+
Size                Varies with device
Price                                0
Content Rating            Everyone 10+
Last Updated            March 16, 2019
Minimum Version     Varies with device
Latest Version                     NaN
Name: 23457, dtype: object
```

In [35]:
```
app.loc[48438, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "NEW
S_AND_MAGAZINES", "4.775640965", "156", "10,000+", "6.9M", "0", "Teen", "March
30, 2019","4.1 and up", "NaN"
```

In [36]:
```
app.loc[48438]
```

Out[36]:
```
App Name           Mojo Times: Bihar Hindi Video News
Category                            NEWS_AND_MAGAZINES
Rating                                    4.775640965
Reviews                                           156
Installs                                       10,000+
Size                                              6.9M
Price                                                0
Content Rating                                    Teen
Last Updated                            March 30, 2019
Minimum Version                             4.1 and up
Latest Version                                     NaN
Name: 48438, dtype: object
```

In [ ]:

In [37]:
```
# Here is an example of misplaced data with a lot of mistakes. It does not see
m important to be fixed, it will be dropped

app[app["Rating"]== "ENTERTAINMENT"]
```

Out[37]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated
113151	Steins	Gate ALARM	ENTERTAINMENT	4.716867447	166	500+	67M	$0.99	Teer

In [38]:
```
app.drop(index=113151, inplace=True)
```

In [39]:
```
# Ensuring that there are no longer wrong entries in the column "Rating"

app[app["Rating"]== "ENTERTAINMENT"]
```

Out[39]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [40]:
```
app[app["Rating"]== "EDUCATION"]
```

Out[40]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Las Update
125479	2-6 Ya? E?itici ?ocuk Zeka Oyunlar?	Alfabe ?ren	EDUCATION	5	1	10+	57M	$2.49	Everyon
125480	2-6 Ya? E?itici ?ocuk Zeka Oyunlar?	T?rk Alfabesi	EDUCATION	4.333333492	54	50,000+	43M	0	Everyon
180371	eShagird - Online academy	ETEA & MDCAT	EDUCATION	4.504273415	117	10,000+	6.9M	0	Everyon

In [41]:
```
# Dropping these data entries which do not seem important and they have a lot
 of mistakes

app.drop(index=125479, inplace=True)
app.drop(index=125480, inplace=True)
app.drop(index=180371, inplace=True)
app[app["Rating"]== "EDUCATION"]
```

Out[41]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [42]:
```
# In this line respecting the column "Rating" there are misplaced but valid da
ta

# Data will be fixed manually, putting them in the correct position

app[app["Rating"]== "SOCIAL"]
```

Out[42]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Mini Ve
165230	Shytter - Twitter client	not notified you follow -	SOCIAL	4.098591328	71	5,000+	7.7M	0	Everyone	30,

In [43]:
```
# Fixing the data entry positions manually

app.loc[165230, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "SOC
IAL", "4.098591328", "71", "5,000+", "7.7M", "0", "Everyone", "March 30, 2019"
,"4.1 and up", "NaN"
```

In [44]:
```
app[app["Rating"]== "SOCIAL"]
```

Out[44]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [45]:
```
app[app["Rating"]== "PRODUCTIVITY"]
```

Out[45]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
168914	CorreosTrack 2.0 (Correos de M\u\xico	Mexpost)	PRODUCTIVITY	4.389830589	59	10,000+	16M	0

In [46]:
```
# Fixing the data entry positions manually for the index position 168914



app.loc[168914, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "PRO
DUCTIVITY", "4.389830589", "59", "10,000+", "16M", "0", "Everyone", "December
 21, 2018","4.1 and up", "NaN"
app[app["Rating"]== "PRODUCTIVITY"] # Ensuring that column "Rating" is fixed f
rom this kind of data entry
```

Out[46]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [ ]:

In [47]:
```
app[app["Rating"]== "MUSIC_AND_AUDIO"]
```

Out[47]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
177165	Music Love Song	Romantic Song Music Love Songs	MUSIC_AND_AUDIO	4.538461685	13	1,000+	Varies with device	0
193869	Equalizer & Volume Booster	Speaker Pro 2019	MUSIC_AND_AUDIO	4.632093906	511	10,000+	2.5M	0
257773	High Volume Booster	super loud speaker booster	MUSIC_AND_AUDIO	4.400000095	10	1,000+	3.5M	0

In [48]:
```
# Same logic here. Misplaced but valid data. They will be edited manually

app.loc[177165, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "MUS
IC_AND_AUDIO", "4.538461685", "13", "1,000+", "Varies with device", "0", "Tee
n", "October 24, 2018","Varies with device", "NaN"
app.loc[193869, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "MUS
IC_AND_AUDIO", "4.632093906", "511", "10,000+", "2.5M", "0", "Everyone", "Sept
ember 25, 2018","2.3 and up", "NaN"
app.loc[257773, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "MUS
IC_AND_AUDIO", "4.400000095", "10", "1,000+", "3.5M", "0", "Everyone", "Novemb
er 7, 2018","4.0 and up", "NaN"
app[app["Rating"]== "PRODUCTIVITY"]
```

Out[48]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [49]:
```
app[app["Rating"]== "TRAVEL_AND_LOCAL"]
```

Out[49]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	I Upda
190759	Friend in Iceland	Tour Guide	TRAVEL_AND_LOCAL	5	6	1,000+	27M	0	Every

In [50]:
```
# Fixing the entries in index position 190759 manually (misplaced but substant
ial values)

app.loc[190759, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "TRA
VEL_AND_LOCAL", "5", "6", "1,000+", "27M", "0", "Everyone", "October 16, 2017"
, "4.0 and up", "NaN"

app[app["Rating"]== "TRAVEL_AND_LOCAL"]
```

Out[50]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [51]:
```
app[app["Rating"]== "LIFESTYLE"]
```

Out[51]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	L Updat
194165	Muslim Baby Names	Islamic Name Boy & Girl+Meaning	LIFESTYLE	4.388349533	927	100,000+	3.7M	0	Everyc

In [52]:
```
# same logic as previously

app.loc[194165, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = "LIF
ESTYLE", "4.388349533", "927", "100,000+", "3.7M", "0", "Everyone", "May 23, 2
018", "4.0 and up", "NaN"
app[app["Rating"]== "LIFESTYLE"]
```

Out[52]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [ ]:

In [53]:
```
app[app["Rating"]== " Economics"]
```

Out[53]:

	App Name	Category	Rating	Reviews	Installs	Size	Price
232811	Learn Accounts - Finance	Accounting	Economics	BOOKS_AND_REFERENCE	4.823529243	17	1,000+

In [54]:
```
# Applying the same logic. Correting the misplaced (but valid) data

app.loc[232811, ["Category", "Rating", "Reviews", "Installs", "Size", "Price",
"Content Rating", "Last Updated", "Minimum Version", "Latest Version"]] = " Ec
onomics", "4.823529243", "17", "1,000+", "17M", "0", "Everyone", "October 22,
 2018", "NaN", "NaN"
app[app["Rating"]== " Economics"]
```

Out[54]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [55]:
```
# Here we had an entry in column "Rating" which was 7. But we want "Rating<=
5".
# It was fixed so now there is no longer rating with numbers more than "5"

app[app["Rating"]==7.000000]
```

Out[55]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [56]:
```
app.drop(index=99584, inplace=True)
app[app["Rating"]==7.000000]
```

Out[56]:

App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	Last Updated	Minimum Version	Latest Version

In [57]:
```
# Converting the column "Rating" to float so that we can apply statistics

app.Rating= app.Rating.astype(float)
```

In [58]:
```
app.describe()
```

Out[58]:

	Rating
count	267040.000000
mean	4.269390
std	0.586244
min	1.000000
25%	4.017699
50%	4.382165
75%	4.648649
max	5.000000

In [59]:
```
# Converting the data in the column "Reviews" to float to that we can apply st
atistics

app.Reviews= app.Reviews.astype(float)
```

In [60]:
```
app.describe()
```

Out[60]:

	Rating	Reviews
count	267040.000000	2.670400e+05
mean	4.269390	1.459628e+04
std	0.586244	4.110638e+05
min	1.000000	1.000000e+00
25%	4.017699	1.600000e+01
50%	4.382165	9.300000e+01
75%	4.648649	6.560000e+02
max	5.000000	8.621429e+07

In [61]:
```
# I want to convert the column "Installs" into float

# I firstly have to remove the "+"

app.Installs= app["Installs"].str.replace("+", "")
```

In [62]:
```
# I had some problems converting the column into float, even when i removed
 "+"

# I am removing the commas

app.Installs= app["Installs"].str.replace(",", "")
```

In [63]:
```
app["Installs"] = app.Installs.convert_objects(convert_numeric = True)
```

```
C:\Users\Anaconda\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: Future
Warning: convert_objects is deprecated.  To re-infer data dtypes for object c
olumns, use Series.infer_objects()
For all other conversions use the data-type specific converters pd.to_datetim
e, pd.to_timedelta and pd.to_numeric.
  """Entry point for launching an IPython kernel.
```

In [64]:
```
# Removing "$" from the data entries in the column "Price" so that it can be c
onverted to float

app["Price"]= app["Price"].str.replace("$", "")
```

In [65]:
```
# Convert the data in "Price" to float


app["Price"]= app.Price.astype(float)
```

In [66]:
```
# the data in the column "Prics" successfully converted to float

# In these columns i can do various statistical applications
app.describe()
```

Out[66]:

	Rating	Reviews	Installs	Price
count	267040.000000	2.670400e+05	2.670400e+05	267040.000000
mean	4.269390	1.459628e+04	6.410638e+05	0.227872
std	0.586244	4.110638e+05	2.046801e+07	3.559421
min	1.000000	1.000000e+00	0.000000e+00	0.000000
25%	4.017699	1.600000e+01	1.000000e+03	0.000000
50%	4.382165	9.300000e+01	1.000000e+04	0.000000
75%	4.648649	6.560000e+02	5.000000e+04	0.000000
max	5.000000	8.621429e+07	5.000000e+09	399.990000

In [67]:
```
# procedure for converting the column "Size" to float
# there are sizes counted in mb, kb, in numbers without measurement unit and w
ith "varies with device"
app.Size.unique()
```

Out[67]:
```
array(['Varies with device', '1.4M', '16M', ..., '601k', '715k', '311k'],
      dtype=object)
```

In [68]:
```
# removing the "m" which is the mb for the size

app.Size= app["Size"].str.replace("M", "")
```

In [69]:
```
# assigning "Varies with device" with a number like "-1" so that i can seperat
e it later

# app.Size= app["Size"].str.replace("Varies with device", "-1")
```

In [70]:
```
# Segmenting the column of the size

y= app.iloc[:, 5:6]
```

In [71]:
```
# I tried to fix the last problems in converting the column "Size" to float
# Here i am trying to remove "k" (kbs) and the blanks, and to convert kbs to m
bs
# It keeps giving me errors and i cannot fix it
# i will not use the column "Size" for statistical applications

#for x in y:
#    x = str(x)
#    x= x.replace(" ", "")
#    x= x.replace(",", ".")
#    if "k" in x:
#        x= x.replace("k", "")
#        x=x.replace(" k", "")
#        x=x.replace("k ", "")
#        x= float(x)
#        x= x/1024
```

In [72]:
```
# There are 11,728 apps whose size varies with device

len(app[app["Size"]== "Varies with device"])
```

Out[72]:
```
11728
```

In [73]:
```
app.Size.describe()
```

Out[73]:
```
count                 267040
unique                  1236
top       Varies with device
freq                   11728
Name: Size, dtype: object
```

In [74]:
```
print ("Apps whose size varies with device are {}% of the dataset".format(1172
8/267040*100))
```

```
Apps whose size varies with device are 4.391851408028759% of the dataset
```

In [75]:
```
##############################################################################
##########################
```

In [76]:
```
##############################################################################
##########################
```

In [77]:
```
##############################################################################
##########################
```

Statistical Analysis

In [78]:
```
#ensuring the shape of dataframe before proceeding to further statistics and v
isualization

app.shape
```

Out[78]: (267040, 11)

In [79]:
```
# the columns, the data of which we can do statistic manipulation

app.describe()
```

Out[79]:

	Rating	Reviews	Installs	Price
count	267040.000000	2.670400e+05	2.670400e+05	267040.000000
mean	4.269390	1.459628e+04	6.410638e+05	0.227872
std	0.586244	4.110638e+05	2.046801e+07	3.559421
min	1.000000	1.000000e+00	0.000000e+00	0.000000
25%	4.017699	1.600000e+01	1.000000e+03	0.000000
50%	4.382165	9.300000e+01	1.000000e+04	0.000000
75%	4.648649	6.560000e+02	5.000000e+04	0.000000
max	5.000000	8.621429e+07	5.000000e+09	399.990000

In [ ]:

In [80]:
```
app.info() # data type for each column
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 267040 entries, 0 to 267051
Data columns (total 11 columns):
App Name           267040 non-null object
Category           267040 non-null object
Rating             267040 non-null float64
Reviews            267040 non-null float64
Installs           267040 non-null int64
Size               267040 non-null object
Price              267040 non-null float64
Content Rating     267040 non-null object
Last Updated       267040 non-null object
Minimum Version    267040 non-null object
Latest Version     267040 non-null object
dtypes: float64(3), int64(1), object(7)
memory usage: 24.4+ MB
```

In [81]:
```
#reinsuring there are no any missing values
app.isnull().any().any()
```

Out[81]: False

In [82]:
```
#*************************************************************************
************************
# Reviewing the unique values and the number of unique values in each column a
fter the cleaning process
```

In [83]:
```
# Values in "Category"
app["Category"].unique()
```

Out[83]:
```
array(['FOOD_AND_DRINK', 'TRAVEL_AND_LOCAL', 'SHOPPING', 'LIFESTYLE',
       'GAME_ACTION', 'GAME_CASUAL', 'GAME_ROLE_PLAYING', 'GAME_PUZZLE',
       'GAME_RACING', 'GAME_ADVENTURE', 'GAME_ARCADE', 'GAME_STRATEGY',
       'GAME_SPORTS', 'GAME_SIMULATION', 'GAME_MUSIC', 'MUSIC_AND_AUDIO',
       'FINANCE', 'EVENTS', 'ENTERTAINMENT', 'EDUCATION',
       'GAME_EDUCATIONAL', 'BOOKS_AND_REFERENCE', 'NEWS_AND_MAGAZINES',
       'PHOTOGRAPHY', 'VIDEO_PLAYERS', 'GAME_WORD', 'ART_AND_DESIGN',
       'GAME_TRIVIA', 'GAME_BOARD', 'BUSINESS', 'PRODUCTIVITY',
       'COMMUNICATION', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME', 'SOCIAL',
       'BEAUTY', 'GAME_CASINO', 'MAPS_AND_NAVIGATION', 'PERSONALIZATION',
       'GAME_CARD', 'TOOLS', 'SPORTS', 'AUTO_AND_VEHICLES',
       'LIBRARIES_AND_DEMO', 'COMICS', 'PARENTING', 'DATING', 'WEATHER',
       'MEDICAL', 'TRAVEL', ' Economics'], dtype=object)
```

In [84]:
```
# There are 51 different categories

app["Category"].nunique()
```

Out[84]:
```
51
```

In [85]:
```
# Unique values of Rating
app["Rating"].unique()
```

Out[85]:
```
array([4.54856157, 4.40067148, 3.65632939, ..., 4.06342983, 3.9687779 ,
       4.60038614])
```

In [86]:
```
# There are 99,845 unique values of Rating

app["Rating"].nunique()
```

Out[86]:
```
99845
```

In [87]:
```
# Unique values of the column "Reviews"
app["Reviews"].unique()
```

Out[87]:
```
array([ 305034., 1207922.,    1967., ...,  296774.,    7974.,   69123.])
```

In [88]:
```
# There are 24,531 different reviews

app["Reviews"].nunique()
```

Out[88]:
```
24531
```

In [89]:
```
# Unique values of installations

app["Installs"].unique()
```

Out[89]:
```
array([   5000000,  100000000,     100000,   10000000,      10000,
          1000000,   50000000,     500000,      50000,       5000,
             1000,  500000000, 1000000000, 5000000000,        100,
              500,         50,          5,         10,          1,
                0], dtype=int64)
```

In [90]:
```
# There are 21 different classes of installations

app["Installs"].nunique()
```

Out[90]:
```
21
```

In [91]:
```
# Unique values in the column "Size"
app["Size"].unique()
```

Out[91]:
```
array(['Varies with device', '1.4', '16', ..., '601k', '715k', '311k'],
      dtype=object)
```

In [92]:
```
# There are 1,236 different sizes for the apps

app["Size"].nunique()
```

Out[92]:
```
1236
```

In [93]:
```
# There are 488 different prices
app["Price"].nunique()
```

Out[93]:
```
488
```

In [94]:
```
# Unique values of the column "Content Rating"
app["Content Rating"].unique()
```

Out[94]:
```
array(['Everyone', 'Teen', 'Everyone 10+', 'Mature 17+',
       'Adults only 18+', 'Unrated'], dtype=object)
```

In [95]:
```
# There are 6 different content ratings

len(app["Content Rating"].unique())
```

Out[95]:
```
6
```

In [96]:
```
print("**************************************************")

print("Minimum number of ratings: %.2f" %app["Rating"].min())
print("Maximum number of ratings: %.2f" %app["Rating"].max())

print("**************************************************")

print("Minimum number of reviews: %.2f" %app["Reviews"].min())
print("Maximum number of reviews: %.2f" %app["Reviews"].max())

print("**************************************************")

print("Minimum number of installs: %.2f" %app["Installs"].min())
print("Maximum number of installs: %.2f" %app["Installs"].max())

print("**************************************************")


print("Minimum number of prices: %.2f" %app["Price"].min())
print("Maximum number of prices: %.2f" %app["Price"].max())

print("**************************************************")
```

```
**************************************************
Minimum number of ratings: 1.00
Maximum number of ratings: 5.00
**************************************************
Minimum number of reviews: 1.00
Maximum number of reviews: 86214292.00
**************************************************
Minimum number of installs: 0.00
Maximum number of installs: 5000000000.00
**************************************************
Minimum number of prices: 0.00
Maximum number of prices: 399.99
**************************************************
```

In [97]:
```
# Getting the measures of central tendency for all the installation grouped by
"Category"

app.groupby("Category").Installs.agg(["min", "mean", "median", "max"])
```

Out[97]:

Category	min	mean	median	max
Economics	1000	1.000000e+03	1000	1000
ART_AND_DESIGN	1	2.105946e+05	5000	100000000
AUTO_AND_VEHICLES	1	1.824824e+05	5000	10000000
BEAUTY	0	1.409585e+05	10000	10000000
BOOKS_AND_REFERENCE	0	1.008366e+05	10000	100000000
BUSINESS	0	1.455845e+05	1000	100000000
COMICS	5	3.832805e+05	50000	10000000
COMMUNICATION	0	2.419759e+06	10000	1000000000
DATING	1	4.595702e+05	10000	10000000
EDUCATION	0	7.466526e+04	5000	100000000
ENTERTAINMENT	0	3.573559e+05	10000	1000000000
EVENTS	1	4.602683e+04	1000	10000000
FINANCE	0	2.236191e+05	10000	100000000
FOOD_AND_DRINK	1	1.930530e+05	5000	50000000
GAME_ACTION	0	4.063549e+06	100000	500000000
GAME_ADVENTURE	1	1.155658e+06	10000	100000000
GAME_ARCADE	1	2.960377e+06	10000	1000000000
GAME_BOARD	5	1.159017e+06	10000	100000000
GAME_CARD	1	8.000250e+05	100000	100000000
GAME_CASINO	5	1.769368e+06	100000	50000000
GAME_CASUAL	0	2.696059e+06	100000	500000000
GAME_EDUCATIONAL	0	6.160472e+05	10000	100000000
GAME_MUSIC	5	2.268880e+06	100000	100000000
GAME_PUZZLE	0	1.184301e+06	10000	100000000
GAME_RACING	1	5.540269e+06	500000	500000000
GAME_ROLE_PLAYING	10	1.139481e+06	100000	50000000
GAME_SIMULATION	5	1.807155e+06	100000	100000000
GAME_SPORTS	10	3.580514e+06	100000	100000000
GAME_STRATEGY	1	2.544039e+06	100000	500000000
GAME_TRIVIA	1	6.068273e+05	10000	100000000
GAME_WORD	0	1.046894e+06	100000	50000000
HEALTH_AND_FITNESS	0	2.837032e+05	10000	500000000
HOUSE_AND_HOME	1	3.353328e+05	5000	100000000
LIBRARIES_AND_DEMO	1	1.630496e+05	5000	10000000

Category	min	mean	median	max
LIFESTYLE	0	1.463108e+05	10000	100000000
MAPS_AND_NAVIGATION	1	2.986137e+05	10000	100000000
MEDICAL	1	5.832896e+04	5000	5000000
MUSIC_AND_AUDIO	0	3.264714e+05	10000	1000000000
NEWS_AND_MAGAZINES	0	4.623790e+05	10000	1000000000
PARENTING	5	1.447583e+05	10000	10000000
PERSONALIZATION	0	3.970350e+05	10000	100000000
PHOTOGRAPHY	0	1.292555e+06	50000	1000000000
PRODUCTIVITY	0	1.391869e+06	10000	1000000000
SHOPPING	0	6.134706e+05	10000	100000000
SOCIAL	1	1.490510e+06	10000	1000000000
SPORTS	1	1.497417e+05	10000	50000000
TOOLS	0	1.104303e+06	10000	5000000000
TRAVEL	10000	1.000000e+04	10000	10000
TRAVEL_AND_LOCAL	0	1.222950e+06	10000	5000000000
VIDEO_PLAYERS	1	3.554788e+06	10000	5000000000
WEATHER	5	5.188793e+05	10000	100000000

In [98]:
```
# Sorting (descending sorting) the dataframe by number of installs

app.sort_values(by="Installs", ascending= False)
```

Out[98]:

	App Name	Category	Rating	Reviews	Installs	Size	Pri
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.(
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.(
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.(
842	Google Chrome: Fast & Secure	COMMUNICATION	4.335205	13636591.0	1000000000	Varies with device	0.(
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.(
2147	Google Street View	TRAVEL_AND_LOCAL	4.215697	2171998.0	1000000000	Varies with device	0.(
28676	Samsung Print Service Plugin	PRODUCTIVITY	4.204499	322275.0	1000000000	Varies with device	0.(
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	Varies with device	0.(
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.(
6412	Google Play Movies & TV	VIDEO_PLAYERS	3.703356	1048972.0	1000000000	Varies with device	0.(
6781	Google Play Games	ENTERTAINMENT	4.304268	8900879.0	1000000000	Varies with device	0.(
1922	Cloud Print	PRODUCTIVITY	4.111720	323021.0	1000000000	Varies with device	0.(
1236	Google Drive	PRODUCTIVITY	4.402619	3683909.0	1000000000	Varies with device	0.(
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.(
2724	Gmail	COMMUNICATION	4.346980	5614163.0	1000000000	Varies with device	0.(
2522	Google News	NEWS_AND_MAGAZINES	3.978565	1058436.0	1000000000	15	0.(
2674	Gboard - the Google Keyboard	TOOLS	4.335172	2841568.0	1000000000	Varies with device	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Pri
632	Messenger \u\u Text and Video Chat for Free	COMMUNICATION	4.085856	65469531.0	1000000000	Varies with device	0.(
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.(
1981	Hangouts	COMMUNICATION	4.040543	3960560.0	1000000000	Varies with device	0.(
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.(
671	WhatsApp Messenger	COMMUNICATION	4.417610	86214292.0	1000000000	Varies with device	0.(
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.(
3180	Google Play Music	MUSIC_AND_AUDIO	3.950097	3878214.0	1000000000	Varies with device	0.(
815	Skype - free IM & video calls	COMMUNICATION	4.134610	10746013.0	1000000000	Varies with device	0.(
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.(
831	Facebook Lite	SOCIAL	4.288809	10866006.0	1000000000	Varies with device	0.(
3138	Twitter	NEWS_AND_MAGAZINES	4.311234	12960504.0	500000000	Varies with device	0.(
820	Pou	GAME_CASUAL	4.330340	10752323.0	500000000	24	0.(
2118	MX Player	VIDEO_PLAYERS	4.486509	7169292.0	500000000	Varies with device	0.(
...	...	...	...	...	...	...	
258894	Athavan Play	NEWS_AND_MAGAZINES	4.200000	5.0	0	5.8	0.(
173309	SimbiBot App | Your intelligent learning partner	EDUCATION	5.000000	2.0	0	5.0	0.(
251707	GodPool	TOOLS	3.666667	3.0	0	2.2	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Pri
166483	?? ?? ?? ??????????	HEALTH_AND_FITNESS	5.000000	2.0	0	20	0.(
136703	Kotlin Tutorial : Learn Kotlin For Android	EDUCATION	4.250000	4.0	0	6.4	0.(
239513	Latest Inspirational Quotes Wallpaper	PERSONALIZATION	5.000000	1.0	0	9.6	0.(
67229	???????? ?????? ??	BOOKS_AND_REFERENCE	3.500000	2.0	0	3.7	0.(
188753	Blank Sticker for Whatsapp	COMMUNICATION	5.000000	2.0	0	3.0	0.(
203989	Terjemah Kitab Qurrotul Uyun	BOOKS_AND_REFERENCE	4.000000	1.0	0	5.8	0.(
251001	Tamil Dictionary Offline & Multilingual Transl...	BOOKS_AND_REFERENCE	5.000000	1.0	0	11	0.(
246287	Motivational Quotes - Motivation, success, goals	LIFESTYLE	5.000000	1.0	0	4.9	0.(
184817	Diabetes Ratgeber AR	HEALTH_AND_FITNESS	1.250000	4.0	0	60	0.(
230876	?????????? ?Feng Shui Khmer Name	EDUCATION	4.333333	3.0	0	5.8	0.(
108150	Cute Wallpapers	PERSONALIZATION	4.692307	13.0	0	7.4	0.(
88725	Nedis 4K CAM	TOOLS	3.500000	4.0	0	8.5	0.(
193415	RC Bot	GAME_EDUCATIONAL	3.000000	4.0	0	3.1	0.(
35872	Idram Merchant	FINANCE	5.000000	4.0	0	2.9	0.(
226376	Say My Text (Speech synthesizer)	TOOLS	3.000000	2.0	0	5.4	0.(
61284	SSVM WORLD SCHOOL	EDUCATION	4.439024	41.0	0	2.5	0.(
265585	DEUTSCH WITZE 2019	COMMUNICATION	5.000000	1.0	0	4.3	0.(
156537	Web R?dio HB Publicidade	ENTERTAINMENT	5.000000	2.0	0	4.7	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Pri
264007	Fm Resplandecer Misiones	MUSIC_AND_AUDIO	5.000000	1.0	0	2.1	0.(
136862	Breakwall VPN - Unlimited Free Proxy VPN	TOOLS	4.272727	11.0	0	11	0.(
128489	FitKids 10-13 Jahre Premium	HEALTH_AND_FITNESS	5.000000	1.0	0	41	3.9
162112	Reasoning Aptitude Test: Tips & Tricks	EDUCATION	4.000000	4.0	0	8.9	0.(
253182	La Barbiera Sasso Marconi	BEAUTY	5.000000	2.0	0	10	0.(
249287	Rich & Rich	LIFESTYLE	5.000000	1.0	0	16	369.9
24784	DUA KE QURAN AMHARIC	EDUCATION	4.730337	89.0	0	7.9	0.(
166479	??? LED ??????	ENTERTAINMENT	1.941176	17.0	0	2.4	0.(
260161	Flugpreise Vergleichen & G►nstige Fl►ge Low Cost	TRAVEL_AND_LOCAL	5.000000	1.0	0	7.9	0.(

267040 rows × 11 columns

In [99]:
```
top_installed_apps=app.sort_values(by="Installs", ascending= False)
```

In [100]:
```
#*************************************************************

# top 10 apps based on the number of installations

#*************************************************************

top_installed_apps.head(10)
```

Out[100]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Conten Ratin
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Tee
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyon
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyon
842	Google Chrome: Fast & Secure	COMMUNICATION	4.335205	13636591.0	1000000000	Varies with device	0.0	Everyon
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyon 10
2147	Google Street View	TRAVEL_AND_LOCAL	4.215697	2171998.0	1000000000	Varies with device	0.0	Everyon
28676	Samsung Print Service Plugin	PRODUCTIVITY	4.204499	322275.0	1000000000	Varies with device	0.0	Everyon
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	Varies with device	0.0	Tee
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.0	Everyon
6412	Google Play Movies & TV	VIDEO_PLAYERS	3.703356	1048972.0	1000000000	Varies with device	0.0	Tee

In [101]:
```
# Apps with 5 billion installations (5b is the 1st greater class of installati
ons in the dataset)

len(app[app["Installs"]>= 5000000000])
```

Out[101]: 3

In [102]:
```
# Apps with more than 1 billion installations (1b is the 2nd greater class of
 installations in the dataset)

len(app[app["Installs"]>= 1000000000])
```

Out[102]: 27

In [ ]:

In [103]:
```
top_installed_and_rated_apps = app.sort_values(by=["Installs", "Rating"], asce
nding=False)
top_installed_and_rated_apps # main top apps
```

Out[103]:

	App Name	Category	Rating	Reviews	Installs	Size	Pric
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.(
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.(
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.(
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.(
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.(
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.(
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.(
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.(
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.(
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.(
671	WhatsApp Messenger	COMMUNICATION	4.417610	86214292.0	1000000000	Varies with device	0.(
1236	Google Drive	PRODUCTIVITY	4.402619	3683909.0	1000000000	Varies with device	0.(
2724	Gmail	COMMUNICATION	4.346980	5614163.0	1000000000	Varies with device	0.(
842	Google Chrome: Fast & Secure	COMMUNICATION	4.335205	13636591.0	1000000000	Varies with device	0.(
2674	Gboard - the Google Keyboard	TOOLS	4.335172	2841568.0	1000000000	Varies with device	0.(
6781	Google Play Games	ENTERTAINMENT	4.304268	8900879.0	1000000000	Varies with device	0.(
831	Facebook Lite	SOCIAL	4.288809	10866006.0	1000000000	Varies with device	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Price
2147	Google Street View	TRAVEL_AND_LOCAL	4.215697	2171998.0	1000000000	Varies with device	0.(
28676	Samsung Print Service Plugin	PRODUCTIVITY	4.204499	322275.0	1000000000	Varies with device	0.(
815	Skype - free IM & video calls	COMMUNICATION	4.134610	10746013.0	1000000000	Varies with device	0.(
1922	Cloud Print	PRODUCTIVITY	4.111720	323021.0	1000000000	Varies with device	0.(
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	Varies with device	0.(
632	Messenger – Text and Video Chat for Free	COMMUNICATION	4.085856	65469531.0	1000000000	Varies with device	0.(
1981	Hangouts	COMMUNICATION	4.040543	3960560.0	1000000000	Varies with device	0.(
2522	Google News	NEWS_AND_MAGAZINES	3.978565	1058436.0	1000000000	15	0.(
3180	Google Play Music	MUSIC_AND_AUDIO	3.950097	3878214.0	1000000000	Varies with device	0.(
6412	Google Play Movies & TV	VIDEO_PLAYERS	3.703356	1048972.0	1000000000	Varies with device	0.(
85295	Samsung Gallery	PHOTOGRAPHY	4.745800	1545980.0	500000000	20	0.(
12638	Security Master - Antivirus, VPN, AppLock, Boo...	TOOLS	4.652842	25532160.0	500000000	Varies with device	0.(
628	Clash of Clans	GAME_STRATEGY	4.606215	48401470.0	500000000	103	0.(
...	...	...	...	...	...	...	..
260161	Flugpreise Vergleichen & G►nstige Fl►ge Low Cost	TRAVEL_AND_LOCAL	5.000000	1.0	0	7.9	0.(
264007	Fm Resplandecer Misiones	MUSIC_AND_AUDIO	5.000000	1.0	0	2.1	0.(
265585	DEUTSCH WITZE 2019	COMMUNICATION	5.000000	1.0	0	4.3	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Price
120624	Hisnul Muslim	BOOKS_AND_REFERENCE	4.750000	12.0	0	3.4	0.(
24784	DUA KE QURAN AMHARIC	EDUCATION	4.730337	89.0	0	7.9	0.(
108150	Cute Wallpapers	PERSONALIZATION	4.692307	13.0	0	7.4	0.(
57454	Wish her happiness.	ENTERTAINMENT	4.666667	12.0	0	42	0.(
88397	Nederland Zingt	MUSIC_AND_AUDIO	4.666667	12.0	0	2.6	0.(
61284	SSVM WORLD SCHOOL	EDUCATION	4.439024	41.0	0	2.5	0.(
230876	?????????? ?Feng Shui Khmer Name	EDUCATION	4.333333	3.0	0	5.8	0.(
255563	Drink H2O	HEALTH_AND_FITNESS	4.333333	9.0	0	1.5	0.(
136862	Breakwall VPN - Unlimited Free Proxy VPN	TOOLS	4.272727	11.0	0	11	0.(
136703	Kotlin Tutorial : Learn Kotlin For Android	EDUCATION	4.250000	4.0	0	6.4	0.(
258894	Athavan Play	NEWS_AND_MAGAZINES	4.200000	5.0	0	5.8	0.(
23720	Bubble Shooter	GAME_PUZZLE	4.130435	23.0	0	19	0.(
159451	IP Info	TOOLS	4.000000	1.0	0	3.1	0.(
162112	Reasoning Aptitude Test: Tips & Tricks	EDUCATION	4.000000	4.0	0	8.9	0.(
193500	Packt Notifier	TOOLS	4.000000	4.0	0	2.1	0.(
203989	Terjemah Kitab Qurrotul Uyun	BOOKS_AND_REFERENCE	4.000000	1.0	0	5.8	0.(
251707	GodPool	TOOLS	3.666667	3.0	0	2.2	0.(
148023	Astucia Naval Batalla Clanes	ENTERTAINMENT	3.583333	12.0	0	80	0.(
67229	???????? ?????? ??	BOOKS_AND_REFERENCE	3.500000	2.0	0	3.7	0.(

	App Name	Category	Rating	Reviews	Installs	Size	Price
88725	Nedis 4K CAM	TOOLS	3.500000	4.0	0	8.5	0.(
136964	iCycle: Recycling Made Easy	LIFESTYLE	3.000000	6.0	0	5.5	0.(
193415	RC Bot	GAME_EDUCATIONAL	3.000000	4.0	0	3.1	0.(
226376	Say My Text (Speech synthesizer)	TOOLS	3.000000	2.0	0	5.4	0.(
39673	Vadaa Hunt	GAME_CASUAL	2.333333	3.0	0	20	0.(
225594	PIP Collage Maker Professional	PHOTOGRAPHY	2.130435	46.0	0	8.3	0.(
166479	??? LED ??????	ENTERTAINMENT	1.941176	17.0	0	2.4	0.(
184817	Diabetes Ratgeber AR	HEALTH_AND_FITNESS	1.250000	4.0	0	60	0.(

267040 rows × 11 columns

In [104]:
```
#*************************************************************************
# top 10 apps based on the number of installations and rating together (main t
op apps)
#*************************************************************************



top_installed_and_rated_apps.head(10)
```

Out[104]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyone
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Teen
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyone
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyone
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.0	Everyone
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.0	Everyone
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.0	Everyone
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Teen
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyone 10+
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.0	Everyone

In [105]:
```
top_installed_and_reviewed_apps = app.sort_values(by=["Installs", "Reviews"],
ascending=False)
top_installed_and_reviewed_apps
```

Out[105]:

	App Name	Category	Rating	Reviews	Installs	
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	V de
821	Google	TOOLS	4.408893	10870728.0	5000000000	V de
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	V de
671	WhatsApp Messenger	COMMUNICATION	4.417610	86214292.0	1000000000	V de
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	V de
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	V de
632	Messenger ⬆ Text and Video Chat for Free	COMMUNICATION	4.085856	65469531.0	1000000000	V de
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	V de
842	Google Chrome: Fast & Secure	COMMUNICATION	4.335205	13636591.0	1000000000	V de
831	Facebook Lite	SOCIAL	4.288809	10866006.0	1000000000	V de
815	Skype - free IM & video calls	COMMUNICATION	4.134610	10746013.0	1000000000	V de
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	
6781	Google Play Games	ENTERTAINMENT	4.304268	8900879.0	1000000000	V de
2724	Gmail	COMMUNICATION	4.346980	5614163.0	1000000000	V de
1981	Hangouts	COMMUNICATION	4.040543	3960560.0	1000000000	V de

	App Name	Category	Rating	Reviews	Installs	
3180	Google Play Music	MUSIC_AND_AUDIO	3.950097	3878214.0	1000000000	V de
1236	Google Drive	PRODUCTIVITY	4.402619	3683909.0	1000000000	V de
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	
2674	Gboard - the Google Keyboard	TOOLS	4.335172	2841568.0	1000000000	V de
2147	Google Street View	TRAVEL_AND_LOCAL	4.215697	2171998.0	1000000000	V de
2522	Google News	NEWS_AND_MAGAZINES	3.978565	1058436.0	1000000000	
6412	Google Play Movies & TV	VIDEO_PLAYERS	3.703356	1048972.0	1000000000	V de
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	V de
1922	Cloud Print	PRODUCTIVITY	4.111720	323021.0	1000000000	V de
28676	Samsung Print Service Plugin	PRODUCTIVITY	4.204499	322275.0	1000000000	V de
628	Clash of Clans	GAME_STRATEGY	4.606215	48401470.0	500000000	
12638	Security Master - Antivirus, VPN, AppLock, Boo...	TOOLS	4.652842	25532160.0	500000000	V de
680	Candy Crush Saga	GAME_CASUAL	4.450046	24657922.0	500000000	
...	...	...	...	...	...	
193500	Packt Notifier	TOOLS	4.000000	4.0	0	
39673	Vadaa Hunt	GAME_CASUAL	2.333333	3.0	0	
230876	??????????Feng Shui Khmer Name	EDUCATION	4.333333	3.0	0	
251707	GodPool	TOOLS	3.666667	3.0	0	
259872	FM Radio India - All India Radio station	MUSIC_AND_AUDIO	5.000000	3.0	0	
67229	???????? ?????? ??	BOOKS_AND_REFERENCE	3.500000	2.0	0	

	App Name	Category	Rating	Reviews	Installs
156537	Web R� dio HB Publicidade	ENTERTAINMENT	5.000000	2.0	0
166483	?? ?? ?? ??????????	HEALTH_AND_FITNESS	5.000000	2.0	0
173309	SimbiBot App | Your intelligent learning partner	EDUCATION	5.000000	2.0	0
183721	ArzpriceLite (Dollar & other currency prices)	FINANCE	5.000000	2.0	0
188753	Blank Sticker for Whatsapp	COMMUNICATION	5.000000	2.0	0
222482	BPSC- Bangladesh Public Service Commission	EDUCATION	5.000000	2.0	0
226376	Say My Text (Speech synthesizer)	TOOLS	3.000000	2.0	0
253182	La Barbiera Sasso Marconi	BEAUTY	5.000000	2.0	0
255326	42Club	HEALTH_AND_FITNESS	5.000000	2.0	0
59549	Ziylan Kurumsal	BUSINESS	5.000000	1.0	0
102432	Desa	SHOPPING	5.000000	1.0	0
128489	FitKids 10-13 Jahre Premium	HEALTH_AND_FITNESS	5.000000	1.0	0
136966	4 Only Jotto: The 4 Letter Logic Word Game	GAME_WORD	5.000000	1.0	0
159451	IP Info	TOOLS	4.000000	1.0	0
180978	ExpressPH VPN Lite	PRODUCTIVITY	5.000000	1.0	0
203989	Terjemah Kitab Qurrotul Uyun	BOOKS_AND_REFERENCE	4.000000	1.0	0
231949	Aquastyle24﹠????????? ? ??????	LIFESTYLE	5.000000	1.0	0
239513	Latest Inspirational Quotes Wallpaper	PERSONALIZATION	5.000000	1.0	0
246287	Motivational Quotes - Motivation, success, goals	LIFESTYLE	5.000000	1.0	0
249287	Rich & Rich	LIFESTYLE	5.000000	1.0	0
251001	Tamil Dictionary Offline & Multilingual Transl...	BOOKS_AND_REFERENCE	5.000000	1.0	0
260161	Flugpreise Vergleichen & G﹠nstige Fl﹠ge Low Cost	TRAVEL_AND_LOCAL	5.000000	1.0	0

	App Name	Category	Rating	Reviews	Installs
264007	Fm Resplandecer Misiones	MUSIC_AND_AUDIO	5.000000	1.0	0
265585	DEUTSCH WITZE 2019	COMMUNICATION	5.000000	1.0	0

267040 rows × 11 columns

In [106]:
```
#*************************************************************************
# top 10 apps based on the number of installations and reviews together
#*************************************************************************



top_installed_and_reviewed_apps.head(10)
```

Out[106]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Conten Ratin
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Tee
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyon
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyon
671	WhatsApp Messenger	COMMUNICATION	4.417610	86214292.0	1000000000	Varies with device	0.0	Everyon
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	Varies with device	0.0	Tee
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Tee
632	Messenger ♀ Text and Video Chat for Free	COMMUNICATION	4.085856	65469531.0	1000000000	Varies with device	0.0	Everyon
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyon
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyon 10
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.0	Everyon

In [107]:
```
top_10_installed_and_rated_apps= top_installed_and_rated_apps.head(10)
```

In [108]:
```
top_10_installed_and_rated_apps.Category.sort_values(ascending=False)
```

Out[108]:
```
813        VIDEO_PLAYERS
2177    TRAVEL_AND_LOCAL
3269               TOOLS
7064               TOOLS
821                TOOLS
653               SOCIAL
1259         PHOTOGRAPHY
539          GAME_ARCADE
3267       COMMUNICATION
6411       COMMUNICATION
Name: Category, dtype: object
```

In [109]:
```
# There are totally 244,396 apps
app["App Name"].nunique()
```

Out[109]:
```
244396
```

In [110]:
```
# here i will see the number of apps which belong to the categories of the mos
t installed and rated apps


count_VIDEO_PLAYERS=0
count_TRAVEL_AND_LOCAL=0
count_TOOLS=0
count_SOCIAL=0
count_PHOTOGRAPHY=0
count_GAME_ARCADE=0
count_COMMUNICATION=0

for x in app["Category"]:
    if x== "VIDEO_PLAYERS":
        count_VIDEO_PLAYERS=count_VIDEO_PLAYERS+1
    elif x== "TRAVEL_AND_LOCAL":
        count_TRAVEL_AND_LOCAL= count_TRAVEL_AND_LOCAL+1
    elif x== "TOOLS":
        count_TOOLS= count_TOOLS+1
    elif x== "SOCIAL":
        count_SOCIAL= count_SOCIAL+1
    elif x== "PHOTOGRAPHY":
        count_PHOTOGRAPHY= count_PHOTOGRAPHY+1
    elif x== "GAME_ARCADE":
        count_GAME_ARCADE= count_GAME_ARCADE+1
    elif x== "COMMUNICATION":
        count_COMMUNICATION= count_COMMUNICATION+1
print ("*********************************************************************
*******************************")
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Video Players\" is: {}".form
at(count_VIDEO_PLAYERS))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Travel and Local\" is: {}".f
ormat(count_TRAVEL_AND_LOCAL))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Tools\" is: {}".format(count
_TOOLS))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Social\" is: {}".format(coun
t_SOCIAL))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Photography\" is: {}".format
(count_PHOTOGRAPHY))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Game Arcade\" is: {}".format
(count_GAME_ARCADE))
print ("*********************************************************************
*******************************")
print ("Number of apps that belong in category: \"Communication\" is: {}".form
```

```
at(count_COMMUNICATION))
print ("*************************************************************************
******************************")
print ("*************************************************************************
******************************")
```

```
*************************************************************************
************************
*************************************************************************
************************
Number of apps that belong in category: "Video Players" is: 2717
*************************************************************************
************************
Number of apps that belong in category: "Travel and Local" is: 6650
*************************************************************************
************************
Number of apps that belong in category: "Tools" is: 21591
*************************************************************************
************************
Number of apps that belong in category: "Social" is: 4745
*************************************************************************
************************
Number of apps that belong in category: "Photography" is: 7240
*************************************************************************
************************
Number of apps that belong in category: "Game Arcade" is: 2343
*************************************************************************
************************
Number of apps that belong in category: "Communication" is: 5486
*************************************************************************
************************
*************************************************************************
************************
```

In [111]:
```
top_10_installed_and_rated_apps["Content Rating"].sort_values(ascending=False)
```

Out[111]:
```
653            Teen
813            Teen
539     Everyone 10+
3267        Everyone
1259        Everyone
3269        Everyone
6411        Everyone
7064        Everyone
2177        Everyone
821         Everyone
Name: Content Rating, dtype: object
```

In [112]:
```
app["Content Rating"].nunique()
```

Out[112]:
```
6
```

In [113]:
```
# There are totally 6 categories of content rating

# In the top 10 installed and rated apps, there are 3 different content rating
s

# I will now see their performance in the whole dataset, along with the other
 3 remaining content ratings in the whole dataset

count_Teen=0
count_Everyone_10 = 0
count_Everyone=0

count_Mature_17=0
count_Adults_only=0
count_Unrated=0


for x in app["Content Rating"]:
    if x== "Teen":
        count_Teen= count_Teen+1
    elif x== "Everyone 10+":
        count_Everyone_10= count_Everyone_10+1
    elif x== "Everyone":
        count_Everyone= count_Everyone+1
    elif x== "Mature 17+":
        count_Mature_17 = count_Mature_17+1
    elif x== "Adults only 18+":
        count_Adults_only= count_Adults_only+1
    elif x== "Unrated":
        count_Unrated= count_Unrated+1
print ("*************************************************************************
******************************")
print ("Number of apps of all the dataset, having the content rating which bel
ong the top apps:")
print ("*")
print ("*")
print ("Number of apps that belong to the content rating \"Teen\" is: {}".form
at(count_Teen))
print ("*************************************************************************
******************************")
print ("Number of apps that belong to the content rating \"Everyone 10+\" is:
{}".format(count_Everyone_10))
print ("*************************************************************************
******************************")
print ("Number of apps that belong to the content rating \"Everyone\" is: {}".
format(count_Everyone))
print ("*************************************************************************
******************************")
print ("*************************************************************************
******************************")
print ("#########################################################################
##############################")
print ("Number of apps having content rating not included in the top apps")
print ("*")
print ("*")
print ("Number of apps that belong to the content rating \"Mature 17+\" is: {}
```

```
".format(count_Mature_17))
print ("Number of apps that belong to the content rating \"Adults only 18+\" i
s: {}".format(count_Adults_only))
print ("Number of apps that belong to the content rating \"Unrated\" is: {}".f
ormat(count_Unrated))
```

```
*************************************************************************
************************
Number of apps of all the dataset, having the content rating which belong the
top apps:
*
*
Number of apps that belong to the content rating "Teen" is: 17263
*************************************************************************
************************
Number of apps that belong to the content rating "Everyone 10+" is: 4661
*************************************************************************
************************
Number of apps that belong to the content rating "Everyone" is: 241582
*************************************************************************
************************
*************************************************************************
************************
#########################################################################
#######################
Number of apps having content rating not included in the top apps
*
*
Number of apps that belong to the content rating "Mature 17+" is: 3489
Number of apps that belong to the content rating "Adults only 18+" is: 12
Number of apps that belong to the content rating "Unrated" is: 33
```

In [114]:

```
# The aforementioned can be found more easily with the below command

app["Content Rating"].value_counts(ascending=False)
```

Out[114]:

```
Everyone           241582
Teen                17263
Everyone 10+         4661
Mature 17+           3489
Unrated                33
Adults only 18+        12
Name: Content Rating, dtype: int64
```

In [115]:
```
# In this and in the next 2 commands, i will try to see if there is any correl
ation between installations, Rating and Reviews

top_10_installed_and_rated_apps
```

Out[115]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyone
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Teen
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyone
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyone
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.0	Everyone
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.0	Everyone
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.0	Everyone
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Teen
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyone 10+
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.0	Everyone

In [116]:
```
#*********************************************************************************
# It seems that none of the best rated apps belong to the top installed (filtered by rating too) apps
#*********************************************************************************
app.sort_values(by="Rating", ascending= False).head(10)
```

Out[116]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	
194506	F-pics	GAME_CASUAL	5.0	3.0	500	7.8	0.0	Everyone	A
193239	Smartler Smart Home	BUSINESS	5.0	2.0	100	19	0.0	Everyone	
108020	NHK World News Reader - Chinese version	NEWS_AND_MAGAZINES	5.0	3.0	100	2.6	0.0	Everyone	
193255	UNE Safe	EDUCATION	5.0	5.0	500	15	0.0	Everyone	
193254	VandySafe	EDUCATION	5.0	1.0	500	24	0.0	Everyone	A
193253	TigerSafe	EDUCATION	5.0	1.0	500	16	0.0	Everyone	D
193249	keypad lock screen 2019	TOOLS	5.0	3.0	100	9.8	0.0	Everyone	
38405	Corrigo	BUSINESS	5.0	5.0	5000	12	0.0	Everyone	D
38416	S►ivu	GAME_ACTION	5.0	1.0	50	65	0.0	Everyone	
193014	Vadii	ENTERTAINMENT	5.0	10.0	500	4.3	0.0	Everyone	Se

In [117]:
```
#*********************************************************************
**********************************

# Relationship between Reviews and main top apps

# It seems that Instagram, Clean Master - Antivirus, Youtube and Subway Surfer
s

# The above 4 apps belong to the top installed (filtered by rating too) and si
multaneously to the top reviewed apps

# So there is correlation of 4 out of 10 apps respecting top installed-rated a
pps and top reviewed apps
#*********************************************************************
**********************************

app.sort_values(by="Reviews", ascending= False).head(10)
```

Out[117]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
671	WhatsApp Messenger	COMMUNICATION	4.417610	86214292.0	1000000000	Varies with device	0.0	Everyone
704	Facebook	SOCIAL	4.087946	85766433.0	1000000000	Varies with device	0.0	Teen
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Teen
632	Messenger ♀ Text and Video Chat for Free	COMMUNICATION	4.085856	65469531.0	1000000000	Varies with device	0.0	Everyone
628	Clash of Clans	GAME_STRATEGY	4.606215	48401470.0	500000000	103	0.0	Everyone 10+
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyone
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Teen
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyone 10+
12638	Security Master - Antivirus, VPN, AppLock, Boo...	TOOLS	4.652842	25532160.0	500000000	Varies with device	0.0	Everyone
1542	Clash Royale	GAME_STRATEGY	4.545474	25449254.0	100000000	81	0.0	Everyone 10+

In [118]:
```
top_10_installed_and_rated_apps
```

Out[118]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyone
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Teen
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyone
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyone
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.0	Everyone
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.0	Everyone
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.0	Everyone
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Teen
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyone 10+
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.0	Everyone

In [119]:
```
# Prices of the apps

app["Price"].value_counts().sort_values(ascending=False).head(10)
```

Out[119]:
```
0.00    255434
0.99      2317
1.99      1552
2.99      1351
4.99       883
3.99       767
1.49       761
2.49       518
3.49       339
9.99       275
Name: Price, dtype: int64
```

In [120]:
```
app.Price.nunique()
```

Out[120]:
```
488
```

In [121]:
```
##########################################################################
##########################
```

In [122]:
```
##########################################################################
##########################
```

In [123]:
```
##########################################################################
##########################
```

Visualising Data

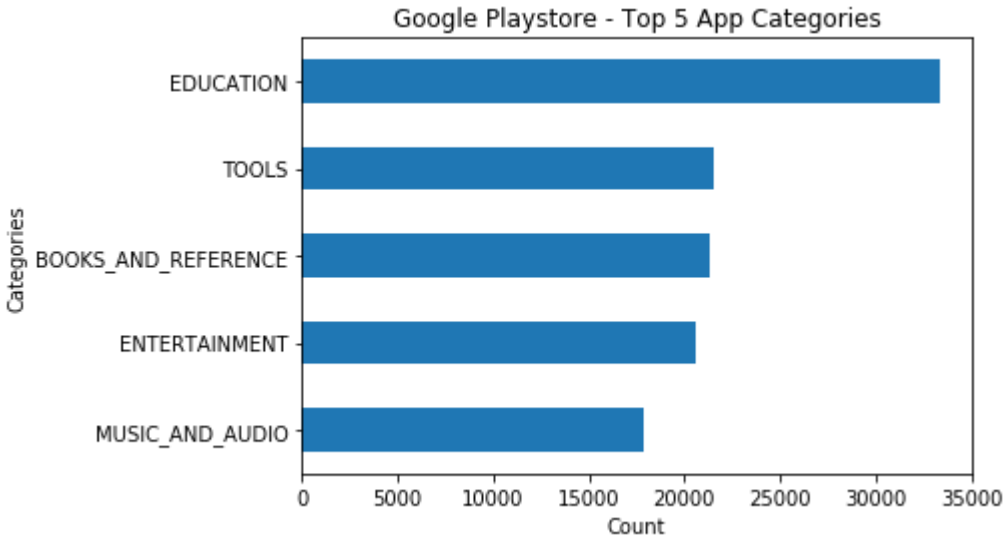
In [124]:
```
app.head(2)
```

Out[124]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating	U
0	DoorDash - Food Delivery	FOOD_AND_DRINK	4.548562	305034.0	5000000	Varies with device	0.0	Everyone	2
1	TripAdvisor Hotels Flights Restaurants Attract...	TRAVEL_AND_LOCAL	4.400671	1207922.0	100000000	Varies with device	0.0	Everyone	2

In [125]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [126]:
```
# Top 5 app Categories of all the dataset


app["Category"].value_counts().nlargest(5).sort_values(ascending=True).plot.ba
rh()
plt.ylabel("Categories")
plt.xlabel("Count")
plt.title("Google Playstore - Top 5 App Categories")
plt.show()
```



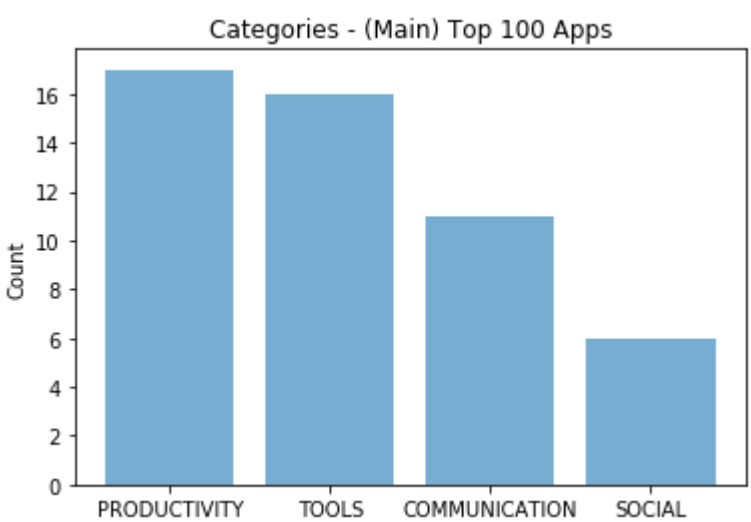
In [127]:
```
app["Category"].value_counts().nlargest(5).sort_values(ascending=False)
```

Out[127]:
```
EDUCATION              33394
TOOLS                  21591
BOOKS_AND_REFERENCE    21377
ENTERTAINMENT          20603
MUSIC_AND_AUDIO        17879
Name: Category, dtype: int64
```

In [128]:
```
# In which category do main 100 top apps belong

top_installed_and_rated_apps["Category"].head(100).value_counts()
```

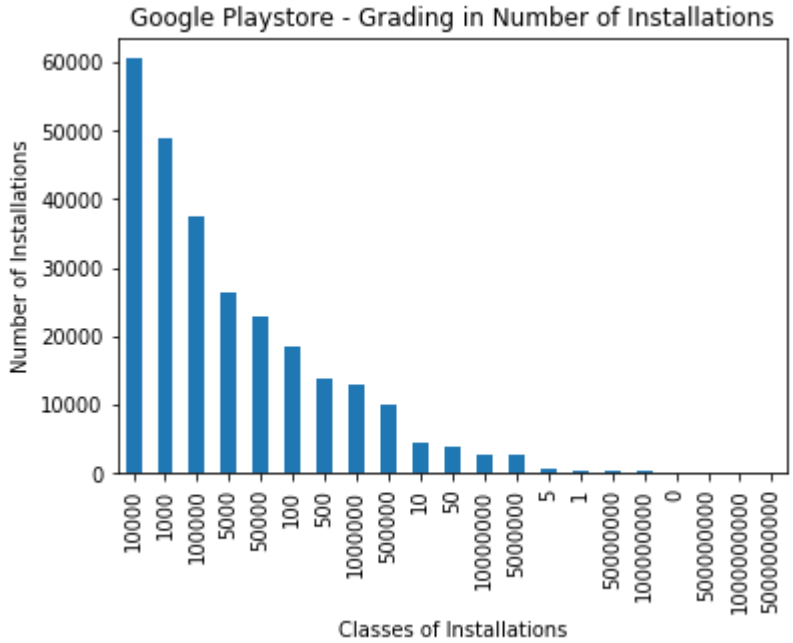
Out[128]:
```
PRODUCTIVITY           17
TOOLS                  16
COMMUNICATION          11
VIDEO_PLAYERS           6
SOCIAL                  6
PHOTOGRAPHY             4
GAME_CASUAL             4
GAME_ACTION             4
NEWS_AND_MAGAZINES      4
GAME_RACING             3
PERSONALIZATION         3
TRAVEL_AND_LOCAL        3
ENTERTAINMENT           3
BOOKS_AND_REFERENCE     2
SHOPPING                2
GAME_SPORTS             2
HEALTH_AND_FITNESS      2
GAME_ARCADE             2
MUSIC_AND_AUDIO         2
LIFESTYLE               1
EDUCATION               1
GAME_STRATEGY           1
GAME_SIMULATION         1
Name: Category, dtype: int64
```

In [129]:
```
status= ("PRODUCTIVITY", "TOOLS", "COMMUNICATION", "SOCIAL")
y_pos= np.arange(len(status))
numbers= [17,16,11,6]
plt.bar(y_pos, numbers, align="center", alpha=0.6)
plt.xticks(y_pos, status)
plt.ylabel("Count")
plt.title("Categories - (Main) Top 100 Apps")
plt.show()
```



In [130]:
```
x=top_installed_and_rated_apps.head(100)
```

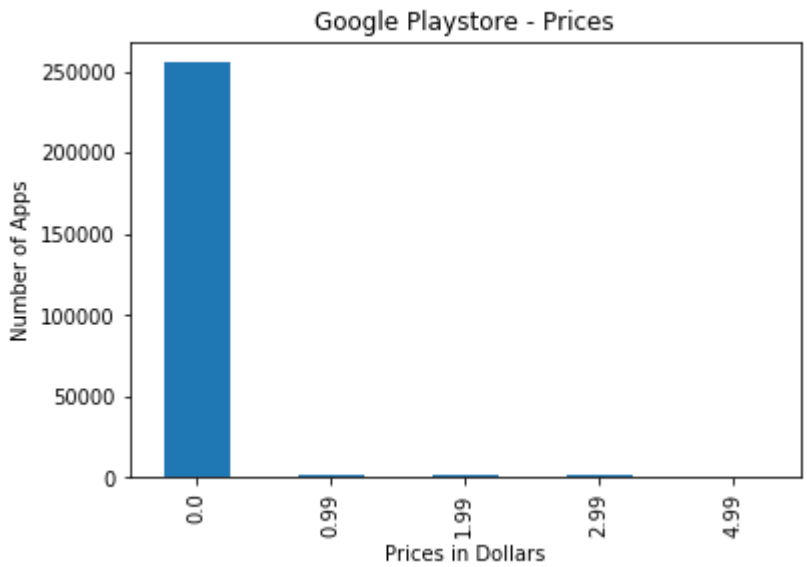
In [131]:
```
# Relationship betweeen: Classes and number of Installations


app["Installs"].value_counts().sort_values(ascending=False).plot.bar()
plt.ylabel("Number of Installations")
plt.xlabel("Classes of Installations")
plt.title("Google Playstore - Grading in Number of Installations")
plt.show()
```



In [132]:
```
# Top 5 Gradings in the number of installations

app["Installs"].value_counts().nlargest(5)
```

Out[132]:
```
10000     60533
1000      48884
100000    37499
5000      26361
50000     22794
Name: Installs, dtype: int64
```

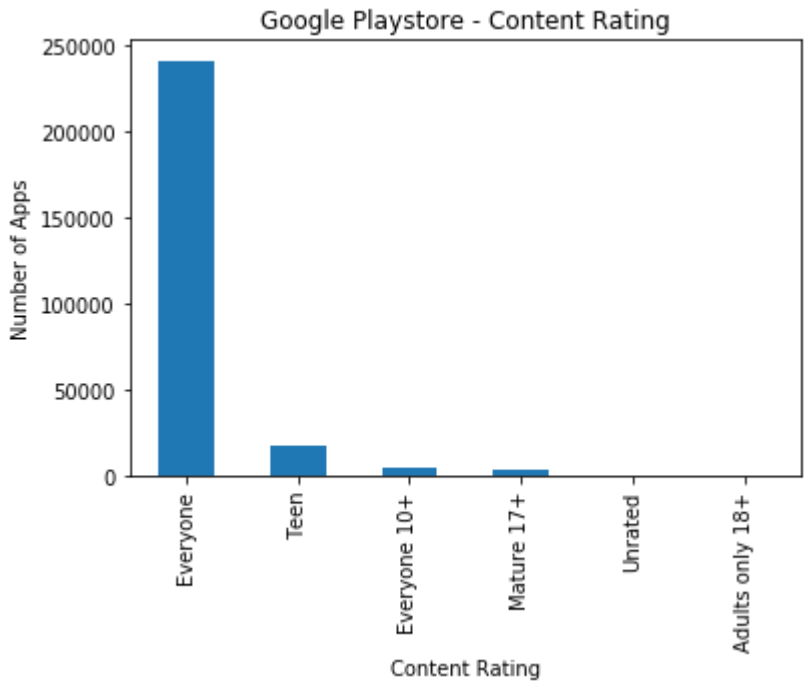
In [133]:
```
app["Price"].value_counts().nlargest(5).sort_values(ascending=False).plot.bar
()
plt.ylabel("Number of Apps")
plt.xlabel("Prices in Dollars")
plt.title("Google Playstore - Prices")
plt.show()
```



In [134]:
```
app["Price"].value_counts().nlargest(5)
```

Out[134]:
```
0.00    255434
0.99      2317
1.99      1552
2.99      1351
4.99       883
Name: Price, dtype: int64
```

In [135]:
```
app["Content Rating"].value_counts().sort_values(ascending=False).plot.bar()
plt.ylabel("Number of Apps")
plt.xlabel("Content Rating")
plt.title("Google Playstore - Content Rating")
plt.show()
```



In [136]:
```
app["Content Rating"].value_counts()
```

Out[136]:
```
Everyone           241582
Teen                17263
Everyone 10+         4661
Mature 17+           3489
Unrated                33
Adults only 18+        12
Name: Content Rating, dtype: int64
```

In [137]:
```
top_installed_and_rated_apps["Content Rating"].head(100).value_counts()
```

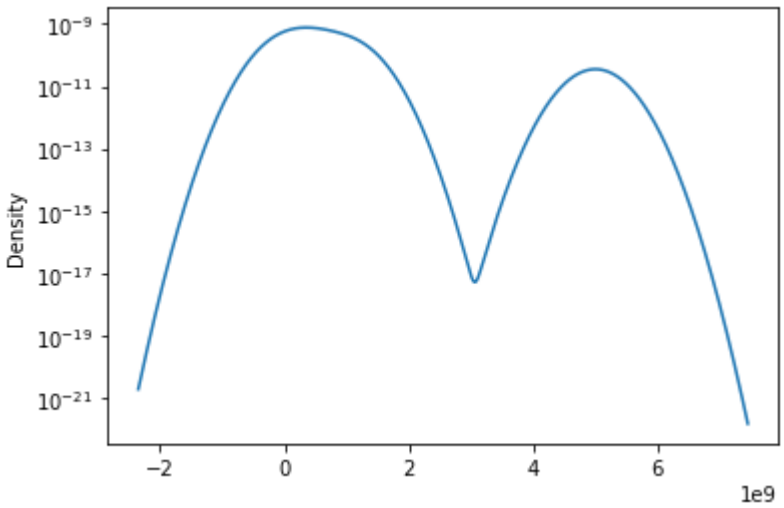
Out[137]:
```
Everyone        69
Teen            24
Everyone 10+     5
Mature 17+       2
Name: Content Rating, dtype: int64
```

In [138]:
```
#####################
```

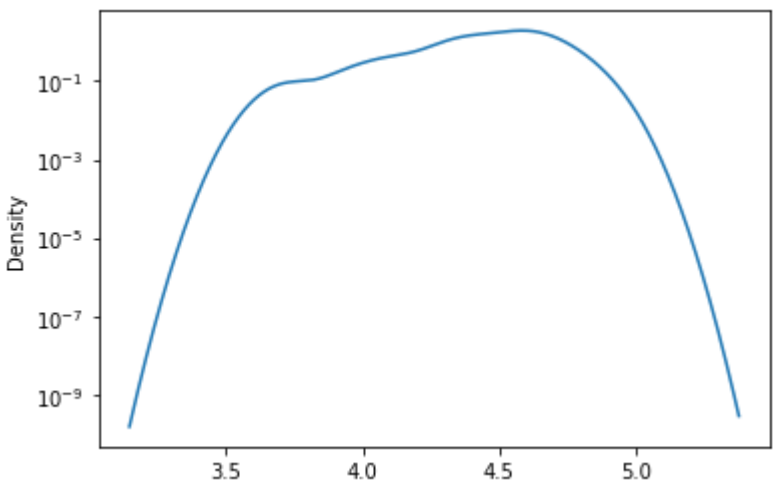
In [139]:
```
top_installed_and_rated_apps.head(100).Installs.value_counts(ascending=False)
```

Out[139]:
```
100000000     38
500000000     35
1000000000    24
5000000000     3
Name: Installs, dtype: int64
```

In [140]:
```
app_category= top_installed_and_rated_apps.head(100).Installs
app_category.plot.density().set_yscale("log")
```



In [141]:
```
app_category= top_installed_and_rated_apps.head(100).Rating
app_category.plot.density().set_yscale("log")
```



In [142]:
```
top_installed_and_rated_apps.head(100).Rating.value_counts(ascending=False)
```

Out[142]:
```
4.582568    1
4.703391    1
4.666019    1
4.657038    1
4.463741    1
4.619152    1
4.342798    1
4.691691    1
4.085856    1
4.598360    1
4.498131    1
4.583171    1
4.726500    1
4.040543    1
4.615935    1
4.569472    1
4.450339    1
4.448093    1
4.335172    1
4.817503    1
4.087946    1
4.579532    1
4.448671    1
3.978565    1
4.618747    1
4.077919    1
4.596871    1
4.450046    1
4.185223    1
4.487701    1
           ..
4.402619    1
4.615577    1
4.390956    1
4.684935    1
4.639062    1
4.820112    1
4.579340    1
4.607466    1
4.648891    1
4.311234    1
4.597058    1
4.745800    1
4.610724    1
4.486509    1
4.304268    1
4.335205    1
4.542380    1
3.755762    1
4.440125    1
4.519560    1
4.424015    1
4.215697    1
4.346980    1
4.569601    1
4.603675    1
4.351907    1
```

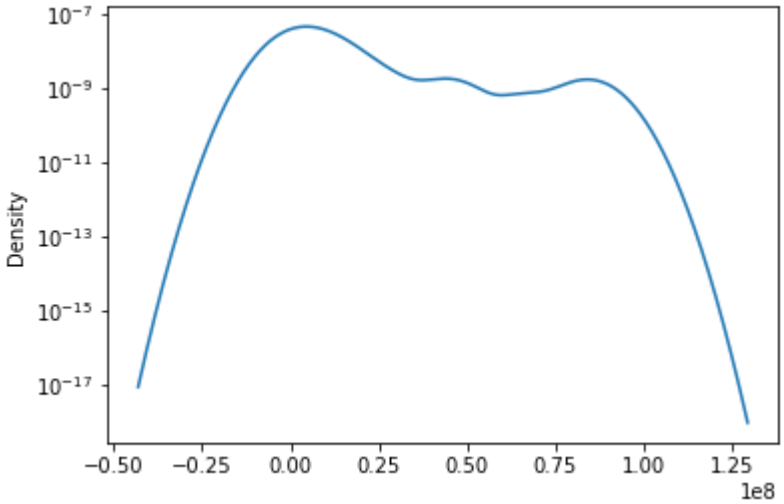
```
4.491666    1
4.355916    1
4.626627    1
4.652842    1
Name: Rating, Length: 100, dtype: int64
```

In [143]:
```
top_installed_and_rated_apps.head(100).Reviews.value_counts()
```

Out[143]:
```
1745403.0     1
19573637.0    1
1888392.0     1
2608408.0     1
966728.0      1
10855572.0    1
8321620.0     1
985165.0      1
38469.0       1
3878214.0     1
667452.0      1
10870728.0    1
11871704.0    1
9670607.0     1
600926.0      1
10450444.0    1
85766433.0    1
12723787.0    1
9123436.0     1
2048739.0     1
1691877.0     1
4344742.0     1
6872215.0     1
2551266.0     1
5614163.0     1
1058436.0     1
13636591.0    1
16278468.0    1
987137.0      1
7387192.0     1
             ..
1048972.0     1
29834812.0    1
2841568.0     1
10746013.0    1
8612822.0     1
14642083.0    1
12582.0       1
3110629.0     1
19026060.0    1
7375938.0     1
41919102.0    1
810890.0      1
4824180.0     1
3683909.0     1
3663432.0     1
16822895.0    1
7261352.0     1
323021.0      1
11504048.0    1
1718072.0     1
32037.0       1
7169292.0     1
2793126.0     1
3182256.0     1
631473.0      1
2920141.0     1
```

```
1976168.0     1
58506.0       1
99127.0       1
24657922.0    1
Name: Reviews, Length: 100, dtype: int64
```

In [144]:
```
app_category= top_installed_and_rated_apps.head(100).Reviews
app_category.plot.density().set_yscale("log")
```



In [145]:
```
####################
```

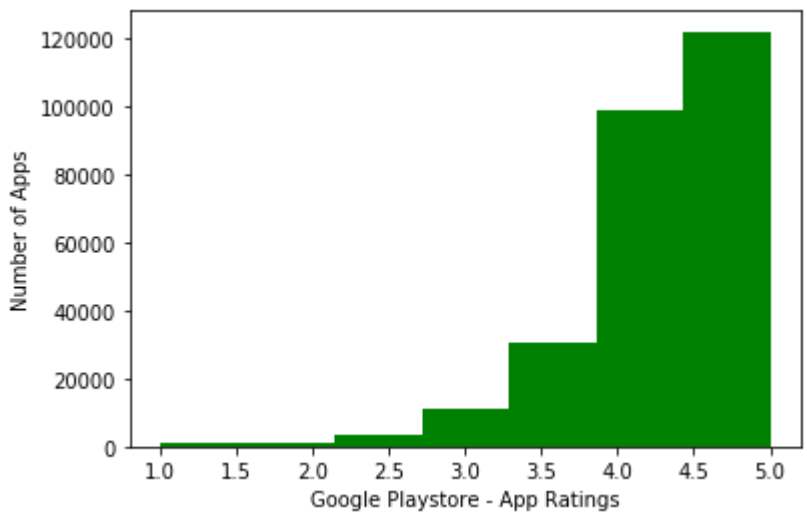
In [146]:
```
app["Rating"].value_counts()
```

Out[146]:
```
5.000000    23805
4.000000     5469
4.500000     3519
3.000000     2581
4.333333     2204
4.666667     2167
3.666667     1913
4.750000     1463
4.200000     1445
4.600000     1215
4.250000     1182
3.500000     1169
4.800000     1116
1.000000     1057
4.428571      958
4.400000      934
4.833333      786
4.714286      718
4.166667      716
3.750000      714
4.555555      692
4.571429      662
3.333333      606
4.375000      598
4.285714      590
3.800000      576
4.857143      572
4.142857      524
4.625000      515
4.444445      507
            ...
4.224391        1
4.456431        1
4.398297        1
4.688091        1
3.949901        1
4.534699        1
3.950251        1
2.906398        1
4.742160        1
4.808743        1
3.225166        1
4.500144        1
4.189717        1
4.640662        1
2.719557        1
4.737041        1
4.602818        1
4.630573        1
3.828959        1
3.825838        1
3.881509        1
4.600719        1
3.815268        1
4.646829        1
4.593723        1
4.664837        1
```

```
4.386788        1
4.727144        1
4.428835        1
3.945443        1
Name: Rating, Length: 99845, dtype: int64
```

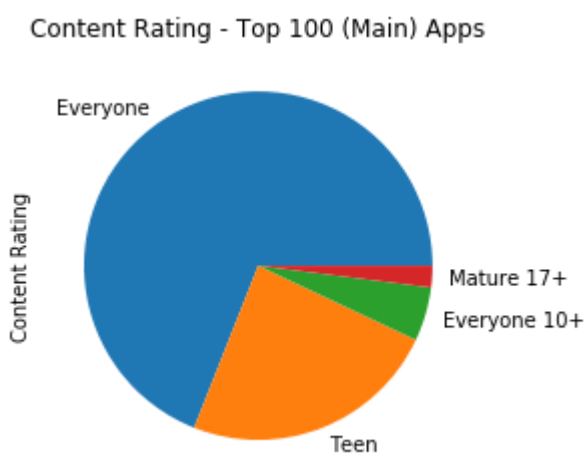
In [147]:
```
app_rating= app["Rating"]
num_bins=7
plt.hist(app_rating, num_bins, facecolor="green", alpha = 1)
plt.xlabel("Google Playstore - App Ratings")
plt.ylabel("Number of Apps")
plt.show
```

Out[147]:
```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [148]:
```
##############################
```

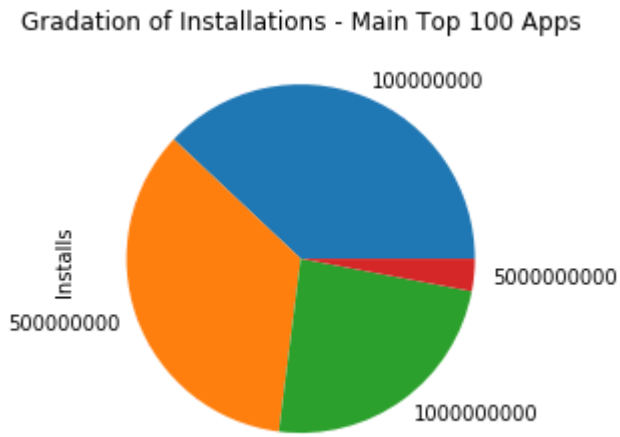
In [149]:
```
app1=top_installed_and_rated_apps.head(100)
app1["Content Rating"].value_counts().plot.pie()
plt.title("Content Rating - Top 100 (Main) Apps")
plt.show()
```



In [150]:
```
app1["Content Rating"].value_counts()
```

Out[150]:
```
Everyone        69
Teen            24
Everyone 10+     5
Mature 17+       2
Name: Content Rating, dtype: int64
```

In [151]:
```
app2= top_installed_and_rated_apps.head(100)
app2["Installs"].value_counts().plot.pie()
plt.title("Gradation of Installations - Main Top 100 Apps")
plt.show()
```



In [152]:
```
app2["Installs"].value_counts()
```

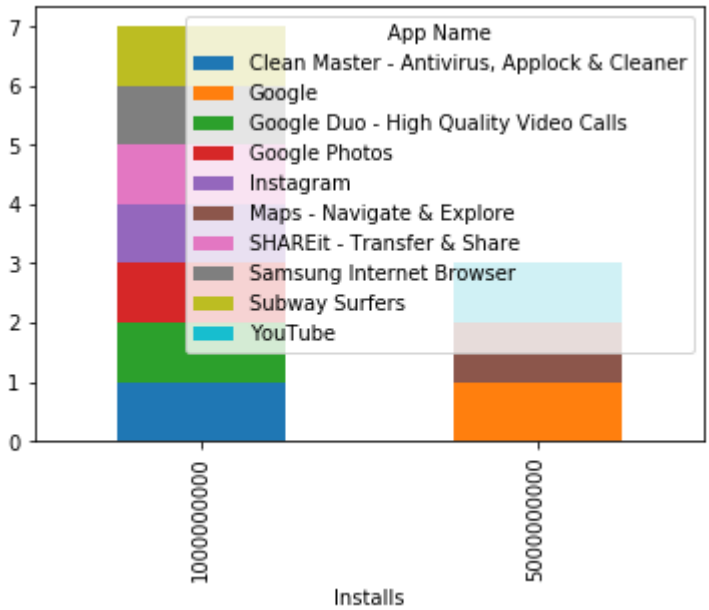
Out[152]:
```
100000000     38
500000000     35
1000000000    24
5000000000     3
Name: Installs, dtype: int64
```

In [153]:
```
top_10_installed_and_rated_apps
```

Out[153]:

	App Name	Category	Rating	Reviews	Installs	Size	Price	Content Rating
821	Google	TOOLS	4.408893	10870728.0	5000000000	Varies with device	0.0	Everyone
813	YouTube	VIDEO_PLAYERS	4.368428	41919102.0	5000000000	Varies with device	0.0	Teen
2177	Maps - Navigate & Explore	TRAVEL_AND_LOCAL	4.342798	10083666.0	5000000000	Varies with device	0.0	Everyone
7064	Clean Master - Antivirus, Applock & Cleaner	TOOLS	4.657038	44171776.0	1000000000	20	0.0	Everyone
6411	Google Duo - High Quality Video Calls	COMMUNICATION	4.596404	3641252.0	1000000000	20	0.0	Everyone
3269	SHAREit - Transfer & Share	TOOLS	4.579340	10450444.0	1000000000	20	0.0	Everyone
1259	Google Photos	PHOTOGRAPHY	4.542380	16278468.0	1000000000	Varies with device	0.0	Everyone
653	Instagram	SOCIAL	4.519560	79726403.0	1000000000	Varies with device	0.0	Teen
539	Subway Surfers	GAME_ARCADE	4.498131	29834812.0	1000000000	85	0.0	Everyone 10+
3267	Samsung Internet Browser	COMMUNICATION	4.424015	832714.0	1000000000	Varies with device	0.0	Everyone

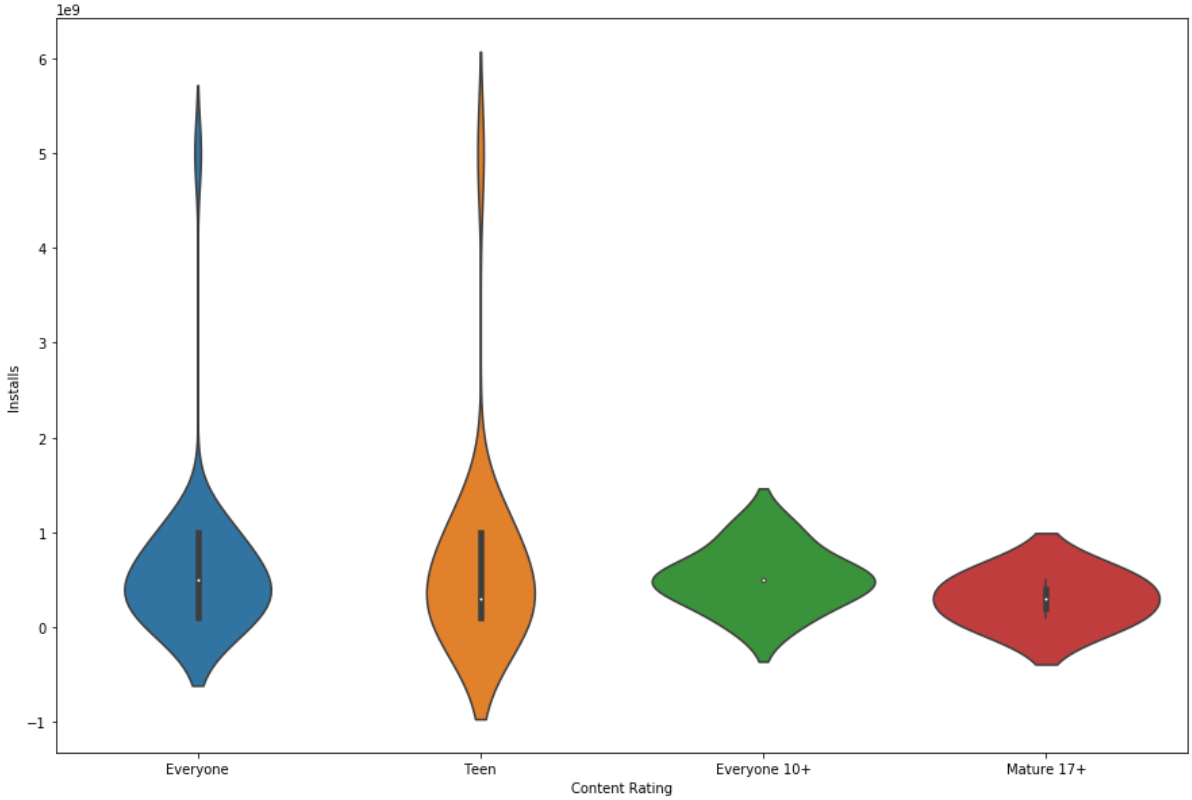
In [154]:
```
# top 10 main apps

app4= top_10_installed_and_rated_apps
top_apps=app4.groupby(["Installs", "App Name"]).size().unstack()
top_apps.plot(kind="bar", stacked=True)
ax=plt.gca()
plt.show()
```



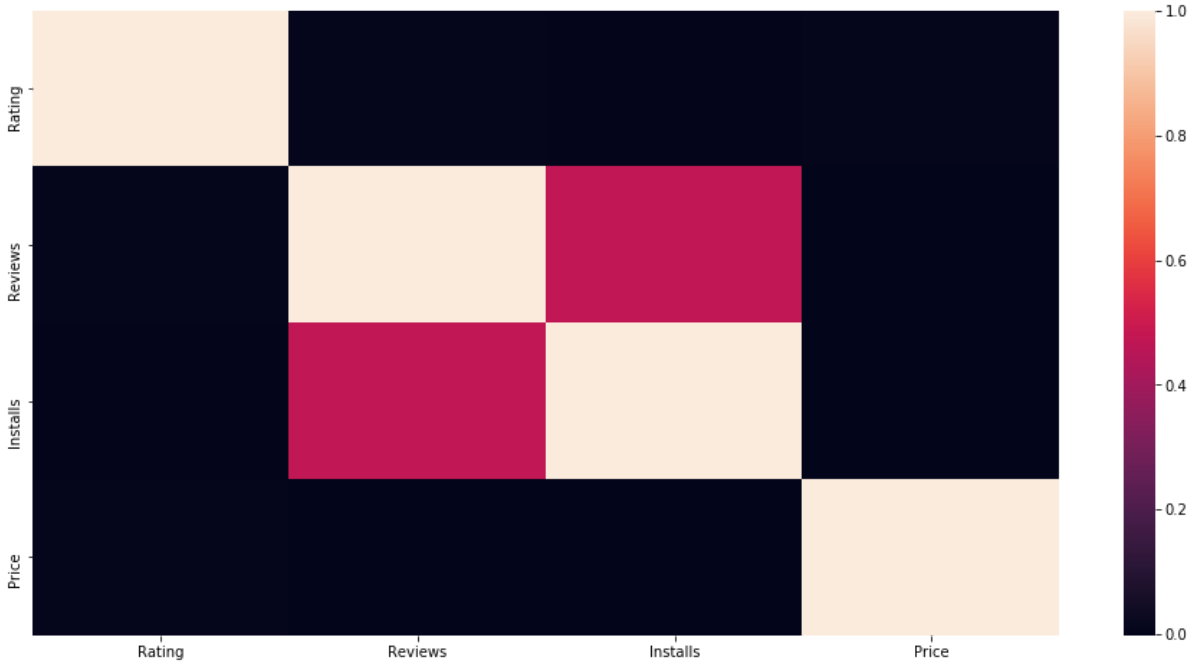
In [155]:
```
top_installed_and_rated_apps.head(100)["Content Rating"].value_counts(ascendin
g= False)
```

Out[155]:
```
Everyone        69
Teen            24
Everyone 10+     5
Mature 17+       2
Name: Content Rating, dtype: int64
```

In [156]:
```
# violin plot - main top 100 apps - content rating vs installs
app5=top_installed_and_rated_apps.head(100)
plt.figure(figsize=(15,10))
sns.violinplot(x= "Content Rating", y="Installs", data= app5)
plt.show()
```



In [157]:
```
plt.figure(figsize=(16,8))
corr= app.corr()
sns.heatmap(corr)
plt.show()
```



In [158]:
```
##########################################################################
##########################
```

In [159]:
```
##########################################################################
##########################
```

In [160]:
```
##########################################################################
##########################
```

Unsupervised Methods

In [161]:
```
new_app = app.head(1000) # taking a part(sample) from the data set to apply su
pervided and unsupervised
# i did not take a bigger sample because of memory crashes
```

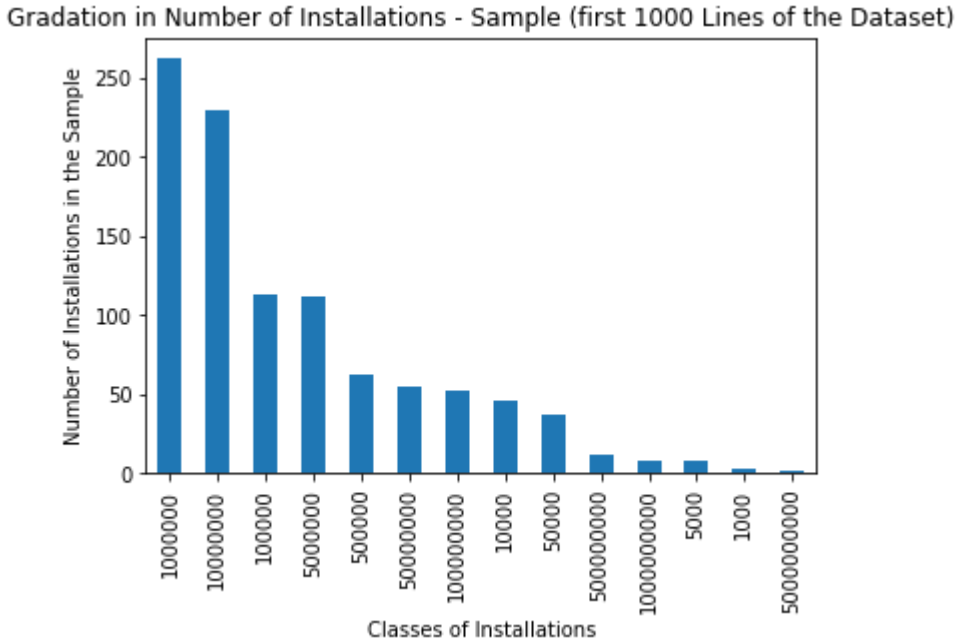
In [162]:
```
new_app.shape
```

Out[162]:
```
(1000, 11)
```

In [203]:
```
new_app["Installs"].value_counts().sort_values(ascending=False)
```

Out[203]:
```
1000000       262
10000000      229
100000        113
5000000       112
500000         62
50000000       54
100000000      52
10000          46
50000          37
500000000      12
1000000000      8
5000            8
1000            3
5000000000      2
Name: Installs, dtype: int64
```

In [165]:
```
# I want to see the gradation of the number of installations of the new datafr
ame(sample), so as to compare later
# so as to compare the unsupervised and supervised methods results

new_app["Installs"].value_counts().sort_values(ascending=False).plot.bar()
plt.ylabel("Number of Installations in the Sample")
plt.xlabel("Classes of Installations")
plt.title("Gradation in Number of Installations - Sample (first 1000 Lines of
 the Dataset)")
plt.show()
```



In [166]:
```
#*********************************
# Import packages from Scikit Learn

from sklearn import cluster # in unsupervised method we have clusters/ data ar
a grouped into clusters
from sklearn import metrics # for the distances between the data
from sklearn.preprocessing import scale # for scaling
from sklearn.preprocessing import LabelEncoder # for converting strings to flo
ats
from sklearn.preprocessing import OrdinalEncoder # for converting strings to f
loats when x(attributes) are strings
```

In [167]:
```
#*****************************************************************************
***

# Segmenting the data i have chosen into attributes (features)=x, and target=
(y)

# y will be the number of installations

# x will be: Category, Rating, Reviews and Content Rating


x= new_app[['Category', 'Rating', 'Reviews', 'Content Rating']] # attributes

y= new_app["Installs"] # y included the classes of installations. e.g. 100,000
in the dataset means more than 100,000 installations
```

In [168]:
```
# x has strings. This command is for converting strings to floats

x_transformed= OrdinalEncoder().fit_transform(x)
```

In [169]:
```
# Preparing the data- Scaling/ Handling the data in such way they can belong i
n a spesific range
# and represent the same degree of difference
#*****************************************************************************
****************************

scaled_data= scale(x_transformed)
```

In [170]:
```
scaled_data
```

Out[170]:
```
array([[-0.77036381,  1.06735713,  0.86717825, -0.57690942],
       [ 2.36819293,  0.28702897,  1.37371577, -0.57690942],
       [ 1.96321786, -1.66552549, -1.2428125 , -0.57690942],
       ...,
       [ 2.1657054 , -1.68980237,  0.0043178 , -0.57690942],
       [ 2.1657054 , -1.72448362,  0.30125358, -0.57690942],
       [ 2.1657054 ,  1.16099651, -0.64894093, -0.57690942]])
```

In [171]:
```
# import python libraries for creating clusters, for converting and for scalin
g

from sklearn import cluster
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import scale
```

In [172]:
```
# i have taken a sample, so now the clusters of installations are 14 from 21 t
hat normally are for the whole dataset
# creating  clusters using Agglomerative Clustering
len(np.unique(y))
```

Out[172]:
```
14
```

In [173]:
```
y.unique()
```

Out[173]:
```
array([   5000000,  100000000,     100000,   10000000,      10000,
          1000000,   50000000,     500000,      50000,       5000,
             1000,  500000000, 1000000000, 5000000000], dtype=int64)
```

In [174]:
```
#*************************************************************************
************************
# Hierarchical agglomerative clustering - bottom-up approach
#*************************************************************************
************************
#*************************************************************************
************************
#*************************************************************************
************************

# Using average in linkage means that i use the average of the distances of ea
ch observation
from sklearn.cluster import AgglomerativeClustering
n_samples, n_features = scaled_data.shape
n_digits = len(np.unique(y))
model = cluster.AgglomerativeClustering(n_clusters=n_digits, linkage="average"
, affinity="cosine")
model.fit(scaled_data)
# this is the model created
```

Out[174]:
```
AgglomerativeClustering(affinity='cosine', compute_full_tree='auto',
                        connectivity=None, distance_threshold=None,
                        linkage='average', memory=None, n_clusters=14,
                        pooling_func='deprecated')
```

In [175]:
```
print (model.labels_)
```

```
[ 0  8  4  5  9  6  3  7  4  7  7  6  4  6  0  0  6  7  7  6  7  5  4  4
 10  7  8  7  5  7  5  9  6  5  5  8  6  7  3  6  6  7  9  3  5  4  2  9
  1  9 10  9  6 10  2 10  6  9  1  6  9  1  6  3  9  9 10  1 10  9  1  7
  9  3  6  3  2  2 10  9  1  6  7  2  1  6  6  6  9  9  9 10  1  6  9  6
 10  6  1  6  6  6  1  6 10  7  3  7  6  6  7  6  5  5  6  7  9  6  7  7
  7  9  7  0  6  5  9  5  9  9  6  7  6  6  6  6  6  6  5  5  5  5  6  9
  6  6  6  7  6  7  5  6  7  6  6  6  3  7  6  6  5  6  7  0  8  7  7  6
  0  6  7  9  6  6  9  7  6  6  6  6  6  6  6  6  6  6  6  6  7  6  6  6
  6  9  9  6  6  9  7  6  4  6  9  6  6  6  6  6  6  6  6  6  6  6  9  5
  6  6  6  9  6  9  6  2  6  6  6  6  9  6  6  6  6  9  6  6  6  6  6  6
  6  9  6  6  0  6  6  6  4  6  6  6  6  9  9  7  5  5  3  3  3  3  7  6
  6  2 11  6  3  0  2  2  9  3  3  4  6  2  2  2  2  1  6  2  3  3  2  7
  6  2  2  9  3  5  3  2  1 13  3  0  5  7  9  5  3  3  2  6  2  2  3 11
  4  3  9  0  9 12  0  0 11  9  8  4  0  0  0  6  7  0  4 11  4 11 11  5
  7  5  5 12 11  5 11  0 11  6 12  6  8  5 12  9  6 12 11  9  8 11 11  6
  0 11  6  8  4  8  6  7 11  5  8  6  6  5  9  8  8  7  6  4  6  4  8  4
  6  8  4  2  7  7  5  8  9  6  6  9  6  5  6  7  8  8  7  6  7  5  5  4
  8  9  6  7  6  8  6  7  5  7  6  7  8  7  5 11  8  6  5  6  6  8  6  6
  5  8  6  6  9 11  7  6  8  8  6  4  9  6  7  8  4 11  2  1  8 11  9  4
  8  9  8  4  4  8  8  3 10  8 11  8  8  9  8  4  8  7  1  8  8  4  8  8
  4 10  4 11  8 10  7  8  4  8  1  1  5 10  2  8  4  4  8  8 11  8  8  4
  4 10  8  3  8  7  3  4 10  0  4  4  8 10  8  4  3  8  9  3  8  8  8  8
  7  4  4 11  8  8  4  9  0  4  8  0  2  3  5  5  5  3  6  7  6  2  6  2
  2  0  5  3  3  5  7  6  3  3  7  6  0  9  9  2  6  3  2  7  3  5  0  9
  6  0  8  2  9  6  5  2  5  3  0  1  6  5  3  6  9  3  5  5  5  3  3  2
  0  2  3  0  3  5  5  2  5  5  5  5  3  9  0  9  5  2  7  5  3  3  0  5
 10  8  0  0  0 12  8 10  5  3  0 10  3  0  3 11  8  0  3  7 11  4  3  6
  0 12  1  3  5  3  9  4  5  1  3  4  3  0  6  1  3 11  5  3 12  3  5  5
  1  9  5  3  3  3  3  8  0  3  5  3  6  5  5  6 11  8  0  3  7  0  3  7
  4  8  8 13  0  3  3  0  3  3  3  6  5  6  9  2  6  6  6  6  7  9 11  0
  6 12  9  4  3  0  3  0  3  0  3 11  3  0  3  3  0  3  0  5  0  3  0  0
  0  3  3  0  1  0  3  3  3  3  0  3  0  3  3  3  3  6  6  7  6  6  3  1
  3  9  9 12  7  7  5  6  6  6  6  6  9  2  2  5  6  5  6  6  3  3  6  0
  0  9  7  5  6  6  6  6  2  3  2  0  6  7  3  9  0  3 13  9  7  3  3  5
  0  0  8 12 12  8 11  7  8  3  8  8 11  7  0  3  5  3  2  8  9  6  1  3
  5  0  5  8  8  3  0  4  7  1 13  0  5  6 11  1  3  3  7  5  9  2  3  1
  0  0  1  2  1  3  1  7  6  5  0  4  3  3 11  9  7  7 11  3  5  3 12 11
  2  5  3  7  3  6  3  2  5  6  6  7  6 11  9 11  0  4  2  1  4  3  3  0
 12  3  0  3  3 12 11  0 11  0  6 11  0 11  3 11  0  3 12  0  6  0 11 11
  3  1 12 12  0 11  1 11 12  4 11 10  3  1  0 11 11  6 10  0 10  9 11 12
  1 11 11 12  8 10  1  0 12 10  3  0  0  8 11  3  4  4  8  8 11  6  8  4
  8  8  8  4  4  8  4  4  4  8  8  4  4  4  4  8]
```

In [176]:
```
# Silhouette score: comprares the similarity of an object to its own cluster w
ith that to other clusters

# models labels= models assigned to the model
#
#
print (metrics.silhouette_score(scaled_data,model.labels_))
print (metrics.completeness_score(y, model.labels_))
print (metrics.homogeneity_score(y, model.labels_))
```

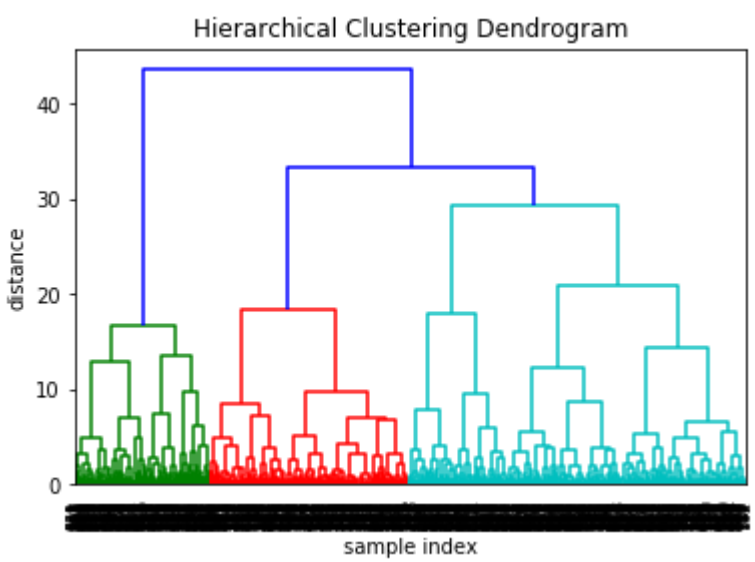
```
0.2046255768331342
0.18939644179827697
0.21934387424949217
```

In [177]:
```
len(np.unique(y))
```

Out[177]: 14

In [178]:
```
from scipy.cluster.hierarchy import dendrogram, linkage
```

In [179]:
```
# Creating Hierarchical Clustering Dendrogram

model= linkage(scaled_data, "ward")
plt.figure()
plt.title("Hierarchical Clustering Dendrogram")
plt.xlabel("sample index")
plt.ylabel("distance")
dendrogram(model, leaf_rotation=90.,  leaf_font_size=8.)
plt.show()
```



In [180]:
```
len(np.unique(y))
```

Out[180]: 14

In [181]:
```
#*************************************************************************
************************
#*************************************************************************
************************
#*************************************************************************
************************

# Clustering using K-means
# need for spesification of numbers of clusters
# clusters in this sample are 14
#*************************************************************************
************************
#*************************************************************************
************************
#*************************************************************************
************************
from sklearn import cluster
from sklearn.preprocessing import LabelEncoder
n_samples, n_features = scaled_data.shape
n_digits = len(np.unique(y))
for k in range(2, 15):
    kmeans = cluster.KMeans(n_clusters=k)
    kmeans.fit(scaled_data)
    print(k)
    print(metrics.silhouette_score(scaled_data, kmeans.labels_))
    print(metrics.completeness_score(y, kmeans.labels_))
    print(metrics.homogeneity_score(y, kmeans.labels_))

# different results on every iteration because we are using random starting po
ints# best score seems to be when k=13 (sometimes when k=14)
```

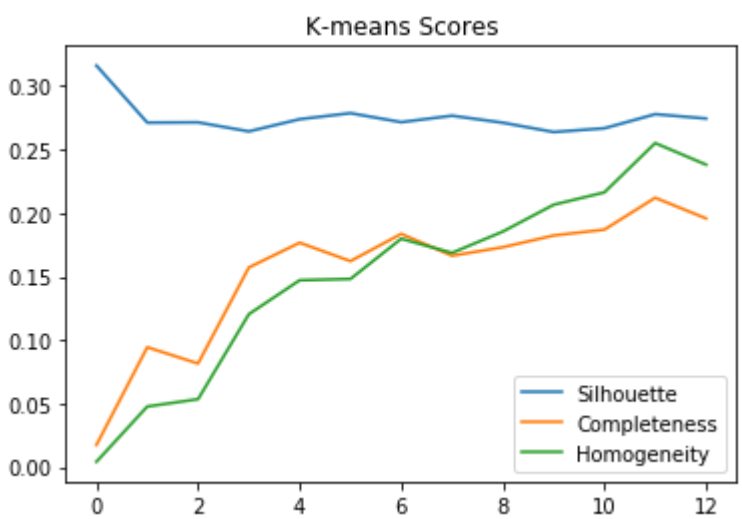
```
2
0.22845909778687695
0.29911438560884956
0.09925828981866817
3
0.27121160128133726
0.09437778328291654
0.0477102164329638
4
0.27143400175806187
0.08226163043801316
0.05396265706603794
5
0.264274824043344
0.1555394783970132
0.11921920342643202
6
0.27390111430441044
0.17666325277367087
0.1471235112545505
7
0.2792393333761522
0.16428121565009332
0.1500578556069895
8
0.2730649235118155
0.17846535898441626
0.17374298348673067
9
0.27794138852154326
0.1695399493758718
0.17130881655705082
10
0.2613079977808327
0.18505021301734736
0.2011817453653992
11
0.2673340279990994
0.18134498619945302
0.20450484736428085
12
0.2811291968308575
0.1734102001019015
0.1978783471902663
13
0.2752941210587467
0.20295881116624495
0.24405419929501107
14
0.2762046058714177
0.19071877853776847
0.2372126205396115
```

In [182]:
```
# same command with above, but now creating a list for every score in order to
show it to a graph

n_samples, n_features = scaled_data.shape
n_digits = len(np.unique(y))

y_silhouette=[]
y_completeness=[]
y_homogeneity=[]


for k in range(2, 15):
    kmeans = cluster.KMeans(n_clusters=k)
    kmeans.fit(scaled_data)
    print(k)
    print(metrics.silhouette_score(scaled_data, kmeans.labels_))
    y_silhouette.append(metrics.silhouette_score(scaled_data, kmeans.labels_))

    print(metrics.completeness_score(y, kmeans.labels_))
    y_completeness.append(metrics.completeness_score(y, kmeans.labels_))


    print(metrics.homogeneity_score(y, kmeans.labels_))
    y_homogeneity.append(metrics.homogeneity_score(y, kmeans.labels_))



print("*************************************************************************
*********************************************")
print("*************************************************************************
*********************************************")
print ("silhouette scores are:\n{}".format(y_silhouette))
print("*************************************************************************
*********************************************")
print ("completeness scores are:\n{}".format(y_completeness))
print("*************************************************************************
*********************************************")
print ("homogeneity scores are:\n{}".format(y_homogeneity))
print("*************************************************************************
*********************************************")
```

```
2
0.3160804206956437
0.017489560528020146
0.004283511685507623
3
0.27121160128133726
0.09437778328291656
0.04771021643296381
4
0.27141363588810574
0.08157230707935777
0.05349742834246446
5
0.26430074479828974
0.15719485789419713
0.12040635967892459
6
0.27390111430441044
0.17666325277367087
0.1471235112545505
7
0.2786596657082882
0.16218919541826798
0.14813925824154767
8
0.2715590805710025
0.1836315535432671
0.17980304161516805
9
0.2766470517662515
0.16643133968411433
0.16865678258206698
10
0.2711341946108843
0.17312850275761454
0.1854562614677368
11
0.2637955340027125
0.18235776113704238
0.20656157089162164
12
0.2667482419005359
0.18698278734387289
0.21630779808853376
13
0.27784068531653167
0.21208420344611517
0.2551057140385796
14
0.27443943071329113
0.19581207116598212
0.23813072195324067
*****************************************************************************
****************************************
*****************************************************************************
****************************************
silhouette scores are:
```

```
[0.3160804206956437, 0.27121160128133726, 0.27141363588810574, 0.264300744798
28974, 0.27390111430441044, 0.2786596657082882, 0.2715590805710025, 0.2766470
517662515, 0.2711341946108843, 0.2637955340027125, 0.2667482419005359, 0.2778
4068531653167, 0.27443943071329113]
*************************************************************************
***************************************
completeness scores are:
[0.017489560528020146, 0.09437778328291656, 0.08157230707935777, 0.1571948578
9419713, 0.17666325277367087, 0.16218919541826798, 0.1836315535432671, 0.1664
3133968411433, 0.17312850275761454, 0.18235776113704238, 0.18698278734387289,
0.21208420344611517, 0.19581207116598212]
*************************************************************************
***************************************
homogeneity scores are:
[0.004283511685507623, 0.04771021643296381, 0.05349742834246446, 0.1204063596
7892459, 0.1471235112545505, 0.14813925824154767, 0.17980304161516805, 0.1686
5678258206698, 0.1854562614677368, 0.20656157089162164, 0.21630779808853376,
0.2551057140385796, 0.23813072195324067]
*************************************************************************
***************************************
```

In [183]:
```
plt.plot(y_silhouette)
plt.plot(y_completeness)
plt.plot(y_homogeneity)
plt.legend(["Silhouette", "Completeness", "Homogeneity"])
plt.title("K-means Scores")
plt.show()
```



In [184]:
```
##########################################################################
##########################
```

In [185]:
```
##########################################################################
##########################
```

In [186]:
```
##########################################################################
##########################
```

Supervised Methods

In [187]:
```
new_app.shape
```

Out[187]:
```
(1000, 11)
```

In [188]:
```
supervised_app_x= new_app[['Category', 'Rating', 'Reviews', 'Content Rating']]
supervised_app_y= new_app["Installs"]
```

In [189]:
```
supervised_x=supervised_app_x.values # attributes
supervised_y= supervised_app_y.values #target
```

In [190]:
```
supervised_x_transformed= OrdinalEncoder().fit_transform(supervised_x) # conve
rsting the string values to floats for applying distance metrics
```

In [191]:
```
# segmenting the data in a training and test set of a 60/40 split
```

In [192]:
```
from sklearn.model_selection import train_test_split
```

In [193]:
```
supervised_x_transformed_train, supervised_x_transformed_test, supervised_y_tr
ain, supervised_y_test= train_test_split(supervised_x_transformed, supervised_
y, test_size=0.4)
```

In [202]:
```
# Creating classifiers to predict the class of installations, using:
# i. Logistic regression
# ii. KNN
```

In [195]:
```
print("LOGISTIC REGRESSION")
print("*************************************")
from sklearn.linear_model import LogisticRegression
lm = LogisticRegression()
lm.fit(supervised_x_transformed_train, supervised_y_train)
lm.predict_proba(supervised_x_transformed_test)
```

```
LOGISTIC REGRESSION
*************************************

C:\Users\Anaconda\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.p
y:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Spec
ify a solver to silence this warning.
  FutureWarning)
C:\Users\Anaconda\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.p
y:469: FutureWarning: Default multi_class will be changed to 'auto' in 0.22.
Specify the multi_class option to silence this warning.
  "this warning.", FutureWarning)
```

Out[195]:
```
array([[4.75617526e-09, 1.81259486e-04, 1.10789242e-03, ...,
        2.98458441e-02, 4.78654008e-03, 2.30747280e-03],
       [6.61140325e-11, 2.40601790e-05, 2.94108324e-04, ...,
        8.90648895e-02, 1.57510022e-02, 1.42627822e-03],
       [6.99311965e-15, 4.98101990e-08, 8.23104163e-06, ...,
        6.79104905e-02, 4.14856054e-03, 5.31386610e-03],
       ...,
       [9.39082494e-17, 2.98744039e-09, 3.72577133e-06, ...,
        1.12538906e-01, 9.36501448e-03, 3.59721025e-03],
       [7.04681434e-07, 7.37782228e-04, 7.53241461e-03, ...,
        1.45888381e-02, 9.52037024e-03, 9.85381049e-03],
       [1.44691128e-17, 4.97243573e-10, 4.91551428e-07, ...,
        1.34717809e-01, 2.36265051e-02, 3.20135727e-02]])
```

In [196]:
```
print(lm.intercept_)
```

```
[-0.66537122 -0.58352936 -0.96793526 -0.25628143 -0.87935984 -0.06730566
 -2.07175772 -3.06811722 -4.18899144 -5.49884143 -3.77476128 -2.94799348]
```

In [197]:
```
print(lm.coef_)
```

```
[[-3.75077123e-02  4.34167948e-04 -3.93743820e-02  2.57042897e-01]
 [ 7.22732620e-03  8.76354227e-04 -2.22162297e-02  4.53485082e-01]
 [ 1.72366584e-02  1.83434259e-03 -1.45889600e-02 -1.42371535e-01]
 [-1.87164237e-03  1.00361294e-03 -7.36022339e-03  4.37352265e-02]
 [-1.42288004e-02 -9.25512852e-05 -4.30297552e-03 -2.26375761e-02]
 [-1.05749621e-02  2.01203881e-04 -1.67336207e-03 -6.16138609e-02]
 [-1.86142401e-02 -1.43674939e-04  1.05959369e-03 -1.05780541e-01]
 [-2.00288858e-02 -8.03359231e-04  4.86202823e-03 -4.27783976e-02]
 [-3.49662353e-02 -1.55720751e-03  5.47227775e-03 -4.71970843e-01]
 [ 1.74118219e-02 -1.47321613e-03  5.30097904e-03 -1.61501934e-01]
 [ 1.82940993e-02 -3.09913919e-03  2.61717781e-03 -4.75906224e-01]
 [-9.10203545e-02 -3.67161479e-03  3.40949919e-03 -3.57945299e-01]]
```

In [198]:
```
predicted = lm.predict(supervised_x_transformed_test)
print(metrics.classification_report(supervised_y_test, predicted))
print(metrics.confusion_matrix(supervised_y_test, predicted))
```

```
              precision    recall  f1-score   support

        1000       0.00      0.00      0.00         3
        5000       0.00      0.00      0.00         3
       10000       0.30      0.15      0.20        20
       50000       0.00      0.00      0.00        14
      100000       0.37      0.21      0.27        48
      500000       0.00      0.00      0.00        26
     1000000       0.43      0.83      0.56       104
     5000000       0.00      0.00      0.00        47
    10000000       0.43      0.87      0.57        79
    50000000       0.00      0.00      0.00        24
   100000000       0.00      0.00      0.00        22
   500000000       0.00      0.00      0.00         5
  1000000000       0.00      0.00      0.00         3
  5000000000       0.00      0.00      0.00         2

    accuracy                           0.42       400
   macro avg       0.11      0.15      0.11       400
weighted avg       0.25      0.42      0.30       400

[[ 0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  6  0 11  0  0  0  0  0  0  0]
 [ 0  0  1  0  8  0  5  0  0  0  0  0  0  0]
 [ 0  0  3  0 10  0 35  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 26  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 86  0 18  0  0  0  0  0]
 [ 0  0  0  0  0  0 28  0 19  0  0  0  0  0]
 [ 0  0  0  0  0  0 10  0 69  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 24  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 22  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  2  0  0  0  0  0]]

C:\Users\Anaconda\Anaconda3\lib\site-packages\sklearn\metrics\classification.
py:1437: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

In [199]:
```
#K nearest neighbours

print("KNN")
print("*************************************")
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()
model.fit(supervised_x_transformed_train, supervised_y_train)
print(model)
```

```
KNN
*************************************
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [200]:
```
predicted= model.predict(supervised_x_transformed_test)
print (metrics.classification_report(supervised_y_test, predicted))
print (metrics.confusion_matrix(supervised_y_test, predicted))
```

```
C:\Users\Anaconda\Anaconda3\lib\site-packages\sklearn\metrics\classification.
py:1437: UndefinedMetricWarning: Precision and F-score are ill-defined and be
ing set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)

             precision    recall  f1-score   support

       1000       0.00      0.00      0.00         3
       5000       0.00      0.00      0.00         3
      10000       0.24      0.35      0.29        20
      50000       0.38      0.21      0.27        14
     100000       0.43      0.48      0.45        48
     500000       0.08      0.04      0.05        26
    1000000       0.56      0.73      0.64       104
    5000000       0.40      0.21      0.28        47
   10000000       0.57      0.70      0.63        79
   50000000       0.29      0.08      0.13        24
  100000000       0.46      0.55      0.50        22
  500000000       0.00      0.00      0.00         5
 1000000000       0.00      0.00      0.00         3
 5000000000       0.00      0.00      0.00         2

   accuracy                           0.47       400
  macro avg       0.24      0.24      0.23       400
weighted avg      0.43      0.47      0.44       400

[[ 0  0  3  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  1  1  0  0  0  0  0  0  0  0  0]
 [ 0  3  7  3  7  0  0  0  0  0  0  0  0  0]
 [ 0  2  7  3  2  0  0  0  0  0  0  0  0  0]
 [ 0  1 10  1 23  3 10  0  0  0  0  0  0  0]
 [ 0  1  1  0 12  1 11  0  0  0  0  0  0  0]
 [ 0  0  0  0  8  6 76  8  6  0  0  0  0  0]
 [ 0  0  0  0  1  1 24 10 11  0  0  0  0  0]
 [ 0  0  0  0  0  1 14  7 55  1  1  0  0  0]
 [ 0  0  0  0  0  0  0  0 16  2  6  0  0  0]
 [ 0  0  0  0  0  0  0  0  6  4 12  0  0  0]
 [ 0  0  0  0  0  0  0  0  2  0  2  0  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  2  0  0  0]]
```

In [201]:
```
print (metrics.accuracy_score(supervised_y_test, predicted))
```

```
0.4725
```

In [ ]: