In [1]:
```
%%HTML
<style type="text/css">
table.dataframe td, table.dataframe th {
    border: 1px  black solid !important;
  color: black !important;
}
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Design	March 26, 2017	1.0	2.3 and up
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19M	50,000+	Free	0	Everyone	Art & Design	April 26, 2018	1.1	4.0.3 and up
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	29M	1,000,000+	Free	0	Everyone	Art & Design	June 14, 2018	6.1.61.1	4.2 and up
8	Garden Coloring Book	ART_AND_DESIGN	4.4	13791	33M	1,000,000+	Free	0	Everyone	Art & Design	September 20, 2017	2.9.2	3.0 and up
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3.1M	10,000+	Free	0	Everyone	Art & Design;Creativity	July 3, 2018	2.8	4.0.3 and up

Find metadata about table like column name , its data type , number of records

In [8]:
```
appdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object 
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

Show basic stats for numerical column which is rating

In [9]:
```
appdata.describe()
```

Out[9]:

	Rating
count	9367.000000
mean	4.193338
std	0.537431
min	1.000000
25%	4.000000
50%	4.300000
75%	4.500000
max	19.000000

Do any column datatype conversion needed for optimization and better analysis?

In [11]:
```
#Make a note that column review have one value as a 3.0M which is not numeric
```

In [12]:
```
appdata["reviews"].astype('float')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3621, in Index.get_loc(self, key, method, tolerance)
   3620 try:
-> 3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:136, in pandas._libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:163, in pandas._libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'reviews'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [12], in <cell line: 1>()
----> 1 appdata["reviews"].astype('float')

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3505, in DataFrame.__getitem__(self, key)
   3503 if self.columns.nlevels > 1:
   3504     return self._getitem_multilevel(key)
-> 3505 indexer = self.columns.get_loc(key)
   3506 if is_integer(indexer):
   3507     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3623, in Index.get_loc(self, key, method, tolerance)
   3621     return self._engine.get_loc(casted_key)
   3622 except KeyError as err:
-> 3623     raise KeyError(key) from err
   3624 except TypeError:
   3625     # If we have a listlike key, _check_indexing_error will raise
   3626     #  InvalidIndexError. Otherwise we fall through and re-raise
   3627     #  the TypeError.
   3628     self._check_indexing_error(key)

KeyError: 'reviews'
```

In [14]:
```
appdata["Reviews"]=appdata["Reviews"].apply(lambda x : x.replace('M','')).astype('float')
```

In [15]:
```
appdata["Reviews"].astype('float')
```

Out[15]:
```
0           159.0
1           967.0
2         87510.0
3        215644.0
4           967.0
           ...
10836        38.0
10837         4.0
10838         3.0
10839       114.0
10840    398307.0
Name: Reviews, Length: 10841, dtype: float64
```

Now column values are corrected and we can see all are float

In [16]:
```
appdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10841 non-null  object 
 1   Category        10841 non-null  object 
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  float64
 4   Size            10841 non-null  object 
 5   Installs        10841 non-null  object 
 6   Type            10840 non-null  object 
 7   Price           10841 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10841 non-null  object 
 10  Last Updated    10841 non-null  object 
 11  Current Ver     10833 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(2), object(11)
memory usage: 1.1+ MB
```

Find out unique category values

In [17]:
```
appdata['Category'].unique()
```

Out[17]:
```
array(['ART_AND_DESIGN', 'AUTO_AND_VEHICLES', 'BEAUTY',
       'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS', 'COMMUNICATION',
       'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS', 'FINANCE',
       'FOOD_AND_DRINK', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME',
       'LIBRARIES_AND_DEMO', 'LIFESTYLE', 'GAME', 'FAMILY', 'MEDICAL',
       'SOCIAL', 'SHOPPING', 'PHOTOGRAPHY', 'SPORTS', 'TRAVEL_AND_LOCAL',
       'TOOLS', 'PERSONALIZATION', 'PRODUCTIVITY', 'PARENTING', 'WEATHER',
       'VIDEO_PLAYERS', 'NEWS_AND_MAGAZINES', 'MAPS_AND_NAVIGATION',
       '1.9'], dtype=object)
```

In [20]:
```
len(appdata['Category'].unique()) # it shows total 34 unique values
```

Out[20]:
```
34
```

In [22]:
```
appdata['Category'].nunique()
```

Out[22]:
```
34
```

What is average rating across all apps ?

In [24]:
```
appdata['Rating'].mean()
```

Out[24]:
```
4.193338315362448
```

What is average rating of only those app which comes under Photography category ?

In [25]:
```
appdata[appdata['Category']=='PHOTOGRAPHY']['Rating'].mean()
```

Out[25]:
```
4.192113564668767
```

How many are free and how many are paid apps ?

In [26]:
```
appdata['Type'].value_counts()
```

Out[26]:
```
Free    10039
Paid      800
0           1
Name: Type, dtype: int64
```

In [27]:
```
#So we have 10039 Free apps and one more ap with value 0 which is also free
```

Which app has max reviews?

In [40]:
```
appdata.head(2)
```

Out[40]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

In [44]:
```
maxreviewvalue=appdata['Reviews'].max()
```

In [45]:
```
appdata[appdata['Reviews']==maxreviewvalue]
```

Out[45]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2544	Facebook	SOCIAL	4.1	78158306.0	Varies with device	1,000,000,000+	Free	0	Teen	Social	August 3, 2018	Varies with device	Varies with device

Get App names and review columns where review is greater than 60000000

In [67]:
```
appdata1=appdata[appdata['Reviews']>60000000  ]
```

In [68]:
```
# get only app name and review for above
appdata2=appdata1[["App", "Reviews"]]
```

In [69]:
```
appdata2.sort_values
```

Out[69]:
```
<bound method DataFrame.sort_values of                 App     Reviews
336   WhatsApp Messenger  69119316.0
381   WhatsApp Messenger  69119316.0
2544           Facebook  78158306.0
2545          Instagram  66577313.0
2604          Instagram  66577446.0
2611          Instagram  66577313.0
3904  WhatsApp Messenger  69109672.0
3909          Instagram  66509917.0
3943           Facebook  78128208.0>
```

Select the top 5 app with max reviews

In [73]:
```
top5eviews = list(appdata["Reviews"].sort_values(ascending=False).head(5).index)
```

In [74]:
```
top5eviews
```

Out[74]:
```
[2544, 3943, 381, 336, 3904]
```

In [76]:
```
appdata.iloc[top5eviews]
```

Out[76]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
2544	Facebook	SOCIAL	4.1	78158306.0	Varies with device	1,000,000,000+	Free	0	Teen	Social	August 3, 2018	Varies with device	Varies with device
3943	Facebook	SOCIAL	4.1	78128208.0	Varies with device	1,000,000,000+	Free	0	Teen	Social	August 3, 2018	Varies with device	Varies with device
381	WhatsApp Messenger	COMMUNICATION	4.4	69119316.0	Varies with device	1,000,000,000+	Free	0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device
336	WhatsApp Messenger	COMMUNICATION	4.4	69119316.0	Varies with device	1,000,000,000+	Free	0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device
3904	WhatsApp Messenger	COMMUNICATION	4.4	69109672.0	Varies with device	1,000,000,000+	Free	0	Everyone	Communication	August 3, 2018	Varies with device	Varies with device

Select top 5 apps which has maximum installs

In [78]:
```
appdata["Installs"].astype('float')  # note that Installs column need a correction and transformation
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [78], in <cell line: 1>()
----> 1 appdata["Installs"].astype('float')

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:5912, in NDFrame.astype(self, dtype, copy, errors)
   5905     results = [
   5906         self.iloc[:, i].astype(dtype, copy=copy)
   5907         for i in range(len(self.columns))
   5908     ]
   5910 else:
   5911     # else, only a single dtype is given
-> 5912     new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   5913     return self._constructor(new_data).__finalize__(self, method="astype")
   5915 # GH 33113: handle empty frame or series

File ~\anaconda3\lib\site-packages\pandas\core\internals\managers.py:419, in BaseBlockManager.astype(self, dtype, copy, errors)
    418 def astype(self: T, dtype, copy: bool = False, errors: str = "raise") -> T:
--> 419     return self.apply("astype", dtype=dtype, copy=copy, errors=errors)

File ~\anaconda3\lib\site-packages\pandas\core\internals\managers.py:304, in BaseBlockManager.apply(self, f, align_keys, ignore_failures, **kwargs)
    302         applied = b.apply(f, **kwargs)
    303     else:
--> 304         applied = getattr(b, f)(**kwargs)
    305 except (TypeError, NotImplementedError):
    306     if not ignore_failures:

File ~\anaconda3\lib\site-packages\pandas\core\internals\blocks.py:580, in Block.astype(self, dtype, copy, errors)
    562 """
    563 Coerce to the new dtype.
    564
   (...)
    576 Block
    577 """
    578 values = self.values
--> 580 new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    582 new_values = maybe_coerce_values(new_values)
    583 newb = self.make_block(new_values)

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py:1292, in astype_array_safe(values, dtype, copy, errors)
   1289     dtype = dtype.numpy_dtype
   1291 try:
-> 1292     new_values = astype_array(values, dtype, copy=copy)
   1293 except (ValueError, TypeError):
   1294     # e.g. astype_nansafe can fail on object-dtype of strings
```

```
   1295     #  trying to convert to float
   1296     if errors == "ignore":

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py:1237, in astype_array(values, dtype, copy)
   1234     values = values.astype(dtype, copy=copy)
   1236 else:
-> 1237     values = astype_nansafe(values, dtype, copy=copy)
   1239 # in pandas we don't store numpy str dtypes, so convert to object
   1240 if isinstance(dtype, np.dtype) and issubclass(values.dtype.type, str):

File ~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py:1181, in astype_nansafe(arr, dtype, copy, skipna)
   1177     raise ValueError(msg)
   1179 if copy or is_object_dtype(arr.dtype) or is_object_dtype(dtype):
   1180     # Explicit copy, or required since NumPy can't view from / to object.
-> 1181     return arr.astype(dtype, copy=True)
   1183 return arr.astype(dtype, copy=copy)

ValueError: could not convert string to float: '10,000+'
```

In [79]:
```
## Lets drop the rows with error values
```

In [83]:
```
appdata[appdata['Installs'] == 'Free'] # This is not numeric value
```

Out[83]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	...	500,000,000+	50+	100+	500+	10+	1+	5+	0+	0	Free
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0	1,000+	Free	0	Everyone	NaN	February 11, 2018	...	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free

1 rows × 35 columns

In [84]:
```
appdata.drop(labels=10472,  axis=0, inplace=True)  # drop the row
```

In [86]:
```
appdata[appdata['Installs'] == 'Free']  ## confirm row is dropped
```

Out[86]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	...	500,000,000+	50+	100+	500+	10+	1+	5+	0+	0	Free

0 rows × 35 columns

In [88]:
```
 appdata.head(3) ## note that Install column have numberic values with + postfix, lets remove it
```

Out[88]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	...	500,000,000+	50+	100+	500+	10+	1+	5+	0+	0	Free
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159.0	19M	10,000+	Free	0	Everyone	Art & Design	...	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free
1	Coloring book moana	ART_AND_DESIGN	3.9	967.0	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	...	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510.0	8.7M	5,000,000+	Free	0	Everyone	Art & Design	...	Free	Free	Free	Free	Free	Free	Free	Free	Free	Free

3 rows × 35 columns

In [89]:
```
appdata['Installs']=appdata['Installs'].apply(lambda x: x.replace('+','').replace(',', '')).astype('float')
```

In [90]:
```
appdata.info() ## So column Installs is corrected now
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10840 entries, 0 to 10840
Data columns (total 35 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  float64
 4   Size            10840 non-null  object 
 5   Installs        10840 non-null  float64
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
 13  10,000+         10840 non-null  object 
 14  500,000+        10840 non-null  object 
 15  5,000,000+      10840 non-null  object 
 16  50,000,000+     10840 non-null  object 
 17  100,000+        10840 non-null  object 
 18  50,000+         10840 non-null  object 
 19  1,000,000+      10840 non-null  object 
 20  10,000,000+     10840 non-null  object 
 21  5,000+          10840 non-null  object 
 22  100,000,000+    10840 non-null  object 
 23  1,000,000,000+  10840 non-null  object 
 24  1,000+          10840 non-null  object 
 25  500,000,000+    10840 non-null  object 
 26  50+             10840 non-null  object 
 27  100+            10840 non-null  object 
 28  500+            10840 non-null  object 
 29  10+             10840 non-null  object 
 30  1+              10840 non-null  object 
 31  5+              10840 non-null  object 
 32  0+              10840 non-null  object 
 33  0               10840 non-null  object 
 34  Free            10840 non-null  object 
dtypes: float64(3), object(32)
memory usage: 3.0+ MB
```

In [91]:
```
top5install=list(appdata['Installs'].sort_values(ascending=False).head(5).index)
```

In [92]:
```
top5install
```

Out[92]:
```
[3896, 3943, 335, 3523, 3565]
```

In [95]:
```
appdata1=appdata.iloc[top5install]
```

In [96]:
```
appdata1[["App","Installs"]]
```

Out[96]:

	App	Installs
3896	Subway Surfers	1.000000e+09
3943	Facebook	1.000000e+09
335	Messenger – Text and Video Chat for Free	1.000000e+09
3523	Google Drive	1.000000e+09
3565	Google Drive	1.000000e+09

In [ ]: