

```
In [1]: 1 import pandas as pd
        2 import matplotlib.pyplot as plt

In [3]: 1 # Read the datasets into pandas dataframe objects
        2 android_df = pd.read_csv("googleplaystore.csv")

In [4]: 1 # Explore the data using pandas methods
        2 android_df
```

Out[4]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	A
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Des
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	A
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	A
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Desig
...
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	

10841 rows × 13 columns

```
In [5]: 1 android_df["Category"].value_counts()
```

```
Out[5]: Category
FAMILY          1972
GAME            1144
TOOLS           843
MEDICAL         463
BUSINESS        460
PRODUCTIVITY    424
PERSONALIZATION 392
COMMUNICATION   387
SPORTS          384
LIFESTYLE       382
FINANCE         366
HEALTH_AND_FITNESS 341
PHOTOGRAPHY     335
SOCIAL          295
NEWS_AND_MAGAZINES 283
SHOPPING        260
TRAVEL_AND_LOCAL 258
DATING          234
BOOKS_AND_REFERENCE 231
VIDEO_PLAYERS   175
EDUCATION       156
ENTERTAINMENT   149
MAPS_AND_NAVIGATION 137
FOOD_AND_DRINK  127
HOUSE_AND_HOME  88
LIBRARIES_AND_DEMO 85
AUTO_AND_VEHICLES 85
WEATHER         82
ART_AND_DESIGN  65
EVENTS          64
PARENTING        60
COMICS           60
BEAUTY           53
1.9              1
Name: count, dtype: int64
```

```
In [6]: 1 android_df[android_df["Category"] == "1.9"]
```

```
Out[6]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Cu
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN	February 11, 2018	1.0.19	4.

```
In [7]: 1 android_df[android_df["Category"] == "1.9"].values
```

```
Out[7]: array([[ 'Life Made WI-Fi Touchscreen Photo Frame', '1.9', 19.0, '3.0M',
        '1,000+', 'Free', '0', 'Everyone', nan, 'February 11, 2018',
        '1.0.19', '4.0 and up', nan]], dtype=object)
```

```
In [8]: 1 clean_lst = ['Life Made WI-Fi Touchscreen Photo Frame', "LIFESTYLE", '1.9', 19.0, '3.0M',
2             '1,000+', 'Free', '0', 'Everyone',"LIFESTYLE", 'February 11, 2018',
3             '1.0.19', '4.0 and up'],]
4 clean_lst
```

```
Out[8]: ['Life Made WI-Fi Touchscreen Photo Frame',
'LIFESTYLE',
'1.9',
19.0,
'3.0M',
'1,000+',
'Free',
'0',
'Everyone',
'LIFESTYLE',
'February 11, 2018',
'1.0.19',
'4.0 and up']
```

```
In [9]: 1 android_df[android_df["Category"] == "1.9"] = clean_lst
```

```
In [10]: 1 android_category = android_df["Category"].value_counts()
2 android_category
```

```
Out[10]: Category
FAMILY                1972
GAME                  1144
TOOLS                  843
MEDICAL                463
BUSINESS               460
PRODUCTIVITY           424
PERSONALIZATION        392
COMMUNICATION           387
SPORTS                  384
LIFESTYLE               383
FINANCE                 366
HEALTH_AND_FITNESS     341
PHOTOGRAPHY            335
SOCIAL                  295
NEWS_AND_MAGAZINES     283
SHOPPING                260
TRAVEL_AND_LOCAL       258
DATING                  234
BOOKS_AND_REFERENCE    231
VIDEO_PLAYERS          175
EDUCATION               156
ENTERTAINMENT          149
MAPS_AND_NAVIGATION    137
FOOD_AND_DRINK         127
HOUSE_AND_HOME          88
AUTO_AND_VEHICLES       85
LIBRARIES_AND_DEMO      85
WEATHER                  82
ART_AND_DESIGN          65
EVENTS                  64
PARENTING               60
COMICS                  60
BEAUTY                   53
Name: count, dtype: int64
```

```
In [11]: 1 app_count = android_df["App"].value_counts()
        2 app_count
```

```
Out[11]: App
ROBLOX                                           9
CBS Sports App - Scores, News, Stats & Watch Live 8
ESPN                                             7
Duolingo: Learn Languages Free                  7
Candy Crush Saga                               7
..
Meet U - Get Friends for Snapchat, Kik & Instagram 1
U-Report                                         1
U of I Community Credit Union                   1
Waiting For U Launcher Theme                    1
iHoroscope - 2018 Daily Horoscope & Astrology    1
Name: count, Length: 9660, dtype: int64
```

```
In [12]: 1 "Instagram" in app_count[app_count > 1].index
```

```
Out[12]: True
```

```
In [13]: 1 android_df[android_df["App"] == "Instagram"]
```

```
Out[13]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Cu
2545	Instagram	SOCIAL	4.5	66577313	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
2604	Instagram	SOCIAL	4.5	66577446	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
2611	Instagram	SOCIAL	4.5	66577313	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
3909	Instagram	SOCIAL	4.5	66509917	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V

```
In [14]: 1 # Check for duplicate row based on the "App" column, marking all duplicates as True
        2 duplicate_apps_df = android_df[android_df.duplicated(subset=['App'], keep=False)]
        3 duplicate_apps_df[duplicate_apps_df['App'] == "Instagram"]
```

```
Out[14]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Cu
2545	Instagram	SOCIAL	4.5	66577313	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
2604	Instagram	SOCIAL	4.5	66577446	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
2611	Instagram	SOCIAL	4.5	66577313	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V
3909	Instagram	SOCIAL	4.5	66509917	Varies with device	1,000,000,000+	Free	0	Teen	Social	July 31, 2018	V

```
In [15]: 1 # number of duplicate apps
2 num_duplicate_apps = duplicate_apps_df['App'].nunique()
3 num_duplicate_apps
```

Out[15]: 798

```
In [16]: 1 duplicate_apps_df.shape
```

Out[16]: (1979, 13)

```
In [17]: 1 android_df.shape[0]
```

Out[17]: 10841

```
In [18]: 1 10841 - 1181
```

Out[18]: 9660

```
In [19]: 1 # Group by app and get the maximum number of reviews for each app
2 reviews_max = android_df.groupby("App")['Reviews'].max()
```

```
In [20]: 1 reviews_max["Instagram"]
```

Out[20]: '66577446'

```
In [21]: 1 reviews_max
```

Out[21]: App

"i DT" Fútbol. Todos Somos Técnicos.	27	
+Download 4 Instagram Twitter	40467	
- Free Comics - Comic Apps	115	
.R	259	
/u/app	573	
...		
뽕티비 - 개인방송, 인터넷방송, BJ방송		414
💎 I'm rich	718	
❤️ WhatsLov: Smileys of love, stickers and GIF	22098	
📏 Smart Ruler ⇄ cm/inch measuring for homework!	19	
🔥 Football Wallpapers 4K Full HD Backgrounds 🤩	11661	

Name: Reviews, Length: 9660, dtype: object

```
In [22]: 1 # Create an empty List to store cleaned data
2 android_clean = []
3 # Create an empty List to keep track already added apps
4 already_added = []
5
6 # Iterate through each row in the DataFrame
7 for index, row in android_df.iterrows():
8     name = row['App']
9     n_reviews = row['Reviews']
10
11     # check if the current app has the maximum number of reviews and has not been added
12     if (reviews_max[name] == n_reviews) and (name not in already_added):
13         android_clean.append(row)
14         already_added.append(name)
```

```
In [23]: 1 android_clean = pd.DataFrame(android_clean)
```

```
In [24]: 1 android_clean.shape
```

```
Out[24]: (9660, 13)
```

```
In [25]: 1 def is_english(app_name):
2         lst = []
3         for i in app_name:
4             if ord(i) > 127:
5                 lst.append(False)
6             else:
7                 lst.append(True)
8
9         non_ascii = 0
10        for j in lst:
11            if j == False:
12                non_ascii += 1
13
14        if non_ascii > 3:
15            return False
16        else:
17            return True
```

```
In [26]: 1 is_english("English Jokes")
```

```
Out[26]: True
```

```
In [27]: 1 android_clean["App"].apply(is_english)
```

```
Out[27]: 0      True
1      True
2      True
3      True
4      True
5      True
...
10836  True
10837  True
10838  True
10839  True
10840  True
Name: App, Length: 9660, dtype: bool
```

```
In [28]: 1 android_english =android_clean[android_clean["App"].apply(is_english)]
```

In [29]:

1 android_english.head()

Out[29]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	U
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	`
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	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design; Creativity	J
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Design	2

In [30]:

1 android_english.shape

Out[30]: (9615, 13)

In [31]:

1 android_english["Price"].unique()

Out[31]: array(['0', '\$4.99', '\$3.99', '\$6.99', '\$1.49', '\$2.99', '\$7.99', '\$5.99', '\$3.49', '\$1.99', '\$9.99', '\$7.49', '\$0.99', '\$9.00', '\$5.49', '\$10.00', '\$11.99', '\$79.99', '\$16.99', '\$14.99', '\$1.00', '\$29.99', '\$12.99', '\$2.49', '\$24.99', '\$10.99', '\$1.50', '\$19.99', '\$15.99', '\$33.99', '\$74.99', '\$39.99', '\$3.95', '\$4.49', '\$1.70', '\$8.99', '\$2.00', '\$3.88', '\$25.99', '\$399.99', '\$17.99', '\$400.00', '\$3.02', '\$1.76', '\$4.84', '\$4.77', '\$1.61', '\$2.50', '\$1.59', '\$6.49', '\$1.29', '\$5.00', '\$13.99', '\$299.99', '\$379.99', '\$37.99', '\$18.99', '\$389.99', '\$19.90', '\$8.49', '\$1.75', '\$14.00', '\$4.85', '\$46.99', '\$109.99', '\$154.99', '\$3.08', '\$2.59', '\$4.80', '\$1.96', '\$19.40', '\$3.90', '\$4.59', '\$15.46', '\$3.04', '\$4.29', '\$2.60', '\$3.28', '\$4.60', '\$28.99', '\$2.95', '\$2.90', '\$1.97', '\$200.00', '\$89.99', '\$2.56', '\$30.99', '\$3.61', '\$394.99', '\$1.26', '\$1.20', '\$1.04'], dtype=object)

In [32]:

1 android_final = android_english[android_english["Price"] == "0"]

```
In [33]: 1 android_final.head()
```

Out[33]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	U
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	`
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	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design; Creativity	J
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5.6M	50,000+	Free	0	Everyone	Art & Design	2

```
In [34]: 1 android_final.shape
```

Out[34]: (8863, 13)


```
In [35]: 1 android_final["Category"].value_counts(normalize = True)*100
```

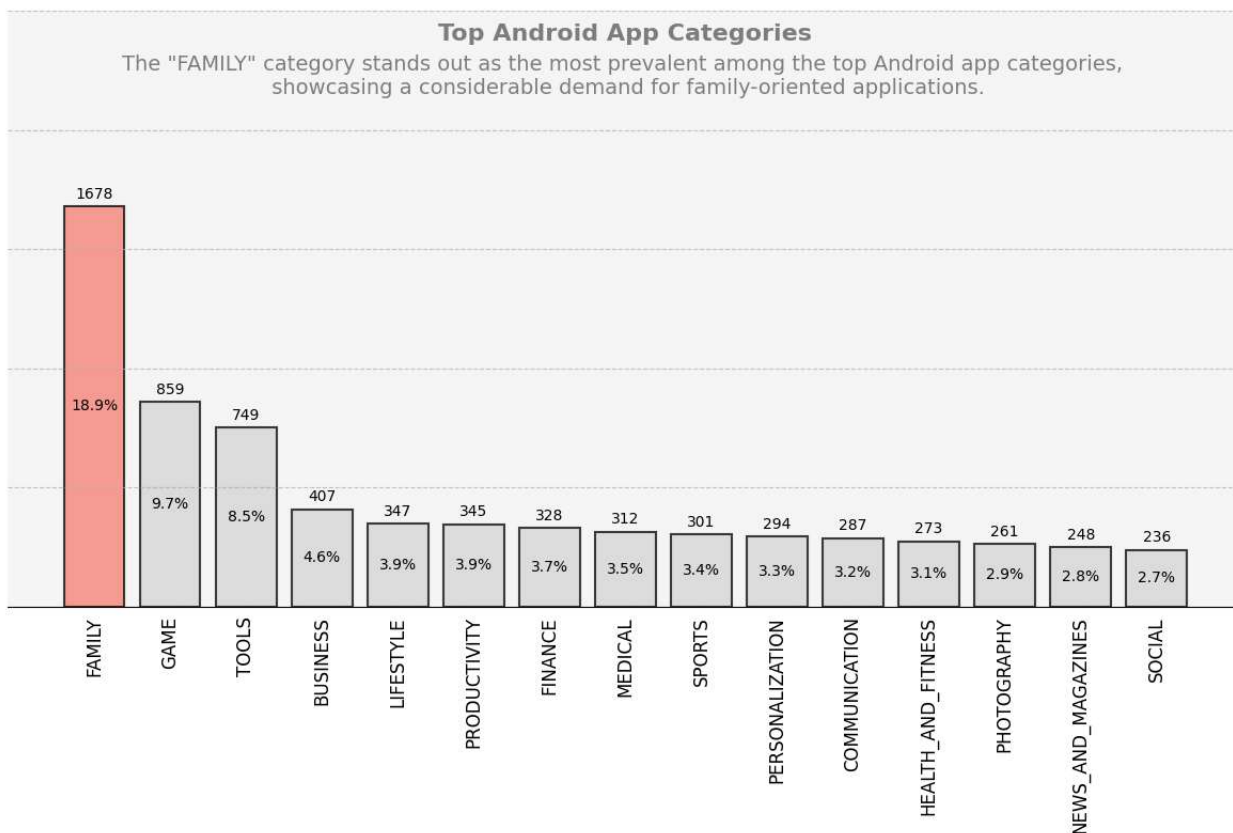
```
Out[35]: Category
FAMILY          18.932641
GAME            9.691978
TOOLS           8.450863
BUSINESS        4.592125
LIFESTYLE       3.915153
PRODUCTIVITY   3.892587
FINANCE        3.700779
MEDICAL        3.520253
SPORTS         3.396141
PERSONALIZATION 3.317161
COMMUNICATION  3.238181
HEALTH_AND_FITNESS 3.080221
PHOTOGRAPHY    2.944827
NEWS_AND_MAGAZINES 2.798150
SOCIAL         2.662755
TRAVEL_AND_LOCAL 2.335552
SHOPPING       2.245289
BOOKS_AND_REFERENCE 2.143744
DATING         1.861672
VIDEO_PLAYERS  1.793975
MAPS_AND_NAVIGATION 1.399075
FOOD_AND_DRINK 1.241115
EDUCATION      1.173418
ENTERTAINMENT  0.959043
LIBRARIES_AND_DEMO 0.936477
AUTO_AND_VEHICLES 0.925195
HOUSE_AND_HOME 0.823649
WEATHER        0.801083
EVENTS         0.710820
PARENTING      0.654406
ART_AND_DESIGN 0.643123
COMICS         0.620557
BEAUTY         0.597992
Name: proportion, dtype: float64
```

In [36]:

```

1  # data
2  categories = android_final["Category"].value_counts().index[:15]
3  counts = android_final["Category"].value_counts().values[:15]
4  percentage = round(android_final["Category"].value_counts(normalize = True)*100,1)[:15]
5
6  # create stylish bar chart
7
8  plt.figure(figsize=(12, 8))
9  bars = plt.bar(categories, counts, color="lightgray" , alpha=0.75, edgecolor="black" ,
10 plt.xticks(rotation=90, fontsize=12)
11 plt.yticks(fontsize=12)
12 plt.grid(axis='y', linestyle='--' , alpha=0.7)
13 plt.grid(axis='x', linestyle='')
14 plt.xticks(fontsize=12)
15 plt.yticks(range(0, 3000, 500), [], fontsize=12)
16 plt.tick_params(bottom = 0, left = 0)
17
18 # find the category with the highest count
19 max_count_category = categories[counts.argmax()]
20
21 max_count_index = list(categories).index(max_count_category)
22 bars[max_count_index].set_color('salmon')
23 bars[max_count_index].set_edgecolor('black')
24
25 for bar, perc in zip(bars, percentage):
26     height = bar.get_height()
27     plt.text(bar.get_x() + bar.get_width()/2, height + 20, '%d' % int(height), ha='center')
28     plt.text(bar.get_x() + bar.get_width()/2, height/2, f'{perc}%', ha='center', va='center')
29
30
31 # adding background color
32 ax = plt.gca()
33 ax.set_facecolor('#f7f7f7')
34
35 #adding chart title inside the chart
36 plt.text(0.5,0.95, 'Top Android App Categories', horizontalalignment='center' , fontsize=14,
37         color = 'gray', fontweight='bold')
38
39 # adding conclusion inside the chart
40 plt.text(0.5,0.86, 'The "FAMILY" category stands out as the most prevalent among the top 15 categories',
41         horizontalalignment = 'center', fontsize=14, transform=plt.gca().transAxes, color='gray')
42
43 #remove spines
44 for i in ["top" , "right" , "left"]:
45     plt.gca().spines[i].set_visible(False)
46
47 plt.tight_layout()
48 plt.show()

```



Most Popular App By Genre

```
In [37]: 1 android_final["Installs"].value_counts(normalize = True)*100
```

```
Out[37]: Installs
1,000,000+      15.739592
100,000+       11.553650
10,000,000+    10.515627
10,000+        10.199707
1,000+         8.405732
100+           6.916394
5,000,000+     6.837414
500,000+       5.573733
50,000+        4.772650
5,000+         4.513145
10+            3.542818
500+           3.249464
50,000,000+    2.290421
100,000,000+   2.121178
50+            1.918086
5+             0.789800
1+             0.507729
500,000,000+   0.270789
1,000,000,000+ 0.225657
0+            0.045131
0             0.011283
Name: proportion, dtype: float64
```

```
In [40]: 1 android_final["Installs_int"] = android_final["Installs"].str.replace(",", "").str.replace
```

C:\Users\user\AppData\Local\Temp\ipykernel_11364\3840374705.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
    android_final["Installs_int"] = android_final["Installs"].str.replace(",", "").str.replace("+", "").astype(int)
```

```
In [43]: 1 install_frq = android_final["Installs_int"].value_counts().sort_index()
2 install_frq = install_frq[install_frq.index > 500]
3 install_frq
```

```
Out[43]: Installs_int
1000      745
5000      400
10000     904
50000     423
100000    1024
500000     494
1000000   1395
5000000    606
10000000   932
50000000   203
100000000  188
500000000   24
1000000000  20
Name: count, dtype: int64
```

```
In [45]: 1 install_frq_per = round(android_final["Installs_int"].value_counts(normalize=True)*100
2 install_frq_per = install_frq_per[install_frq_per.index > 500]
3 install_frq_per
```

```
Out[45]: Installs_int
1000      8.41
5000      4.51
10000     10.20
50000      4.77
100000     11.55
500000      5.57
1000000    15.74
5000000     6.84
10000000    10.52
50000000     2.29
100000000    2.12
500000000     0.27
1000000000    0.23
Name: proportion, dtype: float64
```

```
In [46]: 1 COMMUNICATION
```

```
In [47]: 1 install_frq.index = install_frq.index.map(alphanumeric_units)
        2 install_frq
```

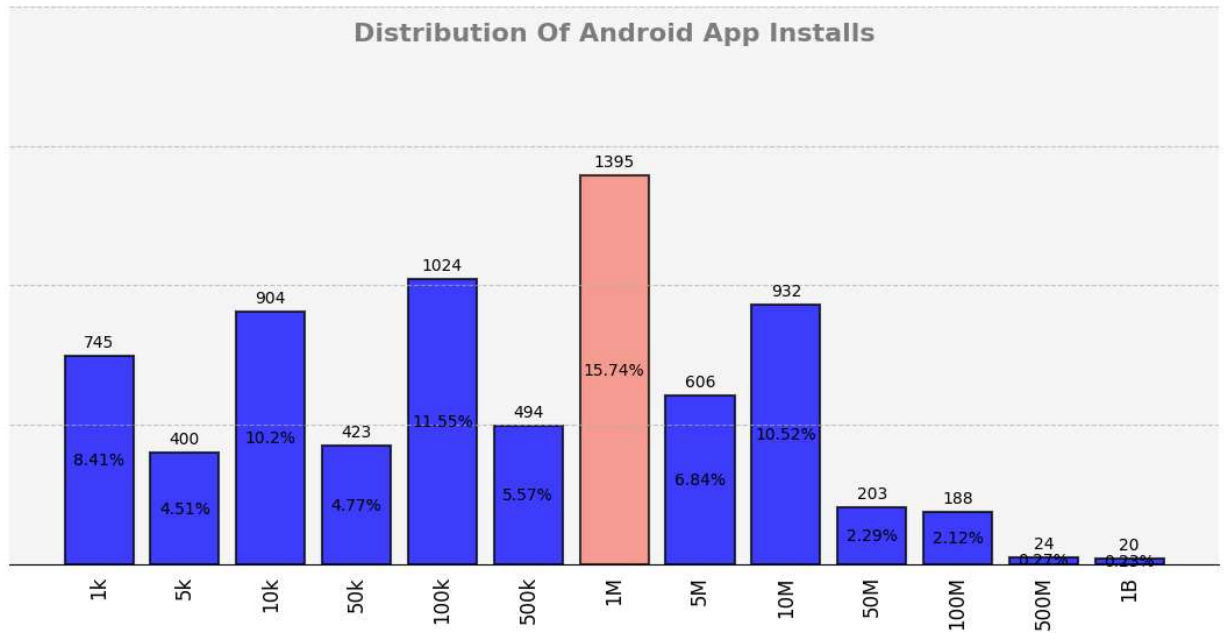
```
Out[47]: Installs_int
1k      745
5k      400
10k     904
50k     423
100k    1024
500k    494
1M      1395
5M      606
10M     932
50M     203
100M    188
500M    24
1B      20
Name: count, dtype: int64
```

In [54]:

```

1  # data
2  categories = install_frq.index
3  counts = install_frq.values
4  percentage = install_frq_per.values
5
6  # create stylish bar chart
7
8  plt.figure(figsize=(12, 7))
9  bars = plt.bar(categories, counts, color="blue" , alpha=0.75, edgecolor="black" , linewidth=1)
10 plt.xticks(rotation=90, fontsize=12)
11 plt.yticks(fontsize=12)
12 plt.grid(axis='y', linestyle='--' , alpha=0.7)
13 plt.grid(axis='x', linestyle='')
14 plt.xticks(fontsize=12)
15 plt.yticks(range(0, 2500, 500), [], fontsize=12)
16 plt.tick_params(bottom = 0, left = 0)
17
18 # find the category with the highest count
19 max_count_category = categories[counts.argmax()]
20
21 max_count_index = list(categories).index(max_count_category)
22 bars[max_count_index].set_color('salmon')
23 bars[max_count_index].set_edgecolor('black')
24
25 for bar, perc in zip(bars, percentage):
26     height = bar.get_height()
27     plt.text(bar.get_x() + bar.get_width()/2, height + 20, '%d' % int(height), ha='center')
28     plt.text(bar.get_x() + bar.get_width()/2, height/2, f'{perc}%', ha='center', va='center')
29
30
31 # adding background color
32 ax = plt.gca()
33 ax.set_facecolor('#f7f7f7')
34
35 #adding chart title inside the chart
36 plt.text(0.5,0.94, 'Distribution Of Android App Installs', horizontalalignment='center',
37         color = 'gray', fontweight='bold')
38
39 # adding conclusion inside the chart
40 plt.text(0.5,-0.35, 'From the data provided, its evident that the majority of Android App Installs are from the top 5 categories',
41         horizontalalignment = 'center', fontsize=9, transform=plt.gca().transAxes, color='gray')
42
43 #remove spines
44 for i in ["top" , "right" , "left"]:
45     plt.gca().spines[i].set_visible(False)
46
47 plt.tight_layout()
48 plt.show()

```



From the data provided, its evident that the majority of Android App installs fall within the lower range. with the highest number of installs being in the 1K to 10M range.

```
In [55]: 1 categories_android = android_final["Category"].unique()
          2 categories_android
```

```
Out[55]: 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'],
              dtype=object)
```

```
In [56]: 1 pd.pivot_table(android_final, values = 'Installs_int', index='Category', aggfunc='mean')
```

Out[56]:

	Installs_int
Category	
ART_AND_DESIGN	1.986335e+06
AUTO_AND_VEHICLES	6.473178e+05
BEAUTY	5.131519e+05
BOOKS_AND_REFERENCE	8.767812e+06
BUSINESS	1.712290e+06
COMICS	8.176573e+05
COMMUNICATION	3.845612e+07
DATING	8.540288e+05
EDUCATION	1.820673e+06
ENTERTAINMENT	1.164071e+07
EVENTS	2.535422e+05
FAMILY	3.694276e+06
FINANCE	1.387692e+06
FOOD_AND_DRINK	1.924898e+06
GAME	1.556097e+07
HEALTH_AND_FITNESS	4.188822e+06
HOUSE_AND_HOME	1.331541e+06
LIBRARIES_AND_DEMO	6.385037e+05
LIFESTYLE	1.433676e+06
MAPS_AND_NAVIGATION	4.056942e+06
MEDICAL	1.206165e+05
NEWS_AND_MAGAZINES	9.549178e+06
PARENTING	5.426036e+05
PERSONALIZATION	5.201483e+06
PHOTOGRAPHY	1.780563e+07
PRODUCTIVITY	1.678733e+07
SHOPPING	7.036877e+06
SOCIAL	2.325365e+07
SPORTS	3.638640e+06
TOOLS	1.068230e+07
TRAVEL_AND_LOCAL	1.398408e+07
VIDEO_PLAYERS	2.472787e+07
WEATHER	5.074486e+06

```
In [57]: 1 # dataframe without scientific notation
2 pd.options.display.float_format = '{:.0f}'.format
```



```
In [60]: 1 categories_installs = pd.pivot_table(android_final, values = 'Installs_int', index='Category',
2       categories_installs = categories_installs.sort_values(by="Installs_int", ascending=False)
3       categories_installs = categories_installs["Installs_int"]
4       categories_installs
```

```
Out[60]: Category
COMMUNICATION      38456119
VIDEO_PLAYERS      24727872
SOCIAL              23253652
PHOTOGRAPHY        17805628
PRODUCTIVITY       16787331
GAME               15560966
TRAVEL_AND_LOCAL   13984078
ENTERTAINMENT      11640706
TOOLS              10682301
NEWS_AND_MAGAZINES 9549178
BOOKS_AND_REFERENCE 8767812
SHOPPING           7036877
PERSONALIZATION    5201483
WEATHER            5074486
HEALTH_AND_FITNESS 4188822
MAPS_AND_NAVIGATION 4056942
FAMILY             3694276
SPORTS             3638640
ART_AND_DESIGN     1986335
FOOD_AND_DRINK     1924898
EDUCATION          1820673
BUSINESS           1712290
LIFESTYLE          1433676
FINANCE            1387692
HOUSE_AND_HOME     1331541
DATING             854029
COMICS             817657
AUTO_AND_VEHICLES  647318
LIBRARIES_AND_DEMO 638504
PARENTING          542604
BEAUTY             513152
EVENTS             253542
MEDICAL            120616
Name: Installs_int, dtype: float64
```

```
In [61]: 1 # alphanumeric units
2 def alphanumeric_units(value):
3     if value >= 1e9:
4         return f'{value / 1e9:.1f}B'
5     elif value >= 1e6:
6         return f'{value / 1e6:.1f}M'
7     elif value >= 1e3:
8         return f'{value / 1e3:.1f}k'
9     else:
10        return f'{value:.1f}'
```

```
In [62]: 1 categories_installs_unit = categories_installs.map(alphanumeric_units)
        2 categories_installs_unit
```

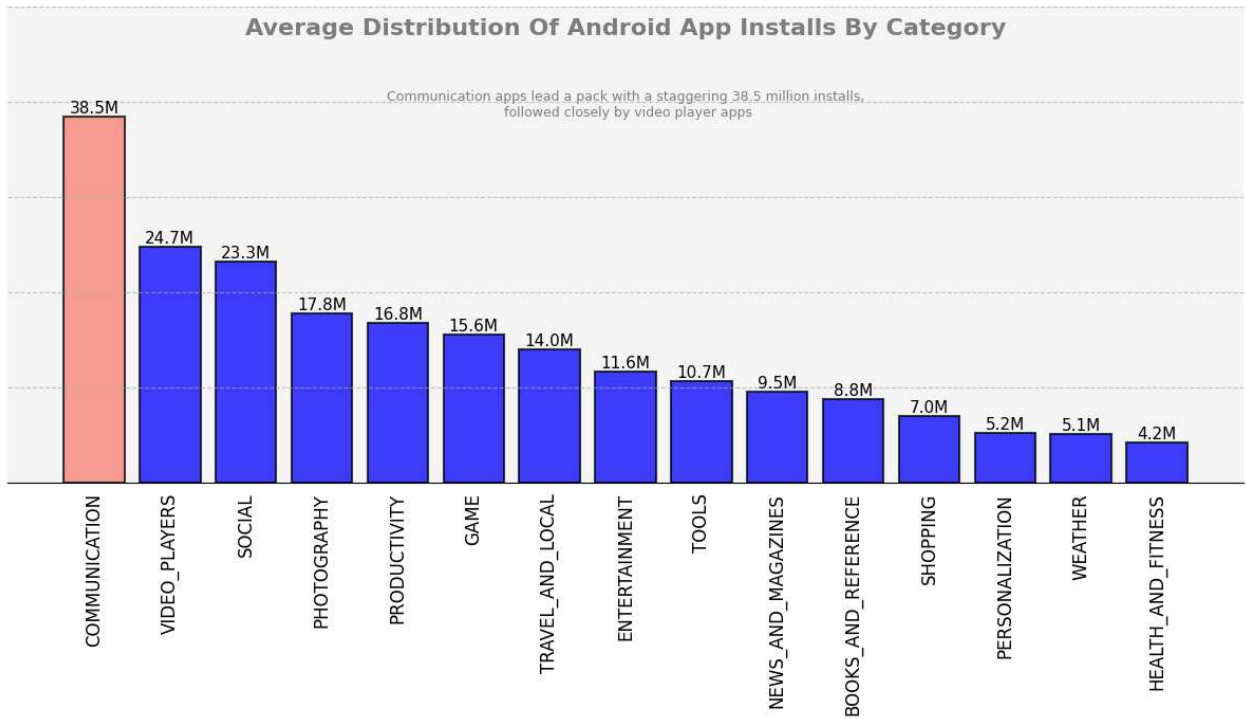
```
Out[62]: Category
COMMUNICATION      38.5M
VIDEO_PLAYERS      24.7M
SOCIAL              23.3M
PHOTOGRAPHY        17.8M
PRODUCTIVITY       16.8M
GAME                15.6M
TRAVEL_AND_LOCAL   14.0M
ENTERTAINMENT      11.6M
TOOLS               10.7M
NEWS_AND_MAGAZINES 9.5M
BOOKS_AND_REFERENCE 8.8M
SHOPPING            7.0M
PERSONALIZATION    5.2M
WEATHER             5.1M
HEALTH_AND_FITNESS 4.2M
MAPS_AND_NAVIGATION 4.1M
FAMILY              3.7M
SPORTS              3.6M
ART_AND_DESIGN      2.0M
FOOD_AND_DRINK      1.9M
EDUCATION           1.8M
BUSINESS            1.7M
LIFESTYLE           1.4M
FINANCE             1.4M
HOUSE_AND_HOME      1.3M
DATING              854.0k
COMICS              817.7k
AUTO_AND_VEHICLES   647.3k
LIBRARIES_AND_DEMO  638.5k
PARENTING           542.6k
BEAUTY              513.2k
EVENTS              253.5k
MEDICAL             120.6k
Name: Installs_int, dtype: object
```

In [67]:

```

1  # data
2  categories = categories_installs.index[:15]
3  counts = categories_installs.values[:15]
4
5  # create stylish bar chart
6
7  plt.figure(figsize=(12, 7))
8  bars = plt.bar(categories, counts, color="blue" , alpha=0.75, edgecolor="black" , linewidth=1)
9  plt.xticks(rotation=90, fontsize=12)
10 plt.yticks(fontsize=12)
11 plt.grid(axis='y', linestyle='--' , alpha=0.7)
12 plt.grid(axis='x', linestyle='')
13 plt.xticks(fontsize=12)
14 plt.yticks(range(0, 60000000, 10000000), [], fontsize=12)
15 plt.tick_params(bottom = 0, left = 0)
16
17 # find the category with the highest count
18 max_count_category = categories[counts.argmax()]
19
20 max_count_index = list(categories).index(max_count_category)
21 bars[max_count_index].set_color('salmon')
22 bars[max_count_index].set_edgecolor('black')
23
24 for bar, units in zip(bars, categories_installs_unit.values):
25     height = bar.get_height()
26     plt.text(bar.get_x() + bar.get_width()/2, height + 25, units , ha='center', va='bottom')
27
28
29
30 # adding background color
31 ax = plt.gca()
32 ax.set_facecolor('#f7f7f7')
33
34 #adding chart title inside the chart
35 plt.text(0.5,0.94, 'Average Distribution Of Android App Installs By Category', horizontalalignment='center',
36         color = 'gray', fontweight='bold')
37
38 # adding conclusion inside the chart
39 plt.text(0.5,0.77, 'Communication apps lead a pack with a staggering 38.5 million installs', horizontalalignment='center',
40         fontweight='bold', fontstyle='italic', fontcolor='red',
41         horizontalalignment = 'center', fontsize=9, transform=plt.gca().transAxes, color='red')
42
43 #remove spines
44 for i in ["top" , "right" , "left"]:
45     plt.gca().spines[i].set_visible(False)
46
47 plt.tight_layout()
48 plt.show()

```



```
In [68]: 1 category_group = android_final.groupby("Category")
```

```
In [70]: 1 COMMUNICATION = category_group.get_group("COMMUNICATION").sort_values(by="Installs_int")
2 COMMUNICATION.head()
```

Out[70]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
336	WhatsApp Messenger	COMMUNICATION	4	69119316	Varies with device	1,000,000,000+	Free	0	Everyone	Communications
382	Messenger – Text and Video Chat for Free	COMMUNICATION	4	56646578	Varies with device	1,000,000,000+	Free	0	Everyone	Communications
464	Hangouts	COMMUNICATION	4	3419513	Varies with device	1,000,000,000+	Free	0	Everyone	Communications
411	Google Chrome: Fast & Secure	COMMUNICATION	4	9643041	Varies with device	1,000,000,000+	Free	0	Everyone	Communications
391	Skype - free IM & video calls	COMMUNICATION	4	10484169	Varies with device	1,000,000,000+	Free	0	Everyone	Communications

```
In [71]: 1 # alphanumeric units
2 def alphanumeric_units(value):
3     if value >= 1e9:
4         return f'{value / 1e9:.0f}B'
5     elif value >= 1e6:
6         return f'{value / 1e6:.0f}M'
7     elif value >= 1e3:
8         return f'{value / 1e3:.0f}k'
9     else:
10        return f'{valu:.1f}'
```

```
In [73]: 1 categories_installs.index[:15]
```

```
Out[73]: Index(['COMMUNICATION', 'VIDEO_PLAYERS', 'SOCIAL', 'PHOTOGRAPHY',
               'PRODUCTIVITY', 'GAME', 'TRAVEL_AND_LOCAL', 'ENTERTAINMENT', 'TOOLS',
               'NEWS_AND_MAGAZINES', 'BOOKS_AND_REFERENCE', 'SHOPPING',
               'PERSONALIZATION', 'WEATHER', 'HEALTH_AND_FITNESS'],
              dtype='object', name='Category')
```

```
In [74]: 1 df = COMMUNICATION[["App", "Installs_int"]].head(15)
2 df["Installs_int_unit"] = df["Installs_int"].map(alphanumeric_units)
3 df
```

```
Out[74]:
```

	App	Installs_int	Installs_int_unit
336	WhatsApp Messenger	1000000000	1B
382	Messenger – Text and Video Chat for Free	1000000000	1B
464	Hangouts	1000000000	1B
411	Google Chrome: Fast & Secure	1000000000	1B
391	Skype - free IM & video calls	1000000000	1B
451	Gmail	1000000000	1B
403	LINE: Free Calls & Messages	500000000	500M
4676	Viber Messenger	500000000	500M
420	UC Browser - Fast Download Private & Secure	500000000	500M
371	Google Duo - High Quality Video Calls	500000000	500M
383	imo free video calls and chat	500000000	500M
393	Who	100000000	100M
4633	UC Browser Mini -Tiny Fast Private & Secure	100000000	100M
4602	Truecaller: Caller ID, SMS spam blocking & Dialer	100000000	100M
4592	Telegram	100000000	100M

```
In [77]: 1 df = category_group.get_group("VIDEO_PLAYERS").sort_values(by="Installs_int", ascending=True)
2 df = df[["App", "Installs_int"]].head(15)
3 df["Installs_int_unit"] = df['Installs_int'].map(alphanumeric_units)
4 df
```

Out[77]:

	App	Installs_int	Installs_int_unit
3665	YouTube	1000000000	1B
3687	Google Play Movies & TV	1000000000	1B
3711	MX Player	500000000	500M
3675	VLC for Android	100000000	100M
4688	VivaVideo - Video Editor & Photo Movie	100000000	100M
4032	Dubsmash	100000000	100M
10647	Motorola FM Radio	100000000	100M
4696	VideoShow-Video Editor, Video Maker, Beauty Ca...	100000000	100M
3672	Motorola Gallery	100000000	100M
3691	Samsung Video Library	50000000	50M
4038	DU Recorder – Screen Recorder, Video Editor, Live	50000000	50M
3693	LIKE – Magic Video Maker & Community	50000000	50M
3686	Vigo Video	50000000	50M
4049	KineMaster – Pro Video Editor	50000000	50M
5612	Ringdroid	50000000	50M

```
In [78]: 1 df = category_group.get_group("PHOTOGRAPHY").sort_values(by="Installs_int", ascending=True)
2 df = df[["App", "Installs_int"]].head(15)
3 df["Installs_int_unit"] = df['Installs_int'].map(alphanumeric_units)
4 df
```

Out[78]:

	App	Installs_int	Installs_int_unit
2884	Google Photos	1000000000	1B
4574	S Photo Editor - Collage Maker , Photo Collage	100000000	100M
2949	Camera360: Selfie Photo Editor with Funny Sticker	100000000	100M
2908	Retrica	100000000	100M
8307	LINE Camera - Photo editor	100000000	100M
2921	Photo Editor Pro	100000000	100M
2847	Sweet Selfie - selfie camera, beauty cam, phot...	100000000	100M
2937	BeautyPlus - Easy Photo Editor & Selfie Camera	100000000	100M
2938	PicsArt Photo Studio: Collage Maker & Pic Editor	100000000	100M
5057	AR effect	100000000	100M
2833	YouCam Makeup - Magic Selfie Makeovers	100000000	100M
2942	Z Camera - Photo Editor, Beauty Selfie, Collage	100000000	100M
2943	PhotoGrid: Video & Pic Collage Maker, Photo Ed...	100000000	100M
2944	Candy Camera - selfie, beauty camera, photo ed...	100000000	100M
2945	YouCam Perfect - Selfie Photo Editor	100000000	100M

In []:

1