WHAT ARE THE FEATURES OF A SUCCESSFUL ANDROID APP?

Hania Abdul Baseer (a1811480)

The University of Adelaide,
September 2021

Introduction

The mobile application industry has seen a massive growth over the past decade. Thousands of new mobile applications are being added to Google Play store every day (Sharma, 2019). From the data obtained from millions of current android applications; we can reach meaningful insights about the effects of basic app features on an app's success. Knowledge of these type of features can be used by implementing them in apps, to increase the chances of the app's success. Hence, in this research project I have attempted to analyse the data of current android applications to answer, "**What are the Features of a Successful Android App?**".

This report covers the materials presented in the pitch, first cut and final presentation of my research project. This report will cover the motivation for my research question, brief description of the dataset used, and the methodology used to analyse the dataset. Next, the report contains details of the data processing and cleaning done before the making of data visualisations as part of the experimental setup. Lastly, the report presents the results and conclusions reached from the analysis of the data visualisations created. At the end of the report, the appendices contain all the relevant codes used for making of the data visualisations.

Motivation

Mobile applications are an important part of our daily lives, as statistics show that an average mobile owner uses 10 different mobile applications each day and 30 applications each month (buildfire, 2021). It is also evident that the mobile app industry has seen a growth over the last decade. The mobile app industry is a key revenue generator for Apple, Google, and thousands of other mobile app developers. However, as profitable as the industry may seem, not all applications are able to achieve success in terms of having high number of installs. Research shows that on average, a mobile app can lose an estimate of 77% of daily active users only during its first three days of installation (Manchanda, 2020). Investigating the reasons behind such issues can lead to reaching useful conclusions. And this can be done by making use of the millions of current apps' data. A mobile application has several basic features such as its price, age rating, genre, ad-support, in-app purchases and many more. Therefore, it is worth exploring whether an app having certain features effect its success. So that, features that can lead to high number of app installs can be implemented in new mobile applications and predictions of an app's success i.e., its number of installs can be made. This project will attempt to see what the features of a successful android app are, where an app's number of installs will be used as the measure of success.

Dataset

The dataset used in this project is obtained from Kaggle – an online community for data scientists and machine learning practitioners. The rich dataset was collected in June 2021 by Gautham Prakash who scrapped it from the Google Play store website using python and scrapy. Since then, it has been last updated three months ago. The dataset contains data of 2.3 million+ google play store applications that were released from the year 2010 to 2021. This dataset is rich and highly usable as it contains 23 attributes of the android apps.

Experimental Setup

The dataset had to be cleaned and prepared first for effective making of data visualisations. For my setup, I firstly removed all the null values. After removing the null values, the rows containing app data decreased from 2,312,944 to 1,287,191 apps. Then I changed the date format of the app release date to a format that was recognizable by python. The date format was changed from MMM dd, yyyy (for example: Feb 26, 2020) to yyyy-mm-dd (for example: 2020-02-26). Using the new date format, I made a new column that only contained the year of the app release for the purpose of my data visualisations.

Methods

The average number of installs of an app was used as the measure of success. Bar plots using matplotlib were used to see the average number of installs based on whether apps contained ads, in-app purchases, were free, content rating and what genre of app it belonged to. In all the bar plots the average number of installs for each x value was taken to normalize the number of installs. For example, since most apps are free, the bar plots would have a high number of installs for category of free apps, hence average number of installs per category were used for all bar plots. Secondly, 11 out of 48 app categories generally belonged to gaming categories so I combined these 11 categories as one and named it as “Games”. For example: category of action and racing were changed to games. For making of the interactive scatter plots, plotly was used. The scatter plots were used as it could show the relation between app rating and average number of installs as well as the relation between the year the app was released and the average of number of installs. Hence, the general method was to compare several basic features of an app with its number of installs, which represents the app’s success. More explanation on methods is provided in the appendix section of the report with all the codes used for data visualisations and its explanation.

Results

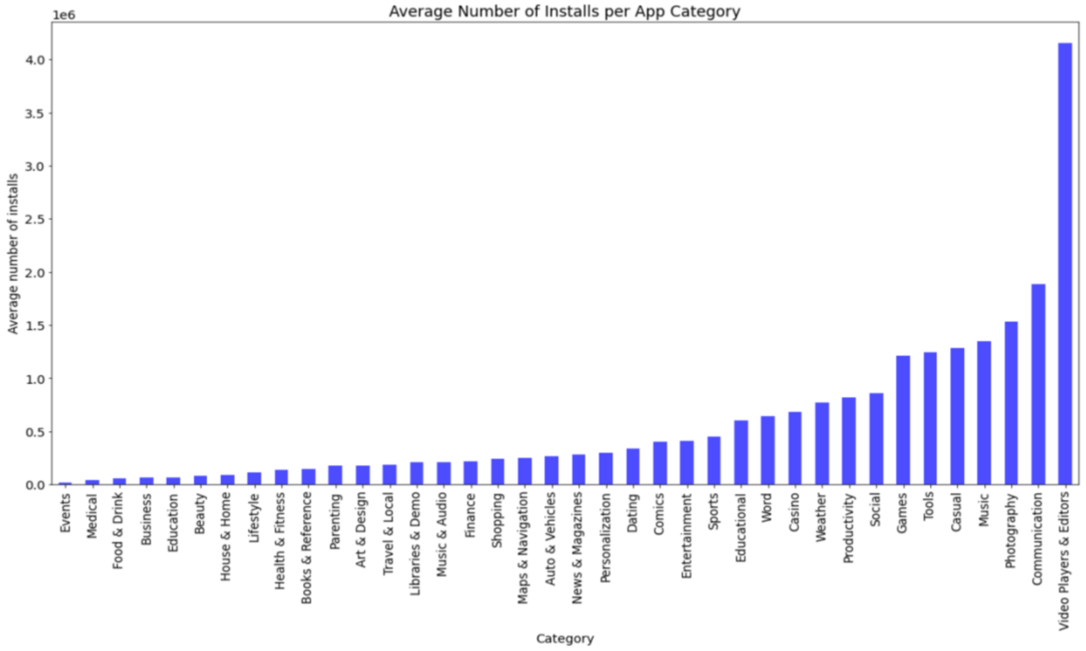


Figure 1: Bar plot showing the average number of installs per app category

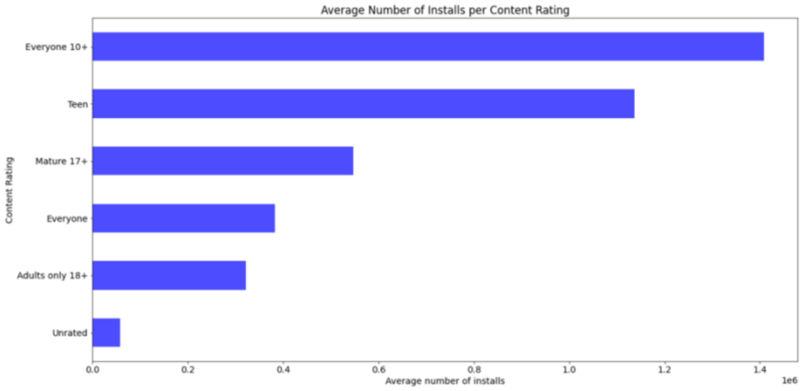


Figure 2: Bar plot showing the average number of app installs per content rating of app

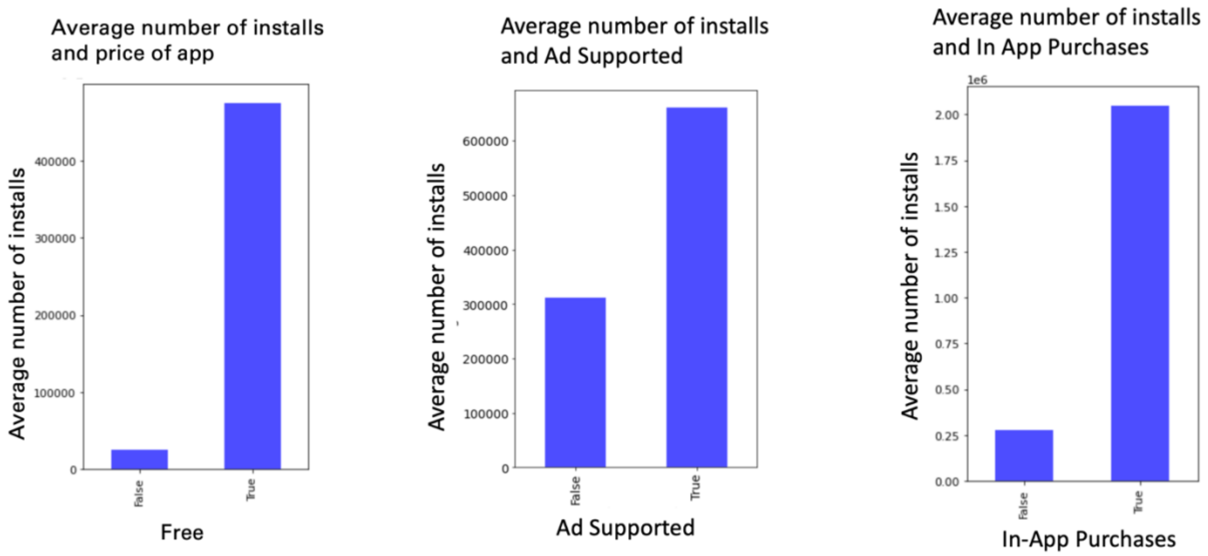


Figure 3: Bar plots showing the average number of app installs based on whether they are free to install, contain in-app purchases and contain ads.

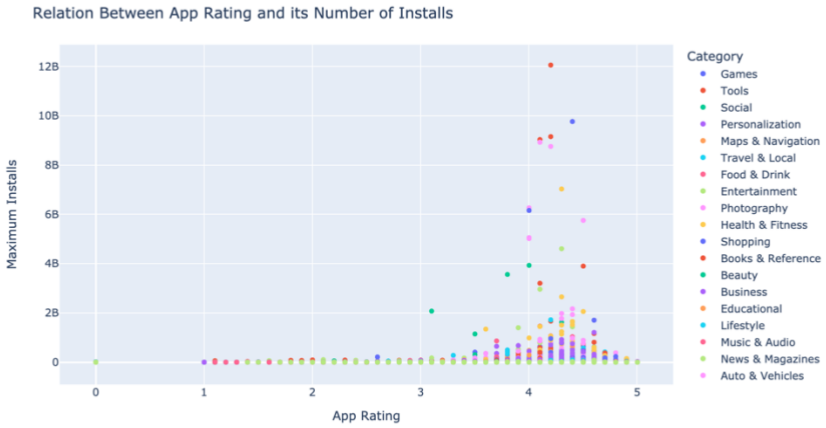


Figure 4: Interactive scatter plot showing the relation between an app's rating and its number of installs. Each dot point represents one app.

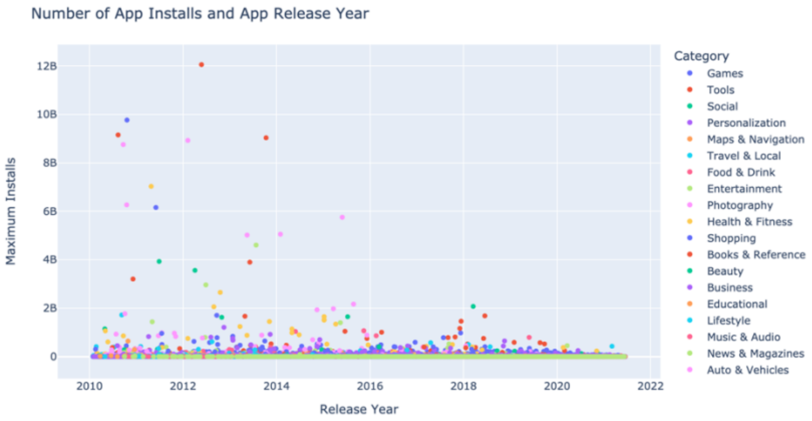


Figure 5: Interactive scatter plot showing the relation between app's release year and its number of installs. Each dot point represents one app.

Discussion

The data visualisations show how some features had higher number of installs. Figure 1 shows how the apps that belonged to the category of “Video Players and Editors” had much higher average number of installs than any other category. Its average number of installs were as high as 4,000,000 which is more than double the average number of installs of apps that belonged to the “Communications” category. The category of communications had the second highest average number of installs. Some examples of apps that belong to the “Video Players and Editors” category are YouTube and Tiktok which have a high number of active users as of 2021. Figure 2 suggests apps that are available to a wide range of age groups had the highest number of installs when compared to apps that were available to a more specific age group. The figure shows how apps that had content rating of “Everyone 10+” had the highest average number of installs while apps that had a content rating of “Adults only 18+” had the lowest number of average installs. This analysis ignores the content rating of “Unrated” as it can’t be used to make meaningful conclusions due to the unknown rating of the app. Figure 3 illustrates how apps that are free, contain in-app purchases and contain ads had more than double the average number of installs than apps that did not have these features. This is explained by the fact that most users do not wish to spend on paid apps and due to the abundance of free app choices in play store, most users opt to use free apps. Moreover, most apps that are free usually contain ads and users need to switch to an upgraded or “premium version” that contain no ads. And since most users do not wish to pay for upgraded version of an app, users opt for free version of apps that contain ads. Apps that contain in-app purchases had higher average number of installs; this could be explained by the fact that most apps on the market are free to install but contain in-app purchases to generate revenue.

Figure 4 suggests that an app with high app rating of more than 3.5 were more likely to have high number of installs. Although the correlation is weak, an app with high number of installs is likely to also have a high rating. And by looking at each data point, apps from the category of “Tools”, “Games” and “Photography” had overall the highest number of installs and an app rating of more 4. Lastly, Figure 5 shows that apps that were released before 2016 had higher number of installs when compared to newer released apps. This is due to the fact that apps released before 2016 had more time to generate installs. However, an interesting observation that can be made from the scatter plot is that most apps that belonged to the category of “Entertainment” and “News & Magazines” had the lowest number of installs throughout the years from 2010 up until 2021.

Conclusion

Based on the analysis of the data visualisations, it is suggested that some common features of a “successful” android app or android apps with highest number of installs are that they are: free, contain in-app purchases, contain ads, belong to the category of “Video Players & Editors”, available to wide range of age groups, have a high app rating and likely to have been released before 2016. These are some of the features that android apps with high number of installs had in common. Although the project has scope for improvement, the main takeaway from this project is that several important features of successful android apps were identified. Which can be used for further research and drawing of actionable insights for developers and other app making businesses to work on. So that they can capture the mobile app market.

References

Buildfire 2021, *Mobile App Download Statistics & Usage Statistics (2021),* viewed 8 September 2021, <https://buildfire.com/app-statistics/>.

Manchanda, A 2020, *10 Major Causes That Lead to Mobile App Failure*, Net Solutions, viewed 8 September 2021, < https://www.netsolutions.com/insights/10-major-causes- that-lead-to-mobile-app-failures/>.

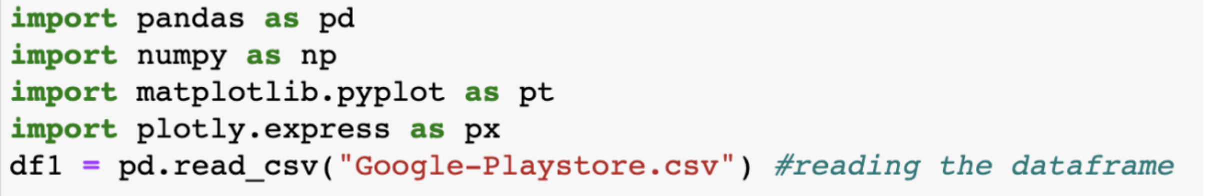
Prakash, G 2021, *Google Play Store Apps*, Kaggle, viewed 8 August 2021, <https://www.kaggle.com/gauthamp10/google-playstore-apps>.

Sharma, A 2021, *Top Google Play Store Statistics 2021 You Must Know*, Appinventiv, viewed 8 September 2021, < https://appinventiv.com/blog/google-play-store-statistics/>.

Appendices:

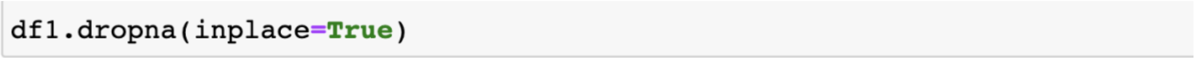
Appendix A: Codes used for data visualisations

Python3 jypter notebooks was used where the dataset was imported to and the libraries used were pandas, numpy, matplotlib, and plotly.

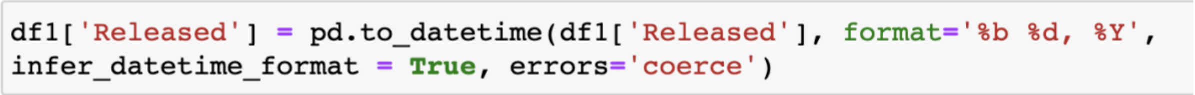


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as pt
import plotly.express as px
df1 = pd.read_csv("Google-Playstore.csv") #reading the dataframe
```

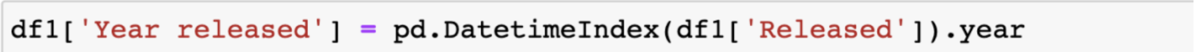
Null values were removed. Then using pandas, date format of year released was changed to a recognisable format by python and a new column was created to contain years only.



```
df1.dropna(inplace=True)
```

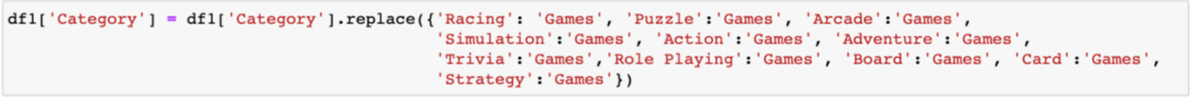


```
df1['Released'] = pd.to_datetime(df1['Released'], format='%b %d, %Y',
infer_datetime_format = True, errors='coerce')
```

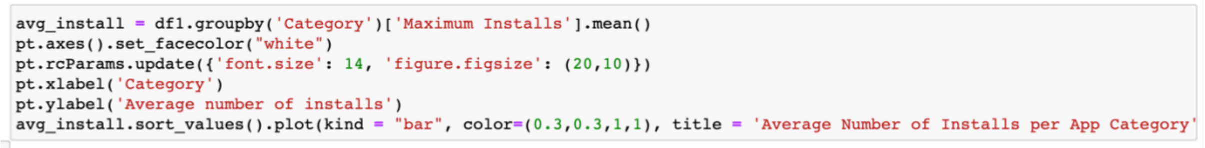


```
df1['Year released'] = pd.DatetimeIndex(df1['Released']).year
```

All the different gaming categories were combined into a category called "Games". Then the average number of installs of each app category was used to make a bar plot comparing the average number of installs of each category.

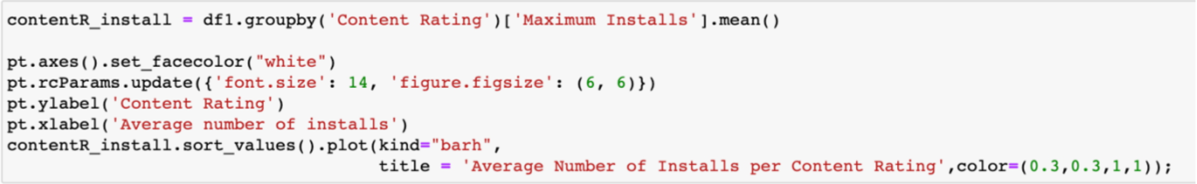


```
df1['Category'] = df1['Category'].replace({'Racing': 'Games', 'Puzzle':'Games', 'Arcade':'Games',
                                            'Simulation':'Games', 'Action':'Games', 'Adventure':'Games',
                                            'Trivia':'Games','Role Playing':'Games', 'Board':'Games', 'Card':'Games',
                                            'Strategy':'Games'})
```



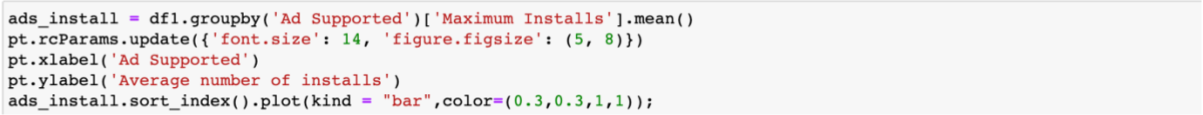
```
avg_install = df1.groupby('Category')['Maximum Installs'].mean()
pt.axes().set_facecolor("white")
pt.rcParams.update({'font.size': 14, 'figure.figsize': (20,10)})
pt.xlabel('Category')
pt.ylabel('Average number of installs')
avg_install.sort_values().plot(kind = "bar", color=(0.3,0.3,1,1), title = 'Average Number of Installs per App Category'
```

Average number of installs of each type of content rating was used to make a barplot with x values as the number of installs and y values as the type of content rating. Matplotlib was used.



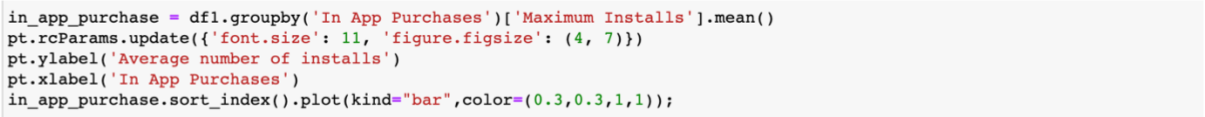
```
contentR_install = df1.groupby('Content Rating')['Maximum Installs'].mean()

pt.axes().set_facecolor("white")
pt.rcParams.update({'font.size': 14, 'figure.figsize': (6, 6)})
pt.ylabel('Content Rating')
pt.xlabel('Average number of installs')
contentR_install.sort_values().plot(kind="barh",
                                    title = 'Average Number of Installs per Content Rating',color=(0.3,0.3,1,1));
```

Average number of installs of app with and without ads were used to make a barplot with x values as the number of installs and y values as true or false, based on whether it had ads. Matplotlib was used.



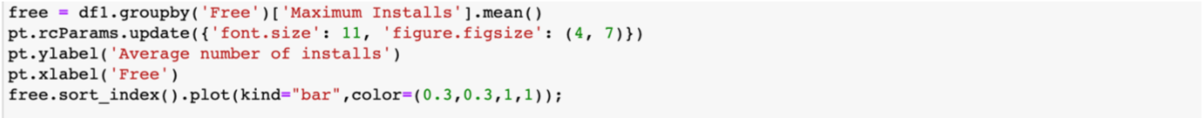
```
ads_install = df1.groupby('Ad Supported')['Maximum Installs'].mean()
pt.rcParams.update({'font.size': 14, 'figure.figsize': (5, 8)})
pt.xlabel('Ad Supported')
pt.ylabel('Average number of installs')
ads_install.sort_index().plot(kind = "bar",color=(0.3,0.3,1,1));
```

Average number of installs of app with and without in-app purchases was used to make a barplot with x values as the number of installs and y values as true or false, based on whether it had in-app purchases. Matplotlib was used.



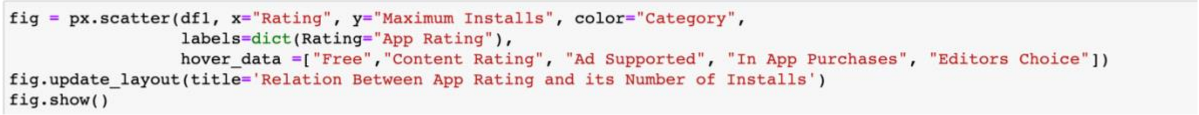
```
in_app_purchase = df1.groupby('In App Purchases')['Maximum Installs'].mean()
pt.rcParams.update({'font.size': 11, 'figure.figsize': (4, 7)})
pt.ylabel('Average number of installs')
pt.xlabel('In App Purchases')
in_app_purchase.sort_index().plot(kind="bar",color=(0.3,0.3,1,1));
```

Average number of installs of app that are free or not were used to make a barplot with x values as the number of installs and y values as true or false, based on whether it was free. Matplotlib was used.



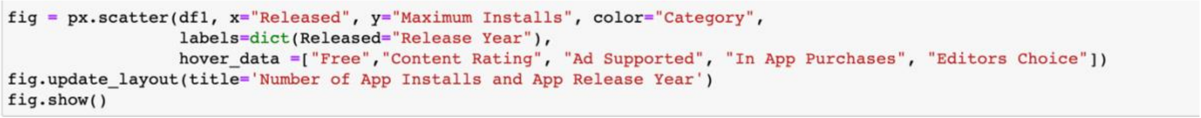
```
free = df1.groupby('Free')['Maximum Installs'].mean()
pt.rcParams.update({'font.size': 11, 'figure.figsize': (4, 7)})
pt.ylabel('Average number of installs')
pt.xlabel('Free')
free.sort_index().plot(kind="bar",color=(0.3,0.3,1,1));
```

Interactive scatter plot was created using Plotly with x values as the app rating and y values as the number of installs, each data point on the scatter plot represents one app, and its colour is based on what category it belongs to. Upon hovering to each data point, more information about the app can be seen like whether its free, its content rating, if it has ads, if it includes in-app purchases and whether it's an editor's choice app.



```
fig = px.scatter(df1, x="Rating", y="Maximum Installs", color="Category",
                 labels=dict(Rating="App Rating"),
                 hover_data =["Free","Content Rating", "Ad Supported", "In App Purchases", "Editors Choice"])
fig.update_layout(title='Relation Between App Rating and its Number of Installs')
fig.show()
```

Interactive scatter plot created using Plotly with x values as the year the app was released and y values as the number of installs, each data point on the scatter plot represents one app, and its colour is based on what category it belongs to. Upon hovering to each data point, more information about the app can be seen like whether its free, its content rating, if it has ads, if it includes in-app purchases and whether it's an editor's choice app.



```
fig = px.scatter(df1, x="Released", y="Maximum Installs", color="Category",
                 labels=dict(Released="Release Year"),
                 hover_data =["Free","Content Rating", "Ad Supported", "In App Purchases", "Editors Choice"])
fig.update_layout(title='Number of App Installs and App Release Year')
fig.show()
```