**Plotly Bar Charts & Scatter Plots: The Most Competitive & Popular App Categories**

If you were to release an app, would you choose to go after a competitive category with many other apps? Or would you target a popular category with a high number of downloads? Or perhaps you can target a category which is both popular but also one where the downloads are spread out among many different apps. That way, even if it’s more difficult to discover among all the other apps, your app has a better chance of getting installed, right? Let’s analyse this with bar charts and scatter plots and figure out which categories are dominating the market.

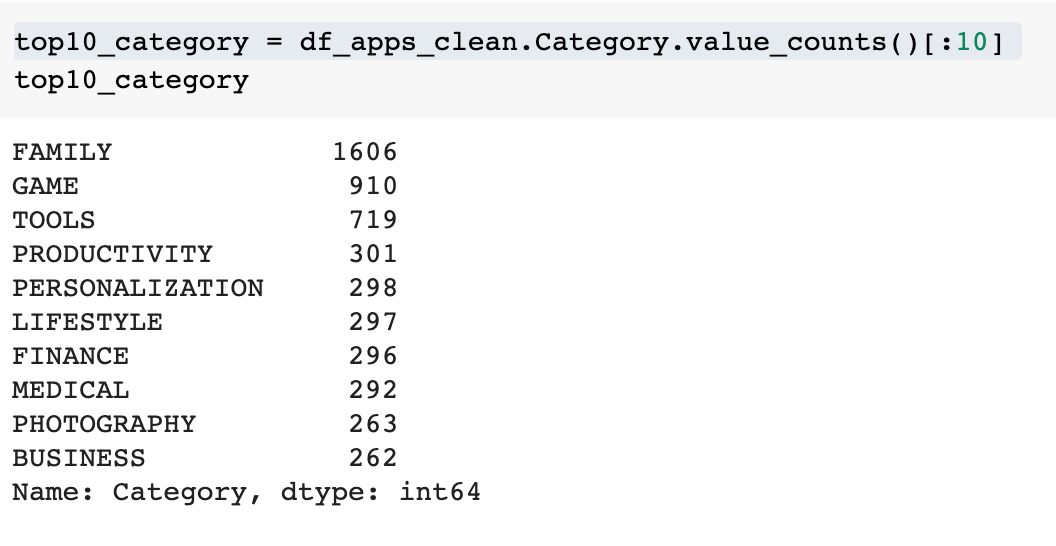
We can find the number of different categories like so:

1. df\_apps\_clean.Category.nunique()

Which shows us that we there are 33 unique categories.

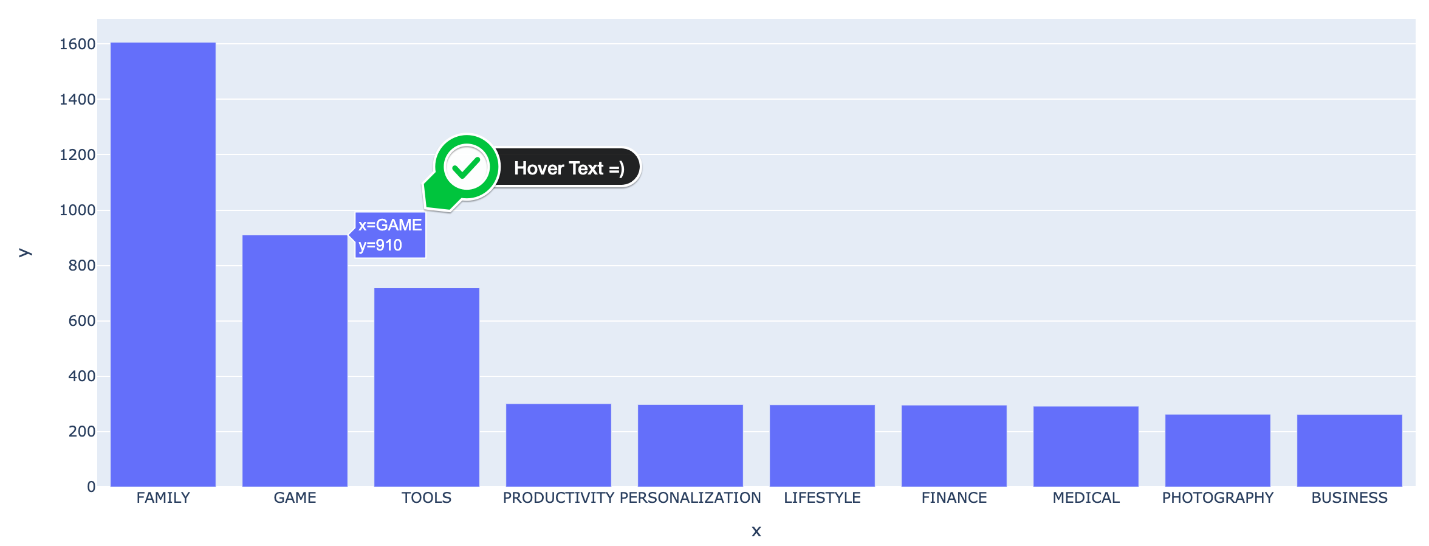
To calculate the number of apps per category we can use our old friend .value\_counts()

1. top10\_category = df\_apps\_clean.Category.value\_counts()[:10]



To visualise this data in a bar chart we can use the plotly express (our px) [bar()](https://plotly.com/python-api-reference/generated/plotly.express.bar.html#plotly.express.bar) function:

1. bar = px.bar(x = top10\_category.index, # index = category name
2. y = top10\_category.values)
4. bar.show()



Based on the number of apps, the Family and Game categories are the most competitive. Releasing yet another app into these categories will make it hard to get noticed.

But what if we look at it from a different perspective? What matters is not just the total number of apps in the category but how often apps are downloaded in that category. This will give us an idea of how popular a category is. First, we have to group all our apps by category and sum the number of installations:

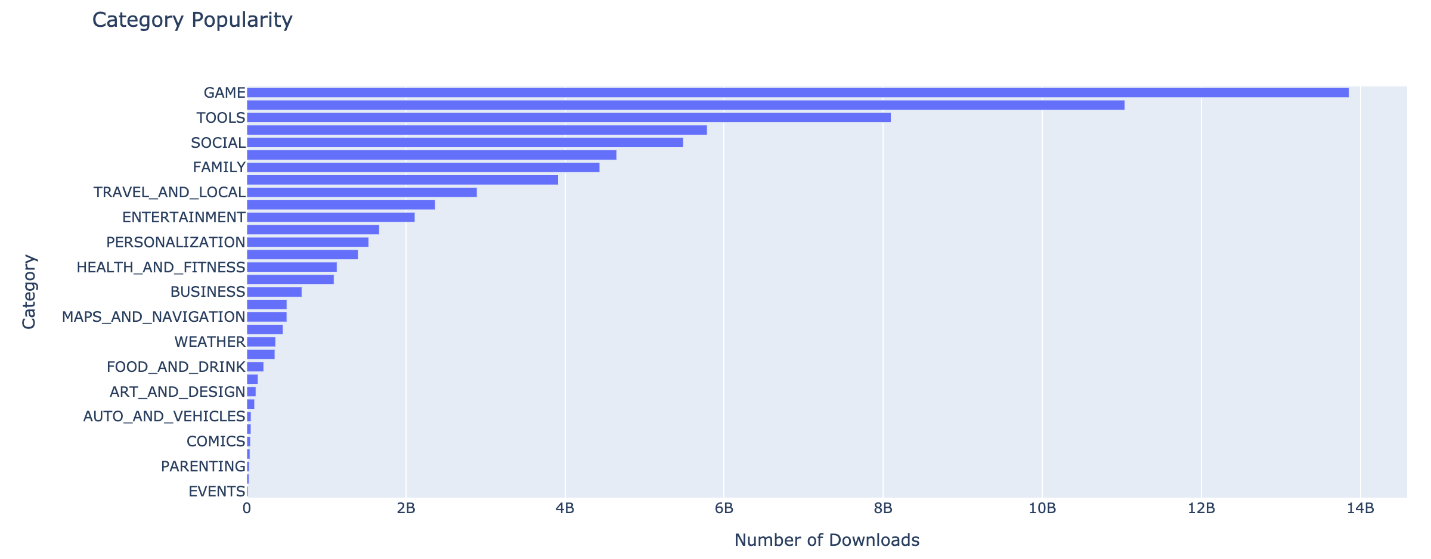
1. category\_installs = df\_apps\_clean.groupby('Category').agg({'Installs': pd.Series.sum})
2. category\_installs.sort\_values('Installs', ascending=True, inplace=True)

Then we can create a horizontal bar chart, simply by adding the orientation parameter:

1. h\_bar = px.bar(x = category\_installs.Installs,
2. y = category\_installs.index,
3. orientation='h')
5. h\_bar.show()

We can also add a custom title and axis labels like so:

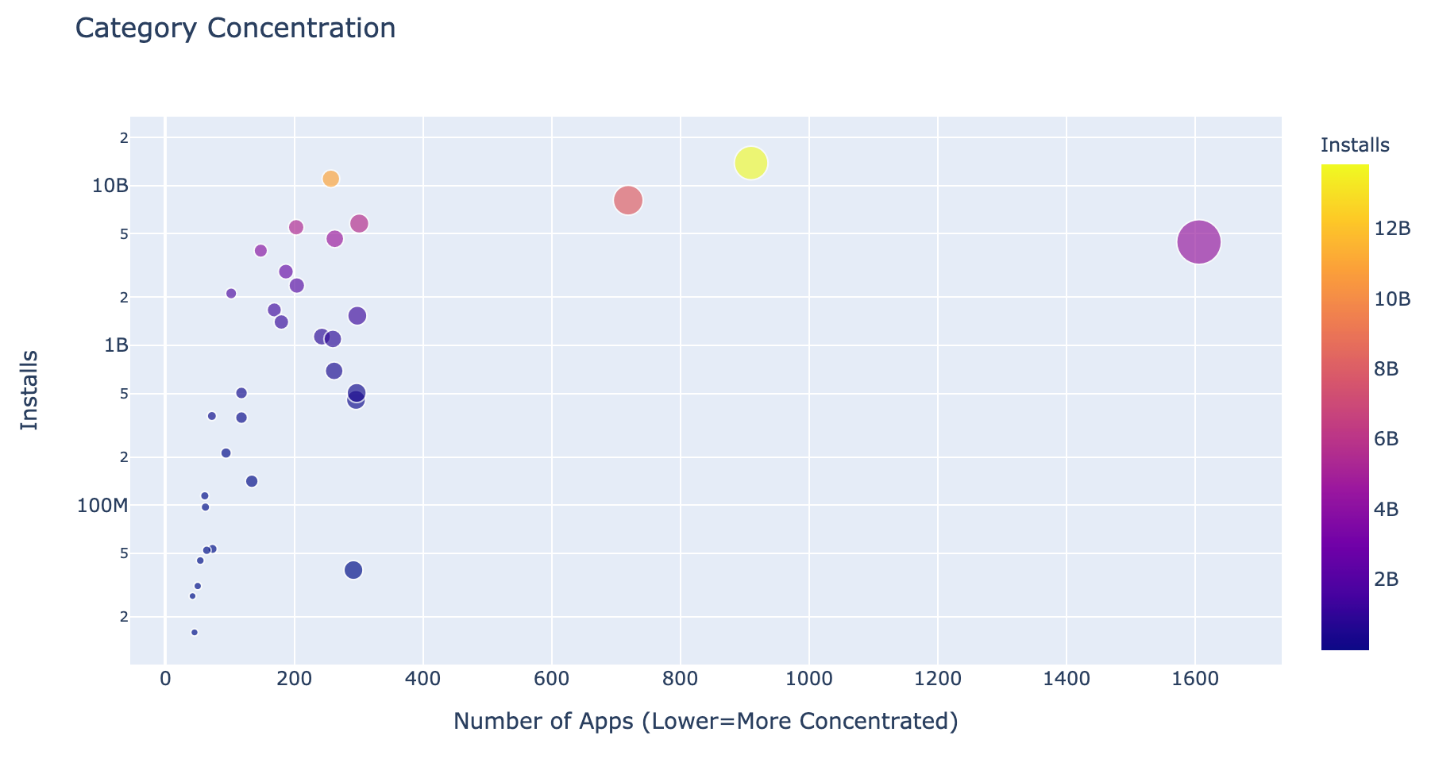
1. h\_bar = px.bar(x = category\_installs.Installs,
2. y = category\_installs.index,
3. orientation='h',
4. title='Category Popularity')
6. h\_bar.update\_layout(xaxis\_title='Number of Downloads', yaxis\_title='Category')
7. h\_bar.show()



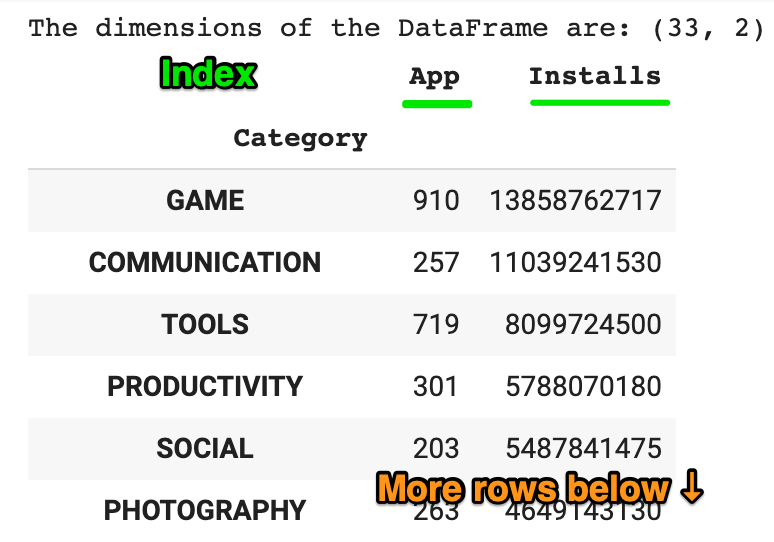
Now we see that Games and Tools are actually the most popular categories. If we plot the popularity of a category next to the number of apps in that category we can get an idea of how concentrated a category is. Do few apps have most of the downloads or are the downloads spread out over many apps?

Challenge

As a challenge, let’s use plotly to create a scatter plot that looks like this:



* Create a DataFrame that has the number of apps in one column and the number of installs in another:



* Then use the [plotly express examples from the documentation](https://plotly.com/python/line-and-scatter/) alongside the [.scatter() API reference](https://plotly.com/python-api-reference/generated/plotly.express.scatter.html) to create scatter plot that looks like the chart above.

*Hint*: Use the size, hover\_name and color parameters in .scatter(). To scale the y-axis, call .update\_layout() and specify that the y-axis should be on a log-scale like so: yaxis=dict(type='log')

.

.

..

…

..

.

.

**Solution: Create a scatter plot with Plotly**

First, we need to work out the number of apps in each category (similar to what we did previously).

1. cat\_number = df\_apps\_clean.groupby('Category').agg({'App': pd.Series.count})

Then we can use .merge() and combine the two DataFrames.

1. cat\_merged\_df = pd.merge(cat\_number, category\_installs, on='Category', how="inner")
2. print(f'The dimensions of the DataFrame are: {cat\_merged\_df.shape}')
3. cat\_merged\_df.sort\_values('Installs', ascending=False)

Now we can create the chart. Note that we can pass in an entire DataFrame and specify which columns should be used for the x and y by column name.

1. scatter = px.scatter(cat\_merged\_df, # data
2. x='App', # column name
3. y='Installs',
4. title='Category Concentration',
5. size='App',
6. hover\_name=cat\_merged\_df.index,
7. color='Installs')
9. scatter.update\_layout(xaxis\_title="Number of Apps (Lower=More Concentrated)",
10. yaxis\_title="Installs",
11. yaxis=dict(type='log'))
13. scatter.show()

What we see is that the categories like Family, Tools, and Game have many different apps sharing a high number of downloads. But for the categories like video players and entertainment, all the downloads are concentrated in very few apps.

