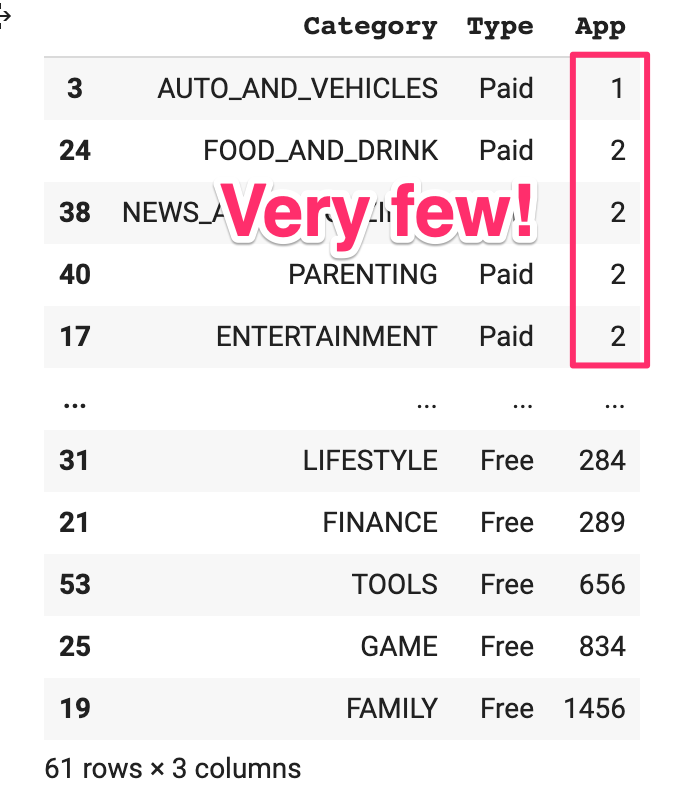
**Grouped Bar Charts and Box Plots with Plotly**

Now that we’ve looked at the total number of apps per category and the total number of apps per genre, let’s see what the split is between free and paid apps.

1. df\_apps\_clean.Type.value\_counts()

We see that the majority of apps are free on the Google Play Store. But perhaps some categories have more paid apps than others. Let’s investigate. We can group our data first by Category and then by Type. Then we can add up the number of apps per each type. Using as\_index=False we push all the data into columns rather than end up with our Categories as the index.

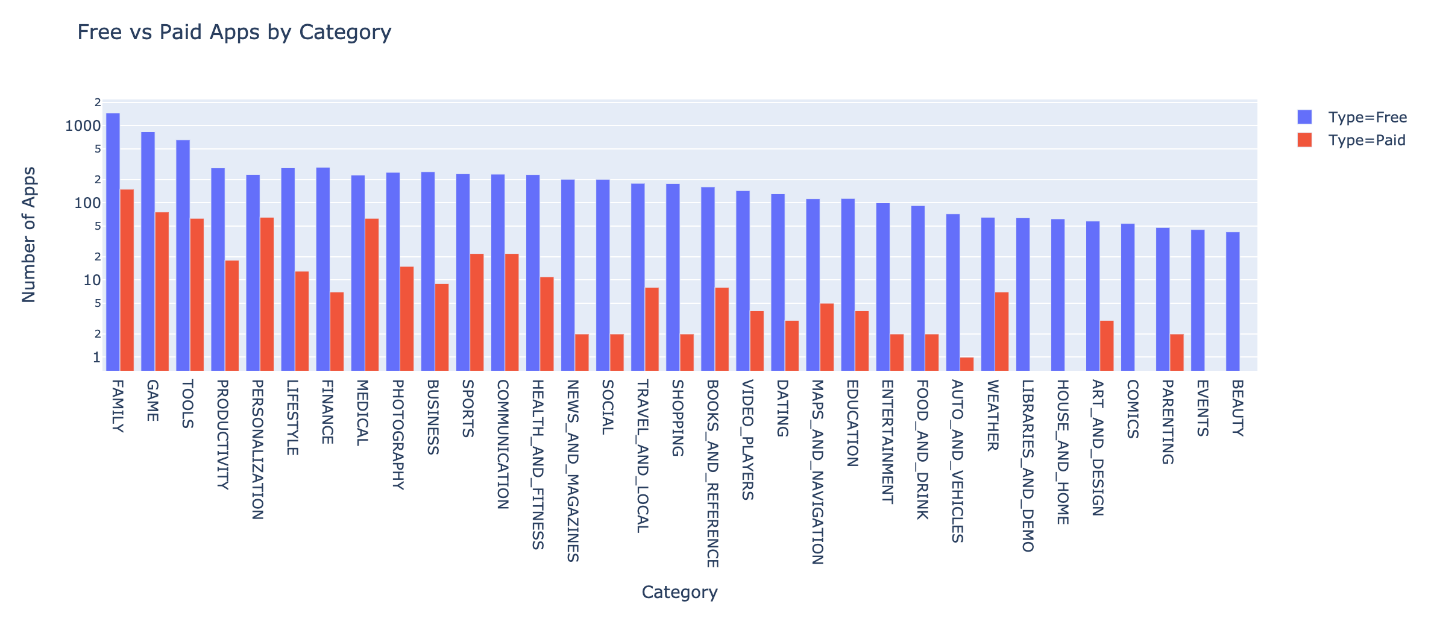
1. df\_free\_vs\_paid = df\_apps\_clean.groupby(["Category", "Type"], as\_index=False).agg({'App': pd.Series.count})
2. df\_free\_vs\_paid.head()



Unsurprisingly the biggest categories have the most paid apps. However, there might be some patterns if we put the numbers of a graph!

Challenge

Use the plotly express bar [chart examples](https://plotly.com/python/bar-charts/#bar-chart-with-sorted-or-ordered-categories) and the [.bar() API reference](https://plotly.com/python-api-reference/generated/plotly.express.bar.html#plotly.express.bar) to create this bar chart:



You'll want to use the df\_free\_vs\_paid DataFrame that you created above that has the total number of free and paid apps per category.

See if you can figure out how to get the look above by changing the categoryorder to 'total descending' as outlined in the documentation [here](https://plotly.com/python/categorical-axes/#automatically-sorting-categories-by-name-or-total-value).

.

.

..

…

..

.

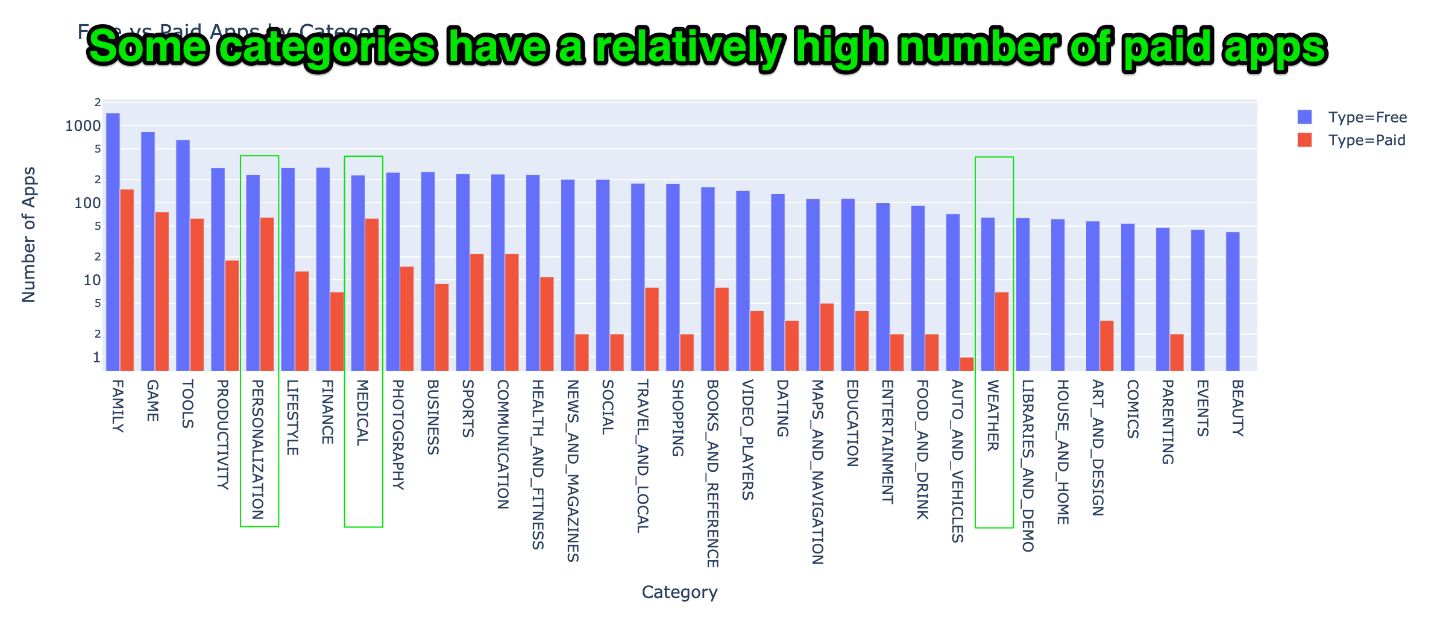
.

**Solution: Contrasting Free vs. Paid Apps per Category**

The key is using the color and barmode parameters for the .bar() method. To get a particular order, you can pass a dictionary to the axis parameter in .update\_layout().

1. g\_bar = px.bar(df\_free\_vs\_paid,
2. x='Category',
3. y='App',
4. title='Free vs Paid Apps by Category',
5. color='Type',
6. barmode='group')
8. g\_bar.update\_layout(xaxis\_title='Category',
9. yaxis\_title='Number of Apps',
10. xaxis={'categoryorder':'total descending'},
11. yaxis=dict(type='log'))
13. g\_bar.show()

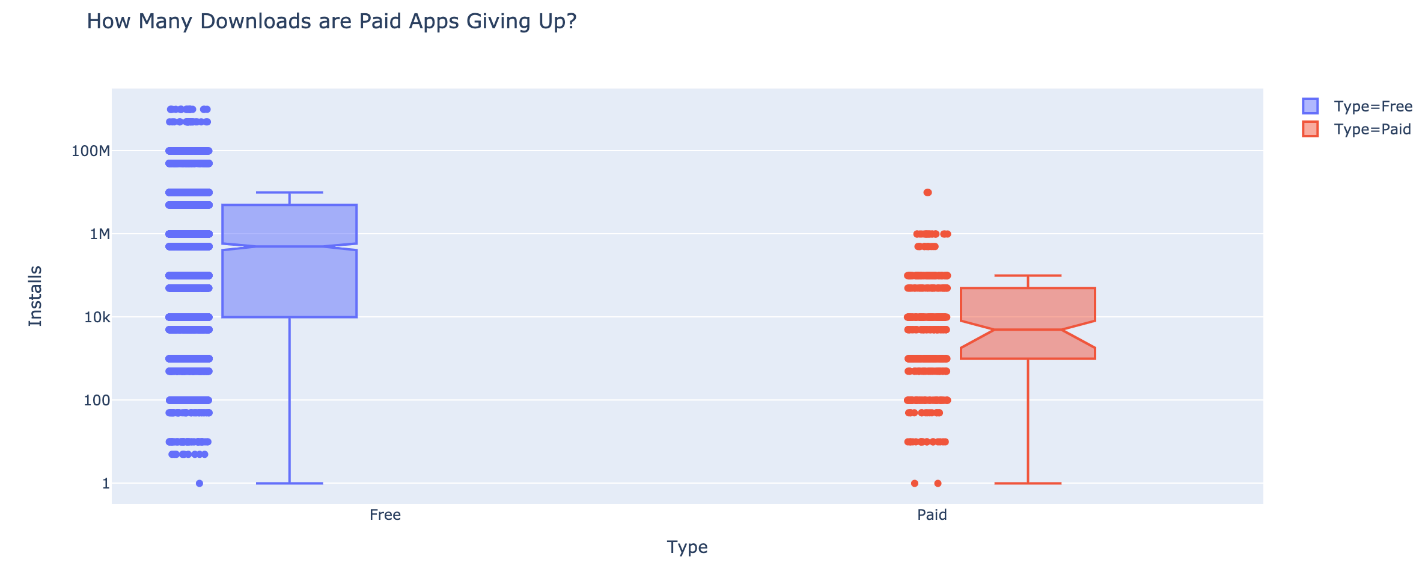
What we see is that while there are very few paid apps on the Google Play Store, some categories have relatively more paid apps than others, including Personalization, Medical and Weather. So, depending on the category you are targeting, it might make sense to release a paid-for app.



But this leads to many more questions:

* How much should you charge? What are other apps charging in that category?
* How much revenue could you make?
* And how many downloads are you potentially giving up because your app is paid?

Let’s try and answer these questions with some Box plots. Box plots show us some handy descriptive statistics in a graph - things like the median value, the maximum value, the minimum value, and some quartiles. Here’s what we’re after:



But how do we get there? This is your challenge.

Challenge

Create a box plot that shows the number of Installs for free versus paid apps. How does the median number of installations compare? Is the difference large or small?

Use the [Box Plots Guide](https://plotly.com/python/box-plots/) and the [.box API reference](https://plotly.com/python-api-reference/generated/plotly.express.box.html) to create the chart above.

.

.

..

…

..

.

.

**Solution: Create Box Plots for the Number of Installs**

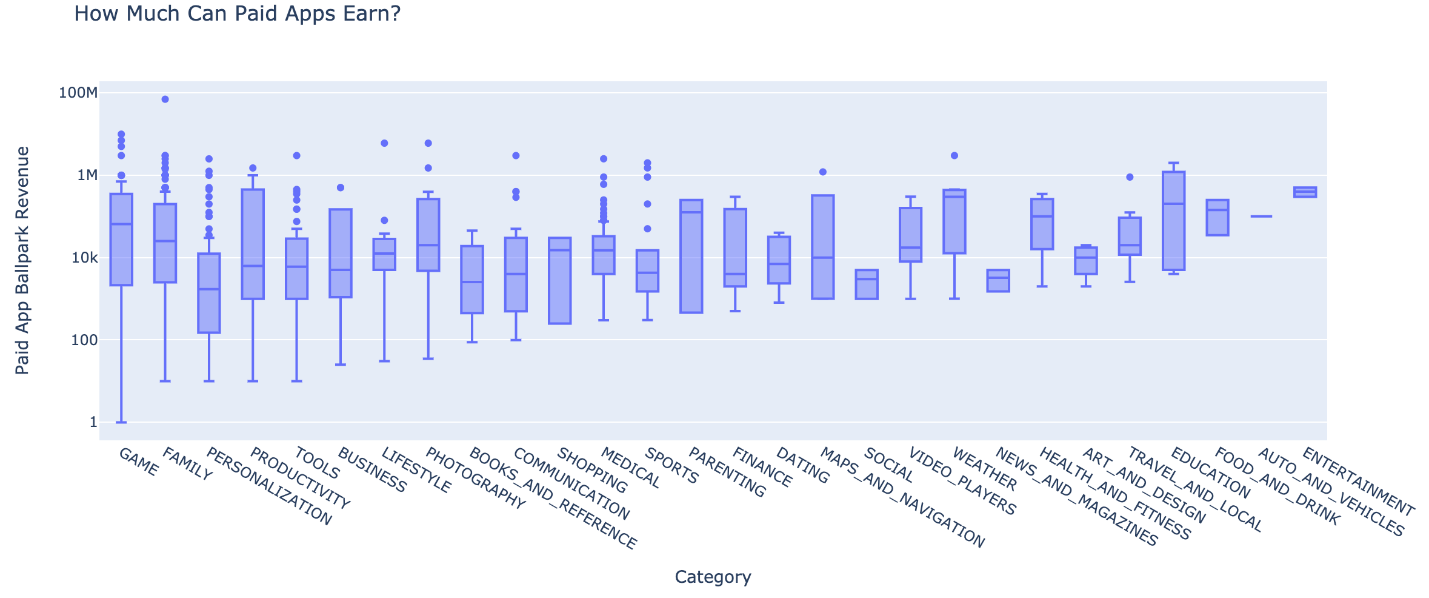
From the hover text in the chart, we see that the median number of downloads for free apps is 500,000, while the median number of downloads for paid apps is around 5,000! This is massively lower.

1. box = px.box(df\_apps\_clean,
2. y='Installs',
3. x='Type',
4. color='Type',
5. notched=True,
6. points='all',
7. title='How Many Downloads are Paid Apps Giving Up?')
9. box.update\_layout(yaxis=dict(type='log'))
11. box.show()

But does this mean we should give up on selling a paid app? Let’s see how much revenue we would estimate per category.

Challenge

See if you can generate the chart below:



Looking at the hover text, how much does the median app earn in the Tools category? [If developing an Android app costs $30,000 or thereabouts](http://howmuchtomakeanapp.com/), does the average photography app recoup its development costs?

Hint: I've used 'min ascending' to sort the categories.

.

.

..

…

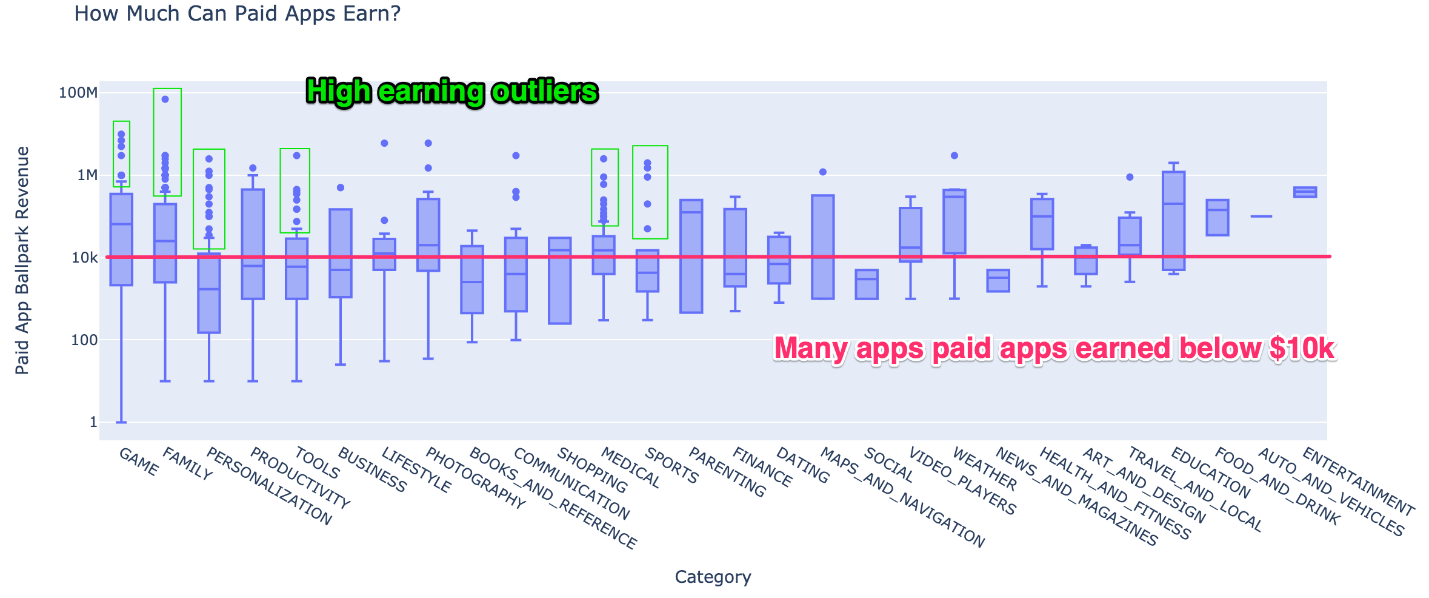
..

.

.

**Solution: App Revenue by Category**

If an Android app costs $30,000 to develop, then the average app in very few categories would cover that development cost. The median paid photography app earned about $20,000. Many more app’s revenues were even lower - meaning they would need other sources of revenue like advertising or in-app purchases to make up for their development costs. However, certain app categories seem to contain a large number of outliers that have much higher (estimated) revenue - for example in Medical, Personalisation, Tools, Game, and Family.



So, if you were to list a paid app, how should you price it? To help you decide we can look at how your competitors in the same category price their apps.

1. df\_paid\_apps = df\_apps\_clean[df\_apps\_clean['Type'] == 'Paid']
2. box = px.box(df\_paid\_apps,
3. x='Category',
4. y='Revenue\_Estimate',
5. title='How Much Can Paid Apps Earn?')
7. box.update\_layout(xaxis\_title='Category',
8. yaxis\_title='Paid App Ballpark Revenue',
9. xaxis={'categoryorder':'min ascending'},
10. yaxis=dict(type='log'))

13. box.show()

Challenge

What is the median price for a paid app? Then compare pricing by category by creating another box plot. But this time examine the prices (instead of the revenue estimates) of the paid apps. I recommend using {categoryorder':'max descending'} to sort the categories.

.

.

..

…

..

.

.

**Solution: App Pricing by Category**

The median price for an Android app is $2.99.

1. df\_paid\_apps.Price.median()

However, some categories have higher median prices than others. This time we see that Medical apps have the most expensive apps as well as a median price of $5.49. In contrast, Personalisation apps are quite cheap on average at $1.49. Other categories which higher median prices are Business ($4.99) and Dating ($6.99). It seems like customers who shop in these categories are not so concerned about paying a bit extra for their apps.

1. box = px.box(df\_paid\_apps,
2. x='Category',
3. y="Price",
4. title='Price per Category')
6. box.update\_layout(xaxis\_title='Category',
7. yaxis\_title='Paid App Price',
8. xaxis={'categoryorder':'max descending'},
9. yaxis=dict(type='log'))
11. box.show()

