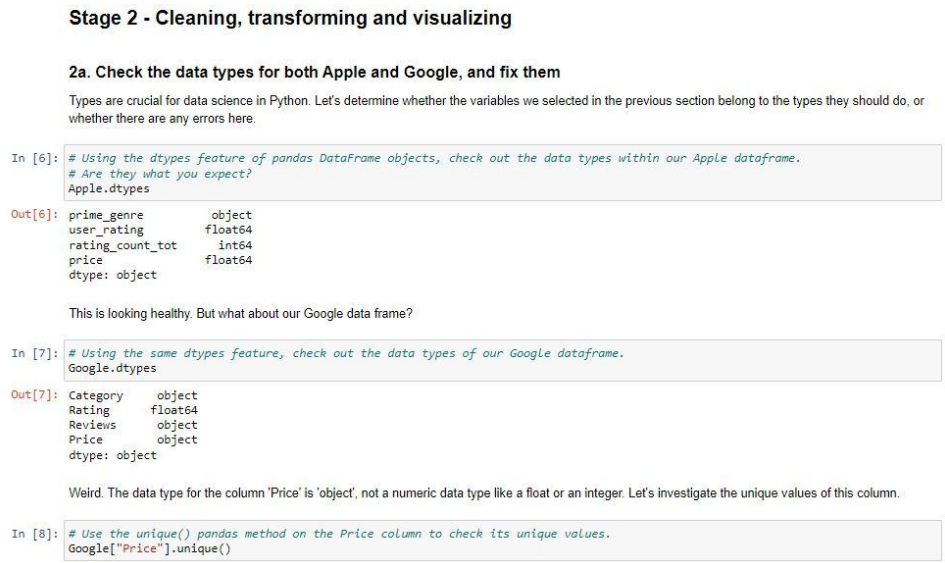
Apps Project

Emeka Nwosu

Stages 1 & 2: Sourcing & Cleaning Data

- Data came in 2 CSV Files (Apple and Google)
- Necessary columns: Category, Rating, Reviews, Price
- Converted price strings to numeric (removed “$”)
- Appended the two datasets rowise



Stage 2 - Cleaning, transforming and visualizing

2a. Check the data types for both Apple and Google, and fix them

Types are crucial for data science in Python. Let's determine whether the variables we selected in the previous section belong to the types they should do, or whether there are any errors here.

In [6]:
```
# Using the dtypes feature of pandas DataFrame objects, check out the data types within our Apple dataframe.
# Are they what you expect?
Apple.dtypes
```

Out[6]:
```
prime_genre          object
user_rating         float64
rating_count_tot      int64
price               float64
dtype: object
```

This is looking healthy. But what about our Google data frame?

In [7]:
```
# Using the same dtypes feature, check out the data types of our Google dataframe.
Google.dtypes
```

Out[7]:
```
Category     object
Rating      float64
Reviews      object
Price        object
dtype: object
```

Weird. The data type for the column 'Price' is 'object', not a numeric data type like a float or an integer. Let's investigate the unique values of this column.

In [8]:
```
# Use the unique() pandas method on the Price column to check its unique values.
Google["Price"].unique()
```

Modeling

- **H_{null}**: the observed difference in the mean rating of Apple Store and Google Play apps is due to chance (and thus not due to the platform)
- **$H_{alternative}$**: the observed difference in the average ratings of apple and google users is not due to chance (and is actually due to platform)

Stage 3 - Modelling

3a. Hypothesis formulation

Our **Null hypothesis** is just:

H_{null}: the observed difference in the mean rating of Apple Store and Google Play apps is due to chance (and thus not due to the platform).

The more interesting hypothesis is called the **Alternate hypothesis**:

$H_{alternative}$: the observed difference in the average ratings of apple and google users is not due to chance (and is actually due to platform)

We're also going to pick a **significance level** of 0.05.

3b. Getting the distribution of the data

Now that the hypotheses and significance level are defined, we can select a statistical test to determine which hypothesis to accept.

There are many different statistical tests, all with different assumptions. You'll generate an excellent judgement about when to use which statistical tests over the Data Science Career Track course. But in general, one of the most important things to determine is the **distribution of the data**.

In [68]:
```
# Create a subset of the column 'Rating' by the different platforms.
# Call the subsets 'apple' and 'google'
apple = df[df["platform"] == "apple"]["Rating"]
google = df[df["platform"] == "google"]["Rating"]
```

In [69]:
```
# Using the stats.normaltest() method, get an indication of whether the apple data are normally distributed
# Save the result in a variable called apple_normal, and print it out
# Since the null hypothesis of the normaltest() is that the data is normally distributed, the lower the p-value in the result of
apple_normal = stats.normaltest(apple)
print(apple_normal)
```

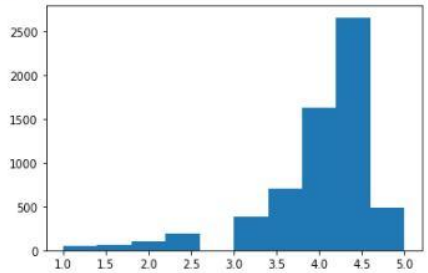
```
NormaltestResult(statistic=1778.9974234584017, pvalue=0.0)
```

Visualization

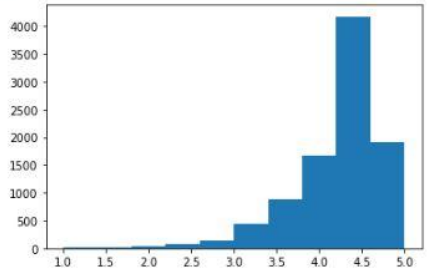
First double plot is the histogram of reviews from each platform

Second single plot is the histogram of the difference in scores

```
In [71]: # Create a histogram of the apple reviews distribution
         # You'll use the plt.hist() method here, and pass your apple data to it
         histoApple = plt.hist(apple)
```




```
In [72]: # Create a histogram of the google data
         histoGoogle = plt.hist(google)
```



Conclusion

When sampled 10,000 times, the difference in means between the two groups is normally distributed around 0

```
In [104]: # Make a variable called 'histo', and assign to it

          histo = plt.hist(difference) # i changed this to n
```