**Project Report**

Submission by: Julie Lanigan

Date: 26/09/2021

**GitHub URL**

[Julie16kk/UCDPA\_JulieLanigan (github.com)](https://github.com/Julie16kk/UCDPA_JulieLanigan)

**Abstract**

The below project details the use of financial data with Python to show my understanding of how coding can be complete to understand and manipulate large datasets.

**Introduction**

I chose Amazon and Apple as the data use set due to their similarities over the past few years as market leaders but to also show the relevance of monitoring stock prices over a set period of time.

**Dataset**

Three data sets were chosen for this project. The first two data sets chosen were Apple & Amazon stock prices, the third data set chosen was Quarterly Market Information Indices. The Apple & Amazon stock price data sets were chosen firstly due to their financial nature but secondly so they could be compared to provide an appropriate analysis of their stock prices. The Quarterly Market Information Indices dataset was chosen as it was an API dataset but also as it was financial related.

The Apple and Amazon stock price data sets were extracted from Kaggle.com (see links below) with the Quarterly Market Information Indices data extracted from data.gov.ie (see link below). The Apple data set shows their stock prices between 2015 – 2020 including closing price, highest / lowest price of the day, opening price and volume of stock traded. The Amazon dataset shows their stock prices between 1997 – 2020 including closing price, highest / lowest price of the day, opening price and volume of stock traded. The Quarterly Market Information Indices dataset shows the House Price Index, Interest Rate, Consumer Price Index and Earnings.

[Quarterly Market Information Indices - Datasets - data.gov.ie](https://data.gov.ie/dataset/quarterly-market-information-indices?package_type=dataset)

[Apple Stock Prices (2015-2020) | Kaggle](https://www.kaggle.com/suyashlakhani/apple-stock-prices-20152020)

[Amazon Stock Price 1997 to 2020 | Kaggle](https://www.kaggle.com/salmanfaroz/amazon-stock-price-1997-to-2020)

**Implementation Process**

The below steps detail the process involved in working with the above datasets in Python and as outlined in the assignment How To Guide. Please note the below steps align to the dataset built in Python and in order to be fully understood should be read fully in Python alongside the data.

**Section 1:**

Real World Scenario – as detailed above three real world scenarios were used with the links included and reference to their source.

**Step 1:**

Create new project folder and Python file - UCDPA\_JulieLanigan

Import required libraries, pandas/seaborn/Numpy/Matplotlib

**Section 2:**

Importing Data – the below steps detail the importing of three data sets (as detailed above) into Python including 2 CSV files, an API dataset and using a Pandas DataFrame.

**Step 2:**

Importing:

Download .csv files to Data folder within project file

Import files to project using pandas

**Step 3:**

API:

Import data from an online API

Data used is Quarterly Market Information Indices from data.gov.ie

The requests.get(api\_path).text pulls the data from the API we used.

The json.loads() function then parses the data into a JSON format.

**Section 3: Analysing Data:**

The below steps detail the cleansing of the Apple & Amazon data sets. Once the datasets were cleansed the data was then sorted, indexed, grouped, missing values were replaced, duplicates were dropped, looping with iterrows was complete and the two dataframes were merged.

**Step 4:**

Analysing:

The below functions are used to investigate the datasets

First to look at the aaplData dataset

the .head function returns the first 5 rows of the dataset

the .tail function returns the last 5 rows

The.info function shows the datatypes of each column

The .describe shows statistics relating to each column

**Step 5:**

The same functions as applied above are applied to investigate the other dataset amazonData

**Step 6:**

Cleaning:

In order to clean the datasets, first the unnecessary columns are dropped.

The columns kept (date, open, high, low, close, adjusted close) are shared by both datasets and can then be used to compare the datasets later using charts.

The .head and .info functions show that the Apple dataset has results 15/05/1997 to 22/05/2020 whereas the Amazon dataset has results from 27/05/2015 to 31/07/2020.

The code removes all rows in both datasets before 01/01/2016 and 22/05/2020.

This will ensure that there is data in both datasets for the selected period.

**Step 7:**

The date column in the Apple dataset contains hours/mins/secs which aren't needed when comparing dates.

The .to\_datetime function is used to remove this and display the date column as yyyy-mm-dd as with the Amazon dataset.

**Step 8:**

The code is used to rename the columns in the Apple dataset.

These have been renamed to be consistent with the Amazon dataset it will be compared with.

Rounding values

The columns in the chosen dataframes use a high number of decimal places.

The code is used to round the numbers within the dataframes to 2 decimal places.

**Step 9:**

Sorting/Indexing:

The columns in the datafile were already sorted by date.

The code is an example of how the datasets could be sorted by date if the .csv file used was not sorted correctly/had rows in the incorrect order.

**Step 10:**

Missing values:

The isnull function is used to check if any of the columns have null values.

In this case there weren't any.

If there were a large number of null values in a column the entire column could be dropped.

If there were a small number of null values we could choose to remove the rows with null values.

**Step 11:**

Checking for duplicates:

The code is used to check if there are any duplicated rows within the datasets.

The print function then gives an index of any duplicated rows which can be removed.

**Step 12:**

Grouping:

The code can be used to group the data by a specified column.

The .first function then prints the first entry in each group.

In this case the date column has unique values so grouping is not needed.

**Step 13:**

Merging dataframes:

The code creates a new column in each dataset to note which stock the row relates to. This is important to ensure that when the data is merged we know which data relates to Apple and which relates to Amazon.

The dataframes are then merged into one dataframe (stockData) using the concatenate function to stack one on top of the other.

**Step 14:**

Looping/iterrows:

The code is an example of a for loop.

This for loop looks at the merged dataframe we created (stockData) and gives a tuple of the column name and series for each column in the dataframe.

**Section 4: Python:**

**Step 15:**

Numpy:

NumPy is a Python library used for working with arrays.

The code below uses Numpy to create an array called stockArray from the chosen elements of our dataframe.

In the case the new array is made up of the data from the Open, High and Low columns of the dataframe.

The .shape function is then used to view the shape of the array.

The .max function for example can be used to display the largest result in each of the array columns.

**Step 16:**

Lists:

A list is a dynamic type of data structure.

At times it may be easier to work with our data in a list structure rather than a dataframe or array.

The code gives an example of how we can convert our dataframe to a new list, declared as stockList.

We could use the append() or remove() functions to add or remove items from the list if needed.

**Step 17:**

Define a custom function to create reusable code:

I used the define function is used to create custom reusable code.

The maximum and minimum opening stock price for both apple and amazon are defined as separate functions which can be called upon to generate the result without needing the code.

These functions can then be called e.g. appleMaxOpen() to run the custom code and return the required result.

**Step 18:**

Creating columns for month and year:

The code is used to create a Year and Month column for the datasets.

This will then allow the datasets to be used to create more views/comparisons.

**Results**

The below details the steps taken to create charts for visualisation and screenshots of the charts.

**Section 5: Visualisation:**

**Step 19:**

Charts:

Below a series of charts are used to investigate the datasets further

The first eight charts look at the open and closing stock values of both Apple and Amazon stock over the time period as a line graph and dist plot

Apple:

Chart

Description automatically generated with medium confidence

Chart

Description automatically generated with medium confidence

Amazon:

Chart

Description automatically generated with medium confidence

Chart

Description automatically generated with medium confidence

Apple:

Chart, histogram

Description automatically generated

Amazon:

Chart, histogram

Description automatically generated

Apple:

Chart, histogram

Description automatically generated

Amazon:

Chart, histogram

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing chart

Description automatically generated

A picture containing application

Description automatically generated

**Insights**

**Section 6: Generate valuable insights**

**Insight 1:**

The first thing which is notable from the charts is that both stocks show an increase over the period being looked at (2016-2020). This result was expected as both companies have shown continuing growth over this period in terms of market share.

**Insight 2:**

We can also see from the charts that both stock's opening and closing prices are almost identical over the period. Again this is an expected result. The growth seen in the stock prices is over a period of years, whereas the opening and closing values for each day would be expected to remain very similar with relatively minor daily rises and falls.

**Insight 3:**

When examining the opening and closing values of the stock prices over time, we can see that despite the vast differences in the range of stock prices for each company (Apple 90-327, Amazon 478-2500), the graphs share very similar trends over the period.

As both companies operate within the same sector, we would expect that they would be affected similarly by external market factors and show similar trends in stock price changes.

This can be seen in the severe fall in both stock prices shown around the start of 2020. This fall coincides with the start of the Covid-19 pandemic. Similarly both companies have shown significant growth since this period.

**Insight 4:**

When plotting the opening stock prices for both companies and using the Month as hue we can see some interesting data.

The Apple opening stock price shows a good mix with low and high opening stock prices occurring during all months.

The Amazon stock price however, shows an interesting trend with the majority of its highest opening stock prices occurring during the start-middle months of the year with almost no opening values over approx. 1800 occurring in the final months of the year.

**References**

N/A