**Project 2  
Stock Market influence of COVID 19**

**Authors**

* Mark Rotil
* Nikita Case
* Eric Lee
* Hadi Hanif
* Murat Cagri KOC

**Introduction**

Conventional thinking tells us that when calamities occur, the stock market reacts poorly. These calamities result from a variety of factors ranging from inaccurate economic modelling to climate change.

The Covid-19 pandemic however, can be classified as a black swan event. News cycles inundate us with recovery rates, death rates, and lockdowns. The goal of this study is to determine just how sensitive the stock market is to these events.

We have identified 9 stocks across following industries:

* Hospitality
* Aviation
* Entertainment
* Technology
* Telecommunications

We will track the prices of these stocks from the 1st of January 2020 to the 4th of December 2020 and then compare it to the previous year.

Then we identify important events related to the pandemic (E.G. the first case of Covid-19 in Canada) and, look to see if there is any corresponding fluctuations in price.

Our hypothesis is that fluctuations of stock prices are strongly correlated with key events in Covid-19 such as lockdowns and that the correlation has weakened as time passed and that this applies across all sectors of the stock market.

**Etraction Process:**

Our first step is to etract the following information from Yahoo Finance in the form of CSVs.

Once that is done, we proceed to prepare the necessary initial import parameters into Jupyter Notebooks:

**Step 1**



Once we have done that we will then import the CSV into the notebook and check to see if the data looks accurate:

**Step 2**



Here we see the data is accurate represented from the CSV.

**Step 2**

We then proceed to scrape the dates of key events from a news source.

****

**Transform**

Now that the data has been loaded we begin the Transformation process. 

We drop the “CLOSE” column from the dataset because we will be using use the Adjusted\_Close on a day to day basis. The difference between the two is that the Adjusted\_Close factors in the price after dividend issuances.

We also transform the convert the format in the Date column from tet to a data type format because it will make the querying process much easier for us to do. We also do a check to determine how the data will be represented.

This process is repeated 9 times, once for each stock.

**Step 3**

**Then after etracting the news we begin to cleanup the news sources by making it into a dictionary.**

****

**Step 4**

Then we convert the dates into a dataframe with ‘Date’ and ‘News’.



Step 5

After converting the dates into a dataframe we then need to replace the dates from a tet to a datetime format. This was done manually to avoid any errors.



**Step 6**

We then apply a filtration of key words. In this case, we chose four words: *lockdown, shutdown, Toronto, deaths*.

This reduced the list of 129 news articles to ~19.

**Loading**

When done with the cleansing, we proceed to the loading of the data into ElephantSQL. ElephantSQL allows us to aggregate the data into single SQL table.

**Step 7**

We first create an engine that allows us eport out Transformed files onto ElephantSQL.



It is also important to note that ElephantSQl limits the number of users that can concurrently function.

We then do a check to see if all the tables have been be placed correctly. To do that, we do a table name check:



Here we see that the tables are correctly showing up.

**Limitations:**

The first challenge we encountered is that there user limitations for ElephantSQL. Our team consisted of 5 users however, we were unable to have 2 users fetch the information for ElephantSQL. Our attempts to use other applications for loading purposes, such as MongoDB but we encountered similar issues. We had to constantly keep track of which users were on.

Another challenge is that in our transformation process is that we had multiple iterations of deletions and uplods of our tables. This is due in part to the fact that when we made modifications we would have to apply them across 9 files, one for each stock. In another cycle we might have merged all the files before putting up on ElephantSQL or restricting the number of stocks.

**Following Steps:**

After doing this, we would then visualize the tables as on a line graph. We would plot all the stocks across a single graph with time and stock prices representing the horizontal and vertical lines respectively. We would then apply dates of key events to determine whether or not decreases in prices correspond to the dates.