STOCKS WITH THE HIGHEST GAIN: 1996 TO 2020

Project By:
Mason Barrios
Joaquin Flores
Jasmine Gomez
Dylan Mora

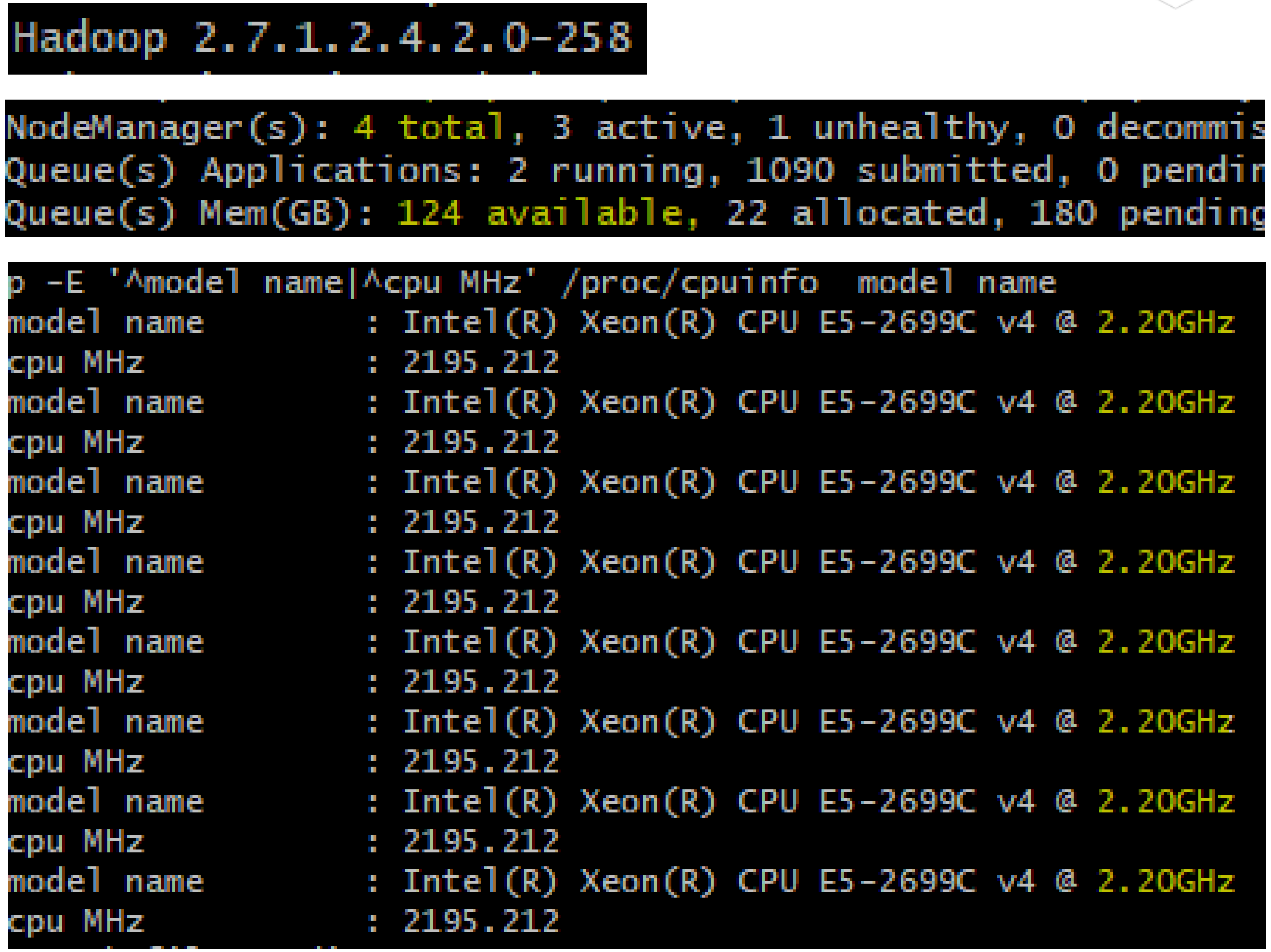
INTRODUCTION

- Stock Data from 1996 to 2020
- Dataset (12GB) from Kaggle.com provided by Dip Modi
- 104,123 individual stock tickers
- Pig utilized to analyze data
- HDFS used for storage

Link to Dataset (2GB Compressed, 12GB Unzipped): https://www.kaggle.com/aceofit/stockmarketdatafrom1996to2020

SPECIFICATIONS

- **Cluster Version:** Hadoop 2.7.1.2.4.2.0-258
- **Number of Nodes:** 4
- **Memory:** 124GB
- **CPU:** Intel(R) Xeon(R) E5-2699C v4
- **CPU Speed:** 2.20GHz



RETRIEVING THE DATA

```
Connecting to doc-14-1s-docs.googleusercontent.com|172.217.8.193|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-zip-compressed]
Saving to: “stock_data”

    [                                                                    <=>

2020-11-26 02:08:44 (132 MB/s) - “stock_data” saved [2469842861]
```

- WGET is used to download the data locally.
- Due to insufficient space, dataset is downloaded to temporary file systems (tmpfs /dev/shm).

MOVING THE DATA

```
-bash-4.1$ hdfs dfs -put Data dataproj/
-bash-4.1$ hdfs dfs -ls dataproj/
Found 2 items
drwxr-xr-x   - mbarrio5 hdfs          0 2020-11-26 02:18 dataproj/Data
-rw-r--r--   2 mbarrio5 hdfs    4730902 2020-11-26 02:17 dataproj/Tickers.xlsx
-bash-4.1$
```

- From the Linux machine, we move the data to HDFS via the –put command
- This function takes 1 hour and 10 minutes to complete

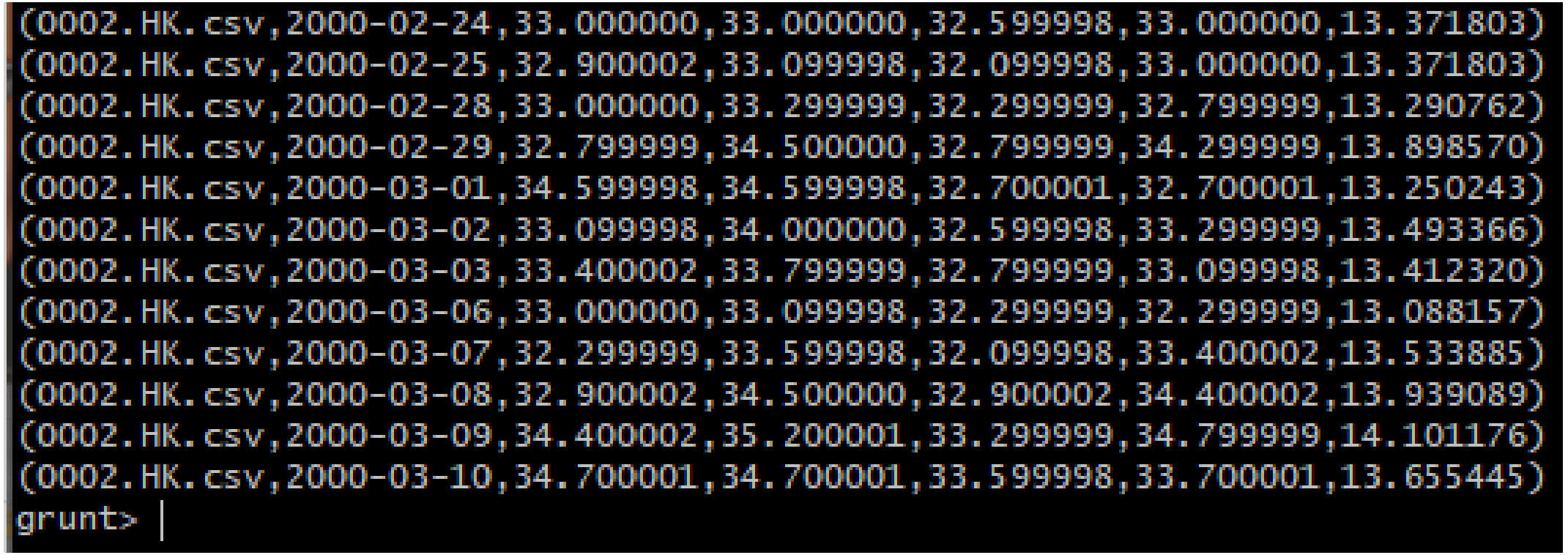
USING APACHE PIG


```
grunt> stock_data = LOAD '/user/mbarrio5/dataproj/Data/Data/0002.HK/0002.HK.csv' USING PigStor
age(',','-tagFile')
>> AS (date:chararray, open:chararray, high:chararray,
>> low:chararray, close:chararray, adjclose:chararray,
>> volume:chararray);
grunt> DESCRIBE stock_data;
stock_data: {date: chararray,open: chararray,high: chararray,low: chararray,close: chararray,a
djclose: chararray,volume: chararray}
grunt> stock_data_subset = limit stock_data 50;
grunt> DUMP stock_data_subset;
```

- In pig we create a sample script and test it against one stock ticker
- After we determine the schema, we modify the script to include all

  104,123 stock tickers

EXAMPLE OF TEST OUTPUT



FINALIZING PIG SCRIPT

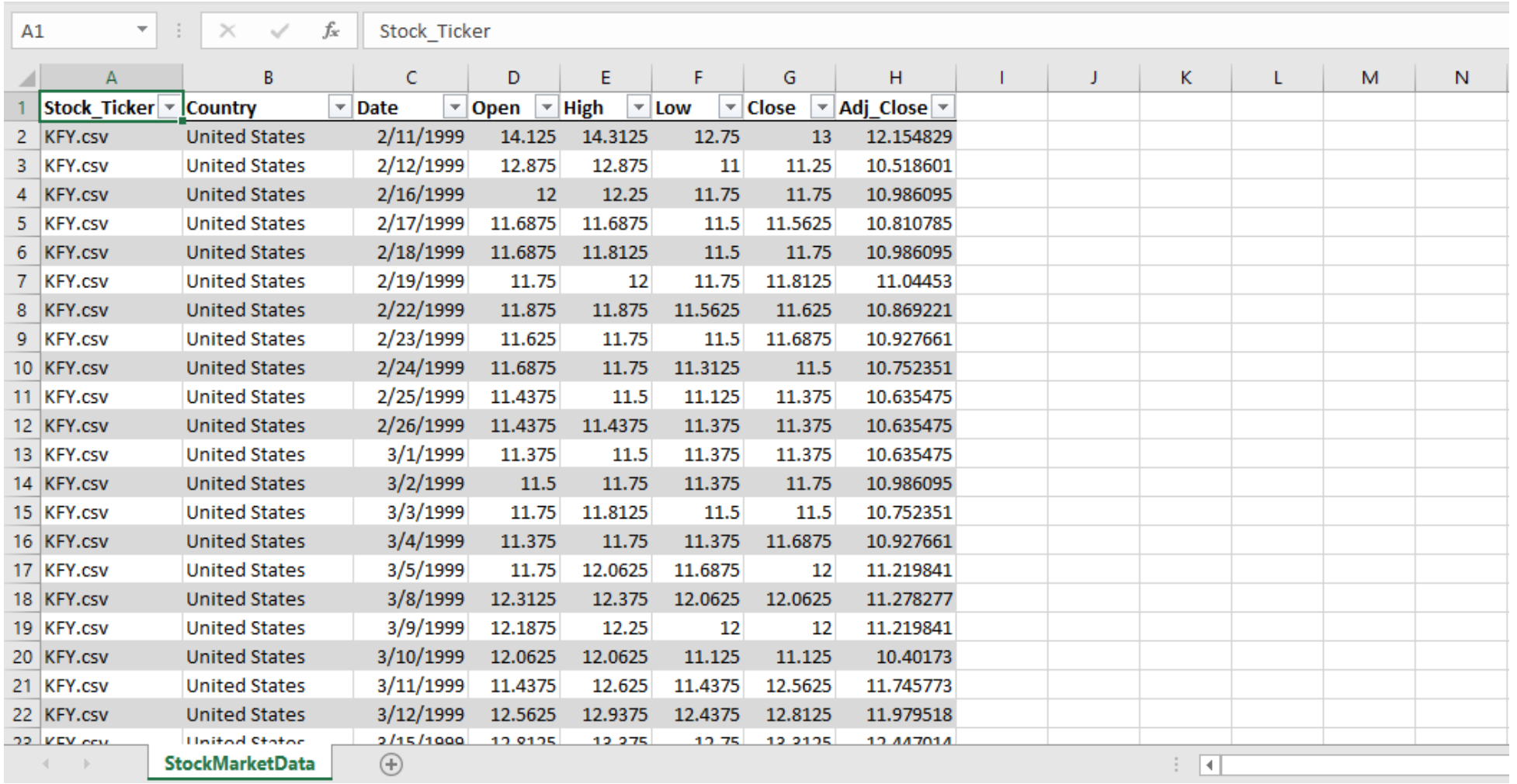
```
$ stock_data = LOAD '/user/mbarrio5/dataproj/Data/Data/*/*'
            USING PigStorage(',','-tagFile')
            AS (date:chararray, open:chararray, high:chararray,
            low:chararray, close:chararray, adjclose:chararray,
            volume:chararray);

$ stock_data_subset = stock_data;

$ store stock_data_subset into 'output/cmpldata' using PigStorage(',');
```

- Script is revised after testing and all 104,123 tickers are stored into a single file (~13GB, process takes ~27 minutes)

DATA IN EXCEL

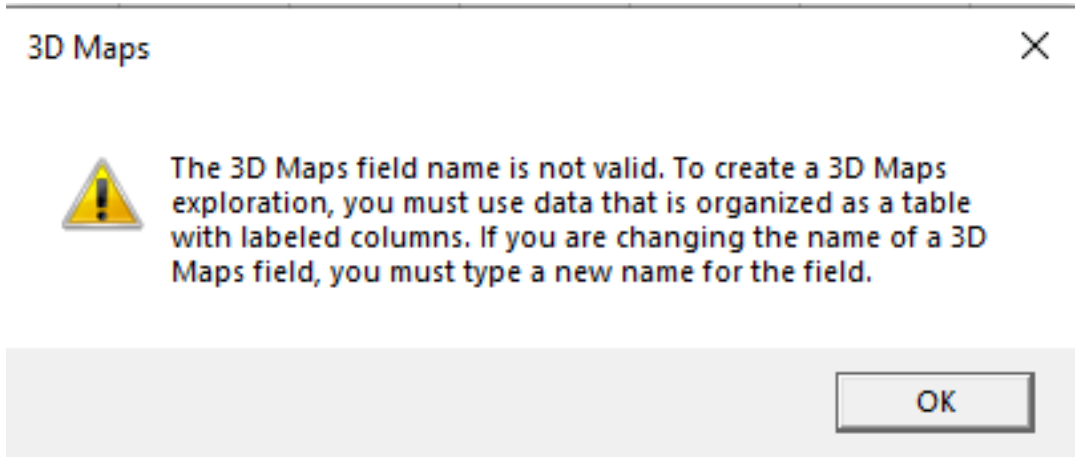


A1 | Stock_Ticker

	A	B	C	D	E	F	G	H
1	Stock_Ticker	Country	Date	Open	High	Low	Close	Adj_Close
2	KFY.csv	United States	2/11/1999	14.125	14.3125	12.75	13	12.154829
3	KFY.csv	United States	2/12/1999	12.875	12.875	11	11.25	10.518601
4	KFY.csv	United States	2/16/1999	12	12.25	11.75	11.75	10.986095
5	KFY.csv	United States	2/17/1999	11.6875	11.6875	11.5	11.5625	10.810785
6	KFY.csv	United States	2/18/1999	11.6875	11.8125	11.5	11.75	10.986095
7	KFY.csv	United States	2/19/1999	11.75	12	11.75	11.8125	11.04453
8	KFY.csv	United States	2/22/1999	11.875	11.875	11.5625	11.625	10.869221
9	KFY.csv	United States	2/23/1999	11.625	11.75	11.5	11.6875	10.927661
10	KFY.csv	United States	2/24/1999	11.6875	11.75	11.3125	11.5	10.752351
11	KFY.csv	United States	2/25/1999	11.4375	11.5	11.125	11.375	10.635475
12	KFY.csv	United States	2/26/1999	11.4375	11.4375	11.375	11.375	10.635475
13	KFY.csv	United States	3/1/1999	11.375	11.5	11.375	11.375	10.635475
14	KFY.csv	United States	3/2/1999	11.5	11.75	11.375	11.75	10.986095
15	KFY.csv	United States	3/3/1999	11.75	11.8125	11.5	11.5	10.752351
16	KFY.csv	United States	3/4/1999	11.375	11.75	11.375	11.6875	10.927661
17	KFY.csv	United States	3/5/1999	11.75	12.0625	11.6875	12	11.219841
18	KFY.csv	United States	3/8/1999	12.3125	12.375	12.0625	12.0625	11.278277
19	KFY.csv	United States	3/9/1999	12.1875	12.25	12	12	11.219841
20	KFY.csv	United States	3/10/1999	12.0625	12.0625	11.125	11.125	10.40173
21	KFY.csv	United States	3/11/1999	11.4375	12.625	11.4375	12.5625	11.745773
22	KFY.csv	United States	3/12/1999	12.5625	12.9375	12.4375	12.8125	11.979518

StockMarketData

ISSUES WITH THE PROJECT



Stock_Ticker	Country	Date	Open
KFY.csv	United States	2/11/1999	14.125
KFY.csv	United States	2/12/1999	12.875
KFY.csv	United States	2/16/1999	12
KFY.csv	United States	2/17/1999	11.6875
KFY.csv	United States	2/18/1999	11.6875
KFY.csv	United States	2/19/1999	11.75
KFY.csv	United States	2/22/1999	11.875
KFY.csv	United States	2/23/1999	11.625
KFY.csv	United States	2/24/1999	11.6875
KFY.csv	United States	2/25/1999	11.4375
KFY.csv	United States	2/26/1999	11.4375
KFY.csv	United States	3/1/1999	11.375
KFY.csv	United States	3/2/1999	11.5
KFY.csv	United States	3/3/1999	11.75
KFY.csv	United States	3/4/1999	11.375
KFY.csv	United States	3/5/1999	11.75
KFY.csv	United States	3/8/1999	12.3125

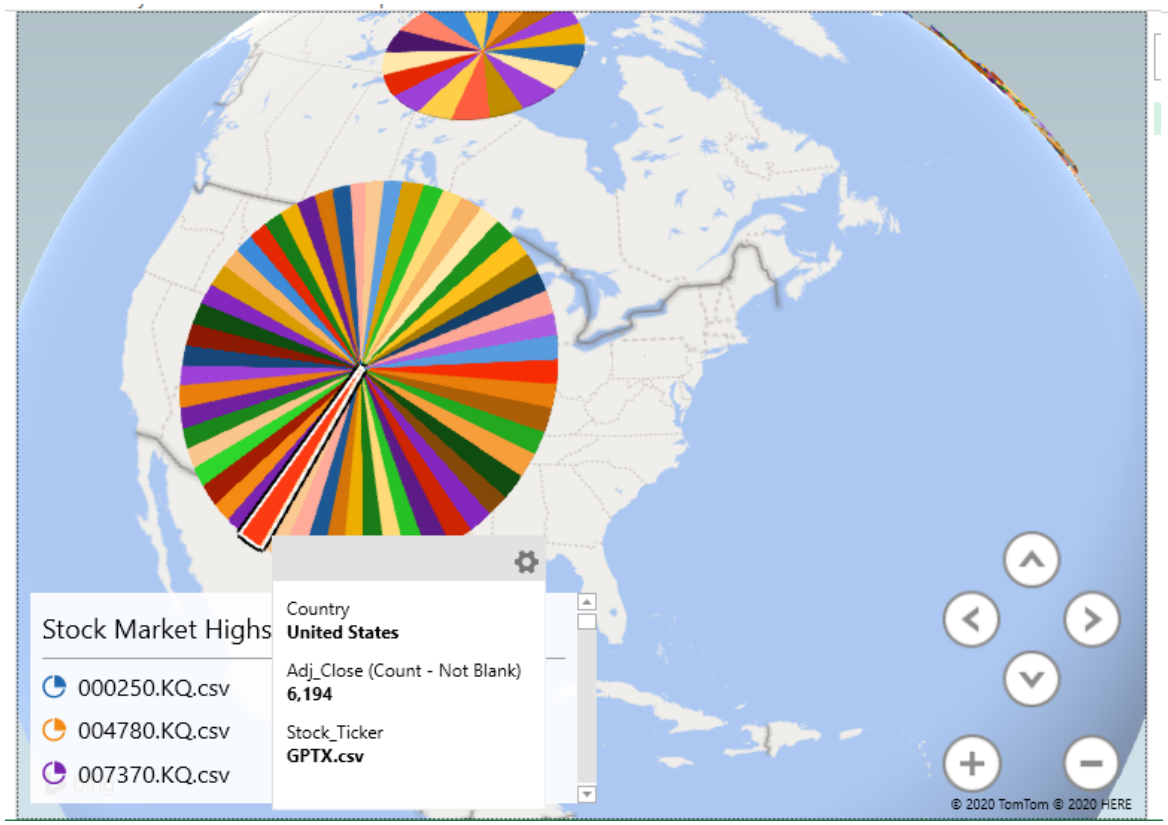
- Time Series (Excel Error)
- Excel did not accept "DATE" column
- We were unable to include a TIME dimension to the data visualization
- Some countries insufficient data, ex: Brazil

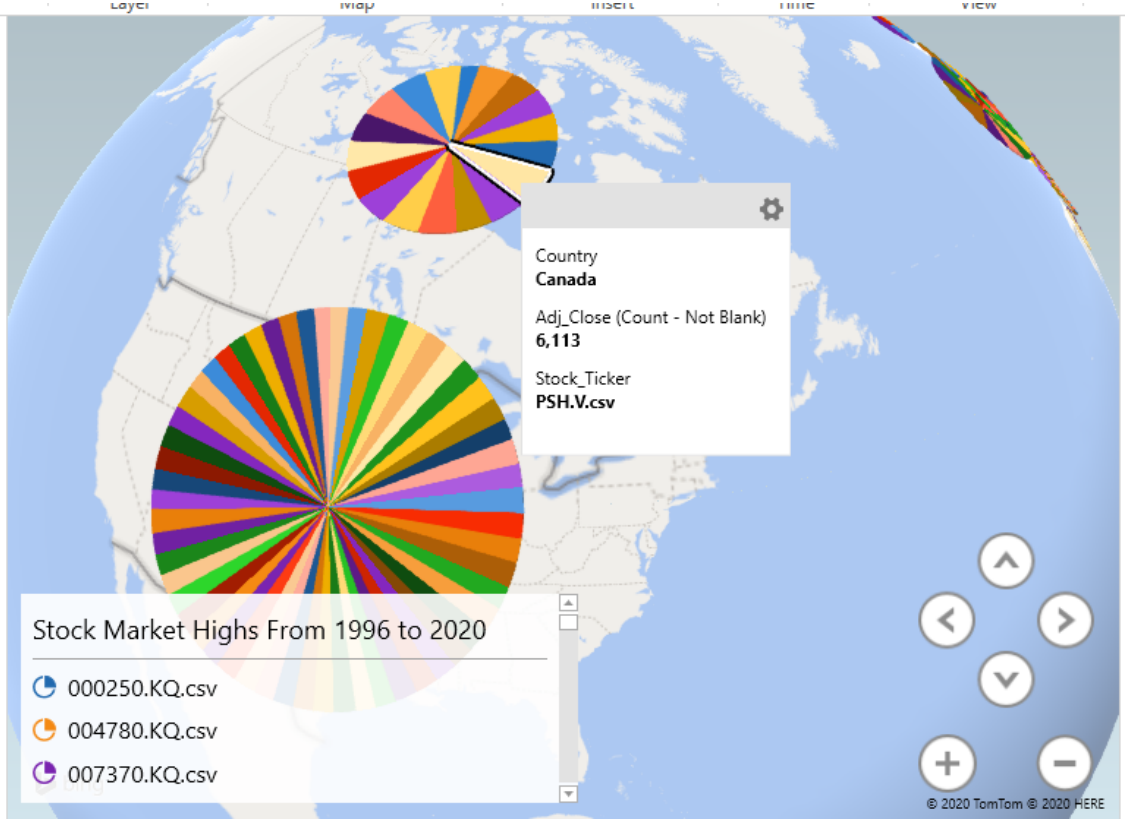
THE FINDINGS: GREATEST GAINS BY COUNTRY

- United States: **Δ** 6,194 GPTX (Global Payments Technologies)
- Canada: **Δ** 6,113 PSH.V (Petrolshale Inc.)
- France: **Δ** 60,080 LTA.PA (Altamir SCA)
- United Kingdom: **Δ** 6,278 JAM.L (JP Morgan)

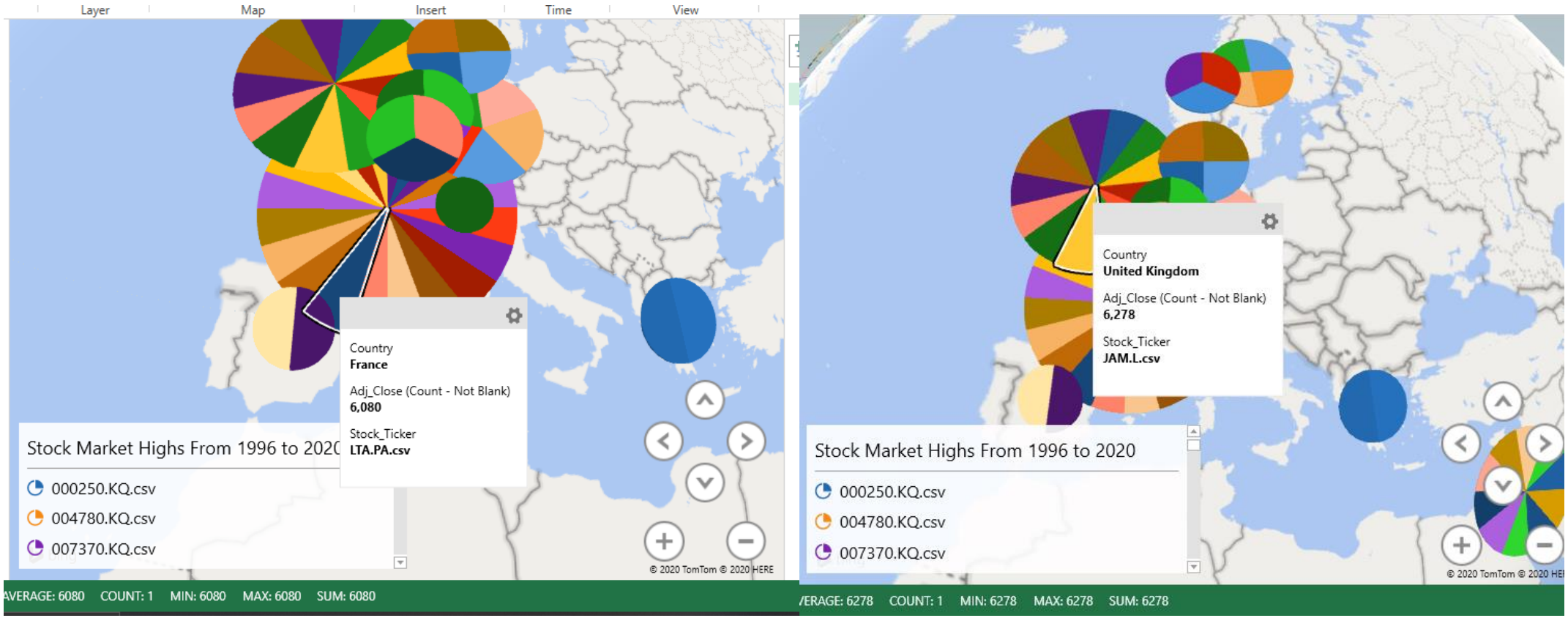
- NOTE: excluded other countries due to lack of data

GPTX (US) AND PSH (CAN)





LTA.PA (FRA) AND JAM.L (UK)



CONCLUSION

Out of the four countries sampled, financial institutions (or finance related) stock tickers saw the highest gains over the course of 24 years.

RELATED WORKS

- **Long short-term memory model (LSTM)** : LSTM is a technique subtle for processing and predicting important events. LSTM is an extended variant of RNN, a deep learning model that is good at processing time-series data.

- **Empirical Mode Decomposition (EMD)** : EMD is a signal analysis method; it can decompose a complex signal into a finite intrinsic mode function (IMF)