Google Flu Trend

Jingwen Zhang

December 23, 2020

1. Introduction

1.1 Background

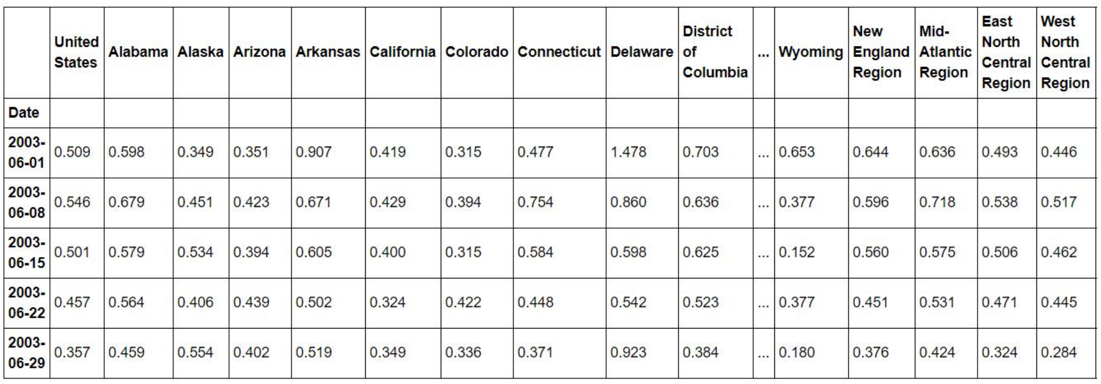
As we know, Google flu trend is a representative data analysis algorithm developed by Google engineers. The tower uses keywords searched by people to predict the flu situation in various regions. Its accuracy and timeliness far exceeds that of traditional methods, and it was once regarded as a classic application case in the field of big data. For a variety of reasons, though, the app is no longer available. But the data he left behind could be used for secondary analysis. This article will analyze data from 2003 to 2009 on influenza infections in different regions of the United States.

1.2 Problem

Part of the data was used to analyse the flu situation in Illinois and compare it with Idaho.The states with the initial letter 'M' were selected and their flu trends were plotted. Finally, draw a heat-map to see which states had the worst flu in which period.

1.3 Data describe

The data mainly includes the time of the flu, as well as the names of U.S. states. Some data are shown in the figure:



	United States	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	District of Columbia	...	Wyoming	New England Region	Mid-Atlantic Region	East North Central Region	West North Central Region
Date																
2003-06-01	0.509	0.598	0.349	0.351	0.907	0.419	0.315	0.477	1.478	0.703	...	0.653	0.644	0.636	0.493	0.446
2003-06-08	0.546	0.679	0.451	0.423	0.671	0.429	0.394	0.754	0.860	0.636	...	0.377	0.596	0.718	0.538	0.517
2003-06-15	0.501	0.579	0.534	0.394	0.605	0.400	0.315	0.584	0.598	0.625	...	0.152	0.560	0.575	0.506	0.462
2003-06-22	0.457	0.564	0.406	0.439	0.502	0.324	0.422	0.448	0.542	0.523	...	0.377	0.451	0.531	0.471	0.445
2003-06-29	0.357	0.459	0.554	0.402	0.519	0.349	0.336	0.371	0.923	0.384	...	0.180	0.376	0.424	0.324	0.284

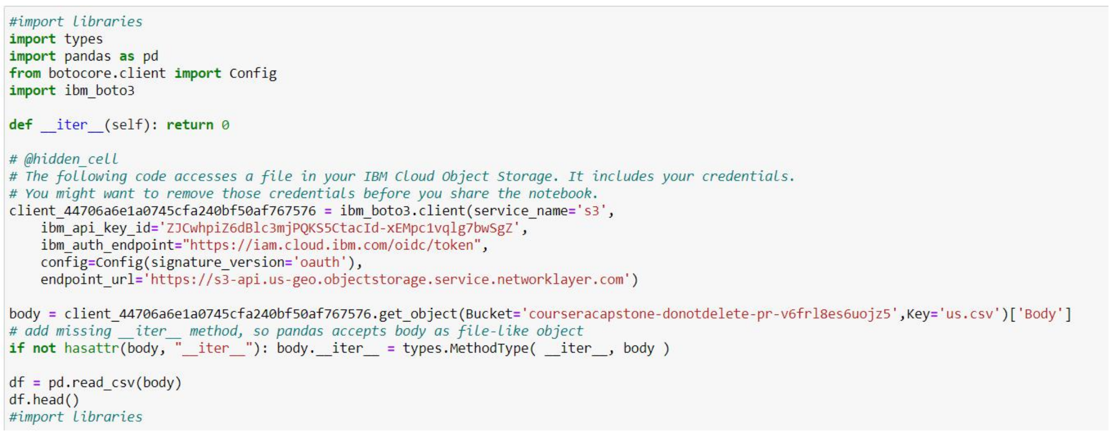
2. Methodology and results

The following part shows the process of data processing and visual analysis：

2.1 Data pre-processing

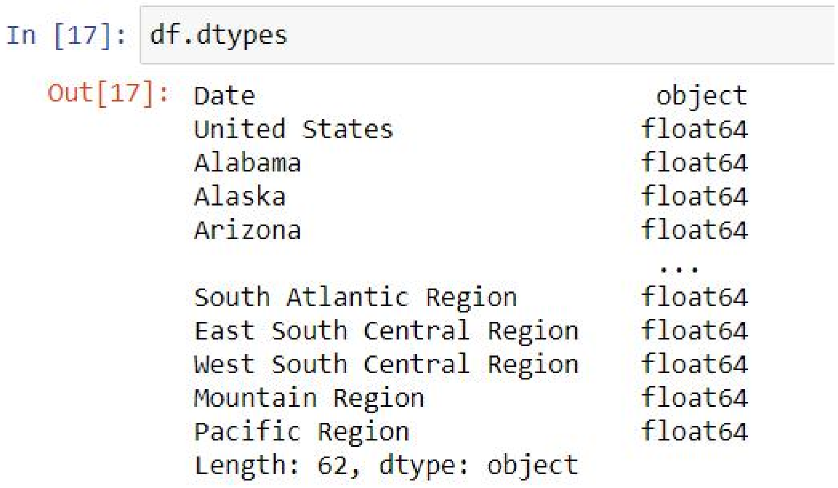
After downloading the data, delete the description section of the data for ease of use, and the rest data will be separated by commas. Therefore, for convenience of processing, we save the deleted data as CSV file.

- First load the data file (the file is already under the same path as the Python script)



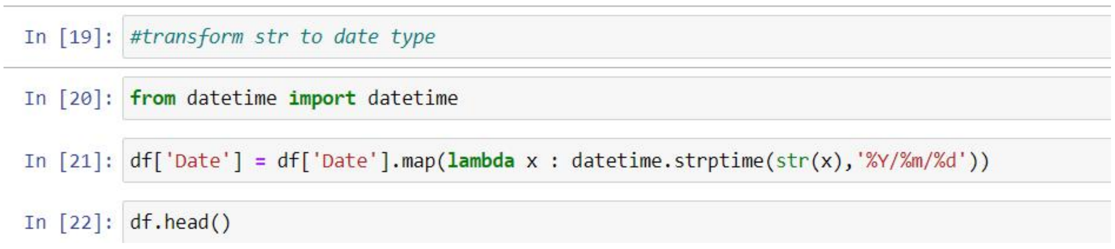
```python
#import libraries
import types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_44706a6e1a0745cfa240bf50af767576 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='ZJCwhpiZ6dBlc3mjPQKS5CtacId-xEMpc1vqlg7bwSgZ',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_44706a6e1a0745cfa240bf50af767576.get_object(Bucket='courseracapstone-donotdelete-pr-v6frl8es6uojz5',Key='us.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)
df.head()
#import libraries
```

- We can use *df.dtypes*（）to look at the types of data



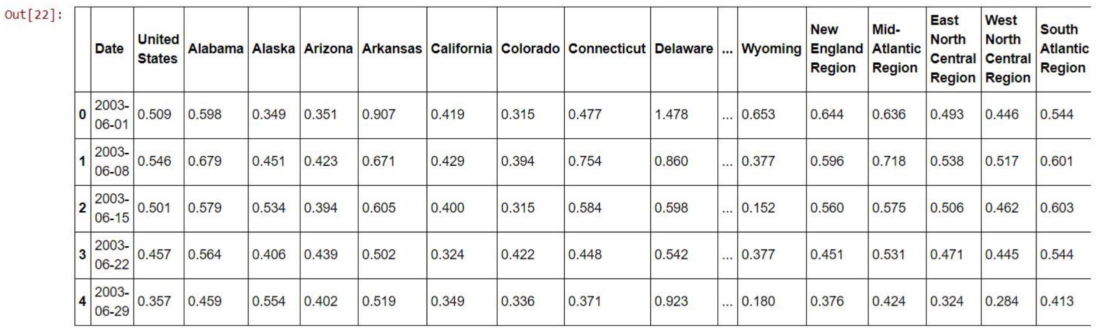
```
In [17]: df.dtypes

Out[17]: Date                            object
         United States                  float64
         Alabama                        float64
         Alaska                         float64
         Arizona                        float64
                                          ...
         South Atlantic Region          float64
         East South Central Region      float64
         West South Central Region      float64
         Mountain Region                float64
         Pacific Region                 float64
         Length: 62, dtype: object
```

- We can see from the output that the type of date is *object*, which is equivalent to a *string*. If we want to use some of the features in date types, we need to cast the type of the first column to a date type.



```
In [19]: #transform str to date type

In [20]: from datetime import datetime

In [21]: df['Date'] = df['Date'].map(lambda x : datetime.strptime(str(x),'%Y/%m/%d'))

In [22]: df.head()
```

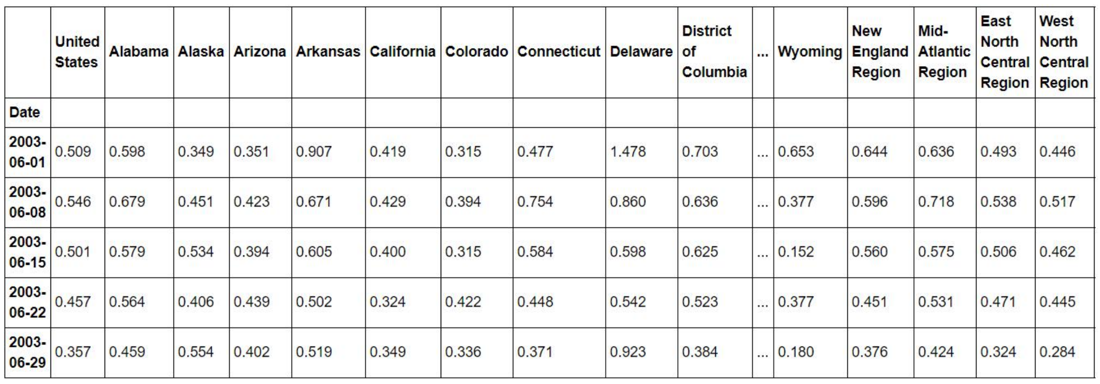
- The transformed data set is shown below：



Out[22]:

	Date	United States	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	...	Wyoming	New England Region	Mid-Atlantic Region	East North Central Region	West North Central Region	South Atlantic Region
0	2003-06-01	0.509	0.598	0.349	0.351	0.907	0.419	0.315	0.477	1.478	...	0.653	0.644	0.636	0.493	0.446	0.544
1	2003-06-08	0.546	0.679	0.451	0.423	0.671	0.429	0.394	0.754	0.860	...	0.377	0.596	0.718	0.538	0.517	0.601
2	2003-06-15	0.501	0.579	0.534	0.394	0.605	0.400	0.315	0.584	0.598	...	0.152	0.560	0.575	0.506	0.462	0.603
3	2003-06-22	0.457	0.564	0.406	0.439	0.502	0.324	0.422	0.448	0.542	...	0.377	0.451	0.531	0.471	0.445	0.544
4	2003-06-29	0.357	0.459	0.554	0.402	0.519	0.349	0.336	0.371	0.923	...	0.180	0.376	0.424	0.324	0.284	0.413

- Then use *df.set_index*（）改变索引：

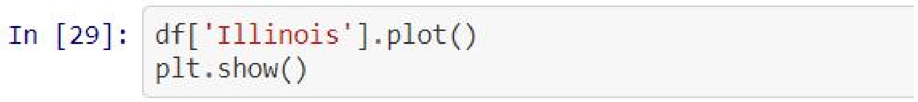


	United States	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	District of Columbia	...	Wyoming	New England Region	Mid-Atlantic Region	East North Central Region	West North Central Region
Date																
2003-06-01	0.509	0.598	0.349	0.351	0.907	0.419	0.315	0.477	1.478	0.703	...	0.653	0.644	0.636	0.493	0.446
2003-06-08	0.546	0.679	0.451	0.423	0.671	0.429	0.394	0.754	0.860	0.636	...	0.377	0.596	0.718	0.538	0.517
2003-06-15	0.501	0.579	0.534	0.394	0.605	0.400	0.315	0.584	0.598	0.625	...	0.152	0.560	0.575	0.506	0.462
2003-06-22	0.457	0.564	0.406	0.439	0.502	0.324	0.422	0.448	0.542	0.523	...	0.377	0.451	0.531	0.471	0.445
2003-06-29	0.357	0.459	0.554	0.402	0.519	0.349	0.336	0.371	0.923	0.384	...	0.180	0.376	0.424	0.324	0.284

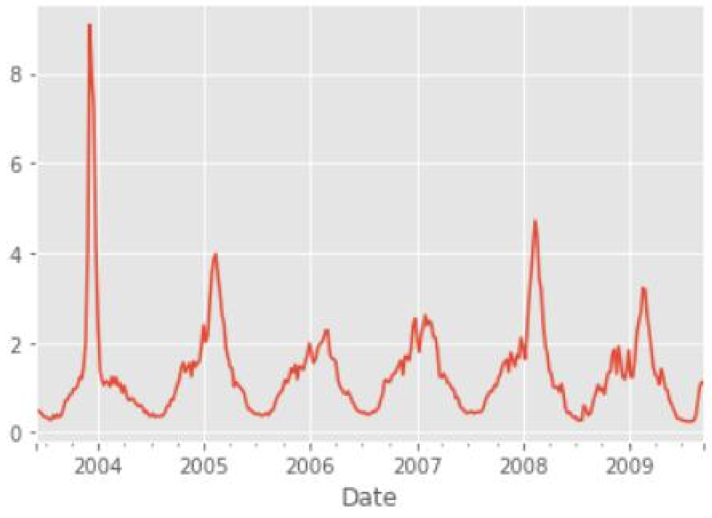
2.2 Data analysis

Because there are so many states in the United States, if you put all the data into one graph, the graph notes will be incomplete. Therefore, this article will select one state for analysis and then use more sub-graphs to show the data of different states.

- Analysis of the “Illinois”.

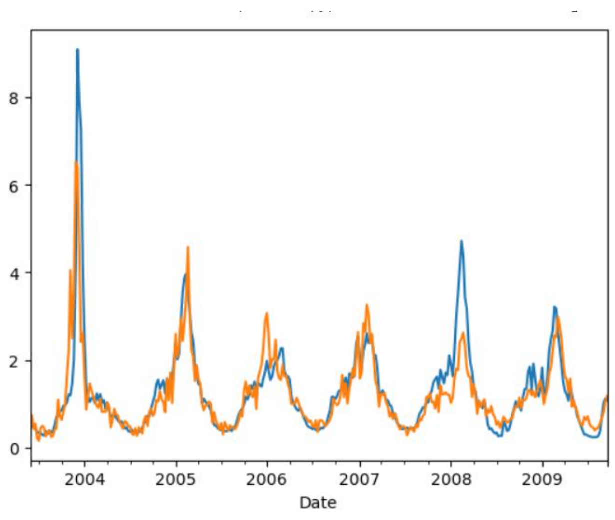


```
In [29]: df['Illinois'].plot()
         plt.show()
```



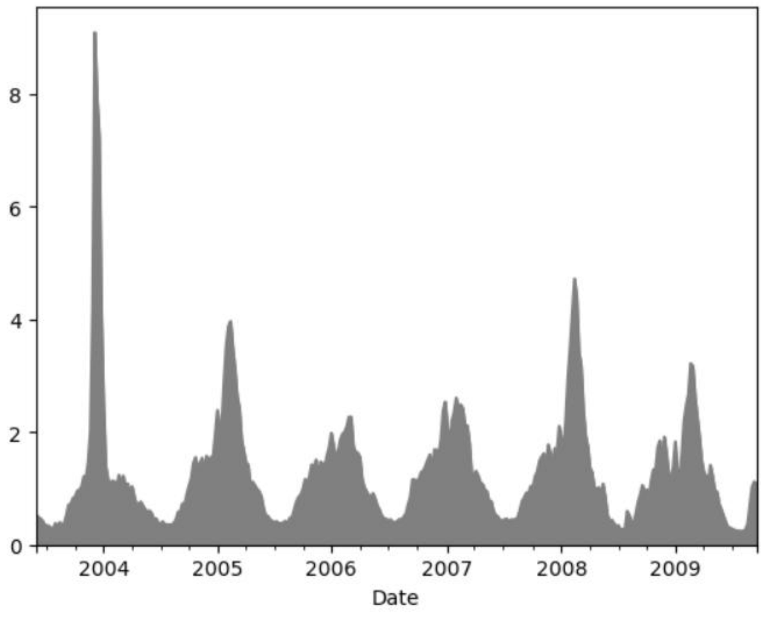
As you can see from the time series, the flu was at its worst in 2003. It has since been repeated, but not to a greater extent than in 2003.

- Compare the two states: Illinois and Idaho:



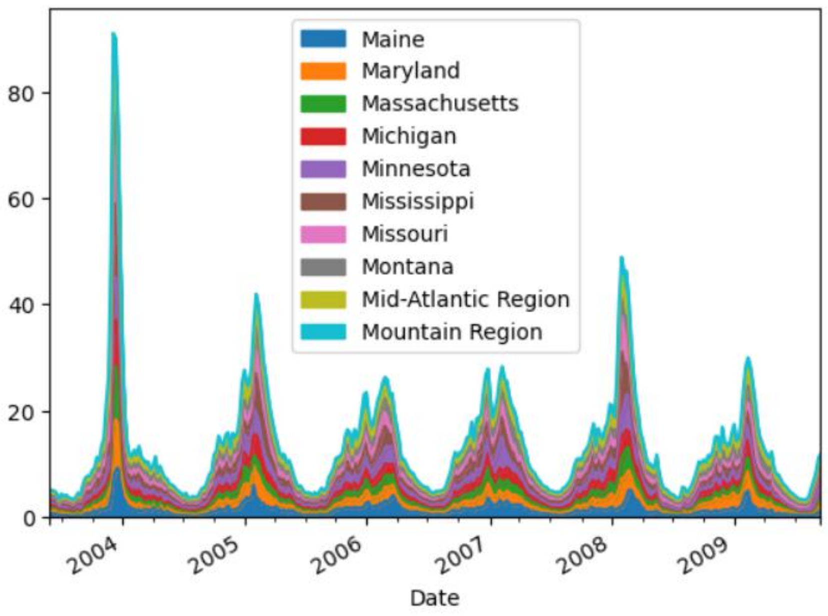
- Because the data density is so high, if all the state data is shown in the same graph, the curves will be highly overlapping, resulting in a small degree of differentiation.Therefore, according to the characteristics of the analysis data, to draw an area map.

```
In [42]: df.Illinois.plot(kind = 'area',label = 'Illinois',color = 'gray')
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7f03021bdbd0>
```

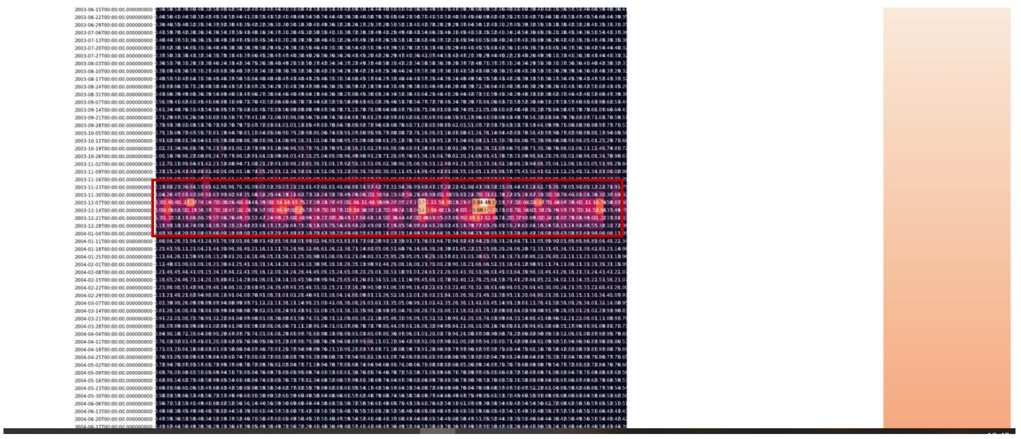


- There are many states in the United States. Suppose we only consider plotting the flu trend of the states with the initial letter M, using lambda and filter() methods, and using list to type the objects returned by filter. Next, draw the states that begin with the letter ‘M’ on the uniform canvas.

```
In [44]: df[[state for state in df.columns.values if state[0] == 'M']].plot(kind = 'area')
         plt.gcf().autofmt_xdate()
```



- The heat map was then chosen to show which states had the most severe flu at which time.The thermal diagram depicts the degree of correlation between data.In the figure below, the names of the states are shown horizontally, the dates on the vertical axis, and the colors indicate severity.Because only part of the heat map is captured, the darker the color, the better the flu, and the brighter the color, the more serious it is.



3. Conclusion and feature directions

This article focuses on data on Google flu trends.Select partial characteristic data for analysis, and draw area map and hot spot map according to the data characteristic. The method can also be used to analyze the current epidemic of COVID-19 infection heat-map. The color of the heat-map map can be used to identify the worst-hit areas and provide evidence for epidemic prevention and control measures.