Data Viusalisation with Python Programming

- Presentation By Uplatz
- Contact us: https://training.uplatz.com
- Email: info@uplatz.com
- Phone: +44 7836 212635

PROJECT

Project on Data Visualisation

Uplatz

Project Title

VISUALIZING COVID-19

Visualizing COVID-19

Project Description:

Within months, COVID-19 went from an epidemic to a pandemic. From the first identified case in December 2019, how did the virus spread so fast and widely? In this project, we will visualize data from the early months of the coronavirus outbreak to see how this virus grew to be a global pandemic.

Visualizing COVID-19

Project Description:

Using Python and some graphing libraries, you can project the total number of confirmed cases of COVID-19, and also display the total number of deaths for a country (this article uses India as an example) on a given date. Humans sometimes need help interpreting and processing the meaning of data, so this article also demonstrates how to create an animated horizontal bar graph for five countries, showing the variation of cases by date.

Visualizing COVID-19

PROJECT TASKS:

- FROM EPIDEMIC TO PANDEMIC
- COIVID-19 CASES IN INDIA, CHINA, US, ITALY, AND SPAIN.
- HOW THE COVID-19 SPREAD OVER A PERIOD OF TIME WITH THE HELP OF VISUALISATION.
- HOW FAST IT HAS SPREAD IN INDIA, US AND SOME OTHER COUNTRIES USING VISUALISATION TOOL.

Visualizing COVID-19

PROJECT TASKS:

- TOTAL NUMBER OF DEATH OVER THE PERIOD OF TIME IN INDIA AND OTHER COUNTRIES.
- WE WILL ALSO PERFORM GRAPHICAL DATA ANALYSIS USING STRIP PLOT.
- FINALLY WE WILL CREATE THE ANIMATED HORIZONTAL BAR CHART/GRAPH FOR SOME COUNTRIES.

Visualizing COVID-19

1. From epidemic to pandemic

The **COVID-19 pandemic**, also known as the **coronavirus pandemic**, is an on-going global pandemic of coronavirus disease 2019 (COVID-19). The outbreak was first identified in December 2019 in Wuhan, China. The World Health Organization declared the outbreak a Public Health Emergency of International Concern on 30 January 2020 and a pandemic on 11 March.

Visualizing COVID-19

1. **From epidemic to pandemic**

As of 2 September 2020, more than 25.8 million cases of COVID-19 have been reported in more than 188 countries and territories, resulting in more than 858,000 deaths; more than 17.1 million people have recovered.

We know that COVID-19 spreads through respiratory droplets, such as through coughing, sneezing, or speaking. But, how quickly did the virus spread across the globe?

Visualizing COVID-19

1. From epidemic to pandemic

In this project, we will visualize COVID-19 data from the month of January to till now .

Please note that information and data regarding COVID-19 is frequently being updated. The data used in this project should not be considered to be the most up to date data available.

Visualizing COVID-19

Task: Download data

Scientific data isn't always open, but fortunately, many modern science and healthcare organizations are eager to share information with each other and the public. Data about COVID-19 cases is available online, and it's updated frequently.

To parse the data, you first must download it: https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv

Visualizing COVID-19

Task: Load the data

Load the data directly into a Pandas DataFrame. Pandas provides a function, **read_csv()**, which can take a URL and give back a DataFrame object, as shown below:

```
import pandas as pd
URL_DATASET = r'https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'
df = pd.read_csv(URL_DATASET)
print(df.head(3))  # Get first 3 entries dataframe
```

Visualizing COVID-19

Task:

The top row of the data set contains column names:

Date, Country, Confirmed, Recovered, and **Deaths**.

The output of the **head** query includes a unique identifier (not listed as a column) plus an entry for each column:

0 2020-01-22 Afghanistan 0 0 0

1 2020-01-22 Albania 0 0 0

1 2020-01-22 Algeria 0 0 0

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Select data for India:

In this step, we will select only those rows in the DataFrame that include India. This is shown in the script below:

```
df_india = df[df['Country'] == 'India']
print(df_india.head(5))
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

How the COVID-19 spread over a period of time in India:

Here we create a bar chart. We will put the dates on the X-axis and the number of confirmed cases on the Y-axis.

```
plt.bar(x = 'Date', height = 'Confirmed', color = 'red', data=df_india )
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Select data for Italy:

In this step, we will select only those rows in the DataFrame that include Italy. This is shown in the script below:

```
df_italy = df[df['Country'] == 'Italy']
print(df_italy.head(10))
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

How the COVID-19 spread over a period of time in Italy:

Here we create a bar chart. We will put the dates on the X-axis and the number of confirmed cases the Y-axis.

```
plt.bar(x = 'Date', height = 'Confirmed', color = 'blue', data=df_italy)
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Select data for China:

In this step, we will select only those rows in the DataFrame that include China. This is shown in the script below:

```
df_china = df[df['Country'] == 'China']
print(df_china.head(10))
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

How the COVID-19 spread over a period of time in China:

Here we create a bar chart. We will put the dates on the X-axis and the number of confirmed cases on the Y-axis.

```
plt.bar(x = 'Date', height = 'Confirmed', color =
'orange', data=df_china)
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Select data for US:

In this step, we will select only those rows in the DataFrame that include US. This is shown in the script below:

```
df_us= df[df['Country'] == ‘US']
print(df_us.head(10))
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

How the COVID-19 spread over a period of time in US:

Here we create a bar chart. We will put the dates on the X-axis and the number of confirmed cases on the Y-axis.

```
plt.bar(x = 'Date', height = 'Confirmed', color = 'orange', data=df_us)
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Select data for Spain:

In this step, we will select only those rows in the DataFrame that include Spain. This is shown in the script below:

```
df_spain = df[df['Country'] == 'Spain']
print(df_spain.head(10))
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

How the COVID-19 spread over a period of time in Spain:

Here we create a bar chart. We will put the dates on the X-axis and the number of confirmed cases on the Y-axis.

```
plt.bar(x = 'Date', height = 'Confirmed', color = 'brown', data=df_spain)
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Total number of death over the period of time in India and other countries.

We create a bar chart. We will put the dates on the X-axis and the number of deaths on the Y-axis.

```
plt.bar(x = 'Date', height = ‘Deaths', color = 'red',
data=df_india )
plt.show()
```

Visualizing COVID-19

COIVID-19 cases in India, China, US, Italy, and Spain:

Total number of death over the period of time in India and other countries.

We create a bar chart. We will put the dates on the X-axis and the number of deaths on the Y-axis.

```
plt.bar(x = 'Date', height = ‘Deaths', color = 'red',
data=df_us )
plt.show()
```

Visualizing COVID-19

Graphical data analysis using strip plot:

Let’s understand the total number of death in case of India, China and US over the period of time with the help of graphical data analysis using strip plot. But we need to filter our data set so that we will get the data only belongs to these countries.

Here we will use query() to check for matches with a list of values corresponding to a column. Here we use **in** operator to check for equality.

```
df5 = df.query('Country in ["India", "China", "US"]')
print(df5)
```

Visualizing COVID-19

Graphical data analysis using strip plot:

Now we can create the strip plot as:

```
sb.stripplot(x='Date', y='Deaths', hue='Country',
data=df5)
plt.show()
```

This plot clearly show you the total number of deaths occurs from 22nd-January to 30th – August in case of **China, India and US.** You can also how fast it has effected the people’s life.

You can also do graphical data analysis for some other countries.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Matplotlib library of Python is a plotting tool used to plot graphs of functions or figures. It can also be used as an animation tool too. The plotted graphs when added with animations gives a more powerful visualization and helps the presenter to catch a larger number of audience. Matplotlib can also easily connect with Pandas to create even more sophisticated animations.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

The easiest way to make a live animation in matplotlib is to use one of the Animation classes.

Animation: This class wraps the creation of an animation using matplotlib.

FuncAnimation: Makes an animation by repeatedly calling a function func.

ArtistAnimation: Animation using a fixed set of Artist objects.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Animations in Matplotlib can be made by using the Animation class in two ways:

By calling a function over and over: It uses a predefined function which when ran again and again creates an animation.

By using fixed objects: Some animated artistic objects when combined with others yield an animation scene.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

It is important to note that we must at all points keep a reference to the animated object or else the animation will stop. This is because the Animation class holds a single pointer reference to the animation object and as the time advances to run the animation this pointer reference must be kept otherwise it will be collected as a garbage value. Though there are two ways, the first way is more common and convenient and here, we will make use of that only.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Now to create animated horizontal bar graph, let’s import the necessary libraries/modules.

Step-1:

```
import matplotlib.animation as animation
```

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-2: **Create a list of all dates**

If you examine the data you downloaded, you notice that it has a column **Date**. Now, this column has a date value for each country. So the same date is occurring a number of times. We need to create a list of dates with only unique values. This will be used on the X-axis of our bar charts. We have a line of code like: **list_dates = df[*'Date'*].unique()**.

The **unique()** method will pick up only the unique values for each date.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-3. Pick five countries and create an ax object

Take a list of five countries. (You can choose whatever countries you prefer, or even increase or decrease the number of countries). I have also taken a list of five colours for the bars of each country. (You can change this too if you like). One important line of code here is:

fig, ax = plt.subplots(figsize=(15, 8)).

This is needed to create an **ax** object.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-3. Pick five countries and create an ax object

plt.subplots() is a function that returns a tuple containing a figure and axes object(s). Thus when using fig, ax = plt.subplots() you unpack this tuple into the variables fig and ax. Having fig is useful if you want to change figure-level attributes or save the figure as an image file later (e.g. with fig.savefig('yourfilename.png')). You certainly don't have to use the returned figure object but many people do use it later so it's common to see.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-3. Pick five countries and create an **ax** object

```
fig, ax = plt.subplots(figsize=(15, 8))

# We will animate for these 5 countries only
list_countries = ['India', 'China', 'US', 'Italy', 'Spain']

# colors for the 5 horizontal bars
list_colors = ['black', 'red', 'green', 'blue', 'yellow']
```

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-4: Write the call back function

If you want to do animation in Matplotlib, you need to create an object of a class called **matplotlib.animation.FuncAnimation**. The signature of this class is available online. The constructor of this class, apart from other parameters, also takes a parameter called **func**, and you have to give this parameter a callback function. So in this step, we will write the callback function, which is repeatedly called in order to render the animation.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-4: Write the call back function

```
def plot_bar(some_date):
    df2 = df[df['Date'].eq(some_date)]
    ax.clear()
  df3 = df2.sort_values(by = 'Deaths', ascending = False)
  df4 = df3[df3['Country'].isin(list_countries)]
  sleep(0.2)  # To slow down the animation
    # makes a horizontal bar plot.
    return ax.barh(df4['Country'], df4['Deaths'],
     color= list_colors)
```

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-5. Create FuncAnimation object

This step has partly been explained in the previous step.

Our code to create an object of this class is:

```
my_anim = animation.FuncAnimation(fig = fig,
func = plot_bar, frames= list_dates, blit=True,
                 interval=20)
```

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-5. Create FuncAnimation object

The important parameters to be given are:

fig, which must be given a fig object, which we created earlier.

func, which must be the call back function.

frames, which must contain the variable on which the animation is to be done. Here in our case, it will be the list of dates we created earlier.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-5. Create FuncAnimation object

blit, Whether blitting is used to optimize drawing. It is a logical operation in which a block of data is rapidly moved or copied in memory, most commonly used to animate two-dimensional graphics.

interval, Delay between frames in milliseconds.

Visualizing COVID-19

Creating an animated horizontal bar graph for five countries:

Step-6.

Finally we can see the animated horizontal bar graph using plt.show()

Thank you

Uplatz