10_April_Batch

April 10, 2020

[1]:
```
import turtle
# turtle graphics design
```

[2]:
```
pen = turtle.Pen()
```

[3]:
```
pen.forward(100)
```

[4]:
```
pen.left(60)
```

[5]:
```
pen.forward(100)
```

[6]:
```
pen.right(60)
```

[7]:
```
pen.right(60)
```

[8]:
```
pen.forward(100)
```

[9]:
```
pen.left(60)
```

[10]:
```
pen.forward(100)
```

[11]:
```
pen.right(90)
```

[12]:
```
pen.forward(100)
```

[14]:
```
pen.right(90)
pen.forward(300)
```

[15]:
```
pen.right(90)
```

[16]:
```
pen.forward(100)
```

[23]:
```
pen.reset()
```

[24]:
```
# squre side length 200
```

[25]:
```
pen.color('red', 'orange')
```

[26]:
```
pen.begin_fill()
```

[27]:
```
for _ in range(4):
    pen.forward(200)
    pen.left(90)
```

[28]:
```
pen.end_fill()
```

[29]:
```
pen.up()
```

[30]:
```
pen.backward(200)
```

[31]:
```
pen.down()
```

[32]:
```
pen.color('black', 'blue')
pen.begin_fill()
pen.circle(200)
pen.end_fill()
```

[33]:
```
pen.reset()
```

[34]:
```
pen.up();
pen.backward(250)
pen.down()
```

[35]:
```
pen.color('red', 'yellow')
pen.begin_fill()
```

[36]:
```
pen.speed(0)
for var in range(300):
    pen.forward(500)
    pen.left(171)
pen.end_fill()
```

[37]:
```
help(pen.speed)
```

```
Help on method speed in module turtle:

speed(speed=None) method of turtle.Turtle instance
    Return or set the turtle's speed.

    Optional argument:
    speed -- an integer in the range 0..10 or a speedstring (see below)

    Set the turtle's speed to an integer value in the range 0 .. 10.
    If no argument is given: return current speed.
```

```
    If input is a number greater than 10 or smaller than 0.5,
    speed is set to 0.
    Speedstrings  are mapped to speedvalues in the following way:
        'fastest' :  0
        'fast'    :  10
        'normal'  :  6
        'slow'    :  3
        'slowest' :  1
    speeds from 1 to 10 enforce increasingly faster animation of
    line drawing and turtle turning.

    Attention:
    speed = 0 : *no* animation takes place. forward/back makes turtle jump
    and likewise left/right make the turtle turn instantly.

    Example (for a Turtle instance named turtle):
    >>> turtle.speed(3)
```

[38]:
```
turtle.exitonclick()
```

[39]:
```
url = "https://ec.europa.eu/programmes/creative-europe/sites/creative-europe/
 ↪files/covid19-cdc-unsplash.jpg"
```

[41]:
```
import requests
```

[42]:
```
page = requests.get(url)
```

[47]:
```
page.headers['Content-type']
```

[47]:
```
'image/jpeg'
```

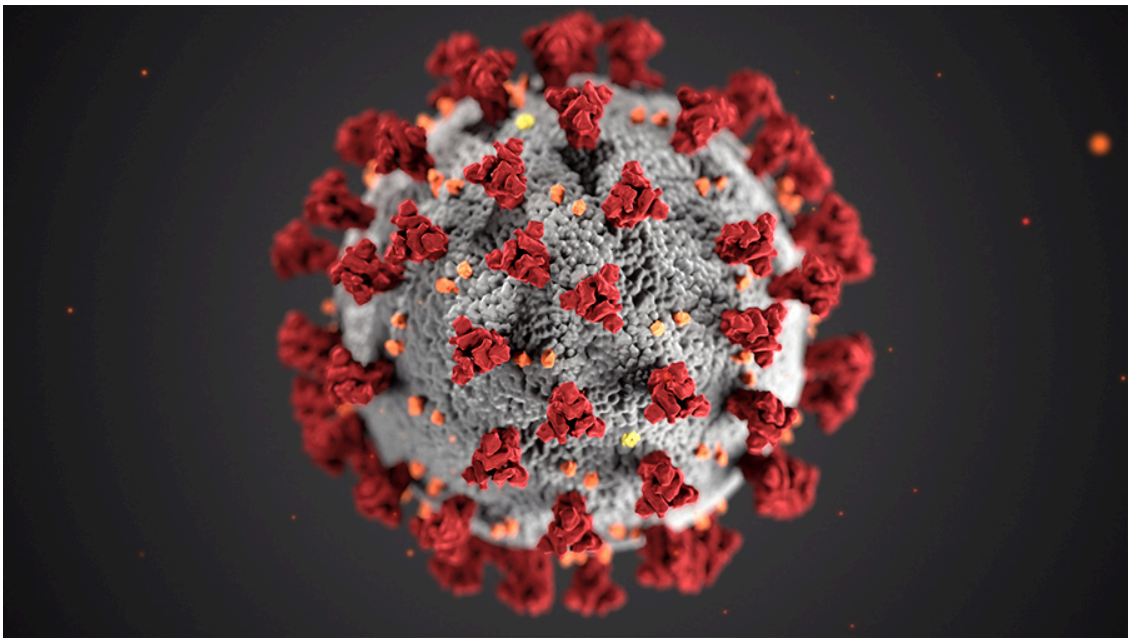
[44]:
```
file = open("corona.jpeg", 'wb')
file.write(page.content)
file.close()
```

[45]:
```
pwd
```

[45]:
```
'C:\\Users\\sachin\\Desktop\\python_Webinar'
```

[46]:
```
from PIL import Image
Image.open('corona.jpeg')
```

[46]:



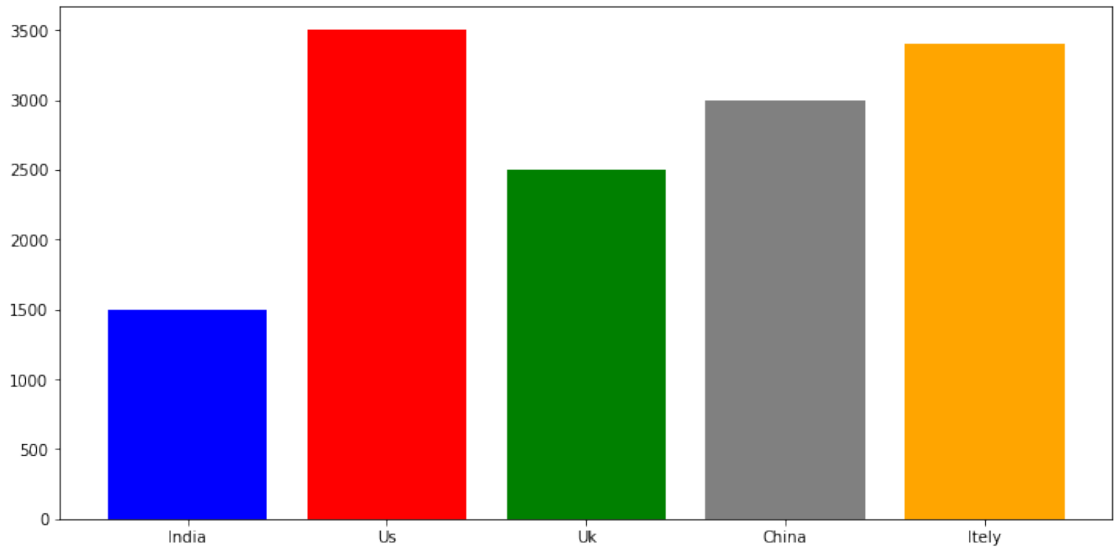
Data Visulation

https://github.com/sachinyadav3496/kiit

[54]:
```
country = [ 'India', 'Us', 'Uk', 'China', 'Itely']
corona =  [ 1500, 3500, 2500, 3000, 3400]
color =   [ 'blue', 'red', 'green', 'gray', 'orange']
```

[55]:
```
import matplotlib.pyplot as plt
```

[56]:
```
plt.figure(figsize=(12, 6))
plt.bar(country, corona, color=color)
plt.show()
```



Ploting Live Data of Corona Patients `sources --> worldometer`

[57]:
```
url = "https://www.worldometers.info/coronavirus/"
```

[58]:
```
page = requests.get(url)
```

[59]:
```
page.headers['Content-type']
```

[59]: 'text/html; charset=UTF-8'

0.0.1 Data Engineering

Data Gathering
Data Cleaning
Data Analysis

[61]:
```
import pandas as pd
```

[62]:
```
sheet = pd.read_html(page.content)
```

[64]:
```
len(sheet)
```

[64]: 2

[65]:
```
table = sheet[0]
```

[66]:
```
table.head()
```

[66]:
	Country,Other	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	\
0	World	1632827	+29,175	97589.0	+1,897	366587.0	
1	USA	475237	+6,671	17055.0	+364	26050.0	
2	Spain	157022	+3,800	15843.0	+396	55668.0	
3	Italy	143626	NaN	18279.0	NaN	28470.0	
4	Germany	119401	+1,166	2607.0	NaN	52407.0	

	ActiveCases	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	TotalTests	\
0	1168651	49416.0	209.0	12.5	NaN	
1	432132	10182.0	1436.0	52.0	2416650.0	
2	85511	7371.0	3358.0	339.0	355000.0	
3	96877	3605.0	2375.0	302.0	853369.0	
4	64387	4895.0	1425.0	31.0	1317887.0	

	Tests/ 1M pop
0	NaN
1	7301.0
2	7593.0
3	14114.0
4	15730.0

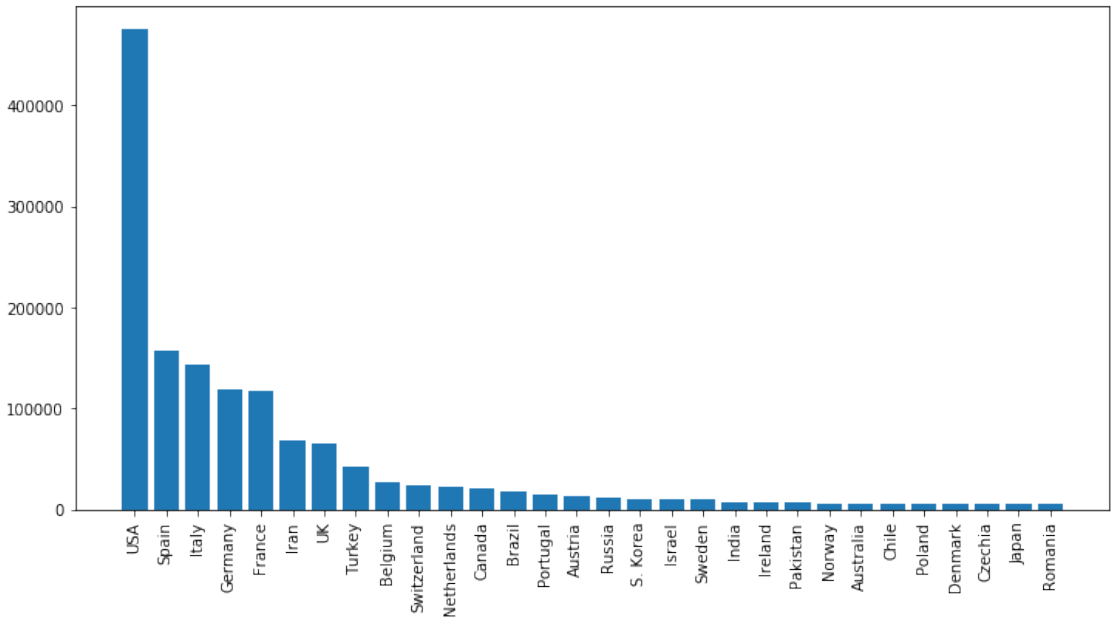
[69]:
```
table = table[1:-1]
```

[67]:
```
table.shape
```

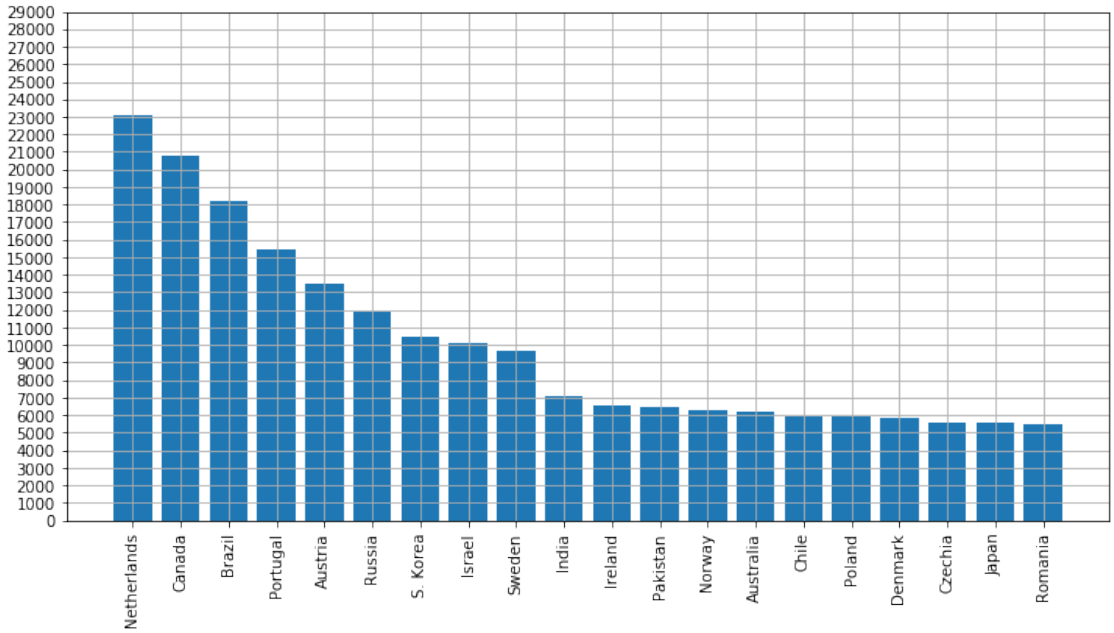
[67]: (214, 12)

[73]:
```
plt.rcParams['figure.figsize'] = 12, 6
```

[75]:
```
plt.bar(table['Country,Other'][:30], table['TotalCases'][:30])
plt.xticks(rotation=90)
plt.show()
```



[77]:
```
plt.bar(table['Country,Other'][10:30], table['TotalCases'][10:30])
plt.xticks(rotation=90)
plt.yticks(range(0, 30000, 1000))
plt.grid()
plt.show()
```



Desktop Application

[81]:
```
import tkinter

root_window = tkinter.Tk()

root_window.wm_minsize(600, 350)
root_window.mainloop()
```

[86]:
```
import tkinter

root_window = tkinter.Tk()

label_window = tkinter.Label(root_window, text="Hello World This is Cool!",
                             font=('monospace', 20, 'bold'), bg='blue',␣
 ↪fg='white')
label_window.pack()


root_window.wm_minsize(600, 350)
root_window.mainloop()
```

[96]:
```
import tkinter

root_window = tkinter.Tk()

#### label window starts here
label_window = tkinter.Label(root_window, text="Hello World This is Cool!",
                             font=('monospace', 20, 'bold'), bg='blue',␣
 ↪fg='white')
label_window.pack()
#_____________________________________________________________________________________________
### Input Box
def process_input():
    ans = variable.get() + " = " + str(eval(variable.get()))
    tmp = tkinter.Label(root_window, text=text,
                        fg='gray', font=('monospace', 20))
    tmp.pack()
    variable.set("")
variable = tkinter.StringVar()

input_box = tkinter.Entry(root_window, textvariable=variable,
                          font=("monospace", 20, 'bold'), fg='blue')
input_box.pack() # grid, place

process = tkinter.Button(root_window, text="!!Do IT!!",
                         font=('monospace', 20, 'bold'), bg='black', fg='cyan',
                        command=process_input)
```

```
process.pack()

#_____________________________________________________________________________
#### exit button code
button = tkinter.Button(root_window, text='!Exit!',
                        font=('times', 15, 'bold'), bg='red', fg='white',␣
 ↪command=root_window.destroy)

button.pack()

#_____________________________________________________________________________
root_window.wm_minsize(600, 350)

root_window.mainloop()
```

[ ]:
```
import tkinter

root_window = tkinter.Tk()

#### label window starts here
label_window = tkinter.Label(root_window, text="My Calculator Applicaiton",
                             font=('monospace', 20, 'bold'), bg='blue',␣
 ↪fg='white')
label_window.pack()
#_____________________________________________________________________________
### Input Box
def process_input():
    text = variable.get() + " = " + str(eval(variable.get()))
    tmp = tkinter.Label(root_window, text=text,
                        fg='gray', font=('monospace', 20))
    tmp.pack()
    variable.set("")
variable = tkinter.StringVar()

input_box = tkinter.Entry(root_window, textvariable=variable,
                          font=("monospace", 20, 'bold'), fg='blue')
input_box.pack() # grid, place

process = tkinter.Button(root_window, text="!!Solve!!",
                         font=('monospace', 15, 'bold'), bg='blue', fg='white',
                        command=process_input)
process.pack()

#_____________________________________________________________________________
#### exit button code
button = tkinter.Button(root_window, text='!Exit!',
```

```
                        font=('times', 15, 'bold'), bg='red', fg='white',␣
 ↪command=root_window.destroy)

button.pack()

#_____________________________________________________________________________________
root_window.wm_minsize(600, 350)

root_window.mainloop()
```

sachinyadav3496

https://www.youtube.com/watch?v=M5ILgNI0iXw&t=272s

[ ]: