Assignment_2_Data_collection

January 11, 2021

1 CSB352: Data Mining

Instructor : [Dr. Chandra Prakash]

• For more information visit the class website.

2 Assignment 2: Data Collection using PYTHON Frameworks

LAB 2 : Python Frameworks- Numpy + Pandas + SkLearn

Assigning Date : 11-01-2021

Due Date: 16-Jan-2021

Student Name: Rohit Byas Sherwan

Roll No : 181210043

3 Assignment Instructions

You must save your as Assignment_NO_Yourname

Agenda for the Assignment 2

1. How to download a dataset and perform operations using numpy and pandas
2. Perform the given tasks. Your source file will most likely end in .pynb if you are using a Jupyter notebook; however, it might also end in .py if you are using a Python script. You have to add your name and roll no in the Google Colab Instructions section below and print it.

4 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

```
try:
  from google.colab import drive
  %tensorflow_version 2.x
  COLAB = True
  print("Hello World")
  print("Note: using Google CoLab")
except:
  print("Hello NITD")
```

```
    print("Note: not using Google CoLab")
    COLAB = False
# Print your name and Roll No.
print('Rohit Byas Sherwan 181210043')
```

```
Hello World
Note: using Google CoLab
Rohit Byas Sherwan 181210043
```

Task 1:

1. Read in the Dataset: Coronavirus Source Data https://ourworldindata.org/coronavirus-source-data
2. Print:

   a. print the first 5 rows and the last 5 rows (in different cells)

   b. the size of the dataframe i.e how many rows and columns

   c. datatype of each column

   d. basic statistics of each column

```
###Your code here
import requests
res=requests.get("https://covid.ourworldindata.org/data/owid-covid-data.csv")
```

```
###Your code here
import numpy as np
import pandas as pd
dataset = pd.read_csv('/content/sample_data/owid-covid-data.csv')
X = dataset.iloc[0:5,:].values
print(X)
print("\n")
Y = dataset.iloc[-5:,:].values
print(Y)
print("\n")
row = len(dataset)
column = len(dataset.columns)
print("rows & columns : ",row,column)
print("\n")
print(dataset.dtypes)
print("\n")
stats = dataset.describe(include='all')
print(stats)
```

```
[['AFG' 'Asia' 'Afghanistan' '2020-02-24' 1.0 1.0 nan nan nan nan
  0.026000000000000002 0.026000000000000002 nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
  nan nan 8.33 38928341.0 54.422 18.6 2.5810000000000004 1.337 1803.987
  nan 597.029 9.59 nan nan 37.746 0.5 64.83 0.498]
 ['AFG' 'Asia' 'Afghanistan' '2020-02-25' 1.0 0.0 nan nan nan nan
```

```
  0.026000000000000002 0.0 nan nan nan nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 8.33
  38928341.0 54.422 18.6 2.5810000000000004 1.337 1803.987 nan 597.029
  9.59 nan nan 37.746 0.5 64.83 0.498]
 ['AFG' 'Asia' 'Afghanistan' '2020-02-26' 1.0 0.0 nan nan nan nan
  0.026000000000000002 0.0 nan nan nan nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 8.33
  38928341.0 54.422 18.6 2.5810000000000004 1.337 1803.987 nan 597.029
  9.59 nan nan 37.746 0.5 64.83 0.498]
 ['AFG' 'Asia' 'Afghanistan' '2020-02-27' 1.0 0.0 nan nan nan nan
  0.026000000000000002 0.0 nan nan nan nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 8.33
  38928341.0 54.422 18.6 2.5810000000000004 1.337 1803.987 nan 597.029
  9.59 nan nan 37.746 0.5 64.83 0.498]
 ['AFG' 'Asia' 'Afghanistan' '2020-02-28' 1.0 0.0 nan nan nan nan
  0.026000000000000002 0.0 nan nan nan nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 8.33
  38928341.0 54.422 18.6 2.5810000000000004 1.337 1803.987 nan 597.029
  9.59 nan nan 37.746 0.5 64.83 0.498]]


[['ZWE' 'Africa' 'Zimbabwe' '2021-01-10' 21477.0 978.0 887.429 507.0 24.0
  18.143 1445.005 65.801 59.708 34.111999999999995 1.615
  1.2209999999999999 nan nan nan nan nan nan nan nan nan 4936.0 251048.0
  16.891 0.332 3710.0 0.25 0.239 4.2 'tests performed' nan nan nan nan
  nan 92.59 14862927.0 42.729 19.6 2.822 1.882 1899.775 21.4 307.846 1.82
  1.6 30.7 36.791 1.7 61.49 0.535]
 ['ZWE' 'Africa' 'Zimbabwe' '2021-01-11' 22297.0 820.0 924.0 528.0 21.0
  20.570999999999998 1500.1760000000002 55.17100000000001 62.168 35.525
  1.413 1.3840000000000001 nan nan nan nan nan nan nan nan nan 1518.0
  252566.0 16.993 0.102 3659.0 0.24600000000000002 0.253 4.0
  'tests performed' nan nan nan nan nan nan 14862927.0 42.729 19.6 2.822
  1.882 1899.775 21.4 307.846 1.82 1.6 30.7 36.791 1.7 61.49 0.535]
 ['ZWE' 'Africa' 'Zimbabwe' '2021-01-12' 23239.0 942.0 863.571 551.0 23.0
  19.0 1563.555 63.379 58.102 37.071999999999996 1.547 1.278 nan nan nan
  nan nan nan nan nan nan 4462.0 257028.0 17.293 0.3 3599.0 0.242 0.24
  4.2 'tests performed' nan nan nan nan nan nan 14862927.0 42.729 19.6
  2.822 1.882 1899.775 21.4 307.846 1.82 1.6 30.7 36.791 1.7 61.49 0.535]
 ['ZWE' 'Africa' 'Zimbabwe' '2021-01-13' 24256.0 1017.0 921.7139999999999
  589.0 38.0 22.570999999999998 1631.98 68.425 62.013999999999996 39.629
  2.557 1.5190000000000001 nan nan nan nan nan nan nan nan nan 3507.0
  260535.0 17.529 0.23600000000000002 3429.0 0.231 0.26899999999999996
  3.7 'tests performed' nan nan nan nan nan nan 14862927.0 42.729 19.6
  2.822 1.882 1899.775 21.4 307.846 1.82 1.6 30.7 36.791 1.7 61.49 0.535]
 ['ZWE' 'Africa' 'Zimbabwe' '2021-01-14' 25368.0 1112.0 956.143 636.0
  47.0 27.143 1706.797 74.817 64.331 42.791000000000004 3.162
  1.8259999999999998 nan nan nan nan nan nan nan nan nan nan nan nan nan
  nan nan nan nan nan nan nan nan nan nan 14862927.0 42.729 19.6
  2.822 1.882 1899.775 21.4 307.846 1.82 1.6 30.7 36.791 1.7 61.49 0.535]]


rows & columns :  60137 55
```

Task 2:

1. On the dataframe generated above, please answer the following:
    1. Data of how many countries is present?
    2. How many continents?

3. How many rows belong to India?
4. what is the window of dates for which data is provided?

```
###Your code here
import numpy as np
import pandas as pd
dataset = pd.read_csv('/content/sample_data/owid-covid-data.csv')
stats = dataset.describe(include='all')
countries = len(dataset['iso_code'].unique())
continents = len(dataset['continent'].dropna().unique())
print("No. of countries : ",countries)
print("No. of continents : ",continents)
India = len(dataset.loc[dataset['iso_code'] == 'IND'])
print("Row belongs to India : ",India)
date1 = dataset.loc[dataset['iso_code'] == 'IND'].iloc[0:1,3:4].values
date2 = dataset.loc[dataset['iso_code'] == 'IND'].iloc[-1:,3:4].values
print(date1[0][0],"to",date2[0][0])
```

```
No. of countries :  192
No. of continents :  6
Row belongs to India :  351
2020-01-30 to 2021-01-14
```

Task 3:

1. On the dataframe generated above, please answer the following:
    1. Extract Data of only India and make into a new dataframe.
    2. Extract only the total cases column.
    3. Convert total cases column into percentage. Percentage of total cases. Total cases is the number of cases as on the last date of the dataset. Use Numpy. DO NOT WRITE A LOOP!
    4. Add this newly generated column to the India Dataframe.

```
###Your code here
import numpy as np
import pandas as pd
dataset = pd.read_csv('/content/sample_data/owid-covid-data.csv')
India = dataset.loc[dataset['iso_code'] == 'IND']
print(India)
totalcase = India['total_cases']
print(totalcase)
percent = totalcase.cumsum()*100/totalcase.sum()
print(percent)
India['percent_total case'] = percent
print(India)
```

Task 4:

1. On the dataframe generated above, please answer the following:
    1. extract the total deaths and total cases as a numpy array.
    2. transpose the array.
    3. create new array which contains only total cases.
    4. from the total cases row , extract only rows which are more than 10000.

```
### Your Code Here
arr = np.array([dataset['total_deaths'], dataset['total_cases']])
print(arr)
print("\n")
arr=arr.transpose()
print(arr)
print("\n")
arr2 = np.array(dataset['total_cases'])
print(arr2)
print("\n")
filter_arr = []

# go through each element in arr
for element in arr2:
  # if the element is completely divisble by 2, set the value to True, otherwise F
  if element > 10000:
    filter_arr.append(True)
  else:
    filter_arr.append(False)

newarr = arr2[filter_arr]
print(newarr)
print("\n")
```

```
[[       nan        nan        nan ... 5.5100e+02 5.8900e+02 6.3600e+02]
 [1.0000e+00 1.0000e+00 1.0000e+00 ... 2.3239e+04 2.4256e+04 2.5368e+04]]


[[       nan 1.0000e+00]
 [       nan 1.0000e+00]
 [       nan 1.0000e+00]
 ...
 [5.5100e+02 2.3239e+04]
 [5.8900e+02 2.4256e+04]
 [6.3600e+02 2.5368e+04]]


[1.0000e+00 1.0000e+00 1.0000e+00 ... 2.3239e+04 2.4256e+04 2.5368e+04]


[10001. 10585. 11176. ... 23239. 24256. 25368.]
```

Task 5:

1. On the dataframe generated above, please answer the following:
    1. Extract data only for 18-08-2020.
    2. which country has the max and min.
    3. take the top 10 and bottom 10 of total cases and total deaths.

```
### Your Code here
case1 = dataset.loc[dataset['date'] == '2020-08-18']
case2 = case1.sort_values('total_cases')
case3 = case2.iloc[0:1,2:3].values
case4 = case2.iloc[-2:-1,2:3].values
print(case3[0][0])
print(case4[0][0])
print("\n")
case5 = case2.iloc[0:10,:].values
case6 = case2.iloc[-10:-1,:].values
print(case5)
print("\n")
print(case6)
print("\n")
```

```
 2068079.0 108.185 1.214 25495.0 1.334 0.07 14.3 'tests performed' nan
 nan nan nan nan 81.94 19116209.0 24.281999999999996 35.4 11.087 6.938
 22767.037 1.3 127.993 8.46 34.2 41.5 nan 2.11 80.18 0.843]
['COL' 'South America' 'Colombia' '2020-08-18' 489122.0 12462.0
 11238.428999999998 15619.0 247.0 306.286 9612.702 244.915
 220.86900000000003 306.96 4.854 6.019 1.03 nan nan nan nan nan nan nan
 nan 35768.0 2238559.0 43.994 0.703 37905.0 0.745 0.315 3.2
 'tests performed' nan nan nan nan nan 87.04 50882884.0 44.223 32.2
 7.646 4.312 13254.948999999999 4.5 124.24 7.44 4.7 13.5
 65.38600000000001 1.71 77.29 0.747]
['MEX' 'North America' 'Mexico' '2020-08-18' 531239.0 5506.0 5531.0
 57774.0 751.0 549.286 4120.28 42.70399999999999 42.898 448.094 5.825
 4.26 0.97 nan nan nan nan nan nan nan nan 14992.0 1223608.0 9.49
 0.11599999999999999 11930.0 0.09300000000000001 0.457 2.2
 'people tested' nan nan nan nan nan 70.83 128932753.0 66.444 29.3 6.857
 4.321000000000001 17336.468999999997 2.5 152.783 13.06 6.9 21.4
 87.84700000000001 1.38 75.05 0.774]
['PER' 'South America' 'Peru' '2020-08-18' 541493.0 5547.0 7401.857
 26481.0 200.0 711.429 16422.89 168.234 224.49 803.14 6.066
 21.576999999999998 1.08 nan nan nan nan nan nan nan nan 8152.0 574616.0
 17.427 0.247 6734.0 0.204 0.237 4.2 'tests performed' nan nan nan nan
 nan 85.19 32971846.0 25.129 29.1 7.151 4.455 12236.706 3.5 85.755 5.95
 4.8 nan nan 1.6 76.74 0.75]
['ZAF' 'Africa' 'South Africa' '2020-08-18' 592144.0 2258.0
 3719.2859999999996 12264.0 282.0 216.143 9984.101999999999
 38.071999999999996 62.711000000000006 206.783 4.755 3.6439999999999997
 0.65 nan nan nan nan nan nan nan nan 14677.0 3430347.0 57.839 0.247
 21624.0 0.365 0.172 5.8 'people tested' nan nan nan nan nan 77.78
 59308690.0 46.754 27.3 5.343999999999999 3.053 12294.876 18.9 200.38
 5.52 8.1 33.2 43.993 2.32 64.13 0.6990000000000001]
['RUS' 'Europe' 'Russia' '2020-08-18' 930276.0 4718.0 4940.714 15836.0
 129.0 104.714 6374.615 32.33 33.856 108.514 0.884 0.718 0.95 nan nan
 nan nan nan nan nan nan 248709.0 33217468.0 227.61900000000003 1.704
 272815.0 1.869 0.018000000000000002 55.2 'tests performed' nan nan nan
 nan nan 68.06 145934460.0 8.823 39.6 14.177999999999999
```

```
 9.392999999999999 24765.953999999998 0.1 431.29699999999997 6.18 23.4
 58.3 nan 8.05 72.58 0.816]
['IND' 'Asia' 'India' '2020-08-18' 2767253.0 64572.0 62516.429000000004
 52888.0 1091.0 971.0 2005.249 46.791000000000004 45.302 38.325

 0.7909999999999999 0.7040000000000001 1.09 nan nan nan nan nan nan nan
 nan 899864.0 30941264.0 22.421 0.652 808488.0 0.586 0.077 12.9
 'samples tested' nan nan nan nan 79.63 1380004385.0
 450.41900000000004 28.2 5.989 3.4139999999999997 6426.674 21.2 282.28
 10.39 1.9 20.6 59.55 0.53 69.66 0.64]
['BRA' 'South America' 'Brazil' '2020-08-18' 3407354.0 47784.0 42532.0
 109888.0 1352.0 980.2860000000001 16030.125 224.803 200.095 516.975
 6.361000000000001 4.612 0.98 nan nan nan nan nan nan nan nan nan nan
 nan nan 67453.0 0.317 nan nan 'tests performed' nan nan nan nan nan
 69.91 212559409.0 25.04 33.5 8.552 5.06 14103.452 3.4
 177.96099999999998 8.11 10.1 17.9 nan 2.2 75.88 0.759]
['USA' 'North America' 'United States' '2020-08-18' 5477609.0 45065.0
 48985.71400000001 172242.0 1302.0 1038.5710000000001 16548.535
 136.14700000000002 147.99200000000002 520.364 3.9339999999999997 3.138
 0.89 8859.0 26.764 43840.0 132.446 nan nan nan nan 879134.0 79257624.0
 239.447 2.656 819171.0 2.475 nan nan 'tests performed' nan nan nan nan
 nan 67.13 331002647.0 35.608000000000004 38.3 15.413 9.732000000000001
 54225.445999999996 1.2 151.089 10.79 19.1 24.6 nan 2.77 78.86 0.924]]
```

Task 6:

1. On the dataframe generated above, please answer the following:
    1. Use group by and

        a. print number of countries in each group

        b. get the total cases and deaths count for each continent as on 18-08-2020.
    2. use apply() and a. calculate the mortality rate for each row. and add that mortality data as a new column to the dataframe
    3. do the same without apply

```
### Your Code here
temp1 = dataset.loc[dataset['date'] == '2020-08-18']
temp = temp1.groupby(['continent']).count()
temp2 = temp.iloc[:,0:1].values
print(temp2)
print('\n')
print("Total Cases:-")
Africa = dataset.loc[dataset['continent'] == 'Africa']['total_cases'].sum()
print("Africa: ",Africa)
Asia = dataset.loc[dataset['continent'] == 'Asia']['total_cases'].sum()
print("Asia: ",Asia)
Europe = dataset.loc[dataset['continent'] == 'Europe']['total_cases'].sum()
print("Europe: ",Europe)
North_America = dataset.loc[dataset['continent'] == 'North America']['total_cases'
print("North_America: ",North_America)
South_America = dataset.loc[dataset['continent'] == 'South_America']['total_cases'
```

```
print("South_America: ",South_America)
Oceania = dataset.loc[dataset['continent'] == 'Oceania']['total_cases'].sum()
print("Oceania: ",Oceania)
print('\n')
print("Total Deaths:-")
Africa1 = dataset.loc[dataset['continent'] == 'Africa']['total_deaths'].sum()
print("Africa: ",Africa1)
Asia1 = dataset.loc[dataset['continent'] == 'Asia']['total_deaths'].sum()
print("Asia: ",Asia1)
Europe1 = dataset.loc[dataset['continent'] == 'Europe']['total_deaths'].sum()
print("Europe: ",Europe1)
North_America1 = dataset.loc[dataset['continent'] == 'North America']['total_death
print("North_America: ",North_America1)
South_America1 = dataset.loc[dataset['continent'] == 'South_America']['total_death
print("South_America: ",South_America1)
Oceania1 = dataset.loc[dataset['continent'] == 'Oceania']['total_deaths'].sum()
print("Oceania: ",Oceania1)

total_deaths = temp1['total_deaths']
population = temp1['population']
mortality_rate = total_deaths*100/population
print(mortality_rate)
temp1['mortality_rate'] = mortality_rate
print(temp1)
```

```
[[54]
 [46]
 [46]
 [23]
 [ 4]
 [12]]


Total Cases:-
Africa:  334824358.0
Asia:  2365829752.0
Europe:  2096917275.0
North_America:  2372819211.0
South_America:  0.0
Oceania:  6099419.0


Total Deaths:-
Africa:  7995472.0
Asia:  43350826.0
Europe:  72184171.0
North_America:  73961123.0
South_America:  0.0
Oceania:  153400.0
176      0.003550
488      0.008062
813      0.003172
1132     0.068595
1433     0.000274
           ...
58744    0.000027
59103    0.010028
59383    0.001800
59686    0.001436
```

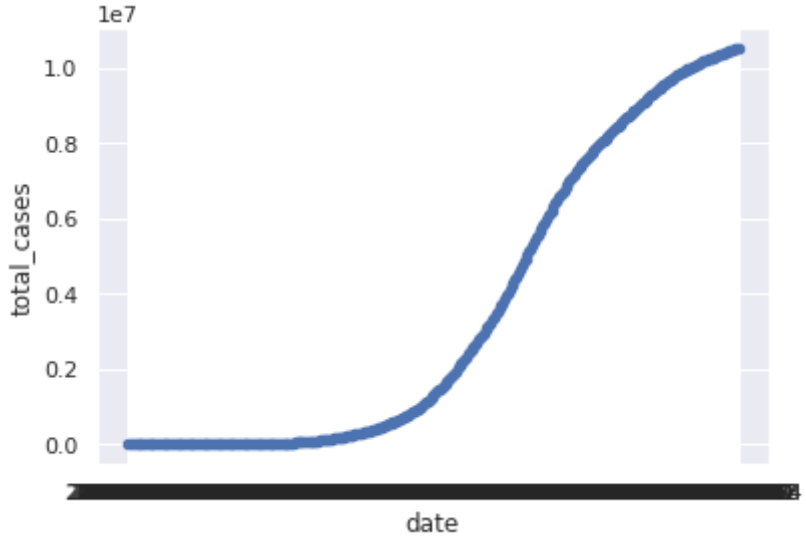
```
59987    0.000949
Length: 187, dtype: float64
       iso_code continent  ... human_development_index mortality_rate
176         AFG      Asia  ...                   0.498       0.003550
488         ALB    Europe  ...                   0.785       0.008062
813         DZA    Africa  ...                   0.754       0.003172
1132        AND    Europe  ...                   0.858       0.068595
1433        AGO    Africa  ...                   0.581       0.000274
...         ...       ...  ...                     ...            ...
58744       VNM      Asia  ...                   0.694       0.000027
59103  OWID_WRL       NaN  ...                     NaN       0.010028
59383       YEM      Asia  ...                   0.452       0.001800
59686       ZMB    Africa  ...                   0.588       0.001436
59987       ZWE    Africa  ...                   0.535       0.000949

[187 rows x 56 columns]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: SettingWithC
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
```

Task 7:

1. On the India's dataframe, plot the total number of cases, date wise
2. Save the India Dataframa as CSV.

```
### Your Code Here.
India = dataset.loc[dataset['iso_code'] == 'IND']
import matplotlib.pyplot as plt
df = pd.DataFrame(India,columns=['date','total_cases'])
df.plot(x ='date', y='total_cases', kind = 'scatter')
plt.show()
India.to_csv('file1.csv')
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should b
```



4.0.1

Task 8:

1. Any intresting finding/Oberservation from the dataset

```
Total number of cases increases at rapid rate initially
but now it is saturating. Basically, it represents the
normal curve which increases rapildly initially and becomes
saturated, finally decreases (This is bell shaped curved)
```