IS 508 PROJECT & ORGANIZATION CHANGE MANAGEMENT

Nangosha Elisha May21/PGDIT/577U

In this test, you will design models to predict the number of deaths due to covid-19 Pandemic in Americas region. Data set can be downloaded from URL: https://covid19.who.int/table • Use python 3.7 or higher for implementation. • Take the screen shots of your codes and upload in the university platform and also share your uploaded codes/output in your Github account with the Instructor. • Use the following methodology with respective performance evaluation metrics such as r-squared, and RMSE:

1. Descriptive statistics (4 marks)
2. Simple linear Regression. (8 marks)
3. Multiple Linear Regression. (8 marks)

Data set from URL: https://covid19.who.int/table As at: July 10th 2021 at 6.01 PM

In [1]:
```
#importing libraries for analysis
! pip install seaborn
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,PolynomialFeatures
%matplotlib inline
```

```
Requirement already satisfied: seaborn in /srv/conda/envs/notebook/lib/python3.6/site-packages (0.11.1)
Requirement already satisfied: pandas>=0.23 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from seaborn) (1.1.5)
Requirement already satisfied: scipy>=1.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib>=2.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from seaborn) (3.3.4)
Requirement already satisfied: numpy>=1.15 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from seaborn) (1.19.5)
Requirement already satisfied: kiwisolver>=1.0.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /srv/conda/envs/notebook/lib/python3.6/site-packages/cycler-0.10.0-py3.6.egg (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib>=2.2->seaborn) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib>=2.2->seaborn) (8.2.0)
Requirement already satisfied: python-dateutil>=2.1 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from matplotlib>=2.2->seaborn) (2.8.1)
Requirement already satisfied: six in /srv/conda/envs/notebook/lib/python3.6/site-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.15.0)
Requirement already satisfied: pytz>=2017.2 in /srv/conda/envs/notebook/lib/python3.6/site-packages (from pandas>=0.23->seaborn) (2021.1)
```

In [2]:
```
#Load file. This has global regions
Data=pd.read_csv("WHO COVID-19 global table data July 10th 2021 at 12.02.11 PM.csv",sep=",")
#view first five rows
Data.head()
```

Out[2]:

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours	Transmission Classification
0	Global	NaN	185291530	2377.196364	2906590	37.290076	458355	4010834	51.456966	55457	0.711485	8516	NaN
1	United States of America	Americas	33451965	10106.250000	108004	32.630000	22569	601231	181.640000	1551	0.470000	301	Community transmission
2	India	South-East Asia	30752950	2228.470000	294699	21.350000	43393	405939	29.420000	5627	0.410000	911	Clusters of cases
3	Brazil	Americas	18909037	8895.880000	351896	165.550000	54022	528540	248.660000	10474	4.930000	1648	Community transmission
4	Russian Federation	Europe	5733218	3928.630000	171858	117.760000	25766	141501	96.960000	4936	3.380000	726	Clusters of cases

Cleaning of imported Data

In [5]:
```
#Extracting only for Americas region
new_file = Data.loc[(Data['WHO Region'] == 'Americas')]
#preview first five rows of new data set
new_file.head()
```

Out[5]:

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours	Transmission Classification
1	United States of America	Americas	33451965	10106.25	108004	32.63	22569	601231	181.64	1551	0.47	301	Community transmission
3	Brazil	Americas	18909037	8895.88	351896	165.55	54022	528540	248.66	10474	4.93	1648	Community transmission
8	Argentina	Americas	4593763	10164.14	123389	273.01	19423	97439	215.59	3135	6.94	456	Community transmission
9	Colombia	Americas	4426811	8700.00	185829	365.21	24229	110578	217.32	4034	7.93	559	Community transmission
15	Mexico	Americas	2558369	1984.27	39100	30.33	8507	234192	181.64	1145	0.89	234	Community transmission

In [6]:
```
#Renaming some columns to make it easy to read'
new_file.rename(columns = {'Name':'Country','Cases - cumulative total':'Cases(Cum.Total)',
                           'Cases - cumulative total per 100000 population':'Cases(Cum.Total)per 100000 population',
                           'Cases - newly reported in last 7 days':'Cases(last 7 days)',
                           'Cases - newly reported in last 7 days per 100000 population':'Cases(last 7 days)per 100000 population',
                           'Cases - newly reported in last 24 hours':'Cases(last 24 hours)',
                           'Deaths - cumulative total':'Deaths(Cum.Total)','Deaths - cumulative total per 100000 population':
                           'Deaths(Cum.Total) per 100000 population', 'Deaths reported in last 7 days':
                           'Deaths(last 7 days)','Deaths - newly reported in last 7 days per 100000 population':
                           'Deaths(last 7 days)per 100000 population','Deaths - newly reported in last 24 hours':
                           'Deaths(last 24 hours)'}, inplace = True)
```

```
/srv/conda/envs/notebook/lib/python3.6/site-packages/pandas/core/frame.py:4308: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  errors=errors,
```

In [7]:
```
#First five raws
new_file.head()
```

Out[7]:

	Country	WHO Region	Cases(Cum.Total)	Cases(Cum.Total)per 100000 population	Cases(last 7 days)	Cases(last 7 days)per 100000 population	Cases(last 24 hours)	Deaths(Cum.Total)	Deaths(Cum.Total) per 100000 population	Deaths - newly reported in last 7 days	Deaths(last 7 days)per 100000 population
1	United States of America	Americas	33451965	10106.25	108004	32.63	22569	601231	181.64	1551	0.47
3	Brazil	Americas	18909037	8895.88	351896	165.55	54022	528540	248.66	10474	4.93
8	Argentina	Americas	4593763	10164.14	123389	273.01	19423	97439	215.59	3135	6.94
9	Colombia	Americas	4426811	8700.00	185829	365.21	24229	110578	217.32	4034	7.93
15	Mexico	Americas	2558369	1984.27	39100	30.33	8507	234192	181.64	1145	0.89

In [9]:
```
#Data set Summary
new_file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 56 entries, 1 to 220
Data columns (total 13 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Country                                 56 non-null     object
 1   WHO Region                              56 non-null     object
 2   Cases(Cum.Total)                        56 non-null     int64
 3   Cases(Cum.Total)per 100000 population   56 non-null     float64
 4   Cases(last 7 days)                      56 non-null     int64
 5   Cases(last 7 days)per 100000 population 56 non-null     float64
 6   Cases(last 24 hours)                    56 non-null     int64
 7   Deaths(Cum.Total)                       56 non-null     int64
 8   Deaths(Cum.Total) per 100000 population 56 non-null     float64
 9   Deaths - newly reported in last 7 days  56 non-null     int64
 10  Deaths(last 7 days)per 100000 population 56 non-null    float64
 11  Deaths(last 24 hours)                   56 non-null     int64
 12  Transmission Classification             56 non-null     object
dtypes: float64(4), int64(6), object(3)
memory usage: 6.1+ KB
```

In [10]:
```
#Countrie in Americas region as per dataset
new_file['WHO Region'].value_counts().to_frame()
```

Out[10]:

	WHO Region
Americas	56

We have 56 countries in America region

In [11]:
```
#Data Set Description
new_file.describe()
```

Out[11]:

	Cases(Cum.Total)	Cases(Cum.Total)per 100000 population	Cases(last 7 days)	Cases(last 7 days)per 100000 population	Cases(last 24 hours)	Deaths(Cum.Total)	Deaths(Cum.Total) per 100000 population	Deaths - newly reported in last 7 days	Deaths(last 7 days)per 100000 population	Dea 2
count	5.600000e+01	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000	56
mean	1.311608e+06	4099.591964	17272.714286	90.348393	2733.410714	34481.214286	77.229286	451.392857	1.582500	70
std	5.112890e+06	3312.030801	55956.745585	202.678019	8633.177160	112121.113116	95.639841	1547.820305	2.377306	240
min	7.000000e+00	102.940000	0.000000	0.000000	-9.000000	0.000000	0.000000	0.000000	0.000000	0
25%	2.086500e+03	1395.055000	10.750000	9.005000	0.000000	17.750000	11.247500	0.000000	0.000000	0
50%	1.556400e+04	3181.080000	310.000000	33.020000	25.000000	261.000000	53.310000	3.500000	0.445000	0
75%	3.425358e+05	6268.192500	7568.250000	92.862500	955.500000	5968.250000	93.797500	122.250000	1.912500	18
max	3.345196e+07	10785.670000	351896.000000	1465.090000	54022.000000	601231.000000	587.600000	10474.000000	10.520000	1648

In [12]:
```
new_file.shape
```

Out[12]:
```
(56, 13)
```

In [13]:
```
#Transmission classification Counts
new_file['Transmission Classification'].value_counts().to_frame()
```

Out[13]:

	Transmission Classification
Community transmission	42
Sporadic cases	5
Clusters of cases	5
No cases	4

We have 42 community transmissions out of the total of 56 records observed.

In [14]:
```
#The country with the highest number of deaths reported in the last 24hrs against transmission classification
death_24hrs=new_file[['Country','Deaths(last 24 hours)']]
highdeath_24hrs = death_24hrs.sort_values(by=['Deaths(last 24 hours)'], ascending= False)

# a. check first
highdeath_24hrs.head()
```

Out[14]:

	Country	Deaths(last 24 hours)
3	Brazil	1848
9	Colombia	559
8	Argentina	456
1	United States of America	301
15	Mexico	234

```
Brazil reported the highest death:1,648 while Puerto Rico registered the lowest 1 in last 24hours as of data report of July 10th 2021
```

2. Simple Linear regression

We shall use the formula:

y=mx+c y=dependent variable, m=slope, x=independent variable, c=y intercept

In [16]:
```
#Confirm importation of model library
from sklearn.linear_model import LinearRegression
```

In [20]:
```
#Prediction of deaths in the next 24 hours using newly reported cases in last 24 hours
x = new_file[['Cases(last 24 hours)']]
y = new_file['Deaths(last 24 hours)']
lm = LinearRegression()
lm.fit(x,y)
lm
```

Out[20]:
```
LinearRegression()
```

In [19]:
```
#Calculating the intercept, c
lm.intercept_
```

Out[19]:
```
-4.010264462076975
```

In [21]:
```
#Calculating the slope, m
lm.coef_
```

Out[21]:
```
array([0.02708923])
```

Model formular;

Deaths(last 24 hours)= 0.02708923 * Cases(last 24 hours) - 4.010264462076975

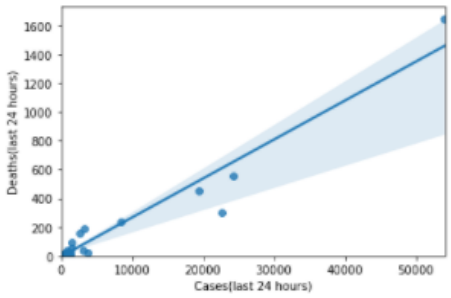
In [23]:
```
#The correlation
new_file[['Cases(last 24 hours)','Deaths(last 24 hours)']].corr()
```

Out[23]:

	Cases(last 24 hours)	Deaths(last 24 hours)
Cases(last 24 hours)	1.000000	0.972184
Deaths(last 24 hours)	0.972184	1.000000

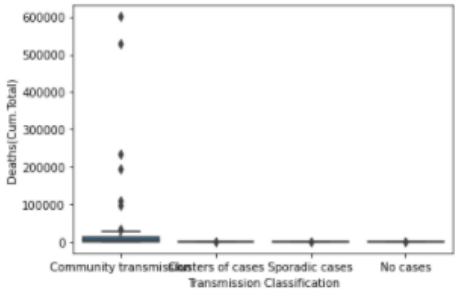
In [25]:
```
#The simple linear regression graph
sns.regplot(x="Cases(last 24 hours)", y="Deaths(last 24 hours)", data=new_file)
plt.ylim(0,)
```

Out[25]:
```
(0.0, 1731.0014913464813)
```



In [26]:
```
#plotting a box plot
sns.boxplot(x="Transmission Classification", y="Deaths(Cum.Total)", data=new_file)
```

Out[26]: <AxesSubplot:xlabel='Transmission Classification', ylabel='Deaths(Cum.Total)'>



The Distribution of Deaths(cumu.Total) between cluster transmission and Community transmission, are elaborate enough to take Transmission classification as a potential good predictor of Cummulative total Deaths.

Model performance evaluation metrics

In [28]:
```
#R-squared Calculation
print('The R-squared is: ', lm.score(x, y))
```

```
The R-squared is:  0.9451426883656249
```

In [29]:
```
#Importation of function mean_squared_error % sqrt from the module metrics
from sklearn.metrics import mean_squared_error #for calculating MSE
from math import sqrt #for calculating RSME
```

In [30]:
```
#calculating MSE
y=lm.predict(x)
mse = mean_squared_error(new_file['Deaths(last 24 hours)'], y)
print('Mean Squared Error:', mse)
```

```
Mean Squared Error: 3117.7858489231335
```

In [31]:
```
#calculating RMSE
#since; mse=sqrt(mean_squared_error(new_df['Deaths(last 24 hours'], y))
print('Root Mean Squared Error:',sqrt(mse))
```

```
Root Mean Squared Error: 55.8371368259793
```

3. Multiple Linear Regression

In this case we are predicting Deaths(last 24 hours) in America by looking at three variables;

Cases reported in the last 24 hours; Cases(last 24 hours) Cases reported in the last 7 days; Cases(last 7 days) Total cumulative Cases reported; Cases(Cum.Total)

Lets use this formula;

$y=b1x1+b2x2+b3x3+c$ y=dependent-variable, b1,b2,b3=slope, x1,x2,x3=independent-variables, c=y-intercept

In [32]:
```
# We are assigning the variables identified above; in a new dataset,z
z = new_file[['Cases(last 24 hours)','Cases(last 7 days)','Cases(Cum.Total)']]
#Fitting the linear model using the three above variables.
lm.fit(z, new_file['Deaths(last 24 hours)'])
```

Out[32]:
```
LinearRegression()
```

In [33]:
```
#Calculating the intercept, c
lm.intercept_
```

Out[33]:
```
-3.152937889926207
```

In [34]:
```
#Calculating the slope; b1,b2,b3
lm.coef_
```

Out[34]:
```
array([ 4.70509139e-02, -2.10651332e-03, -1.45131908e-05])
```

 # Model formular;

```
Deaths(last 24 hours)= 4.70509139 * Cases(last 24 hours) - 2.10651332 * Cases(last 7 days) - 1.45131908 * Cases(Cum.Total) - 3.152937889926207
```

In [36]:
```
#correlation
new_file[['Cases(last 24 hours)','Cases(last 7 days)','Cases(Cum.Total)']].corr()
```

Out[36]:

	Cases(last 24 hours)	Cases(last 7 days)	Cases(Cum.Total)
Cases(last 24 hours)	1.000000	0.992458	0.745889
Cases(last 7 days)	0.992458	1.000000	0.673512
Cases(Cum.Total)	0.745889	0.673512	1.000000

In [37]:
```
#calculate R-squared
print('The R-squared is:',lm.score(z, new_file['Deaths(last 24 hours)']))
```

```
The R-squared is: 0.9707863962523477
```

In [38]:
```
#Calculate: The Mean Squared Error; MSE
#prediction based on model
y_predict_multifit = lm.predict(z)
```

In [39]:
```
#compare the predicted results with the actual results and print
mse=mean_squared_error(new_file['Deaths(last 24 hours)'], y_predict_multifit)
print('Mean Squared Error:',mse)
```

```
Mean Squared Error: 1660.3394815906993
```

In [40]:
```
#calculate: RMSE
#mse=mean_squared_error(new_df['Deaths(last 24 hours)'], y_predict_multifit), we call mse
print('Root Mean Squared Error:',sqrt(mse))
```

```
Root Mean Squared Error: 40.74726348591644
```

The End

Analysis by Elisha Nangosha Email: nangoshaelisha@gmail.com Contact: 0777514549

In [ ]: