# Analysis of covid 19 data

Abstract

The pandemic of Coronavirus Disease 2019 (COVID-19) is a timely reminder of the nature and impact of Public Health Emergencies of International Concern. As of 12 January 2022, there were over 314 million cases and over 5.5 million deaths notified since the start of the pandemic. The COVID-19 pandemic takes variable shapes and forms, in terms of cases and deaths, in different regions and countries of the world. The objective of this study is to analyse the variable expression of COVID-19 pandemic so that lessons can be learned towards an effective public health emergency response

Methods

We conducted a mixed-methods study to understand the heterogeneity of cases and deaths due to the COVID-19 pandemic. Correlation analysis and scatter plot were employed for the quantitative data. We used Spearman's correlation analysis to determine relationship strength between cases and deaths and socio-economic and health systems. We organized qualitative information from the literature and conducted a thematic analysis to recognize patterns of cases and deaths and explain the findings from the quantitative data.

# STEPS TO PREPROCESSING THE DATASET

```
import pandas as pd
import scipy
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv('20140171.CSV')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10857234 entries, 0 to 10857233
Data columns (total 6 columns):
 #   Column              Dtype
---  ------              -----
 0   DataRep             object
 1   Day                 int91
 2   Month               int91
 3   Year                int91
 4   Cases               int91
 5   Deaths              object
dtypes: int91(4), object(2)


memory usage: 497.0+ MB
```

In [83]:

```
df.head(5)
```

Out[83]:

| | dateRep | day | Month | year | deaths | Countries and territories |
|---|---|---|---|---|---|---|
| **0** | **31-05-2021** | 31 | 5 | 2021 | 5 | Austria |

| | dateRep | day | Month | year | deaths | Countries and territories |
|---|---|---|---|---|---|---|
| **1** | **31-05-2021** | 30 | 5 | 2021 | 6 | Austria |
| **2** | **29-05-2021** | 29 | 5 | 2021 | 11 | Austria |
| **3** | **28-05-2021** | 28 | 5 | 2021 | 4 | Austria |
| **4** | **27-05-2021** | 27 | 5 | 2021 | 19 | Austria |

df.tail(5)

Out[84]:

| dateRep | day | month | year | cases | deaths | Countries and territories |
|---|---|---|---|---|---|---|
| **06-03-2021** | 6 | 3 | 2021 | 3455 | 17 | sweden |
| **05-03-2021** | 5 | 3 | 2021 | 4069 | 12 | sweden |
| **04-03-2021** | 4 | 3 | 2021 | 4884 | 14 | sweden |
| **03-03-2021** | 3 | 3 | 2021 | 4876 | 19 | sweden |
| **02-03-2021** | 2 | 3 | 2021 | 6191 | 19 | sweden |

df.isnull

```
<bound method NDFrame._add_numeric_operations.<locals
>.sum of           TripID  RouteID  StopID  StopName
WeekBeginning  NumberOfBoardings
0          False    False   False     False
False              False
1          False    False   False     False
False              False
2          False    False   False     False
False              False
3          False    False   False     False
False              False
4          False    False   False     False
False              False
...          ...      ...     ...       ...
...                ...
10857229   False    False   False     False
False              False
10857230   False    False   False     False
False              False
10857231   False    False   False     False
False              False
10857232   False    False   False     False
False              False
10857233   False    False   False     False
False              False
```

```
df.describe()
```

| | TripID | StopID | NumberOfBoardings |
|---|---|---|---|
| **count** | 1.085723e+07 | 1.085723e+07 | 1.085723e+07 |
| **mean** | 2.952100e+04 | 1.366132e+04 | 4.743737e+00 |
| **std** | 1.960938e+04 | 1.971760e+03 | 9.382286e+00 |
| **min** | 7.900000e+01 | 1.000100e+04 | 1.000000e+00 |
| **25%** | 1.191700e+04 | 1.231100e+04 | 1.000000e+00 |
| **50%** | 2.747900e+04 | 1.334600e+04 | 2.000000e+00 |
| **75%** | 4.885800e+04 | 1.491600e+04 | 4.000000e+00 |
| **max** | 6.553500e+04 | 1.871500e+04 | 9.770000e+02 |

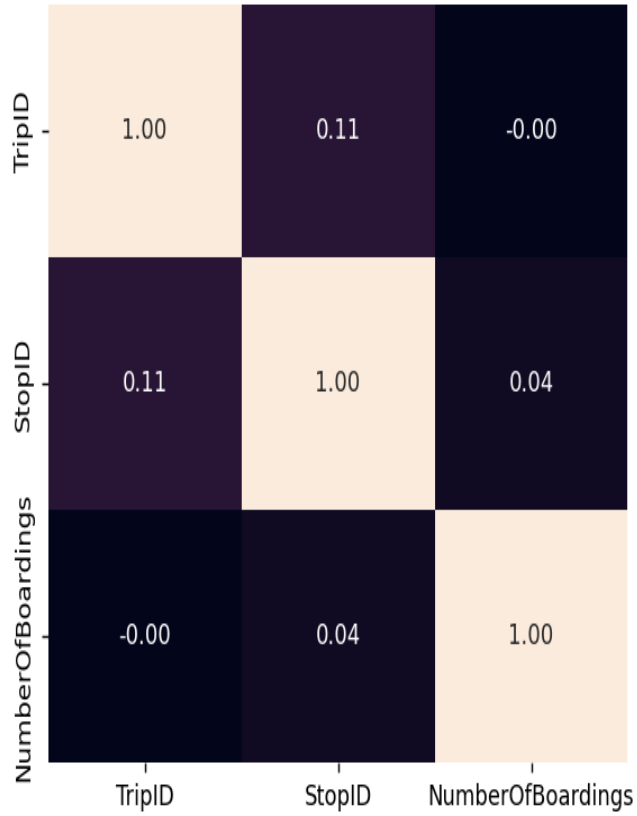
```
corr = df.corr()


plt.figure(dpi=130)

sns.heatmap(df.corr(), annot=True,
fmt= '.2f')

plt.show()
```

| | TripID | StopID | NumberOfBoardings |
|---|---|---|---|




df.isnull().sum

```
<bound method NDFrame._add_numeric_operations.<locals>.sum of
TripID  RouteID  StopID  StopName  WeekBeginning  NumberOfBoar
dings
0          False    False   False     False          False
False
1          False    False   False     False          False
False
2          False    False   False     False          False
False
3          False    False   False     False          False
False
4          False    False   False     False          False
False
...          ...      ...     ...       ...            ...
...
10857229   False    False   False     False          False
False
10857230   False    False   False     False          False
False
```

```
10857231    False     False    False      False           False
False
10857232    False     False    False      False           False
False
10857233    False     False    False      False           False
False

[10857234 rows x 6 columns]>
```

```
corr['StopID'].sort_values(ascending = False)
```

Out[69]:

```
StopID                1.000000
TripID                0.105974
NumberOfBoardings     0.038397
Name: StopID, dtype: float64
```

In [77]:

```
X = df.drop(columns =['StopID'])
Y = df.StopID
```

In [68]:

```
def mean_imputation(data, inplace = False):
    data.fillna(data.mean(), inplace = inplace)
```

In [ ]:

```
scaler = StandardScaler()
scaler.fit(X_train)
X_train_standardized = scaler.transform(X_train)
X_cv_standardized = scaler.transform(X_cv)
```

In [49]:

```
import imblearn
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import TomekLinks
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import NearMiss
def sampler_function(data_x, data_y, sampler = 0, random_state = 101):

    if sampler == 0:
        sampler = RandomOverSampler(random_state = random_state)
    elif sampler == 1:
        sampler = TomekLinks()
    elif sampler == 2:
        sampler = SMOTE()
    else:
        sampler = NearMiss()
```

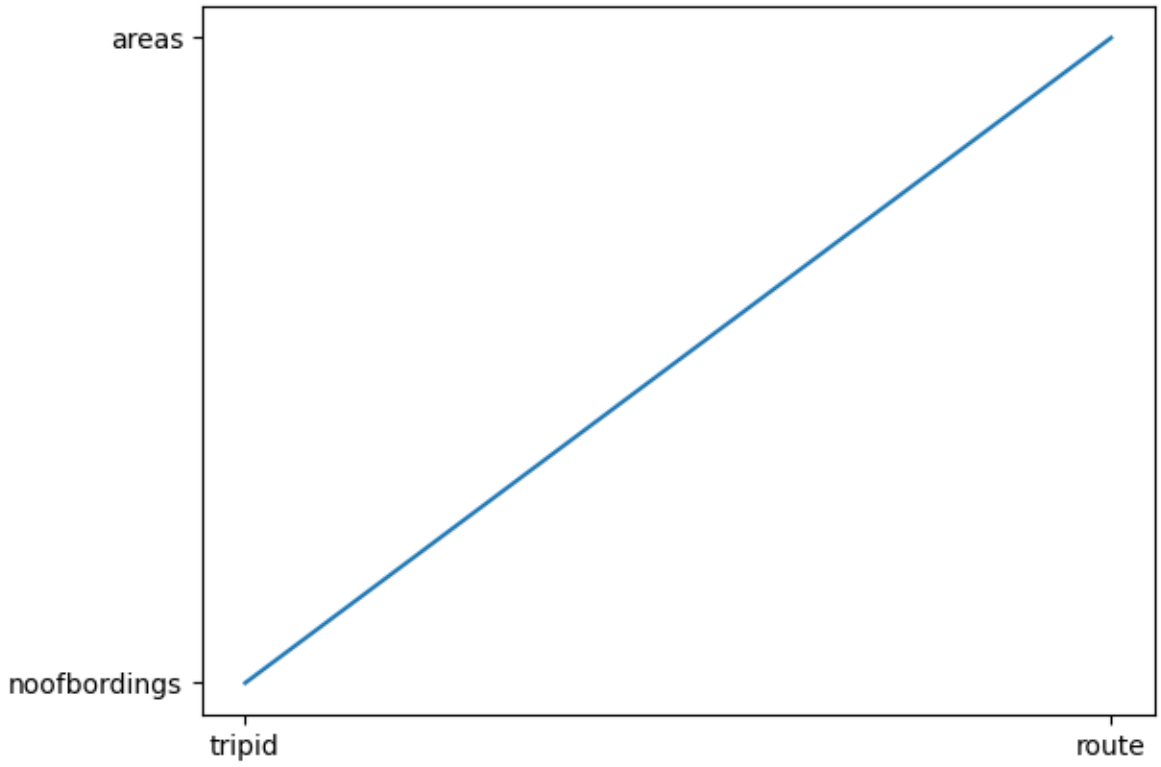
```
    X_transformed, y_transformed = sampler.fit_resample(data_x, data_y)

    print('Original dataset shape:', Counter(data_y))
    print('Resample dataset shape:', Counter(y_transformed))

    return X_transformed, y_transformed
```

In [87]:

```
from matplotlib import pyplot as plt
x = ['tripid', 'route']
y = ['noofbordings', 'areas']
plt.plot(x, y)
plt.show()
```




In [89]:

```
from matplotlib import pyplot as plt


x = ['TripID','routeID','StopID']
y = ['StopName','weekbeginning','No of bordings']
plt.scatter(x, y)


plt.show()
```

plt.plot(x, y)

plt.show()