# Group Members

Lewis Gitonga Muriungi SCT222-0123/2020

Jemimah Asiko SCT-0204/2020

Ron Mbatia SCT222-0205/2020

Ted Wainoga SCT222-0247/2020

# Flow

## Getting Data

We scoured the internet to find a good dataset that fit the following criteria:

1. Had Kenyan data
2. Had death cases
3. Had confirmed cases
4. Had recovery cases – ideally

The data we found was https://covid19.who.int/WHO-COVID-19-global-data.csv from World Health Organization

```
[ ] import pandas as pd
```

```
[ ] data = pd.read_csv('covid.csv')
```

```
[ ] data.head
```

```
<bound method NDFrame.head of        Date_reported Country_code      Country WHO_region  New_cases  \
0         2020-01-03           AF  Afghanistan       EMRO          0
1         2020-01-04           AF  Afghanistan       EMRO          0
2         2020-01-05           AF  Afghanistan       EMRO          0
3         2020-01-06           AF  Afghanistan       EMRO          0
4         2020-01-07           AF  Afghanistan       EMRO          0
...              ...          ...          ...        ...        ...
336535    2023-11-18           ZW     Zimbabwe       AFRO          0
336536    2023-11-19           ZW     Zimbabwe       AFRO          0
336537    2023-11-20           ZW     Zimbabwe       AFRO          0
336538    2023-11-21           ZW     Zimbabwe       AFRO          0
336539    2023-11-22           ZW     Zimbabwe       AFRO          0

        Cumulative_cases  New_deaths  Cumulative_deaths
0                      0           0                  0
1                      0           0                  0
2                      0           0                  0
3                      0           0                  0
4                      0           0                  0
...                  ...         ...                ...
336535            265890           0               5725
336536            265890           0               5725
336537            265890           0               5725
336538            265890           0               5725
336539            265890           0               5725

[336540 rows x 8 columns]>
```

## Pre-Processing the Data

The library used here was Pandas

The first step was to read the data

We explored the dataset

We filtered the data to include only Kenyan data

We chose the columns we would use

We added the recoveries column by subtracting deaths from cases

We filtered to remove 0s from the cases column

Lastly, we renamed the columns for easier use

```
[ ] data = data.loc[data['Country_code'] == "KE"]
```

```
[ ] features = ["Cumulative_cases","Cumulative_deaths"]
    data = data[features]
    data
```

```
[ ] # Create column cumulative recoveries
    data['Cumulative_recoveries'] = data['Cumulative_cases'] - data['Cumulative_deaths']
    data
```

```
# Remove all rows where cumulative cases is 0
data = data.loc[data['Cumulative_cases'] != 0]
data
```

| | Cumulative_cases | Cumulative_deaths | Cumulative_recoveries |
|---|---|---|---|
| **154851** | 1 | 0 | 1 |
| **154852** | 1 | 0 | 1 |
| **154853** | 3 | 0 | 3 |
| **154854** | 3 | 0 | 3 |
| **154855** | 4 | 0 | 4 |
| ... | ... | ... | ... |
| **156195** | 344077 | 5689 | 338388 |
| **156196** | 344077 | 5689 | 338388 |
| **156197** | 344077 | 5689 | 338388 |
| **156198** | 344077 | 5689 | 338388 |

## Model training and visualization

The library involved was sckit-learn

To be more specific LinearRegression and train_test_split

We divided the data into train_data and test_data using train_test_split

We fit the model, using the train_data

We used the model to predict the test_data

Visualizing the model's prediction alongside the test data

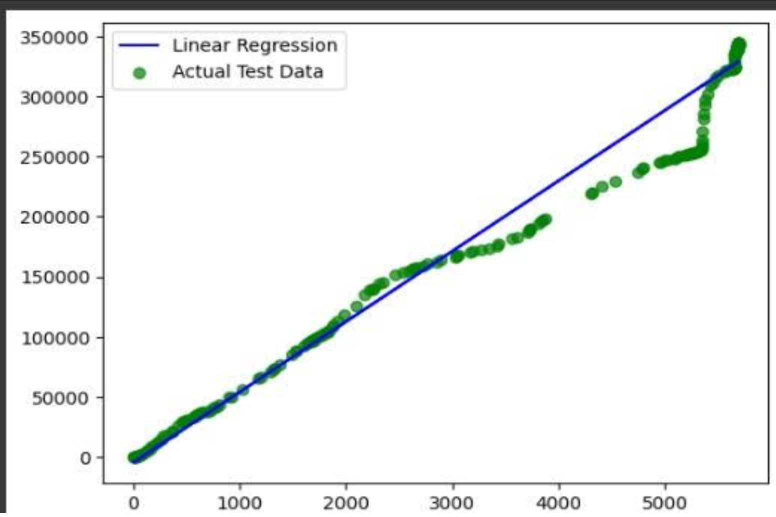
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
```

```
X_train,X_test,y_train,y_test = train_test_split(data.deaths,data.cases)
```

```
#Create linear model
model = LinearRegression()
model.fit(X_train.values.reshape(-1,1),y_train.values)
```

```
#Use model to predict test data
prediction = model.predict(X_test.values.reshape(-1,1))

plt.plot(X_test,prediction,label="Linear Regression",color='b')
plt.scatter(X_test,y_test,label="Actual Test Data",color='g',alpha=.7)
plt.legend()
plt.show()
```




## Data interpretation

The model was pretty on spot roughly 80%.

We can deduce that when cases hit above 150,00 deaths sky rocketed above the prediction. This is a speculation of as cases rise the pool of infection rate also rises hence leading to more deaths

This may be due to constraints such as limited resources to combat high number of viral cases thus leading to more deaths

## Data Applications

Outbreaks – For example the Ebola outbreak in West Africa

This is because the model is used to predict high population of disease cases and deaths. Thus health bodies can adequately plan on how to accommodate them

Example: Gather adequate resources, like provision of masks during Covid-19 outbreak