

## Lab 11

In Lab 10 we worked through plotting the scatter plot for the selected data categories using our latest dataset as shown below.

```
plt.ylabel('Daily deaths')
plt.xlabel('total people vaccinated')
plt.title('e.g., daily deaths vs total people vaccinated')
datasetName = datasetName.dropna()

plt.scatter(datasetName ["people_fully_vaccinated"], datasetName
["new_deaths"], c = "b")

*****
*****
```

In Lab 11 we will be going a step further. We will compute the linear regression on the selected data categories using python's *stats* library (`stats.linregress(x,y)`) as in the code provided below.

**\*\*The r value points to the strength of the relationship between the two data categories (x,y)**

**First some more data cleaning to do.**

**On inspection, the following columns of our the dataset have constant values, blank, or are irrelevant to our task : We will drop them as we learnt in previous labs.**

```
finalDataset =
finalDataset.drop(["iso_code", "continent", "location", "weekly_icu_admissions", "weekly_icu_admissions_per_million", "weekly_hosp_admissions", "weekly_hosp_admissions_per_million", "tests_units", "excess_mortality", "handwashing_facilities", "population_density", "aged_65_older", "gdp_per_capita", "cardiovasc_death_rate", "diabetes_prevalence", "hospital_beds_per_thousand", "life_expectancy", "human_development_index", "stringency_index", "population", "median_age", "extreme_poverty"], axis = 1)
```

**You might want to display the size of the resulting dataset to ensure the drop was successful by using `finalDataset.head()` to display the first 5 records**

**Below is the new code to add to our previous code in Lab 10. Use the correct dataset name in place of `datasetName` and also make sure you use the correct column names as required for your group. Yours might not be the same as the example below (new-deaths vs people\_fully\_vacinated)**

```
y = np.array(datasetName["new_deaths"])
x = np.array(datasetName["people_fully_vaccinated"])
res = stats.linregress(x,y)
plt.plot(x, res.intercept + res.slope*x, 'r')
plt.legend(["Linear Regression", "Actual Values"])
plt.show()
print(f"R-value: {res.rvalue:.6f}")
```