Exercise 2

In this exercise, we will plot COVID hospital admissions in the UK from March 2020 to February 2023, and you will fit an exponential curve to understand the rise in hospital admissions in a three week period from late December 2021 to early January 2022.

The data is in a file called `hospital_cases_2023-02-16.csv` (comma-separated-values format). It was downloaded from the official website for UK COVID-19 data.

```
# If you are running this on Google Colab, uncomment and run the following lines; otherwise ignore this cell
# from google.colab import drive
# drive.mount('/content/drive')
```

```
import math
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

We will use `pandas`, a library for data analysis in Python to load and view the data. Pandas uses a structure called a *data frame* to represent and manipulate data tables. All the required commands are included here, so you won't need to learn Pandas for this exercise. But if you are interested in learning more, this is a good place to start.

```
df_hosp = pd.read_csv('hospital_cases_2023-02-16.csv')  # Create a data frame by loading data from a csv file
# If running on Google Colab change path to '/content/drive/MyDrive/IB-Data-Science/Exercises/hospital_cases_2023-02-16.csv'

df_hosp.head(3)   #display the first three rows
```

	date	hospitalCases
0	27/03/2020	7267
1	28/03/2020	8278
2	29/03/2020	9525

The command `pd.read_csv` loads the data onto a data frame. We have used the `.head()` command to display the top 3 rows of the data frame.

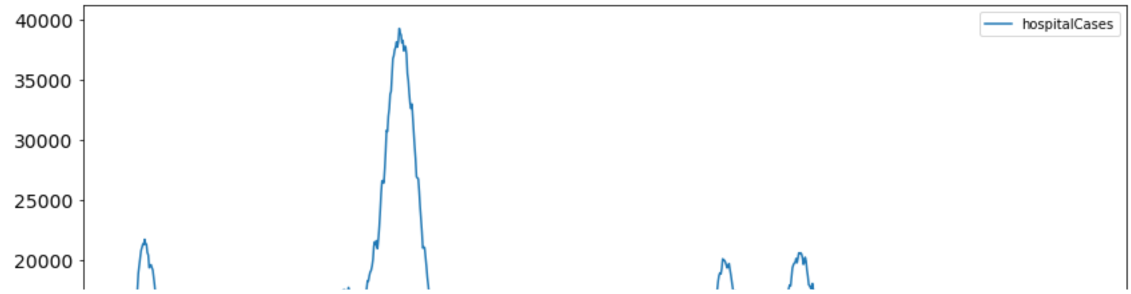
We can also display a random sample of rows from the data frame using `.sample()`, or the last few rows using `.tail()`.

```
df_hosp.tail(3)
```

	date	hospitalCases
1045	05/02/2023	7647
1046	06/02/2023	7795
1047	07/02/2023	7737

You can plot one column against another by just using their column names. Let us plot the `hospitalCases` column versus date.

```
plt.rcParams['figure.figsize'] = [14, 7]
plt.rcParams['axes.titlesize'] = 20
plt.rcParams['axes.labelsize'] = 20
plt.rcParams['xtick.labelsize'] = 14
plt.rcParams['ytick.labelsize'] = 14

df_hosp.plot(x='date', y='hospitalCases')
plt.show()
```



Observe the sharp increase in hospital admissions corresponding to each wave; also notice that the peaks after mid-2021 are smaller (due to the vaccines). We now extract the rows spanning a three week period starting 22 December 2021 (when Omicron first spread in the UK) into a data frame called `df_part`

```
df_part = df_hosp[635:656]
df_part.head(2)
```

	date	hospitalCases
635	22/12/2021	8400
636	23/12/2021	8436

```
df_part.tail(2)
```

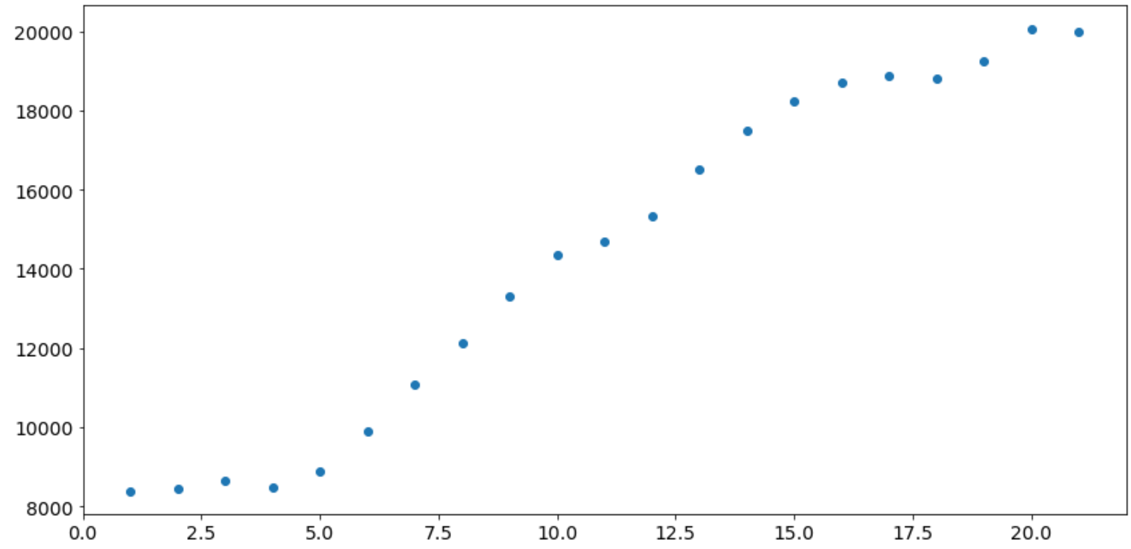
	date	hospitalCases
654	10/01/2022	20065
655	11/01/2022	19967

We now convert the `hospitalCases` column to a numpy array `yvals`

```
yvals = np.array(df_part['hospitalCases'])
N = np.size(yvals)
xvals = np.linspace(1,N,N) #an array containing the values 1,2....,N
```

2a) Plot the data `yvals` vs `xvals` in a scatterplot

```
# Your code for scatterplot here

plt.scatter(xvals, yvals) #Plotting the scatter of the data
plt.show()
```



2b) Fit an exponential model to the data

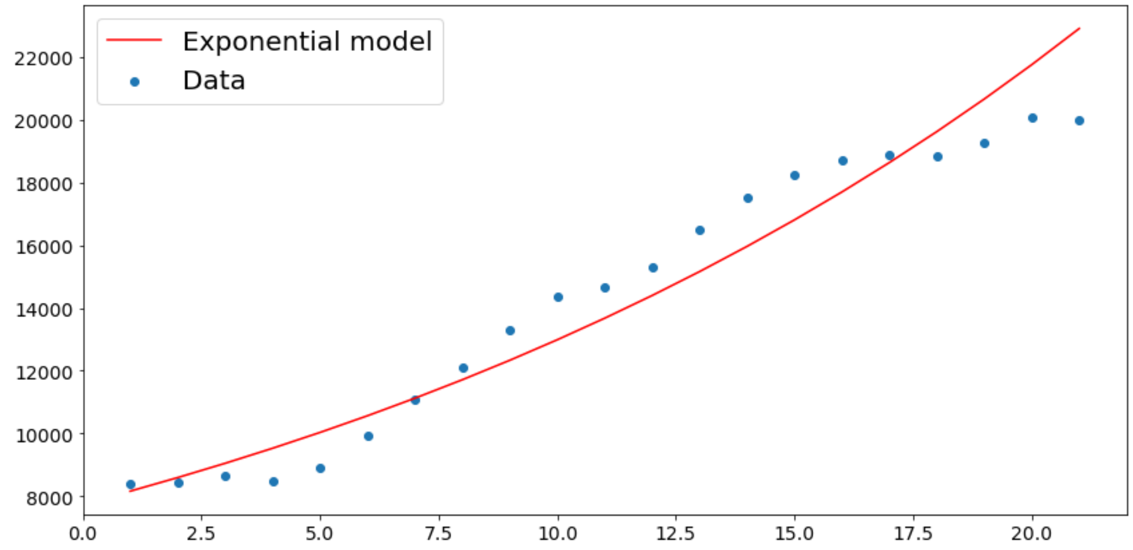
From our knowledge of how the virus spreads, we know that the number of infections, hospital admissions etc. should (roughly) follow an exponential curve. We would therefore like to fit a model of the form $y = c_1 e^{c_2 x}$, where y is the number of admissions on day x.

Note that this is a linear model on a log-scale for y. That is, $\log y = \log c_1 + c_2 x$.

- Fit a linear model for $\log(\mathrm{yvals})$ vs xvals, and print the values of c_1 and c_2
- Plot the fit $y = c_1 e^{c_2 x}$ along with the scatterplot of the data

```
def polyreg(data_matrix, k): # This is the polyreg function for exercise 1
    # Your code here
    # The function should return the the coefficient vector beta, the fit, and the vector of residuals

    data_x = data_matrix[:,0]
    data_y = data_matrix[:,1]

    all_ones = np.ones(np.shape(data_x))

    N = np.shape(data_matrix)[0]

    if k >= N:
        k = N - 1

    else:
        k = k

    data_x_array = []
    for i in range(k+1): # working it out for every vaule of k
            x_calc = data_x ** i
            if i == 0:
                 data_x_array.append(all_ones)
            else:
                 data_x_array.append(x_calc)

    x2_trans = np.vstack(data_x_array)
    x2 = np.transpose(x2_trans)

    beta_poly = np.linalg.lstsq(x2, data_y, rcond=None)[0]
    fit_poly = x2.dot(beta_poly)
    resid = data_y - fit_poly


    return beta_poly, fit_poly, resid

ln_yvals = np.vectorize(lambda x: np.log(x))(yvals) # Calculating the log value of the y values
data_matrix = np.swapaxes(np.array((xvals, ln_yvals)), 0, 1) # Getting the data matrix

beta_exp, fit_exp, resid_exp = polyreg(data_matrix, 1) # Getting the corresponding results from the polyreg for k = 1
c = [np.exp(beta_exp[0]), beta_exp[1]] # Calculating the constants
fit_exp = np.vectorize(lambda x: c[0]*np.exp(c[1]*x))(xvals) # Calculating the fit

print('The c_1 constant is:', np.round(c[0], 2)) # Printing the constant
print('The c_2 constant is:', np.round(c[1], 2))

plt.scatter(xvals, yvals, label = 'Data') #Printing the scatter and the fit
plt.plot(xvals, fit_exp, 'r', label = 'Exponential model')
plt.legend(fontsize = 20)
plt.show()
```

```
The c_1 constant is: 7754.3
The c_2 constant is: 0.05
```



2c) Estimate the weekly growth rate in hospital admissions (in %) over this period

Hint : According to the model, admissions increase every 7 days by a factor of $(c_1 e^{c_2(x+7)})/(c_1 e^{c_2 x}) = e^{7c_2}$.

```
# compute and print weekly growth rate (in %)
growth = (np.exp(7 * c[1]) - 1) * 100 #Calculating the growth over 7 days
growth_rounded = np.round(growth, 2) # Rounding it
print('The weekly growth of hospital admissions is ', growth_rounded, '%')
```

```
The weekly growth of hospital admissions is  43.47 %
```