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
```

```
/var/folders/ds/qvg7h1s93x16vvczwp8vr83m0000gn/T/
ipykernel_57903/188978642.py:4: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major
release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type,
and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at
https://github.com/pandas-dev/pandas/issues/54466

  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
```

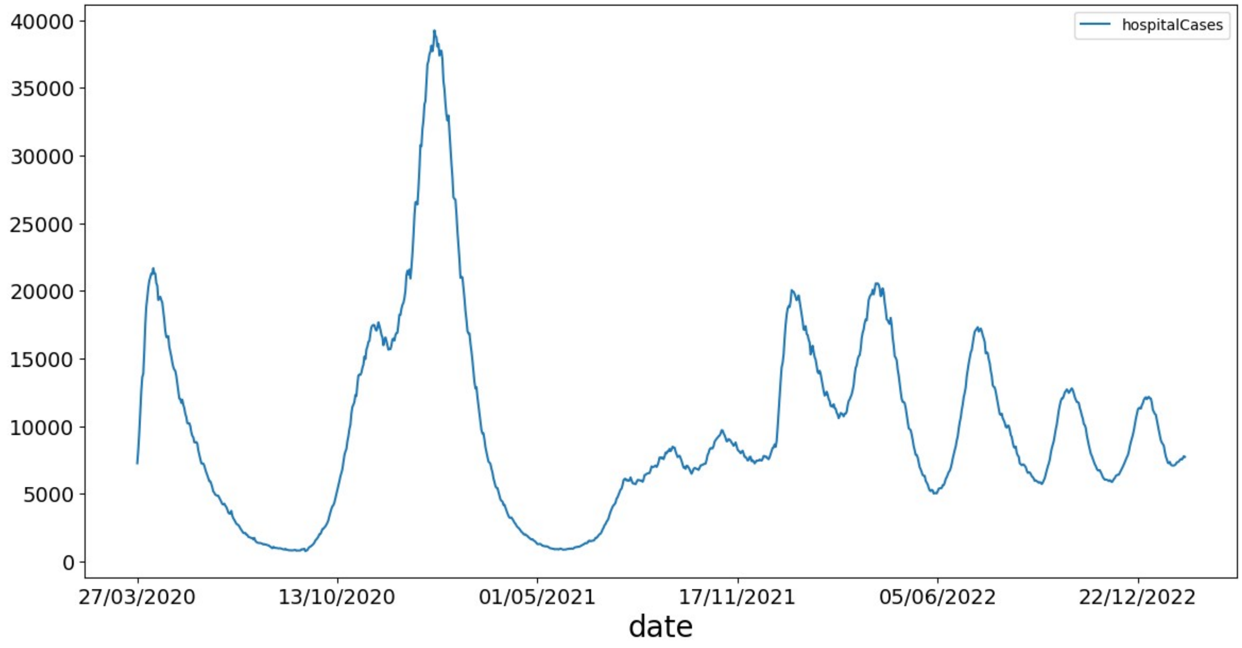
The command `pd.read_csv` loads the data onto a data frame. We have used the `.head()` command to display the top 3 rows of the data frame.

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)
```

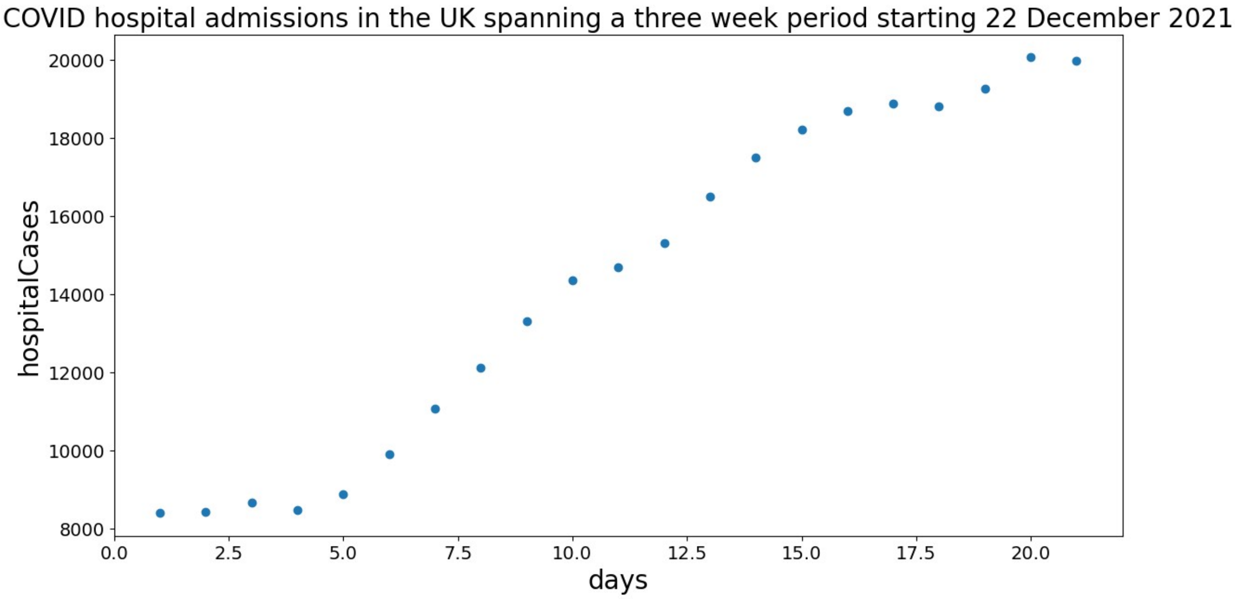
```
           date  hospitalCases
654  10/01/2022          20065
655  11/01/2022          19967
```

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)
plt.title("COVID hospital admissions in the UK spanning a three week
period starting 22 December 2021")
plt.xlabel("days")
plt.ylabel('hospitalCases')
plt.show()
```



2b) Fit an exponential model to the data

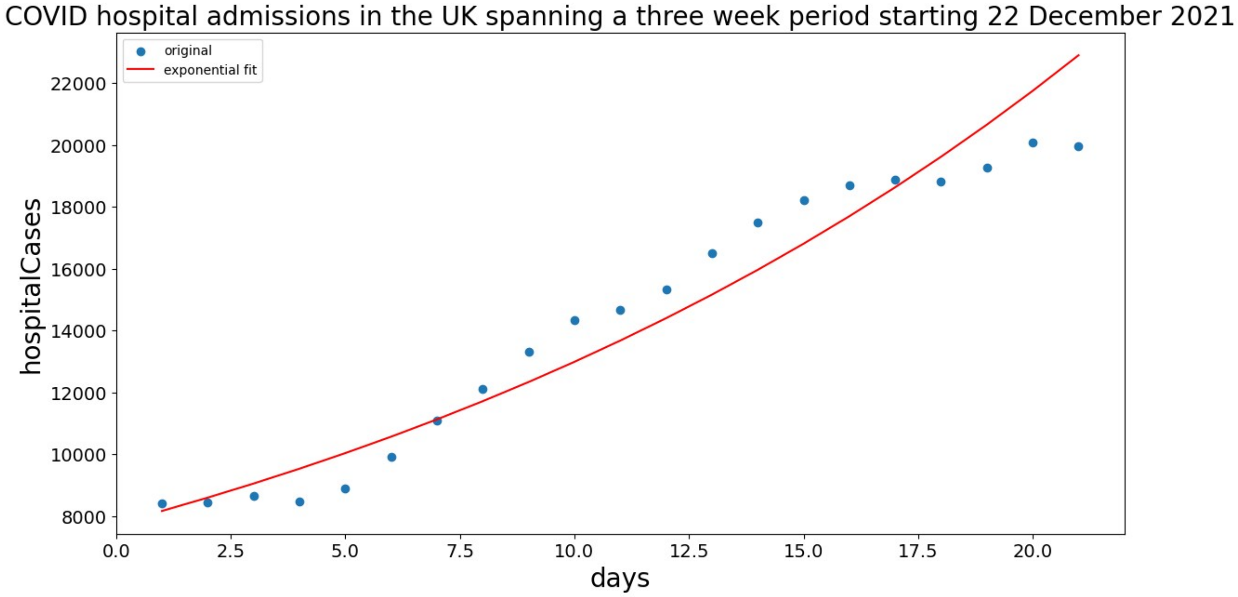
From our knowledge of how the virus spreads, we know that the number of infections, hospital admissions etc. should (roughly) follow an exponential curve. We would therefore like to fit a model of the form $y=c_1 e^{c_2 x}$, where y is the number of admissions on day x.

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

```
# your code here
def exponentialreg(xvals, yvals):
    # Your code here
    # The function should return the the coefficient vector beta, the fit, and the vector of residuals
    log_yvals = np.log(yvals)
    lsq_matrix = np.column_stack((xvals**0, xvals))
    lsq_matrix_T = lsq_matrix.T
    optimal_params = np.linalg.inv(lsq_matrix_T.dot(lsq_matrix)).dot(lsq_matrix_T).dot(log_yvals)

    log_yvals_pred = lsq_matrix.dot(optimal_params.T)
    yvals_pred = np.exp(log_yvals_pred)
    residuals = yvals - yvals_pred

    c1 = np.exp(optimal_params[0])
    c2 = optimal_params[1]

    return (c1,c2), yvals_pred, residuals

best_fit_params, y_vals_pred, residuals = exponentialreg(xvals, yvals)
print(f"c1: {best_fit_params[0]} | c2: {best_fit_params[1]}")

SSE = np.linalg.norm(residuals)**2
var = np.var(yvals)
R2 = 1 - SSE/(var * yvals.shape[-1])
print(f"SSE: {SSE} | R2: {R2}")

# Your code for scatterplot here
plt.scatter(xvals, yvals, label="original")
plt.plot(xvals, y_vals_pred, label="exponential fit", color="red")
plt.title("COVID hospital admissions in the UK spanning a three week period starting 22 December 2021")
plt.xlabel("days")
plt.ylabel('hospitalCases')
plt.legend()
plt.show()
```

```
c1: 7754.297564390236 | c2: 0.051565659399377045
SSE: 29200845.657093737 | R2: 0.9234924160353204
```



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 $\left(c_1 e^{c_2(x+7)}\right)/\left(c_1 e^{c_2 x}\right)=e^{7c_2}$.

```
# compute and print weekly growth rate (in %)
weekly_growth_rate = np.exp(7 * best_fit_params[1])
perentage_weekly_growth_rate = (weekly_growth_rate - 1) * 100
print("weekly growth rate in hospital admissions (in %)
=",perentage_weekly_growth_rate, "%")
```

```
weekly growth rate in hospital admissions (in %) = 43.470552026769305
%
```