02 Exercise Notebook 2

March 23, 2023

0.1 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

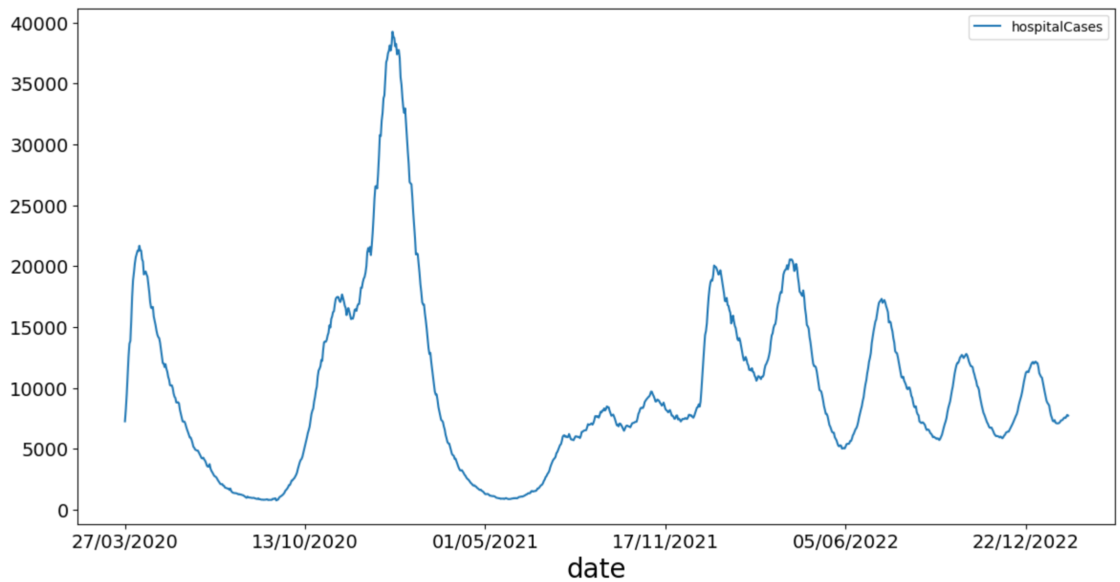
[ ]:
```
df_hosp.sample(3)
```

[ ]:

	date	hospitalCases
1035	26/01/2023	7098
318	08/02/2021	26747
831	06/07/2022	14651

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)
```

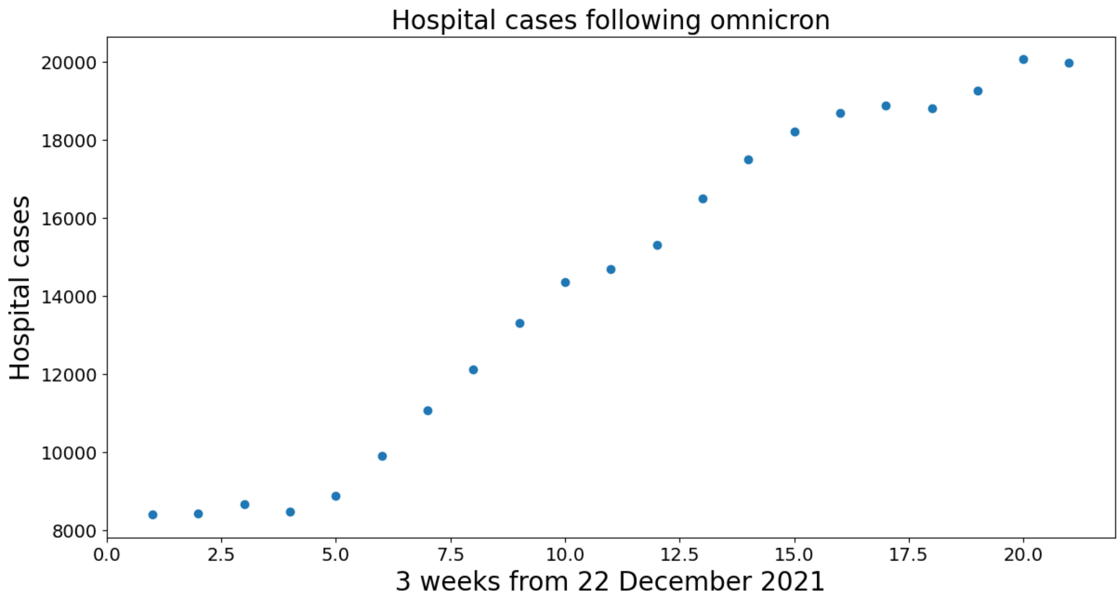
[ ]:
```
           date  hospitalCases
654  10/01/2022          20065
655  11/01/2022          19967
```

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
```

0.2 2a) Plot the data yvals vs xvals in a scatterplot

[ ]:
```
# Your code for scatterplot here
plt.scatter(xvals, yvals)
plt.title('Hospital cases following omnicron')
plt.xlabel('3 weeks from 22 December 2021')
plt.ylabel('Hospital cases')
plt.show()
```



0.3 2b) Fit an exponential model to the data

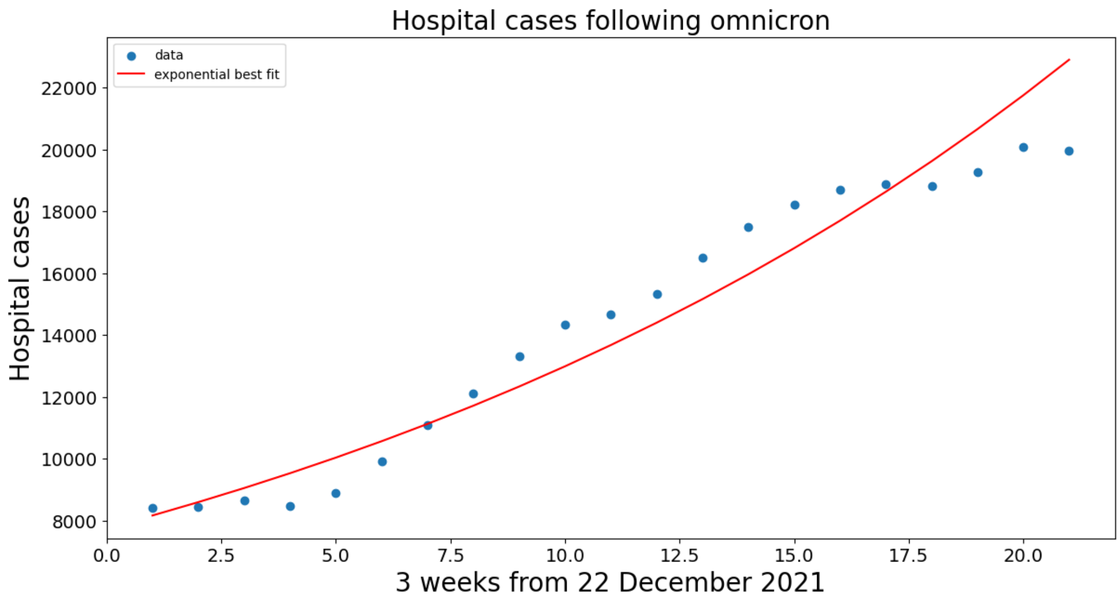
From our knowledge of how the virus spreads, we know that the number of infections, hospital admissions etc. should (roughly) follow an exponential curve. We would therefore like to fit a model of the form $y = c_1 e^{c_2 x}$, where y is the number of admissions on day x.

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(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
all_ones = np.ones(np.shape(xvals))
X = np.column_stack((all_ones, xvals))
XT= X.T
beta_quad = np.linalg.inv(XT.dot(X)).dot(XT).dot(np.log(yvals)) # Calculating␣
↪the least-squares coefficients for log(y) = log(c1) + c2*x

c1,c2 = np.exp(beta_quad[0]), beta_quad[1] # conversion back to non linear␣
↪values
print(f'c1 = {c1:.2f}, c2 = {c2:.2f}')
exp_fit = c1*np.exp(c2*xvals)

plt.scatter(xvals, yvals, label='data')
plt.plot(xvals, exp_fit, 'r', label='exponential best fit')

plt.title('Hospital cases following omnicron')
plt.xlabel('3 weeks from 22 December 2021')
plt.ylabel('Hospital cases')
plt.legend()
plt.show()

SSE = np.sum((yvals - exp_fit)**2)
print(f'SSE = {SSE:.2f}')
```

```
c1 = 7754.30, c2 = 0.05
```




```
SSE = 29200845.66
```

0.4 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 %)
numerical_growth_rate = np.exp(7*c2) # this is the multiplicative factor for
 ↪the exponential growth
percentage_growth_rate = (numerical_growth_rate - 1)*100

print(f'Weekly growth rate = {percentage_growth_rate:.2f}%')
```

```
Weekly growth rate = 43.47%
```