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.

In [1]:
```
# If you are running this on Google Colab, uncomment and run the following lines; o
# from google.colab import drive
# drive.mount('/content/drive')
```

In [1]:
```
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.

In [3]:
```
df_hosp = pd.read_csv('hospital_cases_2023-02-16.csv')  # Create a data frame by lo
# If running on Google Colab change path to '/content/drive/MyDrive/IB-Data-Science

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

Out[3]:

	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()`.

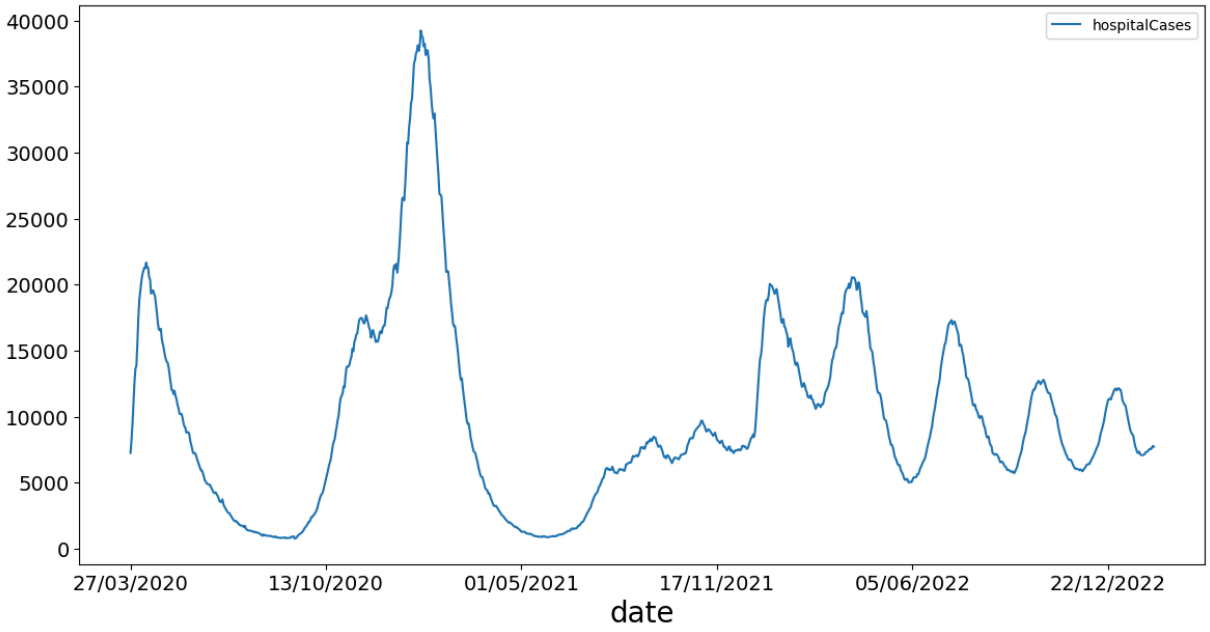
In [4]:
```
df_hosp.tail(3)
```

Out[4]:

	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.

In [5]:
```
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`

In [6]:
```
df_part = df_hosp[635:656]
df_part.head(2)
```

Out[6]:

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

In [7]:
```
df_part.tail(2)
```

Out[7]:

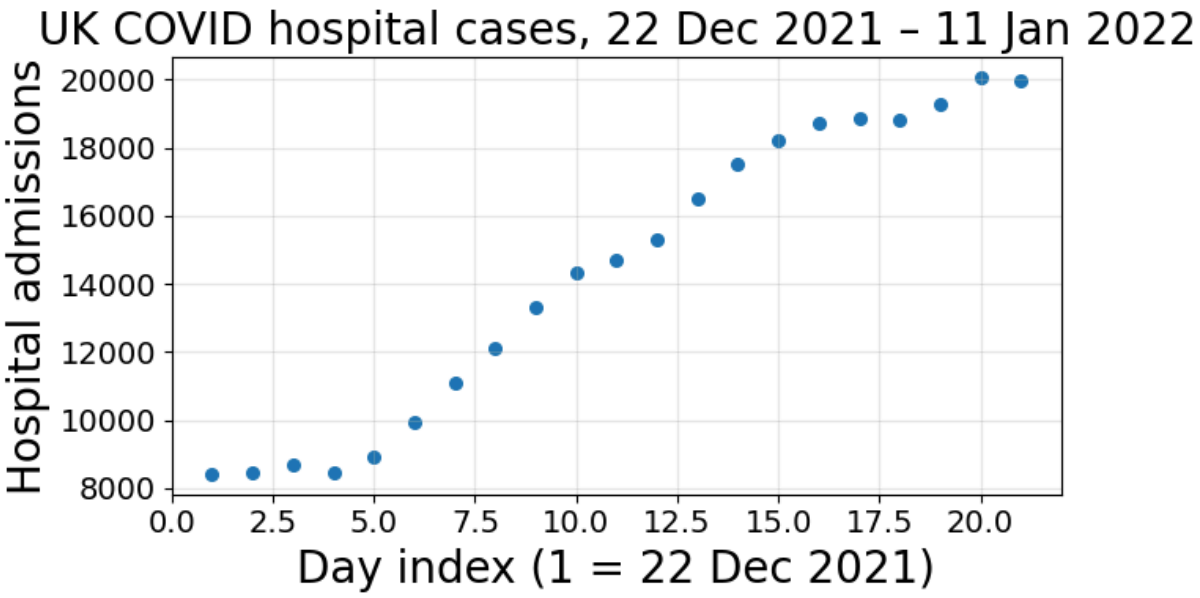
	date	hospitalCases
654	10/01/2022	20065
655	11/01/2022	19967

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

In [8]:
```
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

In [9]:
```
plt.figure(figsize=(7, 4))
plt.scatter(xvals, yvals, s=30)
plt.xlabel("Day index (1 = 22 Dec 2021)")
plt.ylabel("Hospital admissions")
plt.title("UK COVID hospital cases, 22 Dec 2021 – 11 Jan 2022")
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```



2b) Fit an exponential model to the data

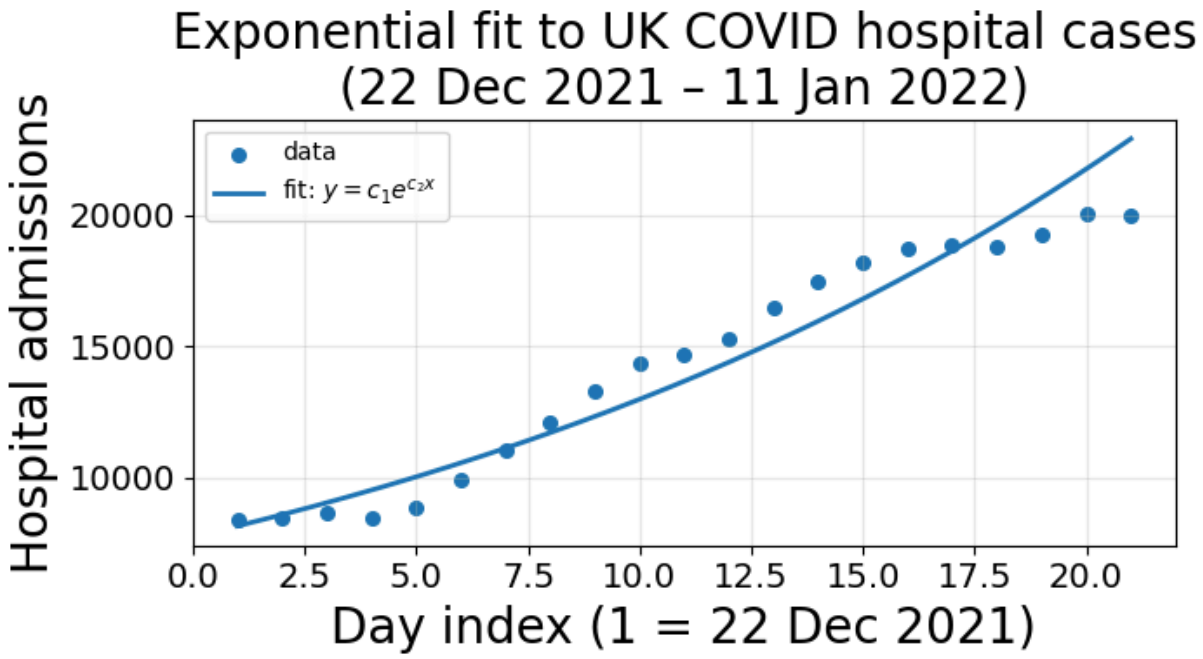
From our knowledge of how the virus spreads, we know that the number of infections, hospital admissions etc. should (roughly) follow an exponential curve. We would therefore like to fit a model of the form $y = c_1 e^{c_2 x}$, where y is the number of admissions on day x.

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 $\log(\text{yvals})$ 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

In [10]:
```
import numpy as np
import matplotlib.pyplot as plt

# ---------- 1) build design matrix and solve for coefficients -----------------
log_y = np.log(yvals)                    # natural-log transform (data are all > 0)
X = np.column_stack([np.ones_like(xvals), xvals])   # [1, x] columns

beta, *_ = np.linalg.lstsq(X, log_y, rcond=None)    # beta = [log c1, c2]
c1 = np.exp(beta[0])
c2 = beta[1]

print(f"Estimated parameters:\n  c1 = {c1:.4g}\n  c2 = {c2:.4g}")

# ---------- 2) plot data and fitted curve -------------------------------------
x_grid = np.linspace(xvals.min(), xvals.max(), 400)
y_fit = c1 * np.exp(c2 * x_grid)

plt.figure(figsize=(7, 4))
plt.scatter(xvals, yvals, s=30, label="data")
plt.plot(x_grid, y_fit, linewidth=2, label=r"fit: $y=c_1 e^{c_2 x}$")
plt.xlabel("Day index (1 = 22 Dec 2021)")
plt.ylabel("Hospital admissions")
plt.title("Exponential fit to UK COVID hospital cases\n(22 Dec 2021 – 11 Jan 2022)"
plt.legend()
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```

```
Estimated parameters:
  c1 = 7754
  c2 = 0.05157
```



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

In [11]:
```
# Weekly growth factor and percentage increase
weekly_factor = np.exp(7 * c2)          # admissions multiply by this each 7-day st
weekly_growth_pct = (weekly_factor - 1) * 100

print(f"Weekly growth rate (model): {weekly_growth_pct:.1f}%")
```

```
Weekly growth rate (model): 43.5%
```