

```
import numpy as np
import pandas as pd

import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
```

EXERCISE 1

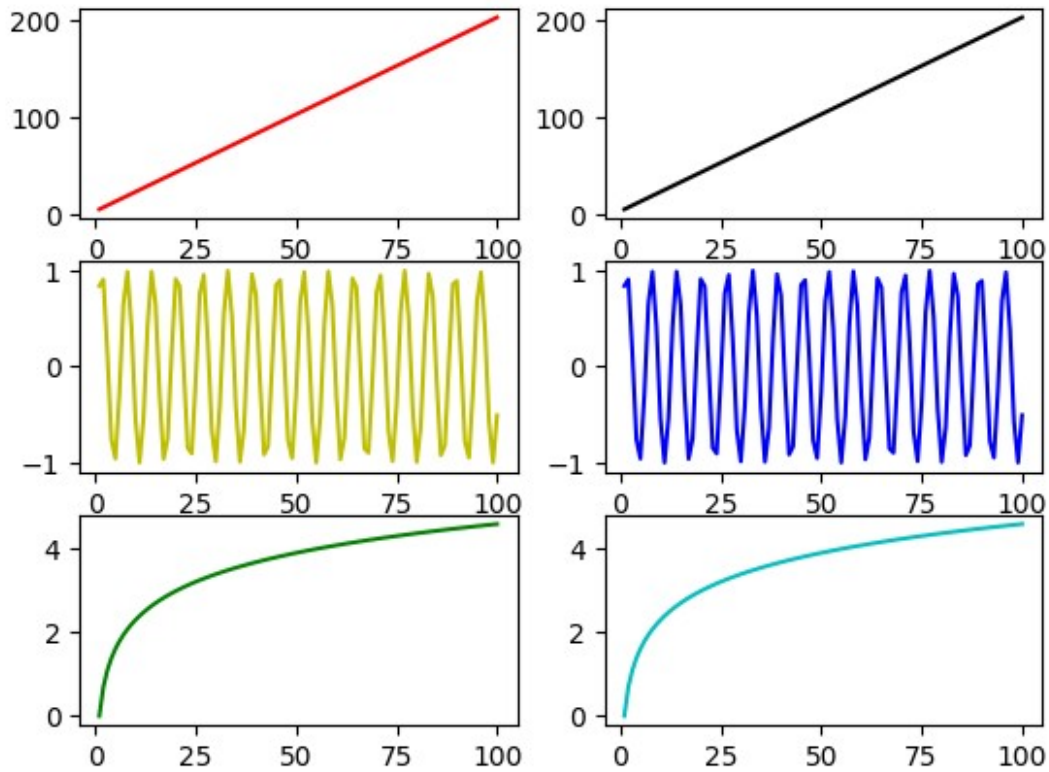
Create a figure with 6 plots arranged in 3 rows and 2 columns. Provide 2 solutions: one with `plt.subplot` and the other with `plt.subplots`.

- The first 2 plots show the linear function x in two colours: black and red.
- The next 2 plots show the $\sin(x)$ function in two colours: yellow and dark blue.
- The last 2 plots show the $\ln(x)$ function in two colours: green and blue.
- x should take values from 1 to 100.

```
fig, ax = plt.subplots(3,2)
vals = np.arange(1, 101)
fx = vals * 2 + 3
sinfx = np.sin(vals)
lnfx = np.log(vals)

ax[0, 0].plot(vals, fx, 'r')
ax[0, 1].plot(vals, fx, 'k')
ax[1, 0].plot(vals, sinfx, 'y')
ax[1, 1].plot(vals, sinfx, 'b')
ax[2, 0].plot(vals, lnfx, 'g')
ax[2, 1].plot(vals, lnfx, 'c')

[<matplotlib.lines.Line2D at 0x21e2b595af0>]
```



```
plt.subplot(3, 2, 1)
plt.plot(vals, fx, 'r')
```

```
plt.subplot(3, 2, 2)
plt.plot(vals, fx, 'k')
```

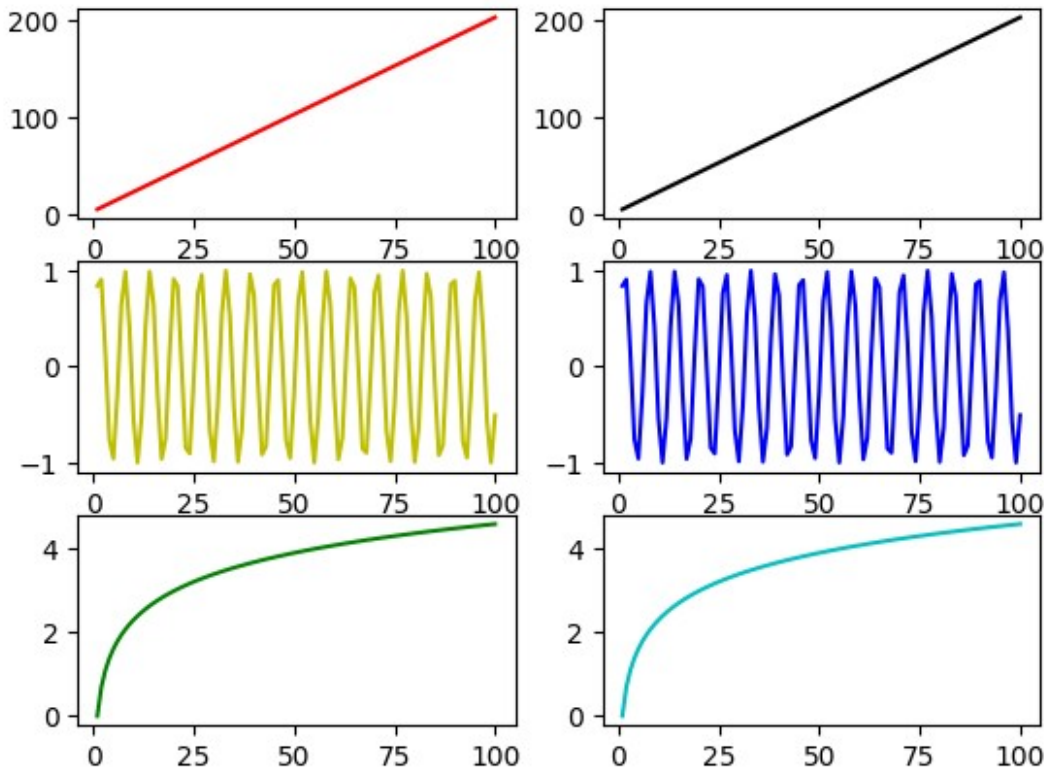
```
plt.subplot(3, 2, 3)
plt.plot(vals, sinfx, 'y')
```

```
plt.subplot(3, 2, 4)
plt.plot(vals, sinfx, 'b')
```

```
plt.subplot(3, 2, 5)
plt.plot(vals, lnfx, 'g')
```

```
plt.subplot(3, 2, 6)
plt.plot(vals, lnfx, 'c')
```

```
[<matplotlib.lines.Line2D at 0x21e2d7c5ee0>]
```



EXERCISE 2

Create a line chart to plot the icescream sales in 2 different regions according to temperature. The data for the first region (say Cluj) is the same as above. The data for the second region (say Sibiu) will contain random values in the interval [200, 700].

- The plot shows 2 lines in colors magenta and blue corresponding to the sales in the 2 regions.
- The plot has a title: "Icescream sales in Cluj vs Sibiu" based on user-defined font.
- The plot has x and y labels.
- The plot has a grid that uses a dotted linestyle, brown color and a line width of 0.75.
- The plot has a legend for the two lines displayed: 'Cluj Sales' and 'Sibiu Sales'.
- The plot has a custom background color (choose one from here: <https://matplotlib.org/stable/tutorials/colors/colors.html>)
- Make different changes to the plot to improve its appearance.

```
temperatures = np.array([15, 20, 25, 30, 35, 40])
cluj_sales = np.random.randint(200, 701, size=temperatures.shape)
sibiu_sales = np.random.randint(200, 701, size=temperatures.shape)

plt.figure(figsize=(10, 6))
plt.title("Icescream Sales in Cluj vs Sibiu", fontsize=16,
fontWeight='bold')
plt.xlabel("Temperature")
plt.ylabel("Sales")
```

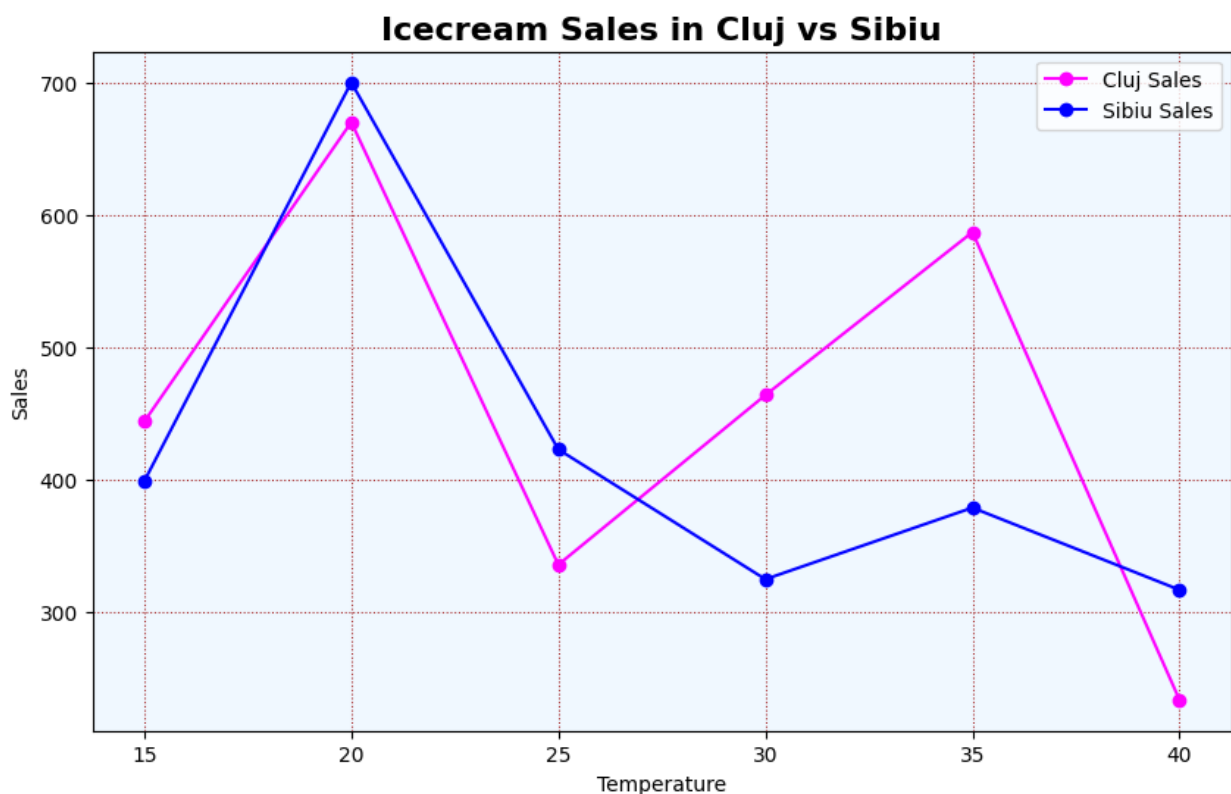
```
plt.grid(color='brown', linestyle=':', linewidth=0.75)

plt.gca().set_facecolor('#f0f8ff')

plt.plot(temperatures, cluj_sales, color='magenta', label='Cluj Sales', marker='o')
plt.plot(temperatures, sibiu_sales, color='blue', label='Sibiu Sales', marker='o')

plt.legend(['Cluj Sales', 'Sibiu Sales'])

<matplotlib.legend.Legend at 0x21e2f29b3e0>
```



EXERCISE 3

Create a line chart to plot the icescream sales from the Sales dataset. Add a second axes with a line chart showing the sales for temepratures above 20 (after position 5 in the sorted values).

- Using TeX markup, add an equation to the plot that uses fractions, radicals, sin/cos, mathematical operators, greek letters.
- Add a text to the plot above the equation with the message "Equation:".
- The main line plot should have: dashed line style, line width 3, line color black, background color aquamarine, plot title with 24pt font and purple color, x and y labels of 14pt and red color.

- Add a second axes as mini-plot inside the main line plot showing the sales for temperatures above 20. The mini-plot should have a title, grid, background color, line style solid and line color black.

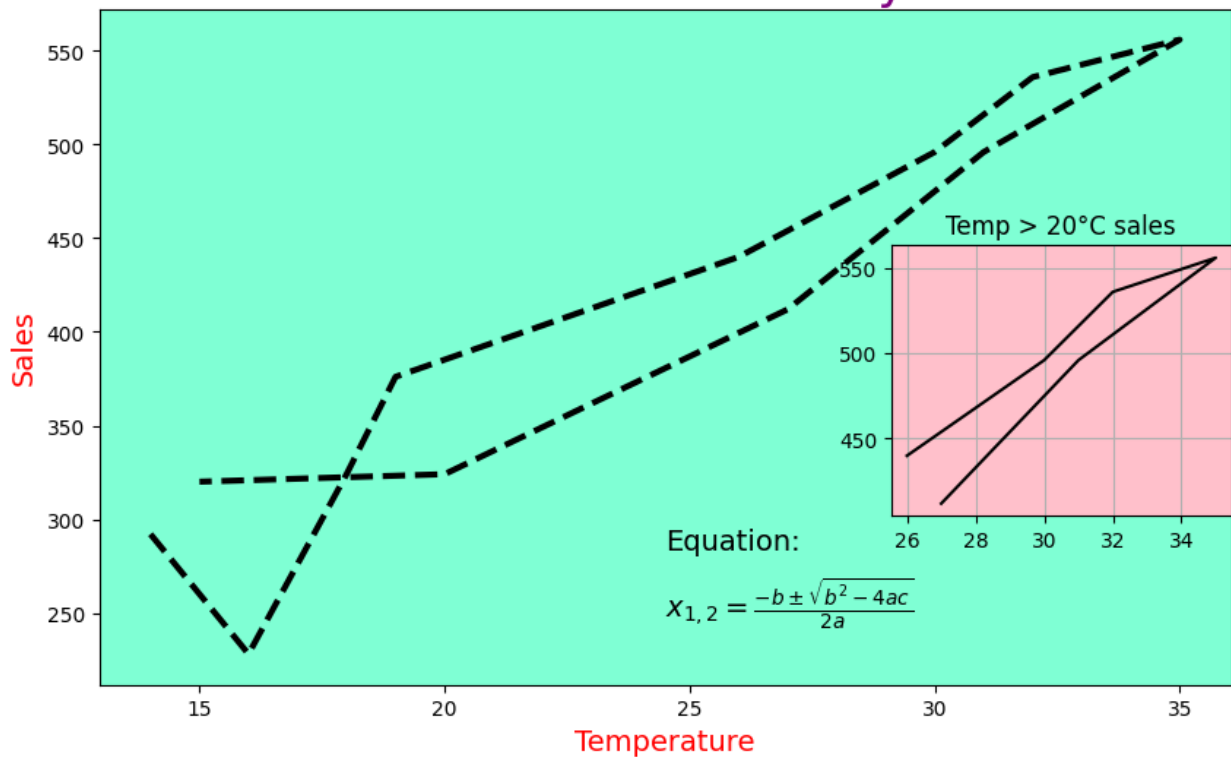
```
df_sales = pd.read_csv('Sales.csv')
temperature = df_sales.iloc[:, 0]
sales_data = df_sales.iloc[:, 1]

plt.figure(figsize=(10, 6))
ax = plt.gca()
ax.set_facecolor('aquamarine')
plt.title('Icescream Sales Analysis', fontsize=24, color='purple')
plt.xlabel('Temperature', fontsize=14, color='red')
plt.ylabel('Sales', fontsize=14, color='red')
plt.plot(temperature, sales_data, color='black', linestyle='--',
linewidth=3, label='Sales')

plt.text(0.5, 0.2, 'Equation:', fontsize=14, transform=ax.transAxes)
plt.text(0.5, 0.1, r'$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$',
fontsize=14, transform=ax.transAxes)

above_20 = np.where(temperature > 20)[0]
inset_ax = ax.inset_axes([0.7, 0.25, 0.3, 0.4]) # [x, y, width,
height] in figure coordinates
inset_ax.plot(temperature[above_20], sales_data[above_20],
color='black', linestyle='-', label='Above 20°C Sales')
inset_ax.set_title('Temp > 20°C sales')
inset_ax.grid(True)
inset_ax.set_facecolor('pink')
```

Icescream Sales Analysis



EXERCISE 4

Plot a line chart with the number of immigrants from Romania to Canada.

- Use the 'seaborn' mpl style.
- Make sure that the line style is dashed and the line color is green.

```
file = pd.ExcelFile('Canada.xlsx')
last_sheet = file.sheet_names[-1]
```

```
df = file.parse(last_sheet)
df
```

	Type	Coverage	OdName	AREA	AreaName	REG	\
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	
1	Immigrants	Foreigners	Albania	908	Europe	925	
2	Immigrants	Foreigners	Algeria	903	Africa	912	
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	
4	Immigrants	Foreigners	Andorra	908	Europe	925	
...	
191	Immigrants	Foreigners	Western Sahara	903	Africa	912	
192	Immigrants	Foreigners	Yemen	935	Asia	922	
193	Immigrants	Foreigners	Zambia	903	Africa	910	
194	Immigrants	Foreigners	Zimbabwe	903	Africa	910	
195	Immigrants	Foreigners	Unknown	999	World	999	

	RegName	DEV	DevName	1980	...	2004	2005
2006 0	Southern Asia	902	Developing regions	16	...	2978	3436
3009 1	Southern Europe	901	Developed regions	1	...	1450	1223
856 2	Northern Africa	902	Developing regions	80	...	3616	3626
4807 3	Polynesia	902	Developing regions	0	...	0	0
1 4	Southern Europe	901	Developed regions	0	...	0	0
1
... 191	Northern Africa	902	Developing regions	0	...	0	0
1 192	Western Asia	902	Developing regions	1	...	124	161
140 193	Eastern Africa	902	Developing regions	11	...	56	91
77 194	Eastern Africa	902	Developing regions	72	...	1450	615
454 195	World	999	World	44000	...	3739	4785
4583							

	2007	2008	2009	2010	2011	2012	2013
0	2652	2111	1746	1758	2203	2635	2004
1	702	560	716	561	539	620	603
2	3623	4005	5393	4752	4325	3774	4331
3	0	0	0	0	0	0	0
4	1	0	0	0	0	1	1
...
191	0	0	0	0	0	0	0
192	122	133	128	211	160	174	217
193	71	64	60	102	69	46	59
194	663	611	508	494	434	437	407
195	4348	4197	3402	3731	2554	1681	1484

```
[196 rows x 43 columns]
```