

## Plotting with Python subplots

1. Using `wget` on the shell, download Seattle weather data: `tinyurl.com/seattle-weather-csv`.

```
wget -O data/weather.csv tinyurl.com/seattle-weather2-csv
```

2. Check with `ls -l` on the shell that the file was created:

```
ls -l data/weather.csv
```

3. Using `head` on the shell, look at the first couple of lines of the file to ascertain that the data have a header:

```
head --lines=2 data/weather.csv
```

4. Import the data into a `pandas` data frame using `pd.read_csv`, save it to `df` and print the data:

```
import pandas as pd
df = pd.read_csv("../data/weather.csv")
print(df)
```

	DATE	PRCP	TMAX	TMIN	RAIN	
0	1948-01-01	0.47	51	42	True	
1	1948-01-02	0.59	45	36	True	
2	1948-01-03	0.42	45	35	True	
3	1948-01-04	0.31	45	34	True	
4	1948-01-05	0.17	45	32	True	
...	...	...	...	...	...	...
25546	2017-12-10	0.00	49	34	False	
25547	2017-12-11	0.00	49	29	False	
25548	2017-12-12	0.00	46	32	False	
25549	2017-12-13	0.00	48	34	False	
25550	2017-12-14	0.00	50	36	False	

```
[25551 rows x 5 columns]
```

Note: These commands may fail if your `*Python*` session is not set to the current working directory, just like in R (`getwd()` to check, `setwd` to set). The equivalent commands in Python are:

```
import os # operating systems interface methods
os.getcwd() # check current working directory
os.chdir("[PATH]/dir") # change current working directory to dir using the PATH
```

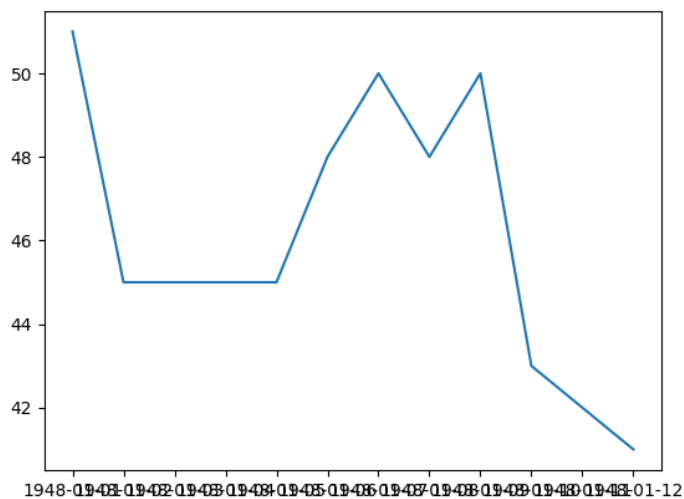
5. Save the first 12 rows of the data frame to df2

```
df2 = (df.head(n=12))
print(df2)
```

	DATE	PRCP	TMAX	TMIN	RAIN
0	1948-01-01	0.47	51	42	True
1	1948-01-02	0.59	45	36	True
2	1948-01-03	0.42	45	35	True
3	1948-01-04	0.31	45	34	True
4	1948-01-05	0.17	45	32	True
5	1948-01-06	0.44	48	39	True
6	1948-01-07	0.41	50	40	True
7	1948-01-08	0.04	48	35	True
8	1948-01-09	0.12	50	31	True
9	1948-01-10	0.74	43	34	True
10	1948-01-11	0.01	42	32	True
11	1948-01-12	0.00	41	26	False

6. Plot maximum temperature against the date:

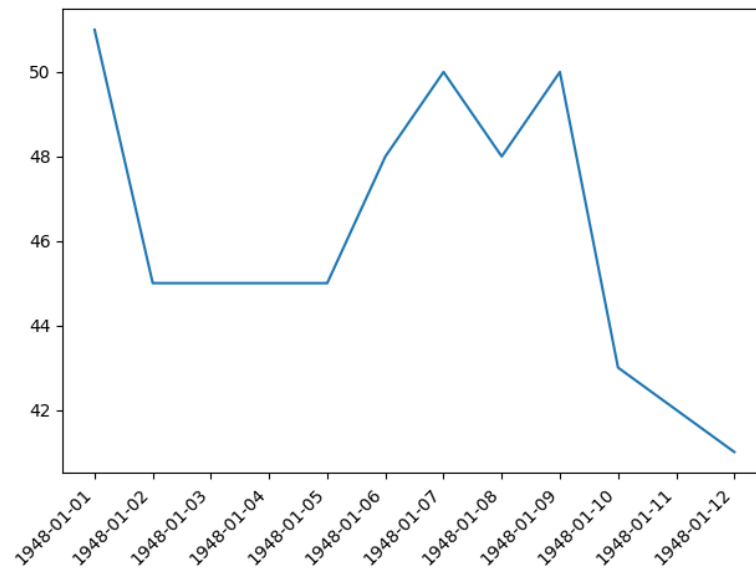
```
import matplotlib.pyplot as plt
fig,ax=plt.subplots()
ax.plot(df2["DATE"],df2["TMAX"])
plt.savefig("seattle.png")
```



7. This last plot is very hard to read because all the x-axis labels are printed on top of one another! Turn the x-axis labels, tighten the layout:

```
fig,ax=plt.subplots()

ax.plot(df2["DATE"],df2["TMAX"])
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
```

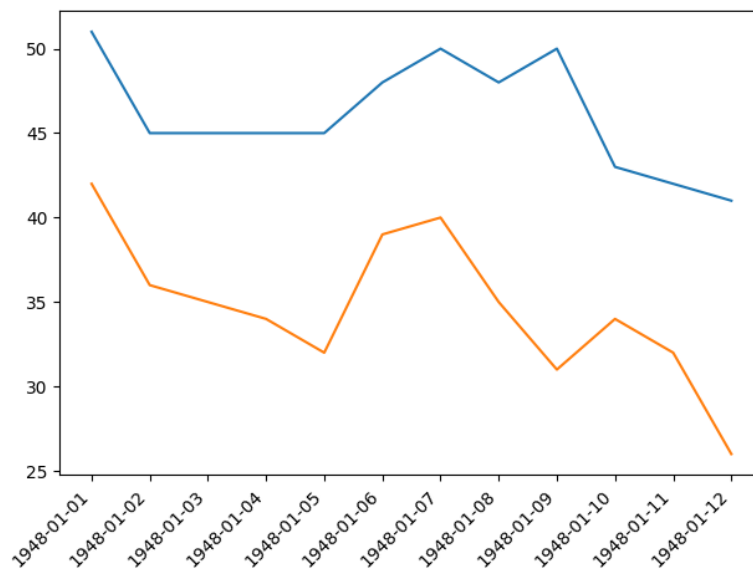


8. Add a second plot with the minimum temperatures:

```
fig,ax=plt.subplots()

ax.plot(df2["DATE"],df2["TMAX"])
ax.plot(df2["DATE"],df2["TMIN"])

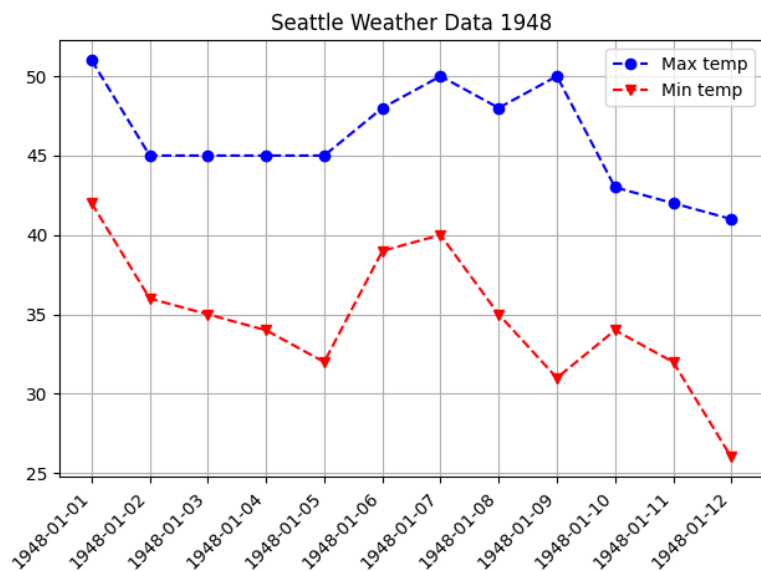
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
```



9. Add a suitable title, x- and y-labels, a legend, different markers and colors, a legend and a grid:

```
fig,ax=plt.subplots()

ax.plot(df2["DATE"],df2["TMAX"],
        label='Max temp',
        color='b',
        marker='o',
        linestyle='--')
ax.plot(df2["DATE"],df2["TMIN"],
        label='Min temp',
        color='r',
        marker='v',
        linestyle='--')
plt.title("Seattle Weather Data 1948")
plt.legend()
plt.grid()
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
```

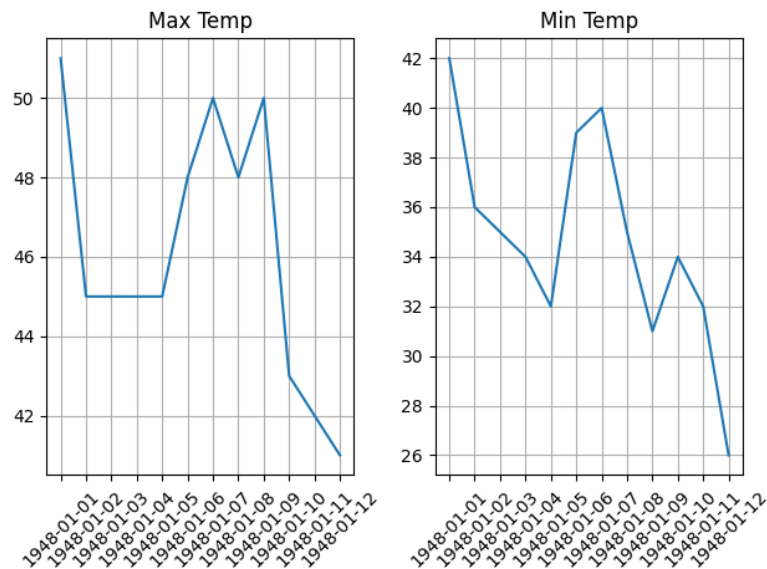


- Put the two plots next to one another instead of plotting them on top of one another:

```
fig,ax=plt.subplots(1,2)

ax[0].plot(df2["DATE"],df2["TMAX"])
ax[0].set_title("Max Temp")
ax[0].grid(True)
ax[0].tick_params(axis='x',rotation=45)

ax[1].plot(df2["DATE"],df2["TMIN"])
ax[1].set_title("Min Temp")
ax[1].grid(True)
ax[1].tick_params(axis='x',rotation=45)
plt.tight_layout()
```



11. You notice that the y-axes of these plots have different ranges. In R, you set `ylim` to run from the smallest (min) to the largest (max) value. In Python, you do the same thing:

```
# create a graphical frame
fig,ax=plt.subplots(1,2)

# set y-axis limits to the same range
min_temp = min(df2["TMIN"].min(),df2["TMAX"].max())
max_temp = max(df2["TMAX"].min(),df2["TMAX"].max())

# first plot
ax[0].plot(df2["DATE"],df2["TMAX"])
ax[0].set_title("Max Temp")
ax[0].grid(True)
ax[0].tick_params(axis='x',rotation=45)
ax[0].set_ylim([min_temp,max_temp])

# second plot
ax[1].plot(df2["DATE"],df2["TMIN"])
ax[1].set_title("Min Temp")
ax[1].grid(True)
```

```
ax[1].tick_params(axis='x',rotation=45)
ax[1].set_ylim([min_temp,max_temp])

plt.tight_layout()
```

