

Lesson 5

Plotting

Plotting in python is based on MatLab plotting, which is useful as a reference. You will need to import a module used both for plotting and making images.

```
>> import matplotlib.pyplot as plt          # (you have already imported numpy)
```

```
>> plt.plot(x1, y1, 'linespec1', x2, y2, 'linespec2' ...) # Multiple line plot
```

Where the x and y are data vectors, and linespec is up to 3 symbols (listed in quotes) that signify line color, line type and plot symbol. The line specifiers are given in the table below:

specifier	color		specifier	marker
r	Red		o	Circle
g	Green		+	Plus
b	Blue		*	Asterisk
y	Yellow		.	Dot
c	Cyan (blue-green)		x	Cross
m	Magenta (purple)		s	Square
			d	Diamond
specifier	Line style		^	Up triangle
-	Solid		v	Down triangle
--	Dashed		>	Left triangle
:	dotted		<	Right triangle
.-	Dot-dash		h	hexagram

Example:

Create a making_plot.py

```
#Library
```

```
import matplotlib.pyplot as plt
```

```
# Commands
```

```
ax = [1, 2, 3]
```

```
bx = [2, 3, 4]
```

```
ay = [4, 6, 8]
```

```
by = [10, 9, 7]
```

```
fig, axes = plt.subplots(figsize = (10, 8)) # create a figure
```

```
plt.plot(ax, ay, 'r:o', bx, by, 'b-d')
```

```
plt.title('test')
plt.xlabel('x values')
plt.ylabel('y values')
plt.show() # after all specifications, shows plot. Close to continue.
```

will plot an ascending red dotted line marked with circles, and a descending blue solid line marked with diamonds. It is titled 'test' with the x and y axes labelled.

To save the plot to an image:

```
>> plt.savefig(folder+'myimage.png')
```

Python will figure out how to save based on the extension: gif, jpeg, etc.

Reading csv files and some basic statistics

In engineering, the most common file that is used to give results is .csv or Comma-separated values. An easy way to read the text files is by using the package pandas in python. In this section, we are going to compute some statistical parameters such median, mode, mean and standard deviation, also it will be used the histogram function to look at the frequency of the data.

Some definitions:

- mean is the average of a set of numbers
- median is the middle value of a list of numbers
- mode is the most repeated value in a dataset
- the standard deviation represents the variation or dispersion of a set of values.

rain.py

Now, we are going to analyze the yearly precipitation data from NOAA at Central Park from 1869 to 2023 ([link](#)):

Monthly & Annual Precipitation at Central Park

Last Updated: 6/16/2024

YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
1869	2.53	6.87	4.61	1.39	4.15	4.40	3.20	1.76	2.81	6.48	2.03	5.02	45.25
1870	4.41	2.83	3.33	5.11	1.83	2.82	3.76	3.07	2.52	4.97	2.42	2.18	39.25
1871	2.07	2.72	5.54	3.03	4.04	7.05	5.57	5.60	2.34	7.50	3.56	2.24	51.26
1872	1.88	1.29	3.74	2.29	2.68	2.93	7.83	6.29	2.95	3.35	4.08	3.18	42.49
1873	5.34	3.80	2.09	4.16	3.69	1.28	4.61	9.56	3.14	2.73	4.63	2.96	47.99
1874	5.33	2.04	2.12	8.77	2.24	2.78	5.06	2.43	8.24	1.70	2.30	2.82	45.83
1875	3.17	2.62	3.48	3.08	1.33	2.72	4.89	8.97	1.89	2.85	3.78	2.12	40.90
1876	0.94	4.81	8.79	3.06	3.03	2.66	3.65	2.28	5.28	1.42	3.31	2.54	41.77
1877	2.62	1.24	5.56	2.73	0.95	2.80	5.73	2.77	1.33	8.14	5.62	0.68	40.17
1878	4.46	3.75	3.27	1.97	3.19	3.08	4.62	7.97	4.05	2.43	4.73	5.14	48.66
1879	2.63	2.02	3.41	4.33	2.02	3.15	3.58	7.95	2.37	0.43	2.20	4.92	39.01

```
# Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
# Commands

rain = [1,2,3,4,4,5]

rain_mean = np.mean(rain)
```

```

rain_median = np.median(rain)
rain_mode = stats.mode(rain)[0]
rain_std = np.std(rain)

data = pd.read_csv(r'precipitation.csv')

# Extract the data for plotting
years = data['YEAR']
year_precip = data['YEAR_PRECIP']

# Create the plot
plt.figure(figsize=(10, 6))
plt.plot(years, year_precip, 'o-b')
plt.xlabel('Year')
plt.ylabel('Yearly Precipitation (mm)')
plt.title('Yearly Precipitation Over Time')
plt.grid(True)
plt.show()

# Statistics
rain_mean = np.mean(rain)
print(rain_mean)
rain_median = np.median(rain)
print(rain_median)
rain_mode = stats.mode(rain)[0]
print(rain_mode)
rain_std = np.std(rain)
print(rain_std)

fig, axes = plt.subplots(figsize = (10, 8)) # create a figure
plt.hist(year_precip, bins = 8, rwidth = 0.9)
plt.title('Histogram of the annual precipitation at Central Park')
plt.xlabel('Precipitation Intervals')
plt.ylabel('Frequency')
plt.show()

```

Hence, we can analyze from the histogram that the most frequent values of rainfall is located between 39 and 41 millimeters.

Now, we are going to look for the maximum and minimum rainfall measurement and also, we have to locate the years when they occurred:

```

rain_max = np.max(year_precip)
print(rain_max)
rain_min = np.min(year_precip)
print(rain_min)

```

```
year_max = data[data['YEAR_PRECIP'] == rain_max]
year_max = year_max.reset_index()
print('Max. precipitation year',year_max.loc[0,'YEAR'])
year_min = data[data['YEAR_PRECIP'] == rain_min]
year_min = year_min.reset_index()
print('Min. precipitation year',year_min.loc[0,'YEAR'])
```