Matplotlib Library

Nicholas Russell Saerang

What is Matplotlib?

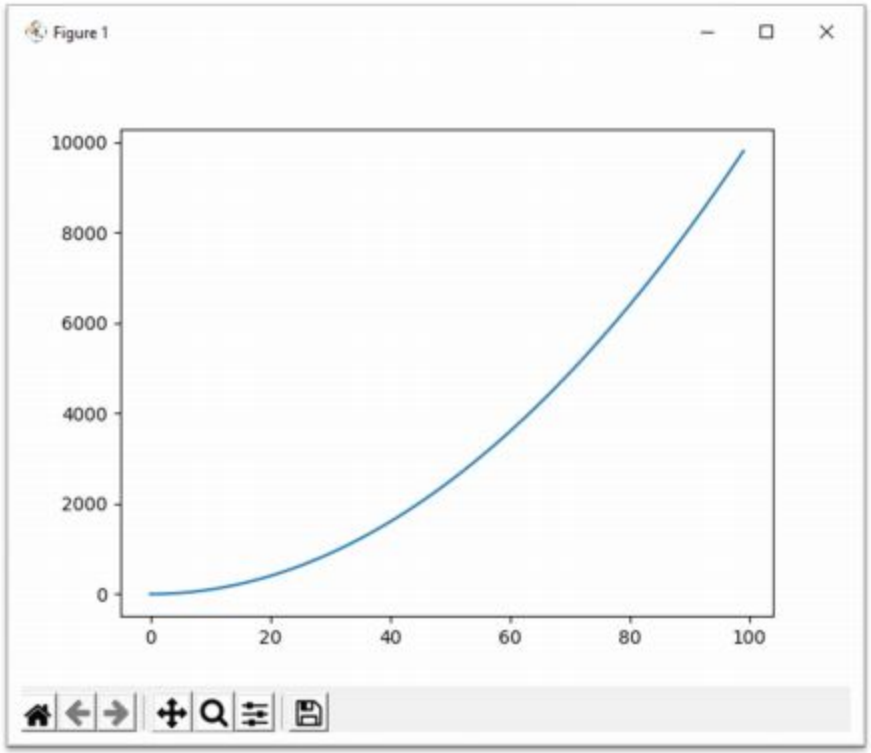
It is a Python library useful for data visualization. We can create bar plots, line plots, histograms, and many more!

Matplotlib is supported with other libraries such as Seaborn which we will also have a try later.

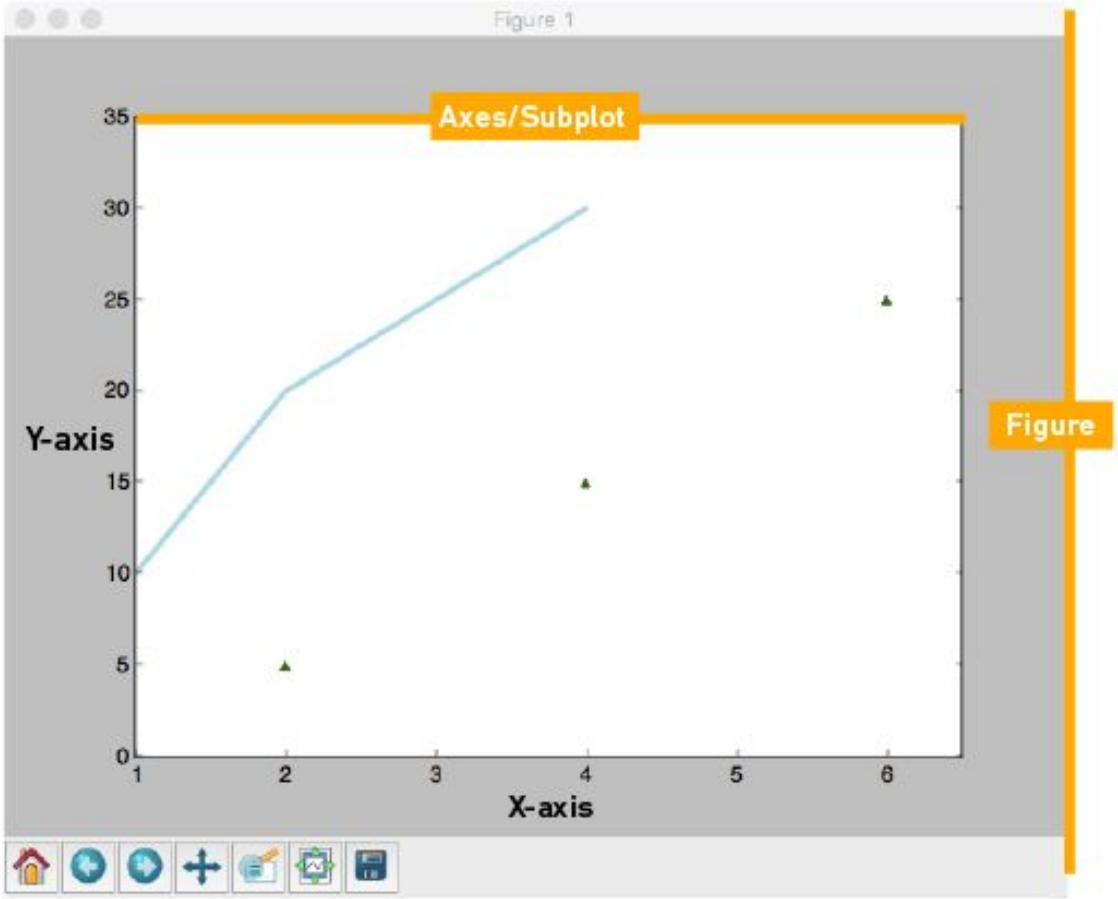
For today's workshop, go to:

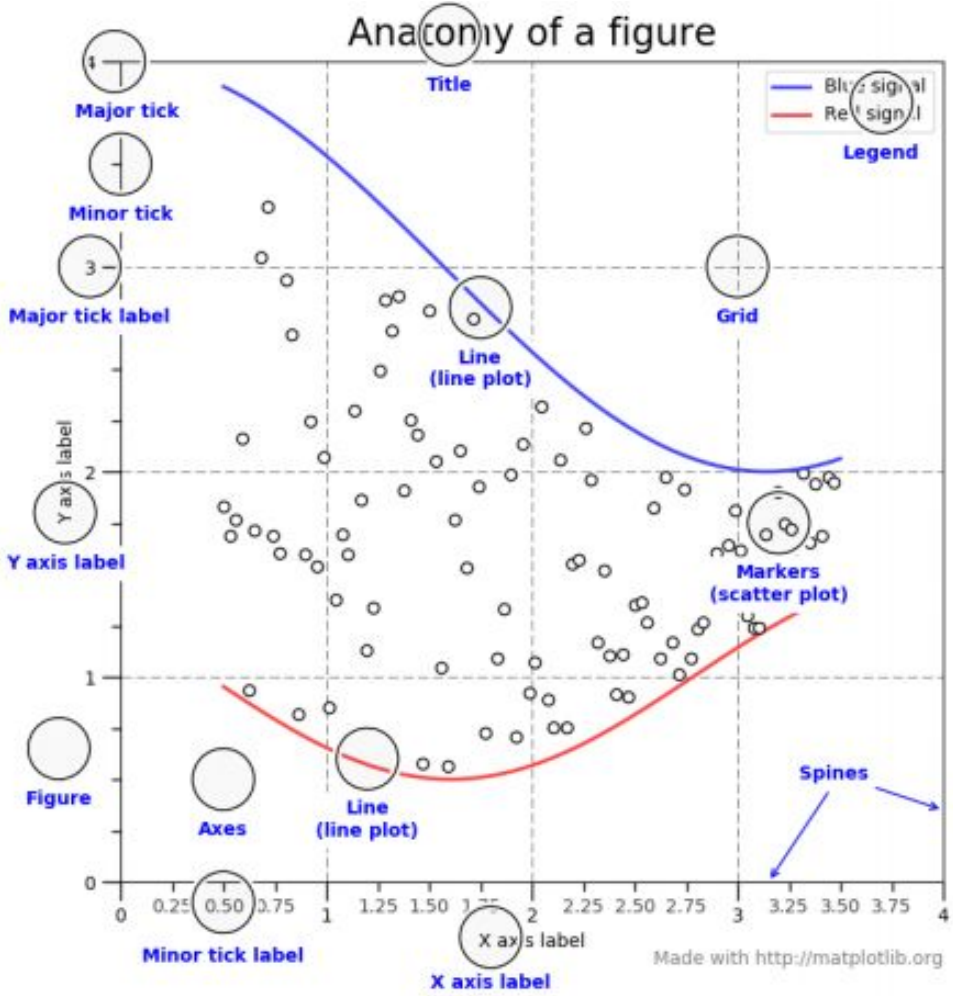
https://colab.research.google.com/drive/16B6vp9XwMwliRXh9o9VIrkXjX2ftJh4?usp=sharing

```
from matplotlib import pyplot as plt

x = tuple(range(100))
y = tuple(map(lambda x:x*x, x))

fig, ax = plt.subplots()

ax.plot(x, y)
fig.show()
```



Plot Anatomy




Anatomy of a figure
Title
Major tick
Minor tick
Major tick label
Legend
Grid
Line
(line plot)
Y axis label
Markers
(scatter plot)
Figure
Axes
Line
(line plot)
Spines
Minor tick label
X axis label
Made with http://matplotlib.org

Workflow

01 Prepare data

02 Create plot

03 Plot

04 Customize plot

05 Save plot

06 Show plot

1. Prepare Data

Datas can be in form of arrays, lists, NumPy arrays, or DataFrame. A trivial example would be like one of these.

```
import matplotlib.pyplot as plt

import seaborn as sns

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

x = np.array([1, 0.2, 0.1])

x = data[‘Year’]
```

```
>>> from matplotlib.cbook import get_sample_data
>>> img = np.load(get_sample_data('axes_grid/bivariate_normal.npy'))
```

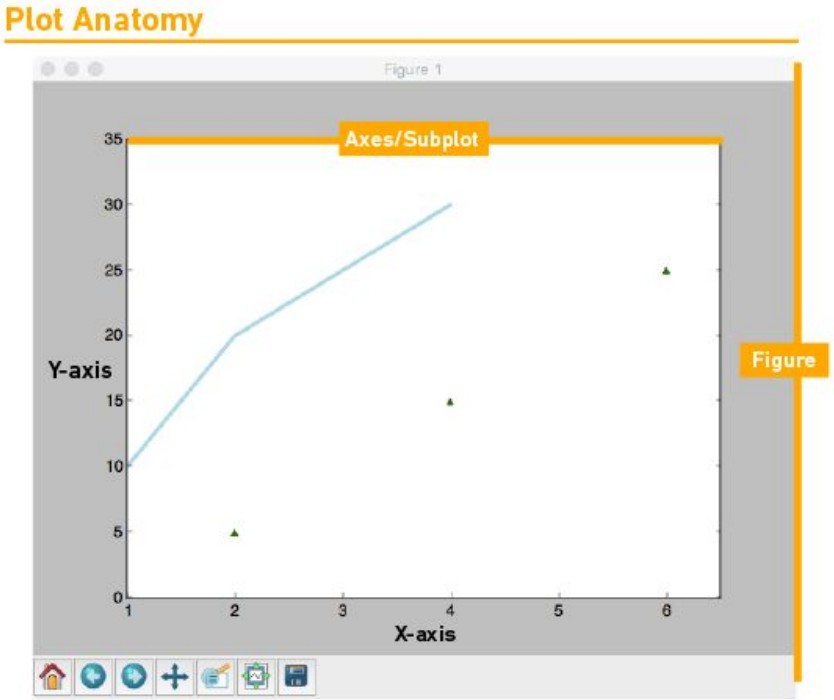
2. Create Plot

There are two quick ways to create a new plot.

```
fig, ax = plt.subplots(figsize = (16,9))
```

```
fig = plt.figure()

ax = fig.add_subplot(111)
```

Recall: Plot Anatomy



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

The whole figure
i.e. the window

The "plot area"
in the figure

A figure can contain multiple plots

```
fig, axes = plt.subplots(2, 2)
```

A 2x2 grid of axes

About add_subplot

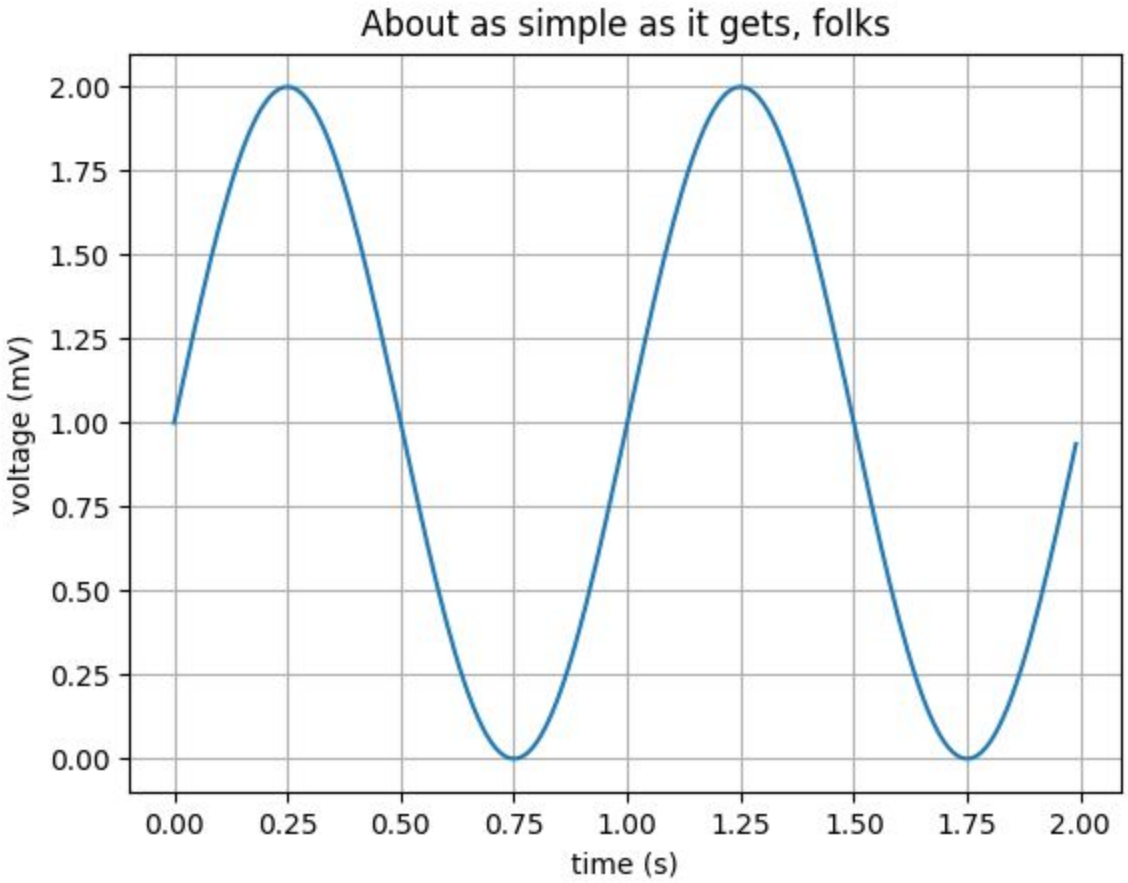
https://matplotlib.org/3.3.4/api/_as_gen/matplotlib.figure.Figure.html#matplotlib.figure.Figure.add_subplot

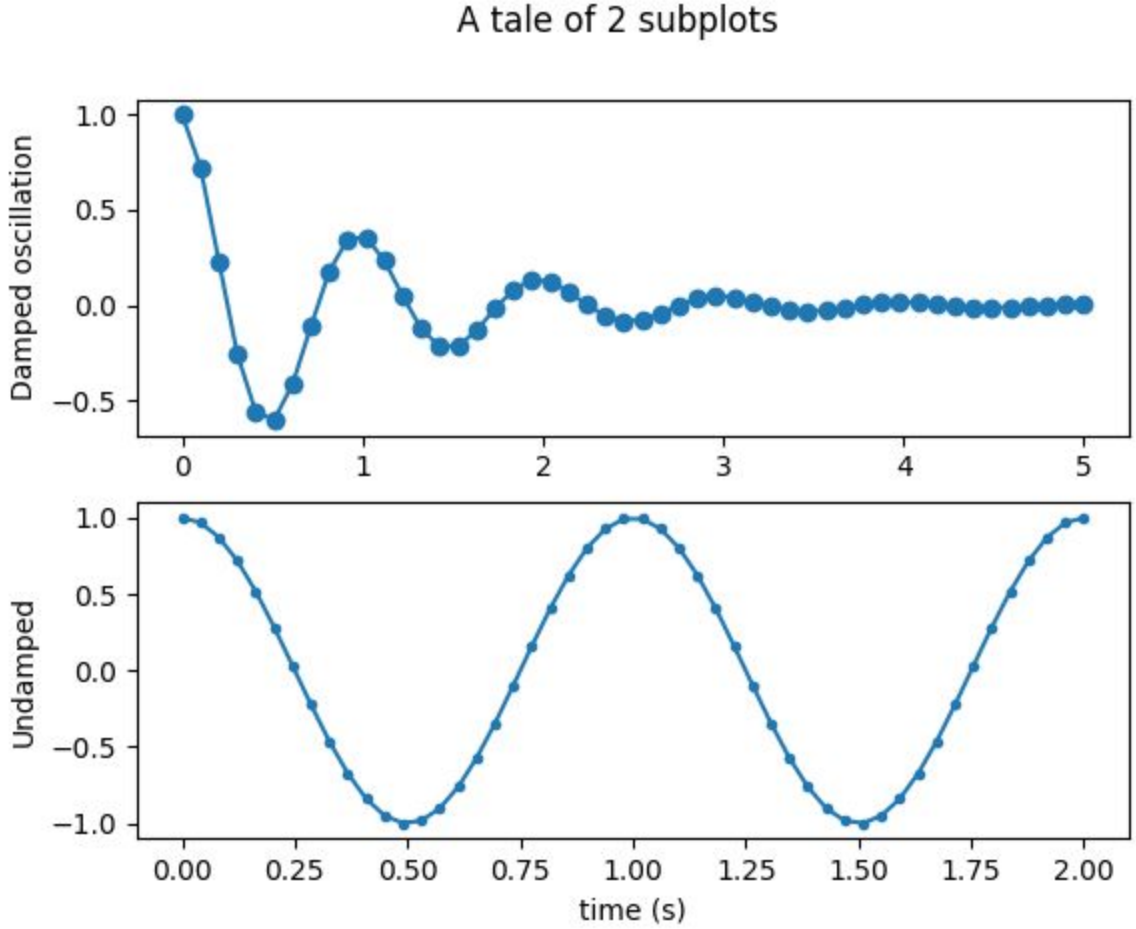
```
>>> fig.add_axes()
>>> ax1 = fig.add_subplot(221) # row-col-num
>>> ax3 = fig.add_subplot(212)
>>> fig3, axes = plt.subplots(nrows=2,ncols=2)
>>> fig4, axes2 = plt.subplots(ncols=3)
```

3. Plot

If we have the X and Y data, we can call the plot function to ax.

```
ax.plot(x,y)

ax.plot(x,y, label = ‘my plot’, alpha = 0.4, ls = ‘--’)

ax.plot(x,y, color = ‘lightblue’, linewidth = 3)

ax.plot(x,y, c = ‘lightblue’, linewidth = 3)

ax.scatter(x,y, marker = ‘o’)
```


About as simple as it gets, folks
voltage (mV)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
time (s)


A tale of 2 subplots
Damped oscillation
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
Undamped
1.0
0.5
0.0
−0.5
−1.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
time (s)

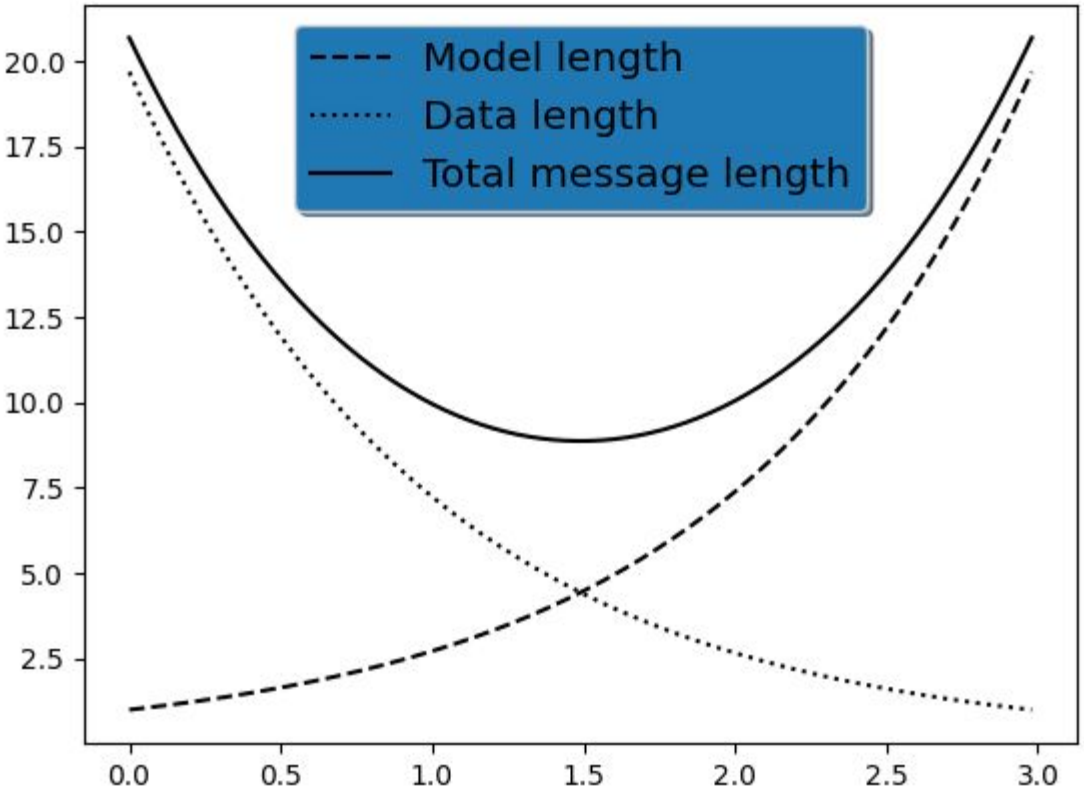
4. Customize Plot

There are many types of plot customizations. The most useful ones are:

- Set axes label
- Set bounds of x and y
- Change type of visualization (bar plot, scatter plot)

Setting Labels and Legend

```
ax.set(title = ‘Plot 1’, xlabel = ‘X-Axis’, ylabel = ‘Y-Axis’)

ax.legend()



ax.set_xlabel(‘X-Axis’)

ax.set_ylabel(‘Y-Axis’)

ax.set_title(‘My Plot’)
```


Model length
Data length
Total message length
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0

Bounding/Limits

Set x limit

```
ax.set_xlim(-1, 10)
```

Set y limit

```
ax.set_ylim(-3, 7)
```

Set both

```
ax.set(xlim = [-1, 10], ylim = [-3, 7])
```

Margin and Scaling

```
>>> ax.margins(x=0.0,y=0.1)
>>> ax.axis('equal')
```

Add padding to a plot

Set the aspect ratio of the plot to 1

```
ax.set_xscale('log')
```

Texts and Annotations

```
>>> ax.text(1,
                -2.1, 'Example Graph',
                 style='italic')

>>> ax.annotate("Sine", xy=(8, 0),
                        xycoords='data',
                        xytext=(10.5, 0),
                        textcoords='data',
                        arrowprops=dict(arrowstyle="->",
                                        connectionstyle="arc3"),)
```

How about math titles?

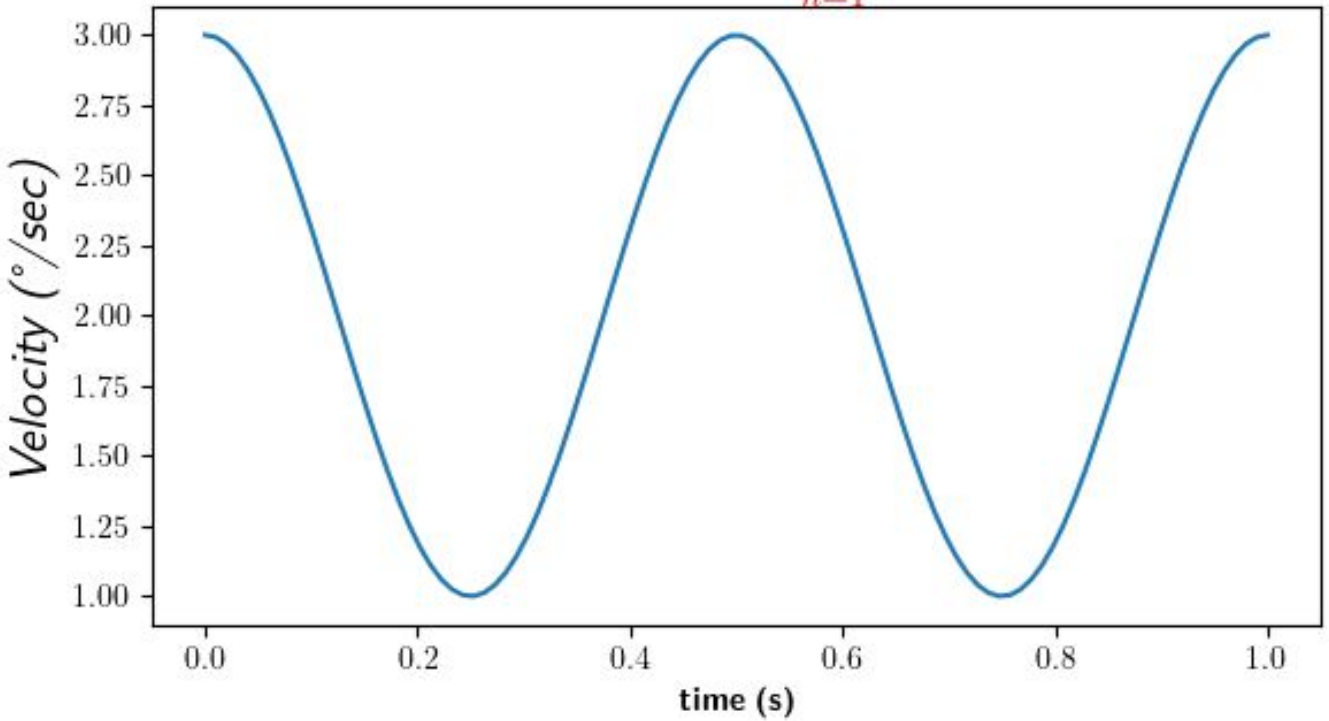
Say you need a sigma symbol, just like this -> Σ

We can put the letter r before the title string, just like below.

```
plt.title(r'$sigma_i = 15$')
```

Notice that the title is enclosed in dollar signs because it's using LaTeX.

TEX is Number $\sum_{n=1}^{\infty} \frac{-e^{i\pi}}{2^n}!$



Different Types of Visualization

```
>>> lines = ax.plot(x,y)
>>> ax.scatter(x,y)
>>> axes[0,0].bar([1,2,3],[3,4,5])
>>> axes[1,0].barh([0.5,1,2.5],[0,1,2])
>>> axes[1,1].axhline(0.45)
>>> axes[0,1].axvline(0.65)
```

Draw points with lines or markers connecting them

Draw unconnected points, scaled or colored

Plot vertical rectangles (constant width)

Plot horiontal rectangles (constant height)

Draw a horizontal line across axes

Draw a vertical line across axes

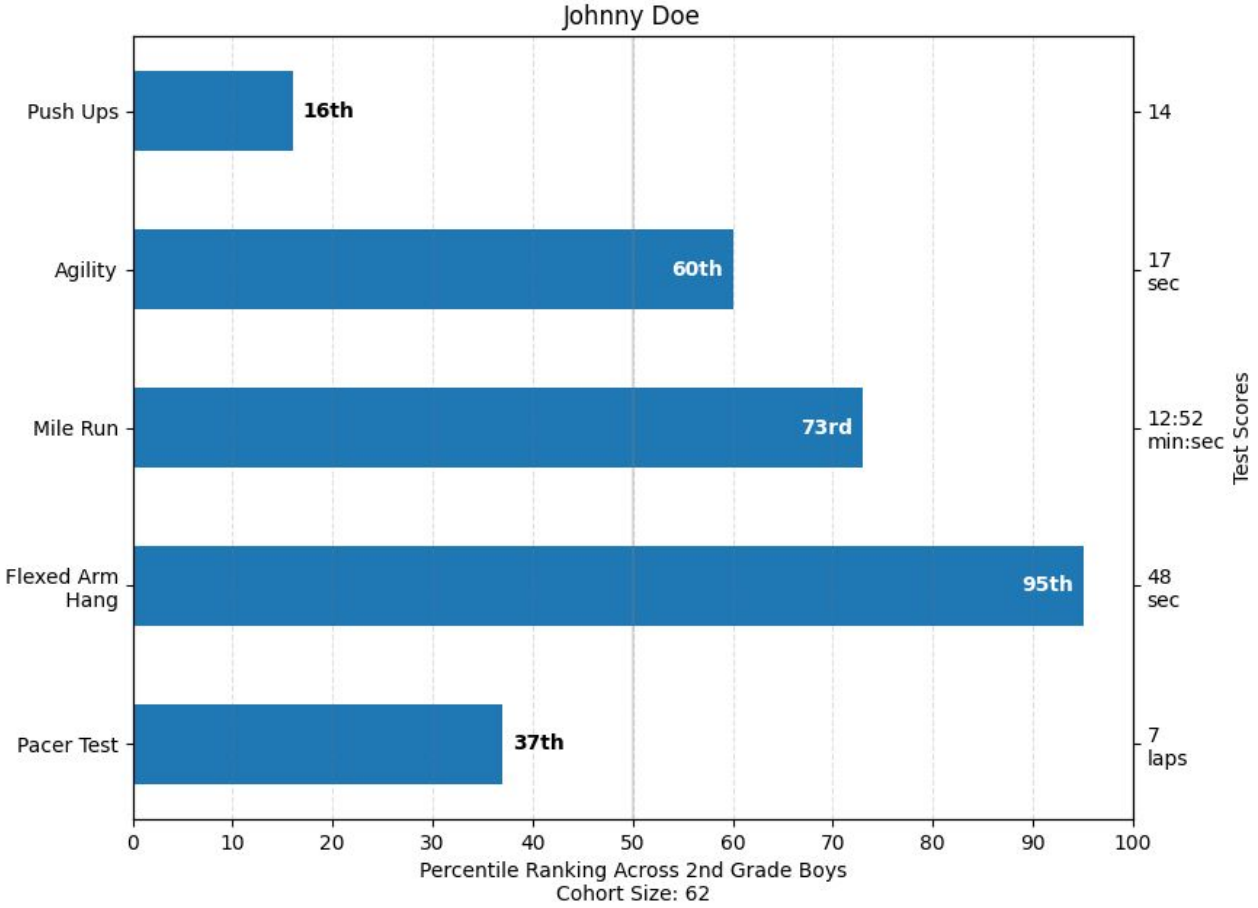
Different Types of Visualization

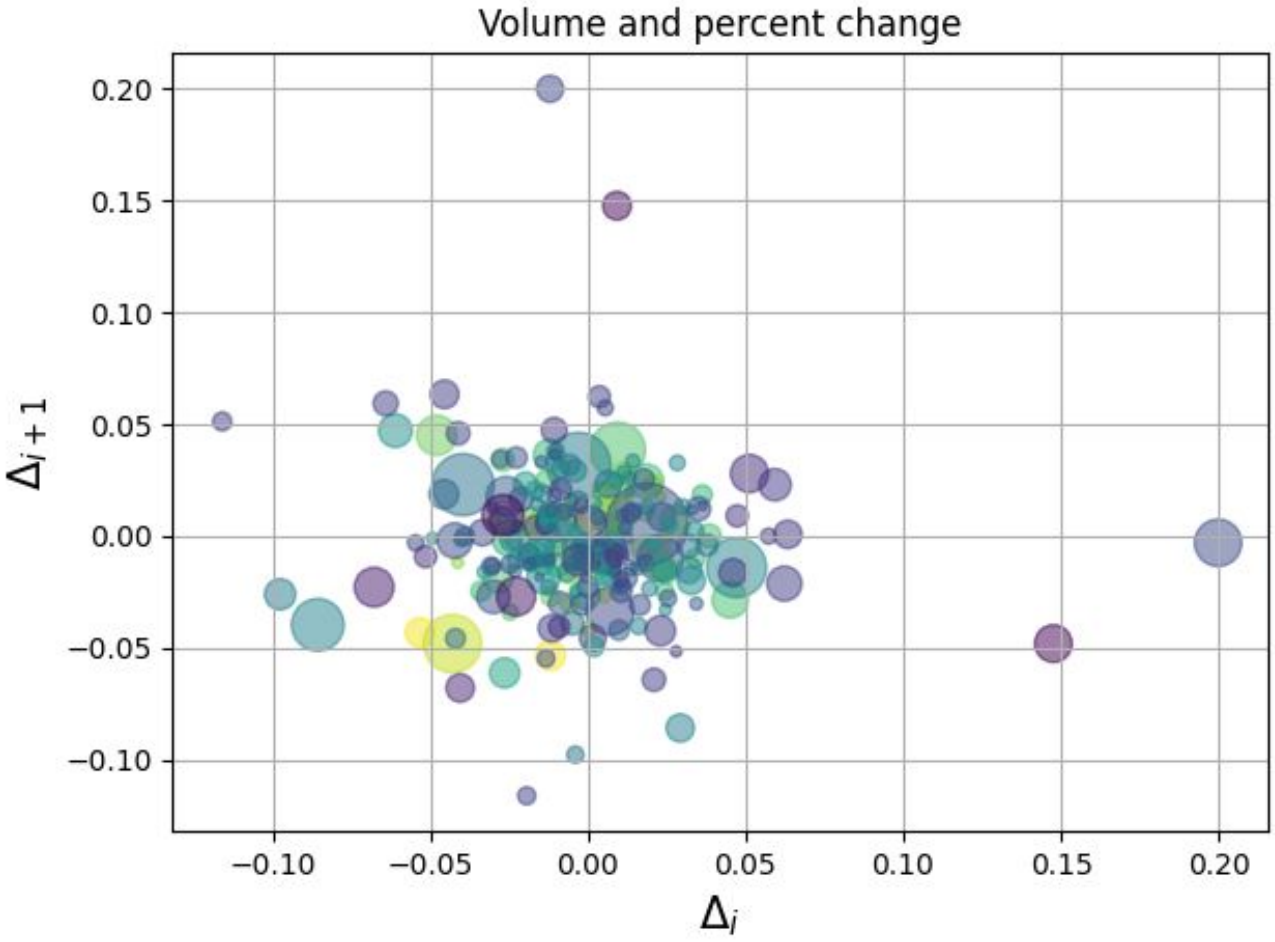
```
>>> ax1.hist(y)
>>> ax3.boxplot(y)
>>> ax3.violinplot(z)
```

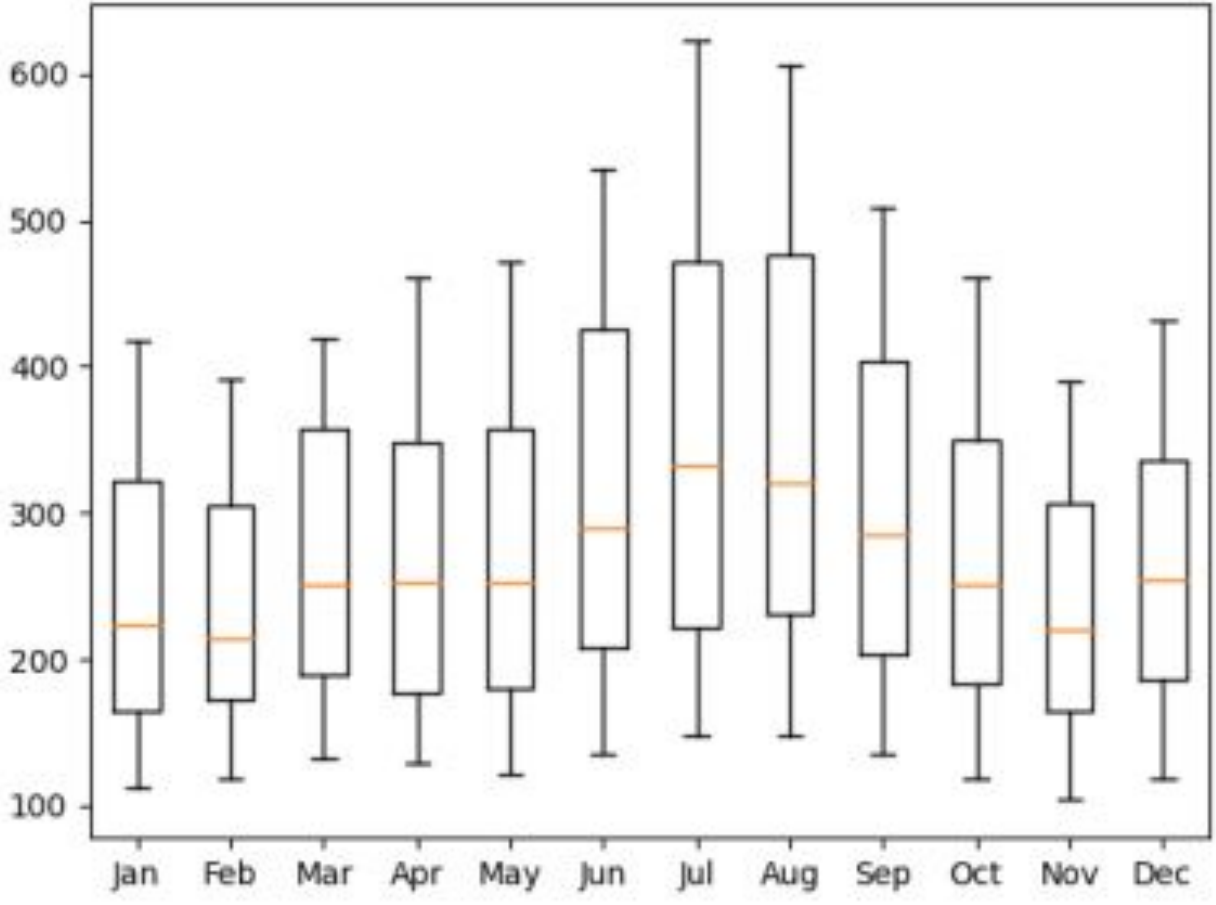
Plot a histogram

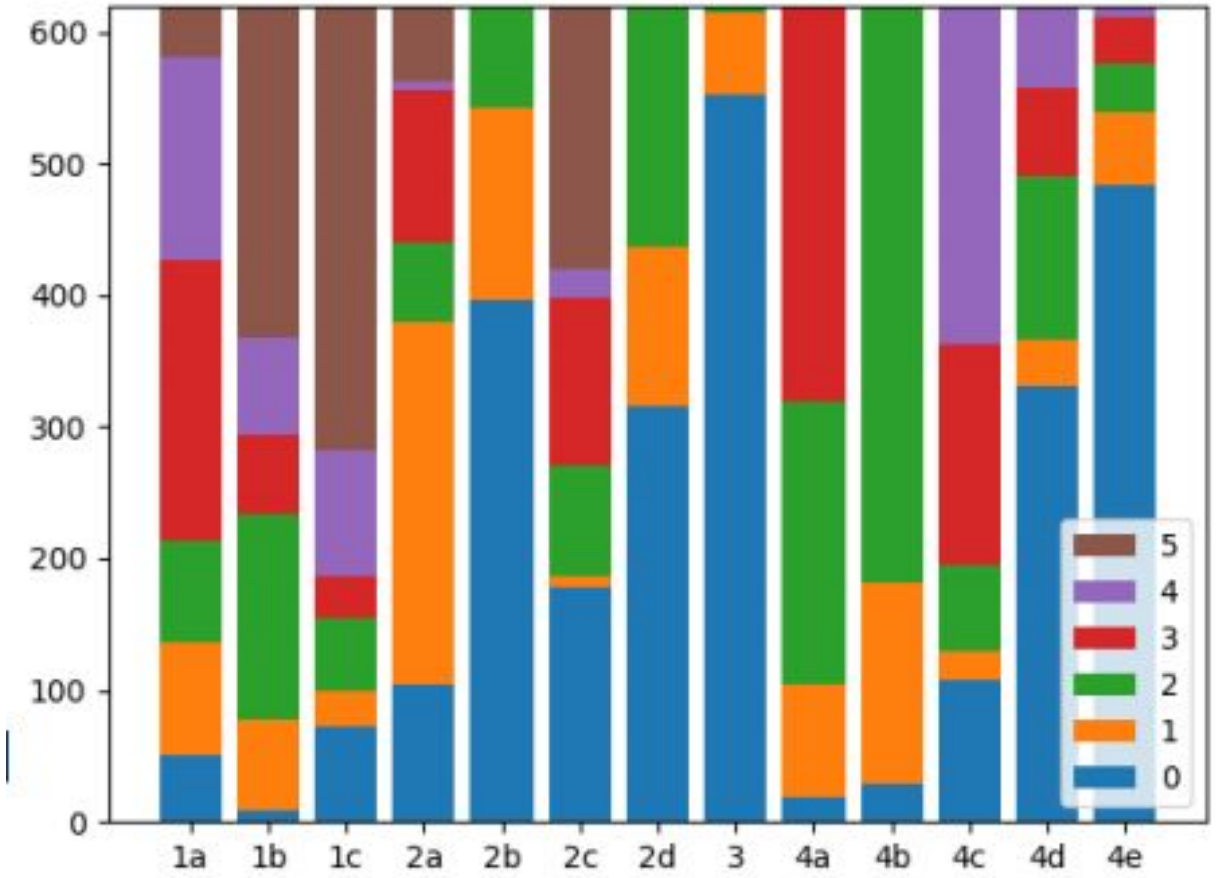
Make a box and whisker plot

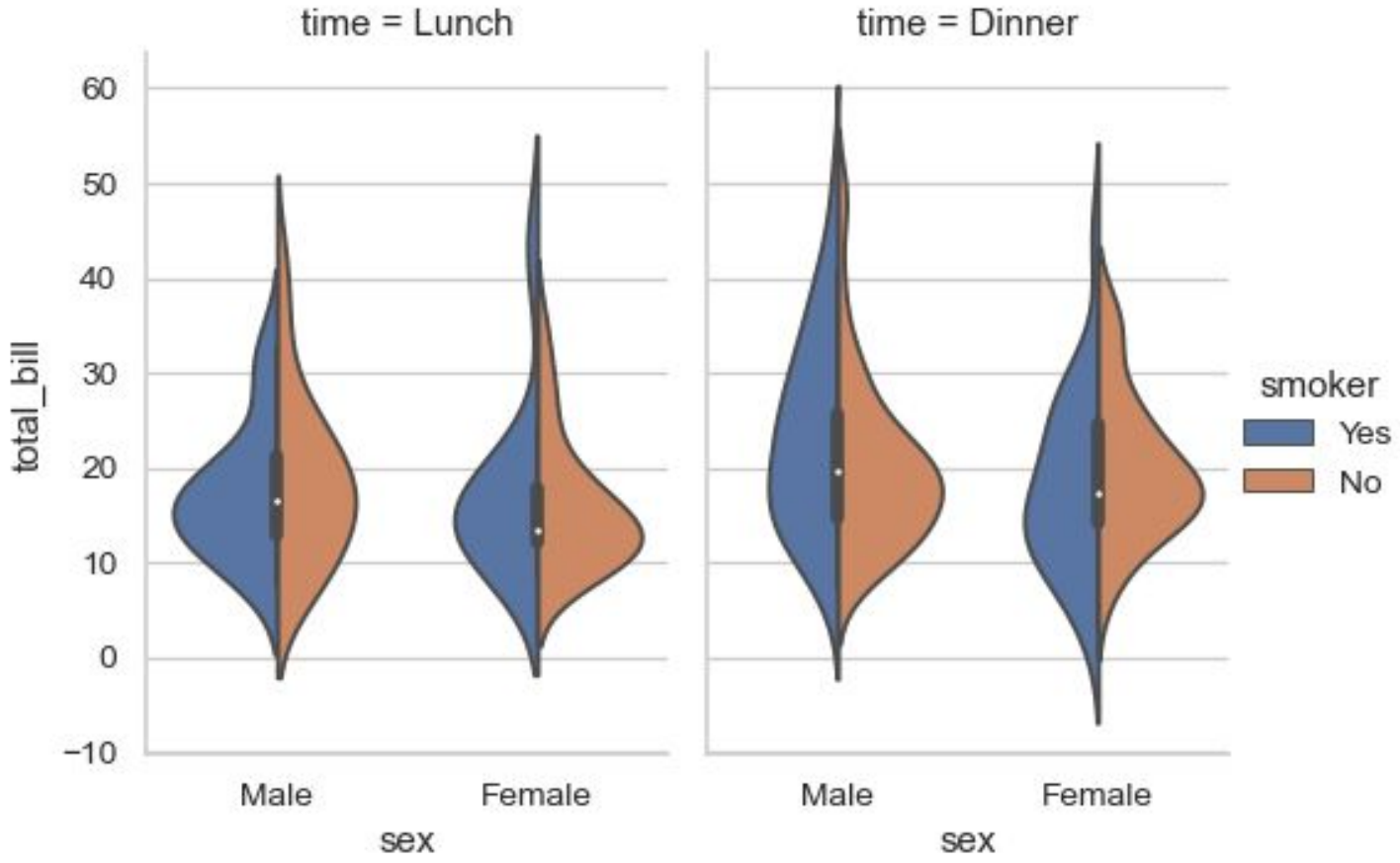
Make a violin plot


Johnny Doe
Push Ups
Agility
Mile Run
Flexed Arm Hang
Pacer Test
16th
60th
73rd
95th
37th
14
17 sec
12:52 min:sec
48 sec
7 laps
Test Scores
0
10
20
30
40
50
60
70
80
90
100
Percentile Ranking Across 2nd Grade Boys
Cohort Size: 62


Volume and percent change
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
Δ_{i+1}
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
Δ_i


600
500
400
300
200
100
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


600
500
400
300
200
100
0
1a
1b
1c
2a
2b
2c
2d
3
4a
4b
4c
4d
4e
5
4
3
2
1
0


time = Lunch
time = Dinner
total_bill
60
50
40
30
20
10
0
−10
Male
Female
sex
smoker
Yes
No

5. Save Plot

Save figures

```
>>> plt.savefig('foo.png')
```

Save transparent figures

```
>>> plt.savefig('foo.png', transparent=True)
```

6. Show Plot

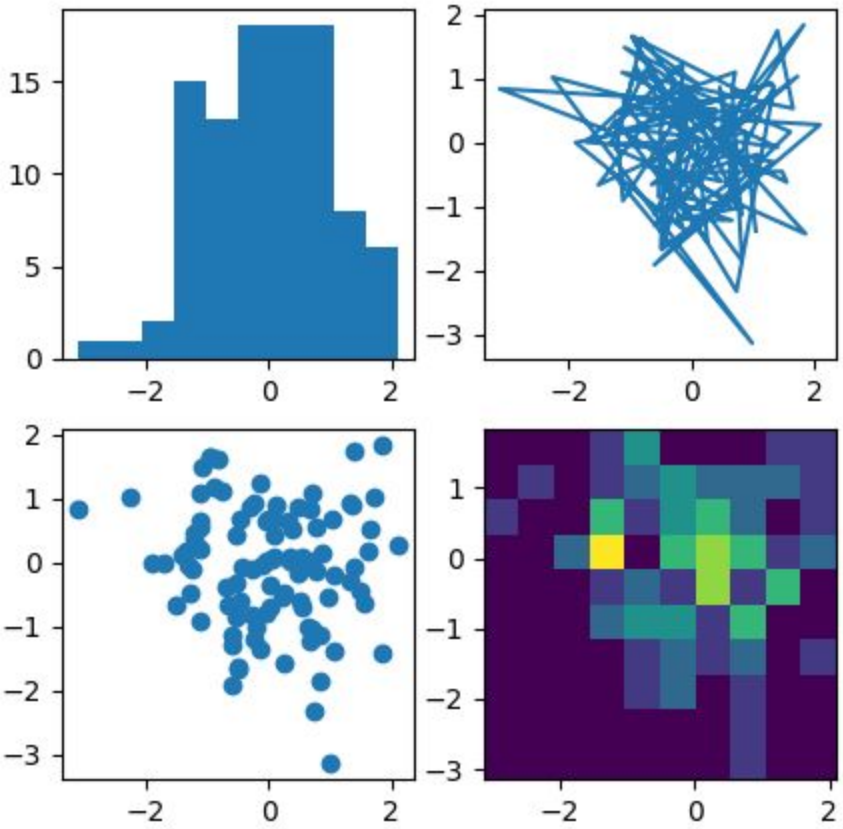
Simply do plt.show() !

(Optional) Close Plot

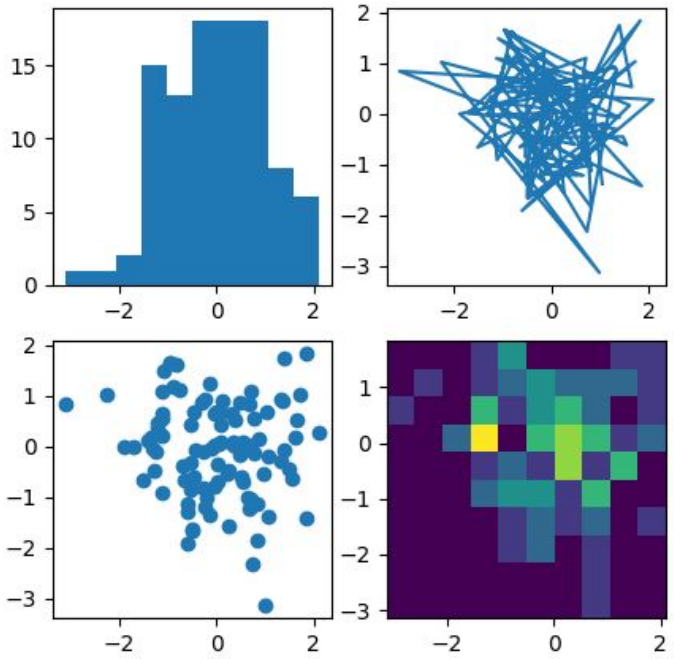
For the sake of formality and/or reproducibility, we usually close and/or clear the plot. The list of possible command is as follows:

- plt.cla()
- plt.clf()
- plt.close()

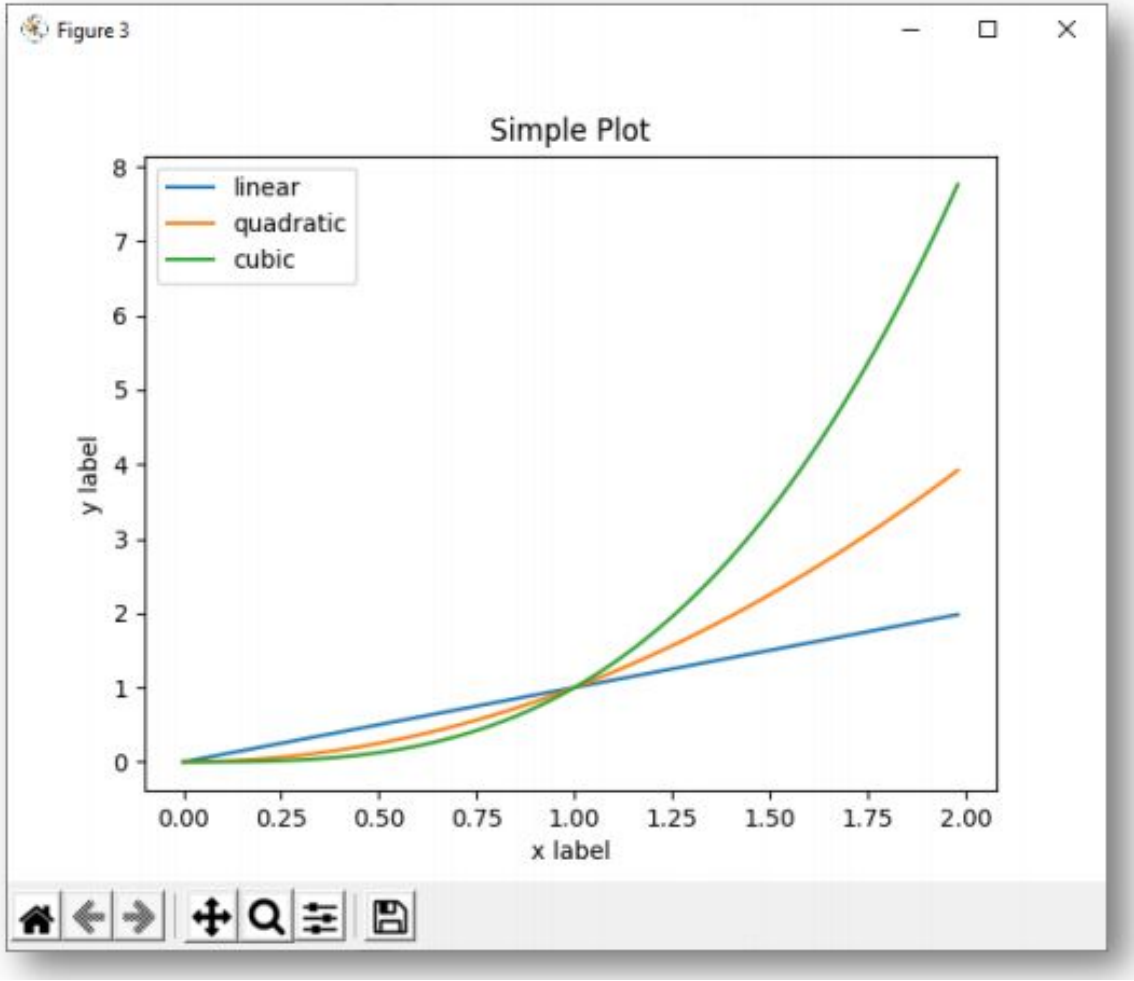
```
cla()   # Clear axis
clf()   # Clear figure
close() # Close a figure window
```


15
10
5
0
−2
0
2
2
1
0
−1
−2
−3
−2
0
2
2
1
0
−1
−2
−3
−2
0
2
1
0
−1
−2
−3
−2
0
2

```python
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(19680801)
data = np.random.randn(2, 100)

fig, axs = plt.subplots(2, 2, figsize=(5, 5))
axs[0, 0].hist(data[0])
axs[1, 0].scatter(data[0], data[1])
axs[0, 1].plot(data[0], data[1])
axs[1, 1].hist2d(data[0], data[1])

plt.show()
```

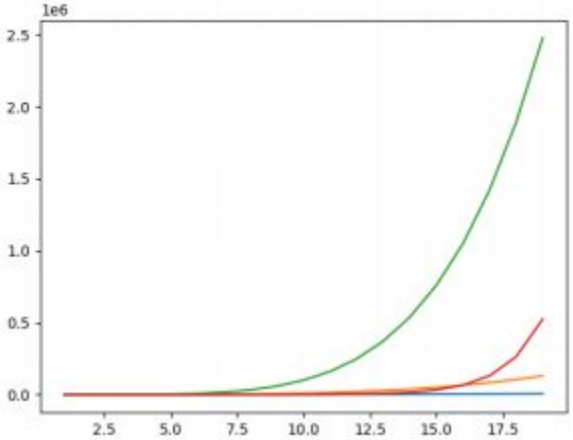


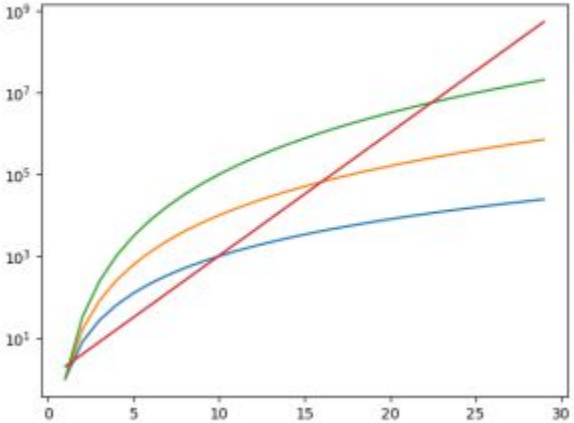
Try to plot this!

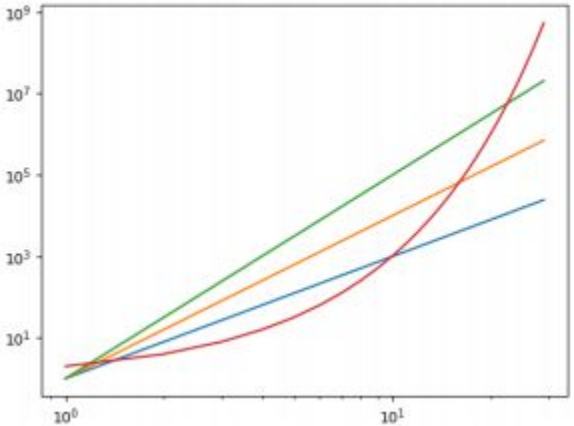


Break Time

- Which line is exponential growth?

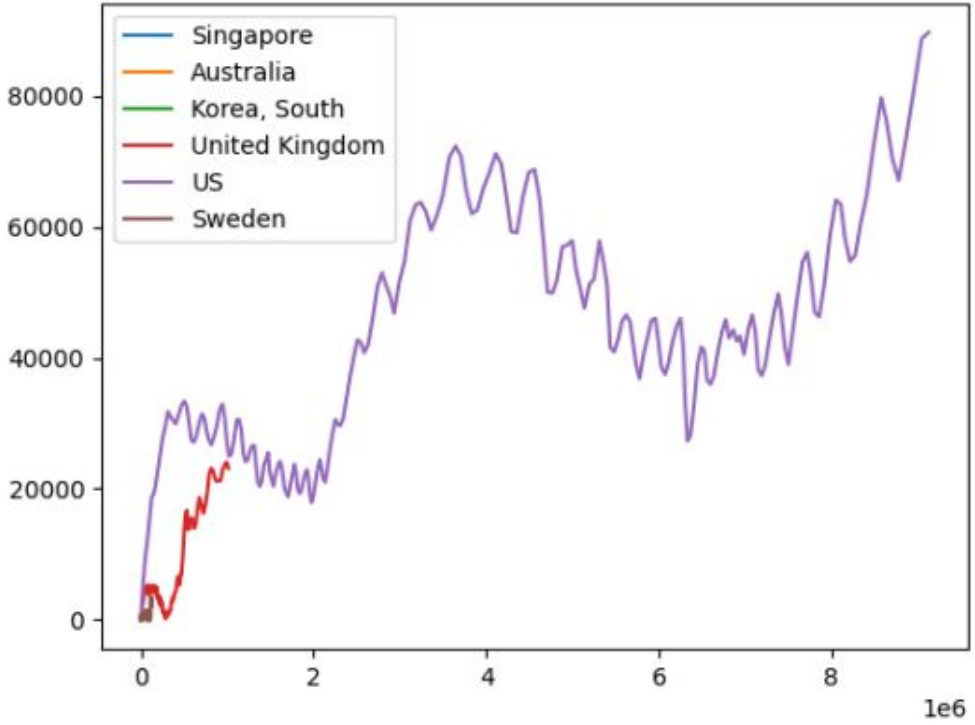






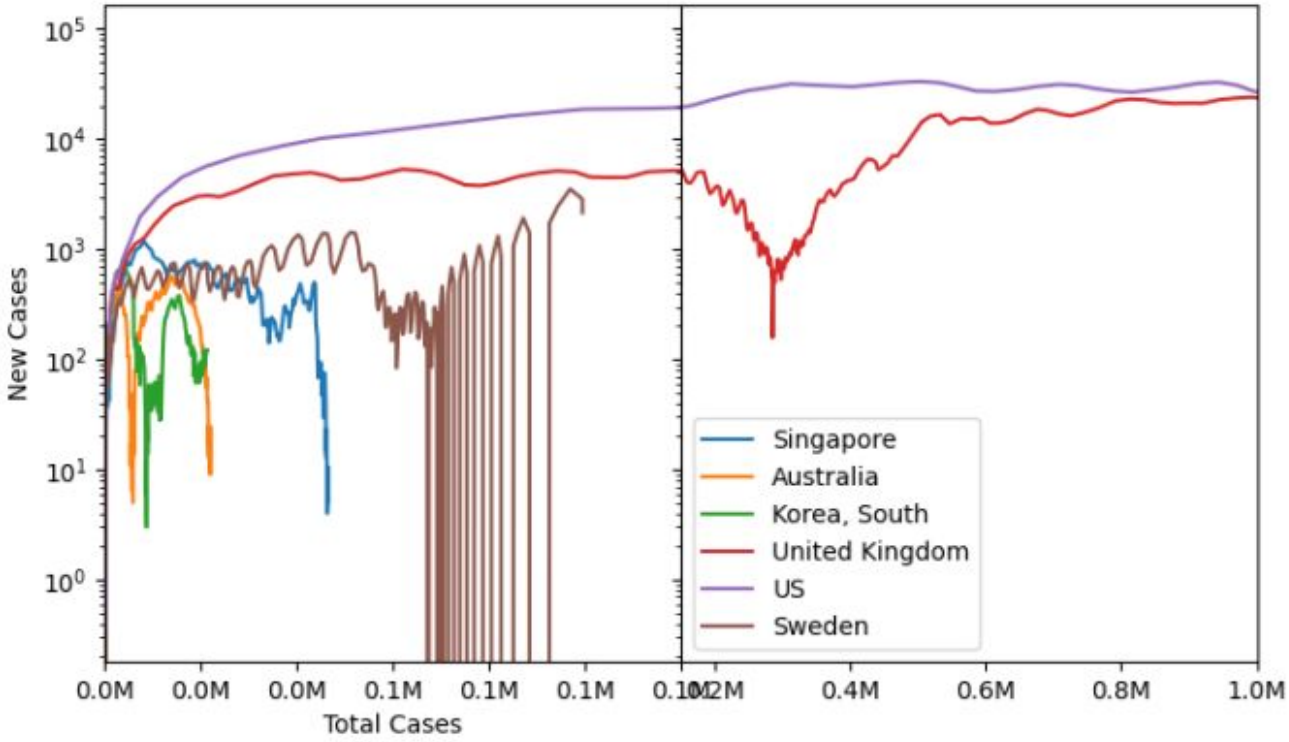
Linear-Linear

- Differences in plots are exponentially large



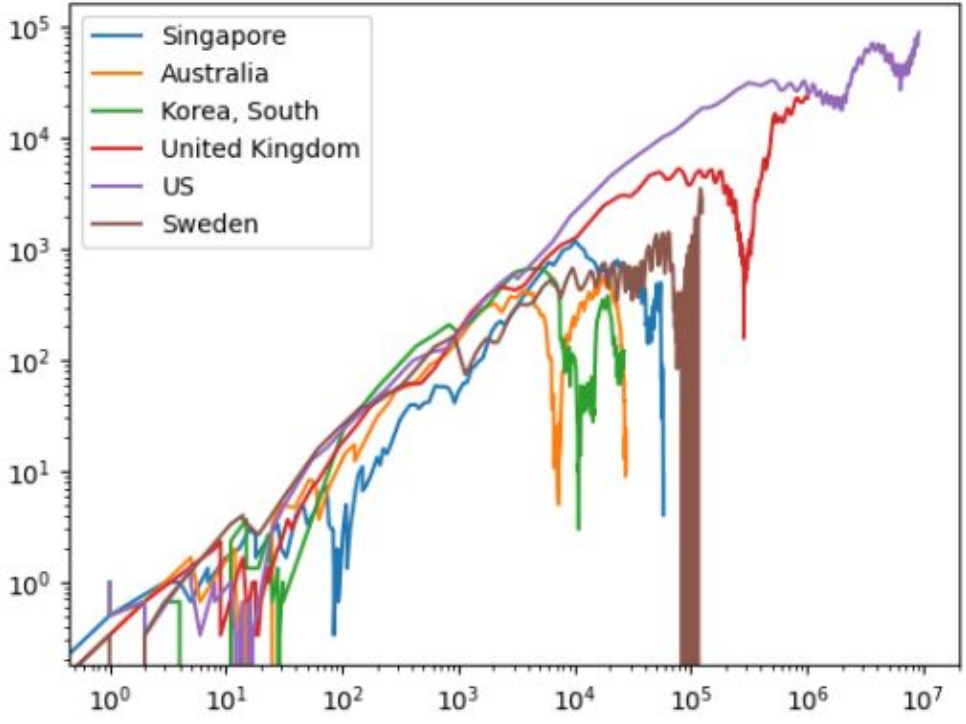
Log-Linear

- X-axis is split into two scales, but both are linear



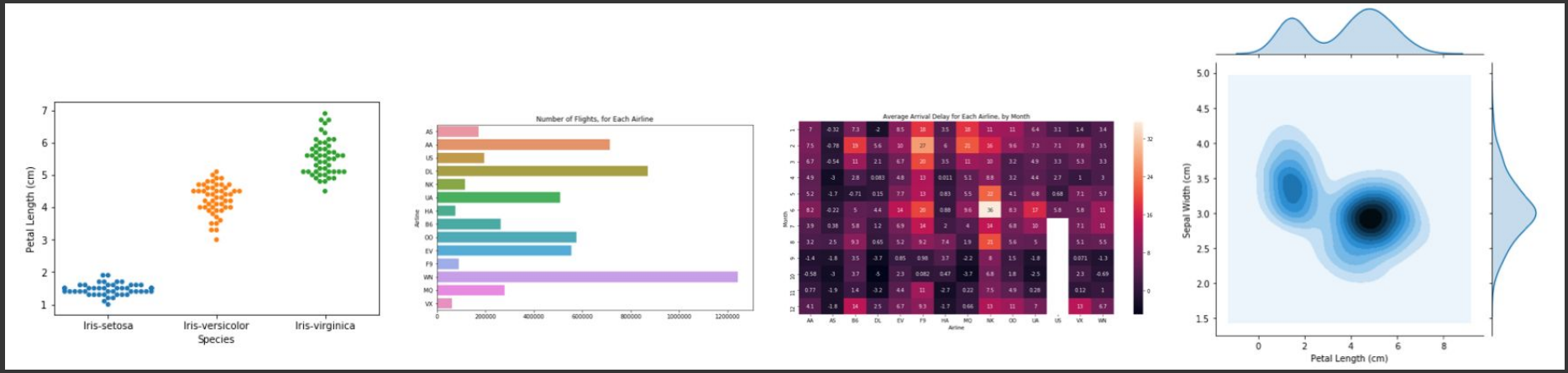
Log-Log

- Seems polynomial growth overall



Seaborn, implemented in Matplotlib

https://www.kaggle.com/learn/data-visualization



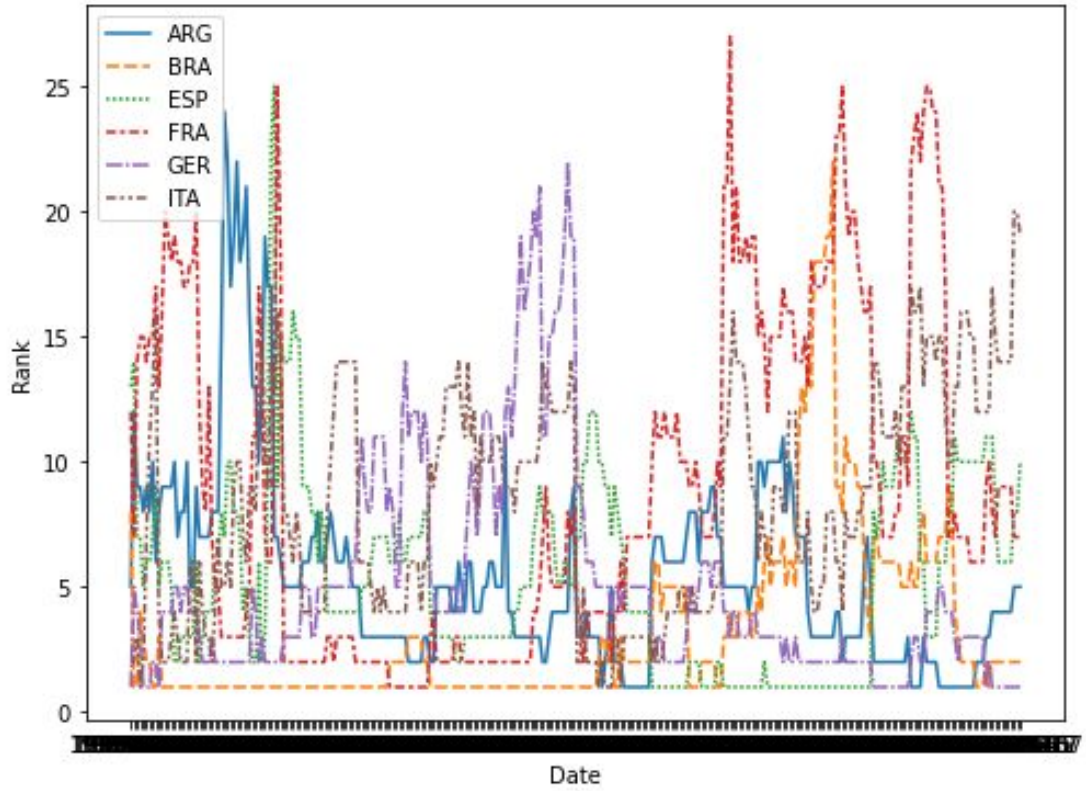
Line Plot

We are using fifa.csv in this example.

Date	ARG	BRA	ESP	FRA	GER	ITA
1993-08-08	5.0	8.0	13.0	12.0	1.0	2.0
1993-09-23	12.0	1.0	14.0	7.0	5.0	2.0
1993-10-22	9.0	1.0	7.0	14.0	4.0	3.0
1993-11-19	9.0	4.0	7.0	15.0	3.0	1.0
1993-12-23	8.0	3.0	5.0	15.0	1.0	2.0

```
plt.figure(figsize = (8,6))
sns.lineplot(data=fifa)
plt.ylabel("Rank")
```

```
Text(0, 0.5, 'Rank')
```



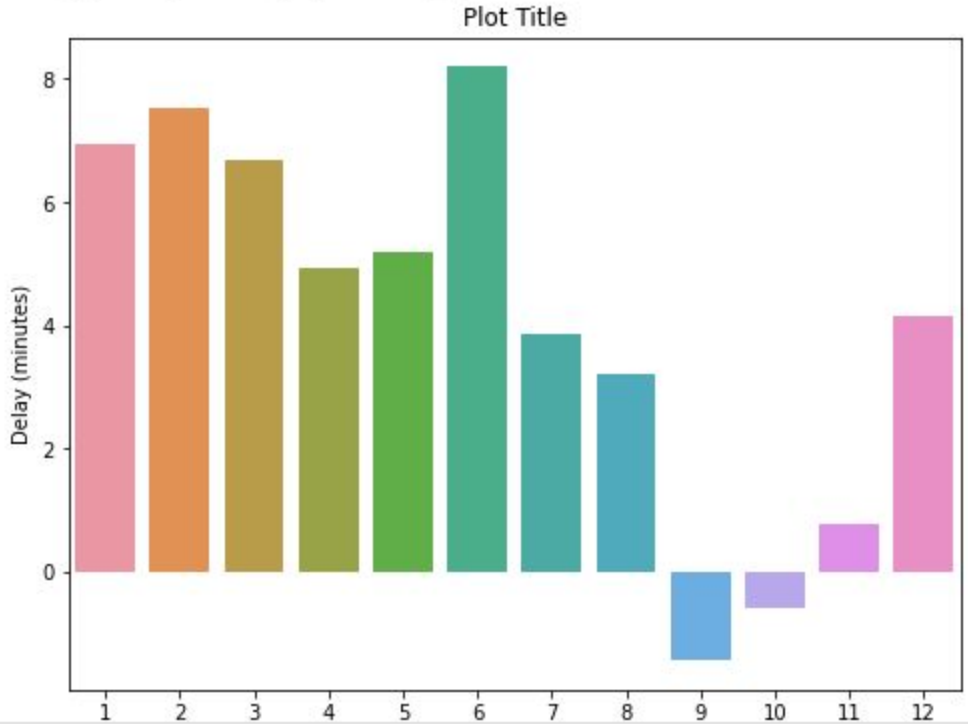
Bar Plot

This time, we are using flight_delays.csv

Month	AA	AS	B6	DL	EV	F9
1	6.955843	-0.320888	7.347281	-2.043847	8.537497	18.357238
2	7.530204	-0.782923	18.657673	5.614745	10.417236	27.424179
3	6.693587	-0.544731	10.741317	2.077965	6.730101	20.074855
4	4.931778	-3.009003	2.780105	0.083343	4.821253	12.640440
5	5.173878	-1.716398	-0.709019	0.149333	7.724290	13.007554

```
plt.figure(figsize=(8,6))
plt.title("Plot Title")
sns.barplot(x=flight_delays.index, y=flight_delays['AA'])
plt.xlabel("Month")
plt.ylabel("Delay (minutes)")
```

```
Text(0, 0.5, 'Delay (minutes)')
```



Heatmap

```
plt.figure(figsize=(12,6))
sns.heatmap(data = flight_delays, annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14462d8c10>
```

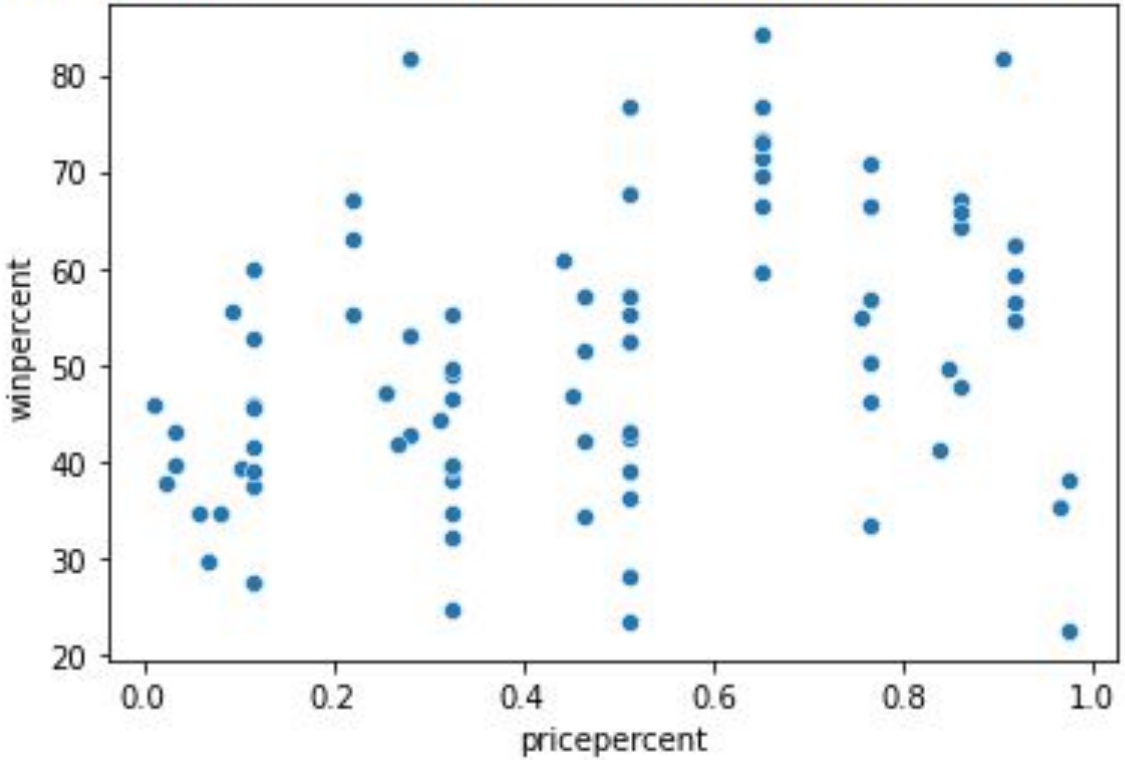
Scatter Plot

Now, we use candy.csv as the dataset.

id	competitorname	chocolate	fruity	caramel	peanutyalmondy	nougat
0	100 Grand	Yes	No	Yes	No	No
1	3 Musketeers	Yes	No	No	No	Yes
2	Air Heads	No	Yes	No	No	No
3	Almond Joy	Yes	No	No	Yes	No
4	Baby Ruth	Yes	No	Yes	Yes	Yes

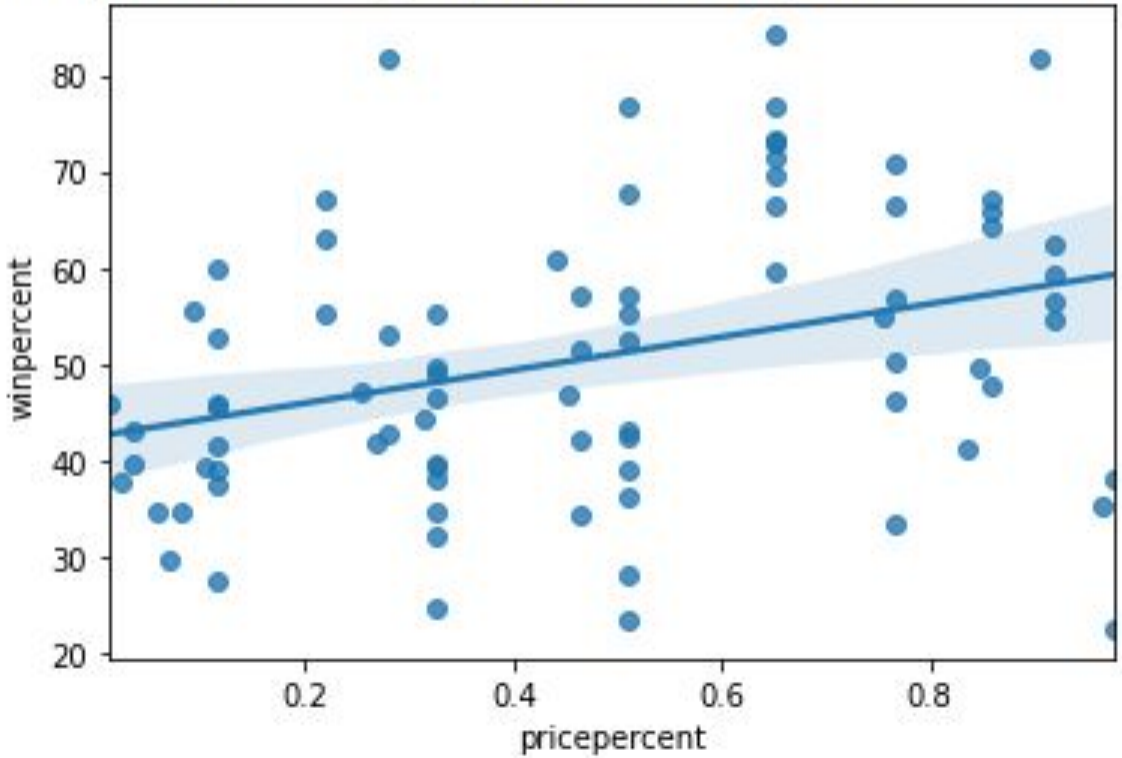
```
sns.scatterplot(x = candy['pricepercent'], y = candy['winpercent'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f144609c8d0>
```



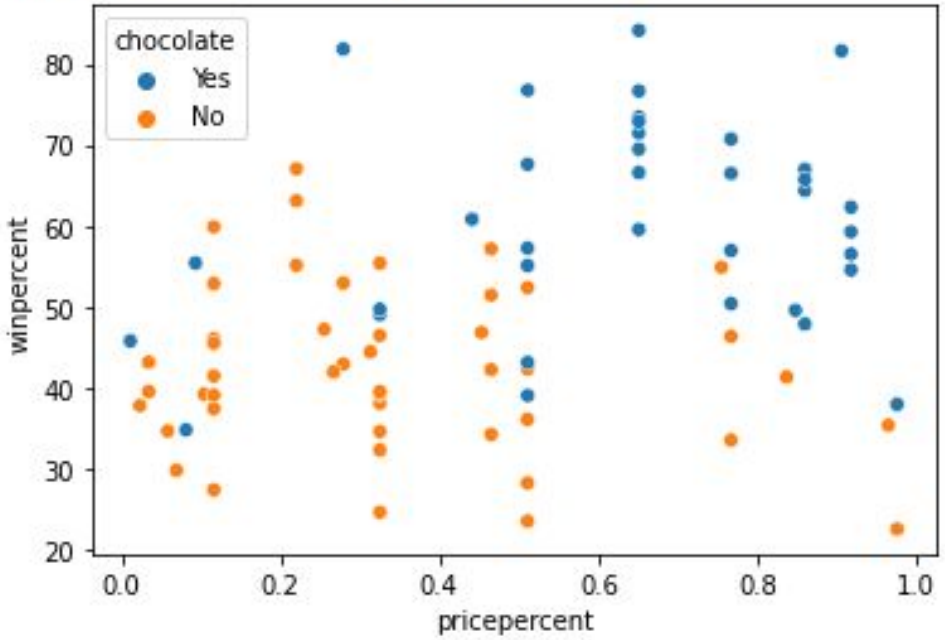
```
sns.regplot(x = candy['pricepercent'], y = candy['winpercent'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1445e794d0>
```



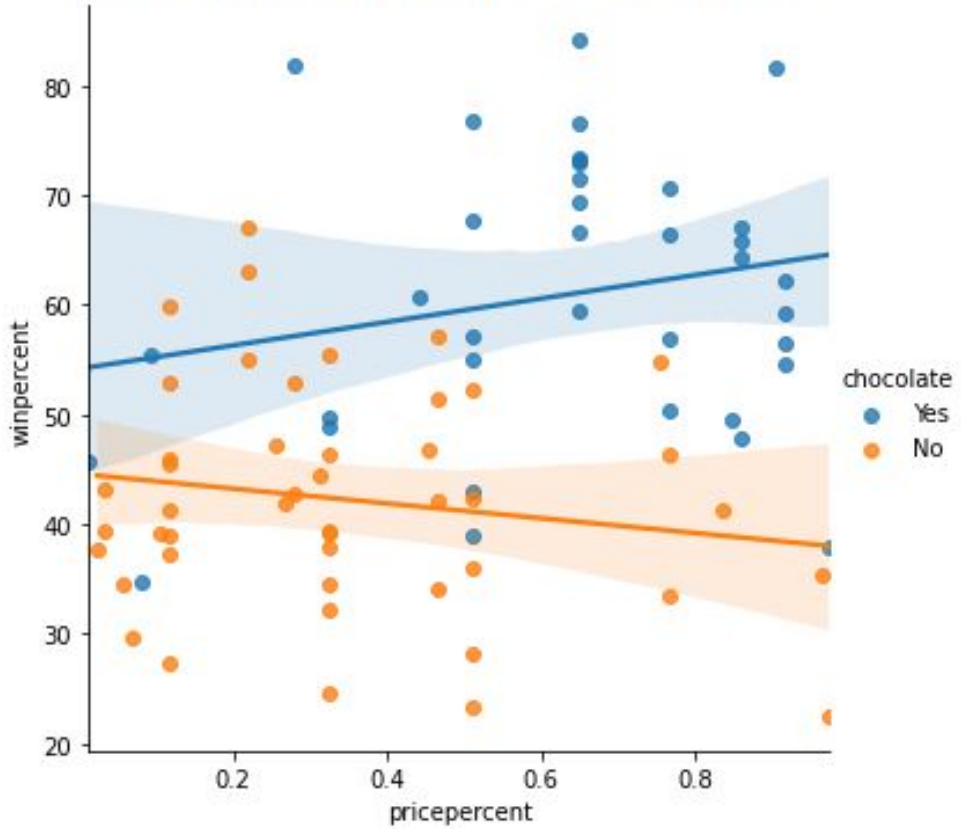
```
sns.scatterplot(x = candy['pricepercent'], y = candy['winpercent'], hue = candy['chocolate'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1445e424d0>
```



```
sns.lmplot(x = 'pricepercent', y = 'winpercent', hue = 'chocolate', data = candy)
```

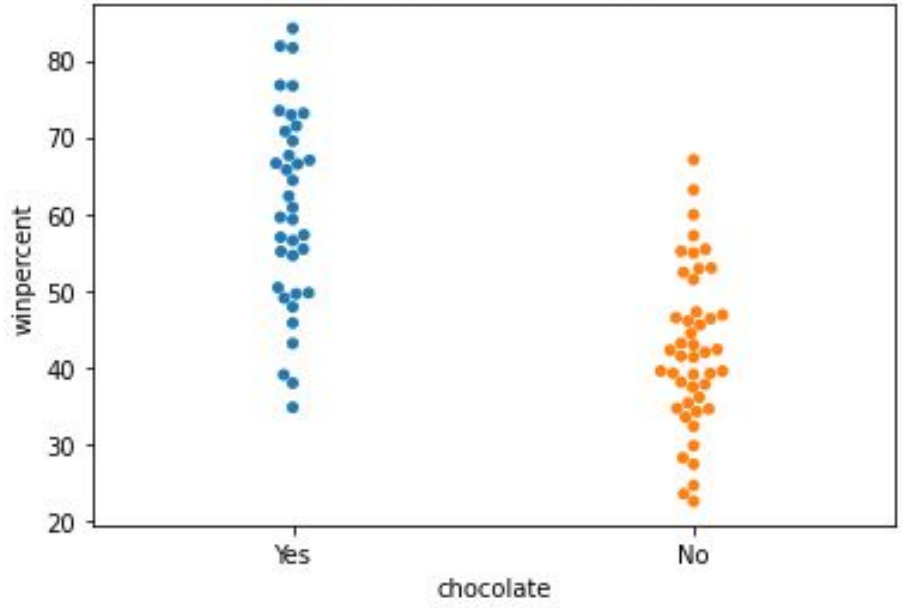
```
<seaborn.axisgrid.FacetGrid at 0x7f1445df51d0>
```



Categorical Scatterplot : Swarmplot

```
sns.swarmplot(x = candy['chocolate'], y = candy['winpercent'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1445ce6b90>
```



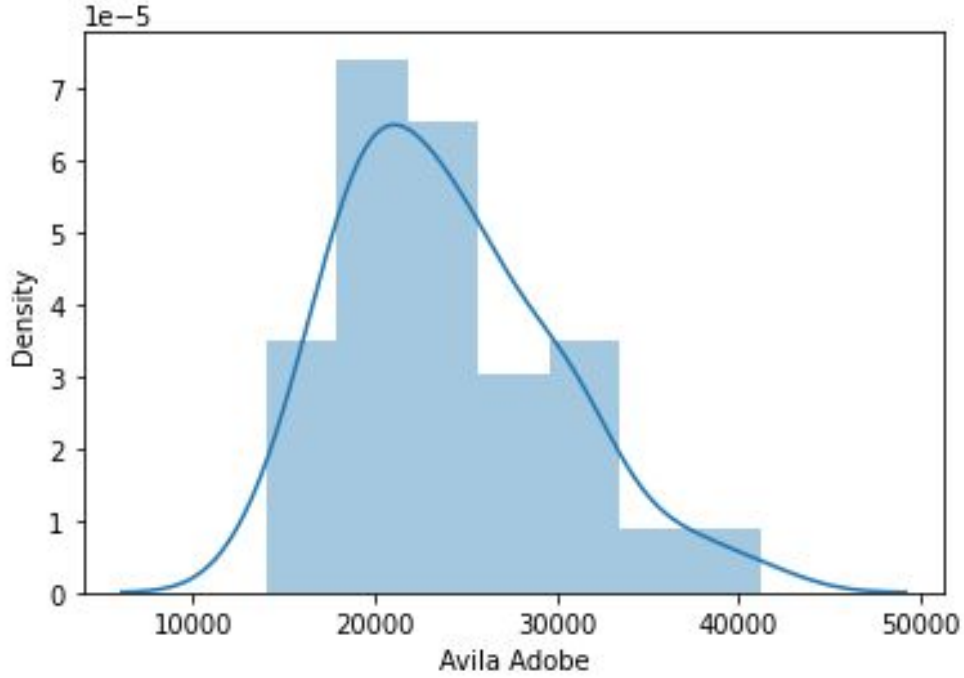
Histograms

Finally, we will demonstrate a Seaborn histogram from museum_visitors.csv as the dataset.

Date	Avila Adobe	Firehouse Museum	Chinese American Museum	America Tropical Interpretive Center
2014-01-01	24778	4486	1581	6602
2014-02-01	18976	4172	1785	5029
2014-03-01	25231	7082	3229	8129
2014-04-01	26989	6756	2129	2824
2014-05-01	36883	10858	3676	10694

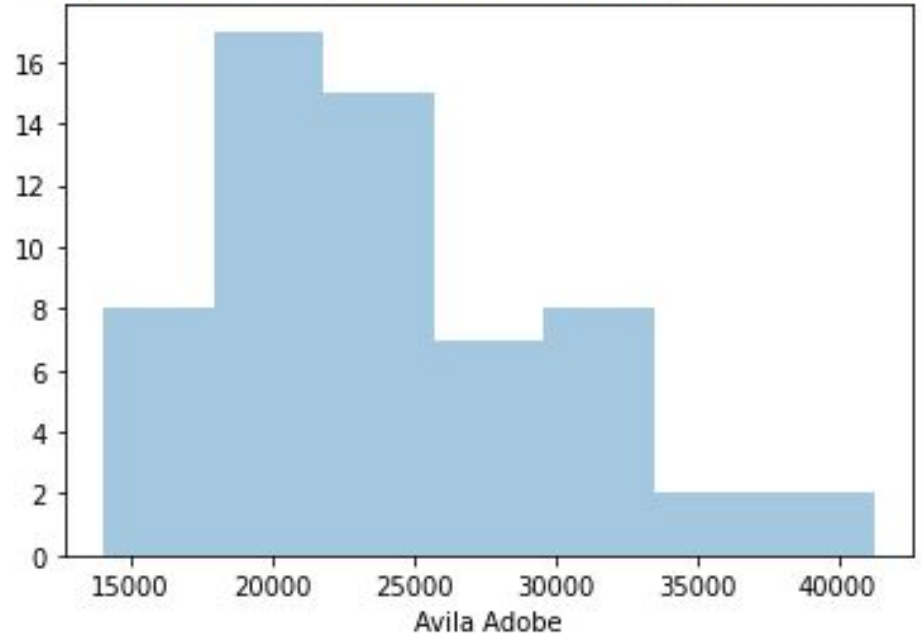
```
sns.distplot(museum['Avila Adobe'], kde = True) # kernel density estimate
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: Futur
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f1443b01c10>
```



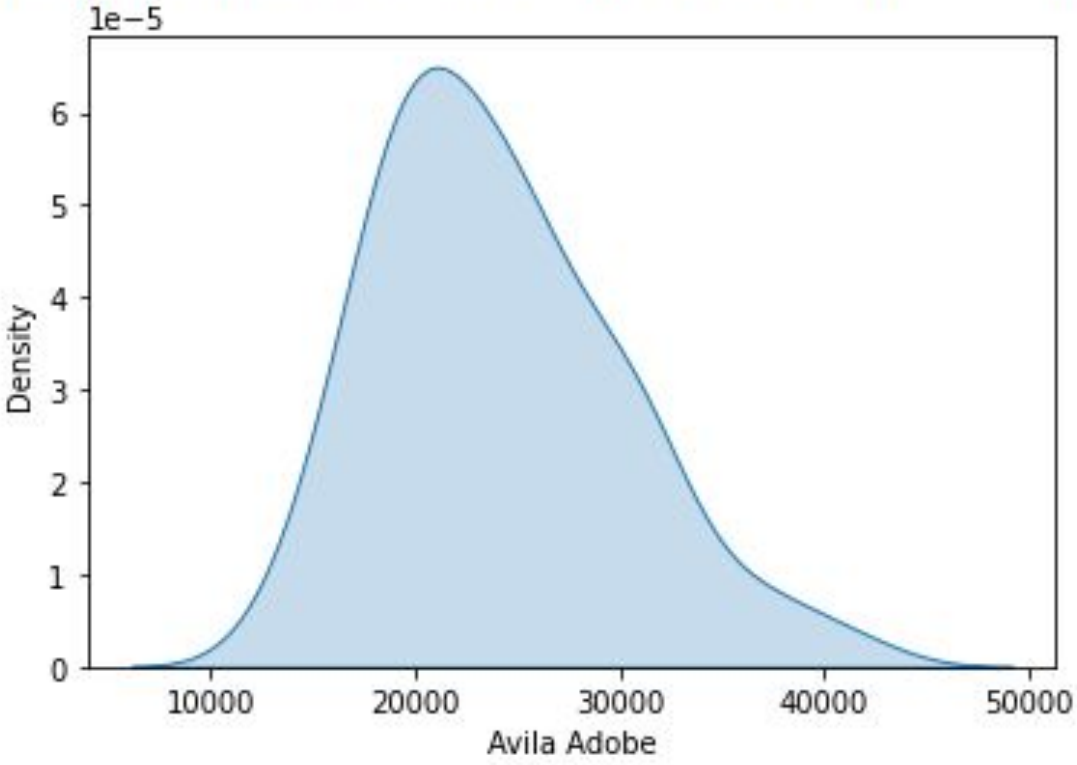
```
sns.distplot(museum['Avila Adobe'], kde = False) # kernel density estimate
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: Future
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f14461b9090>
```



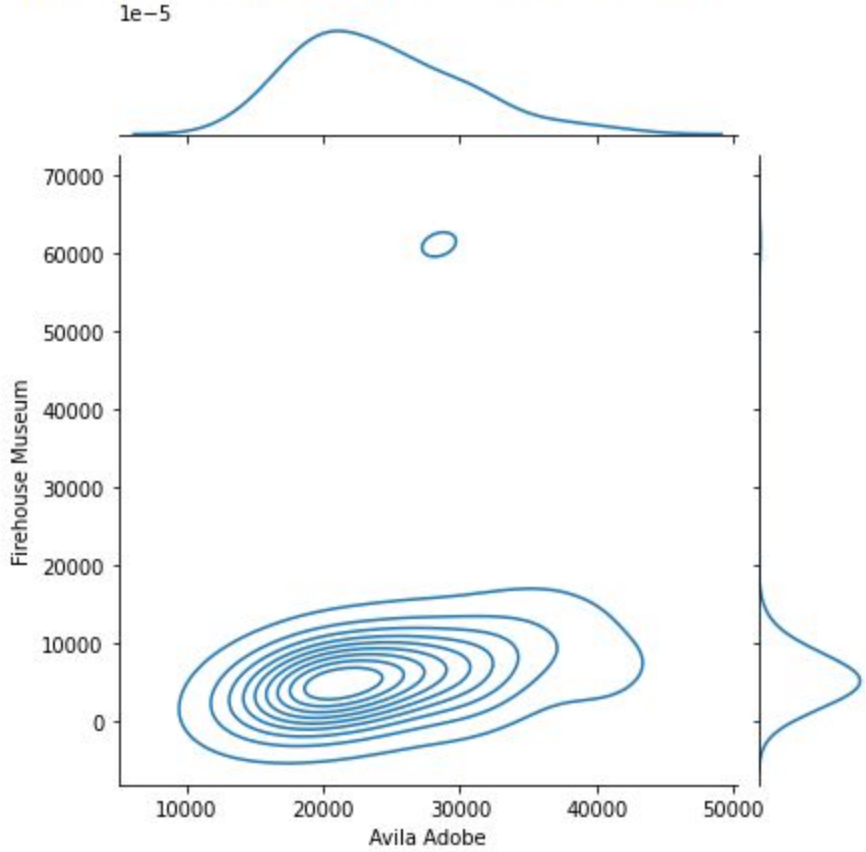
```
sns.kdeplot(data = museum['Avila Adobe'], shade = True)
# shade gives background colour to the area under the curve
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f144f0b8790>
```



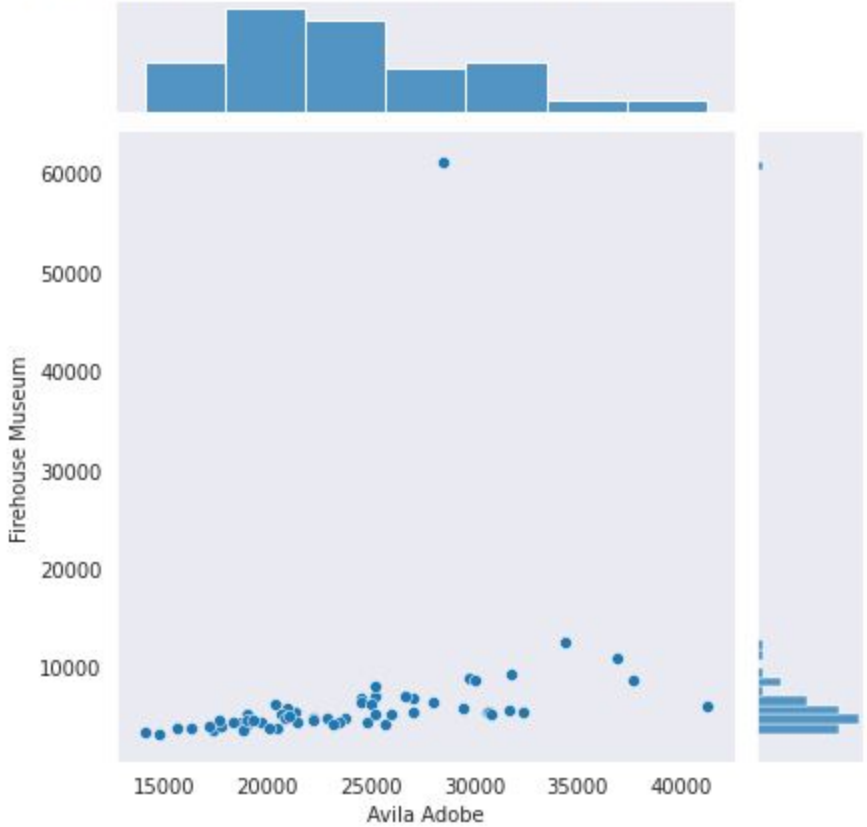
```
sns.jointplot(x = museum['Avila Adobe'], y = museum['Firehouse Museum'], kind = 'kde')
```

```
<seaborn.axisgrid.JointGrid at 0x7f14437c1910>
```



```
sns.jointplot(x = museum['Avila Adobe'], y = museum['Firehouse Museum'])
```

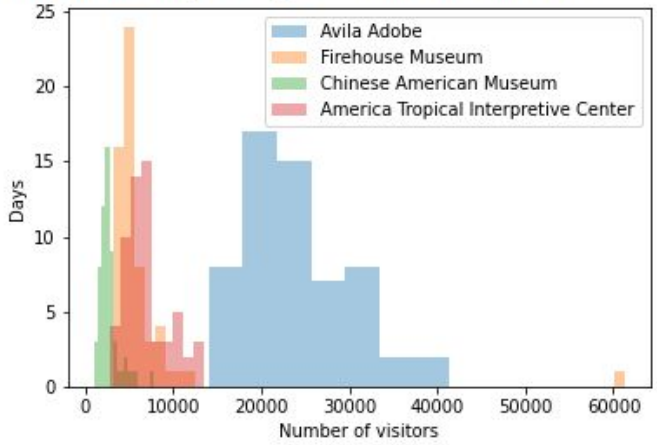
```
<seaborn.axisgrid.JointGrid at 0x7f1442531ad0>
```



Putting it all together

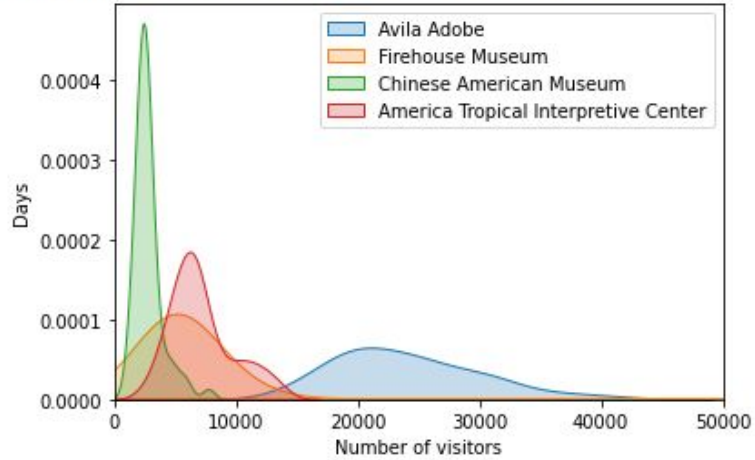
```
for c in museum.columns:
  sns.distplot(museum[c], kde = False, label = c)

plt.xlabel("Number of visitors")
plt.ylabel("Days")
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distr
  warnings.warn(msg, FutureWarning)
<matplotlib.legend.Legend at 0x7f14432eaa90>
```




```
for c in museum.columns:
  sns.kdeplot(museum[c], shade = True, label = c)

plt.xlabel("Number of visitors")
plt.ylabel("Days")
plt.xlim(0,50000)
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f1442cfde90>
```



Styling

Here are 5 different stylings in Seaborn.

(1)"darkgrid"

(2)"whitegrid"

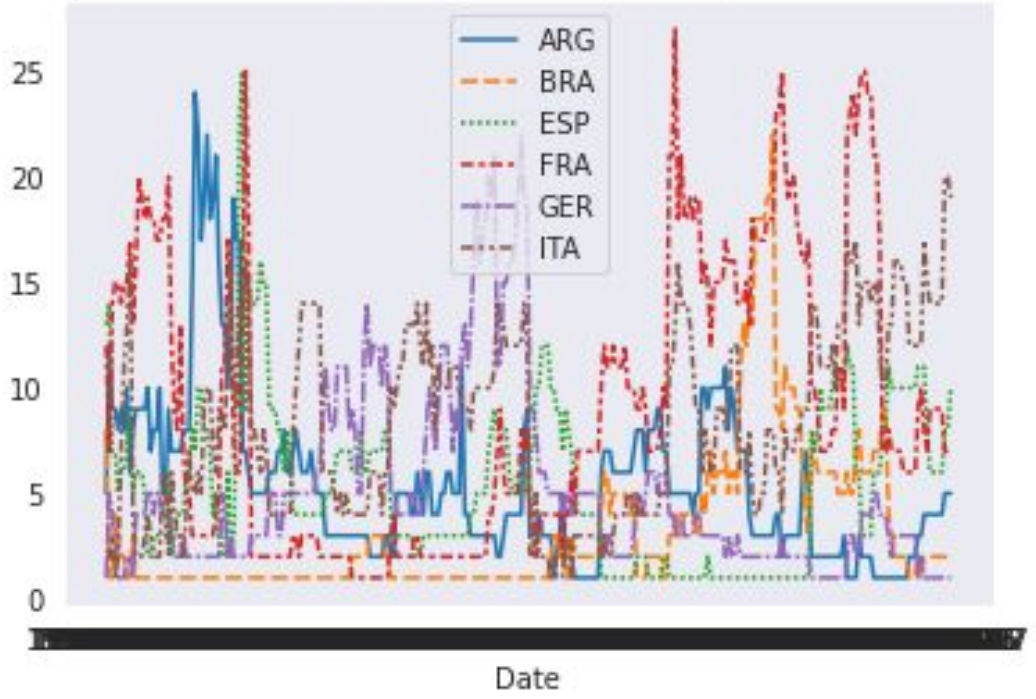
(3)"dark"

(4)"white", and

(5)"ticks"

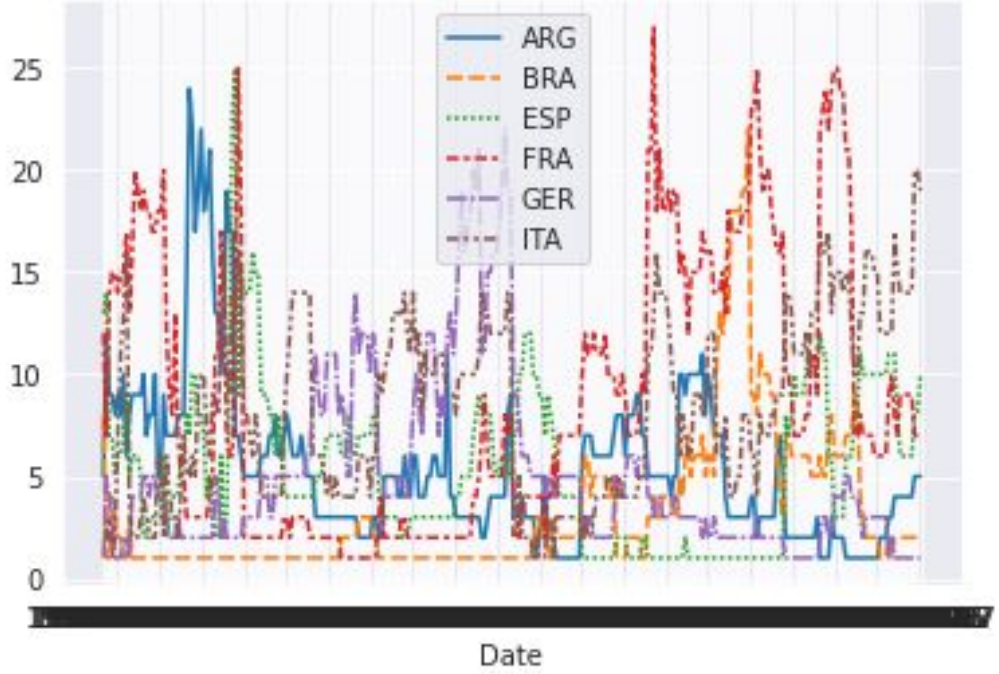
```
sns.set_style("dark")
sns.lineplot(data=fifa)
# (1)"darkgrid", (2)"whitegrid", (3)"dark", (4)"white", and (5)"ticks"
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1443236c10>
```



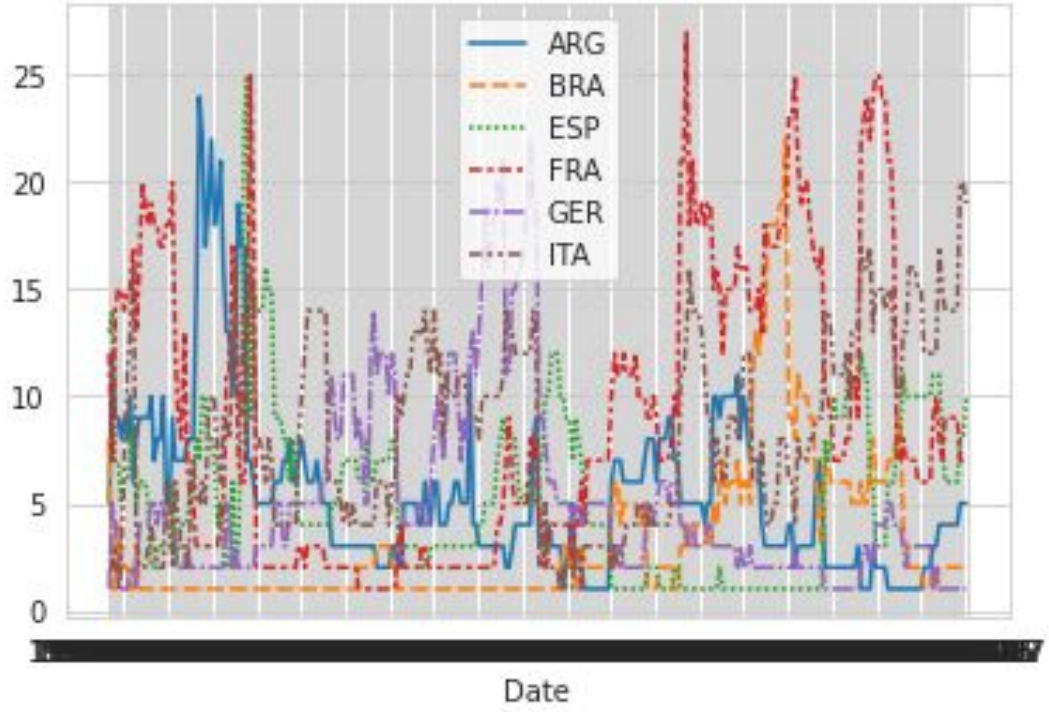
```
sns.set_style("darkgrid")
sns.lineplot(data=fifa)
# (1)"darkgrid", (2)"whitegrid", (3)"dark", (4)"white", and (5)"ticks"
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1441d9e450>
```



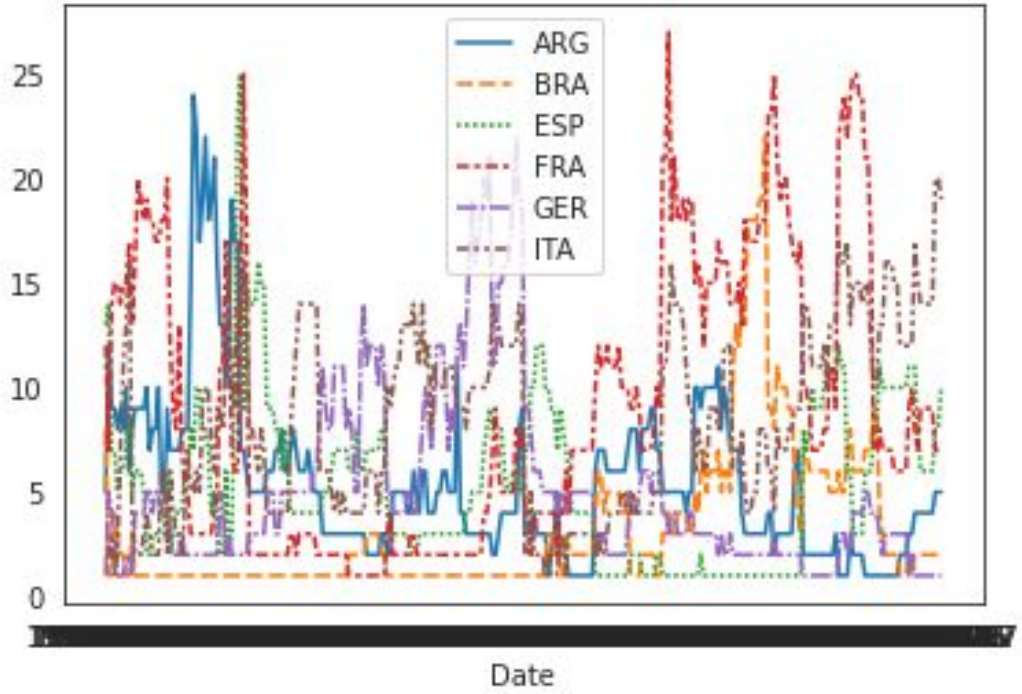
```
sns.set_style("whitegrid")
sns.lineplot(data=fifa)
# (1)"darkgrid", (2)"whitegrid", (3)"dark", (4)"white", and (5)"ticks"
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1441ae0ed0>
```



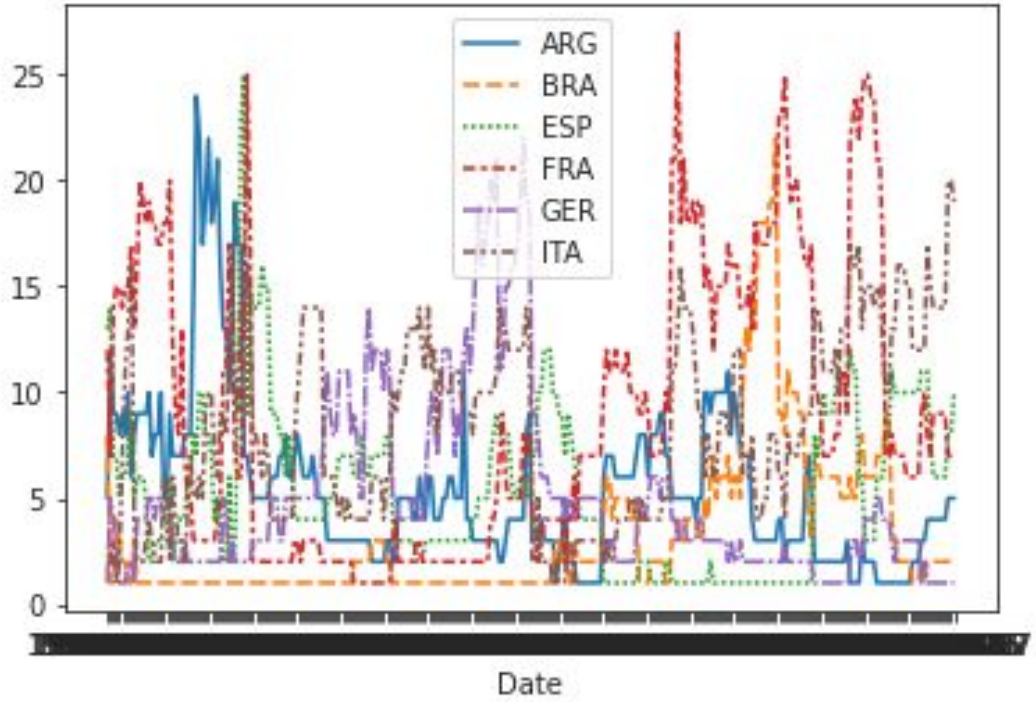
```
sns.set_style("white")
sns.lineplot(data=fifa)
# (1)"darkgrid", (2)"whitegrid", (3)"dark", (4)"white", and (5)"ticks"
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14416833d0>
```



```
sns.set_style("ticks")
sns.lineplot(data=fifa)
# (1)"darkgrid", (2)"whitegrid", (3)"dark", (4)"white", and (5)"ticks"
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f14413a2a90>
```



Notice the small ticks here

Other References

https://matplotlib.org/stable/tutorials/introductory/sample_plots.html

https://matplotlib.org/3.1.0/gallery/index.html

https://seaborn.pydata.org/examples/index.html

https://www.data-to-viz.com/

QnA