

- Shin's Lab -

Python for Data Visualization

Python for Data Visualization

-Chapter.2 Line Plot -

2-00. Intro to Line Plot

2-01. Line Plot Basics

2-02. Labels and Legend

2-03. Line Styles and Markers

2-04. Line Filling

2-05. Exercises

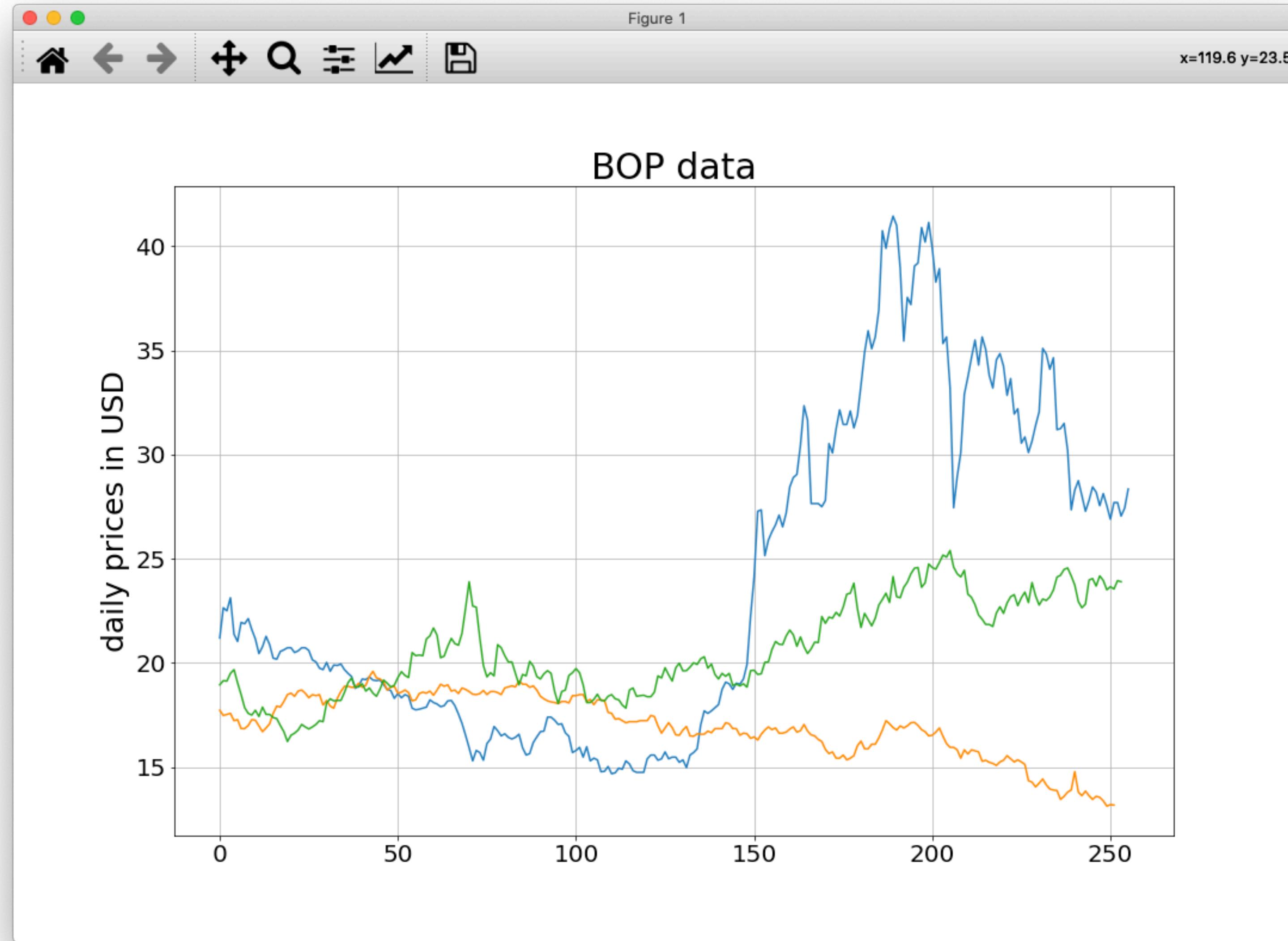
Python for Data Visualization

-Chapter.2 Line Plot -

2-02. Labels and Legend

1. Basic Usage
2. Legend Locations
3. ncol Argument
4. bbox_to_anchor Argument
5. Labels on BOP data

1. Basic Usage



Lecture. 2-02 Labels and Legend

5

1. Basic Usage

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

np.random.seed(0)

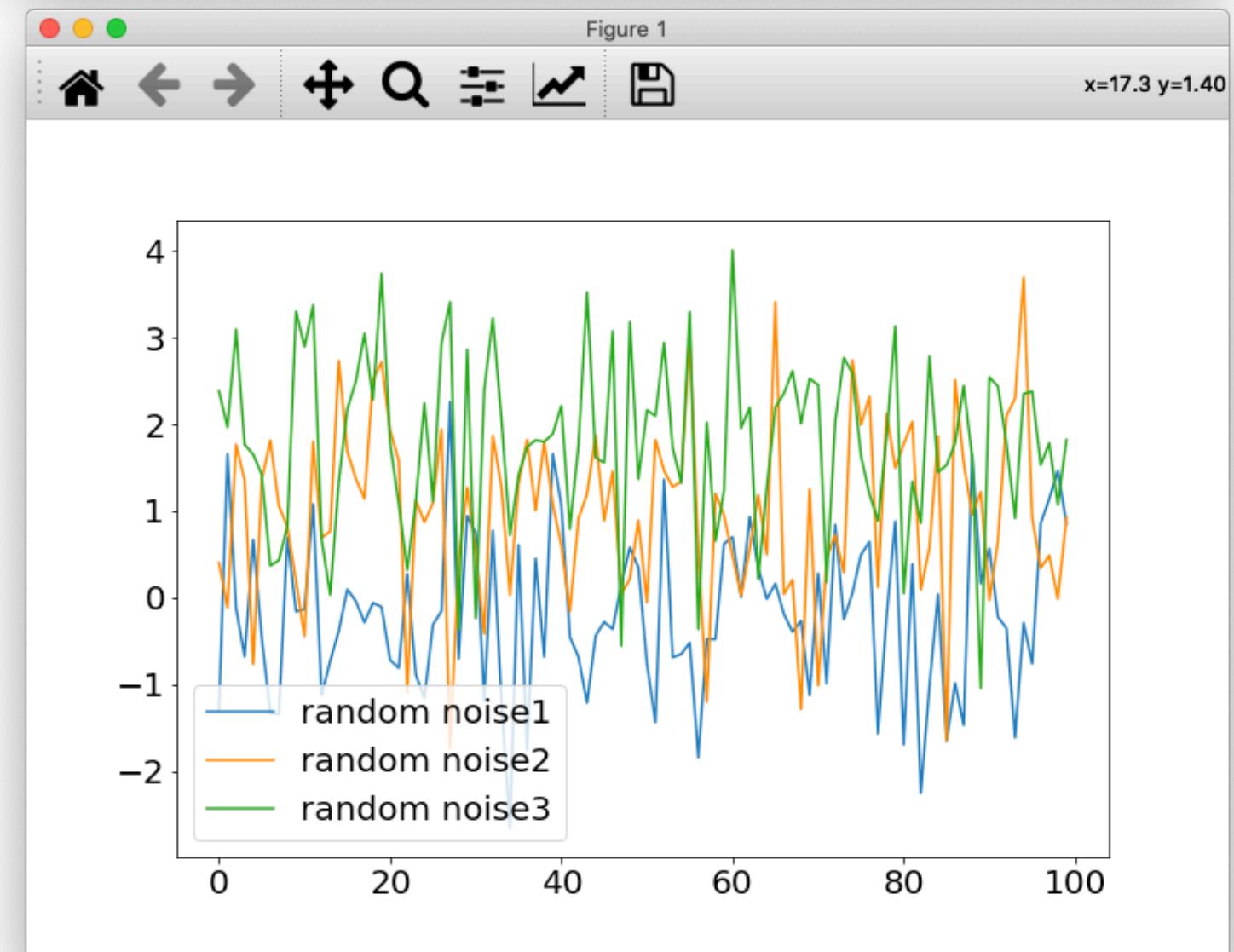
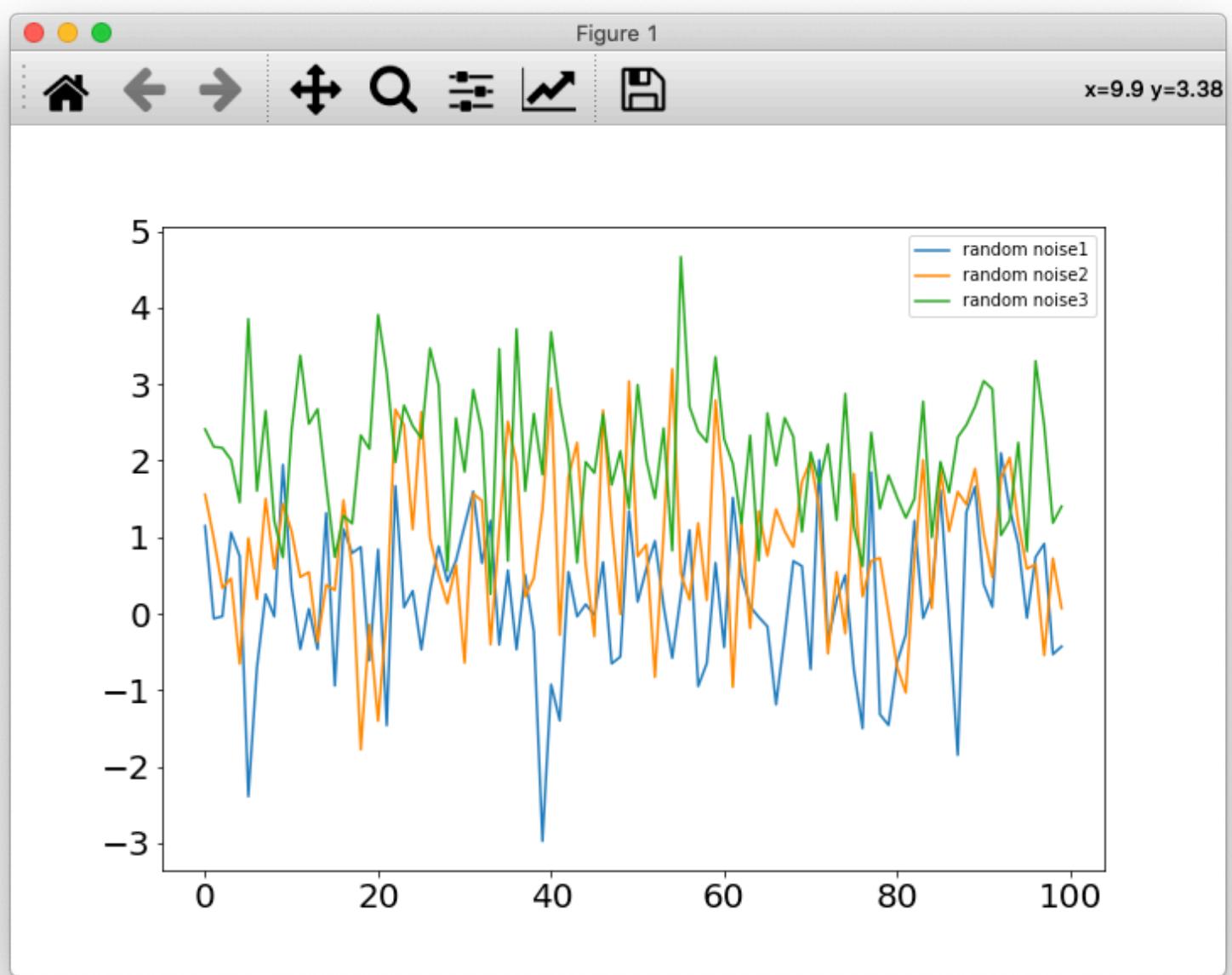
n_data = 100
random_noise1 = np.random.normal(0, 1, (n_data,))
random_noise2 = np.random.normal(1, 1, (n_data,))
random_noise3 = np.random.normal(2, 1, (n_data,))

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=20)

ax.plot(random_noise1,
        label='random noise1')
ax.plot(random_noise2,
        label='random noise2')
ax.plot(random_noise3,
        label='random noise3')

ax.legend()

-----
ax.legend(fontsize=20)
```



Lecture. 2-02 Labels and Legend

6

1. Basic Usage

```
def bop_plot(dataset, t_year, ax):
    t_year_data = get_year_data(dataset, t_year)
    ax.plot(t_year_data[:, -1],
            label='Year ' + str(t_year))

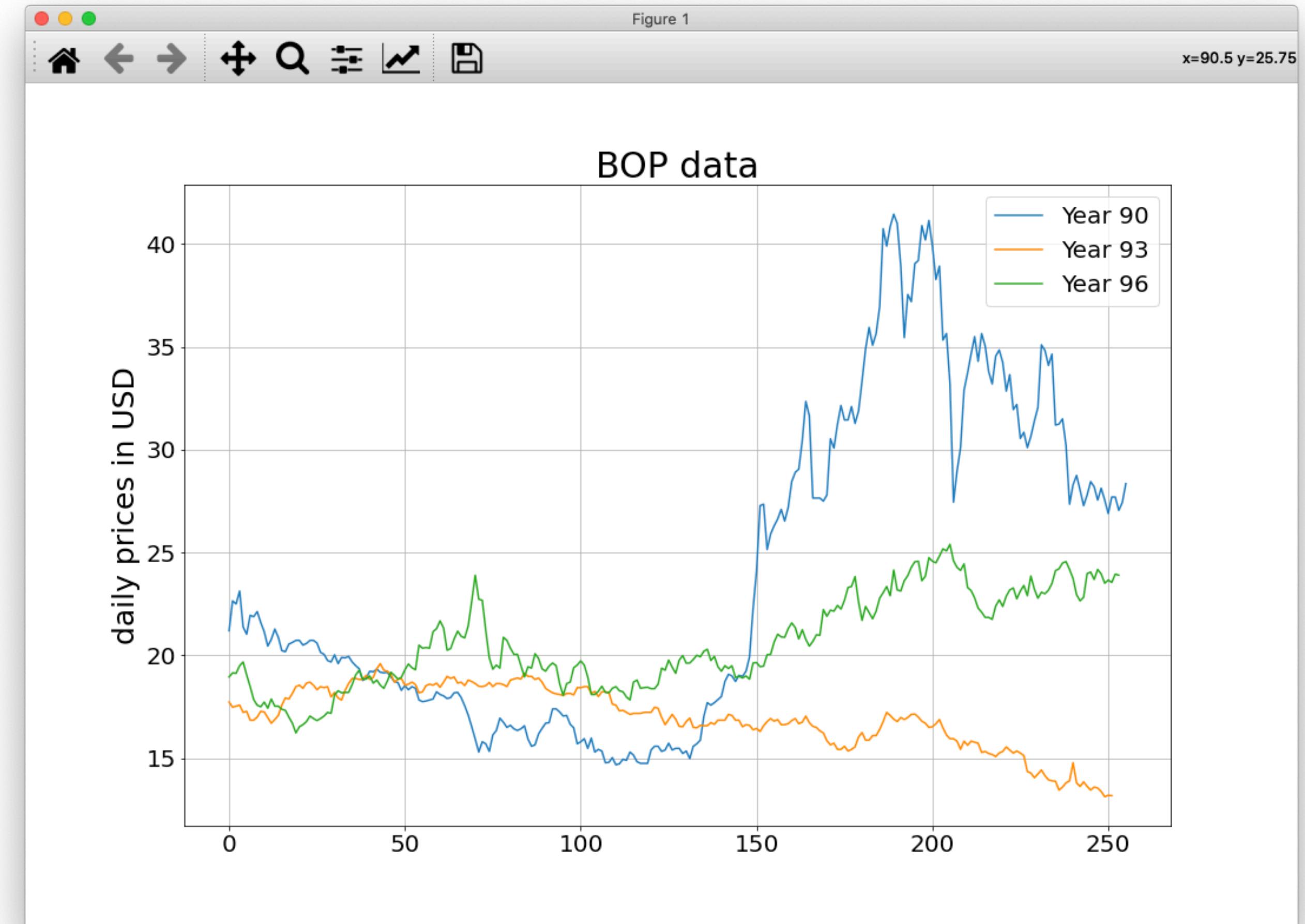
dataset = bop_data_reader()

fig, ax = plt.subplots(figsize=(15, 10))

bop_plot(dataset, 90, ax)
bop_plot(dataset, 93, ax)
bop_plot(dataset, 96, ax)

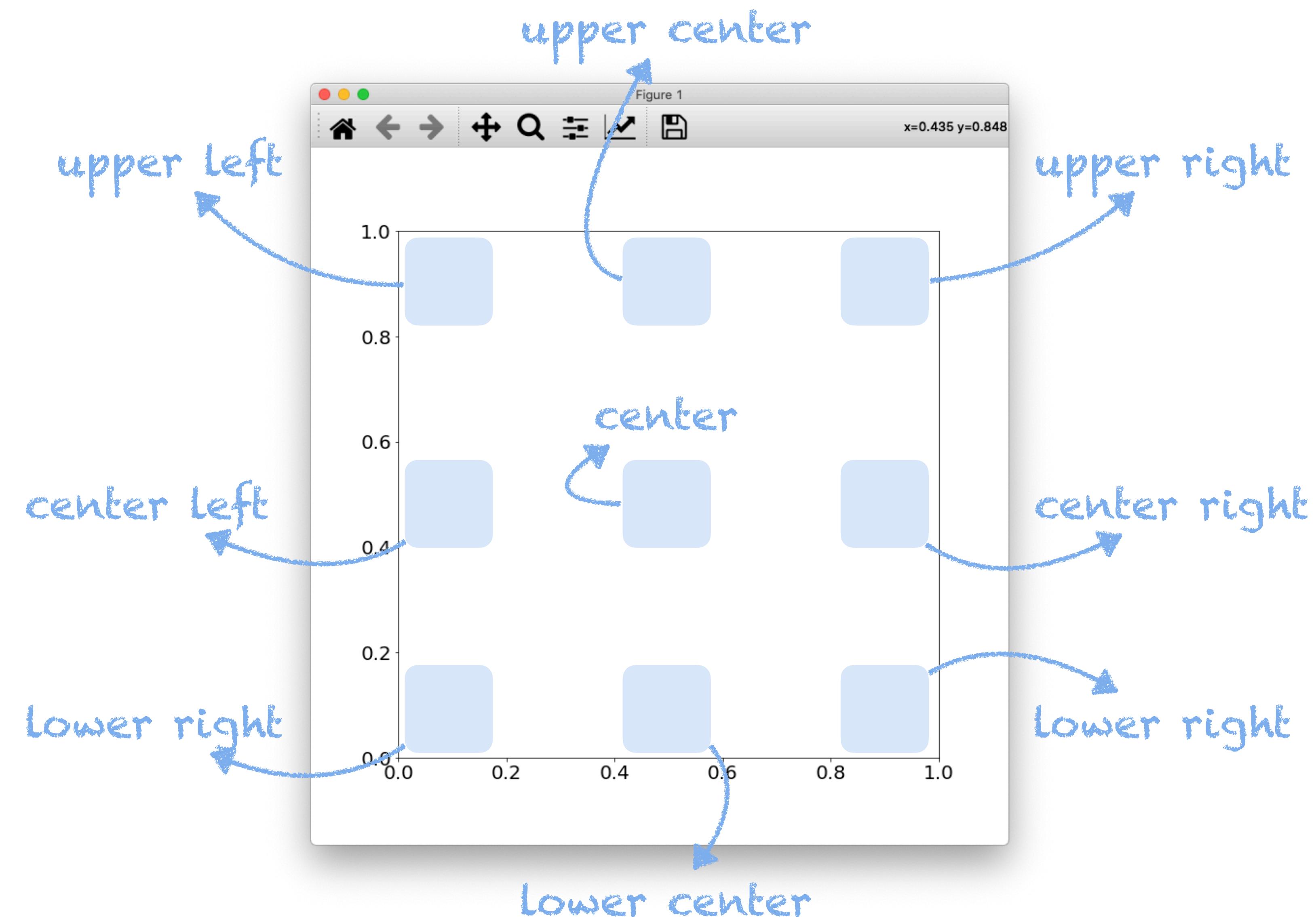
ax.set_title("BOP data",
             fontsize=30)
ax.set_ylabel("daily prices in USD",
              fontsize=25)
ax.tick_params(labelsize=20)
ax.grid()

ax.legend(fontsize=20)
```



2. Legend Locations

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10



2. Legend Locations

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

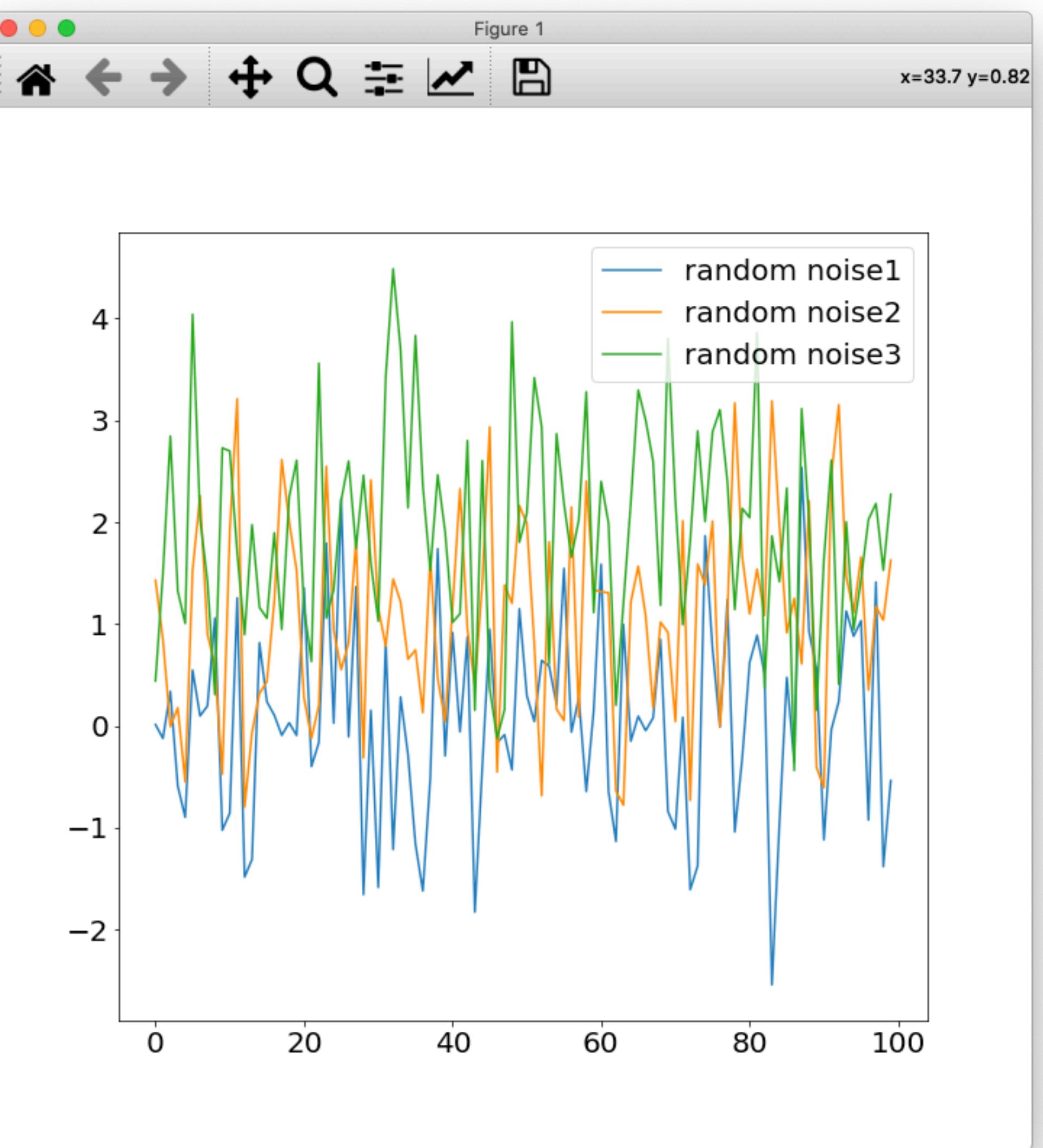
n_data = 100
random_noisel = np.random.normal(0, 1, (n_data,))
random_noise2 = np.random.normal(1, 1, (n_data,))
random_noise3 = np.random.normal(2, 1, (n_data,))

fig, ax = plt.subplots(figsize=(10, 10))

ax.plot(random_noisel,
         label='random noise1')
ax.plot(random_noise2,
         label='random noise2')
ax.plot(random_noise3,
         label='random noise3')

ax.tick_params(labelsize=20)

ax.legend(fontsize=20,
          loc='upper right')
```

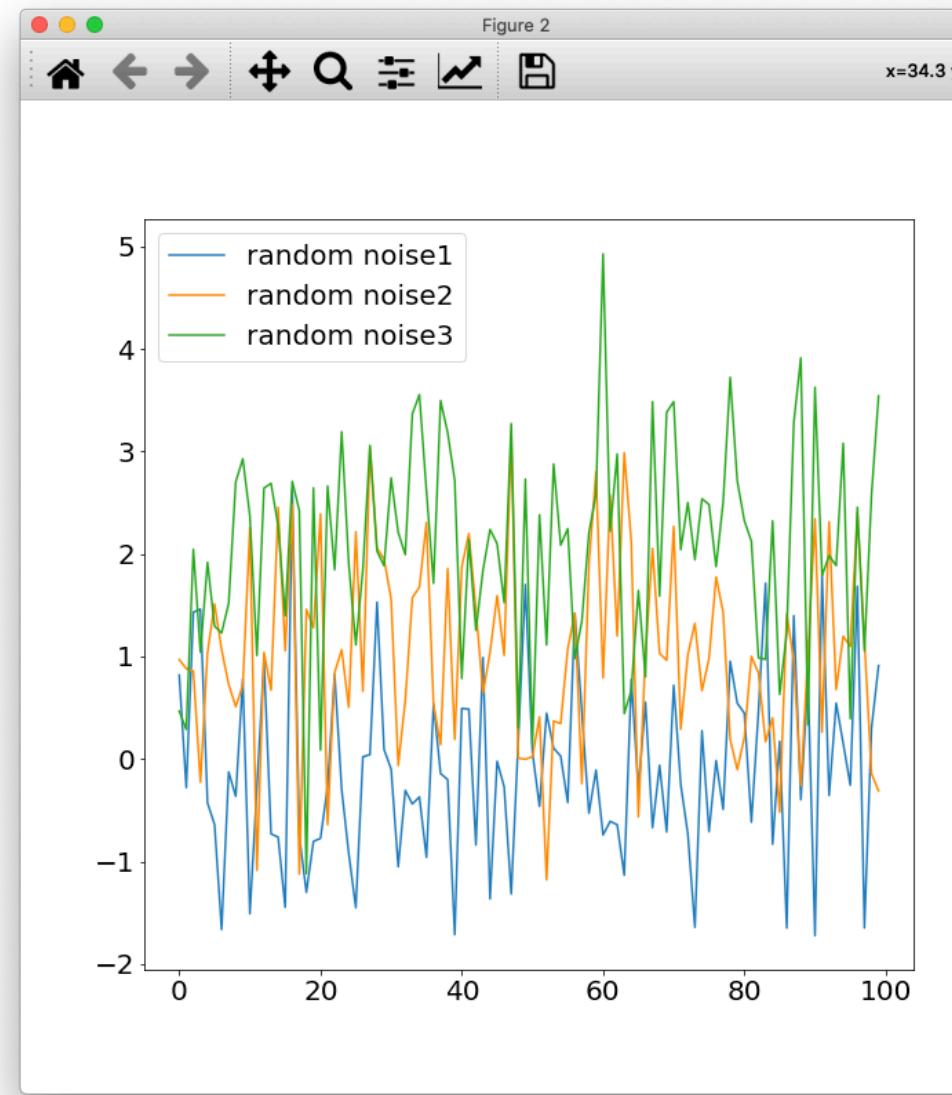


Lecture. 2-02 Labels and Legend

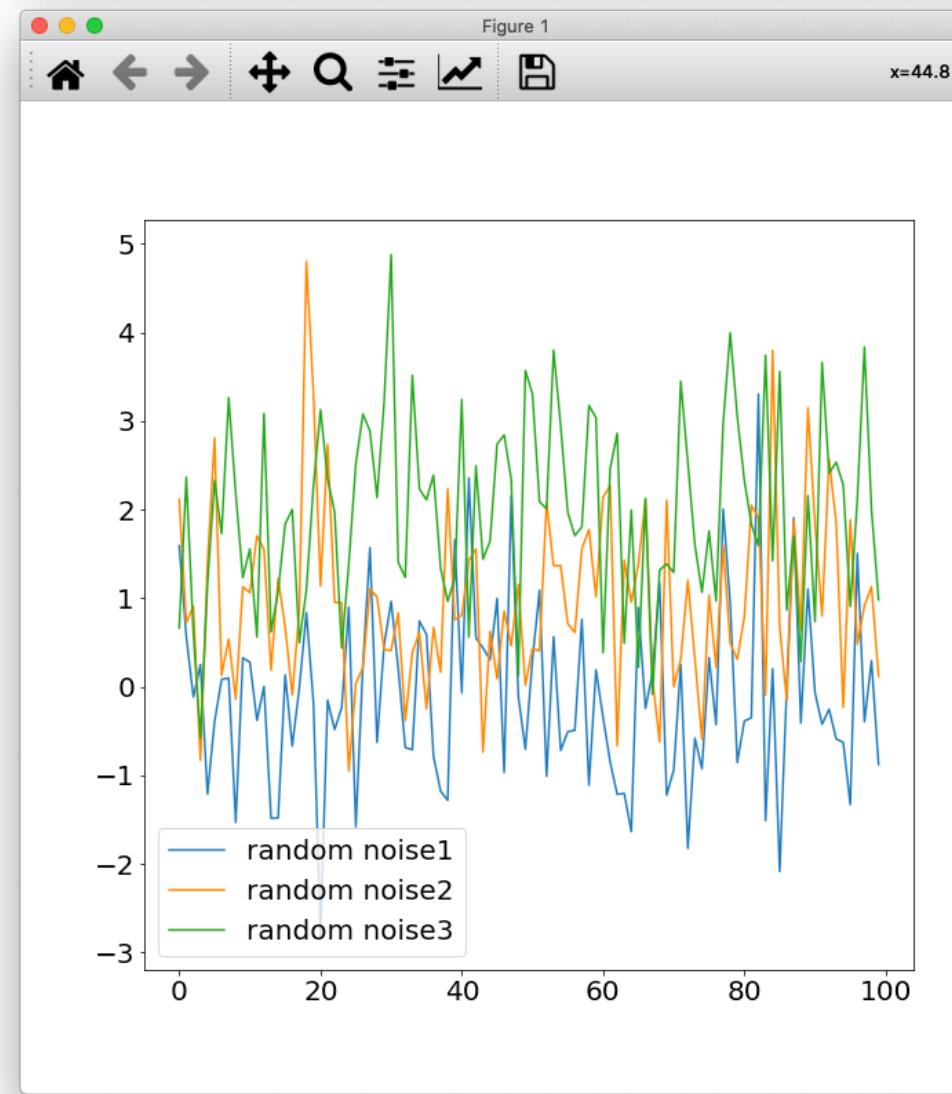
9

2. Legend Locations

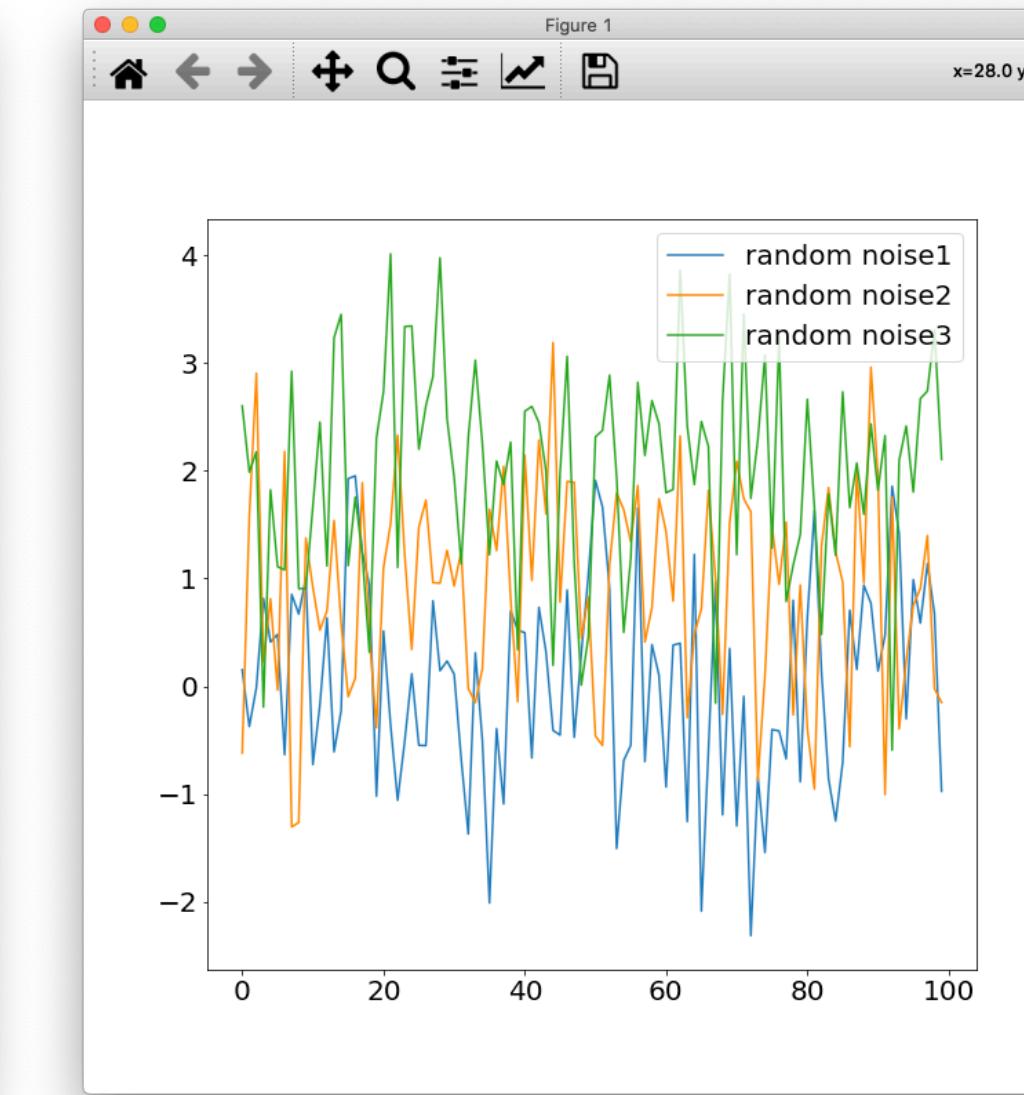
```
ax.legend(fontsize=20,  
         loc='upper left')
```



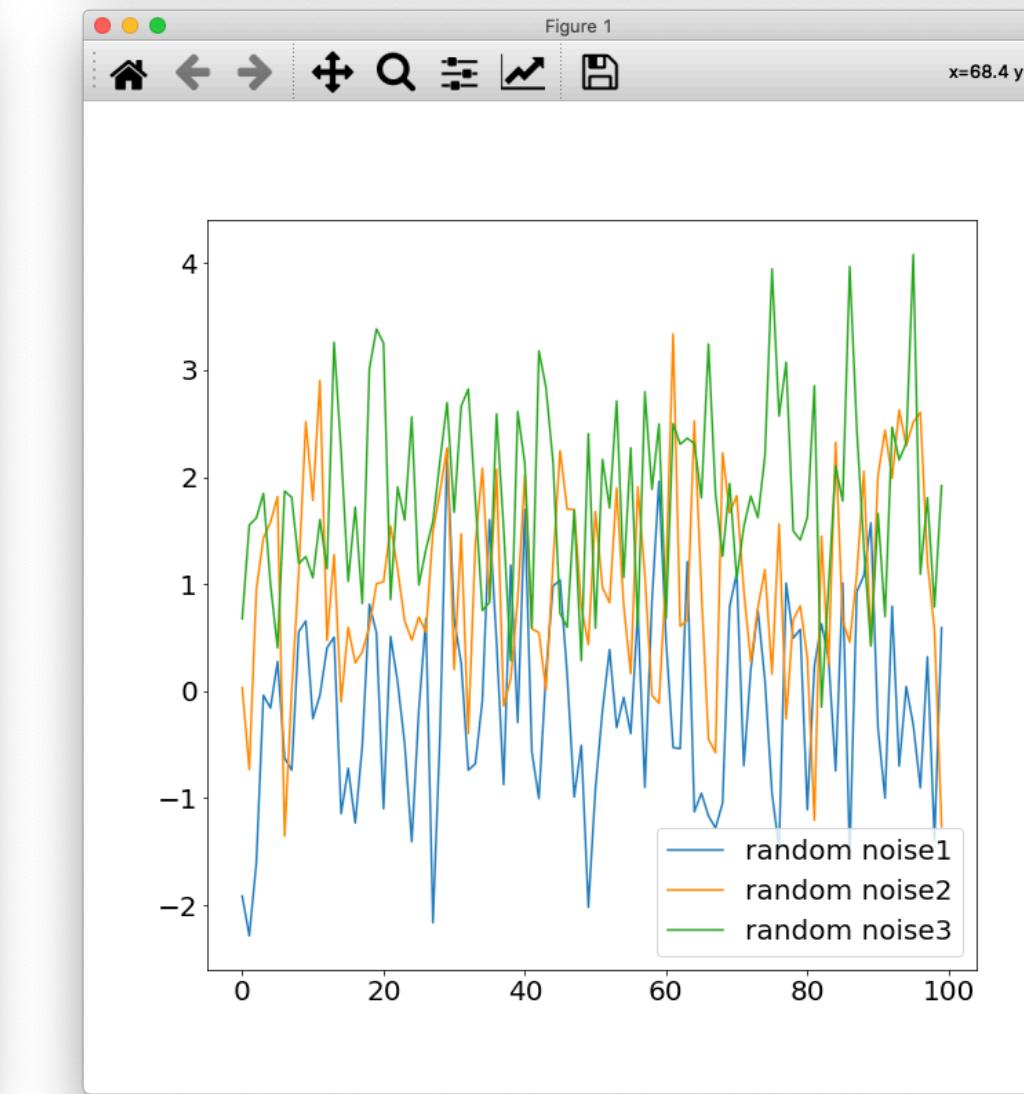
```
ax.legend(fontsize=20,  
         loc='lower left')
```



```
ax.legend(fontsize=20,  
         loc='upper right')
```



```
ax.legend(fontsize=20,  
         loc='lower right')
```



3. ncol Argument

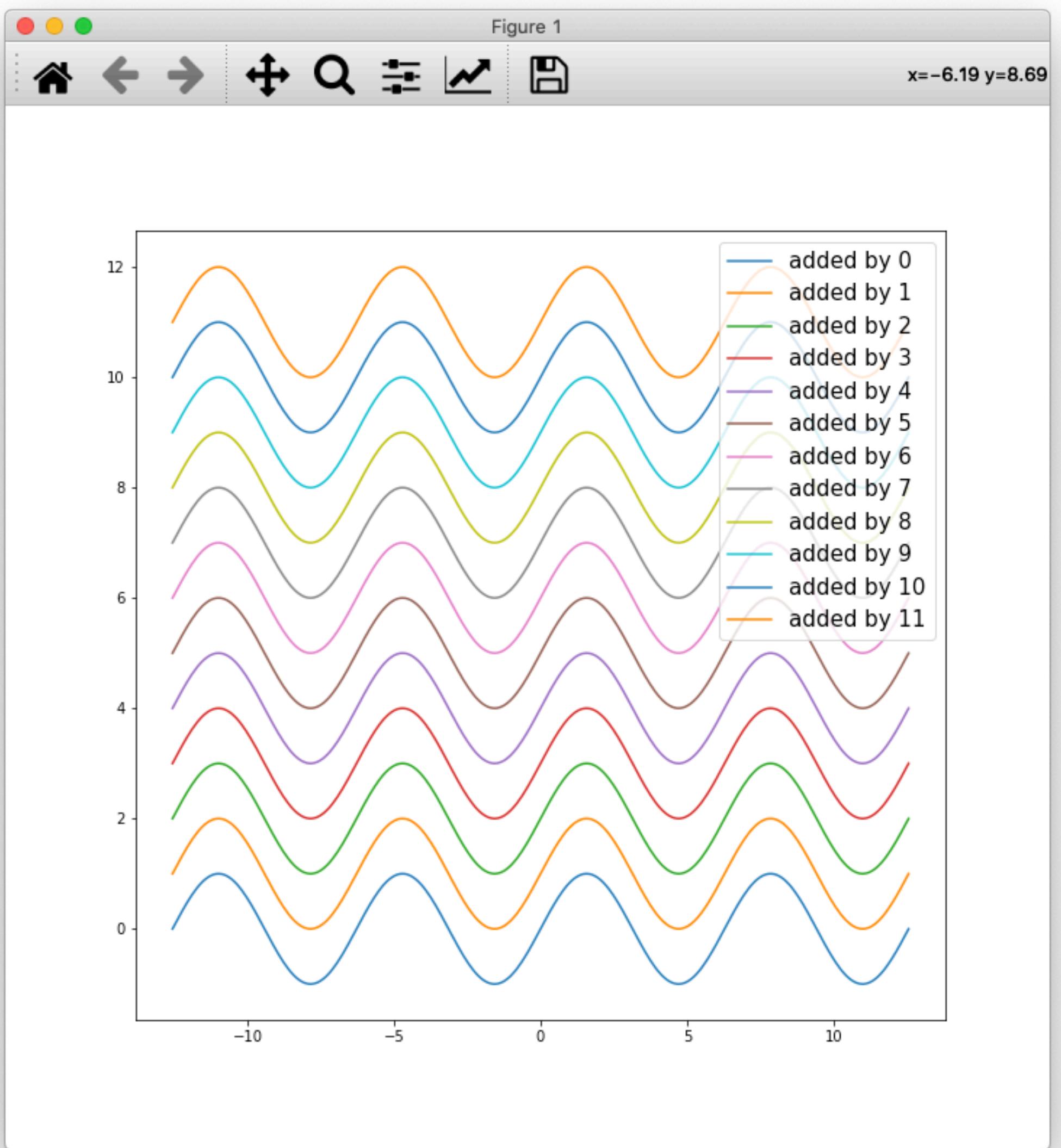
```
import matplotlib.pyplot as plt
import numpy as np

PI = np.pi
t = np.linspace(-4*PI, 4*PI, 300)
sin = np.sin(t)

fig, ax = plt.subplots(figsize=(10, 10))

for ax_idx in range(12):
    label_template = 'added by {}'
    ax.plot(t, sin+ax_idx,
             label=label_template.format(ax_idx))

ax.legend(fontsize=15)
```



3. ncol Argument

```

import matplotlib.pyplot as plt
import numpy as np

PI = np.pi
t = np.linspace(-4*PI, 4*PI, 300)
sin = np.sin(t)

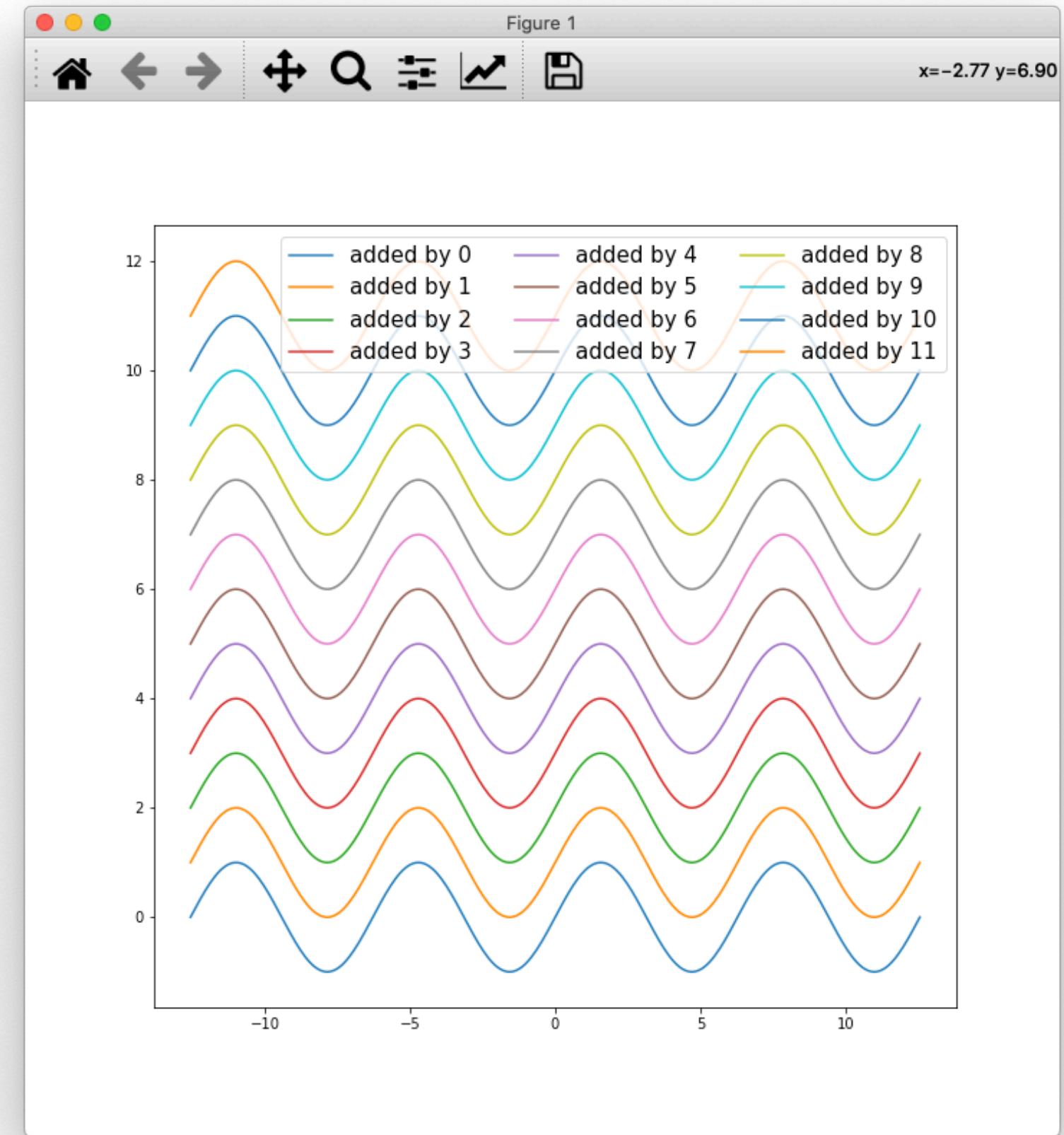
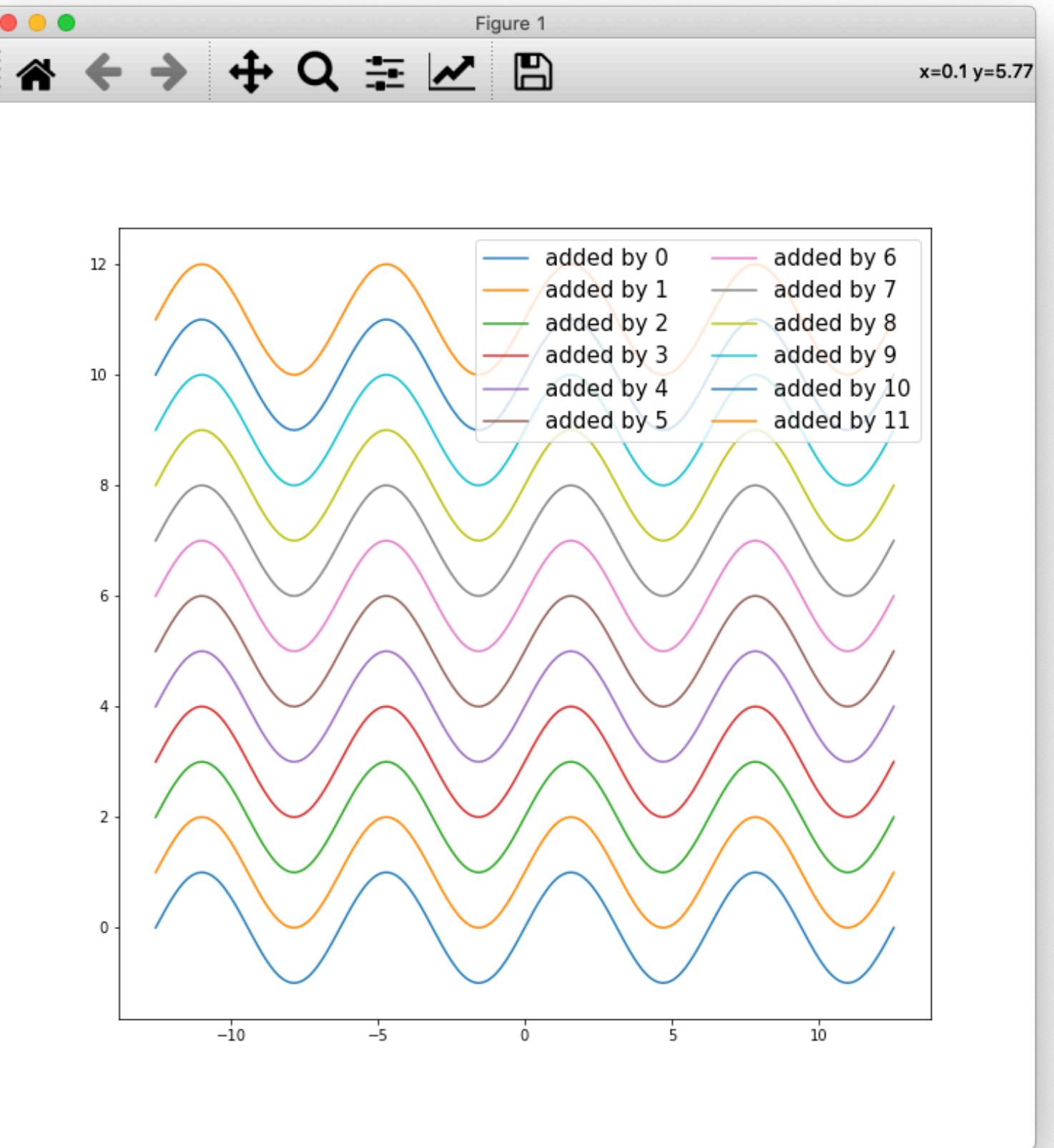
fig, ax = plt.subplots(figsize=(10, 10))

for ax_idx in range(12):
    label_template = 'added by {}'
    ax.plot(t, sin+ax_idx,
             label=label_template.format(ax_idx))

ax.legend(fontsize=15,
          ncol=2)

-----
ax.legend(fontsize=15,
          ncol=3)

```

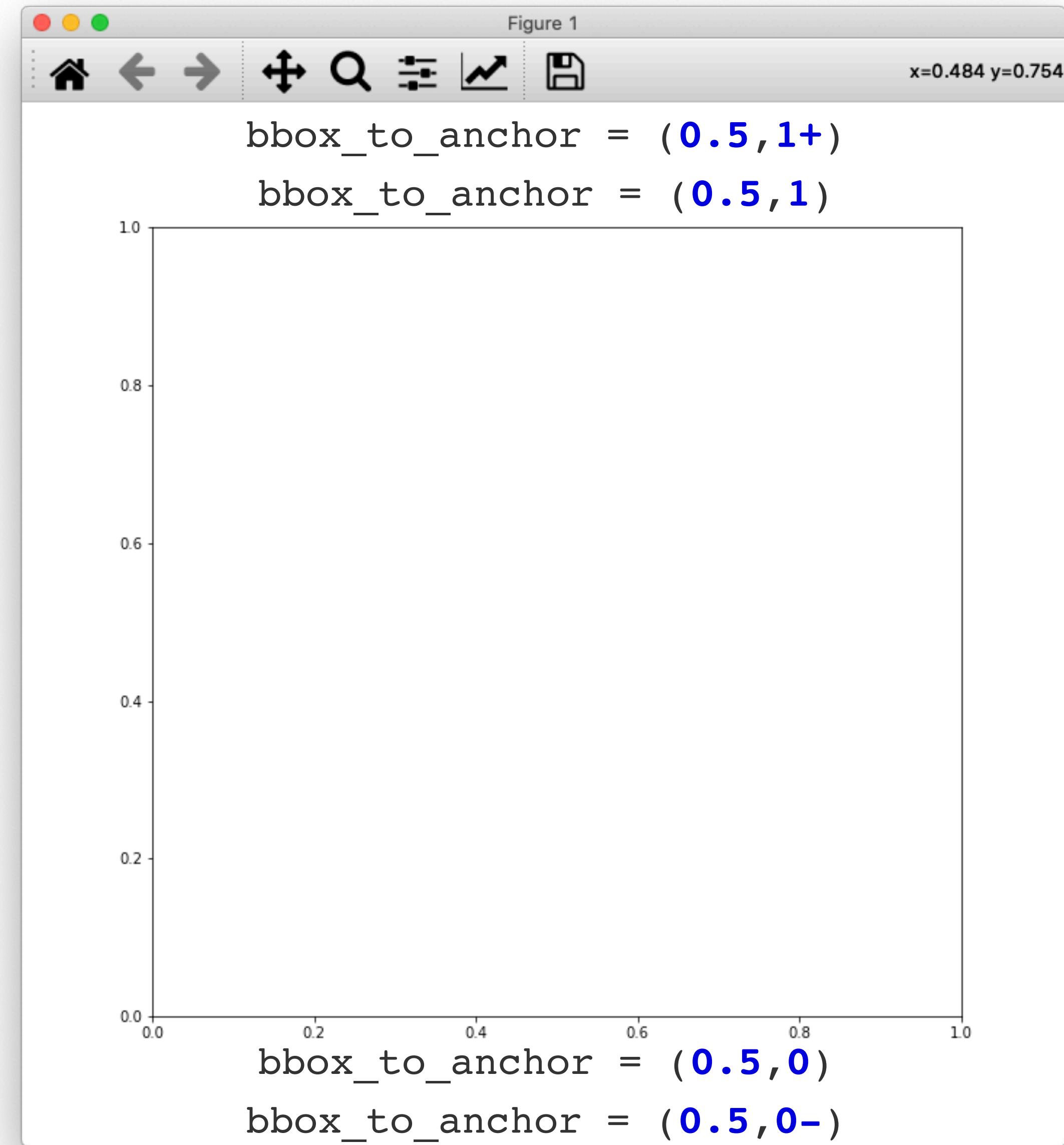


4. bbox_to_anchor Argument

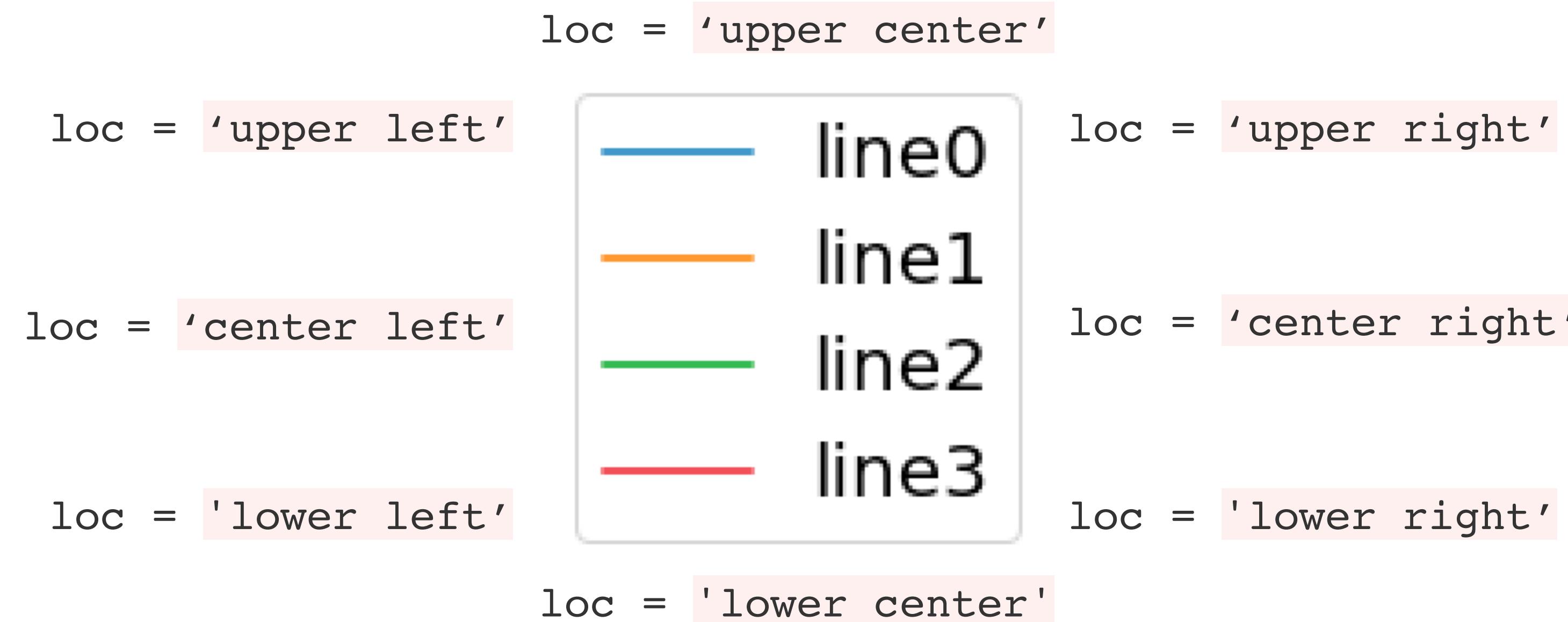
`bbox_to_anchor = (0,1)`

`bbox_to_anchor = (0,0.5)`

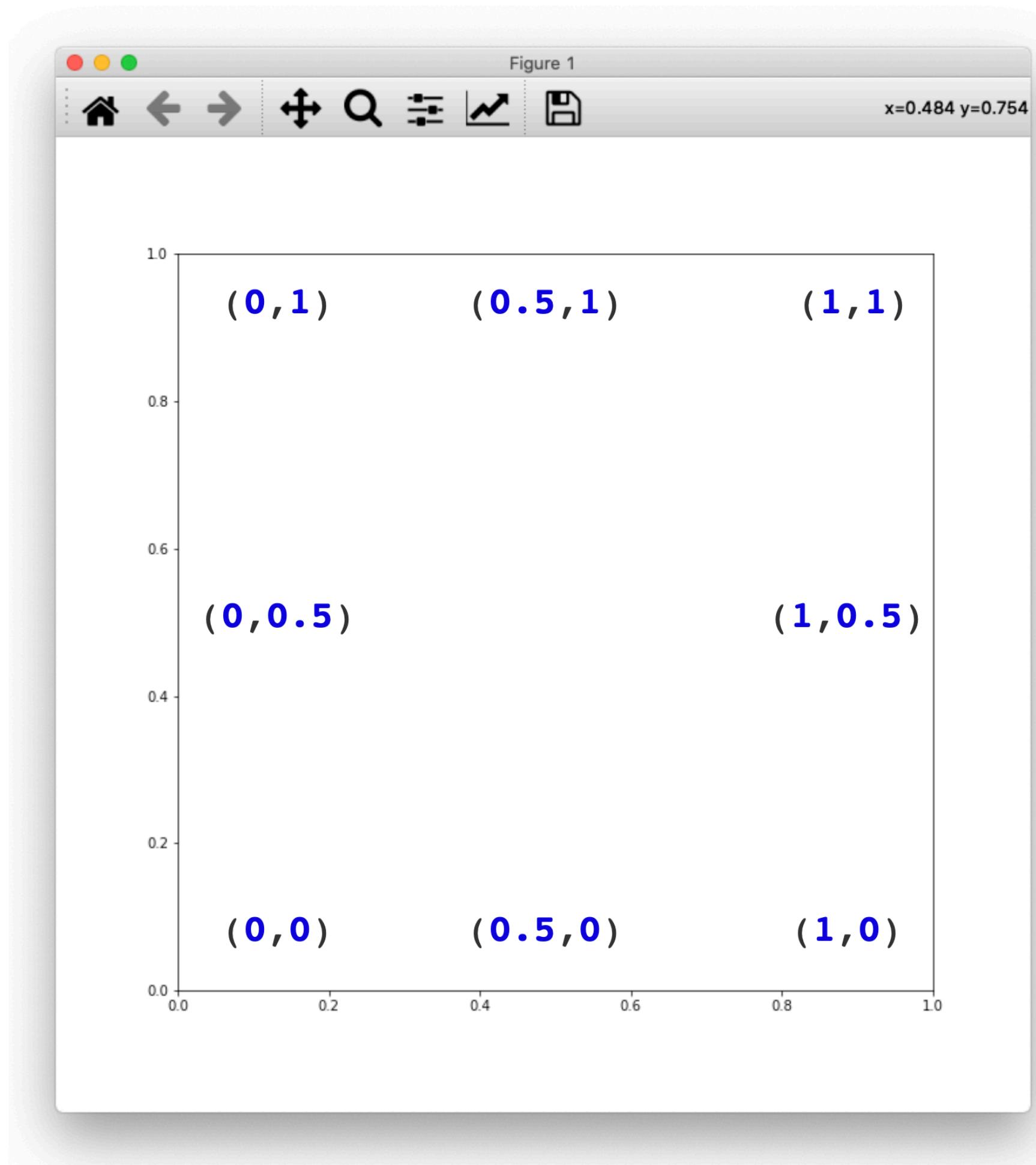
`bbox_to_anchor = (0,0)`



4. bbox_to_anchor Argument



4. bbox_to_anchor Argument



loc = 'upper center'
loc = 'upper left'
loc = 'center left'
loc = 'lower left'
loc = 'lower center'
loc = 'upper right'
loc = 'center right'
loc = 'lower right'

line0
line1
line2
line3

4. bbox_to_anchor Argument

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

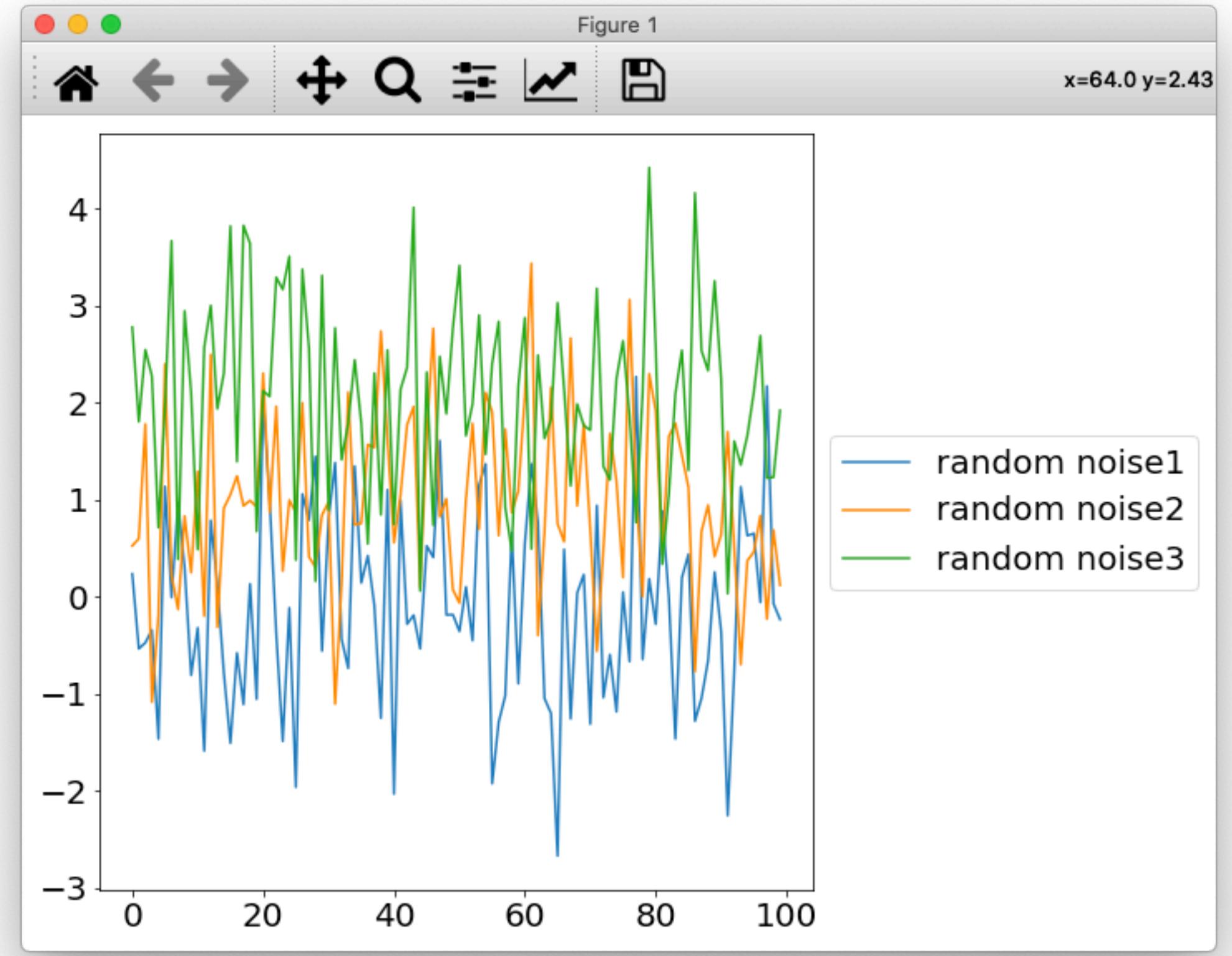
n_data = 100
random_noisel = np.random.normal(0, 1, (n_data,))
random_noise2 = np.random.normal(1, 1, (n_data,))
random_noise3 = np.random.normal(2, 1, (n_data,))

fig, ax = plt.subplots(figsize=(10, 7))
ax.tick_params(labelsize=20)

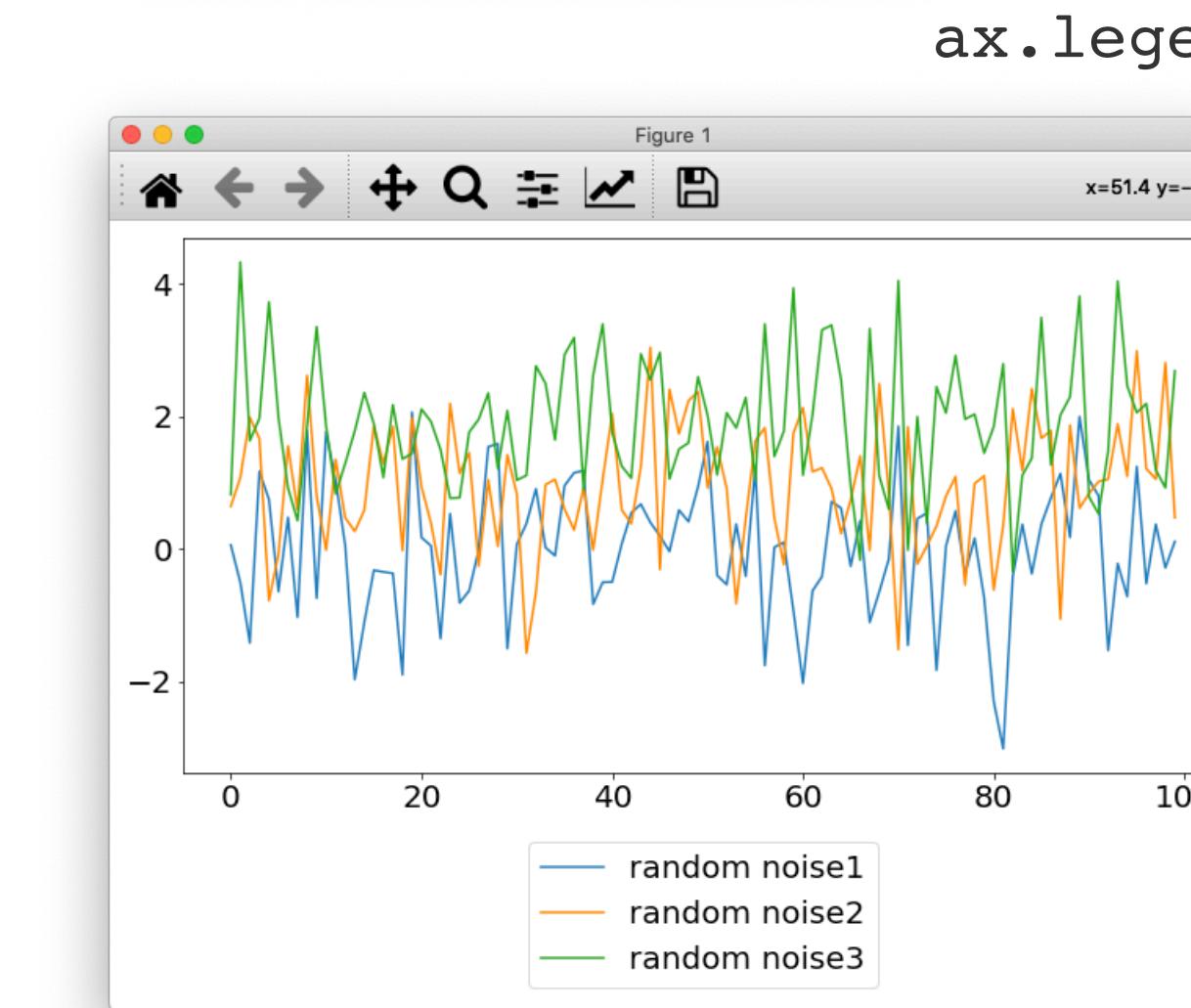
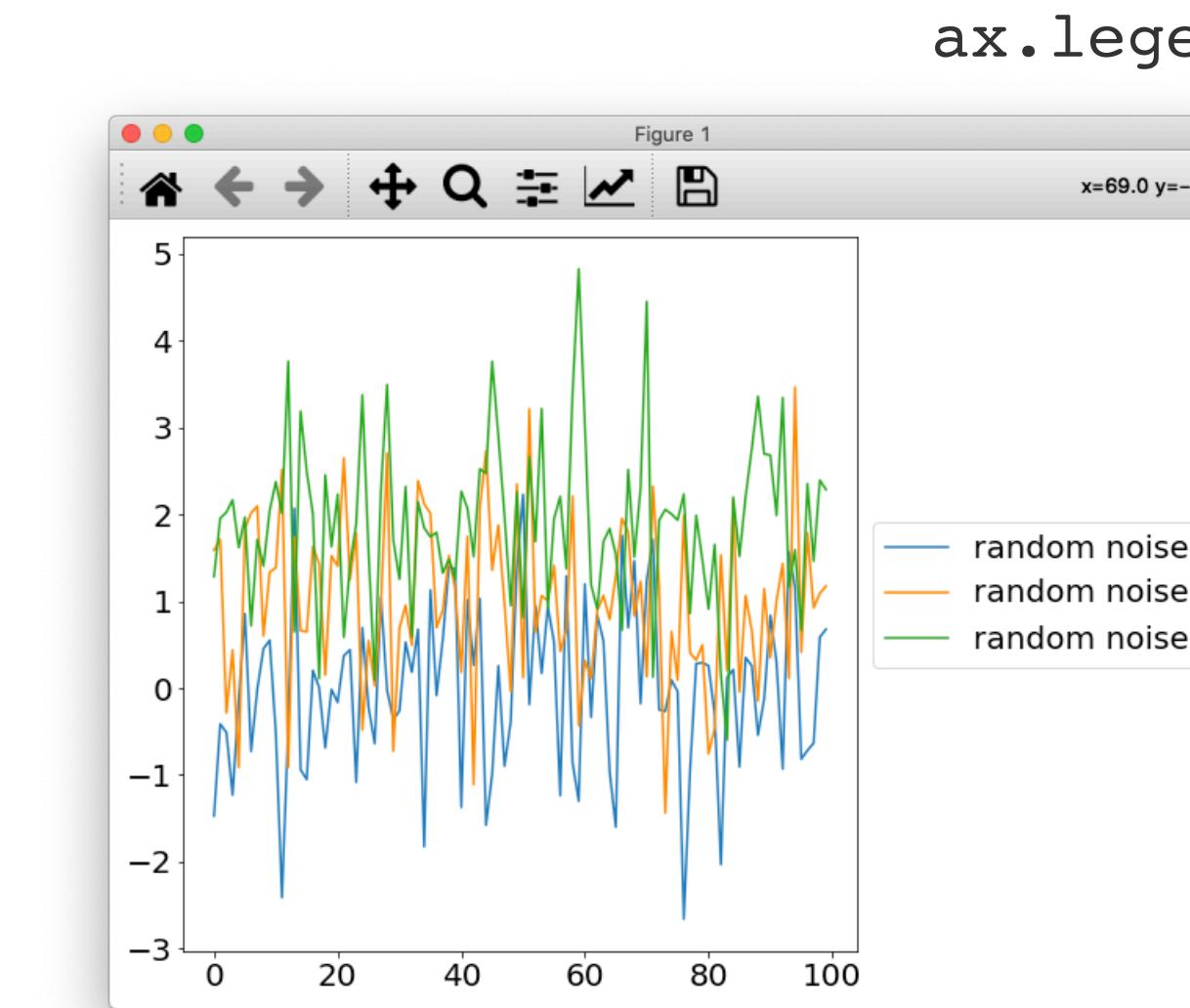
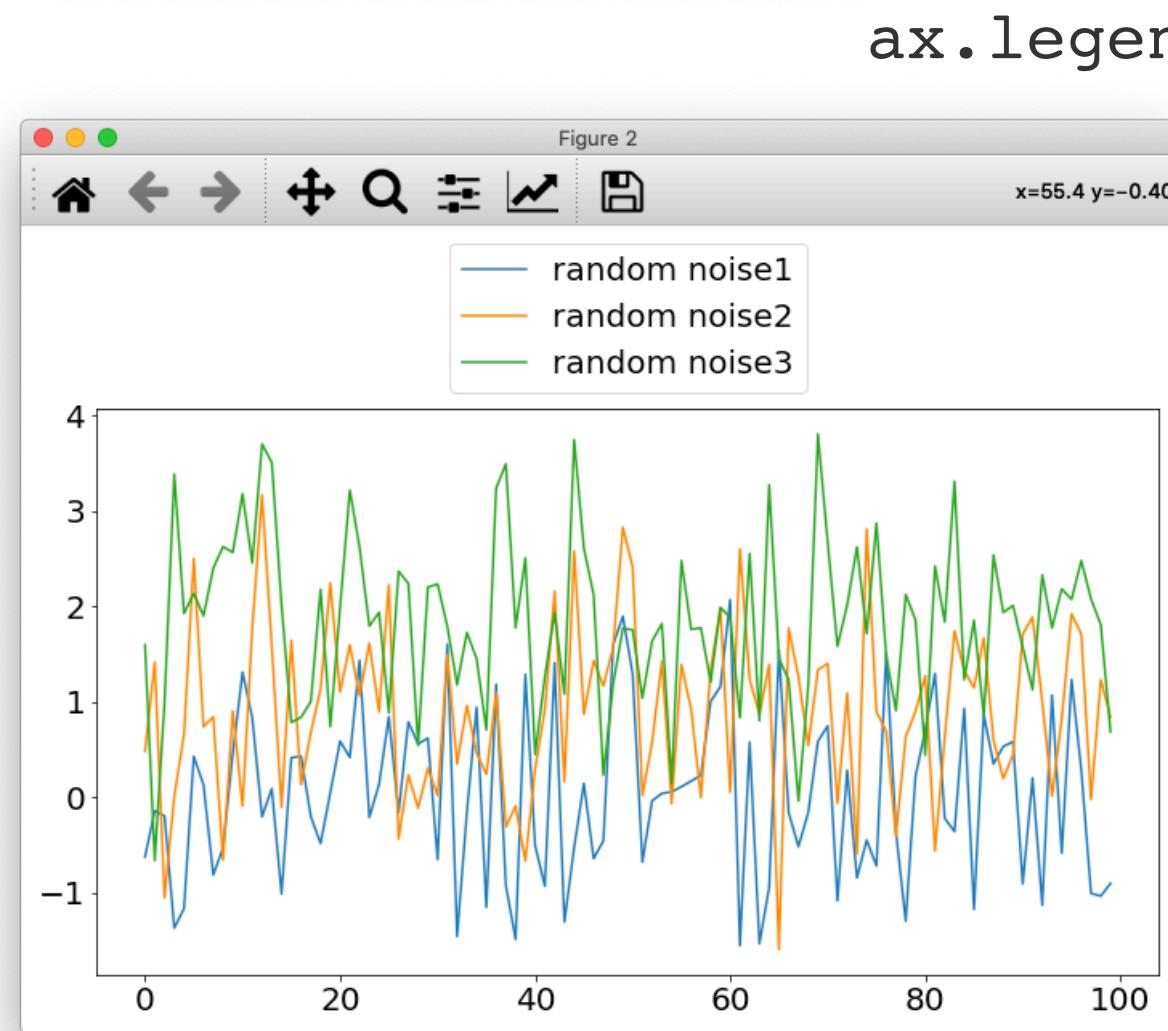
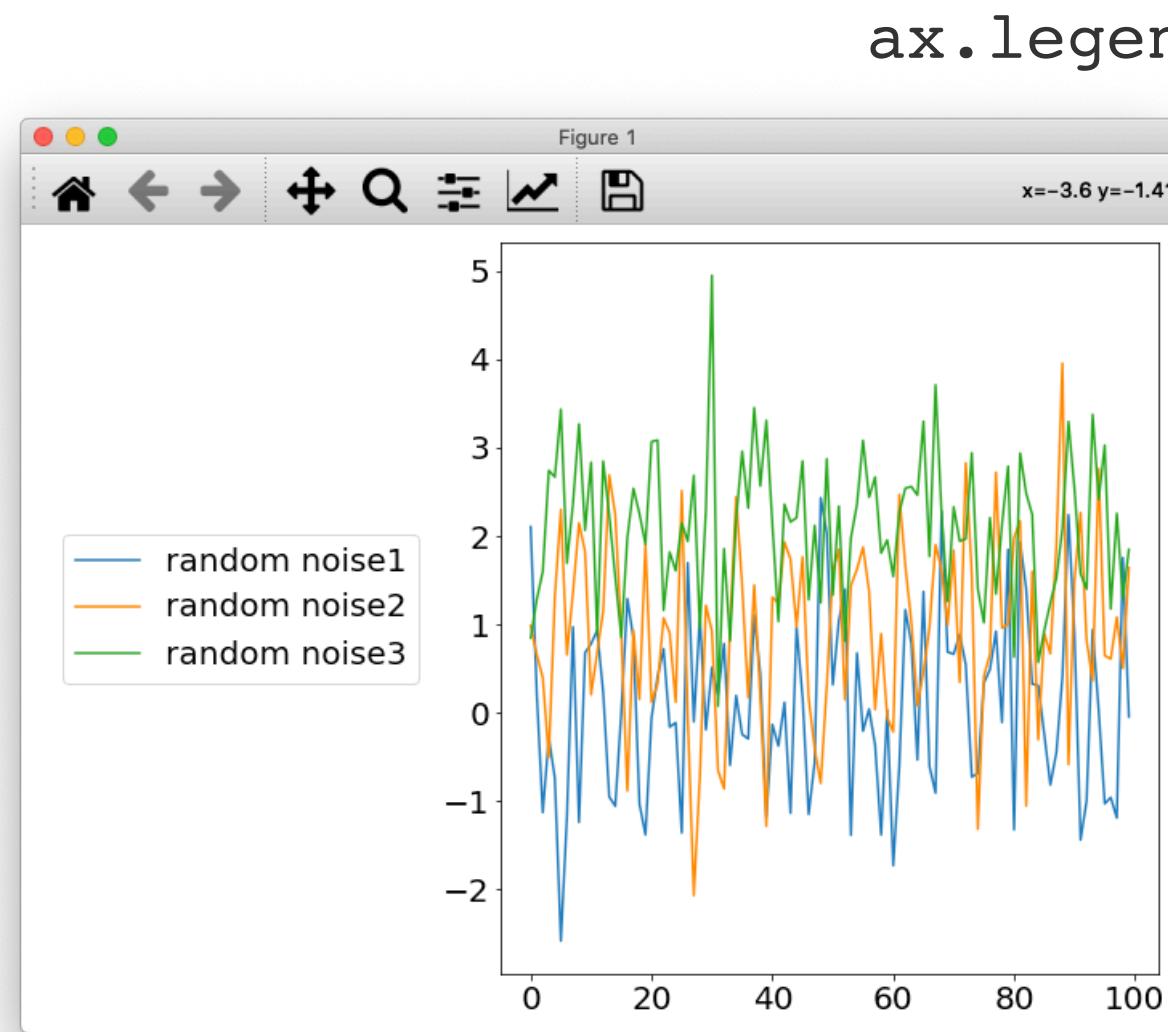
ax.plot(random_noisel,
         label='random noisel')
ax.plot(random_noise2,
         label='random noise2')
ax.plot(random_noise3,
         label='random noise3')

ax.legend(fontsize=20,
          bbox_to_anchor=(1, 0.5),
          loc='center left')

fig.tight_layout()
```



4. bbox_to_anchor Argument



4. bbox_to_anchor Argument

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

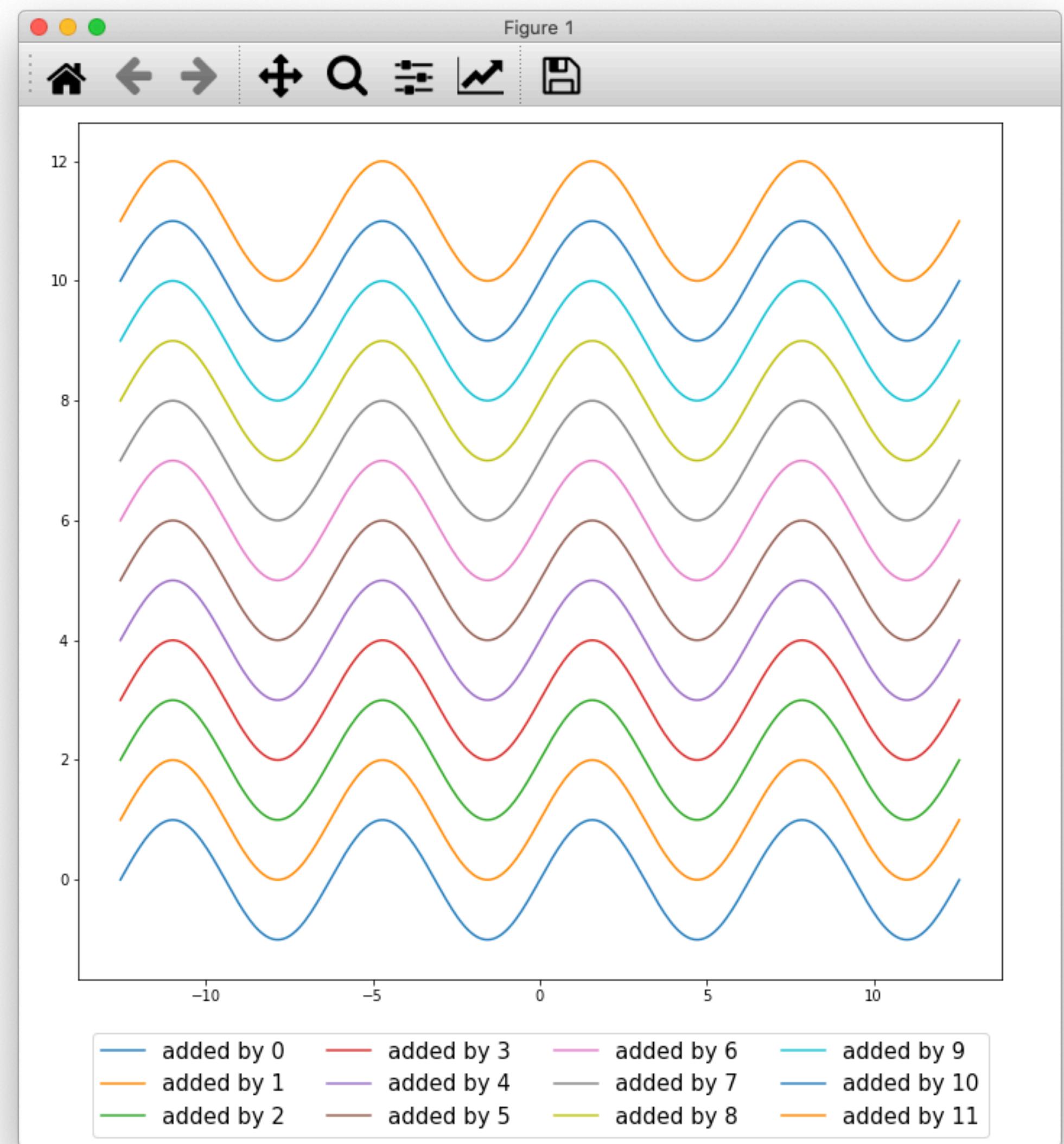
PI = np.pi
t = np.linspace(-4*PI, 4*PI, 300)
sin = np.sin(t)

fig, ax = plt.subplots(figsize=(10, 10))

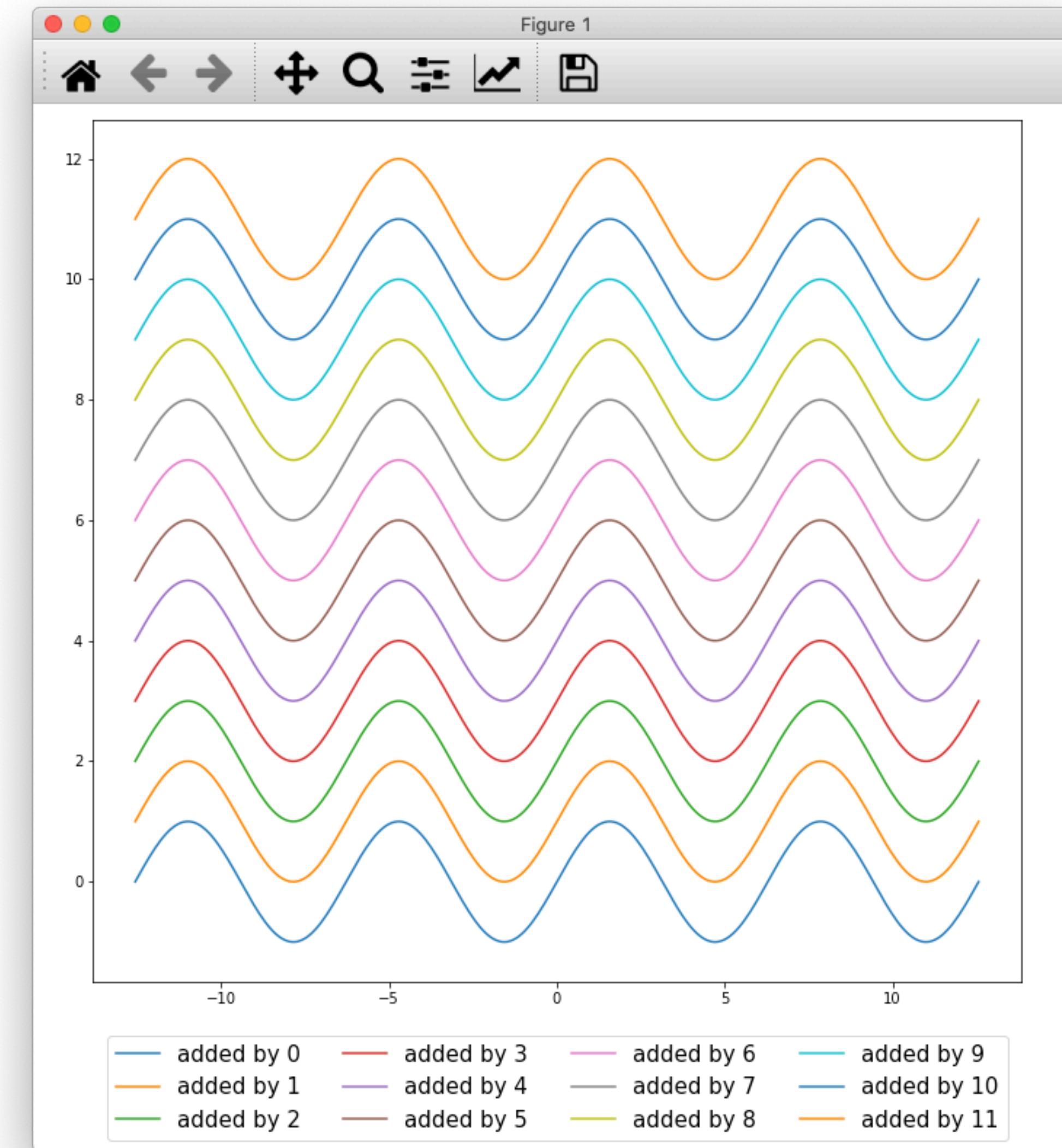
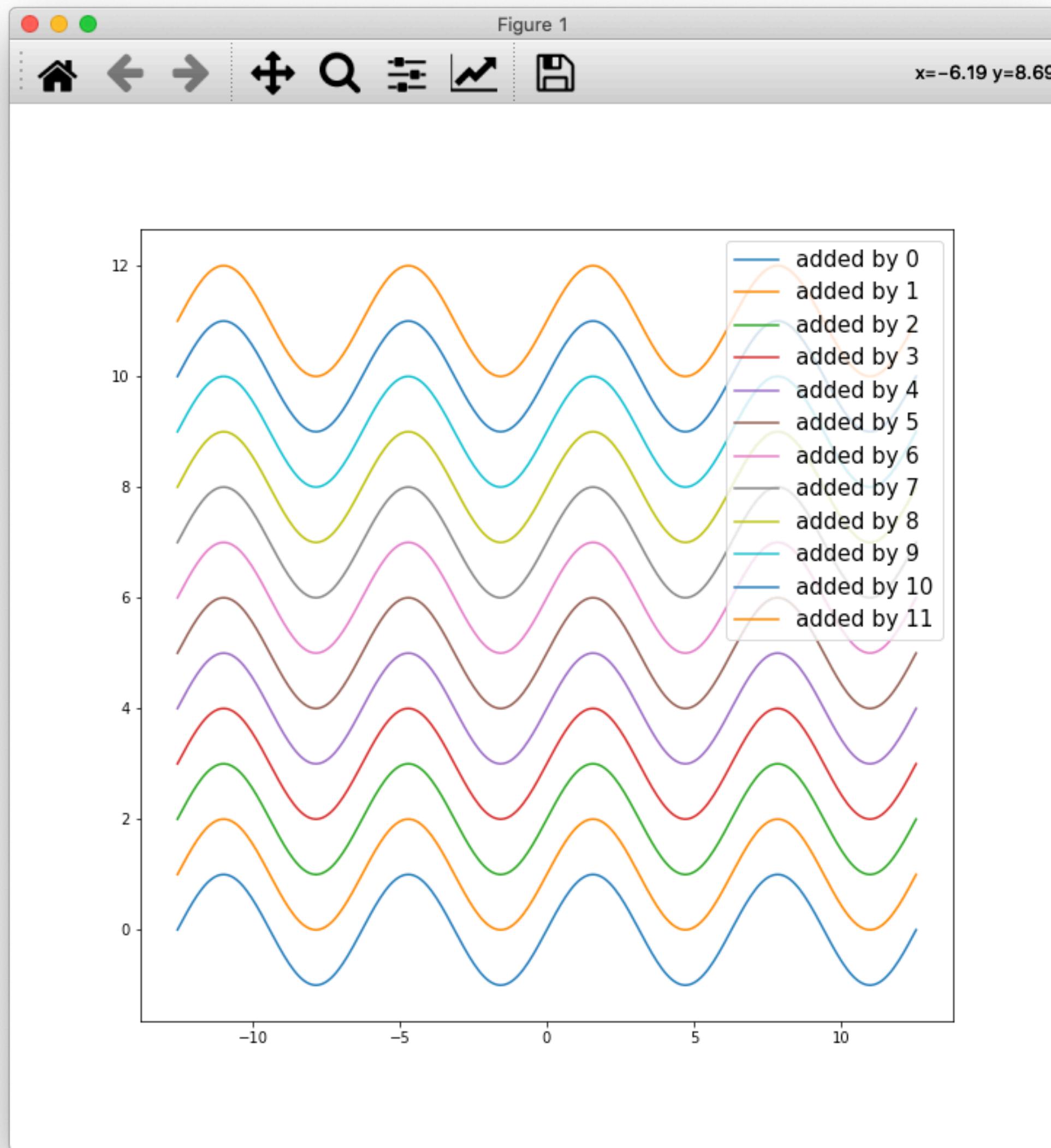
for ax_idx in range(12):
    label_template = 'added by {}'
    ax.plot(t, sin+ax_idx,
             label=label_template.format(ax_idx))

ax.legend(fontsize=15,
          ncol=4,
          bbox_to_anchor=(0.5, -0.05),
          loc='upper center')

fig.tight_layout()
```



4. bbox_to_anchor Argument



5. Labels on BOP data

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

from bop_utils import *

dataset = bop_data_reader()
t_year_list = [90, 92, 94, 96, 98]

fig, ax = plt.subplots(figsize=(14, 10))

for t_year_idx, t_year in enumerate(t_year_list):
    t_data = get_year_data(dataset, t_year)

    ax.plot(t_data[:, -1],
            label='Year ' + str(t_year))

ax.legend(loc='upper left',
          fontsize=20)

major_yticks = np.arange(10, 41, 10)
minor_yticks = np.arange(10, 43, 2)
ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)
ax.tick_params(axis='y',
               length=10,
               labelsize=20)
ax.grid(axis='y',
        which='major')
ax.grid(axis='y',
        which='minor',
        linestyle=':')

ax.set_ylabel("daily prices in USD",
              fontsize=25)
```

5. Labels on BOP data

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

from bop_utils import *

dataset = bop_data_reader()
t_year_list = [90, 92, 94, 96, 98]

fig, ax = plt.subplots(figsize=(14, 10))

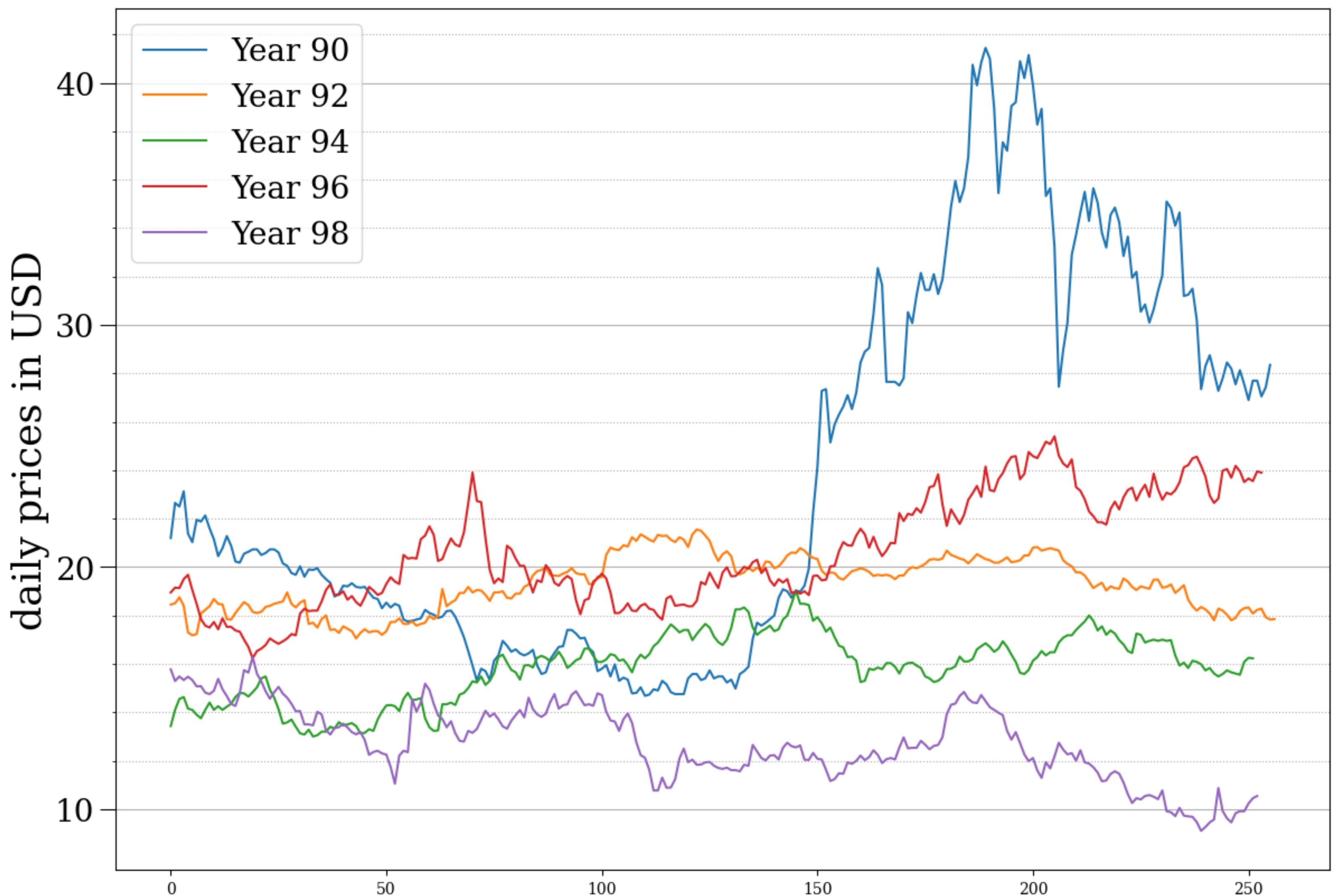
for t_year_idx, t_year in enumerate(t_year_list):
    t_data = get_year_data(dataset, t_year)

    ax.plot(t_data[:, -1],
            label='Year ' + str(t_year))

ax.legend(loc='upper left',
          fontsize=20)

major_yticks = np.arange(10, 41, 10)
minor_yticks = np.arange(10, 43, 2)
ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)
ax.tick_params(axis='y',
               length=10,
               labelsize=20)
ax.grid(axis='y',
        which='major')
ax.grid(axis='y',
        which='minor',
        linestyle=':')

ax.set_ylabel("daily prices in USD",
              fontsize=25)
```



5. Labels on BOP data

```

import matplotlib.pyplot as plt
import numpy as np

from bop_utils import *

dataset = bop_data_reader()
t_year_list = [90, 92, 94, 96, 98]

fig, ax = plt.subplots(figsize=(14, 10))

for t_year_idx, t_year in enumerate(t_year_list):
    t_data = get_year_data(dataset, t_year)

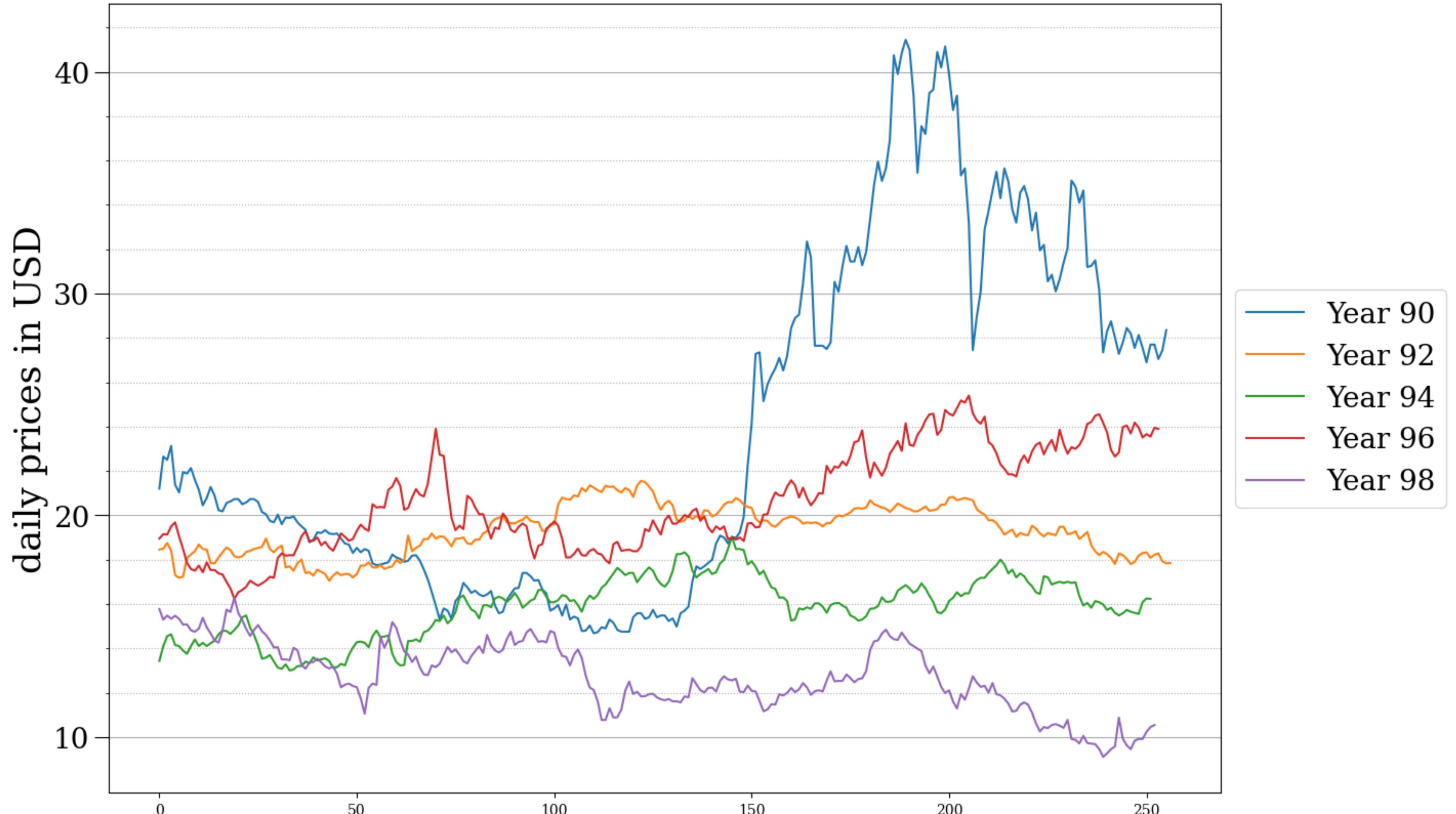
    ax.plot(t_data[:, -1],
            label='Year ' + str(t_year))

ax.legend(loc='center left',
          bbox_to_anchor=(1, 0.5),
          fontsize=20)

major_yticks = np.arange(10, 41, 10)
minor_yticks = np.arange(10, 43, 2)
ax.set_yticks(major_yticks)
ax.set_yticks(minor_yticks,
              minor=True)
ax.tick_params(axis='y',
               length=10,
               labelsize=20)
ax.grid(axis='y',
        which='major')
ax.grid(axis='y',
        which='minor',
        linestyle=':')

ax.set_ylabel("daily prices in USD",
              fontsize=25)

```



Python for Data Visualization

-Chapter.2 Line Plot -

2-02. Labels and Legend

1. Basic Usage
2. Legend Locations
3. ncol Argument
4. bbox_to_anchor Argument
5. Labels on BOP data