

- 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 -

2-01. Line Plot Basics

1. `ax.plot(y)`
2. `ax.plot(x, y)`
3. **Several Line Plots on One Ax**
4. **Several Line Plots on Different Axes**
5. `ax.axvline` and `ax.axhline`

Lecture. 2-01 Line Plot Basics

4

1. ax.plot(y)

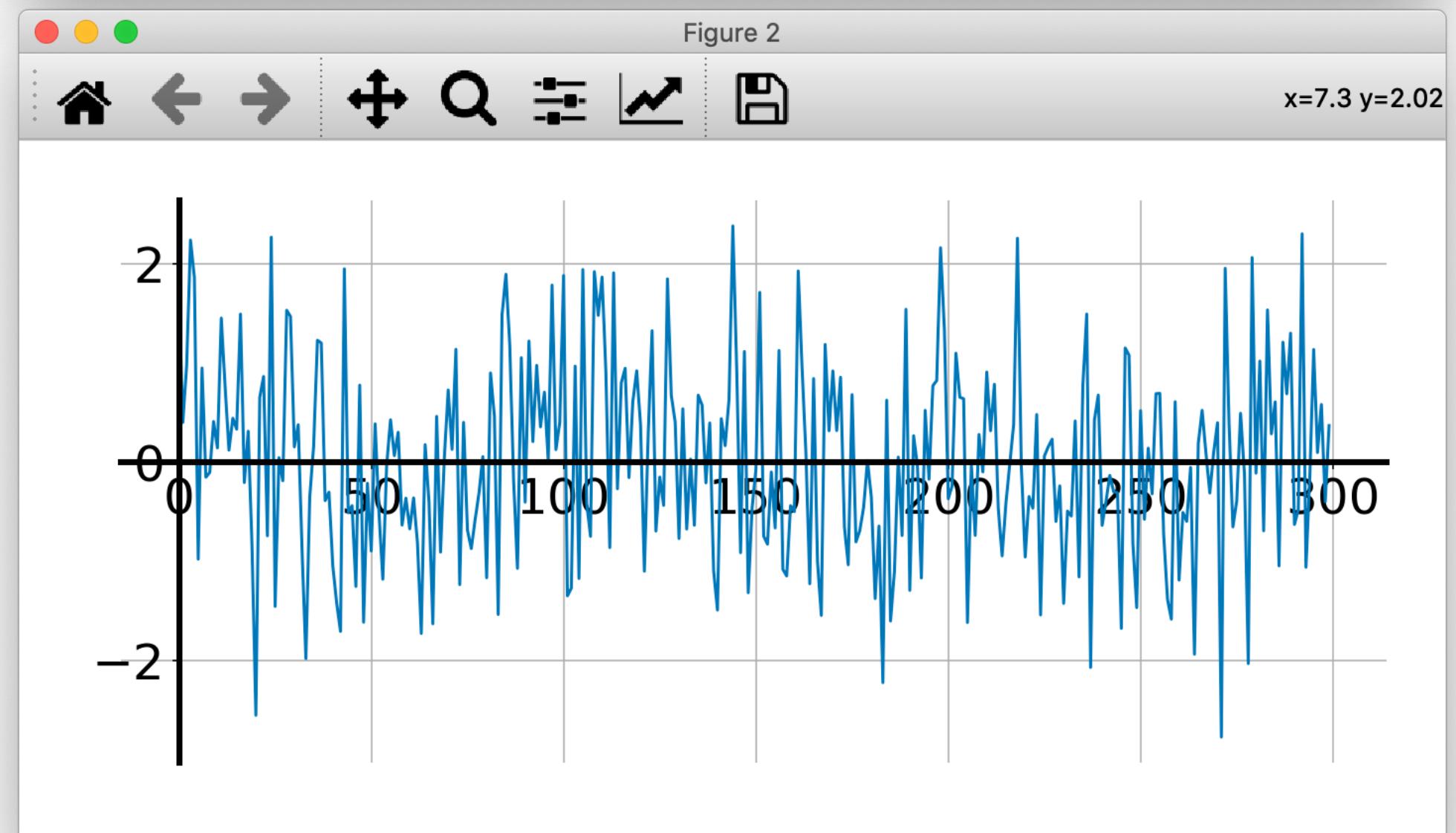
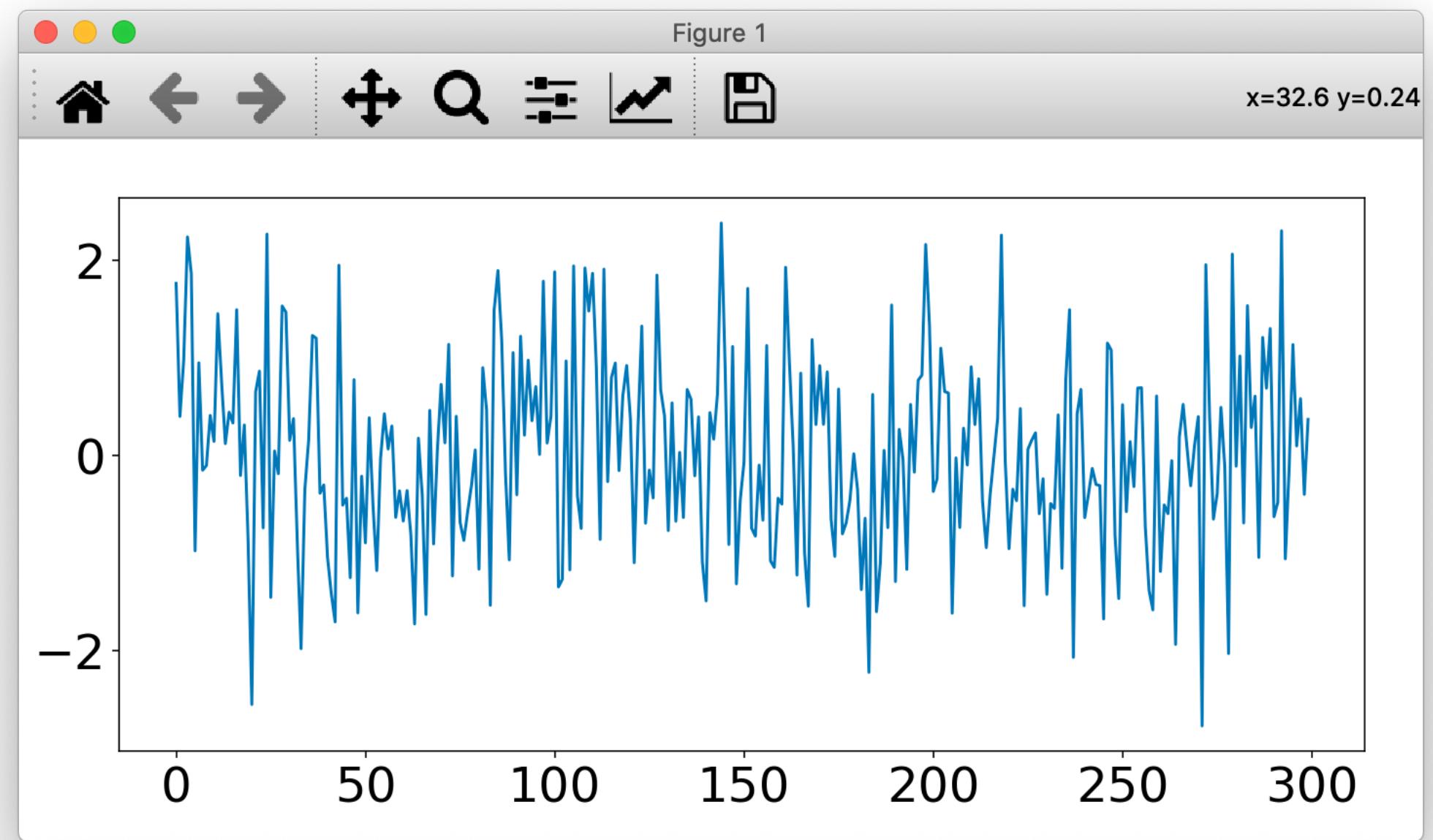
```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(0)

y_data = np.random.normal(loc=0, scale=1, size=(300,))

fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(y_data)

fig.tight_layout(pad=3)
ax.tick_params(labelsize=25)
-----
ax.grid()
for spine_loc, spine in ax.spines.items():
    spine.set_linewidth(3)
    if spine_loc in ['right', 'top']:
        spine.set_visible(False)
    if spine_loc in ['bottom', 'left']:
        spine.set_position(('data', 0))
```



Lecture. 2-01 Line Plot Basics

5

1. ax.plot(y)

```
np.random.seed(0)

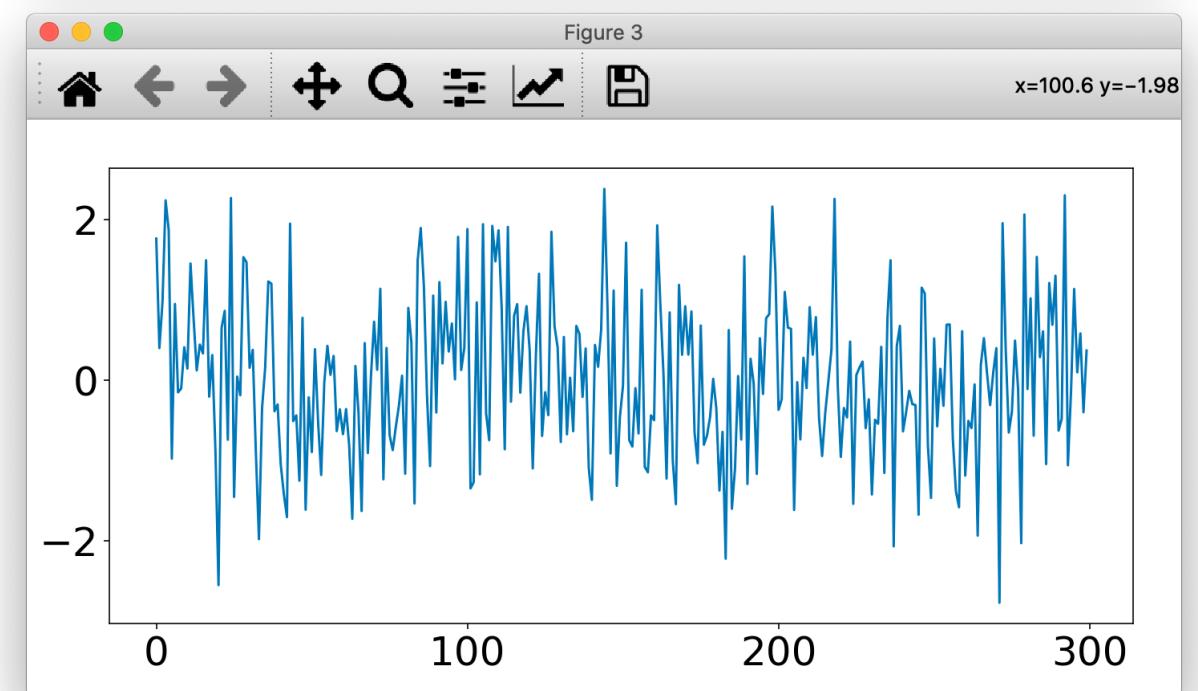
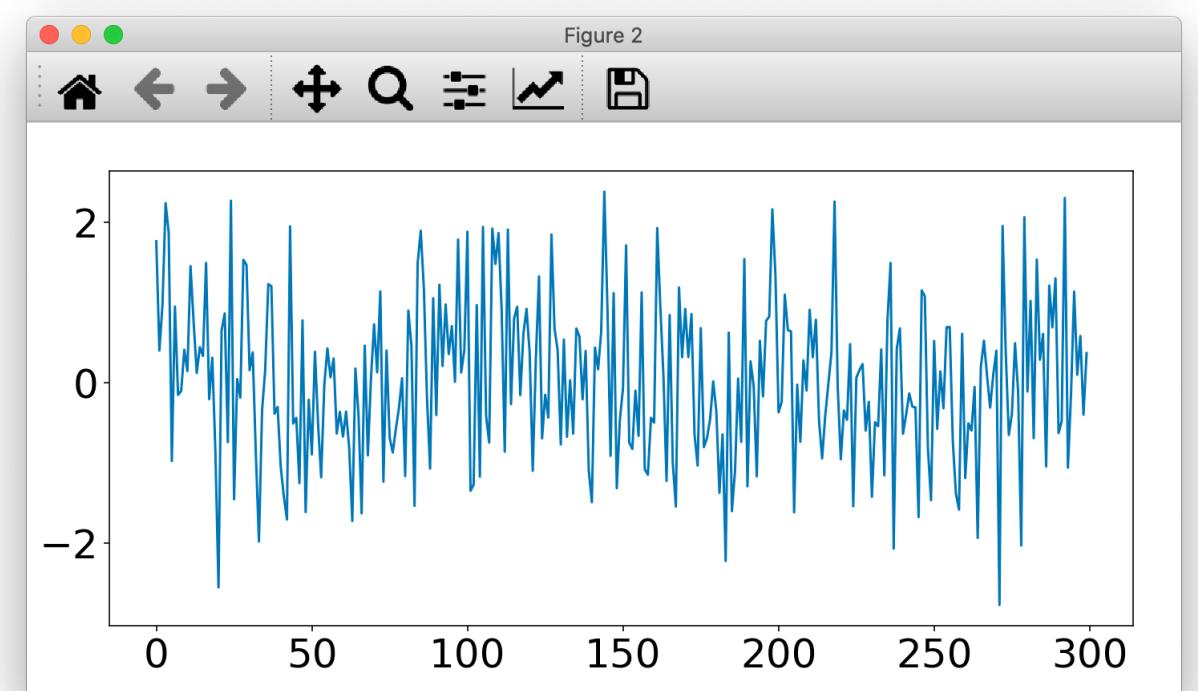
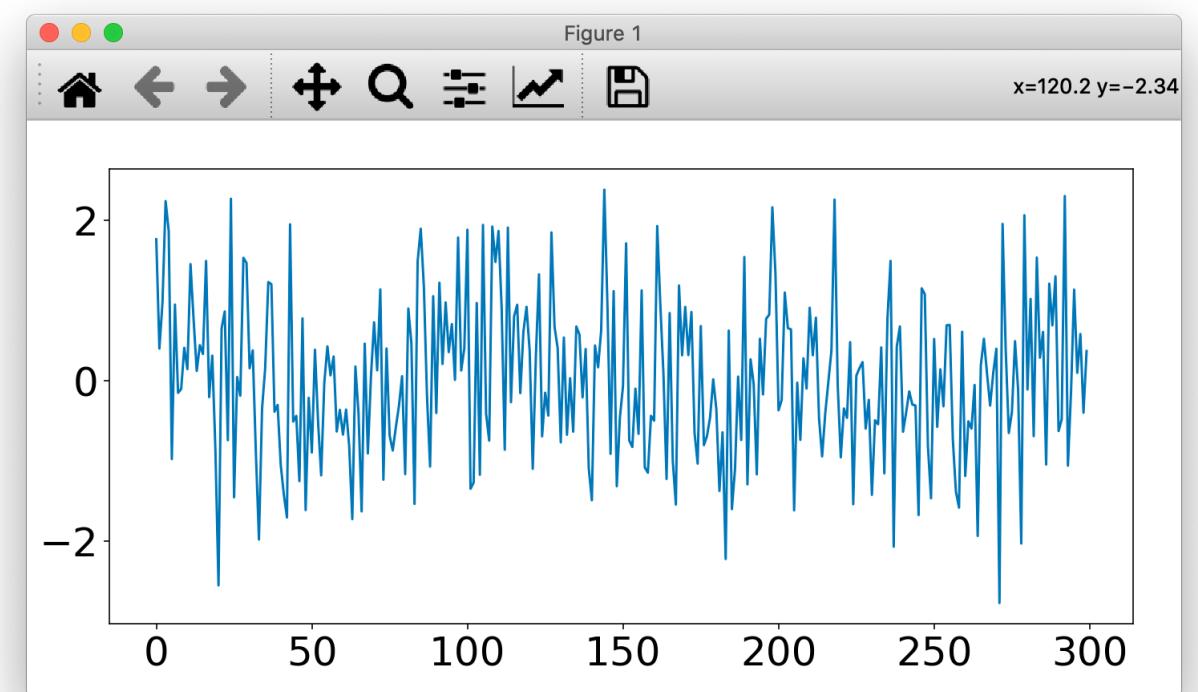
y_data = np.random.normal(loc=0, scale=1, size=(300,))

fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(y_data)

fig.tight_layout(pad=3)
ax.tick_params(labelsize=25)

-----
x_ticks = np.arange(301, step=50)
ax.set_xticks(x_ticks)

-----
x_ticks = np.arange(301, step=100)
ax.set_xticks(x_ticks)
```



Lecture_2-01 Line Plot Basics

6

1. ax.plot(y)

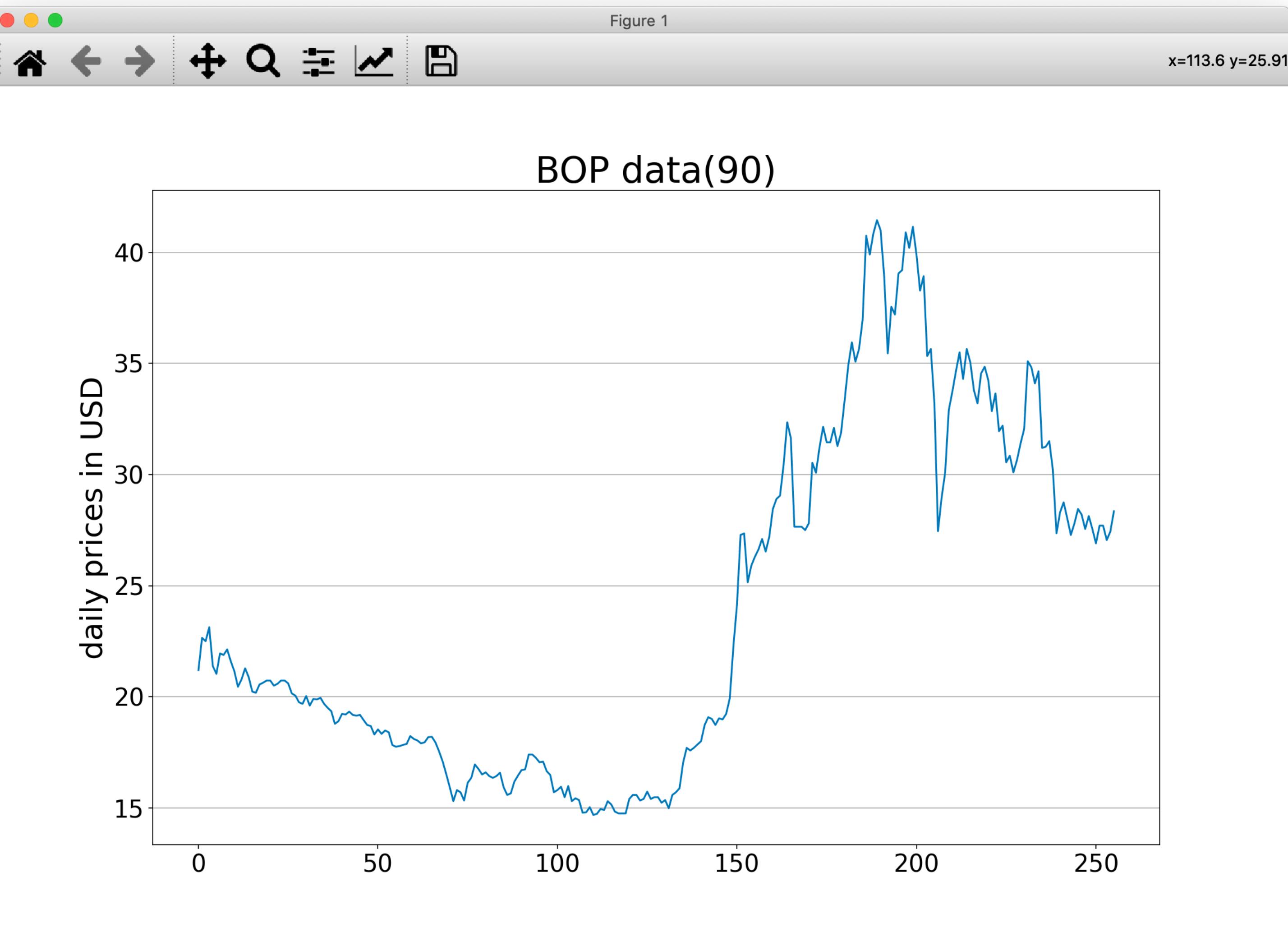
```
import matplotlib.pyplot as plt
import numpy as np

from bop_utils import *

dataset = bop_data_reader()
t_year = 90
t_year_data = get_year_data(dataset, t_year)

fig, ax = plt.subplots(figsize=(15, 10))
ax.plot(t_year_data[:, -1])

ax.set_title("BOP data({})".format(t_year),
             fontsize=30)
ax.set_ylabel("daily prices in USD",
              fontsize=25)
ax.tick_params(labelsize=20)
ax.grid(axis='y')
```



Lecture 2-01 Line Plot Basics

7

2. ax.plot(x, y)

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

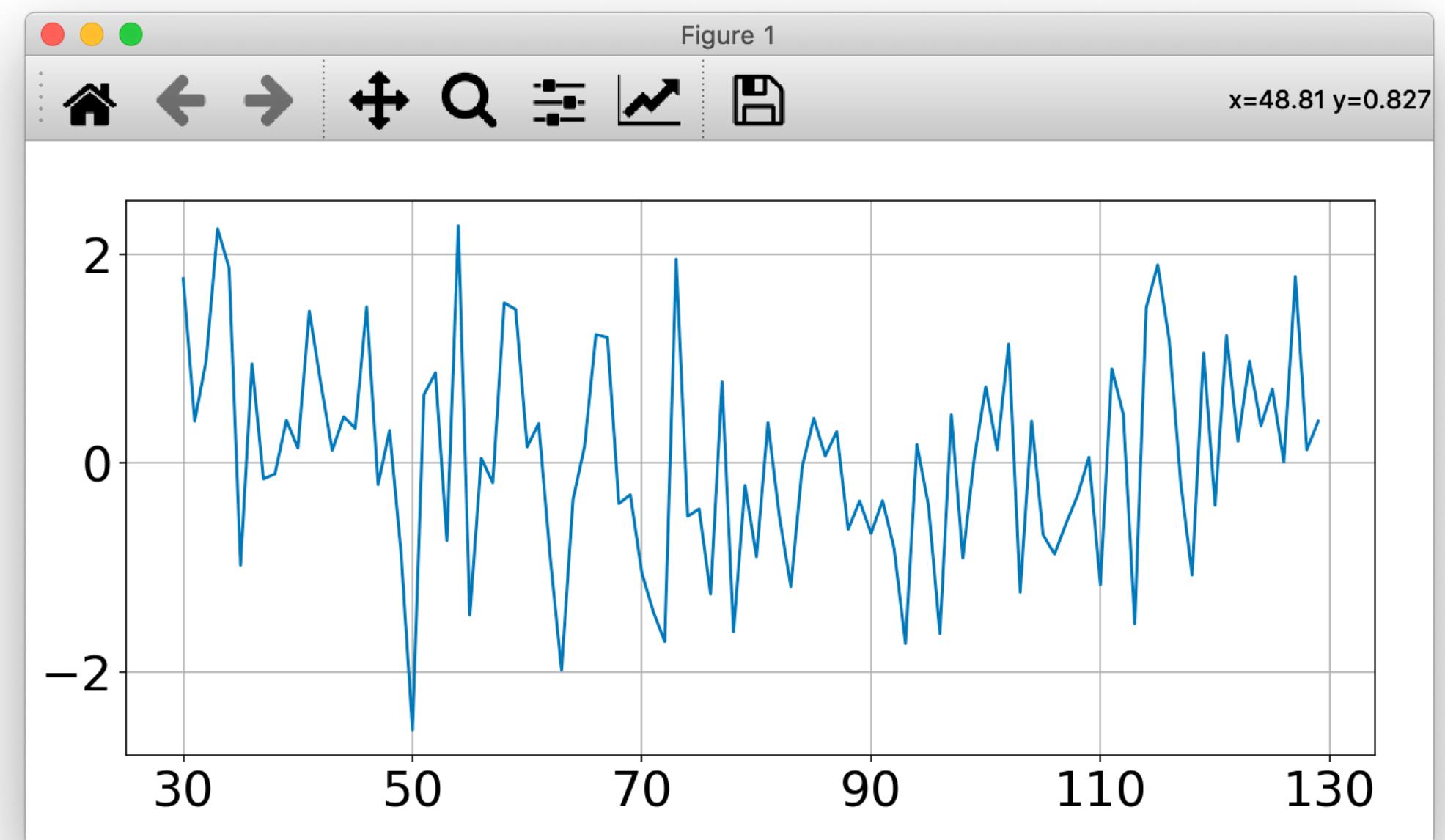
np.random.seed(0)

n_data = 100
s_idx = 30
x_data = np.arange(s_idx, s_idx + n_data)
y_data = np.random.normal(0, 1, (n_data, ))

fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(x_data, y_data)

fig.tight_layout(pad=3)
x_ticks = np.arange(s_idx, s_idx + n_data + 1, 20)
ax.set_xticks(x_ticks)

ax.tick_params(labelsize=25)
ax.grid()
```



Lecture. 2-01 Line Plot Basics

8

2. ax.plot(x, y)

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

np.random.seed(0)

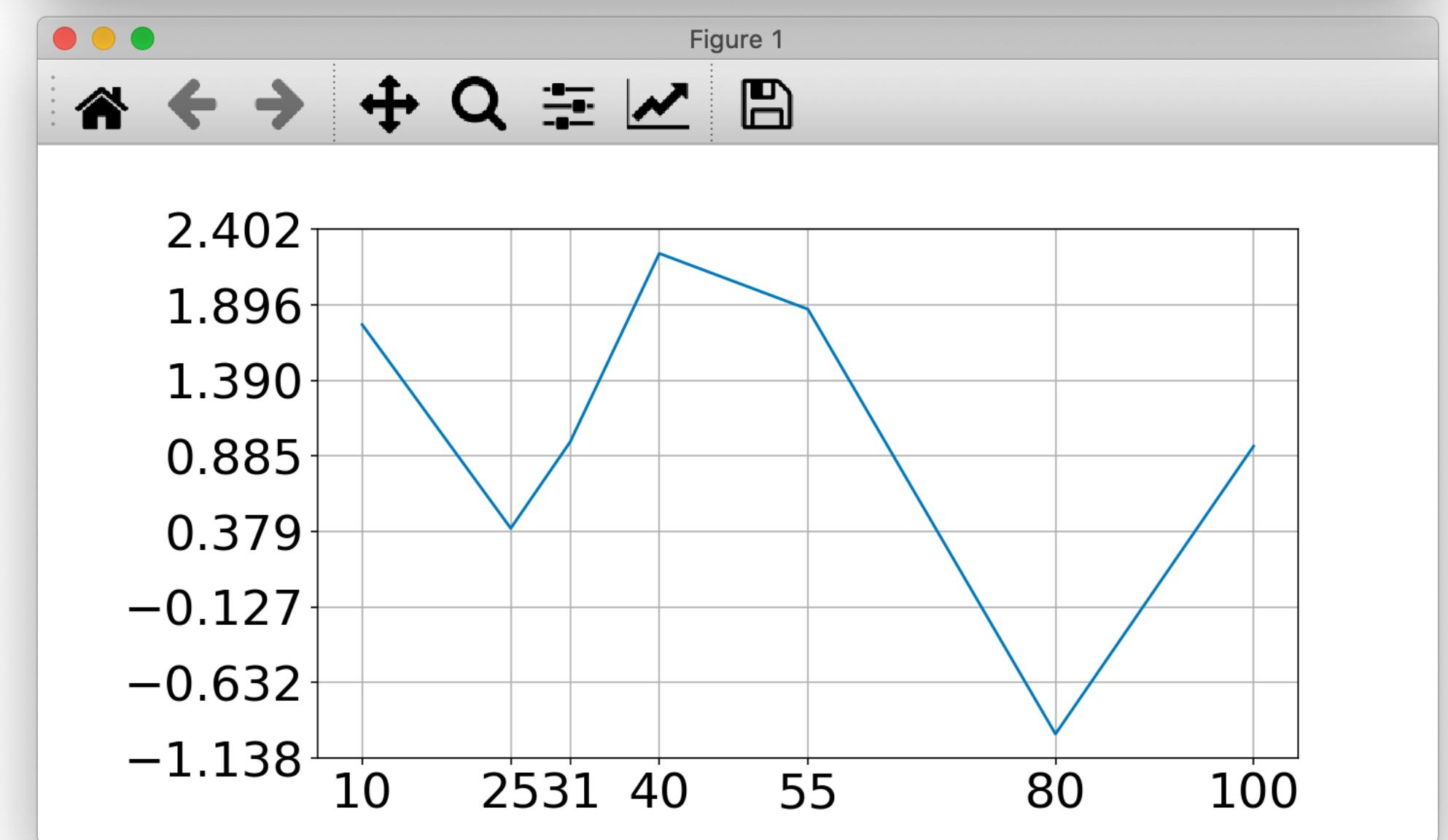
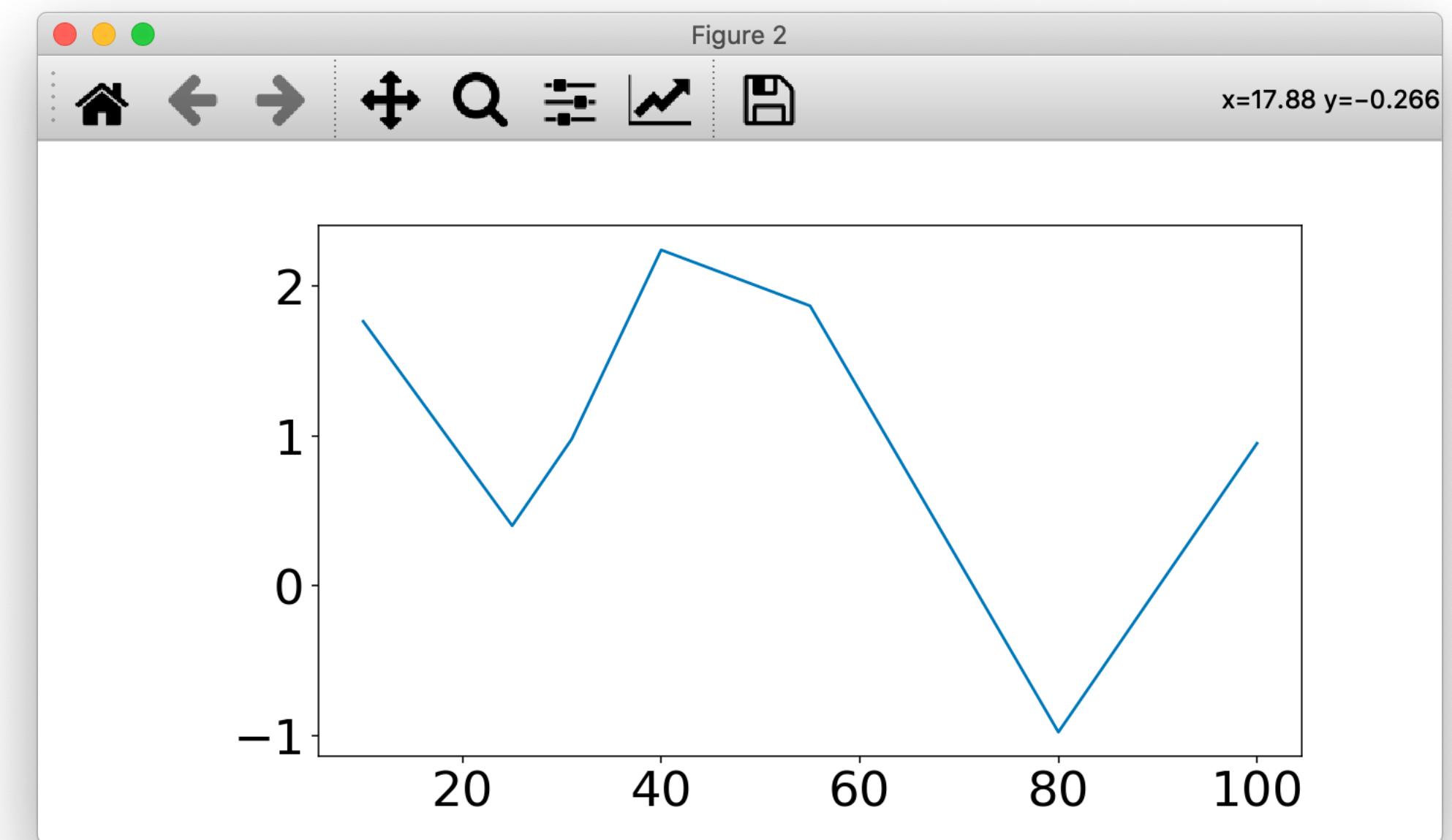
x_data = np.array([10, 25, 31, 40, 55, 80, 100])
y_data = np.random.normal(0, 1, (7,))

fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(x_data, y_data)

fig.subplots_adjust(left=0.2)
ax.tick_params(labelsize=25)

-----
ax.set_xticks(x_data)
ylim = ax.get_ylimits()
yticks = np.linspace(ylim[0], ylim[1], 8)
ax.set_yticks(yticks)

ax.grid()
```



Lecture 2-01 Line Plot Basics

9

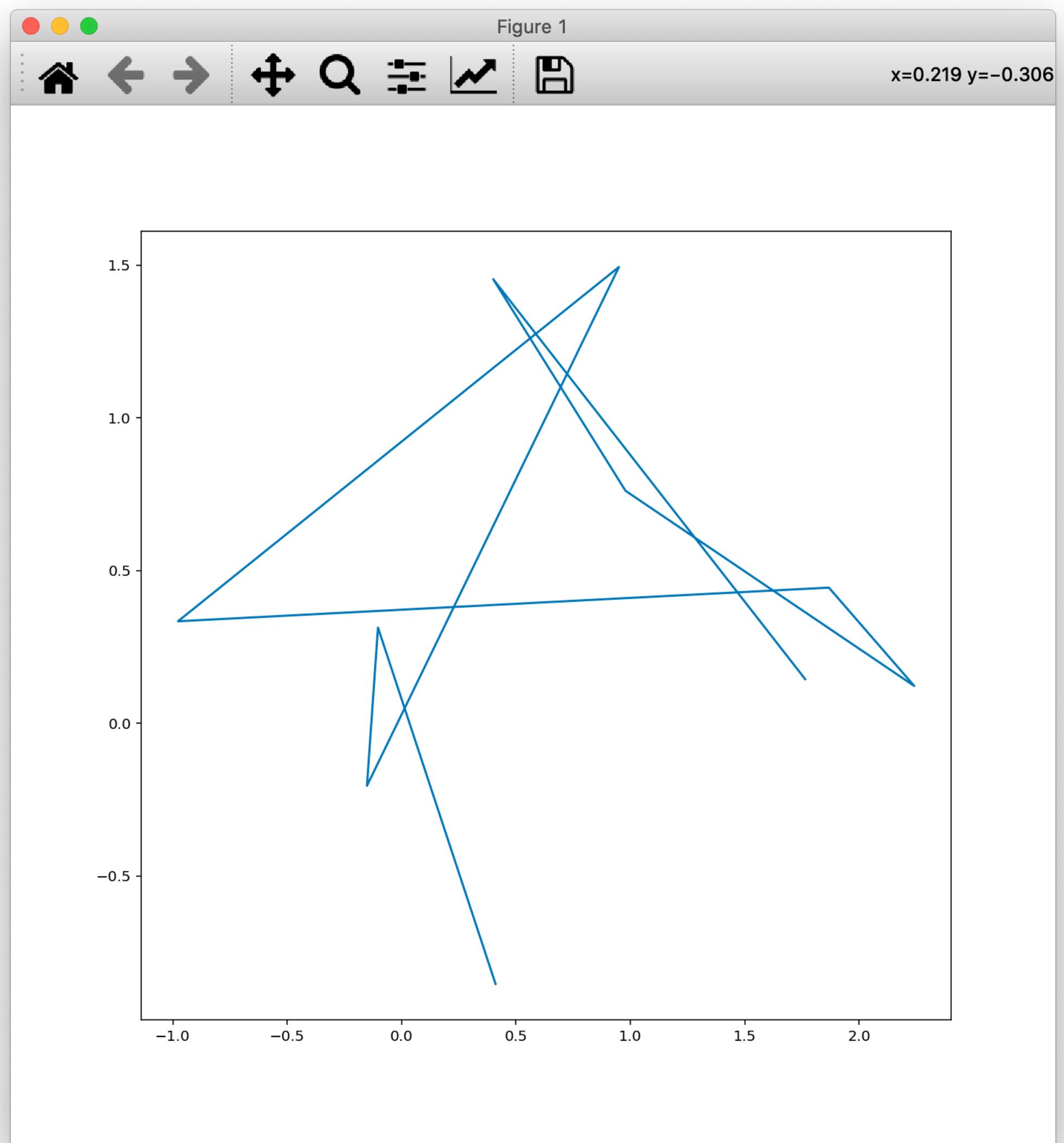
2. ax.plot(x, y)

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

np.random.seed(0)

x_data = np.random.normal(0, 1, (10, ))
y_data = np.random.normal(0, 1, (10, ))

fig, ax = plt.subplots(figsize=(10, 10))
ax.plot(x_data, y_data)
```



Lecture. 2-01 Line Plot Basics

10

2. ax.plot(x, y)

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

from bop_utils import *

dataset = bop_data_reader()
t_year = 89
t_year_data = get_year_data(dataset, t_year).astype(np.int)

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

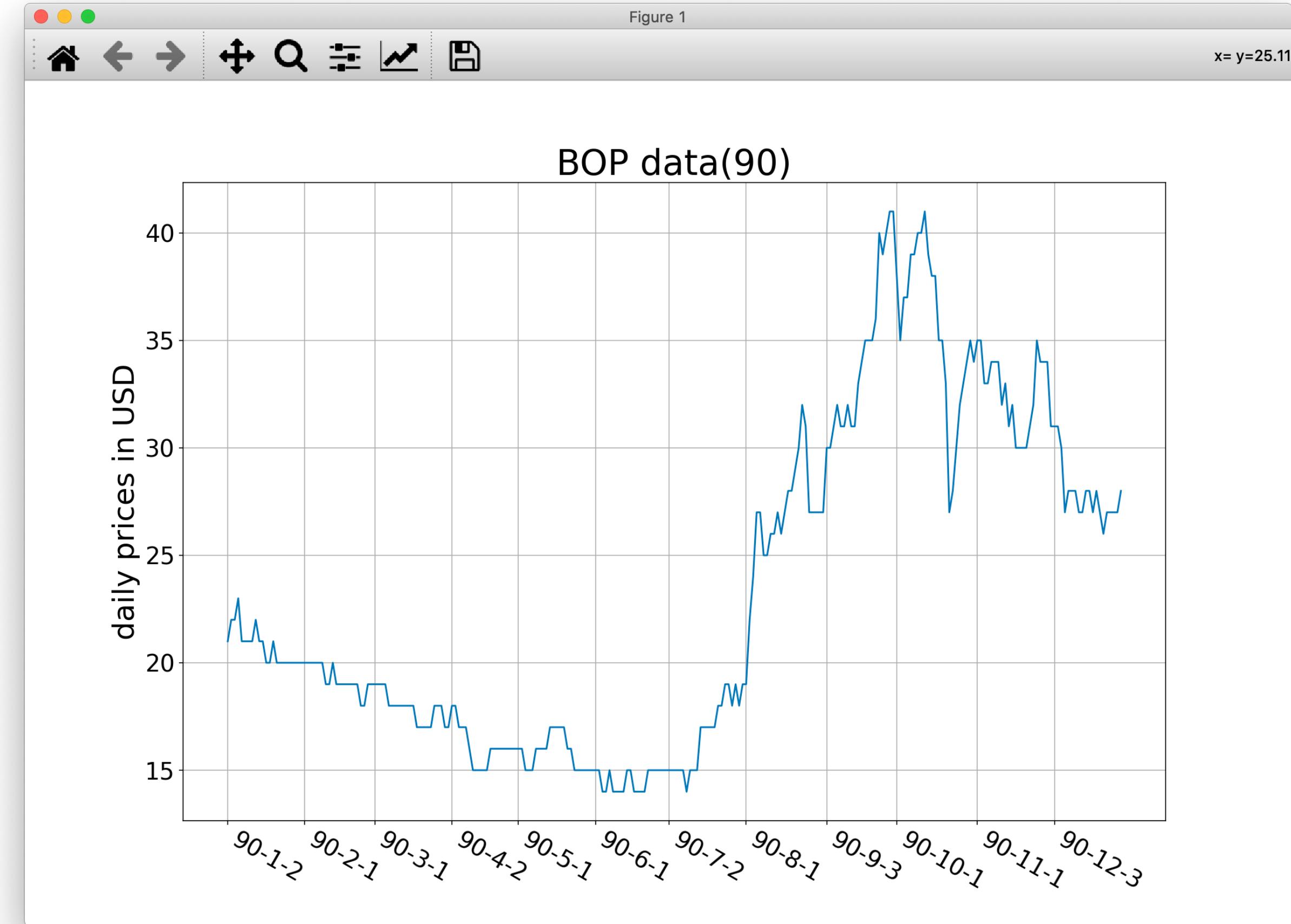
month_dict = {m:None for m in range(1, 1 + 12)}
for data_idx, data in enumerate(t_year_data):
    Y, M, D, price = data
    if month_dict[M] is None:
        month_dict[M] = [data_idx,
                         '-'.join(str(i) for i in [Y, M, D])]

first_day_label = np.array(list(month_dict.values()))

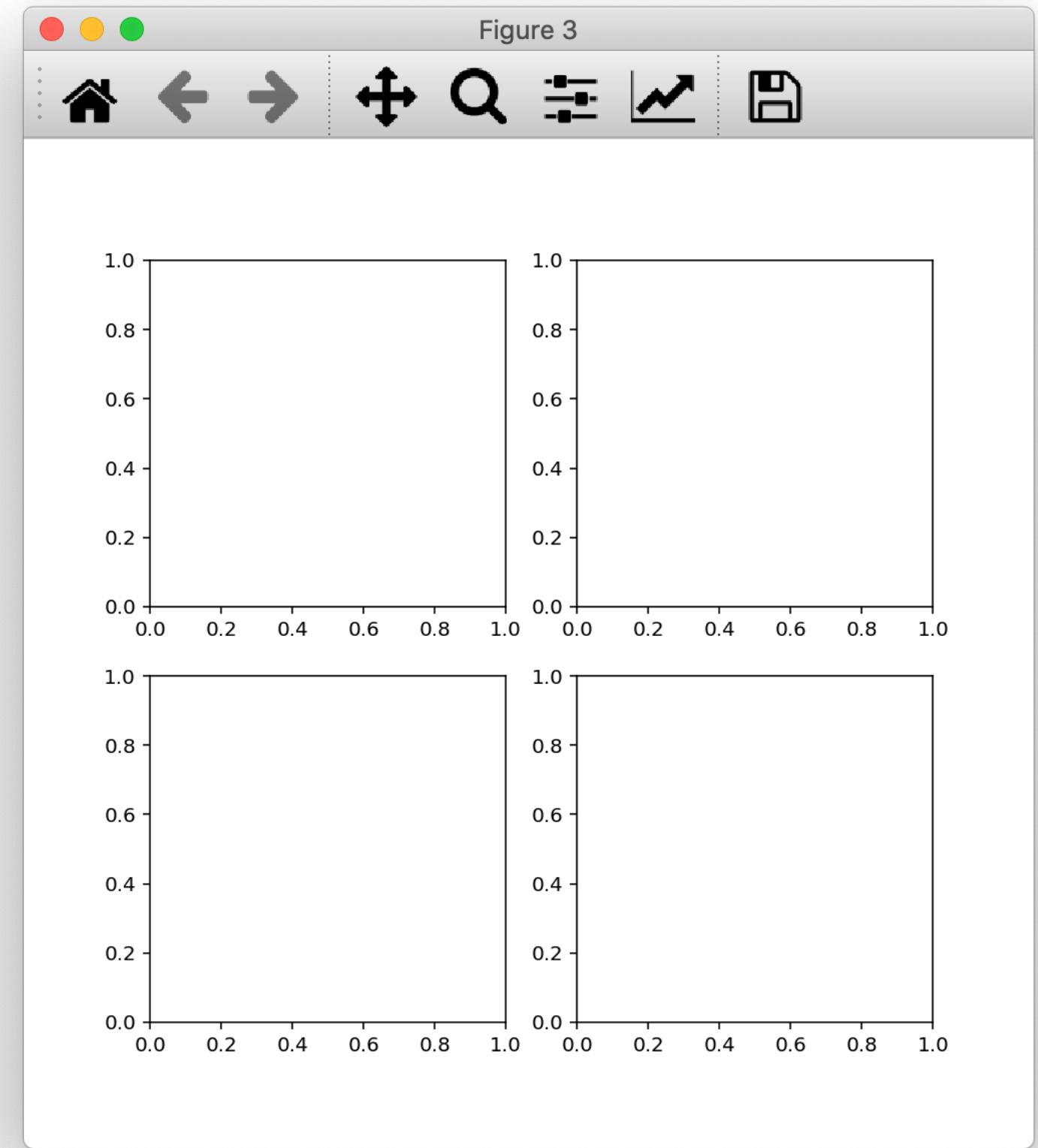
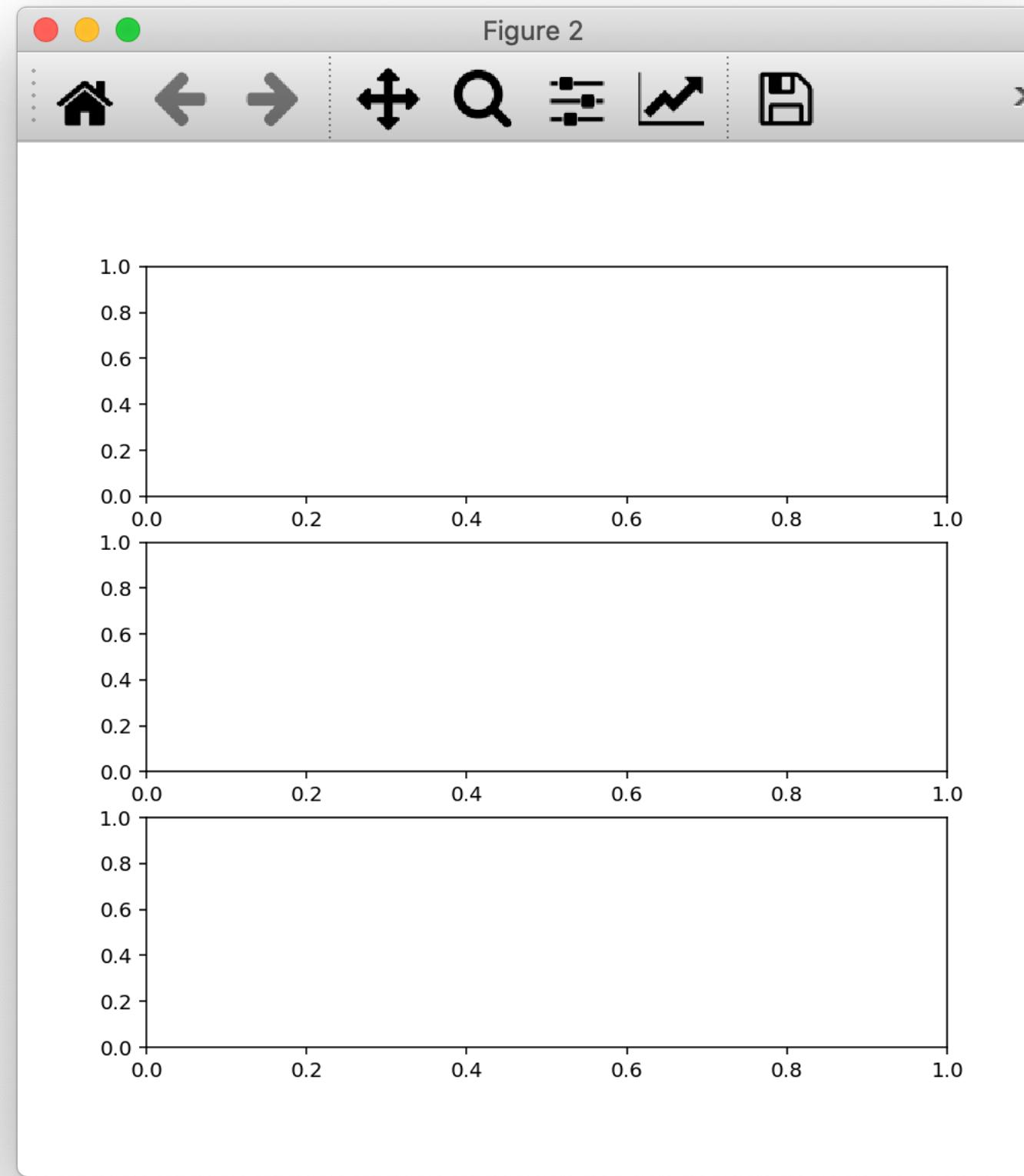
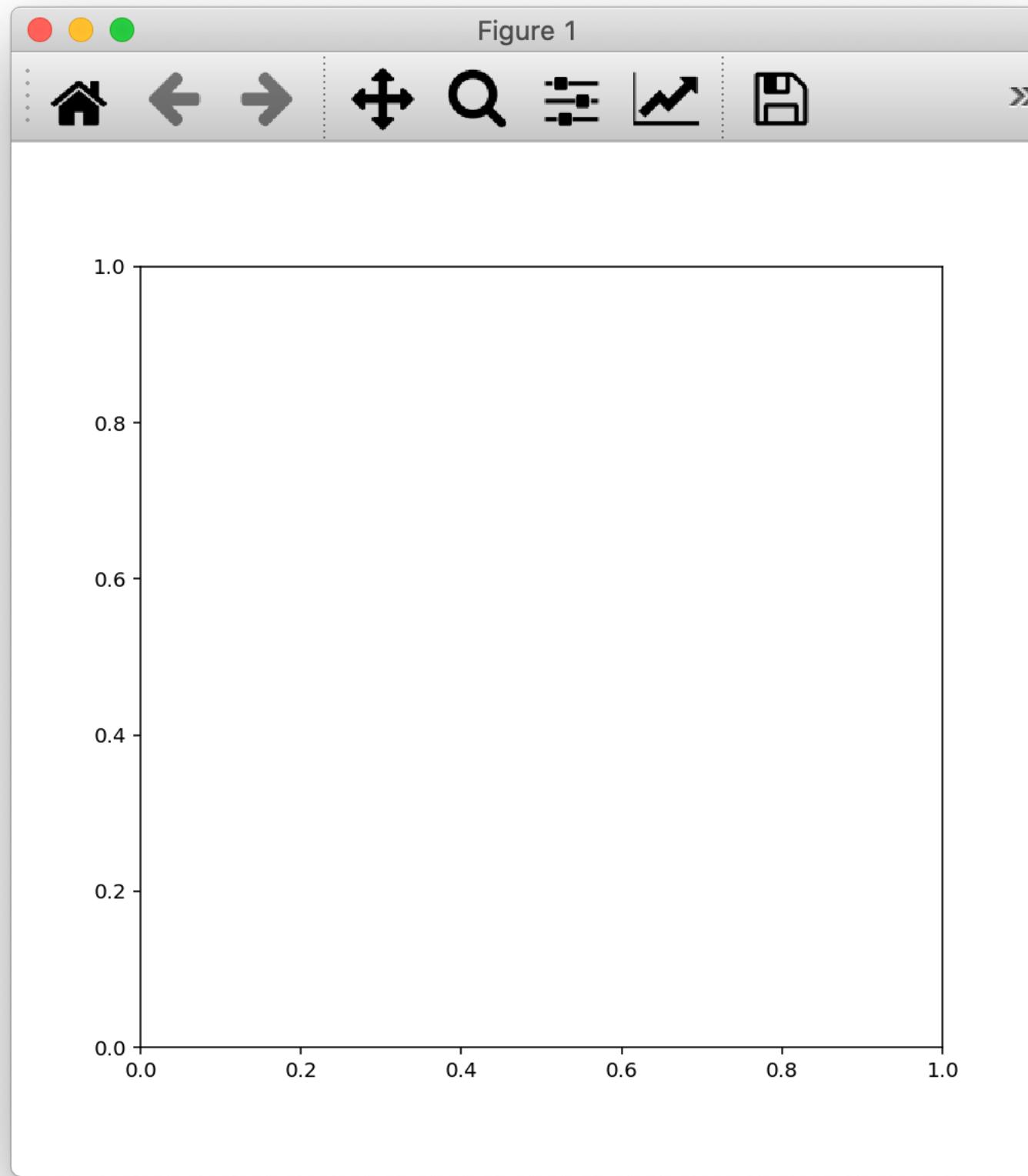
x_arange = np.arange(t_year_data.shape[0])
ax.plot(x_arange, t_year_data[:, -1])

ax.set_xticks(first_day_label[:, 0].astype(np.int))
ax.set_xticklabels(first_day_label[:, 1],
                   rotation=-30,
                   ha='left')

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



3. Several Line Plots on One Ax



3. Several Line Plots on One Ax

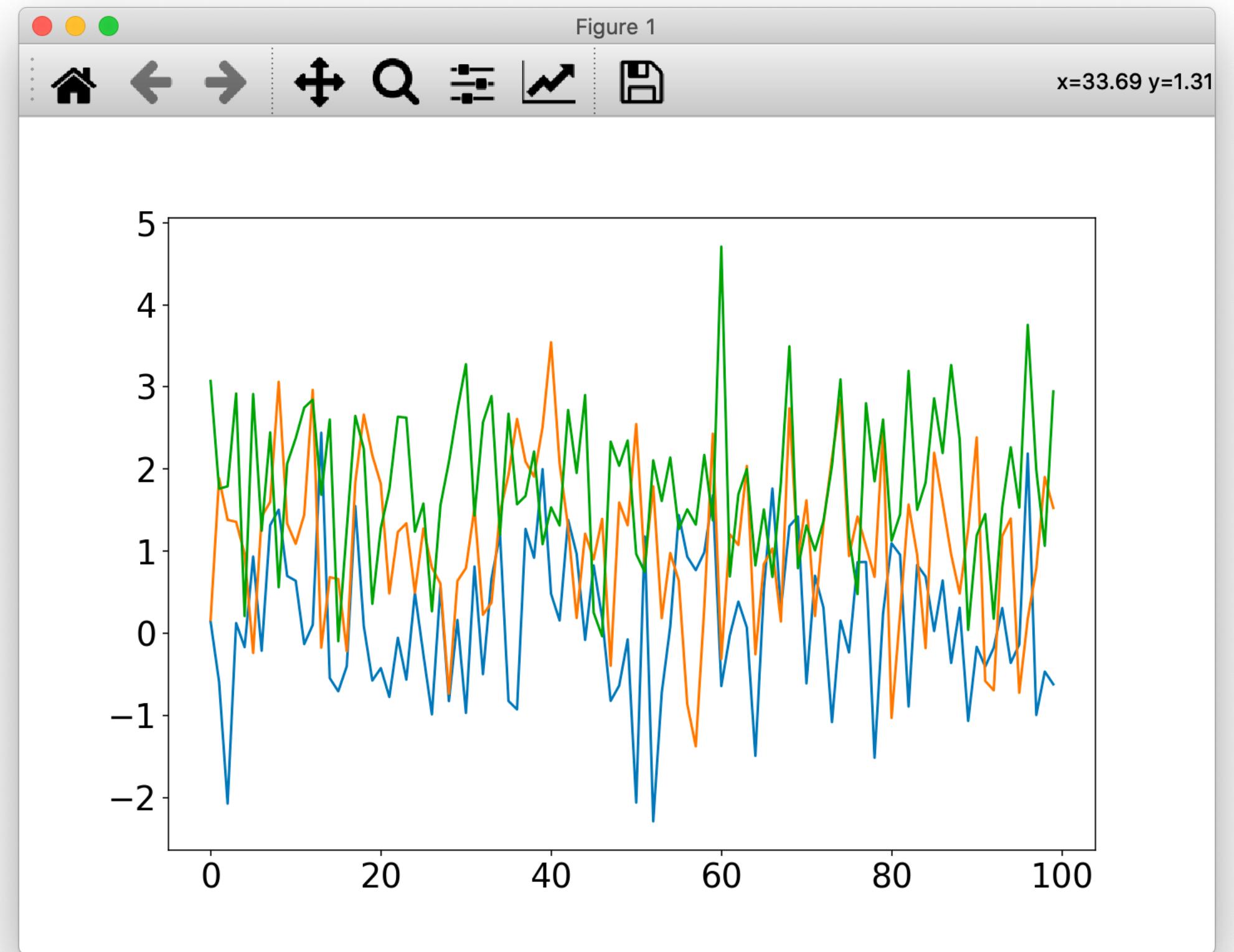
```
n_data = 100

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.plot(random_noise1)
ax.plot(random_noise2)
ax.plot(random_noise3)

ax.tick_params(labelsize=20)
```



3. Several Line Plots on One Ax

```
n_data1, n_data2, n_data3 = 200, 50, 10
```

```
x_data1 = np.linspace(0, 200, n_data1)
```

```
x_data2 = np.linspace(0, 200, n_data2)
```

```
x_data3 = np.linspace(0, 200, n_data3)
```

```
random_noisel = np.random.normal(0, 1, (n_data1,))
```

```
random_noise2 = np.random.normal(1, 1, (n_data2,))
```

```
random_noise3 = np.random.normal(2, 1, (n_data3,))
```

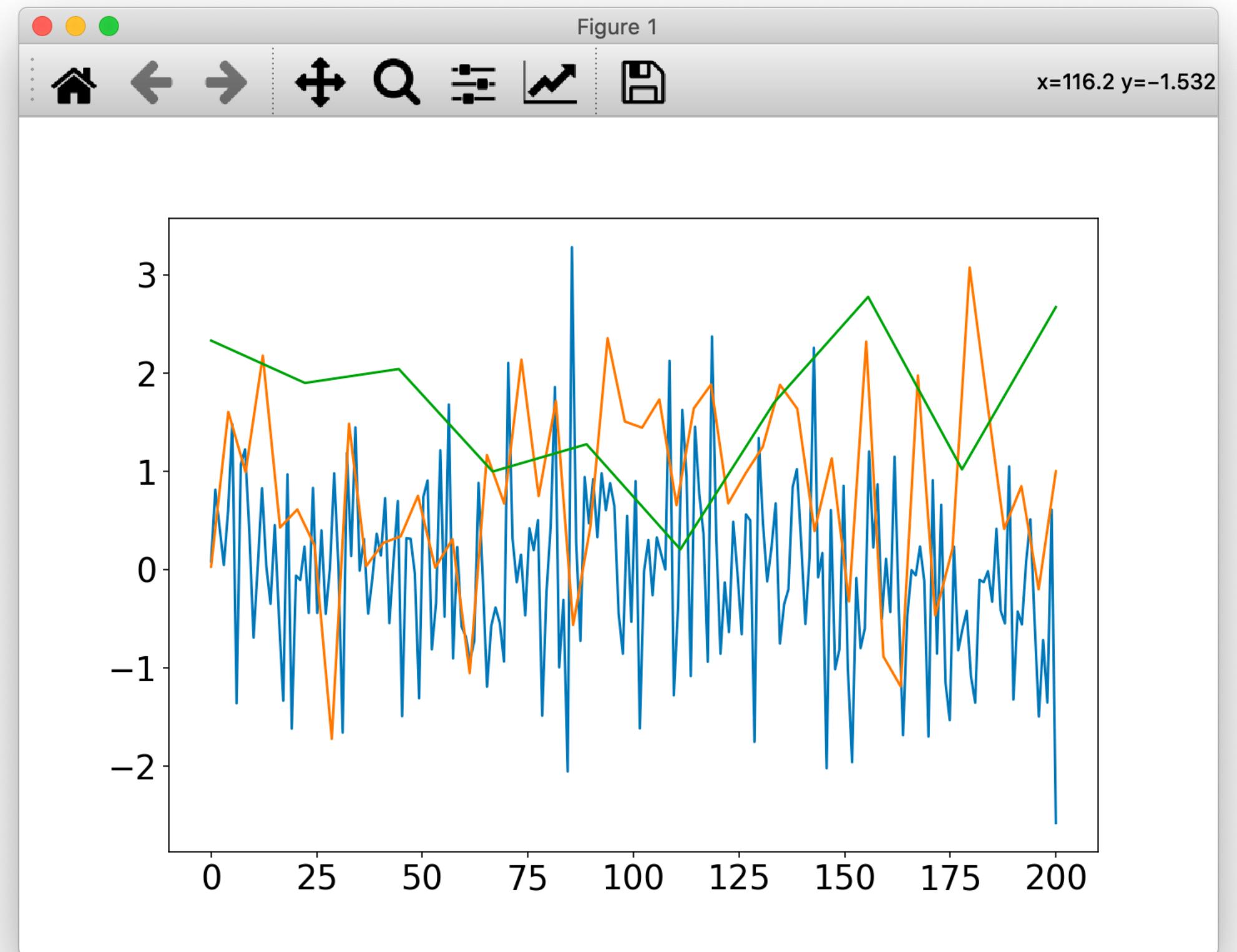
```
fig, ax = plt.subplots(figsize=(10, 7))
```

```
ax.plot(x_data1, random_noisel)
```

```
ax.plot(x_data2, random_noise2)
```

```
ax.plot(x_data3, random_noise3)
```

```
ax.tick_params(labelsize=20)
```



3. Several Line Plots on One Ax

```
PI = np.pi
t = np.linspace(-4*PI, 4*PI, 300)
sin = np.sin(t)
linear = 0.1*t

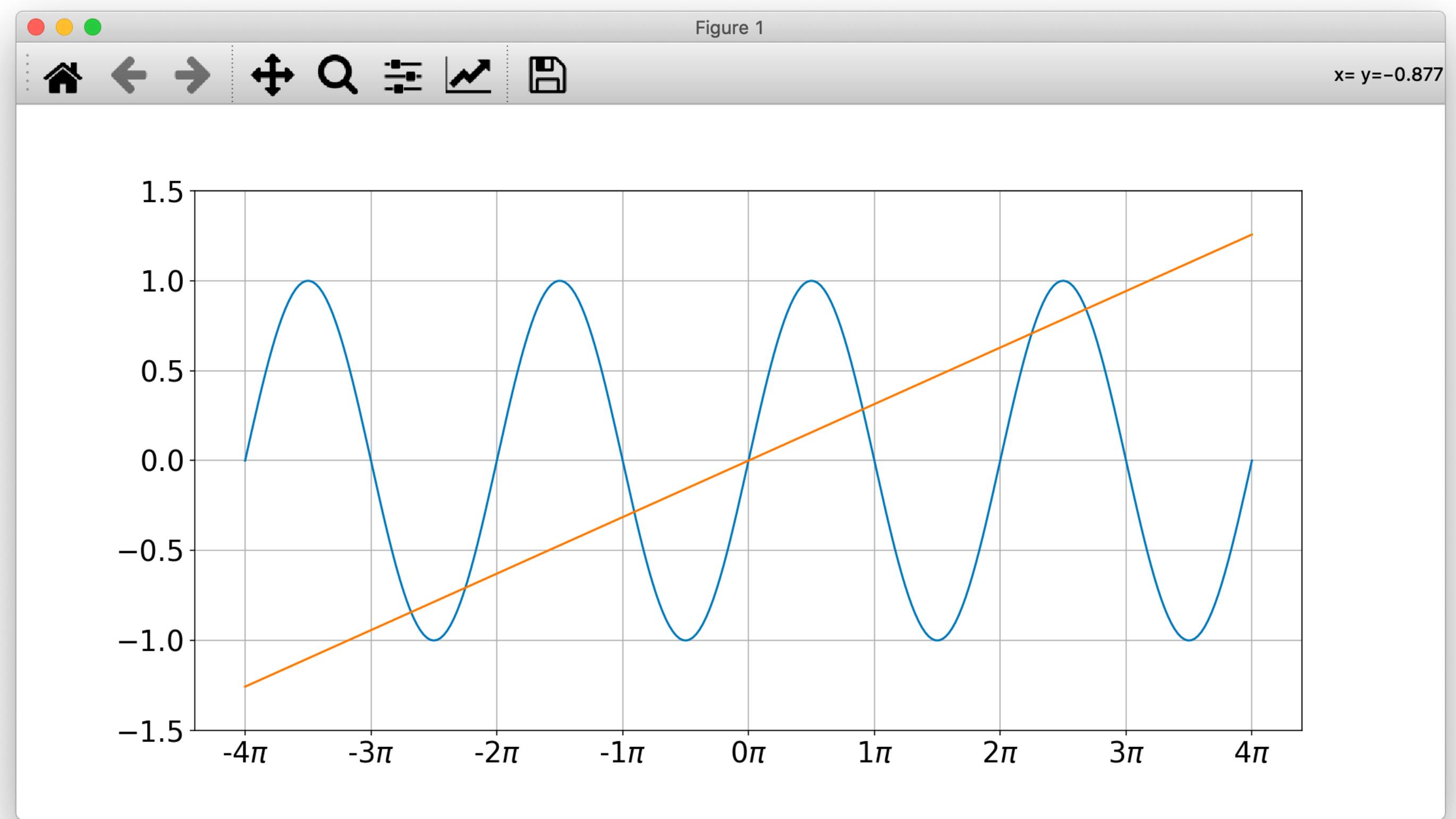
fig, ax = plt.subplots(figsize=(14, 7))
ax.plot(t, sin)
ax.plot(t, linear)

ax.set_ylim([-1.5, 1.5])

x_ticks = np.arange(-4*PI, 4*PI+0.1, PI)
x_ticklabels = [str(i) + r'$\pi$'
                 for i in range(-4, 5)]

ax.set_xticks(x_ticks)
ax.set_xticklabels(x_ticklabels)

ax.tick_params(labelsize=20)
ax.grid()
```



3. Several Line Plots on One Ax

```
from bop_utils import *

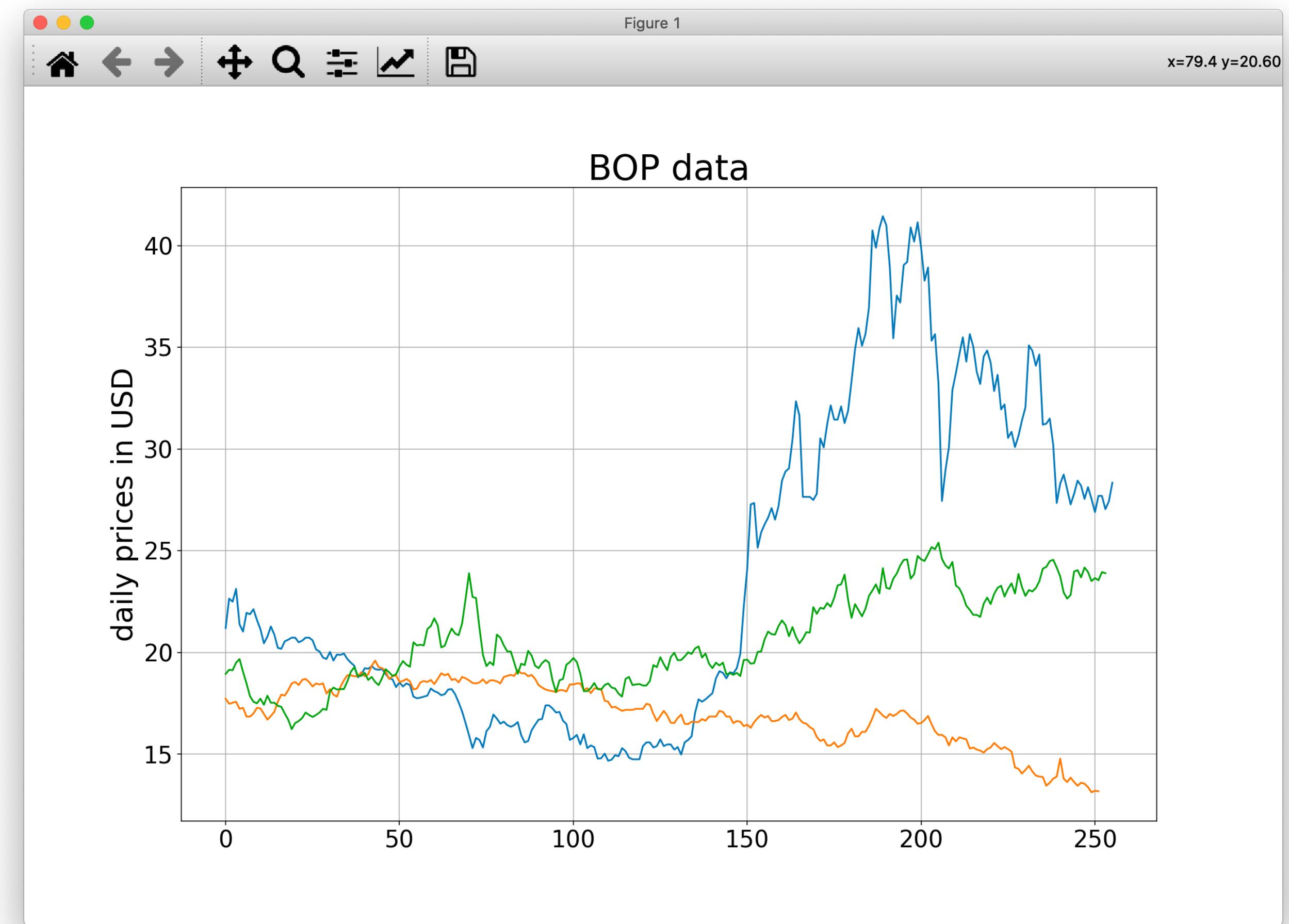
def bop_plot(dataset, t_year, ax):
    t_year_data = get_year_data(dataset, t_year)
    ax.plot(t_year_data[:, -1])

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()
```



4. Several Line Plots on Different Axes

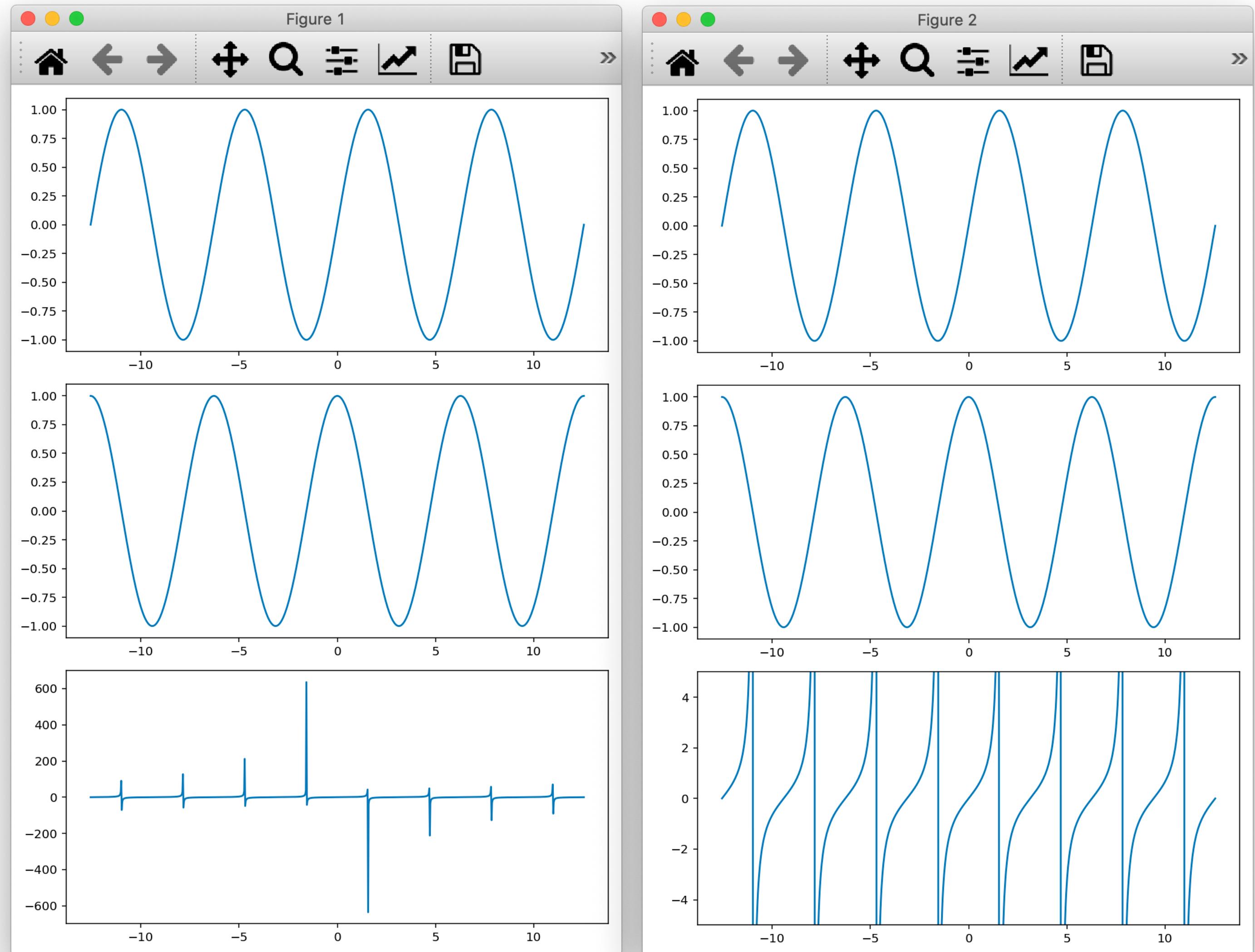
```
PI = np.pi
t = np.linspace(-4*PI, 4*PI, 1000)
sin = np.sin(t)
cos = np.cos(t)
tan = np.tan(t)

fig, axes = plt.subplots(3, 1,
                       figsize=(7, 10))

axes[0].plot(t, sin)
axes[1].plot(t, cos)
axes[2].plot(t, tan)

fig.tight_layout()

-----
axes[2].set_ylim([-5, 5])
```



4. Several Line Plots on Different Axes

```

PI = np.pi
t = np.linspace(-4*PI, 4*PI, 1000).reshape(1, -1)
sin = np.sin(t)
cos = np.cos(t)
tan = np.tan(t)
data = np.vstack((sin, cos, tan))

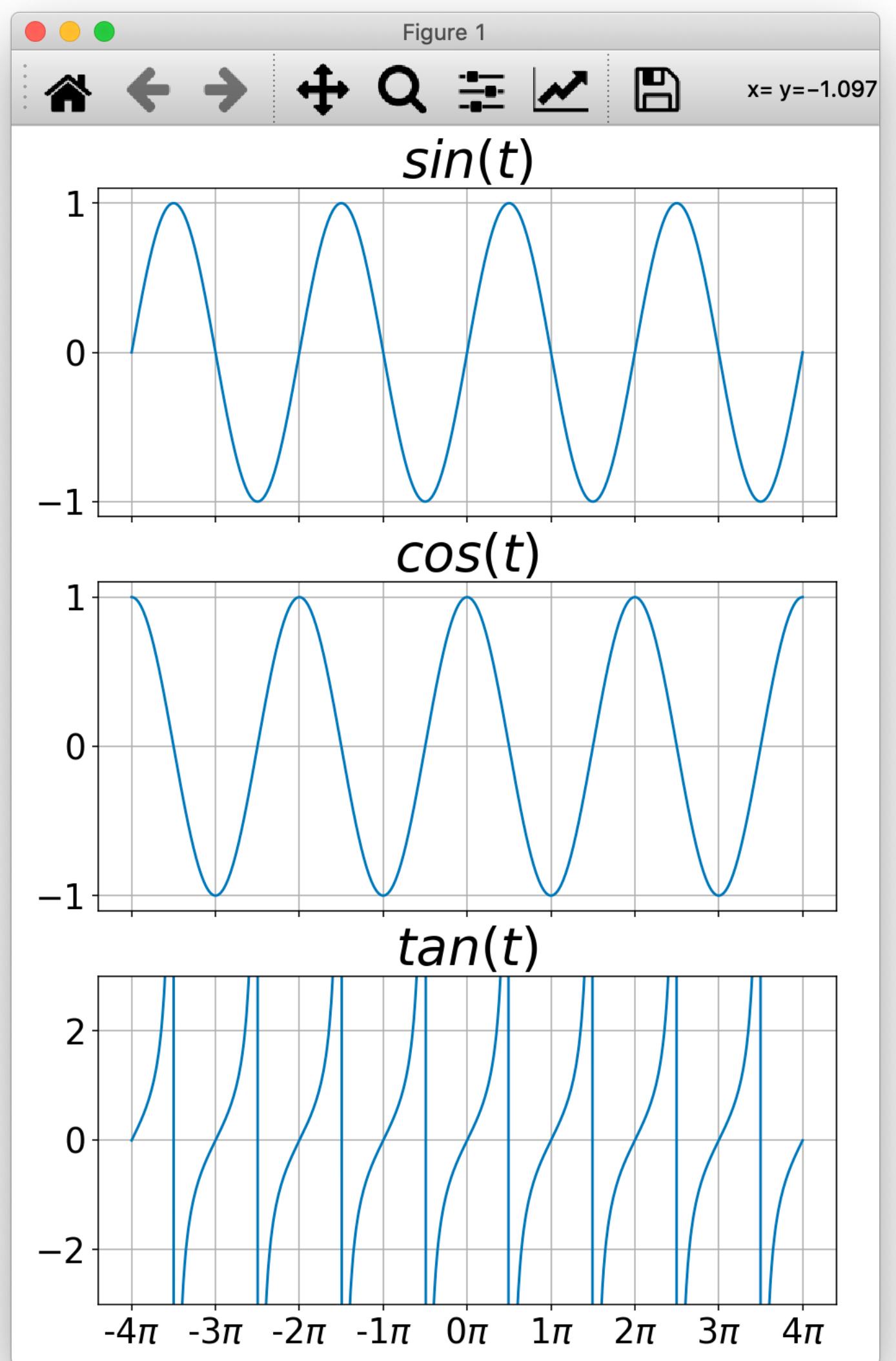
title_list = [r'$\sin(t)$', r'$\cos(t)$', r'$\tan(t)$']
x_ticks = np.arange(-4*PI, 4*PI+PI, PI)
x_ticklabels = [str(i) + r'$\pi$' for i in range(-4, 5)]

fig, axes = plt.subplots(3, 1,
                        figsize=(7, 10),
                        sharex=True)

for ax_idx, ax in enumerate(axes.flat):
    ax.plot(t.flatten(), data[ax_idx])
    ax.set_title(title_list[ax_idx],
                 fontsize=30)
    ax.tick_params(labelsize=20)
    ax.grid()
    if ax_idx == 2:
        ax.set_ylim([-3, 3])

fig.subplots_adjust(left=0.1, right=0.95,
                    bottom=0.05, top=0.95)
axes[-1].set_xticks(x_ticks)
axes[-1].set_xticklabels(x_ticklabels)

```



4. Several Line Plots on Different Axes

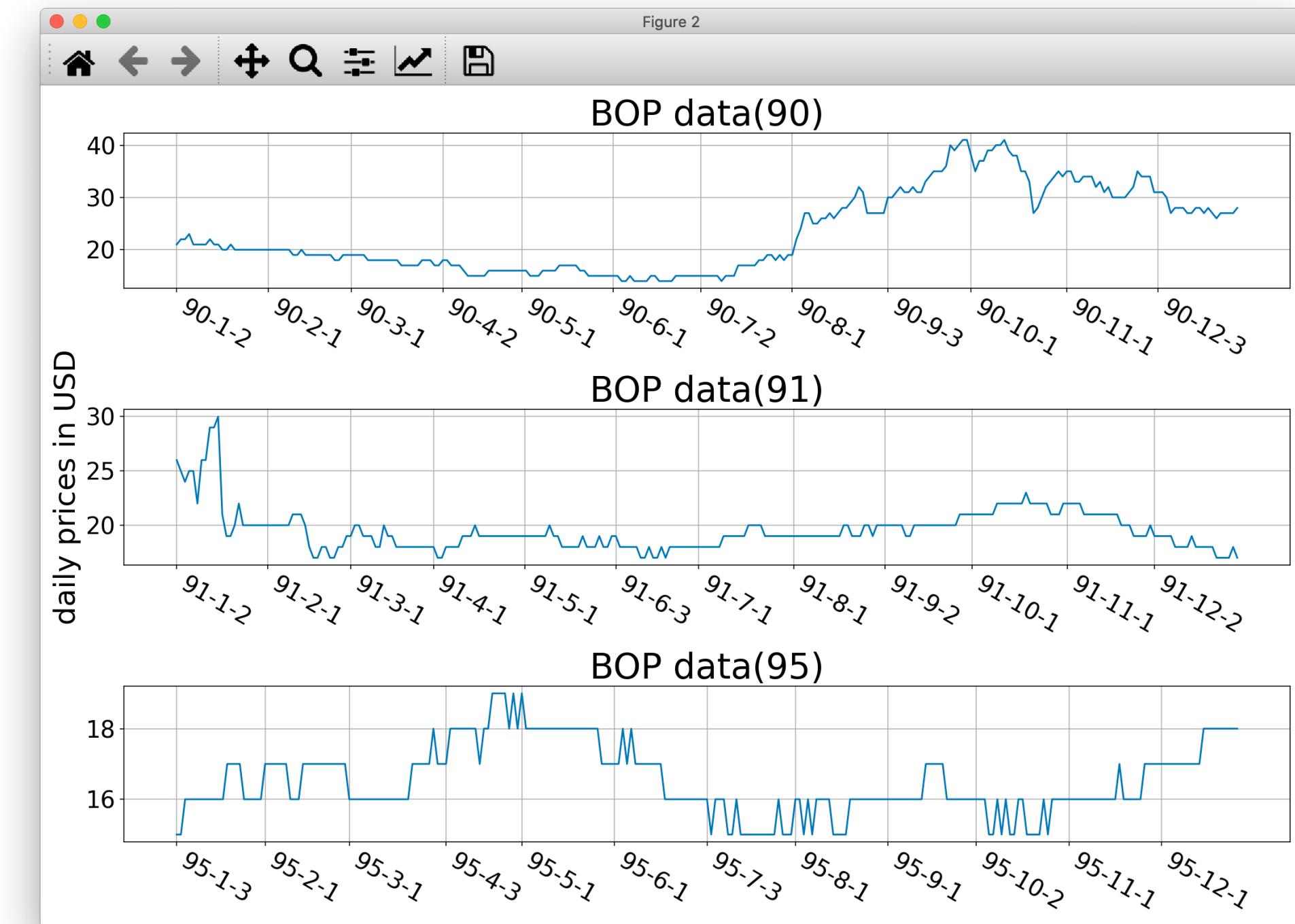
```
def bop_plot(dataset, t_year, ax):
    t_year_data = get_year_data(dataset, t_year).astype(np.int)

    month_dict = {m:None for m in range(1, 1 + 12)}
    for data_idx, data in enumerate(t_year_data):
        Y, M, D, price = data
        if month_dict[M] is None:
            month_dict[M] = [data_idx,
                             '-'.join(str(i)for i in [Y, M, D])]

    first_day_label = np.array(list(month_dict.values()))

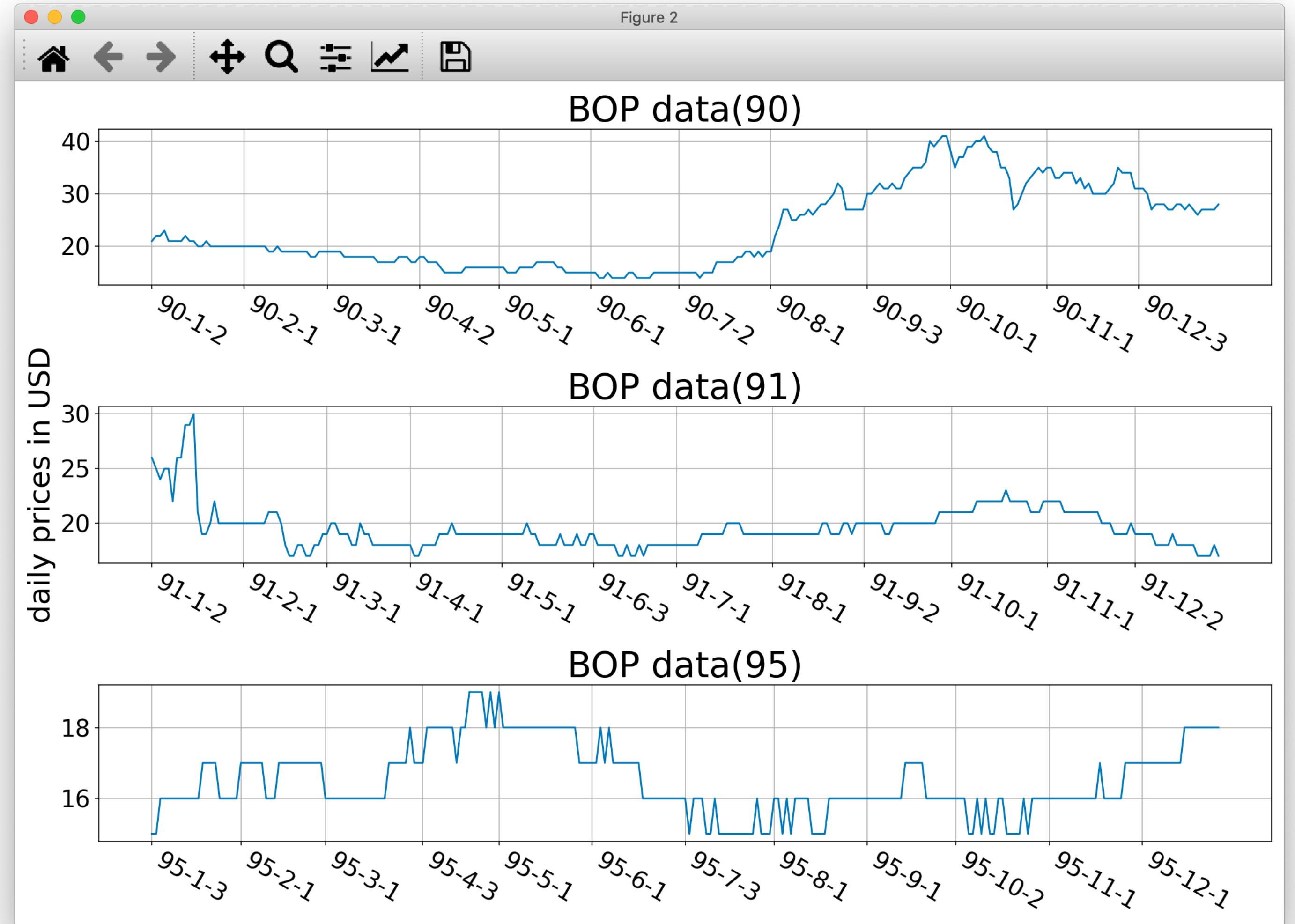
    x_arange = np.arange(t_year_data.shape[0])
    ax.plot(x_arange, t_year_data[:, -1])
    ax.set_xticks(first_day_label[:, 0].astype(np.int))
    ax.set_xticklabels(first_day_label[:, 1],
                       rotation=-30,
                       ha='left')
    ax.set_title("BOP data({})".format(t_year),
                 fontsize=30)

    ax.tick_params(labelsize=20)
    ax.grid()
```



4. Several Line Plots on Different Axes

```
def bop_plot(dataset, t_year, ax):  
  
    t_year_list = [90, 91, 95]  
    fig, axes = plt.subplots(len(t_year_list), 1,  
                           figsize=(15, 10))  
  
    for ax_idx, ax in enumerate(axes.flat):  
        bop_plot(dataset,  
                 t_year_list[ax_idx],  
                 ax)  
        if ax_idx == 1:  
            ax.set_ylabel("daily prices in USD",  
                         fontsize=25)  
  
    fig.tight_layout()
```



5. ax.axvline and ax.axhline

matplotlib.pyplot.axvline

```
matplotlib.pyplot.axvline(x=0, ymin=0, ymax=1, **kwargs)
```

[\[source\]](#)

Add a vertical line across the axes.

Parameters:

x: float, default: 0

x position in data coordinates of the vertical line.

ymin : float, default: 0

Should be between 0 and 1, 0 being the bottom of the plot, 1 the top of the plot.

ymax : float, default: 1

Should be between 0 and 1, 0 being the bottom of the plot, 1 the top of the plot.

matplotlib.pyplot.axhline

```
matplotlib.pyplot.axhline(y=0, xmin=0, xmax=1, **kwargs)
```

[\[source\]](#)

Add a horizontal line across the axis.

Parameters:

y: float, default: 0

y position in data coordinates of the horizontal line.

xmin : float, default: 0

Should be between 0 and 1, 0 being the far left of the plot, 1 the far right of the plot.

xmax : float, default: 1

Should be between 0 and 1, 0 being the far left of the plot, 1 the far right of the plot.

https://matplotlib.org/3.3.1/api/_as_gen/matplotlib.pyplot.axvline.html

https://matplotlib.org/3.3.1/api/_as_gen/matplotlib.pyplot.axhline.html

5. ax.axvline and ax.axhline

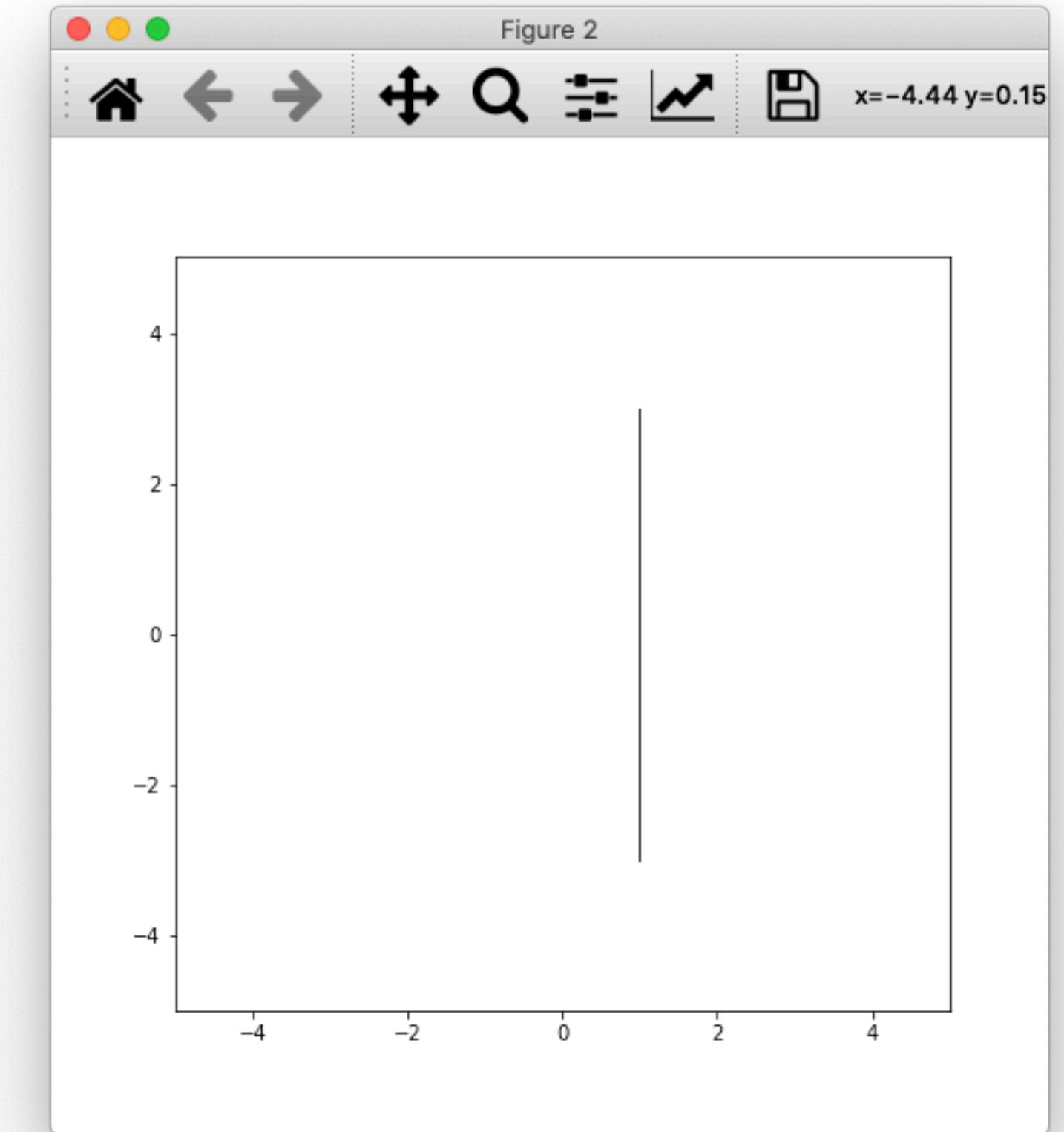
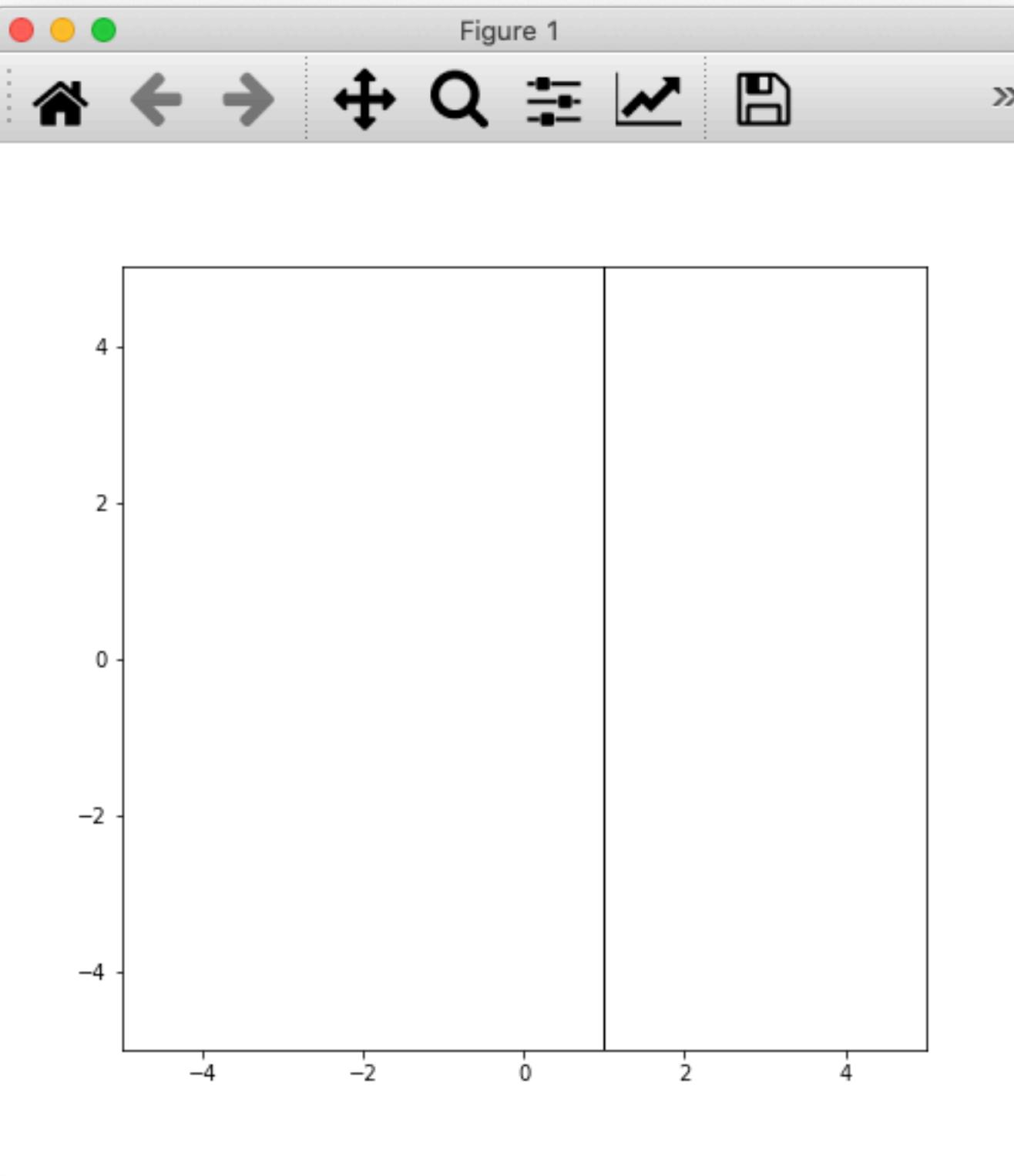
```
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots(figsize=(7, 7))

ax.set_xlim([-5, 5])
ax.set_ylim([-5, 5])

ax.axvline(x=1,
           color='black',
           linewidth=1)

-----
ax.axvline(x=1,
           ymax=0.8, ymin=0.2,
           color='black',
           linewidth=1)
```



5. ax.axvline and ax.axhline

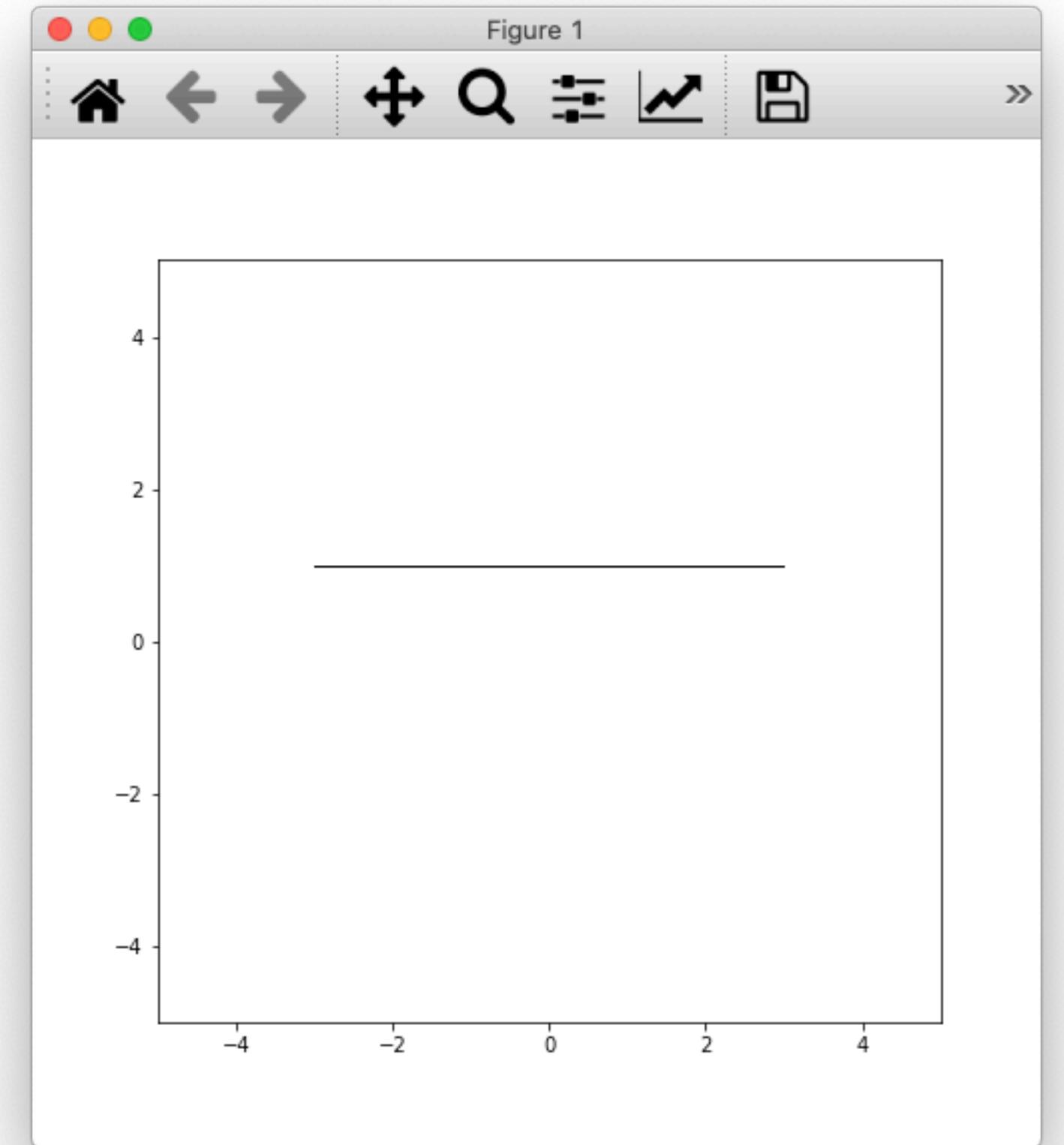
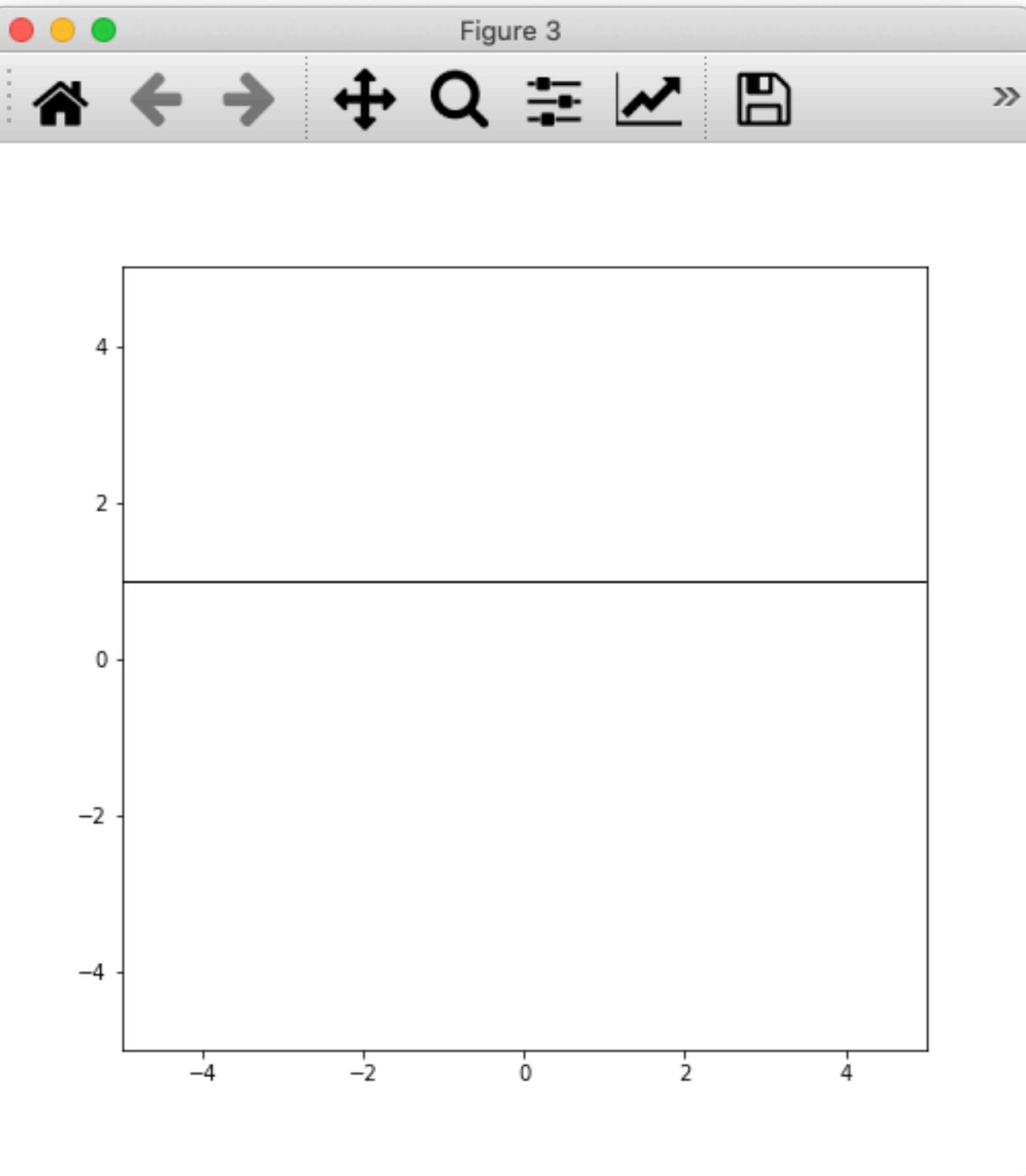
```
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots(figsize=(7, 7))

ax.set_xlim([-5, 5])
ax.set_ylim([-5, 5])

ax.axhline(y=1,
            color='black',
            linewidth=1)

-----
ax.axhline(y=1,
            xmax=0.8, xmin=0.2,
            color='black',
            linewidth=1)
```



Python for Data Visualization

2-01. Line Plot Basics

1. `ax.plot(y)`
2. `ax.plot(x, y)`
3. Several Line Plots on One Ax
4. Several Line Plots on Different Axes
5. `ax.axvline` and `ax.axhline`