

- Shin's Lab -

Python for Data Visualization

Python for Data Visualization

-Chapter.1 Matplotlib Anatomy -

1-01. Making Figures and Axes

1-02. Axes Customizing

1-03. Titles, Labels and Font Dict

1-04. Ticks and Ticklabels

1-05. Grid

1-06. Spines

1-07. Colors in Matplotlib

1-08. Matplotlib Styles and rcParams

Python for Data Visualization

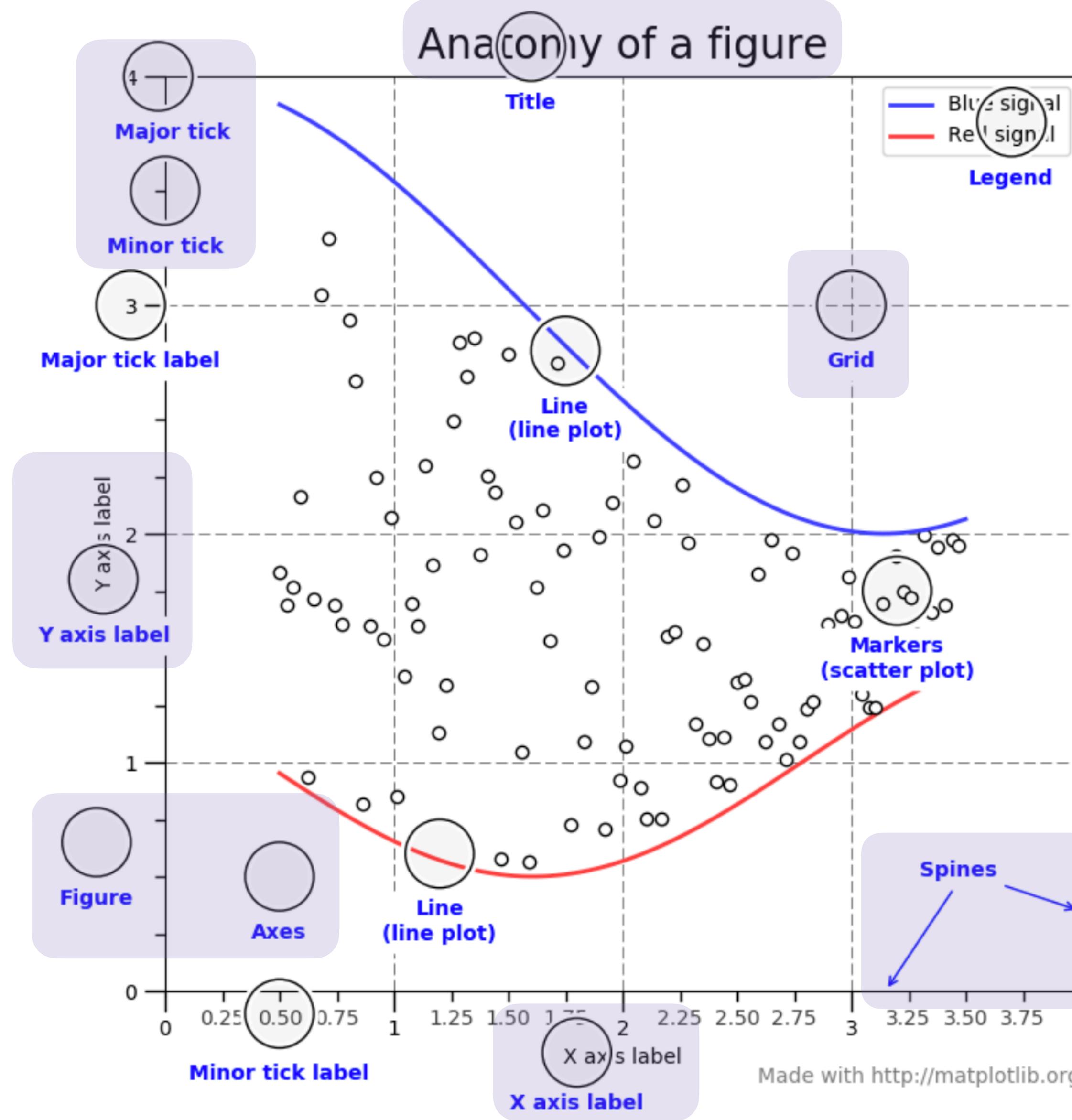
-Chapter.1 Matplotlib Anatomy -

1-07. Colors in Matplotlib

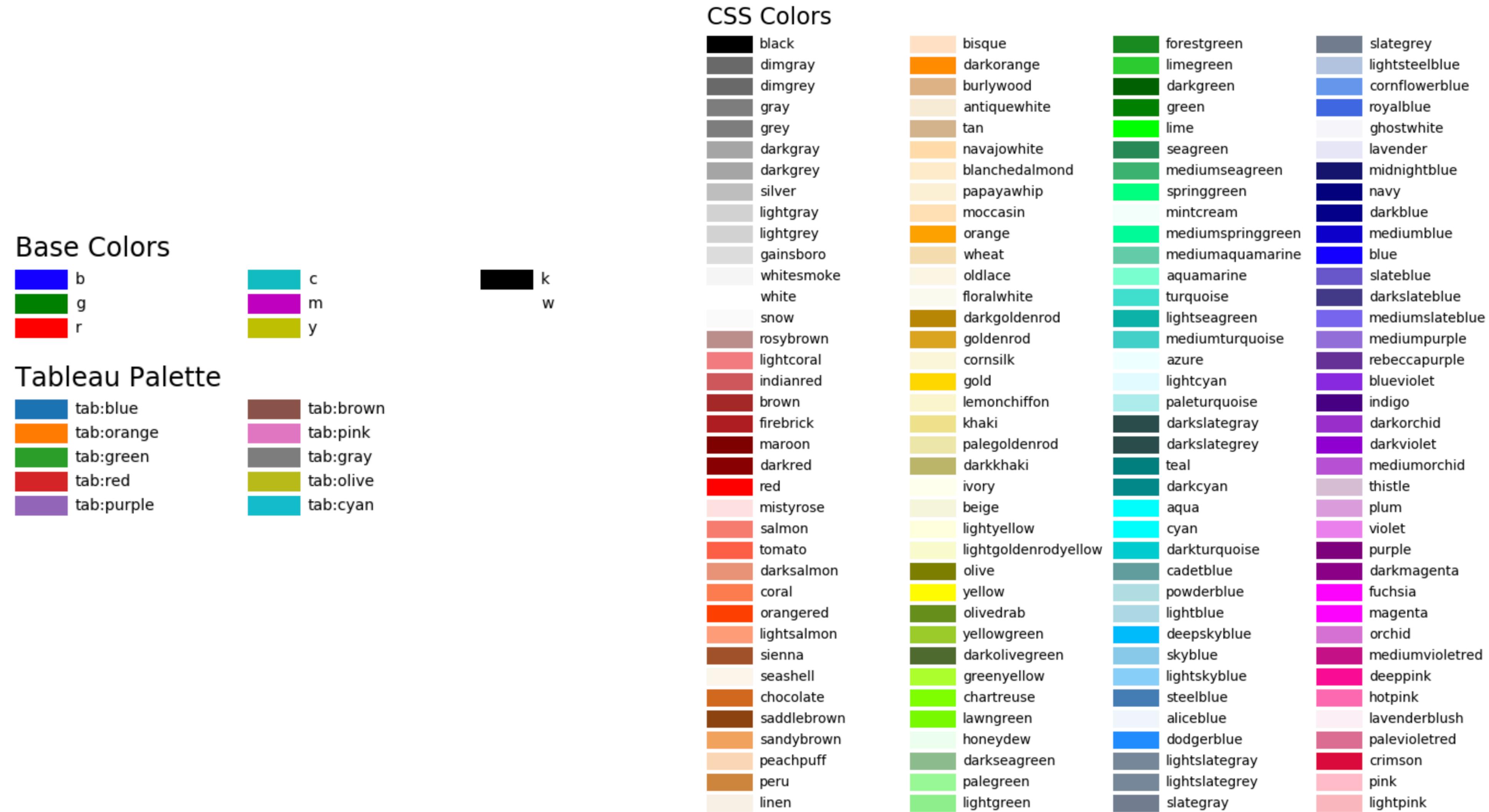
1. Named Colors
2. RGB Colors
3. Colormaps in Matplotlib
4. Discrete Colormaps
5. Continuous Colormaps
6. Colorbar

Lecture_1-07 Colors in Matplotlib

4



1. Named Colors



Lecture_1-07 Colors in Matplotlib

6

1. Named Colors(Base Colors)

```
import matplotlib.pyplot as plt

color_list = ['b', 'g', 'r', 'c', 'm', 'y']

fig, ax = plt.subplots(figsize=(5, 10))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, len(color_list)])

for c_idx, c in enumerate(color_list):
    ax.text(0, c_idx,
            "color='"+c,
            fontsize=20,
            ha='center',
            color=c)
```



Lecture_1-07 Colors in Matplotlib

7

1. Named Colors(tab10 Colors)

```
color_list = ['tab:blue', 'tab:orange',
              'tab:green', 'tab:red',
              'tab:purple', 'tab:brown',
              'tab:pink', 'tab:gray',
              'tab:olive', 'tab:cyan']

fig, ax = plt.subplots(figsize=(5, 10))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, len(color_list)])

for c_idx, c in enumerate(color_list):
    ax.text(0, c_idx,
            "color='"+c,
            fontsize=20,
            ha='center',
            color=c)
```



1. Named Colors(Exercise)

```
title_font_dict = {'fontsize': 30,
                   'color': 'darkblue'}
```

```
label_font_dict = {'fontsize': 20,
                   'color': 'darkblue',
                   'alpha': 0.8}
```

```
fig = plt.figure(figsize=(7, 7),
                  facecolor='powderblue')
```

```
ax = fig.add_subplot(facecolor='powderblue')
```

```
ax.set_title("Fig Title",
             fontdict=title_font_dict)
```

```
ax.set_xlabel("Data Index",
              fontdict=label_font_dict)
```

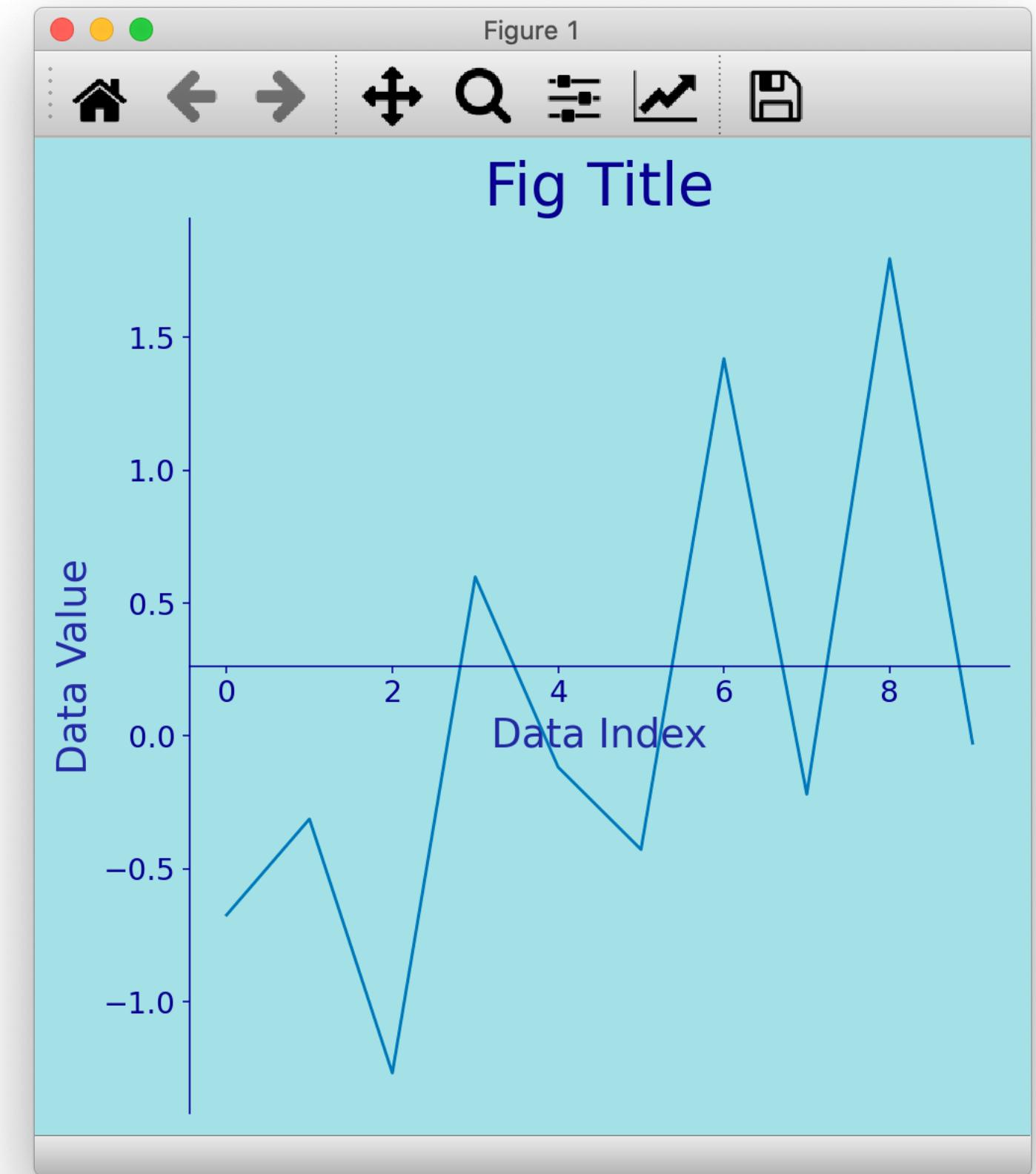
```
ax.set_ylabel("Data Value",
              fontdict=label_font_dict)
```

```
ax.tick_params(labelsize=15,
               colors='darkblue')
```

```
for spine_loc, spine in ax.spines.items():
    if spine_loc in ['left', 'bottom']:
        spine.set_color('darkblue')
    if spine_loc in ['right', 'top']:
        spine.set_visible(False)
    if spine_loc in ['bottom']:
        spine.set_position('center')
```

```
test_np = np.random.normal(0, 1, (10,))
ax.plot(test_np)
```

```
fig.tight_layout()
```



Lecture_1-07 Colors in Matplotlib

9

2. RGB Colors

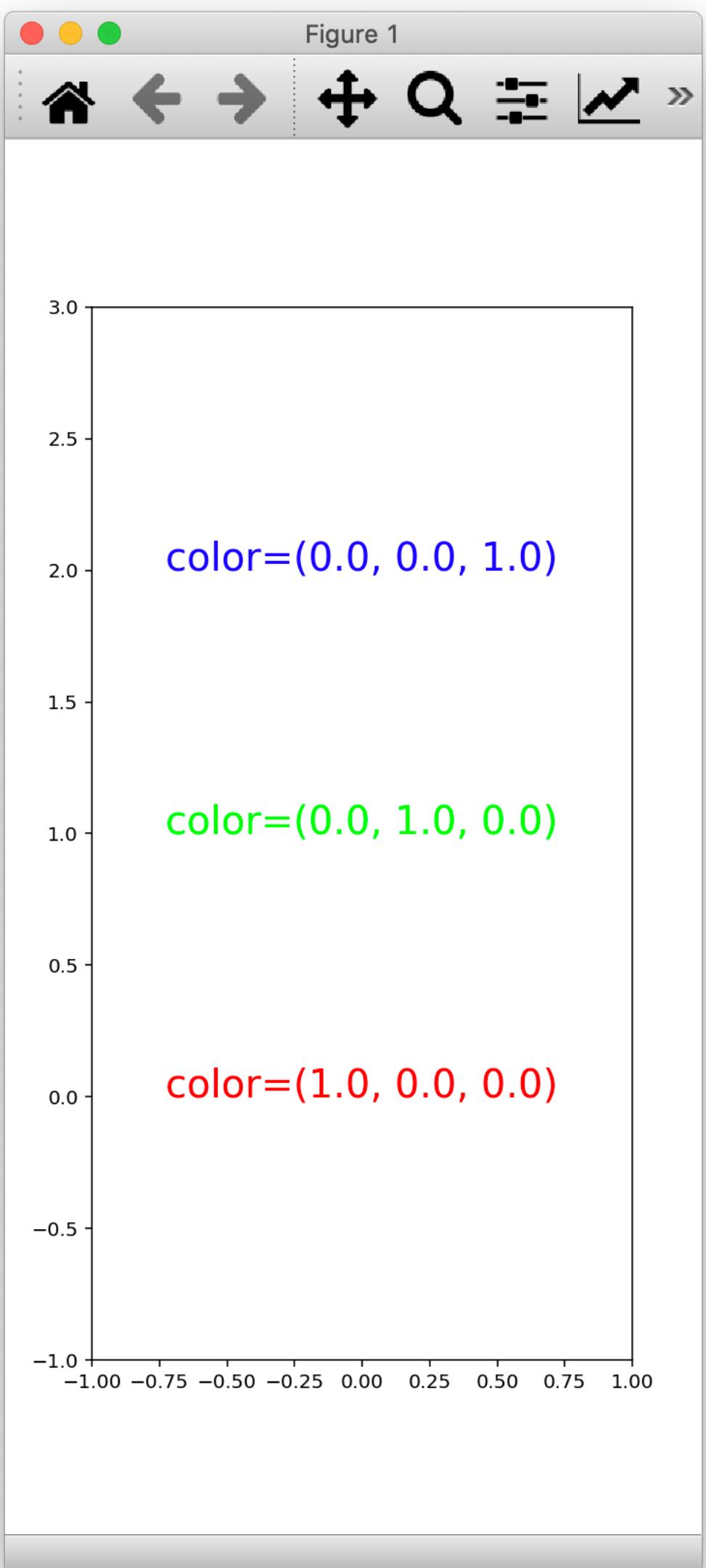
```
import matplotlib.pyplot as plt

color_list = [(1., 0., 0.),
              (0., 1., 0.),
              (0., 0., 1.)]

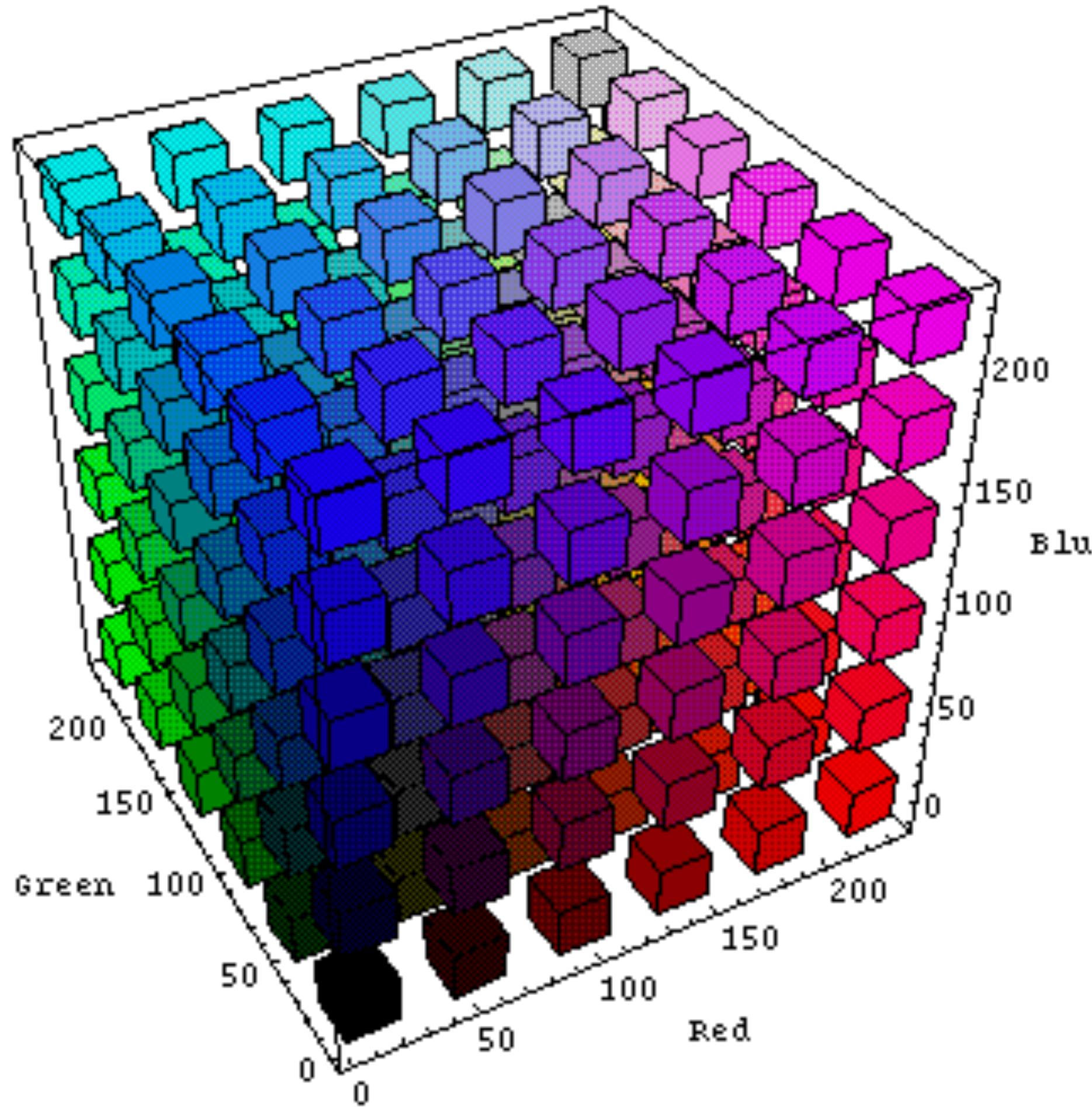
fig, ax = plt.subplots(figsize=(5, 10))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, len(color_list)])

for c_idx, c in enumerate(color_list):
    c_string_list = [c for c in str(c)]
    c_string = ''.join(c_string_list)

    ax.text(0, c_idx,
            "color="+c_string,
            fontsize=20,
            ha='center',
            color=c)
```



2. RGB Colors(Byte to Float)

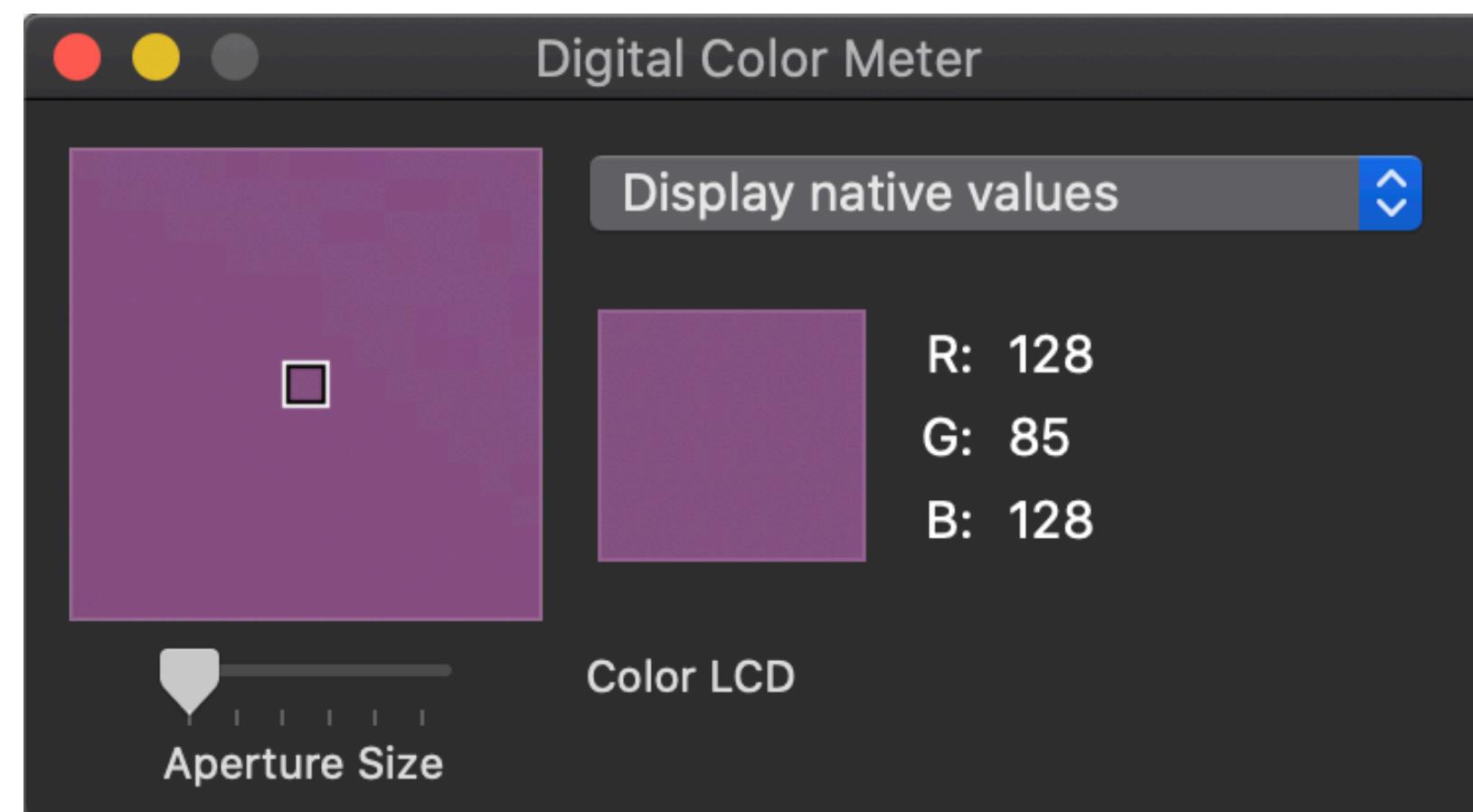
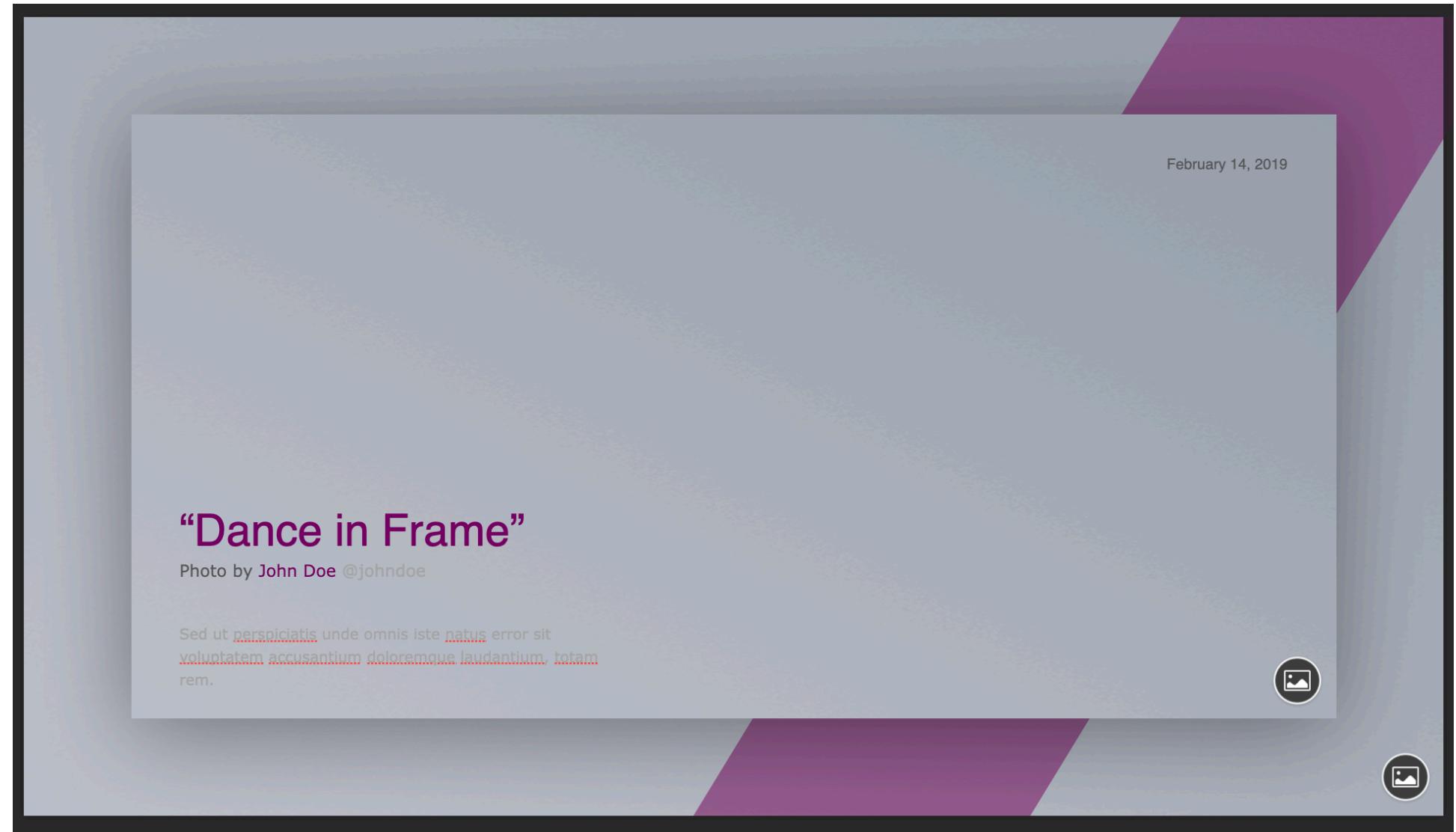
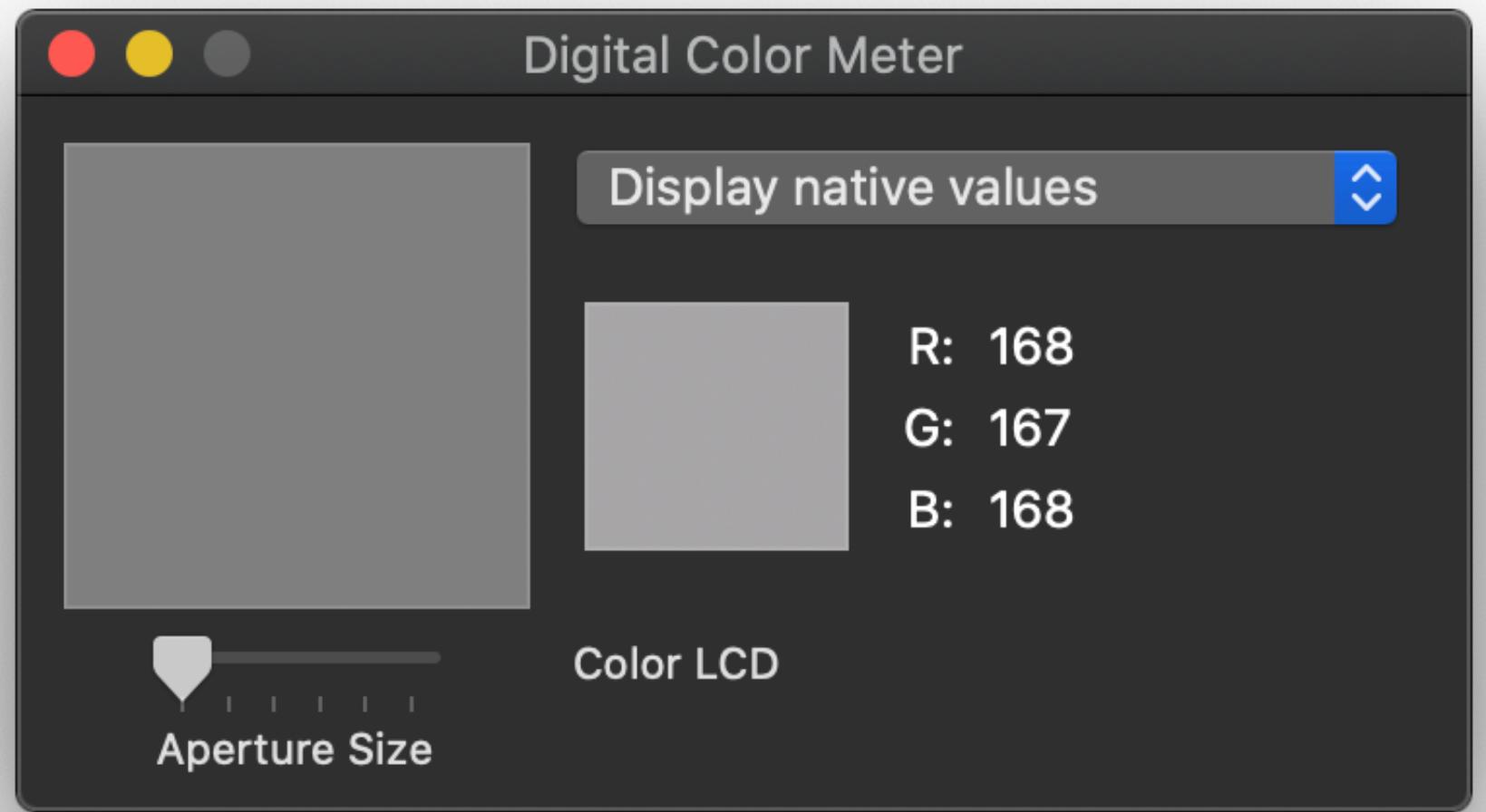


```
def convert255tol(*RGB255):
    color1 = [c/255 for c in RGB255]
    return color1

print(convert255tol(255, 200, 10))
[1.0, 0.7843137254901961, 0.0392156862745098]

fig, ax = plt.subplots(figsize=(7, 7))
ax.set_title("Fig Title",
             fontsize=30,
             color=convert255tol(200, 100, 150))
```

2. RGB Colors(Practical Tip)



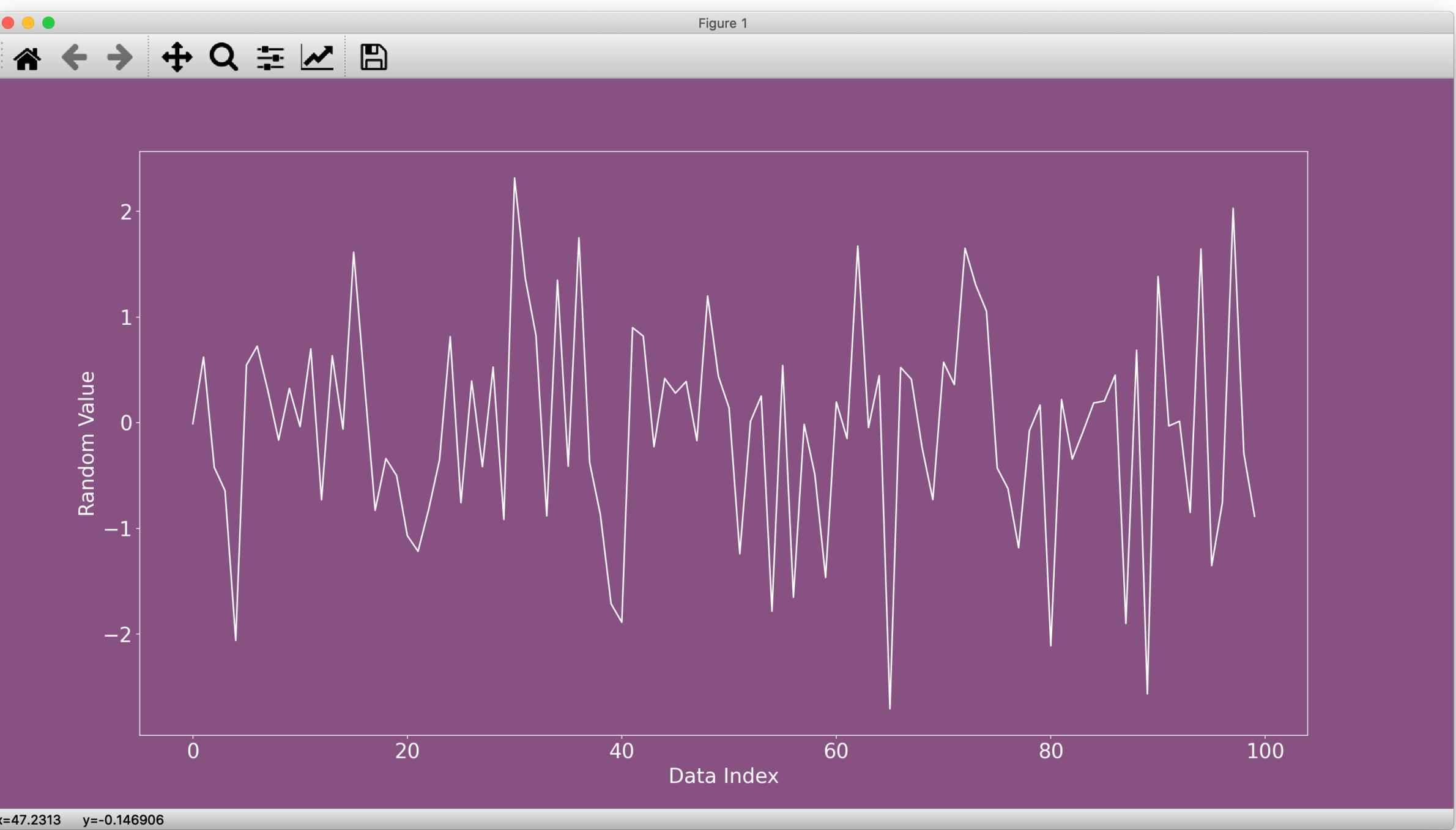
2. RGB Colors(Practical Tip)

```
def convert255tol(*color_255):
    color_1 = [c/255 for c in color_255]
    return color_1

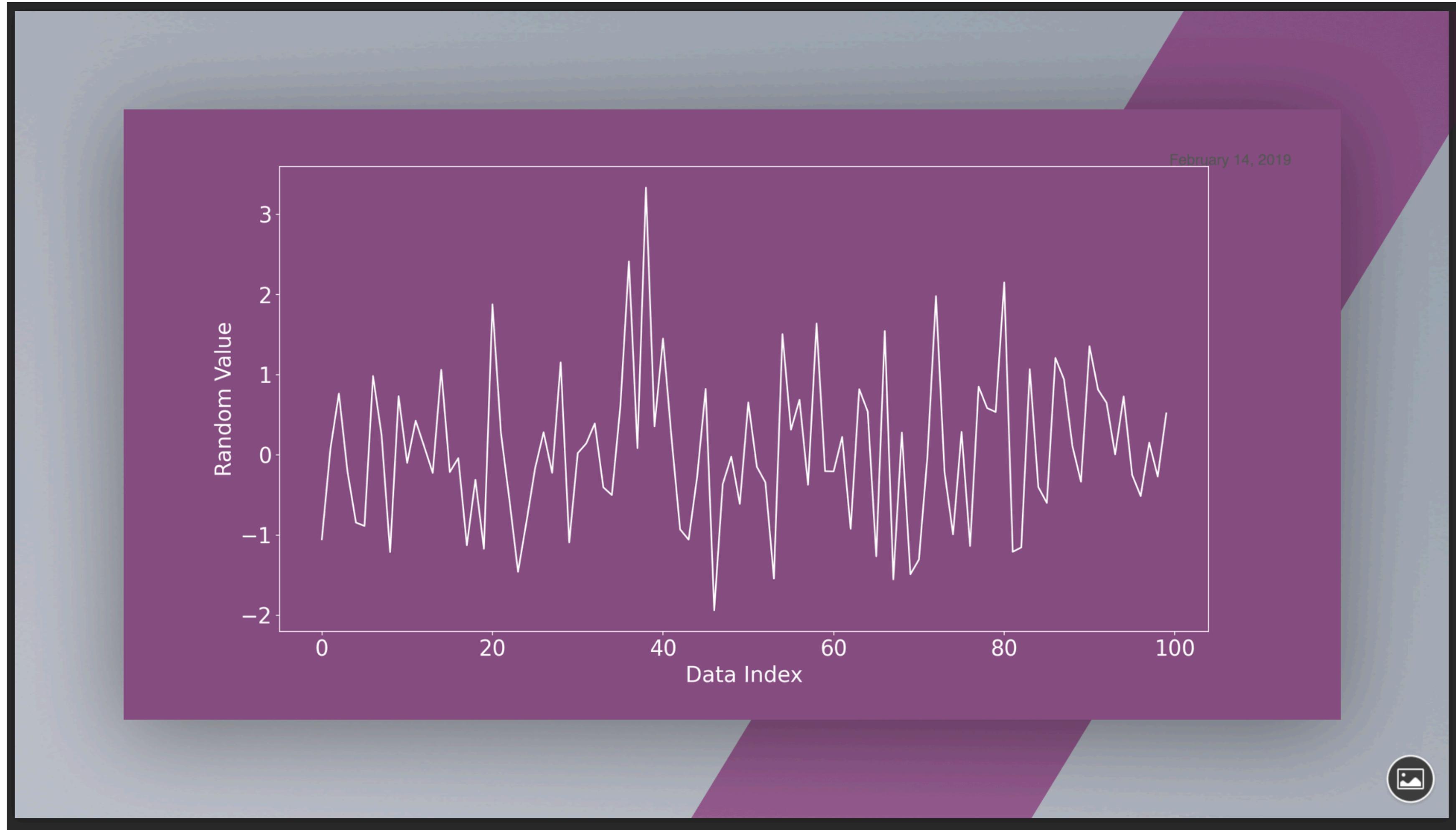
ppt_color = convert255tol(128, 85, 128)
label_font_dict = {'color': 'w',
                   'fontsize': 20}

fig = plt.figure(figsize=(20,10),
                 facecolor=ppt_color)
ax = fig.add_subplot(facecolor=ppt_color)
ax.plot(np.random.normal(0, 1, (100,)),
        color='w')
ax.set_xlabel("Data Index",
              fontdict=label_font_dict)
ax.set_ylabel("Random Value",
              fontdict=label_font_dict)
ax.tick_params(colors='w',
               labelsize=20)

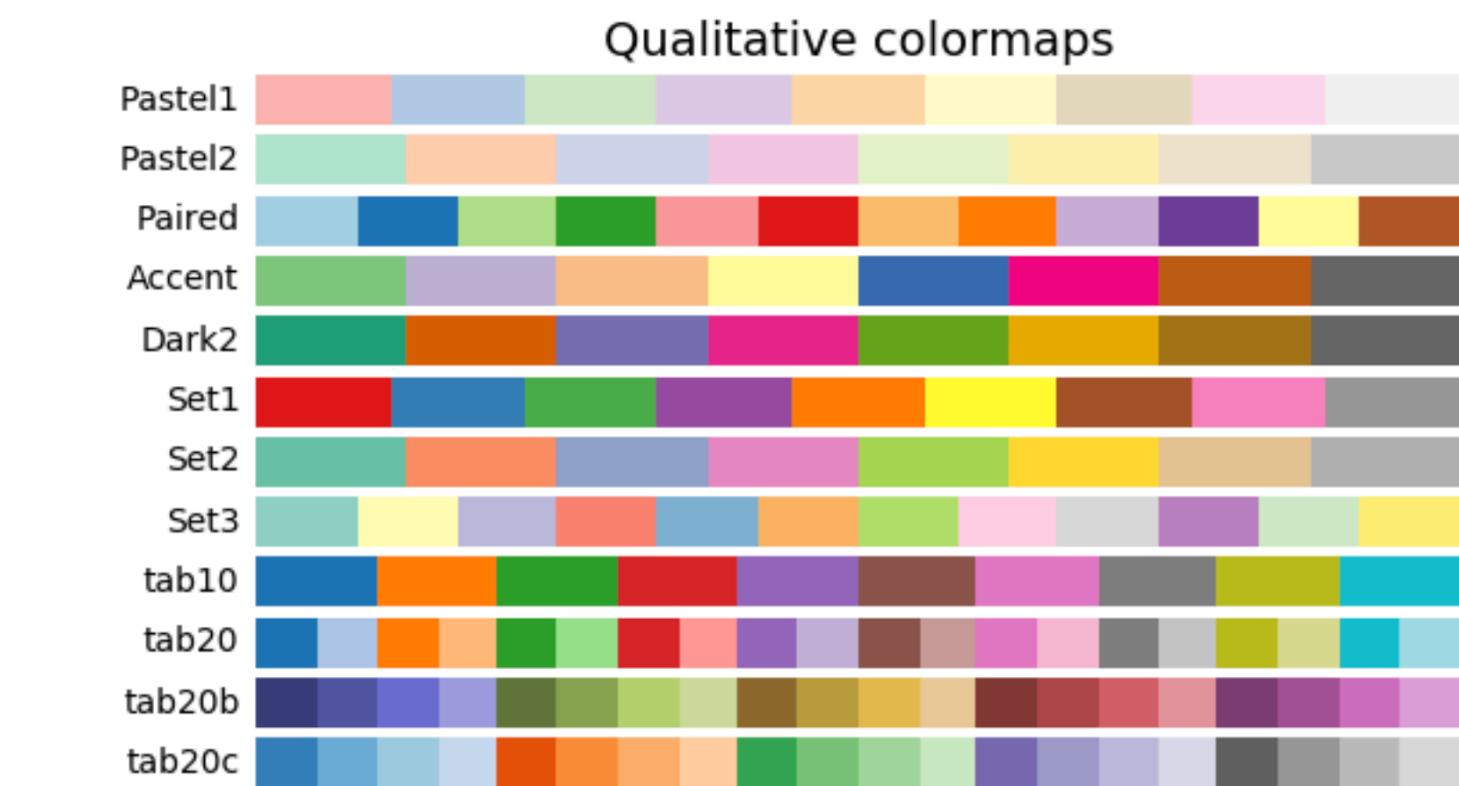
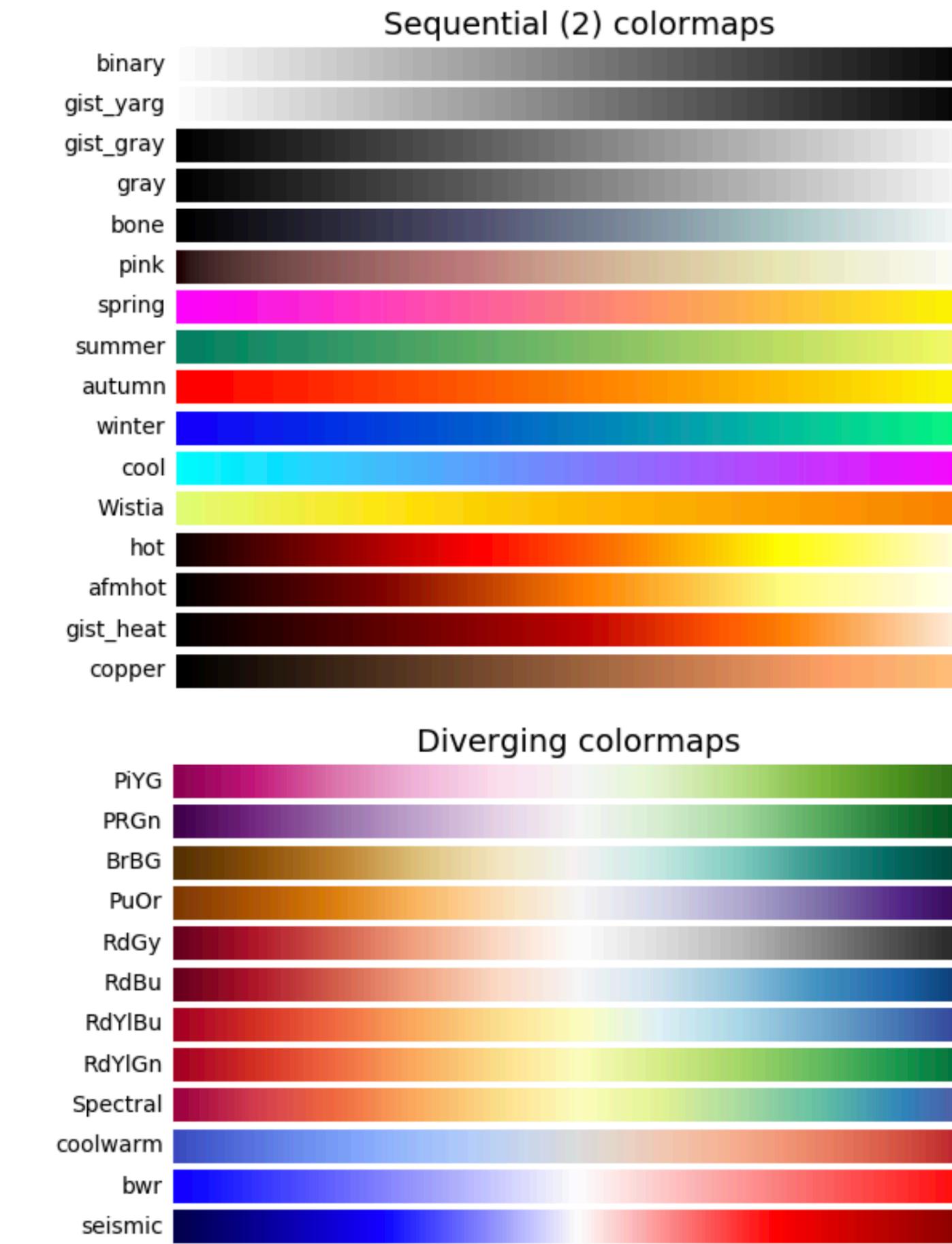
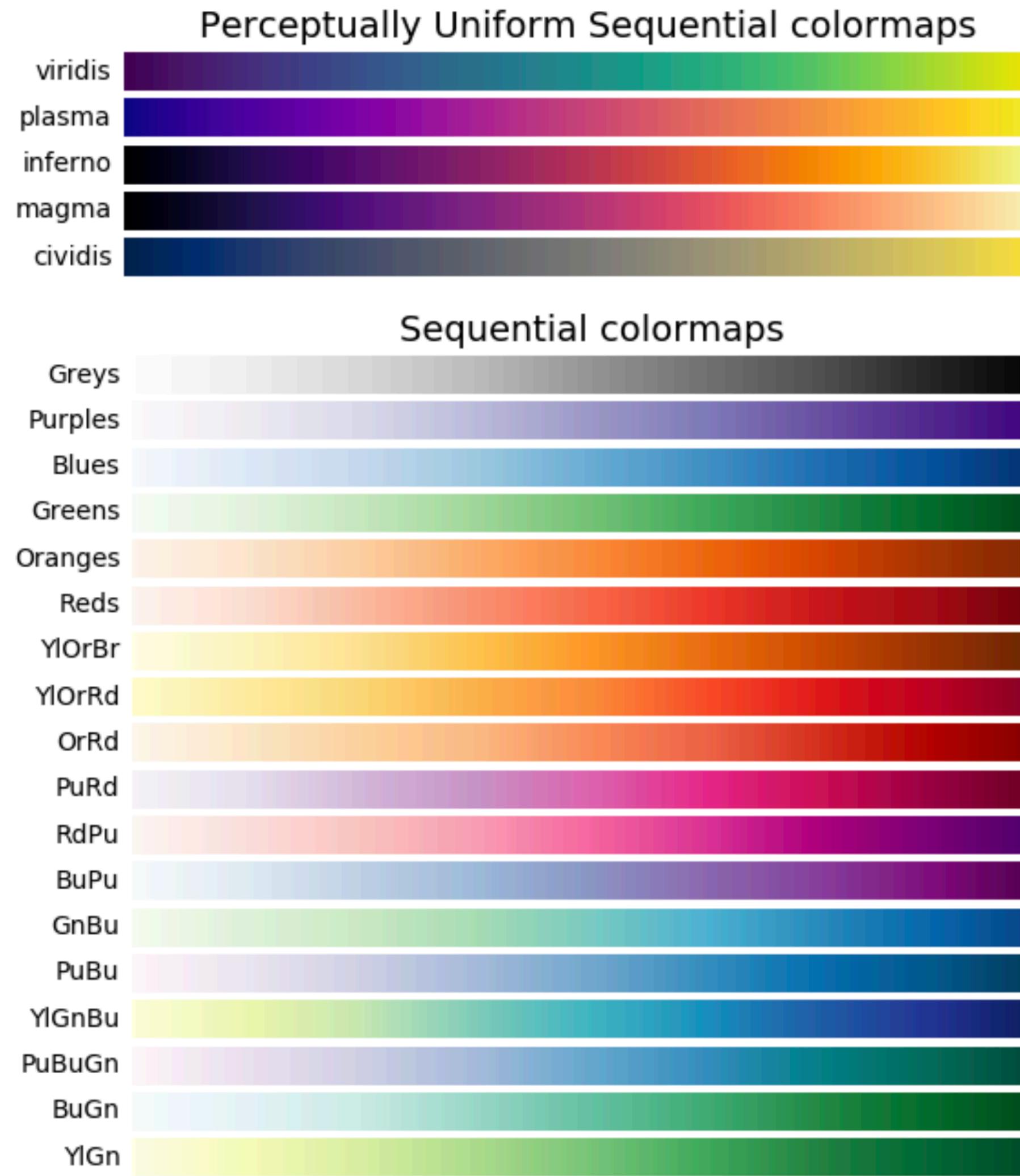
for spine_loc, spine in ax.spines.items():
    spine.set_color('w')
fig.subplots_adjust(bottom=0.1, top=0.9,
                    left=0.1, right=0.9)
```



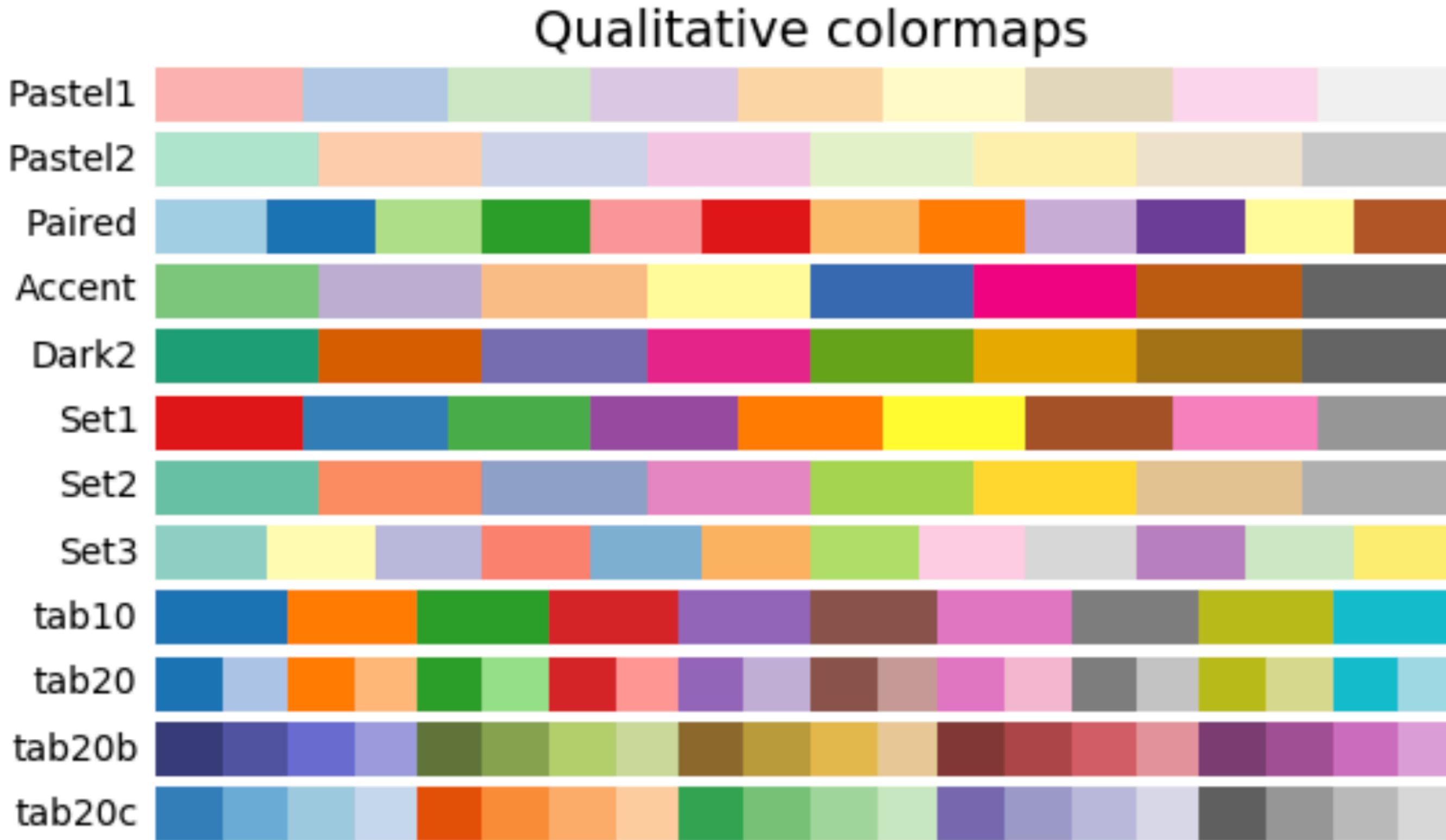
2. RGB Colors(Practical Tip)



3. Colormaps in Matplotlib



4. Discrete Colormaps



4. Discrete Colormaps(Getting Colormaps)

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

cmap = cm.get_cmap('tab10')
print(cmap.colors)

((0.12156862745098039, 0.4666666666666667, 0.7058823529411765),
 (1.0, 0.4980392156862745, 0.054901960784313725),
 (0.17254901960784313, 0.6274509803921569, 0.17254901960784313),
 (0.8392156862745098, 0.15294117647058825, 0.1568627450980392),
 (0.5803921568627451, 0.403921568627451, 0.7411764705882353),
 (0.5490196078431373, 0.33725490196078434, 0.29411764705882354),
 (0.8901960784313725, 0.4666666666666667, 0.7607843137254902),
 (0.4980392156862745, 0.4980392156862745, 0.4980392156862745),
 (0.7372549019607844, 0.7411764705882353, 0.1333333333333333),
 (0.09019607843137255, 0.7450980392156863, 0.8117647058823529))
```

4. Discrete Colormaps(Getting Colors)

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

cmap = cm.get_cmap('tab10')
n_color = len(cmap.colors)

for c_idx in range(n_color):
    print(cmap(c_idx)) (0.12156862745098039, 0.4666666666666667, 0.7058823529411765, 1.0)
                    (1.0, 0.4980392156862745, 0.054901960784313725, 1.0)
                    (0.17254901960784313, 0.6274509803921569, 0.17254901960784313, 1.0)
                    (0.8392156862745098, 0.15294117647058825, 0.1568627450980392, 1.0)
                    (0.5803921568627451, 0.403921568627451, 0.7411764705882353, 1.0)
                    (0.5490196078431373, 0.33725490196078434, 0.29411764705882354, 1.0)
                    (0.8901960784313725, 0.4666666666666667, 0.7607843137254902, 1.0)
                    (0.4980392156862745, 0.4980392156862745, 0.4980392156862745, 1.0)
                    (0.7372549019607844, 0.7411764705882353, 0.1333333333333333, 1.0)
                    (0.09019607843137255, 0.7450980392156863, 0.8117647058823529, 1.0)
```

4. Discrete Colormaps(Number of Colors)

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

cmap = cm.get_cmap('tab10')
n_color = len(cmap.colors)

for c_idx in range(n_color+3):
    print(cmap(c_idx))
```

(0.12156862745098039, 0.4666666666666667, 0.7058823529411765, 1.0)
(1.0, 0.4980392156862745, 0.054901960784313725, 1.0)
(0.17254901960784313, 0.6274509803921569, 0.17254901960784313, 1.0)
(0.8392156862745098, 0.15294117647058825, 0.1568627450980392, 1.0)
(0.5803921568627451, 0.403921568627451, 0.7411764705882353, 1.0)
(0.5490196078431373, 0.33725490196078434, 0.29411764705882354, 1.0)
(0.8901960784313725, 0.4666666666666667, 0.7607843137254902, 1.0)
(0.4980392156862745, 0.4980392156862745, 0.4980392156862745, 1.0)
(0.7372549019607844, 0.7411764705882353, 0.1333333333333333, 1.0)
(0.09019607843137255, 0.7450980392156863, 0.8117647058823529, 1.0)
(0.09019607843137255, 0.7450980392156863, 0.8117647058823529, 1.0)
(0.09019607843137255, 0.7450980392156863, 0.8117647058823529, 1.0)
(0.09019607843137255, 0.7450980392156863, 0.8117647058823529, 1.0)

4. Discrete Colormaps(Aplications)

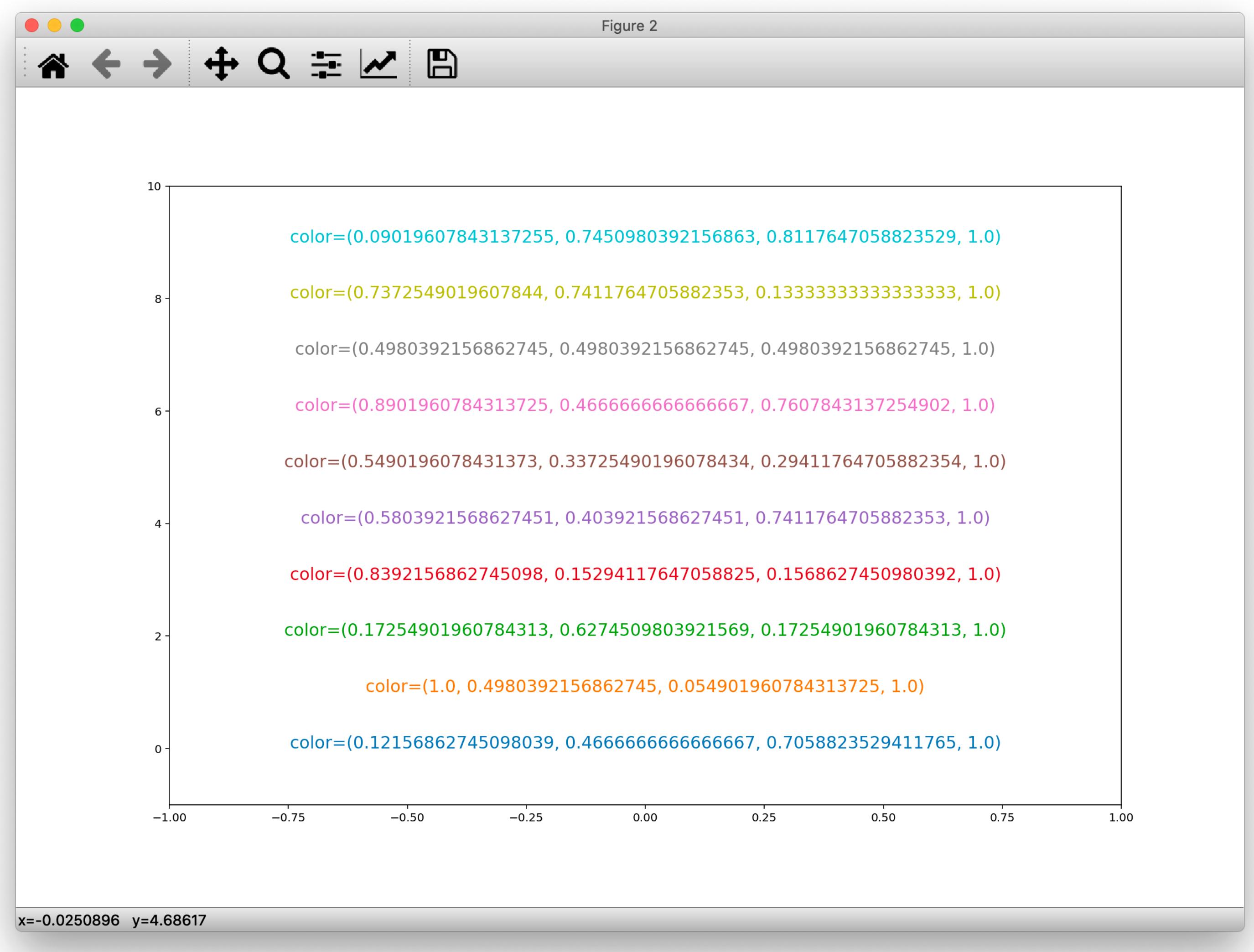
```
import matplotlib.pyplot as plt
import matplotlib.cm as cm

cmap = cm.get_cmap('tab10')
n_color = len(cmap.colors)

fig, ax = plt.subplots(figsize=(15, 10))
ax.set_xlim([-1, 1])
ax.set_ylim([-1, n_color])

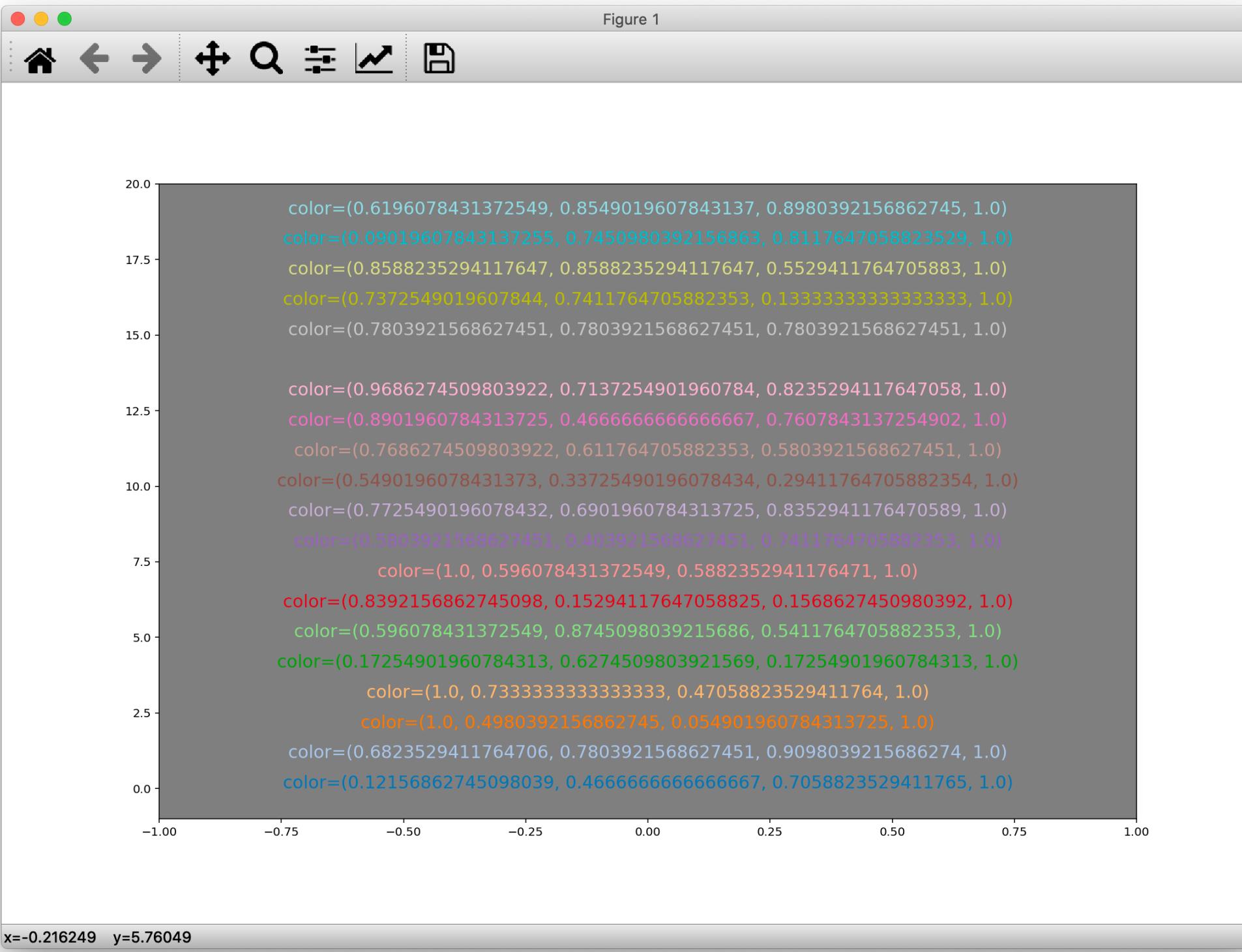
for c_idx in range(n_color):
    color = cmap(c_idx)
    c_string_list = [c for c in str(color)]
    c_string = ''.join(c_string_list)

    ax.text(0, c_idx,
            "color=" + c_string,
            fontsize=15,
            ha='center',
            color=color)
```

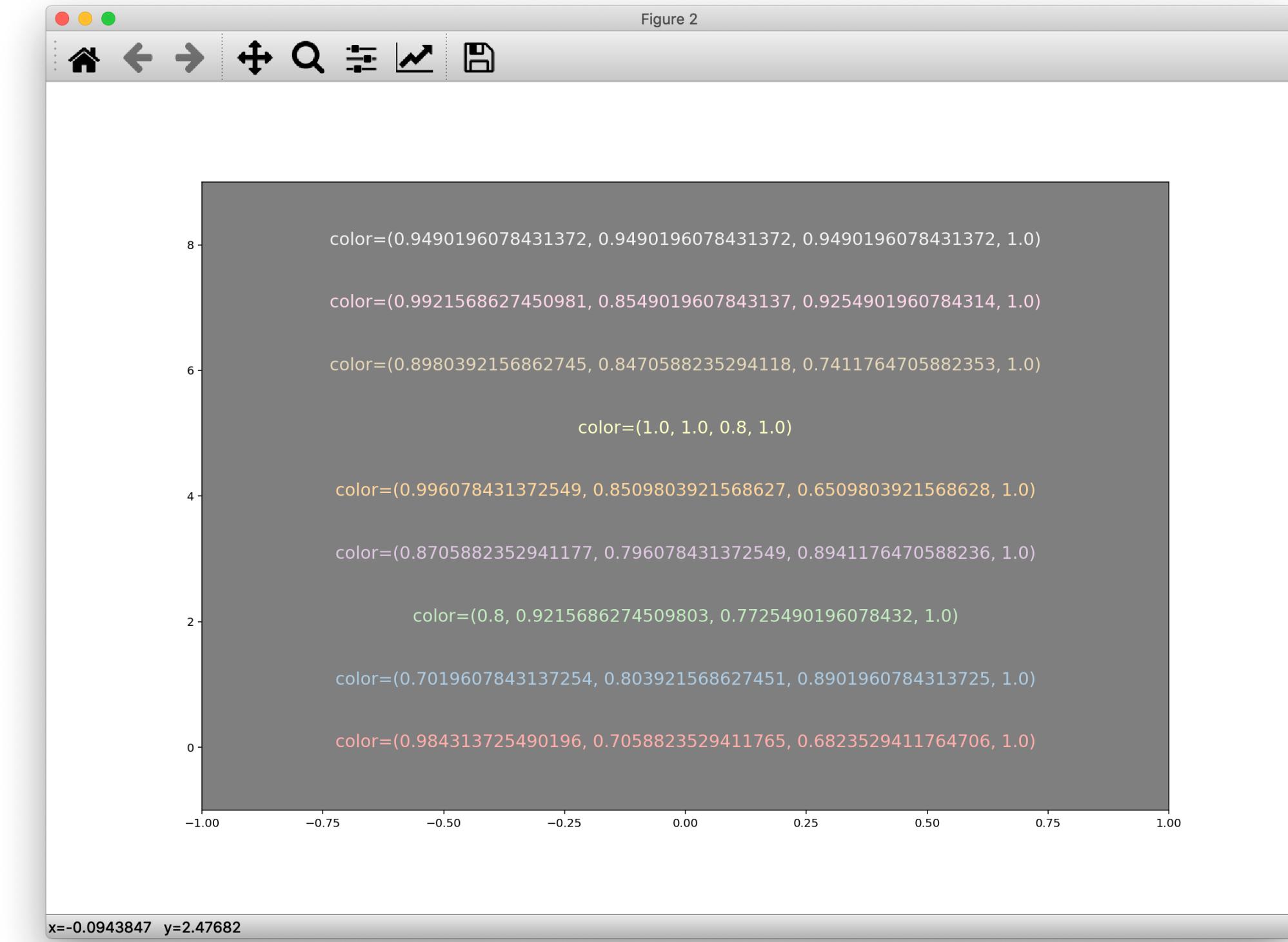


4. Discrete Colormaps(Various Colormaps)

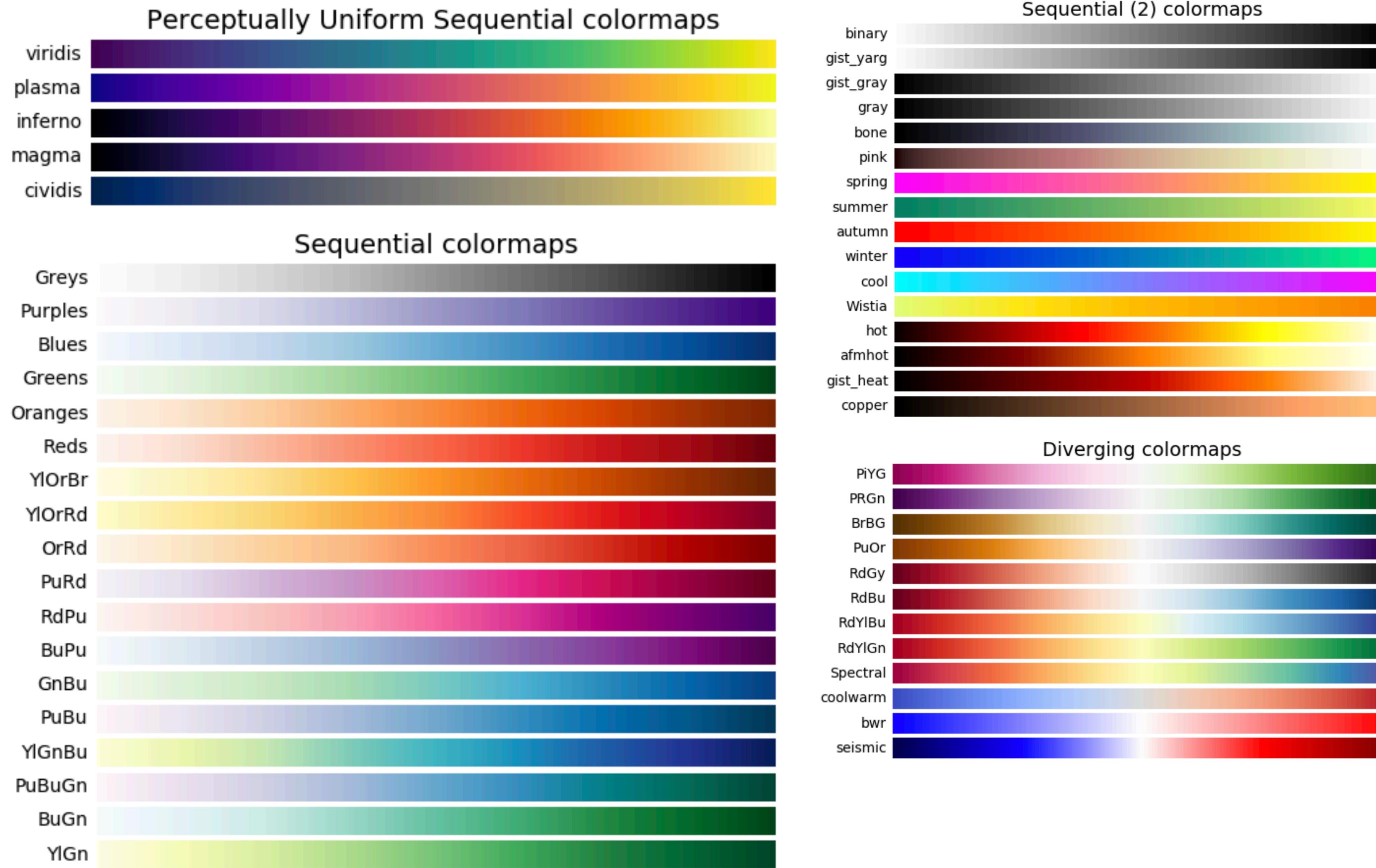
```
cmap = cm.get_cmap('tab20')
```



```
cmap = cm.get_cmap('Pastell1')
```



5. Continuous Colormaps



5. Continuous Colormaps(lut Argument)

rainbow

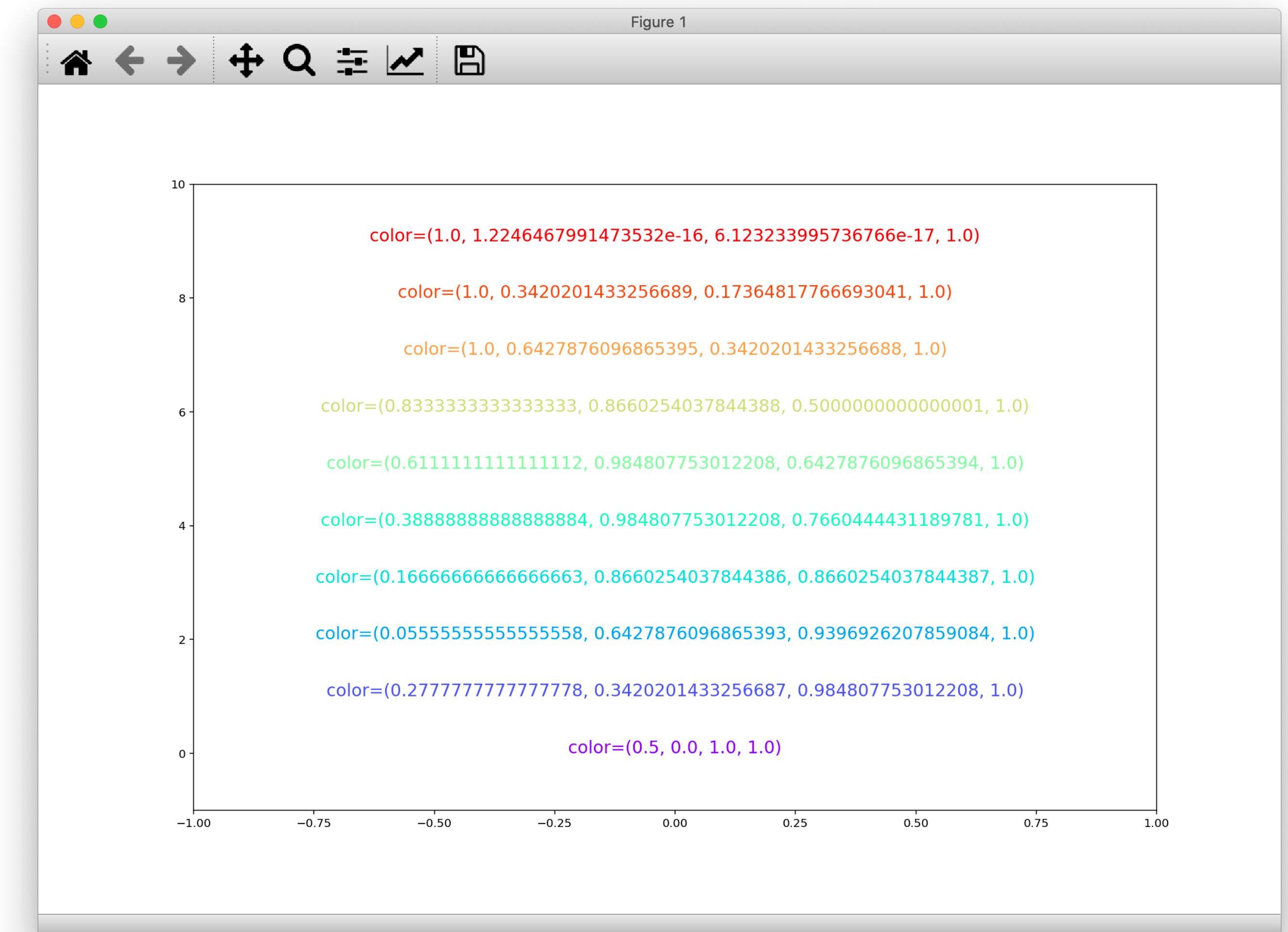


```
n_color = 10
cmap = cm.get_cmap('rainbow', lut=n_color)

for c_idx in range(n_color):
    print(cmap(c_idx)) (0.5, 0.0, 1.0, 1.0)
                    (0.2777777777777778, 0.3420201433256687, 0.984807753012208, 1.0)
                    (0.05555555555555558, 0.6427876096865393, 0.9396926207859084, 1.0)
                    (0.16666666666666663, 0.8660254037844386, 0.8660254037844387, 1.0)
                    (0.3888888888888884, 0.984807753012208, 0.7660444431189781, 1.0)
                    (0.6111111111111112, 0.984807753012208, 0.6427876096865394, 1.0)
                    (0.8333333333333333, 0.8660254037844388, 0.5000000000000001, 1.0)
                    (1.0, 0.6427876096865395, 0.3420201433256688, 1.0)
                    (1.0, 0.3420201433256689, 0.17364817766693041, 1.0)
                    (1.0, 1.2246467991473532e-16, 6.123233995736766e-17, 1.0)
```

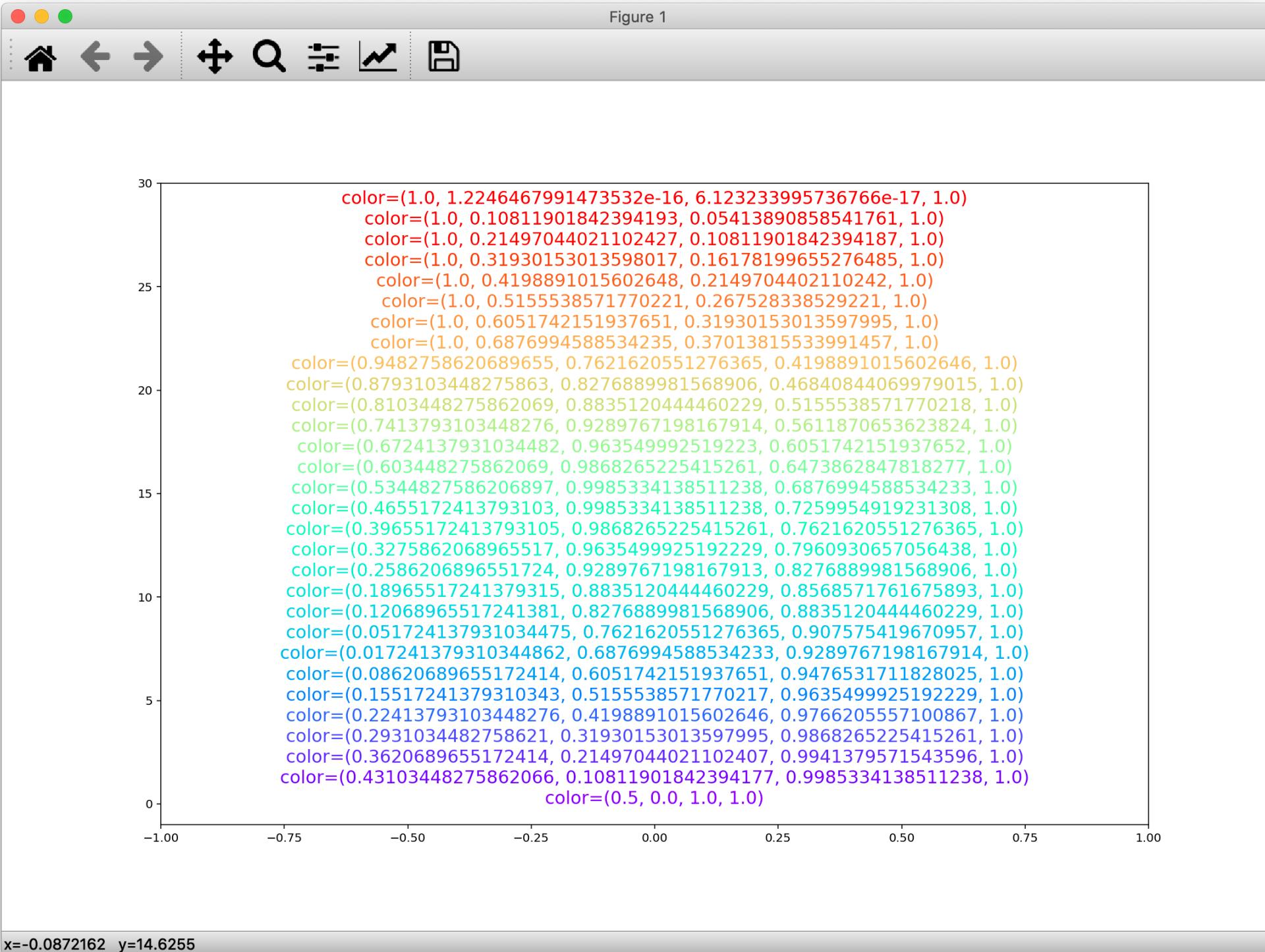
5. Continuous Colormaps(lut Argument)

```
n_color = 10  
cmap = cm.get_cmap('rainbow', lut=n_color)  
  
fig, ax = plt.subplots(figsize=(15, 10))  
ax.set_xlim([-1, 1])  
ax.set_ylim([-1, n_color])  
  
for c_idx in range(n_color):  
    color = cmap(c_idx)  
    c_string_list = [c for c in str(color)]  
    c_string = ''.join(c_string_list)  
  
    ax.text(0, c_idx,  
            "color='"+c_string,  
            fontsize=15,  
            ha='center',  
            color=color)
```

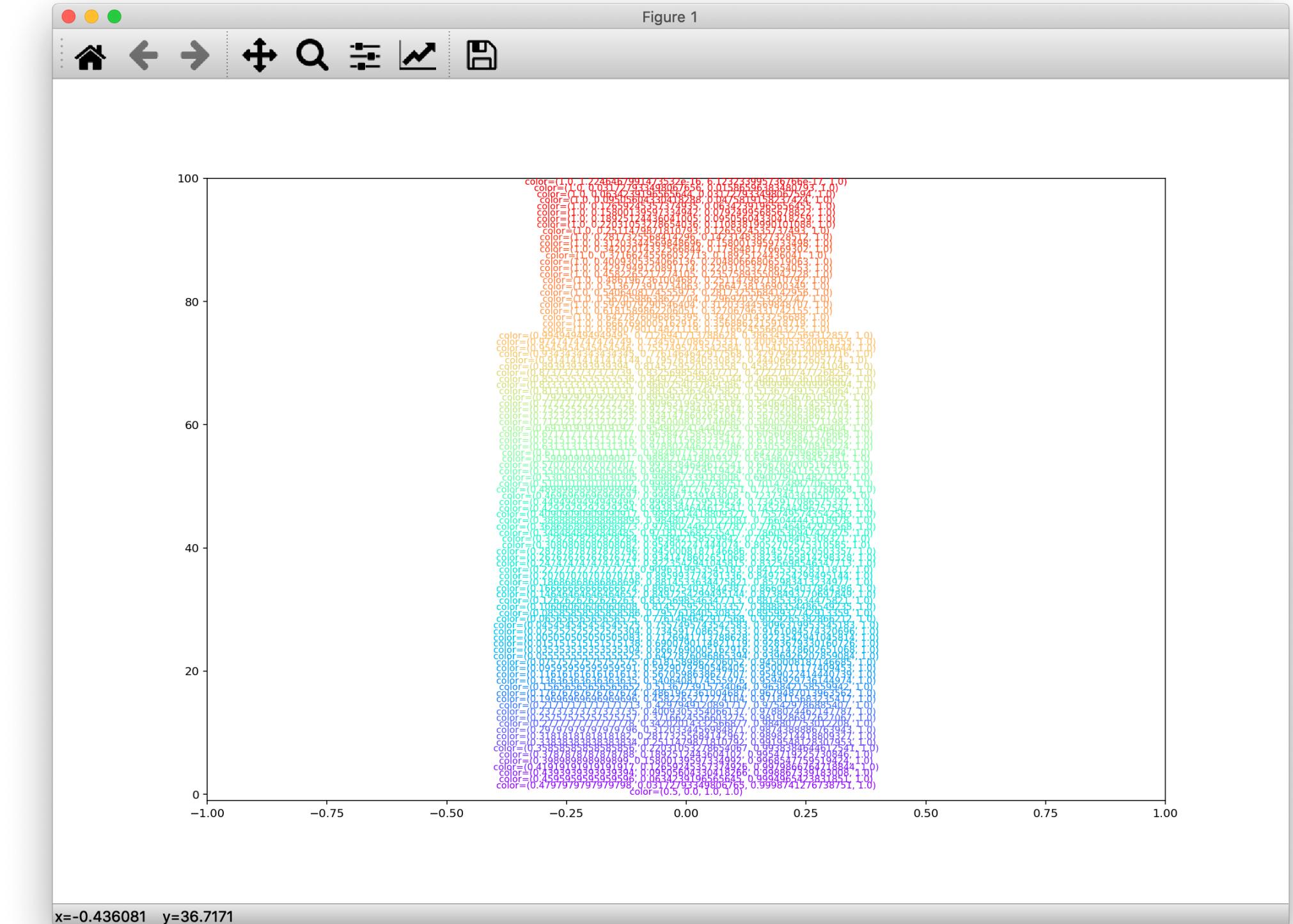


5. Continuous Colormaps(lut Argument)

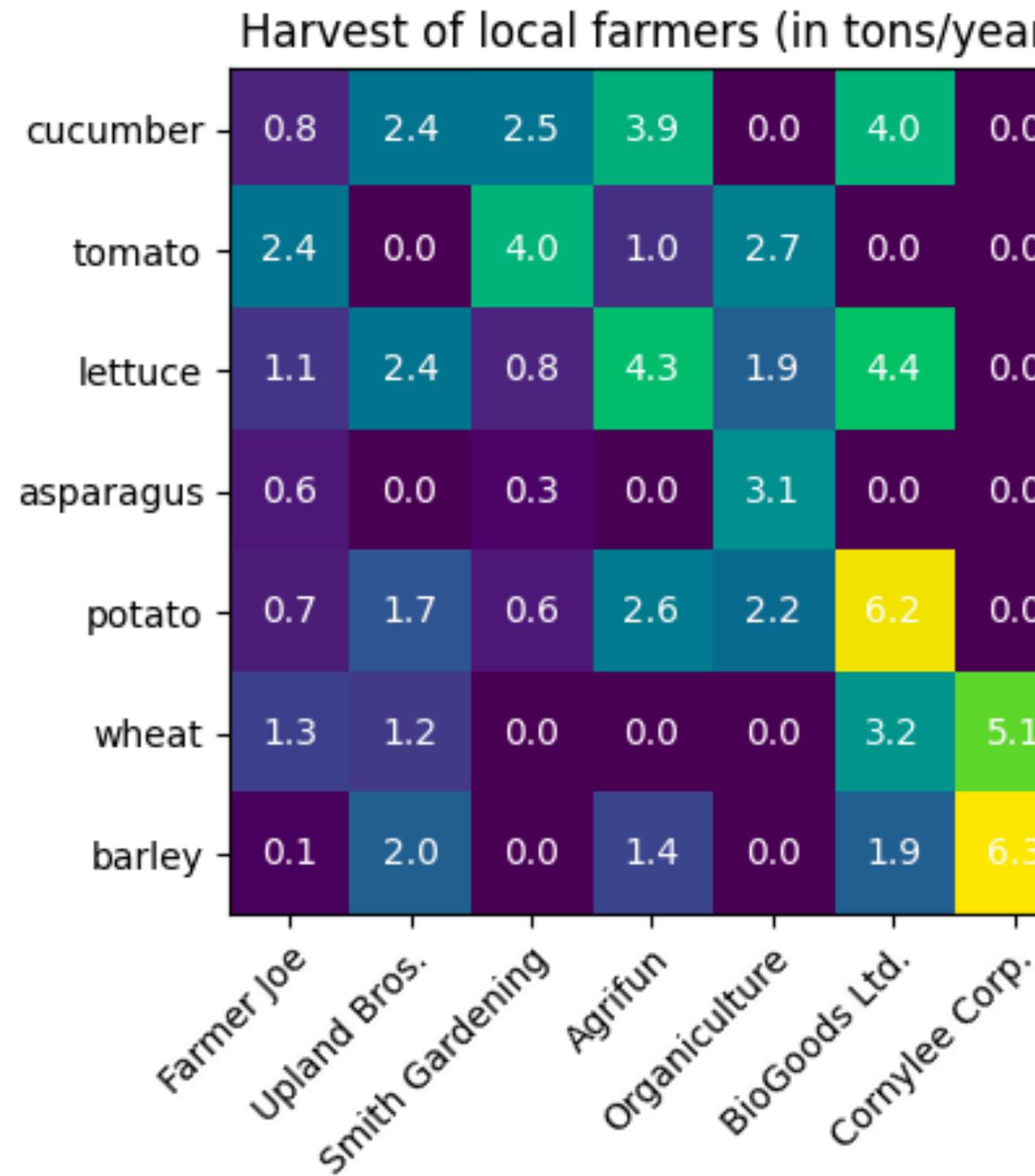
```
n_color = 30
cmap = cm.get_cmap('rainbow', lut=n_color)
```



```
n_color = 100
cmap = cm.get_cmap('rainbow', lut=n_color)
```



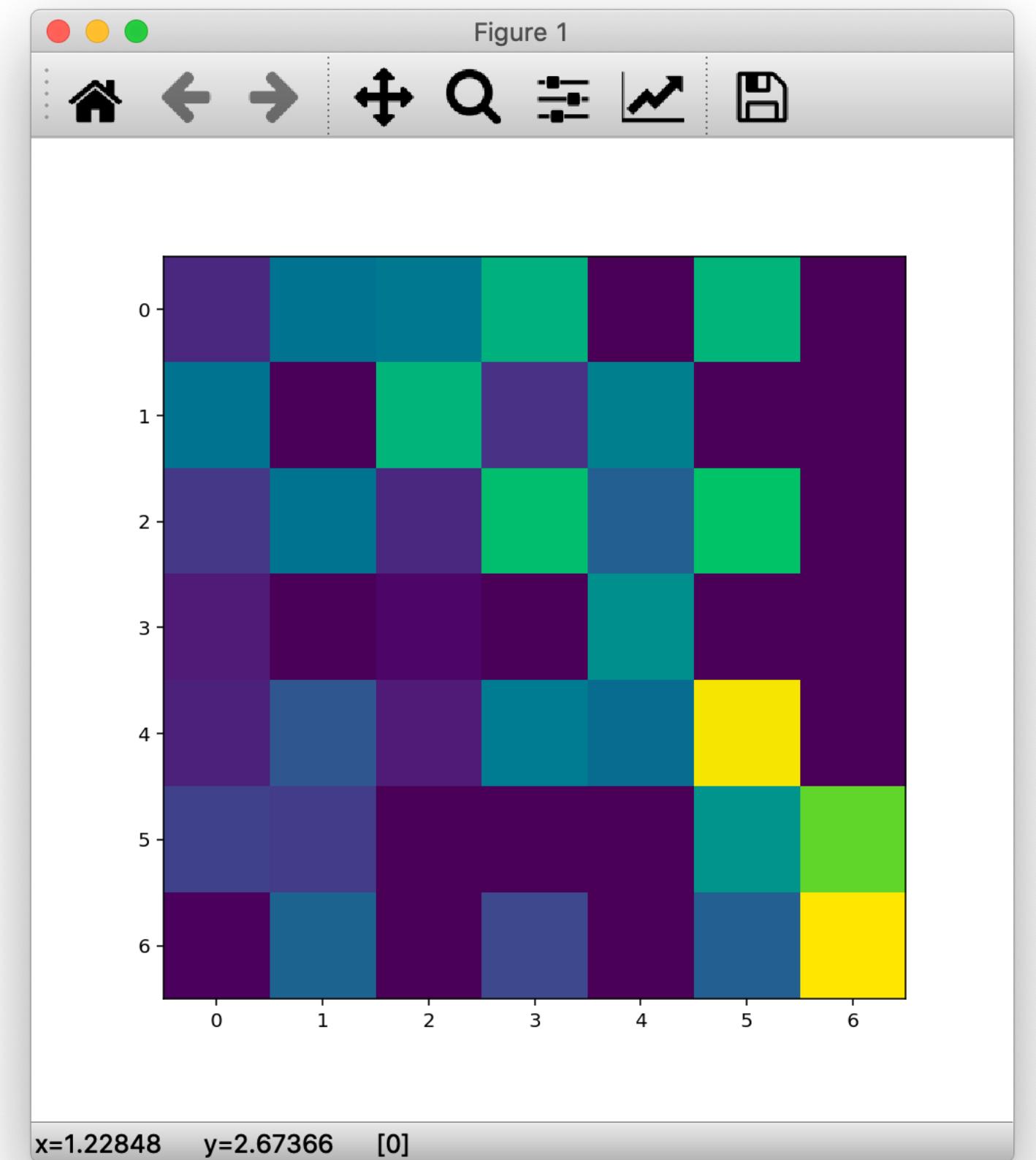
5. Continuous Colormaps(Auto Colormaps)



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

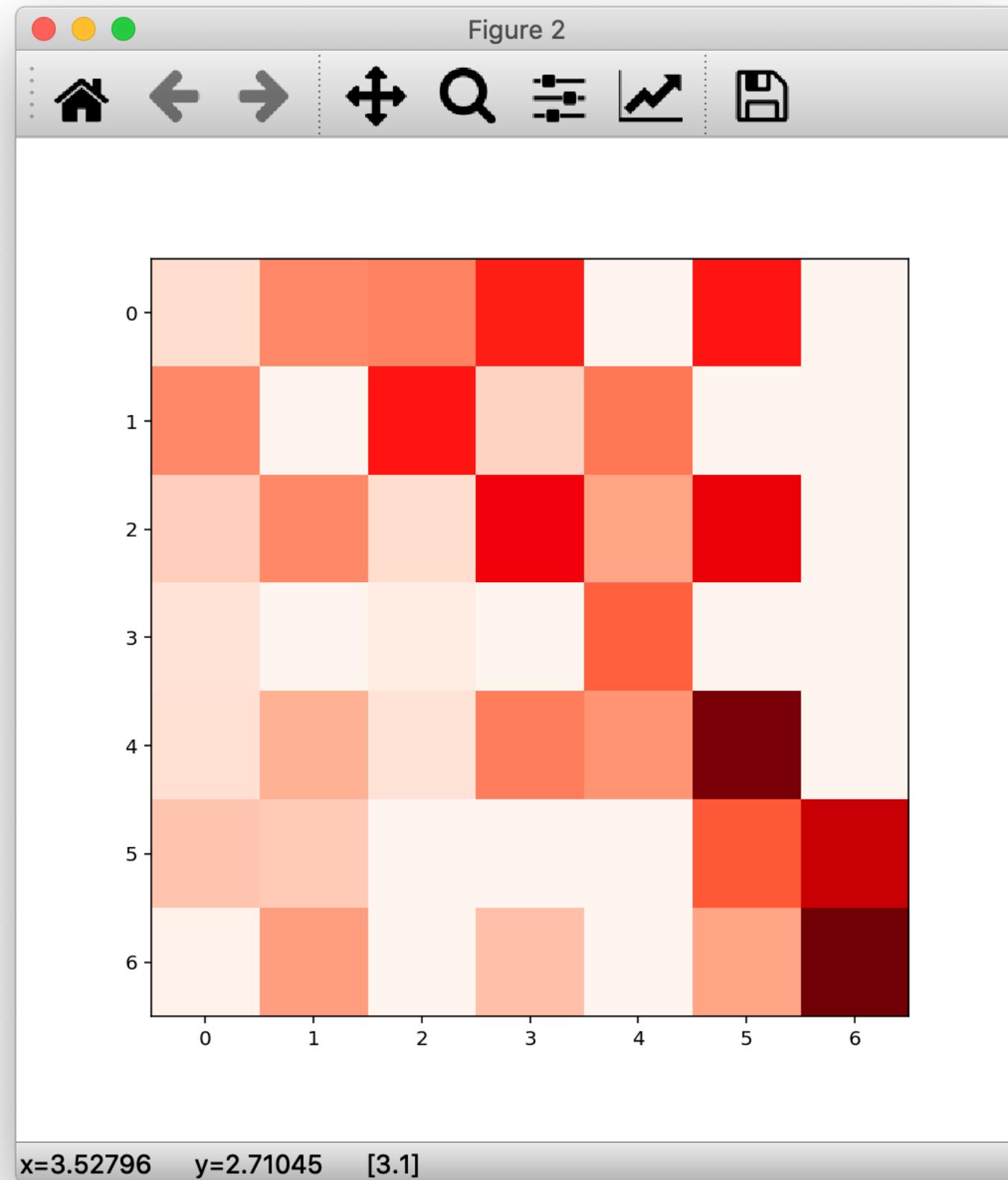
harvest = np.array([[0.8, 2.4, 2.5, 3.9, 0.0, 4.0, 0.0],
                   [2.4, 0.0, 4.0, 1.0, 2.7, 0.0, 0.0],
                   [1.1, 2.4, 0.8, 4.3, 1.9, 4.4, 0.0],
                   [0.6, 0.0, 0.3, 0.0, 3.1, 0.0, 0.0],
                   [0.7, 1.7, 0.6, 2.6, 2.2, 6.2, 0.0],
                   [1.3, 1.2, 0.0, 0.0, 0.0, 3.2, 5.1],
                   [0.1, 2.0, 0.0, 1.4, 0.0, 1.9, 6.3]])

fig, ax = plt.subplots(figsize=(7, 7))
im = ax.imshow(harvest)
```

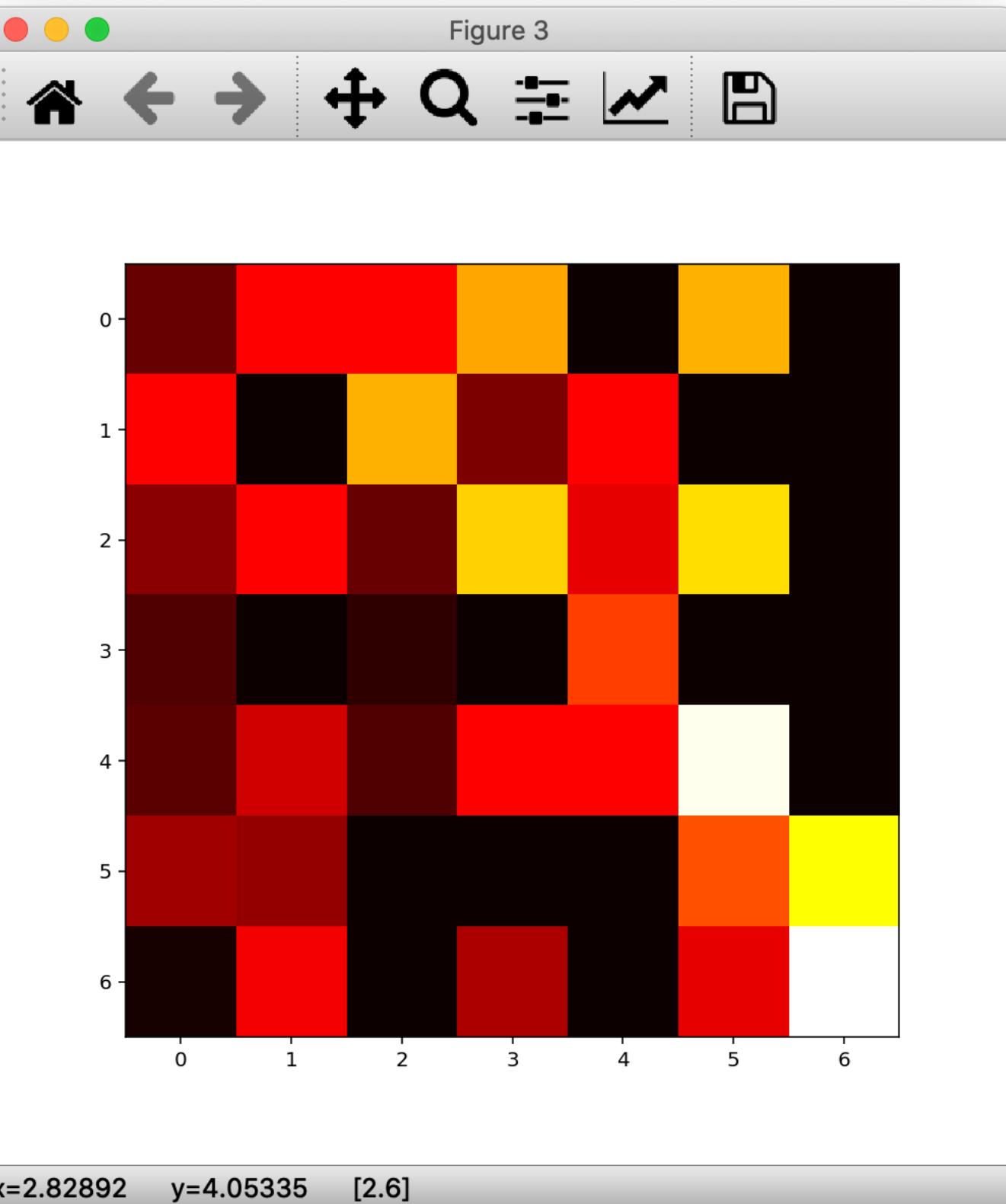


5. Continuous Colormaps(Auto Colormaps)

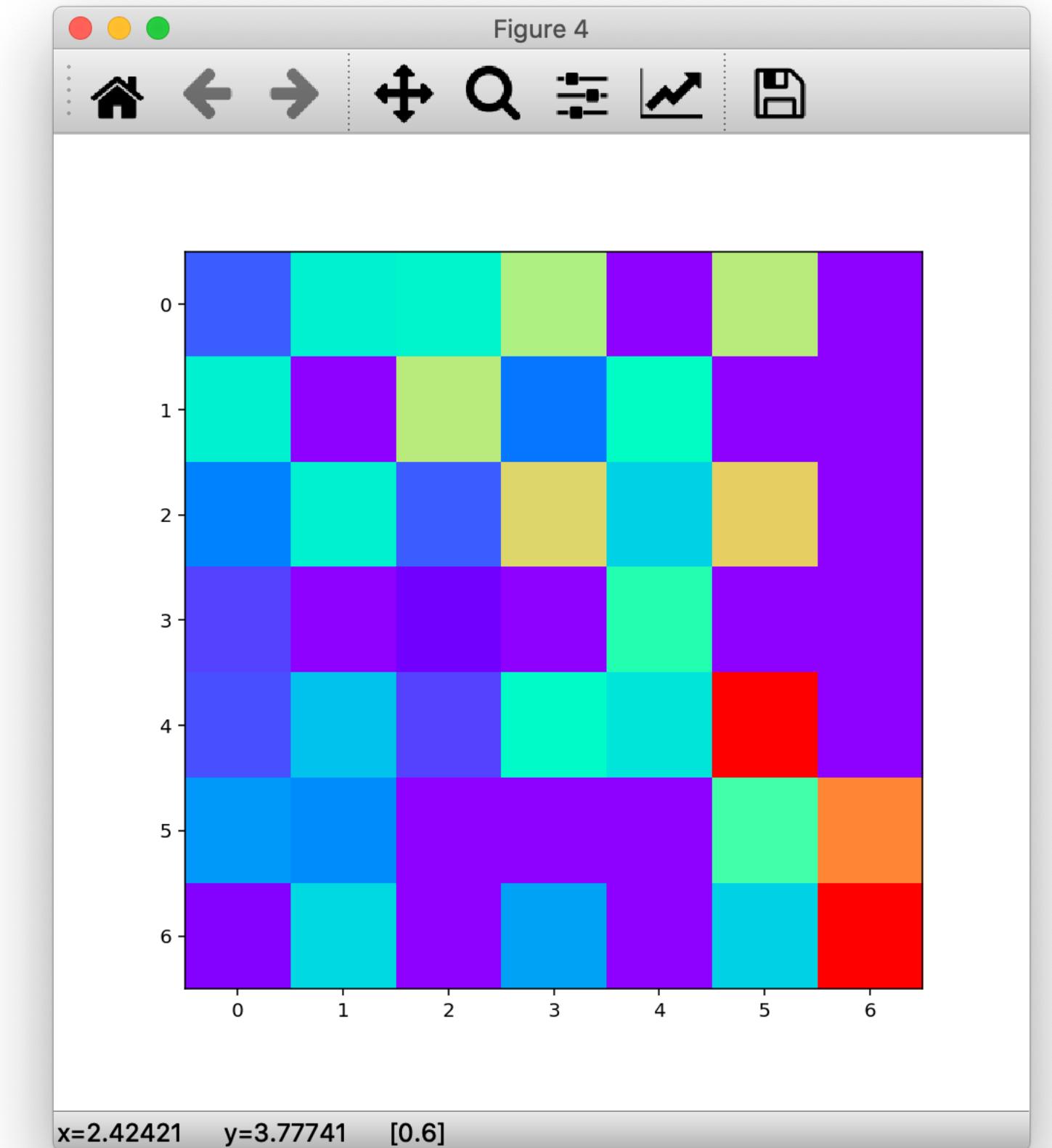
```
im = ax.imshow(harvest,  
                cmap='Reds')
```



```
im = ax.imshow(harvest,  
                cmap='hot')
```



```
im = ax.imshow(harvest,  
                cmap='rainbow')
```



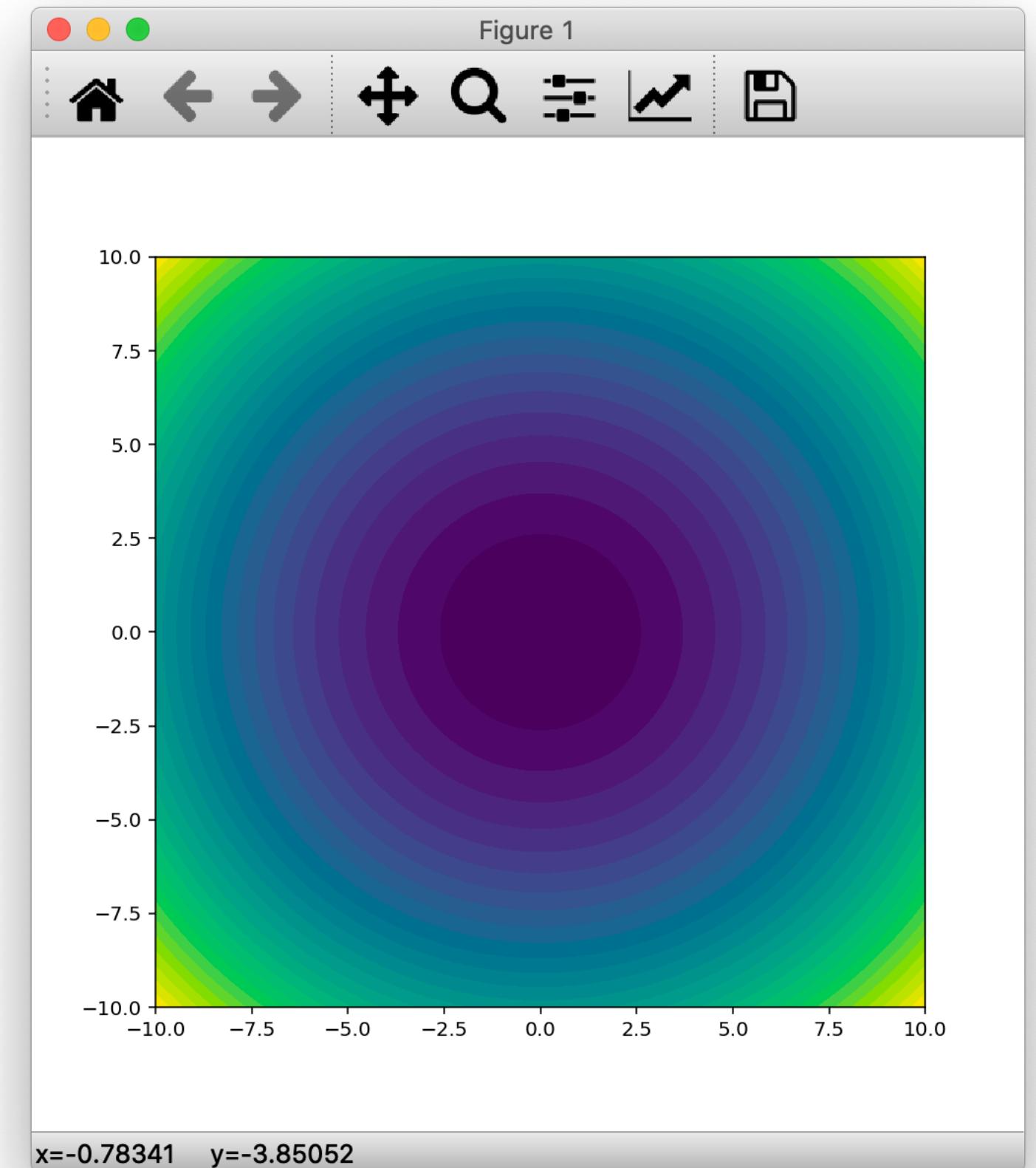
5. Continuous Colormaps(Auto Colormaps)

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

x = np.linspace(-10, 10, 100)
y = x

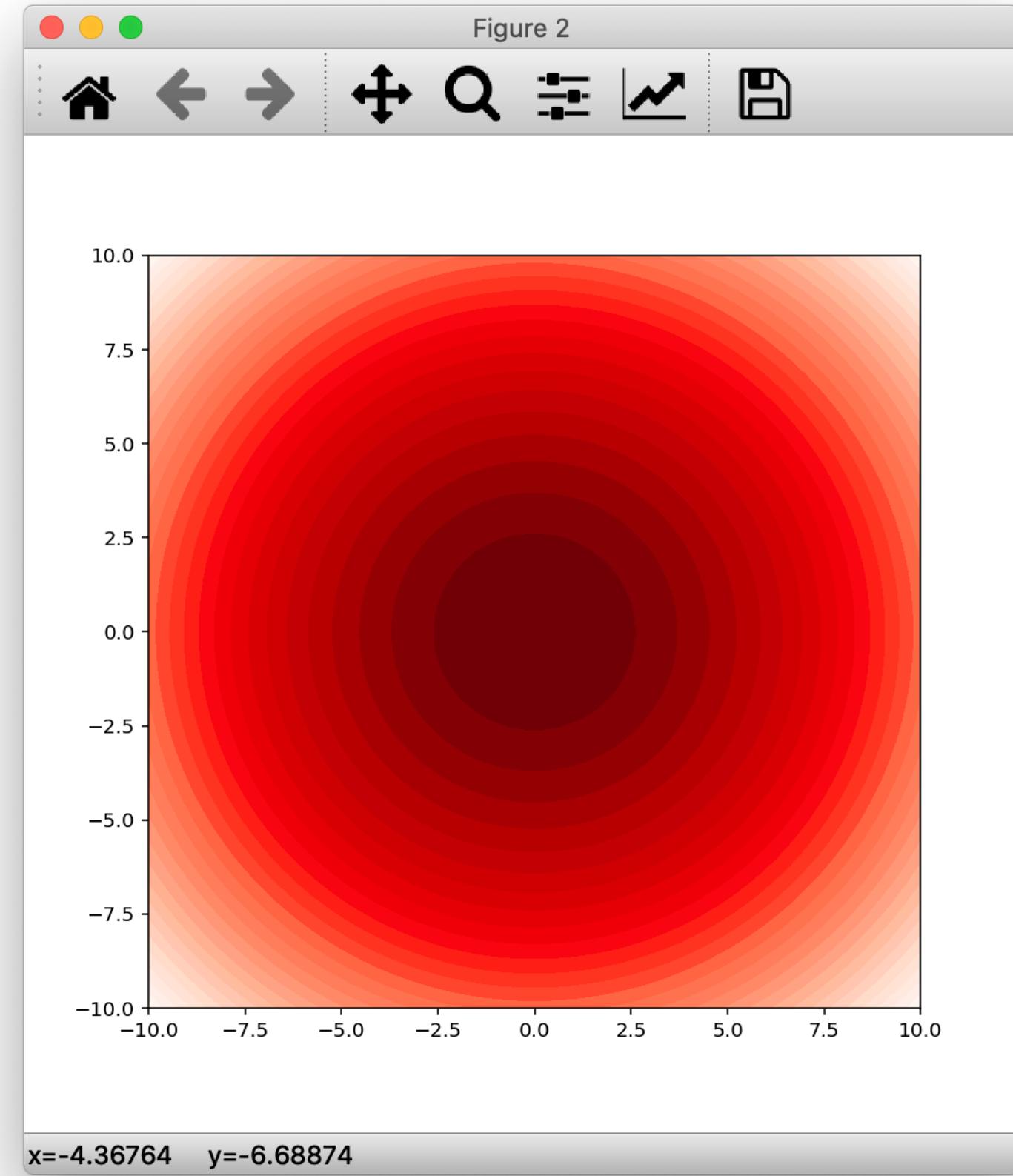
X, Y = np.meshgrid(x, y)
Z = np.power(X, 2) + np.power(Y, 2)

levels = np.linspace(np.min(Z), np.max(Z), 30)
fig, ax = plt.subplots(figsize=(7, 7))
ax.contourf(X, Y, Z,
            levels=levels)
```

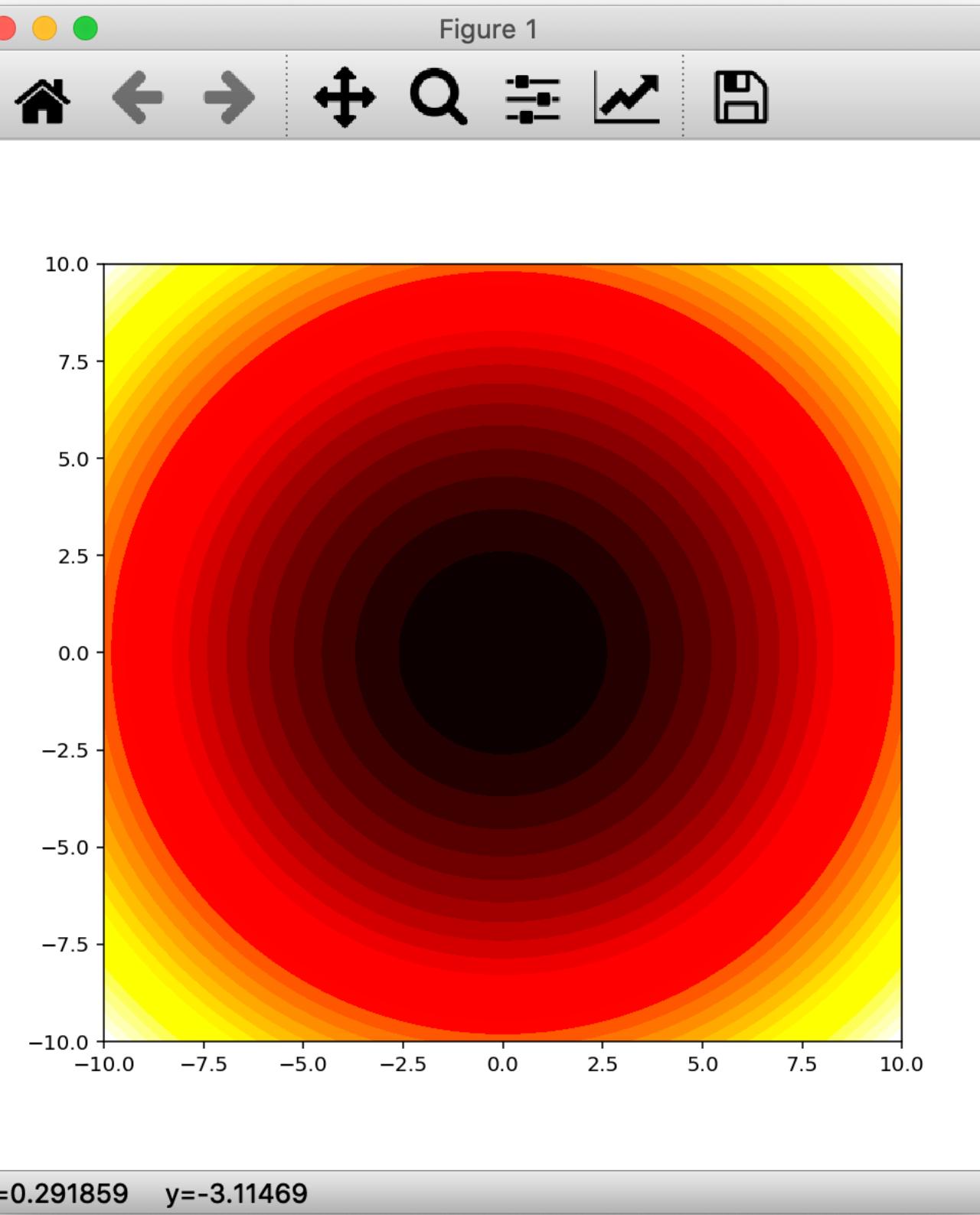


5. Continuous Colormaps(Auto Colormaps)

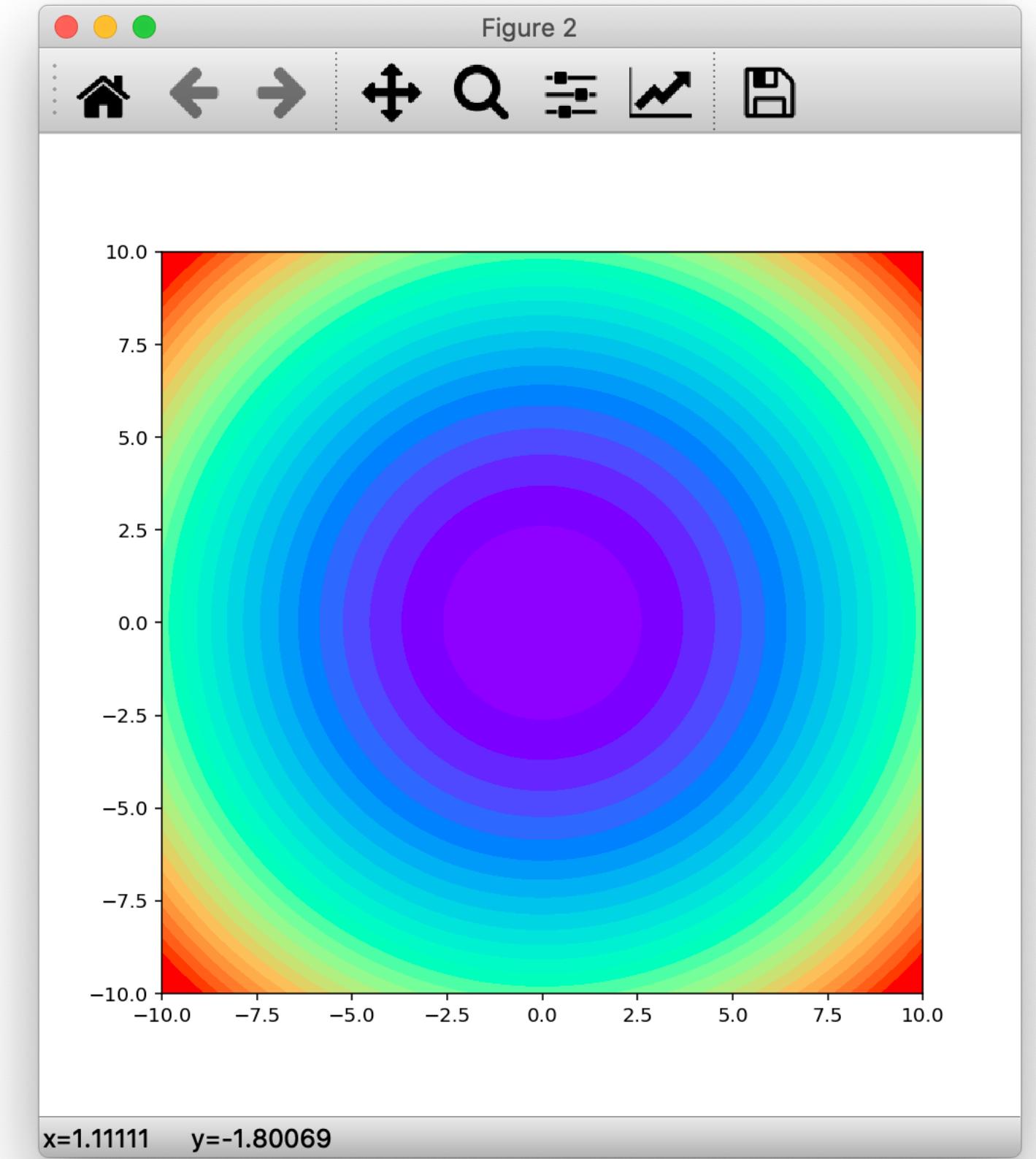
```
cmap = cm.get_cmap('Reds_r',  
                    lut=len(levels))
```



```
cmap = cm.get_cmap('hot',  
                    lut=len(levels))
```



```
cmap = cm.get_cmap('rainbow',  
                    lut=len(levels))
```



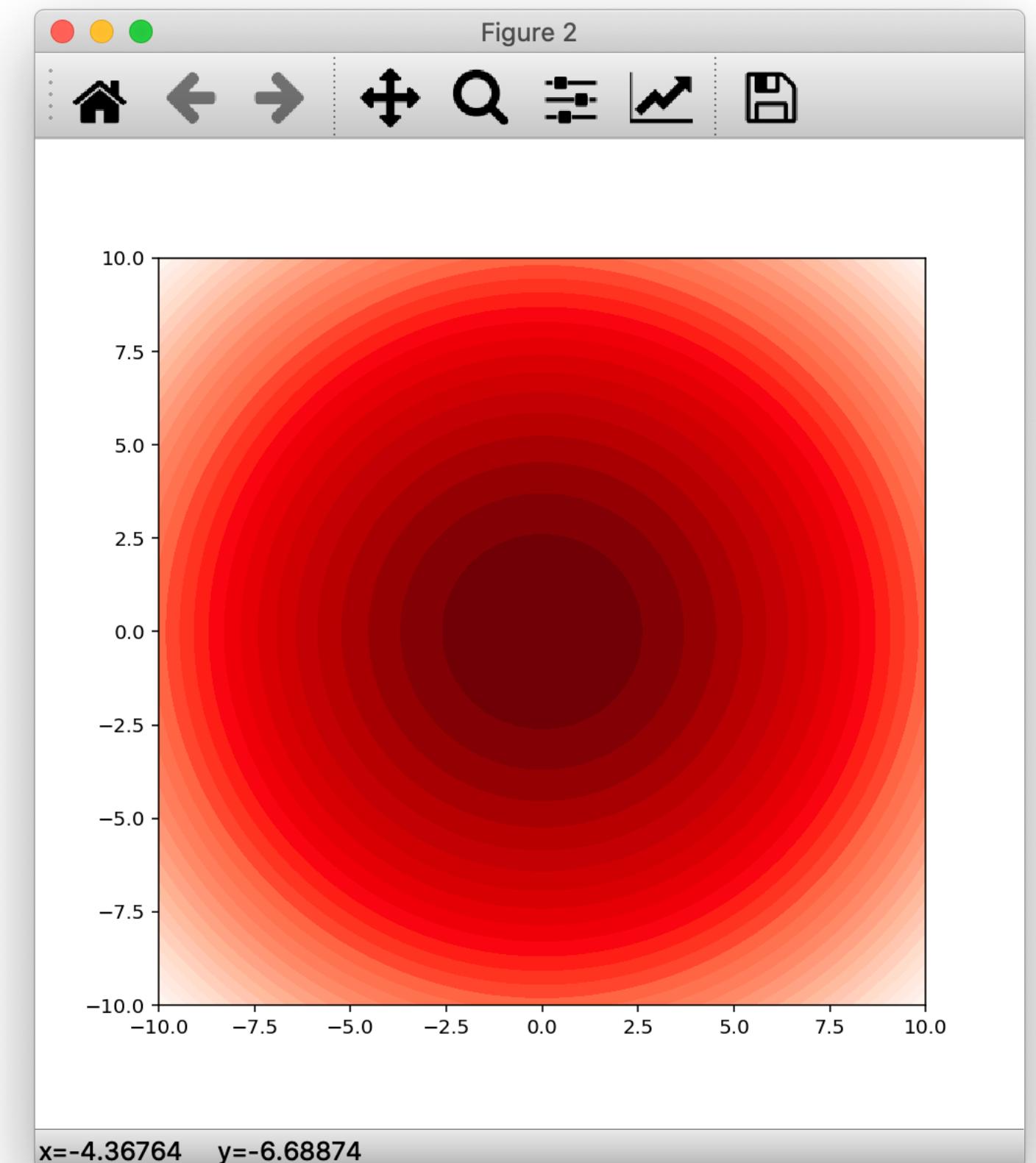
6. Colorbar

```
x = np.linspace(-10, 10, 100)
y = x

X, Y = np.meshgrid(x, y)
Z = np.power(X, 2) + np.power(Y, 2)

levels = np.linspace(np.min(Z), np.max(Z), 30)
cmap = cm.get_cmap('Reds_r', lut=len(levels))

fig, ax = plt.subplots(figsize=(7, 7))
cf = ax.contourf(X, Y, Z,
                  levels=levels,
                  cmap=cmap)
```



6. Colorbar

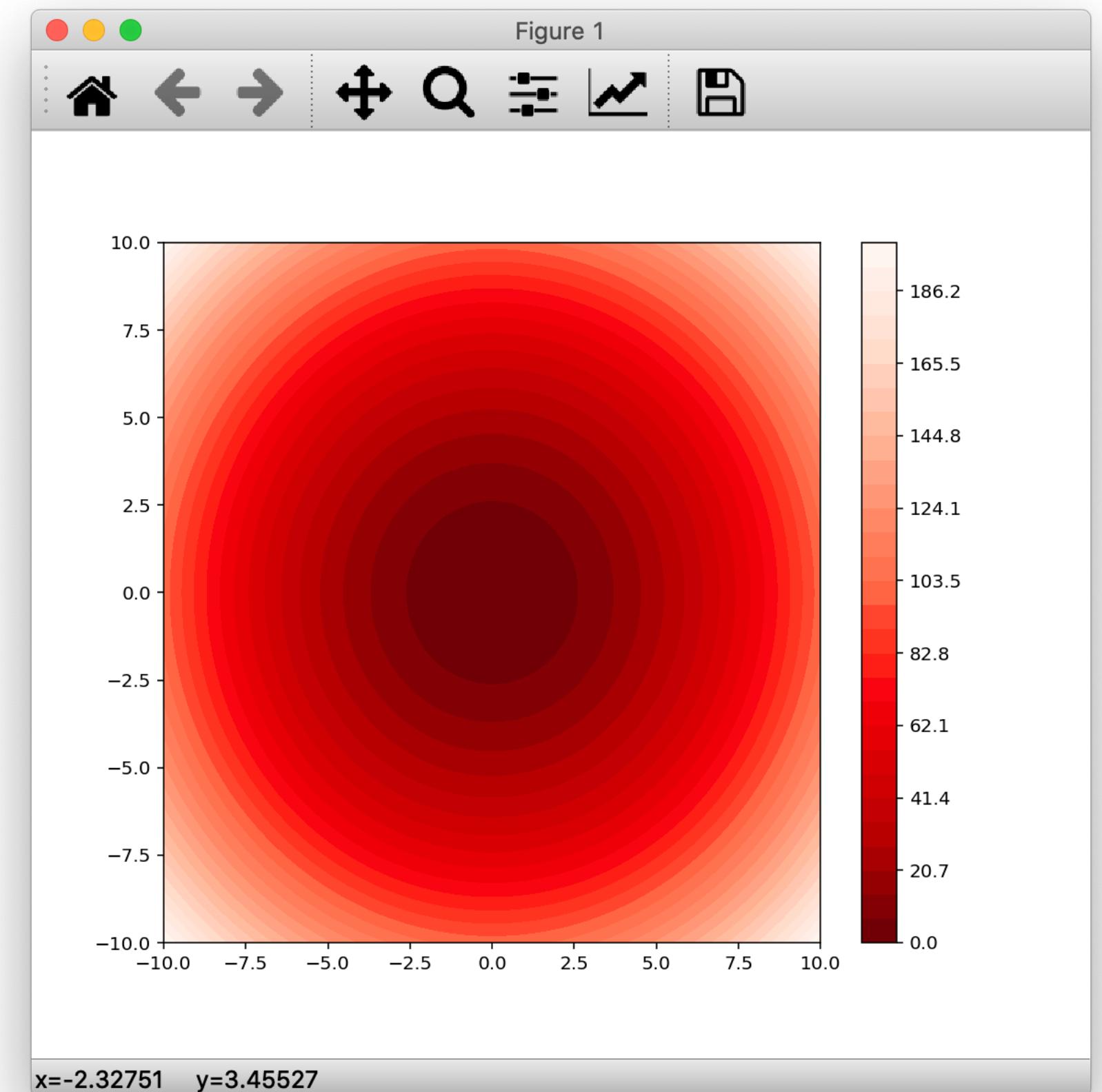
```
x = np.linspace(-10, 10, 100)
y = x

X, Y = np.meshgrid(x, y)
Z = np.power(X, 2) + np.power(Y, 2)

levels = np.linspace(np.min(Z), np.max(Z), 30)
cmap = cm.get_cmap('Reds_r', lut=len(levels))

fig, ax = plt.subplots(figsize=(8, 7))
cf = ax.contourf(X, Y, Z,
                  levels=levels,
                  cmap=cmap)

cbar = fig.colorbar(cf)
```



6. Colorbar(Setting Ticks)

```
x = np.linspace(-10, 10, 100)
y = x

X, Y = np.meshgrid(x, y)
Z = np.power(X, 2) + np.power(Y, 2)

levels = np.linspace(np.min(Z), np.max(Z), 30)
cmap = cm.get_cmap('Reds_r', lut=len(levels))

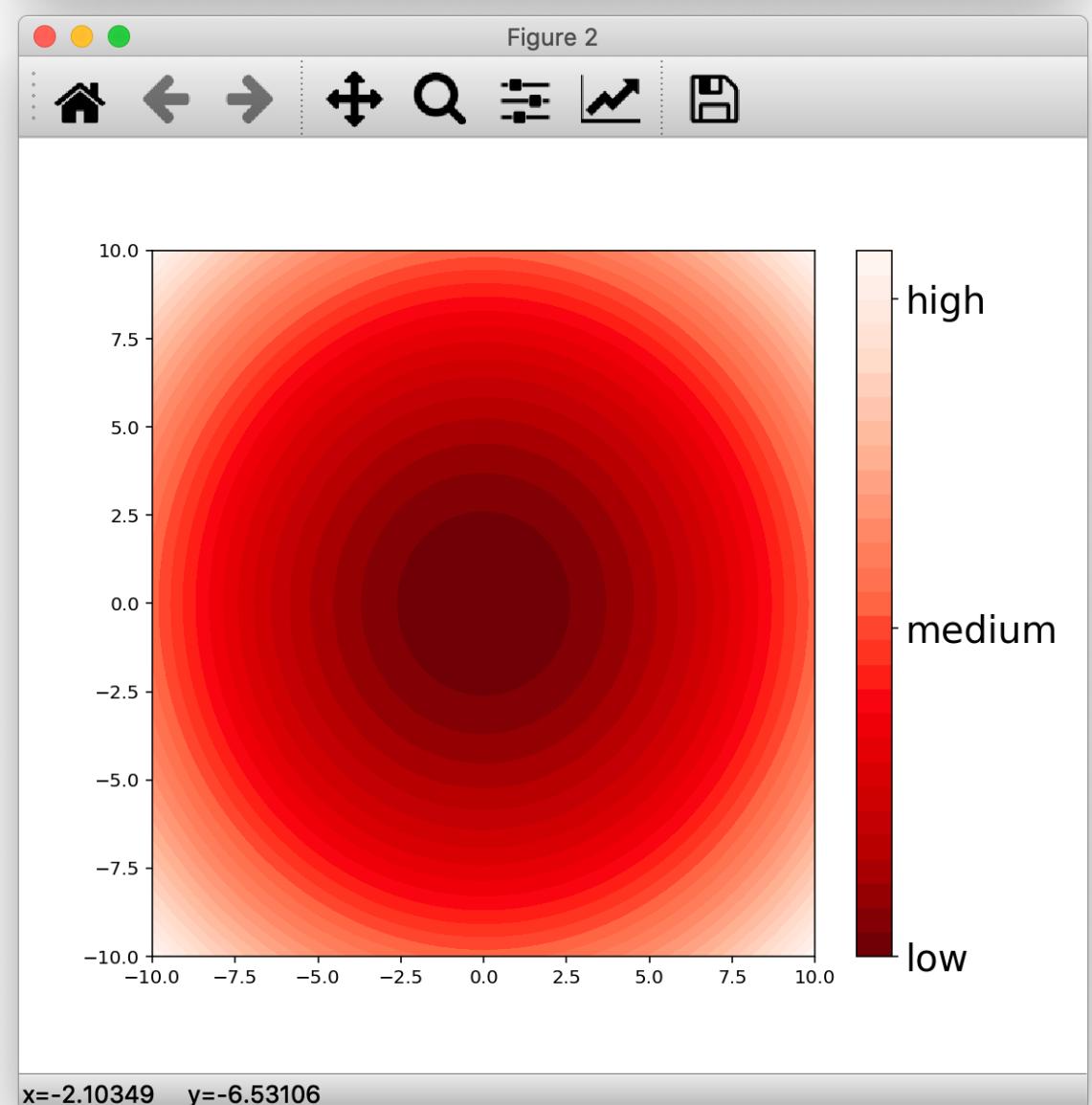
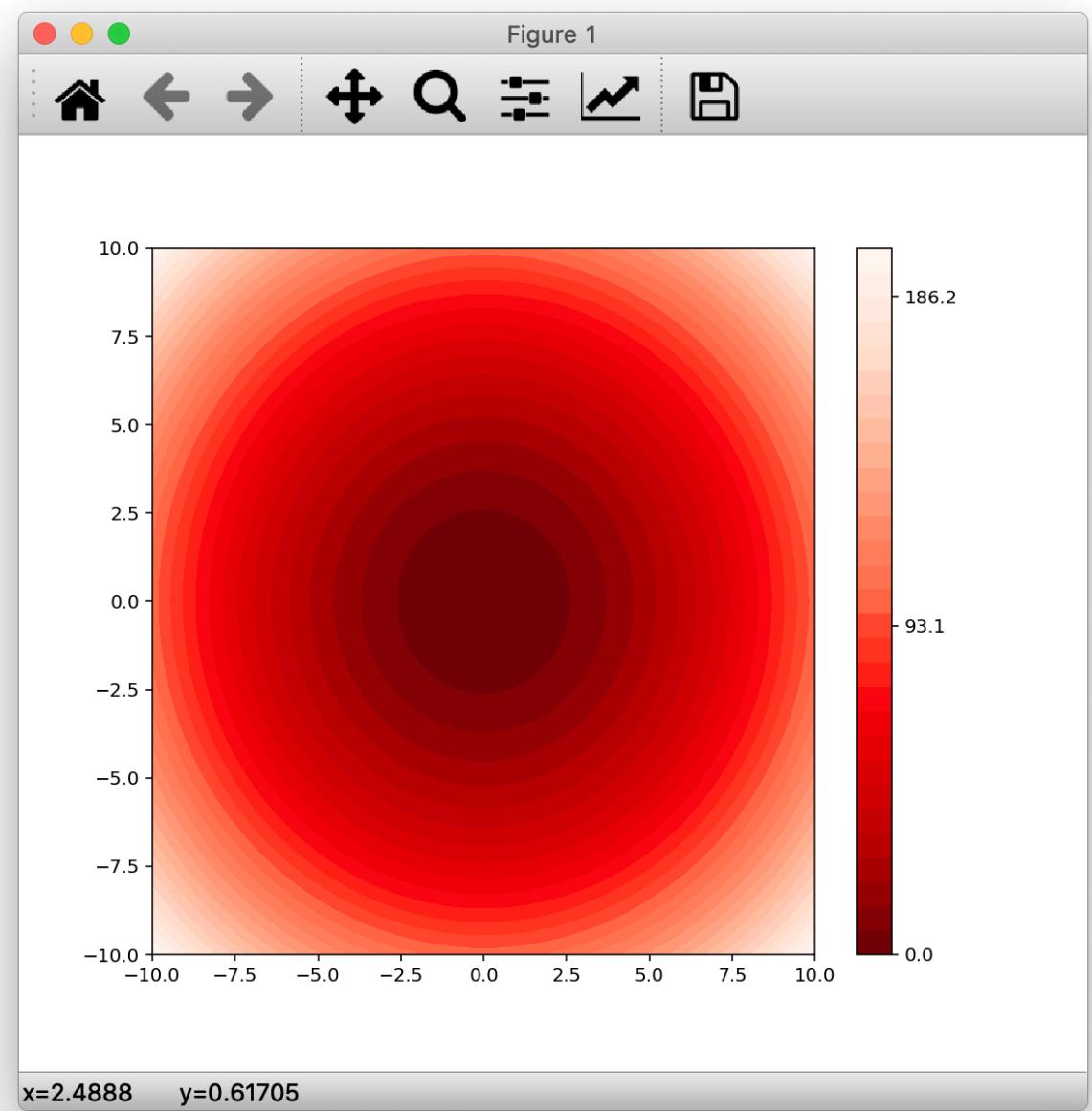
fig, ax = plt.subplots(figsize=(8, 7))
cf = ax.contourf(X, Y, Z,
                  levels=levels,
                  cmap=cmap)

cbar = fig.colorbar(cf)

cbar_ticks = cbar.get_ticks()
cbar_ticks = np.linspace(cbar_ticks[0], cbar_ticks[-1], 3)

cbar.set_ticks(cbar_ticks)

-----
cbar.set_ticklabels(['low', 'medium', 'high'])
cbar.ax.tick_params(labelsize=20)
```



6. Colorbar(Tick Location)

```
x = np.linspace(-10, 10, 100)
y = x

X, Y = np.meshgrid(x, y)
Z1 = np.power(X, 2) + np.power(Y, 2)
Z2 = np.power(X, 3) + np.power(Y, 3)

levels1 = np.linspace(np.min(Z1), np.max(Z1), 30)
cmap1 = cm.get_cmap('Reds_r', lut=len(levels1))

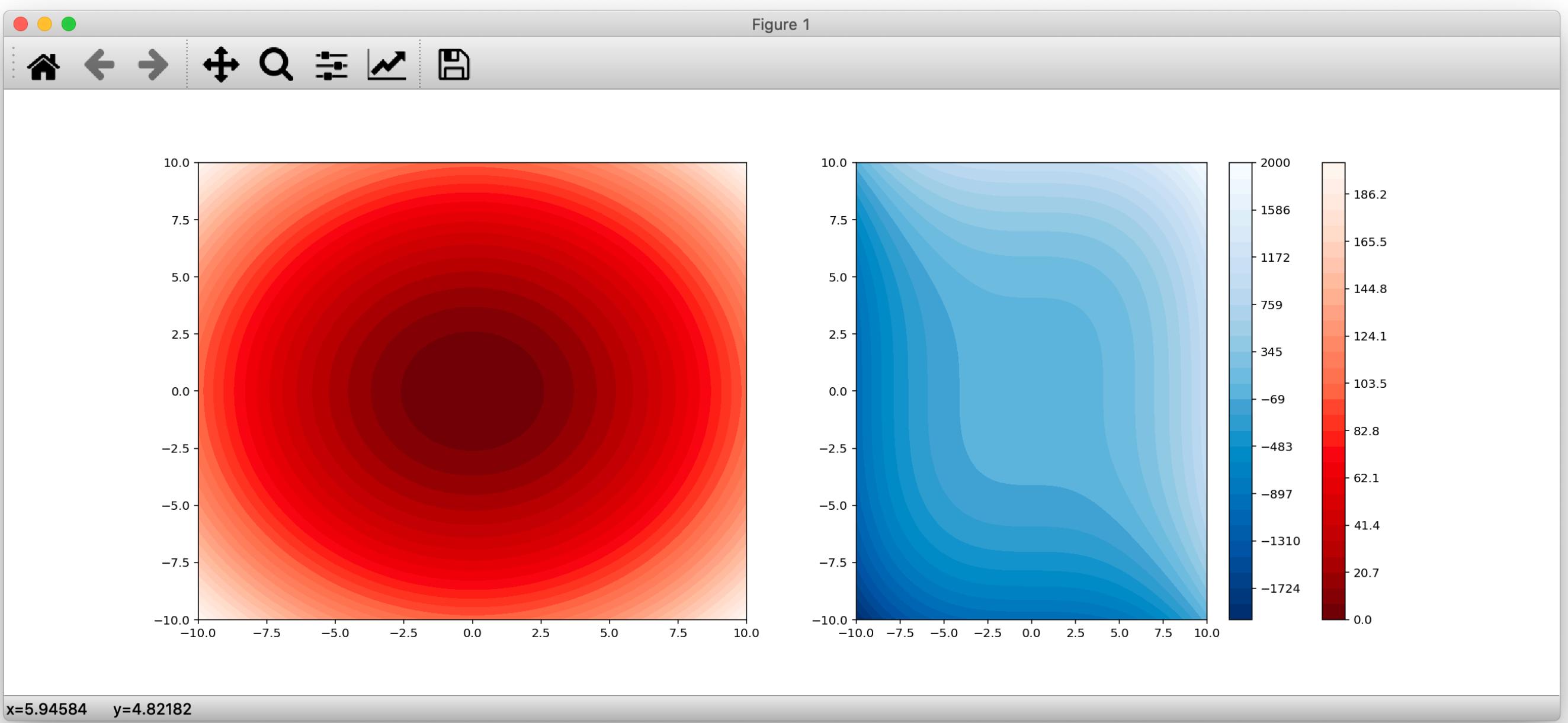
levels2 = np.linspace(np.min(Z2), np.max(Z2), 30)
cmap2 = cm.get_cmap('Blues_r', lut=len(levels2))

fig, axes = plt.subplots(1, 2,
                        figsize=(18, 7))

cf1 = axes[0].contourf(X, Y, Z1,
                       levels=levels1,
                       cmap=cmap1)

cf2 = axes[1].contourf(X, Y, Z2,
                       levels=levels2,
                       cmap=cmap2)

cbar1 = fig.colorbar(cf1)
cbar2 = fig.colorbar(cf2)
```



6. Colorbar(Tick Location)

```

x = np.linspace(-10, 10, 100)
y = x

X, Y = np.meshgrid(x, y)
Z1 = np.power(X, 2) + np.power(Y, 2)
Z2 = np.power(X, 3) + np.power(Y, 3)

levels1 = np.linspace(np.min(Z1), np.max(Z1), 30)
cmap1 = cm.get_cmap('Reds_r', lut=len(levels1))
levels2 = np.linspace(np.min(Z2), np.max(Z2), 30)
cmap2 = cm.get_cmap('Blues_r', lut=len(levels2))

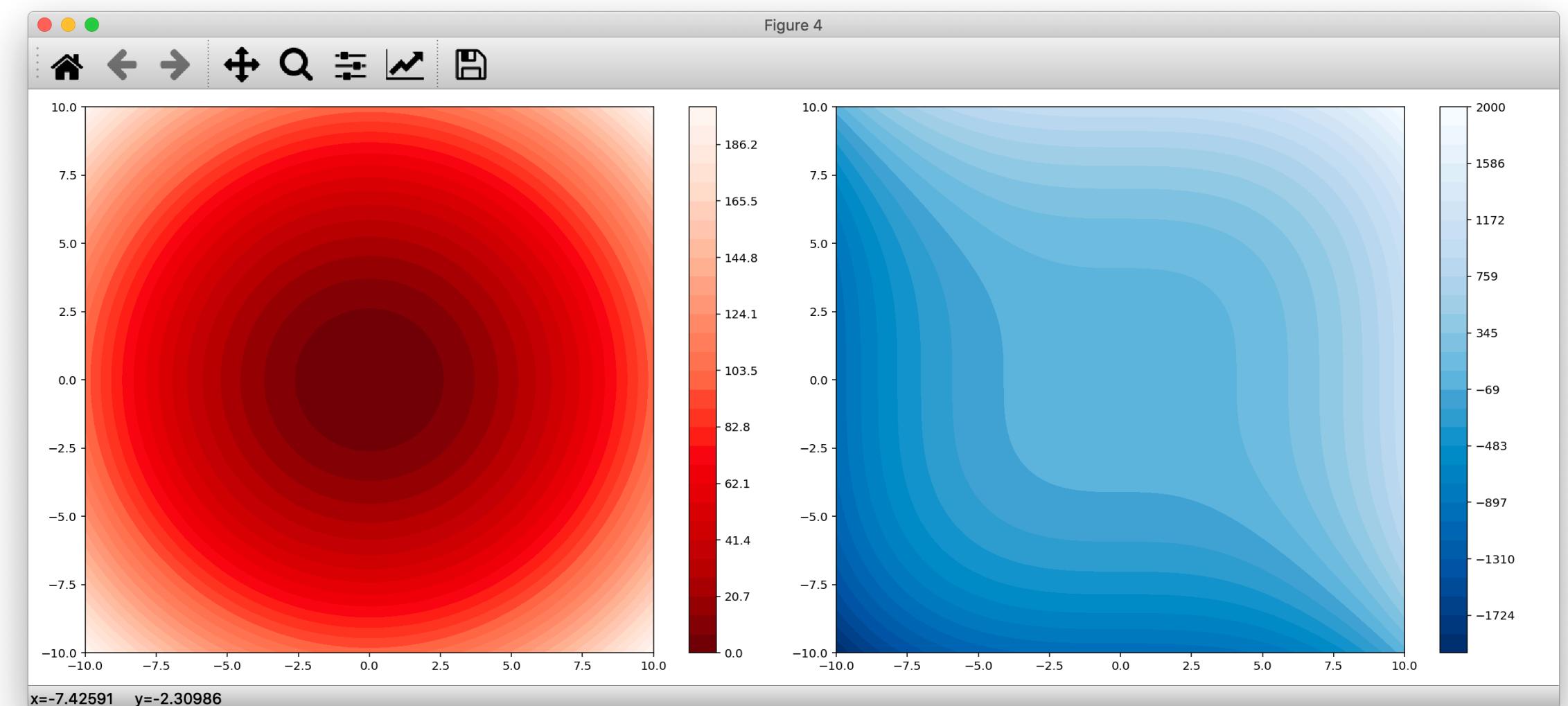
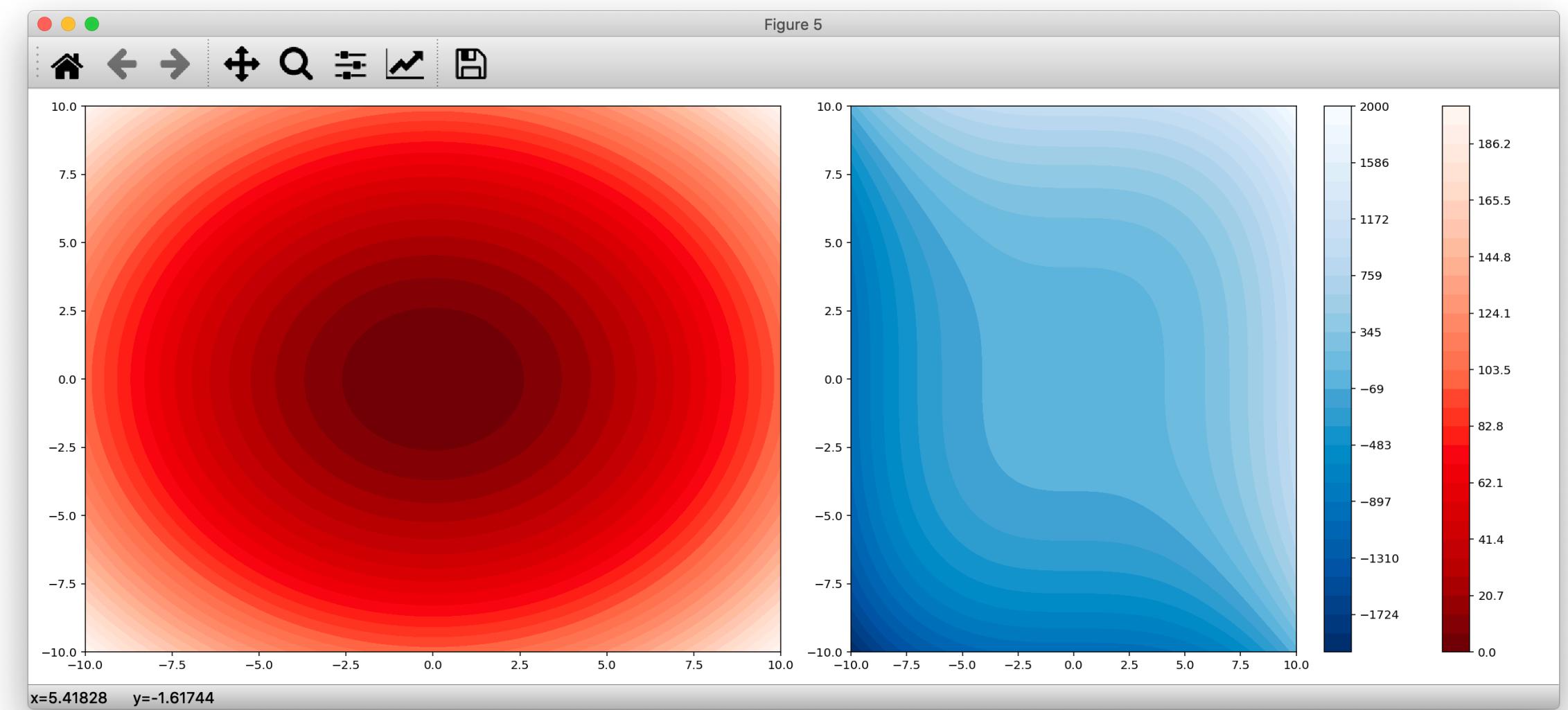
fig, axes = plt.subplots(1, 2,
                        figsize=(18, 7))

cf1 = axes[0].contourf(X, Y, Z1,
                       levels=levels1,
                       cmap=cmap1)
cf2 = axes[1].contourf(X, Y, Z2,
                       levels=levels2,
                       cmap=cmap2)

fig.tight_layout()

cbar1 = fig.colorbar(cf1)
cbar2 = fig.colorbar(cf2)
.....
cbar1 = fig.colorbar(cf1, ax=axes[0])
cbar2 = fig.colorbar(cf2, ax=axes[1])

```



Python for Data Visualization

-Chapter.1 Matplotlib Anatomy -

1-07. Colors in Matplotlib

- 1. Named Colors
- 2. RGB Colors
- 3. Colormaps in Matplotlib
- 4. Discrete Colormaps
- 5. Continuous Colormaps
- 6. Colorbar