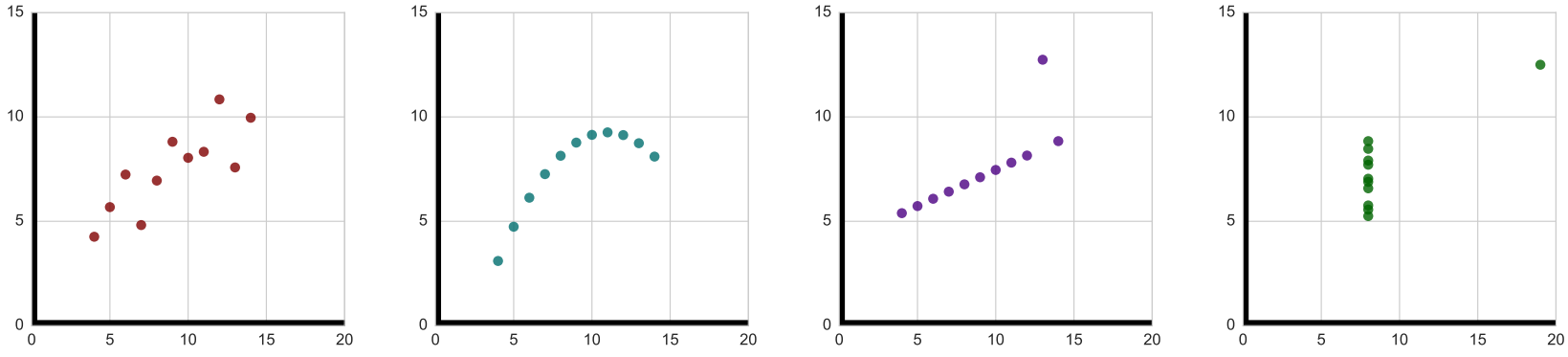
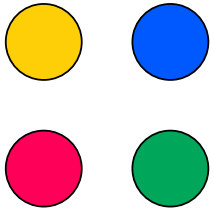
# Producing Pretty Plots in Python

Geraint Ian Palmer
@GeraintPalmer

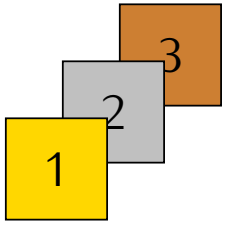
PyCon Namibia 2017



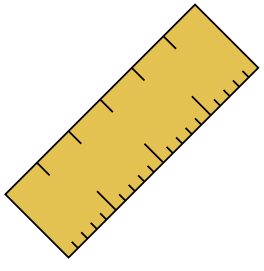
# Types of Data



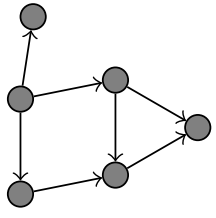
**Nominal**




**Ordinal**

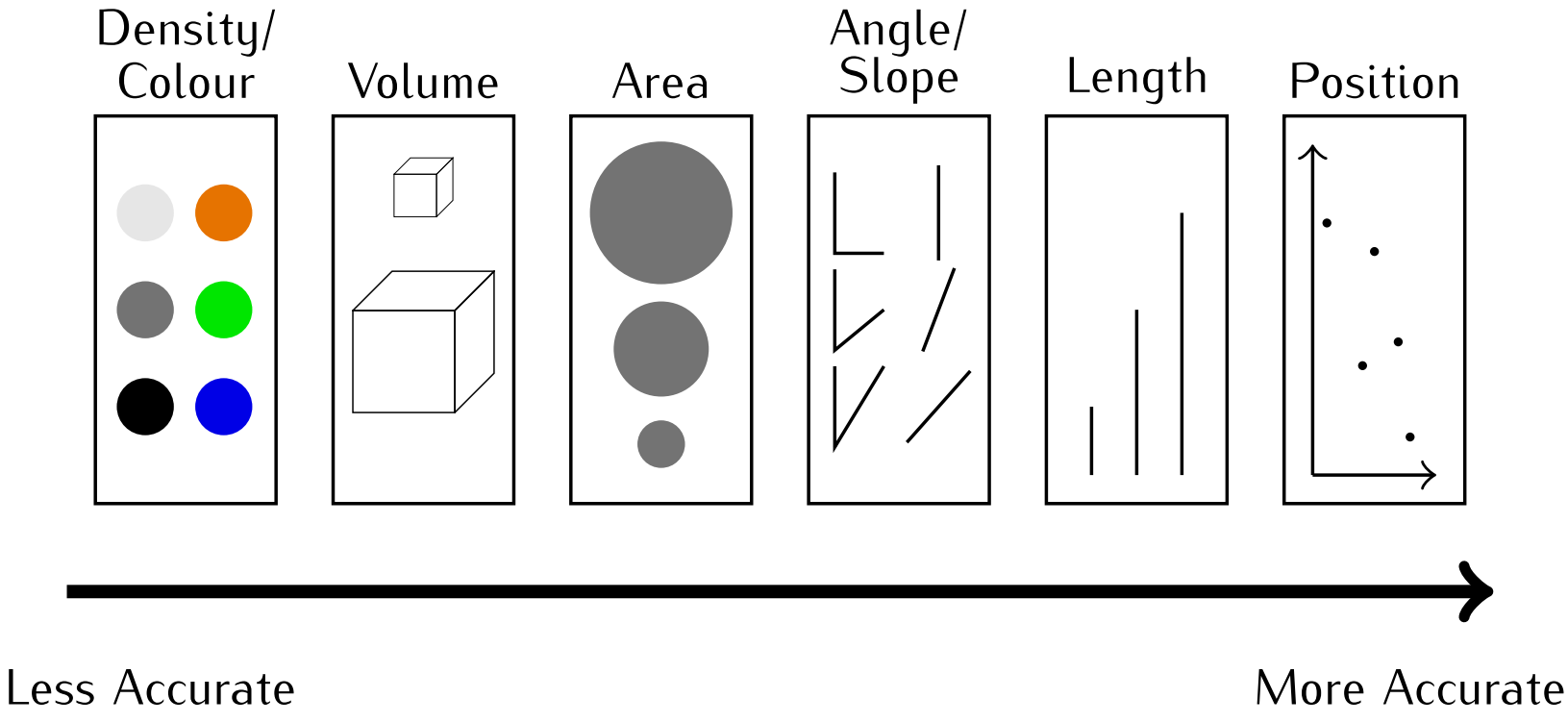


**Quantitative**



**Relational**

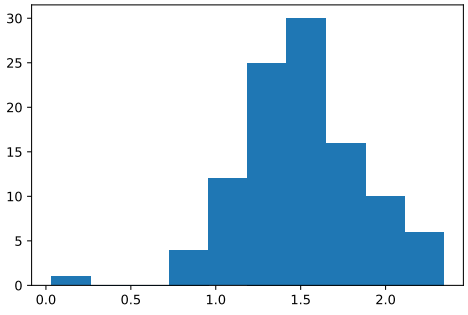
# Perceptual Accuracy



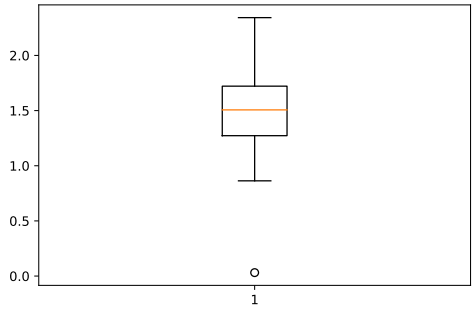

YouTube: Design Principles in Information Visualisation
(Prof. Jessie Kennedy)

# import matplotlib.pyplot as plt

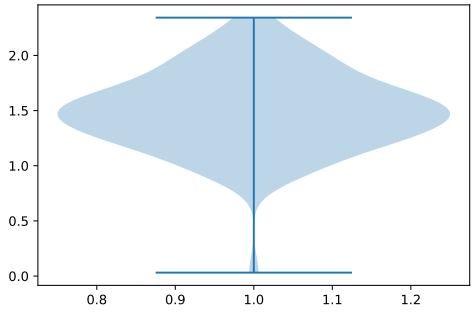
plt.hist(xs)



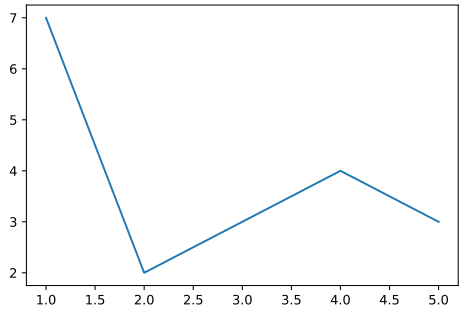

plt.boxplot(xs)



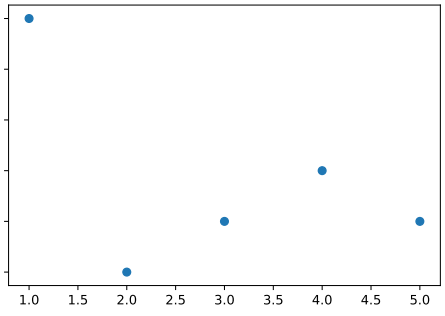

plt.violinplot(xs)



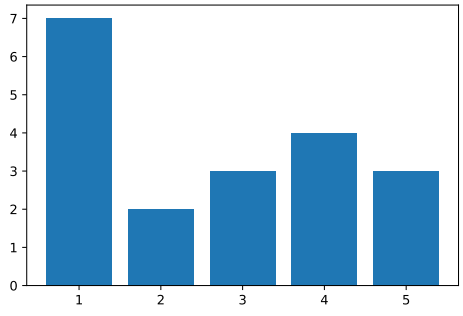

plt.plot(xs, ys)



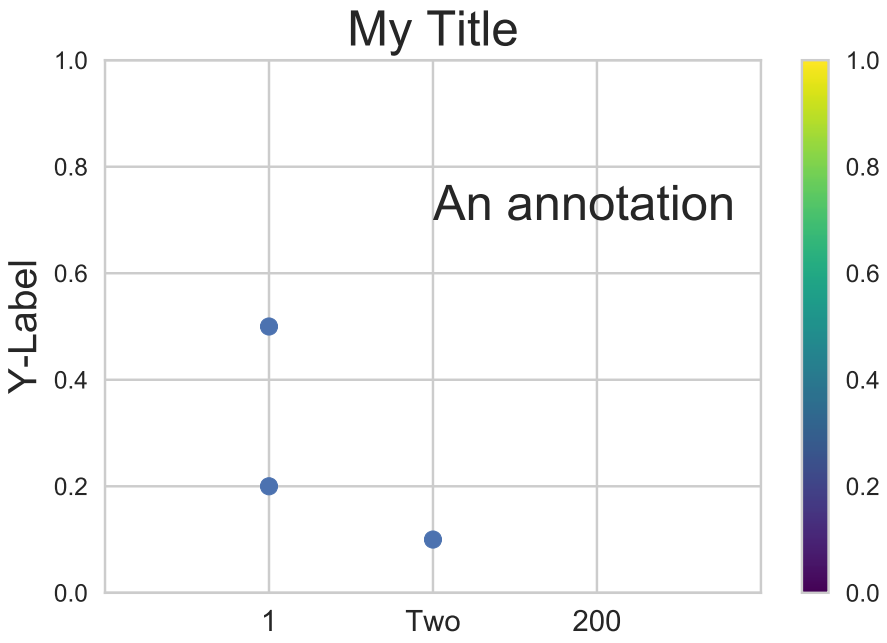

plt.scatter(xs, ys)




plt.bar(xs, ys)

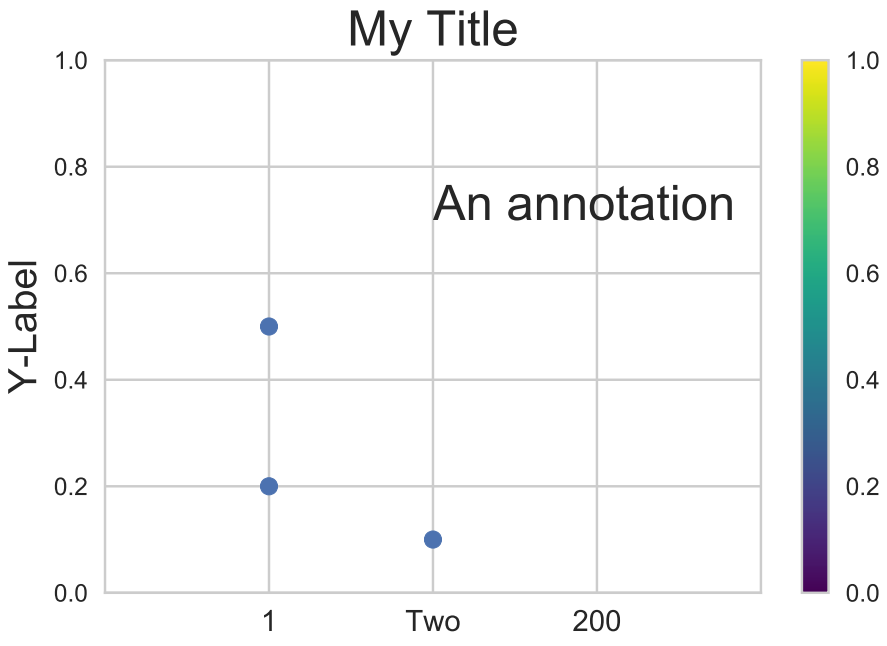


# Anatomy of a Plot



# Anatomy of a Plot

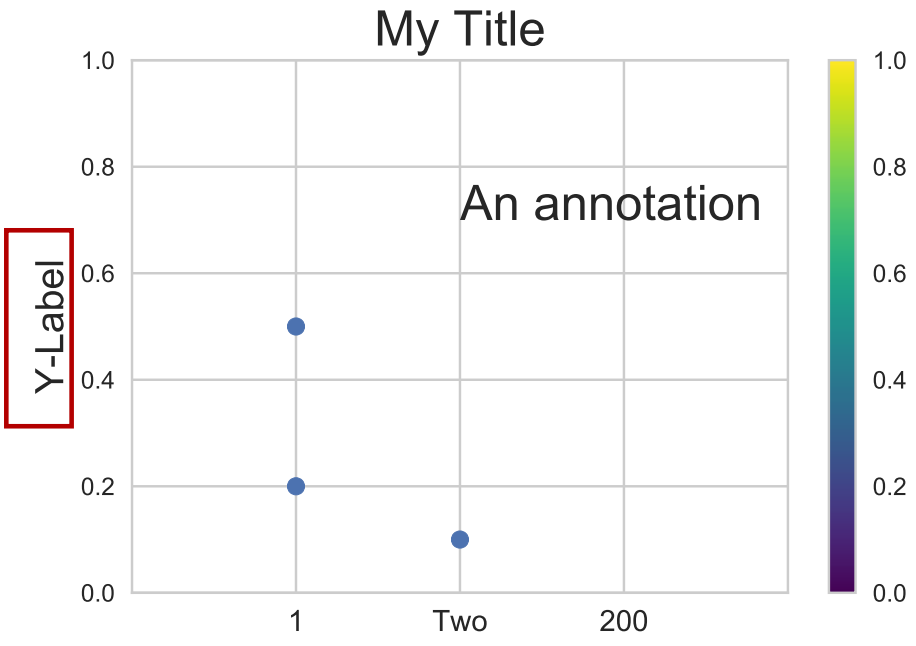
```
fig, ax = plt.subplots(1)
```



# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```

Axis Labels
`ax.set_ylabel()`



# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```

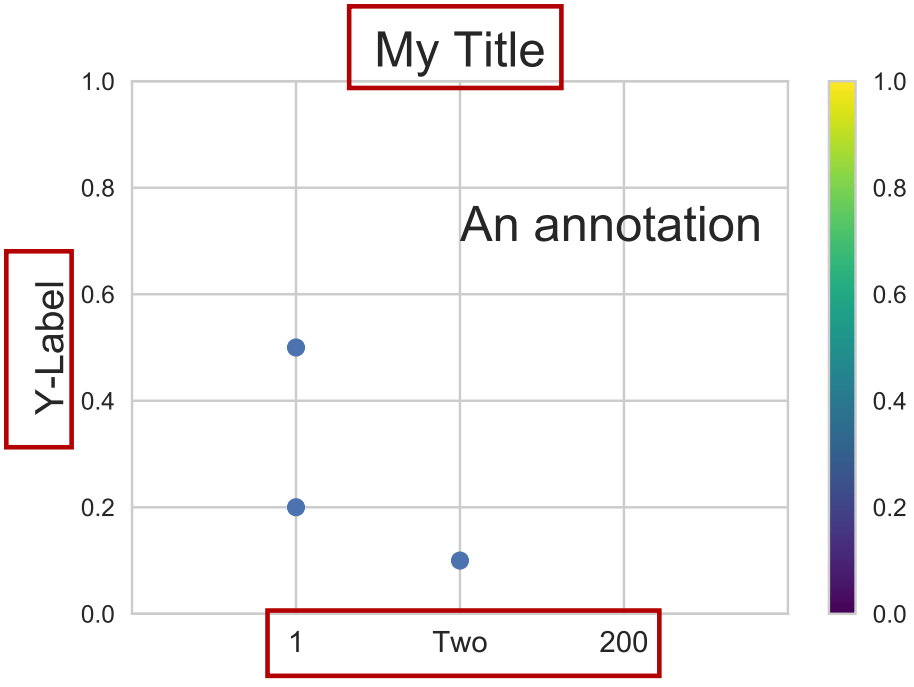
Title
`ax.set_title()`

Axis Labels
`ax.set_ylabel()`

# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```

Title
`ax.set_title()`

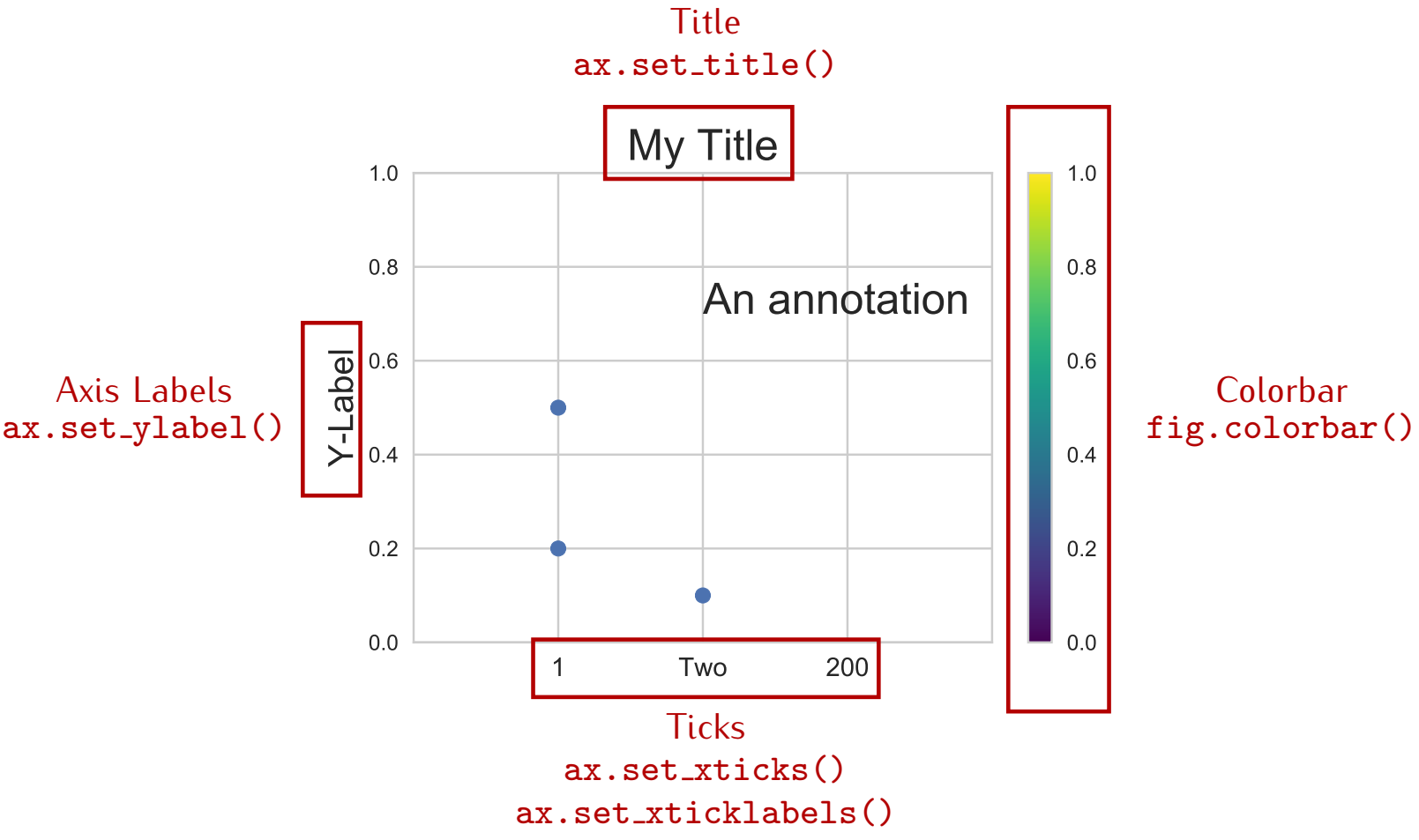



Axis Labels
`ax.set_ylabel()`

Ticks
`ax.set_xticks()`
`ax.set_xticklabels()`

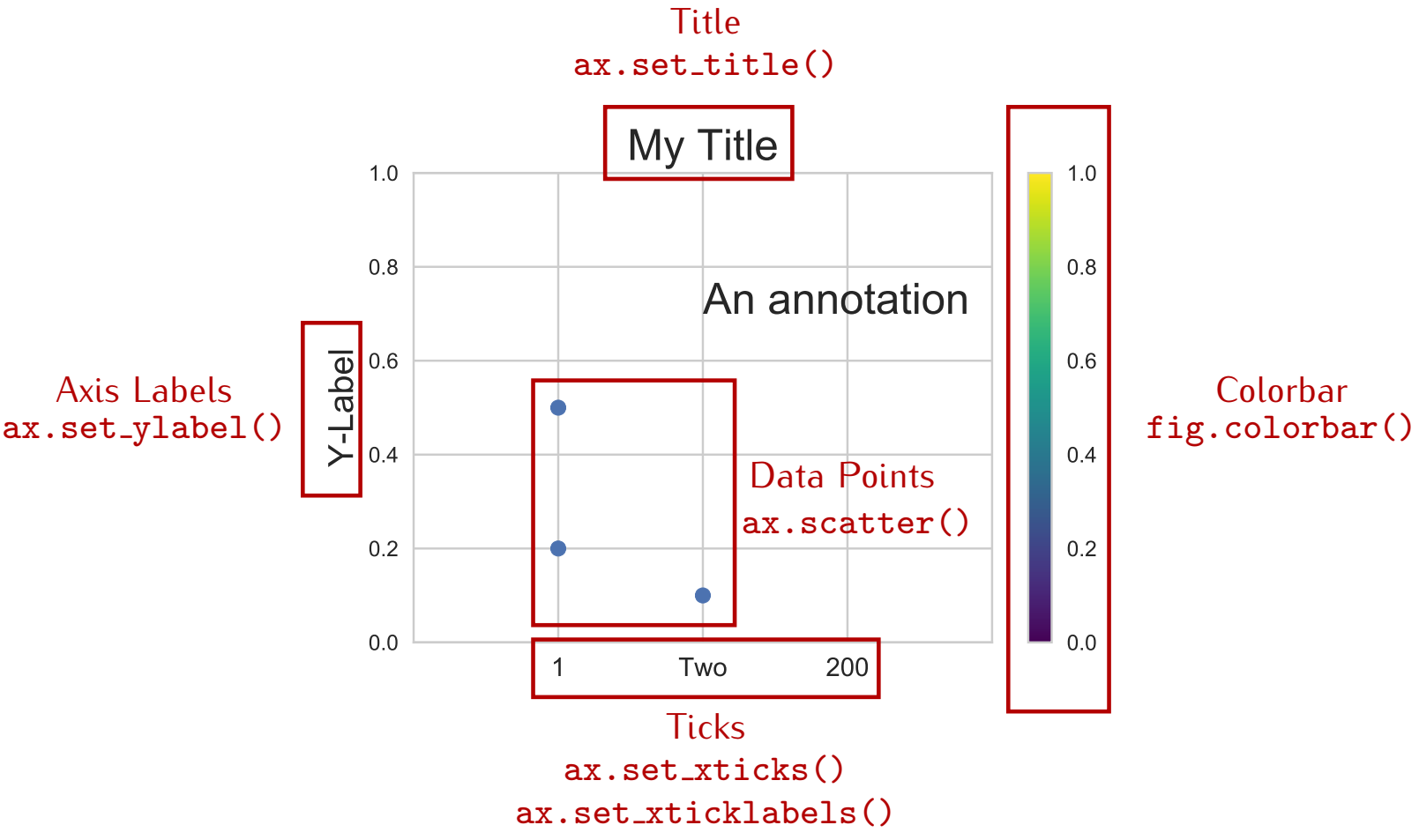
# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```



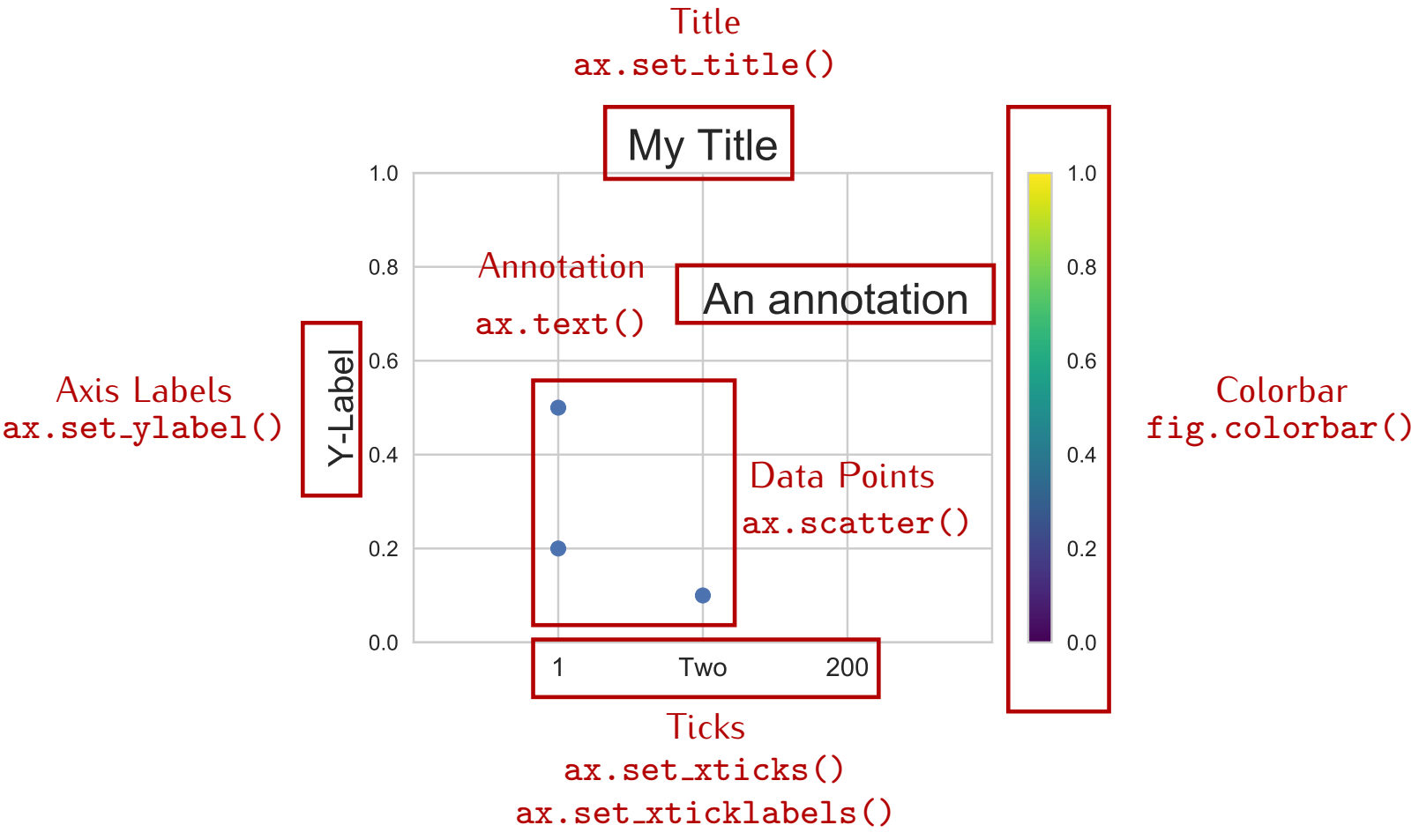
# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```



# Anatomy of a Plot

```
fig, ax = plt.subplots(1)
```

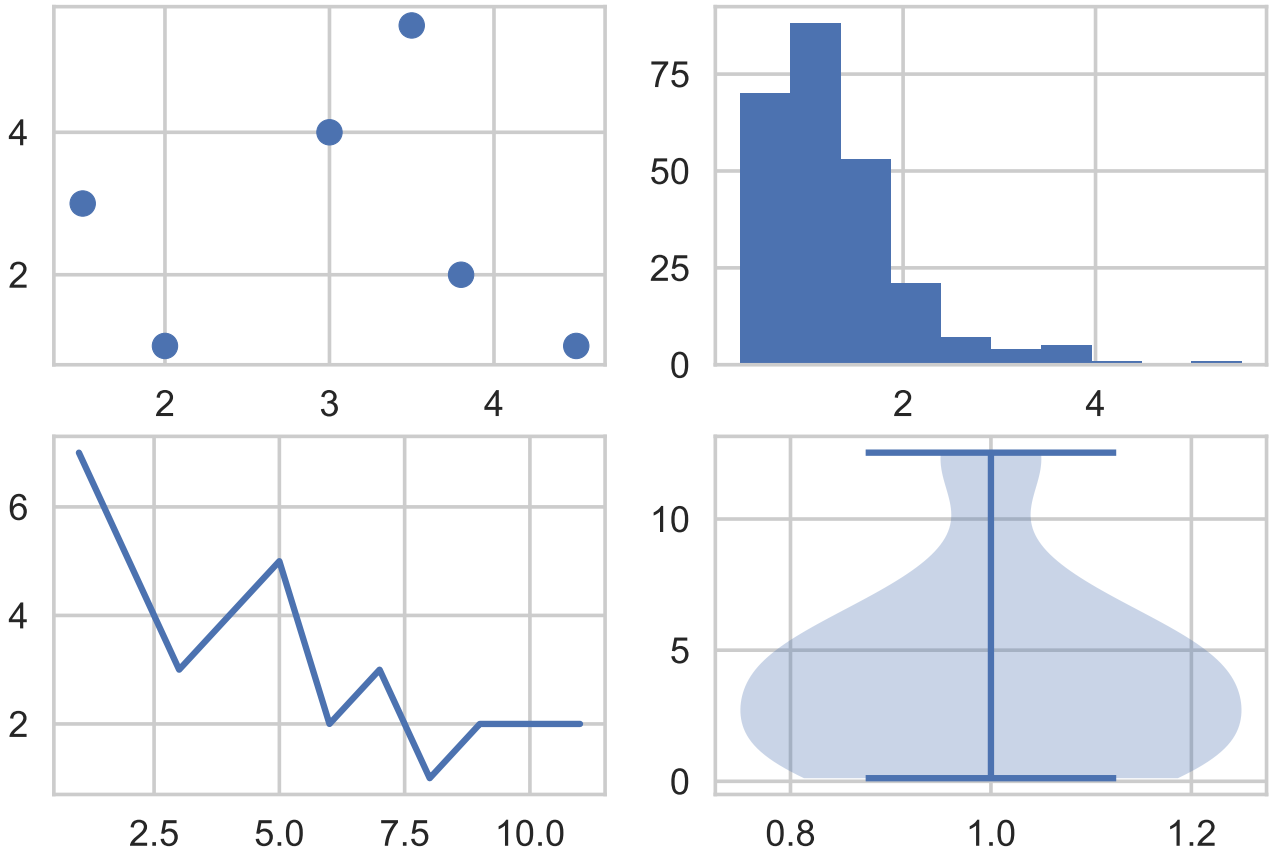


# Anatomy of a Plot

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

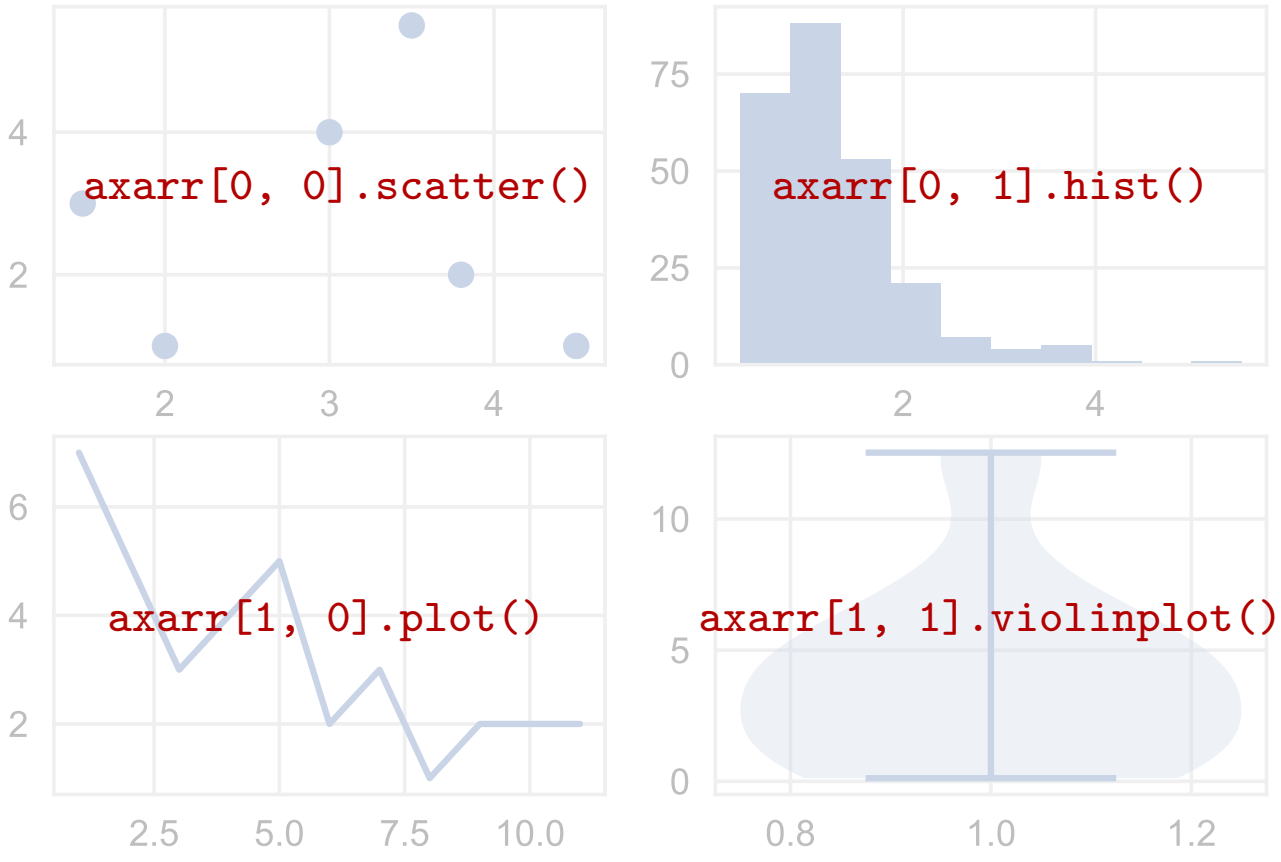
# Anatomy of a Plot

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



# Anatomy of a Plot

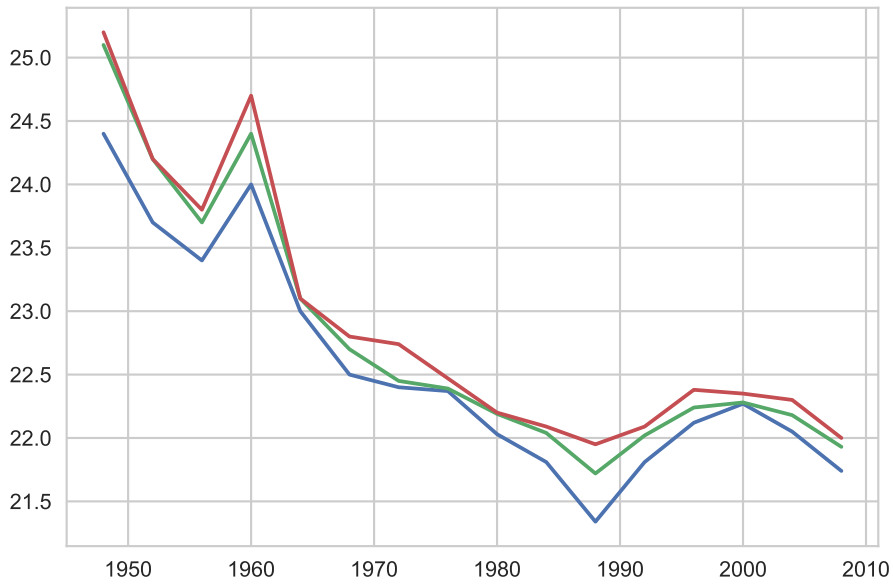
```
fig, axarr = plt.subplots(2, 2)
```



# Women's 200m Olympic Medallists

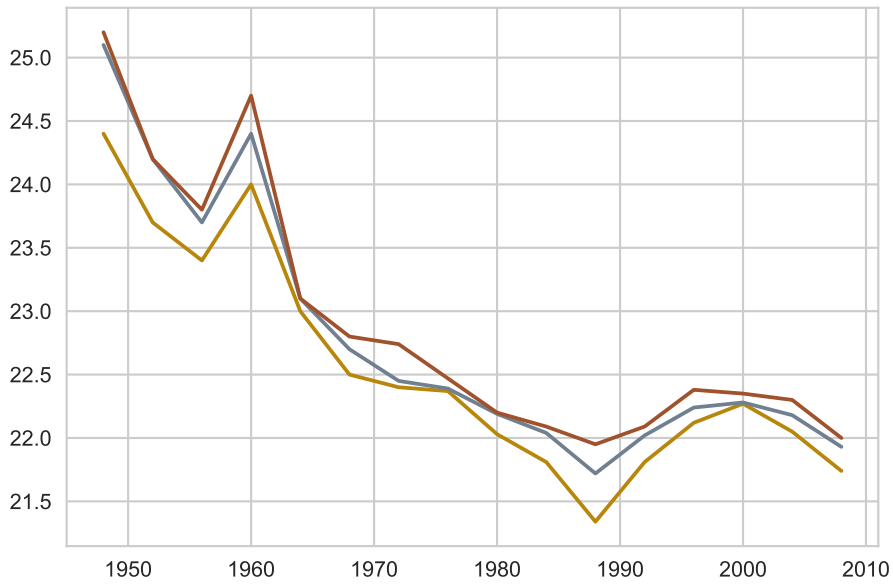
| Year | Athlete | Medal | Country | Result |
|---|---|---|---|---|
| 1948 | Fanny Blankers-Koen | GOLD | NED | 24.40 |
| 1948 | Audrey Williamson | SILVER | GBR | 25.10 |
| 1948 | Audrey Patterson | BRONZE | USA | 25.20 |
| 1952 | Marjorie Jackson | GOLD | AUS | 23.70 |
| 1952 | Bertha Brouwer | SILVER | NED | 24.20 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 2008 | Allyson Felix | SILVER | USA | 21.93 |
| 2008 | Kerron Stewart | BRONZE | JAM | 22.00 |

```
fig, ax = plt.subplots(1)

ax.plot(dates, gold)

ax.plot(dates, silver)

ax.plot(dates, bronze)
```



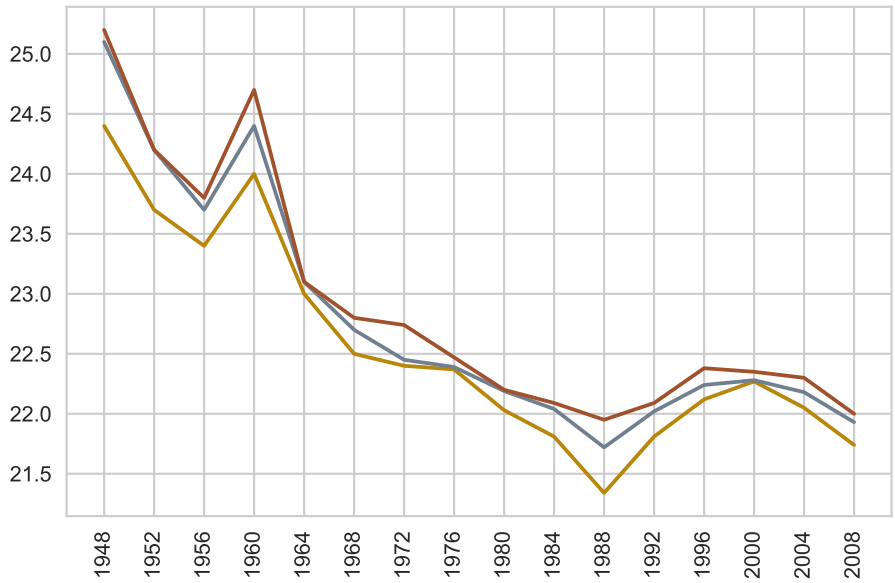

```
plt.show()
```

```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod')

ax.plot(dates, silver, c='slategray')

ax.plot(dates, bronze, c='sienna')
```




```
plt.show()
```

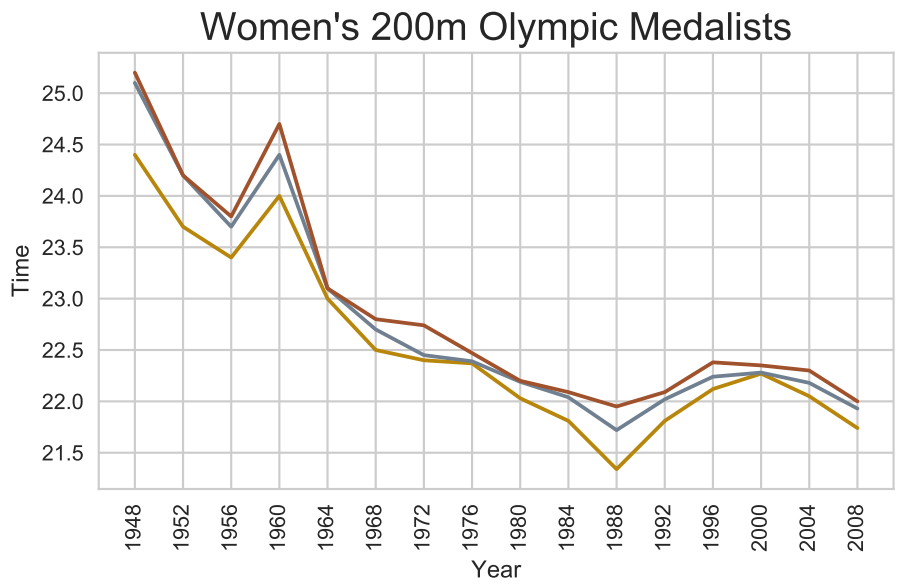
```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod')

ax.plot(dates, silver, c='slategray')

ax.plot(dates, bronze, c='sienna')
```




```
ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')
```

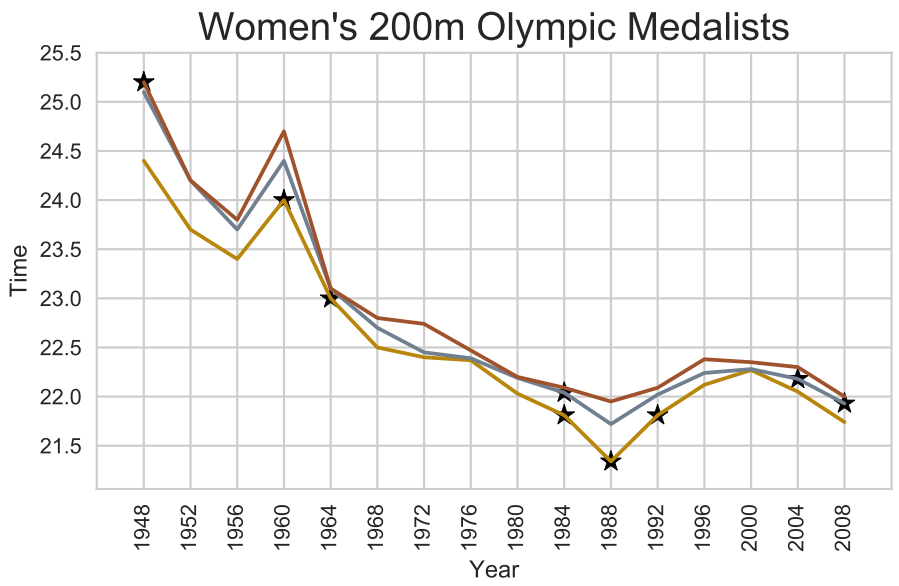
```
plt.show()
```

```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod')

ax.plot(dates, silver, c='slategray')

ax.plot(dates, bronze, c='sienna')
```

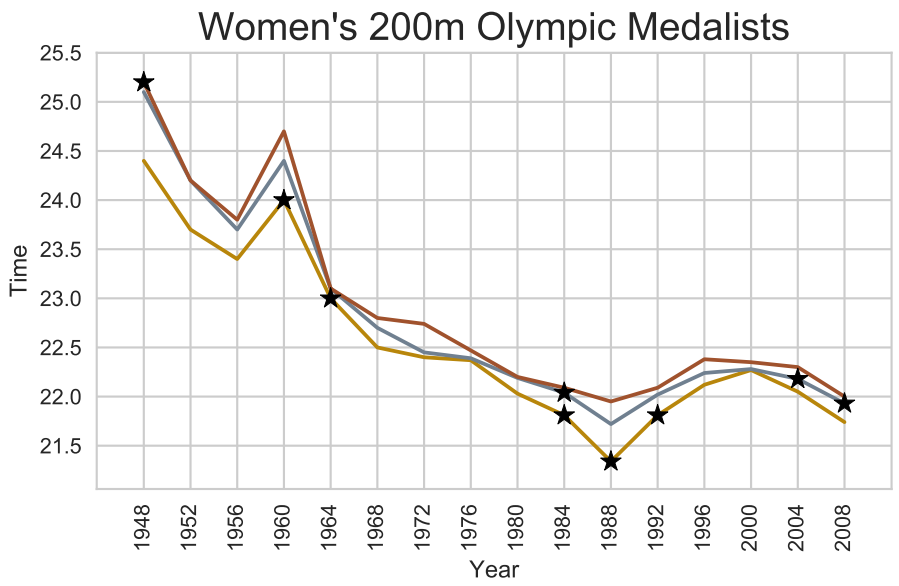



```
ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')

ax.set_xlabel("Year")
ax.set_ylabel("Time")
ax.set_title("Women's 200m Olympic Medalists", fontsize=18)

plt.show()
```

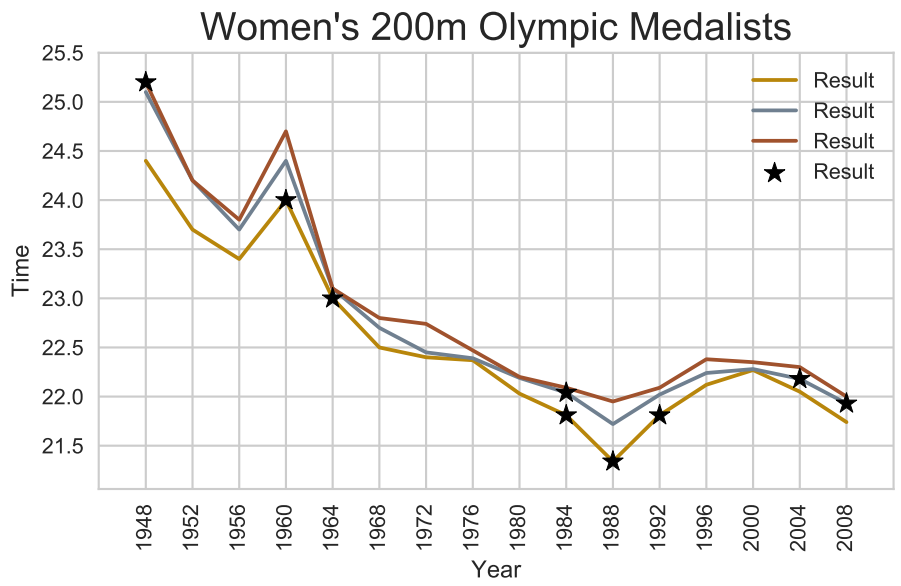
```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod')

ax.plot(dates, silver, c='slategray')

ax.plot(dates, bronze, c='sienna')


ax.scatter(usa_x, usa_y, lw=0.8,
           facecolor='black',
           marker='*', s=100)




ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')

ax.set_xlabel("Year")
ax.set_ylabel("Time")
ax.set_title("Women's 200m Olympic Medalists", fontsize=18)

plt.show()
```



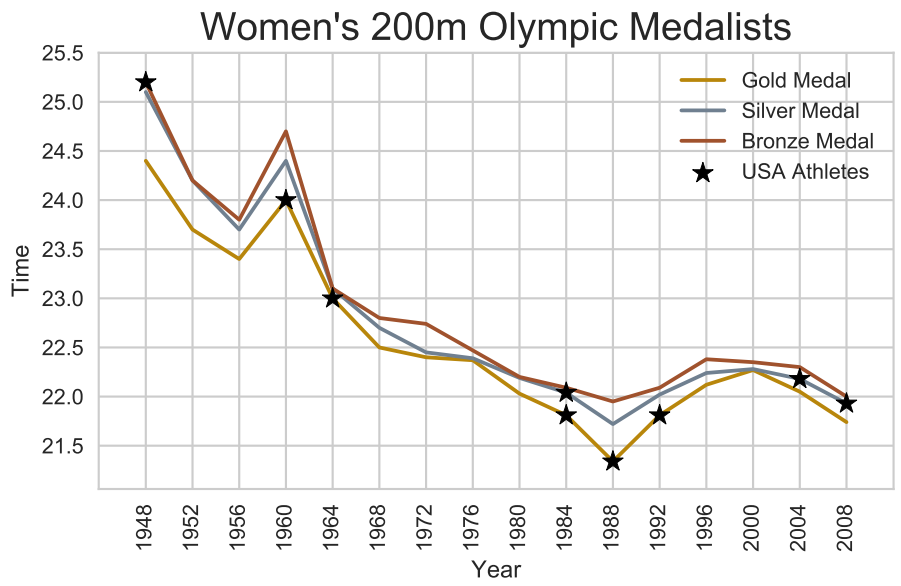
```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod',
        zorder=1)
ax.plot(dates, silver, c='slategray',
        zorder=1)
ax.plot(dates, bronze, c='sienna',
        zorder=1)

ax.scatter(usa_x, usa_y, lw=0.8,
           facecolor='black',
           marker='*', s=100,
           zorder=2)



ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')

ax.set_xlabel("Year")
ax.set_ylabel("Time")
ax.set_title("Women's 200m Olympic Medalists", fontsize=18)

plt.show()
```



```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod',
        zorder=1)
ax.plot(dates, silver, c='slategray',
        zorder=1)
ax.plot(dates, bronze, c='sienna',
        zorder=1)

ax.scatter(usa_x, usa_y, lw=0.8,
           facecolor='black',
           marker='*', s=100,
           zorder=2)

plt.legend()

ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')

ax.set_xlabel("Year")
ax.set_ylabel("Time")
ax.set_title("Women's 200m Olympic Medalists", fontsize=18)

plt.show()
```



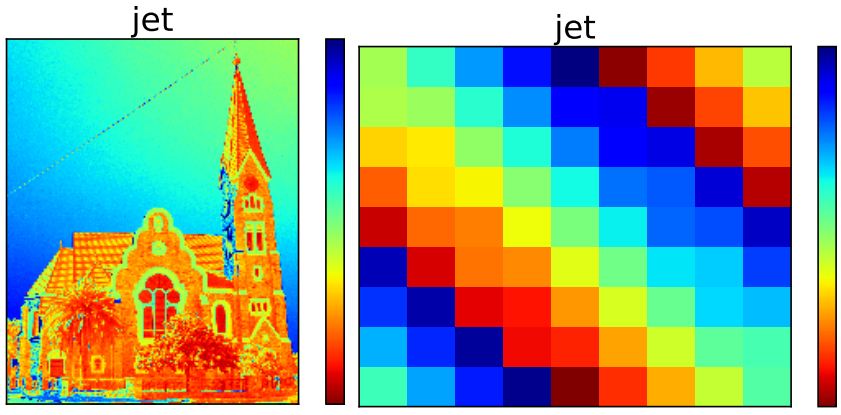
```
fig, ax = plt.subplots(1)

ax.plot(dates, gold, c='darkgoldenrod',
        zorder=1, label='Gold Medal')
ax.plot(dates, silver, c='slategray',
        zorder=1, label='Silver Medal')
ax.plot(dates, bronze, c='sienna',
        zorder=1, label='Bronze Medal')

ax.scatter(usa_x, usa_y, lw=0.8,
           facecolor='black',
           marker='*', s=100,
           zorder=2, label='USA Athletes')

plt.legend()

ax.set_xticks(dates)
ax.set_xticklabels(dates, rotation='vertical')

ax.set_xlabel("Year")
ax.set_ylabel("Time")
ax.set_title("Women's 200m Olympic Medalists", fontsize=18)

plt.show()
```

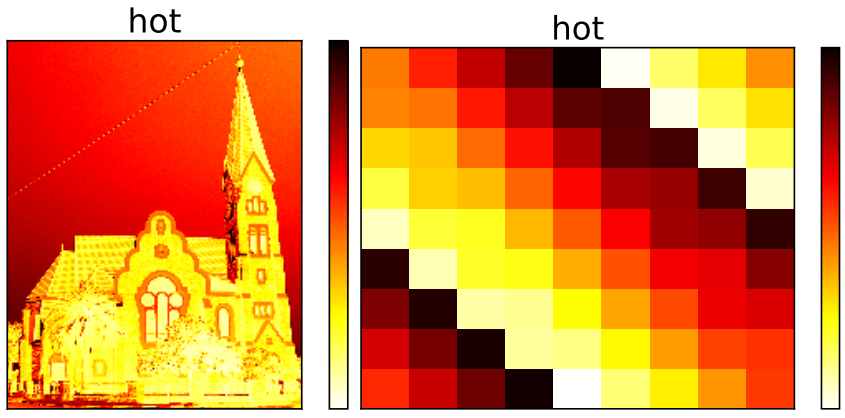


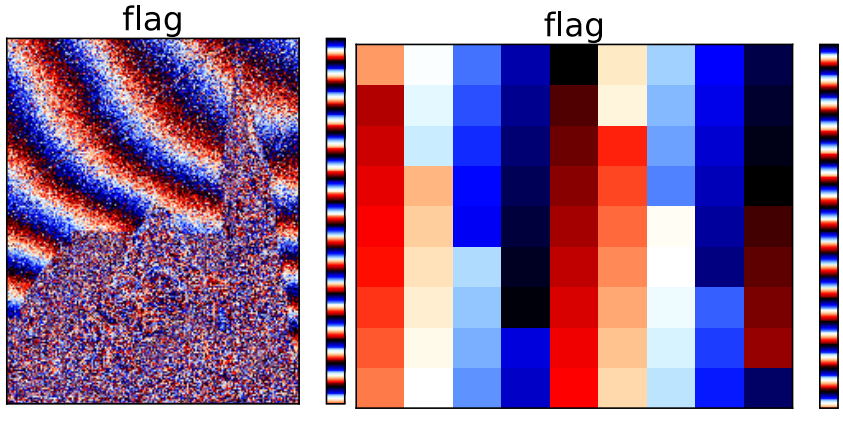
# Choosing Colormaps

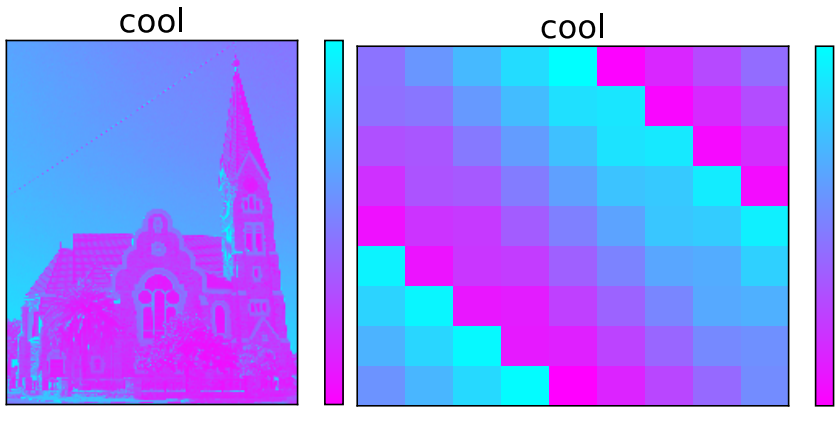
YouTube: A Better Default Colormap for Matplotlib – SciPy 2015 (Nathaniel Smith and Stéfan van der Walt)

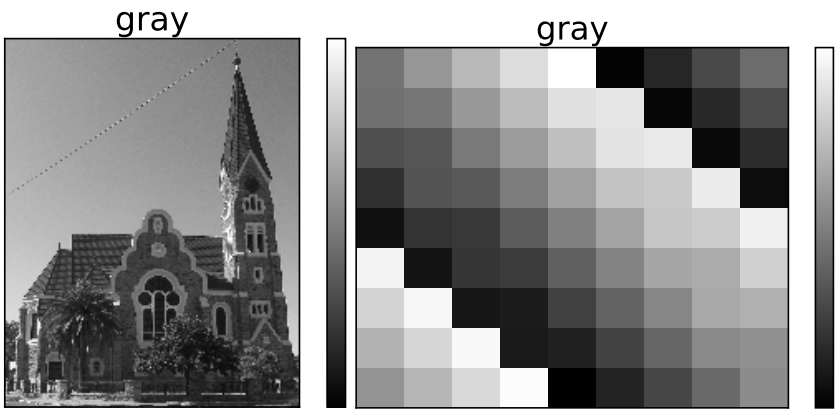
$$\begin{pmatrix} 47 & 58 & 69 & 80 & 1 & 12 & 23 & 34 & 45 \\ 57 & 68 & 79 & 9 & 11 & 22 & 33 & 44 & 46 \\ 67 & 78 & 8 & 10 & 21 & 32 & 43 & 54 & 56 \\ 77 & 7 & 18 & 20 & 31 & 42 & 53 & 55 & 66 \\ 6 & 17 & 19 & 30 & 41 & 52 & 63 & 65 & 76 \\ 16 & 27 & 29 & 40 & 51 & 62 & 64 & 75 & 5 \\ 26 & 28 & 39 & 50 & 61 & 72 & 74 & 4 & 15 \\ 36 & 38 & 49 & 60 & 71 & 73 & 3 & 14 & 25 \\ 37 & 48 & 59 & 70 & 81 & 2 & 13 & 24 & 35 \end{pmatrix}$$

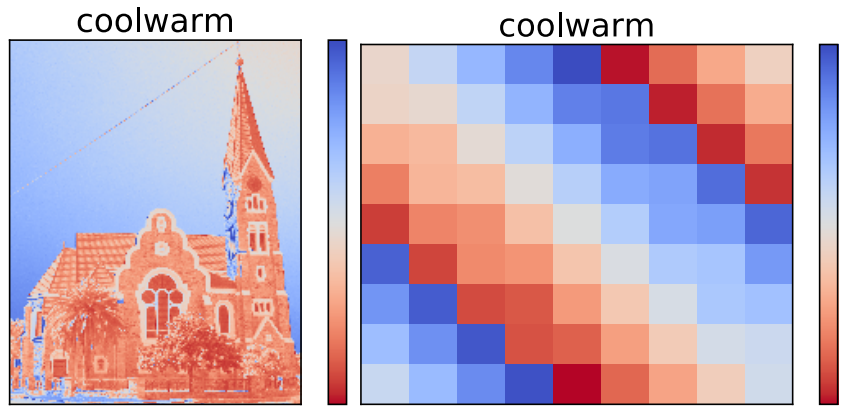

jet
jet

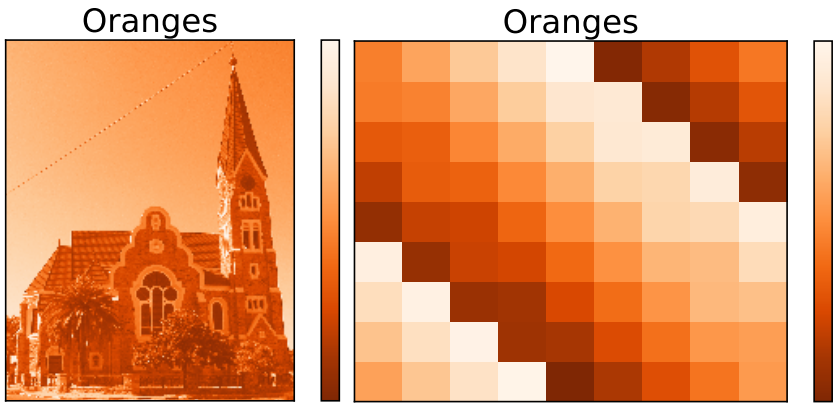

hot
hot

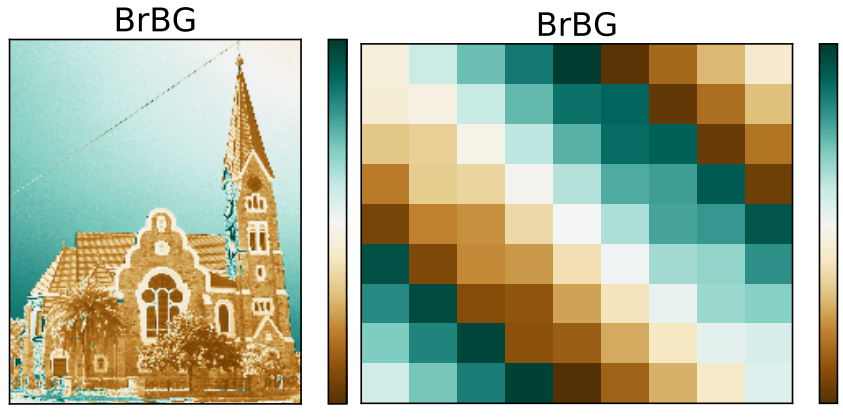

flag
flag


cool
cool

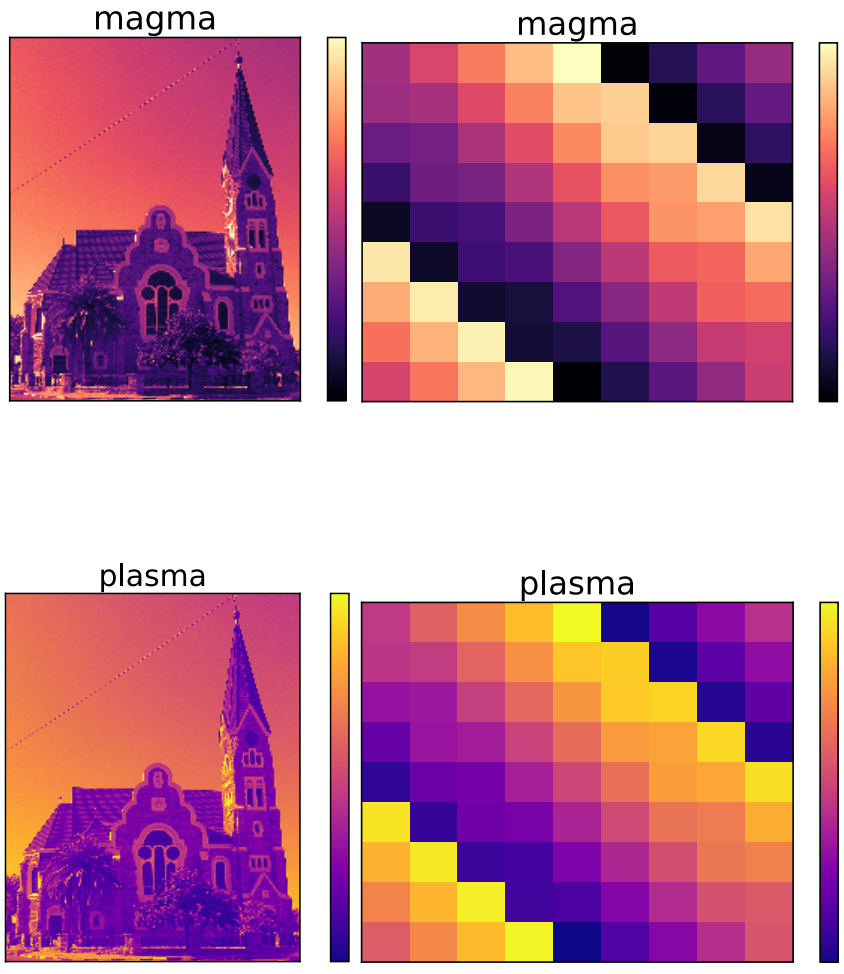

gray
gray


coolwarm
coolwarm


Oranges
Oranges


BrBG
BrBG

# *Perceptually Uniform Sequencial Colormaps*



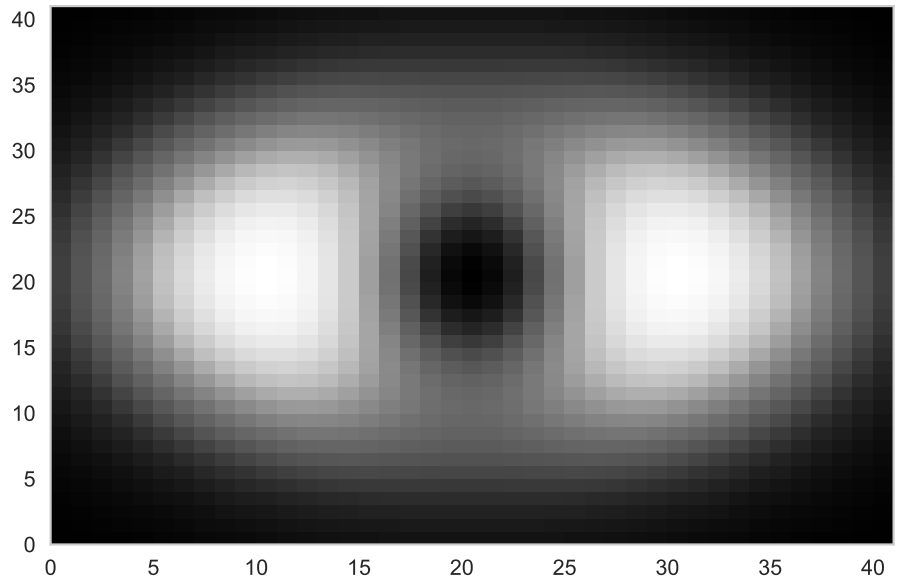
# Heatmaps with pcolor

$$f(x, y) = -\left(x^2 + 3y^2\right) e^{-x^2-y^2}$$

```
xs = np.arange(-2, 2.2, 0.1)
ys = np.arange(-2, 2.2, 0.1)

z = []
for y in ys[:-1]:
    z.append([])
    for x in xs[:-1]:
        z[-1].append(f(x, y))
```

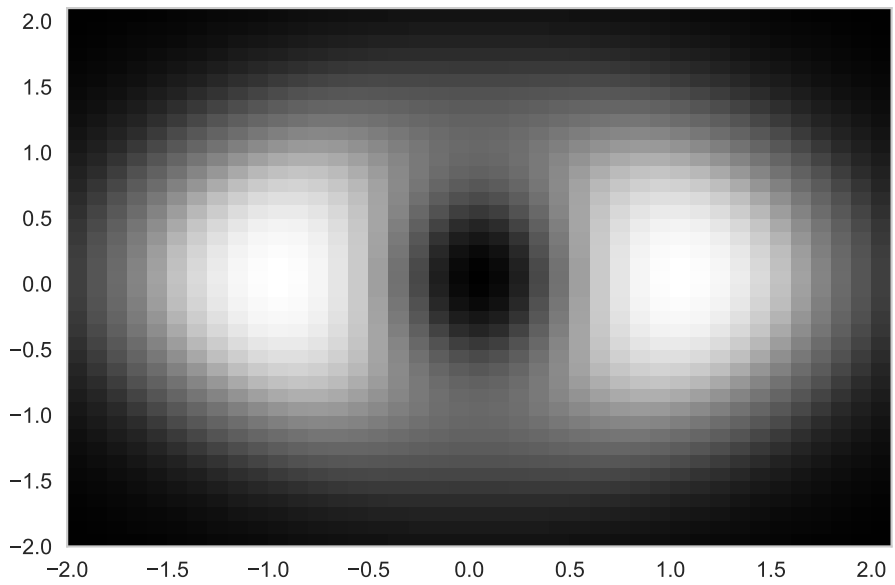
$$\begin{pmatrix}
-0.005 & -0.007 & -0.010 & \cdots & -0.010 & -0.007 & -0.005 \\
-0.008 & -0.011 & -0.014 & \cdots & -0.014 & -0.011 & -0.008 \\
-0.011 & -0.015 & -0.020 & \cdots & -0.020 & -0.015 & -0.011 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
-0.011 & -0.015 & -0.020 & \cdots & -0.020 & -0.015 & -0.011 \\
-0.008 & -0.011 & -0.014 & \cdots & -0.014 & -0.011 & -0.008 \\
-0.005 & -0.007 & -0.010 & \cdots & -0.010 & -0.007 & -0.005
\end{pmatrix}$$

```
fig, ax = plt.subplots(1)

hm = ax.pcolor(z)
```



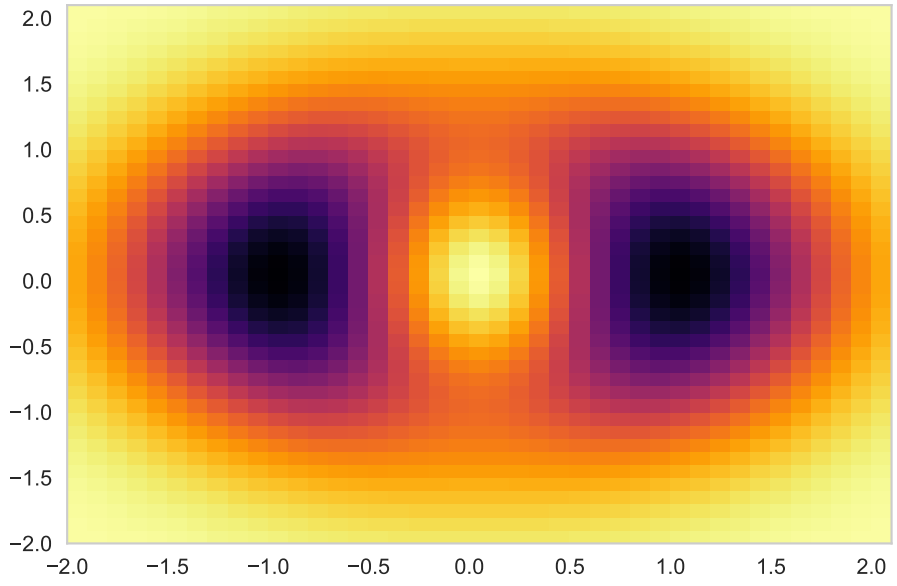

```
plt.show()
```

```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z)
```



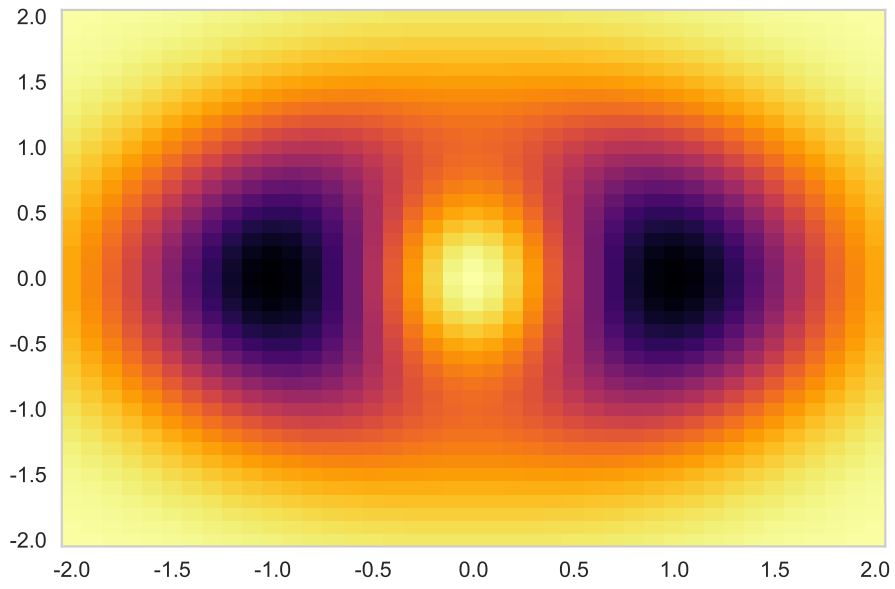

```
plt.show()
```

```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z, cmap='inferno')
```



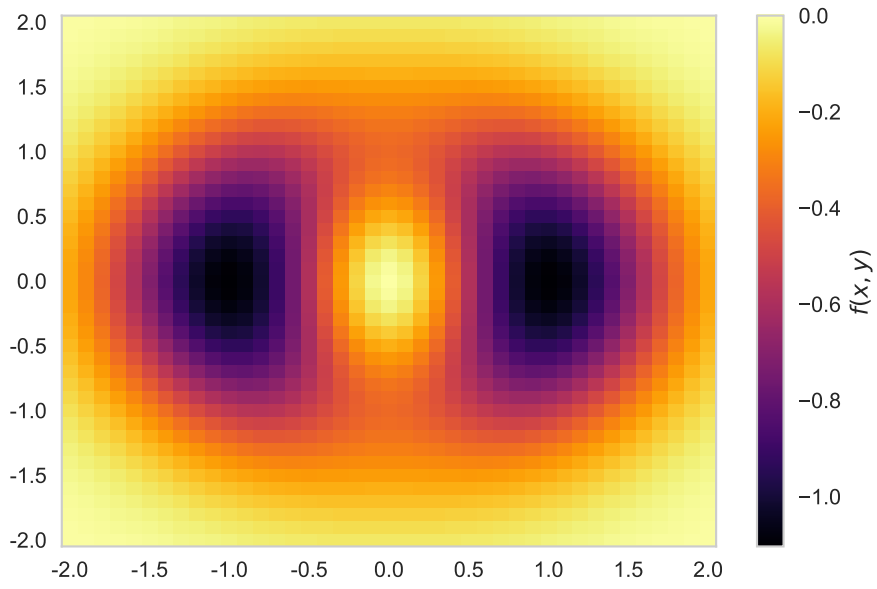

```
plt.show()
```

```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z, cmap='inferno')

ticks = np.linspace(-2, 2, 9)
ax.set_xticks([i+0.05 for i in ticks])
ax.set_yticks([i+0.05 for i in ticks])
ax.set_xticklabels(ticks)
ax.set_yticklabels(ticks)
```



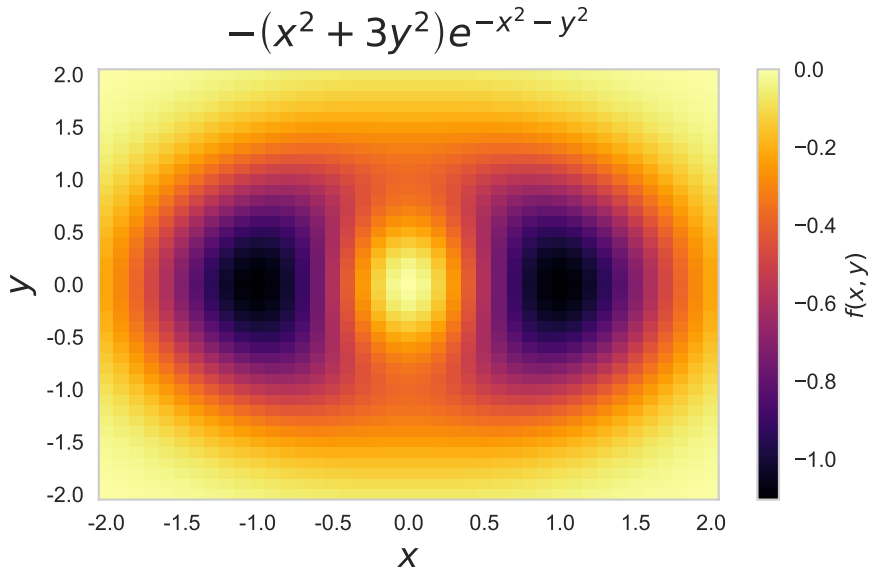

```
plt.show()
```

```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z, cmap='inferno')

ticks = np.linspace(-2, 2, 9)
ax.set_xticks([i+0.05 for i in ticks])
ax.set_yticks([i+0.05 for i in ticks])
ax.set_xticklabels(ticks)
ax.set_yticklabels(ticks)

cbar = fig.colorbar(hm)
cbar.set_label(r"$f(x, y)$")
```

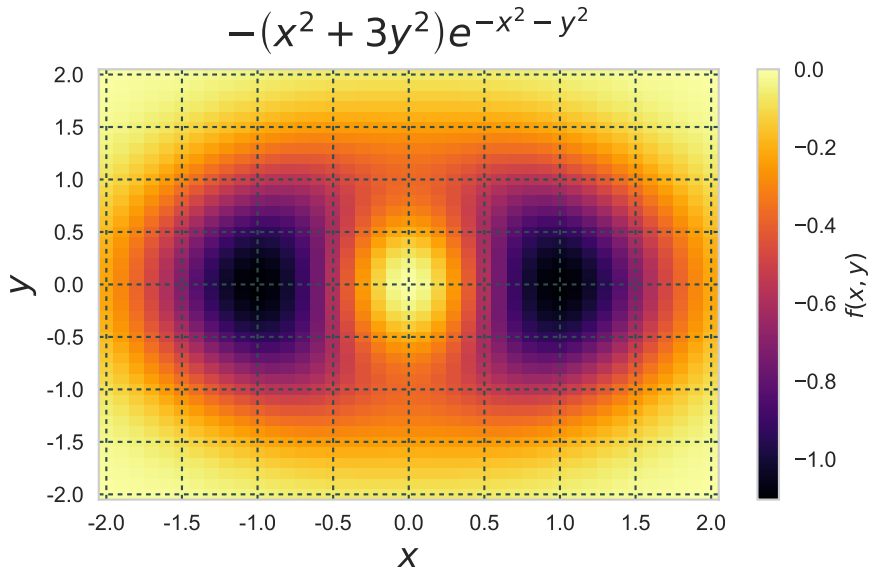



```
plt.show()
```

```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z, cmap='inferno')

ticks = np.linspace(-2, 2, 9)
ax.set_xticks([i+0.05 for i in ticks])
ax.set_yticks([i+0.05 for i in ticks])
ax.set_xticklabels(ticks)
ax.set_yticklabels(ticks)

cbar = fig.colorbar(hm)
cbar.set_label(r"$f(x, y)$")

title = r"$\left(x^2+3y^2\right)e^{-x^2-y^2}$"
ax.set_title(title, fontsize=18)
ax.set_xlabel(r"$x$", fontsize=16)
ax.set_ylabel(r"$y$", fontsize=16)



plt.show()
```



```
fig, ax = plt.subplots(1)
X,Y = np.meshgrid(xs, ys)
hm = ax.pcolor(X, Y, z, cmap='inferno', zorder=0)

ticks = np.linspace(-2, 2, 9)
ax.set_xticks([i+0.05 for i in ticks])
ax.set_yticks([i+0.05 for i in ticks])
ax.set_xticklabels(ticks)
ax.set_yticklabels(ticks)

cbar = fig.colorbar(hm)
cbar.set_label(r"$f(x, y)$")

title = r"$\left(x^2+3y^2\right)e^{-x^2-y^2}$"
ax.set_title(title, fontsize=18)
ax.set_xlabel(r"$x$", fontsize=16)
ax.set_ylabel(r"$y$", fontsize=16)

ax.grid(which='major', color='darkslategrey',
        linestyle=':', linewidth=1, axis='both')

plt.show()
```




- Cardiff University Phoenix Project
- Cardiff School of Mathematics
- PyCon Namibia 2017

- matplotlib
- numpy
- seaborn
- jupyter

www.geraintianpalmer.org.uk/talks

@GeraintPalmer

# Links

- http://matplotlib.org/api/axes_api.html
- http://matplotlib.org/api/pyplot_summary.html
- http://matplotlib.org/examples/index.html
- http://matplotlib.org/examples/color/colormaps_reference.html
- https://www.youtube.com/watch?v=k_lvjRCOpJk&feature=youtu.be&list=PLpX1jXuNTXGrjl6CxJ6Cly1GKO1su9yeD
- https://www.youtube.com/watch?v=k_lvjRCOpJk&feature=youtu.be&list=PLpX1jXuNTXGrjl6CxJ6Cly1GKO1su9yeD
- https://vincentarelbundock.github.io/Rdatasets/datasets.html
- http://www.databaseolympics.com/
- https://en.wikipedia.org/wiki/Magic_square