Matplotlib Tutorial

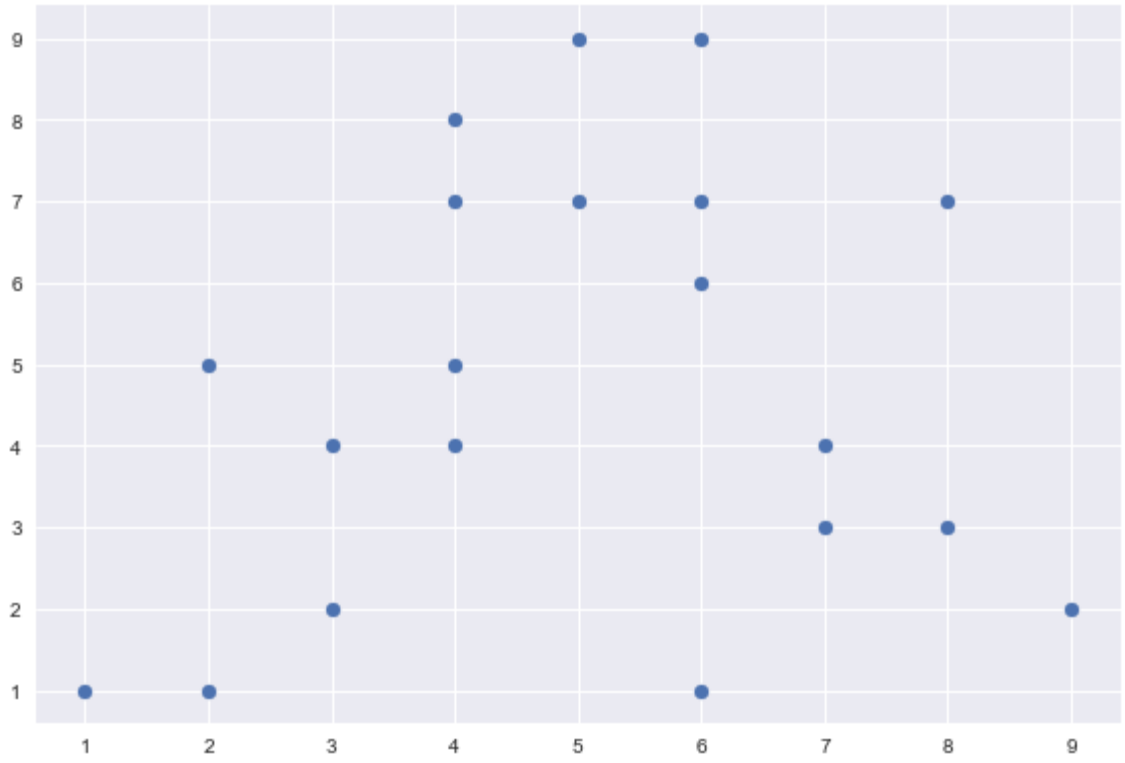
Part 7

`Creating Scatterplot`

Notes and codes

In [1]:

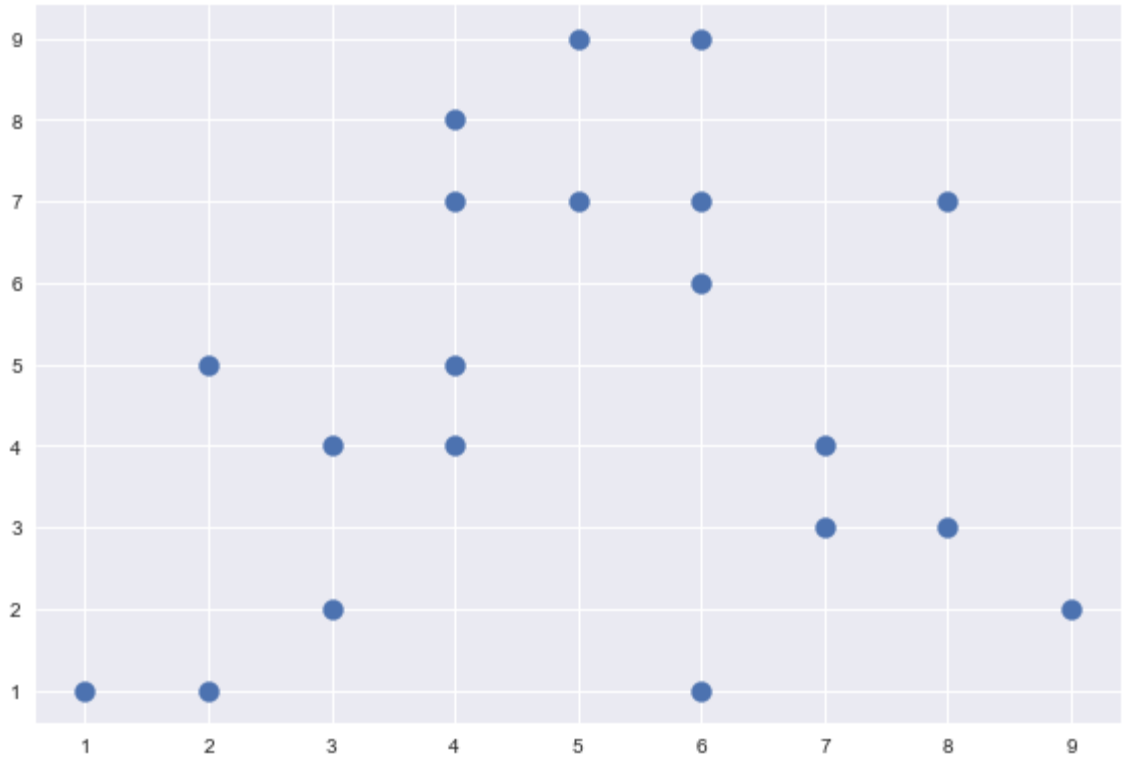
```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

plt.scatter(x,y)

# colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```



Change the dot size

In [2]:

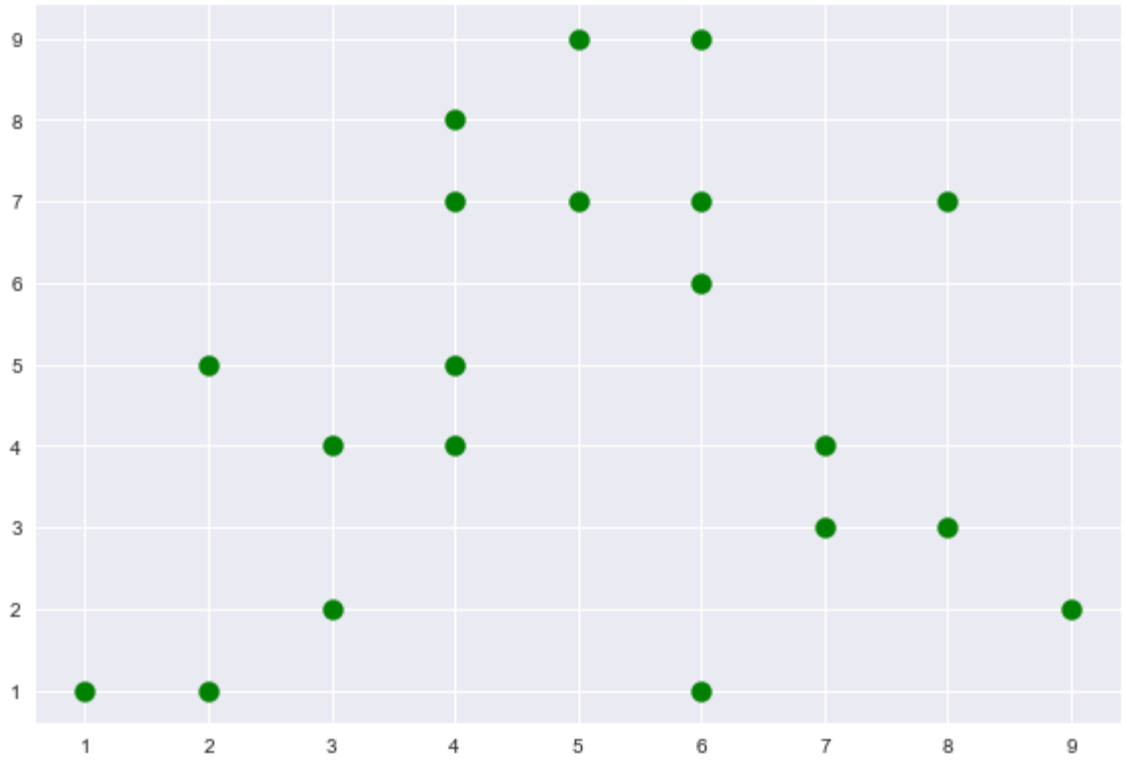
```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

plt.scatter(x,y, s=100)

# colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```



Change the color

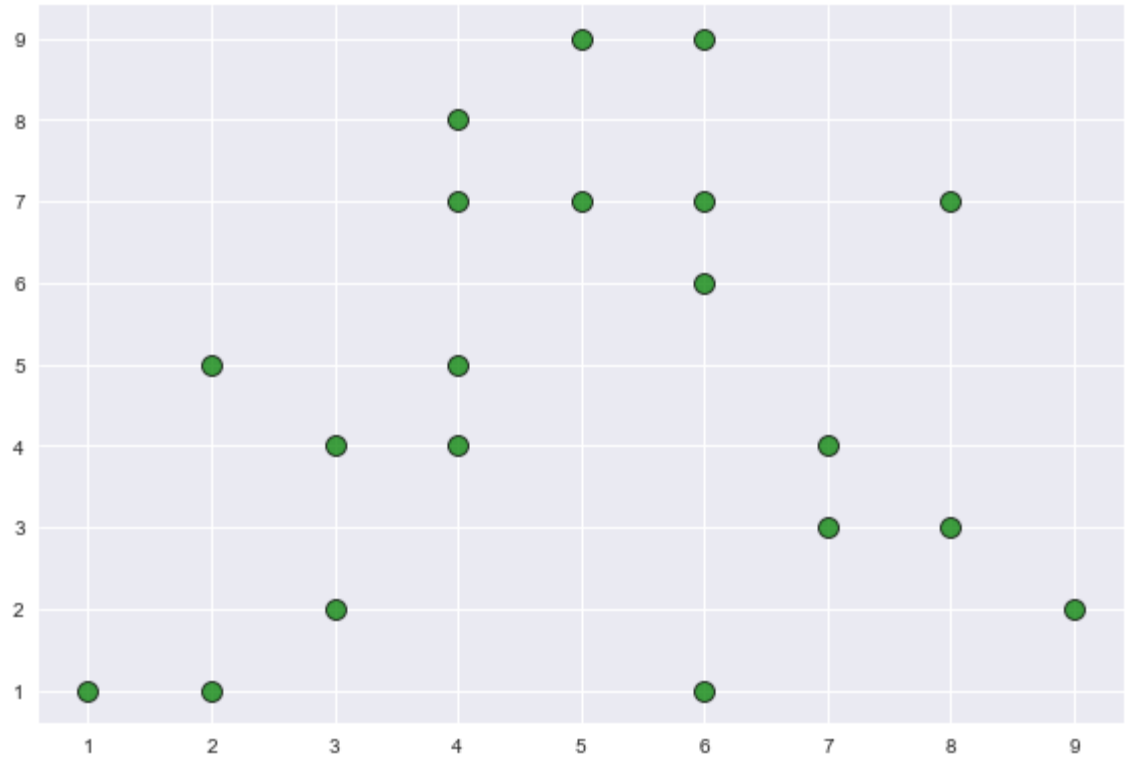
In [3]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

plt.scatter(x,y, s=100, c='green')

# colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```



In [4]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

plt.scatter(x,y, s=100, c='green', edgecolor='black', linewidth=1, alpha=0.75)

# colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```

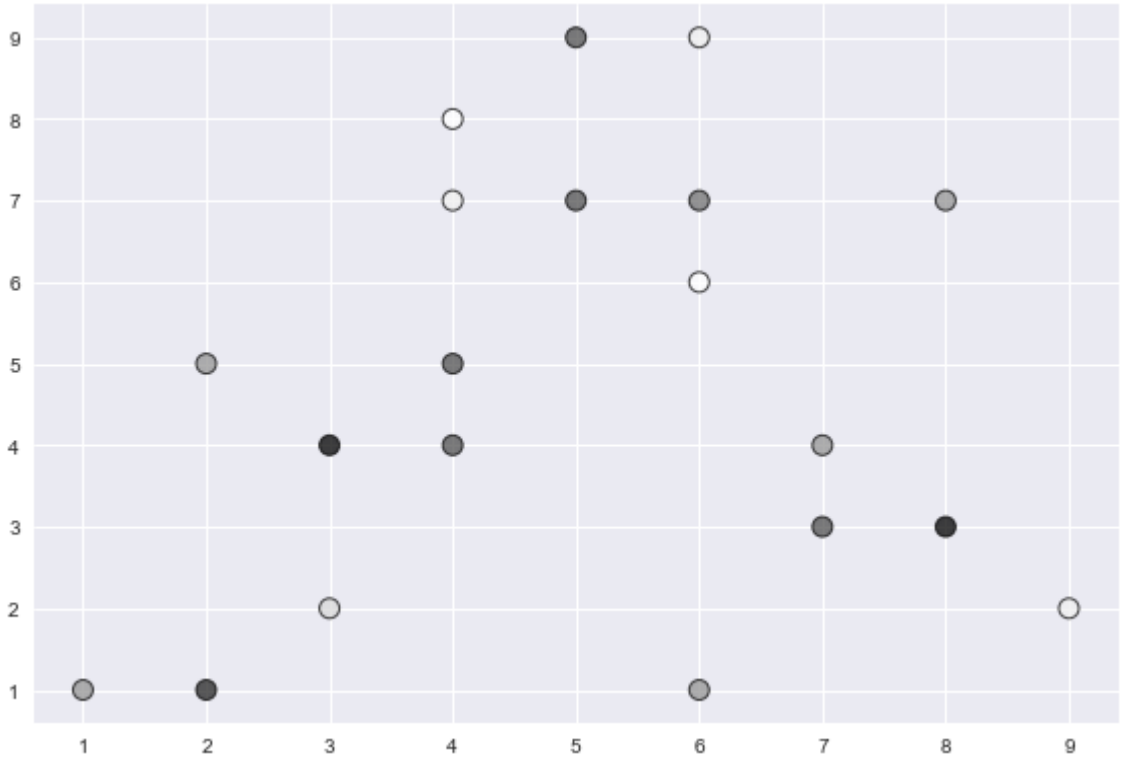


Adding multiple colors

- This will allow you to include more data

In [5]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

plt.scatter(x,y, s=100, c=colors, edgecolor='black', linewidth=1, alpha=0.75)

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```

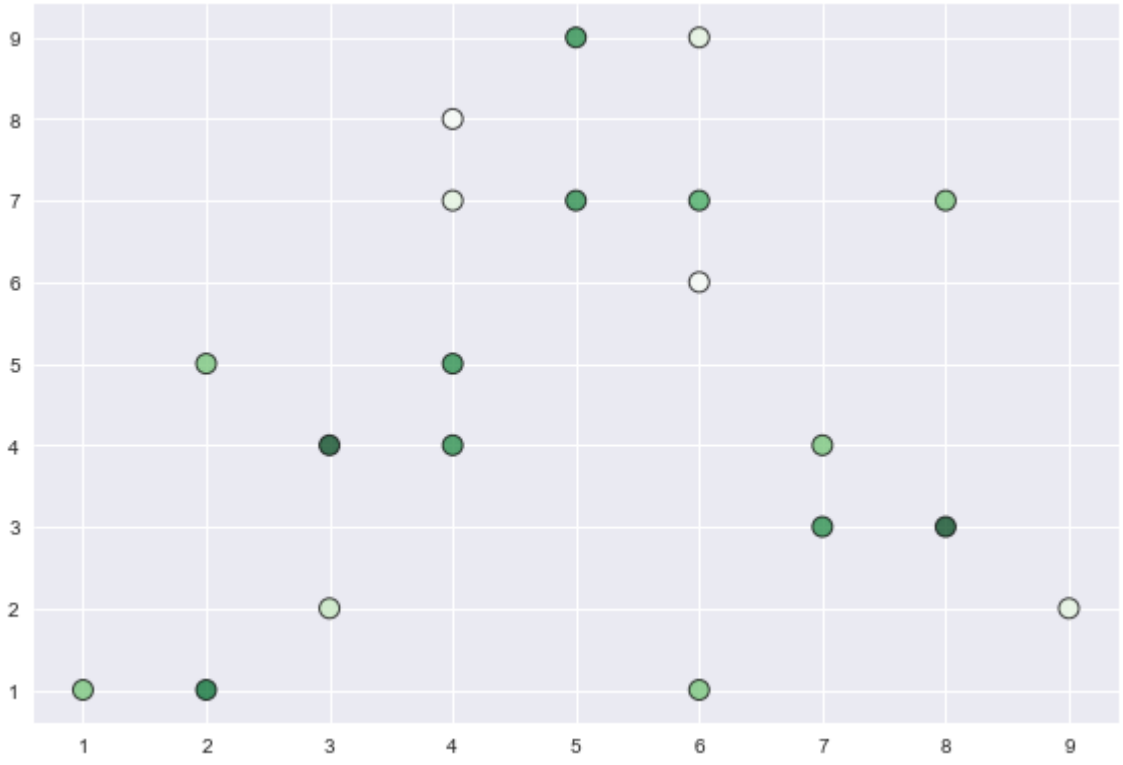


Add cmap to know the intensity

- Lighter shades are closer to 0

In [6]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

plt.scatter(x,y, s=100, c=colors, cmap='Greens', edgecolor='black', linewidth=

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```

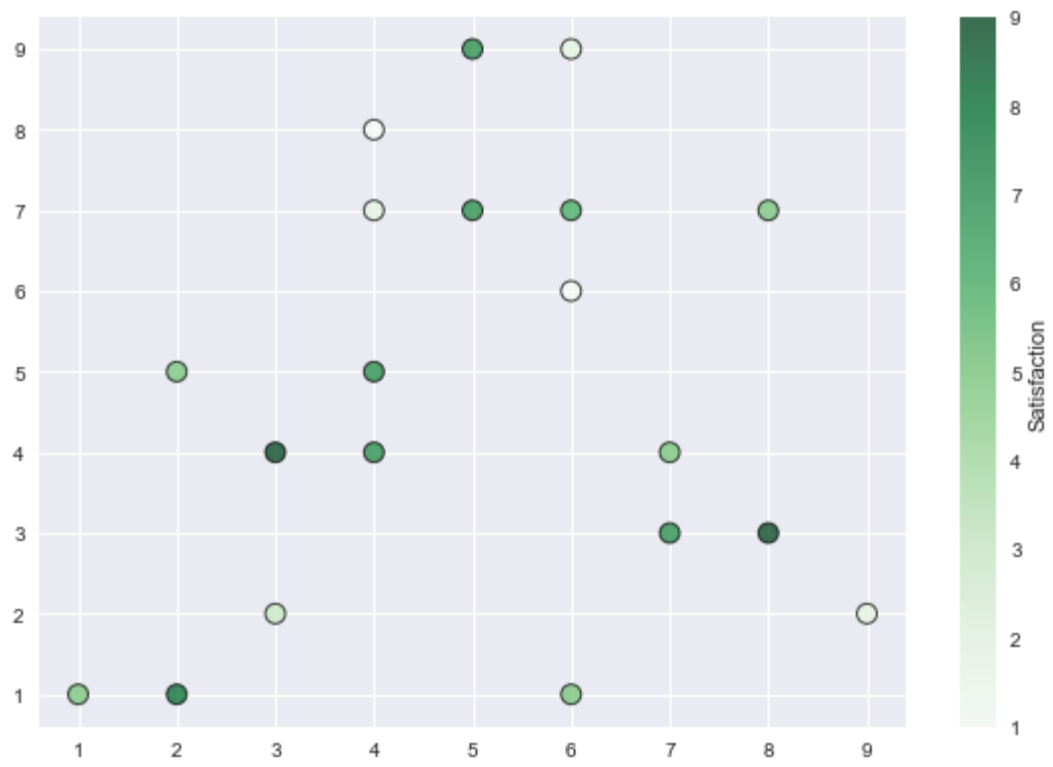


Add a label in color map

- So people would know what these colors `represent`.
- Add a `color bar legend`.
- For instance, we can label it `satisfaction`. This would mean that those in lighter shades are less satisfied and those in darker shades are more satisfied.

In [7]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]

plt.scatter(x,y, s=100, c=colors, cmap='Greens', edgecolor='black', linewidth=
cbar = plt.colorbar()
cbar.set_label('Satisfaction')

# sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
#          538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```

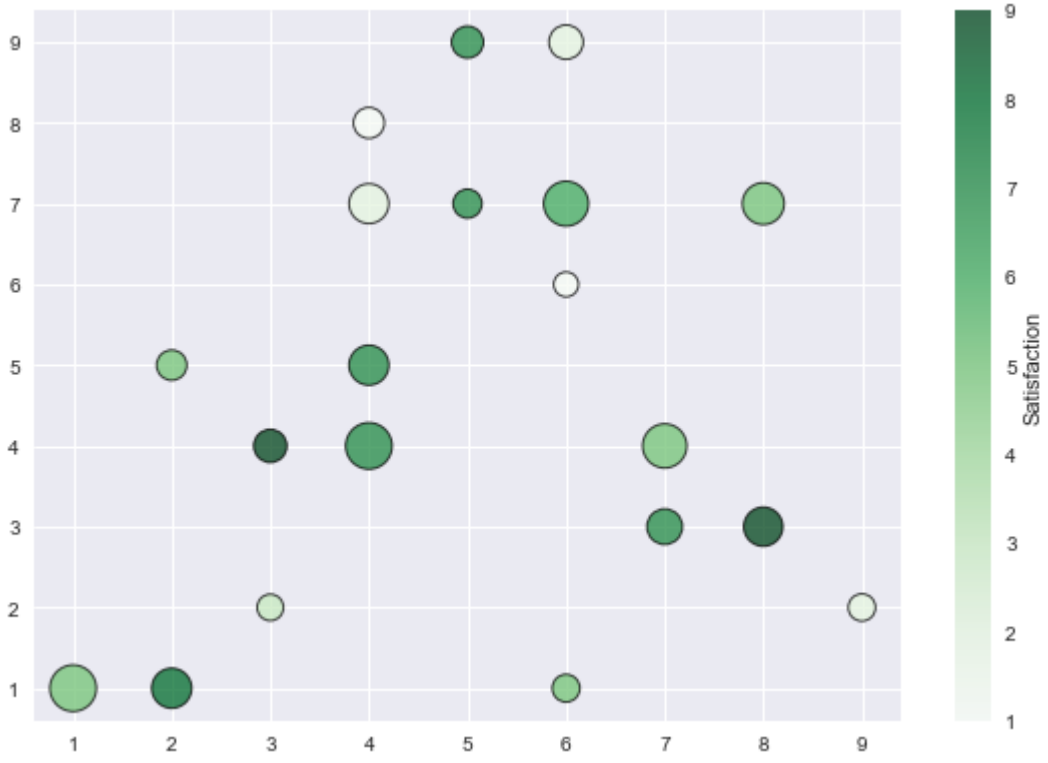


We can change the size of the dots to represent changes

- The bigger the size of the dot, greater level of satisfaction

In [8]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

x = [5, 7, 8, 5, 6, 7, 9, 2, 3, 4, 4, 4, 2, 6, 3, 6, 8, 6, 4, 1]
y = [7, 4, 3, 9, 1, 3, 2, 5, 2, 4, 8, 7, 1, 6, 4, 9, 7, 7, 5, 1]

colors = [7, 5, 9, 7, 5, 7, 2, 5, 3, 7, 1, 2, 8, 1, 9, 2, 5, 6, 7, 5]
sizes = [209, 486, 381, 255, 191, 315, 185, 228, 174,
         538, 239, 394, 399, 153, 273, 293, 436, 501, 397, 539]

plt.scatter(x,y, s=sizes, c=colors, cmap='Greens', edgecolor='black', linewidt
cbar = plt.colorbar()
cbar.set_label('Satisfaction')


# data = pd.read_csv('2019-05-31-data.csv')
# view_count = data['view_count']
# likes = data['likes']
# ratio = data['ratio']

# plt.title('Trending YouTube Videos')
# plt.xlabel('View Count')
# plt.ylabel('Total Likes')

plt.tight_layout()

plt.show()
```



Using real-life data

Data from Top 200 Trending Youtube Videos

In [9]:
```
data = pd.read_csv('2019-05-31-data.txt')
data
```

Out[9]:

	view_count	likes	ratio
0	8036001	324742	96.91
1	9378067	562589	98.19
2	2182066	273650	99.38
3	6525864	94698	96.25
4	9481284	582481	97.22
...	...	...	...
195	1069693	3970	90.66
196	590760	70454	99.18
197	319347	1208	92.50
198	27594927	1351963	96.40
199	26993425	437561	97.42

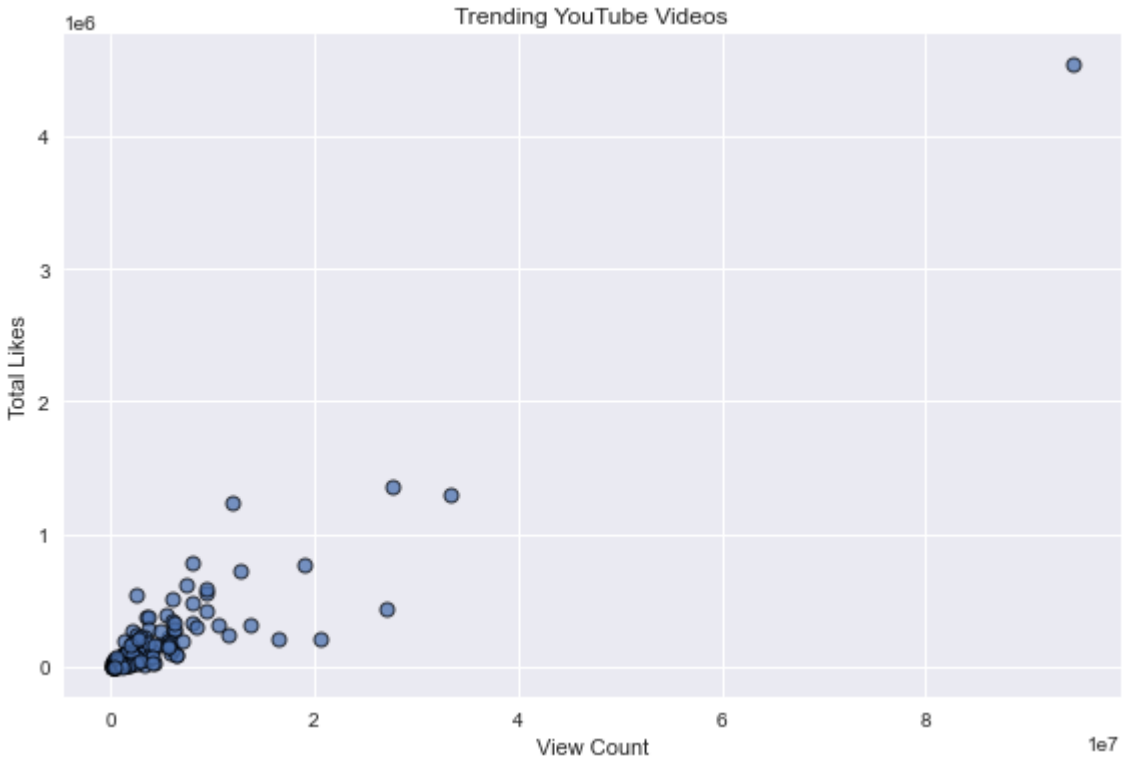
200 rows × 3 columns

Plot the relationship between Youtube `view count` and `likes`

- You will that there wil be an outlier
- But this can be handled by transforming it to logarithmic function, in order to lessen the skewness of the plot

In [10]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

data = pd.read_csv('2019-05-31-data.txt')
view_count = data['view_count']
likes = data['likes']
ratio = data['ratio']

plt.title('Trending YouTube Videos')
plt.xlabel('View Count')
plt.ylabel('Total Likes')

plt.scatter(view_count,likes, edgecolor='black', linewidth=1, alpha=0.75)

plt.tight_layout()

plt.show()
```

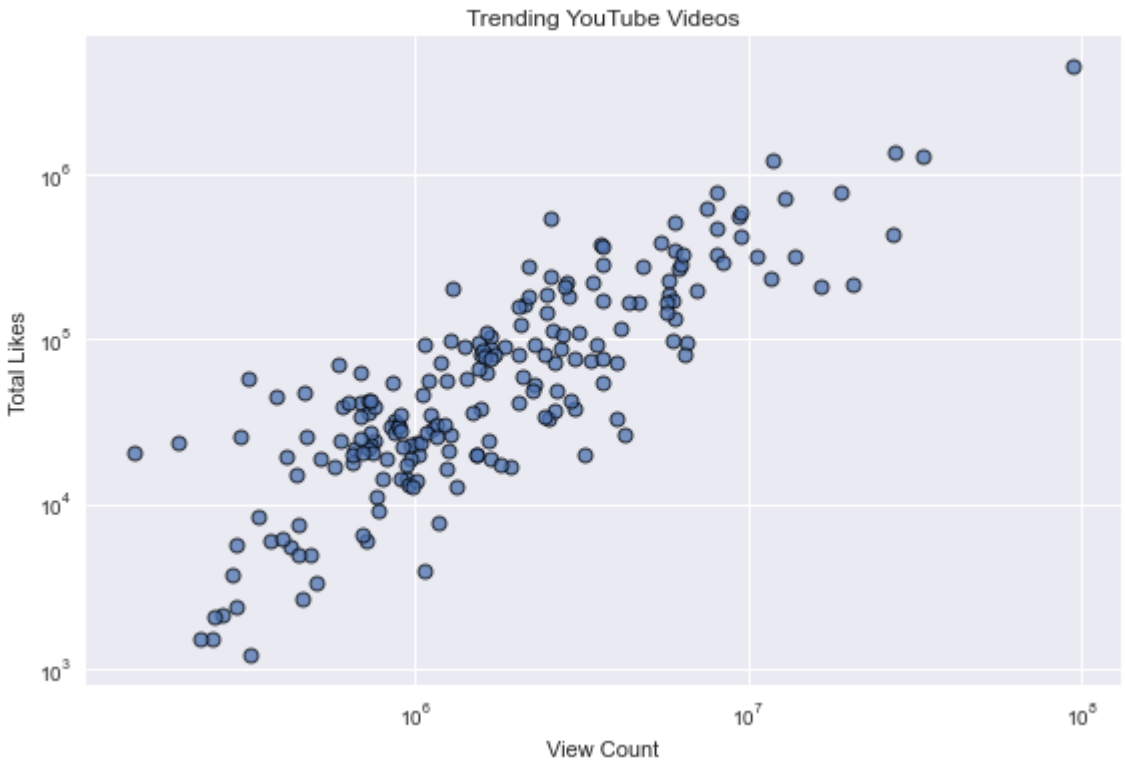


Transforming to log scale

- In this way, the outliers don't skew the plot so much
- We will see know the correlation between our two variables of interest. That is, the more views the video has, the more likes it generates. Hence, `positively related`.

In [11]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

data = pd.read_csv('2019-05-31-data.txt')
view_count = data['view_count']
likes = data['likes']
ratio = data['ratio']

plt.title('Trending YouTube Videos')
plt.xlabel('View Count')
plt.ylabel('Total Likes')

plt.scatter(view_count,likes, edgecolor='black', linewidth=1, alpha=0.75)

plt.xscale('log')
plt.yscale('log')

plt.tight_layout()

plt.show()
```

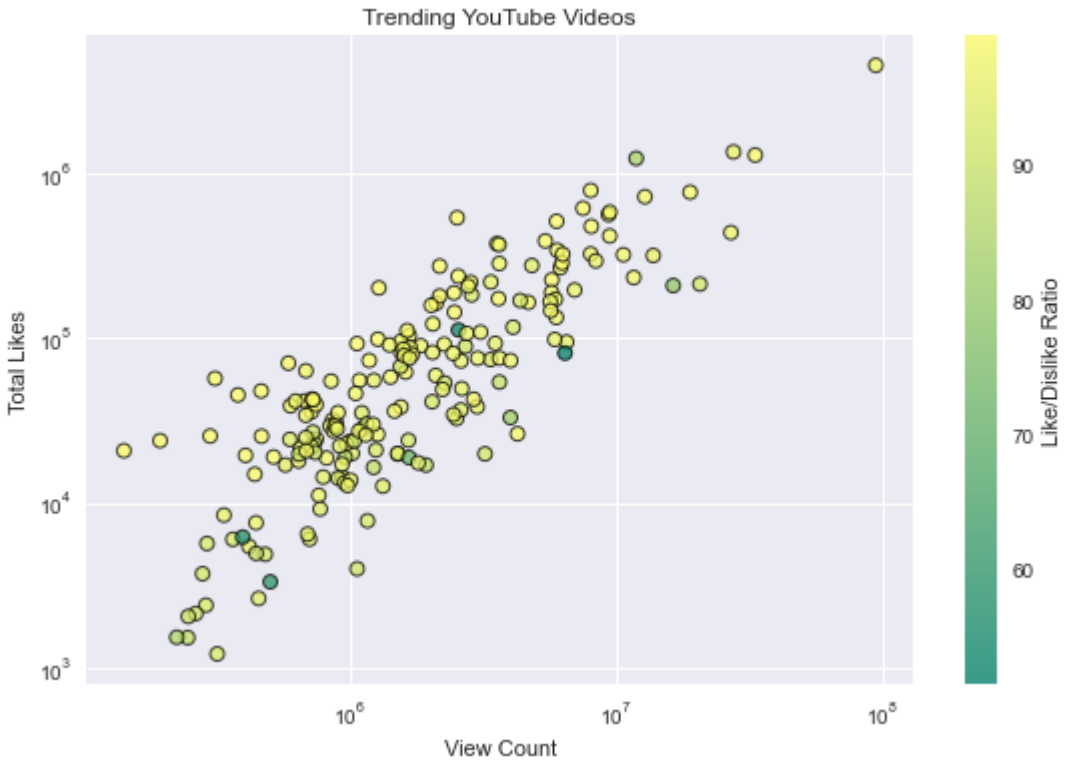


Use a different metric

- Use the like/dislike ratio
- use color maps

In [12]:

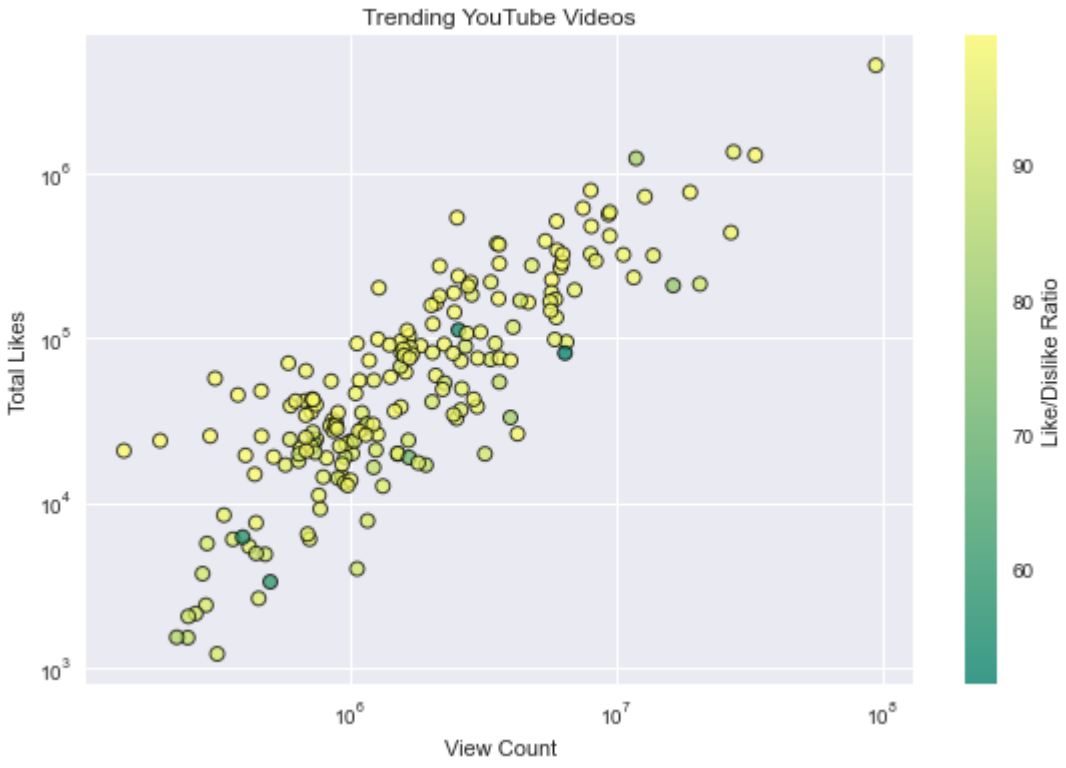
```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

data = pd.read_csv('2019-05-31-data.txt')
view_count = data['view_count']
likes = data['likes']
ratio = data['ratio']

plt.title('Trending YouTube Videos')
plt.xlabel('View Count')
plt.ylabel('Total Likes')

plt.scatter(view_count,likes, edgecolor='black', linewidth=1, alpha=0.75,
            c=ratio, cmap='summer')

cbar = plt.colorbar()
cbar.set_label('Like/Dislike Ratio')

plt.xscale('log')
plt.yscale('log')

plt.tight_layout()

plt.show()
```



Final codes

In [13]:

```
import pandas as pd
from matplotlib import pyplot as plt

plt.style.use('seaborn')

data = pd.read_csv('2019-05-31-data.txt')
view_count = data['view_count']
likes = data['likes']
ratio = data['ratio']

plt.scatter(view_count, likes, c=ratio, cmap='summer',
            edgecolor='black', linewidth=1, alpha=0.75)

cbar = plt.colorbar()
cbar.set_label('Like/Dislike Ratio')

plt.xscale('log')
plt.yscale('log')

plt.title('Trending YouTube Videos')
plt.xlabel('View Count')
plt.ylabel('Total Likes')

plt.tight_layout()
plt.savefig('Plot-Part7')

plt.show()
```



Converting notebook to html

In [14]:

```
!jupyter nbconvert --to html Matplotlib-Part7.ipynb
```

```
[NbConvertApp] Converting notebook Matplotlib-Part7.ipynb to html
[NbConvertApp] Writing 939625 bytes to Matplotlib-Part7.html
```