Introduction to Seaborn

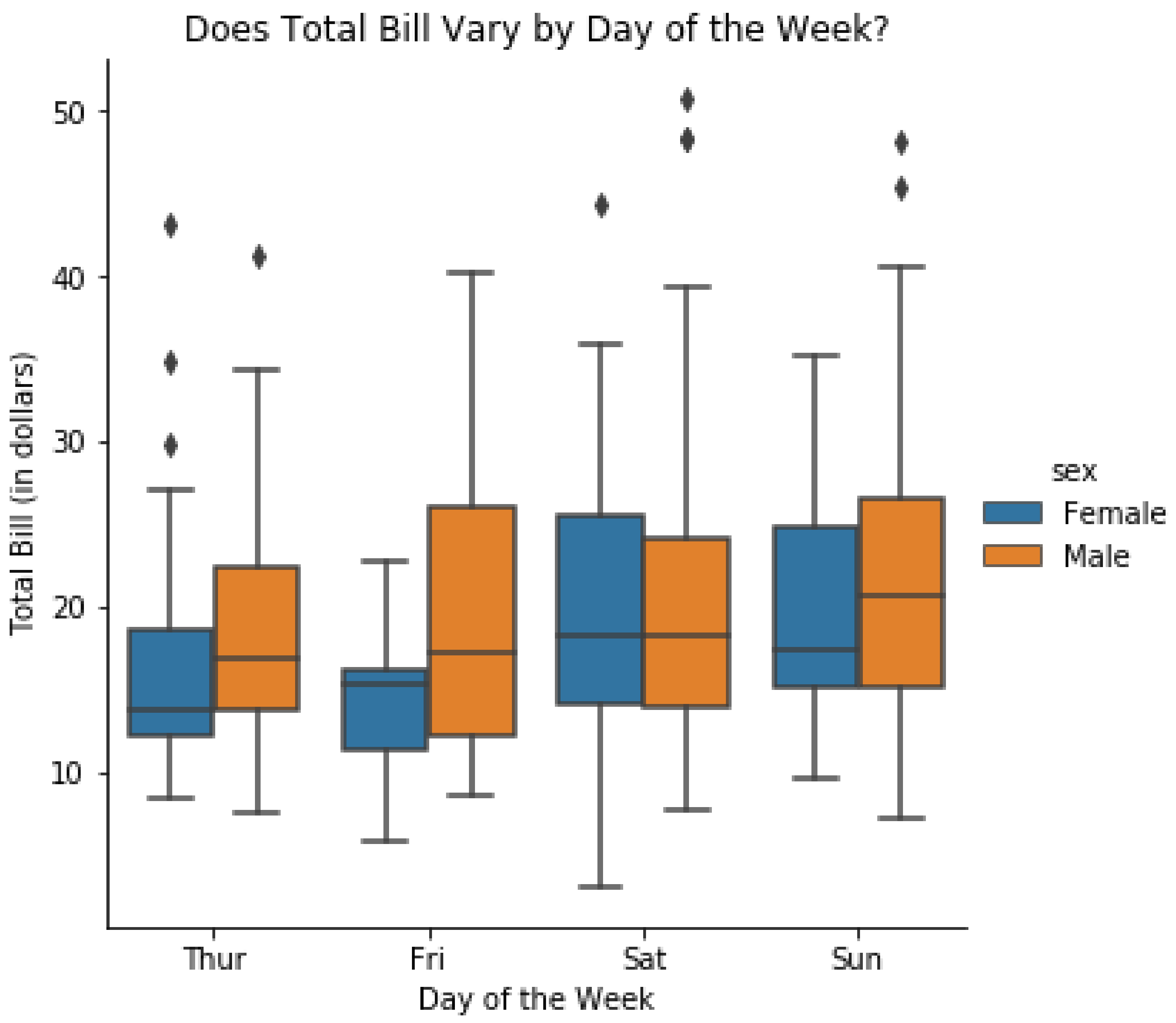
INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

Erin Case
Data Scientist

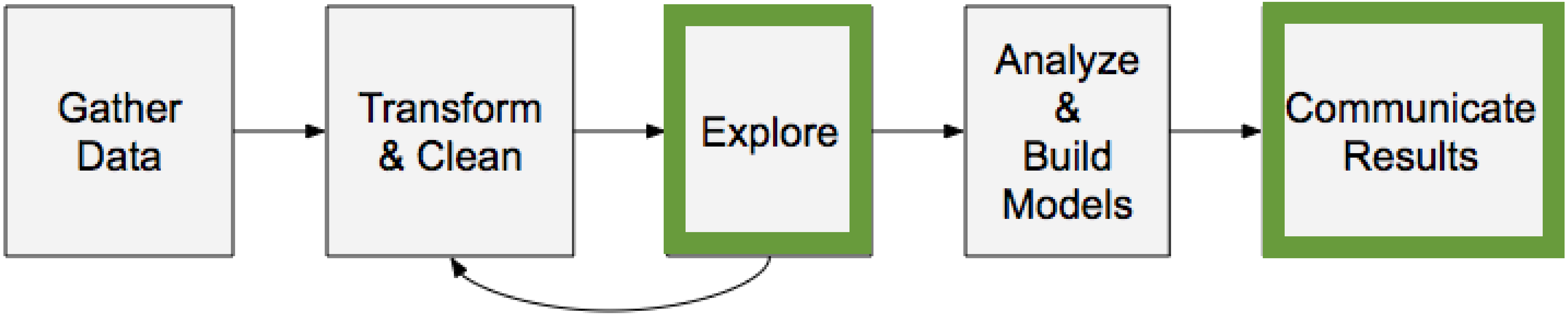
datacamp

What is Seaborn?

- Python data visualization library
- Easily create the most common types of plots



Why is Seaborn useful?



Advantages of Seaborn

- Easy to use
- Works well with `pandas` data structures
- Built on top of `matplotlib`

Getting started

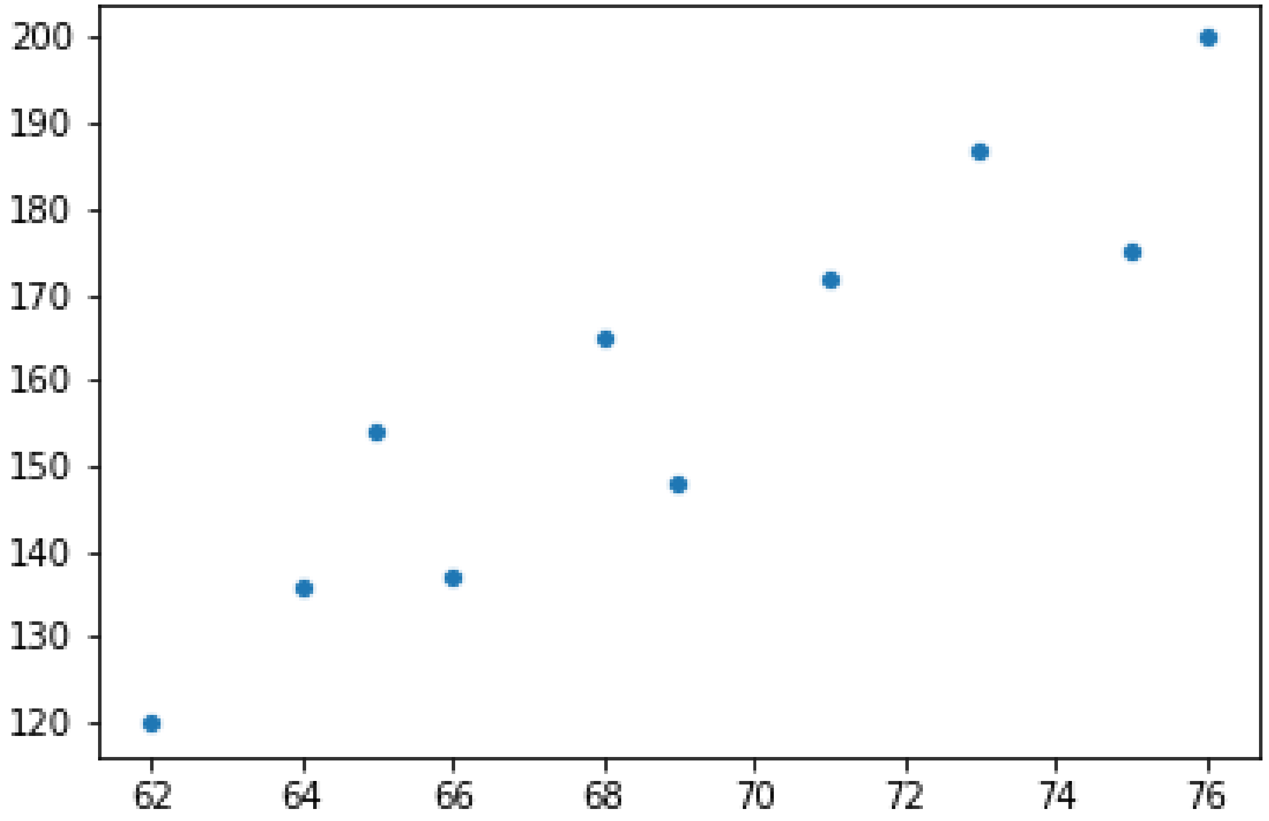
```python
import seaborn as sns
import matplotlib.pyplot as plt
```

Samuel **N**orman **S**eaborn (`sns`)

- "The West Wing" television show

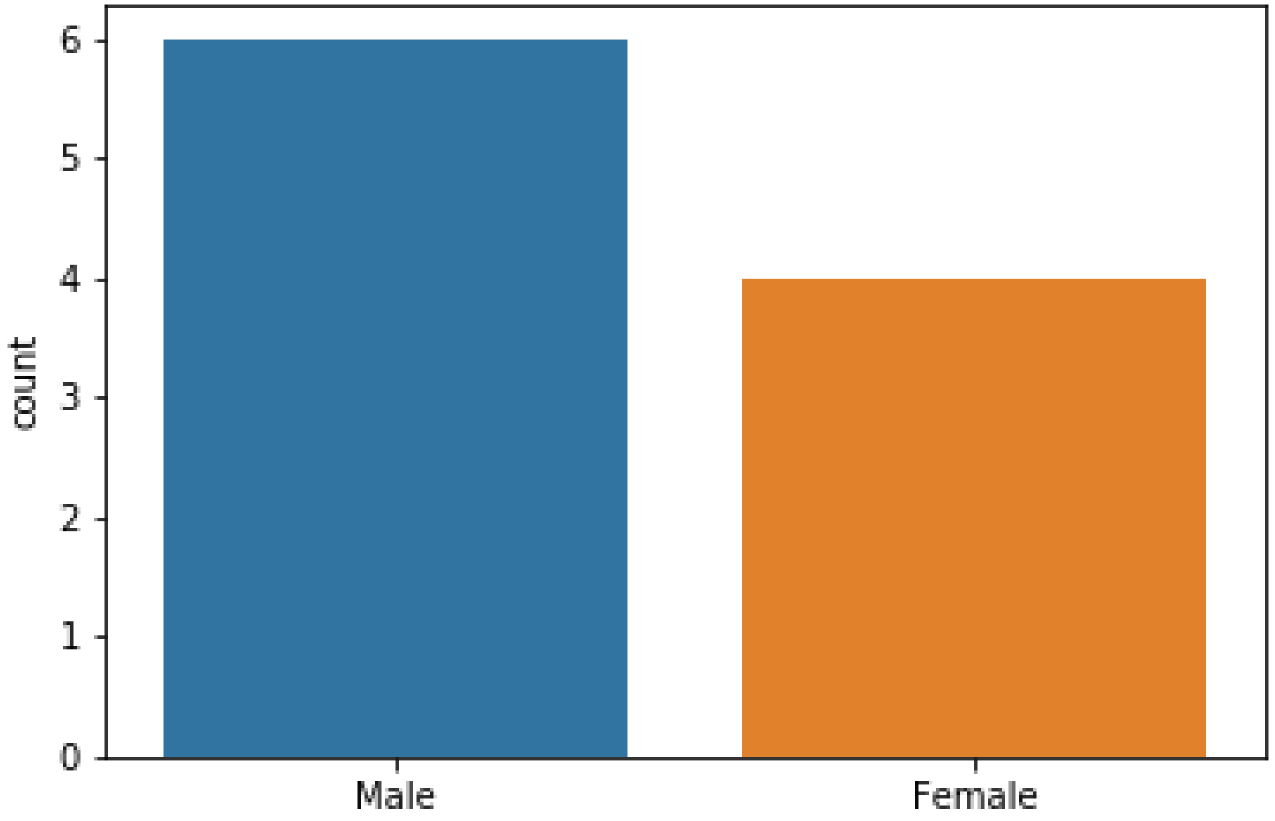
Example 1: Scatter plot

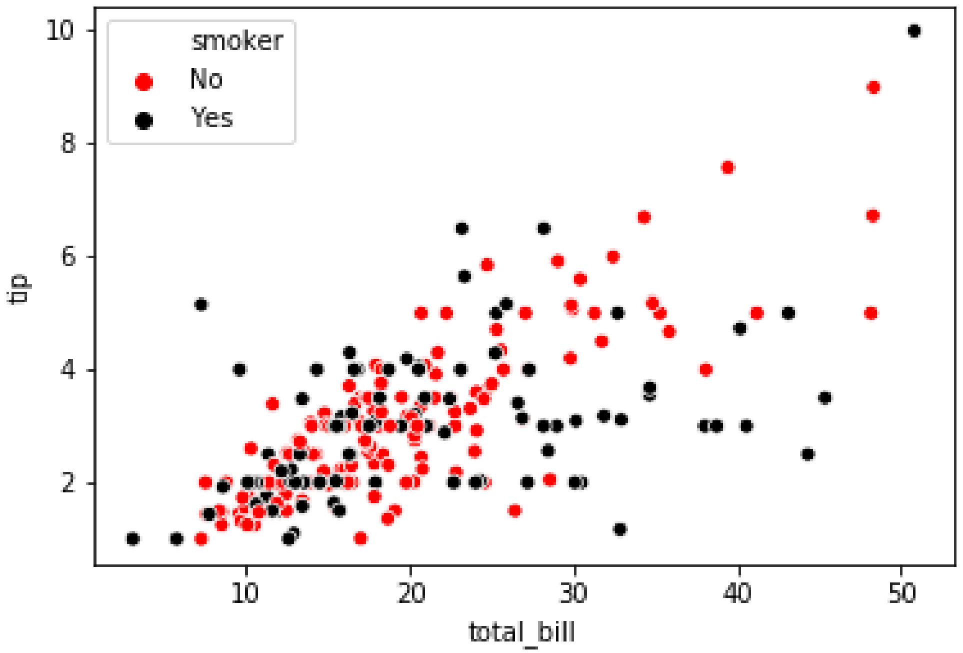
```python
import seaborn as sns
import matplotlib.pyplot as plt
height = [62, 64, 69, 75, 66,
          68, 65, 71, 76, 73]
weight = [120, 136, 148, 175, 137,
          165, 154, 172, 200, 187]
sns.scatterplot(x=height, y=weight)
plt.show()
```

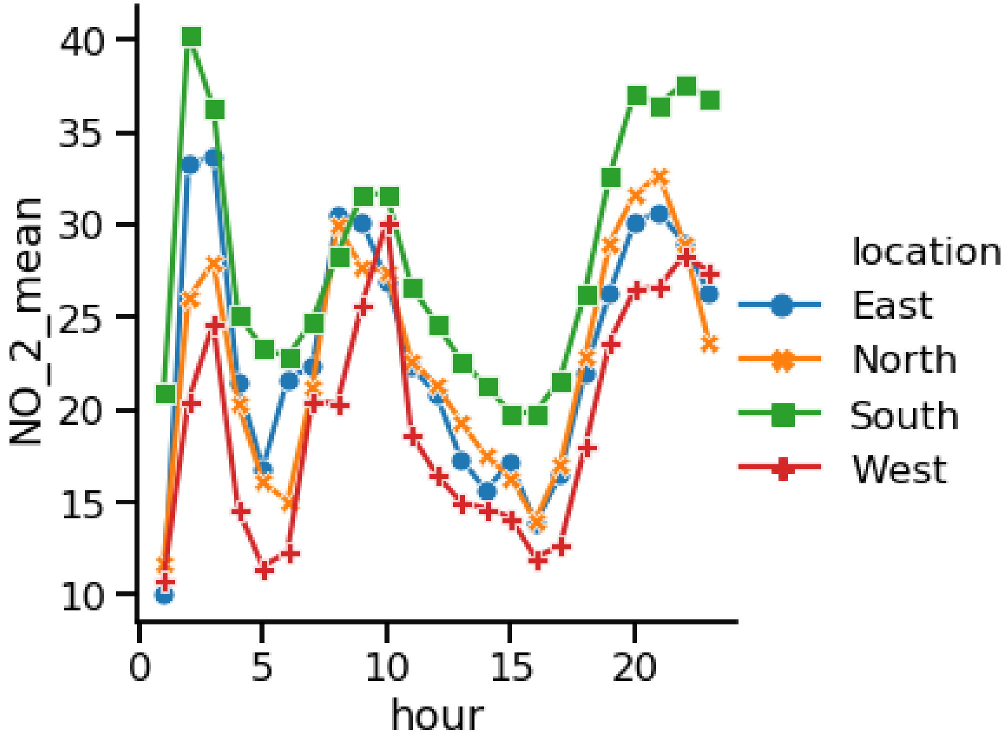


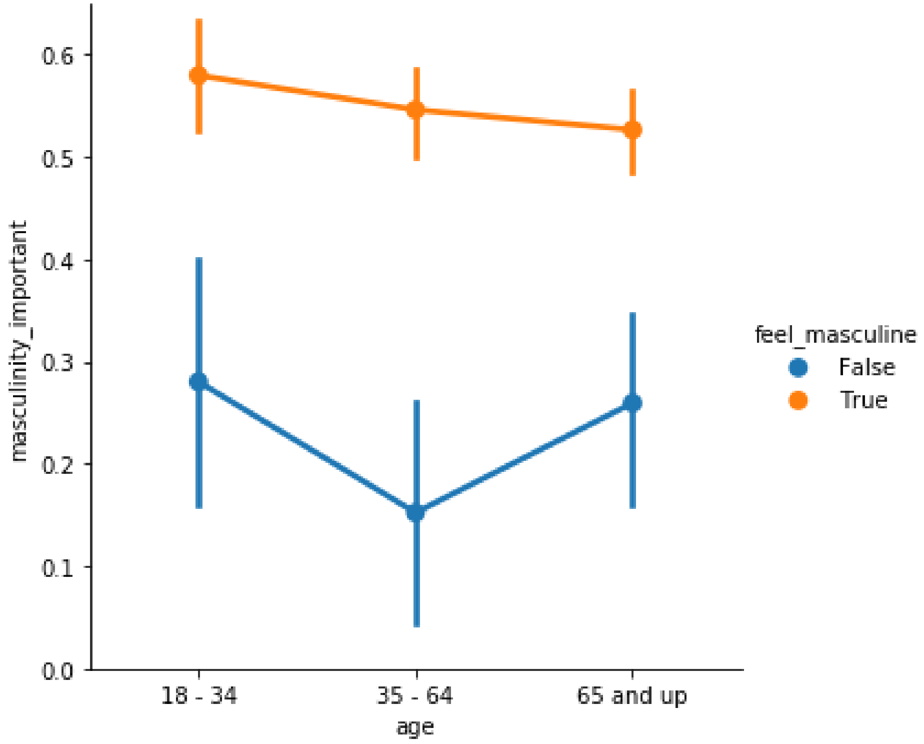
Example 2: Create a count plot

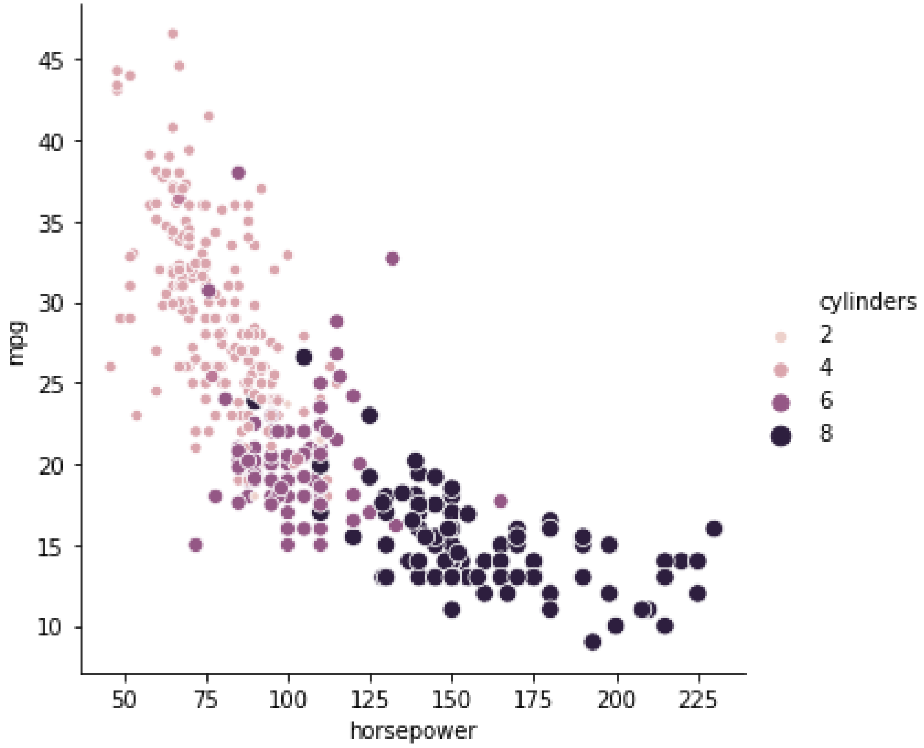
```
import seaborn as sns
import matplotlib.pyplot as plt
gender = ["Female", "Female",
          "Female", "Female",
          "Male", "Male", "Male",
          "Male", "Male", "Male"]
sns.countplot(x=gender)
plt.show()
```




smoker
No
Yes
tip
10
8
6
4
2
total_bill
10
20
30
40
50


NO_2_mean
40
35
30
25
20
15
10
hour
0
5
10
15
20
location
East
North
South
West


masculinity_important
0.6
0.5
0.4
0.3
0.2
0.1
0.0
age
18 - 34
35 - 64
65 and up
feel_masculine
False
True


mpg
45
40
35
30
25
20
15
10
horsepower
50
75
100
125
150
175
200
225
cylinders
2
4
6
8

Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

Using pandas with Seaborn

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

Erin Case
Data Scientist

What is pandas?

- Python library for data analysis
- Easily read datasets from csv, txt, and other types of files
- Datasets take the form of `DataFrame` objects

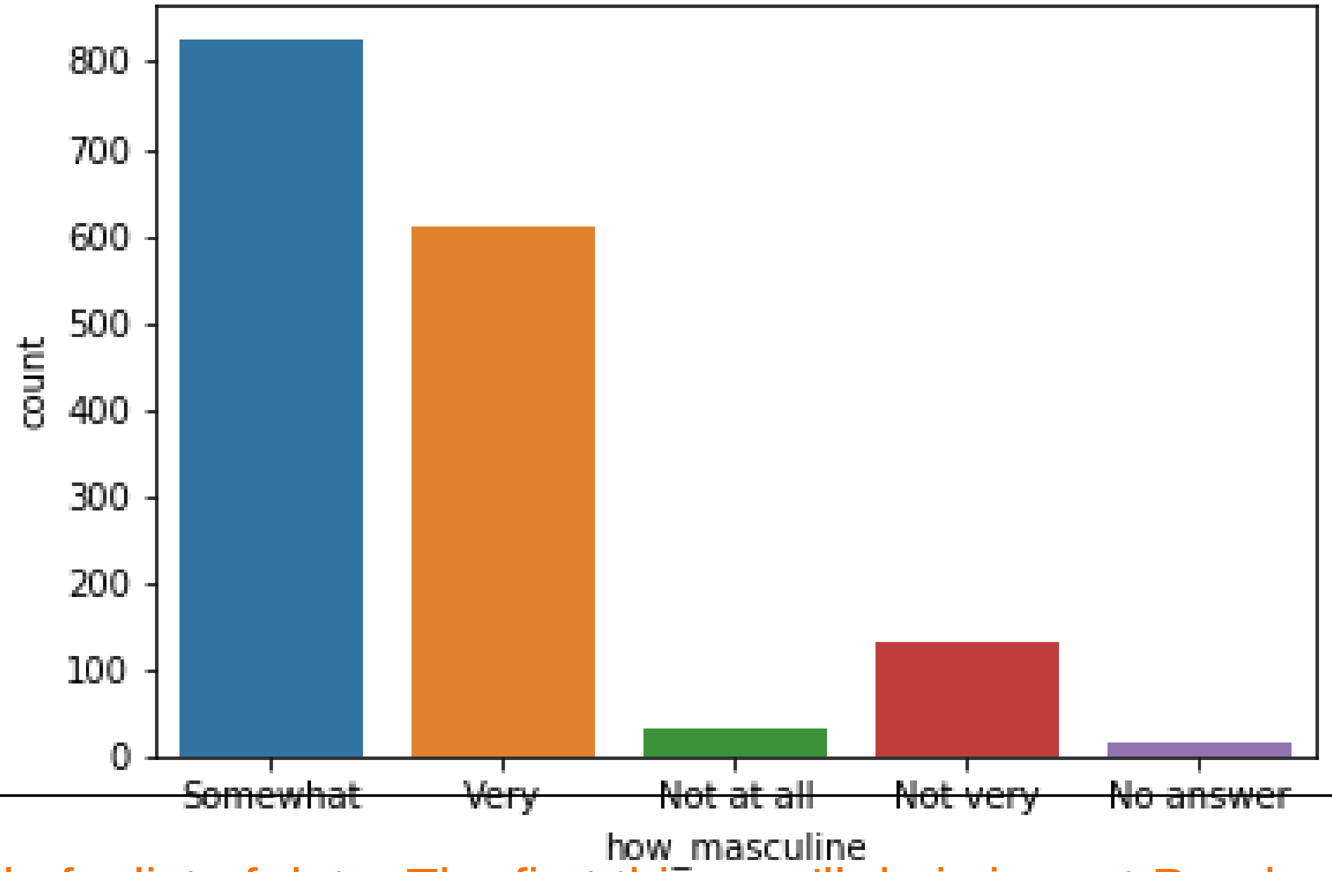
Working with DataFrames

```python
import pandas as pd
df = pd.read_csv("masculinity.csv")
df.head()
```

```
    participant_id      age    how_masculine    how_important
0                1  18 - 34         Somewhat         Somewhat
1                2  18 - 34         Somewhat         Somewhat
2                3  18 - 34             Very         Not very
3                4  18 - 34             Very         Not very
4                5  18 - 34             Very             Very
```

Using DataFrames with countplot()

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("masculinity.csv")
sns.countplot(x="how_masculine",
              data=df)
plt.show()
```

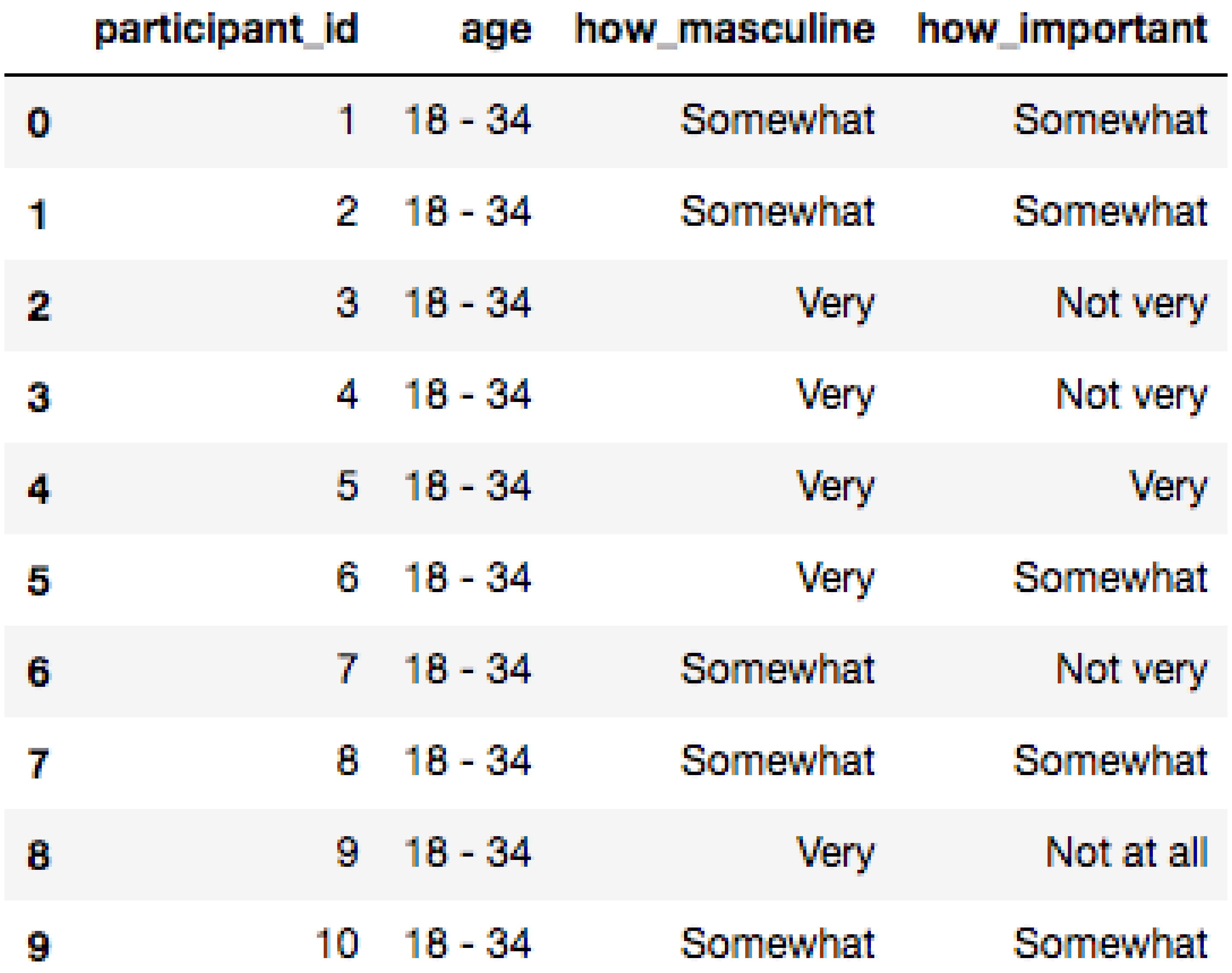


4. Using DataFrames with countplot()

Now let's look at how to make a count plot with a DataFrame instead of a list of data. The first thing we'll do is import Pandas, Matplotlib and Seaborn as we have in past examples. Then, we'll create a Pandas DataFrame called "df" from the masculinity csv file. To create a count plot with a Pandas DataFrame column instead of a list of data, set x equal to the name of the column in the DataFrame - in this case, we'll use the "how_masculine" column. Then, we'll set the data parameter equal to our DataFrame, "df". After calling "plt dot show", we can see that we have a nice count plot of the values in the "how_masculine" column of our data. This plot shows us that the most common response to the question "how masculine or 'manly' do you feel?" is "somewhat", with "very" being the second most common response. Note also that because we're using a named column in the DataFrame, Seaborn automatically adds the name of the column as the x-axis label at the bottom.

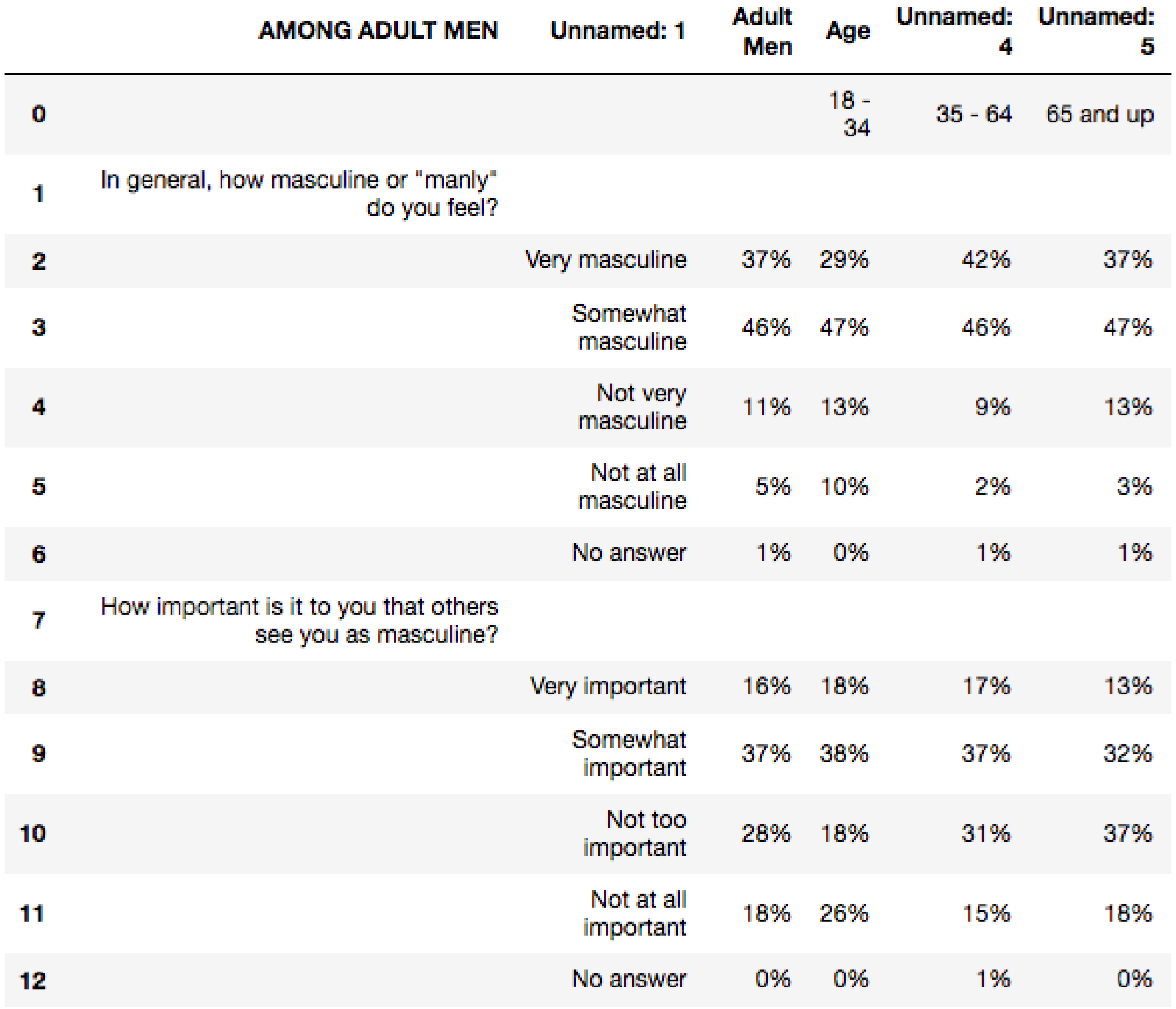
5. "Tidy" data

Let's pause for an important note here. Seaborn works great with Pandas DataFrames, but only if the DataFrame is "tidy". "Tidy data" means that each observation has its own row and each variable has its own column. The "masculinity" DataFrame shown here is tidy because each row is a survey response with one answer to each survey question in each column. Making a count plot with the "how masculine" column works just like passing in a list of that column's values.



	participant_id	age	how_masculine	how_important
0	1	18 - 34	Somewhat	Somewhat
1	2	18 - 34	Somewhat	Somewhat
2	3	18 - 34	Very	Not very
3	4	18 - 34	Very	Not very
4	5	18 - 34	Very	Very
5	6	18 - 34	Very	Somewhat
6	7	18 - 34	Somewhat	Not very
7	8	18 - 34	Somewhat	Somewhat
8	9	18 - 34	Very	Not at all
9	10	18 - 34	Somewhat	Somewhat

6. "Untidy" data
In contrast, here is an example of an "untidy" DataFrame made from the same survey on masculinity. In this untidy DataFrame, notice how each row doesn't contain the same information.



	AMONG ADULT MEN	Unnamed: 1	Adult Men	Age	Unnamed: 4	Unnamed: 5
0				18 - 34	35 - 64	65 and up
1	In general, how masculine or "manly" do you feel?					
2		Very masculine	37%	29%	42%	37%
3		Somewhat masculine	46%	47%	46%	47%
4		Not very masculine	11%	13%	9%	13%
5		Not at all masculine	5%	10%	2%	3%
6		No answer	1%	0%	1%	1%
7	How important is it to you that others see you as masculine?					
8		Very important	16%	18%	17%	13%
9		Somewhat important	37%	38%	37%	32%
10		Not too important	28%	18%	31%	37%
11		Not at all important	18%	26%	15%	18%
12		No answer	0%	0%	1%	0%

Always make sure to check if your DataFrame is tidy before using it with Seaborn.

Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

Adding a third variable with hue

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

In this lesson, we'll see another big advantage that Seaborn offers: the ability to quickly add a third variable to your plots by adding color.

Erin Case

Data Scientist

Tips dataset

```python
import pandas as pd
import seaborn as sns
tips = sns.load_dataset("tips")
tips.head()
```

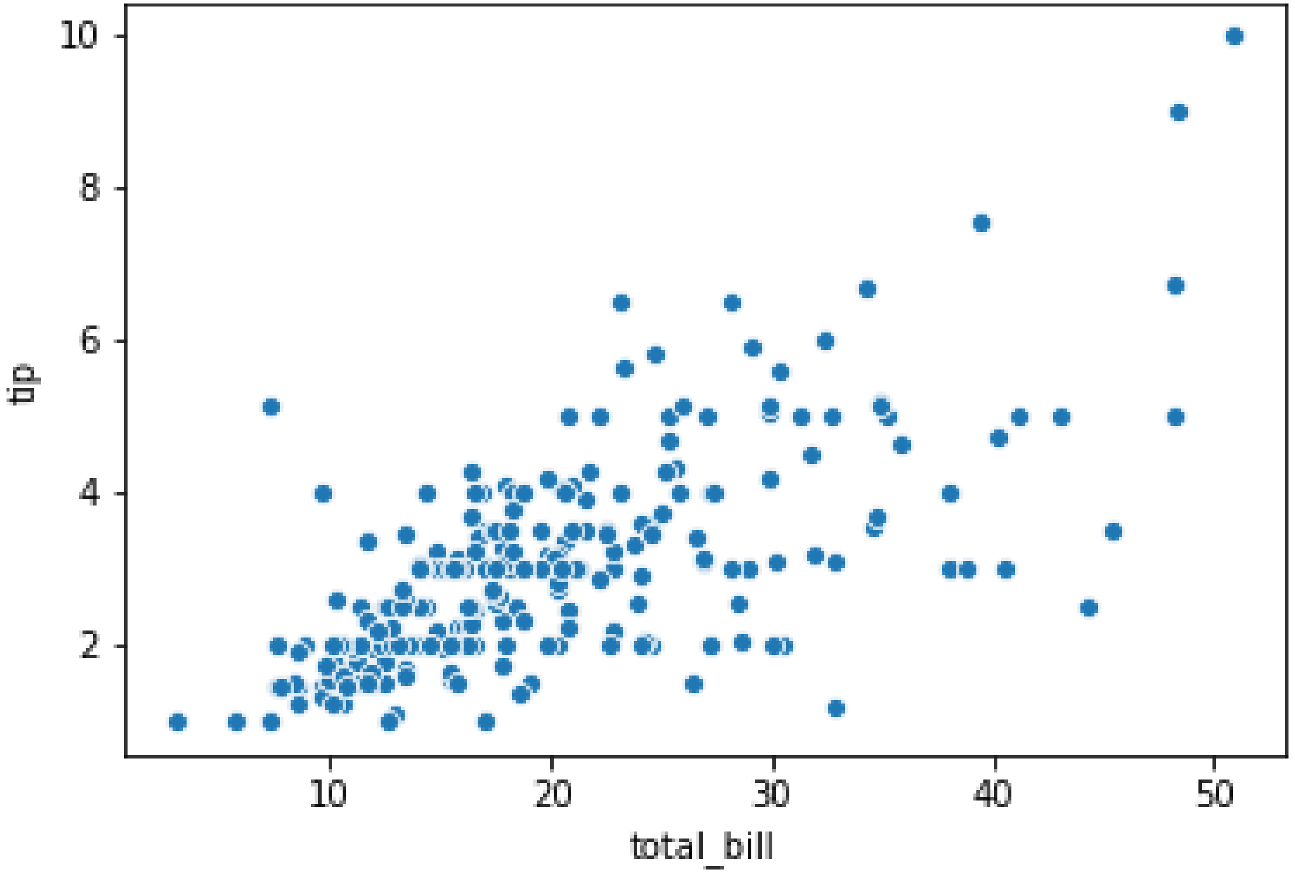
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

2. Tips dataset

To showcase this cool feature in Seaborn, we'll be using Seaborn's built-in tips dataset. You can access it by using the "load dataset" function in Seaborn and passing in the name of the dataset. These are the first five rows of the tips dataset. This dataset contains one row for each table served at a restaurant and has information about things like the bill amount, how many people were at the table, and when the table was served. Let's explore the relationship between the "total_bill" and "tip" columns using a scatter plot.

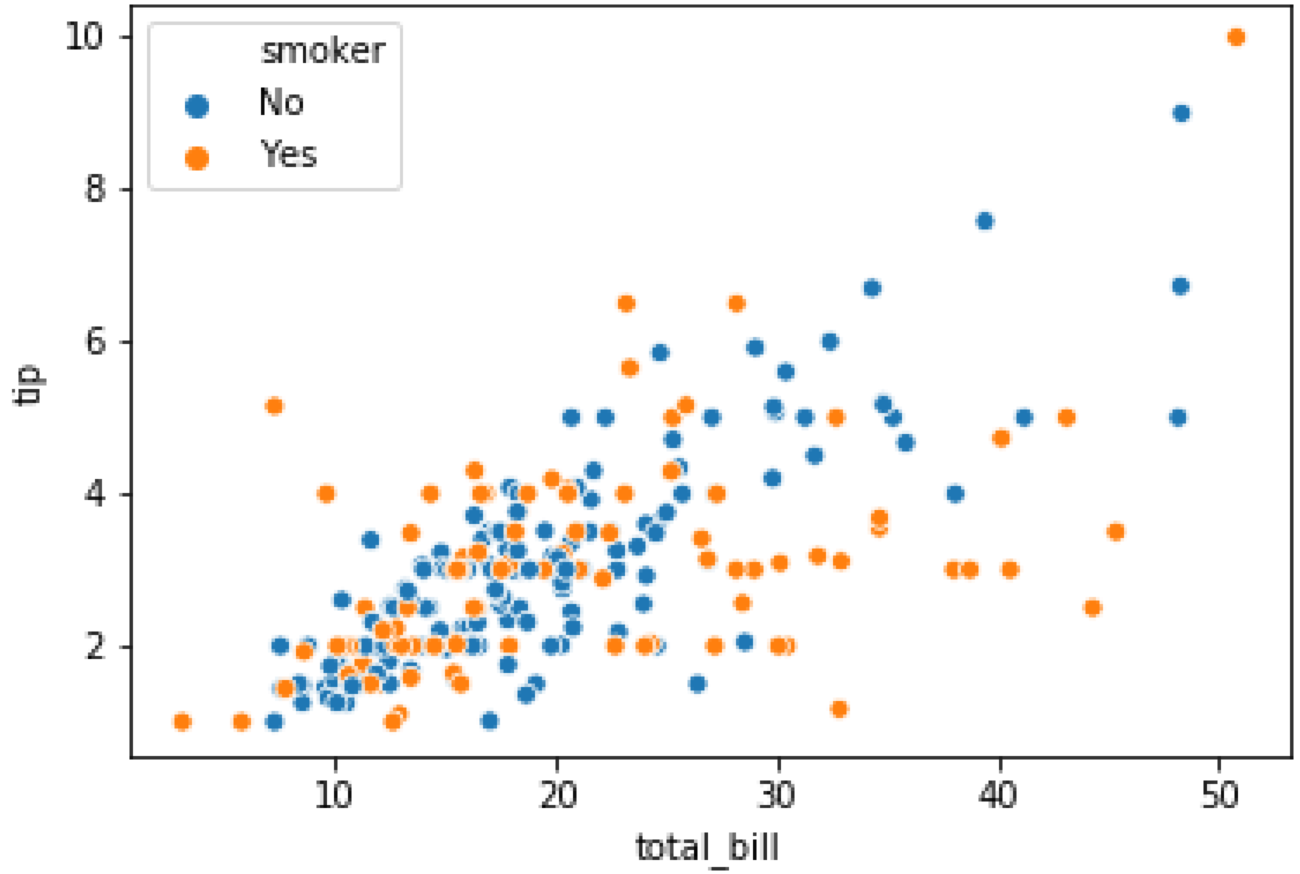
A basic scatter plot

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.scatterplot(x="total_bill",
                y="tip",
                data=tips)


plt.show()
```



A scatter plot with hue

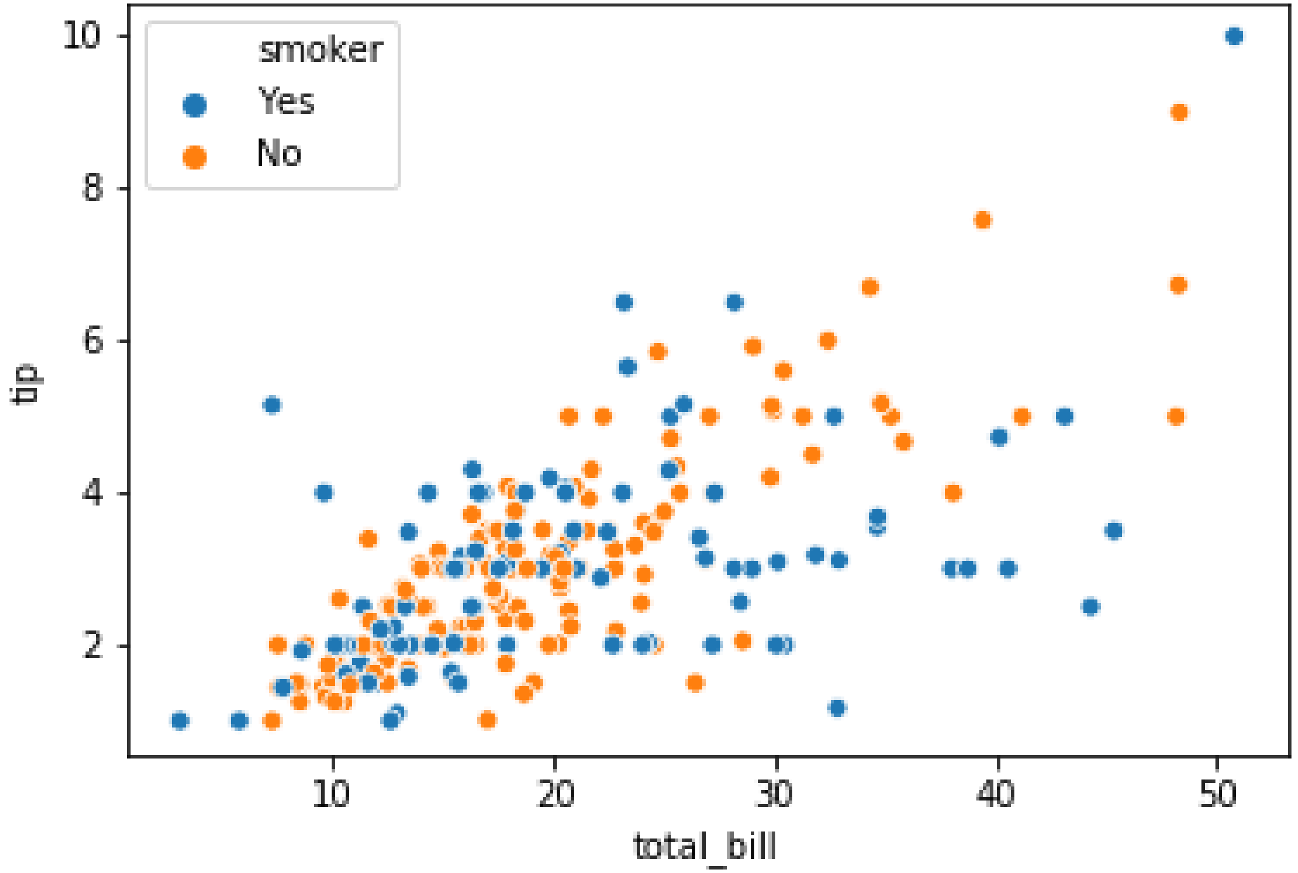
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.scatterplot(x="total_bill",
                y="tip",
                data=tips,
                hue="smoker")
plt.show()
```



4. A scatter plot with hue
You can set the "hue" parameter equal to the DataFrame column name "smoker" and then Seaborn will automatically color each point by whether they are a smoker. Plus, it will add a legend to the plot automatically! If you don't want to use Pandas, you can set it equal to a list of values instead of a column name.

Setting hue order

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.scatterplot(x="total_bill",
                y="tip",
                data=tips,
                hue="smoker",
                hue_order=["Yes",
                           "No"])
plt.show()
```

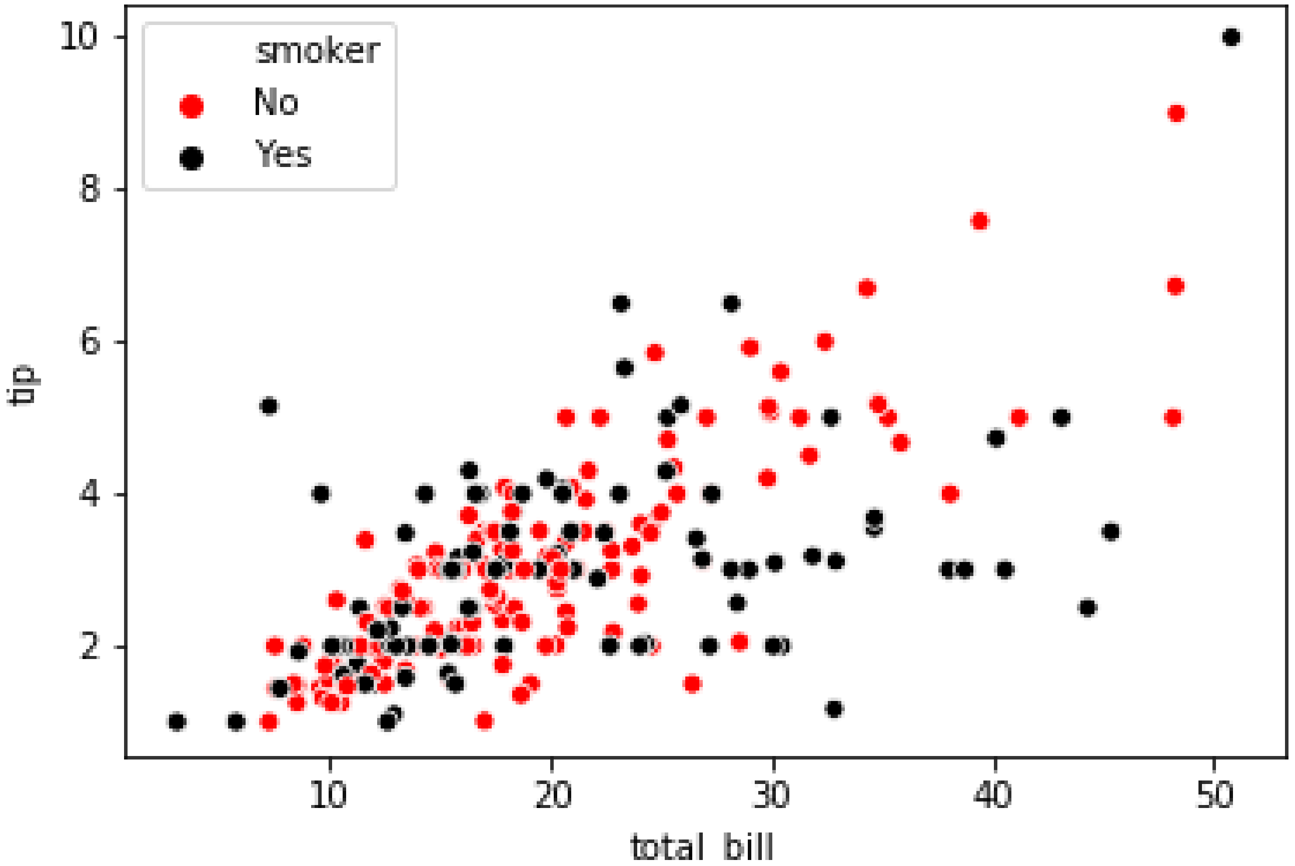


5. Setting hue order

Hue also allows you to assert more control over the ordering and coloring of each value. The "hue order" parameter takes in a list of values and will set the order of the values in the plot accordingly. Notice how the legend for smoker now lists "yes" before "no".

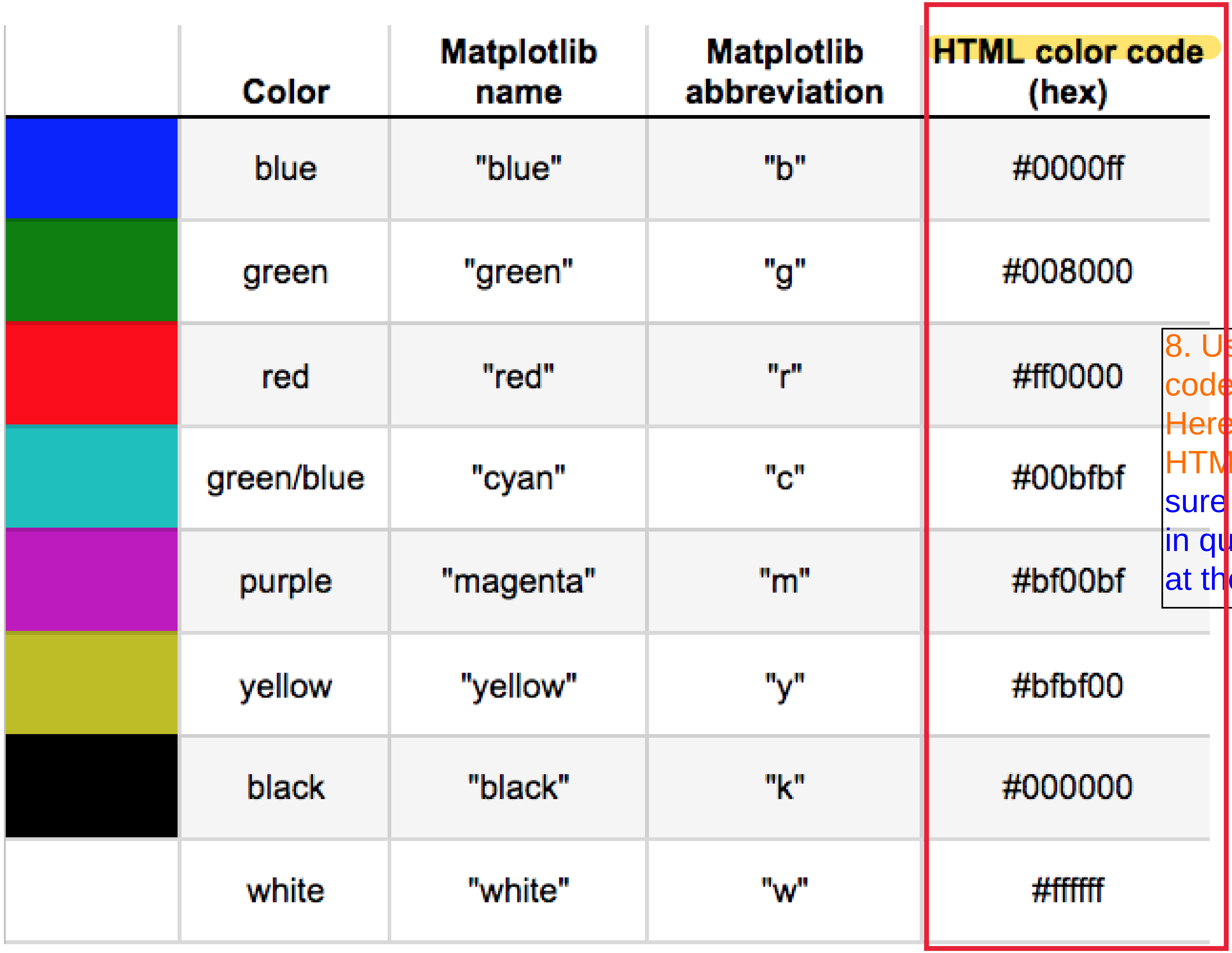
Specifying hue colors

```
import matplotlib.pyplot as plt
import seaborn as sns
hue_colors = {"Yes": "black",
              "No": "red"}
sns.scatterplot(x="total_bill",
                y="tip",
                data=tips,
                hue="smoker",
                palette=hue_colors)


plt.show()
```



6. Specifying hue colors

You can also control the colors assigned to each value using the "palette" parameter. This parameter takes in a dictionary, which is a data structure that has key-value pairs. This dictionary should map the variable values to the colors you want to represent the value. Here, we create a dictionary called "hue colors" that maps the value "Yes" to the color black and the value "No" to the color red. When we set hue equal to "smoker" and the palette parameter equal to this dictionary, we have a scatter plot where smokers are represented with black dots and non-smokers are represented with red dots.

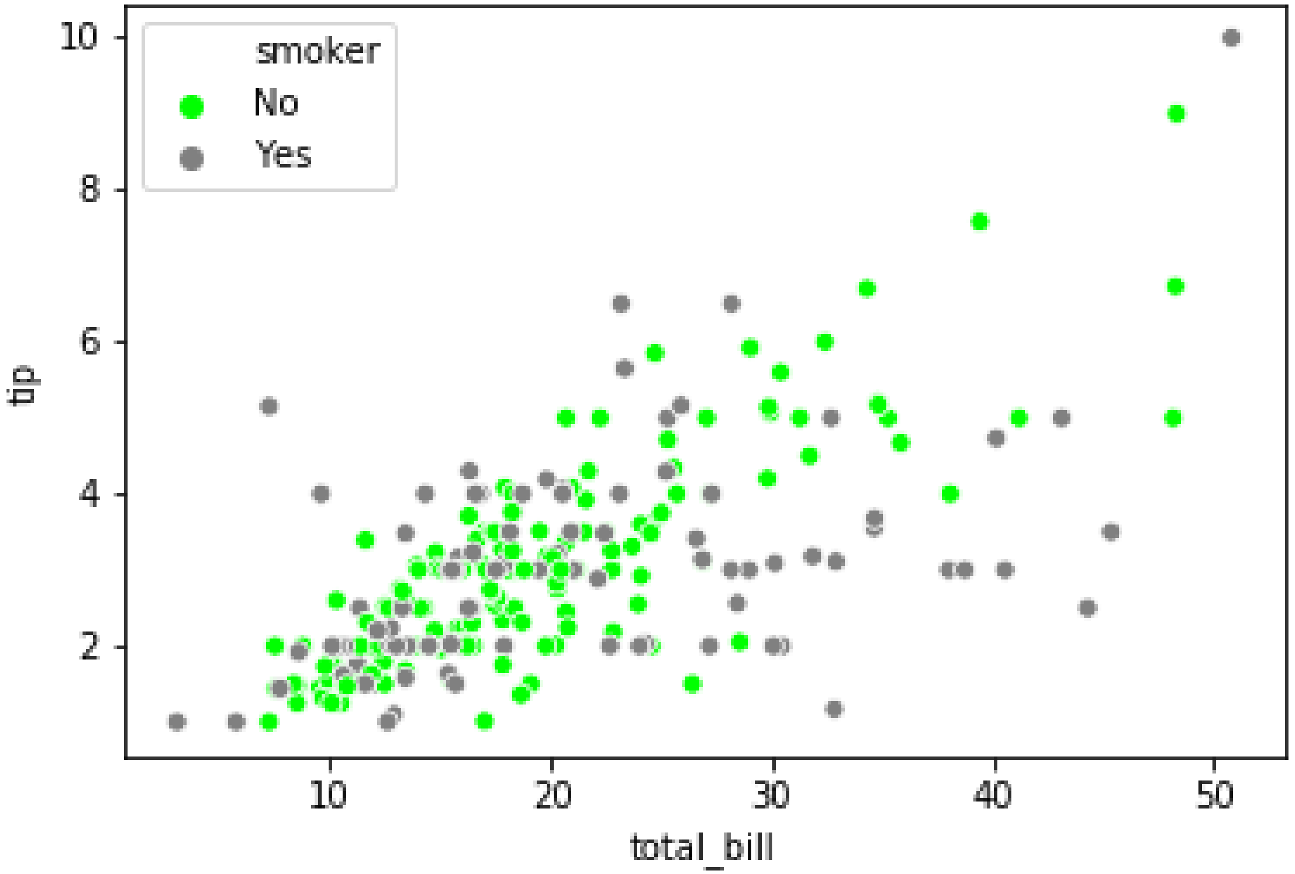


	Color	Matplotlib name	Matplotlib abbreviation	HTML color code (hex)
🟦	blue	"blue"	"b"	#0000ff
🟩	green	"green"	"g"	#008000
🟥	red	"red"	"r"	#ff0000
🟦	green/blue	"cyan"	"c"	#00bfbf
🟪	purple	"magenta"	"m"	#bf00bf
🟨	yellow	"yellow"	"y"	#bfbf00
⬛	black	"black"	"k"	#000000
⬜	white	"white"	"w"	#ffffff

8. Using HTML hex color codes with hue
Here's an example using HTML hex codes. Make sure you put the hex codes in quotes with a pound sign at the beginning.

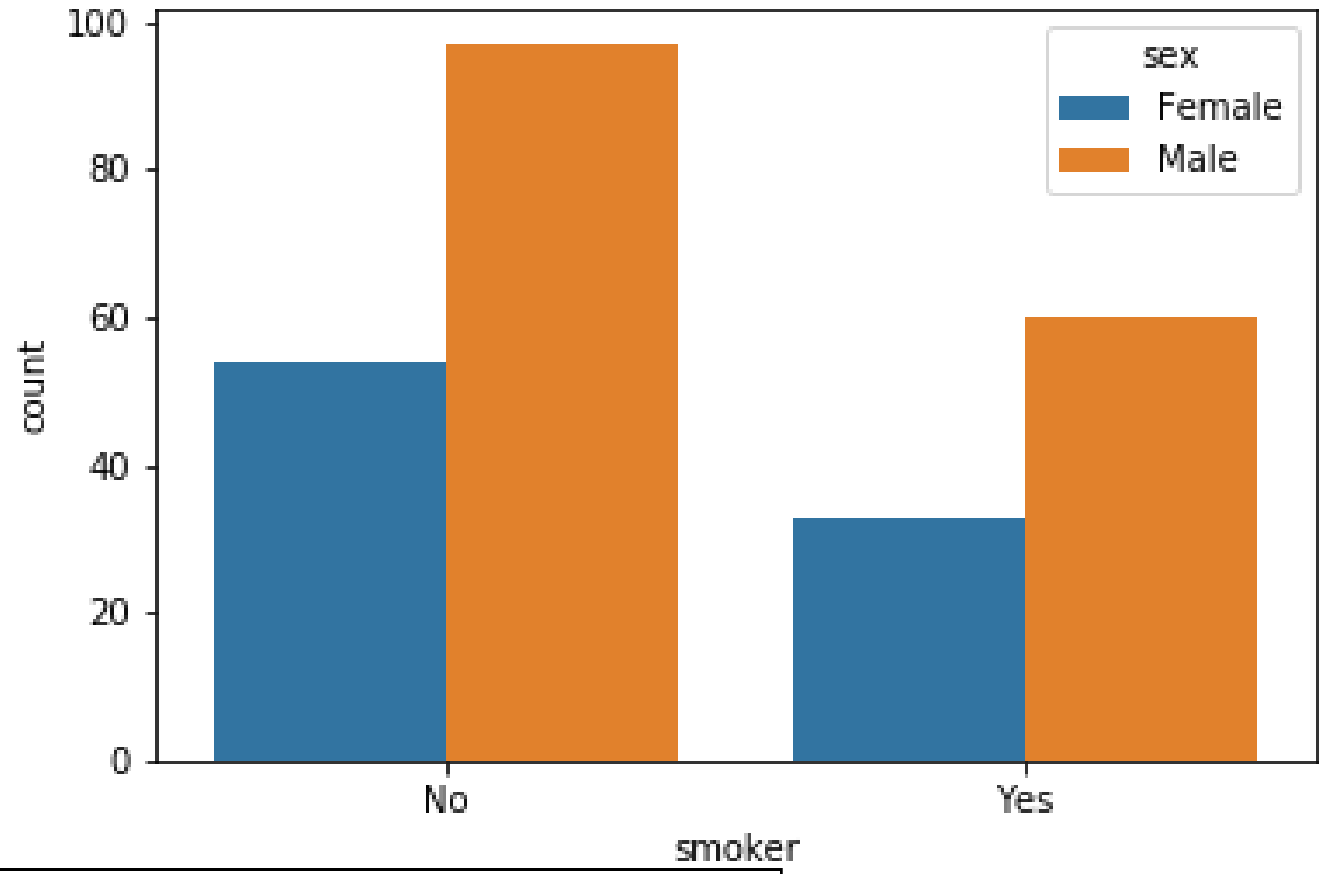
Using HTML hex color codes with hue

```python
import matplotlib.pyplot as plt
import seaborn as sns

hue_colors = {"Yes": "#808080",
              "No": "#00FF00"}

sns.scatterplot(x="total_bill",
                y="tip",
                data=tips,
                hue="smoker",
                palette=hue_colors)
plt.show()
```



Using hue with count plots

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x="smoker",
              data=tips,
              hue="sex")
plt.show()
```



9. Using hue with count plots
As a final note, hue is available in most of Seaborn's plot types. For example, this count plot shows the number of observations we have for smokers versus non-smokers, and setting "hue" equal to "sex" divides these bars into subgroups of males versus females. From this plot, we can see that males outnumber females among both smokers and non-smokers in this dataset.

Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN