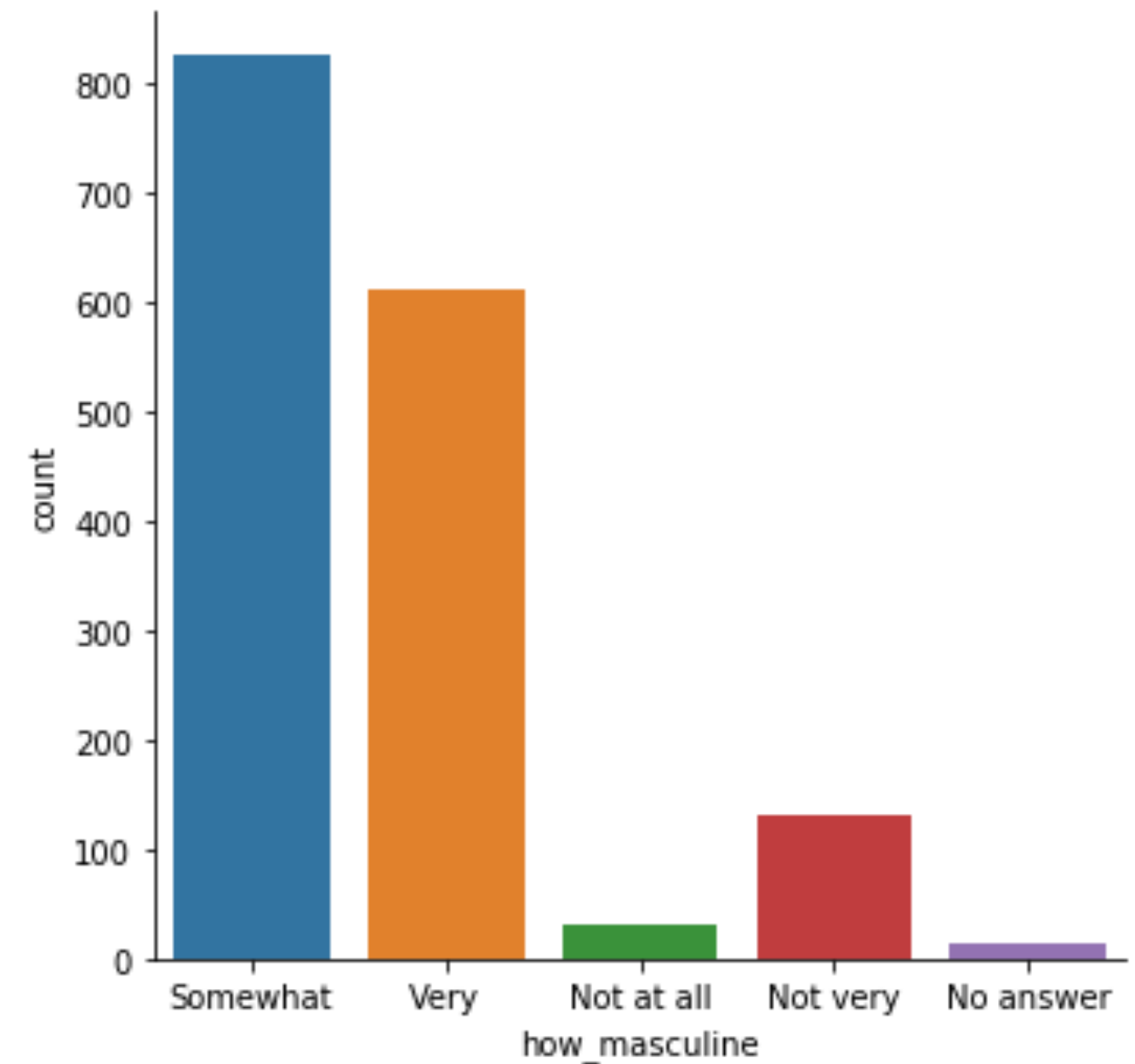


# Categorical plots

- Examples: count plots, bar plots
- Involve a categorical variable
- Comparisons between groups



# catplot()

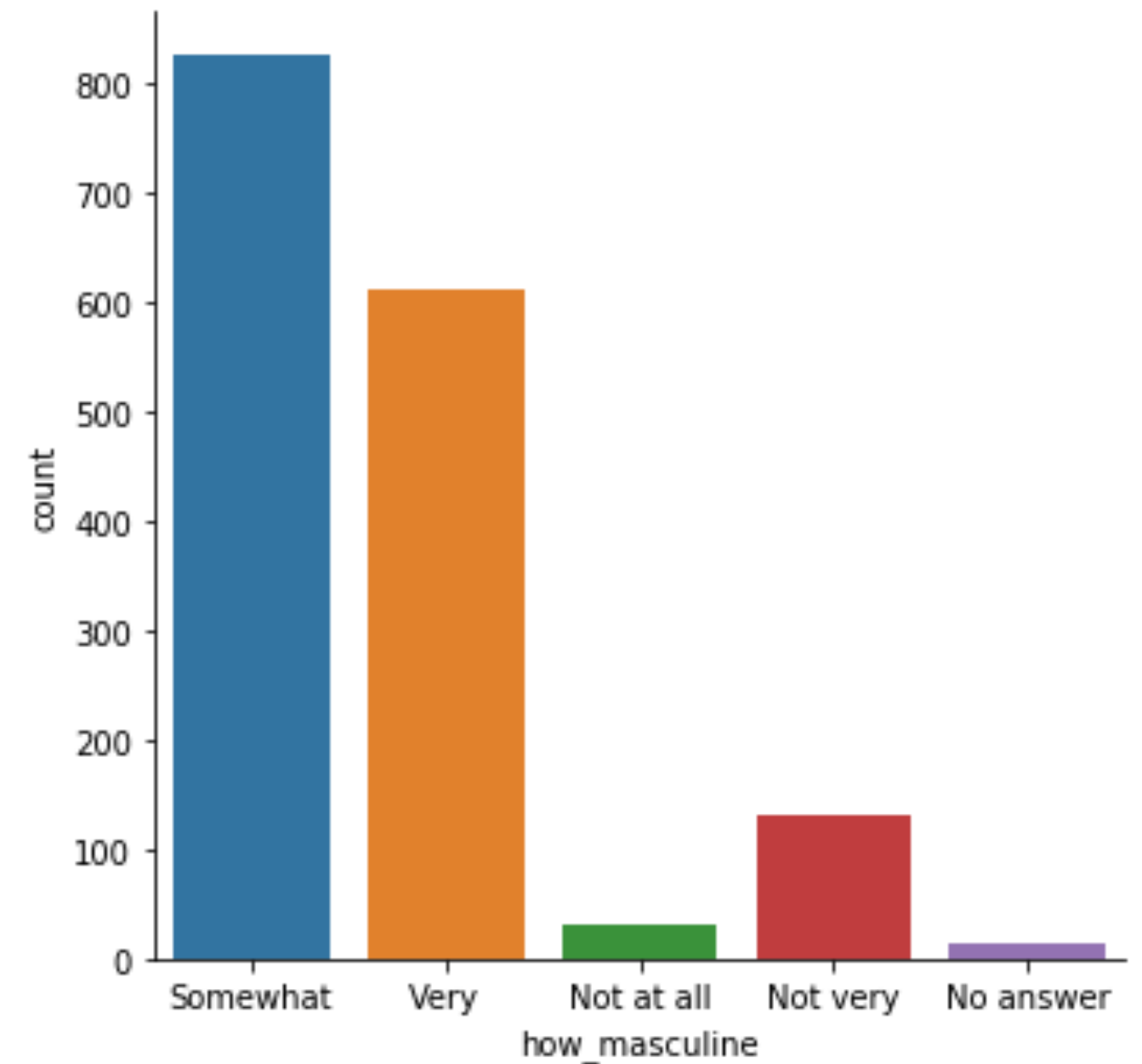
- Used to create categorical plots
- Same advantages of `relplot()`
- Easily create subplots with `col=` and `row=`

# countplot() vs. catplot()

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

sns.countplot(x="how_masculine",
              data=masculinity_data)

plt.show()
```

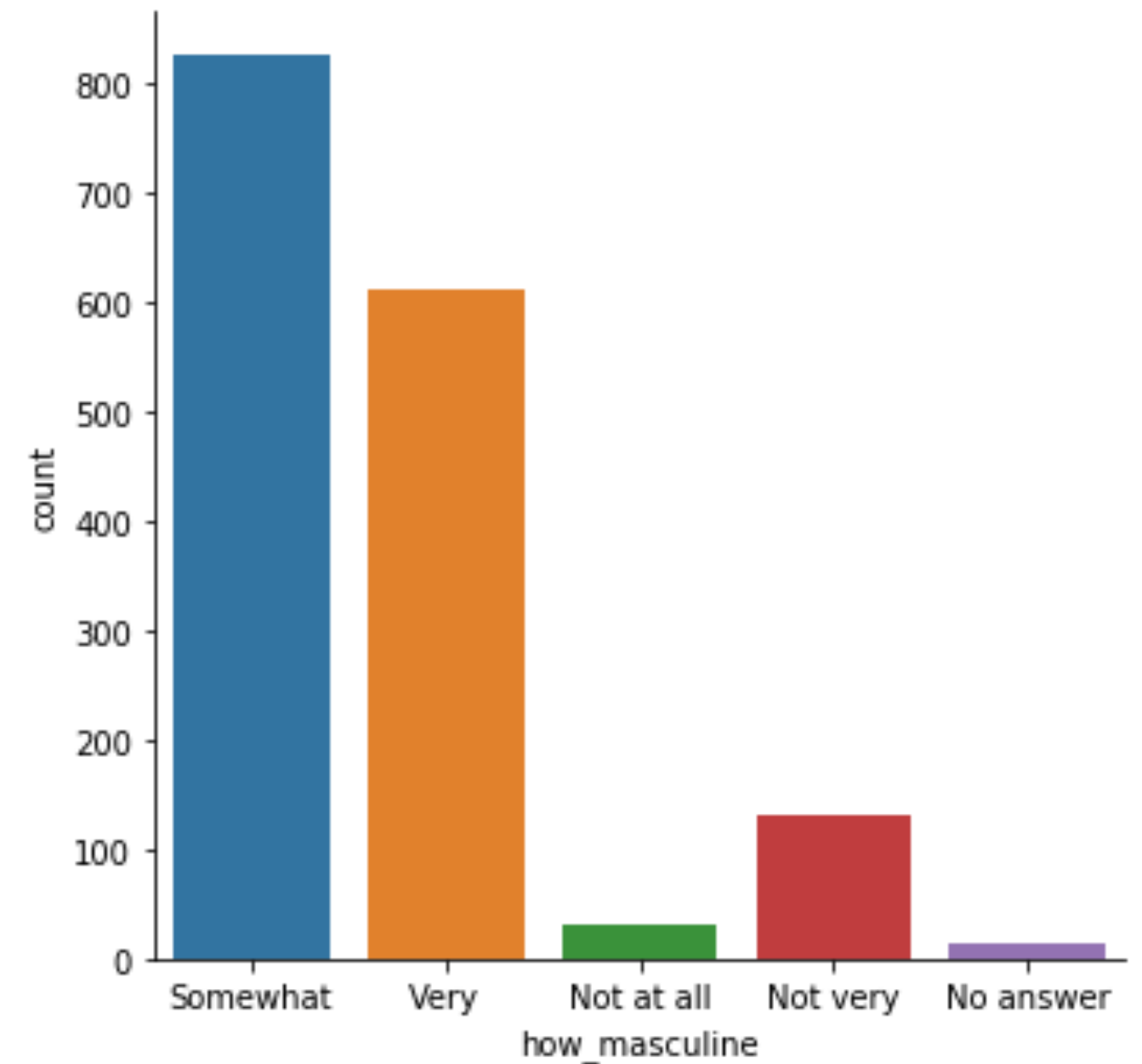


# countplot() vs. catplot()

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

sns.catplot(x="how_masculine",
            data=masculinity_data,
            kind="count")

plt.show()
```

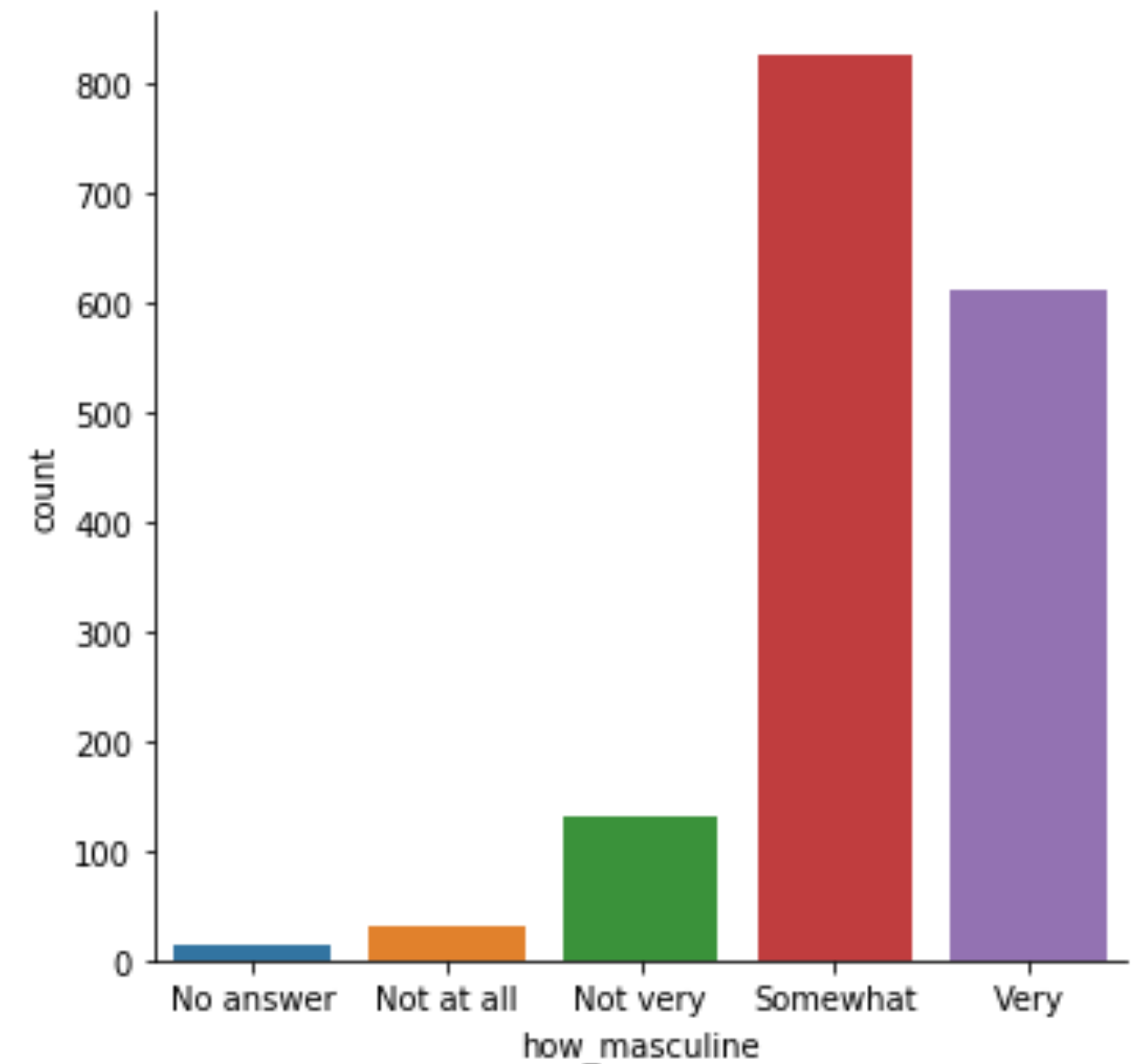


# Changing the order

```
import matplotlib.pyplot as plt
import seaborn as sns
category_order = ["No answer",
                  "Not at all",
                  "Not very",
                  "Somewhat",
                  "Very"]

sns.catplot(x="how_masculine",
            data=masculinity_data,
            kind="count",
            order=category_order)

plt.show()
```



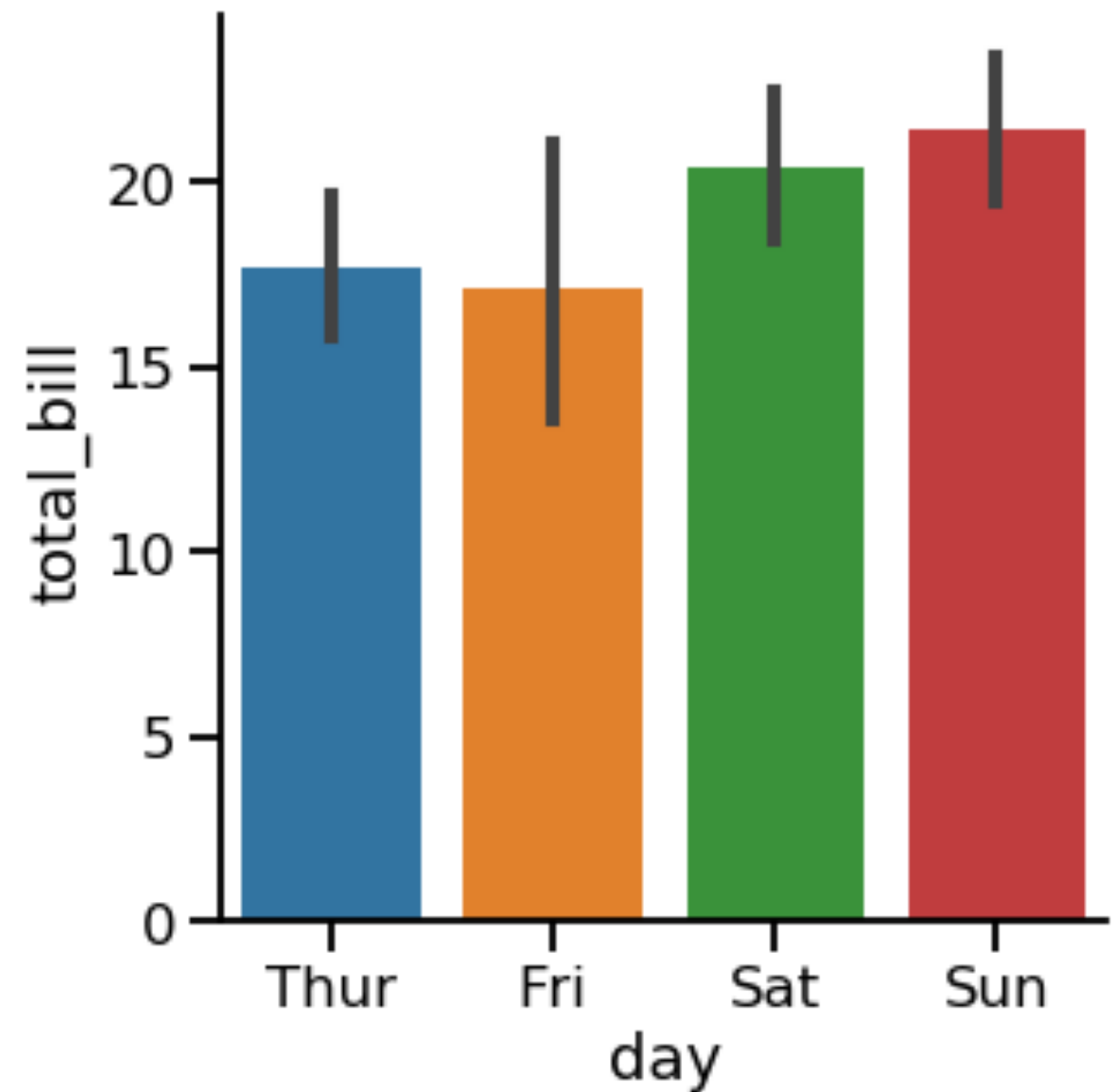
# Bar plots

Displays mean of quantitative variable per category

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

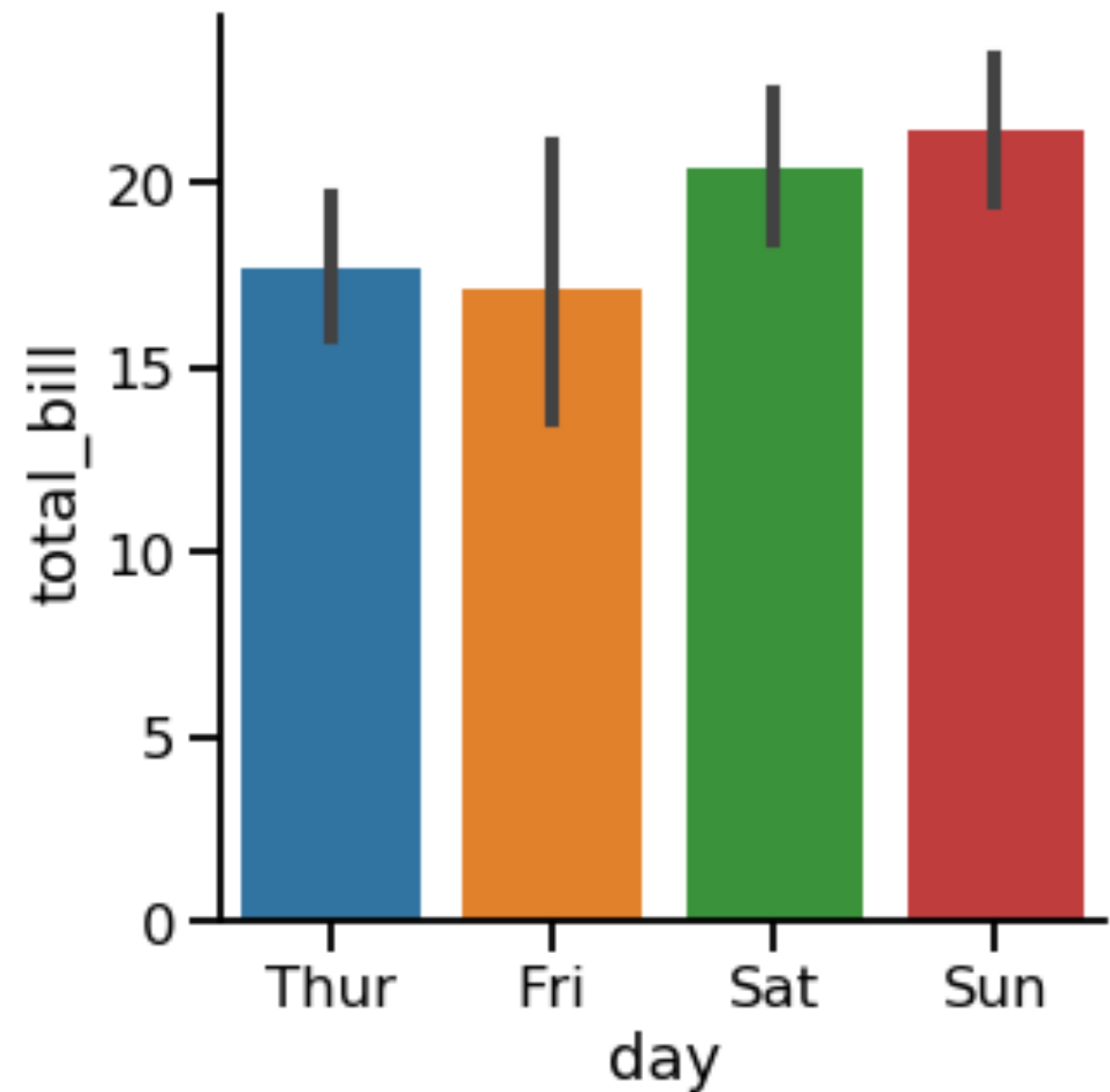
sns.catplot(x="day",
            y="total_bill",
            data=tips,
            kind="bar")

plt.show()
```



# Confidence intervals

- Lines show 95% confidence intervals for the mean
- Shows uncertainty about our estimate
- Assumes our data is a random sample

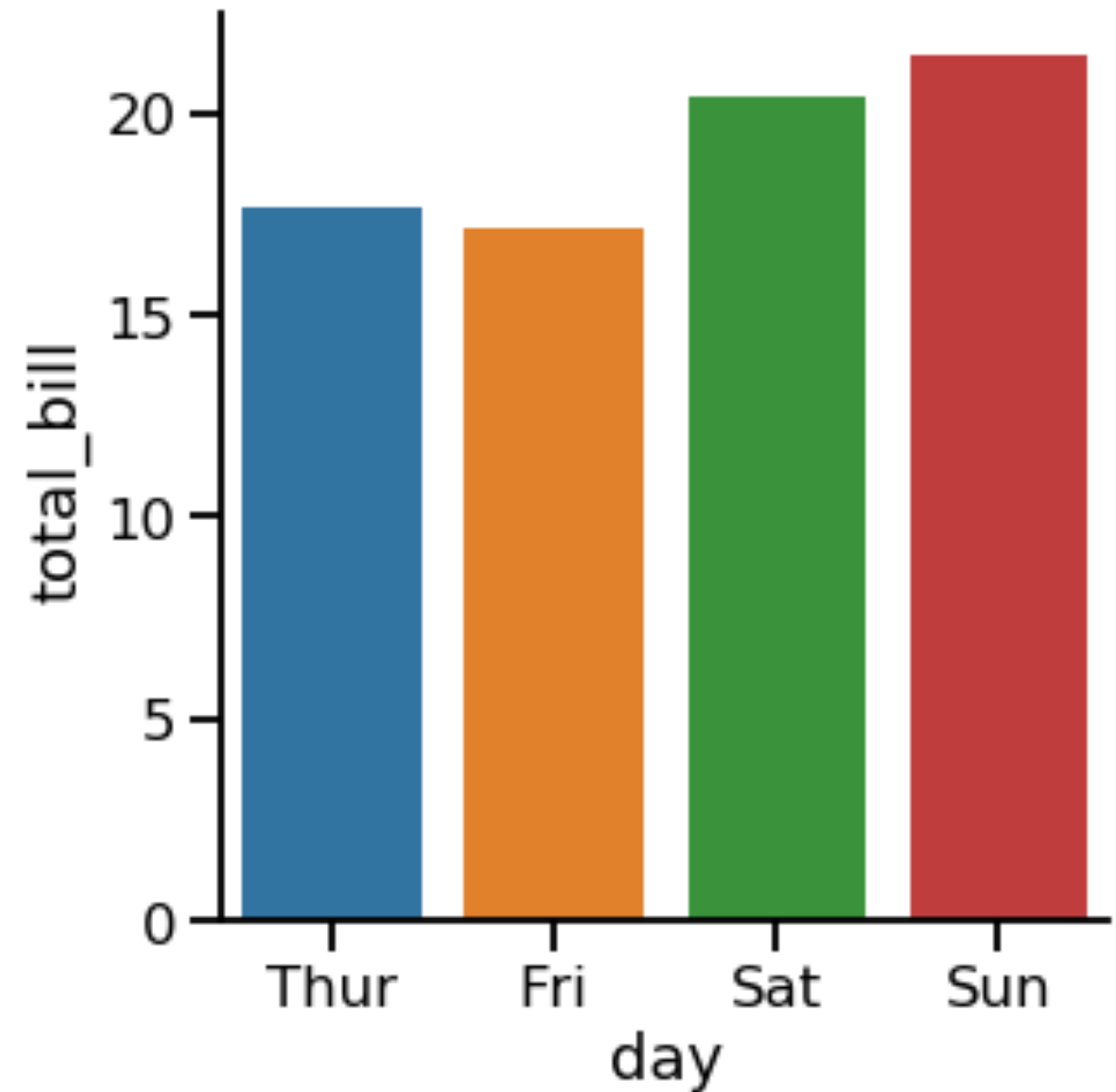


# Turning off confidence intervals

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

sns.catplot(x="day",
            y="total_bill",
            data=tips,
            kind="bar",
            ci=None)

plt.show()
```



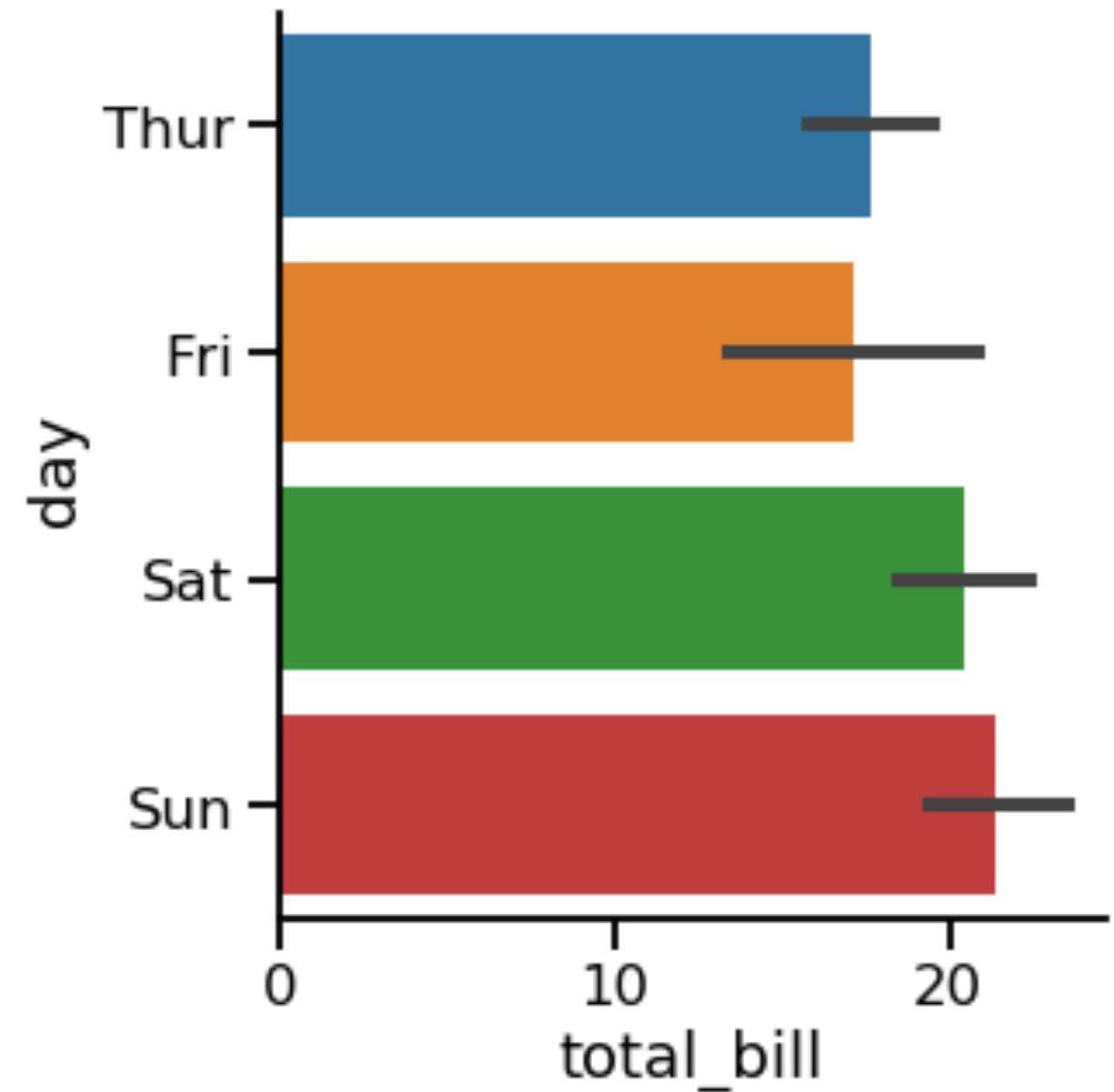


# Changing the orientation

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

sns.catplot(x="total_bill",
            y="day",
            data=tips,
            kind="bar")

plt.show()
```

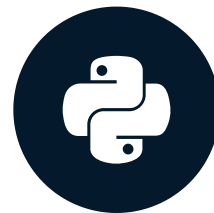


# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

# Creating a box plot

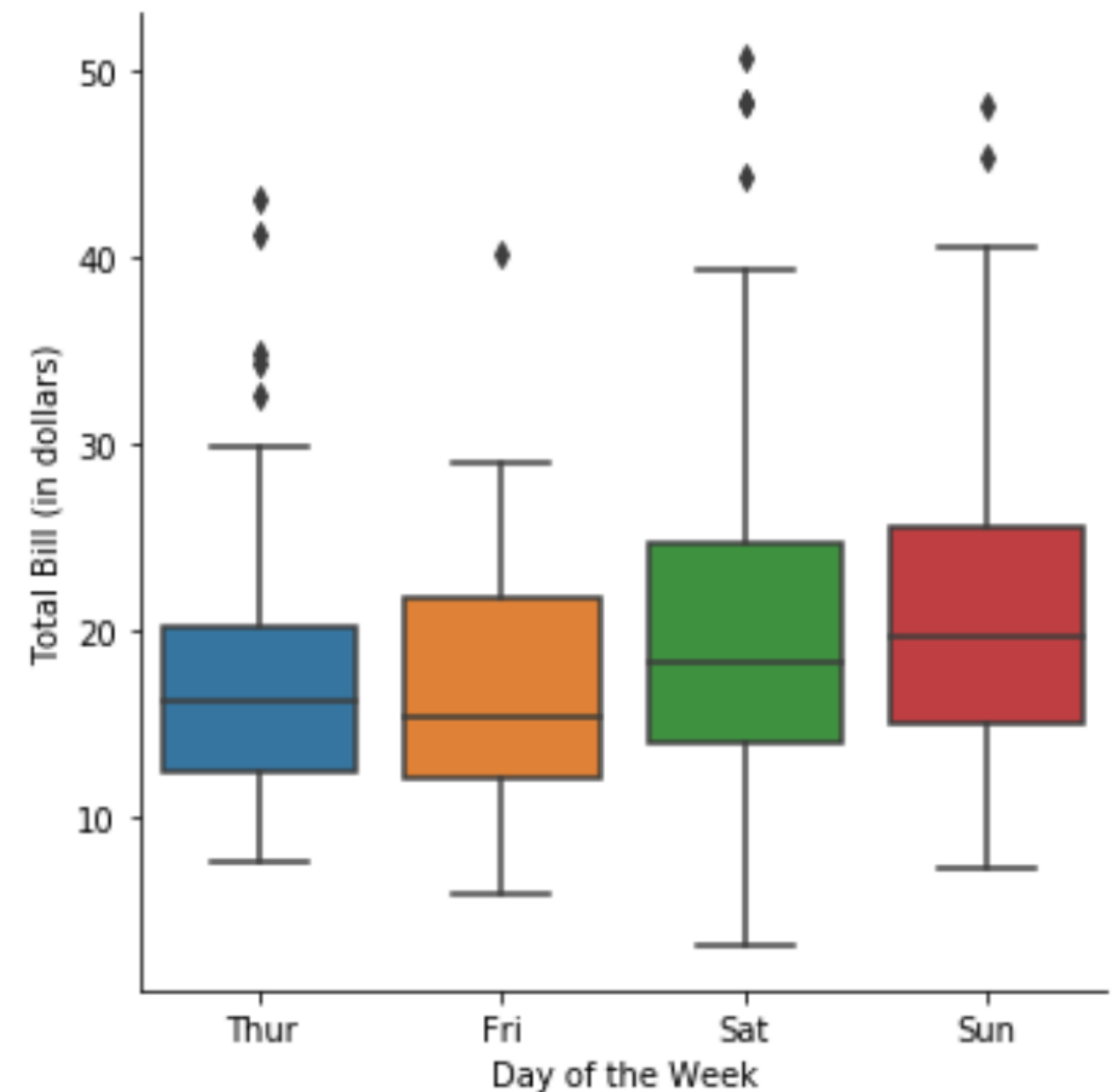
INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



**Erin Case**  
Data Scientist

# What is a box plot?

- Shows the distribution of quantitative data
- See median, spread, skewness, and outliers
- Facilitates comparisons between groups

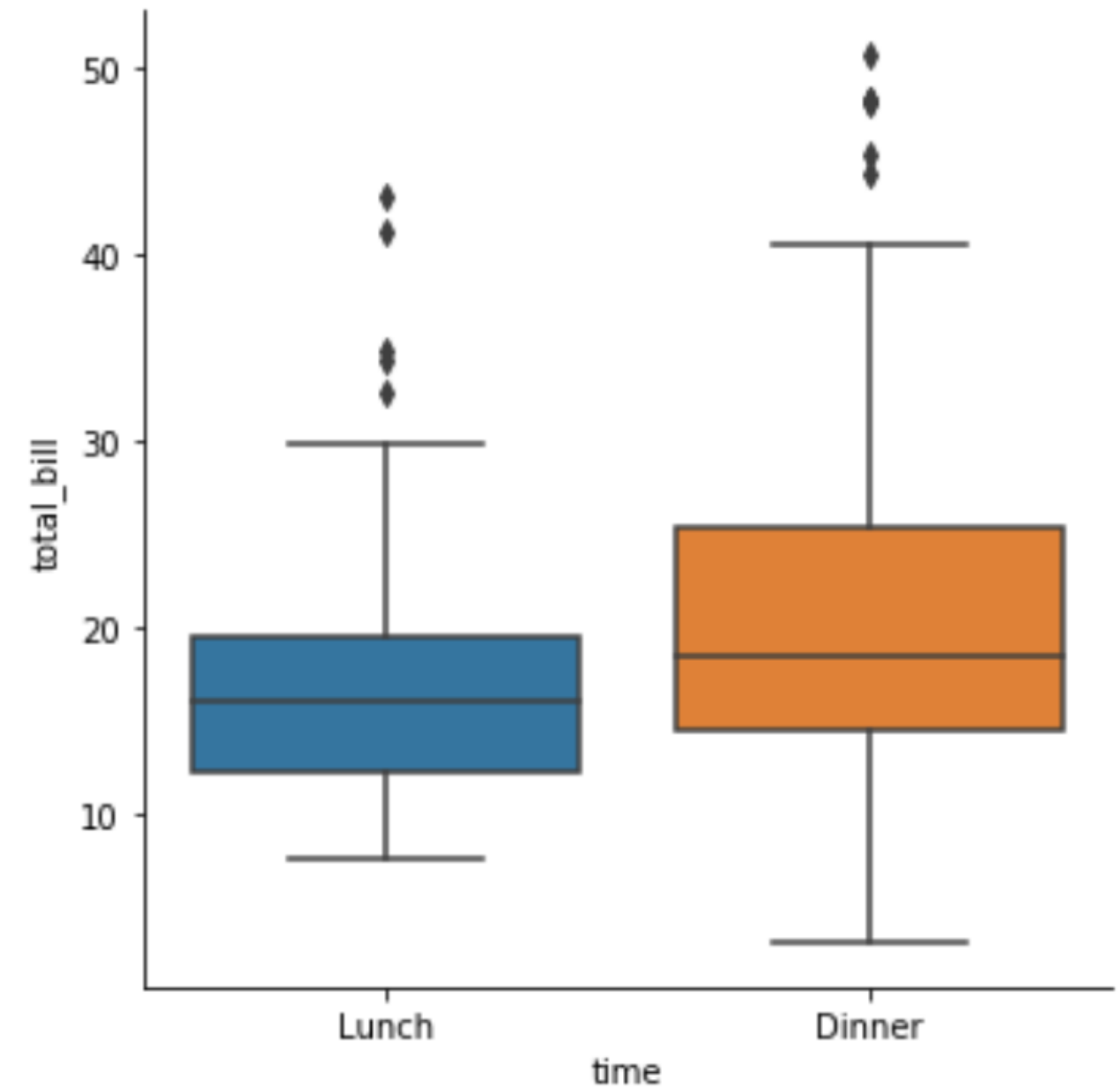


# How to create a box plot

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

g = sns.catplot(x="time",
                y="total_bill",
                data=tips,
                kind="box")

plt.show()
```

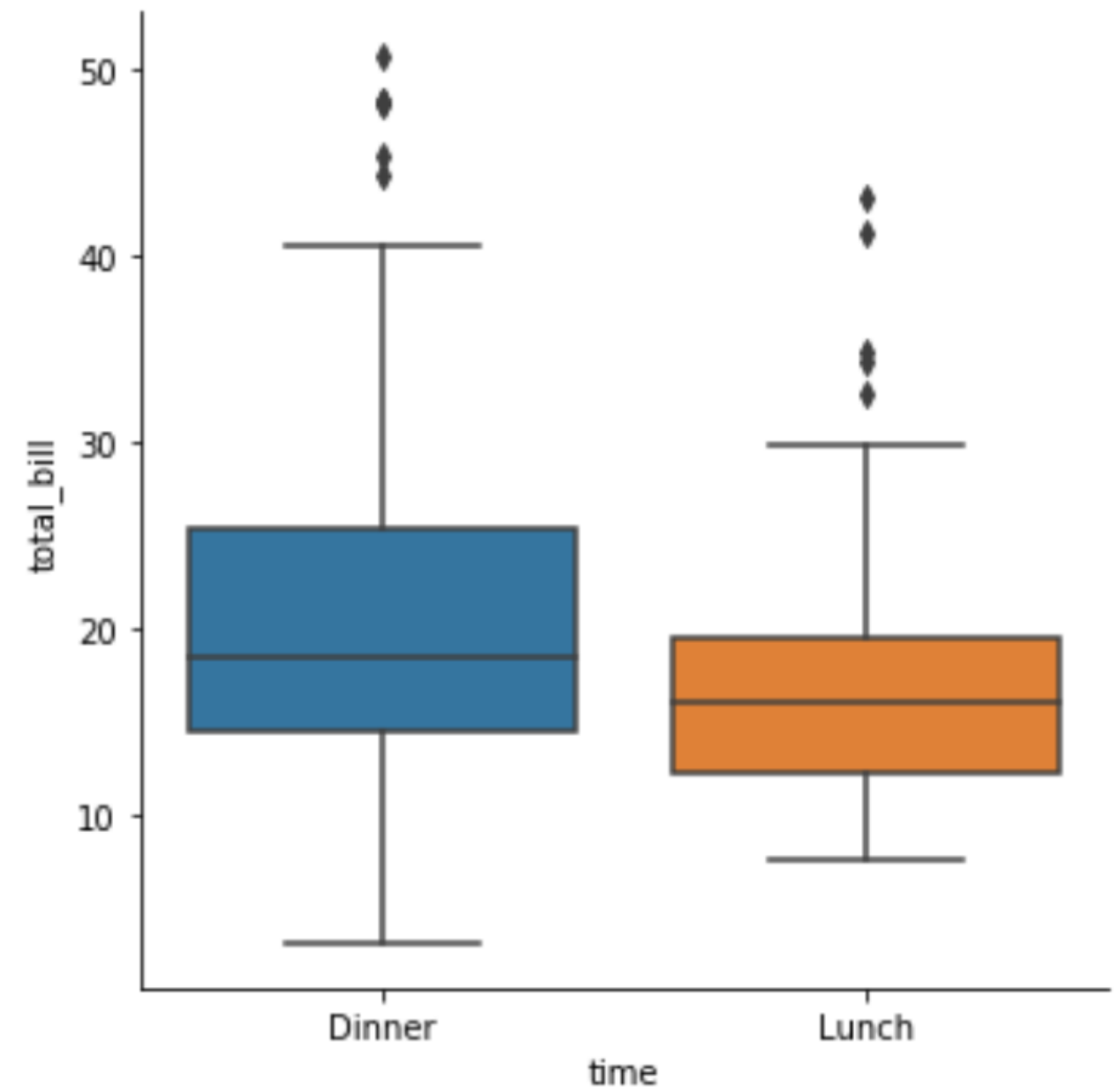


# Change the order of categories

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

g = sns.catplot(x="time",
                y="total_bill",
                data=tips,
                kind="box",
                order=["Dinner",
                     "Lunch"])

plt.show()
```

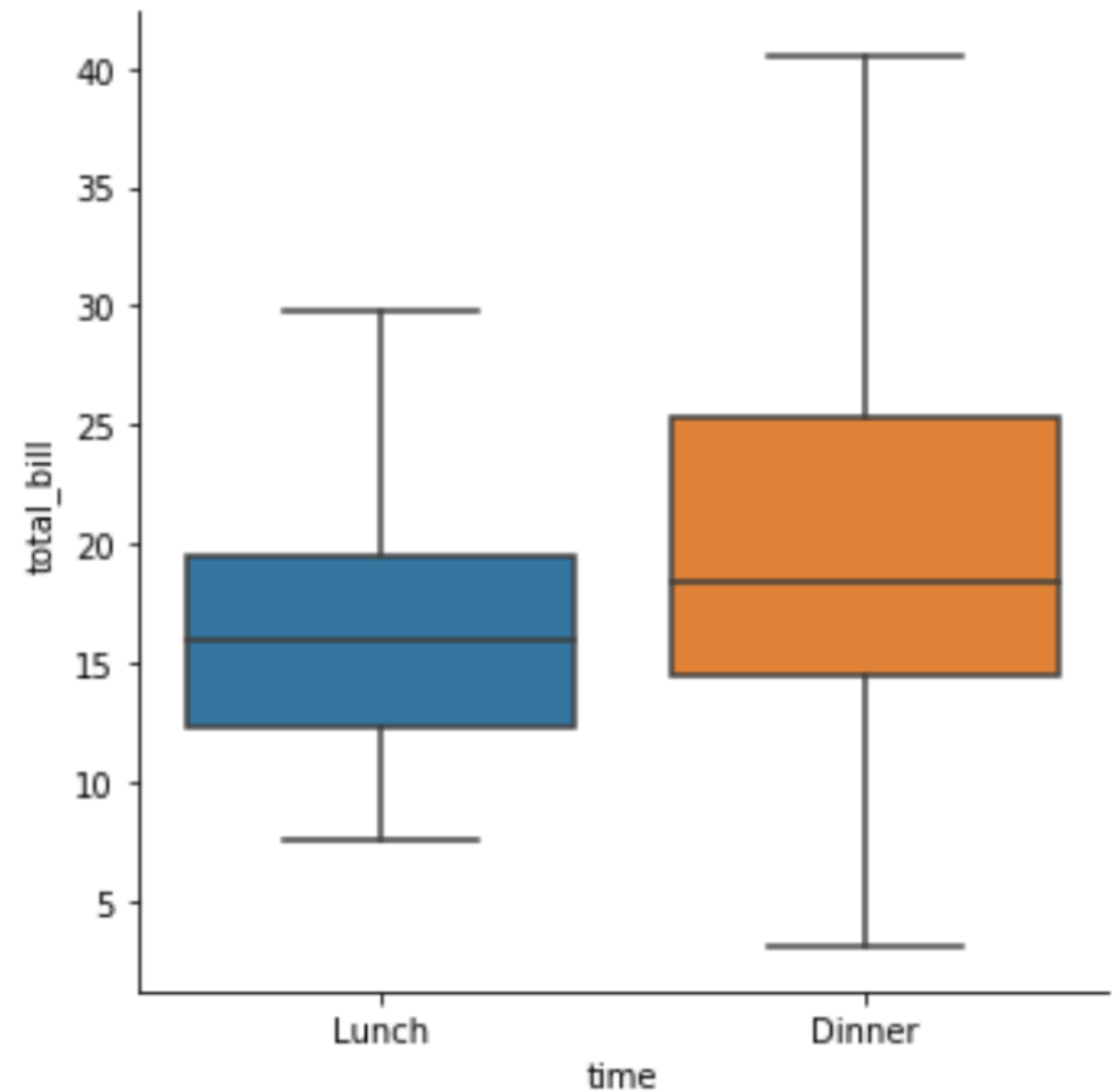


# Omitting the outliers using `sym`

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

g = sns.catplot(x="time",
                y="total_bill",
                data=tips,
                kind="box",
                sym="")

plt.show()
```



# Changing the whiskers using `whis`

- By default, the whiskers extend to  $1.5 \times$  the interquartile range
- Make them extend to  $2.0 \times$  IQR: `whis=2.0`
- Show the 5th and 95th percentiles: `whis=[5, 95]`
- Show min and max values: `whis=[0, 100]`

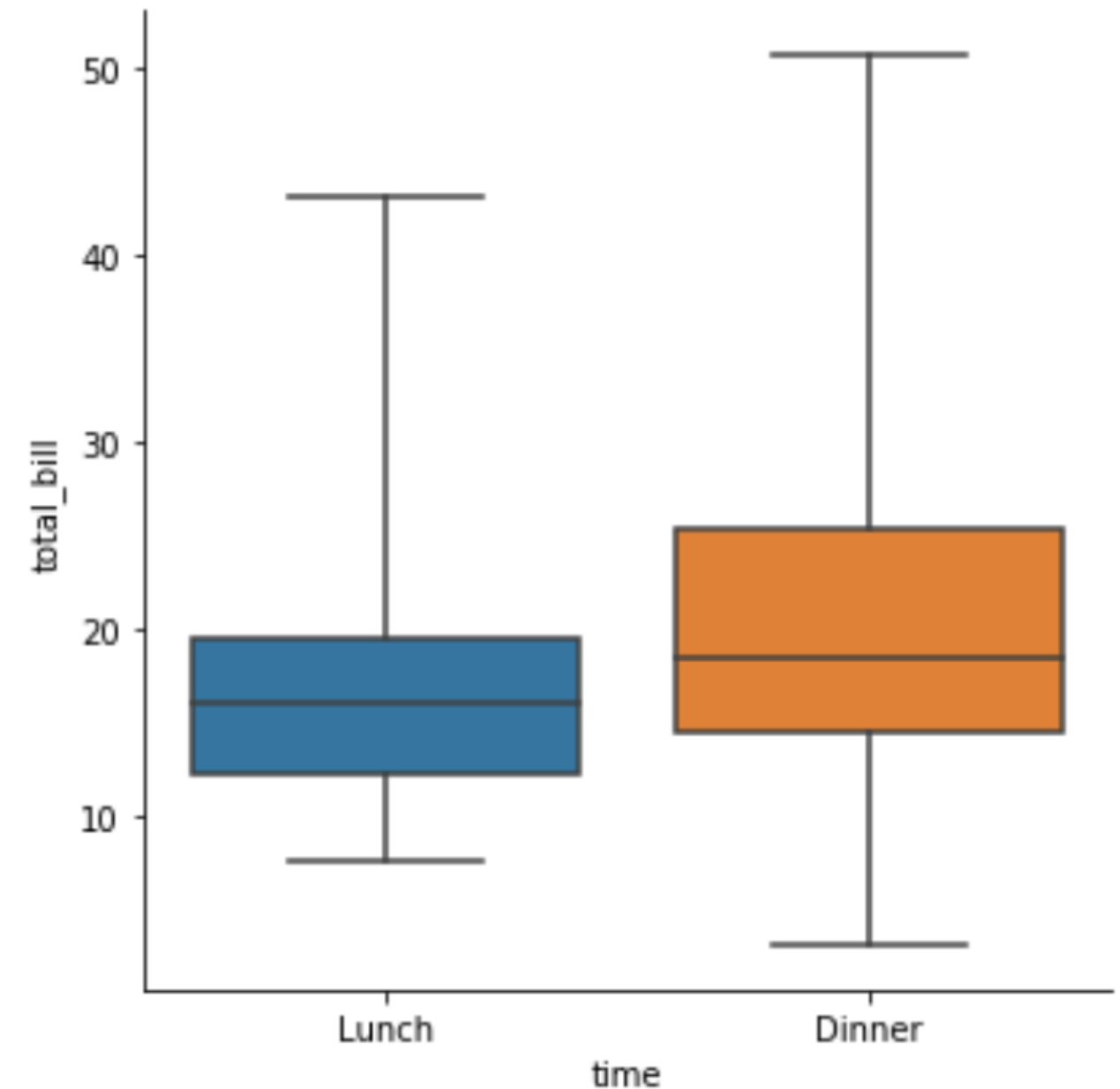


# Changing the whiskers using `whis`

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

g = sns.catplot(x="time",
                y="total_bill",
                data=tips,
                kind="box",
                whis=[0, 100])

plt.show()
```

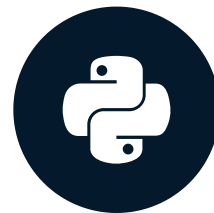


# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

# Point plots

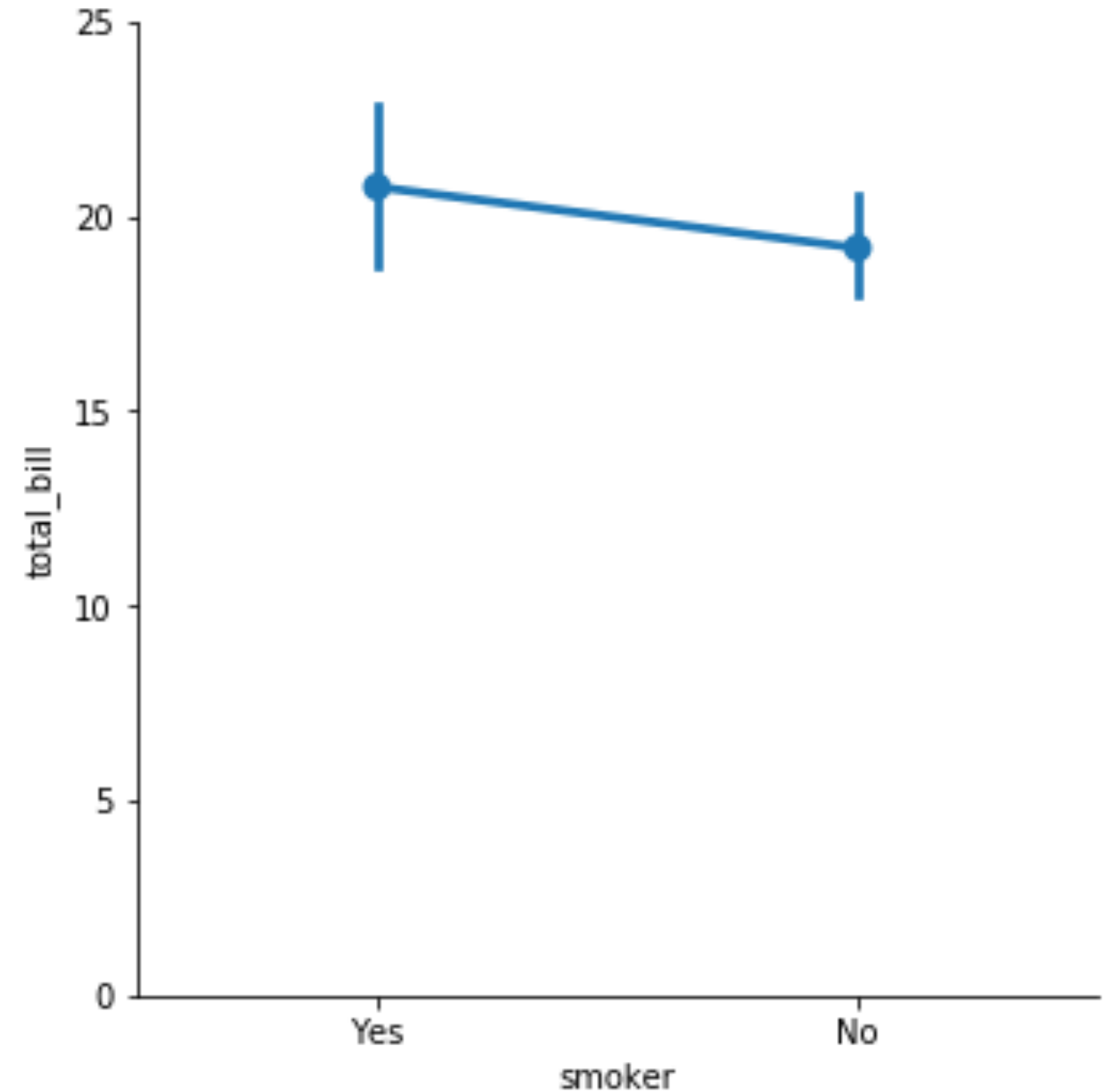
INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



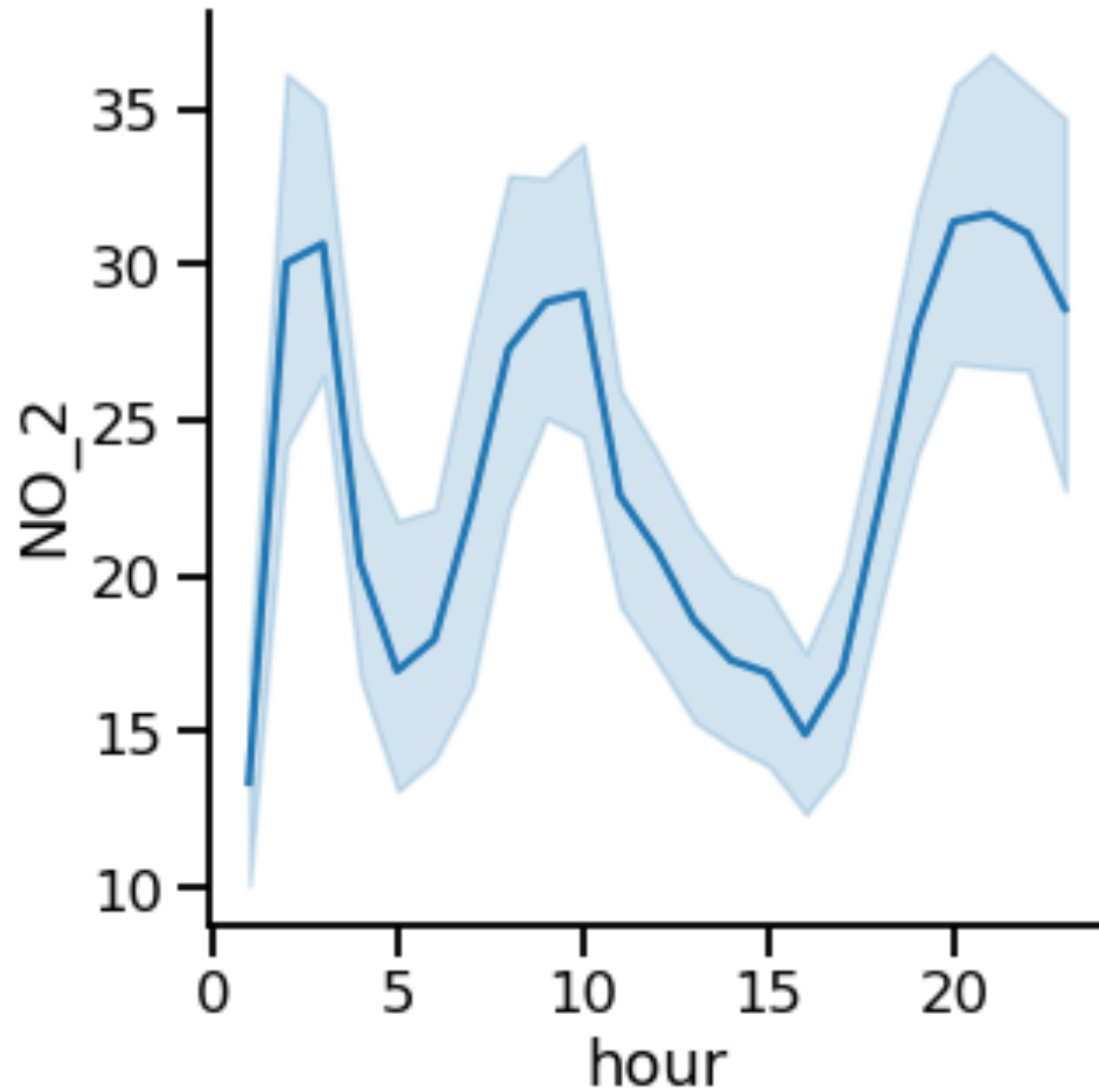
**Erin Case**  
Data Scientist

# What are point plots?

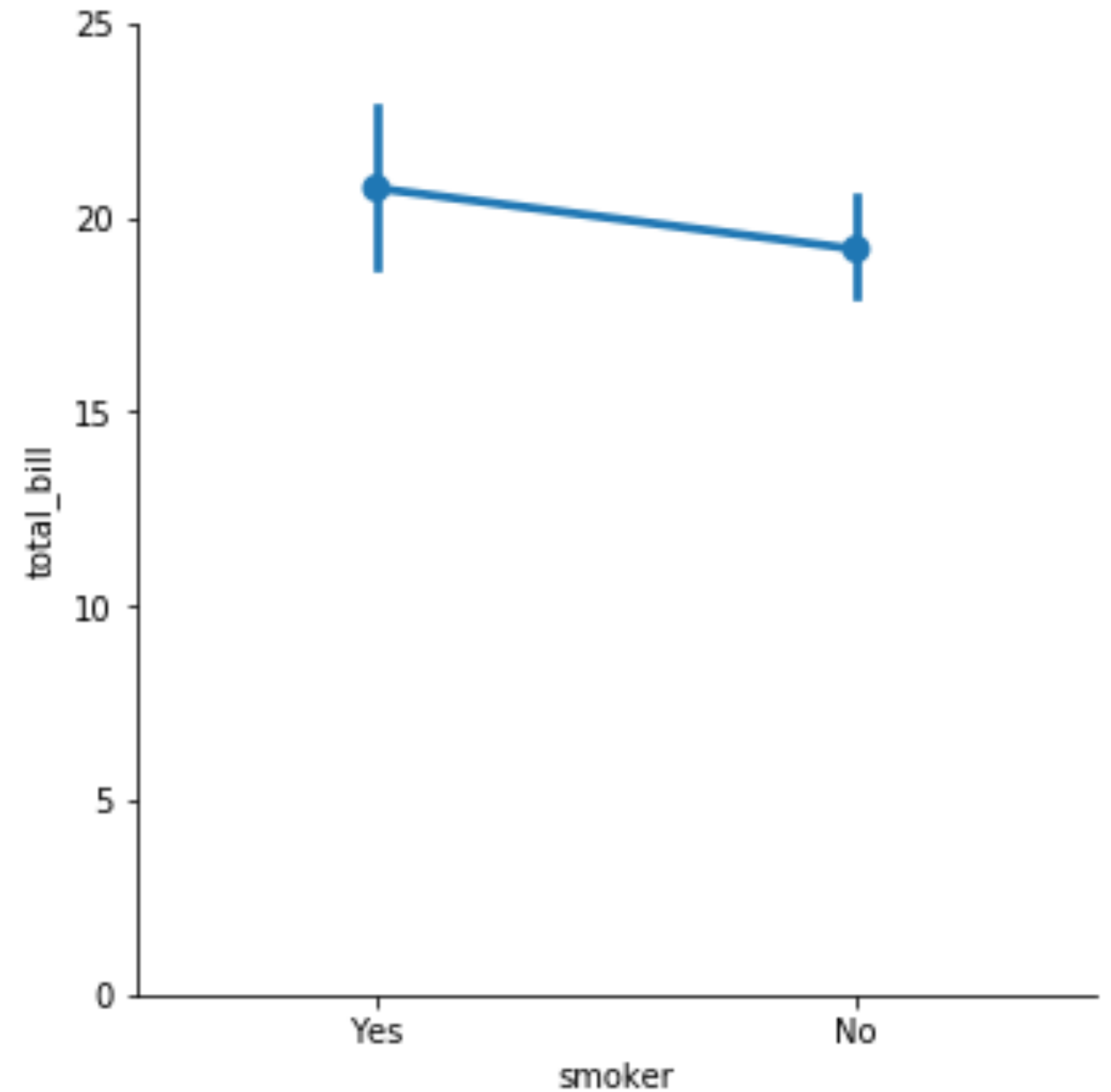
- Points show mean of quantitative variable
- Vertical lines show 95% confidence intervals



**Line plot:** average level of nitrogen dioxide over time



**Point plot:** average restaurant bill, smokers vs. non-smokers



# Point plots vs. line plots

Both show:

- Mean of quantitative variable
- 95% confidence intervals for the mean

Differences:

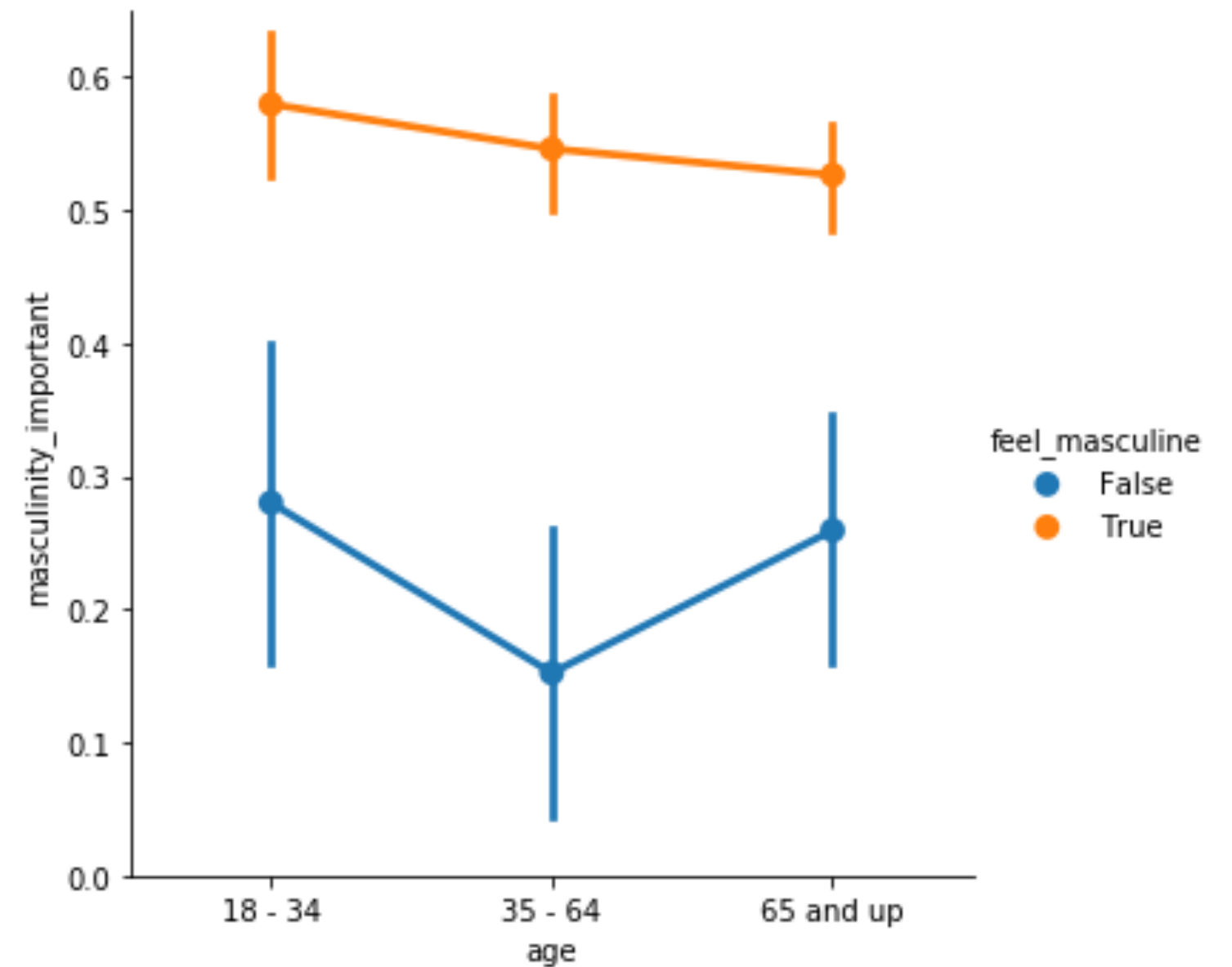
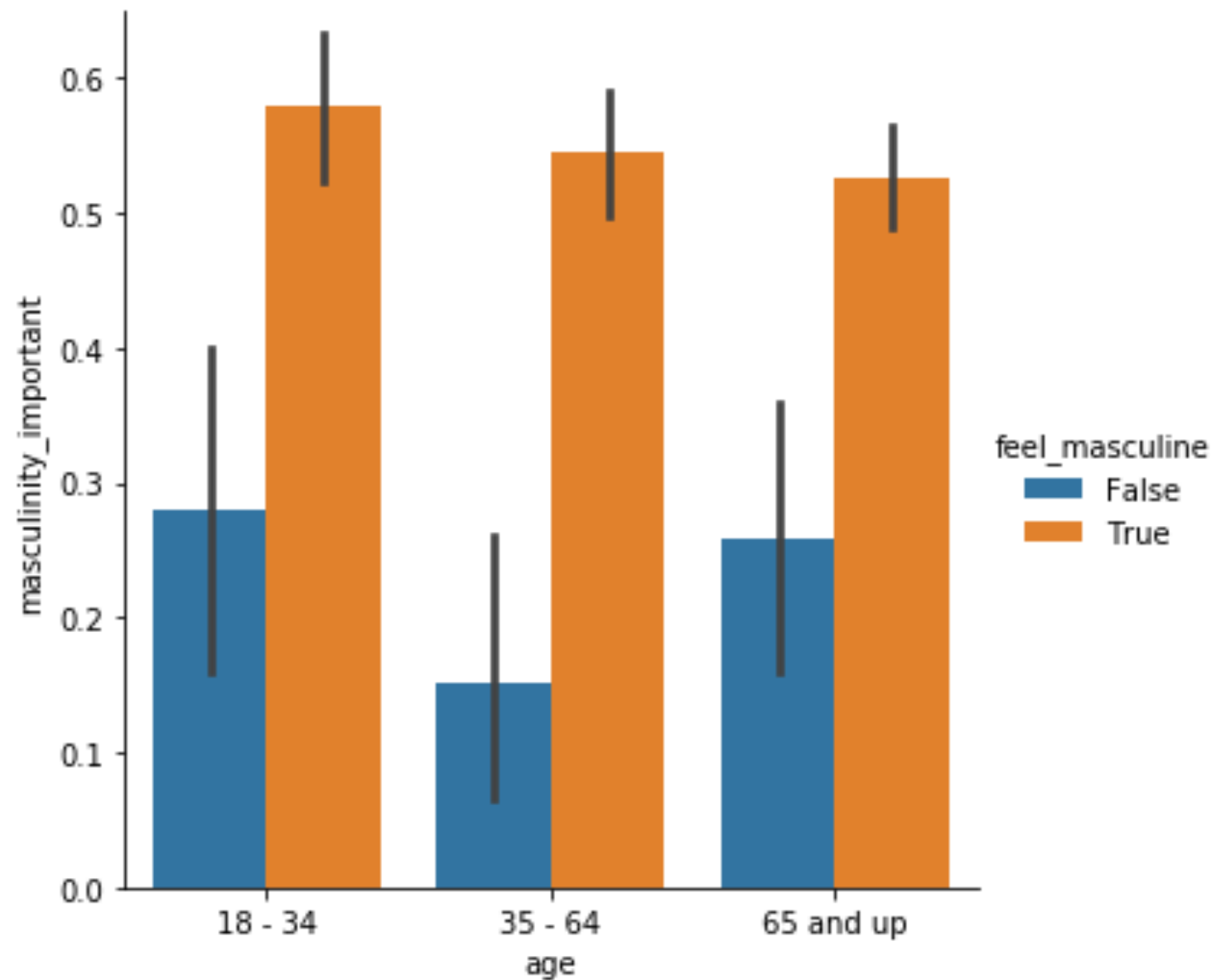
- Line plot has **quantitative** variable (usually time) on x-axis
- Point plot has **categorical** variable on x-axis

# Point plots vs. bar plots

Both show:

- Mean of quantitative variable
- 95% confidence intervals for the mean

# Point plots vs. bar plots



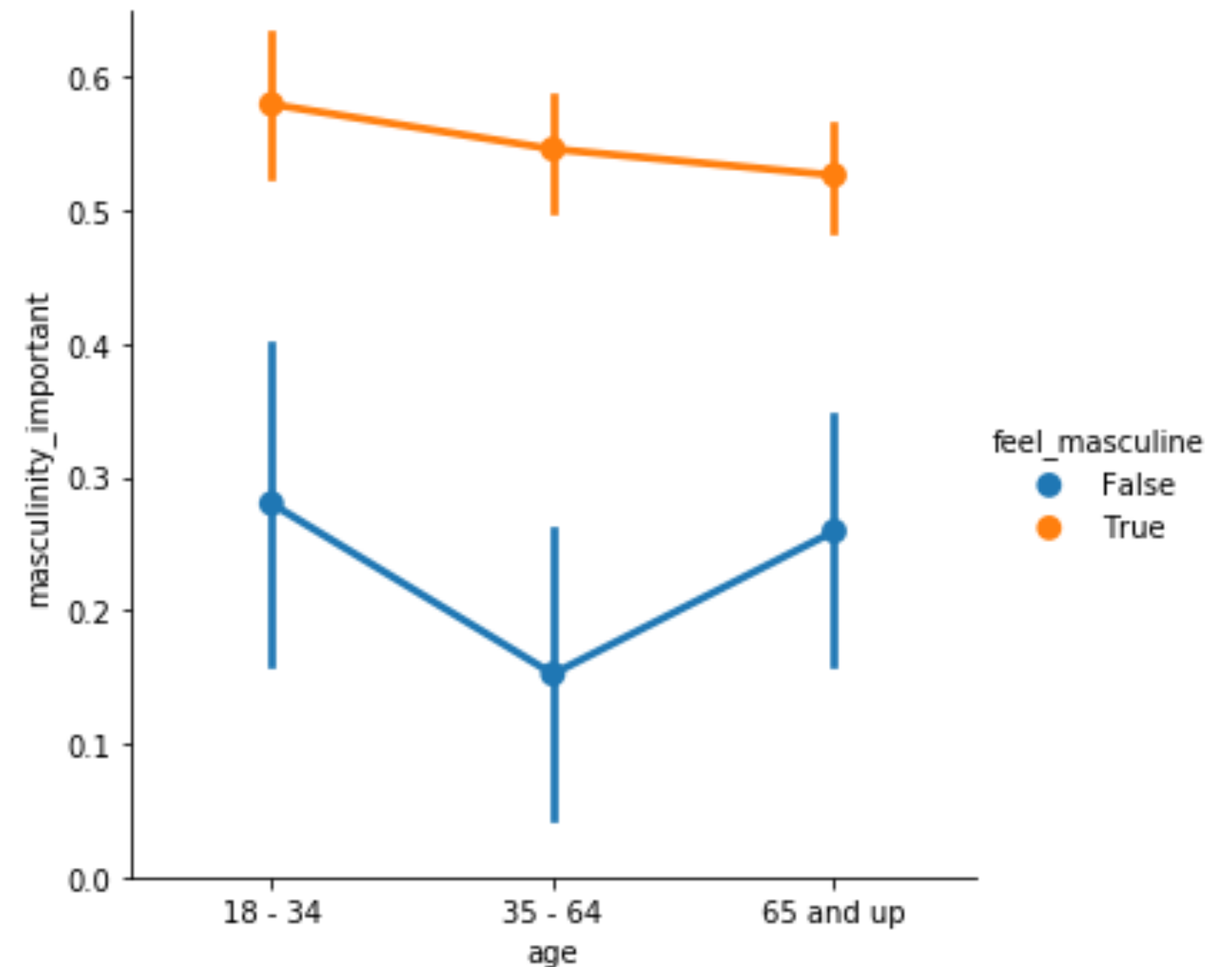


# Creating a point plot

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

sns.catplot(x="age",
            y="masculinity_important",
            data=masculinity_data,
            hue="feel_masculine",
            kind="point")

plt.show()
```

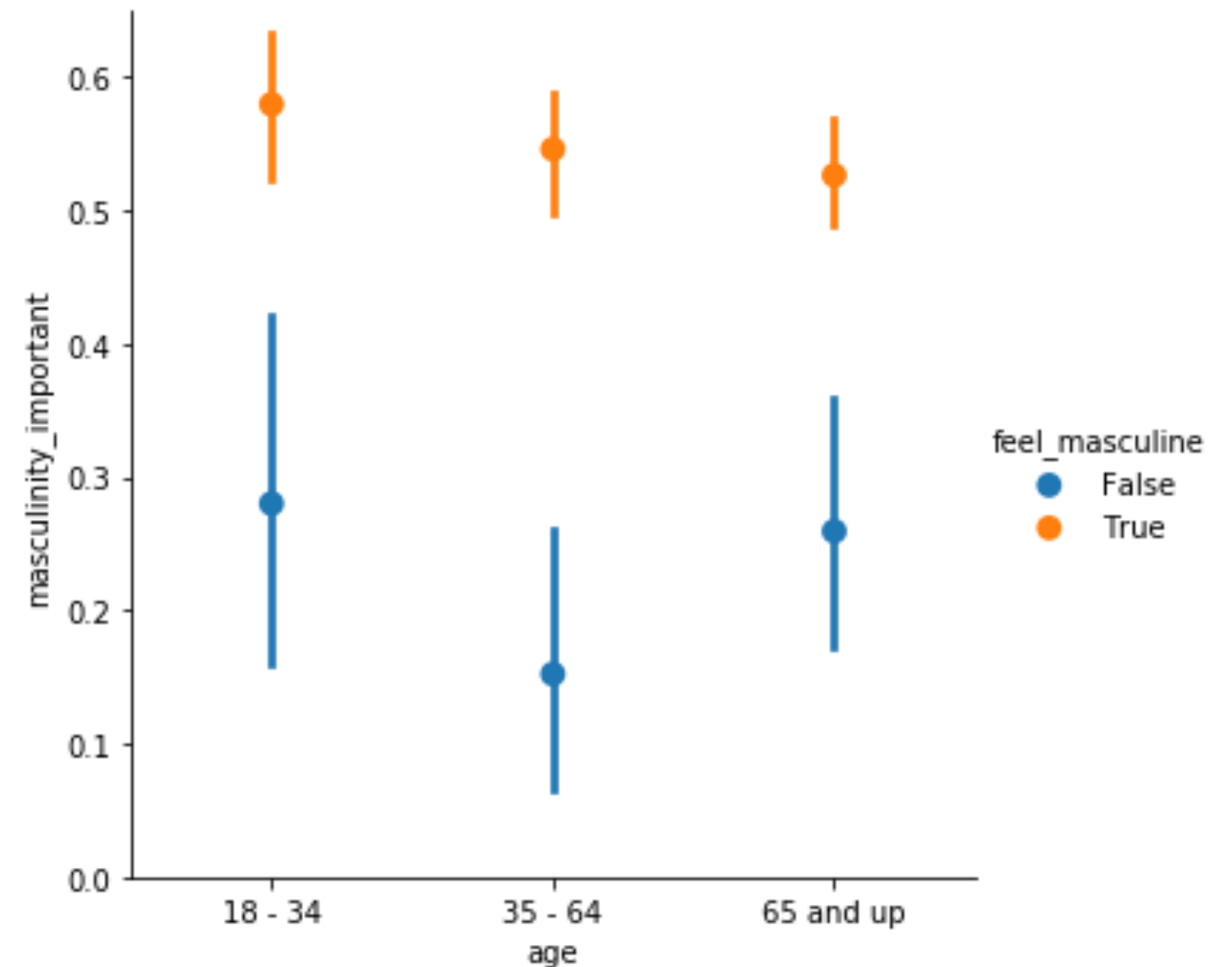


# Disconnecting the points

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

sns.catplot(x="age",
            y="masculinity_important",
            data=masculinity_data,
            hue="feel_masculine",
            kind="point",
            join=False)

plt.show()
```

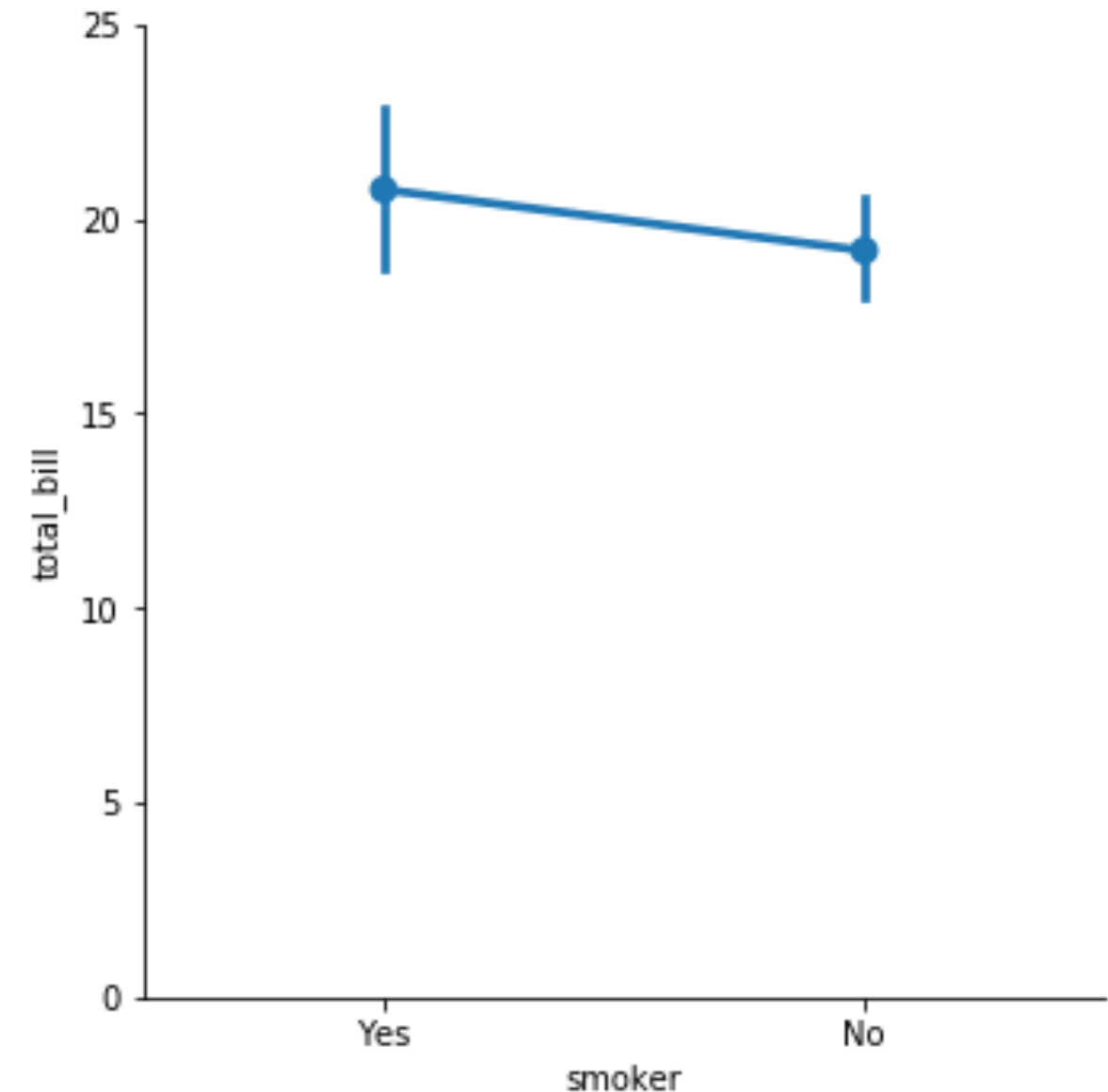


# Displaying the median

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

sns.catplot(x="smoker",
            y="total_bill",
            data=tips,
            kind="point")

plt.show()
```

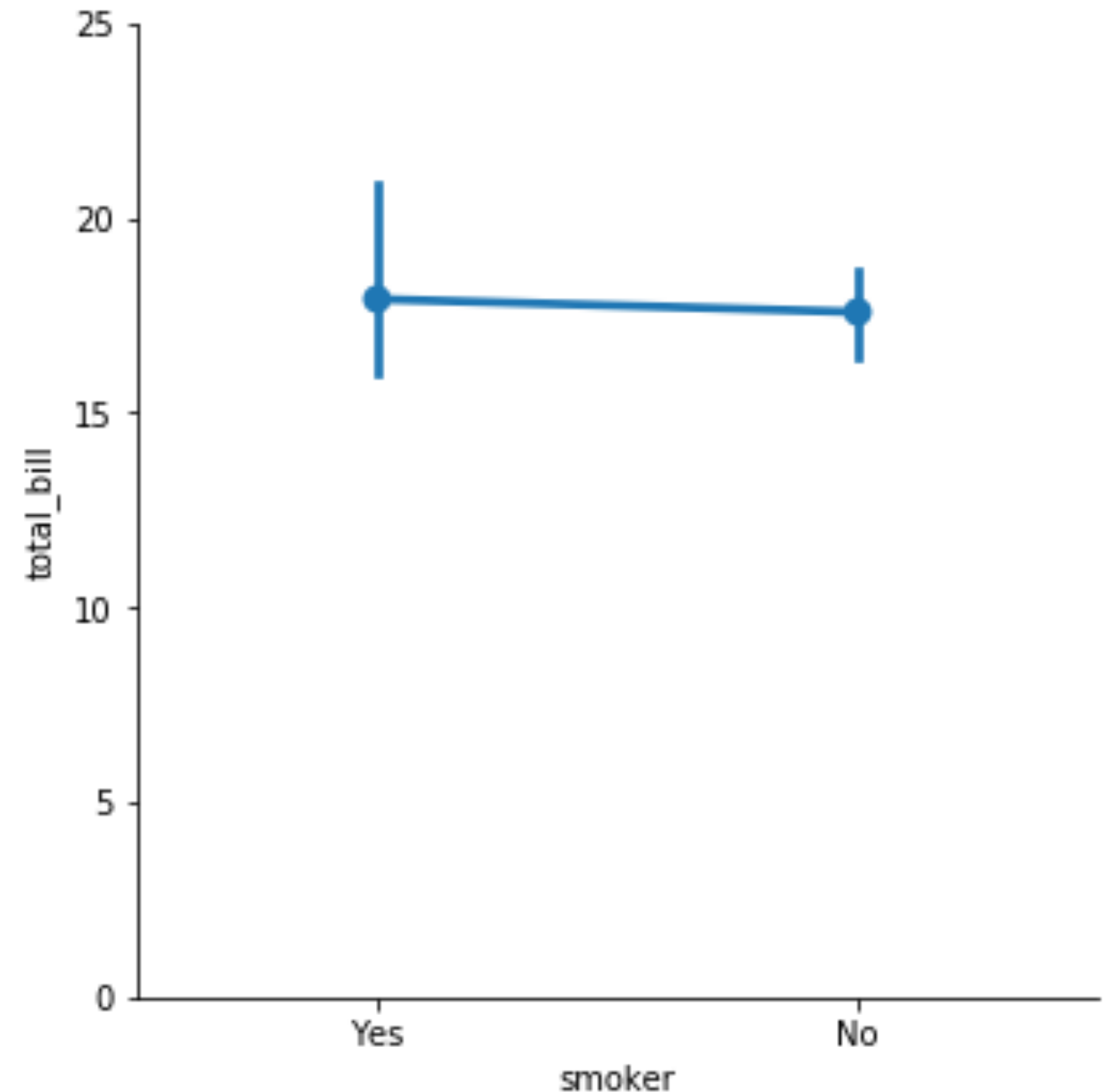


# Displaying the median

```
import matplotlib.pyplot as plt
import seaborn as sns
from numpy import median

sns.catplot(x="smoker",
            y="total_bill",
            data=tips,
            kind="point",
            estimator=median)

plt.show()
```

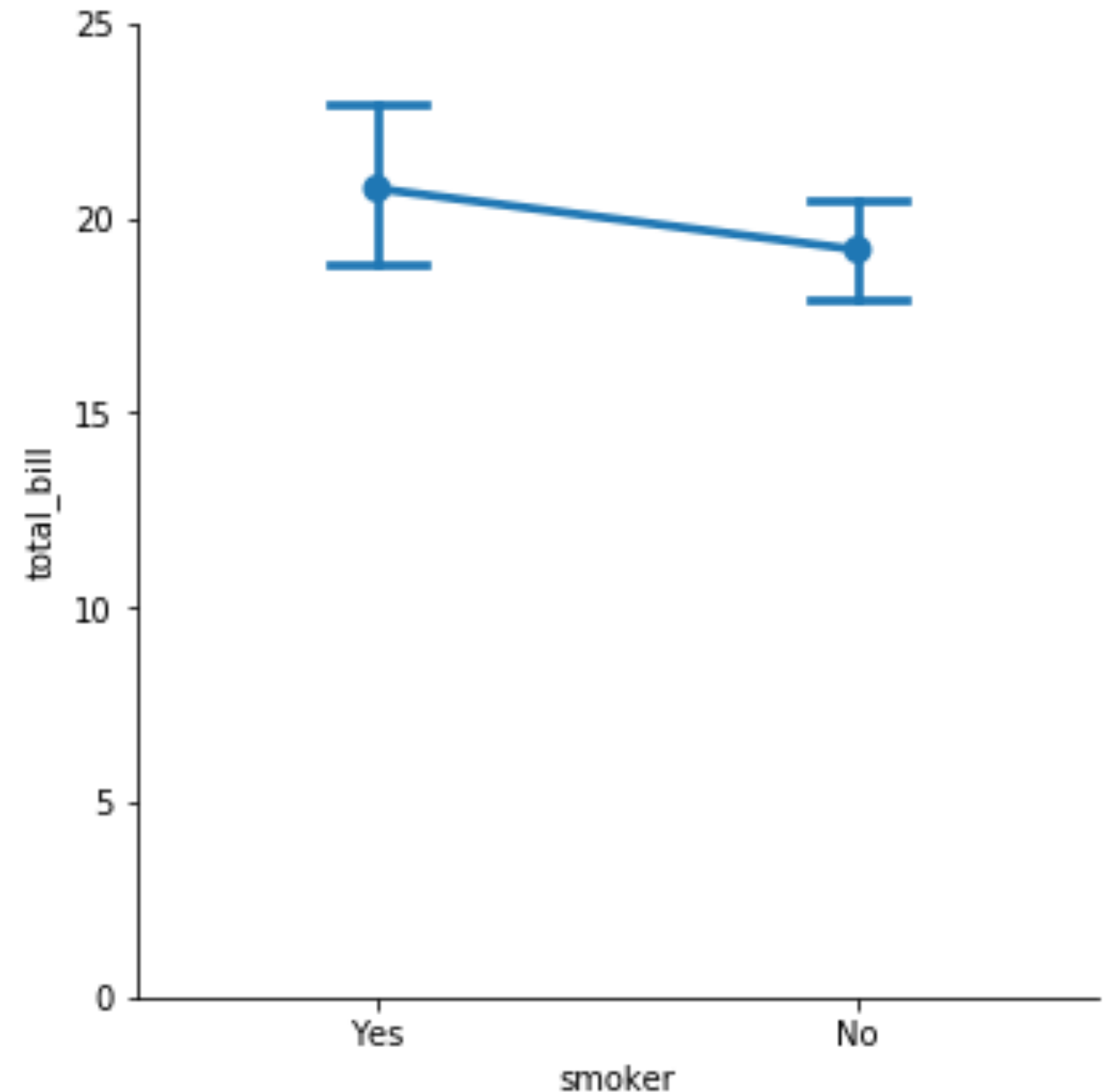


# Customizing the confidence intervals

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

sns.catplot(x="smoker",
            y="total_bill",
            data=tips,
            kind="point",
            capsize=0.2)

plt.show()
```

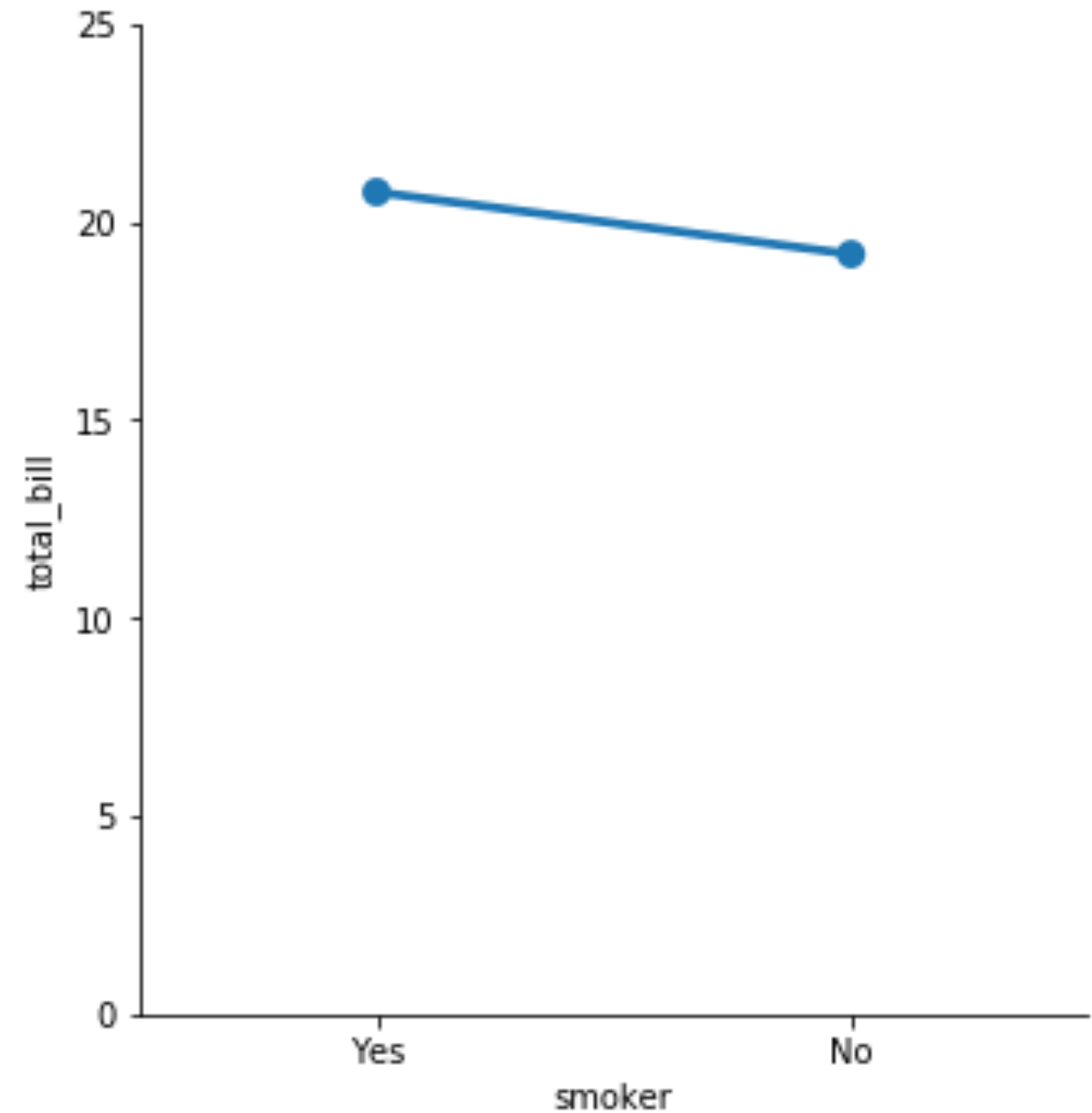


# Turning off confidence intervals

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

sns.catplot(x="smoker",
            y="total_bill",
            data=tips,
            kind="point",
            ci=None)

plt.show()
```



# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN