

# Changing plot style and color

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



**Erin Case**  
Data Scientist

# Why customize?

Reasons to change style:

- Personal preference
- Improve readability
- Guide interpretation

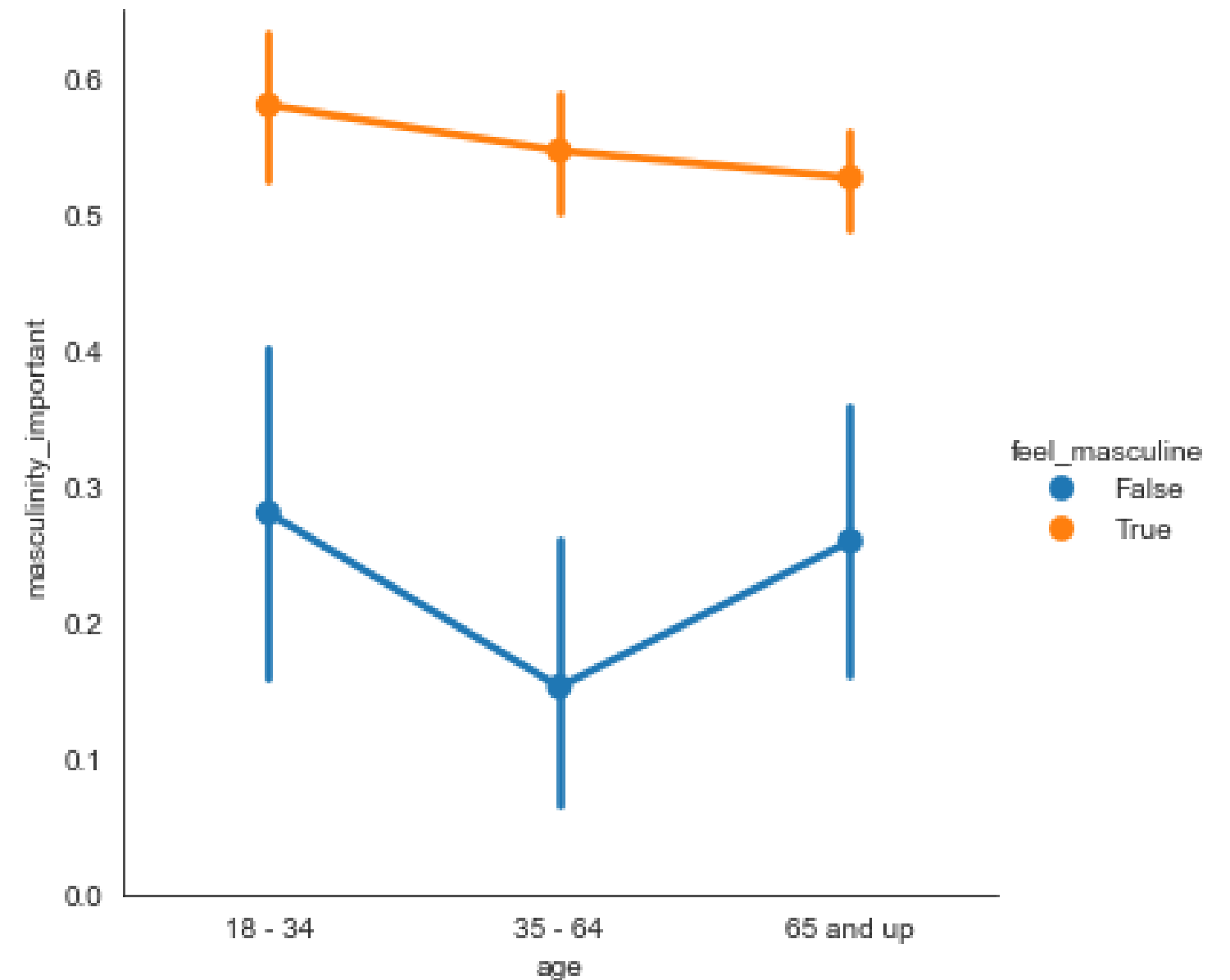
# Changing the figure style

- Figure "style" includes background and axes
- Preset options: "white", "dark", "whitegrid", "darkgrid", "ticks"
- `sns.set_style()`

# Default figure style ("white")

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

```
plt.show()
```

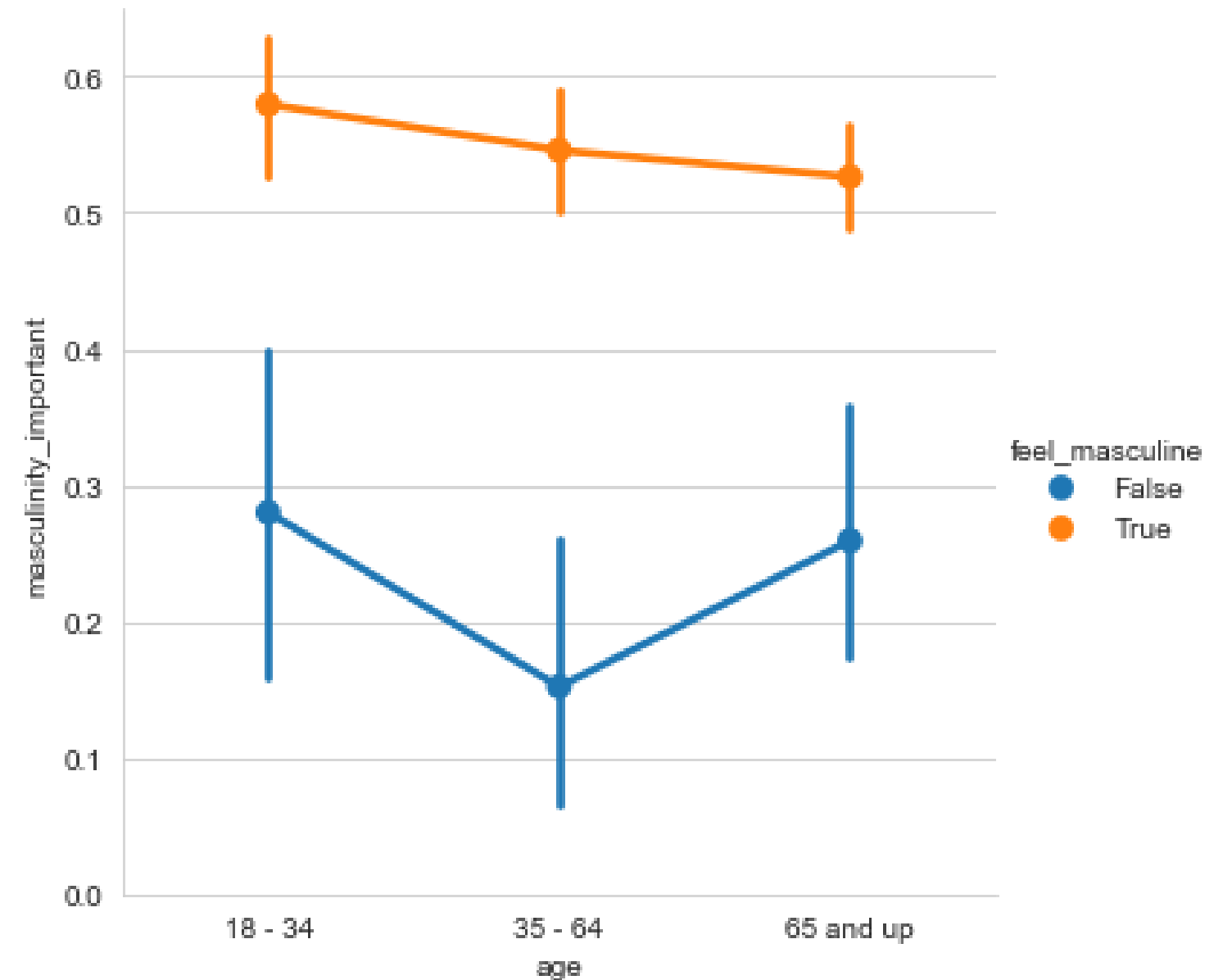


# Figure style: "whitegrid"

```
sns.set_style("whitegrid")

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

plt.show()
```

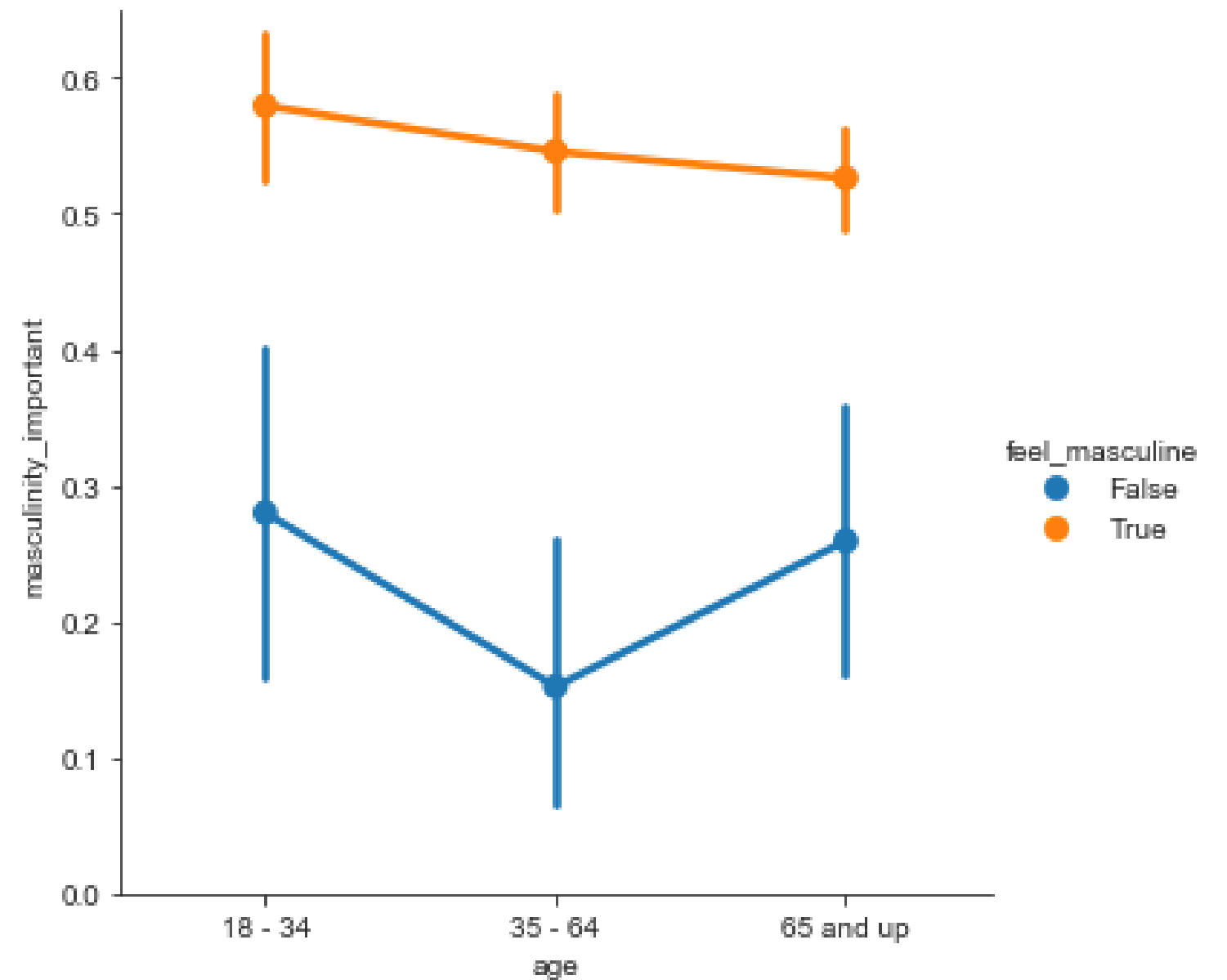


# Other styles

```
sns.set_style("ticks")

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

plt.show()
```

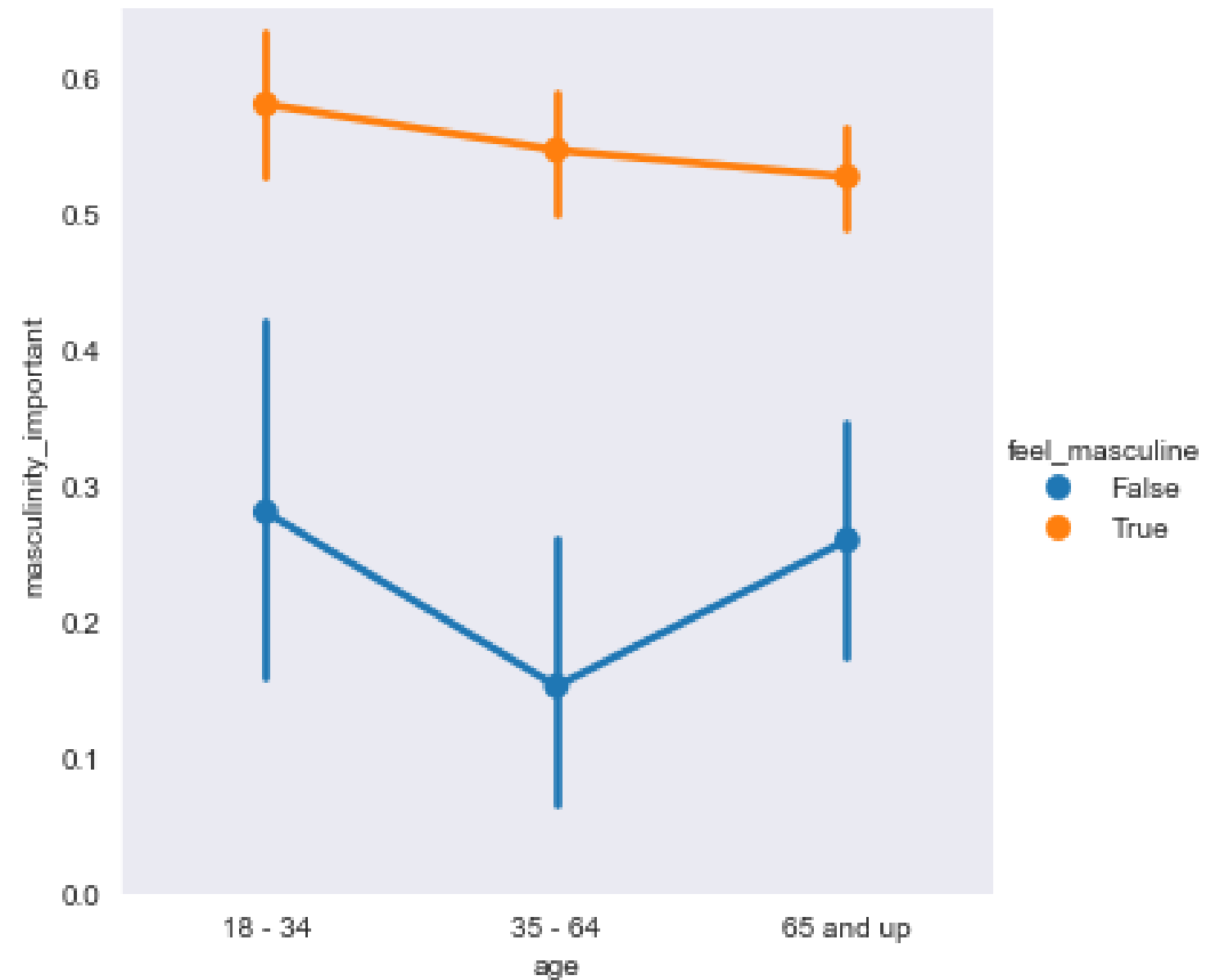


# Other styles

```
sns.set_style("dark")

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

plt.show()
```

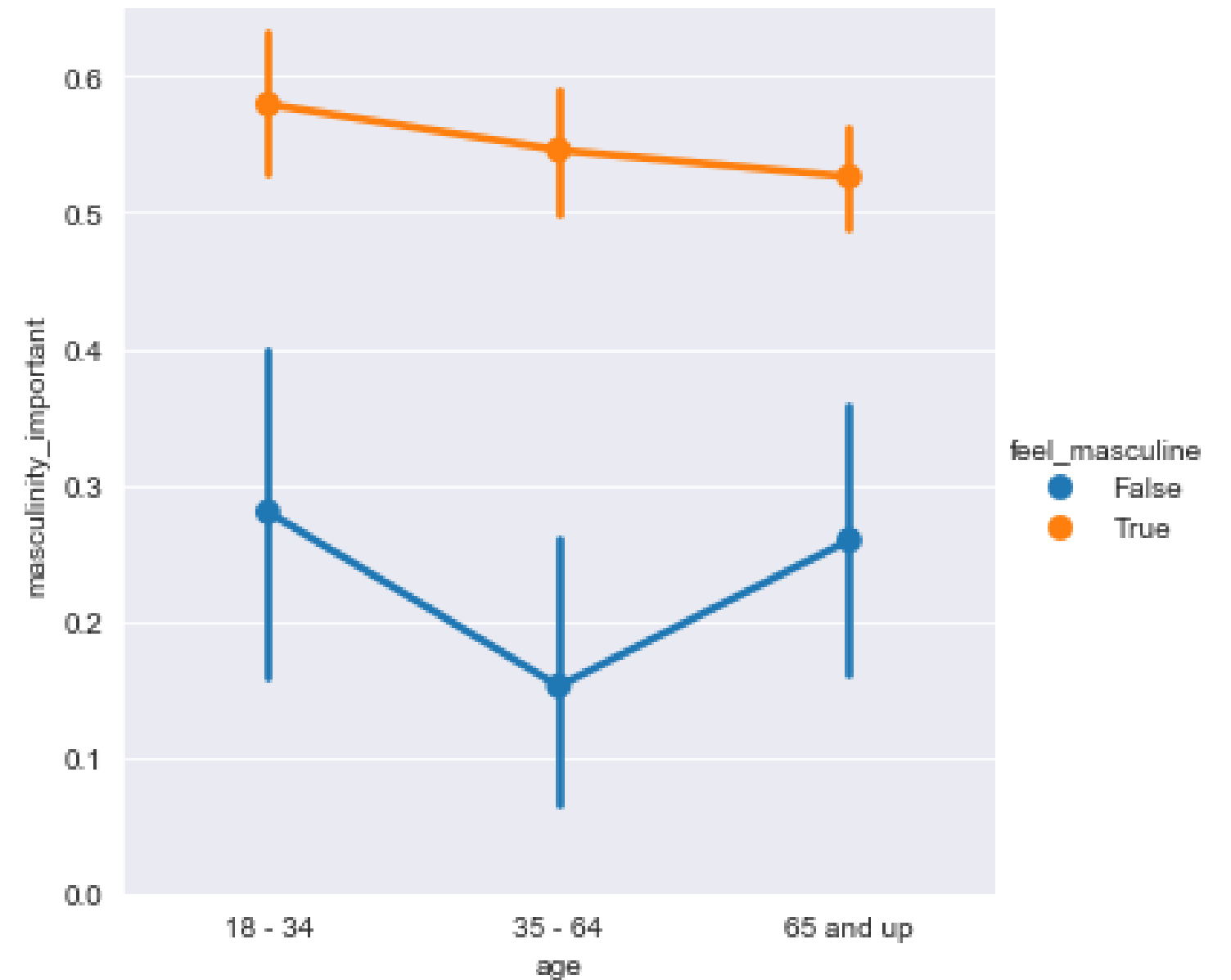


# Other styles

```
sns.set_style("darkgrid")

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

plt.show()
```









# Changing the palette

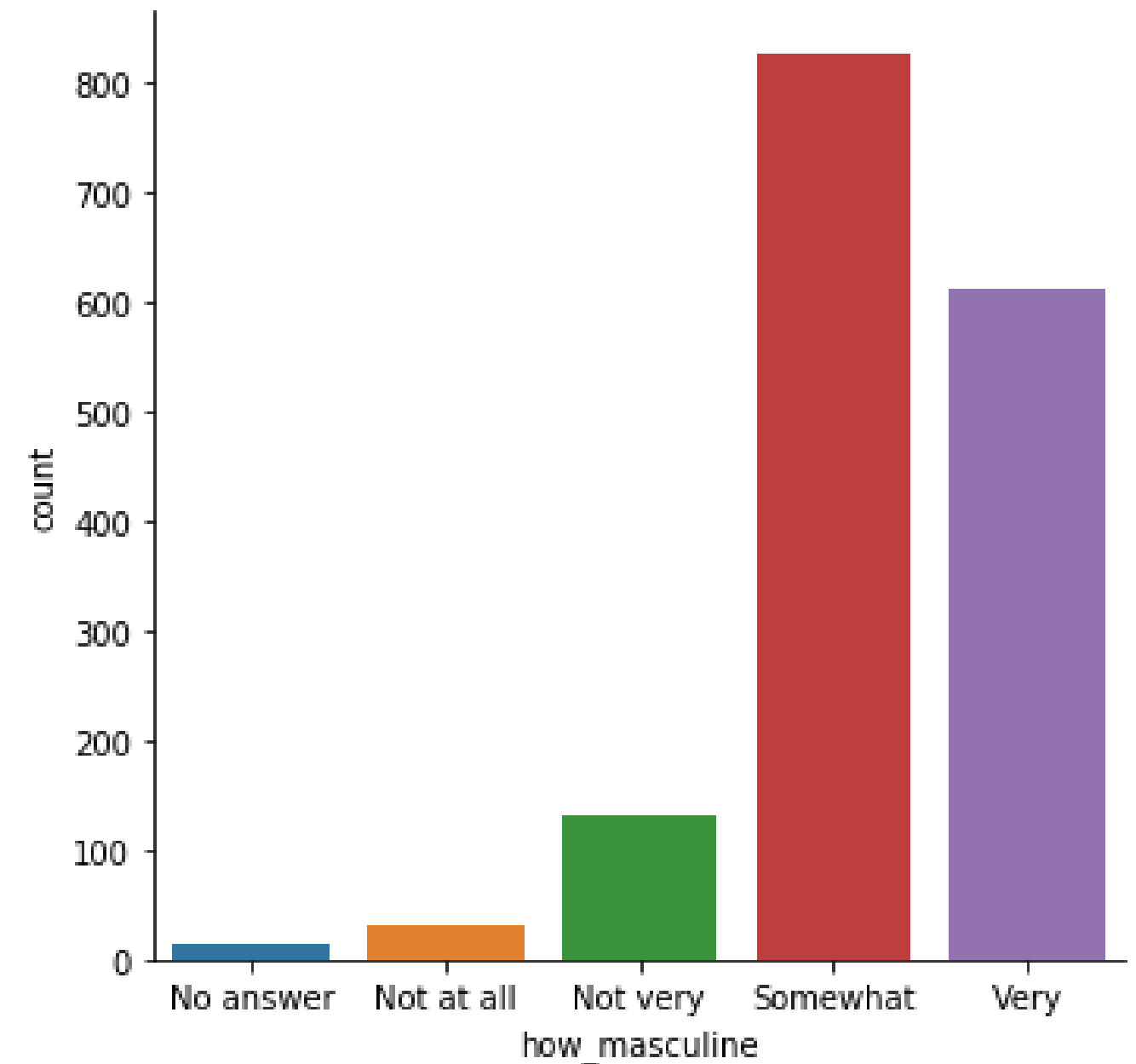
- Figure "palette" changes the color of the main elements of the plot
- `sns.set_palette()`
- Use preset palettes or create a custom palette

# Diverging palettes

|          |  |
|----------|--|
| "RdBu"   |    |
| "PRGn"   |    |
| "RdBu_r" |   |
| "PRGn_r" |  |

# Example (default palette)

```
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()
```



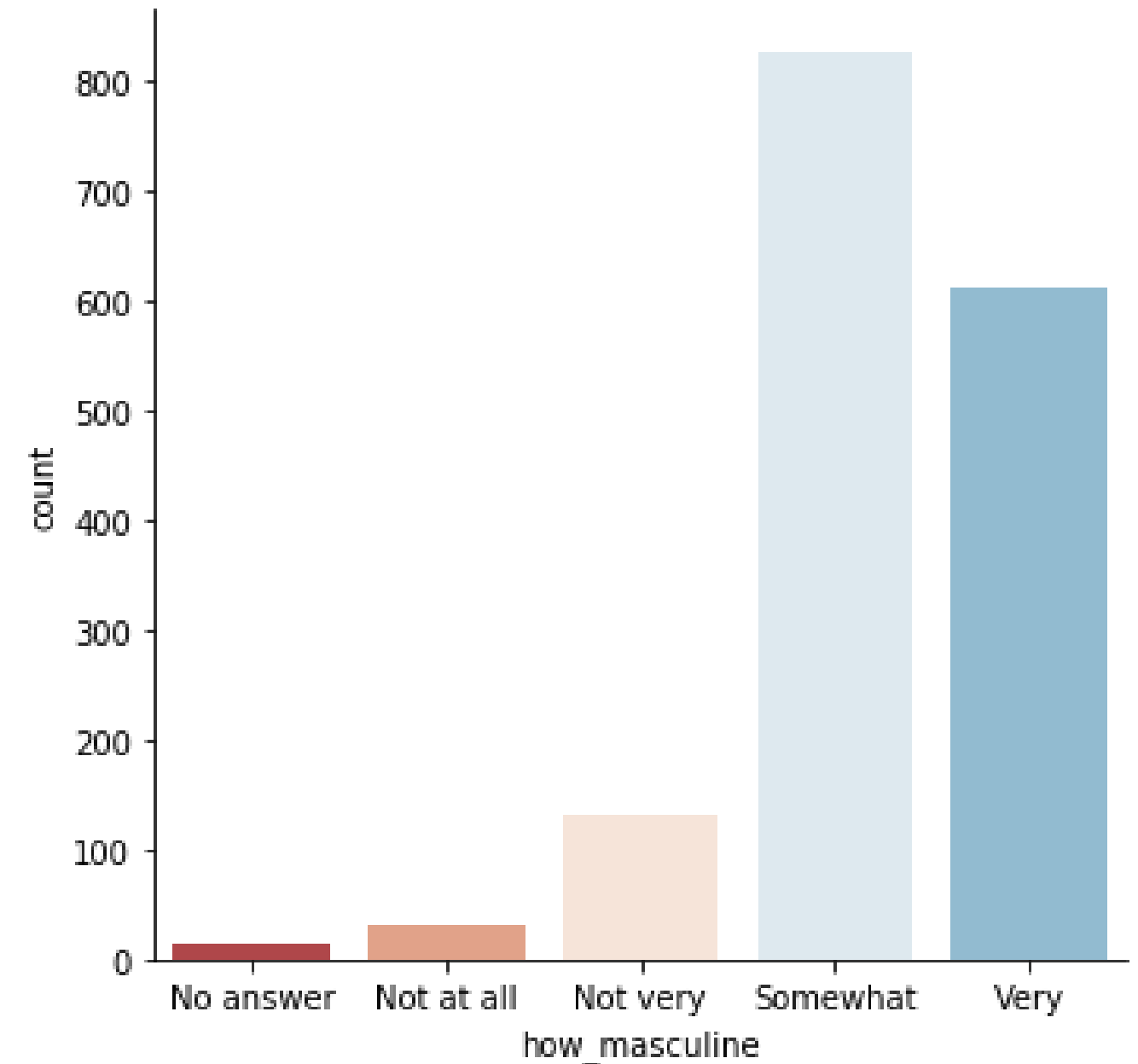
# Example (diverging palette)

```
sns.set_palette("RdBu")





category_order = ["No answer",
                  "Not at all",
                  "Not very",
                  "Somewhat",
                  "Very"]

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

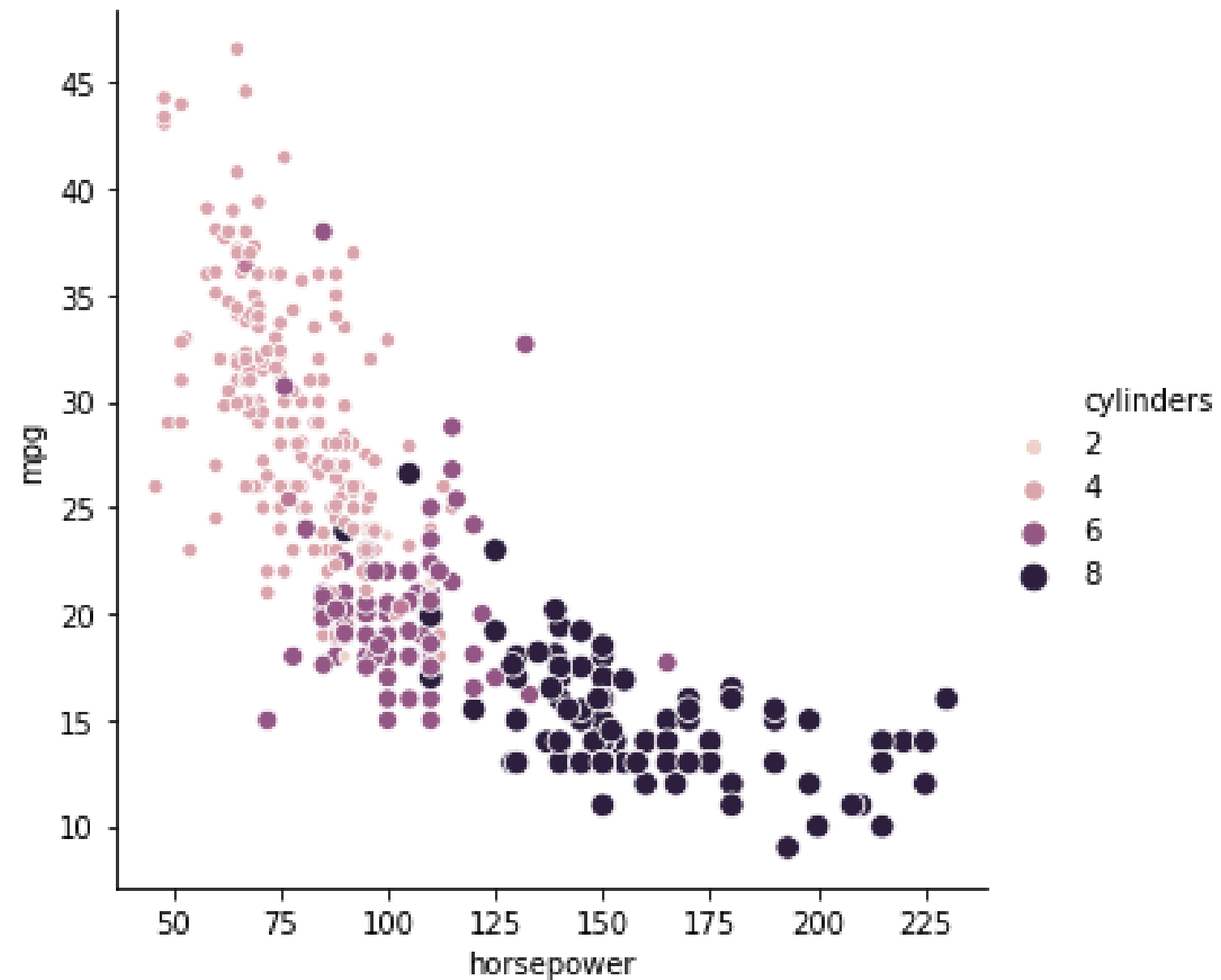
plt.show()
```



# Sequential palettes

|         |  |
|---------|--|
| "Greys" |    |
| "Blues" |    |
| "PuRd"  |   |
| "GnBu"  |  |

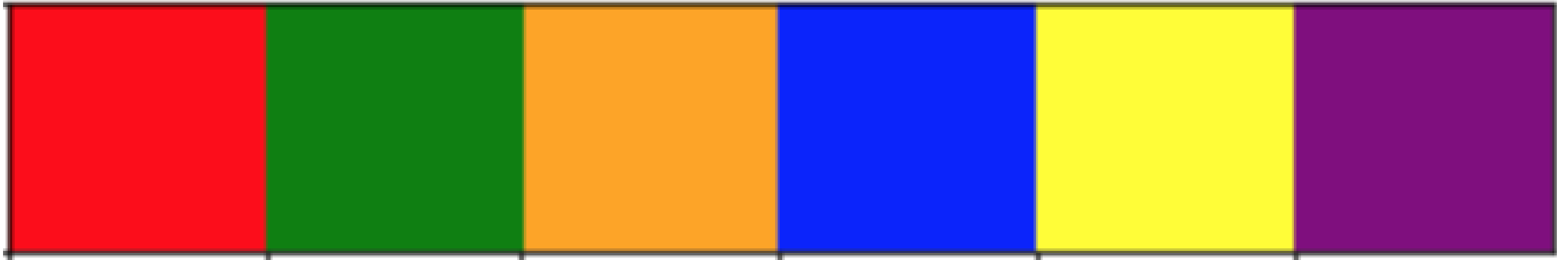
# Sequential palette example



# Custom palettes

```
custom_palette = ["red", "green", "orange", "blue",  
                  "yellow", "purple"]
```

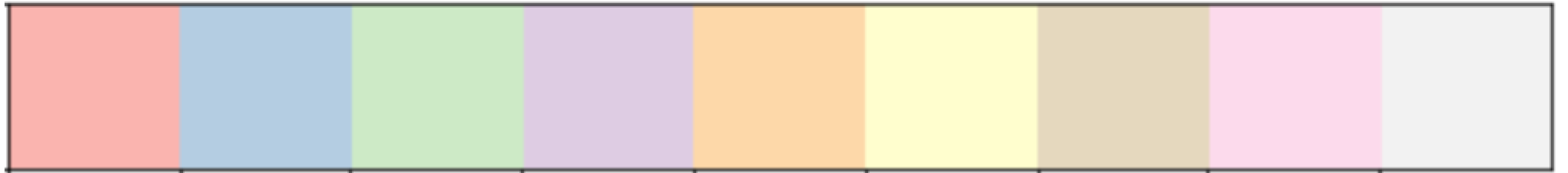
```
sns.set_palette(custom_palette)
```



# Custom palettes

```
custom_palette = ['#FBB4AE', '#B3CDE3', '#CCEBC5',  
                  '#DECBE4', '#FED9A6', '#FFFFCC',  
                  '#E5D8BD', '#FDDAEC', '#F2F2F2']
```

```
sns.set_palette(custom_palette)
```



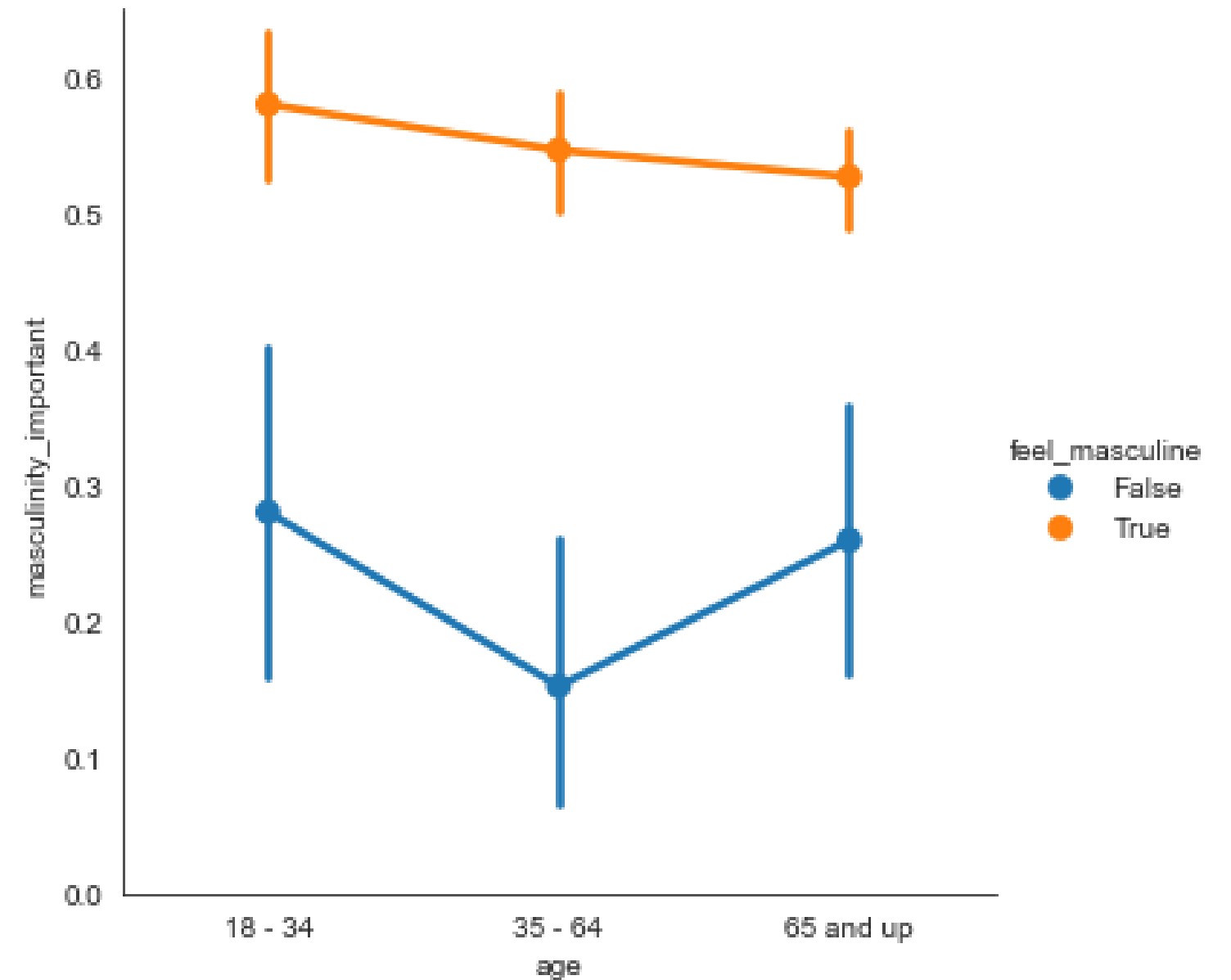


# Changing the scale

- Figure "context" changes the scale of the plot elements and labels
- `sns.set_context()`
- Smallest to largest: "paper", "notebook", "talk", "poster"

# Default context: "paper"

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

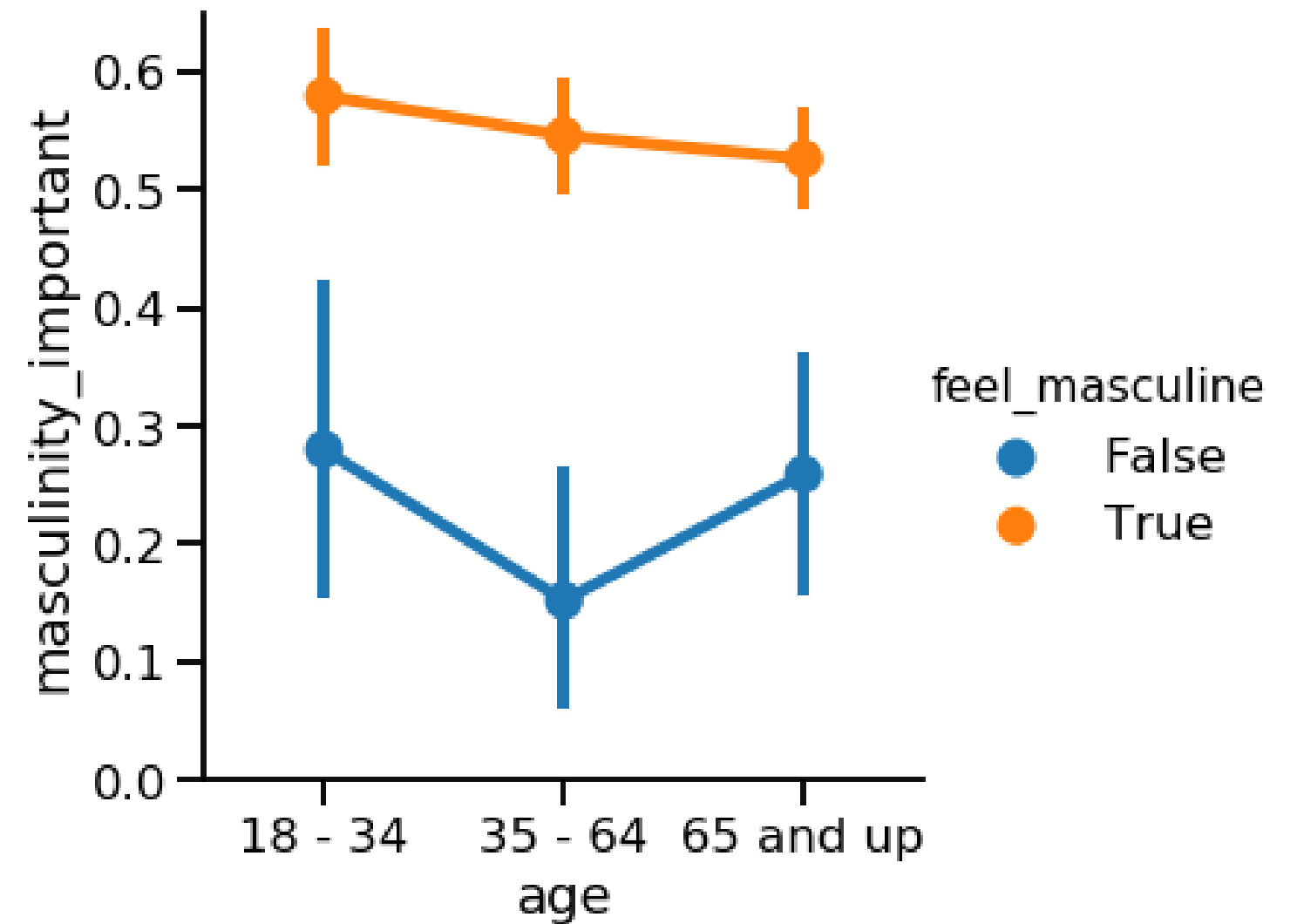


# Larger context: "talk"

```
sns.set_context("talk")

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

plt.show()
```



# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

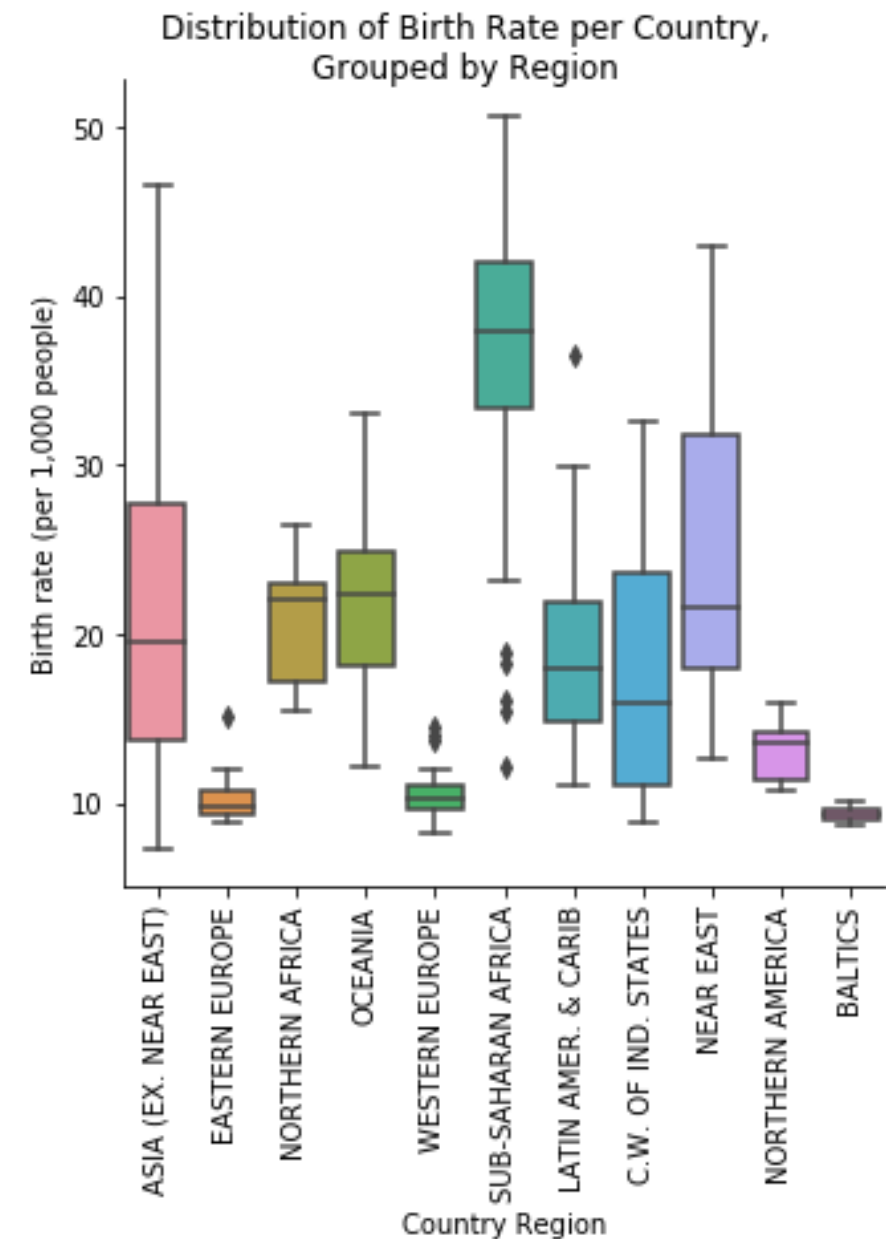
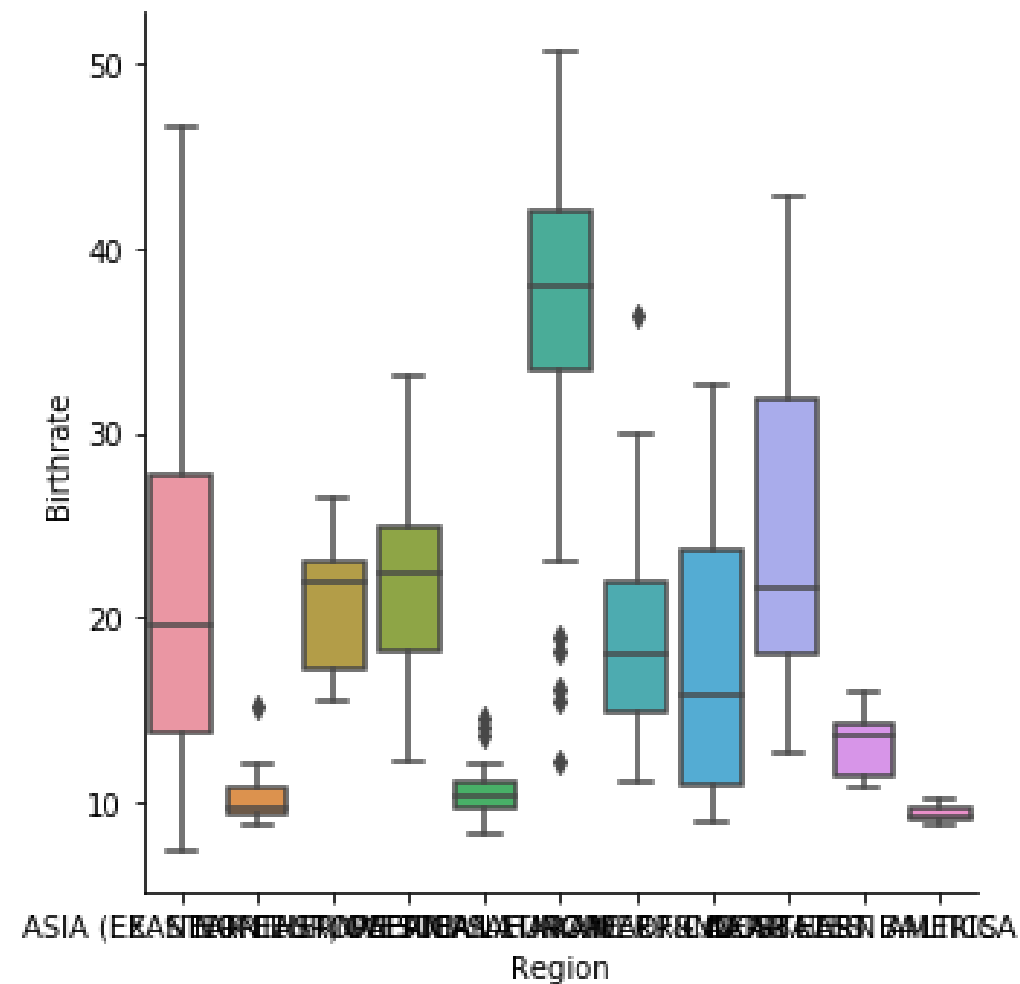
# Adding titles and labels: Part 1

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



**Erin Case**  
Data Scientist

# Creating informative visualizations



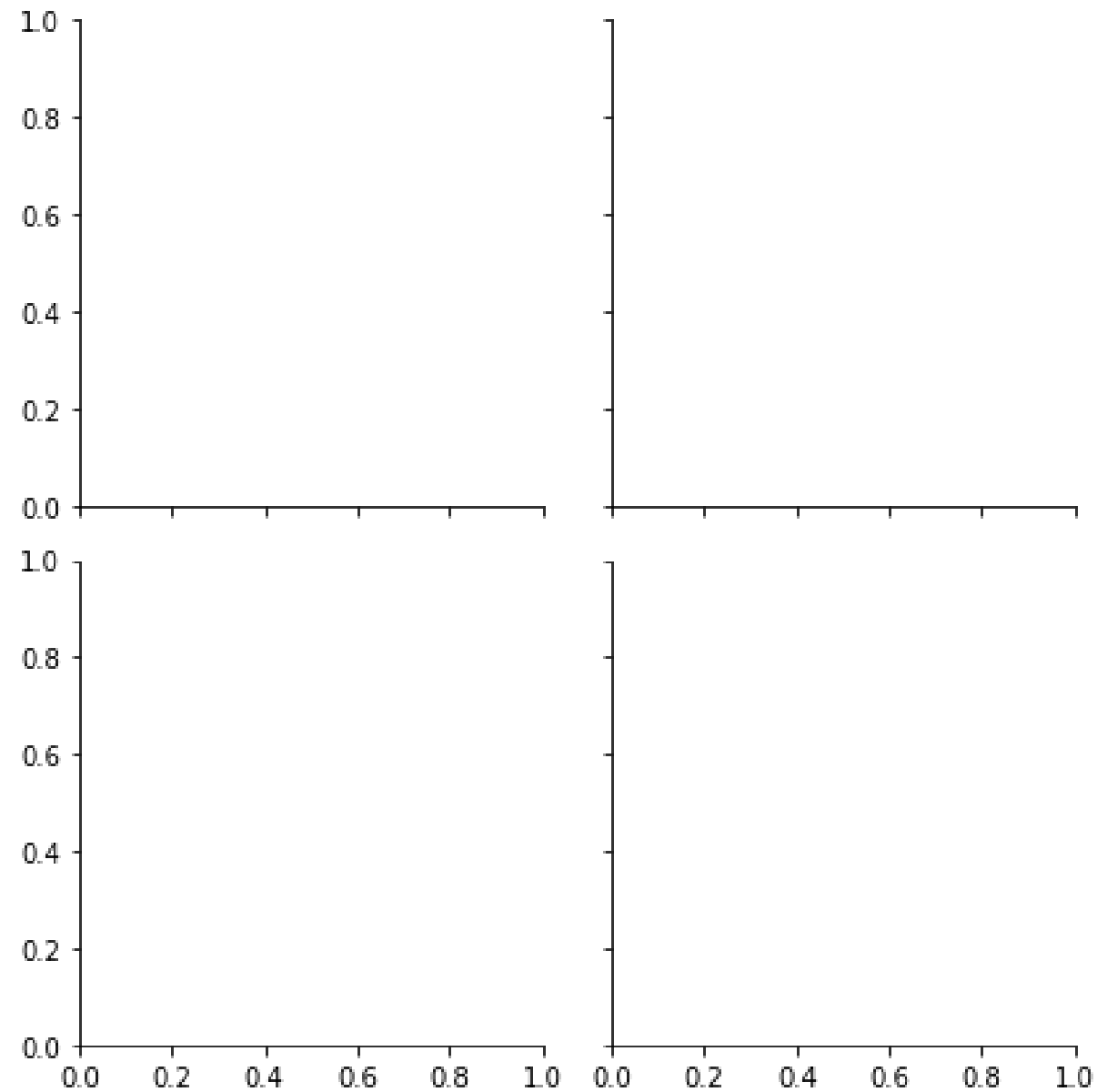
# FacetGrid vs. AxesSubplot objects

Seaborn plots create two different types of objects: `FacetGrid` and `AxesSubplot`

```
g = sns.scatterplot(x="height", y="weight", data=df)
type(g)
```

```
> matplotlib.axes._subplots.AxesSubplot
```

# An Empty FacetGrid



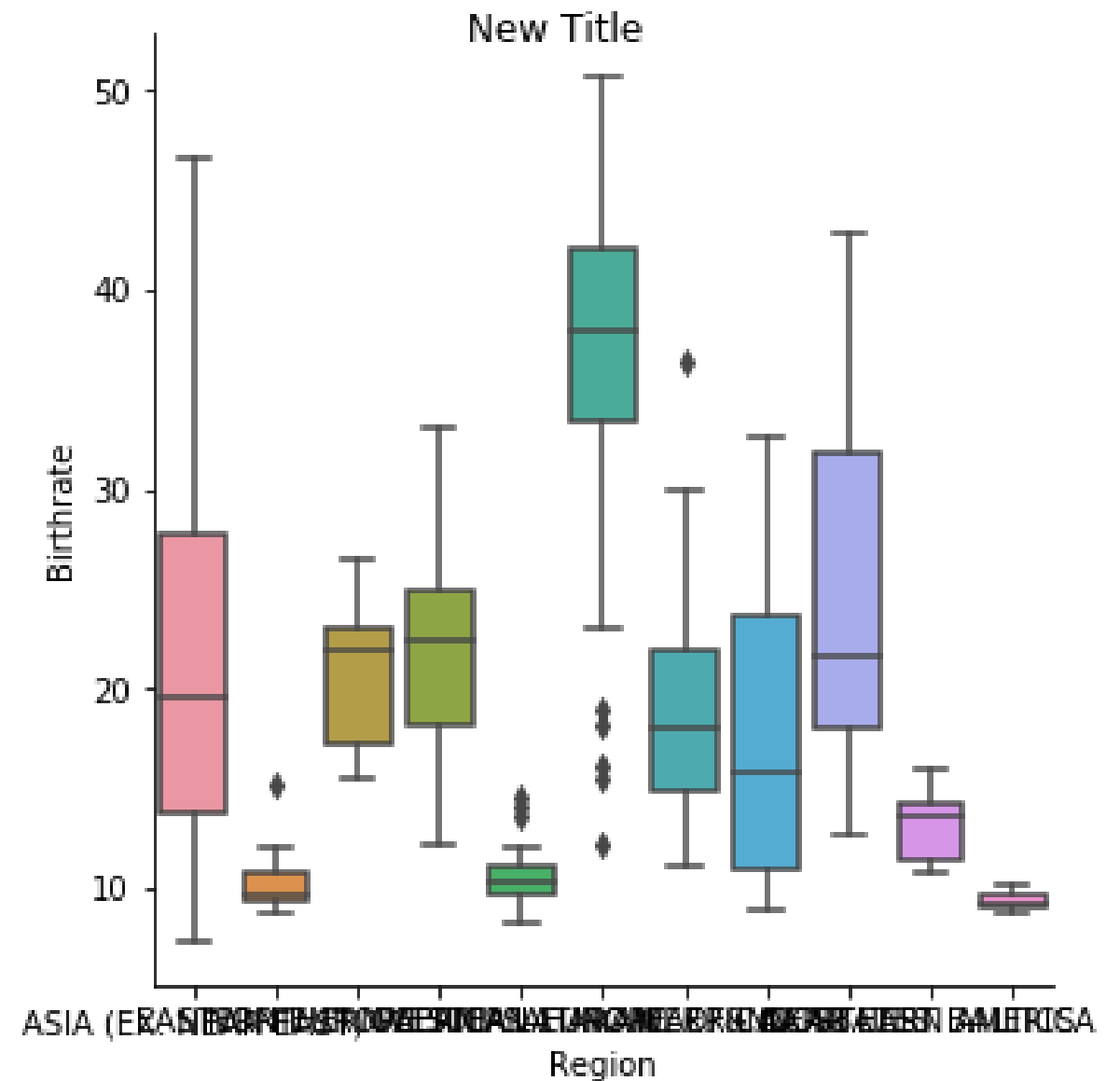


# FacetGrid vs. AxesSubplot objects

| Object Type | Plot Types                         | Characteristics            |
|-------------|------------------------------------|----------------------------|
| FacetGrid   | relplot() , catplot()              | Can create subplots        |
| AxesSubplot | scatterplot() , countplot() , etc. | Only creates a single plot |

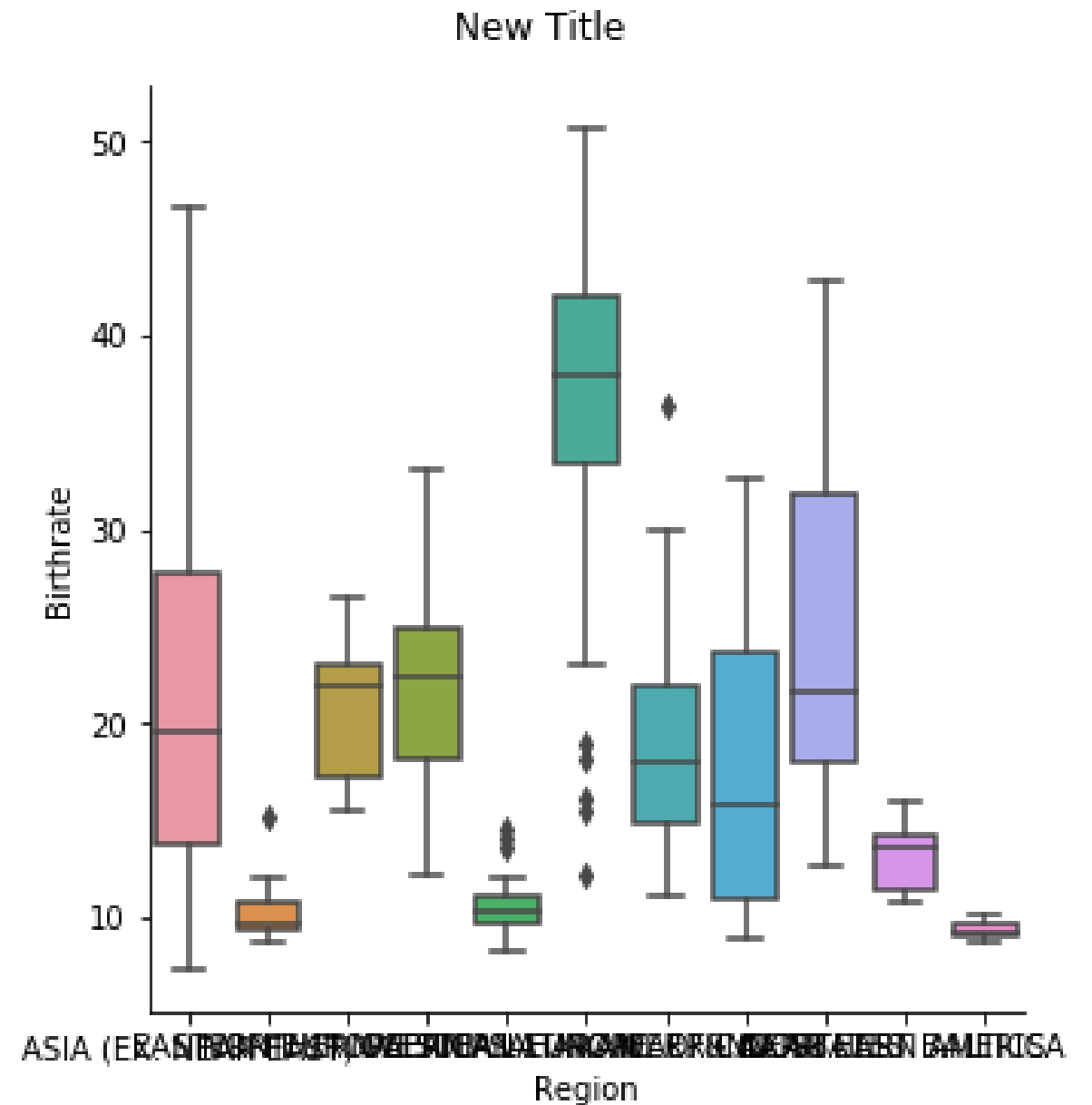
# Adding a title to FacetGrid

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")  
  
g.fig.suptitle("New Title")  
plt.show()
```



# Adjusting height of title in FacetGrid

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")  
  
g.fig.suptitle("New Title",  
              y=1.03)  
  
plt.show()
```



# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

# Adding titles and labels: Part 2

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



**Erin Case**  
Data Scientist

# Adding a title to AxesSubplot

## FacetGrid

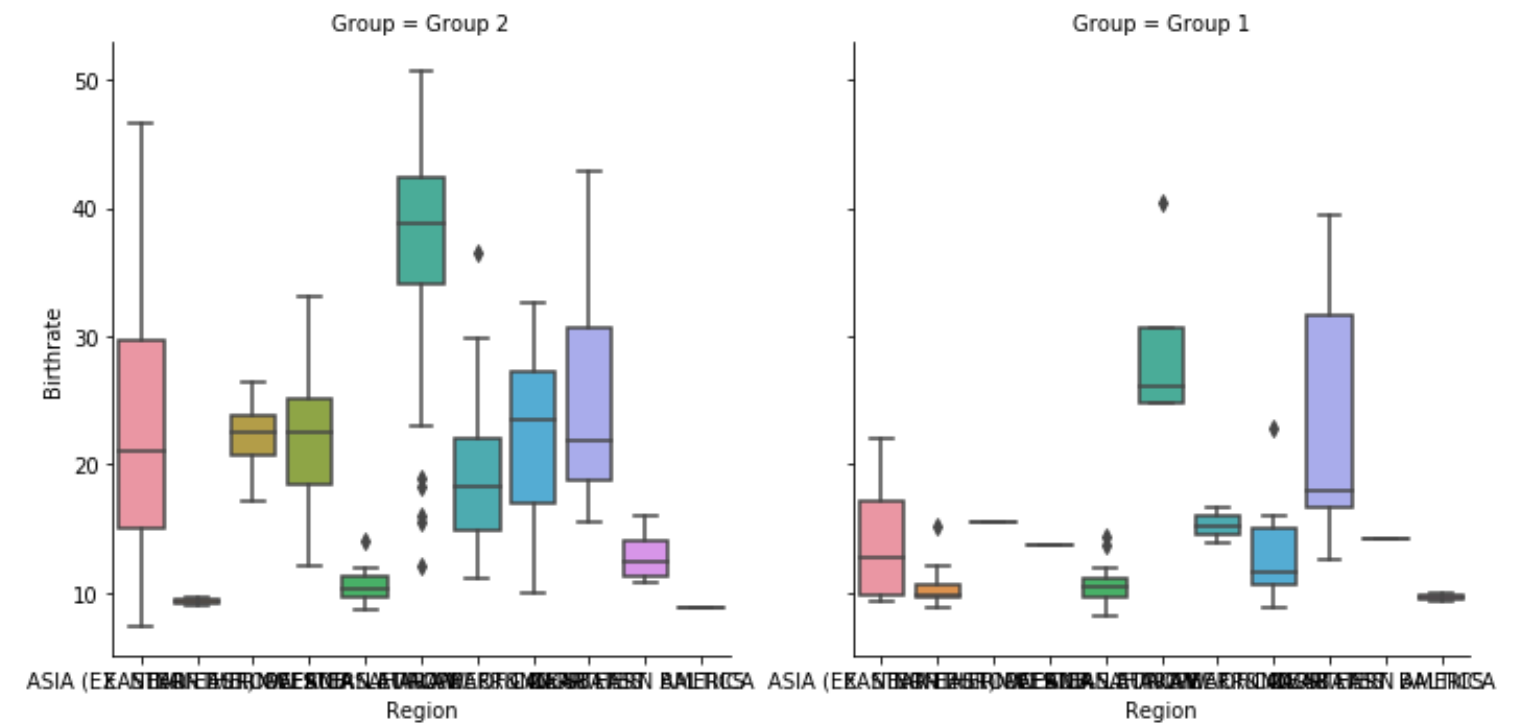
```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")  
  
g.fig.suptitle("New Title",  
               y=1.03)
```

## AxesSubplot

```
g = sns.boxplot(x="Region",  
                y="Birthrate",  
                data=gdp_data)  
  
g.set_title("New Title",  
            y=1.03)
```

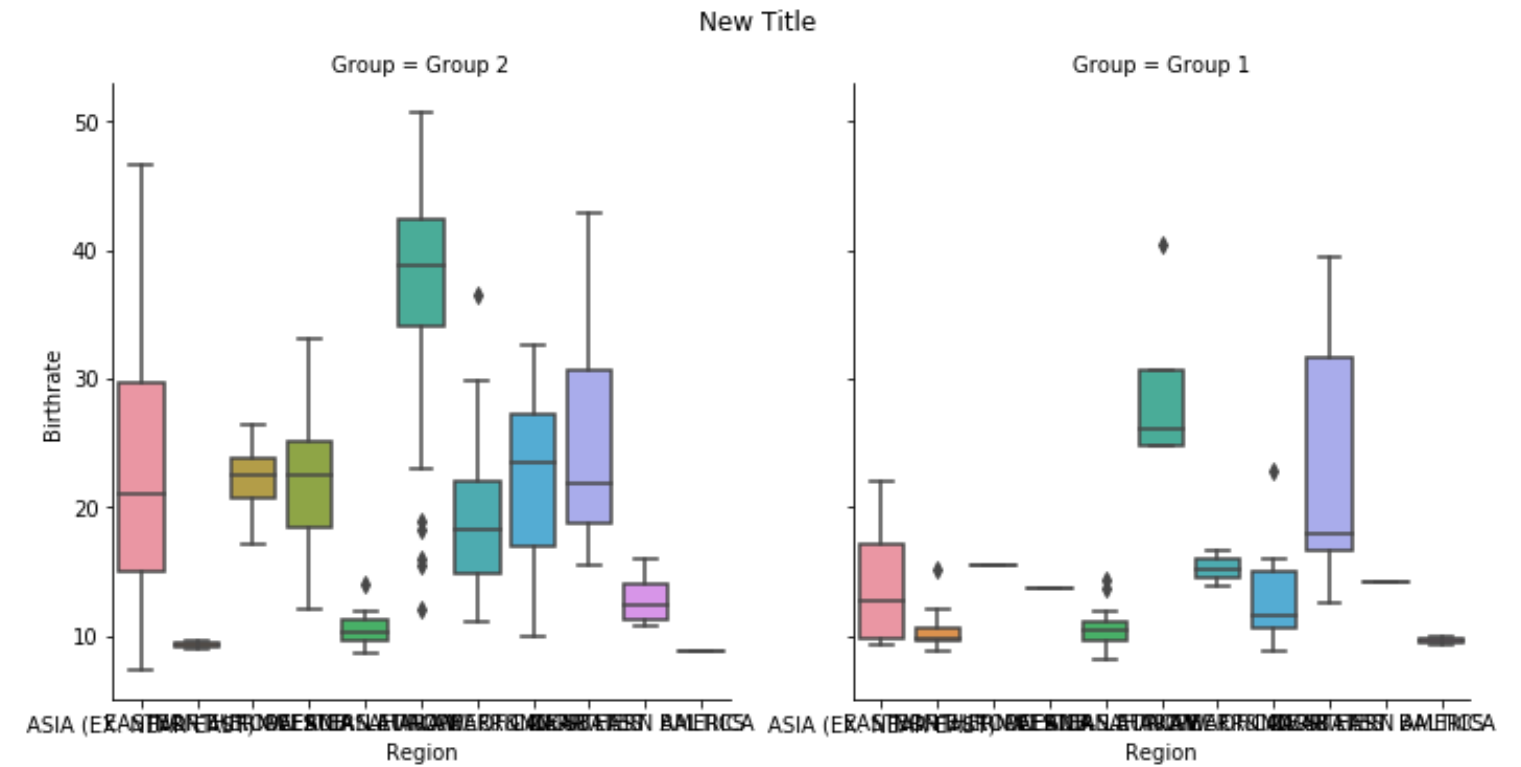
# Titles for subplots

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box",  
                col="Group")
```



# Titles for subplots

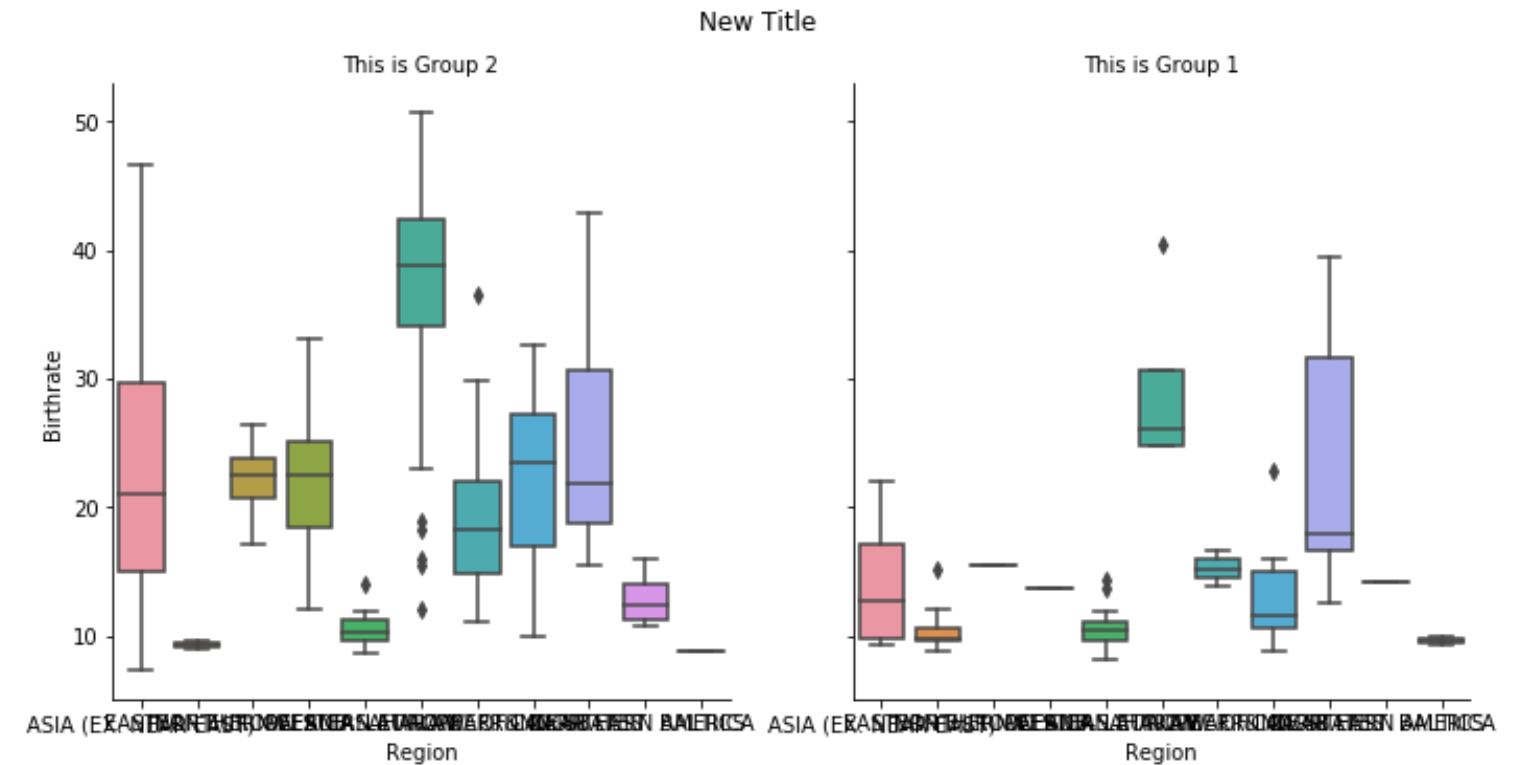
```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box",  
                col="Group")  
  
g.fig.suptitle("New Title",  
               y=1.03)
```





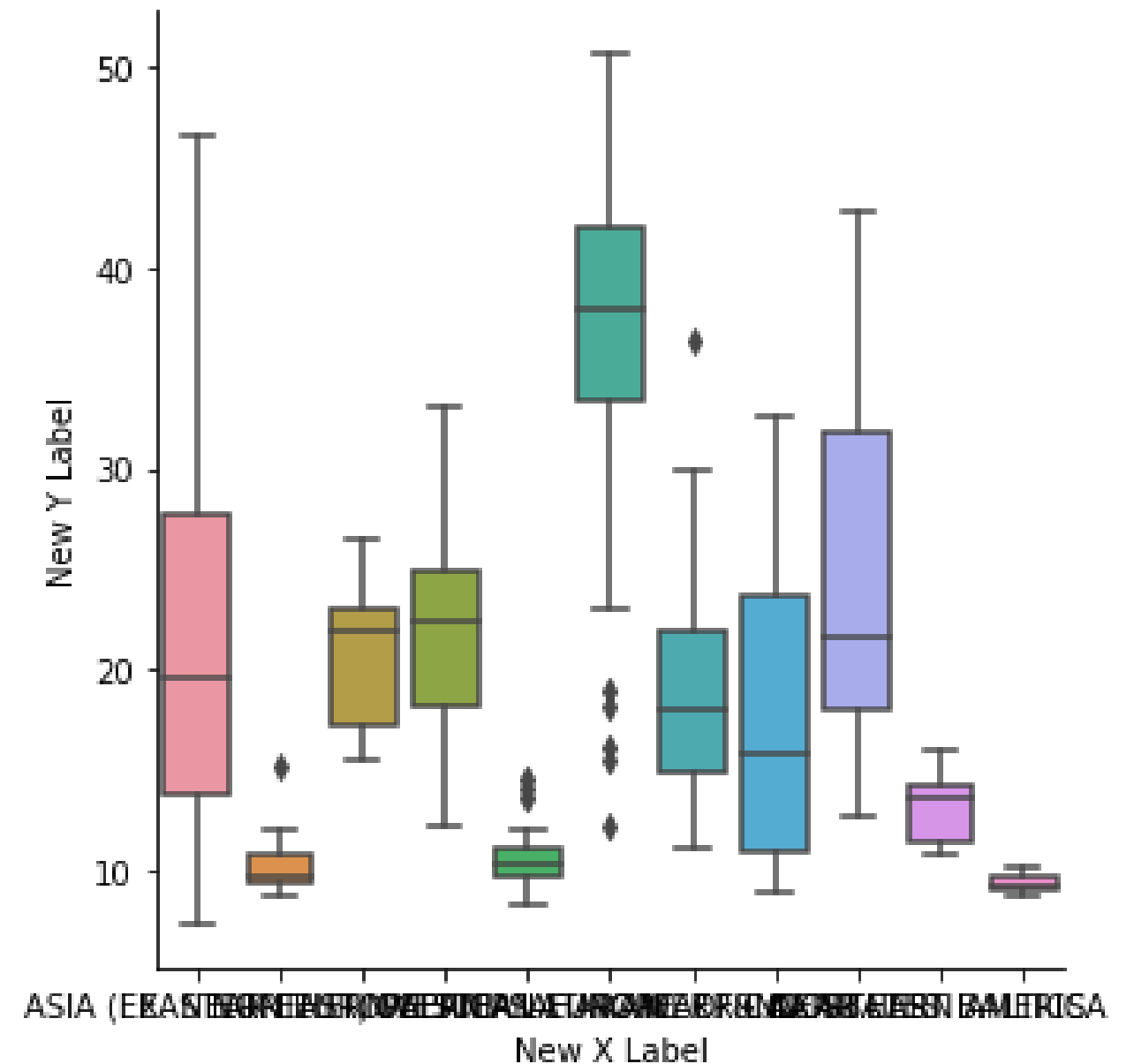
# Titles for subplots

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box",  
                col="Group")  
  
g.fig.suptitle("New Title",  
              y=1.03)  
  
g.set_titles("This is {col_name}")
```



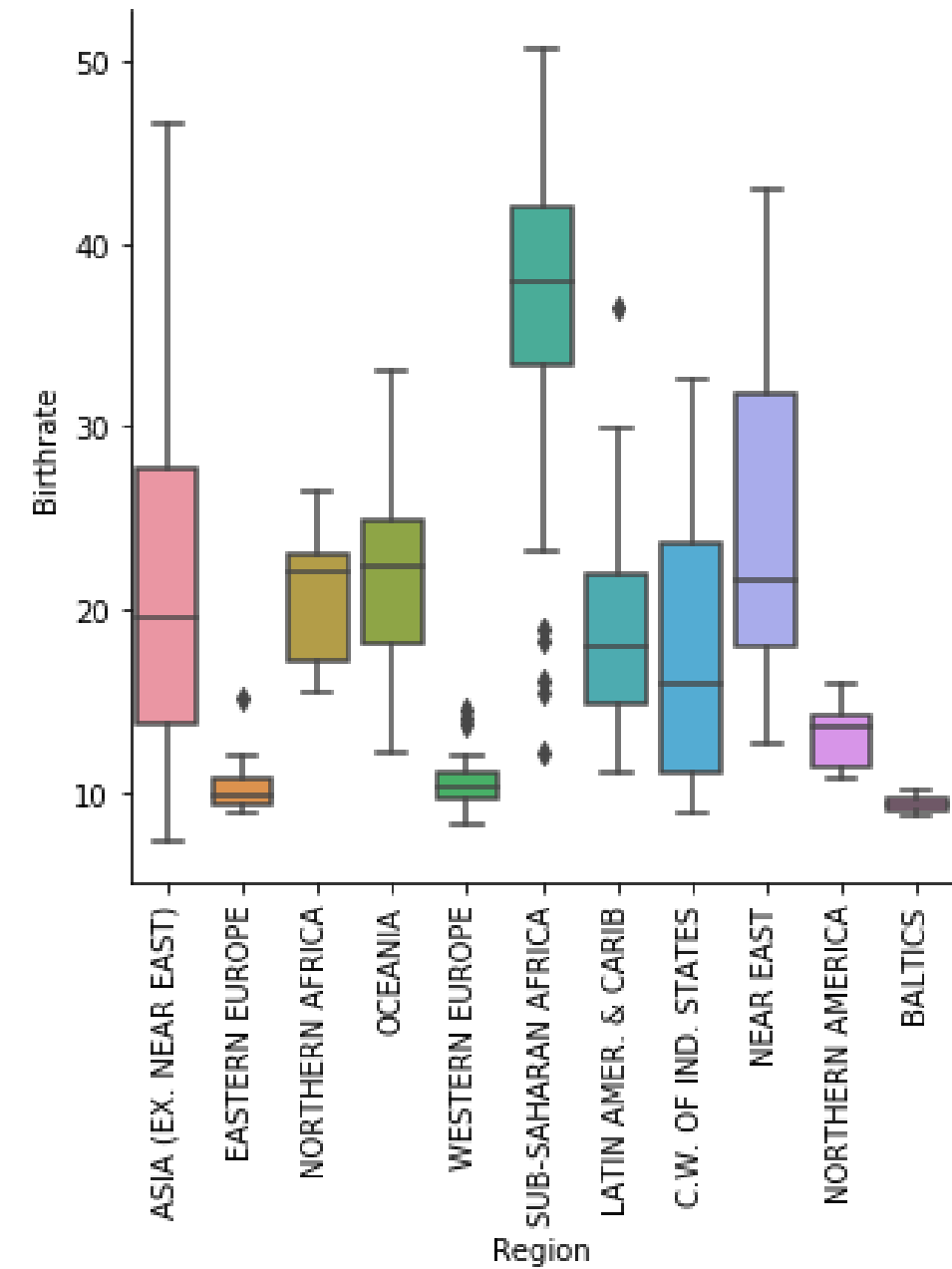
# Adding axis labels

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")  
  
g.set(xlabel="New X Label",  
      ylabel="New Y Label")  
  
plt.show()
```



# Rotating x-axis tick labels

```
g = sns.catplot(x="Region",  
                y="Birthrate",  
                data=gdp_data,  
                kind="box")  
  
plt.xticks(rotation=90)  
plt.show()
```



# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

# Putting it all together

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



**Erin Case**  
Data Scientist

# Getting started

To import Seaborn:

```
import seaborn as sns
```

To import Matplotlib:

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

To show a plot:

```
plt.show()
```

# Relational plots

- Show the relationship between two quantitative variables
- Examples: scatter plots, line plots

```
sns.relplot(x="x_variable_name",  
            y="y_variable_name",  
            data=pandas_df,  
            kind="scatter")
```

# Categorical plots

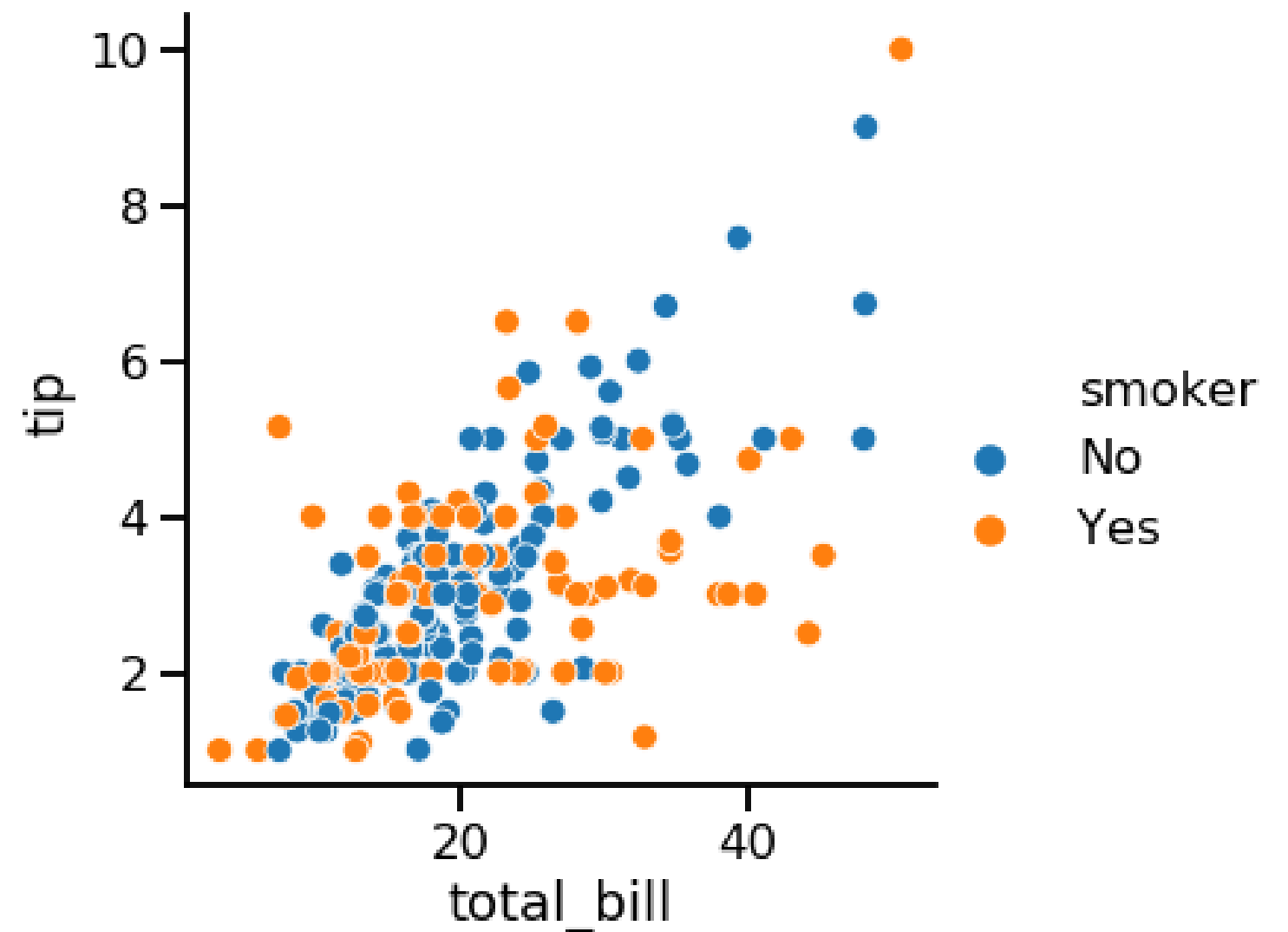
- Show the distribution of a quantitative variable within categories defined by a categorical variable
- Examples: bar plots, count plots, box plots, point plots

```
sns.catplot(x="x_variable_name",  
            y="y_variable_name",  
            data=pandas_df,  
            kind="bar")
```



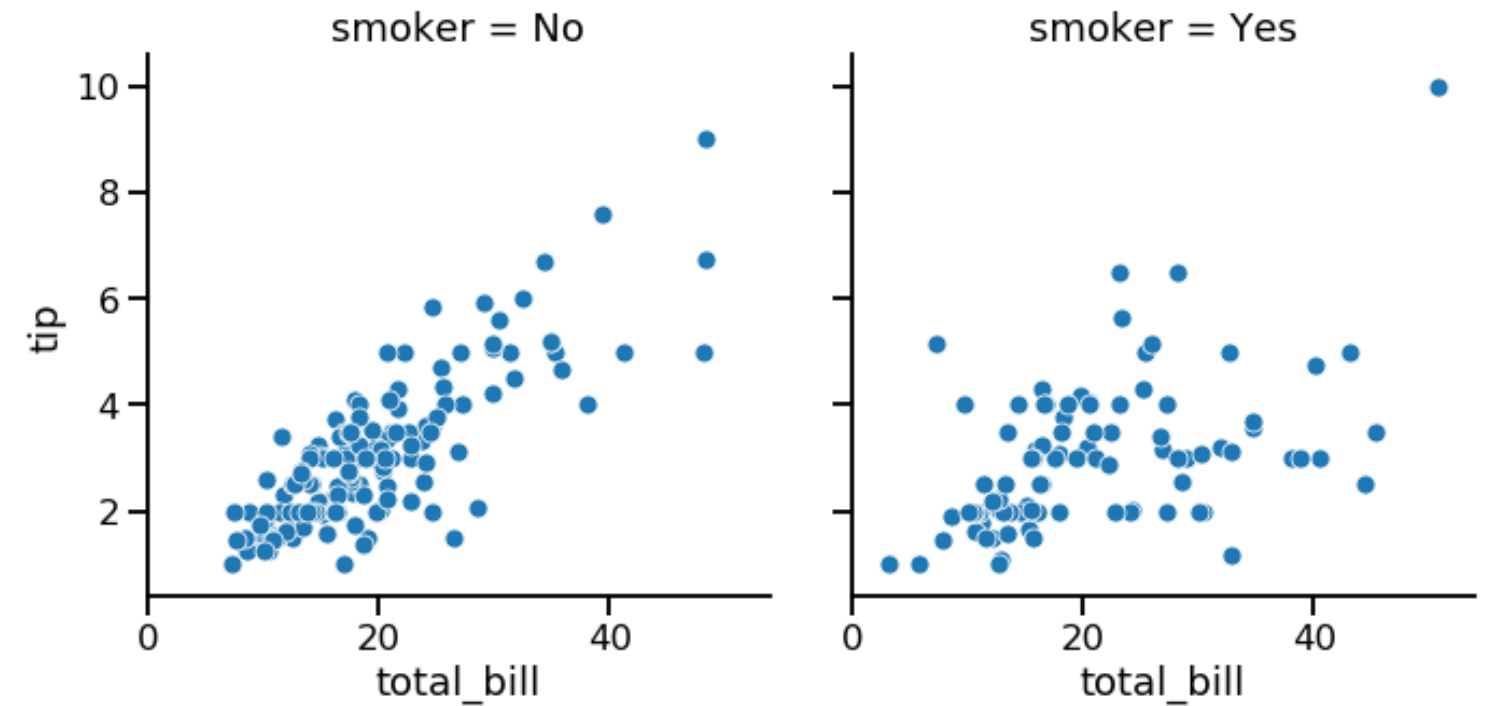
# Adding a third variable (hue)

Setting `hue` will create subgroups that are displayed as different colors on a single plot.



# Adding a third variable (row/col)

Setting `row` and/or `col` in `relplot()` or `catplot()` will create subgroups that are displayed on separate subplots.



# Customization

- Change the background: `sns.set_style()`
- Change the main element colors: `sns.set_palette()`
- Change the scale: `sns.set_context()`

# Adding a title

| Object Type | Plot Types                         | How to Add Title |
|-------------|------------------------------------|------------------|
| FacetGrid   | relplot() , catplot()              | g.fig.suptitle() |
| AxesSubplot | scatterplot() , countplot() , etc. | g.set_title()    |

# Final touches

Add x- and y-axis labels:

```
g.set(xlabel="new x-axis label",  
      ylabel="new y-axis label")
```

Rotate x-tick labels:

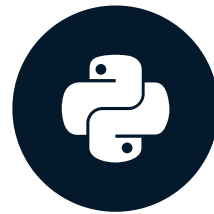
```
plt.xticks(rotation=90)
```

# Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN

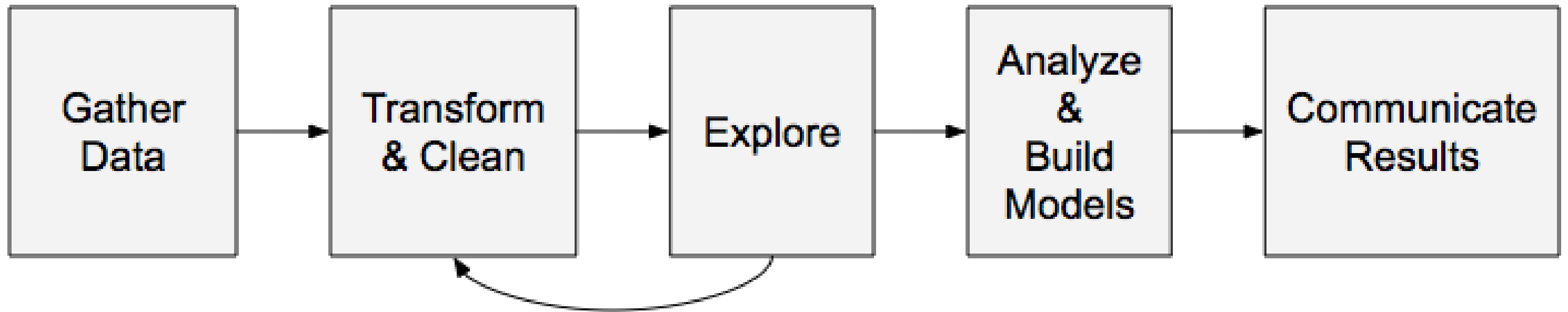
# Well done! What's next?

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN



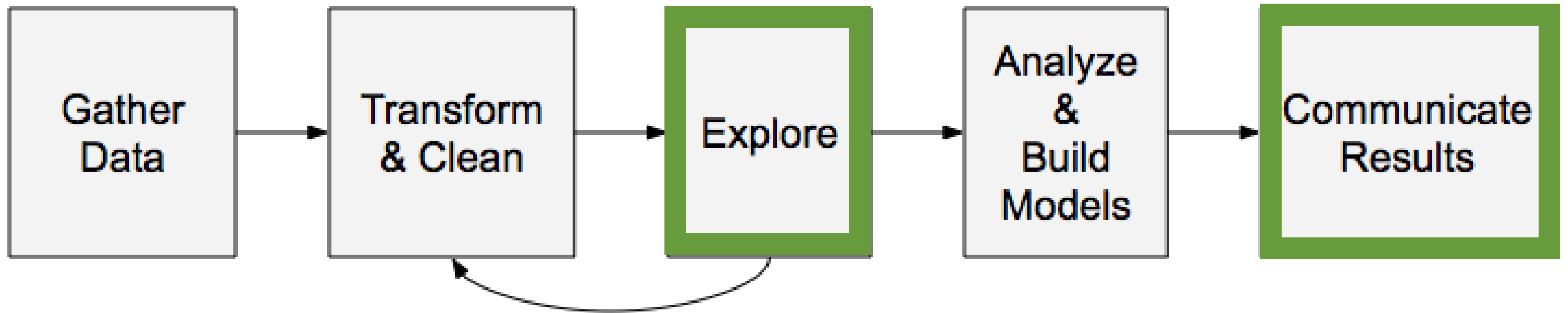
**Erin Case**  
Data Scientist

# Where does Seaborn fit in?

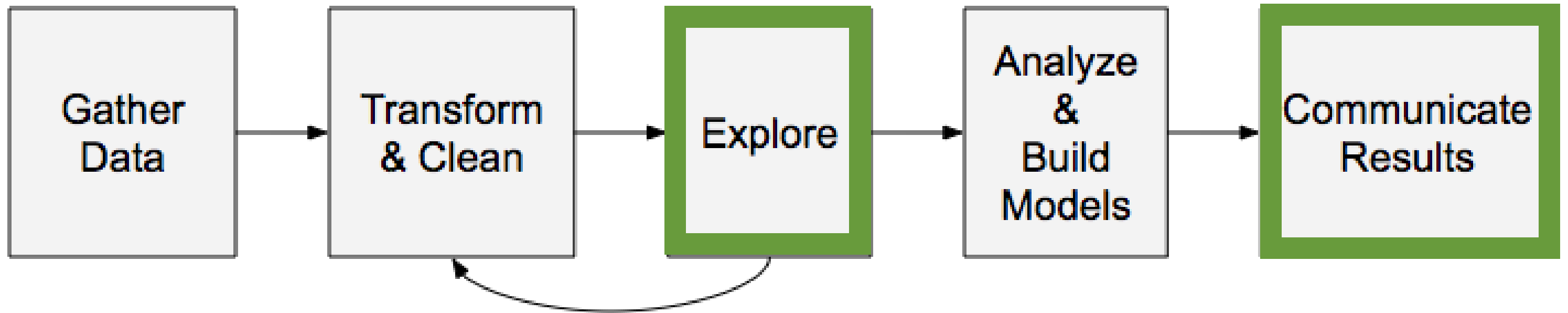




# Where does Seaborn fit in?



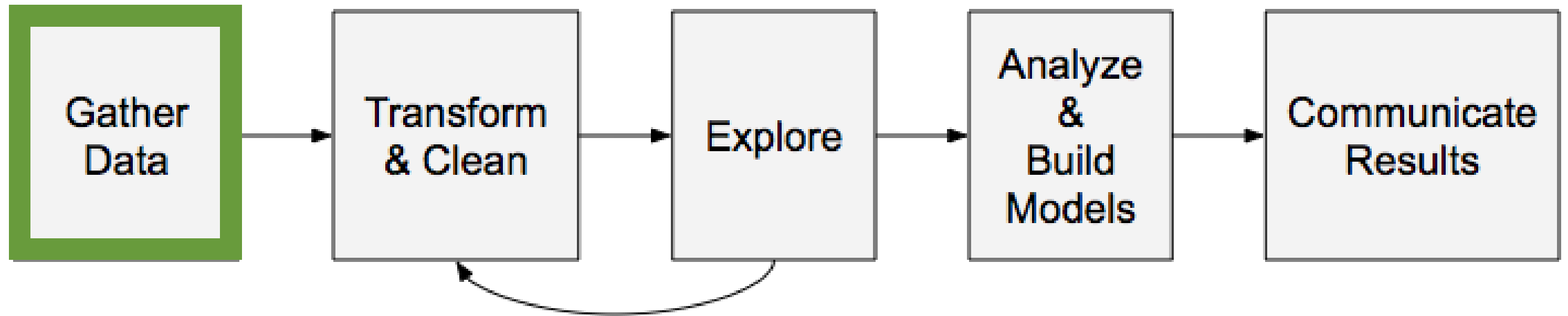
# Next Steps: Explore and communicate results



Next steps:

- Seaborn advanced visualizations
- Matplotlib advanced customizations

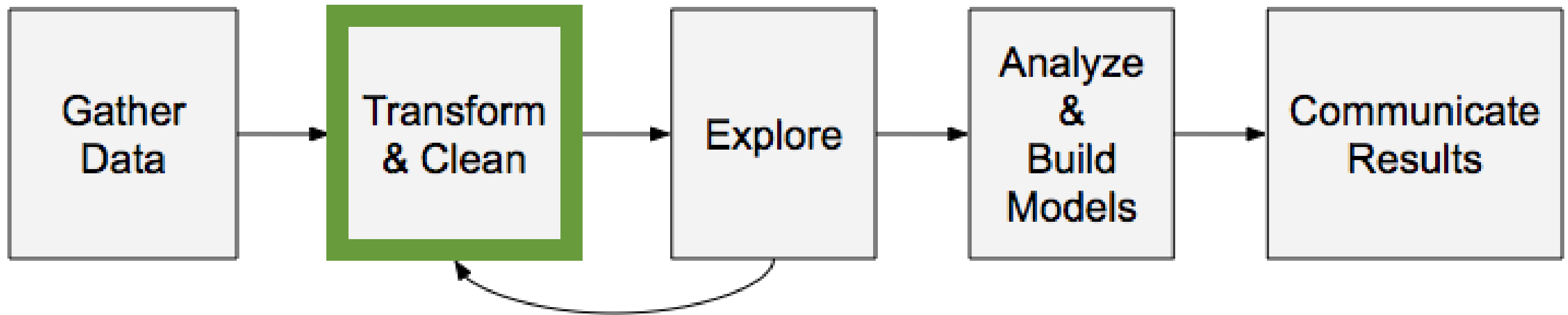
# Next steps: Gather data



Next steps:

- Python
- SQL

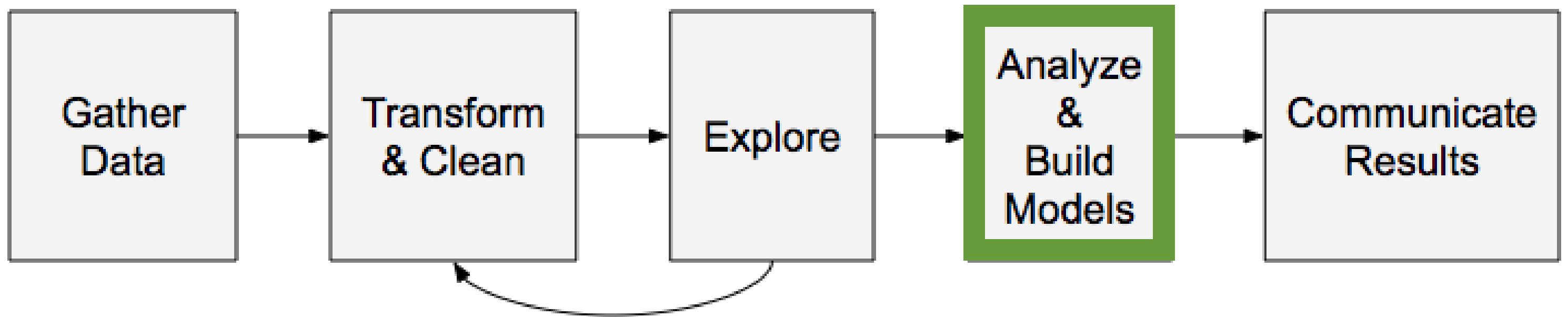
# Next steps: Transform and clean



Next steps:

- Getting data into Pandas DataFrames
- Cleaning data
- Transforming into tidy format

# Next steps: Analyze and build models



Next steps:

- Statistical analysis
- Calculating and interpreting confidence intervals

# Congratulations!

INTRODUCTION TO DATA VISUALIZATION WITH SEABORN