Python for Data Analysis and Visualization

Instructor: Claudia Carroll
Spring 2024

Session 5 (April 8)

Transdisciplinary Institute *in* Applied Data Sciences (TRIADS)

Arts & Sciences at Washington University in St. Louis
Signature Initiative

Today's Lesson Plan

1. Demo One: Basic Plotting with Pandas, Matplotlib and Seaborn

2. Demo 2: Plot Customization

Matplotlib vs. Seaborn

- Both are python libraries for data visualization
- Matplotlib is more customizable, and is therefore more suitable for more advanced programmers with detailed intentions for their graphs
- Seaborn automates many visualization parameters, and offers an automated linear regression function. Seaborn is more suitable for generating fast, visually appealing visualizations with few customizations.

Setup

- Go to my GitHub repository for this class:

  https://github.com/ClaudiaECarroll/python_data_class

- From the Class 5 folder, download the following file: "SAFI_full_shortname.csv"

- Put the file in your Desktop folder for this workshop.

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Demo 1: Visualizations with Pandas

Exercise 1:

1. Make a scatter plot of years_farm vs years_liv and color the points by buildings_in_compound

2. Make a bar plot of the mean number of rooms per wall type, where each wall type is represented by a different color

3. Use seaborn to generate a linear regression correlating livestock counts (liv_count) to the number of plots per farm (no_plots)

Exercise 1: Solution

1. **Make a scatter plot of years_farm vs years_liv and color the points by buildings_in_compound**

```
df.plot.scatter(x = 'years_liv', y = 'years_farm', c
= 'buildings_in_compound', colormap = 'viridis')
```

Exercise 1: Solution

2. **Make a bar plot of the mean number of rooms per wall type, where each wall type is represented by a different color bar.**

```
colors = ["pink", "red", "green", "orange"]
rooms_mean =
df.groupby('respondent_wall_type')['rooms'].mean()
rooms_mean.plot.bar(color=colors)
```

Exercise 1: Solution

3. Use seaborn to generate a linear regression correlating livestock counts (liv_count) to the number of plots per farm (no_plots)

```
sns.lmplot(x='no_plots', y='liv_count',data=df)
```

Transdisciplinary
Institute *in* Applied
Data Sciences (TRIADS)

Demo 2: Customizing Data

Exercise 2:

Using the dataset below, create a scatter plot to visualize the relationship between 'horsepower' and 'mpg' and save it to an image file. You should import all necessary libraries in the same cell, label the x-axis as 'Horsepower' and the y-axis as 'MPG', and title the plot as 'Horsepower vs MPG'. Change the default size and color of the dots to whatever you like.

data = { 'car': ['Toyota', 'Honda', 'Ford', 'Chevrolet', 'BMW', 'Tesla', 'Audi'], 'horsepower': [150, 120, 170, 200, 250, 300, 180], 'mpg': [30, 35, 25, 20, 18, 15, 22]

Hint: You will have to convert the dictionary to a dataframe. See if you can find this command online!

Exercise 2: Solution

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

data = {
    'car': ['Toyota', 'Honda', 'Ford', 'Chevrolet', 'BMW', 'Tesla',
'Audi'],
    'horsepower': [150, 120, 170, 200, 250, 300, 180],
    'mpg': [30, 35, 25, 20, 18, 15, 22]}

df2 = pd.DataFrame(data)

plt.scatter(df2['horsepower'], df2['mpg'], s=100, c='pink')
plt.xlabel('Horsepower')
plt.ylabel('MPG')
plt.title('Horsepower vs MPG')
plt.savefig('horsepower.png')
```