2024S2 - IT2313

Programming for Data Science

Data Visualisation with Seaborn

Introduction on Matplotlib and Seaborn

Matplotlib and Seaborn

Python's two most widely used data visualization libraries are Matplotlib and Seaborn. While both libraries are designed to create high-quality graphics and visualizations, they have several key differences that make them better suited for different use cases.

Matplotlib is a low-level plotting library that provides a wide range of tools for creating highly customizable visualizations. It is a highly flexible library, allowing users to create almost any type of plot they can imagine. This flexibility comes at the cost of a steeper learning curve and more verbose code.

Seaborn, on the other hand, is a high-level interface for creating statistical graphics. It is built on top of Matplotlib and provides a simpler, more intuitive interface for creating common statistical plots. Seaborn is designed to work with Pandas dataframes, making it easy to create visualizations with minimal code. It also offers a range of built-in statistical functions, allowing users to easily perform complex statistical analyses with their visualizations.

Basic Plots with Seaborn

Basic Plots with Seaborn

What is Seaborn?

Python data visualization library

Easy to create the most common types of plots.

Advantages of Seaborn

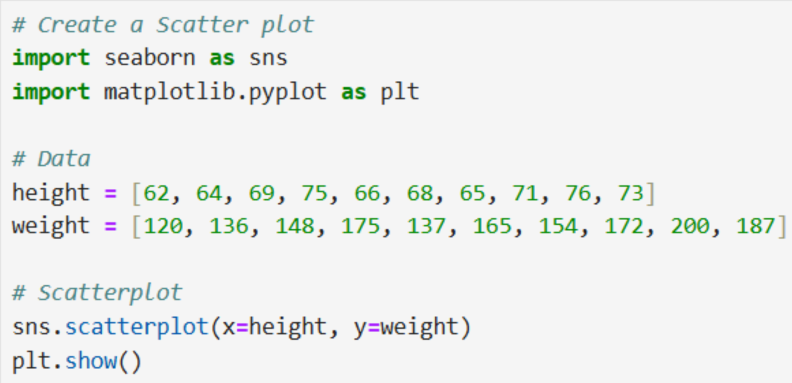
Easy to use

Seaborn supports complex visualizations of data

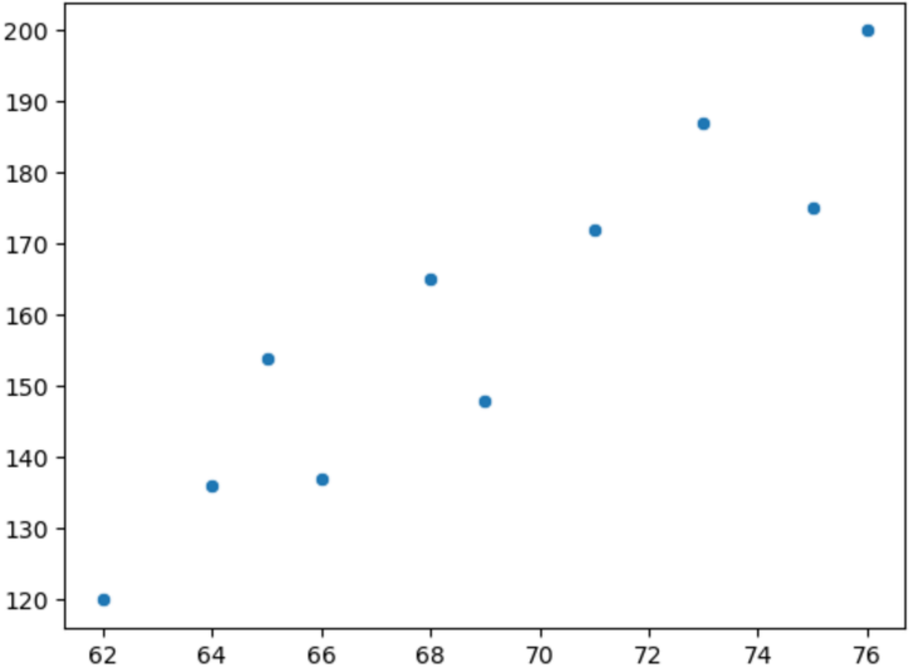
It is built on matplotlib and works best with pandas' dataframes

Basic Plots with Seaborn

Scatter Plot

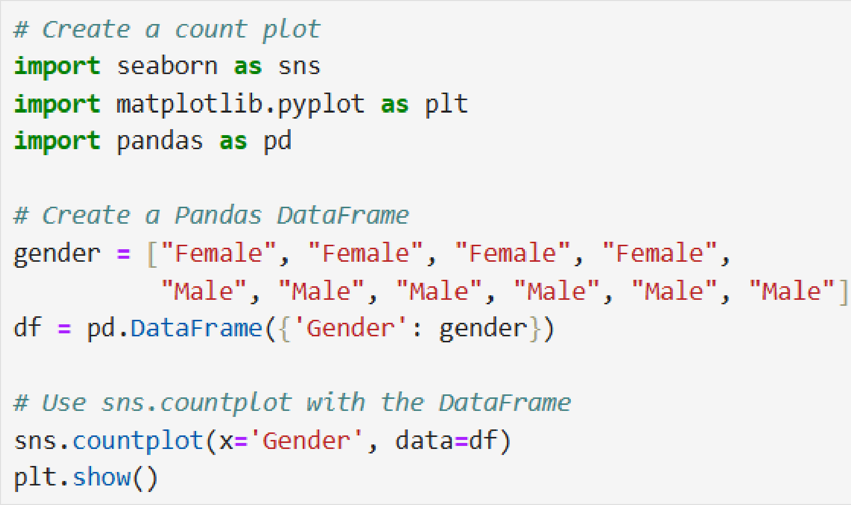


```
# Create a Scatter plot
import seaborn as sns
import matplotlib.pyplot as plt

# Data
height = [62, 64, 69, 75, 66, 68, 65, 71, 76, 73]
weight = [120, 136, 148, 175, 137, 165, 154, 172, 200, 187]

# Scatterplot
sns.scatterplot(x=height, y=weight)
plt.show()
```

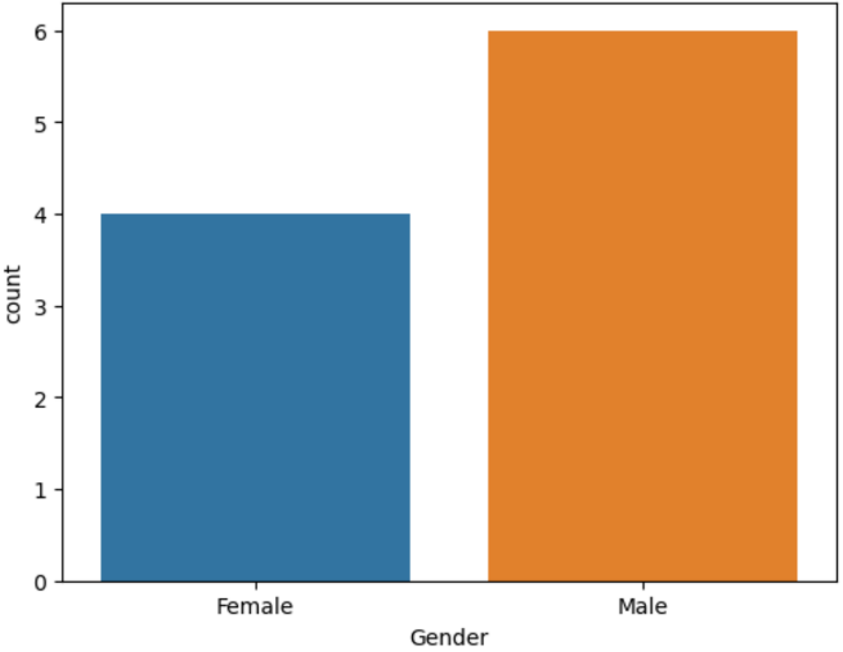


Basic Plots with Seaborn

Count Plot

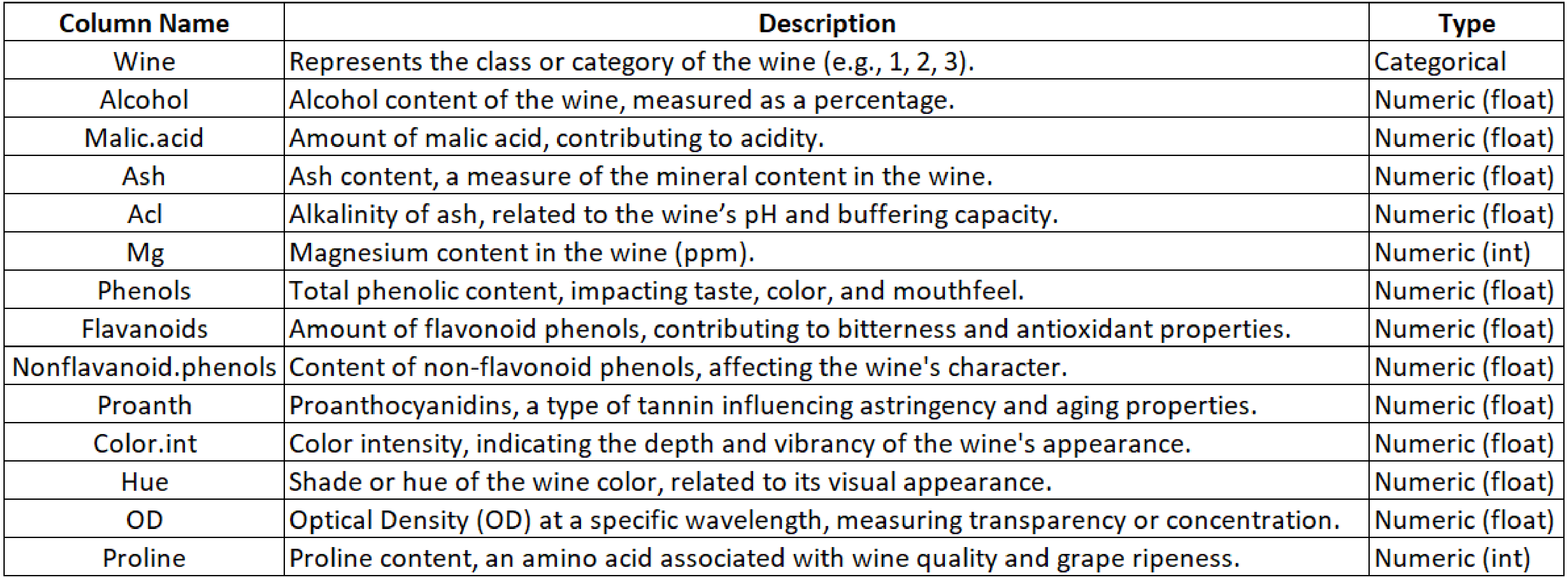


```python
# Create a count plot
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Create a Pandas DataFrame
gender = ["Female", "Female", "Female", "Female",
          "Male", "Male", "Male", "Male", "Male", "Male"]
df = pd.DataFrame({'Gender': gender})

# Use sns.countplot with the DataFrame
sns.countplot(x='Gender', data=df)
plt.show()
```



Basic Plots with Seaborn

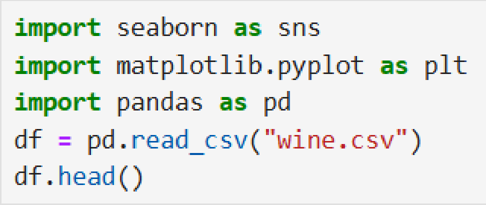
Let perform the basic EDA on the following wine.csv dataset. The features are as shown:



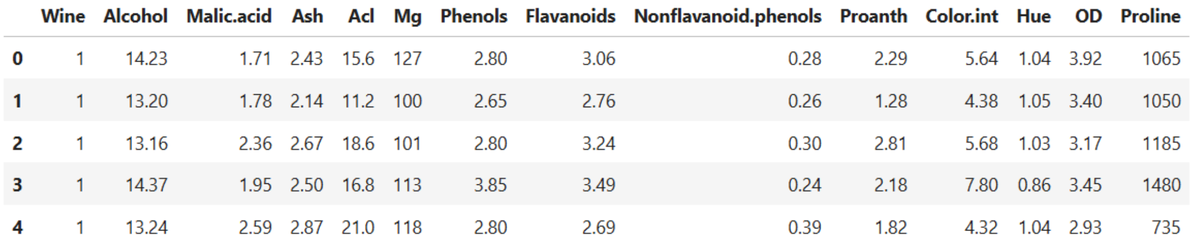
Column Name	Description	Type
Wine	Represents the class or category of the wine (e.g., 1, 2, 3).	Categorical
Alcohol	Alcohol content of the wine, measured as a percentage.	Numeric (float)
Malic.acid	Amount of malic acid, contributing to acidity.	Numeric (float)
Ash	Ash content, a measure of the mineral content in the wine.	Numeric (float)
Acl	Alkalinity of ash, related to the wine’s pH and buffering capacity.	Numeric (float)
Mg	Magnesium content in the wine (ppm).	Numeric (int)
Phenols	Total phenolic content, impacting taste, color, and mouthfeel.	Numeric (float)
Flavanoids	Amount of flavonoid phenols, contributing to bitterness and antioxidant properties.	Numeric (float)
Nonflavanoid.phenols	Content of non-flavonoid phenols, affecting the wine's character.	Numeric (float)
Proanth	Proanthocyanidins, a type of tannin influencing astringency and aging properties.	Numeric (float)
Color.int	Color intensity, indicating the depth and vibrancy of the wine's appearance.	Numeric (float)
Hue	Shade or hue of the wine color, related to its visual appearance.	Numeric (float)
OD	Optical Density (OD) at a specific wavelength, measuring transparency or concentration.	Numeric (float)
Proline	Proline content, an amino acid associated with wine quality and grape ripeness.	Numeric (int)

Basic Plots with Seaborn

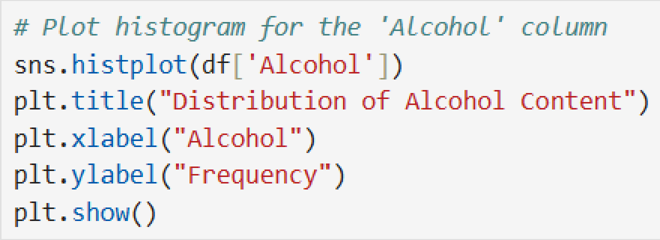
Histogram



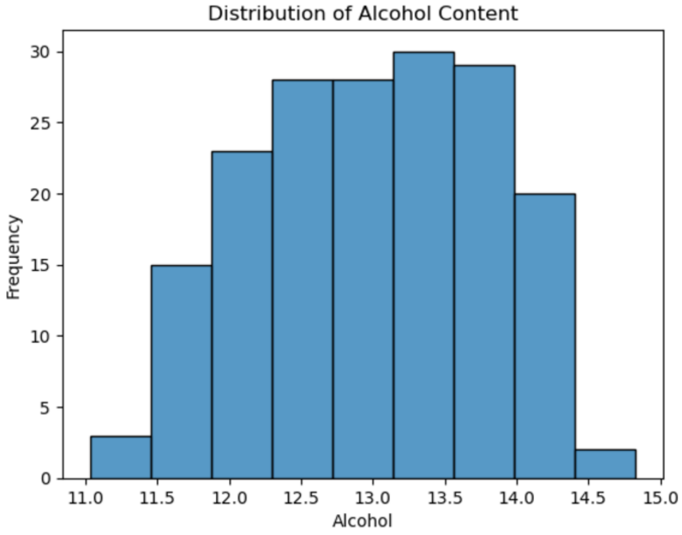
```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv("wine.csv")
df.head()
```



	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Proanth	Color.int	Hue	OD	Proline
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

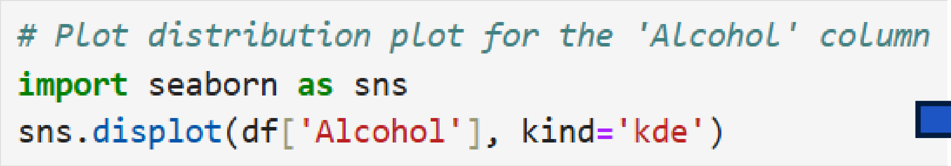


```python
# Plot histogram for the 'Alcohol' column
sns.histplot(df['Alcohol'])
plt.title("Distribution of Alcohol Content")
plt.xlabel("Alcohol")
plt.ylabel("Frequency")
plt.show()
```

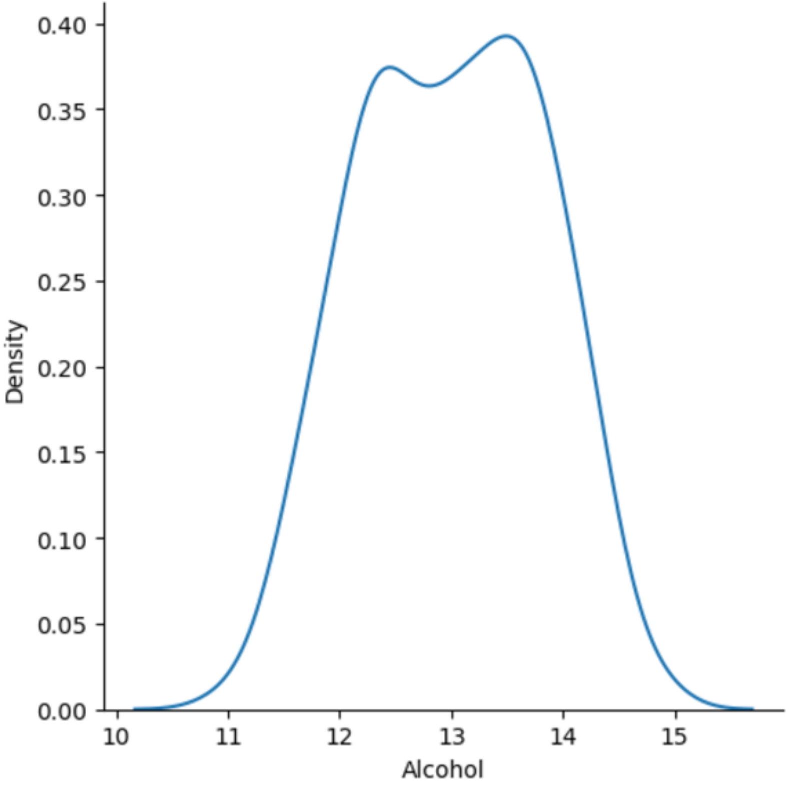


Basic Plots with Seaborn

Distribution Plot

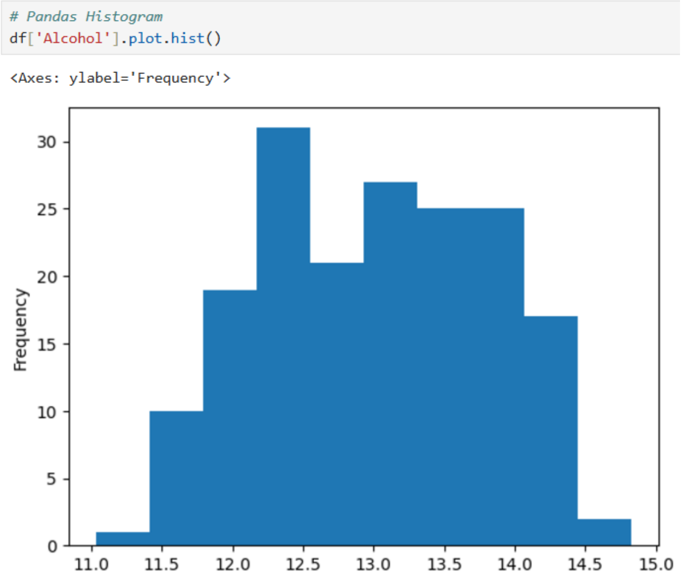


```python
# Plot distribution plot for the 'Alcohol' column
import seaborn as sns
sns.displot(df['Alcohol'], kind='kde')
```



Basic Plots with Seaborn

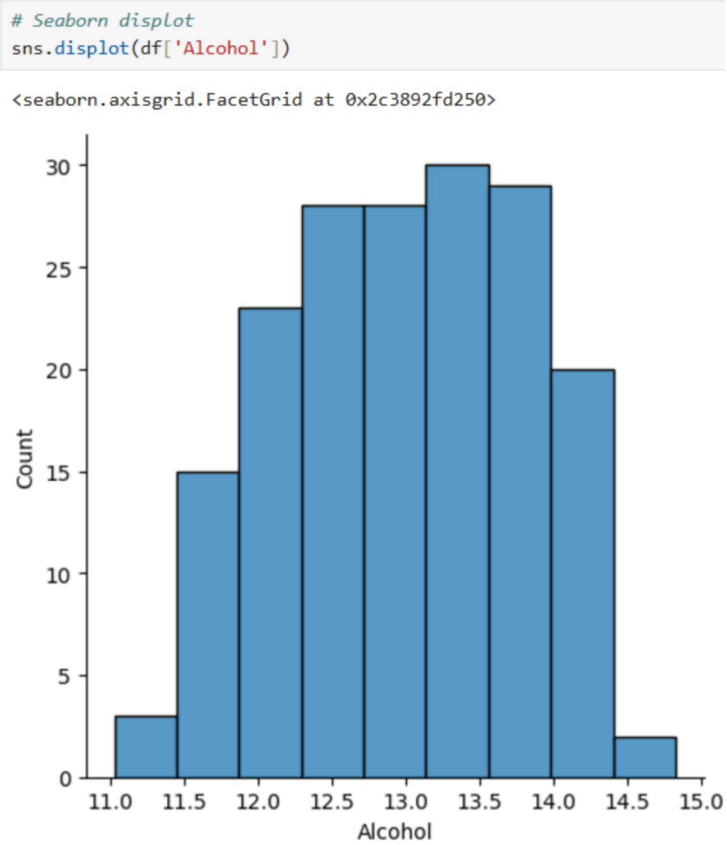
Pandas Histogram vs. Displot



```
# Pandas Histogram
df['Alcohol'].plot.hist()
```

```
<Axes: ylabel='Frequency'>
```

- Actual frequency of observations
- No outline of bars
- Wide bins



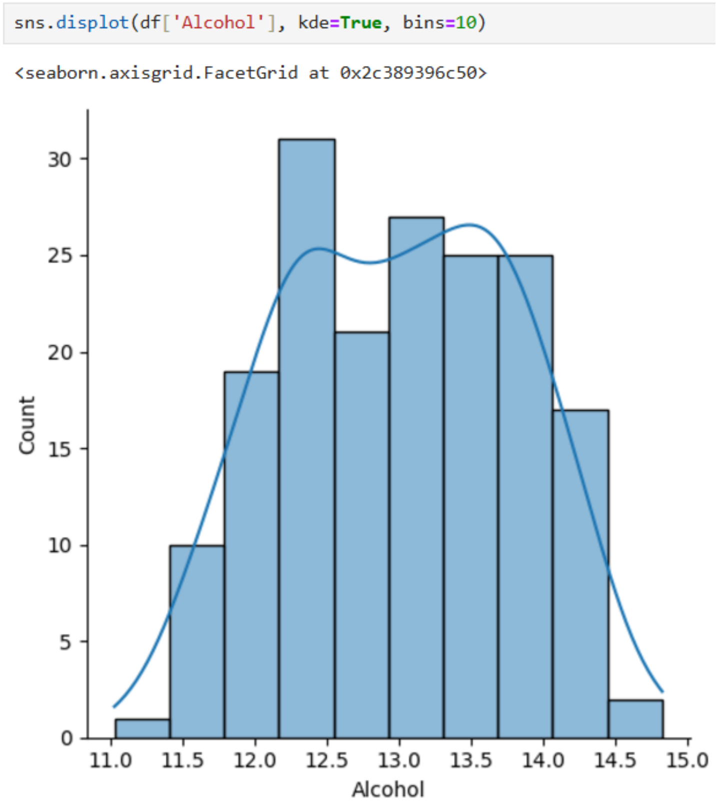
```
# Seaborn displot
sns.displot(df['Alcohol'])
```

```
<seaborn.axisgrid.FacetGrid at 0x2c3892fd250>
```

- Automatic label on x-axis
- Muted color palette
- Cleaner plot

Basic Plots with Seaborn

Creating a Histogram



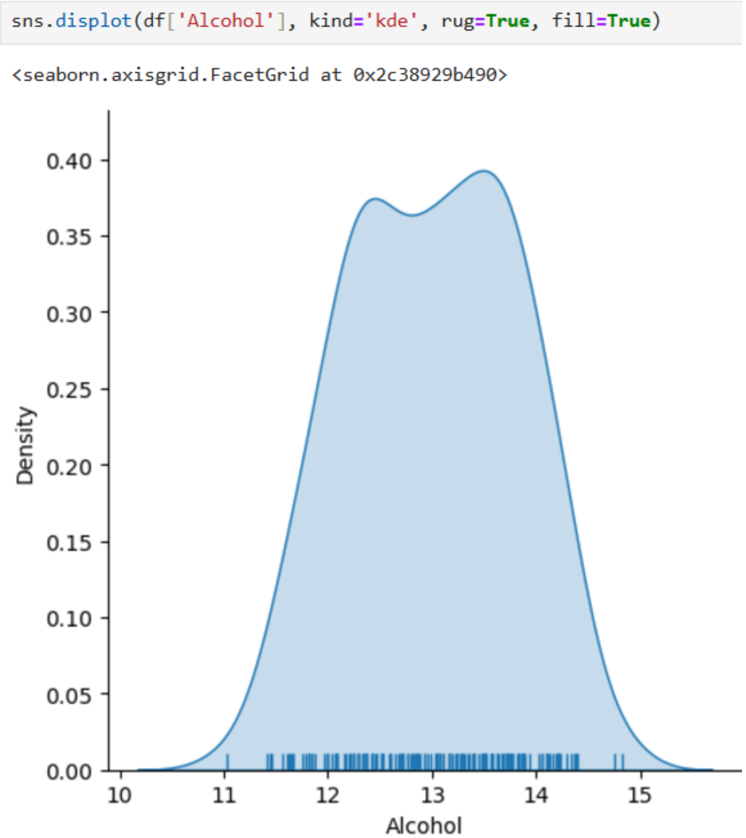
```
sns.displot(df['Alcohol'], kde=True, bins=10)
```

```
<seaborn.axisgrid.FacetGrid at 0x2c389396c50>
```

- The displot function has multiple optional arguments
- You can overlay a KDE plot on the histogram and specify the number of bins to use

Basic Plots with Seaborn

Alternative data distributions



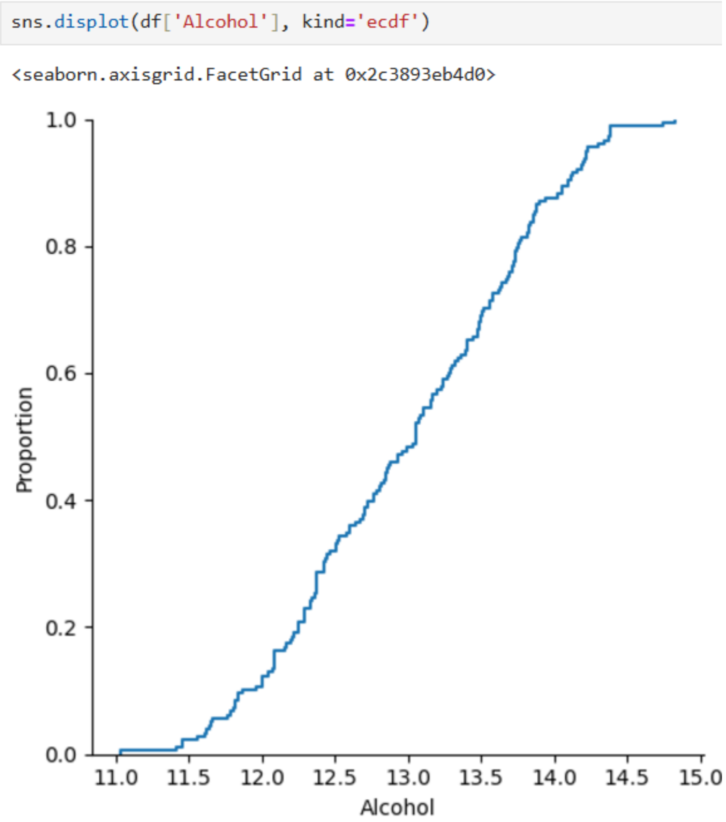
```
sns.displot(df['Alcohol'], kind='kde', rug=True, fill=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x2c38929b490>
```

- The displot function has multiple optional arguments
- You can overlay a KDE plot on the histogram and specify the number of bins to use

Basic Plots with Seaborn

Further plot types



```
sns.displot(df['Alcohol'], kind='ecdf')
```

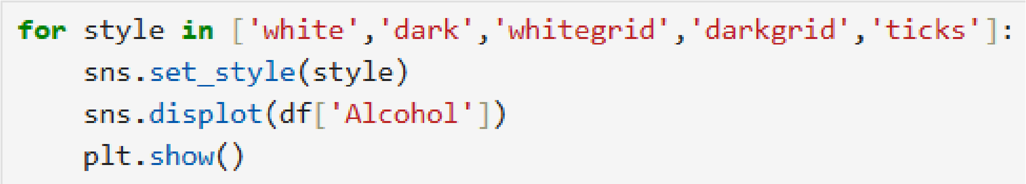
```
<seaborn.axisgrid.FacetGrid at 0x2c3893eb4d0>
```

- The displot function uses several functions including kdeplot , rugplot and ecdfplot
- The ecdfplot shows the cumulative distribution of the data

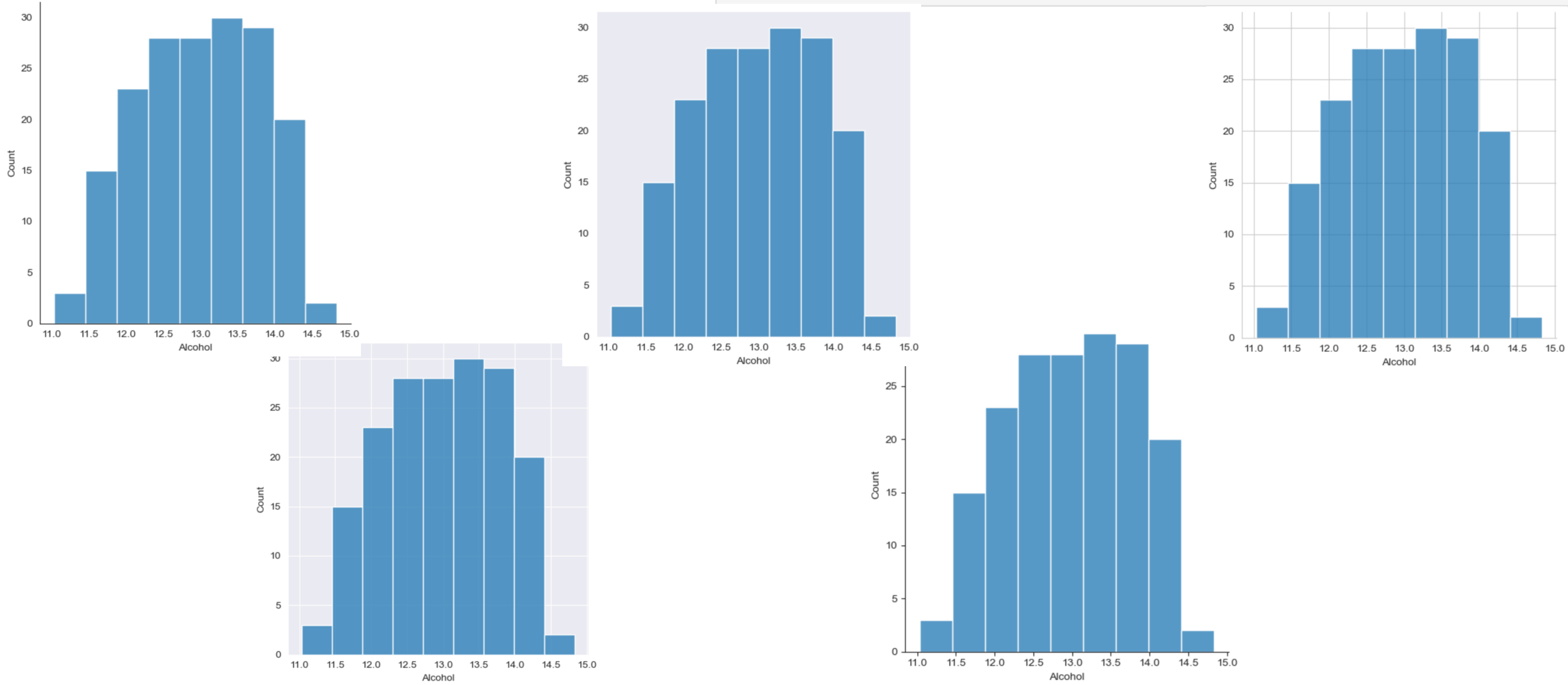
Using Seaborn Styles

Using Seaborn Styles

Theme examples with sns.set_style()

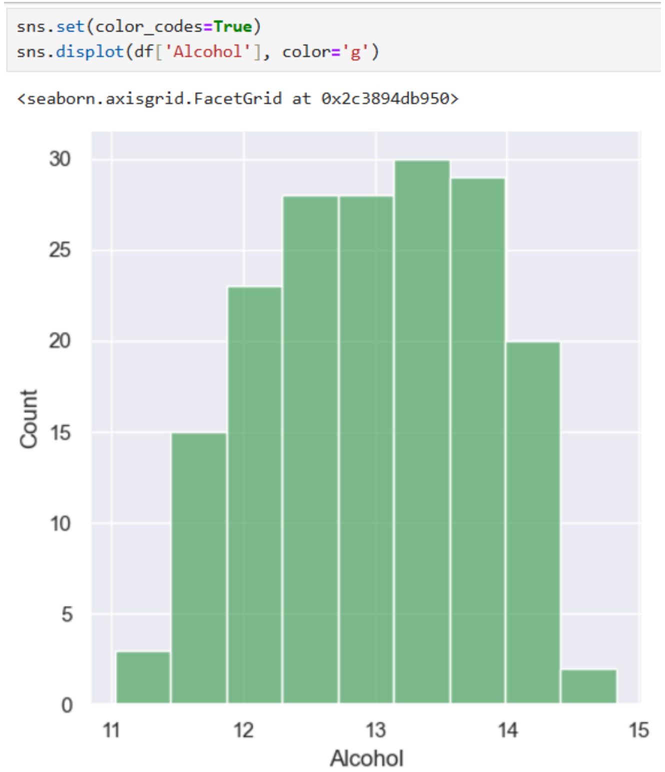


```python
for style in ['white','dark','whitegrid','darkgrid','ticks']:
    sns.set_style(style)
    sns.displot(df['Alcohol'])
    plt.show()
```



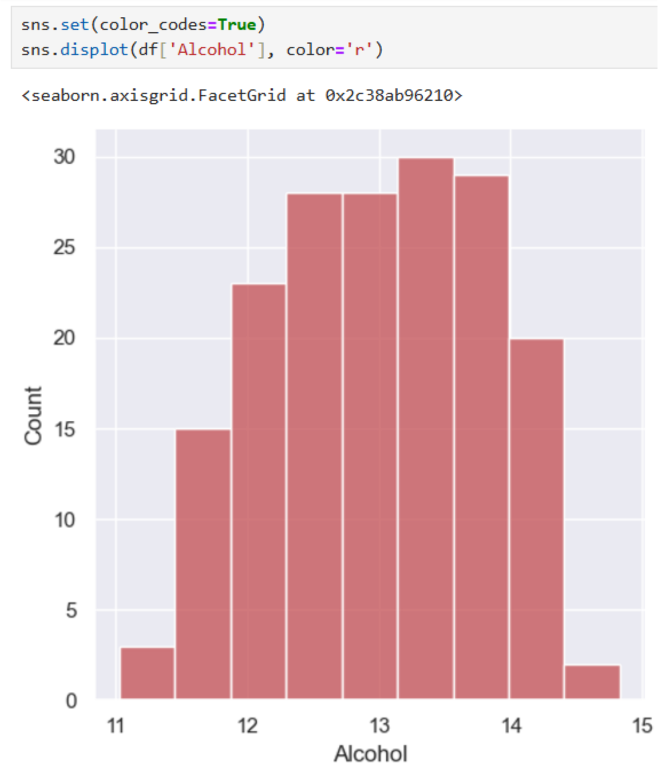
Defining a color for a plot

Seaborn supports assigning colors to plots using matplotlib color codes



```
sns.set(color_codes=True)
sns.displot(df['Alcohol'], color='g')
```

```
<seaborn.axisgrid.FacetGrid at 0x2c3894db950>
```



```
sns.set(color_codes=True)
sns.displot(df['Alcohol'], color='r')
```

```
<seaborn.axisgrid.FacetGrid at 0x2c38ab96210>
```

Defining a color for a plot

Seaborn supports assigning colors to plots using matplotlib color codes

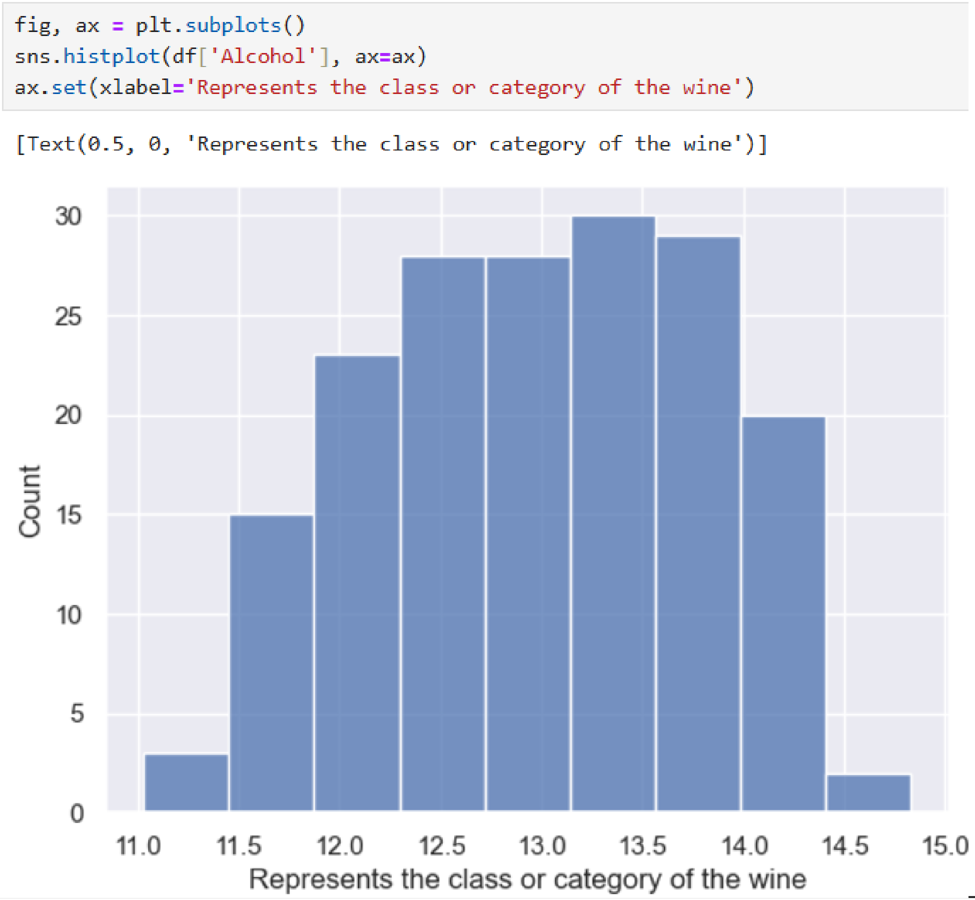
Named Colors

Seaborn supports all named colors recognized by Matplotlib. Some examples include:'r' (red), 'g' (green), 'b' (blue), 'k' (black), 'w' (white)Full names like 'red', 'blue', 'green', 'yellow’

Hexadecimal Codes

You can use hexadecimal color codes for precise colors:Examples: '#FF5733', '#4287f5', '#00FF00'

Axes Naming

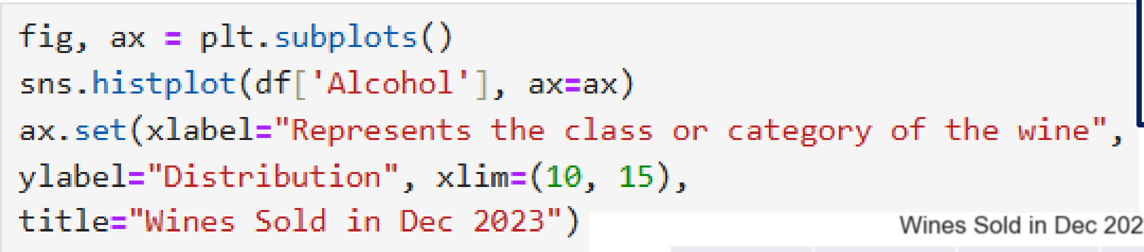


```
fig, ax = plt.subplots()
sns.histplot(df['Alcohol'], ax=ax)
ax.set(xlabel='Represents the class or category of the wine')
```

```
[Text(0.5, 0, 'Represents the class or category of the wine')]
```

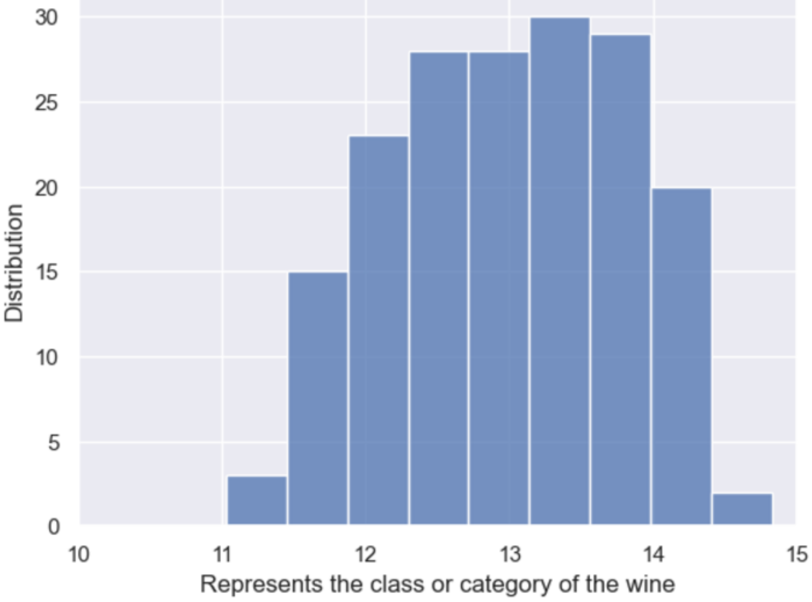
- Most customization available through matplotlib Axes objects
- Axes can be passed to seaborn functions

Axes Naming



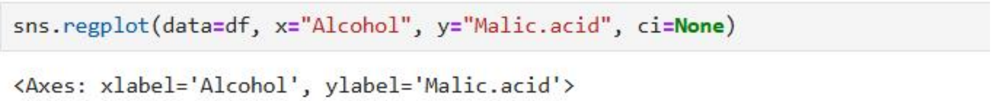
```
fig, ax = plt.subplots()
sns.histplot(df['Alcohol'], ax=ax)
ax.set(xlabel="Represents the class or category of the wine",
ylabel="Distribution", xlim=(10, 15),
title="Wines Sold in Dec 2023")
```

The axes object supports many common customizations



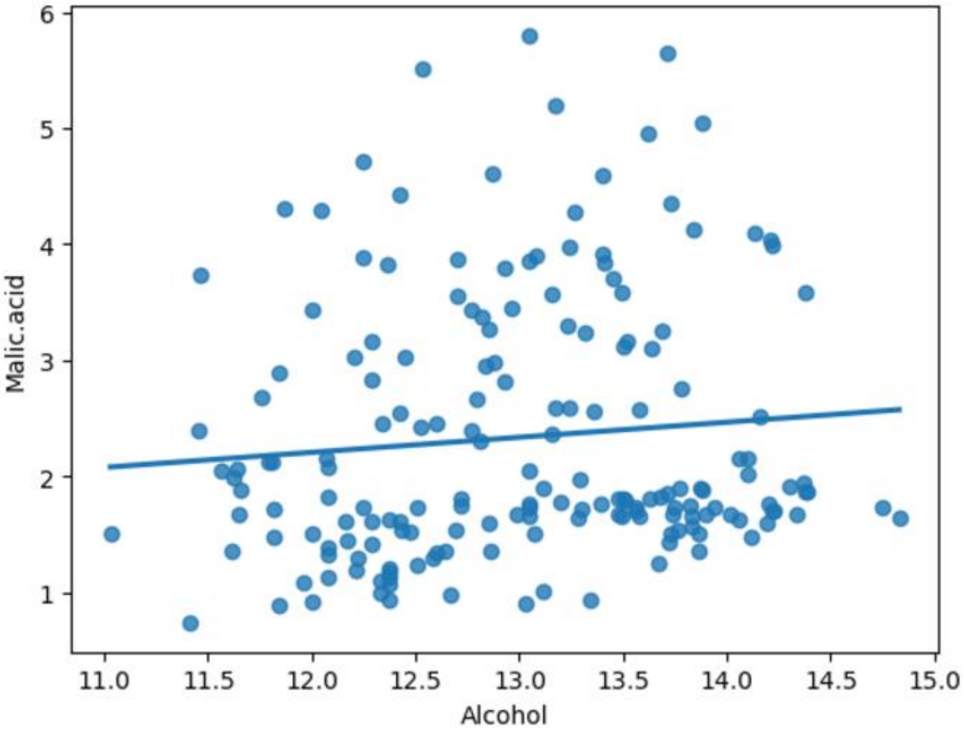
Regression Plots in Seaborn

Regression Plots in Seaborn



```
sns.regplot(data=df, x="Alcohol", y="Malic.acid", ci=None)
```

```
<Axes: xlabel='Alcohol', ylabel='Malic.acid'>
```



- The regplot function generates a scatter plot with a regression line
- Usage is similar to the displot
- The data and x and y variables must be defined

Regression Plot - Bicycle Dataset

Aggregated bicycle sharing data in Washington DC

Data includes:

Rental amounts

Weather information

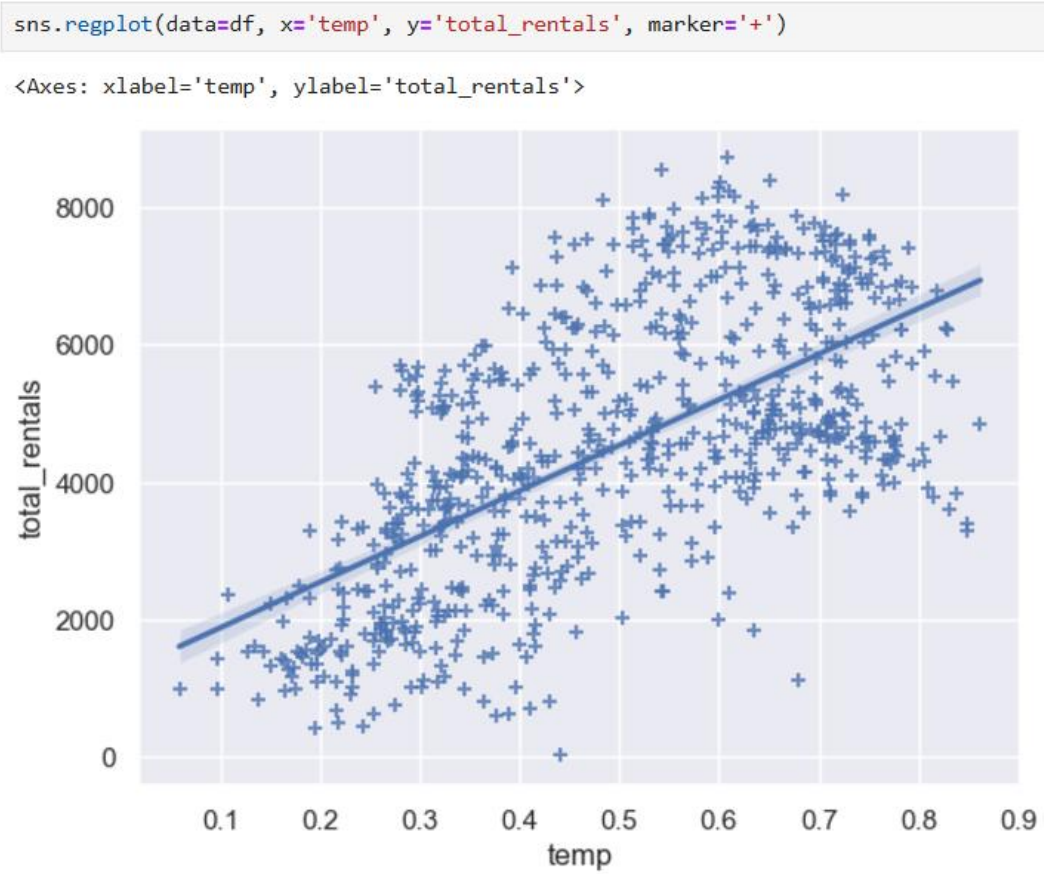
Calendar information

Can we predict rental amounts?

- Most customization available through matplotlib Axes objects
- Axes can be passed to seaborn functions

Regression Plot - Bicycle Dataset

Plotting with regplot()



```
sns.regplot(data=df, x='temp', y='total_rentals', marker='+')
```

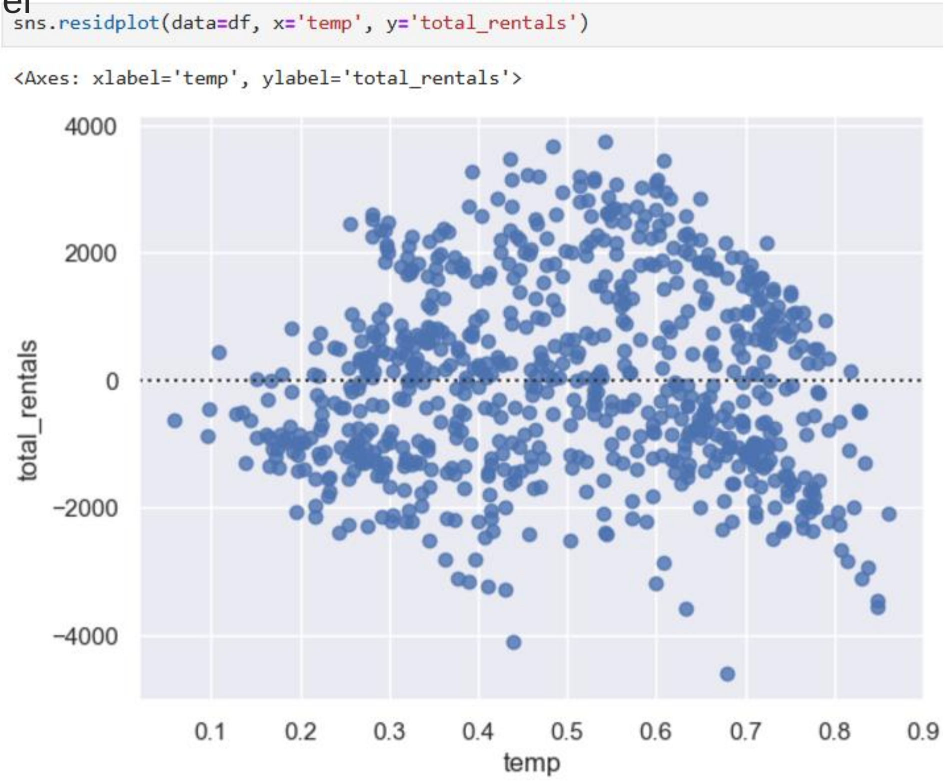
```
<Axes: xlabel='temp', ylabel='total_rentals'>
```

Regression Plot - Bicycle Dataset

Evaluating regression with residplot()

A residual plot is useful for evaluating the fit of a model

Seaborn supports through residplot function

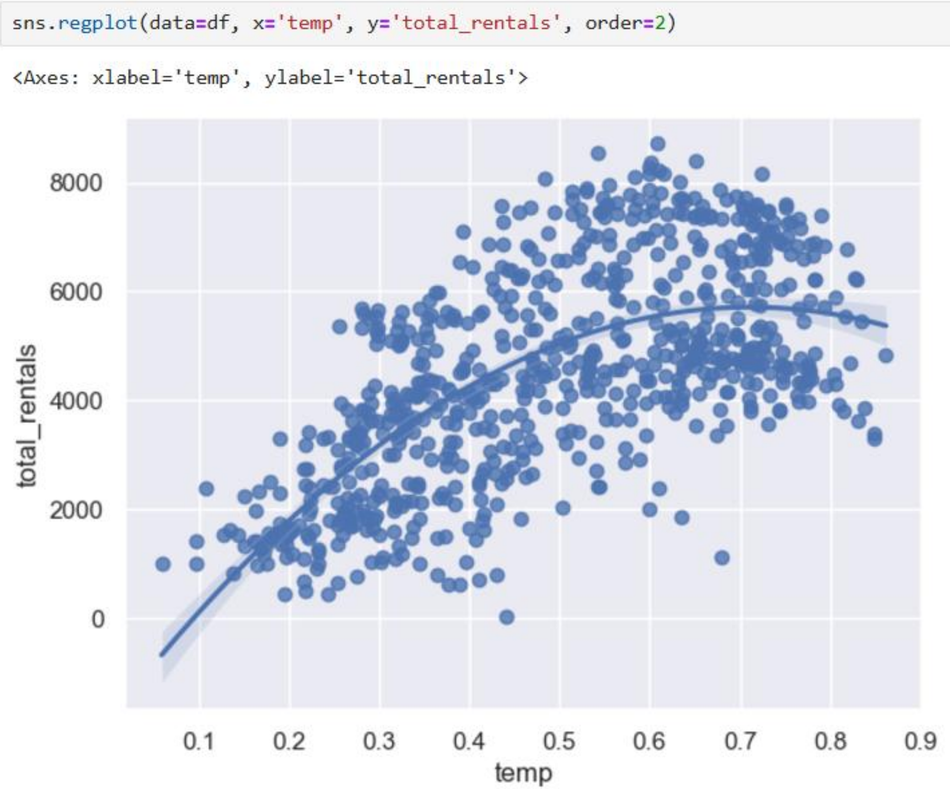


```
sns.residplot(data=df, x='temp', y='total_rentals')
```

```
<Axes: xlabel='temp', ylabel='total_rentals'>
```

Regression Plot - Bicycle Dataset

Polynomial regression



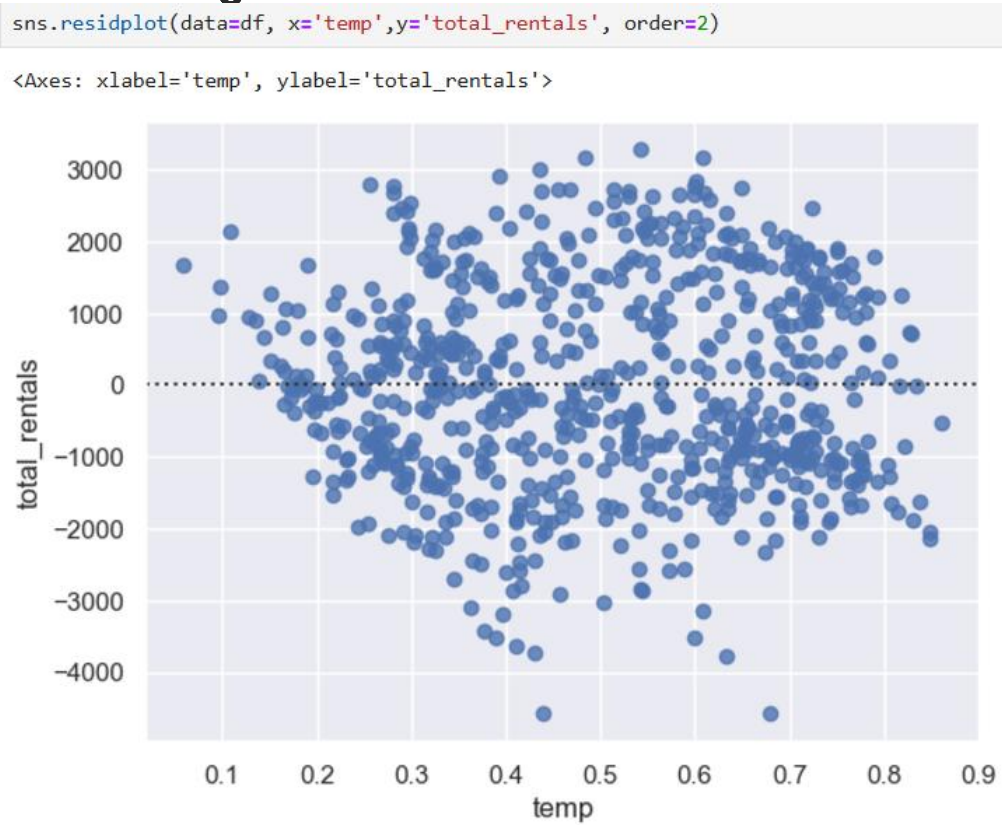
```
sns.regplot(data=df, x='temp', y='total_rentals', order=2)
```

```
<Axes: xlabel='temp', ylabel='total_rentals'>
```

Seaborn supports polynomial regression using the order parameter

Regression Plot - Bicycle Dataset

residplot with polynomial regression

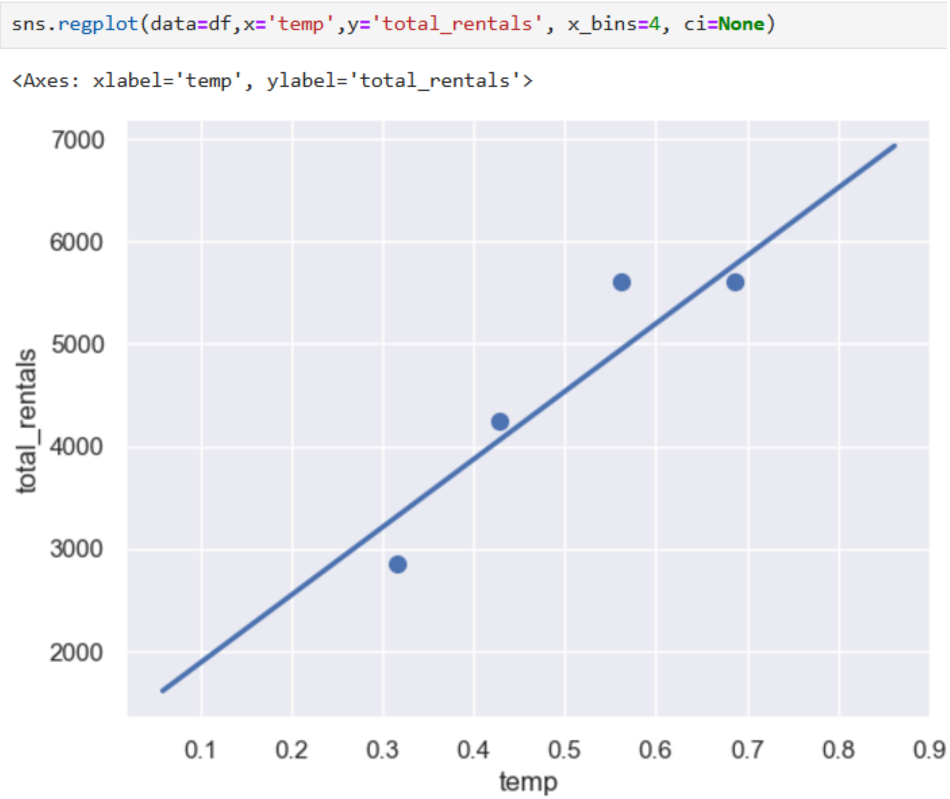


```
sns.residplot(data=df, x='temp',y='total_rentals', order=2)
```

```
<Axes: xlabel='temp', ylabel='total_rentals'>
```

Regression Plot - Bicycle Dataset

Binning the Data



```
sns.regplot(data=df,x='temp',y='total_rentals', x_bins=4, ci=None)
```

```
<Axes: xlabel='temp', ylabel='total_rentals'>
```

x_bins can be used to divide the data into discrete bins

The regression line is still fit against all the data

Case Study

Case Study

Do refer to the handout for the Case Study in the following week.

Thank You!

NYP NANYANG
THE INNOVATIVE POLYTECHNIC

www.nyp.edu.sg