

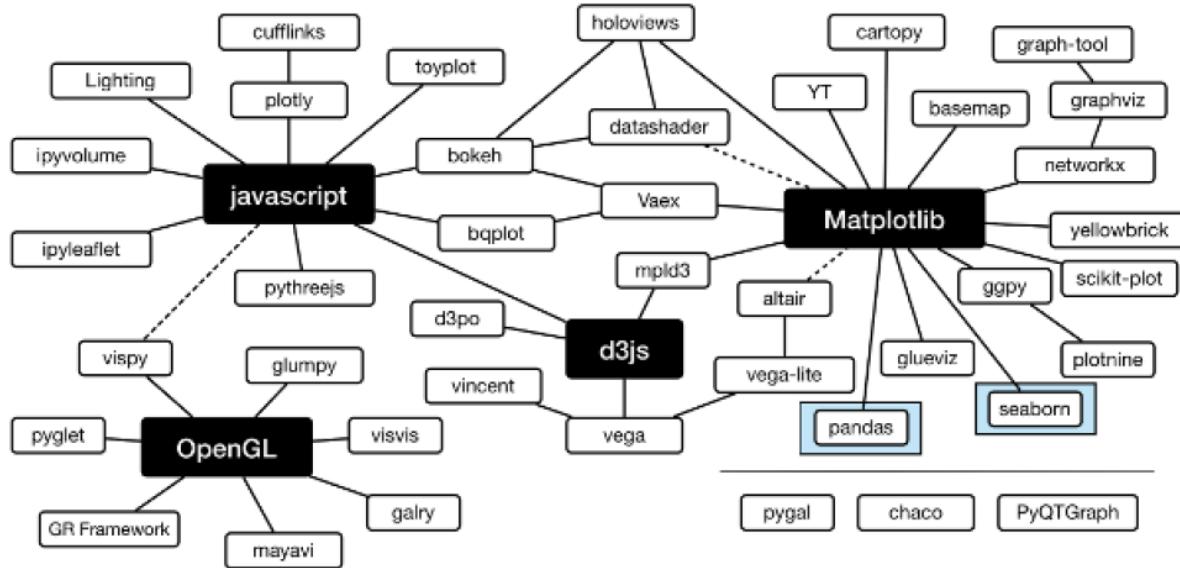


Seaborn Introduction

Introduction To Seaborn

Python Visualization Landscape

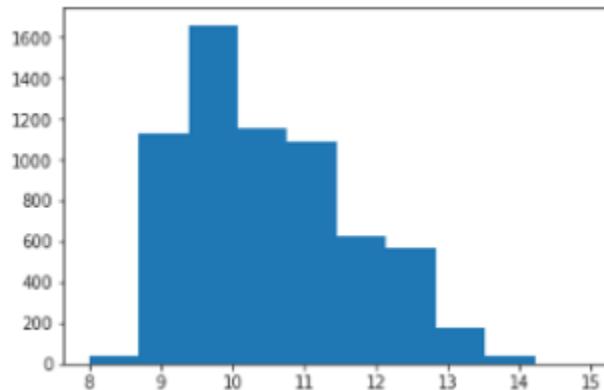
The python visualization landscape is complex and can be overwhelming



Matplotlib

- matplotlib provides the raw building blocks for Seaborn's visualizations
- It can also be used on its own to plot data

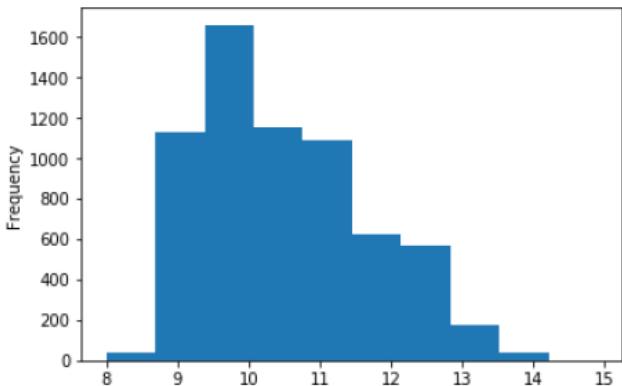
```
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv("wines.csv")
fig, ax = plt.subplots()
ax.hist(df['alcohol'])
```



Pandas

- pandas is a foundational library for analyzing data
- It also supports basic plotting capability

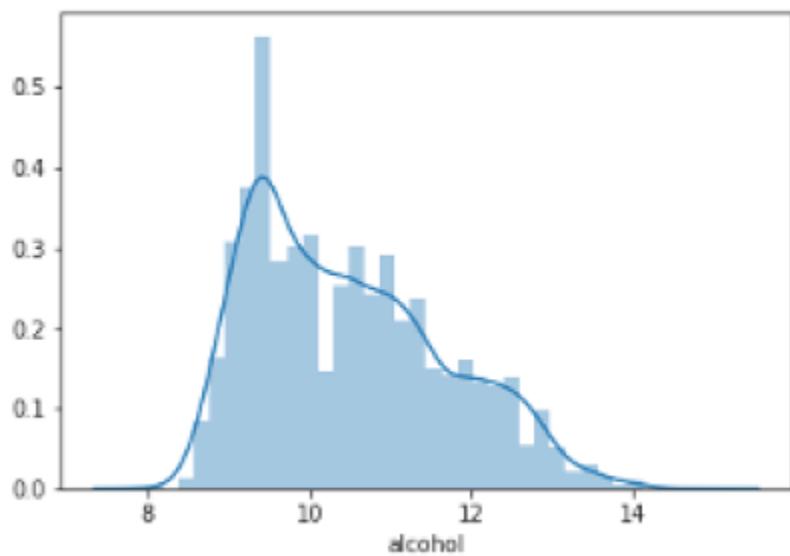
```
import pandas as pd
df = pd.read_csv("wines.csv")
df['alcohol'].plot.hist()
```



Seaborn

- Seaborn supports complex visualizations of data
- It is built on matplotlib and works best with pandas' dataframes
- The distplot is similar to the histogram shown in previous examples
- By default, generates a Gaussian Kernel Density Estimate (KDE)

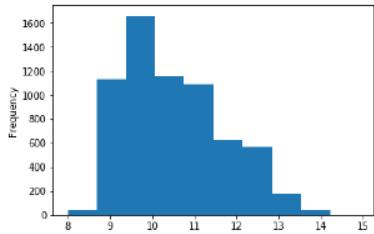
```
import seaborn as sns  
sns.distplot(df['alcohol'])
```



Histogram vs. Distplot

- Pandas histogram

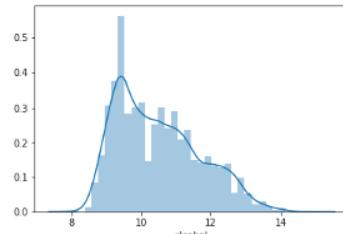
```
df['alcohol'].plot.hist()
```



- Actual frequency of observations
- No automatic labels
- Wide bins

- Seaborn distplot

```
sns.distplot(df['alcohol'])
```



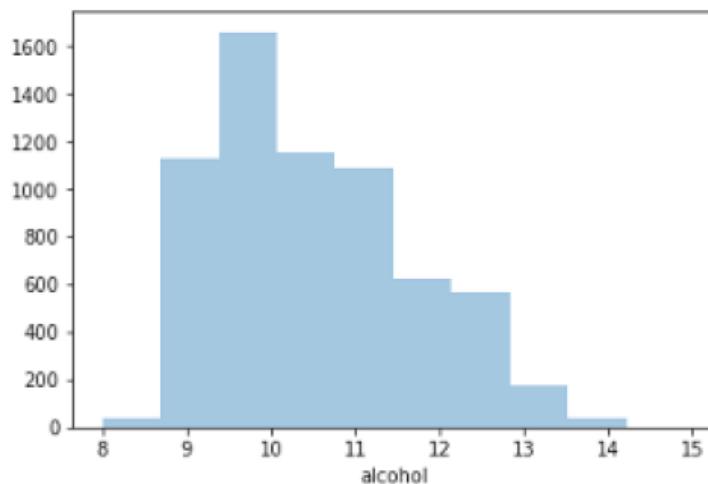
- Automatic label on x axis
- Muted color palette
- KDE plot
- Narrow bins

Using the distribution plot

Creating a histogram

- Distplot function has multiple optional arguments
- In order to plot a simple histogram, you can disable the kde and specify the number of bins to use

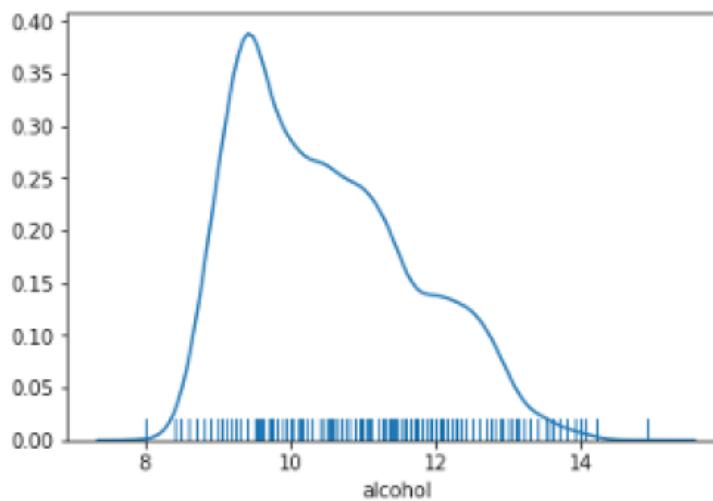
```
sns.distplot(df['alcohol'], kde=False, bins=10)
```



Alternative data distributions

- A rug plot is an alternative way to view the distribution of data
- A kde curve and rug plot can be combined

```
sns.distplot(df['alcohol'], hist=False, rug=True)
```

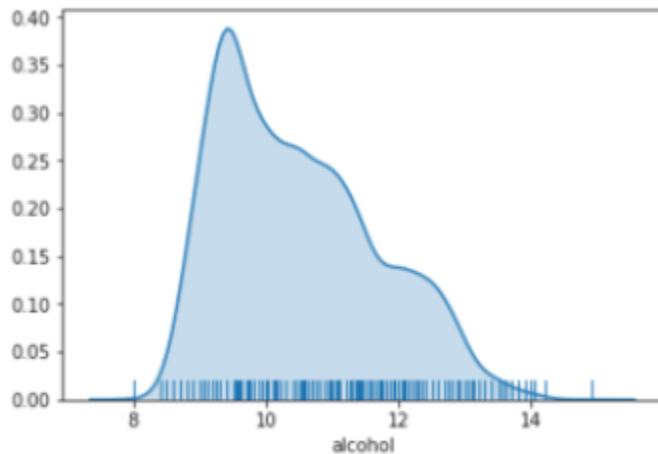


Further Customizations

- The distplot function uses several functions including kdeplot and rugplot

- It is possible to further customize a plot by passing arguments to the underlying function

```
sns.distplot(df['alcohol'], hist=False,
             rug=True, kde_kws={'shade':True})
```



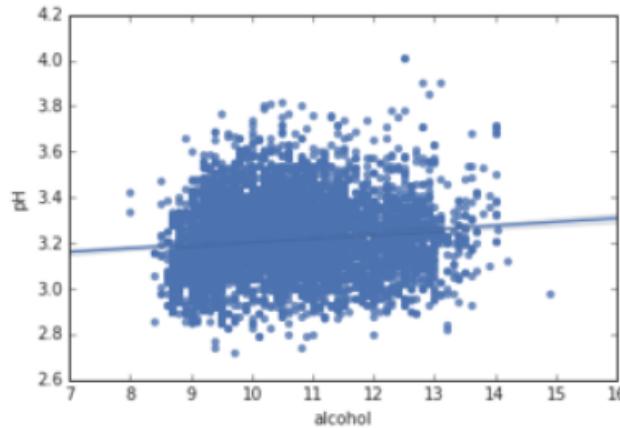
Regression Plots in Seaborn

The previous plots are uni-variant ,i.e we look only at single variable. Linear Regression, on other hand is bi-variant as we are looking through relationship between two variables.

Introduction to regplot

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

```
sns.regplot(x="alcohol", y="pH", data=df)
```



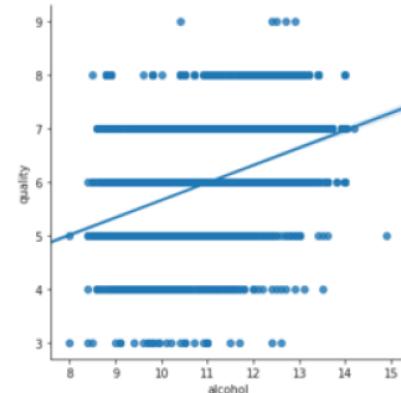
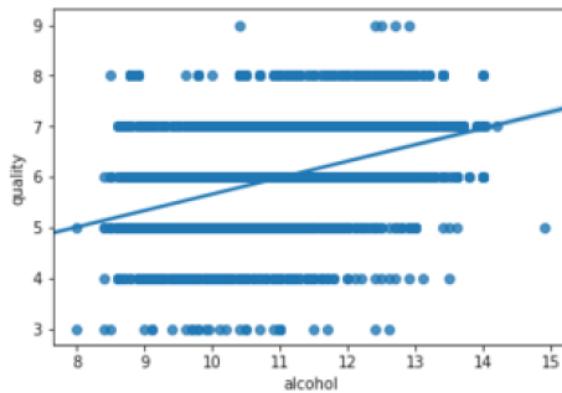
Implot() builds on top of the base regplot()

- `regplot` - low level

```
sns.regplot(x="alcohol",
             y="quality",
             data=df)
```

- `lmplot` - high level

```
sns.lmplot(x="alcohol",
            y="quality",
            data=df)
```

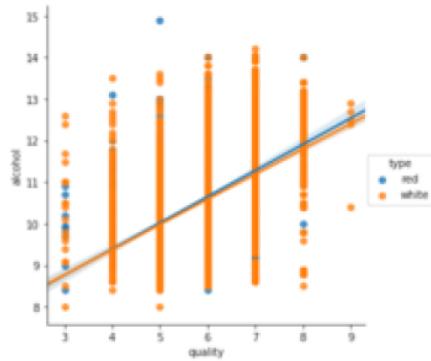


Implot is much more powerful as aspect ratio is proper in Implot and we can use hue and add columns

Implot faceting

- Organize data by colors (
hue)

```
sns.lmplot(x="quality",
            y="alcohol",
            data=df,
            hue="type")
```



- Organize data by columns (
col)

```
sns.lmplot(x="quality",
            y="alcohol",
            data=df,
            col="type")
```

