

Lesson Goals

In this lesson we will learn to use the following charts:

- Box Plots
- Violin Plots
- Histograms
- Bar Plots

We will learn to use these visualizations using the pandas matplotlib API and seaborn when necessary.

Introduction

The goal of exploratory data analysis is to gain initial insight on our dataset. Many times, this can be accomplished with a myriad of summary statistics. However, sometimes we find that "a picture is worth a thousand words" and that data visualizations can capture a large amount of data all at once in a clear and concise manner.

Box Plot (aka Box whisker plot)

Box plots are an amazing way to show the distribution of quantitative data and make comparisons across levels of a categorical variables. They help to visualize the 5-number summary as well as create a boundary that will help us distinguish which outliers are extreme. Those 5 numbers are minimum, 25th percentile, 50th percentile (Median), 75th percentile, maximum

The box is defined as containing the data between the 25th and 75th percentiles. A line is drawn in the box to show where the median (or 50th percentile) lies. Box plots also have two lines extending from each end of the box called whiskers. The whiskers are typically drawn by finding the smallest and largest points still within 1.5 times the interquartile range minus the lower quartile or 1.5 times the interquartile range plus the upper quartile.

The interquartile range is the difference between the first and third quartiles. Since this verbal explanation might be confusing, here are some formulas:

$$\text{IQR} = \text{Third Quartile} - \text{First Quartile}$$

$$\text{Upper Whisker Limit} < \text{Third Quartile} + 1.5 * \text{IQR}$$

$$\text{Lower Whisker Limit} > \text{First Quartile} - 1.5 * \text{IQR}$$

With all of this in mind, let's plot a box plot of our vehicles data. We first load the [vehicles dataset](#) using pandas. And then we will plot a box plot of the combined MPG

```
import pandas as pd
data = pd.read_csv('~path/vehicles.csv')
data.boxplot(column="Combined MPG")
```

Note that we did not use the `.plot` function here. We can infer from this box plot that the data is skewed due to a large number of extreme outliers that have very good MPG. We can also see that the median is not exactly in the middle of the box.

Box plots using Seaborn

More detailed plots can be made using the seaborn library. In this example we are finding the relation between a numerical variable (Highway MPG here) and a categorical variable (Fuel Type).

```
ax = sns.boxplot(x="Fuel Type", y="Highway MPG", data=data)
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

We can also draw box plot with nested grouping by two categorical variables. Let's say we want to add an additional layer classification in the above box plot based on the cylinder types 4.0 liters and 6.0 liters. Note that we have over 10 different categories of values in the column "Cylinders". Therefore the first step is to select/filter the data that we are interested in.

```
cyl = [6.0, 4.0]
data1 = data[data.Cylinders.isin(cyl)]
data1.shape
```

```
(26259, 15)
```

Now we can use the box plot on this new data-frame as shown below:

```
ax = sns.boxplot(x="Fuel Type", y="Highway MPG", hue="Cylinders", data=data1, palette="Set3")
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
```

It is important to note that we are using Facetgrid feature of seaborn library as discussed in the previous class.

More details on seaborn box plots can be found [here](#)

Violin Plot

The violin plot is very similar to the box plot. The main difference is that the box plot is of even width while the violin plot varies in thickness throughout the plot. The variation in thickness signifies how common a value is (with the thickest section signifying the mode average). There is an additional layer inside that signifies values that occur 95% of the time. Finally there is a central dot that signifies the median average value.

The violin plot can be visualized using seaborn.

```
import seaborn as sns
sns.violinplot("Combined MPG", data=data)
```

Histogram

Histograms are great plots for a simplified look at the distribution of a dataset. We separate the data into bins and then plot the count in each bin. The number of bins can greatly alter how the histogram looks. Therefore, we need to be cautious of our selection of the number of bins. The default number of bins with Pandas is 10.

As an example, we will plot the histogram for Fuel Barrels/Year. This is the histogram for the default 10 bins. Notice how it resembles the normal distribution.

```
vehicles['Fuel Barrels/Year'].hist()
```

When we switch to 50 bins, the data almost looks bimodal.

```
vehicles['Fuel Barrels/Year'].hist(bins=50)
```

Using Seaborn for histograms

We can use the `distplot()` in seaborn to plot the distribution of variables

```
sns.distplot(x);
```


```
sns.distplot(x, bins=20)
```

Bar Plots

Bar plots are visually similar to histograms. However, they represent different data. Bar plots display categorical data while histograms display numerical data. Each bar shows the count of each category. This way we are able to clearly visualize the distribution of the data between the different categories.

In this example we would like to plot the bar plot for drivetrain. In order to do this, we first find the counts of each category in this variable using the `value_counts` function.


```
data['Drivetrain'].value_counts().plot.bar()
```



Using Seaborn for Bar plots

We can use the `barplot()` as illustrated below:

```
ax = sns.barplot(x="Fuel Type", y="Highway MPG", hue="Cylinders", data=data1)
ax.set_xticklabels(ax.get_xticklabels(),rotation=90)
```



Summary

In this lesson we learned how to plot different visualizations that can help us discover information about our data. We learned about box plots and to visualize the 5-number summary as well as outliers. We learned about violin plots that add an element of frequency to box plots. We learned about histograms that can help us visualize the distribution of our data by bucketing the data and plotting the count in each bucket. Finally we learned about bar plots. These plots help us visualize the distribution in categorical data. Hopefully, all four visualizations will give us the tools we need to be effective at exploratory data analysis.