Plot reference: NPTEL

Data Visualization

- Data visualization allows us to interpret data
- It allows us to play with various parameters and its impact on overall outcome or prediction
- To provide more insight
- Exploratory tool for data scientist

Types of Visulization

- Scientific Visualization
  - Structural Data – Seismic, Medical
- Information Visualization –

No inherent structure – News, stock market, top grossing movies, facebook connections

- Visual Analytics – Use visualization to understand and synthesize large amounts of multimodal data – audio, video, text, images, networks of people ..

Why visualize data?

- Observe the patterns
- Identify extreme values that could be anomalies
- Easy interpretation
- To provide requires and crisp solution/outcome to management or higher authority
- Incorporate visualization principles to build an interactive visualization of your own data

Types of Plots

- Scatterplot
- Histogram
- Barplot
- Box and whiskers plot
- Pair wise plots

Popular Tools and Software

- Excel
- Python
- R
- Tableau

What is a scatter plot?

- A scatter plot is a set of points that represents the values obtained for two different variables plotted on a horizontal and vertical axes

When to use scatter plots?

- Scatter plots are used to convey the relationship between two numerical variables
- Scatter plots are sometimes called correlation plots because they show how two variables are correlated

Importing data into Spyder

- Importing necessary libraries

```
import pandas as pd
```

'pandas' library to work with dataframes

```
import numpy as np
```

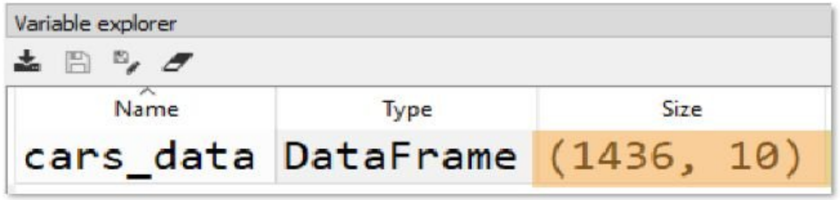
'numpy' library to do numerical operations

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

'matplotlib' library to do visualization

- Importing data

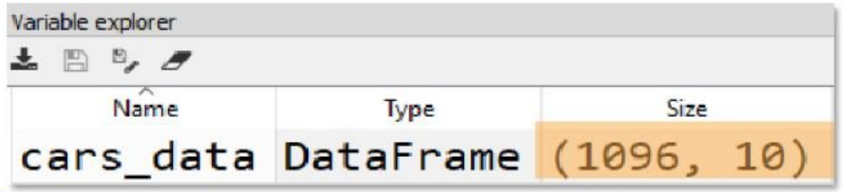
```
cars_data = pd.read_csv('Toyota.csv',index_col=0,
                                       na_values=["??","????"])
```


Variable explorer

Name	Type	Size
cars_data	DataFrame	(1436, 10)

- Removing missing values from the dataframe

```
cars_data.dropna(axis = 0, inplace=True)
```


Variable explorer

Name	Type	Size
cars_data	DataFrame	(1096, 10)

Scatter plot

```
                     x                               y
plt.scatter(cars_data['Age'], cars_data['Price'], c='red')
```

```
plt.title('Scatter plot of Price vs Age of the cars')
```

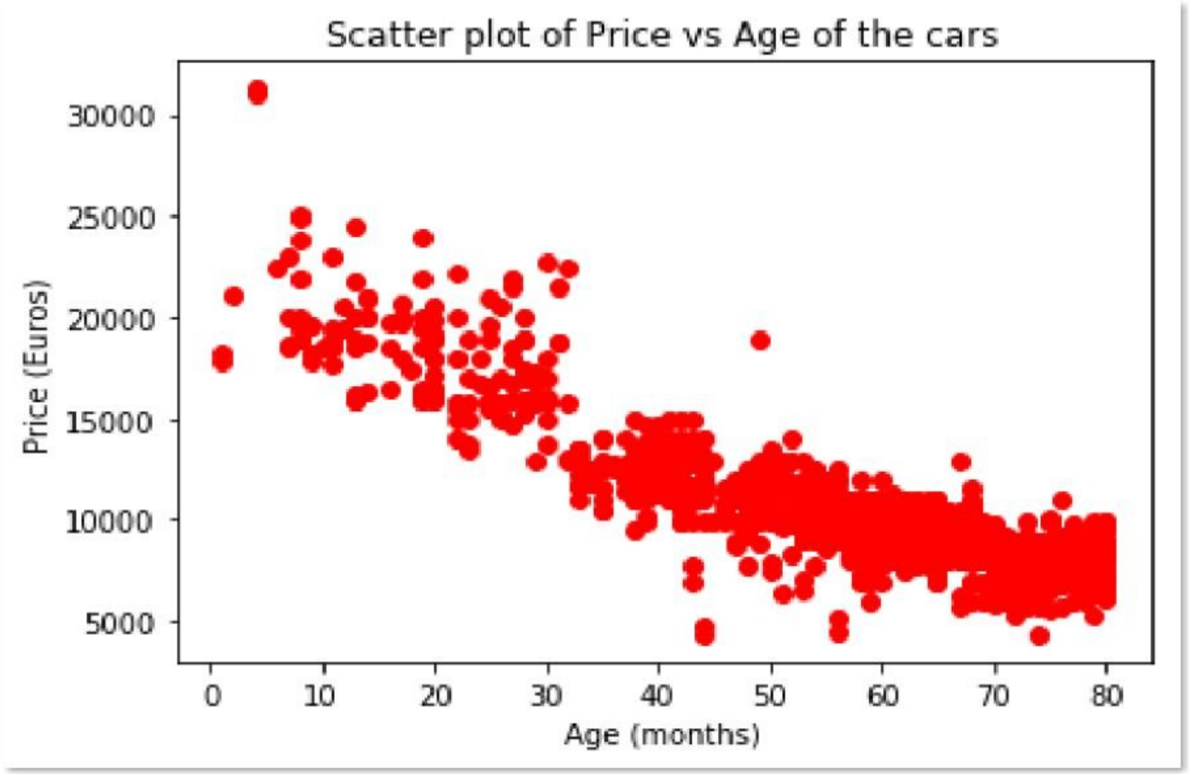
```
plt.xlabel('Age (months)')
```

```
plt.ylabel('Price (Euros)')
```

```
plt.show()
```

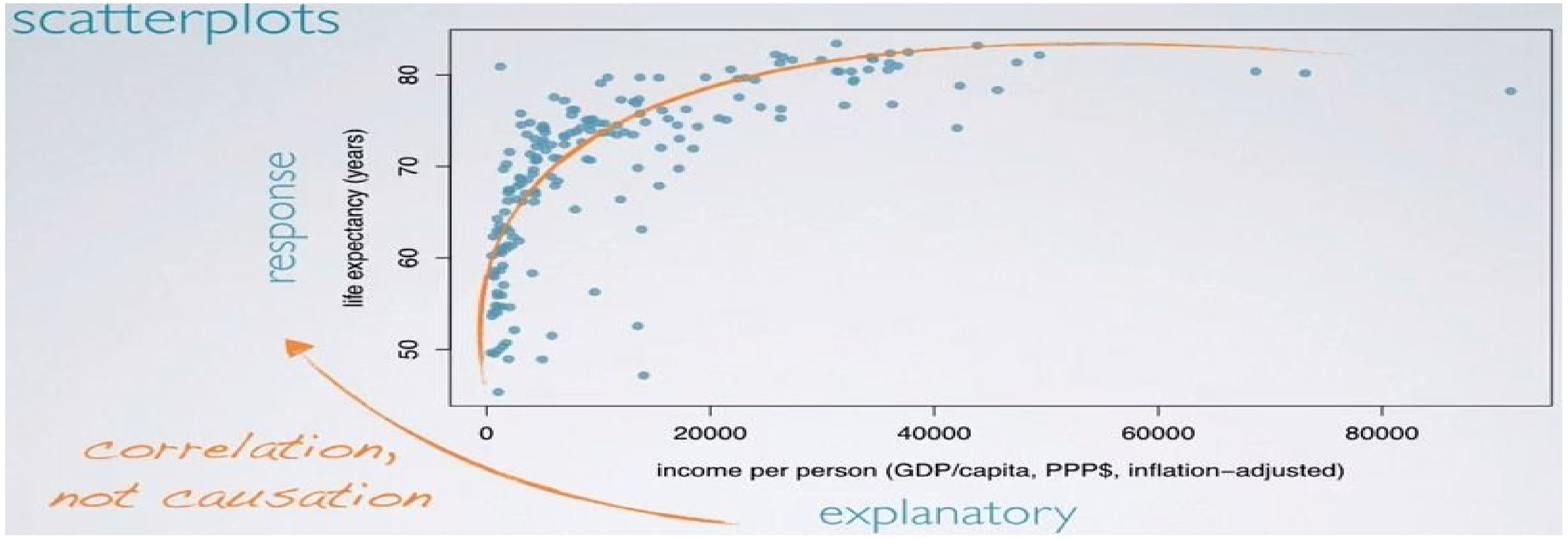
Scatter Plot

- The price of the car decreases as age of the car increases

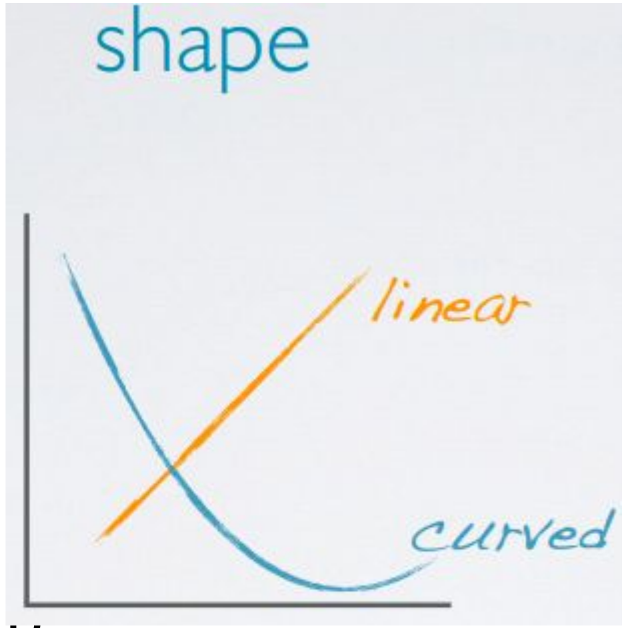


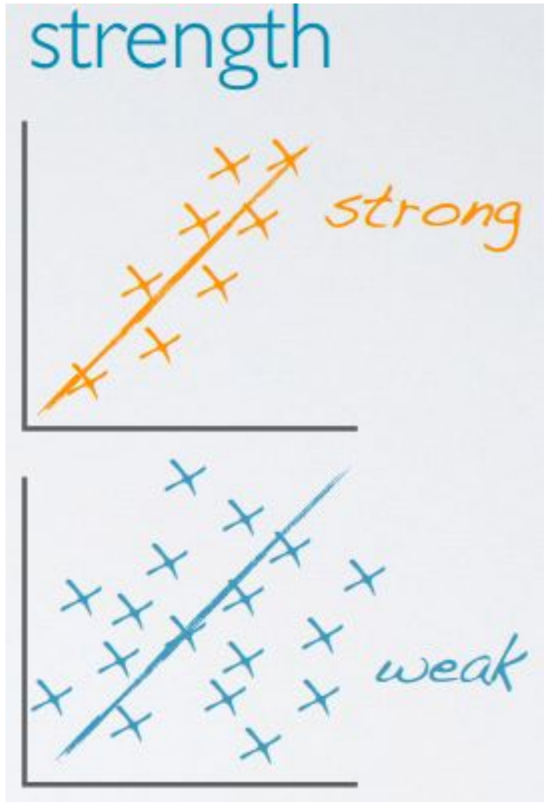
Visualizing Numerical data

Scatter Plot: A common tool for visualizing the relationship between two numerical variables



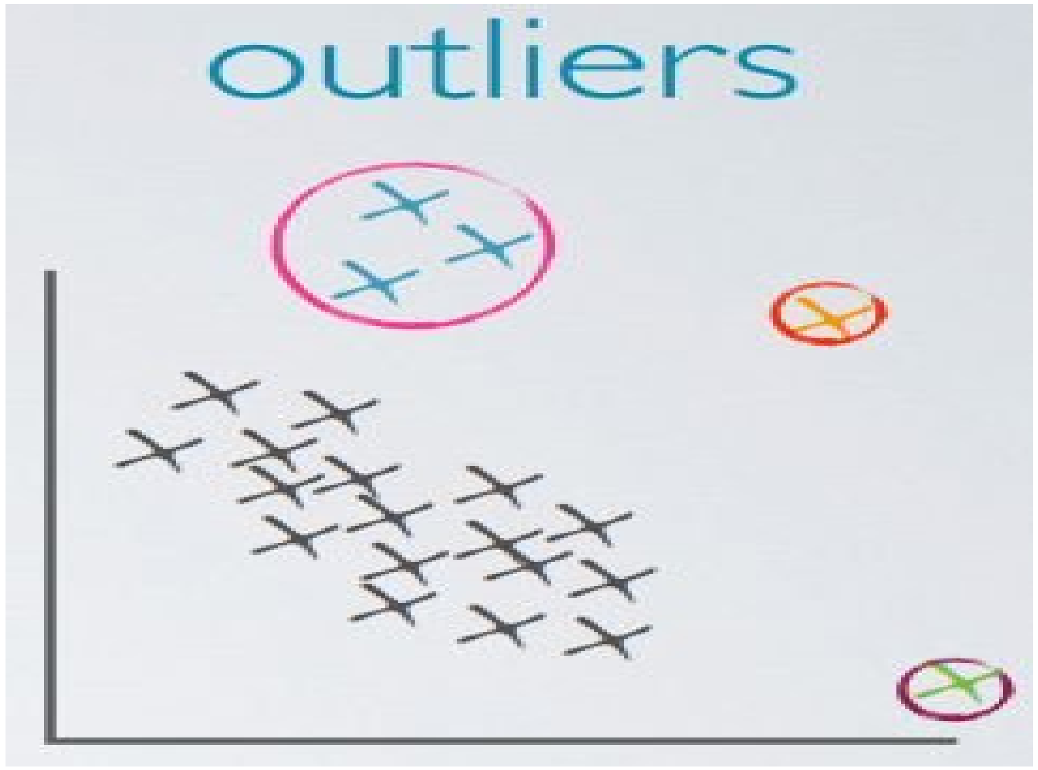
- The shape of the relationship:
- Is it **linear**;
- Or **non-linear**;

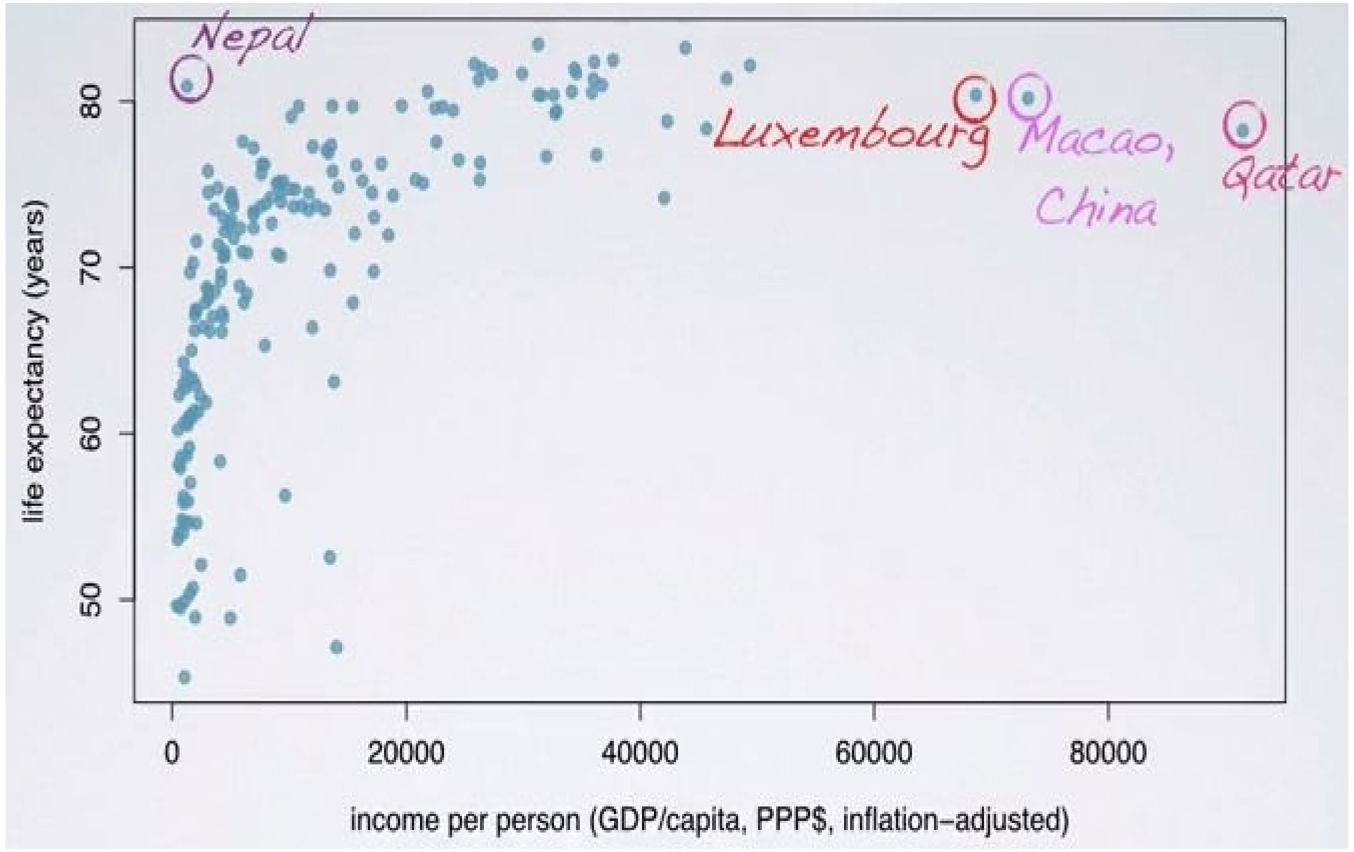




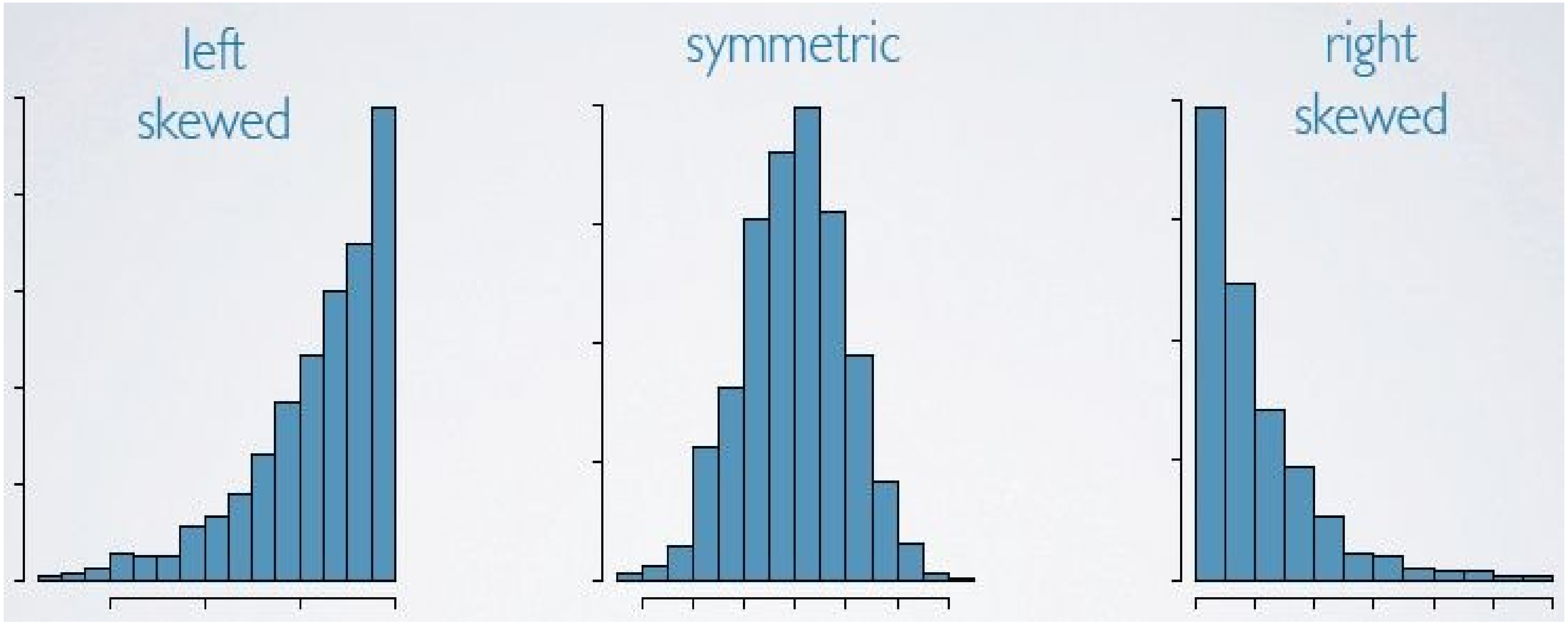
- The strength of the relationship:
- **Strong** indicated by little scatter?
- Or **weak**, indicated by lots of scatter?

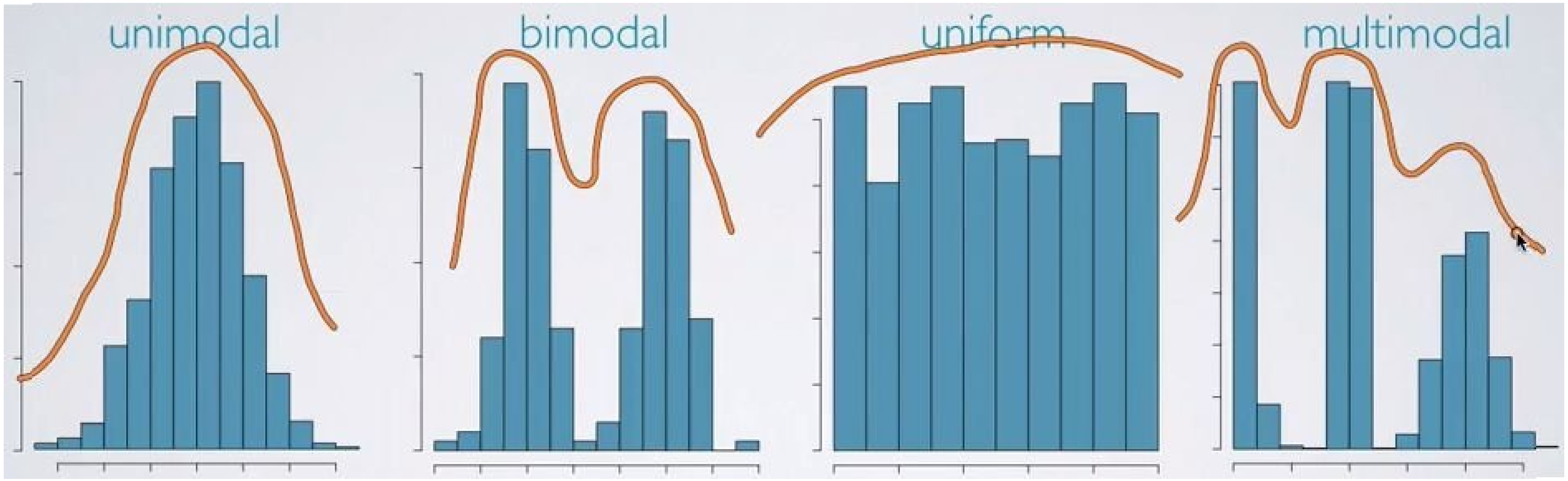
Outliers Detection




Nepal
Luxembourg
Macao, China
Qatar
life expectancy (years)
50
60
70
80
0
20000
40000
60000
80000
income per person (GDP/capita, PPP$, inflation-adjusted)

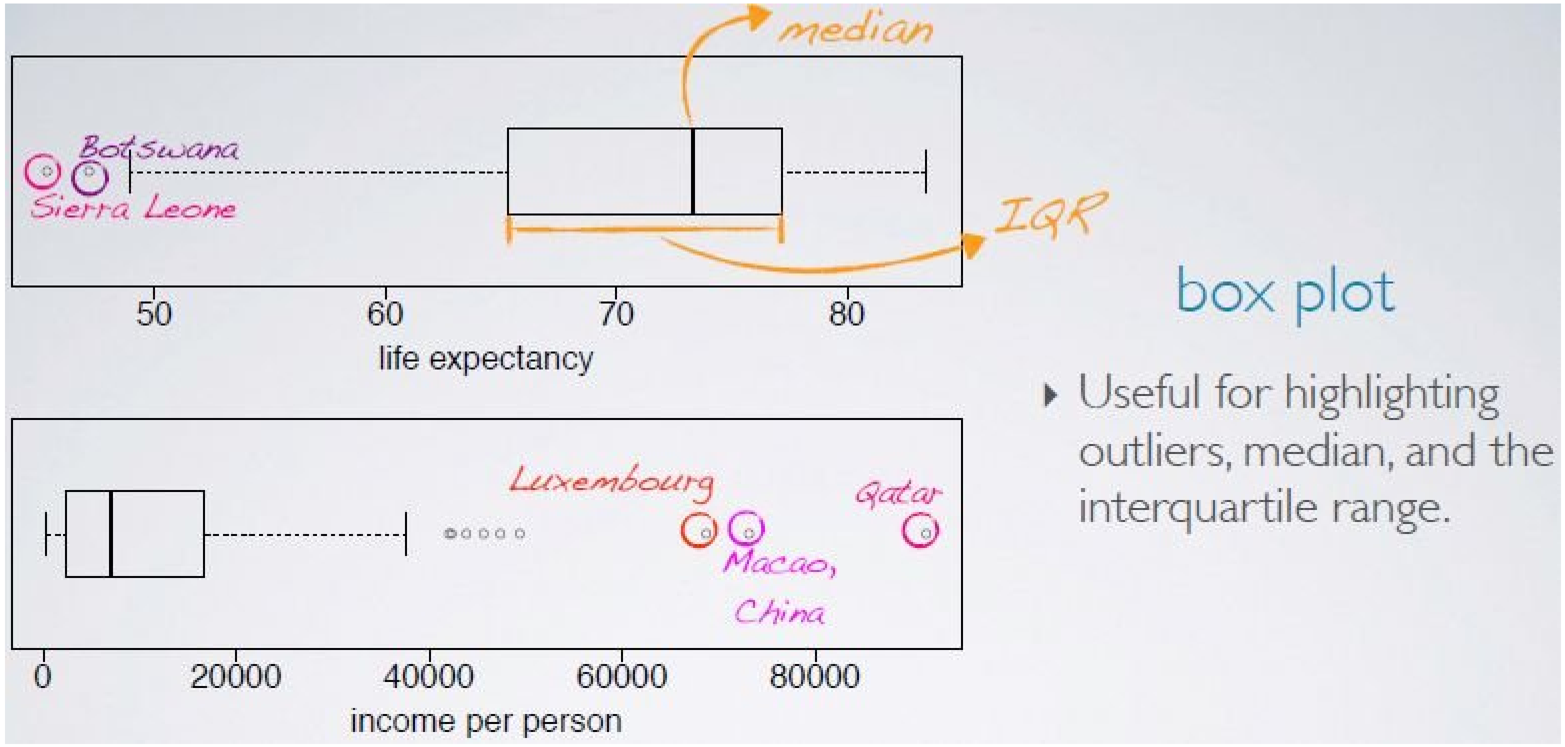
Visualization





Single Mode prediction, 1 or 2 predictions, Continuous and uniform data prediction , More than 2 predictions

Box plot

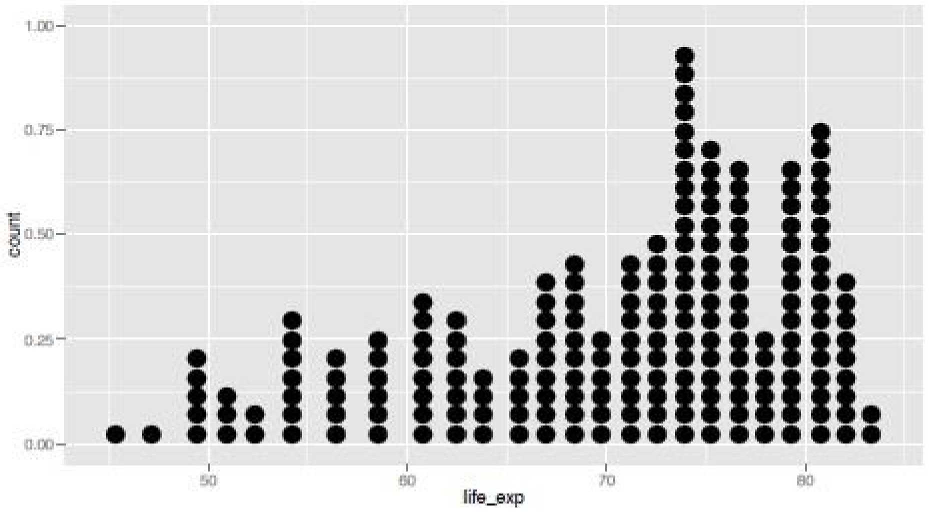


box plot

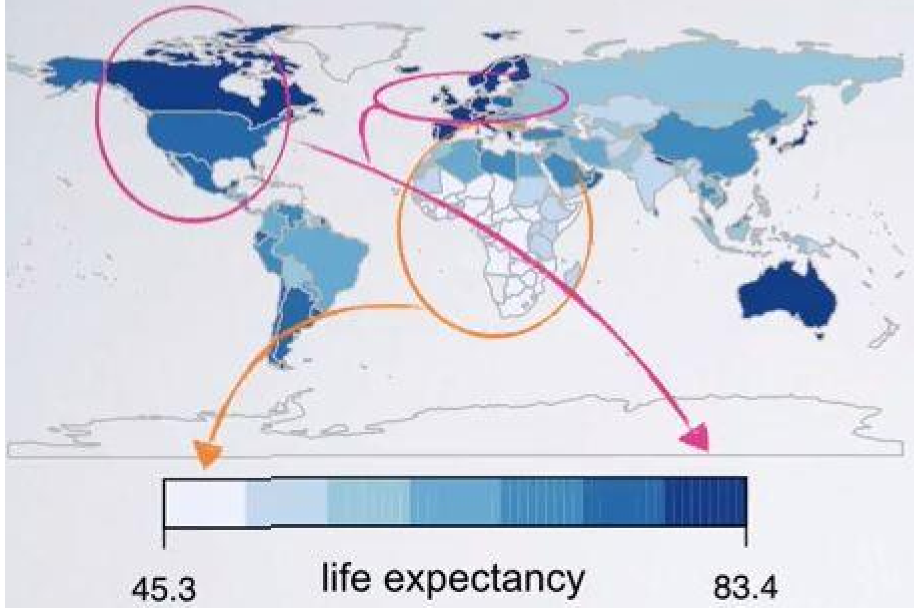
- Useful for highlighting outliers, median, and the interquartile range.

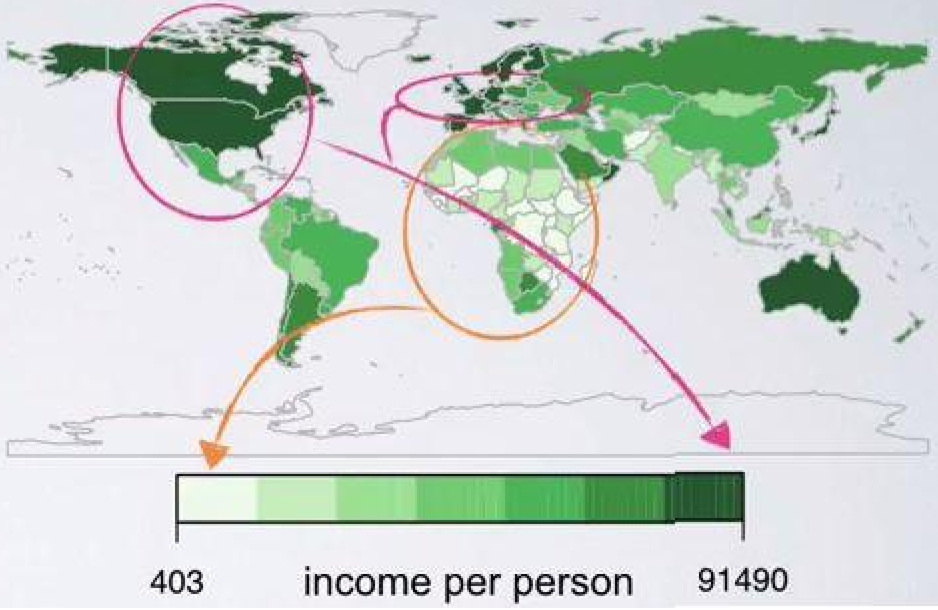
Dot PLOT

- A dot plot is useful especially when individual values are of interest.
- However, as the sample size increases, the dot plot may get too busy.



- Useful for highlighting the spatial distribution.

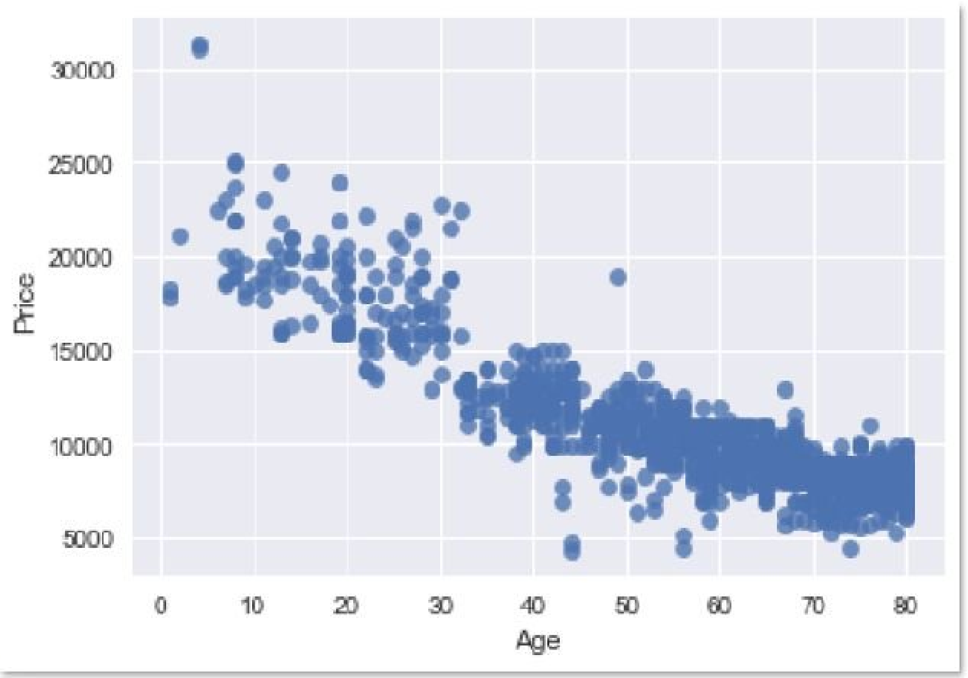




Scatter plot

- Scatter plot of *Price vs Age* without the regression fit line

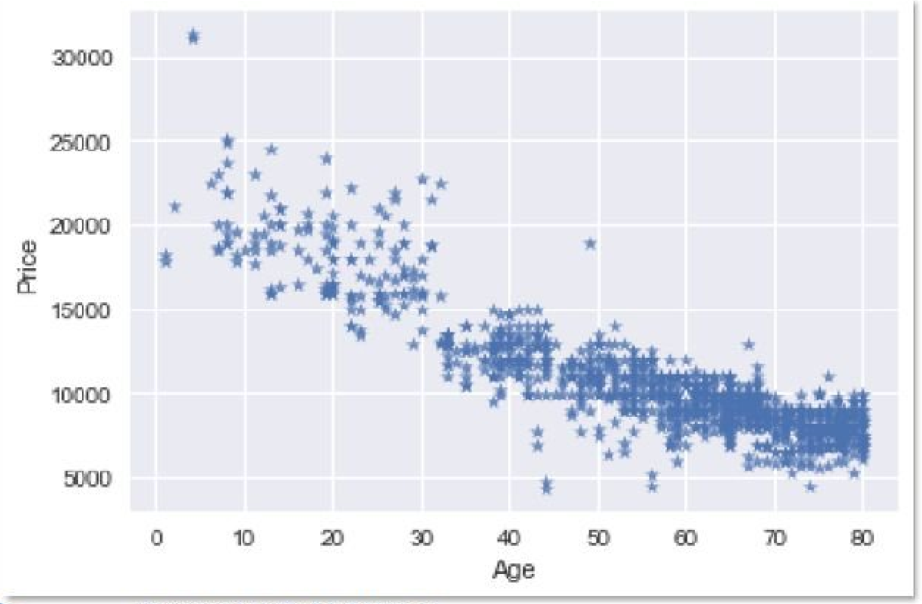
```
sns.regplot(x=cars_data['Age'], y=cars_data['Price'],
                fit_reg=False)
```



Scatter plot

- Scatter plot of *Price vs Age* by customizing the appearance of markers

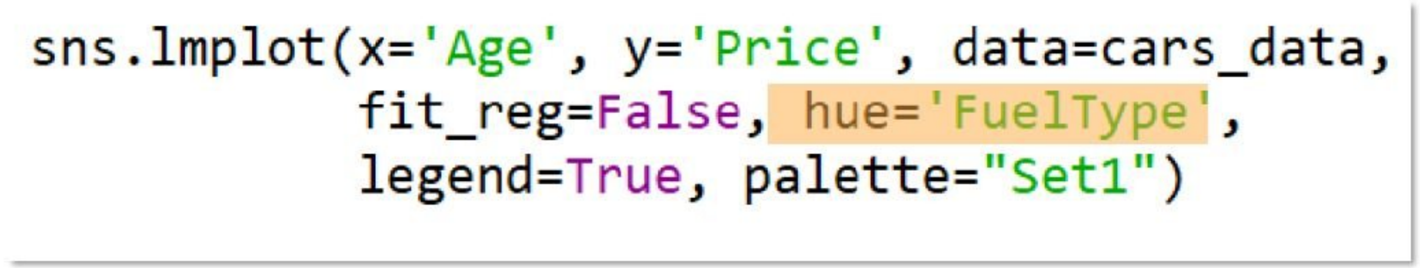
```
sns.regplot(x=cars_data['Age'], y=cars_data['Price'],
            marker="*", fit_reg=False)
```



Scatter plot

- Scatter plot of *Price vs Age by FuelType*

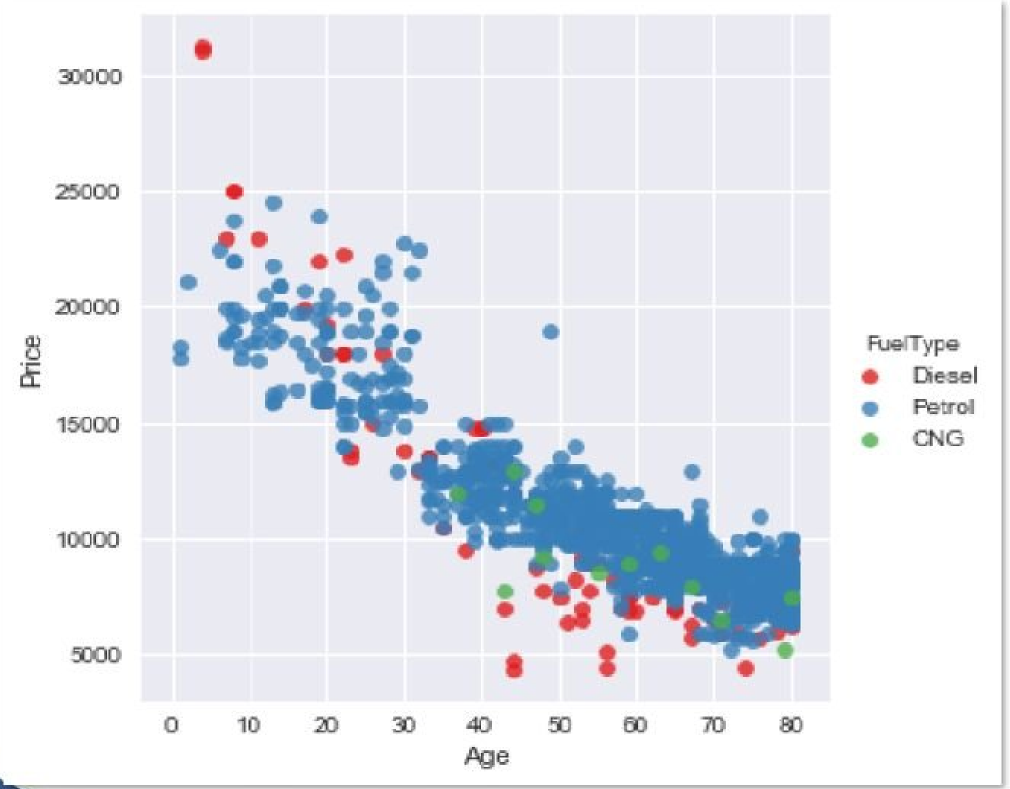
- Using hue parameter, including another variable to show the fuel types categories with different colors



```
sns.lmplot(x='Age', y='Price', data=cars_data,
           fit_reg=False, hue='FuelType',
           legend=True, palette="Set1")
```

Scatter plot

- Scatter plot of *Price vs Age by FuelType*



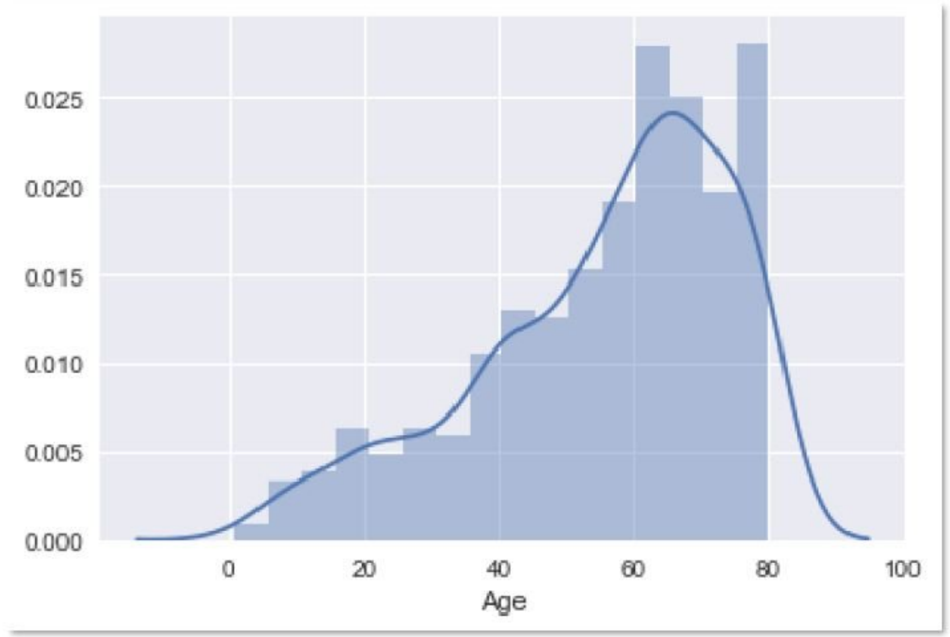
Similarly, custom the appearance of the markers using

- transparency
- shape
- size

Histogram

- Histogram with default kernel density estimate

```
sns.distplot(cars_data['Age'] )
```



Histogram

What is a histogram?

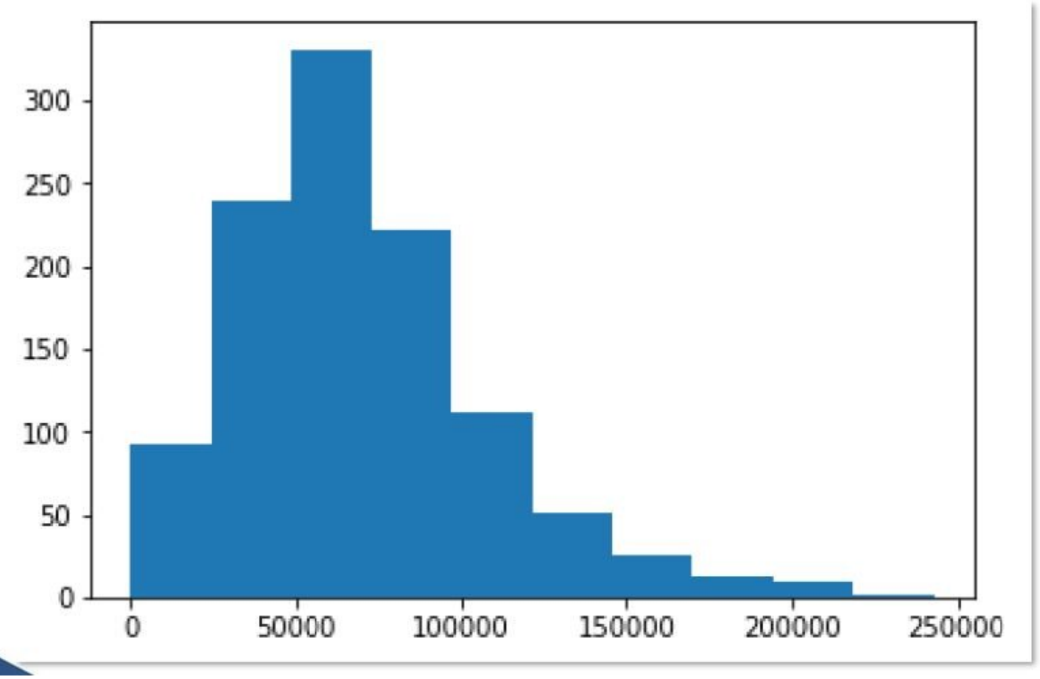
- It is a graphical representation of data using bars of different heights
- Histogram groups numbers into ranges and the height of each bar depicts the frequency of each range or bin

When to use histograms?

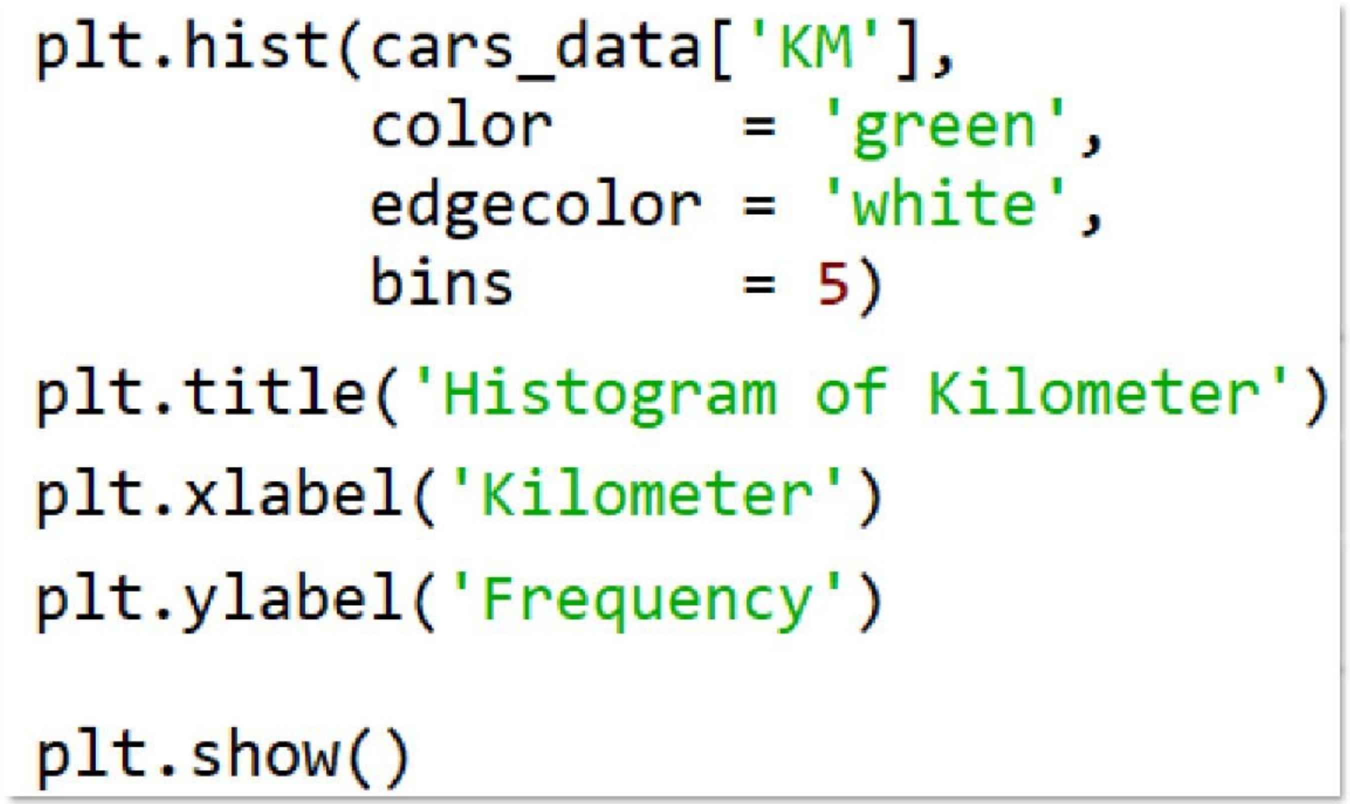
- To represent the frequency distribution of numerical variables x

```
plt.hist(cars_data['KM'])
```

$\longrightarrow$ Histogram with default arguments



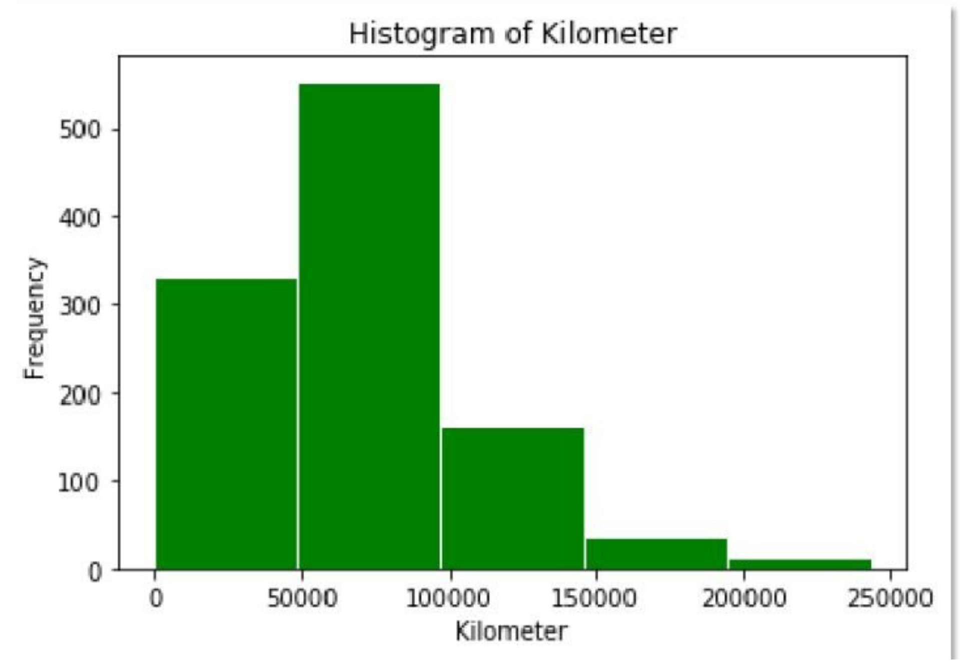
Histogram



```
plt.hist(cars_data['KM'],
         color     = 'green',
         edgecolor = 'white',
         bins      = 5)
plt.title('Histogram of Kilometer')
plt.xlabel('Kilometer')
plt.ylabel('Frequency')

plt.show()
```

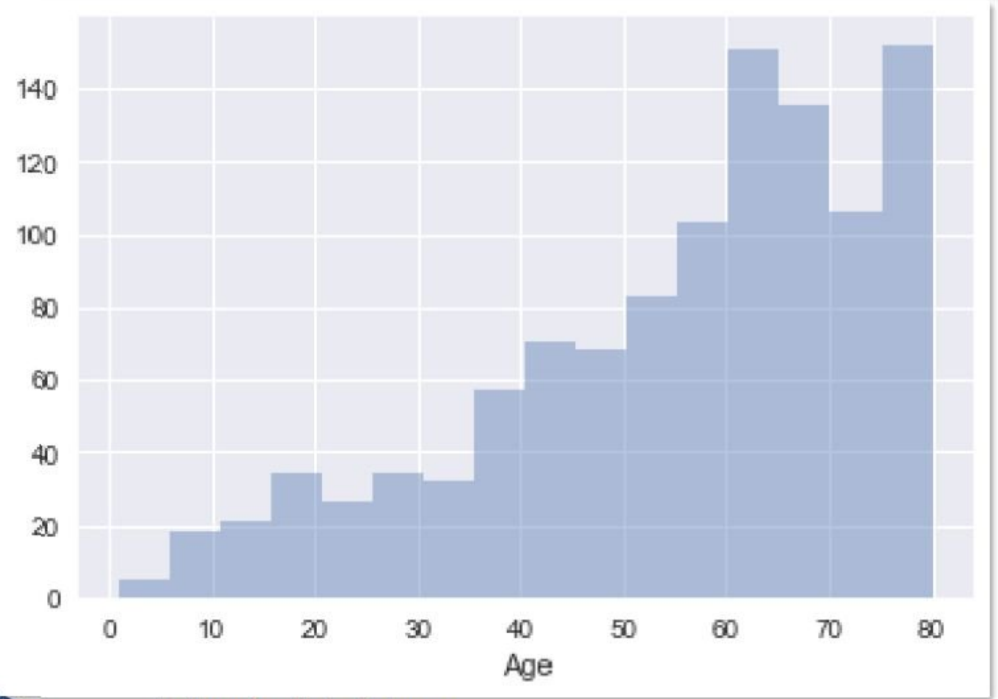
- Frequency distribution of kilometre of the cars shows that most of the cars have travelled between 50000 – 100000 km and there are only few cars with more distance travelled



Histogram

- Histogram without kernel density estimate

```
sns.distplot(cars_data['Age'],kde=False)
```



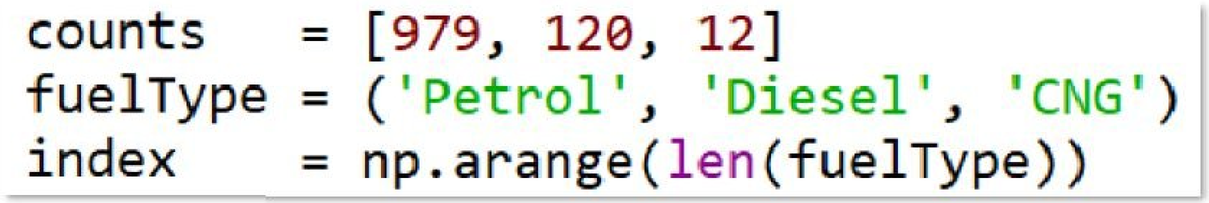
What is a bar plot?

- A bar plot is a plot that presents categorical data with rectangular bars with lengths proportional to the counts that they represent

When to use bar plot?

- To represent the frequency distribution of categorical variables
- A bar diagram makes it easy to compare sets of data between different groups

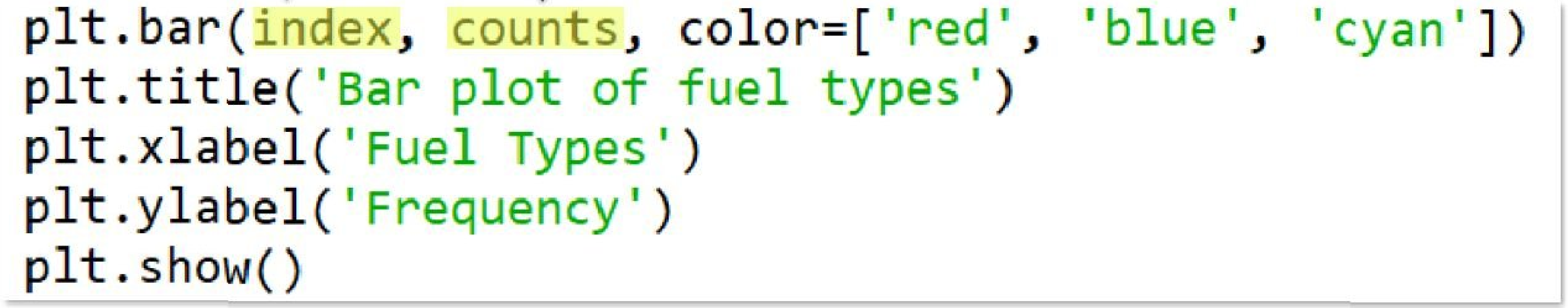
Bar plot



```
counts   = [979, 120, 12]
fuelType = ('Petrol', 'Diesel', 'CNG')
index    = np.arange(len(fuelType))
```

x

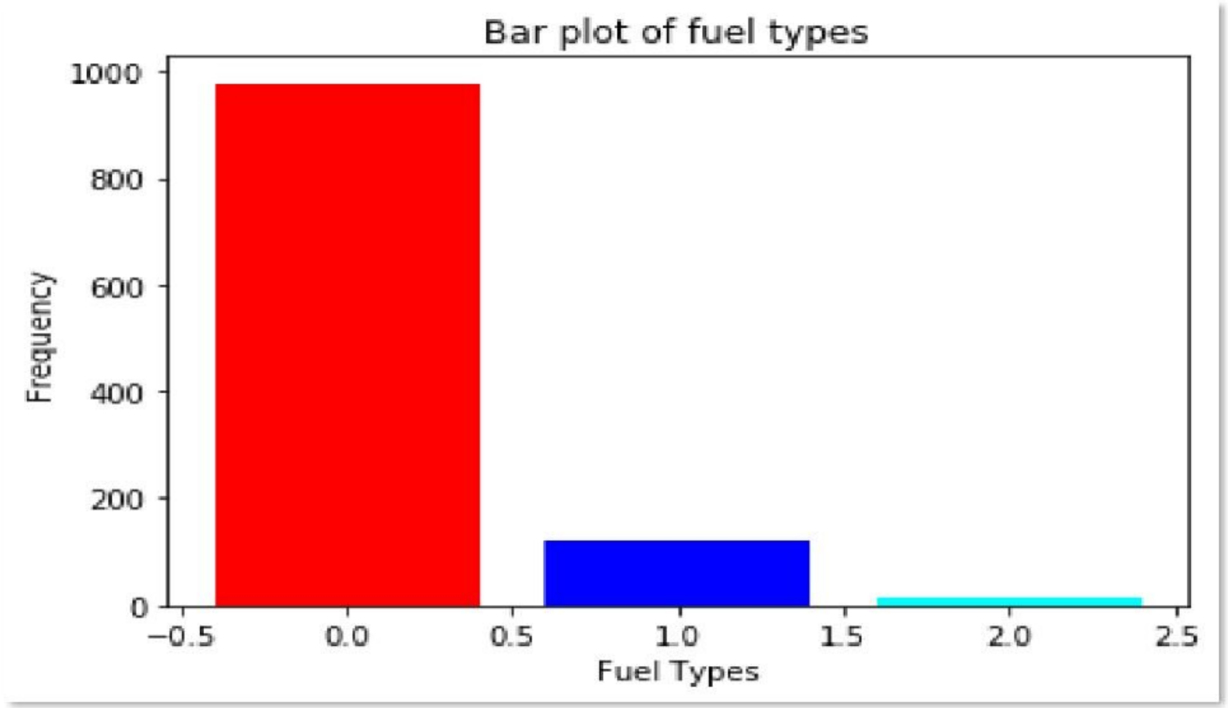
height of the bars



```
plt.bar(index, counts, color=['red', 'blue', 'cyan'])
plt.title('Bar plot of fuel types')
plt.xlabel('Fuel Types')
plt.ylabel('Frequency')
plt.show()
```

Bar plot

- Frequency distribution of fuel type



```python
counts   = [979, 120, 12]
fuelType = ('Petrol', 'Diesel', 'CNG')
index    = np.arange(len(fuelType))
```

x

height of the bars

```python
plt.bar(index, counts, color=['red', 'blue', 'cyan'])
plt.title('Bar plot of fuel types')
plt.xlabel('Fuel Types')
plt.ylabel('Frequency')
plt.xticks(index, fuelType,rotation = 90)
plt.show()
```

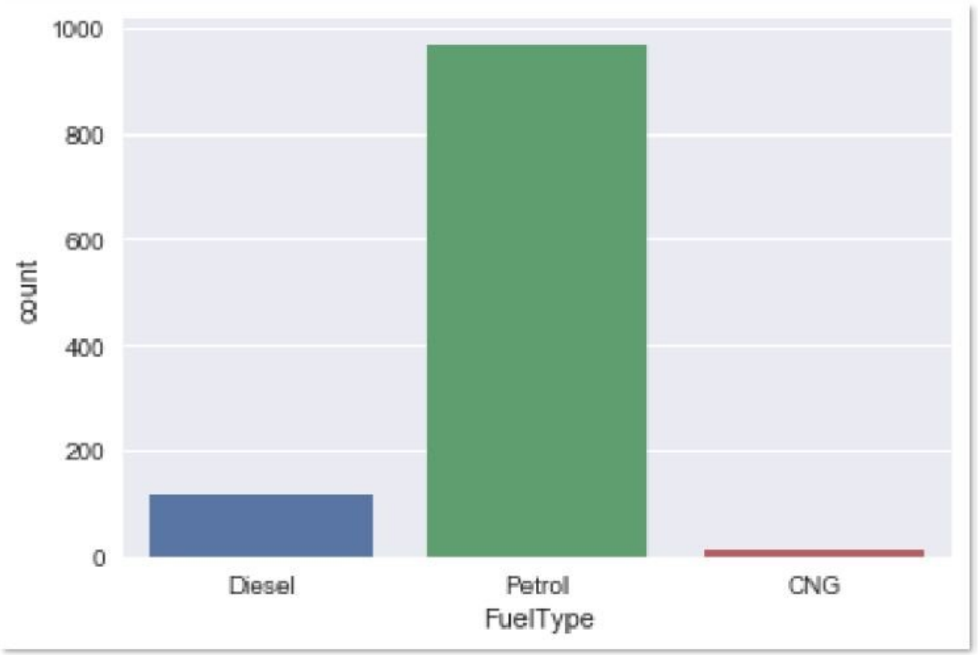
Set the labels of the xticks

Set the location of the xticks

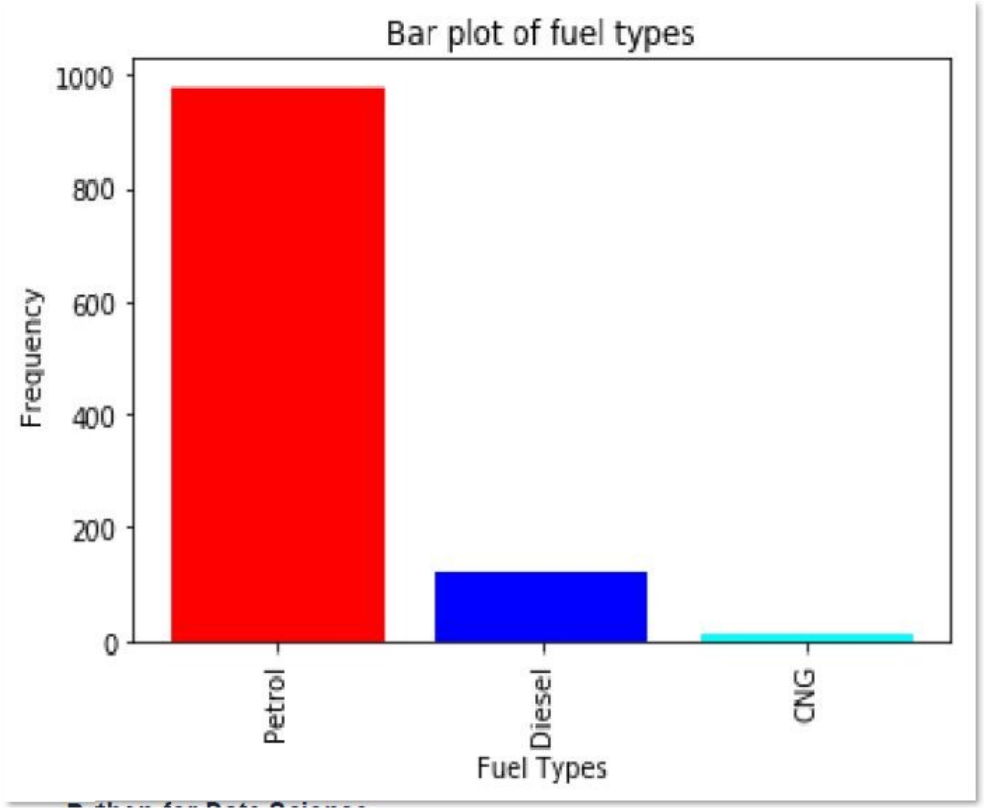
Bar plot

- Frequency distribution of fuel type of the cars

```
sns.countplot(x="FuelType", data=cars_data)
```



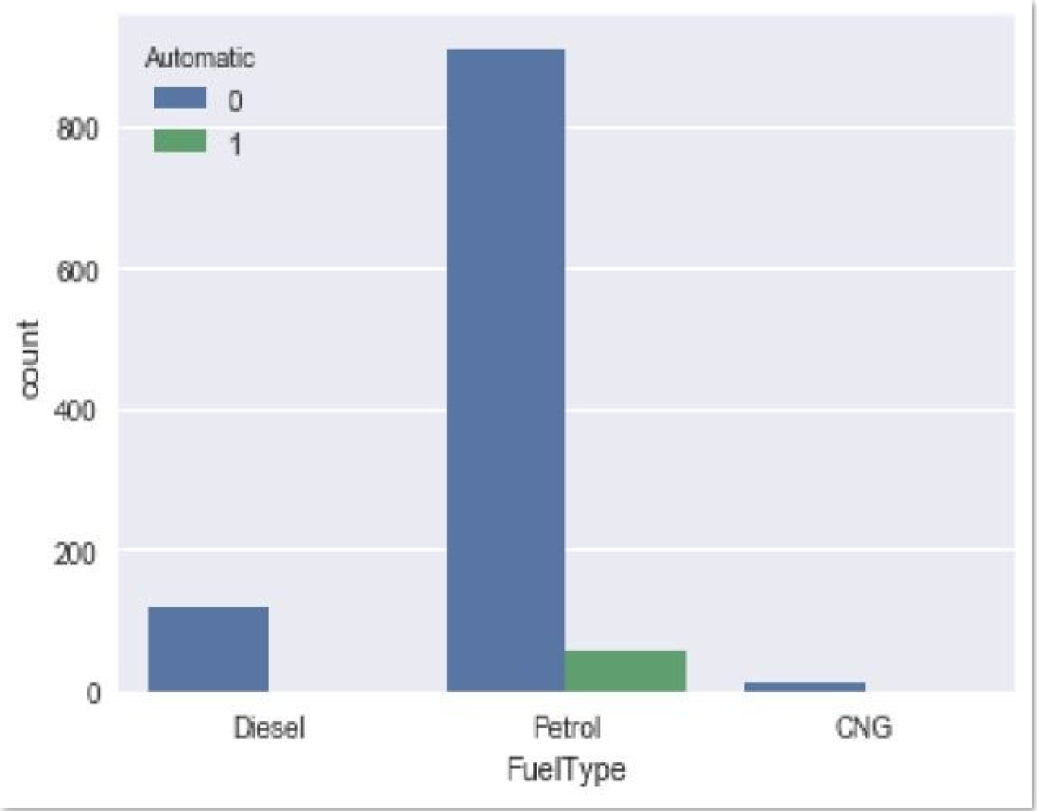
- Bar plot of fuel type shows that most of the cars have petrol as fuel type



Grouped bar plot

- Grouped bar plot of *FuelType* and *Automatic*

```
sns.countplot(x="FuelType", data=cars_data, hue = "Automatic")
```



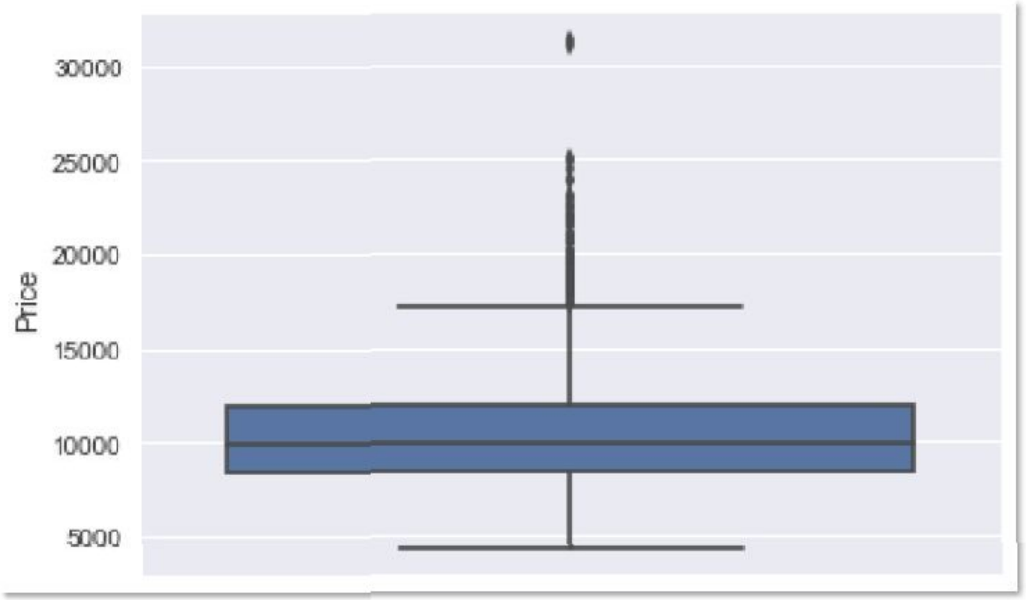
```
pd.crosstab(index   = cars_data['Automatic'],
            columns = cars_data2['FuelType'],
            dropna  = True)
```

```
Out[5]:
FuelType   CNG  Diesel  Petrol
Automatic
0           15     144    1104
1            0       0      73
```

Box and whiskers plot – numerical variable

- Box and whiskers plot of *Price* to visually interpret the five-number summary

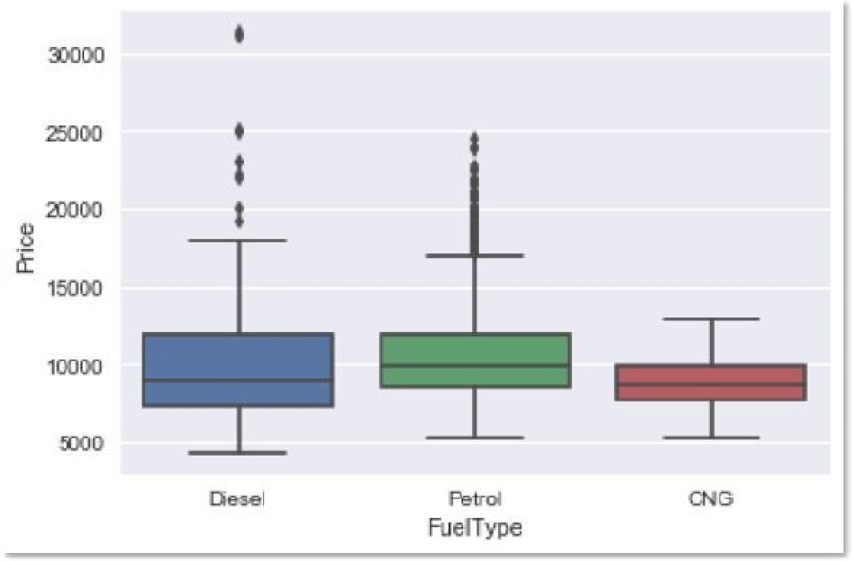
```
sns.boxplot(y=cars_data["Price"] )
```



Box and whiskers plot

- Box and whiskers plot for numerical vs categorical variable
- Price of the cars for various fuel types

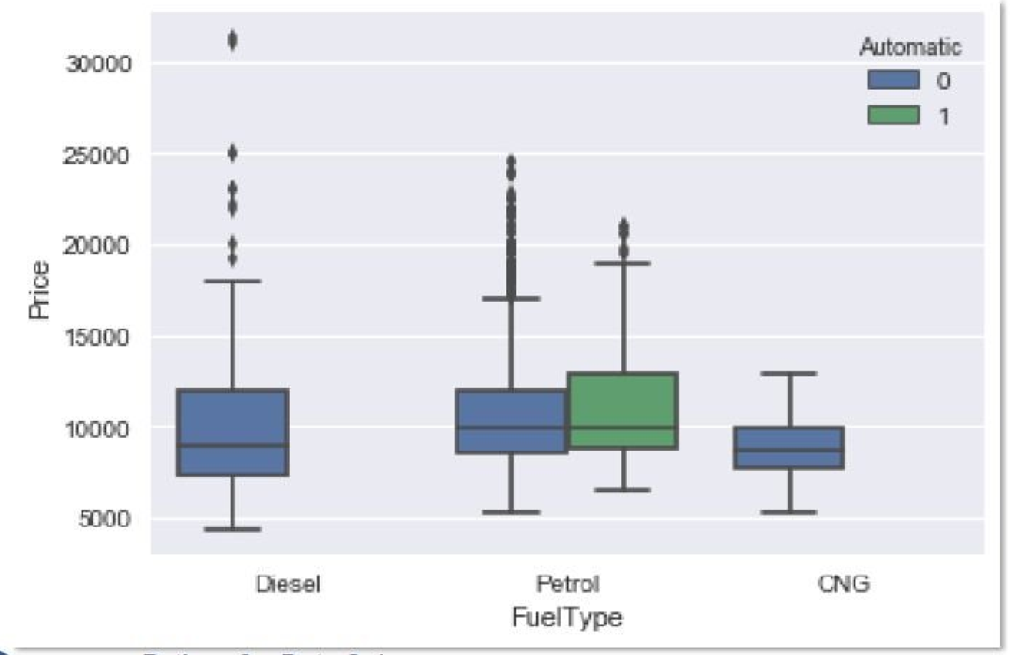
```
sns.boxplot(x = cars_data['FuelType'], y = cars_data["Price"])
```



Grouped box and whiskers plot

- Grouped box and whiskers plot of *Price* vs *FuelType* and *Automatic*

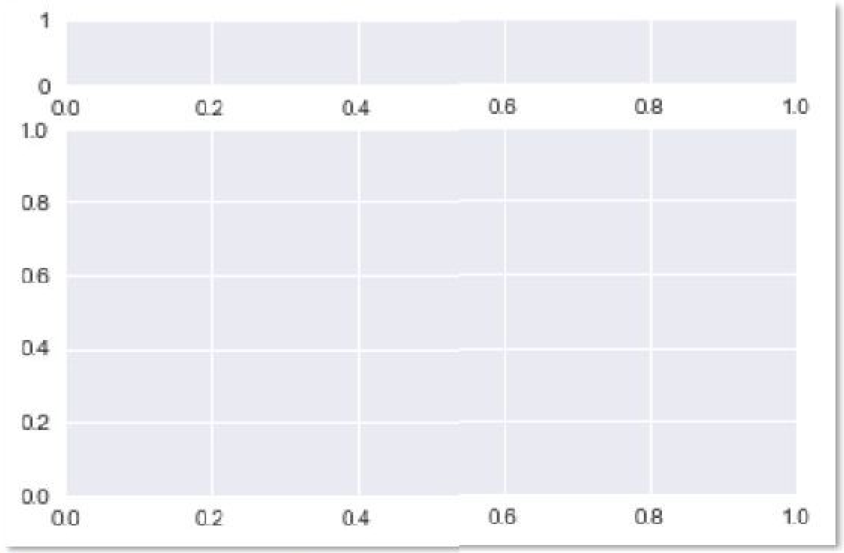
```
sns.boxplot(x = "FuelType",  y = cars_data["Price"],
            hue = "Automatic", data = cars_data)
```



Box-whiskers plot and Histogram

- Let's plot box-whiskers plot and histogram on the same window
- Split the plotting window into 2 parts

```
f,(ax_box, ax_hist)=plt.subplots(2, gridspec_kw={"height_ratios": (.15, .85)})
```

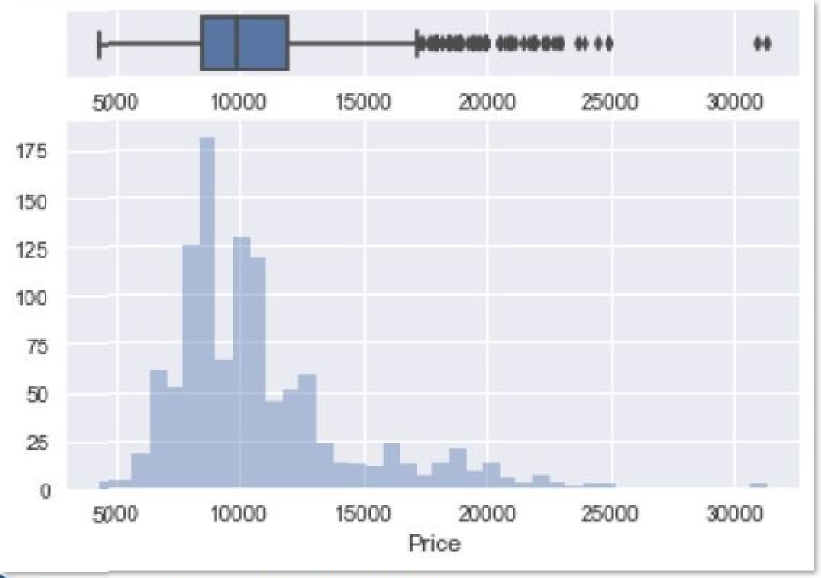


Box-whiskers plot and Histogram

- Now, add create two plots

```
sns.boxplot(cars_data["Price"] , ax=ax_box)

sns.distplot(cars_data["Price"], ax=ax_hist, kde = False)
```



Pairwise relationship using scatter plot and histogram

Code:

```
sns.pairplot(cars_data, kind="scatter", hue="FuelType")
plt.show()
```