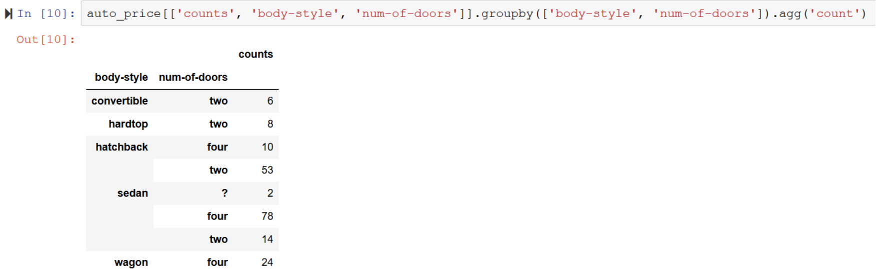
DS Certificate Course II

Notebook:	Data Science Certificate		
Created:	10/22/2019 11:00 AM	**Updated:**	11/5/2019 12:25 PM
Author:	nataliegmoore@gmail.com		
URL:	http://localhost:8888/notebooks/Desktop/DATA/UW/Course%20II/Labs/L01-DataExplor...		

Types of plots:

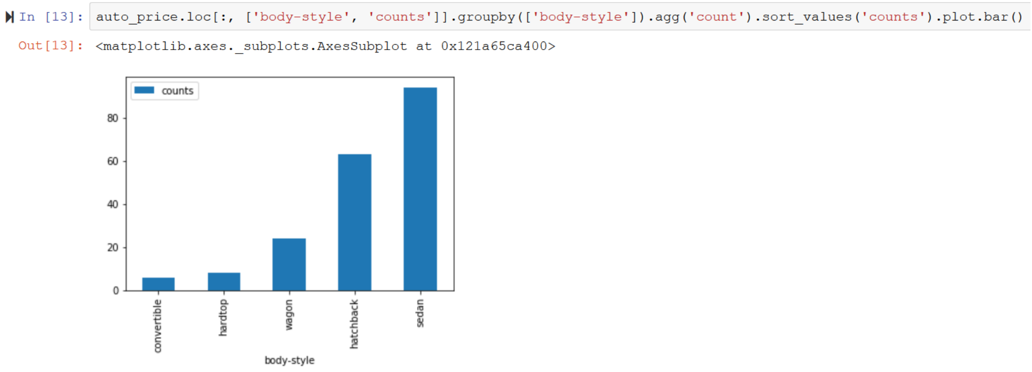
- **tables**:
  - df.describe()
  - df.shape
  - df.dtypes
  - df.head()
  - df.tail()
  - frequency tables:
    - df['counts'] = 1
      df[['counts', 'ColName']].groupby(['ColName']).agg('count')



```
In [10]: auto_price[['counts', 'body-style', 'num-of-doors']].groupby(['body-style', 'num-of-doors']).agg('count')
Out[10]:
```

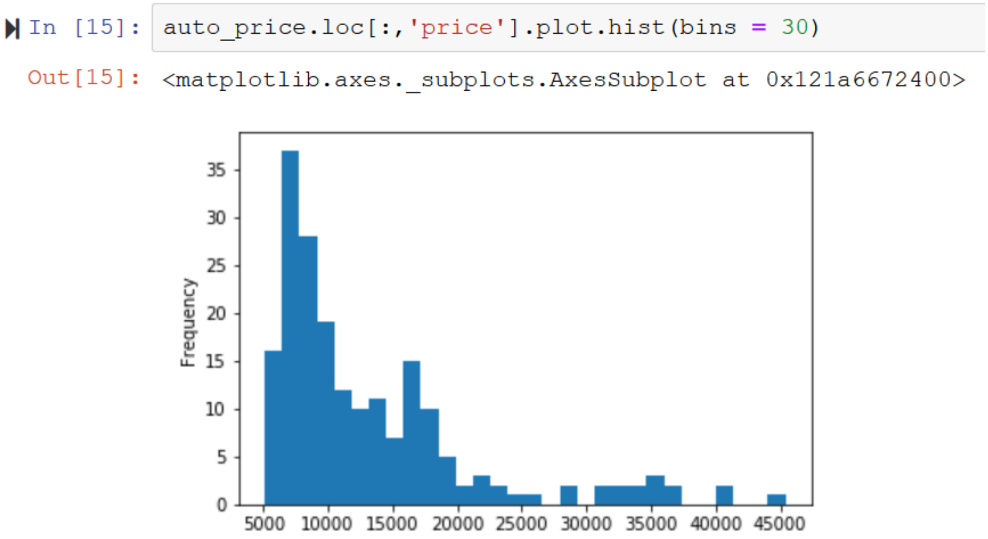
		counts
body-style	**num-of-doors**	
convertible	**two**	6
hardtop	**two**	8
hatchback	**four**	10
	two	53
sedan	**?**	2
	four	78
	two	14
wagon	**four**	24

- **Bar plots** - 1d



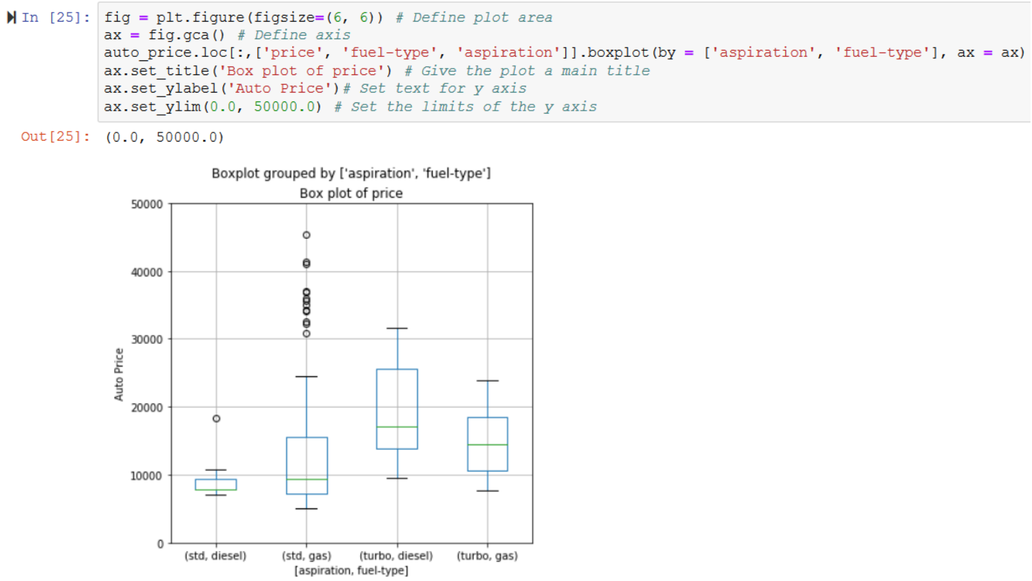
```
In [13]: auto_price.loc[:, ['body-style', 'counts']].groupby(['body-style']).agg('count').sort_values('counts').plot.bar()
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x121a65ca400>
```

- **Histograms** - 1d



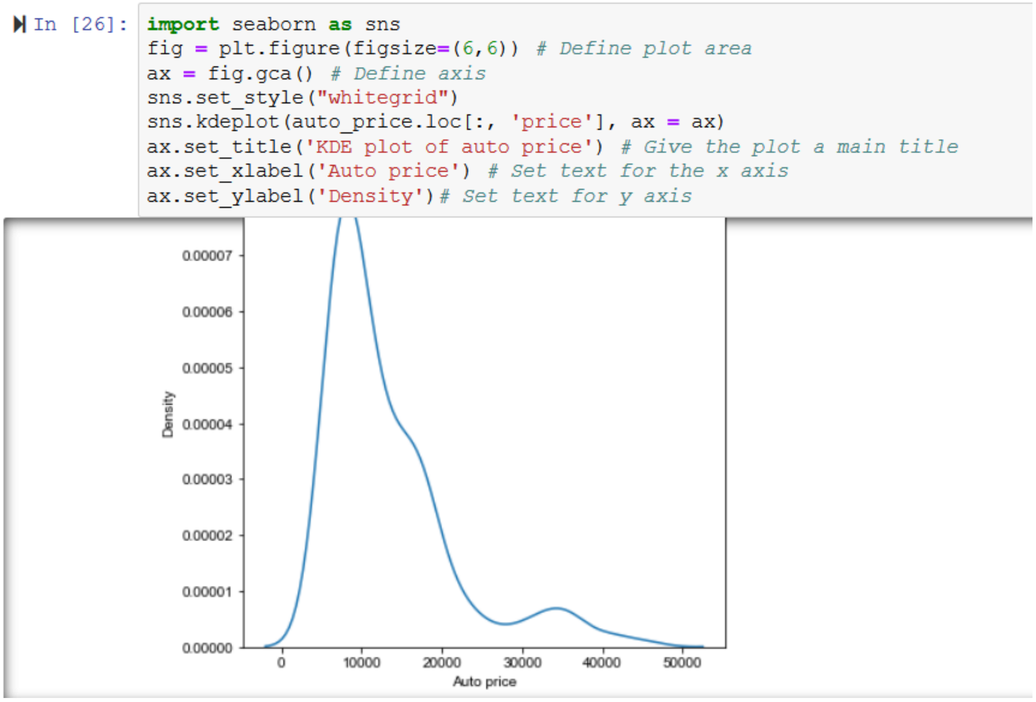
```
In [15]: auto_price.loc[:,'price'].plot.hist(bins = 30)

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x121a6672400>
```

- **Box plots** - 1d +



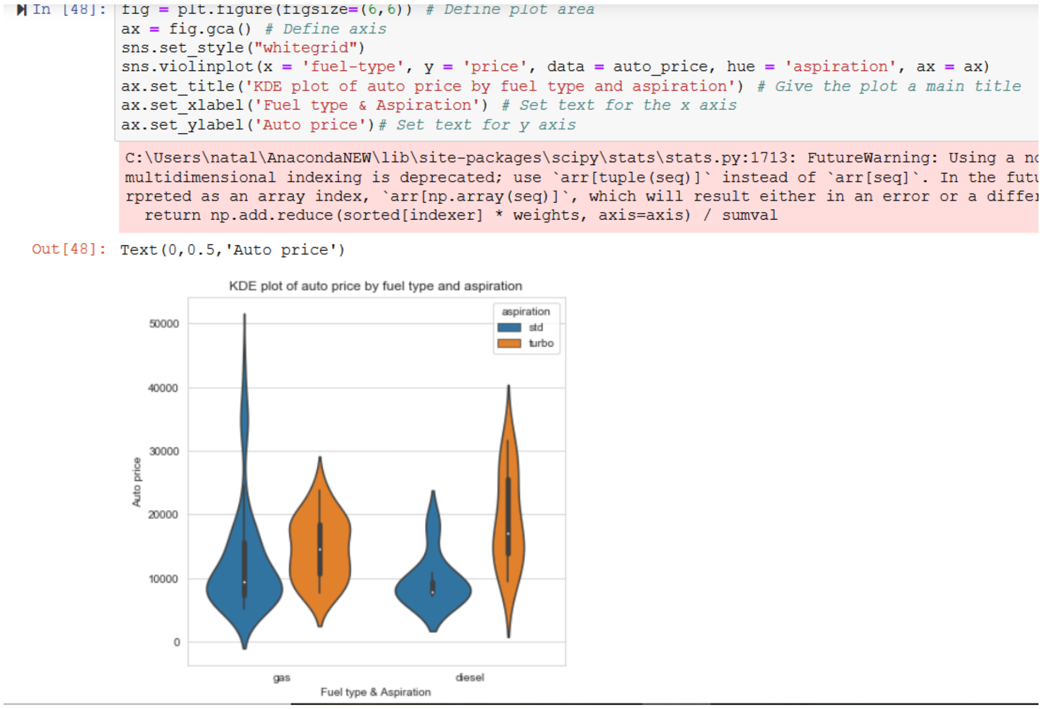
```
In [25]: fig = plt.figure(figsize=(6, 6)) # Define plot area
         ax = fig.gca() # Define axis
         auto_price.loc[:,['price', 'fuel-type', 'aspiration']].boxplot(by = ['aspiration', 'fuel-type'], ax = ax)
         ax.set_title('Box plot of price') # Give the plot a main title
         ax.set_ylabel('Auto Price')# Set text for y axis
         ax.set_ylim(0.0, 50000.0) # Set the limits of the y axis

Out[25]: (0.0, 50000.0)
```

- **Kernel Density Estimation Plots** - 1d +



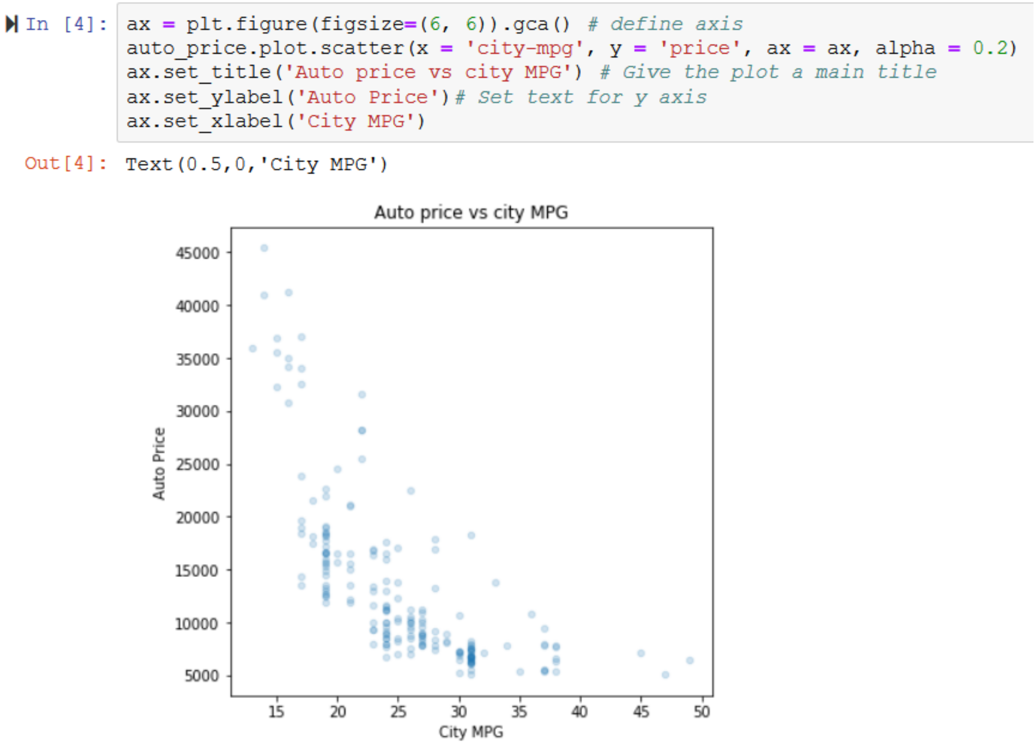
```
In [26]: import seaborn as sns
         fig = plt.figure(figsize=(6,6)) # Define plot area
         ax = fig.gca() # Define axis
         sns.set_style("whitegrid")
         sns.kdeplot(auto_price.loc[:, 'price'], ax = ax)
         ax.set_title('KDE plot of auto price') # Give the plot a main title
         ax.set_xlabel('Auto price') # Set text for the x axis
         ax.set_ylabel('Density')# Set text for y axis
```

- **Violin plots** -1d +



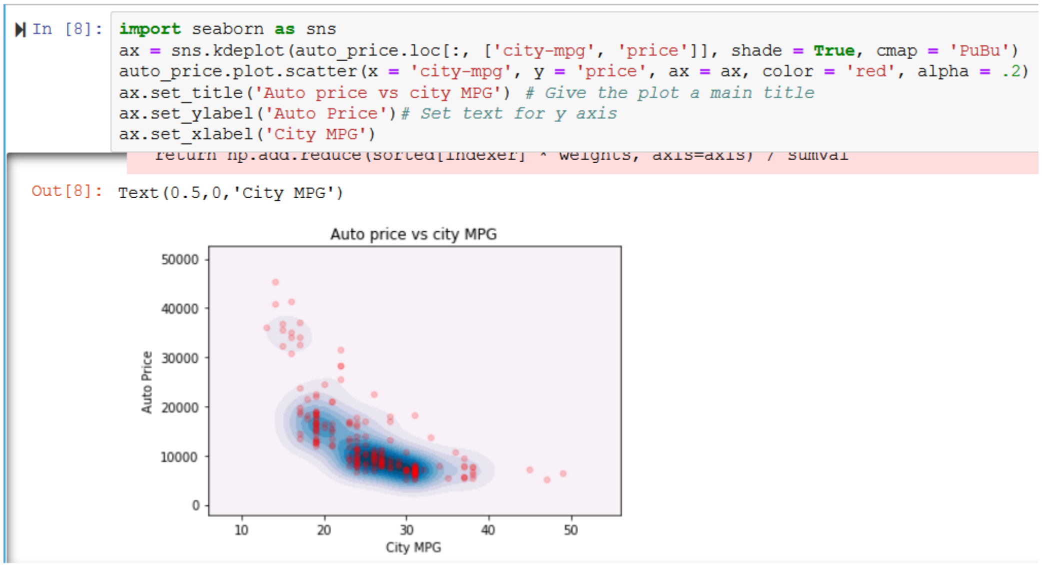
```
In [48]: fig = plt.figure(figsize=(6,6)) # Define plot area
         ax = fig.gca() # Define axis
         sns.set_style("whitegrid")
         sns.violinplot(x = 'fuel-type', y = 'price', data = auto_price, hue = 'aspiration', ax = ax)
         ax.set_title('KDE plot of auto price by fuel type and aspiration') # Give the plot a main title
         ax.set_xlabel('Fuel type & Aspiration') # Set text for the x axis
         ax.set_ylabel('Auto price')# Set text for y axis

         C:\Users\natal\AnacondaNEW\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a no
         multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futu
         rpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a differ
           return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Out[48]: Text(0,0.5,'Auto price')
```

- **Scatter plot** - 2d



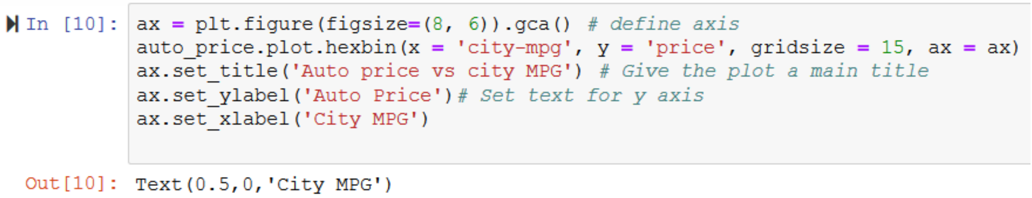
```
In [4]: ax = plt.figure(figsize=(6, 6)).gca() # define axis
        auto_price.plot.scatter(x = 'city-mpg', y = 'price', ax = ax, alpha = 0.2)
        ax.set_title('Auto price vs city MPG') # Give the plot a main title
        ax.set_ylabel('Auto Price')# Set text for y axis
        ax.set_xlabel('City MPG')

Out[4]: Text(0.5,0,'City MPG')
```

- **2d Kernel Density Plots** -2d

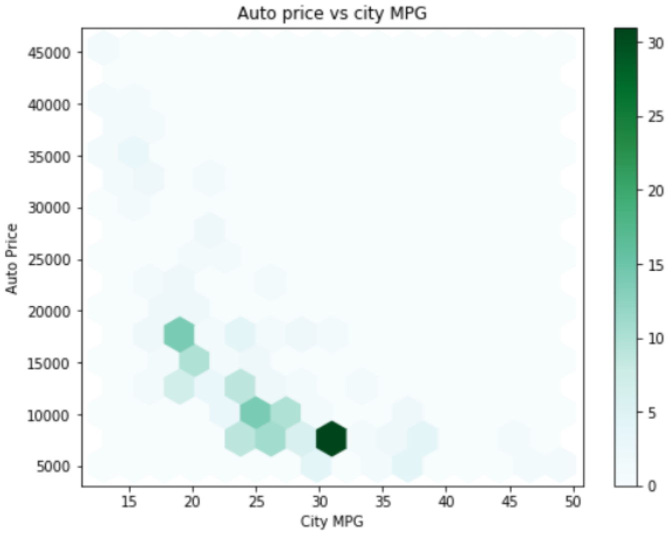


```
In [8]: import seaborn as sns
        ax = sns.kdeplot(auto_price.loc[:, ['city-mpg', 'price']], shade = True, cmap = 'PuBu')
        auto_price.plot.scatter(x = 'city-mpg', y = 'price', ax = ax, color = 'red', alpha = .2)
        ax.set_title('Auto price vs city MPG') # Give the plot a main title
        ax.set_ylabel('Auto Price')# Set text for y axis
        ax.set_xlabel('City MPG')

  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Out[8]: Text(0.5,0,'City MPG')
```

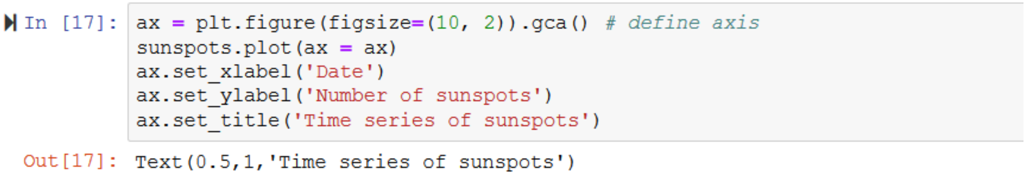
- **Hexbin plots** - 2d



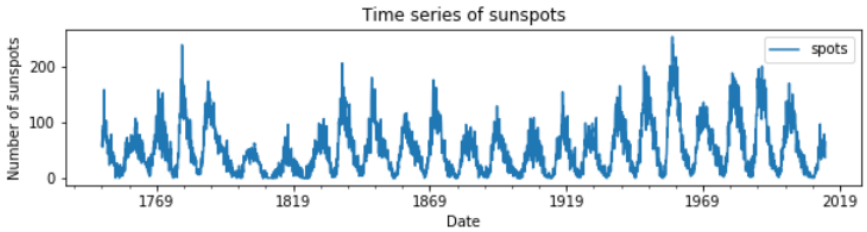
```
In [10]: ax = plt.figure(figsize=(8, 6)).gca() # define axis
         auto_price.plot.hexbin(x = 'city-mpg', y = 'price', gridsize = 15, ax = ax)
         ax.set_title('Auto price vs city MPG') # Give the plot a main title
         ax.set_ylabel('Auto Price')# Set text for y axis
         ax.set_xlabel('City MPG')

Out[10]: Text(0.5,0,'City MPG')
```



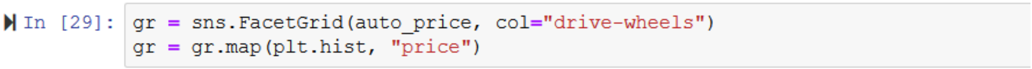
- **Line plots** - 2d



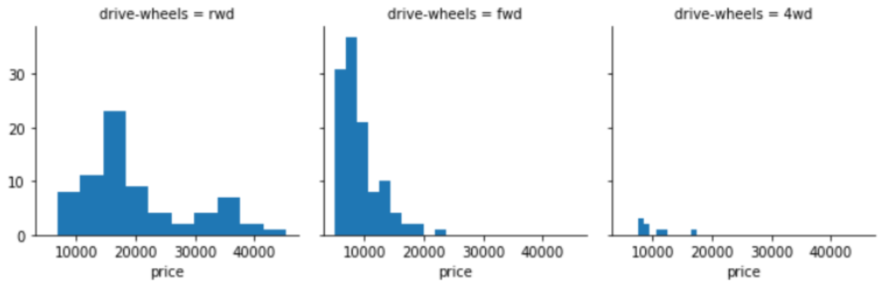
```
In [17]: ax = plt.figure(figsize=(10, 2)).gca() # define axis
         sunspots.plot(ax = ax)
         ax.set_xlabel('Date')
         ax.set_ylabel('Number of sunspots')
         ax.set_title('Time series of sunspots')

Out[17]: Text(0.5,1,'Time series of sunspots')
```

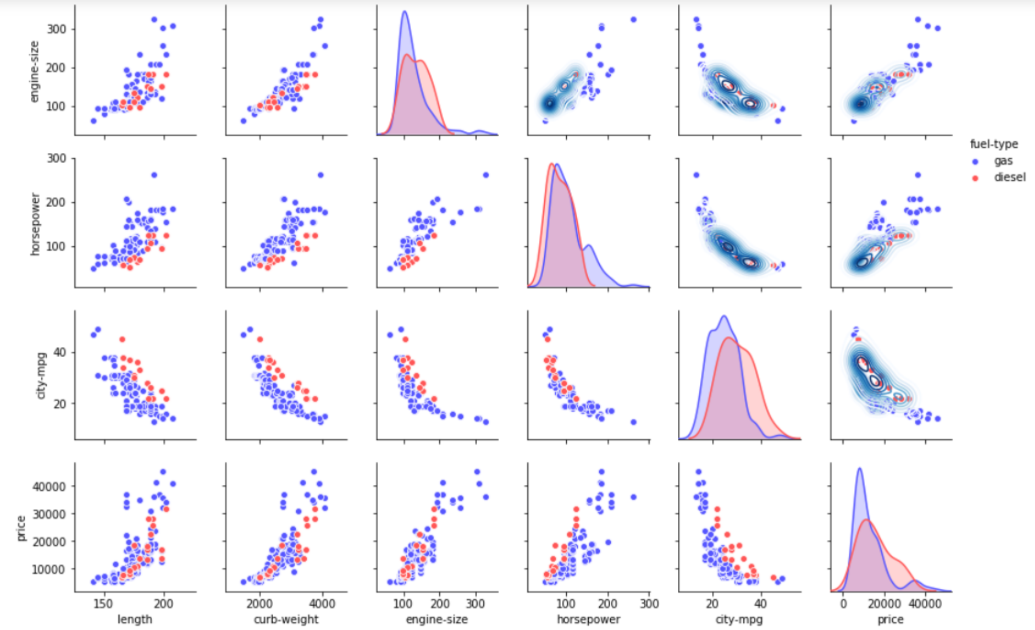


- **Facet plots/Pair plots**



```
In [29]: gr = sns.FacetGrid(auto_price, col="drive-wheels")
         gr = gr.map(plt.hist, "price")
```




engine-size
300
200
100
horsepower
300
200
100
city-mpg
40
20
price
40000
30000
20000
10000
150
200
length
2000
4000
curb-weight
100
200
300
engine-size
100
200
300
horsepower
20
40
city-mpg
0
20000
40000
price
fuel-type
gas
diesel