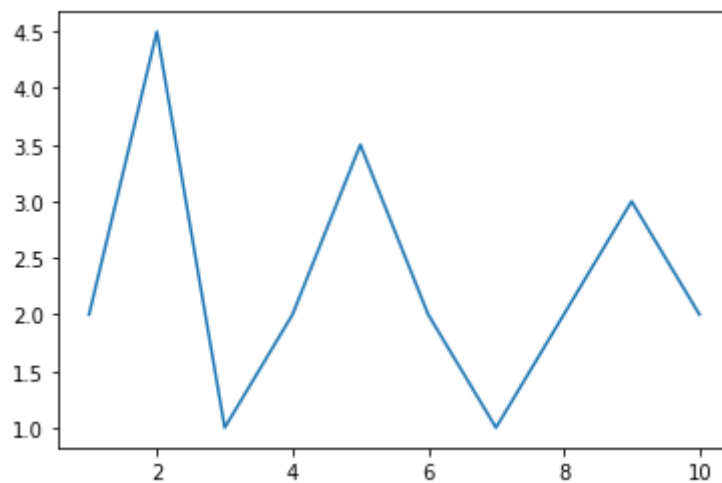


Plotting Line Charts

```
In [5]: %matplotlib inline
import matplotlib.pyplot as plt

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2]
)
```

Out[5]: [

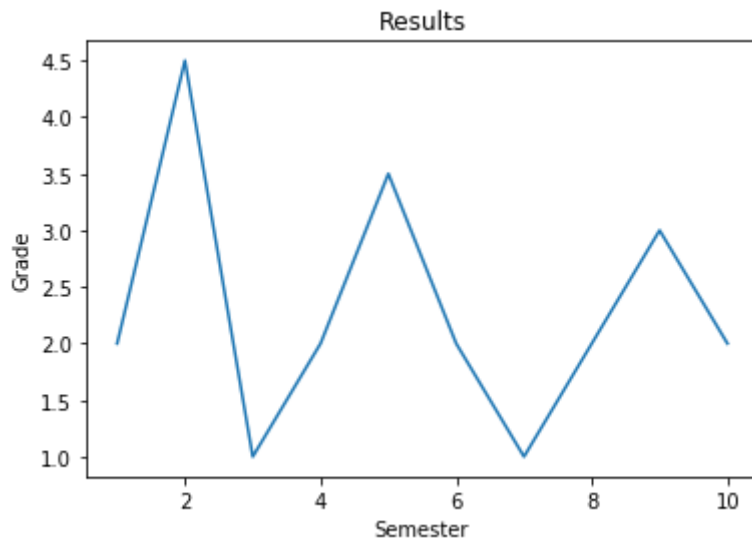


Adding Title and Labels

```
In [2]: %matplotlib inline
import matplotlib.pyplot as plt

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2]
)
plt.title("Results")           # sets the title for the chart
plt.xlabel("Semester")         # sets the label to use for the x-axis
plt.ylabel("Grade")           # sets the label to use for the y-axis
```

Out[2]: Text(0,0.5,'Grade')



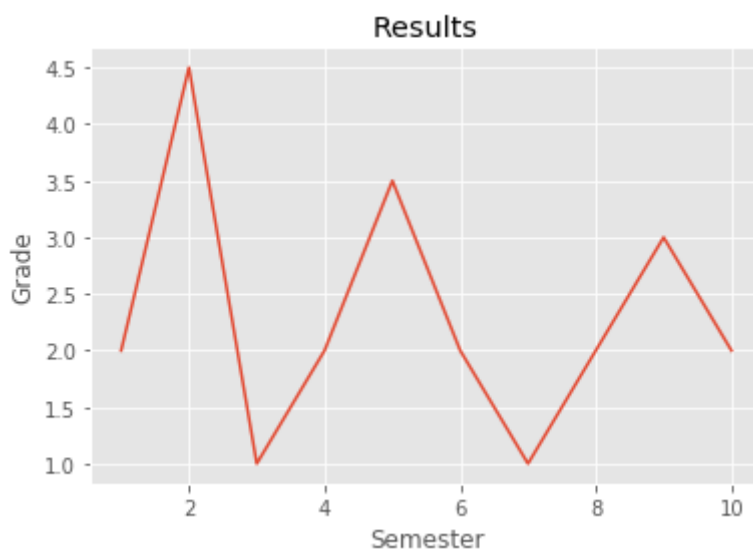
Styling

```
In [3]: %matplotlib inline
import matplotlib.pyplot as plt

from matplotlib import style
style.use("ggplot")

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2]
)
plt.title("Results")           # sets the title for the chart
plt.xlabel("Semester")        # sets the label to use for the x-axis
plt.ylabel("Grade")           # sets the label to use for the y-axis
```

Out[3]: Text(0,0.5,'Grade')



```
In [4]: print(style.available)
```

```
['seaborn-dark', 'seaborn-darkgrid', 'seaborn-ticks', 'fivethirtyeight',
'seaborn-whitegrid', 'classic', '_classic_test', 'fast', 'seaborn-talk',
'seaborn-dark-palette', 'seaborn-bright', 'seaborn-pastel', 'grayscale',
'seaborn-notebook', 'ggplot', 'seaborn-colorblind', 'seaborn-muted', 'sea
born', 'Solarize_Light2', 'seaborn-paper', 'bmh', 'tableau-colorblind10',
'seaborn-white', 'dark_background', 'seaborn-poster', 'seaborn-deep']
```

Plotting Multiple Lines in the Same Chart

```
In [5]: %matplotlib inline
import matplotlib.pyplot as plt

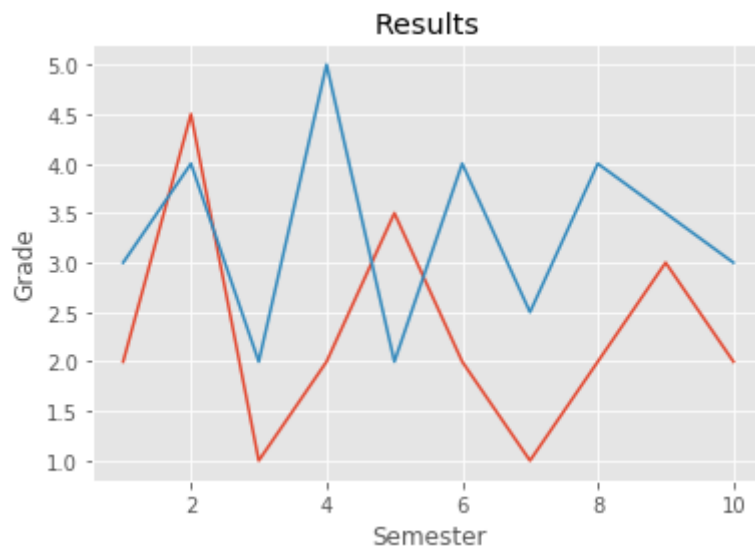
from matplotlib import style
style.use("ggplot")

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2]
)

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [3,4,2,5,2,4,2.5,4,3.5,3]
)

plt.title("Results")           # sets the title for the chart
plt.xlabel("Semester")         # sets the label to use for the x-axis
plt.ylabel("Grade")            # sets the label to use for the y-axis
```

Out[5]: Text(0,0.5,'Grade')



Adding a Legend

```

In [6]: %matplotlib inline
import matplotlib.pyplot as plt

from matplotlib import style
style.use("ggplot")

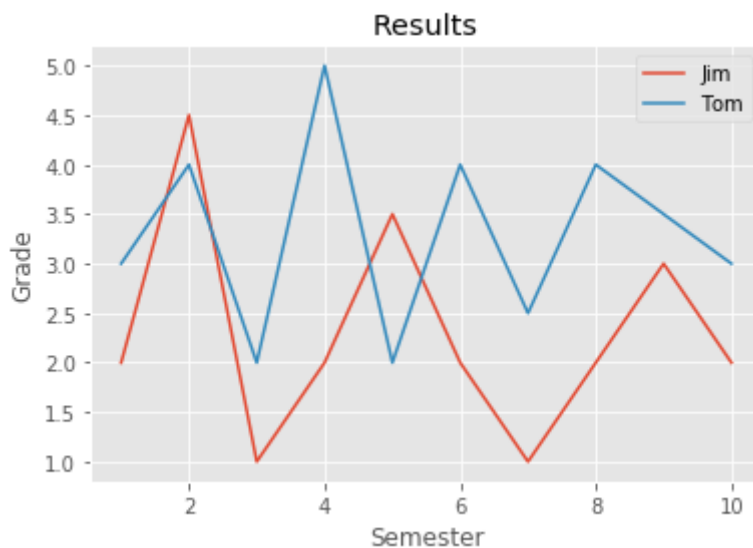
plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2],
    label="Jim"
)

plt.plot(
    [1,2,3,4,5,6,7,8,9,10],
    [3,4,2,5,2,4,2.5,4,3.5,3],
    label="Tom"
)

plt.title("Results")           # sets the title for the chart
plt.xlabel("Semester")        # sets the label to use for the x-axis
plt.ylabel("Grade")           # sets the label to use for the y-axis
plt.legend()

```

Out[6]: <matplotlib.legend.Legend at 0x11d58e898>



Plotting Bar Charts

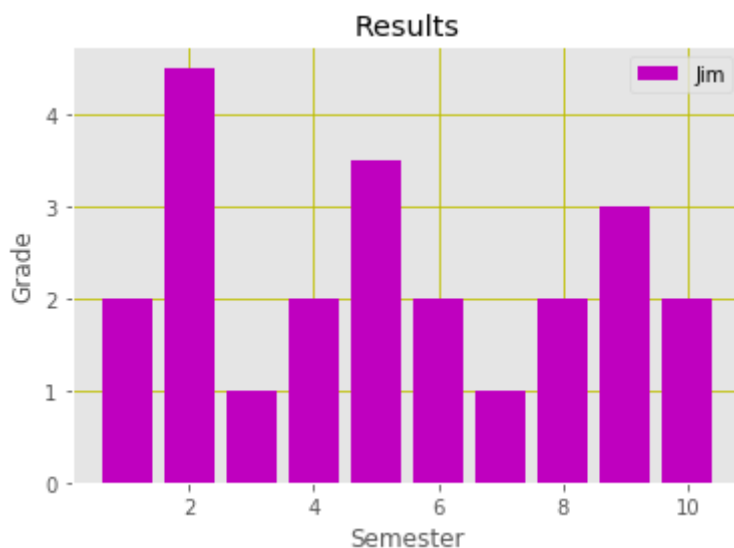
```
In [7]: %matplotlib inline
import matplotlib.pyplot as plt
from matplotlib import style

style.use("ggplot")

plt.bar(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2],
    label = "Jim",
    color = "m",                               # m for magenta
    align = "center"
)

plt.title("Results")
plt.xlabel("Semester")
plt.ylabel("Grade")

plt.legend()
plt.grid(True, color="y")
```



Adding Another Bar to the Chart

```

In [8]: %matplotlib inline
import matplotlib.pyplot as plt
from matplotlib import style

style.use("ggplot")

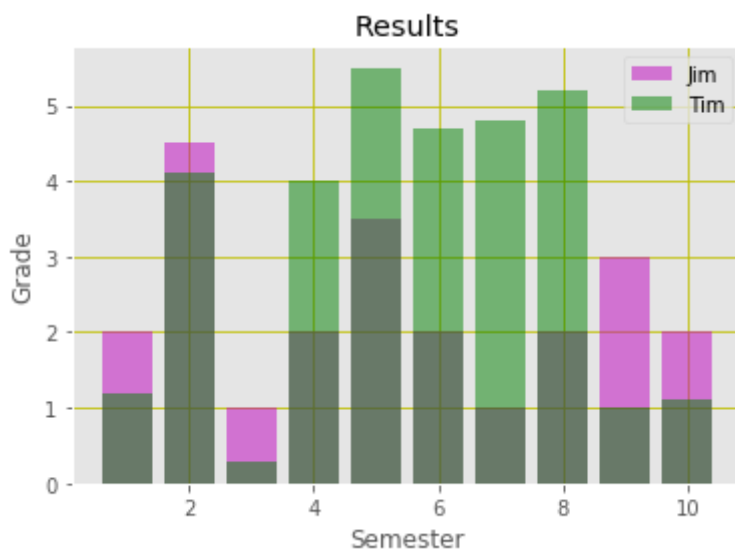
plt.bar(
    [1,2,3,4,5,6,7,8,9,10],
    [2,4.5,1,2,3.5,2,1,2,3,2],
    label = "Jim",
    color = "m",
    align = "center",
    alpha = 0.5
)

plt.bar(
    [1,2,3,4,5,6,7,8,9,10],
    [1.2,4.1,0.3,4,5.5,4.7,4.8,5.2,1,1.1],
    label = "Tim",
    color = "g",
    align = "center",
    alpha = 0.5
)

plt.title("Results")
plt.xlabel("Semester")
plt.ylabel("Grade")

plt.legend()
plt.grid(True, color="y")

```



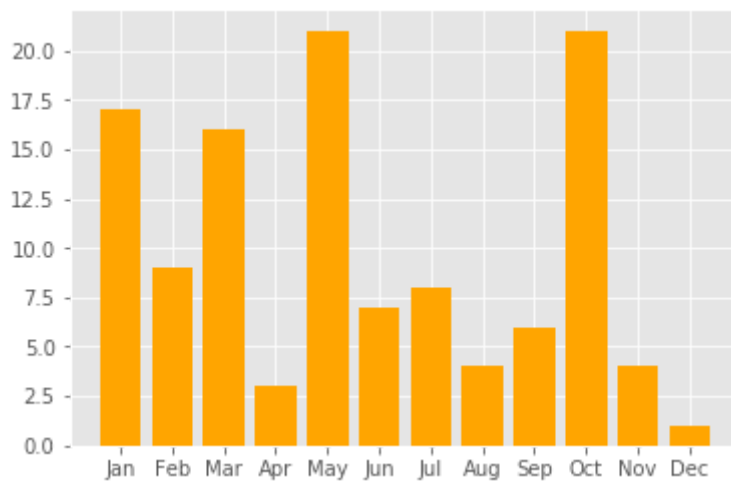
Changing the Tick Marks

```
In [9]: rainfall = [17,9,16,3,21,7,8,4,6,21,4,1]
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
```

```
In [10]: %matplotlib inline
import matplotlib.pyplot as plt

rainfall = [17,9,16,3,21,7,8,4,6,21,4,1]
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

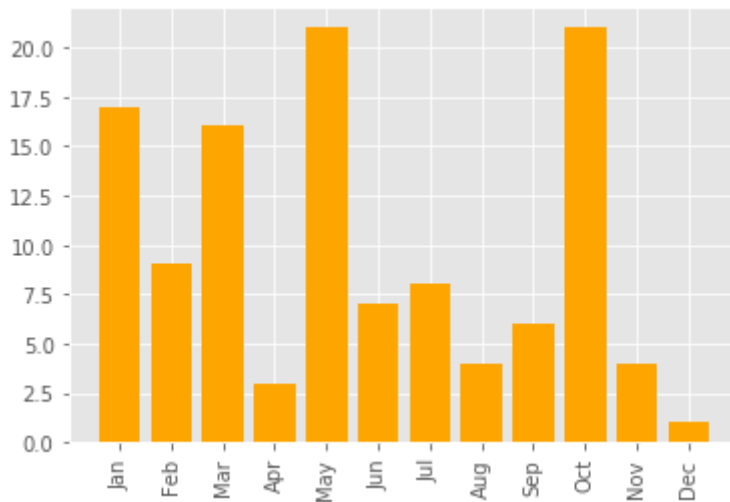
plt.bar(months, rainfall, align='center', color='orange' )
plt.show()
```




```
In [11]: %matplotlib inline
import matplotlib.pyplot as plt

rainfall = [17,9,16,3,21,7,8,4,6,21,4,1]
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

plt.bar(range(len(rainfall)), rainfall, align='center', color='orange' )
plt.xticks(range(len(rainfall)), months, rotation='vertical')
plt.show()
```



Plotting Pie Charts

```
In [12]: labels = ["Chrome", "Internet Explorer", "Firefox", "Edge", "Safari",
                  "Sogou Explorer", "Opera", "Others"]
marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
```

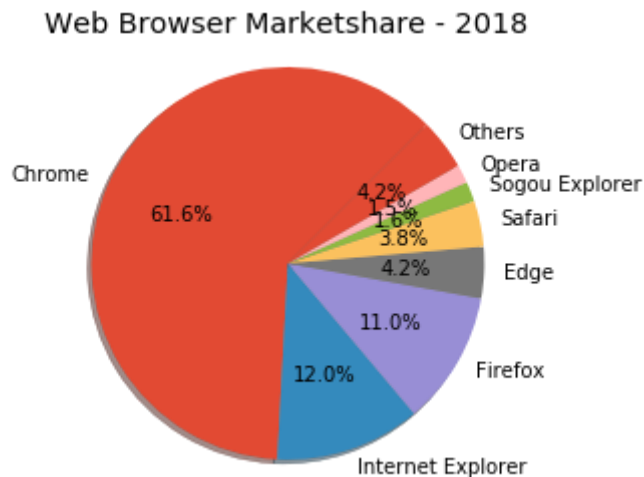
```
In [13]: %matplotlib inline
import matplotlib.pyplot as plt

labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0,0,0,0,0,0)

plt.pie(marketshare,
        explode = explode, # fraction of the radius with which to offset each slice
        labels = labels,
        autopct="%.1f%%", # string or function used to label the wedges with their numeric value
        shadow=True,
        startangle=45)    # rotates the start of the pie chart by angle counter-clockwise from the x-axis

plt.axis("equal")        # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.show()
```



Exploding the Slices

```

In [14]: %matplotlib inline
import matplotlib.pyplot as plt

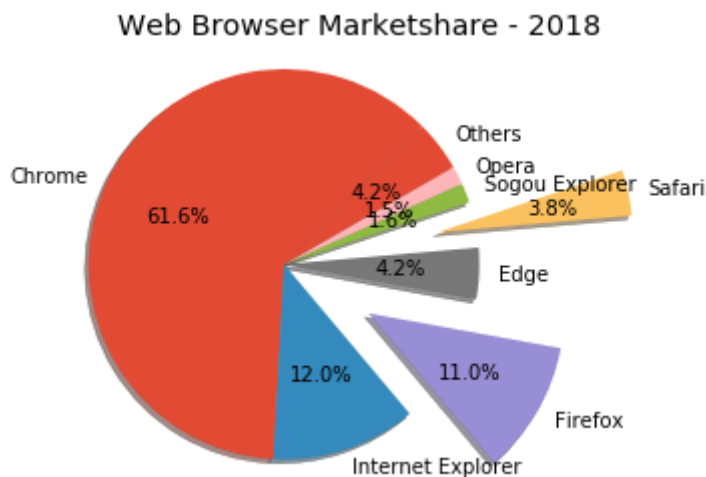
labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0.5,0,0.8,0,0,0)

plt.pie(marketshare,
        explode = explode, # fraction of the radius with which to offset each wedge
        labels = labels,
        autopct="%0.1f%%", # string or function used to label the wedges with their numeric value
        shadow=True,
        startangle=45) # rotates the start of the pie chart by angle degrees counterclockwise from the x-axis

plt.axis("equal") # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.show()

```



Displaying Custom Colors

```

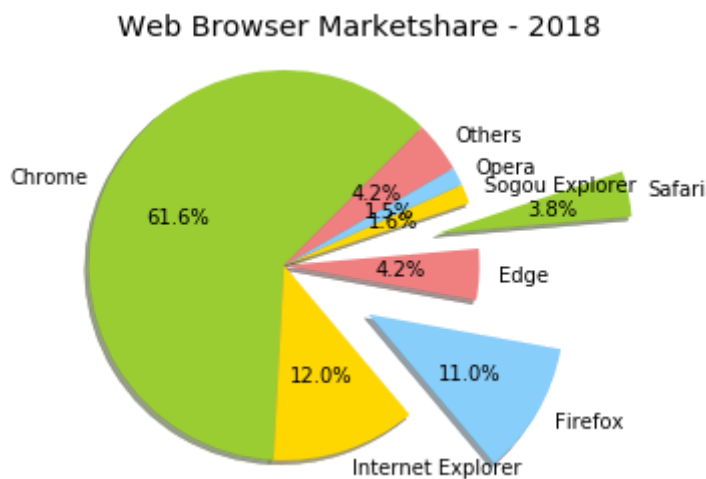
In [15]: %matplotlib inline
import matplotlib.pyplot as plt

labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0.5,0,0.8,0,0,0)
colors      = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']

plt.pie(marketshare,
        explode = explode, # fraction of the radius with which to offset e
        labels = labels,
        colors = colors,
        autopct="%0.1f%%", # string or function used to label the wedges w
                           # their numeric value
        shadow=True,
        startangle=45)     # rotates the start of the pie chart by angle c
                           # counterclockwise from the x-axis
plt.axis("equal")         # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.show()

```



Rotating the Pie Chart

```

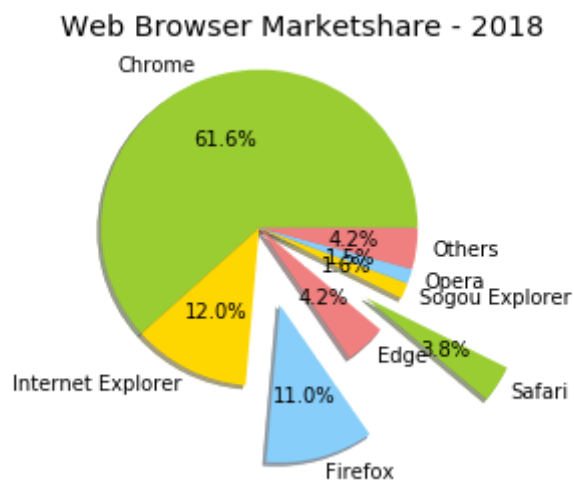
In [16]: %matplotlib inline
import matplotlib.pyplot as plt

labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0.5,0,0.8,0,0,0)
colors      = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']

plt.pie(marketshare,
        explode = explode, # fraction of the radius with which to offset each wedge
        labels = labels,
        colors = colors,
        autopct="%.1f%%", # string or function used to label the wedges with their numeric value
        shadow=True,
        startangle=0)      # rotates the start of the pie chart by angle counter-clockwise from the x-axis
plt.axis("equal")         # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.show()

```



Displaying a Legend

```

In [8]: %matplotlib inline
import matplotlib.pyplot as plt

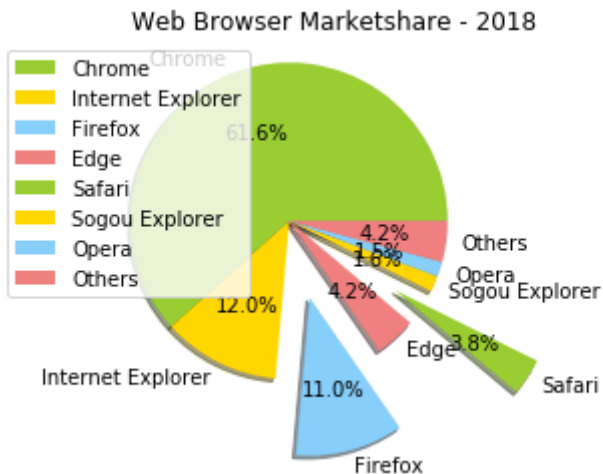
labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0.5,0,0.8,0,0,0)
colors      = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']

pie = plt.pie(marketshare,
              explode = explode, # fraction of the radius with which to offset e
              labels = labels,
              colors = colors,
              autopct="%0.1f%%", # string or function used to label the wedges w
                                # their numeric value

              shadow=True,
              startangle=0)      # rotates the start of the pie chart by angle de
                                # counterclockwise from the x-axis
plt.axis("equal")               # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.legend(pie[0], labels, loc="best")
plt.show()

```



Saving the Chart

```

In [9]: %matplotlib inline
import matplotlib.pyplot as plt

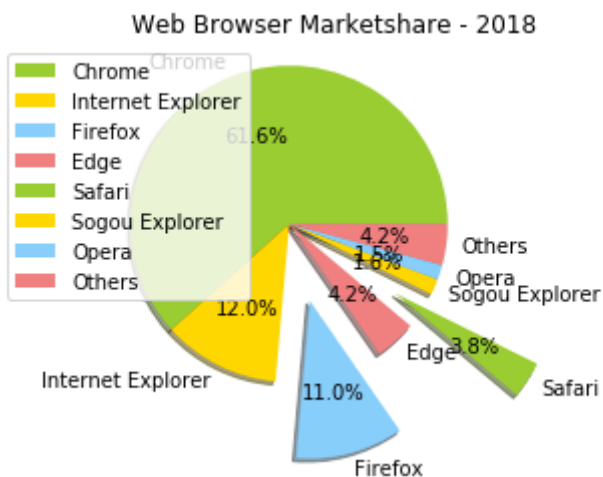
labels      = ["Chrome", "Internet Explorer",
               "Firefox", "Edge", "Safari",
               "Sogou Explorer", "Opera", "Others"]

marketshare = [61.64, 11.98, 11.02, 4.23, 3.79, 1.63, 1.52, 4.19]
explode     = (0,0,0.5,0,0.8,0,0,0)
colors      = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']

pie = plt.pie(marketshare,
              explode = explode, # fraction of the radius with which to offset e
              labels = labels,
              colors = colors,
              autopct="%.1f%%", # string or function used to label the wedges w
                                # their numeric value

              shadow=True,
              startangle=0)     # rotates the start of the pie chart by angle de
                                # counterclockwise from the x-axis
plt.axis("equal")              # turns off the axis lines and labels
plt.title("Web Browser Marketshare - 2018")
plt.savefig("Webrowsers.png", bbox_inches="tight")
plt.legend(pie[0], labels, loc="best")
plt.show()

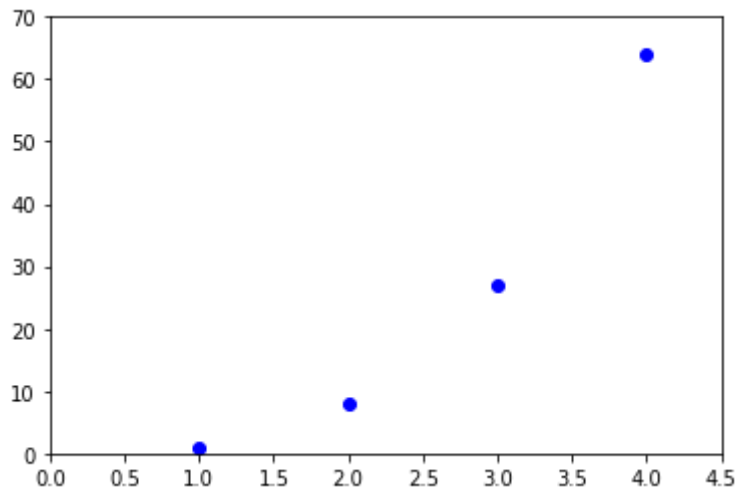
```



Plotting Scatter Plots

```
In [12]: %matplotlib inline
import matplotlib.pyplot as plt

plt.plot([1,2,3,4],          # x-axis
         [1,8,27,64],       # y-axis
         'bo')              # blue circle marker
plt.axis([0, 4.5, 0, 70])   # xmin, xmax, ymin, ymax
plt.show()
```



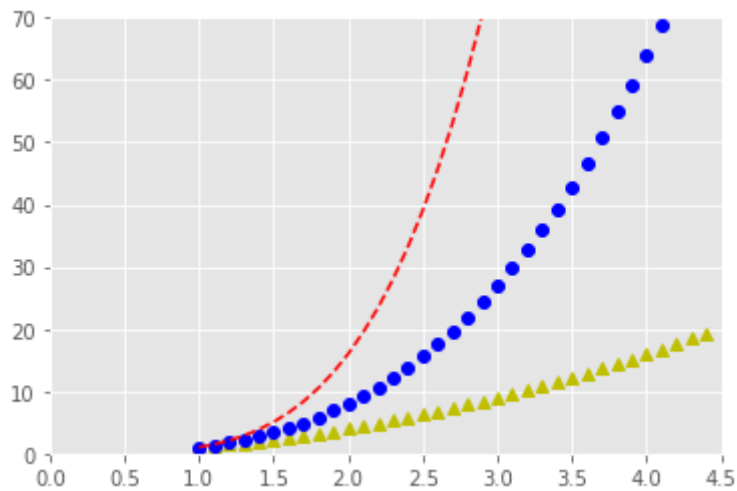
Combining Plots


```
In [20]: %matplotlib inline
import matplotlib.pyplot as plt

import numpy as np

a = np.arange(1,4.5,0.1)    # 1.0, 1.1, 1.2, 1.3...4.4
plt.plot(a, a**2, 'y^',    # yellow triangle_up marker
         a, a**3, 'bo',    # blue circle
         a, a**4, 'r--',)  # red dashed line

plt.axis([0, 4.5, 0, 70])  # xmin, xmax, ymin, ymax
plt.show()
```



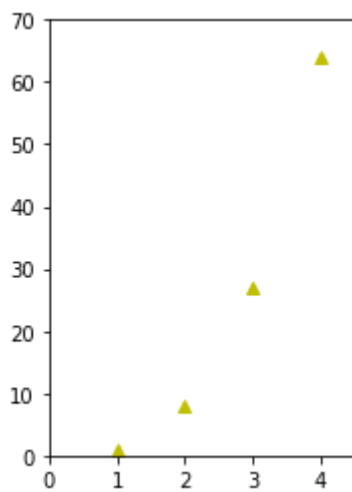
Subplots

```
In [13]: %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

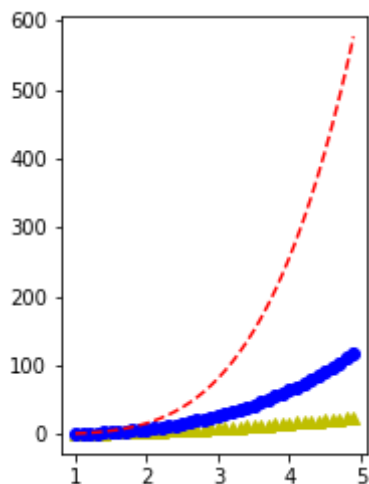
a = np.arange(1,5,0.1)

plt.subplot(121)                                # 1 row, 2 cols, chart 1
plt.plot([1,2,3,4,5],
         [1,8,27,64,125],
         'y^')

plt.axis([0, 4.5, 0, 70])    # xmin, xmax, ymin, ymax
plt.show()
```



```
In [15]: plt.subplot(122)                        # 1 row, 2 cols, chart 2
plt.plot(a, a**2, 'y^',
         a, a**3, 'bo',
         a, a**4, 'r--',)
plt.show()
```



Plotting Using Seaborn

```
In [22]: import seaborn as sns  
sns.__version__
```

```
Out[22]: '0.9.0'
```

You need Seaborn 0.9.0 for catplot. To install Seaborn 0.9.0, type this in Terminal/Anaconda Prompt:

```
sudo -H pip install seaborn==0.9.0
```

Then, restart Jupyter Notebook.

Displaying Categorical Plots

The first example that you will plot is called a categorical plot (formerly known as a factorplot). It is useful in cases when you want to plot the distribution of a certain group of data.

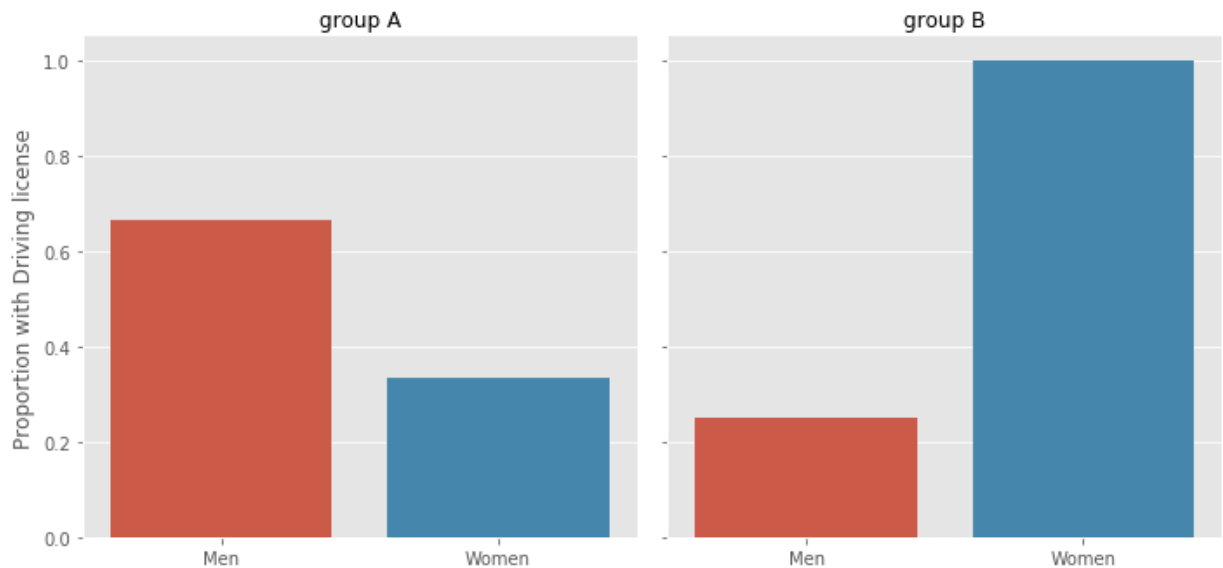
```
In [23]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

#---load data---
data = pd.read_csv('drivinglicense.csv')

#---plot a factorplot---
g = sns.catplot(x="gender", y="license", col="group", data=data, kind="bar")

#---set the labels---
g.set_axis_labels("", "Proportion with Driving license")
g.set_xticklabels(["Men", "Women"])
g.set_titles("{col_var} {col_name}")

#---show plot---
plt.show()
```

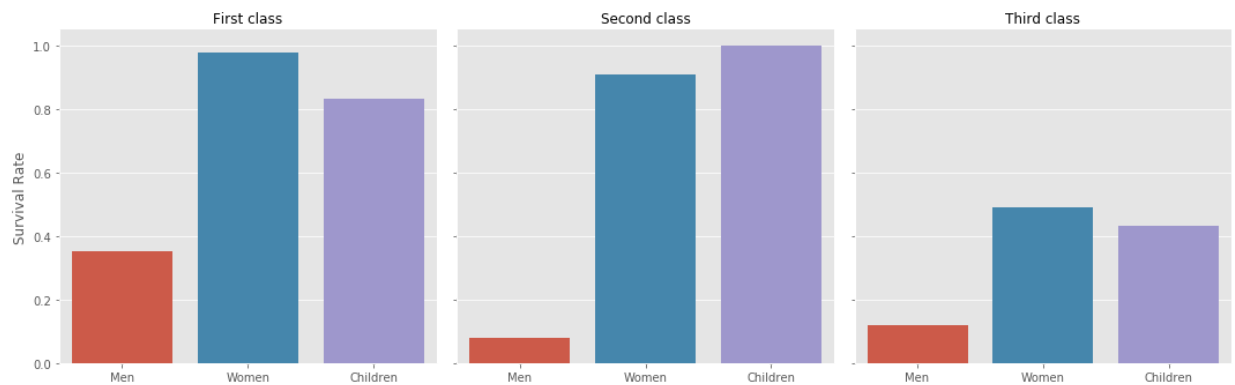


```
In [24]: import matplotlib.pyplot as plt
import seaborn as sns

titanic = sns.load_dataset("titanic")
g = sns.catplot(x="who", y="survived", col="class",
                data=titanic, kind="bar", ci=None, aspect=1)

g.set_axis_labels("", "Survival Rate")
g.set_xticklabels(["Men", "Women", "Children"])
g.set_titles("{col_name} {col_var}")

#---show plot---
plt.show()
```



Displaying Lmplots

Implot is a scatter plot

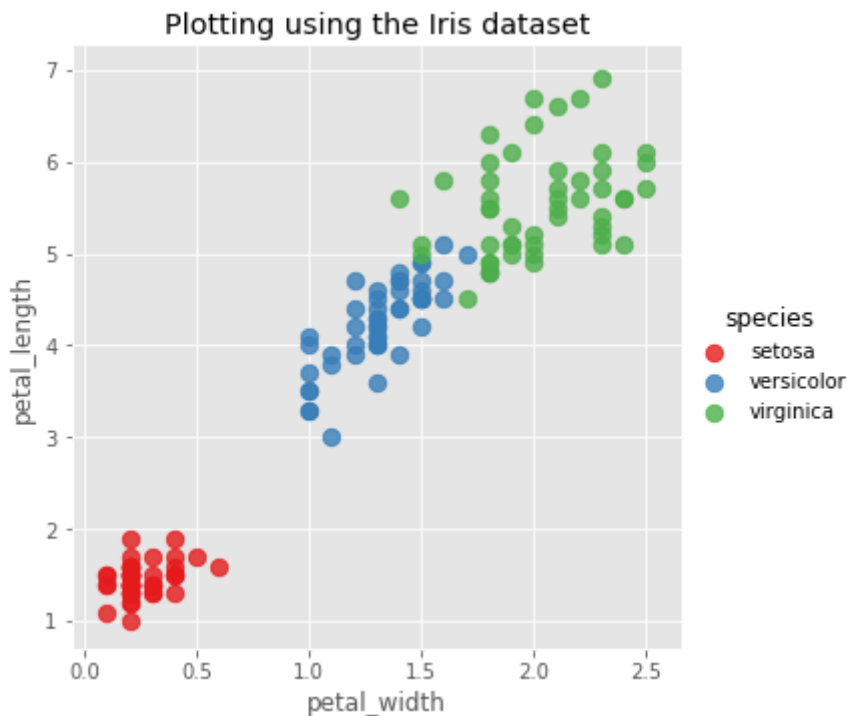
```
In [25]: import seaborn as sns
import matplotlib.pyplot as plt

#---load the iris dataset---
iris = sns.load_dataset("iris")

#---plot the lmpot---
sns.lmplot('petal_width', 'petal_length', data=iris,
           hue='species', palette='Set1',
           fit_reg=False, scatter_kws={"s": 70})

#---get the current polar axes on the current figure---
ax = plt.gca()
ax.set_title("Plotting using the Iris dataset")

#---show the plot---
plt.show()
```



Displaying Swarmplots

A swarmplot is a categorical scatterplot with nonoverlapping points

```
In [26]: import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

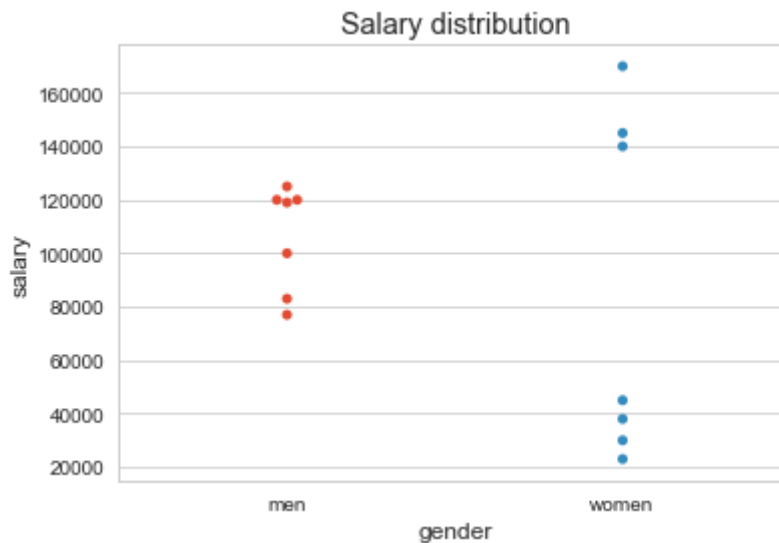
sns.set_style("whitegrid")

#---load data---
data = pd.read_csv('salary.csv')

#---plot the swarm plot---
sns.swarmplot(x="gender", y="salary", data=data)

ax = plt.gca()
ax.set_title("Salary distribution")

#---show plot---
plt.show()
```



```
In [ ]:
```