CIS-2266 Matplotlib Dropbox

CIS-2266 Matplotlib Exercise

- Complete each exercise and ensure each cell has output
- Print to PDF and submit to the dropbox

Orginial Notebook "Making basic plots - Lines, bars, pies, and scatterplots" by Randal S. Olson (http://www.randalolson.com/)

In [20]:

```
# Run this to get started (no output unless error)

import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

Line plots

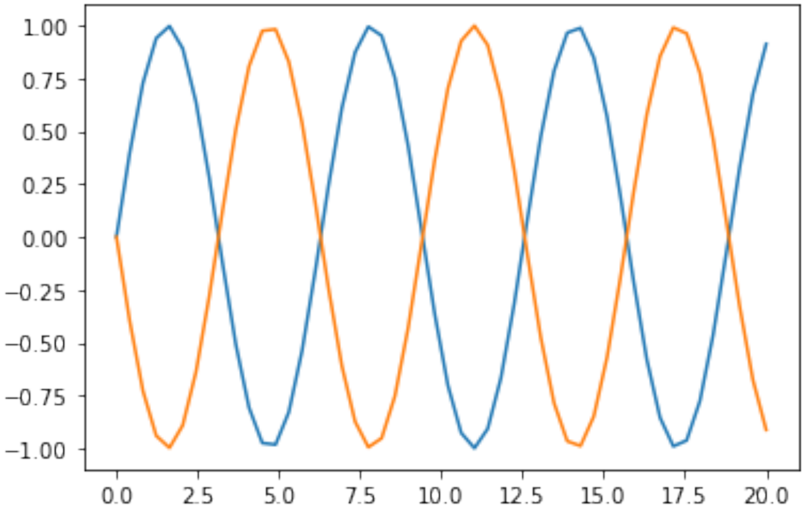
Generate a lineplot of y1 and y2 on x axis

In [21]:
```
# Using x, y1 and y2
# Generate of y1 and y2 on x axis
x = np.linspace(0, 20)
y1 = np.sin(x)
y2 = np.sin(x - np.pi)

#Code goes below:

plt.plot(x,y1)
plt.plot(x,y2)

plt.show()
```



Color and style

Create same plot above with:

- y1 color is black, linestyle solid, linewidth of 2, squar marker at a size of 6

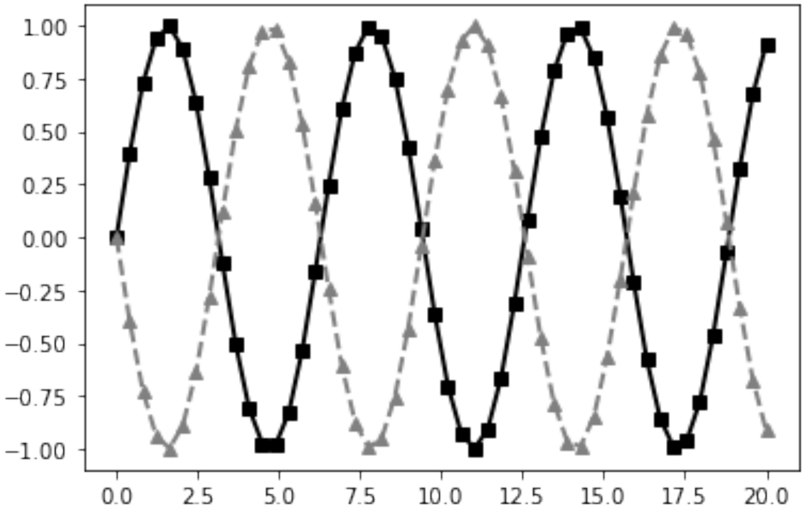
- y2 color is gray, linestyle dashed, linewidth of 2, triangle markers at size of 6

In [22]:

```
# Using:
x = np.linspace(0, 20)
y1 = np.sin(x)
y2 = np.sin(x - np.pi)

# Create same plot above with:
# - y1 color is black, linestyle solid, linewidth of 2, squar marker a
# - y2 color is gray, linestyle dashed, linewidth of 2, triangle marke

# Code goes below
plt.plot(x,y1, color = 'black',linestyle='solid', linewidth = 2, marke
plt.plot(x,y2, color = 'gray', linestyle='dashed',linewidth = 2, marke
plt.show()
```



Vertical bar charts

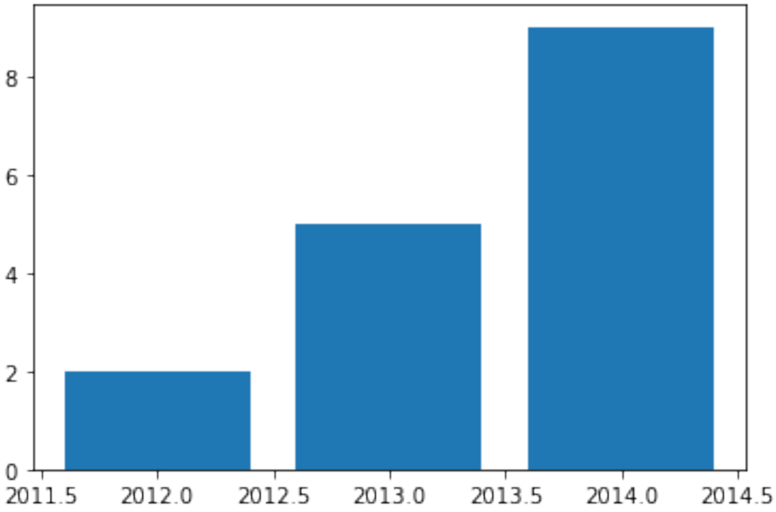
Create a vertical bar chart for the values by years.

Its ok to have years split i.e, (2012.0, 2012.5...)

In [23]:

```
# Using
years = np.arange(2012, 2015)
values = [2, 5, 9]
# Create a vertical bar chart values by years.

# Code goes below:
plt.bar(years,values)
plt.show()
```



Create a multiserise bar chart with:

- category1_values: Color = Blue, No edgecolor, width = 0.4, align = center and a label of y1
- category2_values: Color = Orange, No edgecolor, width = 0.4, align = center and a label of y2

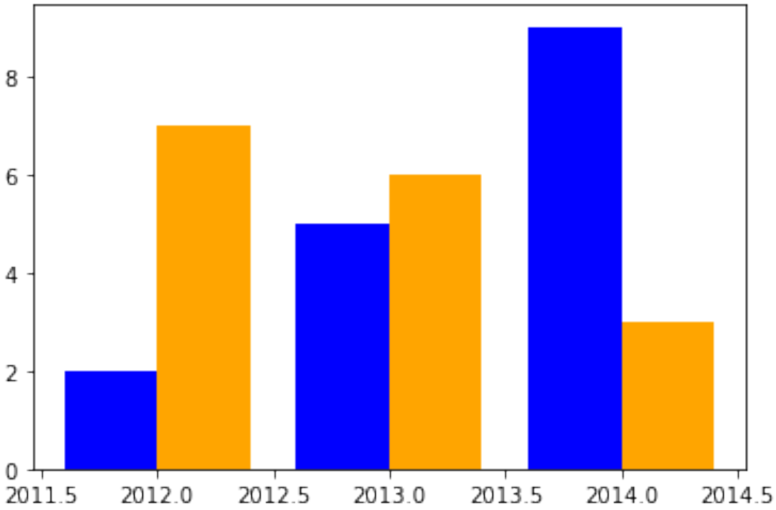
In [24]:

```
# Using:
years = np.arange(2012, 2015)
category1_values = [2, 5, 9]
category2_values = [7, 6, 3]

# Create a multiserise bar chart with:
#   category1_values: Color = Blue, No edgecolor, width = 0.4, align =
#   category2_values: Color = Orange, No edgecolor, width = 0.4, align

# Code goes below:

plt.bar(years-0.2, category1_values, color = 'Blue', edgecolor= 'none'
plt.bar(years+0.2, category2_values, color = 'orange', edgecolor= 'nor
plt.show()
```



Horizontal bar charts

Create a horizontal bar chart for the given values:

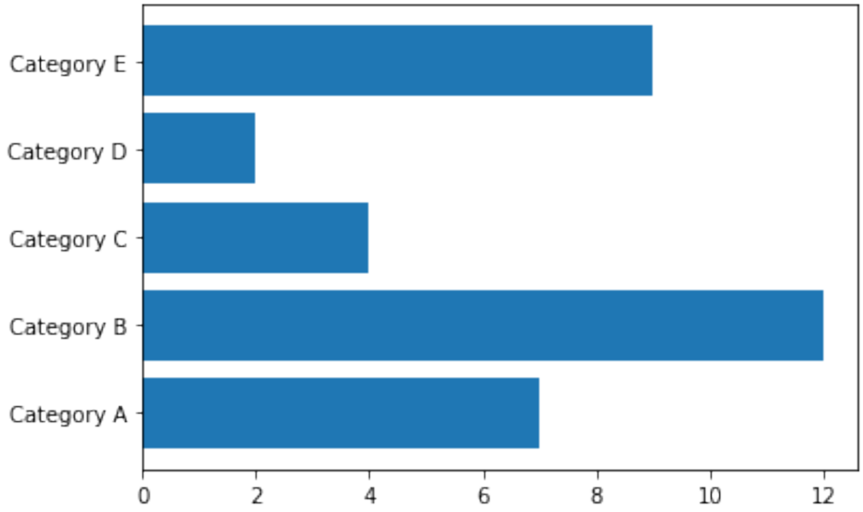
- categories = ['A', 'B', 'C', 'D', 'E']
- values = [7, 12, 4, 2, 9]
- Lable each as "Category and letter... i.e. Category A, Category B ...

In [25]:

```
# Using
categories = ['A', 'B', 'C', 'D', 'E']
values = [7, 12, 4, 2, 9]

# Create a horizontal bar chart for the values
# Lable each as "Category and letter... i.e. Category A, Category B ..

# Code goes below:
plt.barh(categories, values)
plt.yticks(['A', 'B', 'C', 'D', 'E'],
           ['Category A', 'Category B', 'Category C', 'Category D', '
plt.show()
```



Create a horizontal bar chart for the given values with:

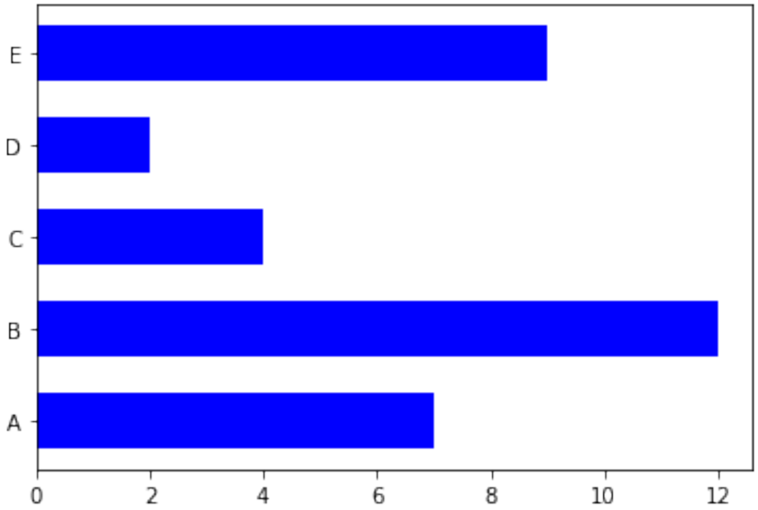
- Color = Blue, no edgecolor, height of 0.6, center aligned

In [26]:

```
# Using:
categories = ['A', 'B', 'C', 'D', 'E']
values = [7, 12, 4, 2, 9]

# Create a horizontal bar chart with
# Color = Blue, no edgecolor, height of 0.6, center aligned

# Code goes below:
plt.barh(categories, values, color= 'Blue', edgecolor= 'none', height
plt.show()
```



Pie charts

Create a pie chart

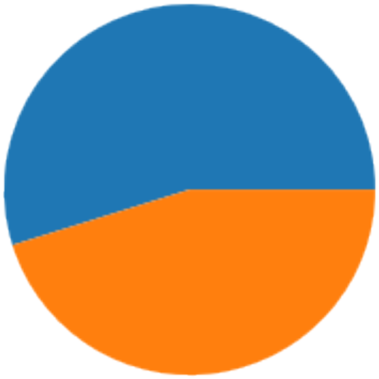
- For the given counts = [17, 14]

In [27]:

```
# Using
counts = [17, 14]

# Create a pie chart

# Code goes below:
plt.pie(counts)
plt.show()
```



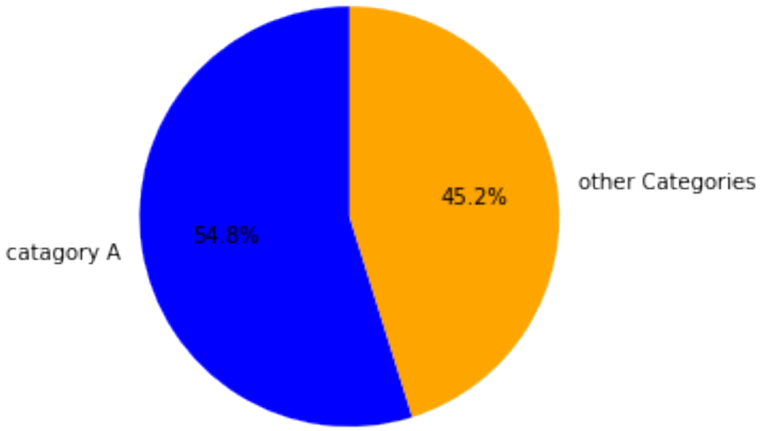
Create a pie chart for counts = [17, 14] using:

- Colors of blue and orange
- Labels of 'Category A' and 'Other categories'
- Set the angle at 90
- Set the percentage display format of 1.1

In [28]:

```
# Using
counts = [17, 14]
# Create a pie chart for counts = [17, 14] using:
#  Colors of blue and orange
#  Labels of 'Category A' and 'Other categories'
#  Set the angle at 90
#  Set the percentage display format of 1.1

# Code goes below:
labels = ['catagory A', 'other Categories']
colors = ['blue', 'orange']
plt.pie(counts,labels=labels, colors=colors, startangle=90 ,autopct='%
plt.axis('equal')
plt.show()
```



Scatter plots

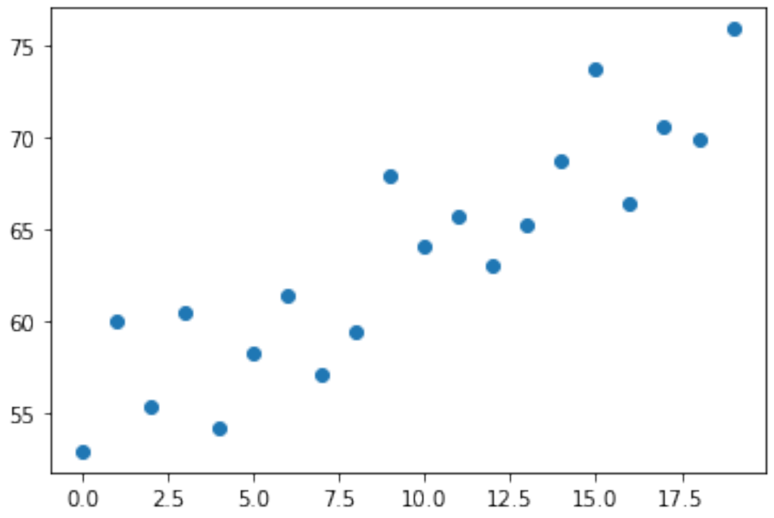
Create a scatter plot for the given data

In [29]:

```
# Using
x = range(20)
y = np.arange(50, 70) + (np.random.random(20) * 10.)

# Create a scatter plot for the given data

#Code goes below:
plt.scatter(x,y)
plt.show()
```



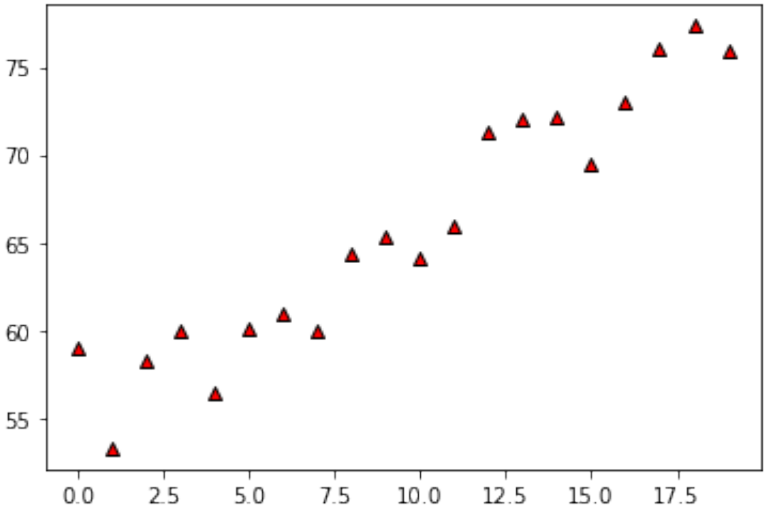
Create a scatter plot with the given data with:

- Color of red
- Marker
- `c` : Set the color of the markers.
- `s` : Set the size of the markers.
- `marker` : Set the marker style, e.g., circles, triangles, or squares.
- `edgecolor` : Set the color of the lines on the edges of the markers.

In [30]:

```
x = np.arange(20)
y = np.arange(50, 70) + (np.random.random(20) * 10.)

plt.scatter(x,y, color = 'red', marker= '^', edgecolor = 'black')
plt.show()
```



END OF EXERCISE