

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

## 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...)

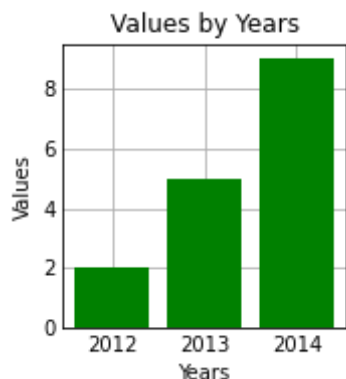
```
In [12]: # Using
years = np.arange(2012, 2015)
values = [2, 5, 9]
# Create a vertical bar chart values by years.

# Code goes below:
```

```
[2012 2013 2014]
```

```
In [22]: plt.title("Values by Years")
plt.xlabel('Years')
plt.ylabel('Values')
plt.bar(years, values, color = "green")
```

Out[22]: <BarContainer object of 3 artists>



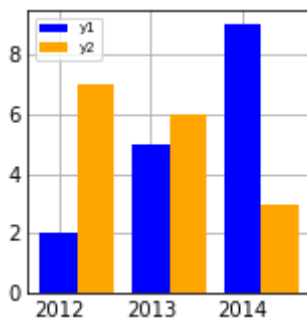
## Create a multiseriess bar chart with:

- 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 [60]: years = np.arange(2012, 2015)
category1_values = [2, 5, 9]
category2_values = [7, 6, 3]
# spacing bars out
xposCat1 = np.arange(len(years))
xposCat2 = xposCat1 + 0.4
```

```
In [76]: plt.bar(xposCat1, category1_values,color = 'blue', edgecolor='none', width
plt.bar(xposCat2, category2_values,color = 'orange', edgecolor='none', width
plt.legend(loc = "upper left",fontsize = 'x-small',)
# replace x indexes by the given values
plt.xticks(ticks = xposCat1,labels = years)
```

```
Out[76]: ([<matplotlib.axis.XTick at 0x12d7d6a10>,
<matplotlib.axis.XTick at 0x12d7d69e0>,
<matplotlib.axis.XTick at 0x12d7d68c0>],
[Text(0, 0, '2012'), Text(1, 0, '2013'), Text(2, 0, '2014')])
```

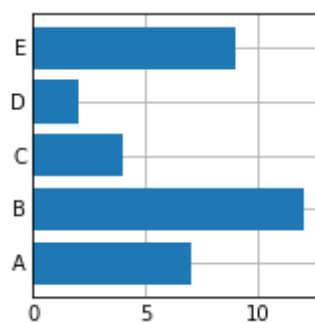


### Create a horizontal bar chart for the given values:

- categories = ['A', 'B', 'C', 'D', 'E']
- values = [7, 12, 4, 2, 9]
- Label each as the letter from the categories parameter... i.e. A, B, C ...

```
In [79]: categories = ['A', 'B', 'C', 'D', 'E']
values = [7, 12, 4, 2, 9]
y_pos = np.arange(len(categories))
```

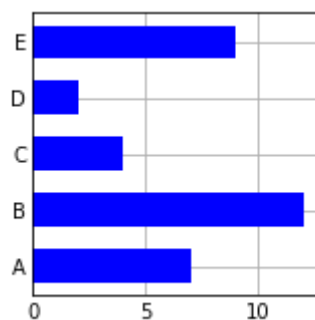
```
In [88]: plt.barh(y_pos,values)
plt.yticks(y_pos,categories)
plt.show()
```



## Create a horizontal bar chart with these values :

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

```
In [91]: plt.barh(y_pos,values, color ="blue",edgecolor='none',height = 0.6,align = 'center')
plt.yticks(y_pos,categories)
plt.show()
```

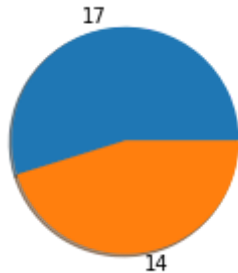


## Create a pie chart

- For the given counts = [17, 14]

```
In [101]: counts = [17, 14]
lab = ["17", "14"]
plt.pie(counts, shadow = True, labels= lab)
```

```
Out[101]: ([<matplotlib.patches.Wedge at 0x12ddfcfd0>,
<matplotlib.patches.Wedge at 0x12ddfe560>],
[Text(-0.1665706012300369, 1.0873151497178102, '17'),
Text(0.16657060123003722, -1.0873151497178102, '14')])
```

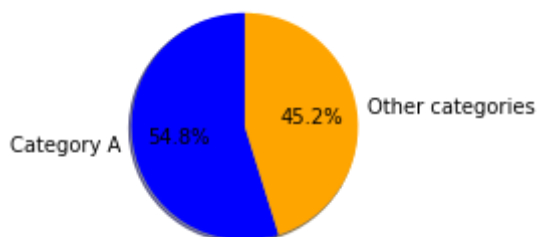


### Create a pie chart for counts = [17, 14] using:

- 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

```
In [109]: counts = [17, 14]
colors = ['blue', 'orange']
lab = ["Category A", "Other categories"]
plt.pie(counts, shadow = True, labels= lab, colors = colors, startangle = 90,
```

```
Out[109]: ([<matplotlib.patches.Wedge at 0x12e02cbb0>,
<matplotlib.patches.Wedge at 0x12e02d540>],
[Text(-1.0873151497178102, -0.1665706012300368, 'Category A'),
Text(1.0873151497178102, 0.16657060123003717, 'Other categories')],
[Text(-0.5930809907551692, -0.09085669158002008, '54.8%'),
Text(0.5930809907551692, 0.09085669158002026, '45.2%')])
```

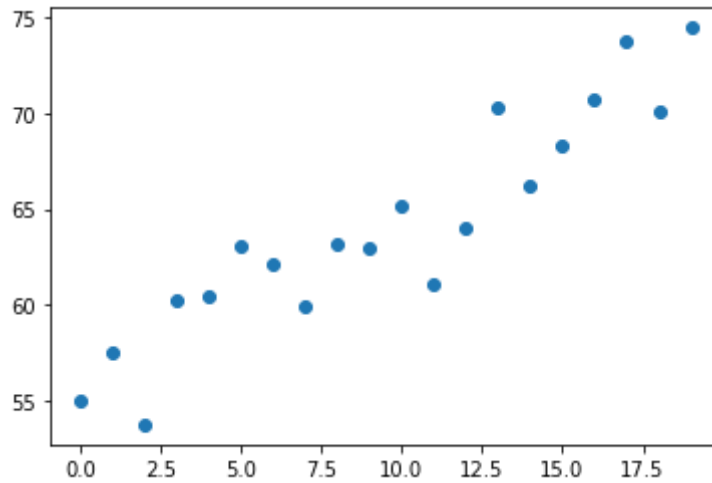


### Create a scatter plot for the given data

```
In [18]: x = range(20)
y = np.arange(50, 70) + (np.random.random(20) * 10. )
x
```

```
Out[18]: range(0, 20)
```

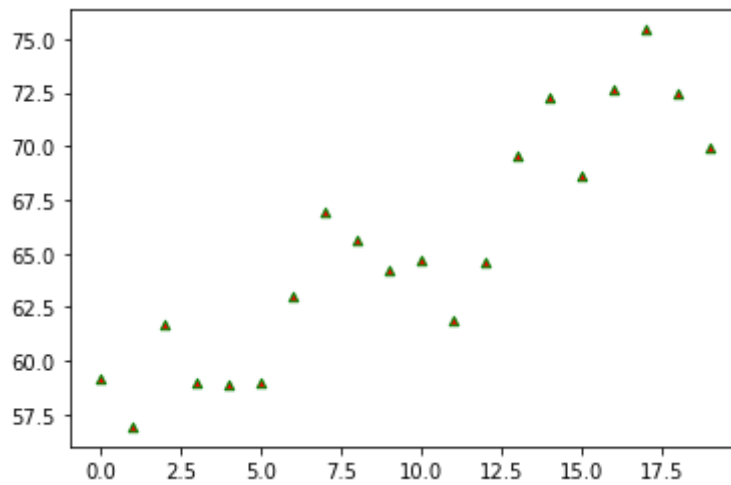
```
In [16]: plt.scatter(x,y)
plt.show()
```



## 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 [24]: x = np.arange(20)
y = np.arange(50, 70) + (np.random.random(20) * 10.)
plt.scatter(x,y,marker = '^', c='red', edgecolor = 'green',s = 20, )
plt.show()
```



**END**

In [ ]: