Data Analytics and Machine Learning using Python

DataRitz Technologies
Enhancing Technology Experience

Session : 09 June 2020

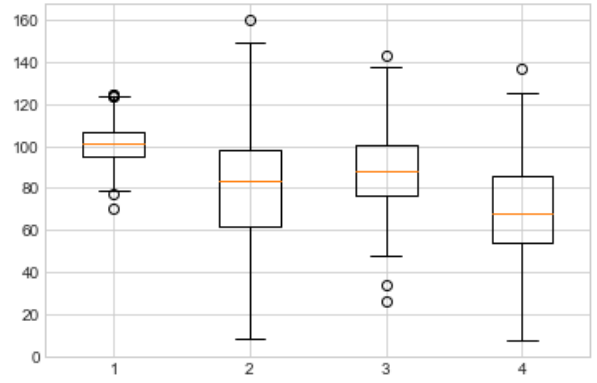
In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Box Plot or Whisker Plot

- A box plot which is also known as a whisker plot displays a summary of a set of data containing the minimum, first quartile, median, third quartile, and maximum. In a box plot, we draw a box from the first quartile to the third quartile.
- A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.
- Box Plot is the visual representation of the depicting groups of numerical data through their quartiles
- A box plot consist of 5 things.
- Minimum
- First Quartile or 25%
- Median (Second Quartile) or 50%
- Third Quartile or 75%
- Maximum

In [145]:
```
# Example 1
np.random.seed(10)
collectn_1 = np.random.normal(100, 10, 200)
collectn_2 = np.random.normal(80, 30, 200)
collectn_3 = np.random.normal(90, 20, 200)
collectn_4 = np.random.normal(70, 25, 200)
data=[collectn_1,collectn_2,collectn_3,collectn_4]
bp=plt.boxplot(data)
plt.show()

# Here we have used np.random.normal to create the fake data where the 1st argument is mean, 2nd is
# values produced
```

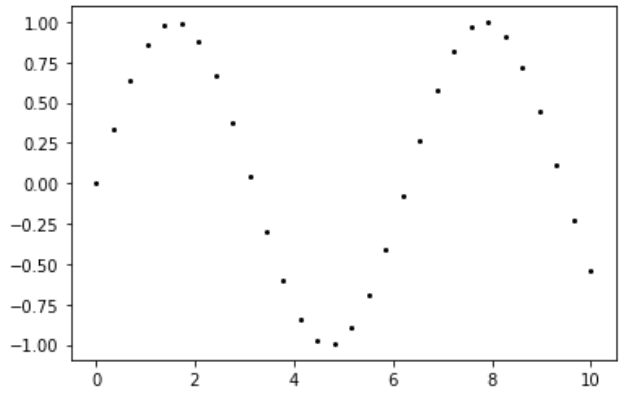


Scatter Plot

- Another commonly used plot type is the simple scatter plot, a close cousin of the line plot. Instead of points being joined by line segments, here the points are represented individually with a dot, circle, or other shape

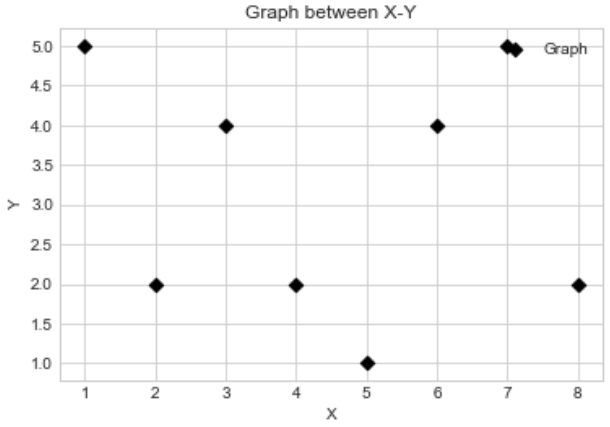
In [13]:
```
# Exampel 1
x = np.linspace(0, 10, 30)
y = np.sin(x)

plt.plot(x, y, 'o', color='black', markersize=2);
```



In [154]:
```
# Example 2
x = [1,2,3,4,5,6,7,8]
y = [5,2,4,2,1,4,5,2]
plt.scatter(x,y,color='black',marker='D',label='Graph')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Graph between X-Y')
plt.legend(loc='upper right')
```

Out[154]: <matplotlib.legend.Legend at 0x1cd1f0d5308>



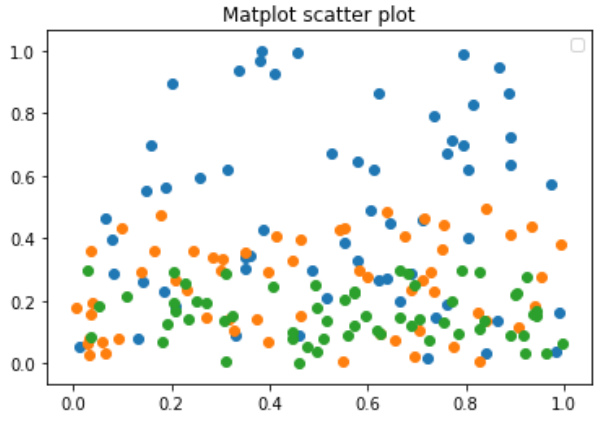
In [6]:
```
# Example 3
N = 60
g1 = (np.random.rand(N), np.random.rand(N))
g2 = (np.random.rand(N), 0.5*np.random.rand(N))
g3 = (np.random.rand(N),0.3*np.random.rand(N))

data = (g1, g2, g3)
colors = ("red", "green", "blue")
groups = ("coffee", "tea", "water")

# Create plot
#fig = plt.figure()
#ax = fig.add_subplot()

for data, color, group in zip(data, colors, groups):
    x, y = data
    plt.scatter(x, y, alpha=1)
plt.title('Matplot scatter plot')
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



Pie Chart

- Pie charts show the size of items (called wedge) in one data series, proportional to the sum of the items.
- The data points in a pie chart are shown as a percentage of the whole pie.
- Matplotlib API has a pie() function that generates a pie diagram representing data in an array.

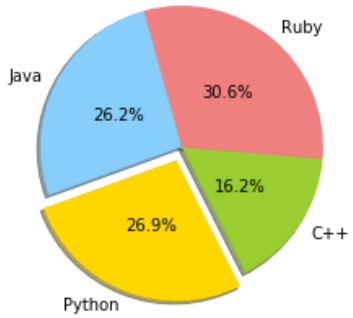
In [7]:
```
plt.pie?
```

In [24]:
```
# Example 1
labels = 'Python', 'C++', 'Ruby', 'Java'
sizes = [215, 130, 245, 210]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
explode = (0.1, 0, 0, 0)  # explode 1st slice

# Plot
plt.pie(sizes, explode=explode, labels=labels, colors=colors, shadow=True, startangle=200,autopct='%

plt.show()

#autopct enables you to show the data in percentage
#startangle - default with x-axis and rotates counter clockwise by that angle
```



Example 2

slices = [7,2,2,13] activities = ['sleeping','eating','working','playing'] cols = ['c','m','r','b']

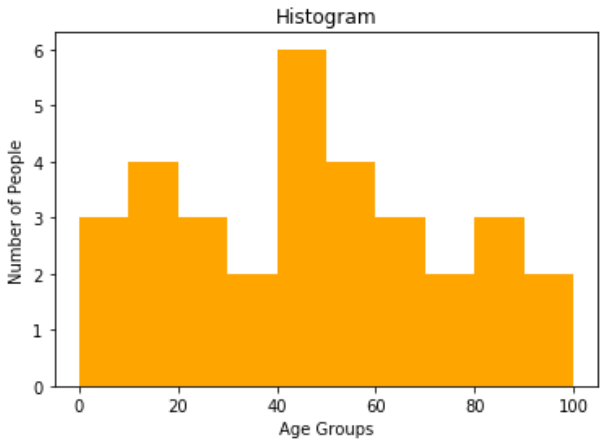
plt.pie(slices,labels=activities,colors=cols,startangle=90,shadow= True,explode=(0,0.1,0,0),autopct='%1.1f%%')

plt.title('Interesting Graph\nCheck it out') plt.show()

Histogram

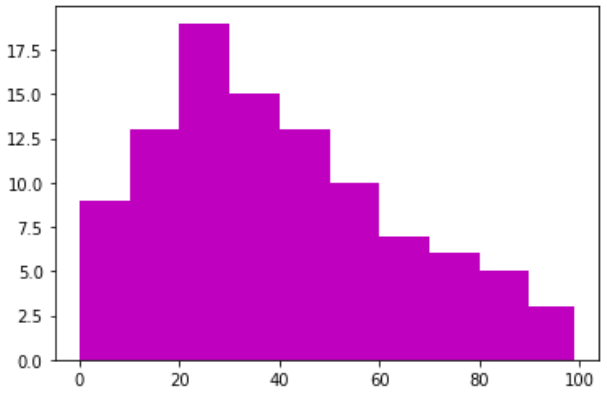
- A histogram shows the frequency on the vertical axis and the horizontal axis is another dimension
- Usually it has bins, where every bin has a minimum and maximum value. Each bin also has a frequency between x and infinite
- Bin refers to the range of values that are divided into series of intervals

In [25]:
```
# Example 1
age = [12,22,55,62,45,21,22,34,42,37,4,2,102,95,85,55,88,70,95,65,55,7,80,75,65,54,44,43,42,48,11,18
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist(age, bins, color='orange')
plt.xlabel('Age Groups')
plt.ylabel('Number of People')
plt.title('Histogram')
plt.show()
```



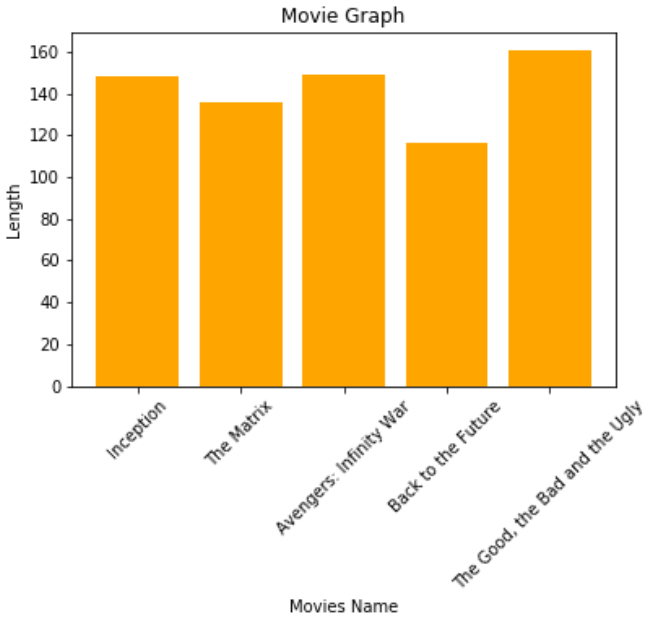
Intervals (bins)	Frequency
0-9	9
10-19	13
20-29	19
30-39	15
40-49	13
50-59	10
60-69	7
70-79	6
80-89	5
90–99	3

In [26]:
```
# Example 2
freq=[1,1,2,3,3,5,7,8,9,10,10,11,11,13,13,15,16,17,18,18,18,19,20,21,21,23,24,24,25,25,25,25,26,26,2
     29,30,30,31,33,34,34,34,35,36,36,37,37,38,38,39,40,41,41,42,43,44,45,45,46,47,48,48,49,50,51,52
     61,63,64,65,66,68,70,71,72,74,75,77,81,83,84,87,89,90,90,91]
bins=[0,10,20,30,40,50,60,70,80,90,99]
plt.hist(freq,bins,color='m')
plt.show()
```



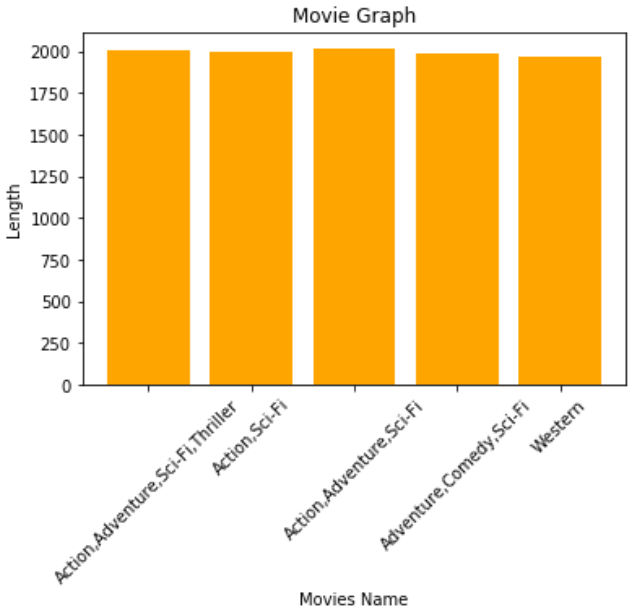
In [111]:
```
plt.bar(top['Title'],top['Runtime'],color='orange')
plt.xlabel('Movies Name')
plt.ylabel('Length')
plt.title('Movie Graph')
plt.xticks(rotation='45')
```

Out[111]: ([0, 1, 2, 3, 4], <a list of 5 Text xticklabel objects>)



In [112]:
```
plt.bar(top['Genres'],top['Year'],color='orange')
plt.xlabel('Movies Name')
plt.ylabel('Length')
plt.title('Movie Graph')
plt.xticks(rotation='45')
#plt.ylim((1955,2020))
```

Out[112]: ([0, 1, 2, 3, 4], <a list of 5 Text xticklabel objects>)



In [4]:
```
df_cars=pd.read_csv('Cars.csv')
bot=df_cars.tail(100)
bot
```

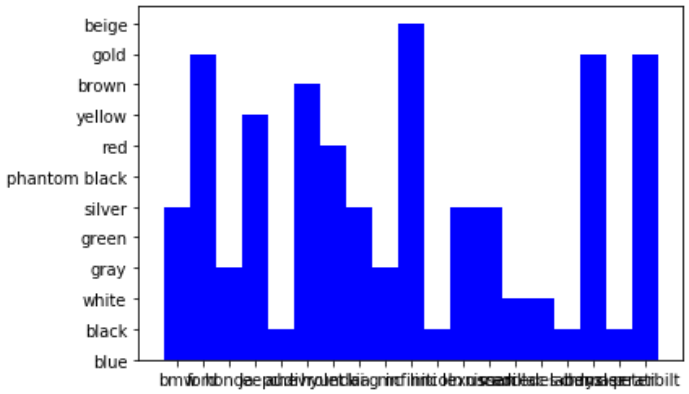
Out[4]:

	Unnamed: 0	price	brand	model	year	title_status	mileage	color	vin	lot	state	cou
401	401	13500	bmw	series	2014	clean vehicle	33542	blue	wba3d5c51ekx97650	167782815	new jersey	
402	402	6490	ford	door	2015	clean vehicle	74754	black	1fm5k7d88fga24308	167660611	georgia	
403	403	51502	ford	door	2017	clean vehicle	27661	black	1ft8w3dt8hec12120	167765457	michigan	
404	404	250	ford	door	2010	salvage insurance	115241	white	3fahp0ha6ar187697	167362678	michigan	
405	405	25	ford	door	1984	salvage insurance	41577	white	2ftcf15y9eca14589	167611661	arkansas	
...	...	...	...	...	...	...	...	...	...	...	...	
496	496	0	ford	pickup	1996	salvage insurance	252588	red	1ftef15n0tlc14455	167357804	oklahoma	
497	497	26000	gmc	limited	2019	clean vehicle	16669	black	2gtv2mecxk1173011	167603430	pennsylvania	
498	498	3900	honda	doors	2017	clean vehicle	47970	blue	2hgfc2f52hh522914	167702907	virginia	
499	499	11200	jeep	mpv	2019	clean vehicle	12905	gray	1c4hjxeg7kw590115	167256952	michigan	
500	500	8610	jeep	mpv	2019	clean vehicle	22123	silver	1c4rjebg1kc776053	167697952	ohio	

100 rows × 13 columns

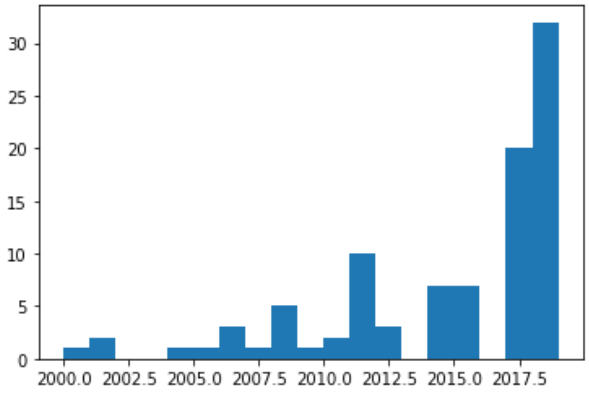
In [5]:
```
plt.bar(bot['brand'],bot['color'],color='blue',width=1)
```

Out[5]:
```
<BarContainer object of 100 artists>
```



In [7]:
```
year=bot['year']
bins=np.arange(2000,2020)
plt.hist(year,bins,rwidth=1)
```

Out[7]:
```
(array([ 1.,  2.,  0.,  0.,  1.,  1.,  3.,  1.,  5.,  1.,  2., 10.,  3.,
         0.,  7.,  7.,  0., 20., 32.]),
 array([2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010,
        2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]),
 <a list of 19 Patch objects>)
```



In [96]:
```
#either benign or malignant
cancer_df=pd.read_csv('data.csv')
```

In [109]:
```
cancer_df.head()
```

Out[109]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280

5 rows × 32 columns

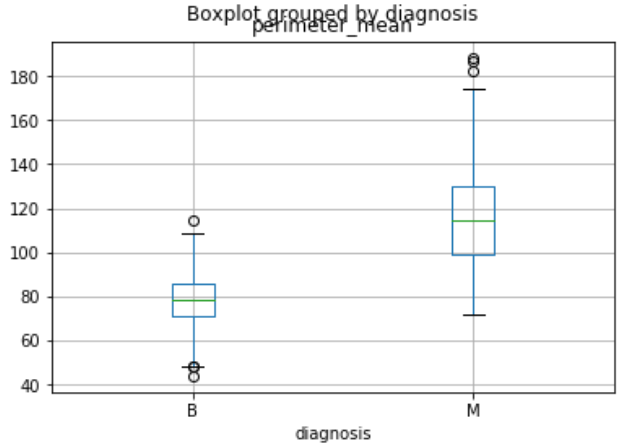
In [110]:
```
# Looking at the distribution of dataset in terms of diagnosis
cancer_df['diagnosis'].value_counts(dropna=False)
```

Out[110]:
```
B    357
M    212
Name: diagnosis, dtype: int64
```

Plotting using Matplotlib

In [108]:
```
cancer_df.boxplot(column='perimeter_mean',by='diagnosis')
```

Out[108]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1993767ce88>
```



In [ ]: