In [2]:
```
import pandas as pd
import datetime as dt
from pandas_datareader import data
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline

#import all modules
```

In [3]:
```
stocks = data.DataReader(name = ["IBM", "ORCL", "MSFT"], data_source = "yahoo"
, start = "2020-10-01", end = "2020-10-20")
dates = pd.date_range(start = "2020-10-01", end = "2020-10-20", freq = "B")
#get the stocks info from the yahoo website, and store them as a dataframe
```

In [3]:
```
stocks.head()  #show the top 5 lines of the stocks dataframe
```

Out[3]:

Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	OR
Date								
2020-10-01	121.089996	59.439316	212.460007	121.089996	59.680000	212.460007	123.300003	60
2020-10-02	120.570000	58.592747	206.190002	120.570000	58.830002	206.190002	121.750000	59
2020-10-05	122.010002	59.319801	210.380005	122.010002	59.560001	210.380005	122.750000	59
2020-10-06	121.970001	59.270000	205.910004	121.970001	59.509998	205.910004	124.830002	60
2020-10-07	124.070000	60.590000	209.830002	124.070000	60.590000	209.830002	124.389999	60

In [4]:
```
stocks.to_csv("stocks.csv")
```

In [112]:

```
stocks.describe()   #show the numeric details of the stocks dataframe
```

Out[112]:

Attributes	Adj Close			Close			High	
Symbols	**IBM**	**ORCL**	**MSFT**	**IBM**	**ORCL**	**MSFT**	**IBM**	
count	14.000000	14.000000	14.000000	14.000000	14.000000	14.000000	14.000000	14
mean	124.353572	60.201562	214.605001	124.353572	60.270000	214.605001	126.143572	60
std	3.542000	0.865569	5.643365	3.542000	0.779674	5.643365	4.018652	0
min	117.370003	58.592747	205.910004	117.370003	58.830002	205.910004	120.150002	59
25%	121.980001	59.484487	210.430004	121.980001	59.634999	210.430004	123.572502	60
50%	124.994999	60.405001	214.434998	124.994999	60.405001	214.434998	125.825001	60
75%	125.937502	60.942499	219.660004	125.937502	60.942499	219.660004	127.299999	61
max	131.490005	61.459999	222.860001	131.490005	61.459999	222.860001	135.500000	61

In [ ]:

```
stocks.describe().to_csv("stocks description.csv") #output the above description info to a csv file.
```

In [9]:

```
stocks.info()  #present the info of each column in the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14 entries, 2020-10-01 to 2020-10-20
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   (Adj Close, IBM)   14 non-null     float64
 1   (Adj Close, ORCL)  14 non-null     float64
 2   (Adj Close, MSFT)  14 non-null     float64
 3   (Close, IBM)       14 non-null     float64
 4   (Close, ORCL)      14 non-null     float64
 5   (Close, MSFT)      14 non-null     float64
 6   (High, IBM)        14 non-null     float64
 7   (High, ORCL)       14 non-null     float64
 8   (High, MSFT)       14 non-null     float64
 9   (Low, IBM)         14 non-null     float64
 10  (Low, ORCL)        14 non-null     float64
 11  (Low, MSFT)        14 non-null     float64
 12  (Open, IBM)        14 non-null     float64
 13  (Open, ORCL)       14 non-null     float64
 14  (Open, MSFT)       14 non-null     float64
 15  (Volume, IBM)      14 non-null     float64
 16  (Volume, ORCL)     14 non-null     float64
 17  (Volume, MSFT)     14 non-null     float64
dtypes: float64(18)
memory usage: 2.7 KB
```

In [8]:

```
stocks.loc["2020-10-05": "2020-10-10", "High"]  #show the high column value of
rows from 2020-10-05 to 2020-10-10
```

Out[8]:

Symbols	IBM	ORCL	MSFT
Date			
2020-10-05	122.750000	59.669998	210.410004
2020-10-06	124.830002	60.529999	210.179993
2020-10-07	124.389999	60.900002	210.110001
2020-10-08	135.500000	61.299999	211.190002
2020-10-09	132.000000	61.380001	215.860001

In [18]:

```
stocks.iloc[2:7, 6:9] #the same output as above, but using index locations ins
tead of index names
```

Out[18]:

Attributes	High		
Symbols	**IBM**	**ORCL**	**MSFT**
Date			
2020-10-05	122.750000	59.669998	210.410004
2020-10-06	124.830002	60.529999	210.179993
2020-10-07	124.389999	60.900002	210.110001
2020-10-08	135.500000	61.299999	211.190002
2020-10-09	132.000000	61.380001	215.860001

In [20]:

```
np.mean(stocks["High"]["IBM"]) #get the mean of all IBM high
```

Out[20]:

```
126.14357212611607
```

In [28]:

```
np.max(stocks["High"]["IBM"]) #get the max of all IBM high
```

Out[28]:

```
135.5
```

In [32]:

```
np.min(stocks["High"]["IBM"]) #get the min of all IBM high
```

Out[32]:

```
120.1500015258789
```

In [35]:

```
np.std(stocks["High"]["IBM"]) #get the std of all IBM high
```

Out[35]:

```
3.8724695360832353
```

In [37]:

```
np.sum(stocks["High"]["IBM"]) #get the sum of all IBM high
```

Out[37]:

```
1766.010009765625
```

In [34]:

```
np.percentile(stocks["High"]["IBM"], 25)
```

Out[34]:

```
123.57250213623047
```

In [30]:

```
np.percentile(stocks["High"]["IBM"], 75)
```

Out[30]:

```
127.29999923706055
```

In [123]:

```
stocks[(stocks["High"]["IBM"] > 126) & (stocks["High"]["ORCL"] > 61.5)]
```

Out[123]:

Attributes	**Adj Close**			**Close**			**High**	
Symbols	**IBM**	**ORCL**	**MSFT**	**IBM**	**ORCL**	**MSFT**	**IBM**	**OI**
Date								
2020-10-12	127.209999	61.459999	221.399994	127.209999	61.459999	221.399994	128.250000	61
2020-10-13	125.099998	60.970001	222.860001	125.099998	60.970001	222.860001	127.150002	61
2020-10-14	125.940002	60.959999	220.860001	125.940002	60.959999	220.860001	126.940002	61

In [5]:

```
stocks[(stocks["High"]["IBM"] > 126) | (stocks["High"]["ORCL"] > 61.5)]
```

Out[5]:

Attributes	**Adj Close**			**Close**			**High**	
Symbols	**IBM**	**ORCL**	**MSFT**	**IBM**	**ORCL**	**MSFT**	**IBM**	**OI**
Date								
2020-10-08	131.490005	60.889999	210.580002	131.490005	60.889999	210.580002	135.500000	61
2020-10-09	127.790001	61.150002	215.809998	127.790001	61.150002	215.809998	132.000000	61
2020-10-12	127.209999	61.459999	221.399994	127.209999	61.459999	221.399994	128.250000	61
2020-10-13	125.099998	60.970001	222.860001	125.099998	60.970001	222.860001	127.150002	61
2020-10-14	125.940002	60.959999	220.860001	125.940002	60.959999	220.860001	126.940002	61
2020-10-16	125.930000	60.290001	219.660004	125.930000	60.290001	219.660004	126.430000	61
2020-10-19	125.519997	59.619999	214.220001	125.519997	59.619999	214.220001	127.349998	6C

In [6]:

```
stocks[(stocks["High"]["IBM"] > np.mean(stocks["High"]["IBM"])) & (stocks["High"]["ORCL"] > np.mean(stocks["High"]["ORCL"]))]
```

Out[6]:

Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	OI
Date								
2020-10-08	131.490005	60.889999	210.580002	131.490005	60.889999	210.580002	135.500000	61
2020-10-09	127.790001	61.150002	215.809998	127.790001	61.150002	215.809998	132.000000	61
2020-10-12	127.209999	61.459999	221.399994	127.209999	61.459999	221.399994	128.250000	61
2020-10-13	125.099998	60.970001	222.860001	125.099998	60.970001	222.860001	127.150002	61
2020-10-14	125.940002	60.959999	220.860001	125.940002	60.959999	220.860001	126.940002	61
2020-10-16	125.930000	60.290001	219.660004	125.930000	60.290001	219.660004	126.430000	61

In [7]:

```
stocks[(stocks["High"]["IBM"] > np.mean(stocks["High"]["IBM"])) | (stocks["High"]["ORCL"] > np.mean(stocks["High"]["ORCL"]))]
```

Out[7]:

Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	OI
Date								
2020-10-07	124.070000	60.590000	209.830002	124.070000	60.590000	209.830002	124.389999	60
2020-10-08	131.490005	60.889999	210.580002	131.490005	60.889999	210.580002	135.500000	61
2020-10-09	127.790001	61.150002	215.809998	127.790001	61.150002	215.809998	132.000000	61
2020-10-12	127.209999	61.459999	221.399994	127.209999	61.459999	221.399994	128.250000	61
2020-10-13	125.099998	60.970001	222.860001	125.099998	60.970001	222.860001	127.150002	61
2020-10-14	125.940002	60.959999	220.860001	125.940002	60.959999	220.860001	126.940002	61
2020-10-16	125.930000	60.290001	219.660004	125.930000	60.290001	219.660004	126.430000	61
2020-10-19	125.519997	59.619999	214.220001	125.519997	59.619999	214.220001	127.349998	60

In [8]:

```
stocks[stocks["High"]["IBM"] > stocks["High"]["MSFT"] * 0.6]
```

Out[8]:

Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	ORCL
Date								
2020-10-08	131.490005	60.889999	210.580002	131.490005	60.889999	210.580002	135.5	61.2999
2020-10-09	127.790001	61.150002	215.809998	127.790001	61.150002	215.809998	132.0	61.3800

In [17]:

```
stocks[stocks["High"]["IBM"] > stocks["Low"]["MSFT"] * 0.58]
```

Out[17]:

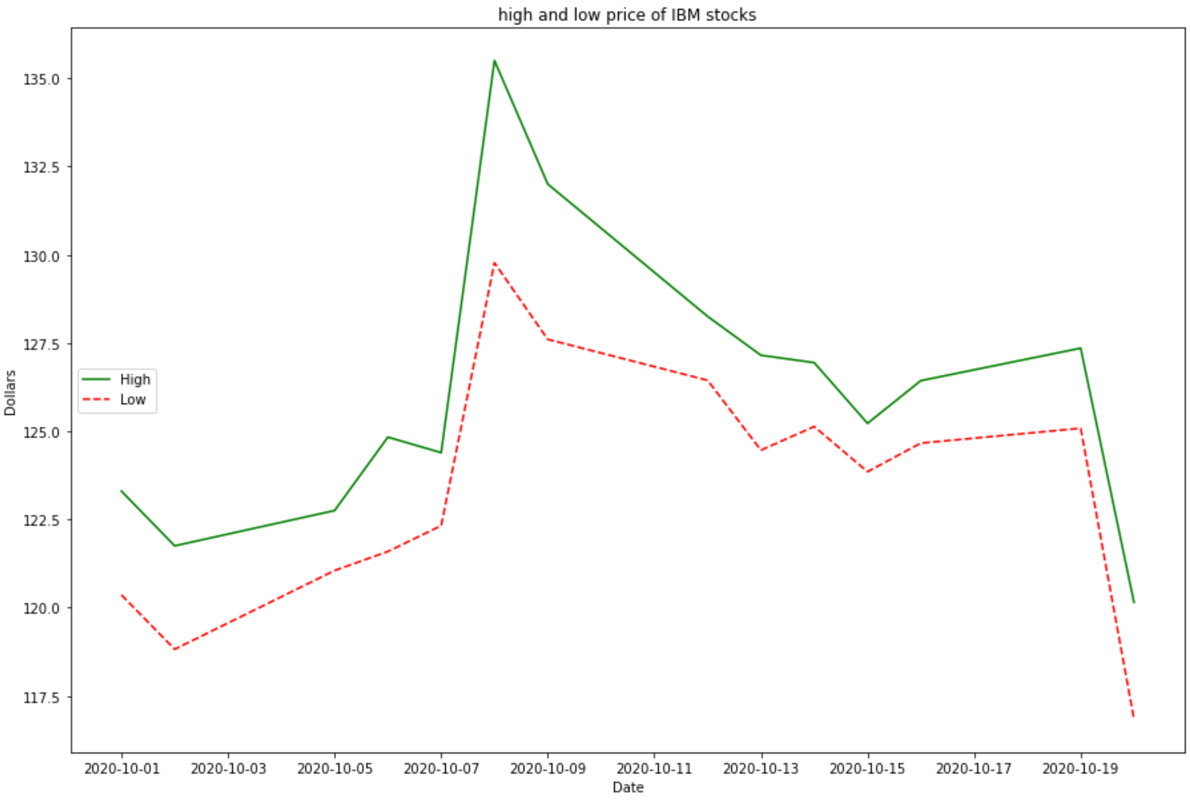
Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	O
Date								
2020-10-01	121.089996	59.439316	212.460007	121.089996	59.680000	212.460007	123.300003	60
2020-10-02	120.570000	58.592747	206.190002	120.570000	58.830002	206.190002	121.750000	59
2020-10-05	122.010002	59.319801	210.380005	122.010002	59.560001	210.380005	122.750000	59
2020-10-06	121.970001	59.270000	205.910004	121.970001	59.509998	205.910004	124.830002	60
2020-10-07	124.070000	60.590000	209.830002	124.070000	60.590000	209.830002	124.389999	60
2020-10-08	131.490005	60.889999	210.580002	131.490005	60.889999	210.580002	135.500000	61
2020-10-09	127.790001	61.150002	215.809998	127.790001	61.150002	215.809998	132.000000	61
2020-10-12	127.209999	61.459999	221.399994	127.209999	61.459999	221.399994	128.250000	61
2020-10-19	125.519997	59.619999	214.220001	125.519997	59.619999	214.220001	127.349998	60

In [30]:
```
stocks[stocks["High"]["ORCL"] < (stocks["Low"]["MSFT"] - stocks["Low"]["IBM"]) * 0.65]
```

Out[30]:

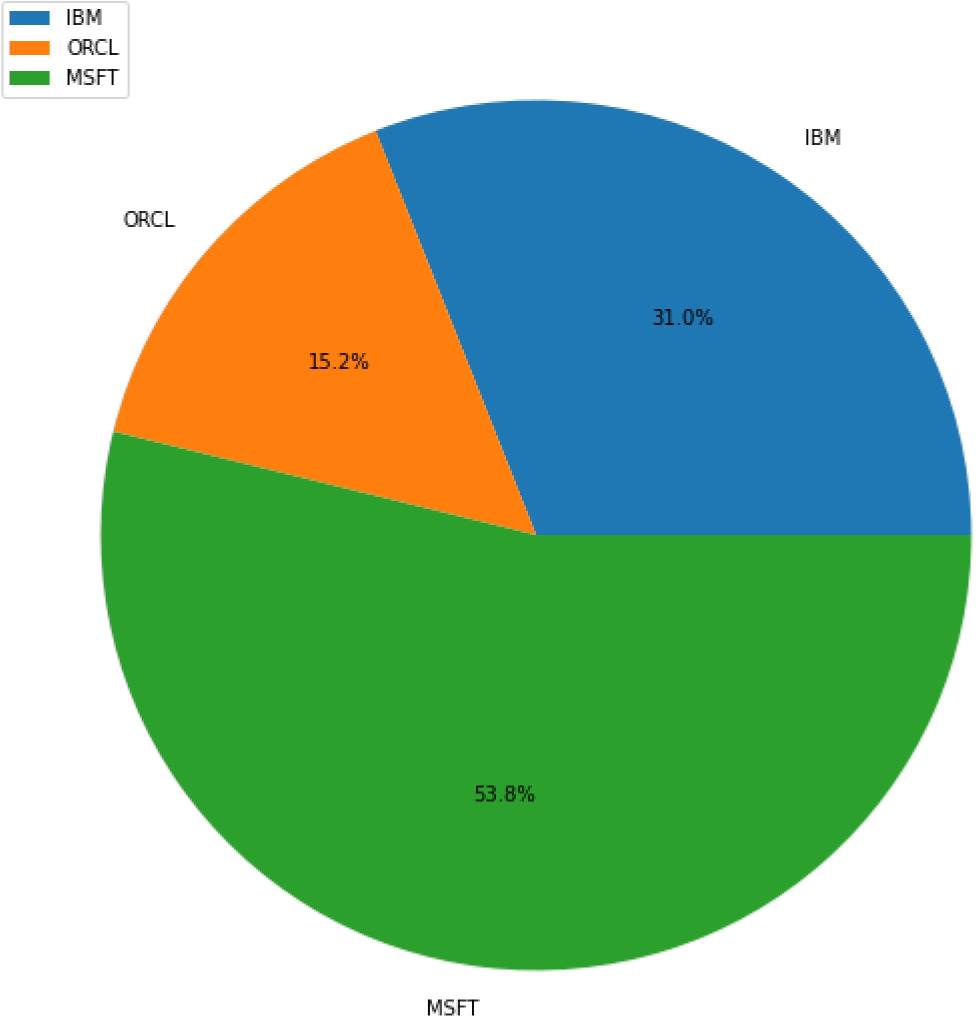
Attributes	Adj Close			Close			High	
Symbols	IBM	ORCL	MSFT	IBM	ORCL	MSFT	IBM	OI
Date								
2020-10-13	125.099998	60.970001	222.860001	125.099998	60.970001	222.860001	127.150002	61
2020-10-16	125.930000	60.290001	219.660004	125.930000	60.290001	219.660004	126.430000	61
2020-10-20	117.370003	59.750000	214.649994	117.370003	59.750000	214.649994	120.150002	60

In [105]:
```
plt.figure(figsize = (15,10))  #create a new figure
plt.plot(stocks["High"]["IBM"], color = "Green", label = "High")  # plot the line of IBM stock high price, x axis is date
plt.plot(stocks["Low"]["IBM"], color = "Red", label = "Low", linestyle = "--") # plot the line of IBM stock low price.
plt.xlabel("Date")
plt.ylabel("Dollars")
plt.title("high and low price of IBM stocks")
plt.legend(loc = 6) #location of the legend
plt.savefig("IBM stock high and low prices - October.jpg")
#use matplotlib to show a line chart to present the high and low price of the IBM stock
```



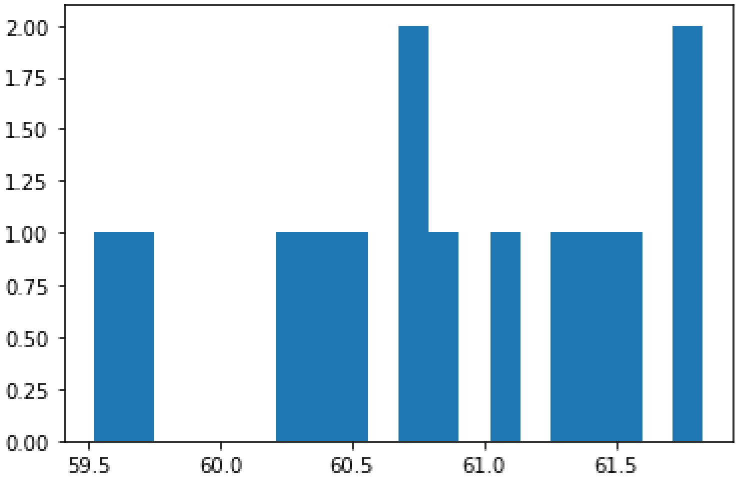
In [114]:

```
plt.figure(figsize = (15,10))  #create a new figure
plt.pie(stocks.loc["2020-10-01", "High"], labels = ["IBM", "ORCL", "MSFT"], autopct='%0.1f%%')
plt.legend()
plt.savefig("IBM stock high Pie chart.jpg")
#use a pie chart to compare the high of each stocks on 2020-10-01
```



```
In [85]: plt.hist(stocks["High"]["ORCL"], bins = 20)
```

```
Out[85]: (array([1., 1., 0., 0., 0., 0., 1., 1., 1., 0., 2., 1., 0., 1., 0., 1., 1.,
                 1., 0., 2.]),
          array([59.52000046, 59.63550053, 59.7510006 , 59.86650066, 59.98200073,
                 60.0975008 , 60.21300087, 60.32850094, 60.44400101, 60.55950108,
                 60.67500114, 60.79050121, 60.90600128, 61.02150135, 61.13700142,
                 61.25250149, 61.36800156, 61.48350163, 61.59900169, 61.71450176,
                 61.83000183]),
          <a list of 20 Patch objects>)
```


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
59.5
60.0
60.5
61.0
61.5