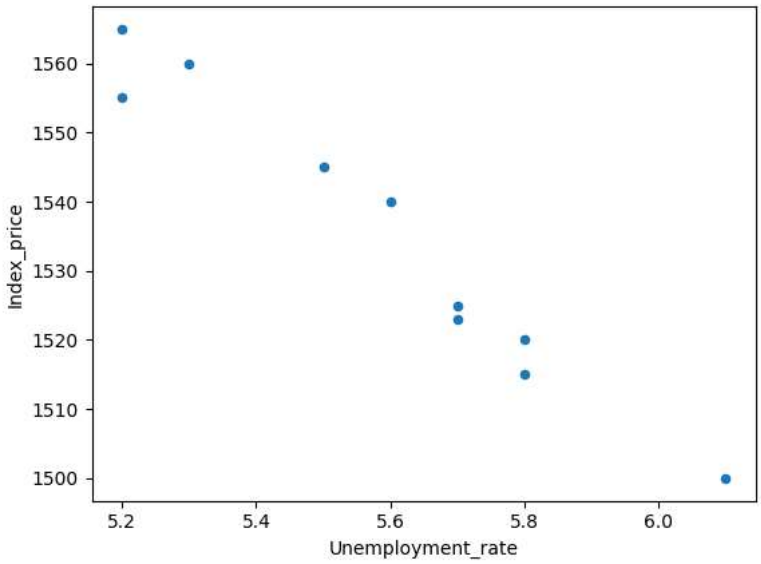
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd


'''data = {'unemployment_rate': [6.1, 5.8, 5.7, 5.7, 5.8, 5.6, 5.5, 5.3, 5.2, 5.2],
        'index_price': [1500, 1520, 1525, 1523, 1515, 1540, 1545, 1560, 1555, 1565]
       }
       '''

df=pd.read_csv('/content/plotdata.csv')
print(df)
```

```
   Year  Unemployment_rate  Index_price
0  1930                6.1         1500
1  1940                5.8         1520
2  1950                5.7         1525
3  1960                5.7         1523
4  1970                5.8         1515
5  1980                5.6         1540
6  1990                5.5         1545
7  2000                5.3         1560
8  2010                5.2         1555
9  2020                5.2         1565
```

```
df.plot(x='Unemployment_rate', y='Index_price', kind='scatter')
plt.show()
```

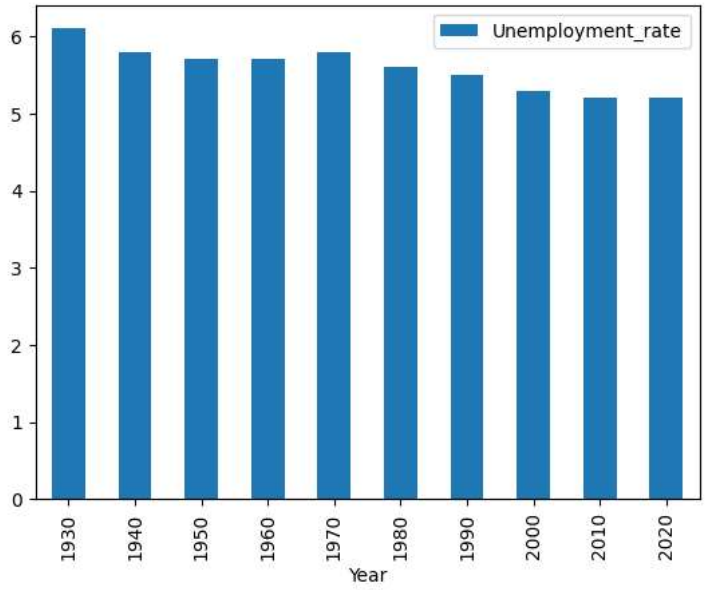


Line Chart

```
df.plot(x ='Year', y='Unemployment_rate', kind='line')
plt.show()
```

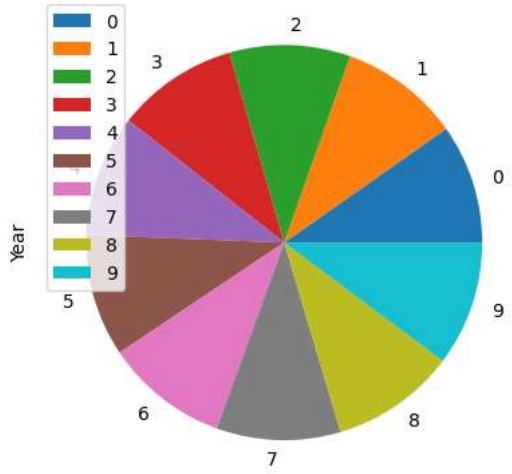
Bar Chart

```
df.plot(x='Year', y='Unemployment_rate', kind='bar')
plt.show()
```



Pie Chart

```
df.plot(y='Year', x='Unemployment_rate', kind='pie')
plt.show()
```



Common Plots using Matplotlib

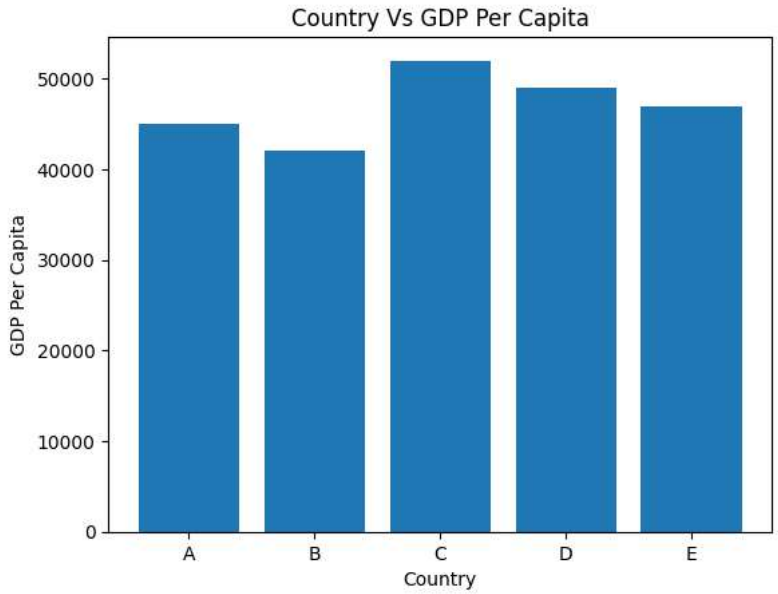
Template

import matplotlib.pyplot as plt

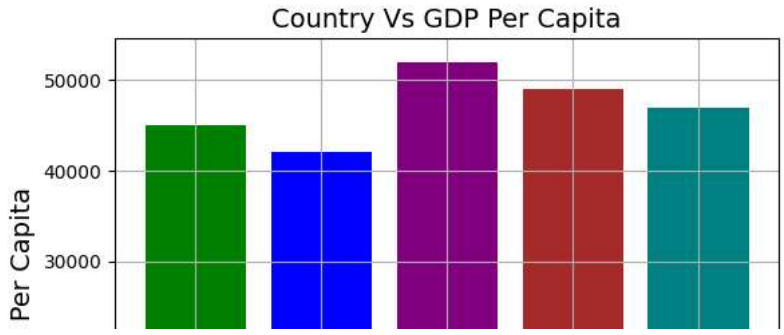
```
x_axis = ['value_1', 'value_2', 'value_3', ...]

y_axis = ['value_1', 'value_2', 'value_3', ...]

plt.bar(x_axis, y_axis)

plt.title('title name')

plt.xlabel('x_axis name')

plt.ylabel('y_axis name')

plt.show()
```

```
import matplotlib.pyplot as plt
```

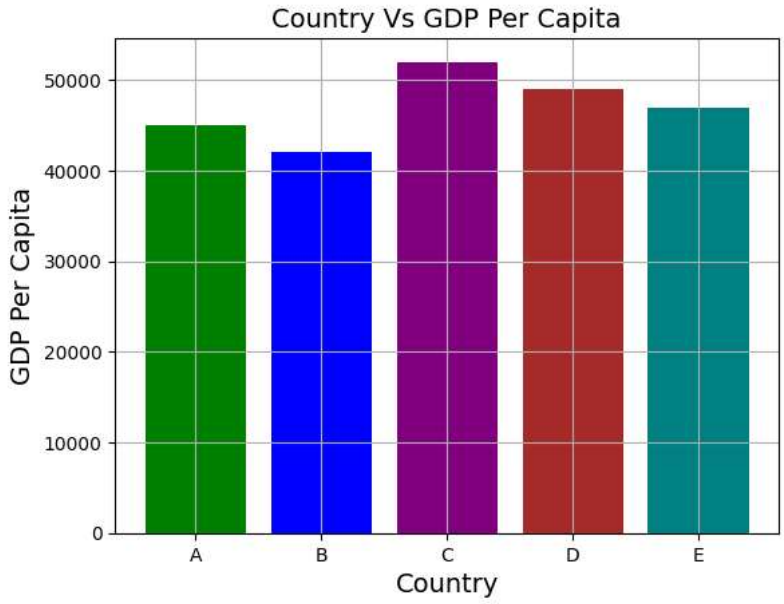
```
import matplotlib.pyplot as plt

country = ['A', 'B', 'C', 'D', 'E']
gdp_per_capita = [45000, 42000, 52000, 49000, 47000]

plt.bar(country, gdp_per_capita)
plt.title('Country Vs GDP Per Capita')
plt.xlabel('Country')
plt.ylabel('GDP Per Capita')
plt.show()
```




```
#You can further style the bar chart using this code:

country = ['A', 'B', 'C', 'D', 'E']
gdp_per_capita = [45000, 42000, 52000, 49000, 47000]

colors = ['green', 'blue', 'purple', 'brown', 'teal']
plt.bar(country, gdp_per_capita, color=colors)
plt.title('Country Vs GDP Per Capita', fontsize=14)
plt.xlabel('Country', fontsize=14)
plt.ylabel('GDP Per Capita', fontsize=14)
plt.grid(True)
plt.show()
```




```
#Bar Chart with DataFrame
data = {'country': ['A', 'B', 'C', 'D', 'E'],
        'gdp_per_capita': [45000, 42000, 52000, 49000, 47000]
       }
df = pd.DataFrame(data)

colors = ['green','blue','purple','brown','teal']
plt.bar(df['country'], df['gdp_per_capita'], color=colors)
plt.title('Country Vs GDP Per Capita', fontsize=14)
plt.xlabel('Country', fontsize=14)
plt.ylabel('GDP Per Capita', fontsize=14)
plt.grid(True)
plt.show()
```




```
data = {'year': [1920, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010],
        'unemployment_rate': [9.8, 12, 8, 7.2, 6.9, 7, 6.5, 6.2, 5.5, 6.3]
       }

df = pd.DataFrame(data)

plt.plot(df['year'], df['unemployment_rate'], color='red', marker='o')
plt.title('unemployment rate vs year', fontsize=14)
plt.xlabel('year', fontsize=14)
plt.ylabel('unemployment rate', fontsize=14)
plt.grid(True)
plt.show()
```


```
my_data = [300, 500, 700]
my_labels = 'Tasks Pending', 'Tasks Ongoing', 'Tasks Completed'
plt.pie(my_data, labels=my_labels, autopct='%1.1f%%')
plt.title('My Tasks')
plt.axis('equal')
plt.show()
```


```
unemployment_rate = [6.1, 5.8, 5.7, 5.7, 5.8, 5.6, 5.5, 5.3, 5.2, 5.2]
index_price = [1500, 1520, 1525, 1523, 1515, 1540, 1545, 1560, 1555, 1565]

plt.scatter(unemployment_rate, index_price, color='green')
plt.title('Unemployment Rate Vs Index Price', fontsize=14)
plt.xlabel('Unemployment Rate', fontsize=14)
plt.ylabel('Index Price', fontsize=14)
plt.grid(True)
plt.show()
```

Plot the histogram using matplotlib

Template

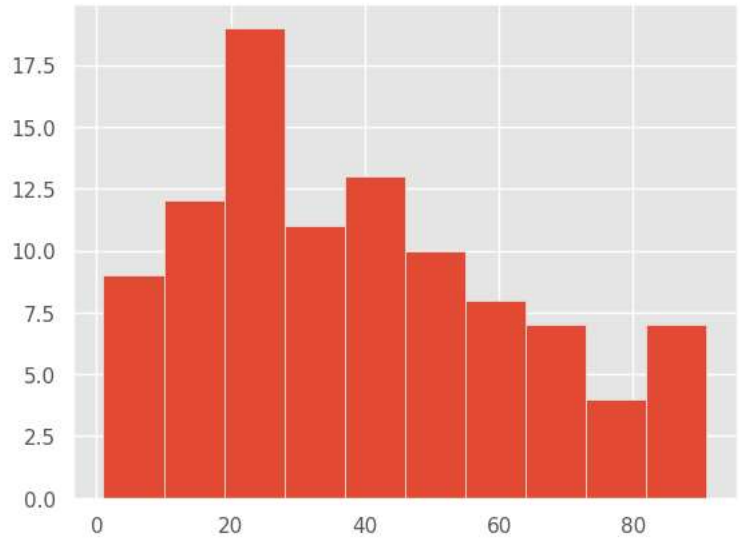
import matplotlib.pyplot as plt

x = [value1, value2, value3,....]

plt.hist(x, bins=number of bins)

plt.show()

```
x = [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, 26, 27, 27, 27, 27, 27,
     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, 53, 54, 55, 55, 56, 57, 58, 60,
     61, 63, 64, 65, 66, 68, 70, 71, 72, 74,
     75, 77, 81, 83, 84, 87, 89, 90, 90, 91
     ]

plt.hist(x, bins=10)
plt.style.use('ggplot')
plt.show()
```



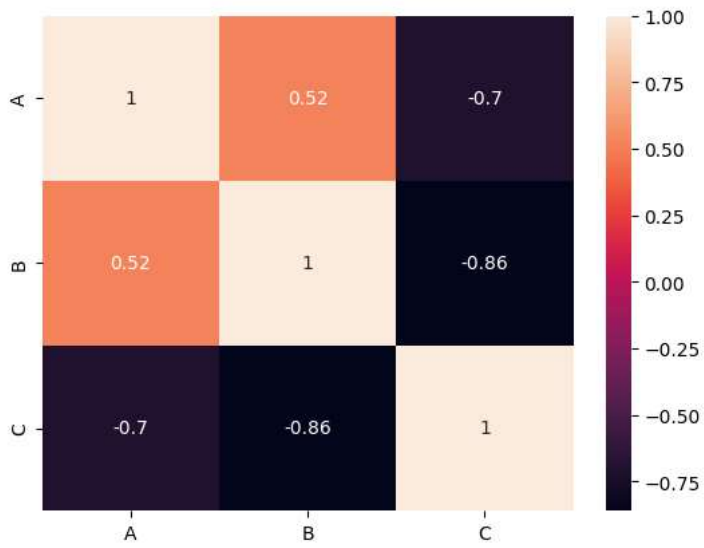
Visual Representation of the Correlation Matrix using Seaborn and Matplotlib

```
data = {'A': [45, 37, 42, 35, 39],
        'B': [38, 31, 26, 28, 33],
        'C': [10, 15, 17, 21, 12]
        }

df = pd.DataFrame(data)
print(df)
corr_matrix = df.corr()
print(corr_matrix)
```

```
    A   B   C
0  45  38  10
1  37  31  15
2  42  26  17
3  35  28  21
4  39  33  12
          A         B         C
```

```
A  1.000000  0.518457 -0.701886
B  0.518457  1.000000 -0.860941
C -0.701886 -0.860941  1.000000
```

```
import seaborn as sn
import matplotlib.pyplot as plt
sn.heatmap(corr_matrix, annot=True)
plt.show()
```

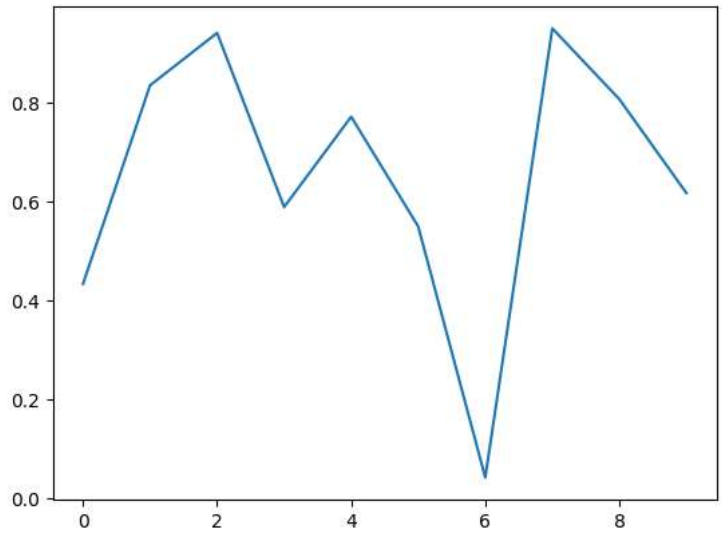



```
x=np.random.random((10,1))
x
```

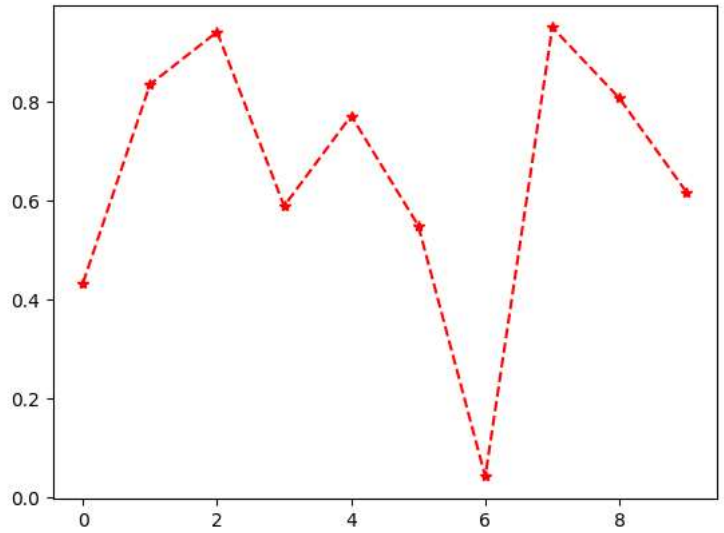
```
array([[0.04848276],
       [0.25505966],
       [0.39925536],
       [0.26920142],
       [0.99046617],
       [0.46235661],
       [0.89869506],
       [0.1462322 ],
       [0.6987918 ],
       [0.81604101]])
```

Double-click (or enter) to edit

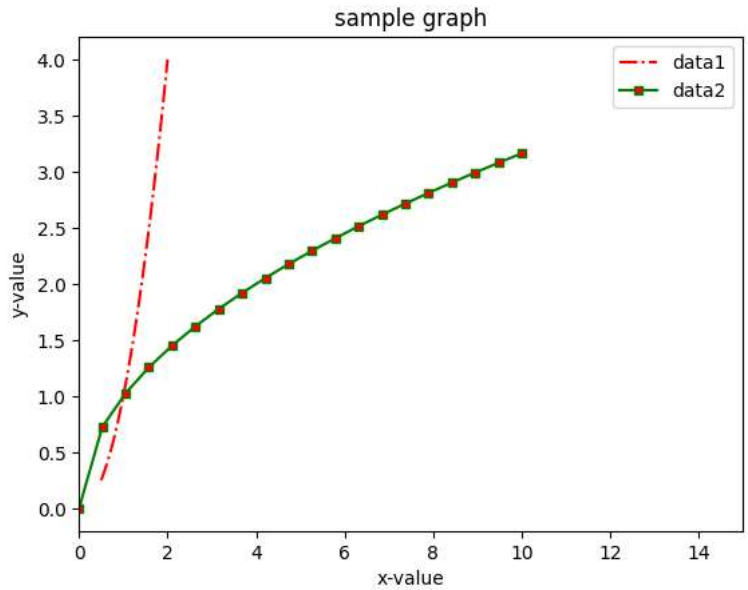
```
#plot line
plt.plot(x)
plt.show()
```



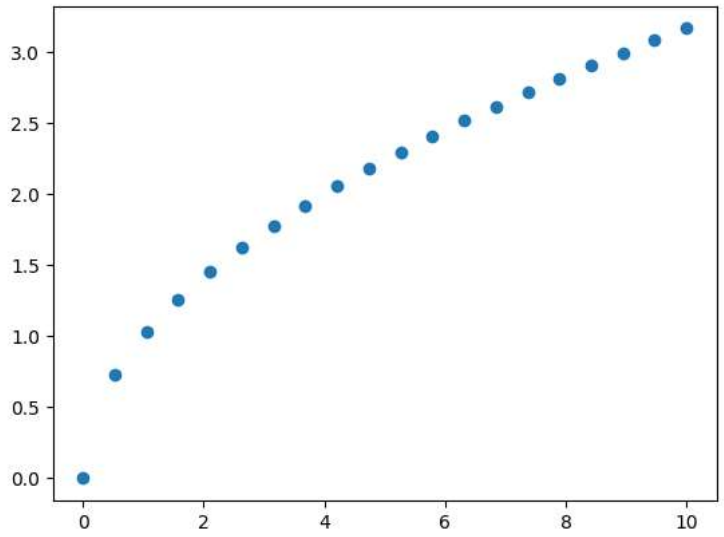
```
#plt.plot(x,'bD')
plt.plot(x,'r*--')
plt.show()
```




```
import numpy as np
#y=sqrt(x)----data2
x=np.linspace(0,10,20)
y=np.power(x,0.5)

#z=square(a) ----data1
a=np.linspace(0.5,2,20)
z=np.square(a)

#plt.plot(a,z,label='data1',color='green',linestyle='dashed',marker='s',markerfacecolor='red',markersize=10)
plt.plot(a,z,label='data1',color='r',linestyle='-.')
#plt.plot(a,z,label='data1','r-.')
plt.plot(x,y,label='data2',color='g',marker='s',markerfacecolor='r',markersize=5)


plt.legend()
plt.title('sample graph')
plt.xlabel('x-value')
plt.ylabel('y-value')
plt.xlim(0,15)
plt.show()
```

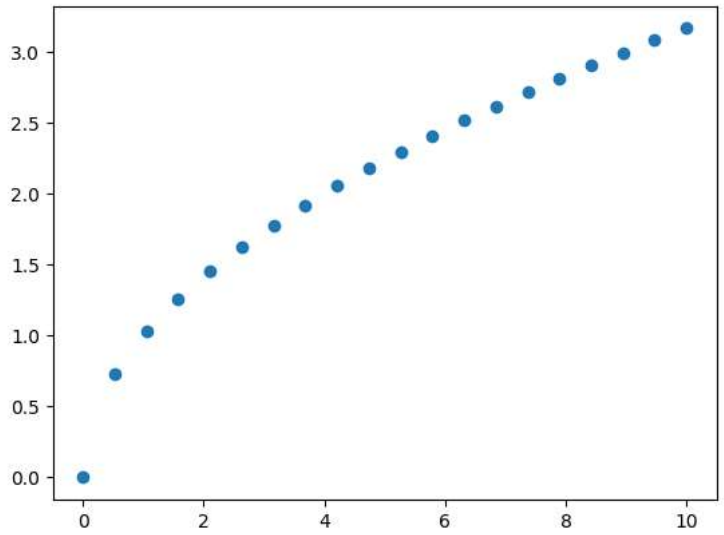


```
plt.plot(x,y,'o')
plt.show()
```




```
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x7b306cae0fd0>
```




```
#plt.hist(y)
plt.hist(y,bins=3)
plt.hist(y, edgecolor='r')
plt.show()
```

```
import seaborn as sns
sns.set()
sns.lineplot(x,y)
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-34-d3fcbf03b7aa> in <cell line: 1>()
----> 1 sns.lineplot(x,y)
      2 plt.show()

TypeError: lineplot() takes from 0 to 1 positional arguments but 2 were given
```

SEARCH STACK OVERFLOW

```
!pip install seaborn==0.9.0
```

```
Requirement already satisfied: seaborn==0.9.0 in c:\users\user\anaconda3\lib\site-packages
Requirement already satisfied: scipy>=0.14.0 in c:\users\user\anaconda3\lib\site-packages (from seaborn==0.9.0)
Requirement already satisfied: pandas>=0.15.2 in c:\users\user\anaconda3\lib\site-packages (from seaborn==0.9.0)
Requirement already satisfied: matplotlib>=1.4.3 in c:\users\user\anaconda3\lib\site-packages (from seaborn==0.9.0)
Requirement already satisfied: numpy>=1.9.3 in c:\users\user\anaconda3\lib\site-packages (from seaborn==0.9.0)
Requirement already satisfied: python-dateutil>=2 in c:\users\user\anaconda3\lib\site-packages (from pandas>=0.15.2->seaborn==0.9.0)
Requirement already satisfied: pytz>=2011k in c:\users\user\anaconda3\lib\site-packages (from pandas>=0.15.2->seaborn==0.9.0)
Requirement already satisfied: six>=1.10 in c:\users\user\anaconda3\lib\site-packages (from matplotlib>=1.4.3->seaborn==0.9.0)
Requirement already satisfied: cycler>=0.10 in c:\users\user\anaconda3\lib\site-packages (from matplotlib>=1.4.3->seaborn==0.9.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\user\anaconda3\lib\site-packages (from matplotlib>=1
You are using pip version 9.0.1, however version 19.2.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

```
import pandas as pd
```

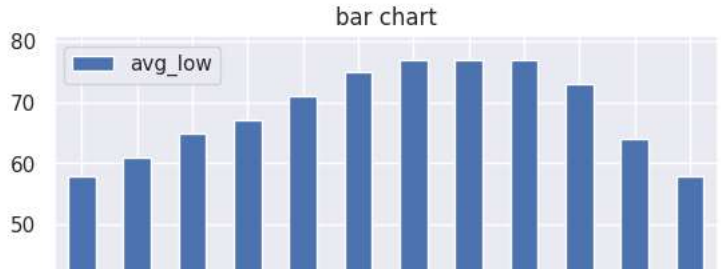
```
filename='/content/weather.csv'
df=pd.read_csv(filename)
df.head()
```

	month	avg_low	avg_high	record_high	record_low	avg_preci
0	Jan	58	42	74	22	2.95
1	Feb	61	45	78	26	3.02
2	Mar	65	48	84	25	2.34
3	Apr	67	50	92	28	1.02
4	May	71	53	98	35	0.48

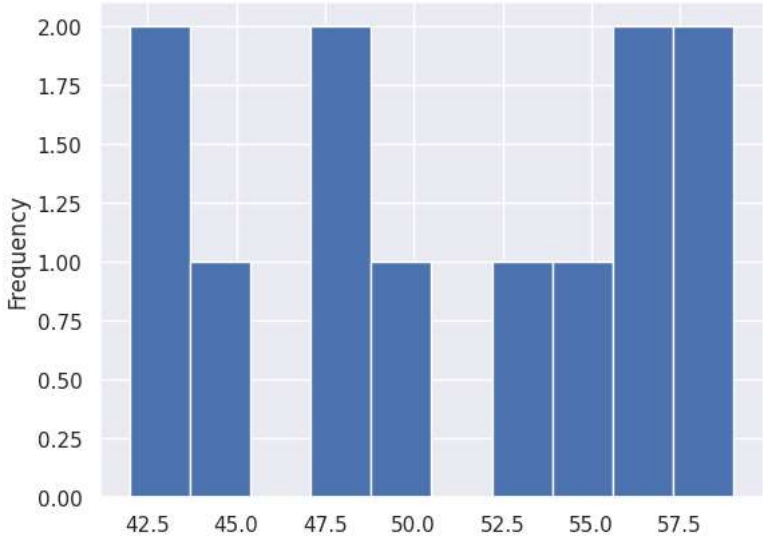
```
df.columns=df.columns.str.lstrip()
```

```
%matplotlib inline
df.plot(x='month',y='avg_low',kind='bar',title="bar chart",legend=True)
```

```
<Axes: title={'center': 'bar chart'}, xlabel='month'>
```




```
df['avg_high'].plot.hist(bins=10)
plt.style.use('ggplot')
```

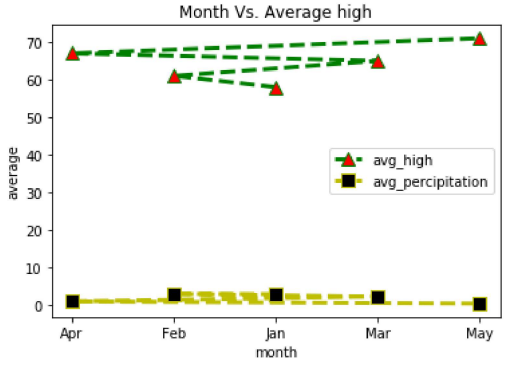



```
df.columns
```

```
Index(['month', 'avg_high', 'avg_low', 'record_high', 'record_low',
       'avg_percipitation'],
      dtype='object')
```

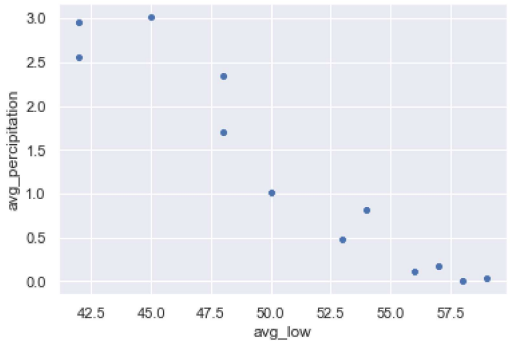
```
x=df.iloc[0:5,0]
#print(x)
y=df.iloc[0:5,1]
z=df.iloc[0:5,5]
plt.plot(x,y,'g',linestyle='dashed',marker='^',markerfacecolor='r',markersize=10,linewidth=3)
plt.plot(x,z,'y',linestyle='dashed',marker='s',markerfacecolor='k',markersize=10,linewidth=3)
plt.title('Mo
          nth Vs. Average high')
plt.xlabel('month')
plt.ylabel('average')
plt.legend()
```

```
<matplotlib.legend.Legend at 0xb3f12fd0b8>
```

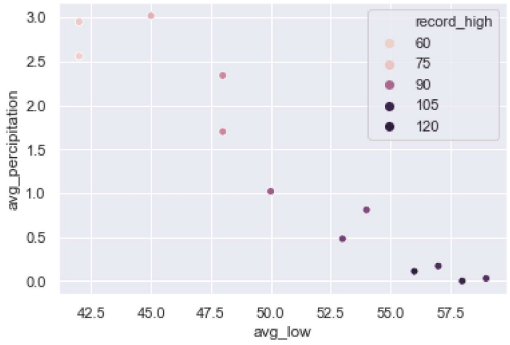



```
plt.bar()
```

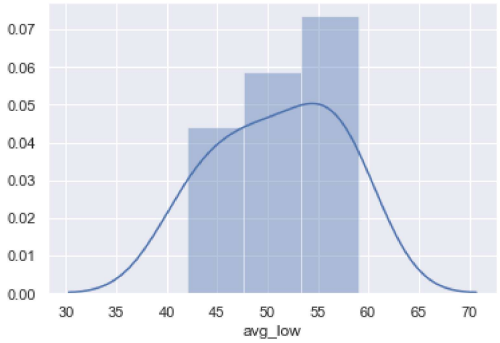
```
ax=sns.scatterplot(x='avg_low',y='avg_percipitation',data=df)
```




```
ax=sns.scatterplot(x='avg_low',y='avg_percipitation',hue='record_high',data=df)
```




```
ax=sns.distplot(df['avg_low'])
```

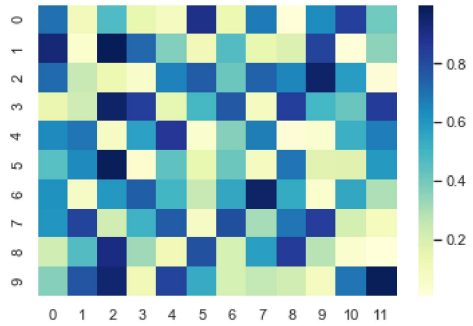



```
uniform_data=np.random.rand(10,12)
uniform_data
```

```
array([[ 0.70344773,  0.09812472,  0.47394285,  0.14278102,  0.0788329 ,
         0.89431206,  0.1272322 ,  0.68077213,  0.01209378,  0.63568675,
         0.83749171,  0.40730232],
       [ 0.93466059,  0.04523419,  0.9979945 ,  0.72265328,  0.37257474,
         0.11917064,  0.46693486,  0.14111717,  0.1910405 ,  0.82762219,
         0.01586194,  0.35782062],
       [ 0.71932964,  0.2515214 ,  0.12621854,  0.05659261,  0.6571262 ,
         0.75553751,  0.41714067,  0.73944558,  0.65076721,  0.96486861,
         0.58474553,  0.01971113],
       [ 0.1364273 ,  0.23056867,  0.96311641,  0.83965569,  0.15839363,
         0.49305324,  0.76700227,  0.08072943,  0.84225936,  0.49407127,
         0.42078366,  0.84942456],
       [ 0.64304758,  0.68778838,  0.06169656,  0.57469241,  0.86684031,
         0.04165984,  0.36245255,  0.67403267,  0.02000103,  0.04253173,
         0.52754396,  0.67121135],
       [ 0.45801261,  0.64227595,  0.98808018,  0.03482514,  0.44240673,
         0.14099633,  0.41290858,  0.08126955,  0.68909843,  0.17468816,
         0.17889924,  0.59792319],
       [ 0.61974019,  0.06091456,  0.60603438,  0.75136733,  0.49780655,
         0.24708835,  0.54810716,  0.96333119,  0.54266217,  0.05046595,
```

```
  0.55705436,  0.29461197],
[ 0.60644019,  0.8295255 ,  0.22031545,  0.50988299,  0.75389876,
  0.0610872 ,  0.796234  ,  0.31212852,  0.69106179,  0.84482262,
  0.20955993,  0.09203925],
[ 0.222307  ,  0.46969717,  0.91428803,  0.3247527 ,  0.10868443,
  0.78816422,  0.20557491,  0.5768842 ,  0.84823093,  0.27968698,
  0.03874419,  0.0049343 ],
[ 0.36462162,  0.77501611,  0.9471405 ,  0.11892888,  0.82977494,
  0.5354319 ,  0.20357565,  0.26047622,  0.22155178,  0.08392495,
  0.68908784,  0.99160178]])
```

```
ax=sns.heatmap(uniform_data,cmap='YlGnBu')
```




```
import pandas as pd
```

```
df=pd.read_csv('/content/weather.csv')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   month               12 non-null     object 
 1    avg_high           12 non-null     int64  
 2    avg_low            12 non-null     int64  
 3    record_high        12 non-null     int64  
 4    record_low         12 non-null     int64  
 5    avg_percipitation  12 non-null     float64
dtypes: float64(1), int64(4), object(1)
memory usage: 704.0+ bytes
```

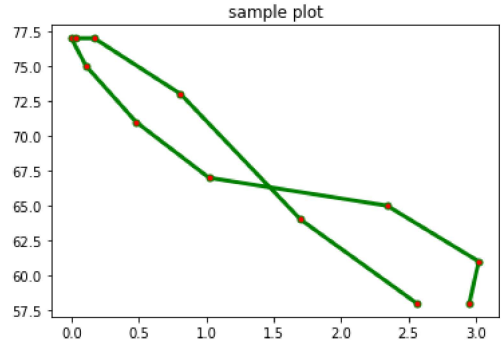
```
df.columns=df.columns.str.lstrip()
```

```
df.columns
```

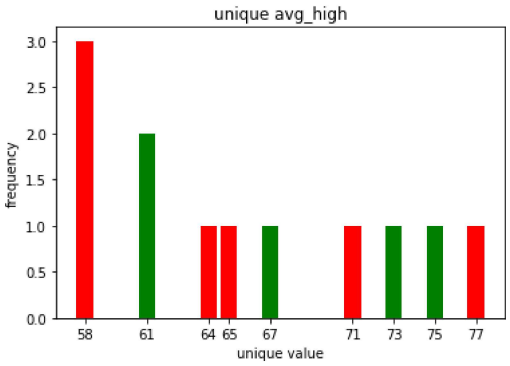
```
Index(['month', 'avg_high', 'avg_low', 'record_high', 'record_low',
       'avg_percipitation'],
      dtype='object')
```

```
plt.plot(df['avg_percipitation'], df['avg_high'],color='g',linestyle='-',linewidth=3,marker='o',markerfacecolor='r',markersize=5)
plt.title('sample plot')
#plt.xlabel('x')
```

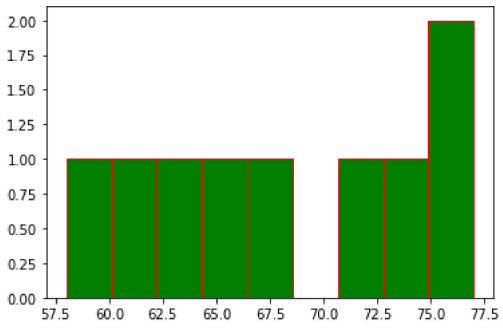
```
Text(0.5, 1.0, 'sample plot')
```



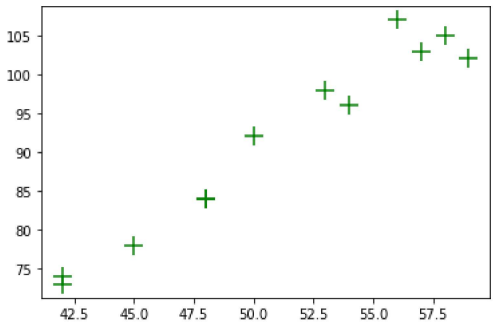
```
u=df['avg_high'].unique()

u
c=df['avg_high'].value_counts()
c

t_lab=u
#plt.bar(u,c,tick_label=t_lab)
plt.bar(u,c,tick_label=t_lab,color=['r','g'])
plt.title('unique avg_high')
plt.xlabel('unique value')
plt.ylabel('frequency')
plt.show()
```



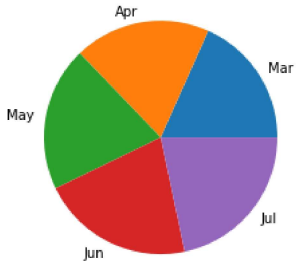

```
range=(min(u),max(u))
bins=9
#plt.hist(u)
plt.hist(u,bins,range,color='g',histtype='bar',edgecolor='r')
plt.show()
```




```
plt.scatter(df['avg_low'],df['record_high'],marker='+',s=200,color='g')
plt.show()
```




```
x=df['month'].iloc[2:7]
y=df['avg_high'].iloc[2:7]


plt.pie(y,labels=x)
plt.show()
```




```
plt.pie(y,labels=x,explode=[0,1,0,0,0],colors=['red','green','blue','yellow','cyan'],startangle=10)
#plt.legend()
plt.show()
```