In [ ]:

```
# Data Visualization - Plot
```

In [1]:

```
# import the required libraries
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
```

In [2]:

```
randomNumber = np.random.rand(10)
print(randomNumber)
```

```
[0.60670869 0.6798678  0.41683952 0.88919799 0.96987478 0.33041737
 0.10840018 0.96670519 0.54566654 0.24966939]
```

In [4]:

```
# style the plot
style.use('ggplot')
plt.plot(randomNumber,'b',label='line one',linewidth=2)
plt.xlabel('Range')
plt.ylabel('Number')
plt.title('First Plot')

plt.legend()
plt.show()
```

In [5]:

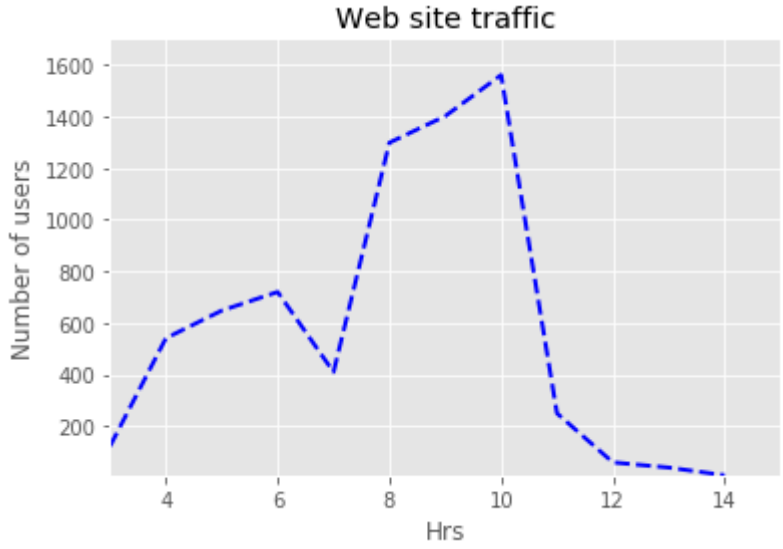
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
```

In [9]:

```
# Website traffic data
web_customers = [123,541,648,720,410,1298,1401,1560,250,60,40,10]
time_hrs = [3,4,5,6,7,8,9,10,11,12,13,14]
```

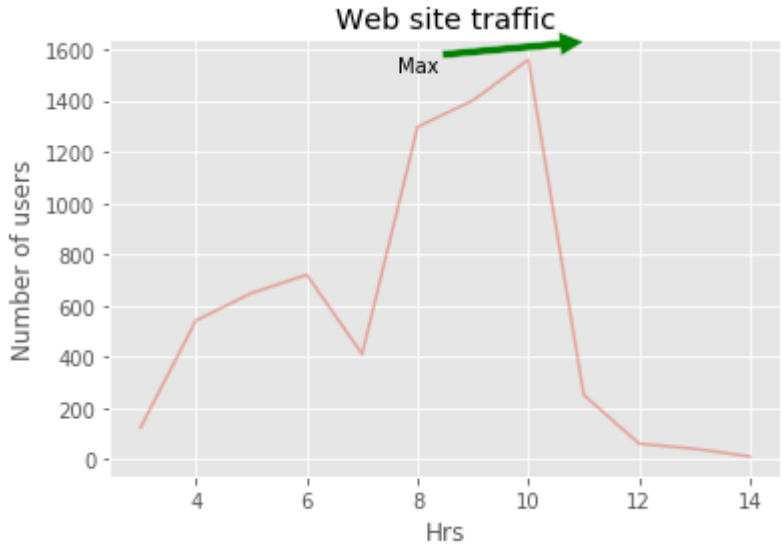
In [17]:

```
# Style the plot
style.use('ggplot')
plt.plot(time_hrs,web_customers,color = 'b',linestyle = '--',linewidth = 2)
plt.axis([3,15,10,1700])
plt.title('Web site traffic')
plt.xlabel('Hrs')
plt.ylabel('Number of users')

plt.show()
```



In [19]:

```
# Style the plot
style.use('ggplot')
plt.plot(time_hrs,web_customers,alpha = .4)
plt.title('Web site traffic')
# Annotate
plt.annotate('Max',ha='center',va='bottom',xytext=(8,1500),xy=(11,1630),arrowprops = {'face
# Set label
plt.xlabel('Hrs')
plt.ylabel('Number of users')

plt.show()
```



In [ ]:

```
# Matplot - SubPlot
```

In [24]:

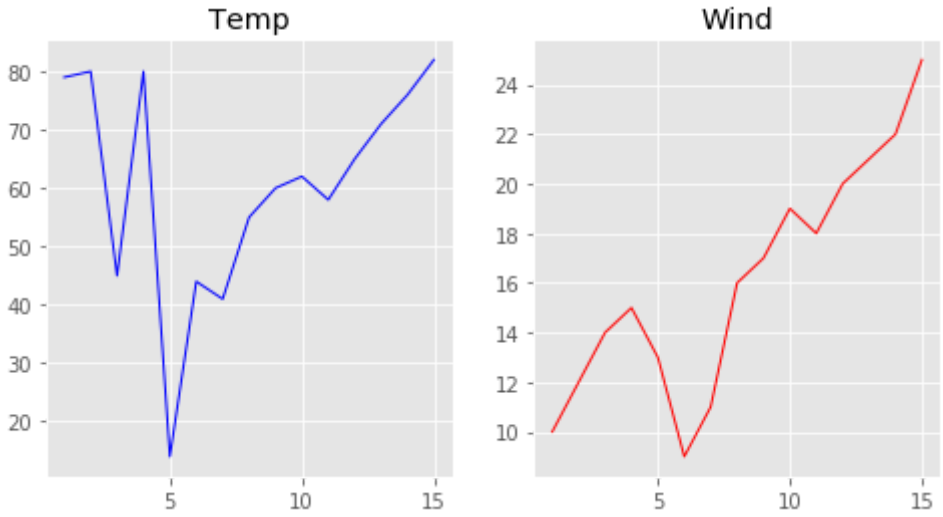
```
# import the required libraries
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
# define temp, wind, time, humidity and precipitation data
temp_data = [79,80,45,80,14,44,41,55,60,62,58,65,71,76,82]
wind_data = [10,12,14,15,13,9,11,16,17,19,18,20,21,22,25]
time_hrs = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
humidity_data = [78,41,46,47,72,81,43,55,40,65,58,85,42,69,87]
precipitation_data = [26,41,48,47,42,55,49,58,60,65,74,82,81,53,87]
```

In [25]:

```
# draw subplots for (1,2,1) and (1,2,2)
plt.figure(figsize=(8,4))
plt.subplots_adjust(hspace=.25)
plt.subplot(1,2,1)
plt.title('Temp')
plt.plot(time_hrs,temp_data,color = 'b',linestyle='-',linewidth=1)
plt.subplot(1,2,2)
plt.title('Wind')
plt.plot(time_hrs,wind_data,color = 'r',linestyle='-',linewidth=1)
```

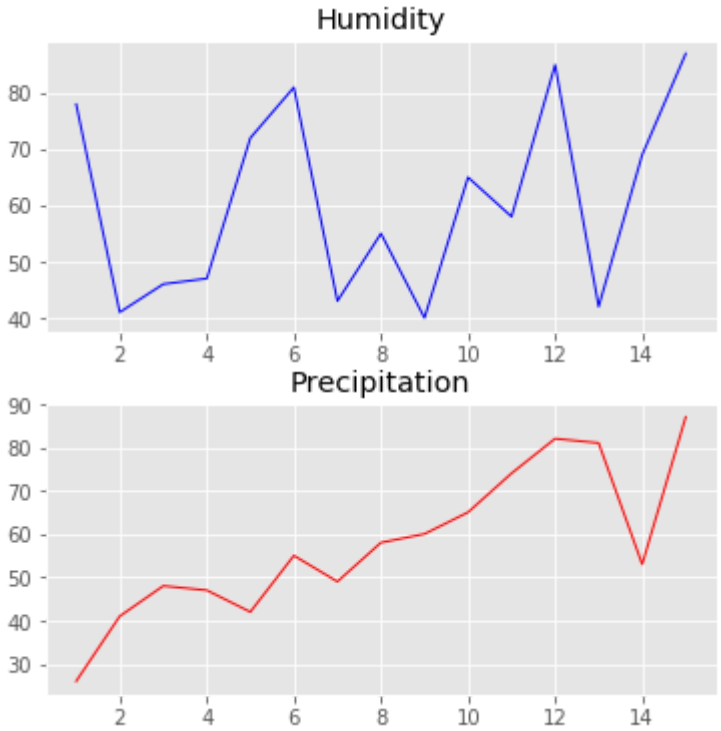
Out[25]:

```
[<matplotlib.lines.Line2D at 0x2462b846c08>]
```



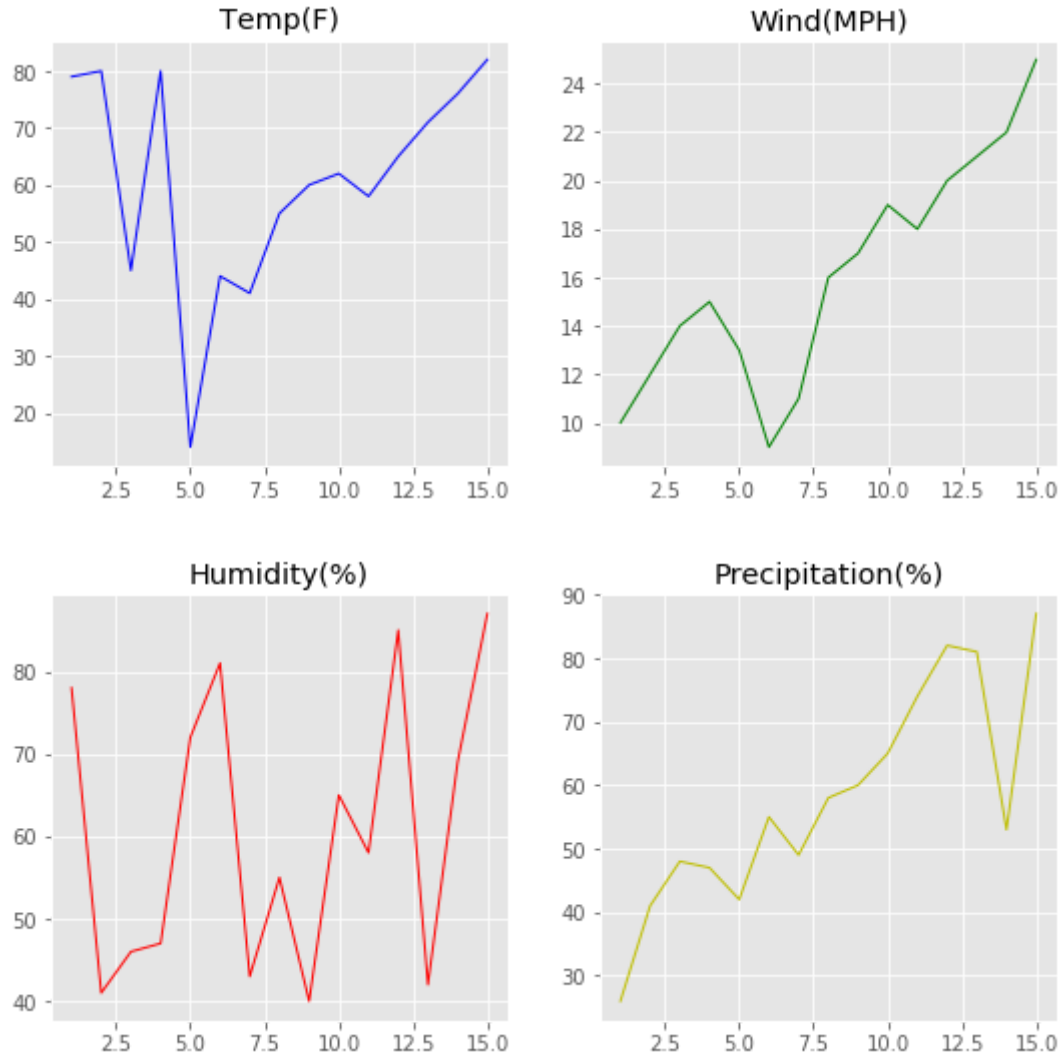
In [27]:

```
# draw for (2,1,1) and (2,1,2)
plt.figure(figsize=(6,6))
plt.subplots_adjust(hspace=.25)
plt.subplot(2,1,1)
plt.title('Humidity')
plt.plot(time_hrs,humidity_data,color='b',linestyle = '-',linewidth='1')
plt.subplot(2,1,2)
plt.title('Precipitation')
plt.plot(time_hrs,precipitation_data,color='r',linestyle='-',linewidth='1')
plt.show()
```



In [30]:

```
# draw subplots for (2,2,1),(2,2,2),(2,2,3) and (2,2,4)
plt.figure(figsize=(9,9))
plt.subplots_adjust(hspace=.3)
plt.subplot(2,2,1)
plt.title('Temp(F)')
plt.plot(time_hrs,temp_data,color='b',Linestyle='-',linewidth='1')
plt.subplot(2,2,2)
plt.title('Wind(MPH)')
plt.plot(time_hrs,wind_data,color='g',linestyle='-',linewidth='1')
plt.subplot(2,2,3)
plt.title('Humidity(%)')
plt.plot(time_hrs,humidity_data,color='r',linestyle='-',linewidth='1')
plt.subplot(2,2,4)
plt.title('Precipitation(%)')
plt.plot(time_hrs,precipitation_data,color='y',linestyle='-',linewidth='1')

plt.show()
```



In [2]:

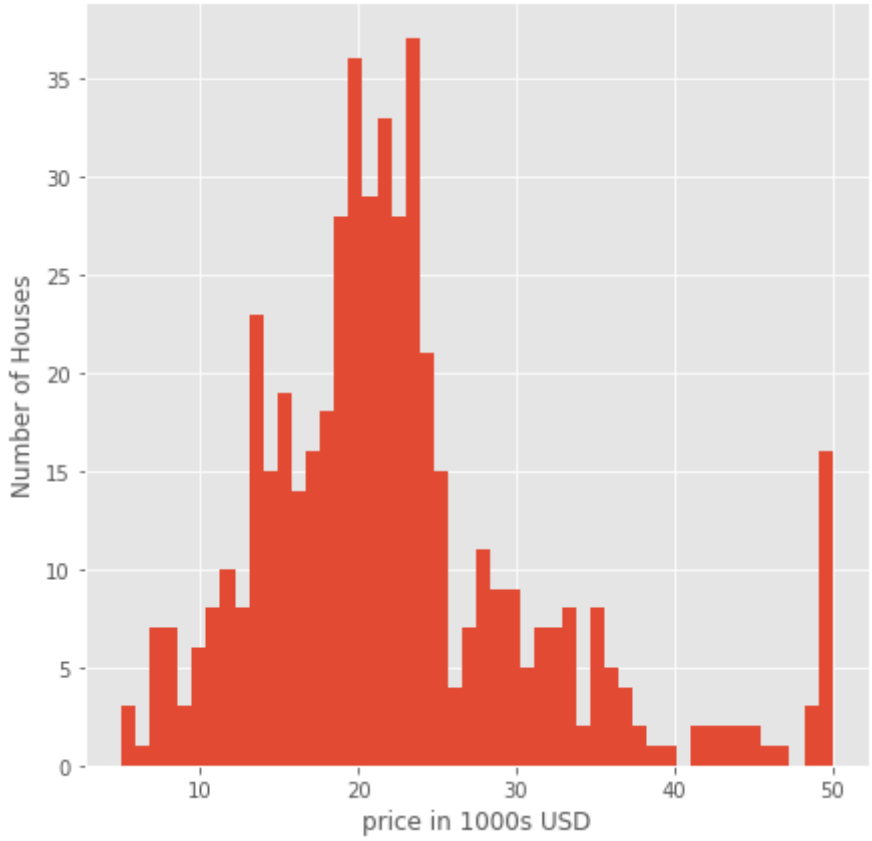
```
# Histogram and Scatter Plots
from sklearn.datasets import load_boston
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline

boston_real_state_data = load_boston()
print(boston_real_state_data)
```

```
{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690
e+02,
        4.9800e+00],
       [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
        9.1400e+00],
       [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
        4.0300e+00],
       ...,
       [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
        5.6400e+00],
       [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
        6.4800e+00],
       [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
        7.8800e+00]]), 'target': array([24. , 21.6, 34.7, 33.4, 36.2, 28.7,
22.9, 27.1, 16.5, 18.9, 15. ,
       18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
       15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
       13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
       21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
       35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
       19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
       20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
       23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
       33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
       21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
       20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
       23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
       15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
       17. , 15.6, 13.1, 41.3, 24.3, 23.3, 27. , 50. , 50. , 50. , 22.7,
       25. , 50. , 23.8, 23.8, 22.3, 17.4, 19.1, 23.1, 23.6, 22.6, 29.4,
       23.2, 24.6, 29.9, 37.2, 39.8, 36.2, 37.9, 32.5, 26.4, 29.6, 50. ,
       32. , 29.8, 34.9, 37. , 30.5, 36.4, 31.1, 29.1, 50. , 33.3, 30.3,
       34.6, 34.9, 32.9, 24.1, 42.3, 48.5, 50. , 22.6, 24.4, 22.5, 24.4,
       20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7, 21.5, 23. ,
       26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7, 31.5, 24.3,
       31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3, 22. , 20.1,
       22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4, 24.8, 29.6,
       42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1, 48.8, 31. ,
       36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4, 35.2, 32.4,
       32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. , 32.2, 22. ,
       20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7, 28.6, 27.1,
       20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4, 33.4, 28.2,
       22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8, 19.8, 23.1,
       21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2, 19.3, 22.6,
       19.8, 17.1, 19.4, 22.2, 20.7, 21.1, 19.5, 18.5, 20.6, 19. , 18.7,
       32.7, 16.5, 23.9, 31.2, 17.5, 17.2, 23.1, 24.5, 26.6, 22.9, 24.1,
       18.6, 30.1, 18.2, 20.6, 17.8, 21.7, 22.7, 22.6, 25. , 19.9, 20.8,
       16.8, 21.9, 27.5, 21.9, 23.1, 50. , 50. , 50. , 50. , 50. , 13.8,
       13.8, 15. , 13.9, 13.3, 13.1, 10.2, 10.4, 10.9, 11.3, 12.3,  8.8,
        7.2, 10.5,  7.4, 10.2, 11.5, 15.1, 23.2,  9.7, 13.8, 12.7, 13.1,
```

```
        12.5,  8.5,  5. ,  6.3,  5.6,  7.2, 12.1,  8.3,  8.5,  5. , 11.9,
        27.9, 17.2, 27.5, 15. , 17.2, 17.9, 16.3,  7. ,  7.2,  7.5, 10.4,
         8.8,  8.4, 16.7, 14.2, 20.8, 13.4, 11.7,  8.3, 10.2, 10.9, 11. ,
         9.5, 14.5, 14.1, 16.1, 14.3, 11.7, 13.4,  9.6,  8.7,  8.4, 12.8,
        10.5, 17.1, 18.4, 15.4, 10.8, 11.8, 14.9, 12.6, 14.1, 13. , 13.4,
        15.2, 16.1, 17.8, 14.9, 14.1, 12.7, 13.5, 14.9, 20. , 16.4, 17.7,
        19.5, 20.2, 21.4, 19.9, 19. , 19.1, 19.1, 20.1, 19.9, 19.6, 23.2,
        29.8, 13.8, 13.3, 16.7, 12. , 14.6, 21.4, 23. , 23.7, 25. , 21.8,
        20.6, 21.2, 19.1, 20.6, 15.2,  7. ,  8.1, 13.6, 20.1, 21.8, 24.5,
        23.1, 19.7, 18.3, 21.2, 17.5, 16.8, 22.4, 20.6, 23.9, 22. , 11.9]),
'feature_names': array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE',
'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7'), 'DESCR': ".. _boston_d
ataset:\n\nBoston house prices dataset\n---------------------------\n\n**Dat
a Set Characteristics:**  \n\n    :Number of Instances: 506 \n\n    :Number
of Attributes: 13 numeric/categorical predictive. Median Value (attribute 1
4) is usually the target.\n\n    :Attribute Information (in order):\n
- CRIM     per capita crime rate by town\n        - ZN       proportion of r
esidential land zoned for lots over 25,000 sq.ft.\n        - INDUS    propor
tion of non-retail business acres per town\n        - CHAS     Charles River
dummy variable (= 1 if tract bounds river; 0 otherwise)\n        - NOX
nitric oxides concentration (parts per 10 million)\n        - RM       avera
ge number of rooms per dwelling\n        - AGE      proportion of owner-occu
pied units built prior to 1940\n        - DIS      weighted distances to fiv
e Boston employment centres\n        - RAD      index of accessibility to ra
dial highways\n        - TAX      full-value property-tax rate per $10,000\n
- PTRATIO  pupil-teacher ratio by town\n        - B        1000(Bk - 0.63)^2
where Bk is the proportion of blacks by town\n        - LSTAT    % lower sta
tus of the population\n        - MEDV     Median value of owner-occupied hom
es in $1000's\n\n    :Missing Attribute Values: None\n\n    :Creator: Harris
on, D. and Rubinfeld, D.L.\n\nThis is a copy of UCI ML housing dataset.\nhtt
ps://archive.ics.uci.edu/ml/machine-learning-databases/housing/\n\n\nThis da
taset was taken from the StatLib library which is maintained at Carnegie Mel
lon University.\n\nThe Boston house-price data of Harrison, D. and Rubinfel
d, D.L. 'Hedonic\nprices and the demand for clean air', J. Environ. Economic
s & Management,\nvol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regr
ession diagnostics\n...', Wiley, 1980.   N.B. Various transformations are us
ed in the table on\npages 244-261 of the latter.\n\nThe Boston house-price d
ata has been used in many machine learning papers that address regression\np
roblems.   \n     \n.. topic:: References\n\n   - Belsley, Kuh & Welsch, 'Re
gression diagnostics: Identifying Influential Data and Sources of Collineari
ty', Wiley, 1980. 244-261.\n   - Quinlan,R. (1993). Combining Instance-Based
and Model-Based Learning. In Proceedings on the Tenth International Conferen
ce of Machine Learning, 236-243, University of Massachusetts, Amherst. Morga
n Kaufmann.\n", 'filename': 'C:\\Users\\Hiteash\\anaconda3\\lib\\site-packag
es\\sklearn\\datasets\\data\\boston_house_prices.csv'}
```

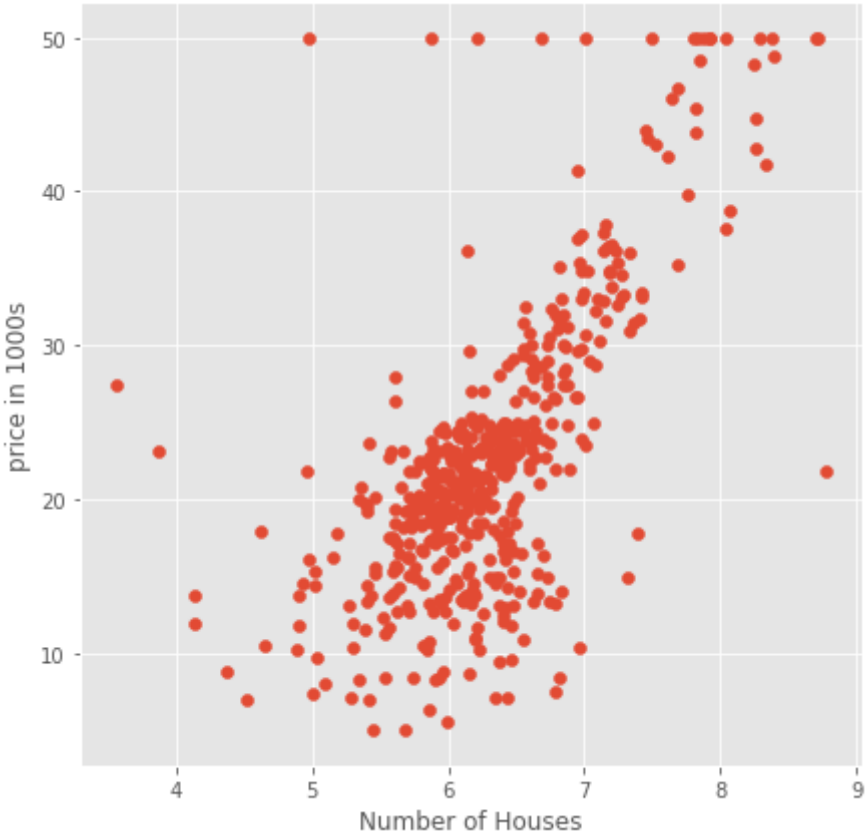
In [3]:

```
# define axis for the data
x_axis = boston_real_state_data.data
y_axis = boston_real_state_data.target
# Histogram plot
style.use('ggplot')
plt.figure(figsize=(7,7))
plt.hist(y_axis,bins=50)
plt.xlabel('price in 1000s USD')
plt.ylabel('Number of Houses')
plt.show()
```



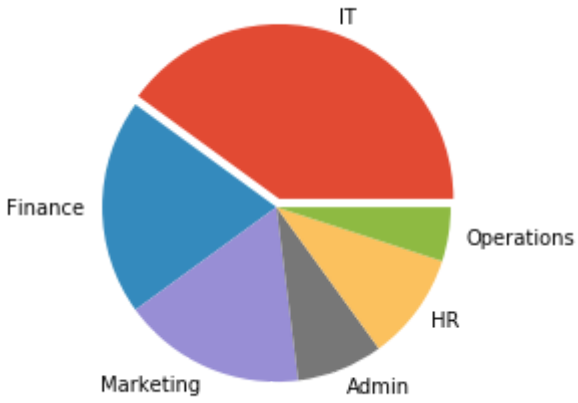
In [4]:

```
# scatter plot
style.use('ggplot')
plt.figure(figsize=(7,7))
plt.scatter(boston_real_state_data.data[:,5],boston_real_state_data.target)
plt.ylabel('price in 1000s')
plt.xlabel('Number of Houses')
plt.show()
```



In [13]:

```
# Pie Chart Plot
import matplotlib.pyplot as plt
%matplotlib inline
job_data = ['40','20','17','8','10','5']
labels = 'IT', 'Finance', 'Marketing', 'Admin', 'HR', 'Operations'
explode = (0.05,0,0,0,0,0)
plt.pie(job_data,labels=labels,explode=explode)
plt.show()
```



In [6]:

```
# Project
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df_auto_dataset = pd.read_csv('C:\Users\Hiteash\Downloads\airtravel.csv')
df_auto_dataset.head()
```

```
  File "<ipython-input-6-836f83e3e88a>", line 7
    df_auto_dataset = pd.read_csv ('C:\Users\Hiteash\Downloads\airtravel.cs
v')
                                  ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in pos
ition 2-3: truncated \UXXXXXXXX escape
```

In [ ]:

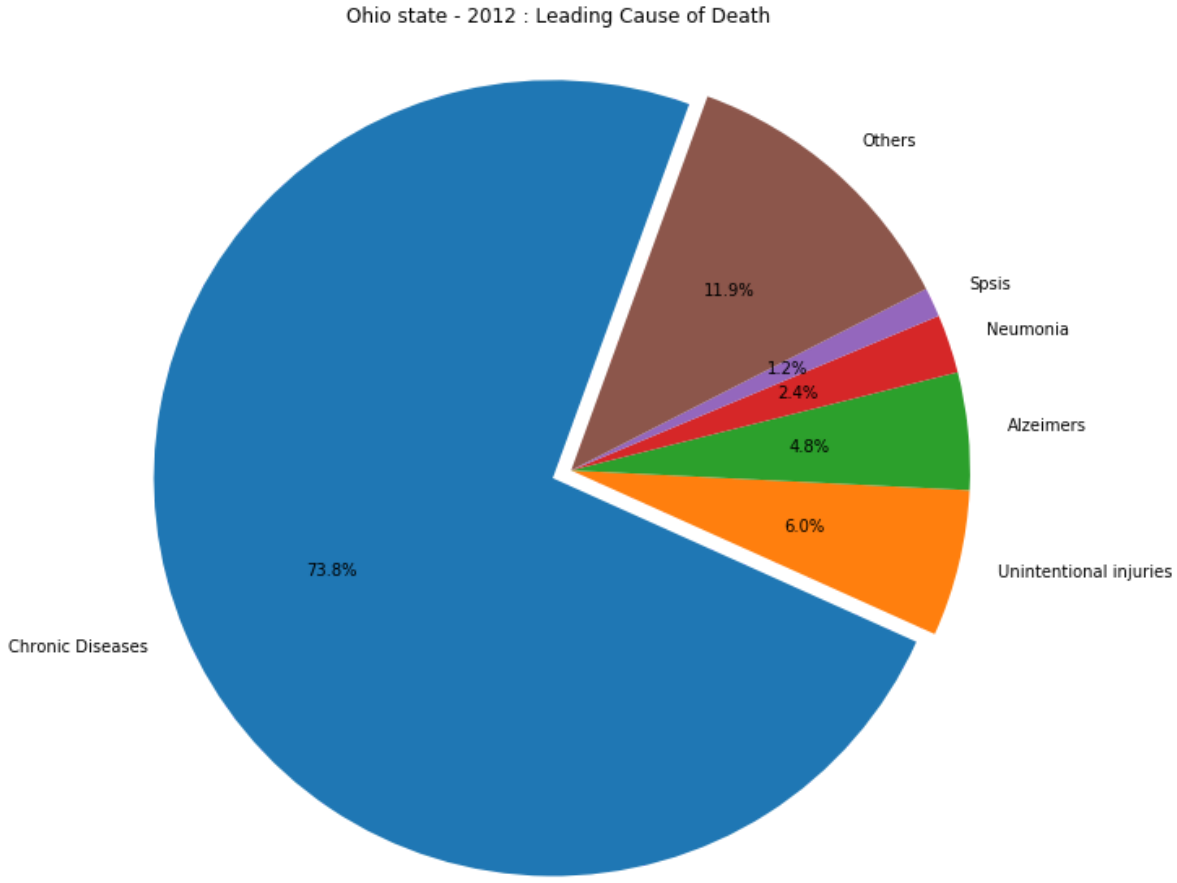
```
def origin(num):
    if num==1:
        return 'USA'
    elif num==2:
        return 'Europe'
    else:
        return 'Asia'
df_auto_dataset['origin'] = df_auto_dataset['origin'].apply(origin)

df_auto_dataset.head(30)
```

In [ ]:

```
sns.pairplot(df_auto_dataset['age','weight','origin'],hue='origin',size=4)
```

In [5]:

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

cause = 'Chronic Diseases','Unintentional injuries','Alzeimers','Influenza' and 'Neumonia',
percentile = [62,5,4,2,1,10]

plt.figure(figsize=(10,10))
explode = (0.05,0,0,0,0,0)
plt.pie(percentile,labels=cause,explode=explode,autopct='%1.1f%%',startangle=70)
plt.axis('equal')
plt.title('Ohio state - 2012 : Leading Cause of Death')
plt.show()
```



In [ ]: