In [1]:
```
# import numpy for generating random numbers
import numpy as np
# import matplotlib library
import matplotlib.pyplot as plt
from matplotlib import style # for grid stylying
%matplotlib inline # for displaying plot within jupyter notebook
```

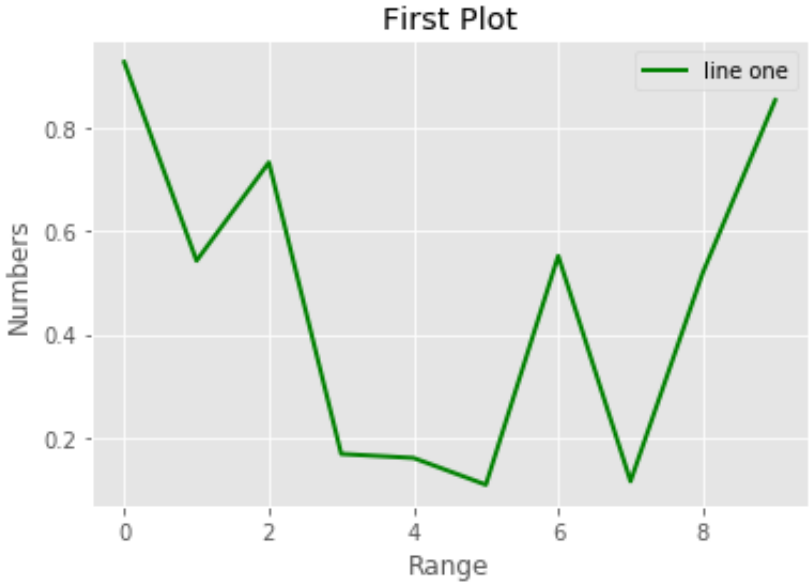
```
UsageError: unrecognized arguments: # for displaying plot within j
upyter notebook
```

In [2]:
```
# generating random numbers (total 10)
randomNumber = np.random.rand(10) # define the dataset
```

In [3]:
```
# view them
print(randomNumber)
```

```
[0.92710679 0.54251582 0.73284416 0.16842533 0.16061529 0.10871984
 0.55179089 0.11547993 0.51967332 0.85344044]
```

In [6]:
```
# select the style of the plot
style.use('ggplot')
# plot the random numbers
plt.plot(randomNumber,'g',label = 'line one',linewidth = 2)
# x-axis is the number of random numbers (index)
plt.xlabel('Range')
# y-axis is actual random numbers
plt.ylabel('Numbers')
# title of the plot
plt.title('First Plot')

plt.legend() # to show what does each line represent
plt.show() # plot the graph as output
```



In [17]:
```
# website traffic data
# no of users/visitors on the website
web_monday = [123,645,950,290,1630,1450,1034,295,465,205,80]
web_tuesday = [95,680,889,1145,1670,1323,1119,1265,510,310,110]
web_wednesday = [105,630,700,1006,1520,1124,1239,1380,580,610,230]
# time distribution - hourly
time_hrs = [7,8,9,10,11,12,13,14,15,16,17]
```

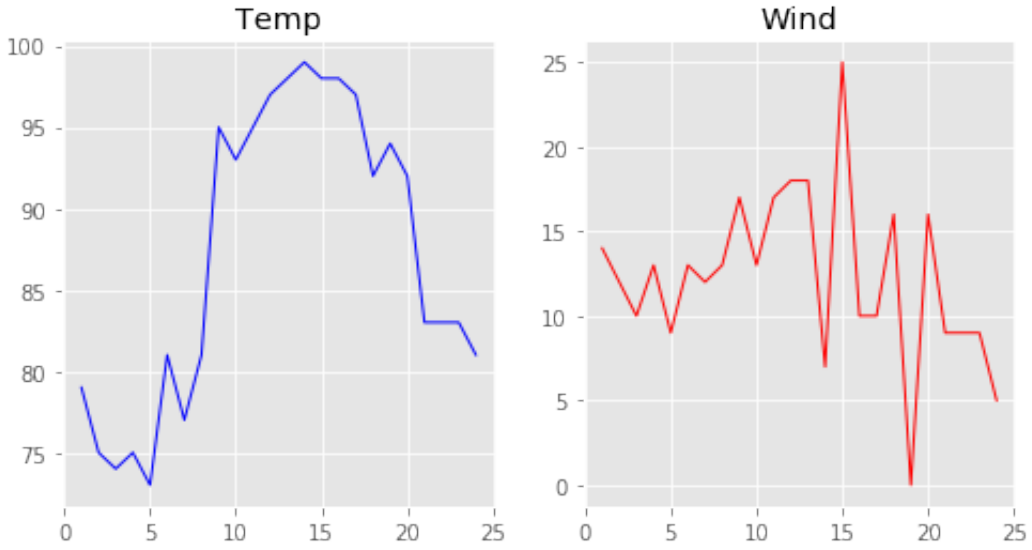
In [19]:
```
# select the style of the plot
style.use('ggplot')
# plot the website traffic data by taking x-axis as time_hrs and y-
axis as web_customers
# alpha is an attribute which controls the transparency of the line
.Lower the alpha value,more transparent the line is.
plt.plot(time_hrs,web_monday,color = 'r',label = 'Monday',linewidth
= 2,alpha = 0.4)
plt.plot(time_hrs,web_tuesday,color = 'g',label = 'Tuesday',linewid
th = 2.5,alpha = 0.4)
plt.plot(time_hrs,web_wednesday,color = 'b',label = 'Wednesday',lin
ewidth = 3,alpha = 0.4)

# set range for axis
plt.axis([6.5,17.5,50,2000])
# x-axis as hours
plt.xlabel('Hours')
# y-axis is number of users visiting website
plt.ylabel('Number of Users')
# title of the plot
plt.title('Website Traffic')

# annotate
# Max -> Annotation Text
# ha & va -> horizonatl and vertical alignment respectively
# xytext -> text position
# xy -> indicates the arrow position
# arrowprops -> indicates the properties of the arrow
plt.annotate('Max',ha = 'center',va = 'bottom',xytext = (8,1500),xy
= (11,1630),arrowprops = {'facecolor':'green'})


plt.legend() # to show what does each line represent
plt.show() # plot the graph as output
```

In [20]:
```
# matplot subplot
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
```

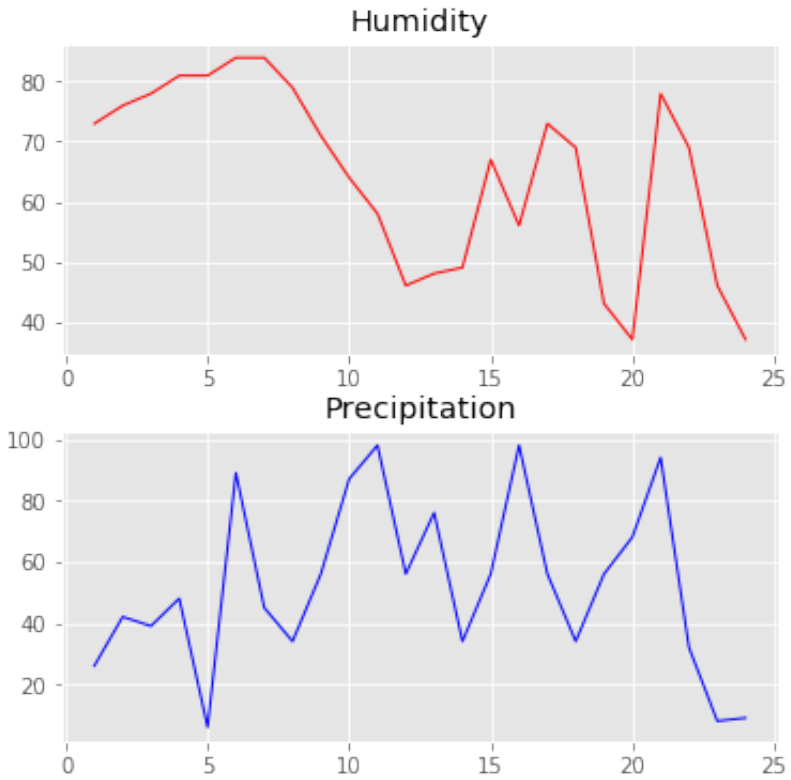
In [21]:
```
# define temp,wind,humidity,precipitation data and Time hrs data
temp_data = [79,75,74,75,73,81,77,81,95,93,95,97,98,99,98,98,97,92,
94,92,83,83,83,81]
wind_data = [14,12,10,13,9,13,12,13,17,13,17,18,18,7,25,10,10,16,0,
16,9,9,9,5]
humidity_data = [73,76,78,81,81,84,84,79,71,64,58,46,48,49,67,56,73
,69,43,37,78,69,46,37]
precip_data = [26,42,39,48,6,89,45,34,56,87,98,56,76,34,56,98,56,34
,56,68,94,32,8,9]
time_hrs = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,2
2,23,24]
```

In [24]:
```
# draw subplots for 1,2,1) and (1,2,2)
plt.figure(figsize=(8,4))
plt.subplots_adjust(hspace = 0.25)
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)
```

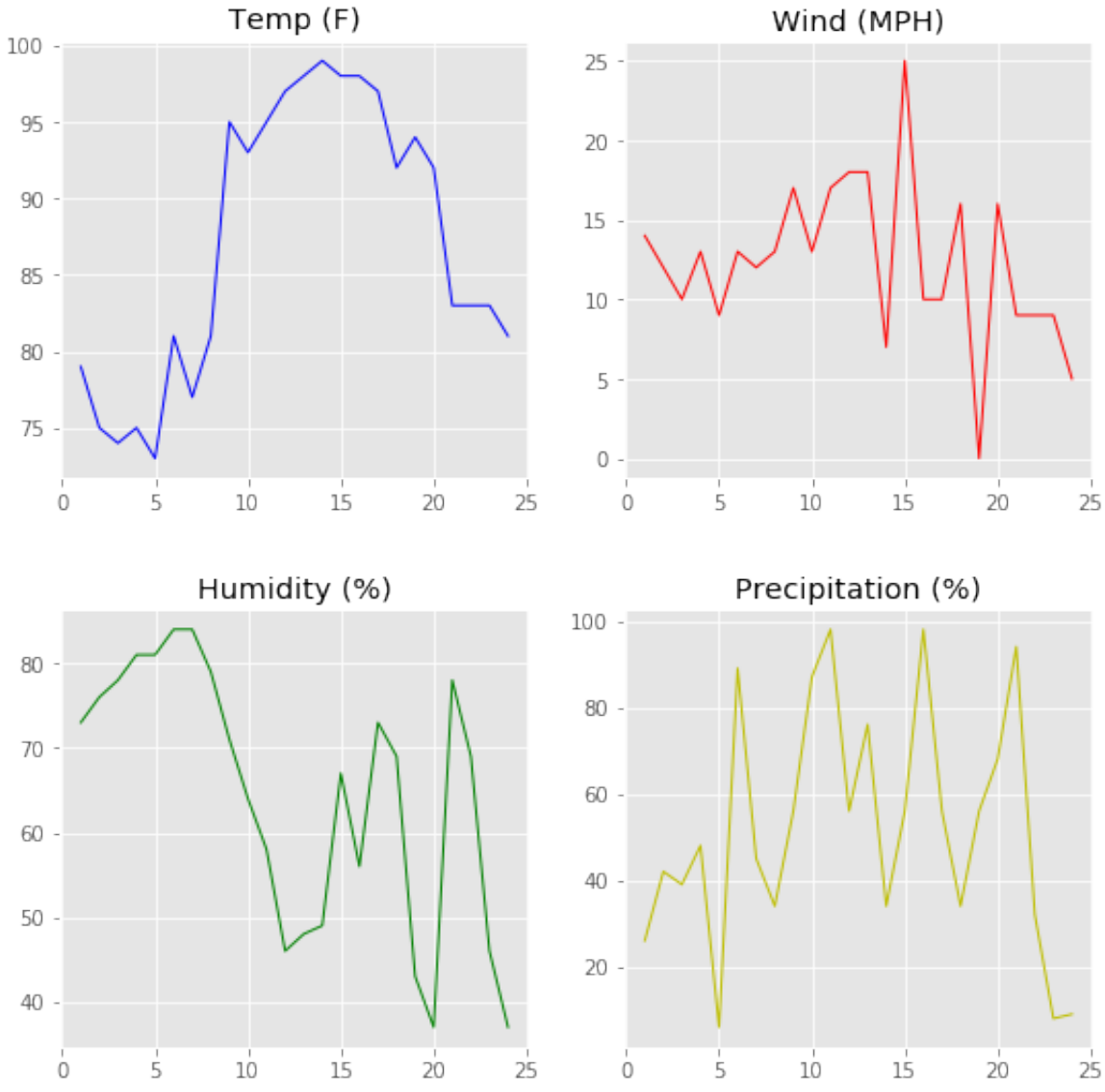
Out[24]:
```
[<matplotlib.lines.Line2D at 0x7f854ce77110>]
```



In [28]:
```
# draw subplots for (2,1,1) and (2,1,2)
plt.figure(figsize=(6,6))
plt.subplots_adjust(hspace = 0.25)
plt.subplot(2,1,1)
plt.title('Humidity')
plt.plot(time_hrs,humidity_data,color = 'r',linestyle = '-',linewid
th = 1)
plt.subplot(2,1,2)
plt.title('Precipitation')
plt.plot(time_hrs,precip_data,color = 'b',linestyle = '-',linewidth
= 1)
plt.show()
```



In [29]:
```
# draw subplots for (2,2,1),(2,2,2),(2,2,3),(2,2,4)
plt.figure(figsize=(9,9))
plt.subplots_adjust(hspace = 0.3)
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 = 'r',linestyle = '-',linewidth =
1)
plt.subplot(2,2,3)
plt.title('Humidity (%)')
plt.plot(time_hrs,humidity_data,color = 'g',linestyle = '-',linewid
th = 1)
plt.subplot(2,2,4)
plt.title('Precipitation (%)')
plt.plot(time_hrs,precip_data,color = 'y',linestyle = '-',linewidth
= 1)
plt.show()
```



In [30]:
```
# histogram and scatter plots
# import the boston dataset from sklearn library
from sklearn.datasets import load_boston
# import matplotlib
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
```

In [31]:
```
# load boston dataset
boston_data = load_boston()
```

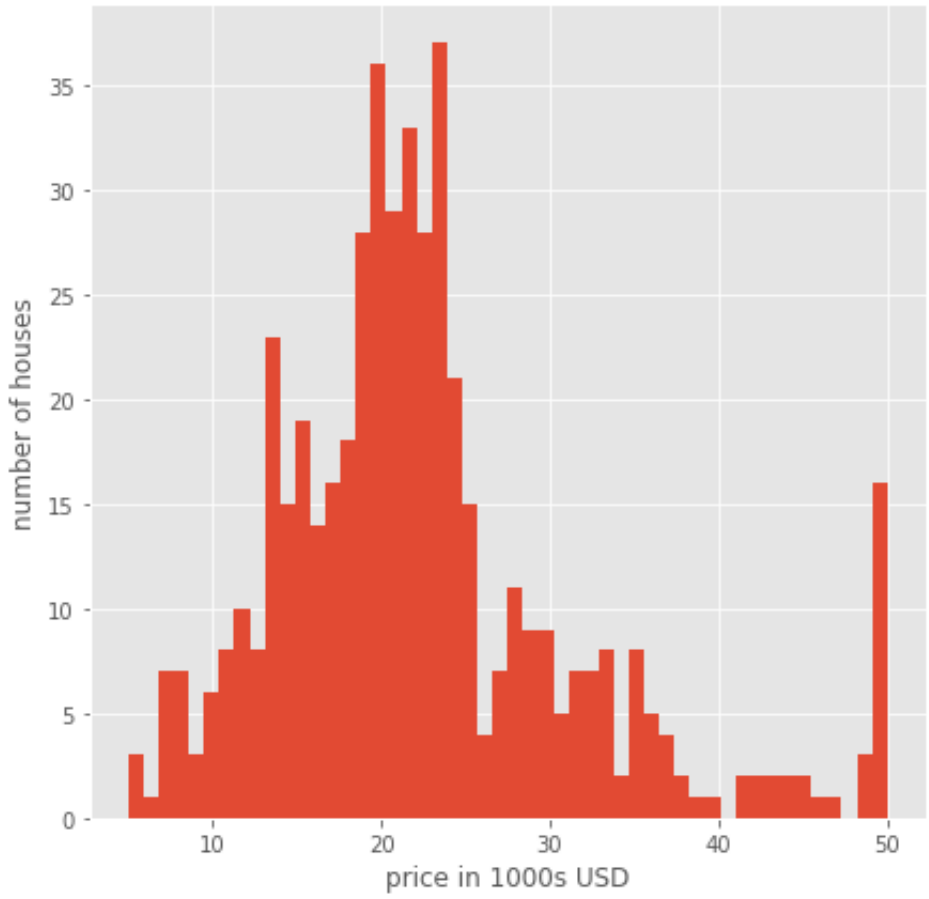
In [32]:
```
#view boston dataset
print(boston_data.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Medi
an Value (attribute 14) is usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots o
ver 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per tow
n
        - CHAS     Charles River dummy variable (= 1 if tract boun
ds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 milli
on)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior
to 1940
        - DIS      weighted distances to five Boston employment ce
ntres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of
blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.
```

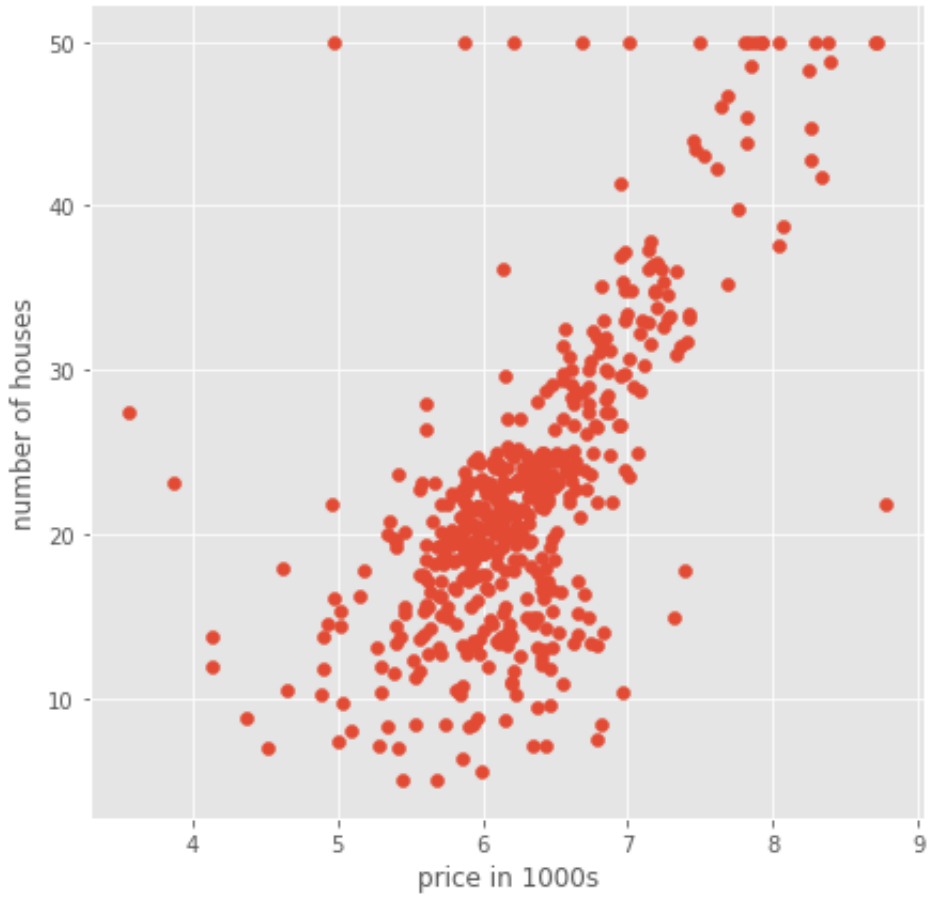
```
This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/


This dataset was taken from the StatLib library which is maintaine
d at Carnegie Mellon University.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'H
edonic
prices and the demand for clean air', J. Environ. Economics & Mana
gement,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression
diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the
table on
pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning
papers that address regression
problems.

.. topic:: References

   - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying I
nfluential Data and Sources of Collinearity', Wiley, 1980. 244-261
.
   - Quinlan,R. (1993). Combining Instance-Based and Model-Based L
earning. In Proceedings on the Tenth International Conference of M
achine Learning, 236-243, University of Massachusetts, Amherst. Mo
rgan Kaufmann.
```

In [33]:
```
# define x-axis for the data
x_axis = boston_data.data

# define y-axis for the data
y_axis = boston_data.target
```

In [34]:
```
# plot histogram
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 [35]:
```
# plot scatter plot
style.use('ggplot')
plt.figure(figsize = (7,7))
plt.scatter(boston_data.data[:,5],boston_data.target)
plt.xlabel('price in 1000s')
plt.ylabel('number of houses')
plt.show()
```



In [36]:
```
# heat map
# import matplot library
import matplotlib.pyplot as plt
# import seaborn library
import seaborn as sns
# to show plot on notebook
%matplotlib inline
```

In [37]:
```
# load flight data from the sns dataset
flight_data = sns.load_dataset('flights')
```

In [38]:
```
# view top 5 records
flight_data.head()
```

Out[38]:

	year	month	passengers
0	1949	January	112
1	1949	February	118
2	1949	March	132
3	1949	April	129
4	1949	May	121

In [39]:
```
# use pivot method to arrange the datasets
flight_data = flight_data.pivot('month','year','passengers')
```

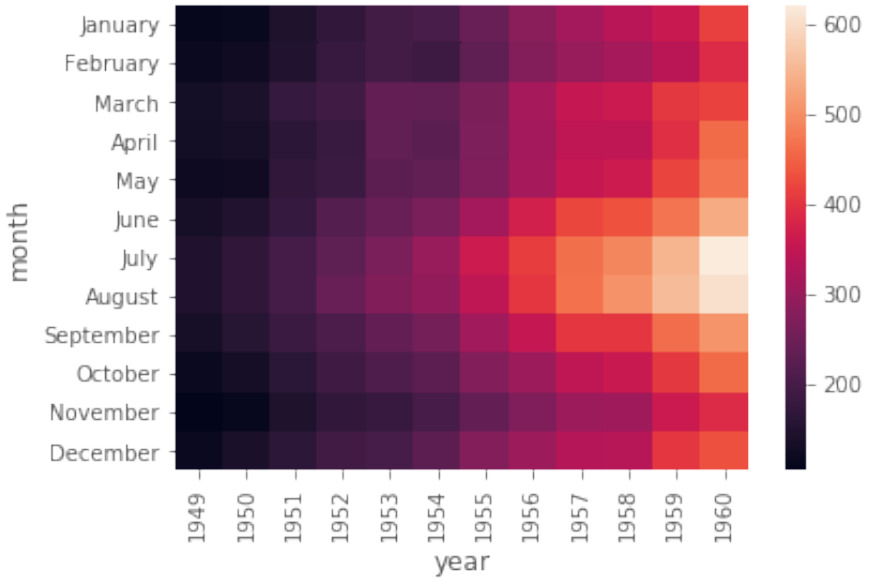
In [40]:
```
# view the datasets
flight_data
```

Out[40]:

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
June	135	149	178	218	243	264	315	374	422	435	472	535
July	148	170	199	230	264	302	364	413	465	491	548	622
August	148	170	199	242	272	293	347	405	467	505	559	606
September	136	158	184	209	237	259	312	355	404	404	463	508
October	119	133	162	191	211	229	274	306	347	359	407	461
November	104	114	146	172	180	203	237	271	305	310	362	390
December	118	140	166	194	201	229	278	306	336	337	405	432

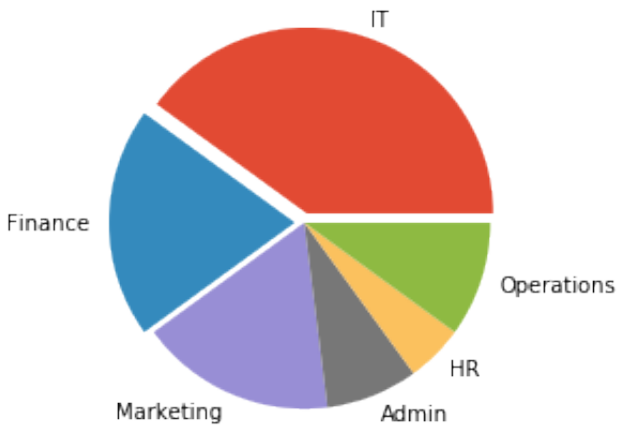
In [41]:
```
# use heatmap method o generate the heatmap of the flights data
sns.heatmap(flight_data)
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x7f853433be50>



In [42]:
```
# pie charts
# import matplot library
import matplotlib.pyplot as plt
# to show plot on notebook
%matplotlib inline
```

In [52]:
```
# job data in percentile
job_data = ['40','20','17','8','5','10']

# define label as different departments
depart = ['IT','Finance','Marketing','Admin','HR','Operations']

# explode the first slice which is IT
explode = (0.05,0.05,0,0,0,0)

# draw the piechart and set the parameters
plt.pie(job_data,labels=labels,explode=explode)

# show the plot
plt.show()
```



In [ ]: