Ch01_Introduction_ Part 1

September 5, 2018

1 Chapter 1 Getting Started with Python

```
In [47]: help?
```

```
In [4]: age,mark,code=10,75,"CIS2403"
        print (age)
        print (mark)
         print (code)


          File "<ipython-input-4-5e544bb51da0>", line 4
        print (code)
        ^
    IndentationError: unexpected indent
```

```
In [5]: TV=15
        Mobile=20
        Tablet = 30

        total = TV +
                Mobile +
                Tablet
        print (total)


          File "<ipython-input-5-68bc7095f603>", line 5
        total = TV +
                     ^
    SyntaxError: invalid syntax
```

```
In [6]: TV=15
        Mobile=20
        Tablet = 30
        total = TV + \
                Mobile + \
                Tablet
        print (total)
```

```
65
```

```
In [7]: days = ['Monday', 'Tuesday', 'Wednesday',
                'Thursday', 'Friday']
        print (days)
```

```
['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday']
```

```
In [8]: sms1 = 'Hellow World'
        sms2 = "Hellow World"
        sms3 = """ Hellow World"""
        sms4 = """ Hellow
            World"""

        print (sms1)
        print (sms2)
        print (sms3)
        print (sms4)
```

```
Hellow World
Hellow World
 Hellow World
 Hellow
    World
```

```
In [9]: TV=15; name="Nour"; print (name); print ("Welcome to\nDubai Festivale 2018")
```

```
Nour
Welcome to
Dubai Festivale 2018
```

```
In [10]: name = input("Enter your name ")
         age = int (input("Enter your age"))

         print ("\nName=", name); print ("\nAge=", age)
```

```
Enter your name Nour
Enter your age12

Name= Nour

Age= 12
```

1.0.1 1.2 Declaring Variable and Assigning Values

```
In [11]: age = 11
         name ="Nour"
         tall=100.50
```

```
In [12]: print (age)
         print (name)
         print (tall)
```

```
11
Nour
100.5
```

```
In [13]: age= mark = code =25
         print (age)
         print (mark)
         print (code)
```

```
25
25
25
```

```
In [14]: age,mark,code=10,75,"CIS2403"
         print (age)
         print (mark)
         print (code)
```

```
10
75
CIS2403
```

```
In [16]: # Expressions
         x=0.6
         x=3.9 * x * (1-x)
         print (round(x, 2))
```

```
0.94
```

```
In [18]: # Python single line comment
```

```
In [19]: ''' This
                 Is
                 Multipline comment'''
```

```
Out[19]: ' This \n        Is \n        Multipline comment'
```

```
In [20]: print ("pi=%s"%"3.14159")
```

```
pi=3.14159
```

```
In [1]: print("The value of %s is = %02f" % ("pi", 3.14159))
```

```
The value of pi is = 3.141590
```

```
In [21]: print ("Your name is %s, and your height is %.2f while your weight is %.2d" %
                ('Ossama', 172.156783, 75.56647))
```

```
Your name is Ossama, and your height is 172.16 while your weight is 75
```

```
In [23]: print ("Hi %(Name)s, your height is %(height).2f" %{'Name':"Ossama",
                                                             'height': 172.156783})
```

```
Hi Ossama, your height is 172.16
```

```
In [24]: x = "price is"
         print ("{1} {0} {2}".format(x, "The", 1920.345))
```

```
The price is 1920.345
```

```
In [34]: class A():x=9
         w=A()
         print ("{0} {1[2]} {2[test]} {3.x}".format("This", ["a", "or", "is"],
                                                    {"test": "another"},w))
         print ("{1[1]} {0} {1[2]} {2[test]} {3.x}".format("This",
                                     ["a", "or", "is"], {"test": "another"},w))
```

```
This is another 9
or This is another 9
```

```
In [42]: import time
         localtime = time.asctime(time.localtime(time.time()))
         print ("Formatted time :", localtime)
         print (time.localtime())
         print (time.time())
```

```
Formatted time : Fri Aug 17 19:12:07 2018
time.struct_time(tm_year=2018, tm_mon=8, tm_mday=17, tm_hour=19, tm_min=12, tm_sec=7, tm_wday=4,
1534533127.8304486
```

```
In [45]: import calendar
         calendar.prcal(2018)
```

```
                                  2018

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7                1  2  3  4                1  2  3  4
 8  9 10 11 12 13 14       5  6  7  8  9 10 11       5  6  7  8  9 10 11
15 16 17 18 19 20 21      12 13 14 15 16 17 18      12 13 14 15 16 17 18
22 23 24 25 26 27 28      19 20 21 22 23 24 25      19 20 21 22 23 24 25
29 30 31                  26 27 28                  26 27 28 29 30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                   1          1  2  3  4  5  6                   1  2  3
 2  3  4  5  6  7  8       7  8  9 10 11 12 13       4  5  6  7  8  9 10
 9 10 11 12 13 14 15      14 15 16 17 18 19 20      11 12 13 14 15 16 17
16 17 18 19 20 21 22      21 22 23 24 25 26 27      18 19 20 21 22 23 24
23 24 25 26 27 28 29      28 29 30 31               25 26 27 28 29 30
30

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                   1             1  2  3  4  5                      1  2
 2  3  4  5  6  7  8       6  7  8  9 10 11 12       3  4  5  6  7  8  9
 9 10 11 12 13 14 15      13 14 15 16 17 18 19      10 11 12 13 14 15 16
16 17 18 19 20 21 22      20 21 22 23 24 25 26      17 18 19 20 21 22 23
23 24 25 26 27 28 29      27 28 29 30 31            24 25 26 27 28 29 30
30 31

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7                1  2  3  4                      1  2
 8  9 10 11 12 13 14       5  6  7  8  9 10 11       3  4  5  6  7  8  9
15 16 17 18 19 20 21      12 13 14 15 16 17 18      10 11 12 13 14 15 16
22 23 24 25 26 27 28      19 20 21 22 23 24 25      17 18 19 20 21 22 23
29 30 31                  26 27 28 29 30            24 25 26 27 28 29 30
                                                    31
```

```
In [46]: ########### End
```

```
In [48]: print (13//5)
```

```
2
```

```
In [50]: print (13<5)
         print (13>5)
         print (13<=5)
         print (2>=5)
         print (13==5)
         print (13!=5)
```

```
False
True
False
False
False
True
```

```
In [56]: x=10
         print (x)
         x=10; x/=2
         print (x)
         x=10; x+=7
         print (x)
         x=10; x-=5
         print (x)
         x=10; x*=5
         print (x)
         x=13; x%=5
         print (x)
         x=10; x**=3
         print(x)
         x=10; x//=2
         print(x)
```

```
10
5.0
17
5
50
3
1000
5
```

```
In [57]: x=10>5 and 4>20
         print (x)

         x=10>5 or 4>20
         print (x)
         x=not(10<4)
         print (x)
```

```
False
True
True
```

```
In [45]: print (13/5)
```

```
2.6
```

```
In [46]: print (13%5)
```

```
3
```

```
In [47]: print (2**3)
```

```
8
```

```
In [7]:
```

```
Out[7]: ' This \n        Is \n        Multipline comment'
```

```
In [10]:     #single line comment

             '''This is
             multiline comment'''
```

```
Out[10]: 'This is \nmultiline comment'
```

```
In [5]: # Expressions
        x=0.6
        x=3.9 *x *(1-x)
        print (round( x,2) )
```

```
0.94
```

```
In [10]: largest = None
         print ('Before:', largest)
         for val in [30, 45, 12, 90, 74, 15]:
             if largest is None or val > largest :
                 largest = val
                 print ('Loop:', val, largest)
         print ('Largest:', largest)
```

```
Before: None
Loop: 30 30
Loop: 45 45
Loop: 90 90
Largest: 90
```

2 Pandas and other libraries

```
In [34]: #Create series from array using pandas and numpy
         import pandas as pd
         import numpy as np
         data = np.array([90,75,50,66])
         s = pd.Series(data,index=['A','B','C','D'])
         print (s)
```

```
A    90
B    75
C    50
D    66
dtype: int64
```

```
In [36]: print (s[1])
```

```
75
```

```
In [37]: #Create series from dictionary using pandas and numpy
         import pandas as pd
         import numpy as np
         data = {'Ahmed' : 92, 'Ali' : 55, 'Omar' : 83}
         s = pd.Series(data,index=['Ali','Ahmed','Omar'])
         print (s)
```

```
Ali      55
Ahmed    92
Omar     83
dtype: int64
```

```
In [38]: print (s[1:])
```

```
Ahmed    92
Omar     83
dtype: int64
```

3 DataFrame

```
In [39]: import pandas as pd
         data = [['Ahmed',35],['Ali',17],['Omar',25]]
         DataFrame1 = pd.DataFrame(data,columns=['Name','Age'])
         print (DataFrame1)
```

```
    Name  Age
0  Ahmed   35
1    Ali   17
2   Omar   25
```

```
In [40]: DataFrame1[1:]
```

```
Out[40]:    Name  Age
         1   Ali   17
         2  Omar   25
```

```
In [41]: import pandas as pd
         data = {'Name':['Ahmed', 'Ali', 'Omar', 'Salwa'],'Age':[35,17,25,30]}
         dataframe2 = pd.DataFrame(data, index=[100, 101, 102, 103])
         print (dataframe2)
```

```
     Age   Name
100   35  Ahmed
101   17    Ali
102   25   Omar
103   30  Salwa
```

```
In [42]: dataframe2[:2]
```

```
Out[42]:      Age   Name
         100   35  Ahmed
         101   17    Ali
```

```
In [43]: dataframe2['Name']
```

```
Out[43]: 100    Ahmed
         101      Ali
         102     Omar
         103    Salwa
         Name: Name, dtype: object
```

4 Panel

```
In [44]: # creating a panel
         import pandas as pd
         import numpy as np
         data = {'Temprature Day1' : pd.DataFrame(np.random.randn(4, 3)),
                 'Temprature Day2' : pd.DataFrame(np.random.randn(4, 2))}
         p = pd.Panel(data)
         print (p['Temprature Day1'])
```

```
          0         1         2
0  1.152400 -1.298529  1.440522
1 -1.404988 -0.105308 -0.192273
2 -0.575023 -0.424549  0.146086
3 -1.347784  1.153291 -0.131740
```

5 1.6.3 PYTHON LAMBDAS, AND THE NUMPY LIBRARY.

```
In [46]: result = lambda x, y : x * y
         result(2,5)
```

```
Out[46]: 10
```

```
In [47]: result(4,10)
```

```
Out[47]: 40
```

```
In [65]: def fahrenheit(T):
             return ((float(9)/5)*T + 32)
         def celsius(T):
             return (float(5)/9)*(T-32)
         Temp = (15.8, 25, 30.5,25)

         F = list ( map(fahrenheit, Temp))
         C = list ( map(celsius, F))
         print (F)
         print (C)
```

```
[60.44, 77.0, 86.9, 77.0]
[15.799999999999999, 25.0, 30.500000000000004, 25.0]
```

```
In [72]: Celsius = [39.2, 36.5, 37.3, 37.8]
         Fahrenheit = map(lambda x: (float(9)/5)*x + 32, Celsius)
         for x in Fahrenheit:
             print(x)
```

```
102.56
97.7
99.14
100.03999999999999
```

```
In [79]: fib = [0,1,1,2,3,5,8,13,21,34,55]
         result = filter(lambda x: x % 2==0, fib)
         for x in result:
             print(x)
```

```
0
2
8
34
```

```
In [81]: f = lambda a,b: a if (a > b) else b
         reduce(f, [47,11,42,102,13])
```

```
        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-81-08f32714e194> in <module>()
          1 f = lambda a,b: a if (a > b) else b
    ----> 2 reduce(f, [47,11,42,102,13])


        NameError: name 'reduce' is not defined
```

```
In [82]: reduce(lambda x,y: x+y, [47,11,42,13])
```

```
        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-82-f82ab4f906ad> in <module>()
    ----> 1 reduce(lambda x,y: x+y, [47,11,42,13])


        NameError: name 'reduce' is not defined
```

```
In [83]: a=np.array([[1,2,3],[4,5,6]])
         b=np.array([[7,8,9],[10,11,12]])
         np.add(a,b)

Out[83]: array([[ 8, 10, 12],
                [14, 16, 18]])

In [84]: np.subtract(a,b) #Same as a-b

Out[84]: array([[-6, -6, -6],
                [-6, -6, -6]])
```

6 Series

```
In [6]: import pandas as pd
        animals = ["Lion", "Tiger", "Bear"]
        pd.Series(animals)
```

```
Out[6]: 0     Lion
        1    Tiger
        2     Bear
        dtype: object
```

```
In [5]: marks = [95, 84, 55, 75]
        pd.Series(marks)
```

```
Out[5]: 0    95
        1    84
        2    55
        3    75
        dtype: int64
```

```
In [11]: # Create series from dictionary where indices are the dictionary keys
         quiz1 = {"Ahmed":75, "Omar": 84, "Salwa": 70}
         q = pd.Series(quiz1)
         q
```

```
Out[11]: Ahmed    75
         Omar     84
         Salwa    70
         dtype: int64
```

```
In [13]: # query series
         q.loc['Ahmed']
```

```
Out[13]: 75
```

```
In [20]: q['Ahmed']
```

```
Out[20]: 75
```

```
In [19]: q.iloc[2]
```

```
Out[19]: 70
```

```
In [21]: q[2]
```

```
Out[21]: 70
```

```
In [25]: # implement numpy operation on a series
         s = pd.Series([70,90,65,25, 99])
         s
```

```
Out[25]: 0    70
         1    90
         2    65
         3    25
         4    99
         dtype: int64
```

```
In [27]: total =0
         for val in s:
             total += val
         print (total)
```

```
349
```

```
In [28]: import numpy as np
         total = np.sum(s)
         print (total)
```

```
349
```

```
In [29]: # add new values to series
         s = pd.Series ([99,55,66,88])
         s.loc['Ahmed'] = 85
         s
```

```
Out[29]: 0        99
         1        55
         2        66
         3        88
         Ahmed    85
         dtype: int64
```

```
In [32]: # Append Series
         test = [95, 84, 55, 75]
         marks = pd.Series(test)
         s = pd.Series ([99,55,66,88])
         s.loc['Ahmed'] = 85
         s
         NewSeries = s.append(marks)
         NewSeries
```

```
Out[32]: 0        99
         1        55
         2        66
         3        88
         Ahmed    85
         0        95
```

```
1        84
2        55
3        75
dtype: int64
```

7 1.6.6 RUN BASIC INFERENTIAL STATISTICAL ANALYSES.

```
In [37]: import numpy as np
         x = np.random.binomial(20, .5, 10000)
         print((x>=15).mean())
```

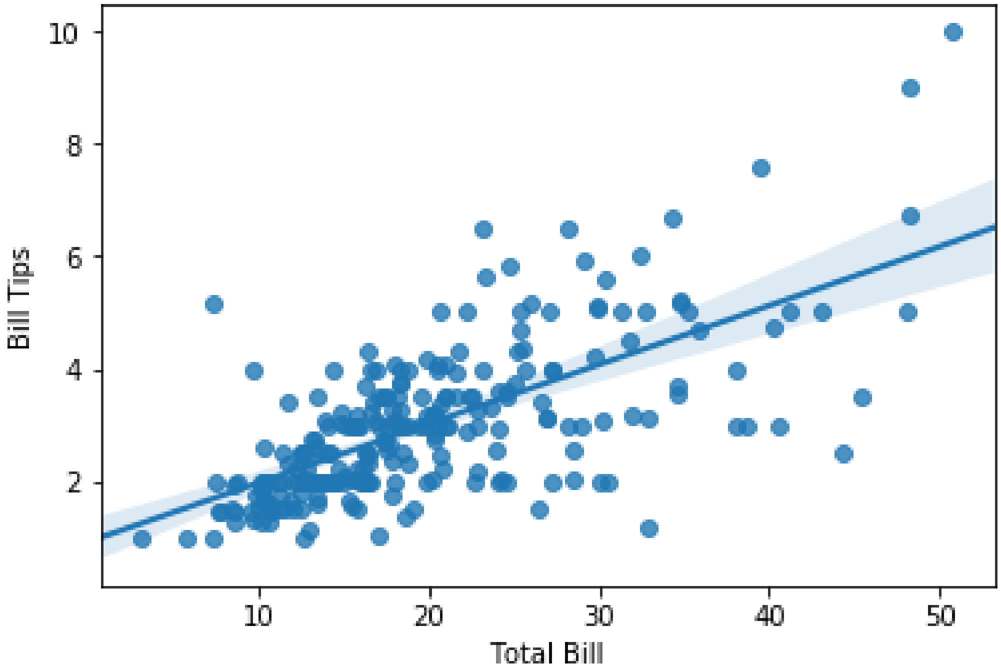
```
0.022
```

```
In [ ]: sb.regplot(x = "Total Bill", y = "Bill's Tips", data = df)
```

8 Regression

```
In [65]: import seaborn as sb
         from matplotlib import pyplot as plt
         df = sb.load_dataset('tips')
         sb.regplot(x = "total_bill", y = "tip", data = df)
         plt.xlabel('Total Bill')
         plt.ylabel('Bill Tips')

         plt.show()
```

```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



```
In [39]: df

Out[39]:      total_bill   tip     sex smoker  day    time  size
         0         16.99  1.01  Female     No  Sun  Dinner     2
         1         10.34  1.66    Male     No  Sun  Dinner     3
         2         21.01  3.50    Male     No  Sun  Dinner     3
         3         23.68  3.31    Male     No  Sun  Dinner     2
         4         24.59  3.61  Female     No  Sun  Dinner     4
         5         25.29  4.71    Male     No  Sun  Dinner     4
         6          8.77  2.00    Male     No  Sun  Dinner     2
         7         26.88  3.12    Male     No  Sun  Dinner     4
         8         15.04  1.96    Male     No  Sun  Dinner     2
         9         14.78  3.23    Male     No  Sun  Dinner     2
         10        10.27  1.71    Male     No  Sun  Dinner     2
         11        35.26  5.00  Female     No  Sun  Dinner     4
         12        15.42  1.57    Male     No  Sun  Dinner     2
         13        18.43  3.00    Male     No  Sun  Dinner     4
         14        14.83  3.02  Female     No  Sun  Dinner     2
         15        21.58  3.92    Male     No  Sun  Dinner     2
         16        10.33  1.67  Female     No  Sun  Dinner     3
         17        16.29  3.71    Male     No  Sun  Dinner     3
         18        16.97  3.50  Female     No  Sun  Dinner     3
         19        20.65  3.35    Male     No  Sat  Dinner     3
         20        17.92  4.08    Male     No  Sat  Dinner     2
         21        20.29  2.75  Female     No  Sat  Dinner     2
```

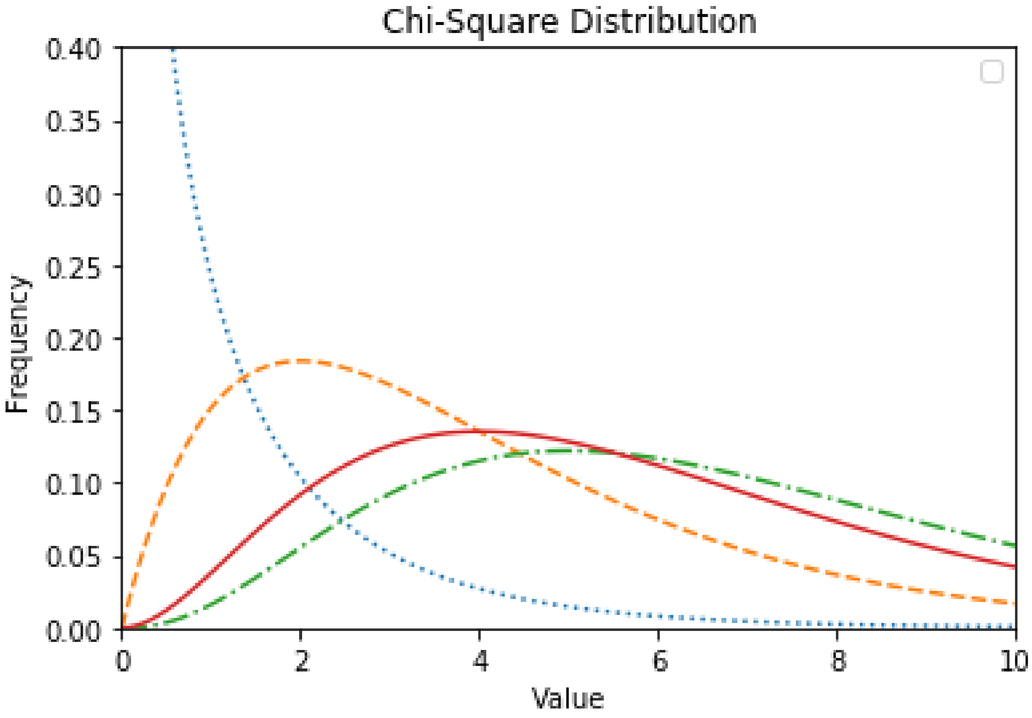
```
22        15.77  2.23  Female     No   Sat  Dinner     2
23        39.42  7.58    Male     No   Sat  Dinner     4
24        19.82  3.18    Male     No   Sat  Dinner     2
25        17.81  2.34    Male     No   Sat  Dinner     4
26        13.37  2.00    Male     No   Sat  Dinner     2
27        12.69  2.00    Male     No   Sat  Dinner     2
28        21.70  4.30    Male     No   Sat  Dinner     2
29        19.65  3.00  Female     No   Sat  Dinner     2
..          ...   ...     ...    ...   ...     ...   ...
214       28.17  6.50  Female    Yes   Sat  Dinner     3
215       12.90  1.10  Female    Yes   Sat  Dinner     2
216       28.15  3.00    Male    Yes   Sat  Dinner     5
217       11.59  1.50    Male    Yes   Sat  Dinner     2
218        7.74  1.44    Male    Yes   Sat  Dinner     2
219       30.14  3.09  Female    Yes   Sat  Dinner     4
220       12.16  2.20    Male    Yes   Fri   Lunch     2
221       13.42  3.48  Female    Yes   Fri   Lunch     2
222        8.58  1.92    Male    Yes   Fri   Lunch     1
223       15.98  3.00  Female     No   Fri   Lunch     3
224       13.42  1.58    Male    Yes   Fri   Lunch     2
225       16.27  2.50  Female    Yes   Fri   Lunch     2
226       10.09  2.00  Female    Yes   Fri   Lunch     2
227       20.45  3.00    Male     No   Sat  Dinner     4
228       13.28  2.72    Male     No   Sat  Dinner     2
229       22.12  2.88  Female    Yes   Sat  Dinner     2
230       24.01  2.00    Male    Yes   Sat  Dinner     4
231       15.69  3.00    Male    Yes   Sat  Dinner     3
232       11.61  3.39    Male     No   Sat  Dinner     2
233       10.77  1.47    Male     No   Sat  Dinner     2
234       15.53  3.00    Male    Yes   Sat  Dinner     2
235       10.07  1.25    Male     No   Sat  Dinner     2
236       12.60  1.00    Male    Yes   Sat  Dinner     2
237       32.83  1.17    Male    Yes   Sat  Dinner     2
238       35.83  4.67  Female     No   Sat  Dinner     3
239       29.03  5.92    Male     No   Sat  Dinner     3
240       27.18  2.00  Female    Yes   Sat  Dinner     2
241       22.67  2.00    Male    Yes   Sat  Dinner     2
242       17.82  1.75    Male     No   Sat  Dinner     2
243       18.78  3.00  Female     No  Thur  Dinner     2

[244 rows x 7 columns]
```

9 Python - Chi-Square Test

```
In [41]: from scipy import stats
         import numpy as np
         import matplotlib.pyplot as plt
```

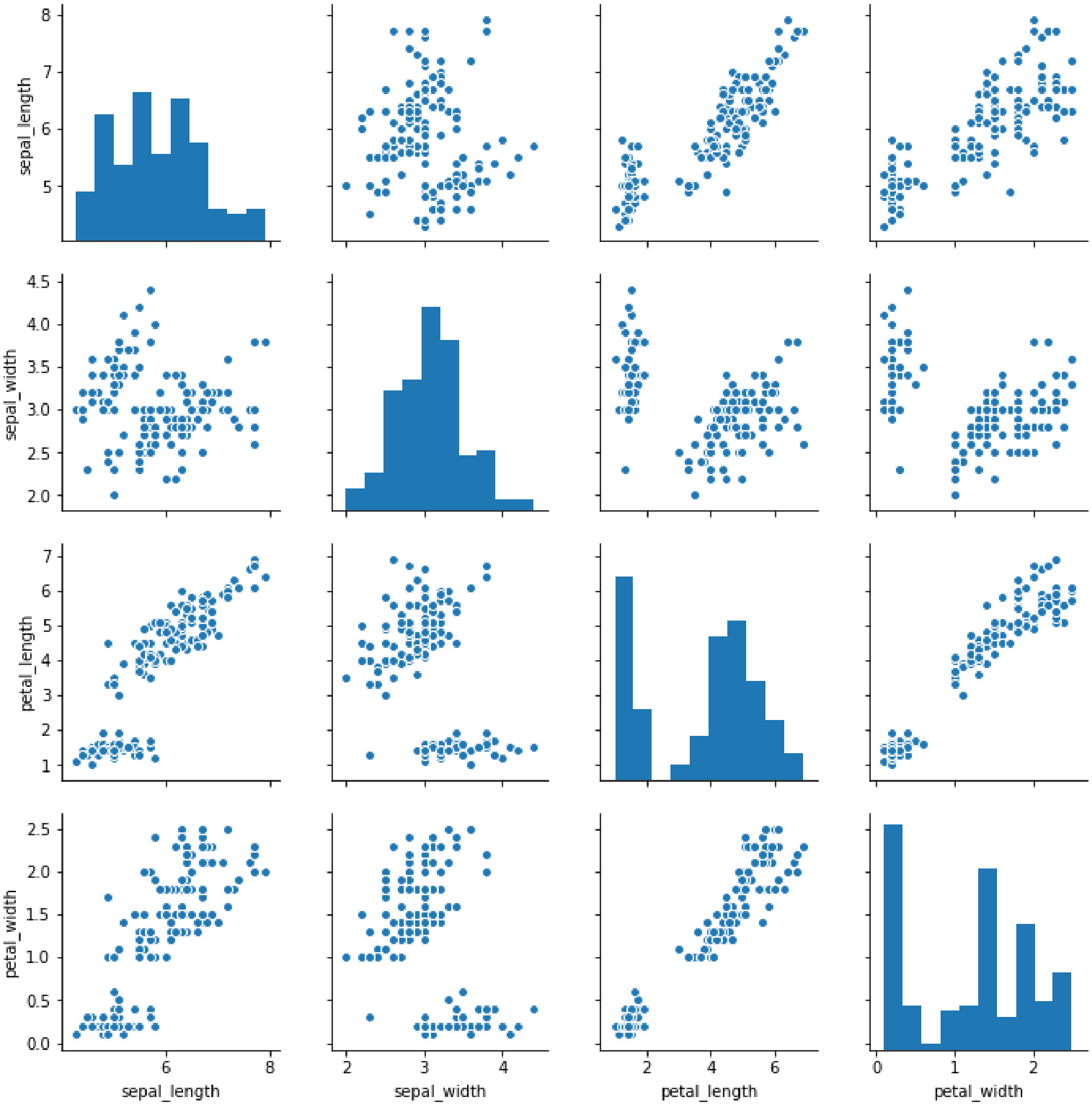
```
x = np.linspace(0, 10, 100)
fig,ax = plt.subplots(1,1)

linestyles = [':', '--', '-.', '-']
deg_of_freedom = [1, 4, 7, 6]
for df, ls in zip(deg_of_freedom, linestyles):
  ax.plot(x, stats.chi2.pdf(x, df), linestyle=ls)

plt.xlim(0, 10)
plt.ylim(0, 0.4)

plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Chi-Square Distribution')

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

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



10 correlation

```
In [42]: import matplotlib.pyplot as plt
         import seaborn as sns
         df = sns.load_dataset('iris')


         #without regression
         sns.pairplot(df, kind="scatter")
         plt.show()
```

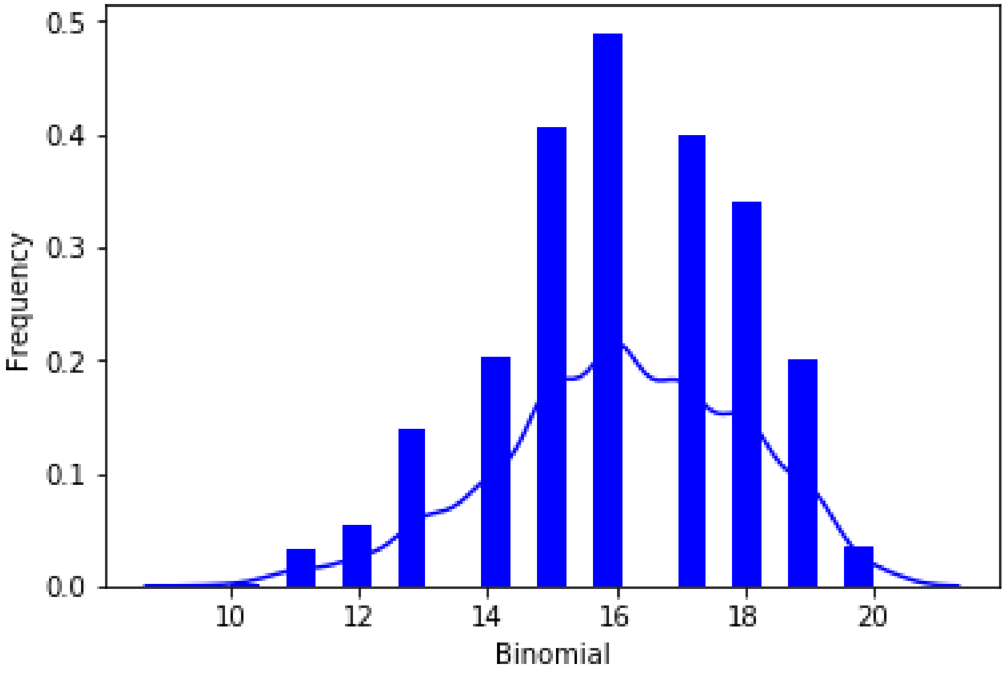


```
In [46]: from scipy.stats import binom
         import seaborn as sb
```

```
        data_binom = binom.rvs(n=20,p=0.8,loc=0,size=1000)
        ax = sb.distplot(data_binom,
                          kde=True,
                          color='blue',
                          hist_kws={"linewidth": 25,'alpha':1})
        ax.set(xlabel='Binomial', ylabel='Frequency')
```

```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

```
Out[46]: [Text(0,0.5,'Frequency'), Text(0.5,0,'Binomial')]
```



```
In [58]: import pandas as pd

        d = {'Name':pd.Series(['Ahmed','Omar','Ali','Salwa','Majid','Othman','Gameel',
           'Ziad','Ahlam','Zahrah','Ayman','Alaa']),
           'Age':pd.Series([34,26,25,27,30,54,23,43,40,30,28,46]),
           'Height':pd.Series([114.23,173.24,153.98,172.0,153.20,164.6,183.8,163.78,172.0,164.8

        #Create a DataFrame
        df = pd.DataFrame(d)

        # Calculate the standard deviation
        print (df.std())
```

```
Age        9.740574
Height    18.552823
dtype: float64
```

```
In [59]: print (df.describe())
```

```
             Age      Height
count  12.000000   12.000000
mean   33.833333  164.448333
std     9.740574   18.552823
min    23.000000  114.230000
25%    26.750000  161.330000
50%    30.000000  168.400000
75%    40.750000  173.455000
max    54.000000  183.800000
```

```
In [60]: print ("Mean Values in the Distribution")
         print (df.mean())
         print ("*******************************")
         print ("Median Values in the Distribution")
         print (df.median())
         print ("*******************************")
         print ("Mode Values in the Distribution")
         print (df['Height'].mode())
```

```
Mean Values in the Distribution
Age        33.833333
Height    164.448333
dtype: float64
*******************************
Median Values in the Distribution
Age        30.0
Height    168.4
dtype: float64
*******************************
Mode Values in the Distribution
0    172.0
dtype: float64
```

10.0.1 1.5 EXERCISES

```
In [2]: # Store input numbers:
        num1 = input('Enter first number: ')
        num2 = input('Enter second number: ')

        sumval = float(num1) + float(num2)  # Add two numbers
        minval = float(num1) - float(num2) # Subtract two numbers
        mulval = float(num1) * float(num2) # Multiply two numbers
        divval = float(num1) / float(num2)  #Divide two numbers

        # Display the sum
        print('The sum of {0} and {1} is {2}'.format(num1, num2, sumval))
        # Display the subtraction
        print('The subtraction of {0} and {1} is {2}'.format(num1, num2, minval))
        # Display the multiplication
        print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mulval))
        # Display the division
        print('The division of {0} and {1} is {2}'.format(num1, num2, divval))
```

```
Enter first number: 10
Enter second number: 5
The sum of 10 and 5 is 15.0
The subtraction of 10 and 5 is 5.0
The multiplication of 10 and 5 is 50.0
The division of 10 and 5 is 2.0
```

```
In [3]: # A. write a python script to prompt the user to enter the triangle first side (a),
        #second side (b) and third side (c) lengths. Then calculate the semi-perimeter (s).
        #calculate the triangle area and display the result to the user.
        #Area of a triangle = (s*(s-a)*(s-b)*(s-c))-1/2.
        a = float(input('Enter first side: '))
        b = float(input('Enter second side: '))
        c = float(input('Enter third side: '))
        s = (a + b + c) / 2  # calculate the semi-perimeter

        # calculate the area
        area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
        print('The area of the triangle is %0.2f' %area)
```

```
Enter first side: 10
Enter second side: 9
Enter third side: 7
The area of the triangle is 30.59
```

```
In [7]: import random
        a = int(input('Enter the starting value : '))
        b = int(input('Enter the end value : '))
        print(random.randint(a,b))
        random.sample(range(a, b), 3)
```

```
Enter the starting value : 10
Enter the end value : 100
14
```

```
Out[7]: [64, 12, 41]
```

```
In [9]: # convert kilometers to miles
        kilometers = float(input('Enter the distance in kilometers: '))
        # conversion factor
        Miles = kilometers * 0.62137
        print('%0.2f kilometers is equal to %0.2f miles' %(kilometers,Miles))
```

```
Enter the distance in kilometers: 120
120.00 kilometers is equal to 74.56 miles
```

```
In [11]: # convert convert Celsius to Fahrenheit
         Celsius = float(input('Enter temperature in Celsius: '))
         # conversion factor
         Fahrenheit = (Celsius * 1.8) + 32
         print('%0.2f Celsius is equal to %0.2f Fahrenheit' %(Celsius,Fahrenheit))
```

```
Enter temperature in Celsius: 25
25.00 Celsius is equal to 77.00 Fahrenheit
```

10.1 End Chapter 1