Practice sheet 1, Solution

Exercise 1:

Complete the following Python Code

[1]:
```
import numpy as np
```

[3]:
```
x = np.linspace(2,15,20)
```

[4]:
```
x
```

[4]:
```
array([ 2.        ,  2.68421053,  3.36842105,  4.05263158,  4.73684211,
        5.42105263,  6.10526316,  6.78947368,  7.47368421,  8.15789474,
        8.84210526,  9.52631579, 10.21052632, 10.89473684, 11.57894737,
       12.26315789, 12.94736842, 13.63157895, 14.31578947, 15.        ])
```

[24]:
```
(15-2)/19
```

[24]:
```
0.6842105263157895
```

[13]:
```
x[3]
```

[13]:
```
4.052631578947368
```

[14]:
```
x[0]
```

[14]:
```
2.0
```

[15]:
```
len(x)
```

[15]:
```
20
```

[16]:
```
x.min()
```

[16]:
```
2.0
```

[18]:
```
import pandas as pd
```

[21]:
```
x.mean()
```

[21]:
```
8.5
```

[22]:
```
x.sum()
```

[22]:
```
170.0
```

Exercise 2

Let's consider the Python code below

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

[25]:
```
df=pd.read_csv("employee.csv")
```

[60]:
```
df.head(3)
```

[60]:

	name	age	income	gender	department	grade	performance_score
0	Allen Smith	45.0	45000	M	Operations	G3	723
1	S Kumar	NaN	16000	F	Finance	G0	520
2	Jack Morgan	32.0	35000	M	Finance	G2	674

1. What's the output of the following Python code

[27]:
```
df.shape
```

[27]:
```
(9, 7)
```

2. What was the Python command used to obtain the following output

[29]:
```
df['gender'].value_counts()
```

[29]:
```
F    5
M    4
Name: gender, dtype: int64
```

3. Complete the following output

[61]:
```
df['age']
```

[61]:
```
0    45.0
1     NaN
2    32.0
3    45.0
4    30.0
5     NaN
6    54.0
7    54.0
8    23.0
Name: age, dtype: float64
```

[31]:
```
df['age'].mean()
```

[31]:
```
40.42857142857143
```

[32]:
```
df['age'].sum()
```

[32]:
```
283.0
```

[33]:
```
df['age'].sum()/df['age'].mean()
```

[33]:
```
7.0
```

4. Deduce from below the Sample variance and the standard deviation of the `Performance_score` variable

[37]:
```
xbar=df['performance_score'].mean()
```

[39]:
```
xs=(df['performance_score']-xbar)**2
```

[40]:
```
xs.sum()
```

[40]:
```
444329.99999999994
```

[41]:
```
df['performance_score'].var()
```

[41]:
```
55541.249999999999
```

[47]:
```
xs.sum()/8
```

[47]:
```
55541.24999999999
```

[44]:
```
xs
```

[44]:
```
0     12618.777778
1      8220.444444
2      4011.111111
3      2988.444444
4     10066.777778
5      1469.444444
6    310992.111111
7     84293.444444
8      9669.444444
Name: performance_score, dtype: float64
```

[45]:
```
df['performance_score'].mean()
```

[45]:
```
610.6666666666666
```

[46]:
```
3**2
```

[46]:
```
9
```

[48]:
```
df['performance_score'].std()
```

[48]:
```
235.6719117756717
```

[50]:
```
import math
```

[54]:
```
import numpy as np
```

[57]:
```
xv=df['performance_score'].var()
```

[58]:
```
math.sqrt(xv)
```

[58]:
```
235.6719117756717
```

[59]:
```
len(df['performance_score'])
```

[59]:
```
9
```

5. Find five syntax errors in the code below

[62]:
```
import pandas as pd
```

[63]:
```
df=pd.read_csv("employee.csv")
```

[65]:
```
df.columns
```

[65]:
```
Index(['name', 'age', 'income', 'gender', 'department', 'grade',
       'performance_score'],
      dtype='object')
```

[66]:
```
df['department'].value_counts(normalize=True)
```

[66]:
```
Operations    0.333333
Finance       0.333333
Sales         0.333333
Name: department, dtype: float64
```

[67]:
```
df['department']
```

[67]:
```
0    Operations
1       Finance
2       Finance
3         Sales
4    Operations
5         Sales
6    Operations
7       Finance
```

```
8         Sales
Name: department, dtype: object
```

[70]:
```
df['department'][0]
```

[70]:
```
'Operations'
```

[71]:
```
df['department'][6:8]
```

[71]:
```
6    Operations
7       Finance
Name: department, dtype: object
```

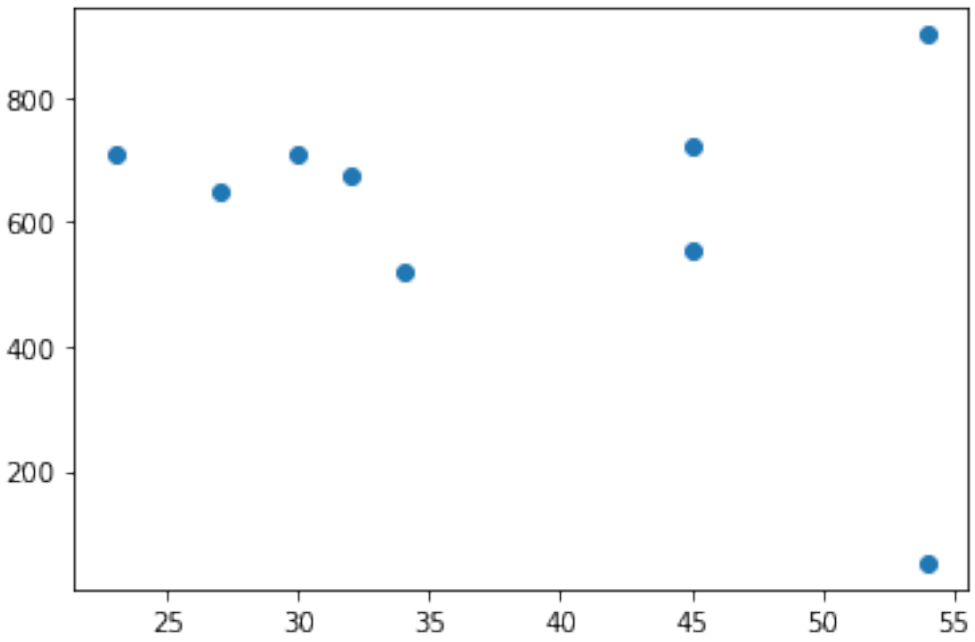
[72]:
```
df['age'].quantile(.25)
```

[72]:
```
30.0
```

[73]:
```
import matplotlib.pyplot as plt
```

[74]:
```
plt.scatter(df['age'],df['performance_score'])
```

[74]:
```
<matplotlib.collections.PathCollection at 0x1ff3494ebb0>
```



Exercise 3

Can you link with an arrow the following scatter plots with the corresponding correlation coefficients

[75]:
```
import numpy as np
```

[82]:
```
x=-3*(2*np.random.random(100)-1)
```

[83]:
```
x
```

[83]:
```
array([-1.81421499, -2.21191416,  1.4774869 , -0.55938045,  2.29464507,
       -1.07672411, -0.49438444, -0.95279691, -1.95298684,  0.88985444,
       -0.75095117, -0.66557009,  0.42643551, -2.16552336,  0.5569421 ,
       -1.97260725, -1.34978351,  2.15049004, -2.34905359,  2.81524067,
       -2.93004142,  0.08208941,  2.08717736,  2.26157565,  1.28143342,
        1.58510606, -2.61585878,  0.66489624, -0.35459923,  1.8835664 ,
       -0.28445194,  1.86667924,  0.08476038, -1.26920465,  1.23211334,
        1.49081758, -0.74002163, -1.26784486, -0.36258574,  0.25697457,
        0.3067825 , -0.35615271,  0.10808201,  0.35783274,  2.34658569,
       -0.0069983 ,  1.41477895,  1.57530582, -0.35499749, -1.03725389,
       -1.48873016,  2.27586908, -1.41099484, -1.79397638, -0.44525382,
       -2.50245834, -1.88485483, -0.99515098, -1.19731725,  1.31535663,
        2.13584236, -0.78051334, -2.54428435,  0.82448144, -0.52115386,
       -2.36290448, -1.08817422,  2.05340519,  0.35919822, -0.19049286,
       -2.59578569,  1.92833441, -0.9580864 ,  2.78173889, -2.88843812,
       -1.06885237,  1.08794641, -0.95352954,  2.24898304, -2.84769935,
       -0.8324937 ,  0.62996308, -0.94772691, -1.88168573, -1.46840959,
       -0.82657132, -1.27414459,  1.8194727 , -0.93118542, -1.21482579,
       -0.88779038, -0.05034034, -1.52932889,  0.59217538,  0.31160294,
        2.03313989, -0.12426516, -1.82268405, -2.49656053,  0.95166856])
```

[108]:
```
y1=2+.2*x+np.random.random(100)
```

[109]:
```
dd=pd.DataFrame({'x':x,'y':y1})
dd.corr()
```

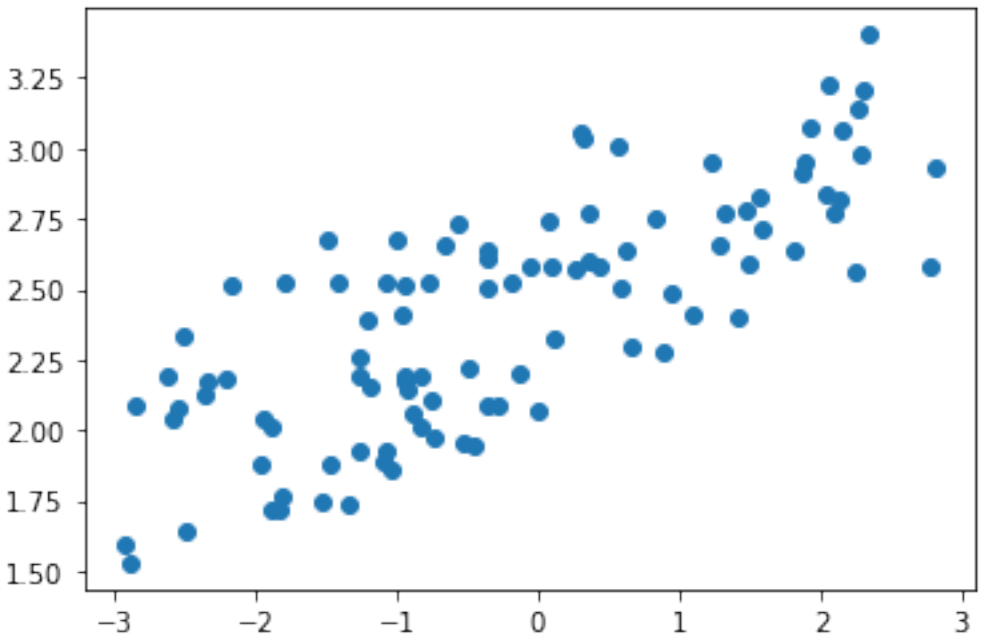
[109]:

	x	y
x	1.000000	0.764247
y	0.764247	1.000000

[110]:
```
import matplotlib.pyplot as plt
```

[111]:
```
plt.scatter(x,y1)
```

[111]:
```
<matplotlib.collections.PathCollection at 0x1ff36be3880>
```



[112]:
```
y2=-3+.85*x+np.random.random(100)
```

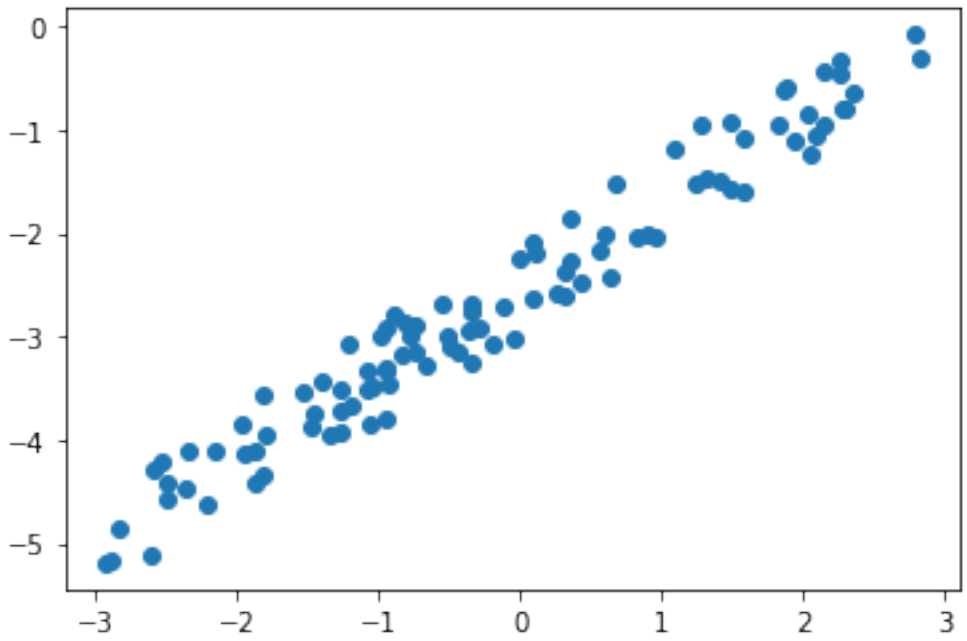
[113]:
```
dd=pd.DataFrame({'x':x,'y':y2})
dd.corr()
```

[113]:

	x	y
x	1.000000	0.977914
y	0.977914	1.000000

[114]:
```
plt.scatter(x,y2)
```

[114]:
```
<matplotlib.collections.PathCollection at 0x1ff36c5c160>
```



[115]:
```
y3=-3+.05*x+np.random.random(100)
```

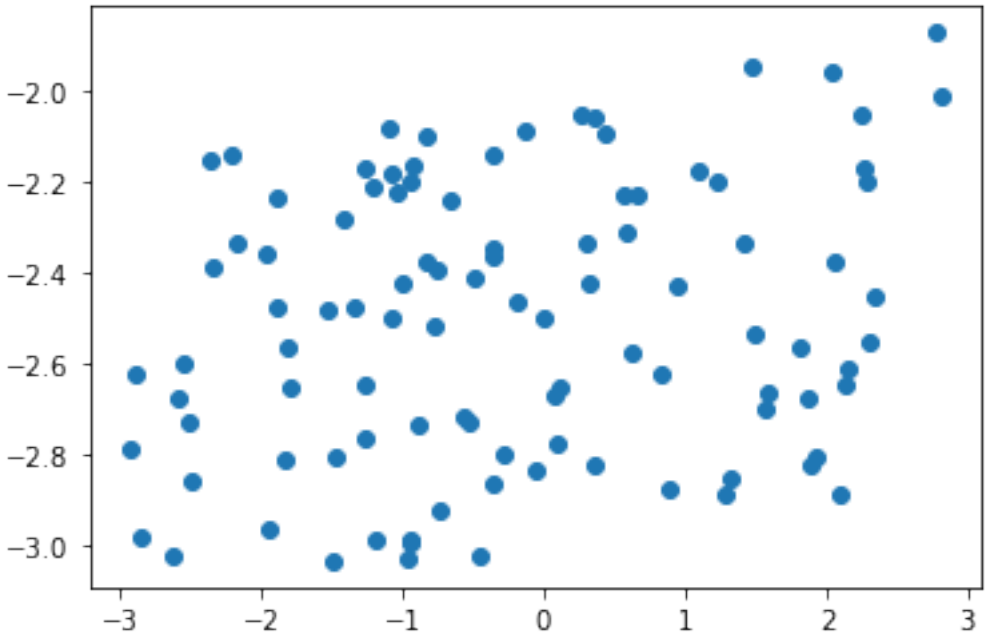
[118]:
```
dd=pd.DataFrame({'x':x,'y':y3})
dd.corr()
```

[118]:

	x	y
x	1.000000	0.224915
y	0.224915	1.000000

[119]:
```
plt.scatter(x,y3)
```

[119]:
```
<matplotlib.collections.PathCollection at 0x1ff36cb6fa0>
```



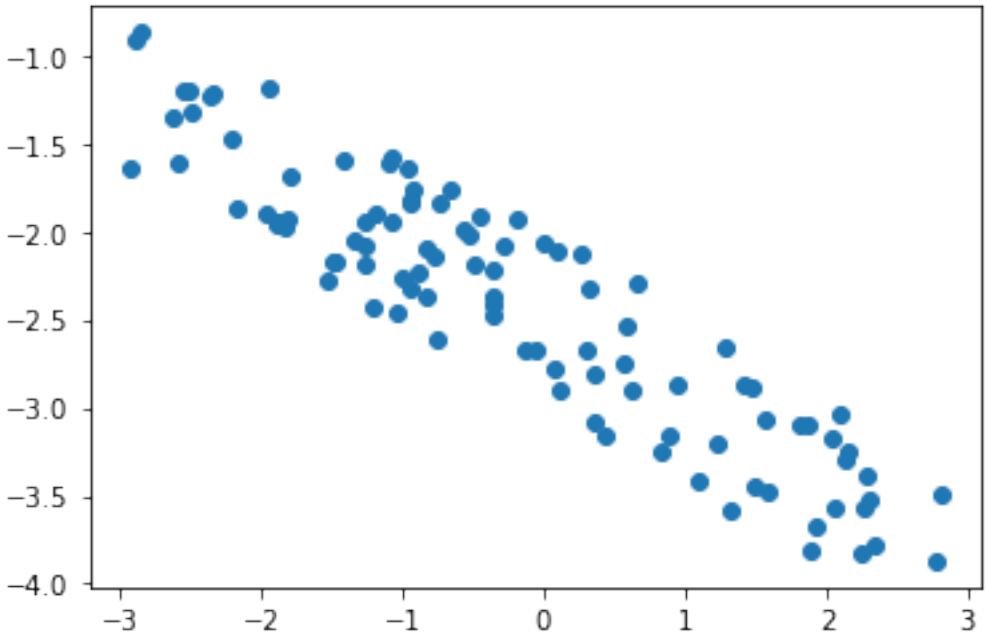
[92]:
```
y4=-3-.45*x+np.random.random(100)
```

[120]:
```
dd=pd.DataFrame({'x':x,'y':y4})
dd.corr()
```

[120]:
```
          x         y
x  1.000000 -0.919487
y -0.919487  1.000000
```

[121]:
```
plt.scatter(x,y4)
```

[121]:
```
<matplotlib.collections.PathCollection at 0x1ff36d24c70>
```



Exercise 4

Complete the following Python code:

[122]:
```
import numpy as np
```

[123]:
```
t = np.random.random((2,2))
print(t)
```

```
[[0.95936864 0.75702103]
 [0.70212368 0.98789156]]
```

[130]:
```
a = np.array([[5,6],[7,8]])
print(a)
```

```
[[5 6]
 [7 8]]
```

[131]:
```
a[0,1]
```

[131]: 6

[133]:
```
a = np.array([[[5],[6]],[[7],[8]]])
print(a)
```

```
[[[5]
  [6]]
```

```
[[7]
 [8]]]
```

[134]:
```
a[0]
```

[134]:
```
array([[5],
       [6]])
```

[135]:
```
a[0,1]
```

[135]:
```
array([6])
```

[136]:
```
a = np.arange(15,23)
print(a)
```

```
[15 16 17 18 19 20 21 22]
```

[137]:
```
a = np.arange(8)
print(a)
```

```
[0 1 2 3 4 5 6 7]
```

[139]:
```
a.reshape(4,2)
```

[139]:
```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7]])
```

[140]:
```
a.reshape(2,4)
```

[140]:
```
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

[141]:
```
b=a.reshape(4,2)
b.transpose()
```

[141]:
```
array([[0, 2, 4, 6],
       [1, 3, 5, 7]])
```

[142]:
```
b.flatten()
```

[142]:
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

[143]:
```
a = np.array([[1, 0, -1],
              [0, 3, 2],
              [-1, 0, 2]])
b=np.array([[2, 3],
            [-1, -1],
```

```
            [3, 3]])
c=np.matmul(a,b)
```

[144]:
```
c
```

[144]:
```
array([[-1,  0],
       [ 3,  3],
       [ 4,  3]])
```

0.0.1 Exercise 5

Let's consider the following Python code

[14]:
```
from scipy import stats
```

[15]:
```
X= stats.poisson(3.2)
```

[16]:
```
X.pmf(0)
```

[16]:
```
0.04076220397836621
```

[17]:
```
import numpy as np
```

[18]:
```
x=np.arange(0,10)
x
```

[18]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

[19]:
```
X.pmf(x)
```

[19]:
```
array([0.0407622 , 0.13043905, 0.20870248, 0.22261598, 0.17809279,
       0.11397938, 0.060789  , 0.02778926, 0.0111157 , 0.00395225])
```

[20]:
```
X.cdf(5)
```

[20]:
```
0.8945918945308227
```

[21]:
```
X.mean()
```

[21]:
```
3.2
```

[22]:
```
X.var()
```

[22]:
```
3.2
```

Let X be a random variable with Poisson distribution with parameter $\lambda = 3.2$. Answer to the following questions using the previous code.

1. Find the Probability that $\{X = 0\}$, $\mathbb{P}(X = 0)$.

[23]:
```
X.pmf(0)
```

[23]:
```
0.04076220397836621
```

2. Compute the probability of $\{2 \leq X < 6\}$, $\mathbb{P}(2 \leq X < 6)$

[24]:
```
X.pmf(x[0:4]).sum()
```

[24]:
```
0.602519724405557
```

3. Compute the probability of $\{X \geq 7\}$, $\mathbb{P}(X \geq 7)$

[25]:
```
1-X.pmf(x[0:5]).sum()
```

[25]:
```
0.21938748893269588
```

4. What's the probability of $\{X = -3\}$, $\mathbb{P}(X = -3)$?

[26]:
```
X.pmf(-3)
```

[26]:
```
0.0
```

5. Compute the following probability: $\mathbb{P}(-2 < X \leq 3)$.

[27]:
```
X.pmf(x[0:2]).sum()
```

[27]:
```
0.1712012567091381
```

Exercice 6

Complete the following code

[159]:
```
from scipy import stats
```

[160]:
```
X= stats.bernoulli(.4)
```

[161]:
```
X.pmf(-1)
```

[161]:
```
0.0
```

[162]:
```
X.pmf(0)
```

[162]:
```
0.6
```

[163]:
```
X.pmf(3)
```

[163]:
```
0.0
```

[164]:
```
X.pmf(1)
```

[164]: 0.4

[165]:
```
X.mean()
```

[165]: 0.4

[166]:
```
X.std()
```

[166]: 0.4898979485566356

[167]:
```
X.var()
```

[167]: 0.24

[168]:
```
X.cdf(-1)
```

[168]: 0.0

[169]:
```
X.cdf(1)
```

[169]: 1.0

[170]:
```
X.cdf(.3)
```

[170]: 0.6

[171]:
```
X.cdf(10.3)
```

[171]: 1.0

[173]:
```
X.pmf(1)-X.pmf(0)
```

[173]: -0.19999999999999996

[ ]: