pt1

23 октября 2020 г.

1 PRACTICAL TASK #1

1.1 PART #1

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

[2]:
```
X = [69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74, 75, 75,␣
 ↪76]
Y = [53, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140,␣
 ↪150, 165, 185, 210, 220]
```

[3]:
```
X = np.array(X)
Y = np.array(Y)
```

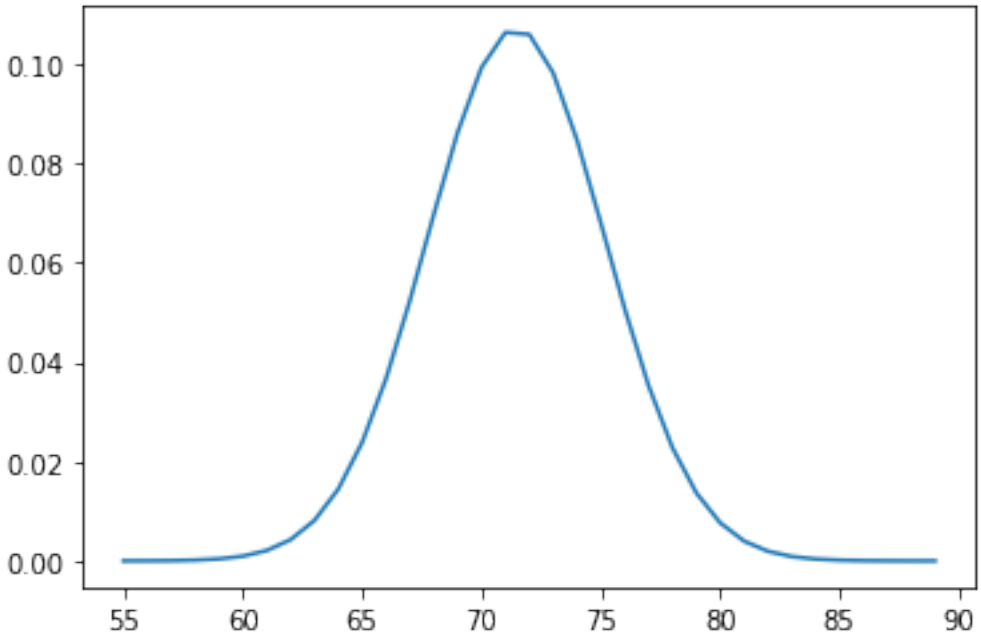
[4]:
```
# Найти среднее, медиану и моду величины X
np.mean(X), np.median(X), *stats.mode(X)[0]
```

[4]:
```
(71.45, 71.5, 74)
```

[5]:
```
# Найти дисперсию Y
np.var(Y)
```

[5]:
```
1961.2100000000003
```

[6]:
```
# Построить график нормального распределения для X
x_axis = np.arange(55, 90)
plt.plot(x_axis, stats.norm.pdf(x_axis, np.mean(X), np.std(X)))
plt.show()
```



[7]:
```
# Найти вероятность того, что возраст больше 80
1 - stats.norm(np.mean(X), np.std(X)).cdf(80)
```

[7]:
```
0.010791377919371459
```

[8]:
```
# Найти двумерное мат. ожидания и ковариационную матрицу для этих двух величин
np.mean([X, Y], axis=1)
```

[8]:
```
array([ 71.45, 159.7 ])
```

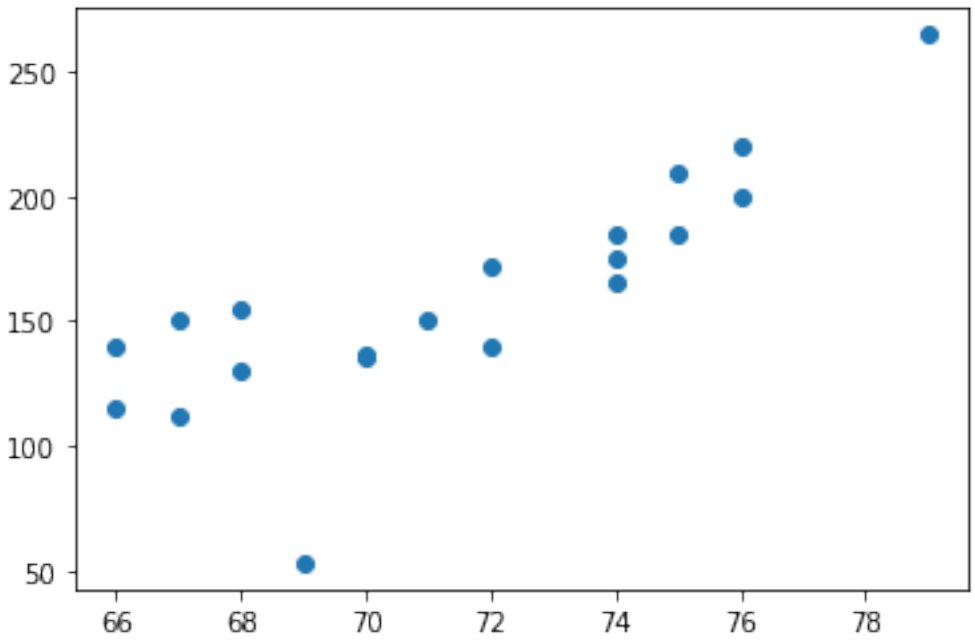
[9]:
```
# Найти ковариационную матрицу для этих двух величин
np.cov(X, Y)
```

[9]:
```
array([[  14.57631579,  141.77368421],
       [ 141.77368421, 2064.43157895]])
```

[10]:
```
# Определять корреляцию между X и Y
np.corrcoef(X, Y)[0, 1]
```

[10]:
```
0.8172811723193554
```

[11]:
```
# Построить диаграмму рассеяния, отображающая зависимость между возрастом и весом
plt.scatter(X, Y)
plt.show()
```



1.2 PART #2

[12]:
```
M = [[17, 17, 12],
     [11, 9, 13],
     [11, 8, 19]]
```

[13]:
```
# Рассчитайте ковариационную матрицу
np.cov(M)
```

[13]:
```
array([[  8.33333333,  -5.        , -15.83333333],
       [ -5.        ,   4.        ,  11.        ],
       [-15.83333333,  11.        ,  32.33333333]])
```

[14]:
```
# и обобщенную дисперсию
np.linalg.det(np.cov(M)) # определитель ковариационной матрицы - обобщенная␣
 ↪дисперсия
```

[14]:
```
2.2204460492503156e-14
```

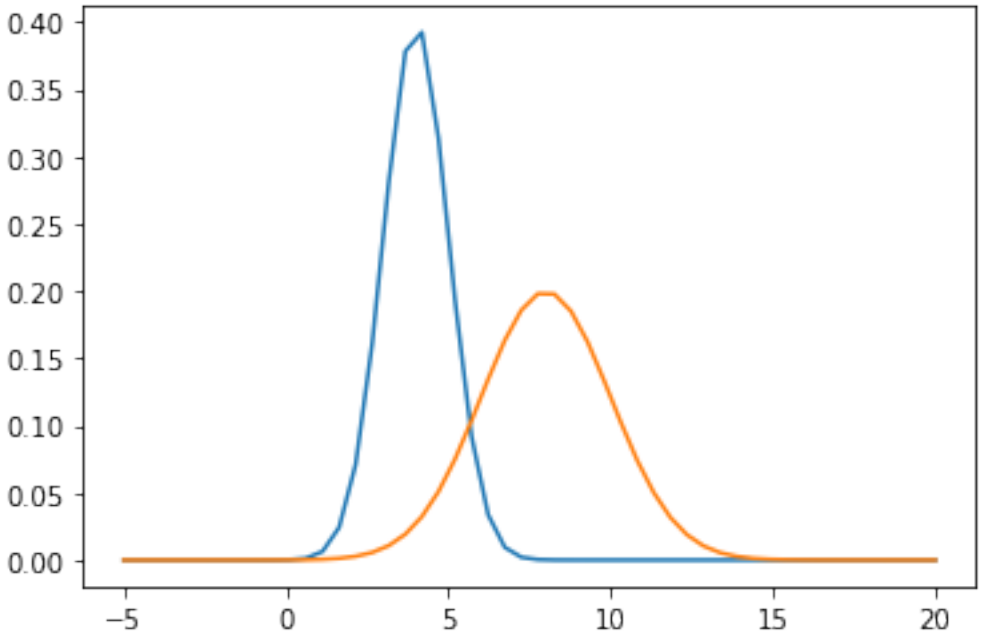
1.3 PART #3

[15]:
```
Na = stats.norm(4, 1)
Nb = stats.norm(8, 2)
```

[16]:
```
# Для каждого из значения {5,6,7} определите какое из распределений
 ↪сгенерировало значение с большей вероятностью.
["Na" if Na.pdf(n) > Nb.pdf(n) else "Nb" for n in [5, 6, 7] ]
```

[16]:
```
['Na', 'Nb', 'Nb']
```

[17]:
```
# Найди значение, которой могло быть сгенерировано обеими распределениями с
 ↪равной вероятностью
ls = np.linspace(-5, 20)
plt.plot(ls, Na.pdf(ls))
plt.plot(ls, Nb.pdf(ls))
plt.show()
```



[18]:
```
ls = np.linspace(5, 6, 100)
[el for el in ls if (abs(Na.pdf(el) - Nb.pdf(el)) < 0.001)]
```

[18]:
```
[5.656565656565657]
```