



GeekBrains

Теория вероятностей и математическая статистика

Вебинары



GeekBrains

Урок 6

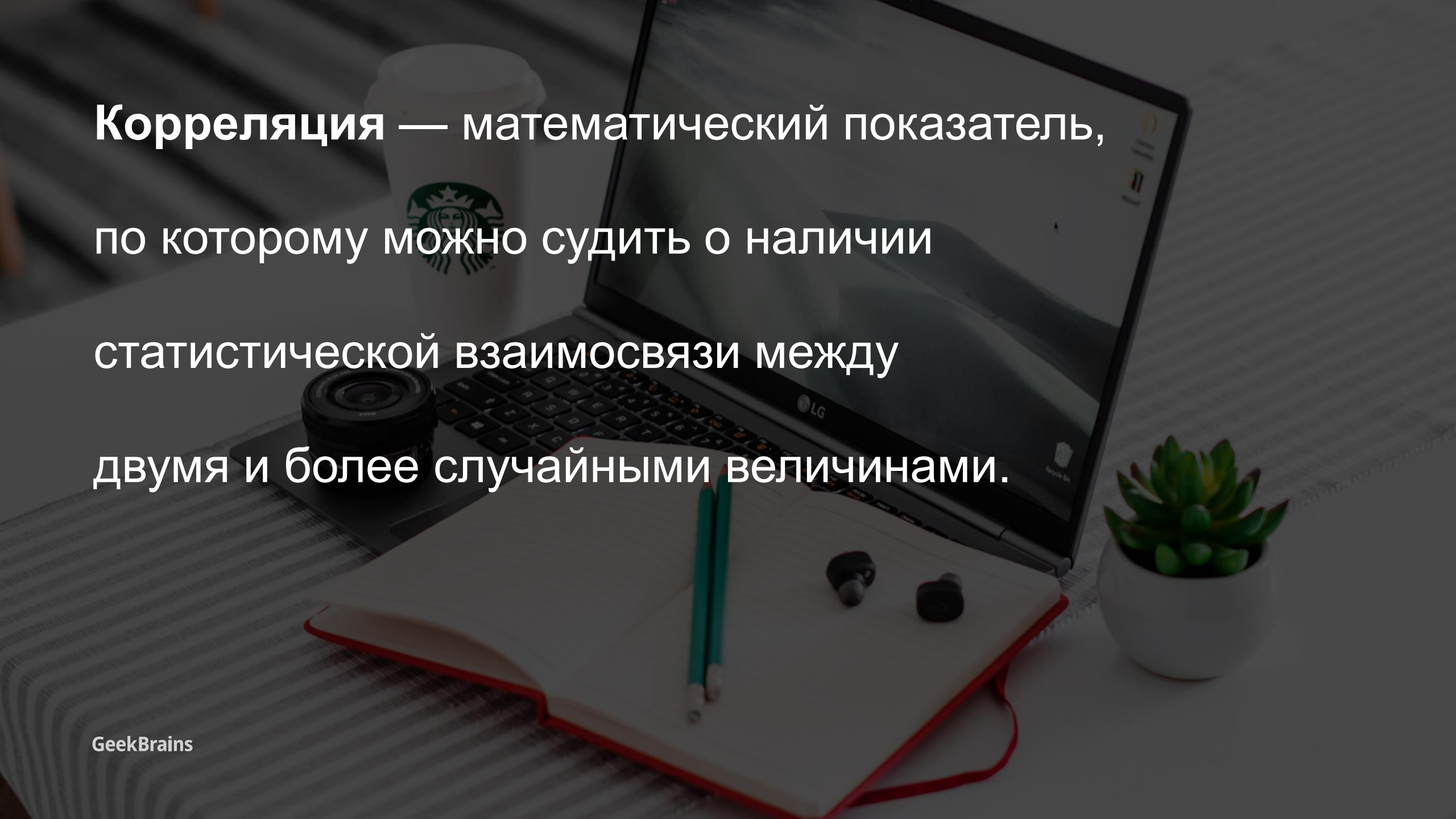
Теория вероятности и математическая статистика

Взаимосвязь величин. Параметрические и непараметрические показатели корреляции.

Корреляционный анализ

На этом уроке мы изучим:

1. Что такое корреляция
2. Коэффициент корреляции
3. Взаимосвязь величин
4. Ковариация
5. Ограничения корреляционного анализа

A desk setup featuring a laptop, a Starbucks cup, a camera lens, a notebook, and a small potted plant. The text is overlaid on the image.

Корреляция — математический показатель,
по которому можно судить о наличии
статистической взаимосвязи между
двумя и более случайными величинами.

Коэффициент корреляции - это коэффициент, показывающий, на сколько велика линейная взаимосвязь между величинами.

Площадь	Цена
27	1.2
37	1.6
42	1.8
48	1.8
54	2.5
56	2.6
77	3
80	3.3

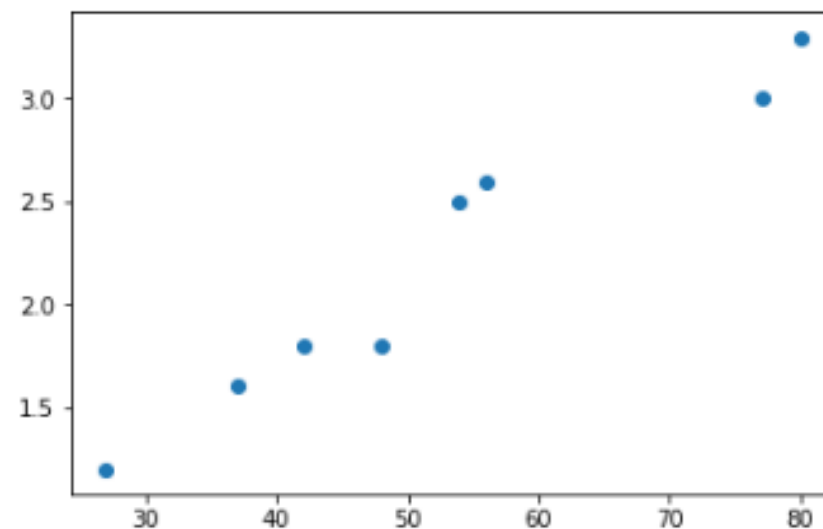
```
In [8]: 1 import numpy as np
```

```
In [11]: 1 s=np.array([27,37,42,48,54,56,77,80])  
2 s
```

```
Out[11]: array([27, 37, 42, 48, 54, 56, 77, 80])
```

```
In [12]: 1 p=([1.2,1.6,1.8,1.8,2.5,2.6,3,3.3])
```

```
: 1  
2 import matplotlib.pyplot as plt  
3 plt.scatter (X,y)  
4 plt.show()
```



```
In [8]: 1 import numpy as np
```

```
In [11]: 1 s=np.array([27,37,42,48,54,56,77,80])  
2 s
```

```
Out[11]: array([27, 37, 42, 48, 54, 56, 77, 80])
```

```
In [18]: 1 p=([1.2,1.6,1.8,1.8,2.5,2.6,3,3.3])  
2 p
```

```
Out[18]: [1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3, 3.3]
```

```
In [19]: 1 np.corrcoef(p,s)
```

```
Out[19]: array([[1.          , 0.97318772],  
                [0.97318772, 1.          ]])
```

```
In [2]: import numpy as np
```

```
In [3]: a=np.array([1,2,3,4,5])
```

```
In [4]: b=np.array([7,4,6,9,0])
```

```
In [8]: np.corrcoef(a,b)
```

```
Out[8]: array([[ 1.          , -0.41602515],  
               [-0.41602515,  1.          ]])
```

```
In [9]: a=np.array([1,2,3,4,5])
```

```
In [10]: b=np.array([11,12,0.8,9,0.4])
```

```
In [11]: np.corrcoef(a,b)
```

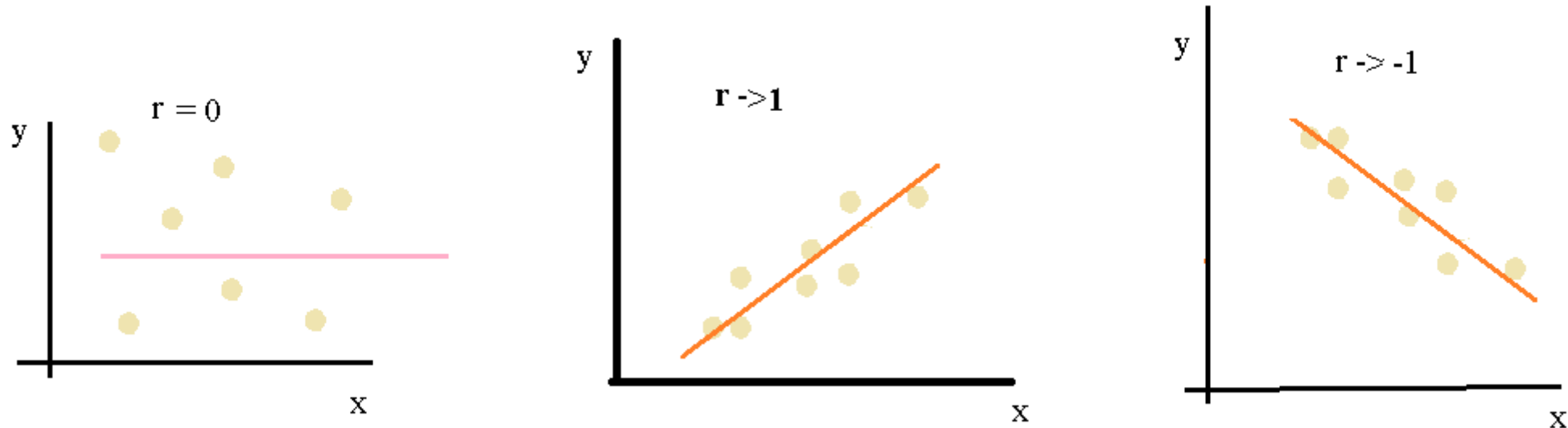
```
Out[11]: array([[ 1.          , -0.68080746],  
               [-0.68080746,  1.          ]])
```

```
In [12]: b=np.array([0.5,0.7,0.9,0.8,1])
```

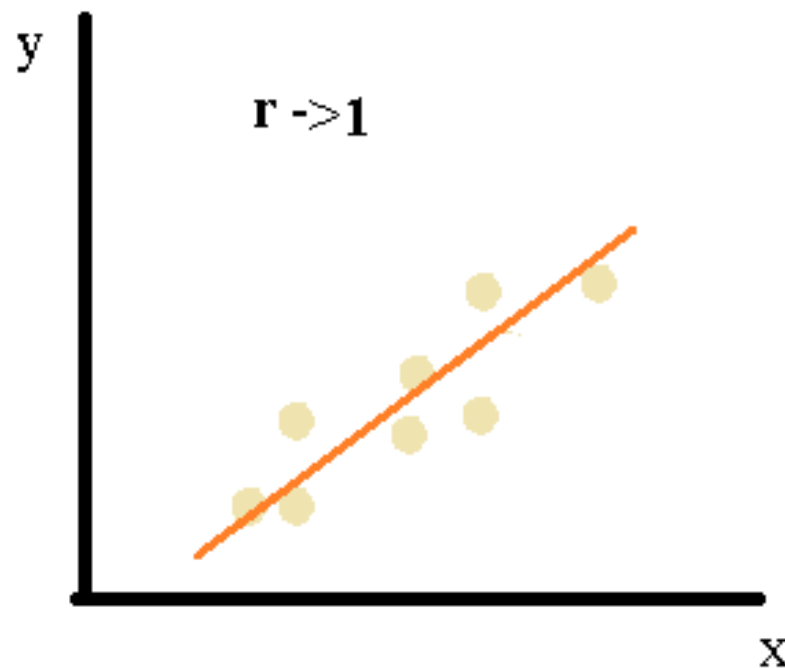
```
In [14]: np.corrcoef(a,b)
```

```
Out[14]: array([[1.          ,  0.90419443],  
               [0.90419443,  1.          ]])
```

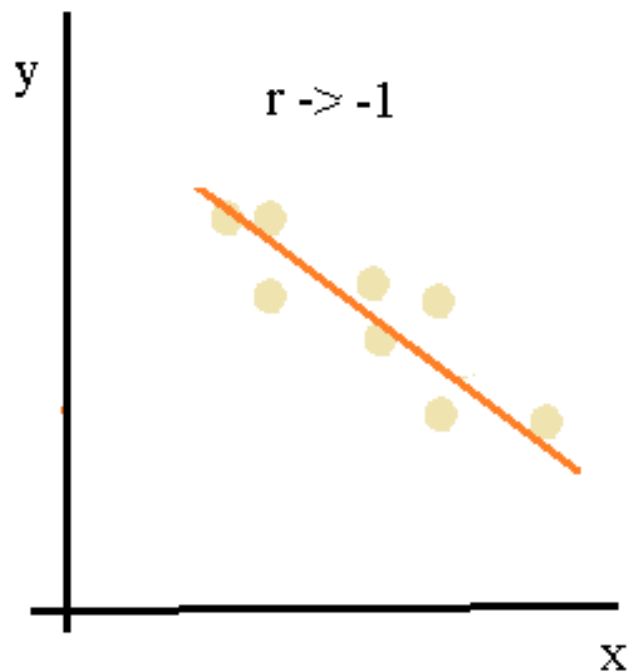
Коэффициент корреляции обозначается символами R или r и может принимать значения от -1 до 1 включительно



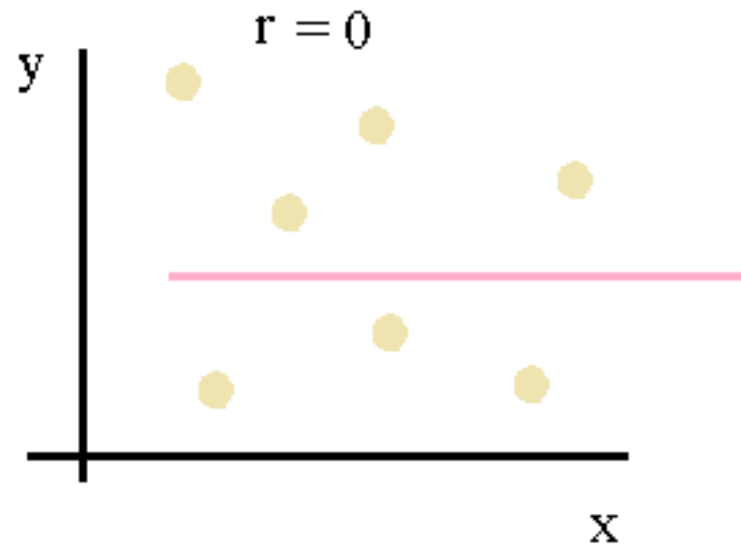
Если **коэффициент корреляции** близок к 1, то между величинами наблюдается прямая связь: увеличение одной величины сопровождается увеличением другой, а уменьшение одной величины сопровождается уменьшением другой.



Если же **коэффициент корреляции** близок к -1 , то между величинами есть обратная корреляционная связь: увеличение одной величины сопровождается уменьшением другой и наоборот.



Коэффициент корреляции, равный 0, говорит о том, что между величинами нет **линейной связи.**



Отсутствие **корреляции** между двумя величинами еще не говорит о том, что между показателями нет связи.

```
In [2]: import numpy as np
```

```
In [4]: s = np.array([0, -1, 1, -2, 2, -3, 3, -4, 4])
s
```

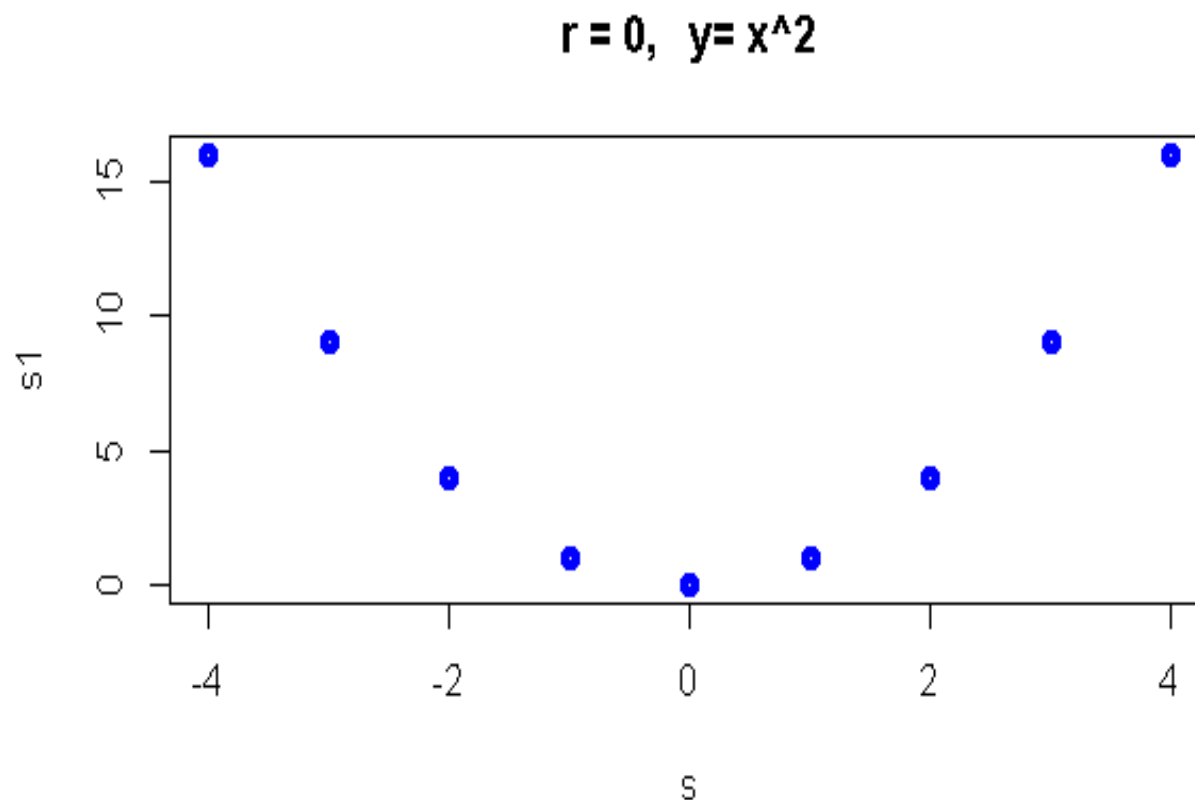
```
Out[4]: array([ 0, -1,  1, -2,  2, -3,  3, -4,  4])
```

```
In [5]: s1 = np.array([0, 1, 1, 4, 4, 9, 9, 16, 16])
s1
```

```
Out[5]: array([ 0,  1,  1,  4,  4,  9,  9, 16, 16])
```

```
In [6]: np.corrcoef(s, s1)
```

```
Out[6]: array([[1., 0.],
               [0., 1.]])
```



Высокая **корреляция** двух величин может свидетельствовать о том, что у них есть **общая причина**

Наличие корреляции еще не значит, что величины взаимосвязаны, но может подразумевать некую скрытую причину, 3-ю переменную

Пример : чем больше театров, тем больше больниц . Прямая корреляция. На самом деле взаимосвязи нет.

Третья скрытая переменная?

Ковариацию можно вычислить по формуле:

$$cov_{XY} = M(XY) - M(X)M(Y)$$

где ***M*** – математическое ожидание.

In [45]:

1	p
---	---

Out[45]: [1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3, 3.3]

In [46]:

1	s
---	---

Out[46]: array([27, 37, 42, 48, 54, 56, 77, 80])

In [48]:

1	cov= np.mean(p*s)-np.mean(p)* np.mean(s)
2	cov

Out[48]: 11.559375000000031

Зная ковариацию и среднее квадратичное отклонение каждого из двух признаков, можно вычислить **коэффициент корреляции Пирсона**:

$$r_{XY} = \frac{COV_{XY}}{\sigma_X \sigma_Y}$$

```
In [11]: 1 s=np.array([27,37,42,48,54,56,77,80])
          2 s
```

```
Out[11]: array([27, 37, 42, 48, 54, 56, 77, 80])
```

```
In [41]: 1 p=([1.2,1.6,1.8,1.8,2.5,2.6,3,3.3])
          2 p
```

```
Out[41]: [1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3, 3.3]
```

```
In [42]: 1 np.corrcoef(p,s)
```

```
Out[42]: array([[1.          , 0.97318772],
                [0.97318772, 1.          ]])
```

```
In [51]: 1 np.cov(p,s,ddof=1)
```

```
Out[51]: array([[ 0.53928571, 13.21071429],
                [ 13.21071429, 341.69642857]])
```

```
In [52]: 1 np.cov(p,s,ddof=0)
```

```
Out[52]: array([[ 0.471875, 11.559375],
                [ 11.559375, 298.984375]])
```

```
In [53]: 1 np.std(p)
```

```
Out[53]: 0.6869315832017042
```

```
In [54]: 1 np.std(s)
```

```
Out[54]: 17.29116465134723
```

```
1 np.cov(p,s)
```

```
array([[ 0.53928571, 13.21071429],
       [ 13.21071429, 341.69642857]])
```

```
In [55]: 1 np.std(p,ddof=1)
```

```
Out[55]: 0.7343607521414215
```

```
In [56]: 1 np.std(s,ddof=1)
```

```
Out[56]: 18.485032555325095
```

```
In [57]: 1 13.21071429/(0.7343607521414215*18.485032555325095)
```

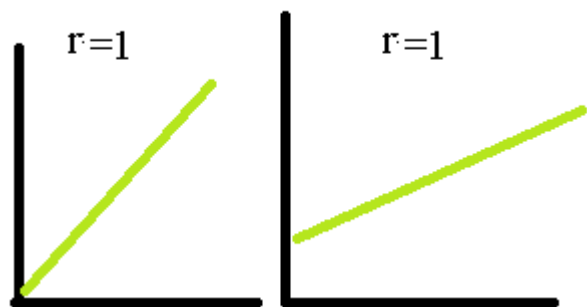
```
Out[57]: 0.9731877247897416
```

```
In [58]: 1 11.559375/(0.6869315832017042*17.29116465134723)
```

```
Out[58]: 0.9731877244740279
```


Плюсы и минусы корреляционного анализа

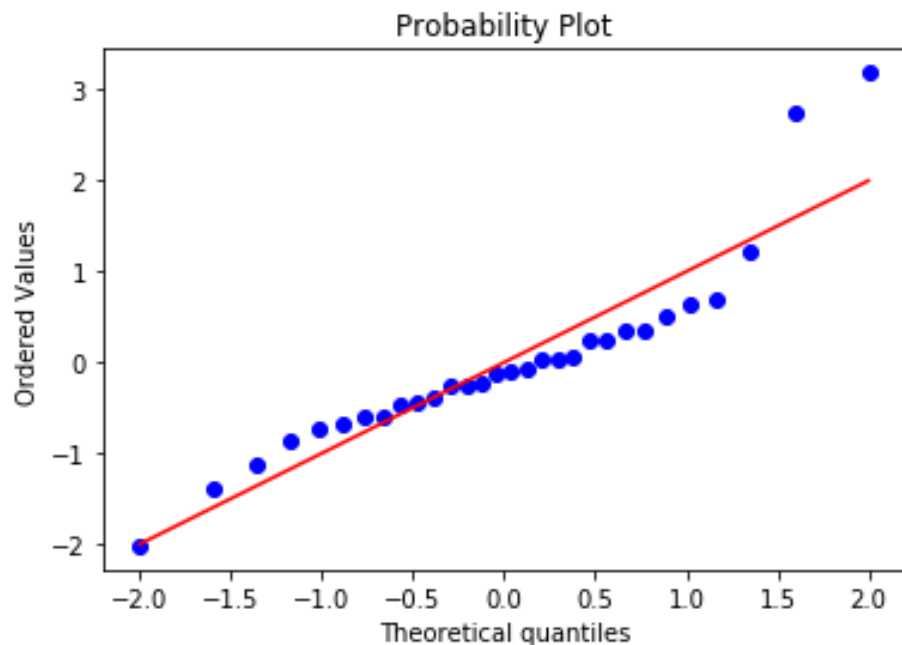
Метод достаточно прост и легко поддается интерпретации, но легко сделать ошибку, посчитав один признак причиной другого. Также данный метод учитывает только наличие линейной связи между признаками.



плюсы	минусы
Простота	Возможна ложная корреляция (есть третья влияющая переменная)
Показывает прямая или обратная связь	Учитывает только наличие линейной связи
На сколько сильна связь	Не показывает угол наклона
	Не показывает точку пересечения с осью y
	$R = 0$ не означает, что нет связи

```
In [52]: import numpy as np
import pylab
import scipy.stats as stats
```

```
stats.probplot(p, dist="norm", plot=pylab)
pylab.show()
```



```
In [39]: from scipy.stats import norm
```

```
In [48]: norm.cdf(1.96)
```

```
Out[48]: 0.9750021048517795
```

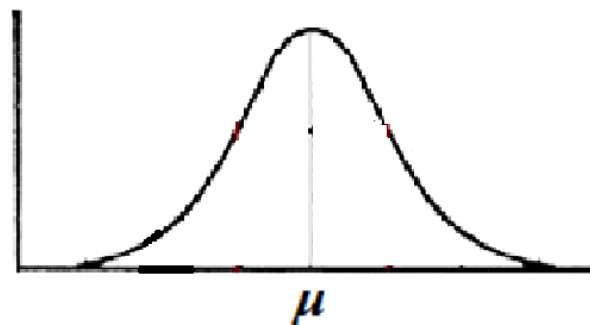
```
In [51]: norm.ppf(0.97500)
```

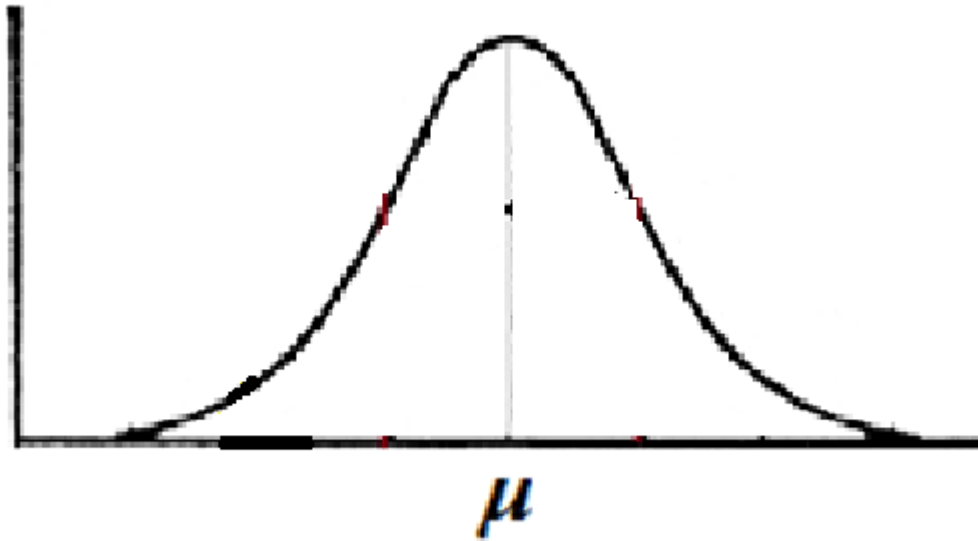
```
Out[51]: 1.959963984540054
```

QQ -график

p

```
Out[35]: array([-0.43470698,  0.35635   ,  0.64508552, -2.02008273,  0.67927448,
                -0.25885112, -0.06118383, -0.74366259, -0.61095246,  0.23907968,
                0.516485   , -0.12540223,  2.73714067, -1.37755292,  1.22983232,
                0.24660636, -0.4000994 , -0.22443351, -0.45667218, -0.25163189,
                0.03710948, -1.13155584,  0.3386186 , -0.59224175, -0.68677237,
                -0.09981573,  3.18777739,  0.03082554, -0.86759793,  0.05009779])
```





```
In [1]: from scipy.stats import norm
```

```
In [48]: norm.cdf(1.96)
```

```
Out[48]: 0.9750021048517795
```

```
In [51]: norm.ppf(0.97500)
```

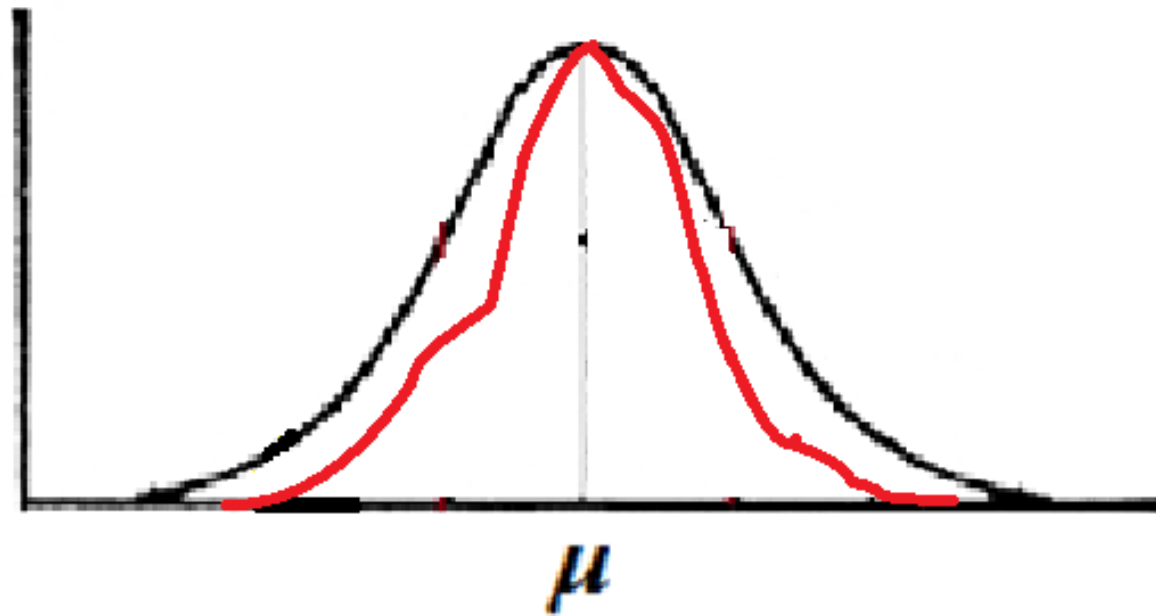
```
Out[51]: 1.959963984540054
```

```
In [5]: norm.ppf(0.10)
```

```
Out[5]: -1.2815515655446004
```

```
In [6]: norm.ppf(0.20)
```

```
Out[6]: -0.8416212335729142
```



ИТОГИ

1. Что такое корреляция
2. Коэффициент корреляции
3. Взаимосвязь величин
4. Ковариация
5. Ограничения корреляционного анализа