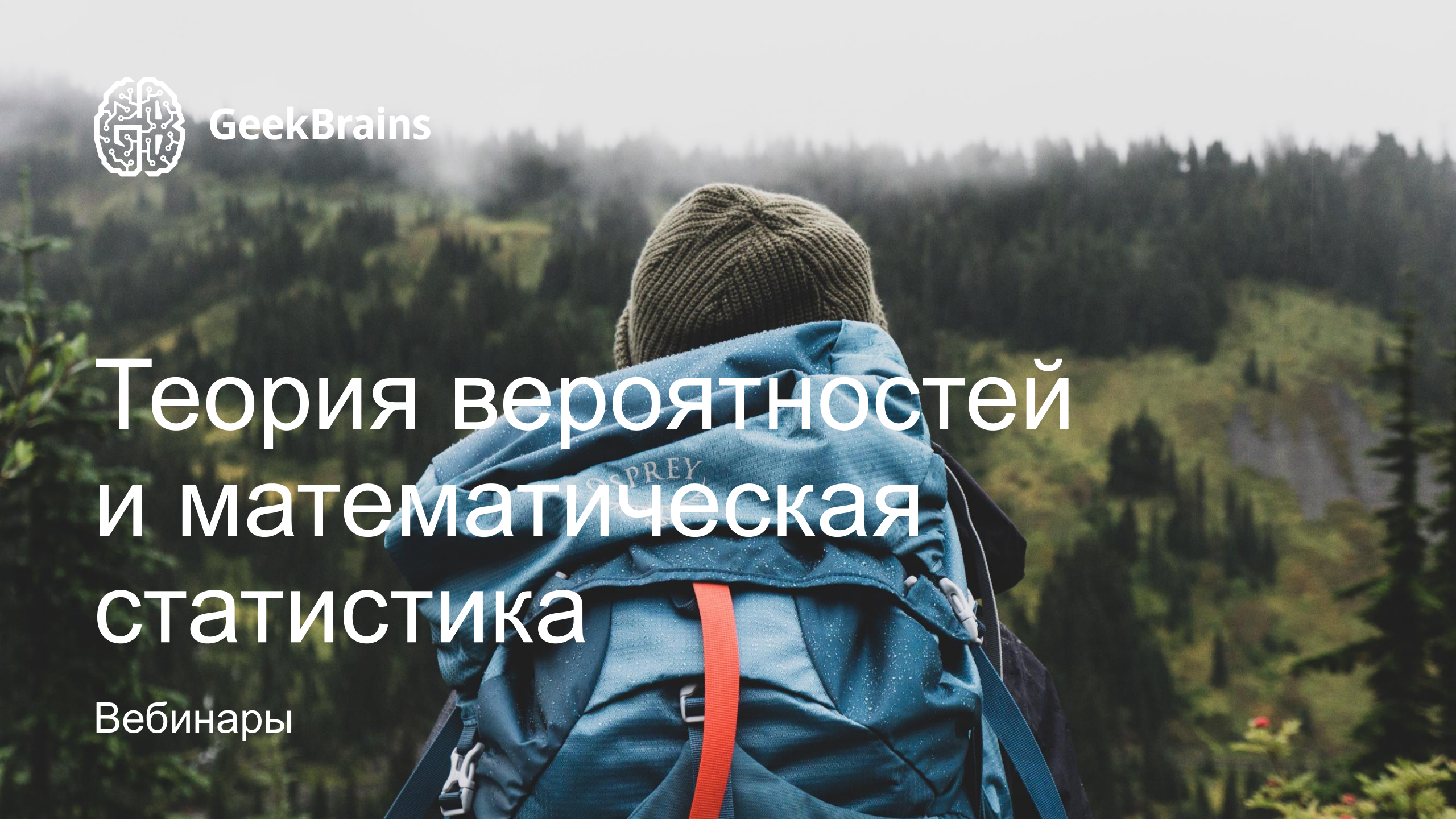
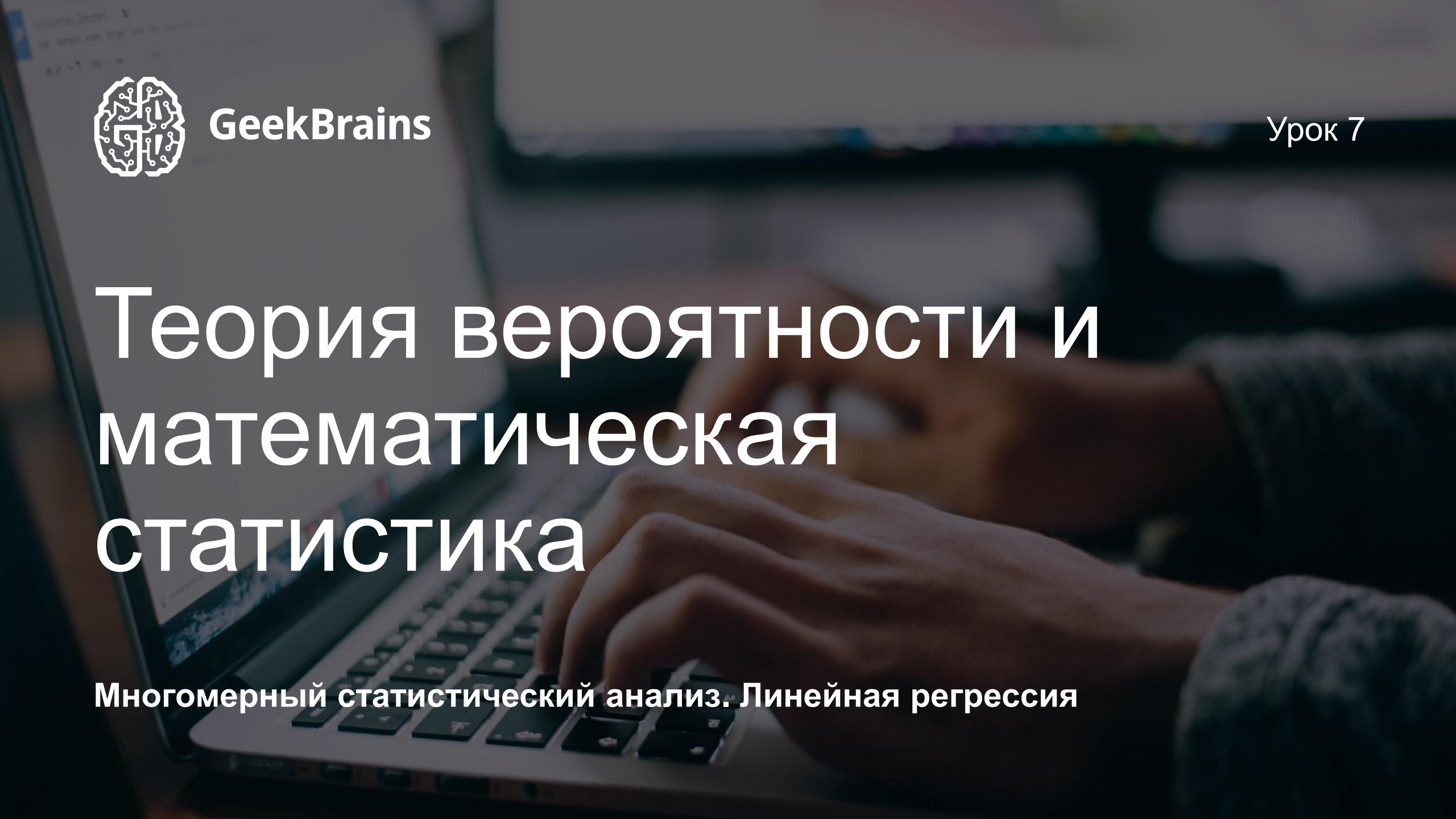


Теория вероятностей и математическая статистика

Вебинары



GeekBrains

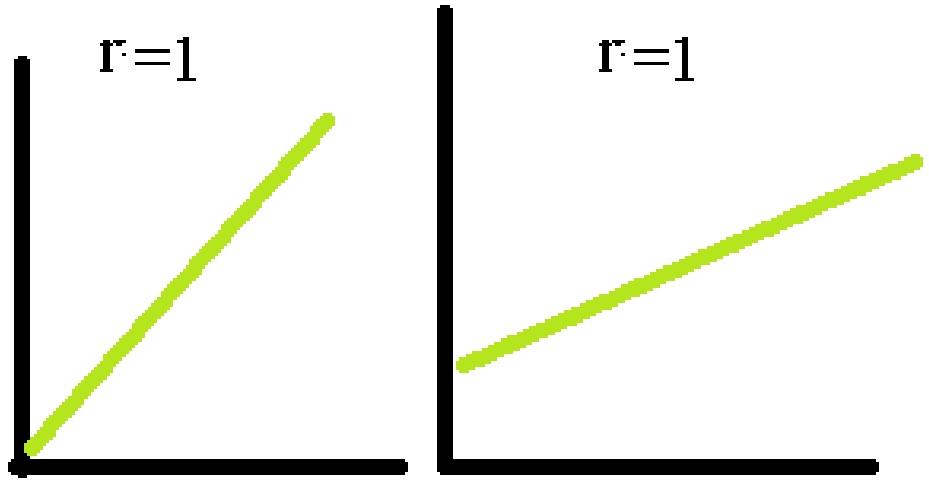
Урок 7

Теория вероятности и математическая статистика

Многомерный статистический анализ. Линейная регрессия

На этом уроке мы изучим:

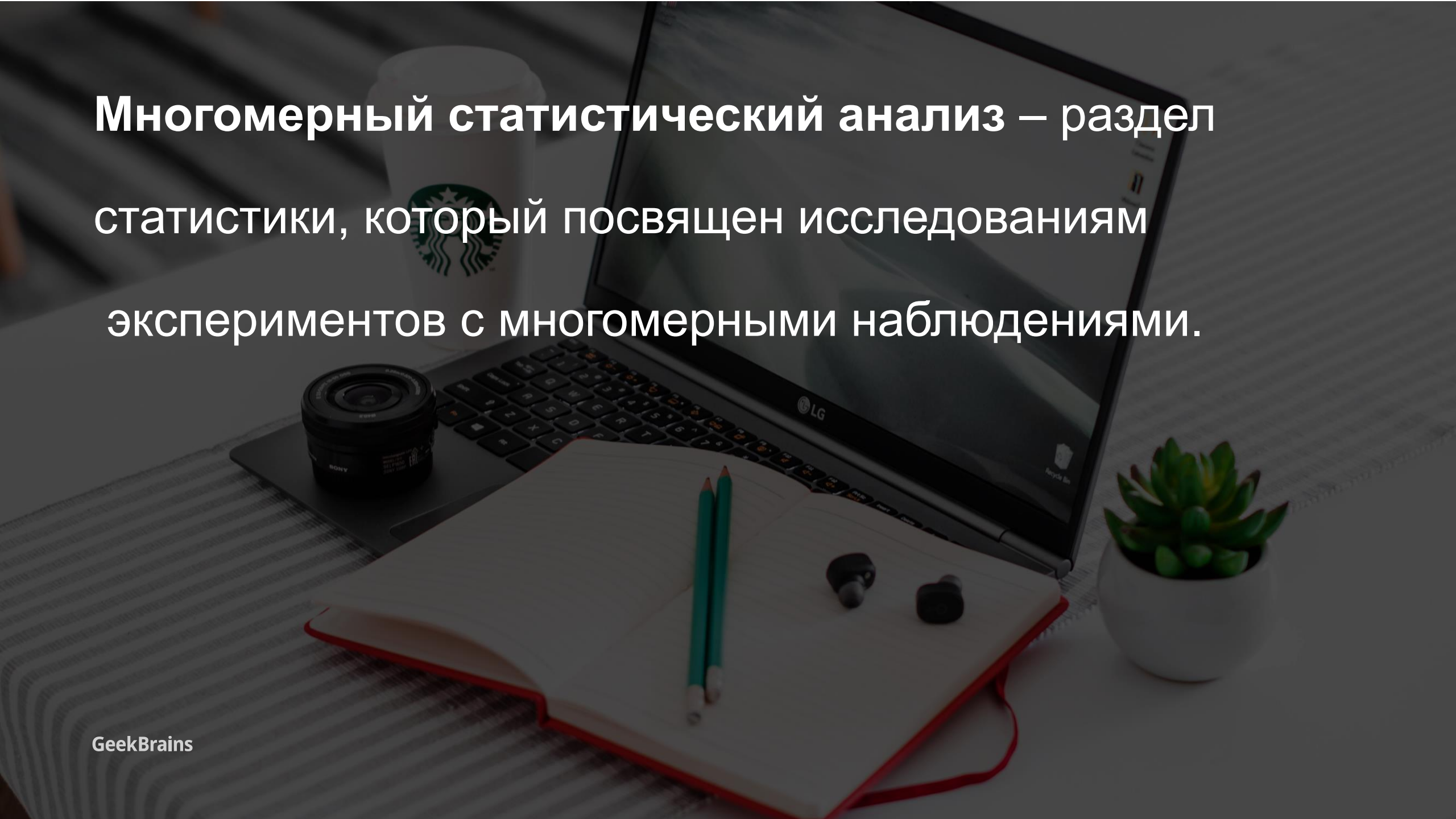
1. Для чего применяют многомерный анализ
2. Что такое линейная регрессия
3. Коэффициент детерминации
4. F-критерий Фишера
5. t-статистика Стьюдента



Недостатки корреляционного анализа :
нет интерсепта
не знаем угла наклона прямой

Линейная регрессия описывает связь признаков (причина) с результатом (следствие).

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

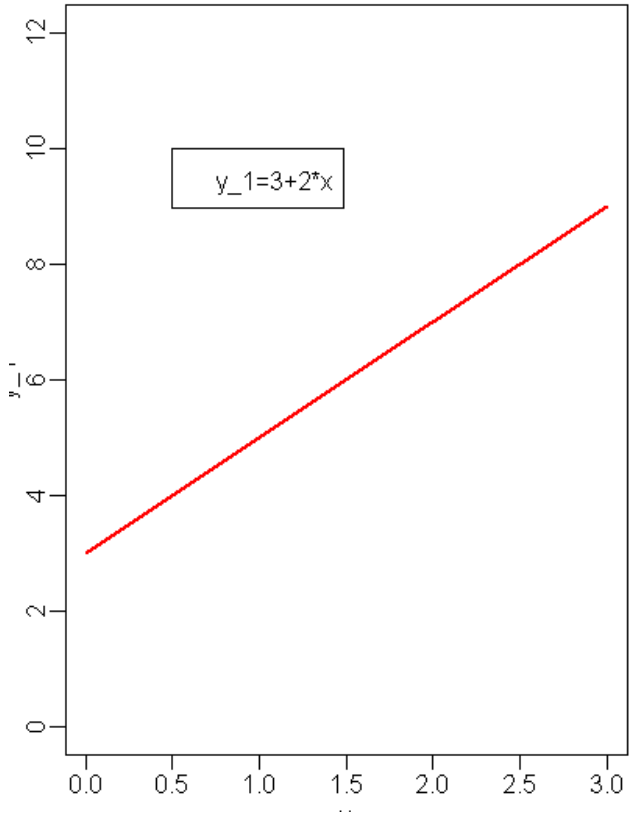


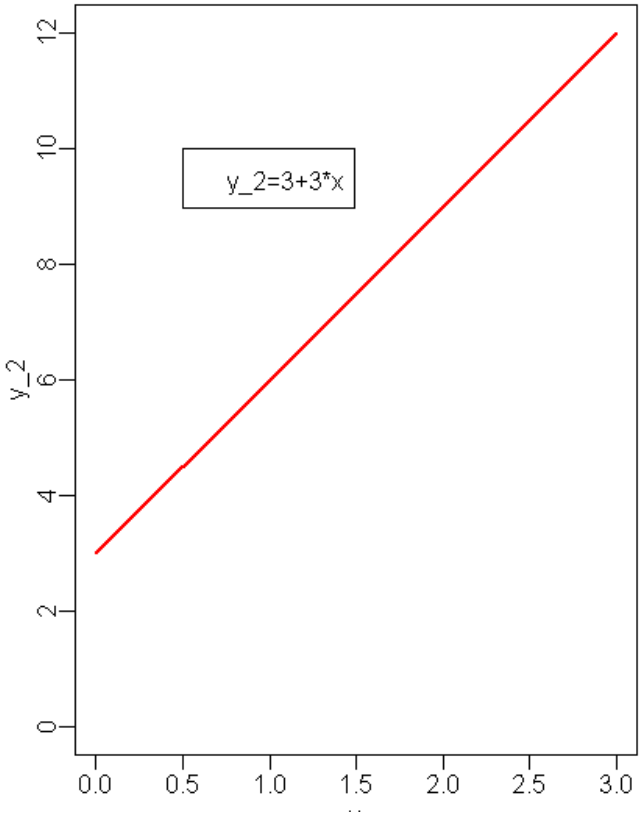
Многомерный статистический анализ – раздел статистики, который посвящен исследованиям экспериментов с многомерными наблюдениями.

Линейная регрессия

Если признак один, то такая линейная регрессия называется парной. Она описывает связь признака ***x*** с результирующим признаком ***y***.

$$y = a + bx$$





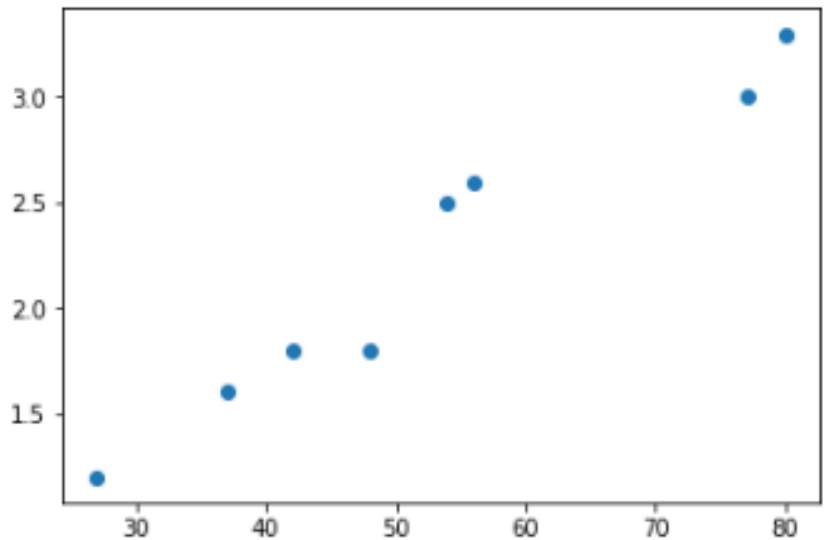
y = 3 + 2*x

X	Y
0	3
1	5
2	7
3	9

y = 3 + 3*x

X	Y
0	3
1	6
2	9
3	12

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



Как образуется линия y_hat?

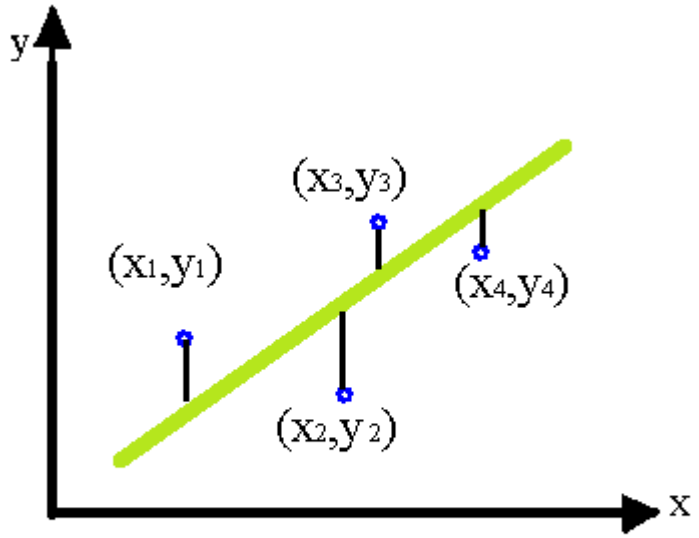
1) После того, как нашли коэффициенты β

,исходя из минимальной среднеквадратичной ошибки,

получаем уравнение линейной регрессии

2) В это уравнение подставляем значения признака и получаем соответствующее значение y. Это оценочное значение y для тех же значений x, что были в измерениях

$$\hat{y} = \beta_0 + \beta_1 * x$$



$$\hat{y} = a + bx$$

$$(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 = \text{минимальна}$$

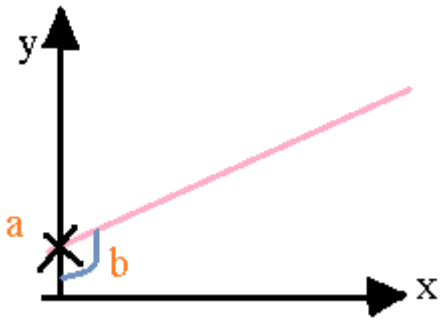
Парная линейная регрессия

Предполагаем связь признаков x и y , которая описывается линейной функцией:

$$y = a + bx$$

где ***a*** и ***b*** – коэффициенты линейной регрессии.

a – показывает, где прямая пересекает ось y при x = 0
b - показывает угол наклона прямой



1. Набираем статистические данные (парные измерения x и y)

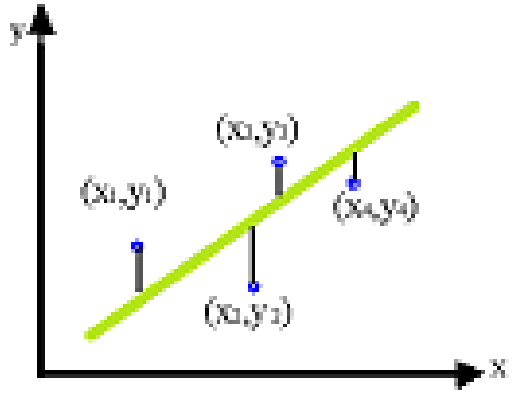
2 Предполагаем линейную связь между x и y

Теоретическое, которое мы изначально предположили

$$y = a + bx$$

3 Считаем коэффициенты, a и b, так, что линия проходит максимально близко к значениям (x,y)

4 Получаем линейную модель $\hat{y} = a + bx$



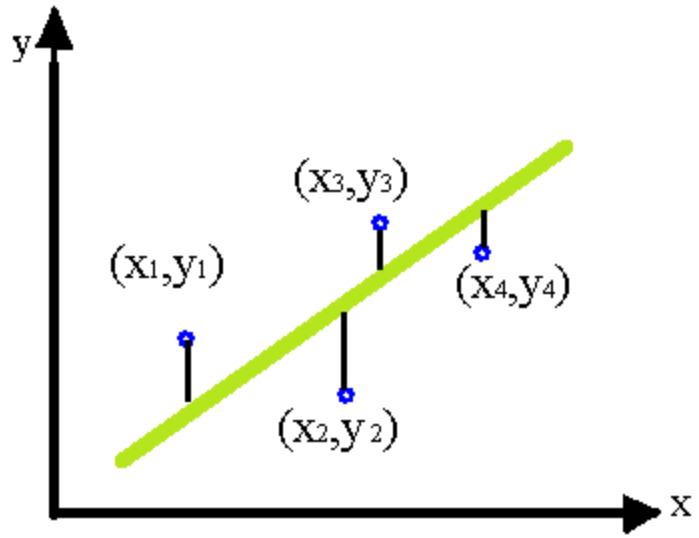
$\hat{y} = a + bx$

Формулы расчета коэффициентов **a** и **b**

* $$b = \frac{n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

* $$a = \bar{y} - b\bar{x}$$

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$



$$\hat{y} = a + bx$$

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

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

In [57]:
```
y=np.array([1.2,1.6,1.8,1.8,2.5,2.6,3,3.3])
y
```

Out[57]:
```
array([1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3. , 3.3])
```

In [59]:
```
b= (np.mean(X*y)-np.mean(X)*np.mean(y))/(np.mean(X**2)-np.mean(X)**2)
b
```

Out[59]:
```
0.03866213744447358
```

In [60]:
```
# 2nd way
```

In [61]:
```
import numpy as np
```

In [62]:
```
n=8
```

In [63]:
```
b=(n*(np.sum(X*y)) - (np.sum(X)* np.sum(y)))/(n*(np.sum(X**2))-((np.sum(X)**2)))
b
```

Out[63]:
```
0.03866213744447358
```

In [64]:
```
a=np.mean(y)-b*np.mean(X)
a
```

Out[64]:
```
0.19040501698457746
```

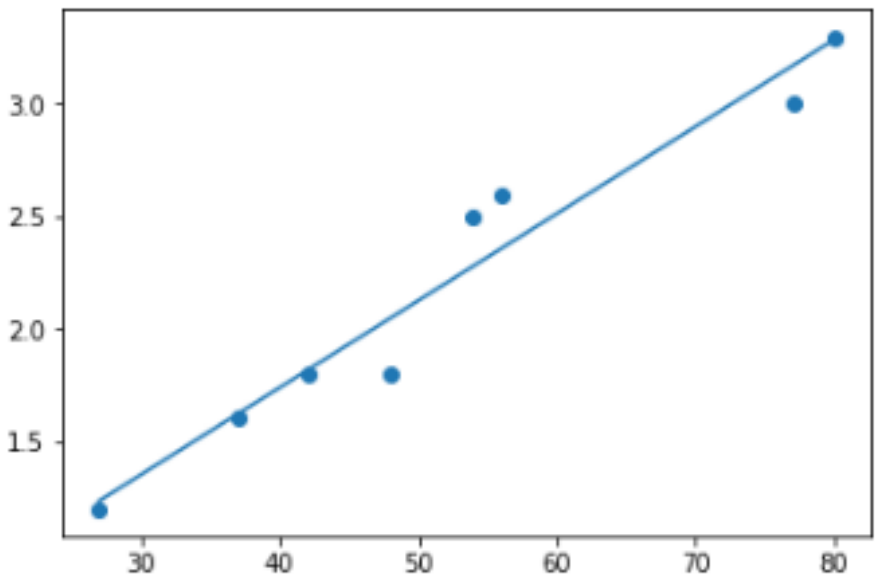
In [65]:
```
y_hat= 0.19+ 0.0387*X
y_hat
```

Out[65]:
```
array([1.2349, 1.6219, 1.8154, 2.0476, 2.2798, 2.3572, 3.1699, 3.286 ])
```

In [66]:
```
y
```

Out[66]:
```
array([1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3. , 3.3])
```

```
1  import matplotlib.pyplot as plt
2  %matplotlib inline
3  plt.scatter(X,y)
4  plt.plot(X,0.19+0.0387*X)
5  plt.show()
```



```
1 y_hat= 0.19+ 0.0387*X
2 y_hat
```

```
array([1.2349, 1.6219, 1.8154, 2.0476, 2.2798, 2.3572, 3.1699, 3.286 ])
```

```
1 y
```

```
array([1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3. , 3.3])
```

```
1 mse=((y-y_hat)**2).sum()/n
2 mse
```

```
0.024967803750000017
```

Функция потерь- мера измерения ошибок, которая функция делает на нашем наборе данных

Одна из самых распространенных функций называется средней квадратичной ошибкой

МАТРИЧНЫЙ МЕТОД ПОИСКА КОЭФФИЦИЕНТОВ

$$y = \beta_0 + \beta_1 * x$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

$$\hat{B} = (X^T * X)^{-1} * X^T * Y$$

```
## матричный метод
```

```
import seaborn as sns
```

```
X= X.reshape((8,1))
X
```

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

```
y=y.reshape((8,1))
y
```

```
array([[1.2],
       [1.6],
       [1.8],
       [1.8],
       [2.5],
       [2.6],
       [3. ],
       [3.3]])
```

In [136]:
```
X=np.hstack([np.ones((8,1)),X])
X
```

Out[136]:
```
array([[ 1., 27.],
       [ 1., 37.],
       [ 1., 42.],
       [ 1., 48.],
       [ 1., 54.],
       [ 1., 56.],
       [ 1., 77.],
       [ 1., 80.]])
```

In [137]:
```
B= np.dot(np.linalg.inv(np.dot(X.T,X)),X.T@y)
B
```

Out[137]:
```
array([[0.19040502],
       [0.03866214]])
```

$$y = \beta_0 + \beta_1 * x$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

$$\hat{B} = (X^T * X)^{-1} * X^T * Y$$

Градиентный спуск

```
X=np.array([27,37,42,48,54,56,77,80])
X
```

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

```
y=np.array([1.2,1.6,1.8,1.8,2.5,2.6,3,3.3])
y
```

```
array([1.2, 1.6, 1.8, 1.8, 2.5, 2.6, 3. , 3.3])
```

```
def mse_(B1,y=y,X=X,n=8):
    return np.sum((B1*X-y)**2)/n
```

```
alpha = 1e-6
alpha
```

```
1e-06
```

```
#mse=(1/n)* np.sum((B1*X-y)**2)
```

```
#mse_p= (2/n)*np.sum((B1*X-y)*X)
```

```
B1=0.1
```

```
n=8
```

```
for i in range (10):
    B1 -=alpha*(2/n)*np.sum((B1*X-y)*X)
    print('B1={}'.format(B1))
```

```
B1=0.099643625
B1=0.09928943698428125
B1=0.09893742253186796
B1=0.09858756830414553
B1=0.09823986104435506
B1=0.09789428757709111
B1=0.0975508348078024
B1=0.09720948972229562
B1=0.09687023938624233
B1=0.0965330709446888
```

```
for i in range (100):
    B1 -=alpha*(2/n)*np.sum((B1*X-y)*X)
    if i%10==0:
        print('Iteration: {i}, B1={B1}, mse={mse}'.format(i=i, B1=B1, mse=mse_(B1)))
```

```
Iteration: 0, B1=0.09619797162156898, mse=9.065656785336
Iteration: 10, B1=0.09295802461401302, mse=8.018824950239837
Iteration: 20, B1=0.08991150325539897, mse=7.093254327129224
Iteration: 30, B1=0.08704685998597309, mse=6.27489845597186
Iteration: 40, B1=0.08435323663819518, mse=5.551337967840769
Iteration: 50, B1=0.0818204232798504l, mse=4.9115921085679695
Iteration: 60, B1=0.07943881951423788, mse=4.345952094816223
Iteration: 70, B1=0.07719939809074752, mse=3.845833773580402
Iteration: 80, B1=0.07509367068789481, mse=3.403647349079136
Iteration: 90, B1=0.0731136557391167, mse=3.0126822000119056
```

```
for i in range (3000):
    B1 -=alpha*(2/n)*np.sum((B1*X-y)*X)
    if i%500==0:
        print('Iteration: {i}, B1={B1}, mse={mse}'.format(i=i, B1=B1, mse=mse_(B1)))
```

```
Iteration: 0, B1=0.07125184817937577, mse=2.6670050391683473
Iteration: 500, B1=0.043278374706070176, mse=0.034094926138670725
Iteration: 1000, B1=0.04198994001494648, mse=0.028509348926302185
Iteration: 1500, B1=0.041930595795328034, mse=0.028497499424626857
Iteration: 2000, B1=0.04192786245042018, mse=0.028497474286546316
Iteration: 2500, B1=0.04192773655484964, mse=0.028497474233217256
```

```
mse_(0.04192773)
```

```
0.028497474233104544
```

Тесноту линейной связи оценивает коэффициент корреляции ***r***.

Чем больше по модулю коэффициент корреляции, тем сильнее связь между ***x*** и ***y***.

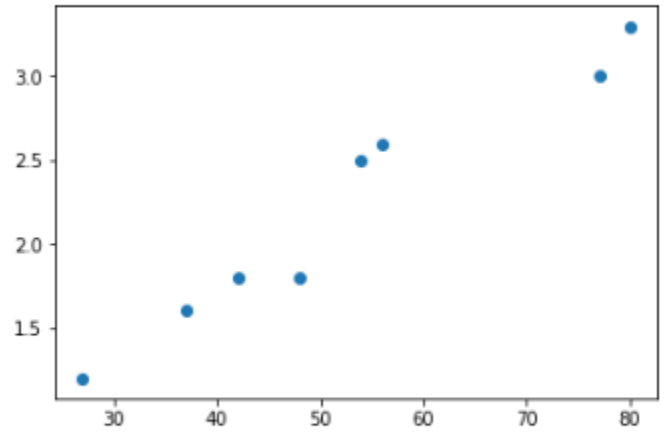
Точность подобранной регрессионной модели показывает **коэффициент детерминации** R^2

R^2 **показывает, какую часть изменчивости y описала регрессионная модель**

$$R^2 = r_{xy}^2$$

R^2 = 0.947

```
1  plt.scatter (X,y)
2  plt.show()
```




```
1  np.corrcoef(X,y)
```

```
array([[1.        , 0.97318772],
       [0.97318772, 1.        ]])
```

5. На этом этапе нам необходимо оценить значимость подобранной модели (т.е. подобранное уравнение линейной регрессии).
Установим уровень значимости $\alpha = 0{,}05$

***F*-критерий Фишера** позволяет оценить значимость модели линейной регрессии.

число измерений n= 8, число параметров p=2 , $\alpha = 0{,}05$
5.1 Находим число степеней свободы
$df_1 = p-1 = 2-1 = 1$
$df_2 = n- p = 8-2=6$
5.2 Рассчитываем *объясненную (фактическую)* сумму квадратных отклонений

SSф=4.72

$$SS_ф = \sum_{i=1} (\hat{y}_i - \bar{y})^2$$

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

5.3 Рассчитываем остаточную сумму квадратных отклонений

SSo= 1.35

$$SS_o = \sum_{i=1} (y_i - \hat{y}_i)^2$$

5.4 Рассчитываем сумму квадратных отклонений в расчете на одну степень свободы

MSф=SSф/df1
MSф= 4.72 / 1= 4.72

MSo=SSo/df2
MSo= 1.35 / 6= 0.225

5.5 Рассчитываем критерий Фишера ***F***

Fp = MSф/ MSo
Fp= 4.72/ 0.225= 20.9

5.6 При ***Fp*** > ***F***т подобранная модель считается **значимой**

Fт зависит от df_1, df_2 и уровня значимости α (вероятность ошибочно отклонить гипотезу о том, что наша модель статистически незначимая)

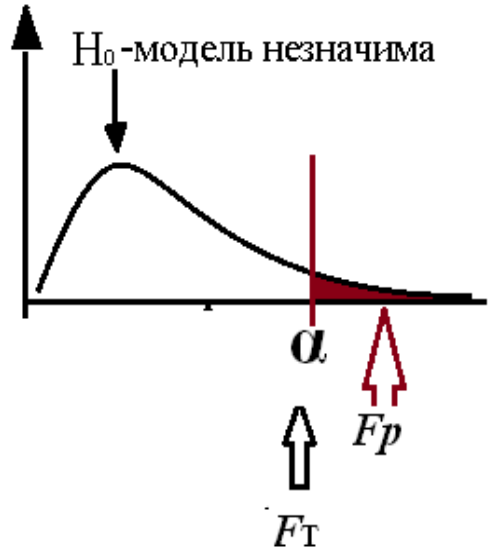
Fт = 5,99

20.9 > 5,99

Модель считается значимой на уровне значимости 0,05

2 Таблица значений *F*-критерия Фишера при уровне значимости $\alpha = 0{,}05$

$n-m-1$	Число независимых факторов									
	1	2	3	4	5	6	8	12	24	∞
1	161,45	199,50	215,72	224,57	230,17	233,97	238,89	243,91	249,04	254,32
2	18,51	19,00	19,16	19,25	19,30	19,33	19,37	19,41	19,45	19,50
3	10,13	9,55	9,28	9,12	9,01	8,94	8,84	8,74	8,64	8,53
4	7,71	6,94	6,59	6,39	6,26	6,16	6,04	5,91	5,77	5,63
5	6,61	5,79	5,41	5,19	5,05	4,95	4,82	4,68	4,53	4,36
6	5,99	5,14	4,76	4,53	4,39	4,28	4,15	4,00	3,84	3,67
7	5,59	4,74	4,35	4,12	3,97	3,87	3,73	3,57	3,41	3,23
8	5,32	4,46	4,07	3,84	3,69	3,58	3,44	3,28	3,12	2,93
9	5,12	4,26	3,86	3,63	3,48	3,37	3,23	3,07	2,90	2,71
10	4,96	4,10	3,71	3,48	3,33	3,22	3,07	2,91	2,74	2,54
11	4,84	3,98	3,59	3,36	3,20	3,09	2,95	2,79	2,61	2,40
12	4,75	3,88	3,49	3,26	3,11	3,00	2,85	2,69	2,50	2,30



F-критерий Фишера

Если фактическое значение F-критерия Фишера больше, чем табличное значение для данных двух степеней свобод и уровня значимости α, то уравнение регрессии признается статистически значимым.

t-статистика Стьюдента позволяет оценить значимость параметров линейной регрессии.

Итоги

1. Для чего применяют многомерный анализ
2. Что такое линейная регрессия
3. Коэффициент детерминации
4. F-критерий Фишера
5. t-статистика Стьюдента