Д. Серікбаев атындағы ШҚТУ

Алгоритмдеу және бағдарламалау тілдері (Python)

PYTHON ЖӘНЕ ДЕРЕКТЕР ҒЫЛЫМЫ

Дәріс 11

Оқытушы:
Жомартқызы Г.

Өскемен
2024

Қалыпты таратылу (Нормальное распределение)

Берілген мәндер айналасында орналасқан массивті құру.

Мысал. Деректердің қалыпты таратылуы:

```
import numpy
x = numpy.random.normal(5.0, 1.0, 15)
print (x)
```

Нәтижесі:

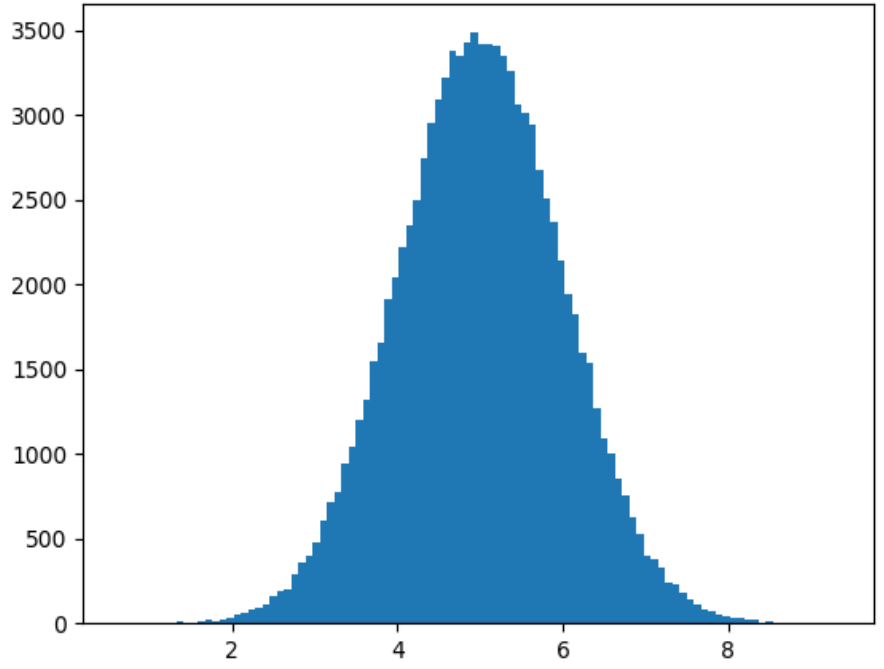
```
[6.48589253 2.71481455 4.10056726 4.67446    5.38789905 4.60871954
 4.40461304 5.56940019 5.08386818 5.57925713 6.07674374 4.08196148
 4.52778292 5.08448122 5.54084547]
```

Орташа мән 5,0 және стандартты ауытқу 1,0 екенін көрсетеміз.

Бұл мәндер шамамен орташа 5,0 шамасында шоғырлануы керек дегенді білдіреді.

Орташа мән 5,0 шамасынан 1,0 шамасына сирек таратылады.

Гистограммадан көріп отырғаныңыздай, мәндердің көпшілігі 4,0-ден 6,0-ге дейін, ал шыңы шамамен 5,0 шамасында.



Ескерту. Қалыпты таралу графигі қоңырау тәрізді пішініне байланысты қоңырау қисығы ретінде де белгілі.

Қалыпты таратылу (Нормальное распределение)

Кездейсоқ деректерді таратылуы.

Қалыпты деректерді таралу мысалы. Әрқайсысы 1000 кездейсоқ сандармен толтырылған екі массив жасайық.

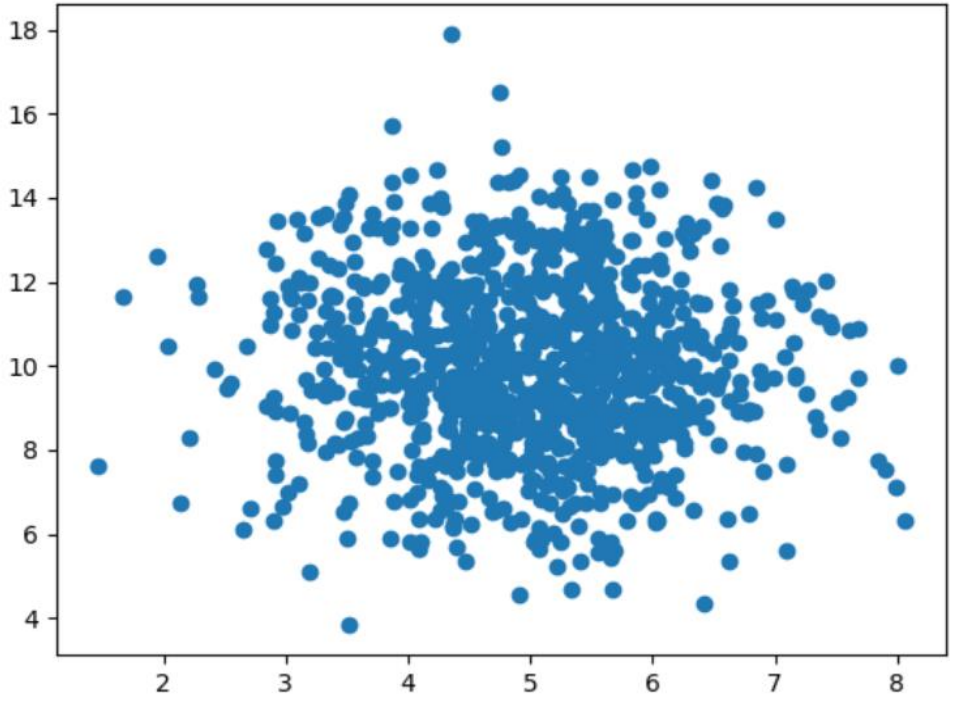
Бірінші массивте орташа мән 1,0 стандартты ауытқумен 5,0 болады.

Екінші массивте орташа мән 2,0 стандартты ауытқумен 10,0 болады:

Мысал. Кездейсоқ деректерді таратылуы

```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.normal(5.0, 1.0, 1000)
y = numpy.random.normal(10.0, 2.0, 1000)
```

Мы видим, что точки сконцентрированы вокруг значения 5 по оси X и 10 по оси Y.



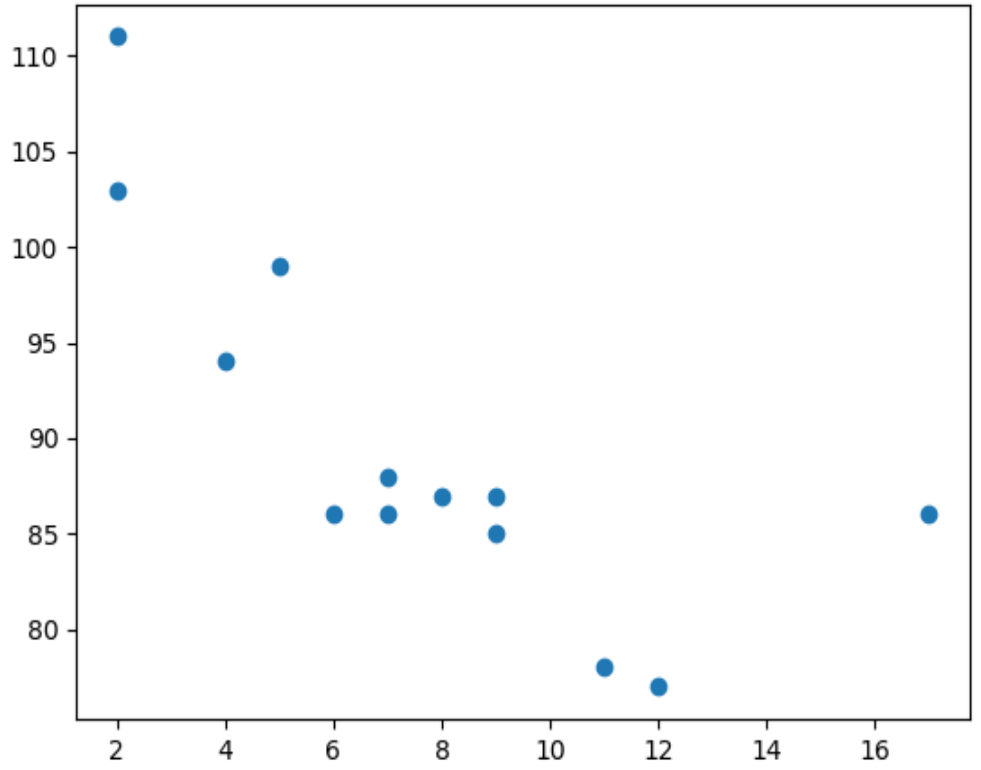
Регрессия.

Регрессия термині айнымалылар арасындағы қатынасты табуда қолданылады. Машиналық оқыту және статистикалық модельдеу бұл қатынасты болашақ оқиғалардың нәтижесін болжау үшін пайдаланады.

Сызықтық регрессия. Сызықтық регрессия деректер нүктелерінің арасындағы байланысты олар арқылы түзу сызық сызу үшін пайдаланады. Бұл сызық болашақ мәндерді болжау үшін пайдаланылуы мүмкін.

Мысал

```
import matplotlib.pyplot as plt
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
plt.scatter(x, y)
plt.show()
...
```



Мысалда X осі жылды, ал Y осі жылдамдықты білдіреді.

13 көліктің жылы мен жылдамдығын тіркелген.

Біз жинаған деректерді сызықтық регрессияда қолдануға болатынын көрейік:

Сызықтық регрессия

Scipy импорттау және сызықтық регрессия сызығын сызу:

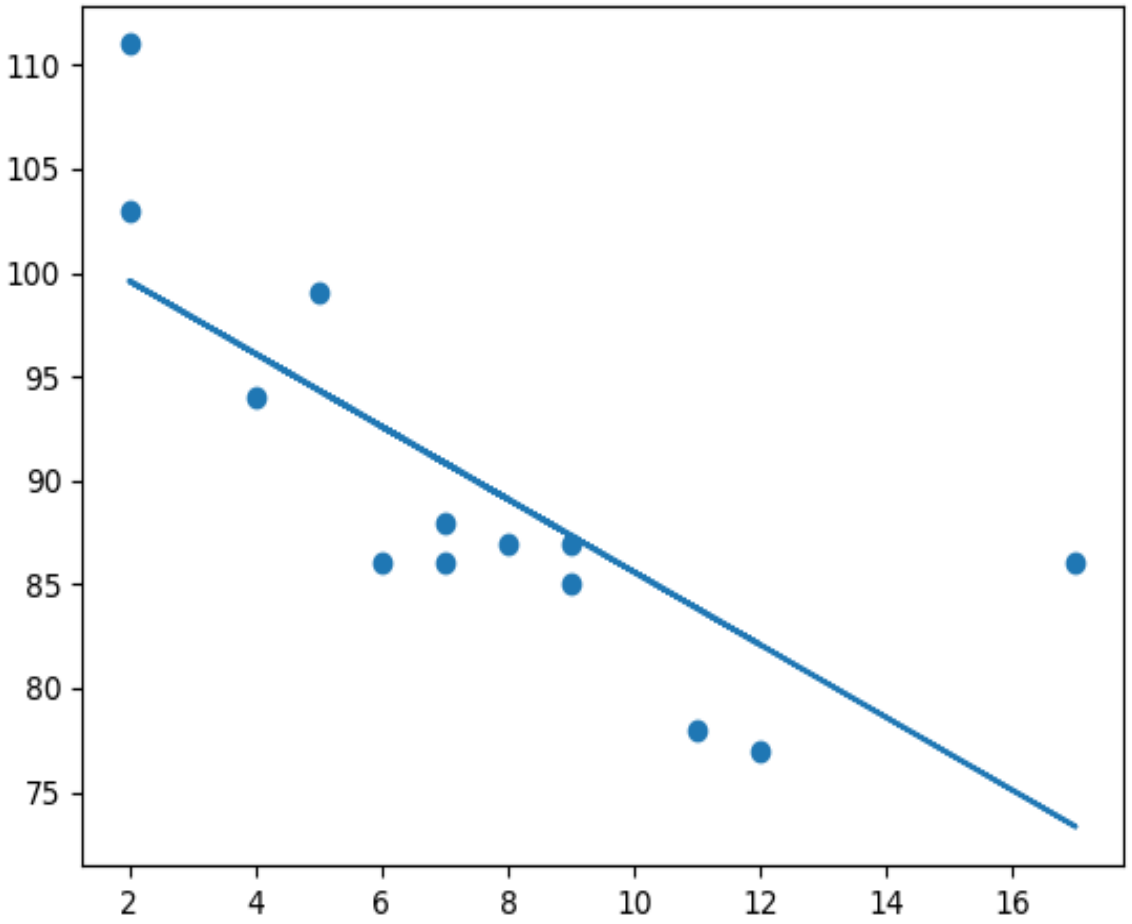
```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg)
#import matplotlib.pyplot as plt
from scipy import stats
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
slope, intercept, r, p, std_err = stats.linregress(x, y)

def myfunc(x):
  return slope * x + intercept

mymodel = list(map(myfunc, x))
plt.scatter(x, y)
plt.plot(x, mymodel)
plt.show()

#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

```
from scipy import stats  -
stats кітапханасын орнату керек
```



Сызықтық регрессия

Сызықтық регрессияның әдісі кейбір маңызды негізгі мәндерін қайтарады:

```
slope, intercept, r, p, std_err = stats.linregress(x, y)
```

slope: float - Наклон линии регрессии.
intercept : float - Пересечение линии регрессии.
rvalue: float - Пирсон корреляция коэффициенті.
pvalue: float - Значение p для проверки гипотезы, нулевая гипотеза которой заключается в том, что наклон равен нулю
stderr : float - Стандартная ошибка оценки наклона (градиента) при предположении об остаточной нормальности.

Сызықтық регрессия

Сызықтық регрессияның әдісі кейбір маңызды негізгі мәндерін қайтарады:

```
slope, intercept, r, p, std_err = stats.linregress(x, y)
```

R байланыс шамасы (корреляция коэффициенті)

X осі мәндері мен Y осі мәндері арасындағы байланыс қандай екенін білу маңызды.
Егер байланыс болмаса, сызықтық регрессия ештеңені болжауға болмайды.
Бұл қатынас – корреляция коэффициенті – r деп аталады.
r мәні -1-ден 1-ге дейін өзгереді, мұнда 0 қосылыстың жоқтығын және 1 (және -1) 100% қосылымды білдіреді.
Python және Scipy модулі сіз үшін бұл мәнді есептейді, сізге тек x және y мәндерін беру қажет.

Мысал

```
from scipy import stats
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
slope, intercept, r, p, std_err =
stats.linregress(x, y)
print(r)
```

Нәтижесі:

```
ect_11_2.py
-0.7585915243761 55
```

Ескерту. -0,76 нәтижесі мінсіз емес, байланыс бар екенін көрсетеді, бірақ болашақ болжамдарда сызықтық регрессияны қолдануға болатынын көрсетеді.

Сызықтық регрессия

Болжам жасау

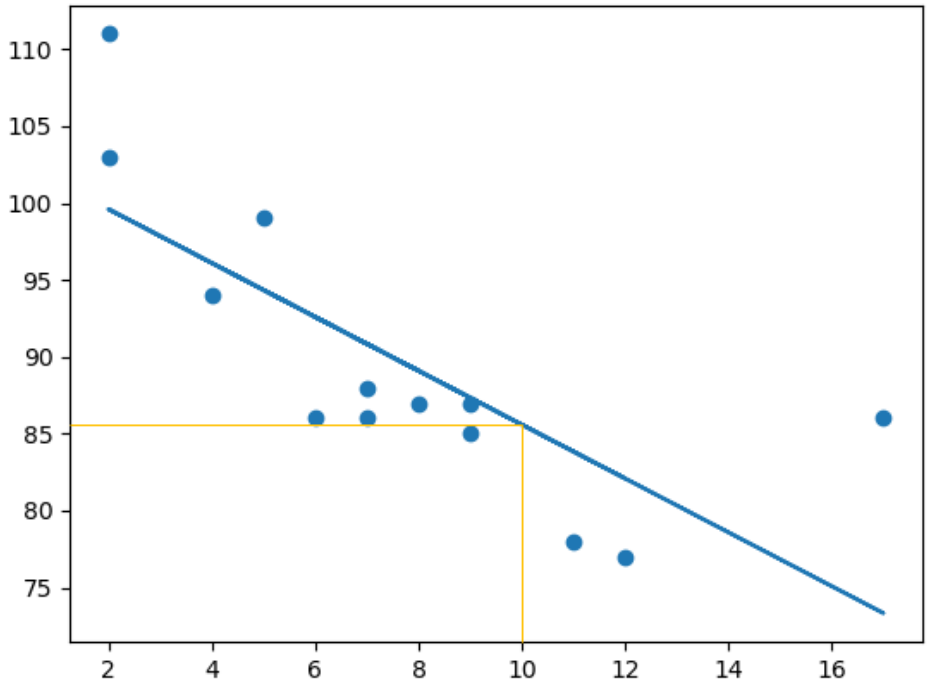
Енді біз жиналған ақпаратты болашақ құндылықтарды болжау үшін пайдалана аламыз.
Мысал: 10 жыл қолданылған көліктің жылдамдығына болжам жасайық.
Ол үшін myfunc() функциясы қажет:

Мысал

```
from scipy import stats
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]
slope, intercept, r, p, std_err =
stats.linregress(x, y)
def myfunc(x):
  return slope * x + intercept
speed = myfunc(10)
print(speed)
```

Нәтижесі:

```
85.59308314937454
```



Мысал 85,6 жылдамдықты болжады, біз оны диаграммадан көре аламыз:

Сызықтық регрессия

Болжам жасау

Енді біз жиналған ақпаратты болашақ құндылықтарды болжау үшін пайдалана аламыз.

Мысал: 10 жыл қолданылған көліктің жылдамдығына болжам жасайық.

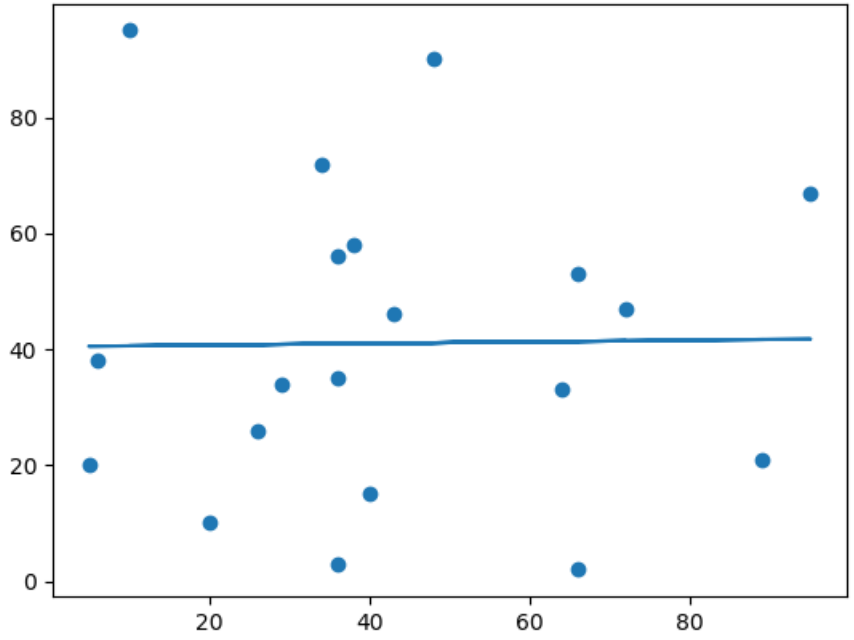
Ол үшін myfunc() функциясы қажет:

Мысал. Өте төмен r мәніналу мысалы

```
import numpy
from scipy import stats
x
= [89,43,36,36,95,10,66,34,38,20,26,29,48,64,
6,5,36,66,72,40]
y
= [21,46,3,35,67,95,53,72,58,10,26,34,90,33,3
8,20,56,2,47,15]
slope, intercept, r, p, std_err =
stats.linregress(x, y)
print(r)
```

Нәтижесі:

```
0.013318141542974 91
```



Нәтиже: 0,013 өте нашар қарым-қатынасты көрсетеді және бұл деректер жинағы сызықтық регрессия үшін жарамсыз екенін айтады.