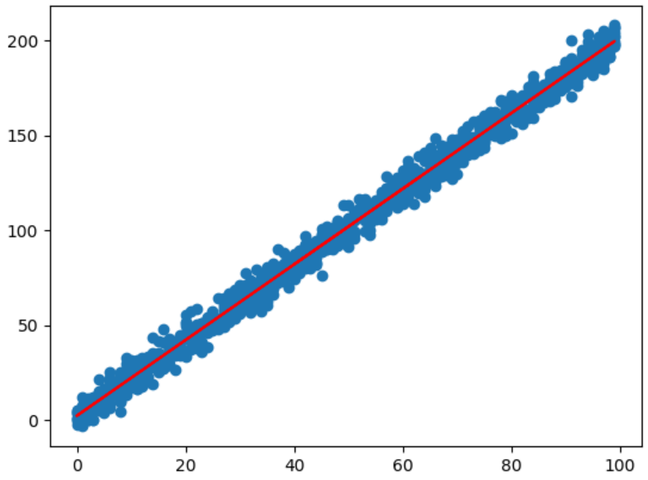
확률통계론 #hw3

2018000337 장호우

Part I

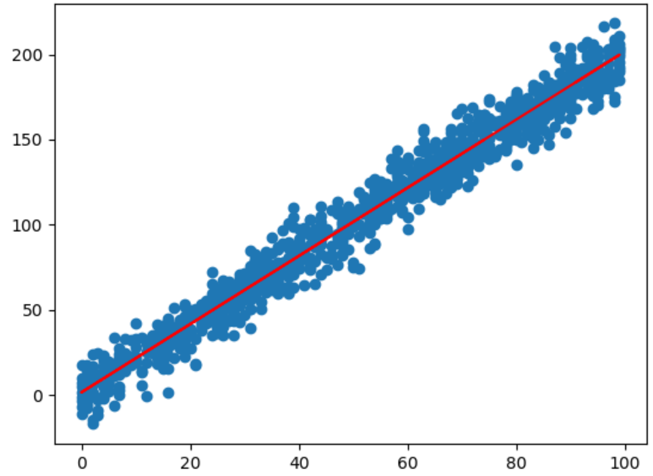
$$Y = aX + b + N(0, \sigma^2)$$

Line fitting: $\hat{Y} = \hat{a}X + \hat{b}$



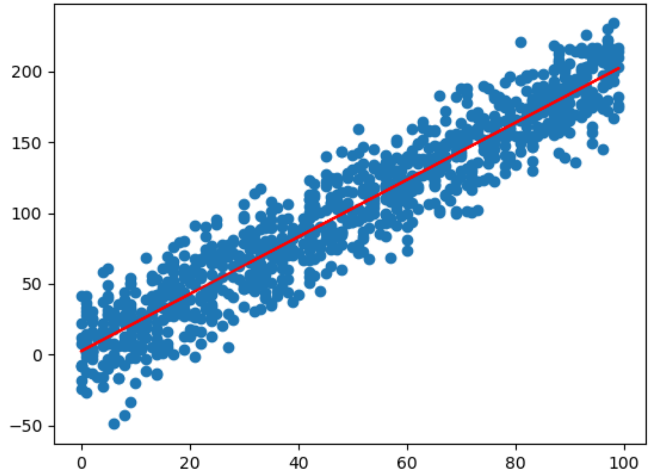
$a = 2, b = 2, \sigma = 5$

The mean of $|\hat{Y} - Y|$ is **3.991**



$a = 2, b = 2, \sigma = 10$

The mean of $|\hat{Y} - Y|$ is **7.970**



$a = 2, b = 2, \sigma = 20$

The mean of $|\hat{Y} - Y|$ is **15.968**

Part I

The mean of $\left|\hat{Y} - Y\right|$ is only related to the standard deviation of normal distribution.

In addition, the mean of $\left|\hat{Y} - Y\right|$ is almost equal to the **0.8** times of the standard deviation of normal distribution.

Part I (Code)

```
import random
import matplotlib
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import linear_model

a = 2
b = 2
std = 5 # 5 or 10 or 20
size = 1000

final_result = []

def once():
    result = []
    x = np.random.randint(0,100,size=size)
    y = a * x + b + np.random.normal(0,std,size=size)

    data = pd.DataFrame({'x': x, 'y': y})
    x = data[['x']]
    y = data['y']

    reg = linear_model.LinearRegression()
    reg.fit(x, y)

    tmp_a = reg.coef_
    tmp_b = reg.intercept_
    tmp_y = tmp_a * x + tmp_b

    for _ in range(0,100):
        n = np.random.randint(0,size)
        result.append(float(format(abs(tmp_y['x'][n] - y[n]),".3f")))

    e = np.mean(np.array(result))
    return float(e)


def total():
    for x in range(10000):
        final_result.append(once())
        print(x)


total()
print("Mean: ",np.mean(np.array(final_result)))
```

Part II

$X : N(m, \sigma^2)$

The running result of program:

10 times	100 times	1000 times
m: 30 std^2: 2401	m: 1 std^2: 1089	m: 17 std^2: 1444
================================	================================	================================
Inferred Mean: 31.67497895371934	Inferred Mean: 1.1865438326537008	Inferred Mean: 17.044635161948726
Inferred Variance: 2220.496571532419	Inferred Variance: 1073.6814350055577	Inferred Variance: 1453.1395322242017

The more samples, the result of inferred mean and variance is more accurate.

Part II (Code)

```
import random
import matplotlib
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

times = 10 # 10, 100 and 1000
size = 100
std = np.random.randint(0,50)
m = np.random.randint(0,50)
list_mean = []
list_var = []

def once(m,std):
    x = np.random.normal(m,std,size=size)
    e = pd.Series(x).mean()
    v = pd.Series(x).var()
    list_mean.append(e)
    list_var.append(v)


def total(times):
    for _ in range(times):
        once(m,std)


total(times)
tmp_mean = np.mean(np.array(list_mean))
tmp_var = np.mean(np.array(list_var))

print(times,"times")
print("m:",m,"std^2:",std*std)
print("==================================")
print("Inferred Mean:",tmp_mean)
print("Inferred Variance:",tmp_var)
```