In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
h = open('X.txt', 'r')

P1 = []
P2 = []
P3 = []

for line in h:
    currentline = line.split(",")
    P1.append(float(currentline[0]))
    P2.append(float(currentline[1]))
    P3.append(float(currentline[2]))

X_df = pd.DataFrame(
    {'X_1': P1,
     'X_2': P2,
     'X_3': P3
    })

# Add a column of 1s (First column) to the observations matrix as it will help us estimate the parameter that
# corresponds to the intercept of the model — the matrix X

X_df.insert(0, 'X_0', 1)

X_Final = X_df.to_numpy()
```

you can use pd.read_csv

In [3]:
```python
h = open('Y.txt', 'r')

Y_list = []

for line in h:
    Y_list.append(float(line))

Y_temp = pd.Series(Y_list)
Y = Y_temp.values.reshape(201, 1)
```

Estimate of $\hat{\beta} = \left(X^T X\right)^{-1} X^T Y$

In [4]:
```python
X_T = X_Final.transpose()

temp_1 = np.dot(X_T, X_df) # (X^T * X)

temp_2 = np.linalg.inv(temp_1) # (X^T * X)^(-1)

temp_3 = np.dot(X_T, Y) # X^T * Y

Beta_hat = np.dot(temp_2, temp_3) # (X^T * X)^(-1) * X^T
```

check X has full column rank -1

In [5]:
```python
print(Beta_hat)
```

```
[[-0.00800698]
 [ 0.88161162]
 [-2.45938171]
 [-0.97715699]]
```

✓

Here β_0 = -0.00800698, β_1 = 0.88161162, β_2 = -2.45938171, β_3 = -0.97715699, for them β_0 is the slope.

The ML-estimator of the unknown parameters σ^2 is $\hat{\sigma}^2_{ML} = \frac{\hat{\epsilon}\hat{\epsilon}}{n}$ where $\hat{\epsilon} = y - X\hat{\beta}$

In [6]:
```python
n = 201

epsilon = Y - np.dot(X_Final, Beta_hat) # y - X * beta_hat
epsilon_mul = np.vdot(epsilon, epsilon) # dot product of epsilon and epsilon
sigma_square_ML = epsilon_mul / n # divided by n
```

In [7]:
```python
print(sigma_square_ML)
```

```
0.9548405627555108
```

✓

The estimate of σ is called the sample standard error of the residuals.

A high standard error shows that sample means are widely spread around the population mean i.e. sample may not closely represent the population. A low standard error shows that sample means are closely distributed around the population mean i.e. sample is representative of the population.

For our case, the value is 0.955, accroding to this artcile [1] the value is acceptable.

[1] Tighe, J., McManus, I., Dewhurst, N.G. et al. The standard error of measurement is a more appropriate measure of quality for postgraduate medical assessments than is reliability: an analysis of MRCP(UK) examinations. BMC Med Educ 10, 40 (2010). https://doi.org/10.1186/1472-6920-10-40

The adjusted estimator $\hat{\sigma}^2_{ad} = \frac{y^T y - \hat{\beta}^T X^T y}{n-p-1}$

In [8]:
```python
p = 3

Y_T = Y.transpose()

Beta_hat_T = Beta_hat.transpose()

t_1 = np.matmul(Y_T, Y) # Y^T * Y

t_2 = np.matmul(Beta_hat_T, X_T) # Beta^T * X^T

t_3 = np.matmul(t_2, Y) # Beta^T * X^T * y

t_4 = n - p - 1

sigma_ad = ((t_1 - t_3) / t_4).squeeze() #squeeze() for convert dataframe to series
```

In [9]:
```python
print(sigma_ad)
```

```
0.9742281883952163
```

✓

We got the $\hat{\sigma}^2_{ad}$ = 0.974, which is also very small and acceptable.

nice discussion!

5/6