In [1]:

```
import jax.numpy as jnp
```

In [40]:

```
n = 500
X_sample_mean = 5.04
X_sample_variance = 8.62
Y_sample_mean = 15.03
Y_sample_variance = 9.24
residual_standard_error = 1.008
```

In [41]:

```
def SST(Y_sample_v, sample_size):
    return (sample_size-1)*Y_sample_v
print("SST = ", SST(Y_sample_variance, n))
```

```
SST =  4610.76
```

In [42]:

```
def SSE(sigma_hat, sample_size):
    return pow(sigma_hat, 2) * (sample_size -2)
print("SSE = ", SSE(residual_standard_error, n))
```

```
SSE =  505.99987200000004
```

In [43]:

```
def SSR(Y_sample_v, sample_size, sigma_hat):
    return SST(Y_sample_v, sample_size) - SSE(sigma_hat, sample_size)
print("SSR = ", SSR(Y_sample_variance, n, residual_standard_error))
```

```
SSR =  4104.760128
```

In [44]:

```
def Sxx(X_sample_v, sample_size):
    return (sample_size-1)*X_sample_v
print("Sxx = ", Sxx(X_sample_variance, n))
```

```
Sxx =  4301.379999999999
```

In [45]:

```
def R_square(Y_sample_v, sample_size, sigma_hat):
    return SSR(Y_sample_v, sample_size, sigma_hat)/SST(Y_sample_v, sample_size)
print("R_square =", R_square(Y_sample_variance, n, residual_standard_error))
```

```
R_square = 0.8902567316451083
```

In [46]:

```
def Beta2(Y_sample_v, sample_size, sigma_hat, X_sample_v):
    return pow(SSR(Y_sample_v, sample_size, sigma_hat)/Sxx(X_sample_v, sample_size), 0.5)
print("Beta2 = ", Beta2(Y_sample_variance, n, residual_standard_error, X_sample_variance))
```

```
Beta2 =  0.9768772278930261
```

In [47]:

```
def Beta1(Y_sample_m, Y_sample_v, sample_size, sigma_hat, X_sample_v, X_sample_m):
    return Y_sample_m - Beta2(Y_sample_v, sample_size, sigma_hat, X_sample_v) * X_sample_m
print("Beta1 = ", Beta1(Y_sample_mean, Y_sample_variance, n, residual_standard_error,X_sample_variar

```

```
Beta1 =  10.106538771419148
```

In [49]:

```
print("SST = ", SST(Y_sample_variance, n))
print("SSE = ", SSE(residual_standard_error, n))
print("SSR = ", SSR(Y_sample_variance, n, residual_standard_error))
print("Sxx = ", Sxx(X_sample_variance, n))
print("R_square =", R_square(Y_sample_variance, n, residual_standard_error))
print("Beta2 = ", Beta2(Y_sample_variance, n, residual_standard_error, X_sample_variance))
print("Beta1 = ", Beta1(Y_sample_mean, Y_sample_variance, n, residual_standard_error,X_sample_variar

```

```
SST =  4610.76
SSE =  505.99987200000004
SSR =  4104.760128
Sxx =  4301.379999999999
R_square = 0.8902567316451083
Beta2 =  0.9768772278930261
Beta1 =  10.106538771419148
```

In [ ]:

```
In [ ]:
```