In [1]:
```
## Calculate the betas for a regression of two X variables onto a Y vector using matrix
```

In [2]:
```
# Import numpy module
import numpy as np

# Consider the Y vector to be a variable labeled "Missed Days at Work";
# there are 12 rows (objects)

Y=np.array([1,0,1,4,3,2,5,6,9,13,15,16])
print(Y)
```

```
[ 1  0  1  4  3  2  5  6  9 13 15 16]
```

In [3]:
```
# Consider column 2 of X to be a variable labeled "Attitude Toward Work" - a 1 to 13 poi
# rating scale where 1 is extremely favorable and 13 is extremely unfavorable;
# and consider column 3 of X to be a variable labeled "Years in Present Position";
# X has 12 rows and 3 columns;  the first column is all 1's

X=np.array([[1,1,1],
            [1,2,1],
            [1,2,2],
            [1,3,2],
            [1,5,4],
            [1,5,6],
            [1,6,5],
            [1,7,4],
            [1,10,8],
            [1,11,7],
            [1,11,9],
            [1,12,10]])
print(X)
```

```
[[ 1  1  1]
 [ 1  2  1]
 [ 1  2  2]
 [ 1  3  2]
 [ 1  5  4]
 [ 1  5  6]
 [ 1  6  5]
 [ 1  7  4]
 [ 1 10  8]
 [ 1 11  7]
 [ 1 11  9]
 [ 1 12 10]]
```

In [4]:
```
# Dimensions of Y: a 12 by 1 vector
Y.shape
```

Out[4]:
```
(12,)
```

In [5]:
```
# Dimensions of X: a 12 by 3 matrix
X.shape
```

Out[5]:
```
(12, 3)
```

In [6]:
```
# Transpose of X
# Here I am taking Transpose as X-Transpose
Transpose=X.T
print(Transpose)
Transpose.shape
```

```
[[ 1  1  1  1  1  1  1  1  1  1  1  1]
 [ 1  2  2  3  5  5  6  7 10 11 11 12]
```

```
 [ 1  1  2  2  4  6  5  4  8  7  9 10]]
```

Out[6]:
```
(3, 12)
```

In [7]:
```python
# Multiplication of X-Transpose and X
# Here I am taking Transpose_M as multiplication of X-Transpose
Transpose_M=Transpose@X
print(Transpose_M)
Transpose_M.shape
```

```
[[ 12  75  59]
 [ 75 639 497]
 [ 59 497 397]]
```

Out[7]:
```
(3, 3)
```

In [8]:
```python
# In this step we are doing Matrix Inverse
Matrix_I=np.linalg.inv(Transpose_M)
print(Matrix_I)
```

```
[[ 0.3169944  -0.0214686  -0.02023368]
 [-0.0214686   0.06093854 -0.07309775]
 [-0.02023368 -0.07309775  0.09703619]]
```

In [9]:
```python
#calculating betas
#for finding betas we have to multiple Matrix Inverse and X Transpose and Y
Betas=Matrix_I@Transpose@Y
print(Betas)
Betas.shape
```

```
[-2.2630379   1.54972927 -0.2385295 ]
```

Out[9]:
```
(3,)
```

In [10]:
```python
#for Calculating Betas
#There is another way of finding betas
#betas=np.linalg.inv((Transpose)@X)@(Transpose)@Y
#print(betas)
#betas.shape
```

```
[-2.2630379   1.54972927 -0.2385295 ]
```

In [11]:
```python
#Calculating the beta values

Beta_0=Betas[0]
print("Beta 0 :", Beta_0)

Beta_1=Betas[1]
print("Beta 1 :", Beta_1)

Beta_2=Betas[2]
print("Beta 2 :", Beta_2)
```

```
Beta 0 : -2.263037902536326
Beta 1 : 1.5497292675976115
Beta 2 : -0.2385294955827928
```

2. Write out the regression equation for predicted Y values using the betas. Leave Y, X and Epsilon as symbols, and insert your estimated beta values:

$$\hat{Y} = \beta0\text{^} + \beta1\text{^} * X1 + \beta2\text{^} * X2 + \varepsilon$$

Sol:

$\beta0$= -2.2630379

$\beta1$= 1.54972927

$\beta2$= -0.2385295

$$\hat{Y} = \beta0\text{^} + \beta1\text{^} * X1 + \beta2\text{^} * X2 + \varepsilon$$

I am substituting the Beta values in the equation.

Y^=-2.2630379 + 1.54972927 * X1 + -0.2385295 $* X2 + \varepsilon$

DTSC - 5502 Assignment

Nandha shriram sabbina

③ showing hand calculations that a defined matrix A and it's inverse A^{-1} gives back the Identity Matrix

given matrix

$$A = \begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix}$$

If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then $A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$$A^{-1} = \frac{1}{2(4) - (3)(-1)} \begin{bmatrix} 4 & -3 \\ -(-1) & 2 \end{bmatrix}$$

$$= \frac{1}{8 - (-3)} \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix}$$

$$= \frac{1}{8+3} \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix}$$

$$= \frac{1}{11} \begin{bmatrix} 4 & -3 \\ 1 & 2 \end{bmatrix}$$

$$\boxed{A^{-1} = \begin{bmatrix} 4/11 & -3/11 \\ 1/11 & 2/11 \end{bmatrix}}$$

calculating $A.A^{-1}$.

$$A.A^{-1} = \begin{bmatrix} 2 & 3 \\ -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} 4/11 & -3/11 \\ 1/11 & 2/11 \end{bmatrix}$$

If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $A' = \begin{bmatrix} p & q \\ r & s \end{bmatrix}$ then

$$A.A^{-1} = \begin{bmatrix} (a \times p) + (b * r) & (a * q) + (b * s) \\ (b * p) + (d * r) & (b * q) + (d * s) \end{bmatrix}$$

similarly for given A and calculated A^{-1}

$$A.A^{-1} = \begin{bmatrix} (2 \times \frac{4}{11}) + 3 * \frac{1}{11}) & (2 \times -3/11) + (3 \times \frac{2}{11}) \\ (-1 * \frac{4}{11}) + (4 * \frac{1}{11}) & (-1 \times \frac{-3}{11}) + (4 \times \frac{2}{11}) \end{bmatrix}$$

$$= \begin{bmatrix} 8/11 + \frac{3}{11} & -6/11 + \frac{6}{11} \\ \frac{-4}{11} + \frac{4}{11} & \frac{3}{11} + \frac{8}{11} \end{bmatrix}$$

$$= \begin{bmatrix} 11/11 & 0 \\ 0 & 11/11 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$ which is identity matrix I

Hence proved that $A.A^{-1} = I$