# Calculate tailArea

/\* Strategy:

\* 1. Calculate X with the formula value

\* 2. Calculate P for that x with a dof = n-2

\* 3. Now we can return the calculation of the tailArea as 1\*2\*P

\*/

//Step 1

Var numeratorX = abs(r) \* sqrt(n-2);

Var denominatorX = sqrt(1-r\*r);

Var x = numeratorX / denominatorX;

//Step 2

Var valueOfP = CalculationP(x, 0.00001,n-2).calculate();

//step3

Var tailArea = 1-2\*valueOfP;

Return tailArea;

# Calculate range

/\* Strategy:

\* 1. Calculate the value of x for p=0.35 and dof = n-2

\* 2. Calculate Sigma value

\* 3. Calculate the the sqrt value of the formula

\* 4. Return the multiplication of all the above

\*/

//step1

Var valueOfX = SearchX(0.35, n-2)

//step 2

Var sigma = geSigma(xLIst, yLIst, beta0, beta1);

//step 3

Var value = sqrt( 1 + (1/n) + ( ( x-xAvg )2 / ( summatory(xLIst, xAvg) )2 ) );

//step4

Return valueOfX \* sigma \* value;

# Calculate Sigma

/\* Strategy:

\* 1. Calculathe the first part of the sqrt (1/n-2)

\* 2. Caculate the value of the summatory and pow it to 2

\* 3. Return the sqr of the multiplication of the above

\*/

//step 1

Var first = 1/n-2;

//step2

Var acum = 0;

for(i : sizeOf( yList ) )

Acum += y[i] - beta0 - beta1 \* x[i]

Var second = acum2;

//step3

Return sqrt(first \* second);

# Calculate LPI & UPI

/\* Strategy:

\* 1. Upi = y + range

\* 2. Lpi = y - range

\*/

Var upi = y + range(xLIst, yLIst, beta0, beta1);

Var lpi = y - range(xLIst, yLIst, beta0, beta1);

Return upy AND lpi;