\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** distributionT

**Method Name:** gammaFunction

**Parameters:** x

**Pseudocode:**

if x == 1

return 1;

if x < 1

return sqrt(PI);

else

return (x-1) \* gammaFunction(x-1);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** distributionT

**Method Name:** distributionT

**Parameters:** dof, dof2

**Pseudocode:**

gammaFunction(dof2) / ((pow(dof\*PI,0.5))\*gammaFunction(dof/2));

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** distributionT

**Method Name:** getResult

**Parameters:** none

**Pseudocode:**

Return result

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** Getter

**Method Name:** calculate

**Parameters:** currX, dof, dof2

**Pseudocode:**

Return pow((currX \* currX) / dof+1, -1 \* ((dof + 1) / 2));

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** Getter

**Method Name:** getResult

**Parameters:** none

**Pseudocode:**

Return result

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** SimpsonDistribution

**Method Name:** calculate

**Parameters:** x, dof, num\_seg

**Pseudocode:**

calculate(0, dof, (dof+1)/2);

finalResult += distT.getResult() \* getterObj.getResult() \* w/3;

while (i smaller than num\_seg) {

if (currX == 0)

end

else if (i % 2 == 0)

Calculate with multiplier 2

else {

Calculate with multiplier 4

}

Increment i

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Class Name:** main

**Method Name:** main

**Parameters:** none

**Pseudocode:**

Ask for x and dof

If dof is not a postivie integer

End program

Calculate initial value fo SimpsonRule

While (SimpsonRule error > error) {

Calculate SimpsonRule

}

Print vlaues of x, dof and SimpsonRule.getResult