***Assignment 1: Be careful with ‘==’***

1. Check the result of this program. Comment why this happened.

**Findings:** It always gives the statement is false, which is not the case mathematically. The actual problem is related to mantissa. **In R real number can be expressed in only 1e3** THAT’s not true!and less than this will be rounded off. So this arose a problem of epsilon. Which comes out as big flaws.

> x<-.001;x

[1] 0.001

1. Specify how the program can be modified to give a correct result

**Finding:** I tried to come up with good solution by declaring a constant epsilon as per R configuration. Program is stated as under.

> test<-function(a,b){

+ VERYSMALL<-4.94065645841247e-324;

+ EPSILON<-0.001;

+ diff= abs(a - b);

+ maxAbs<- max(abs(a) - abs(b));

+ if(abs(diff/maxAbs) < EPSILON){ abs(diff/maxAbs) THIS is equal to 1…. Program is wrongly written

+ print("Teacher said true")

+ } else{

+ print("Teacher lied")};

+

+ if (abs(diff)< VERYSMALL){

+ print("Teacher said true")

+ }

+ };

>

> test(1/3-1/4,1/12)

[1] "Teacher lied"

***Assignment 2: Derivative***

1. Write your own function computing the derivative of function *f(x)=x* in this way. Take ε=10-15

**Findings:**

derivative<-function(x){

e=1.5e-15;

result= ((x+e)-x)/e

}

1. Compute your derivative function at point x=100000.

**Findings:**

> res<-derivative(100000)

> res

[1] 0

1. What is the value you obtained? What is the real value of the derivative? Explain the reason behind the discovered difference.

**Findings:** We got 0, where as 1 is expected from the equation. Actually the nominator and denominator are too much small and R does not support up to that much of mantissa. As a result it considers both nominator and denominator as 0. So that division of both become 0. WRONG

***Assignment 3: Variance***

1. Write your own function *myvar* estimating variance in this way
2. Generate vector x with 10000 random numbers, normally distributed with mean 108 and variance 1
3. For each subset Xi= {x1…xi}, i=1…10000 compute difference Yi= myvar(x)-var(x), where var(x) is a standard variance estimation function in R. Plot the dependence Yi on Xi. Draw necessary conclusions. How well your function works? What is the reason behind such behavior?

**Findings:** Program is written as under and for plotting I use Xi as reference that how many values used for computing respective Yi.

> x<-rnorm(10000,1e8,1)

> myvar

**function**(x){

sqsum<-0;

sum<-0;

Yi<-c();

Xi<-c();

for(j in 1:10000){

sqsum needs to be set to zero within a cycle… all remaining becomes wrong..

for(i in 1:j){

sqsum = sqsum + x[i]^2;

sum = sum + x[i];

};

result<-(1/(j-1))\*(sqsum-(1/j)\*(sum^2));

Yi[j]=result-var(x[1:j]);

Xi[j]=j;

}

print(Yi);

print(Xi);

plot(Yi,Xi);

}

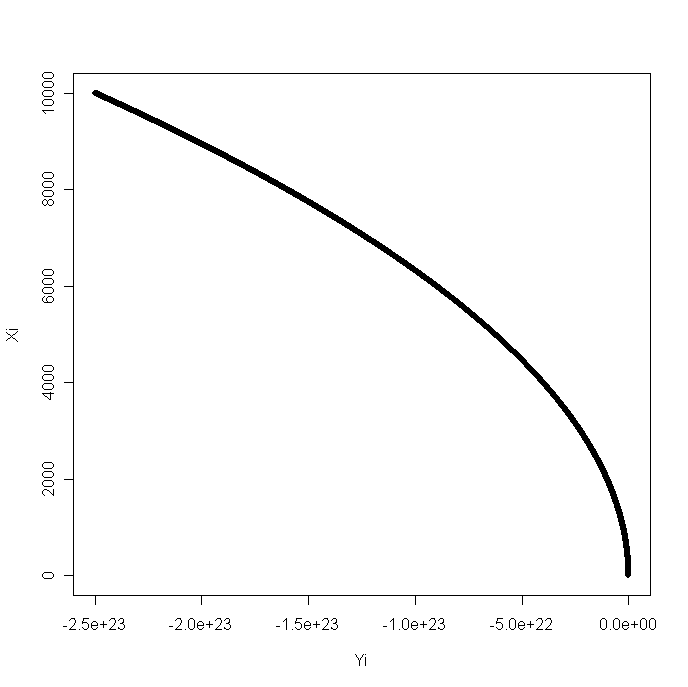


Fig 1.1 Plot between difference if variance i.e. Yi & Xi (index)

From the figure 1.1, we can clearly see that by including more and more values the difference between the variance become larger and larger. So we can conclude the more numbers means larger the error.

**How well your function works? What is the reason behind such behavior?**

My code has complexity level of O((n)2 ). Which is very high that’s why it is taking too much time and this is because of nested for loop. I am sure there must be some way to optimize it but I might need time to look into it.