Lab1_comp_stat

Gowtham & Biswas- LiU id- gowku593 & bisku859

4/14/2020

Question 1

```
## [1] "subtraction is wrong"
```

```
## [1] "subtraction is correct"
```

1

We take two fractional values in first snippet and one fraction and another integer in second snippet. The operations present in two snippets are mathematically correct, but the operator "==" does not allow finite representation of fractions. It is also not possible to represent 1/3 in base 10 or in base 2. Since it is not possible to represent it, nearest number is taken. This will results in giving results which are not correct. In 1st snippet, hence it is giving as wrong.

2

The operator "==" is "exactly equal to". Since the finite representation of value 1/3 is not possible,using "all.equal()" is preferable.

Question 2: Derivative

```
## [1] 1.110223
```

```
## [1] 0
```

2.3

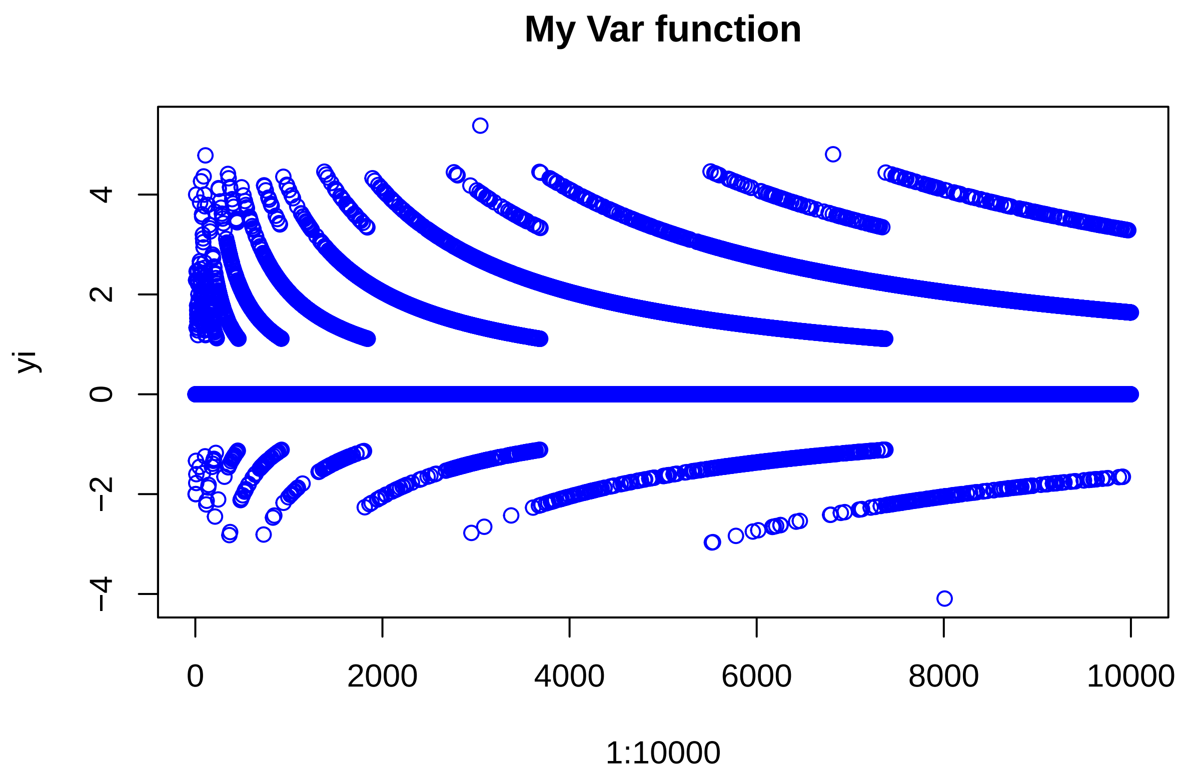
When x=100000, f'(x)= $\frac{100000+10^{-15}-100000}{10^{-15}} = \frac{0}{10^{-15}} = 0$,

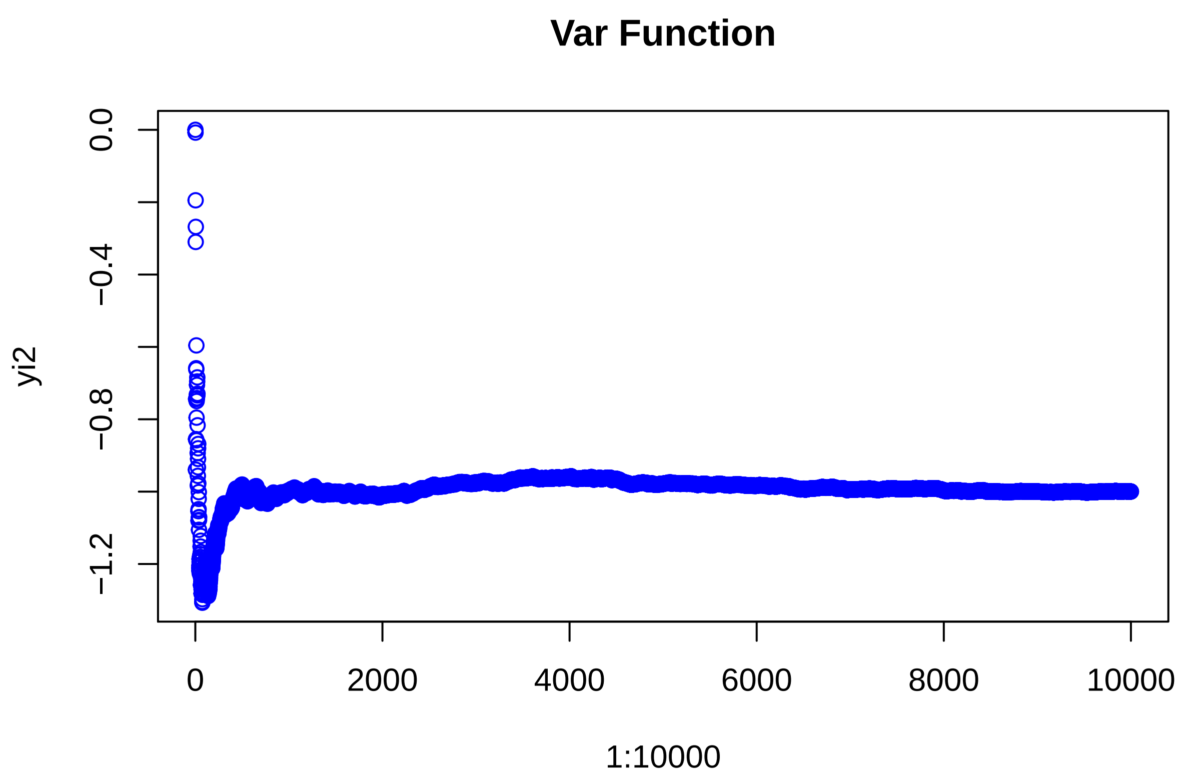
The numerator is behaving like zero when x=100000, because 10^-15 is too small number in comparision with 100000, hence it is neglected.

But when x=1, 10^-15 is not neglected and not a negligible value in comparision with 1. i.e,x=1,

f'(x)= $\frac{1+10^{-15}-1}{10^{-15}} = \frac{10^{-15}}{10^{-15}} = 1$ To prevent this, we can arrange the numbers according to magnitude. This will give: when x=100000 f'(x)=$\frac{100000-100000+10^{-15}}{10^{-15}} = 1$

3 Variance


My Var function
yi
4
2
0
−2
−4
0
2000
4000
6000
8000
10000
1:10000


Var Function
yi2
0.0
−0.4
−0.8
−1.2
0
2000
4000
6000
8000
10000
1:10000

3.3 The variance Yi values are shown in above Mhy Var plot. we can see that the variance never touches 0, but only approaches zero(0).The ideal case should be that should have it touched and stayed at zero for all samples.

We observe that the intervals of Yi which are overlapping, starts to widen as value of x increases i.e for larger subsets of x. Initially the value of Yi starts near to zero and then starts to increase.

In second plot Var function, we can see that output of our function "myVar2()" is around "-1". It starts from zero but settles towards -1 as the sample size progresses.

This behaviour in both the graphs suggests that there could be some issues with the summation precision (rounding of numbers.)

4 Linear Algebra

```
## Loading required package: readxl
```

```
## [1] 1.157834e+15
```

```r
solve(A,b)
#kappa(A)
```

4.3

We get an error, "Error in solve.default(A, b) : system is computationally singular". The function found the system is computationally singular. The matrix A has correlated variables sets. Inverse of A is not possible in this condition.

4.4

kappa(A) provides the condition number of the matrix A, and it is considered it is an "ill conditioned" since the small change in the data leads to large change in result. using kappa(A) to check for condition number gives 1.157834e+15. This high value shows that matrix(A) is "ill conditioned".

4.5

The data set is scaled using scale() function and steps from 2-4 are repeated. The condtion number here is 490486937902 which is lower than the condtion number obtained in step4.4, when we check as mentioned in above step 4. This shows that the scaled data set here gives better answer than the answer in 4.4.

(Have referred R documentation,reference book (Computational Statistics- By James E Gentle)) # Code Appendix

```r
knitr::opts_chunk$set(echo = TRUE)

x1<-1/3; x2<- 1/4
if (x1-x2==1/12){
  print("subtraction is correct")
} else {
    print("subtraction is wrong")
  }
```

```r
x1<-1; x2<-1/2
if(x1-x2==1/2){
  print("subtraction is correct")
}else{
  print("subtraction is wrong")
}
x<-1
f<- function(x){
  Eps<-10^-15
  return(((x+Eps)-x)/Eps)
}

f(x)

x<-100000
f<- function(x){
  Eps<-10^-15
  return(((x+Eps)-x)/Eps)
}

f(x)

# step2 generation of random number
set.seed(12345, kind = "Mersenne-Twister", normal.kind = "Inversion")
x=rnorm(10000, mean = 10^8, sd=1)
a=length(x)
#step 1 defining r function myvar()
myvar<-function(Xi){
  return((1/(length(Xi)-1))*(sum((Xi)^2)-(1/length(Xi))*(sum(Xi))^2))}
i<-c()
# step3  subsetting and plotting
yi<-rep(0,10000)
inp<-c()
for (i in 2:10000) {
  inp=x[1:i]
 yi[i]=myvar(inp)
}
#yi
plot(x=1:10000, y=yi, col="blue",main = "My Var function")

# Step 4:  # The popular formula for Variance is as below
#             s^2= Var = $$\sum_{n=1}^{10} (Xi - $\bar{X}$)^2/ n-1 $$
#The below code snippet estimates the variance  i.e var() as the above code snippet, which is mentioned
set.seed(12345, kind = "Mersenne-Twister", normal.kind = "Inversion")
x <-rnorm(10000, mean=10^8,sd=1)
myvar2<-function(xi){
return((1/(length(xi)-1))*(sum(xi-mean(xi))^2))}
i<-c()
yi2<- rep(0,10000)
inp<-c()
for(i in 2:10000){
inp=x[1:i]
yi2[i]=myvar2(inp) - var(inp)
```

```
}
plot(x=1:10000,y=yi2, col="blue", main="Var Function")
#yi2
# step1
require(readxl)
data<-read_xls("tecator.xls")
#step2 A=xt*x, b=xt*y
y<- data[,"Protein"]  #protein levels
x<- data[,c(-1,-103)]
# observations of the absorbance records of moisture and fat
y<-as.matrix(y)
x<-as.matrix(x)
xt<-t(x)
b<-xt%*%y
A<-xt%*%x
#step 3 A*Beta=b


 #solve(A,b)    # This line is commented so as to knit the rmarkdown document.
#det(A)
#det(b)

#step4
kappa(A)

#  step 5: scale the data set and repeat the steps from 2 to 4.

y<-scale(y)
x<-scale(x)
xt<-t(x)
b<-xt%*%y
A<-xt%*%x
solve(A,b)
#kappa(A)
```