lab01

2022-11-09

Question 1

1. Check the results of the snippets. Comment what is going on.

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

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

```
# and

x1 <- 1 ; x2 <- 1/2
if ( x1 - x2 == 1/2) {
  print("Subtraction is correct")
} else {
  print("Subtraction is wrong")
}
```

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

- 1/3 and 1/12 are infinite decimals, while 1/2 is not.
  Computation of floating-point numbers makes the result different.

2. If there are any problems, suggest improvements.

- give a tolerance $|a - b| < \epsilon$, while ϵ is close to 0

```
x1 <- 1/3 ; x2 <- 1/4
if ( abs((1 / 12) - (x1 - x2)) < exp(-30)) {
  print("Subtraction is correct")
} else {
  print("Subtraction is wrong")
}
```

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

or directly using all.equal()

```
x1 <- 1/3 ; x2 <- 1/4
if (all.equal(1 / 12, x1 - x2) ) {
  print("Subtraction is correct")
} else {
  print("Subtraction is wrong")
}
```

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

Question 2

1. Write your own R function to calculate the derivative of $f(x) = x$ in this way with $\epsilon = 10^{-15}$

```
Q2_derivative <- function(x){

  e <- 10^-15
  fp <-  ((x + e) - x) / e

  return(fp)
}
```

2. Evaluate your derivative function at x = 1 and x = 100000.

```
Q2_derivative(1)
```

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

```
Q2_derivative(100000)
```

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

3. What values did you obtain? What are the true values? Explain the reasons behind the discovered differences.

- True value should be exactly 1. While what we got is 1.110223 at x = 1 and 0 at x = 100000. ϵ + x - x makes floating-point number rounded. And due to 100000 occupies too much mantissa, the result was rounded to 0.

Question 3

1. Write your own R function, myvar, to estimate the variance in this way

```
myvar <- function(x){
  n <- length(x)
  s2 <- (sum(x^2) - (sum(x))^2 / n) / (n - 1)

  return(s2)
}
```

2. Generate a vector x = $(x_1, ..., x_{10000})$ with 10000 random numbers with mean 10^8 and variance 1.

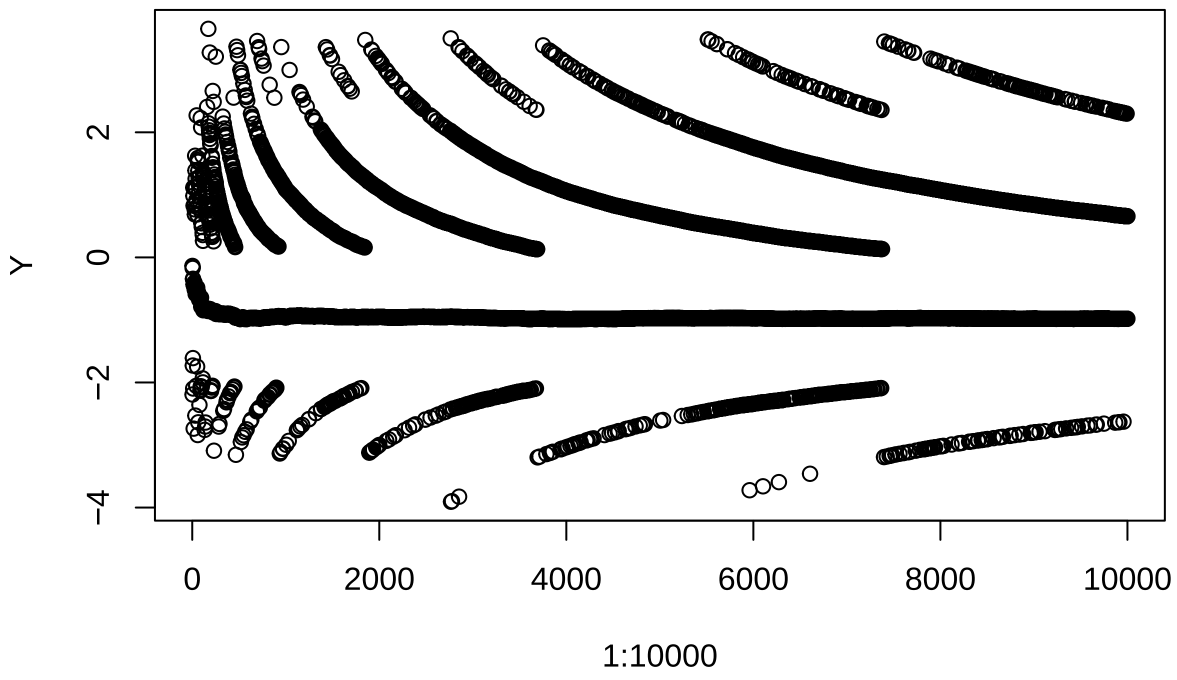
```
x <- rnorm(10000, mean = 10^8, sd = 1)
```

3. For each subset $X_i = \{x_1, ..., x_i, i = 1, ..., 10000$ compute the difference $Y_i = myvar(X_i) - var(X_i)$, where $var(X_i)$ is the standard variance estimation function in R. Plot the dependence Y_i on i. Draw conclusions from this plot. How well does your function work? Can you explain the behaviour?

- Three branches can be seen in the plot: one tends to 0, one tends to -2, and one is concentrated around -1.If the result is correct,all of the result will be equal to 0,so the function doesn't work well.If we look at the calculation results of the myvar function, we will find that many of them are 0. Therefore, the reason may be that two large numbers occupy too many mantissas, so the numbers with smaller digits are rounded, resulting in the subtraction of two large numbers The result is 0.

```
Y <- numeric(length = 10000)
for (i in 1:10000) {
  Y[i] <- myvar(x[1:i]) - var(x[1:i])
}

plot(1:10000, Y)
```



4. How can you better implement a variance estimator? Find and implement a formula that will give the same results as var()?

- We could implement this formula to calculate the variance:

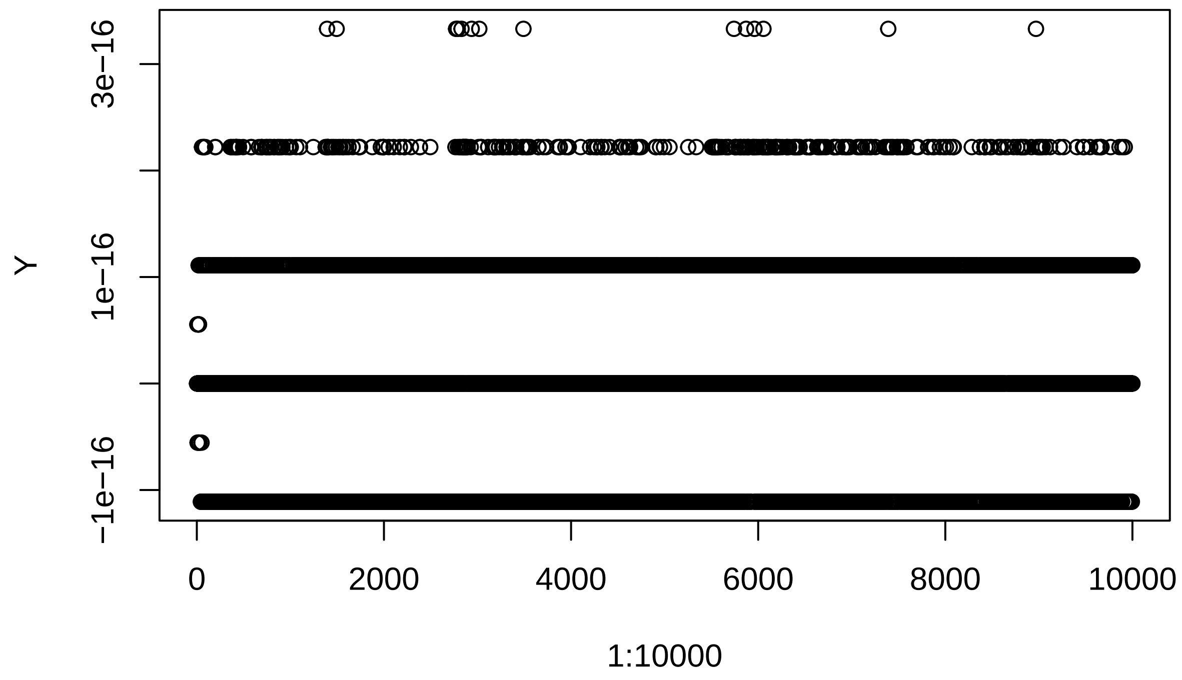
$$Var = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

```
myvar_new=function(x){
  x_mean=sum(x)/(length(x))
  sum_x=sum((x-x_mean)^2)
  return(sum_x/(length(x)-1))
}
myvar_new(x)
```

```
## [1] 0.9812881
```

```
Y <- numeric(length = 10000)
for (i in 1:10000) {
  Y[i] <- myvar_new(x[1:i]) - var(x[1:i])
}

plot(1:10000, Y)
```



Question 4

1. Even if overflow and undeflow would not occur these expressions will not work correctly for all values of n and k. Explain what is the problem in A, B and C respectively.

```
function_a=function(n,k){
  return(prod(1:n)/(prod(1:k)*prod(1:(n-k))))
}

function_b=function(n,k){
  return(prod((k+1):n)/prod(1:(n-k)))
}

function_c=function(n,k){
  return(prod(((k+1):n)/(1:(n-k))))
}
```

- In function A,when k=0,it will return Inf,instead of 1.In function A,B,and C,when n=k,they will return Inf,instead of 1.

2.In mathematical formulae one should suspect oveflow to occur when parameters, here n and k, are large. Experiment numerically with the code of A, B and C, for different values of n and k to see whether oveflow occurs. Graphically present the results of your experiments.

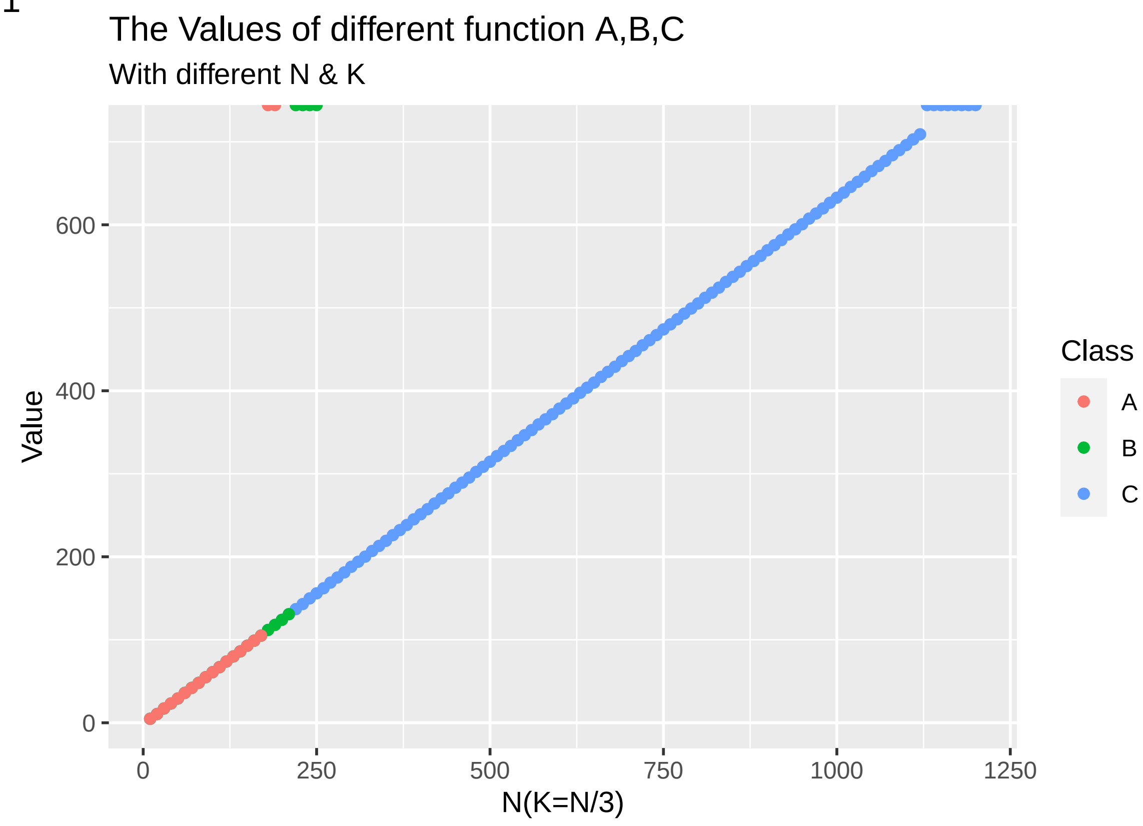
```
record=data.frame()

for(n in seq(10,1200,10)){
  k=floor(n/3)
  a=log(function_a(n,k))
  b=log(function_b(n,k))
  c=log(function_c(n,k))
  #Let's use log to convert the values of a, b and c to linear,
  #so that we can make a more intuitive plot
  record=rbind(record,c(n,k,a,b,c))
}
colnames(record)=c("n","k","a","b","c")
record_plot=data.frame()
record_a=record%>%filter(!is.na(record$a))%>%select(n,k,"values"=a)%>%mutate(Class="A")
record_b=record%>%filter(!is.na(record$b))%>%select(n,k,"values"=b)%>%mutate(Class="B")
record_c=record%>%filter(!is.na(record$c))%>%select(n,k,"values"=c)%>%mutate(Class="C")

record_plot=rbind(record_c,record_b,record_a)

ggplot(data=record_plot,aes(x=n,y=values,color=Class))+geom_point()+
  labs(x="N(K=N/3)",y='Value',
       title="The Values of different function A,B,C",
       subtitle = "With different N & K",
       tag = "Fig. 1")+
  theme_set(theme_bw())
```

Fig. 1



- We notice that when N is around 170,the overflow problem occur in function A,when N is around 220,the overflow problem occur in function B,and when N is around 1130,the overflow problem occur in function C.Among them, k is equal to the result of dividing n by 3 and rounding up.

3.Which of the three expressions have the overflow problem? Explain why.

- The formula A occur the overflow first(Inf when n is larger than 170 and NaN when n is larger than 200),it's because prod(1:n) is much larger than denominator.Then formula B occur the overflow second(Inf when n is larger than 220 and Nan when n is larger than 260).The formula C can calculate much large number than formula A and B.Because this formula ensures that the gap between the numerator and the denominator of each item of the multiplication will not be as large as formula A and formula B