Lab1 Group11

Sofie Jörgensen, Oriol Garrobé Guilera, David Hrabovszki

28 January 2020

Question 1: Be careful when comparing

In this task, we are given two R code snippets, each consisting of a comparision of rational numbers. The first code snippet compares $\frac{1}{3} - \frac{1}{4}$ with $\frac{1}{12}$, and the second code snippet compares $1 - \frac{1}{2}$ with $\frac{1}{2}$. For both code snippets, the comparison is performed by the logical operator `==`, and prints an output, which is either "Subtraction is correct" or "Subtraction is wrong". The code snippets can be found in the Appendix.

1.

After running the two code snippets, we observe that the first snippet prints the output "Subtraction is wrong", while the second snippet prints "Subtraction is correct". In both cases, the comparisions should be mathematically correct, but since floats are rounded in R, the values $\frac{1}{3}$ and $\frac{1}{12}$ are not precisely represented. In other words, all the repeating decimals cannot be stored in R so the value is rounded. Since the snippets use the logical operator `==`, corresponding to exactly equality, this implies that the two sides in the second snippet are not precisely equal. Thus it will return "Subtraction is wrong".

2.

The code causes problems as already mentioned in Question 1, which confirmes that comparisons should be carried out carefully. One suggestion of improvement is to use the functions `all.equal()` and `isTRUE()`, instead of the logical operator `==`. The function `all.equal()` tests if two objects are nearly equal, in contrast to `==` which tests exact equality. The funciton `isTRUE()` is necessary to handle the `FALSE` result of `all.equal()` in the `if` statement. This improvement of the code snippet (see Appendix) prints "Subtraction is correct" when comparing $\frac{1}{3} - \frac{1}{4}$ with $\frac{1}{12}$, which is the mathematically correct output.

Question 2: Derivative

We are going to write our own R function of the definition of a derivative at a point x with a small ϵ, given by the formula $f'(x) = \frac{f(x+\epsilon)-f(x)}{\epsilon}$.

1.

We calculate the derivative of $f(x) = x$, with $\epsilon = 10^{-15}$ using our own R function (see Appendix).

2. and 3.

Now we evaluate our derivative function at $x = 1$ and $x = 100000$ and obtain the results $f'(1) = 1.110223$ and $f'(100000) = 0$ from the output. The true values of the derivatives are 1 for every x, and we expect our function to output a value close to 1. The obtained derivative for $x = 1$ is close to the true value, but for large x we get the result 0. If we consider x within the range $[0, 1[$, we obtain approximately 1. However, when using larger values of x, the difference between large numbers, dominates the small epsilon, which gives us zero in the numerator and thus the derivative evaluated at large points is 0. In general, adding small numbers first will give a better accuracy, than adding large number first, which we are doing when using the definition of the derivative.

Question 3: Variance

The variance based on a vector $\vec{x}$ of n observations can be estimated by using the formula

$$Var(\vec{x}) = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\Big(\sum_{i=1}^{n} x_i\Big)^2\right). \tag{1}$$

1.

We write our own R function, `myvar`, to estimate the variance as given above.

2.

Then we generate 10000 normally distributed random numbers, $x = (x_1, ..., x_{10000})$, with mean 10^8 and variance 1.

3.

The variance can be calculated directly in R by using the standard variance estimation function `var()`. In this task we will compute the difference $Y_i =$`myvar`$(X_i)-$`var`(X_i), for each subset $X_i = \{x_1, ..., x_i\},\ i = 1, ..., 10000$.

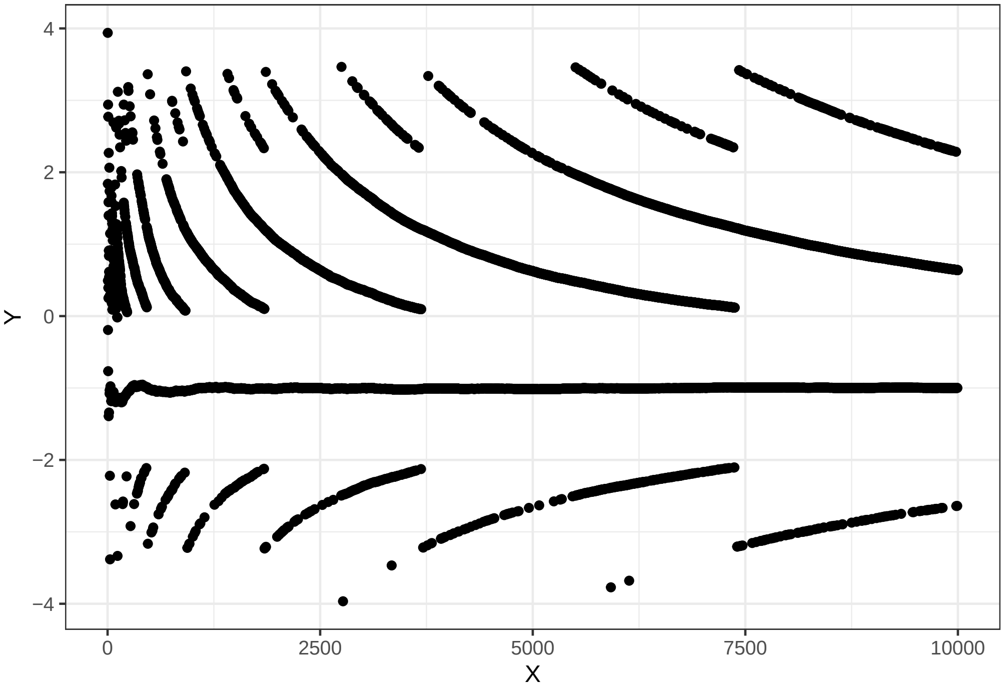

Figure 1: Difference between the functions myvar(X_i) and var(X_i)

From Figure 1 we can observe that…

4.

A better way to implement a variance estimator is to use the formula …

Question 4

1.

Import the data set to R.

```
#4.1
```

2.

Optimal regression coefficients can be found by solving a system of the type $A\vec{\beta} = \vec{b}$ where $A = X^T X$ and $\vec{b} = X^T \vec{y}$. The matrix X are the observations of the absorbance records, levels of moisture and fat, while $\vec{y}$ are the protein levels.

```
#4.2
```

3.

Try to solve $A\vec{\beta} = \vec{b}$ with default solver solve().

```
#4.3
```

4.

Check the condition number of the matrix A.

The condition number κ of square matrix A is: $||A||||A^{-1}||$

```
#4.4
```

5.

Scale the data set and repeat steps 2-4.

```
#4.5
```

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
#1.1
# First code snippet
x1 <- 1/3
x2 <- 1/4
if (x1-x2 == 1/12) {
  print("Subtraction is correct" )
```

```r
} else {
  print("Subtraction is wrong")
}

# Second code snippet.
x1 <- 1
x2 <- 1/2
if (x1-x2 == 1/2) {
  print("Subtraction is correct")
} else {
  print("Subtraction is wrong")
}
#1.2
# Improvement using all.equal() and isTRUE()
x1 <- 1/3
x2 <- 1/4
if (isTRUE(all.equal(x1-x2, 1/12))) {
  print ("Subtraction is correct" )
} else {
  print ("Subtraction is wrong")
}
#2.1
# Function f(x)
f <- function(x) x

epsilon <- 1e-15

# Definition of derivative
derivative <- function(f, x, e){
  (f(x + e) - f(x))/e
}


#2.2 & 2.3
derivative(f, 1, epsilon)
derivative(f,100000, epsilon)
# Variance function
myvar <- function(x) {
  n <-length(x)
  1/(n-1) * (sum(x^2) - sum(x)^2/n)
}
RNGversion('3.6.1')

# Seed
set.seed(1234567890)

# Generate random numbers
x <- rnorm(10000, mean = 10^8 , sd = 1)
# Packages
library(ggplot2) # For plotting

# Difference between the different functions of the variance, for each subset
Y <- c()
```

```r
for (i in seq_along(x)){
  Y[i] <- myvar(x[1:i]) - var(x[1:i])
}

# Create data frame
df <- data.frame(i = seq_along(x), Y)
# Plot the results
ggplot(df, aes(x = i, y = Y)) +
  geom_point() +
  theme_bw() +
  xlab("X")
#4.1


#4.2


#4.3

#4.4

#4.5
```