Homework 1

Santiago Ruiz, Nikita Karetnikov,

2025-03-19

1 Exercise 1

In this exercise, we are asked to calculate the limits for a system of number representations with the following parameters:

- $b = 10$ base
- $m = 3$ mantissa length
- $e_{\min} = -3$
- $e_{\max} = 4$

To calculate the required numbers we would need the following formula:

$$x = (-1)^{w_0} b^e \sum_{i=1}^{m} u_i b^{-i}$$

Let us implement it in R:

```
calculate <- function(w0, b, e, u,m){
  mantissa <- 0

  for (i in 1:m) {
    mantissa <- mantissa + u[i] * b^(-i)
  }

  x <- (-1)^w0 * b^e * mantissa

  return(x)
}
```

Good! Now we will be able to determine the required numbers.

1.1 Determine the largest floating point number

We start with the largest floating point number. We assume that the largest number should have:

- the largest possible exponent
- mantissa with the largest digits in this system (which should be equal to b-1)

```
b <- 10
m <- 3

e_min <- -3
e_max <- 4
```

```r
x_max <- calculate(w0=0, b=b, e=e_max, u= rep(b-1, m),m=m)
print(x_max)
```

```
## [1] 9990
```

1.2 Determine the smallest positive floating point number

Now we should determine the smallest positive floating point number. We assume that the largest number should have:

- the smallest possible exponent
- mantissa with the smallest digits in this system (which should be equal to 0)

But since mantissa should start with non-zero we set its digits to [1,0,0] :

```r
x_min <- calculate(w0 = 0, b = b, e = e_min, u = c(1,0,0), m = m)

print(x_min)
```

```
## [1] 1e-04
```

1.3 Determine the largest floating point number smaller than one.

Now we should find the number that approaches value of 1, but that is smaller then one.

We set: - exponent to 0 - mantissa to have largest digits in this system (which should be equal to b-1)

```r
x_max_2 <- calculate(w0 = 0, b = b, e = 0, u= rep(b-1, m), m = m)
print(x_max_2)
```

```
## [1] 0.999
```

1.4 Determine the smallest floating point number greater than one

Now similarly we should find the number that is closest to one, but is bigger than one.

We set: - exponent to 0 - mantissa digits to be equal to [1,0,1]

```r
x_min_2 <- calculate(w0 = 0, b = b, e = 1, u = c(1,0,1), m = m)
print(x_min_2)
```

```
## [1] 1.01
```

2 Exercise 2

In this exercise we need to understand the limits or R number representation system by running experiments with convergence of the series.

2.1 For which n does the loop stop? (practical part)

We first implement convergence in R. We run the loop until the difference between S_{n+1} and S_n becomes smaller than the smallest number that R can distinguish. .

```r
S_n_prev <- -1
S_n <- 0
n <- 0

while (abs(S_n - S_n_prev) > .Machine$double.eps){
```

```r
  n <-  n + 1
  S_n_prev  <- S_n
  S_n <- S_n_prev + 2^(-2 * n)
}

cat("Loop stops at n =",n)
```

```
## Loop stops at n = 26
```

2.2 For which n does the loop stop? (theoretical part)

Now we need to theoretically approximate the value of n at which the loop stops. We observe that the convergence of our series resembles the formula used to represent numbers in R.

$$\sum_{i=1}^{\infty} 2^{-2i} \qquad x = (-1)^{w_0} b^e \sum_{i=1}^{m} u_i b^{-i}$$

Let us remind ourselves what is mantissa length and base in R is:

```r
m <-  .Machine$double.digits
cat("Mantissa length (m):", m, "\n")
```

```
## Mantissa length (m): 53
```

```r
b <-  .Machine$double.base
cat("Base (b):", b, "\n")
```

```
## Base (b): 2
```

With each iteration, the summation approaches its limit more closely, but the increment decreases rapidly (each term is 2^{-2i}). Due to floating-point representation limits in R (with a mantissa length $m = 53$ bits for double precision), the smallest distinguishable difference around 1 is about 2^{-53}.

The summation stops updating numerically when the next term 2^{-2n} becomes smaller than this limit:

$$2^{-2n} \approx 2^{-m} \quad \Rightarrow \quad n = \frac{m}{2} = \frac{53}{2} = 26.5$$

26.5 is quite close to the practical stopping point we found earlier.

3 Exercise 3

In this exercise we are asked to approximate exp(x) using Taylor series.

$$\exp(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

3.1 (i)

We implement an algorithm in R that approximates the exponential function $\exp(x)$ using the Taylor series expansion. The algorithm iteratively adds terms from the Taylor series until the absolute value of the next summand is smaller than `.Machine$double.eps^(1/2)` times the absolute value of the current approximation.

We run it to identify how many summands we need to approximate $\exp(x = 10)$

```r
calculate_exp <- function(X){
  term <- 1
  sum <- term
  n <- 0
  epsilon <- .Machine$double.eps^(1/2)


  while (abs(term) > epsilon * abs(sum)) {

    n <- n + 1
    term <- (X^n) / factorial(n)
    sum <- sum + term


  }
  return(sum)

}

X <- 40



cat("Our function:", calculate_exp(X), "\n")
```

```
## Our function: 2.353853e+17
```

```r
cat("R actual function", exp(X), "\n")
```

```
## R actual function 2.353853e+17
```

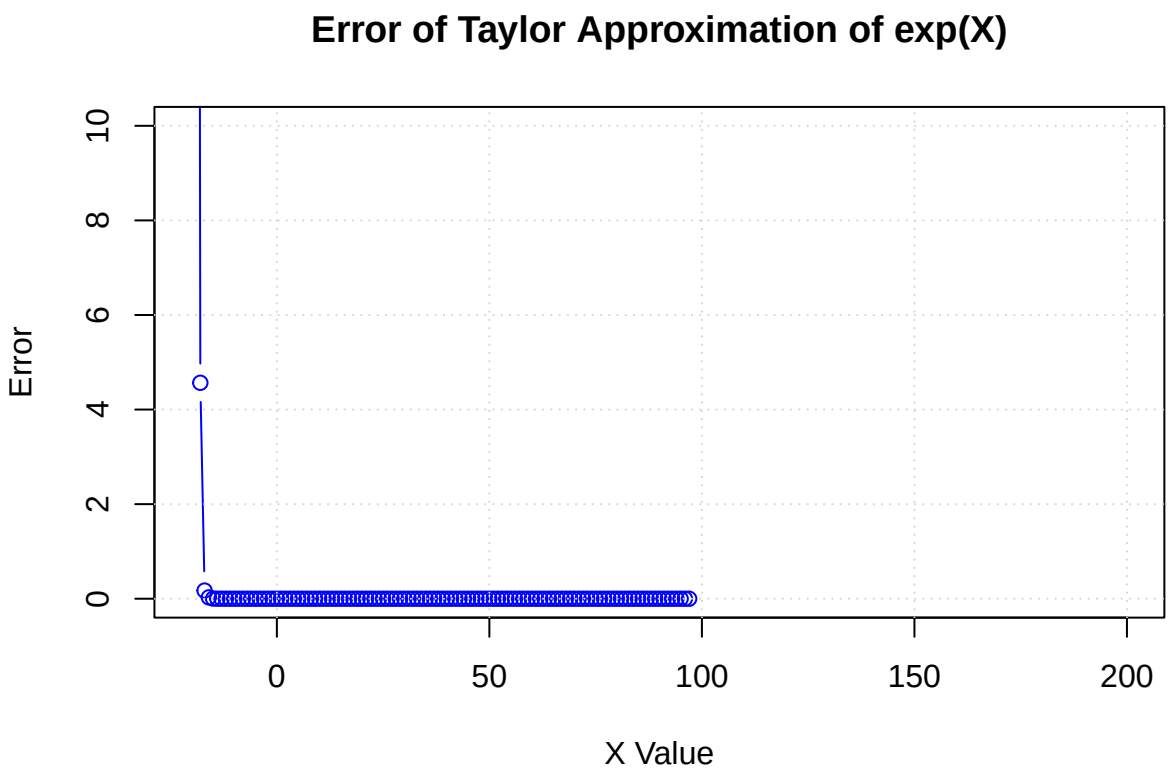
Looks that our implemented function closely matches R's built-in calculation of the exponential function. the exponent.

3.2 (ii)

Next, we evaluate the approximation for various values of X, both positive and negative. We calculate the MAPE for values ranging from -20 to 200.

```r
X_values <- seq(-20, 200, by=1)
errors <- numeric(length(X_values))

for (i in seq_along(X_values)) {
  X <- X_values[i]


  mape <- mean(abs((exp(X) - calculate_exp(X)) / exp(X))) * 100



  errors[i] <- mape
}


plot(X_values, errors, type = "b", col = "blue",
```

```
     xlab = "X Value", ylab = "Error",
     main = "Error of Taylor Approximation of exp(X)",
     ylim = c(0, 10))  # Adjust these limits as needed for your data
grid()
```



We notice that for lower values, MAPE increases starting approximately from -10.
For values of X higher than 100, MAPE is absent as the approximation of the exponent is probably equal to ∞.

Let us check:

```
X <- 101

cat("Our function:", calculate_exp(X), "\n")
```

```
## Our function: Inf
```

```
cat("R actual function", exp(X), "\n")
```

```
## R actual function 7.30706e+43
```

```
X <- -20

cat("Our function:", calculate_exp(X), "\n")
```

```
## Our function: 4.992704e-09
```

```
cat("R actual function", exp(X), "\n")
```

```
## R actual function 2.061154e-09
```

Indeed, either our approximations are too far from the actual values of the function, or our approximations are equal to infinity.

3.3 (iii)

Let us suggest a modification how to fix the numerical instability. We use the following property:

$$e^x = \left(e^{x/n}\right)^n$$

We implement it in our updated function as a wrapper:

```
calculate_exp_2 <- function(x, n = 10) {
    x_n <- x / n
    result <- calculate_exp(x_n)^n
    return(result)
}
```

Let's first check it for single values:

```
X <- -20

cat("Our function:", calculate_exp_2(X), "\n")
```

```
## Our function: 2.061154e-09
```

```
cat("R actual function", exp(X), "\n")
```

```
## R actual function 2.061154e-09
```

```
X <- 101

cat("Our function:", calculate_exp_2(X), "\n")
```

```
## Our function: 7.30706e+43
```

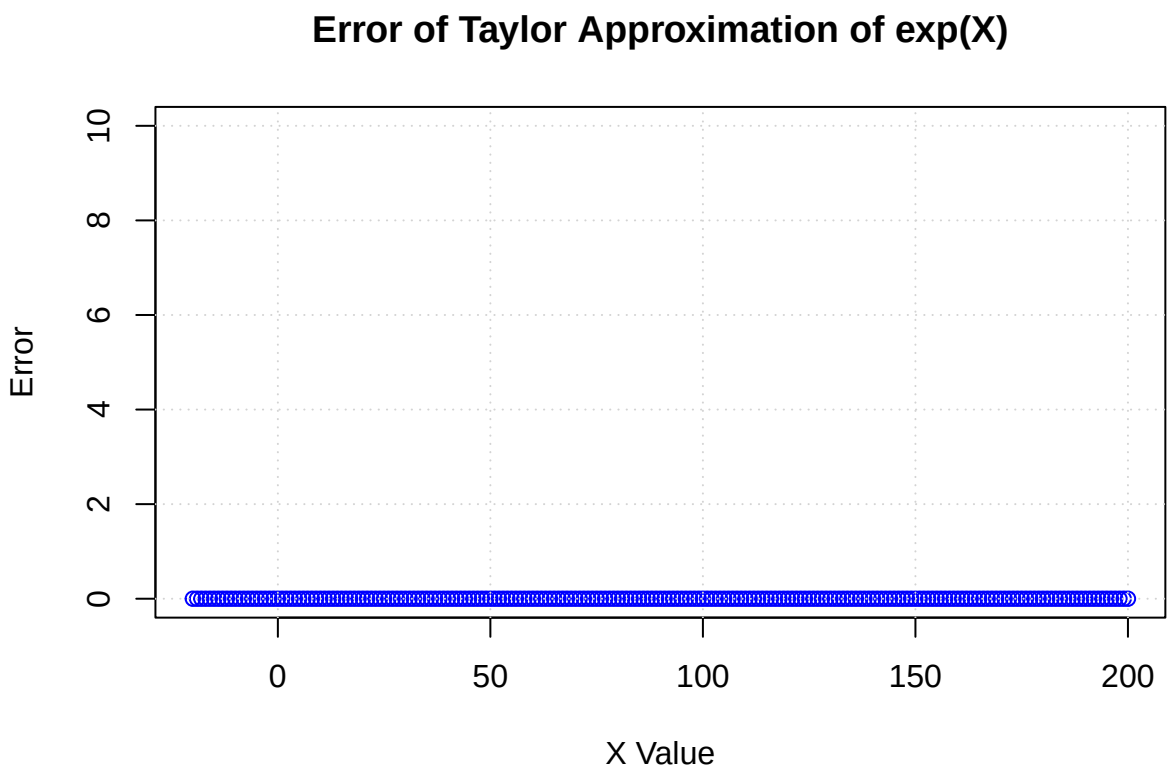
```
cat("R actual function", exp(X), "\n")
```

```
## R actual function 7.30706e+43
```

And now for our range:

```
X_values <- seq(-20, 200, by=1)
errors <- numeric(length(X_values))

for (i in seq_along(X_values)) {
  X <- X_values[i]


 mape <- mean(abs((exp(X) - calculate_exp_2(X)) / exp(X))) * 100



  errors[i] <- mape
}

# plotting

plot(X_values, errors, type = "b", col = "blue",
```

```
     xlab = "X Value", ylab = "Error",
     main = "Error of Taylor Approximation of exp(X)",
     ylim = c(0, 10))  # Adjust these limits as needed for your data
grid()
```



All looks good!

Exercise 4

```
function_x <- function(x) {
  return((sqrt(1 + x) - 1) / x)
}

function_x_rationalisation <- function(x) {
  return(1 / (sqrt(1 + x) + 1))
}

# Determine the values of R for x =1, 10^-4, 10^-32

function_x(1)
```

```
## [1] 0.4142136
```

```
function_x(10^(-4))
```

```
## [1] 0.4999875
```

```
function_x(10^(-32))
```

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

```
# The Taylor series expansion at 0 is

taylor_expansion_at_zero <- function(y) {
  return(1 + y / 2 - y^2 / 8 + y^3 / 16 - 5 * y^4 / 128 + 7 * y^5 / 256)
}

approximation_result_function_x_at_zero <- function(x) {
  return((taylor_expansion_at_zero(x) - 1) / x)
}

approximation_result_function_x_at_zero(1)
```

```
## [1] 0.4257812
```

```
approximation_result_function_x_at_zero(10^(-4))
```

```
## [1] 0.4999875
```

```
approximation_result_function_x_at_zero(10^(-32))
```

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

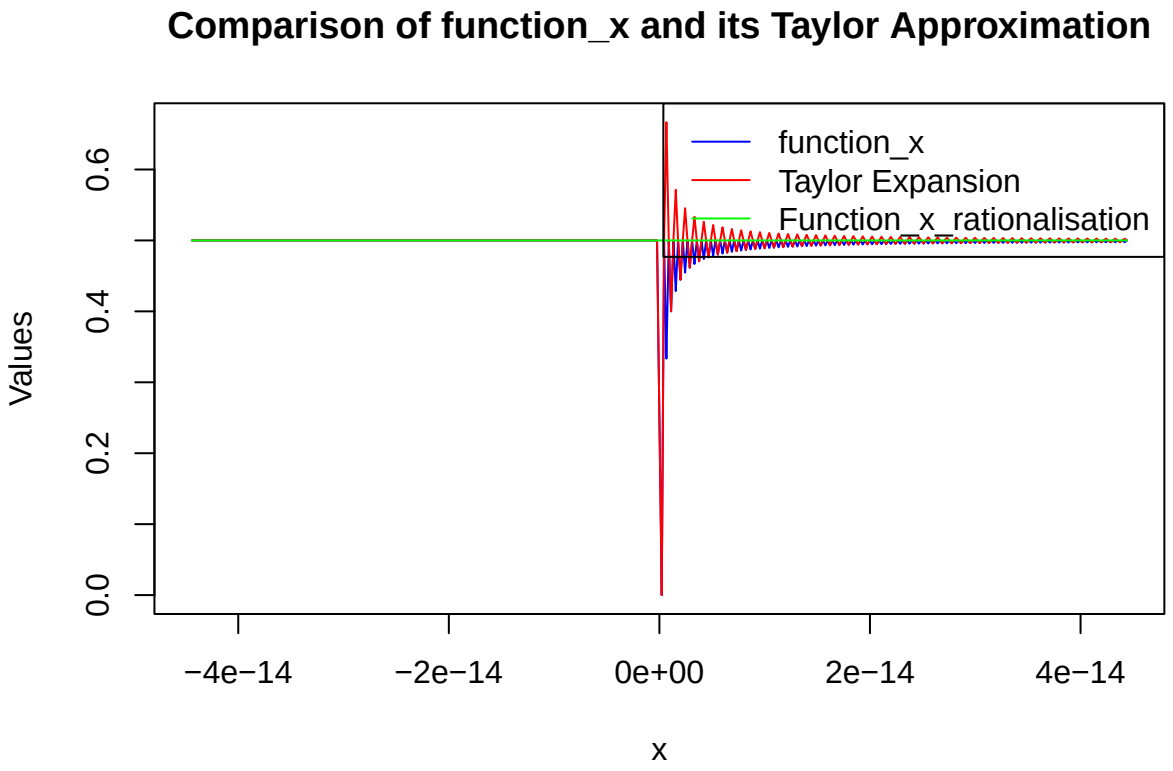
```
# Create a loop that stores the results of a multiplication in a vector
multiplication_results_negative <- numeric(100)
multiplication_results_positive <- numeric(100)

for (i in seq_len(200)) {
  multiplication_results_negative[i] <- .Machine$double.eps*i + 0
  multiplication_results_positive[i] <- -.Machine$double.eps*i + 0
}

combined_results <- c(multiplication_results_negative, multiplication_results_positive)
sorted_results <- sort(combined_results)

print(sorted_results)
```

```
##   [1] -4.440892e-14 -4.418688e-14 -4.396483e-14 -4.374279e-14 -4.352074e-14
##   [6] -4.329870e-14 -4.307665e-14 -4.285461e-14 -4.263256e-14 -4.241052e-14
##  [11] -4.218847e-14 -4.196643e-14 -4.174439e-14 -4.152234e-14 -4.130030e-14
##  [16] -4.107825e-14 -4.085621e-14 -4.063416e-14 -4.041212e-14 -4.019007e-14
##  [21] -3.996803e-14 -3.974598e-14 -3.952394e-14 -3.930190e-14 -3.907985e-14
##  [26] -3.885781e-14 -3.863576e-14 -3.841372e-14 -3.819167e-14 -3.796963e-14
##  [31] -3.774758e-14 -3.752554e-14 -3.730349e-14 -3.708145e-14 -3.685940e-14
##  [36] -3.663736e-14 -3.641532e-14 -3.619327e-14 -3.597123e-14 -3.574918e-14
##  [41] -3.552714e-14 -3.530509e-14 -3.508305e-14 -3.486100e-14 -3.463896e-14
##  [46] -3.441691e-14 -3.419487e-14 -3.397282e-14 -3.375078e-14 -3.352874e-14
##  [51] -3.330669e-14 -3.308465e-14 -3.286260e-14 -3.264056e-14 -3.241851e-14
##  [56] -3.219647e-14 -3.197442e-14 -3.175238e-14 -3.153033e-14 -3.130829e-14
##  [61] -3.108624e-14 -3.086420e-14 -3.064216e-14 -3.042011e-14 -3.019807e-14
##  [66] -2.997602e-14 -2.975398e-14 -2.953193e-14 -2.930989e-14 -2.908784e-14
##  [71] -2.886580e-14 -2.864375e-14 -2.842171e-14 -2.819966e-14 -2.797762e-14
##  [76] -2.775558e-14 -2.753353e-14 -2.731149e-14 -2.708944e-14 -2.686740e-14
##  [81] -2.664535e-14 -2.642331e-14 -2.620126e-14 -2.597922e-14 -2.575717e-14
##  [86] -2.553513e-14 -2.531308e-14 -2.509104e-14 -2.486900e-14 -2.464695e-14
##  [91] -2.442491e-14 -2.420286e-14 -2.398082e-14 -2.375877e-14 -2.353673e-14
```

```
##  [96] -2.331468e-14 -2.309264e-14 -2.287059e-14 -2.264855e-14 -2.242651e-14
## [101] -2.220446e-14 -2.198242e-14 -2.176037e-14 -2.153833e-14 -2.131628e-14
## [106] -2.109424e-14 -2.087219e-14 -2.065015e-14 -2.042810e-14 -2.020606e-14
## [111] -1.998401e-14 -1.976197e-14 -1.953993e-14 -1.931788e-14 -1.909584e-14
## [116] -1.887379e-14 -1.865175e-14 -1.842970e-14 -1.820766e-14 -1.798561e-14
## [121] -1.776357e-14 -1.754152e-14 -1.731948e-14 -1.709743e-14 -1.687539e-14
## [126] -1.665335e-14 -1.643130e-14 -1.620926e-14 -1.598721e-14 -1.576517e-14
## [131] -1.554312e-14 -1.532108e-14 -1.509903e-14 -1.487699e-14 -1.465494e-14
## [136] -1.443290e-14 -1.421085e-14 -1.398881e-14 -1.376677e-14 -1.354472e-14
## [141] -1.332268e-14 -1.310063e-14 -1.287859e-14 -1.265654e-14 -1.243450e-14
## [146] -1.221245e-14 -1.199041e-14 -1.176836e-14 -1.154632e-14 -1.132427e-14
## [151] -1.110223e-14 -1.088019e-14 -1.065814e-14 -1.043610e-14 -1.021405e-14
## [156] -9.992007e-15 -9.769963e-15 -9.547918e-15 -9.325873e-15 -9.103829e-15
## [161] -8.881784e-15 -8.659740e-15 -8.437695e-15 -8.215650e-15 -7.993606e-15
## [166] -7.771561e-15 -7.549517e-15 -7.327472e-15 -7.105427e-15 -6.883383e-15
## [171] -6.661338e-15 -6.439294e-15 -6.217249e-15 -5.995204e-15 -5.773160e-15
## [176] -5.551115e-15 -5.329071e-15 -5.107026e-15 -4.884981e-15 -4.662937e-15
## [181] -4.440892e-15 -4.218847e-15 -3.996803e-15 -3.774758e-15 -3.552714e-15
## [186] -3.330669e-15 -3.108624e-15 -2.886580e-15 -2.664535e-15 -2.442491e-15
## [191] -2.220446e-15 -1.998401e-15 -1.776357e-15 -1.554312e-15 -1.332268e-15
## [196] -1.110223e-15 -8.881784e-16 -6.661338e-16 -4.440892e-16 -2.220446e-16
## [201]  2.220446e-16  4.440892e-16  6.661338e-16  8.881784e-16  1.110223e-15
## [206]  1.332268e-15  1.554312e-15  1.776357e-15  1.998401e-15  2.220446e-15
## [211]  2.442491e-15  2.664535e-15  2.886580e-15  3.108624e-15  3.330669e-15
## [216]  3.552714e-15  3.774758e-15  3.996803e-15  4.218847e-15  4.440892e-15
## [221]  4.662937e-15  4.884981e-15  5.107026e-15  5.329071e-15  5.551115e-15
## [226]  5.773160e-15  5.995204e-15  6.217249e-15  6.439294e-15  6.661338e-15
## [231]  6.883383e-15  7.105427e-15  7.327472e-15  7.549517e-15  7.771561e-15
## [236]  7.993606e-15  8.215650e-15  8.437695e-15  8.659740e-15  8.881784e-15
## [241]  9.103829e-15  9.325873e-15  9.547918e-15  9.769963e-15  9.992007e-15
## [246]  1.021405e-14  1.043610e-14  1.065814e-14  1.088019e-14  1.110223e-14
## [251]  1.132427e-14  1.154632e-14  1.176836e-14  1.199041e-14  1.221245e-14
## [256]  1.243450e-14  1.265654e-14  1.287859e-14  1.310063e-14  1.332268e-14
## [261]  1.354472e-14  1.376677e-14  1.398881e-14  1.421085e-14  1.443290e-14
## [266]  1.465494e-14  1.487699e-14  1.509903e-14  1.532108e-14  1.554312e-14
## [271]  1.576517e-14  1.598721e-14  1.620926e-14  1.643130e-14  1.665335e-14
## [276]  1.687539e-14  1.709743e-14  1.731948e-14  1.754152e-14  1.776357e-14
## [281]  1.798561e-14  1.820766e-14  1.842970e-14  1.865175e-14  1.887379e-14
## [286]  1.909584e-14  1.931788e-14  1.953993e-14  1.976197e-14  1.998401e-14
## [291]  2.020606e-14  2.042810e-14  2.065015e-14  2.087219e-14  2.109424e-14
## [296]  2.131628e-14  2.153833e-14  2.176037e-14  2.198242e-14  2.220446e-14
## [301]  2.242651e-14  2.264855e-14  2.287059e-14  2.309264e-14  2.331468e-14
## [306]  2.353673e-14  2.375877e-14  2.398082e-14  2.420286e-14  2.442491e-14
## [311]  2.464695e-14  2.486900e-14  2.509104e-14  2.531308e-14  2.553513e-14
## [316]  2.575717e-14  2.597922e-14  2.620126e-14  2.642331e-14  2.664535e-14
## [321]  2.686740e-14  2.708944e-14  2.731149e-14  2.753353e-14  2.775558e-14
## [326]  2.797762e-14  2.819966e-14  2.842171e-14  2.864375e-14  2.886580e-14
## [331]  2.908784e-14  2.930989e-14  2.953193e-14  2.975398e-14  2.997602e-14
## [336]  3.019807e-14  3.042011e-14  3.064216e-14  3.086420e-14  3.108624e-14
## [341]  3.130829e-14  3.153033e-14  3.175238e-14  3.197442e-14  3.219647e-14
## [346]  3.241851e-14  3.264056e-14  3.286260e-14  3.308465e-14  3.330669e-14
## [351]  3.352874e-14  3.375078e-14  3.397282e-14  3.419487e-14  3.441691e-14
## [356]  3.463896e-14  3.486100e-14  3.508305e-14  3.530509e-14  3.552714e-14
## [361]  3.574918e-14  3.597123e-14  3.619327e-14  3.641532e-14  3.663736e-14
```

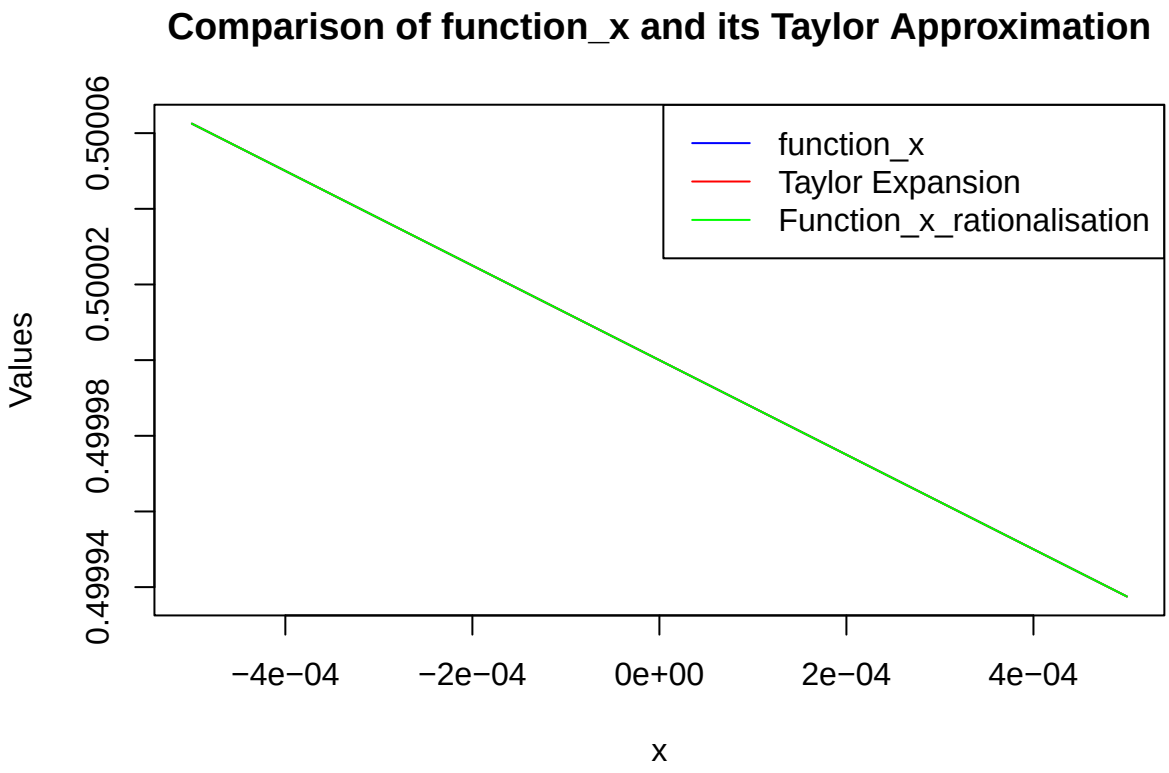
```
## [366]  3.685940e-14  3.708145e-14  3.730349e-14  3.752554e-14  3.774758e-14
## [371]  3.796963e-14  3.819167e-14  3.841372e-14  3.863576e-14  3.885781e-14
## [376]  3.907985e-14  3.930190e-14  3.952394e-14  3.974598e-14  3.996803e-14
## [381]  4.019007e-14  4.041212e-14  4.063416e-14  4.085621e-14  4.107825e-14
## [386]  4.130030e-14  4.152234e-14  4.174439e-14  4.196643e-14  4.218847e-14
## [391]  4.241052e-14  4.263256e-14  4.285461e-14  4.307665e-14  4.329870e-14
## [396]  4.352074e-14  4.374279e-14  4.396483e-14  4.418688e-14  4.440892e-14
```

```r
function_x_values <- sapply(sorted_results, function_x)
approximation_values <- sapply(sorted_results, approximation_result_function_x_at_zero)
function_x_rationalisation_values <- sapply(sorted_results, function_x_rationalisation)

plot(sorted_results, function_x_values, type = "l", col = "blue", ylim = range(c(function_x_values, appr
lines(sorted_results, approximation_values, col = "red")
lines(sorted_results, function_x_rationalisation_values, col = "green")
legend("topright", legend = c("function_x", "Taylor Expansion", "Function_x_rationalisation"), col = c(
```




```r
# With greater values of around o
increment_values <- seq(-0.0005, 0.0005, length.out = 1000)

# Calculate function values for the new vector
function_x_increment_values <- sapply(increment_values, function_x)
approximation_increment_values <- sapply(increment_values, approximation_result_function_x_at_zero)
function_x_rationalisation_increment_values <- sapply(increment_values, function_x_rationalisation)

# Plot the new values
plot(increment_values, function_x_increment_values, type = "l", col = "blue", ylim = range(c(function_x_
```

```
lines(increment_values, approximation_increment_values, col = "red")
lines(increment_values, function_x_rationalisation_increment_values, col = "green")
legend("topright", legend = c("function_x", "Taylor Expansion", "Function_x_rationalisation"), col = c(
```




```
print(function_x_increment_values[1])
```

```
## [1] 0.5000625
```

```
print(approximation_increment_values[1])
```

```
## [1] 0.5000625
```

```
print(function_x_rationalisation_increment_values[1])
```

```
## [1] 0.5000625
```

As shown in the first graph, the divergencews between the values of the taylor approximation, and the function in its two forms differ when the values of zero are very small of the order of .Machine$double.eps that represents the minimal differences tored in doubles so that 1 + .Machine$double.eps is recognized as different than 1. According to what we see in class we can think on two possible answers similar to computing the mean either through the sum of squares of deviations or as the differences between the mean of the squared values and the squared value of the mean, it may be that for values that R uses different process to solve (sqrt) and polynomial fractions that yield different results for very small values of x. It can be also that the taylor approximation is not accurate enough for very small values of x.

- Exercise 5

```
data("KNex", package = "Matrix")
X <- as.matrix(KNex$mm)
```

```r
Y <- KNex$y

# Not accurate type of execution.
start_time <- Sys.time()
fit <- lm.fit(X, Y)
end_time <- Sys.time()
execution_time_lmfit <- end_time - start_time
print(paste("Execution time:", execution_time_lmfit ))
```

```
## [1] "Execution time: 0.527944087982178"
```

```r
summary(fit)
```

```
##                Length Class  Mode
## coefficients    712   -none- numeric
## residuals      1850   -none- numeric
## effects        1850   -none- numeric
## rank              1   -none- numeric
## fitted.values  1850   -none- numeric
## assign            0   -none- NULL
## qr                5   qr     list
## df.residual       1   -none- numeric
```

```r
print(fit$coefficients)
```

```
##            x1            x2            x3            x4            x5
##  8.233613e+02  3.401156e+02  4.729760e+02  3.493175e+02  1.875595e+02
##            x6            x7            x8            x9           x10
##  1.590518e+02 -5.488358e+01  4.976512e+02  5.747553e+02  5.844035e+02
##           x11           x12           x13           x14           x15
##  4.433759e+02  4.598322e+02  4.377678e+02  4.718594e+02  2.518818e+02
##           x16           x17           x18           x19           x20
##  3.870411e+02  3.069063e+02  4.240379e+02  2.789121e+02  2.792760e+02
##           x21           x22           x23           x24           x25
##  1.343340e+02  1.751695e+02  1.964427e+02  1.840951e+02  1.475290e+02
##           x26           x27           x28           x29           x30
##  1.251323e+02  2.065104e+02  2.069963e+02  2.292703e+02  1.559251e+02
##           x31           x32           x33           x34           x35
##  1.519143e+02  7.014589e+01  3.905758e+01  8.530887e+01  1.543696e+02
##           x36           x37           x38           x39           x40
##  2.057099e+02  2.319248e+02  9.776988e+02  7.925844e+02  8.881473e+02
##           x41           x42           x43           x44           x45
##  2.353893e+02  4.792704e+02  3.281097e+02  4.429996e+02  2.186452e+02
##           x46           x47           x48           x49           x50
##  1.563899e+02  2.958011e+02  3.844060e+02  4.234291e+02  5.509931e+02
##           x51           x52           x53           x54           x55
##  6.758963e+02  8.035716e+02  6.621915e+02  6.011474e+02  3.680682e+02
##           x56           x57           x58           x59           x60
##  5.663948e+02  3.949569e+02  2.959356e+02  4.336161e+02  3.104814e+02
##           x61           x62           x63           x64           x65
##  2.953505e+02  4.065316e+02  3.047238e+02  3.895511e+02  3.644747e+02
##           x66           x67           x68           x69           x70
##  4.792018e+02  4.739014e+02  3.893197e+02  4.114851e+02  3.262520e+02
##           x71           x72           x73           x74           x75
##  3.521532e+02  2.109392e+02  4.867653e+01  2.854132e+02  1.665921e+02
```

```
##          x76          x77          x78          x79          x80
##  2.157594e+02 2.942907e+02 1.275330e+03 4.377361e+02 1.141416e+03
##          x81          x82          x83          x84          x85
##  6.743610e+02 5.796302e+02 8.075738e+02 3.486425e+02 3.109191e+02
##          x86          x87          x88          x89          x90
##  3.235872e+02 3.140781e+02 2.645482e+02 2.515995e+02 2.668293e+01
##          x91          x92          x93          x94          x95
##  1.040061e+02 4.310320e+02 6.200368e+02 6.680256e+02 6.720507e+02
##          x96          x97          x98          x99         x100
##  2.009096e+02 1.034493e+02 1.984021e+02 1.733915e+02 1.181538e+02
##         x101         x102         x103         x104         x105
##  2.292999e+02 1.520113e+02 1.810291e+02 8.822708e+01 7.352156e+01
##         x106         x107         x108         x109         x110
##  7.937319e+02 9.772659e+02 6.511196e+02 5.466706e+02 4.611334e+02
##         x111         x112         x113         x114         x115
##  3.752323e+02 6.064726e+02 1.147064e+03 8.704502e+02 7.476932e+02
##         x116         x117         x118         x119         x120
##  1.567143e+03 1.109384e+03 1.234439e+03 3.679044e+02 1.602590e+03
##         x121         x122         x123         x124         x125
##  6.098344e+02 1.056413e+03 1.026361e+03 1.153084e+03 9.729797e+02
##         x126         x127         x128         x129         x130
##  8.503641e+02 1.177652e+03 1.540202e+03 4.386716e+02 9.095268e+02
##         x131         x132         x133         x134         x135
##  5.232499e+02 4.384315e+02 5.248368e+02 3.540961e+02 9.983145e+01
##         x136         x137         x138         x139         x140
##  1.414522e+02 3.041497e+02 4.789437e+02 7.907425e+02 1.280902e+03
##         x141         x142         x143         x144         x145
##  7.796757e+02 1.186552e+03 9.128162e+02 5.312751e+02 8.118134e+02
##         x146         x147         x148         x149         x150
##  4.574875e+02 5.215645e+02 1.584941e+03 1.316831e+03 9.139825e+02
##         x151         x152         x153         x154         x155
##  1.216854e+03 4.147569e+02 1.186420e+03 7.411721e+02 1.027085e+03
##         x156         x157         x158         x159         x160
##  1.075455e+03 5.817779e+02 5.458053e+02 1.212961e+03 1.540449e+03
##         x161         x162         x163         x164         x165
##  1.236405e+03 1.851630e+03 9.306336e+02 1.372048e+03 1.362646e+03
##         x166         x167         x168         x169         x170
##  1.813367e+03 1.286375e+03 1.234847e+03 1.149278e+03 8.808191e+02
##         x171         x172         x173         x174         x175
##  5.961477e+02 1.694015e+03 1.596274e+03 1.444483e+03 2.077174e+03
##         x176         x177         x178         x179         x180
##  1.324911e+03 1.260746e+03 2.005284e+03 9.712938e+02 8.840197e+02
##         x181         x182         x183         x184         x185
##  1.402016e+03 1.214940e+03 1.240159e+03 9.550797e+02 1.063108e+03
##         x186         x187         x188         x189         x190
##  1.199806e+03 1.308148e+03 5.451265e+02 1.181646e+03 6.112288e+02
##         x191         x192         x193         x194         x195
##  1.264504e+03 1.603506e+03 5.882915e+02 1.501174e+03 1.119449e+03
##         x196         x197         x198         x199         x200
##  4.965155e+02 1.197639e+03 1.606044e+03 1.548944e+03 1.697874e+03
##         x201         x202         x203         x204         x205
##  1.882957e+03 1.071097e+03 8.124339e+02 1.149240e+03 9.166338e+02
##         x206         x207         x208         x209         x210
##  1.336850e+03 1.205886e+03 1.723131e+03 1.363899e+03 1.328714e+03
```

```
##          x211          x212          x213          x214          x215
##  1.010324e+03  3.484667e+02  9.050154e+02  9.050259e+02  4.532011e+02
##          x216          x217          x218          x219          x220
##  1.222204e+02  1.273822e+03  5.744192e+02  6.686459e+01  1.649687e+02
##          x221          x222          x223          x224          x225
##  2.399577e+02  7.774156e+01  7.418588e+02  6.931921e+02  6.202376e+02
##          x226          x227          x228          x229          x230
##  4.110494e+02  4.989011e+02  6.085244e+02  5.216533e+02  7.558672e+02
##          x231          x232          x233          x234          x235
##  6.974754e+02  9.130571e+02  8.533190e+02  9.546704e+02  1.003512e+03
##          x236          x237          x238          x239          x240
##  3.718109e+02  5.194766e+02  6.615869e+02  5.448740e+02  6.171181e+02
##          x241          x242          x243          x244          x245
##  5.679168e+02  6.594026e+02  5.743472e+02  6.086175e+02  6.415078e+02
##          x246          x247          x248          x249          x250
##  6.260161e+02  4.611107e+02  4.867161e+02  4.907317e+02  5.547498e+02
##          x251          x252          x253          x254          x255
##  4.522325e+02  2.406329e+02  7.264263e+02  1.087639e+03 -5.977955e+02
##          x256          x257          x258          x259          x260
##  3.021057e+02  1.141091e+02 -3.285763e+02 -2.645895e+02 -9.350325e+01
##          x261          x262          x263          x264          x265
##  8.480110e+01 -4.075880e+02 -4.906305e+02 -2.346200e+02 -3.516962e+02
##          x266          x267          x268          x269          x270
## -2.620967e+02 -2.183761e+02 -2.931662e+02 -1.012603e+02 -1.418004e+02
##          x271          x272          x273          x274          x275
## -6.184407e+01  5.222349e+01 -6.411881e+00 -9.958820e+01 -8.468244e+01
##          x276          x277          x278          x279          x280
## -2.985685e+01  2.176049e+00  2.432122e+00  5.447453e+01 -6.728951e+02
##          x281          x282          x283          x284          x285
## -3.930567e+02 -2.133295e+02 -2.611774e+02 -1.294373e+02 -1.451921e+02
##          x286          x287          x288          x289          x290
## -1.728689e+02 -3.641853e+02 -3.458473e+02 -3.454447e+02 -5.180689e+02
##          x291          x292          x293          x294          x295
## -3.862287e+02 -3.861784e+02 -4.574517e+02 -5.558451e+02 -4.695612e+02
##          x296          x297          x298          x299          x300
## -5.180875e+02 -4.881248e+02 -1.068728e+02 -1.921435e+01 -6.810688e+01
##          x301          x302          x303          x304          x305
## -1.374622e+02 -2.399132e+02 -7.669983e+02 -7.796899e+02 -3.011046e+02
##          x306          x307          x308          x309          x310
## -3.115064e+02 -1.894849e+02 -1.159407e+02  1.281760e+01  1.096485e+01
##          x311          x312          x313          x314          x315
##  7.315653e+01 -1.726737e+01 -1.035809e+02 -2.638569e+02 -2.643346e+02
##          x316          x317          x318          x319          x320
## -4.596594e+01 -3.479583e+01 -2.596215e+01 -1.479574e+02 -6.500528e+02
##          x321          x322          x323          x324          x325
## -4.910204e+02 -4.220341e+02 -4.129418e+02 -3.376096e+02 -1.413364e+02
##          x326          x327          x328          x329          x330
## -4.620433e+02 -3.772590e+02 -5.899099e+02 -7.454992e+02 -4.560824e+02
##          x331          x332          x333          x334          x335
## -5.890350e+02 -5.893740e+02 -3.980780e+02 -5.141775e+02 -5.993717e+02
##          x336          x337          x338          x339          x340
## -1.028976e+03 -1.139867e+03 -6.884250e+02 -3.896642e+02 -5.064374e+02
##          x341          x342          x343          x344          x345
## -3.033843e+02 -2.818138e+02 -9.205401e+01 -2.282751e+02 -8.895736e+01
```

```
##          x346          x347          x348          x349          x350
## -1.619924e+02 -2.307575e+02 -3.658412e+02 -2.333038e+02 -3.596782e+02
##          x351          x352          x353          x354          x355
## -2.922361e+02 -3.922122e+02 -6.113374e+02 -8.086072e+02 -5.711833e+02
##          x356          x357          x358          x359          x360
## -6.494190e+02 -9.423636e+02 -5.152641e+02 -8.230401e+02 -6.839862e+02
##          x361          x362          x363          x364          x365
## -7.121826e+02 -1.325016e+03 -7.963237e+02 -1.125432e+03 -1.101174e+03
##          x366          x367          x368          x369          x370
## -1.766877e+03 -8.479484e+02 -1.125654e+03 -5.945565e+02 -1.108627e+03
##          x371          x372          x373          x374          x375
## -1.108119e+03 -1.360395e+03 -7.511231e+02 -1.225595e+03 -7.566798e+02
##          x376          x377          x378          x379          x380
## -5.474398e+02 -5.653506e+02 -7.081127e+02 -5.923429e+02 -7.931433e+02
##          x381          x382          x383          x384          x385
## -1.012019e+03 -7.623270e+02 -9.025367e+02 -7.947243e+02 -1.210459e+03
##          x386          x387          x388          x389          x390
## -1.054689e+03 -1.763709e+03 -1.049938e+03 -8.759395e+02 -1.174869e+03
##          x391          x392          x393          x394          x395
## -9.273201e+02 -1.385203e+03 -1.093279e+03 -1.112236e+03 -1.150613e+03
##          x396          x397          x398          x399          x400
## -8.561477e+02 -9.969185e+02 -1.123539e+03 -1.181178e+03 -1.298406e+03
##          x401          x402          x403          x404          x405
## -1.741586e+03 -1.005348e+03 -1.230573e+03 -9.531796e+02 -4.945826e+02
##          x406          x407          x408          x409          x410
## -2.747093e+02 -6.867291e+02 -1.412108e+03 -5.811155e+02 -9.800365e+02
##          x411          x412          x413          x414          x415
## -4.883521e+02 -9.262726e+02 -7.239984e+02 -7.241480e+02 -8.507926e+02
##          x416          x417          x418          x419          x420
## -8.218580e+02 -1.446556e+03 -5.960024e+02 -1.023360e+03 -1.071175e+03
##          x421          x422          x423          x424          x425
## -7.289670e+02 -8.630471e+02 -1.169177e+03 -7.313966e+02 -6.988285e+02
##          x426          x427          x428          x429          x430
## -1.572890e+03 -1.471206e+03  4.809613e-02  9.281608e-02  2.396363e-02
##          x431          x432          x433          x434          x435
## -1.575965e-01 -7.802374e-02  9.321093e-02 -3.558827e-02  2.103348e-02
##          x436          x437          x438          x439          x440
## -6.825359e-03 -2.042342e-02 -1.769943e-02 -9.955312e-03  5.246432e-02
##          x441          x442          x443          x444          x445
## -5.147767e-02  8.215849e-03 -4.134768e-02  3.610998e-02  8.504226e-02
##          x446          x447          x448          x449          x450
##  2.163557e-02 -1.058678e-01  4.350869e-03  2.226391e-01 -2.522468e-03
##          x451          x452          x453          x454          x455
## -4.095716e-02  3.475400e-02  1.005511e-01  3.660109e-03 -1.712364e-02
##          x456          x457          x458          x459          x460
##  1.766482e-02 -6.475812e-03  1.520783e-02 -9.332949e-02 -1.183689e-02
##          x461          x462          x463          x464          x465
## -7.010710e-02 -2.387681e-01 -1.048032e-01  1.656755e-01  2.587727e-01
##          x466          x467          x468          x469          x470
##  1.542173e-03  2.386572e-02  1.586551e-01 -1.071491e-01 -5.617120e-02
##          x471          x472          x473          x474          x475
## -2.899340e-03  1.103520e-01  5.841165e-04 -1.269383e-01 -1.359136e-02
##          x476          x477          x478          x479          x480
## -2.052460e-01 -4.727969e-02 -1.940478e-02  1.070982e-01  9.360175e-02
```

```
##          x481          x482          x483          x484          x485
##  2.721813e-01  2.079780e-01  1.260904e-01  1.263730e-01  3.787899e-02
##          x486          x487          x488          x489          x490
## -4.923318e-02 -2.418453e-03  2.466606e-01 -8.667764e-02 -1.349602e-01
##          x491          x492          x493          x494          x495
##  3.507163e-02  1.838088e-01  2.740918e-02  7.079065e-01  1.173965e-01
##          x496          x497          x498          x499          x500
## -1.110851e-01  8.522975e-04 -4.450185e-02 -1.844570e-02  5.874862e-02
##          x501          x502          x503          x504          x505
##  4.519545e-01 -2.231175e-02  5.158078e-01  2.730600e-01  2.584096e-03
##          x506          x507          x508          x509          x510
##  2.544676e-01  8.096752e-02  1.655444e-02  1.074376e-02 -1.200081e-01
##          x511          x512          x513          x514          x515
##  6.674829e-02  6.689370e-02  2.932761e-01  3.664130e-02  8.501443e-02
##          x516          x517          x518          x519          x520
##  1.072941e-01 -1.066597e-01  1.241043e-02  4.322880e-01  4.055412e-02
##          x521          x522          x523          x524          x525
## -5.426553e-02 -2.937913e-01  1.219936e-02  8.333980e-02 -5.151411e-02
##          x526          x527          x528          x529          x530
##  6.182571e-02  1.178115e-01 -1.604041e-03 -1.664804e-01 -3.113249e-01
##          x531          x532          x533          x534          x535
##  4.759662e-02 -3.821369e-02 -3.080331e-02  3.011874e-03  1.319403e-02
##          x536          x537          x538          x539          x540
## -6.577987e-02  4.348857e-02  8.558820e-02 -3.576915e-01  2.090907e-02
##          x541          x542          x543          x544          x545
##  5.775185e-02 -1.219080e-01 -8.115377e-02 -3.532365e-02 -3.110196e-01
##          x546          x547          x548          x549          x550
## -2.505220e-01  4.719523e-01 -1.430548e-01 -4.010046e-01 -1.192994e-01
##          x551          x552          x553          x554          x555
## -3.684780e-01 -5.531258e-01 -3.167508e+00 -6.960263e-01 -1.106045e-01
##          x556          x557          x558          x559          x560
## -1.579831e-01 -3.144784e-02 -1.122970e-01 -6.677214e-02  1.697411e-02
##          x561          x562          x563          x564          x565
##  9.298568e-03 -2.775644e-01 -5.090505e-03  6.527704e-02  1.020965e-01
##          x566          x567          x568          x569          x570
## -1.184595e-02  1.040935e-01  7.467754e-02  4.092099e-01  3.680639e-02
##          x571          x572          x573          x574          x575
##  1.212436e-02 -3.380293e-02  5.752697e-02 -4.950190e-02 -1.077086e-01
##          x576          x577          x578          x579          x580
## -1.812482e-01 -3.295574e-01 -1.105570e-01  1.364813e-01 -6.823645e-02
##          x581          x582          x583          x584          x585
## -8.247208e-01 -4.494110e-01 -5.389062e-01 -1.704268e-01 -9.067183e-02
##          x586          x587          x588          x589          x590
##  1.088935e-02 -3.509323e-02  1.014286e-01  3.139771e-02 -6.786063e-02
##          x591          x592          x593          x594          x595
##  1.237607e-01  1.550431e-01  4.155808e-02 -7.719607e-02  6.440362e-02
##          x596          x597          x598          x599          x600
## -1.208133e-02  6.569299e-02  1.102057e-01  1.189859e-02  7.778211e-02
##          x601          x602          x603          x604          x605
## -1.052973e-01  3.547668e-02 -1.576971e-01 -1.603864e-01 -1.455670e-01
##          x606          x607          x608          x609          x610
## -2.466386e-01 -3.019117e-01  2.045033e-01  2.628887e-01  2.424364e-01
##          x611          x612          x613          x614          x615
##  1.856350e-01  2.131610e-02 -4.123019e-02  6.041438e-02  8.697726e-03
```

```
##          x616          x617          x618          x619          x620
##  5.624960e-02  1.007667e-01  1.334960e-01  2.460480e-01  9.080047e-02
##          x621          x622          x623          x624          x625
##  1.647035e-03 -1.164594e-02  7.394406e-02  5.827446e-02  1.516282e-01
##          x626          x627          x628          x629          x630
##  1.023448e-01  3.550852e-01  8.369020e-02  5.244882e-02  7.180437e-02
##          x631          x632          x633          x634          x635
##  1.122285e-01  2.621647e-01  5.721754e-02  7.065818e-03 -3.262964e-01
##          x636          x637          x638          x639          x640
##  2.543696e-02  6.529512e-02 -5.155784e-02  3.977290e-01 -2.601596e-02
##          x641          x642          x643          x644          x645
##  6.308663e-02  5.362685e-02 -2.363418e+00  2.552907e-01  5.612954e-02
##          x646          x647          x648          x649          x650
##  1.633922e-01  7.536636e-02  1.280213e-01  4.782869e-01  6.878834e-01
##          x651          x652          x653          x654          x655
## -1.023740e+00 -1.231953e+00 -1.780121e+02 -1.541142e+02 -7.055784e-01
##          x656          x657          x658          x659          x660
## -4.392741e-01  2.889989e-02  2.159282e-01  2.131080e-01  8.270576e-03
##          x661          x662          x663          x664          x665
##  1.587280e-02  3.280109e-02  2.725950e-03  1.175521e-01  3.682498e-03
##          x666          x667          x668          x669          x670
##  7.485723e-02 -1.264306e-01  1.307723e-01  8.924368e-02  5.418195e-02
##          x671          x672          x673          x674          x675
##  2.097400e-01  2.154820e-01  1.298764e-01  8.876714e-02  1.066478e-01
##          x676          x677          x678          x679          x680
##  1.554227e-01 -1.400014e-02 -5.357693e-02  4.317700e-02  6.949579e-02
##          x681          x682          x683          x684          x685
##  1.116628e-02  8.723762e-02  7.641936e-02  6.668964e-02  3.623257e-02
##          x686          x687          x688          x689          x690
##  3.210972e-01  8.339955e-02  2.308027e-01  2.882711e-01  2.940148e-01
##          x691          x692          x693          x694          x695
##  2.025230e-01  1.325692e-01 -2.476536e-02 -2.515312e-02  4.254121e-01
##          x696          x697          x698          x699          x700
##  3.294773e-01  1.440453e-01 -1.306660e+00 -3.210345e+00 -3.035992e+00
##          x701          x702          x703          x704          x705
## -1.679047e-01 -6.456320e-01  1.274116e-01 -1.604348e-02 -2.342478e-01
##          x706          x707          x708          x709          x710
## -1.510406e-01 -2.080114e+00 -6.439531e+00 -1.259875e-01 -1.191570e-01
##          x711          x712
## -2.060116e+00 -7.848831e+00
```

```r
# Formula of Least Square estimator B = (X^T X)^-1 X^T Y
# Let us try to find this (X^T X)^-1 part with

######################################################
### Cholesky decomposition
######################################################
# We find the product of the matrix X^TX (This matrix is symetric)
# Crossprod tells him just to do half of the work, by computing only the upper triangular part of the m



R <- chol(crossprod(X))
```

```r
# Using Solve to find the inverse of both matrix and to multiply the right upper triangular matrix

start_time <- Sys.time()
chol_betas <- backsolve(R, forwardsolve(R, crossprod(X, Y), upper.tri = TRUE, transpose = TRUE))
end_time <- Sys.time()
execution_time_chol <- end_time - start_time
print(paste("Execution time:", execution_time_chol))
```

```
## [1] "Execution time: 0.004302978515625"
```

```r
# It requires more time than the lm.fit function.


#####################################################
### QR decomposition
#####################################################
QR <- qr(X)
qr_betas <- backsolve(qr.R(QR), crossprod(qr.Q(QR), Y))
solve(QR, Y)
```

```
##   [1]  8.233613e+02  3.401156e+02  4.729760e+02  3.493175e+02  1.875595e+02
##   [6]  1.590518e+02 -5.488358e+01  4.976512e+02  5.747553e+02  5.844035e+02
##  [11]  4.433759e+02  4.598322e+02  4.377678e+02  4.718594e+02  2.518818e+02
##  [16]  3.870411e+02  3.069063e+02  4.240379e+02  2.789121e+02  2.792760e+02
##  [21]  1.343340e+02  1.751695e+02  1.964427e+02  1.840951e+02  1.475290e+02
##  [26]  1.251323e+02  2.065104e+02  2.069963e+02  2.292703e+02  1.559251e+02
##  [31]  1.519143e+02  7.014589e+01  3.905758e+01  8.530887e+01  1.543696e+02
##  [36]  2.057099e+02  2.319248e+02  9.776988e+02  7.925844e+02  8.881473e+02
##  [41]  2.353893e+02  4.792704e+02  3.281097e+02  4.429996e+02  2.186452e+02
##  [46]  1.563899e+02  2.958011e+02  3.844060e+02  4.234291e+02  5.509931e+02
##  [51]  6.758963e+02  8.035716e+02  6.621915e+02  6.011474e+02  3.680682e+02
##  [56]  5.663948e+02  3.949569e+02  2.959356e+02  4.336161e+02  3.104814e+02
##  [61]  2.953505e+02  4.065316e+02  3.047238e+02  3.895511e+02  3.644747e+02
##  [66]  4.792018e+02  4.739014e+02  3.893197e+02  4.114851e+02  3.262520e+02
##  [71]  3.521532e+02  2.109392e+02  4.867653e+01  2.854132e+02  1.665921e+02
##  [76]  2.157594e+02  2.942907e+02  1.275330e+03  4.377361e+02  1.141416e+03
##  [81]  6.743610e+02  5.796302e+02  8.075738e+02  3.486425e+02  3.109191e+02
##  [86]  3.235872e+02  3.140781e+02  2.645482e+02  2.515995e+02  2.668293e+01
##  [91]  1.040061e+02  4.310320e+02  6.200368e+02  6.680256e+02  6.720507e+02
##  [96]  2.009096e+02  1.034493e+02  1.984021e+02  1.733915e+02  1.181538e+02
## [101]  2.292999e+02  1.520113e+02  1.810291e+02  8.822708e+01  7.352156e+01
## [106]  7.937319e+02  9.772659e+02  6.511196e+02  5.466706e+02  4.611334e+02
## [111]  3.752323e+02  6.064726e+02  1.147064e+03  8.704502e+02  7.476932e+02
## [116]  1.567143e+03  1.109384e+03  1.234439e+03  3.679044e+02  1.602590e+03
## [121]  6.098344e+02  1.056413e+03  1.026361e+03  1.153084e+03  9.729797e+02
## [126]  8.503641e+02  1.177652e+03  1.540202e+03  4.386716e+02  9.095268e+02
## [131]  5.232499e+02  4.384315e+02  5.248368e+02  3.540961e+02  9.983145e+01
## [136]  1.414522e+02  3.041497e+02  4.789437e+02  7.907425e+02  1.280902e+03
## [141]  7.796757e+02  1.186552e+03  9.128162e+02  5.312751e+02  8.118134e+02
## [146]  4.574875e+02  5.215645e+02  1.584941e+03  1.316831e+03  9.139825e+02
## [151]  1.216854e+03  4.147569e+02  1.186420e+03  7.411721e+02  1.027085e+03
## [156]  1.075455e+03  5.817779e+02  5.458053e+02  1.212961e+03  1.540449e+03
## [161]  1.236405e+03  1.851630e+03  9.306336e+02  1.372048e+03  1.362646e+03
## [166]  1.813367e+03  1.286375e+03  1.234847e+03  1.149278e+03  8.808191e+02
## [171]  5.961477e+02  1.694015e+03  1.596274e+03  1.444483e+03  2.077174e+03
```

```
## [176]  1.324911e+03  1.260746e+03  2.005284e+03  9.712938e+02  8.840197e+02
## [181]  1.402016e+03  1.214940e+03  1.240159e+03  9.550797e+02  1.063108e+03
## [186]  1.199806e+03  1.308148e+03  5.451265e+02  1.181646e+03  6.112288e+02
## [191]  1.264504e+03  1.603506e+03  5.882915e+02  1.501174e+03  1.119449e+03
## [196]  4.965155e+02  1.197639e+03  1.606044e+03  1.548944e+03  1.697874e+03
## [201]  1.882957e+03  1.071097e+03  8.124339e+02  1.149240e+03  9.166338e+02
## [206]  1.336850e+03  1.205886e+03  1.723131e+03  1.363899e+03  1.328714e+03
## [211]  1.010324e+03  3.484667e+02  9.050154e+02  9.050259e+02  4.532011e+02
## [216]  1.222204e+02  1.273822e+03  5.744192e+02  6.686459e+01  1.649687e+02
## [221]  2.399577e+02  7.774156e+01  7.418588e+02  6.931921e+02  6.202376e+02
## [226]  4.110494e+02  4.989011e+02  6.085244e+02  5.216533e+02  7.558672e+02
## [231]  6.974754e+02  9.130571e+02  8.533190e+02  9.546704e+02  1.003512e+03
## [236]  3.718109e+02  5.194766e+02  6.615869e+02  5.448740e+02  6.171181e+02
## [241]  5.679168e+02  6.594026e+02  5.743472e+02  6.086175e+02  6.415078e+02
## [246]  6.260161e+02  4.611107e+02  4.867161e+02  4.907317e+02  5.547498e+02
## [251]  4.522325e+02  2.406329e+02  7.264263e+02  1.087639e+03 -5.977955e+02
## [256]  3.021057e+02  1.141091e+02 -3.285763e+02 -2.645895e+02 -9.350325e+01
## [261]  8.480110e+01 -4.075880e+02 -4.906305e+02 -2.346200e+02 -3.516962e+02
## [266] -2.620967e+02 -2.183761e+02 -2.931662e+02 -1.012603e+02 -1.418004e+02
## [271] -6.184407e+01  5.222349e+01 -6.411881e+00 -9.958820e+01 -8.468244e+01
## [276] -2.985685e+01  2.176049e+00  2.432122e+00  5.447453e+01 -6.728951e+02
## [281] -3.930567e+02 -2.133295e+02 -2.611774e+02 -1.294373e+02 -1.451921e+02
## [286] -1.728689e+02 -3.641853e+02 -3.458473e+02 -3.454447e+02 -5.180689e+02
## [291] -3.862287e+02 -3.861784e+02 -4.574517e+02 -5.558451e+02 -4.695612e+02
## [296] -5.180875e+02 -4.881248e+02 -1.068728e+02 -1.921435e+01 -6.810688e+01
## [301] -1.374622e+02 -2.399132e+02 -7.669983e+02 -7.796899e+02 -3.011046e+02
## [306] -3.115064e+02 -1.894849e+02 -1.159407e+02  1.281760e+01  1.096485e+01
## [311]  7.315653e+01 -1.726737e+01 -1.035809e+02 -2.638569e+02 -2.643346e+02
## [316] -4.596594e+01 -3.479583e+01 -2.596215e+01 -1.479574e+02 -6.500528e+02
## [321] -4.910204e+02 -4.220341e+02 -4.129418e+02 -3.376096e+02 -1.413364e+02
## [326] -4.620433e+02 -3.772590e+02 -5.899099e+02 -7.454992e+02 -4.560824e+02
## [331] -5.890350e+02 -5.893740e+02 -3.980780e+02 -5.141775e+02 -5.993717e+02
## [336] -1.028976e+03 -1.139867e+03 -6.884250e+02 -3.896642e+02 -5.064374e+02
## [341] -3.033843e+02 -2.818138e+02 -9.205401e+01 -2.282751e+02 -8.895736e+01
## [346] -1.619924e+02 -2.307575e+02 -3.658412e+02 -2.333038e+02 -3.596782e+02
## [351] -2.922361e+02 -3.922122e+02 -6.113374e+02 -8.086072e+02 -5.711833e+02
## [356] -6.494190e+02 -9.423636e+02 -5.152641e+02 -8.230401e+02 -6.839862e+02
## [361] -7.121826e+02 -1.325016e+03 -7.963237e+02 -1.125432e+03 -1.101174e+03
## [366] -1.766877e+03 -8.479484e+02 -1.125654e+03 -5.945565e+02 -1.108627e+03
## [371] -1.108119e+03 -1.360395e+03 -7.511231e+02 -1.225595e+03 -7.566798e+02
## [376] -5.474398e+02 -5.653506e+02 -7.081127e+02 -5.923429e+02 -7.931433e+02
## [381] -1.012019e+03 -7.623270e+02 -9.025367e+02 -7.947243e+02 -1.210459e+03
## [386] -1.054689e+03 -1.763709e+03 -1.049938e+03 -8.759395e+02 -1.174869e+03
## [391] -9.273201e+02 -1.385203e+03 -1.093279e+03 -1.112236e+03 -1.150613e+03
## [396] -8.561477e+02 -9.969185e+02 -1.123539e+03 -1.181178e+03 -1.298406e+03
## [401] -1.741586e+03 -1.005348e+03 -1.230573e+03 -9.531796e+02 -4.945826e+02
## [406] -2.747093e+02 -6.867291e+02 -1.412108e+03 -5.811155e+02 -9.800365e+02
## [411] -4.883521e+02 -9.262726e+02 -7.239984e+02 -7.241480e+02 -8.507926e+02
## [416] -8.218580e+02 -1.446556e+03 -5.960024e+02 -1.023360e+03 -1.071175e+03
## [421] -7.289670e+02 -8.630471e+02 -1.169177e+03 -7.313966e+02 -6.988285e+02
## [426] -1.572890e+03 -1.471206e+03  4.809613e-02  9.281608e-02  2.396363e-02
## [431] -1.575965e-01 -7.802374e-02  9.321093e-02 -3.558827e-02  2.103348e-02
## [436] -6.825359e-03 -2.042342e-02 -1.769943e-02 -9.955312e-03  5.246432e-02
## [441] -5.147767e-02  8.215849e-03 -4.134768e-02  3.610998e-02  8.504226e-02
```

```
## [446]  2.163557e-02 -1.058678e-01  4.350869e-03  2.226391e-01 -2.522468e-03
## [451] -4.095716e-02  3.475400e-02  1.005511e-01  3.660109e-03 -1.712364e-02
## [456]  1.766482e-02 -6.475812e-03  1.520783e-02 -9.332949e-02 -1.183689e-02
## [461] -7.010710e-02 -2.387681e-01 -1.048032e-01  1.656755e-01  2.587727e-01
## [466]  1.542173e-03  2.386572e-02  1.586551e-01 -1.071491e-01 -5.617120e-02
## [471] -2.899340e-03  1.103520e-01  5.841165e-04 -1.269383e-01 -1.359136e-02
## [476] -2.052460e-01 -4.727969e-02 -1.940478e-02  1.070982e-01  9.360175e-02
## [481]  2.721813e-01  2.079780e-01  1.260904e-01  1.263730e-01  3.787899e-02
## [486] -4.923318e-02 -2.418453e-03  2.466606e-01 -8.667764e-02 -1.349602e-01
## [491]  3.507163e-02  1.838088e-01  2.740918e-02  7.079065e-01  1.173965e-01
## [496] -1.110851e-01  8.522975e-04 -4.450185e-02 -1.844570e-02  5.874862e-02
## [501]  4.519545e-01 -2.231175e-02  5.158078e-01  2.730600e-01  2.584096e-03
## [506]  2.544676e-01  8.096752e-02  1.655444e-02  1.074376e-02 -1.200081e-01
## [511]  6.674829e-02  6.689370e-02  2.932761e-01  3.664130e-02  8.501443e-02
## [516]  1.072941e-01 -1.066597e-01  1.241043e-02  4.322880e-01  4.055412e-02
## [521] -5.426553e-02 -2.937913e-01  1.219936e-02  8.333980e-02 -5.151411e-02
## [526]  6.182571e-02  1.178115e-01 -1.604041e-03 -1.664804e-01 -3.113249e-01
## [531]  4.759662e-02 -3.821369e-02 -3.080331e-02  3.011874e-03  1.319403e-02
## [536] -6.577987e-02  4.348857e-02  8.558820e-02 -3.576915e-01  2.090907e-02
## [541]  5.775185e-02 -1.219080e-01 -8.115377e-02 -3.532365e-02 -3.110196e-01
## [546] -2.505220e-01  4.719523e-01 -1.430548e-01 -4.010046e-01 -1.192994e-01
## [551] -3.684780e-01 -5.531258e-01 -3.167508e+00 -6.960263e-01 -1.106045e-01
## [556] -1.579831e-01 -3.144784e-02 -1.122970e-01 -6.677214e-02  1.697411e-02
## [561]  9.298568e-03 -2.775644e-01 -5.090505e-03  6.527704e-02  1.020965e-01
## [566] -1.184595e-02  1.040935e-01  7.467754e-02  4.092099e-01  3.680639e-02
## [571]  1.212436e-02 -3.380293e-02  5.752697e-02 -4.950190e-02 -1.077086e-01
## [576] -1.812482e-01 -3.295574e-01 -1.105570e-01  1.364813e-01 -6.823645e-02
## [581] -8.247208e-01 -4.494110e-01 -5.389062e-01 -1.704268e-01 -9.067183e-02
## [586]  1.088935e-02 -3.509323e-02  1.014286e-01  3.139771e-02 -6.786063e-02
## [591]  1.237607e-01  1.550431e-01  4.155808e-02 -7.719607e-02  6.440362e-02
## [596] -1.208133e-02  6.569299e-02  1.102057e-01  1.189859e-02  7.778211e-02
## [601] -1.052973e-01  3.547668e-02 -1.576971e-01 -1.603864e-01 -1.455670e-01
## [606] -2.466386e-01 -3.019117e-01  2.045033e-01  2.628887e-01  2.424364e-01
## [611]  1.856350e-01  2.131610e-02 -4.123019e-02  6.041438e-02  8.697726e-03
## [616]  5.624960e-02  1.007667e-01  1.334960e-01  2.460480e-01  9.080047e-02
## [621]  1.647035e-03 -1.164594e-02  7.394406e-02  5.827446e-02  1.516282e-01
## [626]  1.023448e-01  3.550852e-01  8.369020e-02  5.244882e-02  7.180437e-02
## [631]  1.122285e-01  2.621647e-01  5.721754e-02  7.065818e-03 -3.262964e-01
## [636]  2.543696e-02  6.529512e-02 -5.155784e-02  3.977290e-01 -2.601596e-02
## [641]  6.308663e-02  5.362685e-02 -2.363418e+00  2.552907e-01  5.612954e-02
## [646]  1.633922e-01  7.536636e-02  1.280213e-01  4.782869e-01  6.878834e-01
## [651] -1.023740e+00 -1.231953e+00 -1.780121e+02 -1.541142e+02 -7.055784e-01
## [656] -4.392741e-01  2.889989e-02  2.159282e-01  2.131080e-01  8.270576e-03
## [661]  1.587280e-02  3.280109e-02  2.725950e-03  1.175521e-01  3.682498e-03
## [666]  7.485723e-02 -1.264306e-01  1.307723e-01  8.924368e-02  5.418195e-02
## [671]  2.097400e-01  2.154820e-01  1.298764e-01  8.876714e-02  1.066478e-01
## [676]  1.554227e-01 -1.400014e-02 -5.357693e-02  4.317700e-02  6.949579e-02
## [681]  1.116628e-02  8.723762e-02  7.641936e-02  6.668964e-02  3.623257e-02
## [686]  3.210972e-01  8.339955e-02  2.308027e-01  2.882711e-01  2.940148e-01
## [691]  2.025230e-01  1.325692e-01 -2.476536e-02 -2.515312e-02  4.254121e-01
## [696]  3.294773e-01  1.440453e-01 -1.306660e+00 -3.210345e+00 -3.035992e+00
## [701] -1.679047e-01 -6.456320e-01  1.274116e-01 -1.604348e-02 -2.342478e-01
## [706] -1.510406e-01 -2.080114e+00 -6.439531e+00 -1.259875e-01 -1.191570e-01
## [711] -2.060116e+00 -7.848831e+00
```

```
# The betas are identical, but the execution time is different.
fit$coefficients[2]
```

```
##       x2
## 340.1156
```

```
chol_betas[2]
```

```
## [1] 340.1156
```

```
qr_betas[2]
```

```
## [1] 340.1156
```

```
## 2.nd Part

object.size(X)
```

```
## 10537816 bytes
```

```
X_not_sparsed <- KNex$mm
object.size(X_not_sparsed)
```

```
## 109416 bytes
```

```
# The class is a dgCMatrix, which is a sparse matrix. Meaning that it
# saves some space by not saving the zeros.
class(KNex$mm)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

```
class(X)
```

```
## [1] "matrix" "array"
```