Report

Thijs Quast (thiqu264), Anubhav Dikshit (anudi287)

06-3-2019

Contents

Contributions

For the analytical part of Assignment 2, EM-algorithm, group10 helped us.

Question 1

1

```
function_x <- function(x){
  output <- (x^2)/(exp(x)) - (2*exp(-(9*sin(x))/(x^2 + x + 1)))
  return(output)
}
```

2

```
crossover <- function(x, y){
  kid <- (x+y)/2
  return(kid)
}
```

3

```
mutate <- function(x){
  output <- (x^2) %% 30
  return(output)
}
```

4

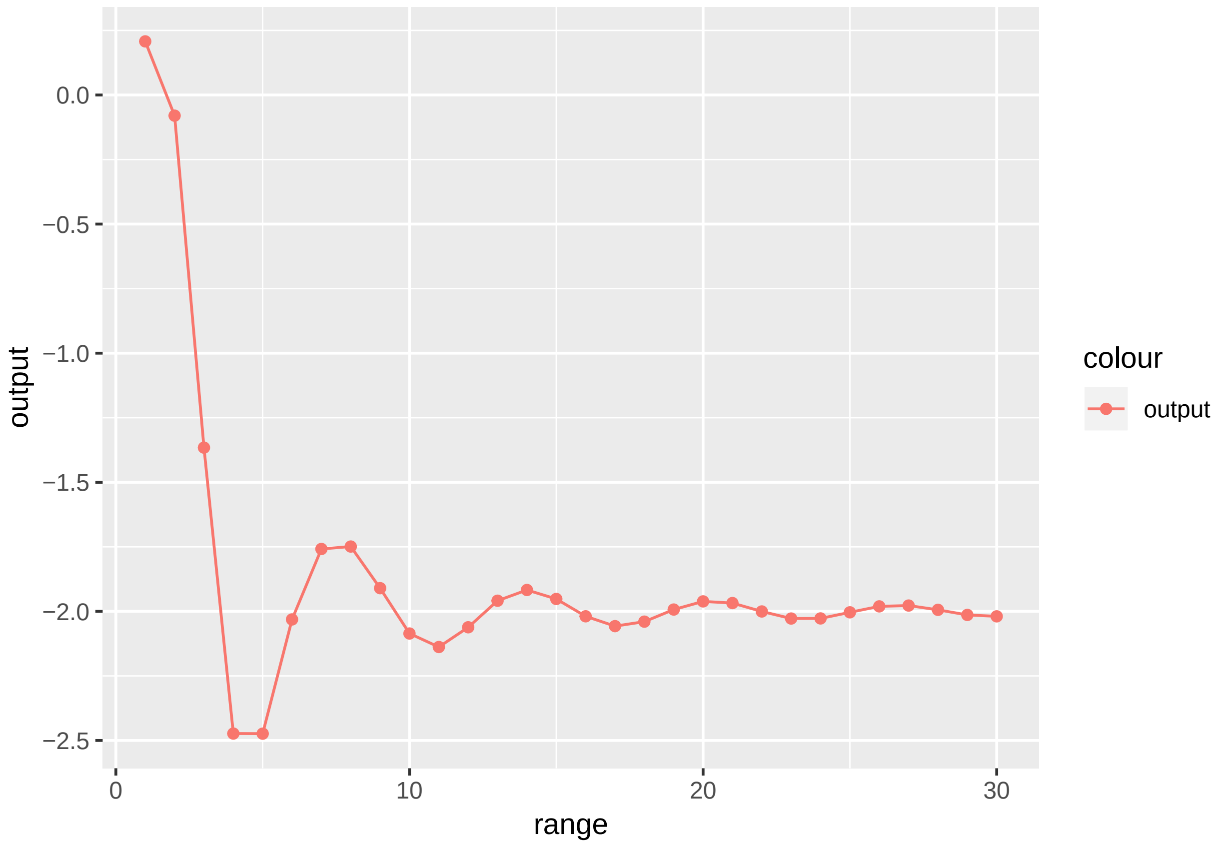
```
library(ggplot2)
```

```
f <- function(maxiter, mutprob){
  answer <- list()

  # a
  range <- seq(1, 30, by = 1)
  output <- c()

  for (i in 1:length(range)){
    output[i] <- function_x(range[i])
  }

  output <- as.data.frame(cbind(output, range))
  plot <- ggplot(output, aes(x = range, y = output, col = "output")) + geom_point() + geom_line()
  answer$Plot <- plot

  # b
  Values <- c()
  X <- seq(0, 30, by = 5)
  X_initial <- X

  # c
  for (i in 1:length(X)){
    Values[i] <- function_x(X[i])
  }

  answer$Values <- Values

  current_max <- 0

  # d

  for (i in 1:maxiter){

    parent1 <- sample(X, 1)
    parent2 <- sample(X, 1)

    victim_index <- which.min(Values)
    kid <- crossover(parent1, parent2)

    random_probability <- rbinom(1, 1, mutprob)

    if (random_probability == 1){
      kid <- mutate(kid)
```

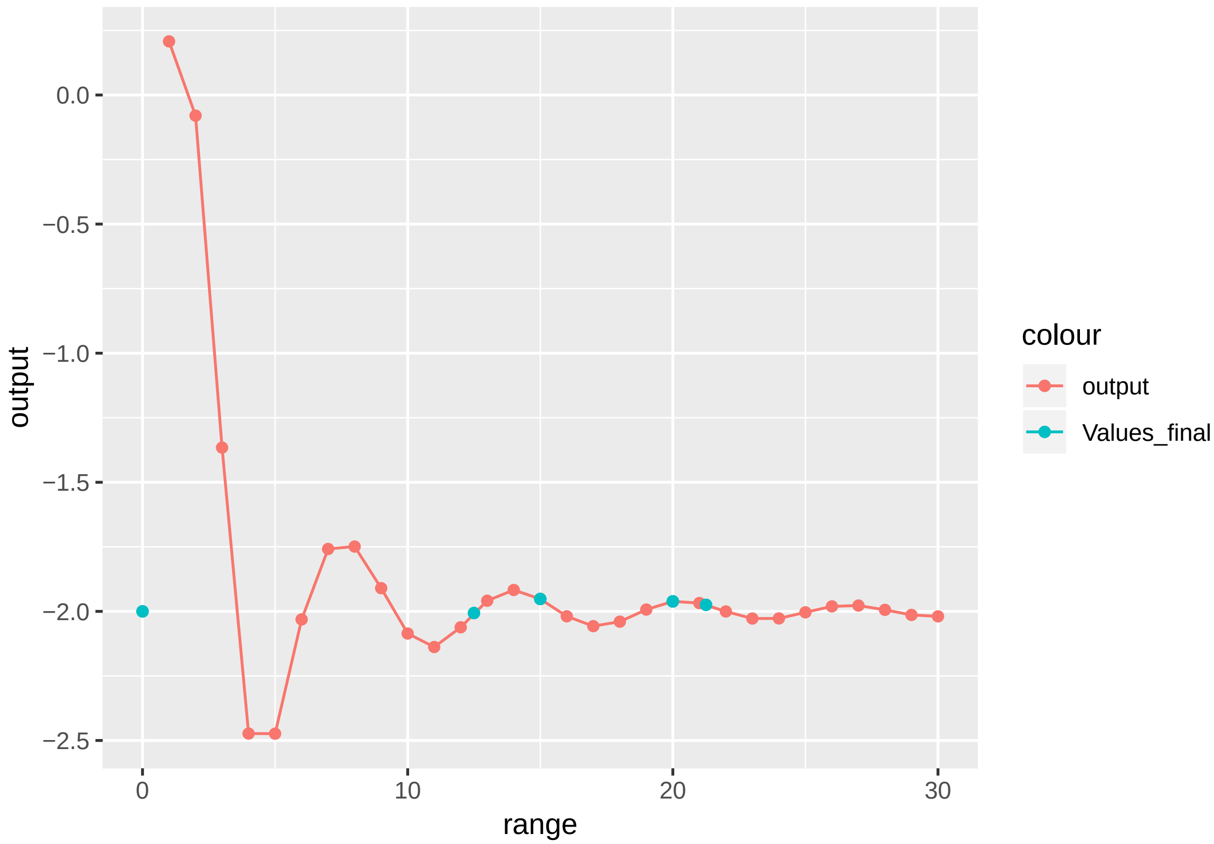
```
    }

    X[victim_index] <- kid

    for (j in 1:length(X)){
      Values[j] <- function_x(X[j])
    }

    current_max <- max(Values)
  }


  answer$max <- current_max

  # Here we have to transform X_final so it fits in the dataframe
  X_final <- c(rep(X, 4), X[1], X[2])
  Values_final <- c(rep(Values, 4), Values[1], Values[2])


  df <- as.data.frame(cbind(output, X_final, Values_final))

  plot2 <- ggplot(df, aes(x = range, y= output, col = "output")) + geom_point() + geom_line()
  plot2 <- plot2 + geom_point(aes(x = X_final, y=Values_final, col="Values_final"))

  answer$plot2 <- plot2

  return(answer)
}
```

```
f(10, 0.1)
```

```
## $Plot
```



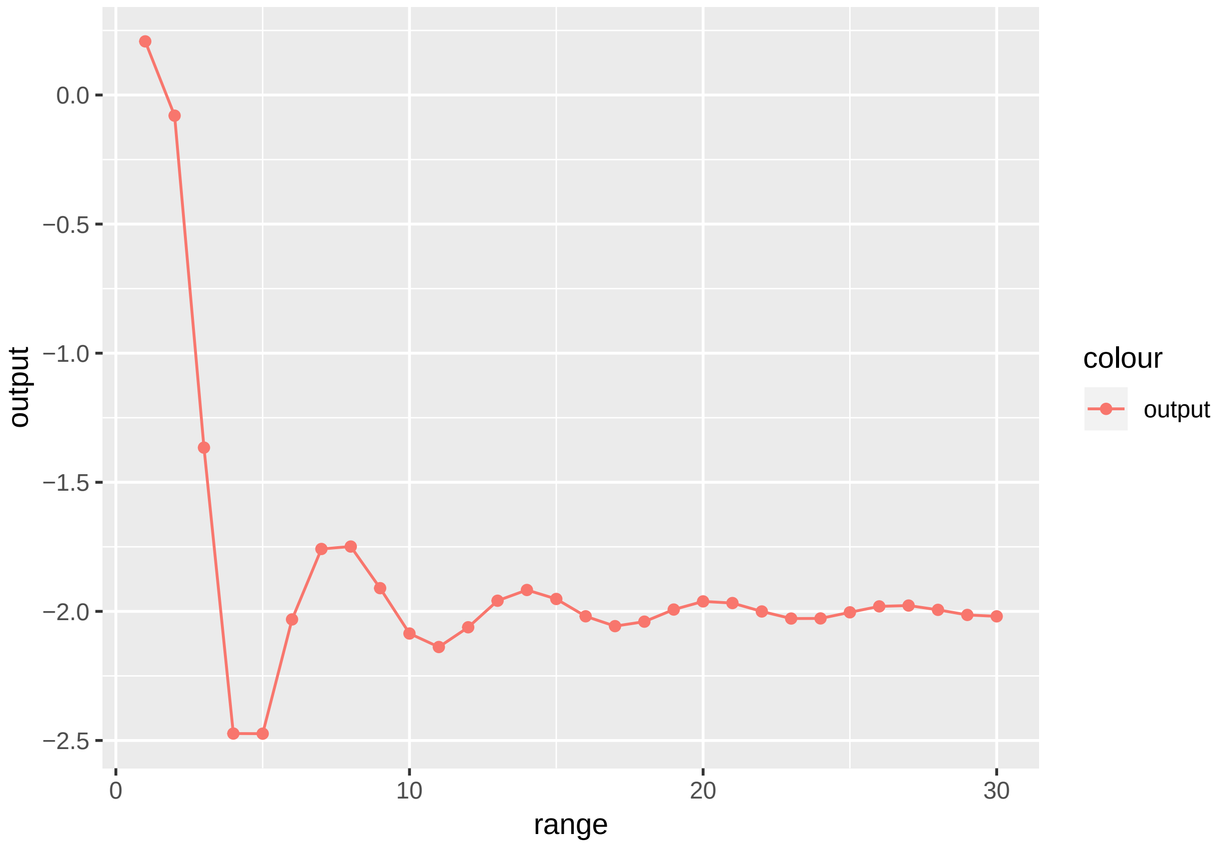

```
##
## $Values
## [1] -2.000000 -2.473573 -2.085654 -1.951947 -1.961344 -2.003663 -2.019194
##
## $max
## [1] -1.951947
##
## $plot2
```



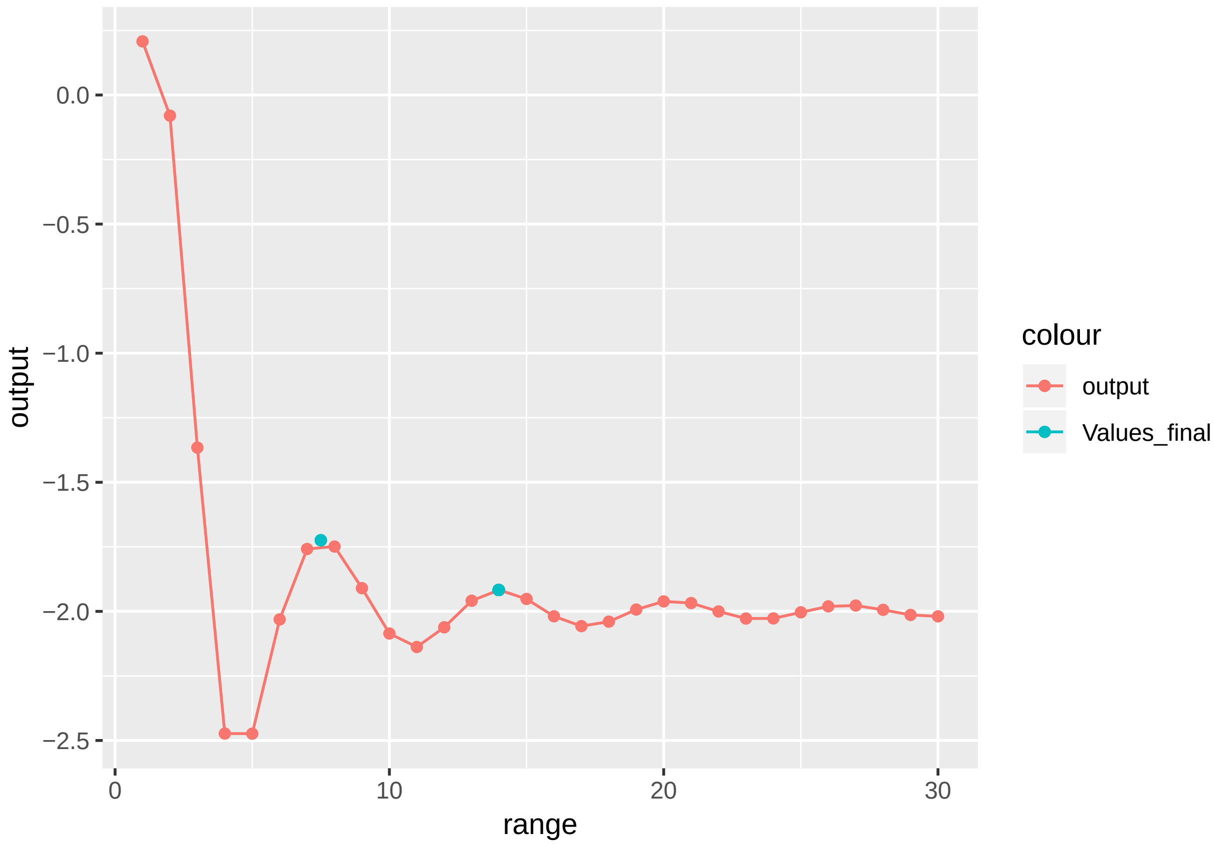
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
output
0
10
20
30
range
colour
output
Values_final


```
f(100, 0.1)
```

```
## $Plot
```

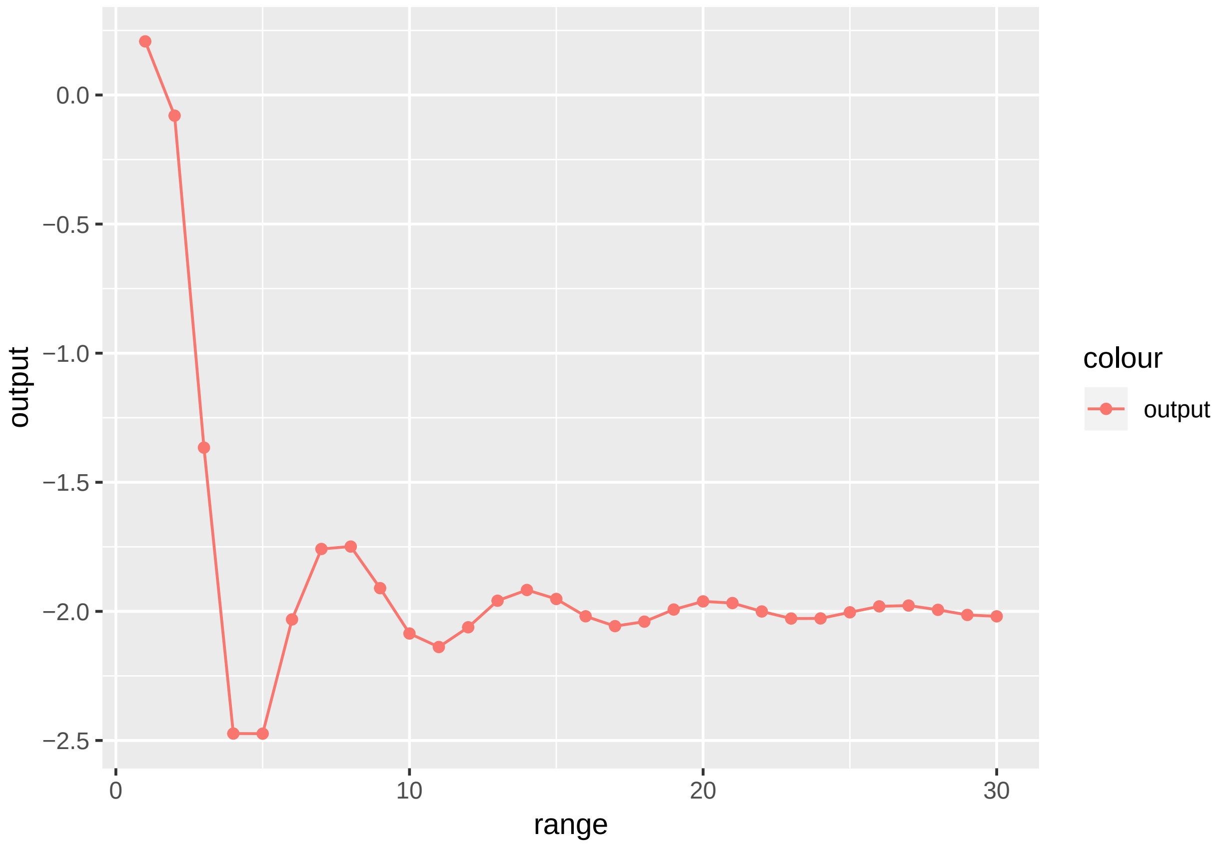



```
##
## $Values
## [1] -2.000000 -2.473573 -2.085654 -1.951947 -1.961344 -2.003663 -2.019194
##
## $max
## [1] -1.724415
##
## $plot2
```

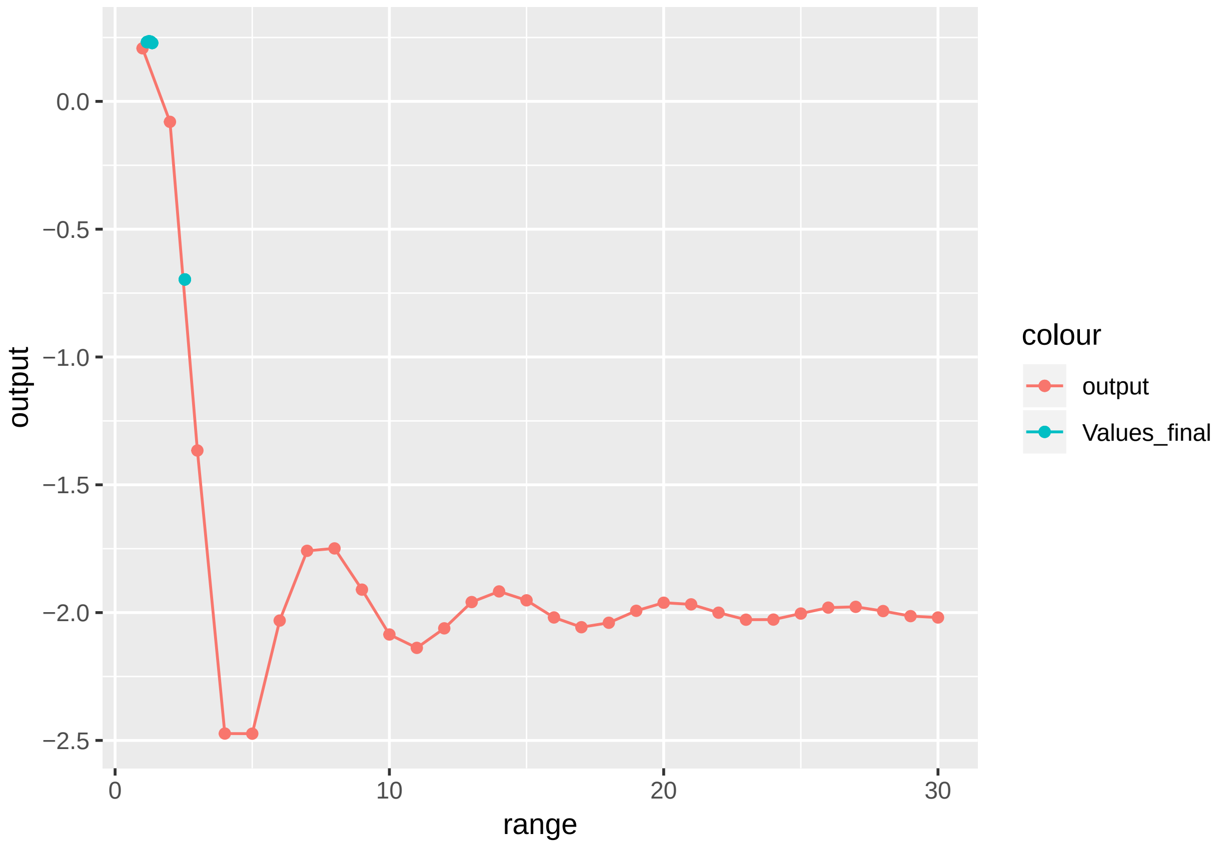



```r
f(100, 0.9)
```

```
## $Plot
```




```
##
## $Values
## [1] -2.000000 -2.473573 -2.085654 -1.951947 -1.961344 -2.003663 -2.019194
##
## $max
## [1] 0.2347839
##
## $plot2
```

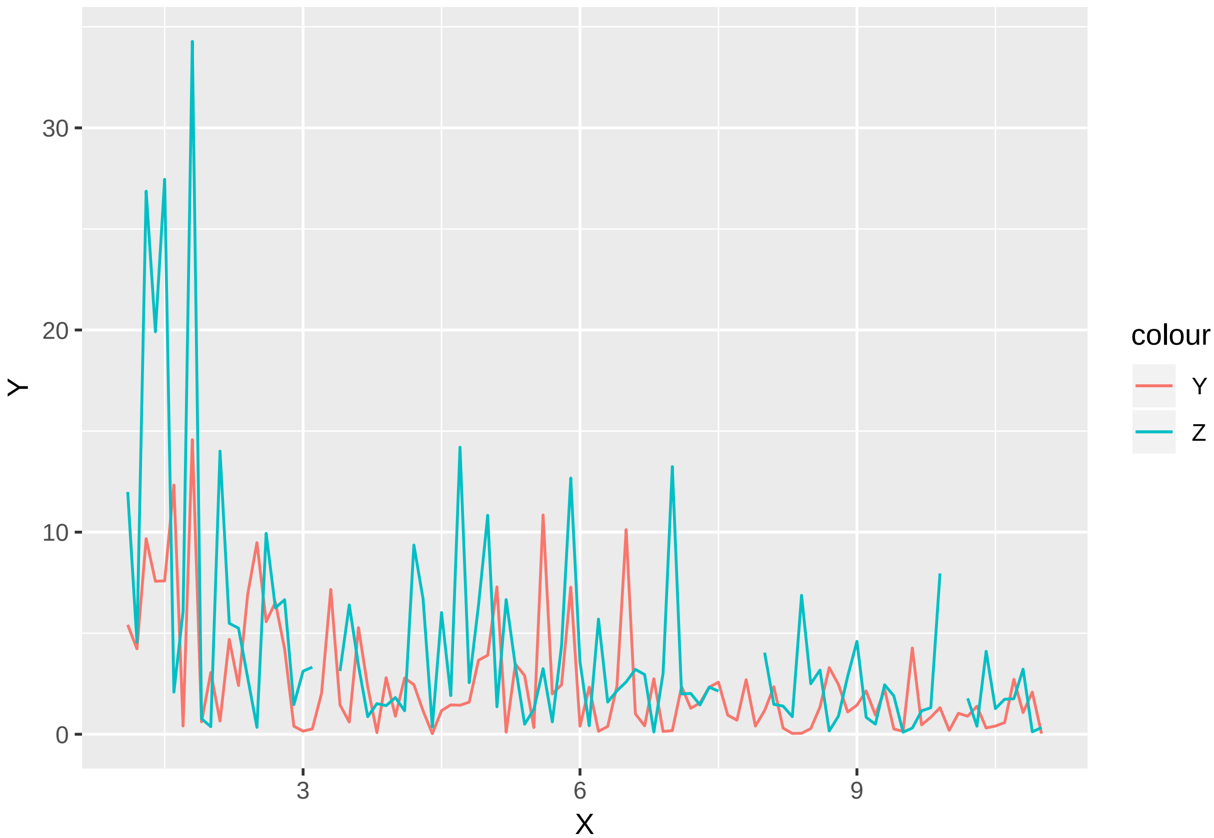


Question 2

2.1

```
physical <- read.csv("physical1.csv")
```

```
physical_plot <- ggplot(physical, aes(X))+
  geom_line(aes(y = Y, col = "Y")) +
  geom_line(aes(y = Z, col = "Z"))

physical_plot
```



To some extent, both variables seem related, as X increases both Y and Z decrease in values and variance. iN general, Z has a higher variance than Y.

2.2

$$Y_i \sim exp(\frac{X_i}{\lambda}), \quad Z_i \sim exp(\frac{X_i}{2*\lambda})$$

$$p(X,Y,Z|\lambda) = \frac{X_i}{\lambda}e^{\frac{-X_i}{\lambda}}Y_i * \frac{X_i}{2\lambda}e^{\frac{-X_i}{2\lambda}}Z_i$$

$$L(\lambda) = \prod_{i=1}^{n}\frac{X_i^2}{2\lambda^2}e^{\frac{-X_i*(Y_i+\frac{Z_i}{2})}{\lambda}}$$

$$logL(\lambda) = 2ln(\prod_{i=1}^{n}X_i) - 2n*log(2\lambda) - \frac{1}{\lambda}(\sum_{i=1}^{n}X_iY_i + \sum_{i=1}^{r}X_i\frac{V_i}{2} + \sum_{i=1+r}^{n}X_i\frac{W_i}{2})$$

$$E[logL((\lambda|X,Y,Z)|\lambda^k)] = 2*ln(\prod_{i=1}^{n}X_i) - \frac{1}{\lambda}(\sum_{i=1}^{n}X_iY_i + \sum_{i=1}^{r}X_i\frac{V_i}{2} + (n-r)\lambda^k$$

$$\lambda = \frac{1}{2n}(\sum_{i=1}^{n}X_iY_i + \sum_{i=1}^{r}X_i*\frac{V_i}{2} + (n-r)\lambda^k)$$

2.3

```r
EM_function <- function(data, eps, kmax, lambda_k){

  X <- data$X
  Y <- data$Y
  Z <- data$Z

  Xobs <- X[!is.na(Z)]
  Zobs <- Z[!is.na(Z)]
  Zmiss <- Z[is.na(Z)]

  n <- length(X)
  r <- length(Zmiss)

  k <- 0
  llvalprev <- 0
  llvalcurr <- lambda_k

  print(c(llvalprev, llvalcurr, k))

  while ((abs(llvalprev-llvalcurr)>eps) && (k<(kmax+1))){
    llvalprev <- llvalcurr

    llvalcurr <- (1/(2*n)) * (sum(X*Y) + sum(Xobs*(Zobs/2)) + (n-r)*llvalprev)

    k <- k+1
  }

  print(c(llvalprev,llvalcurr,k))
}
```

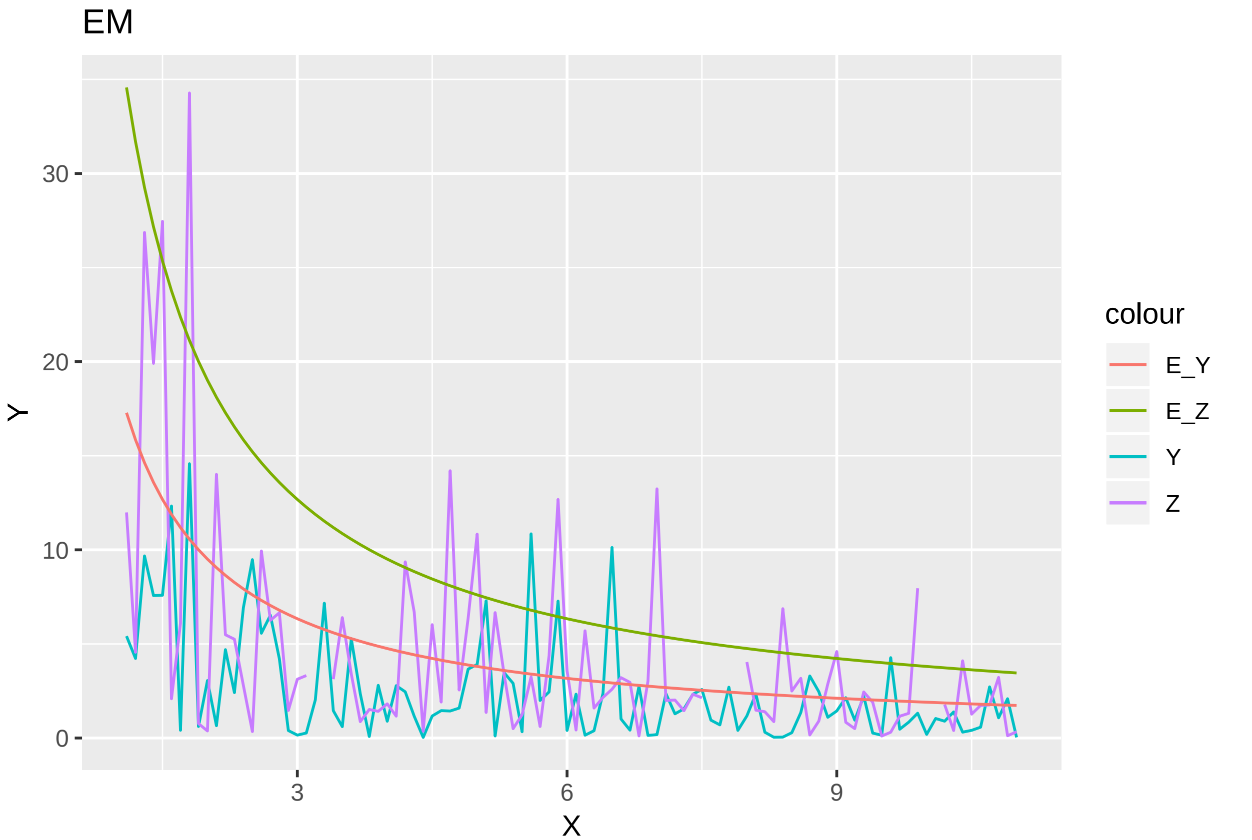
```r
EM_function(physical, 0.001, 50, 100)
```

```
## [1]   0 100   0
## [1] 19.01602 19.01519 15.00000
```

Optimal lambda is 19.01519, which is reached after 15 iterations.

4

```r
lambda <- 19.01519

df <- physical
df$E_Y <- lambda/physical$X
df$E_Z <- (2*lambda)/physical$X

df_plot <- ggplot(df, aes(X))+
  geom_line(aes(y = Y, col = "Y")) +
  geom_line(aes(y = Z, col = "Z")) +
  geom_line(aes(y = E_Y, col = "E_Y")) +
  geom_line(aes(y = E_Z, col = "E_Z")) +
  ggtitle("EM")

df_plot
```


EM
colour
E_Y
E_Z
Y
Z
30
20
10
0
3
6
9
X
Y