Computational Statistics Lab 6

Nahid Farazmand (nahfa911), Aashana Nijhawan(aasni448)

2/28/2019

Contents

---

1 Question 1: Genetic algorithm (one-dimensional maximization)

Objective function:

$$f(x) = \frac{x^2}{e^x} - 2exp(-\frac{9sinx}{x^2+x+1})$$

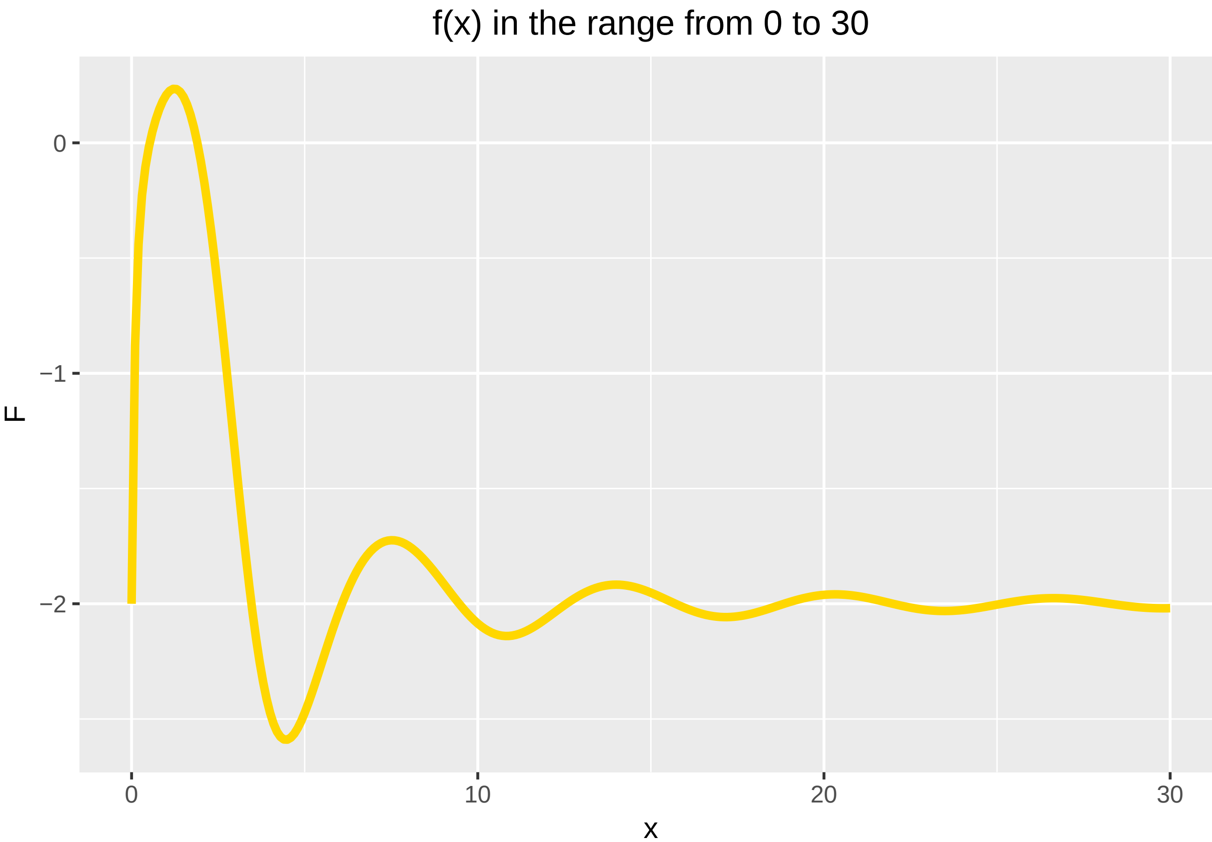
Crossover function:

$$\frac{x+y}{2}$$

Mutation function:

$$X^2 mod 30$$

1.0.1 function f in the range from 0 to 30



Based on the plot, function has a global maximum in this range.

1.1 maximum point

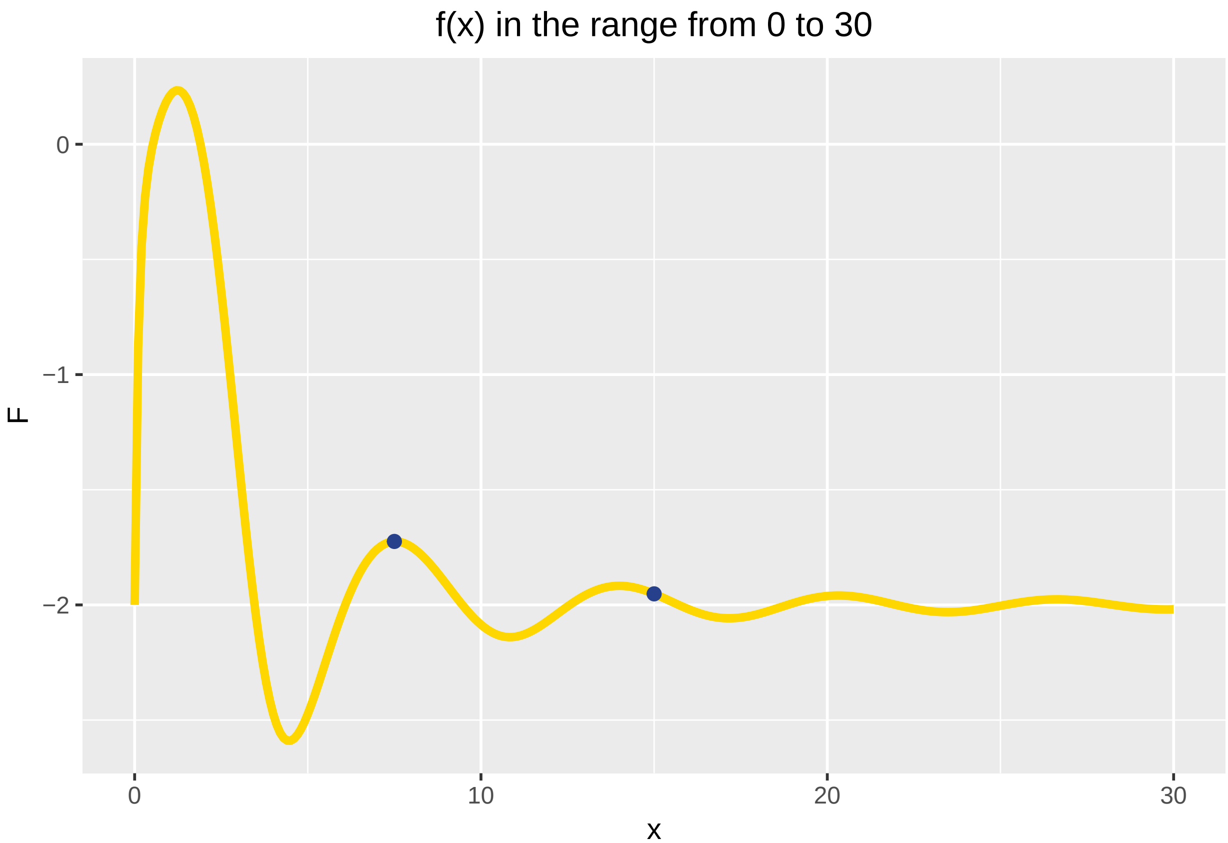
```
##      x         F
## 13 1.2 0.2341007
```

1.1.1 maxiter = 10 and mutprob = 0.1

1.1.1.1 Maximum values found

```
##      X    Values
## 1 15.0 -1.951947
## 7  7.5 -1.724415
```

1.1.1.2 Plot

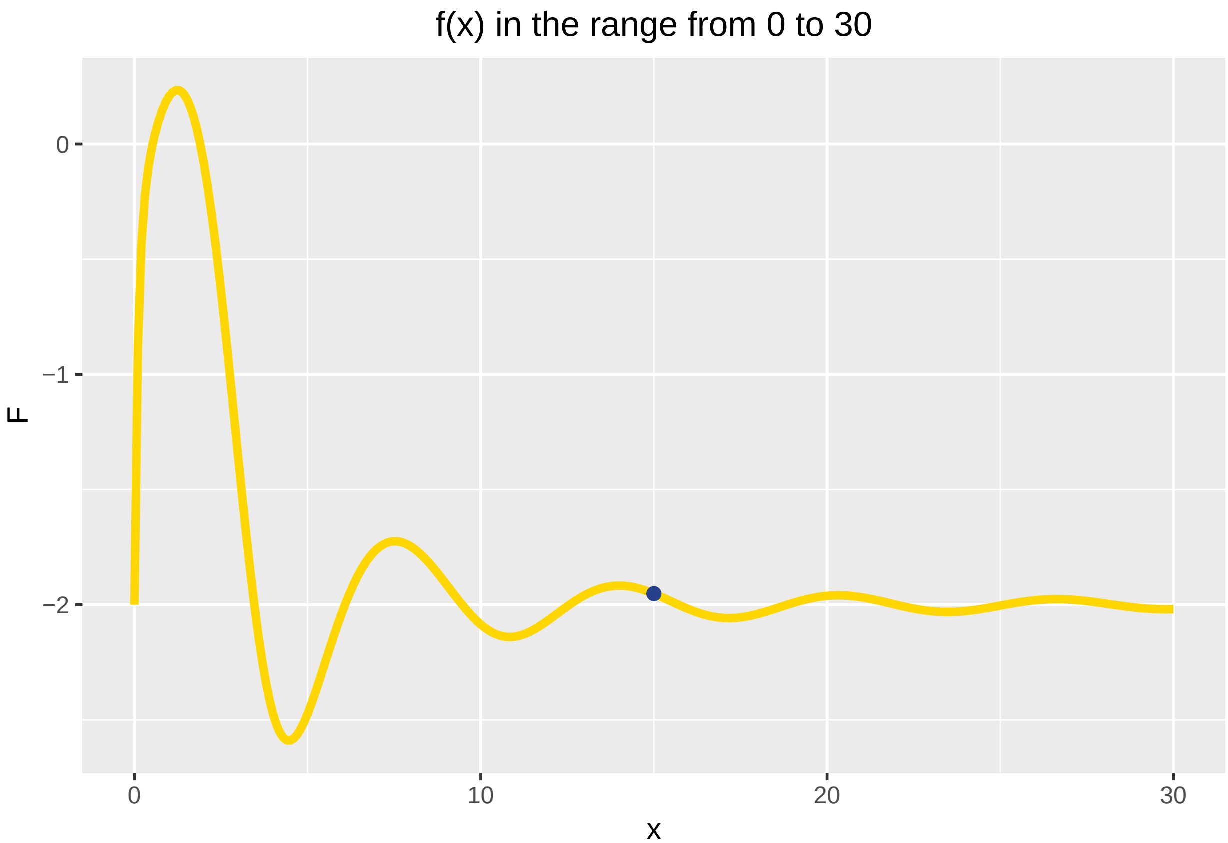


1.1.2 maxiter = 10 and mutprob = 0.5

1.1.2.1 Maximum values found

```
##    X    Values
## 1 15 -1.951947
```

1.1.2.2 Plot

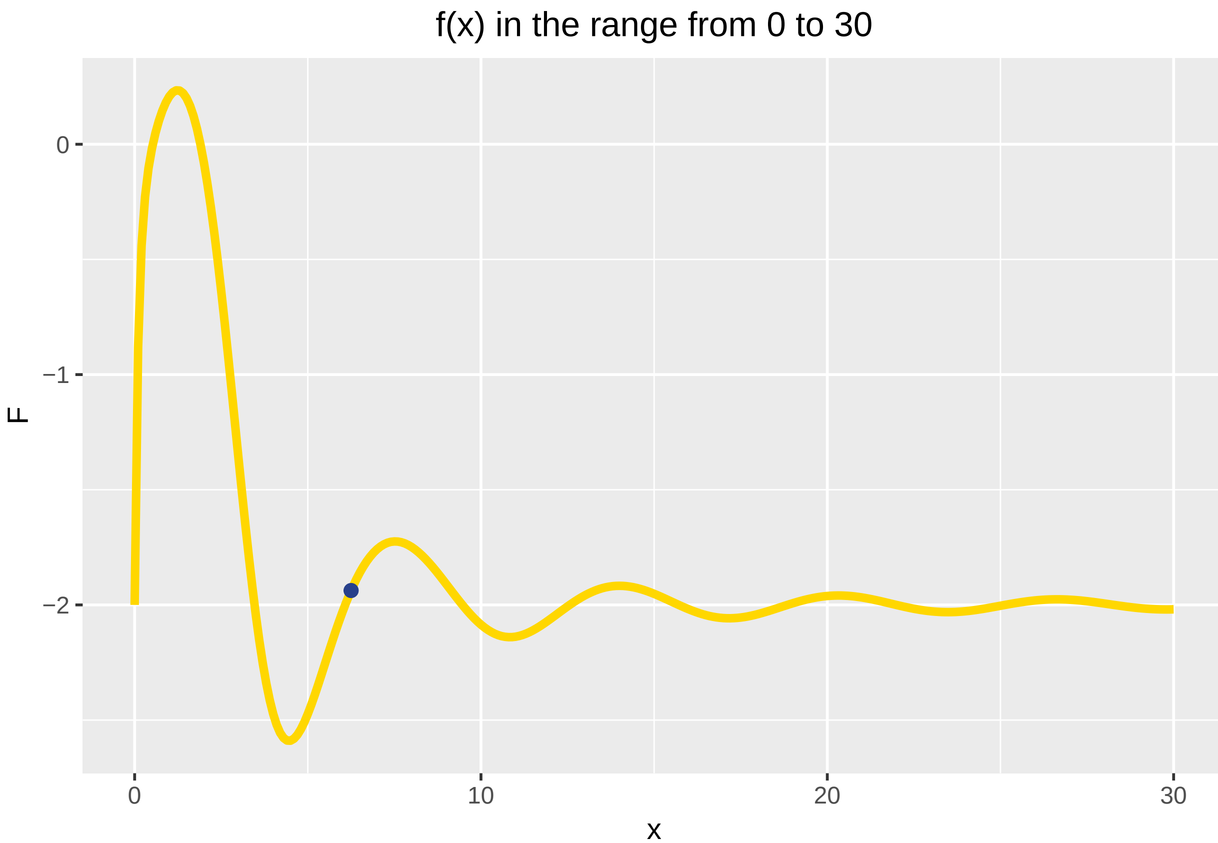


1.1.3 maxiter = 10 and mutprob = 0.9

1.1.3.1 Maximum values found

```
##      X    Values
## 1 6.25 -1.937529
```

1.1.3.2 Plot

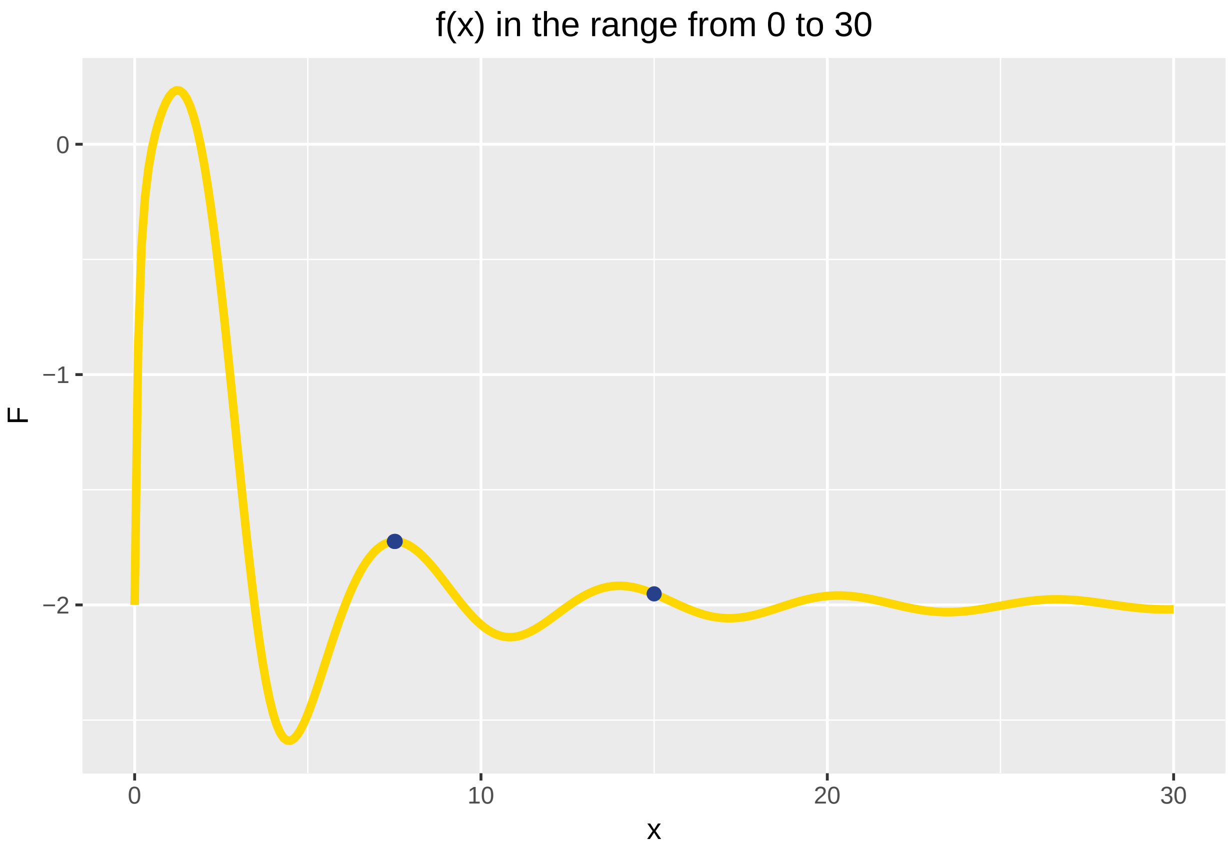


1.1.4 maxiter = 100 and mutprob = 0.1

1.1.4.1 Maximum values found

```
##          X    Values
## 1 15.00000 -1.951947
## 6  7.50000 -1.724415
## 2  7.52346 -1.724358
```

1.1.4.2 Plot

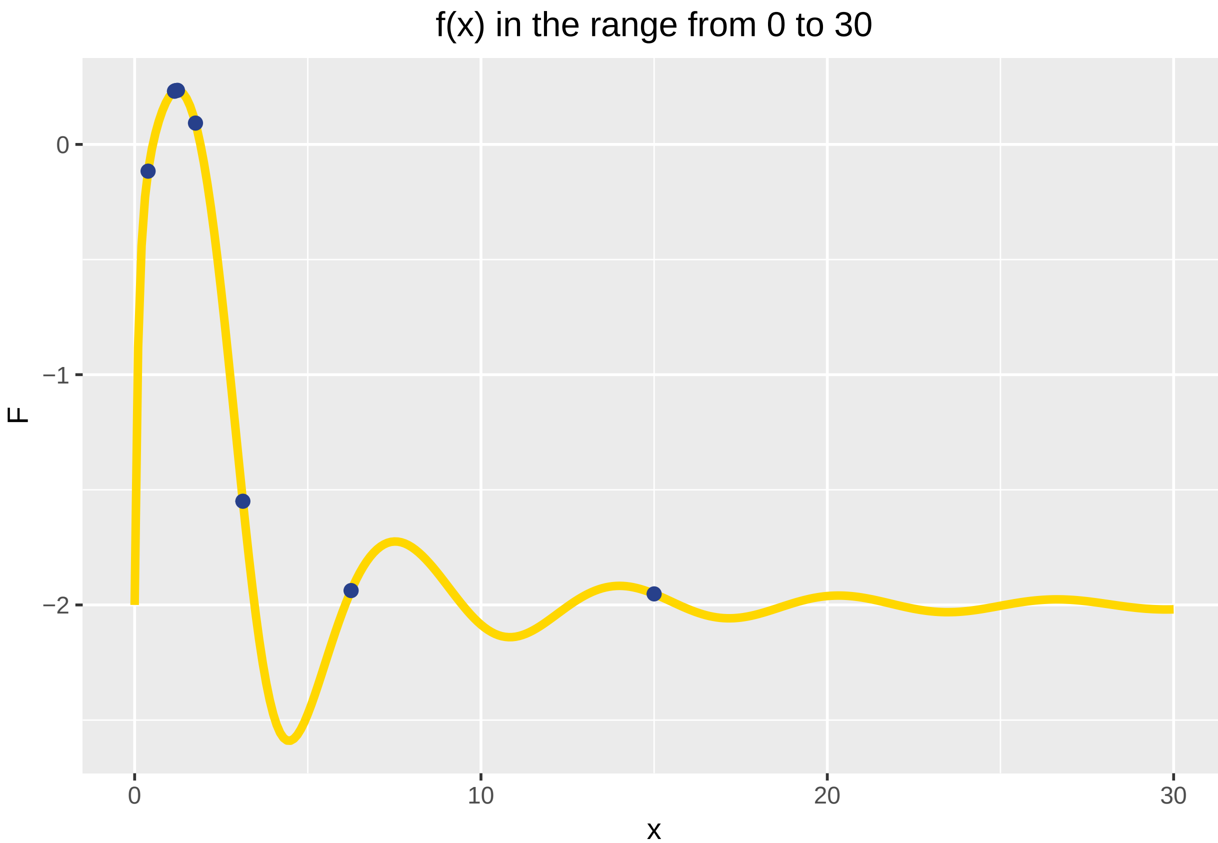


1.1.5 maxiter = 100 and mutprob = 0.5

1.1.5.1 Maximum values found

```
##              X      Values
## 1  15.0000000 -1.9519470
## 7   6.2500000 -1.9375289
## 3   3.1250000 -1.5495431
## 2   0.3883803 -0.1161797
## 28  1.7566901  0.0925651
## 33  1.1503318  0.2310126
## 39  1.1742994  0.2327976
## 46  1.1846050  0.2333965
## 49  1.2352981  0.2348457
```

1.1.5.2 Plot

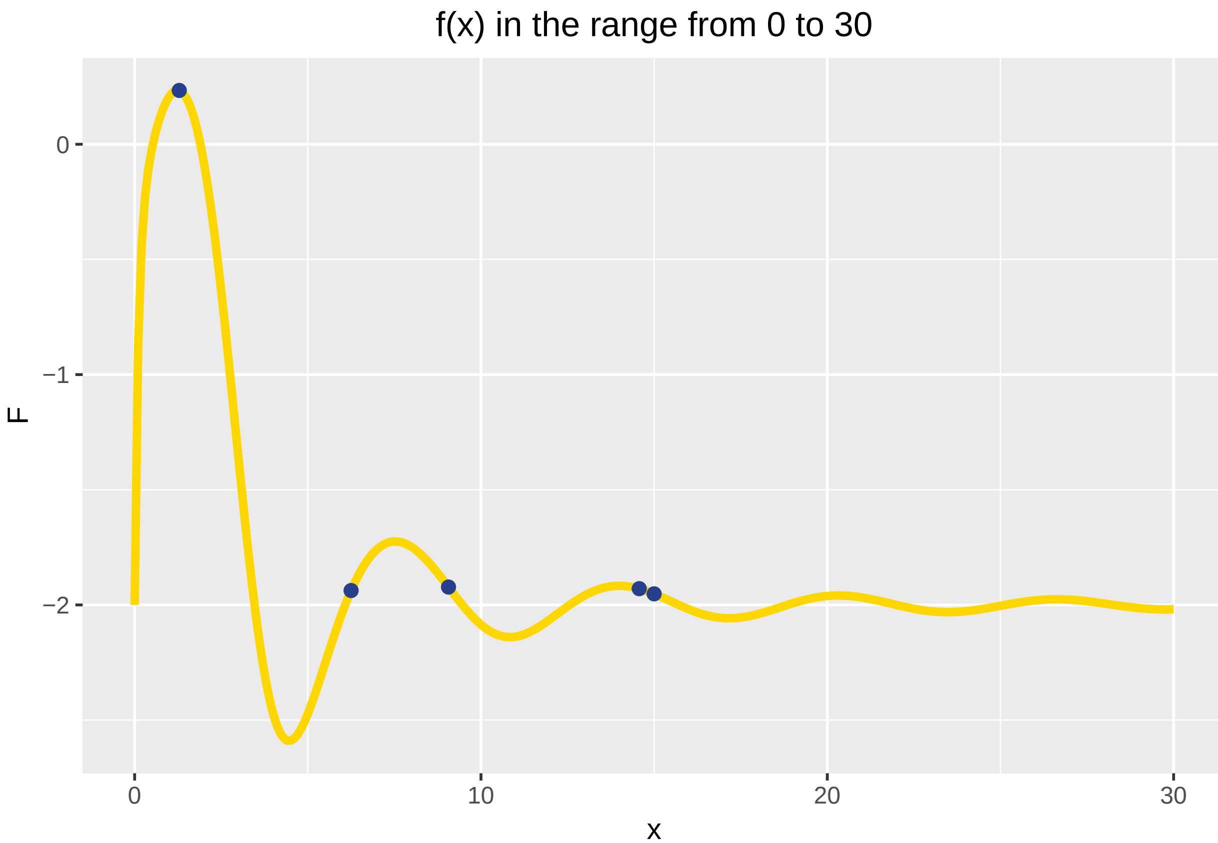


1.1.6 maxiter = 100 and mutprob = 0.9

1.1.6.1 Maximum values found

```
##              X     Values
## 1  15.000000 -1.9519470
## 4   6.250000 -1.9375289
## 19 14.570312 -1.9294709
## 28  9.062500 -1.9224657
## 32  1.288334  0.2336496
```

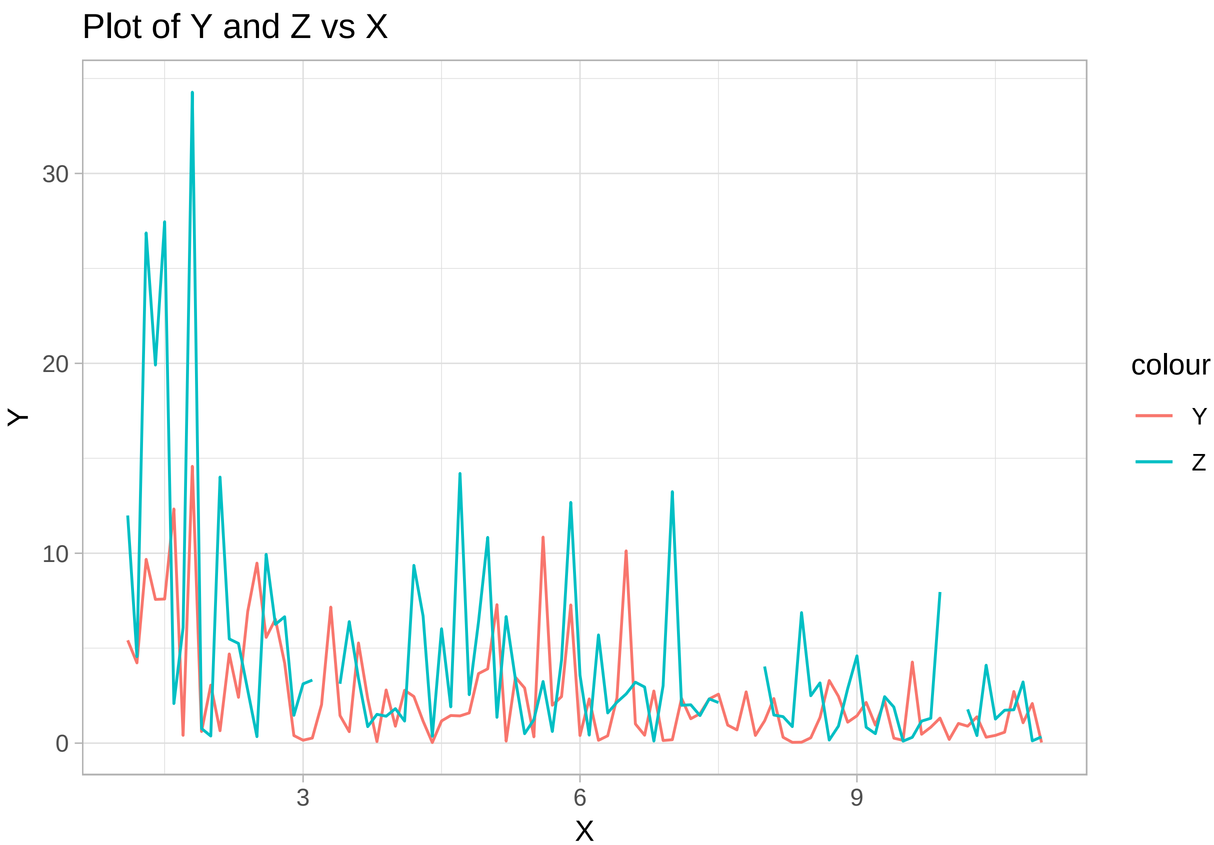
1.1.6.2 Plot

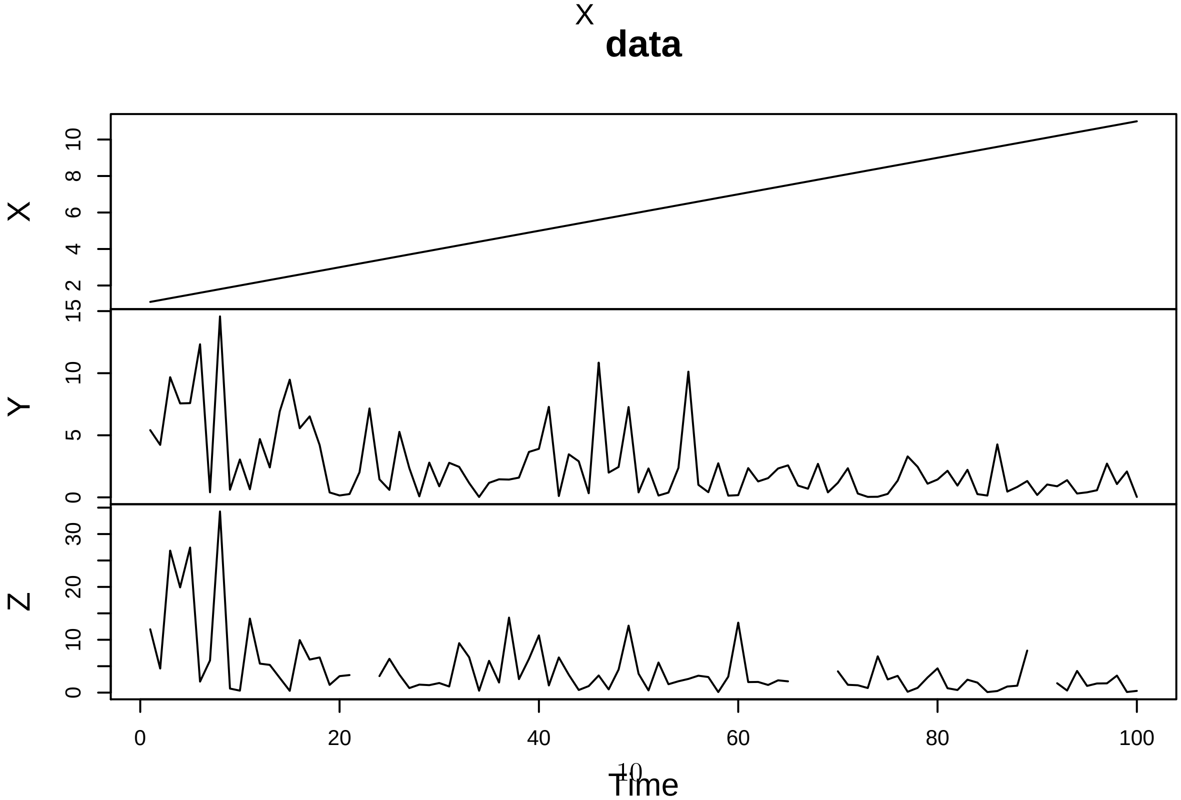


As we can see, when we increase the number of iterations (we create more generations) and we increase the chance of mutation, we can find near optimum solution.

2 Question 2

2.1 Make a time series plot describing dependence of Z and Y versus X. Does it seem that two processes are related to each other? What can you say about the variation of the response values with respect to X?





*Analisys: Both Y and Z have a similar trend, they decrease with time, with very similar amplitude. But Y and Z with respect to X seem to differ minutely.

2.2 Note that there are some missing values of Z in the data which implies problems in esti- mating models by maximum likelihood. Use the following model

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

Where λ is an unknown parameters. The goal is to derive the EM algorithm that estimates λ

$$\begin{aligned}
L(\lambda|Y,Z) &= \prod_{i=1}^{n} f(Y) \times \prod_{i=1}^{n} f(Z) \\
&= \prod_{i=1}^{n} \frac{X_i}{\lambda} \cdot e^{-\frac{X_i}{\lambda} Y_i} \times \prod_{i=1}^{n} \frac{X_i}{2\lambda} \cdot e^{-\frac{X_i}{\lambda} Z_i} \\
&= \frac{X_1 \cdot \ldots \cdot X_n}{\lambda^n} \times e^{-\frac{1}{\lambda}\sum_1^n X_i Y_i} \times \frac{X_1 \cdot \ldots \cdot X_n}{(2\lambda)^n} \times e^{-\frac{1}{2\lambda}\sum_1^n X_i Z_i} \\
lnL(\lambda|Y,Z) &= \sum_{i=1}^{n} ln(X_i) - nln(\lambda) - \frac{1}{\lambda}\sum_{i=1}^{n} X_i Y_i \ + \sum_{i=1}^{n} ln(X_i) - nln(2\lambda) - \frac{1}{2\lambda}\sum_{i=1}^{n} X_i Z_i
\end{aligned}$$

2.2.1 E-step : Derive Q function

Obtaining the expected values for the missing data using an initial parameter estimate.

$$\begin{aligned}
Q(\theta, \theta^k) &= E[\ loglik(\lambda|Y,Z) \mid \lambda^k, (Y,Z)] \\
&= \sum_{i=1}^{n} ln(X_i) - nln(\lambda) - \frac{1}{\lambda}\sum_{i=1}^{n} X_i Y_i \ + \sum_{i=1}^{n} ln(X_i) - nln(2\lambda) \\
&\quad - \frac{1}{2\lambda}\left[\sum_{i=1}^{n} X_i Z_i \ + m \cdot X_i \cdot \frac{2\lambda_{k-1}}{X_i}\right]
\end{aligned}$$

Here, we are taking expectation on the missing values in Z, so we need to seperate the Z_{obs} and Z_{miss}. Here we are assuming there are 'm' missing Z values. λ_k is the lambda value from the previous iteration.

2.2.2 M-step

Obtain the maximum likelihood estimate of the parameters by taking the derivative with respect to λ. Repeat till estimate converges.

$$\begin{aligned}
-\frac{n}{\lambda} - \frac{n}{\lambda} \ + \frac{\sum_{i=1}^{n} X_i Y_i}{\lambda^2} \ + \frac{\sum_{i}^{m} X_i Z_i \ + m \cdot 2\lambda_{k-1}}{2\lambda^2} &:= 0 \\
-2\lambda(2n) + 2\sum_{i=1}^{n} X_i Y_i \ + \sum_{i=1}^{n} X_i Z_i + m \cdot 2\lambda_{k-1} &:= 0
\end{aligned}$$

$$\lambda = \frac{\sum_{i=1}^{n} X_i Y_i + \frac{1}{2}\sum_{i=1}^{n} X_i Z_i + m \cdot \lambda_{k-1}}{2n}$$

2.3 Implement this algorithm in R, use $\lambda\_0 = 100$ and convergence criterion "stop if the change in λ is less than 0.001". What is the optimal λ and how many iterations were required to compute it?

```
## [1] 110.01 100.00
## [1]   1.00000 100.00000  15.32735
## [1]  2.00000 15.32735 11.64593
## [1]  3.00000 11.64593 11.48587
## [1]  4.00000 11.48587 11.47891
## [1]  5.00000 11.47891 11.47861

## [1] 11.47861
```

*Analysis: The Optimal value of lambda is 11.47861 which was converged after 5 iterations.

2.4 Plot E[Y] and E[Z] versus X in the same plot as Y and Z versus X. Comment whether the computed λ seems to be reasonable.

$$E[Y] = \frac{X_i}{\lambda} \ , \quad E[Z] = \frac{X_i}{2\lambda}$$

```
lambda <- 11.47861

X<- data$X
data$E_Y <- lambda/X
data$E_Z <- 2*data$E_Y



ggplot(data=data,aes(x=X, group=1)) +
  geom_line(aes(y = Y, colour = "Y")) +
  geom_line(aes(y = Z, colour = "Z")) +
  geom_line(aes(y = E_Y, colour = "E_Y")) +
  geom_line(aes(y = E_Z, colour = "E_Z")) +
  ggtitle("Plot of Y,Z and their expected value vs. X")
```

3 Appendix

```
knitr::opts_chunk$set(echo = TRUE)
# Question 1
set.seed(123456)
f <- function(x){
  ((x^2)/exp(x))-2*exp(-9*sin(x)/(x^2+x+1))
}
crossover <- function(x,y){
  (x+y)/2
}
mutate <- function(x){
  x^2%%30
}
library(ggplot2)
df = data.frame(x = seq(0,30,0.1),F = sapply(X = seq(0,30,0.1),FUN = f))
ggplot(data = df)+
geom_line(mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
```

```r
df[which.max(df$F),]
GA <- function(maxiter,mutprob){
population <- data.frame(X = seq(0,30,5),Values = sapply(X = seq(0,30,5),FUN = f))
maximums <- data.frame(X = 0,Values = 0)
for(i in 1:maxiter){
  parents <- sample(x = population$X,size = 2)
  victim <- population[which.min(population$Values),]
  cross_kid <- crossover(parents[1],parents[2])
  new_member <- ifelse(runif(1)<= mutprob,mutate(cross_kid),cross_kid)
  population[which(population$X == victim$X)[1],] <- c(new_member,f(new_member))
  maximums[i,] <- population[which.max(population$Values),]
}
 unique(maximums)
}
df1 <- GA(10,0.1)
df1
ggplot()+
geom_line(data = df ,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
geom_point(data = df1 ,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
df2 <- GA(10,0.5)
df2
ggplot()+
geom_line(data = df,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
geom_point(data = df2,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
df3 <- GA(10,0.9)
df3
ggplot()+
geom_line(data = df,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
geom_point(data = df3,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
df4 <- GA(100,0.1)
df4
ggplot()+
geom_line(data = df,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
```

```
geom_point(data = df4,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
df5 <- GA(100,0.5)
df5
ggplot()+
geom_line(data = df,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
geom_point(data = df5,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
df6 <- GA(100,0.9)
df6
ggplot()+
geom_line(data = df,mapping = aes(x = x,y = F),
          color = 'gold',
          size = 1.5)+
geom_point(data = df6,mapping = aes(x = X,y = Values),
          color = 'royalblue4',
          size = 2) +
ggtitle('f(x) in the range from 0 to 30') +
theme(plot.title = element_text(hjust = 0.5))
library(ggplot2)
data <- read.csv("physical1.csv")

ggplot(data=data, aes(x=X))+
  geom_line(aes(y = Y, color="Y"))+
  geom_line(aes(y = Z, color="Z"))+
  ggtitle("Plot of Y and Z vs X")+
  theme_light()

plot.ts(data)

EM.Norm<-function(data,eps,kmax,lambda_0=100){

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

  X <- X[!is.na(Z)]
  Zobs <- Z[!is.na(Z)] #observed data # Yobs
  Zmiss <- Z[is.na(Z)] #missing data  Ymiss

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

  k<-0
```

```r
  #muk<-1
  #sigma2k<-0.1

  llvalprev<- lambda_0+10+10*eps

  llvalcurr<-lambda_0

  print(c(llvalprev, llvalcurr))

  while ((abs(llvalprev-llvalcurr)>eps) && (k<(kmax+1))){
    llvalprev<-llvalcurr
    ## E-step
    llvalcurr <- (1/(2*n))  * (sum(X*Y) + sum(X*Zobs)/2 + m*llvalprev)

    ## M-step
    #muk<-EY/n
    #sigma2k<-EY2/n-muk^2

    ## Compute log-likelihood
    #llvalcurr<-floglik(Yobs,muk,sigma2k,r)
    k<-k+1

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

EM.Norm(data,0.001,500,100)

lambda <- 11.47861

X<- data$X
data$E_Y <- lambda/X
data$E_Z <- 2*data$E_Y



ggplot(data=data,aes(x=X, group=1)) +
  geom_line(aes(y = Y, colour = "Y")) +
  geom_line(aes(y = Z, colour = "Z")) +
  geom_line(aes(y = E_Y, colour = "E_Y")) +
  geom_line(aes(y = E_Z, colour = "E_Z")) +
  ggtitle("Plot of Y,Z and their expected value vs. X")
```