Computational Statistics

Lab 6

Emil K Svensson and Rasmus Holm

2017-03-14

Question 1

1.1

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

1.2

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

1.3

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

1.4

```
genetic <- function(maxiter, mutprob) {
    ## a)
    ## plot(x = 0:30, y= genfunc(0:30),  xlim = c(0,30), type ="l", xlab="", ylab="")

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

    ## c)
    values <- genfunc(X)
    ## points(X, values, col = "red")

    ## d)
    bestvalue <- -Inf

    for (i in 1:maxiter){

        ## i
        parents <- sample(1:length(X), size = 2 )

        ## ii
        victim <- which.min(values)

        ## iii
        child <- crossover(X[parents[1]],X[parents[2]])

        if (mutprob > runif(1,0,1)) {
            child <- mutate(child)
        }

        ## iv
        X[victim] <- child

        values[victim]<- genfunc(child)
        ## values <- genfunc(X)

        ## v
        bestvalue <- max(bestvalue, max(values))
    }

    ## points(x = X, y = values, col = "darkgreen")
    list(opt=bestvalue, pop=X, vals=values)
}
```

1.5

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
func_data <- data.frame(x=seq(0, 30, by=0.1), y=genfunc(seq(0, 30, by=0.1)))

set.seed(123456)
r1 <- genetic(maxiter = 10, mutprob = 0.1)
r1$opt
```

```
## [1] -1.724415
```

```
set.seed(123456)
r2 <- genetic(maxiter = 10, mutprob = 0.5)
r2$opt
```

```
## [1] 0.1433417
```

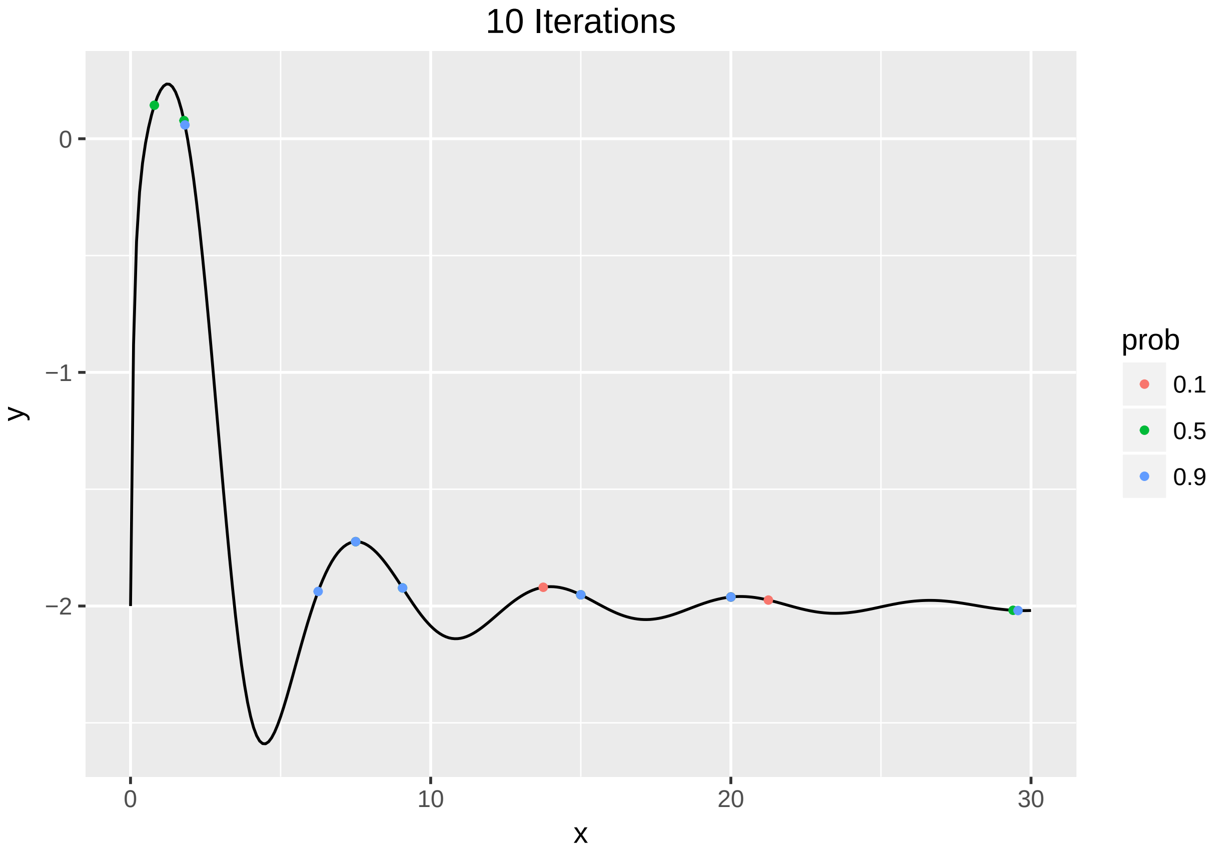
```
set.seed(123456)
r3 <- genetic(maxiter = 10, mutprob = 0.9)
r3$opt
```

```
## [1] 0.05909798
```

```
rd1 <- data.frame(x=r1$pop, y=r1$vals, prob="0.1")
rd2 <- data.frame(x=r2$pop, y=r2$vals, prob="0.5")
rd3 <- data.frame(x=r3$pop, y=r3$vals, prob="0.9")
```

```r
plot_data <- rbind(rd1, rd2, rd3)

ggplot() +
    ggtitle("10 Iterations") +
    geom_line(data=func_data, aes(x=x, y=y)) +
    geom_point(data=plot_data, aes(x=x, y=y, col=prob), size=1) +
    theme(plot.title=element_text(hjust=0.5))
```



```r
set.seed(123456)
r1 <- genetic(maxiter = 100, mutprob = 0.1)
r1$opt
```

```
## [1] -1.724415
```

```r
set.seed(123456)
r2 <- genetic(maxiter = 100, mutprob = 0.5)
r2$opt
```

```
## [1] 0.234853
```

```r
set.seed(123456)
r3 <- genetic(maxiter = 100, mutprob = 0.9)
r3$opt
```

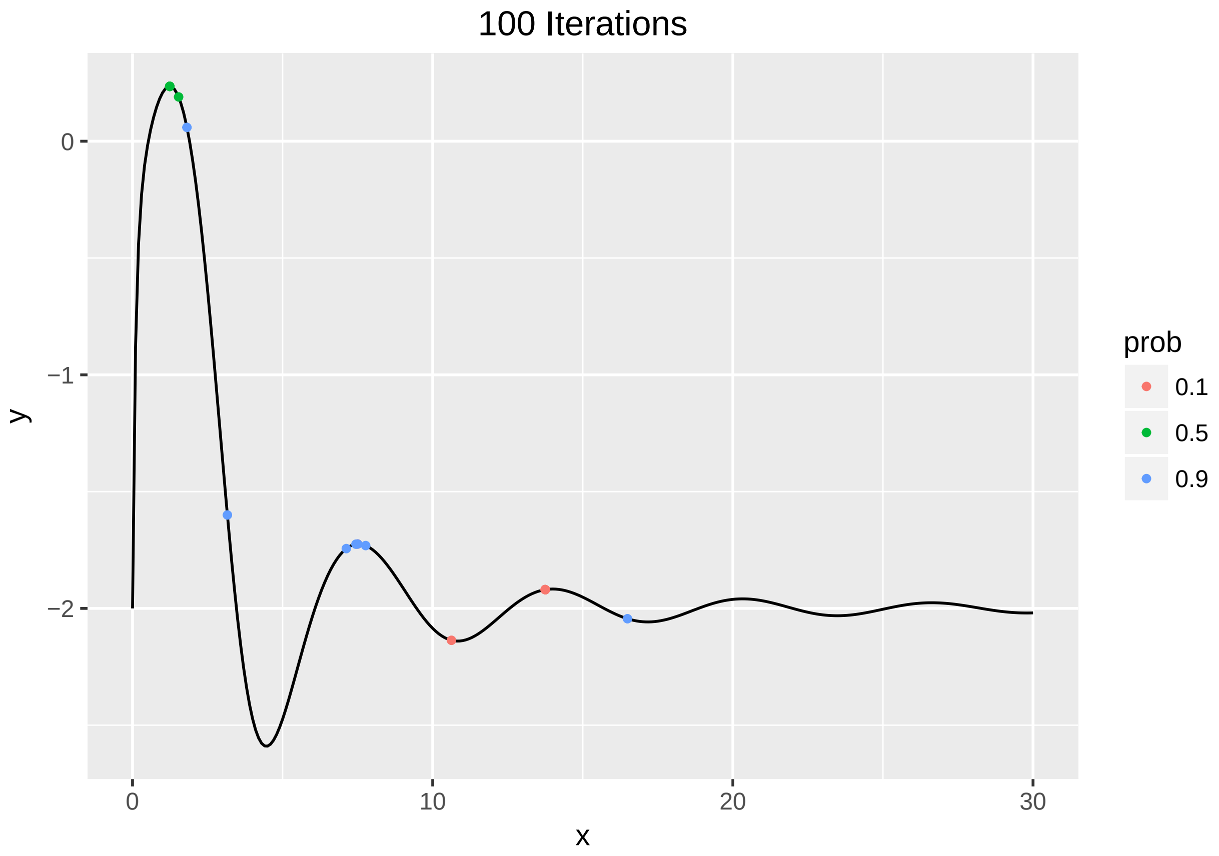
```
## [1] 0.05909798
```

```r
rd1 <- data.frame(x=r1$pop, y=r1$vals, prob="0.1")
rd2 <- data.frame(x=r2$pop, y=r2$vals, prob="0.5")
```

```r
rd3 <- data.frame(x=r3$pop, y=r3$vals, prob="0.9")

plot_data <- rbind(rd1, rd2, rd3)

ggplot() +
    ggtitle("100 Iterations") +
    geom_line(data=func_data, aes(x=x, y=y)) +
    geom_point(data=plot_data, aes(x=x, y=y, col=prob), size=1) +
    theme(plot.title=element_text(hjust=0.5))
```



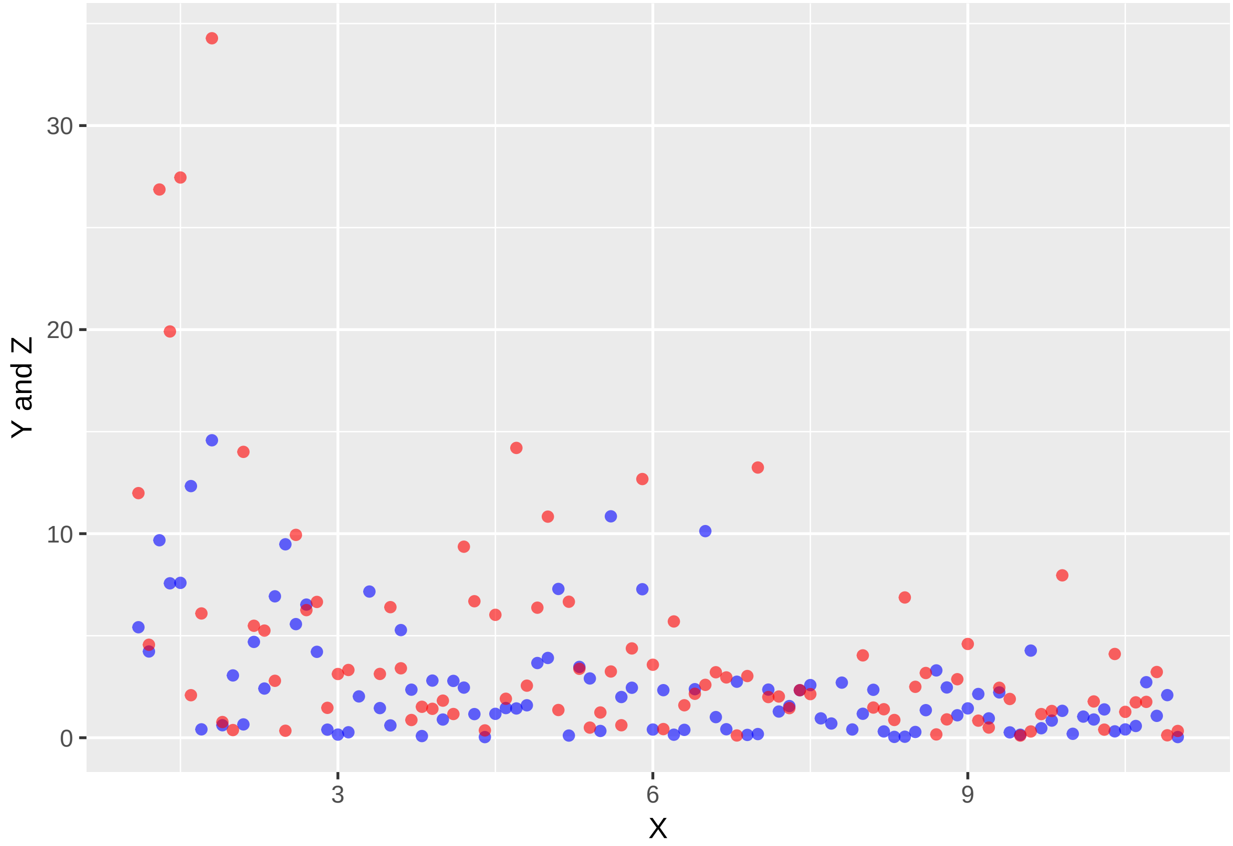
Question 2

2.1

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

```
##          X           Y          Z
## 1    1.1  5.41426038 11.9876973
## 2    1.2  4.22608491  4.5565103
## 3    1.3  9.67823013 26.8668629
## 4    1.4  7.56695696 19.9097345
## 5    1.5  7.58882421 27.4549423
## 6    1.6 12.33157648  2.0836649
## 7    1.7  0.41009367  6.0905027
## 8    1.8 14.57700780 34.2792579
## 9    1.9  0.61131742  0.7681544
## 10   2.0  3.05278879  0.3753011
## 11   2.1  0.65223747 14.0088838
## 12   2.2  4.69600719  5.4871676
## 13   2.3  2.41048971  5.2496933
## 14   2.4  6.92934501  2.7895926
## 15   2.5  9.47906221  0.3409472
## 16   2.6  5.56729641  9.9408793
## 17   2.7  6.52421313  6.2542210
## 18   2.8  4.20964317  6.6552438
## 19   2.9  0.39673717  1.4638228
## 20   3.0  0.15312502  3.1221710
## 21   3.1  0.26625998  3.3199876
## 22   3.2  2.02469517         NA
## 23   3.3  7.16325217         NA
## 24   3.4  1.45184534  3.1260663
## 25   3.5  0.60231149  6.3988460
## 26   3.6  5.27484068  3.4025707
## 27   3.7  2.35405444  0.8674649
## 28   3.8  0.08080272  1.5160847
## 29   3.9  2.79954170  1.4165794
## 30   4.0  0.89086317  1.8162871
## 31   4.1  2.78344372  1.1613882
## 32   4.2  2.45565955  9.3611431
## 33   4.3  1.15746976  6.6887097
## 34   4.4  0.03007524  0.3586731
## 35   4.5  1.16996660  6.0230034
## 36   4.6  1.45407495  1.9101763
## 37   4.7  1.43419971 14.2021234
## 38   4.8  1.58961571  2.5510073
## 39   4.9  3.66072309  6.3722029
## 40   5.0  3.90915954 10.8341204
## 41   5.1  7.29189803  1.3579596
## 42   5.2  0.10664384  6.6641682
## 43   5.3  3.47043571  3.3775118
## 44   5.4  2.90402285  0.4956739
## 45   5.5  0.33081461  1.2361602
## 46   5.6 10.85016640  3.2455308
## 47   5.7  1.99510803  0.6117629
## 48   5.8  2.44902751  4.3758904
## 49   5.9  7.27737634 12.6770674
## 50   6.0  0.40012627  3.5752652
## 51   6.1  2.33010272  0.4231419
## 52   6.2  0.14783192  5.6979513
## 53   6.3  0.38557054  1.5901769
## 54   6.4  2.37878831  2.1532181
## 55   6.5 10.12500980  2.5908817
## 56   6.6  1.01040332  3.2080633
## 57   6.7  0.41557150  2.9549790
## 58   6.8  2.74525109  0.1098097
## 59   6.9  0.13779928  3.0229634
## 60   7.0  0.17967304 13.2420590
## 61   7.1  2.35778883  1.9928370
## 62   7.2  1.28410376  2.0197468
## 63   7.3  1.54968353  1.4499798
## 64   7.4  2.32787297  2.3285021
## 65   7.5  2.57826908  2.1364811
## 66   7.6  0.94802258         NA
## 67   7.7  0.69651804         NA
## 68   7.8  2.70021144         NA
## 69   7.9  0.40366461         NA
## 70   8.0  1.17377661  4.0355408
## 71   8.1  2.34919450  1.4781564
## 72   8.2  0.30749101  1.3962715
## 73   8.3  0.04028545  0.8686591
## 74   8.4  0.04817157  6.8736889
## 75   8.5  0.28078776  2.4954061
## 76   8.6  1.35059321  3.1720180
## 77   8.7  3.29893435  0.1616723
## 78   8.8  2.46321233  0.8984915
## 79   8.9  1.10051288  2.8677836
## 80   9.0  1.43786124  4.5947406
## 81   9.1  2.14276035  0.8369501
## 82   9.2  0.94793495  0.4987080
## 83   9.3  2.22058192  2.4481776
## 84   9.4  0.26146477  1.8986584
## 85   9.5  0.14690426  0.1047659
## 86   9.6  4.26942600  0.3064197
## 87   9.7  0.46574722  1.1586172
## 88   9.8  0.83974572  1.3092032
## 89   9.9  1.31779070  7.9577418
## 90  10.0  0.19227490         NA
## 91  10.1  1.03576598         NA
## 92  10.2  0.89291068  1.7774044
## 93  10.3  1.38657868  0.3956468
## 94  10.4  0.31099918  4.1034586
## 95  10.5  0.40722449  1.2670664
## 96  10.6  0.57482608  1.7325363
## 97  10.7  2.71932023  1.7562741
## 98  10.8  1.07144489  3.2207667
## 99  10.9  2.08799806  0.1193872
## 100 11.0  0.03306950  0.3314914
```

```
ggplot(physical) +
    geom_point(aes(x=X, y=Y), col="blue", alpha = 0.6) +
    geom_point(aes(x=X, y=Z), col="red", alpha = 0.6) +
    labs(y = "Y and Z")
```



2.2

2.3

```
EM <- function(data, lambdazero, maxiter = 500, eps = 0.001){

  Estep <- function(data,lambda){

    r <- sum(is.na(data$z))
    n <- nrow(data)

      (2 * sum(log(data$x)) - 2 * n * log(2 * lambda) -
      sum(data$x * data$y)/lambda -
      sum(data$x * data$z, na.rm = TRUE)/(2*lambda) - (n - r))

  }

  Mstep <- function(data, lambda){
    n <- nrow(data)
    r <- sum(is.na(data$z))

     ( (sum(data$x * data$y) +
      sum(data$x * data$z, na.rm = TRUE) +
       (n-r) * lambda)/(2*n) )

  }


  curlambda <- lambdazero
  prevlambda <- lambdazero*5
  iter <- 1
  while(iter < maxiter && (abs(curlambda - prevlambda) > eps)){

    prevlambda <- curlambda
    curlambda <- Mstep(data,lambda = curlambda)

    iter <- iter + 1
  }

  return(list(lambda = curlambda, iter = iter))
}


colnames(physical) <- c("x","y","z")
res <- EM(data = physical, lambdazero = 100)
print(res)
```

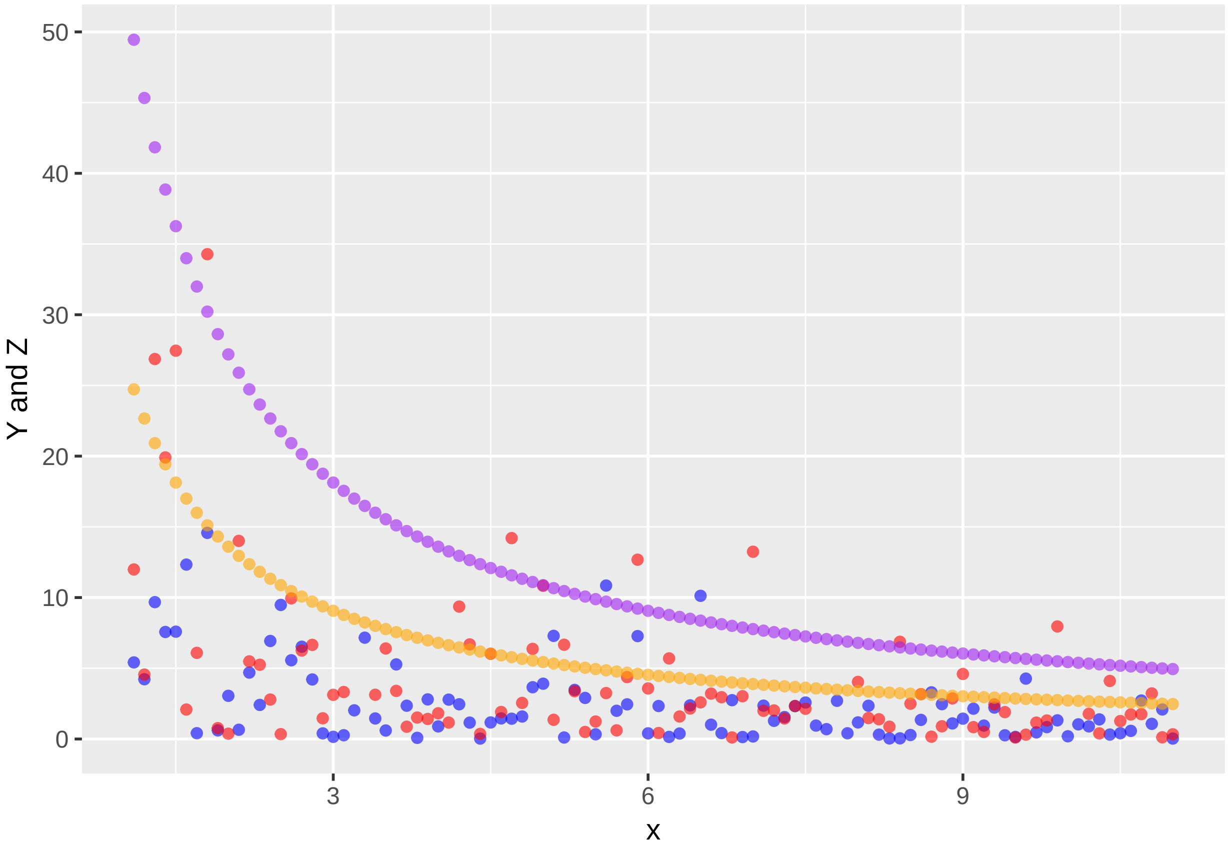
```
## $lambda
## [1] 27.1955
##
## $iter
## [1] 16
```

2.4

```
lamb <- res$lambda
physical$EZ <- (2*lamb)/physical$x
physical$EY <- lamb/physical$x



ggplot(physical) +
    geom_point(aes(x=x, y=y), col="blue", alpha = 0.6) +
    geom_point(aes(x=x, y=z), col="red", alpha = 0.6) +
    geom_point(aes(x=x, y=EZ), col="purple", alpha = 0.6) +
    geom_point(aes(x=x, y=EY), col="orange", alpha = 0.6) +
    labs(y = "Y and Z")
```


50
40
30
20
10
0
Y and Z
3
6
9
x