Homework 3

R Homework 3

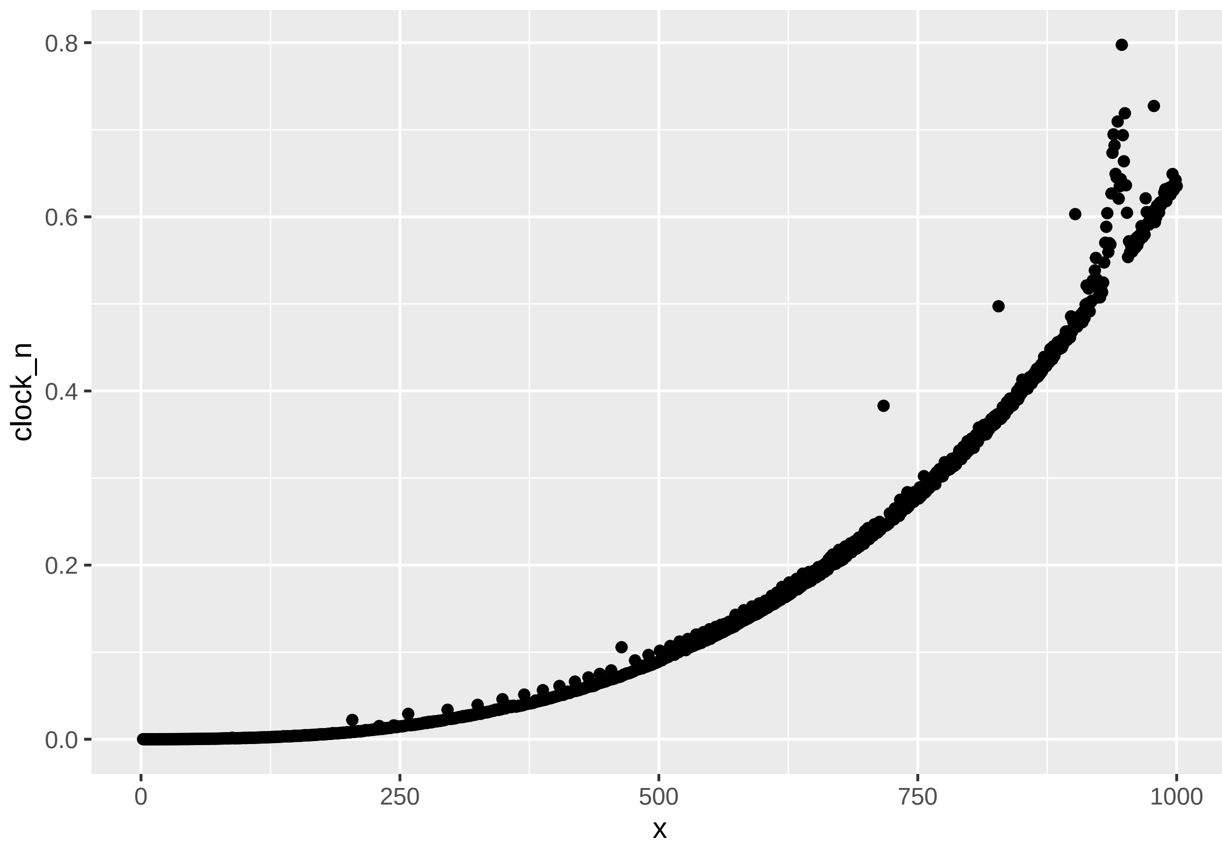
Exercise 1:

A <- n x n matrix, a_ij from normal distribution (0,1) rnorm(n * n,0,1). Consider n >= 2, say n from(2,1000) A^(-1)*(n* n) = solve(A ^-1* n*n). For each matrix A Create matrix A Inverse matrix Store clock time elapsed. Data (clock(n)) n from 2 to 1000 Regress Polynomial clock(n) on n y x

```
n <- 1000
clock_n <- NULL
x <- NULL
df <-NULL
```

```
set.seed(787)
for (i in 2:n) {
  x[i-1] <- i
  start_time <- Sys.time()
  A <- matrix(rnorm(i * i,0,1), nrow = i, ncol = i)
  A_inverse <- solve(A)
  end_time <- Sys.time()
  clock_n[i-1] <- end_time - start_time
}
```

```
df <- data.frame(x = x, y = clock_n)
library(ggplot2)
ggplot(df, aes(x=x,y=clock_n)) + geom_point()
```




```
set.seed(787)
degree <- 6
fit.train <- list()
for (j in 1:degree){
  fit.train[[j]] = lm(clock_n ~ poly(x,j), data=df)
}
```

```
mse <- NULL
for (i in 1: length(fit.train)) {
  mse[i] <- rev(anova(fit.train[[i]])$"Mean Sq")[1]
}
mse <- mean(mse)
print(mse)
```

```
## [1] 0.001275452
```

Exercise 2: Dn:= {xi,yi} ~iid p_xy(x,y),i= 1..,n a: lm(xy)<- lm(y~.,data = xy) coef(lm.xy) b: coef.xy <- solve(t(x)%%x , t(x)%%y)

Check if in R, a and b have same complexity.

```
set.seed(787)
n <- 1000
p <- 1000
Y <- matrix(data = rnorm(n, 0,1),nrow = n, ncol = 1)
```

```r
X <- matrix(data= rnorm(n * p, 0, 1), nrow = n, ncol = p)
Dn <- data.frame(x = X, y =Y)
start_time <- Sys.time()
lm.xy <- lm(y~., data = Dn)
coef1<- coef(lm.xy)
end_time <- Sys.time()
diff_time_1 <- end_time - start_time
print(diff_time_1)
```

```
## Time difference of 0.5769329 secs
```

```r
start_time <- Sys.time()
coef.xy <- solve(t(X)%*%X, t(X)%*%Y)
end_time <- Sys.time()
diff_time_2 <- end_time - start_time
print(diff_time_2)
```

```
## Time difference of 0.7518771 secs
```