R- Final Exam

Shazman Khan

DD-MSBA-4

Q2

Code:

```
library(MASS)
library(rpart)
library(rpart.plot)
library(plotly)
library(ROCR)
summary(Aids2)
as.data.frame(Aids2)
aids=Aids2
2843*0.7
aids$status<-gsub(aids$status,pattern='A',replacement=1)
aids$status<-gsub(aids$status,pattern='D',replacement=0)
#Ans2a
train_index <- sample.split(aids, SplitRatio=0.7)
train <- aids[ train_index,]
test <- aids[!train_index,]
gc_train <- train
gc_test <- test
#Ans 2b
my_mod <- glm(age~death+T.categ+sex, data=gc_train, family=poisson)
my_mod

predict_logit <-predict(my_mod, gc_train, type='response')
predict_logit
```

Output:

```
Call:  glm(formula = age ~ death + T.categ + sex, family = poisson, 
    data = gc_train)

Coefficients:
  (Intercept)          death    T.categhsid      T.categid     T.categhet  
T.categhaem   T.categblood  T.categmother  
    4.1389562     -0.0000293     -0.3201980     -0.2750949      0.1483289  
-0.2098921     -0.0858535     -2.6126071  
 T.categother           sexM  
    0.2780890     -0.1998361  

Degrees of Freedom: 599 Total (i.e. Null);  590 Residual
Null Deviance:      1629 
Residual Deviance: 1386         AIC: 4658
> 
> predict_logit <-predict(my_mod, gc_train, type='response')
> predict_logit
      261       177       416       195       886       437       983       5
05       164       302       902       135 
38.386562 37.631460 38.209251 38.682399 36.672419 43.213116 36.672419 38.0160
46 37.244215 34.485919 37.520252 38.471016 
      669       969       568       838       388       795       694       2
89        74       919       915       308 
36.933395 37.433498 37.073267 37.274786 38.221569 36.672419 37.301009 38.4450
97 38.816382 37.054804 28.281182 37.996000 
      932       356        21       999       548       254       459 
```

Ans2c:

Designed the model by checking correlations and choosing variables that have a good correltaions and then using it to predict values

From the regression model it is clear that age and T.Category are the most valuable variables

This tells us that the transmission of aids is largely dependent on age and the T.Categ which means that it is more susceptible for the youth while the category of transmission playing a huge role in if the patient survives or dies.

Q3

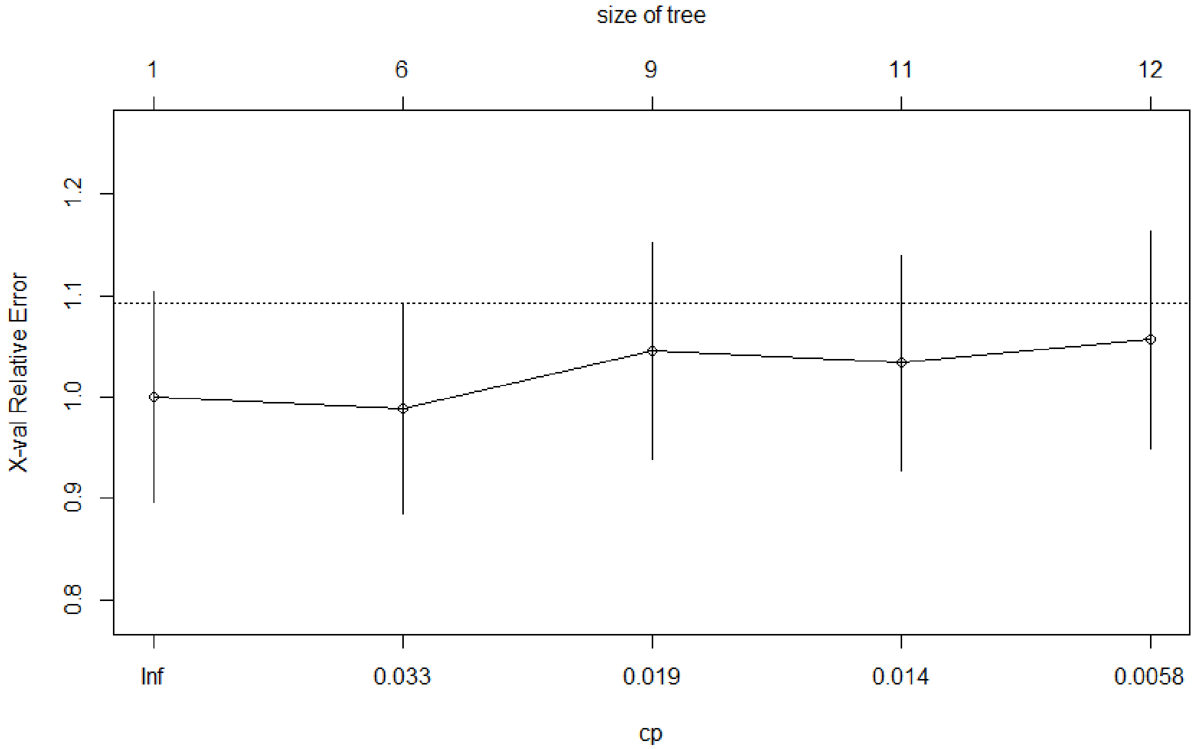
Code:

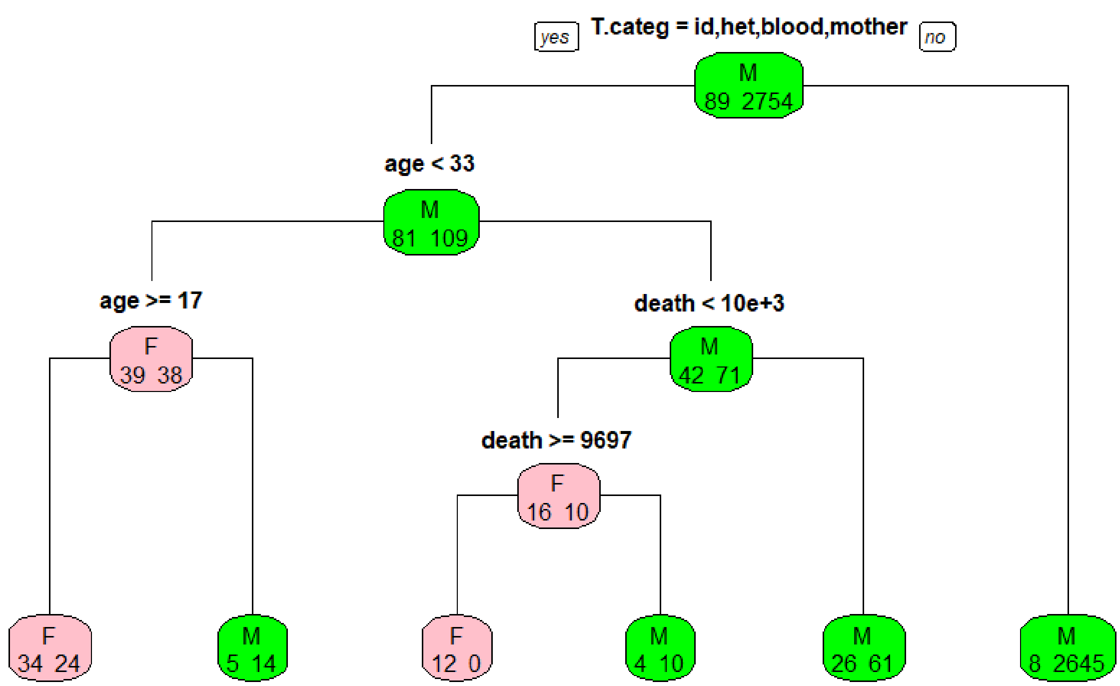
Code

```
library(MASS)

library(rpart)

library(rpart.plot)

library(plotly)

library(ROCR)

summary(Aids2)

as.data.frame(Aids2)

aids=Aids2

2843*0.7

aids$status<-gsub(aids$status,pattern='A',replacement=1)

aids$status<-gsub(aids$status,pattern='D',replacement=0)



train_index <- sample.split(aids, SplitRatio=0.7)

train <- aids[ train_index,]

test <- aids[!train_index,]

gc_train <- train

gc_test <- test

#Ans3a

my_tree <- rpart(sex~age+T.categ+death, data=aids,method="class",control=rpar
t.control(cp=0.033))
rpart.plot(my_tree,type=1,extra=1, box.palette = c("pink","green"))
#Ans 3b
plotcp(my_tree)

predict_tree <- predict(my_tree, aids, type='prob')
predict_tree

predict_logit <-predict(my_mod, gc_train, type='response')
predict_logit


perf_tree<-performance(predict_tree)
perf_logit <- performance(predict_logit)
```

```
plot(perf_tree,col="black")
plot(perf_logit,col='blue',add=TRUE)
```

Output





Ans a: The tree

Ans B: As it can be seen from the relativity plot given by plotcp(my_tree) the optimal cp is at 0.033

Ans C: Its clear from the tree that age and death are the most valuable variables to predict if a patient lives or dies in the transmission tree. The biggest insight from the tree is that the ages of being above 17 for female is plays a vital role in terms of transmission and above the age of 15 for men.

For example if the age is lower than 42 for men the probability of death raises by 10e+3

Ans D: In the performance graph it is clear to see that the tree performs better than the logistical regression and covers more are and is towards the left. Clearly the decision tree has performed better with this regard.

Q4

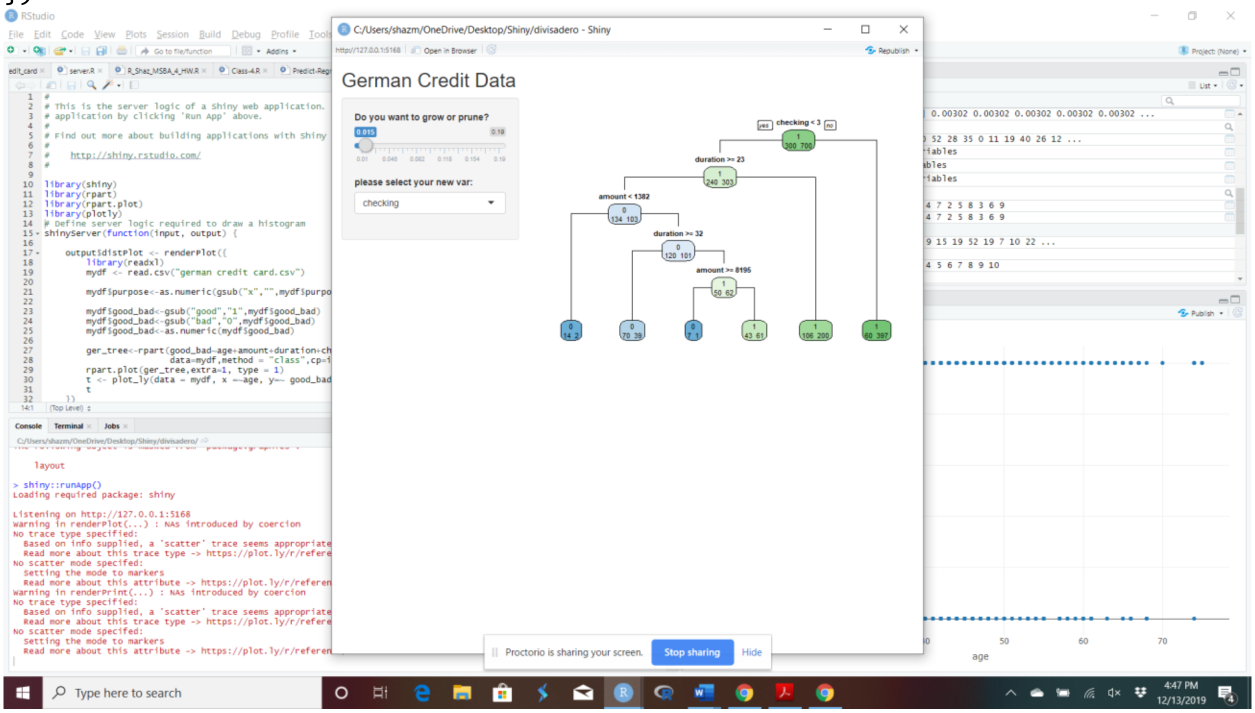
Ans 4a:

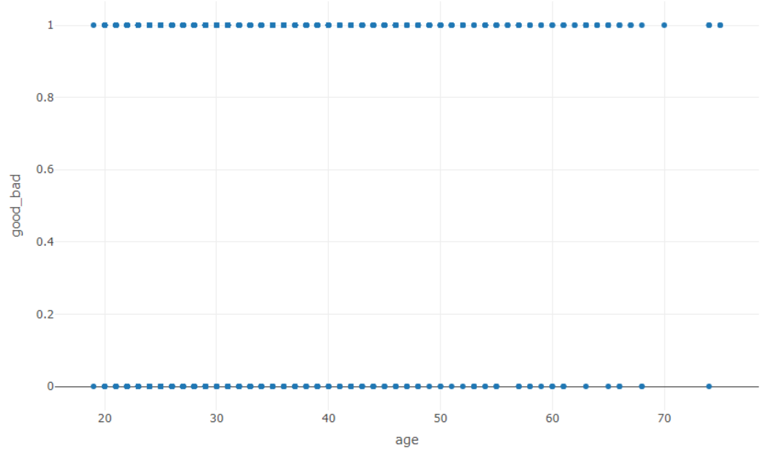
Output

Server code

```
#
# This is the server logic of a Shiny web application. You can run the
# application by clicking 'Run App' above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)
library(rpart)
library(rpart.plot)
library(plotly)
# Define server logic required to draw a histogram
shinyServer(function(input, output) {

    output$distPlot <- renderPlot({
        library(readxl)
        mydf <- read.csv("german credit card.csv")

        mydf$purpose<-as.numeric(gsub("x","",mydf$purpose))

        mydf$good_bad<-gsub("good","1",mydf$good_bad)
        mydf$good_bad<-gsub("bad","0",mydf$good_bad)
        mydf$good_bad<-as.numeric(mydf$good_bad)

        ger_tree<-rpart(good_bad~age+amount+duration+checking,
                        data=mydf,method = "class",cp=input$bins)
        rpart.plot(ger_tree,extra=1, type = 1)
        t <- plot_ly(data = mydf, x =~age, y=~ good_bad)
        t
    })
    output$logit<- renderPrint({
        library(readxl)
        mydf <-  read.csv("german credit card.csv")

        mydf$purpose<-as.numeric(gsub("x","",mydf$purpose))

        mydf$good_bad<-gsub("good","1",mydf$good_bad)
        mydf$good_bad<-gsub("bad","0",mydf$good_bad)
```

```
        mydf$good_bad<-as.numeric(mydf$good_bad)
        my_logit<-glm(paste("good_bad~age+amount+duration+",input$myvar),
                      data=mydf,family = "binomial")
        summary(my_logit)
        t <- plot_ly(data = mydf, x =~age, y=~ good_bad)
        t
    })
})
```



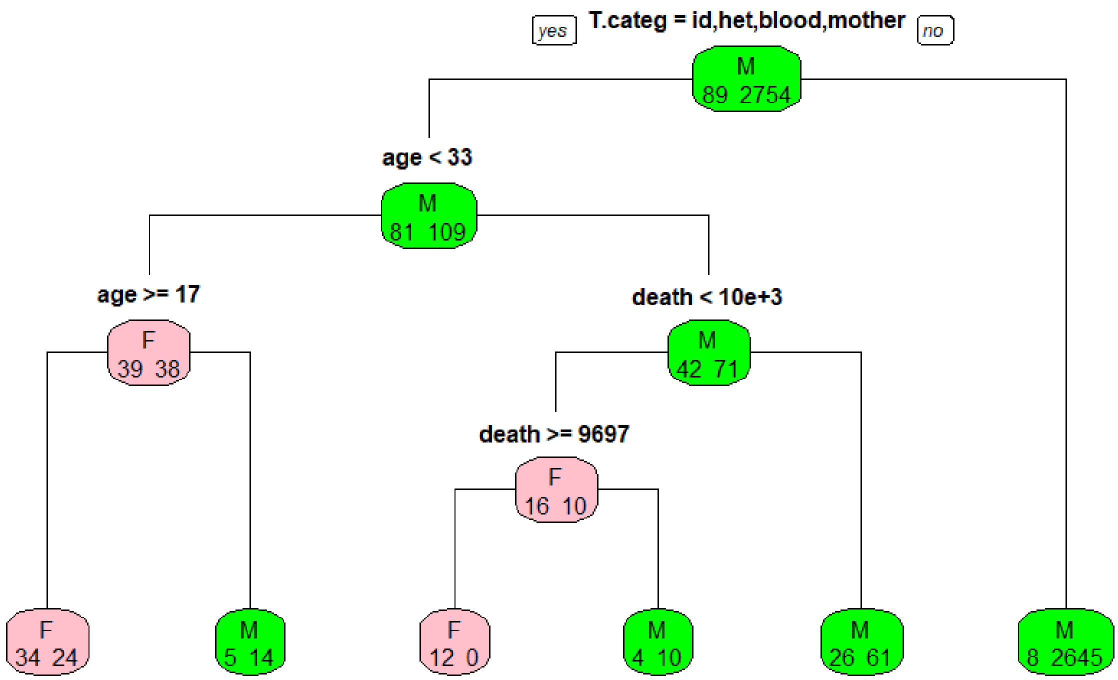


Ans 4b:
Server Code #

```
# This is the server logic of a Shiny web application. You can run the
# application by clicking 'Run App' above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#
```

```
library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

    output$distPlot <- renderPlot({

        # generate bins based on input$bins from ui.R
        library(MASS)
        library(rpart)
        library(rpart.plot)
        library(plotly)
        library(ROCR)
        summary(Aids2)
        as.data.frame(Aids2)
        aids=Aids2
        2843*0.7
        aids$status<-gsub(aids$status,pattern='A',replacement=1)
        aids$status<-gsub(aids$status,pattern='D',replacement=0)

        train_index <- sample.split(aids, SplitRatio=0.7)
        train <- aids[ train_index,]
        test <- aids[!train_index,]
        gc_train <- train
        gc_test <- test
        my_mod <- glm(age~death+T.categ+sex, data=gc_train, family=poisson)
        my_mod

        predict_logit <-predict(my_mod, gc_train, type='response')
        predict_logit

        my_tree <- rpart(sex~age+T.categ+death, data=aids,method="class",cont
rol=rpart.control(cp=0.033))
        rpart.plot(my_tree,type=1,extra=1, box.palette = c("pink","green"))
        plotcp(my_tree)
    })
    output$logit<- renderPrint({
        library(MASS)
        library(rpart)
        library(rpart.plot)
        library(plotly)
        library(ROCR)
        summary(Aids2)
        as.data.frame(Aids2)
        aids=Aids2
        2843*0.7
        aids$status<-gsub(aids$status,pattern='A',replacement=1)
        aids$status<-gsub(aids$status,pattern='D',replacement=0)

        train_index <- sample.split(aids, SplitRatio=0.7)
        train <- aids[ train_index,]
        test <- aids[!train_index,]
        gc_train <- train
        gc_test <- test
        my_mod <- glm(age~death+T.categ+sex, data=gc_train, family=poisson)
        my_mod

        predict_logit <-predict(my_mod, gc_train, type='response')
        predict_logit

        my_tree <- rpart(sex~age+T.categ+death, data=aids,method="class",cont
rol=rpart.control(cp=0.033))
        rpart.plot(my_tree,type=1,extra=1, box.palette = c("pink","green"))
        plotcp(my_tree)
```

```
    })
})
```



Q5

Ans 5a

Code:

```
my_function <- function(net_income,total_assets,shareholders_equity,net_sales
){
  ROA <- net_income/total_assets
  ROE <- net_income/shareholders_equity
  profit_margin <- net_income/net_sales
  ratios <- c(ROA,ROE,profit_margin)
  names <- c("ROA","ROE","Profit Margin")
  final <- cbind(ratios,names)
  print(final)

}
#Company 1
my_function(10000,40000,10000,50000)

#company 2
my_function(8000,45000,20000,35000)

 #company 3
my_function(20000,60000,40000,70000)

 #company 4
my_function(2000,10000,8000,4000)
```

Output
#company 1

```
my_function(10000,40000,10000,50000)
     ratios names
[1,] "0.25" "ROA"
[2,] "1"    "ROE"
[3,] "0.2"  "Profit Margin"
>
> #company 2
> my_function(8000,45000,20000,35000)
     ratios              names
[1,] "0.177777777777778" "ROA"
[2,] "0.4"               "ROE"
[3,] "0.228571428571429" "Profit Margin"
>
>  #company 3
> my_function(20000,60000,40000,70000)
     ratios              names
[1,] "0.333333333333333" "ROA"
[2,] "0.5"               "ROE"
[3,] "0.285714285714286" "Profit Margin"
>
>  #company 4
> my_function(2000,10000,8000,4000)
     ratios names
[1,] "0.2"  "ROA"
[2,] "0.25" "ROE"
[3,] "0.5"  "Profit Margin"
```

Q5B
ANS 5B

Code:

```
my_function <- function(net_income,total_assets,shareholders_equity,net_sales
){
  ROA <- net_income/total_assets
  ROE <- net_income/shareholders_equity
  profit_margin <- net_income/net_sales
  ratios <- c(ROA,ROE,profit_margin)
  names <- c("ROA","ROE","Profit Margin")
  final <- cbind(ratios,names)
  print(final)

}
#Company 1
my_function(10000,40000,10000,50000)

#company 2
my_function(8000,45000,20000,35000)

 #company 3
my_function(20000,60000,40000,70000)

 #company 4
my_function(2000,10000,8000,4000)

Avg_roa= (0.177+0.333+0.2+0.25)/4
Avg_roa

#company weighted roa
ROA1=Avg_roa*0.25
ROA1
ROA2=Avg_roa*0.177
ROA2
```

```
ROA3=Avg_roa*0.333
ROA3
ROA4=Avg_roa*0.2
ROA4
```

Output

```
ROA1=Avg_roa*0.25
> ROA1
[1] 0.06
> ROA2=Avg_roa*0.177
> ROA2
[1] 0.04248
> ROA3=Avg_roa*0.333
> ROA3
[1] 0.07992
> ROA4=Avg_roa*0.2
> ROA4
[1] 0.048
```

Q5d
Ans 5d

The ROA Tells us if the company has optimized its assets to drive value. The higher the ROA the better the companys performance . Weather a company is doing well or not depends on the ROA vs the industry standard since this isn't know its not easy to tell how the company is faring in the industry.
The Return on equity tells us if we are going to make our money back if we invest in the company. A company would strive to achieve a better return on equity. But since the industry standard is not known it is difficult to figure out which firm to invest in based on ROE
Profit margin tells us the profit the firm has earned vs costs. The higher the ratio the better.
In conclusion it is easy to see that in between these 4 companies company 3 has been doing better hence I would invest in company 3.

Q5E
Ans 5e

Code

```
#Investing 10% from comapany 1 into company 4
my_function(3000,14000,9000,9000)
```

Output:

```
      ratios              names
 [1,] "0.214285714285714" "ROA"
 [2,] "0.333333333333333" "ROE"
 [3,] "0.333333333333333" "Profit Margin"

 >
```

As we can see that ROA AND ROE of company 4 increases form such an investment but its profit margin drops quite a bit.

Q1

Ans1a)

Code:

```
barto <- rexp(1000,rate=0.3)
barto
```

Output:
```
[1]  1.514128384  2.442114115  0.086278031  4.660980403  0.667792610  2.06980
5072  0.719387551  1.876248484
   [9]  3.825787871  0.225954320  6.657387037  0.659825224  2.336584789  2.91
9798481  1.070630719  3.464839956
  [17]  1.056264696  1.679832808  4.522413841  0.334369801  4.009987588  1.23
1910288  2.313090206  0.503832835
  [25]  2.199534969  3.215783255  2.736417966  4.107485162  1.414347352  0.33
7204933  2.284926310  0.509329541
  [33]  7.571780920  1.619301016  4.497634098  1.064059289  0.728062259  3.17
4738840  1.379020219  2.733223089
  [41]  3.438464505  4.843509075  1.786017818 17.902562928  0.151506290  7.45
1834150  2.888688594  7.445912230
  [49] 11.109537829  6.221050216 10.643293907  6.230059527  1.486908801  0.91
9346235  1.002389531  0.092402954
  [57]  1.096363697  0.915655123  3.586564784  0.203761771  1.801614358  1.67
3699513  1.786267599  8.088963366
  [65] 12.064868669  3.856239486  2.262524120  2.910845920  4.808643609  2.39
2932810  9.608325664  1.868552306
```

Ans1b:

Code:
```
barto <- rexp(1000,rate=0.3)
barto
mean(barto)
var(barto)
```

Output:
```
mean(barto)
[1] 3.218099
> var(barto)
[1] 9.969917
```

Ans1c

Code:
```
a)
exp(0.3-barto*3)
mean(barto)
```

Output:
```
1] 1.437327e-02 8.881745e-04 1.042025e+00 1.141758e-06 1.820675e-01 2.713773e
-03 1.559589e-01 4.850151e-03
   [9] 1.398726e-05 6.853244e-01 2.860810e-09 1.864717e-01 1.218964e-03 2.119
001e-04 5.437275e-02 4.130529e-05
  [17] 5.676734e-02 8.743030e-03 1.730255e-06 4.950436e-01 8.048999e-06 3.351
605e-02 1.307981e-03 2.977508e-01
  [25] 1.838868e-03 8.719620e-05 3.673285e-04 6.007779e-06 1.938914e-02 4.908
509e-01 1.423297e-03 2.928811e-01
```