30155843_Maryam Mahmoodi_Assessment_2_08042019

April 8, 2019

1 Assessment 2

Student Name: **Maryam Mahmoodi**
Student Number: **30155843**

2 Question 1 Bayes' theorem

In this question, you will apply Bayes' Theorem to calculate the probability that a patient has lung cancer given the chest x-ray result is positive. The probabilities are estimated from a synthetic dataset, hence do not represent the actual probabilities.

Assuming the probability of having lung cancer is 2.95%. If a patient has lung cancer, then the chest x-ray test has 96% chance of getting positive result. On the other hand, if a patient does not have lung cancer, then the x-ray has 8% chance of showing positive result. What is the probability that a patient has lung cancer given that the x-ray result is positive?

Total marks 10. If numerator is calculated correctly 5 marks. If denominator is calculated correctly 5 marks.

Law of Total Probability Theorem

$$Pr[x = 1] = Pr[x = 1 \cap c = 0] + Pr[x = 1 \cap c = 1]$$

$$Pr[x = 1] = Pr[x = 1 | c = 0] \times Pr[c = 0] + Pr[x = 1 | c = 1] \times Pr[c = 1]$$

```
In [1]: p_c1 = 0.0295
        p_x1_c1 = 0.96
        p_x1_c0 = 0.08
        p_c0 = 1 - p_c1
        p_x1 = p_x1_c1 * p_c1 + p_x1_c0 * p_c0
        p_x1
```

```
0.10596
```

3 Question 2 Maximum likelihood estimation

A popular activation function for building an Aritifical Neural Network (ANN) is the hyperbolic tangent function (tanh)

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

that takes values on the real line and ranges from -1 to 1, where $z = \sum_{i=0}^{n} x_i \beta_i$. In this questions, you will be asked to drive an equation for the maximum likelihood estimation of the tanh parameters.

Total marks 15. Derivation of ML estimation 7 marks. Implementation of the ML estimation 8 marks.

```
In [2]: # Data Generation
        # this block of code is for generating test data that is used to confirm
        # your derivation and implementation of the ML estimation
        set.seed(836217)
        n = 1000
        X = cbind(1, rnorm(n), rnorm(n))
        Y = rep(-1, n)
        beta = c(0.43, -1.7, 2.8)
        for (i in 1:n) {
          z = (beta %*% X[i, ])[1]
          p = pnorm(z)
          if (runif(1) < p) Y[i] = 1
        }

        XTest = matrix(c(1, 1.0162462, -0.4012721, 1, -2.0693534, -0.2018362,
                         1, -0.4812785, 0.2505587, 1, 1.1538629, 0.3152341,
                         1, 0.4399999, 0.8282703), 5, 3, byrow = T)
        YTest = c(-1, 1, 1, -1, 1)
```

a) Derive the ML estimation of the tanh parameters.

$$L(g(z|\beta)) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Differentiating the above equation and equating it to zero,

$$\frac{\partial L(g(z|\beta))}{\partial z} = \frac{4e^{2z}}{(e^z + e^{-z})^2} \frac{\partial z}{\partial \beta} = 0$$

Or,

$$\frac{\partial z}{\partial \beta} = 0$$

b) Implement the equation that you derived in part (a). You can test your implementation on the test data **XTest** and **YTest** provided above with $\vec{\beta} = (0.43, -1.7, 2.8)$. If done correctly, you should get the negative log (base e) likelihood equal to 0.2886433. Optimize the parameters using any numerical method available in R e.g., the **optim()** function.

```
In [3]: tanh <- function(z){
            return (exp(z)-exp(-z))/(exp(z)+exp(-z))
        }
```

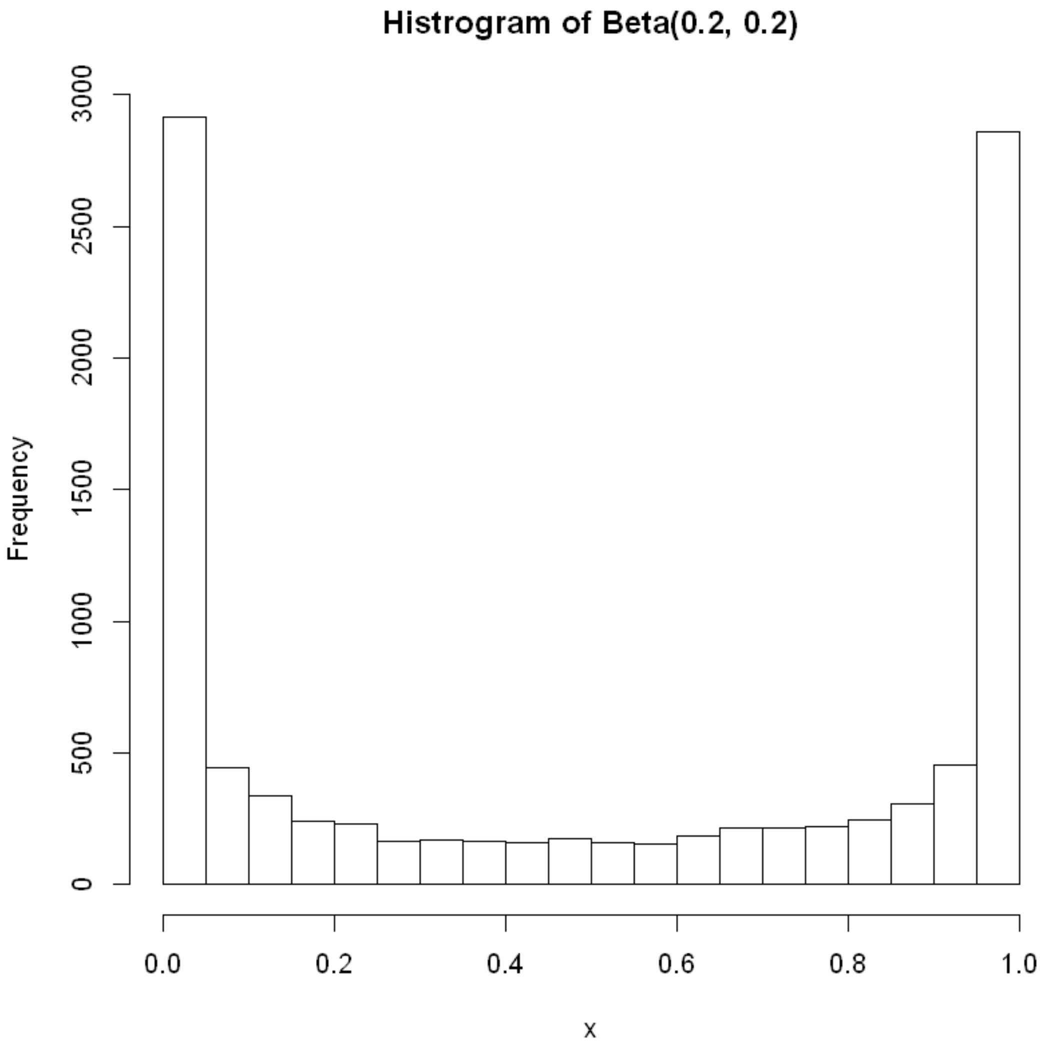
4 Question 3 Central limit theorem

In this question, you will be asked to sample from a Beta distribution, then calculate and plot the sample mean in histogram to visualize the central limit theorem.

Total marks 15. Part a 2 marks, part b plot the sample means 10 marks, explain findings 3 marks.

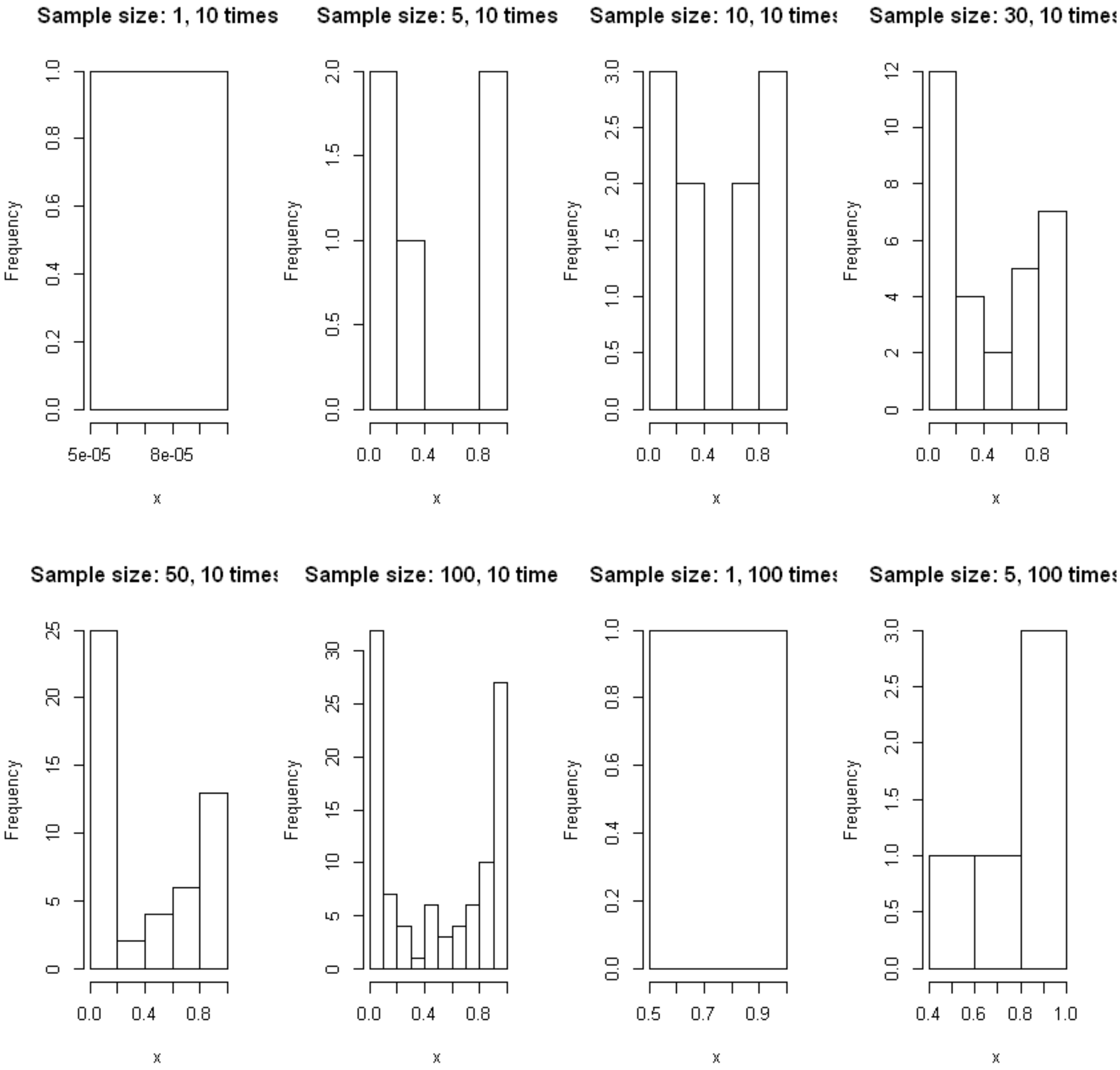
a) Sample 10000 data from a **Beta(0.2, 0.2)** distribution. Plot the samples in a histogram.

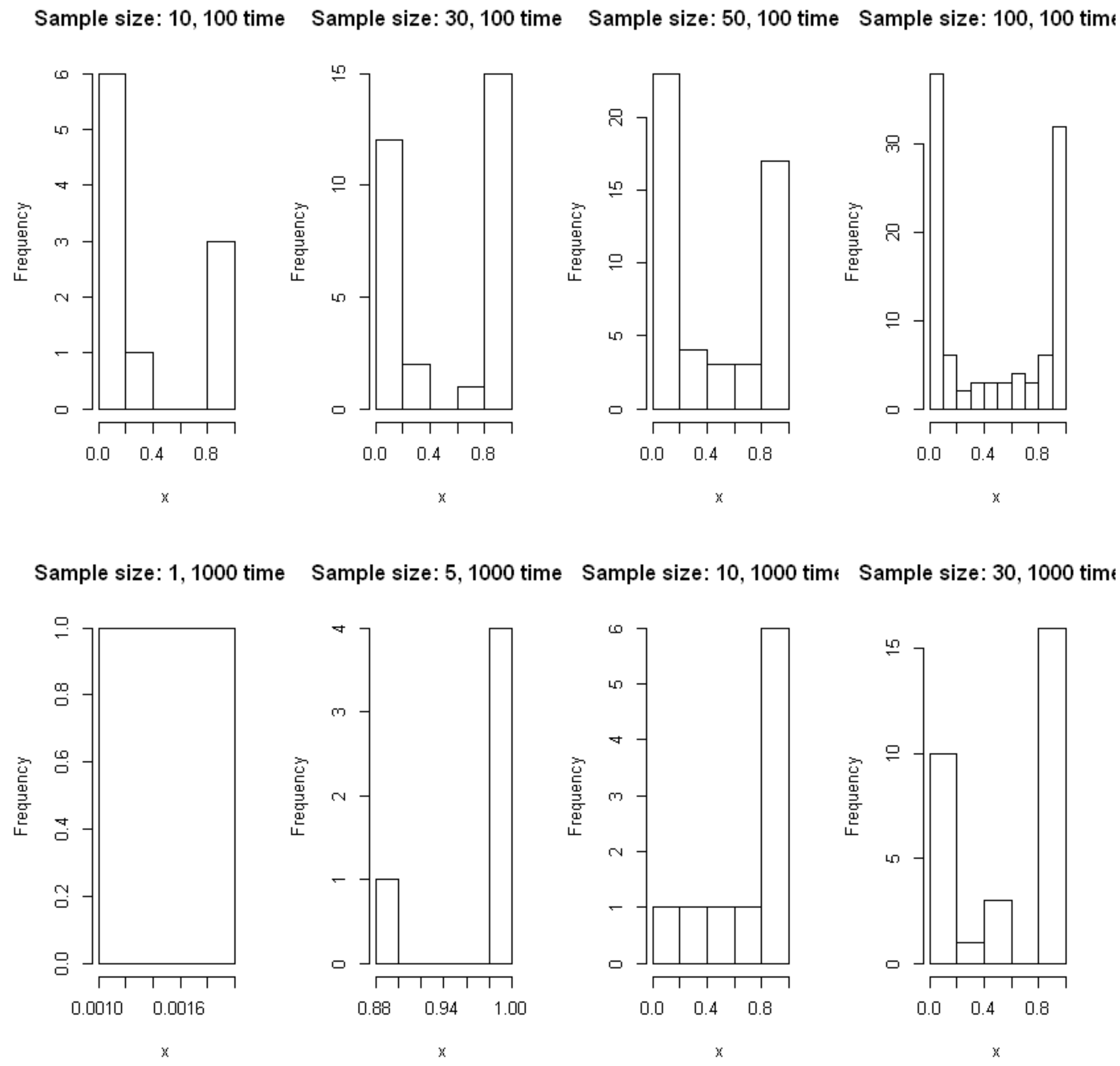
```
In [4]: x <- rbeta(10000, 0.2, 0.2)
        hist(x,
            main='Histrogram of Beta(0.2, 0.2)')
```

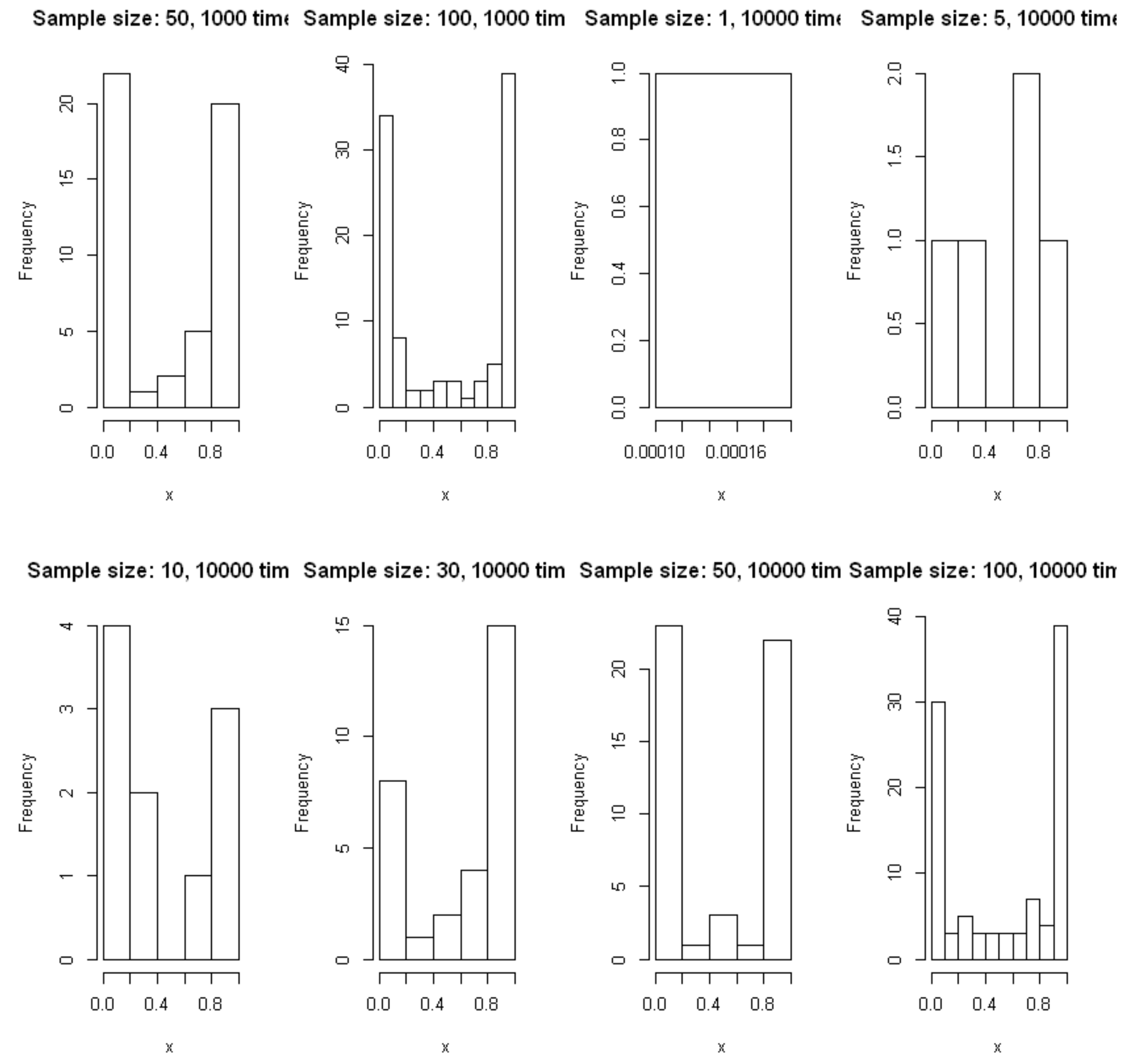


b) Sample from the same Beta distribution above. Repeat the sampling process for 10, 100, 1000, 10000 times. Each time draw 1, 5, 10, 30, 50, 100 samples. Calculate and plot the sample means in histograms. Explain your findings.

```
In [5]: par(mfrow=c(2,4))
        times <- c(10, 100, 1000, 10000)
        samples <- c(1, 5, 10, 30, 50, 100)
        for (tm in times) {
            for (smple in samples) {
                means <- rep(0, tm)
                for (index in tm) {
                    x <- rbeta(smple, 0.2, 0.2)
                }
                hist(x,
                     main=paste('Sample size: ',
                                smple,
                                ', ',
                                tm,
                                ' times',
                                sep = ''))
            }
        }
```


Sample size: 1, 10 times
Sample size: 5, 10 times
Sample size: 10, 10 times
Sample size: 30, 10 times
Sample size: 50, 10 times
Sample size: 100, 10 time
Sample size: 1, 100 times
Sample size: 5, 100 times
Frequency
x
5e-05
8e-05
0.0
0.2
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.5
2.0
2.5
3.0
0
2
4
5
6
8
10
12
15
20
25
30


Sample size: 10, 100 time
Sample size: 30, 100 time
Sample size: 50, 100 time
Sample size: 100, 100 tim
Frequency
x
0.0
0.4
0.8
0
1
2
3
4
5
6
0
5
10
15
0
5
10
15
20
0
10
20
30
Sample size: 1, 1000 time
Sample size: 5, 1000 time
Sample size: 10, 1000 tim
Sample size: 30, 1000 tim
0.0
0.2
0.4
0.6
0.8
1.0
0.0010
0.0016
0.88
0.94
1.00
0
1
2
3
4
0
1
2
3
4
5
6
0
5
10
15



As we are increasing the number of sample size and how many times we are simulating the data to get the means, the distribution of the sample means is reaching close to beta distribution

5 Question 4 hypothesis test

You are provided with a data set called **iData.csv** containing information on births recorded in a particular state.

Total marks 15. Each sub-question a, b, c worths 5 marks.

a) It is claimed that mothers' smoking habit does not have any effect on the birth weight of the babies. To verity this claim, you need to complete the following tasks:

1. Construct a significance test, by writing a null hypothesis, and an alternative hypothesis,
2. perform the test using an appropriate R function,

3. Make a conclusion and explain your findings.

```
In [6]: data <- read.csv('iData.csv')
        head(data)
```

fage	mage	mature	weeks	premie	visits	marital	gained	weight	lowbirthweight	g
19	15	younger mom	37	full term	11	married	38	6.63	not low	f
21	15	younger mom	41	full term	6	married	34	8.00	not low	n
18	15	younger mom	37	full term	12	married	76	8.44	not low	n
17	15	younger mom	35	premie	5	married	15	4.69	low	n
20	16	younger mom	37	full term	13	married	52	6.94	not low	f
30	16	younger mom	45	full term	9	married	28	7.44	not low	n

```
In [7]: t.test(weight ~ habit, data = data)
```

```
Welch Two Sample t-test

data:  weight by habit
t = 2.3625, df = 108.54, p-value = 0.01994
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.05801978 0.66264264
sample estimates:
mean in group nonsmoker    mean in group smoker
               7.246760                6.886429
```

We see from this output that the difference is highly significant. Also from the 95 percent confidence interval we can observe that the value 0 is not in the range and therefore, we reject the null hypothesis.

b) Calculate a 99% confidence interval for the average length of pregnancies.

```
In [8]: weeks <- data$weeks
        t.test(weeks, conf.level = 0.99)$conf.int
```

1. 38.216112716835 2. 38.718887283165

c) Conduct a hypothesis test evaluating whether the average weight gained is different from 30. Make a conclusion and explain your findings.

```
In [9]: mother.weight.gained <- data$gained
        t.test(mother.weight.gained, mu=30, conf.level = 0.95)
```

```
One Sample t-test

data:  mother.weight.gained
```

```
t = 1.5538, df = 799, p-value = 0.1206
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 29.7966 31.7484
sample estimates:
mean of x
  30.7725
```

From the output, we can see that the mean mother's weight gaines for the sample is 30.7725. The two-sided 99% confidence interval tells us that mean filling weight is likely to be between 29.7966 and 31.7484. The p-value of 0.1206 tells us that if the mean of the weight gained were 30 the probability of selecting a sample with a mean weight gained other than this one would be approximately 12.06%.

6 Question 5 prediction

The German Credit Card data set is for customers of a financial institution who have been labelled as "good"" or "bad credit risks (in fact: whether they repaid the loan or did not repay the loan). The data set contains 21 attributes (20 predictors and 1 dependent variable) and 1000 instances, with no missing values (the data are real, but were cleaned up prior to being put into the archive). The specification of these attributes is given in the document **german.doc**.

In this question, you will build two classification models using logistic regression and naive Bayes, and report their accuracies. Use the first 800 rows as the training data and the rest of the data for testing.

Total marks 30. Part a model building 5 marks, report accuracies and explain finding 5 marks. Part b variable selection with reasonable explainations 5 marks, model building 5 marks, report accuracies and compare with the model accuracies in part a 5 marks. Part c 5 marks.

a) Build a logistic regression and a naive Bayes models using all predictors. Report their accuracies in terms of precision, recall and F1 score using the test data. Compare the two models and explain your findings.

```
In [10]: german_credit <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/s
         head(german_credit)
```

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V12	V13	V14	V15	V16	V17
A11	6	A34	A43	1169	A65	A75	4	A93	A101	...	A121	67	A143	A152	2	A17
A12	48	A32	A43	5951	A61	A73	2	A92	A101	...	A121	22	A143	A152	1	A17
A14	12	A34	A46	2096	A61	A74	2	A93	A101	...	A121	49	A143	A152	1	A17
A11	42	A32	A42	7882	A61	A74	2	A93	A103	...	A122	45	A143	A153	1	A17
A11	24	A33	A40	4870	A61	A73	3	A93	A101	...	A124	53	A143	A153	2	A17
A14	36	A32	A46	9055	A65	A73	2	A93	A101	...	A124	35	A143	A153	1	A17

```
In [11]: colnames(german_credit) <- c("chk_acct",
                                      "duration",
                                      "credit_his",
                                      "purpose",
                                      "amount",
                                      "saving_acct",
                                      "present_emp",
                                      "installment_rate",
                                      "sex",
                                      "other_debtor",
                                      "present_resid",
                                      "property",
                                      "age",
                                      "other_install",
                                      "housing",
                                      "n_credits",
                                      "job",
                                      "n_people",
                                      "telephone",
                                      "foreign",
                                      "response")

         german_credit$response <- german_credit$response - 1
         german_credit$response <- as.factor(german_credit$response)
```

Splitting the data into 80:20, train test split using stratifiesd sampling in order to get equal amount of data from each response class

```
In [12]: library(caret)
         set.seed(2018)
         in.train <- createDataPartition(as.factor(german_credit$response),
                                         p=0.8,
                                         list=FALSE)
         german_credit.train <- german_credit[in.train,]
         german_credit.test <- german_credit[-in.train,]
```

```
Warning message:
"package 'caret' was built under R version 3.4.4"Loading required package: lattice
Loading required package: ggplot2
Warning message:
"package 'ggplot2' was built under R version 3.4.4"
```

Logistic Regression

```
In [13]: credit.glm0 <- glm(response ~ ., family = binomial, german_credit.train)
         credit.glm.step <- step(credit.glm0, direction = "backward")
```

```
Start:  AIC=804.63
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + present_emp + installment_rate + sex + other_debtor +
    present_resid + property + age + other_install + housing +
    n_credits + job + n_people + telephone + foreign
```

```
                   Df Deviance    AIC
- job               3   707.46 799.46
- property          3   708.24 800.24
- present_resid     1   706.67 802.67
- age               1   707.21 803.21
- housing           2   709.36 803.36
- present_emp       4   713.67 803.67
- n_credits         1   707.92 803.92
- n_people          1   708.32 804.32
<none>                  706.63 804.63
- telephone         1   709.77 805.77
- other_install     2   712.20 806.20
- sex               3   714.74 806.74
- foreign           1   712.64 808.64
- duration          1   712.87 808.87
- purpose           9   729.78 809.78
- amount            1   713.93 809.93
- credit_his        4   721.70 811.70
- other_debtor      2   717.84 811.84
- installment_rate  1   717.60 813.60
- saving_acct       4   727.24 817.24
- chk_acct          3   762.97 854.97

Step:  AIC=799.46
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + present_emp + installment_rate + sex + other_debtor +
    present_resid + property + age + other_install + housing +
    n_credits + n_people + telephone + foreign

                   Df Deviance    AIC
- property          3   709.12 795.12
- present_resid     1   707.48 797.48
- present_emp       4   713.92 797.92
- housing           2   710.07 798.07
- age               1   708.14 798.14
- n_credits         1   708.64 798.64
- n_people          1   709.17 799.17
<none>                  707.46 799.46
- telephone         1   710.75 800.75
- other_install     2   713.24 801.24
- sex               3   715.56 801.56
- foreign           1   713.59 803.59
- duration          1   713.90 803.90
- purpose           9   730.51 804.51
- amount            1   715.26 805.26
- credit_his        4   722.22 806.22
- other_debtor      2   718.74 806.74
- installment_rate  1   719.01 809.01
- saving_acct       4   728.41 812.41
- chk_acct          3   763.61 849.61

Step:  AIC=795.12
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + present_emp + installment_rate + sex + other_debtor +
    present_resid + age + other_install + housing + n_credits +
    n_people + telephone + foreign

                   Df Deviance    AIC
- present_resid     1   709.13 793.13
- present_emp       4   715.49 793.49
- age               1   710.00 794.00
- housing           2   712.07 794.07
- n_credits         1   710.18 794.18
- n_people          1   710.68 794.68
<none>                  709.12 795.12
- telephone         1   711.95 795.95
- sex               3   717.08 797.08
- other_install     2   715.23 797.23
- foreign           1   715.42 799.42
- duration          1   716.35 800.35
- purpose           9   732.89 800.89
- amount            1   717.70 801.70
- credit_his        4   724.27 802.27
- other_debtor      2   721.25 803.25
- installment_rate  1   721.09 805.09
- saving_acct       4   730.03 808.03
- chk_acct          3   766.61 846.61

Step:  AIC=793.13
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + present_emp + installment_rate + sex + other_debtor +
    age + other_install + housing + n_credits + n_people + telephone +
    foreign

                   Df Deviance    AIC
- present_emp       4   715.67 791.67
- age               1   710.04 792.04
- n_credits         1   710.18 792.18
- housing           2   712.21 792.21
- n_people          1   710.69 792.69
<none>                  709.13 793.13
- telephone         1   711.98 793.98
- sex               3   717.08 795.08
- other_install     2   715.29 795.29
- foreign           1   715.42 797.42
- duration          1   716.37 798.37
- purpose           9   732.98 798.98
- amount            1   717.73 799.73
- credit_his        4   724.30 800.30
- other_debtor      2   721.26 801.26
- installment_rate  1   721.09 803.09
- saving_acct       4   730.20 806.20
- chk_acct          3   766.86 844.86

Step:  AIC=791.67
response ~ chk_acct + duration + credit_his + purpose + amount + 
    saving_acct + installment_rate + sex + other_debtor + age + 
    other_install + housing + n_credits + n_people + telephone + 
    foreign

                   Df Deviance    AIC
- n_credits         1   716.57 790.57
- age               1   716.76 790.76
- n_people          1   716.98 790.98
- housing           2   719.07 791.07
<none>                  715.67 791.67
- telephone         1   718.98 792.98
- other_install     2   722.14 794.14
- duration          1   721.64 795.64
- foreign           1   721.88 795.88
- sex               3   726.00 796.00
- purpose           9   739.81 797.81
- amount            1   724.71 798.71
- credit_his        4   731.72 799.72
- other_debtor      2   728.27 800.27
- installment_rate  1   727.87 801.87
- saving_acct       4   738.83 806.83
- chk_acct          3   774.25 844.25

Step:  AIC=790.57
response ~ chk_acct + duration + credit_his + purpose + amount + 
    saving_acct + installment_rate + sex + other_debtor + age + 
    other_install + housing + n_people + telephone + foreign

                   Df Deviance    AIC
- age               1   717.51 789.51
- n_people          1   718.00 790.00
- housing           2   720.06 790.06
<none>                  716.57 790.57
- telephone         1   719.63 791.63
- other_install     2   723.65 793.65
- duration          1   722.35 794.35
- sex               3   726.70 794.70
- foreign           1   723.05 795.05
- purpose           9   741.14 797.14
- amount            1   725.73 797.73
- credit_his        4   732.11 798.11
- other_debtor      2   729.22 799.22
- installment_rate  1   728.61 800.61
- saving_acct       4   739.90 805.90
- chk_acct          3   774.99 842.99

Step:  AIC=789.51
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + installment_rate + sex + other_debtor + other_install +
    housing + n_people + telephone + foreign

                   Df Deviance    AIC
- n_people          1   718.84 788.84
- housing           2   720.94 788.94
<none>                  717.51 789.51
- telephone         1   721.14 791.14
- other_install     2   724.36 792.36
- sex               3   727.64 793.64
- duration          1   723.75 793.75
- foreign           1   723.94 793.94
- purpose           9   741.76 795.76
- amount            1   726.84 796.84
- credit_his        4   733.73 797.73
- other_debtor      2   730.28 798.28
- installment_rate  1   729.20 799.20
- saving_acct       4   741.53 805.53
- chk_acct          3   776.53 842.53

Step:  AIC=788.84
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + installment_rate + sex + other_debtor + other_install +
    housing + telephone + foreign

                   Df Deviance    AIC
- housing           2   722.33 788.33
<none>                  718.84 788.84
- telephone         1   722.70 790.70
- other_install     2   725.69 791.69
- sex               3   727.75 791.75
- duration          1   724.89 792.89
- foreign           1   725.14 793.14
- purpose           9   743.61 795.61
- amount            1   728.04 796.04
- other_debtor      2   731.40 797.40
- credit_his        4   735.83 797.83
- installment_rate  1   730.10 798.10
- saving_acct       4   742.58 804.58
- chk_acct          3   777.32 841.32

Step:  AIC=788.33
response ~ chk_acct + duration + credit_his + purpose + amount +
    saving_acct + installment_rate + sex + other_debtor + other_install +
    telephone + foreign

                   Df Deviance    AIC
<none>                  722.33 788.33
- telephone         1   726.20 790.20
- other_install     2   728.67 790.67
- foreign           1   728.66 792.66
- duration          1   728.67 792.67
- sex               3   732.97 792.97
- purpose           9   746.49 794.49
- amount            1   731.70 795.70
- other_debtor      2   735.27 797.27
- installment_rate  1   733.34 797.34
- credit_his        4   740.43 798.43
- saving_acct       4   746.10 804.10
- chk_acct          3   784.17 844.17
```

```
In [14]: prob.insample <- predict(credit.glm.step, type = "response")
         predicted.insample <- prob.insample > 0.1667
         predicted.insample <- as.numeric(predicted.insample)
         mean(ifelse(german_credit.train$response != predicted.insample, 1, 0))
```

0.35125

```
In [15]: table(german_credit.train$response,
               predicted.insample,
               dnn = c("Truth", "Predicted"))
```

```
     Predicted
Truth   0   1
    0 304 256
    1  25 215
```

7 Naive Bayes

```
In [16]: library(e1071)
         naive.bayes <- naiveBayes(response ~ .,
                                   family = binomial,
                                   german_credit.train)
         pred.incall <- predict(naive.bayes,
                                newdata = german_credit.train)
         table(german_credit.train$response,
               pred.incall,
               dnn = c("Truth", "Predicted"))
```

```
Warning message:
"package 'e1071' was built under R version 3.4.4"

     Predicted
Truth   0   1
    0 486  74
    1 111 129
```

```
In [17]: pred <- predict(naive.bayes,
                                newdata = german_credit.test)
         table(german_credit.test$response,
               pred,
               dnn = c("Truth", "Predicted"))
```

```
     Predicted
Truth   0   1
    0 118  22
    1  26  34
```

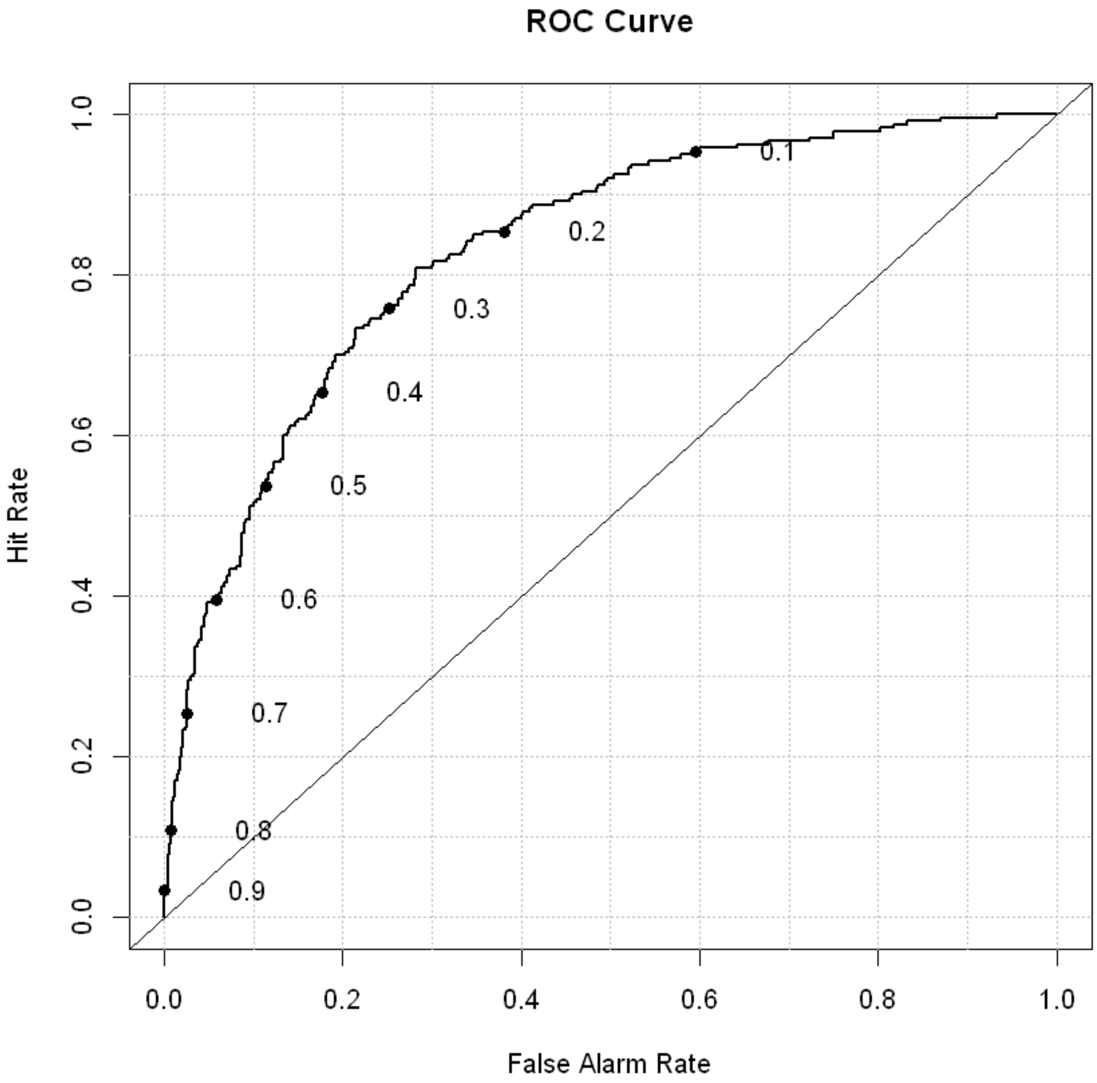
Logistic Model ROC Plot
ROC Plot is plotted below and the AUC is 0.831436011904762

```
In [18]: library(verification)
         roc.plot(german_credit.train$response == "1", prob.insample)
         roc.plot(german_credit.train$response == "1", prob.insample)$roc.vol$Area
```

```
Warning message:
"package 'verification' was built under R version 3.4.4"Loading required package: fields
Warning message:
"package 'fields' was built under R version 3.4.4"Loading required package: spam
Warning message:
"package 'spam' was built under R version 3.4.4"Loading required package: dotCall64
Warning message:
"package 'dotCall64' was built under R version 3.4.4"Loading required package: grid
Spam version 2.2-2 (2019-03-07) is loaded.
Type 'help( Spam)' or 'demo( spam)' for a short introduction
and overview of this package.
Help for individual functions is also obtained by adding the
suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
Attaching package: 'spam'

The following objects are masked from 'package:base':

    backsolve, forwardsolve

Loading required package: maps
Warning message:
"package 'maps' was built under R version 3.4.4"See www.image.ucar.edu/~nychka/Fields for
 a vignette and other supplements.
Loading required package: boot

Attaching package: 'boot'

The following object is masked from 'package:lattice':

    melanoma

Loading required package: CircStats
Warning message:
"package 'CircStats' was built under R version 3.4.4"Loading required package: MASS
Loading required package: dtw
Warning message:
"package 'dtw' was built under R version 3.4.4"Loading required package: proxy
Warning message:
"package 'proxy' was built under R version 3.4.4"
Attaching package: 'proxy'

The following object is masked from 'package:spam':

    as.matrix

The following objects are masked from 'package:stats':

    as.dist, dist

The following object is masked from 'package:base':

    as.matrix

Loaded dtw v1.20-1. See ?dtw for help, citation("dtw") for use in publication.
```

0.831436011904762



b) Based on the summaries of the two models built in (1), select a subset of variables that could achieve the same or better accuracies than the same model built above using all predictors. Report the new models' accuracis in terms of precision, recall and F1 socre.

Final model for GLM

```
In [19]: credit.glm.final <- glm(response ~ chk_acct + duration +
                                   credit_his + amount +
                                   saving_acct  +
                                   other_install + installment_rate,
                                family = binomial, german_credit.train)
```

In-sample misclassification rate

```
In [20]: prob.glm1.insample <- predict(credit.glm.final, type = "response")
         predicted.glm1.insample <- prob.glm1.insample > 0.1667
         predicted.glm1.insample <- as.numeric(predicted.glm1.insample)
         mean(ifelse(german_credit.train$response != predicted.glm1.insample, 1, 0))
```

0.385

c) Plot a ROC curve for each of the models built in (2) and report its AUC.

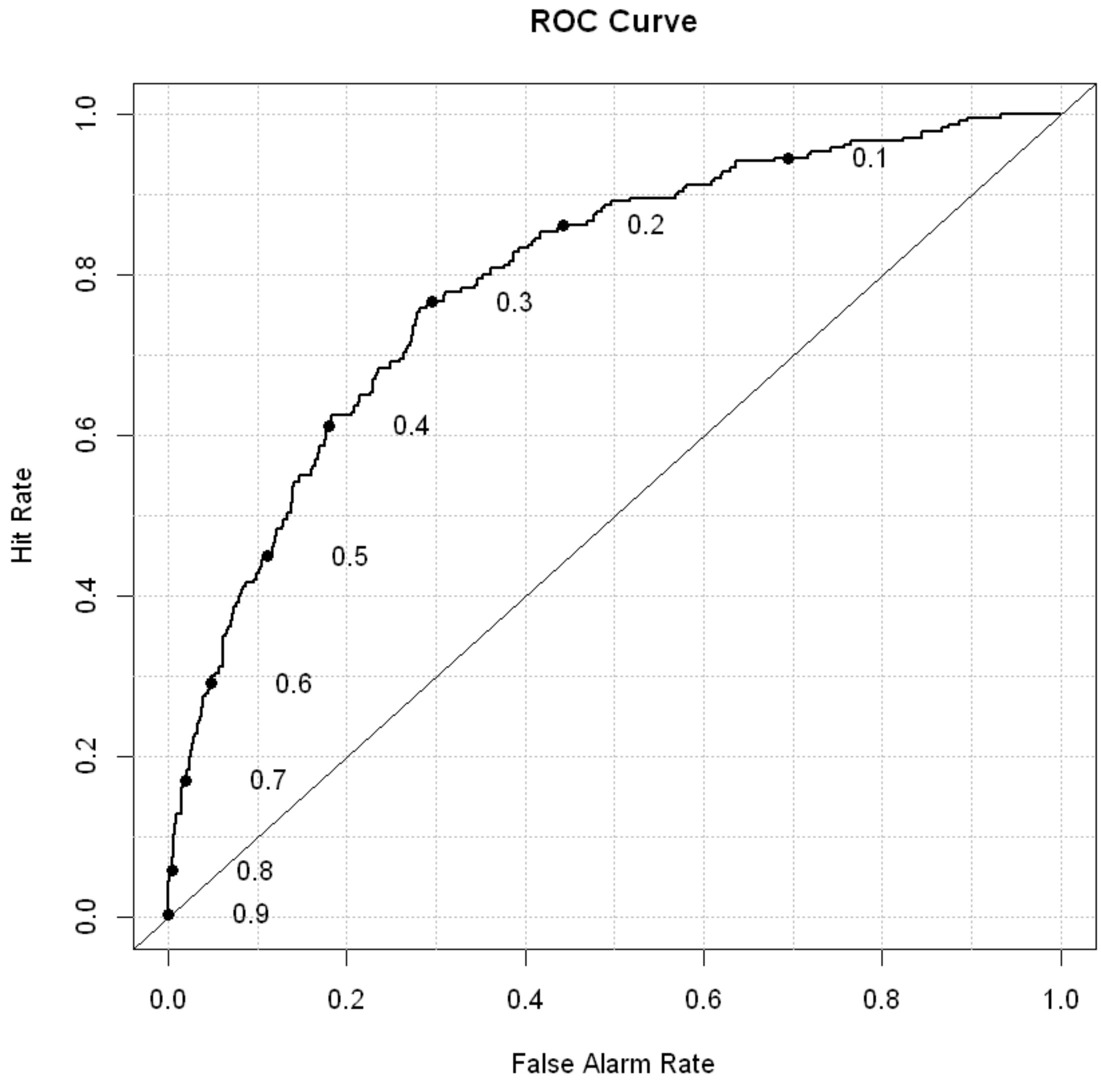
Notice: your models' accuracies will not be assessed for this assignment. However, you are encouraged to try different wrangling or variable selection techniques to obtain the highest accuracies you can. The aim of this question is to assess your understanding of modelling and interpretation of different model evaluation metrics.

In-sample AUC score

```
In [21]: table(german_credit.train$response, predicted.glm1.insample, dnn = c("Truth", "Predicte
         roc.plot(german_credit.train$response == "1", prob.glm1.insample)
         roc.plot(german_credit.train$response == "1", prob.glm1.insample)$roc.vol$Area
```

```
     Predicted
Truth   0   1
    0 278 282
    1  26 214
```

0.794665178571429

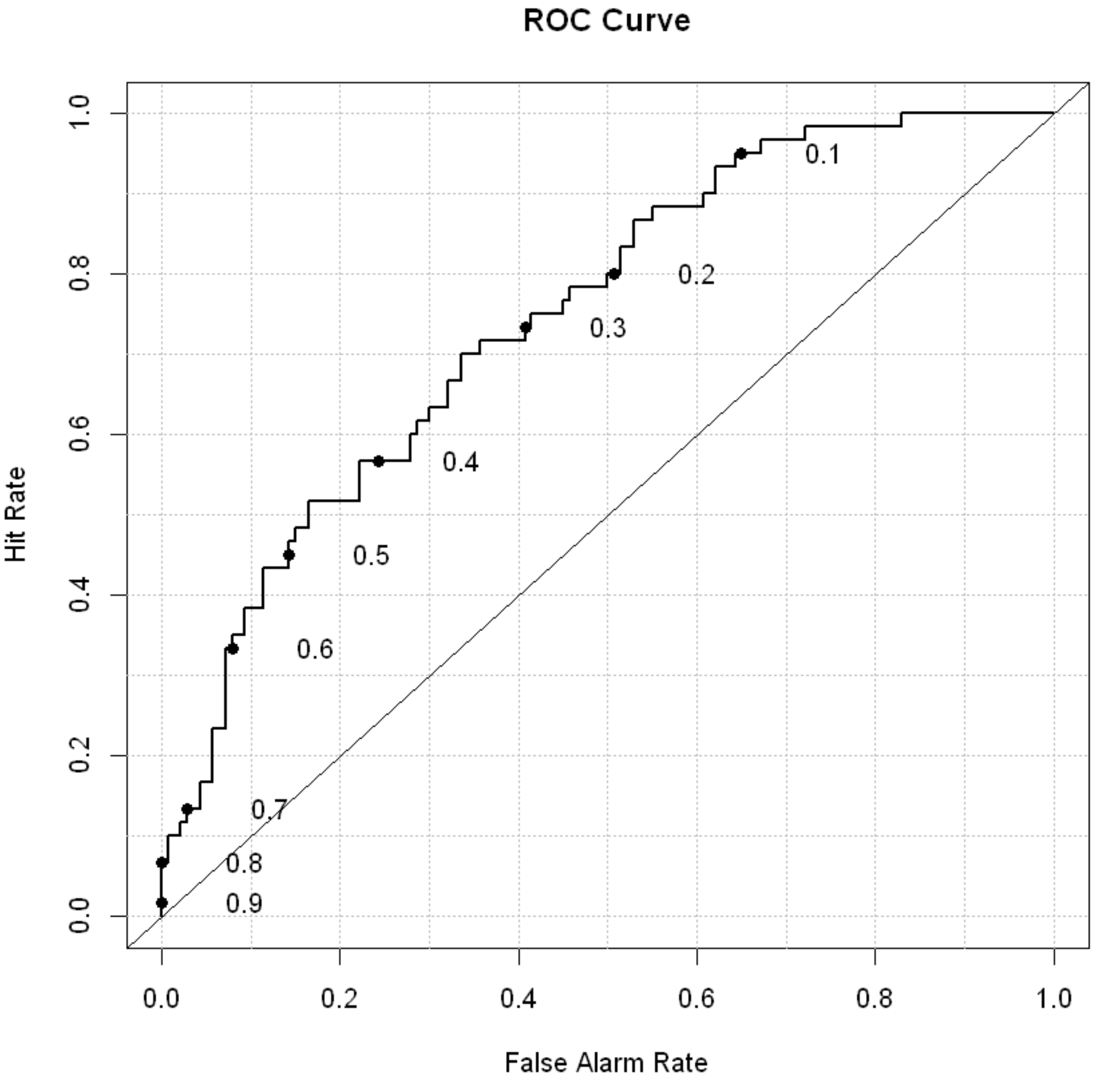


Out of sample misclassification rate and AUC score

```
In [22]: prob.glm1.outsample <- predict(credit.glm.final,
                                        german_credit.test,
                                        type = "response")
         predicted.glm1.outsample <- prob.glm1.outsample > 0.1667
         predicted.glm1.outsample <- as.numeric(predicted.glm1.outsample)
         table(german_credit.test$response,
               predicted.glm1.outsample,
               dnn = c("Truth", "Predicted"))
         mean(ifelse(german_credit.test$response != predicted.glm1.outsample,
                     1,
                     0))
         roc.plot(german_credit.test$response == "1",
                  prob.glm1.outsample)
         roc.plot(german_credit.test$response == "1",
                  prob.glm1.outsample)$roc.vol$Area
```

```
     Predicted
Truth  0  1
    0 62 78
    1  7 53
```

0.425
0.746071428571428



8 Question 6 Rejection sampling

Given $p(x) = N(\mu = 8, \sigma^2 = 4) + N(\mu = 10, \sigma^2 = 25)$ that is a mixture of two Gaussian distributions. Implement the rejection sampling algorithm that samples data from $p(x)$.

Total marks 15. Part a 5 marks. Part b 10 marks.

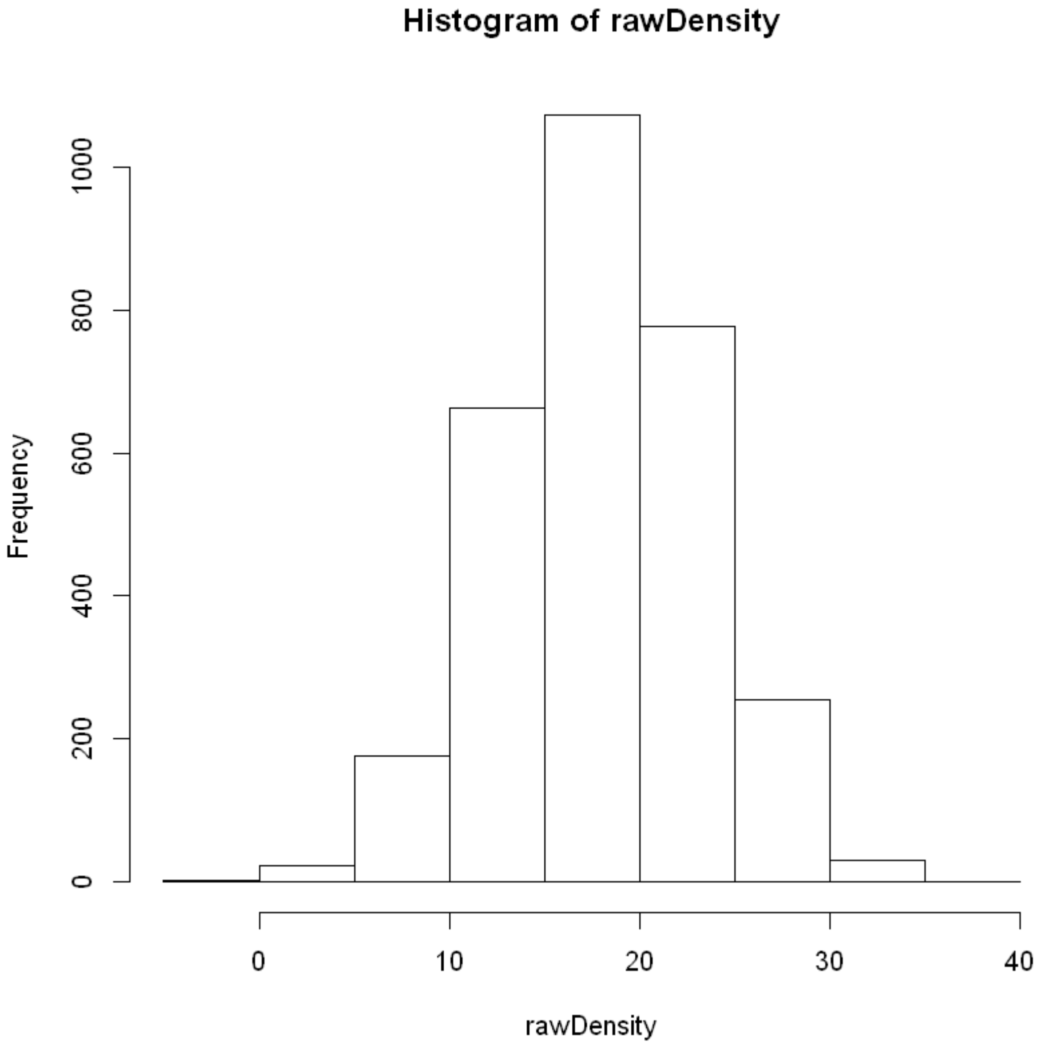
a) Based on the density of $p(x)$, propose a distribution $q(x)$ s.t. $cq(x)$ covers $p(x)$ for a constant c. Plot $p(x)$ and $cq(x)$ in the same plot to justify your choice of $cq(x)$.

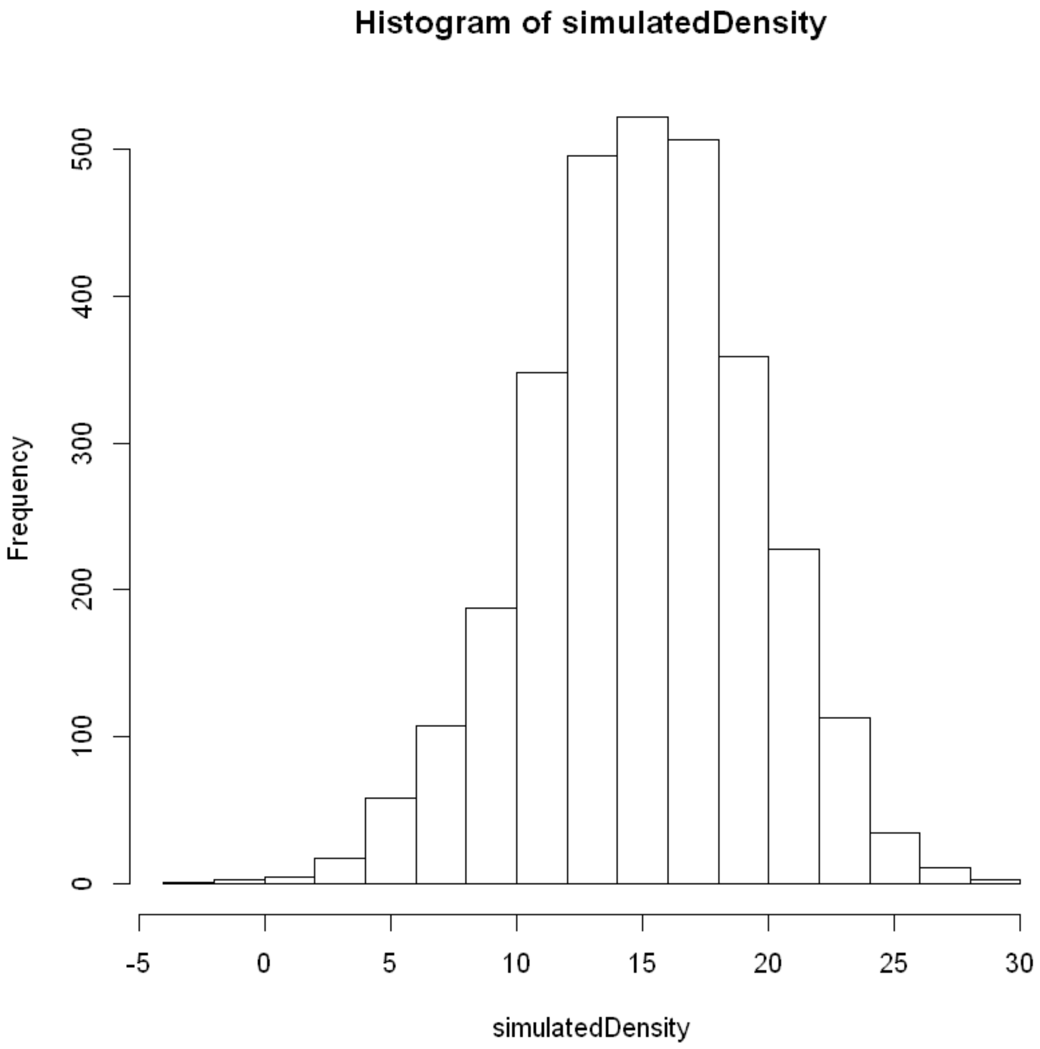
```
In [23]: p.x <- function(n=1)
         {
             return(rnorm(n, mean=8, sd=2) +
                         rnorm(n, mean=10, sd=5))
         }

         RejectionSampling <- function(n)
         {
             RN <- NULL
             for(i in 1:n)
             {
                 OK <- 0
                 while(OK<1)
                 {
                     U <- p.x()
                     if(U <= p.x())
                     {
                         OK <- 1
                         RN <- c(RN,U)
                     }
                 }
             }
             return(RN)
         }
```

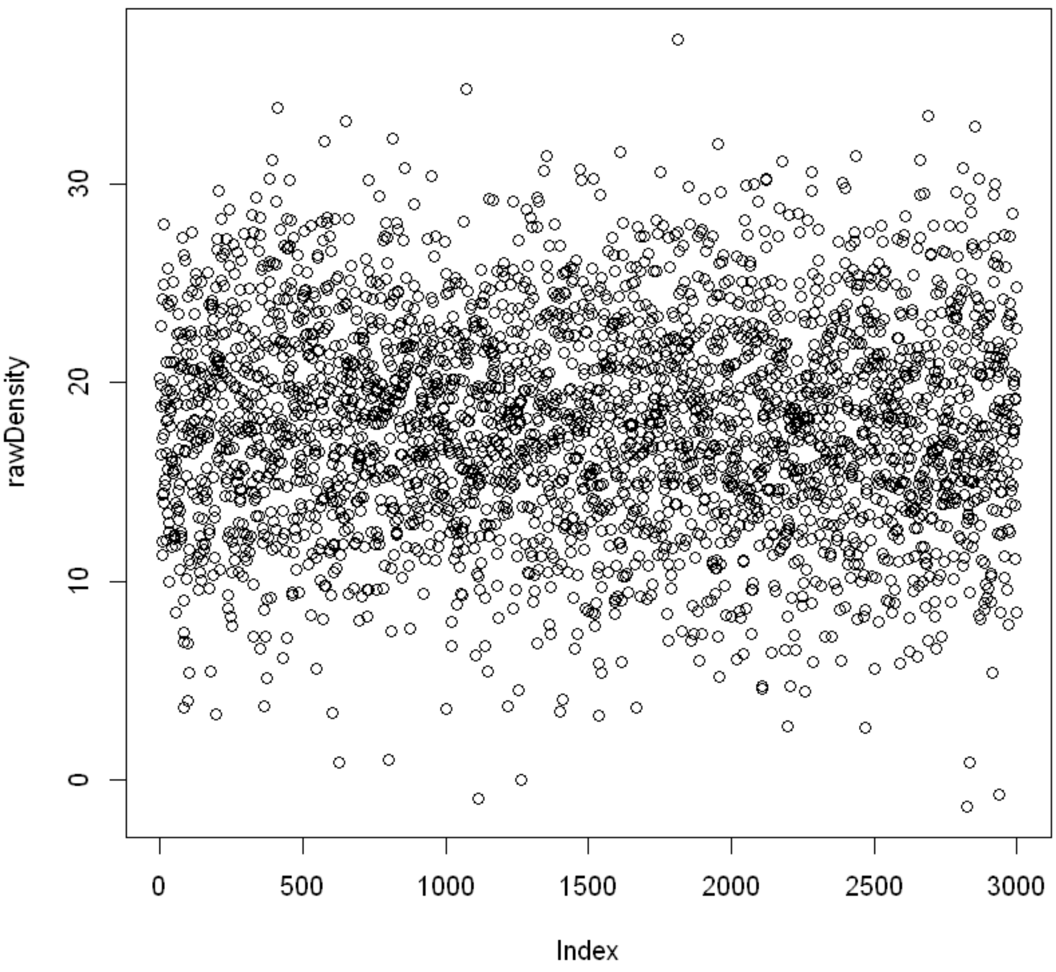
b) Implement the rejection sampling algorithm to sample 10000 data from $p(x)$ for $x \in [-10, 50]$. Plot a histogram (or any other plot) to demonstrate that the sampled data roughly follows the original distribution $p(x)$.

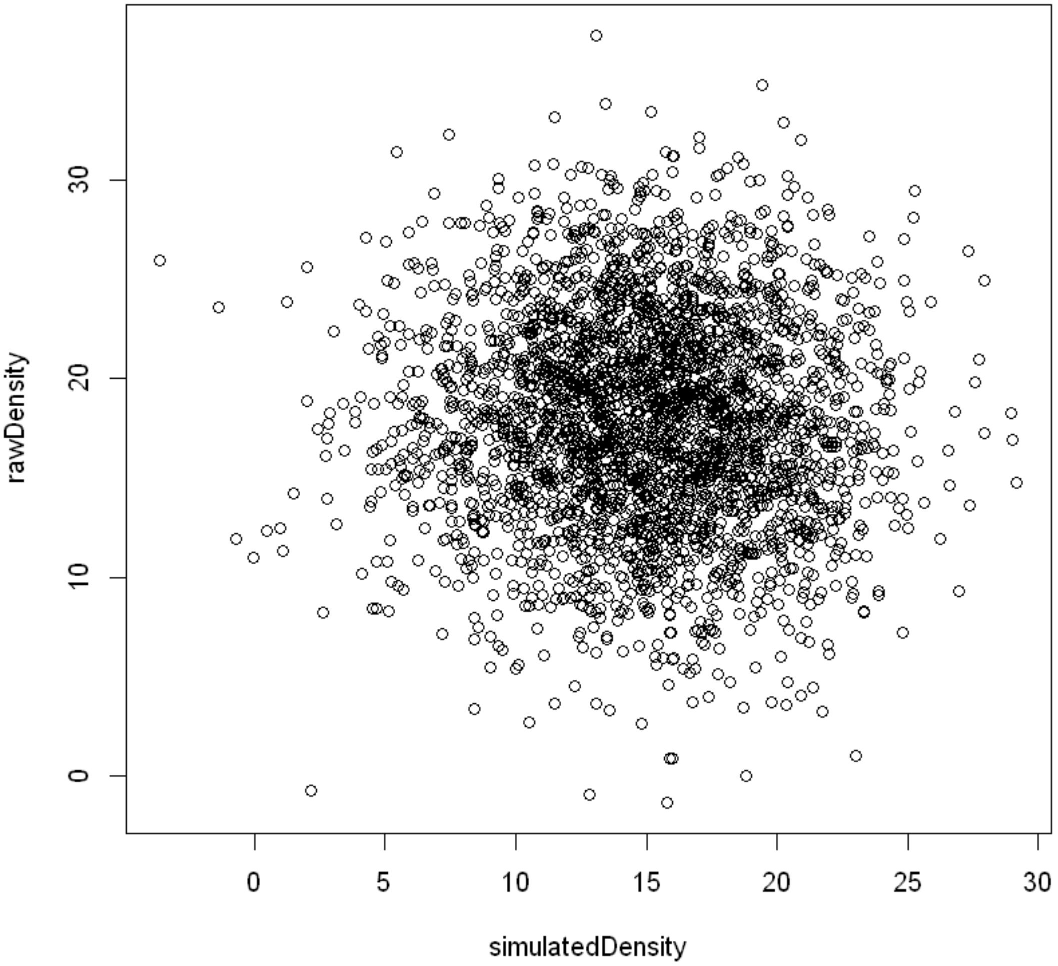
```
In [24]: sampleSize <- 3000
         rawDensity <- p.x(sampleSize)
         simulatedDensity <- RejectionSampling(sampleSize)
         # Calculating the two histograms
         histoRaw <- hist(rawDensity)
         histoSimulated <- hist(simulatedDensity)
         # Q-Q plot raw vs simulated densities
         plot( rawDensity )
         plot( simulatedDensity, rawDensity )
         qqplot(simulatedDensity, rawDensity )
         qqline(simulatedDensity, col = 2)
```

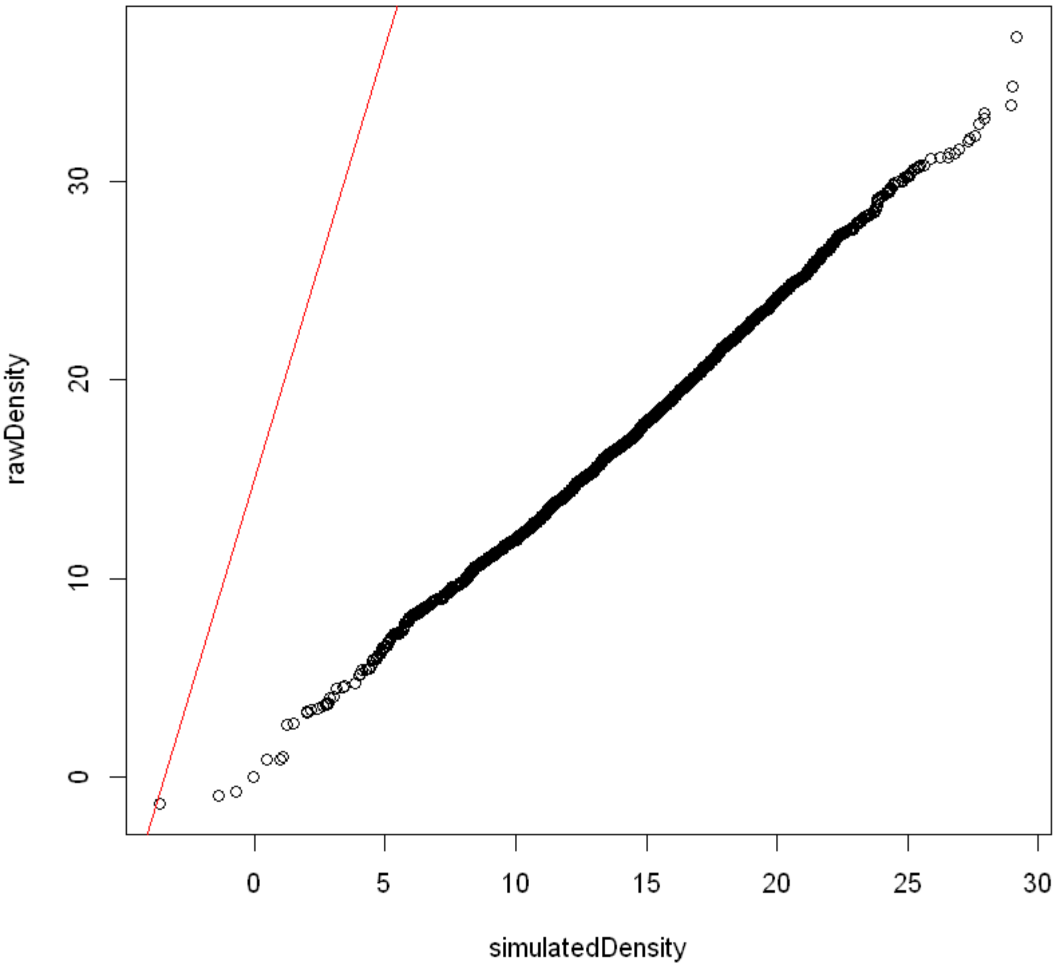
```
# qqplot( rawDensity, simulatedDensity);
# abline(0,1)
# for comparison Q-Q plot of simulated distribution is quite diff from original
qqplot(simulatedDensity, rnorm(1:sampleSize, 0, 1))
```

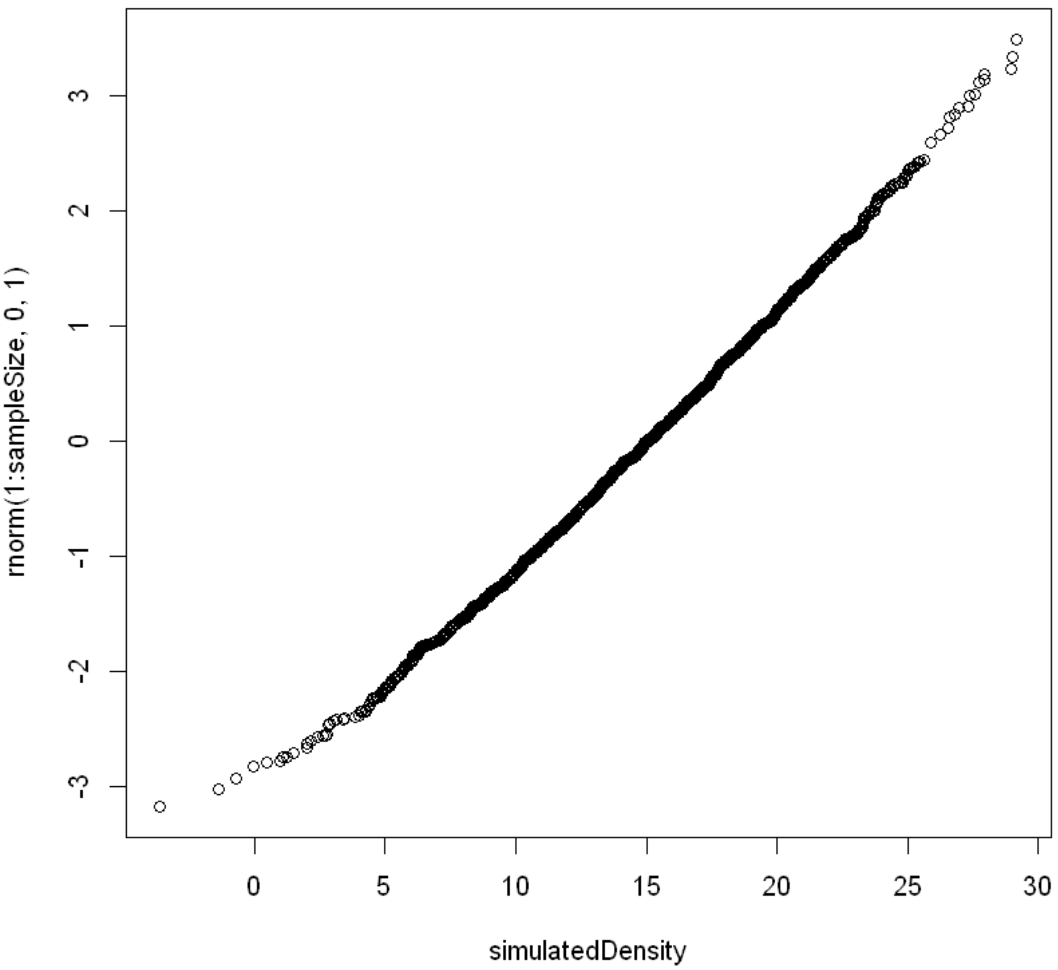



Histogram of simulatedDensity
Frequency
0
100
200
300
400
500
-5
0
5
10
15
20
25
30
simulatedDensity


rawDensity
0
10
20
30
0
500
1000
1500
2000
2500
3000
Index


rawDensity
0
10
20
30
simulatedDensity
0
5
10
15
20
25
30


rawDensity
0
10
20
30
simulatedDensity
0
5
10
15
20
25
30


3
2
1
0
-1
-2
-3
rnorm(1:sampleSize, 0, 1)
0
5
10
15
20
25
30
simulatedDensity