Institute of Actuaries of India

Subject CS1 – Actuarial Statistics
(Paper B)

November 2019 Examination

INDICATIVE SOLUTION

Introduction

The indicative solution has been written by the Examiners with the aim of helping candidates. The solutions given are only indicative. It is realized that there could be other points as valid answers and examiner have given credit for any alternative approach or interpretation which they consider to be reasonable.

Solution 1:

i)

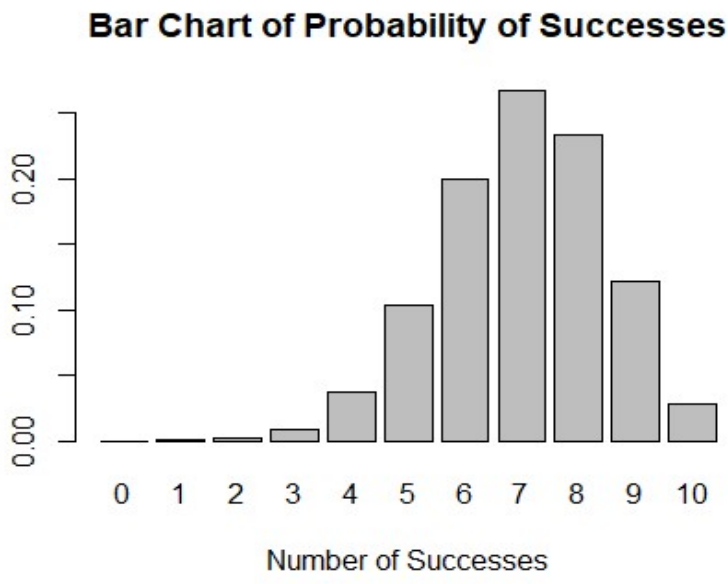
```
p<-0.7

probability distribution table of number of number of wins

prob<-dbinom(0:10,size = 10,prob = 0.7)                                    [3]
probability_Densities<-data.frame(No_Successes = 0:10,prob)
probability_Densities                                                      [1]
```

```
No_Successes         prob
0                0.0000059049
1                0.0001377810
2                0.0014467005
3                0.0090016920
4                0.0367569090
5                0.1029193452
6                0.2001209490
7                0.2668279320
8                0.2334744405
9                0.1210608210
10               0.0282475249
```

[4]

ii)

```
bar chart of the probabilities of number of wins

barplot(prob,main = "Bar Chart of Probability of Successes", xlab = "Numbe
r of Successes", names.arg = 0:10)                                         [4]
```



iii)

```
mean and median number of wins for India against South Africa

mean<-10*0.7 #or
```

```
mean<-sum(probability_Densities$No_Successes*probability_Densities$prob
)
mean                                                                [1]
## [1] 7                                                            [1]
median<-qbinom(0.5,size = 10,prob = 0.7)
median                                                              [1]
## [1] 7                                                            [1]
```

[4]

[12 Marks]

Solution 2:

i)

```
Generate a random sample from a Lognormal distribution
set.seed(100)
data1<-rlnorm(10000,meanlog = 2,sdlog = 0.5)                        [1]
# First 6 observations are shown below
head(data1)                                                         [1]
## [1]  5.748298  7.891337  7.103172 11.512028  7.834097  8.665200
```

[2]

ii)

```
Compute the mean, median and variance of the sample
mean(data1)                                                         [1]
## [1] 8.375649
median(data1)                                                       [1]
## [1] 7.398463
var(data1)                                                          [1]
## [1] 19.51361

# Formula based mean values
mean<-exp(2+0.25/2)
median<-exp(2)
var<-(exp(0.25)-1)*exp(2*2+0.25)

mean                                                                [1]
## [1] 8.372897

Median                                                              [1]

## [1] 7.389056

var                                                                 [1]

## [1] 19.91172

Interpretation: Mean, Median and Variance of the generated sample and t
hose computed based on the parameters are almost equal because the samp
le size is 10,000 which is pretty large. Generating a much larger sampl
e will bridge those smaller differences existing between them as well
                                                                    [2]
```

[8]

iii)

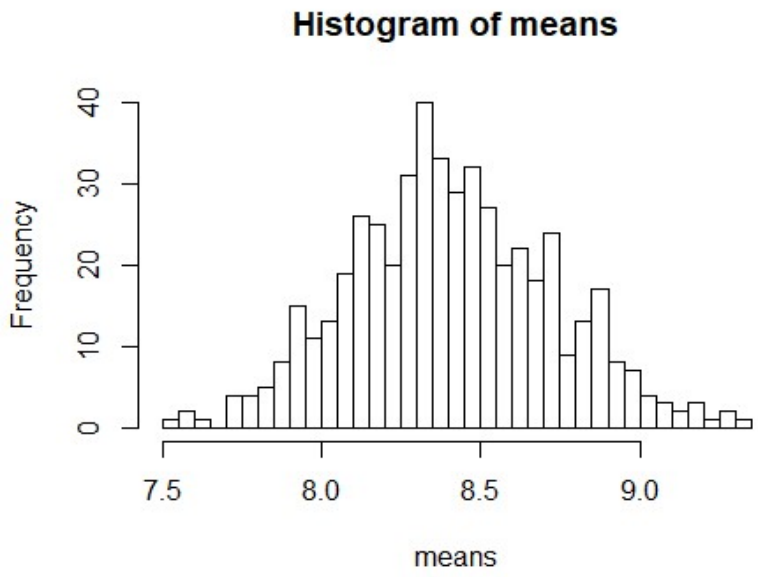
```
Generating 500 different samples of size 200 and computing their sample
means

means<-c()
set.seed(100)
for (i in 1:500){
  selected_rows<-sample(1:10000,200,FALSE)
  selected_data<-data1[selected_rows]
  sample_mean<-mean(selected_data)
  means<-c(means,sample_mean)
}
```

[5]

iv)

```
Histogram of Sample Means
hist(means,breaks = 30)
```

[1]



Interpretation: The sample means tend to follow a normal distribution though the actual data comes from lognormal distribution. Central limit theorem can be verified through this exercise that sample means tend to follow a normal distribution as the sample size increases. Increase in Sample size from 200 to much higher can ensure better normality of the sample means [2]

[3]

[18 Marks]

Solution 3:

```
# Load the data file
indices<-read.csv("D:\\Indices_Returns.csv")
```

i)

```
Compute pearson correlation coefficient and finding the most
correlated and least correlated pair
correlation<-cor(indices[,3:12], method = "pearson")                [1]
correlation<-round(correlation,3)                                   [1]
```

```
correlation                                                         [1]
```

```
##       BM    CD    EN    FM    FI    HC    IN    IT    TE    UT
## BM 1.000 0.882 0.823 0.534 0.819 0.605 0.898 0.447 0.646 0.861
## CD 0.882 1.000 0.783 0.581 0.878 0.637 0.915 0.434 0.696 0.847
## EN 0.823 0.783 1.000 0.446 0.745 0.506 0.799 0.359 0.622 0.793
## FM 0.534 0.581 0.446 1.000 0.485 0.510 0.511 0.303 0.410 0.520
## FI 0.819 0.878 0.745 0.485 1.000 0.502 0.902 0.349 0.623 0.838
## HC 0.605 0.637 0.506 0.510 0.502 1.000 0.588 0.525 0.489 0.530
## IN 0.898 0.915 0.799 0.511 0.902 0.588 1.000 0.370 0.676 0.882
## IT 0.447 0.434 0.359 0.303 0.349 0.525 0.370 1.000 0.291 0.317
## TE 0.646 0.696 0.622 0.410 0.623 0.489 0.676 0.291 1.000 0.669
## UT 0.861 0.847 0.793 0.520 0.838 0.530 0.882 0.317 0.669 1.000
```

[3]

ii)

```
Pair with Minimum Correlation and Pair with maximum correlation
min_cor_location<-which(correlation == min(correlation))[1]
min_cor_pair<-paste(rownames(correlation)[ceiling(min_cor_location/10)]
,colnames(correlation)[ceiling(min_cor_location%%10)])
min_cor_pair
```

```
## [1] "IT TE"                                                       [1]
```

```
max_cor_location<-which(correlation == max(correlation[correlation!=1])
)[1]
max_cor_pair<-paste(rownames(correlation)[ceiling(max_cor_location/10)]
,colnames(correlation)[ceiling(max_cor_location%%10)])
max_cor_pair
```

```
## [1] "CD IN"                                                       [1]
```

[2]

iii)

```
Perform a Principal component analysis of the sectoral return values
PCA_corr<-princomp(indices[,3:12])
summary(PCA_corr)                                                   [4]
```

```
## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     0.2106142 0.06763728 0.05825406 0.04631607
```

```
## Proportion of Variance 0.7294045 0.07522554 0.05580143 0.03527414
## Cumulative Proportion  0.7294045 0.80463008 0.86043151 0.89570565
##                            Comp.5     Comp.6     Comp.7     Comp.8
## Standard deviation     0.04255566 0.03735608 0.03326497 0.03093065
## Proportion of Variance 0.02977884 0.02294646 0.01819564 0.01573154
## Cumulative Proportion  0.92548448 0.94843094 0.96662658 0.98235812
##                             Comp.9     Comp.10
## Standard deviation     0.023803102 0.022500987
## Proportion of Variance 0.009316657 0.008325228
## Cumulative Proportion  0.991674772 1.000000000
```

Alternatively instead of using princomp, the student can use prcomp as well.

```
PCA_corr_1<-prcomp(indices[,3:12])
summary(PCA_corr_1)
```

```
## Importance of components:
##                            PC1     PC2     PC3     PC4     PC5     P6
## Standard deviation     0.2113 0.06784 0.05843 0.04646 0.04269 0.0377
## Proportion of Variance 0.7294 0.07523 0.05580 0.03527 0.02978 0.0225
## Cumulative Proportion  0.7294 0.80463 0.86043 0.89571 0.92548 0.9483
##                            PC7     PC8     PC9    PC10
## Standard deviation     0.03337 0.03103 0.02388 0.02257
## Proportion of Variance 0.01820 0.01573 0.00932 0.00833
## Cumulative Proportion  0.96663 0.98236 0.99167 1.00000
```

iv)

```
Number of PCA components with Eigen value more than 1
sum(PCA_corr$sdev^2/sum(PCA_corr$sdev^2)>(1/10))
```

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

[1]

OR Alternatives

```
sum(PCA_corr_1$sdev^2/sum(PCA_corr_1$sdev^2)>(1/10))
```

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

v)

```
proportion of total variation explained by the first two principal comp
onents
sum(PCA_corr$sdev[1:2]^2)/sum(PCA_corr$sdev^2)
```

```
## [1] 0.8046301
```

[1]

OR Alternatively

```
sum(PCA_corr_1$sdev[1:2]^2)/sum(PCA_corr_1$sdev^2)
```

```
## [1] 0.8046301
```

vi)

```
Paiwise correlations of the transformed components
round(cor(PCA_corr$scores),3)
```

[3]

```
##         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## Comp.1       1      0      0      0      0      0      0      0      0
## Comp.2       0      1      0      0      0      0      0      0      0
```

```
## Comp.3       0      0      1      0      0      0      0      0      0
## Comp.4       0      0      0      1      0      0      0      0      0
## Comp.5       0      0      0      0      1      0      0      0      0
## Comp.6       0      0      0      0      0      1      0      0      0
## Comp.7       0      0      0      0      0      0      1      0      0
## Comp.8       0      0      0      0      0      0      0      1      0
## Comp.9       0      0      0      0      0      0      0      0      1
## Comp.10      0      0      0      0      0      0      0      0      0
##          Comp.10


## Comp.1        0
## Comp.2        0
## Comp.3        0
## Comp.4        0
## Comp.5        0
## Comp.6        0
## Comp.7        0
## Comp.8        0
## Comp.9        0
## Comp.10       1
```

OR Alternatively

```
round(cor(PCA_corr_1$x),3)
```

```
##      PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10
## PC1    1   0   0   0   0   0   0   0   0    0
## PC2    0   1   0   0   0   0   0   0   0    0
## PC3    0   0   1   0   0   0   0   0   0    0
## PC4    0   0   0   1   0   0   0   0   0    0
## PC5    0   0   0   0   1   0   0   0   0    0
## PC6    0   0   0   0   0   1   0   0   0    0
## PC7    0   0   0   0   0   0   1   0   0    0
## PC8    0   0   0   0   0   0   0   1   0    0
## PC9    0   0   0   0   0   0   0   0   1    0
## PC10   0   0   0   0   0   0   0   0   0    1
```

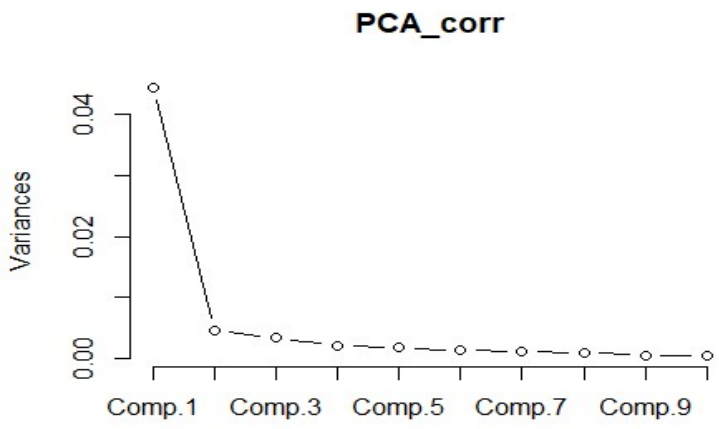
Interpretation

The pairwise correlation between the components after the PCA is performed should be zero as PCA is a way to deal with highly correlated variables. I f N variables are highly correlated than they will all load out on the SA ME Principal Component (Eigenvector) and they will be uncorrelated with ot her components (All these components are orthogonal). Hence the correlatio ns will be zero between the components [2]

[5]

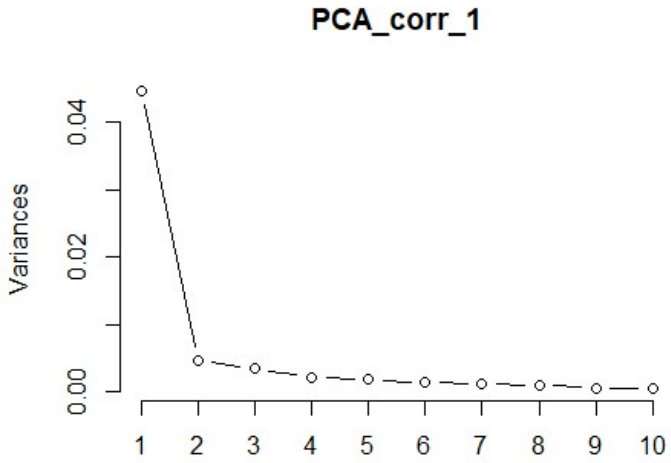
vii)

```
Scree Plot
screeplot(PCA_corr,type = "l")
```
[3]



OR Alternatively

```
screeplot(PCA_corr_1,type = "l")
```




```
Interpretation: Number of significant components is 1 as the scree plot
almost flattened out after the second component                    [1]
```

[4]

[20 Marks]

Solution 4:

```
# Load the data file
indices<-read.csv("D:\\Indices_Returns.csv")
```

i)

```
number of months with negative Sensex returns as a proportion of total
number of months
negative_Sensex<-sum(indices$Sensex<0)
negative_Sensex                                                    [1]

## [1] 69
```

```
proportion_neg<-negative_Sensex/nrow(indices)
proportion_neg                                                    [1]

## [1] 0.4207317
```

[2]

ii)

```
Test whether the proportion of months with negative Sensex returns is l
ess than 50% at 95% confidence level as well as at 99% confidence level

binom.test(negative_Sensex,nrow(indices),p=0.5,alternative   =   "less")
                                                                  [3]

##
##  Exact binomial test
##
## data:  negative_Sensex and nrow(indices)
## number of successes = 69, number of trials = 164, p-value =
## 0.02529
## alternative hypothesis: true probability of success is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.4878846
## sample estimates:
## probability of success
##              0.4207317

Interpretation: p-value corresponding to the test is 0.02529. So the nu
ll hypothesis of Proportion of Negatives is at least 50% is rejected at
95% Confidence level but is failed to be rejected at 99% Confidence lev
el                                                                [1]
```

[4]

iii)

```
Classify the monthly returns of FI and IT
FI<-ifelse(indices$FI<=quantile(indices$FI,0.25),"Low",ifelse(indices$F
I>quantile(indices$FI,0.75),"High","Medium"))                     [1.5]
IT<-ifelse(indices$IT<=quantile(indices$IT,0.25),"Low",ifelse(indices$I
T>quantile(indices$IT,0.75),"High","Medium"))                     [1.5]
table(FI,IT)                                                       [2]

##         IT
## FI       High Low Medium
##   High     14   8     19
##   Low       3  18     20
##   Medium   24  15     43
```

[5]

iv)

```
Test if the returns of FI and IT sectors are independent of each other
chisq.test(FI,IT)                                                 [3]

##
##  Pearson's Chi-squared test
##
## data:  FI and IT
## X-squared = 15.146, df = 4, p-value = 0.004407

Interpretation                                                    [2]
```

- p-value < 0.05 indicating the rejection of null hypothesis of independence of returns of both the sectors
- There is a lot of interdependence in the sectoral returns
- There were very few instances where one sector's returns were below the Q1 and the other sector's returns were above Q3

The numbers in the diagonals are much higher to the ones in the off diagonals indicating the strength of the relationship

[5]

v)

```
Test whether the returns of FI sector are significantly higher compared to
that of IT Sector

t.test(indices$FI,indices$IT, alternative = "greater")                [3]

##
##  Welch Two Sample t-test
##
## data:  indices$FI and indices$IT
## t = 0.22935, df = 314.81, p-value = 0.4094
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.012103  Inf
## sample estimates:
##   mean of x   mean of y
## 0.010674390 0.008720122

Interpretation                                                        [1]
```

- p-value =0.4094 > 0.05 indicates failure to reject the null hypothesis of the return of FI not greater than that of IT sector at 95% confidence level
- There is no sufficient evidence to infer that the returns of FI sector are significantly higher than that of IT Sector

[4]
[20 Marks]

Solution 5:

i)

```
Creation of new column called Sensex_Direction
indices<-read.csv("D:\\Indices_Returns.csv")
indices$Sensex_Direction<-ifelse(indices$Sensex>0,"Positive","Negative"
)                                                                     [2]
indices$Sensex_Direction<-as.factor(indices$Sensex_Direction)         [1]
```

[3]

ii)

```
Fit the model and display the summary

model1<-glm(Sensex_Direction~BM+CD+EN+FI+FM+HC+IN+IT+TE+UT,data = indic
es, family = binomial(link = "logit"))                                [4]

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model1)                                                       [2]
```

```
##
## Call:
## glm(formula = Sensex_Direction ~ BM + CD + EN + FI + FM + HC +
##     IN + IT + TE + UT, family = binomial(link = "logit"), data = ind
ices)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.27544  -0.00117   0.00000   0.01354   1.75651
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0086     0.7315  -1.379  0.16796
## BM            7.7977    16.5255   0.472  0.63703
## CD          -87.5335    42.6785  -2.051  0.04027 *
## EN           93.9675    38.3193   2.452  0.01420 *
## FI          172.8807    60.8192   2.843  0.00448 **
## FM           41.1745    20.1436   2.044  0.04095 *
## HC           -6.4294    13.9394  -0.461  0.64463
## IN            4.1735    18.2152   0.229  0.81877
## IT           78.3494    30.9307   2.533  0.01131 *
## TE           29.9111    13.4184   2.229  0.02581 *
## UT          -14.4767    23.0602  -0.628  0.53015
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 223.213  on 163  degrees of freedom
## Residual deviance:  32.905  on 153  degrees of freedom
## AIC: 54.905
##
## Number of Fisher Scoring iterations: 11
```

[6]

iii)

Sectors significantly impacted

- Sectors which have significantly impacted the direction of Sensex returns are CD, EN, FI, FM, IT and TE at 95% Confidence level [2]
- But only FI has impacted the Sensex direction at 99% Confidence level [2]

[4]

iv)

Relationship between the residual deviance and AIC

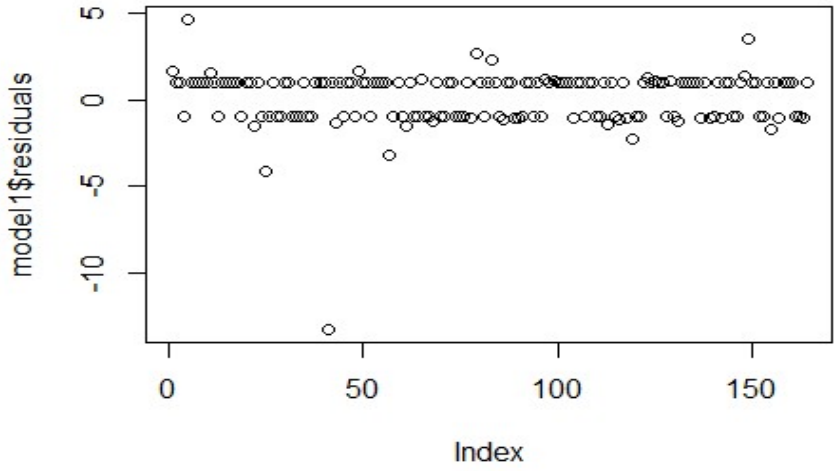
- `all.equal(AIC(model1), model1$deviance+2*11)` **[3]**
- (11 number of model parameters (the number of variables in the model plus the intercept))

```
## [1] TRUE
```

v)

```
Plot the residuals of Model
plot(model1$residuals)
```

[2]




```
which(model1$residuals==min(model1$residuals))

## 41
## 41

indices$Month[which(model1$residuals==min(model1$residuals))]        [2]

## [1] Jun-09

## 164 Levels: Apr-06 Apr-07 Apr-08 Apr-09 Apr-10 Apr-11 Apr-12 ... Sep
-19
```

[4]

vi)

Interpretation **[2]**

- As the residual deviance came down significantly from Null Deviance of 223.21 to 32.90, the variables are able to classify the direction appro priately
- One huge outlier (Jun-09) can impact the accuracy of the result (Removi ng this may reduce the residual deviance further)
- The independent variables are not independent and they are interdepende nt (Correlations are very high among the sectors). Hence the standard e rrors may not be appropriate

As this data is a time series data serial correlation between the observat ions need to be considered and the model may have to be fitted after remov ing serial correlation.

vii)

```
Remove the variables that do not impact the model
model2<-update(model1,~.-BM-HC-IN-UT)                                [3]

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model2)                                                      [1]
```

```
##
## Call:
## glm(formula = Sensex_Direction ~ CD + EN + FI + FM + IT + TE,
##     family = binomial(link = "logit"), data = indices)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.08262  -0.00132   0.00000   0.01757   1.84630
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.9623     0.6803  -1.414  0.15722
## CD          -98.1771    37.5952  -2.611  0.00902 **
## EN           94.2632    33.9930   2.773  0.00555 **
## FI          174.4732    53.3659   3.269  0.00108 **
## FM           37.7105    15.1808   2.484  0.01299 *
## IT           78.7460    26.1203   3.015  0.00257 **
## TE           30.7688    11.8875   2.588  0.00964 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 223.213  on 163  degrees of freedom
## Residual deviance:  33.646  on 157  degrees of freedom
## AIC: 47.646
##
## Number of Fisher Scoring iterations: 10
```

[4]

viii)

```
anova(model1,model2,test = "Chisq")
```

[3]

```
## Analysis of Deviance Table
##
## Model 1: Sensex_Direction ~ BM + CD + EN + FI + FM + HC + IN + IT + TE+
##     UT
## Model 2: Sensex_Direction ~ CD + EN + FI + FM + IT + TE
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       153     32.905
## 2       157     33.646 -4 -0.74102   0.9462
```

Interpretation [1]

- *p-value of the comparison is 0.94 > 0.05 thus not rejecting the null hypothesis of no significant difference between the two models. So the model did not improve significantly based on the friend's suggestion*

[4]

[30 Marks]

****