STAT 6340 Statistical and Machin Learning

Project 6

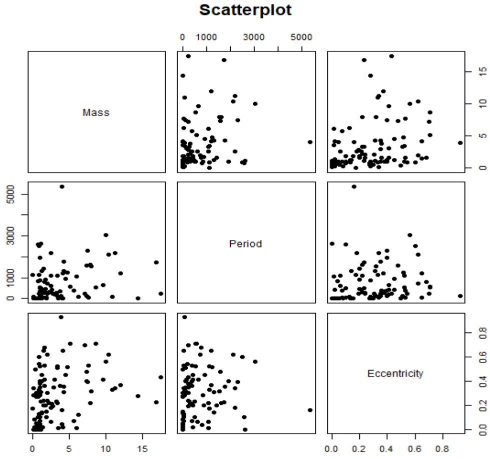
Buddhika Jayawardana

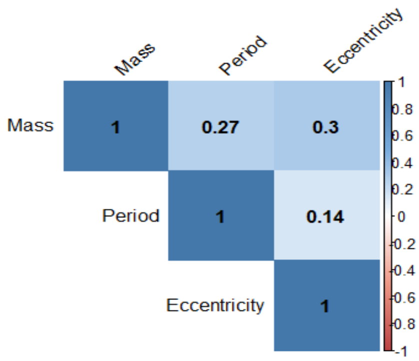
Problem 1

a) The data set contains 3 numerical variables; `Mass`, `Period` and `Eccentricity` of 101 observations. Following is the summery of the data set.

```
> summary(planet)
      Mass             Period          Eccentricity
 Min.   : 0.050   Min.   :   2.985   Min.   :0.0000
 1st Qu.: 0.930   1st Qu.:  44.280   1st Qu.:0.1000
 Median : 1.760   Median : 337.110   Median :0.2700
 Mean   : 3.327   Mean   : 666.531   Mean   :0.2815
 3rd Qu.: 4.140   3rd Qu.:1089.000   3rd Qu.:0.4100
 Max.   :17.500   Max.   :5360.000   Max.   :0.9270
```

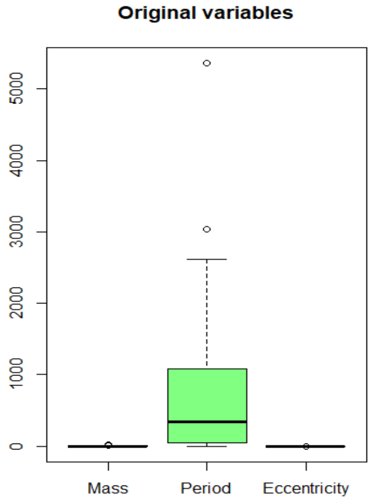
Three variables are ranged differently. `Period` has the largest range `2.985 – 5360.000` and `Eccentricity` has the smallest range `0 – 0.9270`

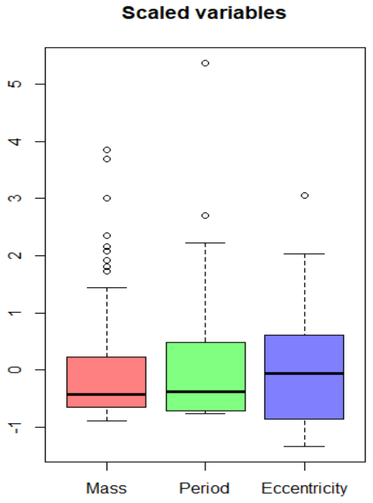




Scatter plots show the correlations between the 3 variables, but their values are small.

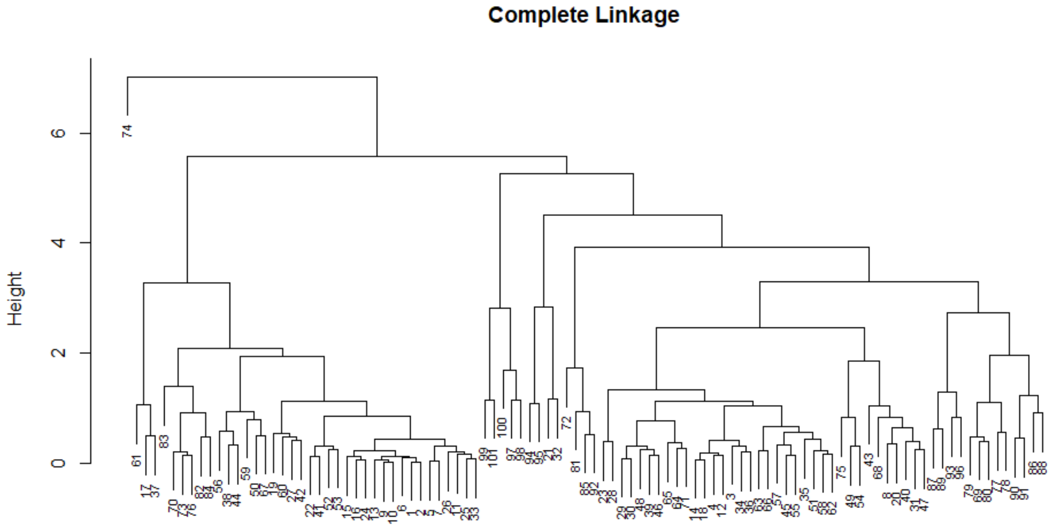
b) Since the range of the period variable is too large compared to the other variables, its good to standardize them before doing any computations.





c) Since the variables are related to dynamics of the exoplanet bodies and the correlations between the variables are low its best to use metric-based distance to cluster the exoplanets.

d) Data is clustered hierarchically using complete linkage and Euclidean distance. Following is the dendrogram of the result.

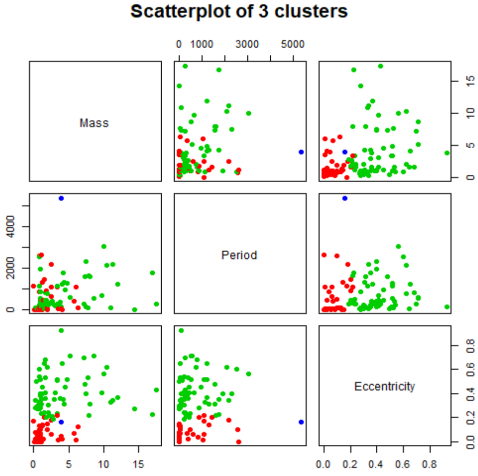


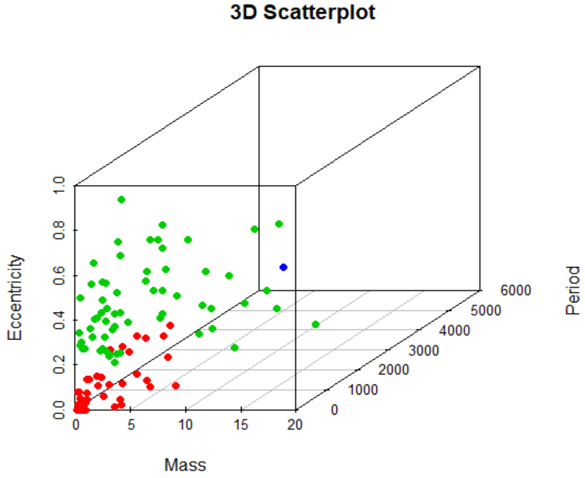
Dendrogram gives 3 clusters as follows.

```
  [1] 1 1 2 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 2 2 1 2 1 1 1 1 2 2 2 2 2 1 2 2 2
 [37] 1 1 2 2 1 1 2 1 2 2 2 2 2 1 2 1 1 2 2 1 2 2 1 1 1 2 2 2 2 2 1 2 2 1 2 2
 [73] 1 3 2 1 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Summary of the cluster-specific means:

```
  Group.1     Mass    Period Eccentricity
1       1 1.703316  488.7479   0.07077105
2       2 4.311774  699.7941   0.41269355
3       3 4.000000 5360.0000   0.16000000
```

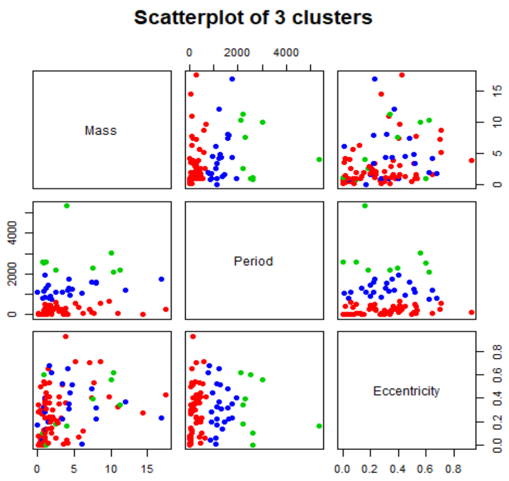


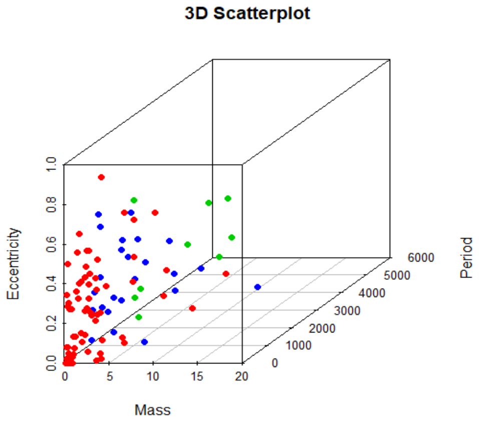


e) When the exoplanets are clustered with K-means, following results are found.

```
Cluster means:
      Mass    Period Eccentricity
1 2.734500  187.5163    0.2606221
2 5.390000 2767.2444    0.3283333
3 4.233333 1235.9729    0.3232917

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 2 1 3 1 1 1 1 3 1 1 1 3 1 1 1 1
 [37] 2 3 1 1 1 1 1 3 1 1 1 1 3 3 1 1 1 3 1 3 1 1 3 1 2 1 1 1 1 1 3 1 3 1 1 1
 [73] 1 2 3 1 3 3 3 3 1 1 3 1 1 3 1 2 1 3 3 1 1 2 2 1 2 3 1 3 1
```

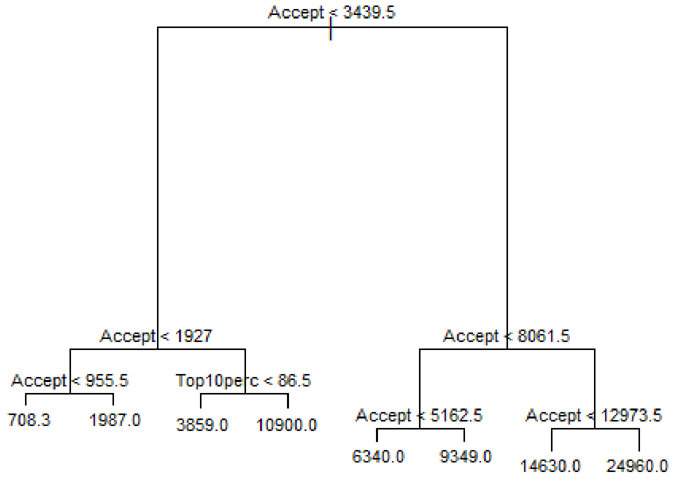




Since the Hierarchical clustering was able to cluster only one exoplanet for the 3rd cluster, K-means clustering works better when you need 3 clusters out of the data set. Also the pairwise scatterplots of the K-means shows somewhat clear separation of these three clusters.

Problem 2

a) MSE = `2031000`



Region	Expression	Mean(Predicted Applications)
1	Accept < 955.5	708.3
2	955.5 < Accept < 1927	1987.0
3	1927 < Accept and Top10per < 86.5	3859.0
4	1927 < Accept and Top10per > 86.5	10900.0
5	3439.5 < Accept < 5162.5	6340.0
6	3439.5 < Accept < 8061.5	9349.0
7	8061.5 < Accept < 12973.5	14630.0
8	12973.5 < Accept	24960.0

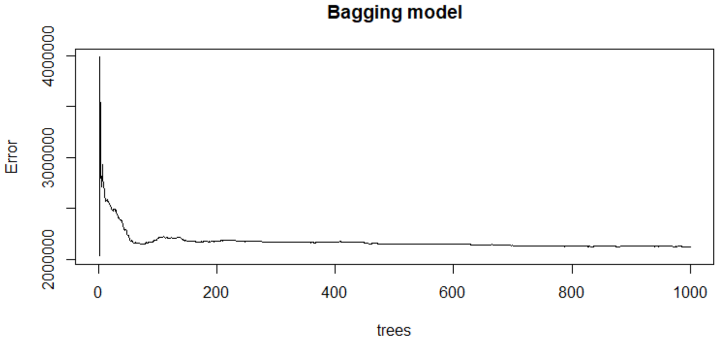
b) When pruning is done for the tree, with LOOCV, best pruning size is found to be 8. Thus pruning is not useful for this tree and the pruned tree is exactly the same as the full tree. So is the MSE.

MSE of the pruned tree = `2031000`

Most important predictors are `Accept` and `Top10per`

c) Bagging approach has MSE = `2124209` and the top important predictor is `Accept`. Bellow are the top 5 important predictors.

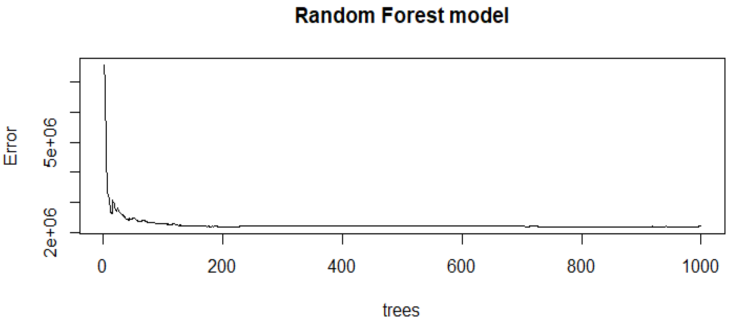
```
                X.IncMSE IncNodePurity
Accept      22016845.6518   10598466352
Top10perc     349136.3838     286343989
Enroll        295881.2979     245209919
Top25perc     188104.5242     161109658
Expend        122938.0253      88231957
```



After 200 trees, there is not much change in terms of error.

d) Random forest approach has MSE = `2213661` and the top two important predictors are `Accept` and `Enroll`. Bellow are the top 5 important predictors.

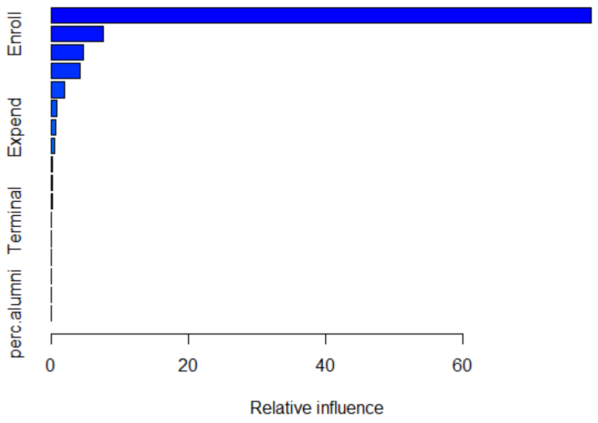
```
             X.IncMSE IncNodePurity
Accept      9357061.39    4787760962
Enroll      3622110.84    2533886881
F.Undergrad 2260502.82    1745082241
P.Undergrad  178029.66     369345190
Top25perc    302858.86     320552183
```



Again after 200 trees, there is not much change in terms of error.

e) Boosting approach has MSE = `3724476` and the top important predictor is `Accept`. Bellow are the top 5 important predictors.

```
                    var     rel.inf
Accept           Accept 78.79257671
Enroll           Enroll  7.55712464
F.Undergrad F.Undergrad  4.68956879
Top10perc     Top10perc  4.22216126
Top25perc     Top25perc  2.01856968
```



f) Out of three methods, minimum MSE = `2124209` gives from Bagging method. Thus the best approach for this data set is Bagging method.

Section 2

```
# problem 1

planet <- read.csv("planet.csv", header = T)
attach(planet)

head(planet)

summary(planet)
#      Mass            Period          Eccentricity
# Min.   : 0.050   Min.   :   2.985   Min.   :0.0000
# 1st Qu.: 0.930   1st Qu.:  44.280   1st Qu.:0.1000
# Median : 1.760   Median : 337.110   Median :0.2700
# Mean   : 3.327   Mean   : 666.531   Mean   :0.2815
# 3rd Qu.: 4.140   3rd Qu.:1089.000   3rd Qu.:0.4100
# Max.   :17.500   Max.   :5360.000   Max.   :0.9270

library(corrplot)

plot(planet, main="Scatterplot", pch=19)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(cor(planet), method="color", col = col(200),
         type="upper",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", tl.srt=45, sig.level = 0.01, insig = "blank")

# part b)

# standardizing the data
planet.scaled <- scale(planet)

apply(planet.scaled, 2, mean)
apply(planet.scaled, 2, sd)


par(mfrow=c(1,2))
boxplot(planet, col = rainbow(3, s = 0.5), main="Original variables")
boxplot(planet.scaled, col = rainbow(3, s = 0.5), main="Scaled variables")
par(mfrow=c(1,1))

# part d)

# hierarchical cluster using complete linkage and Euclidean distance.
hc.complete <- hclust(dist(planet.scaled), method = "complete")

plot(hc.complete, main = "Complete Linkage", xlab = "", sub = "",
     cex = 0.7)

hc.clusters <- cutree(hc.complete, 3)
#  [1] 1 1 2 2 1 1 1 2 1 1 1 2 1 2 1 1 1 2 1 2 2 1 2 1 1 1 1 2 2 2 2 2 1 2
# [35] 2 2 1 1 2 2 1 1 2 1 2 2 2 2 2 1 2 1 1 2 2 1 2 2 1 1 1 2 2 2 2 2 1 2
# [69] 2 1 2 2 1 3 2 1 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2


a <- cbind(planet,hc.clusters)
aggregate(a[, 1:3], list(a$hc.clusters), mean)
#   Group.1     Mass    Period Eccentricity
# 1       1 1.703316  488.7479   0.07077105
# 2       2 4.311774  699.7941   0.41269355
# 3       3 4.000000 5360.0000   0.16000000
```

```
library(scatterplot3d)
scatterplot3d(Mass, Period, Eccentricity, color = (hc.clusters + 1), main="3D
Scatterplot", pch = 16)
plot(planet, main="Scatterplot of 3 clusters", col = (hc.clusters + 1), pch=19)

# part e)
set.seed(3)
km.out <- kmeans(planet, 3, nstart = 20)
# K-means clustering with 3 clusters of sizes 68, 9, 24

# Cluster means:
#       Mass    Period Eccentricity
# 1 2.734500  187.5163    0.2606221
# 2 5.390000 2767.2444    0.3283333
# 3 4.233333 1235.9729    0.3232917
#
# Clustering vector:
#  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 2 1 3 1 1 1 1 3 1 1 1 3 1 1 1 1
# [37] 2 3 1 1 1 1 1 3 1 1 1 1 3 3 1 1 1 3 1 3 1 1 3 1 2 1 1 1 1 1 3 1 3 1 1 1
# [73] 1 2 3 1 3 3 3 3 1 1 3 1 1 3 1 2 1 3 3 1 1 2 2 1 2 3 1 3 1
#
# Within cluster sum of squares by cluster:
#   [1] 2526577 8224052 2492267
# (between_SS / total_SS =  82.7 %)

scatterplot3d(Mass, Period, Eccentricity, color = (km.out$cluster+ 1), main="3D
Scatterplot", pch = 16)
plot(planet, main="Scatterplot of 3 clusters", col = (km.out$cluster+ 1), pch=19)




# problem 2

library(ISLR)

train.data <- College
str(train.data)
attach(train.data)

# part a)

library(tree)
tree.College <- tree(Apps ~ ., train.data)
# node), split, n, deviance, yval
#       * denotes terminal node
#
#   1) root 777 1.162e+10  3002.0
#     2) Accept < 3439.5 651 1.732e+09  1736.0
#       4) Accept < 1927 532 3.251e+08  1182.0
#         8) Accept < 955.5 335 3.457e+07   708.3 *
#         9) Accept > 955.5 197 8.769e+07  1987.0 *
#       5) Accept > 1927 119 5.130e+08  4213.0
#        10) Top10perc < 86.5 113 1.883e+08  3859.0 *
#        11) Top10perc > 86.5 6 4.255e+07 10900.0 *
#     3) Accept > 3439.5 126 3.461e+09  9541.0
#       6) Accept < 8061.5 101 6.351e+08  7770.0
#        12) Accept < 5162.5 53 1.439e+08  6340.0 *
#        13) Accept > 5162.5 48 2.632e+08  9349.0 *
#       7) Accept > 8061.5 25 1.229e+09 16700.0
#        14) Accept < 12973.5 20 1.172e+08 14630.0 *
#        15) Accept > 12973.5 5 6.849e+08 24960.0 *

summary(tree.College)
# Regression tree:
# tree(formula = Apps ~ ., data = train.data)
# Variables actually used in tree construction:
# [1] "Accept"    "Top10perc"
# Number of terminal nodes:  8
# Residual mean deviance:  2031000 = 1.562e+09 / 769
# Distribution of residuals:
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# -8371.0  -429.3  -104.3     0.0   286.7 23140.0

plot(tree.College)
text(tree.College, pretty = 0, cex = 0.7)


# part b)

cv.College <- cv.tree(tree.College, FUN = prune.tree, K=777)
# $size
# [1] 8 7 6 5 4 3 2 1
#
# $dev
# [1]  2381430909  2667272688  3129789615  3196896546  3251213689
# [6]  4065549912  5778625446 12478016777
#
# $k
# [1]       -Inf  202873515  228050501  282105525  426616632  893815939
# [7] 1596851915 6430760151
#
# $method
# [1] "deviance"
#
# attr(,"class")
# [1] "prune"         "tree.sequence"

plot(cv.College$size, cv.College$dev, type = "b")

# get the best size
cv.College$size[which.min(cv.College$dev)]
# [1] 8

prune.College <- prune.tree(tree.College, best = 8)
summary(prune.College)

# Regression tree:
# tree(formula = Apps ~ ., data = train.data)
# Variables actually used in tree construction:
# [1] "Accept"    "Top10perc"
# Number of terminal nodes:  8
# Residual mean deviance:  2031000 = 1.562e+09 / 769
# Distribution of residuals:
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# -8371.0  -429.3  -104.3     0.0   286.7 23140.0
```

```
plot(prune.College)
text(prune.College, pretty = 0)


# part c)

library(randomForest)
set.seed(1)
bag.College <- randomForest(Apps ~ ., data = College, mtry = 17,
                            ntree = 1000, importance = TRUE)
# Call:
#   randomForest(formula = Apps ~ ., data = College, mtry = 17, ntree = 1000,
importance = TRUE)
# Type of random forest: regression
# Number of trees: 1000
# No. of variables tried at each split: 17
#
# Mean of squared residuals: 2124209
# % Var explained: 85.8

bag.imp <- data.frame(bag.College$importance)
bag.imp[order(bag.imp$IncNodePurity, decreasing = T),]
#                  X.IncMSE IncNodePurity
# Accept      22016845.6518   10598466352
# Top10perc     349136.3838     286343989
# Enroll        295881.2979     245209919
# Top25perc     188104.5242     161109658
# Expend        122938.0253      88231957

plot(bag.College, main = "Bagging model")

# part d)

set.seed(1)
forest.College <- randomForest(Apps ~ ., data = College, mtry = 6,
                            ntree = 1000, importance = TRUE)
# Call:
#   randomForest(formula = Apps ~ ., data = College, mtry = 6, ntree = 1000,
importance = TRUE)
# Type of random forest: regression
# Number of trees: 1000
# No. of variables tried at each split: 6
#
# Mean of squared residuals: 2213661
# % Var explained: 85.2

forest.imp <- data.frame(forest.College$importance)
forest.imp[order(forest.imp$IncNodePurity, decreasing = T),]
#               X.IncMSE IncNodePurity
# Accept      9357061.39    4787760962
# Enroll      3622110.84    2533886881
# F.Undergrad 2260502.82    1745082241
# P.Undergrad  178029.66     369345190
# Top25perc    302858.86     320552183


plot(forest.College, main = "Random Forest model")
```

```
# part e)
library(gbm)

set.seed(1)
boost.College <- gbm(Apps ~ ., data = College, distribution = "gaussian",
                    n.trees = 1000, interaction.depth = 1, shrinkage = 0.01, cv.folds =
777)
# gbm(formula = Apps ~ ., distribution = "gaussian", data = College,
#     n.trees = 1000, interaction.depth = 1, shrinkage = 0.01,
#     cv.folds = 777)
# A gradient boosted model with gaussian loss function.
# 1000 iterations were performed.
# The best cross-validation iteration was 999.
# There were 17 predictors of which 14 had non-zero influence.


summary(boost.College)
# var     rel.inf
# Accept           Accept 78.79257671
# Enroll           Enroll  7.55712464
# F.Undergrad F.Undergrad  4.68956879
# Top10perc     Top10perc  4.22216126
# Top25perc     Top25perc  2.01856968
# P.Undergrad P.Undergrad  0.88111834


mean(boost.College$cv.error)
# [1] 3724476
```