1)

a)

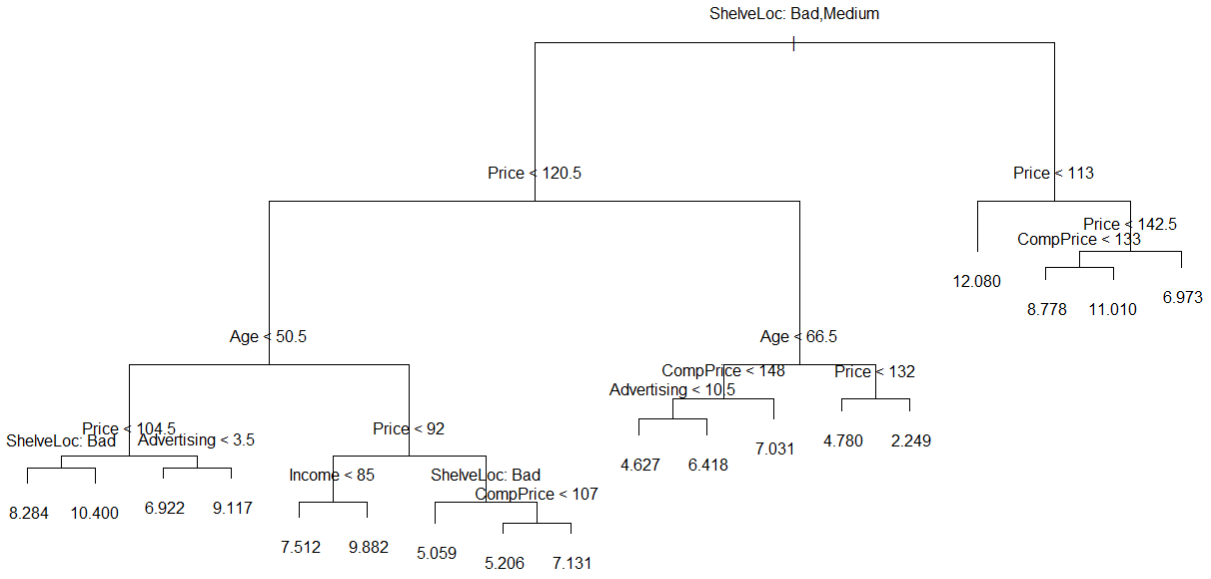
```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> attach(Carseats)
> set.seed(1)
> train=sample(1:nrow(Carseats),nrow(Carseats)/2)
> tree.car=tree(Sales~.,Carseats,subset=train)
>
```

b)

```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> tree.car=tree(Sales~.,Carseats,subset=train)
> summary(tree.car)

Regression tree:
tree(formula = Sales ~ ., data = Carseats, subset = train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "Advertising" "Income"      "CompPrice"
Number of terminal nodes:  18
Residual mean deviance:  2.36 = 429.5 / 182
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.2570 -1.0360  0.1024  0.0000  0.9301  3.9130
> plot(tree.car)
> text(tree.car ,pretty =0)
>
> tree.car.pred=predict(tree.car,newdata=Carseats[-train,])
> Sales.test=Sales[-train]
> mean((tree.car.pred-Sales.test)^2)
[1] 4.148897
>
```

Test MSE:4.15

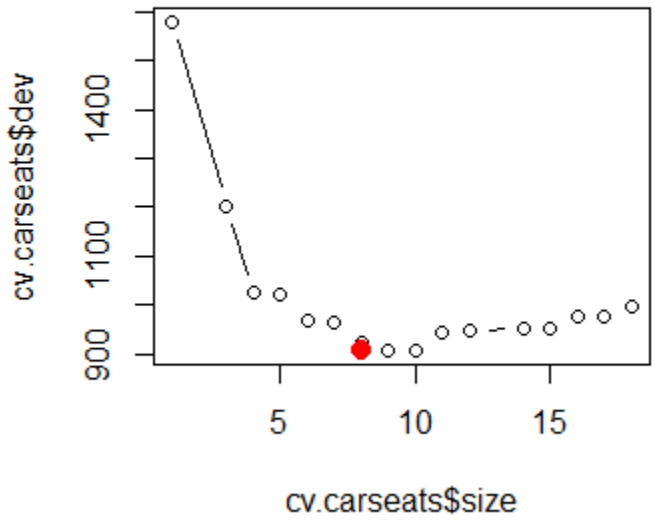


There are a total of 18 terminal nodes.

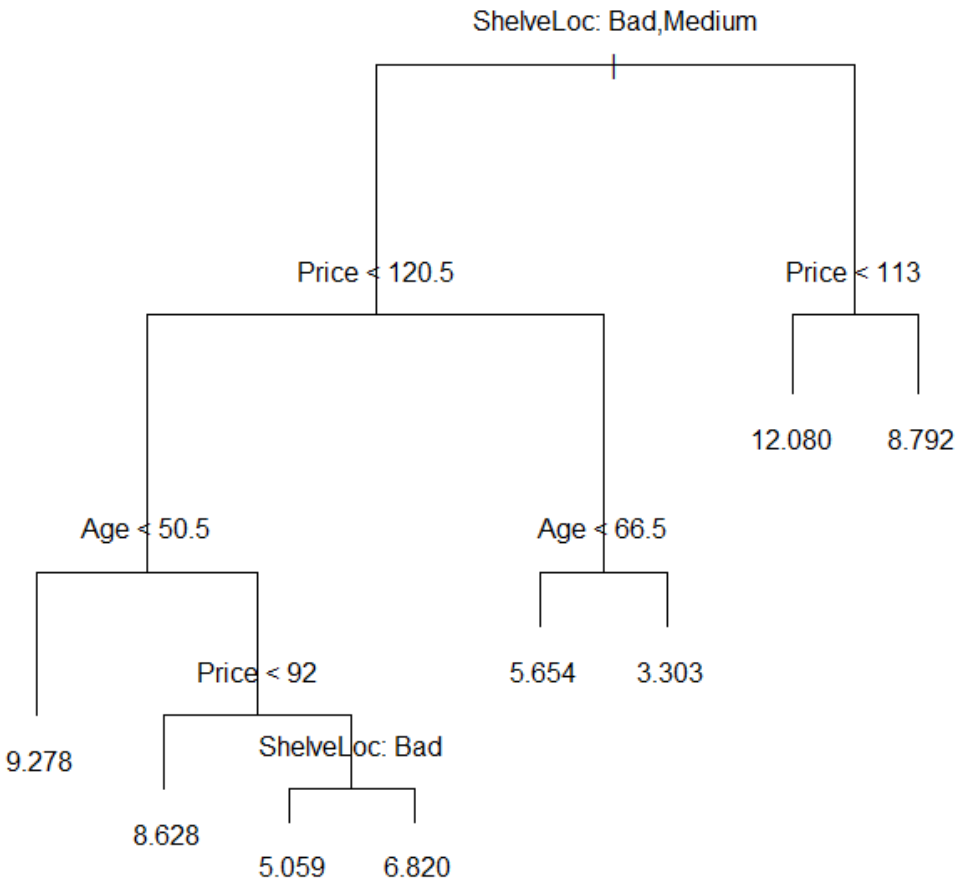
It has been found that the test MSE is 4.15. Certain variables like “Age” is used to find the prediction in certain scenarios only like of Sales if Price is less than 120. The sales is highest for the case where ShelveLoc is Bad or medium and price is lesser than 113. The tree has a large number of branches.

c)

```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> set.seed(2)
> cv.carseats <- cv.tree(tree.car)
> plot(cv.carseats$size, cv.carseats$dev, type = "b")
> tree.min=which.min(cv.carseats$dev)
> points(tree.min, cv.carseats$dev[tree.min],col="red", cex = 2, pch = 20)
>
> prune.car=prune.tree(tree.car,best=8)
> plot(prune.car)
> text(prune.car,pretty=0)
>
> tree.car.pred=predict(prune.car,newdata=Carseats[-train,])
> Sales.test=Sales[-train]
> mean((tree.car.pred-Sales.test)^2)
[1] 5.09085
>
```



According to cross validation, 8 variable model turns out to be the best with least test MSE.



There are a total of 8 terminal nodes.

It has been found that the test MSE is 5.09.

The test MSE has increased on pruning.

Certain variables like “Age” again as in the tree before pruning gives the first level of division is used to find the prediction in certain scenarios only like of Sales if Price is less than 120. The sales is highest for the case where ShelveLoc is Bad or medium and price is lesser than 113. The tree lesser number of branches compared to the tree before pruning.

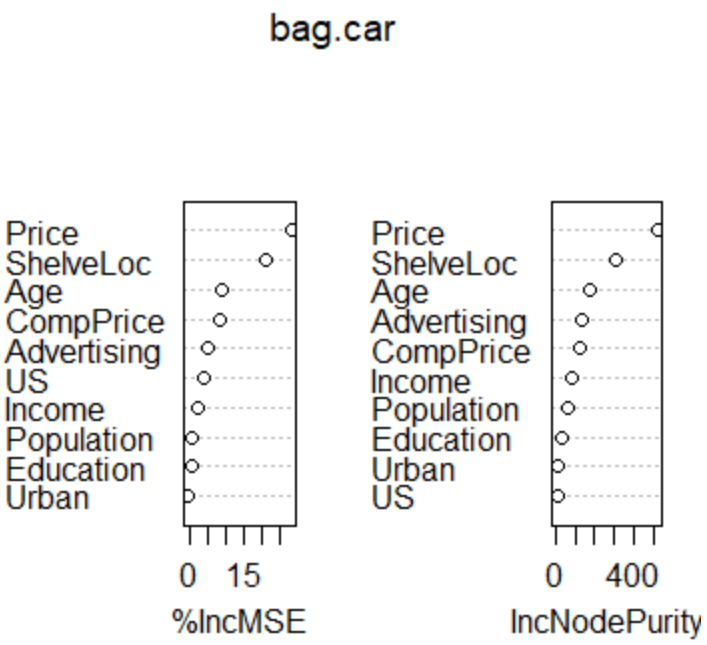
d)

```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> library(randomForest)
> library(MASS)
> set.seed(3)
> bag.car=randomForest(Sales~.,data=Carseats,subset=train,mtry=10,ntree=100,importance=TRUE)
> bag.car

Call:
 randomForest(formula = Sales ~ ., data = Carseats, mtry = 10,      ntree = 100, importance = TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 10

          Mean of squared residuals: 2.894799
                    % Var explained: 62.07
> bag.car.pred=predict(bag.car,newdata=Carseats[-train,])
> Sales.test=Sales[-train]
> mean((bag.car.pred-Sales.test)^2)
[1] 2.583883
> importance(bag.car)
               %IncMSE IncNodePurity
CompPrice    8.3236866    128.598712
Income       2.4831143     79.786622
Advertising  5.0618963    135.343990
Population   0.9584555     63.963902
Price       28.0211679    526.941175
ShelveLoc   20.8543196    313.384280
Age          8.9751455    176.316713
Education    0.5304952     36.167503
Urban       -0.2151944     10.860196
US           3.9492746      9.647443
> varImpPlot(bag.car)
>
```

Test MSE: 2.58



Price is the most important variable. ShelveLoc and Age seem to be the next 2 important variables.

According to the mean decrease in RSS/ impurity averaged over all the trees, the above inference is made.

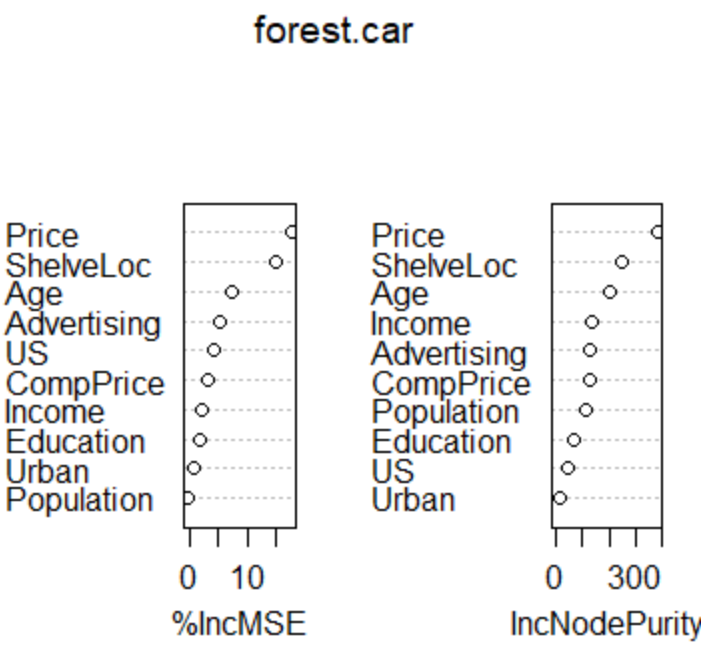
e)

```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> set.seed(3)
> forest.car=randomForest(Sales~.,data=Carseats,subset=train,mtry=3,ntree=100,importance=TRUE)
> forest.car

Call:
 randomForest(formula = Sales ~ ., data = Carseats, mtry = 3,      ntree = 100, importance = TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 3

          Mean of squared residuals: 3.509524
                    % Var explained: 54.02
> forest.car.pred=predict(forest.car,newdata=Carseats[-train,])
> Sales.test=Sales[-train]
> mean((forest.car.pred-Sales.test)^2)
[1] 3.32326
> importance(forest.car)
               %IncMSE IncNodePurity
CompPrice    3.2053934     126.57590
Income       1.9480894     133.81025
Advertising  5.3791741     129.07584
Population  -0.2290071     110.78351
Price       17.6824317     385.51295
ShelveLoc   15.1218761     245.35465
Age          7.3791971     205.36270
Education    1.5704598      68.11049
Urban        0.8578892      13.68792
US           4.2461008      42.58278
> varImpPlot(forest.car)
>
```

Test MSE: 3.32



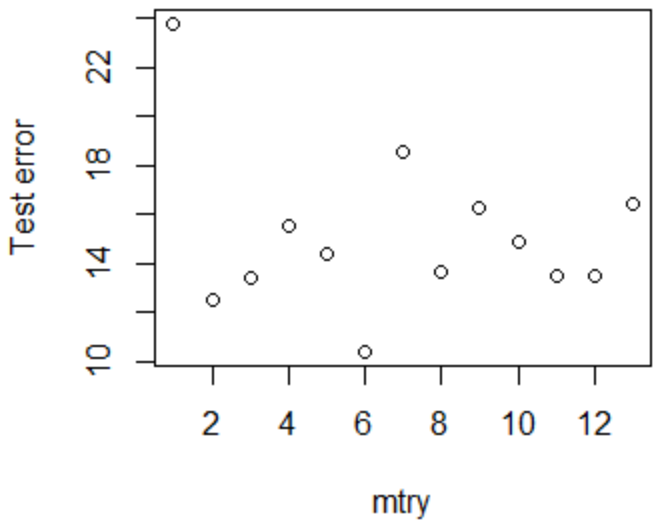
Price is the most important variable. ShelveLoc and Age seem to be the next 2 important variables.

According to the mean decrease in RSS/ impurity averaged over all the trees, the above inference is made.

2)

a)

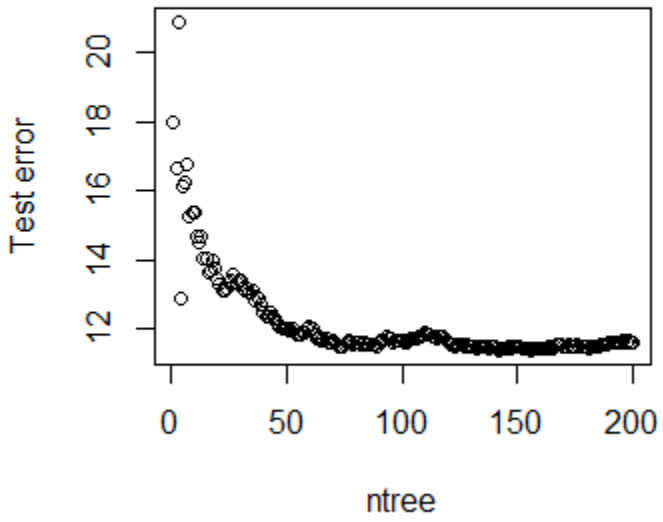
```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> for (i in 1:13)
+ {
+ #set.seed(1)
+ train=sample(1:nrow(Boston),nrow(Boston)/2)
+ boston.test=Boston[-train,"medv"]
+ set.seed(1)
+ bag.boston=randomForest(medv~.,data=Boston,subset=train,mtry=i,ntree=100,importance=TRUE)
+ #bag.boston
+ boston.pred=predict(bag.boston,newdata=Boston[-train,])
+ Test.mean[i]=mean((boston.pred-boston.test)^2)
+ }
> plot(Test.mean,xlab="mtry",ylab="Test error")
>
```



Test error is the maximum when 1 variable alone is considered. Also, when 6 variables are considered, Test error seems to be the least. Moreover there does not seem to be any constant trend between the test error and 'mtry'.

b)

```
Console C:/Users/Karthik/Desktop/Sem 1/ISEN 613/
> for (i in 5:200)
+ {
+   set.seed(1)
+   train=sample(1:nrow(Boston),nrow(Boston)/2)
+   boston.test=Boston[-train,"medv"]
+   set.seed(1)
+   bag.boston=randomForest(medv~.,data=Boston,subset=train,mtry=6,ntree=i,importance=TRUE)
+   #bag.boston
+   boston.pred=predict(bag.boston,newdata=Boston[-train,])
+   Test.mean[i]=mean((boston.pred-boston.test)^2)
+ }
> plot(Test.mean,xlab="ntree",ylab="Test error")
>
```



With an increase in the tree length, the test error decreases.

After a certain increase in the tree length, the test error does not decrease significantly, the increase of which only results in increased computational time.