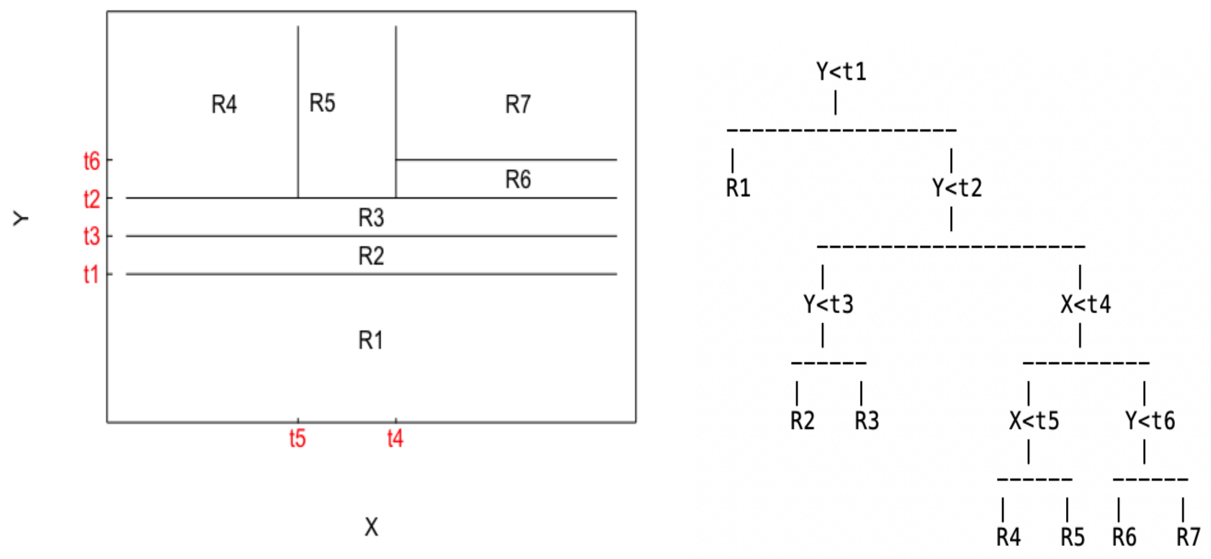
Solution to Series 11

1. Here we give one example with six regions.



2. The 10 estimates of $P(\text{Class is Red}|X)$ are

$$0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, \text{ and } 0.75,$$

and the corresponding classification results are

Green, Green, Green, Green, Red, Red, Red, Red, Red and Red.

There are 4 Greens and 6 Reds, hence under the majority vote, the final classification is Red.

The average probability of the 10 estimated probabilities is 0.45. Therefore, based on the average probability approach, the final classification is Green. One can see that the results are different by using these two different approaches.

3. a)

```
> library(ISLR)
> data(Carseats)
> # set random seed as the train set is randomly selected,
> # and implementing randomForest contains randomness
> set.seed(39)
> # using random seed 10, you will see that the test MSE of the pruned tree is improved
> # set.seed(10)
>
> n <- nrow(Carseats)
> p <- ncol(Carseats)
> train = sample(n, round(n/2))
> Carseats.train = Carseats[train, ]
> Carseats.test = Carseats[-train, ]
```

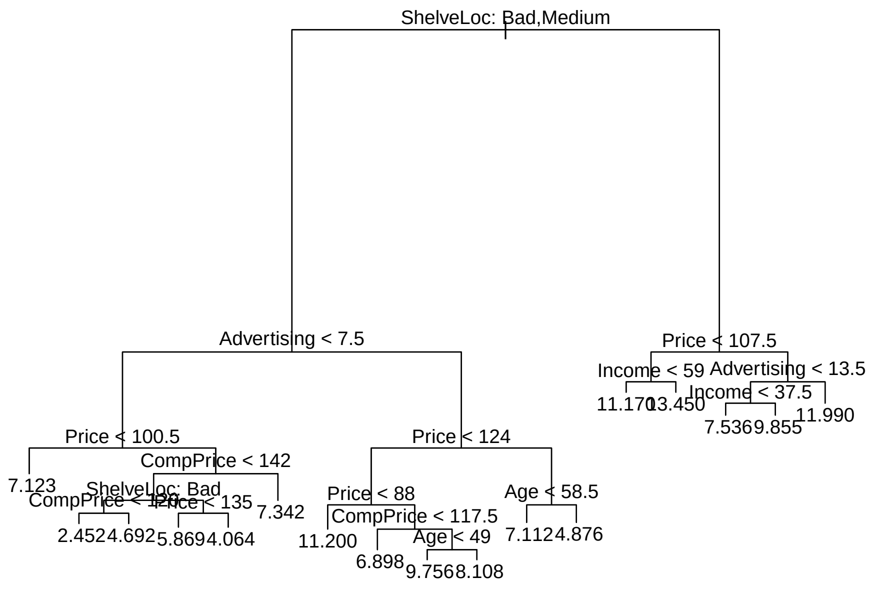
b)

```
> library(tree)           # you may need to update your R-version to install "tree" package
> # fit a regression tree
> tree.carseats = tree(Sales ~ ., data = Carseats.train)
> summary(tree.carseats)

Regression tree:
tree(formula = Sales ~ ., data = Carseats.train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Advertising" "Price"       "CompPrice"
[5] "Age"         "Income"
Number of terminal nodes:  17
Residual mean deviance:  2.068 = 378.5 / 183
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-4.13300 -0.97010 -0.09117  0.00000  0.91310  3.10100

> plot(tree.carseats)
> text(tree.carseats, pretty = 0)
> pred.carseats = predict(tree.carseats, Carseats.test)
> mean((Carseats.test$Sales - pred.carseats)^2)

[1] 5.134747
```



The test MSE is 5.134747.

c)

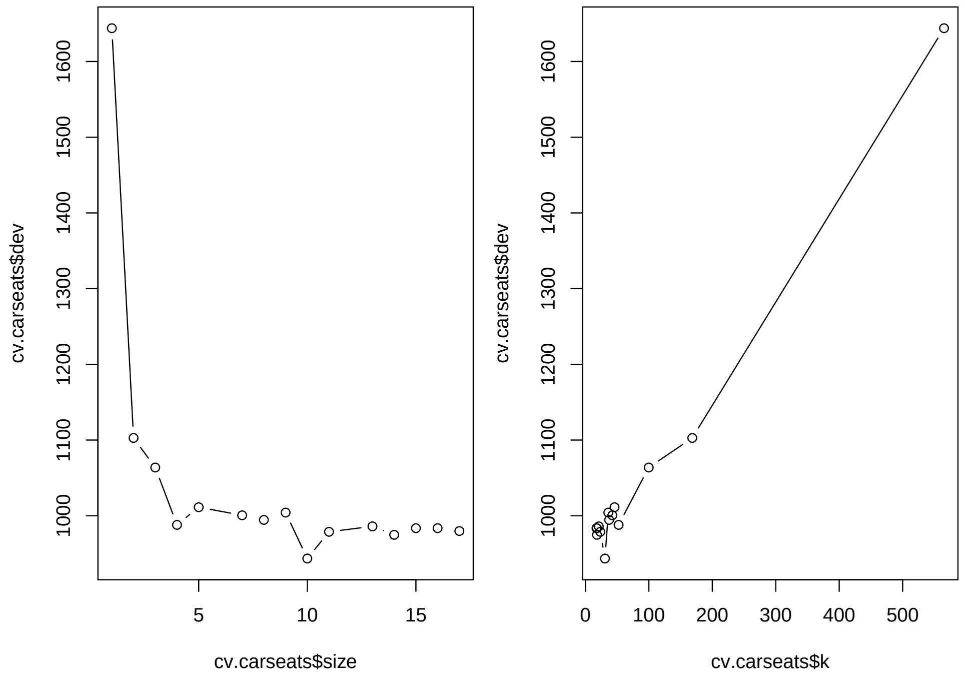
```
> cv.carseats = cv.tree(tree.carseats, FUN = prune.tree)
> par(mfrow = c(1, 2))
> plot(cv.carseats$size, cv.carseats$dev, type = "b")
> plot(cv.carseats$k, cv.carseats$dev, type = "b")
> best.size <- cv.carseats$size[which.min(cv.carseats$dev)]
> best.size

[1] 10
```

```
> pruned.carseats = prune.tree(tree.carseats, best = best.size)
> par(mfrow = c(1, 1))
> plot(pruned.carseats)
> text(pruned.carseats, pretty = 0)
> pred.pruned = predict(pruned.carseats, Carseats.test)
> mean((Carseats.test$Sales - pred.pruned)^2)

[1] 5.358271

>
```



The test MSE of the pruned tree is 5.358271, so pruning the tree didn't improve the test MSE. But note that it is not always true, sometimes pruning the tree can improve the test MSE. For example, when use "set.seed(10)" (so we will have different training and testing sets), you will see that pruning improving the test MSE.

d)

```
> library(randomForest)
> # use the baggin approach to fit a model
> bag.carseats = randomForest(Sales ~ ., data = Carseats.train, mtry = p-1, ntree = 500,
 importance = T)
> bag.pred = predict(bag.carseats, Carseats.test)
> mean((Carseats.test$Sales - bag.pred)^2)

[1] 3.315656

> importance(bag.carseats)

              %IncMSE IncNodePurity
CompPrice   23.3746577    161.519653
Income       6.9992800     89.094905
Advertising 23.5313121    191.145427
Population   0.5537334     57.860247
Price       46.6265611    349.999843
ShelveLoc   69.4954039    602.716975
Age          7.5820194     94.514851
```

```
Education    3.2086797      34.291840
Urban       -0.9076486       5.082664
US           0.8498543       6.547575
```

The test MSE of bagging is 3.315656, and one can see that variables "ShelveLoc", "Price", "Advertising" and "CompPrice" are the 4 most important variables for predicting "Sales".

e)

```
> # First we fit the randomForest using the default "mtry". Note that by default,
> # randomForest() uses p/3 variables when building a random forest of
> # regression trees, and /sqrt(p) variables when building a random forest of
> # classification trees.
>
>
> rf.carseats = randomForest(Sales ~ ., data = Carseats.train, ntree = 500, importance = T)
> rf.pred = predict(rf.carseats, Carseats.test)
> mean((Carseats.test$Sales - rf.pred)^2)

[1] 3.652072

> importance(rf.carseats)

               %IncMSE IncNodePurity
CompPrice   13.9573828     137.05401
Income       4.2797590     129.56698
Advertising 19.0990059     196.89319
Population   1.4010573     106.10286
Price       30.3575555     310.19452
ShelveLoc   42.9651175     407.37688
Age          6.6493877     129.65096
Education   -0.1957945      57.54397
Urban       -1.6583554      11.93021
US           4.5035641      40.98320
```

When using default value for "mtry", the test MSE of randomForest is 3.652072, and variables "ShelveLoc", "Price", "Advertising" and "CompPrice" are the 4 most important variables for predicting "Sales".

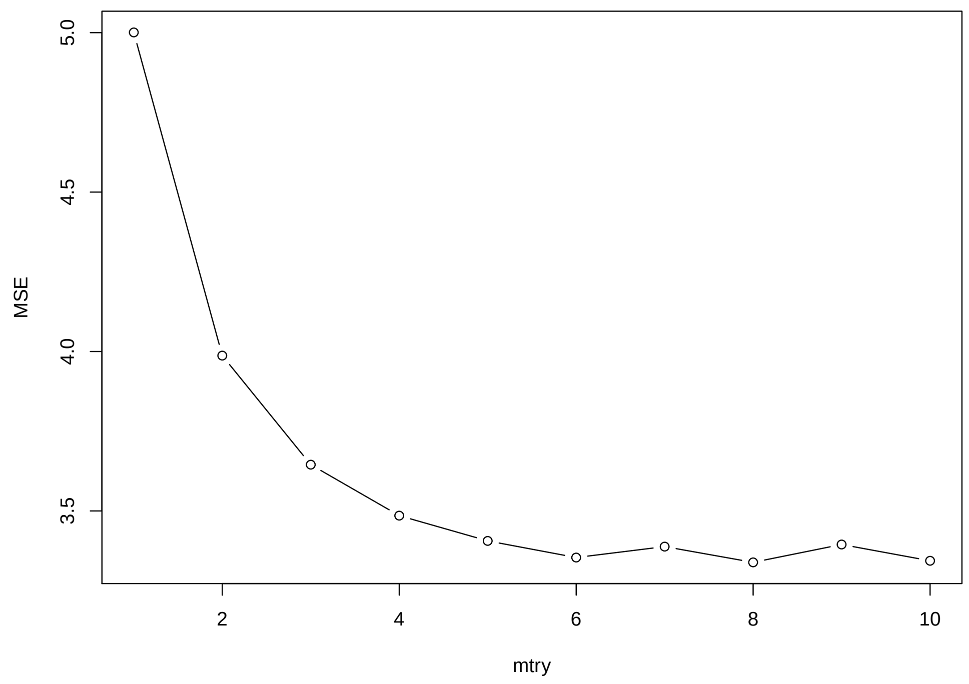
```
> # Now we try different "mtry"
> mse_vector <- rep(NA, p-1)
> for (i in 1:(p-1)) {
   rf.carseats = randomForest(Sales ~ ., data = Carseats.train, mtry=i, ntree = 500,
                              importance = T)
   rf.pred = predict(rf.carseats, Carseats.test)
   mse_vector[i] <- mean((Carseats.test$Sales - rf.pred)^2)
 }
> plot(1:(p-1), mse_vector, xlab="mtry", ylab="MSE", type="b")
> which.min(mse_vector)

[1] 8

> min(mse_vector)

[1] 3.338769

>
```



One can see that using different "mtry"s gives different test MSEs. In the above case, "mtry=8" gives the mininmal test MSE 3.338769.