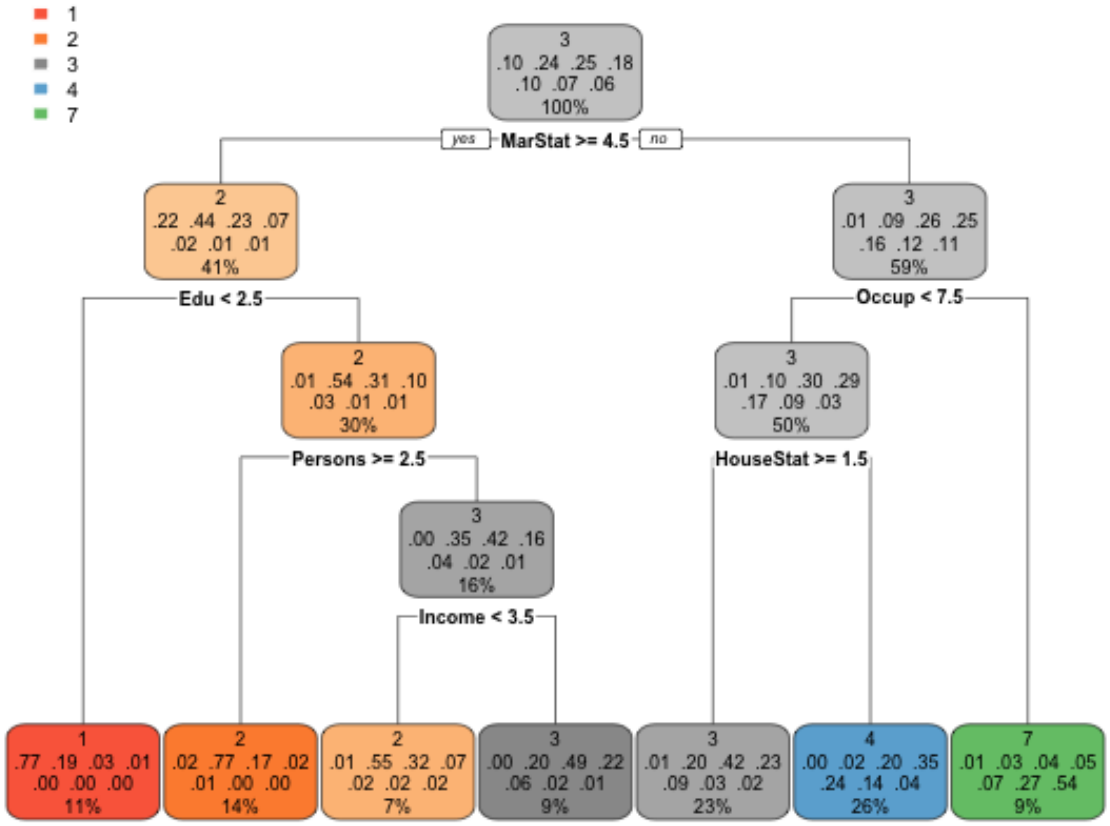
1. Write a short report on the relation between the age and the other demographic predictors.

The first split is on marital status. If you are single, then you are likely younger. Subsequent splits are on education, number of persons in the household, annual income of household, occupation, householder status. People who have received lower level of education are likely younger. People who are retired or unemployed are likely older. Among those single/never-married who have obtained a degree of high school or higher, if their annual household income is higher, then they are likely to be older. Among those non-single who are not retired/unemployed, they are likely older if they have their own household.



We first build a large tree so that we are able to use RPART's cross-validation feature to determine the optimal tree size. We use *printcp* to find the optimal value of the complexity parameter, which is about 0.010000, with 7 terminal nodes.

```
age.tree=rpart(age~.,data=age_data,method="class")
printcp(age.tree)
# Classification tree:
```

```
## rpart(formula = age ~ ., data = age_data, method = "class")

## Variables actually used in tree construction:
## [1] Edu       HouseStat Income    MarStat   Occup     Persons

## Root node error: 6536/8710 = 0.7504

## n= 8710

##         CP nsplit rel error  xerror      xstd
## 1 0.119339      0   1.00000 1.00000 0.0061797
## 2 0.088280      1   0.88066 0.90116 0.0066813
## 3 0.057834      2   0.79238 0.80018 0.0069939
## 4 0.050031      3   0.73455 0.74220 0.0070930
## 5 0.017977      4   0.68452 0.68069 0.0071378
## 6 0.010000      6   0.64856 0.65713 0.0071388
```

(a) Were surrogate splits used in the construction of the optimal tree you obtained? What does a surrogate split mean? Give an example of a surrogate split from your optimal decision tree. Which variable is the split on? which variable(s) is the surrogate split on?

Surrogate splits were used. When the data has some missing predictor values in some variables, we estimate the missing data using the other independent variables. These constructed variables that are used to impute the missing values are called surrogate variables. Surrogate splits exploit correlations between predictors to try and alleviate the effect of missing data. A surrogate split calculates a set of surrogate variables that can be used to estimate which branch of the tree we should traverse when the primary variable is missing.
By triggering $summary(age.tree)$, we discover that there are several surrogate splits in the tree we constructed. From part of the summary output, we can see that, for example, in the 2nd node of the tree, the split is on the education, but there are surrogate splits on the number of people in households under 18 and annual household income.

```
# Node number 2: 3610 observations,    complexity param=0.08828029
#  predicted class=2  expected loss=0.5551247  P(node) =0.4144661
```

```
#    class counts:   795  1606   826   258    70    34    21
#   probabilities: 0.220 0.445 0.229 0.071 0.019 0.009 0.006
#  left son=4 (994 obs) right son=5 (2616 obs)
#  Primary splits:
#      Edu       < 2.5 to the left,  improve=557.5415, (32 missing)
#      Under18   < 0.5 to the right, improve=296.0605, (0 missing)
#      HouseStat < 2.5 to the right, improve=271.2863, (78 missing)
#      Income    < 1.5 to the left,  improve=244.6477, (157 missing)
#      Persons   < 2.5 to the right, improve=199.5268, (168 missing)
#  Surrogate splits:
#      Under18 < 0.5 to the right, agree=0.801, adj=0.273, (32 split)
#      Income  < 1.5 to the left,  agree=0.737, adj=0.041, (0 split)
```

(b) Using your optimal decision tree, predict your age.

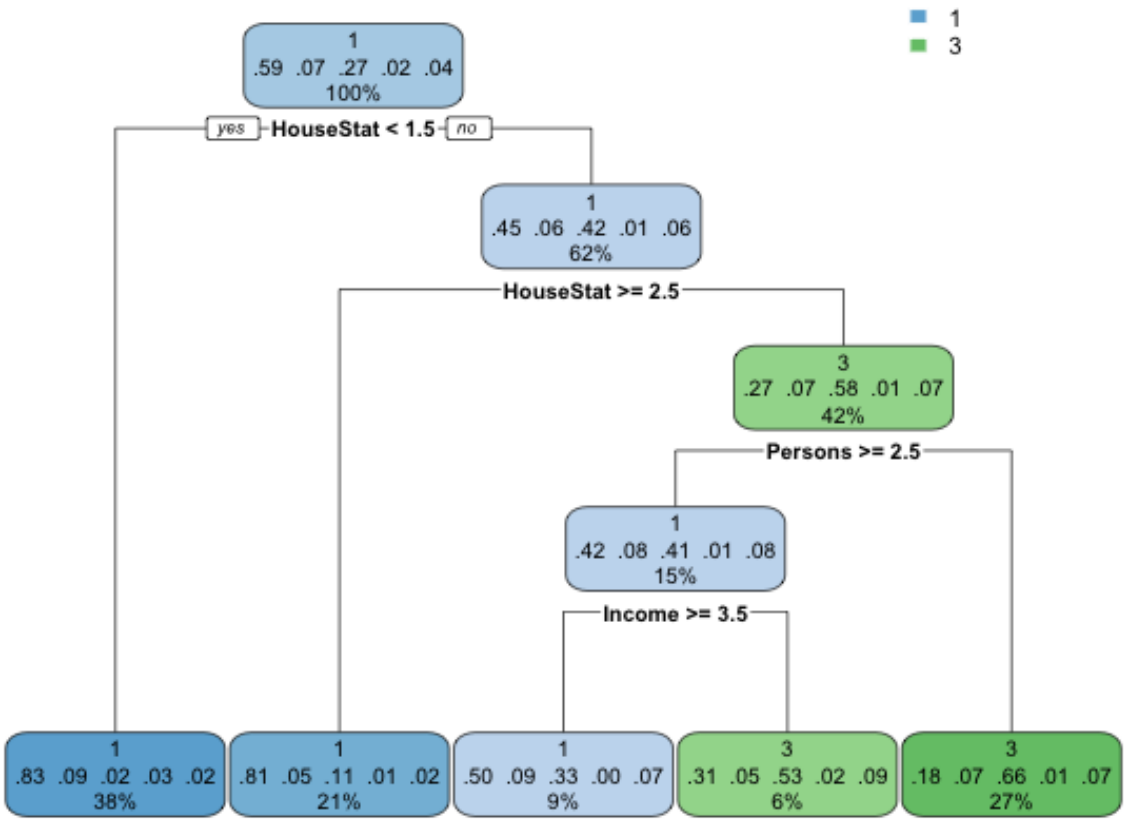
I use this tree model to predict my age, and the prediction is right to the truth.

```
new.obs=c(NA,6,1,2,5,6,1,2,1,1,0,2,2,1)
age_data[nrow(age_data)+1, ]=new.obs
age_data[nrow(age_data), ]
out<-predict(age.tree,age_data[nrow(age_data), ])
colnames(out)[max.col(out, ties.method=c("random"))]
#[1] "2"
# Bin 2 is for age 18 thru 24. That's the truth.
```

2. The goal in this problem is to construct a classification tree to predict the type of home from the other 13 demographics attributes. Give an estimate of the misclassification error of an optimal tree. Plot the optimal tree if possible (otherwise plot a smaller tree) and interpret the results.

We first train the tree model on the entire data set, and find that the misclassification error is 0.2620659. Then we randomly select 90% of the data set as training sample, and the remaining 10% as the test data set. We train the tree model on the training data set and get the same terminal nodes with slightly different split on the income node. The misclassification error on the training data set is 0.2635927, and the misclassification error of the trained tree model on the test data set is 0.248337. We estimate that the misclassification error is about $0.25 \sim 0.26$.

The tree tells us that we can predict the type of the house you are living in by looking at your household status, the number of persons you are living with and your annual household income. If the person owns his/her household, then that person is more likely to live in a house. Among those who don't have their own household, if they live with parents/family, they are more likely to live in a house. For the rest, if there are only one or two persons in the household, then you are more likely to live in an apartment; otherwise, if your annual household income is higher, then you are more likely to live a house, else you are more likely to live in an apartment.




```
#randomly select 90% of the data set as the training sample
train<-sample(1:nrow(housetype_data),0.9*nrow(housetype_data))
inputData<-housetype_data[train, ] #training data
testData<-housetype_data[-train, ] #test data

#train the tree model on the training data set
housetype.tree=rpart(TypeHome~.,data=inputData,method="class")
printcp(housetype.tree)

## Variables actually used in tree construction:
# [1] HouseStat Income    Persons

# Root node error: 3308/8111 = 0.40784

# n= 8111
```

```
#          CP nsplit rel error  xerror     xstd
# 1 0.158857      0   1.00000 1.00000 0.013379
# 2 0.017987      2   0.68229 0.68229 0.012201
# 3 0.010000      4   0.64631 0.65145 0.012025

#predict probabilities on the training data set
out<-predict(housetype.tree)

#predict response
pred.response<-colnames(out)[max.col(out, ties.method=c("random"))]

# misclassification error on the training data set
mean(inputData$TypeHome !=pred.response)
# [1] 0.2635927

# predict probabilities on the test set
out_test<-predict(housetype.tree,testData)

#prediction on the test data set
test.response<-colnames(out_test)[max.col(out_test,ties.method=c("random"))]

# misclassification error on the test data set
mean(testData$TypeHome !=test.response)
# [1] 0.248337
```

3. Suppose the loss for an incorrect classification prediction is the same regardless of either the predicted value c_k or the true value c_l of the outcome y. Show that in this case misclassification risk reduces to the classification error rate. What is the Bayes rule for this case in terms of the probabilities of y realizing each of its values $\{Pr(y = c_k)\}_{k=1}^{K}$? Derive this rule from the general (unequal loss) Bayes rule, for this particular loss structure $L_{kl} = 1(k \neq l)$.

Let L_{kl} be a loss associated with assigning it to the k-th group when the truth is that it's a member of group l. Then the expected loss, which is misclassification risk is

$$R_k = \sum_{l=1}^{K} L_{kl} Pr(c_l|\hat{y} = c_k) \tag{1}$$

where $Pr(l|\hat{y} = j)$ is the probability that y is a member of l-th group given that it's assigned into j-th group. The risk (1) is minimized by assigning y to group k^*, where

$$k^* = \arg \min_{1 \le k \le K} R_k \tag{2}$$

Since now $L_{kl} = \delta_{kl} = \mathbb{I}(k \neq l)$, (2) now reduces to

$$k^* = \arg \max_{1 \le k \le K} Pr(c_k|\hat{y} = c_k) \tag{3}$$

which means that the risk reduces to the classification error rate.(2),(3) are Bayes decision rules and their associated misclassification risk represents the minimum that we can achieve.

4. Does a low error rate using a classification rule derived by substituting probability estimates $\{\hat{Pr}(y = c_k)\}_{k=1}^{K}$ in place of the true probabilities $\{Pr(y = c_k)\}_{k=1}^{K}$ in the Bayes rule imply accurate estimates of those probabilities? Why?

This does not necessarily imply an accurate estimate of the probability since the Bayes decision rule is based on the class with the highest conditional probability. Consider the case that there are only two classes and we have the true probability $(0.99, 0.01)$ but the estimated probability $(0.51, 0.49)$. And suppose the probability of the class j object being classified as class j is a constant, which is $Pr(\hat{y} = c_1|y = c_1) = Pr(\hat{y} = c_2|y = c_2)$. Then since

$$Pr(y = c_j|\hat{y} = c_j) = \frac{Pr(\hat{y} = c_j|y = c_j)Pr(y = c_j)}{\sum_{k=1}^{2} Pr(\hat{y} = c_k|y = c_k)Pr(y = c_k)} = \frac{Pr(y = c_j)}{Pr(y = c_1) + Pr(y = c_2)}$$

The Bayes rule would still give the correct prediction but the estimated probabilities are very inaccurate.