Statistical Methods Quiz #4 (50 minutes)

'25. 12. 17.(Wednesday) Cadet number(교번): Name: Score:

<Instructions>

1. It is open-book, closed-web test.
2. You are not allowed to discuss on the problems during the exam.
3. Import *customer.csv* data first.(Use all the string variables as factors.)

1. Import *customer.csv* dataset. Use the **str** function to examine the structure of the *customer* data. How many observations and how many variables does the dataset contain? [15 points]

<R-code>

```
str(customer)
```

<Output>

```
'data.frame':       4018 obs. of  10 variables:
 $ Gender          : Factor w/ 2 levels "Female","Male": 2 2 1 2 1 2 2 1 1 2 ...
 $ Ever_Married    : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 1 2 2 1 2 ...
 $ Age             : int  40 25 46 27 65 47 65 40 43 43 ...
 $ Graduated       : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 2 2 ...
 $ Profession      : Factor w/ 9 levels "Artist","Doctor",..: 1 6 1 6 8 1 5 1 1 1 ...
 $ Work_Experience: int  7 7 2 1 0 9 0 8 1 0 ...
 $ Spending_Score : Factor w/ 3 levels "Average","High",..: 3 3 3 3 1 3 3 3 3 3 ...
 $ Family_Size     : int  1 4 1 3 4 2 4 1 1 2 ...
 $ Var_1           : Factor w/ 7 levels "Cat_1","Cat_2",..: 6 4 6 6 6 6 6 6 6 4 ...
 $ Segmentation    : Factor w/ 4 levels "A","B","C","D": 1 4 2 3 2 2 2 2 1 2 ...
```

The number of observations: 4018

The number of variables: 10

2. Randomly select 3500 observations from the *customer* data and save them as the training data(**customer.train**). Save the remaining observations as the test data(**customer.test**). [15 points]

<R-code>

```
n = nrow(customer)
Index = sample(1:n, 3500)
customer.train = customer[Index,]
customer.test = customer[-Index,]
```

3. Using the training data, fit a **classification tree** with **Segmentation** as the response variable and all other variables as explanatory variables. Also, visualize the fitted tree. [15 points]

<R-code>

```
library(tree)
model = tree(Segmentation ~ ., data = customer.train)
plot(model)
text(model)
```

<Output (plot)>

4. Using the test data, obtain the predicted response variable. Then, create a **confusion matrix** and compute the **accuracy**. [20 points]

<R-code>

```
pred = predict(model, customer.test, type = "class")
true = customer.test$Segmentation
table(pred, true)
mean(pred == true)
```

<Output>

```
     true
pred  A  B  C  D
   A 73 35 23 36
   B 27 28 47 11
   C  9 23 37 10
   D 27 23 16 93

[1] 0.4459459
```

Accuracy: 0.4459459

5. Using the training data, fit a **random forest** model with **Segmentation** as the response variable and all other variables as explanatory variables. (Set the number of variables randomly selected at each split to **3**.) Using the test data, obtain the predicted response variable. Then, create a **confusion matrix** and compute the **accuracy**. Based on accuracy, which model performs better: the **single decision tree** or the **random forest**? (Select one.) [20 points]

<R-code>

```
library(randomForest)
model2 = randomForest(Segmentation ~ ., data = customer.train,
                      mtry = 3)
pred = predict(model2, customer.test, type = "class")
true = customer.test$Segmentation
table(pred, true)
mean(pred == true)
```

<Output>

```
    true
pred  A  B  C  D
   A 68 37 19 40
   B 25 21 28 16
   C 17 36 66 13
   D 26 15 10 81

[1] 0.4555985
```

Accuracy: 0.4555985

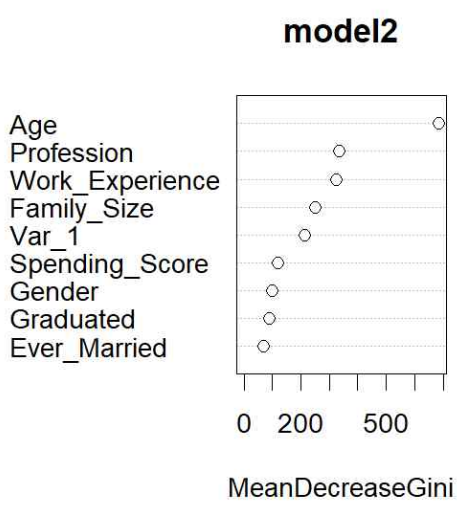
The (single decision tree / random forest) model has better performance.

6. For the random forest model, visualize the **variable importance**. Which variable is the most important? [15 points]

<R-code>

```
varImpPlot(model2)
```

<Output (plot)>



The most important variable: Age