BACS HW16
Student ID: 110065508

Due: 2022/06/05
Special thanks to 108071001

To begin with some preprocessing.

```
library(rpart)
library(rpart.plot)
library(magrittr)
library(dplyr)

dataset <- read.csv("insurance.csv")
dataset <- na.omit(dataset)
# Split the dataset
train_indices<- sample(1:nrow(dataset), size = 0.8*nrow(dataset))
train_set <- dataset[train_indices,]
test_set <- dataset[-train_indices,]
```

Question 1. (a) Create an OLS regression model.

```
# Question 1 (a)
ins_lm <- lm(charges ~ age + factor(sex) + bmi + factor(smoker) +
               factor(region), data=dataset)
report_lm <- summary(ins_lm)
write.table(round(report_lm$coefficients, digits=5),
            file="1a.csv",
            sep = ",",
            col.names=NA)
```

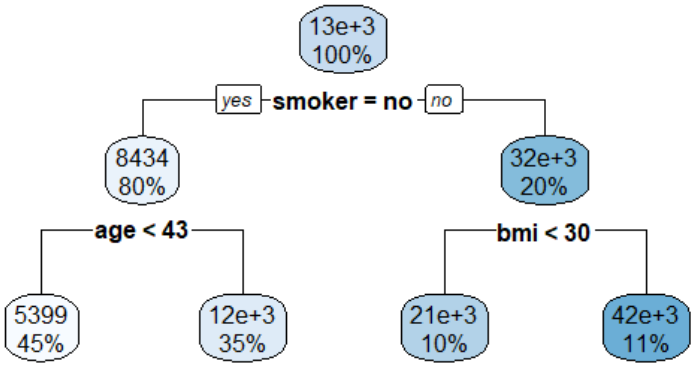
	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	-11556.95734	985.62619	-11.7255	0
age	258.5397	11.93749	21.65779	0
factor(sex)male	-111.57001	334.25839	-0.33378	0.7386
bmi	340.45918	28.71413	11.85685	0
factor(smoker)yes	23862.90634	414.82279	57.52554	0
factor(region)northwest	-304.10359	478.01318	-0.63618	0.52477
factor(region)southeast	-1039.2021	480.64936	-2.16208	0.03079
factor(region)southwest	-916.44143	479.72079	-1.91036	0.0563

(b) Create a decision (regression) tree.

```
library(rpart)
library(rpart.plot)
library(magrittr)
library(dplyr)

dataset <- read.csv("insurance.csv")
dataset <- na.omit(dataset)
# Split the dataset
train_indices<- sample(1:nrow(dataset), size = 0.8*nrow(dataset))
train_set <- dataset[train_indices,]
test_set <- dataset[-train_indices,]
```

(i) Plot a visual representation of the tree



(ii) The depth is 2.

(iii) There are 4 leaf groups

(iv) As the following table shows.

smoker, age<43	smoker, age>=43	non-smoker, bmi<30	non-smoker, bmi>=30
28675	36716	7977	8843

(v) Please refer to the plot in (i).

■

Question 2. The codes to compute the RMSE:

```
# Question 2
mse_oos <- function(actuals, preds) { # MSEs in this homework are all RMSE
  sqrt(mean( (actuals - preds)^2 ))
}

mse_oos(dataset$charges, predict(ins_lm, dataset))
mse_oos(dataset$charges, predict(ins_tree, dataset))
```

(a) The RMSE_{oos} for the OLS regression model is 6068.683.

(b) The RMSE_{oos} for the decision tree model is 4321.024. ■

Question 3. (a) Thanks to 108071021 for help of this part (3(a)) of codes

```
# Question 3 (a)
# Thanks to 108071021 for help of this part of codes
bagged_retrain <- function(model, dataset, b){
  resample <- unique(sample(1:nrow(dataset), replace = TRUE))
  train_data <- dataset[resample,]
  train_model <- update(model, data = train_data)
  train_model
}

bagged_learn <- function(model, dataset, b=100){
  lapply(1:b, bagged_retrain, model = model, dataset = dataset)
}

pred <- function(model, dataset, b){
  model = model[[b]]
  predict(model, dataset)
}

bagged_predict <- function(bagged_model, dataset, b){
```

```
  prediction <- lapply(1:b, pred, model = bagged_model, dataset = dataset)
  mse_oos(unlist(prediction), rep(unlist(dataset[7]), times = b))
}

# Question 3 (b)
lm_bagged_models <- bagged_learn(ins_lm, train_set, 100)
lm_bagged_mse <- bagged_predict(lm_bagged_models, test_set, 100)

# Question 3 (c)
ins_tree_models <- bagged_learn(ins_tree, train_set, 100)
ins_tree_bagged_mse <- bagged_predict(ins_tree_models, test_set, 100)
```

(b) The RMSE_{oos} for the bagged OLS regression model is 6654.777.

(c) The RMSE_{oos} for the bagged decision tree model is 5073.554. ■

Question 4. Thanks to 傅奕軒's code in the Microsoft Teams.

```
# Question 4 (a)
boosted_learn <- function(model, dataset, outcome, n=100, rate=0.1, type) {
  # get data frame of only predictor variables
  predictors <- dataset[, -which(names(dataset) %in% outcome)]
  # Initialize residuals and models
  res <- dataset[,outcome] # get vector of actuals to start
  models <- list()
  for (i in 1:n) {
    this_model <- update(model, data = cbind(charges = res, predictors))
    # update residuals with learning rate
    if (type == "l")
    {
      res <- res - rate * this_model$fitted.values
    }
    else
    {
      res <- res - rate * this_model$y
    }
    models[[i]] <- this_model # Store model
  }
  list(models=models, rate=rate)
}

boost_predict <- function(boosted_learning, new_data) {
  boosted_models <- boosted_learning$models
  rate <- boosted_learning$rate
  n <- length(boosted_models)

  # get predictions of new_data from each model
  predictions = lapply(1:n, function(i){
    rate*predict(boosted_models[[i]], new_data)
  })

  pred_frame = as.data.frame(predictions) #%>% unname
  pred <- apply(pred_frame, 1, sum)
  mse_oos(pred, new_data[,7]) # 7 for charges
}

# Question 4 (b)
```

```r
ins_lm_boosted_model <- boosted_learn(ins_lm, train_set, outcome = "charges",
                                      type = "l")
ins_lm_boosted_mse <- boost_predict(ins_lm_boosted_model, test_set)

# Question 4 (c)
ins_tree_stump <- rpart(charges ~ age + sex + bmi + smoker + region,
                        data=dataset, cp=0, maxdepth=1)
ins_tree_boosted_model <- boosted_learn(ins_tree_stump, train_set,
                                        outcome="charges", type='t')
ins_tree_boost_mse <- boost_predict(ins_tree_boosted_model, test_set)
```

(b) The RMSE_{oos} for the boosted OLS regression model is 6638.21.

(c) The RMSE_{oos} for the boosted decision tree model is 5166.023.

Note that the RMSE boosted method varies a lot. ■

Question 5. (a) The bagging of the decision tree

```r
# Question 5 (a)
flag <- 0 # break the loop or not
maxdepth_ins <- 1 # maxdepth of the tree
maxdepth_vector <- c()
bagged_mse <- c()

while(flag == 0){
  control <- rpart.control(maxdepth = maxdepth_ins, cp = 0)

  # Using code in Question 3 (c)
  ins_tree <- rpart(charges ~ bmi + age + sex + children + smoker + region,
                    data = train_set, control = control)
  ins_tree_models <- bagged_learn(ins_tree, train_set, 100)
  ins_tree_bagged_mse <- bagged_predict(ins_tree_models, test_set, 100)

  # append the result to the vector
  maxdepth_vector <- c(maxdepth_ins)
  bagged_mse <- c(bagged_mse, ins_tree_bagged_mse)

  # To determine to brake the loop or not
  if(length(bagged_mse) >= 2){
    if(bagged_mse[maxdepth_ins-1] < bagged_mse[maxdepth_ins]){
      flag <- 1 # break
    }
  }
  maxdepth_ins <- maxdepth_ins + 1
}

result_dataframe_5a <- cbind(maxdepth_vector, bagged_mse)
names(result_dataframe_5a) <- c("maxdepth", "bagged_mse")
write.table(result_dataframe_5a, file="5a.csv", sep = ",", col.names=NA)
```

maxdepth_vector	bagged_mse
1	7108.743
2	4775.779
3	4572.248
4	4473.494
5	4512.021

(b) The boosting of the decision tree

```r
# Question 5 (b)
flag <- 0 # break the loop or not
maxdepth_ins <- 1 # maxdepth of the tree
maxdepth_vector <- c()
boost_mse <- c()

while(flag == 0){
  control <- rpart.control(maxdepth = maxdepth_ins, cp = 0)

  # Using code in Question 4 (c)
  ins_tree_stump <- rpart(charges ~ age + sex + bmi + smoker + region,
                          data=dataset, control = control)
  ins_tree_boosted_model <- boosted_learn(ins_tree_stump, train_set,
                                          outcome="charges", type='t')
  ins_tree_boost_mse <- boost_predict(ins_tree_boosted_model, test_set)

  # append the result to the vector
  maxdepth_vector <- c(maxdepth_ins)
  boost_mse <- c(boost_mse, ins_tree_boost_mse)

  # To determine to brake the loop or not
  if(length(boost_mse) >= 2){
    if(boost_mse[maxdepth_ins-1] < boost_mse[maxdepth_ins]){
      flag <- 1 # break
    }
  }
  maxdepth_ins <- maxdepth_ins + 1
}

result_dataframe_5b <- cbind(maxdepth_vector, boost_mse)
names(result_dataframe_5b) <- c("maxdepth", "bagged_mse")
write.table(result_dataframe_5b, file="5b.csv", sep = ",", col.names=NA)
```

maxdepth_vector	boost_mse
1	7095.81
2	4730.264
3	4481.213
4	4414.42
5	4471.158

■