PubH 6886 Fall 2025 Assignment Four

AUTHOR
Annie Allred

```
library(tidyverse)
library(caret)
library(splines)
library(rpart)
library(rpart.plot)
library(randomForest)
library(gbm)
```

Question One

(35 pts.) Data in the file `bmd_females.csv` was collected on a sample of 259 adolescent females. Measures collected include `rspnbmd`: relative spinal bone mineral density measurements corresponding to the difference in spinal bone mineral density measurements taken at two consecutive visits divided by the average and `age`: (in years) corresponding to the average age over the two visits. For each part below, if you use the `train()` function to perform cross-validation, set the seed to 1234 (`set.seed(1234)`) prior to running the function.

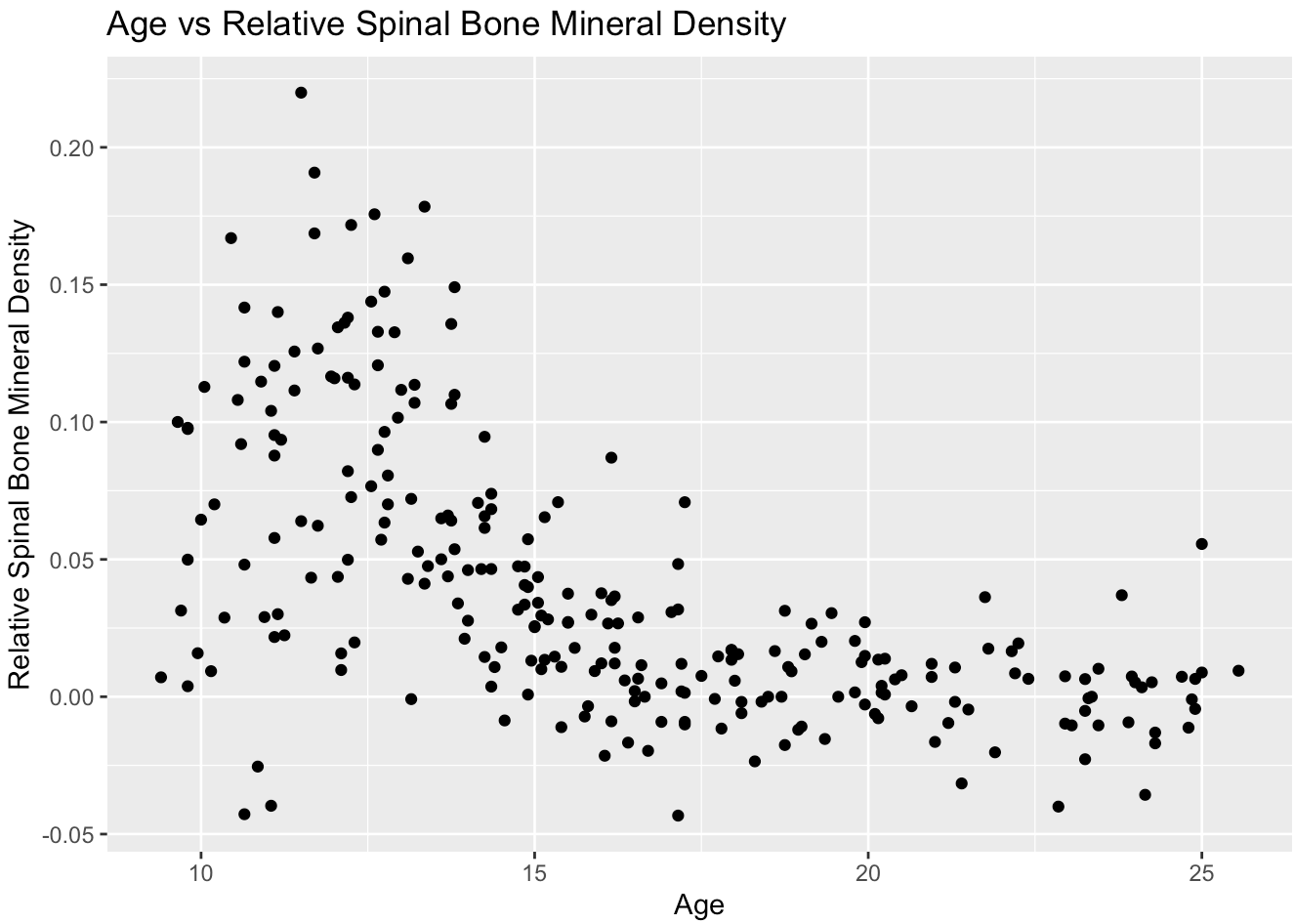
```
bmdF <- read.csv("bmd_females.csv")
```

(a)

(5 pts.) Make a scatter plot showing the relationship between `rspnbmd` and `age`. Clearly label the axes. Does there appear to be a linear relationship between `rspnbmd` and `age`?

- The relationship between `rspnbmd` and `age` does not appear to be linear. The relationship appears to be concave up, decreasing.

```
bmdF %>%
  ggplot(aes(x = age, y = rspnbmd)) +
  geom_point() +
  xlab("Age") +
  ylab("Relative Spinal Bone Mineral Density")+
  ggtitle("Age vs Relative Spinal Bone Mineral Density")
```



(b)

(10 pts.) Regardless of your response in part (a), fit a linear regression model with `rspnbmd` as the response and with `age` as the only predictor. Use 10-fold CV to compute the CV RMSE and CV R^2.

- The 10-fold CV RMSE for a univariate linear regression model with `rspnbmd` as the response and `age` as the only predictor is 0.040. The 10-fold CV $R^2 = 0.404$.

```
# setting the seed
set.seed(1234)
age_X <- cbind(age=bmdF$age)
# using train() to find the CV RMSE and R^2 values
q1_lm <- train(x = age_X, y = bmdF$rspnbmd, method = "lm",
               trControl = trainControl(method = "cv", number = 10,
                                        savePredictions = TRUE))
q1_lm
```

```
Linear Regression

259 samples
  1 predictor

No pre-processing
```

```
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 231, 232, 233, 232, 232, 235, ...
Resampling results:

  RMSE        Rsquared   MAE
  0.04033105  0.4035097  0.03045912

Tuning parameter 'intercept' was held constant at a value of TRUE
```

(c)

(20 pts.) Fit a cubic spline model with `rspnbmd` as the response and `age` as the only predictor. To arrive at your final model, use 10-fold CV RMSE to select the number of internal knots (consider values from 1 to 7). Provide a table or graph that shows how you selected the number of internal knots. Provide a graphical description of the final model that you select and report the CV RMSE and CV R^2. Does your selected cubic spline model predict `rspnbmd` more accurately than the linear model that you fit in part (b)?

- In order to determine how many knots is optimal for a cubic spline in this data set, seven models were created where the only difference between then was the number of knots. Their model matrices were then ran the train() function. Of the seven models, the model with five knots had the smallest 10-fold CV RMSE, which was 0.0349. The R^2 value of the 3 knot model is 0.534, which also happened to be the largest R^2 value of the seven models.

- The cubic spline model with five knots is more accurate than the linear regression model. This is because the 10-fold CV cubic spline RMSE value is smaller than the 10-fold CV linear regression RMSE value.

```
# cubic splines with internal knots
q1_k1 <- lm(rspnbmd ~ bs(age, df = 4, degree = 3), data = bmdF) # 1 knot
q1_k2 <- lm(rspnbmd ~ bs(age, df = 5, degree = 3), data = bmdF) # 2 knots
q1_k3 <- lm(rspnbmd ~ bs(age, df = 6, degree = 3), data = bmdF) # 3 knots
q1_k4 <- lm(rspnbmd ~ bs(age, df = 7, degree = 3), data = bmdF) # 4 knots
q1_k5 <- lm(rspnbmd ~ bs(age, df = 8, degree = 3), data = bmdF) # 5 knots
q1_k6 <- lm(rspnbmd ~ bs(age, df = 9, degree = 3), data = bmdF) # 6 knots
q1_k7 <- lm(rspnbmd ~ bs(age, df = 10, degree = 3), data = bmdF) # 7 knots

# getting the 10-fold CV RMSE and R^2 values for each of the above models
set.seed(1234)
q1_k1_train <- train(x = model.matrix(q1_k1)[,-1], y = bmdF$rspnbmd,
                     method = "lm",
                     trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k2_train <- train(x = model.matrix(q1_k2)[,-1], y = bmdF$rspnbmd,
                     method = "lm",
                     trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k3_train <- train(x = model.matrix(q1_k3)[,-1], y = bmdF$rspnbmd,
                     method = "lm",
                     trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k4_train <- train(x = model.matrix(q1_k4)[,-1], y = bmdF$rspnbmd,
```

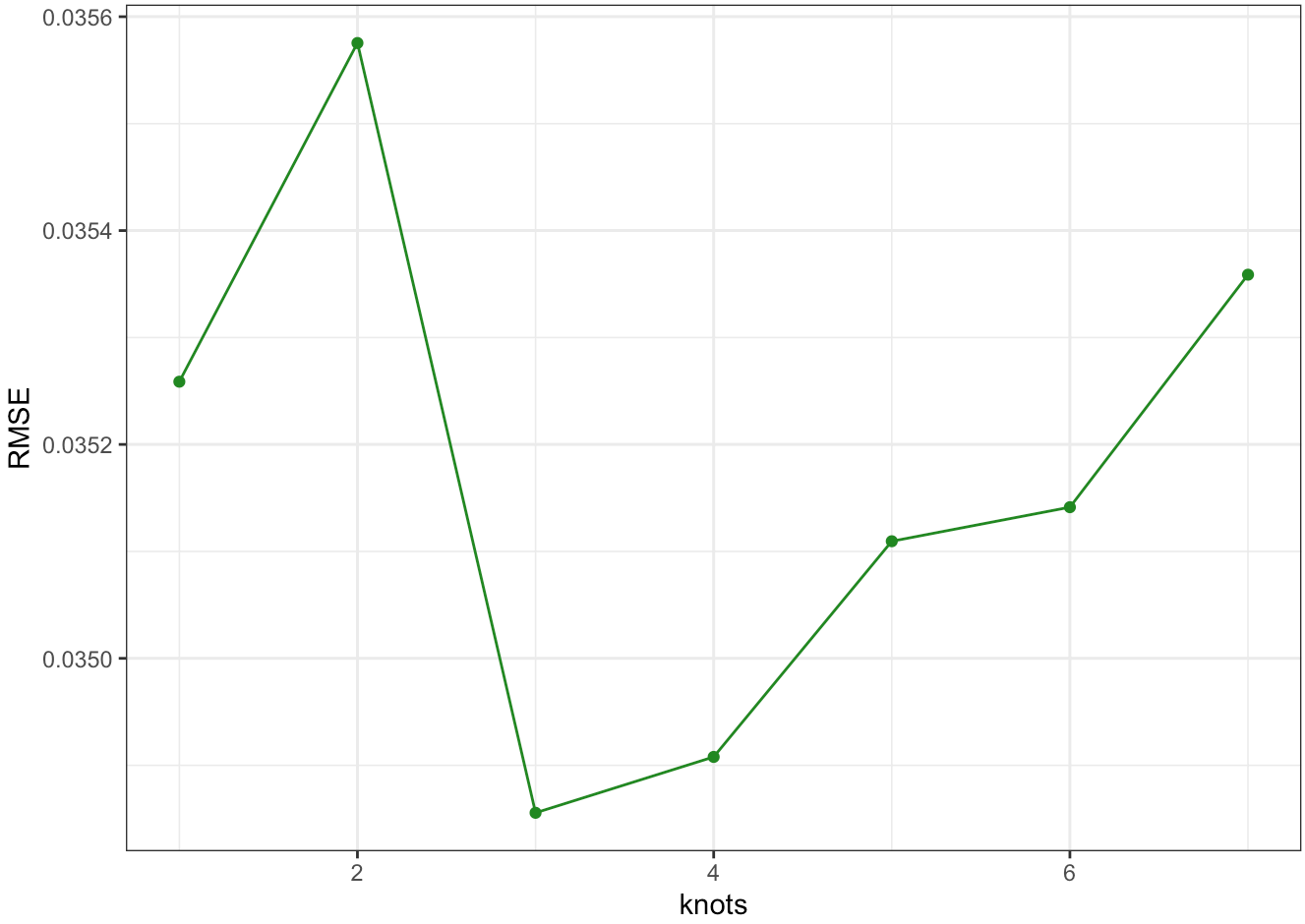
```
                    method = "lm",
                    trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k5_train <- train(x = model.matrix(q1_k5)[,-1], y = bmdF$rspnbmd,
                    method = "lm",
                    trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k6_train <- train(x = model.matrix(q1_k6)[,-1], y = bmdF$rspnbmd,
                    method = "lm",
                    trControl = trainControl(method = "cv", number = 10))
set.seed(1234)
q1_k7_train <- train(x = model.matrix(q1_k7)[,-1], y = bmdF$rspnbmd,
                    method = "lm",
                    trControl = trainControl(method = "cv", number = 10))

# combing results into a data frame
cv_knots <- rbind(q1_k1_train$results, q1_k2_train$results, q1_k3_train$results,
                  q1_k4_train$results, q1_k5_train$results, q1_k6_train$results,
                  q1_k7_train$results)
cv_knots <- cbind(knots = 1:7, cv_knots)
# the results of the seven models, indexed by the number of knots
cv_knots
```

```
  knots intercept       RMSE  Rsquared        MAE      RMSESD RsquaredSD
1     1      TRUE 0.03525868 0.5253990 0.02542427 0.005963497  0.1789007
2     2      TRUE 0.03557537 0.5187825 0.02551802 0.006072972  0.1819790
3     3      TRUE 0.03485556 0.5337522 0.02537922 0.005778045  0.1737838
4     4      TRUE 0.03490781 0.5310495 0.02544817 0.005509287  0.1653186
5     5      TRUE 0.03510949 0.5257104 0.02567362 0.005552688  0.1682096
6     6      TRUE 0.03514133 0.5242406 0.02584874 0.005611564  0.1677688
7     7      TRUE 0.03535874 0.5188705 0.02594829 0.005476241  0.1639513
        MAESD
1 0.004586092
2 0.004595407
3 0.004758885
4 0.004417425
5 0.004649612
6 0.004433831
7 0.004214646
```

```
# finds which model had the lowest RMSE
min(cv_knots$RMSE)
```

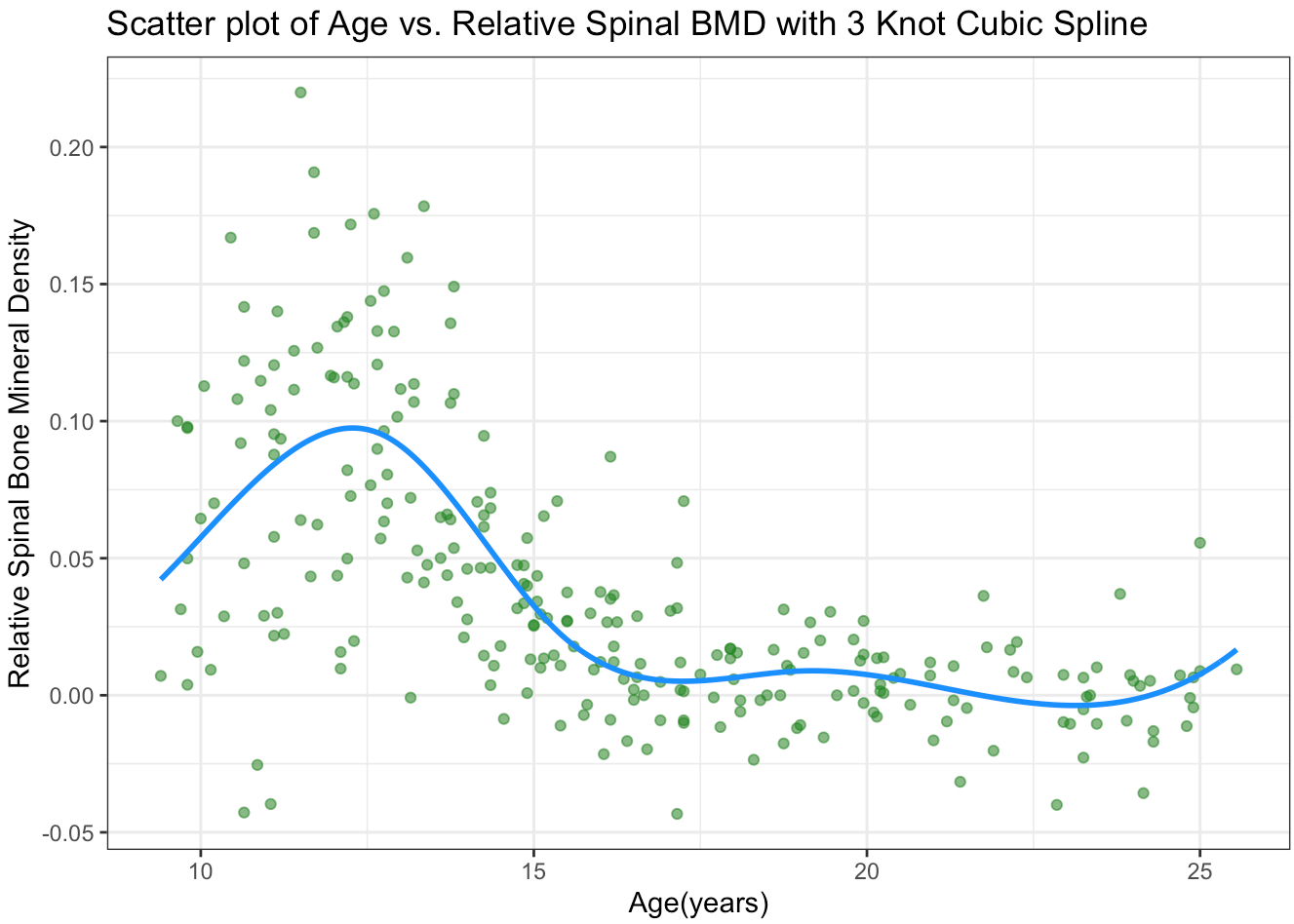
```
[1] 0.03485556
```

```
cv_knots %>%
  ggplot(aes(x = knots, y = RMSE)) +
  geom_point(col = "forestgreen") +
  geom_line(col = "forestgreen") + theme_bw()
```




```
age_grid1 <- seq(min(bmdF$age), max(bmdF$age), length.out= 259)
set.seed(1234)
q1_k3_pred <- predict(q1_k3, newdata = data.frame(age = age_grid1))

q1_final <- data.frame(age = age_grid1, predicted_bmd = q1_k3_pred)

ggplot(bmdF, aes(x = age, y = rspnbmd)) +
  geom_point(color = "forestgreen", alpha = 0.6) +
  geom_line(data = q1_final,
            aes(x = age, y = predicted_bmd),
            size = 1,
            col = "dodgerblue") +
  labs(title = "Scatter plot of Age vs. Relative Spinal BMD with 3 Knot Cubic Spline",
       x = "Age(years)",
       y = "Relative Spinal Bone Mineral Density") +
  theme_bw()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```



Question Two

(80 pts.) Data in the file `sahd_data.csv` was collected on a sample of 462 males in a heart-disease high-risk region of the Western Cape, South Africa. Information about the variables in the data set can be found here: https://hastie.su.domains/ElemStatLearn/datasets/SAheart.info.txt. You will use these data construct and evaluate several models to predict coronary heart disease status (`chd` = 1 if yes, 0 if no) based on the other variables available in the data set. For each part below, if you use the `train()` function to perform cross-validation, set the seed to 1234 (`set.seed(1234)`) prior to running the function. For each part below, if you use the `train()` function to perform cross-validation, set the seed to 1234 (`set.seed(1234)`) prior to running the function. Note that you should convert the `famhist` values from Present/Absent to 1/0 and you should convert the `chd` values from 0/1 to no/yes and convert it to a factor variable in R.

```
sahd <- read.csv("sahd_data.csv") %>%
  mutate(chd = factor(chd, levels = c(0,1), labels = c("no", "yes")),
         famhist = factor(famhist, levels =  c("Present", "Absent"),
                          labels = c(1,0)))
#str(sahd)
```

(a)

(10 pts.) Fit a logistic model with coronary heart disease status as the response and the other variables as potential predictors. Use a threshold of 0.5 and report the 10-fold CV accuracy for this model.

- The 10-fold CV accuracy for a logistic classification model is 0.716.

```
set.seed(1234)
q2_glm <- train(chd ~ ., data = sahd, method = "glm", family = "binomial",
                trControl = trainControl(method = "cv", number = 10))
q2_glm
```

```
Generalized Linear Model

462 samples
  9 predictor
  2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 416, 416, 415, 416, 415, 416, ...
Resampling results:

  Accuracy   Kappa
  0.7164662  0.3470991
```

(b)

(20 pts.) Grow and prune a classification tree with coronary heart disease status as the response and the other variables as potential predictors. Use 10-fold CV accuracy with the one-SE rule to determine the optimal amount of pruning. Use the Gini index for splitting nodes and set the minimum number of observations to split a node to 30 and the minimum number of observations in each terminal node to 10. Plot the classification tree and provide the 10-fold CV accuracy.

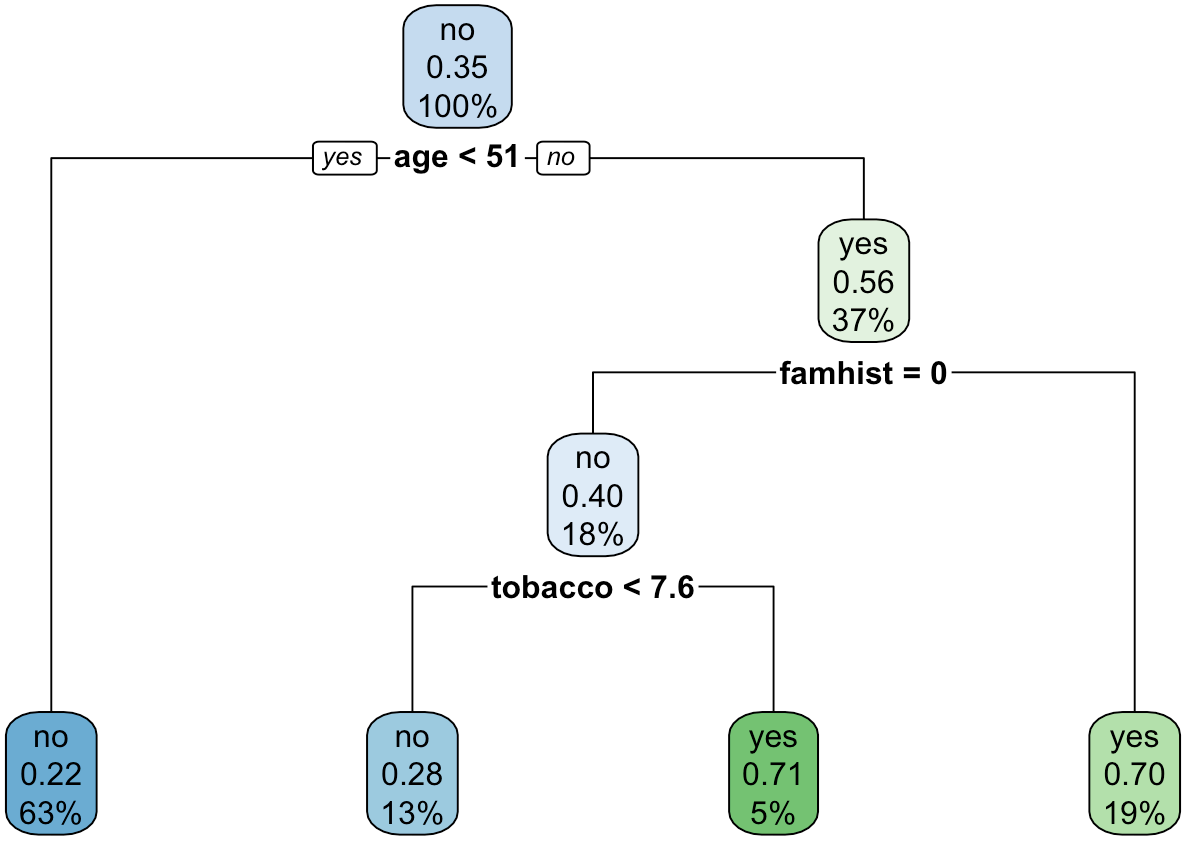
- Using the one-SE rule from 10-fold CV, the classification tree with $alpha = 0.025$ is the best one. The 10-fold CV accuracy is 0.723

```
set.seed(1234)
q2_rtree <- rpart(chd ~ ., data = sahd, method = "class",
                  parms = list(split = "gini"),
                  control = rpart.control(minsplit = 30, minbucket = 10))
#q2_rtree
# printcp(q2_rtree)
# plotcp(q2_rtree)

# create tuning grid from alphas chosen from rpart
tg_rtree <- data.frame(cp = q2_rtree$cptable[,1])
# the x-value for train
sahd_x <- sahd[,1:9]
# the y-value for train
sahd_y <- sahd$chd
```

```r
set.seed(1234)
#running the train() function
q2_rtree_train <- train(x = sahd_x, y = sahd_y, method = "rpart",
                        parms = list(split = "gini"),
                        control = rpart.control(minsplit = 30, minbucket = 10),
                        tuneGrid = tg_rtree,
                        trControl = trainControl(method = "cv", number = 10,
                                                 selectionFunction = "oneSE"))
q2_rtree_train
```

```
CART

462 samples
  9 predictor
  2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 416, 416, 415, 416, 415, 416, ...
Resampling results across tuning parameters:

  cp       Accuracy   Kappa
  0.01000  0.7141536  0.32757419
  0.01875  0.7228955  0.34350898
  0.02500  0.7229880  0.34691413
  0.06250  0.7056429  0.29261753
  0.10000  0.6734505  0.21943824
  0.12500  0.6430157  0.08046635

Accuracy was used to select the optimal model using  the one SE rule.
The final value used for the model was cp = 0.025.
```

```r
# a plot of the classification tree
rpart.plot(q2_rtree_train$finalModel)
```



(c)

(20 pts.) Fit a random forest classification model of 500 trees with coronary heart disease status as the response and the other variables as potential predictors. Use 10-fold CV accuracy to select the number of predictors to consider at each split in any given tree of the random forest. For your selected random forest, provide a relevant plot that provides evidence that 500 trees is enough for this random forest. (Hint, the `finalModel` object has an element `err.rate` with each row providing information on OOB error rate for the corresponding row's number of trees from 1 to 500.) Report the OOB accuracy for the selected model. Present a variable importance plot based on mean decrease in Gini index. What are the top three predictors of coronary heart disease status based on the random forest model that you selected?

- Using the 10-fold CV, the random forest classification model with 500 trees best predicts coronary heart disease status when only one predictor is considered at each split. The 10-fold CV OOB accuracy is 0.703.
- The three most important predictors, relative to all other included variables are cumulative tobacco use (tobacco), age at onset (age), and low density lipoprotein cholesterol (ldl).
- Since the plot levels out, this is evidence that 500 is sufficient to run the model.

```
set.seed(1234)
# random forest
q2_forest_train <- train(x = sahd_x, y = sahd_y, method ="rf", ntree = 500,
```

```
                      tuneGrid = data.frame(mtry = 1:9),
                      trControl = trainControl(method = "cv", number = 10))
```

```
q2_forest_train
```

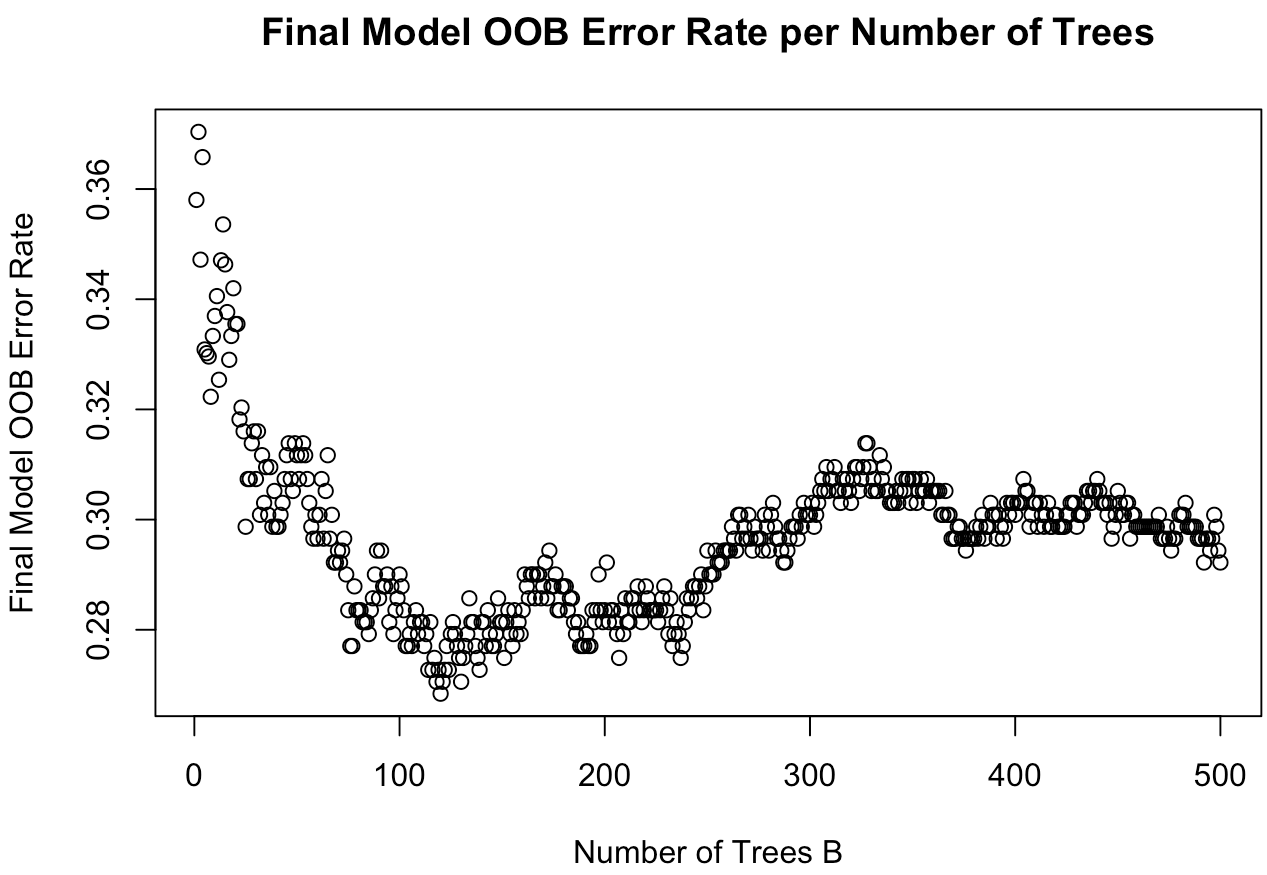
```
Random Forest

462 samples
  9 predictor
  2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 416, 416, 415, 416, 415, 416, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  1     0.7034690  0.2761715
  2     0.6839963  0.2519052
  3     0.6925069  0.2759361
  4     0.6775208  0.2453919
  5     0.6754394  0.2408634
  6     0.6839500  0.2614664
  7     0.6861702  0.2723560
  8     0.6797872  0.2503938
  9     0.6861240  0.2702533

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 1.
```

```
q2_forest_train$finalModel
```

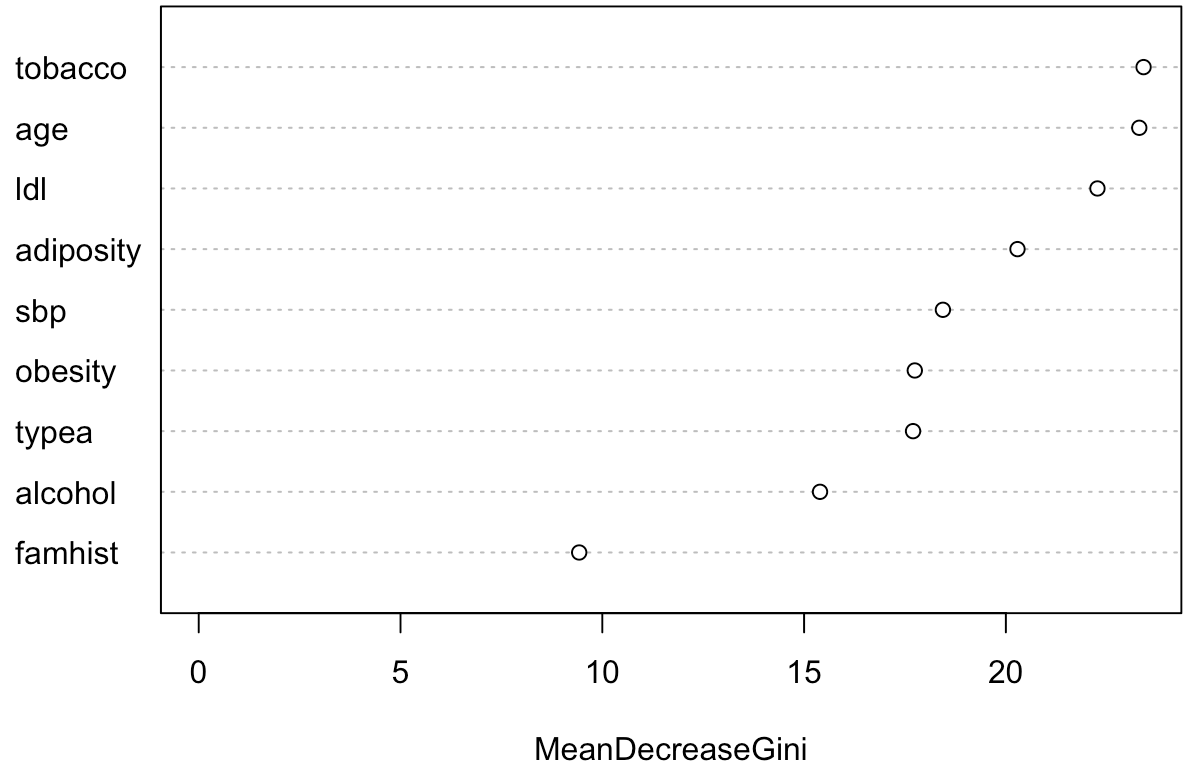
```

Call:
 randomForest(x = x, y = y, ntree = 500, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 29.22%
Confusion matrix:
     no yes class.error
no  269  33   0.1092715
yes 102  58   0.6375000
```

```
# the plot of error rate per number of trees
plot(x = 1:500, y = q2_forest_train$finalModel$err.rate[,1],
```

```
     xlab = "Number of Trees B", ylab = "Final Model OOB Error Rate",
     main = "Final Model OOB Error Rate per Number of Trees")
```




```
varImpPlot(q2_forest_train$finalModel, type = 2, main = "")
```



(d)

(25 pts.) Fit a boosted classification tree with coronary heart disease status as the response and the other variables as potential predictors. Use 10-fold CV accuracy to select the optimal combination of tuning parameters from the following: for the number of trees consider 5 to 50 in increments of 5; for the number of splits for the trees consider 1 - 6; for the shrinkage parameter consider 0.20, 0.25, 0.30, 0.35, 0.40, and 0.45. Set the minimum number of observations in each terminal node to 10 and sample 50% of the training data before fitting each new tree. Report the optimal tuning parameter combination and the corresponding 10-fold CV accuracy. Present a variable importance plot based on relative influence and construct partial dependence plots for the three most important predictors in the data set. Based on the partial dependence plots, what can be said about the relationship between each of the three predictor variables and coronary heart disease status?

- Using the 10-fold CV, the optimal tuning parameter combination has 35 trees, an interaction depth of 1, and a shrinkage parameter of 0.3. The minimum number of observations in each terminal node was given as 10. The corresponding 10-fold CV accuracy is 0.738.

- From on the variable importance plot based on relative influence, we can see that three most important predictors, relative to all other variables in the data set, are age at onset (age), cumulative tobacco use (tobacco), and low density lipoprotein cholesterol (ldl).

    - After adjusting for all other variables in the dataset, the log odds of not having coronary heart disease decreases as age at onset increases. In other words, the odds of having coronary heart disease increases as

age at onset increases.

- After adjusting for all other variables in the dataset, the log odds of not having coronary heart disease generally decreases as the cumulative amount of tobacco consumed increases. In other words, the odds of having coronary heart disease increases as the cumulative amount of tobacco consumed increases.

- After adjusting for all other variables in the dataset, the log odds of not having coronary heart disease deceases as low density lipoprotein cholesterol increases. In other words, the odds of having coronary heart disease increases as low density lipoprotein cholesterol increases.

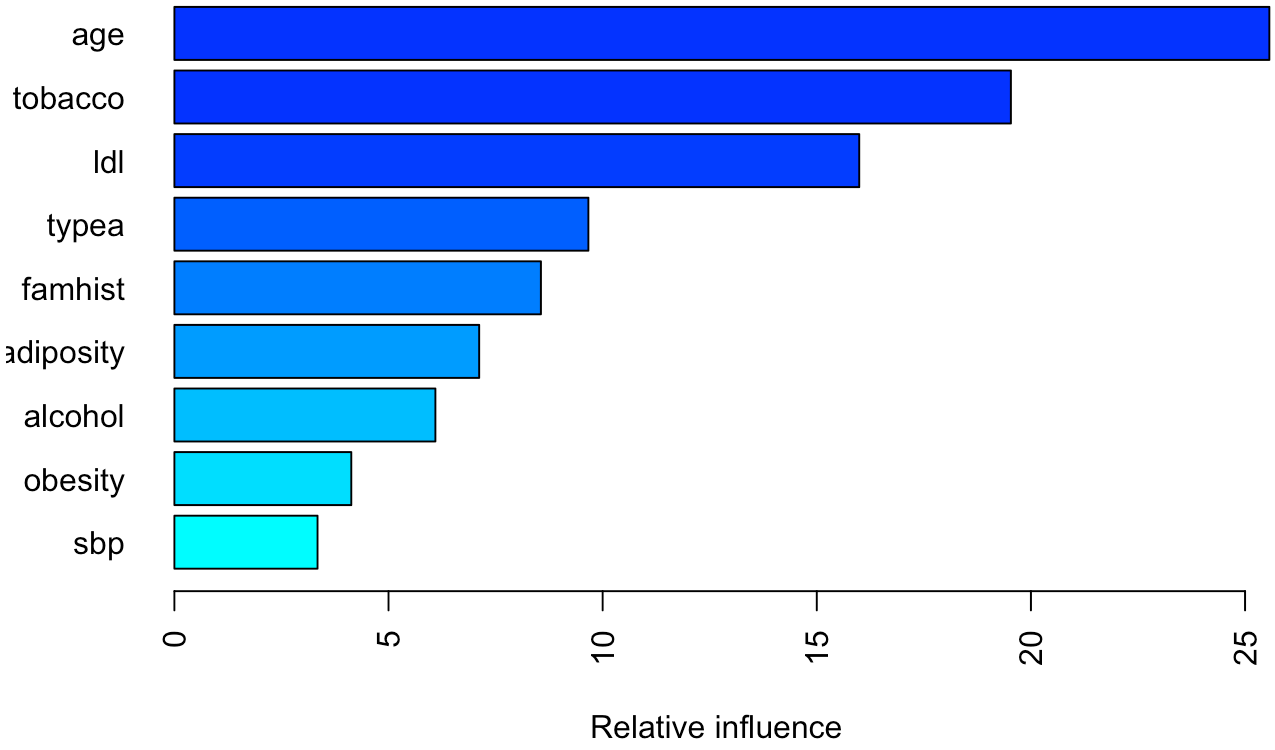
```
# setting up tuning grid
tg_boostedClass <- expand.grid(n.trees = seq(5, 50, 5),
                               interaction.depth = 1:6,
                               shrinkage = c(0.20, 0.25, 0.30, 0.35, 0.40, 0.45),
                               n.minobsinnode = 10)
```

```
set.seed(1234)
# 10-fold CV boosted classification tree using train()
q2_bClass_train <- train(x = sahd_x, y = sahd_y, method = "gbm",
                         bag.fraction = 0.50, tuneGrid = tg_boostedClass,
                         trControl = trainControl(method = "cv", number = 10))
```

```
# getting the accuracy and optimal combination of tuning parameters
q2_bClass_train$results[which.max(q2_bClass_train$results$Accuracy),]
```

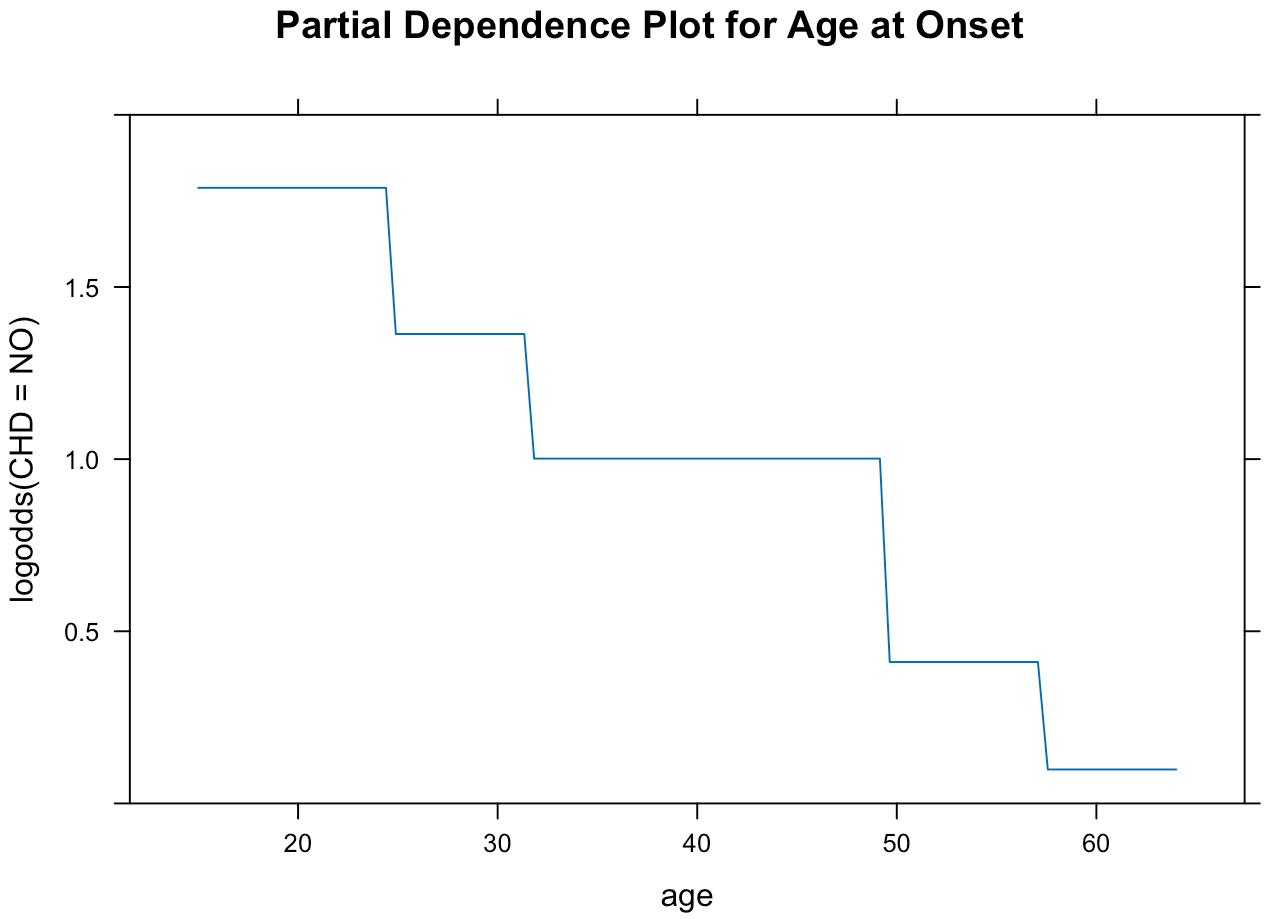
```
    shrinkage interaction.depth n.minobsinnode n.trees  Accuracy    Kappa
127       0.3                 1             10      35 0.7381129 0.395246
    AccuracySD   KappaSD
127 0.05992985 0.1359651
```

```
# The relative influence plot
summary(q2_bClass_train$finalModel, method = relative.influence,
        normalize = TRUE, las = 2)
```

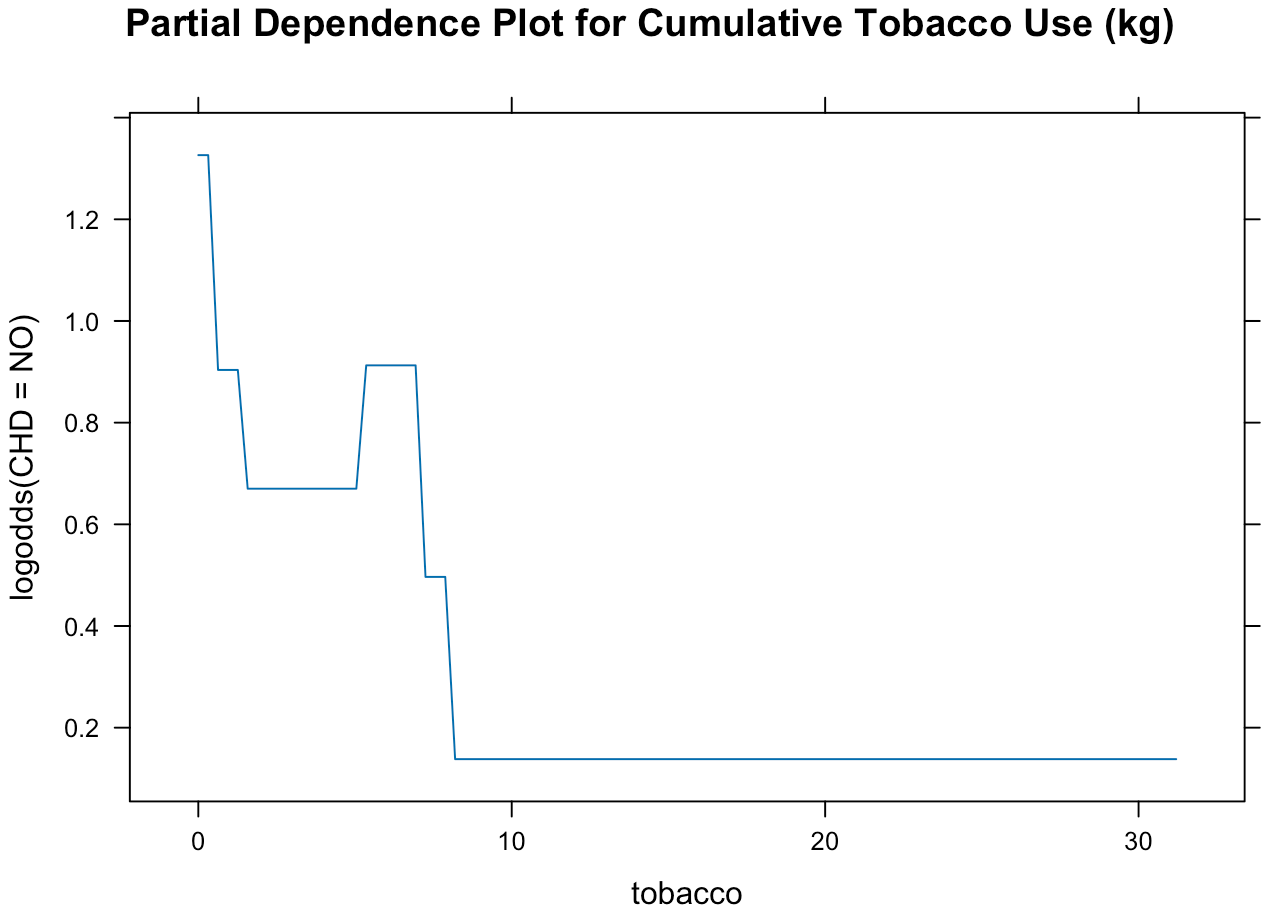


	var	rel.inf
age	age	25.567394
tobacco	tobacco	19.532306
ldl	ldl	15.991037
typea	typea	9.666861
famhist	famhist	8.558206
adiposity	adiposity	7.116763
alcohol	alcohol	6.094453
obesity	obesity	4.130542
sbp	sbp	3.342438

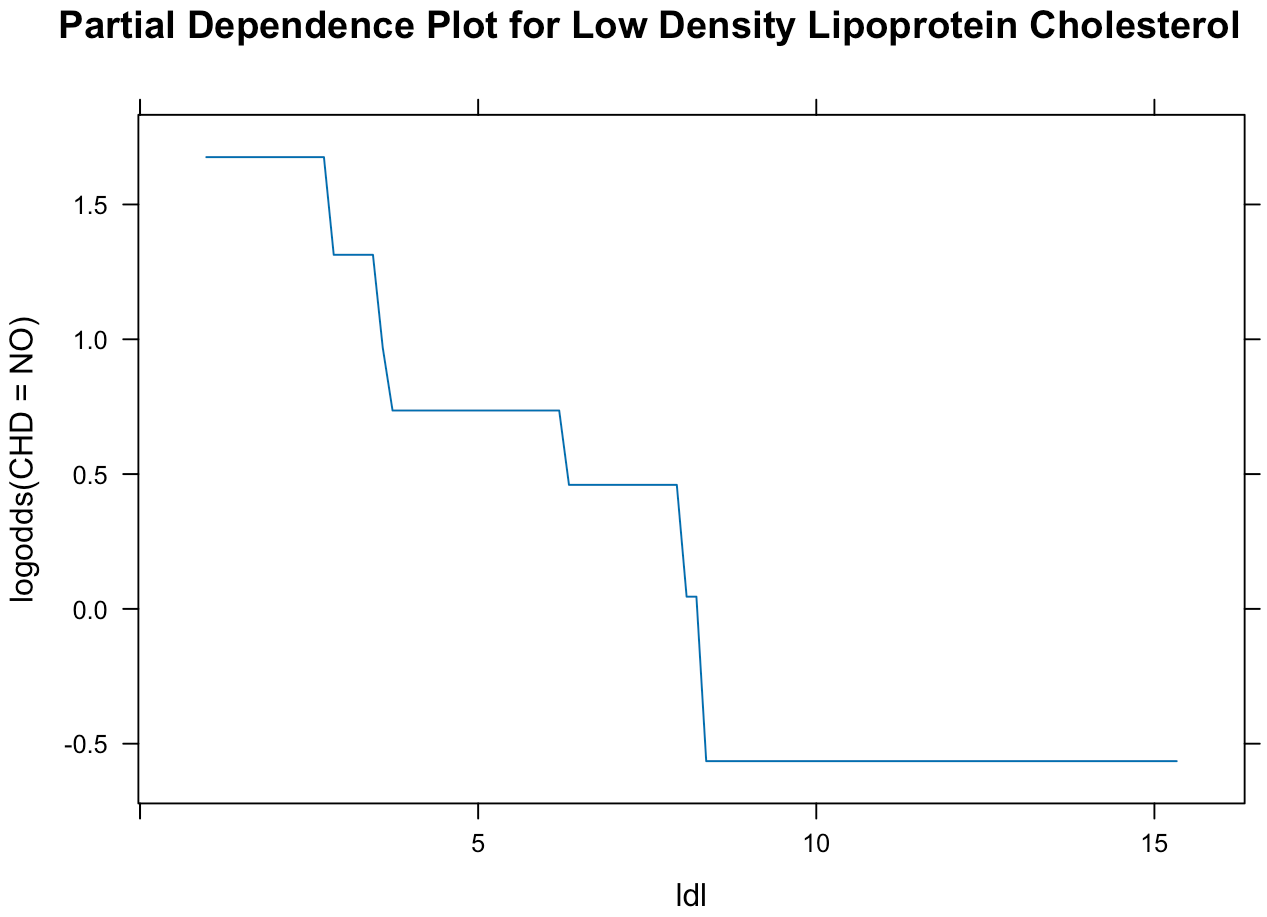
```
par(mfrow = c(2, 2))
plot(q2_bClass_train$finalModel, i = "age", ylim = c(0,2),
     ylab = "logodds(CHD = NO)",
     main = "Partial Dependence Plot for Age at Onset")
```




```
plot(q2_bClass_train$finalModel, i = "tobacco",
     ylab = "logodds(CHD = NO)",
     main = "Partial Dependence Plot for Cumulative Tobacco Use (kg)")
```




```
plot(q2_bClass_train$finalModel, i = "ldl",
     ylab = "logodds(CHD = NO)",
     main = "Partial Dependence Plot for Low Density Lipoprotein Cholesterol")
```



(e)

(5 points) Of the models considered in parts (a) - (d), which is the best with respect to prediction accuracy? Justify your response.

- Of the four model considered above, the best model with respect to prediction accuracy is the boosted classification tree model. This is because it has the highest accuracy rate of the four models.