Tree-based methods

Load the packages that we will use in the workshop

```
library(caret)
library(tidyverse)
library(rpart)
library(rpart.plot)
library(ipred)
library(bst)
library(ranger)
```

Classification trees

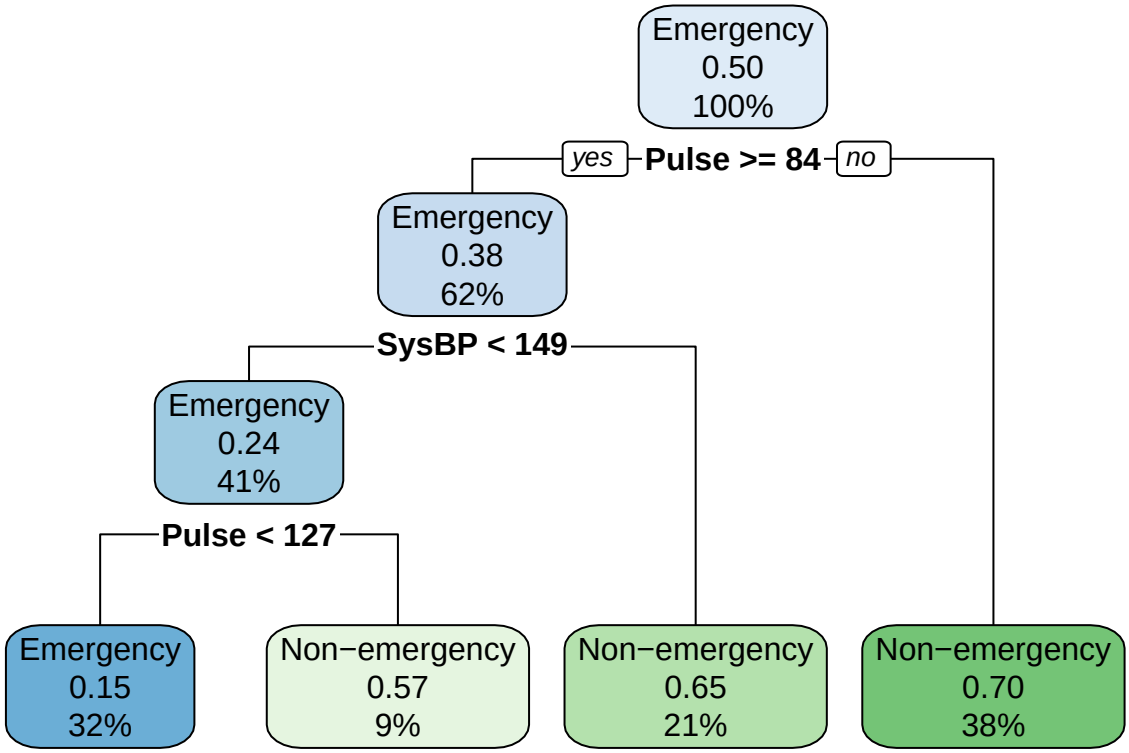
Exercise 1 - Classification trees using rpart package

Using the rpart() function, create a classification tree using the adapted ICU data (made from the ICU dataset available in the Stat2Data package).

```
# Read in the data
ICU = read.delim("ICU.adapted.txt")

# Check the structure of the data
str(ICU)
```

```
## 'data.frame':    80 obs. of  10 variables:
##  $ X        : int  38 104 107 40 79 70 110 157 188 116 ...
##  $ ID       : int  170 438 442 180 331 292 462 727 877 505 ...
##  $ Survive  : int  1 1 0 1 0 1 1 1 1 1 ...
##  $ Age      : int  51 80 69 64 63 61 69 75 88 40 ...
##  $ AgeGroup : int  2 3 2 2 2 2 2 3 3 1 ...
##  $ Sex      : int  0 1 1 1 1 0 1 0 1 0 ...
##  $ Infection: int  0 0 0 1 1 1 0 0 0 0 ...
##  $ SysBP    : int  110 162 170 158 36 68 150 144 190 130 ...
##  $ Pulse    : int  99 44 60 90 86 124 85 120 88 65 ...
##  $ Emergency: Factor w/ 2 levels "Emergency","Non-emergency": 1 1 1 1 1 1 1 1 1 1 ...
```

```
ICU.tree = rpart(Emergency ~ SysBP + Pulse, data = ICU)

# Plot the tree using the rpart.plot() function
# Shows predicted class, predicted probability of class, percentage of observations in node

rpart.plot(ICU.tree)
```




```
# What are the splitting rules for this tree?
rpart.rules(ICU.tree)
```

```
##  Emergency
##       0.15 when Pulse is 84 to 127 & SysBP <  149
##       0.57 when Pulse >=       127 & SysBP <  149
##       0.65 when Pulse >=        84 & SysBP >= 149
##       0.70 when Pulse <  84
```

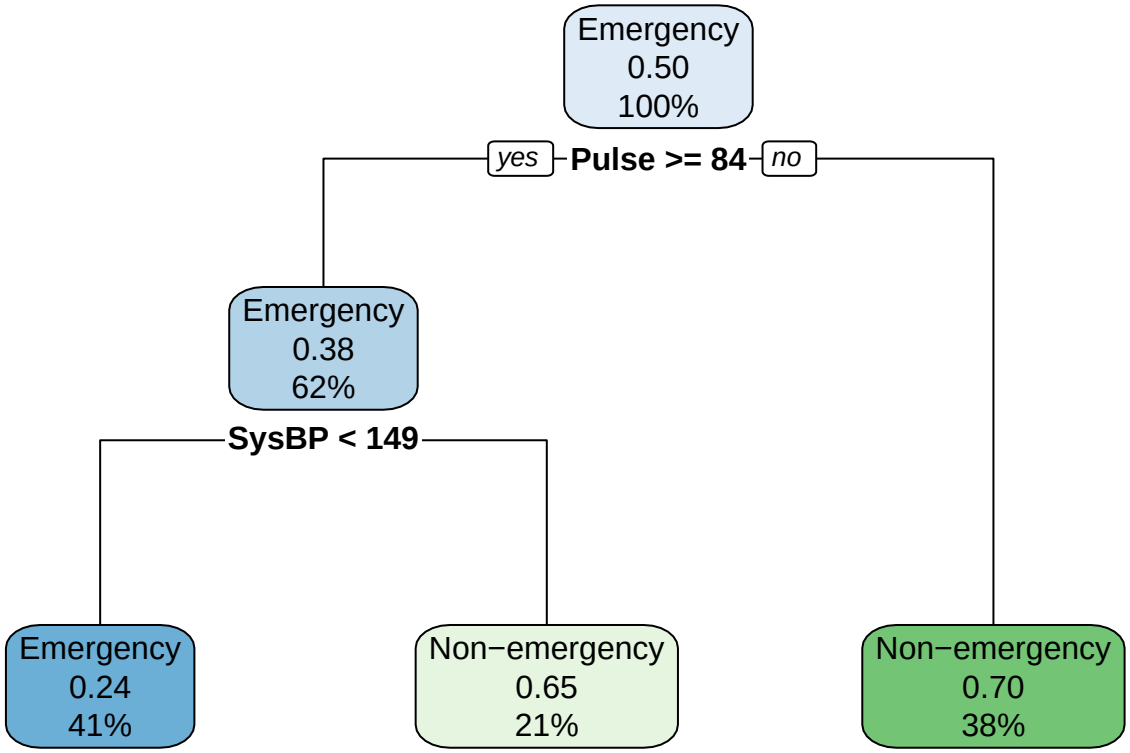
```
# Is the tree the same as before? What is your explanation for this?

# Use the predict() function to classify the data and calculate the classification
# error rate of our tree.


predicted.emergency = predict(ICU.tree)

predicted.emergency = predict(ICU.tree, type = "class")

ICU.table = table(predicted.emergency, ICU$Emergency)
ICU.table
```

```
##
## predicted.emergency Emergency Non-emergency
##       Emergency            22             4
##       Non-emergency        18            36
```

```
ICU.error = (ICU.table[2,1] + ICU.table[1,2]) / sum(ICU.table)
```

Exercise 2 - How large should my tree be?

```
# Set the Cp parameter to 0.1 and construct another tree with the ICU data. What do you notice?

ICU.tree.cp.01 = rpart(Emergency ~ SysBP + Pulse, data = ICU, control = rpart.control(cp = 0.1))
## Higher cp - smaller tree and vice versa

rpart.plot(ICU.tree.cp.01)
```



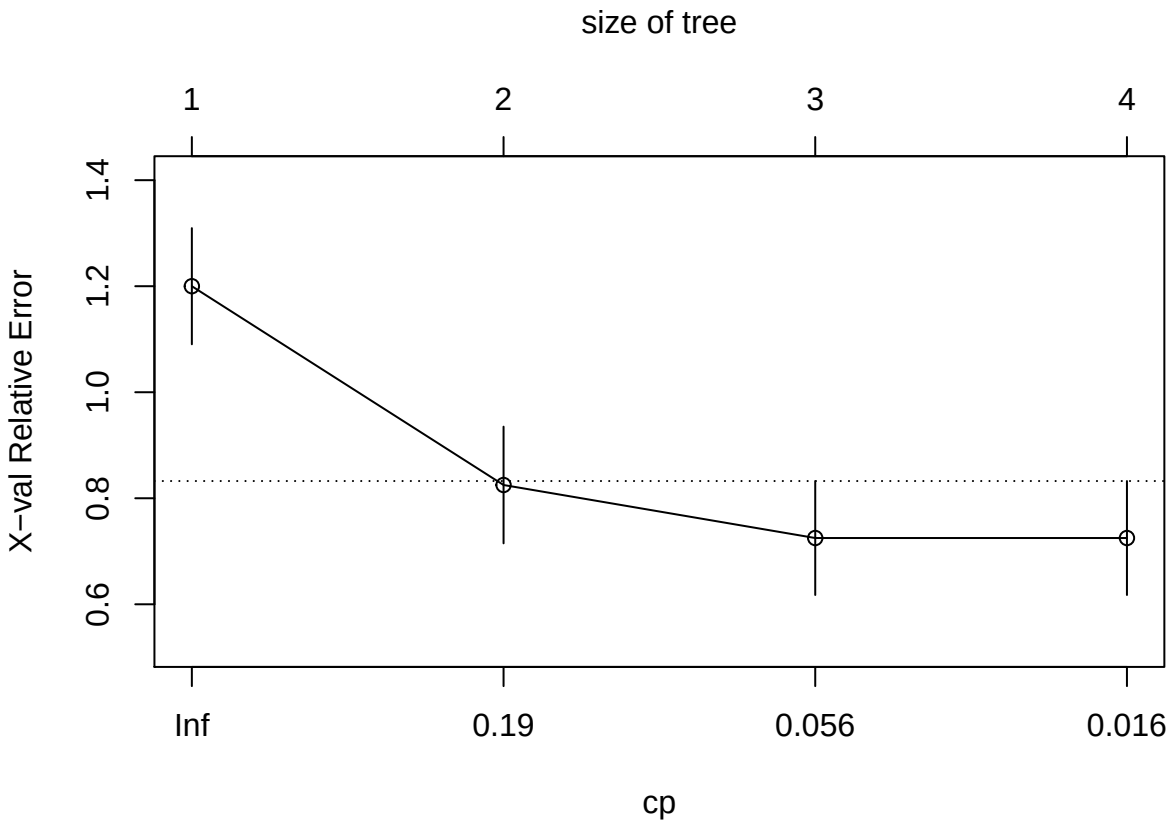

```
# Check the cp parameter of the original tree that you created (ICU.tree object) -
# use the summary() and plotcp() functions

summary(ICU.tree)$cptable
```

```
## Call:
## rpart(formula = Emergency ~ SysBP + Pulse, data = ICU)
##   n= 80
##
##      CP nsplit rel error xerror      xstd
## 1 0.300      0     1.000  1.200 0.1095445
## 2 0.125      1     0.700  0.825 0.1100781
## 3 0.025      2     0.575  0.725 0.1074927
## 4 0.010      3     0.550  0.725 0.1074927
```

```
##
## Variable importance
## Pulse SysBP
##    62    38
##
## Node number 1: 80 observations,    complexity param=0.3
##   predicted class=Emergency      expected loss=0.5  P(node) =1
##     class counts:    40    40
##    probabilities: 0.500 0.500
##   left son=2 (50 obs) right son=3 (30 obs)
##   Primary splits:
##       Pulse < 83.5  to the right, improve=3.840000, (0 missing)
##       SysBP < 149   to the left,  improve=2.849003, (0 missing)
##
## Node number 2: 50 observations,    complexity param=0.125
##   predicted class=Emergency      expected loss=0.38  P(node) =0.625
##     class counts:    31    19
##    probabilities: 0.620 0.380
##   left son=4 (33 obs) right son=5 (17 obs)
##   Primary splits:
##       SysBP < 149   to the left,  improve=3.674082, (0 missing)
##       Pulse < 119.5 to the left,  improve=1.007619, (0 missing)
##   Surrogate splits:
##       Pulse < 87.5  to the right, agree=0.68, adj=0.059, (0 split)
##
## Node number 3: 30 observations
##   predicted class=Non-emergency  expected loss=0.3  P(node) =0.375
##     class counts:     9    21
##    probabilities: 0.300 0.700
##
## Node number 4: 33 observations,    complexity param=0.025
##   predicted class=Emergency      expected loss=0.2424242  P(node) =0.4125
##     class counts:    25     8
##    probabilities: 0.758 0.242
##   left son=8 (26 obs) right son=9 (7 obs)
##   Primary splits:
##       Pulse < 126.5 to the left,  improve=1.9234100, (0 missing)
##       SysBP < 105   to the left,  improve=0.4267677, (0 missing)
##
## Node number 5: 17 observations
##   predicted class=Non-emergency  expected loss=0.3529412  P(node) =0.2125
##     class counts:     6    11
##    probabilities: 0.353 0.647
##
## Node number 8: 26 observations
##   predicted class=Emergency      expected loss=0.1538462  P(node) =0.325
##     class counts:    22     4
##    probabilities: 0.846 0.154
##
## Node number 9: 7 observations
##   predicted class=Non-emergency  expected loss=0.4285714  P(node) =0.0875
##     class counts:     3     4
##    probabilities: 0.429 0.571
```

```
##      CP nsplit rel error xerror      xstd
## 1 0.300      0     1.000  1.200 0.1095445
## 2 0.125      1     0.700  0.825 0.1100781
## 3 0.025      2     0.575  0.725 0.1074927
## 4 0.010      3     0.550  0.725 0.1074927
```

```
plotcp(ICU.tree)
```




```
# What is the prediction accuracy of the original tree (ICU.tree object) but on
# new data?

ICU.test = read.delim("ICU.adapted.test.txt")
predicted.test.emergency = predict(ICU.tree, newdata = ICU.test, type = "class")

ICU.test.table = table(predicted.test.emergency, ICU.test$Emergency)

ICU.test.error = (ICU.test.table[1,2] + ICU.test.table[2,1]) / sum(ICU.test.table)
ICU.test.error
```

```
## [1] 0.7
```

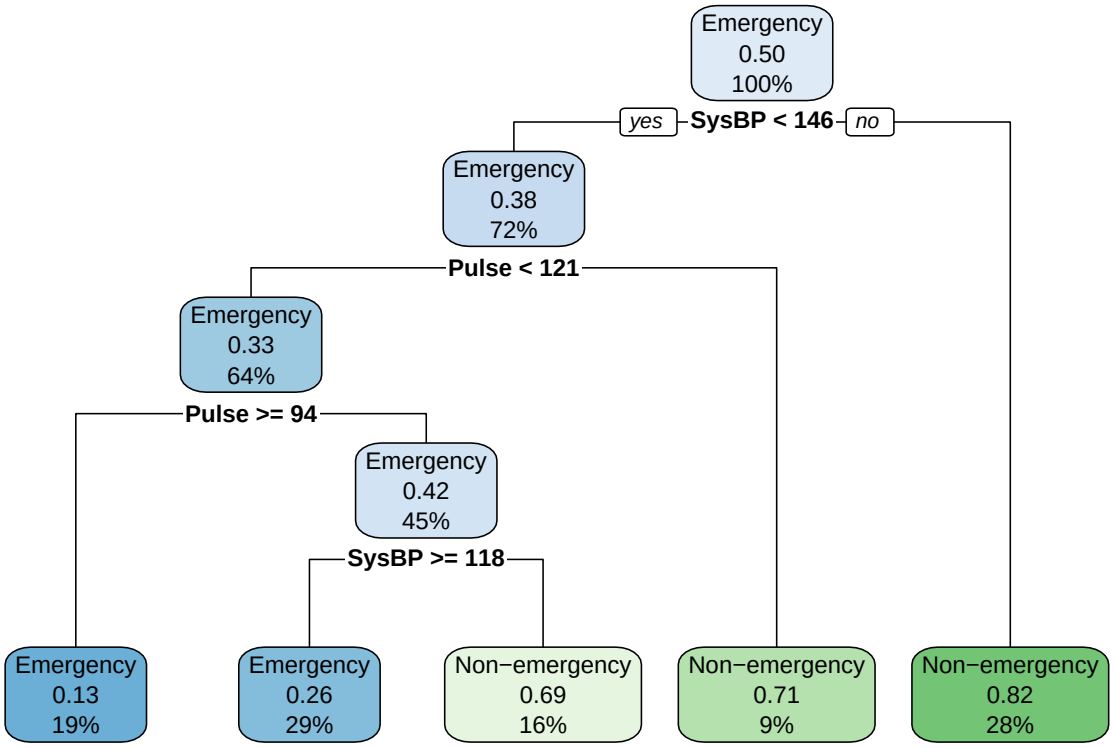
This exercise shows the influence of the cp parameter on the tree size and the prediction accuracy

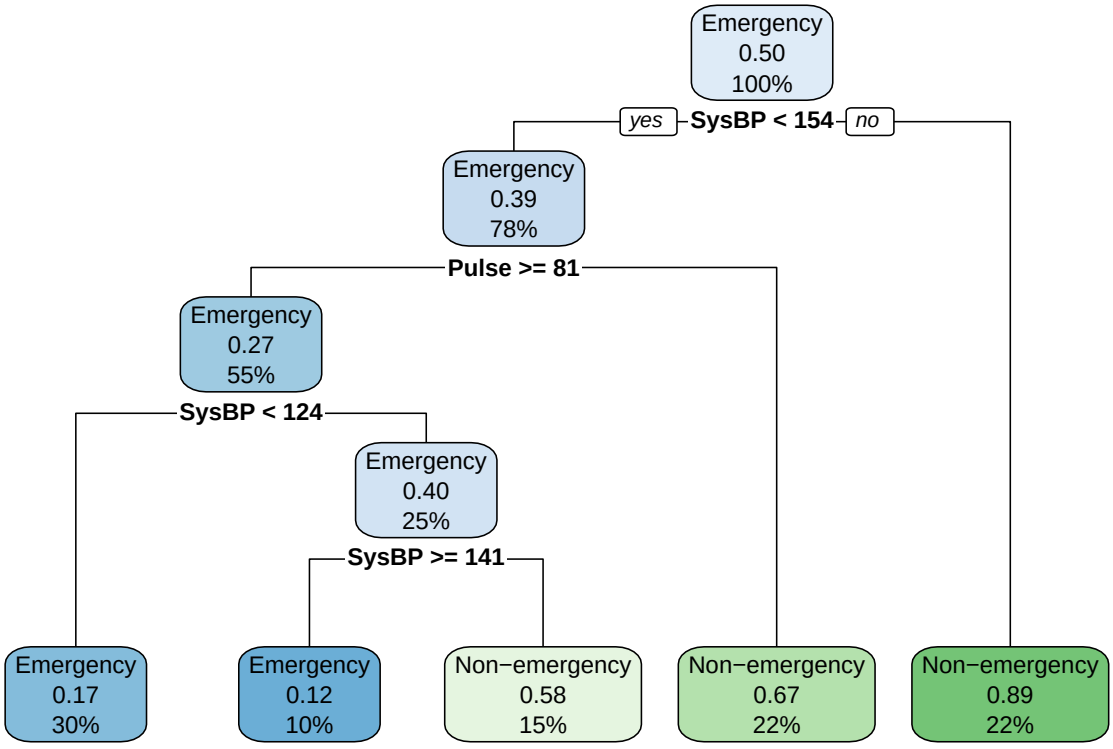
Exercise 3 - Stability of trees

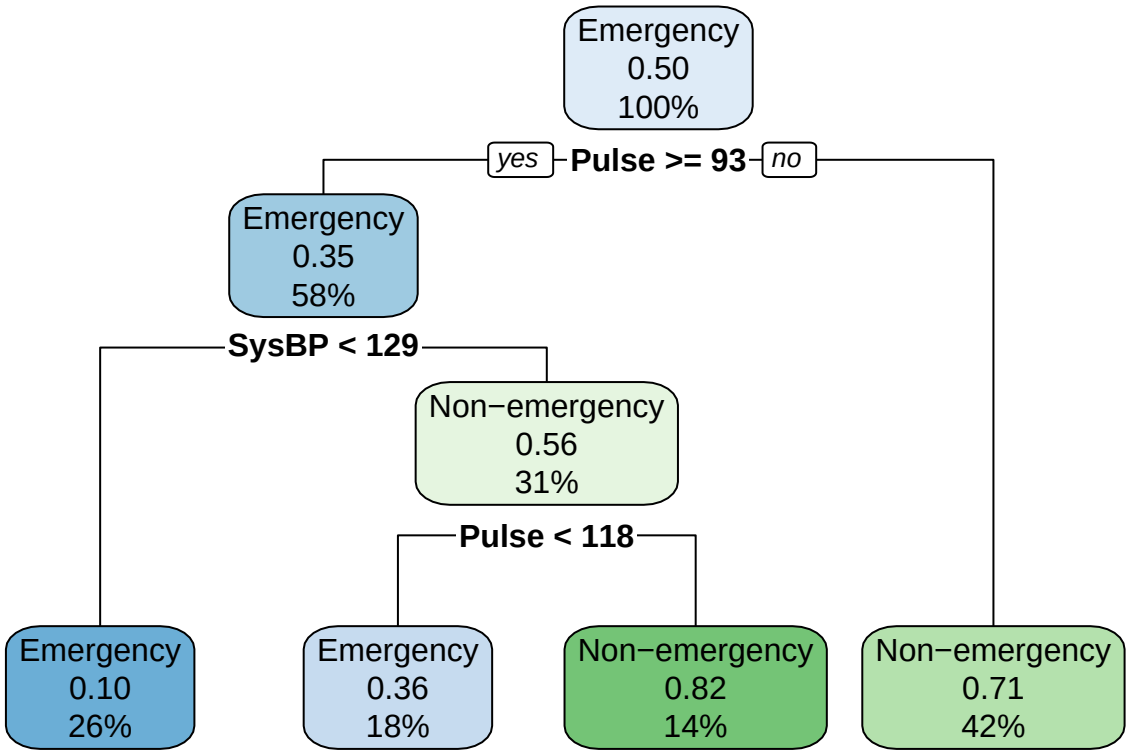
Read in the original ICU dataset from the Stat2Data package. You want to construct a training set with 50 Emergency and 50 Non-emergency patients and train a classification tree. Repeat this process 3 times and

compare the resulting classification trees. Set the seed to 1, 2 and 3 respectively. What do you notice?

```
# Read in the original ICU data from the Stat2Data package
ICU.original = read.delim("ICU.original.txt")

# Repeat the process three times
for (i in 1:3){
  set.seed(i)
  # Sample 40 observations from each class
 emergency = sample(which(ICU.original$Emergency == "Emergency"), 40)
  notemergency = sample(which(ICU.original$Emergency == "Non-emergency"), 40)
  ICU.sampled = rbind(ICU.original[emergency,], ICU.original[notemergency,])

  # Build a classification tree with rpart
 ICU.tree.sampled = rpart(Emergency ~ SysBP + Pulse, data = ICU.sampled,
                          method = "class")
 # Visualize the tree
 rpart.plot(ICU.tree.sampled)
}
```


Emergency
0.50
100%
yes
SysBP < 146
no
Emergency
0.38
72%
Pulse < 121
Emergency
0.33
64%
Pulse >= 94
Emergency
0.42
45%
SysBP >= 118
Emergency
0.13
19%
Emergency
0.26
29%
Non−emergency
0.69
16%
Non−emergency
0.71
9%
Non−emergency
0.82
28%


Emergency
0.50
100%
yes
SysBP < 154
no
Emergency
0.39
78%
Pulse >= 81
Emergency
0.27
55%
SysBP < 124
Emergency
0.40
25%
SysBP >= 141
Emergency
0.17
30%
Emergency
0.12
10%
Non−emergency
0.58
15%
Non−emergency
0.67
22%
Non−emergency
0.89
22%



You are supposed to notice the instability of trees when change input data (take different samples of patients from the same dataset)

Regression trees

Exercise 4 - Training a regression tree using the Caret package

Using the rpart method implemented in caret package, train a regression tree to predict gene expression from histone modifications. Report the accuracy of your model (hcp.model)

```
# Read in data for histone modifications and gene expression
hm.data.hcp = read.delim("hm.data.hcp.txt")


# Create training and test sets using the CreateDataPartition() function
# (80:20 split)
set.seed(12)
index.train <- createDataPartition(y = hm.data.hcp$expression,
                                   p = 0.8,
                                   list = FALSE)
training.hcp <- hm.data.hcp[index.train,]
testing.hcp <- hm.data.hcp[-index.train,]

# What are the dimensions of the training and test datasets?
dim(training.hcp)
```

```
## [1] 400  32
```

```
dim(testing.hcp)
```

```
## [1] 100  32
```

```
# Define the control parameters for the model using the trainControl() function.
# Use 5-fold cross-validation and return all resampled measures.
# Also print the training log.
# Save the control parameters in an object hcp.ctrl since we will use them in
# subsequent models.
hcp.ctrl <- trainControl(method = "cv",
                         number = 5,
                         returnResamp = "all",
                         verboseIter = TRUE)

# Train the regression tree using the train() function
hcp.model <- train(  expression ~ .,
                     training.hcp,
                   method = "rpart",
                   trControl = hcp.ctrl)
```

```
## + Fold1: cp=0.03009
## - Fold1: cp=0.03009
## + Fold2: cp=0.03009
## - Fold2: cp=0.03009
## + Fold3: cp=0.03009
## - Fold3: cp=0.03009
## + Fold4: cp=0.03009
## - Fold4: cp=0.03009
## + Fold5: cp=0.03009
## - Fold5: cp=0.03009
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```
## Aggregating results
## Selecting tuning parameters
## Fitting cp = 0.0301 on full training set
```

```
# Test the model on the test data
# Predict expression values for test data
predicted.hcp = predict(hcp.model, newdata = testing.hcp)

# Calculate the accuracy of the model using the postResample() function
test.hcp.accuracy = postResample(pred = predicted.hcp, obs = testing.hcp$expression)

test.hcp.accuracy
```

```
##      RMSE  Rsquared       MAE
## 1.6603689 0.2530859 1.1663846
```

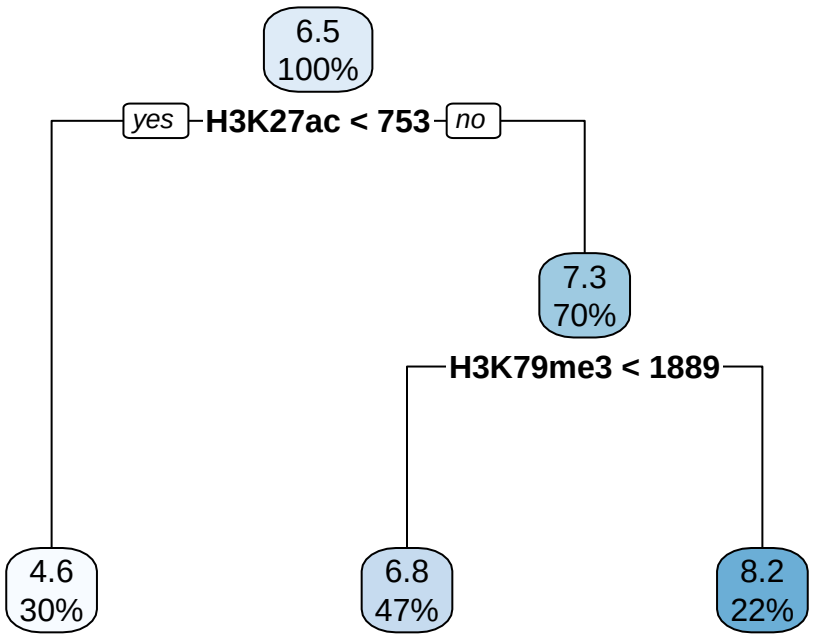
Exercise 5 - Visualize the regression tree and the predictions

Plot the binary tree representing the model and visualize the observed and predicted expression data. What do you notice?

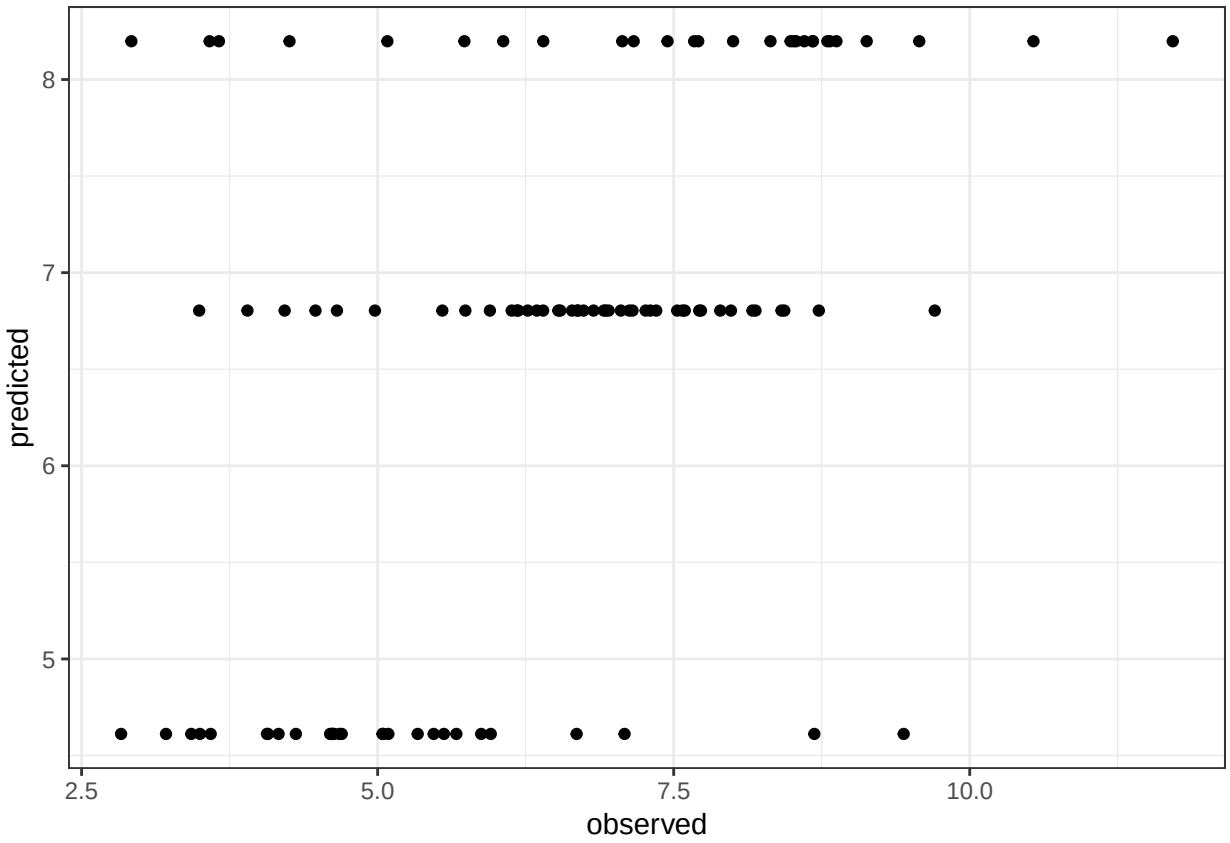
```
# What does our model actually look like? Print the final model and
# visualize the regression tree using rpart.plot()
hcp.model$finalModel
```

```
## n= 400
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 400 1488.4150 6.454349
##   2) H3K27ac< 753 121  228.6196 4.611699 *
##   3) H3K27ac>=753 279  670.7795 7.253491
##     6) H3K79me3< 1889 189  389.1397 6.803634 *
##     7) H3K79me3>=1889 90  163.0704 8.198191 *
```

```
rpart.plot(hcp.model$finalModel)
```




```
# Create a scatterplot of predicted and observed values of gene expression
# on the test set. What do you notice?
hcp.results = data.frame(observed = testing.hcp$expression, predicted = predicted.hcp)

ggplot(hcp.results, aes(x = observed, y = predicted)) +
  geom_point() +
  theme_bw()
```



This is a demonstration of the lack of smoothness of regression trees

Ensemble methods

Exercise 6 - Random Forest

Using the ranger method implemented in caret package, train a regression tree to predict gene expression from histone modifications. Report the accuracy of your model (hcp.model.rf)

```
# Train the model using 'ranger' method. Set the number of trees to 500
# and importance to "permutation"
hcp.model.rf <- train(expression ~ .,
                      data = training.hcp,
                      method = 'ranger',
                     # should be set high at least p/3
                      tuneLength = 10,
                      trControl = hcp.ctrl,
                     ## parameters passed onto the ranger function
                     # the bigger the better.
                      num.trees = 500,
                      importance = "permutation")
```

```
## + Fold1: mtry= 2, min.node.size=5, splitrule=variance
## - Fold1: mtry= 2, min.node.size=5, splitrule=variance
## + Fold1: mtry= 5, min.node.size=5, splitrule=variance
## - Fold1: mtry= 5, min.node.size=5, splitrule=variance
```

```
## + Fold1: mtry= 8, min.node.size=5, splitrule=variance
## - Fold1: mtry= 8, min.node.size=5, splitrule=variance
## + Fold1: mtry=11, min.node.size=5, splitrule=variance
## - Fold1: mtry=11, min.node.size=5, splitrule=variance
## + Fold1: mtry=14, min.node.size=5, splitrule=variance
## - Fold1: mtry=14, min.node.size=5, splitrule=variance
## + Fold1: mtry=18, min.node.size=5, splitrule=variance
## - Fold1: mtry=18, min.node.size=5, splitrule=variance
## + Fold1: mtry=21, min.node.size=5, splitrule=variance
## - Fold1: mtry=21, min.node.size=5, splitrule=variance
## + Fold1: mtry=24, min.node.size=5, splitrule=variance
## - Fold1: mtry=24, min.node.size=5, splitrule=variance
## + Fold1: mtry=27, min.node.size=5, splitrule=variance
## - Fold1: mtry=27, min.node.size=5, splitrule=variance
## + Fold1: mtry=31, min.node.size=5, splitrule=variance
## - Fold1: mtry=31, min.node.size=5, splitrule=variance
## + Fold1: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold1: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold1: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 2, min.node.size=5, splitrule=variance
## - Fold2: mtry= 2, min.node.size=5, splitrule=variance
## + Fold2: mtry= 5, min.node.size=5, splitrule=variance
## - Fold2: mtry= 5, min.node.size=5, splitrule=variance
## + Fold2: mtry= 8, min.node.size=5, splitrule=variance
## - Fold2: mtry= 8, min.node.size=5, splitrule=variance
## + Fold2: mtry=11, min.node.size=5, splitrule=variance
## - Fold2: mtry=11, min.node.size=5, splitrule=variance
## + Fold2: mtry=14, min.node.size=5, splitrule=variance
## - Fold2: mtry=14, min.node.size=5, splitrule=variance
## + Fold2: mtry=18, min.node.size=5, splitrule=variance
## - Fold2: mtry=18, min.node.size=5, splitrule=variance
## + Fold2: mtry=21, min.node.size=5, splitrule=variance
## - Fold2: mtry=21, min.node.size=5, splitrule=variance
## + Fold2: mtry=24, min.node.size=5, splitrule=variance
## - Fold2: mtry=24, min.node.size=5, splitrule=variance
## + Fold2: mtry=27, min.node.size=5, splitrule=variance
## - Fold2: mtry=27, min.node.size=5, splitrule=variance
```

```
## + Fold2: mtry=31, min.node.size=5, splitrule=variance
## - Fold2: mtry=31, min.node.size=5, splitrule=variance
## + Fold2: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 2, min.node.size=5, splitrule=variance
## - Fold3: mtry= 2, min.node.size=5, splitrule=variance
## + Fold3: mtry= 5, min.node.size=5, splitrule=variance
## - Fold3: mtry= 5, min.node.size=5, splitrule=variance
## + Fold3: mtry= 8, min.node.size=5, splitrule=variance
## - Fold3: mtry= 8, min.node.size=5, splitrule=variance
## + Fold3: mtry=11, min.node.size=5, splitrule=variance
## - Fold3: mtry=11, min.node.size=5, splitrule=variance
## + Fold3: mtry=14, min.node.size=5, splitrule=variance
## - Fold3: mtry=14, min.node.size=5, splitrule=variance
## + Fold3: mtry=18, min.node.size=5, splitrule=variance
## - Fold3: mtry=18, min.node.size=5, splitrule=variance
## + Fold3: mtry=21, min.node.size=5, splitrule=variance
## - Fold3: mtry=21, min.node.size=5, splitrule=variance
## + Fold3: mtry=24, min.node.size=5, splitrule=variance
## - Fold3: mtry=24, min.node.size=5, splitrule=variance
## + Fold3: mtry=27, min.node.size=5, splitrule=variance
## - Fold3: mtry=27, min.node.size=5, splitrule=variance
## + Fold3: mtry=31, min.node.size=5, splitrule=variance
## - Fold3: mtry=31, min.node.size=5, splitrule=variance
## + Fold3: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=18, min.node.size=5, splitrule=extratrees
```

```
## + Fold3: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 2, min.node.size=5, splitrule=variance
## - Fold4: mtry= 2, min.node.size=5, splitrule=variance
## + Fold4: mtry= 5, min.node.size=5, splitrule=variance
## - Fold4: mtry= 5, min.node.size=5, splitrule=variance
## + Fold4: mtry= 8, min.node.size=5, splitrule=variance
## - Fold4: mtry= 8, min.node.size=5, splitrule=variance
## + Fold4: mtry=11, min.node.size=5, splitrule=variance
## - Fold4: mtry=11, min.node.size=5, splitrule=variance
## + Fold4: mtry=14, min.node.size=5, splitrule=variance
## - Fold4: mtry=14, min.node.size=5, splitrule=variance
## + Fold4: mtry=18, min.node.size=5, splitrule=variance
## - Fold4: mtry=18, min.node.size=5, splitrule=variance
## + Fold4: mtry=21, min.node.size=5, splitrule=variance
## - Fold4: mtry=21, min.node.size=5, splitrule=variance
## + Fold4: mtry=24, min.node.size=5, splitrule=variance
## - Fold4: mtry=24, min.node.size=5, splitrule=variance
## + Fold4: mtry=27, min.node.size=5, splitrule=variance
## - Fold4: mtry=27, min.node.size=5, splitrule=variance
## + Fold4: mtry=31, min.node.size=5, splitrule=variance
## - Fold4: mtry=31, min.node.size=5, splitrule=variance
## + Fold4: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 2, min.node.size=5, splitrule=variance
## - Fold5: mtry= 2, min.node.size=5, splitrule=variance
## + Fold5: mtry= 5, min.node.size=5, splitrule=variance
## - Fold5: mtry= 5, min.node.size=5, splitrule=variance
## + Fold5: mtry= 8, min.node.size=5, splitrule=variance
## - Fold5: mtry= 8, min.node.size=5, splitrule=variance
```

```
## + Fold5: mtry=11, min.node.size=5, splitrule=variance
## - Fold5: mtry=11, min.node.size=5, splitrule=variance
## + Fold5: mtry=14, min.node.size=5, splitrule=variance
## - Fold5: mtry=14, min.node.size=5, splitrule=variance
## + Fold5: mtry=18, min.node.size=5, splitrule=variance
## - Fold5: mtry=18, min.node.size=5, splitrule=variance
## + Fold5: mtry=21, min.node.size=5, splitrule=variance
## - Fold5: mtry=21, min.node.size=5, splitrule=variance
## + Fold5: mtry=24, min.node.size=5, splitrule=variance
## - Fold5: mtry=24, min.node.size=5, splitrule=variance
## + Fold5: mtry=27, min.node.size=5, splitrule=variance
## - Fold5: mtry=27, min.node.size=5, splitrule=variance
## + Fold5: mtry=31, min.node.size=5, splitrule=variance
## - Fold5: mtry=31, min.node.size=5, splitrule=variance
## + Fold5: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=31, min.node.size=5, splitrule=extratrees
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 5, splitrule = variance, min.node.size = 5 on full training set
```

```
# Inspect the final randomForest model
hcp.model.rf$finalModel
```

```
## Ranger result
##
## Call:
##  ranger::ranger(dependent.variable.name = ".outcome", data = x,      mtry = min(param$mtry, ncol(x))
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      400
## Number of independent variables:  31
## Mtry:                             5
## Target node size:                 5
## Variable importance mode:         permutation
## Splitrule:                        variance
## OOB prediction error (MSE):       1.761024
```

```
## R squared (OOB):                   0.5279217
```

```
# Using the predict() function, predict the expression values of the test dataset
predicted.hcp.rf <- predict(hcp.model.rf , newdata = testing.hcp)

# Calculate the accuracy of the model using the postResample() function
test.hcp.accuracy.rf = postResample(pred = predicted.hcp.rf, obs = testing.hcp$expression)
test.hcp.accuracy.rf
```

```
##      RMSE  Rsquared       MAE
## 1.5527371 0.3295244 1.0956180
```

Notice that random forest gives a higher prediction accuracy than individual regression tree from exercise 4

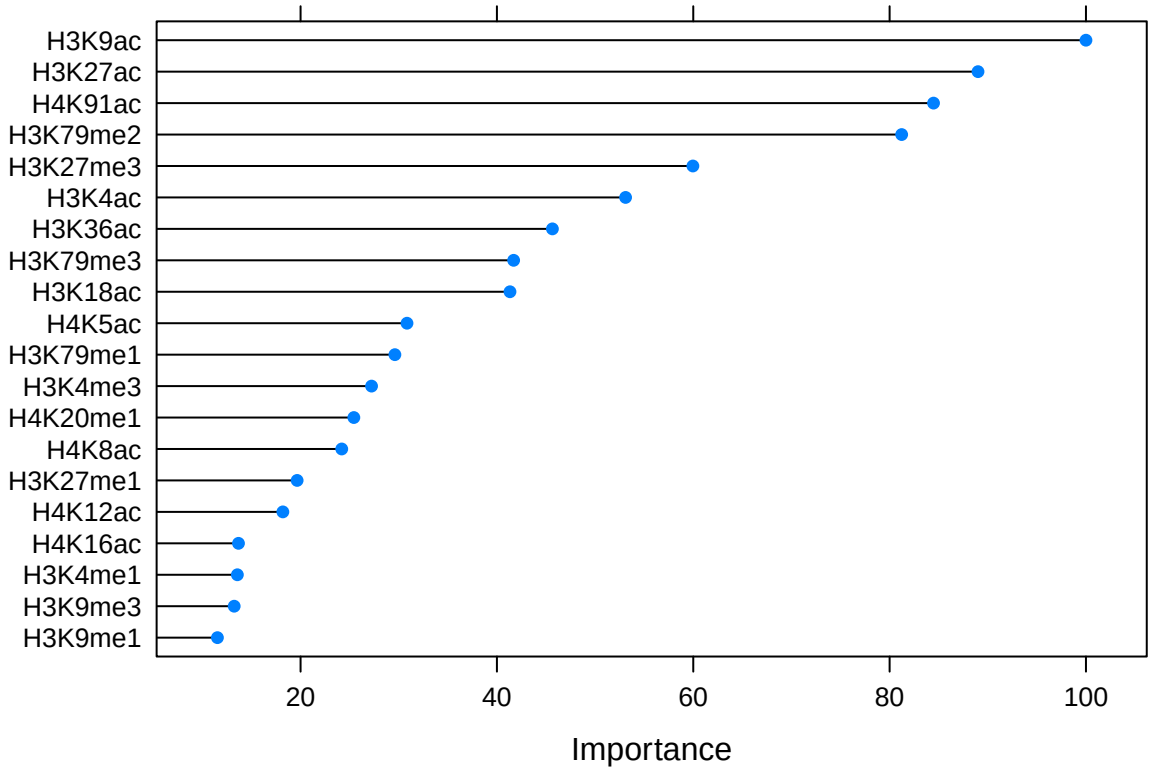
Exercise 7 - Variable importance of Random Forest models

Compare variable importance of random forest models used for predicting expression of genes with HCP or LCP promoters.

```
# Inspect and visualize the variable importance of randomForest model used to
# predict gene expression from histone modifications. Use the varImp() function
rfImp.hcp <- varImp(hcp.model.rf)
rfImp.hcp
```

```
## ranger variable importance
##
##   only 20 most important variables shown (out of 31)
##
##          Overall
## H3K9ac    100.00
## H3K27ac    89.00
## H4K91ac    84.47
## H3K79me2   81.23
## H3K27me3   59.96
## H3K4ac     53.11
## H3K36ac    45.65
## H3K79me3   41.70
## H3K18ac    41.33
## H4K5ac     30.84
## H3K79me1   29.61
## H3K4me3    27.23
## H4K20me1   25.43
## H4K8ac     24.19
## H3K27me1   19.65
## H4K12ac    18.20
## H4K16ac    13.68
## H3K4me1    13.56
## H3K9me3    13.24
## H3K9me1    11.53
```

```
plot(rfImp.hcp,top = 20)
```




```
# Using the ranger method implemented in caret package, train a regression tree
# to predict gene expression from histone modifications but for LCP promoters.
# Report the accuracy of your model (lcp.model.rf)

# Read in expression and histone modification data for genes with LCP promoters
hm.data.lcp = read.delim("hm.data.lcp.txt")

# Create training and test sets using the createDataPartition() function (80:20)
index.train <- createDataPartition(y = hm.data.lcp$expression, p= 0.8, list = FALSE)
training.lcp <- hm.data.lcp[index.train,]
testing.lcp <- hm.data.lcp[-index.train,]

# Train the model using 'ranger' method. Set the number of trees to 500 and
# importance to "permutation"
lcp.model.rf <- train(expression ~ .,
                      data = training.lcp,
                      method = 'ranger',
                     # should be set high at least p/3
                      tuneLength = 10,
                      trControl = hcp.ctrl,
                     ## parameters passed onto the ranger function
                     # the bigger the better.
                      num.trees = 500,
                      importance = "permutation")
```

```
## + Fold1: mtry= 2, min.node.size=5, splitrule=variance
```

```
## - Fold1: mtry= 2, min.node.size=5, splitrule=variance
## + Fold1: mtry= 5, min.node.size=5, splitrule=variance
## - Fold1: mtry= 5, min.node.size=5, splitrule=variance
## + Fold1: mtry= 8, min.node.size=5, splitrule=variance
## - Fold1: mtry= 8, min.node.size=5, splitrule=variance
## + Fold1: mtry=11, min.node.size=5, splitrule=variance
## - Fold1: mtry=11, min.node.size=5, splitrule=variance
## + Fold1: mtry=14, min.node.size=5, splitrule=variance
## - Fold1: mtry=14, min.node.size=5, splitrule=variance
## + Fold1: mtry=18, min.node.size=5, splitrule=variance
## - Fold1: mtry=18, min.node.size=5, splitrule=variance
## + Fold1: mtry=21, min.node.size=5, splitrule=variance
## - Fold1: mtry=21, min.node.size=5, splitrule=variance
## + Fold1: mtry=24, min.node.size=5, splitrule=variance
## - Fold1: mtry=24, min.node.size=5, splitrule=variance
## + Fold1: mtry=27, min.node.size=5, splitrule=variance
## - Fold1: mtry=27, min.node.size=5, splitrule=variance
## + Fold1: mtry=31, min.node.size=5, splitrule=variance
## - Fold1: mtry=31, min.node.size=5, splitrule=variance
## + Fold1: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold1: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold1: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold1: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold1: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold1: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 2, min.node.size=5, splitrule=variance
## - Fold2: mtry= 2, min.node.size=5, splitrule=variance
## + Fold2: mtry= 5, min.node.size=5, splitrule=variance
## - Fold2: mtry= 5, min.node.size=5, splitrule=variance
## + Fold2: mtry= 8, min.node.size=5, splitrule=variance
## - Fold2: mtry= 8, min.node.size=5, splitrule=variance
## + Fold2: mtry=11, min.node.size=5, splitrule=variance
## - Fold2: mtry=11, min.node.size=5, splitrule=variance
## + Fold2: mtry=14, min.node.size=5, splitrule=variance
## - Fold2: mtry=14, min.node.size=5, splitrule=variance
## + Fold2: mtry=18, min.node.size=5, splitrule=variance
## - Fold2: mtry=18, min.node.size=5, splitrule=variance
## + Fold2: mtry=21, min.node.size=5, splitrule=variance
## - Fold2: mtry=21, min.node.size=5, splitrule=variance
## + Fold2: mtry=24, min.node.size=5, splitrule=variance
```

```
## - Fold2: mtry=24, min.node.size=5, splitrule=variance
## + Fold2: mtry=27, min.node.size=5, splitrule=variance
## - Fold2: mtry=27, min.node.size=5, splitrule=variance
## + Fold2: mtry=31, min.node.size=5, splitrule=variance
## - Fold2: mtry=31, min.node.size=5, splitrule=variance
## + Fold2: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold2: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold2: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold2: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold2: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 2, min.node.size=5, splitrule=variance
## - Fold3: mtry= 2, min.node.size=5, splitrule=variance
## + Fold3: mtry= 5, min.node.size=5, splitrule=variance
## - Fold3: mtry= 5, min.node.size=5, splitrule=variance
## + Fold3: mtry= 8, min.node.size=5, splitrule=variance
## - Fold3: mtry= 8, min.node.size=5, splitrule=variance
## + Fold3: mtry=11, min.node.size=5, splitrule=variance
## - Fold3: mtry=11, min.node.size=5, splitrule=variance
## + Fold3: mtry=14, min.node.size=5, splitrule=variance
## - Fold3: mtry=14, min.node.size=5, splitrule=variance
## + Fold3: mtry=18, min.node.size=5, splitrule=variance
## - Fold3: mtry=18, min.node.size=5, splitrule=variance
## + Fold3: mtry=21, min.node.size=5, splitrule=variance
## - Fold3: mtry=21, min.node.size=5, splitrule=variance
## + Fold3: mtry=24, min.node.size=5, splitrule=variance
## - Fold3: mtry=24, min.node.size=5, splitrule=variance
## + Fold3: mtry=27, min.node.size=5, splitrule=variance
## - Fold3: mtry=27, min.node.size=5, splitrule=variance
## + Fold3: mtry=31, min.node.size=5, splitrule=variance
## - Fold3: mtry=31, min.node.size=5, splitrule=variance
## + Fold3: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold3: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold3: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=14, min.node.size=5, splitrule=extratrees
```

```
## - Fold3: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold3: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold3: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 2, min.node.size=5, splitrule=variance
## - Fold4: mtry= 2, min.node.size=5, splitrule=variance
## + Fold4: mtry= 5, min.node.size=5, splitrule=variance
## - Fold4: mtry= 5, min.node.size=5, splitrule=variance
## + Fold4: mtry= 8, min.node.size=5, splitrule=variance
## - Fold4: mtry= 8, min.node.size=5, splitrule=variance
## + Fold4: mtry=11, min.node.size=5, splitrule=variance
## - Fold4: mtry=11, min.node.size=5, splitrule=variance
## + Fold4: mtry=14, min.node.size=5, splitrule=variance
## - Fold4: mtry=14, min.node.size=5, splitrule=variance
## + Fold4: mtry=18, min.node.size=5, splitrule=variance
## - Fold4: mtry=18, min.node.size=5, splitrule=variance
## + Fold4: mtry=21, min.node.size=5, splitrule=variance
## - Fold4: mtry=21, min.node.size=5, splitrule=variance
## + Fold4: mtry=24, min.node.size=5, splitrule=variance
## - Fold4: mtry=24, min.node.size=5, splitrule=variance
## + Fold4: mtry=27, min.node.size=5, splitrule=variance
## - Fold4: mtry=27, min.node.size=5, splitrule=variance
## + Fold4: mtry=31, min.node.size=5, splitrule=variance
## - Fold4: mtry=31, min.node.size=5, splitrule=variance
## + Fold4: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold4: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold4: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold4: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold4: mtry=31, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 2, min.node.size=5, splitrule=variance
## - Fold5: mtry= 2, min.node.size=5, splitrule=variance
## + Fold5: mtry= 5, min.node.size=5, splitrule=variance
```

```
## - Fold5: mtry= 5, min.node.size=5, splitrule=variance
## + Fold5: mtry= 8, min.node.size=5, splitrule=variance
## - Fold5: mtry= 8, min.node.size=5, splitrule=variance
## + Fold5: mtry=11, min.node.size=5, splitrule=variance
## - Fold5: mtry=11, min.node.size=5, splitrule=variance
## + Fold5: mtry=14, min.node.size=5, splitrule=variance
## - Fold5: mtry=14, min.node.size=5, splitrule=variance
## + Fold5: mtry=18, min.node.size=5, splitrule=variance
## - Fold5: mtry=18, min.node.size=5, splitrule=variance
## + Fold5: mtry=21, min.node.size=5, splitrule=variance
## - Fold5: mtry=21, min.node.size=5, splitrule=variance
## + Fold5: mtry=24, min.node.size=5, splitrule=variance
## - Fold5: mtry=24, min.node.size=5, splitrule=variance
## + Fold5: mtry=27, min.node.size=5, splitrule=variance
## - Fold5: mtry=27, min.node.size=5, splitrule=variance
## + Fold5: mtry=31, min.node.size=5, splitrule=variance
## - Fold5: mtry=31, min.node.size=5, splitrule=variance
## + Fold5: mtry= 2, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 2, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 5, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 5, min.node.size=5, splitrule=extratrees
## + Fold5: mtry= 8, min.node.size=5, splitrule=extratrees
## - Fold5: mtry= 8, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=11, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=11, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=14, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=14, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=18, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=18, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=21, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=21, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=24, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=24, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=27, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=27, min.node.size=5, splitrule=extratrees
## + Fold5: mtry=31, min.node.size=5, splitrule=extratrees
## - Fold5: mtry=31, min.node.size=5, splitrule=extratrees
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 11, splitrule = variance, min.node.size = 5 on full training set
```

```
#Inspect the final model
lcp.model.rf$finalModel
```

```
## Ranger result
##
## Call:
##  ranger::ranger(dependent.variable.name = ".outcome", data = x,      mtry = min(param$mtry, ncol(x))
##
## Type:                             Regression
## Number of trees:                  500
## Sample size:                      400
## Number of independent variables:  31
## Mtry:                             11
## Target node size:                 5
```

```
## Variable importance mode:         permutation
## Splitrule:                        variance
## OOB prediction error (MSE):       1.313338
## R squared (OOB):                  0.5535528
```

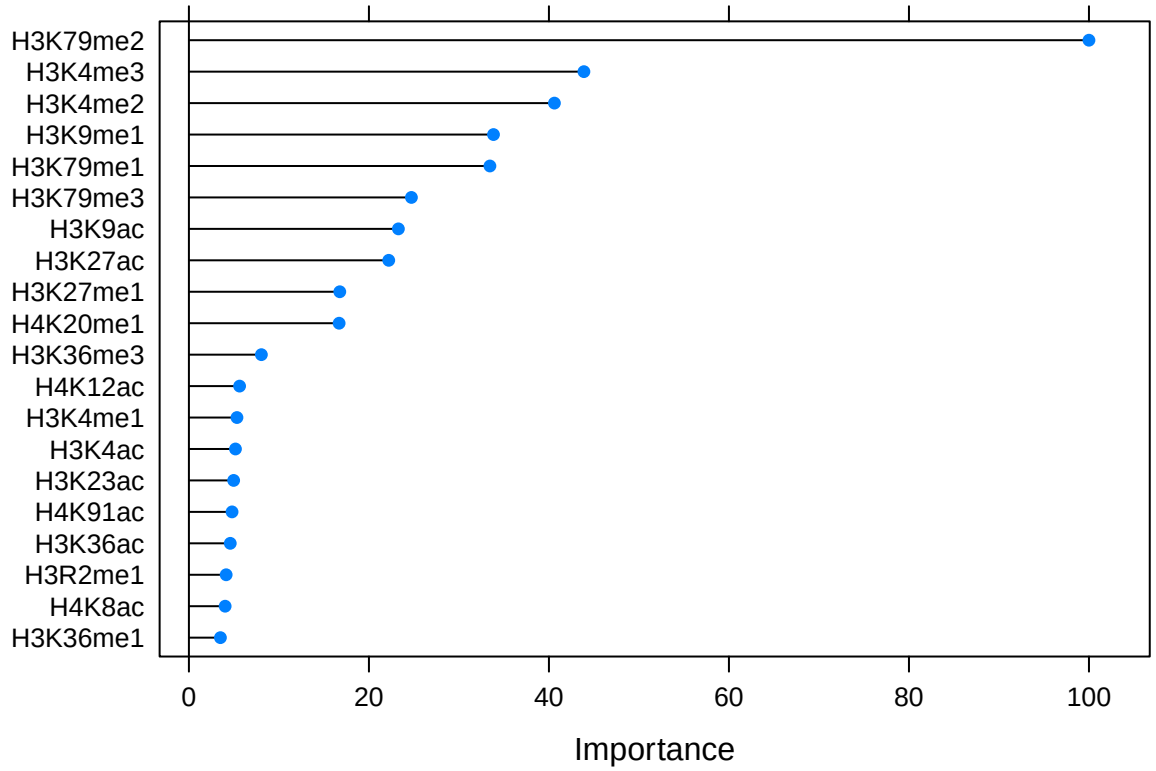
```
# Using the predict() function, predict the expression values of the test dataset
# of genes with LCP promoters
predicted.lcp.rf <- predict(lcp.model.rf , testing.lcp)

# Calculate the accuracy of the model using the postResample() function
test.accuracy.lcp.rf = postResample(pred = predicted.lcp.rf, obs = testing.lcp$expression)
test.accuracy.lcp.rf
```

```
##     RMSE Rsquared      MAE
## 1.208272 0.494549 0.992078
```

```
# Inspect and visualize the variable importance of randomForest model used to
# predict gene expression from histone modifications for genes with LCP promoters.
# Use the varImp() function
rfImp.lcp <- varImp(lcp.model.rf)
rfImp.lcp
```

```
## ranger variable importance
##
##   only 20 most important variables shown (out of 31)
##
##          Overall
## H3K79me2 100.000
## H3K4me3   43.888
## H3K4me2   40.611
## H3K9me1   33.841
## H3K79me1  33.444
## H3K79me3  24.720
## H3K9ac    23.273
## H3K27ac   22.202
## H3K27me1  16.763
## H4K20me1  16.688
## H3K36me3   8.051
## H4K12ac    5.632
## H3K4me1    5.336
## H3K4ac     5.172
## H3K23ac    4.970
## H4K91ac    4.793
## H3K36ac    4.594
## H3R2me1    4.138
## H4K8ac     4.024
## H3K36me1   3.500
```

```
plot(rfImp.lcp,top = 20)
```



The purpose of this exercise was to demonstrate how we can use variable importance to infer biological mechanisms - genes with high-CpG (HCP) and low-CpG (LCP) promoters seem to be regulated by different histone modfications. For more information you can check out https://www.pnas.org/content/107/7/2926 where we used linear regression with feature selection to choose important variables but the final results are similar.

Exercise 8 - Bagging

Using the "treebag" method implemented in caret package, train a regression tree to predict gene expression from histone modifications for genes with HCP promoters. Report the accuracy of your model (hcp.model.bag)

```
# Train the model using 'treebag' method. Set nbagg to 10
hcp.model.bag <- train(expression~.,
                       data = training.hcp,
                       method = "treebag",
                       trControl = hcp.ctrl,
                       nbagg = 10)
```

```
## + Fold1: parameter=none
## - Fold1: parameter=none
## + Fold2: parameter=none
## - Fold2: parameter=none
## + Fold3: parameter=none
## - Fold3: parameter=none
## + Fold4: parameter=none
## - Fold4: parameter=none
## + Fold5: parameter=none
```

```
## - Fold5: parameter=none
## Aggregating results
## Fitting final model on full training set
```

```
# Using the predict() function, predict the expression values of the test dataset
predicted.hcp.bag <- predict(hcp.model.bag, newdata = testing.hcp)

# Calculate the accuracy of the model using the postResample() function
test.hcp.accuracy.bag = postResample(pred = predicted.hcp.bag, obs = testing.hcp$expression)

test.hcp.accuracy.bag
```

```
##      RMSE  Rsquared       MAE
## 1.5901745 0.3140327 1.1200287
```

Exercise 9 - Boosting

Using the "bstTree" method implemented in caret package, train a regression tree to predict gene expression from histone modifications for genes with HCP promoters. Report the accuracy of your model (hcp.model.boost)

```
# Train the model using 'bstTree' method
hcp.model.boost <- train(expression~.,
                         data=training.hcp,
                         method="bstTree",
                         trControl=hcp.ctrl)
```

```
## + Fold1: maxdepth=1, nu=0.1, mstop=150
## - Fold1: maxdepth=1, nu=0.1, mstop=150
## + Fold1: maxdepth=2, nu=0.1, mstop=150
## - Fold1: maxdepth=2, nu=0.1, mstop=150
## + Fold1: maxdepth=3, nu=0.1, mstop=150
## - Fold1: maxdepth=3, nu=0.1, mstop=150
## + Fold2: maxdepth=1, nu=0.1, mstop=150
## - Fold2: maxdepth=1, nu=0.1, mstop=150
## + Fold2: maxdepth=2, nu=0.1, mstop=150
## - Fold2: maxdepth=2, nu=0.1, mstop=150
## + Fold2: maxdepth=3, nu=0.1, mstop=150
## - Fold2: maxdepth=3, nu=0.1, mstop=150
## + Fold3: maxdepth=1, nu=0.1, mstop=150
## - Fold3: maxdepth=1, nu=0.1, mstop=150
## + Fold3: maxdepth=2, nu=0.1, mstop=150
## - Fold3: maxdepth=2, nu=0.1, mstop=150
## + Fold3: maxdepth=3, nu=0.1, mstop=150
## - Fold3: maxdepth=3, nu=0.1, mstop=150
## + Fold4: maxdepth=1, nu=0.1, mstop=150
## - Fold4: maxdepth=1, nu=0.1, mstop=150
## + Fold4: maxdepth=2, nu=0.1, mstop=150
## - Fold4: maxdepth=2, nu=0.1, mstop=150
## + Fold4: maxdepth=3, nu=0.1, mstop=150
## - Fold4: maxdepth=3, nu=0.1, mstop=150
## + Fold5: maxdepth=1, nu=0.1, mstop=150
## - Fold5: maxdepth=1, nu=0.1, mstop=150
## + Fold5: maxdepth=2, nu=0.1, mstop=150
## - Fold5: maxdepth=2, nu=0.1, mstop=150
```

```
## + Fold5: maxdepth=3, nu=0.1, mstop=150
## - Fold5: maxdepth=3, nu=0.1, mstop=150
## Aggregating results
## Selecting tuning parameters
## Fitting mstop = 150, maxdepth = 1, nu = 0.1 on full training set
```

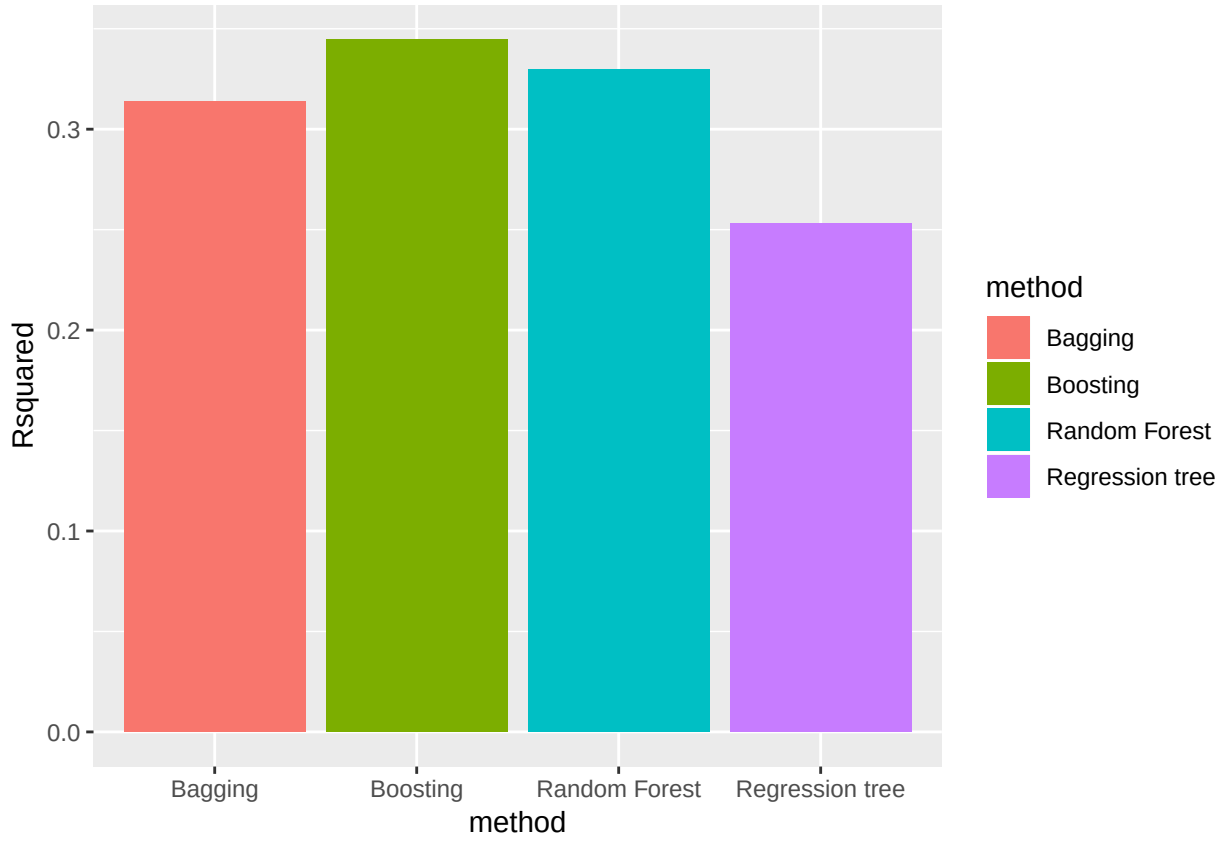
```
# Using the predict() function, predict the expression values of the test dataset
predicted.hcp.boost <- predict(hcp.model.boost, newdata = testing.hcp)

# Calculate the accuracy of the model using the postResample() function
test.hcp.accuracy.boost = postResample(pred = predicted.hcp.boost, obs = testing.hcp$expression)

test.hcp.accuracy.boost
```

```
##      RMSE  Rsquared       MAE
## 1.5358010 0.3447751 1.0675493
```

Exercise 10 - Compare the performance of different models

Use the previously obtained Rsquared values (from postResample() funciton) to compare the accuracy of regression tree, random forest, bagging and boosting models.

```
accuracy = data.frame(Rsquared = c(test.hcp.accuracy[2],
                                   test.hcp.accuracy.bag[2],
                                   test.hcp.accuracy.boost[2],
                                   test.hcp.accuracy.rf[2]),
                      method = c("Regression tree",
                                 "Bagging",
                                 "Boosting",
                                 "Random Forest")
                      )

ggplot(accuracy, aes(x = method, y = Rsquared, fill = method )) +
  geom_bar(stat = "identity")
```



Ensemble methods clearly perform better than individual trees.