Assignment 9 by Team 4

Shun-Lung Chang, Muhammad Hammad Hassan, Kavish Tyagi

In this assignment we aim at predicting **Salary** in the **Hitters** data set which is available in the ISLR package in R.

1. Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

```
dat <- Hitters[!is.na(Hitters$Salary), ]
dat$Salary <- log(dat$Salary)
```

2. Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.

```
train_dat <- dat[1:200, ]
test_dat <- dat[201:nrow(dat), ]
```

3. Fit a tree to the training data, with Salary as the response and the other variables as predictors. Use the `summary()` function to produce summary statistics about the tree, and describe the results obtained. What is the training error rate? How many terminal nodes does the tree have?

```
tree_mod <- tree(Salary ~ ., data = train_dat)
summary(tree_mod)
```

```
Regression tree:
tree(formula = Salary ~ ., data = train_dat)
Variables actually used in tree construction:
[1] "CAtBat"  "CRuns"   "AtBat"   "Walks"   "Hits"    "CHits"   "CRBI"
[8] "PutOuts"
Number of terminal nodes:  10
Residual mean deviance:  0.1665 = 31.64 / 190
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.003000 -0.221300  0.009429  0.000000  0.246800  2.342000
```

```
pred_train <- predict(tree_mod, train_dat)
mse_train <- mean((pred_train - train_dat$Salary) ^ 2)
mse_train
```

```
[1] 0.1581972
```

4. Type in the name of the tree object in order to get a detailed text output. Pick one of the terminal nodes, and interpret the information displayed.

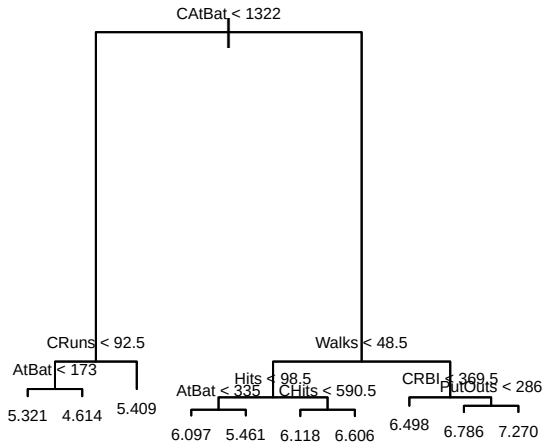
```
tree_mod
```

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 200 166.4000 5.940
   2) CAtBat < 1322 70  23.3000 4.971
     4) CRuns < 92.5 43  12.6000 4.696
       8) AtBat < 173 5   7.2760 5.321 *
       9) AtBat > 173 38   3.1220 4.614 *
     5) CRuns > 92.5 27   2.2610 5.409 *
   3) CAtBat > 1322 130  42.0400 6.462
     6) Walks < 48.5 80  20.2600 6.232
      12) Hits < 98.5 41   9.8330 6.019
        24) AtBat < 335 36   6.8960 6.097 *
        25) AtBat > 335 5   1.1580 5.461 *
      13) Hits > 98.5 39   6.6220 6.456
        26) CHits < 590.5 12   1.8470 6.118 *
        27) CHits > 590.5 27   2.7950 6.606 *
     7) Walks > 48.5 50  10.8100 6.829
      14) CRBI < 369.5 16   0.9873 6.498 *
      15) CRBI > 369.5 34   7.2280 6.985
        30) PutOuts < 286 20   2.6210 6.786 *
        31) PutOuts > 286 14   2.6770 7.270 *
```

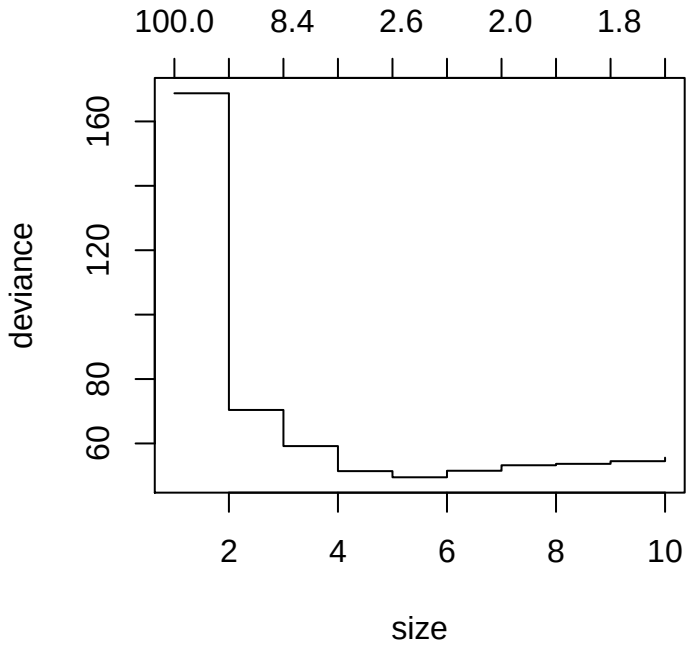
5. Create a plot of the tree, and interpret the results.

```
plot(tree_mod)
text(tree_mod, cex = 0.5)
```



6. Apply the `cv.tree()` function to the training set in order to determine the optimal tree size. Produce a plot with tree size on the x-axis and cross-validated classification mean squared error on the y-axis.

```
set.seed(42)
cv_model <- cv.tree(tree_mod)
plot(cv_model)
```



7. Which tree size corresponds to the lowest cross-validated MSE?

```
cv_model$size[which(cv_model$dev == min(cv_model$dev))]
```

```
[1] 5
```

8. Produce a pruned tree corresponding to the optimal tree size obtained using cross-validation. If cross-validation does not lead to selection of a pruned tree, then create a pruned tree with five terminal nodes.

```
tree_pruned <- prune.tree(tree_mod, best = 5)
tree_pruned
```

```
node), split, n, deviance, yval
      * denotes terminal node

 1) root 200 166.400 5.940
   2) CAtBat < 1322 70  23.300 4.971
     4) CRuns < 92.5 43  12.600 4.696 *
     5) CRuns > 92.5 27   2.261 5.409 *
   3) CAtBat > 1322 130  42.040 6.462
     6) Walks < 48.5 80  20.260 6.232
```

```
      12) Hits < 98.5 41   9.833 6.019 *
      13) Hits > 98.5 39   6.622 6.456 *
     7) Walks > 48.5 50  10.810 6.829 *
```

9. Compare the training MSE between the pruned and un-pruned trees. Which is higher?

```
pred_train_pruned <- predict(tree_pruned, train_dat)
mse_train_pruned <- mean((pred_train_pruned - train_dat$Salary) ^ 2)
mse_train_pruned
```

```
[1] 0.2106276
```

10. Compare the test error rates between the pruned and unpruned trees. Which is higher?

```
pred_test <- predict(tree_mod, test_dat)
mse_test <- mean((pred_test - test_dat$Salary) ^ 2)
mse_test
```

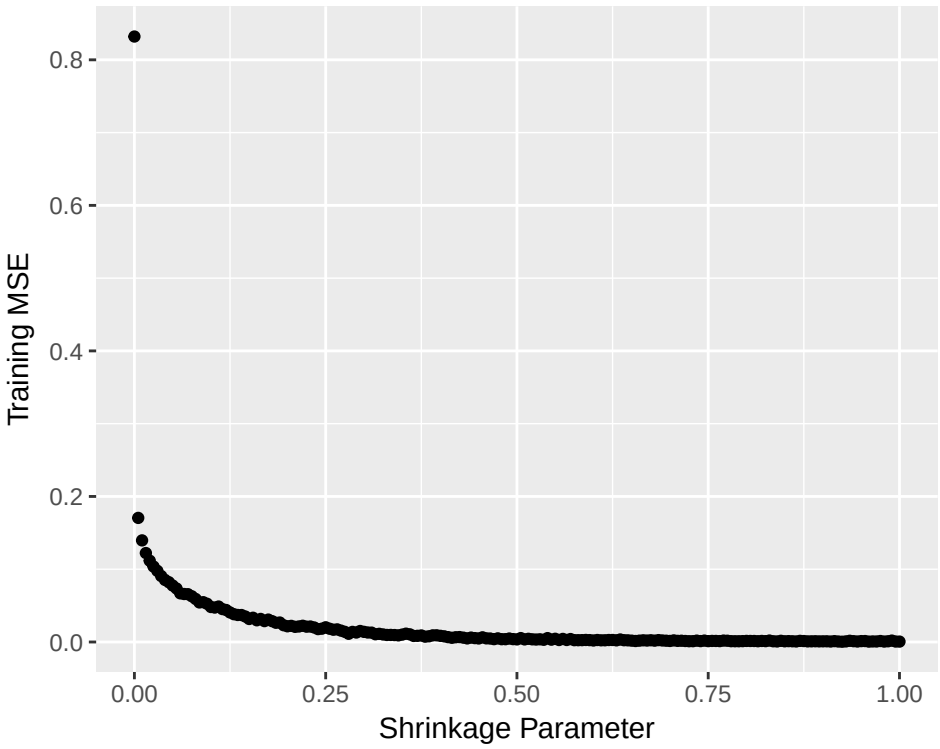
```
[1] 0.3116231
```

```
pred_test_pruned <- predict(tree_pruned, test_dat)
mse_test_pruned <- mean((pred_test_pruned - test_dat$Salary) ^ 2)
mse_test_pruned
```

```
[1] 0.3983201
```

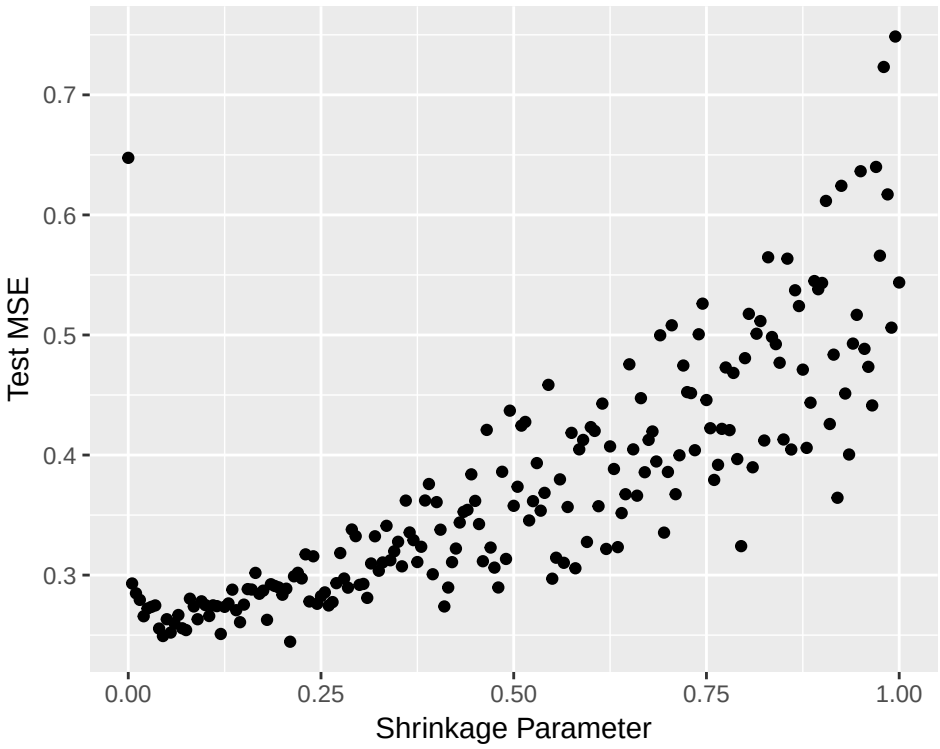
11. Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter lambda. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

```
mse_train <- sapply(seq(0, 1, by = 0.005), function(lambda) {
    mod <- gbm(Salary ~ .,distribution = "gaussian",
        data = train_dat, n.trees = 1000, shrinkage = lambda)
    mod$train.error[length(mod$train.error)]
})
```



12. Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

```
mse_test <- sapply(seq(0, 1, by = 0.005), function(lambda) {
    mod <- gbm(Salary ~ .,distribution = "gaussian",
        data = train_dat, n.trees = 1000, shrinkage = lambda)
    pred <- predict(mod, test_dat, n.trees = 1000)
    mean((pred - test_dat$Salary) ^ 2)
})
```



13. Which variables appear to be the most important predictors in the boosted model?

```
gbm_mod <- gbm(Salary ~ .,distribution = "gaussian",
               data = train_dat, n.trees = 1000, shrinkage = 0.225)
kable(summary(gbm_mod, plotit = FALSE)[1:5, ])
```

	var	rel.inf
CAtBat	CAtBat	15.440660
CRBI	CRBI	10.344093
CRuns	CRuns	9.311075
PutOuts	PutOuts	8.849853
Walks	Walks	7.382570

14. Now apply bagging to the training set. What is the test set MSE for this approach?

```
bagging_mod <- bagging(Salary ~ ., data = train_dat)
pred_test_bagging <- predict(bagging_mod, test_dat)

mse_test_bagging <- mean((pred_test_bagging - test_dat$Salary) ^ 2)
mse_test_bagging
```

```
[1] 0.3245806
```

15. Now apply random forest to the training set. What is the test set MSE for this approach? Which variables appear to be the most important predictors in the random forest model?

```
rf_mod <- randomForest(Salary ~ ., data = train_dat)

pred_test_rf <- predict(rf_mod, test_dat)
mse_test_rf <- mean((pred_test_rf - test_dat$Salary) ^ 2)
mse_test_rf
```

```
[1] 0.2121882
```

```
d <- data.frame(Features = rownames(rf_mod$importance),
                Importance = rf_mod$importance[, 1])
kable(d[order(d$Importance, decreasing = TRUE)[1:5], ])
```

	Features	Importance
CAtBat	CAtBat	38.11250
CHits	CHits	26.87855
CRuns	CRuns	23.55741
CRBI	CRBI	15.09501
CWalks	CWalks	14.24258