W23 STAT 362 R for Data Science

Assignment 6

Due: 7 Apr 11:59pm.

Q1 (30 points): Run the following code, which simulates 100 realized values of two correlated random variables X and Y.

```
set.seed(1)
n <- 1000
X <- rnorm(n, 0.01, 0.05)
Y <- 0.5 * X + rnorm(n, 0, 0.05)
data <- cbind(X, Y)
```

Suppose that you are interested in estimating σ_X/σ_Y, where σ_X is the standard deviation of X and σ_Y is the standard deviation of Y. Since you can use the sample standard deviation to estimate the (theoretical) standard deviation, a natural estimate of σ_X/σ_Y will be s_X/s_Y, where s_X and s_Y are the sample standard deviations of $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$, respectively.

Use bootstrap to estimate the standard error of s_X/s_Y using the above simulated data.

Q2-Q8 (70 points): Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients. Download the dataset `concrete.csv` from onQ. For information about this dataset, see https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength

Run the following code

```
concrete <- read.csv("concrete.csv") # write your own path
names(concrete)[1] <- "cement"
set.seed(2)
index <- sample(nrow(concrete), 700) # indices corresponding to the training data
concrete_train <- concrete[index, ]
concrete_test <- concrete[-index, ]
```

Q2: Fit a regression tree using `tree()` from the package `tree` with `strength` as the response and other variables as `predictors` using only the training data. Find the mean squared test error for the regression tree using the testing data.

Q3: Plot the regression tree in Q2.

Q4: Fit a linear regression model with `strength` as the response and other variables as `predictors` using the training data. Find the mean squared test error for the model using the testing data.

Q5: Fit a random forest with `strength` as the response and other variables as `predictors` using the training data. Find the mean squared test error for the model using the testing data.

Q6: Create a variable importance plot for the model in Q5. Which variable is the "most important"?

Q7: Install the package `FNN`. The function `knn.reg` in the package `FNN` can be used to perform k nearest neighbor regression. The usage of this function is as follows:

```
knn.reg(train, test, y, k)
```

`train`: matrix or data frame of training set cases

`test`: matrix or data frame of test set cases

`y`: response of each observation in the training set

`k`: number of neighbours considered

The output of the function is a list and you can extract the prediced values using `$pred`.

Now, perform knn regression with `strength` as the response using all the remaining variables as the predictors, obtain the predicted values on the test data and compute the mean squared test error.

Remember to scale the data (no need to scale the response) first as in Assignment 4

Q8: In this particular dataset and particular split of training and testing data, which methods above give you the smallest test error?