STAT 380 – Assessing Model Accuracy Part 2 (Lecture 7)

EXAMPLE 1: (Sample data dictionary) L06_Insurance_m.csv contains information about a number of health insurance policies. In particular, the data set contains some attributes of the policy holder (such as age, sex, etc.) and the total charges billed by the health care provider. (While similar, this dataset is different than the Insurance.txt dataset used earlier in the course.) Here are the variables included:

- age: age of primary beneficiary
- sex: sex of primary beneficiary
- bmi: Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9
- children: Number of children covered by the health insurance policy (i.e., the number of dependents)
- smoker: Status indicating whether the person is a smoker (options include 'yes' and 'no')
- region: the beneficiary's residential area in the US (options include northeast, southeast, southwest, northwest).
- charges: Individual medical costs as billed by health insurance

Read L06_Insurance_m.csv into R and name the result Ins.

a. In Example 2 (Lecture 6), we saw that there is at least 1 NA in the dataset. Write a function called countNA that takes a single argument call dat, where dat represents a vector (column of a dataframe). The function should return the number of NA values in the column. (Code may be found in STAT380_L7.Rmd.)

```
#This function will count the number of NA's in a vector data
countNA <- function(dat){
  numNA <- sum(is.na(dat))
  return(numNA)
}
```

b. Apply the function to each column of Ins. How much missing data is present?

Inefficient approach:

```
## Example 1b - Apply function to one column at a time
```{r}
#Apply countNA to each column of Ins
countNA(dat = Ins$age)
countNA(dat = Ins$sex)
countNA(dat = Ins$bmi)
countNA(dat = Ins$children)
countNA(dat = Ins$smoker)
countNA(dat = Ins$region)
countNA(dat = Ins$charges)
```
```

Efficient approach:

```
## Example 1b - Apply function to all columns at once
```{r}
apply(X = Ins, MARGIN = 2, FUN = countNA)
```
```

```
     age      sex      bmi children   smoker   region  charges
       0        0        0        0        0        0        1
```

c. Remove any observations with missing data from the dataset. Name the new dataset Ins. (You should have 1337 observations after this step.)

```
## Example 1c - Remove any observation (row) that includes an NA
```{r}
Ins <-
  Ins %>%
  na.omit()
```
```

d. To learn how to reproducibly generate random numbers in R, consider the following examples.

```r
#Randomly generate 9 numbers from 1 to 20
sample(x = 1:20, size = 9, replace = TRUE)
```

```
## [1]  2  7 19 13 13 16  9 16 12
```

```r
#Randomly generate 9 numbers from 1 to 20 using a seed of 123
set.seed(1234)
sample(x = 1:20, size = 9, replace = TRUE) #16  5 12 15  9  5  6 16  4
set.seed(NULL)

#Randomly generate 3 numbers from 1 to 20 3 times using a seed of 123
set.seed(1234)
sample(x = 1:20, size = 3, replace = TRUE) #16  5 12
sample(x = 1:20, size = 3, replace = TRUE) #15  9  5
sample(x = 1:20, size = 3, replace = TRUE) #6 16  4
set.seed(NULL)

#What does set.seed(NULL) do?
set.seed(1234)
sample(x = 1:20, size = 3, replace = TRUE) #16  5 12
sample(x = 1:20, size = 3, replace = TRUE) #15  9  5
set.seed(NULL)
sample(x = 1:20, size = 3, replace = TRUE)
```

e. Using a seed of 123, perform an 80/20 training/testing split.

IDEA: Randomly select 80% of the row numbers and store the result as train_ind. Those row numbers determine the training set. The unselected row numbers, which correspond to the remaining 20%, determine the testing set.

```r
set.seed(123)
train_ind <- sample(x = 1:nrow(Ins), size = floor(0.8*nrow(Ins)))
set.seed(NULL)

Train <- Ins[train_ind, ]
Test <- Ins[-train_ind, ]
```

NOTE: There are other ways to do this in R; however, since there is randomness involved and we must agree on a correct answer for grading purposes, the expectation is that you use this method.

f. Build a regression model for predicting charges based on person's age using the training data. Write the estimated regression equation.

g. What proportion of the variation in charges is explained by the model containing age?

h. Using the model from f., predict the charges for each person in the testing data. HINT: Make sure the number of predictions matches the number of rows in Test.

i. Compute MSE and RMSE for the testing data.

NOTE: The holdout set (or validation set) approach works well on large datasets. On small datasets, the results are sensitive to the split (i.e., the result can vary a lot based on which observations were assigned to the training set and which are assigned to the testing set).

j. Explore some multiple linear regression models. For each model you try, complete a row in the table shown below. Which gives the best MSE (or best RMSE) based on the testing set?

Predictor Variables in the model	R^2	R^2_{adj}	Test Set MSE	Test Set RMSE