CME 195/Stats 195 HW#2
Due April 19, 2016 *Instructions*:
1. Upload R script to Coursework dropbox using the filename <your SUNetid>stats195hw2.R).
E.g., I would name my submission "hgm7stats195hw2.R".
2. Follow any indications below on naming your functions/variables.

## Problem 1

In these two problems we will compute the training error and test error of a linear model. When you fit a model to data, the model usually better fits the data it was fit to than new data (the training error is usually lower than the test error). You can read up on training versus test error on Wikipedia, etc., but do not need this background to do the problem.

(a) Read the data set

`http://vincentarelbundock.github.io/Rdatasets/csv/Ecdat/Hedonic.csv`

into a data frame in R. The file is in comma-separated value ("csv") format with column labels and a column of row numbers. Take a glance at the resulting data frame to make sure it looks OK. The data set contains a number of measurements (the columns) on the homes in certain regions (the rows).

(b) Take your data frame from (a) and create a new data frame consisting of the columns `mv` and `age`. Call the new data frame `dat`. `mv` refers to median home value in the region and will be our response. `age` is a measure of the typical age of homes in the region and will be our predictor.

(c) Perform a linear regression of `mv` against `age`, but only use the first half of all the observations (the rows) of `dat`, not all of `dat`. We will use the remaining half of the observations to test our linear regression later on. You can separate out the first half of the observations using tools we have already encountered. Another option is to use the function `cut` and/or the `subset=` parameter of `lm`, which we haven't used in class but you can learn about in the help pages.

Create a scatterplot of the points used in the regression, along with the regression line, as we have done in class. Try to annotate the plot with a useful title, axis labels, perhaps a sub-title, etc.

(d) What is the mean sum of squares of the residuals? The residual for a given value of `age` is the difference between, on the one hand, the actual value of `mv` corresponding to that given value of `age` (i.e., the point on your scatter plot where x-axis is equal to the given value of `age`) and, on the other hand, the predicted value of `mv` corresponding to that given value of `age` (i.e., the point on your regression line where the x-axis is equal to the given value of `age`). You can access the residuals in a linear model `my.lm` using `my.lm$residuals`. Given numbers $x_1, \ldots, x_n$, the mean sum of squares is $\frac{1}{n}\sum_1^n x_i^2$. This mean sum of squares, corresponding to the first half of the data set, is the *training error* or *in-sample error*. *Ans.* 0.07938241.

(e) Apply your linear model from (c) to the second half of the `age` observations in `dat`, i.e., use the model to predict home values of the second half of the observations. You can go about this using `predict.lm(my.lm, new.observations)`, where `my.lm` is your linear model from (c) and `new.observations` is a *data frame* containing the second half of the `age` observations. Another way to go about this is to look at the slope and intercept of the regression line (e.g., `summary(my.lm)`) and compute the value of this line when the independent variable (i.e., the x-values) assumes the values of the `age` observations in the second half of the data set. Either way, you should get a vector of values of the same length as the number of observations in the second half of the `dat`, each entry representing a predicted value of `mv`.

Make a scatterplot of the second half of `dat`, i.e., the second half of the `mv` observations versus the second half of the `age` observations. Use `lines` to plot the predicted values against the second half of the `age` observations.

(f) How well did the linear model predict the second half of the `mv` observations? Take the difference of the predictions in (e) and the second half of the `mv` observations, and obtain the mean sum of squares of these differences. This mean sum of squares, corresponding to the second half of the data set, is the *test error* or *out-of-sample error*. *Ans.* 0.2098782.

## Problem 2

Write a function `two.errors` to perform the computations in Problem 1 on arbitrary predictor and response variables. The function should take two arguments `x` and `y` (corresponding to `age` and `mv` in Problem 1) and print out the training error and test error obtained by splitting the input vectors in half, as in Problem 1 (you can assume the inputs will have the same length and that this length will be an even number, as was the case in Problem 1).

```
two.errors <- function(x,y) {

   ... # fill in computations of training error and test error here

   cat("training error: ", training.error, "\n")
   cat("test error: ", test.error, "\n")
}
```

*Extra credit.* Modify `two.errors` to first randomly re-order the input vectors before computing the training and test error. Remember that corresponding elements of `x` and `y` must permute together. Call this function `two.errors.shuffle`. You would typically shuffle the data in this way before computing the errors.

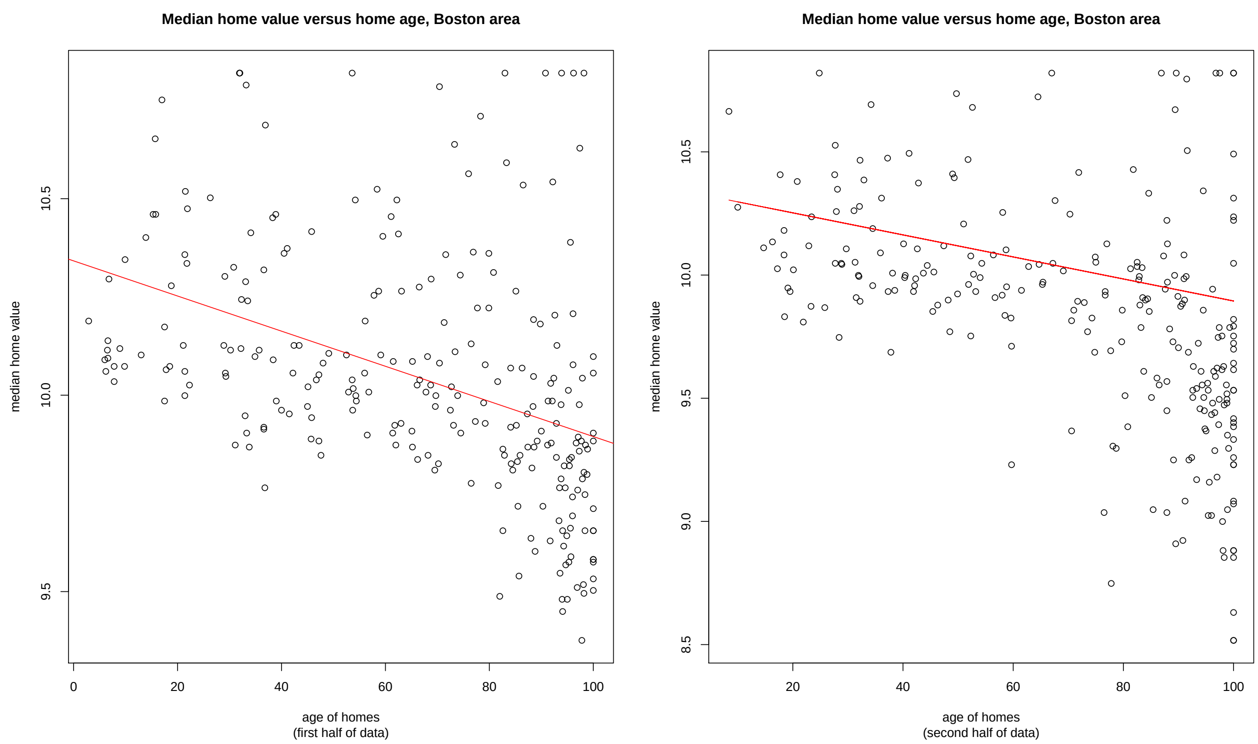


Figure 1: 1(c) and 1(e)