ISyE 7406: Data Mining & Statistical Learning
HW#2

In this homework, please write your software codes (in R, Python, etc.) by yourself, and no collaborations allowed! It is **cheating** *if you copy and paste codes or solutions from your classmates or online.*

Problem 1 (Linear Regression in R/Rstudio) Consider the data set "*fat*" in the "faraway" library of R. This data file is also available at Canvas, Or if you save this file "fat.csv" in your laptop, say, in the folder "C://Temp", you can read it in R as

```
fat  <- read.table(file = "C://Temp/fat.csv", sep=",", header=TRUE);
```

The dataset *fat* has 252 observations and 18 variables, and, for more detailed description, see the link `(http://cran.r-project.org/web/packages/faraway/faraway.pdf) (page #37)`.
For more background information, you can also see

`http://en.wikipedia.org/wiki/Body_fat_percentage`

The purpose of this homework is to help you better understand linear regression. Here we assume that the percentage of body fat using Brozek's equation (`brozek`, the first column) as the response variable, and the other 17 variables as potential predictors. We will use several different statistical methods to fit this dataset in the problem of predicting `brozek` using the other 17 potential predictors. For that purpose, it is useful to split it into the following sub-tasks. Also below we use R to demonstrate our main ideas, but you are free to use Python or other software.

(a) First, we should split the original data set into disjoint training and testing data sets, so that we can better evaluate and compare different models. One possible simple way is to random select a proportion, say, 10% of observations from the data for use as a **test** sample, and use the remaining data as a *training* sample building different models. **Note that in practice, it is more reasonable to select much larger proportion, say** 30% **or** 20%**, as testing sample**. Here we chose only 10% as the testing sample, so that we can list those testing observations explicitly below.

```
n = dim(fat)[1];       ### total number of observations
n1 = round(n/10);      ### number of observations randomly selected for testing data
## To fix our ideas, let the following 25 rows of data as the testing subset:
flag = c(1,  21,  22,  57,  70,  88,  91,  94, 121, 127, 149, 151, 159, 162,
         164, 177, 179, 194, 206, 214, 215, 221, 240, 241, 243);
fat1train = fat[-flag,];    fat1test  = fat[flag,];
```

(b) Second, for the training data "`fat1train`," do some **exploratory (or preliminary) data analysis** such as scatter plots or summary statistics of some variables that you feel are important (e.g., explain the unusual pattern).

(c) Based on the **training** data "`fat1train`," build the following models

(i) Linear regression with all predictors.

(ii) Linear regression with the best subset of $k = 5$ predictors variables;

(iii) Linear regression with variables (stepwise) selected using AIC;

(iv) Ridge regression;

(v) LASSO;

(vi) Principal component regression;

(vii) Partial least squares.

(d) Use the models you find in part (c) to predict the response in the **testing** data "`fat1test`" in part (a). Report the performance of each model $\hat{f}$ on the testing data, say, $\{(Y_i^{test}, \mathbf{x}_i^{test})\}_{i=1}^{n_1}$. Here $n_1 = 25$ and we assume that the performance of each model is evaluated by the following testing error

$$TE = \frac{1}{n_1} \sum_{i=1}^{n_1} [Y_i^{test} - \hat{f}(\mathbf{x}_i^{test})]^2.$$

(e) The above steps are sufficient when one has a large data set. As mentioned in HW#1, for a relatively small data, one may want to use Cross-Validation to further assess the robustness of each method. Using **Monte Carlo Cross-Validation algorithm** that repeats the above computation $B = 100$ times, compute and compare the "average" performances of each model mentioned in part (c).

Write a report to summarize your findings. The report should include (i) **Introduction,** (ii) **Exploratory (or preliminary) Data Analysis** of training data in part (a), (iii) **Methods**, (iv) **Results** and (v) **Findings.** Also see the guidelines on the final report of our course project. Please attach your R code (without, or with limited, output) in the appendix of your report, and please do not just dump the R output in the body of the report.

Remark: Note that in part (e), the same original data is repeatedly used B times as a whole, but it is used differently at different loops due to the different split of training and testing data. The idea of repeating the similar data analysis process B times is essential in many well-known statistical tools such as **bootstrapping** and **Random Forest**, and has been widely used in other fields such as **bioinformatics** or **computational biology**.

For your convenience, I also post some R codes on the last page of this pdf file of this homework at Canvas that might be useful. Please feel free to modify those R codes if you want. To encourage everyone learn the materials, each student must write their R or any other software codes by themselves, and no collaborations allowed! It is **cheating** if you copy and paste your classmates' computing codes.

Appendix: the following R code might be useful, and you are free to use Python or other software:

```
### Read the data
### Suppose you save the data file ‘‘fat.csv" in the folder ‘‘C://Temp" of your laptop,
fat  <- read.table(file = "C://Temp/fat.csv",  sep=",", header=TRUE);

### Split the data as in Part (a)
n = dim(fat)[1];      ### total number of observations
n1 = round(n/10);     ### number of observations randomly selected for testing data
## To fix our ideas, let the following 25 rows of data as the testing subset:
flag = c(1,  21,  22,  57,  70,  88,  91,  94, 121, 127, 149, 151, 159, 162,
         164, 177, 179, 194, 206, 214, 215, 221, 240, 241, 243);
fat1train = fat[-flag,];
fat1test  = fat[flag,];

###In Part (b)-(d), Please see the R code for linear regression at Canvas.
### Please write your own R or other software code to analyze the training data "fat1train"
### and evaluate different models on the testing data "fat1test".

### Part (e): the following R code might be useful, and feel free to modify it.
###    save the TE values for all models in all $B=100$ loops
B= 100;            ### number of loops
TEALL = NULL;      ### Final TE values
set.seed(7406);    ### You might want to set the seed for randomization

for (b in 1:B){
  ### randomly select 25 observations as testing data in each loop
  flag <- sort(sample(1:n, n1));
  fattrain <- fat[-flag,];
  fattest  <- fat[flag,];
  ### you can write your own R code here to first fit each model to "fattrain"
  ### then get the testing error (TE) values on the testing data "fattest"
  ### Suppose that you save the TE values for these five models as
  ###   te1, te2, te3, te4, te5, te6, te7, respectively, within this loop
  ###   Then you can save these 5 Testing Error values by using the R code
  ###
  TEALL = rbind( TEALL, cbind(te1, te2, te3, te4, te5, te6, te7) );
}
dim(TEALL);  ### This should be a Bx7 matrices
### if you want, you can change the column name of TEALL
colnames(TEALL) <- c("mod1", "mod2", "mod3", "mod4", "mod5", "mod6", "mod7");

## You can report the sample mean and sample variances for the seven models
apply(TEALL, 2, mean);
apply(TEALL, 2, var);
### END ###
```