Homework Assignment 1

The purpose of this homework is to apply linear regression on a real data set. We will consider three ways of improving prediction accuracy: feature selection, ridge regression and the lasso. We will split the data set into two parts (training and test set), then fit the models on the training set (including choice of the best parameter using cross-validation) and test their performance on the separate test set.

1. Download housing data set, concerning housing values in the suburbs of Boston. The detailed explanation what the input and output variables mean, can be found here.
2. Read the data set into R (e.g. using `read.table`). Divide the set into two parts. Use the first 350 examples as a training set and the rest (156 examples) as a test set (you can access e.g. first 350 objects by `df[1:350,]`, where `df` is a data frame).
3. Fit the least squares model to the training set using `lm` (hint: to specify that `MEDV` is the output variable, while all the others are input variables, you can use formula '`MEDV` $\sim$ `.`').
4. Report estimated coefficients, their standard errors, and statistical significance (all you can get by using `summary`).
5. Estimate the prediction accuracy of regression using 5-fold cross-validation (use the `crossval` function from the `bootstrap` package or you can also write cross-validation by yourself using `sample` to draw a random permutation). Compare the cross-validation error with the residual sum of squares ("training error").
6. Perform the full subset-search using the function `regsubsets` from package `leaps`. The function returns for $k = 1, \ldots, p$ the best subset of size k in terms of mean squared error on the training set. By cross-validation, choose the best k (to access the subsets, you can use `summary(regsubsets.output)$which`, where `regsubsets.output` is the output of the `regsubsets` function).
7. Apply ridge regression to the training set. Use function `lm.ridge` from package `MASS`. Choose several values of shrinkage and get the best one using cross-validation (you can use either `select` procedure for `lm.ridge` output, which uses "approximated" cross-validation (GCV), or you can use your own cross-validation method). Hint: you can access regression coefficients by `ridge.model$coefs` (where `ridge.model` is the object returned by `lm.ridge`), but they are normalized. For prediction, you must "denormalize" them, e.g.:

   ```
   predict.ridge <- function(ridge.model, x_train, y_train, x_test) {
     intercept = -sum(ridge.model$coef * colMeans(x_train) / ridge.model$scales)
          + mean(y_train)
     ridge.coeffs = ridge.model$coef / ridge.model$scales
     ypred = as.matrix(x_test) %*% as.vector(ridge.coeffs) + intercept
   }
   ```

   (where `x_train` is the design matrix (training input), `y_train` – training output vector, `x_test` – test input matrix).
8. Apply lasso regression to the training set. Use functions `lars` and `cv.lars` with option `type="lasso"` to fit the lasso model and choose best shrinkage with cross-validation. Use `predict.lars` for prediction.
9. Run all three methods (best-subset, ridge and lasso) with optimally tuned parameters (subset size or shrinkage) on the test set. Report prediction accuracies and draw conclusions.

Note. Include comments when necessary. You are also encouraged to include plots presenting some of the results.