Lasso estimation in LM and GLM

Write a report that contains the results of the computations that you are asked to carry out below, as well as the explanation of what you are doing. The main text (2 or 3 pages) should include pieces of source code and graphical and numerical output.
Upload your answers in a .pdf document (use LaTeX or R Markdown, for instance), as well as the source code (*.R or *.Rmd, for instance). Your work must be reproducible.

1. Lasso for the Boston Housing data

The Boston House-price dataset concerns housing values in 506 suburbs of Boston corresponding to year 1978. They are available here:
`https://archive.ics.uci.edu/ml/datasets/Housing`
This is the list of the available variables:

```
1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centres
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV: Median value of owner-occupied homes in $1000's
```

The Boston House-price corrected dataset (available in `boston.Rdata`) contains the same data (with some corrections) and it also includes the UTM coordinates of the geographical centers of each neighborhood.

1. For the Boston House-price corrected dataset use Lasso estimation (in `glmnet`) to fit the regression model where the response is `CMEDV` (the corrected version of `MEDV`) and the explanatory variables are the remaining 13 variables in the previous list. Try to provide an interpretation to the estimated model.

2. Use `glmnet` to fit the previous model using ridge regression. Compare the 10-fold cross validation results from function `cv.glmnet` with those you obtained in the previous practice with your own functions.

2. SPAM E-mail Database

The zip file `SPAM E-mail Database` contains a data base of e-mails classified in two classes: “SPAM” or “NO SPAM”. It is available in Atenea. The data were downloaded on 03-05-2016 from

`http://web.stanford.edu/~hastie/ElemStatLearn/datasets/spam.info.txt`

`http://web.stanford.edu/~hastie/ElemStatLearn/datasets/spam.data`

`http://web.stanford.edu/~hastie/ElemStatLearn/datasets/spam.traintest`

From the description file:
The “spam” concept is diverse: advertisements for products/web sites, make money fast schemes, chain letters, pornography... Our collection of spam e-mails came from our postmaster and individuals who had filed spam. Our collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either have to blind such non-spam indicators or get a very wide collection of non-spam to generate a general purpose spam filter.

Attribute Information:

- The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.
- Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail.
- The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

Task:

1. Use the script `spam.R` to read the data from the SPAM e-mail database.
2. Divide the data into two parts: 2/3 for the training sample, 1/3 for the test sample. You should do it in a way that 2/3 of the SPAM e-mails are in the training sample and 1/3 in the test sample, and that the same happens for NO SPAM e-mails.
3. Consider the following classification rules:
   - Logistic regression fitted by maximum likelihood (IRWLS, `glm`).
   - Logistic regression fitted by Lasso (`glment`).
   - (Optional) k-nn binary regression (you can use functions `knn` and `knn.cv` from the R package `class`).

   Use the training sample to fix the tuning parameters (when needed) and to estimate the model parameters (when needed).
4. Use the test sample to compute the confusion matrix and the misclassification rate for each rule.