Comparing discriminant rules. ROC curve and other methods

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 should include pieces of source code and graphical and numerical output.

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 SPAM e-mail are 2/3 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 three classification rules:
   - Logistic regression fitted by maximum likelihood (IRWLS, `glm`).
   - Logistic regression fitted by Lasso (`glment`).
   - k-nn binary regression (you can use your own implementation or functions `knn` and `knn.cv` from the R package `class`).

   Use the training sample to fix the tunning parameters (when needed) and to estimate the model parameters (when needed).
4. Use the test sample to compute and plot the ROC curve for each rule.
5. Compute also the misclassification rate for each rule when using the cut point $c = 1/2$.
6. Compute ℓ_{val} for each rule.