STA 437/1005, Fall 2010 — Assignment #2

Due at start of lecture on December 6. Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper-left corner, without any folder or other packaging around it. If really necessary, you can submit it by email (to radford@stat.utoronto.ca*), but please do this only if you can't easily hand in a paper copy.*

I will go over the solution in class on December 8. If you have a legitimate excuse (eg, illness) for not handing in this assignment on time, please contact me as soon as possible, to arrange to hand it in before class on December 8, or to have the marks for this assignment taken from other work.

This assignment is worth 10% of the course grade. It is to be done by each student individually. You may discuss this assignment in general terms with other students, but the work you hand in should be your own. In particular, you should not leave any discussion of this assignment with any written notes or other recordings, nor receive any written or other material from anyone else by other means such as email.

For this assignment, you will look further at the same gene expression dataset you looked at in Part II of Assignment #1. In particular, you will see how logistic regression works for this data, and look at testing which genes are related to cancer recurrence, using both the Bonferroni correction approach, and the False Discovery Rate approach to handling the problem of multiple tests. With complex datasets such as this one, many variations in the analysis are possible. In the last two parts of this assignment, you will explore whether better results can be obtained by adjusting measurements for each patient, or by using logistic regression with quadratic terms.

You should produce a report explaining your findings, following the outline of what to do given below. You should justifying your findings with suitable output or plots from R, but hand in only a reasonable amount of output, not every conceivable plot. You *must* hand in a listing of the R commands you used for the analysis, including the commands used to produce the plots you hand in. You do not need to repeat in your report what I have said about the dataset.

The dataset, along with some hints on using R for this assignment, and some relevant lecture notes, are (or soon will be) available from the course web page, at

`http://www.utstat.utoronto.ca/~radford/sta437/`

You should check the web page regularly in case there are any corrections made to the assignment (any corrections will be highlighted in bold at the top of the web page).

Part 1: logistic regression

In Assignment 1, you tried to predict cancer recurrence by fitting a regression model by least squares. This obviously does not match the usual assumption that residuals are normally distributed, since the response is either 0 or 1. Here, you should try modelling cancer recurrence using logistic regression. In R, you can use commands of the form

```
fit <- glm (recur ~ ..., family = binomial)
print(summary(fit))
```

where ... specifies the covariates to use.

You should start by using as covariates the first five principal components found from the covariance matrix of the log of the expression levels, with no outliers removed, and compare with the corresponding result using least squares (with the `lm` function, as for Assignment 1). You should compare both the regression coefficients found, and the predicted probabilities of recurrence, which you can obtain with `predict(fit,type="response")`, and comment on what you find. You should then investigate whether some subset of these covariates would give a better logistic regression model.

Part 2: t tests with Bonferroni correction

Least squares or logistic regression focuses on what covariates are useful for predicting recurrence, which does not necessarily identify all variables that are related to recurrence, and does not clearly indicate how statistically significant any observed association is. In this part, you will try to identify which genes are associated with recurrence using t tests, with the Bonferroni correction for multiple testing.

You should use two-sample t tests on the log of the expression levels for each gene, for patients with and without recurrence, without assuming that the variance in each group is the same. This can be done with the `t.test` function. If `g` is the vector of logs of gene expression levels in the 39 patients, and `recur` is the vector of recurrence indicators, the command

```
t.test (g[recur==0,i], g[recur==1,i], var.equal=FALSE) $ p.value
```

will return the p-value for a test of the null hypothesis that the mean expression levels for gene i are the same in the two groups. You can use a `for` loop in R to do this for all 1644 genes.

The Bonferroni adjustment is to multiply the p-value from `t.test` by the number of tests done. You should see how many genes are then found to have expression levels that differ significantly between the two groups, when the significance level is set as 0.1, 0.05, 0.02, 0.01, and 0.005. Compare these numbers with the numbers of genes that are significant at these levels without the Bonferroni adjustment.

Part 3: False Discovery Rates for t tests

As discussed in class, the simultaneous guarantee from the Bonferroni adjustment may be too stringent. Here, you will look at identifying genes whose expression levels relate to recurrence using False Discovery Rate (FDR) to control for multiple testing, using the same p-values from t tests as in Part 2.

The method discussed in class puts an upper bound on FDR if the i hypotheses with smallest p-values are rejected of $mp_{(i)}/i$, where m is the total number of hypotheses tested.

If we have an estimate, $\widehat{m}_0$, of the number of true null hypotheses, we can instead estimate (rather than upper bound) the FDR as $\widehat{m}_0 p_{(i)}/i$. You should look at both the upper bound for FDR and this estimate for FDR, using an estimate $\widehat{m}_0$ found by counting the number of p-values above 0.7, which we might assume are for null hypotheses that are true (or at least almost true).

Discuss the results you obtain, comparing with the results using the Bonferroni adjustment, and noting any odd or otherwise interesting aspects of what you see. You may find it useful to look at various plots, such as a histogram of p-values or plots of estimated FDR for different numbers of rejected hypotheses.

Part 4: adjustment of data for each patient

We don't know all the details of how the gene expression measurements were processed for this dataset. However, it seems possible that the measurements for each patient might be scaled by some amount that is different for each patient, due to variation in some physical aspect of the measurement process. This scaling would show up as a shift in the log of the expression levels.

In this part, you will investigate whether trying to correct for such a possible shift helps. You can find the mean of the log expression levels for each patient with `rowMeans(g)`, where `g` is the matrix or data frame of logs of expression levels for genes. (This should produce a vector of length 39, one element per patient.) You can then produce a new dataset, in which the mean for each patient is subtracted from the expression levels for that patient, with an expression like `g-rowMeans(g)`. (Note that when a vector is subtracted from a matrix, R subtracts the vector from each column of the matrix.)

You should try using this adjusted data for all the parts of the analysis above. Comment on whether the adjustment seems to help in predicting recurrence, and how it affects the test results using Bonferroni adjustment or FDR estimation. Try to explain any differences you see. This may require looking at various plots or computing various informative quantities.

Part 5: logistic regression with quadratic terms

Finally, you should investigate whether using quadratic terms in the logistic regression model helps. (This would, for example, allow the dependence of recurrence probability on a covariate to be non-monotonic.) To include quadratic terms in a logistic regression, you can use a command such as

```
fit <- glm (recur ~ xx + I(xx^2), family=binomial)
```

where `xx` is some covariate. You can consider as covariates principal components as in Part 1, or principal components from data with per-patient adjustment as in Part 4.