Bayesian Statistics
Exercises 4.

1. Consider the following setting in the context of "multiple hypothesis tests". Let $i = 1, \ldots, n$ index individuals and $j = 1, \ldots, m$ index genes (or pixels in an image if you prefer). Assume we have measurements on each individual at each gene for "treatments" $k = 0, 1$. Let Y_{ijk} denote the measurement on individual i, gene j, treatment k, and let D_{ij} denote the difference between the measurements in the two treatments: $D_{ij} := Y_{ij0} - Y_{ij1}$. We will assume that D is sufficient for all our inferences, and that $D_{ij} \sim N(\beta_j, \sigma_j^2)$. Thus β_j is the "treatment effect" at gene j. For each gene j we wish to test the null $H_j : \beta_j = 0$ (that is, that there is no treatment effect). For simplicity you can assume that $\sigma_j = 1$ is known for all j. You can also assume that $n = 10$ and $m = 1,000$.

   Assume that the true effects β_j are independent, and identically distributed, with

   $$\beta_j \sim \pi_0 \delta_0 + (1 - \pi_0) N(0, \sigma_b^2). \tag{1}$$

   where δ_0 denotes a point mass on zero. That is, $\beta_j = 0$ with probability π_0, and $\beta_j \sim N(0, \sigma_b^2)$ with probability $1 - \pi_0$.

   i) Write an R function to simulate data D under this model, for user-specified π_0 and σ_b. The function should take π_0 and σ_b as input, and return a list, with elements D (a matrix) and β (a vector).

   ii) Write an R function to compute a p value p_j for each column of the data matrix D, testing $H_0 : \beta_j = 0$. This function should take as input the data matrix D and output a vector of p values. You can use any reasonable two-sided test, but state which test you use. Apply your R function to data simulated under a) $\pi_0 = 1$, b) $\pi_0 = 0.5, \sigma_b = 3$; c) $\pi_0 = 0, \sigma_b = 3$. Provide histograms of the p values in each case and comment on their distributions.

   iii) Write an R function to apply the Benjamini–Hochberg rule to control FDR at a user-specified level α. This function should input a vector of p values, and a level α, and output a vector of binary (0/1) indicators, $\gamma = (\gamma_1, \ldots, \gamma_m)$ say, where $\gamma_j = 1$ indicates that the rule would reject $H_j : \beta_j = 0$.

iv) Write an R function to compute the empirical False Discovery Rate (i.e. the number of false discoveries divided by the number of discoveries) for any given value for the vector β of true values of β, and the vector γ of reject decisions. That is, the function should return V/R in the notation of the notes. Remember to deal correctly with the special case of no discoveries, $R = 0$.

v) Perform a simulation study to estimate the actual FDR ($\mathrm{E}(V/R)$) achieved by the BH rule in the three cases a), b) and c) above. In each case perform the test procedure for different levels α, and plot the estimated $\mathrm{E}(V/R)$ as a function of α (say for $\alpha = (0.05, 0.1, \ldots, 0.5)$). Comment on the results. [NOTE: to estimate the actual FDR you have to estimate $\mathrm{E}(V/R)$ where the expectation is over datasets D. To do this you will want to do a simulation study where you simulate a large number of datasets D, not just one dataset!]

vi) Repeat the simulation study, but this time estimate the pFDR instead of the FDR, and plot this as a function of α.

2. The `qvalue` package in R implements Storey's approach to estimating FDR. To install this package use

```
source("http://bioconductor.org/biocLite.R")
biocLite("qvalue")
library("qvalue")
```

The package takes a vector of p values, and outputs a list which includes an estimate of π_0 (obtained using the p values near 1) and a vector of q values. Try, for example, for a vector of p values `p`,

```
res=qvalue(p)
res$pi0
res$qvalues
```

The q value for a particular observation is an estimate of the pFDR if you reject all things that are as or more significant than that observation. You can convert the vector of q values into a list of reject decisions at a given α level (the γ vector above) using, say,

```
compute.gamma=function(q,alpha){return(q<alpha)}
```

i) Repeat the simulation study above, using `qvalue` instead of the BH procedure. Produce plots of the FDR vs the α level for `qvalue` and compare them with those obtained for BH.

ii) Perform a simulation study (e.g. by modifying the simulations you have already performed), to see how accurately `qvalue` is able to estimate the proportion of nulls π_0. Try varying π_0 from 0 to 1 for at least 3 different values of σ_b, and in each case provide plots of the true π_0 vs the estimated π_0 from `qvalue`. Comment on the results.

3. The library `ashr` (for installation instructions, see the README at `www.github.com/stephens999/ash`) implements an empirical Bayes approach to the above problem. In contrast to the BH procedure, and `qvalue`, instead of working with the p values for each test, it works with a vector of estimates of the elements of β ($\hat{\beta}$), and their corresponding standard errors (s). In outline, the method first estimates the distribution, $g(\cdot;\pi)$, of β_j from the data, where π are hyper parameters to be estimated by maximum likelihood (this is why it is "Empirical Bayes"). Then, given the mle $\hat{\pi}$, it computes the posterior probability of H_j, $\Pr(H_j|\hat{\beta}_j, s_j, \hat{\pi})$, and other posterior quantities of interest (like the posterior mean for β_j). It returns a lot of things, but for this exercise all you need to know is that it returns a vector of q values (same interpretation as those from `qvalue`). If `betahat` is a vector of the estimates, and `s` is a vector of the standard errors, then you can apply `ash`, and obtain the q values, as follows:

```
res.ash = ash(betahat,s, method="fdr")
res.ash$qvalue
```

(a) Write a function that computes, from data D, a vector of estimates of β (`betahat`) and their corresponding (estimated) standard errors (`s`). Note that under the null, `betahat/s` will have a normal distribution, so `betahat^2/s^2` has a chi-squared distribution so you can compute a vector of p values using

```
zscore = betahat/s
pval = pchisq(zscore^2,df=1,lower.tail=F)
```

For one simulated data set plot a graph of the p values you get this way against the p values you used for the BH procedure to check that they are highly correlated.

(b) Repeat the simulation study from questions 1 and 2, showing estimated FDR vs α level, using the `ash` q-values. Compare results with BH and `qvalue`. Do all three methods provide conservative control of FDR?

(c) In addition to control of FDR, it is also of interest to compare methods in terms of *how many discoveries* they make (say when controlling the FDR at $\alpha = 0.05$). Take one of the simulation scenarios and compare the methods on how many discoveries they make at FDR=0.05.