STA 410/2102, Fall 2014 — Assignment #2

Due at the start of class on November 13. Please hand it in on 8 1/2 by 11 inch paper, stapled in the upper left, with no other packaging.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you should not leave any discussion with someone else with any written notes (either paper or electronic).

In this assignment, you will use R's `integrate` function for univariate numerical quadrature, applying it to maximum likelihood estimation for rounded data and to Bayesian inference for a model of bivariate normal data with known mean but unknown variance and correlation.

The `integrate` function takes as its first argument an R function defining what to integrate. This function must accept a *vector* of values, and return a vector of results of applying the function to those values. Many functions will naturally operate in this way, but if necessary, the function will need to loop over elements in its argument in order to compute the integrand at each element value. The second and third arguments to `integrate` are the low and high bounds for integration. To get the result of the integration, extract the element `value` from the result of `integrate`.

First problem. For this problem, we observe n i.i.d. values from a normal distribution with unknown mean, μ, and unknown standard deviation, σ, and wish to find maximum likelihood estimates for μ and σ. However, our data is rounded to d decimal places (eg, we might have data like 2.3, 1.2, 3.0, -1.7 if $d = 1$). We wish to find maximum likelihood estimates that account for this, by maximizing the probability of obtaining n data points that round to the data that is recorded. This probability is found by integrating the normal probability density function over the range of real values that round to what was recorded.

You should write an R function called `rnd_norm_mle1` that uses `nlm` to find the maximum likelihood estimates for μ and $\log(\sigma)$ (use the log of σ so that there is no constraint on valid values). The function should take as arguments a vector of rounded data values and the number of decimal places to which these values were rounded. Inside your `rnd_norm_mle1` function, the function you give to `nlm` should compute the log likelihood using `integrate`. (You should not try to compute the gradient or Hessian of the log likelihood.)

You should also write an R function called `rnd_norm_mle2` that does the same thing as `rnd_norm_mle1` except that rather than using `integrate`, it finds the integrals using R's `pnorm` function for computing the cumulative distribution function for a normal distribution. (You'll need to call `pnorm` twice, for each end of the integration range, and take the difference.)

You should test your functions on three datasets generated as follows:

```
set.seed(1); d1 <- 1; x1 <- round(rnorm(25,2.24,1),d1)
set.seed(2); d2 <- 1; x2 <- round(rnorm(50,2.24,0.1),d2)
set.seed(3); d3 <- 0; x3 <- round(rnorm(4000,2.24,0.8),d3)
```

Comment on how the maximum likelihood estimates of μ and σ compare to simply using the sample mean and sample standard deviation of the rounded data points. Also comment on the speed using `integrate` versus using `pnorm`.

Second problem. For this problem, we observe n pairs of values (to high precision, so rounding is not an issue), which we will denote by (X_{11}, X_{12}), (X_{21}, X_{22}), $\ldots$, (X_{n1}, X_{n2}). The n pairs are i.i.d. but the two values within each pair may be dependent, having a bivariate normal distribution with means of zero, with both elements having the same unknown standard deviation, σ, and with an unknown correlation, ρ, between the elements in the pair. The density function for this bivariate normal distribution for a pair (X_{i1}, X_{i2}) is

$$f(x_{i1}, x_{i2}) \;=\; \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp\Big(-\, (x_{i1}^2 + x_{i2}^2 - 2\rho x_{i1} x_{i2}) \,/\, (2(1-\rho^2)\sigma^2)\Big)$$

We wish to do Bayesian inference for this model, using a prior distribution for σ and ρ with the following density function over $(0, \infty) \times (-1, +1)$:

$$f(\sigma, \rho) \;=\; e^{-\sigma} \Big/ \left(4|\rho|^{1/2}\right)$$

You should write a function `bvn_likelihood` that takes as arguments a $2 \times n$ matrix of data values, `X`, and parameter values `sigma` and `rho`, and returns the likelihood for these parameter values given the data. You should also write a function `bvn_prior` that takes as arguments parameters values `sigma` and `rho`, and returns the prior density for these parameter values.

You should then write a function `bvn_normalize` to compute the normalizing constant for the posterior distribution given data `X` — that is, the integral of the product of the prior density and the likelihood over the whole parameter space. You should use R's `integrate` function to do this. Note that `integrate` will accept `Inf` (infinity) as an upper bound. It also will handle singularities in the integrand at the end-points of the interval, but not in the interior of the integral (so you may need to break up an integral into parts to handle such singularities). Since the normalizing constant is a double integral, you will need to use an inner call of `integrate` inside the function integrated by an outer call of `integrate`.

You should also write a function `bvn_posterior_rho` with arguments `X` and `rho` that computes the marginl posterior density (with correct normalization) of `rho` given data `X`. You should then write a function for plotting both the prior density and the marginal posterior density for ρ, over its range $(-1, +1)$, given a dataset `X`. Since these densities are singular at zero, you should put an upper limit of 5 on the density plotted (using an option of `ylim=c(0,5)` for `plot`).

You should try out your functions on two datasets `Xa` and `Xb` generated as follows:

```
set.seed(1)
n <- 40
x1 <- rnorm(n); x2 <- rnorm(n); z <- rnorm(n)
Xa <- cbind (x1, x2); Xb <- cbind (x1+0.5*z, x2+0.5*z)
```

Discuss how the posterior distributions you see for these datasets compare to the sample correlations and to the true values of the correlation.

For graduate students in STA 2012 (undergrads may do this for bonus marks): Investigate for what size of n the functions you wrote start to fail because of floating point overflow or underflow, and suggest a way of alleviating this problem.