Intro to Bayesian Data Analysis
Computer Activities

Getting Started (skip if you're already familiar with R, R Studio, ggplot2)

Be sure that you have R and RStudio loaded (or updated). I loaded them on my computer a few weeks ago, so I have **R version 3.3.0** and **RStudio version 0.99.902**. You should also install the ggplot2 package—at the command line: install.packages(”ggplot2”) .

I like to create R projects for each of the separate projects I work on—this way files for each project are in their own project folder on my computer. I urge you to do this too. For this class, I have a GWEDA project set up; you might want to have a Bayes (or pick another name) project set up on your computer.

R is an object oriented language—data and functions that operate on data are all just objects in your R directory. For now, you'll create a couple of data objects and a function object, and later you'll learn to import data files. Here's the code that I used to create the plot of the binomial likelihood function in this morning's presentation:

```
# access the ggplot2 library:

       library(ggplot2)

# create a vector of pi's

       pi.values = seq(0,1,length=100)
       pi.values    # type this at the command prompt to see the values

# create a function object for the binomial likelihood

       lik.fun = function (p) p^22*(1-p)^35

# create a data frame containing the pi's and the likelihood evaluations

       lik.df = data.frame(x=pi.values,y=lik.fun(pi.values))
       lik.df     # type this at the command prompt to see the data frame
```

```
# plot the likelihood using ggplot

       q = ggplot(lik.df,aes(x,y)) + geom_line() + ylab("")
       q + theme(axis.text.y=element_blank()) + xlab(expression(pi))
```

There's a bit to parse in these lines of code—please ask if you want additional help understanding anything. The R package ggplot2 is a great package for creating presentation- and publication-ready graphics, but there's a bit of a learning curve with it. The basic function is ggplot, and if you type ?ggplot at the command prompt, you'll see that the two basic arguments to ggplot are (1) data, which is the data frame that contains the variables that you want to plot and (2) mapping, which is essentially a list of variables (aesthetic mappings) that you want to plot. To get a line plot, you add *geom_line*(); for a scatterplot, you add *geom_point*(); etc. (The "geom" designation is for "geometric object") There is extensive help about the package on-line (when I want to find out how to do something, I usually go to StackExchange with my questions).

Lichen Demo—Posterior Distribution

You can use the following R code, and I suggest that you go through it line by line, to get an appreciation for what's going on with Bayes theorem.

```
#
# Demo for Lichen Example
#

library(ggplot2)

# generate 1,000,000 draws from a uniform prior distribution for pi:
     prior.pis = runif(1000000)

# with each value of pi, generate a binomial(57,pi) observation
     x.tilde = rbinom(1000000,57,prior.pis)

# if the binomial observation is 22, then keep the corresponding pi
    lichen = data.frame(post.pis = subset(prior.pis,x.tilde==22))

# set up values to plot the exact posterior
    lichen$pi.values=seq(0,1,length=length(lichen$post.pis))
    lichen$exact.post = dbeta(lichen$pi.values,23,36)

# plot histogram of posterior theta's and superimpose exact posterior
```

```
q = ggplot(lichen,aes(post.theta))
q = q + geom_histogram(aes(y=..density..),color="blue",bins=100)
q = q + xlab(expression(pi)) + xlim(c(0,1)) + ylab("")
q
q + geom_line(aes(x=pi.values,y=exact.post))
```

Conjugate Priors

Let $\mathcal{F}$ denote a family of prior distributions and $\mathcal{L}$ denote a family of Likelihood functions.

> $\mathcal{F}$ is said to be **conjugate** for $\mathcal{L}$ if any member of $\mathcal{F}$, when combined with any member of $\mathcal{L}$ using Bayes theorem, results in a posterior distribution that is a member of $\mathcal{F}$.

- The Beta family of priors is conjugate for the binomial family of likelihoods.
- The gamma family of priors is conjugate for the Poisson family of likelihoods.
- The Normal family of prior is conjugate for the Normal family of likelihoods in the case where σ^2 is assumed known.

Feel free to check out shiny demos created by Brian Reich, Associate Professor of Statistics at North Carolina State University:

- http://teaching.stat.ncsu.edu/shiny/bjreich/BetaBinom/
- http://teaching.stat.ncsu.edu/shiny/bjreich/PoissonGamma/
- http://teaching.stat.ncsu.edu/shiny/bjreich/NormalNormal

Posterior Summaries

We can also use R to obtain summaries of Beta (or other closed form) posterior distributions. In R, common probability distributions (e.g., Normal, Beta, binomial, Poisson, gamma, etc.) are built-in functions—actually, there are several functions for each probability distribution:

- **dbeta** — gives the functional form of the pdf (or pmf)
- **pbeta** — gives the cumulative distribution function (allows you to put in a quantile and get back a probability)

- **qbeta** — gives the inverse cdf (allows you to put in a probability and get back a quantile)
- **rbeta** — generates random draws from a beta distribution

The functions above are specific to the Beta distribution, but analogous functions exist for the other distributions (e.g., pnorm, qpois, rbinom).

For the lichen problem, if we want to obtain a 95% posterior interval for π, we can do so by finding the 0.025 and the 0.975 quantiles of the Beta(23,36) distribution. In R:

```
       qbeta(c(0.025,0.975),23,36)
```

You can type "?qbeta" at the command line in R to learn more about the function.

CO_2 Example

The data we consider are 101 measurements of the concentration of CO_2 in air trapped in bubbles at different depths in an ice core taken from Law Dome in East Antarctica. Use the following R script (and instructions in it) to move the data into R and then create a plot):

```
#
# Read in the CO2.csv file (if you have the file in your Bayes project folder):
#

       CO2 = read.csv("CO2.csv")

#
# Take a look at the data:
#

       ggplot(CO2.df,aes(x=depth,y=y)) + geom_point() + ylab("Ice Core Conc")
```

As the ice core depth increases, we are going back in time. We'd like to estimate *continuous* atmospheric CO_2 concentrations through time. To achieve this, consider that (1) we have to convert depth to time; (2) ice core CO_2 concentration doesn't equal atmospheric CO_2 concentration (in fact, ice core concentrations are a smoother version—i.e., they are less variable—than atmospheric concentrations); and (3) we only have 101 discrete measurements, and those measurements likely have serial correlation.

A number of fairly ad hoc methods have been used to "unsmooth" the ice core data to find an atmospheric CO_2 record that, when smoothed, is consistent with the original observations. Our ultimate goal will be to do this "unsmoothing" in a statistically rigorous way using a Bayesian hierarchical model. To that end, it's useful to start thinking about the steps we'll take along the way, and how we might translate those into a statistical model:

1. relate the observed concentrations, $\mathbf{y}$, to atmospheric concentrations
2. convert depths to times
3. relate atmospheric concentrations to times
4. deal with any residual serial correlation to model error variance