Week 7 Module

Normal Distribution

Recall the the pdf of the normal distribution can be written as:

$$p(y|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y-\mu)^2\right).$$

Now assume we are interested in modeling a continuous quantity, such as housing prices in the Seattle area, using the normal distribution as a sampling model.

- What are the parameters that we want to estimate in this setting?

- What parameters require prior distributions?

- What are reasonable prior distributions for these parameters?

Estimating the mean and standard deviation of a normal distribution

The goal will be to estimate the posterior distribution:

$$p(\mu,\sigma^2|\{y_1,\ldots,y_n\}) = \frac{p(\{y_1,\ldots,y_n\}|\mu,\sigma^2)p(\mu,\sigma^2)}{\int\int p(\{y_1,\ldots,y_n\}|\mu,\sigma^2)p(\mu,\sigma^2)d\mu d\sigma^2}$$

- What is $p(\{y_1,\ldots,y_n\}|\mu,\sigma^2)$?

$$\begin{aligned} p(\{y_1,\ldots,y_n\}|\mu,\sigma^2) &= \prod_i p(y_i|\mu,\sigma^2) \\ &= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i-\mu)^2\right) \\ &= \prod_i \frac{n}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\sum_i (y_i-\mu)^2\right) \end{aligned}$$

- How do we write $p(\mu, \sigma^2)$?: this is a joint prior distribution,

- What is $\int\int p(\{y_1, \ldots, y_n\}|\mu, \sigma^2)p(\mu, \sigma^2)d\mu d\sigma^2$?

Unfortunately

Special case with σ^2 known

There is a special case that assumes σ^2 is known which permits a conjugate prior for μ. It can be shown (DBDA p.452) that $p(\mu) \sim Normal()$ is a conjugate prior in this situation. However, outside of textbook examples, we are never given σ^2.

Joint Priors

Commonly independent priors are placed on μ and σ^2 such that $p(\mu, \sigma^2) = p(\mu) \times p(\sigma^2)$.

- $p(\mu) \sim N(M, S^2)$: A normal prior with mean M and standard deviation S.
  - what impact do you suppose S has on the posterior?
  - what impact do you suppose M has on the posterior?
- $p(\sigma^2) \sim Unif(0, C)$ where C is a large constant OR $p(\sigma^2) \sim InvGamma(Sh, R)$ or equivalently $p(\tau) \sim Gamma(Sh, R)$ where $tau = 1/\sigma^2$ is known as the precision parameter. JAGS parameterizes this model in terms of precision.
  - The shape parameter Sh and the rate parameter R are not as intuitive as the parameters in the prior for μ, but we will spend some time visualizing these value.

JAGS

There are five steps for fitting Bayesian models in JAGS:

1. Load data and store as list object.
2. Specify the model as a text variable.
3. Specify Starting Point
4. Generate MCMC chains.
5. Examine and summarize results.

1. Load data First we will simulate data to use for this example, this is useful as we know the actual μ and σ^2 values.

```
set.seed(02242023)
num.obs <- 50
true.mu <- 0
true.sigmasq <- 1

y <- rnorm(num.obs, mean = true.mu, sd = sqrt(true.sigmasq))
```

We specify our priors at this point too.

```
M <- 0
S <- 100
C <- 100000
```

This data will be stored as a list for use in JAGS.

```
dataList = list(y = y, Ntotal = num.obs, M = M, S = S, C = C)
```

2. Specify Model One thing to note is that JAGS uses the precision as the second term in a normal density function.

```
modelString = "model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dnorm(mu, 1/sigma^2) # sampling model
  }
  mu ~ dnorm(M,1/S^2)
  sigma ~ dunif(0,C)
} "
writeLines( modelString, con='NORMmodel.txt')
```

3. Specify Starting Point In complicated models, using starting points can be very important. Consider a random starting point

```
initsList <- function(){
  # function for initializing starting place of theta
  # RETURNS: list with random start point for theta
  return(list(mu = rnorm(1, mean = 0, sd = 100), sigma = runif(1,0,1000)))
}
```

4. Generate MCMC chains Now, finally we can start running the MCMC chains. First the chains are initialized and a burnin period is created.

```
library(rjags)
library(runjags)
```

```
jagsModel <- jags.model( file = "NORMmodel.txt", data = dataList,
                         inits =initsList, n.chains = 2, n.adapt = 100)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 50
##    Unobserved stochastic nodes: 2
##    Total graph size: 63
##
## Initializing model
```

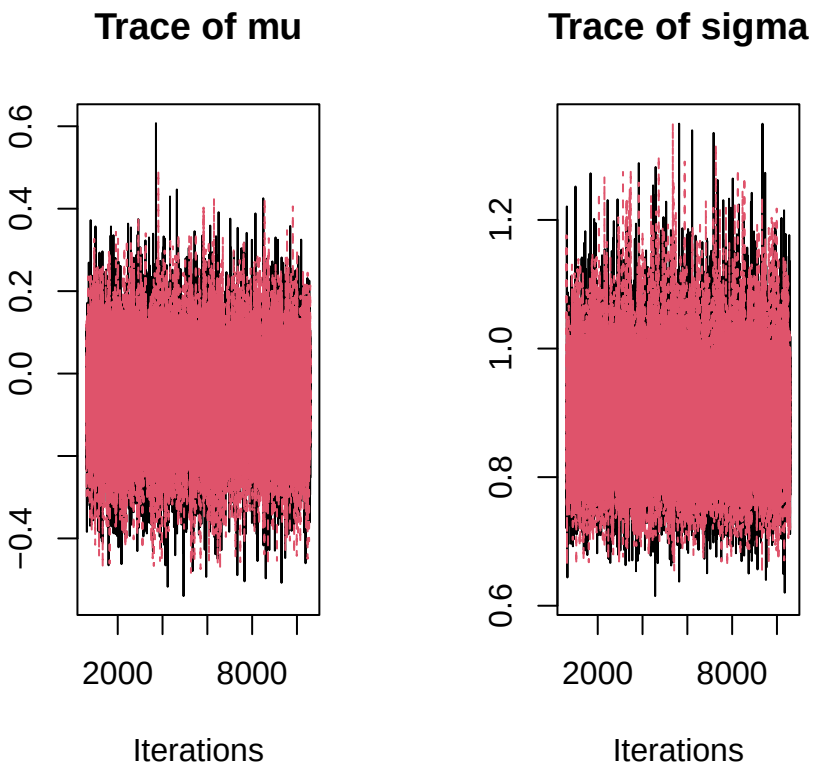
```
update(jagsModel, n.iter = 500)
```

Now the chains are run for a longer period of time - think of the traveling politician.

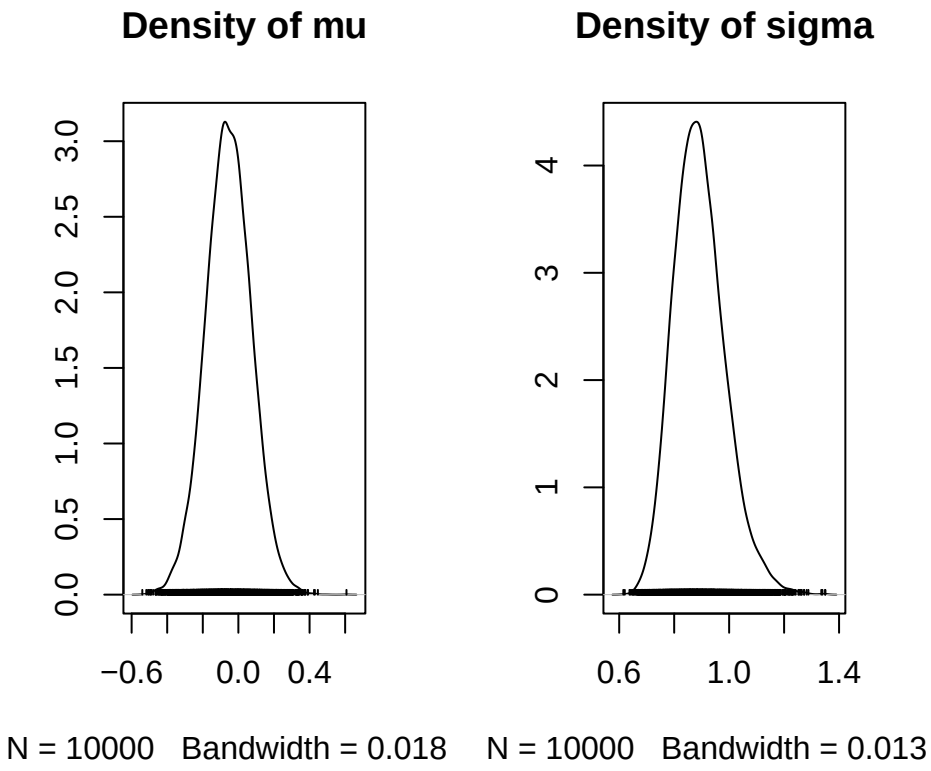
```
num.mcmc <- 10000
codaSamples <- coda.samples( jagsModel, variable.names = c('mu', 'sigma'), n.iter = num.mcmc)
```

5. Evaluate Model Now we can use the coda package to visualize the results from the MCMC. Note codaSamples is a list that contains an entry for each seperate chain. Within the entry there is a column for each variable μ and σ.

```
par(mfcol=c(1,2))
traceplot(codaSamples)
```



```
par(mfcol=c(1,2))
densplot(codaSamples)
```



coda also contains the ability to summarize the **marginal distributions of each parameter**, using intervals.

```
HPDinterval(codaSamples)
```

```
## [[1]]
##            lower     upper
## mu    -0.3122895 0.1904975
## sigma  0.7190644 1.0790985
## attr(,"Probability")
## [1] 0.95
##
## [[2]]
##            lower     upper
## mu    -0.3082108 0.1927643
## sigma  0.7226309 1.0817487
## attr(,"Probability")
## [1] 0.95
```

```
summary(codaSamples)
```

```
##
## Iterations = 601:10600
## Thinning interval = 1
```

```
## Number of chains = 2
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD  Naive SE Time-series SE
## mu    -0.05498 0.12768 0.0009028      0.0009150
## sigma  0.89236 0.09308 0.0006582      0.0007061
##
## 2. Quantiles for each variable:
##
##          2.5%     25%      50%     75%  97.5%
## mu    -0.3070 -0.1400 -0.05629 0.03011 0.1969
## sigma  0.7315  0.8269  0.88543 0.94851 1.0995
```

Summarizing a distribution with posterior samples

Now we have a collection of posterior samples to represent the joint posterior distribution $p(\mu, \sigma^2|\{y_1, \ldots, y_n\})$ as well as the marginal posterior distributions $p(\mu|\{y_1, \ldots, y_n\})$ and $p(\sigma^2|\{y_1, \ldots, y_n\})$.

- How should we summarize these results?:

- Suppose that a collaborator wanted to know whether the mean of y is greater than -.2?

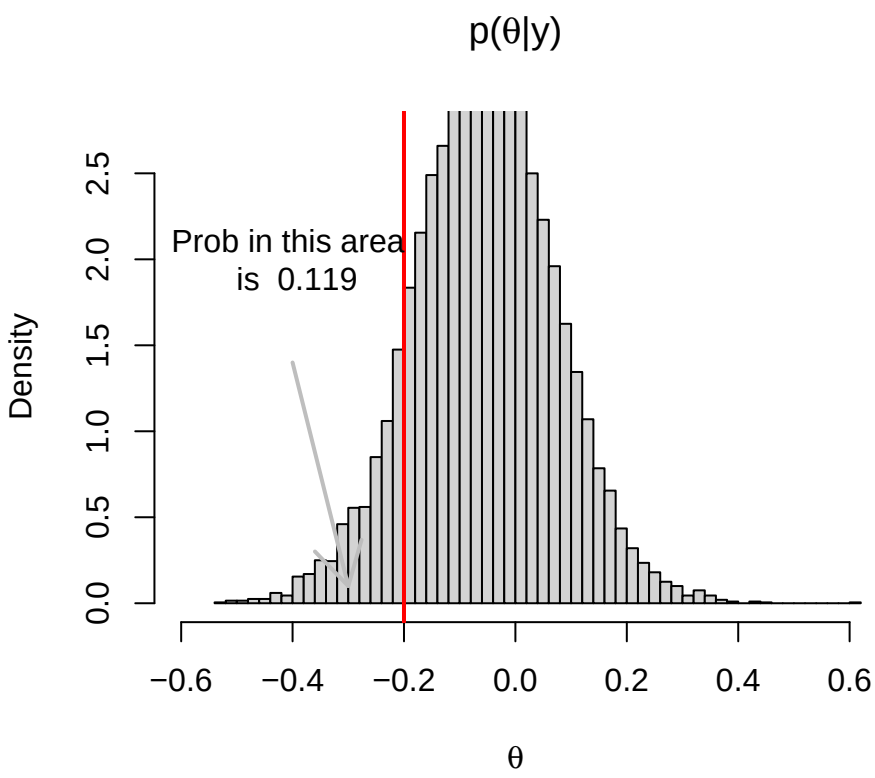
  -How would we answer this from a classical standpoint? We might use null hypothesis testing and a p-value to summarize this.

```
t.test(y, mu = -.2, alternative = 'greater')
```

```
##
##  One Sample t-test
##
## data:  y
## t = 1.1788, df = 49, p-value = 0.1221
## alternative hypothesis: true mean is greater than -0.2
## 95 percent confidence interval:
##  -0.261134       Inf
## sample estimates:
##   mean of x
## -0.05522207
```

- The interpretation here is that we'd reject the null hypothesis that the true mean is -0.2 . In other words it is unlikely given the data that the true mean is -0.2, with a **one-sided test**.
- We could also use a confidence interval, the 95% one-sided confidence interval for μ would be (-0.261134, ∞). What is the interpretation here? It is *not* the probability that μ is in this interval is 95%. Rather the interpretation has to do with coverage and long run frequencies. If a statistician creates 100 confidence intervals, 95 are expected to contain the true value.
- As a Bayesian, we can actually construct an interval that has probabilistic interpretation. For instance there is a 95% probability that μ is in the following interval: (-0.3122895, 0.1904975).
- Furthermore, we can also use our posterior samples to calculate $Pr(\mu > -0.2) = 0.8806$.

```
par(mfcol=c(1,1))
prob.greater <- mean(codaSamples[[1]][,'mu'] > -0.2)
hist(codaSamples[[1]][,'mu'], main=expression(paste('p(',theta,'|y)')), prob=T,
     xlab=expression(theta),breaks='FD',xlim = c(-.6,.6), ylim=c(0,2.75))
abline(v=-0.2,lwd=2, col='red')
arrows(x0 = -.4, y0 = 1.4 , x1 = -0.3, y1 = 0.1 , col='gray', lwd = 2)
text(x=-.4, y=2, paste('Prob in this area \n is ', round(1-prob.greater,3)))
```


p(θ|y)
Prob in this area
is 0.119
Density
0.0
0.5
1.0
1.5
2.0
2.5
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
θ