Lab 3

Harry Bendekgey and Peter Brody-Moore
Math 153: Bayesian Statistics

March 8, 2018

A Normal Likelihood (a)

We are interested in the case where our data is assumed to be coming from a normal distribution parametrized by its mean μ and its variance σ^2. We start with the case where σ^2 is known, and thus $\theta = \mu$.
We have as the likelihood

$$L(x|\mu) = \prod \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2} \propto \exp\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\}.$$

Defining $s^2 = \frac{1}{n}\sum(x_i - \bar{x})^2$, show

$$\sum(x_i-\mu)^2 = ns^2 + n(\bar{x}-\mu)^2.$$

$\sum(x_i-\mu)^2 = \sum([(x_i-\bar{x}) - (\mu-\bar{x})]^2 =$

$\sum((x_i-\bar{x})^2) + \sum((\bar{x}-\mu)^2) - 2\sum((x_i-\bar{x})(\mu-\bar{x})) =$

$\sum((x_i-\bar{x})(\mu-\bar{x})) = (\mu-\bar{x})(\sum(x_i) - n\bar{x}) = (\mu-\bar{x})(n\bar{x}-n\bar{x}) = 0$

So third term cancels. This means:

$\sum(x_i-\mu)^2 = ns^2 + n(\bar{x}-\mu)^2.$

A consequence of this is that

$$p(X|\mu) \propto \exp\{-\frac{n}{2\sigma^2}(\bar{x}-\mu)^2\} \propto \mathcal{N}(\mu, \frac{\sigma^2}{n})$$

A Normal Prior (b)

As we noticed with the binomial pmf, if we think about the likelihood being a function of p instead of x, we notice that it looks a lot like a Beta density. Thus, using a Beta prior, we got a Beta posterior out (conjugacy!). If we view the likelihood as a function of μ, it looks like a normal pdf as well. Let's see

what happens if we use a normal prior.
In particular, we may try

$$f(\mu) = \mathcal{N}(\mu_0, \sigma_0^2)$$

We can now write down the posterior $f(\mu|x) \propto p(x|\mu)f(\mu)$ as usual, and computing the posterior, we get:

$$f(\mu|x) \propto \exp\{-\frac{1}{2\sigma_n^2}(\mu - \mu_n)^2\}$$

with

$$\sigma_n^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}} \quad \text{and} \quad \mu_n = \sigma_n^2(\frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2}).$$

The formula for σ_n^2 is really unfortunate. It's *almost* pretty. To fix this, Bayesian's prefer not to talk about the variance, where large numbers correspond to large uncertainty, but instead to precision, where large numbers correspond to less uncertainty.
Define $\lambda_i = \frac{1}{\sigma_i^2}$. The posterior becomes $f(\mu|x) = \mathcal{N}(\mu_n, \lambda_n)$ where

$$\lambda_n = \lambda_0 + n\lambda \quad \text{and} \quad \mu_n = \frac{\bar{x}n\lambda + \mu_0\lambda_0}{\lambda_n}.$$

We can even simplify this more. Define $\lambda_0 = \kappa_0\lambda = \frac{\kappa_0}{\sigma^2}$. Then we get:

$$\lambda_n = \lambda(n + \kappa_0) \quad \text{and} \quad \mu_n = \frac{\bar{x}n + \mu_0\kappa_0}{n + \kappa_0}.$$

Note that in the equation for μ_n, the λ cancels.
Recall that the variance of $\bar{x}$ is $\frac{\sigma^2}{n}$, and therefore its precision is $n\lambda$; that is to say, the precision of the mean is the sum of the precisions of the data points. Thus we can see our posterior λ_n is just the precision of having seen $n + \kappa_0$ data points, and we can therefore consider our prior to be one in which κ_0 data points have been seen.

Further recall the equation for the Beta poserior mean:

$$E(p|X = x) = \frac{x + \mu}{\psi + n} = \frac{n}{\psi + n}\frac{x}{n} + \frac{\psi}{\psi + n}\mu = \lambda\hat{p} + (1 - \lambda)\mu$$

Here, we can see the posterior mean is a weighted average of the prior mean and the mean of the data. The weight of each of those two values is decided by the comparative sample size (acknowledging that the prior doesn't have a real sample size at all, but we can use this equation to think of its weight as its sample size) so as n gets large, the data gets weighed more and more against the prior, which is what we want to see.

We want to find a similar explanation for μ in the normal posterior. Let's take a look:

$$\mu_n = \frac{\bar{x}n + \mu_0\kappa_0}{n + \kappa_0} = \frac{n}{n + \kappa_0}\bar{x} + \frac{\kappa_0}{n + \kappa_0}\mu_0$$

This is exactly what we want to see. The posterior mean is a weighted average of the prior mean and the experimental mean, weighed according to the relative sample sizes. We now have great interpretations of our inputs μ_0 and σ_0^2: μ_0 is the "prior mean" and σ_0^2 is the variance of the data divided by the "prior sample size".

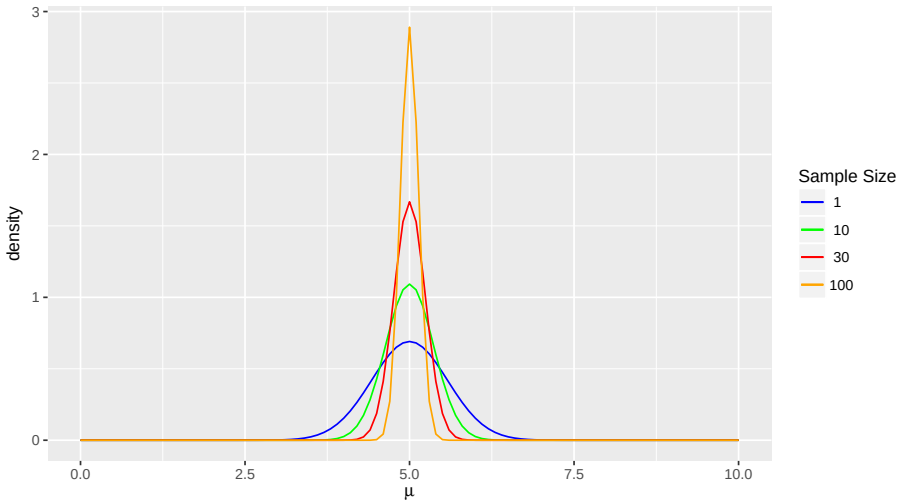
A Normal Posterior (c)

Let's see how this works in action:

```
post_dist <- function(x.bar, n, mu_0, kappa_0, sigma2) {
  w <- n / (n + kappa_0)
  lambda_n <- (n + kappa_0)/sigma2
  return(function(x) dnorm(x,w*x.bar+(1-w)*mu_0,sqrt(1/lambda_n)))
}

graph_posts <- function(x.bar, mu_0=5, kappa_0 = 5, sigma2 = 2) {
  plot <- ggplot(data.frame(x = c(0, 10)), aes(x)) +
    stat_function(fun = post_dist(x.bar,1,mu_0,kappa_0,sigma2),
                  aes(col=" 1")) +
    stat_function(fun = post_dist(x.bar,10,mu_0,kappa_0,sigma2),
                  aes(col=" 10")) +
    stat_function(fun = post_dist(x.bar,30,mu_0,kappa_0,sigma2),
                  aes(col=" 30")) +
    stat_function(fun = post_dist(x.bar,100,mu_0,kappa_0,sigma2),
                  aes(col="100")) +
    scale_color_manual("Sample Size",
                       values=c("blue","green","red","orange")) +
    labs(x = expression(mu), y = "density")
  return (plot)
}
```

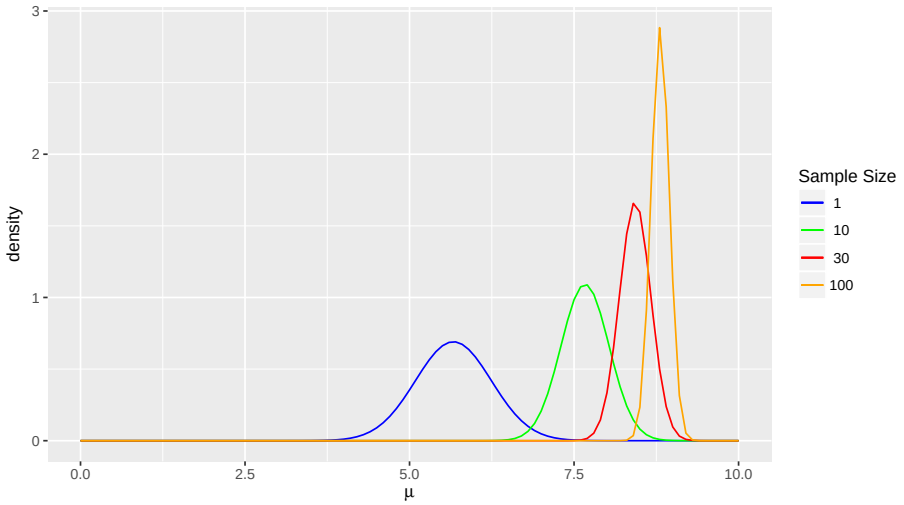
We are interested in the evolution of our posterior density as we see more data. We begin by choosing a smart prior, in this case $\mu_0 = 5$ with prior sample size 5. Then we fix $\bar{x} = 5$ see how our posterior density changes as we see more and more data.

```
graph_posts(5)
```



As expected, over time we get more sure of ourselves, and our posterior gets tighter. But what happens when we have a bad choice of prior? Here we will keep the same prior but fix $\bar{x} = \mu = 9$. We see what happens to our posterior over time:

```
graph_posts(9)
```



Just like in the Beta distribution, the prior becomes overrun by data and the posterio asymptotically approaches symmetry around $\mu_n = 9$. The distributions also get tighter over time, as our precision increases every time we see new data.

A Gamma Prior (d,e)

While the math is nice here, I have yet to personally work with data for which μ is unknown and σ^2 (or equivalently λ) is known. Let's deal with the case where both are unknown. Writing the likelihood and seeing if anything looks familiar, we have

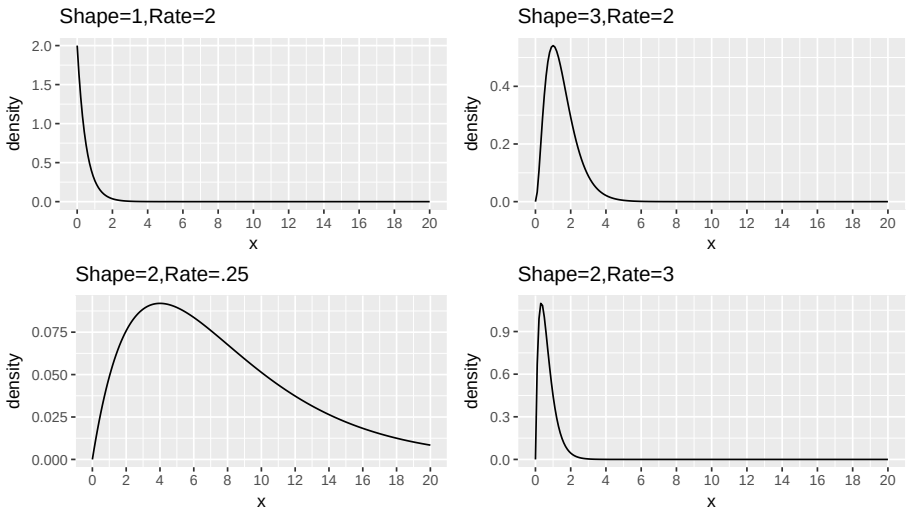
$$p(x|\mu,\lambda) \propto \lambda^{n/2} \exp\{-\frac{\lambda}{2}[n(\mu-\bar{x})^2 + ns^2]\}$$

As a function of μ only, this looks again like the kernel of a normal distribution. As a function of λ only, recall the gamma density:

$$f(x|\alpha,\beta) \propto x^{\alpha-1}e^{-\beta x}\mathbf{1}(x>0)$$

The gamma distribution takes in two parameters: a shape and a rate. The rate can also be replace by the scale, which is equivalent to the inverse of the rate. In this section, we look at the plots of four different gamma distributions. In the first two, we look at what happens when we keep the rate constant and increase the shape. In the second two, we keep the shape constant and increase the rate.

```
genGamma<- function(shape,rate){
  x <- seq(0,20,length=200)
  y <- dgamma(x,shape=shape,rate=rate)
  data <- data.frame(x,y)
  return(ggplot(aes(x,y),data=data) + geom_line() +
    ylab('density'))
}

ggarrange(genGamma(1,2) + ggtitle("Shape=1,Rate=2") +
          scale_x_continuous(breaks = seq(0, 20, 2)),
          genGamma(3,2) + ggtitle("Shape=3,Rate=2") +
          scale_x_continuous(breaks = seq(0, 20, 2)),
          genGamma(2,.25) + ggtitle("Shape=2,Rate=.25") +
          scale_x_continuous(breaks = seq(0, 20, 2)),
          genGamma(2,3) + ggtitle("Shape=2,Rate=3") +
          scale_x_continuous(breaks = seq(0, 20, 2)),
          nrow=2,ncol=2)
```



The gamma prior with shape=1 and rate=2 places a high probability on a low value, with virtually zero probability of the value being greater than 2. When we increase the shape to 3 and keep the rate at 2, the density shifts right. It has a mode of 1, with a medium chance of being between 0-1 and 1-2, a low chance between 2-4, and near zero chance of being greater than 4.
Looking at the third graph with shape 2 and rate .25, there is a high probability that the value equals 4, with the density relative spread out between 0 and 20. When you increase the rate to 3 while keeping the shape constant, the density gets much tighter. It centers at 1/3, with a medium probability of being between 1 and 1/3 or 1 and 2, and almost no probability of being greater than 2.

Next, we want to find a gamma prior that encodes the information that we think $\sigma^2 = 4$, and are 90 percent sure that $\sigma^2 \geq 2$. However, bayesians prefer to deal with precision (the inverse of variance) over variance. Therefore, we will translate this into precision language. Our best guess for λ^* is 1/4, and we are 90 percent sure that $\lambda^* \leq 1/2$. To do this, we set $\frac{\alpha-1}{\beta}$ (gamma mode) = 1/4 to get β = 4*(α-1). Furthermore, we know that pgamma(.5,α, β) $\geq$ .9. However, we can't solve this analytically so we have to write a grid search function. Our function from lab two will be helpful.

```
find_alpha <- function(bound, confidence_level, mode, maxCheck) {
  a <- 1
  while(a < maxCheck) {
    if (pgamma(bound,a,(a-1)/mode) >= confidence_level &&
          abs(confidence_level-pgamma(bound,a+.1,(a+.1-1)/mode))>
          abs(confidence_level-pgamma(bound,a,(a-1)/mode))) {
          # If the next is not closer to the certainty
            return(c(a,(a-1)/mode))
```

```
      } else {
        a <- a + .1
      }
  }
  return(0)
}
```

```
find_alpha(.5,.9,.25,50)

## [1]  5 16
```

This function computes various α and β values that fit $\beta = 4*(\alpha\text{-}1)$. Using a step of .1, we choose the α and β that get us closest to our confidence level. Using our function we get $\Gamma(5, 16)$ as our prior. This gives us a mode of .25 and makes us 90 percent sure that λ^* is no smaller than 2.

A Normal-Gamma Posterior (f,g)

The conjugate prior for (μ, λ) ends up being normal and gamma priors, respectively (though the normal prior is a conditional density, conditioned on the value λ). So we need to specify the 4-tuple $(\mu_0, \kappa_0, \alpha_0, \beta_0)$ to specify our prior. We write

$$(\mu, \lambda) \sim \mathcal{NG}(\mu_0, \kappa_0, \alpha_0, \beta_0)$$

The posterior distribution is

$$p(\mu, \lambda|x) = \mathcal{NG}(\frac{\kappa_0\mu_0 + n\bar{x}}{\kappa_0 + n}, \kappa_0 + n, \alpha_0 + n/2, \beta_0 + \frac{n}{2}s^2 + \frac{\kappa_0 n(\bar{x} - \mu_0)^2}{2(\kappa_0 + n)})$$

Let's stop doing calculus now. I can generate from one of these Normal-Gamma distributions via a two-step process.

1. Generate λ^* from a $\Gamma(\alpha_n, \beta_n)$.
2. Then generate μ^* from a $\mathcal{N}(\mu_n, \kappa_n\lambda^*)$

I have written a function that takes in a data set as well as the vector of hyperparameters, and returns the posterior mean for μ, a 95 percent credible interval for μ, and a 95 percent prediction interval for a new observation $\tilde{x}$

```
simulate_ng_posterior <- function(data, params) {
  # calculate posterior hyperparameters
  n <- length(data)
  xbar <- mean(data)
  s2 <- var(data)
  post_params <- c()
```

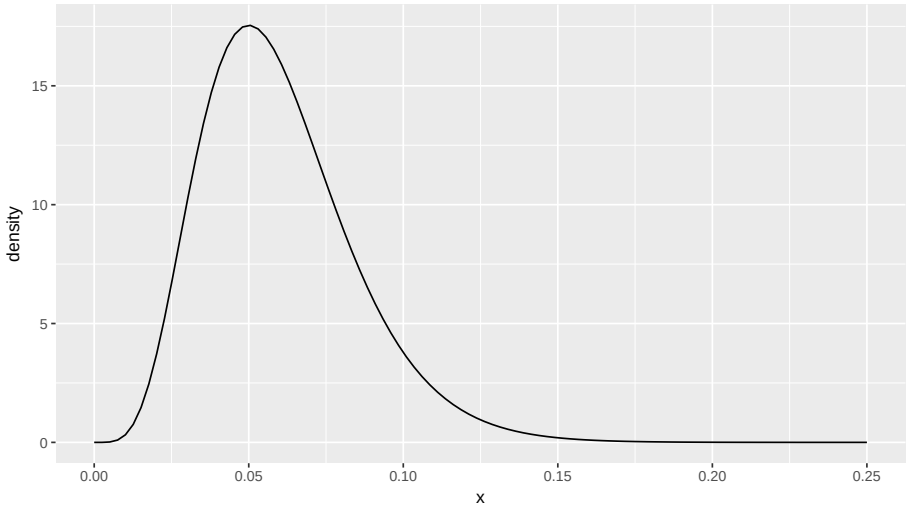
```
  post_params[1] <- (params[1]*params[2]+n*xbar)/(params[2]+n) #mu
  post_params[2] <- params[2] + n #kappa
  post_params[3] <- params[3] + n/2 #alpha
  post_params[4] <- params[4] + n/2 * s2 +
        (params[2]*n*(xbar - params[1])^2)/(2*(params[2]+n)) #beta

  # generate a lot of values for lambda, mu, and new x.
  lambdas <-rgamma(10000,shape=post_params[3],rate=post_params[4])
  mus <- c()
  xs <- c()
  for (i in 1:10000) {
    mus[i] <- rnorm(1,mean=post_params[1],
                    sd=1/sqrt(post_params[2]*lambdas[i]))
    xs[i] <- rnorm(1,mean=mus[i],
                   sd=1/sqrt(lambdas[i]))
  }

  # return posterior mean, credible intrvl, and prediction intrvl
  return(list("mean"= mean(mus),
              "CI" = quantile(mus,c(0.025,0.975)),
              "PI" = quantile(xs,c(0.025,0.975))))
}
```

Let's see this in action. Given a data set about the gas mileage of cars in 1974, we want to create a credible interval for the average gas mileage and a prediction interval for a single model's gas mileage. We remove one car model from the dataset so we can see if our prediction interval captures that value:

```
data('mtcars')
deleted_row <- sample(1:nrow(mtcars), 1)
deleted_mpg <- mtcars[c(deleted_row),1]
mtcars <- mtcars[-c(deleted_row)]
```

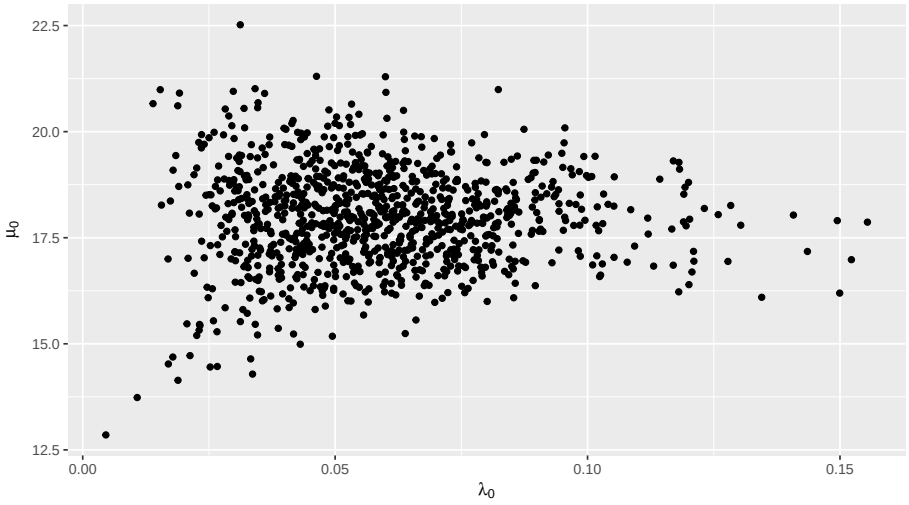
First we choose two hyperparameters α_0 and β_0 to generate λ^*. We estimate a fairly large spread (low precision), so our best guess is $\lambda^* = 1/20$. Furthermore, we are pretty confident that it will be between $1/30$ and $1/10$. This corresponds to a best guess for σ^2 of 20 and high confidence that σ^2 is between 10 and 30. To accomplish this, we set $\alpha_0 = 6$ and $\beta_0 = 100$, which gives a mode of .05, and an 81 percent change of being between $1/10$ and $1/30$. This gamma prior for λ^* can be visualized below:

```
x <- seq(0,.25,length=100)
y <- dgamma(x,6,100)
data <- data.frame(x,y)
ggplot(aes(x,y),data=data) + geom_line() + ylab('density')
```



Next we need to generate hyperparamters for the normal prior. Our best guess for the mean mpg is 18. So, we set $\mu_0 = 18$. Furthermore, we set $\kappa_0 = 15$, which signifies 15 pseudo-observations given by the prior. We can visualize this joint prior by sampling from both distributions and creating a scatter plot:

```
lambda0s <-rgamma(1000,shape=6,rate=100); mu0s <- c()
for (i in 1:1000) {
  mu0s[i] <- rnorm(1,mean=18, sd=1/sqrt(15*lambda0s[i]))
}
data <- data.frame(lambda0s,mu0s)
ggplot(aes(lambda0s,mu0s),data=data) + geom_point() +
  labs(x=expression(lambda[0]), y=expression(mu[0]))
```



Now we plug these 4 hyperparameters into the function above to generate a posterior mean for μ, a credible interval for μ, and prediction interval for x.

```
simulate_ng_posterior(mtcars$mpg, c(18,15,21,1))
```

```
## $mean
## [1] 19.4
##
## $CI
##  2.5% 97.5%
##  18.2  20.6
##
## $PI
##  2.5% 97.5%
##  11.1  27.6
```

We slightly underguessed μ but we were pretty close. The gas mileage of the deleted car was:

```
deleted_mpg
```

```
## [1] 14.3
```

Our prediction interval covers the deleted observation. Nice!