Day 5: Intro to Linear Regression

Stephen R. Proulx

1/10/2021

Today's objectives:

- Learn the notation for describing Bayesian statistical models
- Writing likelihoods for multiple observations of data
- Simulating from a prior
- grid approximation with 2 parameters
- Calculating on the log scale

Notation

1. Start with your likelihood. Usually this is a single line, but in rare cases (like if there are multiple types of data), it could be more. You can tell a line is part of the likelihood if it has data and parameters in it.

In RMarkdown we can use LaTeX to typeset equations. A nice intro to LaTeX is here: https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes .

I'll recreate the description on pate 77. We start typesetting a LaTeX equation with the "$$" symbol. The symbol "~" can be generated with the `\sim` command:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

Math symbols in greek are generally produced with `lettername`.

2. Next we put in any "transformations", which are sometimes called "link" functions. You can tell that a line is one of the transformations because it only involves parameters (but including hyper-parameters), not data, and because it does not involve a probability density (or the symbol ~)

$$\mu_i = \beta x_i$$

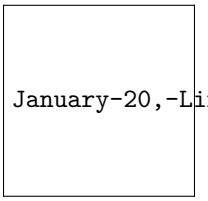
3. And then then all the priors. Each true parameter has a prior. How do you know it is a "true" paremter? Because it has a prior. The priors are all probability statements, so they have the symbol ~, and they no not inovled the data.

$$\beta \sim \text{Normal}(0, 10) \sigma \sim \text{Exponential}(1) x_i \sim \text{Normal}(0, 1)$$

Height data, mean and variance

Here we will go through the example in the book that fits human height data using a normal likelihood function. Because normal distributions have both a mean and standard deviation, this is a two parameter model, so a grid approximation will really be a grid this time.

```
data("Howell1")
d<-Howell1
d2<- d%>% filter(age>=18)
ggplot(data=d2, aes(x=height)) + geom_histogram(binwidth = 2.5)
```



The model described in the book which we will fit: $$ y_i $\sim$ Normal(μ, σ)\$\mu$ $\sim$ Normal(178, 20)\$\sigma$ $\sim$ Uniform(0, 50)

$$

A first important question is, what does the likelihood function really mean, and why is it a good choice for this model? When we write

$$y_i \sim \text{Normal}(\mu, \sigma)$$

what is actually meant is:

$$Pr(\text{data}|\text{parmeters}) = \prod PDF(\text{Normal}(y_i, \mu, \sigma))$$

This means that to get the likelihood of a dataset that invovles multiple observations (which we label $_i$), we are multiplying together the likelihood of each individual datapoint. This is because we are assuming that each height is independent of each other, and the joint probability of independent events is the product of their probabilities.

An additional important point is that we can do our work on the log scale, which converts products into sums, and then convert back to the natural scale. This is largely a computational trick done in the software behind the scenes.

Prior predictive simulation of height data

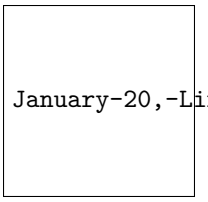
It can be very useful to first see what sort of data, in broad terms, your priors will produce. If they are producing absurd values, you have good prior knowledge to exclude those parameters.

Here we apply the prior by drawing values for μ and σ and then drawing a normally distributed height from that.

```
prior_sim <- tibble( mu=rnorm(1e4,mean=178,sd=20), sigma=runif(1e4,min=0,max=50)) %>%
  mutate(y=rnorm(n(),mean=mu,sd=sigma))
```

Let's visualize it. It will be an over-dispersed normal, because we have variance in the parameters and the normal sampling variability.

```
ggplot(data=prior_sim, aes(x=y))+ geom_density()
```



Grid approximation of the posterior

Now we can do a grid approximation to generate the posterior, and in this case we actually have two parameters so we atually have a grid.

In terms of coding, the trick here is to use `expand` to produce all combinations of two columns.

```
#code to grid out the posterior

n <- 200 # how many steps to use in the grid.

d_grid <-
  tibble(mu    = seq(from = 150, to = 160, length.out = n),
         sigma = seq(from = 4,   to = 9,   length.out = n)) %>%
  # expand can be used to combine all the elements from two rows
  expand(mu, sigma)
```

Have a quick look at the grid to see how it worked.

```
view(d_grid)
```

We need to write a special function to calculate our likelihood. This function takes as input the values of μ and σ that we are considereing. It also needs to use the data, in our case still stored in the dataframe `d2`.

We code this by summing up the log likelihoods

```
height_lik_f <- function(mu_input,sigma_input){
  sum(dnorm(
  d2$height ,
  mean=mu_input,
  sd=sigma_input,
  log=TRUE ))
}
```

And we convert this to a "vectorized" function so we can use it in `dplyr` functions.

```
height_lik_f_vec <- Vectorize(height_lik_f)
```

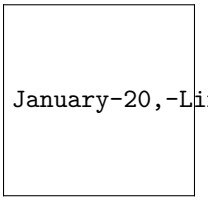
Repeat your mantra: likelihood * prior and then normalize! This time we do it on the log scale and then convert back to the natural scale.

```
posterior_table <- d_grid %>%
  mutate(log_likelihood=height_lik_f_vec(mu,sigma),
         log_prior_mu = dnorm(mu,    mean = 178, sd  = 20, log = T),
         log_prior_sigma = dunif(sigma, min  = 0,   max = 50, log = T),
         raw_log_posterior = log_likelihood + log_prior_mu + log_prior_sigma,
         log_posterior = raw_log_posterior - max(raw_log_posterior) ,
         raw_posterior = exp(log_posterior),
         posterior = raw_posterior/sum(raw_posterior))
```

Exploring the posterior probability

We can view the posterior probability, which has parameters in two dimensions, using a contour plot. This figure uses the calculated probabilities, not samples from the posterior distribuiton.

```
contour_xyz(posterior_table$mu, posterior_table$sigma , posterior_table$posterior)
```



We sample from the posterior in exactly the same way as before. Each row of our dataframe contains values for both μ and σ.

```
samples_height_model <- sample_n(posterior_table, weight =posterior, size=1e4, replace=TRUE) %>%
  select(mu,sigma)
```

We can view a summary with the `precis` command. This gives a table with means and quantiles, and also a chunky little histogram.

```
precis(samples_height_model)
```

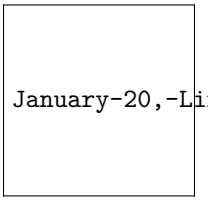
```
##             mean        sd       5.5%      94.5%   histogram
## mu    154.604508 0.4137554 153.969849 155.276382
## sigma   7.768226 0.2971587   7.316583   8.271357
```

Now that we have samples, we can view them in a number of ways.

We can look at the scatter plot of the points themselves. This is a fine thing to glance at.

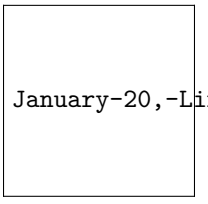
```
ggplot(data=samples_height_model, aes(x=mu,y=sigma)) +
  geom_point(size = .9, alpha = 0.1) +
  scale_x_continuous(limits = c(153,156.5),breaks=seq(from=153,to=156,by=1),labels=c("153.0","154.0","15
  scale_y_continuous(limits = c(6.5,9.0), breaks=seq(from =7,to=9,by=0.5))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



We can view a contour plot of the samples:

```
ggplot(data=samples_height_model, aes(mu, sigma)) +
  geom_density_2d(bins=10)
```

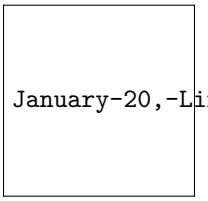


Comparing the marginal plots

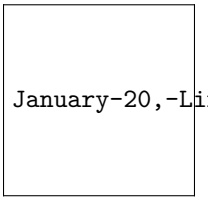
What do the "marginal" densities mean? They tell us how one parameter is distributed if we don't know the value of the other parameters.

From samples, we can get them from our samples without any real work by just focusing on one parameter at a time:

```
ggplot(data=samples_height_model, aes(x=mu)) +
  geom_density()
```



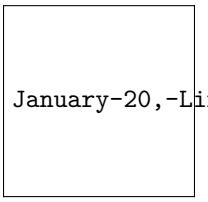

```
ggplot(data=samples_height_model, aes(x=sigma)) +
  geom_density()
```



Lots of packages have methods for plotting samples from posteriors. One common method is some kind of "pair" plot, which combines a scatter plot of each pair of parameters and a marginal density plot of each individual parameter. We'll use the `bayesplot` package version, with some specific options that make it look nicer.

```
bayesplot::mcmc_pairs(samples_height_model,diag_fun = "dens",
  off_diag_fun =  "hex")
```

```
## Warning: Only one chain in 'x'. This plot is more useful with multiple
## chains.
```



Exercise: Quantify the distributions

All of the methods we've used for quantifying a single paremeter's posterior distribution can still be used in the same way as before. For both μ and σ, calculate the mean, median, 5 and 95% quantiles, and the HPDIs.