Regresión Bayesiana

Victor Gallego y Roi Naveiro

20/05/2019

```r
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)
theme_set(theme_minimal())
library(ggrepel)
library(rstanarm)
```

```
## Loading required package: Rcpp

## rstanarm (Version 2.17.2, packaged: 2017-12-20 23:59:28 UTC)

## - Do not expect the default priors to remain the same in future rstanarm versions.

## Thus, R scripts should specify priors explicitly, even if they are just the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

## options(mc.cores = parallel::detectCores())

## - Plotting theme set to bayesplot::theme_default().
```

```r
library(reshape2)
library(tidyr)
```

```
## 
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
## 
##     smiths
```

Exploración y preprocesado de datos

Explora los datos, en particular las variables *brainwt* (peso del cerebro), *bodywt* (peso corporal) y *sleep_total* (horas de sueño diarias).

```r
msleep %>%
  select(name, sleep_total, brainwt, bodywt) %>%
    arrange(desc(brainwt / bodywt))
```

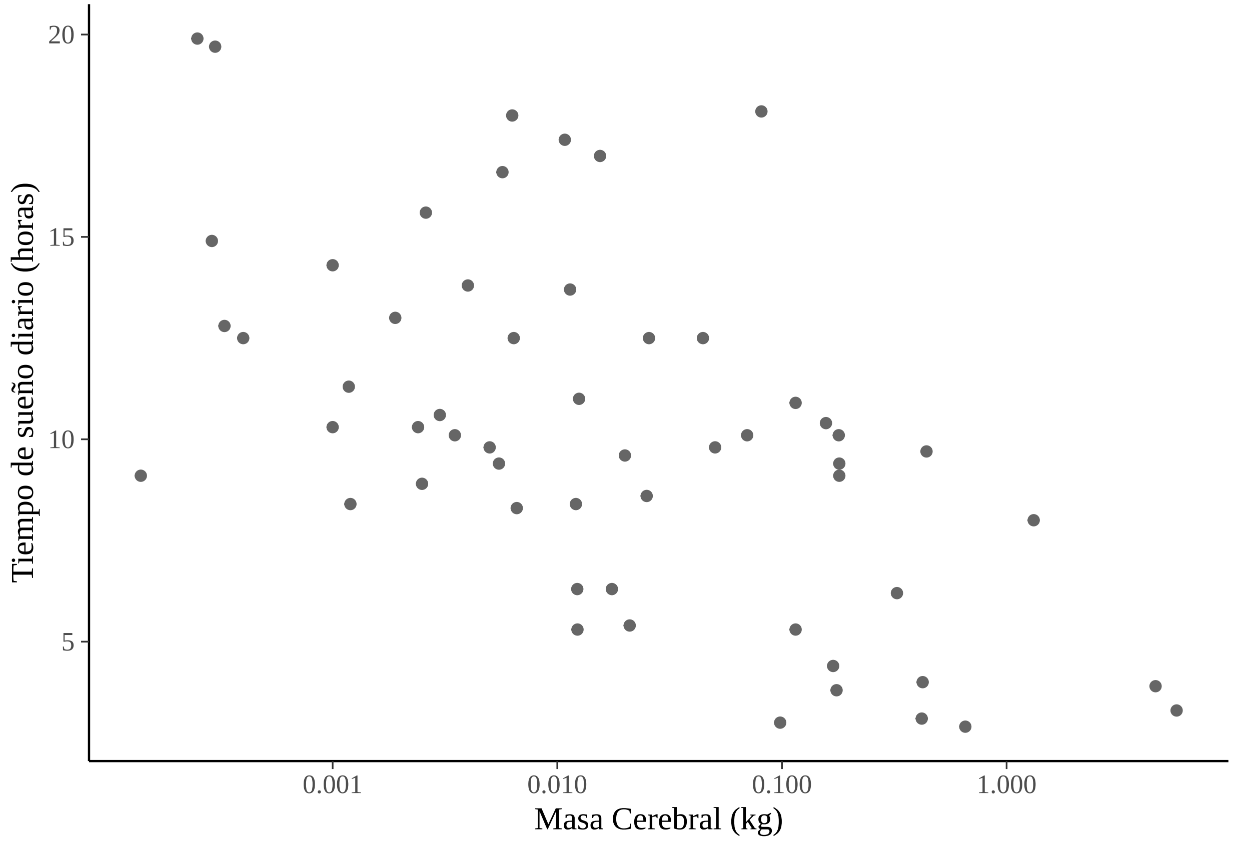
```
## # A tibble: 83 x 4
##    name                           sleep_total  brainwt bodywt
##    <chr>                                <dbl>    <dbl>  <dbl>
##  1 Thirteen-lined ground squirrel        13.8 0.004     0.101
##  2 Owl monkey                            17   0.0155    0.48 
##  3 Lesser short-tailed shrew              9.1 0.000140  0.005
##  4 Squirrel monkey                        9.6 0.02      0.743
##  5 Macaque                               10.1 0.179     6.8  
##  6 Little brown bat                      19.9 0.00025   0.01 
##  7 Galago                                 9.8 0.005     0.2  
##  8 Mole rat                              10.6 0.003     0.122
##  9 Tree shrew                             8.9 0.0025    0.104
## 10 Human                                  8   1.32     62
## # ... with 73 more rows
```

Elimina las especies con valores ausentes en la variable brainwt. Transforma las variables brainwt, bodywt y sleep_total usando log10.

```
msleep <- msleep %>%
  filter(!is.na(brainwt)) %>%
  mutate(log_brainwt = log10(brainwt),
         log_bodywt = log10(bodywt),
         log_sleep_total = log10(sleep_total))
```

Representa gráficamente la masa cerebral (en escala logarítmica) frente a las horas de sueño diarias.

```
ggplot(msleep) +
  aes(x = brainwt, y = sleep_total) +
  geom_point(color = "grey40") +
  scale_x_log10(breaks = c(.001, .01, .1, 1)) +
    labs(x = "Masa Cerebral (kg)", y = "Tiempo de sueño diario (horas)")
```



Modelo de regresión Bayesiana

Ajusta el modelo de regresión bayesiana con variable respuesta: **logaritmo de horas de sueño**, y covariable **logaritmo de peso cerebral** utilizando stan_glm.

```
bay_reg <- stan_glm(
  log_sleep_total ~ log_brainwt,
  family = gaussian(),
  data = msleep,
  prior = normal(0, 3),
  prior_intercept = normal(0, 3))
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
##
## Gradient evaluation took 3.8e-05 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0.38 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.057459 seconds (Warm-up)
##                0.054098 seconds (Sampling)
##                0.111557 seconds (Total)
##
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
##
## Gradient evaluation took 1.3e-05 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.052315 seconds (Warm-up)
##                0.054814 seconds (Sampling)
##                0.107129 seconds (Total)
##
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
##
## Gradient evaluation took 1.4e-05 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0.14 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.052945 seconds (Warm-up)
##                0.053587 seconds (Sampling)
##                0.106532 seconds (Total)
##
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
##
## Gradient evaluation took 1.3e-05 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.053197 seconds (Warm-up)
##                0.055996 seconds (Sampling)
##                0.109193 seconds (Total)
```

Explora la salida del modelo. ¿Se cumple la condición de convergencia implicada por el estadístico Gelman-Rubin?

```r
summary(bay_reg, probs=c(0.1, 0.5, 0.9))
```

```
##
## Model Info:
##
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      log_sleep_total ~ log_brainwt
##  algorithm:    sampling
##  priors:       see help('prior_summary')
##  sample:       4000 (posterior sample size)
##  observations: 56
##  predictors:   2
##
## Estimates:
##                 mean   sd   10%   50%   90%
## (Intercept)    0.7    0.0  0.7   0.7   0.8
## log_brainwt   -0.1    0.0 -0.2  -0.1  -0.1
## sigma          0.2    0.0  0.2   0.2   0.2
## mean_PPD       1.0    0.0  0.9   1.0   1.0
## log-posterior 15.3    1.3 13.7  15.7  16.6
##
## Diagnostics:
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  3574
## log_brainwt   0.0  1.0  3739
## sigma         0.0  1.0  3109
## mean_PPD      0.0  1.0  4000
## log-posterior 0.0  1.0  1830
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Análisis de los resultados

Pinta el histograma de la distribución a posteriori empírica de los parámetros sobre los que se ha hecho inferencia.

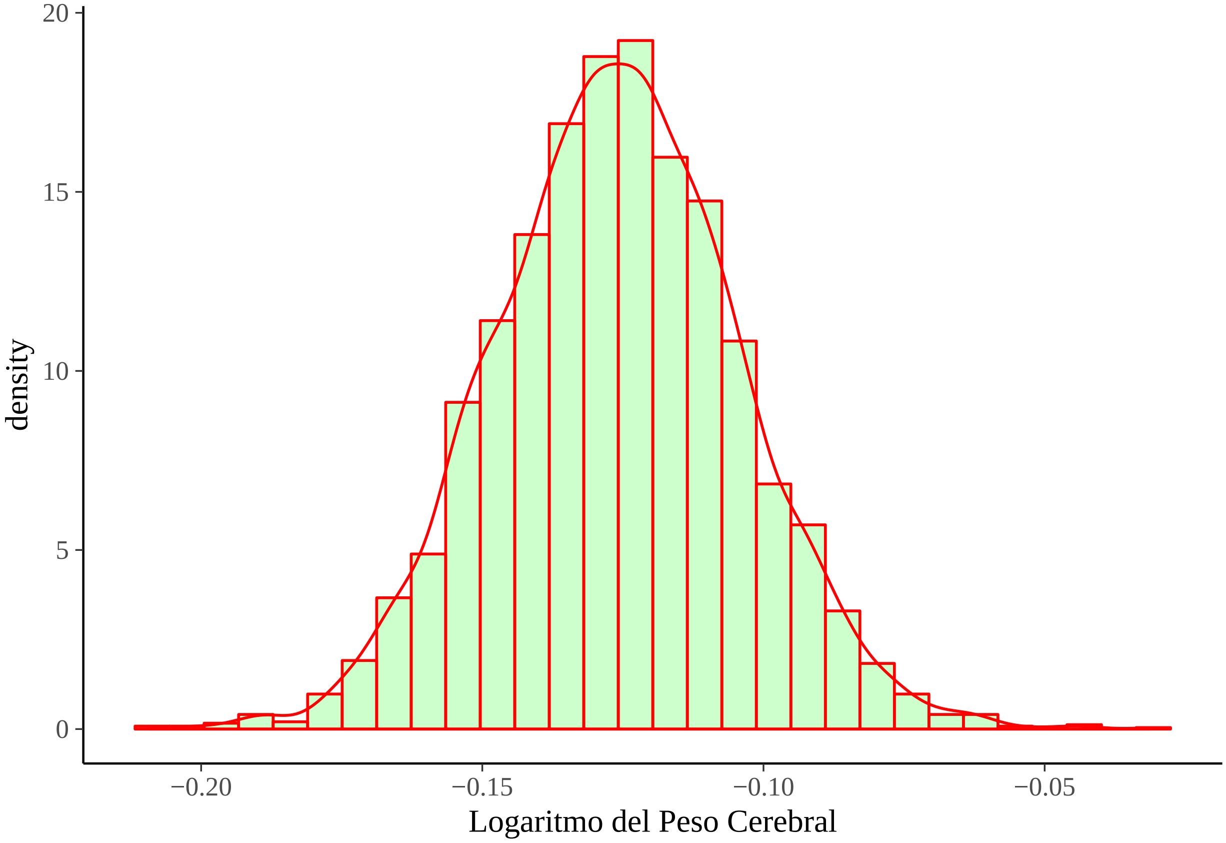
¿Es el coeficiente del peso cerebral significativamente menor que 1?

```r
samples <- bay_reg %>%
  as_data_frame %>%
  rename(intercept = `(Intercept)`)
```

```
## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

```r
ggplot(data=samples, aes(samples$log_brainwt)) +
  geom_histogram(aes(y =..density..),
                 col="red",
                 fill="green",
                 alpha=.2) +
  geom_density(col=2) + xlab("Logaritmo del Peso Cerebral")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Predicción

Muestrea de la distribución predictiva a posteriori en una red de valores de peso cerebral. Representa las observaciones junto con la mediana y el intervalo del 95% de probabilidad de las muestras a posteriori.

Primero construye una red de 80 valores de peso cerebral contenidos en el mismo intervalo que las medidas observadas de esta variable.

```
x_rng <- range(msleep$log_brainwt)
x_steps <- seq(x_rng[1], x_rng[2], length.out = 80)
new_data <- data_frame(
  observation = seq_along(x_steps),
  log_brainwt = x_steps)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

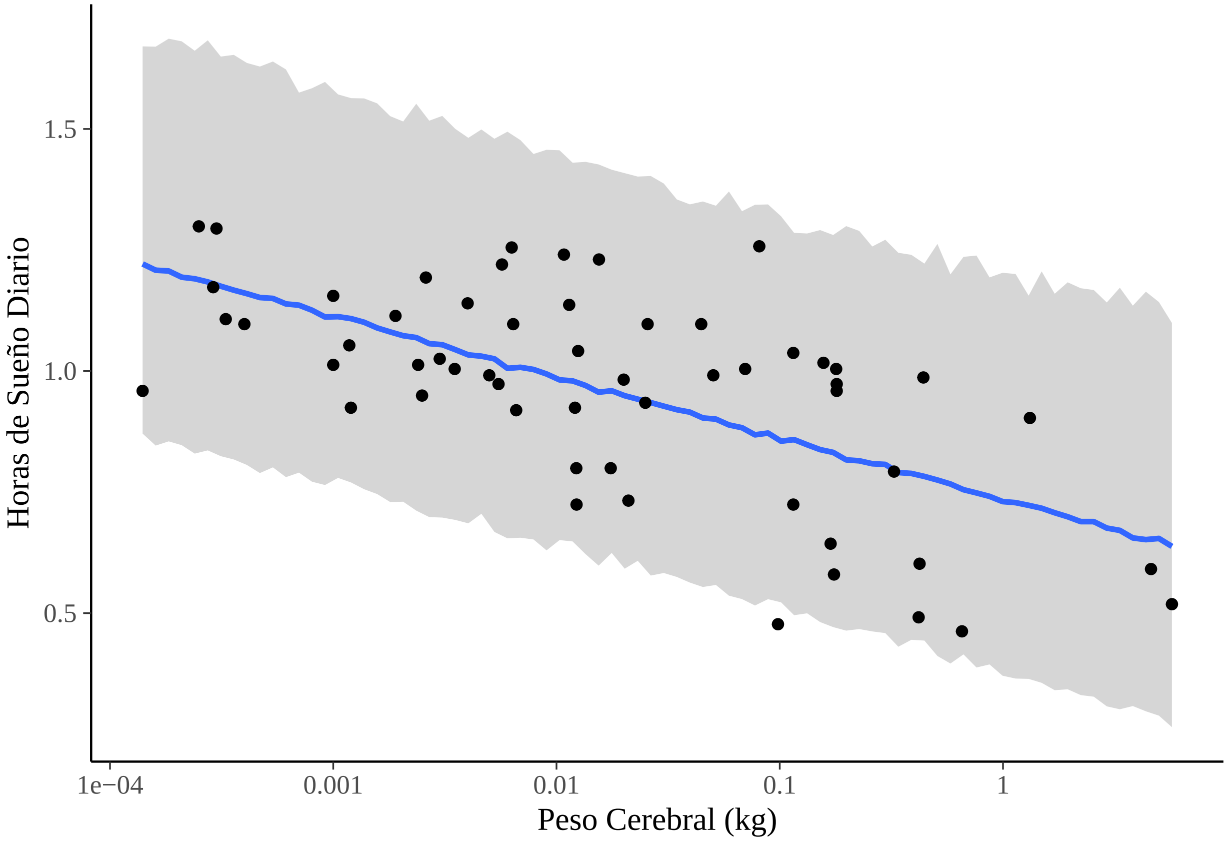
Muestrea de la distribución a posteriori utilizando la función *posterior_predict*

```
pred_post = posterior_predict(bay_reg, newdata = new_data)

df_pred = data.frame(t(apply(t(pred_post), 1, quantile, probs = c(0.025, 0.5, 0.995), na.rm = TRUE)))

names(df_pred)<-c("lower","median","upper")

df_pred$log_brainwt = new_data$log_brainwt

head(df_pred)
```

```
##       lower   median    upper log_brainwt
## 1 0.8708343 1.221307 1.670594   -3.853872
## 2 0.8460737 1.208285 1.669989   -3.795509
## 3 0.8547688 1.206746 1.686485   -3.737146
## 4 0.8469327 1.193821 1.681054   -3.678784
## 5 0.8295054 1.190655 1.661581   -3.620421
## 6 0.8361489 1.184151 1.683192   -3.562058
```

Representa gráficamente

```
ggplot(msleep) +
  aes(x = log_brainwt) +
  geom_ribbon(aes(ymin = lower, ymax = upper), data = df_pred,
              alpha = 0.4, fill = "grey60") +
  geom_line(aes(y = median), data = df_pred, colour = "#3366FF", size = 1) +
  geom_point(aes(y = log_sleep_total)) +
  scale_x_continuous(labels = function(x) 10 ^ x) +
  labs(x = "Peso Cerebral (kg)", y = "Horas de Sueño Diario")
```


Horas de Sueño Diario
1e−04
Peso Cerebral (kg)