Computational Biology Group

D BSSE

ETH zürich

Introduction to Bayesian Statistics with R

7: Exercise solutions

Jack Kuipers

29 November 2022

First we load the tidyverse, cowplot, brms, define a helper function and set a seed.

```
library(tidyverse); options(dplyr.summarise.inform = FALSE) # suppress summarise warnings
library(cowplot); library(brms)
tidy_output <- function(x){as.character(signif(x, 3))}
set.seed(42)
```

Exercise 7.1 - Bayesian multiple regression

Run a Bayesian multiple regression model akin to

$$\text{seap\_s} = \text{log10conc} + \text{experiment} + \text{cytokine}$$

with `brms` *and the* `brm()` *function.*

- *Did you make your model robust?*
- *What did you select for your priors?*
- *Visualise the posterior distribution of the slope coefficient between SEAP and the log concentration.*

We read in the data

```
gene_circuit_df <- read.csv("./data/genetic_circuit.csv")
```

and we *relevel* the factor variable `cytokine` so we can later compare `IL4` and `IL13` to `IL4 + IL13`

```
gene_circuit_df$cytokine <- relevel(as.factor(gene_circuit_df$cytokine),
                                    ref = "IL-4 and IL-13")
```

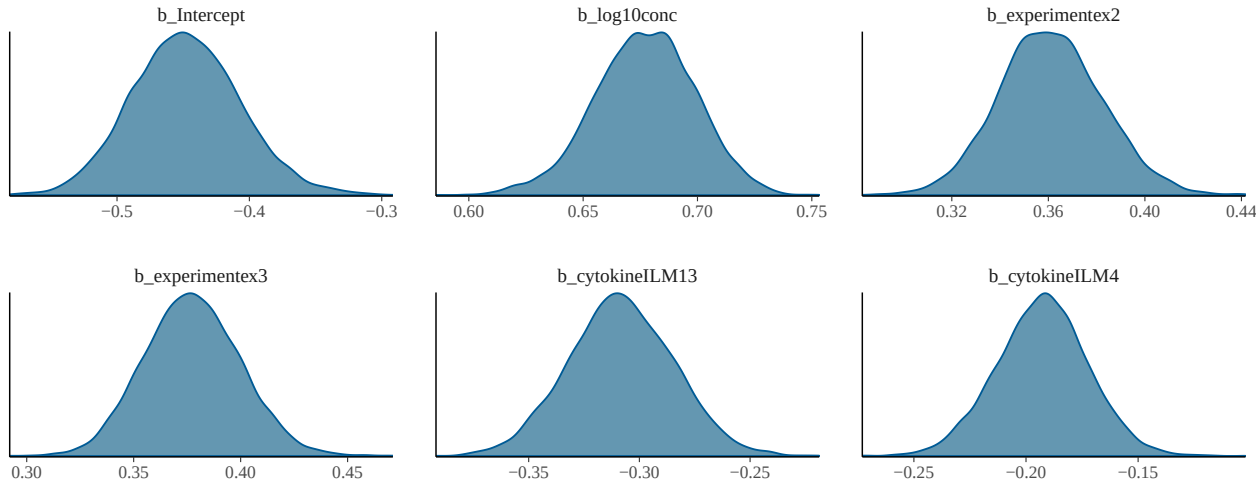
Before running the data through the suggested model, we have moderately sized data with 147 entries, and continuous predictors with a range of about 1 or so (after our transformations), so we would expect our regression coefficients to also be of that sort of scale. Not knowing much about the data, we can have heavier tails and pick a Student-t with a low degree of freedom. I might also be skeptical we would get very normal data, so let's also have a Student-t link like in the robust t-test.

With our helper function, we therefore run the following

```
brmfit_ex7 <- run_model(brm(seap_s ~ log10conc + experiment + cytokine, family = student,
  prior = prior(student_t(3, 0, 1), class = "b"), # sets for all regression coefficients
  gene_circuit_df), "./brm_models_exercises/blm_ex7")
```

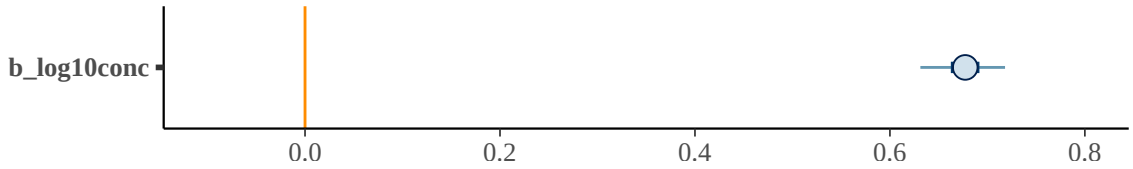
For the posterior distribution of the regression coefficients

```
mcmc_plot(brmfit_ex7, variable = "b_", regex = TRUE, type = "dens")
```



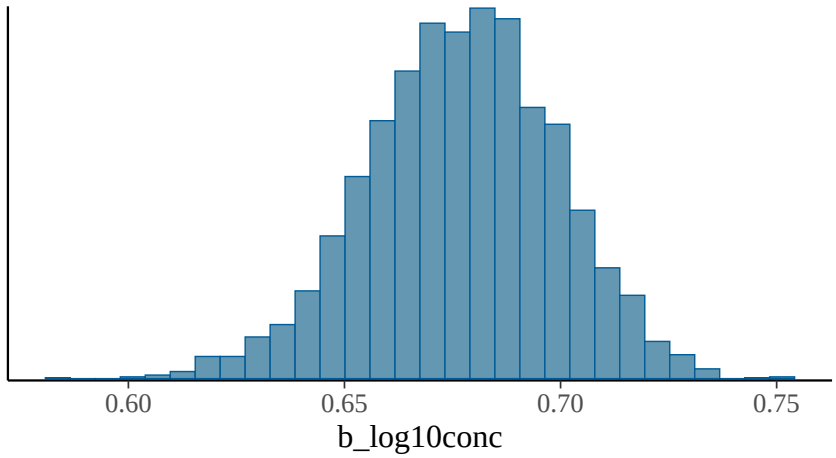
they all seem quite far from 0, suggesting strong batch effects from the different experiments and that the combination of cytokines is more effective than the individual ones. For the linear response of SEAP to the cytokines, the regression coefficient with `log10conc` is very far from 0, as the following plot also highlights:

```
mcmc_plot(brmfit_ex7, variable = "b_log10conc", type = "intervals", prob_outer = 0.95) +
  xlim(-0.1, 0.8) + geom_vline(xintercept = 0, color = "darkorange")
```



For the posterior visualisation, we already have the kernel density above, and maybe we can use a histogram to complement it

```
mcmc_plot(brmfit_ex7, variable = "b_log10conc", type = "hist")
```



Exercise 7.2 - Multiple regression

Run a multiple regression of

$$\text{seap\_s} = \text{log10conc} + \text{experiment} + \text{cytokine}$$

- *Visualise the residuals. Does a robust model for the Bayesian model in Exercise 7.1 make sense?*
- *Examine the regression coefficients.*

We simply run our data through the suggested model

```
seap_fit <- lm(seap_s ~ log10conc + experiment + cytokine, data = gene_circuit_df)
```

From there we can plot the residuals

```
res <- seap_fit$residuals
fitted <- seap_fit$fitted.values
d <- data.frame(residuals = unname(res), fitted = fitted,
                cytokine = gene_circuit_df$cytokine,
                experiment = gene_circuit_df$experiment)
p <- ggplot(d) + theme_minimal() + theme(legend.position = "bottom") +
  scale_x_continuous("Fitted")
p1 <- p + geom_point(aes(fitted, residuals, colour = cytokine)) +
  scale_y_continuous(expression(epsilon))
p2 <- p + geom_point(aes(fitted, residuals, colour = experiment)) +
  scale_y_continuous("")
cowplot::plot_grid(p1, p2, ncol = 2)
```

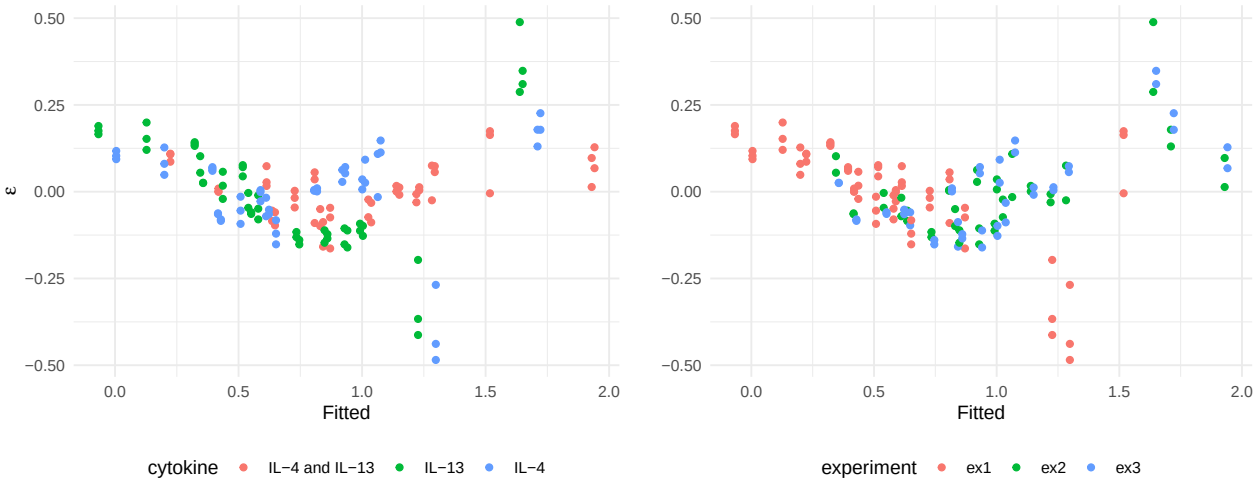


Figure 1: Visualization of residuals against fitted data.

We can see a clear wobble in the residuals, probably suggesting that a linear model between (log) concentration and SEAP may be too simplistic. We would want a biologically-inspired functional relationship between the two to model the data better. This would also have allowed a better design of the concentrations at which to perform the experiments. In particular, from the right plot above, experiment 1 seems to have a quite different slope than the others, so we would may want to include an interaction term between concentration and the experiment (or at least with experiment 1). However, not all concentration/experiment pairs were collected in the data, making this also tricky.

Staying with the current model, we plot the residual distribution.

```
p.res <- data.frame(Residuals = res) %>% ggplot() + theme_minimal()
p1 <- p.res +
  geom_histogram(aes(Residuals), bins = 60, fill = "darkgrey", colour = "black") +
  scale_y_continuous("Count")
p.q <- p.res + scale_y_continuous("Empirical Quantile") +
  scale_x_continuous("Theoretical Quantile")
```

```
p2 <- p.q + stat_qq(aes(sample = Residuals)) + stat_qq_line(aes(sample = Residuals))
cowplot::plot_grid(p1, p2, ncol = 2)
```

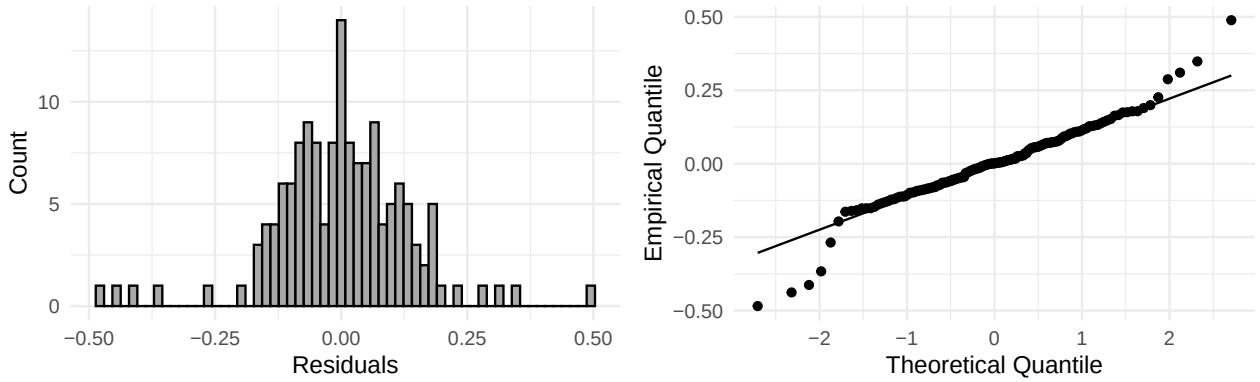


Figure 2: Residual distributions.

The residuals are not normally distributed, with heavier tails. A robust model in Exercise 7.1 would make a lot of sense.

With a better noise model and more appropriate link between concentration and SEAP, these might improve, but there are also many other reasons why we might see non-normality in the residues as artefacts: we couldn't measure SEAP expression accurately, the controls/circuits did not work, technical errors, not enough measurements, all of which are not uncommon for biological data.

To complete the exercise let's look at the summary of the fit:

```
summary(seap_fit)$coefficients %>% data.frame %>%
  rownames_to_column %>%
  rename(Coefficient = rowname, "Standard error" = Std..Error,
         "$t$-statistic" = t.value, "$p$-value" = Pr...t..) %>%
  mutate_if(is.numeric, tidy_output) -> seap_fit_coefficients
seap_fit_coefficients %>% kable(caption="Coefficients estimated in the linear model.")
```

Table 1: Coefficients estimated in the linear model.

Coefficient	Estimate	Standard error	t-statistic	p-value
(Intercept)	-0.423	0.0411	-10.3	7.19e-19
log10conc	0.647	0.0189	34.2	3.42e-70
experimentex2	0.412	0.0268	15.4	6.8e-32
experimentex3	0.423	0.0268	15.8	5.68e-33
cytokineIL-13	-0.291	0.0272	-10.7	6.39e-20
cytokineIL-4	-0.219	0.0272	-8.05	3.03e-13

The regression coefficients reveal the following:

- `log10conc` has a slope significantly away from zero as we would expect if the gene circuit were working properly and reacting to the cytokine concentrations.
- Cytokine `IL4` and `IL13` separately have a lower SEAP than in combination.
- The coefficients for the experiments have very low p-values, potentially indicating severe batch effects.