

chp6

Vasco Brazão

09/06/2021

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.3      v purrr 0.3.4
## v tibble 3.0.6       v dplyr 1.0.4
## v tidyr 1.1.2        v stringr 1.4.0
## v readr 1.4.0        v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(here)
```

```
## here() starts at C:/Users/admin/Documents/statistical-rethinking
```

I did 6E1 - 6M1 in a paper notebook and am too lazy to type things up right now.

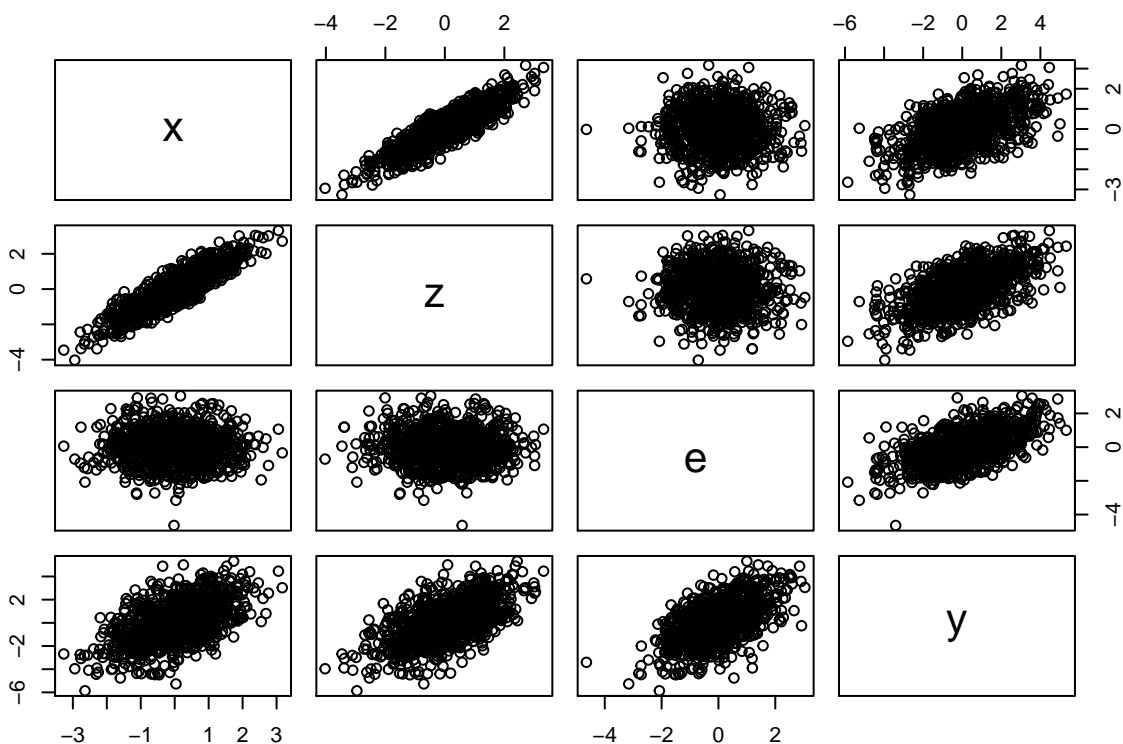
6M2

We should simulate from DAG $X \rightarrow Z \rightarrow Y$, such that the correlation between X and Z is very large.

```
Nsim <- 1000

dat <- tibble::tibble(
  x = rnorm(n = Nsim),
  z = rnorm(n = Nsim, mean = x, sd = 0.5),
  e = rnorm(n = Nsim),
  y = rnorm(n = Nsim, mean = z + e)
)

pairs(dat)
```



```
cor(dat)
```

```
##           x           z           e           y
## x  1.00000000  0.89515150 -0.01235672  0.5415898
## z  0.89515150  1.00000000 -0.02549268  0.5933578
## e -0.01235672 -0.02549268  1.00000000  0.5772075
## y  0.54158978  0.59335783  0.57720745  1.0000000
```

```
summary(lm(y ~ x + z, data = dat))
```

```
##
## Call:
## lm(formula = y ~ x + z, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6671 -1.0135  0.0034  0.9786  4.9045
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.00460    0.04535  -0.101   0.919
## x             0.09331    0.10148   0.920   0.358
## z             0.85081    0.08903   9.557 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.433 on 997 degrees of freedom
## Multiple R-squared:  0.3526, Adjusted R-squared:  0.3513
## F-statistic: 271.5 on 2 and 997 DF,  p-value: < 2.2e-16
```

First, we can see that we were successful in simulating from the DAG and creating a rather large association between the predictors.

Next, we see no “problem” in the linear model results – only z has a large estimated coefficient. Let’s replicate this result in `quap` and `brms`.

```
q6m2 <- rethinking::quap(
  alist(
    y ~ dnorm(mu, sigma),
    mu <- a + bx*x + bz*z,
    a ~ dnorm(0, 2),
    bx ~ dnorm(0, 2),
    bz ~ dnorm(0, 2),
    sigma ~ dexp(1)
  ), data = dat
)
```

```
rethinking::precis(q6m2)
```

```
##              mean          sd          5.5%          94.5%
## a      -0.004557387 0.04524329 -0.07686491 0.06775013
## bx       0.095216812 0.10105422 -0.06628735 0.25672097
## bz       0.849207917 0.08866293  0.70750743 0.99090841
## sigma   1.430228434 0.03195203  1.37916292 1.48129395
```

```
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)
```

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

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

```
## To avoid recompilation of unchanged Stan programs, we recommend calling
```

```
## rstan_options(auto_write = TRUE)
```

```
## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
```

```
##
```

```
## Attaching package: 'rstan'
```

```
## The following object is masked from 'package:tidyr':
```

```
##
```

```
##      extract
```

```
## Loading required package: parallel

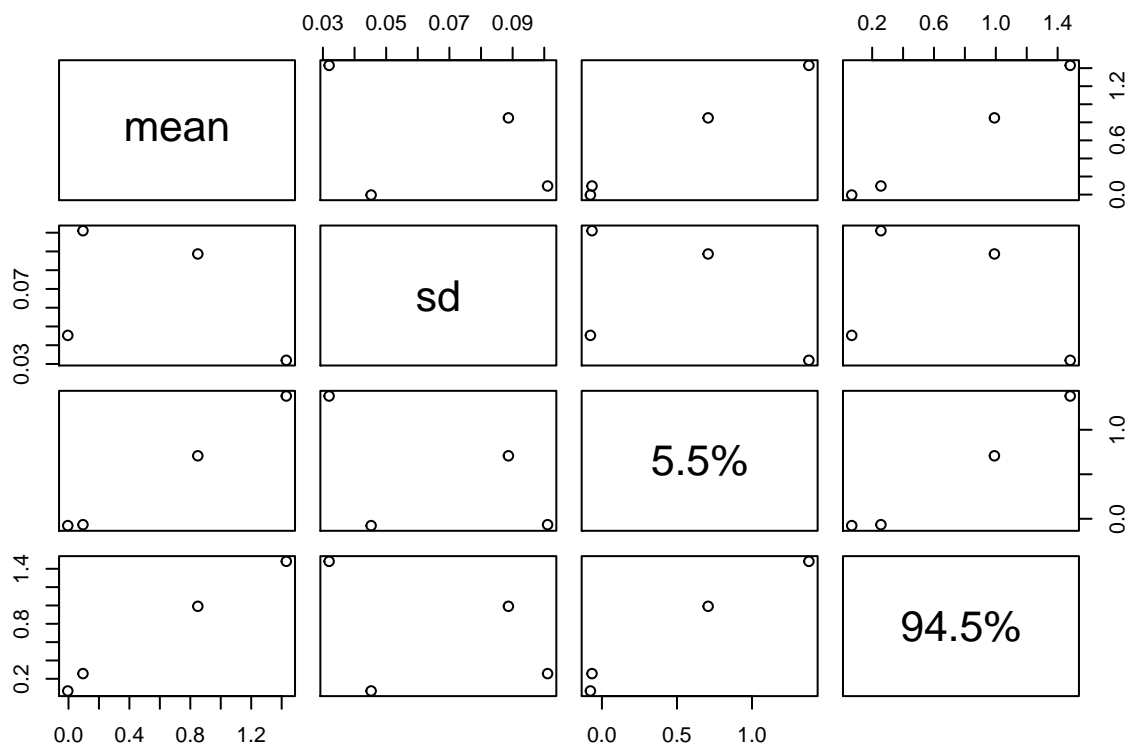
## rethinking (Version 2.13)

##
## Attaching package: 'rethinking'

## The following object is masked from 'package:purrr':
##
##   map

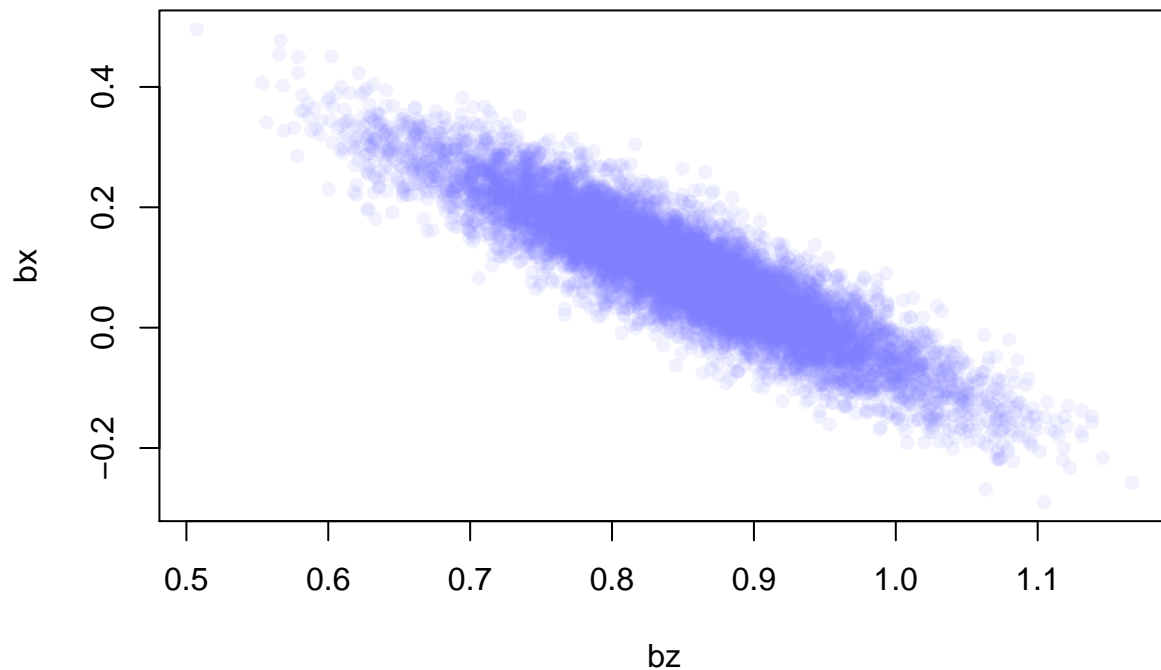
## The following object is masked from 'package:stats':
##
##   rstudent
```

```
plot(precis(q6m2))
```



```
post <- rethinking::extract.samples(q6m2)

plot(bx ~ bz, post, col = col.alpha(rangi2, 0.1), pch = 16)
```



```
cor(post$bx, post$bz)
```

```
## [1] -0.895938
```

Here we see that, conditional on the model, bx and bz are quite strongly correlated.

The difference here is that X does not actually affect Y directly, while in the legs example both legs are affected by the height of the person.

Let's just code this model in brms.

```
detach(package:rethinking, unload=TRUE)
```

```
library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.14.4). Useful instructions  
## can be found by typing help('brms'). A more detailed introduction  
## to the package is available through vignette('brms_overview').
```

```
##
```

```
## Attaching package: 'brms'
```

```
## The following object is masked from 'package:rstan':
##
##      loo
```

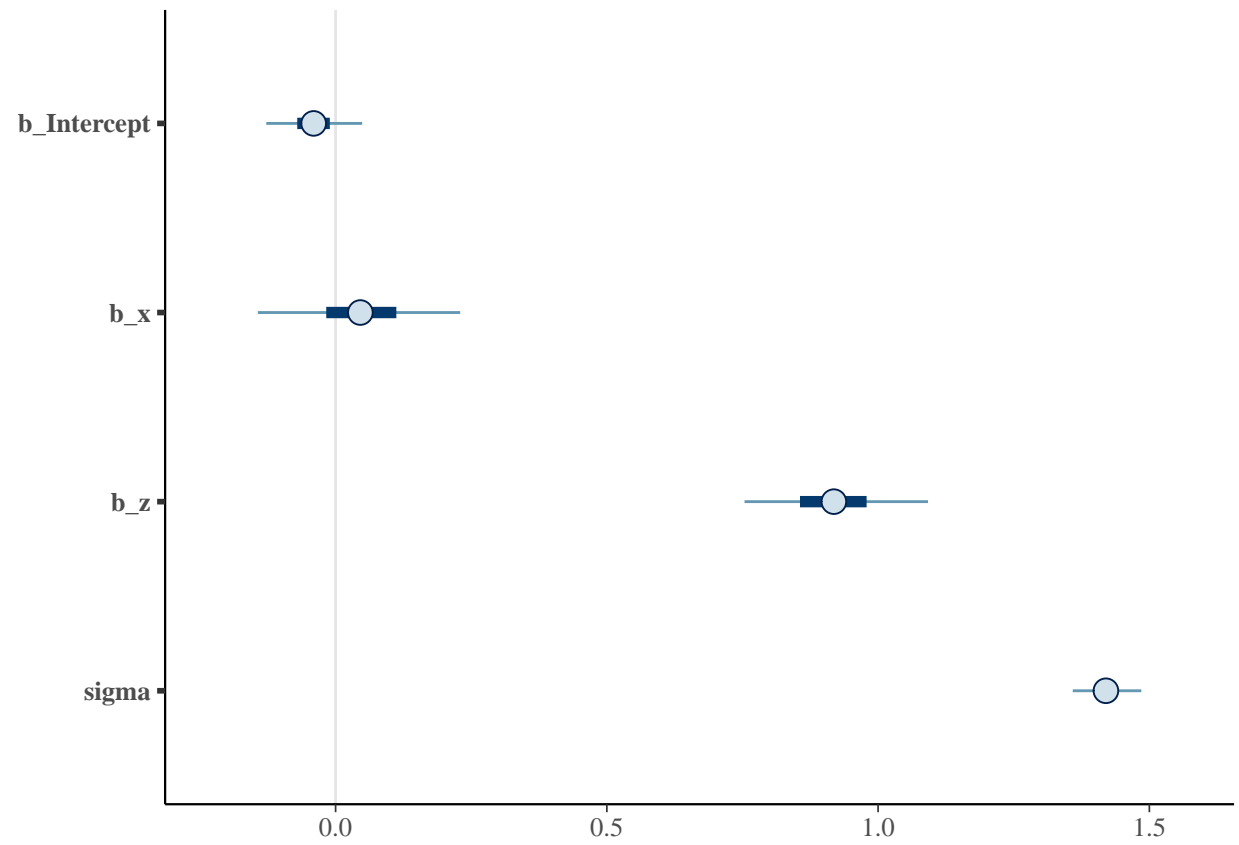
```
## The following object is masked from 'package:stats':
##
##      ar
```

```
b6m2 <- brm(
  data = dat,
  family = gaussian,
  y ~ 1 + x + z,
  prior = c(prior(normal(0, 2), class = Intercept),
            prior(normal(0, 2), class = b),
            prior(exponential(1), class = sigma)),
  iter = 2000, warmup = 1000, chains = 4, cores = 4,
  file = here::here("chp6", "b6m2")
)
```

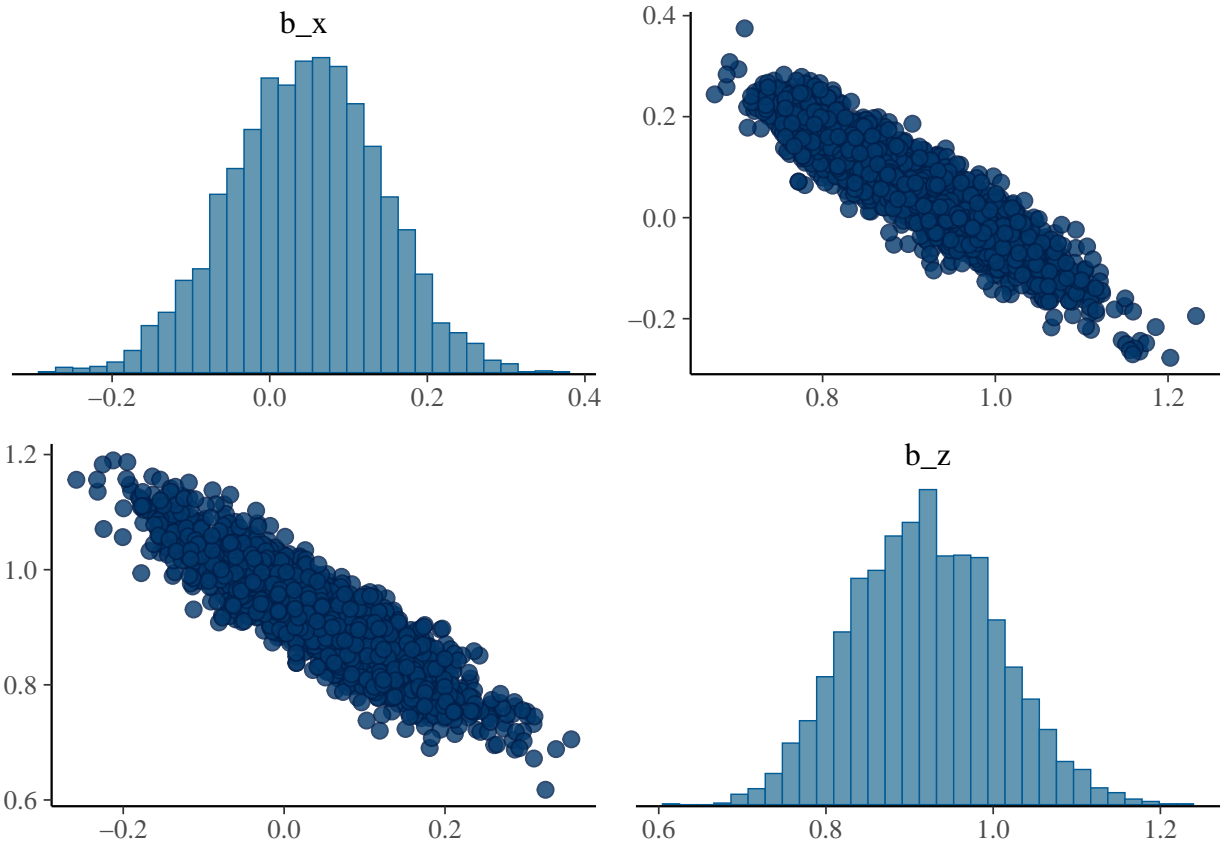
```
print(b6m2)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: y ~ 1 + x + z
## Data: dat (Number of observations: 1000)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.04      0.04   -0.13    0.05 1.00     2824     2289
## x              0.05      0.10   -0.14    0.23 1.00     1889     2363
## z              0.92      0.09    0.75    1.09 1.00     1969     2467
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma       1.42      0.03    1.36    1.49 1.00     2858     2404
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
brms::mcmc_plot(b6m2,
  type = "intervals",
  prob = .5,
  prob_outer = .95,
  point_est = "mean")
```



```
pairs(b6m2, pars = parnames(b6m2)[2:3])
```



Ooh what cool plots we can make.

6M3

Going clockwise, starting at top left DAG:

1. Z
2. nothing
3. nothing
4. A

Needed a lot of help from wjakethompson. Perhaps I'm a bit rusty. Totally forgot that we want to include indirect causal paths.