

Variable Selection

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.3      v tidyr      1.3.1
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(BAS)      # spike-and-slab
library(rstanarm) # horseshoe
```

```
Loading required package: Rcpp
This is rstanarm version 2.32.1
- See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
- Default priors may change, so it's safest to specify priors, even if equivalent to the default
- For execution on a local, multicore CPU with excess RAM we recommend calling
  options(mc.cores = parallel::detectCores())
```

```
library(loo)
```

```
This is loo version 2.8.0
- Online documentation and vignettes at mc-stan.org/loo
- As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' argument
```

```
library(kableExtra)
```

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

group_rows

```
df <- read.csv("Data/student-scores-clean.csv") |>
  select(-id, -first_name, -last_name, -email) |>
  mutate(
    gender = factor(gender, labels = c("Female", "Male")),
    part_time_job = factor(part_time_job),
    extracurricular_activities = factor(extracurricular_activities),
    career_aspiration = factor(career_aspiration)
  )

# keep only non-collinear behavioral/demographic predictors
vars_behavioral <- c("gender", "part_time_job",
  "absence_days", "extracurricular_activities",
  "weekly_self_study_hours", "career_aspiration")

df_sub <- df %>% select(average_score, all_of(vars_behavioral))
```

```
bas_fit <- bas.lm(
  average_score ~ .,
  data = df_sub,
  prior = "ZS-null",
  modelprior = uniform(),
  n.models = 2^length(vars_behavioral)
)

# posterior inclusion probabilities (PIP)
pip <- summary(bas_fit)
```

```
hs_fit <- stan_glm(
  average_score ~ .,
  data = df_sub,
  family = gaussian(),
  prior_intercept = normal(0, 10),
```

```

prior      = hs(),      # global-local horseshoe
chains     = 4,
iter       = 2000,
seed       = 2025
)

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 3.9e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.39 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 2000 [0%] (Warmup)

Chain 1: Iteration: 200 / 2000 [10%] (Warmup)

Chain 1: Iteration: 400 / 2000 [20%] (Warmup)

Chain 1: Iteration: 600 / 2000 [30%] (Warmup)

Chain 1: Iteration: 800 / 2000 [40%] (Warmup)

Chain 1: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 1: Iteration: 1001 / 2000 [50%] (Sampling)

Chain 1: Iteration: 1200 / 2000 [60%] (Sampling)

Chain 1: Iteration: 1400 / 2000 [70%] (Sampling)

Chain 1: Iteration: 1600 / 2000 [80%] (Sampling)

Chain 1: Iteration: 1800 / 2000 [90%] (Sampling)

Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 6.643 seconds (Warm-up)

Chain 1: 3.818 seconds (Sampling)

Chain 1: 10.461 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 1.1e-05 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

Chain 2:

Chain 2: Iteration: 1 / 2000 [0%] (Warmup)

Chain 2: Iteration: 200 / 2000 [10%] (Warmup)

Chain 2: Iteration: 400 / 2000 [20%] (Warmup)

```

Chain 2: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 2: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 7.671 seconds (Warm-up)
Chain 2:           3.594 seconds (Sampling)
Chain 2:           11.265 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 1.1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 2000 [ 0%] (Warmup)
Chain 3: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 3: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 3: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 8.759 seconds (Warm-up)
Chain 3:           6.778 seconds (Sampling)
Chain 3:           15.537 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 1.2e-05 seconds

```

Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.12 seconds.
Chain 4: Adjust your expectations accordingly!

Chain 4:

Chain 4:

```
Chain 4: Iteration:    1 / 2000 [  0%] (Warmup)
Chain 4: Iteration:   200 / 2000 [ 10%] (Warmup)
Chain 4: Iteration:   400 / 2000 [ 20%] (Warmup)
Chain 4: Iteration:   600 / 2000 [ 30%] (Warmup)
Chain 4: Iteration:   800 / 2000 [ 40%] (Warmup)
Chain 4: Iteration:  1000 / 2000 [ 50%] (Warmup)
Chain 4: Iteration:  1001 / 2000 [ 50%] (Sampling)
Chain 4: Iteration:  1200 / 2000 [ 60%] (Sampling)
Chain 4: Iteration:  1400 / 2000 [ 70%] (Sampling)
Chain 4: Iteration:  1600 / 2000 [ 80%] (Sampling)
Chain 4: Iteration:  1800 / 2000 [ 90%] (Sampling)
Chain 4: Iteration:  2000 / 2000 [100%] (Sampling)
```

Chain 4:

Chain 4: Elapsed Time: 9.657 seconds (Warm-up)

Chain 4: 6.807 seconds (Sampling)

Chain 4: 16.464 seconds (Total)

Chain 4:

Warning: There were 21 divergent transitions after warmup. See <https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup> to find out why this is a problem and how to eliminate them.

Warning: Examine the pairs() plot to diagnose sampling problems

```
print(hs_fit, digits = 2)
```

stan_glm

family: gaussian [identity]

formula: average_score ~ .

observations: 2000

predictors: 22

	Median	MAD_SD
(Intercept)	77.59	0.68
genderMale	-0.28	0.24
part_time_job1	-0.08	0.25
absence_days	0.03	0.05

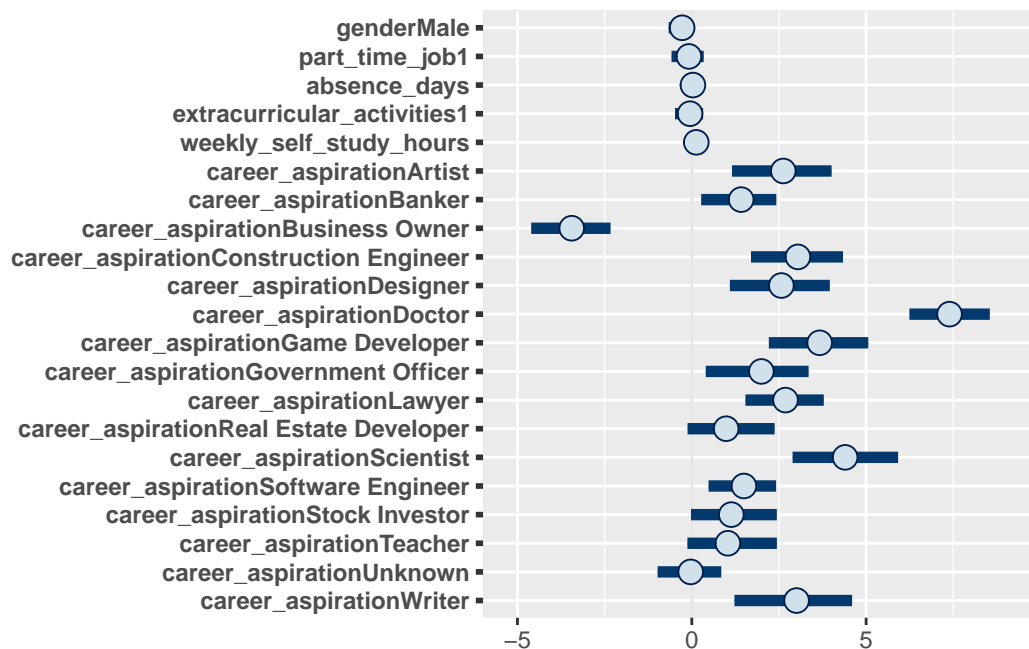
extracurricular_activities1	-0.05	0.21
weekly_self_study_hours	0.13	0.02
career_aspirationArtist	2.62	0.87
career_aspirationBanker	1.42	0.64
career_aspirationBusiness Owner	-3.44	0.69
career_aspirationConstruction Engineer	3.04	0.78
career_aspirationDesigner	2.57	0.84
career_aspirationDoctor	7.39	0.70
career_aspirationGame Developer	3.67	0.85
career_aspirationGovernment Officer	2.00	0.88
career_aspirationLawyer	2.69	0.66
career_aspirationReal Estate Developer	0.99	0.87
career_aspirationScientist	4.40	0.93
career_aspirationSoftware Engineer	1.50	0.59
career_aspirationStock Investor	1.13	0.80
career_aspirationTeacher	1.04	0.88
career_aspirationUnknown	-0.03	0.47
career_aspirationWriter	3.00	0.97

Auxiliary parameter(s):

	Median	MAD	SD
sigma	4.75	0.08	

* For help interpreting the printed output see ?print.stanreg
 * For info on the priors used see ?prior_summary.stanreg

```
plot(hs_fit, pars = c("beta"), prob = 0.9)
```



```
loo_fit <- loo(hs_fit)

# extract into a tibble
loo_df <- tibble(
  Metric      = c("elpd_loo", "se_elpd_loo", "p_loo"),
  Estimate    = c(loo_fit$estimates["elpd_loo", "Estimate"],
                  loo_fit$estimates["elpd_loo", "SE"],
                  loo_fit$estimates["p_loo", "Estimate"])
)

# display
kable(
  loo_df,
  digits = 2,
  caption = "LOO Cross-Validation Metrics"
)
```

Table 1: LOO Cross-Validation Metrics

Metric	Estimate
elpd_loo	-5965.97
se_elpd_loo	31.85

Metric	Estimate
p_loo	22.22

```
df_sub <- df %>%
  select(
    average_score,
    -starts_with("career_aspiration")
  )
```

```
# random intercept
model1 <- stan_lmer(
  average_score ~ weekly_self_study_hours
    + absence_days
    + extracurricular_activities
    + (1 | career_aspiration),
  data = df,
  prior_intercept = normal(75, 10),
  prior           = normal(0, 5),
  prior_covariance = decov(regularization = 2),
  chains = 4, iter = 2000, seed = 123
)
```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 7.3e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.73 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 2000 [0%] (Warmup)

Chain 1: Iteration: 200 / 2000 [10%] (Warmup)

Chain 1: Iteration: 400 / 2000 [20%] (Warmup)

Chain 1: Iteration: 600 / 2000 [30%] (Warmup)

Chain 1: Iteration: 800 / 2000 [40%] (Warmup)

Chain 1: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 1: Iteration: 1001 / 2000 [50%] (Sampling)

Chain 1: Iteration: 1200 / 2000 [60%] (Sampling)

Chain 1: Iteration: 1400 / 2000 [70%] (Sampling)

Chain 1: Iteration: 1600 / 2000 [80%] (Sampling)

Chain 1: Iteration: 1800 / 2000 [90%] (Sampling)

Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 5.246 seconds (Warm-up)
Chain 1: 1.873 seconds (Sampling)
Chain 1: 7.119 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).

Chain 2:
Chain 2: Gradient evaluation took 4.9e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.49 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration: 1 / 2000 [0%] (Warmup)
Chain 2: Iteration: 200 / 2000 [10%] (Warmup)
Chain 2: Iteration: 400 / 2000 [20%] (Warmup)
Chain 2: Iteration: 600 / 2000 [30%] (Warmup)
Chain 2: Iteration: 800 / 2000 [40%] (Warmup)
Chain 2: Iteration: 1000 / 2000 [50%] (Warmup)
Chain 2: Iteration: 1001 / 2000 [50%] (Sampling)
Chain 2: Iteration: 1200 / 2000 [60%] (Sampling)
Chain 2: Iteration: 1400 / 2000 [70%] (Sampling)
Chain 2: Iteration: 1600 / 2000 [80%] (Sampling)
Chain 2: Iteration: 1800 / 2000 [90%] (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 8.151 seconds (Warm-up)
Chain 2: 1.663 seconds (Sampling)
Chain 2: 9.814 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).

Chain 3:
Chain 3: Gradient evaluation took 4.8e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.48 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 2000 [0%] (Warmup)
Chain 3: Iteration: 200 / 2000 [10%] (Warmup)
Chain 3: Iteration: 400 / 2000 [20%] (Warmup)
Chain 3: Iteration: 600 / 2000 [30%] (Warmup)

```

Chain 3: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 9.492 seconds (Warm-up)
Chain 3:                2.001 seconds (Sampling)
Chain 3:                11.493 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 5.2e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.52 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 2000 [ 0%] (Warmup)
Chain 4: Iteration: 200 / 2000 [ 10%] (Warmup)
Chain 4: Iteration: 400 / 2000 [ 20%] (Warmup)
Chain 4: Iteration: 600 / 2000 [ 30%] (Warmup)
Chain 4: Iteration: 800 / 2000 [ 40%] (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%] (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 4.214 seconds (Warm-up)
Chain 4:                1.741 seconds (Sampling)
Chain 4:                5.955 seconds (Total)
Chain 4:

```

```

# random intercept + slope on study hours
model2 <- stan_lmer(
  average_score ~ weekly_self_study_hours

```

```

        + absence_days
        + extracurricular_activities
        + (1 + weekly_self_study_hours | career_aspiration),
data = df,
prior_intercept = normal(75, 10),
prior           = normal(0, 5),
prior_covariance = decov(regularization = 2),
chains = 4, iter = 2000, seed = 123
)

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 0.000104 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.04 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 2000 [0%] (Warmup)

Chain 1: Iteration: 200 / 2000 [10%] (Warmup)

Chain 1: Iteration: 400 / 2000 [20%] (Warmup)

Chain 1: Iteration: 600 / 2000 [30%] (Warmup)

Chain 1: Iteration: 800 / 2000 [40%] (Warmup)

Chain 1: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 1: Iteration: 1001 / 2000 [50%] (Sampling)

Chain 1: Iteration: 1200 / 2000 [60%] (Sampling)

Chain 1: Iteration: 1400 / 2000 [70%] (Sampling)

Chain 1: Iteration: 1600 / 2000 [80%] (Sampling)

Chain 1: Iteration: 1800 / 2000 [90%] (Sampling)

Chain 1: Iteration: 2000 / 2000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 23.954 seconds (Warm-up)

Chain 1: 5.913 seconds (Sampling)

Chain 1: 29.867 seconds (Total)

Chain 1:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 6.9e-05 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.69 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

```

Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%] (Warmup)
Chain 2: Iteration:   200 / 2000 [ 10%] (Warmup)
Chain 2: Iteration:   400 / 2000 [ 20%] (Warmup)
Chain 2: Iteration:   600 / 2000 [ 30%] (Warmup)
Chain 2: Iteration:   800 / 2000 [ 40%] (Warmup)
Chain 2: Iteration:  1000 / 2000 [ 50%] (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 2: Iteration:  1200 / 2000 [ 60%] (Sampling)
Chain 2: Iteration:  1400 / 2000 [ 70%] (Sampling)
Chain 2: Iteration:  1600 / 2000 [ 80%] (Sampling)
Chain 2: Iteration:  1800 / 2000 [ 90%] (Sampling)
Chain 2: Iteration:  2000 / 2000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 21.928 seconds (Warm-up)
Chain 2:                7.9 seconds (Sampling)
Chain 2:                29.828 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 6.8e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.68 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%] (Warmup)
Chain 3: Iteration:   200 / 2000 [ 10%] (Warmup)
Chain 3: Iteration:   400 / 2000 [ 20%] (Warmup)
Chain 3: Iteration:   600 / 2000 [ 30%] (Warmup)
Chain 3: Iteration:   800 / 2000 [ 40%] (Warmup)
Chain 3: Iteration:  1000 / 2000 [ 50%] (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%] (Sampling)
Chain 3: Iteration:  1200 / 2000 [ 60%] (Sampling)
Chain 3: Iteration:  1400 / 2000 [ 70%] (Sampling)
Chain 3: Iteration:  1600 / 2000 [ 80%] (Sampling)
Chain 3: Iteration:  1800 / 2000 [ 90%] (Sampling)
Chain 3: Iteration:  2000 / 2000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 22.113 seconds (Warm-up)
Chain 3:                6.755 seconds (Sampling)
Chain 3:                28.868 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).

Chain 4:

Chain 4: Gradient evaluation took 6.9e-05 seconds

Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.69 seconds.

Chain 4: Adjust your expectations accordingly!

Chain 4:

Chain 4:

Chain 4: Iteration: 1 / 2000 [0%] (Warmup)

Chain 4: Iteration: 200 / 2000 [10%] (Warmup)

Chain 4: Iteration: 400 / 2000 [20%] (Warmup)

Chain 4: Iteration: 600 / 2000 [30%] (Warmup)

Chain 4: Iteration: 800 / 2000 [40%] (Warmup)

Chain 4: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 4: Iteration: 1001 / 2000 [50%] (Sampling)

Chain 4: Iteration: 1200 / 2000 [60%] (Sampling)

Chain 4: Iteration: 1400 / 2000 [70%] (Sampling)

Chain 4: Iteration: 1600 / 2000 [80%] (Sampling)

Chain 4: Iteration: 1800 / 2000 [90%] (Sampling)

Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)

Chain 4:

Chain 4: Elapsed Time: 662.01 seconds (Warm-up)

Chain 4: 7.682 seconds (Sampling)

Chain 4: 669.692 seconds (Total)

Chain 4:

```
print(model1, digits = 2)
```

```
stan_lmer
```

```
family:      gaussian [identity]
```

```
formula:      average_score ~ weekly_self_study_hours + absence_days + extracurricular_acti  
            (1 | career_aspiration)
```

```
observations: 2000
```

```
-----
```

	Median	MAD_SD
(Intercept)	79.18	0.68
weekly_self_study_hours	0.14	0.02
absence_days	0.05	0.05
extracurricular_activities1	-0.09	0.27

```
Auxiliary parameter(s):
```

```
Median MAD_SD
```

sigma 4.74 0.08

Error terms:

Groups	Name	Std.Dev.
career_aspiration	(Intercept)	2.56
Residual		4.74

Num. levels: career_aspiration 17

* For help interpreting the printed output see ?print.stanreg

* For info on the priors used see ?prior_summary.stanreg

```
print(model2, digits = 2)
```

stan_lmer

family: gaussian [identity]
formula: average_score ~ weekly_self_study_hours + absence_days + extracurricular_acti
(1 + weekly_self_study_hours | career_aspiration)
observations: 2000

	Median	MAD_SD
(Intercept)	79.03	0.58
weekly_self_study_hours	0.14	0.03
absence_days	0.04	0.05
extracurricular_activities1	-0.09	0.27

Auxiliary parameter(s):

	Median	MAD_SD
sigma	4.72	0.07

Error terms:

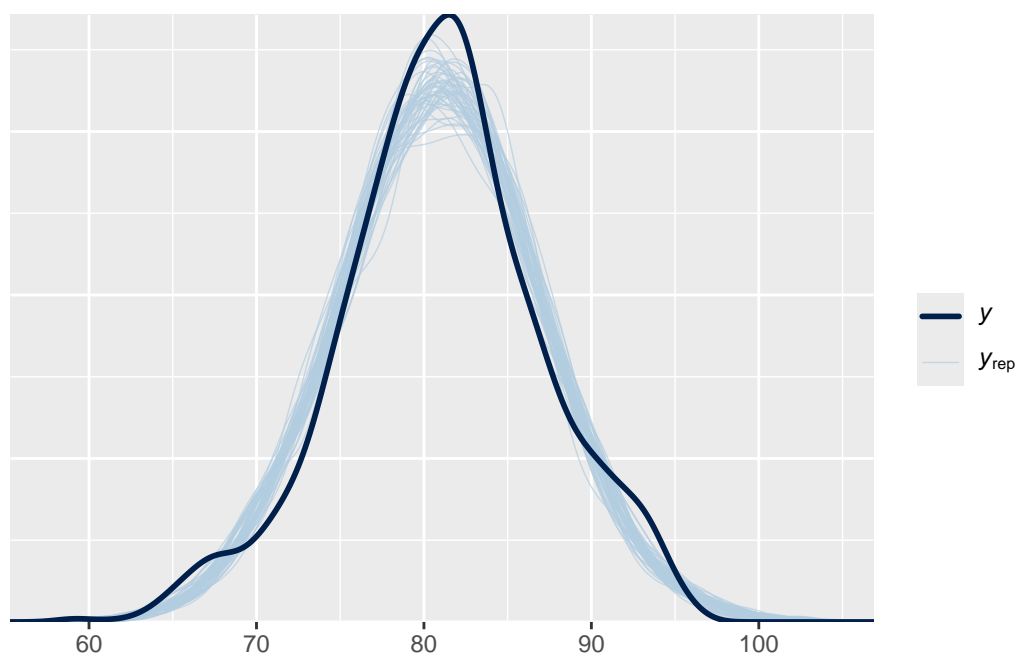
Groups	Name	Std.Dev.	Corr
career_aspiration	(Intercept)	1.9058	
	weekly_self_study_hours	0.0912	0.11
Residual		4.7212	

Num. levels: career_aspiration 17

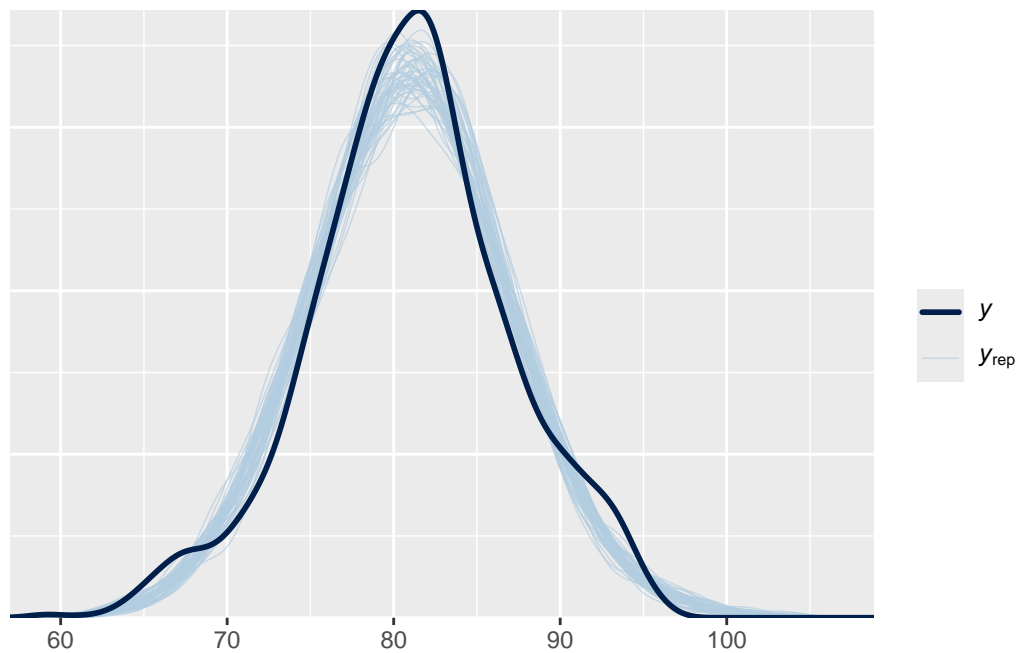
* For help interpreting the printed output see ?print.stanreg

* For info on the priors used see ?prior_summary.stanreg

```
#Posterior predictive checks  
pp_check(model1)
```



```
pp_check(model2)
```



```
# Compare with LOO
loo1 <- loo(model1)
loo2 <- loo(model2)
loo_compare(loo1, loo2)
```

	elpd_diff	se_diff
model2	0.0	0.0
model1	-4.9	2.9