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 becom
```

```
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 defa
- 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 'co
```

```
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 0.000186 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.86 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: 6.253 seconds (Warm-up)
Chain 1:                3.518 seconds (Sampling)
Chain 1:                9.771 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: 7.564 seconds (Warm-up)
Chain 2:                3.588 seconds (Sampling)
Chain 2:                11.152 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.2e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.12 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: 8.887 seconds (Warm-up)
Chain 3:                7.01 seconds (Sampling)
Chain 3:                15.897 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.4e-05 seconds
```

```
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.14 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: 9.966 seconds (Warm-up)
Chain 4:                6.939 seconds (Sampling)
Chain 4:                16.905 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
```

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

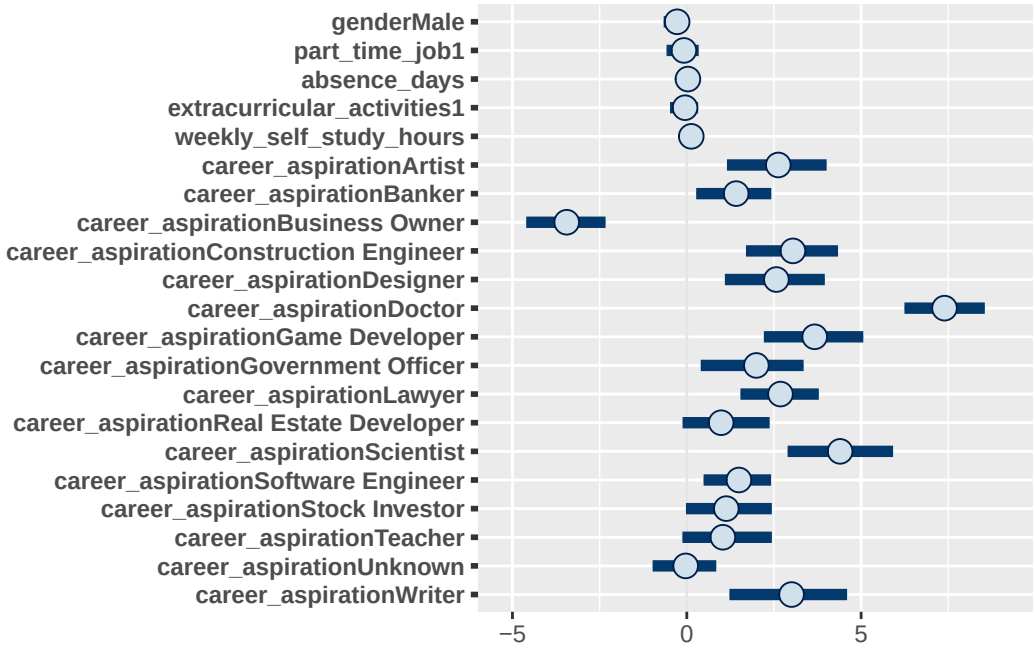
```
stan_glm
 family:       gaussian [identity]
 formula:      average_score ~ .
 observations: 2000
 predictors:   22
------
                                    Median MAD_SD
(Intercept)                         77.59   0.68
genderMale                          -0.28   0.24
part_time_job1                      -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
```

```r
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