

Covariate Selection - Projection Prediction

Data Preprocessing

```
# load data
apes1 <- read_csv("../data/laac_data_trial.csv")

##
## -- Column specification -----
## cols(
##   .default = col_double(),
##   group = col_character(),
##   subject = col_character(),
##   heat = col_character(),
##   test_day = col_character(),
##   observer = col_character(),
##   task = col_character(),
##   sex = col_character(),
##   rearing = col_character(),
##   le_present = col_character(),
##   dist_present = col_character()
## )
## i Use 'spec()' for the full column specifications.

apes2 <- read_csv("../data/laac_data_task.csv")

##
## -- Column specification -----
## cols(
##   time_point = col_double(),
##   group = col_character(),
##   subject = col_character(),
##   rank = col_double(),
##   task = col_character(),
##   performance = col_double(),
##   sex = col_character(),
##   rearing = col_character(),
##   age = col_double(),
##   time_in_leipzig = col_double(),
##   le_present = col_character(),
##   le_mean = col_double(),
##   le_max = col_double(),
##   dist_present = col_character(),
##   dist_mean = col_double(),
##   dist_max = col_double(),
```

```
##   time_outdoors = col_double()
## )
```

```
fn0 <- function(x, ...) {
  # helper function
  # sum over correct choice variable (code)
  to_return = tibble(cogn = sum(x$code))
  return(to_return)
}

code_sum <- apes1 %>%
  # contains summed code variable [for each task, time point, session and subject]
  group_by(time_point, session, subject, task) %>%
  group_modify(fn0)

apes1_tmp <- apes1 %>%
  # helper for merging
  select(-c(date, trial_session, trial_time_point, code)) %>%
  unique(by = c("time_point", "session", "subject"))

apes1_new <-
  as_tibble(merge(apes1_tmp, code_sum, by = c("time_point", "session", "subject", "task"))) %>%
  mutate(across(c(subject, group, heat, test_day, le_present, dist_present, sex, rearing, observ
  mutate(observer = fct_relevel(observer, "no")) %>%
  jtools::center(.,vars = c("sick_severity",
                           "le_mean",
                           "time_outdoors",
                           "age",
                           "time_in_leipzig")) %>%
  group_by(group, time_point) %>%
  mutate(rank_gmc = rank - mean(rank, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(time_point)

grp_size <- tibble(
  # number of apes for each species
  a_chimp = 20,
  b_chimp = 6,
  bonobo = 12,
  gorilla = 6,
  orangutan = 6
)

apes1_new <- apes1_new %>%
  # create rank variable depending on species
  group_by(group, time_point) %>%
  mutate(
```

```

    rel_rank = case_when(
      group == "a_chimp" ~ percent_rank(grp_size$a_chimp:1)[rank],
      group == "b_chimp" ~ percent_rank(grp_size$b_chimp:1)[rank],
      group == "bonobo" ~ percent_rank(grp_size$bonobo:1)[rank],
      group == "gorilla" ~ percent_rank(grp_size$gorilla:1)[rank],
      group == "orangutan" ~ percent_rank(grp_size$orangutan:1)[rank]
    )
  ) %>%
  ungroup()

apes1_new <- apes1_new %>%
  # create coding for heat variable
  mutate(heat_mod = case_when(
    sex == "f" & heat == "yes" ~ "_f_fheat",
    sex == "m" & heat == "yes" ~ "_m_fheat",
    sex == "f" & heat == "no" ~ "_f_noheat",
    sex == "m" & heat == "no" ~ "_m_noheat"),
    heat_mod = as_factor(heat_mod)
  ) %>%
  mutate(heat_mod = fct_relevel(heat_mod, "_f_noheat"))

apes1_new <- apes1_new %>%
  select(-heat, -heat_mod)

apes1_new <- apes1_new %>%
  # recode rearing categories: hand -> unknown
  mutate(rearing = fct_recode(rearing, "hand" = "unknown"))

t_cau <- filter(apes1_new, task == "causality")
t_inf <- filter(apes1_new, task == "inference")
t_quant <- filter(apes1_new, task == "quantity")
t_gaze <- filter(apes1_new, task == "gaze_following")

t_gaze <- t_gaze %>%
  # create dummy variable indicating if session 1 or 2
  group_by(time_point, session) %>%
  mutate(tp_mod = cur_group_id()) %>%
  ungroup() %>%
  mutate(day2 = case_when(session == 1 ~ "no",
                           session == 2 ~ "yes"),
    day2 = factor(day2)) %>%
  select(tp_mod, day2, everything())

t_gaze <- t_gaze %>%
  group_by(subject) %>%
  filter(!duplicated(tp_mod)) %>% # remove duplicates
  ungroup()

```

Covariate Selection

```
# covariate needed for projection prediction
# placed here to compare with formula
all_fixed_effects <- c("sick_severity",
                      "test_day", "test_tp",
                      "rel_rank",
                      "observer",
                      "age", "time_in_leipzig",
                      "sex",
                      "rearing",
                      "le_mean",
                      "dist_mean",
                      "time_outdoors",
                      "sociality")

fm <- formula(cogn ~ sick_severity +
              test_day + test_tp +
              rel_rank + # rank_gmc +
              observer +
              age + time_in_leipzig +
              sex +
              rearing +
              le_mean + # le_max + # le_present +
              dist_mean + # dist_max + # + dist_present +
              time_outdoors +
              sociality + # sociality_total
              # heat_mod + # heat +
              (1|subject)
              )

fm_gaze <- update(fm, . ~ . +day2)
```

Reference Model: 2-level Multilevel Model (random intercepts only)

```
m_cau_2l <- brm(fm, data = t_cau,
                warmup = 1e3, iter = 3e3, cores = ncores, chains = 2,
                seed = 2021,
                save_pars = save_pars(all = TRUE)
                )
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
m_inf_2l <- brm(fm, data = t_inf,
               warmup = 1e3, iter = 3e3, cores = ncores, chains = 2,
               seed = 2021,
               save_pars = save_pars(all = TRUE)
               )
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
m_quant_2l <- brm(fm, data = t_quant,
                 warmup = 1e3, iter = 3e3, cores = ncores, chains = 2,
                 seed = 2021,
                 save_pars = save_pars(all = TRUE)
                 )
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
m_gaze_2l <- brm(fm_gaze, data = t_gaze,
                 warmup = 1e3, iter = 3e3, cores = ncores, chains = 2,
                 seed = 2021,
                 save_pars = save_pars(all = TRUE)
                 )
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

```
## Start sampling
```

```
summary(m_cau_2l)
```

```
## Family: gaussian
```

```
## Links: mu = identity; sigma = identity
```

```
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leip
```

```
## Data: t_cau (Number of observations: 450)
```

```
## Samples: 2 chains, each with iter = 3000; warmup = 1000; thin = 1;
```

```
## total post-warmup samples = 4000
```

```
##
```

```
## Group-Level Effects:
```

```
## ~subject (Number of levels: 41)
```

```
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept) 2.52 0.33 1.94 3.23 1.00 1023 2148
```

```
##
```

```
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      8.21      1.34    5.58    10.90 1.00     1815     2173
## sick_severity   0.24      0.20   -0.14     0.63 1.00     3562     3151
## test_dayyes     0.47      1.10   -1.65     2.60 1.00     4816     3275
## test_tp         0.01      0.04   -0.07     0.08 1.00     4988     2921
## rel_rank       -0.20      1.38   -2.90     2.53 1.00     2130     2234
## observeryes     0.29      0.33   -0.38     0.95 1.00     3696     3010
## age            0.01      0.05   -0.08     0.12 1.00     1275     2153
## time_in_leipzig 0.08      0.07   -0.05     0.22 1.00     1562     2110
## sexf            0.01      0.95   -1.88     1.84 1.00     1300     1878
## rearinghand    -0.11      1.34   -2.76     2.68 1.00     1461     2253
## le_mean        -0.74      0.54   -1.79     0.31 1.00     5006     2798
## dist_mean      -0.41      0.22   -0.84     0.02 1.00     4516     3010
## time_outdoors  -0.06      0.06   -0.18     0.06 1.00     4207     3031
## sociality       0.09      0.24   -0.37     0.57 1.00     3991     2748
##
## Family Specific Parameters:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma        2.22      0.08    2.08    2.39 1.00     4446     2859
##
## 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).
```

```
summary(m_inf_2l)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leip
## Data: t_inf (Number of observations: 451)
## Samples: 2 chains, each with iter = 3000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    2.70      0.35    2.12    3.48 1.00     1051     1852
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept        5.33      1.30    2.82    7.89 1.00     1658     2153
## sick_severity    -0.12      0.17   -0.44     0.20 1.00     3693     3003
## test_dayyes      0.91      0.94   -0.96     2.72 1.00     4556     2981
## test_tp         -0.09      0.03   -0.16    -0.02 1.00     4363     3314
## rel_rank         0.06      1.27   -2.39     2.62 1.00     1932     2293
## observeryes      0.47      0.29   -0.10     1.04 1.00     4323     2572
## age            -0.06      0.05   -0.16     0.04 1.00     1584     1945
```

```
## time_in_leipzig      0.29      0.08      0.14      0.44 1.00      1461      1982
## sexf                 0.48      1.02     -1.49      2.51 1.00      1466      1835
## rearinghand         -0.22      1.37     -2.88      2.43 1.00      1579      1965
## le_mean             -0.56      0.45     -1.44      0.34 1.00      4284      3312
## dist_mean           -0.10      0.18     -0.46      0.25 1.00      4367      3287
## time_outdoors       -0.02      0.05     -0.11      0.07 1.00      4151      2983
## sociality            0.03      0.20     -0.36      0.42 1.00      4439      2822
```

```
##
```

```
## Family Specific Parameters:
```

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sigma      1.86      0.07      1.74      2.00 1.00      4120      2638
```

```
##
```

```
## 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).
```

```
summary(m_quant_2l)
```

```
## Family: gaussian
```

```
## Links: mu = identity; sigma = identity
```

```
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leip
```

```
## Data: t_quant (Number of observations: 421)
```

```
## Samples: 2 chains, each with iter = 3000; warmup = 1000; thin = 1;
```

```
##      total post-warmup samples = 4000
```

```
##
```

```
## Group-Level Effects:
```

```
## ~subject (Number of levels: 41)
```

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept)      1.96      0.27      1.52      2.55 1.00      1412      2286
```

```
##
```

```
## Population-Level Effects:
```

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## Intercept          7.94      1.08      5.88     10.10 1.00      2412      2407
```

```
## sick_severity      -0.31      0.16     -0.62     -0.01 1.00      3577      2777
```

```
## test_dayyes         0.27      0.82     -1.35      1.86 1.00      4487      3120
```

```
## test_tp             0.02      0.03     -0.04      0.09 1.00      4735      3002
```

```
## rel_rank            0.55      1.08     -1.57      2.61 1.00      2749      2738
```

```
## observeryes        -0.30      0.30     -0.89      0.29 1.00      4451      2564
```

```
## age                -0.01      0.04     -0.10      0.07 1.00      2029      2138
```

```
## time_in_leipzig     0.13      0.06      0.02      0.25 1.00      2149      2334
```

```
## sexf                0.64      0.76     -0.84      2.12 1.00      2166      2614
```

```
## rearinghand        -1.85      1.07     -3.99      0.20 1.00      1914      1986
```

```
## le_mean             0.15      0.44     -0.71      1.00 1.00      5682      3063
```

```
## dist_mean          -0.11      0.19     -0.46      0.27 1.00      4580      3370
```

```
## time_outdoors       -0.00      0.05     -0.10      0.10 1.00      3910      2872
```

```
## sociality          -0.02      0.21     -0.43      0.40 1.00      4563      2886
```

```
##
```

```
## Family Specific Parameters:
```

```
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      1.76      0.07      1.64      1.90 1.00      4337      2257
##
## 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).
```

```
summary(m_gaze_2l)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leip
## Data: t_gaze (Number of observations: 879)
## Samples: 2 chains, each with iter = 3000; warmup = 1000; thin = 1;
##           total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)      0.76      0.10      0.60      0.98 1.00      1388      1801
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           1.00      0.39      0.23      1.78 1.00      2022      2269
## sick_severity       -0.14      0.05     -0.24     -0.03 1.00      4955      2709
## test_dayyes         0.24      0.22     -0.20      0.68 1.00      5180      3047
## test_tp             0.01      0.01     -0.01      0.02 1.00      5968      2920
## rel_rank            -0.08      0.36     -0.79      0.61 1.00      3002      2933
## observeryes        -0.13      0.08     -0.28      0.02 1.00      4972      2944
## age                 0.02      0.02     -0.01      0.06 1.00      1765      1926
## time_in_leipzig    -0.01      0.02     -0.05      0.03 1.00      2002      2063
## sexf                0.21      0.29     -0.36      0.80 1.00      1739      1521
## rearinghand        -0.72      0.40     -1.52      0.07 1.00      1789      1853
## le_mean            -0.04      0.12     -0.27      0.21 1.00      6071      2794
## dist_mean          -0.10      0.05     -0.19     -0.00 1.00      4391      3317
## time_outdoors       0.09      0.01      0.07      0.12 1.00      4399      3201
## sociality          -0.14      0.06     -0.25     -0.03 1.00      5674      2760
## day2yes            -0.12      0.05     -0.21     -0.03 1.00      5812      2800
##
## Family Specific Parameters:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      0.71      0.02      0.68      0.74 1.00      5416      2861
##
## 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).
```


Predictive Projection

```
# delay random intercept to last place so that it doesn't soak up all the variance
s_terms <- c("1", all_fixed_effects,
            paste0(paste(all_fixed_effects, collapse = " + "), " + (1 | subject)"))
s_terms_gaze <- c("1", c(all_fixed_effects, "day2"),
                paste0(paste(c(all_fixed_effects, "day2"), collapse = " + "), " + (1 | subject)
```

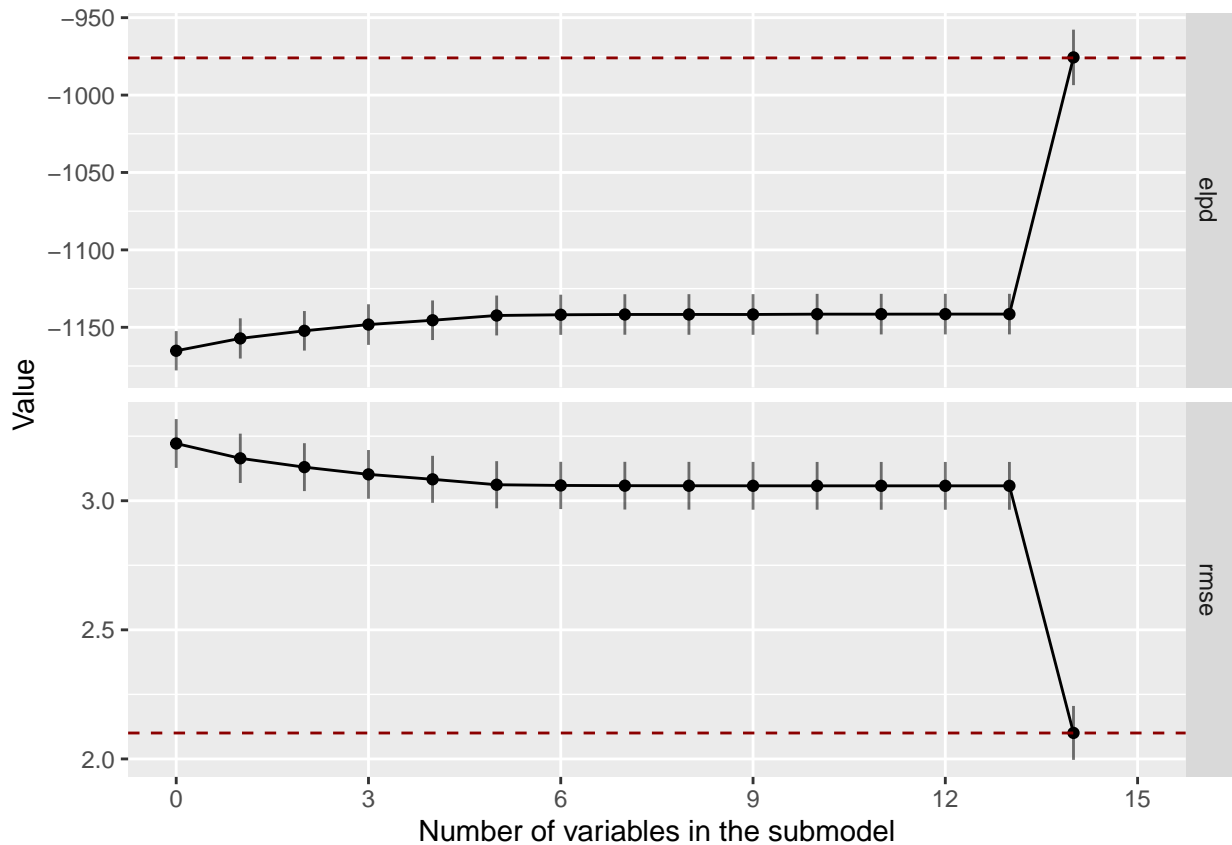
```
refM_cau <- get_refmodel(m_cau_2l)
refM_inf <- get_refmodel(m_inf_2l)
refM_quant <- get_refmodel(m_quant_2l)
refM_gaze <- get_refmodel(m_gaze_2l)
```

```
vs_cau <- varsel(refM_cau, search_terms = s_terms, ndraws_pred = 400)
```

```
## [1] "10% of terms selected."
## [1] "20% of terms selected."
## [1] "30% of terms selected."
## [1] "40% of terms selected."
## [1] "50% of terms selected."
## [1] "60% of terms selected."
## [1] "70% of terms selected."
## [1] "80% of terms selected."
## [1] "90% of terms selected."
## [1] "100% of terms selected."
```

```
summary(vs_cau); plot(vs_cau, stats = c('elpd', 'rmse'))
```

```
##    size  solution_terms      elpd  elpd.se
## 2     0      <NA> -1165.1547 12.75987
## 3     1  sick_severity -1157.1903 13.06783
## 4     2    dist_mean -1152.2625 12.84686
## 5     3 time_in_leipzig -1148.2296 13.19233
## 6     4    sociality -1145.4489 12.85860
## 7     5     le_mean -1142.3717 12.95893
## 8     6  time_outdoors -1141.8980 13.04219
## 9     7     rearing -1141.7203 13.15962
## 10    8        sex -1141.6997 13.18919
## 11    9     test_tp -1141.7134 13.19616
## 12   10    test_day -1141.5093 13.18265
## 13   11    rel_rank -1141.5028 13.19056
## 14   12        age -1141.4820 13.18293
## 15   13    observer -1141.4668 13.18189
## 16   14  (1 | subject) -975.6632 18.02598
```

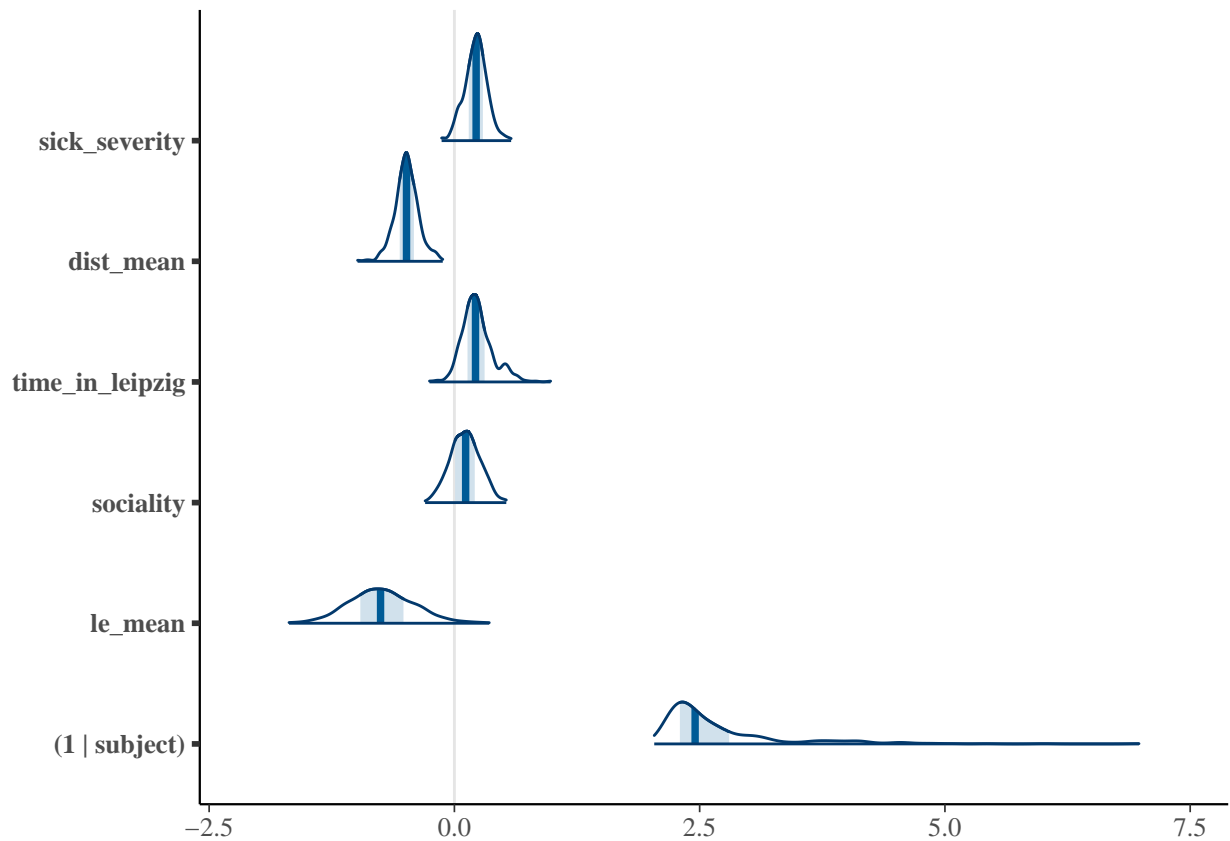


```
suggest_size(vs_cau)
```

```
## [1] 14
```

```
randint_ind_vscau <- length(solution_terms(vs_cau))
relevant_cov_vscau <- c(1, 2, 3, 4, 5, randint_ind_vscau)

proj_cau <- project(vs_cau, solution_terms = relevant_cov_vscau)
mcmc_areas(as.matrix(proj_cau), pars = solution_terms(vs_cau)[relevant_cov_vscau])
```



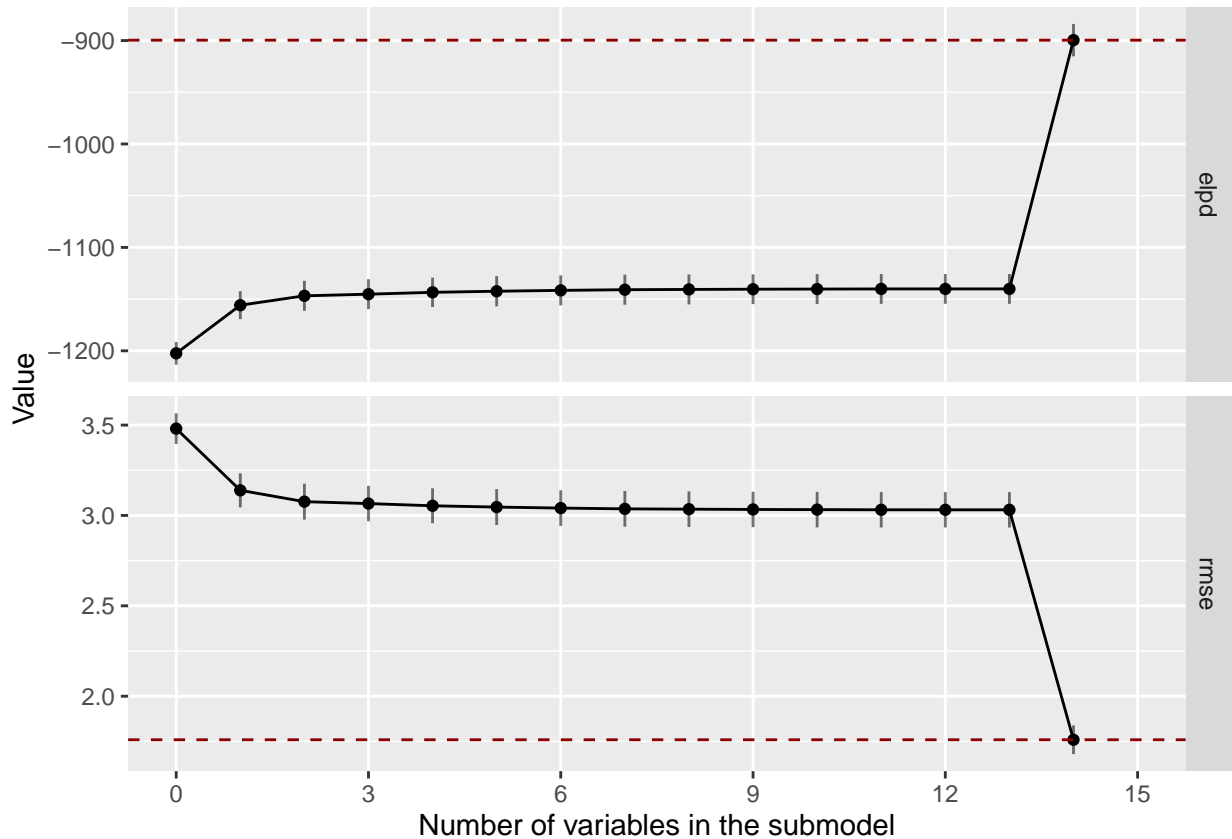
```
vs_inf <- varsel(refM_inf, search_terms = s_terms)
```

```
## [1] "10% of terms selected."
## [1] "20% of terms selected."
## [1] "30% of terms selected."
## [1] "40% of terms selected."
## [1] "50% of terms selected."
## [1] "60% of terms selected."
## [1] "70% of terms selected."
## [1] "80% of terms selected."
## [1] "90% of terms selected."
## [1] "100% of terms selected."
```

```
summary(vs_inf); plot(vs_inf, stats = c('elpd', 'rmse'))
```

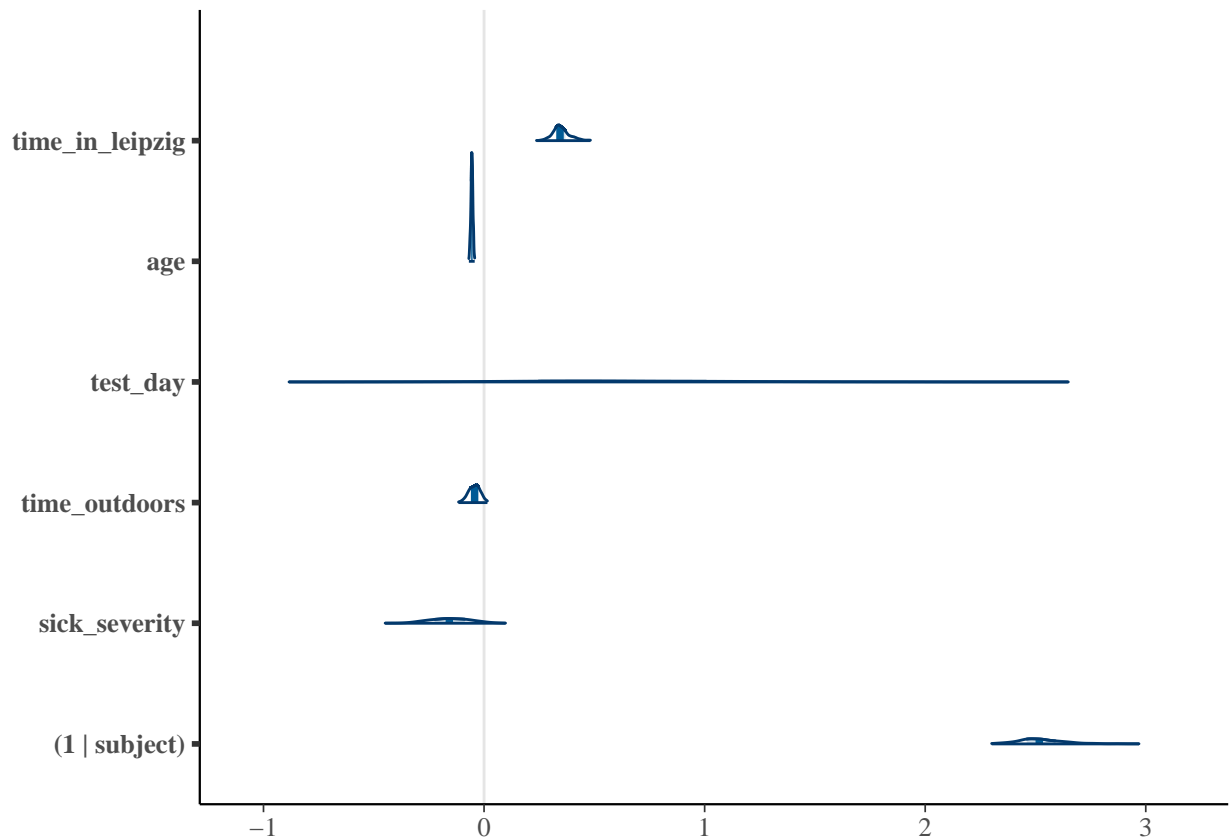
```
##   size solution_terms      elpd elpd.se
## 2    0             <NA> -1202.4266 11.02236
## 3    1 time_in_leipzig -1155.8835 13.59820
## 4    2              age -1146.7921 14.61305
## 5    3      test_day -1145.2303 14.51340
## 6    4 time_outdoors -1143.4169 14.28021
## 7    5 sick_severity -1142.3735 14.68275
```

```
## 8      6      rearing -1141.5402 14.58025
## 9      7      sex -1140.9073 14.62366
## 10     8      test_tp -1140.6369 14.59719
## 11     9      rel_rank -1140.4115 14.51587
## 12    10     sociality -1140.2778 14.53506
## 13    11     dist_mean -1140.1431 14.48880
## 14    12      le_mean -1140.1356 14.49207
## 15    13     observer -1140.1289 14.49431
## 16    14 (1 | subject) -899.5917 15.76802
```



```
randint_ind_vsinf <- length(solution_terms(vs_inf))
relevant_cov_vsinf <- c(1, 2, 3, 4, 5, randint_ind_vsinf)

proj_inf <- project(vs_inf, solution_terms = relevant_cov_vsinf)
mcmc_areas(as.matrix(proj_inf), pars = solution_terms(vs_inf)[relevant_cov_vsinf])
```



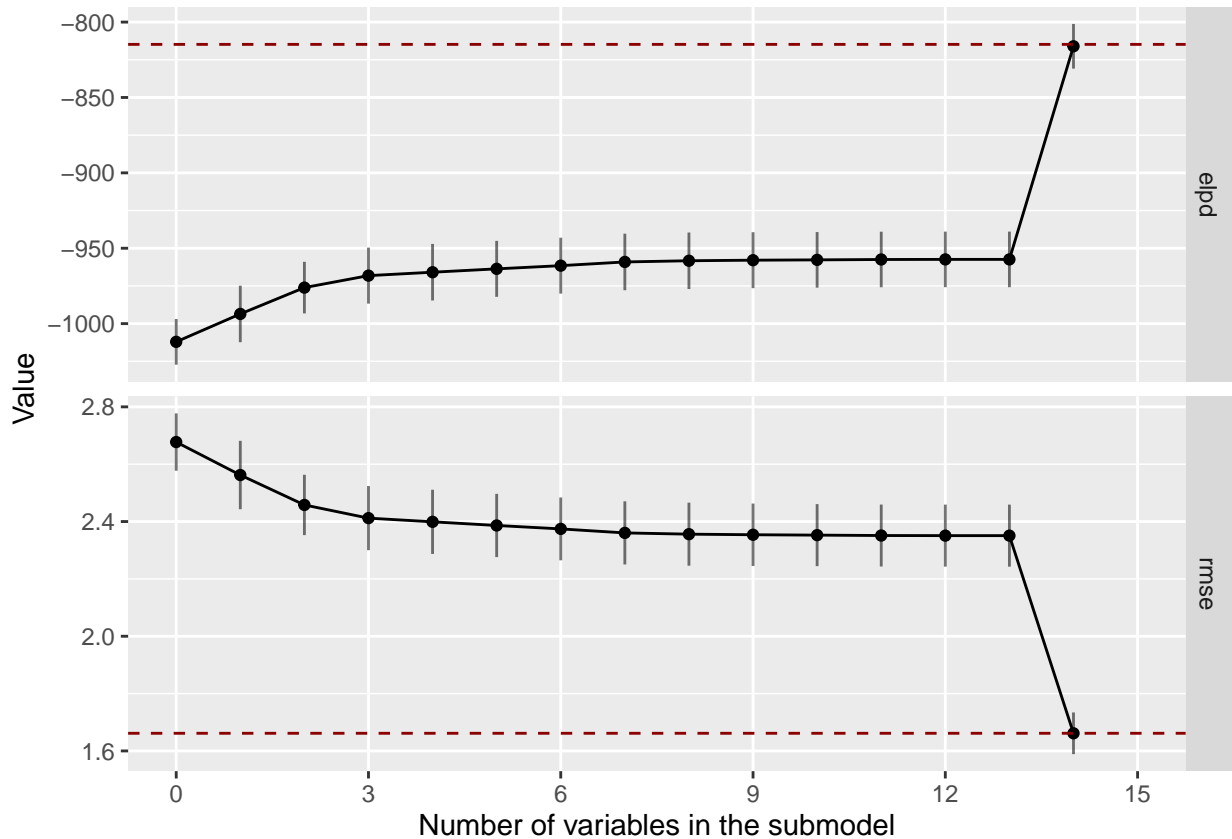
```
vs_quant <- varsel(refM_quant, search_terms = s_terms)
```

```
## [1] "10% of terms selected."
## [1] "20% of terms selected."
## [1] "30% of terms selected."
## [1] "40% of terms selected."
## [1] "50% of terms selected."
## [1] "60% of terms selected."
## [1] "70% of terms selected."
## [1] "80% of terms selected."
## [1] "90% of terms selected."
## [1] "100% of terms selected."
```

```
summary(vs_quant); plot(vs_quant, stats = c('elpd', 'rmse'))
```

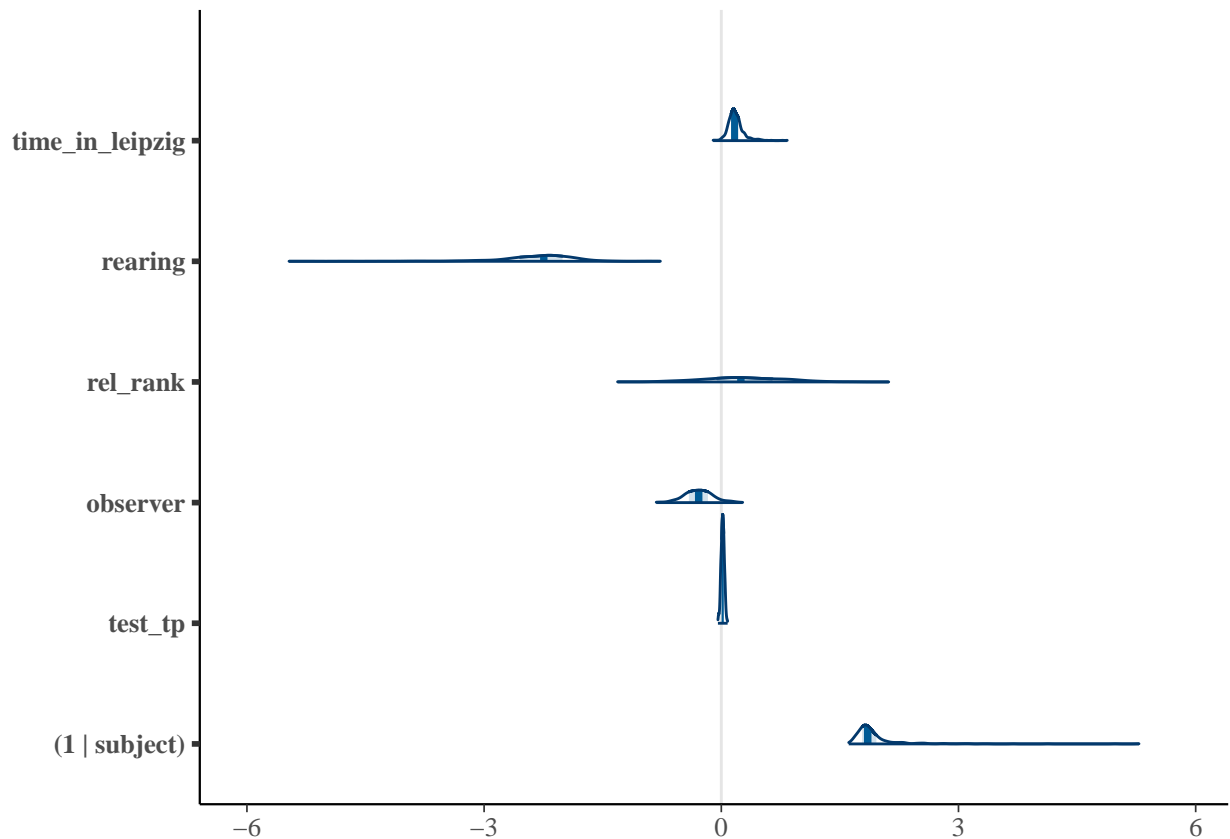
```
##   size solution_terms      elpd elpd.se
## 2    0              <NA> -1012.0705 15.24521
## 3    1 time_in_leipzig -993.5810 18.85258
## 4    2          rearing -976.1095 17.20583
## 5    3          rel_rank -968.1634 18.67238
## 6    4          observer -965.9048 18.85044
## 7    5           test_tp -963.6660 18.67117
```

```
## 8      6      sex -961.5605 18.64816
## 9      7  sick_severity -959.0900 18.86816
## 10     8    dist_mean -958.2960 18.85114
## 11     9      age -957.9304 18.66299
## 12    10  time_outdoors -957.7050 18.53919
## 13    11    le_mean -957.4614 18.52051
## 14    12    test_day -957.3991 18.53897
## 15    13    sociality -957.3966 18.54068
## 16    14  (1 | subject) -816.0449 14.90607
```



```
randint_ind_vsquant <- length(solution_terms(vs_quant))
relevant_cov_vsquant <- c(1, 2, 3, 4, 5, randint_ind_vsquant)

proj_quant <- project(vs_quant, solution_terms = relevant_cov_vsquant)
mcmc_areas(as.matrix(proj_quant), pars = solution_terms(vs_quant)[relevant_cov_vsquant])
```



```
vs_gaze <- varsel(refM_gaze, search_terms = s_terms_gaze)
```

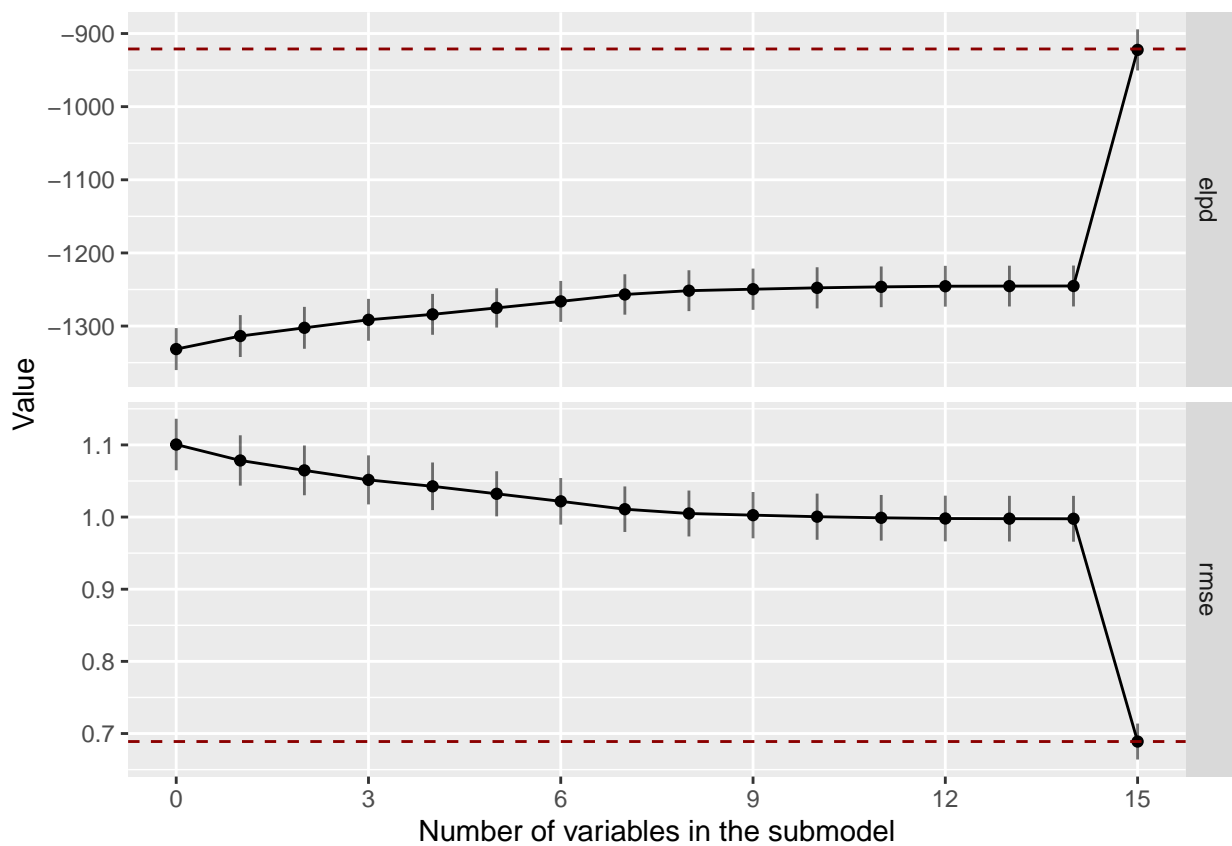
```
## [1] "10% of terms selected."
## [1] "20% of terms selected."
## [1] "30% of terms selected."
## [1] "40% of terms selected."
## [1] "50% of terms selected."
## [1] "60% of terms selected."
## [1] "70% of terms selected."
## [1] "80% of terms selected."
## [1] "90% of terms selected."
## [1] "100% of terms selected."
```

```
## Loading required namespace: optimx
```

```
summary(vs_gaze); plot(vs_gaze, stats = c('elpd', 'rmse'))
```

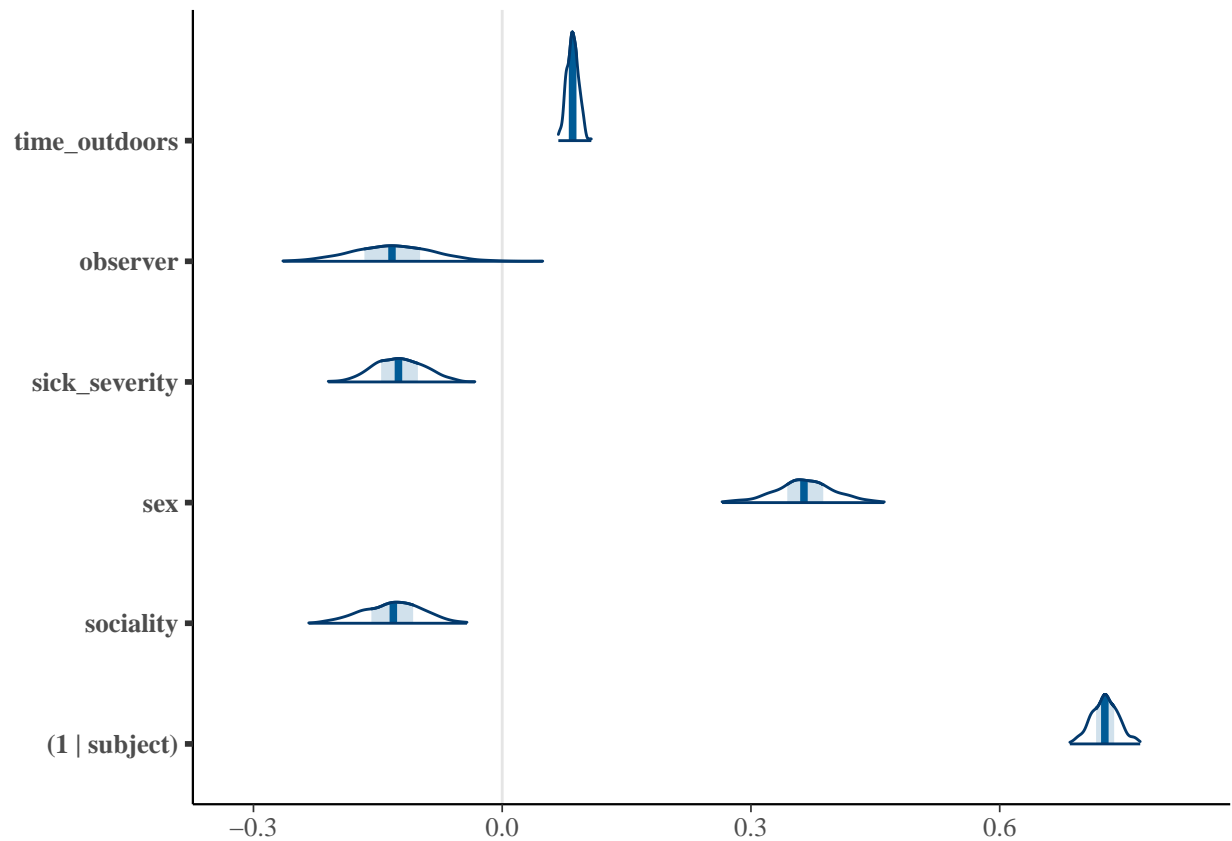
```
##   size solution_terms      elpd elpd.se
## 2    0              <NA> -1331.4733 28.85321
## 3    1  time_outdoors -1313.6480 28.80648
## 4    2      observer -1302.3842 28.81210
## 5    3  sick_severity -1291.4357 28.78066
```

```
## 6      4      sex -1283.9689 28.10549
## 7      5    sociality -1275.1735 27.00371
## 8      6      rearing -1266.2579 28.15469
## 9      7      age -1256.8043 27.76371
## 10     8 time_in_leipzig -1251.6163 28.10718
## 11     9      test_day -1249.5938 28.33698
## 12    10      day2 -1247.7317 28.30457
## 13    11     rel_rank -1246.3827 27.99107
## 14    12     test_tp -1245.5258 28.06011
## 15    13     le_mean -1245.3383 28.05591
## 16    14     dist_mean -1245.2042 28.15349
## 17    15 (1 | subject) -922.3862 28.29454
```



```
randint_ind_vsgaze <- length(solution_terms(vs_gaze))
relevant_cov_vsgaze <- c(1, 2, 3, 4, 5, randint_ind_vsgaze)

proj_gaze <- project(vs_gaze, solution_terms = relevant_cov_vsgaze)
mcmc_areas(as.matrix(proj_gaze), pars = solution_terms(vs_gaze)[relevant_cov_vsgaze])
```

Covariate Selection (cross validated)

```

cvs_cau <- cv_varsel(refM_cau, search_terms = s_terms, cv_method = "loo", method = "forward")

## Warning in cv_varsel.refmodel(refM_cau, search_terms = s_terms, cv_method =
## "loo", : K provided, but cv_method is L00.

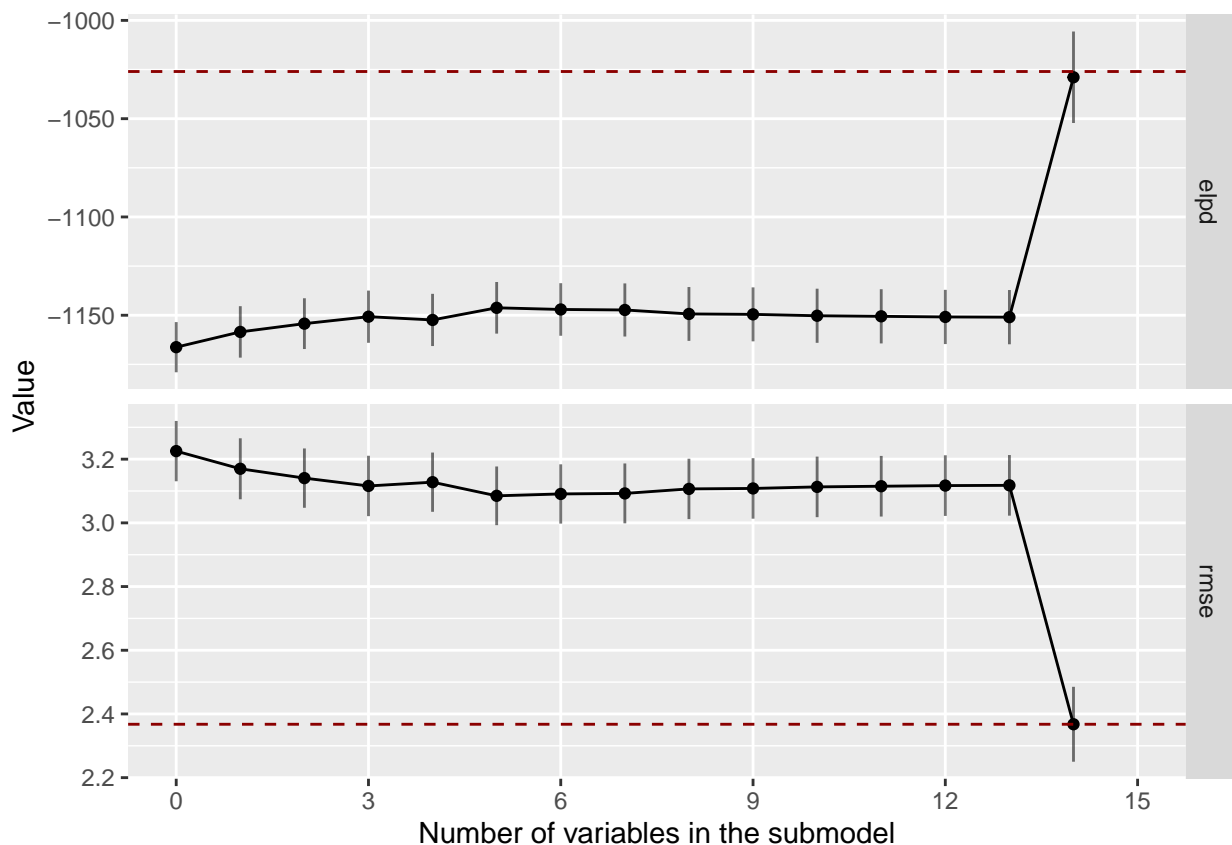
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for de

## [1] "Computing L00s..."
## |

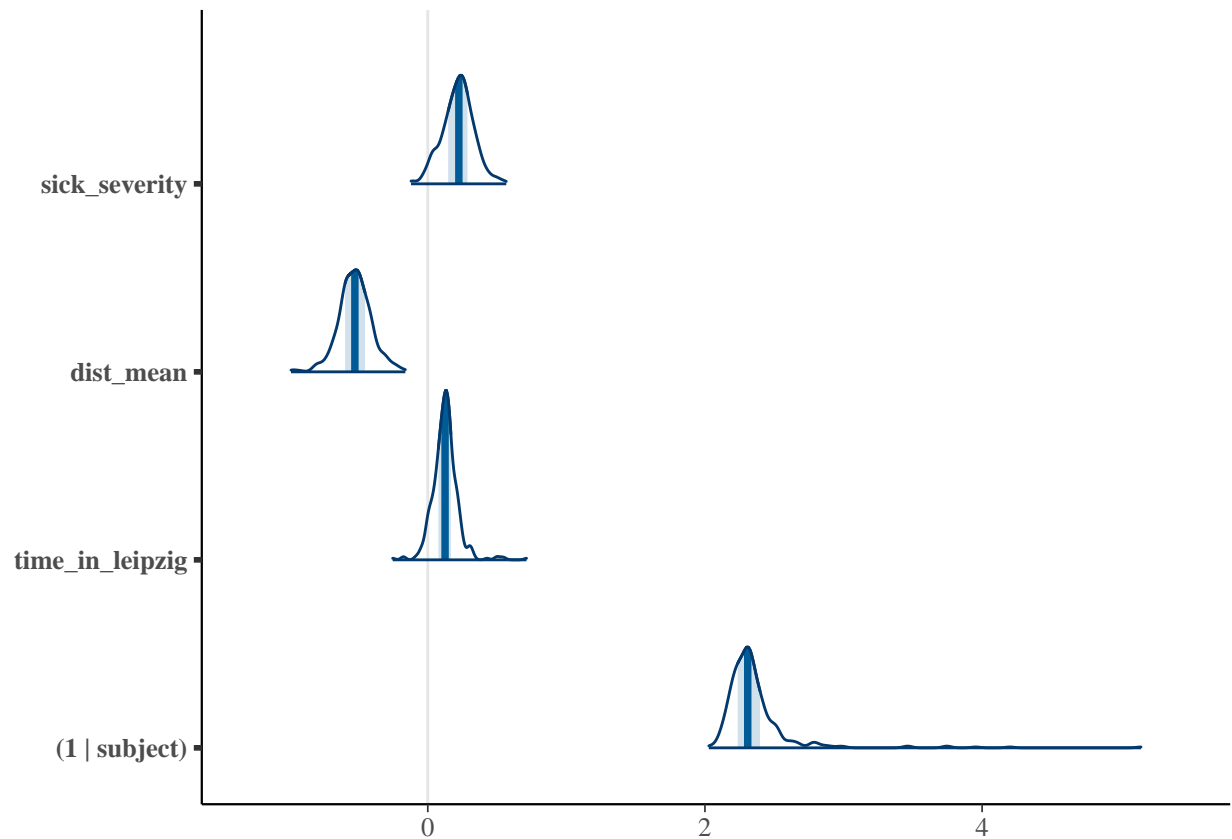
```

```
summary(cvs_cau); plot(cvs_cau, stats = c('elpd', 'rmse'))
```

##	size	solution_terms	elpd	elpd.se
## 2	0	<NA>	-1166.257	12.83275
## 3	1	sick_severity	-1158.487	13.16005
## 4	2	dist_mean	-1154.281	12.98930
## 5	3	time_in_leipzig	-1150.742	13.34983
## 6	4	sociality	-1152.396	13.35339
## 7	5	le_mean	-1146.224	13.20788
## 8	6	time_outdoors	-1147.065	13.44194
## 9	7	rearing	-1147.320	13.58690
## 10	8	sex	-1149.344	13.77470
## 11	9	test_tp	-1149.556	13.80771
## 12	10	test_day	-1150.263	13.84607
## 13	11	rel_rank	-1150.560	13.86582
## 14	12	age	-1150.868	13.87439
## 15	13	observer	-1150.976	13.87992
## 16	14	(1 subject)	-1028.931	23.40641



```
proj_cau_cv <- project(cvs_cau, solution_terms = c(1, 2, 3, 14))
mcmc_areas(as.matrix(proj_cau_cv), pars = solution_terms(cvs_cau)[c(1, 2, 3, 14)])
```



relevant covariates: (1 | subject), sick_severity, dist_mean, time_in_leipzig

```
cv_inf <- cv_varsel(refM_inf, search_terms = s_terms, cv_method = "loo", method = "forward")

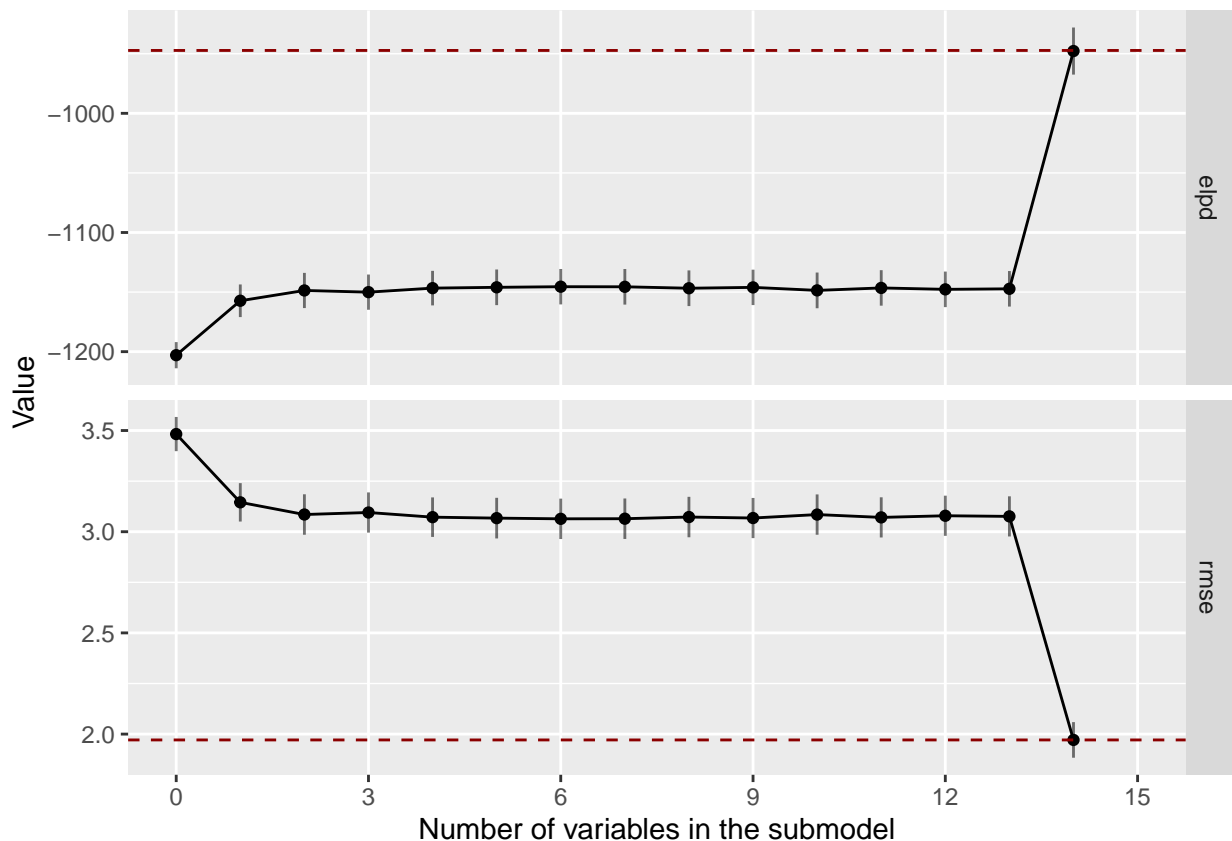
## Warning in cv_varsel.refmodel(refM_inf, search_terms = s_terms, cv_method =
## "loo", : K provided, but cv_method is LOO.

## Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic') f

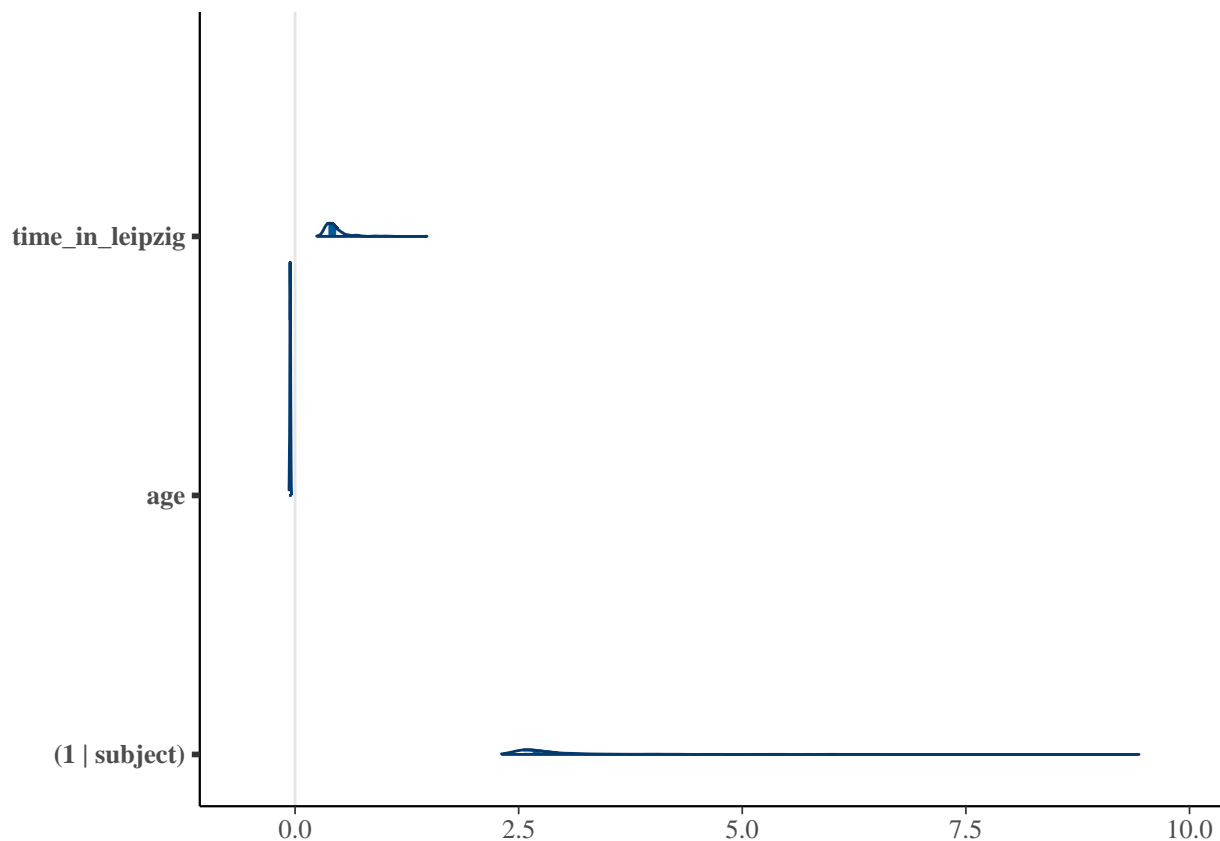
## [1] "Computing LOOs..."
## |
```

```
summary(cvs_inf); plot(cvs_inf, stats = c('elpd', 'rmse'))
```

##	size	solution_terms	elpd	elpd.se
## 2	0	<NA>	-1202.9573	11.04137
## 3	1	time_in_leipzig	-1157.2867	13.77772
## 4	2	age	-1148.6480	14.85128
## 5	3	test_day	-1150.0501	14.86731
## 6	4	time_outdoors	-1146.6846	14.58397
## 7	5	sick_severity	-1146.0034	15.04132
## 8	6	rearing	-1145.4893	14.94893
## 9	7	sex	-1145.5726	15.01614
## 10	8	test_tp	-1146.7745	15.05126
## 11	9	rel_rank	-1146.0588	14.94705
## 12	10	sociality	-1148.5757	15.07609
## 13	11	dist_mean	-1146.5146	14.97164
## 14	12	le_mean	-1147.7149	15.00879
## 15	13	observer	-1147.2558	14.99820
## 16	14	(1 subject)	-947.7434	19.87062



```
proj_inf_cv <- project(cvs_inf, solution_terms = c(1, 2, 14))
mcmc_areas(as.matrix(proj_inf_cv), pars = solution_terms(cvs_inf)[c(1, 2, 14)])
```



relevant covariates: (1 | subject), time_in_leipzig, age

```
cvs_quant <- cv_varsel(refM_quant, search_terms = s_terms, cv_method = "loo", method = "forward")
```

```
## Warning in cv_varsel.refmodel(refM_quant, search_terms = s_terms, cv_method =
## "loo", : K provided, but cv_method is LOO.
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for de
```

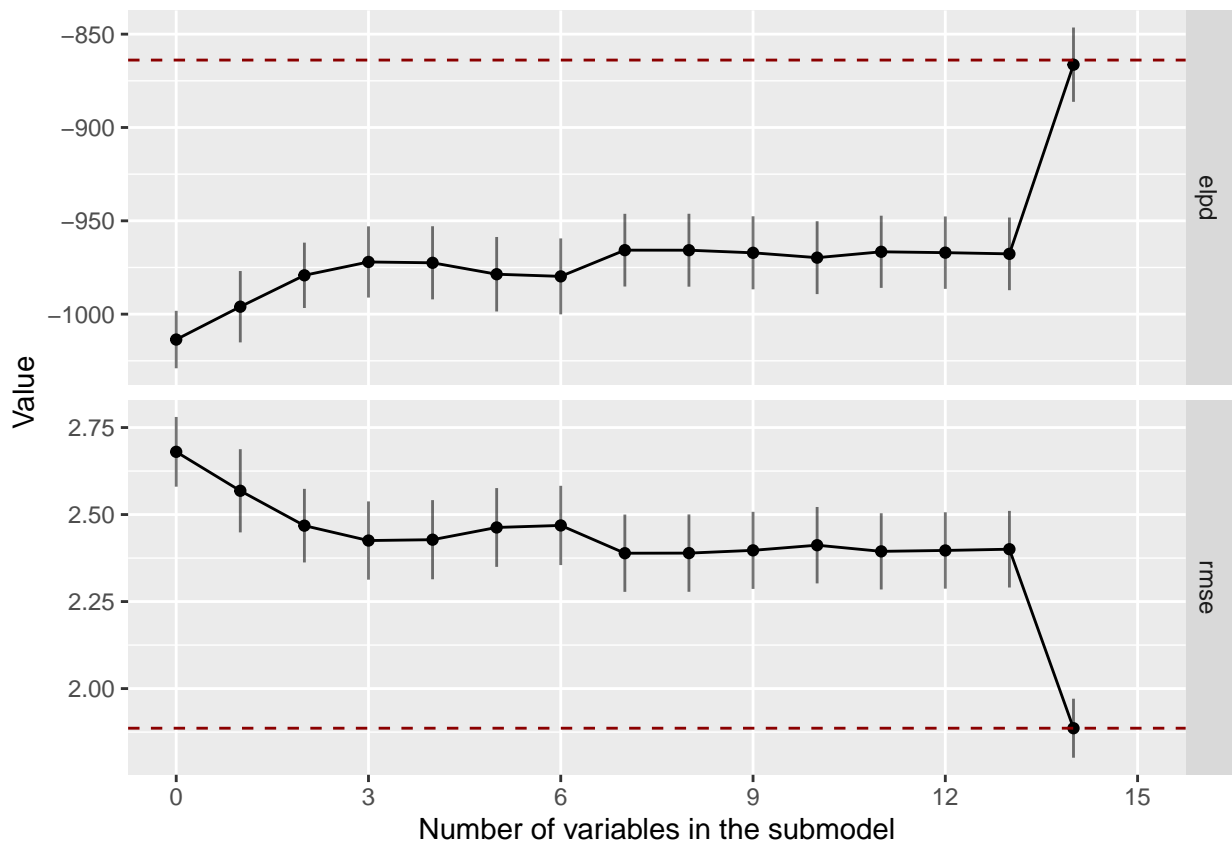
```
## [1] "Computing LOOs..."
```

```
## |
```

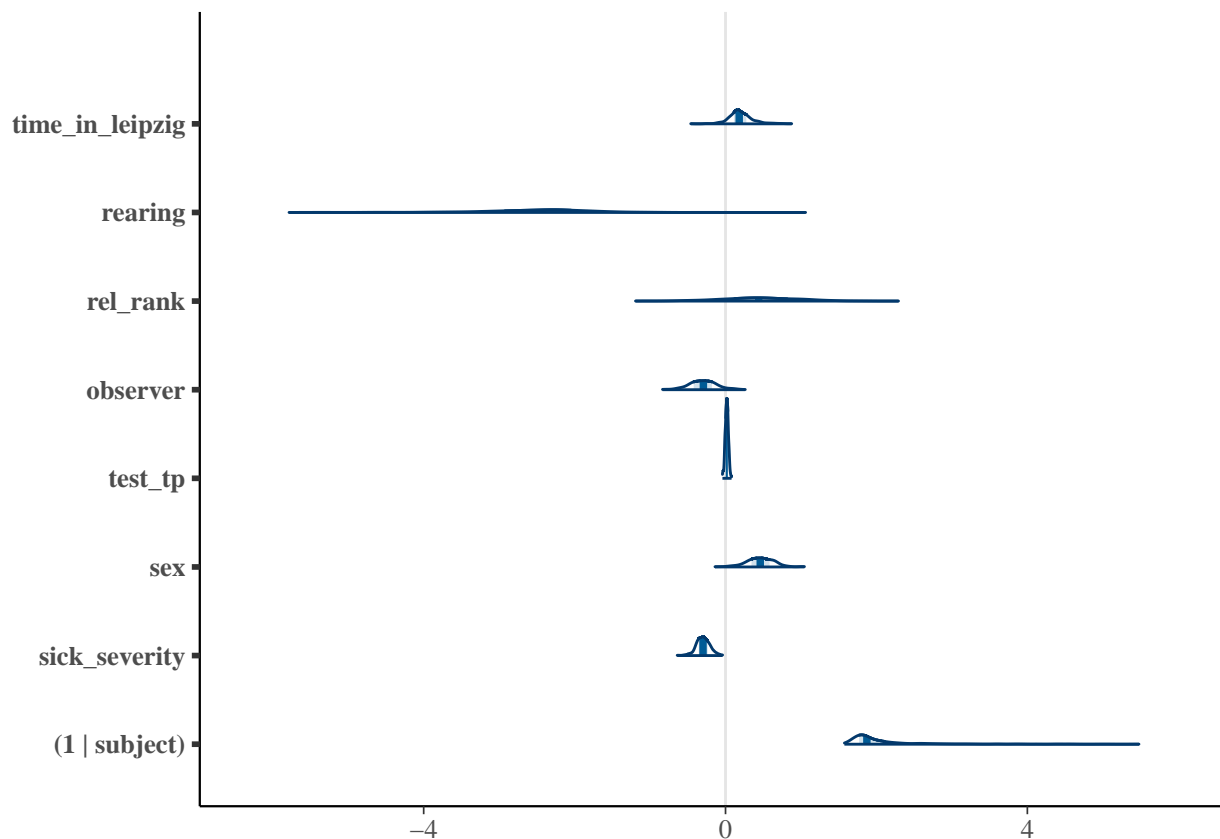
```
|
```

```
summary(cvs_quant); plot(cvs_quant, stats = c('elpd', 'rmse'))
```

##	size	solution_terms	elpd	elpd.se
## 2	0	<NA>	-1013.6006	15.47693
## 3	1	time_in_leipzig	-996.0081	19.21820
## 4	2	rearing	-979.1899	17.60931
## 5	3	rel_rank	-972.0382	19.18568
## 6	4	observer	-972.4871	19.70495
## 7	5	test_tp	-978.6112	20.06486
## 8	6	sex	-979.7609	20.45022
## 9	7	sick_severity	-965.7165	19.58618
## 10	8	dist_mean	-965.7454	19.64410
## 11	9	age	-967.1341	19.67638
## 12	10	time_outdoors	-969.7352	19.62518
## 13	11	le_mean	-966.5915	19.44065
## 14	12	test_day	-967.0479	19.49654
## 15	13	sociality	-967.6955	19.58039
## 16	14	(1 subject)	-866.3420	20.01568



```
proj_quant_cv <- project(cvs_quant, solution_terms = c(1, 2, 3, 4, 5, 6, 7, 14))
mcmc_areas(as.matrix(proj_quant_cv), pars = solution_terms(cvs_quant)[c(1, 2, 3, 4, 5, 6, 7, 14)])
```



relevant covariates: (1 | subject), time_in_leipzig, rearing, rel_rank, observer, test_tp, sex, sick_severity

```
cv_s_gaze <- cv_varsel(refM_gaze, search_terms = s_terms_gaze, cv_method = "loo", method = "forwa
```

```
## Warning in cv_varsel.refmodel(refM_gaze, search_terms = s_terms_gaze, cv_method
## = "loo", : K provided, but cv_method is L00.
```

```
## Warning: Some Pareto k diagnostic values are slightly high. See help('pareto-k-diagnostic') f
```

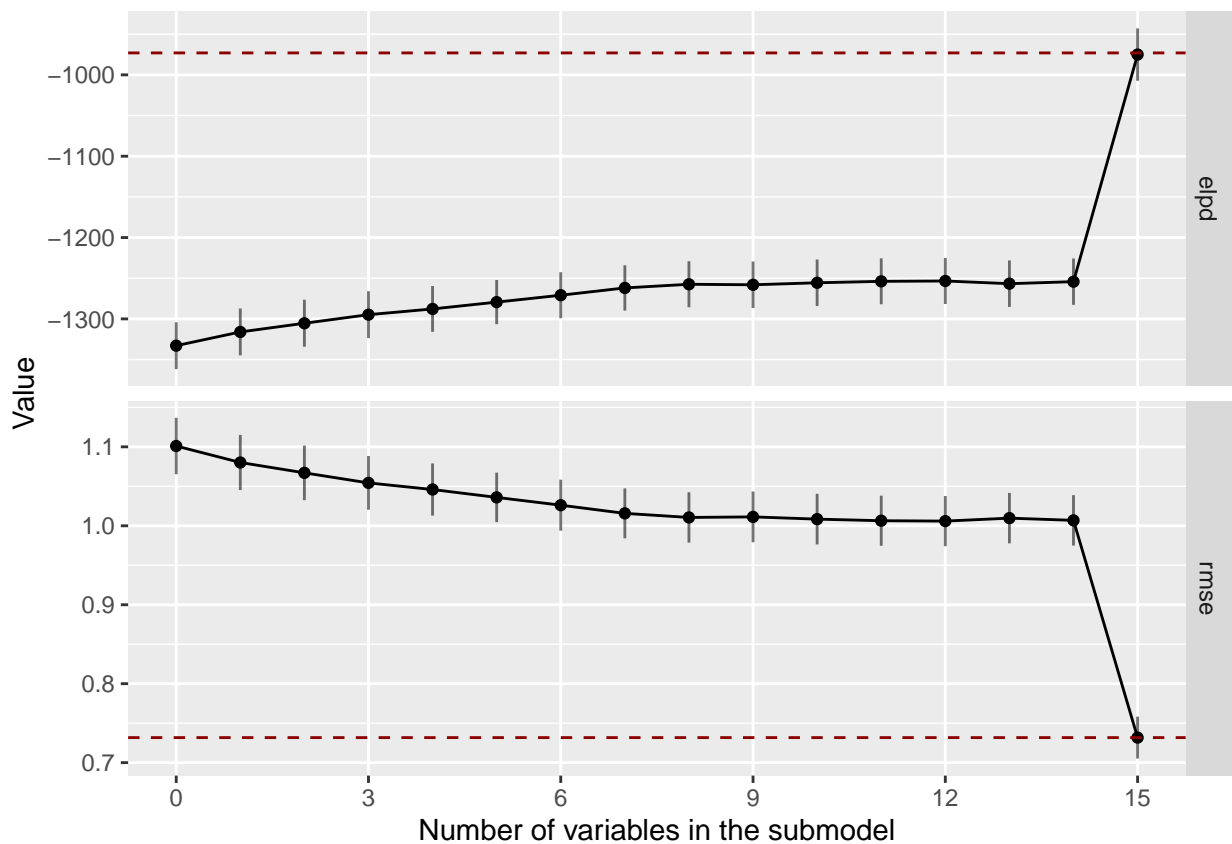
```
## [1] "Computing L00s..."
```

```
## |
```

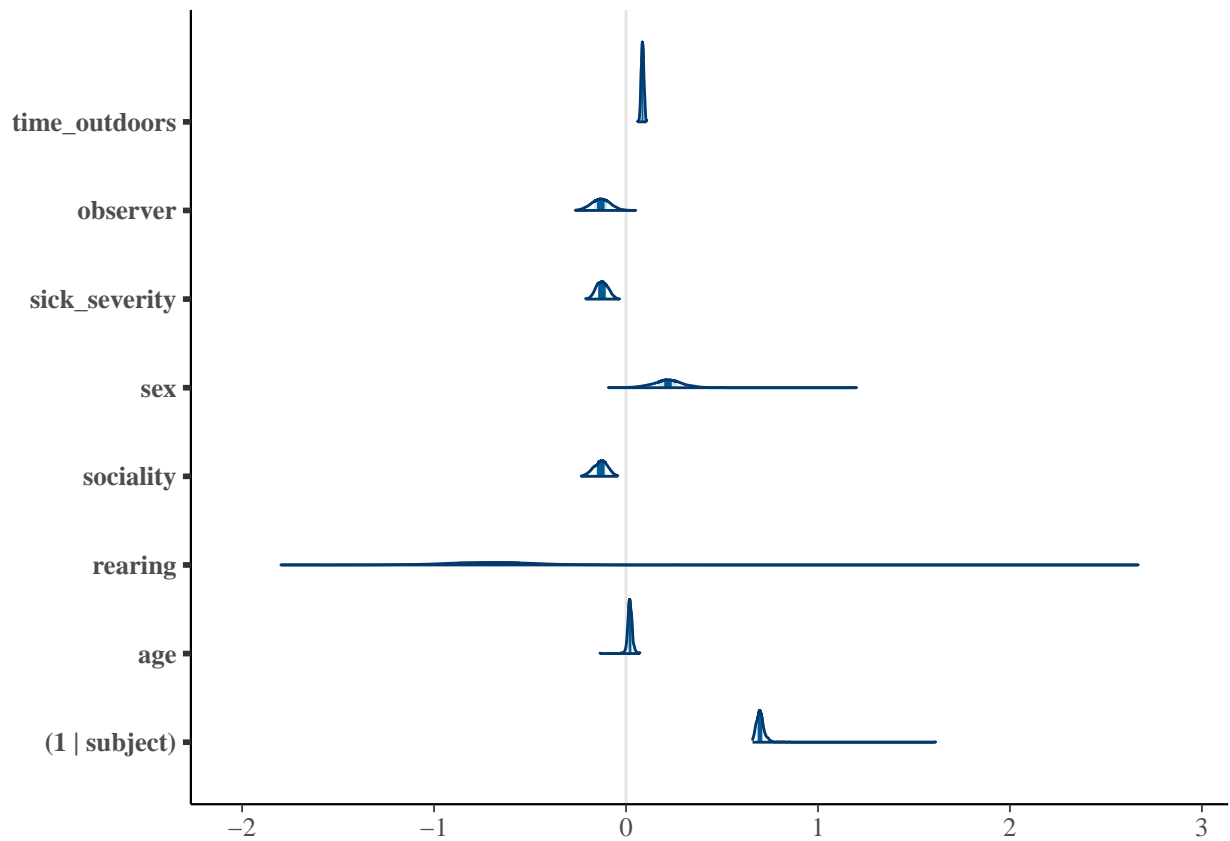
```
|
```

```
summary(cvs_gaze); plot(cvs_gaze, stats = c('elpd', 'rmse'))
```

##	size	solution_terms	elpd	elpd.se
## 2	0	<NA>	-1332.918	28.98704
## 3	1	time_outdoors	-1316.045	28.97957
## 4	2	observer	-1305.367	29.03035
## 5	3	sick_severity	-1294.808	29.01140
## 6	4	sex	-1287.769	28.35422
## 7	5	sociality	-1279.368	27.28371
## 8	6	rearing	-1270.908	28.44104
## 9	7	age	-1261.944	28.05747
## 10	8	time_in_leipzig	-1257.445	28.42059
## 11	9	test_day	-1258.074	28.78184
## 12	10	day2	-1255.591	28.71990
## 13	11	rel_rank	-1253.837	28.38426
## 14	12	test_tp	-1253.406	28.46528
## 15	13	le_mean	-1256.707	28.76652
## 16	14	dist_mean	-1254.236	28.64494
## 17	15	(1 subject)	-975.021	32.38561



```
proj_gaze_cv <- project(cvs_gaze, solution_terms = c(1, 2, 3, 4, 5, 6, 7, 15))
mcmc_areas(as.matrix(proj_gaze_cv), pars = solution_terms(cvs_gaze)[c(1, 2, 3, 4, 5, 6, 7, 15)])
```

relevant covariates: (1 | subject), time_outdoors, observer, sick_severity, sex, sociality, rearing, age
 Choosing the relevant covariates is based on inspection elpd/rmse plots.

pareto k values too high

```
#library(loo)
#loo(m_cau_2l)
#loo(m_inf_2l)
#loo(m_quant_2l)
#loo(m_gaze_2l)
```

Pareto k values are ok.