

Covariate Selection

Supplementary material

Contents

Data pre-processing	1
Packages & Options	1
Data	1
Covariate selection	3
Debugging Area	4
Reference Model: 2-level Multilevel Model (random intercepts only)	4
Predictive Projection	9

Data pre-processing

Packages & Options

```
library(tidyverse)
library(brms)
library(projpred)
library(bayesplot)

# retrieve # of cores
ncores <- parallel::detectCores()

# for output clarity
options(scipen = 999)
```

Data

```
# load data
apes1 <- read_csv("../../data/laac_data_trial.csv")
apes2 <- read_csv("../../data/laac_data_task.csv")

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, observer), as.numeric)) %>%
  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"))

apes1_new <- apes1_new %>%
  mutate(observer_mod = case_when(
    observer == "yes" ~ "yes",
    observer == "no" ~ "no",
    observer != "no" & observer != "yes" & observer != "NA" ~ "yes",
    TRUE ~ "no"
  ), observer_mod = as_factor(observer_mod))

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_grat <- filter(apes1_new, task == "delay_of_gratification")

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 %>%
  # remove duplicates created by day2
  group_by(subject) %>%
  filter(!duplicated(tp_mod)) %>%
  ungroup()

# filter data to only include time points from phase 2
t_cau <- filter(t_cau, time_point >= 15)
t_inf <- filter(t_inf, time_point >= 15)
t_quant <- filter(t_quant, time_point >= 15)
t_gaze <- filter(t_gaze, time_point >= 15)
t_grat <- filter(t_grat, time_point >= 15)

```

Covariate selection

```

# covariate needed for projection prediction
# placed here for easy editing of formula
all_fixed_effects <- c("sick_severity",
                      "test_day", "test_tp",
                      "rel_rank",
                      "observer_mod",

```

```

      "age", "time_in_leipzig",
      "sex", "group",
      "rearing",
      "le_mean",
      "dist_mean",
      "time_outdoors",
      "sociality")

fm <- formula(cogn ~ sick_severity +
  test_day + test_tp +
  rel_rank + # rank_gmc +
  observer_mod +
  age + time_in_leipzig +
  sex + group +
  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 -test_day)

```

Debugging Area

```

#info <- debug_contr_error2(fm, t_cau)
#info

```

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

```

m_cau_2l <- brm(fm, data = t_cau,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)
m_inf_2l <- brm(fm, data = t_inf,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)
m_quant_2l <- brm(fm, data = t_quant,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)
m_gaze_2l <- brm(fm_gaze, data = t_gaze,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)

```

```
m_grat_2l <- brm(fm, data = t_grat,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)
```

```
library(loo)
lapply(list(m_cau_2l, m_inf_2l, m_quant_2l, m_gaze_2l, m_grat_2l), loo)
```

```
summary(m_cau_2l)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer_mod + age + time_in_leipzig
## Data: t_cau (Number of observations: 1051)
## Draws: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
## total post-warmup draws = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 43)
##
```

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	2.37	0.30	1.87	3.05	1.00	5288	7473

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

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	7.54	1.47	4.64	10.42	1.00	8246	7779
sick_severity	0.12	0.07	-0.01	0.25	1.00	21234	9175
test_dayyes	-0.77	0.75	-2.25	0.72	1.00	17534	10102
test_tp	-0.01	0.02	-0.05	0.03	1.00	20226	9633
rel_rank	1.02	0.88	-0.67	2.74	1.00	17040	8305
observer_modno	-0.10	0.14	-0.38	0.18	1.00	20814	8414
age	-0.05	0.05	-0.14	0.05	1.00	8668	8179
time_in_leipzig	0.07	0.08	-0.08	0.22	1.00	8278	8275
sexf	0.18	0.88	-1.54	1.88	1.00	9043	8567
groupb_chimp	2.70	1.65	-0.48	5.98	1.00	7602	8158
groupa_chimp	-0.58	1.20	-2.90	1.84	1.00	6742	7359
grouporangutan	2.30	1.75	-1.13	5.71	1.00	8256	8203
groupbonobo	0.57	1.36	-2.13	3.27	1.00	7387	7361
rearinghand	0.81	1.21	-1.60	3.16	1.00	8609	7976
le_mean	0.08	0.34	-0.58	0.75	1.00	20966	8582
dist_mean	0.61	0.27	0.09	1.12	1.00	18092	9895
time_outdoors	-0.14	0.02	-0.17	-0.10	1.00	19472	9544
sociality	-0.33	0.15	-0.63	-0.04	1.00	20214	8476

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

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	1.56	0.03	1.49	1.63	1.00	15205	8570

```
##
## Draws were sampled 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_mod + age + time_in_leipzig
## Data: t_inf (Number of observations: 1063)
## Draws: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
## total post-warmup draws = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 43)
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept) 2.29 0.30 1.78 2.95 1.00 4899 7011
##
## Population-Level Effects:
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept 7.40 1.51 4.47 10.44 1.00 7815 7543
## sick_severity -0.06 0.08 -0.22 0.09 1.00 17812 7754
## test_dayyes -0.42 0.91 -2.21 1.36 1.00 17050 9673
## test_tp 0.05 0.02 0.00 0.10 1.00 16442 9759
## rel_rank 0.64 0.98 -1.27 2.57 1.00 14287 8711
## observer_modno 0.10 0.17 -0.24 0.44 1.00 18357 8618
## age -0.05 0.05 -0.14 0.04 1.00 8031 7157
## time_in_leipzig 0.37 0.08 0.23 0.53 1.00 7822 6805
## sexf 0.82 0.85 -0.85 2.46 1.00 8441 7656
## groupb_chimp 1.75 1.59 -1.37 4.89 1.00 7856 7307
## groupa_chimp -1.75 1.18 -4.04 0.54 1.00 6838 7650
## grouporangutan -2.35 1.70 -5.76 0.93 1.00 7490 6744
## groupbonobo -1.18 1.32 -3.76 1.45 1.00 7733 7831
## rearinghand -1.08 1.21 -3.52 1.24 1.00 8112 7152
## le_mean 0.16 0.41 -0.64 0.98 1.00 18499 8084
## dist_mean -0.20 0.33 -0.86 0.45 1.00 17806 8957
## time_outdoors -0.13 0.02 -0.17 -0.09 1.00 15273 9173
## sociality -0.49 0.18 -0.86 -0.13 1.00 19108 8808
##
## Family Specific Parameters:
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma 1.91 0.04 1.83 2.00 1.00 15903 8101
##
## Draws were sampled 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_mod + age + time_in_leipzig
## Data: t_quant (Number of observations: 1004)
## Draws: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
## total post-warmup draws = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 43)
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept) 1.56 0.21 1.21 2.03 1.00 4552 6743
##
## Population-Level Effects:

```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## Intercept	7.02	1.11	4.81	9.21	1.00	6374	7711
## sick_severity	0.13	0.07	-0.01	0.28	1.00	20363	8548
## test_dayyes	0.88	0.62	-0.36	2.09	1.00	22958	8686
## test_tp	0.02	0.02	-0.01	0.06	1.00	16317	10267
## rel_rank	2.38	0.79	0.82	3.93	1.00	11526	8819
## observer_modno	0.13	0.17	-0.19	0.46	1.00	19091	8113
## age	-0.02	0.03	-0.08	0.05	1.00	7367	7810
## time_in_leipzig	0.14	0.05	0.04	0.25	1.00	6097	7340
## sexf	0.04	0.59	-1.10	1.21	1.00	6842	7467
## groupb_chimp	1.76	1.10	-0.40	3.93	1.00	5848	7618
## groupa_chimp	0.13	0.83	-1.49	1.76	1.00	5008	6722
## grouporangutan	-0.56	1.22	-2.94	1.83	1.00	5660	6785
## groupbonobo	0.94	0.93	-0.92	2.77	1.00	5394	7273
## rearinghand	-2.11	0.82	-3.76	-0.52	1.00	6820	7306
## le_mean	-0.64	0.32	-1.27	-0.00	1.00	20803	8852
## dist_mean	1.19	0.29	0.62	1.77	1.00	18419	9458
## time_outdoors	-0.09	0.02	-0.13	-0.06	1.00	16390	9198
## sociality	0.14	0.16	-0.18	0.46	1.00	20704	7972

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## sigma	1.65	0.04	1.58	1.73	1.00	18875	8142

Draws were sampled 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_tp + rel_rank + observer_mod + age + time_in_leipzig + sex + gr

Data: t_gaze (Number of observations: 1083)

Draws: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;

total post-warmup draws = 12000

##

Group-Level Effects:

~subject (Number of levels: 43)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## sd(Intercept)	0.66	0.09	0.51	0.85	1.00	4561	7261

##

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
## Intercept	0.30	0.42	-0.52	1.13	1.00	7037	7679
## sick_severity	0.00	0.02	-0.04	0.05	1.00	17583	8751
## test_tp	-0.01	0.01	-0.02	0.00	1.00	15983	9819
## rel_rank	-0.03	0.26	-0.54	0.48	1.00	14321	8642
## observer_modno	0.07	0.05	-0.04	0.17	1.00	17981	8591
## age	0.02	0.01	-0.00	0.05	1.00	8134	7235
## time_in_leipzig	-0.01	0.02	-0.05	0.04	1.00	8140	8397
## sexf	0.14	0.24	-0.33	0.61	1.00	7322	7628
## groupb_chimp	0.03	0.46	-0.86	0.94	1.00	6704	7332
## groupa_chimp	0.36	0.34	-0.31	1.04	1.00	5789	6739
## grouporangutan	-0.24	0.49	-1.22	0.71	1.00	7179	7834

```
## groupbonobo      0.32      0.38     -0.43      1.06 1.00      6380      7169
## rearinghand      -0.48      0.34     -1.14      0.19 1.00      7627      7380
## le_mean          -0.06      0.11     -0.28      0.16 1.00     18055     8125
## dist_mean        -0.09      0.10     -0.28      0.10 1.00     16178     9170
## time_outdoors     0.00      0.01     -0.01      0.02 1.00     16329     9994
## sociality         -0.12      0.06     -0.22     -0.01 1.00     16565     8040
## day2yes           -0.03      0.03     -0.10      0.03 1.00     16744     7502
```

```
##
```

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

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      0.56      0.01      0.54      0.59 1.00     15358     7755
```

```
##
```

```
## Draws were sampled 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_grat_2l)
```

```
## Family: gaussian
```

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

```
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer_mod + age + time_in_leipzig
```

```
## Data: t_grat (Number of observations: 2041)
```

```
## Draws: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
```

```
## total post-warmup draws = 12000
```

```
##
```

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

```
## ~subject (Number of levels: 43)
```

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)      3.03      0.39      2.38      3.90 1.00     4494     6554
```

```
##
```

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

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          6.23      1.80      2.71      9.90 1.00     7917     8157
## sick_severity       0.09      0.06     -0.04      0.22 1.00     21594     8819
## test_dayyes        -1.12      0.59     -2.28      0.03 1.00     21851     8176
## test_tp            -0.10      0.02     -0.14     -0.06 1.00     19107     9579
## rel_rank            0.96      0.88     -0.78      2.69 1.00     18046     7968
## observer_modno     -0.11      0.16     -0.42      0.19 1.00     21689     8551
## age                0.03      0.06     -0.09      0.14 1.00      8220     7670
## time_in_leipzig     0.23      0.10      0.04      0.43 1.00      8039     7898
## sexf               -2.19      1.12     -4.41      0.01 1.00      8560     7618
## groupb_chimp        0.34      2.11     -3.83      4.47 1.00      7331     7664
## groupa_chimp       -0.28      1.53     -3.29      2.69 1.00      6624     7718
## grouporangutan     -1.51      2.22     -5.89      2.77 1.00      7601     7463
## groupbonobo        -1.33      1.73     -4.76      2.04 1.00      7412     7594
## rearinghand        -0.49      1.53     -3.58      2.49 1.00      8376     7904
## le_mean            0.01      0.45     -0.86      0.88 1.00     21047     8516
## dist_mean          -0.02      0.29     -0.58      0.53 1.00     19833     9230
## time_outdoors      -0.11      0.02     -0.14     -0.07 1.00     18892     9994
## sociality          -0.28      0.16     -0.60      0.04 1.00     22494     8449
```

```
##
```

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

```
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma      2.17      0.03      2.11      2.24 1.00     21383     8538
```

```
##
```



```
## Draws were sampled 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)"))
tmp <- all_fixed_effects[-2]
s_terms_gaze <- c("1", c(tmp, "day2"),
                  paste0(paste(c(tmp, "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)
refM_grat <- get_refmodel(m_grat_2l)

vs_cau <- varsel(refM_cau, search_terms = s_terms)
summary(vs_cau); plot(vs_cau, stats = c('elpd', 'rmse'))

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

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, ndraws = 10)
#mcmc_areas(as.matrix(proj_inf), pars = solution_terms(vs_inf)[relevant_cov_vsinf])

vs_quant <- varsel(refM_quant, search_terms = s_terms)
summary(vs_quant); plot(vs_quant, stats = c('elpd', 'rmse'))

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

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])
```

```

vs_grat <- varsel(refM_grat, search_terms = s_terms)
summary(vs_grat); plot(vs_grat, stats = c('elpd', 'rmse'))

randint_ind_vsgrat <- length(solution_terms(vs_grat))
relevant_cov_vsgrat <- c(1, 2, 3, 4, 5, randint_ind_vsgrat)

cvs_cau <- cv_varsel(refM_cau,
                     search_terms = s_terms, cv_method = "L00", method = "forward",
                     seed = 2020)

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

```

```

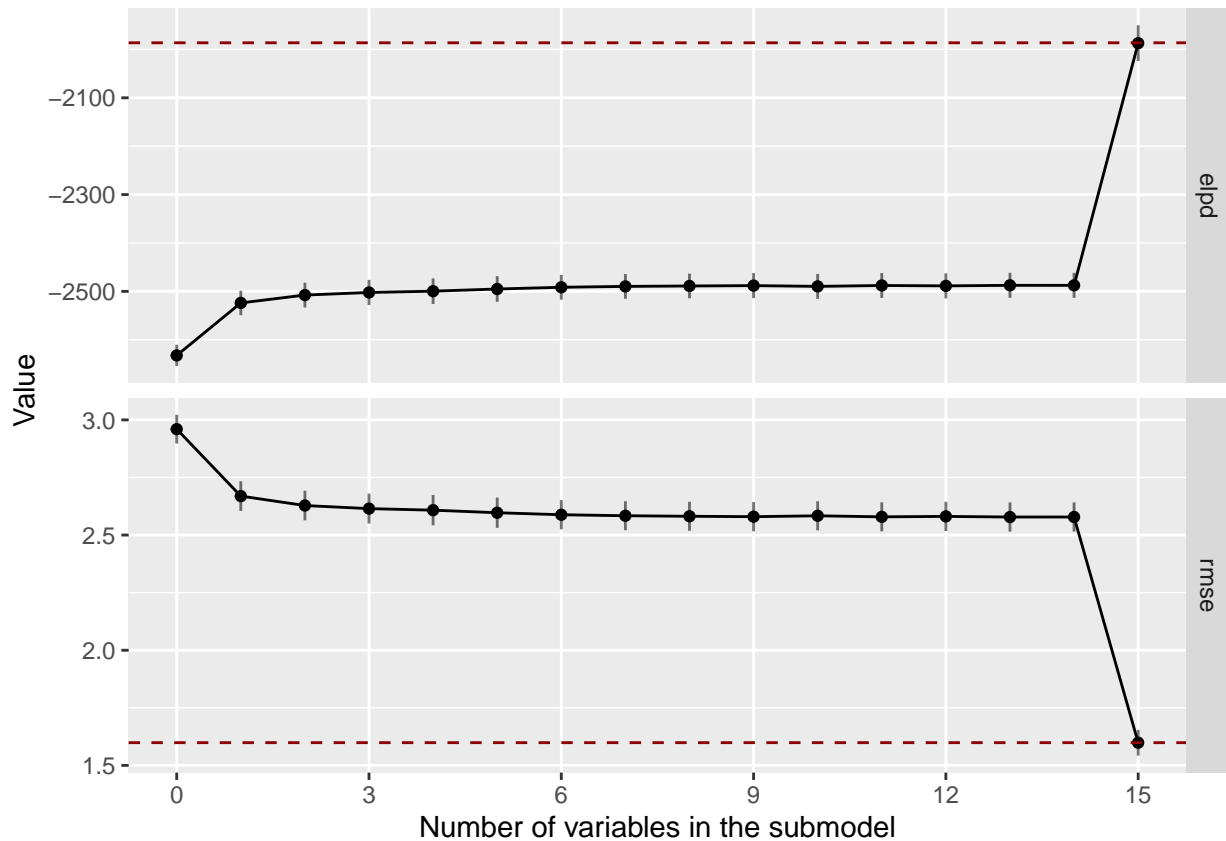
summary(cvs_cau); plot(cvs_cau, stats = c('elpd', 'rmse'))

```

```

##    size  solution_terms      elpd  elpd.se
## 2     0              <NA> -2632.381 22.17875
## 3     1             group -2523.896 25.40957
## 4     2  time_outdoors -2507.663 25.81262
## 5     3  observer_mod -2502.323 26.15601
## 6     4 time_in_leipzig -2499.616 26.56242
## 7     5             age -2495.134 26.70207
## 8     6          rearing -2491.561 25.78467
## 9     7  sick_severity -2489.756 25.69297
## 10    8      sociality -2488.840 25.68971
## 11    9             sex -2488.112 26.04062
## 12   10          rel_rank -2489.686 25.90108
## 13   11          test_day -2487.843 25.88234
## 14   12          dist_mean -2488.633 26.02771
## 15   13          test_tp -2487.511 26.00442
## 16   14          le_mean -2487.586 26.00413
## 17   15  (1 | subject) -1987.106 37.12607

```



```
# 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), group

```
cvs_inf <- cv_varsel(refM_inf,
  search_terms = s_terms, cv_method = "LOO", method = "forward",
  seed = 2020)
```

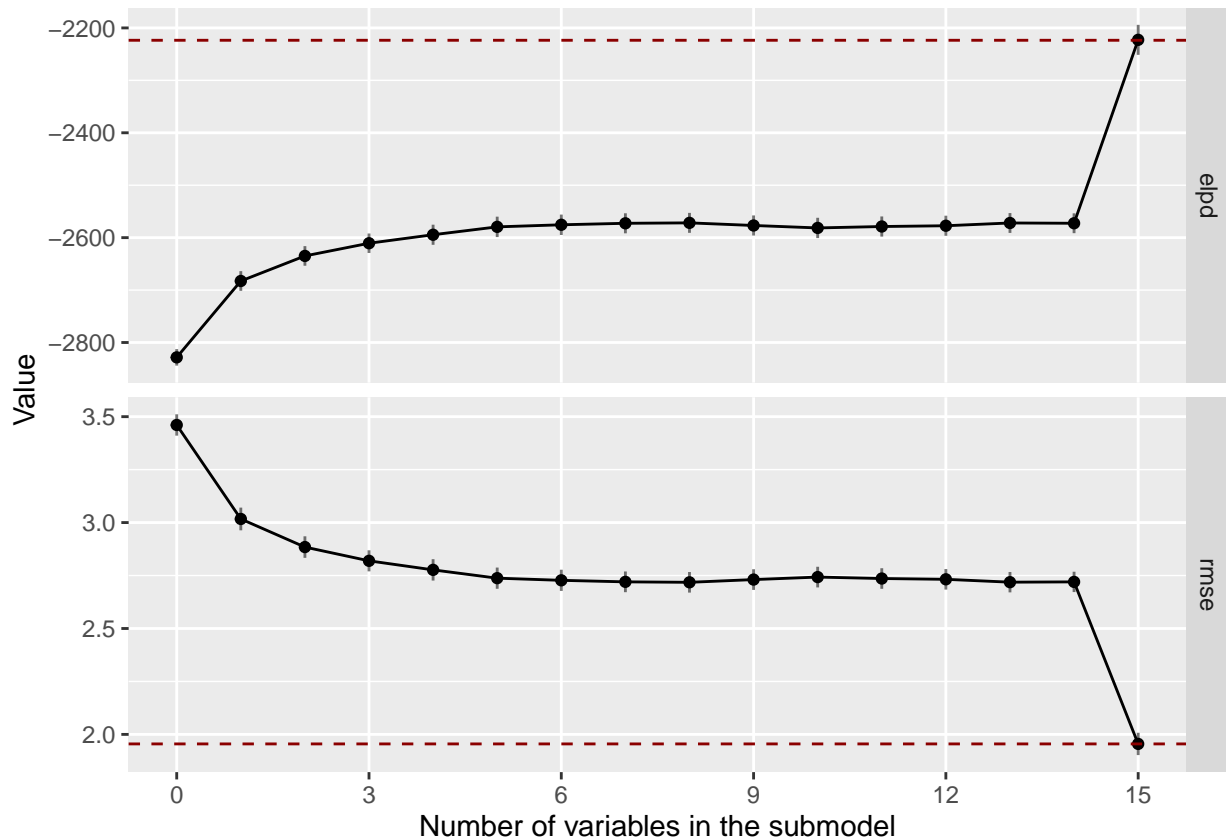
```
## [1] "Computing LOOs..."
```

```
## |
```

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

```
##   size solution_terms   elpd elpd.se
## 2    0              <NA> -2828.352 15.82216
## 3    1 time_in_leipzig -2682.695 18.91740
```

```
## 4      2      group -2634.904 18.81065
## 5      3      age -2610.675 18.71129
## 6      4 time_outdoors -2594.494 19.32609
## 7      5      sex -2579.451 19.57018
## 8      6      sociality -2575.496 19.58552
## 9      7 sick_severity -2572.666 19.22843
## 10     8      rearing -2571.815 19.16138
## 11     9      dist_mean -2576.845 19.32863
## 12    10      test_tp -2581.475 19.43935
## 13    11      rel_rank -2578.800 19.34059
## 14    12      le_mean -2577.327 19.25856
## 15    13      test_day -2572.052 19.05043
## 16    14 observer_mod -2572.573 19.03634
## 17    15 (1 | subject) -2222.716 28.76281
```



```
# 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, group, age

```
cvs_quant <- cv_varsel(refM_quant,
  search_terms = s_terms, cv_method = "L00", method = "forward",
  seed = 2020)
```

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

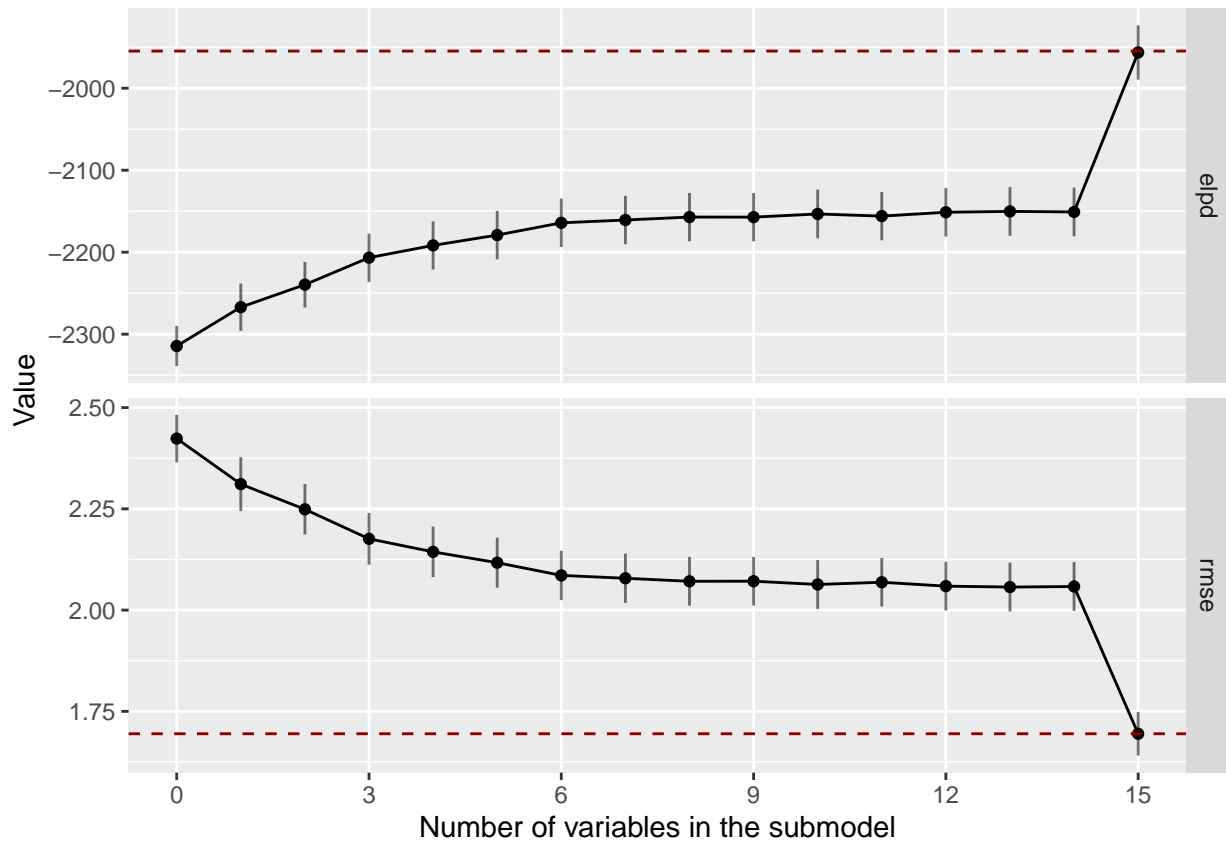
```
## |
```

|

```
## [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."
## [1] "Done."
```

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

```
##      size solution_terms      elpd elpd.se
## 2      0              <NA> -2314.440 24.60445
## 3      1          rel_rank -2267.022 29.08643
## 4      2          rearing -2239.612 27.93706
## 5      3 time_in_leipzig -2206.693 29.65681
## 6      4             group -2191.728 29.49371
## 7      5  time_outdoors -2179.124 29.60804
## 8      6  observer_mod -2164.195 29.74271
## 9      7          dist_mean -2160.771 29.75924
## 10     8          test_tp -2157.152 29.65326
## 11     9          test_day -2157.242 29.63068
## 12    10  sick_severity -2153.379 29.77315
## 13    11             age -2156.008 29.67334
## 14    12             sex -2151.314 29.78852
## 15    13          le_mean -2150.254 29.87548
## 16    14        sociality -2150.911 29.92237
## 17    15  (1 | subject) -1956.428 33.29908
```



```
# proj_quant_cv <- project(cvs_quant, solution_terms = c(1, 2, 3, 15))
# mcmc_areas(as.matrix(proj_quant_cv), pars = solution_terms(cvs_quant)[c(1, 2, 3, 15)])
```

relevant covariates: (1 | subject), rel_rank, rearing, time_in_leipzig, group, time_outdoors

```
fm_gaze <- update(fm, . ~ . +day2 -test_day -(1|subject))
m_gaze_2l <- brm(fm_gaze, data = t_gaze,
  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
  seed = 2021,
  save_pars = save_pars(all = TRUE)
)
```

Warning: Rows containing NAs were excluded from the model.

```
s_terms_gaze <- c("1", c(tmp, "day2"),
  paste0(paste(c(tmp, "day2"), collapse = " + ")))
refM_gaze <- get_refmodel(m_gaze_2l)
```

```
cvs_gaze <- cv_varsel(refM_gaze,
  search_terms = s_terms_gaze, cv_method = "LOO", method = "forward",
  seed = 2020)
```

[1] "Computing LOOs..."

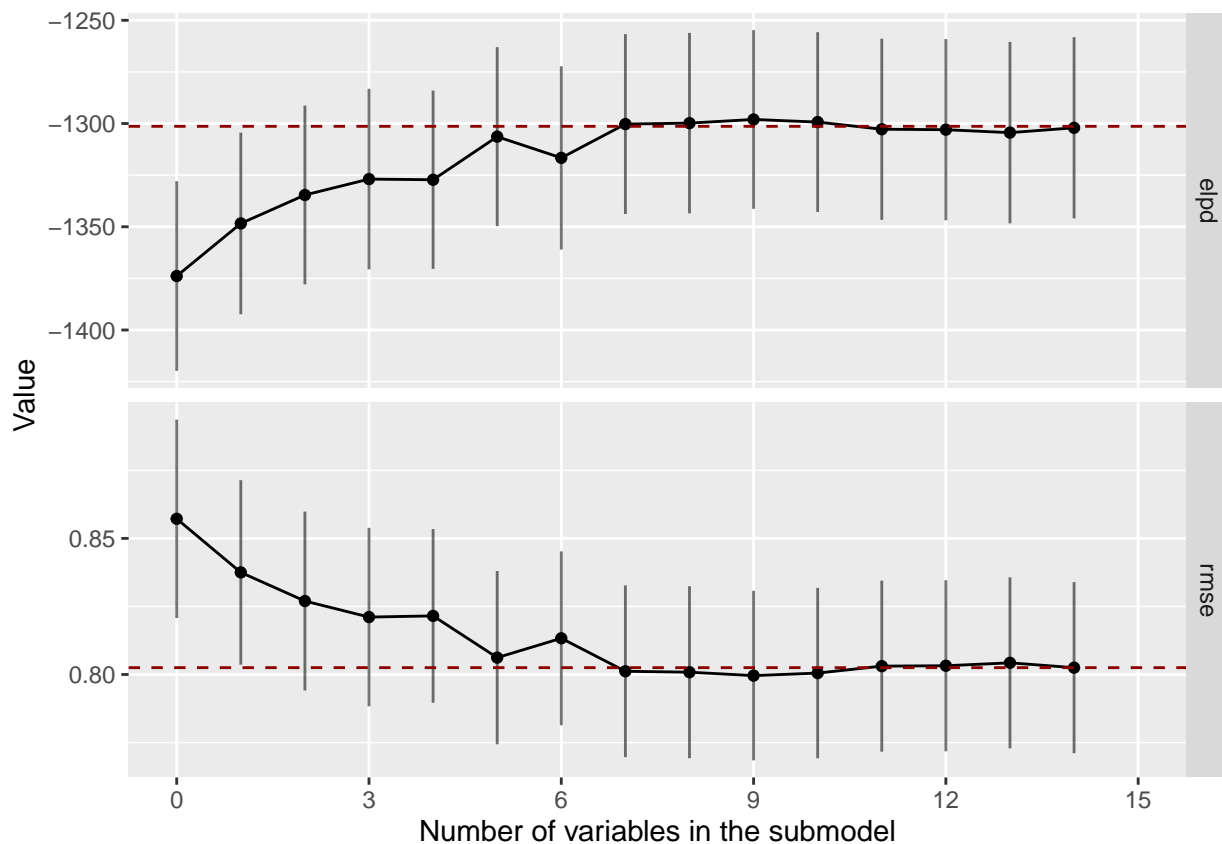
|

|

```
## [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."
## [1] "Done."
```

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

##	size	solution_terms	elpd	elpd.se
## 2	0	<NA>	-1373.872	46.20086
## 3	1	group	-1348.390	44.23423
## 4	2	sex	-1334.614	43.54273
## 5	3	observer_mod	-1326.907	43.97819
## 6	4	age	-1327.224	43.43340
## 7	5	rearing	-1306.359	43.60318
## 8	6	sick_severity	-1316.637	44.63035
## 9	7	sociality	-1300.271	43.79745
## 10	8	time_in_leipzig	-1299.806	43.94170
## 11	9	rel_rank	-1298.034	43.51579
## 12	10	day2	-1299.294	43.77045
## 13	11	test_tp	-1302.760	44.11562
## 14	12	le_mean	-1302.986	44.14591
## 15	13	dist_mean	-1304.406	44.19067
## 16	14	time_outdoors	-1302.068	44.11677



```
# 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: group, sex (random intercept removed due to convergence issues)

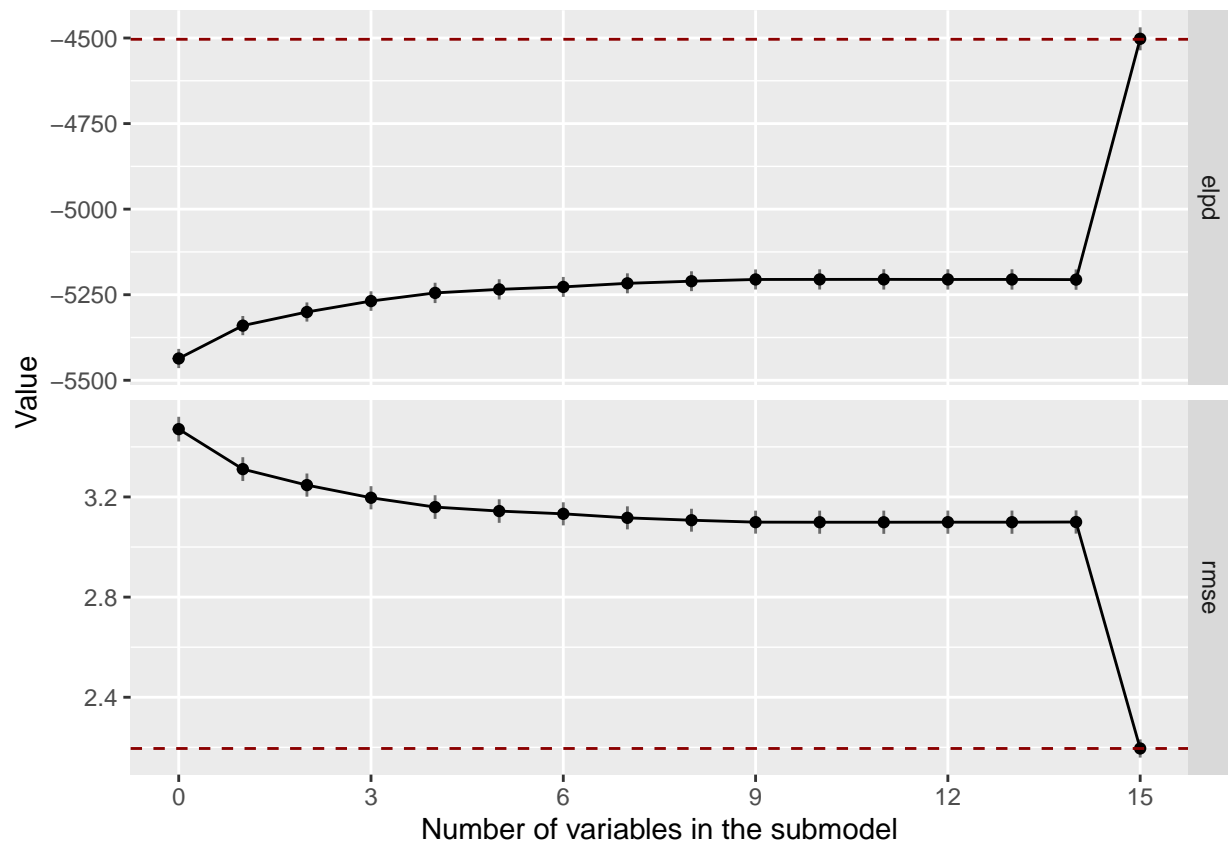
```
cvs_grat <- cv_varsel(refM_grat,
                      search_terms = s_terms, cv_method = "L00", method = "forward",
                      seed = 2020)
```

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

```
##      |
```

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

```
##      size  solution_terms      elpd  elpd.se
## 2      0      <NA> -5436.348 28.22279
## 3      1      rel_rank -5340.265 28.21010
## 4      2      observer_mod -5300.551 28.14984
## 5      3      sex -5268.591 28.76628
## 6      4      group -5244.726 29.81364
## 7      5 time_in_leipzig -5234.392 29.81946
## 8      6      time_outdoors -5227.208 29.35141
## 9      7      test_tp -5216.730 29.35081
## 10     8      sick_severity -5210.542 29.18732
## 11     9      sociality -5205.454 29.44042
## 12    10      dist_mean -5205.300 29.72444
## 13    11      age -5205.259 29.86887
## 14    12      test_day -5205.431 29.83157
## 15    13      le_mean -5205.401 29.88028
## 16    14      rearing -5205.862 29.89071
## 17    15      (1 | subject) -4502.324 33.76243
```

```
# 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), rel_rank, observer_mod, sex, group