ghrmodel_laos

2025-03-03

Load

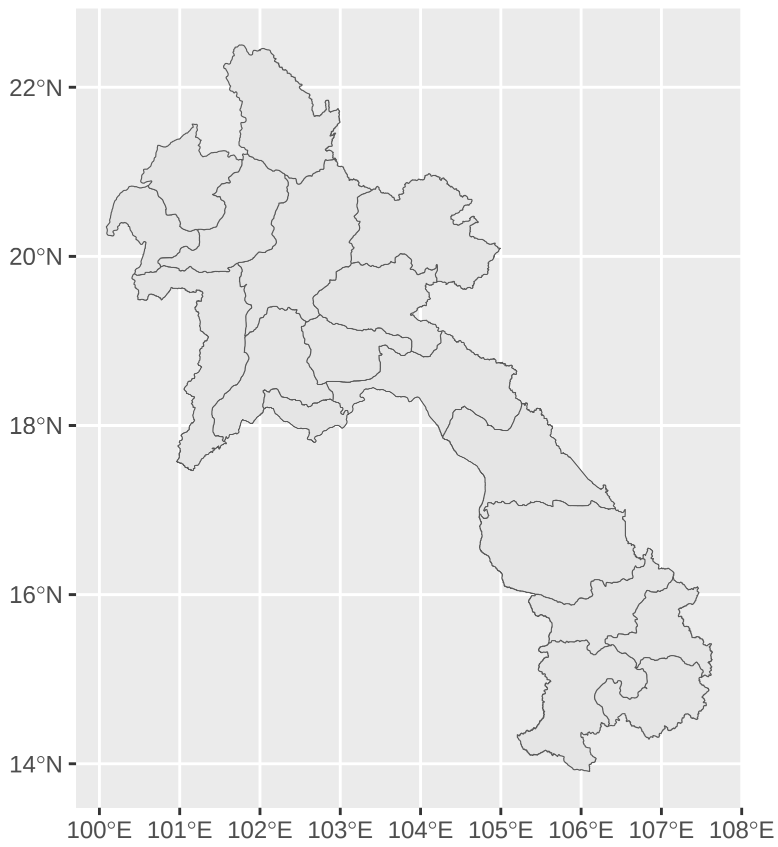
```
## Load Harmonized Data ----
data <- read.csv("harmonized_data.csv")
head(data)
```

```
##   X year month       date    province dengue_cases population   tasmax    tasmin      tas prlr
## 1 1 2023     1 2023-01-01  Xaisomboun            1     114000 23.89677  7.970968 15.93387  8.8
## 2 2 2023     1 2023-01-01      Attapu           10     166000 30.78710 20.403226 25.59516  0.8
## 3 3 2023     1 2023-01-01   Champasak            5     772000 30.78710 18.535484 24.66129 32.8
## 4 4 2023     1 2023-01-01      Xekong            3     134000 30.63548 16.403226 23.51935  5.0
## 5 5 2023     1 2023-01-01     Salavan            9     457000 29.26774 17.719355 23.49355 19.2
## 6 6 2023     1 2023-01-01 Savannakhet           15    1102000 28.32581 15.461290 21.89355  0.7
```

```
## Load shapefile
laos<- st_read("gadm41_LAO1_cleaned.shp")
```

```
## Reading layer `gadm41_LAO1_cleaned' from data source
##   `/home/akawieck/Documents/projects/DHIS2/dhis2.personal/ghrsuite_laos/gadm41_LAO1_cleaned.shp' usin
## Simple feature collection with 18 features and 11 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 100.0868 ymin: 13.90968 xmax: 107.635 ymax: 22.5004
## Geodetic CRS:  WGS 84
```

```
ggplot() +
  geom_sf(data = laos)
```



```
# Create Adjacences Matrix
nb <- spdep::poly2nb(laos)
g <- spdep::nb2mat(nb, style = "B")
```

GHRmodel description

Model dengue in Laos

We learned from the exploration of the data that precipitation, minimum temperature and average temperature might be the most relevant variables affecting dengue incidence. We can also observe that there may be a relationship between dengue outbreaks and increased average temperatures several months before.

Prepare data for modeling in INLA

```
# Assign numeric IDs to non-numeric variables for INLA modeling

data<- data %>%
  dplyr::mutate(dplyr::across(c("year", "month", "province"),
                ~ as.numeric(as.factor(.)),
                .names = "{.col}_id"),
         dplyr::across(c("dengue_cases", "population",
"tasmax", "tasmin", "tas", "prlr"),
              ~ as.numeric(.))) %>%
  dplyr::select(-X) %>%
  # person time = 100000 person-month
  mutate(dengue_incidence=(dengue_cases / population) * 100000 ) # calculate incidence
```

Create lagged covariates

Here we create covariates lagged between 1-8 months for each observation, goruping by province.

```
# Lag covariates and attach to the original data
data <- ghrmodel::lag_cov(data = data,
              var = c("tas", "prlr"),
              time = "date",
              lag = c(8),
              group = "province",
              full = TRUE) # Merge = TRUE the matrix is merged to the data

dplyr::glimpse(data)
```

```
## Rows: 1,944
## Columns: 30
## $ year             <int> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2016
## $ month            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ date             <chr> "2015-01-01", "2015-02-01", "2015-03-01", "2015-04-01", "2015-05-01", "2015-
## $ province         <chr> "Attapu", "Attapu", "Attapu", "Attapu", "Attapu", "Attapu", "Attapu", "Attap
## $ dengue_cases     <dbl> 13, 39, 36, 21, 46, 68, 166, 171, 67, 44, 13, 8, 14, 13, 8, 5, 5, 9, 20, 9,
## $ population       <dbl> 166000, 166000, 166000, 166000, 166000, 166000, 166000, 166000, 166000, 1660
## $ tasmax           <dbl> 31.98065, 33.42500, 36.29032, 37.68000, 36.58710, 33.94667, 30.60968, 31.493
## $ tasmin           <dbl> 17.55161, 20.41786, 24.76129, 25.89333, 26.89677, 25.67000, 24.76774, 24.951
## $ tas              <dbl> 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.83667, 27.71613, 28.241
## $ prlr             <dbl> 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 263.3, 80.3, 0.0, 0.0
## $ year_id          <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3
## $ month_id         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
## $ province_id      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
## $ dengue_incidence <dbl> 7.831325, 23.493976, 21.686747, 12.650602, 27.710843, 40.963855, 100.000000
## $ tas.l1           <dbl> NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.83667, 27.71613, 28
## $ tas.l2           <dbl> NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.83667, 27.71613
## $ tas.l3           <dbl> NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.83667, 27.7
## $ tas.l4           <dbl> NA, NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.83667,
## $ tas.l5           <dbl> NA, NA, NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29.836
## $ tas.l6           <dbl> NA, NA, NA, NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129, 29
## $ tas.l7           <dbl> NA, NA, NA, NA, NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.76129
## $ tas.l8           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 24.78387, 26.94286, 30.55484, 31.81667, 31.7
## $ prlr.l1          <dbl> NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 263.3, 80.3, 0.0,
## $ prlr.l2          <dbl> NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 263.3, 80.3, 0
## $ prlr.l3          <dbl> NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 263.3, 80
## $ prlr.l4          <dbl> NA, NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 263.3
## $ prlr.l5          <dbl> NA, NA, NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9, 26
## $ prlr.l6          <dbl> NA, NA, NA, NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 148.9
## $ prlr.l7          <dbl> NA, NA, NA, NA, NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282.7, 1
## $ prlr.l8          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 7.6, 0.0, 2.3, 0.7, 71.0, 353.6, 293.6, 282
```

Explore non-linear effects

Priors

```r
# Write  formulas with 3 random effects with customized prior
prior_re1 <- list(prec = list(prior = 'loggamma', param = c(0.01, 0.01)))
prior_re2 <- list(prec = list( prior = 'pc.prec', param = c(0.5 / 0.31, 0.01)),
                  phi = list( prior = 'pc', param = c(0.5, 2 / 3)))
```

Prior	Type	Parameters	Meaning
`prior_re1`	Log-Gamma	`(0.01, 0.01)`	Weak prior on precision, allowing large variance.
`prior_re2$prec`	PC Prior for Precision	`(0.5 / 0.31, 0.01)`	Shrinks precision toward a reasonable range, avoiding overfitting.
`prior_re2$phi`	PC Prior for Spatial Dependency	`(0.5, 2/3)`	Encourages structured spatial correlation.

prior_re1: Log-Gamma Prior

The loggamma(0.01, 0.01) prior (shape = 0.01, rate = 0.01) is commonly used as a weakly informative prior, ensuring flexibility in the variance structure without forcing a strong assumption.

prior_re2: PC Prior for Precision

0.5 / 0.31 1.61 sets the prior on precision.

0.01 is the probability that the standard deviation (= sqrt(1/precision)) exceeds a given threshold.

The Penalized Complexity (PC) prior is designed to avoid overfitting by shrinking unnecessary complexity. This implies a preference for a moderate variance but allows the data to override it if needed.

prior_re2: Spatial Dependency (phi)

0.5: The median of phi (controls spatial dependency balance).

2 / 3 0.67: Probability that phi > 0.5.

phi is a mixing parameter that controls the balance between the structured spatial effect (neighbor-dependent) or unstructured spatial effect (independent random noise). A prior that favors phi > 0.5 encourages spatial smoothing, meaning that neighboring areas share more information.

Univariate predictors

First we will test the models with univariable predictors, to determine whether we should include linear or non-linear predictors in our model based on goodness of fit.

Write univariate model functions

```r
# Create a list of linear lagged univariable predictors.
# Includes all covariates that include the pattern tas.l and prlr.l
```

```r
cov_uni_l <- ghrmodel::extract_covariates(data=data,
                                           pattern= c("tas.l",
                                                      "prlr.l"))

dplyr::glimpse(cov_uni_l)
```

```
## List of 16
##  $ : chr "tas.l1"
##  $ : chr "tas.l2"
##  $ : chr "tas.l3"
##  $ : chr "tas.l4"
##  $ : chr "tas.l5"
##  $ : chr "tas.l6"
##  $ : chr "tas.l7"
##  $ : chr "tas.l8"
##  $ : chr "prlr.l1"
##  $ : chr "prlr.l2"
##  $ : chr "prlr.l3"
##  $ : chr "prlr.l4"
##  $ : chr "prlr.l5"
##  $ : chr "prlr.l6"
##  $ : chr "prlr.l7"
##  $ : chr "prlr.l8"
```

```r
# Create a list of non linear lagged univariable predictors not replicated
cov_uni_l_nl <- ghrmodel::non_linear_covariates(covariates = cov_uni_l,
                                      method = "quantile",
                                      pattern = c("tas", "prlr"),
                                      n = 10)
dplyr::glimpse(cov_uni_l_nl)
```

```
## List of 16
##  $ : chr "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l1, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l2, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l3, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l4, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l5, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l6, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l7, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l8, method='quantile', n=10), model='rw2')"
```

```r
# Create a list of non linear lagged univariable predictors replicated by province
cov_uni_l_nl_rep <- ghrmodel::non_linear_covariates(covariates = cov_uni_l,
                                      method = "quantile",
                                      pattern = c("tas", "prlr"),
                                      n = 10,
                                      replicate = "province")

dplyr::glimpse(cov_uni_l_nl_rep)
```

```
## List of 16
##  $ : chr "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l1, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l2, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l3, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l4, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l5, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l6, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l7, method='quantile', n=10), model='rw2', replicate =province)"
##  $ : chr "f(INLA::inla.group(prlr.l8, method='quantile', n=10), model='rw2', replicate =province)"
```

```
cov_uni_list <- c( "tas",
                     "prlr",
                      cov_uni_l,
                      cov_uni_l_nl) #, cov_uni_l_nl_rep

dplyr::glimpse(cov_uni_list)
```

```
## List of 34
##  $ : chr "tas"
##  $ : chr "prlr"
##  $ : chr "tas.l1"
##  $ : chr "tas.l2"
##  $ : chr "tas.l3"
##  $ : chr "tas.l4"
##  $ : chr "tas.l5"
##  $ : chr "tas.l6"
##  $ : chr "tas.l7"
##  $ : chr "tas.l8"
##  $ : chr "prlr.l1"
##  $ : chr "prlr.l2"
##  $ : chr "prlr.l3"
##  $ : chr "prlr.l4"
##  $ : chr "prlr.l5"
##  $ : chr "prlr.l6"
##  $ : chr "prlr.l7"
##  $ : chr "prlr.l8"
##  $ : chr "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l1, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l2, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l3, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l4, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l5, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l6, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l7, method='quantile', n=10), model='rw2')"
##  $ : chr "f(INLA::inla.group(prlr.l8, method='quantile', n=10), model='rw2')"
```

```
saveRDS(cov_uni_list, "cov_uni_list.rds")
```

Write univariate model formulas

```
cov_uni_formulas <- write_inla_formulas(outcome = "dengue_cases",
                                        covariates = cov_uni_list ,
                                        re1 = list(id ="month_id",
                                                   re ="rw1", cyclic = TRUE,
                                                   hyper = "prior_re1",
                                                   replicate = "province_id" ),
                                        re2 = list(id = "year_id",
                                                   re = "rw1",
                                                   hyper = "prior_re1"),
                                        re3 = list(id = "province_id",
                                                   re = "bym2",
                                                   graph = "g",
                                                   hyper = "prior_re2"),
                                                   baseline = TRUE)

head(cov_uni_formulas)
```

```
## [1] "dengue_cases ~ 1 + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, constr = T
## [2] "dengue_cases ~ 1 + tas + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, cons
## [3] "dengue_cases ~ 1 + prlr + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, con
## [4] "dengue_cases ~ 1 + tas.l1 + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, c
## [5] "dengue_cases ~ 1 + tas.l2 + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, c
## [6] "dengue_cases ~ 1 + tas.l3 + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, c
```

```
# transform formulas list into a GHRformulas object

cov_uni_formulas_ghr <- ghrmodel::as_GHRformulas(formulas = cov_uni_formulas)

class(cov_uni_formulas_ghr)
```

```
## [1] "GHRformulas" "list"
```

```
str(cov_uni_formulas_ghr)
```

```
## List of 4
##  $ formulas: chr [1:35] "dengue_cases ~ 1 + f(month_id, model = 'rw1', replicate = province_id, cycli
##  $ vars    :'data.frame':    35 obs. of  1 variable:
##   ..$ covariate_1: chr [1:35] NA "tas" "prlr" "tas.l1" ...
##  $ re      : Named chr [1:3] "f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, cons
##   ..- attr(*, "names")= chr [1:3] "re_1" "re_2" "re_3"
##  $ outcome : chr "dengue_cases"
##  - attr(*, "class")= chr [1:2] "GHRformulas" "list"
```

```
dplyr::glimpse(cov_uni_formulas_ghr$vars)
```

```
## Rows: 35
## Columns: 1
## $ covariate_1 <chr> NA, "tas", "prlr", "tas.l1", "tas.l2", "tas.l3", "tas.l4", "tas.l5", "tas.l6", "
```

Fit univariate models

```
m_uni <- ghrmodel::fit_models(formulas = cov_uni_formulas_ghr   ,
                              data = data,
                              family = "nbinomial",       # specify family
                              name = "m",
                              offset = "population",
                              config = TRUE,
                              pb = TRUE,
                              nthreads = 8)


saveRDS(m_uni, "m_uni.rds")
```

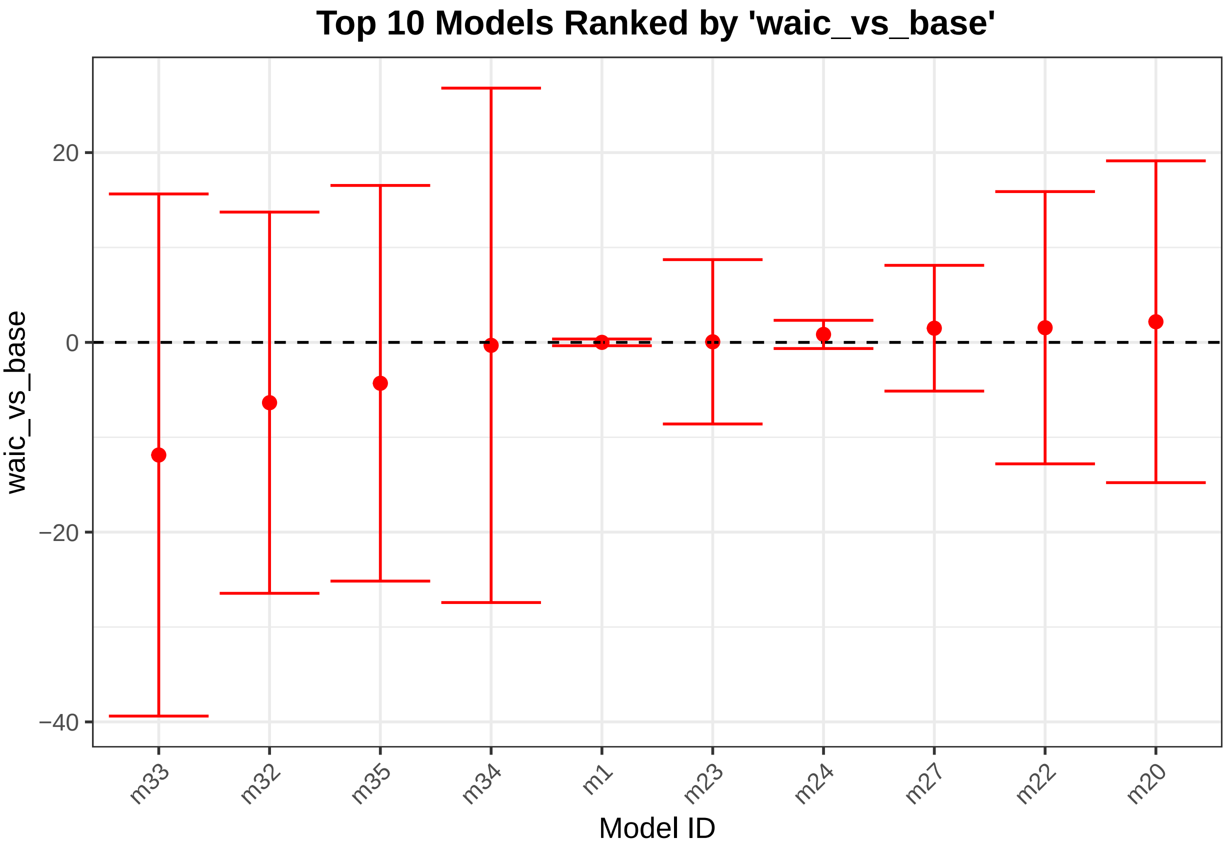
```
m_uni <- readRDS("m_uni.rds")
```

Rank univariate models

```
# goodness of fit metrics in a dataframe
m_uni_gof <- m_uni$mod.gof

m_uni_gof <- m_uni_gof %>%
  dplyr::arrange(waic)%>%
  dplyr::mutate(rank_waic = dense_rank(waic)) %>%
  dplyr::arrange(crps)%>%
  dplyr::mutate(rank_crps = dense_rank(crps))%>%
  dplyr::arrange(mae)%>%
  dplyr::mutate(rank_mae = dense_rank(mae))
```

WAIC

```r
rank_uni_waic_vs_base <- ghrmodel::rank_models(
  models = m_uni,
  metric = "waic_vs_base",
  plot = TRUE,
  n = 10,
  intercept = TRUE,
  ci = TRUE
)
```



```r
m_uni_gof %>%
  dplyr::filter(model_id %in% rank_uni_waic_vs_base)%>%
  dplyr::select(model_id, covariate_1, waic, rank_waic)%>%
  dplyr::arrange(waic)
```

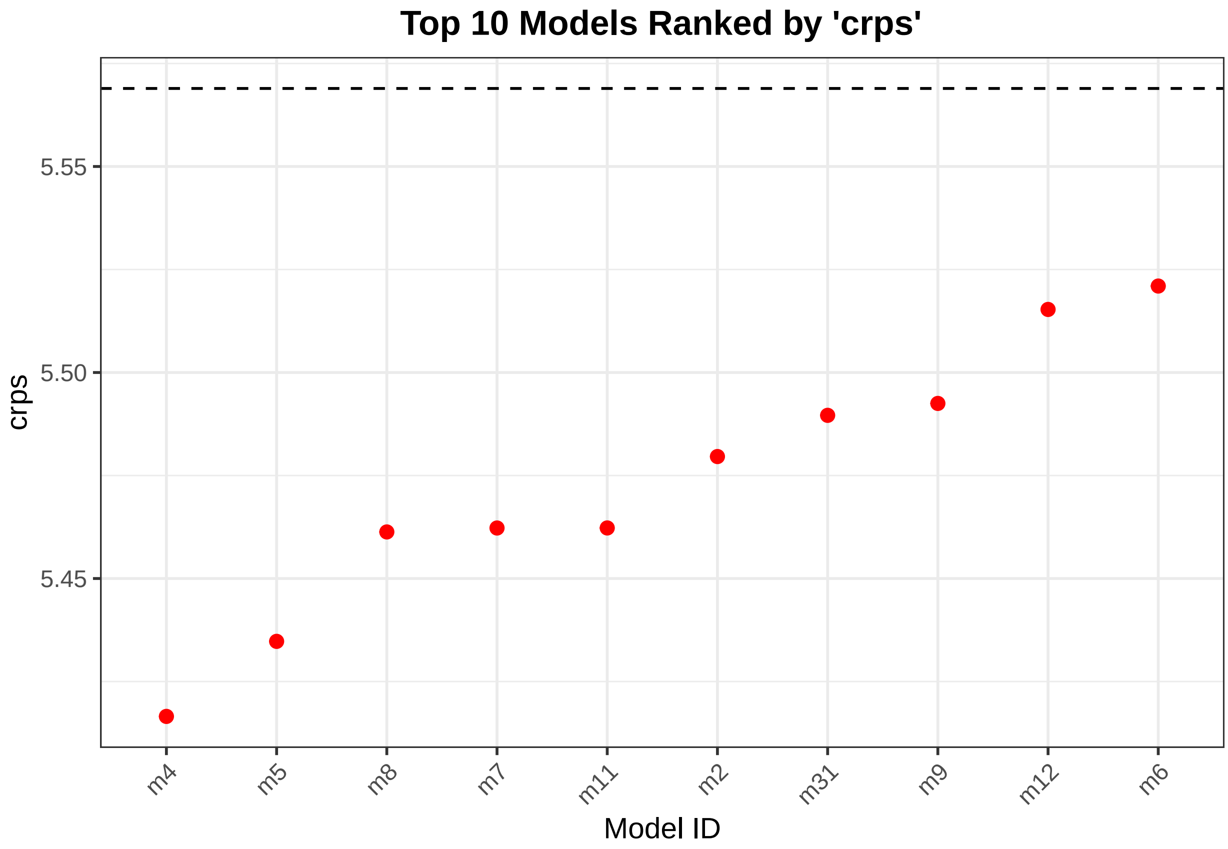
```
##    model_id        covariate_1     waic rank_waic
## 1       m33 prlr.l6_nl_q10_rw2 14187.43         1
## 2       m32 prlr.l5_nl_q10_rw2 14192.95         2
## 3       m35 prlr.l8_nl_q10_rw2 14194.99         3
## 4       m34 prlr.l7_nl_q10_rw2 14198.99         4
## 5        m1               <NA> 14199.31         5
## 6       m23  tas.l4_nl_q10_rw2 14199.36         6
## 7       m24  tas.l5_nl_q10_rw2 14200.14         7
## 8       m27  tas.l8_nl_q10_rw2 14200.80         8
## 9       m22  tas.l3_nl_q10_rw2 14200.85         9
## 10      m20  tas.l1_nl_q10_rw2 14201.48        10
```

Best fitting models for precipitation have non linear precipitation terms. Lags 5-8 fit better that the base model but not significantly so.

Non-linear temperature measures also fit better, but these terms do not improve the model over the base model.

CRPS

```
rank_uni_crps <- ghrmodel::rank_models(
  models = m_uni,
  metric = "crps",
  plot = TRUE,
  n = 10,
  intercept = TRUE
)
```

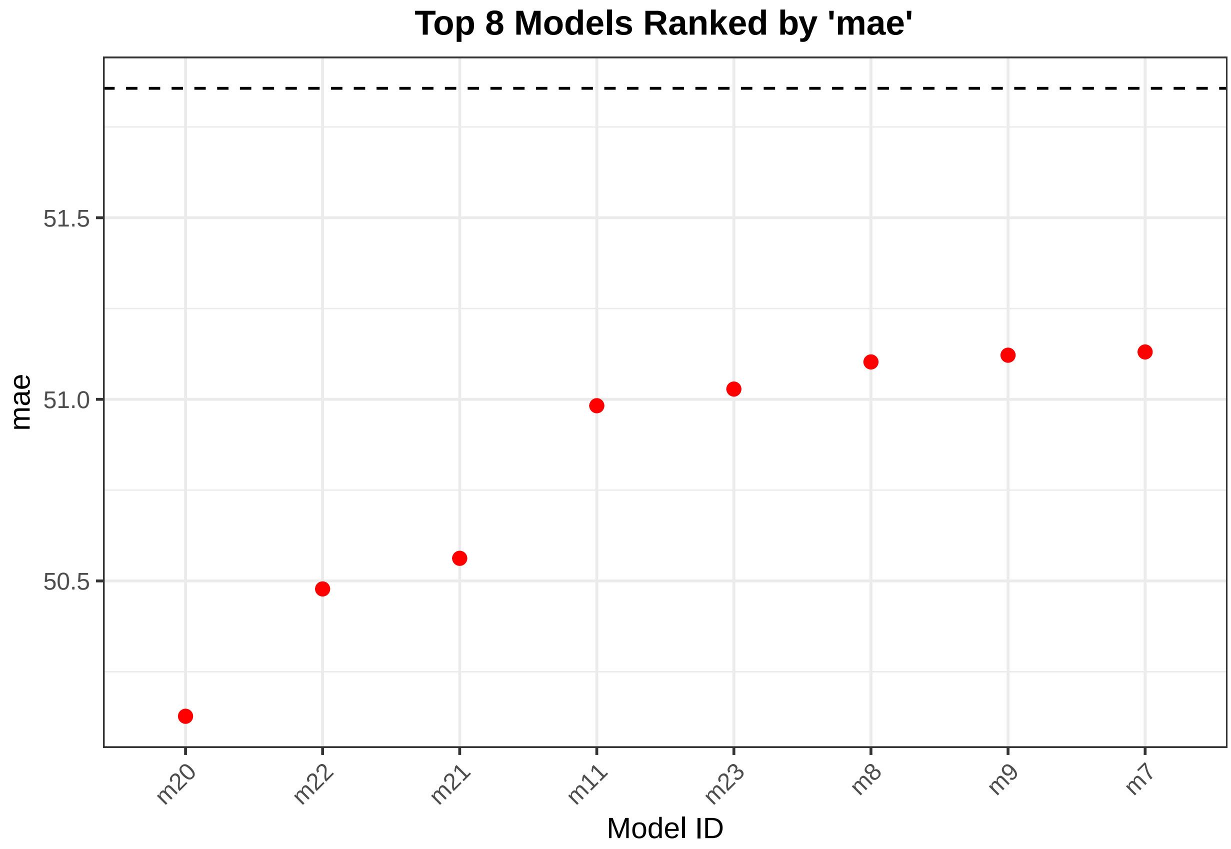


```
m_uni_gof %>%
  dplyr::filter(model_id %in% rank_uni_crps)%>%
  dplyr::select(model_id, covariate_1, crps)%>%
  dplyr::arrange(crps)
```

```
##    model_id        covariate_1     crps
## 1        m4             tas.l1 5.416537
## 2        m5             tas.l2 5.434750
## 3        m8             tas.l5 5.461306
## 4        m7             tas.l4 5.462259
## 5        m11              tas.l8 5.462273
## 6         m2                 tas 5.479605
## 7        m31 prlr.l4_nl_q10_rw2 5.489607
## 8         m9              tas.l6 5.492496
## 9        m12             prlr.l1 5.515304
## 10        m6              tas.l3 5.520986
```

MAE

```
rank_uni_mae <- ghrmodel::rank_models(
  models = m_uni,
  metric = "mae",
  plot = TRUE,
  n = 8,
  intercept = TRUE
)
```



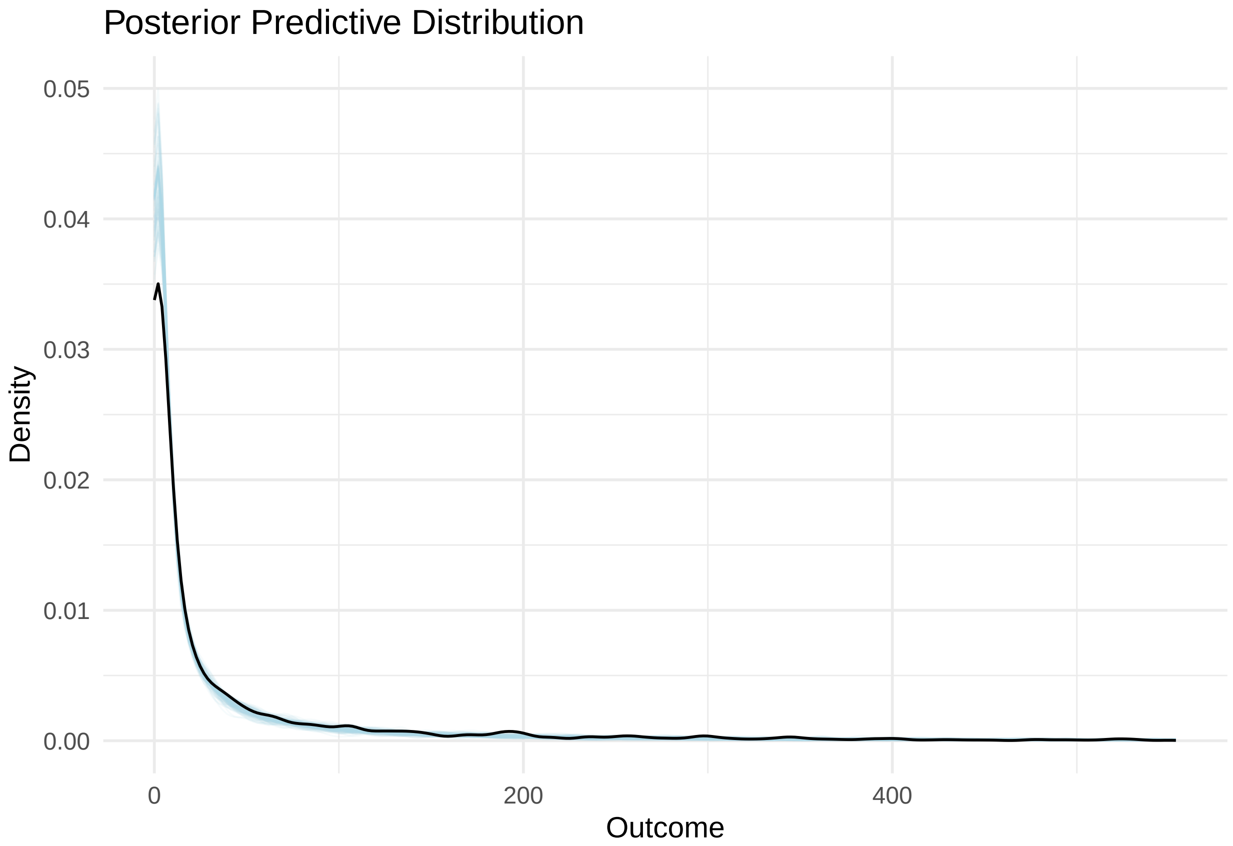
```
m_uni_gof %>%
  dplyr::filter(model_id %in% rank_uni_mae)%>%
  dplyr::select(model_id, covariate_1, mae)%>%
  dplyr::arrange(mae)
```

```
##   model_id        covariate_1      mae
## 1      m20 tas.l1_nl_q10_rw2 50.12735
## 2      m22 tas.l3_nl_q10_rw2 50.47793
## 3      m21 tas.l2_nl_q10_rw2 50.56224
## 4      m11             tas.l8 50.98238
## 5      m23 tas.l4_nl_q10_rw2 51.02828
## 6       m8             tas.l5 51.10294
## 7       m9             tas.l6 51.12154
## 8       m7             tas.l4 51.13045
```

Posterior prediction check of univariate models

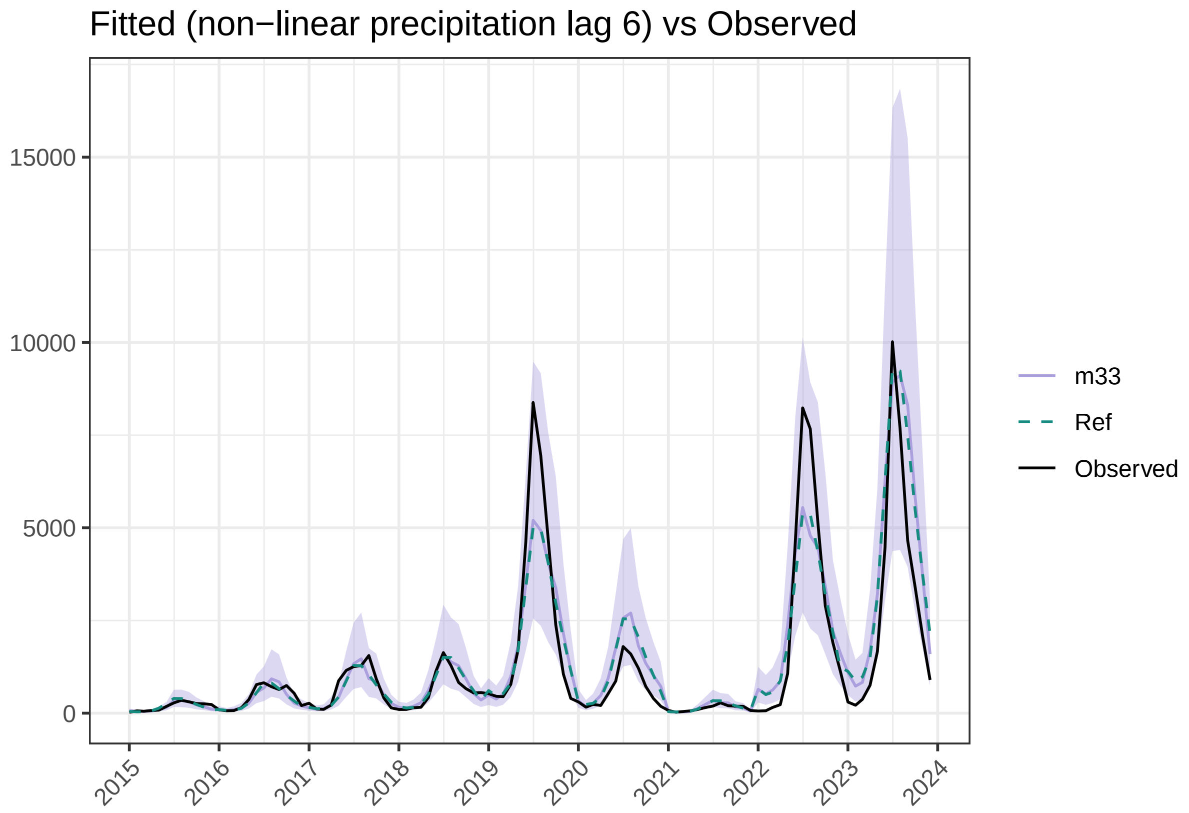
This function refits (or retrieves) the specified model, generates posterior predictions, and compares these predictions to the observed data by plotting their density estimates.

```
# Best fitting model according to WAIC
ppc_m33<-ghrmodel::post_pred_check(models = m_uni, model_id = "m33", s = 100, predictions = TRUE)
```



Fit vs. Observed plots for univariate models

```
# Plot model outputs
ghrmodel::plot_fit(
  models = m_uni,
  time = "date",
  model_id = "m33",
  model_ref = "m1",
  #area = "province",
  selected_area = NULL,
  title = "Fitted (non-linear precipitation lag 6) vs Observed"
)
```



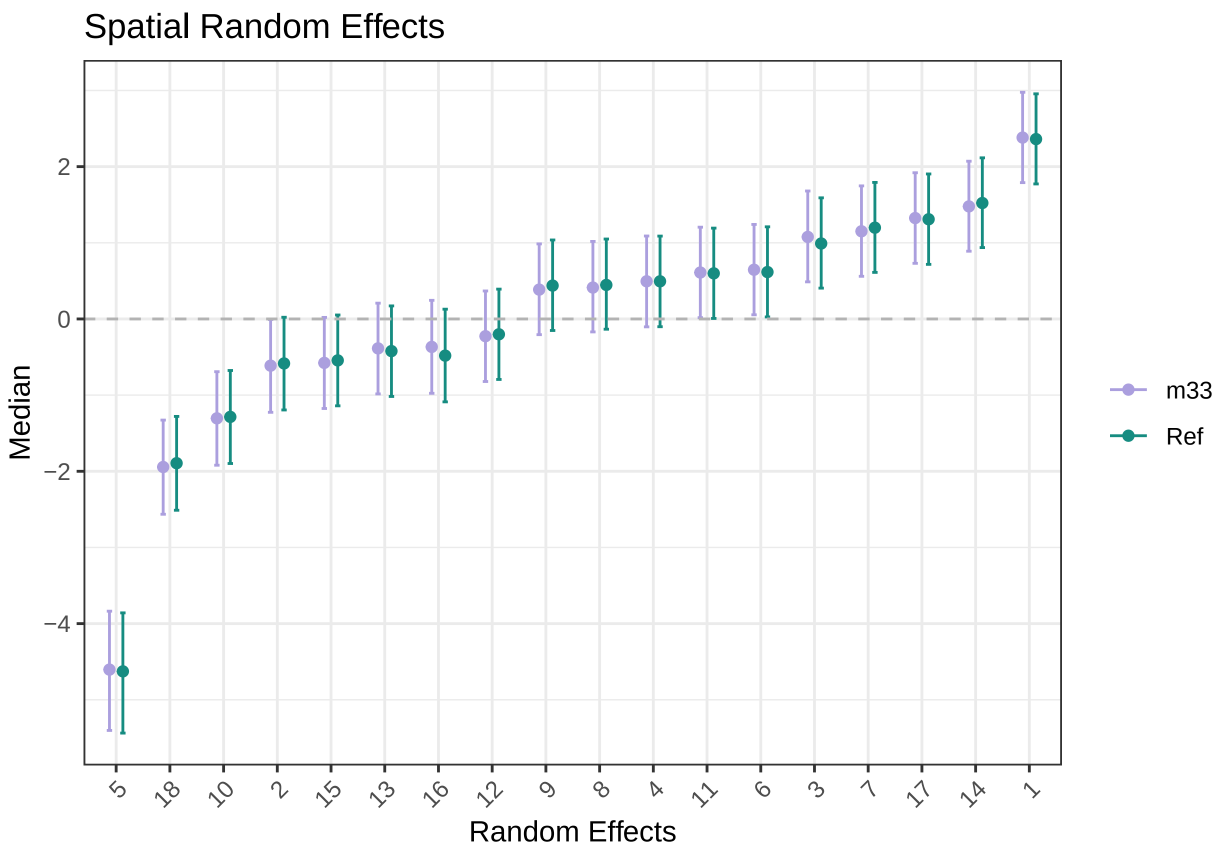
Pretty similar to the only random effect model.

Plot random effects for univariate models

Spatial random effects

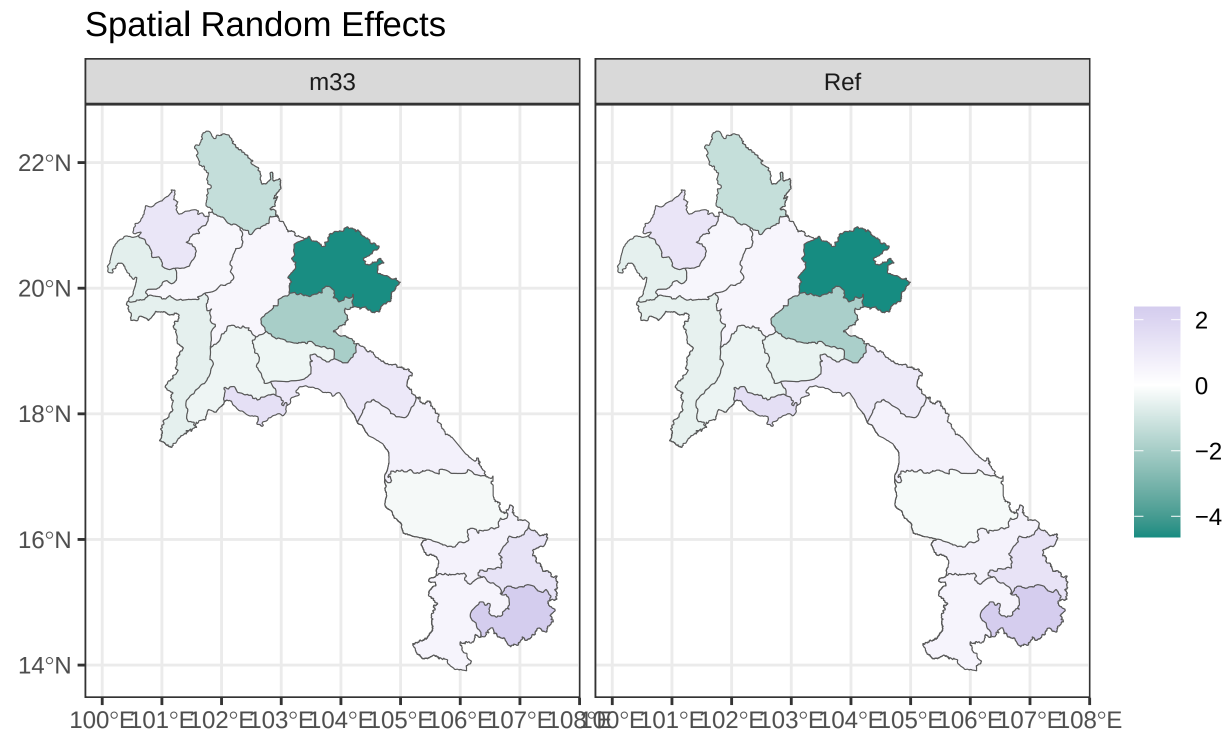
- Coeffecient plot

```
ghrmodel::plot_re_sp(
  models = m_uni,
  model_id = "m33",
  model_ref = "m1",
 # map = MS_map,
 #  map_area = "code",
  re_id = "province_id",
  label_model = NULL,
  title = "Spatial Random Effects"
)
```



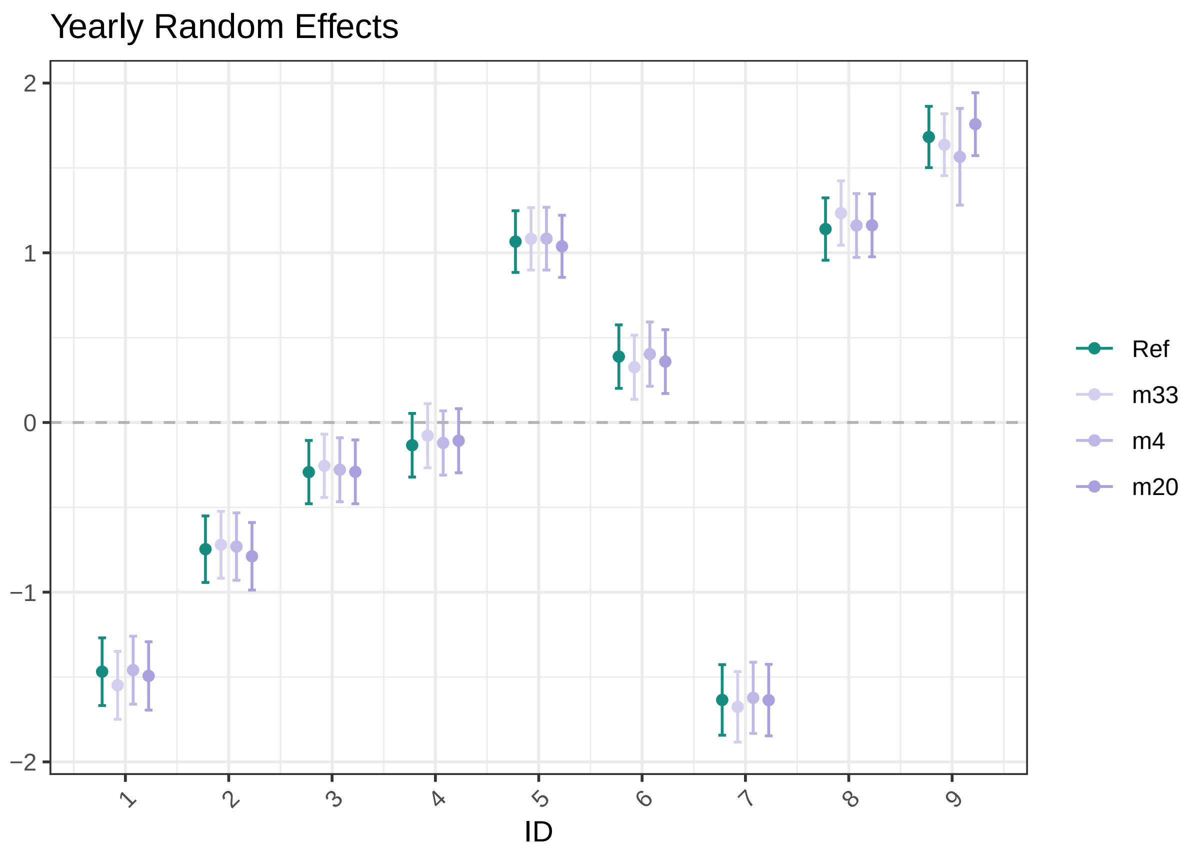
- Map plot

```
ghrmodel::plot_re_sp(
  models = m_uni,
  model_id = "m33",
  model_ref = "m1",
  map = laos,
  map_area = "NAME_1",
  re_id = "province_id",
  label_model = NULL,
  title = "Spatial Random Effects"
)
```


Spatial Random Effects
m33
Ref
22°N
20°N
18°N
16°N
14°N
100°E 101°E 102°E 103°E 104°E 105°E 106°E 107°E 108°E
100°E 101°E 102°E 103°E 104°E 105°E 106°E 107°E 108°E
2
0
−2
−4

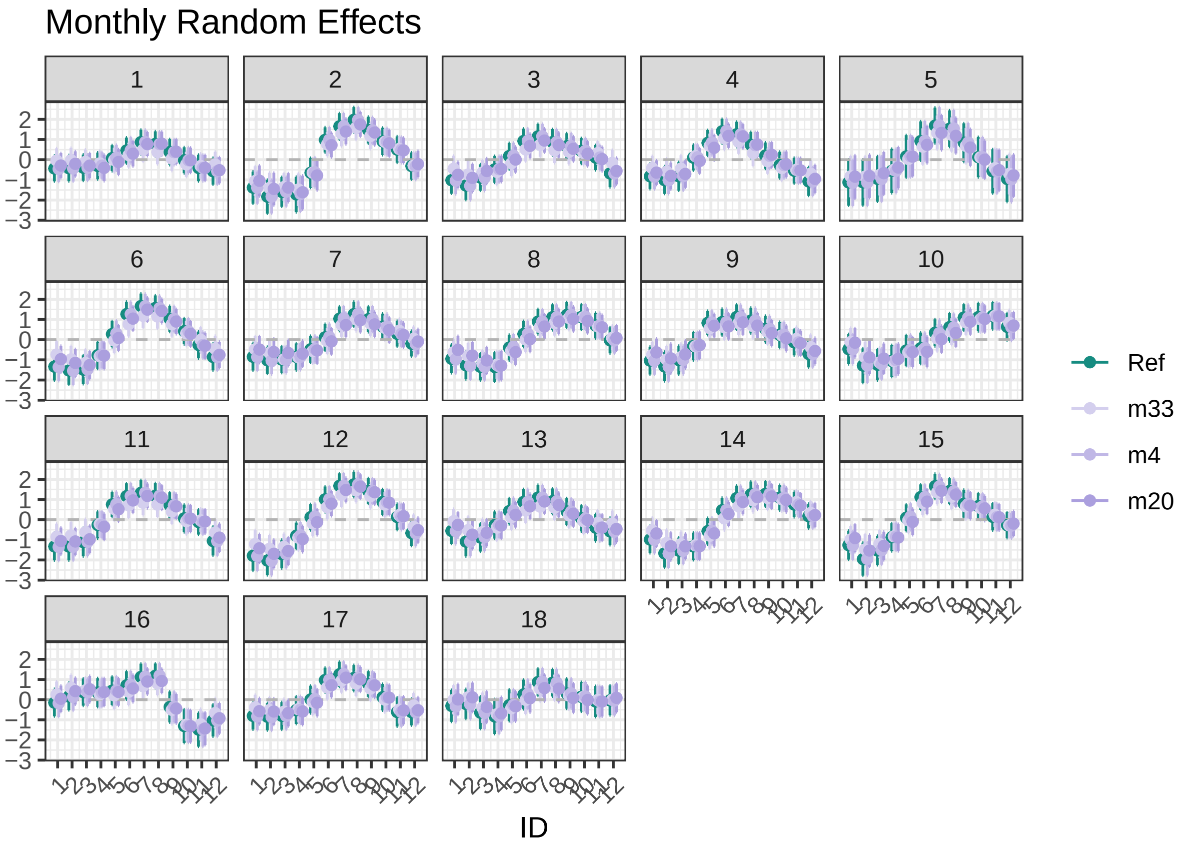
Yearly random effects

```
ghrmodel::plot_re_t(
  models = m_uni,
  model_ids = c("m33", "m4", "m20"),
  model_ref = "m1",
  re_id = "year_id",
  label_model = NULL,
  title = "Yearly Random Effects"
)
```



Monthly random effects

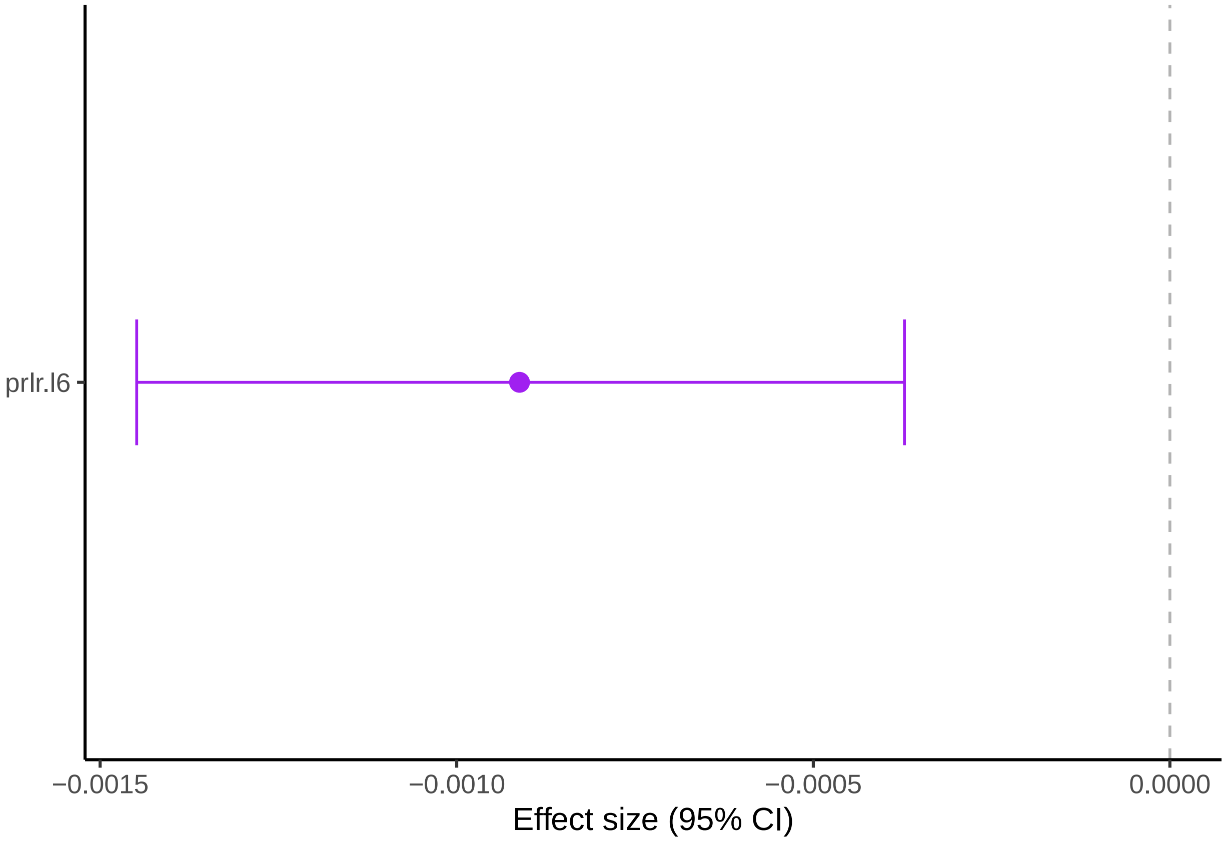
```
ghrmodel::plot_re_t(
  models = m_uni,
  model_ids = c("m33", "m4", "m20"),
  model_ref = "m1",
  re_id = "month_id",
  replicated_id = "province_id",
  label_model = NULL,
  title = "Monthly Random Effects"
)
```



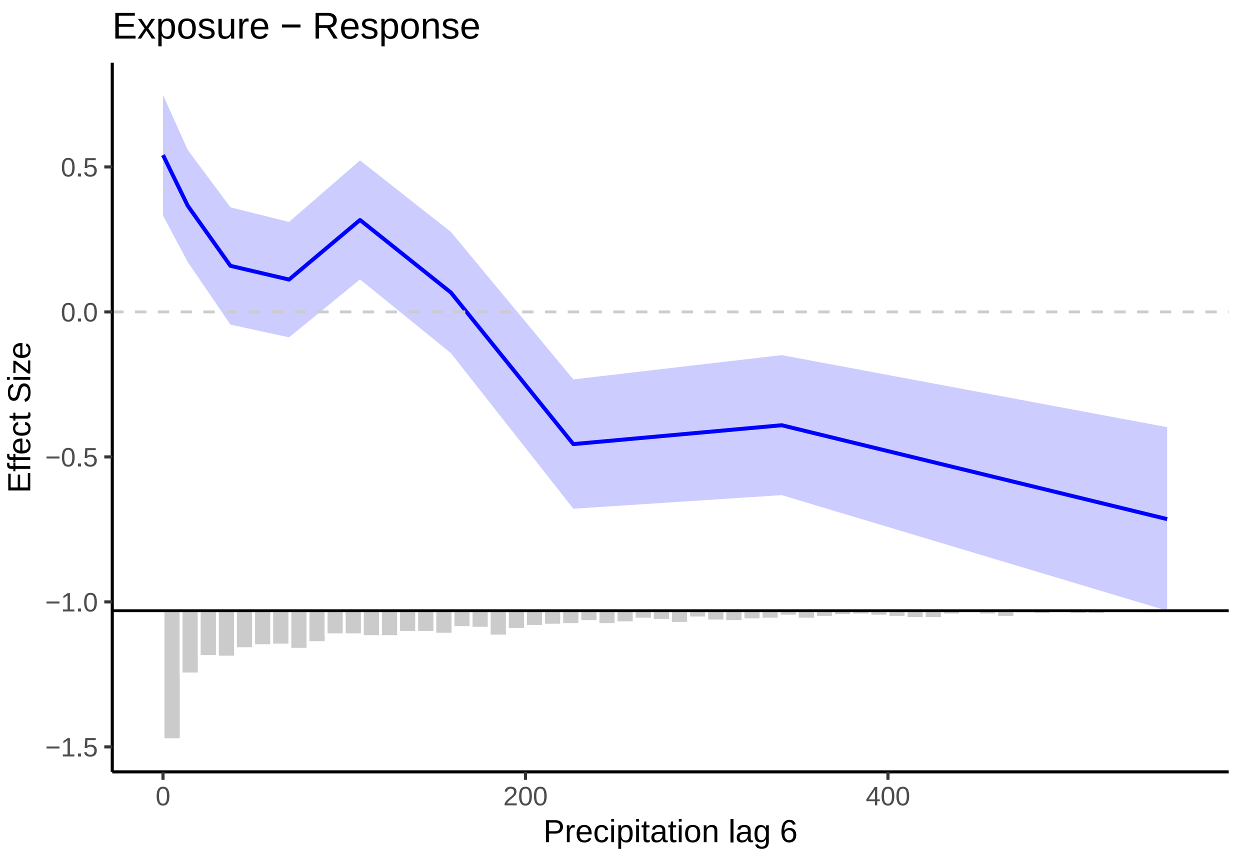
Plot fixed effects for univariate models

Linear coefficients

```r
# linear coefficient of precipitation at 6 month lag
ghrmodel::plot_lin_coef(
  models = m_uni,
  model_id = "m17",
  color = "purple"
)
```



```r
# non- linear coefficient of precipitation at 6 month lag

ghrmodel::plot_nl_coef(
  models = m_uni,
  model_id = "m33",
  var = "prlr.l6",
  title = "Exposure - Response",
  var_label = "Precipitation lag 6",
  color = "blue",
  show_hist = TRUE
)
```



This suggests that when there were small amounts of precipitation 6 months previously, this can lead to an increase in dengue incidence

Multivariate predictors

From the univariate models we learned that non-linear precipitation lagged at 5-8 months fits the model well, as well as linear and non-linear average temperature lagged 0-8 months, so we will test models with combinations of these covariates.

Write multivariate model functions

```
# Create a list of combined multivariate predictors
cov_multi_list <- ghrmodel::combine_covariates(covariates = cov_uni_l,
                                    pattern=c("tas.l",
                                              "prlr.l"))

dplyr::glimpse(cov_multi_list)
```

```
## List of 64
##  $ : chr [1:2] "tas.l1" "prlr.l1"
##  $ : chr [1:2] "tas.l2" "prlr.l1"
##  $ : chr [1:2] "tas.l3" "prlr.l1"
##  $ : chr [1:2] "tas.l4" "prlr.l1"
##  $ : chr [1:2] "tas.l5" "prlr.l1"
##  $ : chr [1:2] "tas.l6" "prlr.l1"
##  $ : chr [1:2] "tas.l7" "prlr.l1"
##  $ : chr [1:2] "tas.l8" "prlr.l1"
##  $ : chr [1:2] "tas.l1" "prlr.l2"
##  $ : chr [1:2] "tas.l2" "prlr.l2"
##  $ : chr [1:2] "tas.l3" "prlr.l2"
##  $ : chr [1:2] "tas.l4" "prlr.l2"
##  $ : chr [1:2] "tas.l5" "prlr.l2"
##  $ : chr [1:2] "tas.l6" "prlr.l2"
##  $ : chr [1:2] "tas.l7" "prlr.l2"
##  $ : chr [1:2] "tas.l8" "prlr.l2"
##  $ : chr [1:2] "tas.l1" "prlr.l3"
##  $ : chr [1:2] "tas.l2" "prlr.l3"
##  $ : chr [1:2] "tas.l3" "prlr.l3"
##  $ : chr [1:2] "tas.l4" "prlr.l3"
##  $ : chr [1:2] "tas.l5" "prlr.l3"
##  $ : chr [1:2] "tas.l6" "prlr.l3"
##  $ : chr [1:2] "tas.l7" "prlr.l3"
##  $ : chr [1:2] "tas.l8" "prlr.l3"
##  $ : chr [1:2] "tas.l1" "prlr.l4"
##  $ : chr [1:2] "tas.l2" "prlr.l4"
##  $ : chr [1:2] "tas.l3" "prlr.l4"
##  $ : chr [1:2] "tas.l4" "prlr.l4"
##  $ : chr [1:2] "tas.l5" "prlr.l4"
##  $ : chr [1:2] "tas.l6" "prlr.l4"
##  $ : chr [1:2] "tas.l7" "prlr.l4"
##  $ : chr [1:2] "tas.l8" "prlr.l4"
##  $ : chr [1:2] "tas.l1" "prlr.l5"
##  $ : chr [1:2] "tas.l2" "prlr.l5"
##  $ : chr [1:2] "tas.l3" "prlr.l5"
##  $ : chr [1:2] "tas.l4" "prlr.l5"
##  $ : chr [1:2] "tas.l5" "prlr.l5"
##  $ : chr [1:2] "tas.l6" "prlr.l5"
##  $ : chr [1:2] "tas.l7" "prlr.l5"
##  $ : chr [1:2] "tas.l8" "prlr.l5"
##  $ : chr [1:2] "tas.l1" "prlr.l6"
##  $ : chr [1:2] "tas.l2" "prlr.l6"
##  $ : chr [1:2] "tas.l3" "prlr.l6"
##  $ : chr [1:2] "tas.l4" "prlr.l6"
##  $ : chr [1:2] "tas.l5" "prlr.l6"
##  $ : chr [1:2] "tas.l6" "prlr.l6"
##  $ : chr [1:2] "tas.l7" "prlr.l6"
##  $ : chr [1:2] "tas.l8" "prlr.l6"
##  $ : chr [1:2] "tas.l1" "prlr.l7"
##  $ : chr [1:2] "tas.l2" "prlr.l7"
##  $ : chr [1:2] "tas.l3" "prlr.l7"
##  $ : chr [1:2] "tas.l4" "prlr.l7"
##  $ : chr [1:2] "tas.l5" "prlr.l7"
##  $ : chr [1:2] "tas.l6" "prlr.l7"
##  $ : chr [1:2] "tas.l7" "prlr.l7"
##  $ : chr [1:2] "tas.l8" "prlr.l7"
##  $ : chr [1:2] "tas.l1" "prlr.l8"
##  $ : chr [1:2] "tas.l2" "prlr.l8"
##  $ : chr [1:2] "tas.l3" "prlr.l8"
##  $ : chr [1:2] "tas.l4" "prlr.l8"
##  $ : chr [1:2] "tas.l5" "prlr.l8"
##  $ : chr [1:2] "tas.l6" "prlr.l8"
##  $ : chr [1:2] "tas.l7" "prlr.l8"
##  $ : chr [1:2] "tas.l8" "prlr.l8"
```

```
# Create a list of combined multivariate predictors, with non-linear precipitation

cov_multi_nl_list <- ghrmodel::non_linear_covariates(covariates = cov_multi_list,
                                           method = "quantile",
                                           pattern = c("prlr.l", "tas.l"),
                                           n = 10)
dplyr::glimpse(cov_multi_nl_list)
```

```
## List of 64
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l6, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l7, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
##  $ : chr [1:2] "f(INLA::inla.group(tas.l8, method='quantile', n=10), model='rw2')" "f(INLA::inla.grou
```

Write multivariate model formulas

```
cov_multi_nl_formulas <- write_inla_formulas(outcome = "dengue_cases",
                                             covariates = cov_multi_nl_list ,
                                             re1 = list(id ="month_id",
                                                        re ="rw1", cyclic = TRUE,
                                                        hyper = "prior_re1",
                                                        replicate = "province_id" ),
                                             re2 = list(id = "year_id",
                                                        re = "rw1",
                                                        hyper = "prior_re1"),
                                             re3 = list(id = "province_id",
                                                        re = "bym2",
                                                        graph = "g",
                                                        hyper = "prior_re2"),
                                                        baseline = TRUE)


head(cov_multi_nl_formulas)
```

```
## [1] "dengue_cases ~ 1 + f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, constr = T
## [2] "dengue_cases ~ 1 + f(INLA::inla.group(tas.l1, method='quantile', n=10), model='rw2') + f(INLA::i
## [3] "dengue_cases ~ 1 + f(INLA::inla.group(tas.l2, method='quantile', n=10), model='rw2') + f(INLA::i
## [4] "dengue_cases ~ 1 + f(INLA::inla.group(tas.l3, method='quantile', n=10), model='rw2') + f(INLA::i
## [5] "dengue_cases ~ 1 + f(INLA::inla.group(tas.l4, method='quantile', n=10), model='rw2') + f(INLA::i
## [6] "dengue_cases ~ 1 + f(INLA::inla.group(tas.l5, method='quantile', n=10), model='rw2') + f(INLA::i
# transform formulas list into a GHRformulas object

cov_multi_nl_formulas_ghr <- ghrmodel::as_GHRformulas(formulas = cov_multi_nl_formulas)

class(cov_multi_nl_formulas_ghr)
```

```
## [1] "GHRformulas" "list"
```

```
str(cov_multi_nl_formulas_ghr)
```

```
## List of 4
##  $ formulas: chr [1:65] "dengue_cases ~ 1 + f(month_id, model = 'rw1', replicate = province_id, cycli
##  $ vars    :'data.frame':    65 obs. of  2 variables:
##   ..$ covariate_1: chr [1:65] NA "tas.l1_nl_q10_rw2" "tas.l2_nl_q10_rw2" "tas.l3_nl_q10_rw2" ...
##   ..$ covariate_2: chr [1:65] NA "prlr.l1_nl_q10_rw2" "prlr.l1_nl_q10_rw2" "prlr.l1_nl_q10_rw2" ...
##  $ re      : Named chr [1:3] "f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, cons
##   ..- attr(*, "names")= chr [1:3] "re_1" "re_2" "re_3"
##  $ outcome : chr "dengue_cases"
##  - attr(*, "class")= chr [1:2] "GHRformulas" "list"
```

```
dplyr::glimpse(cov_multi_nl_formulas_ghr$vars)
```

```
## Rows: 65
## Columns: 2
## $ covariate_1 <chr> NA, "tas.l1_nl_q10_rw2", "tas.l2_nl_q10_rw2", "tas.l3_nl_q10_rw2", "tas.l4_nl_q1
## $ covariate_2 <chr> NA, "prlr.l1_nl_q10_rw2", "prlr.l1_nl_q10_rw2", "prlr.l1_nl_q10_rw2", "prlr.l1_nl
```

Fit multivariate models

```
m_multi <- ghrmodel::fit_models(formulas = cov_multi_nl_formulas_ghr  ,
                           data = data,
                           family = "nbinomial",      # specify family
                           name = "mm",
                           offset = "population",
                           config = TRUE,
                           pb = TRUE,
                           nthreads = 8)
```

```
##   |                                                                          |
## Model 1 of 65 total run time 0.11 minutes.
##   |                                                                          |==
## Model 2 of 65 total run time 0.28 minutes.
##   |                                                                          |===
## Model 3 of 65 total run time 0.44 minutes.
##   |                                                                          |====
## Model 4 of 65 total run time 0.61 minutes.
##   |                                                                          |=====
## Model 5 of 65 total run time 0.77 minutes.
##   |                                                                          |======
## Model 6 of 65 total run time 0.94 minutes.
##   |                                                                         |=======
## Model 7 of 65 total run time 1.1 minutes.
##   |                                                                         |========
## Model 8 of 65 total run time 1.27 minutes.
##   |                                                                         |=========
## Model 9 of 65 total run time 1.43 minutes.
##   |                                                                         |==========
## Model 10 of 65 total run time 1.6 minutes.
##   |                                                                         |===========
## Model 11 of 65 total run time 1.76 minutes.
##   |                                                                         |============
## Model 12 of 65 total run time 1.91 minutes.
##   |                                                                         |=============
## Model 13 of 65 total run time 2.09 minutes.
##   |                                                                         |==============
## Model 14 of 65 total run time 2.25 minutes.
##   |                                                                         |===============
## Model 15 of 65 total run time 2.41 minutes.
##   |                                                                         |================
## Model 16 of 65 total run time 2.58 minutes.
##   |                                                                         |=================
## Model 17 of 65 total run time 2.74 minutes.
##   |                                                                         |==================
## Model 18 of 65 total run time 2.91 minutes.
##   |                                                                         |===================
## Model 19 of 65 total run time 3.09 minutes.
##   |                                                                         |====================
## Model 20 of 65 total run time 3.24 minutes.
##   |                                                                         |=====================
## Model 21 of 65 total run time 3.4 minutes.
##   |                                                                         |======================
## Model 22 of 65 total run time 3.58 minutes.
##   |                                                                         |=======================
## Model 23 of 65 total run time 3.75 minutes.
##   |                                                                         |=======================
## Model 24 of 65 total run time 3.91 minutes.
##   |                                                                         |=======================
## Model 25 of 65 total run time 4.08 minutes.
##   |                                                                         |=======================
## Model 26 of 65 total run time 4.24 minutes.
##   |                                                                         |=======================
## Model 27 of 65 total run time 4.41 minutes.
##   |                                                                         |=======================
## Model 28 of 65 total run time 4.58 minutes.
##   |                                                                         |=======================
## Model 29 of 65 total run time 4.75 minutes.
##   |                                                                         |=======================
## Model 30 of 65 total run time 4.92 minutes.
##   |                                                                         |=======================
## Model 31 of 65 total run time 5.08 minutes.
##   |                                                                         |=======================
## Model 32 of 65 total run time 5.25 minutes.
##   |                                                                         |=======================
## Model 33 of 65 total run time 5.42 minutes.
##   |                                                                           |========================
## Model 34 of 65 total run time 5.59 minutes.
##   |                                                                           |========================
## Model 35 of 65 total run time 5.74 minutes.
##   |                                                                           |========================
## Model 36 of 65 total run time 5.9 minutes.
##   |                                                                           |========================
## Model 37 of 65 total run time 6.07 minutes.
##   |                                                                           |========================
## Model 38 of 65 total run time 6.22 minutes.
##   |                                                                           |========================
## Model 39 of 65 total run time 6.41 minutes.
##   |                                                                           |========================
## Model 40 of 65 total run time 6.56 minutes.
##   |                                                                           |========================
## Model 41 of 65 total run time 6.73 minutes.
##   |                                                                           |========================
## Model 42 of 65 total run time 6.89 minutes.
##   |                                                                           |========================
## Model 43 of 65 total run time 7.06 minutes.
##   |                                                                           |========================
## Model 44 of 65 total run time 7.22 minutes.
##   |                                                                           |========================
## Model 45 of 65 total run time 7.4 minutes.
##   |                                                                           |========================
## Model 46 of 65 total run time 7.57 minutes.
##   |                                                                           |========================
## Model 47 of 65 total run time 7.73 minutes.
##   |                                                                           |========================
## Model 48 of 65 total run time 7.89 minutes.
##   |                                                                           |========================
## Model 49 of 65 total run time 8.06 minutes.
##   |                                                                           |========================
## Model 50 of 65 total run time 8.23 minutes.
##   |                                                                           |========================
## Model 51 of 65 total run time 8.41 minutes.
##   |                                                                           |========================
## Model 52 of 65 total run time 8.58 minutes.
##   |                                                                           |========================
## Model 53 of 65 total run time 8.76 minutes.
##   |                                                                           |========================
## Model 54 of 65 total run time 8.92 minutes.
##   |                                                                           |========================
## Model 55 of 65 total run time 9.09 minutes.
##   |                                                                           |========================
## Model 56 of 65 total run time 9.27 minutes.
##   |                                                                           |========================
## Model 57 of 65 total run time 9.46 minutes.
##   |                                                                           |========================
## Model 58 of 65 total run time 9.64 minutes.
##   |                                                                           |========================
## Model 59 of 65 total run time 9.81 minutes.
##   |                                                                           |========================
## Model 60 of 65 total run time 9.98 minutes.
##   |                                                                          |=======================
## Model 61 of 65 total run time 10.14 minutes.
##   |                                                                          |=======================
## Model 62 of 65 total run time 10.32 minutes.
##   |                                                                          |=======================
## Model 63 of 65 total run time 10.48 minutes.
##   |                                                                          |=======================
## Model 64 of 65 total run time 10.65 minutes.
##   |                                                                          |=======================
## Model 65 of 65 total run time 10.81 minutes.
```

```
saveRDS(m_multi, "m_multi.rds")
```

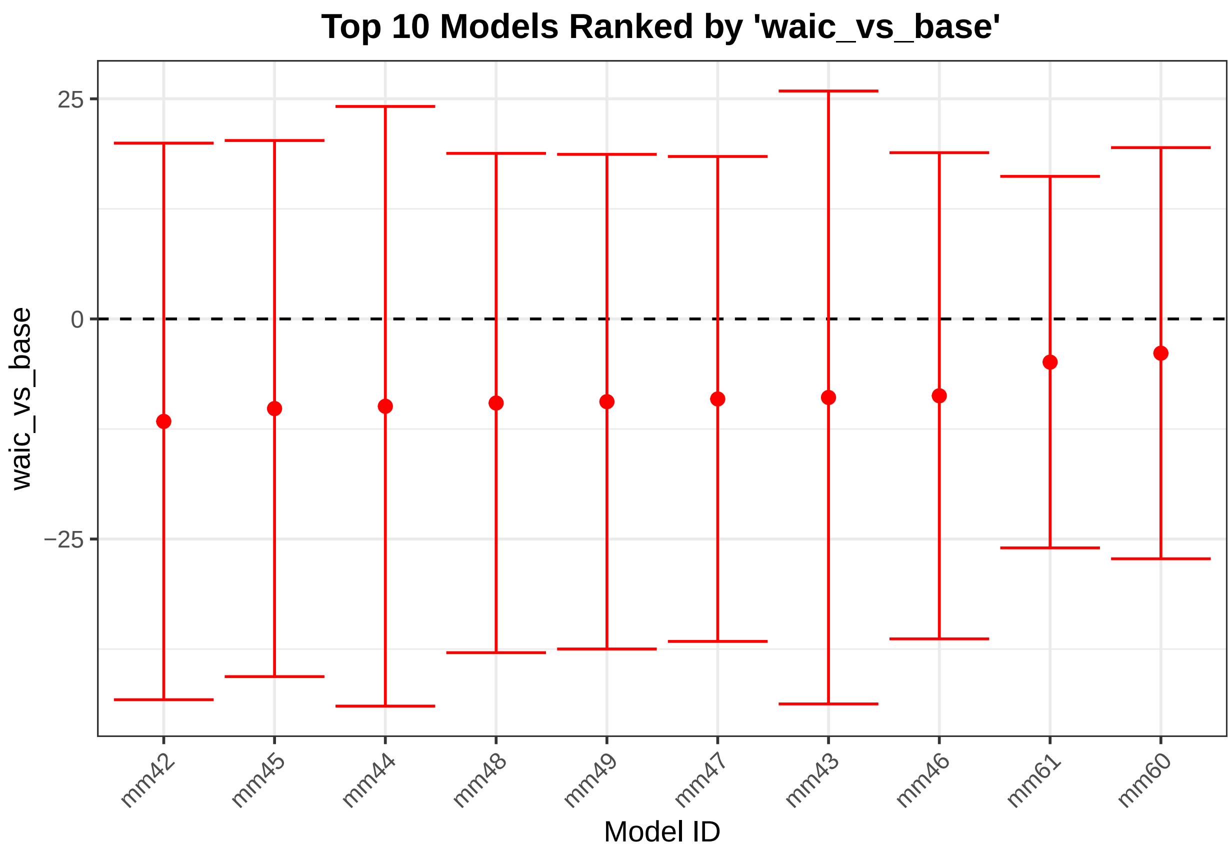
```
m_multi <- readRDS("m_multi.rds")
```

Rank multivariate models

```
# goodness of fit metrics in a dataframe
m_multi_gof <- m_multi$mod.gof

m_multi_gof <- m_multi_gof %>%
  dplyr::arrange(waic)%>%
  dplyr::mutate(rank_waic = dense_rank(waic)) %>%
  dplyr::arrange(crps)%>%
  dplyr::mutate(rank_crps = dense_rank(crps))%>%
  dplyr::arrange(mae)%>%
  dplyr::mutate(rank_mae = dense_rank(mae))
```

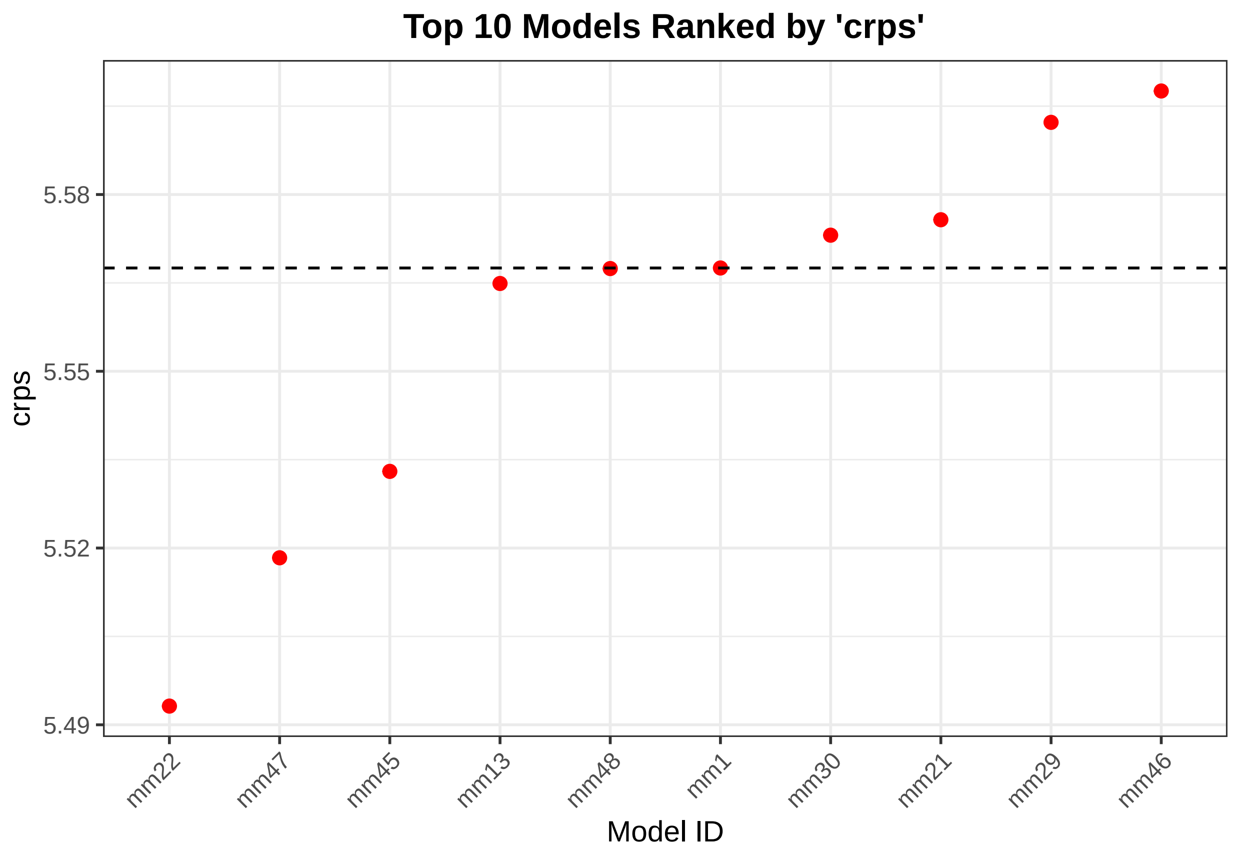
WAIC

```
rank_multi_waic_vs_base <- ghrmodel::rank_models(
  models = m_multi,
  metric = "waic_vs_base",
  plot = TRUE,
  n = 10,
  intercept = TRUE,
  ci=TRUE
)
```



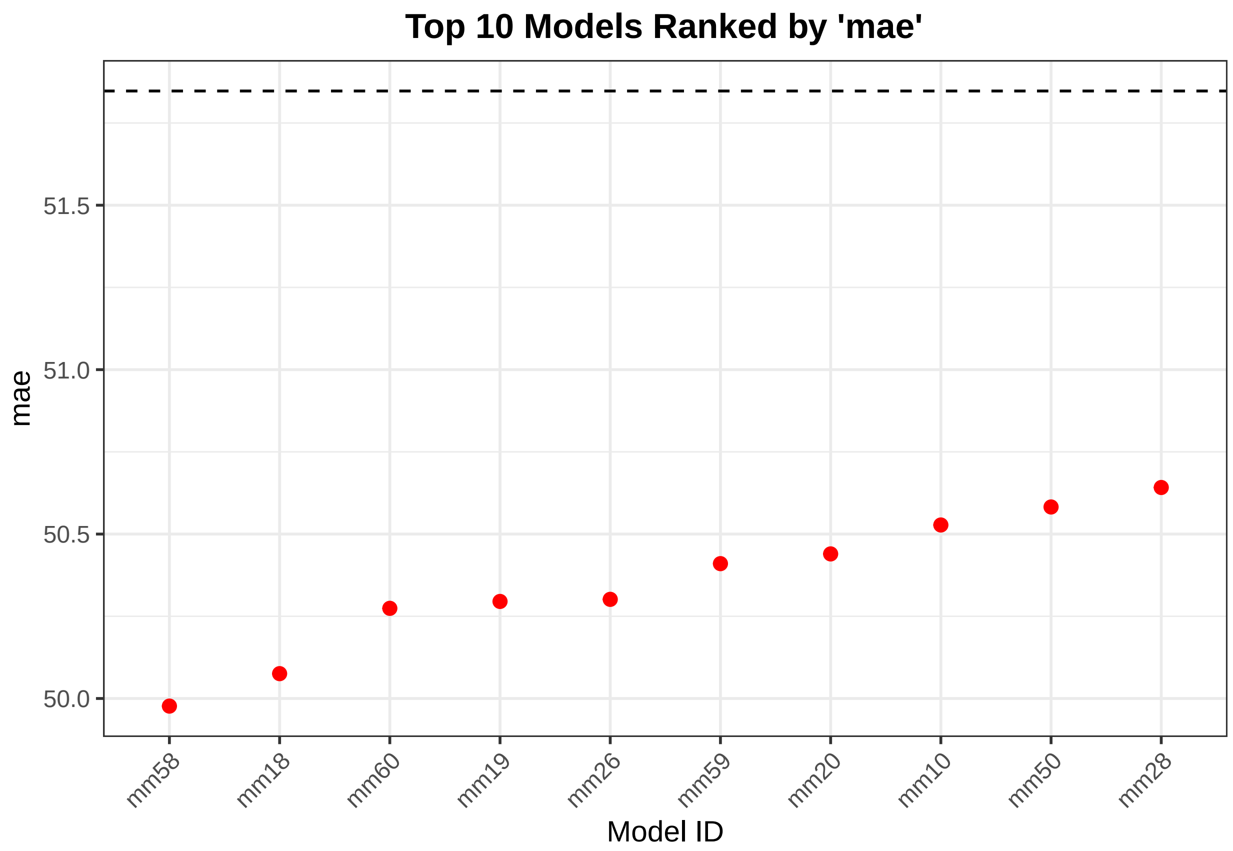
CRPS

```r
rank_multi_crps <- ghrmodel::rank_models(
  models = m_multi,
  metric = "crps",
  plot = TRUE,
  n = 10,
  intercept = TRUE
)
```



MAE

```
rank_multi_mae <- ghrmodel::rank_models(
  models = m_multi,
  metric = "mae",
  plot = TRUE,
  n = 10,
  intercept = TRUE
)
```



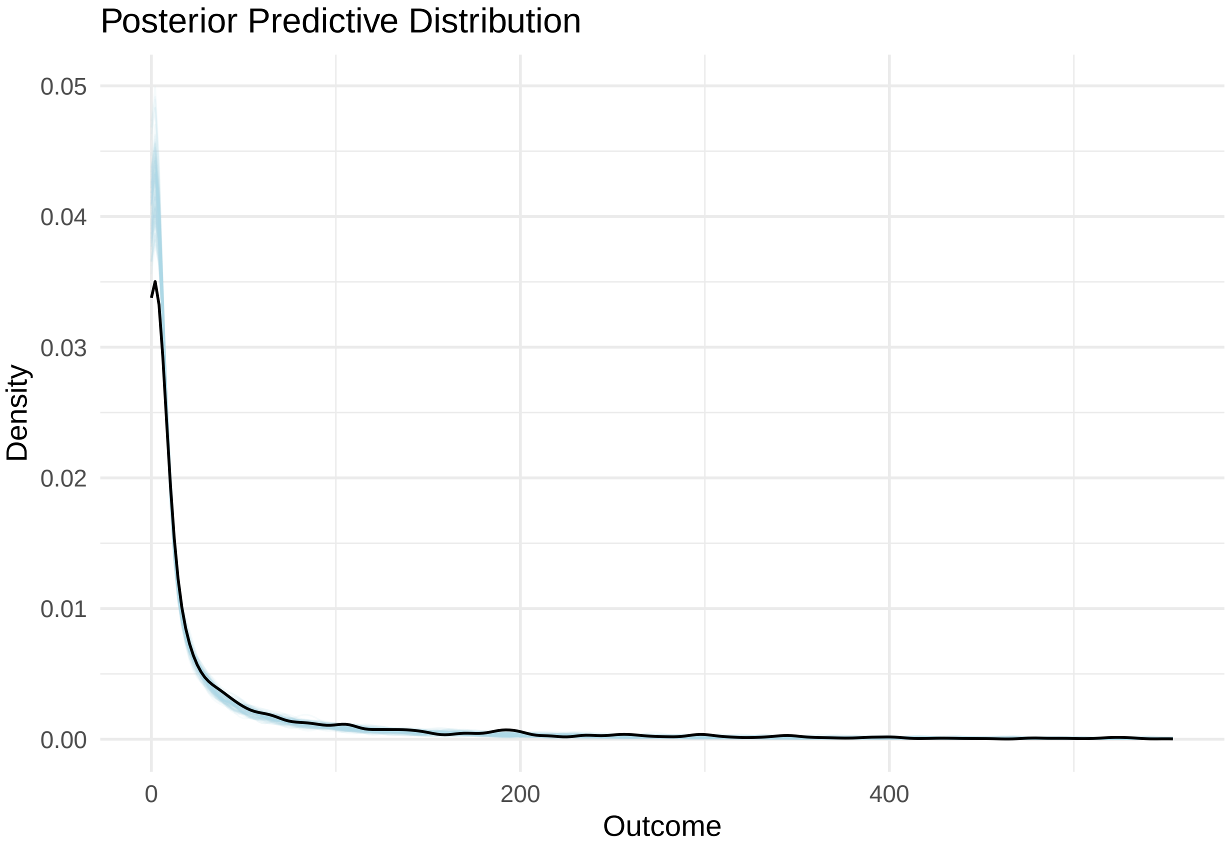
```
m_multi_gof_best <- m_multi_gof %>%
  dplyr::filter(model_id %in% rank_multi_waic_vs_base |
                model_id %in% rank_multi_crps |
                model_id %in% rank_multi_mae)%>%
  dplyr::select(model_id, covariate_1, covariate_2, waic, crps, mae,
                rank_waic, rank_mae, rank_crps)


dplyr::glimpse(m_multi_gof_best)
```

```
## Rows: 25
## Columns: 9
## $ model_id    <chr> "mm58", "mm18", "mm60", "mm19", "mm26", "mm59", "mm20", "mm10", "mm50", "mm28",
## $ covariate_1 <chr> "tas.l1_nl_q10_rw2", "tas.l1_nl_q10_rw2", "tas.l3_nl_q10_rw2", "tas.l2_nl_q10_rw
## $ covariate_2 <chr> "prlr.l8_nl_q10_rw2", "prlr.l3_nl_q10_rw2", "prlr.l8_nl_q10_rw2", "prlr.l3_nl_q1
## $ waic        <dbl> 14194.64, 14210.20, 14193.93, 14211.61, 14205.16, 14196.56, 14206.91, 14213.92,
## $ crps        <dbl> 5.885251, 5.704924, 5.724994, 5.802877, 5.767277, 5.846889, 5.630139, 5.835155,
## $ mae         <dbl> 49.97687, 50.07557, 50.27416, 50.29510, 50.30145, 50.41005, 50.43967, 50.52758,
## $ rank_waic   <int> 12, 56, 10, 59, 43, 19, 50, 63, 22, 44, 1, 9, 37, 7, 3, 46, 57, 24, 2, 35, 52, 4
## $ rank_mae    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 17, 18, 21, 22, 23, 25, 28, 31, 32, 34, 50, 5
## $ rank_crps   <int> 62, 32, 38, 52, 44, 56, 13, 54, 48, 24, 35, 20, 9, 33, 31, 8, 4, 6, 3, 7, 1, 5,
```

Posterior prediction check of multivariate models

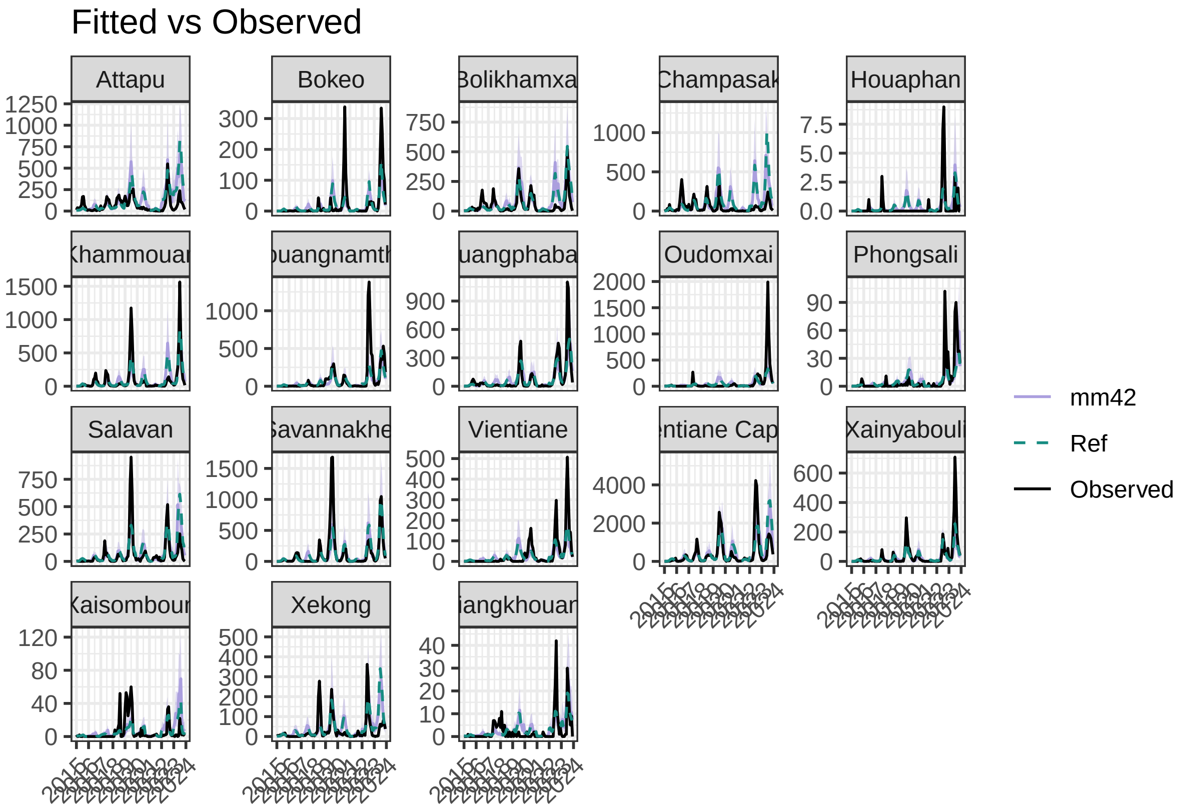
This function refits (or retrieves) the specified model, generates posterior predictions, and compares these predictions to the observed data by plotting their density estimates.

```
ppc_mm42<-ghrmodel::post_pred_check(models = m_multi, model_id = "mm42", s = 100, predictions = TRUE)
```



Fit vs. Observed plots for multivariate models

```
# Plot model outputs
ghrmodel::plot_fit(
  models = m_multi,
  time = "date",
  model_id = "mm42",
  model_ref = "mm1",
  area = "province",
  selected_area = NULL,
  title = "Fitted vs Observed"
)
```



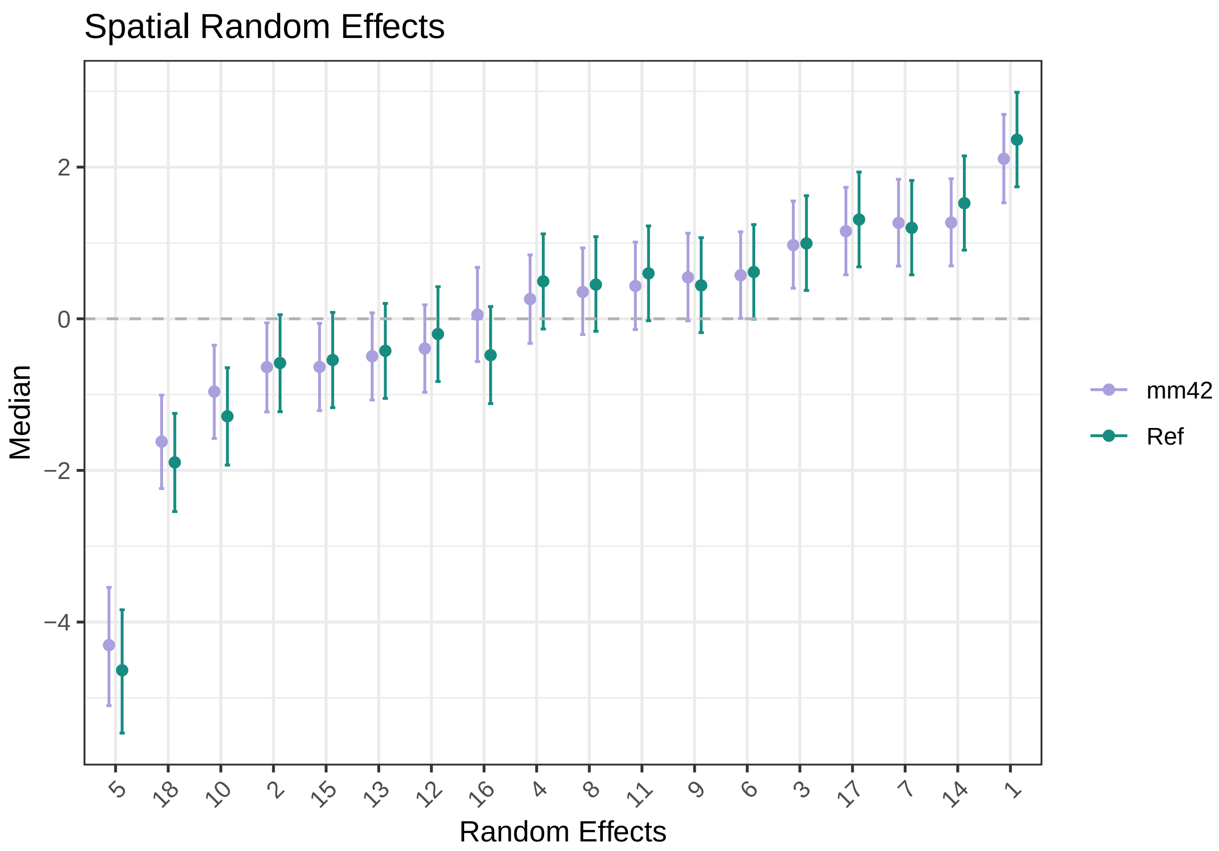
Pretty similar to the only random effect model.

Plot random effects for multivariate models

Spatial random effects

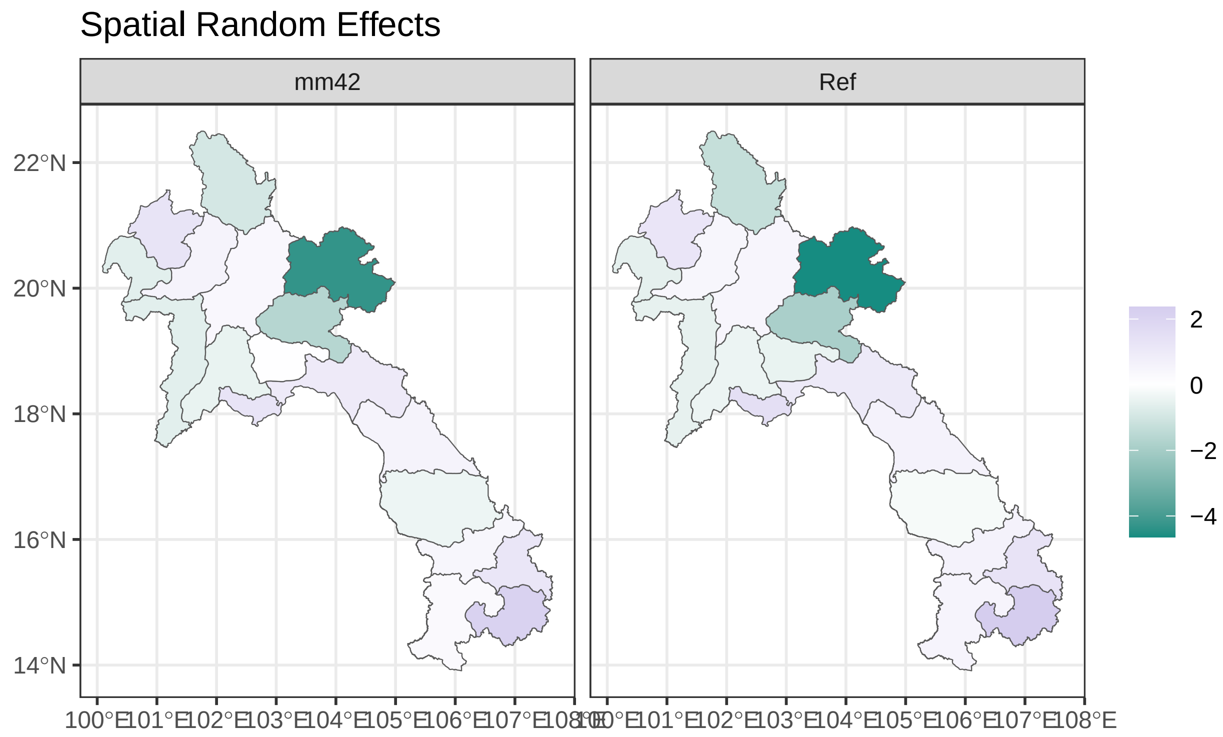
- Coeffecient plot

```
ghrmodel::plot_re_sp(
  models = m_multi,
  model_id = "mm42",
  model_ref = "mm1",
 # map = MS_map,
 #  map_area = "code",
  re_id = "province_id",
  label_model = NULL,
  title = "Spatial Random Effects"
)
```



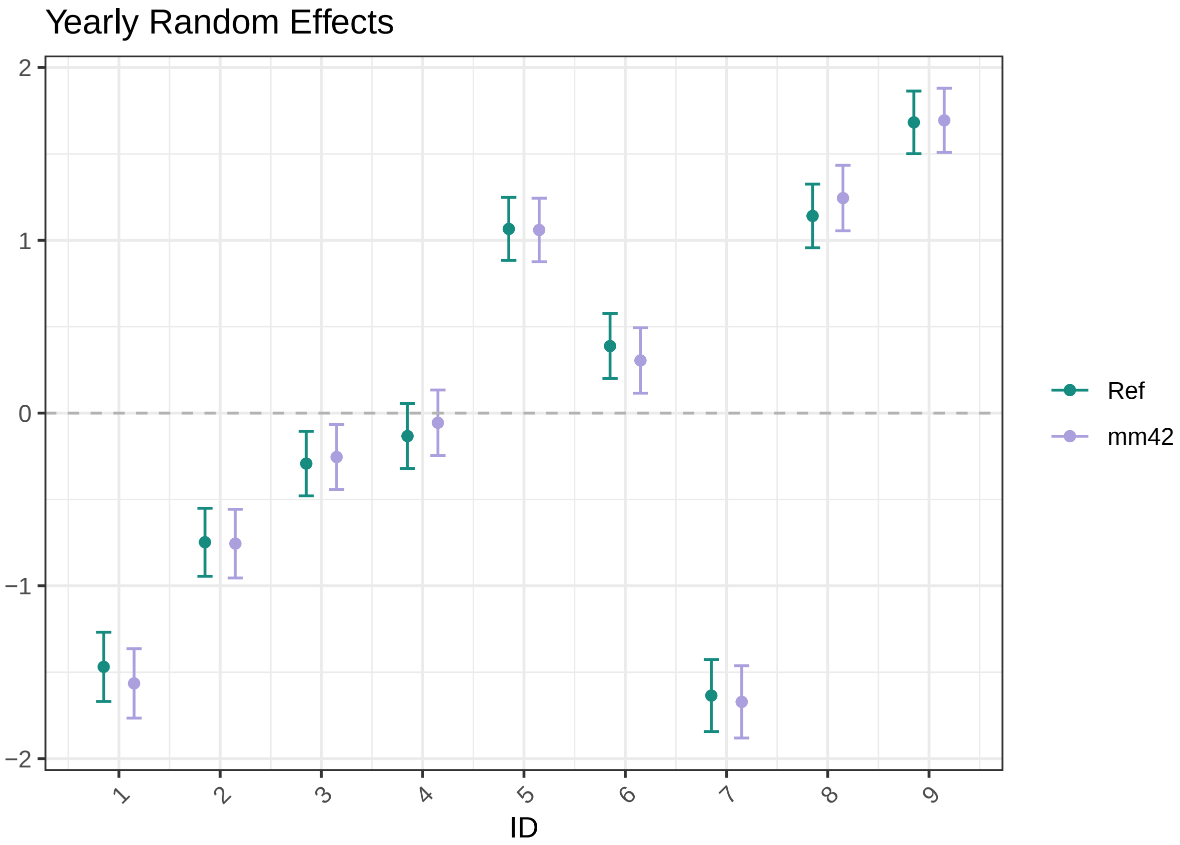
* Map plot

```
ghrmodel::plot_re_sp(
  models = m_multi,
  model_id = "mm42",
  model_ref = "mm1",
  map = laos,
  map_area = "NAME_1",
  re_id = "province_id",
  label_model = NULL,
  title = "Spatial Random Effects"
)
```



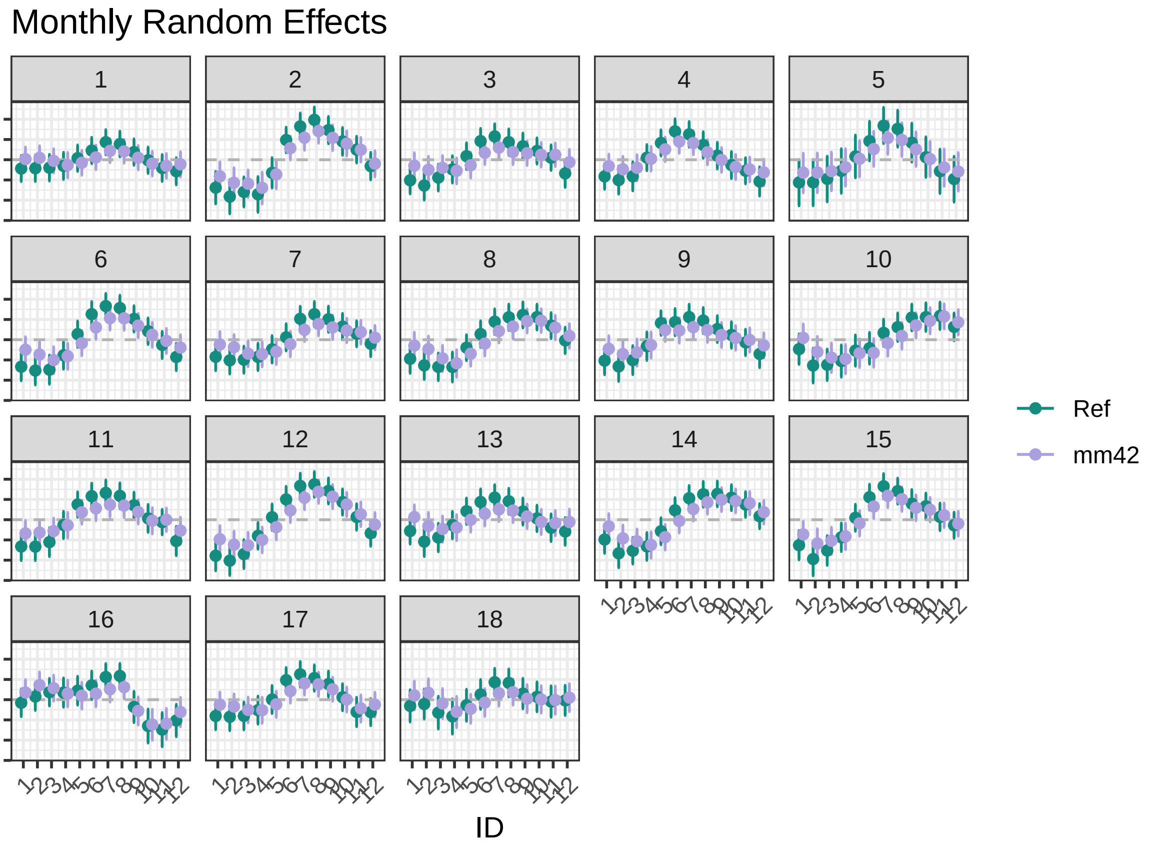
Yearly random effects

```
ghrmodel::plot_re_t(
  models = m_multi,
  model_ids = c("mm42"),
  model_ref = "mm1",
  re_id = "year_id",
  label_model = NULL,
  title = "Yearly Random Effects"
)
```



Monthly random effects

```
ghrmodel::plot_re_t(
  models = m_multi,
  model_ids = c("mm42"),
  model_ref = "mm1",
  re_id = "month_id",
  replicated_id = "province_id",
  label_model = NULL,
  title = "Monthly Random Effects"
)
```



Varying and replicated predictors

Write varying and replicated model functions

```
m_multi_cov_best<- m_multi_gof_best |>
  filter(rank_waic %in% seq(1:5) |
           rank_mae %in% seq(1:5) |
           rank_crps %in% seq(1:5))

table(m_multi_cov_best$covariate_1)
```

```
##
## tas.l1_nl_q10_rw2 tas.l2_nl_q10_rw2 tas.l3_nl_q10_rw2 tas.l4_nl_q10_rw2 tas.l5_nl_q10_rw2 tas.l6_nl_
##                 4                 1                 2                 2                 1
## tas.l7_nl_q10_rw2 tas.l8_nl_q10_rw2
##                 1                 1
```

```
table(m_multi_cov_best$covariate_2)
```

```
##
## prlr.l2_nl_q10_rw2 prlr.l3_nl_q10_rw2 prlr.l4_nl_q10_rw2 prlr.l6_nl_q10_rw2 prlr.l8_nl_q10_rw2
##                  1                  3                  1                  6                  2
```

```
#tas.l1_nl_q10_rw2
#tas.l4_nl_q10_rw2
#prlr.l6_nl_q10_rw2

m_multi_cov_best_list <- list( c("tas.l1", "prlr.l6"),
                               c("tas.l4", "prlr.l6"))
```

```
# Create a list of combined multivariate non-linear predictors repeated by province

#cov_varying_nl_rep_list <- ghrmodel::non_linear_covariates(covariates = m_multi_cov_best_list,
                                          # method = "quantile",
                                          # pattern = c("prlr.l"),
                                          # n = 10,
                                          # rep = "province")


# Create a list of combined multivariate non-linear predictors repeated by province

cov_varying_nl_var_list <- ghrmodel::varying_covariates(covariates = m_multi_cov_best_list,
                                                        pattern = c( "prlr.l"),
                                                        unit = "province")


#cov_varying_list <- c(cov_varying_nl_var_list,cov_varying_nl_rep_list)
```

Write varying model formulas

```
cov_varying_formulas <- write_inla_formulas(outcome = "dengue_cases",
                                        covariates = cov_varying_nl_var_list ,
                                        re1 = list(id ="month_id",
                                                   re ="rw1", cyclic = TRUE,
                                                   hyper = "prior_re1",
                                                   replicate = "province_id" ),
                                        re2 = list(id = "year_id",
                                                   re = "rw1",
                                                   hyper = "prior_re1"),
                                        re3 = list(id = "province_id",
                                                   re = "bym2",
                                                   graph = "g",
                                                   hyper = "prior_re2"),
                                                   baseline = TRUE)

#head(cov_varying_formulas)
```

```
# transform formulas list into a GHRformulas object

cov_varying_formulas_ghr <- ghrmodel::as_GHRformulas(formulas = cov_varying_formulas)

class(cov_varying_formulas_ghr)
```

```
## [1] "GHRformulas" "list"
```

```r
str(cov_varying_formulas_ghr)
```

```
## List of 4
##  $ formulas: chr [1:3] "dengue_cases ~ 1 + f(month_id, model = 'rw1', replicate = province_id, cycli
##  $ vars    :'data.frame':    3 obs. of  2 variables:
##   ..$ covariate_1: chr [1:3] NA "tas.l1" "tas.l4"
##   ..$ covariate_2: chr [1:3] NA "f(province, prlr.l6, model = 'iid')" "f(province, prlr.l6, model =
##  $ re      : Named chr [1:3] "f(month_id, model = 'rw1', replicate = province_id, cyclic = TRUE, cons
##   ..- attr(*, "names")= chr [1:3] "re_1" "re_2" "re_3"
##  $ outcome : chr "dengue_cases"
##  - attr(*, "class")= chr [1:2] "GHRformulas" "list"
```

```r
dplyr::glimpse(cov_varying_formulas_ghr$vars)
```

```
## Rows: 3
## Columns: 2
## $ covariate_1 <chr> NA, "tas.l1", "tas.l4"
## $ covariate_2 <chr> NA, "f(province, prlr.l6, model = 'iid')", "f(province, prlr.l6, model = 'iid')"
```

Fit multivariate models

```r
m_varying <- ghrmodel::fit_models(formulas = cov_varying_formulas_ghr ,
                            data = data,
                            family = "nbinomial",      # specify family
                            name = "m",
                            offset = "population",
                            config = TRUE,
                            pb = TRUE,
                            nthreads = 8)
```

```
##   |           |   |   0%  |           |=  |  33%
## Model 1 of 3 total run time 0.11 minutes.
##   |           |== |  67%
## Model 2 of 3 total run time 0.24 minutes.
##   |           |===| 100%
## Model 3 of 3 total run time 0.37 minutes.
```

```r
saveRDS(m_varying , "m_varying.rds")
```

```r
m_varying <- readRDS("m_varying.rds")
```

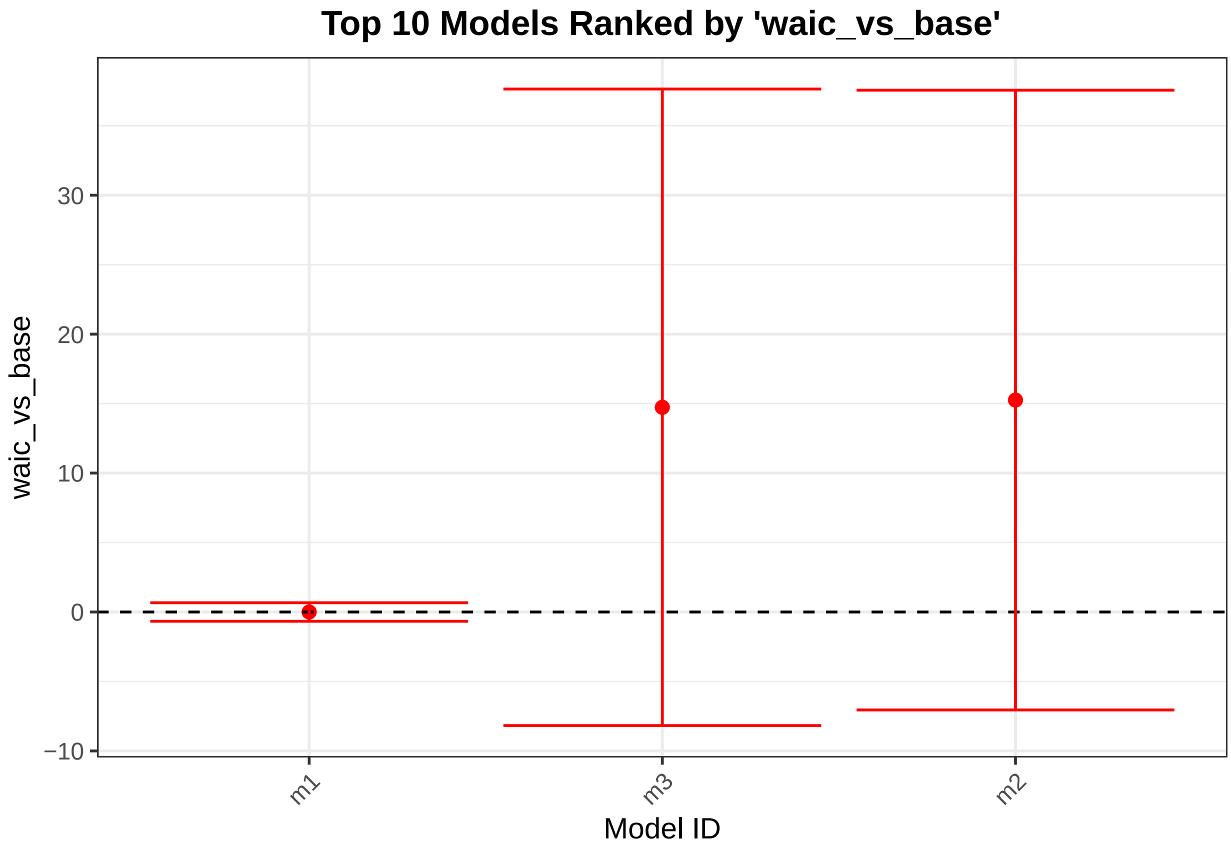
Rank multivariate models

```r
# goodness of fit metrics in a dataframe
m_varying_gof <- m_varying$mod.gof

m_varying_gof <- m_varying_gof %>%
  dplyr::arrange(waic)%>%
  dplyr::mutate(rank_waic = dense_rank(waic)) %>%
  dplyr::arrange(crps)%>%
  dplyr::mutate(rank_crps = dense_rank(crps))%>%
  dplyr::arrange(mae)%>%
  dplyr::mutate(rank_mae = dense_rank(mae))
```

WAIC

```
rank_varying_waic_vs_base <- ghrmodel::rank_models(
  models = m_varying,
  metric = "waic_vs_base",
  plot = TRUE,
  n = 10,
  intercept = TRUE,
  ci=TRUE
)
```

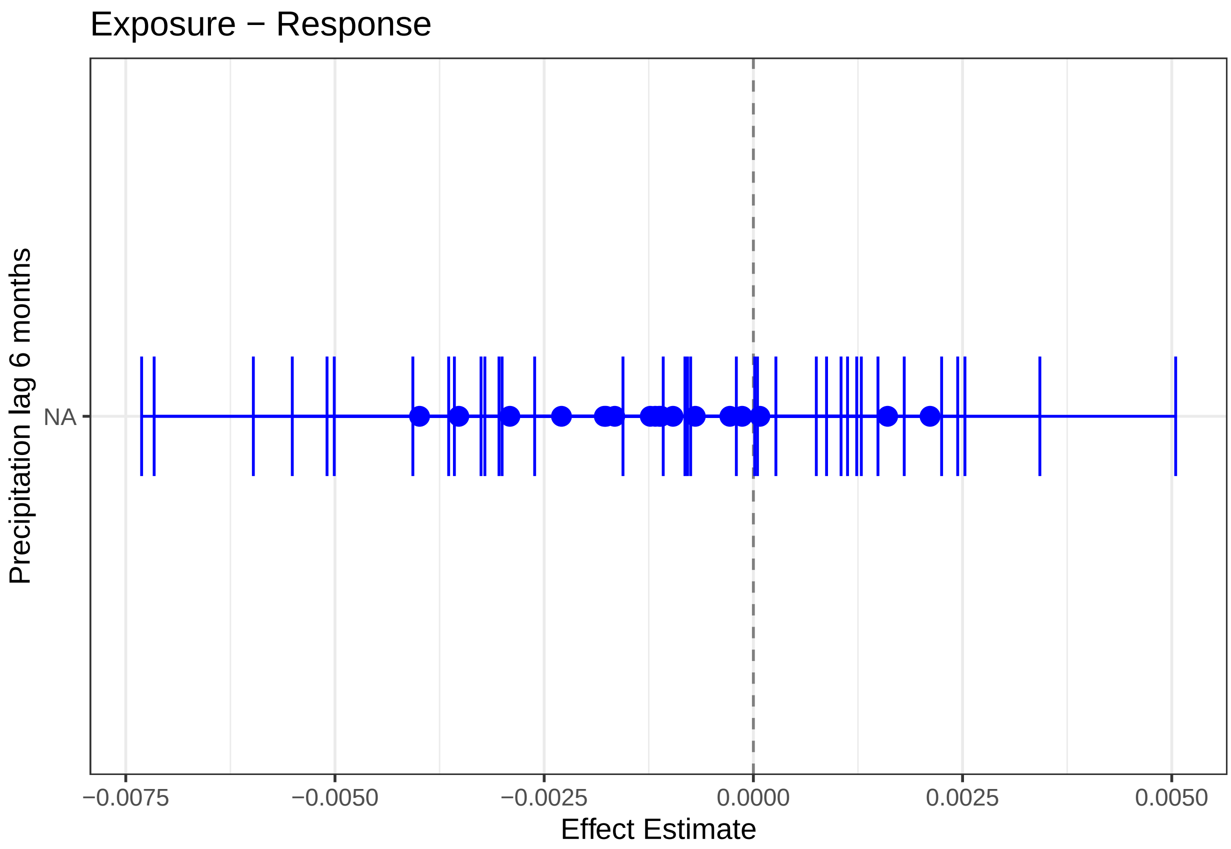
```
## Warning in rank_models.GHRmodels(models = m_varying, metric = "waic_vs_base", : 'n' (10) exceeds the
## models (3). Returning all available models.
```



These models fit worse, given that there is already a monthly effect that varies by province

```r
ghrmodel::plot_varying_coef(
  models = m_varying,
  model_id = "m2",
  var = "province",
  title = "Exposure - Response",
  var_label = "Precipitation lag 6 months",
  color = "blue"
)
```

```
## Warning in ghrmodel::plot_varying_coef(models = m_varying, model_id = "m2", : NAs introduced by coer
```


Exposure − Response
Precipitation lag 6 months
NA
Effect Estimate