HW2_Carnivore_teeth

Table of contents

Load libraries

```
library(tidyverse)
library(janitor)
library(naniar)
library(rstanarm)
library(broom.mixed)
library(ggpubr)
library(cowplot)
library(geodata)
library(countries)
library(terra)
library(tidyterra)
library(ggdist)
library(tidybayes)
library(bayesplot)
ggplot2::theme_set(theme_cowplot())
```

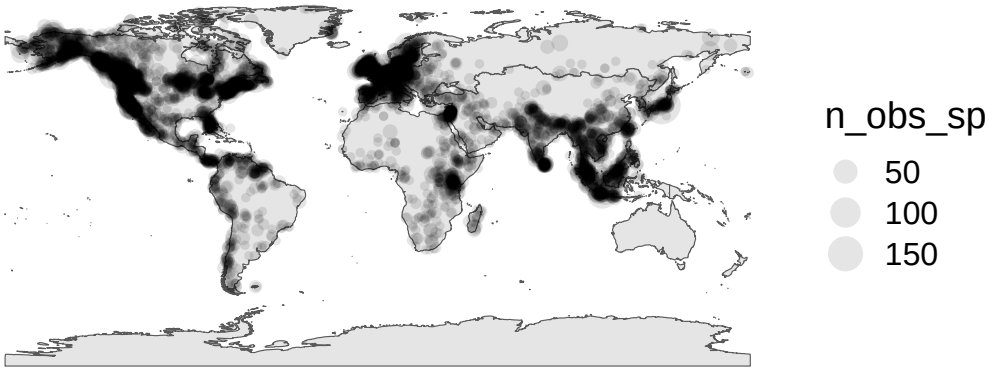
Step 1: Come up with a model

```
read_csv_cust <- function(path){
  read.csv(path) %>% as_tibble() %>%
  clean_names()
}
teeth <- read_csv_cust("carnivores/carnivoreteeth.csv")
mass <- read_csv_cust("carnivores/carnivorebodymass.csv")
```

Visualize data with world map

```
# Aggregate to continent level
countries <- geodata::world(resolution = 5, path = "maps")
cntry_codes <- country_codes()
countries <- merge(countries, cntry_codes, by.x = "GID_0", by.y = "ISO3", all.x = TRUE)
continent <- aggregate(countries, by = "continent")

# Create map
```

```
teeth_vect <- teeth %>% reframe(n_obs_sp = n(), .by = c(long, lat)) %>%
  vect(geom=c("long", "lat"), crs = "epsg:4326", keepgeom=TRUE)
continent %>% ggplot() +
  geom_spatvector() +
  geom_spatvector(data = teeth_vect, alpha = .1, aes(size = n_obs_sp))
```



Data exploration

```
explore_table <- function(df){
  map(df, \(col){
  table(col) %>% head(n = 6)
})
}
explore_uniq <- function(df){
  map_dfr(df, \(col){
  unique(col) %>% length()
})
}

# Teeth - PM4 = premolar, CsuppL = canine
explore_table(teeth)
```

```
$species
col
      Acinonyx jubatus Ailuropoda melanoleuca        Ailurus fulgens
                     5                       3                     4
        Alopex lagopus         Aonyx capensis          Aonyx cinerea
                   538                       4                    98
```

```
$msw2
col
      Acinonyx jubatus Ailuropoda melanoleuca        Ailurus fulgens
                     5                       3                      4
        Alopex lagopus      Amblonyx cinereus         Aonyx capensis
                   538                      98                      4

$msw3
col
      Acinonyx jubatus Ailuropoda melanoleuca        Ailurus fulgens
                     5                       3                      4
        Aonyx capensis          Aonyx cinerea    Arctictis binturong
                    10                      98                     33

$country
col
  Admiralty        Aero Afghanistan     Afognak       Aland      Alaska
        118           1           8           5           1        1164

$native
col
native
 18868

$family
col
    Canidae  Eupleridae     Felidae Herpestidae   Hyaenidae  Mephitidae
       3212          44        1660         743          40         336

$genus
col
    Acinonyx   Ailuropoda      Ailurus        Aonyx    Arctictis Arctogalidia
           5            3            4          107           33          123

$sex
col
Female   Male
  7560  11308

$pm4
col
 2.6 2.66 2.69 2.71 2.72 2.73
```

```
   1    1    1    1    1    1

$csup_l
col
0.85 0.87 0.95 0.98    1 1.01
   1    1    1    1    2    1

$lat
col
-51.97 -51.75 -51.63  -51.6  -51.5 -50.02
     1      5      1      9      7      1

$long
col
-172.72 -171.83 -171.77 -171.75  -171.5 -171.47
     17       4       1      50       3       1

$x
< table of extent 0 >

$x_1
< table of extent 0 >

$x_2
< table of extent 0 >

$x_3
< table of extent 0 >
```

```
explore_uniq(teeth)
```

```
# A tibble: 1 x 16
  species  msw2  msw3 country native family genus   sex   pm4 csup_l   lat  long
    <int> <int> <int>   <int>  <int>  <int> <int> <int> <int>  <int> <int> <int>
1     242   231   242     499      1     11   103     2  2299   1574  3103  4456
# i 4 more variables: x <int>, x_1 <int>, x_2 <int>, x_3 <int>
```

```
# Mass -- species x sex mass midpoints (at middle of species range?)
explore_table(mass)
```

```
$family
```

```
col
    Canidae     Felidae Herpestidae   Hyaenidae  Mustelidae Procyonidae
         78          81          78           8         138          30

$species
col
      Acinonyx jubatus Ailuropoda melanoleuca        Ailurus fulgens
                     2                       2                     2
        Alopex lagopus         Aonyx capensis          Aonyx cinerea
                     2                       2                     2

$sex
col
Female   Male
   251    252

$mass_midpoint
col
 55  56 111 118 130 134
  1   1   1   1   1   1

$log_midpoint_mass
col
1.74 1.75 2.04 2.07 2.11 2.13
   1    1    1    1    1    1

$x
< table of extent 0 >

$x_1
< table of extent 0 >
```

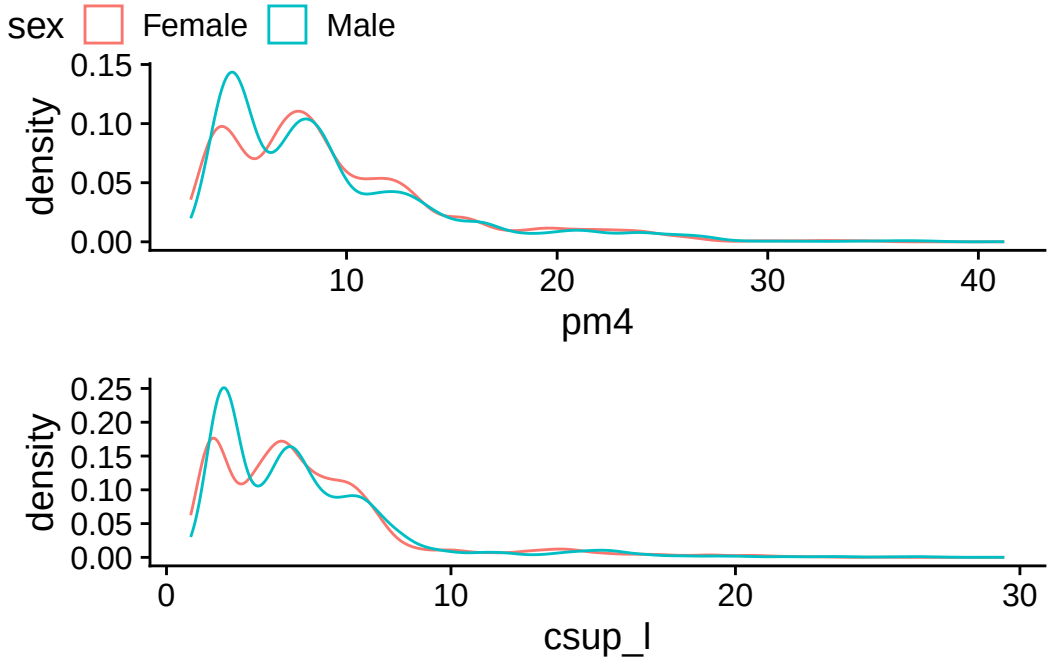
```
explore_uniq(mass)
```

```
# A tibble: 1 x 7
  family species   sex mass_midpoint log_midpoint_mass     x   x_1
   <int>   <int> <int>         <int>             <int> <int> <int>
1      8     252     2           361               211     1     1
```

```
# msw2 and msw3 appear to be different taxonomies , for now let's just stick with the 'specie
teeth2 <- teeth %>% select(-c(msw2, msw3, native, starts_with("x")))
mass2 <- mass %>% select(-starts_with("x"))
```

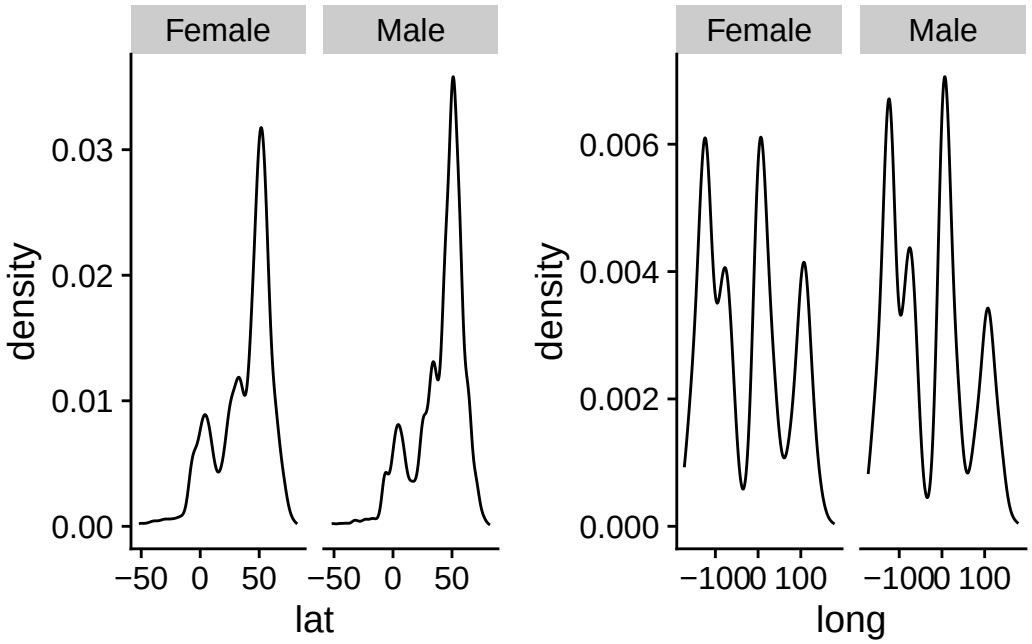
```
# Note teeth , latitude, etc. are per individual, whereas mass is per sex X species
df_join <- teeth2 %>% left_join(mass2[, c("species", "sex", "mass_midpoint")]) %>%
  rename(mass = mass_midpoint) %>%
  arrange(family, genus, species, sex)
```

```
plot_density <- function(df, col, ...){
  df %>% ggplot(aes(x = {{ col }}, ...)) +
    geom_density()
}

## Response variables
DV_plots <- imap(teeth2[,c("pm4", "csup_l")], \(col, name){
  teeth2 %>% plot_density(col = col, color = sex) +
    labs(x = name)
})

# Almost identical distributions, won't matter much which tooth we use in models
ggarrange(DV_plots[[1]], DV_plots[[2]], nrow = 2, common.legend = TRUE)
```




```
## Independent variables
# Geographic vars
IV_plots <- imap(teeth2[,c("lat", "long")], \(col, name){
```

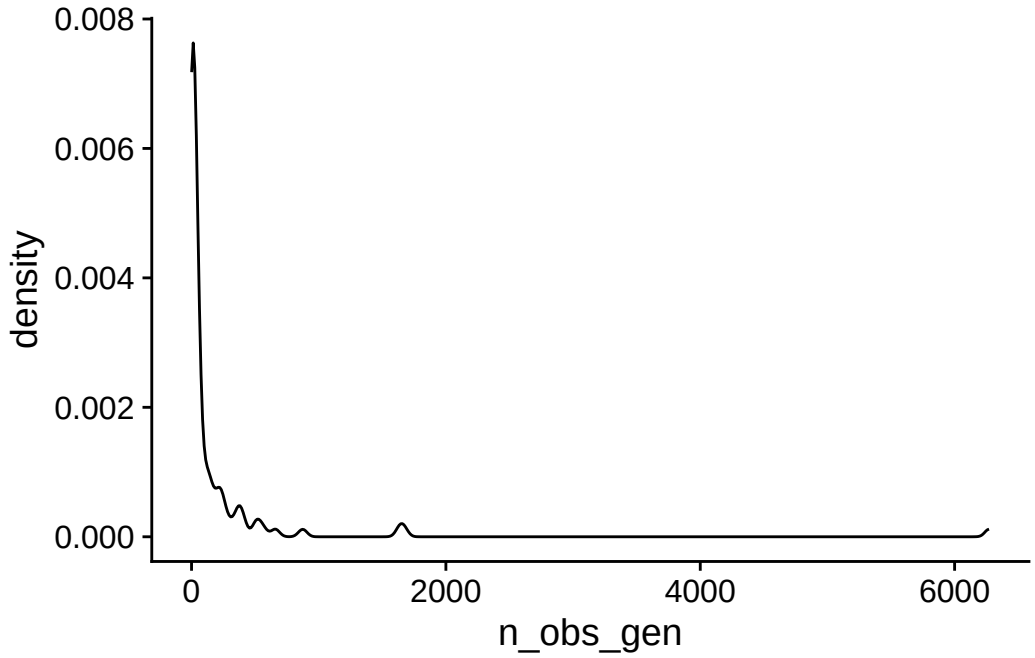
```
  teeth2 %>% plot_density(col = col) + #, color = species
    labs(x = name) +
    facet_wrap(~ sex) +
    guides(color = "none")
})
ggarrange(IV_plots[[1]], IV_plots[[2]])
```



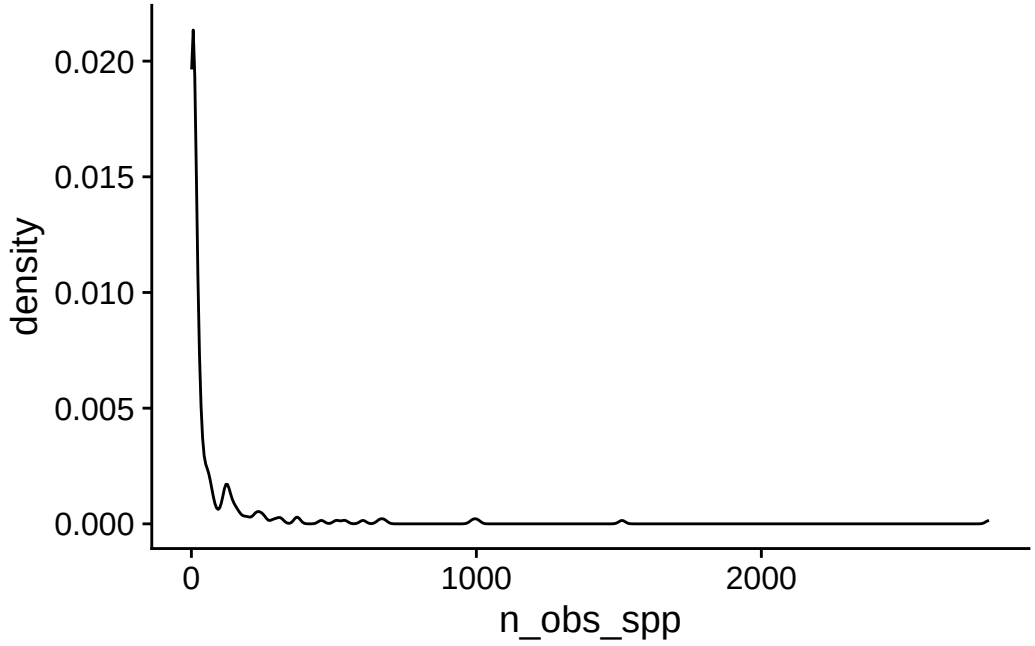
After looking at the data, I think an interesting question would be 'how does tooth size vary with geography?'. I believe tooth size could vary due to macrogeographic rules (e.g., Bergmann's rule). While body mass, or surface area to volume would be ideal for examining Bergmann's rule, allometric scaling theory states that as an organism's volume increases, most linear features (e.g. tooth length) also increase. I'd be interested to see if there is a relationship with longitude as well, given that precipitation gradients are often associated with longitudinal clines, and more precipitation -> higher productivity.

There are several observation per species or per genus in most cases, and these data points are not independent. There is likely some sort of phylogenetic model that would be more appropriate, but the least we could do is account for this non-independence by partially pooling how teeth change with latitude by species.
So let's investigate observations per genus & species, & subset our data appropriately

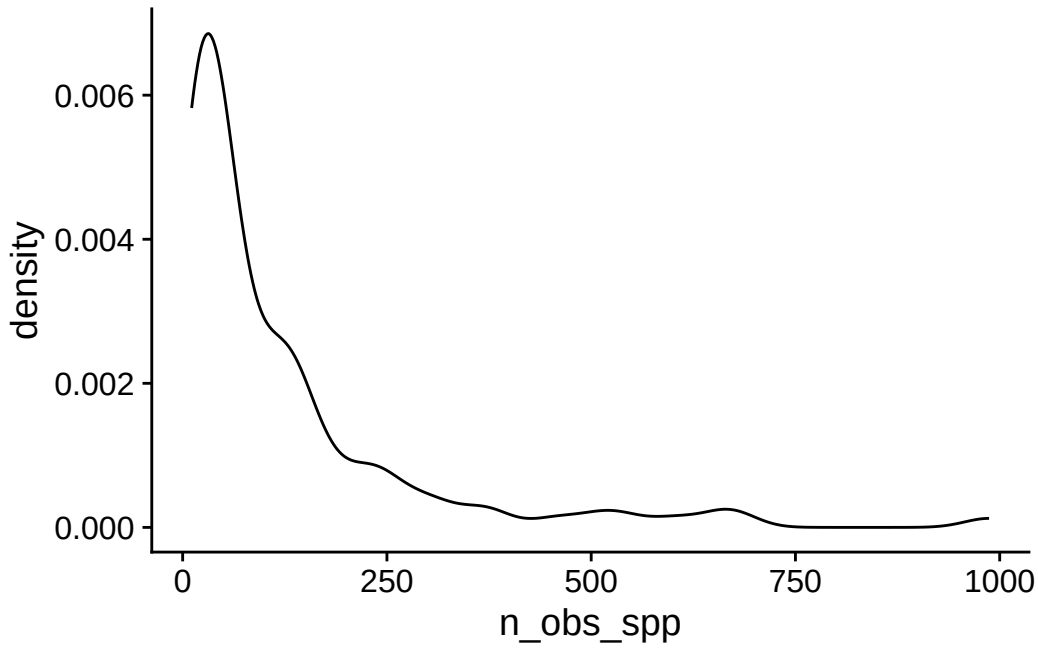
```
#103 genera
teeth2 %>% summarize(n_obs_gen = n(), .by = genus) %>%
  plot_density(col = n_obs_gen)
```




```
#242 species
teeth_spp <- teeth2 %>% reframe(across(), n_obs_spp = n(), .by = c(species))
# Most species have very few observations, although some have 2000+
teeth_spp %>% distinct(species, n_obs_spp) %>%
  plot_density(col = n_obs_spp)
```

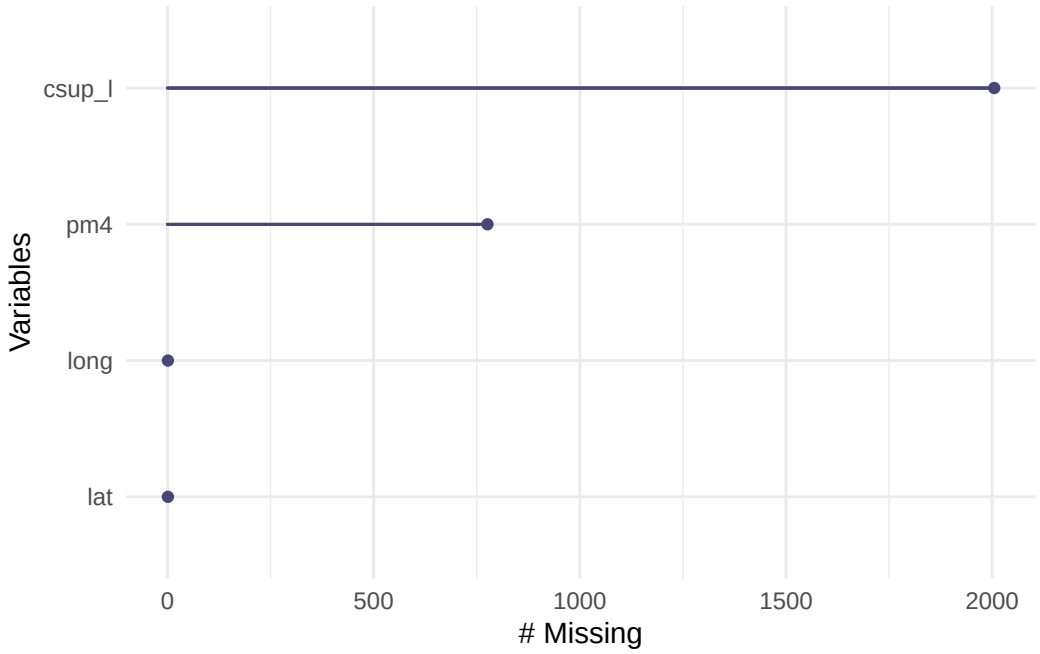


```
## 107 species between 10 & 1000 observation
# The species with huge numbers of observations will dominate the partial pooling of the rand
teeth_sub <- teeth_spp %>% filter(n_obs_spp > 10 & n_obs_spp < 1000)
teeth_sub %>% distinct(species, n_obs_spp) %>%
  plot_density(col = n_obs_spp)
```



Check NAs on key variables

```
teeth_sub %>%
  select(csup_l, pm4, lat, long) %>%
  gg_miss_var()
```

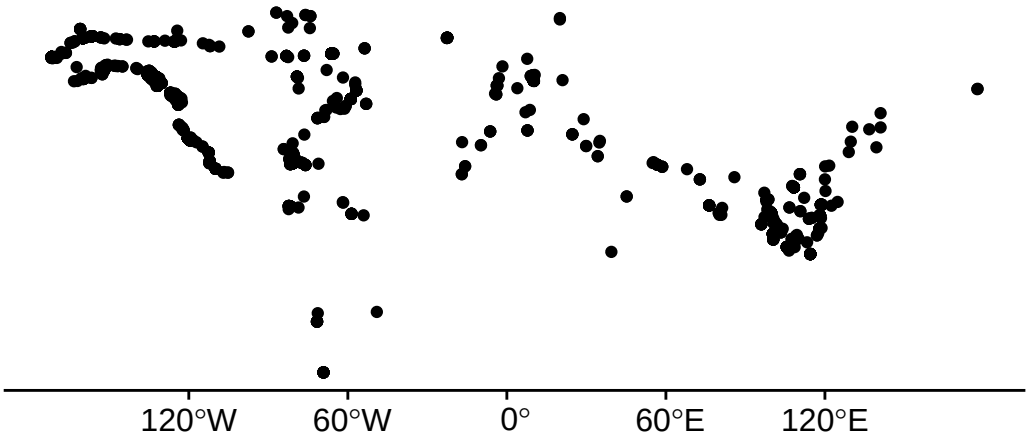


Continent

There are a few more steps necessary to make things run smoothly. First, we may expect biological relationships to vary by continent, so let's extract that information spatially. Ultimately, I envision random intercepts and slopes for latitude being drawn from a normal distribution at the level of continents, with species nested within each continent. This implies that species within a continent share a common distribution for their random effects (intercepts and slopes) but have species-specific deviations. In statistical terms, this is modeled as a nested random effect structure: (latitude | continent / species).
NOTE:: I did not simulate the inclusion of continent as I ran out of time (and it seems challenging!).

```
cont_join <- terra::extract(continent[,"continent"], teeth_vect) %>%
  select(-id.y)
teeth_sub2 <- teeth_vect %>% as_tibble() %>%
  cbind(cont_join) %>%
  right_join(teeth_sub) %>%
  as_tibble()

# For whatever reason there are still 1200 NAs for continent column. Fill in the blanks
cont_nas <- teeth_sub2 %>% filter(is.na(continent))

# Visualize the values that are NAs
```

```r
nas_vect <- cont_nas %>% vect(geom = c("long", "lat"), crs = "epsg:4326")
nas_vect %>% ggplot() +
  geom_spatvector()
```




```r
nearest_ids <- nas_vect %>%
  terra::nearest(continent, pairs = TRUE) %>%
  #filter(to_id != 0) %>% # remove one row that came up with 0
  pull(to_id)

cont_17 <- continent %>% mutate(cont_id = row_number()) %>%
  distinct(continent, cont_id) %>%
  as.data.frame()

coalesce_cont <- function(df){
  df %>% mutate(continent = coalesce(continent.x, continent.y)) %>%
  select(-continent.x, -continent.y)
}

fill_nas <- cont_nas %>% mutate(cont_id = nearest_ids) %>%
  left_join(cont_17, by = "cont_id") %>%
  coalesce_cont() %>%
  select(-cont_id)

teeth_sub3 <- teeth_sub2 %>%
  left_join(fill_nas, by = setdiff(names(teeth_sub2), "continent")) %>%
  coalesce_cont()

# Distribution of species per continent
```

```r
teeth_sub3 %>% distinct(species, continent) %>%
  tabyl(continent)
```

```
     continent  n     percent valid_percent
        Africa 23 0.127777778    0.12849162
          Asia 63 0.350000000    0.35195531
        Europe 23 0.127777778    0.12849162
 North America 35 0.194444444    0.19553073
       Oceania 19 0.105555556    0.10614525
 South America 16 0.088888889    0.08938547
          <NA>  1 0.005555556            NA
```

Formatting

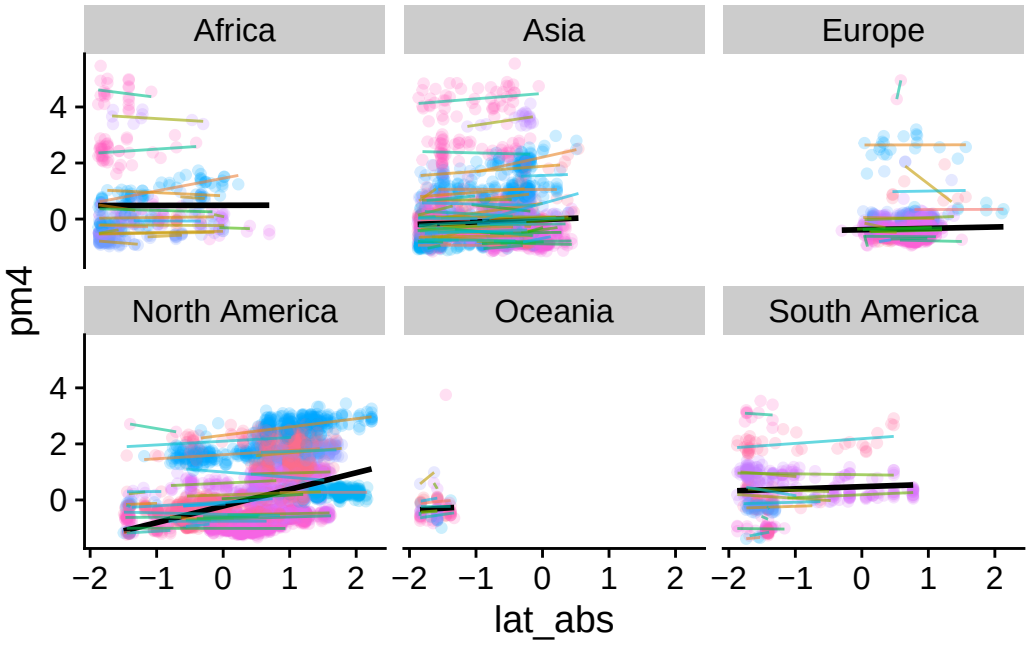
```r
# Formatting
teeth_fin <- teeth_sub3 %>%
  mutate(lat_abs = abs(lat),
         long = ifelse(long < 0, long + 360, long)) %>%
  group_by(species) %>%
  mutate(across(c(lat, long), ~ diff(range(.)), .names = "{.col}_spread")) %>% # Calculate sp
  ungroup() %>%
  mutate(across(where(is.double) & !contains("spread"), ~ as.numeric(scale(.x)))) # scale
# Remove species with very small latitudinal or longitudinal spreads
teeth_lat <- teeth_fin %>% filter(lat_spread > 1.5)
teeth_long <- teeth_fin %>% filter(lat_spread > 5)
```

Visualize (approximate) model

Let's plot the (approximate) model we are asking stan to fit. I would like to plot the quadratic of latitude for each species X continent , but there is a species that is nearly perfectly vertical (little latitudinal change w/ a large change in tooth size), despite that I removed individuals with very small latitudinal spreads.

```r
plot_model_ask <- function(df, IV, DV){
  df %>% ggplot(aes(x = {{ IV }}, y = {{ DV }})) +
    geom_point(alpha = .2, aes(color = species)) +
    geom_smooth(aes(group = continent),
                se = FALSE, color = "black", #, color = continent
                method = "lm") +
```

```
    stat_smooth(geom='line', alpha = 0.6, se = FALSE, method = "lm",
                aes(color = interaction(continent, species),
                    group = interaction(continent, species))) +
  guides(color = "none") +
  labs(x = deparse(substitute(IV)), y = deparse(substitute(DV)))
}

# Latitude plots
teeth_lat %>%
  plot_model_ask(IV = lat_abs, DV = pm4) +
  facet_wrap(~continent)
```



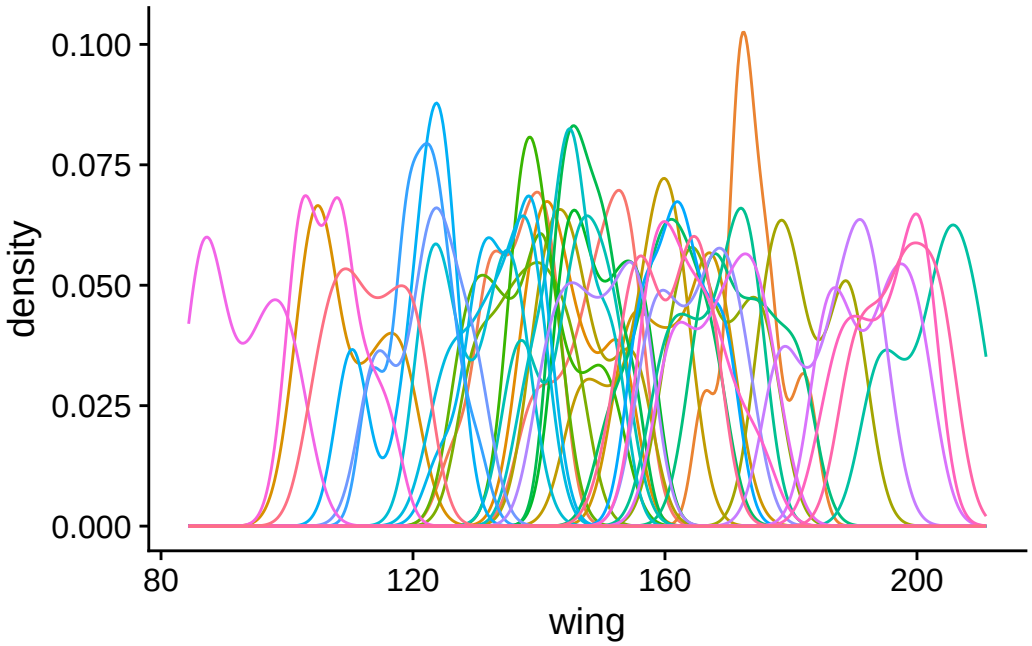

```
# Longitude plots
long_tooth_plots <- map2(c("csup_l", "pm4"), teeth_lat[, c("csup_l", "pm4")],
  \(DV_name, DV) {
    teeth_lat %>% plot_model_ask(IV = long, DV = DV) +
      labs(y = DV_name)
  }
)
#ggarrange(long_tooth_plots[[1]], long_tooth_plots[[2]])
```

I don't see many differences in how tooth size changes depending on which tooth is used. Let's use pm4 as it has fewer NAs

Step 2: Simulate test data

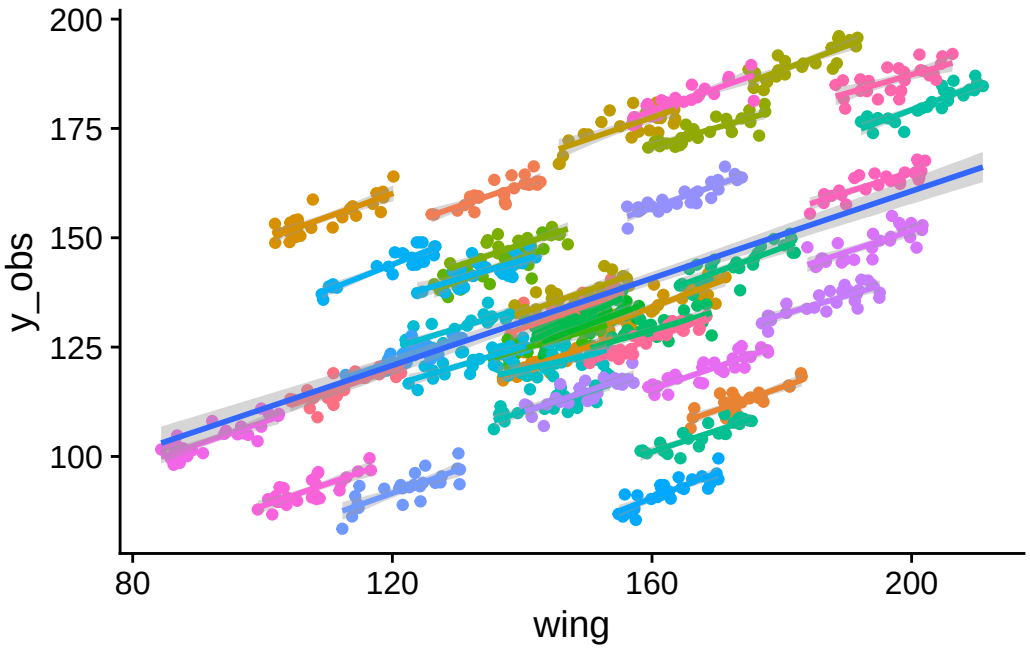
My brain is done for tonight, leaving this in as a place holder (this was the in-class exercise we did on Wednesday). This would be very similar to the simulation I would do, as this also simulates data hierarchically where the relationship between the IV & DV can vary by species (both random slope & intercept).

```
N <- 1000
N_spp <- 40 #40 species
spp <- 1:N_spp

## True parameter values
# Need to simulate 5 parameters (means and variances of intercept & slope, plus sigma for the
a_spp <- rnorm(N_spp, 60, 20) # intercept in grams
b_spp <- rnorm(N_spp, .5, .01) # slope in mm / gram
sigma_y <- 2

## Simulate species specific wing lengths
# 40 mean wing sizes
mu_iv_length <- rnorm(N_spp, 150, 30) # Wing size

# 20 min & max wing sizes (1 per species)
variance <- 30
min_max_iv <- map(mu_iv_length, \(iv_mu){
  min <- iv_mu - sqrt(3 * variance)
  max <- iv_mu + sqrt(3 * variance)
  tibble(iv_mu, min, max)
}) %>% list_rbind() %>%
  mutate(spp = 1:N_spp)

# Simulate a uniform distribution for wing length for each species
N_obs_spp <- N / N_spp # Number of observations per species
Wing_length_df <- min_max_iv %>% rowwise() %>%
  mutate(wing = list(runif(N_obs_spp, min = min, max = max))) %>%
  unnest(wing) %>%
  select(spp, wing)
Wing_length_l <- Wing_length_df %>% group_split(spp)

# Visualize 40 different wing length distributions
ggplot(Wing_length_df, aes(x = wing, color = as_factor(spp))) +
  geom_density() +
  guides(color = "none")
```



Generate ypred with your model

```
y_pred <- pmap(list(a_spp, b_spp, Wing_length_l), \(a, b, wl){
  y_pred <- a + b * wl[, "wing"]
  return(y_pred)
}) %>% list_rbind(names_to = "spp") %>%
  rename(y_hat = wing) %>%
  as_tibble()

# Simulate error and create "fake" dataframe
error <- rnorm(N, 0, sigma_y)
fake <- y_pred %>% mutate(y_obs = y_hat + error) %>%
  bind_cols(Wing_length_df[,"wing"]) %>%
  tibble()

# Visualize data and the (approximate) model you are asking stan to fit
ggplot(data = fake, aes(x = wing, y = y_obs)) +
  geom_point(aes(color = as_factor(spp))) +
  geom_smooth(aes(color = as_factor(spp)), method = "lm") +
  geom_smooth(method = "lm", extend = TRUE) +
  guides(color = "none")
```



Fit simulated model

First we will model our simulated data & wrangle the data

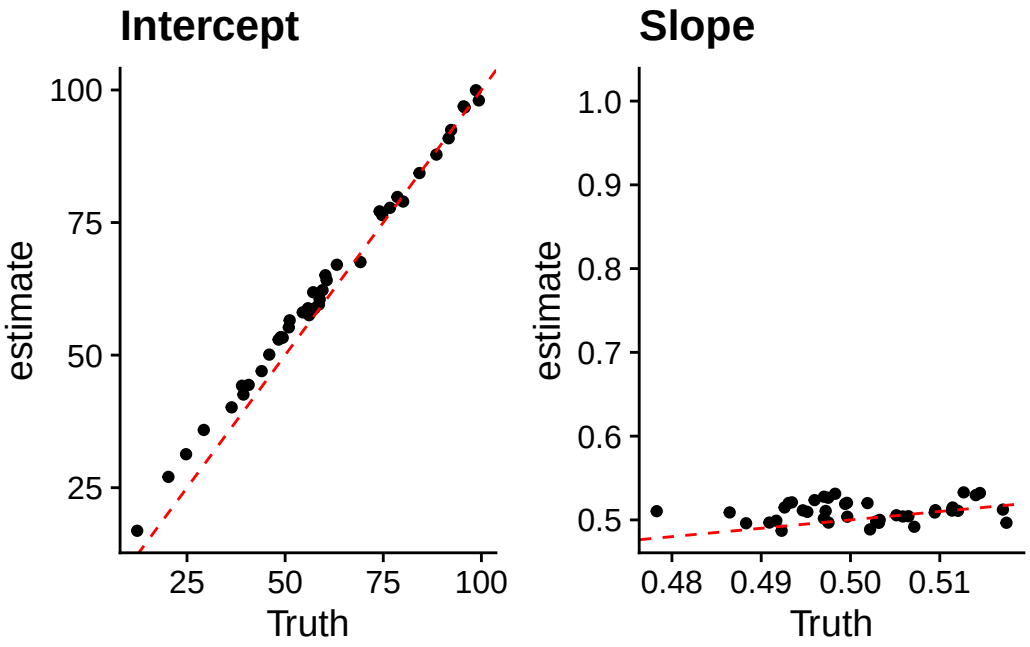
```
mod <- stan_lmer(y_obs ~ wing + (wing | spp), cores = 4, data = fake)
# NOTE:: There are species-specific estimates for intercept & slope for all 40 species

# Create tbl of true parameter values
truth <- tibble(
  random_effect = paste0("spp:", 1:N_spp),
  intercept = a_spp,
  wing = b_spp
) %>% pivot_longer(cols = c(intercept, wing), names_to = "term", values_to = "Truth")

# General function to extract the parameter terms
broom_tidy_mcmc <- function(mod){
  special_cases <- broom.mixed::tidyMCMC(mod) %>%
    filter(!str_detect(term, " ")) %>%
    pull(term)
  tidy_df <- broom.mixed::tidyMCMC(mod) %>%
    mutate(
      term2 = str_extract(term, "(?<=\\[)[^ ]+"),
      term2 = ifelse(str_detect(term2, "\\(.*\\)"), "intercept", term2),
```

```
      random_effect = str_extract(term, "(?<= )[\\w:]+")
    )
  tidy_df %>% mutate(term2 = ifelse(term2 == "(Intercept)", "global_intercept", term2),
                     term2 = ifelse(term %in% special_cases, term, term2),
                     term2 = ifelse(term2 == "(Intercept)", "global_intercept", term2),
                     term = term2) %>%
    select(-term2)
}

# Tbl of true & estimated parm values
parm_vals <- broom_tidy_mcmc(mod) %>%
  left_join(truth)
```

Test estimates vs truth

Ensure that we are able to return our modeled estimates

```
# Slopes
parms_b <- parm_vals %>%
  mutate(estimate = ifelse(term == "wing", .51 + estimate, estimate)) %>%
  filter(term == "wing") %>%
  ggplot(aes(x = Truth, y = estimate)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  guides(color = "none") +
  labs(title = "Slope")

# Intercept
parms_a <- parm_vals %>%
  mutate(estimate = ifelse(term == "intercept", 63.1 + estimate, estimate)) %>%
  filter(term == "intercept") %>%
  ggplot(aes(x = Truth, y = estimate)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  guides(color = "none") +
  labs(title = "Intercept")
ggarrange(parms_a, parms_b)
```



We achieve a good mix around the 1:1 line, some above and some below, signifying that your model did a good job estimating parameters from the simulated data

Step 4: Fit model to empirical data

Fit model

```
fit_lat <- stan_lmer(pm4 ~ poly(lat_abs, 2) + (lat_abs | continent / species),
                     data = teeth_lat, cores = 4, adapt_delta = .99)
#summary(fit_lat)
```

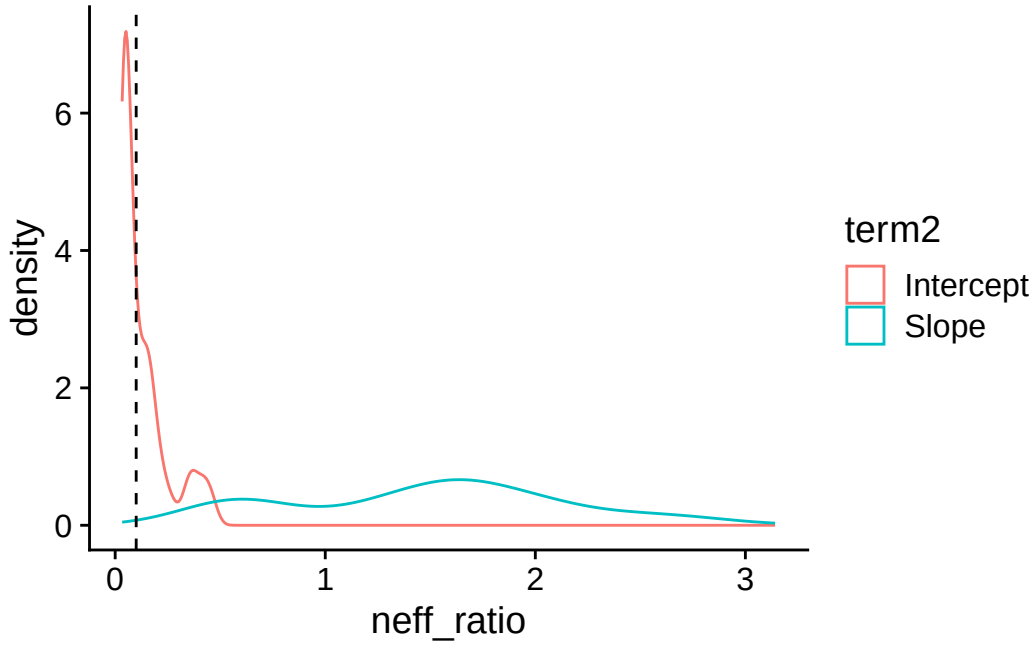
IMPORTANT:: There was 1 divergent transitions after warmup! Yikes.

Will have to leave the longitude model for another day

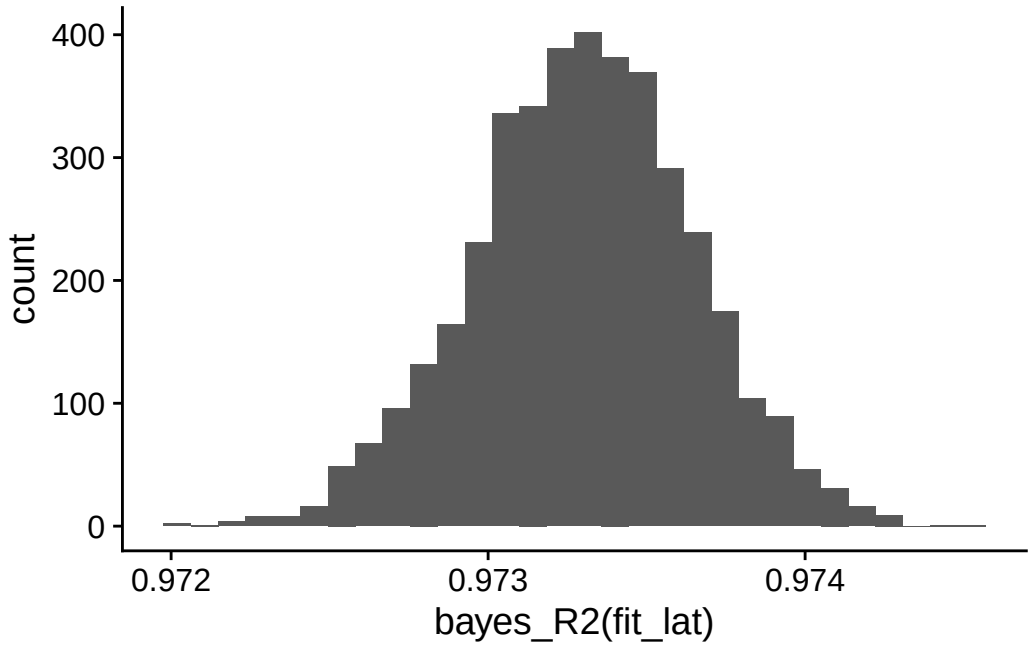
```
fit_long <- stan_lmer(pm4 ~ poly(long, 2) + (long | continent / species),
                      data = teeth_long, cores = 4,  adapt_delta = .99)
```

Model diagnostics

```r
diagnostics <- data.frame(rhat = rhat(fit_lat), neff_ratio = neff_ratio(fit_lat)) %>%
  rownames_to_column("term") %>% tibble()

# Low effective sample sizes when estimating intercepts, but good when estimating slope
diagnostics %>% mutate(term2 = case_when(
  str_detect(term, "Intercept") ~ "Intercept",
  str_detect(term, "lat_abs") ~ "Slope"
  )) %>%
  filter(!is.na(term2)) %>%
  ggplot() +
  geom_density(aes(x = neff_ratio, color = term2)) +
  geom_vline(xintercept = .1, linetype = "dashed")
```

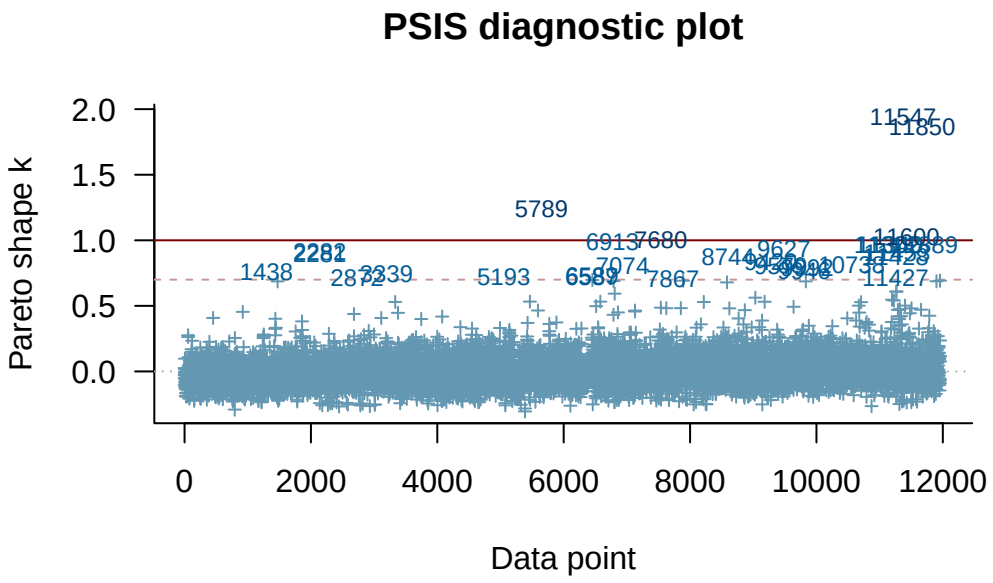



```r
## Examine the autocorrelation in specific species with low neff
prob_spp <- diagnostics %>% filter(neff_ratio < 0.03) %>%
  mutate(species = str_split_i(term, ":", 3)) %>%
  pull(species)
# Just a few examples for better visualization
# Visualize the autocorrelation in a few species
# plot(fit_lat, "acf", pars = "(Intercept)", regex_pars = paste0(prob_spp[1:2],":North_Americ

# Very high bayes R2
ggplot() + geom_histogram(aes(x=bayes_R2(fit_lat)))
```



There are several things that give me pause examining this model... There are several intercept values that have low effective sample size ratios. And there is a very high bayes R2 values could indicate overfitting.. Let's examine Loo values

```
loo_lat <- loo(fit_lat)
remove_ids <- loo::pareto_k_ids(loo_lat)
plot(loo_lat, diagnostic = "k", label_points = TRUE) # diagnostic = "ESS", "n_eff
```

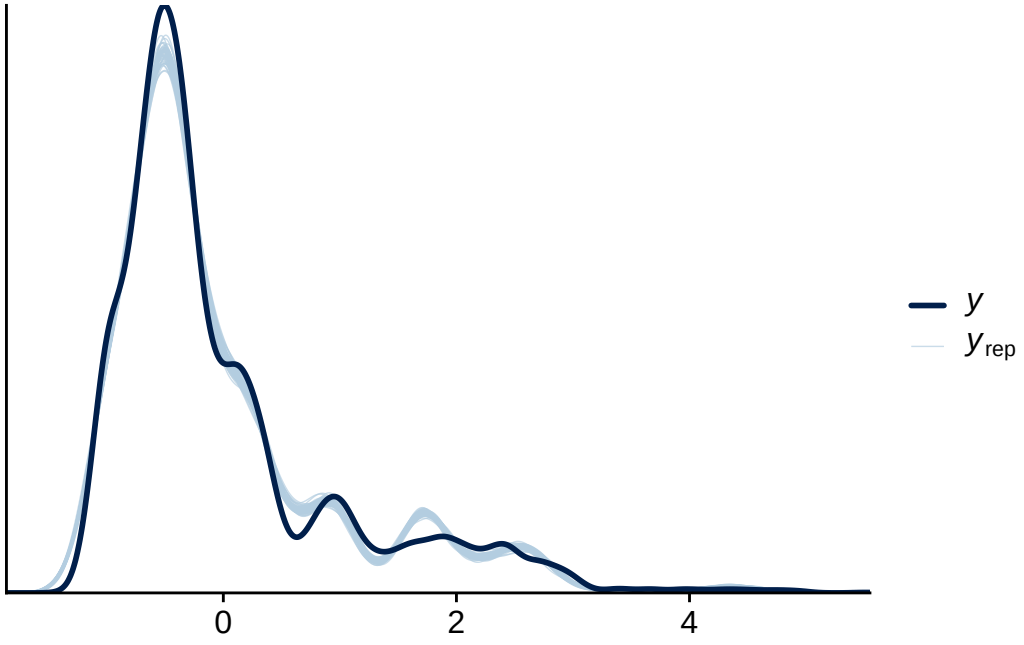


It seems there are some 30 points with high pareto k values. We could consider removing these values and refitting the model to see if that helps. We could also consider kfold cross-validation, where the model is refit K times, each time leaving out one of the K subsets.

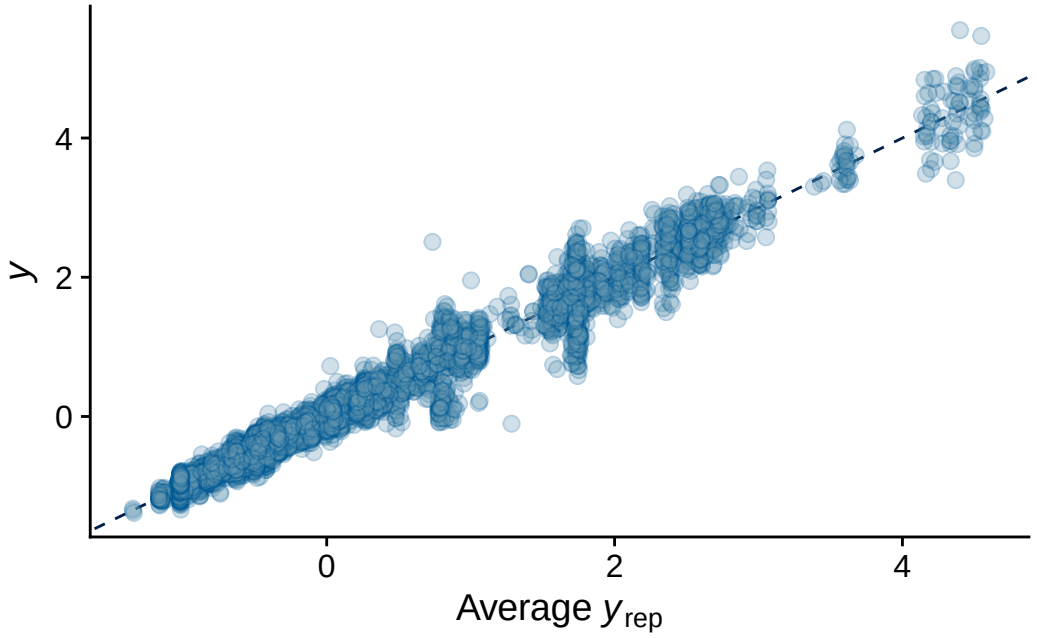
Posterior predictive checks

Let's examine if our data a plausible sample from the posterior distribution?

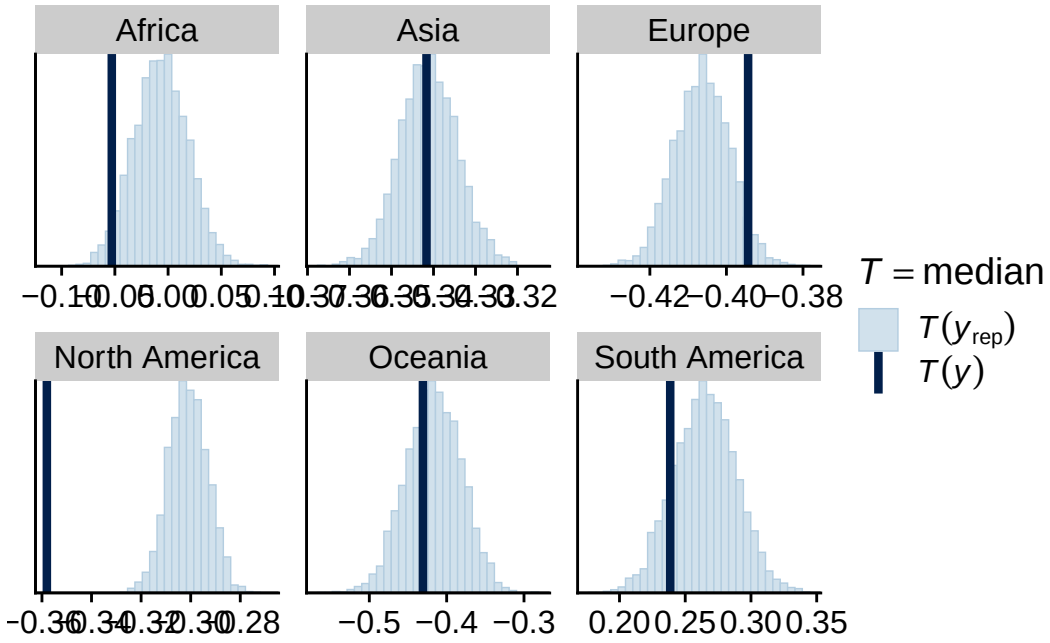
```
pp_check(fit_lat)
```



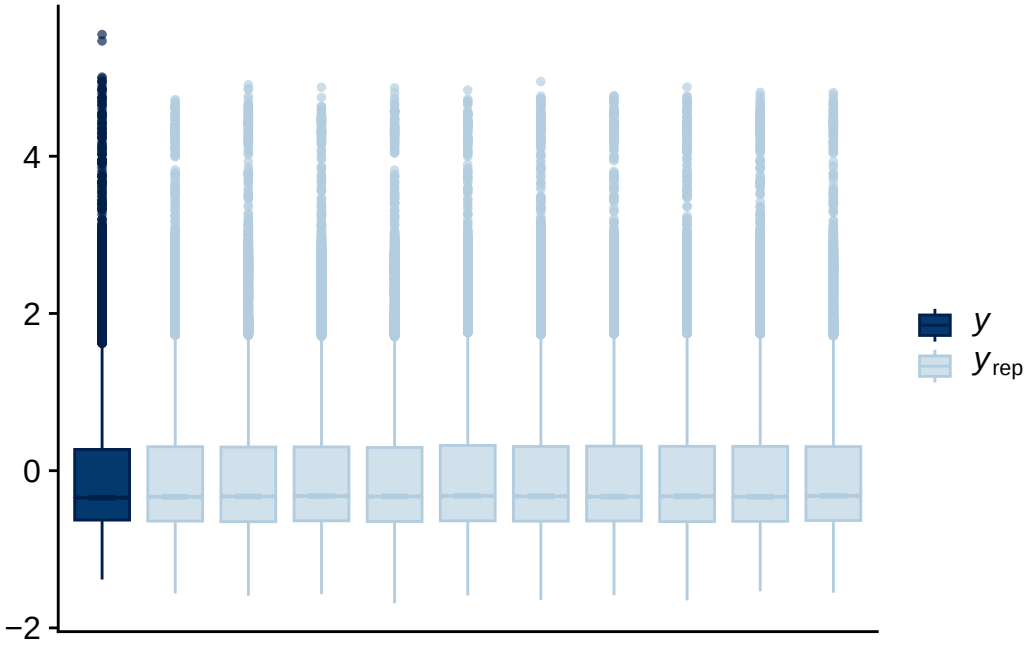
```
pp_check(fit_lat, plotfun = "scatter_avg", alpha = .3)
```




```
pp_check(fit_lat, plotfun = "stat_grouped", stat = "median", group = "continent")
```



```
pp_check(fit_lat, plotfun = "boxplot", nreps = 10)
```



The posterior predictive checks are pretty good, except it is clear that species in N America are not fitting the data well at all. I wonder if the NAs from above (along the coast of North America) are causing problems? Perhaps a simpler model would be better, especially given that there was a divergent transition in this model.