Modeling biodiversity metrics

Alex Baecher & Ed Basham

12/18/2020

```
{source("scripts/01_curation/00_package_loading.R"); load_packages(
  c(
    "lme4",
    "AICcmodavg",
    "glmmTMB",
    "TMB",
    "tidyverse",
    "readr",
    "scales",
    "formula.tools",
    "sjPlot",
    "ggpubr",
    "scales",
    "visreg",
    "bbmle"

    )
  )

}
```

Load data

```
dat <- read_csv("data/stripped_data/final/data_joined.csv") %>%
  filter(taxa != "All mammals") %>%
  filter(taxa != "Primates") %>%
  dplyr::mutate(weight = spatial_rank + temporal_bredth_rank + temporal_resolution_rank,
                taxa = as.factor(taxa),
                taxa_order = factor(taxa, levels = c("Birds","Bats","Small mammals","Amphibians")),
                elevation = as.numeric(elevation),
                scaled_met = as.numeric(rescale(corrected_biodiversity_metric_value, to = c(0.00001, 0.9
                link = as.factor(link),
                method = as.factor(method),
                treatment = as.factor(treatment),
                continent = as.factor(continent),
                biodiversity_metric = as.factor(biodiversity_metric),
                season = as.factor(season),
                forest_type = as.factor(forest_type),
                mean_strata_height_p = as.numeric(mean_strata_height_p)) %>%
  dplyr::select(link, study_id.x, method, taxa, continent, biodiversity_metric, taxa_order,
                treatment, season, forest_type, elevation, canopy_height, latitude, longitude,
                scaled_met, strata = mean_strata_height_p)
```

```
##
## -- Column specification -----------------------------------------------------------------------------
## cols(
##   .default = col_character(),
##   sites = col_double(),
##   data_quality = col_double(),
##   sum_total_richness = col_double(),
##   sum_total_abundance = col_double(),
##   spatial_rank = col_double(),
##   temporal_resolution_rank = col_double(),
##   temporal_bredth_rank = col_double(),
##   elevation = col_double(),
##   canopy_height = col_double(),
##   latitude = col_double(),
##   longitude = col_double(),
##   linking_id = col_double(),
##   biodiversity_metric_value = col_double(),
##   var_biodiversity_metric = col_double(),
##   var_biodiversity_metric_min = col_double(),
##   var_biodiversity_metric_max = col_double(),
##   subsequent_correction = col_double(),
##   proportion_of_max_biodiversity = col_double(),
##   statistic_standard_deviation = col_logical(),
##   min_strata_height = col_double()
##   # ... with 18 more columns
## )
## i Use `spec()` for the full column specifications.
```

```r
abund <- dat %>%
  subset(biodiversity_metric == "abundance")
rich <- dat %>%
  subset(biodiversity_metric == "richness")
```

Diagnostics

```r
# ggplot(abund, aes(x = strata, y = scaled_met, color = method))+
#   geom_point() +
#   geom_smooth(method = "loess")+
#   facet_wrap(~taxa)
#
# ggplot(rich, aes(x = strata, y = scaled_met, color = method))+
#   geom_point() +
#   geom_smooth(method = "loess")+
#   facet_wrap(~taxa)
#
# ggplot(abund, aes(x = strata, y = scaled_met, color = link))+
#   geom_point() +
#   #geom_line()+
#   geom_smooth(method = "lm", alpha = 0.01) +
#   facet_wrap(~taxa) +
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   theme(legend.position = "none")+
#   labs(title = "Abundance", x = "Strata Height (proportion of study specific max forest height)", y =
#
# ggplot(rich, aes(x = strata, y = scaled_met, color = link))+
#   geom_point()+
#   #geom_line()+
#   geom_smooth(method = "lm", alpha = 0.01)+
#   facet_wrap(~taxa)+
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   theme(legend.position = "none")+
#   labs(title = "Richness", x = "Strata Height (proportion of study specific max forest height)", y =
#
# ggplot(abund, aes(x = strata, y = scaled_met, color = continent))+
#   geom_point()+
#   geom_smooth(alpha = 0.01)+
#   facet_wrap(~taxa)+
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   labs(title = "Abundance", x = "Strata Height (proportion of study specific max forest height)", y =
#
#
# ggplot(rich, aes(x = strata, y = scaled_met, color = continent))+
#   geom_point()+
#   geom_smooth(method = "lm", alpha = 0.01)+
#   facet_wrap(~taxa)+
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   labs(title = "Richness", x = "Strata Height (proportion of study specific max forest height)", y =
#
#
# ggplot(rich, aes(x = strata, y = scaled_met))+
#   geom_point()+
#   geom_smooth(method = "lm")+
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   facet_wrap(~taxa)
#
# ggplot(abund, aes(x = strata, y = scaled_met))+
#   geom_point()+
#   geom_smooth(method = "lm")+
#   coord_cartesian(xlim=c(0,1), ylim= c(0,1)) +
#   facet_wrap(~taxa)
```

Richness GLMs

```
mods_rich <- list()
# strata only models
mods_rich[[1]]  <- glmmTMB(scaled_met ~ 1 +                                                (strata|link
mods_rich[[2]]  <- glmmTMB(scaled_met ~ strata +                                           (strata|link
mods_rich[[3]]  <- glmmTMB(scaled_met ~ poly(strata, 2) +                                  (strata|link
mods_rich[[4]]  <- glmmTMB(scaled_met ~ poly(strata, 3) +                                  (strata|link
# strata and taxa
mods_rich[[5]]  <- glmmTMB(scaled_met ~ strata + taxa +                                    (strata|link
mods_rich[[6]]  <- glmmTMB(scaled_met ~ poly(strata, 2) + taxa +                           (strata|link
mods_rich[[7]]  <- glmmTMB(scaled_met ~ poly(strata, 3) + + taxa +                         (strata|link
# strata and continent
mods_rich[[8]]  <- glmmTMB(scaled_met ~ strata + continent +                               (strata|link
mods_rich[[9]]  <- glmmTMB(scaled_met ~ poly(strata, 2) + continent +                      (strata|link
mods_rich[[10]] <- glmmTMB(scaled_met ~ poly(strata, 3) + continent +                      (strata|link
# strata and elevation
mods_rich[[11]] <- glmmTMB(scaled_met ~ strata + elevation +                               (strata|link
mods_rich[[12]] <- glmmTMB(scaled_met ~ poly(strata, 2) + elevation +                      (strata|link
mods_rich[[13]] <- glmmTMB(scaled_met ~ poly(strata, 3) + elevation +                      (strata|link
# strata method
mods_rich[[14]] <- glmmTMB(scaled_met ~ strata + method +                                  (strata|link
mods_rich[[15]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method +                         (strata|link
mods_rich[[16]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method +                         (strata|link
# strata, elevation, and continent
mods_rich[[17]] <- glmmTMB(scaled_met ~ strata + elevation + continent +                   (strata|link
mods_rich[[18]] <- glmmTMB(scaled_met ~ poly(strata, 2) + elevation + continent +          (strata|link
mods_rich[[19]] <- glmmTMB(scaled_met ~ poly(strata, 3) + elevation + continent +          (strata|link
# strata, method, and continent
mods_rich[[20]] <- glmmTMB(scaled_met ~ strata + method + continent +                      (strata|link
mods_rich[[21]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method + continent +             (strata|link
mods_rich[[22]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method + continent +             (strata|link
# strata, method, and elevation
mods_rich[[23]] <- glmmTMB(scaled_met ~ strata + method + continent + elevation +          (strata|link
mods_rich[[24]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata|link
mods_rich[[25]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata|link
# strata interactions with methods or continent or elevation
mods_rich[[26]] <- glmmTMB(scaled_met ~ strata*method +                                    (strata|link
mods_rich[[27]] <- glmmTMB(scaled_met ~ strata*continent +                                 (strata|link
mods_rich[[28]] <- glmmTMB(scaled_met ~ strata*elevation +                                 (strata|link
# strata interactions with methods or continent or elevation plus additive relationships continent and
mods_rich[[29]] <- glmmTMB(scaled_met ~ strata*method + continent +                        (strata|link
mods_rich[[30]] <- glmmTMB(scaled_met ~ strata*method + elevation +                        (strata|link
mods_rich[[31]] <- glmmTMB(scaled_met ~ strata*continent + method +                        (strata|link
mods_rich[[32]] <- glmmTMB(scaled_met ~ strata*continent + elevation +                     (strata|link
mods_rich[[33]] <- glmmTMB(scaled_met ~ strata*elevation + method +                        (strata|link
mods_rich[[34]] <- glmmTMB(scaled_met ~ strata*elevation + continent +                     (strata|link
```

```
rich_mod_names <- list()
for(i in 1:length(mods_rich)){
  rich_mod_names[[i]] <- as.character(mods_rich[[i]]$call$formula)
}

rich_modnames <- unlist(rich_mod_names)

rich_aictab <- aictab(cand.set = mods_rich, modnames = unlist(rich_mod_names))

bbmle::AICtab(mods_rich, mnames = unlist(rich_mod_names))
```

```
##                                                                    dAIC
## scaled_met ~ strata + taxa + (strata | link)                         0.0
## scaled_met ~ poly(strata, 3) + +taxa + (strata | link)               0.3
## scaled_met ~ poly(strata, 2) + taxa + (strata | link)                0.6
## scaled_met ~ strata * method + (strata | link)                       0.9
## scaled_met ~ strata * method + elevation + (strata | link)           2.6
## scaled_met ~ strata * method + continent + (strata | link)           4.6
## scaled_met ~ strata + method + (strata | link)                       6.0
## scaled_met ~ poly(strata, 3) + method + (strata | link)              6.2
## scaled_met ~ poly(strata, 2) + method + (strata | link)                          6.8
## scaled_met ~ strata * elevation + method + (strata | link)                       7.1
## scaled_met ~ strata * continent + method + (strata | link)                       8.5
## scaled_met ~ strata + method + continent + (strata | link)                       9.2
## scaled_met ~ poly(strata, 3) + method + continent + (strata | link)              9.5
## scaled_met ~ poly(strata, 2) + method + continent + (strata | link)             10.1
## scaled_met ~ strata + method + continent + elevation + (strata | link)          11.1
## scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata | link) 11.4
## scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata | link) 12.0
## scaled_met ~ 1 + (strata | link)                                                27.1
## scaled_met ~ strata + (strata | link)                                           27.7
## scaled_met ~ strata * elevation + (strata | link)                               28.3
## scaled_met ~ strata + elevation + (strata | link)                               28.5
## scaled_met ~ poly(strata, 2) + (strata | link)                                  28.5
## scaled_met ~ poly(strata, 3) + (strata | link)                                  28.7
## scaled_met ~ poly(strata, 2) + elevation + (strata | link)                      29.3
## scaled_met ~ poly(strata, 3) + elevation + (strata | link)                      29.5
## scaled_met ~ strata + continent + (strata | link)                               31.6
## scaled_met ~ strata * continent + (strata | link)                               32.0
## scaled_met ~ strata * elevation + continent + (strata | link)                   32.1
## scaled_met ~ strata + elevation + continent + (strata | link)                   32.2
## scaled_met ~ poly(strata, 2) + continent + (strata | link)                      32.5
## scaled_met ~ strata * continent + elevation + (strata | link)                   32.6
## scaled_met ~ poly(strata, 3) + continent + (strata | link)                      32.6
## scaled_met ~ poly(strata, 2) + elevation + continent + (strata | link)          33.1
## scaled_met ~ poly(strata, 3) + elevation + continent + (strata | link)          33.2
##                                                                                 df
## scaled_met ~ strata + taxa + (strata | link)                                    9
## scaled_met ~ poly(strata, 3) + +taxa + (strata | link)                          11
## scaled_met ~ poly(strata, 2) + taxa + (strata | link)                           10
## scaled_met ~ strata * method + (strata | link)                                  20
## scaled_met ~ strata * method + elevation + (strata | link)                      21
## scaled_met ~ strata * method + continent + (strata | link)                      22
## scaled_met ~ strata + method + (strata | link)                                  13
## scaled_met ~ poly(strata, 3) + method + (strata | link)                         15
## scaled_met ~ poly(strata, 2) + method + (strata | link)                         14
## scaled_met ~ strata * elevation + method + (strata | link)                      15
## scaled_met ~ strata * continent + method + (strata | link)                      17
## scaled_met ~ strata + method + continent + (strata | link)                      15
## scaled_met ~ poly(strata, 3) + method + continent + (strata | link)             17
## scaled_met ~ poly(strata, 2) + method + continent + (strata | link)             16
## scaled_met ~ strata + method + continent + elevation + (strata | link)          16
## scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata | link) 18
## scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata | link) 17
## scaled_met ~ 1 + (strata | link)                                                5
## scaled_met ~ strata + (strata | link)                                           6
## scaled_met ~ strata * elevation + (strata | link)                               8
## scaled_met ~ strata + elevation + (strata | link)                               7
## scaled_met ~ poly(strata, 2) + (strata | link)                                  7
## scaled_met ~ poly(strata, 3) + (strata | link)                                  8
## scaled_met ~ poly(strata, 2) + elevation + (strata | link)                      8
## scaled_met ~ poly(strata, 3) + elevation + (strata | link)                      9
## scaled_met ~ strata + continent + (strata | link)                               8
## scaled_met ~ strata * continent + (strata | link)                               10
## scaled_met ~ strata * elevation + continent + (strata | link)                  10
## scaled_met ~ strata + elevation + continent + (strata | link)                  9
## scaled_met ~ poly(strata, 2) + continent + (strata | link)                     9
## scaled_met ~ strata * continent + elevation + (strata | link)                  11
## scaled_met ~ poly(strata, 3) + continent + (strata | link)                     10
## scaled_met ~ poly(strata, 2) + elevation + continent + (strata | link)         10
## scaled_met ~ poly(strata, 3) + elevation + continent + (strata | link)         11
```

Model predictions

Getting model esimates from top model

```
top_rich_mod <- mods_rich[[as.numeric(str_extract(aictab(mods_rich)[1,1], "(\\d)+"))]]
```

```
## Warning in aictab.AICglmmTMB(mods_rich):
## Model names have been supplied automatically in the table
```

```
model_estimates_rich <- sjPlot::get_model_data(top_rich_mod, type = "re", se = T, transform = NULL)
```

Getting model estimates from each model

```
rich_beta_intercept <- matrix(NA, nrow = 50, ncol = length(mods_rich))
rich_beta_intercept_se <- rich_beta_intercept
rich_beta_strata <- rich_beta_intercept
rich_beta_strata_se <- rich_beta_intercept

for(i in 1:length(mods_rich)){
  rich_beta_intercept[,i] <- sjPlot::get_model_data(mods_rich[[i]], type = "re", se = T, transform = NUL
    select(estimate) %>%
    as.matrix()

  rich_beta_intercept_se[,i] <- sjPlot::get_model_data(mods_rich[[i]], type = "re", se = T, transform =
    select(estimate, conf.low) %>%
    mutate(se = (estimate - conf.low)/1.96) %>%
    select(se) %>%
    as.matrix()

  rich_beta_strata[,i] <- sjPlot::get_model_data(mods_rich[[i]], type = "re", se = T, transform = NULL)
    select(estimate) %>%
    as.matrix()

  rich_beta_strata_se[,i] <- sjPlot::get_model_data(mods_rich[[i]], type = "re", se = T, transform = NUL
    select(estimate, conf.low) %>%
    mutate(se = (estimate - conf.low)/1.96) %>%
    select(se) %>%
    as.matrix()
}
```

Model averaging model estimates

```r
rich_modavg_beta_intercept_list <- list()
rich_modavg_beta_strata_list <- list()

rich_modavg_beta_intercept_matrix <- data.frame(
  Mod.avg.est = rep(NA, nrow(rich_beta_intercept)),
  Uncond.SE = rep(NA, nrow(rich_beta_intercept)),
  Lower.CL = rep(NA, nrow(rich_beta_intercept)),
  Upper.CL = rep(NA, nrow(rich_beta_intercept)))

rich_modavg_beta_strata_matrix <- rich_modavg_beta_intercept_matrix

for(i in 1:nrow(rich_beta_intercept)){
  rich_modavg_beta_intercept_list[[i]] <- modavgCustom(logL = rich_aictab$LL,
                                                       K = rich_aictab$K,
                                                       modnames = rich_modnames,
                                                       estimate = rich_beta_intercept[i,],
                                                       se = rich_beta_intercept_se[i,],
                                                       nobs = nrow(rich))
  rich_modavg_beta_intercept_matrix$Mod.avg.est[i] <- rich_modavg_beta_intercept_list[[i]]$Mod.avg.est
  rich_modavg_beta_intercept_matrix$Uncond.SE[i] <- rich_modavg_beta_intercept_list[[i]]$Uncond.SE
  rich_modavg_beta_intercept_matrix$Lower.CL[i] <- rich_modavg_beta_intercept_list[[i]]$Lower.CL
  rich_modavg_beta_intercept_matrix$Upper.CL[i] <- rich_modavg_beta_intercept_list[[i]]$Upper.CL

  rich_modavg_beta_strata_list[[i]] <- modavgCustom(logL = rich_aictab$LL,
                                                    K = rich_aictab$K,
                                                    modnames = rich_modnames,
                                                    estimate = rich_beta_strata[i,],
                                                    se = rich_beta_strata_se[i,],
                                                    nobs = nrow(rich))
  rich_modavg_beta_strata_matrix$Mod.avg.est[i] <- rich_modavg_beta_strata_list[[i]]$Mod.avg.est
  rich_modavg_beta_strata_matrix$Uncond.SE [i] <- rich_modavg_beta_strata_list[[i]]$Uncond.SE
  rich_modavg_beta_strata_matrix$Lower.CL[i] <- rich_modavg_beta_strata_list[[i]]$Lower.CL
  rich_modavg_beta_strata_matrix$Upper.CL[i] <- rich_modavg_beta_strata_list[[i]]$Upper.CL
}

modavg_estimates_rich <- bind_rows(rich_modavg_beta_intercept_matrix,rich_modavg_beta_strata_matrix) %>
  dplyr::mutate(sig = if_else(sign(Lower.CL) == sign(Upper.CL), "*", ""))

modavg_estimates_rich$link <- model_estimates_rich$term
modavg_estimates_rich$facet <- model_estimates_rich$facet
```

Model averaged predictions of expected values

From Marc Mazerolle:

To get model averaged predictions from a glmmTMB model:

Doing what Marc said ^

```r
rich_pred_fit <- matrix(NA, nrow = nrow(rich), ncol = length(mods_rich))
rich_pred_se <- matrix(NA, nrow = nrow(rich), ncol = length(mods_rich))

for(i in 1:length(mods_rich)){
  rich_pred_fit[,i] <- predict(mods_rich[[i]], se.fit = T, type = "response")$fit
  rich_pred_se[,i] <- predict(mods_rich[[i]], se.fit = T, type = "response")$se.fit
}

rich_modavg_preds_list <- list()

rich_modavg_preds_matrix <- data.frame(
  Mod.avg.est = rep(NA, nrow(rich)),
  Uncond.SE = rep(NA, nrow(rich)),
  Lower.CL = rep(NA, nrow(rich)),
  Upper.CL = rep(NA, nrow(rich)))

for(i in 1:nrow(rich_pred_fit)){
  rich_modavg_preds_list[[i]] <- modavgCustom(logL = rich_aictab$LL,
                                   K = rich_aictab$K,
                                   modnames = rich_modnames,
                                   estimate = rich_pred_fit[i,],
                                   se = rich_pred_se[i,],
                                   nobs = nrow(rich_pred_fit))
  rich_modavg_preds_matrix$Mod.avg.est[i] <- rich_modavg_preds_list[[i]]$Mod.avg.est
  rich_modavg_preds_matrix$Uncond.SE [i] <- rich_modavg_preds_list[[i]]$Uncond.SE
  rich_modavg_preds_matrix$Lower.CL[i] <- rich_modavg_preds_list[[i]]$Lower.CL
  rich_modavg_preds_matrix$Upper.CL[i] <- rich_modavg_preds_list[[i]]$Upper.CL
}
```

Putting model estimates and predictions into something useful for plotting

```r
rich$predictions_fit <- rich_modavg_preds_matrix$Mod.avg.est
rich$predictions_se.fit <- rich_modavg_preds_matrix$Uncond.SE
rich$predictions_95CI_lower <- rich_modavg_preds_matrix$Lower.CL
rich$predictions_95CI_upper <- rich_modavg_preds_matrix$Upper.CL
rich$predictions_95CI_lower[rich$predictions_95CI_lower < 0] <- 0
rich$predictions_95CI_upper[rich$predictions_95CI_upper > 1] <- 1

taxa_link_rich <- rich %>%
  group_by(taxa_order,link,continent) %>%
  summarise() %>%
  full_join(modavg_estimates_rich, by = c("link")) %>%
  rename(CI_95_lower = Lower.CL,
         CI_95_upper = Upper.CL,
         se = Uncond.SE,
         estimate = Mod.avg.est) %>%
  mutate(CI_80_lower = estimate - se*1.28,
         CI_80_upper = estimate + se*1.28)
```

```
## `summarise()` regrouping output by 'taxa_order', 'link' (override with `.groups` argument)
```

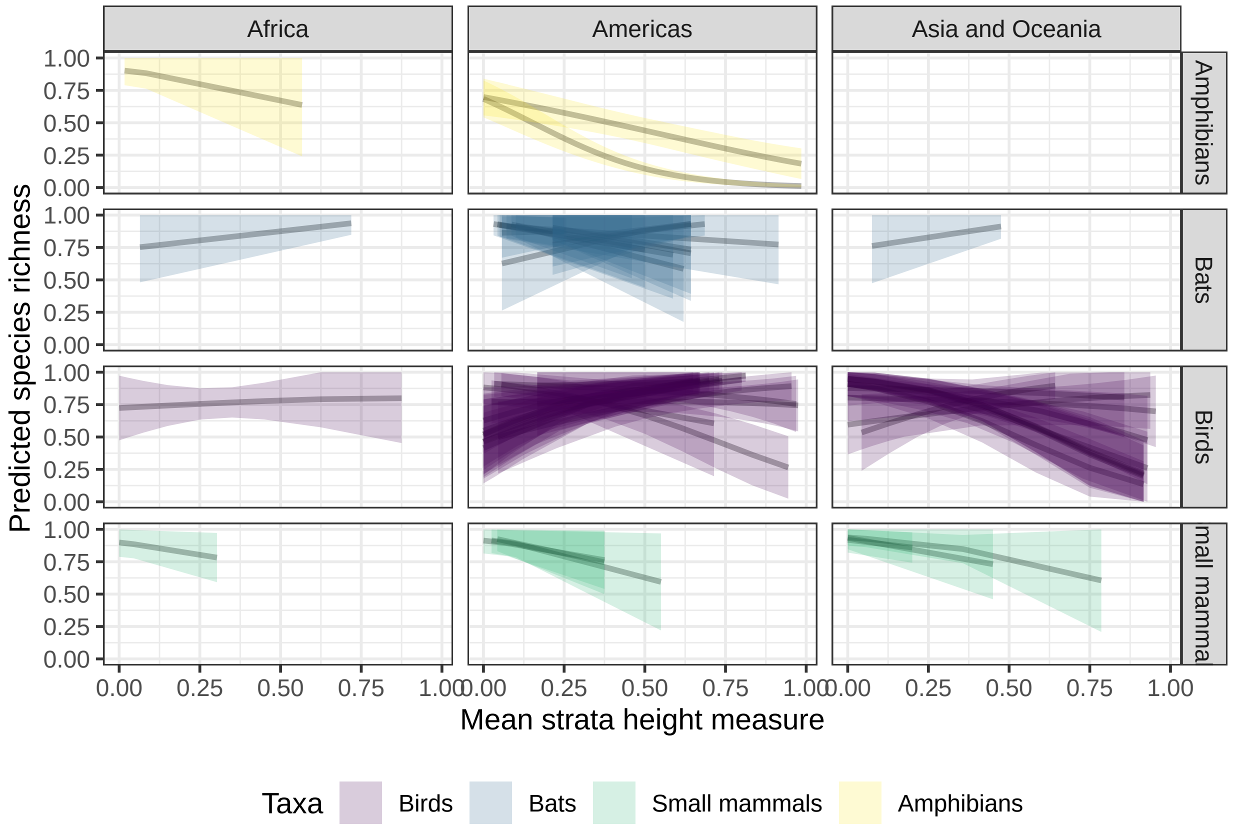
```
richness_intercept <- "Richness at lowest strata"
richness_slope <- "Change in richness with increasing strata"
taxa_link_rich$facet[taxa_link_rich$facet == "link (Intercept)"] <- richness_intercept
taxa_link_rich$facet[taxa_link_rich$facet == "strata"] <- richness_slope

taxa_link_rich_order <- taxa_link_rich %>%
  group_by(facet, factor(taxa_order, levels = c("Birds","Bats","Small mammals","Amphibians"))) %>%
  arrange(estimate, .by_group = T)

taxa_link_rich_order$order <- rep(NA, nrow(taxa_link_rich))
taxa_link_rich_order$order[taxa_link_rich_order$facet == richness_intercept] <-
  1:length(taxa_link_rich_order$order[taxa_link_rich_order$facet == richness_intercept])
taxa_link_rich_order$order[taxa_link_rich_order$facet == richness_slope] <-
  1:length(taxa_link_rich_order$order[taxa_link_rich_order$facet == richness_slope])

taxa_link_rich_avg <- taxa_link_rich_order %>%
  group_by(taxa_order, facet) %>%
  summarize(estimate = mean(estimate),
          CI_95_upper = mean(CI_95_upper),
          CI_95_lower = mean(CI_95_lower),
          CI_80_upper = mean(CI_80_upper),
          CI_80_lower = mean(CI_80_lower))
```

```
## `summarise()` regrouping output by 'taxa_order' (override with `.groups` argument)
```

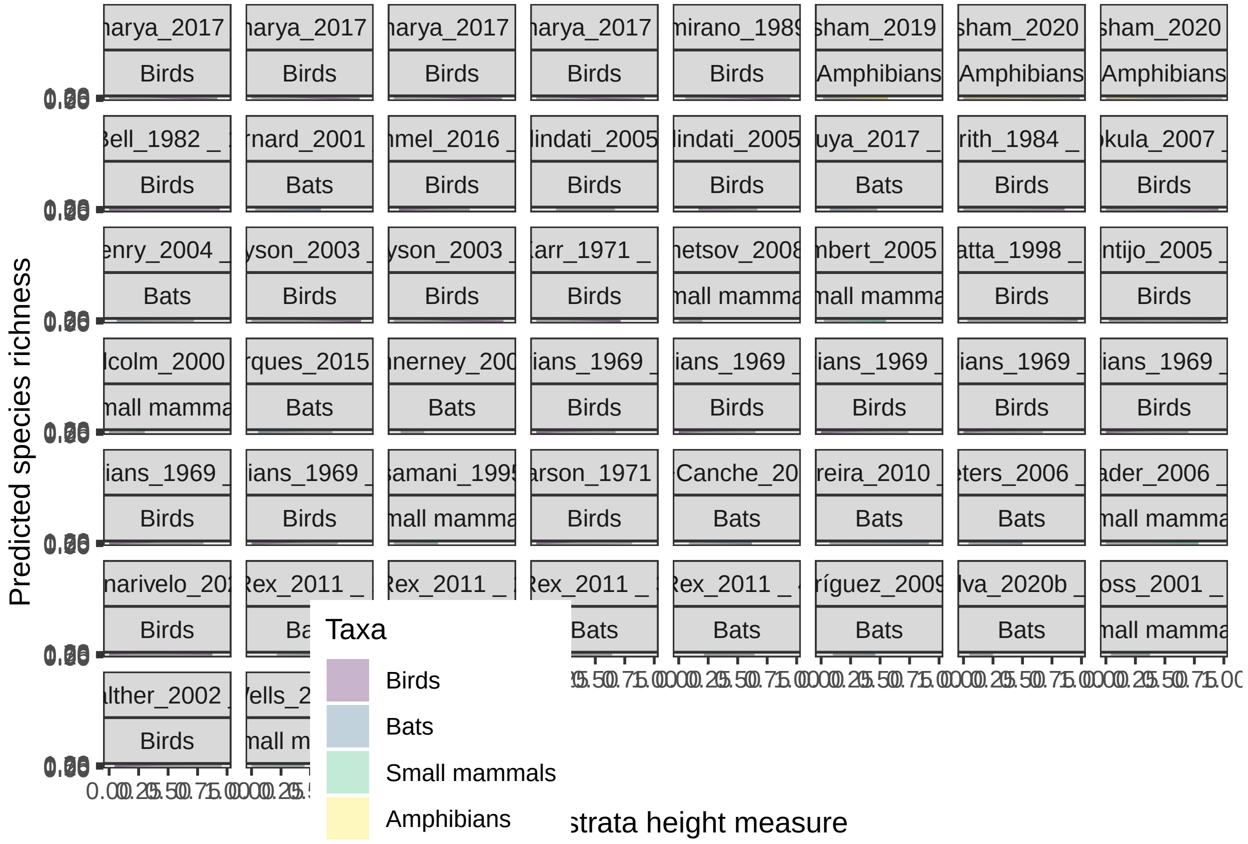
```
taxa_link_rich_avg$order <- rep(NA, nrow(taxa_link_rich_avg))
taxa_link_rich_avg$order[taxa_link_rich_avg$facet == richness_intercept] <-
  1:length(taxa_link_rich_avg$order[taxa_link_rich_avg$facet == richness_intercept])
taxa_link_rich_avg$order[taxa_link_rich_avg$facet == richness_slope] <-
  1:length(taxa_link_rich_avg$order[taxa_link_rich_avg$facet == richness_slope])

taxa_link_rich_order_slope <- taxa_link_rich_order %>%
  filter(facet == richness_slope) %>%
  dplyr::mutate(sig = if_else(sign(CI_95_upper) == sign(CI_95_lower), "*", ""))

taxa_link_rich_avg_slope <- taxa_link_rich_avg %>%
  filter(facet == richness_slope) %>%
  dplyr::mutate(sig = if_else(sign(CI_95_upper) == sign(CI_95_lower), "*", ""))
```

Plot predictions

```
ggplot(rich, aes(y = predictions_fit, x = strata,
                 ymin = predictions_95CI_lower,
                 ymax = predictions_95CI_upper,
                 fill = taxa_order, group = link)) +
    ylab("Predicted species richness") + xlab("Mean strata height measure") +
    geom_line(size = 1, alpha = 0.3) +
    geom_ribbon(color = NA, alpha = 0.2) +
    scale_fill_viridis_d("Taxa") +
    facet_grid(rows = vars(taxa), cols = vars(continent)) +
    theme_bw() + theme(legend.position = "bottom")
```

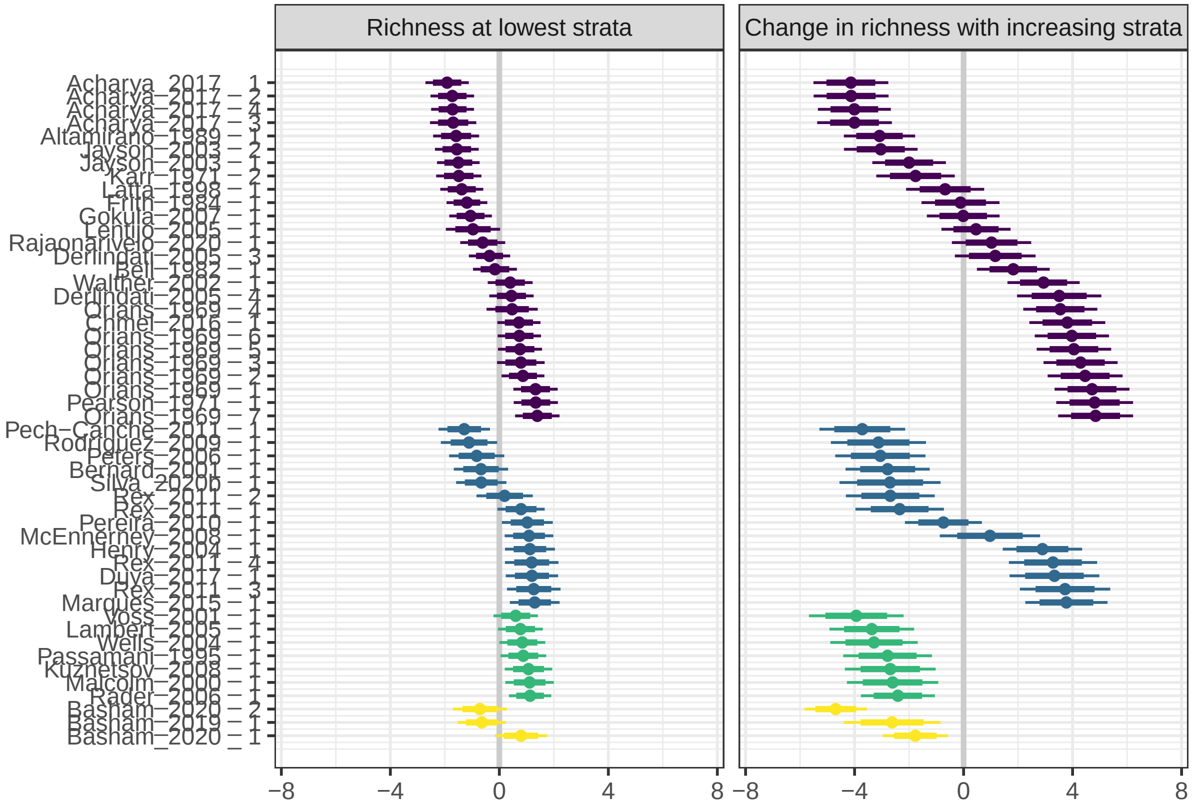


```
ggsave("analysis/figures/richness_strata_full_continent.jpeg", width = 8, height = 6, units = "in", dpi


ggplot(rich, aes(y = predictions_fit, x = strata,
                 ymin = predictions_95CI_lower,
                 ymax = predictions_95CI_upper,
                 fill = taxa_order)) +
    ylab("Predicted species richness") + xlab("Mean strata height measure") +
    geom_line(size = 1, alpha = 0.3) +
    geom_ribbon(color = NA, alpha = 0.3) +
    facet_wrap(~link + taxa) + theme_bw() +
    scale_fill_viridis_d("Taxa") + theme_bw() +
    theme(legend.position = c(0.3,0.06), legend.box = "horizontal")
```

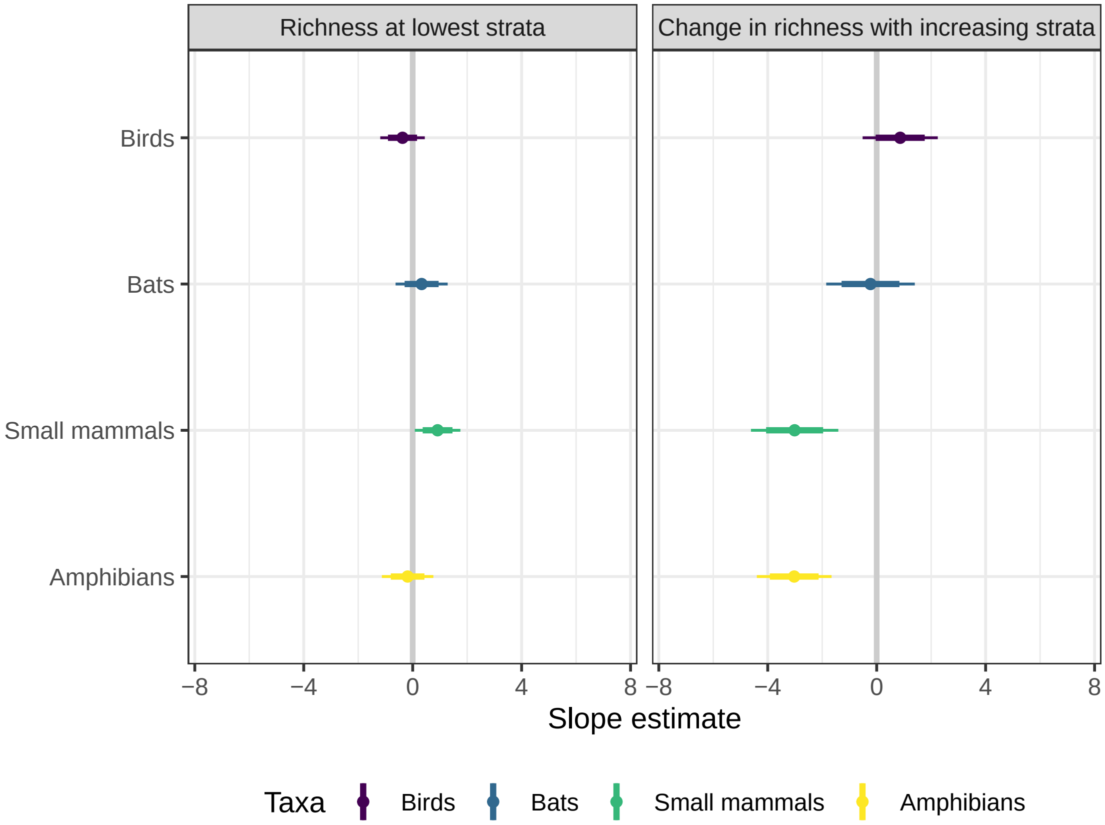


```r
ggsave("analysis/figures/richness_strata_full.jpeg", width = 18, height = 12, units = "in", dpi = 300)
```

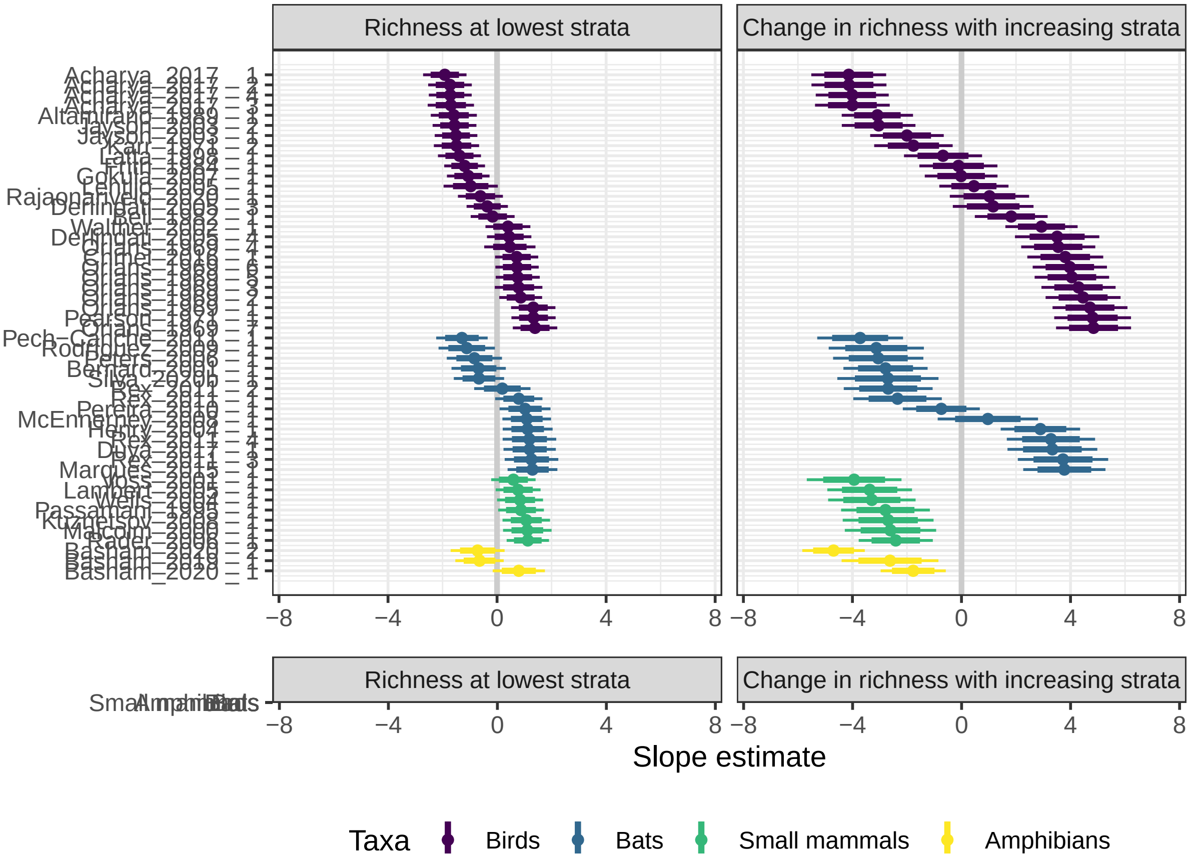
```r
## Two panel figure
col_2_rich_est <- ggplot(taxa_link_rich_order, aes(x = rev(order), y = estimate, col = taxa_order)) +
  ylab(" ") + xlab(" ") + coord_flip() + ylim(-7.5,7.5) +
  geom_hline(yintercept = 0, col = "grey80", size = 1, linetype = 1) + geom_point(size = 1.5) +
  geom_linerange(aes(ymax = CI_80_upper,
                     ymin = CI_80_lower), size = 1.1) +
  geom_linerange(aes(ymax = CI_95_upper,
                     ymin = CI_95_lower)) +
  scale_x_continuous(breaks = rev(taxa_link_rich_order$order[1:50]), labels = taxa_link_rich_order$link
  facet_grid(cols = vars(factor(facet, levels = c(richness_intercept, richness_slope)))) +
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "none"); col_2_rich_est
```



```
col_2_rich_est_avg <- ggplot(taxa_link_rich_avg, aes(x = rev(taxa_order), y = estimate, col = taxa_orde
  ylab("Slope estimate") + xlab(" ") + coord_flip() + ylim(-7.5,7.5) +
  geom_hline(yintercept = 0, col = "grey80", size = 1, linetype = 1) + geom_point(size = 1.5) +
  geom_linerange(aes(ymax = CI_80_upper,
                     ymin = CI_80_lower), size = 1.1) +
  geom_linerange(aes(ymax = CI_95_upper,
                     ymin = CI_95_lower)) +
  scale_x_discrete(labels = rev(c("Birds","Bats","             Small mammals","Amphibians"))) +
  facet_grid(cols = vars(factor(facet, levels = c(richness_intercept, richness_slope)))) +
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "bottom"); col_2_rich_est_avg
```



```
ggarrange(col_2_rich_est, col_2_rich_est_avg, ncol = 1, nrow = 2, heights = c(4,1), widths = c(4,1))
```



```
ggsave("analysis/figures/predictions_richness_parameters_estimates_taxa_2_panel.jpeg", width = 8, heigh
```

Abundance GLMs

```
mods_abund <- list()
# strata only models
mods_abund[[1]]  <- glmmTMB(scaled_met ~ 1 +                                                (strata|lin
mods_abund[[2]]  <- glmmTMB(scaled_met ~ strata +                                           (strata|lin
mods_abund[[3]]  <- glmmTMB(scaled_met ~ poly(strata, 2) +                                  (strata|lin
mods_abund[[4]]  <- glmmTMB(scaled_met ~ poly(strata, 3) +                                  (strata|lin
# strata and taxa
mods_abund[[5]]  <- glmmTMB(scaled_met ~ strata + taxa +                                    (strata|lin
mods_abund[[6]]  <- glmmTMB(scaled_met ~ poly(strata, 2) + taxa +                           (strata|lin
mods_abund[[7]]  <- glmmTMB(scaled_met ~ poly(strata, 3) + taxa +                         (strata|link)
# strata and continent
mods_abund[[8]]  <- glmmTMB(scaled_met ~ strata + continent +                               (strata|lin
mods_abund[[9]]  <- glmmTMB(scaled_met ~ poly(strata, 2) + continent +                      (strata|lin
mods_abund[[10]] <- glmmTMB(scaled_met ~ poly(strata, 3) + continent +                      (strata|lin
# strata and elevation
mods_abund[[11]] <- glmmTMB(scaled_met ~ strata + elevation +                               (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[12]] <- glmmTMB(scaled_met ~ poly(strata, 2) + elevation +                      (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[13]] <- glmmTMB(scaled_met ~ poly(strata, 3) + elevation +                      (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
# strata method
mods_abund[[14]] <- glmmTMB(scaled_met ~ strata + method +                                  (strata|lin
mods_abund[[15]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method +                         (strata|lin
mods_abund[[16]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method +                         (strata|lin
# strata, elevation, and continent
mods_abund[[17]] <- glmmTMB(scaled_met ~ strata + elevation + continent +                   (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[18]] <- glmmTMB(scaled_met ~ poly(strata, 2) + elevation + continent +          (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[19]] <- glmmTMB(scaled_met ~ poly(strata, 3) + elevation + continent +          (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
# strata, method, and continent
mods_abund[[20]] <- glmmTMB(scaled_met ~ strata + method + continent +                      (strata|lin
mods_abund[[21]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method + continent +             (strata|lin
mods_abund[[22]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method + continent +             (strata|lin
# strata, method, and elevation
mods_abund[[23]] <- glmmTMB(scaled_met ~ strata + method + continent + elevation +          (strata|lin
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[24]] <- glmmTMB(scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[25]] <- glmmTMB(scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
# strata interactions with methods or continent or elevation
mods_abund[[26]] <- glmmTMB(scaled_met ~ strata*method +                                    (strata|lin
mods_abund[[27]] <- glmmTMB(scaled_met ~ strata*continent +                                 (strata|lin
mods_abund[[28]] <- glmmTMB(scaled_met ~ strata*elevation +                                 (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
# strata interactions with methods or continent or elevation plus additive relationships continent and
mods_abund[[29]] <- glmmTMB(scaled_met ~ strata*method + continent +                        (strata|lin
mods_abund[[30]] <- glmmTMB(scaled_met ~ strata*method + elevation +                        (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[31]] <- glmmTMB(scaled_met ~ strata*continent + method +                        (strata|lin
mods_abund[[32]] <- glmmTMB(scaled_met ~ strata*continent + elevation +                     (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[33]] <- glmmTMB(scaled_met ~ strata*elevation + method +                        (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

```
mods_abund[[34]] <- glmmTMB(scaled_met ~ strata*elevation + continent +                     (strata|lin
```

```
## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation

## Warning in (function (start, objective, gradient = NULL, hessian = NULL, : NA/
## NaN function evaluation
```

Extracting model formulas to label aic table

```
abund_mod_names <- list()
for(i in 1:length(mods_abund)){
  abund_mod_names[[i]] <- as.character(mods_abund[[i]]$call$formula)
}

abund_modnames <- unlist(abund_mod_names)

abund_aictab <- aictab(cand.set = mods_abund, modnames = unlist(abund_mod_names))

bbmle::AICtab(mods_abund, mnames = unlist(abund_mod_names))
```

```
##                                                                                 dAIC
## scaled_met ~ poly(strata, 3) + method + (strata | link)                           0.0
## scaled_met ~ poly(strata, 3) + taxa + (strata | link)                             0.5
## scaled_met ~ poly(strata, 3) + method + continent + (strata | link)               3.9
## scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata | link)   4.0
## scaled_met ~ strata * method + (strata | link)                                   19.4
## scaled_met ~ strata * method + elevation + (strata | link)                       20.9
## scaled_met ~ strata * method + continent + (strata | link)                       22.9
## scaled_met ~ poly(strata, 2) + taxa + (strata | link)                            23.7
## scaled_met ~ poly(strata, 2) + method + (strata | link)                          25.0
## scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata | link)  28.9
## scaled_met ~ poly(strata, 2) + method + continent + (strata | link)              28.9
## scaled_met ~ poly(strata, 3) + (strata | link)                                   31.3
## scaled_met ~ poly(strata, 3) + elevation + (strata | link)                       32.8
## scaled_met ~ poly(strata, 3) + continent + (strata | link)                       33.7
## scaled_met ~ poly(strata, 3) + elevation + continent + (strata | link)           35.6
## scaled_met ~ strata * elevation + method + (strata | link)                       43.2
## scaled_met ~ strata + taxa + (strata | link)                                     46.6
## scaled_met ~ strata + method + (strata | link)                                   48.6
## scaled_met ~ strata + method + continent + (strata | link)                       52.5
## scaled_met ~ strata + method + continent + elevation + (strata | link)           53.6
## scaled_met ~ strata * continent + method + (strata | link)                       54.5
## scaled_met ~ poly(strata, 2) + (strata | link)                                   56.9
## scaled_met ~ poly(strata, 2) + elevation + (strata | link)                       58.2
## scaled_met ~ poly(strata, 2) + continent + (strata | link)                       59.2
## scaled_met ~ poly(strata, 2) + elevation + continent + (strata | link)           61.0
## scaled_met ~ strata * elevation + (strata | link)                                82.9
## scaled_met ~ strata * elevation + continent + (strata | link)                    85.7
## scaled_met ~ strata + elevation + (strata | link)                                86.9
## scaled_met ~ strata + (strata | link)                                            87.1
## scaled_met ~ strata + continent + (strata | link)                                89.0
## scaled_met ~ strata + elevation + continent + (strata | link)                    89.9
## scaled_met ~ strata * continent + (strata | link)                                92.2
## scaled_met ~ strata * continent + elevation + (strata | link)                    93.0
## scaled_met ~ 1 + (strata | link)                                                101.5
##                                                                                 df
## scaled_met ~ poly(strata, 3) + method + (strata | link)                      12
## scaled_met ~ poly(strata, 3) + taxa + (strata | link)                        11
## scaled_met ~ poly(strata, 3) + method + continent + (strata | link)          14
## scaled_met ~ poly(strata, 3) + method + continent + elevation + (strata | link) 15
## scaled_met ~ strata * method + (strata | link)                                  14
## scaled_met ~ strata * method + elevation + (strata | link)                      15
## scaled_met ~ strata * method + continent + (strata | link)                      16
## scaled_met ~ poly(strata, 2) + taxa + (strata | link)                        10
## scaled_met ~ poly(strata, 2) + method + (strata | link)                      11
## scaled_met ~ poly(strata, 2) + method + continent + elevation + (strata | link) 14
## scaled_met ~ poly(strata, 2) + method + continent + (strata | link)          13
## scaled_met ~ poly(strata, 3) + (strata | link)                                  8
## scaled_met ~ poly(strata, 3) + elevation + (strata | link)                      9
## scaled_met ~ poly(strata, 3) + continent + (strata | link)                      10
## scaled_met ~ poly(strata, 3) + elevation + continent + (strata | link)          11
## scaled_met ~ strata * elevation + method + (strata | link)                      12
## scaled_met ~ strata + taxa + (strata | link)                                    9
## scaled_met ~ strata + method + (strata | link)                                  10
## scaled_met ~ strata + method + continent + (strata | link)                      12
## scaled_met ~ strata + method + continent + elevation + (strata | link)          13
## scaled_met ~ strata * continent + method + (strata | link)                      14
## scaled_met ~ poly(strata, 2) + (strata | link)                                  7
## scaled_met ~ poly(strata, 2) + elevation + (strata | link)                      8
## scaled_met ~ poly(strata, 2) + continent + (strata | link)                      9
## scaled_met ~ poly(strata, 2) + elevation + continent + (strata | link)          10
## scaled_met ~ strata * elevation + (strata | link)                               8
## scaled_met ~ strata * elevation + continent + (strata | link)                   10
## scaled_met ~ strata + elevation + (strata | link)                               7
## scaled_met ~ strata + (strata | link)                                           6
## scaled_met ~ strata + continent + (strata | link)                               8
## scaled_met ~ strata + elevation + continent + (strata | link)                   9
## scaled_met ~ strata * continent + (strata | link)                               10
## scaled_met ~ strata * continent + elevation + (strata | link)                   11
## scaled_met ~ 1 + (strata | link)                                                5
```

Model predictions

Getting model esimates from top model

```
top_abund_mod <- mods_abund[[as.numeric(str_extract(aictab(mods_abund)[1,1], "(\\d)+"))]]
```

```
## Warning in aictab.AICglmmTMB(mods_abund):
## Model names have been supplied automatically in the table
```

```
model_estimates_abund <- sjPlot::get_model_data(top_abund_mod, type = "re", se = T, transform = NULL)

length_abund <- nrow(model_estimates_abund[model_estimates_abund$facet == "link (Intercept)",])
```

Getting model estimates from all models

```
abund_beta_intercept <- matrix(NA, nrow = length_abund, ncol = length(mods_abund))
abund_beta_intercept_se <- abund_beta_intercept
abund_beta_strata <- abund_beta_intercept
abund_beta_strata_se <- abund_beta_intercept

for(i in 1:length(mods_abund)){
  abund_beta_intercept[,i] <- sjPlot::get_model_data(mods_abund[[i]], type = "re", se = T, transform =
    select(estimate) %>%
    as.matrix()

  abund_beta_intercept_se[,i] <- sjPlot::get_model_data(mods_abund[[i]], type = "re", se = T, transform
    select(estimate, conf.low) %>%
    mutate(se = (estimate - conf.low)/1.96) %>%
    select(se) %>%
    as.matrix()

  abund_beta_strata[,i] <- sjPlot::get_model_data(mods_abund[[i]], type = "re", se = T, transform = NUL
    select(estimate) %>%
    as.matrix()

  abund_beta_strata_se[,i] <- sjPlot::get_model_data(mods_abund[[i]], type = "re", se = T, transform =
    select(estimate, conf.low) %>%
    mutate(se = (estimate - conf.low)/1.96) %>%
    select(se) %>%
    as.matrix()
}
```

Model averaging model estimates from all models

```
abund_modavg_beta_intercept_list <- list()
abund_modavg_beta_strata_list <- list()

abund_modavg_beta_intercept_matrix <- data.frame(
  Mod.avg.est = rep(NA, nrow(abund_beta_intercept)),
  Uncond.SE = rep(NA, nrow(abund_beta_intercept)),
  Lower.CL = rep(NA, nrow(abund_beta_intercept)),
  Upper.CL = rep(NA, nrow(abund_beta_intercept)))

abund_modavg_beta_strata_matrix <- abund_modavg_beta_intercept_matrix

for(i in 1:nrow(abund_beta_intercept)){
  abund_modavg_beta_intercept_list[[i]] <- modavgCustom(logL = abund_aictab$LL,
                                                        K = abund_aictab$K,
                                                        modnames = abund_modnames,
                                                        estimate = abund_beta_intercept[i,],
                                                        se = abund_beta_intercept_se[i,],
                                                        nobs = nrow(abund))
  abund_modavg_beta_intercept_matrix$Mod.avg.est[i] <- abund_modavg_beta_intercept_list[[i]]$Mod.avg.es
  abund_modavg_beta_intercept_matrix$Uncond.SE [i] <- abund_modavg_beta_intercept_list[[i]]$Uncond.SE
  abund_modavg_beta_intercept_matrix$Lower.CL[i] <- abund_modavg_beta_intercept_list[[i]]$Lower.CL
  abund_modavg_beta_intercept_matrix$Upper.CL[i] <- abund_modavg_beta_intercept_list[[i]]$Upper.CL

  abund_modavg_beta_strata_list[[i]] <- modavgCustom(logL = abund_aictab$LL,
                                                     K = abund_aictab$K,
                                                     modnames = abund_modnames,
                                                     estimate = abund_beta_strata[i,],
                                                     se = abund_beta_strata_se[i,],
                                                     nobs = nrow(abund))
  abund_modavg_beta_strata_matrix$Mod.avg.est[i] <- abund_modavg_beta_strata_list[[i]]$Mod.avg.est
  abund_modavg_beta_strata_matrix$Uncond.SE [i] <- abund_modavg_beta_strata_list[[i]]$Uncond.SE
  abund_modavg_beta_strata_matrix$Lower.CL[i] <- abund_modavg_beta_strata_list[[i]]$Lower.CL
  abund_modavg_beta_strata_matrix$Upper.CL[i] <- abund_modavg_beta_strata_list[[i]]$Upper.CL
}

modavg_estimates_abund <- bind_rows(abund_modavg_beta_intercept_matrix,abund_modavg_beta_strata_matrix)
  dplyr::mutate(sig = if_else(sign(Lower.CL) == sign(Upper.CL), "*", ""))

modavg_estimates_abund$link <- model_estimates_abund$term
modavg_estimates_abund$facet <- model_estimates_abund$facet
```

Model averaged predictions of expected values

From Marc Mazerolle:

To get model averaged predictions from a glmmTMB model:

```


abund_pred_fit <- matrix(NA, nrow = nrow(abund), ncol = length(mods_abund))
abund_pred_se <- matrix(NA, nrow = nrow(abund), ncol = length(mods_abund))

for(i in 1:length(mods_abund)){
  abund_pred_fit[,i] <- predict(mods_abund[[i]], se.fit = T, type = "response")$fit
  abund_pred_se[,i] <- predict(mods_abund[[i]], se.fit = T, type = "response")$se.fit
}

abund_modavg_preds_list <- list()
```

```
abund_modavg_preds_matrix <- data.frame(
  Mod.avg.est = rep(NA, nrow(abund)),
  Uncond.SE = rep(NA, nrow(abund)),
  Lower.CL = rep(NA, nrow(abund)),
  Upper.CL = rep(NA, nrow(abund)))

for(i in 1:nrow(abund_pred_fit)){
  abund_modavg_preds_list[[i]] <- modavgCustom(logL = abund_aictab$LL,
                                  K = abund_aictab$K,
                                  modnames = abund_modnames,
                                  estimate = abund_pred_fit[i,],
                                  se = abund_pred_se[i,],
                                  nobs = nrow(abund_pred_fit))
  abund_modavg_preds_matrix$Mod.avg.est[i] <- abund_modavg_preds_list[[i]]$Mod.avg.est
  abund_modavg_preds_matrix$Uncond.SE [i] <- abund_modavg_preds_list[[i]]$Uncond.SE
  abund_modavg_preds_matrix$Lower.CL[i] <- abund_modavg_preds_list[[i]]$Lower.CL
  abund_modavg_preds_matrix$Upper.CL[i] <- abund_modavg_preds_list[[i]]$Upper.CL
}
```

```
abund$predictions_fit <- abund_modavg_preds_matrix$Mod.avg.est
abund$predictions_se.fit <- abund_modavg_preds_matrix$Uncond.SE
abund$predictions_95CI_lower <- abund_modavg_preds_matrix$Lower.CL
abund$predictions_95CI_upper <- abund_modavg_preds_matrix$Upper.CL
abund$predictions_95CI_lower[abund$predictions_95CI_lower < 0] <- 0
abund$predictions_95CI_upper[abund$predictions_95CI_upper > 1] <- 1

taxa_link_abund <- abund %>%
  group_by(taxa_order,link,continent) %>%
  summarise() %>%
  full_join(modavg_estimates_abund, by = c("link")) %>%
  rename(CI_95_lower = Lower.CL,
         CI_95_upper = Upper.CL,
         se = Uncond.SE,
         estimate = Mod.avg.est) %>%
  mutate(CI_80_lower = estimate - se*1.28,
         CI_80_upper = estimate + se*1.28)
```

Doing what Marc said ^

```
## `summarise()` regrouping output by 'taxa_order', 'link' (override with `.groups` argument)
```

```
abundance_intercept <- "Abundance at lowest strata"
abundance_slope <- "Change in abundance with increasing strata"
taxa_link_abund$facet[taxa_link_abund$facet == "link (Intercept)"] <- abundance_intercept
taxa_link_abund$facet[taxa_link_abund$facet == "strata"] <- abundance_slope

taxa_link_abund_order <- taxa_link_abund %>%
  group_by(facet, factor(taxa_order, levels = c("Birds","Bats","Small mammals","Amphibians"))) %>%
  arrange(estimate, .by_group = T)

taxa_link_abund_order$order <- rep(NA, nrow(taxa_link_abund))
```

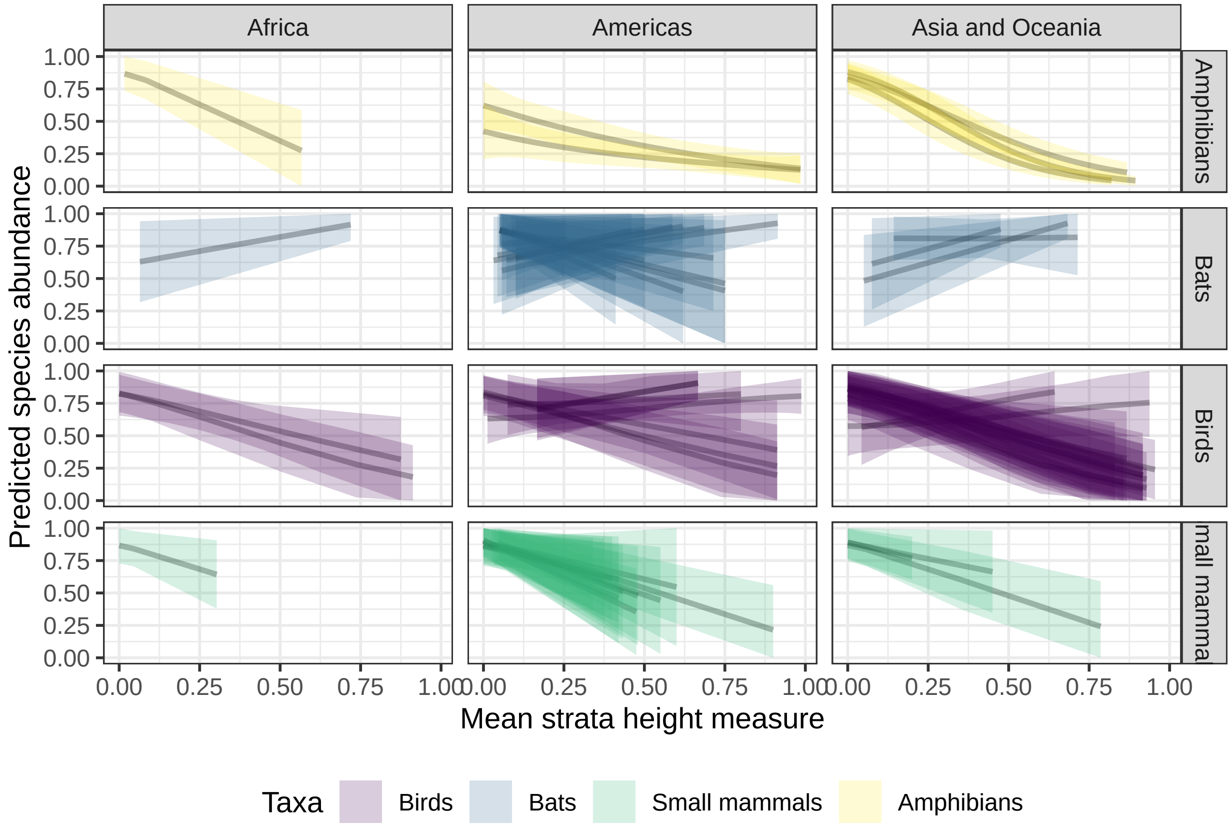
```
taxa_link_abund_order$order[taxa_link_abund_order$facet == abundance_intercept] <-
  1:length(taxa_link_abund_order$order[taxa_link_abund_order$facet == abundance_intercept])
taxa_link_abund_order$order[taxa_link_abund_order$facet == abundance_slope] <-
  1:length(taxa_link_abund_order$order[taxa_link_abund_order$facet == abundance_slope])

taxa_link_abund_avg <- taxa_link_abund_order %>%
  group_by(taxa_order, facet) %>%
  summarize(estimate = mean(estimate),
          CI_95_upper = mean(CI_95_upper),
          CI_95_lower = mean(CI_95_lower),
          CI_80_upper = mean(CI_80_upper),
          CI_80_lower = mean(CI_80_lower))
```

```
## `summarise()` regrouping output by 'taxa_order' (override with `.groups` argument)
```

```
taxa_link_abund_avg$order <- rep(NA, nrow(taxa_link_abund_avg))
taxa_link_abund_avg$order[taxa_link_abund_avg$facet == abundance_intercept] <-
  1:length(taxa_link_abund_avg$order[taxa_link_abund_avg$facet == abundance_intercept])
taxa_link_abund_avg$order[taxa_link_abund_avg$facet == abundance_slope] <-
  1:length(taxa_link_abund_avg$order[taxa_link_abund_avg$facet == abundance_slope])

taxa_link_abund_order_slope <- taxa_link_abund_order %>%
  filter(facet == abundance_slope) %>%
  dplyr::mutate(sig = if_else(sign(CI_95_upper) == sign(CI_95_lower), "*", ""))

taxa_link_abund_avg_slope <- taxa_link_abund_avg %>%
  filter(facet == abundance_slope) %>%
  dplyr::mutate(sig = if_else(sign(CI_95_upper) == sign(CI_95_lower), "*", ""))
```

Plot predictions

```
ggplot(abund, aes(y = predictions_fit, x = strata,
                 ymin = predictions_95CI_lower,
                 ymax = predictions_95CI_upper,
                 fill = taxa_order, group = link)) +
  ylab("Predicted species abundance") + xlab("Mean strata height measure") +
  geom_line(size = 1, alpha = 0.3) +
  geom_ribbon(color = NA, alpha = 0.2) +
  scale_fill_viridis_d("Taxa") +
  facet_grid(rows = vars(taxa), cols = vars(continent)) + theme_bw() +
  theme(legend.position = "bottom");
```

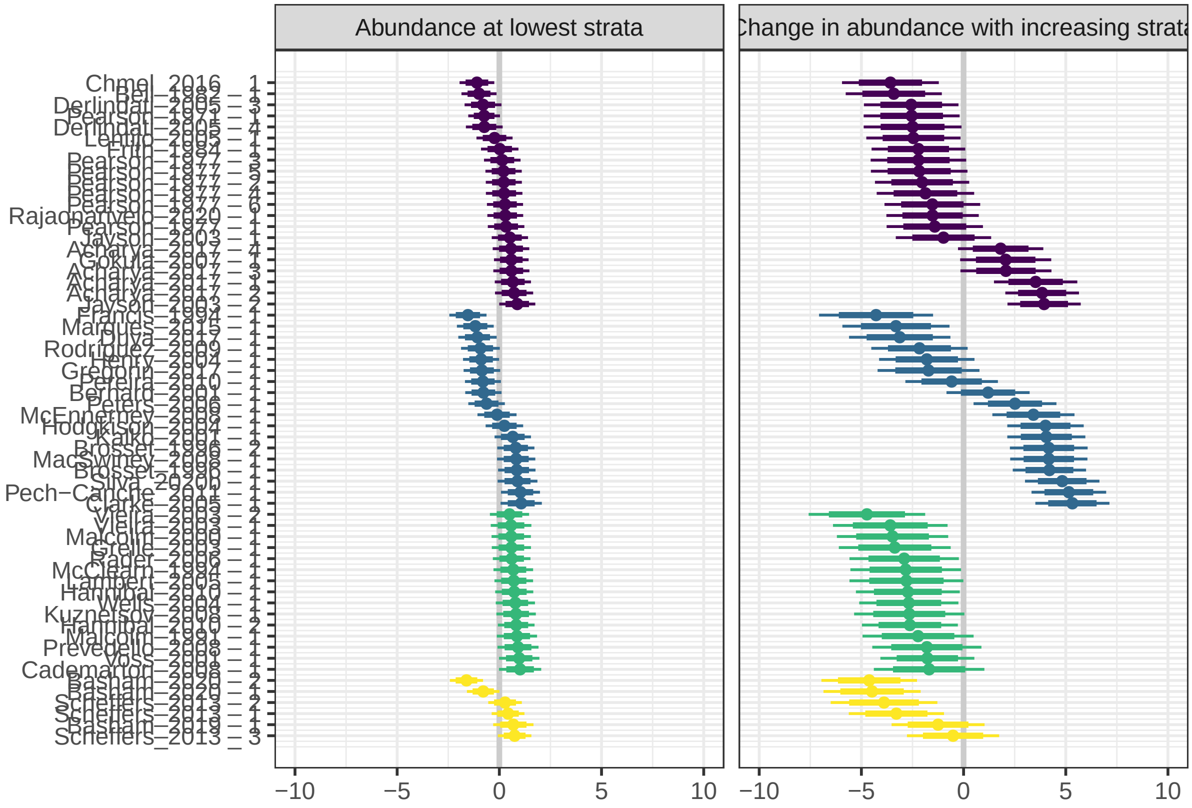


```
ggsave("analysis/figures/abundance_strata_full_continent.jpeg", width = 8, height = 6, units = "in", dp

ggplot(abund, aes(y = predictions_fit, x = strata,
                     ymin = predictions_95CI_lower,
                     ymax = predictions_95CI_upper,
                     fill = taxa_order)) +
  ylab("Predicted species abundance") + xlab("Mean strata height measure") +
  geom_line(size = 1, alpha = 0.3) +
  geom_ribbon(color = NA, alpha = 0.3) +
  facet_wrap(~link + taxa) + theme_bw() +
  scale_fill_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = c(0.54,0.05), legend.box = "horizontal");
```

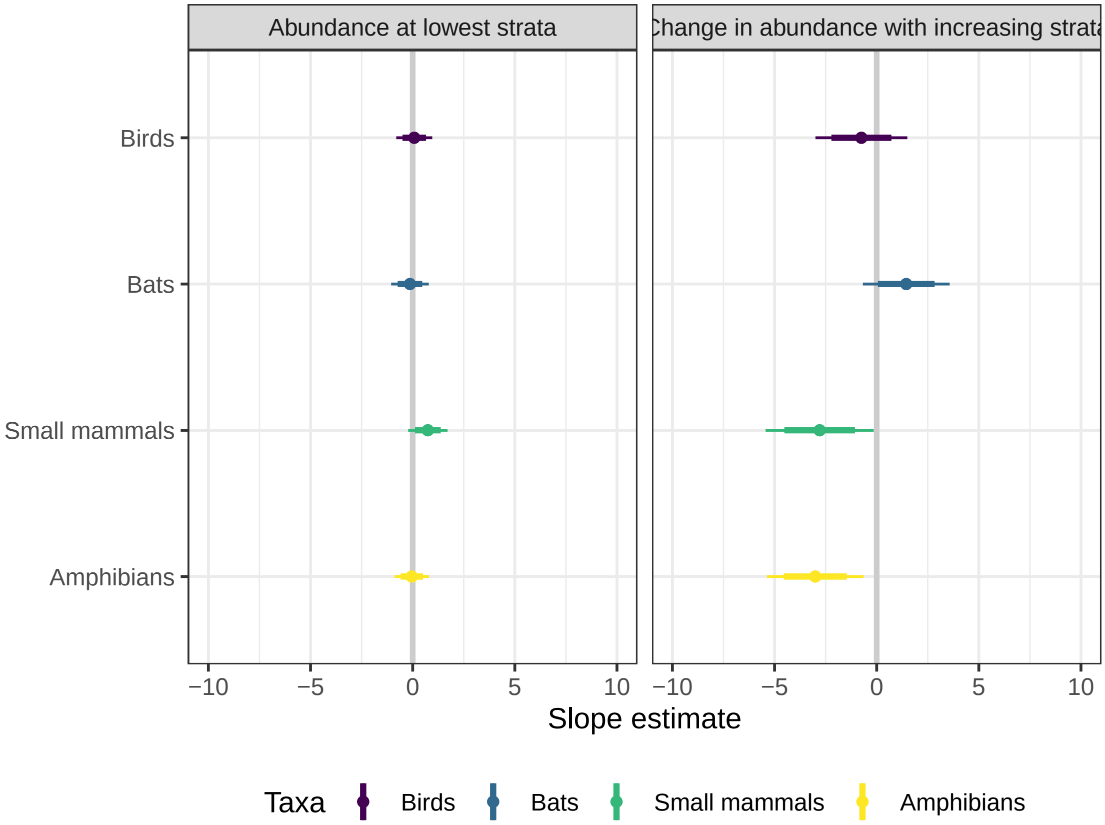
Predicted species abundance
0.00
larya_2017 _ larya_2017 _ larya_2017 _ larya_2017 _ sham_2019 _ sham_2020 _ sham_2020 _ sell_1982 _
Birds Birds Birds Birds Amphibians Amphibians Amphibians Birds
rnard_2001 osset_1996 _ osset_1996 _ martori_200 nmel_2016 _ arke_2005 _ lindati_2005 lindati_2005
Bats Bats Bats nall mamma Birds Bats Birds Birds
uya_2017 _ ancis_1994 _ rith_1984 _ okula_2007 _ gorin_2017 relle_2003 _ nnibal_2010 nnibal_2010
Bats Bats Birds Birds Bats nall mamma nall mamma nall mamma
enry_2004 _ gkison_2004 yson_2003 _ yson_2003 _ alko_2001 _ netsov_2008 nbert_2005 ntijo_2005 _
Bats Bats Birds Birds Bats nall mamma nall mamma Birds
Swiney_200 lcolm_1991 lcolm_2000 rques_2015 Clearn_1994 nerney_200 arson_1971 arson_1977
Bats nall mamma nall mamma Bats nall mamma Bats Birds Birds
arson_1977 arson_1977 arson_1977 arson_1977 arson_1977 Canche_20 reira_2010 _ eters_2006 _
Birds Birds Birds Birds Birds Bats Bats Bats
edello_2008 ader_2006 _ narivelo_20 ríguez_2009 effers_2013 effers_2013 effers_2013 lva_2020b _
nall mamma nall mamma Birds B nphibians Amphibians Bats
eira_2003 _ eira_2003 _ oss_2001 _ ells_
nall mamma nall mamma nall mamma nall r
0.000.250.500.751.00
Mean
Taxa
Birds
Bats
Small mammals
Amphibians

```
ggsave("analysis/figures/abundance_strata_full.jpeg", width = 18, height = 12, units = "in", dpi = 300)
```

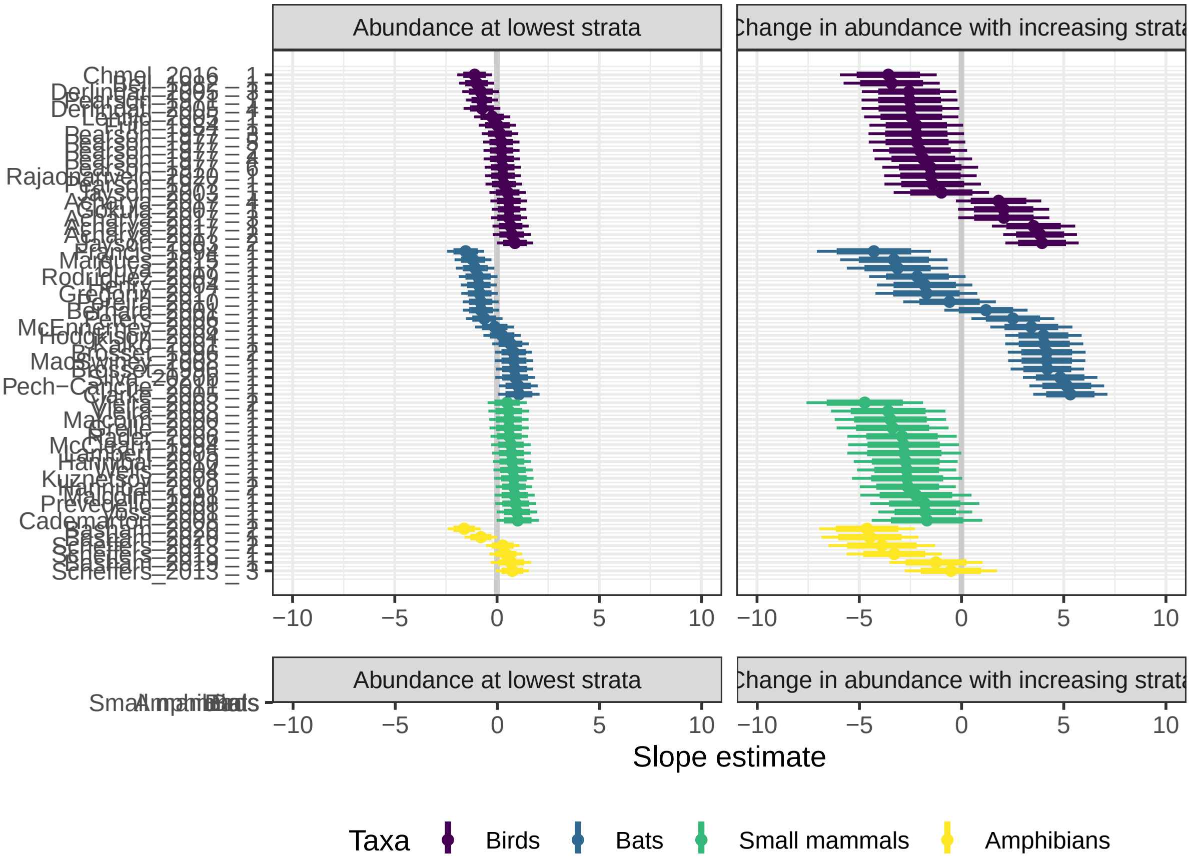
```
## Two panel figure
col_2_abund_est <- ggplot(taxa_link_abund_order, aes(x = rev(order), y = estimate, col = taxa_order)) +
  ylab(" ") + xlab(" ") + coord_flip() + ylim(-10,10) +
  geom_hline(yintercept = 0, col = "grey80", size = 1, linetype = 1) + geom_point(size = 1.5) +
  geom_linerange(aes(ymax = CI_80_upper,
                     ymin = CI_80_lower), size = 1.1) +
  geom_linerange(aes(ymax = CI_95_upper,
                     ymin = CI_95_lower)) +
  scale_x_continuous(breaks = rev(taxa_link_abund_order$order[1:length_abund]), labels = taxa_link_abun
  facet_grid(cols = vars(factor(facet, levels = c(abundance_intercept, abundance_slope)))) +
  scale_color_viridis_d("Taxa") + theme_bw() + theme(legend.position = "none"); col_2_abund_est
```



```
col_2_abund_est_avg <- ggplot(taxa_link_abund_avg, aes(x = rev(taxa_order), y = estimate, col = taxa_ord
  ylab("Slope estimate") + xlab(" ") + coord_flip() + ylim(-10,10) +
  geom_hline(yintercept = 0, col = "grey80", size = 1, linetype = 1) + geom_point(size = 1.5) +
  geom_linerange(aes(ymax = CI_80_upper,
                     ymin = CI_80_lower), size = 1.1) +
  geom_linerange(aes(ymax = CI_95_upper,
                     ymin = CI_95_lower)) +
  scale_x_discrete(labels = rev(c("Birds","Bats","               Small mammals","Amphibians"))) +
  facet_grid(cols = vars(factor(facet, levels = c(abundance_intercept, abundance_slope)))) +
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "bottom"); col_2_abund_est_avg
```



```
ggarrange(col_2_abund_est, col_2_abund_est_avg, ncol = 1, nrow = 2, heights = c(4,1), widths = c(4,1))
```



```
ggsave("analysis/figures/predictions_abundance_parameters_estimates_taxa_2_panel.jpeg", width = 8, heig
```

staggered richness and abundance plots (I don't like this. . .)

```
# taxa_link_abund_order$metric <- "abund"
# taxa_link_rich_order$metric <- "rich"
#
# diversity_metric_preds <- bind_rows(taxa_link_abund_order, taxa_link_rich_order)
#
# bio_intercept <- "Metric at lowest strata"
# bio_slope <- "Change in metric with increasing strata"
# diversity_metric_preds$facet[diversity_metric_preds$facet %in% c(abundance_intercept, richness_interc
# diversity_metric_preds$facet[diversity_metric_preds$facet %in% c(abundance_slope, richness_slope)] <-
#
# diversity_metric_preds_order <- diversity_metric_preds %>%
#   dplyr::group_by(facet, factor(taxa_order, levels = c("Bats","Birds","Small mammals","Amphibians")),
#   dplyr::arrange(estimate, .by_group = T)
#
# diversity_metric_preds_order$order <- rep(NA, nrow(diversity_metric_preds_order))
#
# diversity_intercept_length <- length(diversity_metric_preds_order$order[diversity_metric_preds_order$
# diversity_slope_length <- length(diversity_metric_preds_order$order[diversity_metric_preds_order$face
# diversity_metric_preds_order$order[diversity_metric_preds_order$facet == bio_intercept] <- 1:diversit
# diversity_metric_preds_order$order[diversity_metric_preds_order$facet == bio_slope] <- 1:diversity_sl
#
# diversity_metric_preds_avg <- diversity_metric_preds %>%
#   group_by(metric, facet, taxa_order) %>%
#   summarize(estimate = mean(estimate),
#           CI_95_upper = mean(CI_95_upper),
#           CI_95_lower = mean(CI_95_lower),
#           CI_80_upper = mean(CI_80_upper),
#           CI_80_lower = mean(CI_80_lower)) %>%
#   arrange(estimate, .by_group = T)
#
# diversity_metric_preds_avg$order <- rep(NA, nrow(diversity_metric_preds_avg))
#
# diversity_metric_preds_avg$order[diversity_metric_preds_avg$facet == bio_intercept] <-
#   1:length(diversity_metric_preds_avg$order[diversity_metric_preds_avg$facet == bio_intercept])
# diversity_metric_preds_avg$order[diversity_metric_preds_avg$facet == bio_slope] <-
#   1:length(diversity_metric_preds_avg$order[diversity_metric_preds_avg$facet == bio_slope])
```

More intercept and slope figures (I don't like this so much either. . .)

```
# ## Two panel figure
# pos_dodge <- 0.8
#
# taxa_col <- viridis::viridis(4)
# taxa_col_link <- list(
#   "Birds" = taxa_col[1],
#   "Bats" = taxa_col[2],
#   "Small mammals" = taxa_col[3],
#   "Amphibians" = taxa_col[4]
# )
#
# diversity_metric_preds_order$taxa_col <- unname(unlist(taxa_col_link[diversity_metric_preds_order$tax
#
# col_2_bio_est <- ggplot(diversity_metric_preds_order, aes(x = reorder(link, order), y = estimate, col
#   ylab(" ") + xlab(" ") + ylim(-8,8) + coord_flip() +
#   geom_hline(yintercept = 0, alpha = 0.5, size = 1) +
#   geom_linerange(aes(ymin = CI_80_lower,
#                      ymax = CI_80_upper), size = 1.1, alpha = 0.5,
#                  position = position_dodge2(width = pos_dodge,
#                                             preserve = "total",
#                                             padding = 0.5)) +
#   geom_linerange(aes(ymin = CI_95_lower,
#                      ymax = CI_95_upper), alpha = 0.8,
#                  position = position_dodge2(width = pos_dodge,
#                                             preserve = "total",
#                                             padding = 0.5)) +
#   geom_linerange(aes(ymin = estimate - 0.1,
#                      ymax = estimate + 0.1), size = 2, alpha = 0.5,
#                  position = position_dodge2(width = pos_dodge,
#                                             preserve = "total",
#                                             padding = 0.5)) +
#   facet_grid(cols = vars(factor(facet, levels = c(bio_intercept, bio_slope)))) +
#   scale_x_discrete(labels = diversity_metric_preds_order$link[diversity_metric_preds_order$facet == b
#   scale_color_viridis_d("Taxa") + theme_bw() +
#   theme(legend.position = "bottom"); col_2_bio_est
#
# ggsave("analysis/figures/predictions_bio_parameters_link_taxa_2_panel_dodge.jpeg", width = 8, height
```

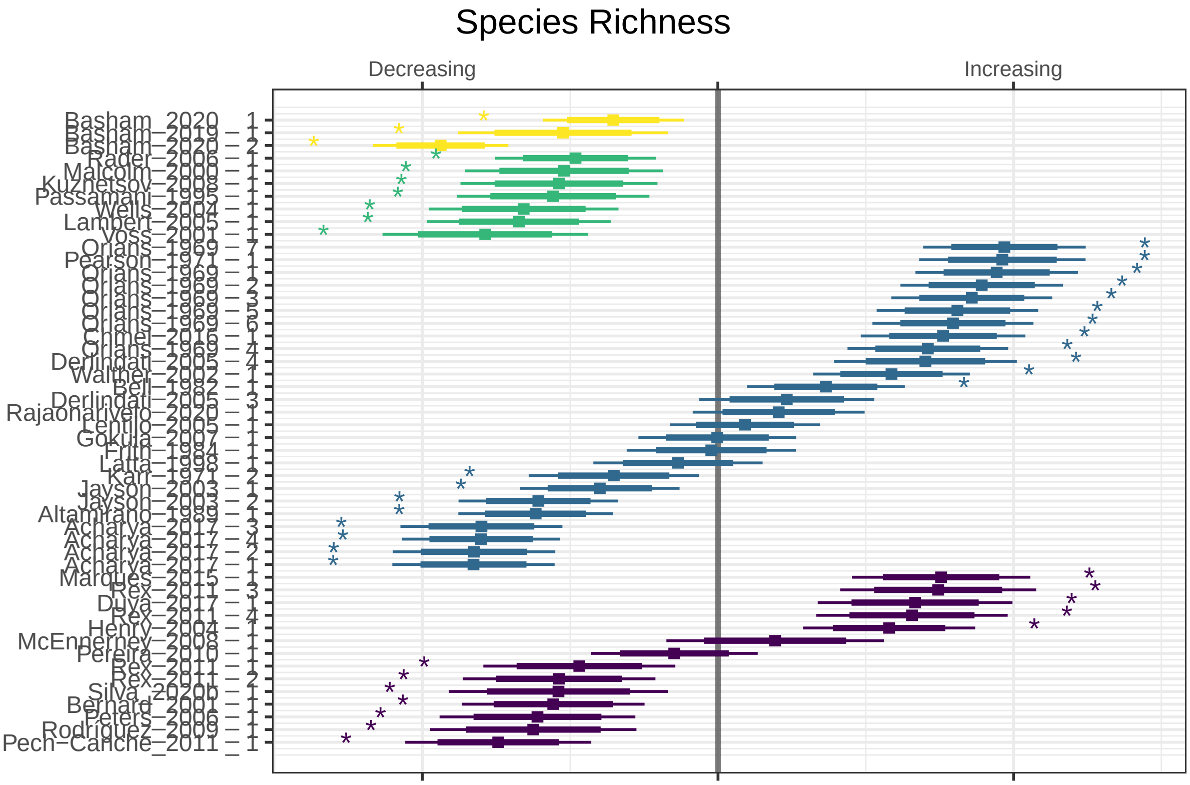
Plotting slopes of abundance and richness

```
taxa_link_slope <- bind_rows(
  data.frame(taxa_link_abund_order_slope, metric = "abundance"),
  data.frame(taxa_link_rich_order_slope, metric = "richness")) %>%
  select(!c(facet, factor....., continent, order)) %>%
  group_by(taxa_order = factor(taxa_order, levels = c("Bats","Birds","Small mammals","Amphibians"))) %>
  arrange(estimate, .by_group = T)
taxa_link_order_slope <- taxa_link_slope
taxa_link_order_slope$order <- rep(NA, nrow(taxa_link_order_slope))
taxa_link_order_slope$order[taxa_link_slope$metric == "richness"] <- 1:length(taxa_link_order_slope$orde
taxa_link_order_slope$order[taxa_link_slope$metric == "abundance"] <- 1:length(taxa_link_order_slope$ord

taxa_link_slope_rich <- taxa_link_slope %>%
  filter(metric == "richness") %>%
  rowid_to_column("order")
taxa_link_slope_abund <- taxa_link_slope %>%
  filter(metric == "abundance") %>%
  rowid_to_column("order")
```

Creating slope estimate figures for both richness and abundance

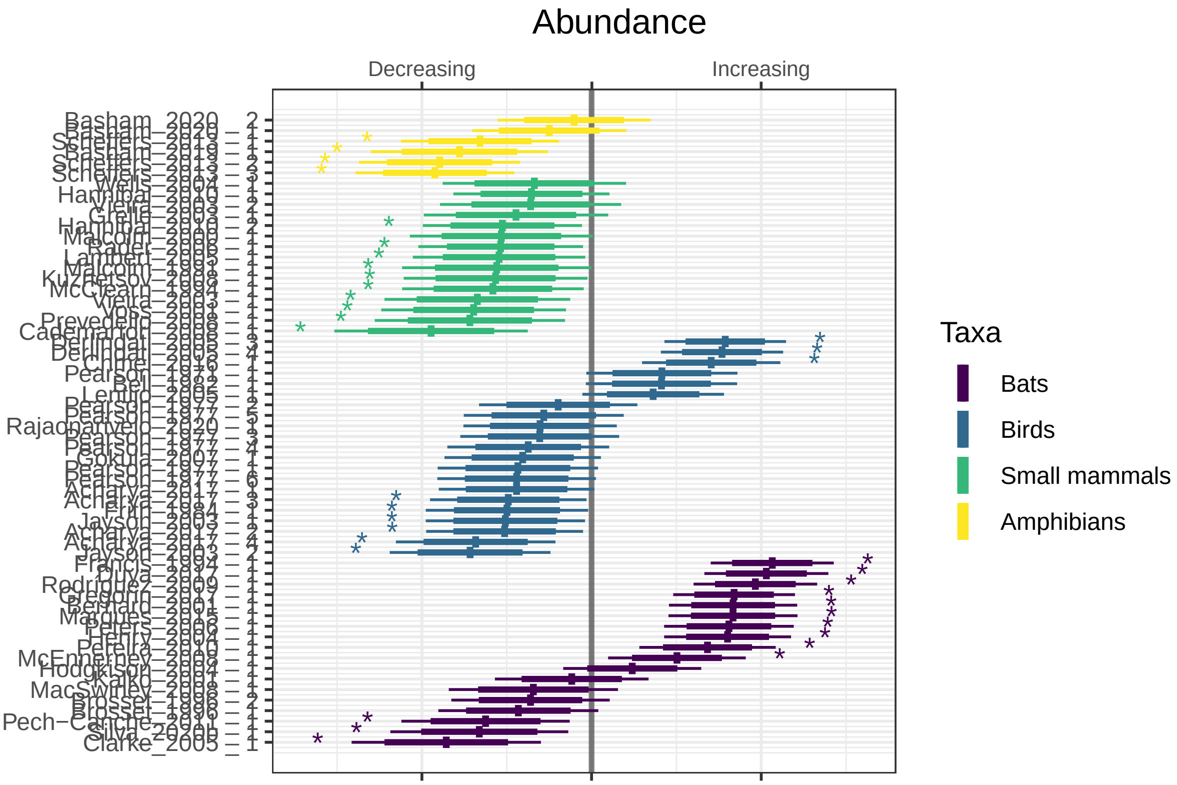
```
col_2_slope_rich_est <- ggplot(taxa_link_slope_rich, aes(x = order, y = estimate, col = taxa_order)) +
  ylab(" ") + xlab(" ") + ggtitle("                   Species Richness") + ylim(-9,9) + coord_flip() +
  geom_hline(yintercept = 0, alpha = 0.5, size = 1) +
  geom_linerange(aes(ymin = CI_80_lower,
                     ymax = CI_80_upper), size = 1.1) +
  geom_linerange(aes(ymin = CI_95_lower,
                     ymax = CI_95_upper)) +
  geom_linerange(aes(ymin = estimate - 0.1,
                     ymax = estimate + 0.1), size = 2) +
  scale_x_continuous(breaks = taxa_link_slope_rich$order,
                     labels = taxa_link_slope_rich$link) +
  scale_y_continuous(breaks = c(-5,0,5), labels = c("","",""),
                     sec.axis = sec_axis(~., breaks = c(-5,0,5), labels = c("Decreasing"," ","Increasin
  geom_text(aes(x = order, y = if_else(sign(estimate) == 1, CI_95_upper + 1, CI_95_lower - 1), label =
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "none", axis.text.x.top = element_text(size = 8)); col_2_slope_rich_est
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

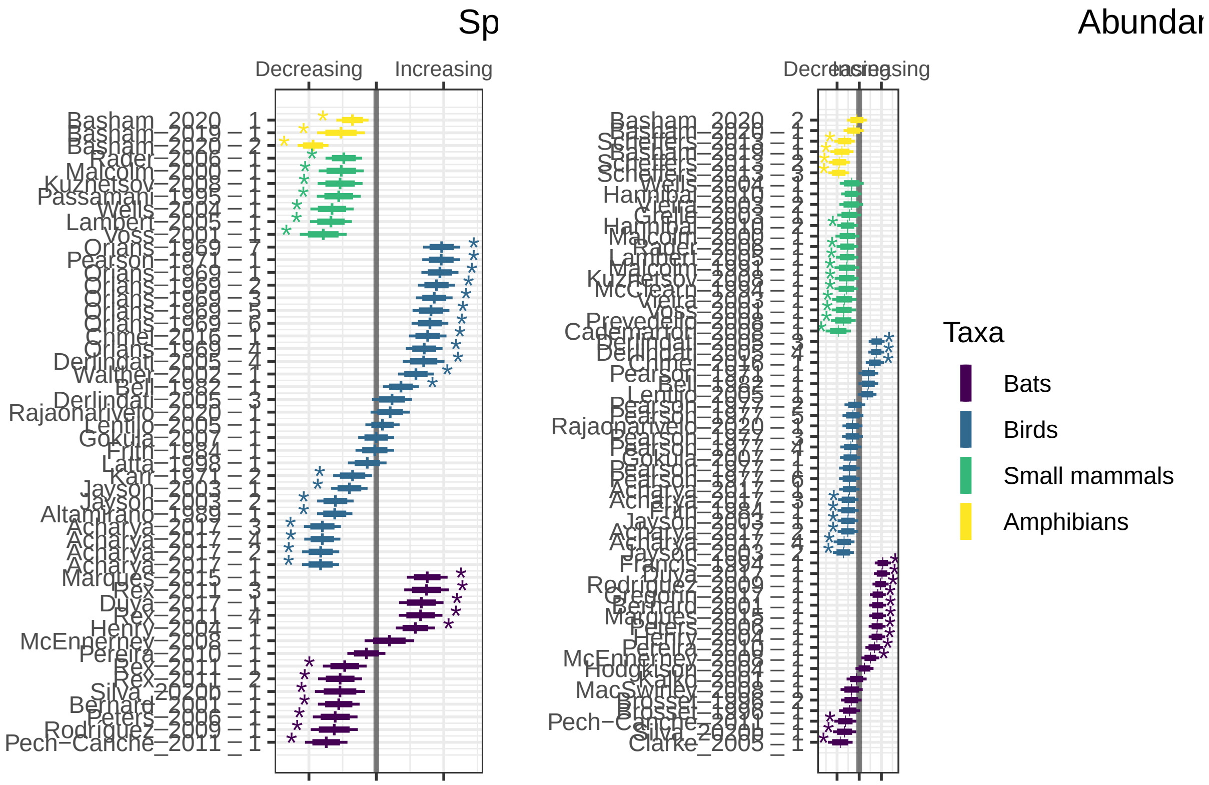


```
col_2_slope_abund_est <- ggplot(taxa_link_slope_abund, aes(x = order, y = estimate, col = taxa_order)) +
   ylab(" ") + xlab(" ") + ggtitle("                              Abundance") + ylim(-9,9) + coord_flip() +
  geom_hline(yintercept = 0, alpha = 0.5, size = 1) +
  geom_linerange(aes(ymin = CI_80_lower,
                     ymax = CI_80_upper), size = 1.1) +
  geom_linerange(aes(ymin = CI_95_lower,
                     ymax = CI_95_upper)) +
  geom_linerange(aes(ymin = estimate - 0.1,
                     ymax = estimate + 0.1), size = 2) +
  scale_x_continuous(breaks = taxa_link_slope_abund$order,
                     labels = taxa_link_slope_abund$link) +
  scale_y_continuous(breaks = c(-5,0,5), labels = c("","",""),
                     sec.axis = sec_axis(~., breaks = c(-5,0,5), labels = c("Decreasing"," ","Increasin
  geom_text(aes(x = order, y = if_else(sign(estimate) == 1, CI_95_upper + 1, CI_95_lower - 1), label =
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "right", axis.text.x.top = element_text(size = 8)); col_2_slope_abund_est
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```



```
col_2_slope_est <- ggarrange(col_2_slope_rich_est, col_2_slope_abund_est, widths = c(1,1.3)); col_2_slop
```



Averaging estimates to the taxa level

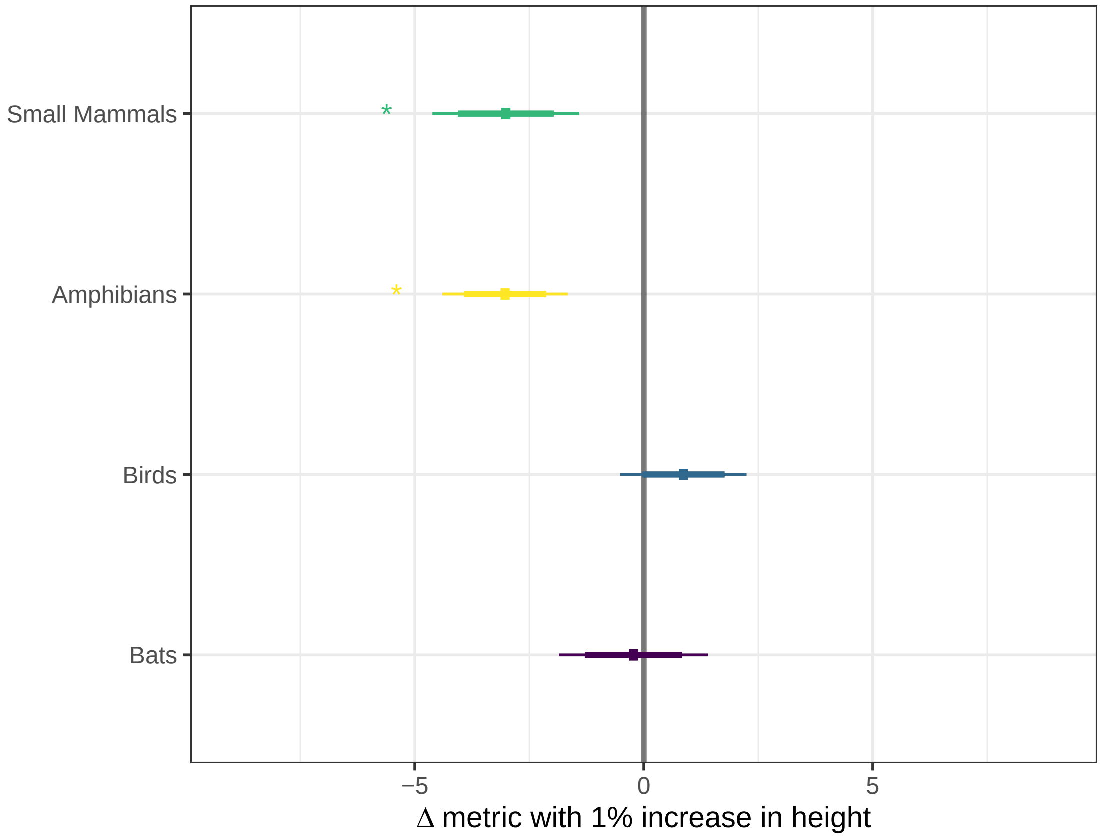
```
taxa_link_slope_avg <- taxa_link_slope %>%
  select(taxa_order, estimate, se, CI_95_lower, CI_95_upper, CI_80_lower, CI_80_upper, sig, metric) %>%
  group_by(metric, taxa_order) %>%
  summarise(estimate = mean(estimate),
          CI_95_upper = mean(CI_95_upper),
          CI_95_lower = mean(CI_95_lower),
          CI_80_upper = mean(CI_80_upper),
          CI_80_lower = mean(CI_80_lower)) %>%
  dplyr::mutate(sig = if_else(sign(CI_95_upper) == sign(CI_95_lower), "*", ""))
```

```
## `summarise()` regrouping output by 'metric' (override with `.groups` argument)
```

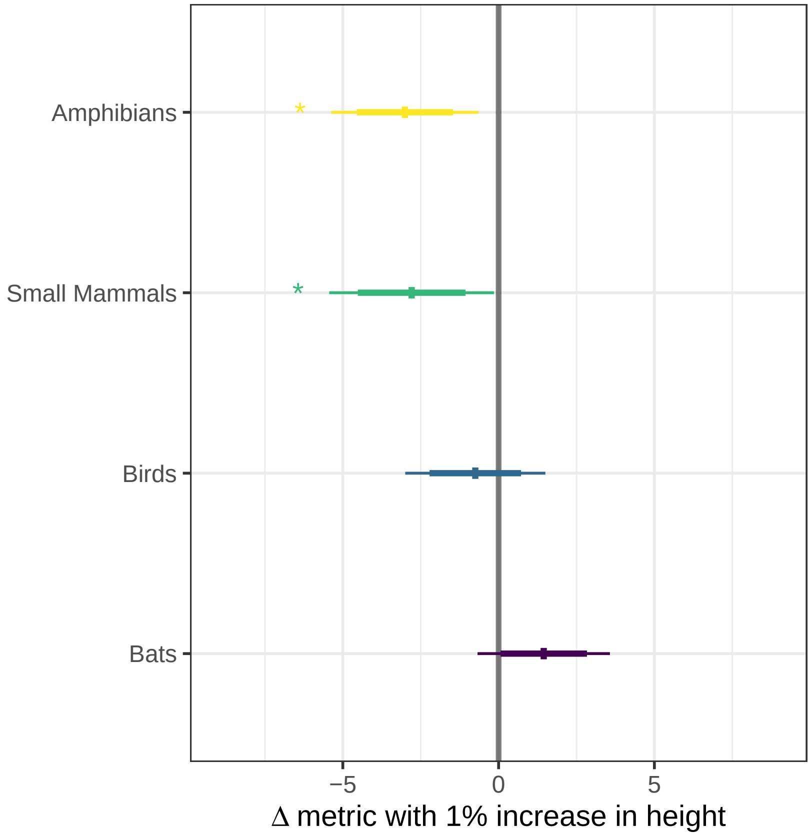
```
taxa_link_slope_rich_avg <- taxa_link_slope_avg %>%
  filter(metric == "richness") %>%
  rowid_to_column("order")

taxa_link_slope_abund_avg <- taxa_link_slope_avg %>%
  filter(metric == "abundance") %>%
  rowid_to_column("order")
```

Creating 4 panel plots of model estimates

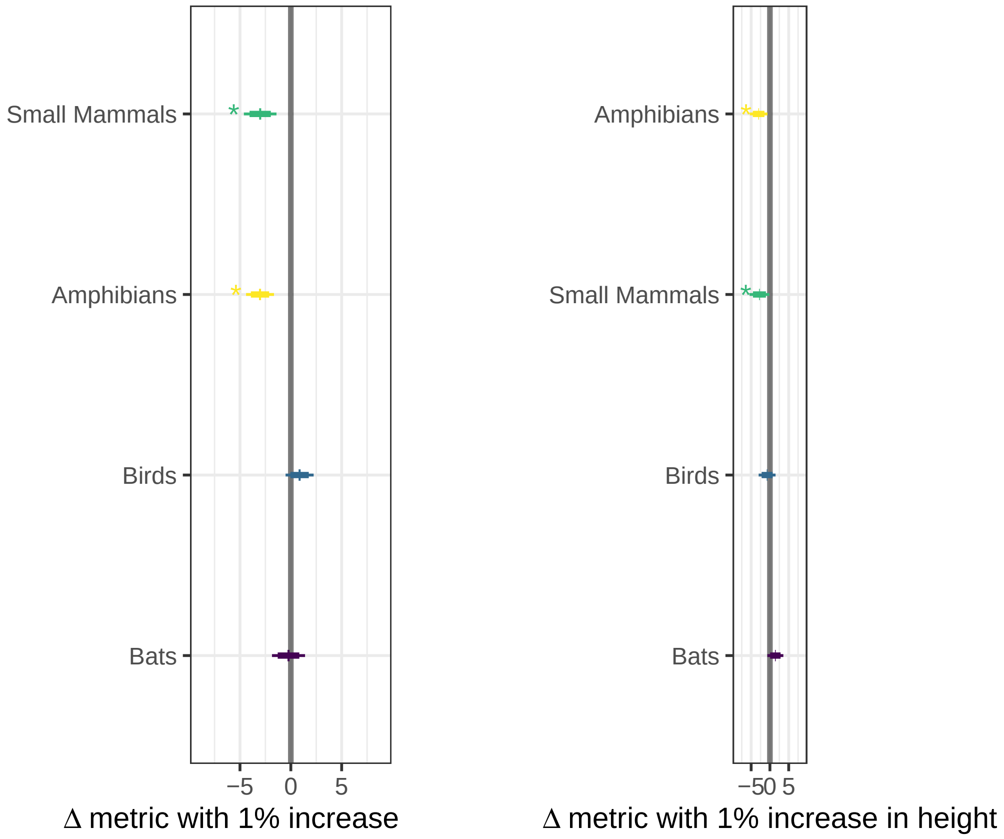
```
col_2_slope_rich_avg <- ggplot(taxa_link_slope_rich_avg, aes(x = rev(reorder(taxa_order, estimate)), y =
  ylab(expression(Delta ~ "metric with 1% increase in height")) + xlab(" ") +
  ylim(-9,9) + coord_flip() +
  geom_hline(yintercept = 0, alpha = 0.5, size = 1) +
  geom_linerange(aes(ymin = CI_80_lower,
                     ymax = CI_80_upper), size = 1.1) +
  geom_linerange(aes(ymin = CI_95_lower,
                     ymax = CI_95_upper)) +
  geom_linerange(aes(ymin = estimate - 0.1,
                     ymax = estimate + 0.1), size = 2) +
  geom_text(aes(x = rev(taxa_order), y = if_else(sign(estimate) == 1, CI_95_upper + 1, CI_95_lower - 1)
  scale_x_discrete(labels = rev(c("              Small Mammals", "Amphibians", "Birds", "Bats"))) +
  scale_color_viridis_d("Taxa") + theme_bw() +
  theme(legend.position = "none", axis.text.x.top = element_text(size = 2)); col_2_slope_rich_avg
```



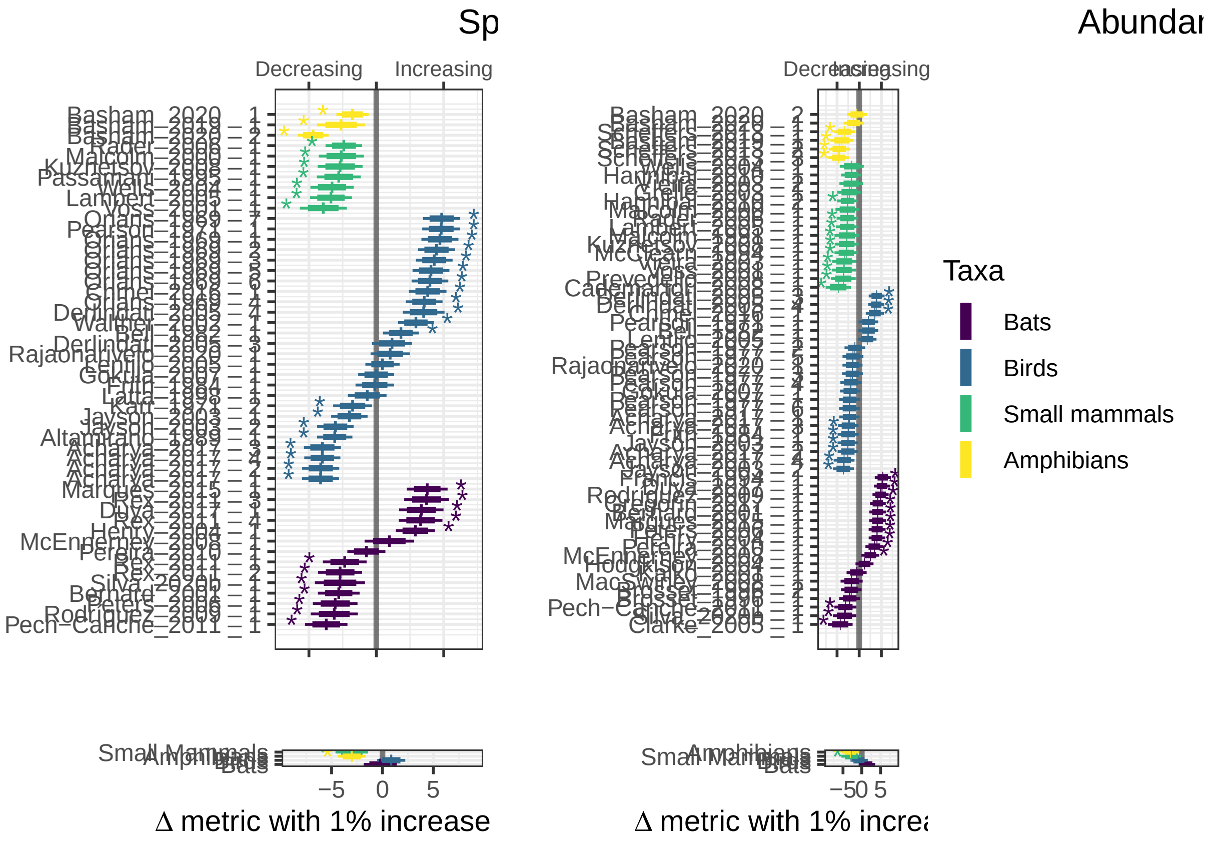
```
col_2_slope_abund_avg <- ggplot(taxa_link_slope_abund_avg, aes(x = rev(reorder(taxa_order, estimate)), y
  ylab(expression(Delta ~ "metric with 1% increase in height")) + xlab(" ") +
  ylim(-9,9) + coord_flip() +   geom_hline(yintercept = 0, alpha = 0.5, size = 1) +
  geom_linerange(aes(ymin = CI_80_lower,
                     ymax = CI_80_upper), size = 1.1) +
  geom_linerange(aes(ymin = CI_95_lower,
                     ymax = CI_95_upper)) +
  geom_linerange(aes(ymin = estimate - 0.1,
                     ymax = estimate + 0.1), size = 2) +
  geom_text(aes(x = rev(taxa_order), y = if_else(sign(estimate) == 1, CI_95_upper + 1, CI_95_lower - 1)
  scale_x_discrete(labels = c("Bats", "Birds", "             Small Mammals", "Amphibians")) +
  scale_color_viridis_d(" ") + theme_bw() +
  theme(legend.position = "right",
        axis.text.x.top = element_text(size = 2),
        legend.text=element_text(color="white")) + guides(color=guide_legend(override.aes=list(fill="wh
```



```
col_2_slope_avg <- ggarrange(col_2_slope_rich_avg, col_2_slope_abund_avg, widths = c(1,1.3)); col_2_slop
```



```
ggarrange(col_2_slope_est, col_2_slope_avg, nrow = 2, heights = c(6,1))
```



```
ggsave("analysis/figures/predictions_bio_parameters_link_taxa_4_panel_dodge.jpeg", width = 12, height =
```