tidymodels-m333

Grayson White

```
# Import data
spatial <- read_csv("../data/plot_level/plt_spatial.csv")
response <- read_csv("../data/plot_level/plot_response.csv")

# Join data
## Keep only observations in both ‘spatial‘ and ‘response‘
dat <- inner_join(spatial, response,
                  by = c("PLT_CN" = "PLT_CN",
                         "INVYR" = "INVYR"))

# Create columns for province, sections, and subsections
dat <- dat %>%
  mutate(
    subsection = ECOSUBCD.x,
    section = str_remove_all(ECOSUBCD.x, "[:lower:]"),
    province = str_sub(section, end = -2)
  )

# Select small subset of columns to work with for this EDA
dat_small <- dat %>%
  dplyr::select(PLT_CN, INVYR, PLOT.x, LON_PUBLIC.x, LAT_PUBLIC.x, LON_PUBLIC.y, LAT_PUBLIC.y,
         ELEV_PUBLIC.x, ELEV_PUBLIC.y, forgrp, forprob, nlcd11, demLF, evtLF, forbio,
         BALIVE_TPA, CNTLIVE_TPA, BIOLIVE_TPA, VOLNLIVE_TPA, subsection, section, province)

# Remove redundent columns, rename columns for ease of use
dat_small <- dat_small %>%
  dplyr::select(-LON_PUBLIC.y, -LAT_PUBLIC.y, -ELEV_PUBLIC.y) %>%
  rename(PLOT = PLOT.x,
         LON_PUBLIC = LON_PUBLIC.x,
         LAT_PUBLIC = LAT_PUBLIC.x,
         ELEV_PUBLIC = ELEV_PUBLIC.x)

north_rocky <- dat_small %>%
  filter(province == "M333")
```

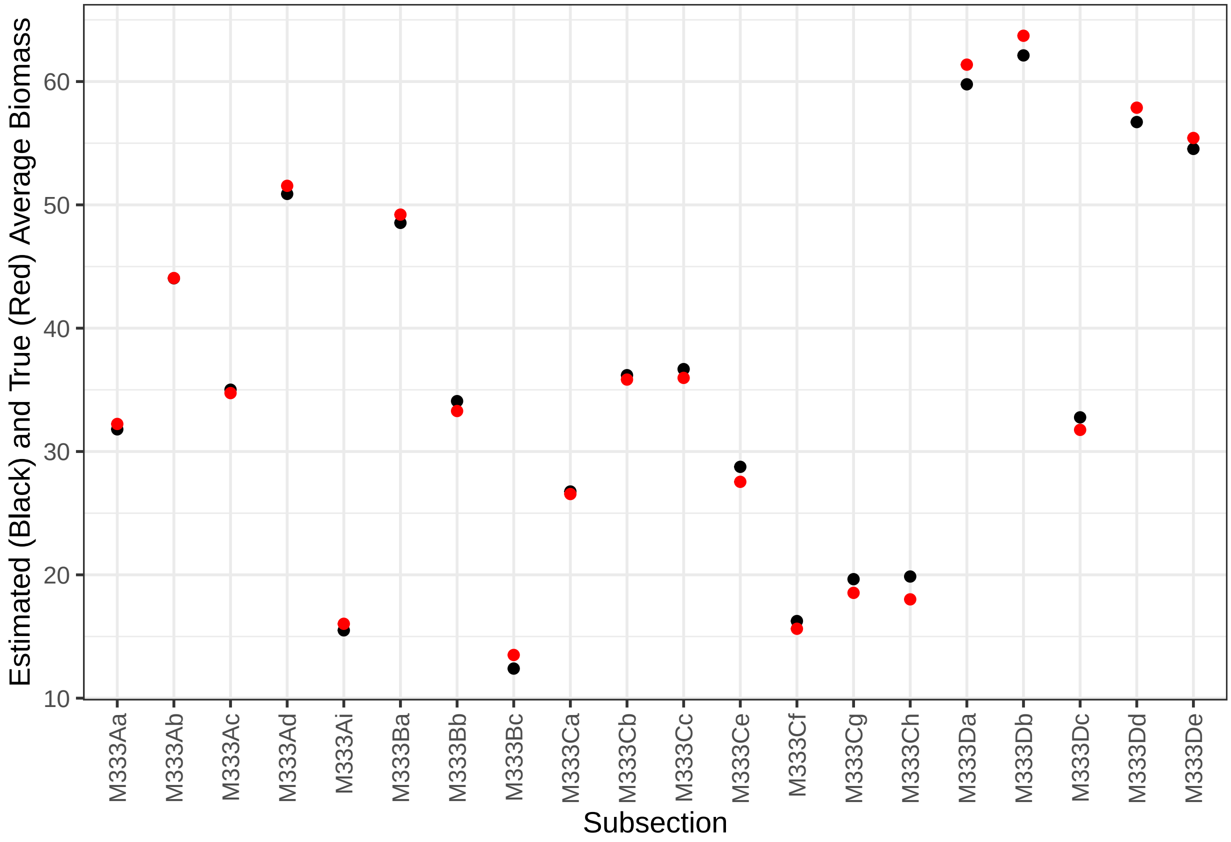
```
north_rocky_2 <- north_rocky %>%
  group_by(subsection) %>%
  summarize(
    mean_BIOLIVE_TPA = mean(BIOLIVE_TPA),
    var_BIOLIVE_TPA = var(BIOLIVE_TPA),
    mean_CNTLIVE_TPA = mean(CNTLIVE_TPA),
    var_CNTLIVE_TPA = var(CNTLIVE_TPA),
    mean_BALIVE_TPA = mean(BALIVE_TPA),
    mean_VOLNLIVE_TPA = mean(VOLNLIVE_TPA),
    mean_forprob = mean(forprob),
    var_forprob = var(forprob),
    n = n()
  )
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
set.seed(1)
hier_model_spec <- linear_reg() %>%
  set_engine("stan-glmer",
             prior_aux =  rstanarm::exponential(rate = 1),
             prior = NULL,
             prior_intercept = NULL,
             prior_covariance = rstanarm::decov(shape = 2))

hier_model_fit <-
  hier_model_spec %>%
  fit(mean_BIOLIVE_TPA ~ 1 +  mean_forprob + (1 | subsection),
      data = north_rocky_2)
```

```
## Warning: There were 24 divergent transitions after warmup. See
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quant
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
model_df_2 <- data.frame(fitted = hier_model_fit$fit$fitted.values,
          true = north_rocky_2$mean_BIOLIVE_TPA,
          subsection = north_rocky_2$subsection)

model_df_2 %>%
  ggplot(aes(x = subsection,
             y = fitted)) +
  geom_point() +
  geom_point(aes(y = true), color = "red") +
  theme_bw() +
  theme(axis.text.x = element_text(
    angle = 90,
    vjust = 0.5,
    hjust = 1
  )) +
  labs(
    x = "Subsection",
    y = "Estimated (Black) and True (Red) Average Biomass"
  )
```



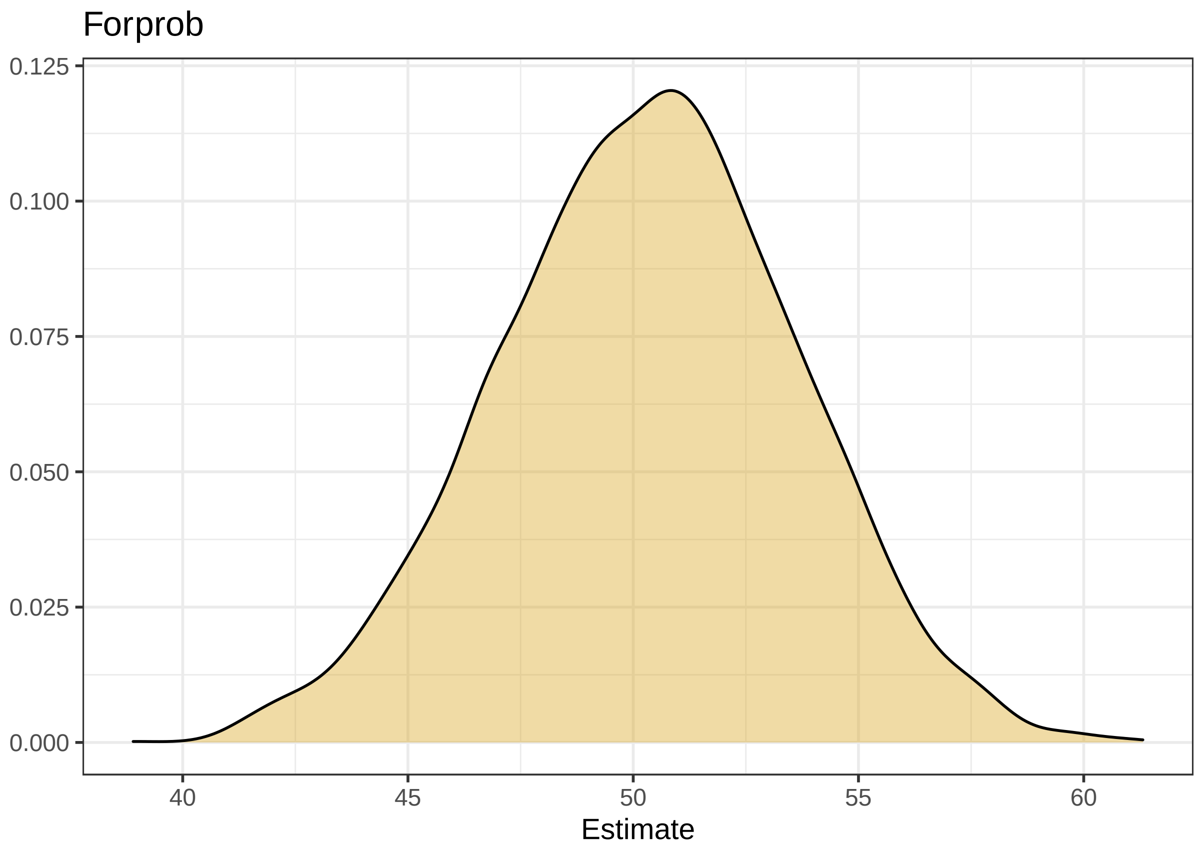
Fit a model

```
set.seed(1)
hier_model_spec <- linear_reg() %>%
  set_engine("stan-glmer",
             prior_aux =  rstanarm::exponential(rate = 1),
             prior = NULL,
             prior_intercept = NULL,
             prior_covariance = rstanarm::decov(shape = 2))

nr_train <- north_rocky %>%
  sample_frac(0.75)

nr_test <- north_rocky %>%
  anti_join(nr_train)
```

```
## Joining, by = c("PLT_CN", "INVYR", "PLOT", "LON_PUBLIC", "LAT_PUBLIC", "ELEV_PUBLIC", "forgrp", "for
```

```
hier_model_fit <-
  hier_model_spec %>%
  fit(BIOLIVE_TPA ~ 1 +  forprob + (1 | subsection),
      data = nr_train)
```

```
forprob_samples <- data.frame(
  samples = c(
    hier_model_fit$fit$stanfit@sim$samples[[1]]$‘beta[1]‘,
    hier_model_fit$fit$stanfit@sim$samples[[2]]$‘beta[1]‘,
    hier_model_fit$fit$stanfit@sim$samples[[3]]$‘beta[1]‘,
    hier_model_fit$fit$stanfit@sim$samples[[4]]$‘beta[1]‘
  )
)

ggplot(forprob_samples,
       aes(samples)) +
  scale_fill_manual(values = c("goldenrod")) +
  geom_density(aes(fill = "Posterior"), alpha = 0.4) +
  theme_bw() +
    theme(
    legend.position = "none"
  ) +
  labs(title = "Forprob",
       y = "",
       x = "Estimate")
```



```
results_test <- hier_model_fit %>%
  predict(new_data = nr_test) %>%
  mutate(
    truth = nr_test$BIOLIVE_TPA
  )

results_test %>%
  rmse(truth = truth, estimate = .pred)
```

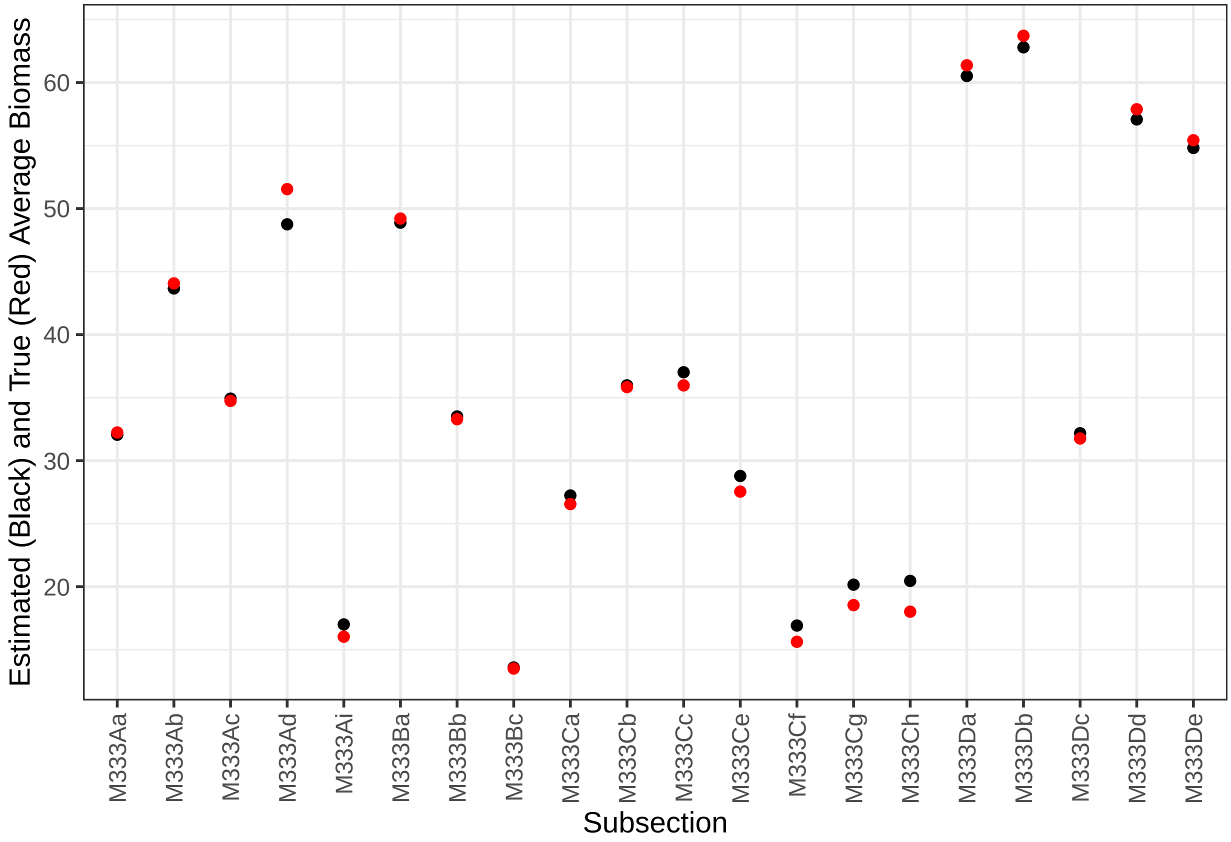
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        35.4
```

```
results_train <- hier_model_fit %>%
  predict(new_data = nr_train) %>%
  mutate(
    truth = nr_train$BIOLIVE_TPA
  )

results_train %>%
  rmse(truth = truth, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        35.3
```

```
nr_fit <-
  hier_model_spec %>%
  fit(BIOLIVE_TPA ~ 1 +  forprob + (1 | subsection),
      data = north_rocky)

model_df <- data.frame(fitted = nr_fit$fit$fitted.values,
           true = north_rocky$BIOLIVE_TPA,
           subsection = north_rocky$subsection)

model_df %>%
  group_by(subsection) %>%
  summarize(mean_fit = mean(fitted),
            mean_true = mean(true)) %>%
  ggplot(aes(x = subsection,
             y = mean_fit)) +
  geom_point() +
  geom_point(aes(y = mean_true), color = "red") +
  theme_bw() +
  theme(axis.text.x = element_text(
    angle = 90,
    vjust = 0.5,
    hjust = 1
  )) +
  labs(
    x = "Subsection",
    y = "Estimated (Black) and True (Red) Average Biomass"
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```


Estimated (Black) and True (Red) Average Biomass
60
50
40
30
20
M333Aa
M333Ab
M333Ac
M333Ad
M333Ai
M333Ba
M333Bb
M333Bc
M333Ca
M333Cb
M333Cc
M333Ce
M333Cf
M333Cg
M333Ch
M333Da
M333Db
M333Dc
M333Dd
M333De
Subsection