

modeling-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 redundant 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")
```

Fit a model

```
Xpop <- north_rocky %>%
  group_by(subsection) %>%
  summarize(mean = mean(BIOLIVE_TPA),
            count = n())
```

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

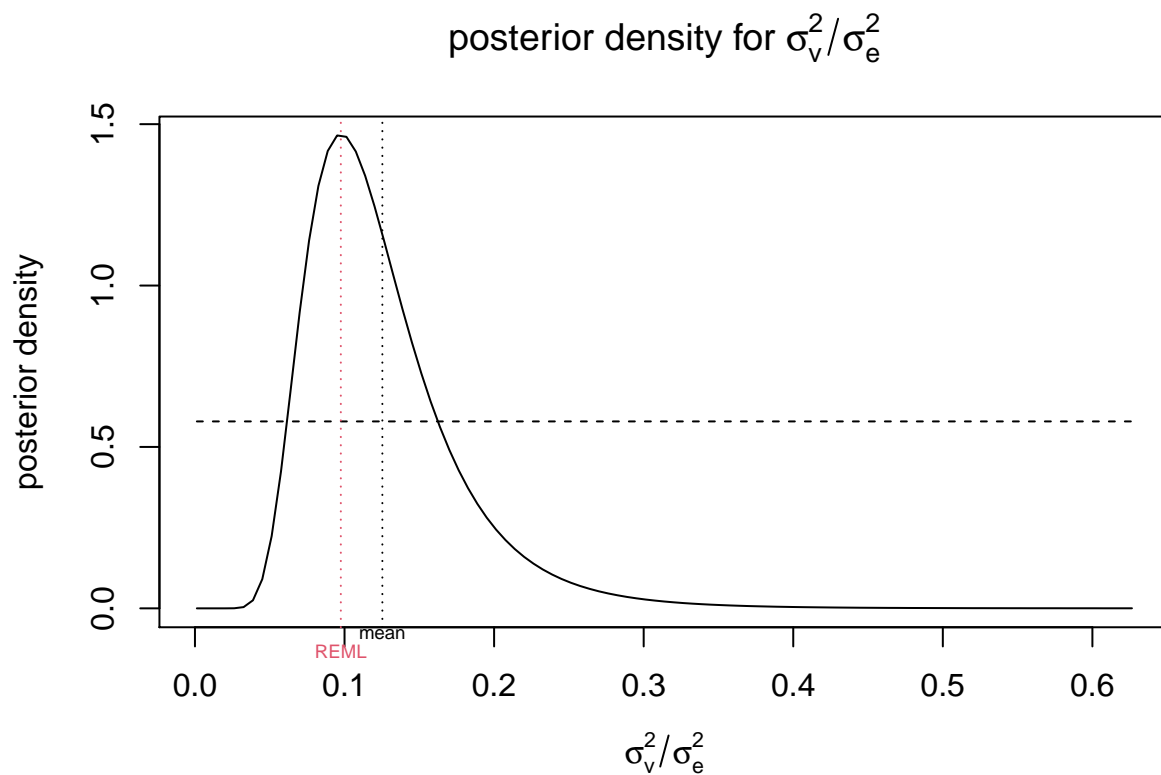
```
m1 <- fSAE.Unit(  
  y = north_rocky$BIOLIVE_TPA,  
  X = north_rocky$forprob,  
  Xpop = Xpop %>% dplyr::select(mean),  
  Narea = Xpop %>% dplyr::select(count) %>% pull(),  
  area = north_rocky$subsection,  
  method = "HB",  
  keep.data = TRUE  
)
```

```
## Warning in fSAE.Unit(y = north_rocky$BIOLIVE_TPA, X = north_rocky$forprob, : not  
## all sampled area names can be matched to row names of Xpop
```

```
## REML estimate of variance ratio: 0.09752
```

```
## numerical integration of f(x) (normalization constant): 1.158 with absolute error < 3.3e-09  
## numerical integration of x*f(x): 0.1451 with absolute error < 3.2e-10
```

```
## posterior mean for variance ratio: 0.1253
```

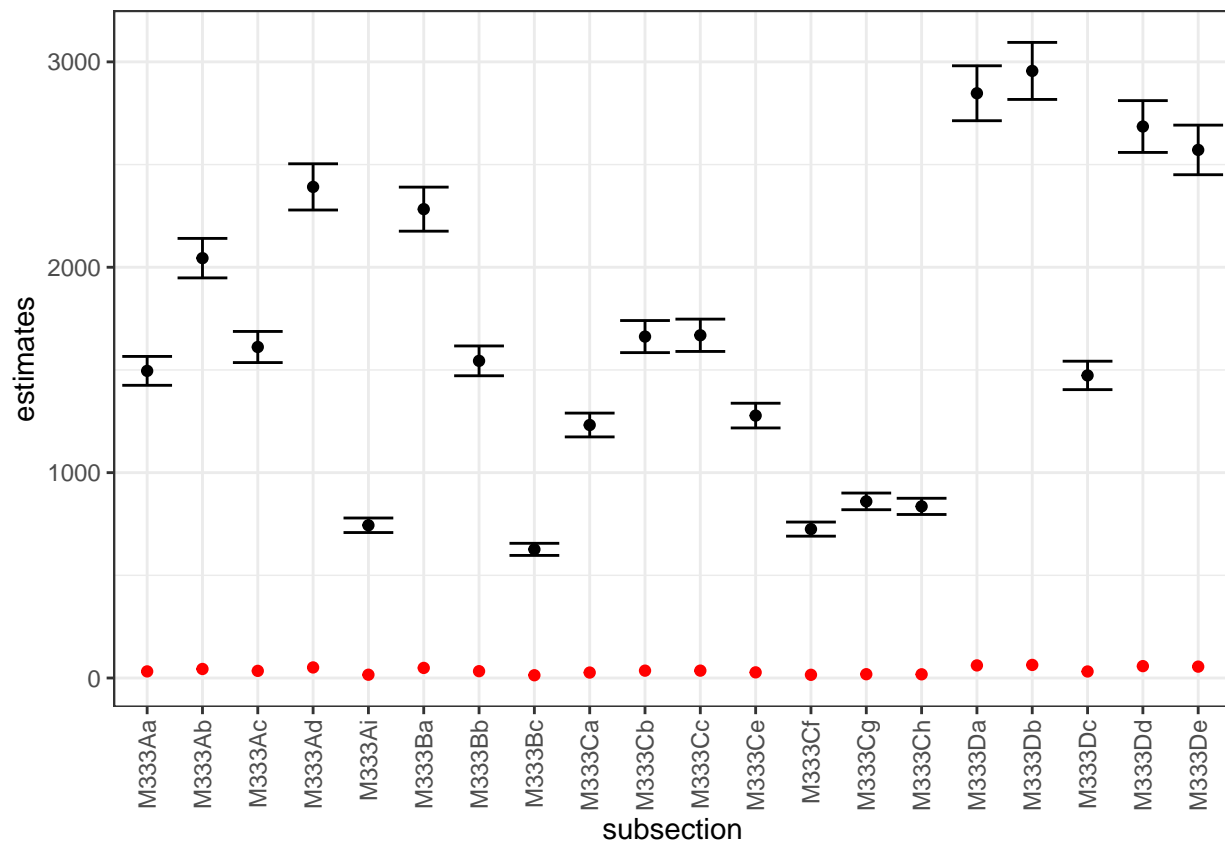


```

model_df <- tibble(
  estimates = EST(m1),
  standard_error = SE(m1),
  mean = Xpop$mean,
  subsection = Xpop$subsection
)

ggplot(model_df, aes(x = subsection,
                     y = estimates)) +
  geom_point() +
  geom_errorbar(aes(ymin = estimates - standard_error,
                   ymax = estimates + standard_error)) +
  geom_point(aes(y = mean), color = "red") +
  theme_bw() +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)
  )

```



```
plot(m1)
```