

area-level

```
dat_small <- read_csv("../data/subsets/dat_small.csv")

dat_area <- dat_small %>%
  group_by(subsection, section, province) %>%
  summarize(mean_BIOLIVE = mean(BIOLIVE_TPA),
            mean_BALIVE = mean(BALIVE_TPA),
            mean_CNTLIVE = mean(CNTLIVE_TPA),
            mean_VOLNLIVE = mean(VOLNLIVE_TPA),
            var_BIOLIVE = var(BIOLIVE_TPA),
            var_BALIVE = var(BALIVE_TPA),
            var_CNTLIVE = var(CNTLIVE_TPA),
            var_VOLNLIVE = var(VOLNLIVE_TPA),
            mean_nlcd11 = mean(nlcd11),
            var_nlcd11 = var(nlcd11))

m333_area <- dat_area %>%
  filter(province == "M333")

mod_area <- stan_lmer(mean_BIOLIVE ~ mean_nlcd11 + 1 + (1 | section),
                     data = m333_area)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000226 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.26 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 1: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 1: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 1: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 1: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 1: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 1: Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 1: Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 1: Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 1: Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 1: Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 1: Iteration:  2000 / 2000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 0.260743 seconds (Warm-up)
## Chain 1:                0.201088 seconds (Sampling)
## Chain 1:                0.461831 seconds (Total)
## Chain 1:
```

```

##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4.5e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.45 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 2: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 2: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 2: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 2: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 2: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 2: Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 2: Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 2: Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 2: Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 2: Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 2: Iteration:  2000 / 2000 [100%] (Sampling)
## Chain 2:
## Chain 2: Elapsed Time: 0.287803 seconds (Warm-up)
## Chain 2:                    0.179845 seconds (Sampling)
## Chain 2:                    0.467648 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.7e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.17 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 3: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 3: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 3: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 3: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 3: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 3: Iteration:  1001 / 2000 [ 50%] (Sampling)
## Chain 3: Iteration:  1200 / 2000 [ 60%] (Sampling)
## Chain 3: Iteration:  1400 / 2000 [ 70%] (Sampling)
## Chain 3: Iteration:  1600 / 2000 [ 80%] (Sampling)
## Chain 3: Iteration:  1800 / 2000 [ 90%] (Sampling)
## Chain 3: Iteration:  2000 / 2000 [100%] (Sampling)
## Chain 3:
## Chain 3: Elapsed Time: 0.270411 seconds (Warm-up)
## Chain 3:                    0.17359 seconds (Sampling)
## Chain 3:                    0.444001 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.5e-05 seconds

```

```
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.25 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%] (Warmup)
## Chain 4: Iteration:   200 / 2000 [ 10%] (Warmup)
## Chain 4: Iteration:   400 / 2000 [ 20%] (Warmup)
## Chain 4: Iteration:   600 / 2000 [ 30%] (Warmup)
## Chain 4: Iteration:   800 / 2000 [ 40%] (Warmup)
## Chain 4: Iteration:  1000 / 2000 [ 50%] (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 4:
## Chain 4: Elapsed Time: 0.274294 seconds (Warm-up)
## Chain 4:                0.196694 seconds (Sampling)
## Chain 4:                0.470988 seconds (Total)
## Chain 4:
```

```
mod_area_lm <- lm(mean_BIOLIVE ~ mean_nlcd11, data = m333_area)
```

```
mod_area_df <- data.frame(
  fitted = mod_area$fitted.values,
  section = m333_area %>% dplyr::select(section),
  true = mod_area$y
)
```

```
mod_area_lm_df <- data.frame(
  fitted = mod_area_lm$fitted.values,
  section = m333_area %>% dplyr::select(section),
  true = m333_area$mean_BIOLIVE
)
```

```
yardstick::rmse(mod_area_df, truth = true, estimate = fitted)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>       <dbl>
## 1 rmse    standard         4.16
```

```
yardstick::rmse(mod_area_lm_df, truth = true, estimate = fitted)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>       <dbl>
## 1 rmse    standard         7.41
```

```
MSE_bayes <- (rmse(mod_area_df, truth = true, estimate = fitted) %>% select(.estimate) %>% pull())^2
MSE_direct <- (m333_area %>%
  group_by(section) %>%
  summarize(mean = mean(mean_BIOLIVE)) %>%
  right_join(m333_area) %>%
  rmse(truth = mean_BIOLIVE,
        estimate = mean) %>%
  dplyr::select(.estimate) %>%
  pull())^2
```

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

```
## Joining, by = "section"
```

```
aparatant_ss <- (MSE_direct / MSE_bayes) * nrow(dat_area)
aparatant_ss
```

```
## [1] 3386.895
```

```
cv_bayes <- mod_area_df %>%
  group_by(section.section) %>%
  yardstick::rmse(truth = true, estimate = fitted) %>%
  select(section.section, .estimate) %>%
  right_join(mod_area_df) %>%
  group_by(section.section) %>%
  summarise(cv = .estimate / mean(fitted)) %>%
  unique()
```

```
## Joining, by = "section.section"
```

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

```
cv_lm <- mod_area_lm_df %>%
  group_by(section.section) %>%
  yardstick::rmse(truth = true, estimate = fitted) %>%
  select(section.section, .estimate) %>%
  right_join(mod_area_lm_df) %>%
  group_by(section.section) %>%
  summarise(cv = .estimate / mean(fitted)) %>%
  unique()
```

```
## Joining, by = "section.section"
```

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

```
left_join(cv_bayes, cv_lm, by = "section.section") %>%
  rename(cv_bayes = cv.x,
        cv_lm = cv.y) %>%
  pivot_longer(cols = c("cv_bayes", "cv_lm")) %>%
  ggplot(aes(x = section.section,
            y = value,
```

```

    color = name)) +
  geom_point() +
  scale_color_manual(values = c("goldenrod", "forestgreen")) +
  theme_bw()

```

