area-level

```r
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 0.000113 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.13 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: 0.276262 seconds (Warm-up)
## Chain 1:                0.20257 seconds (Sampling)
## Chain 1:                0.478832 seconds (Total)
## Chain 1:
```

```
##
## SAMPLING FOR MODEL ’continuous’ NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2.2e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.22 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: 0.284404 seconds (Warm-up)
## Chain 2:                0.244013 seconds (Sampling)
## Chain 2:                0.528417 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL ’continuous’ NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 5.6e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.56 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: 0.255445 seconds (Warm-up)
## Chain 3:                0.172168 seconds (Sampling)
## Chain 3:                0.427613 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL ’continuous’ NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.15 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: 0.30609 seconds (Warm-up)
## Chain 4:                0.167525 seconds (Sampling)
## Chain 4:                0.473615 seconds (Total)
## Chain 4:
```

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

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

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        4.15
```

```r
MSE_bayes <- (rmse(mod_area_df, truth = true, estimate = fitted) %>% select(.estimate) %>% pull())^2
MSE_direct <- (m333_area %>% # I am not sure if this is how the calc the direct MSE
  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"
```

```r
aparant_ss <- (MSE_direct / MSE_bayes) * nrow(m333_area)
aparant_ss
```

```
## [1] 141.7971
```

```
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()
#
# 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()
```

```
# lets try to vary over the province
```