8 Mixed Effects Models and Small Area Estimation

Immanuel Klein

```r
library("tidyverse")
library("ggplot2")
```

This task is about estimating the total field size of corn and soybeans across various counties in Iowa by using both linear and mixed-effects models. We first fit a linear model for both corn and soybeans by county, using the number of pixels as the predictor for hectares of crops. We then extended this to a linear mixed-effects model, using county-level random effects for the shared variability within each county. With this extended model, we compare different predictors such as BLUP (Best Linear Unbiased Predictor) and survey-based estimates in their accuracy and reliability. Finally, we plot the total estimated field size of corn and soybeans on a map together.

Exercise (a)

```r
library(JoSAE)
library(nlme)

data(landsat)

corn.model <-
  lmList(HACorn ~ PixelsCorn | CountyName, data = landsat)

soybeans.model <-
  lmList(HASoybeans ~ PixelsSoybeans | CountyName, data = landsat)

summary(corn.model)
```

```
Call:
  Model: HACorn ~ PixelsCorn | CountyName
   Data: landsat

Coefficients:
   (Intercept)
                 Estimate Std. Error      t value  Pr(>|t|)
Cerro Gordo  165.76000000        NaN          NaN       NaN
Hamilton      96.32000000        NaN          NaN       NaN
Worth         76.08000000        NaN          NaN       NaN
Humboldt    -272.70292308        NaN          NaN       NaN
Franklin     115.56683286   97.95973  1.179738221 0.2592433
Pocahontas    -8.78651636   34.49348 -0.254729755 0.8029170
Winnebago      0.08145147   57.37266  0.001419691 0.9988888
Wright       -59.96862032   49.58024 -1.209526608 0.2479975
Webster        5.48252687   63.33110  0.086569268 0.9323331
Hancock       28.54476316   53.62203  0.532332727 0.6034755
Kossuth       50.48087468   49.24900  1.025013094 0.3240529
Hardin        16.39881070   32.59514  0.503106020 0.6233055
   PixelsCorn
             Estimate Std. Error   t value    Pr(>|t|)
Cerro Gordo        NA         NA        NA          NA
Hamilton           NA         NA        NA          NA
Worth              NA         NA        NA          NA
Humboldt    1.0603077        NaN       NaN         NaN
Franklin    0.1268856  0.2870322 0.4420604 0.665710295
Pocahontas  0.5006440  0.1478366 3.3864685 0.004867249
Winnebago   0.3872573  0.1938520 1.9976959 0.067116660
Wright      0.5802991  0.1376822 4.2147730 0.001011343
Webster     0.4258783  0.2380999 1.7886542 0.096985870
Hancock     0.2846382  0.1866418 1.5250503 0.151200141
Kossuth     0.1904752  0.1548051 1.2304194 0.240340223
Hardin      0.3446977  0.1111900 3.1000784 0.008445494

Residual standard error: 18.11868 on 13 degrees of freedom
```

```
summary(soybeans.model)
```

```
Call:
  Model: HASoybeans ~ PixelsSoybeans | CountyName
   Data: landsat
```

```
Coefficients:
   (Intercept)
              Estimate Std. Error     t value  Pr(>|t|)
Cerro Gordo   8.0900000        NaN         NaN       NaN
Hamilton    106.0300000        NaN         NaN       NaN
Worth       103.6000000        NaN         NaN       NaN
Humboldt    -60.6714634        NaN         NaN       NaN
Franklin    -19.8303128   49.56224 -0.40010931 0.6955730
Pocahontas  -62.4479223   38.78831 -1.60996771 0.1314083
Winnebago    81.1814777   41.26111  1.96750604 0.0708352
Wright        0.9008238   21.07498  0.04274375 0.9665554
Webster     -25.6522301   52.04278 -0.49290664 0.6302993
Hancock      51.0786087   31.82859  1.60480264 0.1325449
Kossuth      13.5769597   43.14219  0.31470263 0.7579814
Hardin        1.3200740   16.18004  0.08158655 0.9362182
   PixelsSoybeans
             Estimate Std. Error   t value     Pr(>|t|)
Cerro Gordo          NA         NA        NA           NA
Hamilton             NA         NA        NA           NA
Worth                NA         NA        NA           NA
Humboldt    0.69939024        NaN       NaN          NaN
Franklin    0.42699004 0.28867669 1.4791289 1.629254e-01
Pocahontas  0.69939996 0.14639390 4.7775210 3.611176e-04
Winnebago   0.04629555 0.25329139 0.1827758 8.577935e-01
Wright      0.52662596 0.10555893 4.9889284 2.477553e-04
Webster     0.54365580 0.20218984 2.6888384 1.858523e-02
Hancock     0.28694638 0.13479506 2.1287604 5.296895e-02
Kossuth     0.53856942 0.22042556 2.4433165 2.958209e-02
Hardin      0.41953325 0.07167233 5.8534900 5.652694e-05

Residual standard error: 14.16772 on 13 degrees of freedom
```

- It only makes sense to presume that `PixelsCorn` can be predicted with `HACorn` while `PixelsSoybeans` can be predicted with `HASoybeans`. The assumption is that the number of pixels corresponding to each crop in the satellite images is directly related to the area covered by that crop.
- Nonetheless, there might be some limitations:
  - The model is fitted to each county separately, which might lead to overfitting. The model may not generalize well to other counties that are not included in the dataset.
  - The model assumes that the observations are independent. However, if there is some correlation between segments within the same county, this assumption might

be violated.

– There may be other factors that influence the hectares of crops (e. g., weather conditions such as clouds) that are not included in the model.

Exercise (b)

```
corn.model.random <- lme(
  # Fixed effects, same assumption as in (a)
  HACorn ~ PixelsCorn,
  # Random effect that is shared for each county
  random = ~ 1 | CountyName,
  data = landsat
  )

# Fit the mixed-effects model for soybeans
soybeans.model.random <- lme(
  # Fixed effects, same assumption as in (a)
  HASoybeans ~ PixelsSoybeans,
  # Random effect that is shared for each county
  random = ~ 1 | CountyName,
  data = landsat
  )

summary(corn.model.random)
```

```
Linear mixed-effects model fit by REML
  Data: landsat
       AIC      BIC    logLik
  326.6529 332.8743 -159.3264

Random effects:
 Formula: ~1 | CountyName
        (Intercept) Residual
StdDev:    7.926246 17.03993

Fixed effects:  HACorn ~ PixelsCorn
               Value Std.Error DF  t-value p-value
(Intercept) 5.466189 13.543455 24 0.403604  0.6901
PixelsCorn  0.387836  0.043575 24 8.900464  0.0000
 Correlation:
```

```
           (Intr)
PixelsCorn -0.961

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max 
-2.8145038 -0.5576048  0.1739121  0.6701859  1.5489571 

Number of Observations: 37
Number of Groups: 12 
```

```r
summary(soybeans.model.random)
```

```
Linear mixed-effects model fit by REML
  Data: landsat 
       AIC      BIC    logLik
  321.0191 327.2405 -156.5095

Random effects:
 Formula: ~1 | CountyName
        (Intercept) Residual
StdDev:    15.46753 13.41709

Fixed effects:  HASoybeans ~ PixelsSoybeans 
                   Value Std.Error DF   t-value p-value
(Intercept)    -3.822356  9.325208 24 -0.409895  0.6855
PixelsSoybeans  0.475678  0.039701 24 11.981527  0.0000
 Correlation: 
               (Intr)
PixelsSoybeans -0.835

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max 
-1.8087062 -0.5328769 -0.1997715  0.4419333  1.8973958 

Number of Observations: 37
Number of Groups: 12 
```

- Both crops have significant pixel coefficients, which indicates that there is a significant relationship between the number of pixels and hectares for both corn and soybeans.
- The intercepts are not significant in both models, which seems to be common in contexts where the predictors (like pixels) capture most of the variability. Thus, the intercept gets less interpretable.

- Lower AIC and BIC values in the soybeans model indicate a slightly better fit compared to the corn model. Pixel data might be a more reliable predictor of soybean hectares than corn hectares, maybe because of differences in growth patterns and such.

Exercise (c)

```
# Variance of error term and random effect
sigma.v2 <- as.numeric(VarCorr(corn.model.random)["(Intercept)", "Variance"])
sigma.e2 <- as.numeric(corn.model.random$sigma^2)

# Covariance matrix of beta.hat
V.beta.hat <- vcov(corn.model.random)

# Extract slope per county from second model
beta.hat.corn <- data.frame(
  CountyName = rownames(coef(corn.model.random)[2]),
  PredictorValue = coef(corn.model.random)[2]$PixelsCorn
)

results.corn <- landsat %>%
  group_by(CountyName) %>%
  summarize(
    xip.mean = first(MeanPixelsCorn),
    xi.mean = mean(PixelsCorn, na.rm = TRUE),
    yi.mean = mean(HACorn, na.rm = TRUE),
    n = n()
  ) %>%
  left_join(beta.hat.corn, by = "CountyName") %>%
  mutate(
    gamma.i = sigma.v2 / (sigma.v2 + (sigma.e2 / n)),
    RP.Corn = xip.mean * PredictorValue,
    ASP.Corn = xip.mean * PredictorValue +
      (yi.mean - xi.mean * PredictorValue),
    BLUP.Corn = xip.mean * PredictorValue +
      gamma.i * (yi.mean - xi.mean * PredictorValue),
    SP.Corn = yi.mean,
    # MSE for Regression predictor for corn
    MSE.RP.Corn = (1 - 0)^2 * sigma.v2 + (0^2 * sigma.e2) / n +
              2 * (0 - gamma.i) *
              (xip.mean - 0 * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - 0 * xi.mean)^2 * V.beta.hat[2, 2],
```

```r
    # MSE for Adjusted Survey Predictor for corn
    MSE.ASP.Corn = (1 - 1)^2 * sigma.v2 + (1^2 * sigma.e2) / n +
              2 * (1 - gamma.i) *
              (xip.mean - 1 * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - 1 * xi.mean)^2 * V.beta.hat[2, 2],
    # MSE for BLUP for corn
    MSE.BLUP.Corn = (1 - gamma.i)^2 * sigma.v2 + (gamma.i^2 * sigma.e2) / n +
              2 * (gamma.i - gamma.i) *
              (xip.mean - gamma.i * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - gamma.i * xi.mean)^2 * V.beta.hat[2, 2]
  ) %>%
  select(CountyName,
         RP.Corn,
         MSE.RP.Corn,
         ASP.Corn,
         MSE.ASP.Corn,
         BLUP.Corn,
         MSE.BLUP.Corn,
         SP.Corn)

results.corn
```

```
# A tibble: 12 x 8
   CountyName  RP.Corn MSE.RP.Corn ASP.Corn MSE.ASP.Corn BLUP.Corn MSE.BLUP.Corn
   <chr>         <dbl>       <dbl>    <dbl>        <dbl>     <dbl>         <dbl>
 1 Cerro Gordo   115.        169.     135.         321.      118.          151.
 2 Hamilton      117.        173.     132.         332.      119.          183.
 3 Worth         112.        165.      90.3        297.      108.          165.
 4 Humboldt      113.        126.     109.         199.      112.           98.8
 5 Franklin      123.        104.     150.          98.7     134.          103.
 6 Pocahontas     99.7        89.5    116.         102.      106.           92.7
 7 Winnebago     113.         97.2    113.          96.8     113.           97.7
 8 Wright        117.         99.5    125.         108.      120.           88.4
 9 Webster       102.         72.2    117.          72.6     109.           70.9
10 Hancock       122.         55.5    121.          61.5     121.           82.9
11 Kossuth       116.         56.2    104.          58.9     110.           65.1
12 Hardin        126.         36.6    131.          54.2     129.           78.9
# i 1 more variable: SP.Corn <dbl>
```

```r
# Do same thing for soy
# Variance of error term and random effect
sigma.v2 <- as.numeric(VarCorr(soybeans.model.random)["(Intercept)", "Variance"])
```

```r
sigma.e2 <- as.numeric(soybeans.model.random$sigma^2)

# Covariance matrix of beta.hat
V.beta.hat <- vcov(soybeans.model.random)

beta.hat.soybeans <- data.frame(
  CountyName = rownames(coef(soybeans.model.random)[2]),
  PredictorValue = coef(soybeans.model.random)[2]$PixelsSoybeans
)

results.soybeans <- landsat %>%
  group_by(CountyName) %>%
  summarize(
    xip.mean = first(MeanPixelsSoybeans),
    xi.mean = mean(PixelsSoybeans, na.rm = TRUE),
    yi.mean = mean(HASoybeans, na.rm = TRUE),
    n = n()
  ) %>%
  left_join(beta.hat.soybeans, by = "CountyName") %>%
  mutate(
    gamma.i = sigma.v2 / (sigma.v2 + (sigma.e2 / n)),
    RP.Soybeans = xip.mean * PredictorValue,
    ASP.Soybeans = xip.mean * PredictorValue +
      (yi.mean - xi.mean * PredictorValue),
    BLUP.Soybeans = xip.mean * PredictorValue +
      gamma.i * (yi.mean - xi.mean * PredictorValue),
    SP.Soybeans = yi.mean,
    # MSE for Regression predictor for soy
    MSE.RP.Soybeans = (1 - 0)^2 * sigma.v2 + (0^2 * sigma.e2) / n +
              2 * (0 - gamma.i) *
              (xip.mean - 0 * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - 0 * xi.mean)^2 * V.beta.hat[2, 2],
    # MSE for Adjusted Survey Predictor for soy
    MSE.ASP.Soybeans = (1 - 1)^2 * sigma.v2 + (1^2 * sigma.e2) / n +
              2 * (1 - gamma.i) *
              (xip.mean - 1 * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - 1 * xi.mean)^2 * V.beta.hat[2, 2],
    # MSE for BLUP for soy
    MSE.BLUP.Soybeans = (1 - gamma.i)^2 * sigma.v2 + (gamma.i^2 * sigma.e2) / n +
              2 * (gamma.i - gamma.i) *
              (xip.mean - gamma.i * xi.mean)^2 * V.beta.hat[2, 2] +
              (xip.mean - gamma.i * xi.mean)^2 * V.beta.hat[2, 2]
```

```r
  ) %>%
  select(CountyName,
         RP.Soybeans,
         MSE.RP.Soybeans,
         ASP.Soybeans,
         MSE.ASP.Soybeans,
         BLUP.Soybeans,
         MSE.BLUP.Soybeans,
         SP.Soybeans)

results.soybeans
```

```
# A tibble: 12 x 8
   CountyName  RP.Soybeans MSE.RP.Soybeans ASP.Soybeans MSE.ASP.Soybeans
   <chr>             <dbl>           <dbl>        <dbl>            <dbl>
 1 Cerro Gordo        90.2            231.         72.2            233. 
 2 Hamilton           93.5            231.         95.9            181. 
 3 Worth              97.6            230.         82.3            186. 
 4 Humboldt          105.             205.         74.7            107. 
 5 Franklin           89.5            206.         61.4             60.8
 6 Pocahontas        118.             182.        113.              60.3
 7 Winnebago          88.2            207.        101.              61.5
 8 Wright            105.             193.        116.              63.1
 9 Webster           118.             173.        109.              45.1
10 Hancock            94.5            193.        102.              38.1
11 Kossuth            97.3            191.        123.              36.2
12 Hardin             84.2            201.         73.7             32.2
# i 3 more variables: BLUP.Soybeans <dbl>, MSE.BLUP.Soybeans <dbl>,
#   SP.Soybeans <dbl>
```

- The MSE for the RP is relatively high compared to the other predictors, this simple model just may not be as reliable. This is because it does not adjust for the difference between the observed and predicted county-level means.
- The MSE for ASP is still high but lower than MSE.RP.
- The MSE for BLUP is consistently lower than both MSE.RP and MSE.ASP, showing that BLUP provides the most reliable estimates with the smallest prediction error, which makes sense. This highlights the advantage of incorporating both fixed and random effects.
- The SP values vary widely, showing differences in the actual observed data between counties as it directly reflects the county-level data without any adjustments.

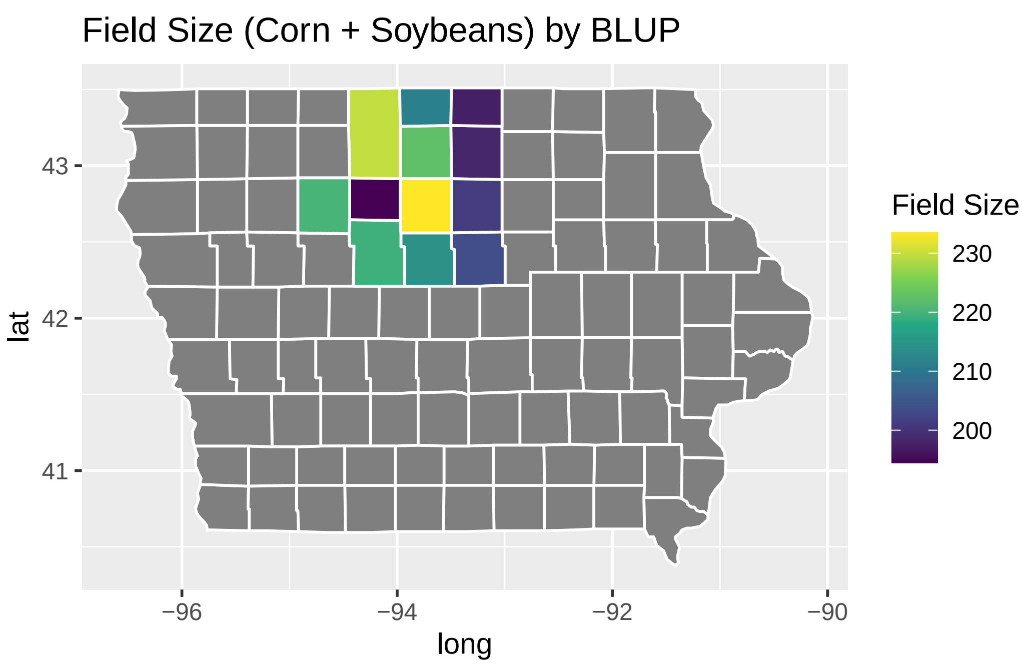
Exercise (d)

```
library(maps)
library(mapdata)

# Calc total county field size by joining the tables and adding BLUP and SP
total.field.size <- results.corn %>%
  full_join(
    results.soybeans,
    by = "CountyName"
  ) %>%
  mutate(
    Total.BLUP = BLUP.Corn + BLUP.Soybeans,
    Total.SP = SP.Corn + SP.Soybeans
  ) %>%
  select(CountyName, Total.BLUP, Total.SP)
total.field.size
```

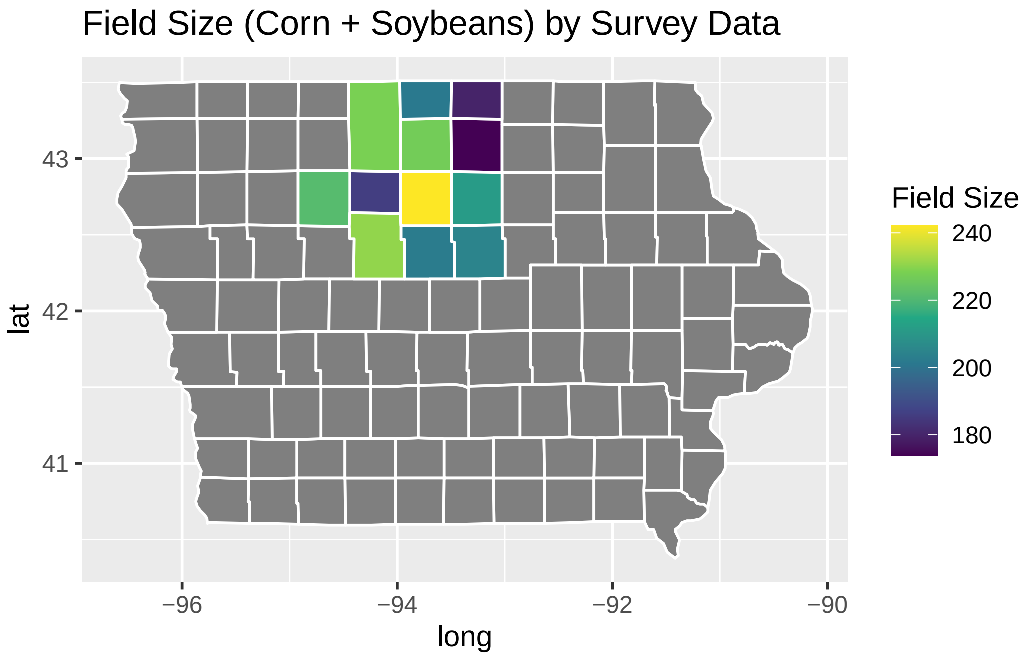
```
# A tibble: 12 x 3
   CountyName  Total.BLUP Total.SP
   <chr>            <dbl>    <dbl>
 1 Cerro Gordo       198.     174.
 2 Hamilton          214.     202.
 3 Worth             197.     180.
 4 Humboldt          194.     186.
 5 Franklin          201.     211.
 6 Pocahontas        220.     221.
 7 Winnebago         211.     201.
 8 Wright            233.     242.
 9 Webster           219.     231.
10 Hancock           222.     227.
11 Kossuth           230.     228.
12 Hardin            204.     205.
```

```
# Some preparation for plotting
iowa.counties <- map_data("county", region = "iowa")
iowa.counties$CountyName <- tools::toTitleCase(iowa.counties$subregion)
plot.data <- iowa.counties %>%
  left_join(total.field.size, by = "CountyName")

# Plot BLUP estimates
```

```r
ggplot(plot.data, aes(long, lat, group = group)) +
  geom_polygon(aes(fill = Total.BLUP), color = "white") +
  scale_fill_viridis_c() +
  labs(
    title = "Field Size (Corn + Soybeans) by BLUP",
    fill = "Field Size"
  )
```




```r
# Plot SP estimates
ggplot(plot.data, aes(long, lat, group = group)) +
  geom_polygon(aes(fill = Total.SP), color = "white") +
  scale_fill_viridis_c() +
  labs(
    title = "Field Size (Corn + Soybeans) by Survey Data",
    fill = "Field Size"
  )
```



- The BLUP estimates for total field size per county seem to be a bit more conservative than the SP estimates, though they are quite close in value. This is expected as BLUP adjusts for both fixed effects and random effects, so it shrinks estimates towards the overall mean. The SP estimates directly reflect the survey data without any adjustments. These estimates are purely data-driven, so it makes sense that they have a higher variability.
- Both maps show that Wright County has the highest total field size estimate, with the SP map showing an even higher estimate than the BLUP map. Both the counties Humboldt and Franklin have relatively high estimates as well, with SP showing slightly higher values than BLUP, particularly in the Franklin County. While BLUP seems to moderate the estimates, the general pattern of field sizes across the counties is mostly consistent: The larger fields are concentrated in a few key counties.