Piedmont Triad Simulation Parameters

Sarah Lotspeich

2024-02-11

```r
# Load packages
library(ggplot2) ## for maps and other plots
library(tidycensus) ## for shapefiles
library(possum) ## for imputation
library(spdep) ## for adjacency matrix
library(spaMM) ## for spatial mixed-effects model
```

Load Data

```r
# Load data
## Just the Piedmont Triad
food_access = read.csv(file = "https://raw.githubusercontent.com/sarahlotspeich/food_access_imputation/r

## Inspect merged dataset (used for analysis)
food_access |>
  summary()
```

```
##    LocationID          CountyName            Xstar              X_full
##  Min.   :3.700e+10   Length:387          Min.   : 0.03008   Min.   : 0.03477
##  1st Qu.:3.707e+10   Class :character    1st Qu.: 0.56767   1st Qu.: 0.88860
##  Median :3.708e+10   Mode  :character    Median : 1.00683   Median : 1.51435
##  Mean   :3.709e+10                       Mean   : 1.75278   Mean   : 2.46340
##  3rd Qu.:3.708e+10                       3rd Qu.: 2.11709   3rd Qu.: 2.99173
##  Max.   :3.720e+10                       Max.   :10.79727   Max.   :15.76243
##
##    Xstar_time          Xstar_q20          X_time             O_POP
##  Min.   :0.0000000   Min.   : 8.36    Min.   : 8.808    Min.   :1103
##  1st Qu.:0.0000000   1st Qu.:11.00    1st Qu.:10.947    1st Qu.:3091
##  Median :0.0000000   Median :13.37    Median :11.255    Median :4095
##  Mean   :0.0002687   Mean   :17.11    Mean   :11.879    Mean   :4240
##  3rd Qu.:0.0010000   3rd Qu.:22.59    3rd Qu.:11.963    3rd Qu.:5282
##  Max.   :0.0020000   Max.   :47.62    Max.   :28.179    Max.   :9324
##
##    Y_BPHIGH          Y_CHD           Y_DIABETES        Y_OBESITY
##  Min.   : 402    Min.   : 68.0    Min.   : 105.0    Min.   : 425
##  1st Qu.:1058    1st Qu.:200.5    1st Qu.: 325.5    1st Qu.:1038
##  Median :1441    Median :274.0    Median : 451.0    Median :1431
##  Mean   :1474    Mean   :286.0    Mean   : 470.8    Mean   :1482
##  3rd Qu.:1782    3rd Qu.:350.0    3rd Qu.: 588.5    3rd Qu.:1854
```

1

```
## Max.  :3347  Max.  :702.0  Max.  :1226.0  Max.  :3290
##
##    X_partial
## Min.  : 0.2821
## 1st Qu.: 1.2363
## Median : 2.2992
## Mean  : 3.6500
## 3rd Qu.: 4.7797
## Max.  :15.7624
## NA's  :339
```

- Sample size for the Piedmont Triad: 387

- Median population: 4095 people per census tract

```
## Summarize prevalence of diabetes
summary(food_access$Y_DIABETES/food_access$O_POP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02600 0.09202 0.10719 0.11298 0.12902 0.23541
```

- Median prevalence of diabetes: 0.11

```
## Summarize prevalence of obesity
summary(food_access$Y_OBESITY/food_access$O_POP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2201  0.3141  0.3422  0.3508  0.3800  0.5211
```

- Median prevalence of obesity: 0.34

```
# All of NC (with population > 0)
nonzero_pop = read.csv("https://raw.githubusercontent.com/sarahlotspeich/food_access_imputation/refs/hea
```

- Sample size for all of North Carolina: 2169

Additive and Multiplicative Errors

```
food_access |>
  ggplot(aes(x = X_full, y = Xstar)) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_abline(intercept = 0, slope = 1, linetype = 2) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```
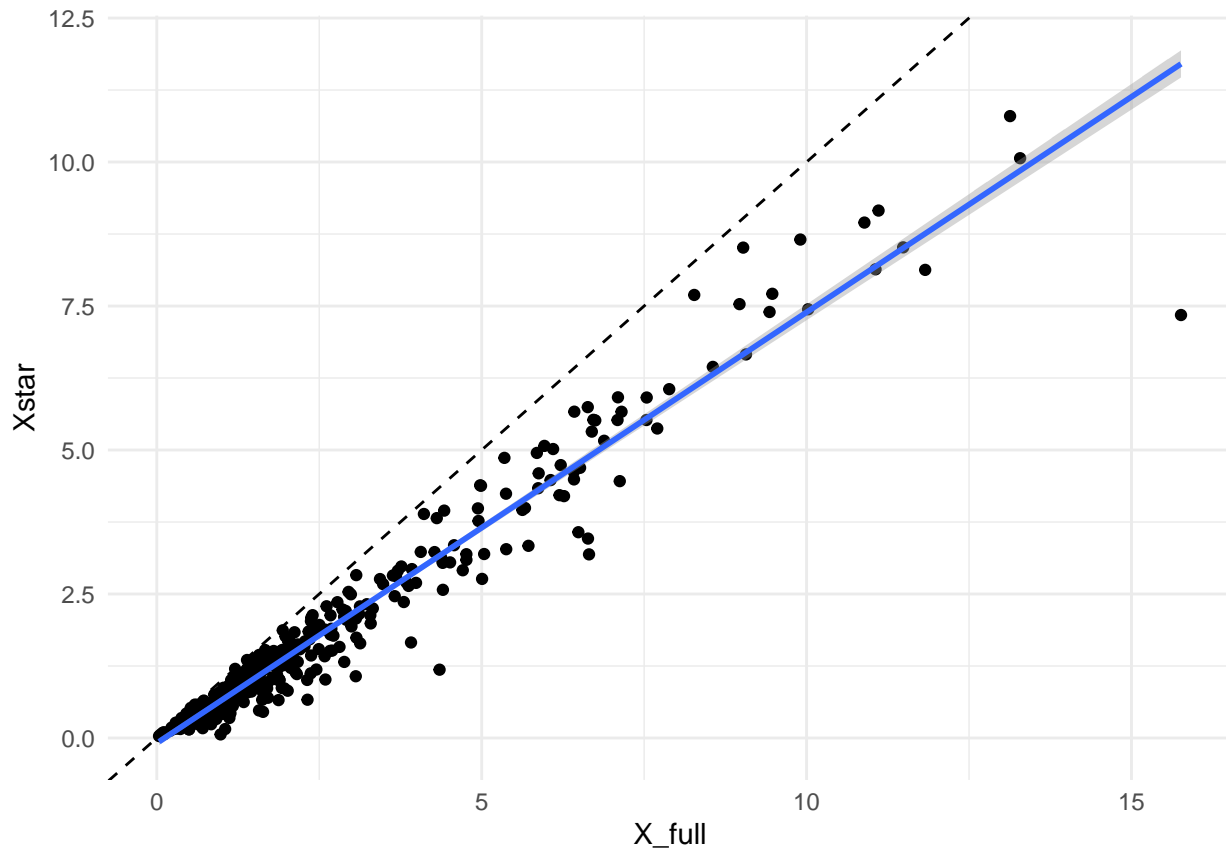
```
## Add "error" between straight-line and map-based distances
food_access = food_access |>
  dplyr::mutate(U = Xstar - X_full,
                W = Xstar / X_full,
                LocationID = as.character(LocationID))

## Summarize distribution of errors
food_access |>
  dplyr::select(U, W) |>
  summary()
```

```
##        U                 W
##  Min.   :-8.4189   Min.   :0.06338
##  1st Qu.:-0.9421   1st Qu.:0.60594
##  Median :-0.4705   Median :0.70998
##  Mean   :-0.7106   Mean   :0.69147
##  3rd Qu.:-0.2290   3rd Qu.:0.79128
##  Max.   : 0.0000   Max.   :1.00000
```

```
## Standard deviation of errors
food_access |>
  dplyr::summarize(sdU = sd(U),
                   sdW = sd(W))
```

```
##         sdU       sdW
## 1 0.7716128 0.1531462
```

```
fitU = lm(formula = U ~ 1, data = food_access)
(params_list = list(mean = fitU$coefficients,
                    sd = sigma(fitU)))
```

```
## $mean
## (Intercept)
##  -0.7106214
##
## $sd
## [1] 0.7716128
```
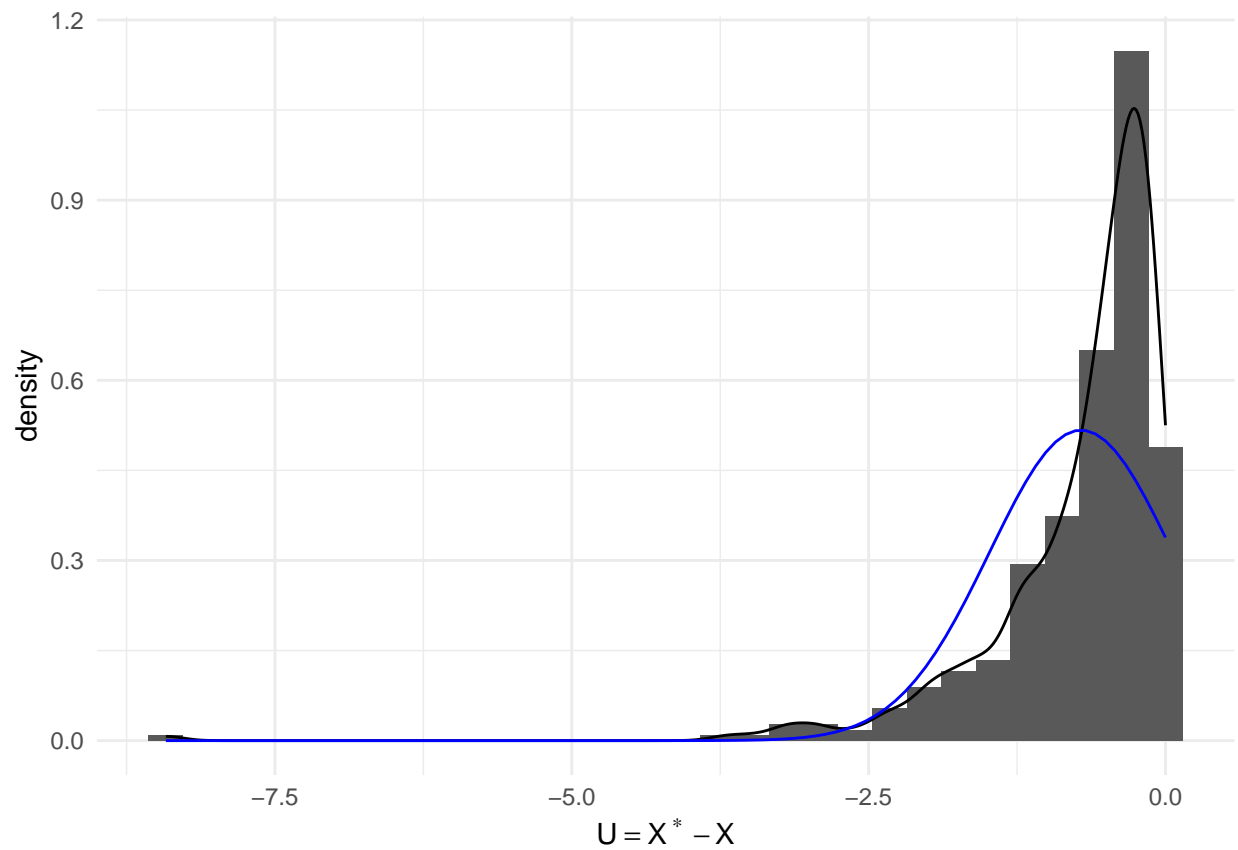

Figure 1: Histogram of additive errors

```
fitW = lm(formula = W ~ 1, data = food_access)
(params_list = list(mean = fitW$coefficients,
                    sd = sigma(fitW)))
```

```
## $mean
## (Intercept)
##   0.6914693
##
## $sd
## [1] 0.1531462
```
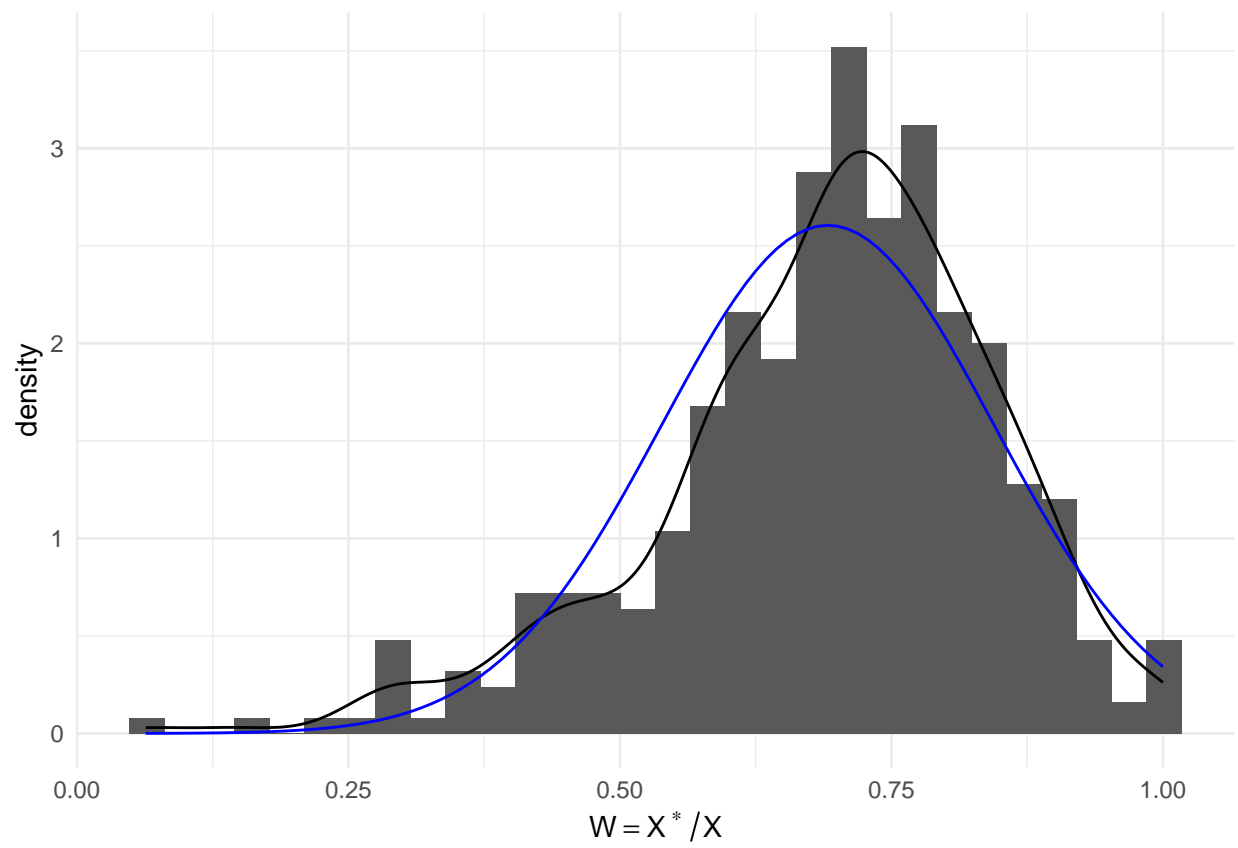
Figure 2: Histogram of multiplicative errors

Models

Gold Standard Non-Spatial Analysis

```r
# Model 1b: Diagnosed diabetes among adults aged >=18 years
## Predictor X = proximity to healthy foods based on map-based distance
mod_diab = glm(formula = Y_DIABETES ~ X_full,
               family = poisson(link = "log"),
               offset = log(O_POP),
               data = food_access)
summary(mod_diab)
```

```
##
## Call:
## glm(formula = Y_DIABETES ~ X_full, family = poisson(link = "log"),
##     data = food_access, offset = log(O_POP))
##
## Coefficients:
##               Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -2.1777770  0.0033311 -653.779   <2e-16 ***
## X_full      -0.0078792  0.0009407   -8.376   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 12936  on 386  degrees of freedom
## Residual deviance: 12865  on 385  degrees of freedom
## AIC: 15926
##
## Number of Fisher Scoring iterations: 4
```

```r
# Model 2b: Obesity among adults aged >=18 years
## Predictor X = proximity to healthy foods based on map-based distance
mod_obes = glm(formula = Y_OBESITY ~ X_full,
               family = poisson(link = "log"),
               offset = log(O_POP),
               data = food_access)
summary(mod_obes)
```

```
##
## Call:
## glm(formula = Y_OBESITY ~ X_full, family = poisson(link = "log"),
##     data = food_access, offset = log(O_POP))
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.0348461  0.0018776 -551.17   <2e-16 ***
## X_full      -0.0064159  0.0005283  -12.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 10931  on 386  degrees of freedom
## Residual deviance: 10782  on 385  degrees of freedom
## AIC: 14292
## 
## Number of Fisher Scoring iterations: 4
```

Gold Standard Spatial Analysis

```
## Source a script to build adjacency matrix for census tracts in the Piedmont Triad as
### --> adj_matrix
devtools::source_url("https://raw.githubusercontent.com/sarahlotspeich/food_access_imputation/refs/head
```

```
## i SHA-1 hash of file is "0fe28a5b8f506ff670caa57e80b0065e3c9b2482"
## Getting data from the 2011-2015 5-year ACS
## 
## Downloading feature geometry from the Census website.  To cache shapefiles for use in future session
```

```
##   |                                                                 |
```

```
# Model 1a: Diagnosed diabetes among adults aged >=18 years
## Predictor X = proximity to healthy foods based on map-based distance
mod_diab = fitme(formula = Y_DIABETES ~ X_full + adjacency(1 | LocationID) + offset(log(O_POP)),
                 family = poisson(link = "log"),
                 adjMatrix = adj_matrix,
                 data = food_access)
summary(mod_diab)
```

```
## formula: Y_DIABETES ~ X_full + adjacency(1 | LocationID) + offset(log(O_POP))
## Estimation of corrPars and lambda by ML (p_v approximation of logL).
## Estimation of fixed effects by ML (p_v approximation of logL).
## Estimation of lambda by 'outer' ML, maximizing logL.
## family: poisson( link = log )
##   ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE  t-value
## (Intercept) -2.217018  0.02233 -99.2883
## X_full      -0.002572  0.00554  -0.4643
##   -------------- Random effects ---------------
## Family: gaussian( link = identity )
##                 --- Correlation parameters:
##      1.rho
## 0.06212734
##             --- Variance parameters ('lambda'):
```

```
## lambda = var(u) for u ~ Gaussian;
##     LocationID  :   0.07374
## # of obs: 387; # of groups: LocationID, 387
##   ------------- Likelihood values  -------------
##                          logLik
## logL       (p_v(h)): -2400.749
```

```
# Model 2a: Obesity among adults aged >=18 years
## Predictor X = proximity to healthy foods based on map-based distance
mod_obes = fitme(formula = Y_OBESITY ~ X_full + adjacency(1 | LocationID) + offset(log(O_POP)),
                 family = poisson(link = "log"),
                 adjMatrix = adj_matrix,
                 data = food_access)
summary(mod_obes)
```

```
## formula: Y_OBESITY ~ X_full + adjacency(1 | LocationID) + offset(log(O_POP))
## Estimation of corrPars and lambda by ML (p_v approximation of logL).
## Estimation of fixed effects by ML (p_v approximation of logL).
## Estimation of lambda by 'outer' ML, maximizing logL.
## family: poisson( link = log )
##   ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)  -1.0469 0.011692 -89.535
## X_full       -0.0056 0.002822  -1.985
##   -------------- Random effects --------------
## Family: gaussian( link = identity )
##               --- Correlation parameters:
##      1.rho
## 0.07119065
##           --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##     LocationID  :   0.019
## # of obs: 387; # of groups: LocationID, 387
##   ------------- Likelihood values  -------------
##                          logLik
## logL       (p_v(h)): -2589.258
```