Submodels and final models

Aidan Brushett

2025-04-30

Before you begin

This script is number 6 of 6 in a series of scripts used to replicate the analyses presented in the paper: "Life on the edge: Industrial footprint and edge effects variably affect the distribution of a boreal small mammal"

This script was used to fit models to simulated data and get a sense of the predictive accuracy of our red squirrel models.

When running these scripts, please ensure that you have downloaded the complete GitHub repository. This will ensure you have all the files, data, and proper folder structure you will need to run this code and associated analyses.

Also make sure you open RStudio through the R project (OSM_red_squirrel_distribution.Rproj). This will automatically set your working directory to the correct place (wherever you saved the repository) and ensure you don't have to change the file paths for some of the data. This analysis was initially run in R v4.3.0. If you have any questions or concerns, please contact one of the authors (in order):

Aidan Brushett M.Sc. Student University of Victoria
School of Environmental Studies
Email: aidanbrushett@uvic.ca

Emerald Arthurs M.Sc. Student University of Victoria
School of Environmental Studies

---

0. Setup

```
rm(list = ls())
#library(MASS)
library(glmmTMB)
#library(lme4)
library(tidyverse)
library(MuMIn)
#library(PerformanceAnalytics)
library(ggpubr)
```

1. Re-construct the final dataset (best spatial scales)

1.1. Reimport the data

```
covs <- read_csv("./data/processed/OSM_all_covariates_HFI_SBFI_final.csv")
```

```
## Rows: 9460 Columns: 55
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): array, site
## dbl (53): array_year, lat, long, easting_12n, northing_12n, buffer_dist, cfi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
response <- read_csv("./data/processed/OSM_monthly_detections_2021_2022_2023.csv") %>%

  # Only species we want is red squirrel
  filter(species == "red squirrel") %>%

  # Only want detections column
  select(-species, -presence) %>%

  rename(squirrel = detections)
```

```
## Rows: 63934 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): array, site, species
## dbl (4): month, year, presence, detections
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Add the covariates to the response variable
data <- response %>%

  left_join(covs, by = c("array", "site"))
```

```
## Warning in left_join(., covs, by = c("array", "site")): Detected an unexpected many-to-many relations
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 1 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
# Make sure there are 20 rows per site/month/year
data %>%

  group_by(site, month, year) %>%
```

```
  summarize(n_obs = n()) %>%

  arrange(n_obs)
```

```
## `summarise()` has grouped output by 'site', 'month'. You can override using the
## `.groups` argument.

## # A tibble: 4,918 x 4
## # Groups:   site, month [4,754]
##    site    month  year n_obs
##    <chr>   <dbl> <dbl> <int>
##  1 LU13_11     1  2023    22
##  2 LU13_11     2  2023    22
##  3 LU13_11     3  2023    22
##  4 LU13_11     4  2023    22
##  5 LU13_11     5  2023    22
##  6 LU13_11     6  2023    22
##  7 LU13_11     7  2023    22
##  8 LU13_11     8  2023    22
##  9 LU13_11     9  2022    22
## 10 LU13_11     9  2023    22
## # i 4,908 more rows
```

```
# 20 for everything. Looks good!!

# z-scaling for variables WITHIN each buffer
data_scaled <- data %>%

  group_by(buffer_dist) %>%

  mutate(across(cfi_site:last_col(), ~ as.numeric(scale(.)))) %>%

  ungroup(.)

# The mean will be 0 even though we grouped first, since the mean for each buffer is still 0.
summary(data_scaled)
```

```
##     array               site               month             year     
##  Length:108196      Length:108196      Min.   : 1.000   Min.   :2021  
##  Class :character   Class :character   1st Qu.: 4.000   1st Qu.:2022  
##  Mode  :character   Mode  :character   Median : 7.000   Median :2023  
##                                        Mean   : 6.634   Mean   :2023  
##                                        3rd Qu.:10.000   3rd Qu.:2024  
##                                        Max.   :12.000   Max.   :2024  
##                                                                        
##     squirrel         array_year        lat             long       
##  Min.   : 0.0000   Min.   :2021   Min.   :54.56   Min.   :-115.0  
##  1st Qu.: 0.0000   1st Qu.:2022   1st Qu.:55.55   1st Qu.:-113.0  
##  Median : 0.0000   Median :2022   Median :56.56   Median :-111.8  
##  Mean   : 0.9339   Mean   :2022   Mean   :56.30   Mean   :-112.2  
##  3rd Qu.: 1.0000   3rd Qu.:2023   3rd Qu.:57.10   3rd Qu.:-111.2  
##  Max.   :70.0000   Max.   :2023   Max.   :57.56   Max.   :-110.1  
```

```
##
##   easting_12n      northing_12n      buffer_dist      cfi_site
##  Min.   :254793   Min.   :6046183   Min.   :  50   Min.   :-1.0421
##  1st Qu.:375199   1st Qu.:6155882   1st Qu.:1000   1st Qu.:-0.6582
##  Median :448084   Median :6268110   Median :2375   Median :-0.3196
##  Mean   :427832   Mean   :6241258   Mean   :2393   Mean   : 0.0000
##  3rd Qu.:487051   3rd Qu.:6330343   3rd Qu.:3750   3rd Qu.: 0.2842
##  Max.   :559033   Max.   :6379755   Max.   :5000   Max.   : 6.7755
##
##  cfi_site_with_harvest cfi_site_with_vegedges  harvest_0_15
##  Min.   :-1.1835       Min.   :-1.1245        Min.   :-0.4446
##  1st Qu.:-0.6693       1st Qu.:-0.6442        1st Qu.:-0.3841
##  Median :-0.3013       Median :-0.3122        Median :-0.3240
##  Mean   : 0.0000       Mean   : 0.0000        Mean   : 0.0000
##  3rd Qu.: 0.3302       3rd Qu.: 0.2777        3rd Qu.:-0.2203
##  Max.   : 6.7241       Max.   : 6.7198        Max.   : 8.8550
##
##  harvest_gt_15     harvest_total      osm_industrial       pipe_trans
##  Min.   :-0.6256   Min.   :-0.67528   Min.   :-0.50992   Min.   :-0.9652
##  1st Qu.:-0.4893   1st Qu.:-0.56852   1st Qu.:-0.41059   1st Qu.:-0.6878
##  Median :-0.3716   Median :-0.41880   Median :-0.30491   Median :-0.3105
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.1280   3rd Qu.:-0.03056   3rd Qu.:-0.04273   3rd Qu.: 0.2548
##  Max.   : 6.9091   Max.   : 5.08535   Max.   :10.57756   Max.   : 7.4013
##
##     railways          roads            seismic        seismic_lines
##  Min.   :-0.111   Min.   :-1.3575   Min.   :-1.0230   Min.   :-1.5413
##  1st Qu.:-0.091   1st Qu.:-0.5403   1st Qu.:-0.6049   1st Qu.:-0.6608
##  Median :-0.052   Median :-0.1591   Median :-0.3247   Median :-0.2403
##  Mean   : 0.000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.038   3rd Qu.: 0.4028   3rd Qu.: 0.2055   3rd Qu.: 0.4071
##  Max.   :26.485   Max.   :15.0263   Max.   : 5.7997   Max.   : 5.0866
##  NA's   :4918
##  seismic_lines_3D      trails          veg_edges        wells_active
##  Min.   :-0.4103   Min.   :-1.0576   Min.   :-0.8885   Min.   :-0.56774
##  1st Qu.:-0.3846   1st Qu.:-0.6221   1st Qu.:-0.4756   1st Qu.:-0.47773
##  Median :-0.3565   Median :-0.2290   Median :-0.1844   Median :-0.38959
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.:-0.2986   3rd Qu.: 0.2535   3rd Qu.: 0.1645   3rd Qu.:-0.01219
##  Max.   : 6.2937   Max.   :15.8529   Max.   :13.8053   Max.   :19.32623
##
##  wells_inactive     wells_total       pct_betu_pap     fire_0_15
##  Min.   :-0.9289   Min.   :-0.7841   Min.   :-0.09   Min.   :-0.3601
##  1st Qu.:-0.6250   1st Qu.:-0.5801   1st Qu.:-0.07   1st Qu.:-0.3444
##  Median :-0.3368   Median :-0.3392   Median :-0.07   Median :-0.3258
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00   Mean   : 0.0000
##  3rd Qu.: 0.1762   3rd Qu.: 0.1184   3rd Qu.:-0.05   3rd Qu.:-0.3000
##  Max.   :13.2921   Max.   : 8.9950   Max.   :20.22   Max.   : 4.5335
##                                      NA's   :59016
##    fire_gt_15       pct_lari_lar      lc_broadleaf     lc_coniferous
##  Min.   :-0.3240   Min.   :-0.6475   Min.   :-1.0151   Min.   :-2.37235
##  1st Qu.:-0.2827   1st Qu.:-0.5490   1st Qu.:-0.7086   1st Qu.:-0.70739
##  Median :-0.2498   Median :-0.3749   Median :-0.4357   Median :-0.07207
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
```

```
##  3rd Qu.:-0.2232   3rd Qu.: 0.1673   3rd Qu.: 0.4674   3rd Qu.: 0.61317
##  Max.   : 7.4090   Max.   : 9.4549   Max.   : 4.4843   Max.   : 2.74250
##
##    lc_herbs        lc_mixedwood       lc_shrubs          lc_water
##  Min.   :-0.7852   Min.   :-1.3024   Min.   :-0.3114   Min.   :-0.5413
##  1st Qu.:-0.6572   1st Qu.:-0.6833   1st Qu.:-0.2894   1st Qu.:-0.3722
##  Median :-0.3921   Median :-0.3053   Median :-0.2693   Median :-0.2429
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2136   3rd Qu.: 0.4097   3rd Qu.:-0.2334   3rd Qu.:-0.1295
##  Max.   : 6.7377   Max.   : 5.8728   Max.   : 8.9180   Max.   :13.1921
##
##   lc_wetland       lc_wetland_treed   pct_pice_gla      pct_pice_mar
##  Min.   :-1.0140   Min.   :-1.7479   Min.   :-0.4873   Min.   :-2.1399
##  1st Qu.:-0.6385   1st Qu.:-0.7748   1st Qu.:-0.4199   1st Qu.:-0.9071
##  Median :-0.4023   Median :-0.1654   Median :-0.3262   Median : 0.1614
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2744   3rd Qu.: 0.6188   3rd Qu.:-0.1272   3rd Qu.: 0.8697
##  Max.   : 6.0380   Max.   : 3.4576   Max.   : 9.5250   Max.   : 1.7085
##
##   pct_pinu_ban      pct_popu_tre     nonanthro_cai_mn   nonanthro_ed
##  Min.   :-0.3977   Min.   :-1.6340   Min.   :-1.6760   Min.   :-1.1470
##  1st Qu.:-0.3679   1st Qu.:-0.8626   1st Qu.:-0.7096   1st Qu.:-0.5623
##  Median :-0.3372   Median :-0.1685   Median :-0.1475   Median :-0.2731
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.1929   3rd Qu.: 0.8477   3rd Qu.: 0.6428   3rd Qu.: 0.2039
##  Max.   : 5.0205   Max.   : 2.5358   Max.   : 3.5209   Max.   : 5.1082
##
##  nonanthro_tca     landscape_cai_mn  landscape_ed    landscape_tca
##  Min.   :-3.1665   Min.   : NA      Min.   : NA      Min.   : NA
##  1st Qu.:-0.7096   1st Qu.: NA      1st Qu.: NA      1st Qu.: NA
##  Median : 0.1140   Median : NA      Median : NA      Median : NA
##  Mean   : 0.0000   Mean   :NaN      Mean   :NaN      Mean   :NaN
##  3rd Qu.: 0.8431   3rd Qu.: NA      3rd Qu.: NA      3rd Qu.: NA
##  Max.   : 1.6937   Max.   : NA      Max.   : NA      Max.   : NA
##                    NA's   :108196   NA's   :108196   NA's   :108196
##  nonanthro_cohesion  landscape_cohesion landscape_contag   landscape_mesh
##  Min.   :-17.73901   Min.   :-3.90056   Min.   :-3.07601   Min.   :-1.8860
##  1st Qu.: -0.09964   1st Qu.:-0.69566   1st Qu.:-0.72484   1st Qu.:-0.6640
##  Median :  0.26597   Median :-0.03058   Median :-0.08274   Median :-0.2884
##  Mean   :  0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:  0.50409   3rd Qu.: 0.73933   3rd Qu.: 0.64601   3rd Qu.: 0.3927
##  Max.   :  0.88374   Max.   : 2.95922   Max.   : 4.08776   Max.   : 7.5132
##
##   landscape_np      landscape_shei     landscape_siei
##  Min.   :-1.1343   Min.   :-4.12918   Min.   :-4.3075
##  1st Qu.:-0.5236   1st Qu.:-0.64137   1st Qu.:-0.5770
##  Median :-0.2777   Median : 0.07569   Median : 0.2271
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.1454   3rd Qu.: 0.70654   3rd Qu.: 0.7397
##  Max.   : 7.0051   Max.   : 3.20495   Max.   : 1.9831
##
```

```
rm(covs, response)
```

1.2. Re-identify top scales:

```
# Pull from the model or specify manually. Manual means we don't need to re-run this every time.
nat_buffer <- 100

comp_buffer <- 4250

config_buffer <- 2250
```

1.3. Pull out the final data again

We will pull out the predictors from the submodels **at the appropriate spatial scales** and merge this into one big dataset for final models. This selects a couple extra columns that we don't actually want to model but that's fine, this was efficient. This is the *unscaled* data so that we can run the sampler and apply appropriate constraints before scaling the data again.

```
data_final <- bind_cols(

  # response variables
  data %>%
    select(1:squirrel) %>%
    distinct(),

  # natural data scaled by natural buffer
  data %>%
    filter(buffer_dist == nat_buffer) %>%
    select(fire_0_15:lc_wetland_treed) %>%
    mutate(natural_buffer = nat_buffer), # won't use this column, just keeping track of the scale someh

  # composition data scaled by composition buffer
  data %>%
    filter(buffer_dist == comp_buffer) %>%
    select(harvest_0_15:wells_total) %>%
    mutate(comp_buffer = comp_buffer), # won't use this column, just keeping track of the scale somehow

  # configuration data
  data %>%
    filter(buffer_dist == config_buffer) %>%
    select(contains("nonanthro") | contains("landscape") | contains("cfi")) %>%
    mutate(config_buffer = config_buffer) # won't use this column, just keeping track of the scale some
  ) %>%

  relocate(contains("buffer"), .after=("squirrel"))

summary(data_final)
```

```
##     array               site               month              year
##  Length:4918        Length:4918        Min.   : 1.000   Min.   :2021
##  Class :character   Class :character   1st Qu.: 4.000   1st Qu.:2022
##  Mode  :character   Mode  :character   Median : 7.000   Median :2023
##                                        Mean   : 6.634   Mean   :2023
```

```
##                                        3rd Qu.:10.000   3rd Qu.:2024
##                                        Max.   :12.000   Max.   :2024
##     squirrel       natural_buffer  comp_buffer   config_buffer
##  Min.   : 0.0000   Min.   :100    Min.   :4250   Min.   :2250
##  1st Qu.: 0.0000   1st Qu.:100    1st Qu.:4250   1st Qu.:2250
##  Median : 0.0000   Median :100    Median :4250   Median :2250
##  Mean   : 0.9339   Mean   :100    Mean   :4250   Mean   :2250
##  3rd Qu.: 1.0000   3rd Qu.:100    3rd Qu.:4250   3rd Qu.:2250
##  Max.   :70.0000   Max.   :100    Max.   :4250   Max.   :2250
##    fire_0_15         fire_gt_15     pct_lari_lar       lc_broadleaf
##  Min.   :0.00000   Min.   :0.000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.000000   1st Qu.:0.001251
##  Median :0.00000   Median :0.000   Median :0.000000   Median :0.031500
##  Mean   :0.05903   Mean   :0.025   Mean   :0.017705   Mean   :0.165230
##  3rd Qu.:0.00000   3rd Qu.:0.000   3rd Qu.:0.002441   3rd Qu.:0.207147
##  Max.   :0.99998   Max.   :1.000   Max.   :0.410436   Max.   :0.997400
##  lc_coniferous       lc_herbs         lc_mixedwood        lc_shrubs
##  Min.   :0.0000   Min.   :0.000000   Min.   :0.000000   Min.   :0.0000000
##  1st Qu.:0.1166   1st Qu.:0.000000   1st Qu.:0.001708   1st Qu.:0.0000000
##  Median :0.3411   Median :0.004752   Median :0.024024   Median :0.0000000
##  Mean   :0.3693   Mean   :0.030944   Mean   :0.077900   Mean   :0.0254530
##  3rd Qu.:0.5895   3rd Qu.:0.033106   3rd Qu.:0.093809   3rd Qu.:0.0006952
##  Max.   :0.9961   Max.   :0.431069   Max.   :0.710038   Max.   :0.9556450
##     lc_water          lc_wetland       lc_wetland_treed   harvest_0_15
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.001763   1st Qu.:0.04007   1st Qu.:0.000000
##  Median :0.000000   Median :0.013062   Median :0.15314   Median :0.000000
##  Mean   :0.002367   Mean   :0.061577   Mean   :0.23877   Mean   :0.015072
##  3rd Qu.:0.000000   3rd Qu.:0.064013   3rd Qu.:0.38478   3rd Qu.:0.009129
##  Max.   :0.180988   Max.   :0.635488   Max.   :0.95080   Max.   :0.190679
##  harvest_gt_15      harvest_total     osm_industrial       pipe_trans
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.001380   1st Qu.:0.001343
##  Median :0.006122   Median :0.01429   Median :0.004808   Median :0.009432
##  Mean   :0.034421   Mean   :0.04964   Mean   :0.017392   Mean   :0.013699
##  3rd Qu.:0.037029   3rd Qu.:0.05920   3rd Qu.:0.017930   3rd Qu.:0.019834
##  Max.   :0.357731   Max.   :0.38829   Max.   :0.316999   Max.   :0.075590
##     railways             roads              seismic         seismic_lines
##  Min.   :0.000e+00   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.000e+00   1st Qu.:0.001762   1st Qu.:0.004996   1st Qu.:0.004314
##  Median :0.000e+00   Median :0.003128   Median :0.008535   Median :0.006728
##  Mean   :1.247e-05   Mean   :0.004132   Mean   :0.012189   Mean   :0.007705
##  3rd Qu.:0.000e+00   3rd Qu.:0.005610   3rd Qu.:0.015080   3rd Qu.:0.009866
##  Max.   :1.701e-03   Max.   :0.023507   Max.   :0.075208   Max.   :0.025486
##  seismic_lines_3D       trails              veg_edges          wells_active
##  Min.   :0.000000   Min.   :0.0000000   Min.   :0.000000   Min.   :0.000e+00
##  1st Qu.:0.000000   1st Qu.:0.0005512   1st Qu.:0.002505   1st Qu.:5.372e-05
##  Median :0.000000   Median :0.0013095   Median :0.005061   Median :1.047e-03
##  Mean   :0.004484   Mean   :0.0018157   Mean   :0.007228   Mean   :4.726e-03
##  3rd Qu.:0.001867   3rd Qu.:0.0024169   3rd Qu.:0.009277   3rd Qu.:5.248e-03
##  Max.   :0.064481   Max.   :0.0096659   Max.   :0.062127   Max.   :6.110e-02
##  wells_inactive      wells_total       nonanthro_cai_mn   nonanthro_ed
##  Min.   :0.000000   Min.   :0.000000   Min.   : 0.4909   Min.   :  2.463
##  1st Qu.:0.001104   1st Qu.:0.001783   1st Qu.:21.4830   1st Qu.: 48.377
```

```
##  Median :0.002435   Median :0.005198   Median :32.7976   Median : 70.769
##  Mean   :0.004890   Mean   :0.009951   Mean   :37.0582   Mean   : 96.125
##  3rd Qu.:0.007435   3rd Qu.:0.013065   3rd Qu.:50.4579   3rd Qu.:107.282
##  Max.   :0.026174   Max.   :0.091417   Max.   :97.6249   Max.   :503.042
##  nonanthro_tca    nonanthro_cohesion landscape_cai_mn  landscape_ed
##  Min.   :  59.2   Min.   : 94.09     Min.   :0        Min.   :0
##  1st Qu.: 784.2   1st Qu.: 99.31     1st Qu.:0        1st Qu.:0
##  Median :1066.2   Median : 99.60     Median :0        Median :0
##  Mean   :1014.0   Mean   : 99.37     Mean   :0        Mean   :0
##  3rd Qu.:1297.2   3rd Qu.: 99.77     3rd Qu.:0        3rd Qu.:0
##  Max.   :1554.2   Max.   :100.00     Max.   :0        Max.   :0
##  landscape_tca landscape_cohesion landscape_contag landscape_mesh
##  Min.   :0     Min.   :99.08      Min.   :44.10    Min.   :  23.75
##  1st Qu.:0     1st Qu.:99.37      1st Qu.:58.15    1st Qu.:  73.26
##  Median :0     Median :99.48      Median :62.90    Median : 127.52
##  Mean   :0     Mean   :99.49      Mean   :63.57    Mean   : 169.60
##  3rd Qu.:0     3rd Qu.:99.62      3rd Qu.:68.46    3rd Qu.: 214.44
##  Max.   :0     Max.   :99.88      Max.   :90.14    Max.   :1246.62
##   landscape_np  landscape_shei   landscape_siei      cfi_site
##  Min.   :  25   Min.   :0.1603   Min.   :0.1494   Min.   :0.00000
##  1st Qu.: 207   1st Qu.:0.5517   1st Qu.:0.6497   1st Qu.:0.02116
##  Median : 349   Median :0.6453   Median :0.7651   Median :0.04418
##  Mean   : 540   Mean   :0.6334   Mean   :0.7280   Mean   :0.06532
##  3rd Qu.: 580   3rd Qu.:0.7238   3rd Qu.:0.8379   3rd Qu.:0.08529
##  Max.   :4255   Max.   :0.9884   Max.   :0.9898   Max.   :0.47178
##  cfi_site_with_harvest cfi_site_with_vegedges
##  Min.   :0.00000       Min.   :0.00000
##  1st Qu.:0.02849       1st Qu.:0.02669
##  Median :0.05899       Median :0.05142
##  Mean   :0.08354       Mean   :0.07503
##  3rd Qu.:0.11090       3rd Qu.:0.09767
##  Max.   :0.54318       Max.   :0.47364
```

1.4. Re-fit the top model:

Have to fit using the scaled data. Hacky way to deal with scaling and code is a bit repetitive but it works.

```
data_final_scaled <- bind_cols(

  # response variables
  data_scaled %>%
    select(1:squirrel) %>%
    distinct(),

  # natural data scaled by natural buffer
  data_scaled %>%
    filter(buffer_dist == nat_buffer) %>%
    select(fire_0_15:lc_wetland_treed) %>%
    mutate(natural_buffer = nat_buffer), # won't use this column, just keeping track of the scale someh

  # composition data scaled by composition buffer
  data_scaled %>%
    filter(buffer_dist == comp_buffer) %>%
```

```
    select(harvest_0_15:wells_total) %>%
    mutate(comp_buffer = comp_buffer), # won't use this column, just keeping track of the scale somehow

  # configuration data
  data_scaled %>%
    filter(buffer_dist == config_buffer) %>%
    select(contains("nonanthro") | contains("landscape") | contains("cfi")) %>%
    mutate(config_buffer = config_buffer) # won't use this column, just keeping track of the scale some
  ) %>%

  relocate(contains("buffer"), .after=("squirrel"))

  m_edgeXcfi <- glmmTMB(squirrel ~

                          # natural covariates
                          fire_0_15 +
                          lc_broadleaf +
                          lc_coniferous +
                          lc_mixedwood +

                          # configuration variables
                          #landscape_shei +
                          nonanthro_ed +
                          cfi_site +
                          nonanthro_ed*cfi_site +
                          #landscape_mesh +

                          (1|array/site),

                        data = data_final_scaled,
                        family = nbinom2,
                        na.action = na.fail)

summary(m_edgeXcfi)
```

```
##  Family: nbinom2  ( log )
## Formula:
## squirrel ~ fire_0_15 + lc_broadleaf + lc_coniferous + lc_mixedwood +
##     nonanthro_ed + cfi_site + nonanthro_ed * cfi_site + (1 |      array/site)
## Data: data_final_scaled
##
##      AIC      BIC   logLik deviance df.resid
##   9394.4   9465.9  -4686.2   9372.4     4907
##
## Random effects:
##
## Conditional model:
##  Groups     Name        Variance Std.Dev.
##  site:array (Intercept) 2.502    1.5816
##  array      (Intercept) 0.410    0.6403
## Number of obs: 4918, groups:  site:array, 430; array, 10
##
## Dispersion parameter for nbinom2 family (): 0.811
```

```
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.42793    0.23031  -6.200 5.65e-10 ***
## fire_0_15              0.15770    0.11215   1.406   0.1597
## lc_broadleaf          -0.51842    0.12835  -4.039 5.36e-05 ***
## lc_coniferous         -0.03228    0.11093  -0.291   0.7710
## lc_mixedwood           0.16358    0.09223   1.774   0.0761 .
## nonanthro_ed          -0.08884    0.16631  -0.534   0.5932
## cfi_site               0.29548    0.12080   2.446   0.0144 *
## nonanthro_ed:cfi_site -0.25618    0.10011  -2.559   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. Run the simulations

2.1. Simulation function

Manually set parameters for debugging the function (eval false)

```
model = m_edgeXcfi
original_data = data_final
n_array = 10
n_site_per_array = 43
n_obs_per_site = 10
sampler_type = "normal"
```

```
simulate_top_model <- function(model, original_data, n_array, n_site_per_array, n_obs_per_site, sampler

  # Fixed effects in order of model appearance
  B0 <- fixef(model)$cond[1]
  B1 <- fixef(model)$cond[2]
  B2 <- fixef(model)$cond[3]
  B3 <- fixef(model)$cond[4]
  B4 <- fixef(model)$cond[5]
  B5 <- fixef(model)$cond[6]
  B6 <- fixef(model)$cond[7]
  B7 <- fixef(model)$cond[8]

  # Random effects
  SD_site <- sqrt(as.numeric(summary(model)$varcor$cond[["site:array"]]))
  SD_array <- sqrt(as.numeric(summary(model)$varcor$cond[["array"]]))

  if(sampler_type == "normal"){
    # Generate data values based on normal distribution
    fire_0_15 <- rnorm(100000, mean = mean(original_data$fire_0_15),
                       sd = sd(original_data$fire_0_15))
    lc_broadleaf <- rnorm(100000, mean = mean(original_data$lc_broadleaf),
                          sd = sd(original_data$lc_broadleaf))
    lc_coniferous <- rnorm(100000, mean = mean(original_data$lc_coniferous),
                           sd = sd(original_data$lc_coniferous))
    lc_mixedwood <- rnorm(100000, mean = mean(original_data$lc_mixedwood),
```

```r
                      sd = sd(original_data$lc_mixedwood))
  nonanthro_ed <- rnorm(100000, mean = mean(original_data$nonanthro_ed),
                      sd = sd(original_data$nonanthro_ed))
  cfi_site <- rnorm(100000, mean = mean(original_data$cfi_site),
                  sd = sd(original_data$cfi_site))
}


if(sampler_type == "uniform"){
  # Generate data values based on uniform distribution
  fire_0_15 <- runif(100000, min = min(original_data$fire_0_15),
                   max = max(original_data$fire_0_15))
  lc_broadleaf <- runif(100000, min = min(original_data$lc_broadleaf),
                      max = max(original_data$lc_broadleaf))
  lc_coniferous <- runif(100000, min = min(original_data$lc_coniferous),
                       max = max(original_data$lc_coniferous))
  lc_mixedwood <- runif(100000, min = min(original_data$lc_mixedwood),
                      max = max(original_data$lc_mixedwood))
  nonanthro_ed <- runif(100000, min = min(original_data$nonanthro_ed),
                      max = max(original_data$nonanthro_ed))
  cfi_site <- runif(100000, min = min(original_data$cfi_site),
                  max = max(original_data$cfi_site))
}

data_sim <- tibble(
              fire_0_15,
              lc_broadleaf,
              lc_coniferous,
              lc_mixedwood,
              nonanthro_ed,
              cfi_site
              ) %>%

  mutate(landcover = lc_broadleaf + lc_coniferous + lc_mixedwood) %>%

  filter(if_all(everything(), ~ . >= 0),
         landcover<1) %>%

  filter(row_number(.) <= n_array * n_site_per_array) %>%

  # Scale the data using the original scaled values
  mutate(
    fire_0_15 = (fire_0_15 - mean(original_data$fire_0_15, na.rm = TRUE)) /
      sd(original_data$fire_0_15, na.rm = TRUE),
    lc_broadleaf = (lc_broadleaf - mean(original_data$lc_broadleaf, na.rm = TRUE)) /
      sd(original_data$lc_broadleaf, na.rm = TRUE),
    lc_coniferous = (lc_coniferous - mean(original_data$lc_coniferous, na.rm = TRUE)) /
      sd(original_data$lc_coniferous, na.rm = TRUE),
    lc_mixedwood = (lc_mixedwood - mean(original_data$lc_mixedwood, na.rm = TRUE)) /
      sd(original_data$lc_mixedwood, na.rm = TRUE),
    nonanthro_ed = (nonanthro_ed - mean(original_data$nonanthro_ed, na.rm = TRUE)) /
      sd(original_data$nonanthro_ed, na.rm = TRUE),
    cfi_site = (cfi_site - mean(original_data$cfi_site, na.rm = TRUE)) /
```

```
      sd(original_data$cfi_site, na.rm = TRUE)
  ) %>%

  mutate(array = paste0("Array", rep(1:n_array, each = n_site_per_array)),
         site = paste0(array, "_", 1:n_site_per_array)) %>%

  crossing(obs = 1:n_obs_per_site, .) %>%

  group_by(array) %>%

  mutate(ranef_array = rnorm(n = 1, mean = 0, sd = SD_array)) %>%

  ungroup() %>%

  group_by(array, site) %>%

  mutate(ranef_site = rnorm(n = 1, mean = 0, sd = SD_site)) %>%

  ungroup() %>%

  mutate(eta = B0 +
           B1 * fire_0_15 +
           B2 * lc_broadleaf +
           B3 * lc_coniferous +
           B4 * lc_mixedwood +
           B5 * nonanthro_ed +
           B6 * cfi_site +
           B7 * nonanthro_ed * cfi_site +
           ranef_array +
           ranef_site,

         mu = exp(eta),

         squirrel = rnbinom(n = n(), mu = mu, size = sigma(model)))

sim_naive_occ <- data_sim %>%
  group_by(site) %>%
  summarize(dets = max(squirrel)) %>%
  mutate(presence = ifelse(dets > 0, 1, 0)) %>%
  summarize(sum(presence)) %>%
  as.numeric()

sim_indet <- sum(data_sim$squirrel)

sim <- glmmTMB(squirrel ~

                          # natural covariates
                          fire_0_15 +
                          lc_broadleaf +
                          lc_coniferous +
                          lc_mixedwood +

                          # configuration variables
```

```
                            #landscape_shei +
                            nonanthro_ed +
                            cfi_site +
                            nonanthro_ed*cfi_site +
                            #landscape_mesh +

                            (1|array/site),

                          data = data_sim,
                          family = nbinom2,
                          na.action = na.fail)


  sim_summary <- tibble(cov = names(fixef(sim)$cond),
                value = fixef(sim)$cond) %>%

      pivot_wider(names_from = cov) %>%

      mutate(`site:array` = sqrt(as.numeric(VarCorr(sim)$cond[[1]])),
             array = sqrt(as.numeric(VarCorr(sim)$cond[[2]])),
             convergence = sim$fit$convergence,
             n_array = n_array,
             n_site_per_array = n_site_per_array,
             n_obs_per_site = n_obs_per_site,
             sim_indet = sim_indet,
             sim_naive_occ = sim_naive_occ,
             sampler_type = sampler_type,
             sim_date = as.character(Sys.Date()),
             )

  # Only return a model if it converged
  #if(sim$fit$convergence==0){
  #  return(sim_summary)
  #}

}
```

2.2. Run simulations

Erase all previous simulations:

```
sim_master_results <- list()
save(sim_master_results, file = "./data/raw/simulation_checkpoint.RData")
```

Run a fresh set of 1000 simulations (in batches of 10) for each combination of parameters that we want.

```
# 1000 simulations with normal sampler
for(i in 1:100) {

cat("\r\r Working on `normal` batch", i)

load("./data/raw/simulation_checkpoint.RData")
```

```
sim_results <- purrr::map_dfr(1:10, ~
                simulate_top_model(model = m_edgeXcfi,
                                   original_data = data_final,
                                   n_array = 10,
                                   n_site_per_array = 43,
                                   n_obs_per_site = 10,
                                   sampler_type = "normal"
                                   )
                )

sim_master_results[[length(sim_master_results)+1]] <- sim_results

save(sim_master_results, file = "./data/raw/simulation_checkpoint.RData")
}

# 1000 simulations with uniform sampler
for(i in 1:100) {

cat("\r\r Working on `uniform` batch", i)

load("./data/raw/simulation_checkpoint.RData")

sim_results <- purrr::map_dfr(1:10, ~
                simulate_top_model(model = m_edgeXcfi,
                                   original_data = data_final,
                                   n_array = 10,
                                   n_site_per_array = 43,
                                   n_obs_per_site = 10,
                                   sampler_type = "uniform"
                                   )
                )

sim_master_results[[length(sim_master_results)+1]] <- sim_results

save(sim_master_results, file = "./data/raw/simulation_checkpoint.RData")
}

# 1000 simulations with reduced sites per array
for(i in 1:100) {

cat("\r\r Working on `reduced sites` batch", i)

load("./data/raw/simulation_checkpoint.RData")

sim_results <- purrr::map_dfr(1:10, ~
                simulate_top_model(model = m_edgeXcfi,
                                   original_data = data_final,
                                   n_array = 10,
                                   n_site_per_array = 22,
                                   n_obs_per_site = 10,
                                   sampler_type = "uniform"
                                   )
                )
```

```
sim_master_results[[length(sim_master_results)+1]] <- sim_results

save(sim_master_results, file = "./data/raw/simulation_checkpoint.RData")
}
```

3. Visualize results of the simulations

Let's take a look at the master results

```
load("./data/raw/simulation_checkpoint.RData")

# Fetch the simulation results
sim_master_results_df <-  sim_master_results %>%
    bind_rows() %>%
    distinct() # Make sure there are no duplicate simulations.
```

Let's also import the pretty names

```
pretty_names <- read_csv("./tables/OSM_all_covariates_formatted_names.csv")
```

```
## Rows: 44 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Covariate, PrettyName
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Plot using the normal (Gaussian) sampler

```
true_vals_fixef <- enframe(fixef(m_edgeXcfi)$cond,
                           name = "term", value = "value")

true_vals_ranef <- enframe(VarCorr(m_edgeXcfi)$cond,
                           name = "term", value = "value") %>%
  mutate(value = sqrt(as.numeric(value)))

true_vals <- bind_rows(true_vals_fixef,
                       true_vals_ranef)


# Run simulations
sim_coefficient_plots <- purrr::map(true_vals$term, ~{

  # Pretty label for the plot
  label <- (pretty_names %>% filter(Covariate == .x))$PrettyName

  true_val <- (true_vals %>% filter(term==.x))$value

  # Fetch the simulation results
```

```
  sim_master_results_df %>%

    # For a specific subset of simulations
    filter(sampler_type == "normal",
           n_site_per_array == 43,
           convergence == 0) %>%

    # Pivot to long format to make our lives easier in purrr
    pivot_longer(., cols = 1:10) %>%

    # Coefficient values for just the variable we want
    filter(name == .x) %>%

    # Plot it
    ggplot(., aes(x = value)) +
  # Dotted vertical line at x = 0
      geom_vline(xintercept = 0, color = "grey70") +

      geom_density(fill = ifelse(str_detect(label, "Random")==TRUE,
                                 "darkblue", "darkred"), alpha = 0.3) +

      geom_vline(xintercept = true_val,
                 linetype = "dashed",
                 color = ifelse(str_detect(label, "Random")==TRUE,
                                "darkblue", "darkred"),
                 linewidth=0.8) +  # Dotted vertical line at x = 0

        # Crop the axis since model weights are low
      scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +

      scale_x_continuous(limits = c(true_val-0.75, true_val+0.75),
                         expand = c(0,0)) +

      labs(y = "",
           x = "",
           title = label) +

      theme_bw() +

      theme(panel.grid = element_blank())  # Remove background panel grid
})

ggpubr::ggarrange(plotlist = sim_coefficient_plots, ncol = 3, nrow = 4) +
  bgcolor("white")

ggsave("./figures/top_model_simulations_coefficient_density_gaussian_sampler.png",
       width =9, height = 8)
```

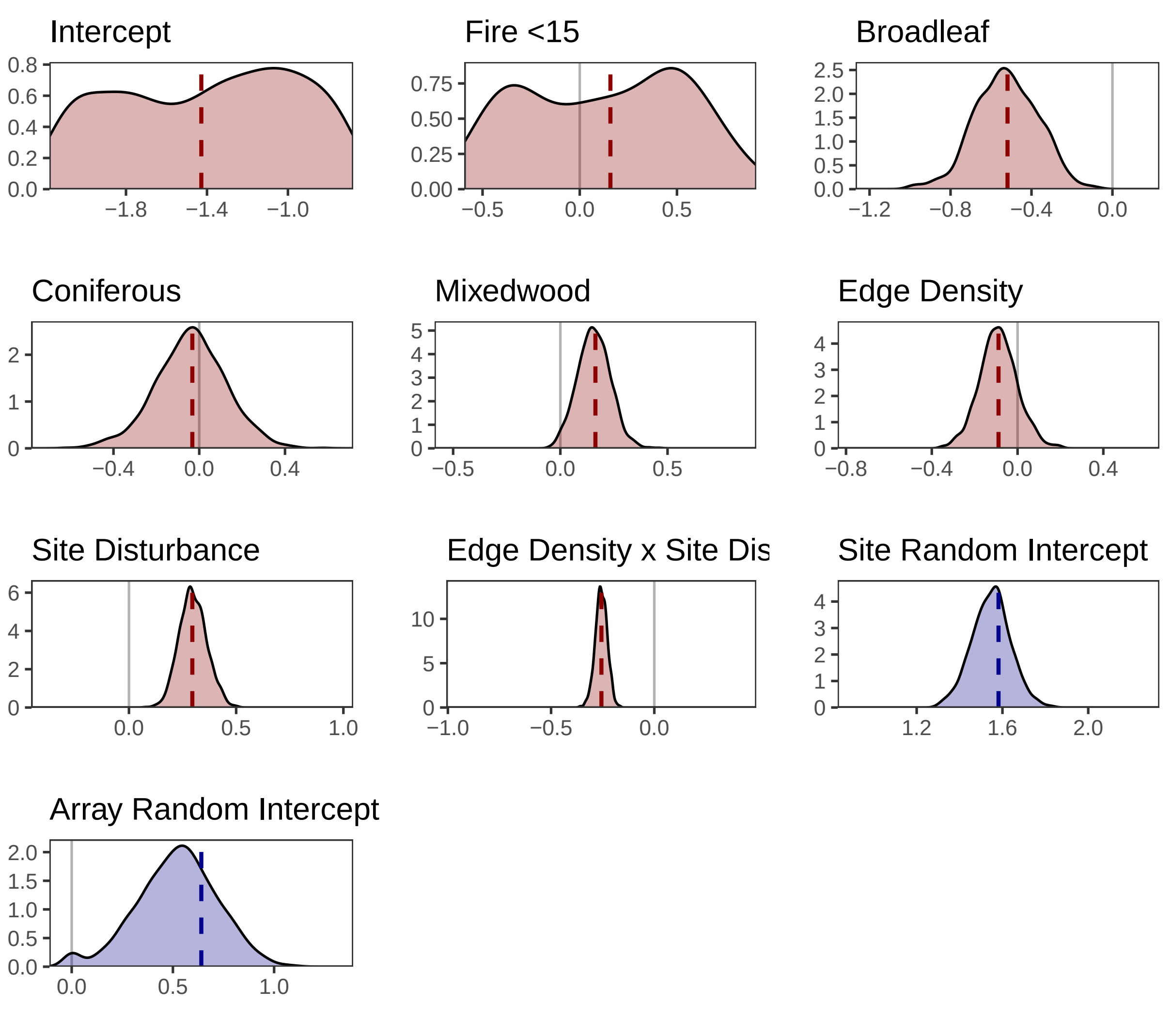
Plot using the uniform distribution sampler

```
true_vals_fixef <- enframe(fixef(m_edgeXcfi)$cond,
                           name = "term", value = "value")
```

```r
true_vals_ranef <- enframe(VarCorr(m_edgeXcfi)$cond,
                           name = "term", value = "value") %>%
  mutate(value = sqrt(as.numeric(value)))

true_vals <- bind_rows(true_vals_fixef,
                       true_vals_ranef)


# Run simulations
sim_coefficient_plots <- purrr::map(true_vals$term, ~{

  # Pretty label for the plot
  label <- (pretty_names %>% filter(Covariate == .x))$PrettyName

  true_val <- (true_vals %>% filter(term==.x))$value

  # Fetch the simulation results
  sim_master_results_df %>%

    # For a specific subset of simulations
    filter(sampler_type == "uniform",
           n_site_per_array == 43,
           convergence == 0) %>%

    # Pivot to long format to make our lives easier in purrr
    pivot_longer(., cols = 1:10) %>%

    # Coefficient values for just the variable we want
    filter(name == .x) %>%

    # Plot it
    ggplot(., aes(x = value)) +

      geom_vline(xintercept = 0, color = "grey70") +

      geom_density(fill = ifelse(str_detect(label, "Random")==TRUE,
                                 "darkblue", "darkred"), alpha = 0.3) +

      geom_vline(xintercept = true_val,
                 linetype = "dashed",
                 color = ifelse(str_detect(label, "Random")==TRUE,
                                "darkblue", "darkred"),
                 linewidth=0.8) +  # Dotted vertical line at x = 0

        # Crop the axis since model weights are low
      scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +

      scale_x_continuous(limits = c(true_val-0.75, true_val+0.75),
                         expand = c(0,0)) +

      labs(y = "",
           x = "",
           title = label) +
```

```
      theme_bw() +

      theme(panel.grid = element_blank())  # Remove background panel grid
})

ggpubr::ggarrange(plotlist = sim_coefficient_plots, ncol = 3, nrow = 4) +
  bgcolor("white")
```

```
## Warning: Removed 428 rows containing non-finite outside the scale range
## (`stat_density()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_vline()`).

## Warning: Removed 854 rows containing non-finite outside the scale range
## (`stat_density()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_vline()`).
```



```
ggsave("./figures/top_model_simulations_coefficient_density_uniform_sampler.png",
       width = 9, height = 8)
```

Plot using the uniform sampler *and* reduced sample size

```
true_vals_fixef <- enframe(fixef(m_edgeXcfi)$cond,
                           name = "term", value = "value")

true_vals_ranef <- enframe(VarCorr(m_edgeXcfi)$cond,
                           name = "term", value = "value") %>%
  mutate(value = sqrt(as.numeric(value)))

true_vals <- bind_rows(true_vals_fixef,
                       true_vals_ranef)


# Run simulations
sim_coefficient_plots <- purrr::map(true_vals$term, ~{

  # Pretty label for the plot
  label <- (pretty_names %>% filter(Covariate == .x))$PrettyName

  true_val <- (true_vals %>% filter(term==.x))$value

  # Fetch the simulation results
  sim_master_results_df %>%

    # For a specific subset of simulations
    filter(sampler_type == "uniform",
           n_site_per_array == 22,
           convergence == 0) %>%

    # Pivot to long format to make our lives easier in purrr
    pivot_longer(., cols = 1:10) %>%

    # Coefficient values for just the variable we want
    filter(name == .x) %>%

    # Plot it
    ggplot(., aes(x = value)) +

      geom_vline(xintercept = 0, color = "grey70") +

      geom_density(fill = ifelse(str_detect(label, "Random")==TRUE,
                                 "darkblue", "darkred"), alpha = 0.3) +

      geom_vline(xintercept = true_val,
                 linetype = "dashed",
                 color = ifelse(str_detect(label, "Random")==TRUE,
                               "darkblue", "darkred"),
                 linewidth=0.8) +  # Dotted vertical line at x = 0

        # Crop the axis since model weights are low
```

```
      scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +

      scale_x_continuous(limits = c(true_val-0.75, true_val+0.75),
                         expand = c(0,0)) +

      labs(y = "",
           x = "",
           title = label) +

      theme_bw() +

      theme(panel.grid = element_blank())  # Remove background panel grid
})

ggpubr::ggarrange(plotlist = sim_coefficient_plots, ncol = 3, nrow = 4) +
  bgcolor("white")

ggsave("./figures/top_model_simulations_coefficient_density_reduced_sample_size.png",
       width = 9, height = 8)
```

Let's look at some summary statistics for our empirical samples. Do they match the simulations?

```
true_vals <- enframe(fixef(m_edgeXcfi)$cond,
                     name = "term", value = "mean_model")
true_se <- enframe(summary(m_edgeXcfi)$coefficients$cond[, "Std. Error"],
                   name = "term", value = "se_model")

  # Fetch the simulation results
sim_uniform_results <- sim_master_results_df %>%

    # For a specific subset of simulations
    filter(sampler_type == "uniform",
           n_site_per_array == 43,
           convergence == 0) %>%

  select(1:8) %>%

  mutate(sim_id = row_number()) %>%

  pivot_longer(., cols = 1:8, names_to = "term") %>%

  group_by(term) %>%

  summarize(
    mean_estimate = mean(value),
    se_estimate = sd(value)
  ) %>%

  left_join(true_vals, by = "term") %>%

  left_join(true_se, by = "term") %>%

  mutate(mean_bias = mean_estimate - mean_model,
```

```r
        se_ratio = se_estimate / se_model)%>%

  select(term, contains("mean"), contains("se")) %>%

  arrange(mean_bias)

sim_uniform_results
```

```
## # A tibble: 8 x 7
##   term          mean_estimate mean_model mean_bias se_estimate se_model se_ratio
##   <chr>                 <dbl>      <dbl>     <dbl>       <dbl>    <dbl>    <dbl>
## 1 lc_mixedwood         0.157      0.164  -0.00647       0.0760   0.0922    0.824
## 2 lc_coniferous       -0.0384    -0.0323 -0.00608       0.164    0.111     1.48
## 3 lc_broadleaf        -0.524     -0.518  -0.00548       0.155    0.128     1.21
## 4 nonanthro_ed~       -0.257     -0.256  -0.000673      0.0293   0.100     0.293
## 5 nonanthro_ed        -0.0890    -0.0888 -0.000197      0.0889   0.166     0.535
## 6 cfi_site             0.299      0.295   0.00344       0.0641   0.121     0.530
## 7 (Intercept)         -1.40      -1.43    0.0310        0.949    0.230     4.12
## 8 fire_0_15            0.289      0.158   0.131         3.72     0.112    33.1
```

```r
summary(m_edgeXcfi)
```

```
##  Family: nbinom2  ( log )
## Formula:
## squirrel ~ fire_0_15 + lc_broadleaf + lc_coniferous + lc_mixedwood +
##     nonanthro_ed + cfi_site + nonanthro_ed * cfi_site + (1 |      array/site)
## Data: data_final_scaled
##
##      AIC      BIC   logLik deviance df.resid
##   9394.4   9465.9  -4686.2   9372.4     4907
##
## Random effects:
##
## Conditional model:
##  Groups     Name        Variance Std.Dev.
##  site:array (Intercept) 2.502    1.5816
##  array      (Intercept) 0.410    0.6403
## Number of obs: 4918, groups:  site:array, 430; array, 10
##
## Dispersion parameter for nbinom2 family (): 0.811
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.42793    0.23031  -6.200 5.65e-10 ***
## fire_0_15              0.15770    0.11215   1.406   0.1597
## lc_broadleaf          -0.51842    0.12835  -4.039 5.36e-05 ***
## lc_coniferous         -0.03228    0.11093  -0.291   0.7710
## lc_mixedwood           0.16358    0.09223   1.774   0.0761 .
## nonanthro_ed          -0.08884    0.16631  -0.534   0.5932
## cfi_site               0.29548    0.12080   2.446   0.0144 *
## nonanthro_ed:cfi_site -0.25618    0.10011  -2.559   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

End script