

Scale analysis and final models

Aidan Brushett and Emerald Arthurs

2025-05-02

Contents

Before you begin	2
0. Setup	3
1. Prepare the data for modeling	3
1.1. Import the data	3
1.2. Scale the data and store values	7
2. Scale analysis:	7
2.1. Natural sub-model:	8
2.2. Anthro composition submodel:	8
2.3. Configuration sub-model:	9
2.4. Let's put it all together into a plot:	10
2.5. Identify top scales:	11
3. Construct the final dataset (best spatial scales)	12
4. Fit models using the best-fit scales for each 'category' of variables.	16
5. Perform model selection	24
6. Model inspection and validation	28
6.0. A bit of set up for tables and figures	28
6.1. Check out a general summary:	28
6.2. Pseudo-R2	29
6.3. Check dispersion:	29
6.4. Check other diagnostics including VIFs.	30
6.5. Variance inflation factors:	30
6.6. Likelihood ratio test for random effect structure:	32

7. Interpret results	34
7.1. Model summary	34
7.2. Odds ratio plots for the conditional model:	34
7.3. Odds ratio plots for the second-place model	36
7.4. Export the combined top and second-ranking model figures in one!	38
8. Plot conditional effects	39
8.1. Decid:	39
8.2. Conifer:	40
8.3. Mixedwood:	41
8.4. Recent fire	42
8.5. Cumulative site disturbance:	43
8.6. Edge density:	44
8.6. Edge x CFI interaction	45
8.7. CFI x Edge interaction	47
8.8. Combined plot:	49
9. Make a pretty AIC table for the report	50
10. Make a pretty model summary table	51
10.1. for the top model	51
10.2. For the second place model	52

Before you begin

This script is number 5 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, evaluate, and visualize all models of landscape structure and its influence on the spatial distribution of red squirrels in our study area. This includes all model fitting, model selection, diagnostics, and effects plots.

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(rphylopic)
library(corr)
library(performance)
#library(PerformanceAnalytics)
library(broom.mixed)
library(cowplot)
library(writexl)

`%nin%` <- Negate(`%in%`)
```

1. Prepare the data for modeling

We will also apply standardized z-scaling to the data. This is done *per buffer* since technically they are different datasets.

1.1. Import 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 relationship
## 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   : 0.00000 Mean   : 0.00000 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 : 0.00000
##	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 : 0.00000
##	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 : 0.00000
##	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 : 0.00000
##	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. Scale the data and store values

I also want to make sure we store the values we used to scale the data so that we can back-transform it into ecologically meaningful values for figures. Either it's annoying to fit models (typing `scale()` all the time or annoying to make plots. Would have been faster to write 'scale' in all the models but oh well... maybe I'll fix this one day and learn from my mistakes. Laziness does not pay off in the long run.

```
data_scaled_summary <- data %>%
  select(buffer_dist, cfi_site:last_col()) %>%
  group_by(buffer_dist) %>%
  pivot_longer(cols = 2:last_col(), values_to = "value", names_to = "variable") %>%
  group_by(buffer_dist, variable) %>%
  # Store the mean and the sd for backtransforming later (don't judge me)
  summarize(
    mean = mean(value, na.rm = TRUE),
    sd = sd(value, na.rm = TRUE),
    .groups = "drop"
  )
```

2. Scale analysis:

Another nice way to visualize our results for the scale analysis is to plot the model weight for the global sub-models at each scale.

2.1. Natural sub-model:

```
natural_scale <- purrr::map(unique(data$buffer_dist), ~{  
  
  # Subset the data based on the current buffer distance  
  data_subset <- data_scaled %>%  
  
    filter(buffer_dist == .x)  
  
  # Fit the initial model for the current buffer distance  
  glmmTMB(squirrel ~  
  
    lc_coniferous +  
    lc_mixedwood +  
    lc_broadleaf +  
    lc_shrubs +  
    lc_herbs +  
    fire_0_15 +  
  
    (1|array/site),  
  
    data = data_subset,  
    family = nbinom2,  
    na.action = na.fail)  
}  
) %>%  
  
set_names(unique(data$buffer_dist)) %>%  
  
# Model selection and AIC for each scale  
MuMin::model.sel(natural_scale) %>%  
  
as.data.frame(.) %>%  
  
dplyr::mutate(buffer = as.numeric(rownames(.)),  
              submodel = "NATURAL")
```

2.2. Anthro composition submodel:

```
comp_scale <- purrr::map(unique(data$buffer_dist), ~{  
  
  # Subset the data based on the current buffer distance  
  data_subset <- data_scaled %>%  
  
    filter(buffer_dist == .x)  
  
  # Fit the initial model for the current buffer distance  
  glmmTMB(squirrel ~  
  
    pipe_trans +  
    seismic +  
    wells_inactive +  
    #harvest_total +  
    wells_active +  
    osm_industrial +
```



```

        harvest_0_15 +
        #roads +

        (1|array/site),

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

}
) %>%

set_names(unique(data$buffer_dist)) %>%

# Model selection and AIC for each scale
MuMIn::model.sel(comp_scale) %>%

as.data.frame(.) %>%

dplyr::mutate(buffer = as.numeric(rownames(.)),
              submodel = "COMP")

```

2.3. Configuration sub-model:

```

config_scale <- purrr::map(unique(data$buffer_dist), ~{

  # Subset the data based on the current buffer distance
  data_subset <- data_scaled %>%

    filter(buffer_dist == .x)

  # Fit the initial model for the current buffer distance
  glmmTMB(squirrel ~

    # configuration variables
    landscape_shei +
    nonanthro_ed +
    landscape_mesh +

    (1|array/site),

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

}
) %>%

set_names(unique(data$buffer_dist)) %>%

# Model selection and AIC for each scale
MuMIn::model.sel(config_scale) %>%

as.data.frame(.) %>%

dplyr::mutate(buffer = as.numeric(rownames(.)),

```

```
submodel = "CONFIG")
```

2.4. Let's put it all together into a plot:

```
# One big data frame
fig_scale <- bind_rows(natural_scale,
                      comp_scale,
                      config_scale) %>%

# Plot the model weights as a line.
ggplot(., aes(x = buffer, y = weight, color = submodel)) +

  geom_line(size = 1, alpha = 0.7) +

  # Old color scheme I didn't like (overwritten)
  scale_color_manual(values = c("#D24D57",
                                "#1976D2",
                                "#388E3C")) +

  scale_color_manual(values = c("goldenrod2", "darkblue", "forestgreen"),
                    labels = c("Disturbance \ncomposition", "Configuration", "Natural
                               ↪ \nlandcover")) +
  #scale_color_viridis_d() +

  theme_bw() +

  # Crop the axis since model weights are low
  scale_y_continuous(limits = c(0, 0.41), expand = expansion(add = c(0, 0))) +

  scale_x_continuous(expand = c(0,0), limits = c(0, 5100)) +

  #scale_x_log10(expand = c(0,0), limits = c(50, 5100)) +

  labs(x = "Buffer radius (m)",
       y = "AICc weight",
       color = "Covariate subset") +

  # Remove all grid lines
  theme(panel.grid = element_blank()) +

  # Squirrel icon for funsies. Wrong spp. but whatever.
  add_phylopic(
    uuid = get_uuid(name = "Sciurus vulgaris"),
    x = 3000,
    y = 0.34,
    height = 0.09
  ) +

  theme(
    legend.position = c(0.98, 0.99),          # Top-right corner
    legend.justification = c(1, 1)           # Anchor the legend's top-right
  )
```

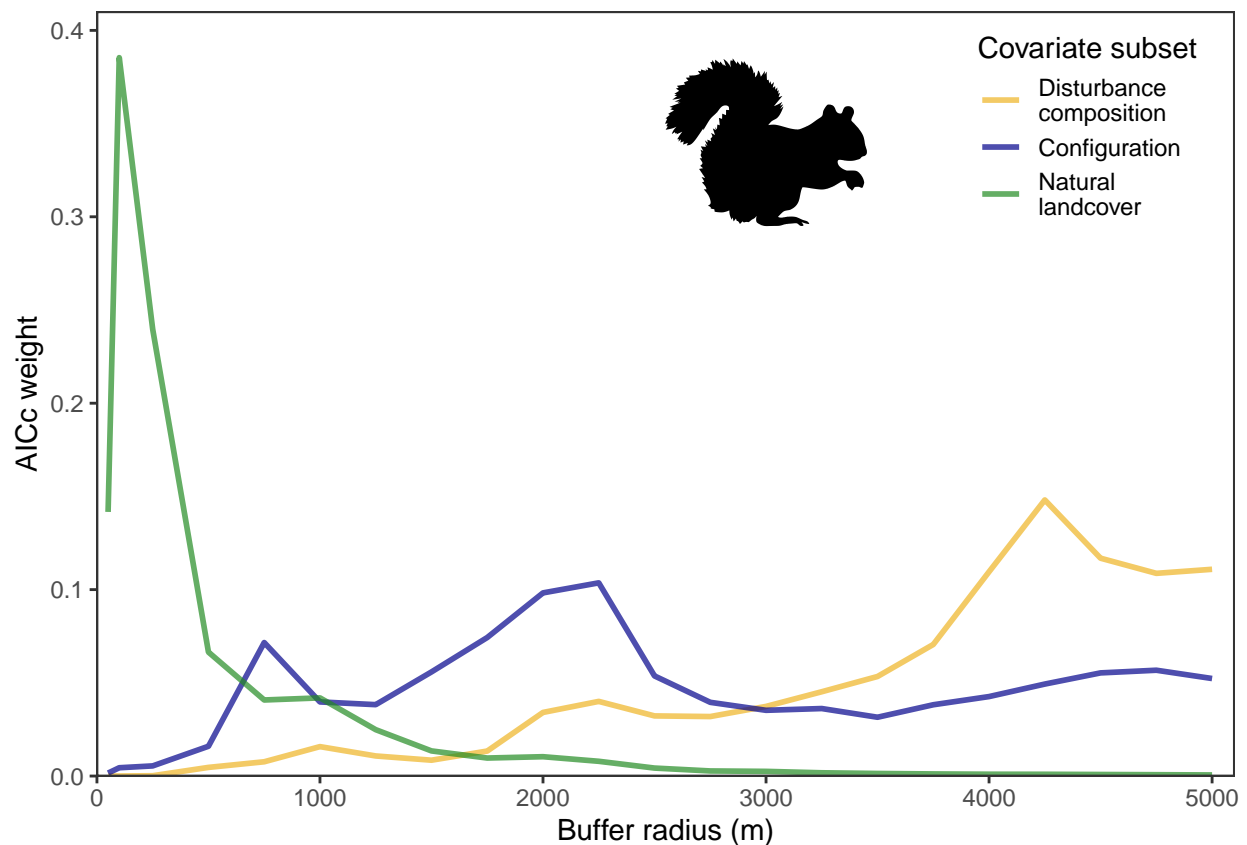
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.

## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
fig_scale
```



```
ggsave("./figures/redsquirrel_neg_binomial_submodel_scale_modelweight.png", width = 5, height =
  ↪ 3.5, dpi = 500)

#ggsave("./figures/redsquirrel_neg_binomial_submodel_scale_modelweight_logscale.png", width = 7,
  ↪ height = 4, dpi = 500)
```

2.5. Identify top scales:

What are the numbers from the above scale analysis?

```
natural_scale$buffer[1]
```

```
## [1] 100
```

```
comp_scale$buffer[1]
```

```
## [1] 4250
```

```
config_scale$buffer[1]
```

```
## [1] 2250
```

Store them manually here before we move to the next step.

```
# Let's specify these values manually. This means we don't need to re-run this every time we open a  
↪ new R session.  
nat_buffer <- 100  
comp_buffer <- 4250  
config_buffer <- 2250
```

3. Construct the final dataset (best spatial scales)

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 *#sueme*

```
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  
    ↪ somehow  
  
  # 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
  ↪ somehow
) %>%

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

select(-landscape_ed, -landscape_tca, -landscape_cai_mn)

summary(data_final_scaled)

```

```

##      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.      :-0.2844   Min.      :-0.190   Min.      :-0.3749   Min.      :-0.6332
## 1st Qu.: -0.2844   1st Qu.: -0.190   1st Qu.: -0.3749   1st Qu.: -0.6284
## Median : -0.2844   Median : -0.190   Median : -0.3749   Median : -0.5125
## Mean     : 0.0000   Mean     : 0.000   Mean     : 0.0000   Mean     : 0.0000
## 3rd Qu.: -0.2844   3rd Qu.: -0.190   3rd Qu.: -0.3232   3rd Qu.: 0.1606
## Max.     : 4.5335   Max.      : 7.409   Max.      : 8.3161   Max.      : 3.1892
##      lc_coniferous      lc_herbs      lc_mixedwood      lc_shrubs
## Min.      :-1.3520   Min.      :-0.52107   Min.      :-0.6342   Min.      :-0.2440
## 1st Qu.: -0.9249   1st Qu.: -0.52107   1st Qu.: -0.6203   1st Qu.: -0.2440
## Median : -0.1031   Median : -0.44105   Median : -0.4386   Median : -0.2440
## Mean     : 0.0000   Mean     : 0.00000   Mean     : 0.0000   Mean     : 0.0000
## 3rd Qu.: 0.8063   3rd Qu.: 0.03641   3rd Qu.: 0.1295   3rd Qu.: -0.2374
## Max.     : 2.2948   Max.      : 6.73769   Max.      : 5.1466   Max.      : 8.9180
##      lc_water      lc_wetland      lc_wetland_treed      harvest_0_15
## Min.      :-0.1678   Min.      :-0.56217   Min.      :-1.0134   Min.      :-0.4207
## 1st Qu.: -0.1678   1st Qu.: -0.54608   1st Qu.: -0.8433   1st Qu.: -0.4207
## Median : -0.1678   Median : -0.44292   Median : -0.3634   Median : -0.4207
## Mean     : 0.0000   Mean     : 0.00000   Mean     : 0.0000   Mean     : 0.0000
## 3rd Qu.: -0.1678   3rd Qu.: 0.02224   3rd Qu.: 0.6197   3rd Qu.: -0.1659
## Max.     :12.6617   Max.      : 5.23963   Max.      : 3.0220   Max.      : 4.9011
##      harvest_gt_15      harvest_total      osm_industrial      pipe_trans
## Min.      :-0.58153   Min.      :-0.6425   Min.      :-0.49935   Min.      :-0.9207
## 1st Qu.: -0.58153   1st Qu.: -0.6425   1st Qu.: -0.45973   1st Qu.: -0.8304
## Median : -0.47811   Median : -0.4576   Median : -0.36131   Median : -0.2868
## Mean     : 0.00000   Mean     : 0.0000   Mean     : 0.00000   Mean     : 0.0000
## 3rd Qu.: 0.04405   3rd Qu.: 0.1236   3rd Qu.: 0.01547   3rd Qu.: 0.4123
## Max.     : 5.46216   Max.      : 4.3829   Max.      : 8.60241   Max.      : 4.1597
##      railways      roads      seismic      seismic_lines
## Min.      :-0.102   Min.      :-1.0714   Min.      :-0.9918   Min.      :-1.5177

```

```
## 1st Qu.: -0.102 1st Qu.: -0.6144 1st Qu.: -0.5853 1st Qu.: -0.6680
## Median : -0.102 Median : -0.2604 Median : -0.2973 Median : -0.1923
## Mean : 0.000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: -0.102 3rd Qu.: 0.3832 3rd Qu.: 0.2352 3rd Qu.: 0.4256
## Max. : 13.822 Max. : 5.0237 Max. : 5.1278 Max. : 3.5024
## seismic_lines_3D trails veg_edges wells_active
## Min. : -0.4013 Min. : -1.0235 Min. : -0.8285 Min. : -0.53247
## 1st Qu.: -0.4013 1st Qu.: -0.7128 1st Qu.: -0.5413 1st Qu.: -0.52642
## Median : -0.4013 Median : -0.2853 Median : -0.2484 Median : -0.41446
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000
## 3rd Qu.: -0.2343 3rd Qu.: 0.3389 3rd Qu.: 0.2348 3rd Qu.: 0.05882
## Max. : 5.3692 Max. : 4.4249 Max. : 6.2924 Max. : 6.35239
## wells_inactive wells_total nonanthro_cai_mn nonanthro_ed
## Min. : -0.8992 Min. : -0.7462 Min. : -1.6276 Min. : -1.0673
## 1st Qu.: -0.6961 1st Qu.: -0.6125 1st Qu.: -0.6932 1st Qu.: -0.5441
## Median : -0.4514 Median : -0.3564 Median : -0.1896 Median : -0.2889
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: 0.4679 3rd Qu.: 0.2336 3rd Qu.: 0.5964 3rd Qu.: 0.1271
## Max. : 3.9137 Max. : 6.1091 Max. : 2.6958 Max. : 4.6367
## nonanthro_tca nonanthro_cohesion landscape_cohesion landscape_contag
## Min. : -2.8806 Min. : -6.39094 Min. : -2.60717 Min. : -2.49910
## 1st Qu.: -0.6934 1st Qu.: -0.06247 1st Qu.: -0.74738 1st Qu.: -0.69574
## Median : 0.1575 Median : 0.28132 Median : -0.06761 Median : -0.08636
## Mean : 0.0000 Mean : 0.00000 Mean : 0.00000 Mean : 0.00000
## 3rd Qu.: 0.8542 3rd Qu.: 0.49120 3rd Qu.: 0.76526 3rd Qu.: 0.62715
## Max. : 1.6299 Max. : 0.76886 Max. : 2.39415 Max. : 3.40955
## landscape_mesh landscape_np landscape_shei landscape_siei
## Min. : -1.0155 Min. : -0.79501 Min. : -3.52743 Min. : -3.9765
## 1st Qu.: -0.6708 1st Qu.: -0.51403 1st Qu.: -0.60965 1st Qu.: -0.5383
## Median : -0.2930 Median : -0.29480 Median : 0.08809 Median : 0.2545
## Mean : 0.0000 Mean : 0.00000 Mean : 0.00000 Mean : 0.0000
## 3rd Qu.: 0.3122 3rd Qu.: 0.06182 3rd Qu.: 0.67378 3rd Qu.: 0.7550
## Max. : 7.4986 Max. : 5.73543 Max. : 2.64626 Max. : 1.7992
## cfi_site cfi_site_with_harvest cfi_site_with_vegedges
## Min. : -0.9601 Min. : -1.0134 Min. : -1.0053
## 1st Qu.: -0.6491 1st Qu.: -0.6678 1st Qu.: -0.6476
## Median : -0.3108 Median : -0.2978 Median : -0.3164
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: 0.2936 3rd Qu.: 0.3319 3rd Qu.: 0.3033
## Max. : 5.9740 Max. : 5.5758 Max. : 5.3407
```

Let's look at the correlation among this final set of candidate variables at their respective spatial scales:

```
data_final_corr <- data_final_scaled %>%

  select(9:ncol()) %>%

  corrr::correlate(., use = "pairwise.complete.obs",
                  method = "pearson",
                  diagonal = NA,
                  quiet = TRUE) %>%

  as_tibble() %>%

  mutate(across(everything(), ~ if_else(row_number() >= min(which(is.na(.))), NA, .))) %>%
```

```

select(rev(names(.))) %>%

relocate(term) %>%

pivot_longer(cols = -term, names_to = "var2", values_to = "corr") %>%

rename(var1 = term) %>%

arrange(desc(abs(corr))) %>%

drop_na(.)

```

```

## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(...)` .
## Caused by warning in `min()` :
## ! no non-missing arguments to min; returning Inf

```

```
head(data_final_corr, 15)
```

```

## # A tibble: 15 x 3
##   var1          var2          corr
##   <chr>        <chr>        <dbl>
## 1 cfi_site     cfi_site_with_vegedges  0.981
## 2 landscape_contag landscape_shei -0.965
## 3 nonanthro_ed landscape_np    0.957
## 4 landscape_shei landscape_siei  0.950
## 5 wells_active wells_total    0.944
## 6 seismic      nonanthro_ed    0.936
## 7 landscape_contag landscape_siei -0.931
## 8 nonanthro_cohesion landscape_np -0.930
## 9 seismic_lines_3D nonanthro_ed    0.916
## 10 nonanthro_ed nonanthro_cohesion -0.913
## 11 seismic      seismic_lines_3D    0.911
## 12 seismic_lines_3D landscape_np    0.907
## 13 seismic      nonanthro_cohesion -0.900
## 14 harvest_gt_15 harvest_total    0.896
## 15 seismic      landscape_np    0.894

```

Let's make a table for the Supporting Information to show high correlations.

```

covs_formatted <- read_csv("./tables/OSM_all_covariates_formatted_names.csv")

xxx <- data_final_corr %>%
  filter(var1 != "cfi_site_with_harvest" & var2 != "cfi_site_with_harvest",
         var1 != "cfi_site_with_vegedges" & var2 != "cfi_site_with_vegedges",
         var1 != "nonanthro_cai_mn" & var2 != "nonanthro_cai_mn",
         var1 != "nonanthro_tca" & var2 != "nonanthro_tca") %>%
  left_join(covs_formatted, by = c("var1" = "Covariate")) %>%
  mutate(var1 = PrettyName) %>%
  select(-PrettyName) %>%
  left_join(covs_formatted, by = c("var2" = "Covariate")) %>%
  mutate(var2 = PrettyName) %>%
  select(-PrettyName) %>%
  drop_na() %>%

```

```
filter(abs(corr)>0.50)

writexl::write_xlsx(xxx, "./tables/final_dataset_high_correlation.xlsx")
```

4. Fit models using the best-fit scales for each ‘category’ of variables.

nbinom2 allows the variance to increase with the square of the mean, which seemed to be the best fit to our data (vs nbinom1) in an initial exploration of AIC scores (huge drop in AICc)

```
m_null <- glmmTMB(squirrel ~ # Squirrel as response variable

  #lc_coniferous +
  #lc_mixedwood +
  #lc_broadleaf +
  #lc_shrubs +
  #lc_herbs +
  #fire_0_15 +

  (1|array/site), # Nested random effects for every model

  data = data_final_scaled, # Scaled data as the data
  family = nbinom2, # nbinom2 specification
  na.action = na.fail) # fail the model if there are NAs. Probably a data
  ↪ error.

# Not gonna annotate every model since it should be obvious by now.
m_fire <- glmmTMB(squirrel ~

  #lc_coniferous +
  #lc_mixedwood +
  #lc_broadleaf +
  #lc_shrubs +
  #lc_herbs +
  fire_0_15 +

  (1|array/site),

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

m_forest <- glmmTMB(squirrel ~

  lc_coniferous +
  lc_mixedwood +
  lc_broadleaf +
  #lc_shrubs +
  #lc_herbs +
  #fire_0_15 +

  (1|array/site),

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



```

m_forest_fire <- glmmTMB(squirrel ~

    lc_coniferous +
    lc_mixedwood +
    lc_broadleaf +
    #lc_shrubs +
    #lc_herbs +
    fire_0_15 +

    (1|array/site),

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

m_landcover <- glmmTMB(squirrel ~

    lc_coniferous +
    lc_mixedwood +
    lc_broadleaf +
    lc_shrubs +
    lc_wetland +
    lc_wetland_treed +
    #fire_0_15 +

    (1|array/site),

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

m_landcover_fire <- glmmTMB(squirrel ~

    lc_coniferous +
    lc_mixedwood +
    lc_broadleaf +
    lc_shrubs +
    lc_wetland +
    lc_wetland_treed +
    fire_0_15 +

    (1|array/site),

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

m_seis_wells <- glmmTMB(squirrel ~

    lc_coniferous +
    lc_mixedwood +
    lc_broadleaf +
    fire_0_15 +

    pipe_trans +
    seismic +
    wells_inactive +
    #harvest_total +

```

```

        #wells_active +
        #osm_industrial +
        #harvest_0_15 +
        #roads +

        (1|array/site),

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

m_harvest <- glmmTMB(squirrel ~
        lc_coniferous +
        lc_mixedwood +
        lc_broadleaf +
        fire_0_15 +

        #pipe_trans +
        #seismic +
        #wells_inactive +
        #harvest_total +
        #wells_active +
        #osm_industrial +
        harvest_0_15 +
        #roads +

        (1|array/site),

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

m_harvest_seis <- glmmTMB(squirrel ~
        lc_coniferous +
        lc_mixedwood +
        lc_broadleaf +
        fire_0_15 +

        pipe_trans +
        seismic +
        wells_inactive +
        #harvest_total +
        #wells_active +
        #osm_industrial +
        harvest_0_15 +
        #roads +

        (1|array/site),

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

m_active <- glmmTMB(squirrel ~
        lc_coniferous +
        lc_mixedwood +
        lc_broadleaf +
        fire_0_15 +

```

```

      #pipe_trans +
      #seismic +
      #wells_inactive +
      #harvest_total +
      wells_active +
      osm_industrial +
      #harvest_0_15 +
      #roads +

      (1|array/site),

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

m_active_harvest <- glmmTMB(squirrel ~
  lc_coniferous +
  lc_mixedwood +
  lc_broadleaf +
  fire_0_15 +

  #pipe_trans +
  #seismic +
  #wells_inactive +
  #harvest_total +
  wells_active +
  osm_industrial +
  harvest_0_15 +
  #roads +

  (1|array/site),

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

m_active_seis <- glmmTMB(squirrel ~
  lc_coniferous +
  lc_mixedwood +
  lc_broadleaf +
  fire_0_15 +

  pipe_trans +
  seismic +
  wells_inactive +
  #harvest_total +
  wells_active +
  osm_industrial +
  #harvest_0_15 +
  #roads +

  (1|array/site),
  data = data_final_scaled,
  family = nbinom2,
  na.action = na.fail)

m_active_seis_harvest <- glmmTMB(squirrel ~
  lc_coniferous +

```

```

lc_mixedwood +
lc_broadleaf +
fire_0_15 +

pipe_trans +
seismic +
wells_inactive +
#harvest_total +
wells_active +
osm_industrial +
harvest_0_15 +
#roads +

(1|array/site),

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

m_het <- glmmTMB(squirrel ~

# natural covariates
fire_0_15 +
lc_broadleaf +
lc_coniferous +
lc_mixedwood +

# configuration variables
landscape_shei +
#nonanthro_ed +
#landscape_mesh +

(1|array/site),

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

m_edge <- glmmTMB(squirrel ~

# natural covariates
fire_0_15 +
lc_broadleaf +
lc_coniferous +
lc_mixedwood +

# configuration variables
#landscape_shei +
nonanthro_ed +
#landscape_mesh +

(1|array/site),

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

m_cohesion <- glmmTMB(squirrel ~

```

```

      # natural covariates
      fire_0_15 +
      lc_broadleaf +
      lc_coniferous +
      lc_mixedwood +

      # configuration variables
      nonanthro_cohesion +
      #landscape_mesh +

      (1|array/site),

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

m_cohesion_mesh <- glmmTMB(squirrel ~

      # natural covariates
      fire_0_15 +
      lc_broadleaf +
      lc_coniferous +
      lc_mixedwood +

      # configuration variables
      nonanthro_cohesion +
      landscape_mesh +

      (1|array/site),

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

m_cohesion_shei <- glmmTMB(squirrel ~

      # natural covariates
      fire_0_15 +
      lc_broadleaf +
      lc_coniferous +
      lc_mixedwood +

      # configuration variables
      nonanthro_cohesion +
      #landscape_mesh +
      landscape_shei +

      (1|array/site),

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

m_conn <- glmmTMB(squirrel ~

      # natural covariates
      fire_0_15 +

```

```

        lc_broadleaf +
        lc_coniferous +
        lc_mixedwood +

        # configuration variables
        #landscape_shei +
        #nonanthro_ed +
        landscape_mesh +

        (1|array/site),

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

m_edge_conn <- glmmTMB(squirrel ~

        # natural covariates
        fire_0_15 +
        lc_broadleaf +
        lc_coniferous +
        lc_mixedwood +

        # configuration variables
        #landscape_shei +
        nonanthro_ed +
        landscape_mesh +

        (1|array/site),

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

m_het_conn <- glmmTMB(squirrel ~

        # natural covariates
        fire_0_15 +
        lc_broadleaf +
        lc_coniferous +
        lc_mixedwood +

        # configuration variables
        landscape_shei +
        #nonanthro_ed +
        landscape_mesh +

        (1|array/site),

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

m_het_edge <- glmmTMB(squirrel ~

        # natural covariates
        fire_0_15 +
        lc_broadleaf +
        lc_coniferous +

```

```

        lc_mixedwood +

        # configuration variables
        landscape_shei +
        nonanthro_ed +
        #landscape_mesh +

        (1|array/site),

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

m_config <- glmmTMB(squirrel ~

        # natural covariates
        fire_0_15 +
        lc_broadleaf +
        lc_coniferous +
        lc_mixedwood +

        # configuration variables
        landscape_shei +
        nonanthro_ed +
        landscape_mesh +

        (1|array/site),

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

m_edge_cfi <- glmmTMB(squirrel ~

        # natural covariates
        fire_0_15 +
        lc_broadleaf +
        lc_coniferous +
        lc_mixedwood +

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

        (1|array/site),

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

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)

```

5. Perform model selection

Let's gather all the models into a list and perform model selection:

```

#rm(m_nat_comp_config2)

# Pull all models from the global environment
models <- mget(ls(pattern = "m_"))

# PERform model selection on them
models_rank <- MuMIn::model.sel(models) %>%

  as.data.frame(.) %>%

  select(~`disp`((Int))`) %>%

  # Tidy up the disgusting glmmTMB names (thanks ChatGPT!!)
  rename_with(~ gsub("cond\\(|disp\\(|\\|\\|\\|\\(|", "", .)) %>%

  rename(intercept = `Int`) %>%

  mutate(model = rownames(.)) %>%

  # Model at the first row
  relocate(model)

```

models_rank

##	model	intercept	fire_0_15	lc_broadleaf
## m_edgeXcfi	m_edgeXcfi	-1.427925	0.15769889	-0.5184199
## m_seis_wells	m_seis_wells	-1.572581	0.11240507	-0.4778583
## m_harvest_seis	m_harvest_seis	-1.570776	0.09684067	-0.4733282
## m_edge_cfi	m_edge_cfi	-1.567200	0.16163124	-0.4928295
## m_edge_conn	m_edge_conn	-1.562466	0.19104506	-0.5133915
## m_active_seis	m_active_seis	-1.572573	0.13022726	-0.4723316
## m_config	m_config	-1.559354	0.20258102	-0.5380583
## m_active_seis_harvest	m_active_seis_harvest	-1.570856	0.11423316	-0.4679858
## m_het_edge	m_het_edge	-1.561443	0.17179761	-0.5412209
## m_edge	m_edge	-1.566267	0.14768820	-0.5036329
## m_cohesion_mesh	m_cohesion_mesh	-1.568183	0.21849236	-0.4991341

## m_cohesion	m_cohesion	-1.570700	0.17538729	-0.4934427
## m_cohesion_shei	m_cohesion_shei	-1.566806	0.19961846	-0.5268128
## m_conn	m_conn	-1.567658	0.28029701	-0.4736139
## m_forest_fire	m_forest_fire	-1.569886	0.23832582	-0.4704491
## m_het	m_het	-1.566800	0.26194637	-0.4950755
## m_harvest	m_harvest	-1.566950	0.20601272	-0.4674565
## m_het_conn	m_het_conn	-1.565877	0.29001804	-0.4889981
## m_active	m_active	-1.572968	0.27440936	-0.4499958
## m_active_harvest	m_active_harvest	-1.569942	0.23895698	-0.4442878
## m_forest	m_forest	-1.580889	NA	-0.5103000
## m_landcover_fire	m_landcover_fire	-1.570350	0.31937125	-0.7657364
## m_landcover	m_landcover	-1.579396	NA	-0.7986393
## m_fire	m_fire	-1.557730	0.28489807	NA
## m_null	m_null	-1.570733	NA	NA
##	lc_coniferous	lc_mixedwood	osm_industrial	wells_active
## m_edgeXcfi	-0.032283340	0.16358190	NA	NA
## m_seis_wells	-0.005779811	0.15468778	NA	NA
## m_harvest_seis	-0.003229019	0.15087108	NA	NA
## m_edge_cfi	-0.017969211	0.15372909	NA	NA
## m_edge_conn	-0.045979284	0.11772131	NA	NA
## m_active_seis	-0.003575806	0.15444343	0.1105531	-0.01413357
## m_config	-0.055956349	0.08846544	NA	NA
## m_active_seis_harvest	-0.001126694	0.15039267	0.1126439	-0.01882282
## m_het_edge	-0.049405069	0.10145326	NA	NA
## m_edge	-0.033103091	0.15055022	NA	NA
## m_cohesion_mesh	-0.046715422	0.12522553	NA	NA
## m_cohesion	-0.036190489	0.15209934	NA	NA
## m_cohesion_shei	-0.051553523	0.10730891	NA	NA
## m_conn	-0.021979643	0.15116866	NA	NA
## m_forest_fire	-0.015128002	0.17173150	NA	NA
## m_het	-0.025436180	0.13820635	NA	NA
## m_harvest	-0.013254231	0.16388419	NA	NA
## m_het_conn	-0.027890557	0.13257960	NA	NA
## m_active	0.003295678	0.18029519	0.1233570	0.09076456
## m_active_harvest	0.007694183	0.17198065	0.1265564	0.10381425
## m_forest	-0.021489508	0.15286291	NA	NA
## m_landcover_fire	-0.319353140	0.02963789	NA	NA
## m_landcover	-0.316265282	0.01933788	NA	NA
## m_fire	NA	NA	NA	NA
## m_null	NA	NA	NA	NA
##	harvest_0_15	pipe_trans	seismic	wells_inactive
## m_edgeXcfi	NA	NA	NA	NA
## m_seis_wells	NA	0.3052326	-0.4471425	0.005864648
## m_harvest_seis	0.07299016	0.3108955	-0.4382964	0.017820765
## m_edge_cfi	NA	NA	NA	NA
## m_edge_conn	NA	NA	NA	NA
## m_active_seis	NA	0.3015702	-0.4310483	-0.019989374
## m_config	NA	NA	NA	NA
## m_active_seis_harvest	0.07650757	0.3099517	-0.4226267	-0.005497010
## m_het_edge	NA	NA	NA	NA
## m_edge	NA	NA	NA	NA
## m_cohesion_mesh	NA	NA	NA	NA
## m_cohesion	NA	NA	NA	NA
## m_cohesion_shei	NA	NA	NA	NA

## m_conn	NA	NA	NA	NA
## m_forest_fire	NA	NA	NA	NA
## m_het	NA	NA	NA	NA
## m_harvest	0.10036954	NA	NA	NA
## m_het_conn	NA	NA	NA	NA
## m_active	NA	NA	NA	NA
## m_active_harvest	0.11765607	NA	NA	NA
## m_forest	NA	NA	NA	NA
## m_landcover_fire	NA	NA	NA	NA
## m_landcover	NA	NA	NA	NA
## m_fire	NA	NA	NA	NA
## m_null	NA	NA	NA	NA
##	nonanthro_cohesion	landscape_mesh	landscape_shei	
## m_edgeXcfi	NA	NA	NA	NA
## m_seis_wells	NA	NA	NA	NA
## m_harvest_seis	NA	NA	NA	NA
## m_edge_cfi	NA	NA	NA	NA
## m_edge_conn	NA	-0.2691175	NA	NA
## m_active_seis	NA	NA	NA	NA
## m_config	NA	-0.2368008	0.11685528	NA
## m_active_seis_harvest	NA	NA	NA	NA
## m_het_edge	NA	NA	0.17376761	NA
## m_edge	NA	NA	NA	NA
## m_cohesion_mesh	0.2292213	-0.2291634	NA	NA
## m_cohesion	0.2044128	NA	NA	NA
## m_cohesion_shei	0.2247754	NA	0.15869581	NA
## m_conn	NA	-0.1948194	NA	NA
## m_forest_fire	NA	NA	NA	NA
## m_het	NA	NA	0.12465232	NA
## m_harvest	NA	NA	NA	NA
## m_het_conn	NA	-0.1707704	0.07849967	NA
## m_active	NA	NA	NA	NA
## m_active_harvest	NA	NA	NA	NA
## m_forest	NA	NA	NA	NA
## m_landcover_fire	NA	NA	NA	NA
## m_landcover	NA	NA	NA	NA
## m_fire	NA	NA	NA	NA
## m_null	NA	NA	NA	NA
##	nonanthro_ed	cfi_site	cfi_site:nonanthro_ed	lc_shrubs
## m_edgeXcfi	-0.08883836	0.2954803	-0.2561767	NA
## m_seis_wells	NA	NA	NA	NA
## m_harvest_seis	NA	NA	NA	NA
## m_edge_cfi	-0.37852471	0.2660567	NA	NA
## m_edge_conn	-0.31100945	NA	NA	NA
## m_active_seis	NA	NA	NA	NA
## m_config	-0.32282048	NA	NA	NA
## m_active_seis_harvest	NA	NA	NA	NA
## m_het_edge	-0.28828172	NA	NA	NA
## m_edge	-0.26158022	NA	NA	NA
## m_cohesion_mesh	NA	NA	NA	NA
## m_cohesion	NA	NA	NA	NA
## m_cohesion_shei	NA	NA	NA	NA
## m_conn	NA	NA	NA	NA
## m_forest_fire	NA	NA	NA	NA

## m_het	NA	NA		NA	NA
## m_harvest	NA	NA		NA	NA
## m_het_conn	NA	NA		NA	NA
## m_active	NA	NA		NA	NA
## m_active_harvest	NA	NA		NA	NA
## m_forest	NA	NA		NA	NA
## m_landcover_fire	NA	NA		NA	-0.22584267
## m_landcover	NA	NA		NA	-0.01122732
## m_fire	NA	NA		NA	NA
## m_null	NA	NA		NA	NA
##	lc_wetland	lc_wetland_treed	df	logLik	AICc
## m_edgeXcfi	NA		NA 11	-4686.194	9394.442
## m_seis_wells	NA		NA 11	-4687.487	9397.028
## m_harvest_seis	NA		NA 12	-4687.250	9398.564
## m_edge_cfi	NA		NA 10	-4689.652	9399.348
## m_edge_conn	NA		NA 10	-4689.873	9399.790
## m_active_seis	NA		NA 13	-4686.866	9399.806
## m_config	NA		NA 11	-4689.408	9400.869
## m_active_seis_harvest	NA		NA 14	-4686.606	9401.297
## m_het_edge	NA		NA 10	-4691.011	9402.068
## m_edge	NA		NA 9	-4692.103	9402.242
## m_cohesion_mesh	NA		NA 10	-4691.417	9402.879
## m_cohesion	NA		NA 9	-4693.068	9404.173
## m_cohesion_shei	NA		NA 10	-4692.160	9404.365
## m_conn	NA		NA 9	-4693.631	9405.299
## m_forest_fire	NA		NA 8	-4694.841	9405.711
## m_het	NA		NA 9	-4694.274	9406.585
## m_harvest	NA		NA 9	-4694.382	9406.800
## m_het_conn	NA		NA 10	-4693.424	9406.893
## m_active	NA		NA 10	-4693.488	9407.021
## m_active_harvest	NA		NA 11	-4692.861	9407.775
## m_forest	NA		NA 7	-4697.320	9408.663
## m_landcover_fire	-0.1640682	-0.2667872	11	-4693.915	9409.883
## m_landcover	-0.1353234	-0.3071780	10	-4695.846	9411.737
## m_fire	NA		NA 5	-4704.376	9418.764
## m_null	NA		NA 4	-4707.890	9423.789
##	delta	weight			
## m_edgeXcfi	0.000000	5.591327e-01			
## m_seis_wells	2.586045	1.534484e-01			
## m_harvest_seis	4.121863	7.119733e-02			
## m_edge_cfi	4.906134	4.810182e-02			
## m_edge_conn	5.348357	3.855972e-02			
## m_active_seis	5.364269	3.825416e-02			
## m_config	6.427454	2.248076e-02			
## m_active_seis_harvest	6.855755	1.814707e-02			
## m_het_edge	7.625992	1.234673e-02			
## m_edge	7.800059	1.131758e-02			
## m_cohesion_mesh	8.437427	8.229071e-03			
## m_cohesion	9.731446	4.308820e-03			
## m_cohesion_shei	9.923648	3.914011e-03			
## m_conn	10.857026	2.454379e-03			
## m_forest_fire	11.268831	1.997650e-03			
## m_het	12.142867	1.290402e-03			
## m_harvest	12.358630	1.158437e-03			

```
## m_het_conn          12.450986 1.106159e-03
## m_active            12.578990 1.037581e-03
## m_active_harvest    13.333557 7.114916e-04
## m_forest            14.221449 4.564220e-04
## m_landcover_fire    15.441196 2.480286e-04
## m_landcover         17.295472 9.814110e-05
## m_fire              24.322243 2.924200e-06
## m_null              29.346857 2.370969e-07
```

```
# Fetch the top model
top_model <- get(ls(pattern = rownames(models_rank[1]))[1])
```

```
## Warning in grep(pattern, all.names, value = TRUE): argument 'pattern' has
## length > 1 and only the first element will be used
```

```
# Or explore any model manually using the same code...
top_model <- m_edgeXcfi
```

6. Model inspection and validation

6.0. A bit of set up for tables and figures

Let's read in the 'pretty' names of the covariates first. These will be what we use in figures and tables.

```
covs_formatted <- 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.
```

6.1. Check out a general summary:

```
summary(top_model)
```

```
## 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
```

6.2. Pseudo-R2

```
performance::r2(top_model)
```

```
## # R2 for Mixed Models
##
##   Conditional R2: 0.744
##   Marginal R2: 0.099
```

6.3. Check dispersion:

```
performance::check_overdispersion(top_model)
```

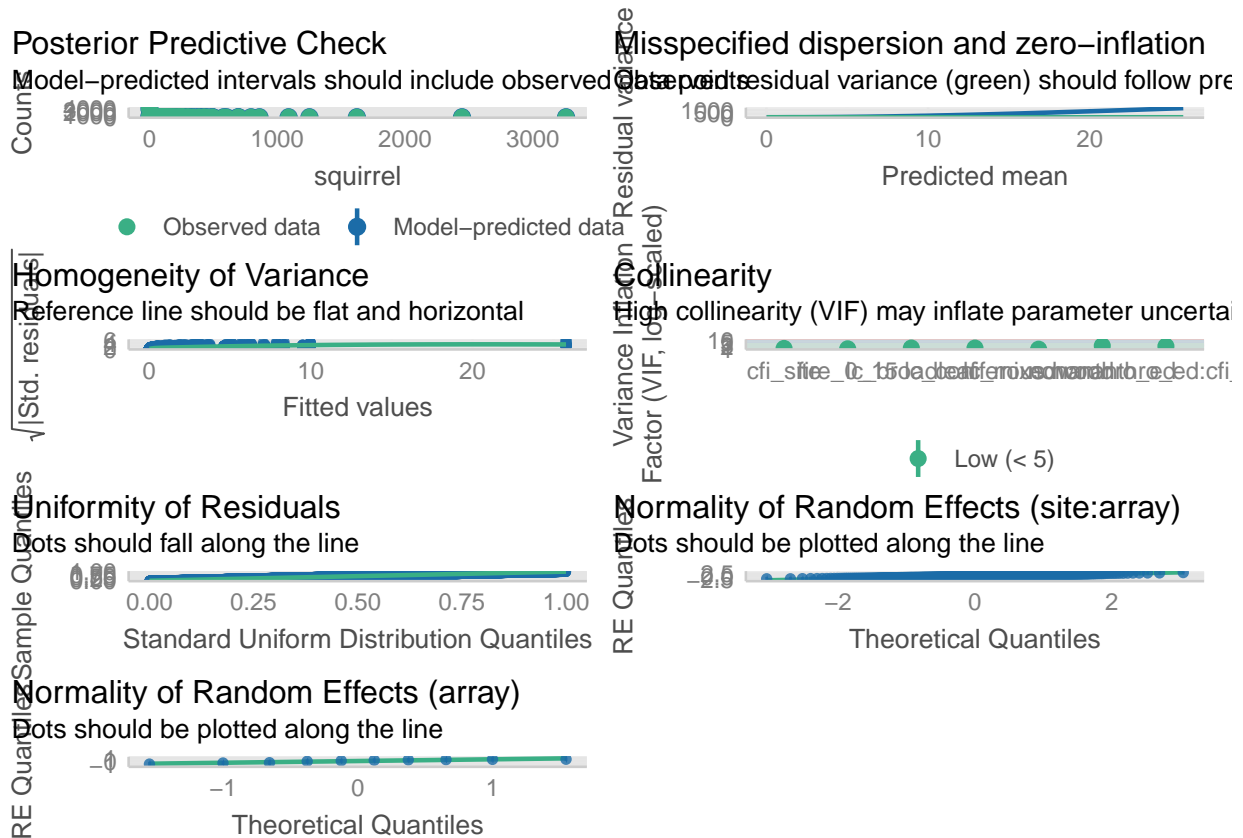
```
## # Overdispersion test
##
##   dispersion ratio = 0.179
##   p-value = 0.176
##
## No overdispersion detected.
```

Previous notes from fitting proportional binomial models: “The model is overdispersed. Variance is much higher than the mean which violates an assumption of the bernoulli distribution. A zero inflated bernoulli (ZIB) might be more appropriate. Re-binning the data into monthly occurrence frequency might also help eliminate a lot of the zeros that are (likely) the cause of this.”

6.4. Check other diagnostics including VIFs.

```
performance::check_model(top_model)
```

```
## `check_outliers()` does not yet support models of class `glmmTMB`.
```



```
ggsave("./figures/top_model_neg_binomial_diagnostics.png", height = 12, width = 9)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

6.5. Variance inflation factors:

```
# vif from the car package
performance::check_collinearity(top_model) %>%

as.data.frame(.) %>%

# Add the pretty names
```

```

left_join(covs_formatted, by = c("Term" = "Covariate")) %>%

# Manual fix for a couple of them
mutate(PrettyName = case_when(
  Term == "nonanthro_ed:cfi_site" ~ "Edge Density × CFI",
  TRUE ~ PrettyName)) %>%

# plot with ggplot
ggplot(aes(x = reorder(PrettyName, VIF), # reorders from smallest VIF to largest (not sure I want
  ↪ like this)
  y = VIF)) +

  geom_hline(yintercept = 3, linetype='dashed', col = 'firebrick') +

  # Error bars behind the points
  geom_errorbar(aes(ymin = VIF_CI_low, ymax = VIF_CI_high), width = 0.1) + # Error bars

  # plot as points
  geom_point(size = 2.5) +

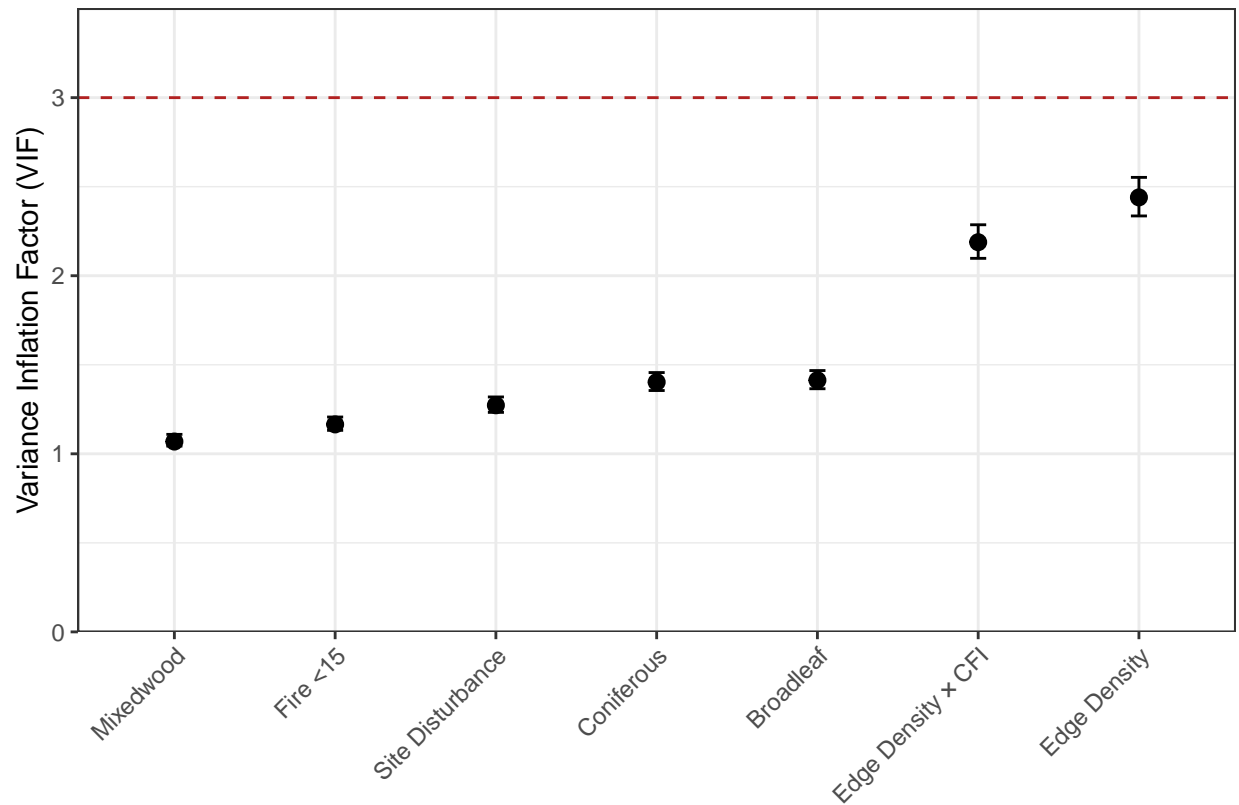
  # add labels
  labs(x = '',
    y = 'Variance Inflation Factor (VIF)') +

  scale_y_continuous(limits = c(0, 3.2), expand = expansion(add = c(0, 0.3))) +

  # set theme
  theme_bw() +

  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels by 45 degrees

```



```
ggsave("./figures/top_model_neg_binomial_VIF.png", dpi=500, width = 5, height = 3.5)
```

6.6. Likelihood ratio test for random effect structure:

```
m_edgeXcfi_noraneef <- 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_noraneef)
```

```
## Family: nbinom2 ( log )
## Formula:
## squirrel ~ fire_0_15 + lc_broadleaf + lc_coniferous + lc_mixedwood +
##          nonanthro_ed + cfi_site + nonanthro_ed * cfi_site
## Data: data_final_scaled
##
##      AIC      BIC   logLik deviance df.resid
## 10710.8 10769.4 -5346.4 10692.8     4909
##
##
## Dispersion parameter for nbinom2 family (): 0.186
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.12680    0.04487  -2.826  0.00471 **
## fire_0_15       0.21237    0.03793   5.600 2.15e-08 ***
## lc_broadleaf   -0.54628    0.05178 -10.550 < 2e-16 ***
## lc_coniferous  -0.22706    0.04452  -5.101 3.39e-07 ***
## lc_mixedwood   -0.06637    0.04303  -1.542  0.12296
## nonanthro_ed   -0.16580    0.06878  -2.411  0.01593 *
## cfi_site        0.35124    0.05050   6.955 3.52e-12 ***
## nonanthro_ed:cfi_site -0.22707    0.04611  -4.924 8.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m_edgeXcfi_noraneef, m_edgeXcfi)
```

```
## Data: data_final_scaled
## Models:
## m_edgeXcfi_noraneef: squirrel ~ fire_0_15 + lc_broadleaf + lc_coniferous + lc_mixedwood + , zi=-0, disp=1
## m_edgeXcfi_noraneef:          nonanthro_ed + cfi_site + nonanthro_ed * cfi_site, zi=-0, disp=~1
## m_edgeXcfi: squirrel ~ fire_0_15 + lc_broadleaf + lc_coniferous + lc_mixedwood + , zi=-0, disp=~1
## m_edgeXcfi:          nonanthro_ed + cfi_site + nonanthro_ed * cfi_site + (1 | , zi=-0, disp=~1
## m_edgeXcfi:          array/site), zi=-0, disp=~1
##              Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m_edgeXcfi_noraneef  9 10710.8 10769.4 -5346.4 10692.8
## m_edgeXcfi          11  9394.4  9465.9 -4686.2  9372.4 1320.5      2 < 2.2e-16
##
## m_edgeXcfi_noraneef
## m_edgeXcfi          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7. Interpret results

7.1. Model summary

```
summary(top_model)
```

```
## 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
```

7.2. Odds ratio plots for the conditional model:

```
# Custom order for the x-axis covariates
custom_order <- c(
  "Broadleaf",
  "Mixedwood",
  "Coniferous",
  "Fire <15",
  "Site Disturbance",
  "Edge Density",
  "Edge Density × \nSite Disturbance"
)
```

```

odds_fig_top <-

confint(top_model) %>%

  as_tibble(rownames = "term") %>%

  # Rename the columns manually
  set_names(c('term', 'lower', 'upper', 'estimate')) %>%

  # Go from log odds back to odds
  mutate(across(c('lower', 'upper', 'estimate'), exp)) %>%

  # Remove things we dont wanna plot
  filter(term %nin% c('(Intercept)', 'Std.Dev.(Intercept)|array',
    ↪ 'Std.Dev.(Intercept)|site:array')) %>%

  # Pretty names
  left_join(covs_formatted, by = c("term" = "Covariate")) %>%

  # Fix a couple manually
  mutate(PrettyName = case_when(
    term == "nonanthro_ed:cfi_site" ~ "Edge Density × \nSite Disturbance",
    TRUE ~ PrettyName)) %>%

  # Apply the custom order for covariates
  mutate(PrettyName = factor(PrettyName, levels = custom_order)) %>%

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

    geom_hline(yintercept = 1, linetype='dashed', col = 'grey20') +

    # Error bars behind the points
    geom_errorbar(aes(x = PrettyName, ymin = lower, ymax = upper), width = 0.1) + # Error bars

    # plot as points
    geom_point(aes(y = estimate), size = 2.5) +

    scale_y_continuous(limits = c(0.4, 1.8)) +

    # add labels
    labs(x = '',
      y = 'Odds Ratio (standardized)') +

    annotate("text",
      x = 0.5,          # Left edge of panel
      y = 1.8,          # Top edge of panel
      label = " ",
      hjust = 0,         # Left-aligned
      vjust = 1,         # Top-aligned
      size = 5) +        # Adjust size as needed

    # set theme
    theme_bw() +

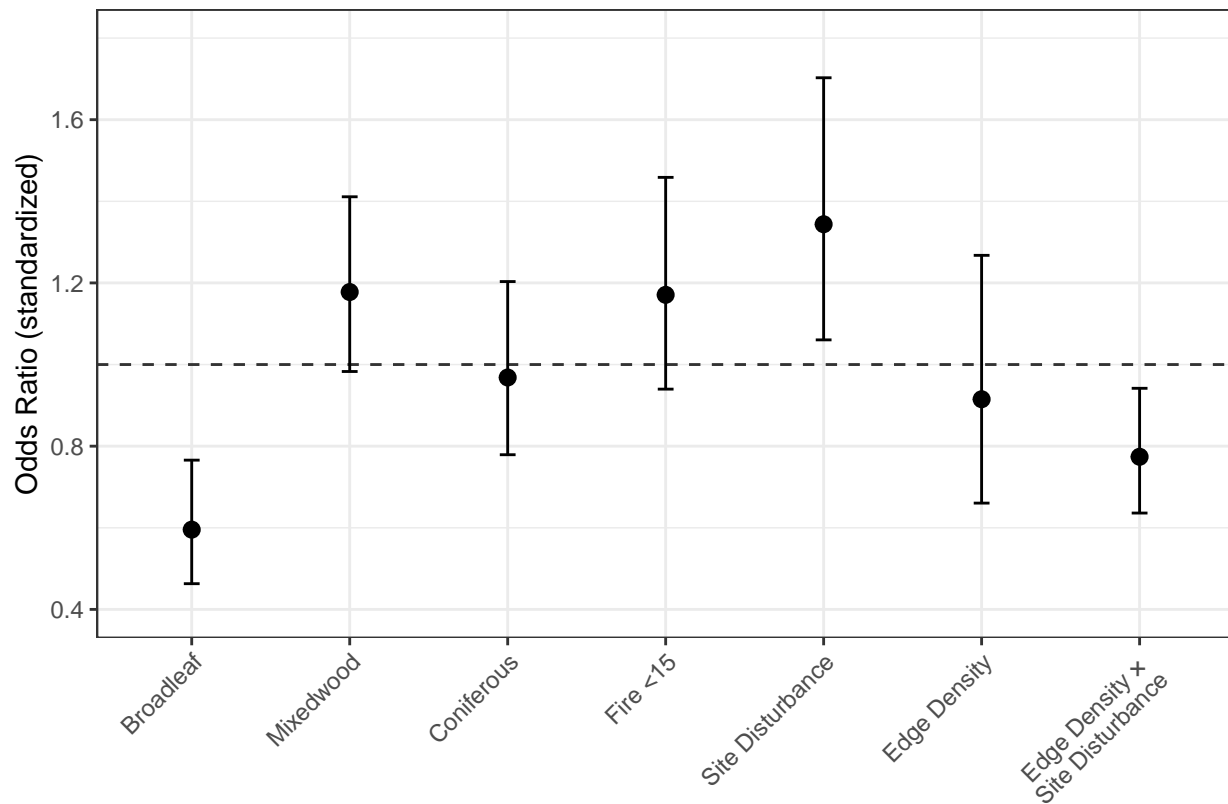
    theme(plot.title = element_text(face = "bold")) +

    theme(axis.text.x = element_text(angle = 45, hjust = 1)) #+ # Rotate x-axis labels by 45
    ↪ degrees

```

```
#add_phylopic(
#  uuid = get_uuid(name = "Sciurus vulgaris"),
#  x = 6.6,
#  y = 1.6,
#  height = 0.22
# )
```

odds_fig_top



```
ggsave(odds_fig_top, file = "./figures/top_model_neg_binomial_odds_ratios.png", width = 5, height =
  3.5)
```

7.3. Odds ratio plots for the second-place model

```
# Custom order for the x-axis covariates
custom_order <- c(
  "Broadleaf",
  "Mixedwood",
  "Coniferous",
  "Fire <15",
  "Inactive Well Sites",
  "Seismic Lines",
  "Pipelines & \nTransmission Lines"
```

```

)

odds_fig_second <-

# Exact same code as before, not annotating because I'm lazy.
confint(m_seis_wells) %>%

  as_tibble(rownames = "term") %>%

  set_names(c('term', 'lower', 'upper', 'estimate')) %>%

  mutate(across(c('lower', 'upper', 'estimate'), exp)) %>%

  filter(term %nin% c('(Intercept)', 'Std.Dev.(Intercept)|array',
    ↪ 'Std.Dev.(Intercept)|site:array')) %>%

  left_join(covs_formatted, by = c("term" = "Covariate")) %>%

  mutate(PrettyName = case_when(
    term == "nonanthro_ed:cfi_site" ~ "Edge Density × CFI",
    term == "pipe_trans" ~ "Pipelines & \nTransmission Lines",
    TRUE ~ PrettyName)) %>%

  mutate(PrettyName = factor(PrettyName, levels = custom_order)) %>%

  ggplot(., aes(x = PrettyName)) +

    geom_hline(yintercept = 1, linetype='dashed', col = 'grey20') +

    geom_errorbar(aes(x = PrettyName, ymin = lower, ymax = upper), width = 0.1) + # Error bars

    # add labels
    labs(x = '',
         y = '') +

    # plot as bars
    geom_point(aes(y = estimate), size = 2.5) +

    scale_y_continuous(limits = c(0.4, 1.8)) +

    # set theme
    theme_bw() +

    theme(axis.text.x = element_text(angle = 45, hjust = 1)) + # Rotate x-axis labels by 45
    ↪ degrees

    annotate("text",
      x = 0.5,          # Left edge of panel
      y = 1.8,          # Top edge of panel
      label = " ΔAICc=2.59",
      hjust = 0,        # Left-aligned
      vjust = 1,        # Top-aligned
      size = 5)         # Adjust size as needed

#add_phylopic(
#  uuid = get_uuid(name = "Sciurus vulgaris"),
#  x = 6.6,

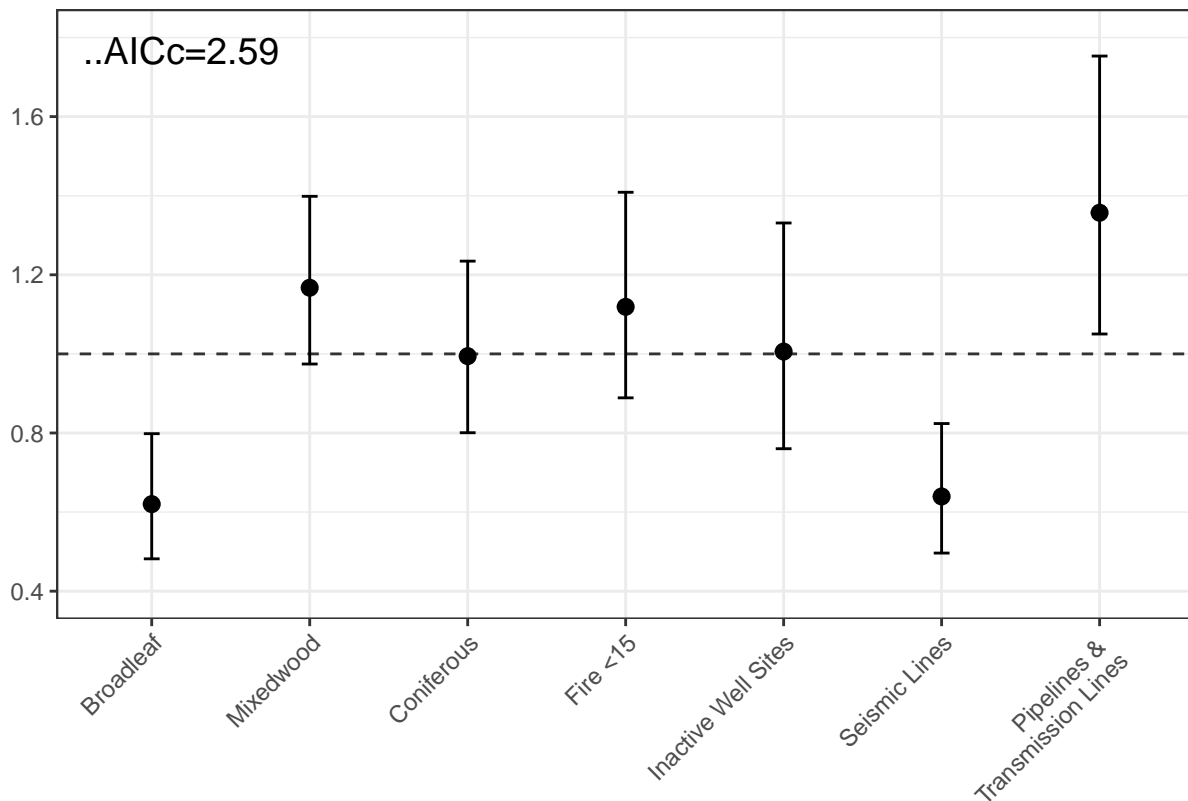
```

```
# y = 1.6,
# height = 0.22
# )
```

```
odds_fig_second
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ΔAICc=2.59' in 'mbscsToSbcs': dot substituted for <ce>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ΔAICc=2.59' in 'mbscsToSbcs': dot substituted for <94>
```



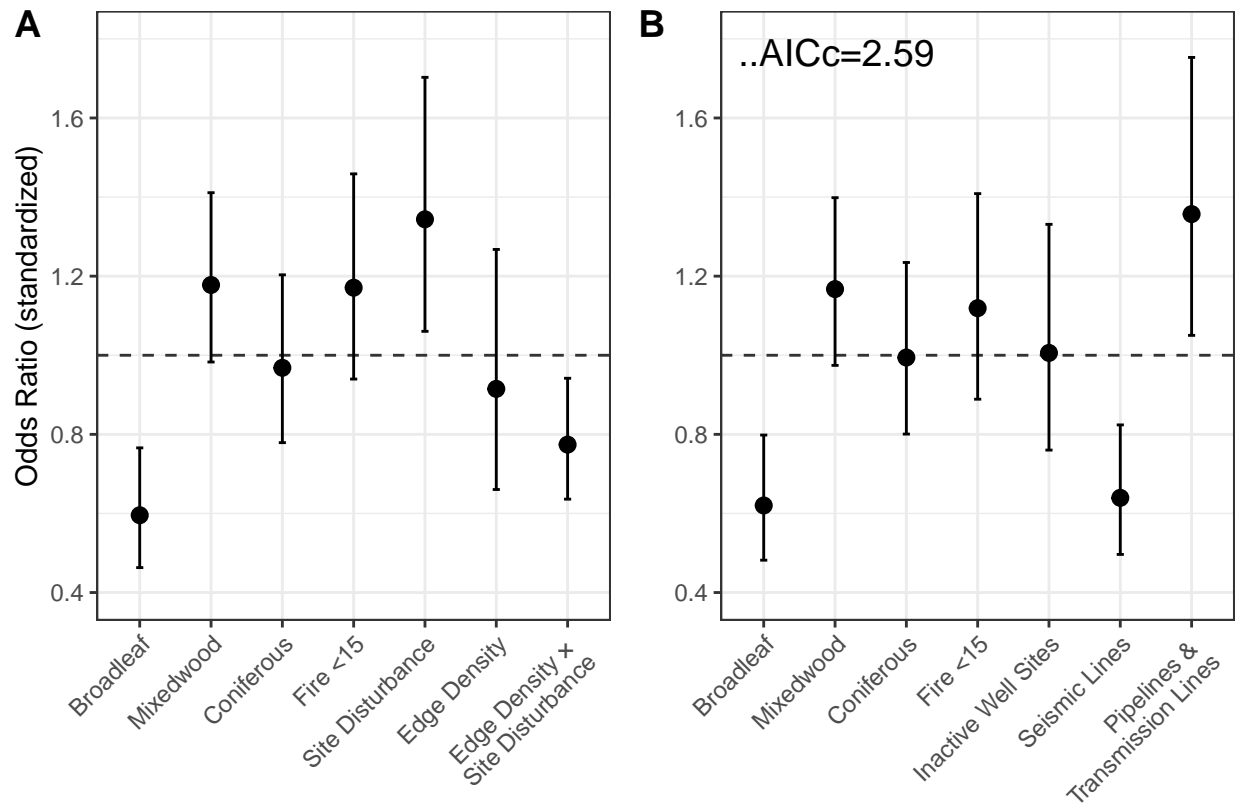
```
ggsave(odds_fig_second, file= "../figures/second_place_model_neg_binomial_odds_ratios.png", width =
  ↪ 5, height = 3.5)
```

7.4. Export the combined top and second-ranking model figures in one!

```
ggpubr::ggarrange(odds_fig_top, odds_fig_second,
  ncol = 2,
  nrow = 1,
  align = "hv",
  labels = c("A", "B") # Add plot labels
)
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ΔAICc=2.59' in 'mbcsToSbcs': dot substituted for <ce>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ΔAICc=2.59' in 'mbcsToSbcs': dot substituted for <94>
```



```
ggsave("./figures/odds_ratio_top_model_second_model.png", width = 7, height = 4.2, dpi = 700)
```

8. Plot conditional effects

We will use our scaling summary here to backtransform the data into ecologically meaningful values. This is a bit hacky but it works.

8.1. Decid:

```
# Get just the row that contains the mean and SD for backtransforming...
# Hacky I KNOW don't judge me we had 2 weeks to do this
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == nat_buffer,
         variable == "lc_broadleaf")
```

```

# Useggeffects to predict at all values of lc_broadleaf in the dataset
f_decid <- ggeffects::ggpredict(top_model, terms = c("lc_broadleaf [all]")) %>%

  as.data.frame() %>%

  # Apply the backtransform to the values for the x axis, predictions unchanged :)
  mutate(x = x * backtransform$sd + backtransform$mean) %>%

  ggplot(., aes(x = x, y = predicted)) +

    geom_line(size = 0.5) + # Set lines to black

    # Confidence ribbon in front of the line
    geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.4, fill = 'forestgreen') +

    # Show unscaled data points for just the buffer distances that we want.
    geom_rug(data = data%>%filter(buffer_dist==nat_buffer), aes(x = lc_broadleaf), color =
  ↪ "grey20", inherit.aes = FALSE) +

    labs(x = "Broadleaf",
         y = NULL) +

    scale_x_continuous(expand = c(0, 0)) + # Remove left blank space

    theme_bw() +

    theme(panel.grid = element_blank()) #+

```

```

## You are calculating adjusted predictions on the population-level (i.e.
## `type = "fixed"`) for a *generalized* linear mixed model.
## This may produce biased estimates due to Jensen's inequality. Consider
## setting `bias_correction = TRUE` to correct for this bias.
## See also the documentation of the `bias_correction` argument.

```

```

# Squirrel icon for fun
#add_phylopic(
#  uuid = get_uuid(name = "Sciurus vulgaris"),
#  x = 0.89,
#  y = 0.46,
#  height = 0.11
# )

ggsave(f_decid, file = "./figures/top_model_neg_binomial_broadleaf.png", dpi=400, width = 6, height
↪ = 4)

```

8.2. Conifer:

```

# Get just the row that contains the mean and SD for backtransforming...
# Yep, still hacky - but fast and it works
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == nat_buffer,
         variable == "lc_coniferous")

# Useggeffects to predict at all values of lc_coniferous in the dataset

```



```
f_conifer <- ggeffects::ggpredict(top_model, terms = c("lc_coniferous [all]")) %>%

  as.data.frame() %>%

  # Backtransform the x-axis values to their original scale (predictions stay in response scale)
  mutate(x = x * backtransform$sd + backtransform$mean) %>%

  # Plot predictions with ggplot2
  ggplot(., aes(x = x, y = predicted)) +

    geom_line(size = 0.5) + # Black line for model prediction

    # Add a confidence ribbon (plotted underneath the line)
    geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.4, fill = 'forestgreen') +

    # Add raw data rug for just the buffer distance of interest (in unscaled space)
    geom_rug(data = data %>% filter(buffer_dist == nat_buffer),
             aes(x = lc_coniferous),
             color = "grey20", inherit.aes = FALSE) +

    # Axis labels
    labs(x = "Coniferous",
         y = NULL) +

    # Remove empty space on left side of x axis
    scale_x_continuous(expand = c(0, 0)) +

    # Minimal clean theme
    theme_bw() +

    theme(panel.grid = element_blank())# +

    # Squirrel icon for fun (lower left corner this time)
    #add_phylopic(
    #  uuid = get_uuid(name = "Sciurus vulgaris"),
    #  x = 0.08,
    #  y = 0.4,
    #  height = 0.06
    #)

# Save the figure
ggsave(f_conifer, file = "./figures/top_model_neg_binomial_conifer.png", dpi = 400, width = 6,
  ↪ height = 4)
```

8.3. Mixedwood:

```
# Get just the row that contains the mean and SD for backtransforming...
# Still using our quick-and-dirty approach by filtering to the variable and buffer of interest
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == nat_buffer,
         variable == "lc_mixedwood")

# Use ggeffects to predict model outcomes across all observed lc_mixedwood values (scaled units)
f_mixedwood <- ggeffects::ggpredict(top_model, terms = c("lc_mixedwood [all]")) %>%

  as.data.frame() %>%
```

```

# Backtransform x values to original scale (predicted values are already in response scale)
mutate(x = x * backtransform$sd + backtransform$mean) %>%

# Start the plot
ggplot(., aes(x = x, y = predicted)) +

  geom_line(size = 0.5) + # Model prediction line (thin, black)

  # Confidence interval ribbon under the line
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.4, fill = 'forestgreen') +

  # Rug plot to show actual unscaled values from the selected buffer
  geom_rug(data = data %>% filter(buffer_dist == nat_buffer),
    aes(x = lc_mixedwood),
    color = "grey20", inherit.aes = FALSE) +

  # Axis labels
  labs(x = "Mixedwood",
    y = NULL) +

  # Tighten up left side of the x-axis
  scale_x_continuous(expand = c(0, 0)) +

  # Clean and minimal styling
  theme_bw() +
  theme(panel.grid = element_blank()) #+

  # Icon again! Now in the upper-left corner for variety
  #add_phylopic(
  #  uuid = get_uuid(name = "Sciurus vulgaris"),
  #  x = 0.06,
  #  y = 1.4,
  #  height = 0.28
  #)

# Save to file
ggsave(f_mixedwood, file = "./figures/top_model_neg_binomial_mixedwood.png", dpi = 400, width = 6,
  ↪ height = 4)

```

8.4. Recent fire

```

# Get just the row that contains the mean and SD for backtransforming...
# Same trick, different variable (recent fire in this case)
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == nat_buffer,
    variable == "fire_0_15")

# Use ggeffects to predict at all observed values of fire_0_15 (in scaled units)
f_fire <- ggeffects::ggpredict(top_model, terms = c("fire_0_15 [all]")) %>%

  as.data.frame() %>%

  # Backtransform x-axis to original scale, leave predictions as-is
  mutate(x = x * backtransform$sd + backtransform$mean) %>%

```

```

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

  geom_line(size = 0.5) + # Prediction line (black, thin)

  # Confidence interval ribbon in orange
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.4, fill = 'forestgreen') +

  # Add raw data rug for this buffer distance only (unscaled)
  geom_rug(data = data %>% filter(buffer_dist == nat_buffer),
    aes(x = fire_0_15),
    color = "grey20", inherit.aes = FALSE) +

  # Axis labels
  labs(x = "Fire (0-15 years)",
    y = NULL) +

  # No extra white space on x-axis
  scale_x_continuous(expand = c(0, 0)) +

  # Clean theme with grid off
  theme_bw() +
  theme(panel.grid = element_blank())

  # Optional: squirrel icon, currently disabled
  # add_phylopic(
  #   uid = get_uid(name = "Sciurus vulgaris"),
  #   x = 0.08,
  #   y = 1.32,
  #   height = 0.26
  # )

# Save the plot
ggsave(f_fire, file = "./figures/top_model_neg_binomial_fire.png", dpi = 400, width = 6, height =
↵ 4)

```

8.5. Cumulative site disturbance:

```

# Get just the row that contains the mean and SD for backtransforming...
# This time for cumulative site disturbance (cfi_site), using the config buffer (might need to
↵ change to comp buffer later)
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == config_buffer,
    variable == "cfi_site")

# Use ggeffects to predict across all observed values of cfi_site (scaled)
f_cfi <- ggeffects::ggpredict(top_model, terms = c("cfi_site [all]")) %>%

  as.data.frame() %>%

  # Backtransform x-axis values to original scale
  mutate(x = x * backtransform$sd + backtransform$mean) %>%

  # Begin plot
  ggplot(., aes(x = x, y = predicted)) +

```

```

geom_line(size = 0.5) + # Black prediction line

# Confidence ribbon (red tone for disturbance)
geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.4, fill = 'goldenrod3') +

# Add raw data rug for this buffer distance only
geom_rug(data = data %>% filter(buffer_dist == config_buffer),
         aes(x = cfi_site),
         color = "grey20", inherit.aes = FALSE) +

# Axis labels
labs(x = "Site disturbance",
     y = NULL) +

# Remove padding on left of x-axis
scale_x_continuous(expand = c(0, 0), limits = c(0, 0.302)) +

scale_y_continuous(limits = c(0, 1.8)) +

# Minimalist, clean plot theme
theme_bw() +
theme(panel.grid = element_blank())

# Optional squirrel icon (commented out for now)
# add_phylopic(
#   uuid = get_uuid(name = "Sciurus vulgaris"),
#   x = 0.04,
#   y = 5.6,
#   height = 1.2
# )

# Save the plot
ggsave(f_cfi, file = "./figures/top_model_neg_binomial_cfi.png", dpi = 400, width = 6, height = 4)

```

```

## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

8.6. Edge density:

```

# Get just the row that contains the mean and SD for backtransforming...
# This time for cumulative site disturbance (cfi_site), using the config buffer (might need to
# ↪ change to comp buffer later)
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == config_buffer,
         variable == "nonanthro_ed")

# Use ggeffects to predict across all observed values of cfi_site (scaled)
f_edge <- ggeffects::ggpredict(top_model, terms = c("nonanthro_ed [all]")) %>%

as.data.frame() %>%

# Backtransform x-axis values to original scale
mutate(x = x * backtransform$sd + backtransform$mean) %>%

# Begin plot

```

```

ggplot(., aes(x = x, y = predicted)) +

  geom_line(size = 0.5) + # Black prediction line

  # Confidence ribbon (red tone for disturbance)
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.48, fill = 'darkblue') +

  # Add raw data rug for this buffer distance only
  geom_rug(data = data %>% filter(buffer_dist == config_buffer),
    aes(x = nonanthro_ed),
    color = "grey20", inherit.aes = FALSE) +

  # Axis labels
  labs(x = "Edge density",
    y = NULL) +

  # Remove padding on left of x-axis
  scale_x_continuous(expand = c(0, 0), limits = c(2, 202.85)) +

  scale_y_continuous(limits = c(0, 0.48)) +

  # Minimalist, clean plot theme
  theme_bw() +
  theme(panel.grid = element_blank())

  # Optional squirrel icon (commented out for now)
  # add_phylopic(
  #   uid = get_uid(name = "Sciurus vulgaris"),
  #   x = 0.04,
  #   y = 5.6,
  #   height = 1.2
  # )

# Save the plot
ggsave(f_edge, file = "./figures/top_model_neg_binomial_edge.png", dpi = 400, width = 6, height =
↵ 4)

```

```

## Warning: Removed 37 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

8.6. Edge x CFI interaction

```

# Get just the row that contains the mean and SD for backtransforming...
# This time for non-anthropogenic edge density (nonanthro_ed), using config buffer
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == config_buffer,
    variable == "nonanthro_ed")

# Predict squirrel occurrence across full range of nonanthro_ed
# for two quantiles (20th and 89th) of cumulative site disturbance (cfi_site)
f_edge_cfi <- ggeffects::ggpredict(

  top_model,
  terms = c(
    "nonanthro_ed [all]",

```

```

paste0("cfi_site [",
       quantile(data_final_scaled$cfi_site, 0.89), ", ",
       quantile(data_final_scaled$cfi_site, 0.20), "]" )
)
) %>%

as.data.frame() %>%

# Backtransform x values from scaled to original units (m/ha)
mutate(x = x * backtransform$sd + backtransform$mean) %>%

# Plot predictions and confidence ribbons, colored by cfi quantile group
ggplot(., aes(x = x, y = predicted, color = group)) +

  geom_line(size = 0.5) + # Thin black line for predictions

  # Confidence interval ribbons, filled by group
  geom_ribbon(
    aes(ymin = conf.low, ymax = conf.high, fill = group),
    alpha = 0.4, color = NA
  ) +

  # Manually assign colors to line and fill for visual distinction
  scale_color_manual(
    values = c('grey20', 'grey20'), # Lines stay neutral
    labels = c( # Hacky way to get the labels to the unscaled values
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$cfi_site, 0.20), 3)),
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$cfi_site, 0.89), 3))
    )
  ) +
  scale_fill_manual(
    values = c('darkblue', 'darkred'), # Fill colors for clarity
    labels = c(
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$cfi_site, 0.20), 3)),
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$cfi_site, 0.89), 3))
    )
  ) +

# OR JUST LABEL THEM AS QUANTILES

  # Manually assign colors to line and fill for visual distinction
  scale_color_manual(
    values = c('grey20', 'grey20'), # Lines stay neutral
    labels = c("10% quantile", "90% quantile")
  ) +
  scale_fill_manual(
    values = c('dodgerblue2', 'darkred'), # Fill colors for clarity
    labels = c("10% quantile", "90% quantile")
  ) +

# Rug plot shows actual values of edge density in the unscaled data
geom_rug(
  data = data %>% filter(buffer_dist == config_buffer),
  aes(x = nonanthro_ed),
  alpha = 0.5,
  color = "grey20",
  inherit.aes = FALSE
) +

```

```

# Axes and legend
labs(
  x = "Anthropogenic edge density (m per Ha)",
  y = "Expected count",
  color = "Site \ndisturbance",
  fill = "Site \ndisturbance"
) +

# Trim x and y axis range to what's observed and relevant
scale_x_continuous(limits = c(2, 202.85), expand = c(0, 0)) +
scale_y_continuous(limits = c(0, 1.1), expand = c(0, 0)) +

# Clean, minimal theme
theme_bw() +
theme(panel.grid = element_blank()) +
theme(
  legend.position = "right",
  legend.margin = margin(0, 0, 0, 0),
  legend.box.margin = margin(0, 0, 0, -5), # Pull legend closer to plot
  plot.margin = margin(5, 5, 5, 5)        # Optional: adjust plot margins
) +

# Optional: add a squirrel icon again for flair :)
add_phylopic(
  uuid = get_uuid(name = "Sciurus vulgaris"),
  x = 180,
  y = 0.95,
  height = 0.23
)

```

```

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

```

```

# Save it!
ggsave(f_edge_cfi, file = "./figures/top_model_edge_cfi_interaction.png", dpi = 400, width = 7,
  ↪ height = 4)

```

```

## Warning: Removed 74 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

8.7. CFI x Edge interaction

Do edges change the effect of site disturbance? This is a sort of backwards interpretation but I'm curious nonetheless.

```

# Backtransform helper: grab mean and SD for cumulative site disturbance (cfi_site)
backtransform <- data_scaled_summary %>%
  filter(buffer_dist == config_buffer,
    variable == "cfi_site")

# Predict squirrel occurrence across full range of cfi_site,
# at two values of edge density (nonanthro_ed)

```

```

f_cfi_edge <- ggeffects::ggpredict(

  top_model,
  terms = c(
    "cfi_site [all]",
    paste0("nonanthro_ed [",
      quantile(data_final_scaled$cfi_site, 0.89), ", ",
      quantile(data_final_scaled$cfi_site, 0.20), "]" )
  )
) %>%

as.data.frame() %>%

# Backtransform x (cfi_site) from scaled units to proportion
mutate(x = x * backtransform$sd + backtransform$mean) %>%

ggplot(., aes(x = x, y = predicted, color = group)) +

  # Prediction lines
  geom_line(size = 0.5) +

  # Confidence bands by group (edge density levels)
  geom_ribbon(
    aes(ymin = conf.low, ymax = conf.high, fill = group),
    alpha = 0.4, color = NA
  ) +

  # Set color and fill manually for clarity
  scale_color_manual(
    values = c('grey20', 'grey20'),
    labels = c( # Hacky way to get the labels to the unscaled values
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$nonanthro_ed, 0.20),
        ↪ 1)),
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$nonanthro_ed, 0.80),
        ↪ 1))
    )
  ) +
  scale_fill_manual(
    values = c('darkblue', 'goldenrod3'),
    labels = c(
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$nonanthro_ed, 0.20),
        ↪ 1)),
      paste0(round(quantile((data %>% filter(buffer_dist == config_buffer))$nonanthro_ed, 0.80),
        ↪ 1))
    )
  ) +

  # Rug for cfi values (unscaled) used in data
  geom_rug(
    data = data %>% filter(buffer_dist == config_buffer),
    aes(x = cfi_site),
    color = "grey20",
    inherit.aes = FALSE
  ) +

  # Axis labels and legend tweaks
  labs(
    x = "Cumulative site disturbance",
    y = "Expected count",

```



```

    color = "edge \ndensity \n(m/Ha)",
    fill = "edge \ndensity \n(m/Ha)"
  ) +

  # Match x-axis range to data
  scale_x_continuous(limits = c(0, 0.229), expand = c(0, 0)) +
  scale_y_continuous(limits = c(0, 2.0)) +

  # Clean look
  theme_bw() +
  theme(panel.grid = element_blank())

  # Optional mascot
  # add_phylopic(
  #   uuid = get_uuid(name = "Sciurus vulgaris"),
  #   x = 0.04,
  #   y = 1.7,
  #   height = 0.2
  # )

# Save the figure
ggsave(f_cfi_edge, file = "./figures/top_model_cfi_edge_interaction.png", dpi = 400, width = 7,
  ↪ height = 4)

```

```

## Warning: Removed 32 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

8.8. Combined plot:

```

f_full_right <- cowplot::plot_grid(f_edge, f_cfi, ncol = 1, align = "v", axis = "lr")

```

```

## Warning: Removed 37 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

```

## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

```

f_full_top <- plot_grid(f_edge_cfi, f_full_right, ncol = 2, rel_widths = c(3, 1))

```

```

## Warning: Removed 74 rows containing missing values or values outside the scale range
## (`geom_line()`).

```

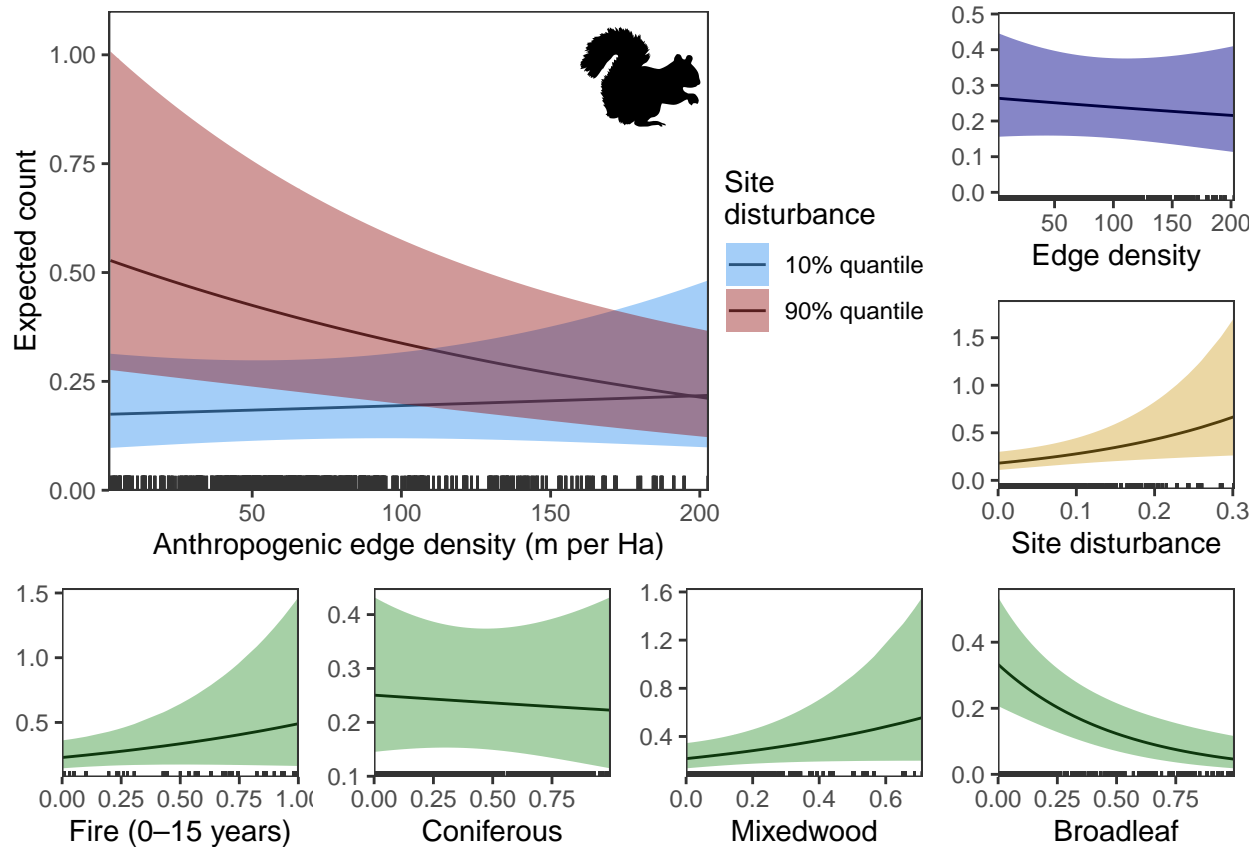
```

f_full_bottom <- plot_grid(f_fire, f_conifer, f_mixedwood, f_decid, nrow = 1, align = "v", axis =
  ↪ "lr")

f_full <- plot_grid(f_full_top, f_full_bottom, ncol = 1, rel_heights = c(2, 1), align = "v", axis =
  ↪ "l")

plot(f_full)

```



```
ggsave(f_full, file = "./figures/top_model_all_effects.png", width = 9, height = 6, dpi = 500)
```

9. Make a pretty AIC table for the report

```
models_rank_table <- models_rank %>%

# Select the model name and all columns up to (but not including) the 'df' column
select(model, 1:(which(names(.) == "df") - 1)) %>%

# Reshape from wide to long format: each row is now a model-predictor pair
pivot_longer(cols = -model, names_to = "predictor", values_to = "value") %>%

# Remove predictors that were not included in a model (i.e., NA values)
drop_na(value) %>%

# Join with a table of formatted covariate names for nicer labels in the output
left_join(covs_formatted, by = c("predictor" = "Covariate")) %>%

# Manually overwrite PrettyName for specific interaction terms
mutate(PrettyName = case_when(
  predictor == "cfi_site:nonanthro_ed" ~ "Edge Density × CFI",
  TRUE ~ PrettyName)) %>%

# Group by model to prepare for collapsing predictors into formula strings
```

```

group_by(model) %>%

# Combine all covariates into a single formula string per model
summarize(formula = paste(PrettyName, collapse = " + ")) %>%

# Clean up the formula strings for display
mutate(
  formula = str_replace(formula, "NA \\+", ""),      # Remove 'NA +' if present at the beginning
  formula = str_replace(formula, "NA", "1"),        # Replace 'NA' (empty models) with
  ↪ intercept-only model '1'
  formula = str_replace(formula, ":", " * "),        # Replace ':' with '*' to show interactions
  ↪ more clearly
  formula = str_replace(
    formula,
    "Fire <15 \\+ Broadleaf \\+ Coniferous \\+ Mixedwood +",
    "CORE "                                           # Group common landcover types under "CORE"
  )
) %>%

# Reattach AIC info by joining back with the full model ranking table
left_join(models_rank %>% select(model, df:ncol()),
  by = "model") %>%

# Order models by ΔAICc (delta)
arrange(delta) %>%

# Select and rename final columns for table output
select(
  "Hypothesis" = "model",
  "Covariates" = "formula",
  df,
  AICc,
  "ΔAICc" = "delta",
  "AICw" = "weight"
)

# Export the formatted model selection table to Excel
writexl::write_xlsx(models_rank_table, "./tables/OSM_model_selection_summary.xlsx")

```

10. Make a pretty model summary table

10.1. for the top model

```

# Extract fixed effects from the top model, including confidence intervals
fixed_effects <- broom.mixed::tidy(top_model, effects = "fixed", conf.int = TRUE) %>%

# Round estimates, standard errors, test statistics, p-values, and confidence intervals to 3
↪ decimal places
mutate(across(c(estimate, std.error, statistic, p.value, conf.low, conf.high), ~ round(.x, 3)))
↪ %>%

# Remove the 'component' column, as it is not needed
select(-component) %>%

```

```

# Rename columns for more understandable names
rename_with(~ c("Type", "Term", "Estimate", "SE", "z", "p", "CI Low", "CI High")) %>%

# Join with formatted covariate names for prettier labels
left_join(covs_formatted, by = c("Term" = "Covariate")) %>%

# Replace term names with pretty labels
mutate(Covariate = case_when(
  Term == "nonanthro_ed:cfi_site" ~ "Edge Density × CFI", # Human-readable label for
  ↪ interaction
  is.na(PrettyName) ~ "Intercept", # Replace NA PrettyName with
  ↪ "Intercept"
  TRUE ~ PrettyName)) %>%

# Drop the 'Term' and 'PrettyName' columns after formatting
select(-Term, -PrettyName)

# Extract random effects from the top model, including confidence intervals
random_effects <- broom.mixed::tidy(top_model, effects = "ran_pars", conf.int = TRUE) %>%

# Modify the group names for better readability
mutate(group = case_when(
  group == "site:array" ~ "Site [nested]", # Rename 'site:array' to 'Site [nested]'
  group == "array" ~ "Array", # Rename 'array' to 'Array'
  TRUE ~ group), # Keep other group names unchanged

# Add a new column 'effect' to indicate random effects
effect = "Random") %>%

# Select relevant columns and rename them for clarity
select("Type" = "effect", "Covariate" = group, "SD (intercept)" = "estimate")

# Combine both fixed and random effects into one summary table
# Add a 'Table' column to distinguish between fixed and random effects
top_model_summary <- bind_rows(
  tibble(Table = "Fixed Effects", fixed_effects),
  tibble(Table = "Random Effects", random_effects)
)

# Export the summary table to an Excel file
writexl::write_xlsx(top_model_summary, "../tables/OSM_top_model_summary.xlsx")

```

10.2. For the second place model

```

# Exact same code as before so not gonna re-type the annotations... you get the deal.
fixed_effects <- broom.mixed::tidy(m_seis_wells, effects = "fixed", conf.int = TRUE) %>%

mutate(across(c(estimate, std.error, statistic, p.value, conf.low, conf.high), ~ round(.x, 3)))
  ↪ %>%

select(-component) %>%

rename_with(~ c("Type", "Term", "Estimate", "SE", "z", "p", "CI Low", "CI High")) %>%

left_join(covs_formatted, by = c("Term" = "Covariate")) %>%

```

```

mutate(Covariate = case_when(
  Term == "nonanthro_ed:cfi_site" ~ "Edge Density × CFI",
  is.na(PrettyName) ~ "Intercept",
  TRUE ~ PrettyName)) %>%

select(-Term, -PrettyName)

fixed_effects

## # A tibble: 8 x 8
##   Type Estimate SE      z      p `CI Low` `CI High` Covariate
##   <chr>      <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl> <chr>
## 1 fixed    -1.57  0.247 -6.37  0      -2.06    -1.09 Intercept
## 2 fixed    -0.006  0.11  -0.052 0.958  -0.222    0.211 Coniferous
## 3 fixed     0.155  0.092  1.68  0.093  -0.026    0.335 Mixedwood
## 4 fixed    -0.478  0.129 -3.71  0      -0.73    -0.225 Broadleaf
## 5 fixed     0.112  0.117  0.957 0.339  -0.118    0.343 Fire <15
## 6 fixed     0.305  0.131  2.34  0.02    0.049    0.561 Pipelines & Transmission~
## 7 fixed    -0.447  0.129 -3.46  0.001  -0.701   -0.194 Seismic Lines
## 8 fixed     0.006  0.143  0.041 0.967  -0.274    0.286 Inactive Well Sites

random_effects <- broom.mixed::tidy(top_model, effects = "ran_pars", conf.int = TRUE) %>%

  mutate(group = case_when(
    group == "site:array" ~ "Site [nested]",
    group == "array" ~ "Array",
    TRUE ~ group),

    effect = "Random") %>%

  select("Type" = "effect", "Covariate" = group, "SD (intercept)" = "estimate")

# Add a title for each table and stack them
top_model_summary <- bind_rows(
  tibble(Table = "Fixed Effects", fixed_effects),
  tibble(Table = "Random Effects", random_effects)
)

writexl::write_xlsx(top_model_summary, "./tables/OSM_second_place_model_summary.xlsx")

```