

NECN Weibull establishment parameters

Sam Flake

2024-08-30

LAI establishment curves

NECN v7 uses a modified Weibull curve to describe species shade preferences, replacing the clunky LAI table that was found in the NECN text input file in previous versions. The LAI table was based on expert judgment, not data, and applied the same discontinuous LAI breaks to all species. The new Weibull parameters are designed to be 1) more accurate and customizable to each species, and 2) parameterized from data.

This script has an example for how to parameterize the Weibull establishment curves for two species. I recommend, instead of running this Rmd, look at the file `weibull_params_example.R`. For a more detailed script (more similar to what you would need for a real NECN model), see `get_weibull_params_from_fia_all_data.R`. That script has more options to include additional species, deal with rare species, and choose how the curves are fit. It should be pretty much plug-and-play with new species or study areas with a little modification (the species choices, matching them up to FIA SPCDs, and your LAI parameters from the NECN functional table file).

Establishment in NECN and a little Bayesian tangent

During the reproduction phase of NECN, sites are evaluated for whether or not a new cohort can be recruited there using several criteria. The first hurdle which must be passed is whether there is enough light. For each species, the LAI of the site determines the probability that a recruit can establish there (this is what the LAI establishment table and the Weibull parameters determine). So, what NECN needs, is a way to represent the probability of recruitment given the LAI of the site, or $p(\text{recruit}|\text{LAI})$.

The way I've gone about doing this is by using inventory data from the USFS Forest Inventory and Analysis (FIA) project. For each species, I assume that the probability of recruitment at a given LAI value is related to the probability of the FIA project finding a seedling in a plot of that LAI – it's a little different, but it should be strongly and directly related. The easiest way to calculate something related to establishment and LAI is to find where all the seedlings of a species are located, and look at the LAI of those sites. However, instead of giving us $p(\text{recruit}|\text{LAI})$, this gives us $p(\text{LAI}|\text{recruit})$ – what is the probability of a value of LAI, given that there are seedlings there?

While $p(\text{LAI}|\text{recruit})$ isn't quite right, we can relate it to $p(\text{recruit}|\text{LAI})$ by using Bayes' theorem:

$$p(\text{recruit}|\text{LAI}) = p(\text{LAI}|\text{recruit}) * p(\text{recruit}) / p(\text{LAI})$$

While I don't use a Bayesian analysis here formally, I find the logic compelling – the probability of recruitment given LAI is related to the histogram of LAIs where a seedling is found, by: 1) the landscape distribution of LAI (what is the probability of a given LAI?), and 2) the fecundity of the species (what is the probability of recruitment?).

Hopefully this motivating logic is apparent in the script below, when we get to the steps that scale the probability of recruitment to set the max $p(\text{recruit})$ equal for each species (which allows the area under the curve, or fecundity, to differ among species) and when we limit the landscape from which we sample (accounting for the distribution of LAI).

Weibull overview

The Weibull distribution is a probability distribution function described by two parameters, a shape parameters and a scale parameter. It is great for distributions that are always positive and have long right tails – like the typical distribution of leaf area index on a landscape. By changing the shape and scale parameters, we can get a wide variety of distributions, from a really shade-intolerant species with shape parameter < 1, to a shade-tolerant with a shape parameter around 2.

It has the following equation:

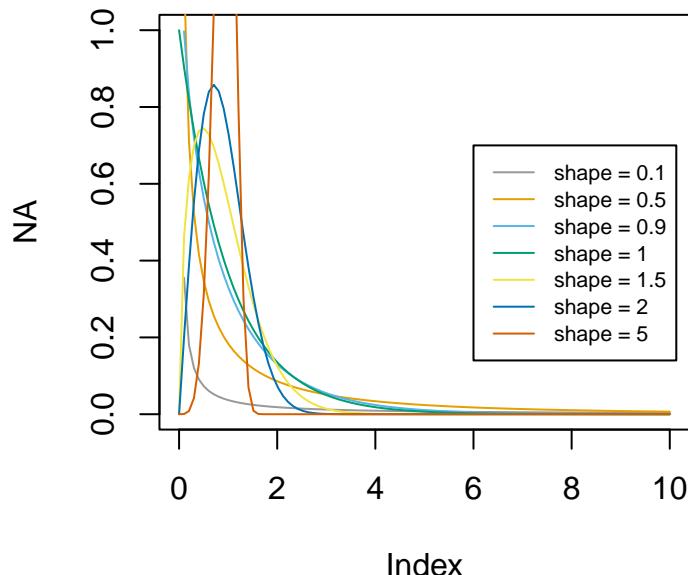
$$y = (a/b) * ((X/b)^{(a-1)}) * \exp(- (X/b)^a)$$

where a is the shape parameter and b is the scale parameter. In our case, X is plot-level LAI.

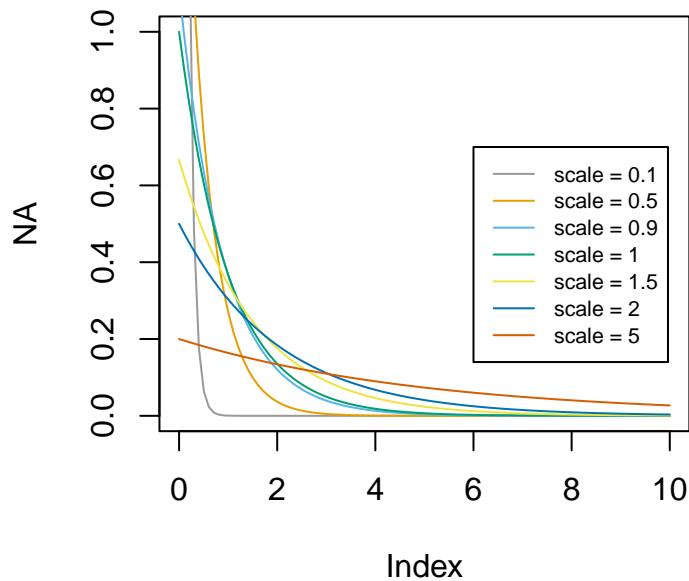
I've also modified the equation to allow for what I'm calling location and adjustment parameters, c and d. If you use these, then the curve is not describing a PDF anymore, just a Weibull-ish curve that might fit your data better. The area under the curve if you're using c or d will probably not be 1, so some species might have greater reproductive success than others – this may be good or bad, depending on your needs.

$$y = (((a/b) * ((lai/b)^{(a-1)}) * \exp(- (lai/b)^a)) + c) * d$$

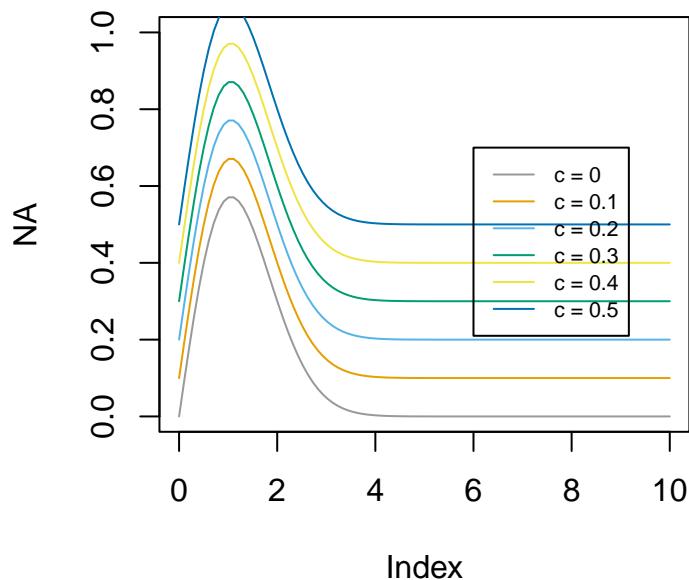
Here's a set of 2-parameter Weibull distributions with varying shape parameters and scale = 1.



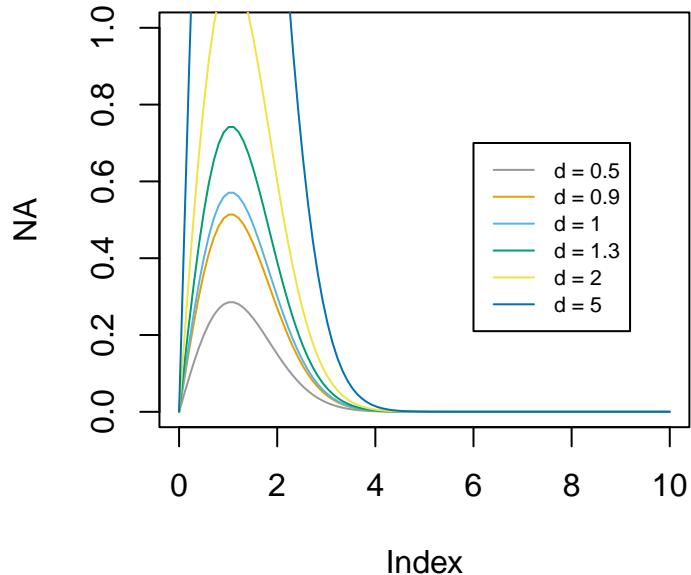
And here's the same with shape = 1 and scale varying.



Here's a Weibull curve with shape = 2 and scale = 1.5, with c varying:



And here's what the d parameter does, again with shape = 2 and scale = 1.5:



So there's a variety of shapes we can fit using Weibull, which solves one issue with the LAI tables, but we still need to be able to parameterize them from real data.

Parameterizing light preferences for species

This script has several steps:

1. import and wrangle FIA data for plots, trees, seedlings, and sitetrees
2. assemble FIA plots into age/species cohorts like LANDIS-II
3. estimate leaf area for plots using the same algorithm as NECN
4. get amount of regeneration per plot per species
5. fit distributions to regen~LAI values using nonlinear least squares

Step 0: Loading libraries and functions

We also have a big function that fits the weibull parameters, `fit_weibull()`, but I won't put it here – take a look at `get_weibull_params_from_fia_all_data.R` for details.

We also have a few options that will make sense once we get through the script. Mostly, we need to define what makes a plot “available” for recruitment, so we're not including a bunch of plots out of the range of a species where we wouldn't expect to find them. Don't worry about these for now, but they're in the script for when you need them later.

```
#should we restrict our analysis to plots that are near seedlings, near adult trees, or unrestricted?
#this is by far the slowest part of the script
range_method <- "seedling" # "seedling" or "adult"; other options skip this step and use all the data fr
#range buffer size -- how many meters can plots be from seedlings to count in the calculation?
```

```

range_buffer <- 10000
#should maximum suitability be set to 1 and everything scaled to match? This will (almost?) 
# always increase total suitable light levels and thus regeneration
scale_max_p <- TRUE
#should the area under the curve be set to 1 for all species? This will cancel out
# differences in abundance/fecundity among species. Probably not recommended if you have
# a lot of range-restricted species, unless you're using the range_method option above.
# But this is very much recommended if you're wanting to fit a true Weibull PDF where AUC = 1
# (i.e. using use_c_d == "neither")
set_auc_to_1 <- FALSE
#what formula should we use? See above equation fit_weibull. For a regular Weibull PDF,
# set use_c_d = "neither", to only use the shape and scale parameters. Otherwise,
# you can use the c parameter to translate the whole curve up or the d parameter to stretch the
# whole curve. This gives more flexibility in the kinds of shapes that can be fit,
# and allows for the area under the curve to be greater than 1.
use_c_d <- "d"

```

Step 1: Wrangling data

I've got some FIA data pre-wrangled for us. It's FIA plots randomly selected from Michigan and Minnesota, and we're looking at establishment of balsam fir (*Abies balsamea*) and quaking aspen (*Populus tremuloides*). We need tree data to estimate LAI and seedling data to look at recruitment. We also need the sitetree table to make some estimates of tree ages.

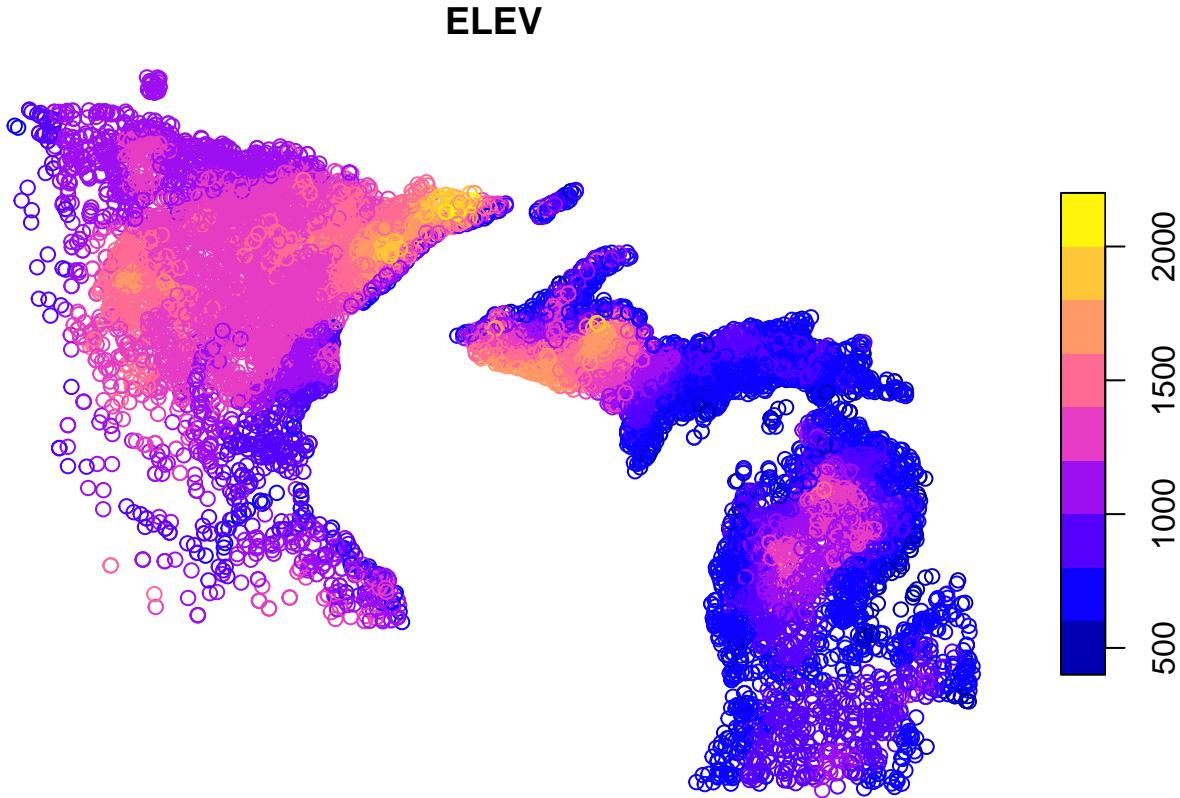
```

#species reference data
sp_ref <- read.csv("./example_data/REF_SPECIES.csv")

trees <- read.csv("./example_data/trees.csv")
plot <- read.csv("./example_data/plot.csv")
seedlings <- read.csv("./example_data/seedlings.csv")
sitetrees <- read.csv("./example_data/sitetrees.csv")

```

Here's what our plots look like:



Step 2: Crosswalk LANDIS-II and FIA

this is the functional table and species table from your NECN project. We need them for a few things, like calculating LAI and crosswalking LANDIS-II species names and FIA SPCDs

```
func_table <- read.csv("./example_data/NECN_Functional_Table_inv_moisture.csv")
sp_table <- read.csv("./example_data/NECN_Spp_Table_inv_necn7.csv") %>%
  left_join(func_table, by = "FunctionalGroupIndex")
```

We need to assign each FIA species group a value for KLAI and MaximumLAI. You may have a lot of species in FIA that aren't in your LANDIS project; I'd suggest just coming up with a catchall functional group for those and putting all the extra SPGRPCDs in there. Multiple SPGRPCDs might match each NECN functional group, and some FGs might not have a corresponding SPGRPCD if there are no FIA species that match. That's fine.

```
spgrp_lai <- data.frame(SPGRPCD = unique(trees$SPGRPCD),
                         FunctionalGroupIndex = numeric(length(unique(trees$SPGRPCD))) %>%
  arrange(SPGRPCD) %>%
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(5,4, 3, 2), 1, FunctionalGroupIndex)) %>% #jack an
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(6), 2, FunctionalGroupIndex)) %>% #balsam fir an
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(9), 3, FunctionalGroupIndex)) %>% #eastern whit
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(36, 35), 4, FunctionalGroupIndex)) %>% #black spr
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(7), 5, FunctionalGroupIndex)) %>% #tamarack, cede
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(30, 31, 33, 42), 6, FunctionalGroupIndex)) %>% #n
```

```

  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(37, 38, 41), 7, FunctionalGroupIndex)) %>% #aspen
  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(25, 26, 28, 29, 27, 32, 43, 40, 39, 55), 8, FunctionalGroupIndex))
#  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(), 9, FunctionalGroupIndex)) %>% #wet broadleaf
#  mutate(FunctionalGroupIndex = ifelse(SPGRPCD %in% c(), 10, FunctionalGroupIndex)) %>% #shrubs
  left_join(func_table %>% dplyr::select(FunctionalGroupIndex, KLAI, MaximumLAI))

## Joining with `by = join_by(FunctionalGroupIndex)`

#here, we're just doing ABBA and POTR, not all the species
spp_to_use_all <- c("ABBA", "POTR5") #sp_table$SpeciesCode

```

We need to get SPCD for each species. This will be different depending on how the species are named, but we want a crosswalk from the names used in LANDIS to FIA SPCD somehow. In this project, I used the USDA PLANTS symbol, which is found in the FIA species reference table to crosswalk to FIA species code. We want a column of SpeciesCode that is your LANDIS species identifier, and a column of SPCD that is the FIA species number.

```

sp_ref$SpeciesCode <- sp_ref$SPECIES_SYMBOL
spp_crosswalk <- sp_ref[sp_ref$SPECIES_SYMBOL %in% spp_to_use_all, ] %>%
  dplyr::arrange(SPECIES_SYMBOL) %>%
  dplyr::select(SpeciesCode, SPCD)

# for seedlings, some species don't have enough to get good estimates. This table
# combines several SPCD for each species (e.g., uncommon Populus spp.) to
# improve parameter estimates for rare species. You have to do this by hand for each
# species you're modeling. Here it's simple because I'm just using two common species
spp_crosswalk_combine <- tibble(SpeciesCode = spp_to_use_all,
  SPCD = list(12, "#Balsam fir"
              746 "#quaking aspen"
            ))
spp_to_use <- spp_crosswalk$SpeciesCode
spcd_to_use <- spp_crosswalk$SPCD
all_spcd <- unique(trees$SPCD)

```

Step 3: Calculate per-tree LAI contribution and plot-level LAI

I'm using the same logic that NECN does to estimate plot-level LAI, so that we don't have mismatches when we apply the parameters we're generating to an NECN model. So we need to get ages and biomass for cohorts, just like LANDIS-II.

```

#we need to make sure the trees have ages so we can bin them
enough_trees <- table(sitetrees$SPCD) %>% `|`[which(table(sitetrees$SPCD) > 100)] %>% names()

#make an age model for each species -- you could do this however you want; this is a quick and dirty
# cubic regression for each species, and lumping all rare species into one model
age_model <- lm(log(AGEDIA) ~ poly(log(DIA), 3)*as.factor(SPCD),
  data = sitetrees[!is.na(sitetrees$DIA) & !is.na(sitetrees$AGEDIA) & sitetrees$SPCD %in%]
age_model2 <- lm(log(AGEDIA) ~ poly(log(DIA), 3),

```

```

    data = sitetrees[!is.na(sitetrees$DIA) & !is.na(sitetrees$AGEDIA), ])

#assign ages to trees based on species and diameter
trees <- trees %>%
  mutate(PLOT.YEAR = paste(PLT_CN, INVYR, sep="."))
right_join(., plot, by = c("PLT_CN" = "CN")) %>%
  dplyr::mutate(DIA_cm = DIA * 2.54,
                 HT_m = HT / 3.3808) %>%
  dplyr::filter(STATUSCD == 1) %>%
  dplyr::left_join(spgrp_lai)

## Joining with 'by = join_by(SPPRCDF)'

for(i in 1:length(all_spcd)){
  spcd = all_spcd[i]
  sp = spp_crosswalk[match(spcd, spp_crosswalk$SPCD), "SpeciesCode"]

  if(spcd %in% enough_trees){

    trees[trees$SPCD == spcd, "TOTAGE2"] <- exp(predict(age_model,
                                                       newdata = trees[trees$SPCD == spcd, ]))

  } else{
    trees[trees$SPCD == spcd, "TOTAGE2"] <- exp(predict(age_model2,
                                                       newdata = trees[trees$SPCD == spcd, ]))

  }
}

trees$age <- ifelse(is.na(trees$TOTAGE), trees$TOTAGE2, trees$TOTAGE)
trees$age <- ifelse(trees$age > 500, 500, trees$age)
breaks <- seq(0, max(trees$age, na.rm = TRUE) + (10 - max(trees$age, na.rm = TRUE)) %% 10, by = 5)
trees$age_bin <- base::cut(trees$age, breaks = breaks, labels = breaks[-1], right = TRUE)

#use the NECN LAI calculation to get plot-level LAI
trees_bin <- trees %>%
  group_by(PLOT.YEAR, SPCD, age_bin) %>%
  summarise(cohort_biomass = sum(CARBON_AG/0.47, na.rm = TRUE) * 0.11, #sum biomass and convert to g m-
            KLA1 = KLA1[1],
            MaximumLAI = MaximumLAI[1]) %>%
  mutate(LAI_tree = MaximumLAI * cohort_biomass/(KLA1 + cohort_biomass))

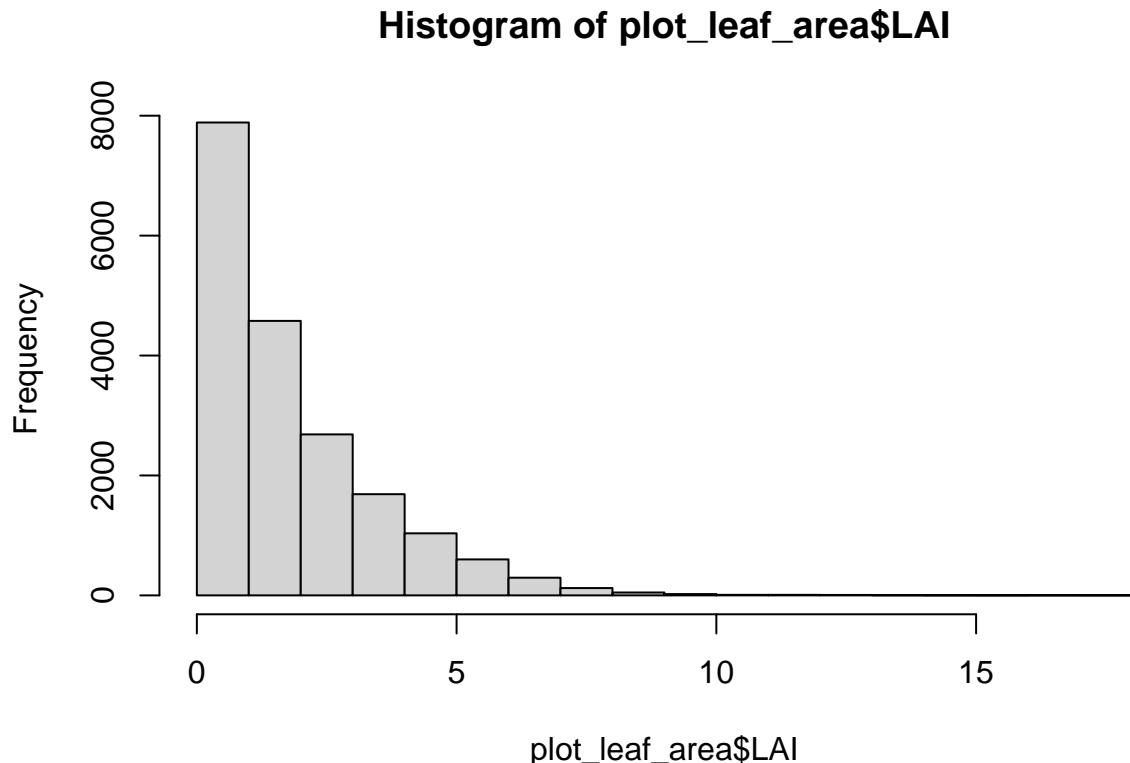
## 'summarise()' has grouped output by 'PLOT.YEAR', 'SPCD'. You can override using
## the '.groups' argument.

#calculate leaf area index per plot
plot_leaf_area <- trees_bin %>%
  group_by(PLOT.YEAR) %>%
  #m2 per tree * trees per acre * acre per m2 = meters squared leaf area per meter squared ground
  summarise(LAI = sum(LAI_tree)) %>%
  filter(!is.na(LAI) & !is.infinite(LAI)) %>

```

```
filter(LAI < 20)

hist(plot_leaf_area$LAI)
```



Step 4: See where seedlings are present and which plots they could be present

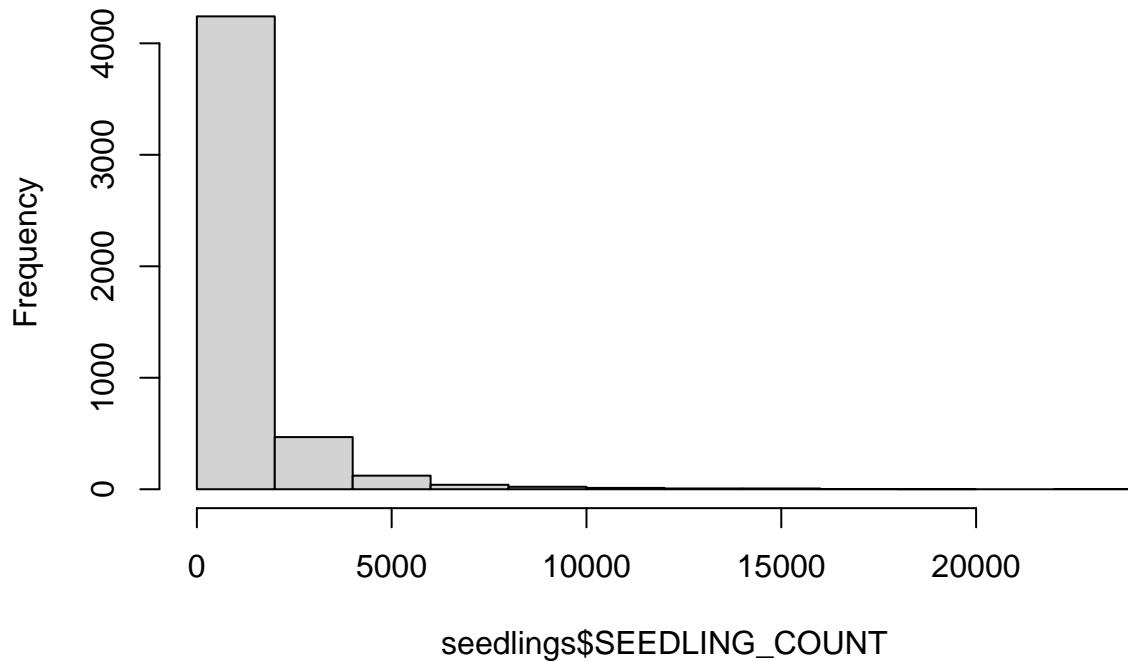
We could be done at this point, if we wanted – just join the seedlings to the LAI table and make a histogram and call it good. I think it's working better to go a little further, and spatially restrict the plots we're considering as potential for recruitment.

First let's get how many seedlings we have per plot

```
# get seedlings
seedlings <- seedlings %>%
  mutate(PLOT.YEAR = paste(PLT_CN, INVYR, sep="."),
         TPA_UNADJ = ifelse(is.na(TPA_UNADJ), 0, TPA_UNADJ)) %>%
  filter(PLOT.YEAR %in% plot_leaf_area$PLOT.YEAR) %>% #filter out plots with bad LAI data, inappropriate
# filter(TOTAGE <= 5) %>% #only useful in RMRS zone, and only collected for a subset of trees
  group_by(PLOT.YEAR) %>%
  mutate(SEEDLING_COUNT = sum(TPA_UNADJ)) %>%
  slice_head(n = 1)

hist(seedlings$SEEDLING_COUNT)
```

Histogram of seedlings\$SEEDLING_COUNT



Next we need to figure out which plots are available for recruitment for each species. Here, I'm restricting the available plots to those within 10 km of a seedling, as a way to represent the geographical range of the species' natural recruitment. There's other options in the other script. You could get very fancy with this part if you wanted to.

First let's do this with balsam fir to demonstrate:

```
SPCD <- spp_crosswalk_combine[[1, "SPCD"]][[1]]  
  
seedling_sub <- seedlings[seedlings$SPCD %in% SPCD,]  
  
plot_sf <- plot %>%  
  sf::st_as_sf(coords = c("LON", "LAT")) %>%  
  st_as_sf %>%  
  sf::st_set_crs("EPSG:4326") %>%  
  sf::st_transform("EPSG:5070") %>%  
  mutate(PLOT_YEAR = paste(CN, INVYR, sep="."))  
  
seedling_sf <- left_join(seedling_sub, select(plot, CN, LAT, LON), by = c("PLT_CN" = "CN")) %>%  
  sf::st_as_sf(coords = c("LON", "LAT"))%>%  
  sf::st_set_crs("EPSG:4326") %>%  
  sf::st_transform("EPSG:5070")  
  
#create a zone within buffer distance of seedlings  
seedling_buffer <- sf::st_buffer(seedling_sf, range_buffer) %>%  
  st_union() %>%  
  st_as_sf()
```

```
#figure out which plots are in range  
plot_sf$in_range <- lengths(st_within(plot_sf, seedling_buffer))
```

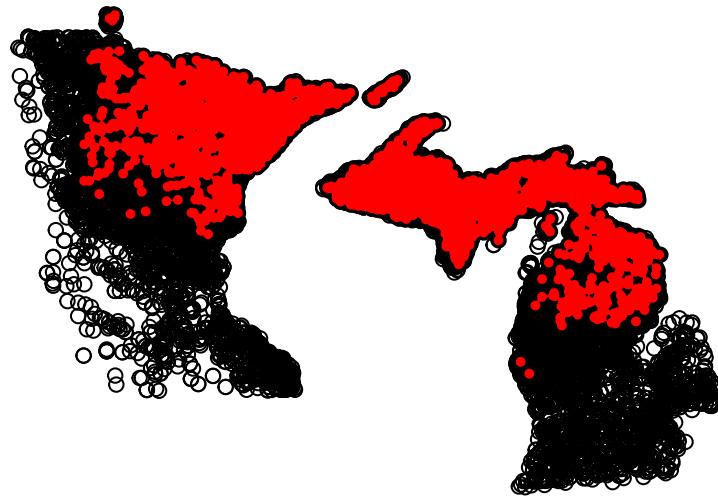
Here's what all of our plots look like:

```
plot(st_geometry(plot_sf))
```



But here's where balsam fir regeneration occurs – just the Upper Peninsula of Michigan and northern Minnesota, mostly. This is the 10 km buffer we created around all the plots with balsam fir seedlings.

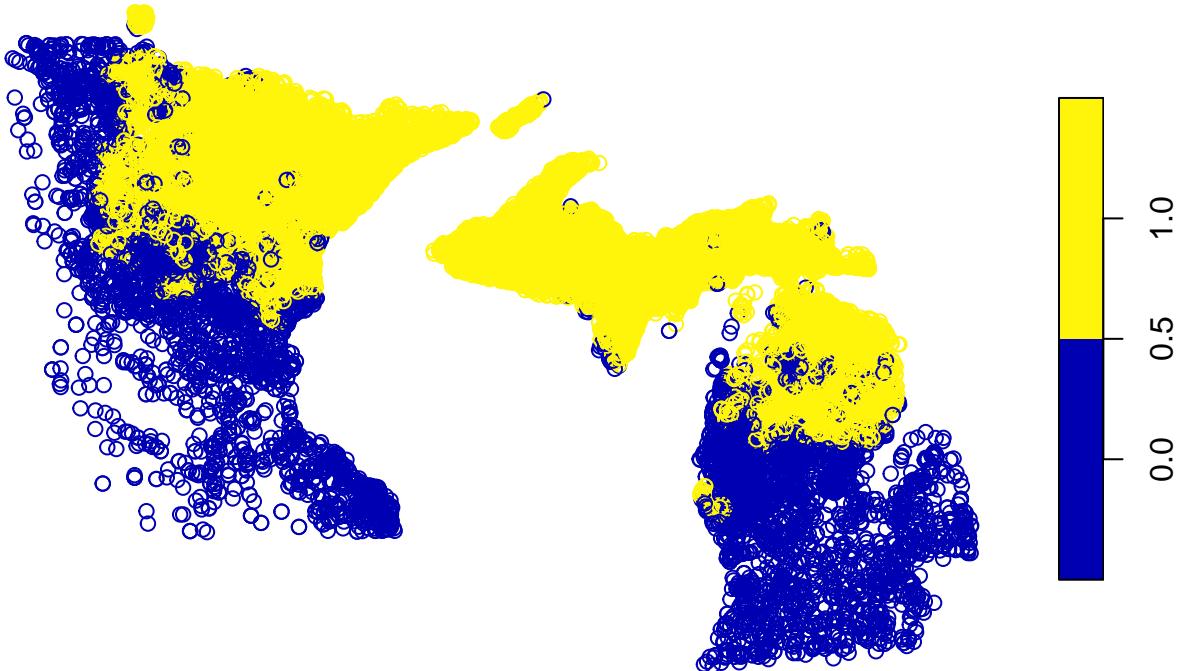
```
plot(st_geometry(plot_sf))  
plot(st_geometry(seedling_buffer), col = "red", add = TRUE)
```



So we want to just use a subset of the plots:

```
plot(plot_sf["in_range"])
```

in_range

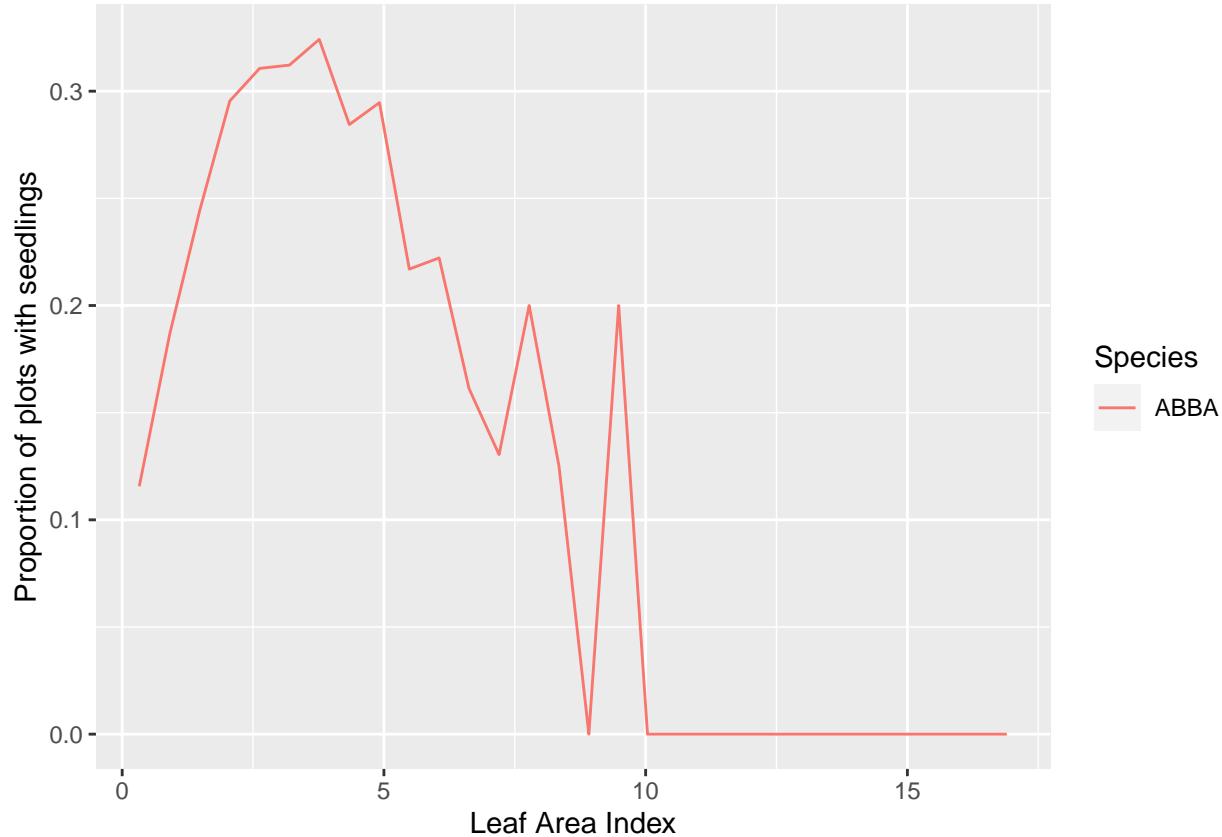


When we sum up the number of seedlings in each plot, we get something like this, where we have a column of plot IDs, a column of LAI for each plot, and then the number of seedlings for each species (in this case, just balsam fir). We've counted seedlings when they're present, assigned a 0 if the plot is in range and has no seedlings, and an NA if it's out of range.

```
## # A tibble: 6 x 3
##   PLOT.YEAR      LAI   ABBA
##   <chr>     <dbl> <dbl>
## 1 100180926010661.2005 0.00941    0
## 2 100181596010661.2005 1.54       0
## 3 100181813010661.2005 2.55       3
## 4 100182567010661.2005 0.786      0
## 5 100183466010661.2005 2.38       NA
## 6 100184322010661.2005 4.01       NA
```

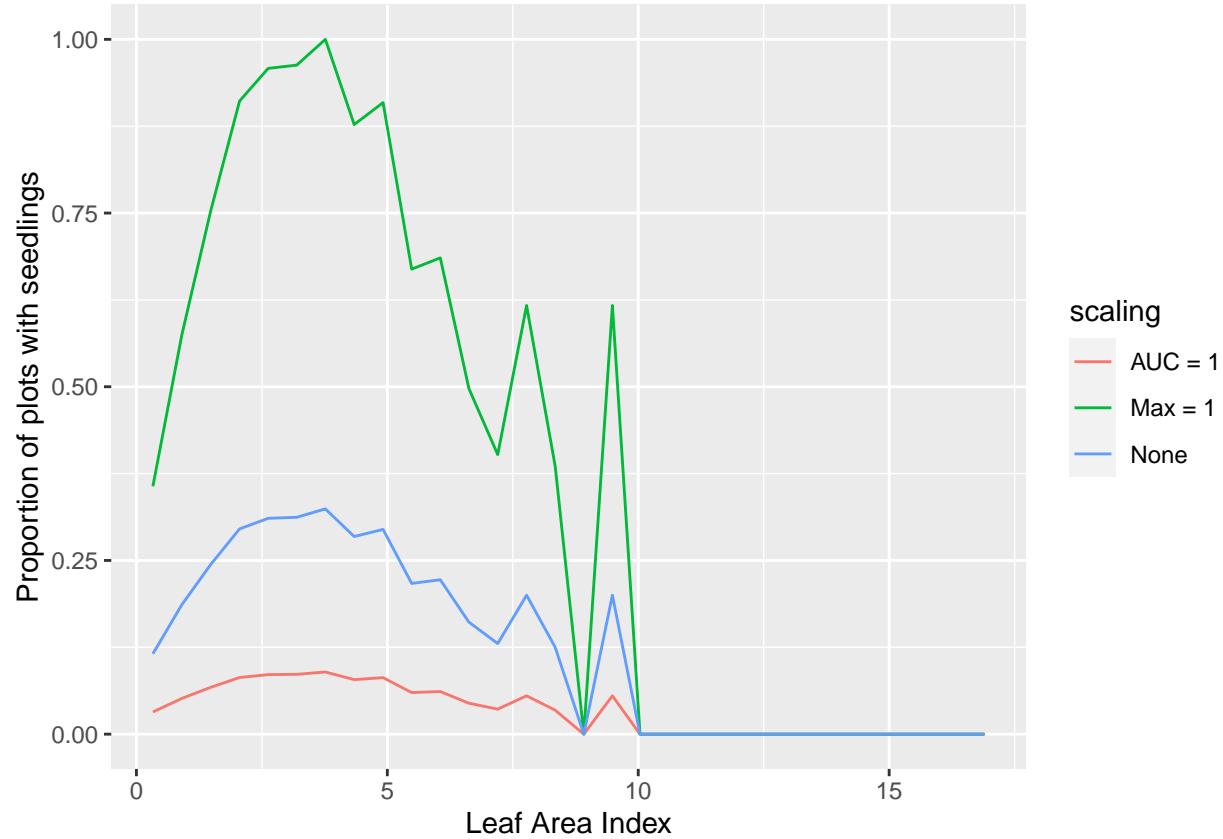
Step 5: Fit a distribution to the seedling data

Now that we know the LAI for a bunch of plots and whether or not there's seedlings present in those plots, we can start to explore how seedlings are distributed with respect to LAI and try to fit distributions to that data. Here, I've binned LAI into 30 equal bins, and, for each bin, found what proportion of the plots have balsam fir seedlings:

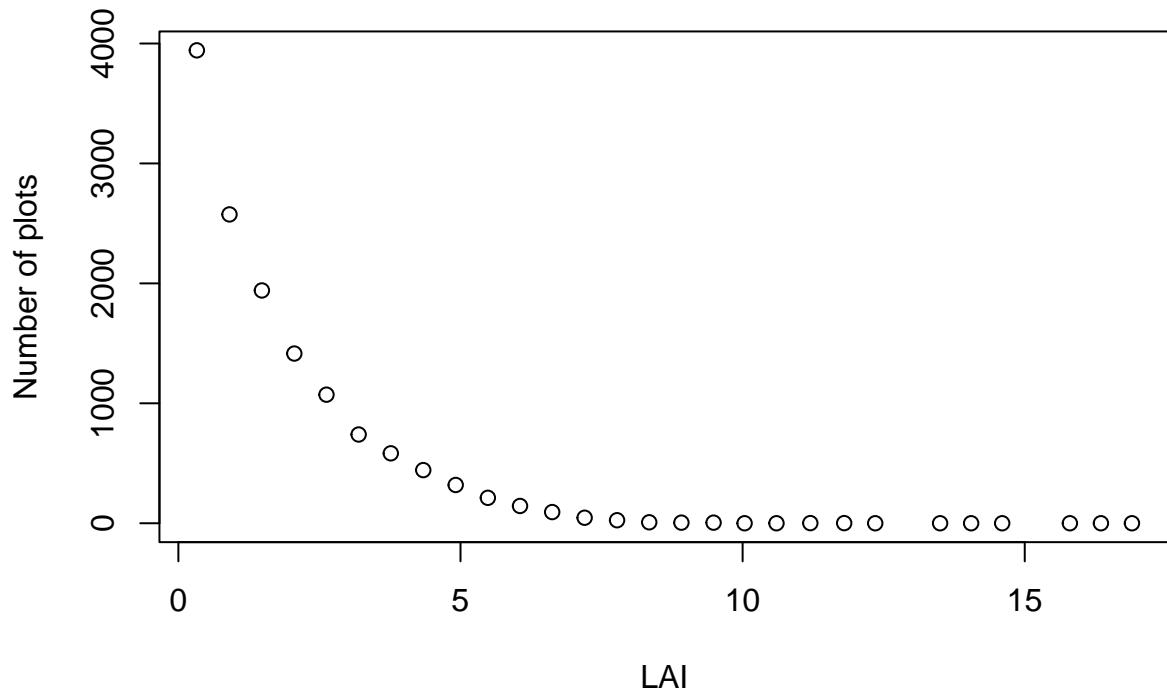


We can see that there's a peak in seedlings at around $\text{LAI} = 4$ or so, which is a little past the point where most sorts of forests' canopies have closed. This pattern of regeneration is what we expect from a shade-tolerant tree like balsam fir. There are still some seedlings in more open sites, perhaps advanced regeneration in a gap or just early colonists of a blowdown or fire.

We might want to fiddle with this data some, depending on how our model is parameterized. We could set the sum of the area under the curve equal to 1, which would make this closer to a PDF, or we could scale the whole thing so that the maximum probability of finding a seedling is equal to 1, which more closely matches the behavior of the LAI-establishment tables from NECN v6.



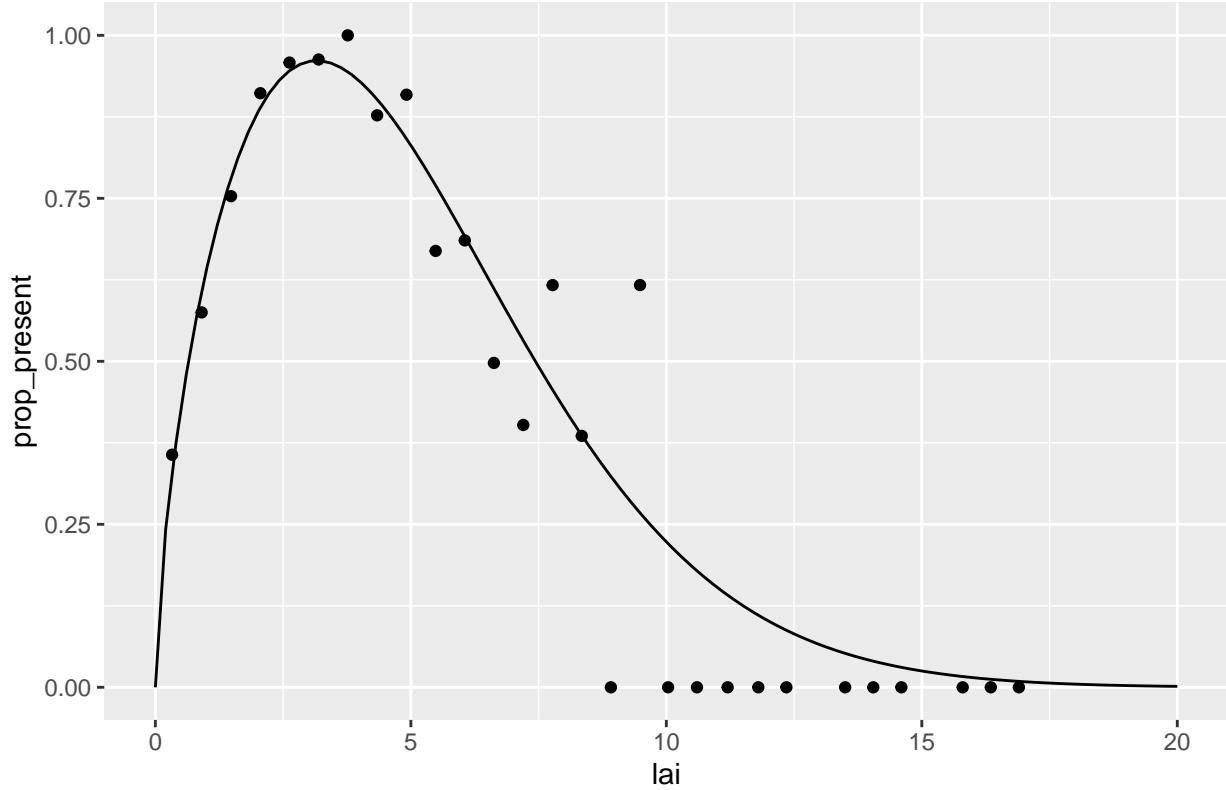
Note that things get a little funky at higher LAIs – that's because there aren't that many plots there. We can account for that when we fit our distributions by including the number of plots in each bin as weights, which will downweight the highly variable right tail of that distribution and put more emphasis on fitting the left of the distribution where most of the data are. Here's how those weights look:



Now that we have the probability of finding a seedling at a given LAI, we can fit a curve to that data and generate the inputs we need for NECN. I've found that using that a, b, and d parameters works pretty well for this species, and I'll be using the version of the seedling data that scales the maximum $p(\text{seedling})$ to equal 1, so I can compare to the NECN v6 values more easily.

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

ABBA



Let's compare this to a previous version of the model, based on parameters inherited from a previous model. Those parameters looked like this:

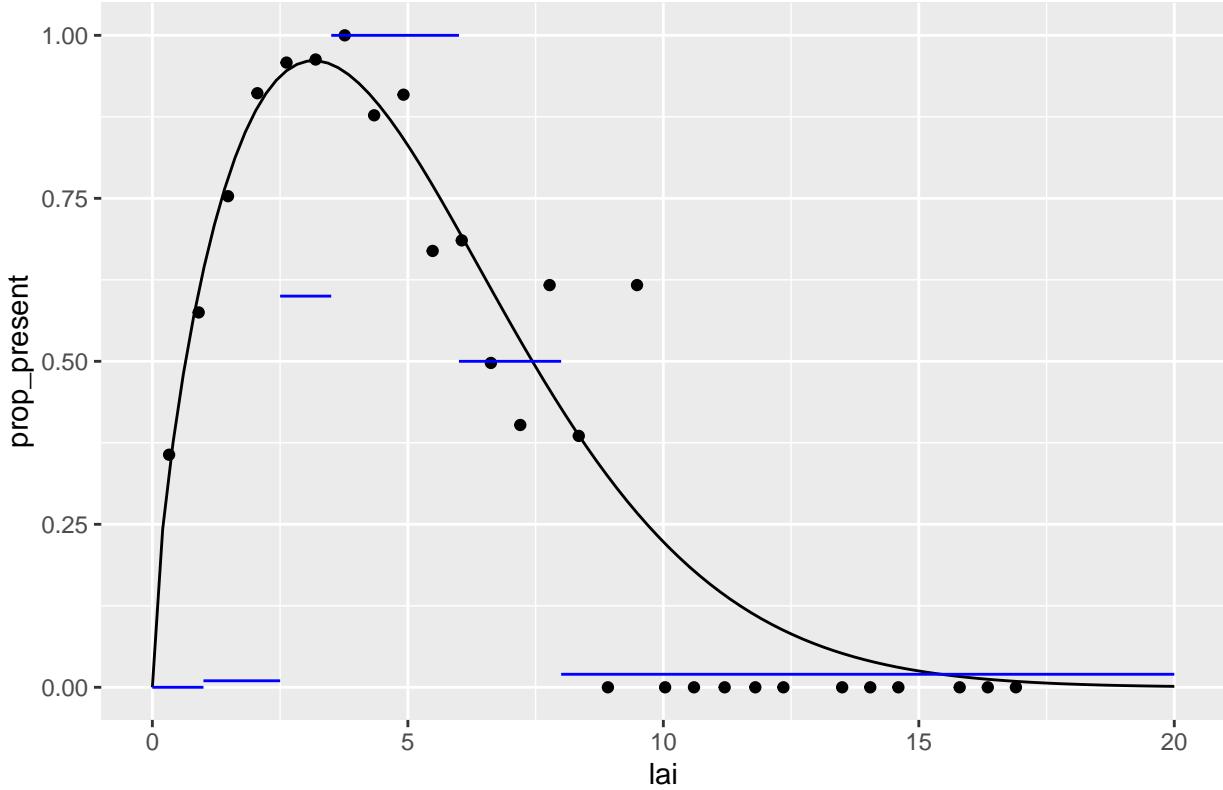
```
>> Shade      Max LAI
>> Class
>> -----
 1      1
 2      2.5
 3      3.5
 4      6
 5      8

LightEstablishmentTable

>> Spp Shade   Probability
>> Class      by Actual Shade
>> -----
>>      0       1       2       3       4       5
 1  1.0     0.5    0.02    0.0     0.0     0.0
 2  0.6     1.0    0.5    0.02    0.0     0.0
 3  0.01    0.6    1.0    0.5    0.02    0.0
 4  0.0     0.01   0.6    1.0    0.5    0.02
 5  0.0     0.0    0.01   0.5    1.0    1.0
```

We had balsam fir as shade tolerance class 4, which means its recruitment would have peaked at LAI between 3.5 and 6. We can look at the FIA data and see that this is a little off. Let's plot the original LAI table values as blue horizontal lines, on top of the curve we just calibrated, and see how it compares.

ABBA

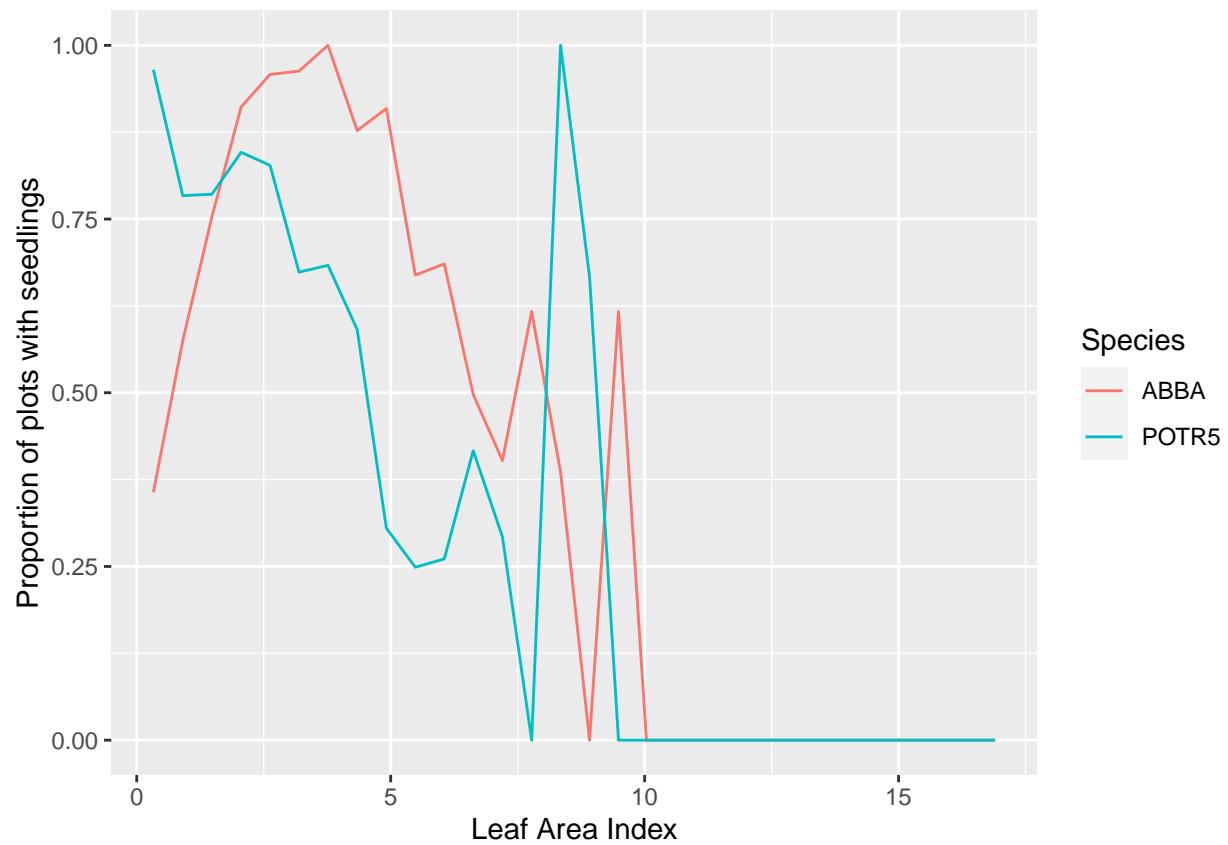


We're actually not that far off in this case – the shape of the curve is similar, and the LAI of maximum recruitment success isn't that dissimilar. But we've now got a repeatable workflow, based on real data, that we can reference. We can quickly generate curves for new species rather than copying from projects from dissimilar areas.

As far as parameters go, we end up with a shape parameter of `{r}mod_output[[1]]$shape`, a scale parameter of `{r}mod_output[[1]]$scale`, and an adjust parameter of `{r}mod_output[[1]]$adjust`. Because I set the script not to use the location parameter, it is set to 0 (so the right tail asymptotically approaches 0). The shape parameter greater than 1 means that the species has a peak in recruitment at $\text{LAI} > 0$; a really shade-intolerant species should have a negative-exponential shape, with shape parameter less than 1. Let's take a look at one and make sure that this all works like we expect.

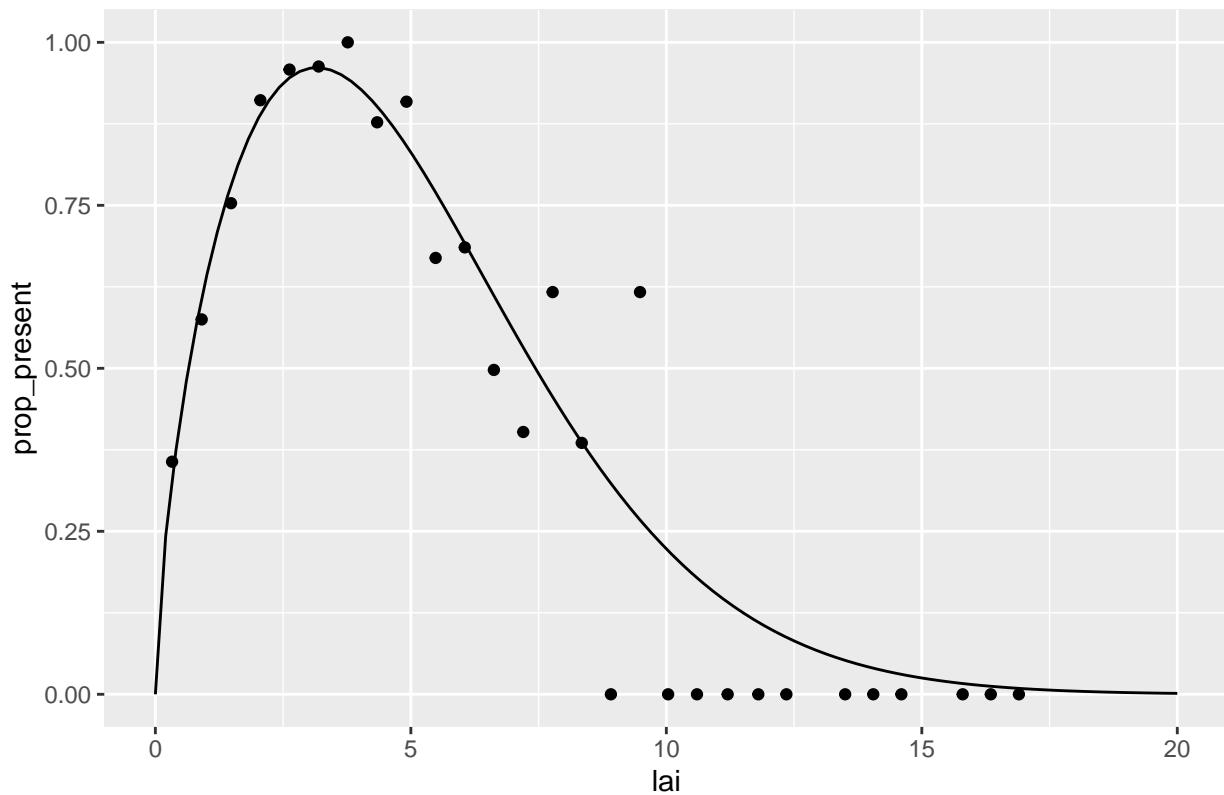
Adding a species

Let's do the same thing for a shade-intolerant species, quaking aspen. I'll just run through the whole script with both species, balsam fir (ABBA) and quaking aspen so we can compare.



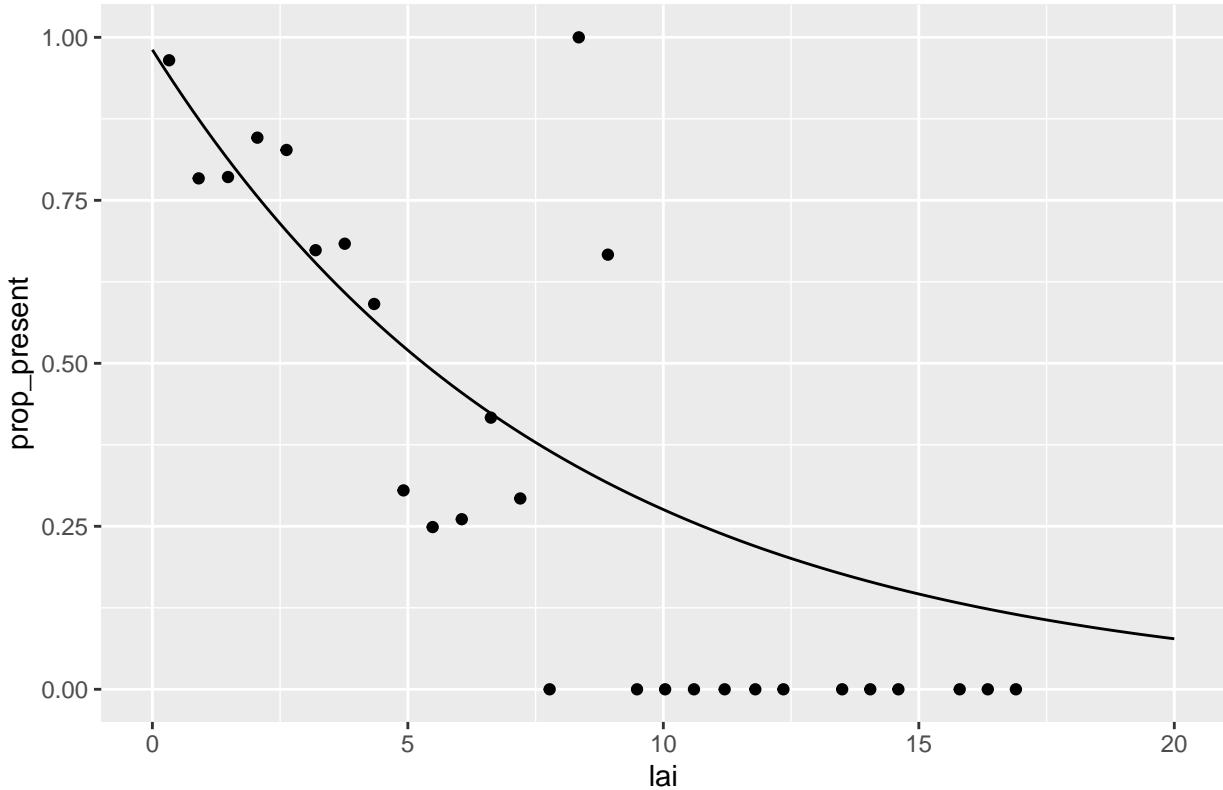
```
## [[1]]
```

ABBA



```
##  
## [[2]]
```

POTR5



Looking at the previous two panels, we can see that the distribution of recruits in FIA plots differ strongly between balsam fir and quaking aspen, and that our Weibull curves were able to represent them pretty well. As far as the parameters go, they look like this:

```
print(dplyr::select(weibull_models, Species, shape, scale, location, adjust))
```

```
## # A tibble: 2 x 5
## # Rowwise: Species
##   Species shape scale location adjust
##   <chr>    <dbl>  <dbl>    <dbl>  <dbl>
## 1 ABBA     1.64   5.60      0    7.00
## 2 POTR5    1       7.88      0    7.73
```

The shape parameter is the most important to determining whether the species has its highest or lowest recruitment at LAI = 0. You may need to manually set that for some species to get the right shape. Here, the curve-fitting algorithm got it right, but if it doesn't, you can modify the species_bounds variable in the script.

This script can be used for any number of species, as long as they're in FIA. It would take a little work to adapt it to other plot networks, but it shouldn't be that difficult – the main steps would remain the same. If no plot data is available, of course, using inherited parameters from other sites or similar species remains one of the easiest ways to get a model up and running so it can be calibrated. Let me know if you need any help with this script or NECN regeneration in general, at swflake@ncsu.edu.