

# Multilevel Urban Tree Allometric equations

Tedward Erker

September 26, 2018

## Contents

<b>1 Objective</b>	<b>2</b>
<b>2 Caveat</b>	<b>3</b>
<b>3 Why needed</b>	<b>3</b>
<b>4 What should the structure of the equations be?</b>	<b>5</b>
4.1 Height . . . . .	5
4.2 DBH . . . . .	6
4.3 sigmoidal forms, see the citations within: . . . . .	6
4.4 It would be nice to have repeat measures of trees to get diameter increments. Individual variation can be large and noisy. . . . .	6
4.5 Could the response be multivariate? For example: height, crown length, leaf area etc are covariate. Modeling them together could make sense	ATTACH
4.6 [?] . . . . .	7
<b>5 Introduction</b>	<b>7</b>
5.1 past work . . . . .	9
5.2 urban trees equations . . . . .	11
5.3 past modelling . . . . .	11
<b>6 Methods</b>	<b>11</b>
6.1 Data . . . . .	11
6.1.1 Tidy data for this analysis . . . . .	12
6.1.2 The Species for which we have age and dbh: . . . . .	22
6.1.3 Cities and Climate . . . . .	26
6.1.4 Get new climate data. Growing degree days and Precip. HDD is super high for minneapolis, but harsh winters aren't relevant. . . . .	27
6.1.5 Making figures for this data section . . . . .	39
6.2 Modelling . . . . .	41
6.2.1 Model . . . . .	42

6.2.2	fitting model . . . . .	44
6.2.3	model form: Weibull . . . . .	44
6.2.4	a look at gamma family . . . . .	53
6.2.5	<b>TODO</b> genus: none; species: single; cities: single; climate: none; hetero: no; family gamma . . . . .	57
6.2.6	<b>TODO</b> genus: many; species: many; cities: many; climate: b1linint; hetero: no; family: Gamma . . . . .	67
6.2.7	<b>TODO</b> genus: many; species: many; cities: many; climate: b3linint; hetero: no; family: Gamma Would need to fix priors! . . . . .	77
6.2.8	Full model . . . . .	87
6.2.9	for diagnostics . . . . .	88
6.2.10	look at distributions . . . . .	88
<b>7</b>	<b>Results</b>	<b>94</b>
<b>8</b>	<b>Discussoins</b>	<b>94</b>
<b>9</b>	<b>Exploring age and dbh</b>	<b>95</b>
9.1	load libraries . . . . .	95
9.2	functions . . . . .	95
9.3	read in data . . . . .	95
9.4	fix some species things . . . . .	95
9.4.1	fix lower case species codes . . . . .	95
9.4.2	fix QUAG1 to be QUAG . . . . .	95
9.4.3	fix common names . . . . .	96
9.5	species are . . . . .	96
9.6	tidy a few names and select variables of interest here . . . . .	100
9.7	plot age versus dbh for all trees . . . . .	101
9.8	AGE versus dbh by cities . . . . .	102
9.9	AGE versus DBH by species . . . . .	104
9.10	AGE versus DBH by species for species that are in more than 1 city . . . . .	105
9.10.1	Each panel is a species, Each color is a different city . . .	106
9.10.2	Adding Loess trend lines . . . . .	107
9.11	Plot Urban Tree Allometric equations on top of data . . . . .	108
9.12	<b>NEXT</b> modelling . . . . .	117
9.12.1	lme4 . . . . .	117
<b>10</b>	<b>misc</b>	<b>127</b>
10.1	Meeting with Jun [2018-08-27 Mon] . . . . .	127

## 1 Objective

Concept: Application of bayesian multilevel modelling approach using Stan via the R package brms. Motivation: To create better allometric equations for

urban trees.

## 2 Caveat

I am not an expert on this, but I think I have a contribution to make. I'm using a dataset I did not work to collect and it may have idiosyncrasies that I may not understand. I also don't fully understand the modelling approach used to create the equations, namely the weighting.

## 3 Why needed

Whoops. They do have equations by tree type: Tree types: BDL = broadleaf deciduous large, BDM = broadleaf deciduous medium, BDS = broadleaf deciduous small, BEL = broadleaf evergreen large, BEM = broadleaf evergreen medium, BES = broadleaf evergreen small, CEL = conifer evergreen large, CEM = conifer evergreen medium, CES = conifer evergreen small, PEL = palm evergreen large, PEM = palm evergreen medium, PES= palm evergreen small.

[http://www.cpp.edu/~sagarver/GEO309/exercises/allometry/Predicting\\_Diameter\\_Height\\_Crown\\_Width\\_Leaf\\_Area\\_peper.pdf](http://www.cpp.edu/~sagarver/GEO309/exercises/allometry/Predicting_Diameter_Height_Crown_Width_Leaf_Area_peper.pdf)

The urban tree allometry dataset is an incredibly valuable resource for making better predictions about tree growth in urban environments, and it is essential for accurate ecosystem service evaluation. However, there are a number of limitations with the current set of equations that multilevel modelling can address.

Limitations:

1. Limited number of species in each region. There are only equations for the ten most common urban tree species in each region. If a user wants an equation for a different species, they either need to use an equation for that species from a different region, or select the equation from the species they think is most similar.
2. Hard boundaries of regions. There are separate allometric equations for 16 distinct climatic regions in the US. However climate varies continuously across space. Users of the equations have to pick which region they are closest to, or consider averaging two equations, but the weighting could be difficult to determine. Climatic differences have been discretized/binned into relatively few groups, when it could be more continuous.
3. Some of the existing equations are based on smaller sample sizes than reported. For example, dbh as a function of age for... only has two observations and so the best fitting equation is a straight line, but we know that isn't right.
4. used the best fitting model of several forms based on AIC (i think). This could potentially lead to overfitting, and depending on the model form selected, makes extrapolation very tenuous.

Solution: Allow for information to be shared across species and across regions when fitting models. Include climate variables as predictors in the model so that the allometric equations vary continuously across space.

A multilevel model would allow for the partial pooling of information across species and across cities, so that in the cases where we have little or even no data, we can still make reasonable predictions. If we want to make predictions about red maple in our city where we have no observations, a multilevel model would allow us to take the red maple equation from another city and adjust it to fit the climate of our city.

Improvements:

1. Climate based rather than regions based. Continuous rather than discrete, gradient rather than hard boundaries.
2. Rather than set Apps min and Apps Max, we can relax these hard boundaries, but include information about how uncertainty increases. Appsmin and Appsmax significantly truncates predictive envelope.
3. Can use an equation form that makes sense to extrapolate with (this is probably more realistic and worth the decrease in equation form flexibility).
4. Species can be nested within Genus, Genus within broader type (conifer/broadleaf/palm etc.).
1. One equation for 10 species in each region. What about species not on the list? What if I'm interested in the equation for red maple in the southeast, but the only equation comes from the northeast region? How should I adjust the equation?
2. What if my location is on the border of two regions? How should I average the equations from each region, especially if they are of two different forms (e.g. cubic and log-log)?
  - (a) this could allow for easy integration of new cities too, because right now equations for each region just come from 1 or 2 cities.
1. Currently urban tree allometric equations are built separately for each species and for each region. So there is a separate equation for red maple in the northeast and a separate equation for red maple in the midwest. These are hard differences when such hard divisions don't exist. Better models for red maple could probably be made if the different regions could be pooled to the extent that their climates are similar.
- 2.

For practitioners who wish to use

what was the metric used to select from models? was it  $r^2$ ? they used AIC shouldn't we select the model form based on expert knowledge about universal

tree growth patterns, rather than a small sample of observations? We'd expect to occasionally find with small samples that a cubic relationship best fits the data. But this relationship suffers from lack of basis in what we know about how trees grow and may give grossly inaccurate predictions if considered outside the range of the data.

[?] note on pg 130: for biomass equations: "Zianis et al. (2005) found that more than two-thirds of the equations they examined were functions of just DBH, and more than 75% of the studies that reported a sample size had less than 50 trees. The use of just DBH assumes that the relationship between DBH and height is static, which is often not the case, as noted above."

"The application to other populations of simple model forms fitted to small datasets can produce large prediction errors (e.g. Wang et al., 2002a). In addition, development of universal (Pilli et al., 2006) and generalized (Muukkonen, 2007) static equations ignores significant species variability and complex relationships, particularly when the goal is to estimate regional and national biomass (Zianis and Mancuccini, 2004)."

## 4 What should the structure of the equations be?

DBH ~ age  
assymetrical

### 4.1 Height

from [?]: For example, a cumulative growth curve of height over age shows three primary stages: (1) juvenile period where growth is rapid and often exponential; (2) a long period of maturation where the trend is nearly linear; and (3) old age, where growth is nearly asymptotic

height ~ dbh - weibull? see pg 116 [?]

Constraining the model with very low dbh's In addition to asymptotic behavior, most height-to-diameter equation forms are also constrained to predict a height of 1.3 or 1.37 (i.e. breast height) when DBH is equal to zero, but this constraint may cause poorer model performance across the full range of DBH (Newton and Ampsonah, 2007).

Hardwood heights tend to be harder to predict because of the lack of a true leader and the difficulty of measuring hardwood heights accurately (e.g. Kitahara et al., 2010).

Why Weibull is a decent option: pg116 [?]

Significant differences between model forms can exist, however, when limited data are available. For example, Temesgen and von Gadow (2004) found that the percentage difference in root mean square error between the best and worst equation for five commonly used model forms varied from 5 to 33%. Huang et al. (1992) found in their analysis that the Chapman–Richards, Weibull, and a modified logistic-type function were consistently among the best performing

models because they were flexible, able to assume a variety of shapes, and extrapolated well.

## 4.2 DBH

[?] A diameter growth curve would show much the same trend, except there is a tendency toward more sustained growth rate as the tree matures (Hann and Hanus, 2002b). While height increment may nearly cease in maturity, diameter increment must continue in order to produce the xylem and phloem needed for tree survival

## 4.3 sigmoidal forms, see the citations within:

[?] Various theoretical sigmoid model forms are used to predict growth in forestry (Zeide, 1993), but most of them can be generalized with a single equation form (Garcia, 2005c). The most common sigmoid model forms include the Gompertz (1825), Bertalanffy (1949), and Richards (1959) equations.

Although these theoretical models offer some biological interpretability (e.g. Zeide, 2004), it is easily shown that well-formulated empirical sigmoid equations can be just as accurate or even more accurate for a wide range of data (e.g. Martin and Ek, 1984)

[?]

[?]

**4.4 It would be nice to have repeat measures of trees to get diameter increments. Individual variation can be large and noisy.**

**4.5 Could the response be multivariate? For example: height, crown length, leaf area etc are covariate. Modeling them together could make sense      ATTACH**

[?] have a multivariate response The decision to fit all three-response variables simultaneously using a full covariance matrix, rather than to fit each variable separately, acknowledges the fact that response variables are likely related to each other.

We therefore expand the basic allometric model to the multivariate case

$$\begin{aligned} Y_{ij} &\sim N(X_{ij}\beta_j, \Sigma) \\ Y_{ij} &= [H_{ij}, \theta_{ij}, R_{ij}]^T \\ X_{ij} &= [1, D_{ij}]^T \end{aligned} \tag{3}$$

where  $H = \log(\text{tree height})$ ,  $\theta = \text{logit}(\text{relative canopy depth})$ ,  $R = \log(\text{canopy radius})$ ,  $D = \log(\text{DBH})$ ,  $\Sigma$  is a 3x3 covariance matrix, and relative canopy depth is defined as  $1 - (\text{height to base of crown}) / (\text{tree height})$ . The decision to fit all three-response variables simultaneously using a full covariance matrix, rather than to fit each variable separately, acknowledges the fact that response variables are likely related to each other. The use of logit

#### 4.6 [?]

see pg 168

they follow frelich 1992 look at his equation form. also

## 5 Introduction

Multilevel models have been used for decades in tree growth equations (Lappi and Bailey 1988). Indeed the test dataset, "orange", in R is used to demonstrate the fitting of nonlinear mixed effects models cite. Multilevel modeling provides a coherent framework to account for the many levels of observation or of groupings in data and to pool information across groups. This paper has two main contributions. First, we demonstrate the use of Stan via the "brms" package in R to fit bayesian nonlinear multilevel models to predict tree diameter growth from age. Second, we apply the method to the Urban Tree Database [?]. This dataset is the result an large effort to collect age and size data on thousands of trees in 17? cities across the US. Multilevel modelling has the potential to extract more information from the data and improve predictions compared to the existing modelling approach for the data. Improving predictions of tree size from tree age will improve our ability to predict the important ecosystem services these trees provide urban dwellers.

### Existing Approach

Can predict negative diameters

Tests several model forms

extrapolation to higher ages than measured is fraught because equation form determined by best fit, not biologically meaningful. A separate model for each species and city combination. In a case with 2 observations the model had to be a bit of a guess. What equation to use for unobserved species and city combinations is not clear. (e.g. should species or city be the primary predictor?) Equations derived in the 17 cities are meant to apply to work in 17 large and heterogeneous regions.

maybe the framework should be broader, just growth equations for all trees with urban tree as a case study/example of application.

Big problem - urban ecosystem services depends on trees. Predicting depends on growth narrower within - urban trees grow differently, there had been a lack of data. yet narrower summary of approach and results

1. theoretically sound model form (through use of weibull)
2. expansion of predictive envelope/potential (through use of multilevel modeling).

Objective is to show how multilevel models can be used to predict tree growth

1. Information about species functional groups and phylogeny to predict growth for unobserved species
2. Information about
3. Information about climate to predict growth in unobserved cities
- 4.

Urban trees have important ecological effects in cities that can affect human wellbeing. Forecasting ecosystem service provision by trees requires a model of tree growth, since most services are directly related to tree size (e.g. carbon storage in biomass, air filtration by leaf area). Until recently urban tree growth equations have been rare because of lacking data. In 2016 the US Forest Service published the "Urban Tree Database and Allometric Equations", a valuable dataset of the most common tree species in a dozen cities across the US. In addition to allometric equations, they also provide growth equations.

While they are a great step forward and an incredible resource, their growth equations have limitations. First, their method of model selection simply chose the curve with the lowest AIC. This meant that any amount of extrapolation was highly suspect. In the extreme case where a species only had two observations in a city, the only possible fit was a line. Second, they only fit equations for the most common species in each of the cities they sampled. It is not clear how to modify an equation to fit an unobserved species or an unobserved city. If red maple was not a species sampled in my city, should I use the red maple equation from a climatically dissimilar city? Or should I use the equation from the species

most similar to red maple from my city? What if I want to make a prediction about an unobserved species in an unobserved city?

We remove these limitations by fitting a multilevel weibull model to the data. The weibull curve has a long history of use in forestry and captures asymptotic growth well. While it may not provide the best fit to every dataset, it realistically models growth given what we know about biological constraints, is flexible, and is less unreasonable to extrapolate from. The multilevel model allows information to be shared across species and across cities, and, when combined with species' phylogenetic data and cities' climate data, allows for predictions at combinations of cities and species with little or no individual level data, provided we know genus/functional group of the species and the climate of the city.

DBH for illustrative purposes.

on selecting a model form [?]: "This reality should lead to model forms that are complex enough to accurately and adequately characterize the expected major behaviors of the population, but simple enough to avoid being overly influenced by the sample's peculiarities."

## 5.1 past work

[?]

[?]

[?]

[?]

Growth estimate of Eucalyptus stands based on nonlinear multilevel mixed-effects model theory

Miina, J., Heinonen, J., 2008. Stochastic simulation of forest regeneration establishment using a multilevel multivariate model. *Forest Science* 54, 206–219.

Miina, J., Saksu, T., 2006. Predicting regeneration establishment in Norway spruce plantations using a multivariate multilevel model. *New Forests* 32, 265–283.

Uzoh, F.C.C., Oliver, W.W., 2008. Individual tree diameter increment model for managed even-aged stands of ponderosa pine throughout the western United States using a multilevel linear mixed effects model. *Forest Ecology and Management* 256, 438–445.

Kershaw Jr., J.A., Benjamin, J.G., Weiskittel, A.R., 2009. Approaches for modeling vertical distribution of maximum knot size in black spruce: A comparison of fixed- and mixed-effects nonlinear models. *Forest Science* 55, 230–237.

Gregoire, T.G., Schabenberger, O., 1996. A non-linear mixed-effects model to predict cumulative bole volume of standing trees. *Journal of Applied Statistics* 23, 257–271.

Leites, L.P., Robinson, A.P., 2004. Improving taper equations of loblolly pine with crown dimensions in a mixed-effects modeling framework. *Forest Science* 50, 204–212.

Sharma, M., Parton, J., 2007. Height-diameter equations for boreal tree species in Ontario using a mixed-effects modeling approach. *Forest Ecology*

and Management 249, 187–198.

Wang, Y., LeMay, V.M., Baker, T.G., 2007b. Modelling and prediction of dominant height and site index of *Eucalyptus globulus* plantations using a nonlinear mixed-effects model approach. Canadian Journal of Forest Research 37, 1390–1403.

Urban Tree Growth Modeling [https://www.fs.fed.us/psw/publications/mcpherson/psw\\_2012\\_mcpherson001.pdf](https://www.fs.fed.us/psw/publications/mcpherson/psw_2012_mcpherson001.pdf)

Urban Tree Database and Allometric Equations E. Gregory McPherson, Natalie S. van Doorn, and Paula J. Peper

The economic, social, and ecological benefits of trees are directly related to their size, as indicated by leaf area, crown volume, and bio-mass (Scott et al. 1998, Stoffberg et al. 2010, Xiao et al. 2000a). Growth equations underpin the calculations produced by many computer models used in urban forestry, such as i-Tree, National Tree Benefit Calculator, OpenTreeMap, and ecoSmart Landscapes (fig. 1).

Quotes from the utd publication:

The development of allometric equations for urban open-grown trees has been sporadic. Fleming (1988) measured trees in New Jersey having full healthy crowns to develop linear relationships between d.b.h., height, crown spread, and age. Frelich (1992) measured only healthy trees (12 species, 221 trees total) growing in Minneapolis and St. Paul, Minnesota, to predict linear size relationships. Nowak (1994b) developed an allometric equation for leaf area based on data from park trees in Chicago. Tree dimensions and leaf area were predicted for the most abundant street tree species in Modesto and Santa Monica, California (Peper et al. 2001a, 2001b). In New Haven, Connecticut, Troxel et al. (2013) developed allometric equations for predicting d.b.h. from age and height, crown diameter, and crown volume from d.b.h. for early growth (15 years) of 10 street tree species. Outside of North America, growth equations have been developed for street-side *Tilia* species in Copenhagen, Denmark (Larsen and Kristoffersen 2002), and *T. cordata* Mill., *Fraxinus excelsior* L. and *Aesculus hippocastanum* L. in Warsaw, Poland (Lukaszewicz and Kosmala 2008, Lukaszewicz et al. 2005). Predictive models were developed from allometric data for five street tree species in northeastern Italy by Semenzato et al. (2011). Stoffberg et al. (2008) used allometric relationships between age and d.b.h., height, and crown diameter to estimate dimensions at 10, 15, and 30 years after planting for three street tree species in Tshwane, South Africa. The allometric equations from all these studies reflect the effects of local site conditions, management practices, and growing season on growth, limiting application outside their region of origin (fig. 5).

## 5.2 urban trees equations

"Growth equations for forest trees may not be directly transferable to open-grown urban trees because they grow and partition bole, branch, twig, and leaf biomass differently (Anderegg et al. 2015, Nowak 1994a, Peper and McPherson 1998) (fig. 4). For example, in forests, tree crowns compete for limited space and may not reach their maximum expansion potential (Martin et al. 2012)." from the

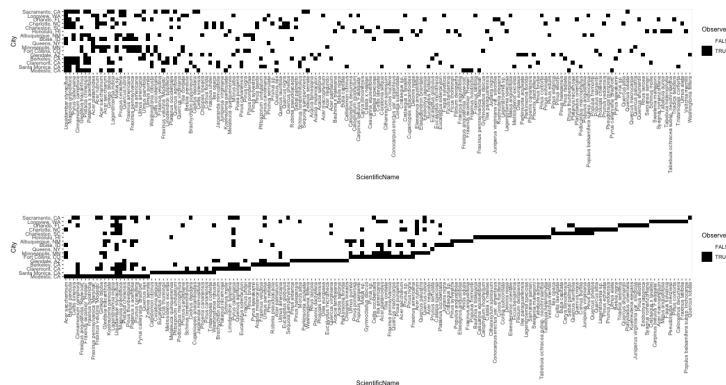
## 5.3 past modelling

This report presents the third, most recent and most complete sets of growth equations. The equations presented in this report were developed using more sophisticated statistical methods than before. For example, in the first studies, logarithmic regression and exponential models predominantly provided the best fits to measured data (Peper et al. 2001a, 2001b). In these equations, the best model fits ranged from polynomials (from simple linear to quartic) to logarithmic and exponential models (Peper et al. 2014). The newest equations have been integrated with numerical models of tree benefits in the ecoSmart Landscapes software (McPherson et al. 2014).

# 6 Methods

## 6.1 Data

The urban tree database consists of measurements on 14487 trees of 170 species in 17 cities. However, largely because of the difficulty is measuring tree age, there are only 12687 trees with complete age and diameter data (161 species, 17 cities, 309 species x city combinations). Some species were measured in multiple cities, but not most.



repeat measures on the same individuals would help much.

I wonder if the UFIA would have this eventually and if I could write the code to incorporate the data.

In their dataset, AGE refers to years after planting. Not true age. This adds some noise and is the reason why in the dataset trees can have significant dbh at "age" 0.

"AGE" is not actually known. A modelling extension could incorporate that uncertainty. How would it be down? multivariate?

uncertainty about age:

Records of planting dates seldom extend beyond 30 to 40 years. Similarly, detecting the presence and size of individual trees using high-resolution aerial imagery becomes difficult prior to 1990. As a result, predictions of urban tree dimensions reflect the increasing uncertainty about true tree age compounded by naturally increasing variability associated with aging (fig. 8).

dbh is cm

utd equations sometimes predict negative values. see top of page 25.  
Think about Apps Min and Apps Max

### 6.1.1 Tidy data for this analysis

1. load libraries
2. functions

```
options(asciiType = "org")
ascii.nowarn.print <- function(x,...) {
  #op <- options(warn = -1)
  #      on.exit(options(op))

  suppressWarnings(print(ascii(x,...)))
}

3. read in data data pdf: https://www.fs.fed.us/psw/publications/documents/psw\_gtr253/psw\_gtr\_253.pdf data webpage: https://www.fs.usda.gov/rds/archive/Product/RDS-2016-0005

d <- read_csv("../data/RDS-2016-0005/Data/TS3_Raw_tree_data.csv")

str(d)

Classes 'tbl_df', 'tbl' and 'data.frame': 14487 obs. of  41 variables:
 $ DbaseID      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Region       : chr  "InlVal" "InlVal" "InlVal" "InlVal" ...
 $ City         : chr  "Modesto, CA" "Modesto, CA" "Modesto, CA" "Modesto, CA" ...
```

```

$ Source      : chr  "Motown2.xls: Completed Data" "Motown2.xls: Completed Data" "Mo
$ TreeID     : int   1 2 3 4 5 6 7 8 9 10 ...
$ Zone        : chr  "Nursery" "Nursery" "Nursery" "Nursery" ...
$ Park/Street : chr  "Nursery" "Nursery" "Nursery" "Nursery" ...
$ SpCode      : chr  "ACSA1" "BEPE" "CESI4" "CICA" ...
$ ScientificName: chr  "Acer saccharinum" "Betula pendula" "Celtis sinensis" "Cinnamom
$ CommonName   : chr  "Silver maple" "European white birch" "Chinese hackberry" "Camp
$ TreeType    : chr  "BDL" "BDM" "BDL" "BEM" ...
$ address     : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ street       : chr  "Nursery" "Nursery" "Nursery" "Nursery" ...
$ side         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ cell         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ OnStreet    : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ FromStreet   : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ ToStreet    : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ Age          : int   0 0 0 0 0 0 0 0 0 ...
$ DBH (cm)    : num   2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
$ TreeHt (m)  : num   2 1.5 1.8 2 2 2 2 2 2 1.6 ...
$ CrnBase     : num   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ CrnHt (m)   : num   0.5 0.8 0.6 0.9 0.9 0.8 0.8 0.8 0.8 ...
$ CdiaPar (m) : num   1 0.6 0.7 1 1 0.8 0.8 0.8 1 0.7 ...
$ CDiaPerp (m): num   1 0.6 0.7 1 1 0.8 0.8 0.8 1 0.7 ...
$ AvgCdia (m) : num   1 0.6 0.7 1 1 0.8 0.8 0.8 1 0.7 ...
$ Leaf (m2)   : num   2.5 1.9 2.2 2 2.2 2.2 2.2 2.2 2.1 1.3 ...
$ Setback     : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ TreeOr      : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ CarShade    : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ LandUse     : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ Shape        : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ WireConf    : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh1         : num   2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
$ dbh2         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh3         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh4         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh5         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh6         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh7         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh8         : int   -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
- attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame':
24255 obs. of  5 variables:
..$ row      : int  1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 ...
..$ col      : chr  "side" "side" "side" "side" ...
..$ expected: chr  "an integer" "an integer" "an integer" "an integer" ...
..$ actual   : chr  "C" "C" "C" "C" ...
..$ file     : chr  '../data/RDS-2016-0005/Data/TS3_Raw_tree_data.csv' '../data/RDS
- attr(*, "spec")=List of 2

```

```

..$ cols      :List of 41
.. ..$ DbaseID       : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ Region        : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ City          : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ Source         : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ TreeID         : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ Zone           : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ Park/Street    : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ SpCode          : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ ScientificName: list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ CommonName     : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ TreeType        : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ address         : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ street          : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. ..$ side            : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ cell             : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ OnStreet        : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ FromStreet      : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ ToStreet         : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ Age              : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. ..$ DBH (cm)        : list()
.. . . . . attr(*, "class")= chr "collector_double" "collector"
.. ..$ TreeHt (m)      : list()
.. . . . . attr(*, "class")= chr "collector_double" "collector"
.. ..$ CrnBase         : list()
.. . . . . attr(*, "class")= chr "collector_double" "collector"
.. ..$ CrnHt (m)       : list()

```

```

... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ CdiaPar (m) : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . . $ CDiaPerp (m) : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . . . $ AvgCdia (m) : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . . . $ Leaf (m2) : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . . . $ Setback : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ TreeOr : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ CarShade : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ LandUse : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ Shape : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ WireConf : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh1 : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . . . $ dbh2 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh3 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh4 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh5 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh6 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh7 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . . . $ dbh8 : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... $ default: list()
... . . . attr(*, "class")= chr "collector_guess" "collector"
... - attr(*, "class")= chr "col_spec"

```

#### 4. explanation of variables from metadata

DbaseID = Unique id number for each tree.

Region = 16 U.S. climate regions, abbreviations are used (see 1Regionalinformation.csv).

City = City/state names where data collected.

Source = Original \*.xls filename (not available in this data publication).

TreeID = Number assigned to each tree in inventory by city.

Zone = Number/ID/name of the management area or zone that the tree is located in within a city; or nursery if young tree data collected there.

Park/Street = Data listed as Park, Street, Regional Big Tree, or Nursery (for young tree measurements).

SpCode = 4 to 6 letter code consisting of the first two letters of the genus name and the first two letters of the species name followed by two optional letters to distinguish two species with the same four-letter code (See [2Regionalspeciesandcounts.csv](#) for a list of the SpCodes and corresponding scientific names.)

ScientificName = Botanical name of species.

CommonName = Common name of species.

Tree Type = 3 letter code where first two letters refer to life form (BD=broadleaf deciduous, BE=broadleaf evergreen, CE=coniferous evergreen, PE=palm evergreen) and the third letter is mature height (S=small which is < 8 meters, M=medium which is 8-15 meters, and L=large which is > 15 meters).

Address = From inventory, street number of building where tree is located.

Street = From inventory, the name of the street the tree is located on. (NOTE: zero values denote data were not recorded in that city. These values were left unchanged because they originated from city inventories.)

Side = From inventory, side of building or lot tree is located on (F=front, M=median, S=side, P=park). (NOTE: zero values denote data were not recorded in that city. These values were left unchanged because they originated from city inventories.)

Cell = From inventory, the cell number (i.e., 1, 2, 3, ...), where protocol determines the order trees at same address are numbered (e.g., driving direction or as street number increases).

OnStreet = From inventory (omitted if not a field in city's inventory), for trees at corner addresses when tree is on cross street rather than addressed street.

FromStreet = From inventory, the name of the first cross street that forms a boundary for trees lining un-addressed boulevards. Trees are typically numbered in order (1, 2, 3 ...) on boulevards that have no development adjacent to them, no obvious parcel addresses.

ToStreet = From inventory, the name of the last cross street that forms a boundary for trees lining un-addressed boulevards.

Age = Number of years since planted. (NOTE: zero values represent newly planted trees, < 1 year old.)

DBH (cm) = Diameter at breast height (1.37 meters [m]) measured to nearest 0.1 centimeters (tape). For multi-stemmed trees forking below 1.37 m measured above the butt flare and below the point where the stem begins forking, as per protocol.

TreeHt (m) = From ground level to tree top to nearest 0.5 m (omitting erratic leader).

CrnBase (m) = Average distance between ground and lowest foliage layer to nearest 0.5 m (omitting erratic branch).

CrnHt (m) = Calculated as TreeHT minus Crnbase to nearest 0.5 m. (NOTE: zero values indicate no live crown was present, hence no other tree dimension data were available.)

CdiaPar (m) = Crown diameter measurement taken to the nearest 0.5 m parallel to the street (omitting erratic branch).

CDiaPerp (m) = Crown diameter measurement taken to the nearest 0.5 m perpendicular to the street (omitting erratic branch).

AvgCdia (m) = The average of crown diameter measured parallel and perpendicular to the street.

Leaf (m<sup>2</sup>) = Estimated using digital imaging method to nearest 0.1 squared meter (m<sup>2</sup>).

Setback = Distance from tree to nearest air-conditioned/heated space (may not be same address as tree location): 1=0-8 m, 2=8.1-12 m, 3=12.1-18 m, 4=> 18 m.

TreeOr = Taken with compass, the coordinate of tree taken from imaginary lines extending from walls of the nearest conditioned space (may not be same address as tree location).

CarShade = Number of parked automotive vehicles with some part under the tree's drip line. Car must be present (0=no autos, 1=1 auto, etc.).

LandUse = Predominant land use type where tree is growing (1=single family residential, 2=multi-family residential [duplex, apartments, condos], 3=industrial/institutional/large commercial [schools, gov't, hospitals], 4=park/vacant/other [agric., unmanaged riparian areas of greenbelts], 5=small commercial [minimart, retail boutiques, etc.], 6=transportation corridor).

Shape = Visual estimate of crown shape verified from each side with actual measured dimensions of crown height and average crown diameter (1=cylinder [maintains same crown diameter in top and bottom thirds of tree], 2=ellipsoid, the tree's center [whether vertical or horizontal is the widest, includes spherical], 3=paraboloid [widest in bottom third of crown], 4=upside down paraboloid [widest in top third of crown]).

WireConf = Utility lines that interfere with or appear above tree (0=no lines, 1=present and no potential conflict, 2=present and conflicting, 3=present and potential for conflicting). (NOTE: -1 denotes data were not collected.)

dbh1 = Dbh (centimeters [cm]) for multi-stemmed trees; for non-multi-stemmed trees, dbh1 is same as Dbh (cm).

dbh2 = Dbh (cm) for second stem of multi-stemmed trees.

dbh3 = Dbh (cm) for third stem of multi-stemmed trees.

dbh4 = Dbh (cm) for fourth stem of multi-stemmed trees.

dbh5 = Dbh (cm) for fifth stem of multi-stemmed trees.

dbh6 = Dbh (cm) for sixth stem of multi-stemmed trees.

dbh7 = Dbh (cm) for seventh stem of multi-stemmed trees.

dbh8 = Dbh (cm) for eight stem of multi-stemmed trees.

5. fix some species things

(a) change lower case species codes

```
d$SpCode <- toupper(d$SpCode)
```

(b) change QUAG1 to be QUAG

```
d$SpCode[d$SpCode == "QUAG1"] <- "QUAG"
```

(c) fix common names

There may be other common names I need to fix.

```
d$CommonName[d$CommonName == "Kurrajong"] <- "Kurrajong/Bottle tree"  
d$CommonName[d$CommonName == "Bottle tree"] <- "Kurrajong/Bottle tree"
```

```
d$CommonName[d$CommonName == "Apple"] <- "Apple/Crabapple"  
d$CommonName[d$CommonName == "Crabapple"] <- "Apple/Crabapple"
```

```
d$CommonName[d$CommonName == "silver maple"] <- "Silver maple"  
d$CommonName[d$CommonName == "camphor tree"] <- "Camphor tree"  
d$CommonName[d$CommonName == "ginkgo"] <- "Ginkgo"  
d$CommonName[d$CommonName == "honeylocust"] <- "Honeylocust"  
d$CommonName[d$CommonName == "ginkgo"] <- "Ginkgo"  
d$CommonName[d$CommonName == "common crapemyrtle"] <- "Common crapemyrtle"  
d$CommonName[d$CommonName == "sweetgum"] <- "Sweetgum"  
d$CommonName[d$CommonName == "southern magnolia"] <- "Southern magnolia"
```

(d) change scientific names (remove cultivated variety)

```
d$ScientificName[d$ScientificName == "Prunus cerasifera cvs."] <- "Prunus cerasifera"  
d$ScientificName[d$ScientificName == "Pyrus calleryana cvs."] <- "Pyrus calleryana"
```

6. tidy a few names and select variables of interest here

```
d <- d %>%
  rename(DBH = 'DBH (cm)', Leaf = 'Leaf (m2)', Species = SpCode) %>% select(Region, C
```

7. Remove missing data (either DBH or Age)

```
d <- filter(d, DBH != -1, Age != -1) %>%
  rename(AGE = Age)
```

8. save data

```
saveRDS(d, ".../data/tidy_age_dbh.rds")
```

9. str of data now

```
d <- readRDS("../data/tidy_age_dbh.rds")
str(d)

Classes 'tbl_df', 'tbl' and 'data.frame': 12687 obs. of  8 variables:
$ Region      : chr "InlVal" "InlVal" "InlVal" "InlVal" ...
$ City        : chr "Modesto, CA" "Modesto, CA" "Modesto, CA" "Modesto, CA" ...
$ TreeID      : int 1 2 3 4 5 6 7 8 9 10 ...
$ Species     : chr "ACSA1" "BEPE" "CESI4" "CICA" ...
$ DBH         : num 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
$ Leaf         : num 2.5 1.9 2.2 2.2 2.2 2.2 2.2 2.2 2.1 1.3 ...
$ AGE          : int 0 0 0 0 0 0 0 0 0 0 ...
$ ScientificName: chr "Acer saccharinum" "Betula pendula" "Celtis sinensis" "Cinnamom
- attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame':
24255 obs. of  5 variables:
..$ row      : int 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 ...
..$ col      : chr "side" "side" "side" "side" ...
..$ expected: chr "an integer" "an integer" "an integer" "an integer" ...
..$ actual   : chr "C" "C" "C" "C" ...
..$ file     : chr "'.../data/RDS-2016-0005/Data/TS3_Raw_tree_data.csv'" "'.../data/RDS
- attr(*, "spec")=List of 2
..$ cols    :List of 41
.. . . $ DbaseID      : list()
.. . . . . attr(*, "class")= chr "collector_integer" "collector"
.. . . $ Region       : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. . . $ City         : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. . . $ Source        : list()
.. . . . . attr(*, "class")= chr "collector_character" "collector"
.. . . $ TreeID       : list()
```

```

... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ Zone : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ Park/Street : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ SpCode : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ ScientificName: list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ CommonName : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ TreeType : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ address : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ street : list() "collector_character" "collector"
... . . . . attr(*, "class")= chr "collector_character" "collector"
... . $ side : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ cell : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ OnStreet : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ FromStreet : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ ToStreet : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ Age : list() "collector_integer" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ DBH (cm) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ TreeHt (m) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ CrnBase : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ CrnHt (m) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ CdiaPar (m) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ CDiaPerp (m) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ AvgCdia (m) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ Leaf (m2) : list() "collector_double" "collector"
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ Setback : list() "collector_double" "collector"

```

```

... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ TreeOr      : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ CarShade    : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ LandUse     : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ Shape       : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ WireConf    : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh1        : list()
... . . . . attr(*, "class")= chr "collector_double" "collector"
... . $ dbh2        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh3        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh4        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh5        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh6        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh7        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... . $ dbh8        : list()
... . . . . attr(*, "class")= chr "collector_integer" "collector"
... $ default: list()
... . . . attr(*, "class")= chr "collector_guess" "collector"
... . attr(*, "class")= chr "col_spec"

```

10. save subset of data for testing

```

d <- readRDS("../data/tidy_age_dbh.rds")
clim <- read.csv("../data/cities_gdd.csv", stringsAsFactors = F) %>%
  select(-X)

species.to.filter <- c("LIST", "MAGR", "PYCA", "CICA", "GLTR", "PICH", "PLAC", "ACPL")

d2 <- d %>%
  mutate(Genus = stringr::str_extract(ScientificName, '\w*')) %>%
  select(-Leaf,-TreeID) %>%
  left_join(.,clim) %>%
  mutate(Precip = round((Precip - mean(Precip))/ 1000, 4),
         GDD = round((GDD - mean(GDD))/ 1000, 4))

```

```

    saveRDS(d2, "../data/age_dbh_full.rds")

d3 <- d2 %>%
  filter(Species %in% species.to.filter)

  saveRDS(d3, "../data/age_dbh_testing.rds")

Joining, by = "City"

```

11. send to krusty

```

rsync -avz ../data/age_dbh_full.rds erker@krusty:~/allo/data/
rsync -avz ../data/age_dbh_testing.rds erker@krusty:~/allo/data/

```

### 6.1.2 The Species for which we have age and dbh:

```

sampled_species <- readRDS("../data/tidy_age_dbh.rds") %>%
  pull(ScientificName) %>%
  unique %>%
  as.character() %>%
  sort %>%
  data.frame()
nrow(sampled_species)
sampled_species %>% ascii.nowarn.print

[1] 161
|   |
|-----+
| 1 | Acacia farnesiana
| 2 | Acacia melanoxylon
| 3 | Acacia salicina
| 4 | Acer macrophyllum
| 5 | Acer negundo
| 6 | Acer palmatum
| 7 | Acer platanoides
| 8 | Acer rubrum
| 9 | Acer saccharinum
| 10 | Acer saccharum
| 11 | Bauhinia x blakeana
| 12 | Betula nigra
| 13 | Betula pendula
| 14 | Brachychiton populneum
| 15 | Butia capitata
| 16 | Callistemon citrinus
| 17 | Calocedrus decurrens

```

18	<i>Calophyllum inophyllum</i>	
19	<i>Carpinus betulus</i> 'Fastigiata'	
20	<i>Carya illinoinensis</i>	
21	<i>Cassia x nealiae</i>	
22	<i>Casuarina equisetifolia</i>	
23	<i>Catalpa speciosa</i>	
24	<i>Cedrus deodara</i>	
25	<i>Celtis laevigata</i>	
26	<i>Celtis occidentalis</i>	
27	<i>Celtis sinensis</i>	
28	<i>Ceratonia siliqua</i>	
29	<i>Chilopsis linearis</i>	
30	<i>Cinnamomum camphora</i>	
31	<i>Citharexylum spinosum</i>	
32	<i>Cocos nucifera</i>	
33	<i>Conocarpus erectus</i> var. <i>argenteus</i>	
34	<i>Cordia subcordata</i>	
35	<i>Cornus florida</i>	
36	<i>Crataegus</i> sp.	
37	<i>Crataegus x lavallei</i>	
38	<i>Cupaniopsis anacardiooides</i>	
39	<i>Delonix regia</i>	
40	<i>Elaeagnus angustifolia</i>	
41	<i>Elaeodendron orientale</i>	
42	<i>Eriobotrya japonica</i>	
43	<i>Eucalyptus ficifolia</i>	
44	<i>Eucalyptus globulus</i>	
45	<i>Eucalyptus microtheca</i>	
46	<i>Eucalyptus sideroxylon</i>	
47	<i>Fagus sylvatica</i>	
48	<i>Ficus benjamina</i>	
49	<i>Ficus thonningii</i>	
50	<i>Filicium decipiens</i>	
51	<i>Fraxinus americana</i>	
52	<i>Fraxinus angustifolia</i>	
53	<i>Fraxinus angustifolia</i> 'Raywood'	
54	<i>Fraxinus excelsior</i> 'Hessei'	
55	<i>Fraxinus holotricha</i>	
56	<i>Fraxinus latifolia</i>	
57	<i>Fraxinus pennsylvanica</i>	
58	<i>Fraxinus pennsylvanica</i> 'Marshall'	
59	<i>Fraxinus uhdei</i>	
60	<i>Fraxinus velutina</i>	
61	<i>Fraxinus velutina</i> 'Modesto'	
62	<i>Ginkgo biloba</i>	
63	<i>Gleditsia triacanthos</i>	

64	Gymnocladus dioicus
65	Ilex opaca
66	Ilex paraguariensis
67	Jacaranda mimosifolia
68	Juglans nigra
69	Juniperus virginiana
70	Juniperus virginiana var. silicicola
71	Koelreuteria elegans
72	Koelreuteria paniculata
73	Lagerstroemia indica
74	Lagerstroemia sp.
75	Lagerstroemia speciosa
76	Liquidambar styraciflua
77	Liriodendron tulipifera
78	Magnolia grandiflora
79	Malus angustifolia
80	Malus sp.
81	Melaleuca quinquenervia
82	Metrosideros excelsa
83	Morus alba
84	Olea europaea
85	Parkinsonia aculeata
86	Parkinsonia florida
87	Phoenix canariensis
88	Phoenix dactylifera
89	Picea pungens
90	Pinus brutia
91	Pinus canariensis
92	Pinus contorta
93	Pinus echinata
94	Pinus edulis
95	Pinus eldarica
96	Pinus elliottii
97	Pinus halepensis
98	Pinus nigra
99	Pinus ponderosa
100	Pinus radiata
101	Pinus sylvestris
102	Pinus taeda
103	Pinus thunbergiana
104	Pistacia chinensis
105	Pittosporum undulatum
106	Platanus occidentalis
107	Platanus racemosa
108	Platanus x acerifolia
109	Platycladus orientalis

110	Podocarpus macrophyllus
111	Populus angustifolia
112	Populus balsamifera subsp. trichocarpa
113	Populus fremontii
114	Populus sargentii
115	Prosopis chilensis
116	Prunus caroliniana
117	Prunus cerasifera
118	Prunus serrulata
119	Prunus sp.
120	Prunus yedoensis
121	Pseudotsuga menziesii
122	Pyrus calleryana
123	Pyrus calleryana 'Bradford'
124	Pyrus kawakamii
125	Pyrus sp.
126	Quercus agrifolia
127	Quercus alba
128	Quercus ilex
129	Quercus laurifolia
130	Quercus lobata
131	Quercus macrocarpa
132	Quercus nigra
133	Quercus palustris
134	Quercus phellos
135	Quercus rubra
136	Quercus shumardii
137	Quercus virginiana
138	Rhus lancea
139	Robinia pseudoacacia
140	Sabal palmetto
141	Samanea saman
142	Schinus molle
143	Schinus terebinthifolius
144	Sequoia sempervirens
145	Swietenia mahagoni
146	Syagrus romanzoffiana
147	Tabebuia aurea
148	Tabebuia heterophylla
149	Tabebuia ochracea subsp. neochrysantha
150	Tilia americana
151	Tilia cordata
152	Triadica sebifera
153	Tristaniopsis conferta
154	Ulmus alata
155	Ulmus americana

156	<i>Ulmus parvifolia</i>	
157	<i>Ulmus pumila</i>	
158	<i>Veitchia merrillii</i>	
159	<i>Washingtonia filifera</i>	
160	<i>Washingtonia robusta</i>	
161	<i>Zelkova serrata</i>	

161 Species

### 6.1.3 Cities and Climate

Region	City	CDD	HDD	Precip	Longitude	Latitude
CenFla	Orlando, FL	1806	289	1367	-81.37924	28.53834
GulfCo	Charleston, SC	1124	1221	1555	-79.9311	32.7765
InlEmp	Claremont, CA	134	872	523	-117.7198	34.0967
InlVal	Modesto, CA	1052	1439	315	-120.99688	37.63910
SacVal	Sacramento, CA	773	1718	470	-121.49440	38.58157
InterW	Albuquerque, NM	677	2416	250	-106.60555	35.08533
LoMidW	Indianapolis, IN	510	3153	392	-86.15807	39.76840
MidWst	Minneapolis, MN	355	4436	622	-93.26501	44.97775
NMtnPr	Fort Collins, CO	349	3332	452	-105.08442	40.58526
NoCalc	Berkeley, CA	39	1786	564	-122.27275	37.87159
NoEast	Queens, NY	560	2819	1041	-73.7949	40.7282
PacfNW	Longview, WA	157	2468	1059	-94.74049	32.50070
Piedmt	Charlotte, NC	847	1891	1426	-80.84313	35.22709
SoCalC	Santa Monica, CA	266	710	570	-118.49119	34.01945
SWDsrt	Glendale, AZ	2128	637	174	-112.1860	33.5387
TpIntW	Boise, ID	387	3325	417	-116.2023	43.6150
Tropic	Honolulu, HI	2416	0	2206	-157.85833	21.30694

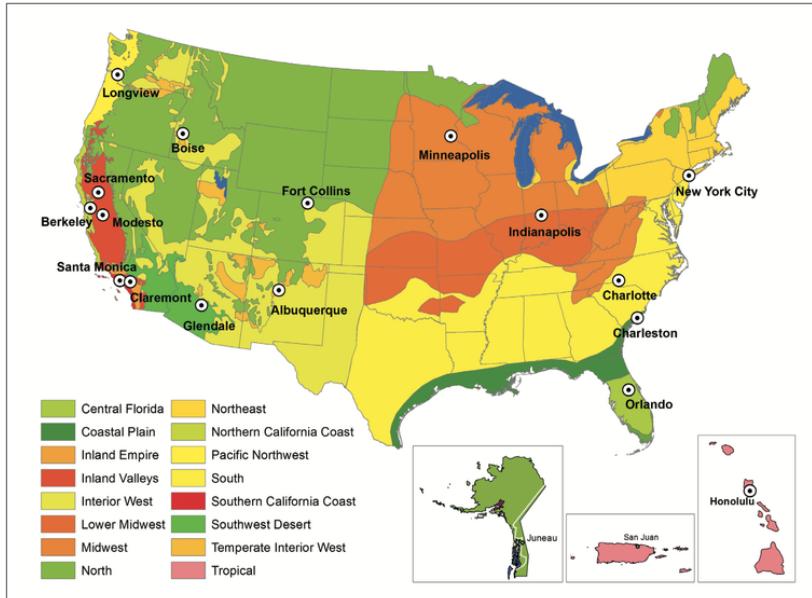


Figure 9—Climate zones were aggregated from 45 Sunset climate zones into 16 zones. Each zone has a reference city where tree growth data were collected. Sacramento, California, was added as a second reference city (with Modesto) to the Inland Valleys zone.

The ironic thing about this figure is that Fort Collins is the "North". This includes a great variety of areas including Cheyenne, WY, which according to figure 5 (McPherson and Peper 2012) in the same document have green ash trees that have 55% of the leaf area of Fort Collins green ash at age 60. This incredible variability within regions, hence the need for more continuous approach. Notice how often the reference city is on the border of the climate zone.

Get a figure showing the climate of census tracts. Basically remake the above figure to be more continuous

Make a figure showing the reference cities in GDD and Precip space to reveal where there are significant holes that could be filled. What future cities to sample.

When I make the marginal effects plot of GDD versus Precip, I should make the plot cover the values seen in the US.

#### 6.1.4 Get new climate data. Growing degree days and Precip. HDD is super high for minneapolis, but harsh winters aren't relevant.

```
wget -O ./data/gdd.txt https://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/supplemental/p
wget -O ./data/precip.txt https://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/products/pr
wget -O ./data/temp-station-info.txt https://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/
```

```
library(ggplot2)
library(dplyr)
```

```

library(tidyr)
library(stringr)
d <- read.table("../data/gdd.txt")

colnames(d) <- c("station", "grdd")

d <- d %>%
  mutate(gdd = as.numeric(str_match(grdd, "-*[0-9]+") [,1]),
         qual = str_match(grdd, "[A-Z]") [,1])
head(d)
dim(d)

d <- d %>%
  filter(qual %in% c("C", "S", "R"),
         gdd >= 0)
dim(d)

ll <- read.table("../data/temp-station-info.txt", fill = T, stringsAsFactors = F)
ll <- ll[,1:3] # get station, lat, and long
colnames(ll) <- c("station", "lat", "long")

ll <- ll %>%
  mutate(station = as.character(station),
         lat = as.numeric(lat))

dl <- left_join(d, ll)

write.csv(dl, "../data/gdd_qt_ll.csv")

p <- read.table("../data/precip.txt")
colnames(p) <- c("station", "precip_qual")

p <- p %>%
  mutate(precip = as.numeric(str_match(precip_qual, "[0-9]+") [,1]),
         qual = str_match(precip_qual, "[A-Z]") [,1])
head(p)
dim(p)

p <- p %>%
  filter(qual %in% c("C", "S", "R"),
         precip >= 0)
dim(p)

pl <- left_join(p, ll)

```

```

write.csv(pl, "../data/gdd_qt_ll.csv")

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

  filter, lag

The following objects are masked from 'package:base':

  intersect, setdiff, setequal, union
  station  grdd  gdd qual
1 AQW00061705 12073C 12073    C
2 CAW00064757 2636Q 2636    Q
3 CQC00914080 11168P 11168    P
4 CQC00914801 11656R 11656    R
5 FMC00914395 11423P 11423    P
6 FMC00914419 11860P 11860    P
[1] 7501    4
[1] 6340    4
Warning message:
In evalq(as.numeric(lat), <environment>) : NAs introduced by coercion
Joining, by = "station"
Warning message:
Column 'station' joining factor and character vector, coercing into character vector
  station precip_qual precip qual
1 AQC00914000      21392R 21392    R
2 AQW00061705      12263C 12263    C
3 CAW00064757      3172Q 3172    Q
4 CQC00914080      8339P 8339    P
5 CQC00914801      9124R 9124    R
6 CQC00914855      6976P 6976    P
[1] 9307    4
[1] 7440    4
Joining, by = "station"
Warning message:
Column 'station' joining factor and character vector, coercing into character vector

library(sp)
library(jsonlite)
library(dplyr)

cities <- city_climate %>%

```

```

    select(City, Latitude, Longitude, Precip)

coordinates(cities) <- ~ Longitude + Latitude
proj4string(cities) <- CRS("+init=epsg:4326")

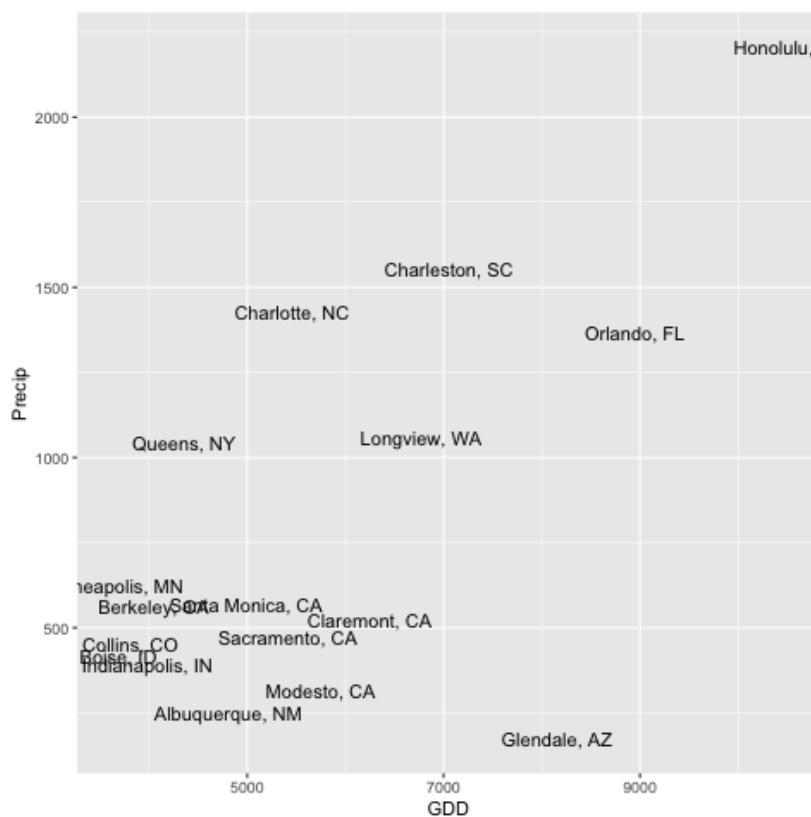
gdd <- read.csv("../data/gdd_qt_ll.csv", stringsAsFactors =F)
coordinates(gdd) <- ~long + lat
proj4string(gdd) <- CRS("+init=epsg:4326")

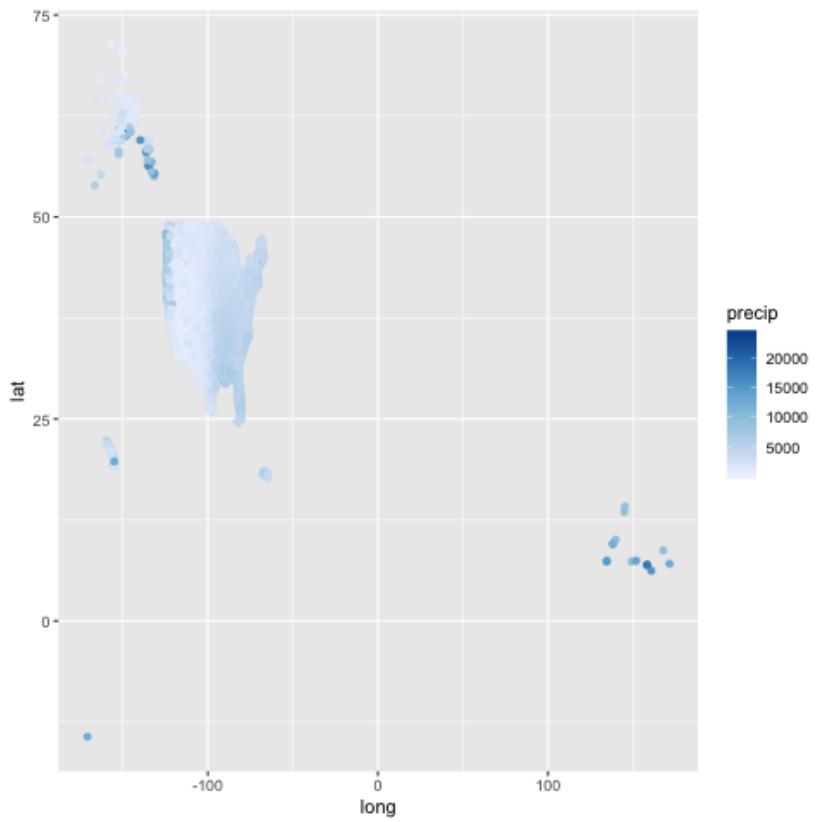
gdd.dists <- spDists(cities, gdd, longlat = T)
gdd.dists.min <- apply(gdd.dists, 1, function(x) which(x == min(x))[1])
gdd <- gdd[gdd.dists.min,]

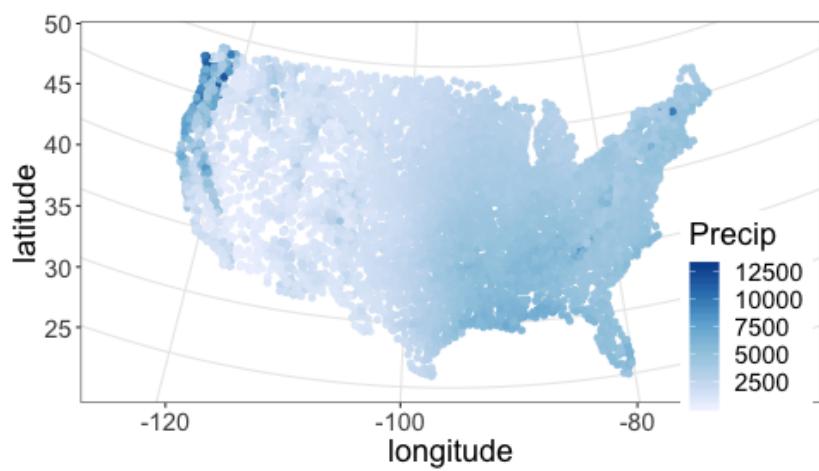
d <- bind_cols(gdd@data,cities@data, data.frame(coordinates(cities))) %>%
  select(gdd, Precip, City, Longitude, Latitude) %>%
  rename(GDD = gdd)

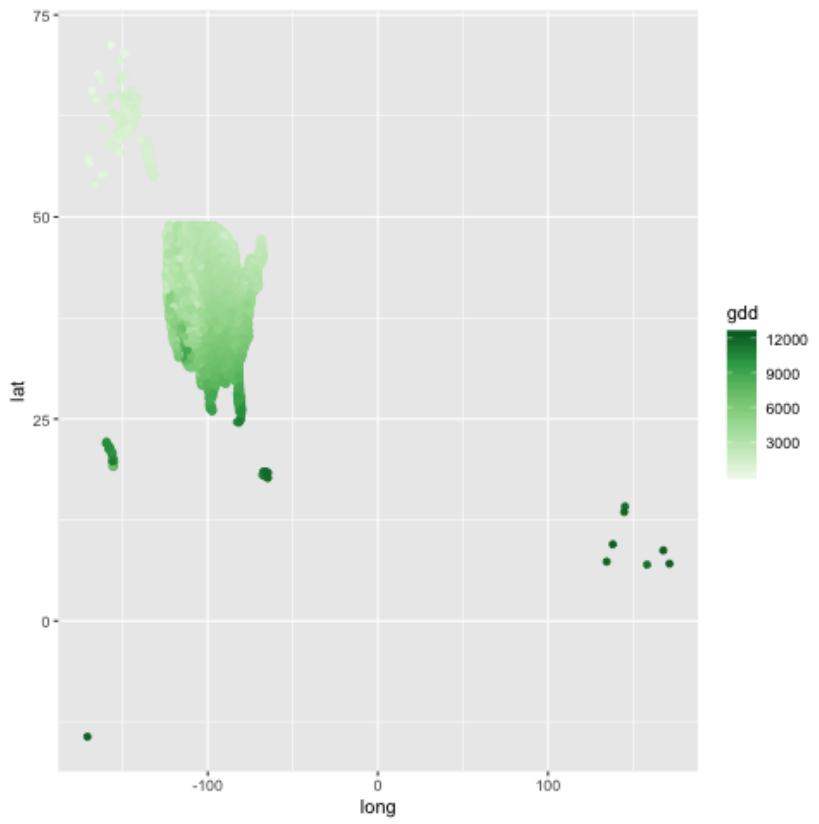
write.csv(d, "../data/cities_gdd.csv")

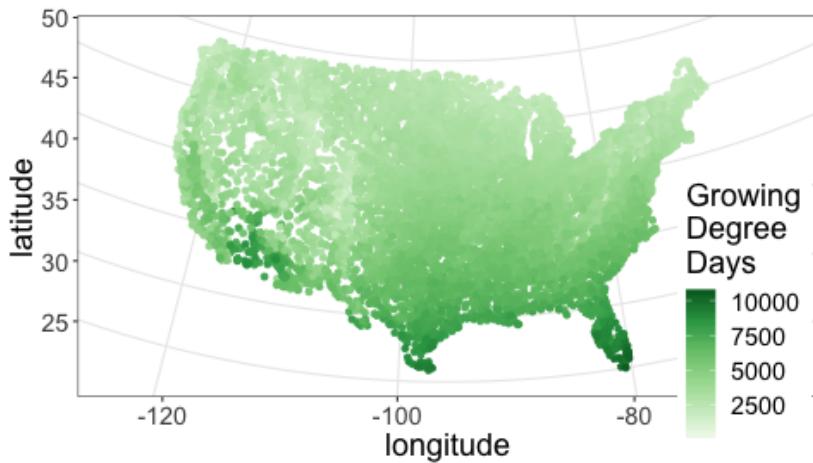
```











Get census tract centroids and join

```
mkdir ../data/census_centroid_pop
wget -O census_centroid_pop.zip http://faculty.baruch.cuny.edu/geoportal/data/us_popctr/popctr/
unzip census_centroid_pop.zip -d ../data/census_centroid_pop/
wget -O ../data/census_centroid_pop/metadata.xml http://faculty.baruch.cuny.edu/geoportal/metadata/
rm census_centroid_pop.zip
```

```
library(sp)
library(raster)
library(dplyr)
library(tidyr)
library(ggplot2)

trks <- shapefile("../data/census_centroid_pop/popctr_tracts2010.shp")

trks <- trks[trks@data$POPULATION != 0,]
```

```

trks <- spTransform(trks, CRS("+init=epsg:4326"))

d <- left_join(dl, pl) %>%
  dplyr::select(station, gdd, lat, long, precip)

dsp <- d
coordinates(dsp) <- ~long + lat
proj4string(dsp) <- CRS("+init=epsg:4326")

# find closest gdd

out <- lapply(1:length(trks), function(i) {
  dists <- spDists(trks[i,], dsp, longlat = T)
  dists.min <- apply(dists, 1, function(x) which(x == min(x))[1])
  out <- dsp[dists.min,]
})

dsp.trks <- do.call("rbind", out)

d <- bind_cols(dsp.trks@data, trks@data) %>%
  dplyr::select(LATITUDE, LONGITUDE, POPULATION, gdd, precip, TRACT)

f <- function(x,m = 255) {round(m * (x - min(x, na.rm = T)) / max(x, na.rm = T), 0) }

d <- d %>%
  rename(lat = LATITUDE, long = LONGITUDE, pop = POPULATION) %>%
  filter(complete.cases(.)) %>%
  filter( lat < 50, lat > 25, long < 0) %>%
  mutate(gdd_col = ifelse(gdd > 10000, NA, gdd),
         gdd_col = ifelse(gdd < 2000, NA, gdd_col),
         precip_col = ifelse(precip > 7000, NA, precip),
         precip_col = ifelse(precip < 100, NA, precip_col),
         red = f(gdd_col, m = 80) + 50,
         green = f(precip_col,100) + f(gdd_col,150),
         blue = f(precip_col, m = 180) + 30) %>%
  rowwise() %>%
  mutate(col = ifelse(!is.na(red) & !is.na(blue) & !is.na(green), rgb(red, green, blue,

```

The dots should instead be a contour plot of the US population.

```

library(hdrcde)

pal <- expand.grid(gdd = seq(2000,10000,100), precip = seq(100,7000,100)) %>%
  mutate(red = f(gdd, m = 80) + 50,

```

```

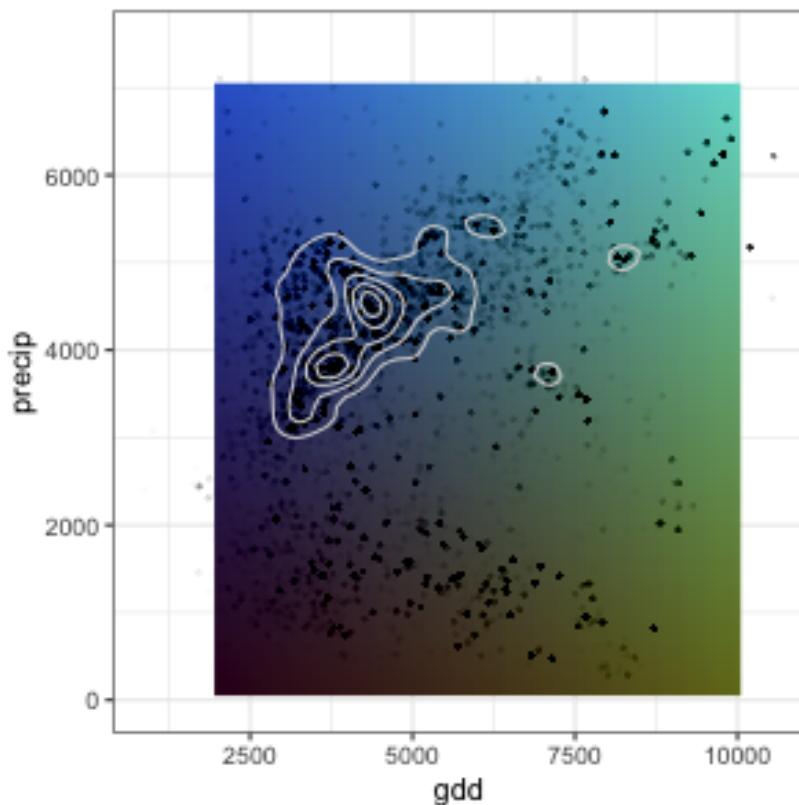
green = f(precip,100) + f(gdd,150),
blue = f(precip, m = 180) + 30,
col = rgb(red, green, blue, maxColorValue = 255))

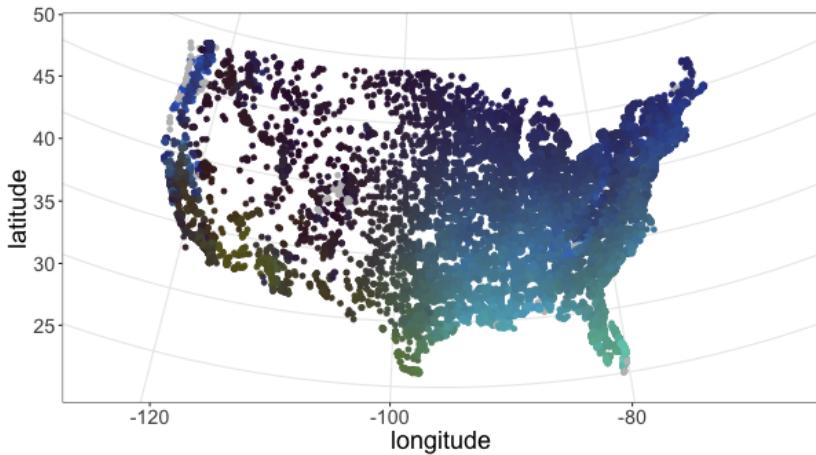
# con <- ggtern:::kde2d.weighted(x = d$gdd, y = d$precip, n = 1000, lims = c(range(d$gdd),
# con <- MASS::kde2d(x = d$gdd, y = d$precip, n = 100, lims = c(range(d$gdd), range(d$precip)))
# con2 <- hdr.2d(x = con$x, y = con$y, den = con, prob = c(50,90))
# con2 <- hdr.2d(x = d$gdd, y = d$precip, prob = c(20,40,60,80))

con3 <- expand.grid(gdd = con2$den$x, precip = con2$den$y)
con3 <- expand.grid(gdd = con$x, precip = con$y)

con3$z <- as.vector(con$z)

```





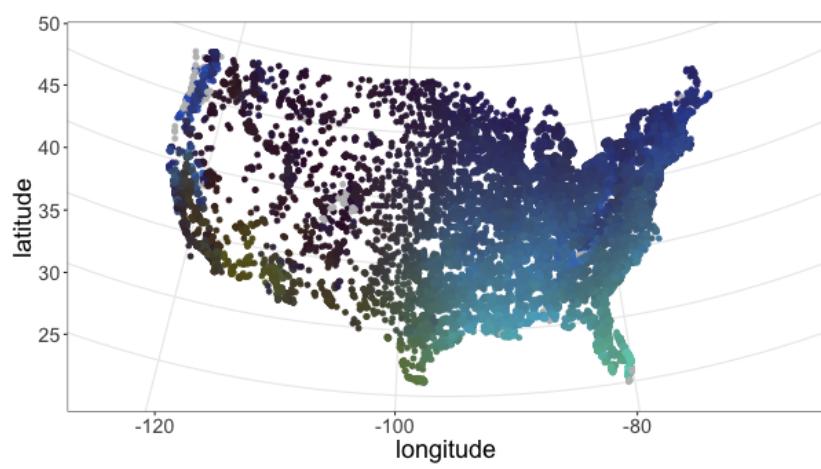
Trying to do log scale

```
d <- bind_cols(dsp.trks@data, trks@data) %>%
  dplyr::select(LATITUDE, LONGITUDE, POPULATION, gdd, precip, TRACT)

f <- function(x,m = 255) {round(m * (x - min(x, na.rm = T)) / max(x, na.rm = T), 0) }

d <- d %>%
  rename(lat = LATITUDE, long = LONGITUDE, pop = POPULATION) %>%
  filter(complete.cases(.)) %>%
  filter( lat < 50, lat > 25, long < 0) %>%
  mutate(gdd_col = log(gdd),
         precip_col = log(precip),
         red = f(gdd_col, m = 80) + 50,
         green = f(precip_col,100) + f(gdd_col,150),
         blue = f(precip_col, m = 180) + 30) %>%
  rowwise() %>%
  mutate(col = ifelse(!is.na(red) & !is.na(blue) & !is.na(green), rgb(red, green, blue,
```

The dots should instead be a contour plot of the US population.



### 6.1.5 Making figures for this data section

1. species by city raster, black if the combination exists

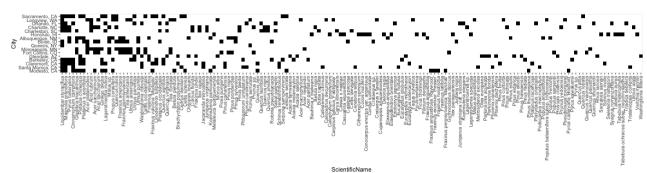
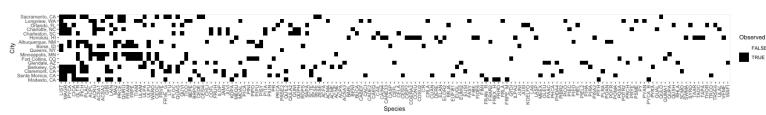
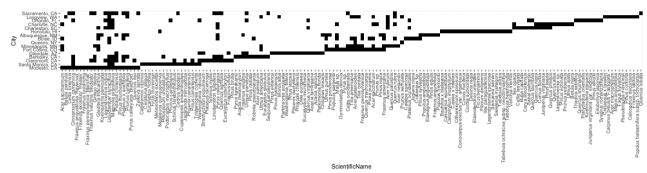
```
library(dplyr)
library(ggplot2)

d <- readRDS("../data/tidy_age_dbh.rds")

cs <- expand.grid(City = unique(d$City), Species = unique(d$Species)) %>%
  mutate(join = paste0(City, Species))

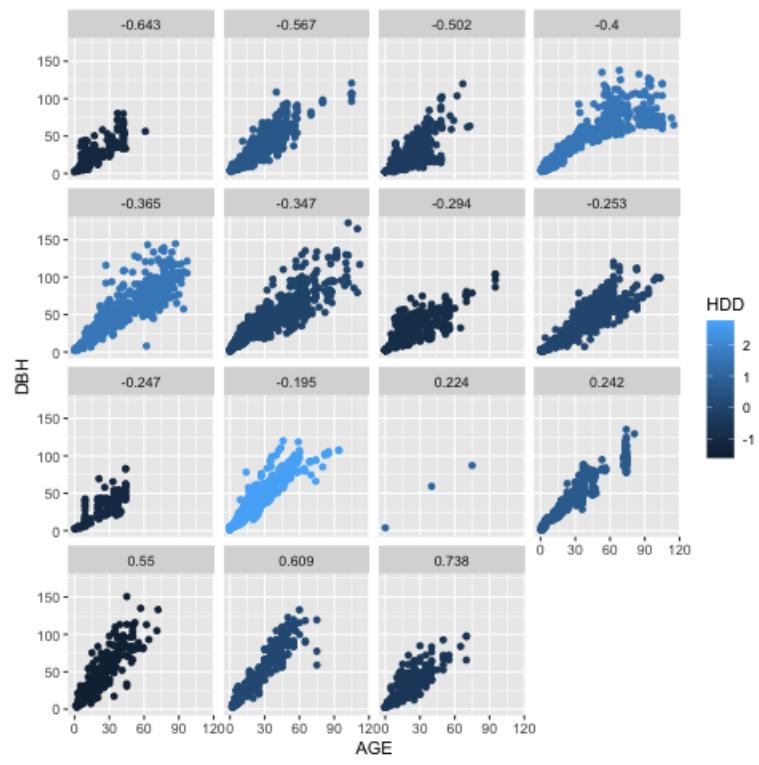
dj <- d %>% mutate(join = paste0(City, Species)) %>% pull(join)

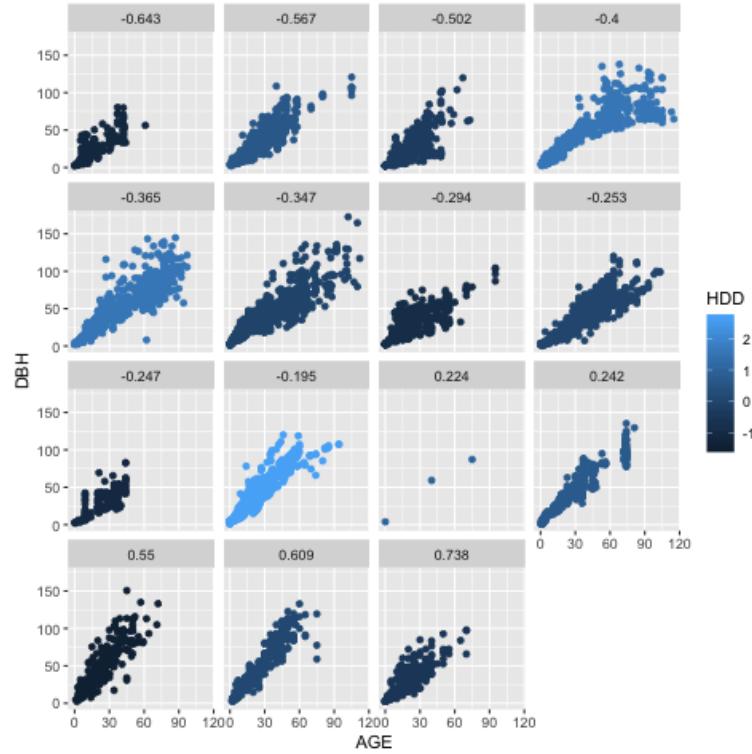
cs <- cs %>% mutate(Observed = join %in% dj)
```



2. look at climate variables

```
d <- readRDS("../data/age_dbh_testing.rds")
```





Is minneapolis an influential point?

## 6.2 Modelling

WRITE OUT THE MODEL..... then convert to latex.

Use `brm(..., sample_prior = "only")`

- Goal is to model tree dimensions as a function of age (and other factors).
  - I'm not doing allometric models (relating dimensions to one another) because
    - \* they are usually log-log, simpler and less interesting
    - \* But doing them would be a natural extension
- from simple to complex
  - single species, single location, homoskedasticity
  - single species, single location, heteroskedasticity
  - multiple species, single location, heteroskedasticity
  - multiple species, vary by climate, heteroskedasticity

- for each of these approaches follow the workflow:
  - problem
  - model
  - fake data
  - fit
  - diagnostics
  - graph fit
  - PPCs (Posterior Predictive Check)
  - Compare models
- criteria for model form
  - Theoretically realistic
    - \* follow what we know about biology
    - \* increase confidence in extrapolation
    - \* assymetrical sigmoidal for growth
  - Interpretable and parsimonious
    - \* few parameters that correspond as much as possible to an interpretable part of growth
  - Can fit in Stan
- think hard about smart priors to help with convergence.
  - I need to address the initial value errors.

$y$  = diameter at breast height (cm)  $x$  = age since transplant (years)

### 6.2.1 Model

$$y_i \sim \text{Gamma}(\mu_i, \alpha_y)$$

$$\mu_i = \beta_{0sc[i]} + \beta_{1sc[i]}(1 - \exp(-\beta_{2sc[i]}x_i^{\beta_{3sc[i]}}))$$

the mean,  $\mu_i$ , is modelled using the weibull curve, a function that has been used for decades in forestry to successfully model tree growth [?] (pg 116).

$\beta_0$  is intercept.  $x$  is transplant age so when  $x = 0$ ,  $y$  can be well above 0.

$\beta_1$  is asymptote of sigmoidal weibull curve. For most species there are no data near the asymptote and this should not be considered the real maximum dbh of a tree.

$\beta_2$  and  $\beta_3$  affect the rate of growth.  $\beta_2$  provides flexibility to have slow or fast growth at young ages (small  $x$ ).

All  $\beta$ 's must be positive and they are likely correlated with one another.

for each beta,  $j = 0, 2, 3$ , species,  $s$ , and each city,  $c$ .

$$\beta_{j sc[i]} = \beta_j + \gamma_{js[i]} + \delta_{jc[i]}$$

for beta<sub>1</sub>

$$\beta_{1sc[i]} = \beta_1 + \tau_1 * Precip_i + \tau_2 * GDD_i + \tau_3 * (Precip_i * GDD_i) + \gamma_{1s[i]} + \delta_{1c[i]}$$

where  $\beta_j$  is the mean for  $\beta$  coefficient  $\beta_j$ .  $\gamma_{js}$  is the contribution of genetic (species) effect for species s on  $\beta_j$ .  $\delta_{jc}$  is the city effect for city c on  $\beta_j$ .

for each i. Species random effect:

$$\gamma_{js[i]} \sim N(\gamma_{jg[i]}, \sigma_{js[i]})$$

for genus, g.

species nested within Genus.

$$\gamma_{jg[i]} \sim N(0, \sigma_{jg})$$

for each j. City random effect:

$$\delta_{jc} \sim N(0, \sigma_{\delta j})$$

for gamma  $\alpha = \mu^2/\sigma^2$   $\beta = \mu/\sigma^2$

parameter	mu	sd	alpha (of gamma)	beta (of gamma)
$\beta_0$	3	1.5	4.	1.3333333
$\beta_1$	1.5	.3	25.	16.666667
$\beta_2$	1.25	.15	69.444444	55.555556
$\beta_3$	1	.15	44.444444	44.444444

$$\beta_0 \sim Gamma(4, 1.33)$$

$$\beta_1 \sim Gamma(25, 16.7)$$

$$\beta_2 \sim Gamma(69.4, 55.5)$$

$$\beta_3 \sim Gamma(44.4, 44.4)$$

these priors selected because they create a wide range of possible mean curves, but they are physically possible.

variability by cities and species, something wide

$$\sigma_{\delta 0} \sim half-Cauchy()$$

$$\sigma_{\delta 1} \sim half-Cauchy()$$

$$\sigma_{\delta 2} \sim half-Cauchy()$$

$$\sigma_{\delta 3} \sim \text{half-Cauchy}()$$

$$\sigma_{\gamma 0} \sim \text{half-Cauchy}()$$

$$\sigma_{\gamma 1} \sim \text{half-Cauchy}()$$

$$\sigma_{\gamma 2} \sim \text{half-Cauchy}()$$

$$\sigma_{\gamma 3} \sim \text{half-Cauchy}()$$

variability by species for intercept,  $\beta_0$

$$\sigma_{\delta 0} \sim$$

### 6.2.2 fitting model

### 6.2.3 model form: Weibull

model form: [?]: "This reality should lead to model forms that are complex enough to accurately and adequately characterize the expected major behaviors of the population, but simple enough to avoid being overly influenced by the sample's peculiarities."

1. why weibull it was good for height:

Why Weibull is a decent option: pg116 [?]

Significant differences between model forms can exist, however, when limited data are available. For example, Temesgen and von Gadow (2004) found that the percentage difference in root mean square error between the best and worst equation for five commonly used model forms varied from 5 to 33%. Huang et al. (1992) found in their analysis that the Chapman–Richards, Weibull, and a modified logistic-type function were consistently among the best performing models because they were flexible, able to assume a variety of shapes, and extrapolated well.

It should also be good for dbh. height and dbh are different though:

A diameter growth curve would show much the same trend, except there is a tendency toward more sustained growth rate as the tree matures (Hann and Hanus, 2002b). While height increment may nearly cease in maturity, diameter increment must continue in order to produce the xylem and phloem needed for tree survival

## 2. look at weibull

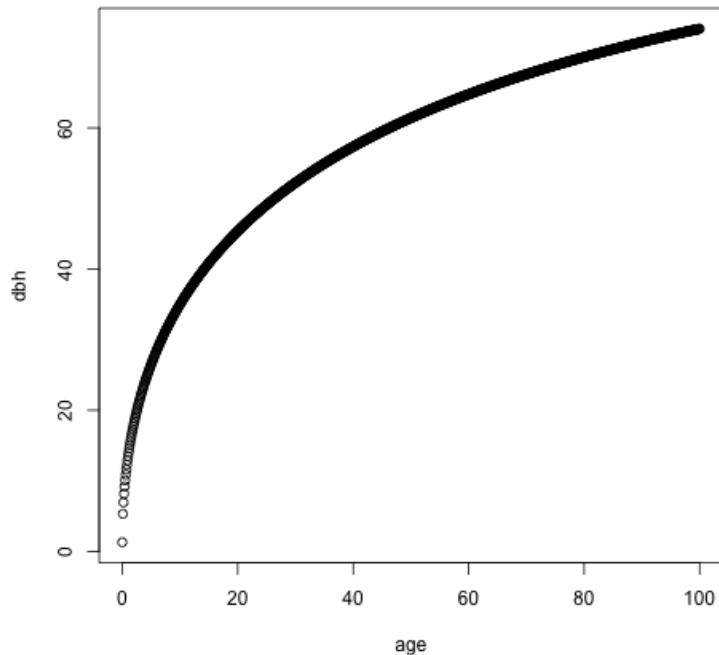
$$\beta_0(1 - \exp(\beta_1 * DBH^{\beta_2}))$$

from huang 1992

$$H = 1.3 + a * (1 - \exp(-b * DBH^c))$$

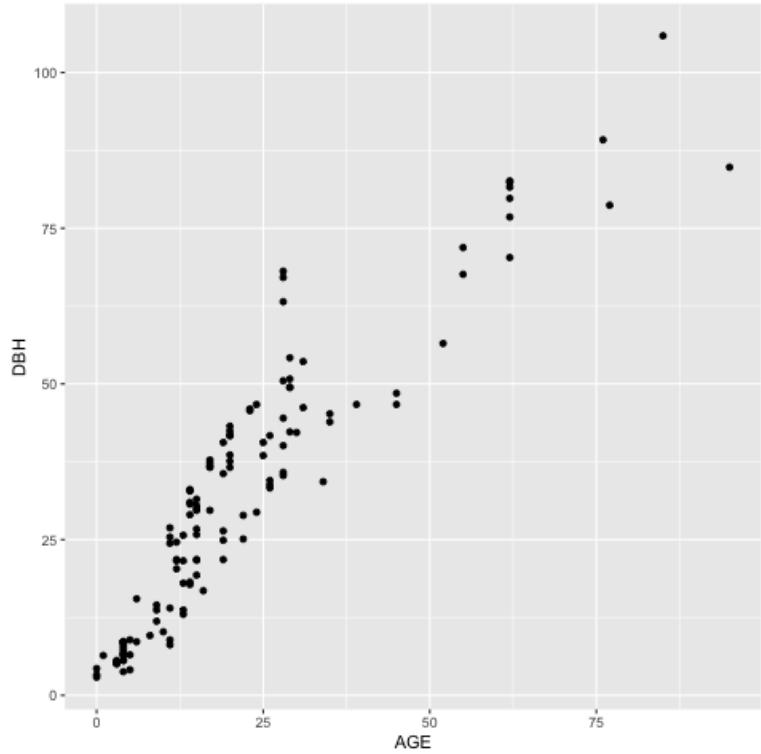
from wikipedia: cdf:

$$1 - e^{-(x/\lambda)^k}$$



play with the parameters to understand the flexibility and control.

```
d <- readRDS("../data/tidy_age_dbh.rds")
```



```
# m <- nls(DBH ~ a * (1 - exp(-b * AGE^c)), fr
m <- nls(DBH ~ a * (1 - exp(-b * AGE^c)), fram_ftcollins, list(a = 100, b = .1, c = .5)
```

```
summary(m)
```

Formula: DBH ~ a \* (1 - exp(-b \* AGE^c))

Parameters:

	Estimate	Std. Error	t value	Pr(> t )
a	1.298e+02	4.928e+01	2.633	0.013606 *
b	9.289e-03	2.137e-03	4.346	0.000165 ***
c	1.112e+00	1.745e-01	6.373	6.75e-07 ***

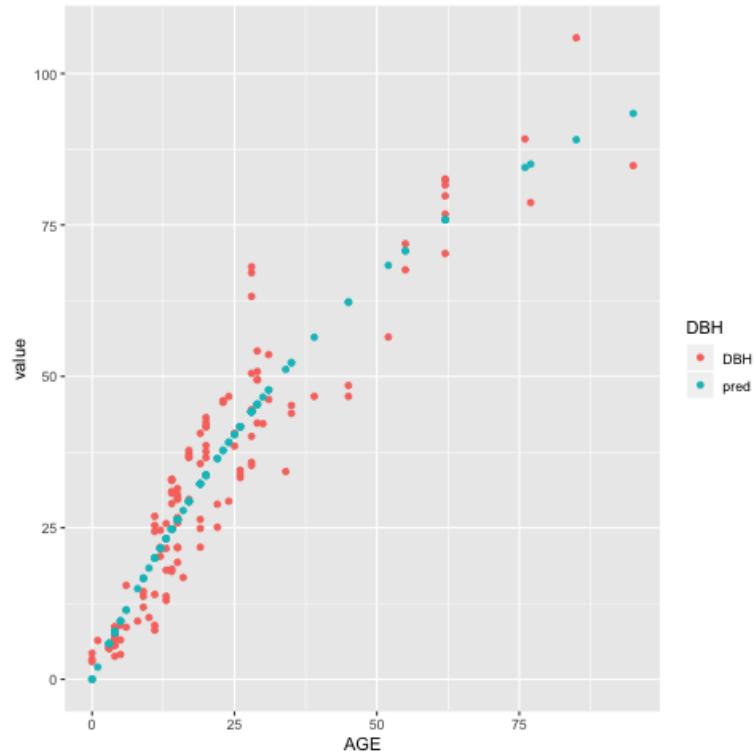
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.467 on 28 degrees of freedom

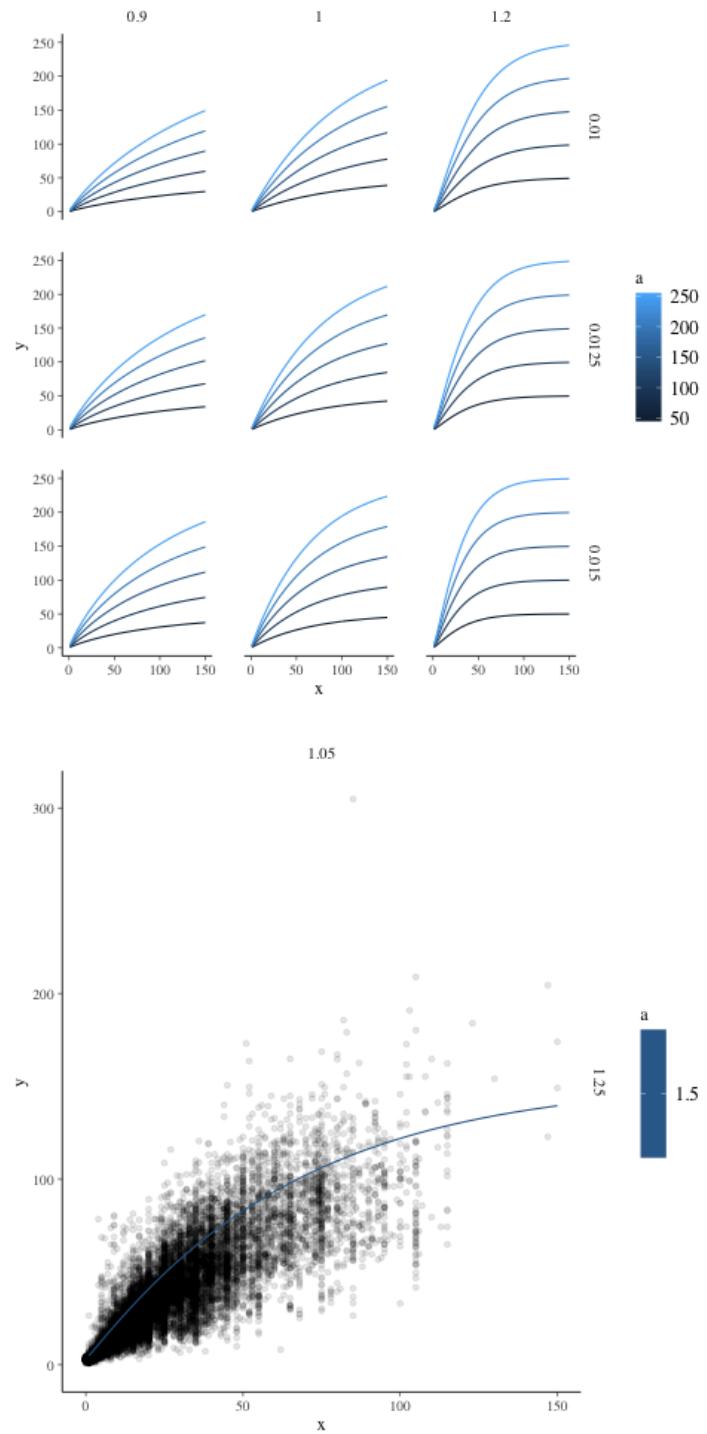
Number of iterations to convergence: 18

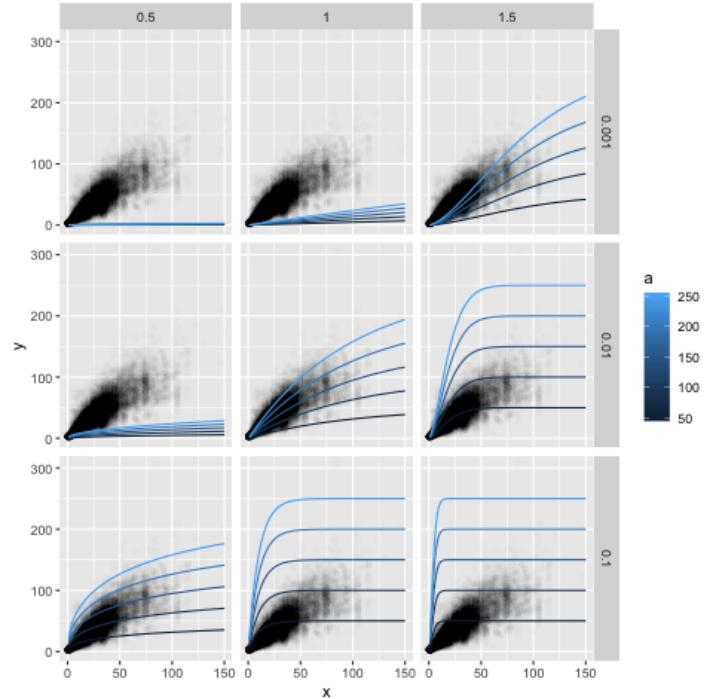
Achieved convergence tolerance: 1.869e-06



"a" can't be negative because dbh must be positive "b" can't be negative because it causes negative growth and negative dbh values, impossible "c" can't be negative because it causes negative growth, impossible

- (a) what is the range of reasonable parameter values, can be helpful for priors

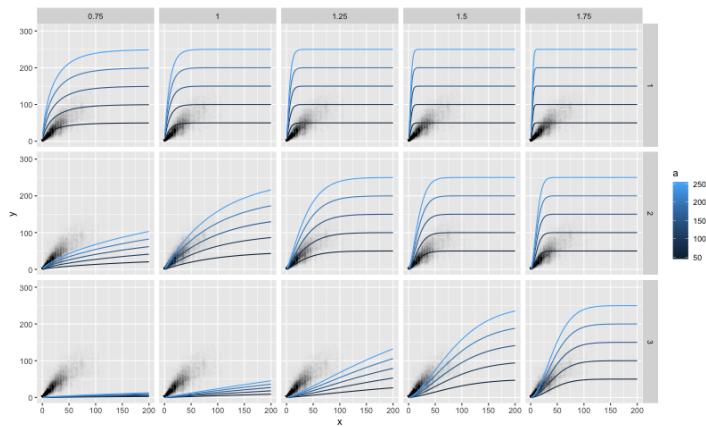


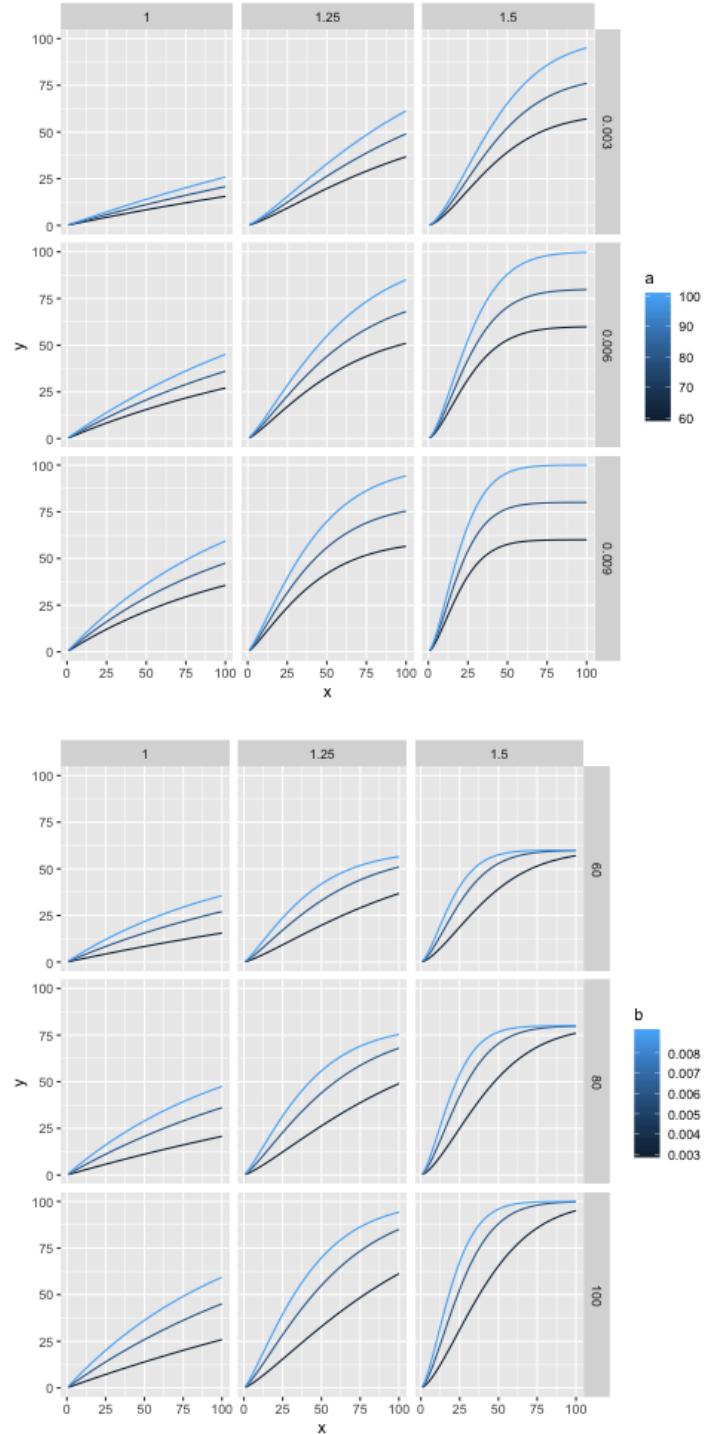


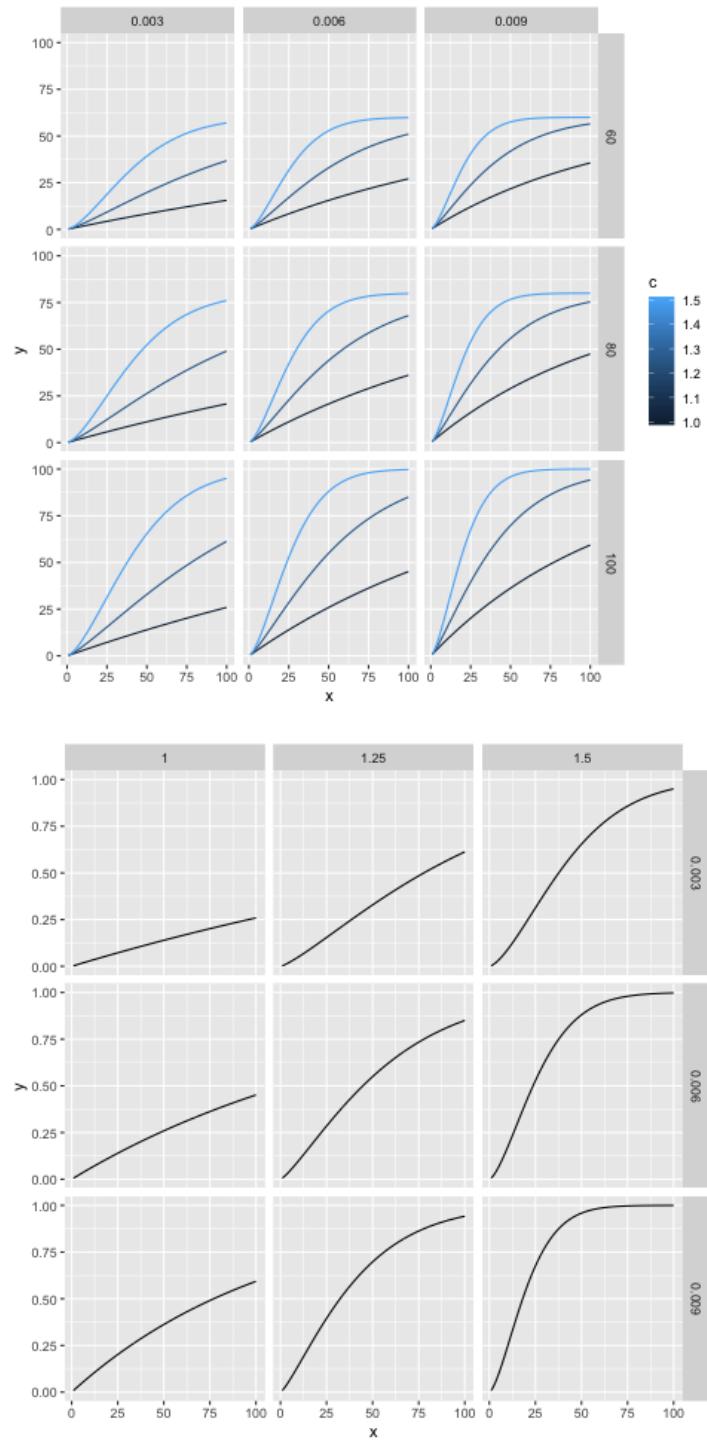
range of realistic values: b1 : 50 - 250 b2 : 0.01 - 0.1 b3 : .5 - 1.5

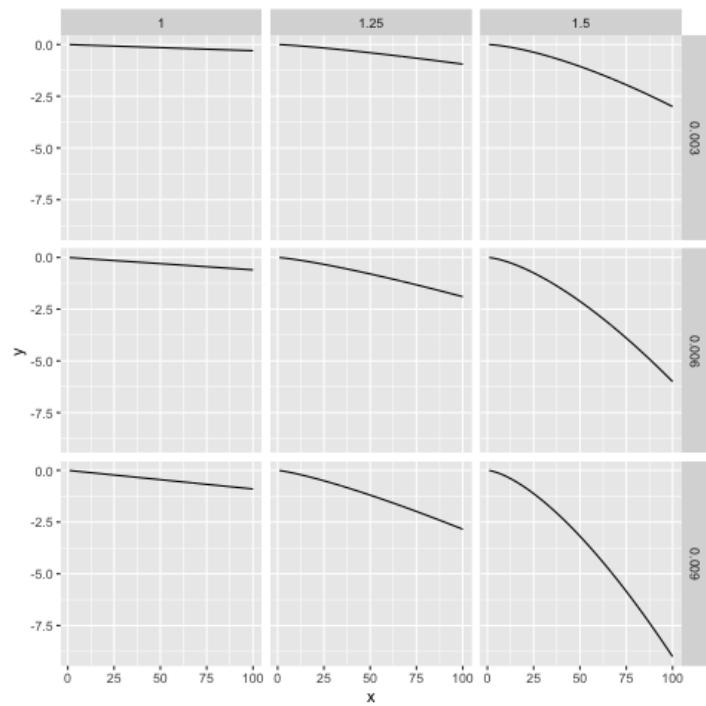
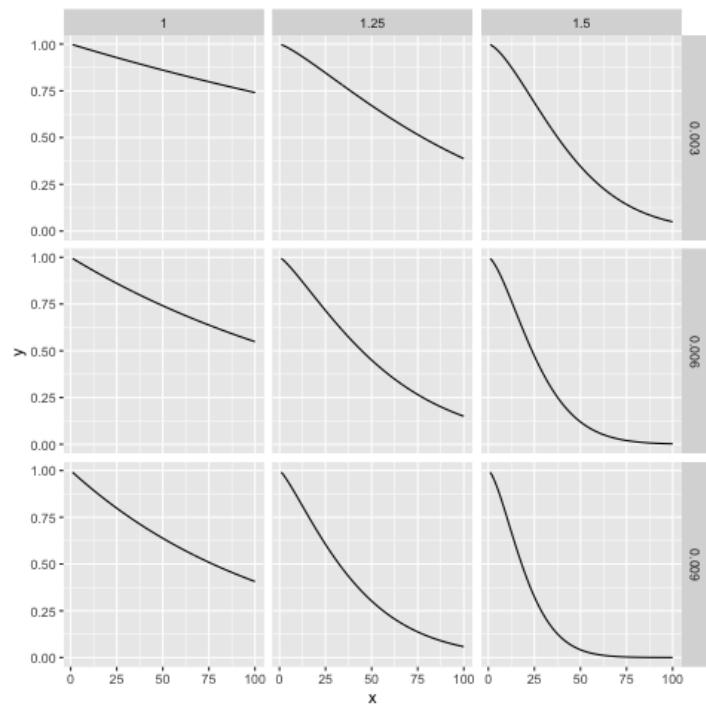
b

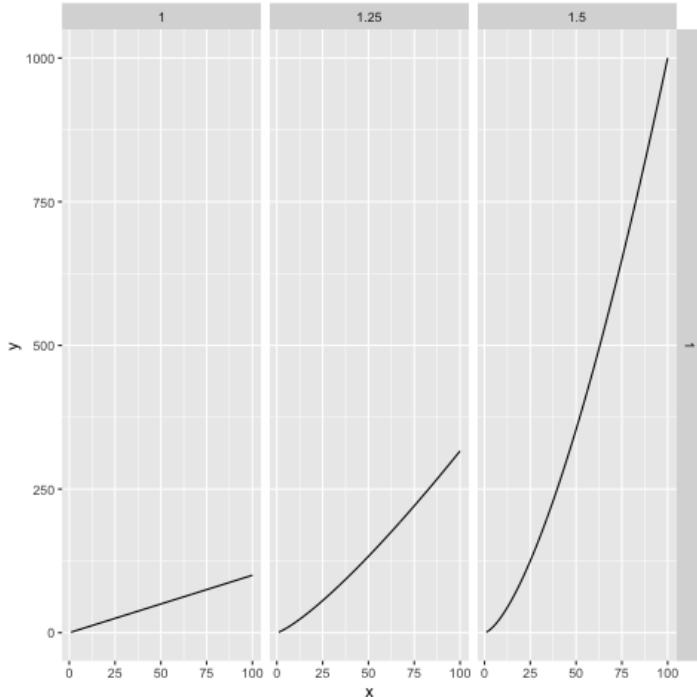
the higher values of b2 only make sense with lower values of b3. they are correlated. I wonder if I could include that information in the prior?











#### 6.2.4 a look at gamma family

```

library(brms)
source("allo_functions.R")

genus <- "none"
species <- "single"
cities <- "single"
climate <- "none"
hetero <- "no"
family <- "Gamma"

form <- generate_formula(genus, species, cities, climate, hetero, family)
nlprior <- generate_prior(genus, species, cities, climate, family)

n <- 100
  b1 <- 1.2
  b2 <- 2
  b3 <- 1
  shape <- 40

```

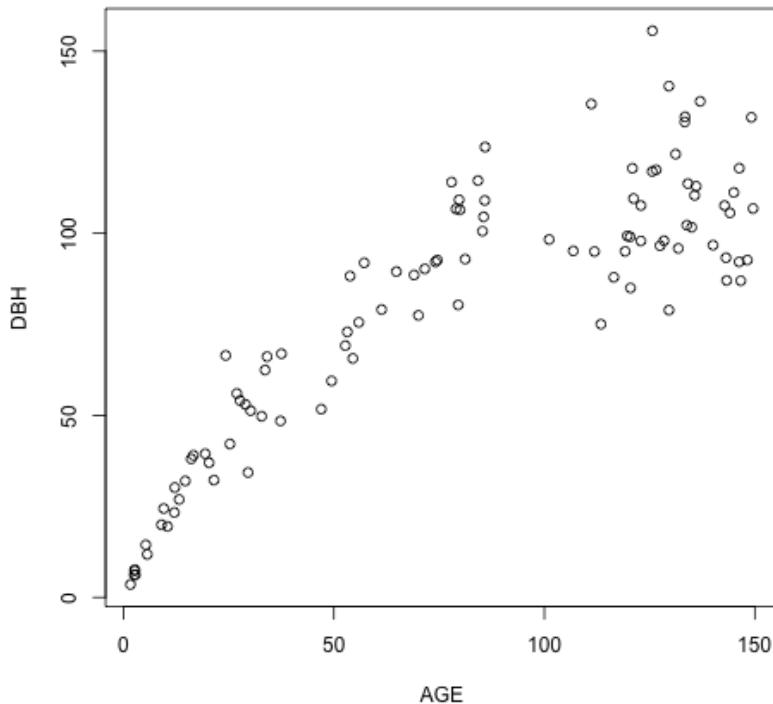
```

w <- function(a,b,c,x) 100 * a * (1 - exp(-(b/100) * x^c))
# w <- function(a,b,c,x) a * (1 - exp(-b *
AGE <- runif(n , 0, 150)

mu <- w(b1,b2,b3, AGE)

DBH <- rgamma(n, rate = shape / mu, shape = shape)
df <- data.frame(AGE, DBH)

```



```

mod <- brm(form, chains = 2, cores = 2, data = df, init_r = .3, prior = nlprior, iter = 1000)

Compiling the C++ model
Start sampling

SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 1).

```

```
Gradient evaluation took 0.000672 seconds
1000 transitions using 10 leapfrog steps per transition would take 6.72 seconds.
Adjust your expectations accordingly!
```

```
Iteration: 1 / 1000 [ 0%] (Warmup)
```

```
SAMPLING FOR MODEL 'gamma brms-model' NOW (CHAIN 2).
```

```
Gradient evaluation took 0.00056 seconds
1000 transitions using 10 leapfrog steps per transition would take 5.6 seconds.
Adjust your expectations accordingly!
```

```
Iteration: 1 / 1000 [ 0%] (Warmup)
Iteration: 100 / 1000 [ 10%] (Warmup)
Iteration: 100 / 1000 [ 10%] (Warmup)
Iteration: 200 / 1000 [ 20%] (Warmup)
Iteration: 200 / 1000 [ 20%] (Warmup)
Iteration: 300 / 1000 [ 30%] (Warmup)
Iteration: 300 / 1000 [ 30%] (Warmup)
Iteration: 400 / 1000 [ 40%] (Warmup)
Iteration: 400 / 1000 [ 40%] (Warmup)
Iteration: 500 / 1000 [ 50%] (Warmup)
Iteration: 501 / 1000 [ 50%] (Sampling)
Iteration: 500 / 1000 [ 50%] (Warmup)
Iteration: 501 / 1000 [ 50%] (Sampling)
Iteration: 600 / 1000 [ 60%] (Sampling)
Iteration: 700 / 1000 [ 70%] (Sampling)
Iteration: 600 / 1000 [ 60%] (Sampling)
Iteration: 800 / 1000 [ 80%] (Sampling)
Iteration: 700 / 1000 [ 70%] (Sampling)
Iteration: 900 / 1000 [ 90%] (Sampling)
Iteration: 800 / 1000 [ 80%] (Sampling)
Iteration: 1000 / 1000 [100%] (Sampling)
```

```
Elapsed Time: 5.63711 seconds (Warm-up)
              4.44767 seconds (Sampling)
              10.0848 seconds (Total)
```

```
Iteration: 900 / 1000 [ 90%] (Sampling)
Iteration: 1000 / 1000 [100%] (Sampling)
```

```
Elapsed Time: 5.94742 seconds (Warm-up)
              4.96327 seconds (Sampling)
```

```

10.9107 seconds (Total)

mod

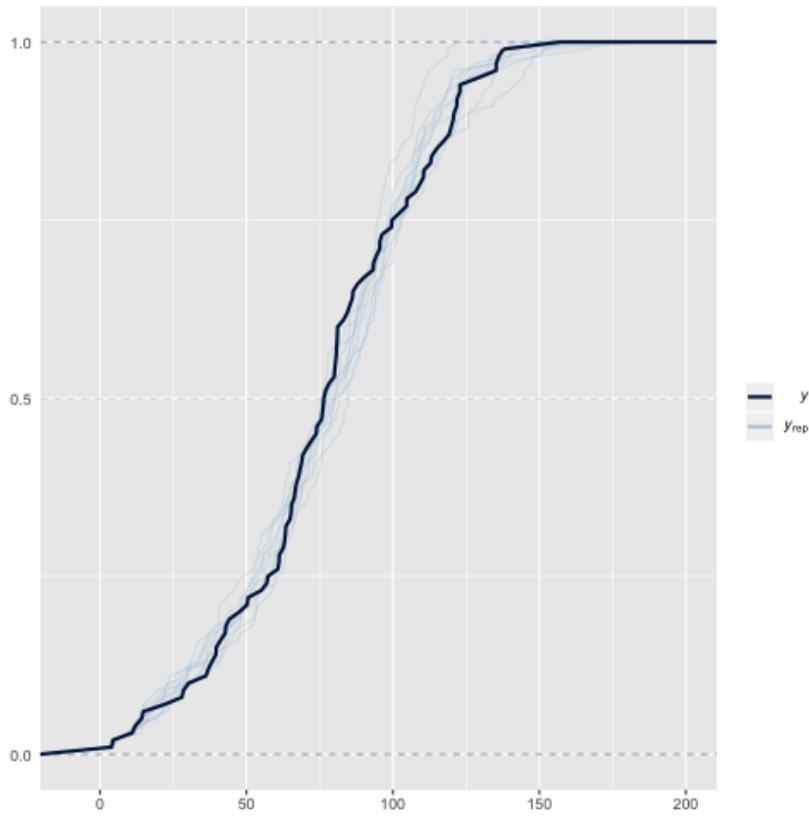
Family: gamma
Links: mu = identity; shape = identity
Formula: DBH ~ 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3)))
         b1 ~ 1
         b2 ~ 1
         b3 ~ 1
Data: df (Number of observations: 100)
Samples: 2 chains, each with iter = 1000; warmup = 500; thin = 1;
         total post-warmup samples = 1000

Population-Level Effects:
Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
b1_Intercept    1.17      0.09     1.04     1.38      397 1.00
b2_Intercept    2.19      0.25     1.74     2.76      453 1.00
b3_Intercept    0.97      0.05     0.87     1.06      442 1.00

Family Specific Parameters:
Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
shape     24.82      3.40    18.66    31.75      458 1.00

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
is a crude measure of effective sample size, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).

```



#### 6.2.5 TODO genus: none; species: single; cities: single; climate: none; hetero: no; family gamma

1. model R code

```

library(dplyr)
library(brms)
source("allo_functions.R")

genus <- "none"
species <- "single"
cities <- "single"
climate <- "none"
hetero <- "no"
family <- "Gamma"

data_form <- formula(DBH ~ b0 + 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3))))
b0_form <- formula(b0 ~ 1)

```

```

b1_form <- formula(b1 ~ 1)
b2_form <- formula(b2 ~ 1)
b3_form <- formula(b3 ~ 1)

form <- bf(data_form, b0_form, b1_form, b2_form, b3_form, nl = T)

nlprior <- c(prior(gamma(4, 1.33), nlpars = "b0", lb = 0),
              prior(gamma(25, 16.66), nlpars = "b1", lb = 0),
              prior(gamma(69.44, 55.556), nlpars = "b2", lb = 0),
              prior(gamma(44.444, 44.444), nlpars = "b3", lb = 0),
              prior(gamma(20, 1), class = "shape"))

d <- readRDS("../data/age_dbh_testing.rds")

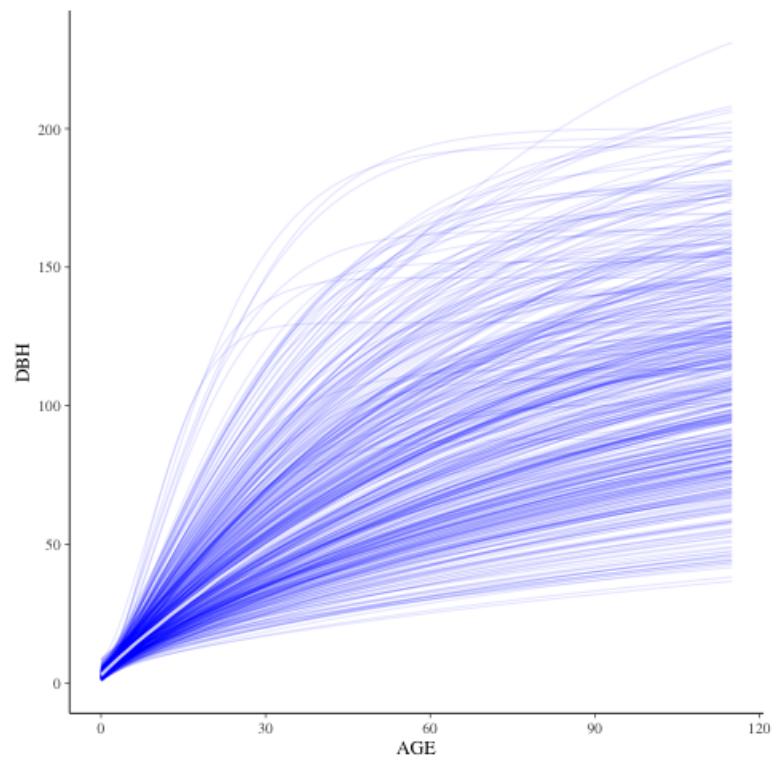
prior_mod <- brm(form,
                  data = d,
                  prior = nlprior,
                  family = Gamma("identity"),
                  sample_prior = "only",
                  chains = 2, cores = 2, init_r = .3, iter = 500)

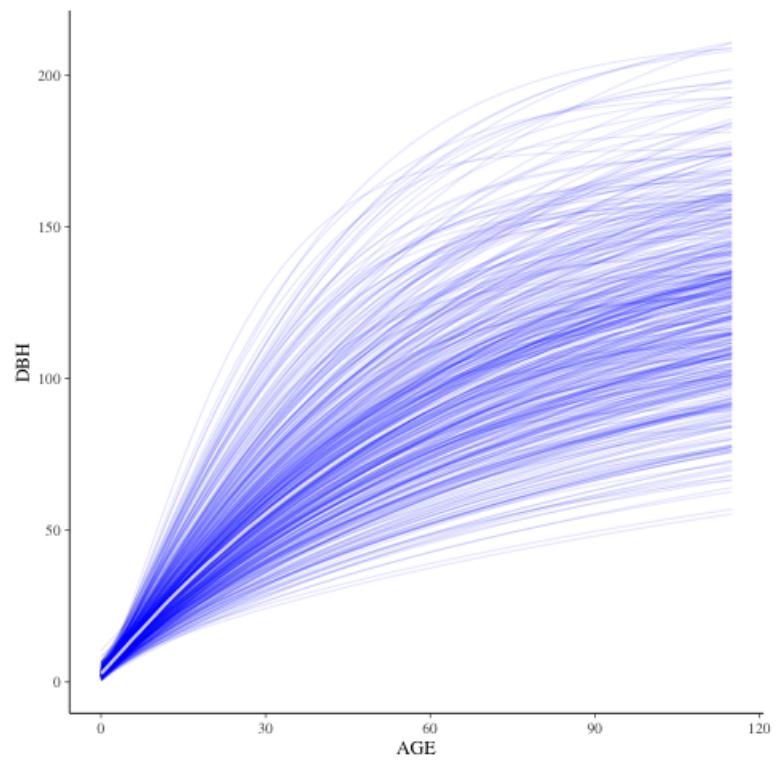
pred <- predict(prior_mod, newdata = d)

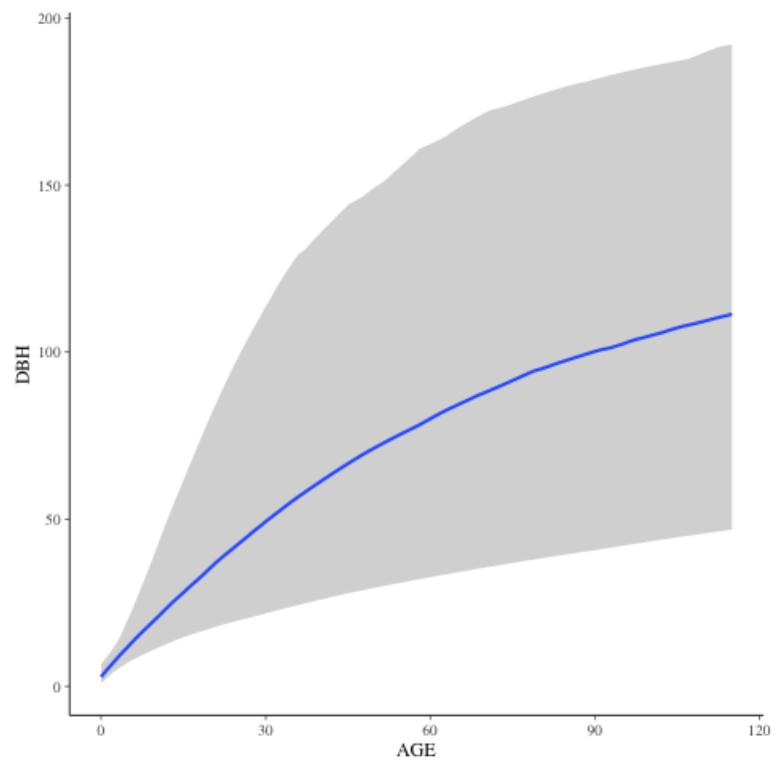
mod <- brm(form,
            data = d,
            prior = nlprior,
            family = Gamma("identity"),
            chains = 8, cores = 8, init_r = .3, iter = 2000)

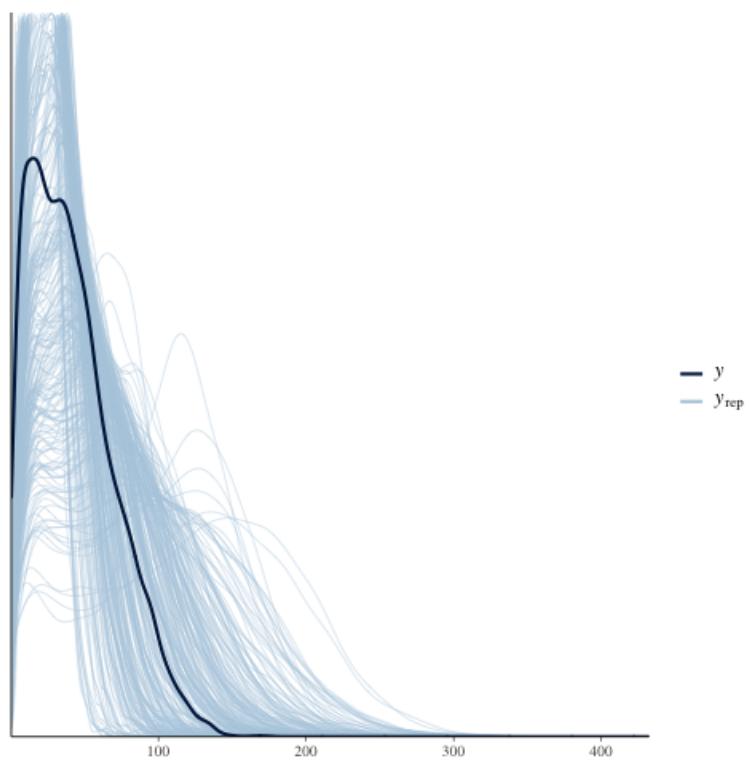
saveRDS(mod, paste0("../models/genus_", genus, "_species_", species, "_cities_", cities,

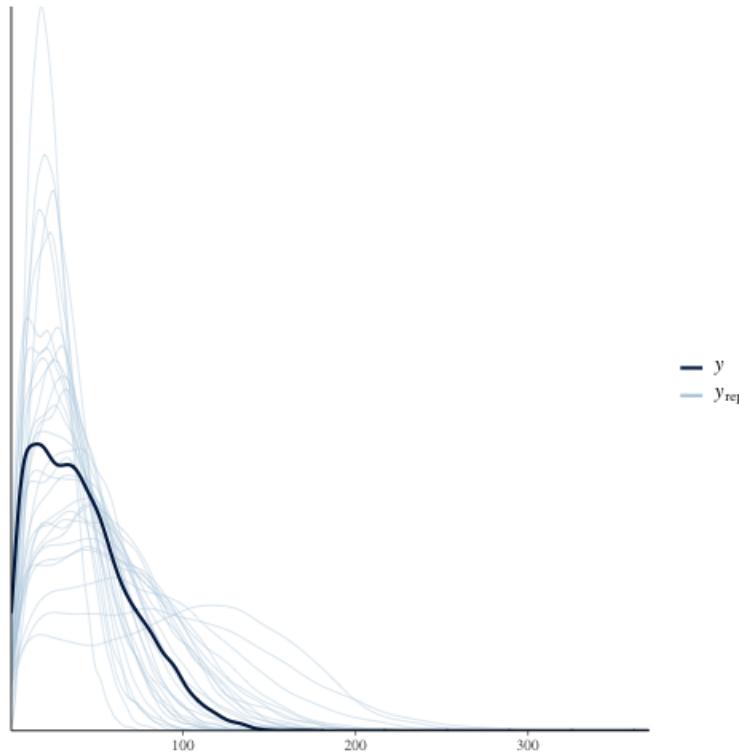
```











2. tangle C-c C-v t

3. send to krusty

```
rsync -avz genus_none_species_single_cities_single_climate_none_hetero_no_family_gamma.
```

4. run on krusty

run from krusty terminal

```
ssh krusty
```

```
cd allo/code
```

```
nohup R CMD BATCH genus_none_species_single_cities_single_climate_none_hetero_no_family
exit
```

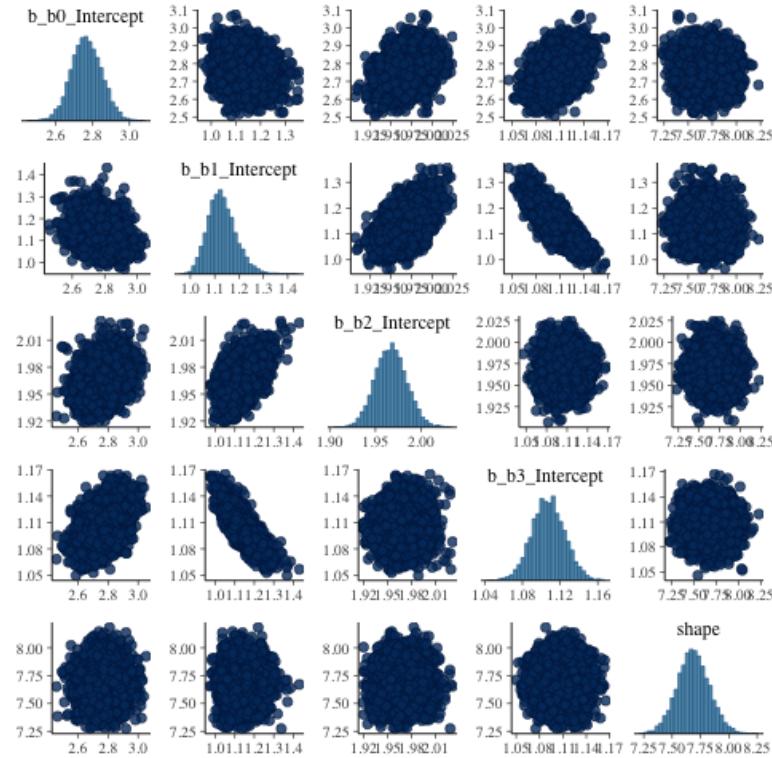
```
cat genus_none_species_single_cities_single_climate_none_hetero_no_family_gamma.Rout
```

5. pull back from krusty

```
rsync -avz erker@krusty:~/allo/models/genus_none_species_single_cities_single_climate_n
```

6. diagnostics

```
mod_genus_none_species_single_cities_single_climate_none_hetero_no_family_Gamma <- read
mod <- mod_genus_none_species_single_cities_single_climate_none_hetero_no_family_Gamma
```



```
summary(mod)
```

```
Family: gamma
Links: mu = identity; shape = identity
Formula: DBH ~ b0 + 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3)))
        b0 ~ 1
        b1 ~ 1
        b2 ~ 1
        b3 ~ 1
Data: d (Number of observations: 5548)
Samples: 6 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 6000
```

#### Population-Level Effects:

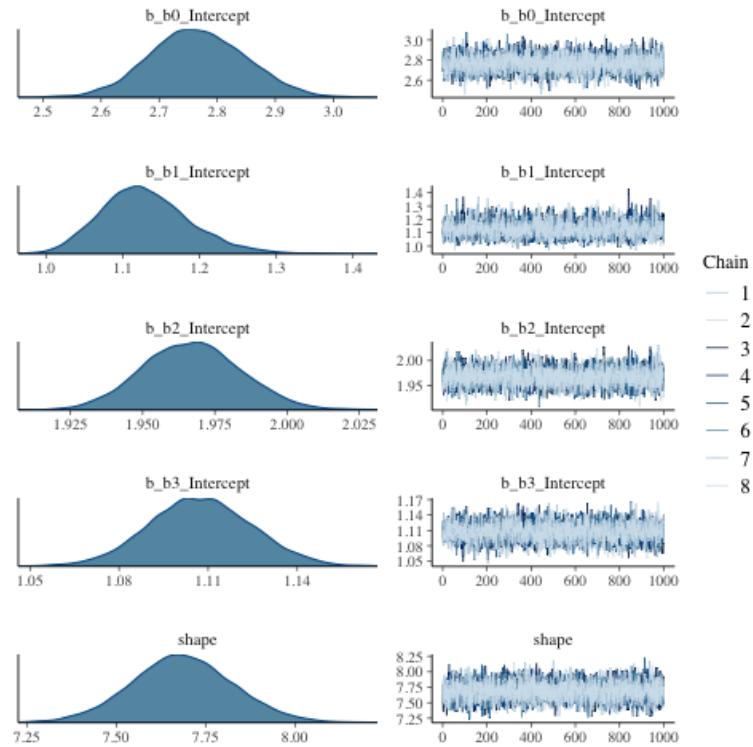
	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
b0_Intercept	2.77	0.08	2.61	2.93	3416	1.00
b1_Intercept	1.12	0.05	1.03	1.24	2694	1.00

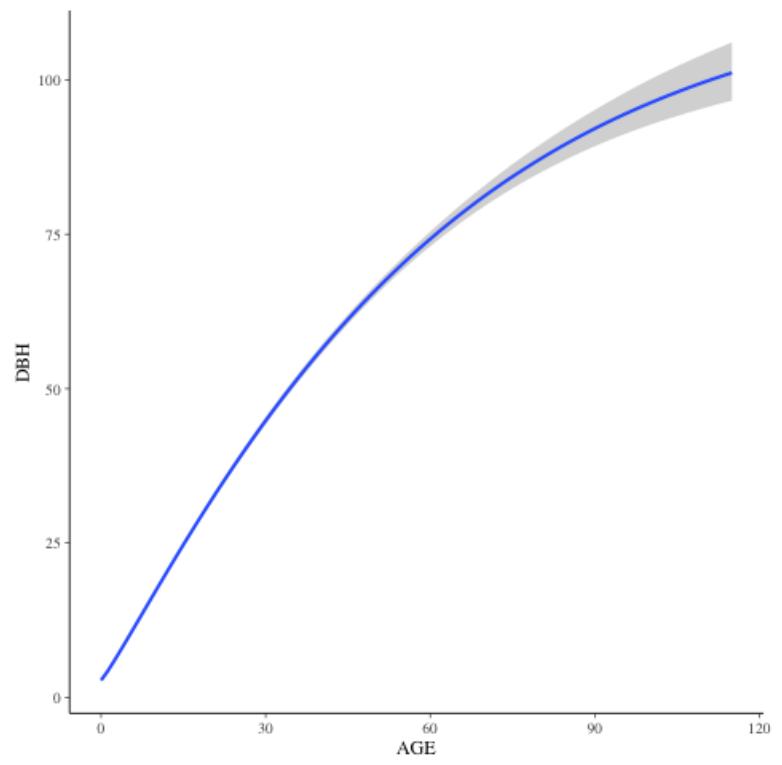
b2_Intercept	1.08	0.04	1.00	1.17	3530	1.00
b3_Intercept	1.11	0.02	1.08	1.14	2806	1.00

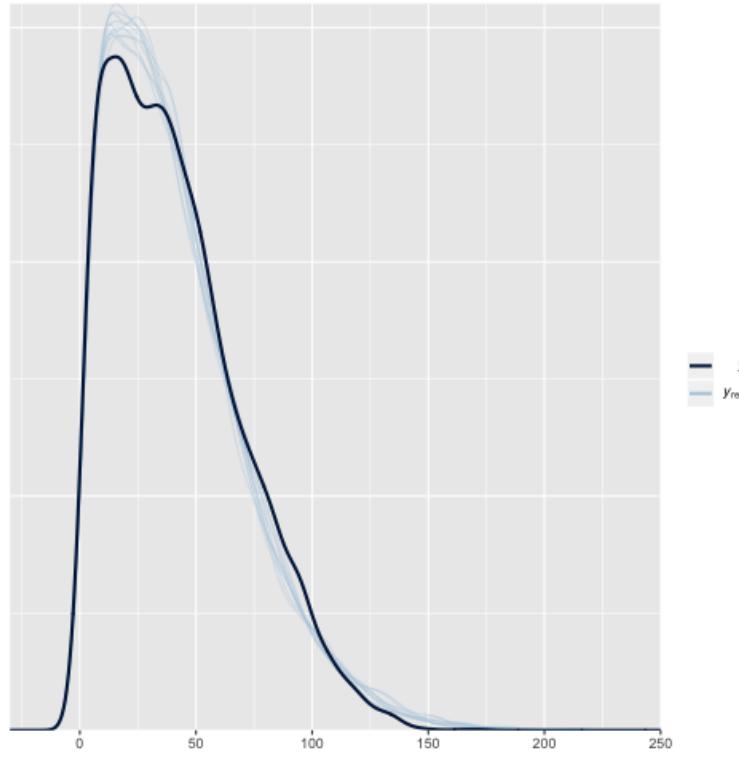
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
shape	7.69	0.14	7.42	7.96	3579	1.00

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).







#### 6.2.6 TODO genus: many; species: many; cities: many; climate: b1linint; hetero: no; family: Gamma

1. model R code

```

library(dplyr)
library(brms)
source("allo_functions.R")

genus <- "many"
species <- "many"
cities <- "many"
climate <- "b1linint"
hetero <- "no"
family <- "Gamma"

data_form <- formula(DBH ~ b0 + 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3))))
b0_form <- formula(b0 ~ (1 | City) + (1 | Genus/Species))
b1_form <- formula(b1 ~ Precip * GDD + (1 | City) + (1 | Genus / Species))
b2_form <- formula(b2 ~ (1 | City) + (1 | Genus/Species))

```

```

b3_form <- formula(b3 ~ (1 | City) + (1 | Genus/Species))

form <- bf(data_form, b0_form, b1_form, b2_form, b3_form, nl = T)

nlprior <- c(prior(gamma(7.5, 3), nlpars = "b0", lb = 0),
              prior(gamma(8, 6), nlpars = "b1", lb = 0),
              prior(gamma(8, 8), nlpars = "b2", lb = 0),
              prior(gamma(8, 6), nlpars = "b3", lb = 0),
              prior(gamma(5, .3), class = "shape"),
              prior(gamma(1.3, 1.3), class = "sd", nlpars = "b0"),
              prior(gamma(10, 10), class = "sd", nlpars = "b1"),
              prior(gamma(1.3, 1.3), class = "sd", nlpars = "b2"),
              prior(gamma(1.3, 1.3), class = "sd", nlpars = "b3"))

d <- readRDS("../data/age_dbh_testing.rds")

prior_mod <- brm(form,
                  data = d,
                  prior = nlprior,
                  family = Gamma("identity"),
                  sample_prior = "only",
                  chains = 8, cores = 8, init_r = .3, iter = 2000)

mod <- brm(form,
            data = d,
            prior = nlprior,
            family = Gamma("identity"),
            chains = 8, cores = 8, init_r = .3, iter = 2000)

saveRDS(mod, paste0("../models/genus_", genus, "_species_", species, "_cities_", cities,

```

2. tangle
3. send to krusty

```
rsync -avz genus_many_species_many_cities_many_climate_bilinint_hetero_no_family_Gamma.
```

4. run on krusty  
run from krusty terminal

```
ssh krusty
cd allo/code
nohup R CMD BATCH genus_many_species_many_cities_many_climate_bilinint_hetero_no_family
exit

cat genus_many_species_many_cities_many_climate_bilinint_hetero_no_family_Gamma.Rout
```

5. pull back from krusty

```
rsync -avz erker@krusty:~/allo/models/genus_many_species_many_cities_many_climate_bilinint_hetero_no_family_Gamma.Rout .
```

6. assess model

```
mod_genus_many_species_many_cities_many_climate_bilinint_hetero_no_family_Gamma <- readRDS(mod_genus_many_species_many_cities_many_climate_bilinint_hetero_no_family_Gamma)
```



```
../figs/pairs_genus_many_species_many_cities_many_climate_bilinint_hetero_no_family
```

```
summary(mod)
```

```
Family: gamma
Links: mu = identity; shape = identity
```

Formula: DBH ~ b0 + 100 \* b1 \* (1 - exp(-(b2/100) \* AGE^(b3)))  
 b0 ~ (1 | City) + (1 | Genus/Species)  
 b1 ~ Precip \* GDD + (1 | City) + (1 | Genus/Species)  
 b2 ~ (1 | City) + (1 | Genus/Species)  
 b3 ~ (1 | City) + (1 | Genus/Species)  
 Data: d (Number of observations: 5548)  
 Samples: 8 chains, each with iter = 2000; warmup = 1000; thin = 1;  
 total post-warmup samples = 8000

**Group-Level Effects:**  
 ~City (Number of levels: 15)

	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	1.41	0.31	0.90	2.12	231	1.03
sd(b1_Intercept)	0.85	0.27	0.43	1.48	1625	1.01
sd(b2_Intercept)	0.34	0.08	0.22	0.53	3866	1.00
sd(b3_Intercept)	0.19	0.04	0.12	0.29	3661	1.00

~Genus (Number of levels: 18)

	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	0.42	0.20	0.05	0.83	1916	1.00
sd(b1_Intercept)	1.16	0.29	0.63	1.78	1825	1.01
sd(b2_Intercept)	0.13	0.07	0.01	0.29	1529	1.01
sd(b3_Intercept)	0.11	0.05	0.01	0.21	1268	1.01

~Genus:Species (Number of levels: 25)

	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	0.27	0.16	0.02	0.60	1542	1.01
sd(b1_Intercept)	0.64	0.21	0.31	1.14	3183	1.00
sd(b2_Intercept)	0.14	0.05	0.06	0.24	2888	1.00
sd(b3_Intercept)	0.11	0.03	0.05	0.18	2355	1.01

**Population-Level Effects:**

	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
b0_Intercept	2.31	0.37	1.57	3.04	3547	1.00
b1_Intercept	0.68	0.23	0.31	1.19	8000	1.00
b1_Precip	0.67	0.23	0.29	1.19	8000	1.00
b1_GDD	0.22	0.06	0.11	0.35	8000	1.00
b1_Precip:GDD	0.19	0.06	0.09	0.32	8000	1.02
b2_Intercept	0.80	0.11	0.60	1.02	2819	1.00
b3_Intercept	1.05	0.06	0.93	1.18	3680	1.00

**Family Specific Parameters:**

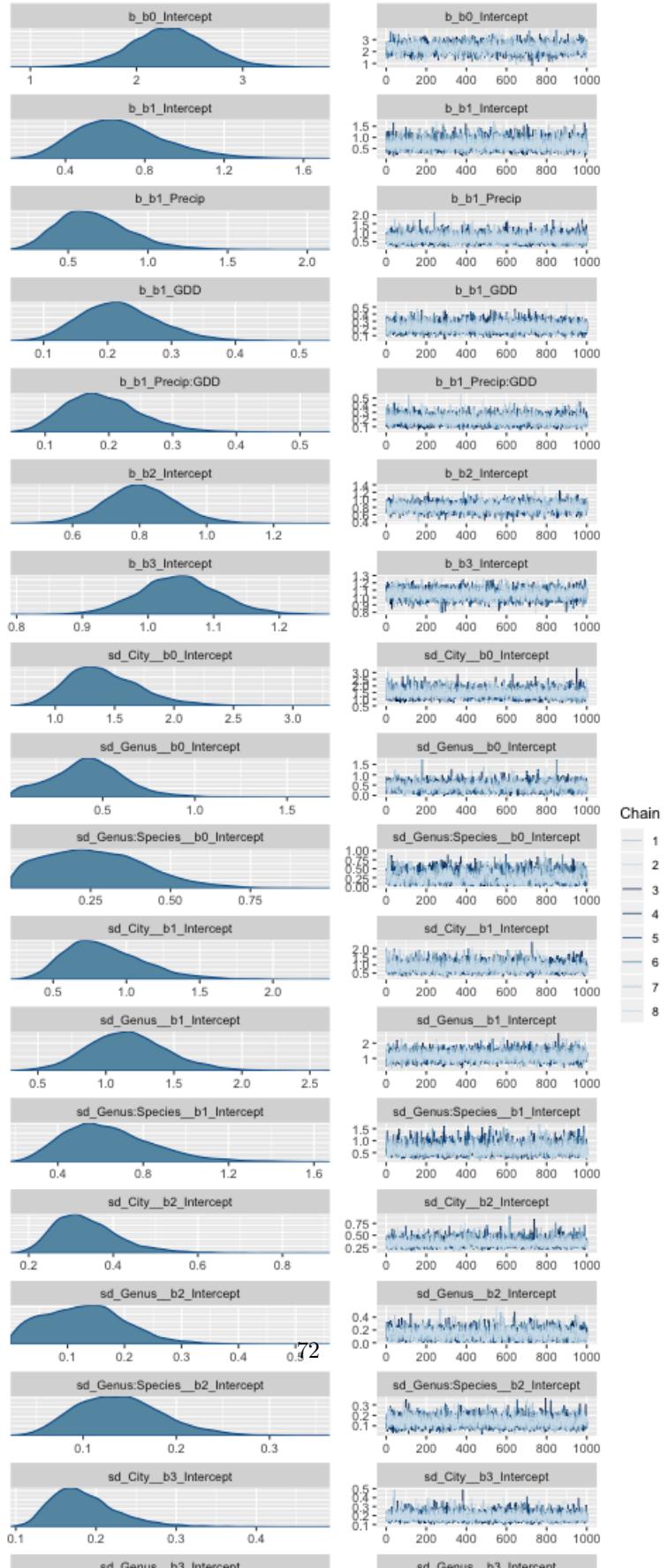
	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
shape	15.06	0.29	14.50	15.64	8000	1.00

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Warning message:

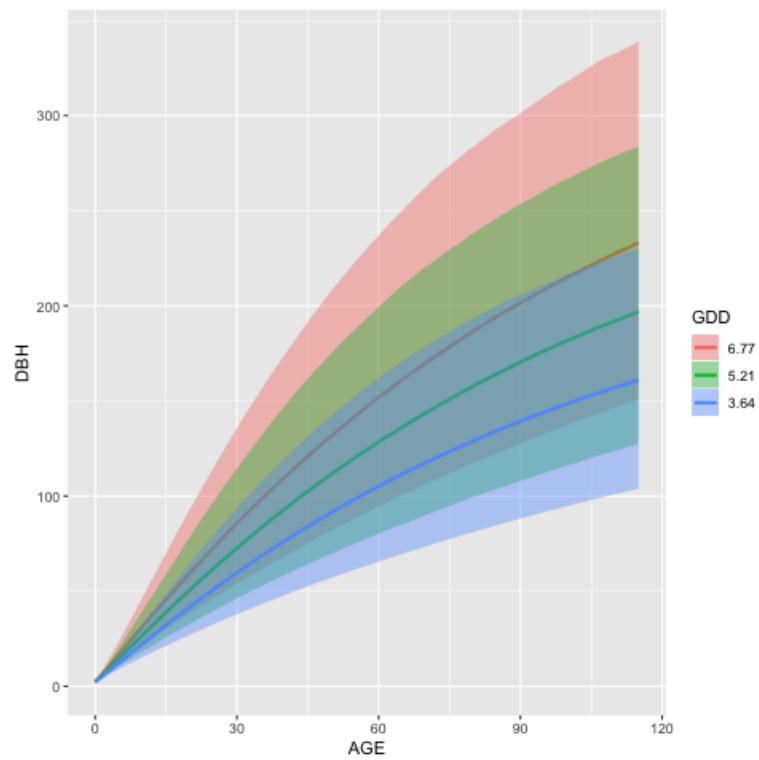
There were 3 divergent transitions after warmup. Increasing adapt\_delta above 0.8 may help. See <http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>

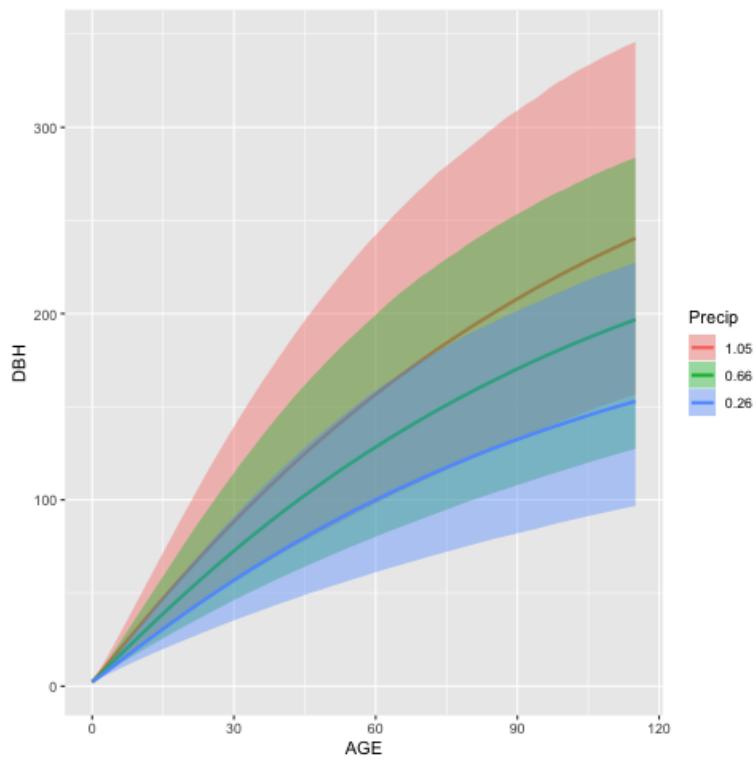


Chain

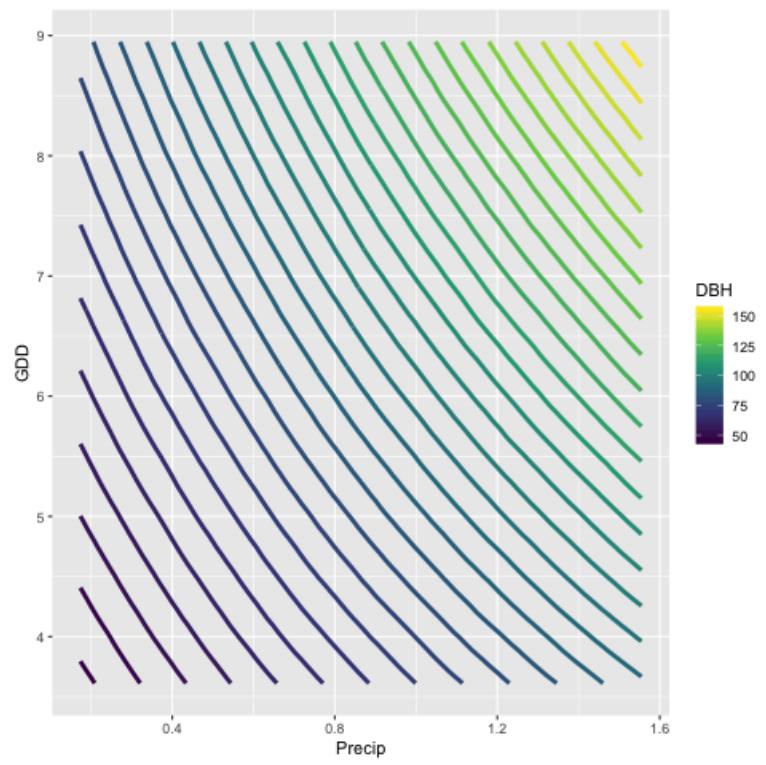
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

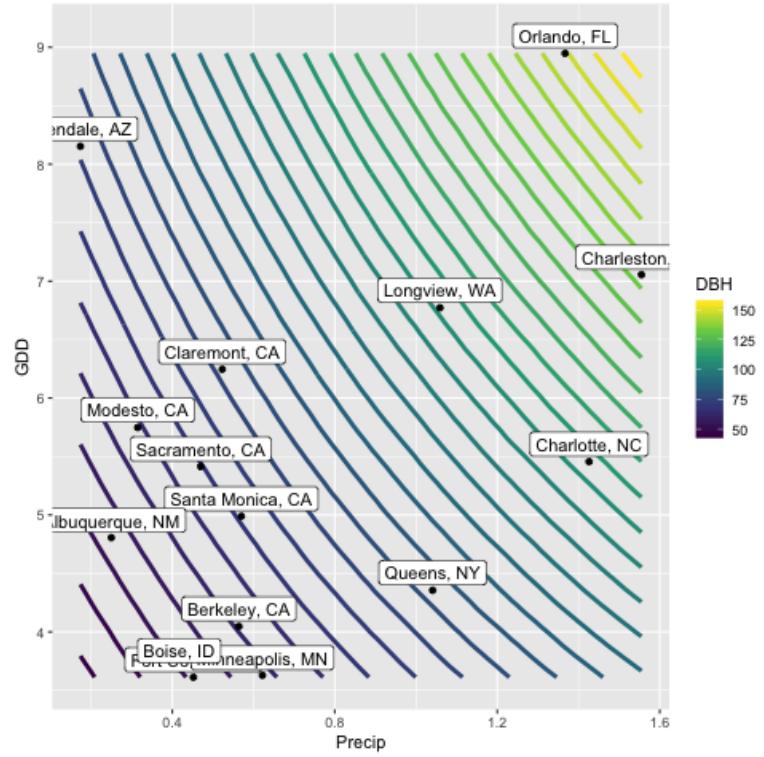
.72





```
precip.gdd <- marginal_effects(mod, effects = "Precip:GDD", surface = T)
```



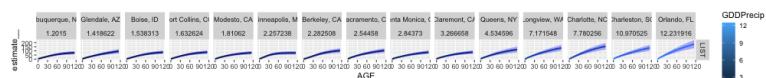


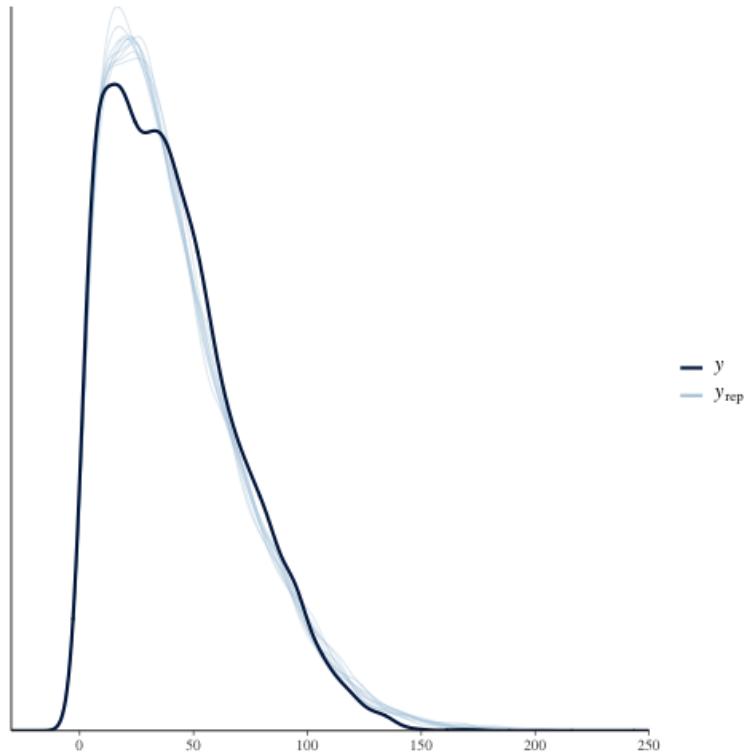
```

cond <- expand.grid(Species = unique(mod$data$Species), City = unique(mod$data$City))
cond <- left_join(cond, unique(select(mod$data, Species, Genus)))
cond <- left_join(cond, unique(select(mod$data, City, Precip, GDD)))
# cond <- filter(cond, Species %in% c("LIST"))

me <- marginal_effects(mod, effects = "AGE", conditions = cond, re_formula = NULL,

```





### 6.2.7 TODO genus: many; species: many; cities: many; climate: b3linint; hetero: no; family: Gamma Would need to fix priors!

1. model R code

```

library(dplyr)
library(brms)
source("allo_functions.R")

genus <- "many"
species <- "many"
cities <- "many"
climate <- "b3linint"
hetero <- "no"
family <- "Gamma"

data_form <- formula(DBH ~ b0 + 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3))))
b0_form <- formula(b0 ~ (1 | City) + (1 | Genus/Species))
b1_form <- formula(b1 ~ (1 | City) + (1 | Genus/Species))
b2_form <- formula(b2 ~ (1 | City) + (1 | Genus/Species))

```

```

b3_form <- formula(b3 ~ Precip * GDD + (1 | City) + (1 | Genus / Species))

form <- bf(data_form, b0_form, b1_form, b2_form, b3_form, nl = T)

nlprior <- c(prior(gamma(7.5, 3), nlpars = "b0", lb = 0),
            prior(gamma(8, 6), nlpars = "b1", lb = 0),
            prior(gamma(8, 8), nlpars = "b2", lb = 0),
            prior(gamma(8, 6), nlpars = "b3", lb = 0),
            prior(gamma(5, .3), class = "shape"),
            prior(gamma(1.3, 1.3), class = "sd", nlpars = "b0"),
            prior(gamma(10, 10), class = "sd", nlpars = "b1"),
            prior(gamma(1.3, 1.3), class = "sd", nlpars = "b2"),
            prior(gamma(1.3, 1.3), class = "sd", nlpars = "b3"))

d <- readRDS("../data/age_dbh_testing.rds")

mod <- brm(form,
            data = d,
            prior = nlprior,
            family = Gamma("identity"),
            chains = 8, cores = 8, init_r = .3, iter = 2000)

saveRDS(mod, paste0("../models/genus_", genus, "_species_", species, "_cities_", cities, "."))

```

2. tangle

3. send to krusty

```
rsync -avz genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma.
```

4. run on krusty

run from krusty terminal

```
ssh krusty
cd allo/code
nohup R CMD BATCH genus_many_species_many_cities_many_climate_b3linint_hetero_no_family
exit
```

```
cat genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma.Rout
```

5. pull back from krusty

```
rsync -avz erker@krusty:~/allo/models/genus_many_species_many_cities_many_climate_b3lin
```

6. assess model

```
mod_genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma <- read  
mod <- mod_genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma
```

```
.../figs/pairs_genus_many_species_many_cities_many_climate_b3linint_hetero_no_family
```

```
summary(mod)
```

```
Family: gamma  
Links: mu = identity; shape = identity  
Formula: DBH ~ b0 + 100 * b1 * (1 - exp(-(b2/100) * AGE^(b3)))  
        b0 ~ (1 | City) + (1 | Genus/Species)  
        b1 ~ (1 | City) + (1 | Genus/Species)  
        b2 ~ (1 | City) + (1 | Genus/Species)  
        b3 ~ Precip * GDD + (1 | City) + (1 | Genus/Species)  
Data: d (Number of observations: 5548)  
Samples: 8 chains, each with iter = 2000; warmup = 1000; thin = 1;  
         total post-warmup samples = 8000
```

Group-Level Effects:

~City (Number of levels: 15)

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	1.30	0.33	0.85	2.09	12	1.29
sd(b1_Intercept)	0.52	0.17	0.28	0.89	6	1.83
sd(b2_Intercept)	0.66	0.34	0.27	1.20	4	4.38
sd(b3_Intercept)	0.73	0.39	0.24	1.31	6	1.85

~Genus (Number of levels: 18)

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	0.54	0.31	0.04	1.13	5	1.93
sd(b1_Intercept)	0.55	0.17	0.28	0.79	7	1.58
sd(b2_Intercept)	0.47	0.39	0.03	1.13	4	4.95
sd(b3_Intercept)	0.54	0.38	0.13	1.24	4	6.31

~Genus:Species (Number of levels: 25)

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
sd(b0_Intercept)	0.59	0.32	0.09	1.15	5	2.62
sd(b1_Intercept)	0.46	0.22	0.19	0.83	5	2.59
sd(b2_Intercept)	0.49	0.49	0.05	1.41	4	10.84
sd(b3_Intercept)	0.40	0.43	0.02	1.14	4	8.20

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
b0_Intercept	2.75	0.46	2.11	3.73	27	1.23
b1_Intercept	1.03	0.36	0.48	1.62	5	2.40
b2_Intercept	0.93	0.14	0.68	1.23	10	1.40
b3_Intercept	0.65	0.25	0.27	1.08	5	2.32
b3_Precip	0.52	0.27	0.16	0.97	5	2.22
b3_GDD	0.26	0.31	0.04	0.91	4	8.89
b3_Precip:GDD	0.27	0.36	0.01	0.93	4	6.22

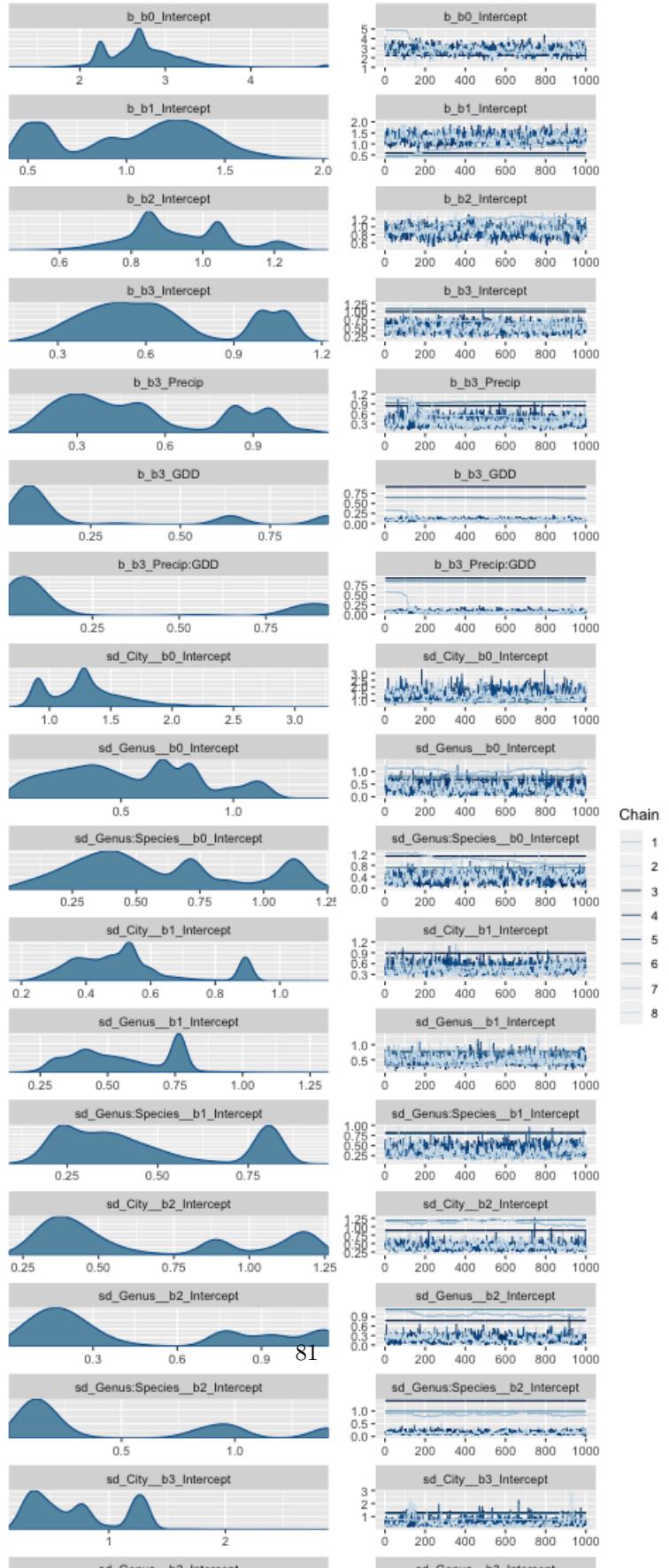
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
shape	11.32	5.36	2.47	15.25	4	4.96

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Warning messages:

- 1: The model has not converged (some Rhats are > 1.1). Do not analyse the results! We recommend running more iterations and/or setting stronger priors.
- 2: There were 2173 divergent transitions after warmup. Increasing adapt\_delta above 0.8. See <http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>



.. /figs/marginal\_effects\_AGEGLD\_genus\_many\_species\_many\_cities\_many\_climate\_b3linin

```
../figs/marginal_effects_AGEPrecip_genus_many_species_many_cities_many_climate_b3li
```

```
precip.gdd <- marginal_effects(mod, effects = "Precip:GDD", surface = T)
```

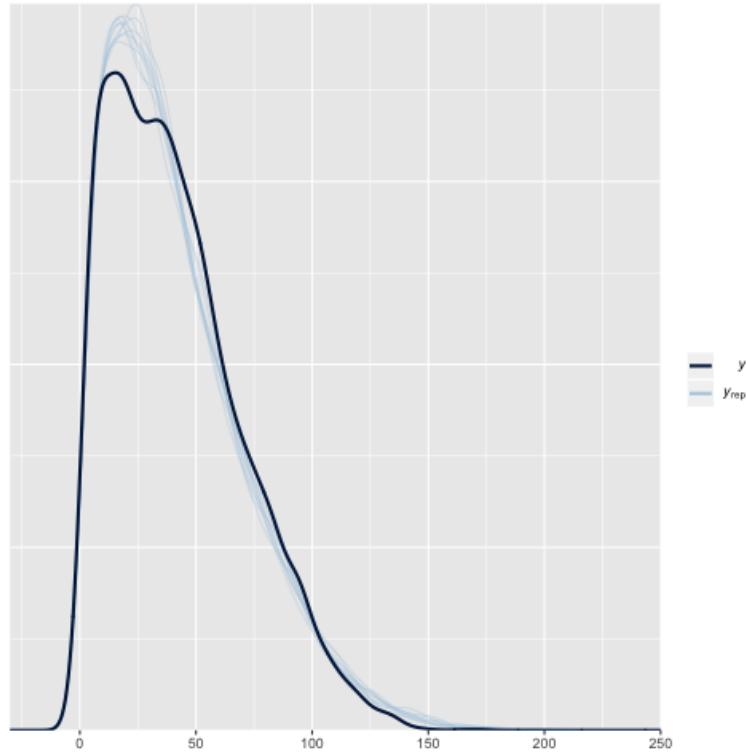
..../figs/marginal\_effects\_surface\_PrecipGDD\_genus\_many\_species\_many\_cities\_many\_clim

```
../figs/marginal_effects_surface_PrecipCDD_genus_many_species_many_cities_many_clim
```

```
cond <- expand.grid(Species = unique(mod$data$Species), City = unique(mod$data$City))
cond <- left_join(cond, unique(select(mod$data, Species, Genus)))
cond <- left_join(cond, unique(select(mod$data, City, Precip, GDD)))
# cond <- filter(cond, Species %in% c("LIST"))

me <- marginal_effects(mod, effects = "AGE", conditions = cond, re_formula = NULL,
```

..../figs/marginal\_effects\_BySpecies\_Genus\_many\_species\_many\_cities\_many\_climate\_b3li



#### 6.2.8 Full model

$$DBH \sim N(\beta_0 * (1 - \exp(-\beta_1 * AGE^{\beta_2})), \sigma)$$

$$\beta_0, \beta_1, \beta_2 \sim$$

$$\sigma \sim$$

species within genus within ...

Model selection

1. Do all the betas vary by groups (City, Species, or both), or just beta0?
  - (a) There would be 4 models to compare:
    - i.  $\beta_0 + \beta_1 + \beta_2 \sim \text{City} + \text{Species}$
    - ii.  $\beta_0 \sim \text{City} + \text{Species}$ ,  $\beta_1 + \beta_2 \sim 1$
    - iii.  $\beta_0 \sim \text{City} + \text{Species}$ ,  $\beta_1 + \beta_2 \sim \text{City}$
    - iv.  $\beta_0 \sim \text{City} + \text{Species}$ ,  $\beta_1 + \beta_2 \sim \text{Species}$
  - (b) It might be worth doing this test on a reasonable subset of the data, otherwise this may take a very long time

2. What is the proper form of the climate variables on the beta's?

(a)

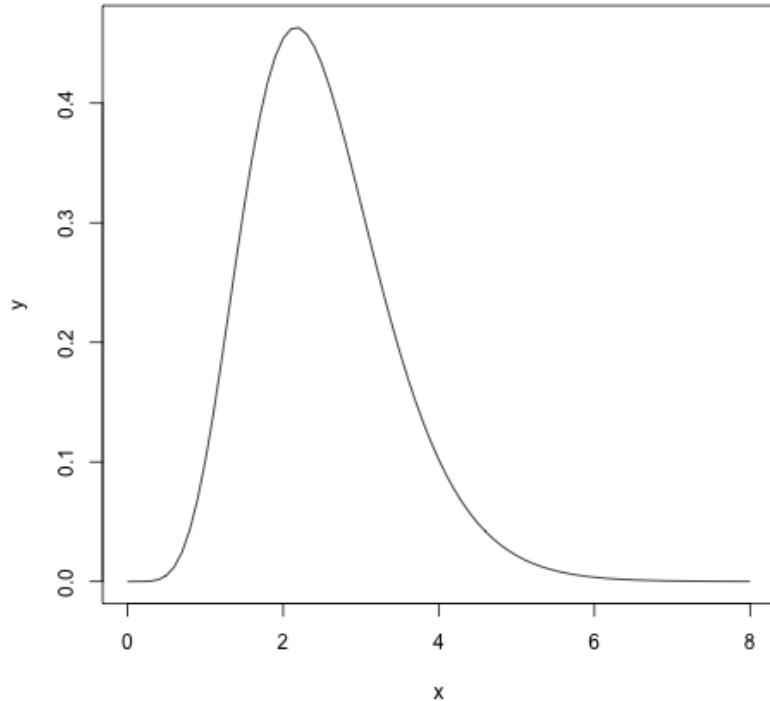
#### 6.2.9 for diagnostics

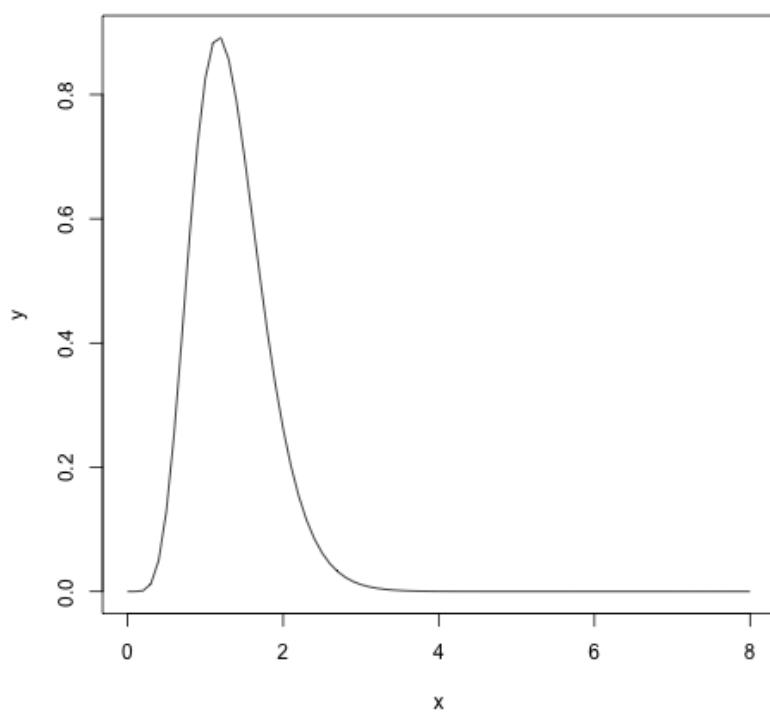
<https://arxiv.org/pdf/1709.01449.pdf>

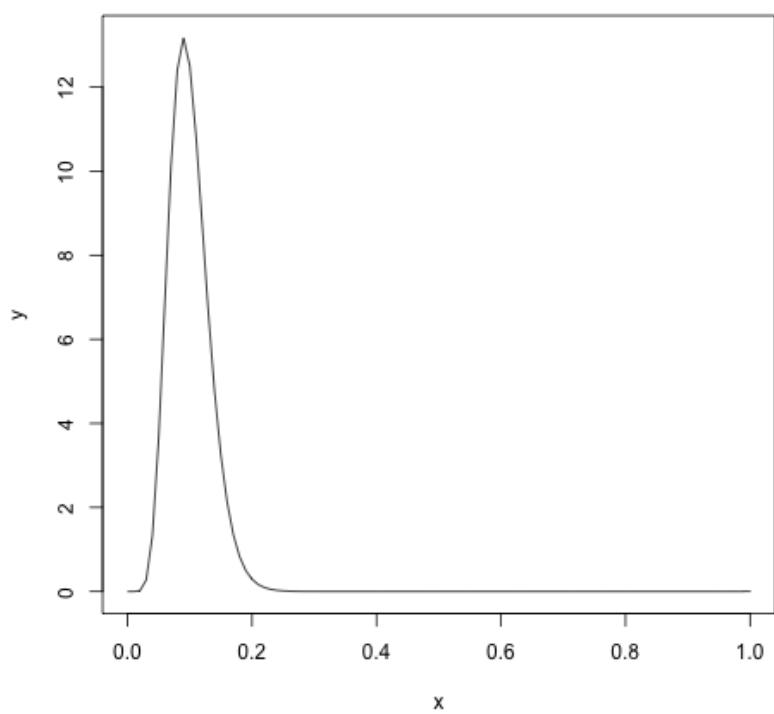
#### 6.2.10 look at distributions

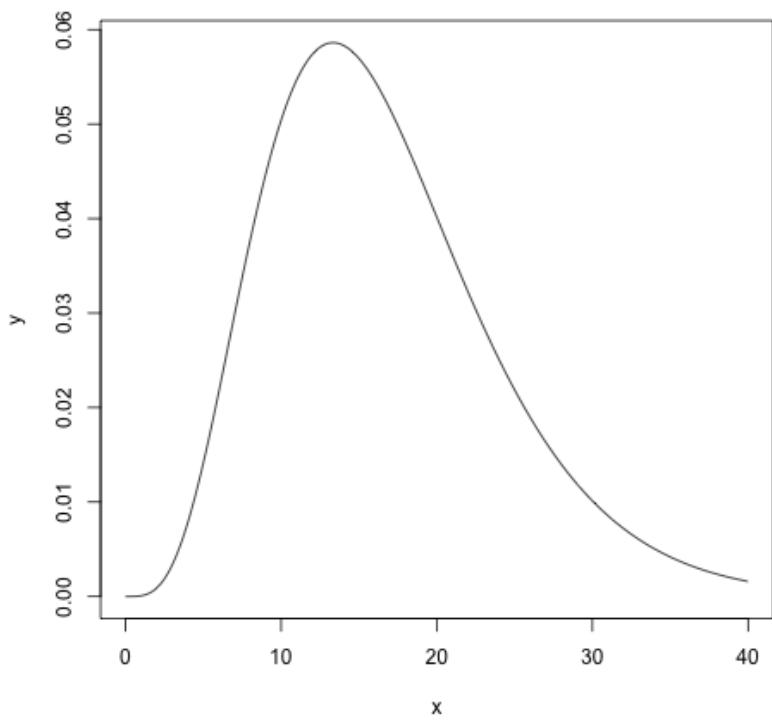
Priors

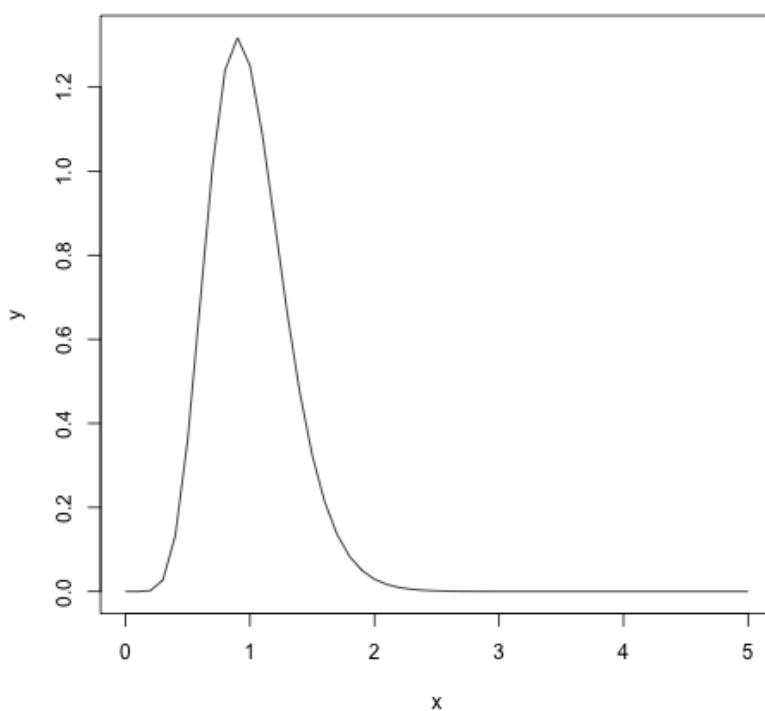
gamma alpha called shape beta called rate mean is alpha/beta; variance is alpha/beta<sup>2</sup>

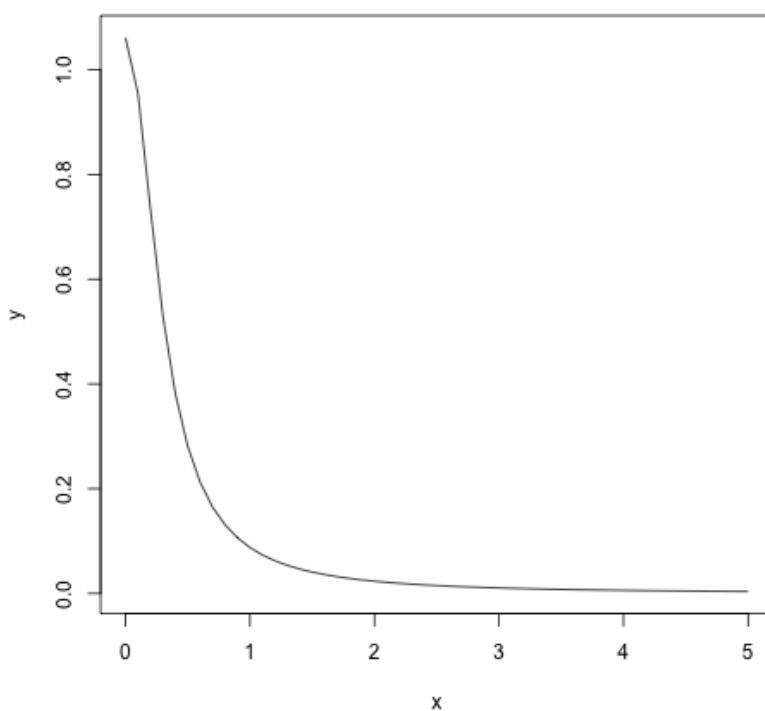


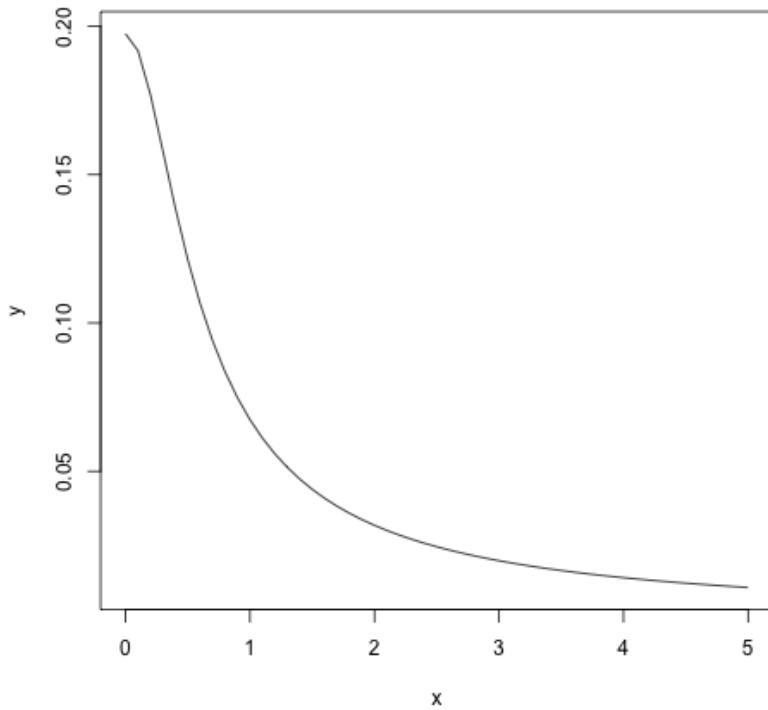












## 7 Results

figure showing the age versus dbh for several species and city combinations. overlay our models and the existing models for comparison.

figure showing the marginal effects smooth plot of GDD versus precip, with a contour showing in what climate space 50%, 75%, and 95% of the US population lives. The marginal effects and the uncertainty in them could also be mapped back into geographic space and the map could show that. That would be neat.

## 8 Discusssoins

make better

- more data, duh, perhaps used results to identify where to sample
- more cities, this is important for interpolation across climate space. Could allow for nonlinear relationships and for more variables.
- better climate predictors

- interactions between climate and species.
- use phylogenetic distance, gaussian process, instead of multiple levels of taxonomy.
- extend species with species level predictors (leaf morphology, wood characteristics, shade tolerance, etc).
- smarter priors (e.g. max dbh based on champion trees?) is this possible? I think it would be a very neat extension, but need to think about how these champions are not urban trees most the time. They provide the upper limit on the asymptote, but for urban trees the asymptote could be quite lower.
- incorporate uncertainty in AGE

## 9 Exploring age and dbh

### 9.1 load libraries

### 9.2 functions

```
options(asciiType = "org")
ascii.nowarn.print <- function(x,...) {
  op <- options(warn = -1)
  #       on.exit(options(op))

  suppressWarnings(print(ascii(x,...)))

}
```

### 9.3 read in data

```
d <- read_csv("../data/RDS-2016-0005/Data/TS3_Raw_tree_data.csv")
#str(d)
```

### 9.4 fix some species things

#### 9.4.1 fix lower case species codes

```
d$SpCode <- toupper(d$SpCode)
```

#### 9.4.2 fix QUAG1 to be QUAG

```
d$SpCode[d$SpCode == "QUAG1"] <- "QUAG"
```

### 9.4.3 fix common names

Not all are fixed!

```
d$CommonName[d$CommonName == "Kurrajong"] <- "Kurrajong/Bottle tree"
d$CommonName[d$CommonName == "Bottle tree"] <- "Kurrajong/Bottle tree"

d$CommonName[d$CommonName == "Apple"] <- "Apple/Crabapple"
d$CommonName[d$CommonName == "Crabapple"] <- "Apple/Crabapple"

d$CommonName[d$CommonName == "silver maple"] <- "Silver maple"
d$CommonName[d$CommonName == "camphor tree"] <- "Camphor tree"
d$CommonName[d$CommonName == "ginkgo"] <- "Ginkgo"
d$CommonName[d$CommonName == "honeylocust"] <- "Honeylocust"
d$CommonName[d$CommonName == "ginkgo"] <- "Ginkgo"
d$CommonName[d$CommonName == "common crapemyrtle"] <- "Common crapemyrtle"
d$CommonName[d$CommonName == "sweetgum"] <- "Sweetgum"
d$CommonName[d$CommonName == "southern magnolia"] <- "Southern magnolia"
```

## 9.5 species are

TreeType	FunctionalType
BDL	broadleaf deciduous
BDM	broadleaf deciduous
BDS	broadleaf deciduous
BEL	broadleaf evergreen
BEM	broadleaf evergreen
BES	broadleaf evergreen
CEL	conifer evergreen
CEM	conifer evergreen
CES	conifer evergreen
PEL	palm evergreen
PEM	palm evergreen
PES	palm evergreen

I got rid of the small medium and large distinctions for simplification here.

```
sp <- d %>% select(SpCode, ScientificName, TreeType) %>%
  left_join(tt) %>%
  select(-TreeType) %>%
  unique() %>%
  arrange(ScientificName) %>%
  ascii.nowarn.print()
```

Joining, by = "TreeType"			
	SpCode	ScientificName	FunctionalType
1	ACFA	<i>Acacia farnesiana</i>	broadleaf deciduous
2	ACME	<i>Acacia melanoxylon</i>	broadleaf evergreen
3	ACSA3	<i>Acacia salicina</i>	broadleaf evergreen
4	ACMA	<i>Acer macrophyllum</i>	broadleaf deciduous
5	ACNE	<i>Acer negundo</i>	broadleaf deciduous
6	ACPA	<i>Acer palmatum</i>	broadleaf deciduous
7	ACPL	<i>Acer platanoides</i>	broadleaf deciduous
8	ACRU	<i>Acer rubrum</i>	broadleaf deciduous
9	ACSA1	<i>Acer saccharinum</i>	broadleaf deciduous
10	ACSA2	<i>Acer saccharum</i>	broadleaf deciduous
11	AEHI	<i>Aesculus hippocastanum</i>	broadleaf deciduous
12	BABL	<i>Bauhinia x blakeana</i>	broadleaf evergreen
13	BENI	<i>Betula nigra</i>	broadleaf deciduous
14	BEPE	<i>Betula pendula</i>	broadleaf deciduous
15	BRPO	<i>Brachychiton populneum</i>	broadleaf evergreen
16	BUCA	<i>Butia capitata</i>	palm evergreen
17	CACI	<i>Callistemon citrinus</i>	broadleaf evergreen
18	CADE2	<i>Calocedrus decurrens</i>	conifer evergreen
19	CAIN4	<i>Calophyllum inophyllum</i>	broadleaf evergreen
20	CABEF	<i>Carpinus betulus 'Fastigiata'</i>	broadleaf deciduous
21	CAIL	<i>Carya illinoiensis</i>	broadleaf deciduous
22	CANE33	<i>Cassia x nealiae</i>	broadleaf deciduous
23	CAEQ	<i>Casuarina equisetifolia</i>	broadleaf evergreen
24	CASP	<i>Catalpa speciosa</i>	broadleaf deciduous
25	CEDE	<i>Cedrus deodara</i>	conifer evergreen
26	CELA	<i>Celtis laevigata</i>	conifer evergreen
27	CEOCC	<i>Celtis occidentalis</i>	broadleaf deciduous
28	CESI4	<i>Celtis sinensis</i>	broadleaf deciduous
29	CESI3	<i>Ceratonia siliqua</i>	broadleaf evergreen
30	CECA	<i>Cercis canadensis</i>	broadleaf deciduous
31	CHLI	<i>Chilopsis linearis</i>	broadleaf deciduous
32	CICA	<i>Cinnamomum camphora</i>	broadleaf evergreen
33	CISP2	<i>Citharexylum spinosum</i>	broadleaf evergreen
34	CONU	<i>Cocos nucifera</i>	palm evergreen
35	COERA2	<i>Conocarpus erectus var. argenteus</i>	broadleaf evergreen
36	COSU2	<i>Cordia subcordata</i>	broadleaf evergreen
37	COFL	<i>Cornus florida</i>	broadleaf deciduous
38	CR	<i>Crataegus sp.</i>	broadleaf deciduous
39	CRLA	<i>Crataegus x lavallei</i>	broadleaf deciduous
40	CUAN	<i>Cupaniopsis anacardioides</i>	broadleaf evergreen
41	DERE	<i>Delonix regia</i>	broadleaf deciduous
42	ELAN	<i>Elaeagnus angustifolia</i>	broadleaf deciduous
43	ELOR2	<i>Elaeodendron orientale</i>	broadleaf evergreen

44   ERJA   <i>Eriobotrya japonica</i>	broadleaf evergreen
45   EUFI81   <i>Eucalyptus ficifolia</i>	broadleaf evergreen
46   EUGL   <i>Eucalyptus globulus</i>	broadleaf evergreen
47   EUMI2   <i>Eucalyptus microtheca</i>	broadleaf evergreen
48   EUSI   <i>Eucalyptus sideroxylon</i>	broadleaf evergreen
49   FASY   <i>Fagus sylvatica</i>	broadleaf deciduous
50   FIBE   <i>Ficus benjamina</i>	broadleaf evergreen
51   FIMI   <i>Ficus thonningii</i>	broadleaf evergreen
52   FIDE6   <i>Filicium decipiens</i>	broadleaf evergreen
53   FRAM   <i>Fraxinus americana</i>	broadleaf deciduous
54   FRAN2   <i>Fraxinus angustifolia</i>	broadleaf deciduous
55   FRAN_R   <i>Fraxinus angustifolia</i> 'Raywood'	broadleaf deciduous
56   FREX_H   <i>Fraxinus excelsior</i> 'Hessei'	broadleaf deciduous
57   FRHO   <i>Fraxinus holotricha</i>	broadleaf deciduous
58   FRLA   <i>Fraxinus latifolia</i>	broadleaf deciduous
59   FRPE   <i>Fraxinus pennsylvanica</i>	broadleaf deciduous
60   FRPE_M   <i>Fraxinus pennsylvanica</i> 'Marshall'	broadleaf deciduous
61   FRUH   <i>Fraxinus uhdei</i>	broadleaf deciduous
62   FRVE   <i>Fraxinus velutina</i>	broadleaf deciduous
63   FRVE_G   <i>Fraxinus velutina</i> 'Modesto'	broadleaf deciduous
64   GIBI   <i>Ginkgo biloba</i>	broadleaf deciduous
65   GLTR   <i>Gleditsia triacanthos</i>	broadleaf deciduous
66   GYDI   <i>Gymnocladus dioicus</i>	broadleaf deciduous
67   ILOP   <i>Ilex opaca</i>	broadleaf evergreen
68   ILPA2   <i>Ilex paraguariensis</i>	broadleaf evergreen
69   JAMI   <i>Jacaranda mimosifolia</i>	broadleaf deciduous
70   JUNI   <i>Juglans nigra</i>	broadleaf deciduous
71   JUVI   <i>Juniperus virginiana</i>	conifer evergreen
72   JUSI   <i>Juniperus virginiana</i> var. <i>silicicola</i>	conifer evergreen
73   KOELFO   <i>Koelreuteria elegans</i>	broadleaf deciduous
74   KOPA   <i>Koelreuteria paniculata</i>	broadleaf deciduous
75   LAIN   <i>Lagerstroemia indica</i>	broadleaf deciduous
76   LA6   <i>Lagerstroemia</i> sp.	broadleaf deciduous
77   LASP   <i>Lagerstroemia speciosa</i>	broadleaf deciduous
78   LIST   <i>Liquidambar styraciflua</i>	broadleaf deciduous
79   LITU   <i>Liriodendron tulipifera</i>	broadleaf deciduous
80   MAGR   <i>Magnolia grandiflora</i>	broadleaf evergreen
81   PYAN   <i>Malus angustifolia</i>	broadleaf deciduous
82   MA2   <i>Malus</i> sp.	broadleaf deciduous
83   MEQU   <i>Melaleuca quinquenervia</i>	broadleaf evergreen
84   MEEX   <i>Metrosideros excelsa</i>	broadleaf evergreen
85   MOAL   <i>Morus alba</i>	broadleaf deciduous
86   MO   <i>Morus</i> sp.	broadleaf deciduous
87   OLEU   <i>Olea europaea</i>	broadleaf evergreen
88   PAAC   <i>Parkinsonia aculeata</i>	broadleaf deciduous
89   CEFL   <i>Parkinsonia florida</i>	broadleaf deciduous

90   PHCA   <i>Phoenix canariensis</i>	palm evergreen
91   PHDA4   <i>Phoenix dactylifera</i>	palm evergreen
92   PIPU   <i>Picea pungens</i>	conifer evergreen
93   PIBR2   <i>Pinus brutia</i>	conifer evergreen
94   PICA   <i>Pinus canariensis</i>	conifer evergreen
95   PICO   <i>Pinus contorta</i>	conifer evergreen
96   PIEC   <i>Pinus echinata</i>	conifer evergreen
97   PIED   <i>Pinus edulis</i>	conifer evergreen
98   PIEL2   <i>Pinus eldarica</i>	conifer evergreen
99   PIEL   <i>Pinus elliottii</i>	conifer evergreen
100   PIHA   <i>Pinus halepensis</i>	conifer evergreen
101   PINI   <i>Pinus nigra</i>	conifer evergreen
102   PIPO   <i>Pinus ponderosa</i>	conifer evergreen
103   PIRA   <i>Pinus radiata</i>	conifer evergreen
104   PIST   <i>Pinus strobus</i>	conifer evergreen
105   PISY   <i>Pinus sylvestris</i>	conifer evergreen
106   PITA   <i>Pinus taeda</i>	conifer evergreen
107   PITH   <i>Pinus thunbergiana</i>	conifer evergreen
108   PICH   <i>Pistacia chinensis</i>	broadleaf deciduous
109   PIUN   <i>Pittosporum undulatum</i>	broadleaf evergreen
110   PLOC   <i>Platanus occidentalis</i>	broadleaf deciduous
111   PLRA   <i>Platanus racemosa</i>	broadleaf deciduous
112   PLAC   <i>Platanus x acerifolia</i>	broadleaf deciduous
113   THOR   <i>Platycladus orientalis</i>	conifer evergreen
114   POMA   <i>Podocarpus macrophyllus</i>	conifer evergreen
115   POAN   <i>Populus angustifolia</i>	broadleaf deciduous
116   POTR2   <i>Populus balsamifera</i> subsp. <i>trichocarpa</i>	broadleaf deciduous
117   PODE   <i>Populus deltoides</i>	broadleaf deciduous
118   POFR   <i>Populus fremontii</i>	broadleaf deciduous
119   POSA   <i>Populus sargentii</i>	broadleaf deciduous
120   PRCH   <i>Prosopis chilensis</i>	broadleaf deciduous
121   PRCA   <i>Prunus caroliniana</i>	broadleaf evergreen
122   PRCE   <i>Prunus cerasifera</i>	broadleaf deciduous
123   PRCE   <i>Prunus cerasifera</i> cvs.	broadleaf deciduous
124   PRSE2   <i>Prunus serrulata</i>	broadleaf deciduous
125   PR   <i>Prunus</i> sp.	broadleaf deciduous
126   PRYE   <i>Prunus yedoensis</i>	broadleaf deciduous
127   PSME   <i>Pseudotsuga menziesii</i>	conifer evergreen
128   PYCA   <i>Pyrus calleryana</i>	broadleaf deciduous
129   PYCA_B   <i>Pyrus calleryana</i> 'Bradford'	broadleaf deciduous
130   PYCA   <i>Pyrus calleryana</i> cvs.	broadleaf deciduous
131   PYKA   <i>Pyrus kawakamii</i>	broadleaf evergreen
132   PY   <i>Pyrus</i> sp.	broadleaf deciduous
133   QUAG   <i>Quercus agrifolia</i>	broadleaf evergreen
134   QUAL   <i>Quercus alba</i>	broadleaf deciduous
135   QUIL2   <i>Quercus ilex</i>	broadleaf evergreen

136   QULA2   <i>Quercus laurifolia</i>	broadleaf deciduous
137   QULO   <i>Quercus lobata</i>	broadleaf deciduous
138   QUMA1   <i>Quercus macrocarpa</i>	broadleaf deciduous
139   QUNI   <i>Quercus nigra</i>	broadleaf deciduous
140   QUPA   <i>Quercus palustris</i>	broadleaf deciduous
141   QUPH   <i>Quercus phellos</i>	broadleaf deciduous
142   QURU   <i>Quercus rubra</i>	broadleaf deciduous
143   QUSH   <i>Quercus shumardii</i>	broadleaf deciduous
144   QUVI   <i>Quercus virginiana</i>	broadleaf evergreen
145   RHLA   <i>Rhus lancea</i>	broadleaf evergreen
146   ROPS   <i>Robinia pseudoacacia</i>	broadleaf deciduous
147   SAPA   <i>Sabal palmetto</i>	palm evergreen
148   PISA2   <i>Samanea saman</i>	broadleaf deciduous
149   SCMO   <i>Schinus molle</i>	broadleaf evergreen
150   SCTE   <i>Schinus terebinthifolius</i>	broadleaf evergreen
151   SESE   <i>Sequoia sempervirens</i>	conifer evergreen
152   SWMA   <i>Swietenia mahagoni</i>	broadleaf evergreen
153   SYRO   <i>Syagrus romanzoffiana</i>	palm evergreen
154   TAAR   <i>Tabebuia aurea</i>	broadleaf evergreen
155   TAPA   <i>Tabebuia heterophylla</i>	broadleaf evergreen
156   TAOC   <i>Tabebuia ochracea</i> subsp. <i>neochrysantha</i>	broadleaf evergreen
157   TIAM   <i>Tilia americana</i>	broadleaf deciduous
158   TICO   <i>Tilia cordata</i>	broadleaf deciduous
159   TITO   <i>Tilia tomentosa</i>	broadleaf deciduous
160   TRSE6   <i>Triadica sebifera</i>	broadleaf deciduous
161   TRCO   <i>Tristaniopsis conferta</i>	broadleaf evergreen
162   ULAL   <i>Ulmus alata</i>	broadleaf deciduous
163   ULAM   <i>Ulmus americana</i>	broadleaf deciduous
164   ULPA   <i>Ulmus parvifolia</i>	broadleaf deciduous
165   ULPU   <i>Ulmus pumila</i>	broadleaf deciduous
166   VEME   <i>Veitchia merrillii</i>	palm evergreen
167   WAFI   <i>Washingtonia filifera</i>	palm evergreen
168   WARO   <i>Washingtonia robusta</i>	palm evergreen
169   ZESE   <i>Zelkova serrata</i>	broadleaf deciduous

## 9.6 tidy a few names and select variables of interest here

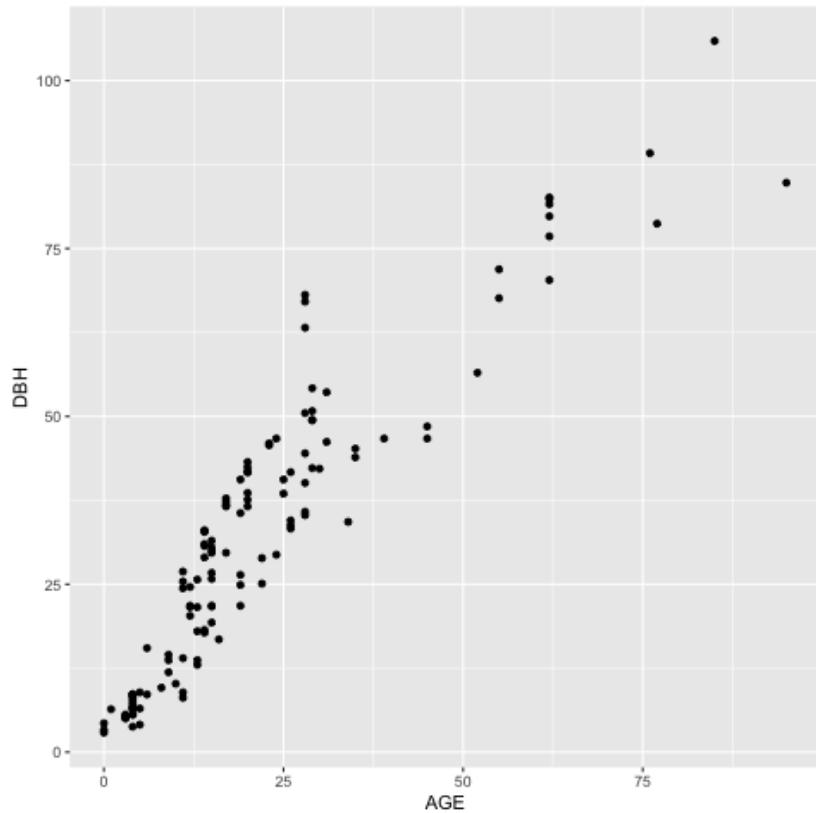
```
d <- d %>% rename(DBH = 'DBH (cm)', Leaf = 'Leaf (m2)') %>% select(Region, City, TreeID, SpC)

summary(d$Leaf)
sum(d$Leaf == -1) / length(d$Leaf)
sum(d$DBH == -1) / length(d$DBH)

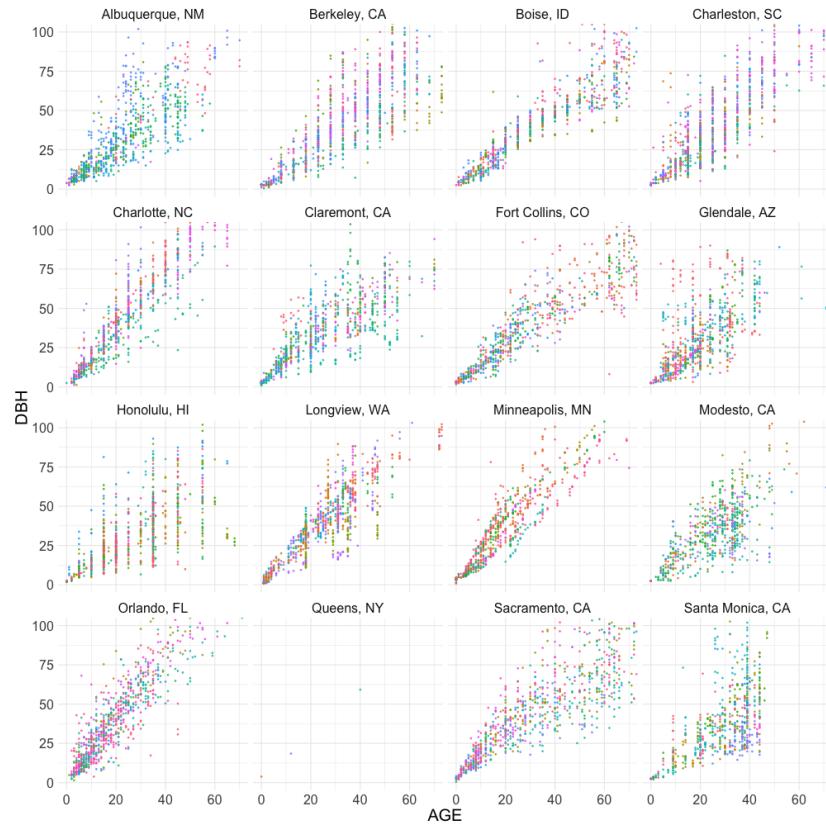
      Min. 1st Qu. Median   Mean 3rd Qu.   Max.
      -1.0    12.9   116.1  302.5   389.2  9516.0
[1] 0.1143094
[1] 0
```

```
d <- filter(d, DBH != -1, Age != -1) %>%
  rename(AGE = Age)
```

## 9.7 plot age versus dbh for all trees



## 9.8 AGE versus dbh by cities





## 9.9 AGE versus DBH by species



Woah, there are lots of species. Clearly there is some variability in the relationship between age and dbh across species.

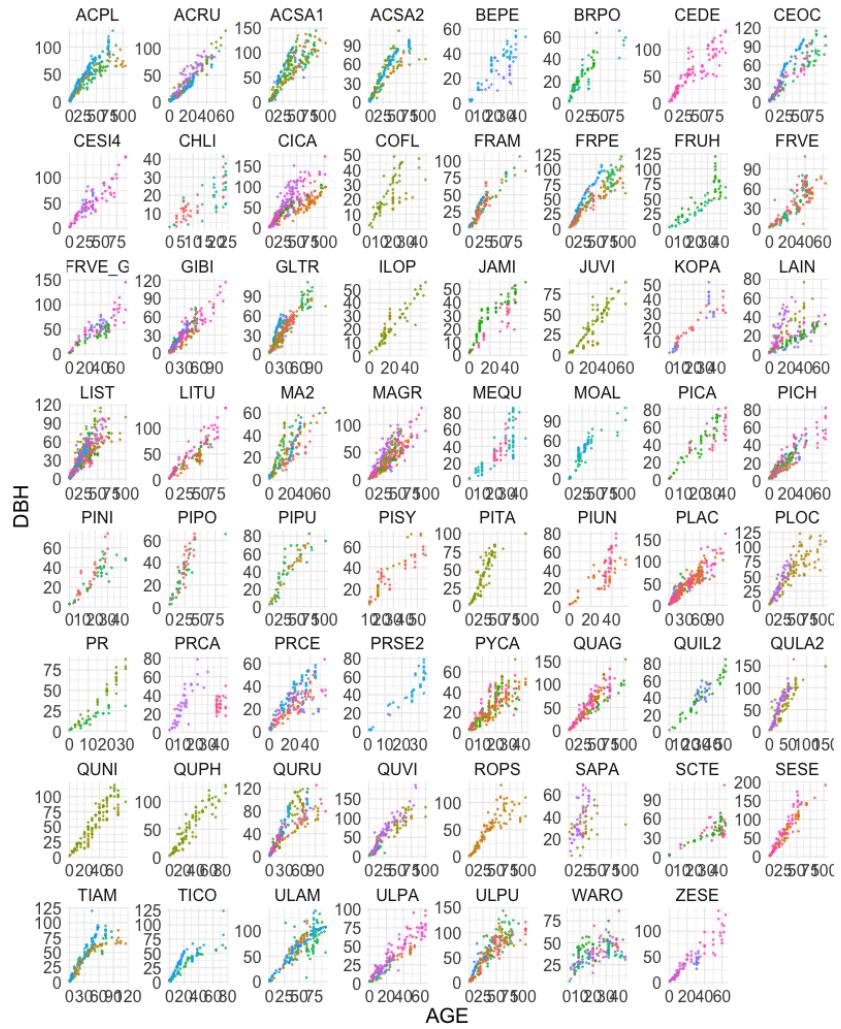
## 9.10 AGE versus DBH by species for species that are in more than 1 city

Does the relationship between AGE and DBH for a species change depending on the city where it is?

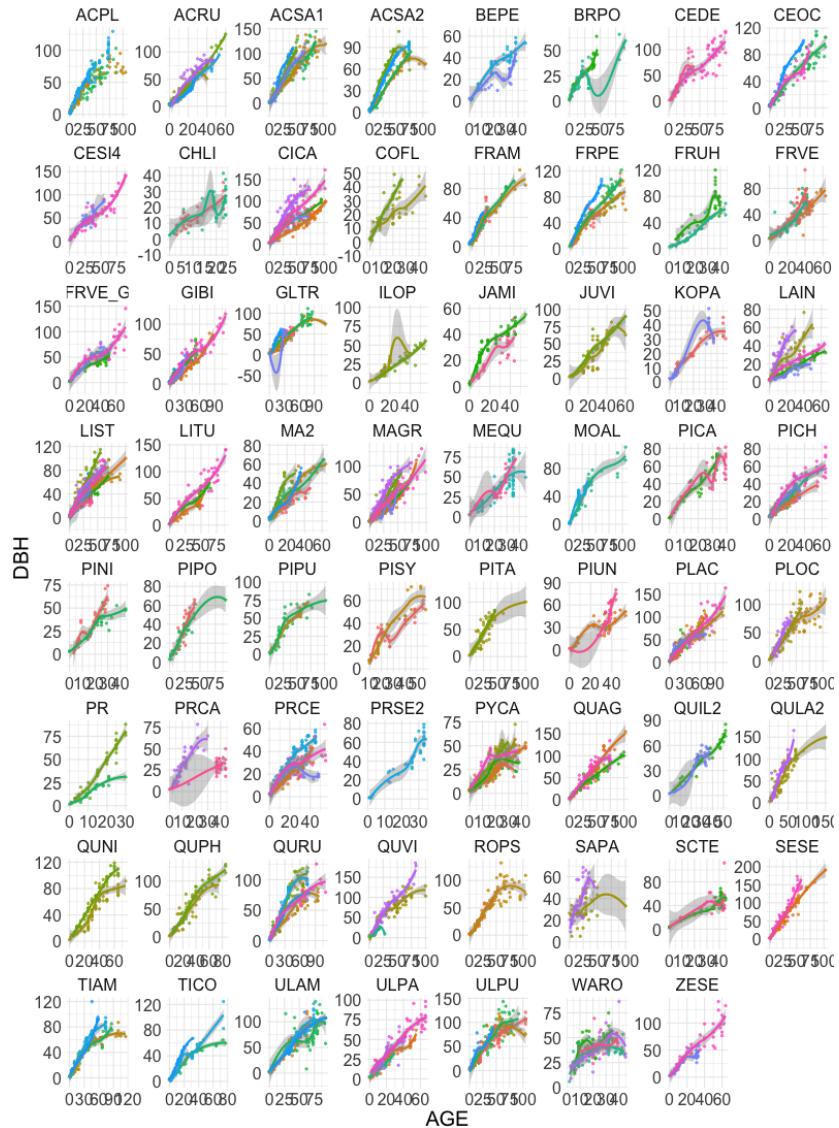
Is there evidence for a different equation for every species city combination?  
Or can we use one equation for each species, regardless of city?

```
sp.w.multiple.cities <- d %>% group_by(City, SpCode) %>% summarize(n = n()) %>% ungroup() %>%  
  summarize(n_cities_per_species = n()) %>%  
  filter(n_cities_per_species > 1) %>%  
  pull(SpCode)  
  
d.sp.w.multiple.cities <- filter(d, SpCode %in% sp.w.multiple.cities)
```

### 9.10.1 Each panel is a species, Each color is a different city



### 9.10.2 Adding Loess trend lines



ACPL's cities

```
filter(d.sp.w.multiple.cities, SpCode == "ACPL") %>% pull(City) %>% unique
[1] "Fort Collins, CO" "Minneapolis, MN" "Queens, NY"      "Boise, ID"
[5] "Longview, WA"
```

MOAL's cities

```

filter(d.sp.w.multiple.cities, SpCode == "MOAL") %>% pull(City) %>% unique
[1] "Glendale, AZ" "Longview, WA"

```

## 9.11 Plot Urban Tree Allometric equations on top of data

```

predict.allo <- function(x, EqName, a, b, c, d, e) {
  if(EqName == "loglogw1") {
    y = exp(a + b*log(log(x + 1) + c/2))
  }
  else if(EqName == "loglogw2") {
    y = exp(a + b*log(log(x + 1))+(sqrt(x) * (c/2)))
  }
  else if (EqName == "loglogw3") {
    y = exp(a + b*log(log(x + 1)) + x * c/2)
  }
  else if (EqName == "loglogw4") {
    y = exp(a + b*log(log(x + 1)) + x^2 * c/2)
  }
  else if (EqName == "expow1") {
    y = exp(a+ b * (x) + (c/2))
  }
  else if (EqName == "lin") {
    y = a + b * x
  }
  else if (EqName == "quad") {
    y = a + b * x + c* x^2
  }
  else if (EqName == "cub") {
    y = a+b * x+c *x^2 + d * x^3
  }
  else if (EqName == "quart") {
    y = a+b * x+c *x^2 + d * x^3 + e * x^4
  }
  return(y)
}

eqn <- read.csv("data/RDS-2016-0005/Data/TS6_Growth_coefficients_fromNatalie.csv", stringsAsFactors = TRUE)
eqn <- eqn %>%
  filter(Predicts.component %in% c("dbh"), Independent.variable == "age")

age_min_max = d %>%
  group_by(Region, SpCode) %>%

```

```

    summarize(minAGE = min(AGE, na.rm = T),
              maxAGE = max(AGE, na.rm = T))

eqn <- left_join(eqn, age_min_max)

DBH_min_max = d %>%
  group_by(Region, SpCode) %>%
  summarize(minDBH = min(DBH, na.rm = T),
            maxDBH = max(DBH, na.rm = T))

eqn <- left_join(eqn, DBH_min_max)

# fill in the NAs due to equations existing for spec
eqn$minAGE[is.na(eqn$minAGE)] <- 0
eqn$maxAGE[is.na(eqn$maxAGE)] <- 100

newdata <- lapply(1:nrow(eqn), function(i) {
  x <- seq(eqn$minAGE[i], eqn$maxAGE[i], (eqn$maxAGE[i] - eqn$minAGE[i]) / 20)
  cbind(x, eqn[i,])
})

newdata <- bind_rows(newdata)

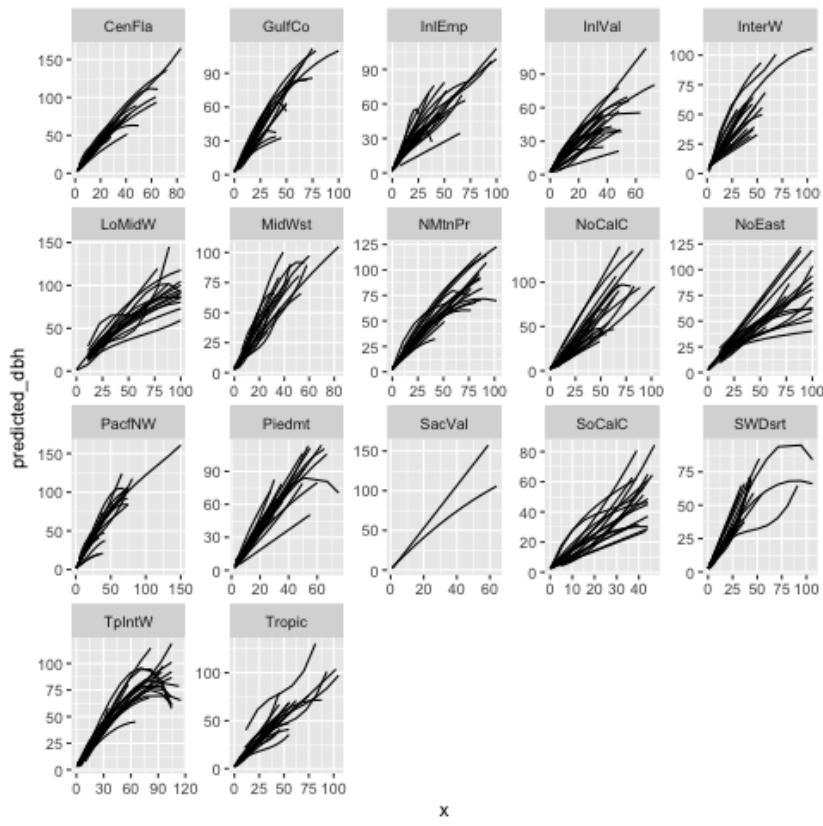
predictions <- newdata %>% rowwise %>% mutate(predicted_dbh = predict.allo(x = x, EqName = EqName))

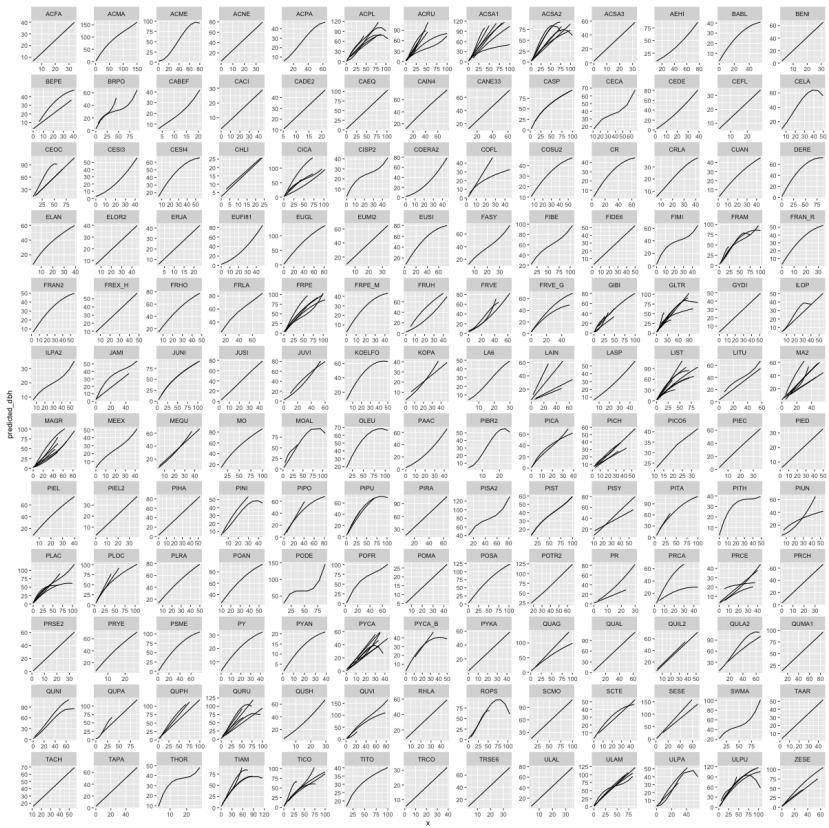
#filter out predictions that are outside range of data
predictions_apprange <- predictions %>%
  filter(predicted_dbh > Apps.min & predicted_dbh < Apps.max)

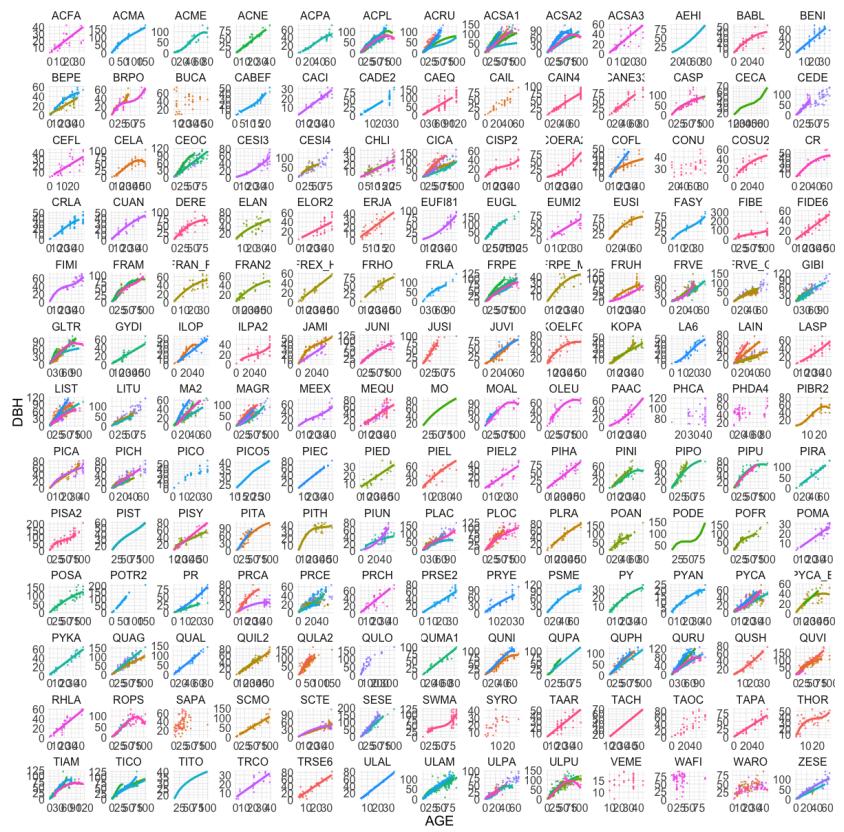
predictions_datarange <- predictions %>%
  filter(predicted_dbh > minDBH & predicted_dbh < maxDBH)

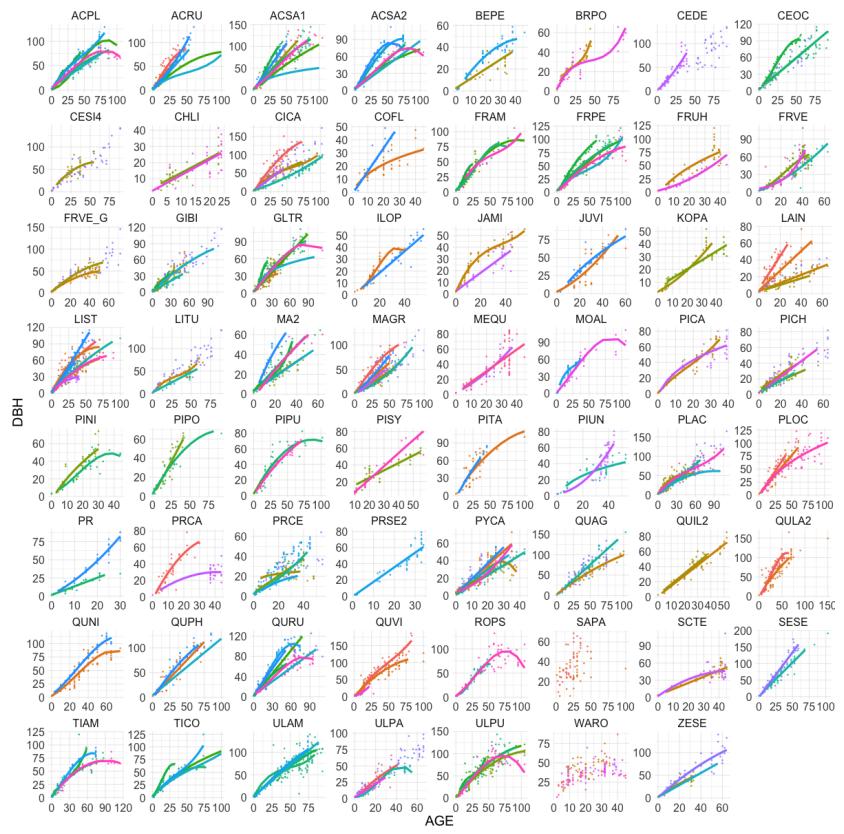
Joining, by = c("Region", "SpCode")
Joining, by = c("Region", "SpCode")
There were 50 or more warnings (use warnings() to see the first 50)

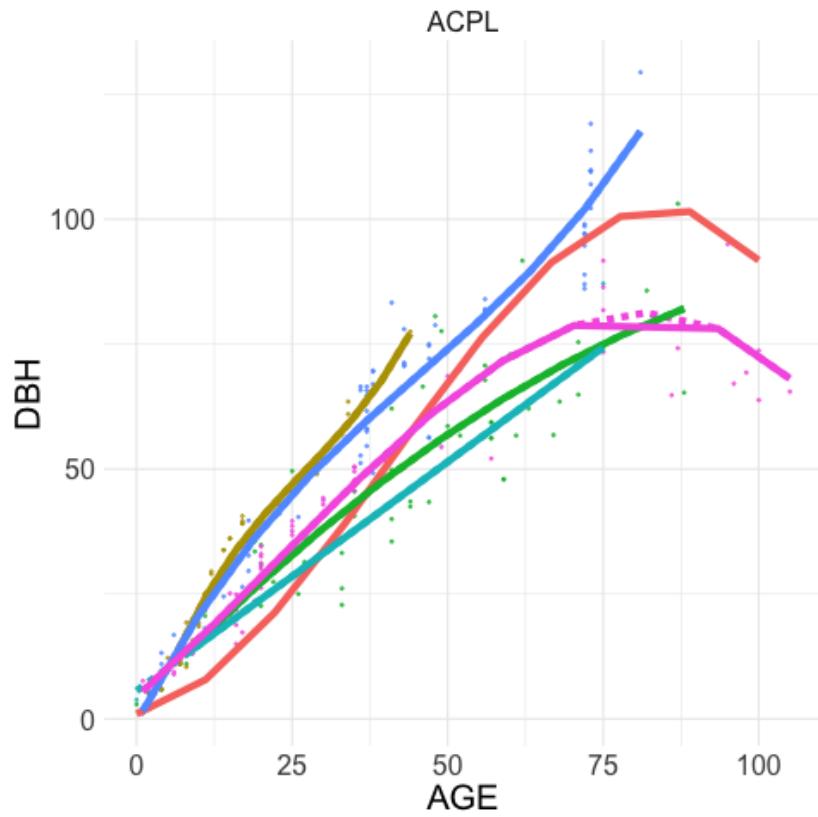
```



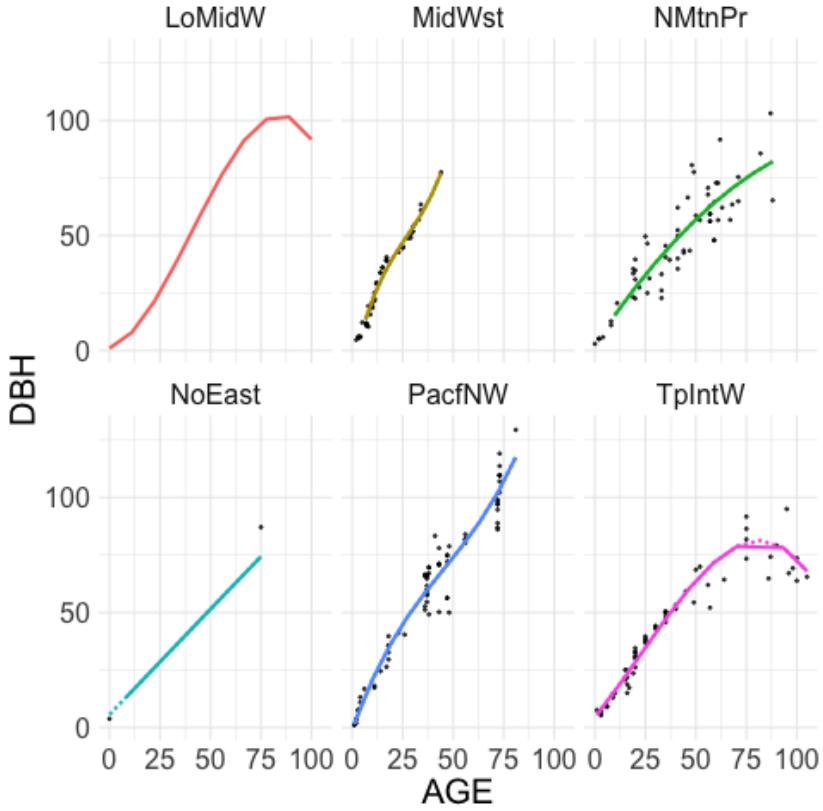








Funny thing about the Apps range is that for functions that eventually decrease, they don't properly stop inference. Look at the pink line above. The application range should be given for both the predictor and the response variable.



We should be able to borrow information from other regions to extend the applicable range for regions with smaller ranges.

There is an equation for the lower midwest, but no data?

```
ascii.nowarn.print(filter(eqn, SpCode == "ACPL"))

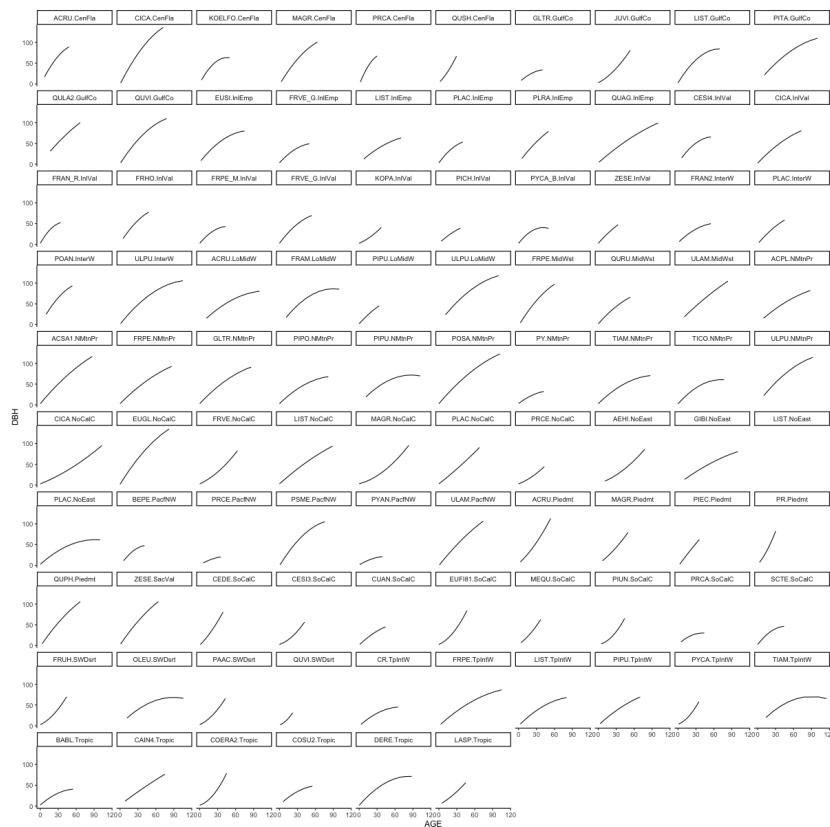
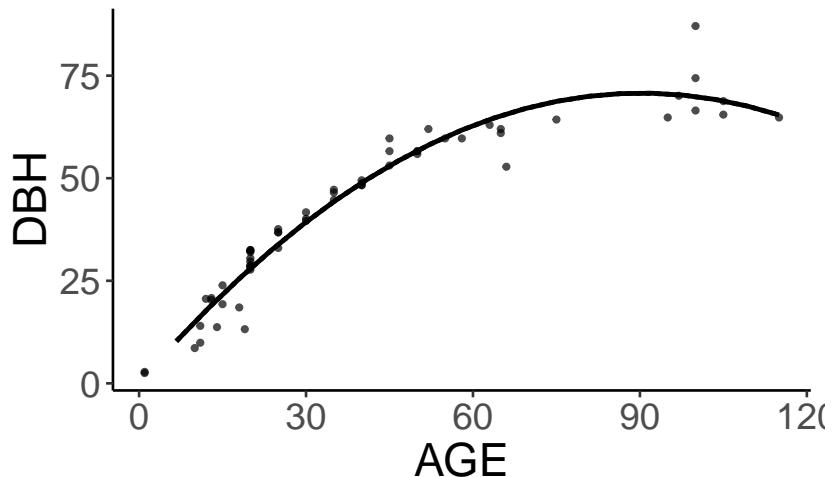
|   | Region | Scientific.Name | SpCode | Independent.variable | Predicts.component | Units
|---+-----+-----+-----+-----+-----+-----+
| 1 | LoMidW | Acer platanoides | ACPL | age | dbh | centi
| 2 | MidWst | Acer platanoides | ACPL | age | dbh | centi
| 3 | NMtnPr | Acer platanoides | ACPL | age | dbh | centi
| 4 | NoEast | Acer platanoides | ACPL | age | dbh | centi
| 5 | PacfNW | Acer platanoides | ACPL | age | dbh | centi
| 6 | TpIntW | Acer platanoides | ACPL | age | dbh | centi

ascii.nowarn.print(filter(d, SpCode == "ACPL", Region == "NoEast"))

|   | Region | City | TreeID | SpCode | DBH | Leaf | Age | AGE |
|---+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | NoEast | Queens, NY | 900243.00 | ACPL | 3.80 | 3.10 | 0.00 | 0.00 |
| 2 | NoEast | Queens, NY | 46681.00 | ACPL | 87.10 | 445.50 | 75.00 | 75.00 |
```

The reason that the equation for ACPL in the North East is linear is because there are only two observations. Yikes.

TIAM TpIntW



TIAM TpIntW

## 9.12 NEXT modelling

Just do the climate effects now  
Subset down to 5 species

```
sp.sub <- c("ACPL", "QURU", "PIST", "CEOCC", "FRPE")
ds <- filter(d, SpCode %in% sp.sub)
```

### 9.12.1 lme4

```
library(lme4)
```

```
ds <- mutate(ds, logDBH = log(DBH), logAGE = log(AGE + 1), AGE2 = AGE^2, AGE2scaled = AGE2/100)
#mod <- lmer(DBH ~ AGE + (1 + AGE | SpCode) + (1 + poly(AGE, 2) | SpCode)
#mod <- lmer(DBH ~ poly(AGE, 2) + (1 + poly(AGE, 2) | SpCode)
#mod <- lmer(DBH ~ poly(AGE, 2) + (1 + poly(AGE, 2) | Region)
#mod <- lmer(DBH ~ sqrt(AGE) + (1 + sqrt(AGE) | SpCode)
#mod <- lmer(DBH ~ AGE + (1 + AGE | SpCode) + (1 + poly(AGE, 2) | Region)
```

#no intercept. This is a good looking model.

```
mod <- lmer(DBH ~ AGE + AGE2scaled - 1 + (AGE + AGE2scaled - 1 | SpCode) + (AGE - 1 | Region))
```

```
summary(mod)
```

Linear mixed model fit by REML [lmerMod]

Formula:

```
DBH ~ AGE + AGE2scaled - 1 + (AGE + -1 | SpCode) + (AGE + AGE2scaled - 1 | Region)
```

Data: ds

REML criterion at convergence: 6096.9

Scaled residuals:

Min	1Q	Median	3Q	Max
-3.4742	-0.5288	-0.0049	0.5247	3.8558

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
Region	AGE	0.16326	0.40406	
	AGE2scaled	0.08519	0.29187	-0.93
SpCode	AGE	0.00158	0.03975	
Residual		91.47154	9.56408	

Number of obs: 822, groups: Region, 8; SpCode, 4

```

Fixed effects:
  Estimate Std. Error t value
AGE        1.8479    0.1511 12.233
AGE2scaled -0.8656    0.1199 -7.221

Correlation of Fixed Effects:
  AGE
AGE2scaled -0.895

dsAGE_min_max = ds %>%
  group_by(Region, SpCode) %>%
  summarize(minAGE = min(AGE, na.rm = T),
            maxAGE = max(AGE, na.rm = T)) %>%
  data.frame()

newdata <- lapply(1:nrow(dsAGE_min_max), function(i) {
  x <- seq(dsAGE_min_max$minAGE[i], dsAGE_min_max$maxAGE[i], (dsAGE_min_max$maxAGE[i] - dsAGE_min_max$minAGE[i]) / 10)
  cbind(dsAGE_min_max[i,], AGE = x)
})

newdata <- bind_rows(newdata)

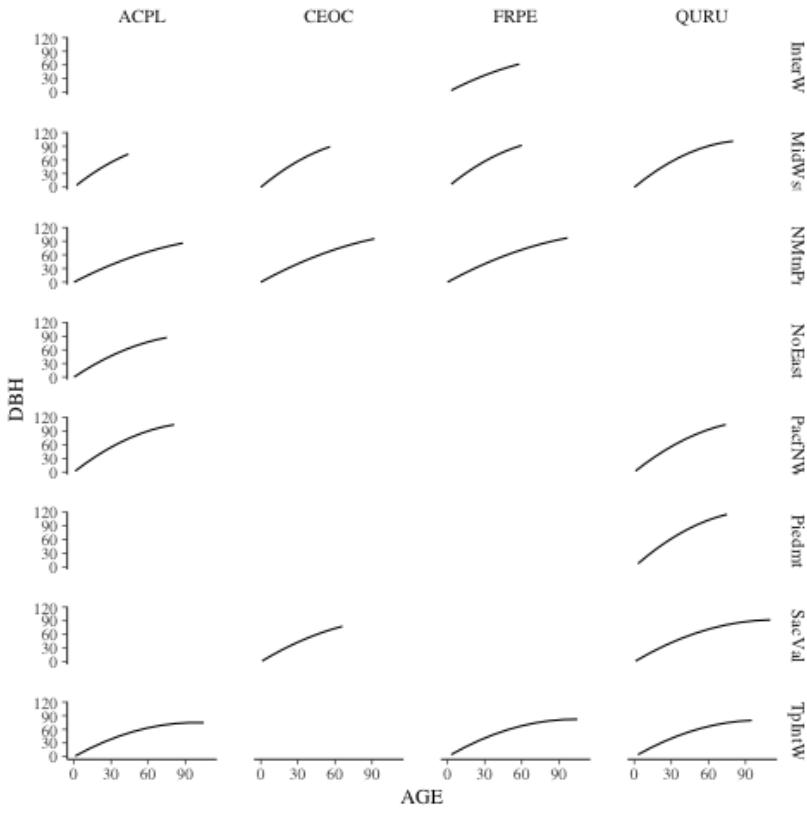
newdata <- mutate(newdata, logAGE = log(AGE + 1), AGE2scaled = AGE^2/100)

There were 17 warnings (use warnings() to see them)

pred <- predict(mod, newdata)

pred <- cbind(newdata, pred) %>%
  mutate(DBH = pred)

```



```

dsAGE_min_max = ds %>%
  group_by(SpCode) %>%
  summarize(minAGE = min(AGE, na.rm = T),
            maxAGE = max(AGE, na.rm = T)) %>%
  data.frame()

newdata <- lapply(1:nrow(dsAGE_min_max), function(i) {
  x <- seq(dsAGE_min_max$minAGE[i], dsAGE_min_max$maxAGE[i], (dsAGE_min_max$maxAGE[i] - dsAGE_min_max$minAGE[i]) / 10)
  cbind(dsAGE_min_max[i,], AGE = x)
})

newdata <- bind_rows(newdata)

newdata <- mutate(newdata, logAGE = log(AGE + 1), AGE2scaled = AGE^2/100)

regions <- rep(unique(ds$Region), each = nrow(newdata))

```

```

newdata <- do.call("rbind", replicate(length(unique(regions)), newdata, simplify = FALSE))

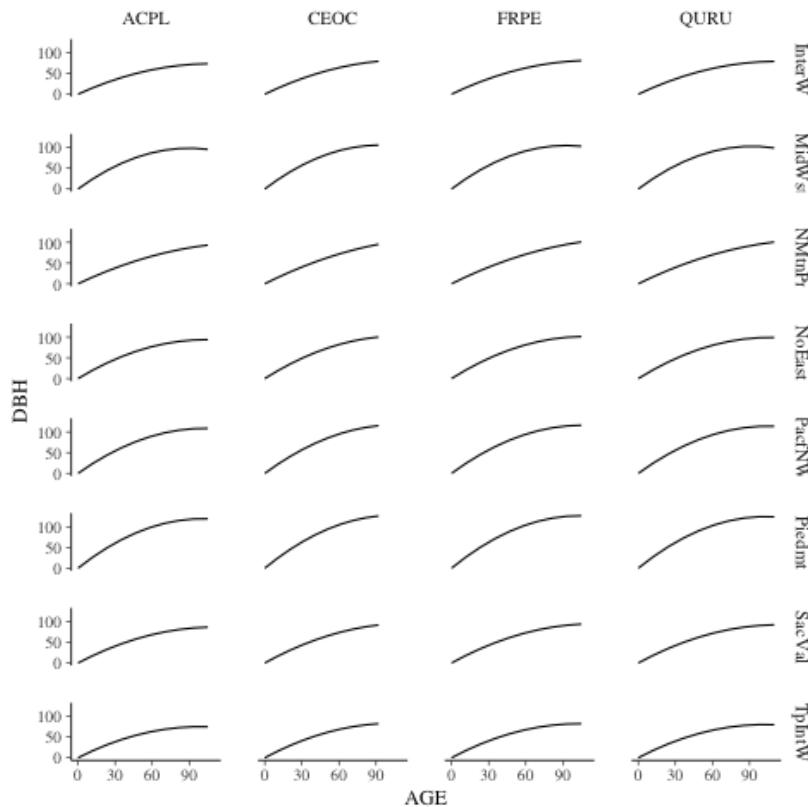
newdata$Region <- regions

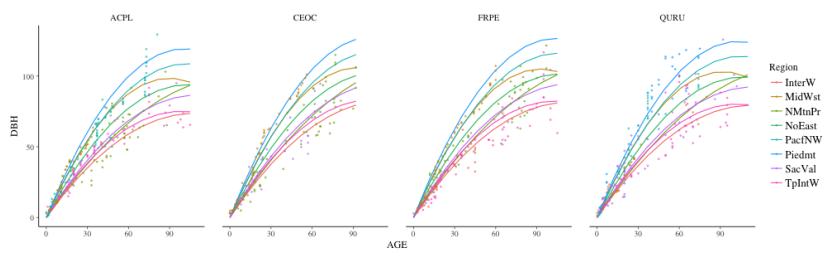
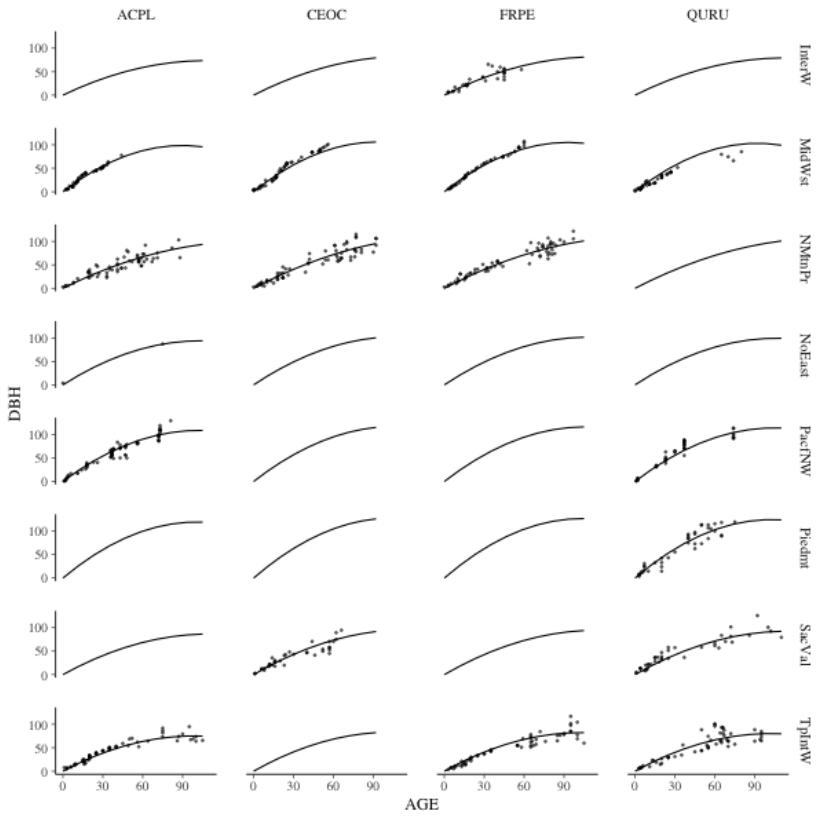
Warning messages:
1: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
2: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
3: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
4: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
5: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded

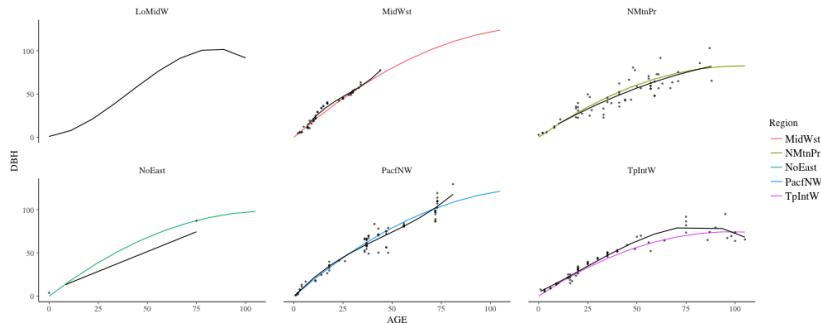
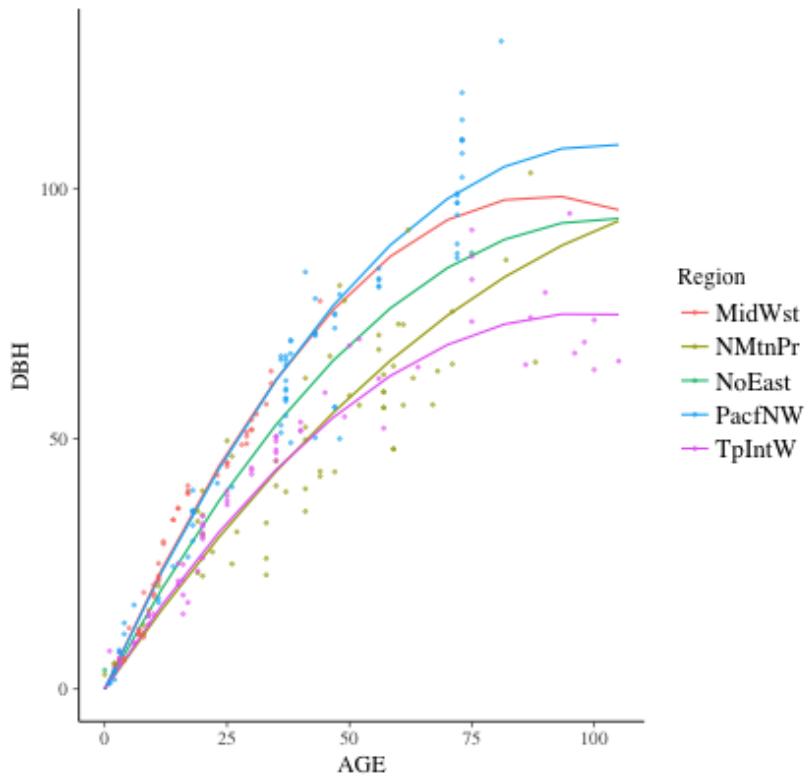
pred <- predict(mod, newdata)

pred <- cbind(newdata, pred) %>%
  mutate(DBH = pred)

```







```
filter(d, SpCode == "ACPL") %>% group_by(Region) %>% summarize(n = n())

# A tibble: 5 x 2
  Region     n
  <chr> <int>
1 MidWst    48
2 NMtnPr    60
3 NoEast     2
4 PacfNW    74
5 TpIntW   62
```

```
ascii.nowarn.print(filter(eqn, SpCode == "ACPL"))
```

Region	SpCode	Independent.variable	Predicts.component	Model.weight	EqName	a	b
LoMidW	ACPL	age	dbh	1/age	cub	0.98	0.25
MidWst	ACPL	age	dbh	1	cub	-7.95	3.81
NMtnPr	ACPL	age	dbh	1/age <sup>2</sup>	quad	2.84	1.32
NoEast	ACPL	age	dbh	1	lin	5.62	0.92
PacfNW	ACPL	age	dbh	1/age	cub	-0.85	2.44
TpIntW	ACPL	age	dbh	1/age	cub	4.42	1.08

There are 5 regions with data for dbh and age of ACPL. the North east has two data points. Even though there are no data for the LoMidW region, there is an equation for it with a reported n of 13. Where does this equation come from? The n for the north east is reportedly 48, but is really 2.

1. What about adding climate?

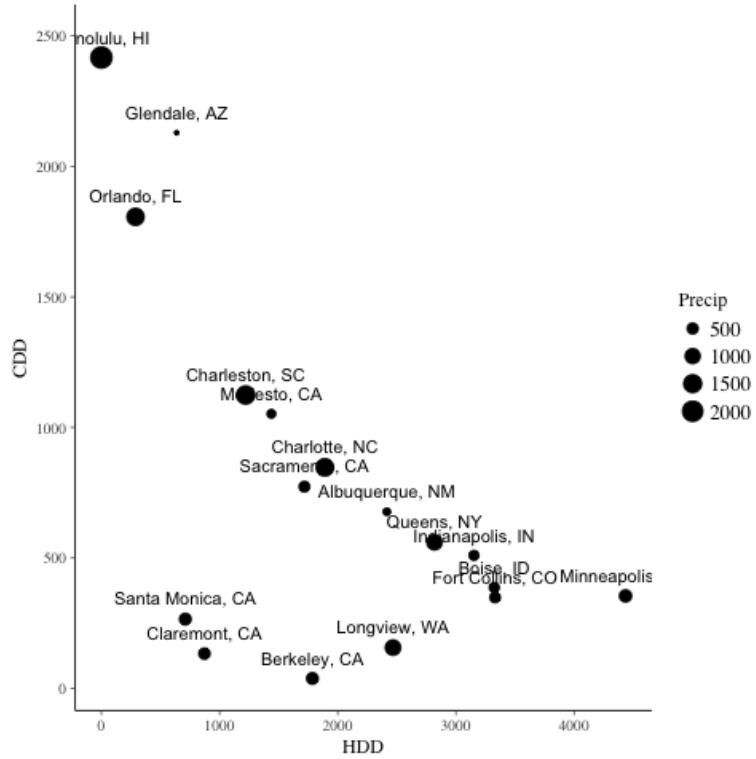
```
sp.sub <- c("ACPL", "QURU", "PIST", "CEO", "FRPE")
ds <- filter(d, SpCode %in% sp.sub)

ds <- mutate(ds, logDBH = log(DBH), logAGE = log(AGE + 1), AGE2 = AGE^2, AGE2scaled = AGE2 / 100)
```

from table 1 see above

```
str(climate)

'data.frame': 17 obs. of  5 variables:
 $ Region: chr  "CenFla" "GulfCo" "InlEmp" "InlVal" ...
 $ City   : chr  "Orlando, FL" "Charleston, SC" "Claremont, CA" "Modesto, CA" ...
 $ CDD    : int  1806 1124 134 1052 773 677 510 355 349 39 ...
 $ HDD    : int  289 1221 872 1439 1718 2416 3153 4436 3332 1786 ...
 $ Precip: int  1367 1555 523 315 470 250 392 622 452 564 ...
```



Rescale precip, HDD, and CDD by dividing by 100

```
climate <- climate %>% mutate(HDD = HDD/100, CDD = CDD/100, Precip = Precip/100)
ds <- left_join(ds, climate)
```

```
Joining, by = c("Region", "City")
```

Just precip for now

```
mod_precip <- lmer(DBH ~ AGE + AGE2scaled - 1 + Precip + Precip:AGE +(AGE + AGE2scaled

dsAGE_min_max = ds %>%
  group_by(SpCode) %>%
  summarize(minAGE = min(AGE, na.rm = T),
            maxAGE = max(AGE, na.rm = T)) %>%
  data.frame()

newdata <- lapply(1:nrow(dsAGE_min_max), function(i) {
  x <- seq(dsAGE_min_max$minAGE[i], dsAGE_min_max$maxAGE[i], (dsAGE_min_max$maxAGE[i] -
    dsAGE_min_max$minAGE[i])/100)
  cbind(dsAGE_min_max[i,], AGE = x)
```

```
})

newdata <- bind_rows(newdata)

newdata <- mutate(newdata, logAGE = log(AGE + 1), AGE2scaled = AGE^2/100)

regions <- rep(c("wet","dry"), each = nrow(newdata))
newprecip <- rep(c(2.5,14), each = nrow(newdata))

newdata <- do.call("rbind", replicate(length(unique(regions)), newdata, simplify = FALSE))

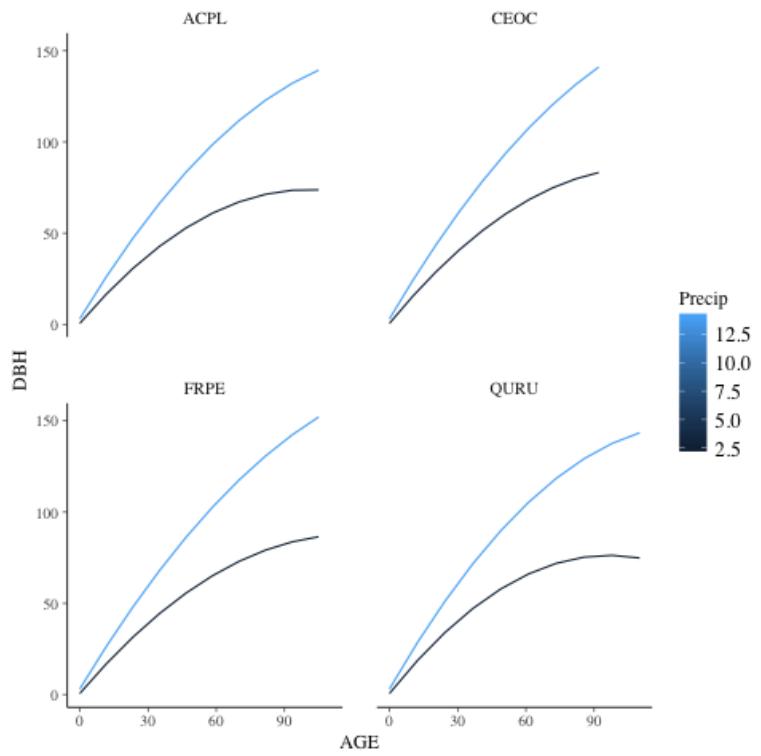
newdata$Region <- regions
newdata$Precip <- newprecip
```

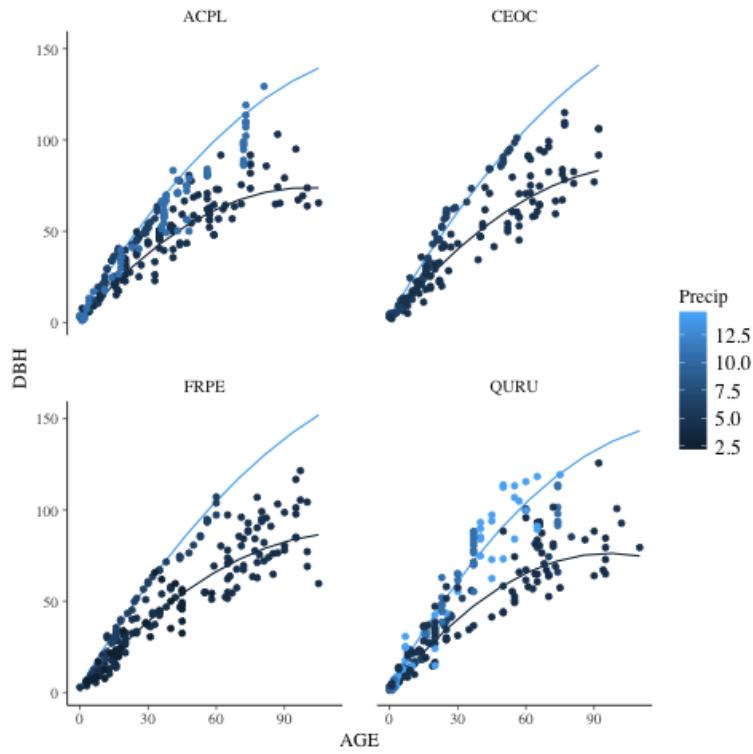
Warning messages:

```
1: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
2: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
3: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
4: In data.frame(..., check.names = FALSE) :
  row names were found from a short variable and have been discarded
```

```
pred <- predict(mod_precip, newdata, allow.new.levels = T)

pred <- cbind(newdata, pred) %>%
  mutate(DBH = pred)
```





```
ggplot() +
```

## 10 misc

### 10.1 Meeting with Jun [2018-08-27 Mon]

What should the structure of the paper be?

- Premise of paper: showing a new method to fit multilevel non-linear growth models for trees
- Our working example is the urban tree
- Show different types of posterior predictions (e.g. new tree in a observed city of an observed species, versus new tree in an unobserved city of an unobserved species, but known climate and genus)
- Show predictions compared to the existing method.
  - will show better potential to extrapolate, both to unobserved ages and unobserved species/city combinations.

- Model selection
- posterior predictive checks