

Multilevel Urban Tree Allometric equations

erker

October 16, 2018

Methods

Data

The urban tree database (UTD) 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 by city combinations 1).

Some species were measured in multiple cities, but not most. The number of trees of each city by species combination sampled ranged from 1 (both *Liquidambar styraciflu* and *Prunus serrulata* in Queens, NY) to 79 (*Quercus laurifolia* in Charleston, SC). The median number of trees in each species-city combination was 37.

Age is defined in this dataset as time since planting, since this is the record kept by cities. Actual age of the trees may be several years more. Diameter (cm) of the trees is measured at breast height (1.37m above ground).

In the UTD, trees are classified taxonomically down to cultivar for some individuals, but here we aggregate cultivars up to the species level. Species are then nested within Genera.

The 17 cities in the UTD cover much of the US geographically, 2, and much of the variation in climate, 3.



Figure 1: Number of trees sampled of each species and city combination in the urban tree database.

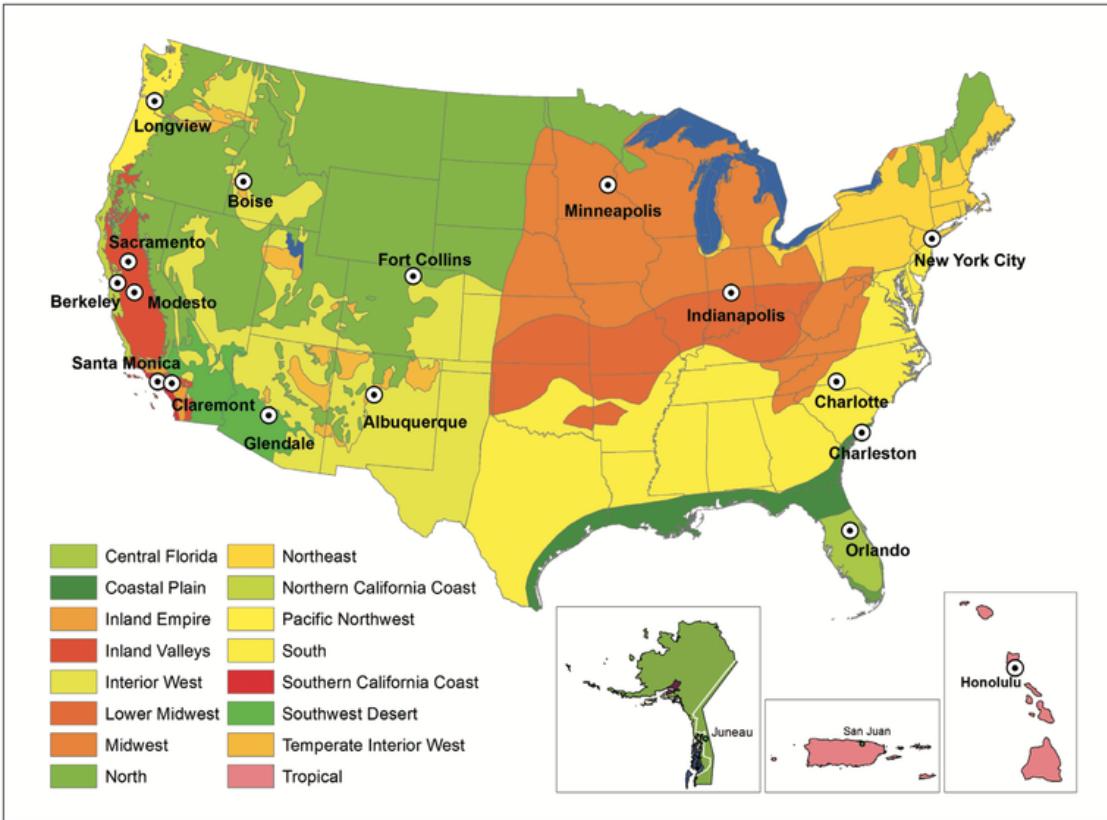


Figure 9—Climate zones were aggregated from 45 Sunset climate zones into 16 zones. Each zone has a reference city where tree growth data were collected. Sacramento, California, was added as a second reference city (with Modesto) to the Inland Valleys zone.

Figure 2: 16 climate regions and 17 representative cities in the UTD (?).

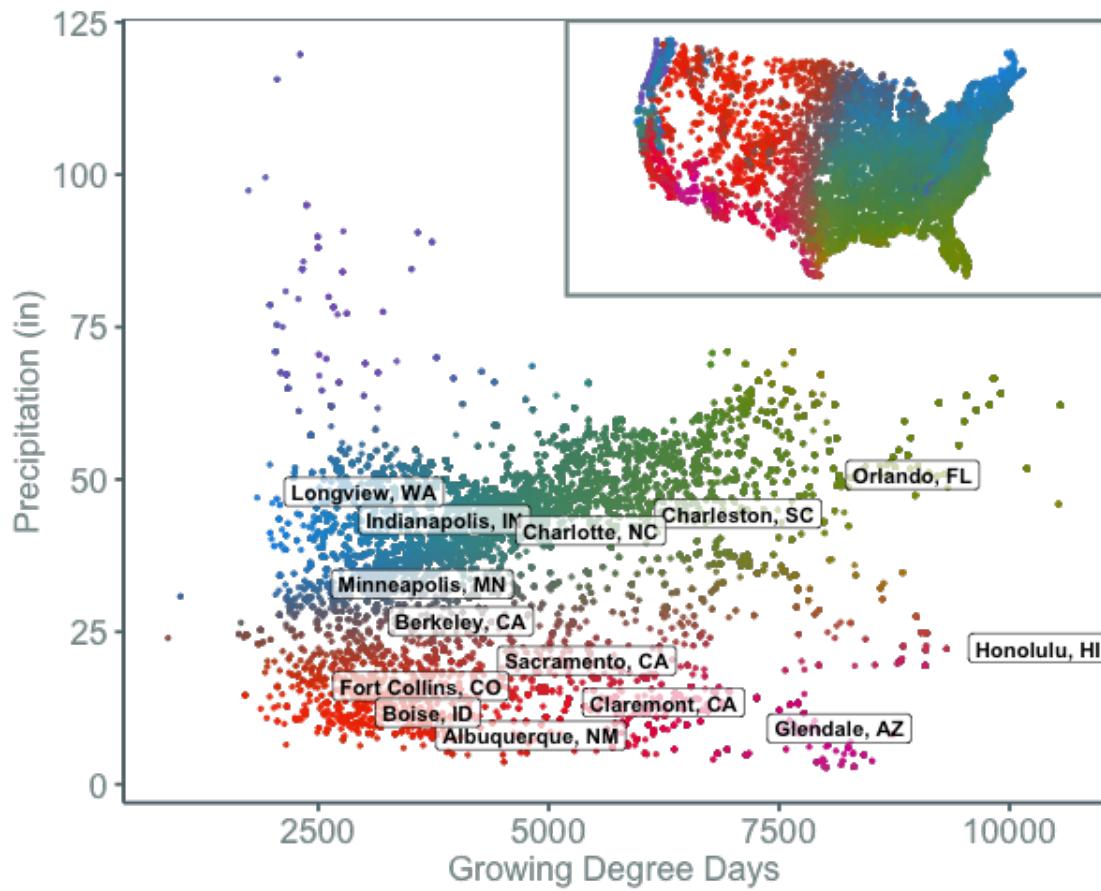


Figure 3: US census tract centroids with UTD reference cities overlaid in growing degree day (GDD) and precipitation climate space and matching color gradient in geographic space. The reference cities cover climate space well, and variation in precipitation and growing degree days is continuous.

Rather than using the aggregated sunset zones as done in UTD, we used growing degree days (GDD) and precipitation data from climate NOAA's climate normals. Figure 3 shows each census tract centroid in the conterminous US plotted in GDD and precipitation space. We approximated the GDD and precipitation for each tract by assigning the values of the weather station closest to the centroid. This allows us to vary our model continuously across geographic space in a way that better captures the natural gradients of climate.

Tidy data for this analysis

- load libraries
- functions

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

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

}
```

- 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" ...
$ 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" "Cinn ...
$ CommonName    : chr  "Silver maple" "European white birch" "Chinese hackberry" " ...
$ TreeType       : chr  "BDL" "BDM" "BDL" "BEM" ...
$ address        : int  -1 -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 -1 ...
$ cell            : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ OnStreet       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ FromStreet     : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ ToStreet        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ Age             : int  0 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 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 -1 ...
$ CrnHt (m)      : num  0.5 0.8 0.6 0.9 0.9 0.8 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 -1 -1 ...
$ TreeOr       : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ CarShade     : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ LandUse      : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ Shape        : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ WireConf     : int -1 -1 -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 2.5 ...
$ dbh2         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh3         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh4         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh5         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh6         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh7         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh8         : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
- attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame': 24255 obs. of 5 v
..$ 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/
- 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"

```

- explanation of variables from metadata

DbaseID = Unique id number for each tree.

Region = 16 U.S. climate regions, abbreviations are used (see 1_{Regionalinformation.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 2_{Regionalspeciesandcounts.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 (m²) = Estimated using digital imaging method to nearest 0.1 squared meter (m²).

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.

- fix some species things

- change lower case species codes

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

- change QUAG1 to be QUAG

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

- 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"

```

- change scientific names (remove cultivated variety)

```

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

```

- tidy a few names and select variables of interest here

```

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

```

- Remove missing data (either DBH or Age)

```

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

```

- save data

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

- 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 ...
$ Leaf         : num  2.5 1.9 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" "Cinn...
- attr(*, "problems")=Classes 'tbl_df', 'tbl' and 'data.frame': 24255 obs. of  5 v...
..$ 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...
- 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"

```

- 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", "A")

d2 <- d %>%
  mutate(Generus = 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 = c("Region", "City", "Species", "DBH", "AGE", "ScientificName")
Error in mutate_impl(.data, dots) :
  Evaluation error: object 'Precip' not found.

Error in saveRDS(d2, "../data/age_dbh_full.rds") : object 'd2' not found
Error in eval(lhs, parent, parent) : object 'd2' not found
Error in saveRDS(d3, "../data/age_dbh_testing.rds") :
  object 'd3' not found

```

- 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/
```

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	Calophyllum inophyllum	
19	Carpinus betulus 'Fastigiata'	
20	Carya illinoinensis	
21	Cassia x nealiae	
22	Casuarina equisetifolia	
23	Catalpa speciosa	
24	Cedrus deodara	
25	Celtis laevigata	
26	Celtis occidentalis	
27	Celtis sinensis	
28	Ceratonia siliqua	
29	Chilopsis linearis	
30	Cinnamomum camphora	
31	Citharexylum spinosum	
32	Cocos nucifera	
33	Conocarpus erectus var. argenteus	
34	Cordia subcordata	
35	Cornus florida	
36	Crataegus sp.	
37	Crataegus x lavallei	

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 'Raywood'</i>	
54	<i>Fraxinus excelsior 'Hessei'</i>	
55	<i>Fraxinus holotricha</i>	
56	<i>Fraxinus latifolia</i>	
57	<i>Fraxinus pennsylvanica</i>	
58	<i>Fraxinus pennsylvanica 'Marshall'</i>	
59	<i>Fraxinus uhdei</i>	
60	<i>Fraxinus velutina</i>	
61	<i>Fraxinus velutina 'Modesto'</i>	
62	<i>Ginkgo biloba</i>	
63	<i>Gleditsia triacanthos</i>	
64	<i>Gymnocladus dioicus</i>	

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	Ulmus parvifolia	
157	Ulmus pumila	
158	Veitchia merrillii	
159	Washingtonia filifera	
160	Washingtonia robusta	
161	Zelkova serrata	

161 Species

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	-122.9382	46.1382
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

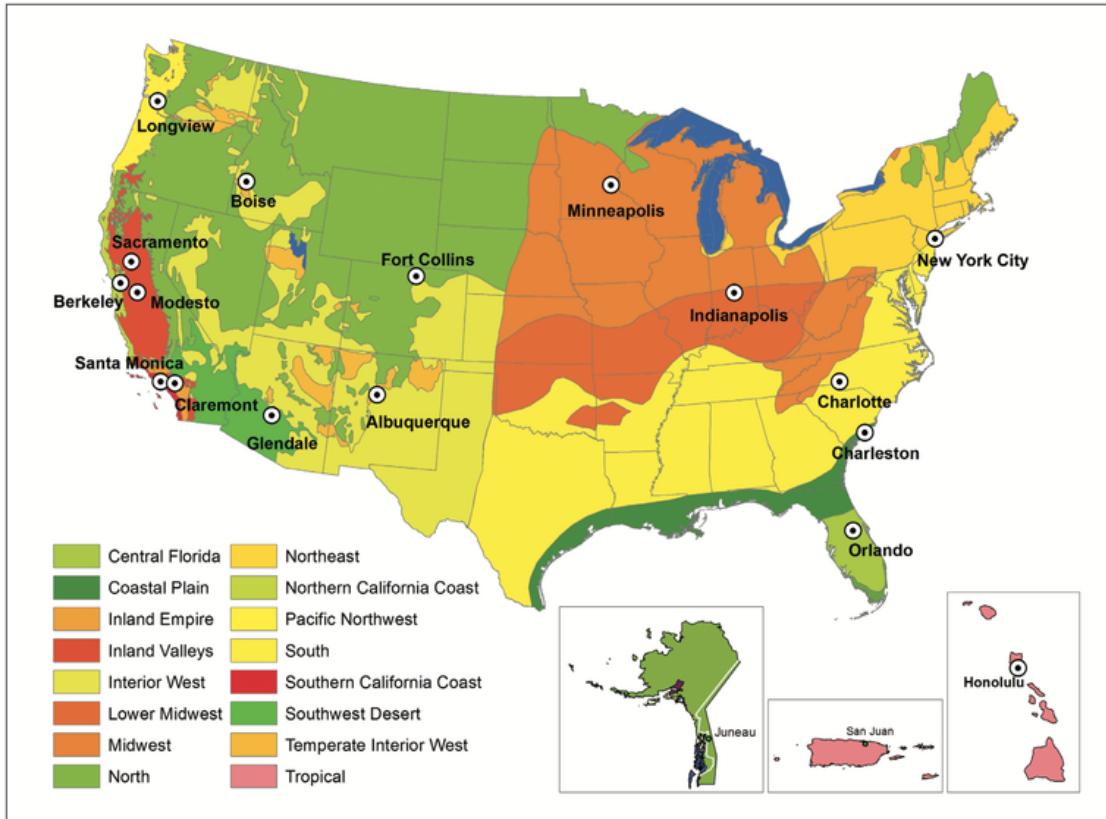


Figure 9—Climate zones were aggregated from 45 Sunset climate zones into 16 zones. Each zone has a reference city where tree growth data were collected. Sacramento, California, was added as a second reference city (with Modesto) to the Inland Valleys zone.

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. Fort Collins equations are meant to apply to the northern parts of maine... definitely space for improvement here (though very few live there so maybe doesn't matter so much). 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.

We also have unequal observations across cities. NY has very few.

Get new climate data. Growing degree days and Precip. Make plots

Plot, census tracts in GDD and Precip Space. An inset of the US continental with the color scheme.

overlay the reference cities on this plot

units of precip are 100ths of inches

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

library(ggplot2)
library(plyr)
library(dplyr)
library(tidyr)
library(stringr)

d <- read.table("../data/gdd.txt", stringsAsFactors = F)

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

d <- d %>%
  mutate(grdd = 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)) %>%
  filter(complete.cases(.))

dl <- left_join(d, ll)

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

p <- read.table("../data/precip.txt", stringsAsFactors = F)
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) %>%
  filter(complete.cases(.))

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

```

	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"

```

	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"

:var cityclimate=cityclimate

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

city_climate <- read.csv("../data/city_climate.csv")

cities <- city_climate %>%
  dplyr::select(City, Latitude, Longitude)

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])

```

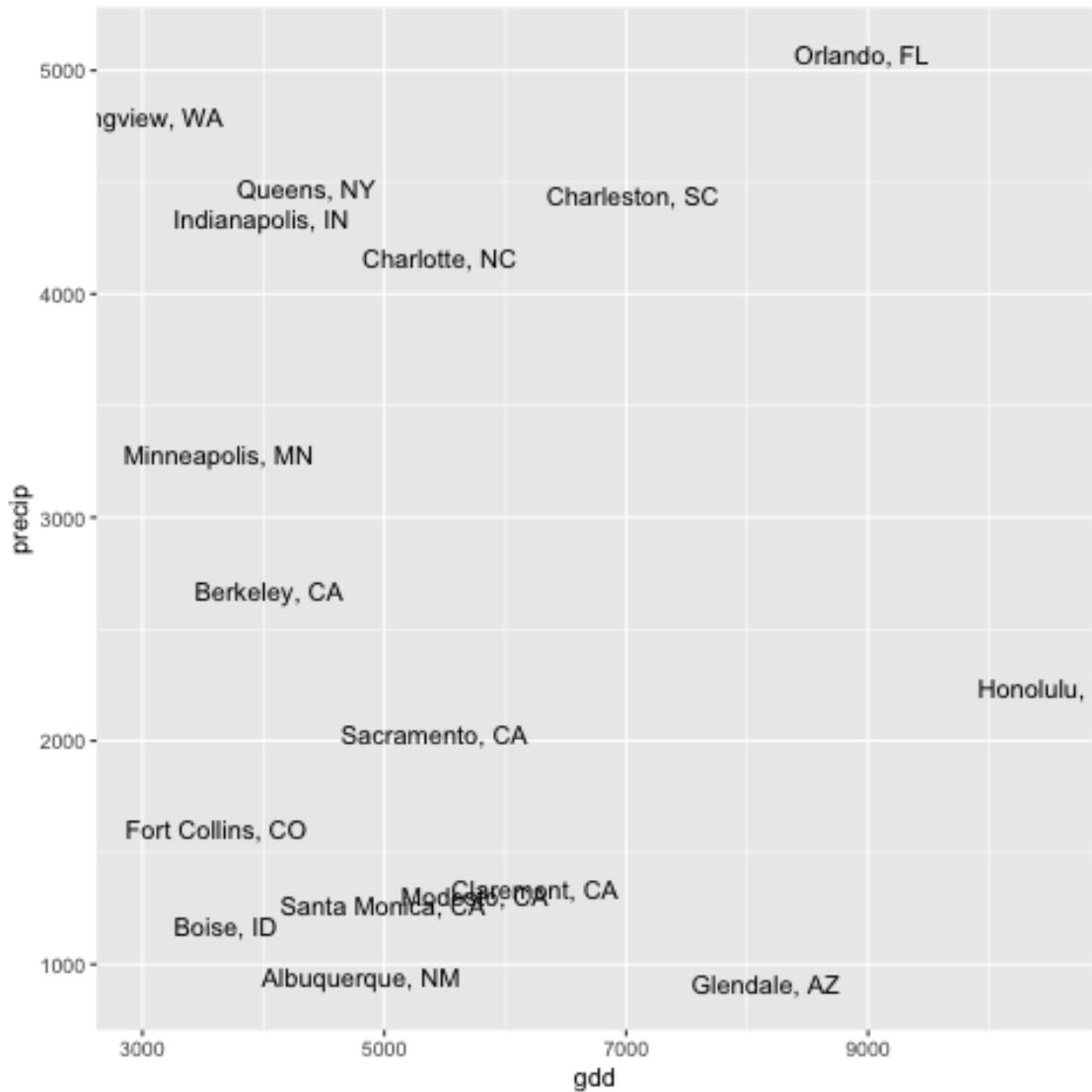
```
gdd <- gdd[gdd.dists.min,]

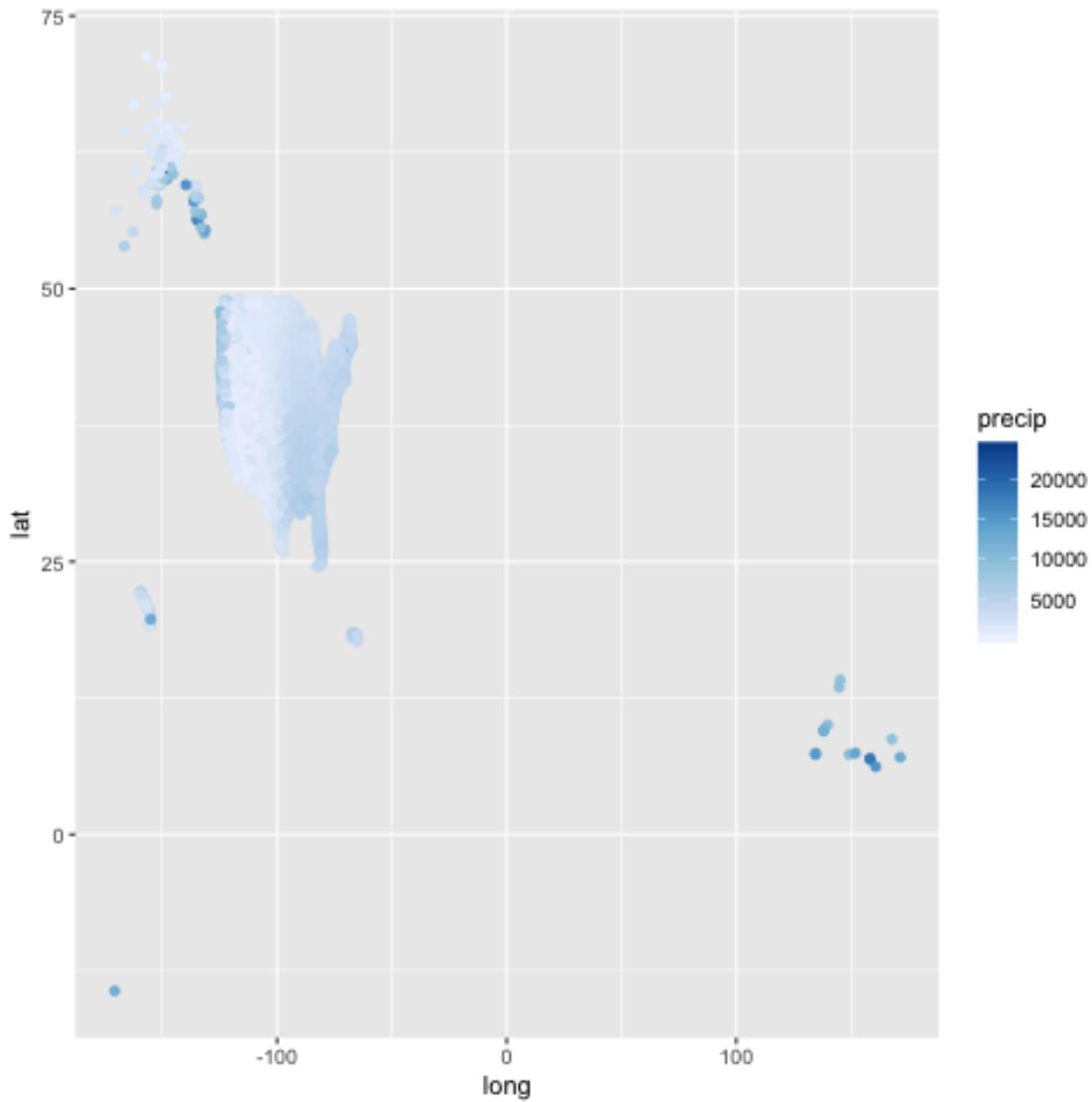
precip <- read.csv("../data/precip_qt_ll.csv", stringsAsFactors =F)
coordinates(precip) <- ~long + lat
proj4string(precip) <- CRS("+init=epsg:4326")

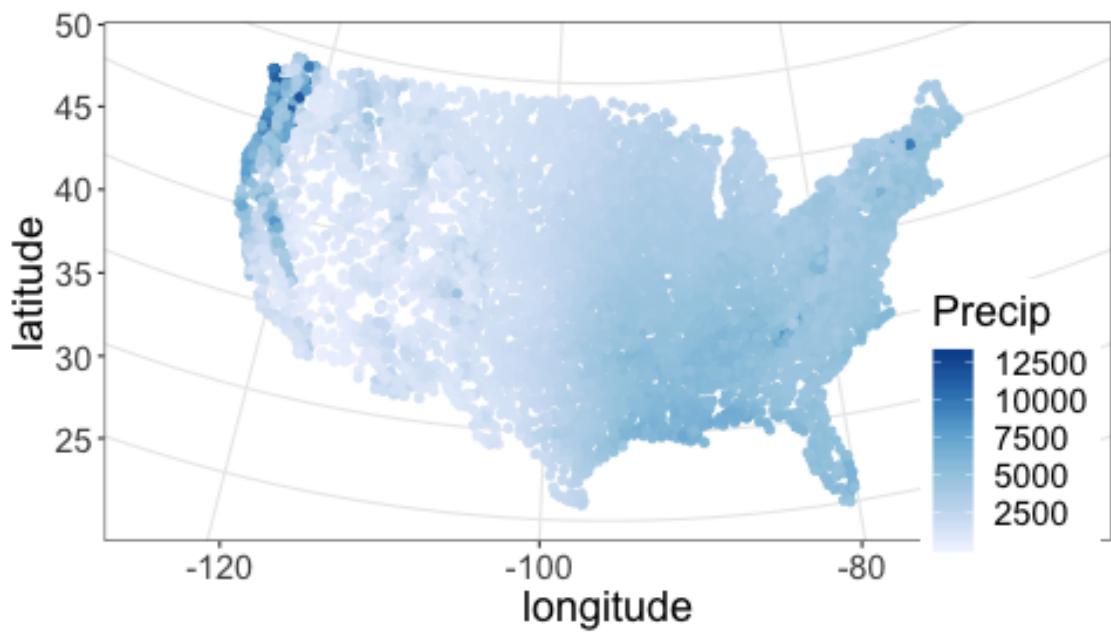
precip.dists <- spDists(cities, precip, longlat = T)
precip.dists.min <- apply(precip.dists, 1, function(x) which(x == min(x))[1])
precip <- precip[precip.dists.min,]

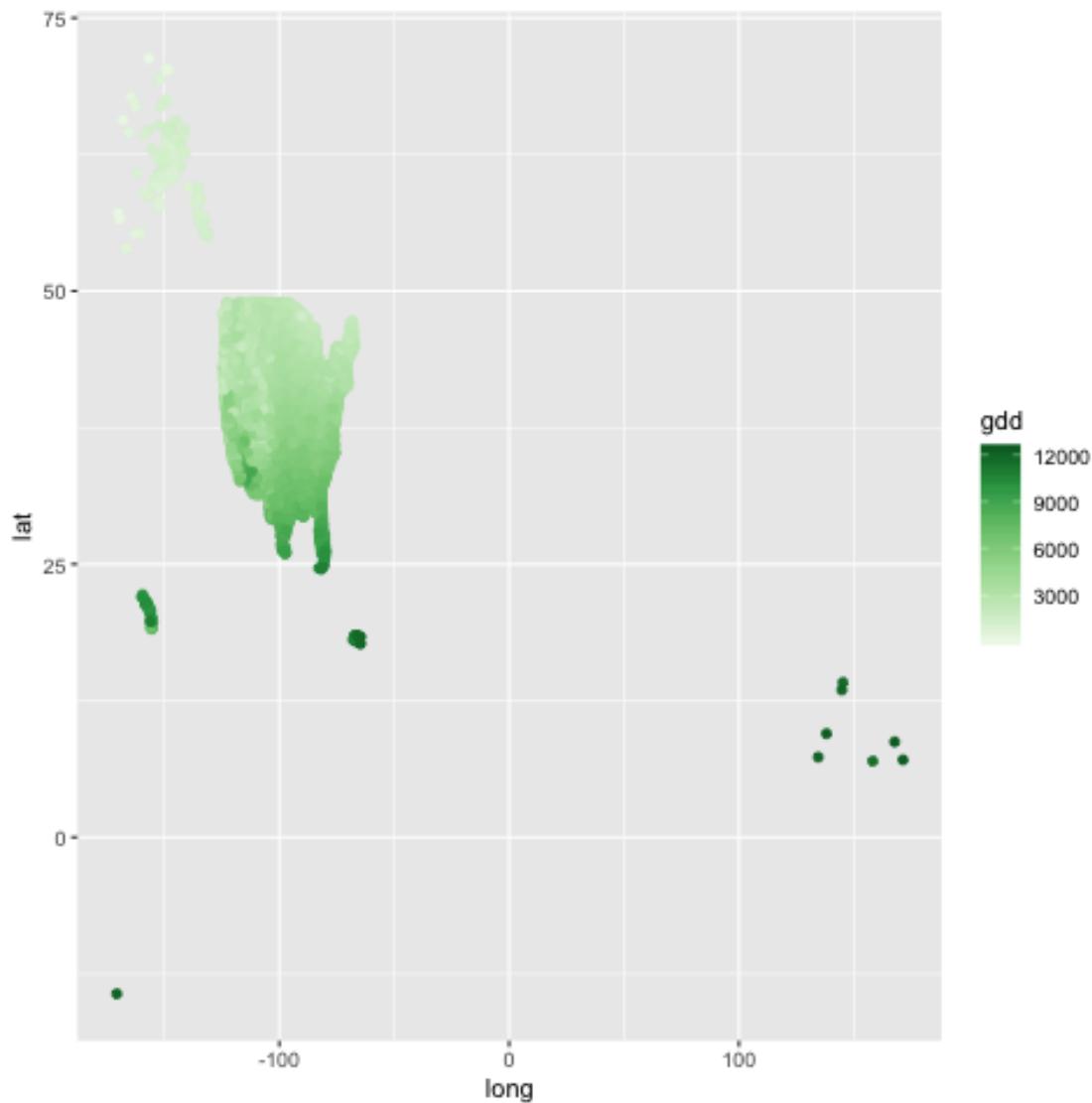
d <- bind_cols(precip@data, gdd@data,cities@data, data.frame(coordinates(cities))) %
  dplyr::select(gdd, precip, City, Longitude, Latitude)

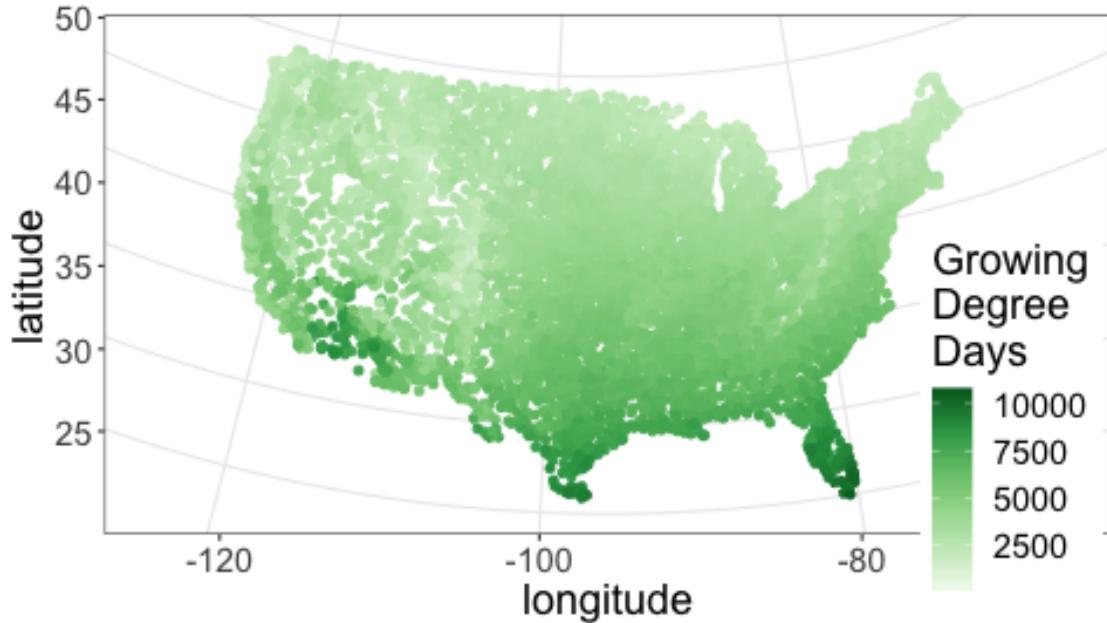
write.csv(d, "../data/cities_gdd_precip.csv")
```











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/
```

```
unzip census_centroid_pop.zip -d ../data/census_centroid_pop/
```

```
wget -O ../data/census_centroid_pop/metadata.xml http://faculty.baruch.cuny.edu/geoporta
```

```
rm census_centroid_pop.zip
```

```
library(sp)
```

```
library(raster)
```

```

library(dplyr)
library(tidyr)
library(ggplot2)
library(reshape2)

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 and precip

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,]
})


```

```

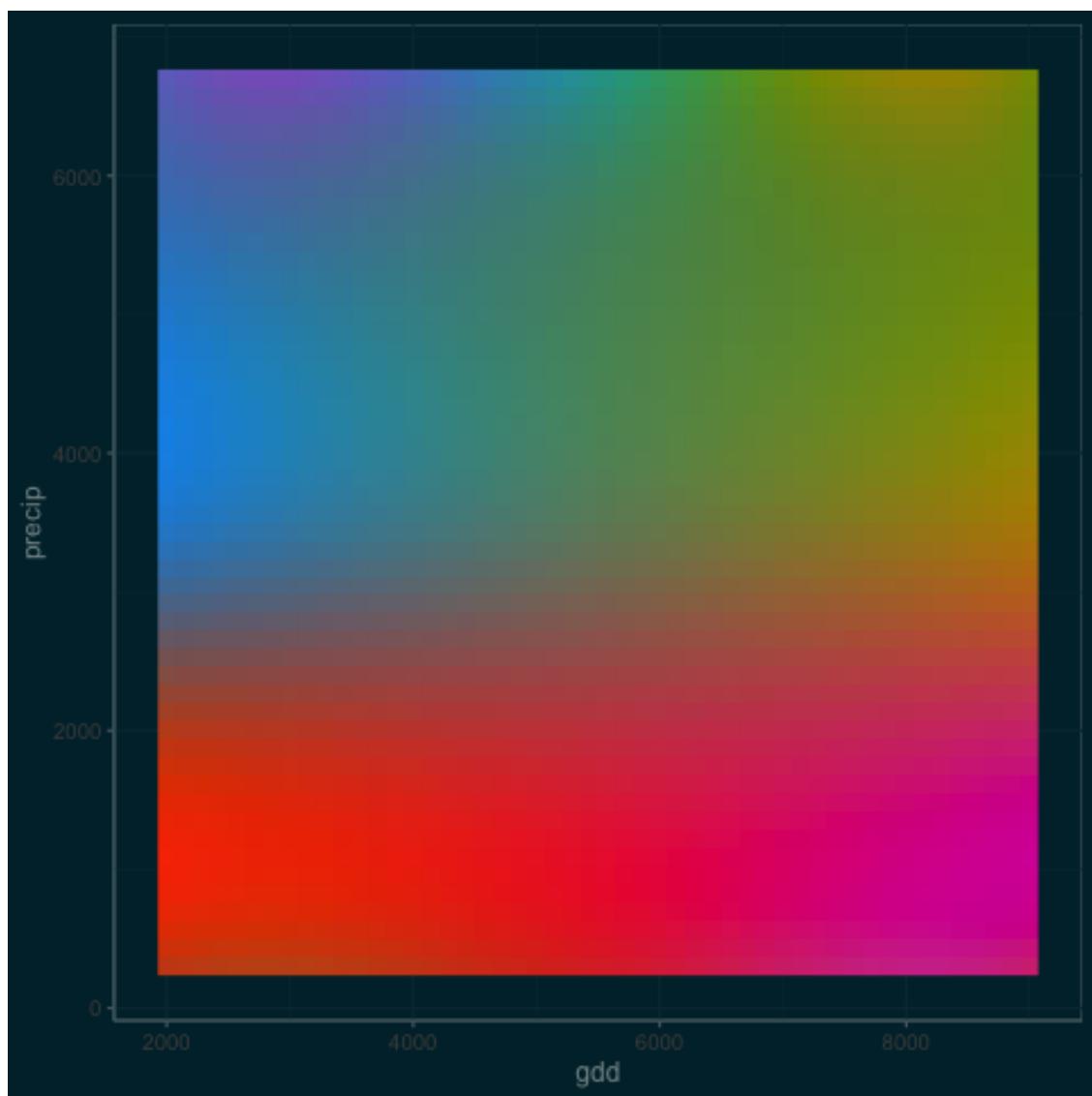
dsp.trks <- do.call("rbind", out)

Joining, by = c("station", "qual", "lat", "long")

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

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

```



```
library(sp)
```

```

library(reshape2)
library(dplyr)
library(tidyr)
library(ggplot2)

d <- read.csv("../data/censustractcentroids_gdd_precip_lat_long.csv")
dn <- d %>%
  rename(lat = LATITUDE, long = LONGITUDE, pop = POPULATION) %>%
  filter(complete.cases(.)) %>%
  filter( lat < 50, lat > 25, long < 0)

m <- as.matrix(dplyr::select(dn, gdd, precip))
cm <- as.matrix(dplyr::select(cols, gdd_col, precip_col))

whichmin <- apply(m, 1, function(mm) {
  cm[which.min(colSums((t(cm) - mm)^2)),]
})

precip_gdd_closest <- t(whichmin) %>%
  data.frame()

precip_gdd_closest_value <- left_join(precip_gdd_closest, cols) %>%
  rename(gdd = gdd_col, precip = precip_col) %>%
  dplyr::select(value)

dim(dn)

```

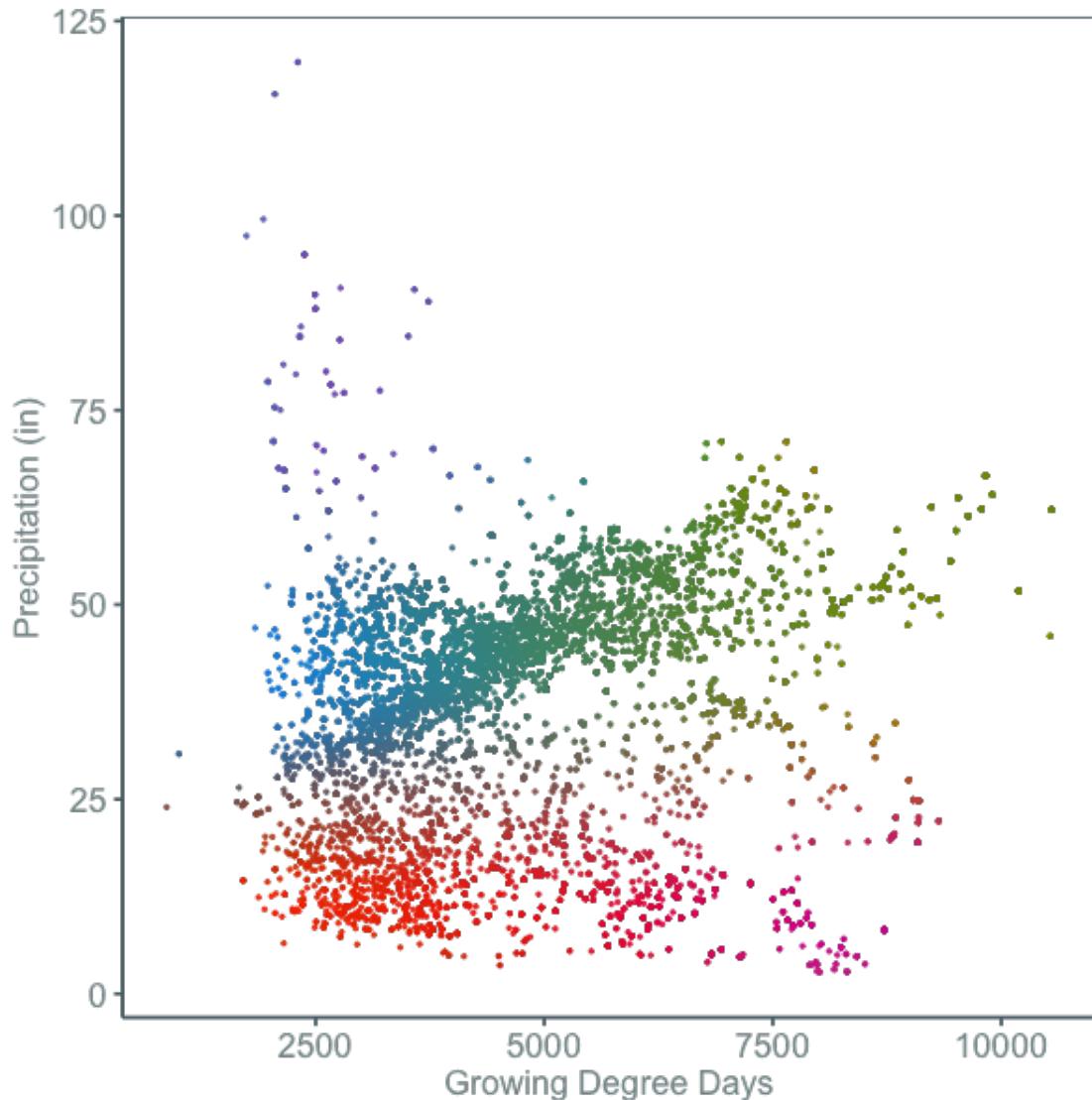
```
dim(precip_gdd_closest)
```

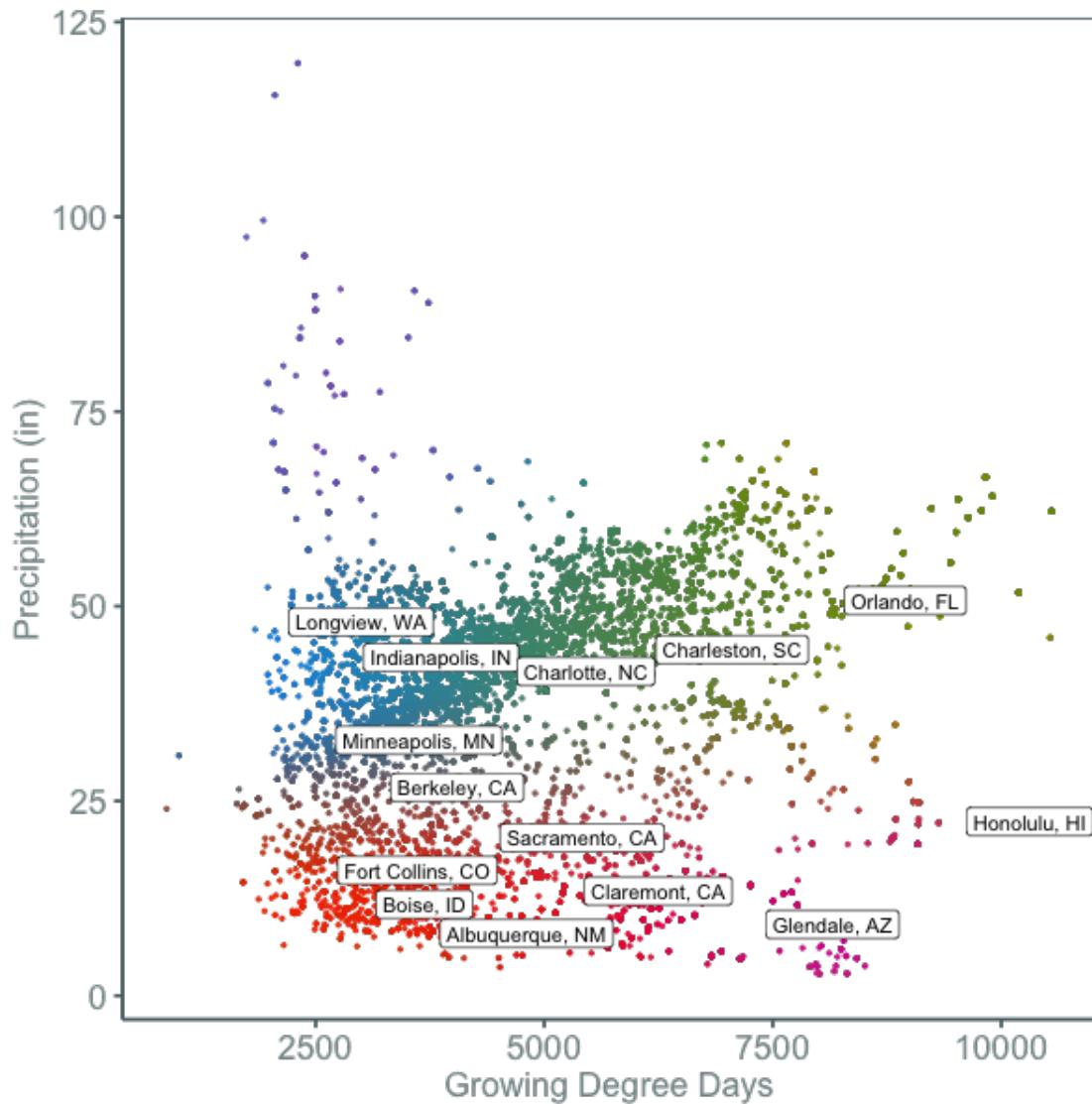
```
ddn <- cbind(dn, precip_gdd_closest_value)
```

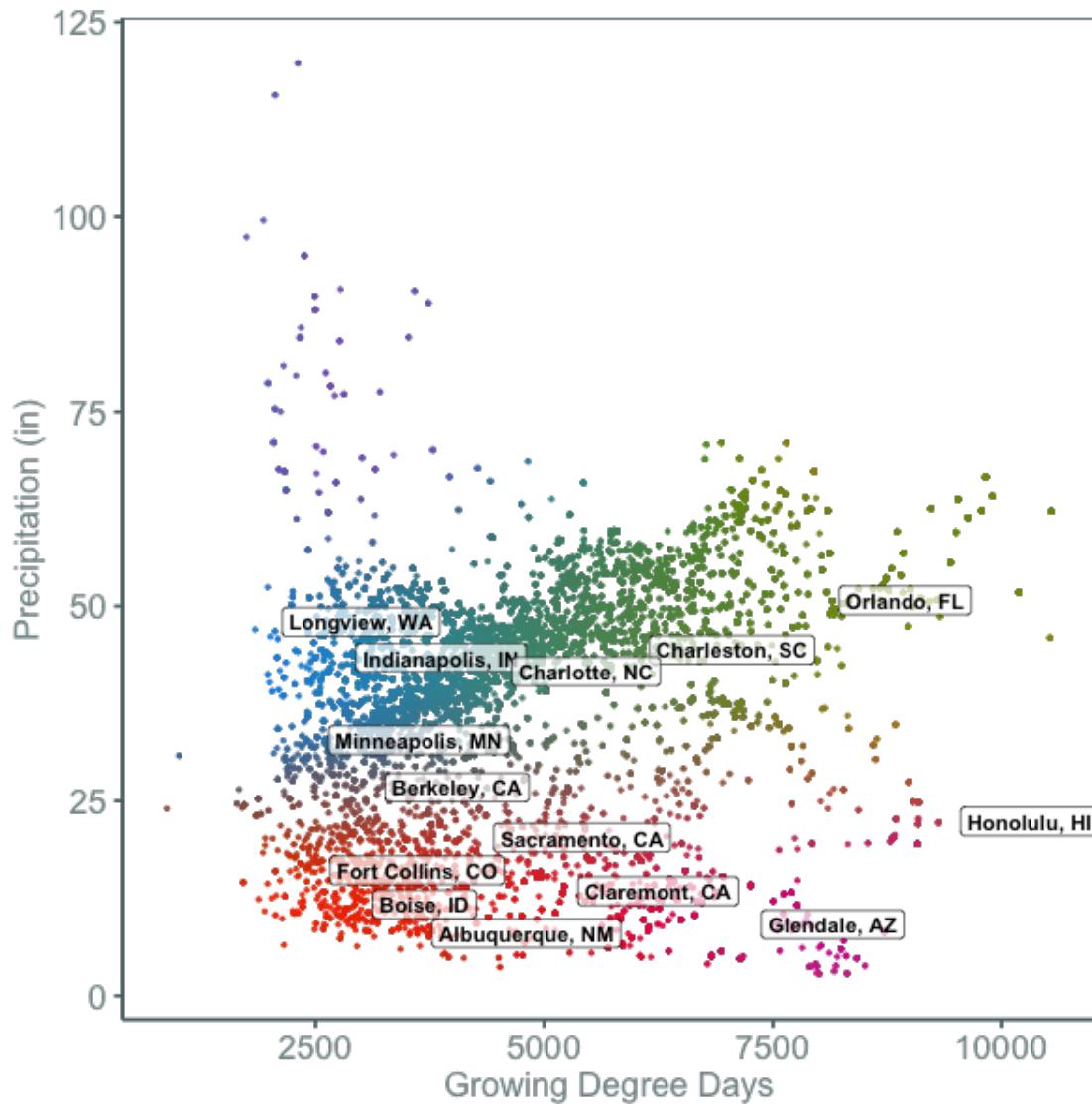
```
Joining, by = c("gdd_col", "precip_col")
```

```
[1] 38487    7
```

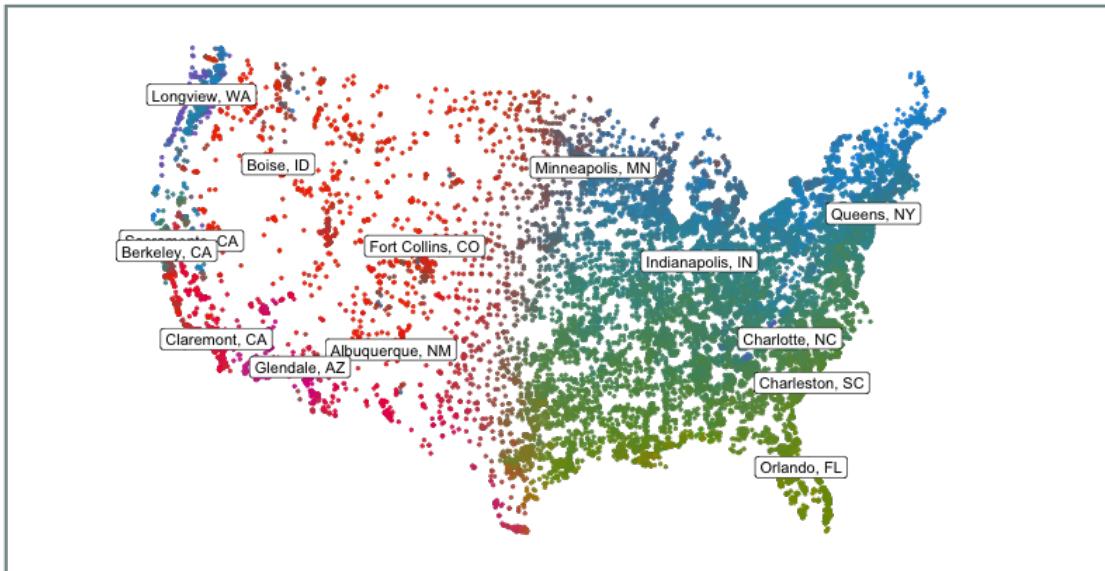
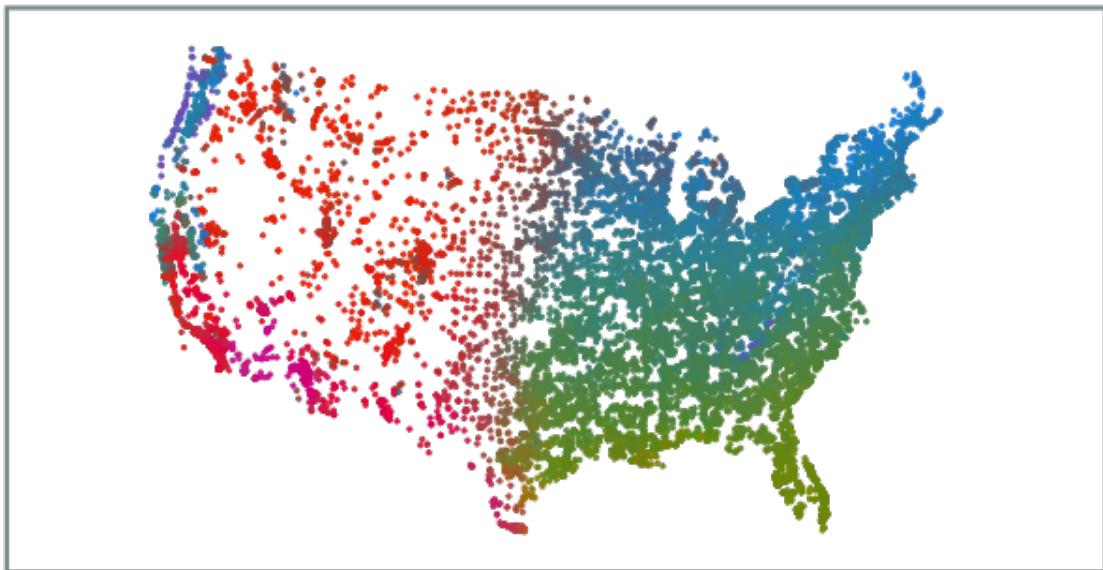
```
[1] 38487    2
```







```
## ddn <- ddn %>%
##   mutate(value = ifelse(gdd > 9000, "gray", value),
##         value = ifelse(gdd < 2000, "gray", value),
##         value = ifelse(precip > 6700, "gray", value),
##         value = ifelse(precip < 300, "gray", value))
```



```
library(grid)
vp1 <- viewport(width = 0.5, height = 0.35, x = 0.743, y = 0.824)

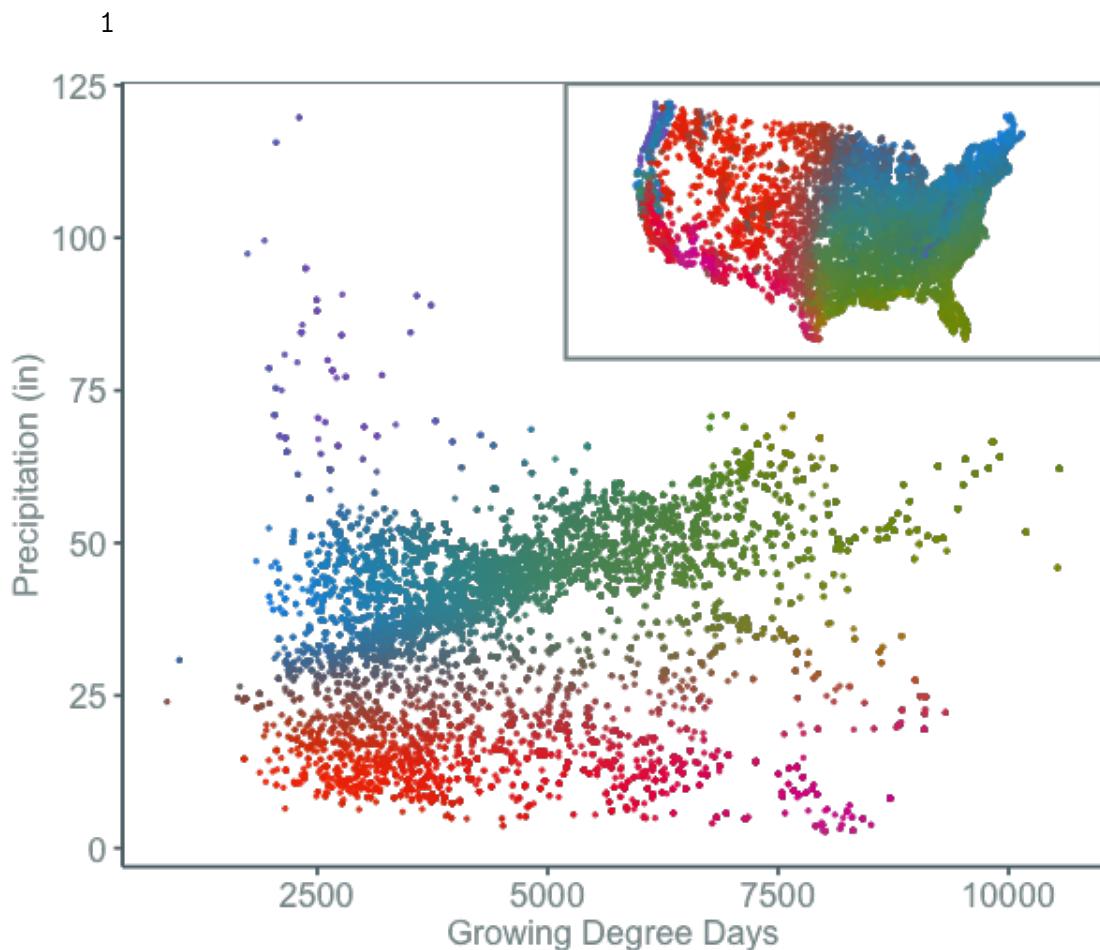
#Just draw the plot twice
```

```

png("../figs/climate_space_wMap.png", bg = "transparent", width = 600)
print(p)
print(mp, vp = vp1)
dev.off()

```

null device



```

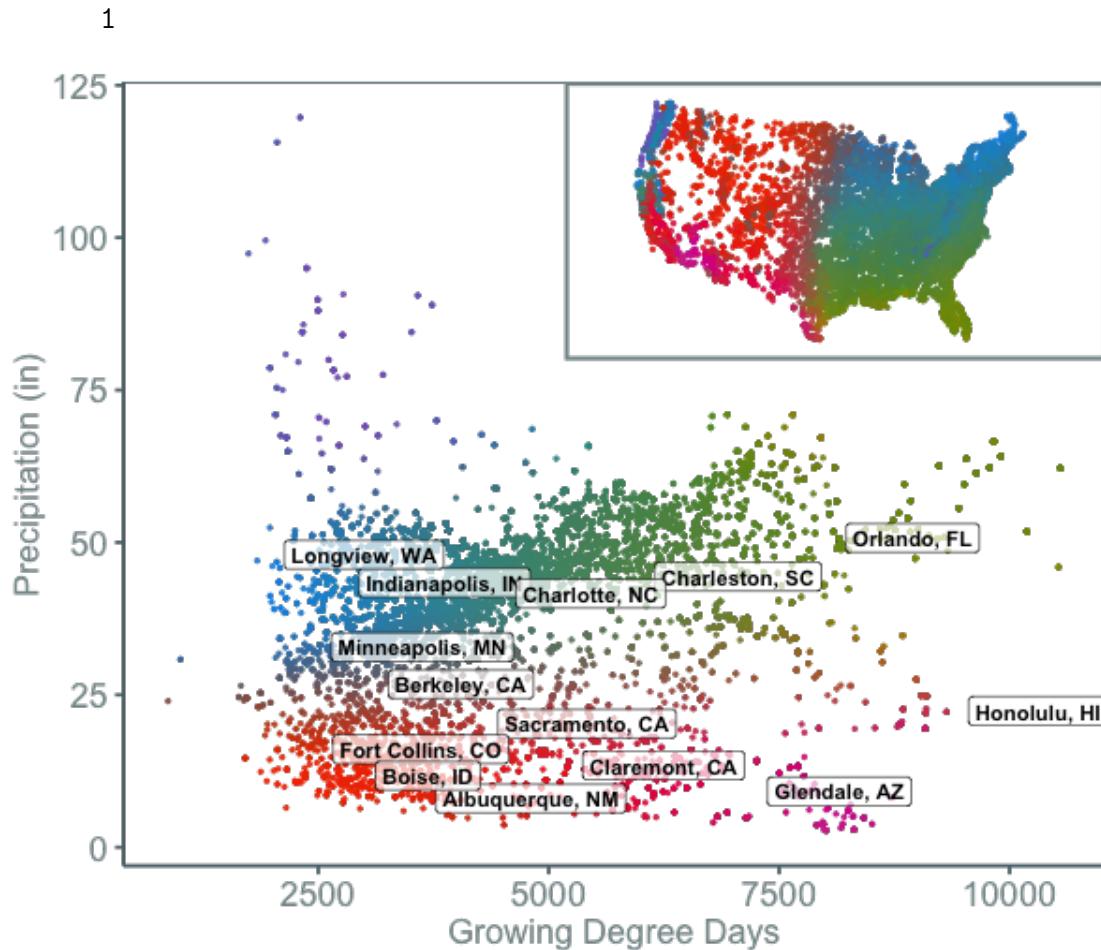
library(grid)
vp1 <- viewport(width = 0.5, height = 0.35, x = 0.743, y = 0.824)

#Just draw the plot twice
png("../figs/climate_space_wMap_labels.png", bg = "transparent", width = 600)

```

```
print(pc)  
print(mp, vp = vp1)  
dev.off()
```

```
null device
```



TODO: overlay the reference cities.

```
dn <- d %>%
  rename(lat = LATITUDE, long = LONGITUDE, pop = POPULATION) %>%
  filter(complete.cases(.)) %>%
  filter( lat < 50, lat > 25, long < 0)

col_gdd <- sapply(dn$gdd, function(x) cols$gdd_col[which.min(abs(x - cols$gdd_col))])
col_precip <- sapply(dn$precip, function(x) cols$precip_col[which.min(abs(x - cols$precip_col))])
```

```

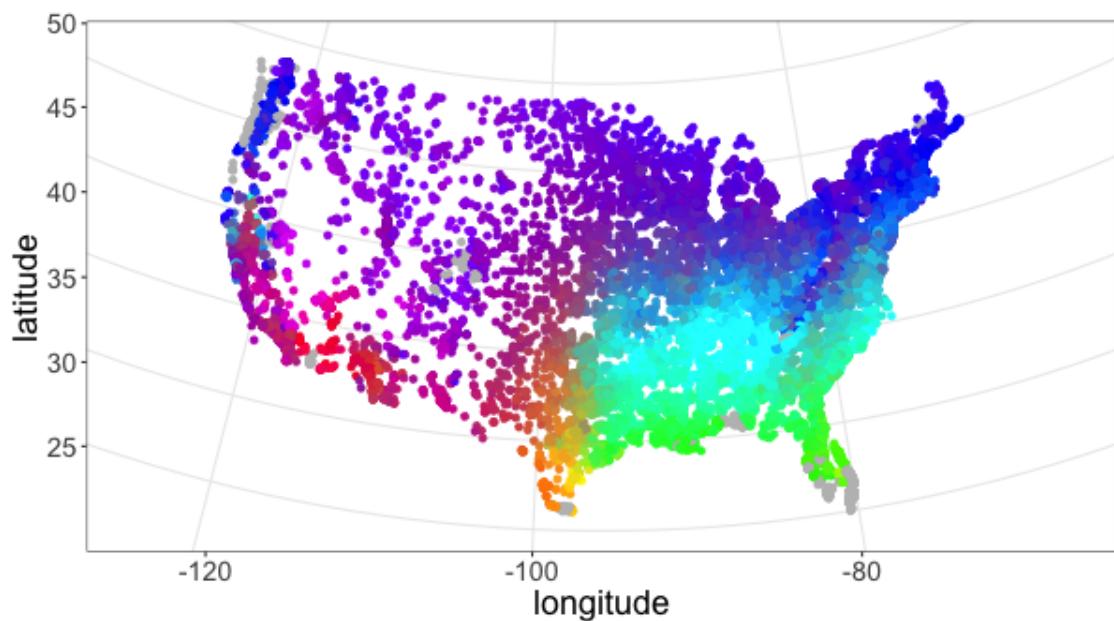
dn <- dn %>%
  mutate(gdd_col = col_gdd,
        precip_col = col_precip)

ddn <- left_join(dn, cols)

ddn <- ddn %>%
  mutate(value = ifelse(gdd > 9000, "gray", value),
        value = ifelse(gdd < 2000, "gray", value),
        value = ifelse(precip > 6700, "gray", value),
        value = ifelse(precip < 300, "gray", value))

Joining, by = c("gdd_col", "precip_col")

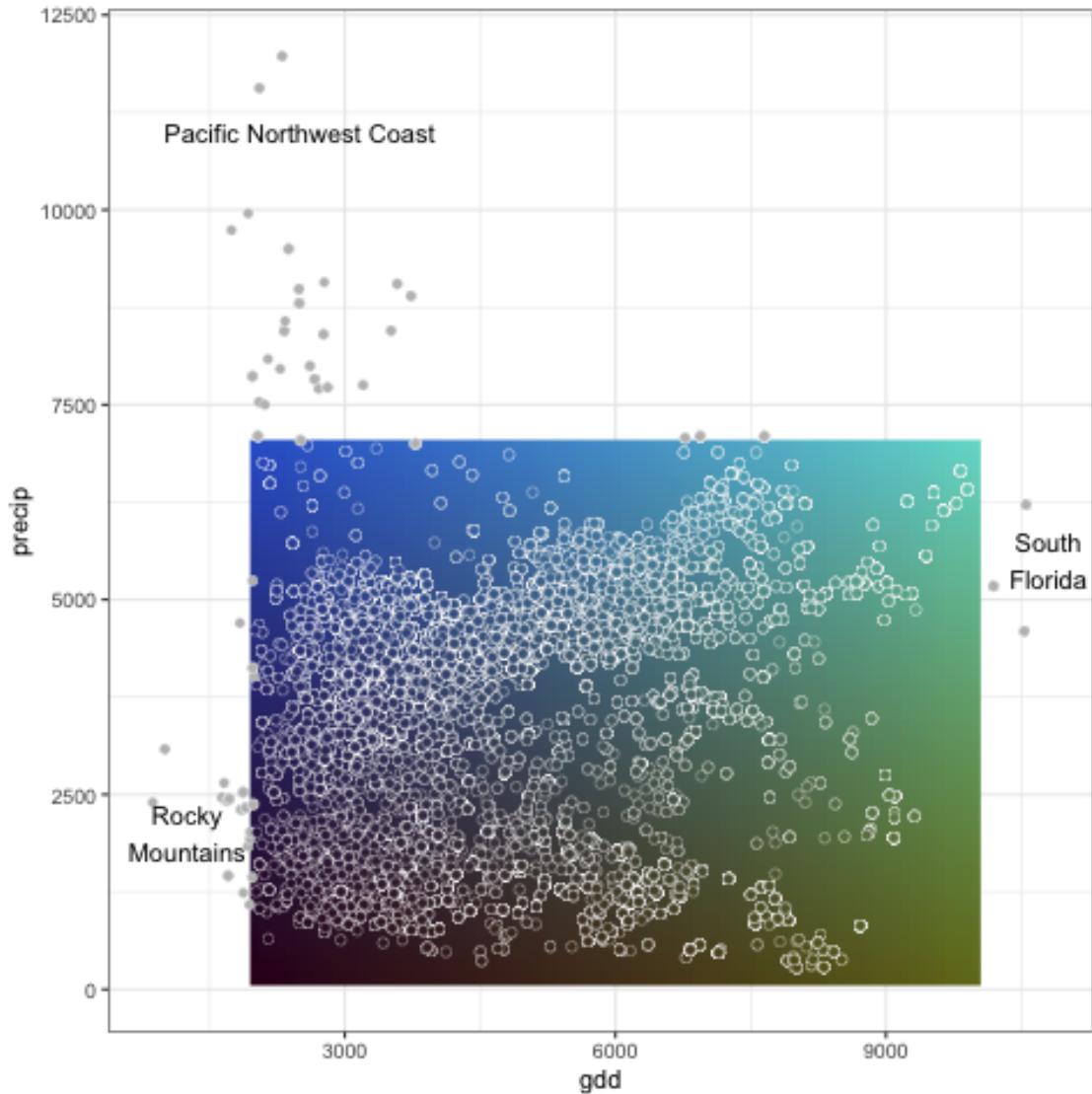
```



```

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$gdd), range(d$precip)))

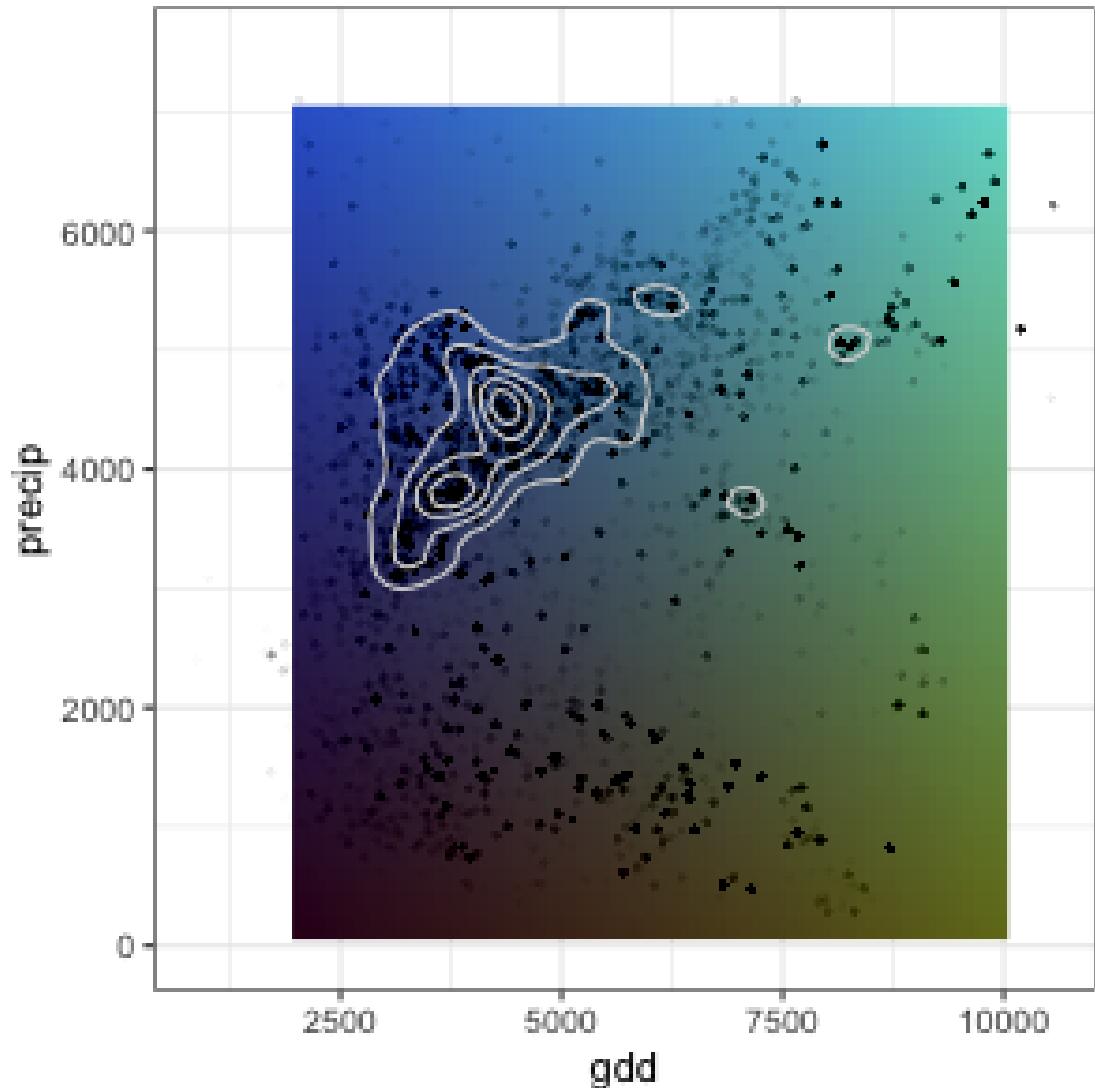
con <-  MASS::kde2d(x = d$gdd, y = d$precip, n = 100, lims = c(range(d$gdd), range(d$precip)))

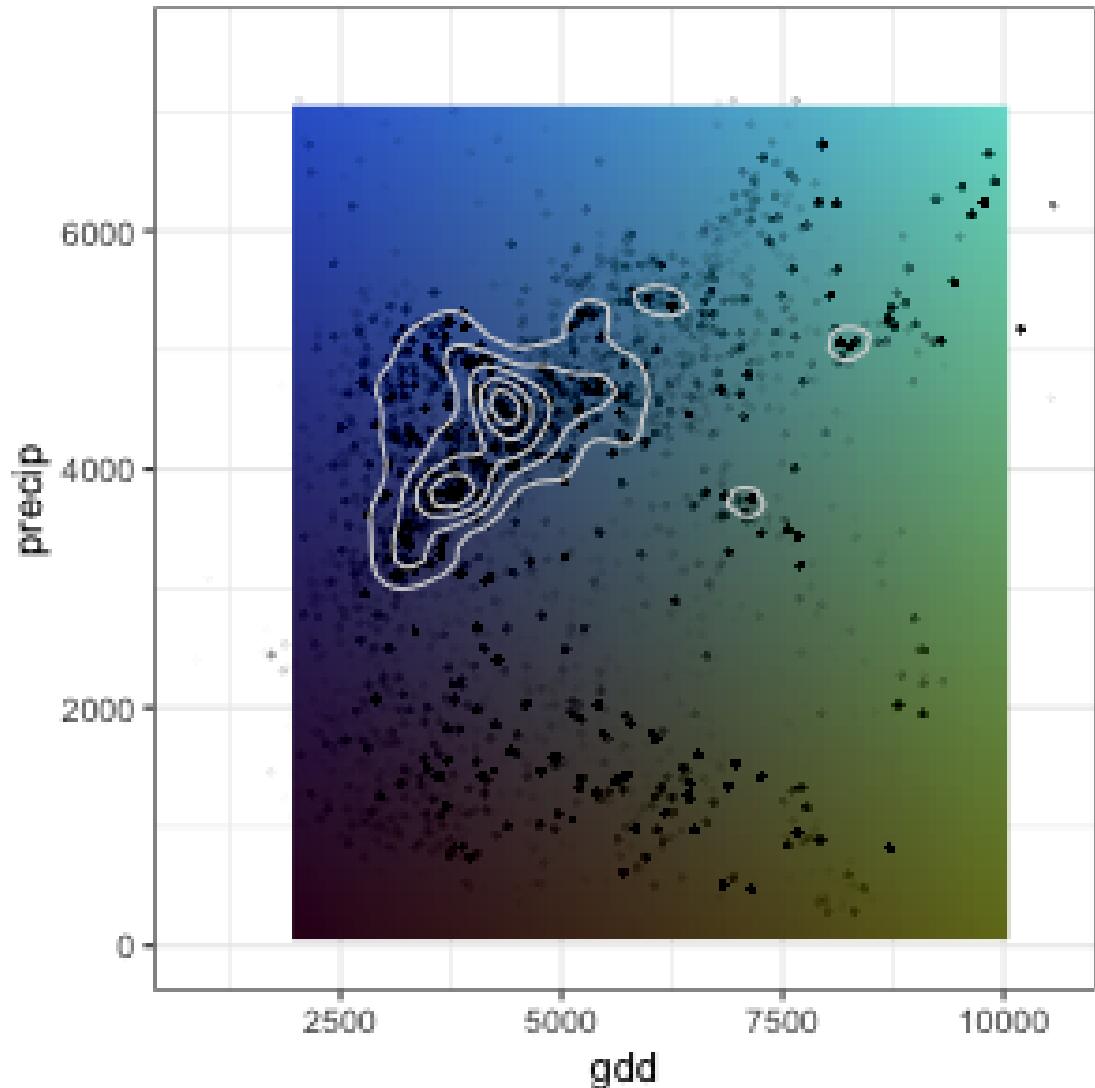
#  con2 <-  hdr.2d(x = con$x, y = con$y, den = con, prob = c(50,90))
#  con2 <-  hdr.2d(x = d$gdd, y = d$precip, prob = c(20,40,60,80))

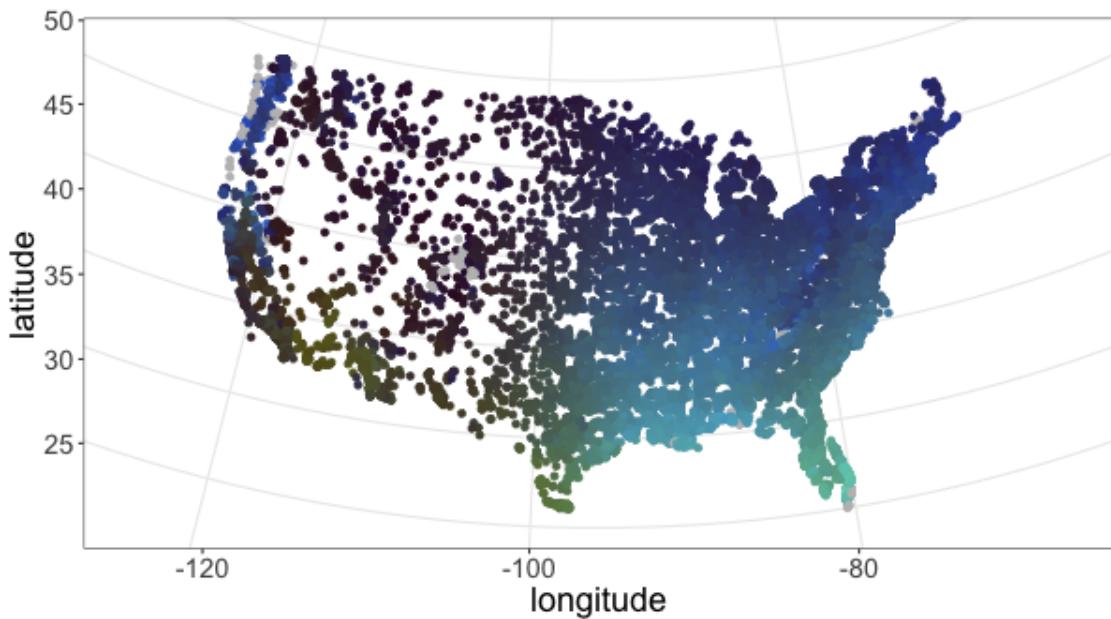
con3 <- expand.grid(gdd = con2$den$x, precip = con2$den$y)
con3 <- expand.grid(gdd = con$x, precip = con$y)

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

Error in expand.grid(gdd = con2$den$x, precip = con2$den$y) :
object 'con2' not found
```







Trying to do log scale

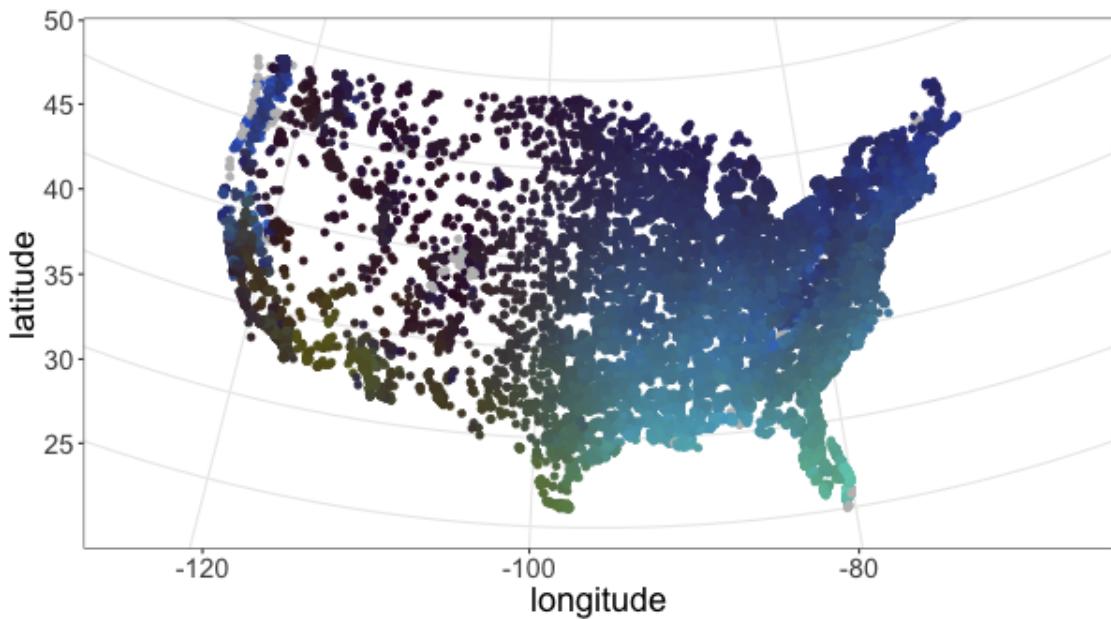
```
d <- bind_cols(dsp.trks@data, trks@data) %>%
  dplyr::select(LATITUDE, LONGITUDE, POPULATION, gdd, precip, TRACT)

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, bl
```

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



Making figures for this data section

- species by city raster, black if the combination exists

```

library(dplyr)
library(ggplot2)

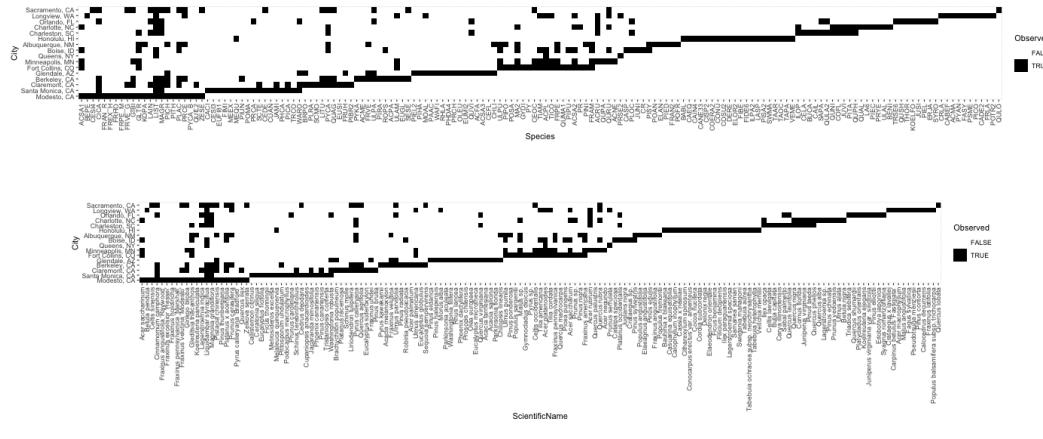
d <- readRDS("../data/tidy_age_dbh.rds")

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)

```

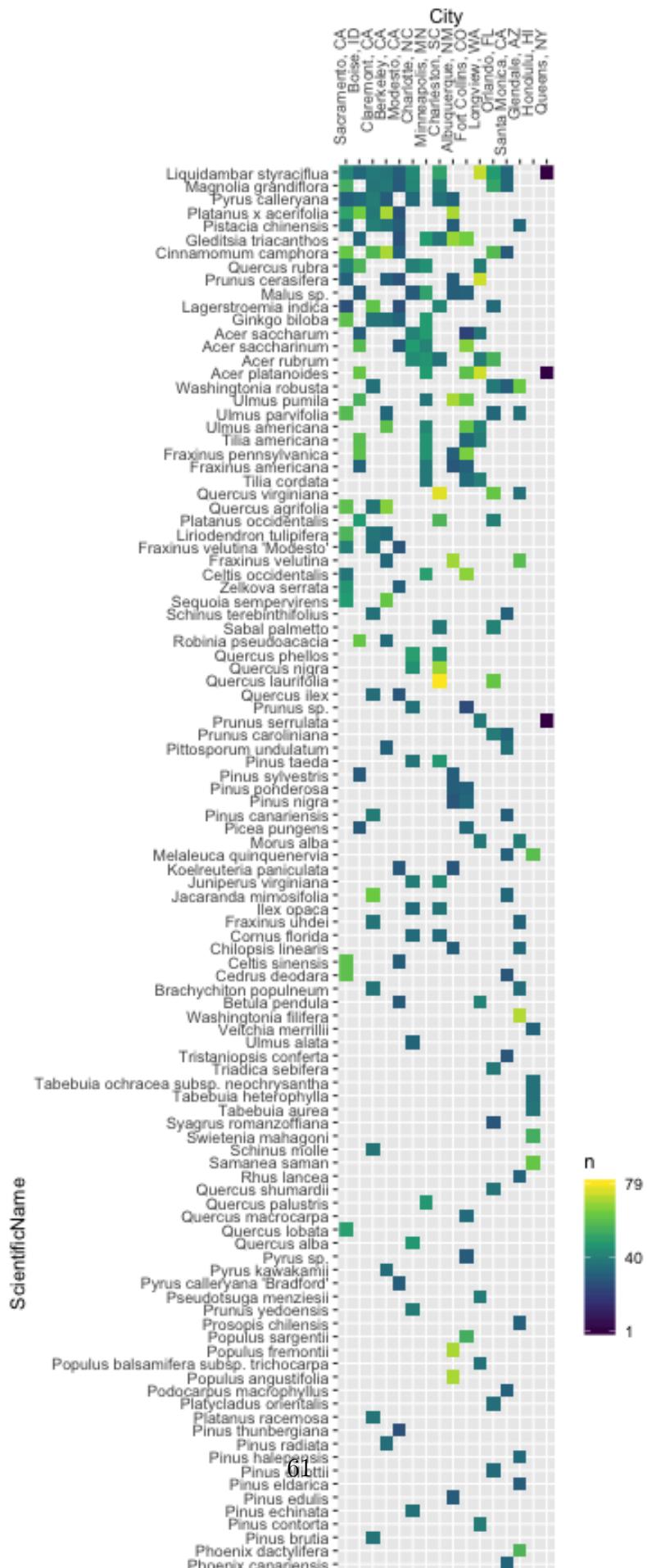


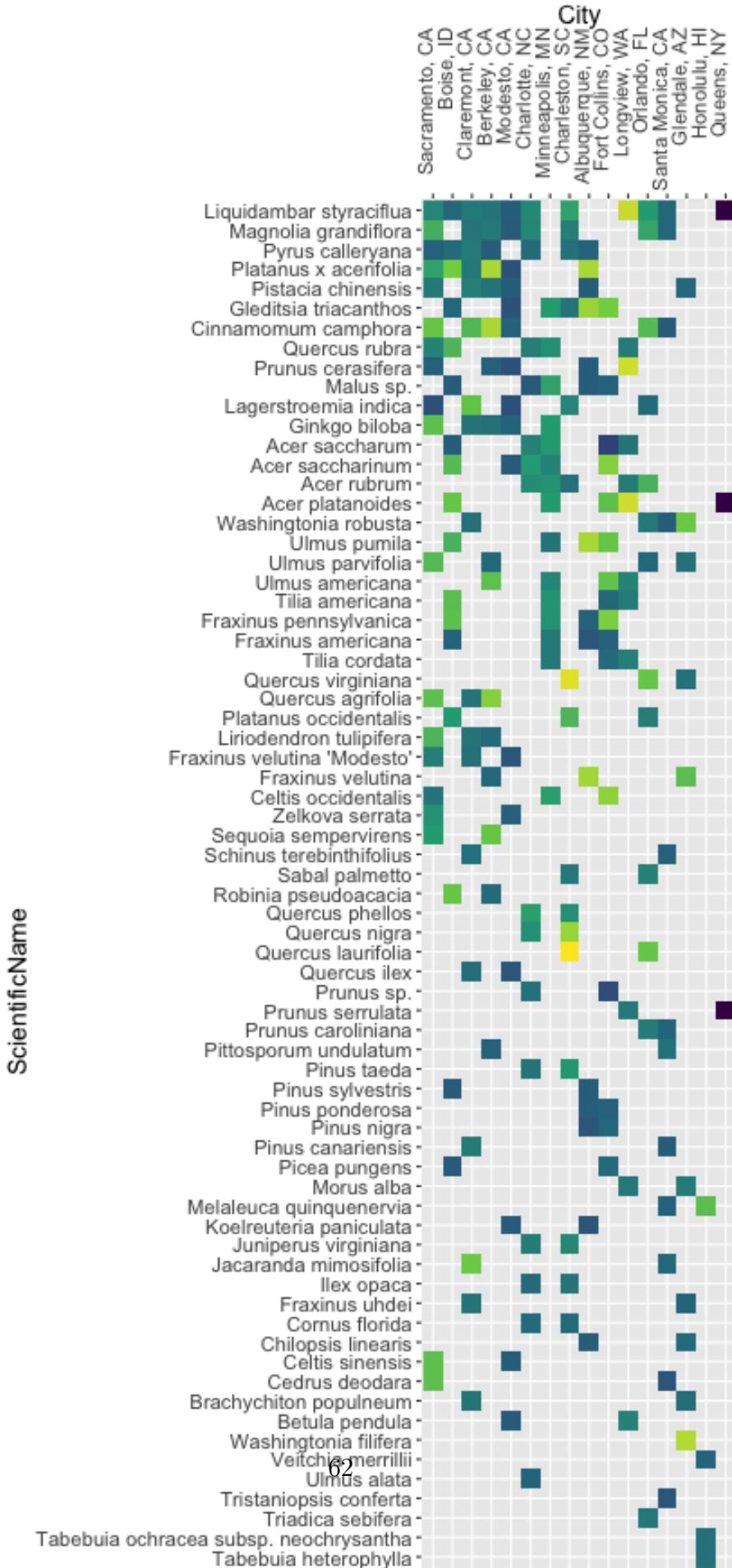
object 'cs_sp' not found

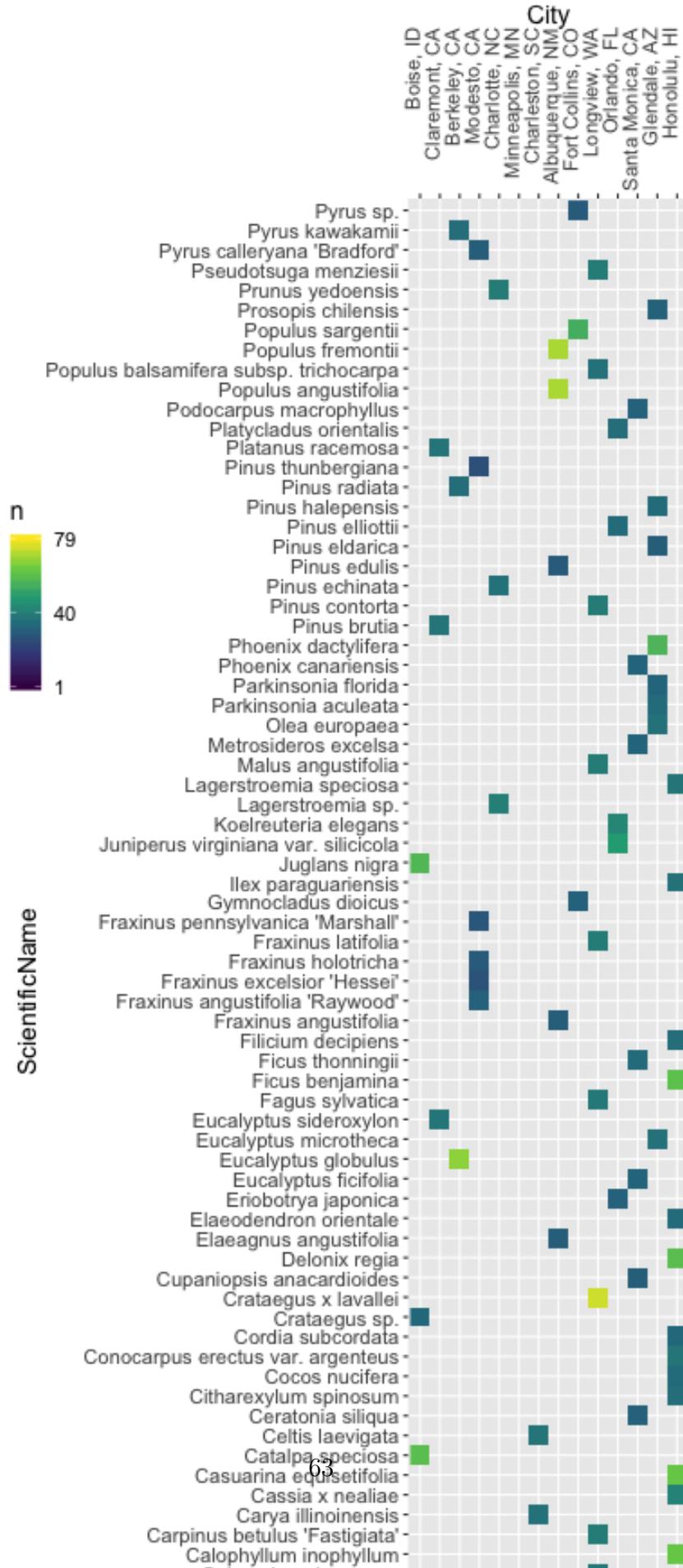
- trees as fill. That is richer information.

Also along the top I could have a N cities row, and along the right I could have a N species column. But that's a lot of work for what is readily apparent when looking.

I should order left to right by n cities that a species is observed in and then by city (alphabetic)



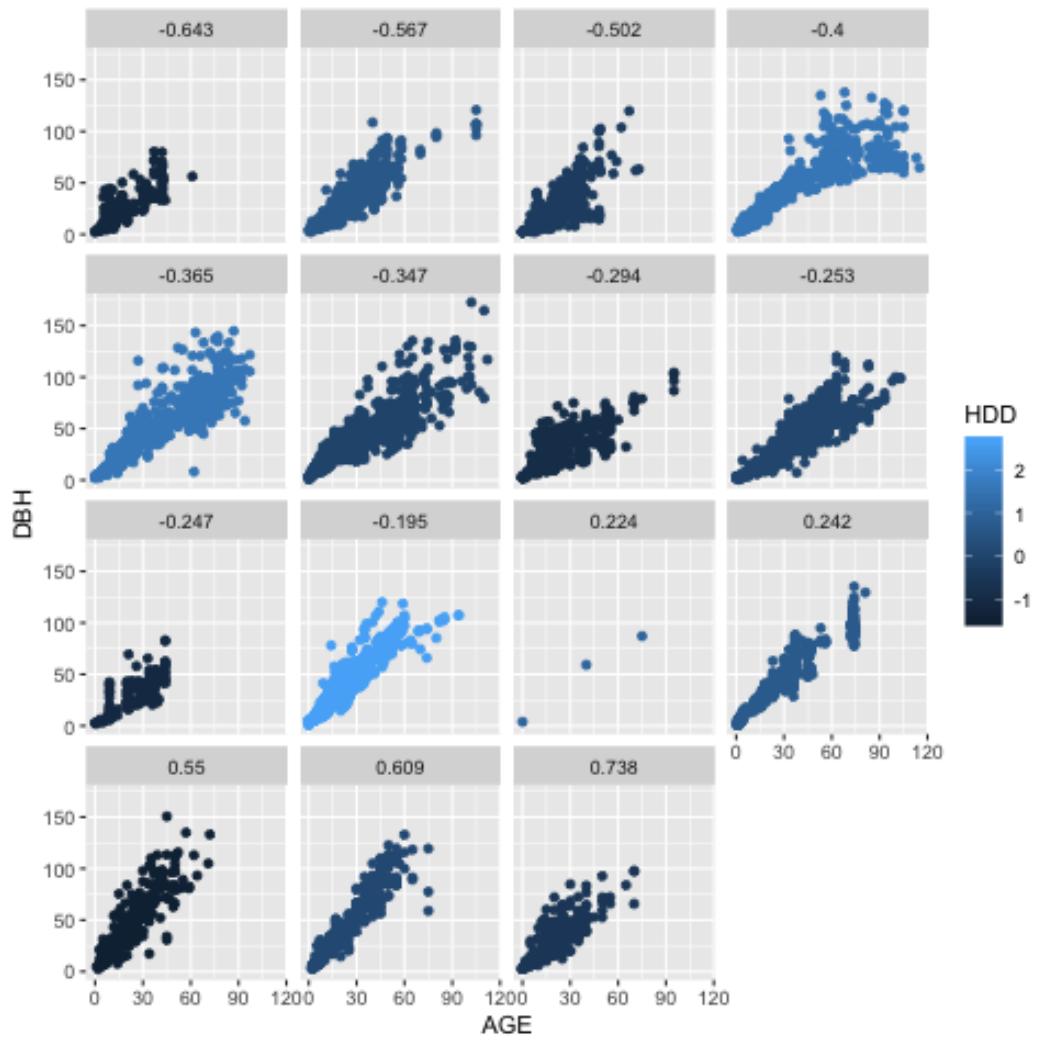


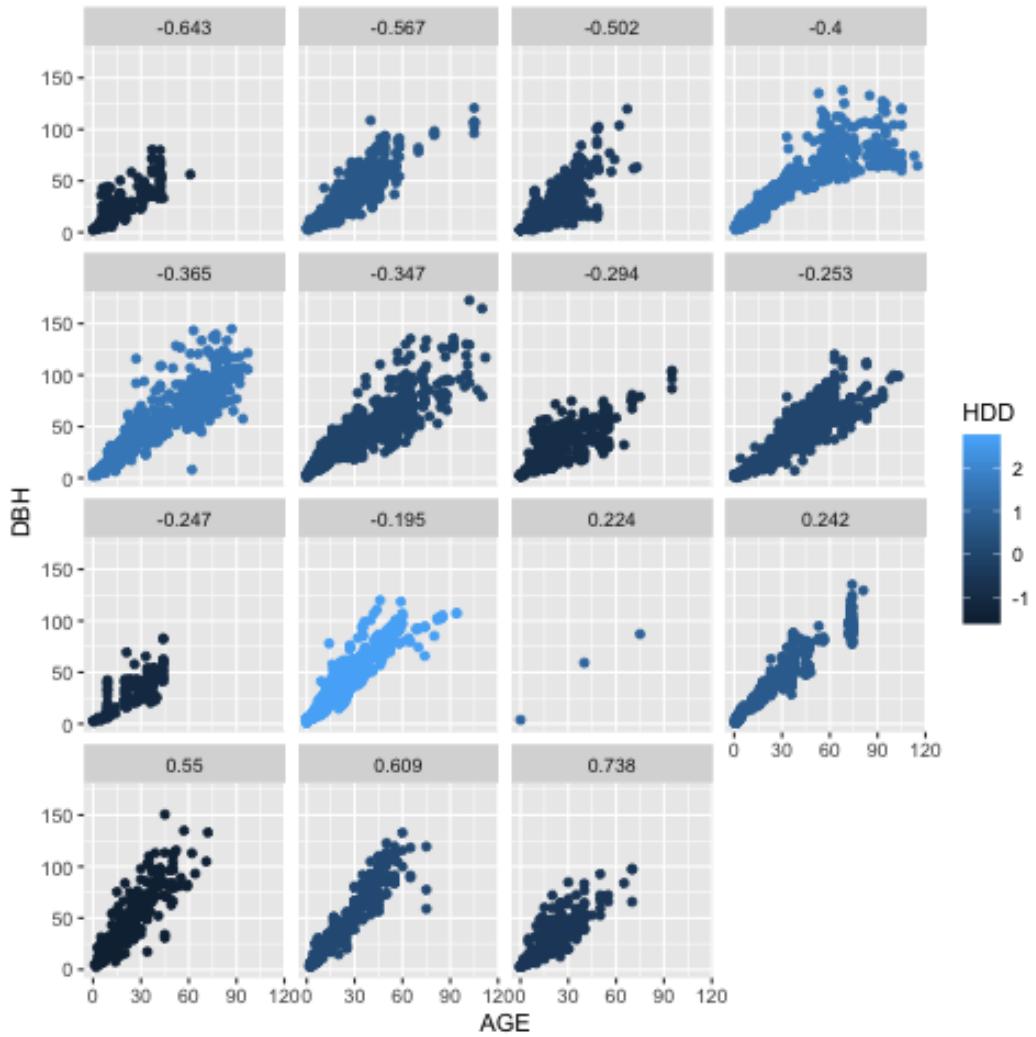




- look at climate variables

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





Is minneapolis an influential point?

Modelling

Model requirements

We sought a model of tree growth that would adequately represent the known biological dynamics of tree growth, namely that diameter growth rate starts slow, reaches a maximum at a young age, and then gradually declines to nearly zero. Diameter for trees much always increase however slightly because the growth of new wood is essential for proper function. This is different than tree height which often reaches its asymptote. Instead the asymptote

in our curve could be considered a pragmatic way to constrain diameter growth, or a practical asymptote. While trees could theoretically continue to increase in diameter indefinitely, they don't in reality. The asymptote represents this practical maximum diameter. An additional feature to the data is that age is time since transplanting. This means trees can have substantial diameter at age 0.

The type of curve that meets these criteria would be an asymmetric sigmoidal curve with an intercept. A modified weibull is such a curve that has worked well in forestry and is the one we use here ?. However, there are many other curves that meet these criteria and could be used.

Another characteristic of tree growth curves is heteroscedasticity, namely that as the age of a tree increases, so does the variability around the mean. Often past modellers controlled this using log - log transformations ?, but we wanted to keep units in their original scale. We tested fitting models where the variance was a linear function or a smoothed spline function of age. However, this still could yield negative predictions at low ages. Instead we adopted the approach of modeling DBH from a gamma distribution, which yielded more realistic posterior predictions.

Model

(Notation below. I don't use conventional subscript letters. The nesting of species within genus isn't obvious either until lower levels. I did this because genus could also end up being nested. There is a fair bit going on here, suggestions for how to improve are much appreciated.)

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

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

the mean, μ_i , is modelled using the weibull curve, a function that has been used for

decades in forestry to successfully model tree growth ? (pg 116).

β_0 is intercept. x is transplant age so when x = 0, y can be well above 0.

β_1 is asymptote of sigmoidal weibull curve. For most species there are no data near the true asymptote and so this parameter should be considered a highly uncertain estimate of the real maximum dbh of a tree.

β_2 and β_3 affect the rate of growth. β_2 provides flexibility to have slow or fast growth at young ages (small x).

All β '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.

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

for β_{1s}

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

where β_j is the mean for β coefficient β_j . γ_{js} is the contribution of genetic (species) effect for species s on β_j . δ_{jc} is the city effect for city c on β_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:

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

Priors:

Priors were selected to be slightly informative and make very biologically unreasonable parameters improbable. The quantity of data overwhelms the priors

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

parameter	mu	sd	alpha (of gamma)	beta (of gamma)
β_0	3	1.5	4.	1.3333333
β_1	1.5	.3	25.	16.666667
β_2	1.25	.15	69.444444	55.555556
β_3	1	.15	44.444444	44.444444

$$\beta_0 \sim \text{Gamma}(4, 1.33)$$

$$\beta_1 \sim \text{Gamma}(25, 16.7)$$

$$\beta_2 \sim \text{Gamma}(69.4, 55.5)$$

$$\beta_3 \sim \text{Gamma}(44.4, 44.4)$$

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

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

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

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

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

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

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

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

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

variability by species for intercept, β_0

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

bits

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)

fitting model

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."

- 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

- look at weibull

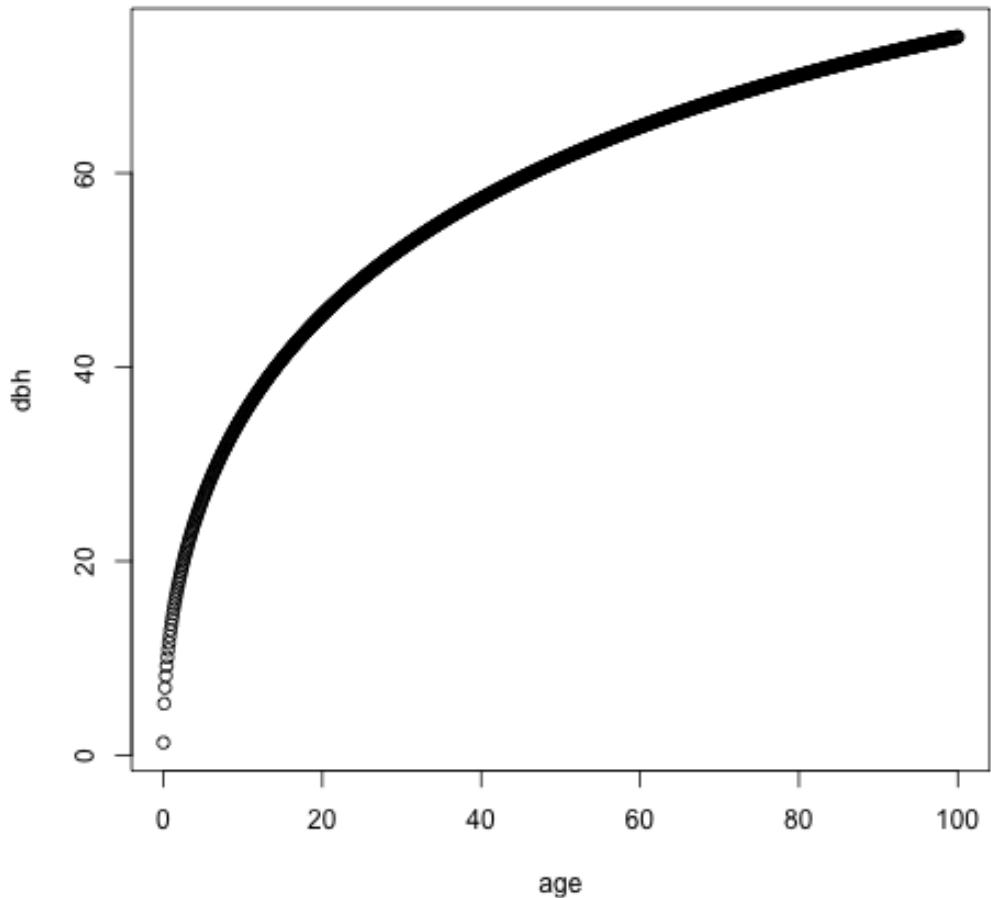
$$\beta_0(1 - \exp(\beta_1 * DBH^{\beta_2}))$$

from huang 1992

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

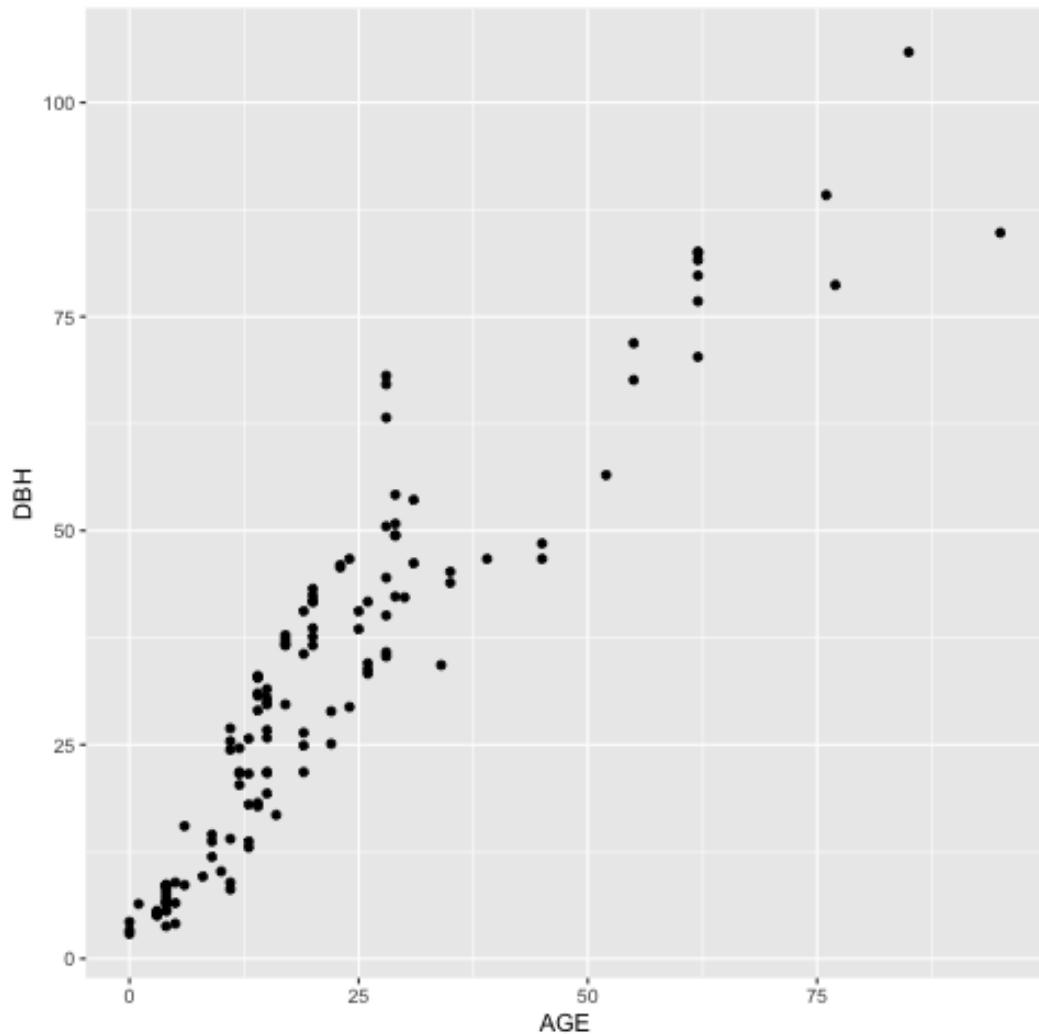
from wikipedia: cdf:

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



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

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

Parameters:

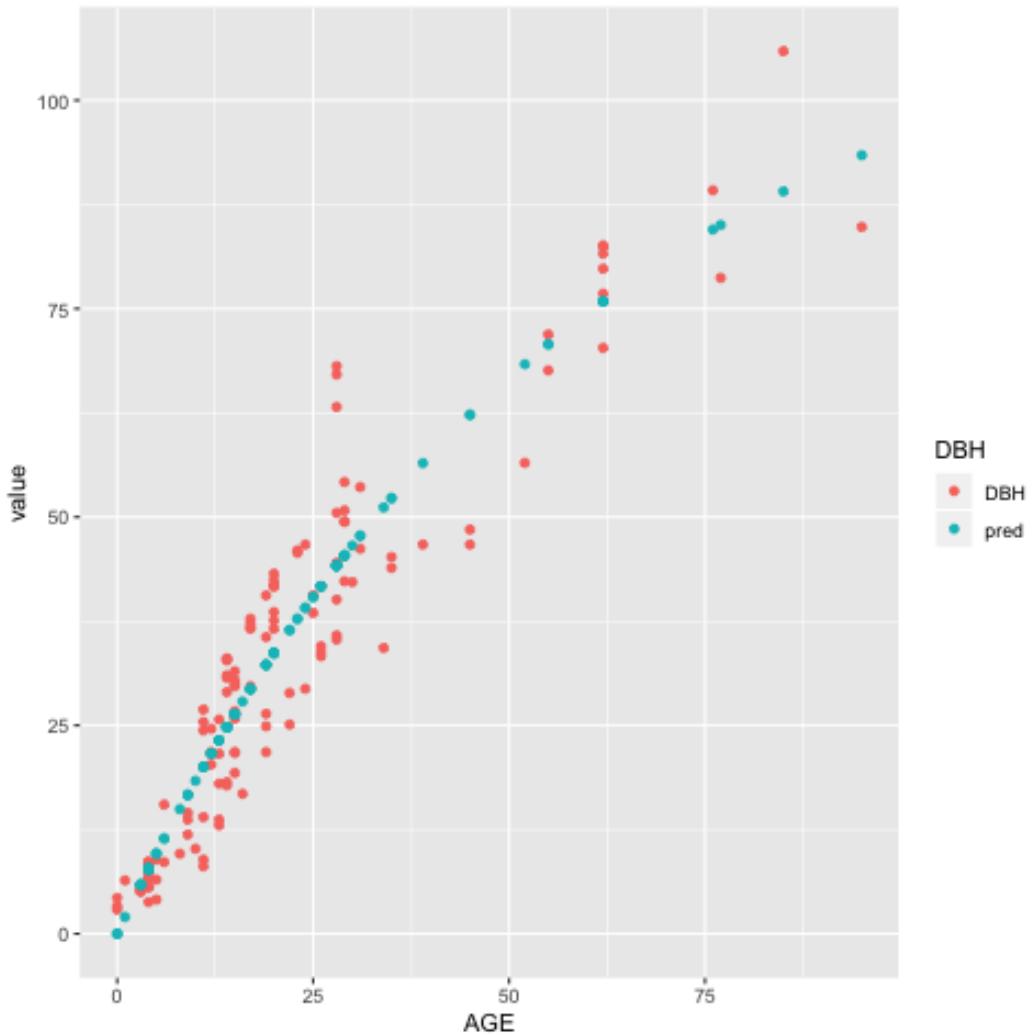
	Estimate	Std. Error	t value	Pr(> t)							
a	1.298e+02	4.928e+01	2.633	0.013606 *							
b	9.289e-03	2.137e-03	4.346	0.000165 ***							
c	1.112e+00	1.745e-01	6.373	6.75e-07 ***							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'. '	0.1	' '	1

Residual standard error: 5.467 on 28 degrees of freedom

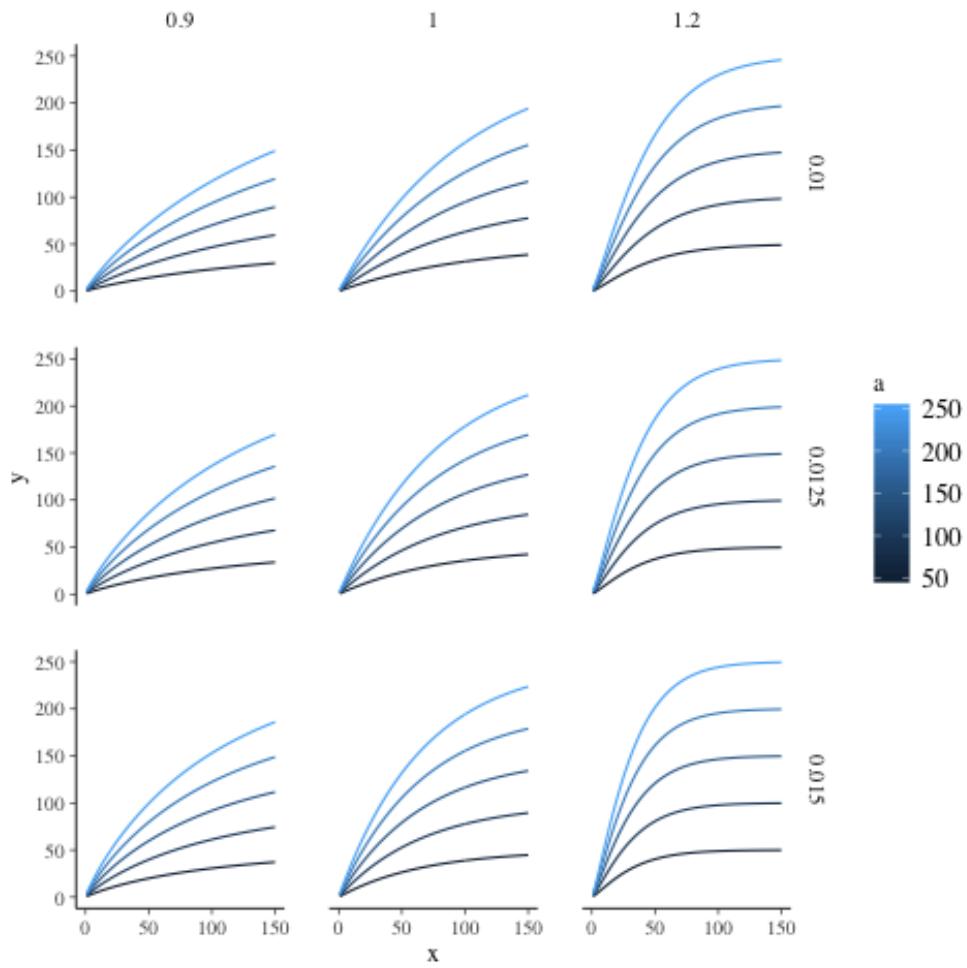
Number of iterations to convergence: 18

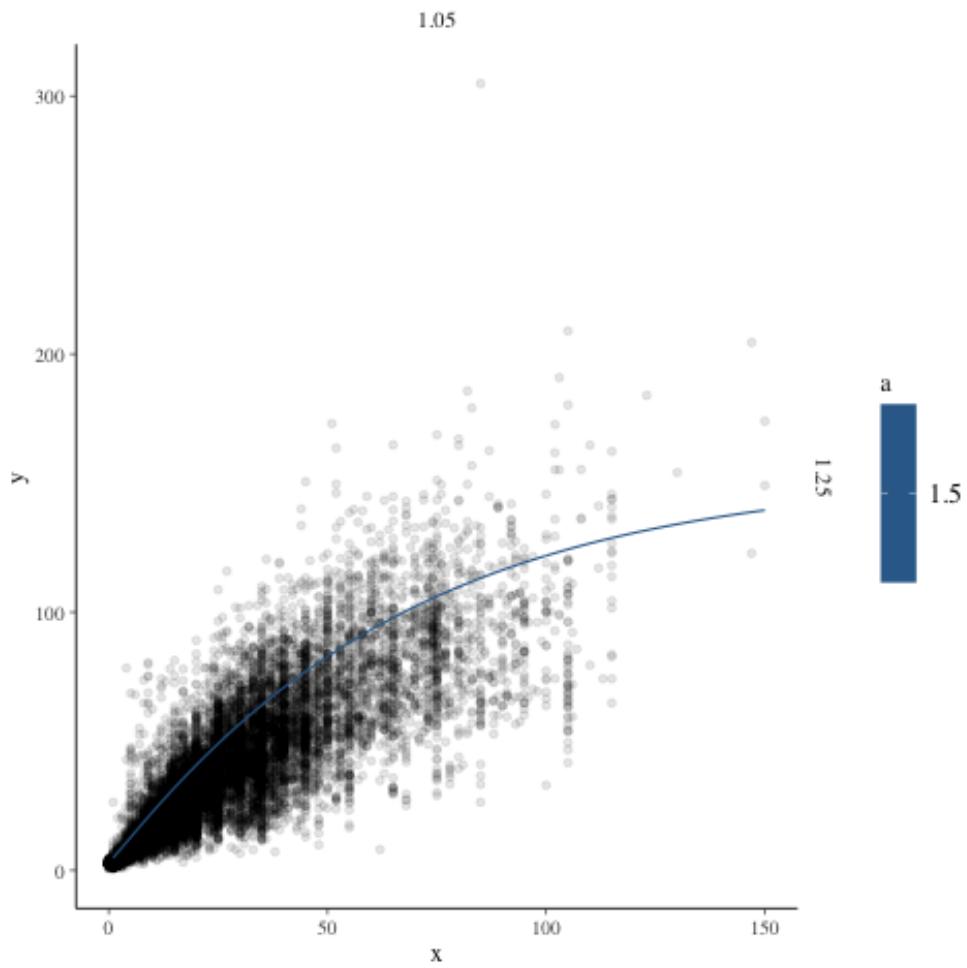
Achieved convergence tolerance: 1.869e-06

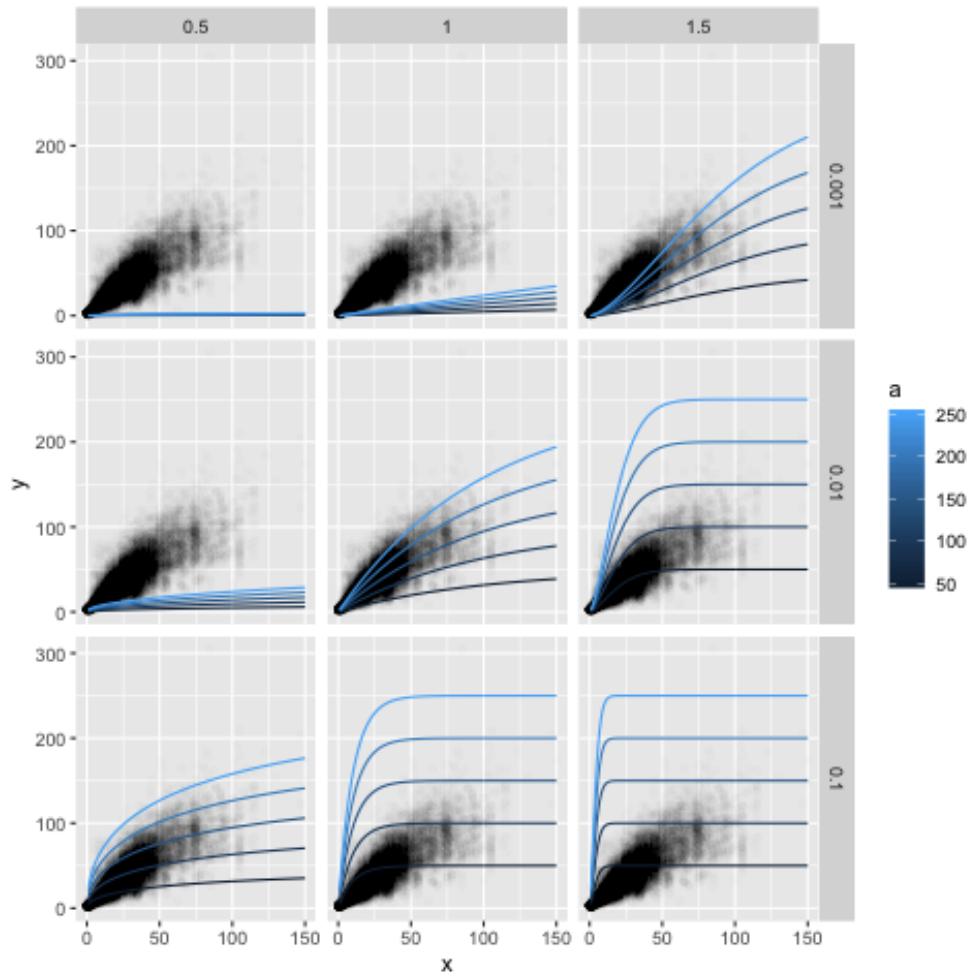


"a" can't be negative because dbh must be positive "b" can't be negative because it causes negative growth and negative dbh values, impossible "c" can't be negative because it causes negative growth, impossible

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



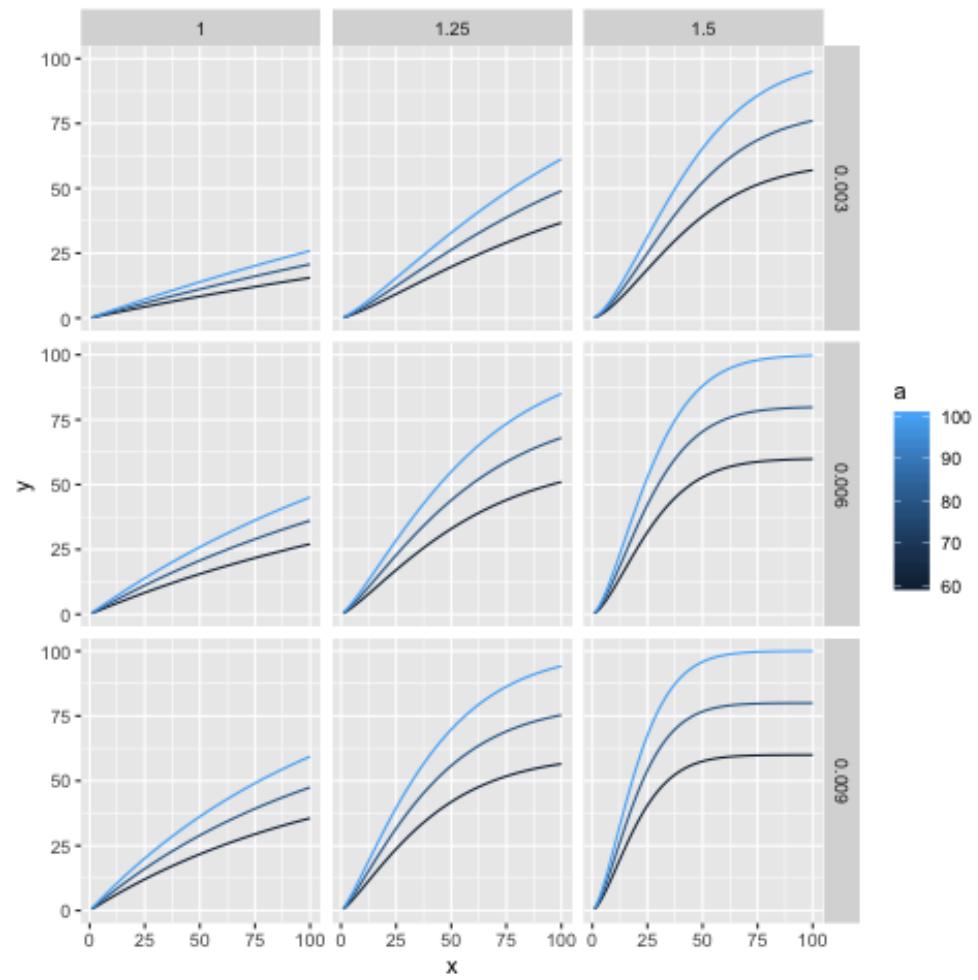
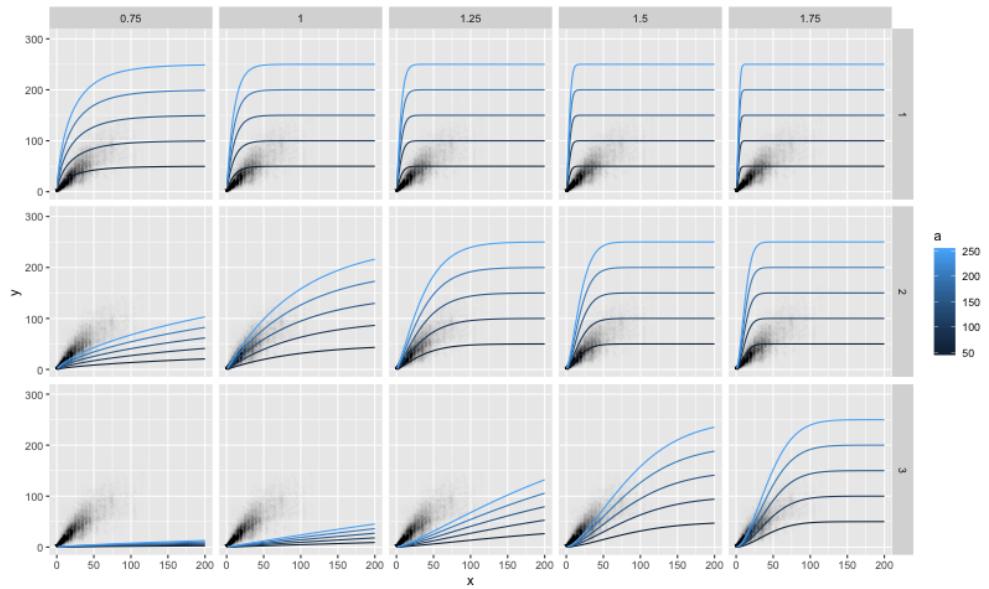


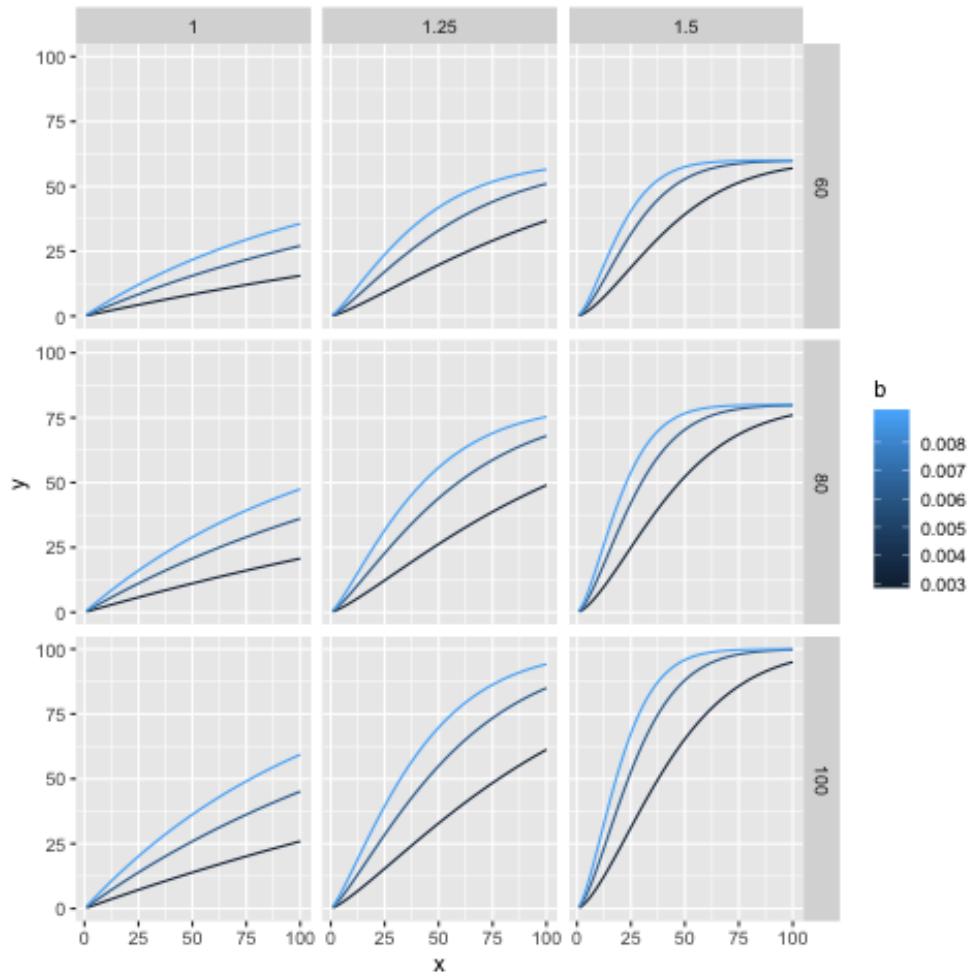


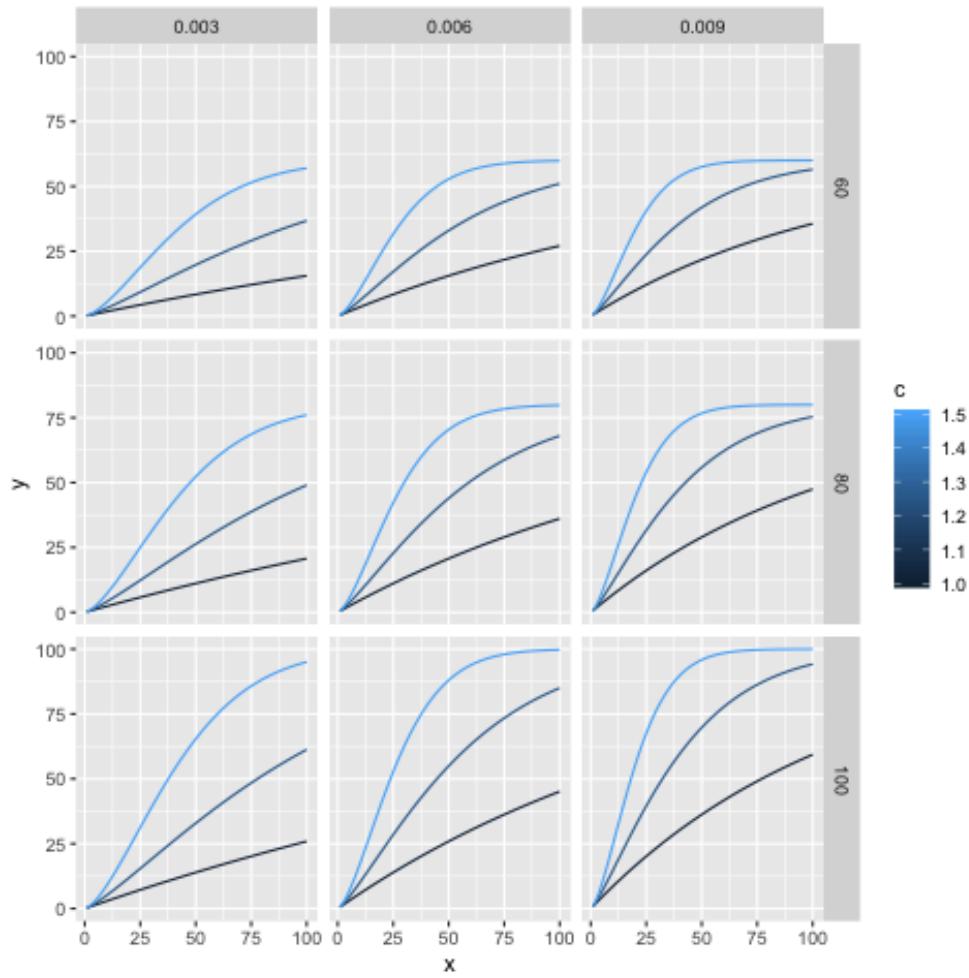
range of realistic values: b1 : 50 - 250 b2 : 0.01 - 0.1 b3 : .5 - 1.5

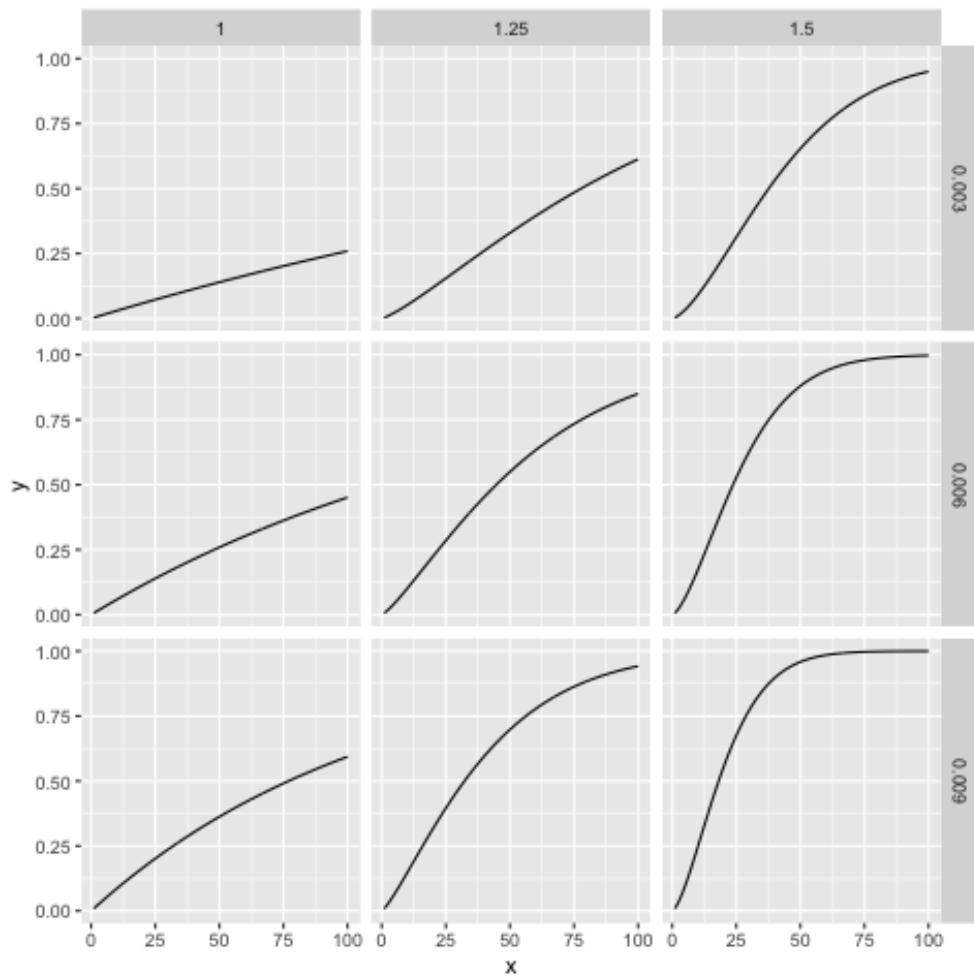
b

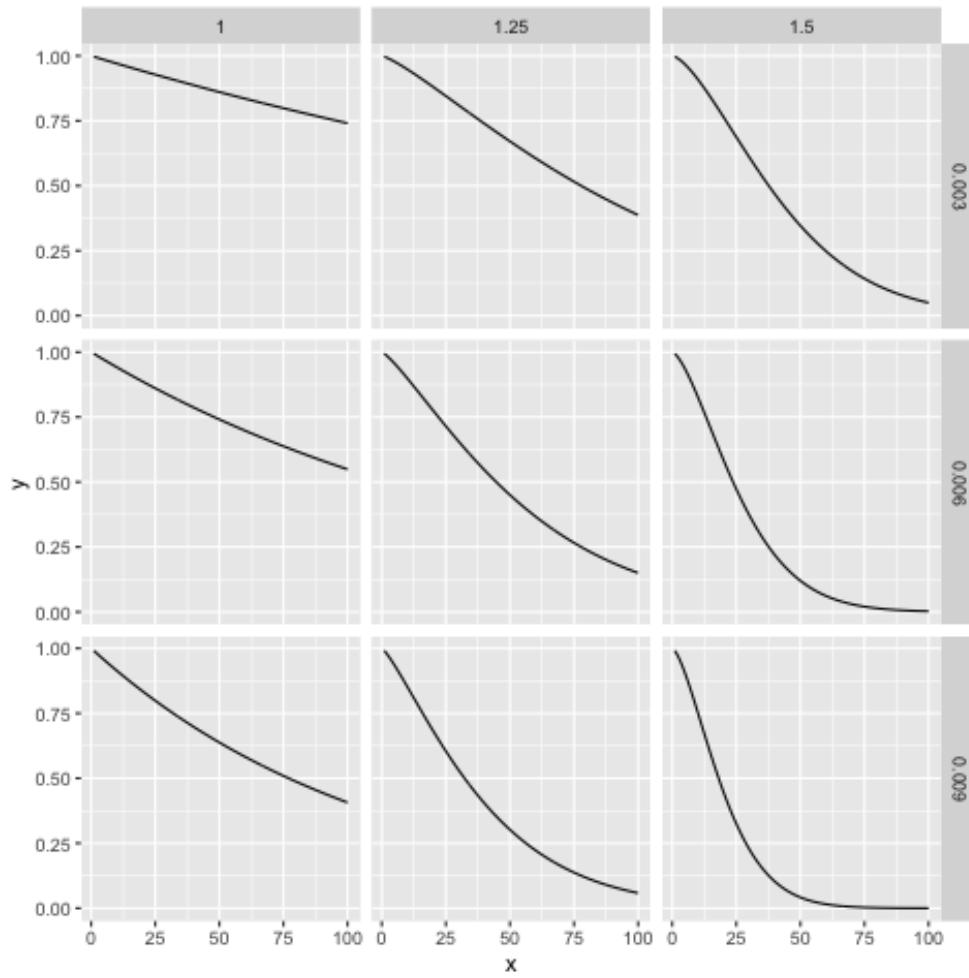
the higher values of b2 only make sense with lower values of b3. they are correlated. I wonder if I could include that information in the prior?

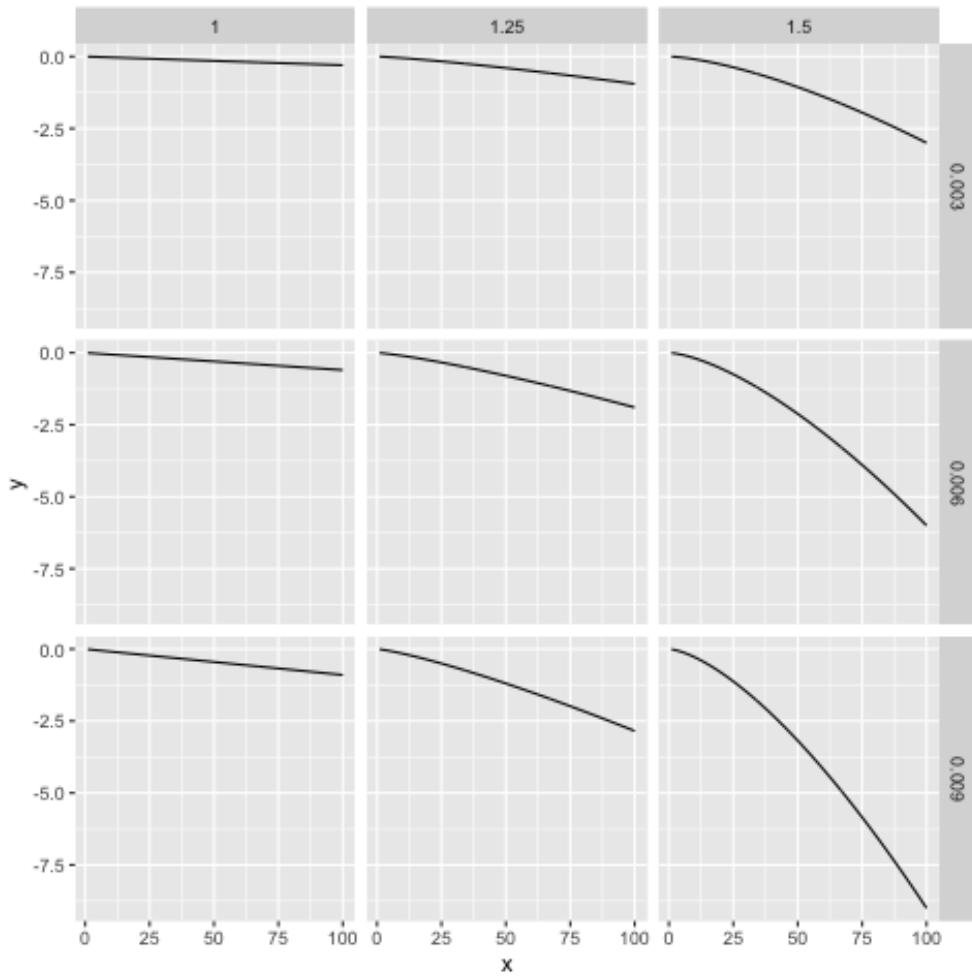


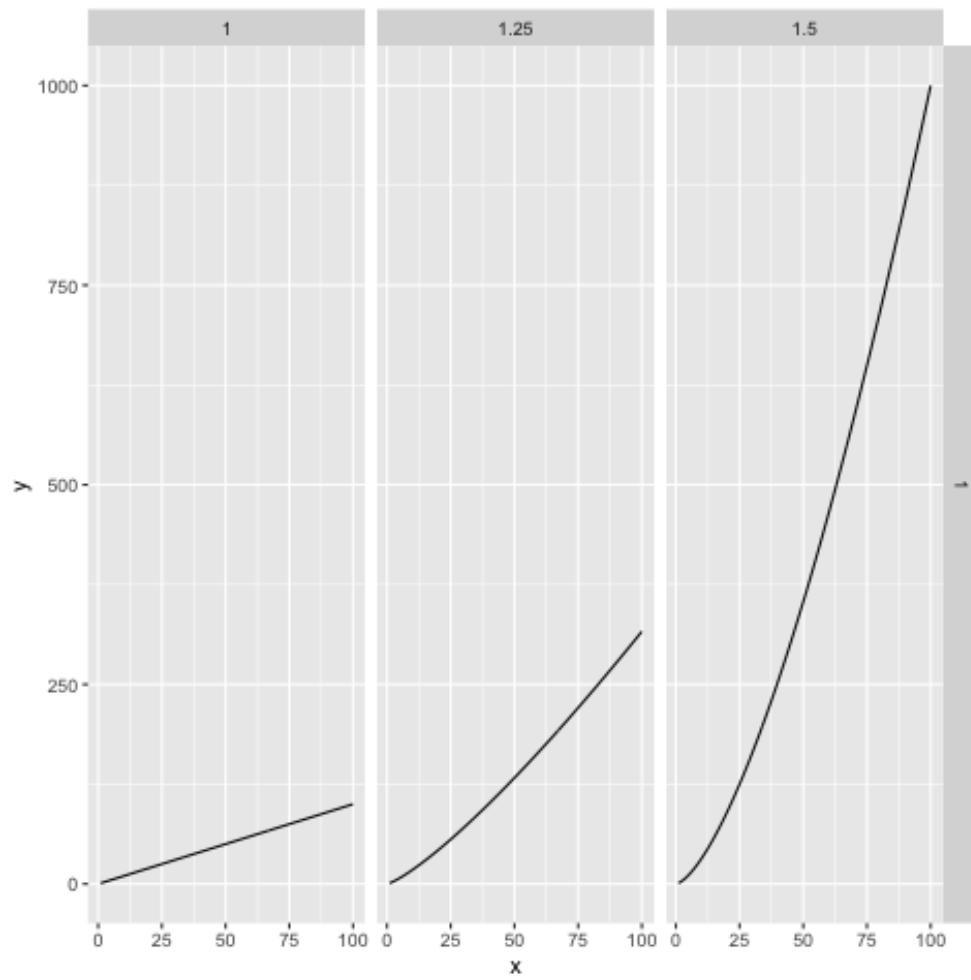












a look at gamma family

```
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(-

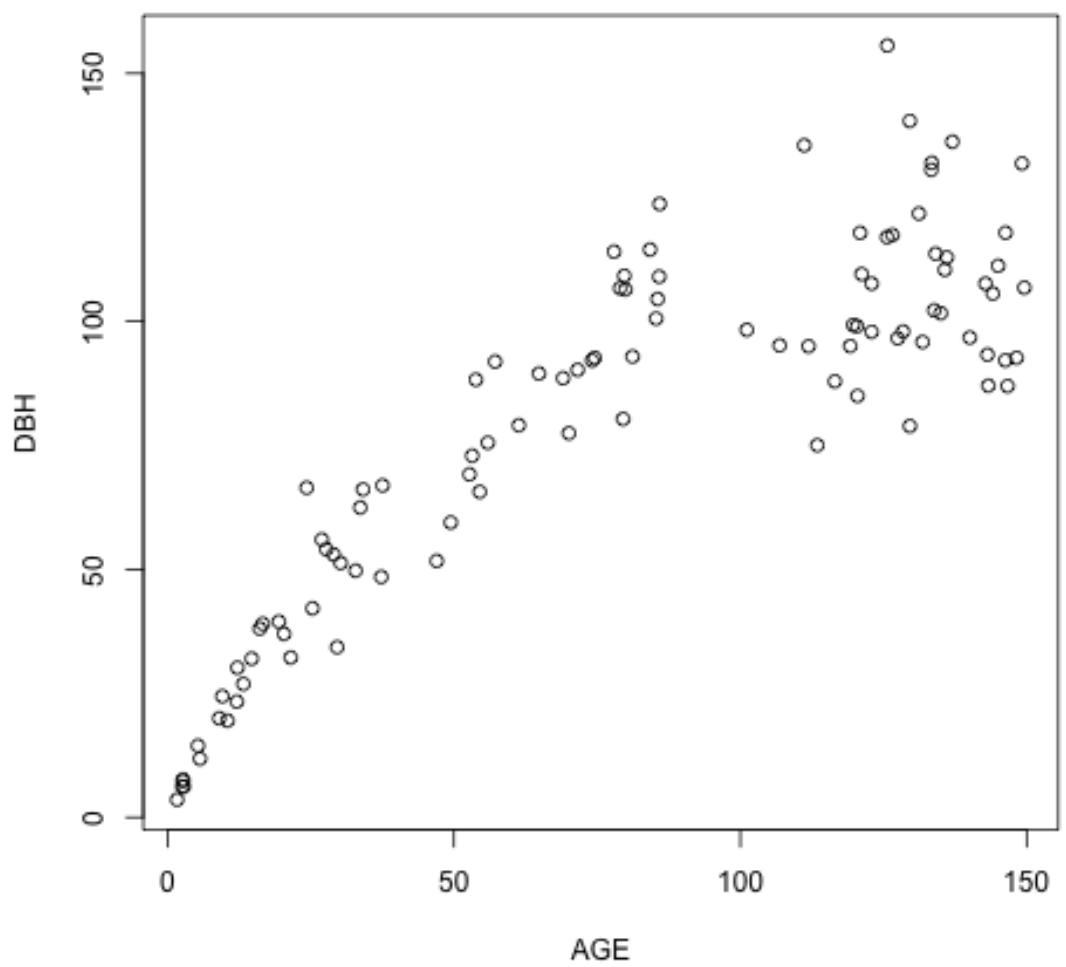
AGE <- runif(n , 0, 150)

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 =
```

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
```

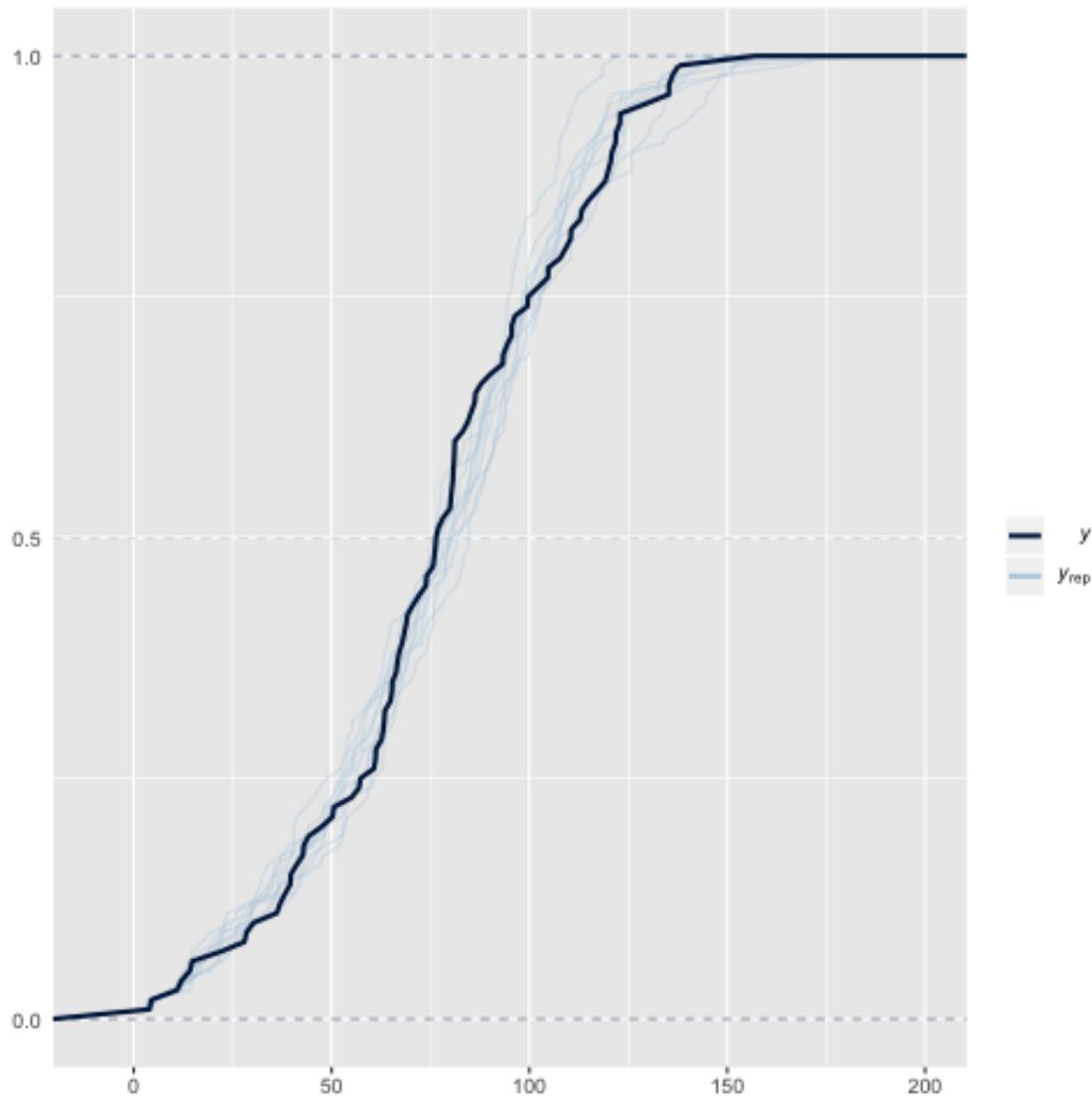
Population-Level Effects:

	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).



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

- 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,

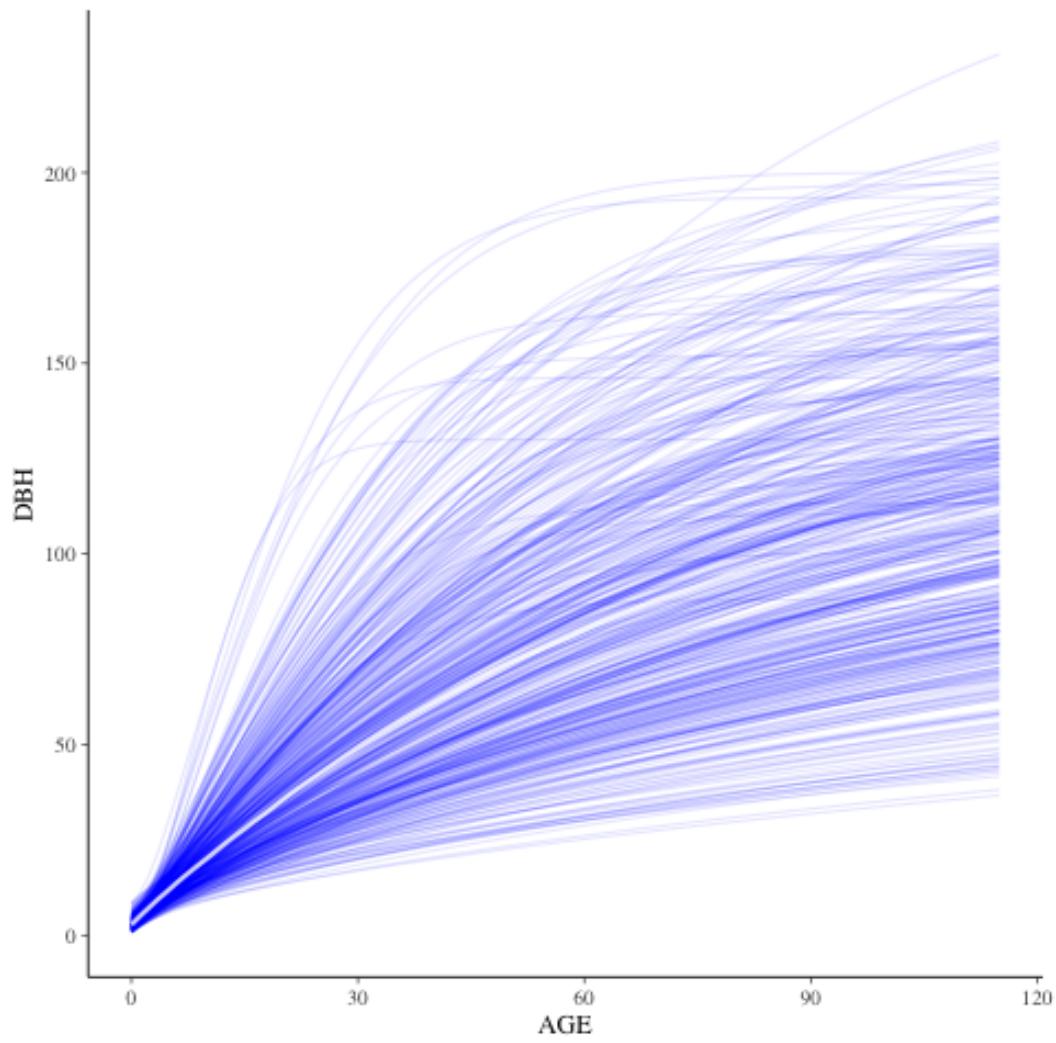
```

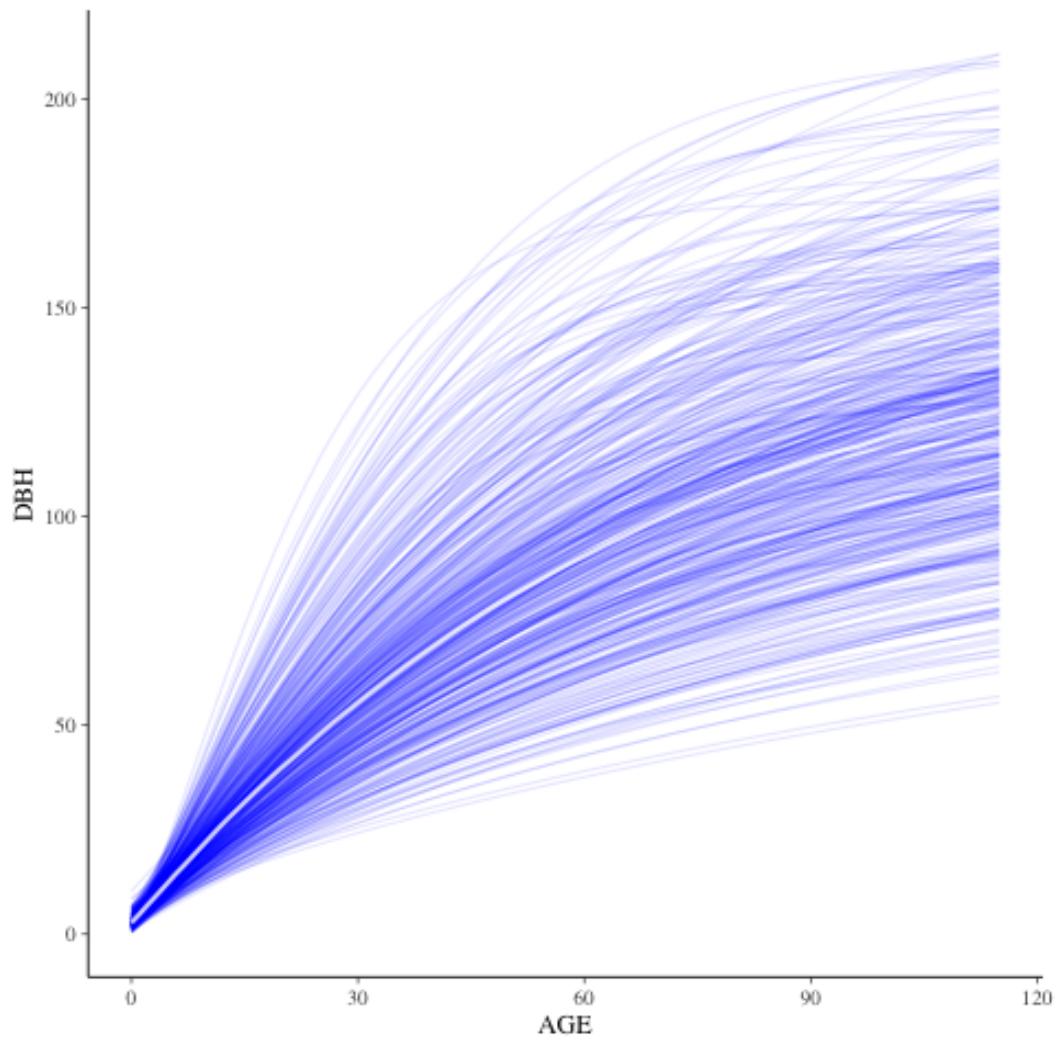
```
prior = nlprior,  
family = Gamma("identity"),  
sample_prior = "only",  
chains = 2, cores = 2, init_r = .3, iter = 500)
```

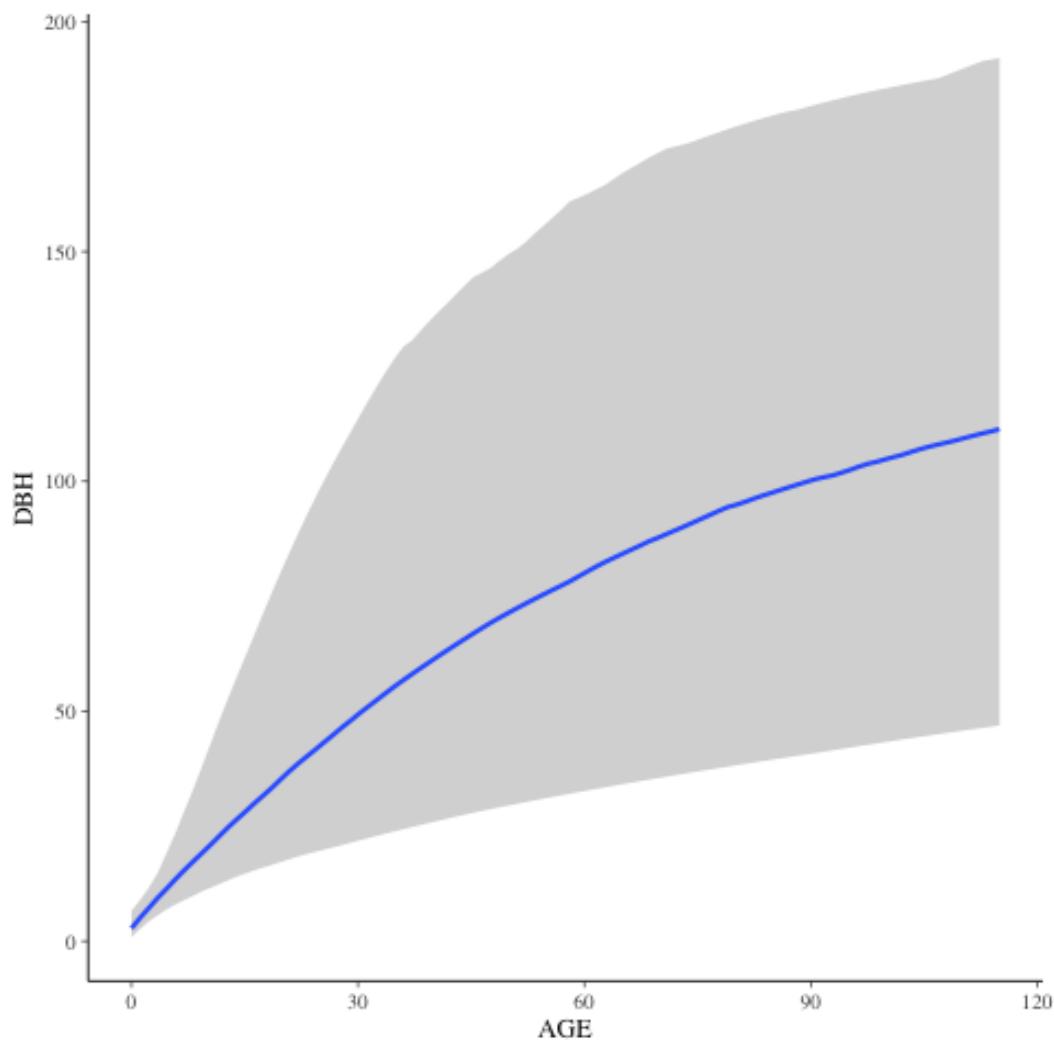
```
pred <- predict(prior_mod, newdata = d)
```

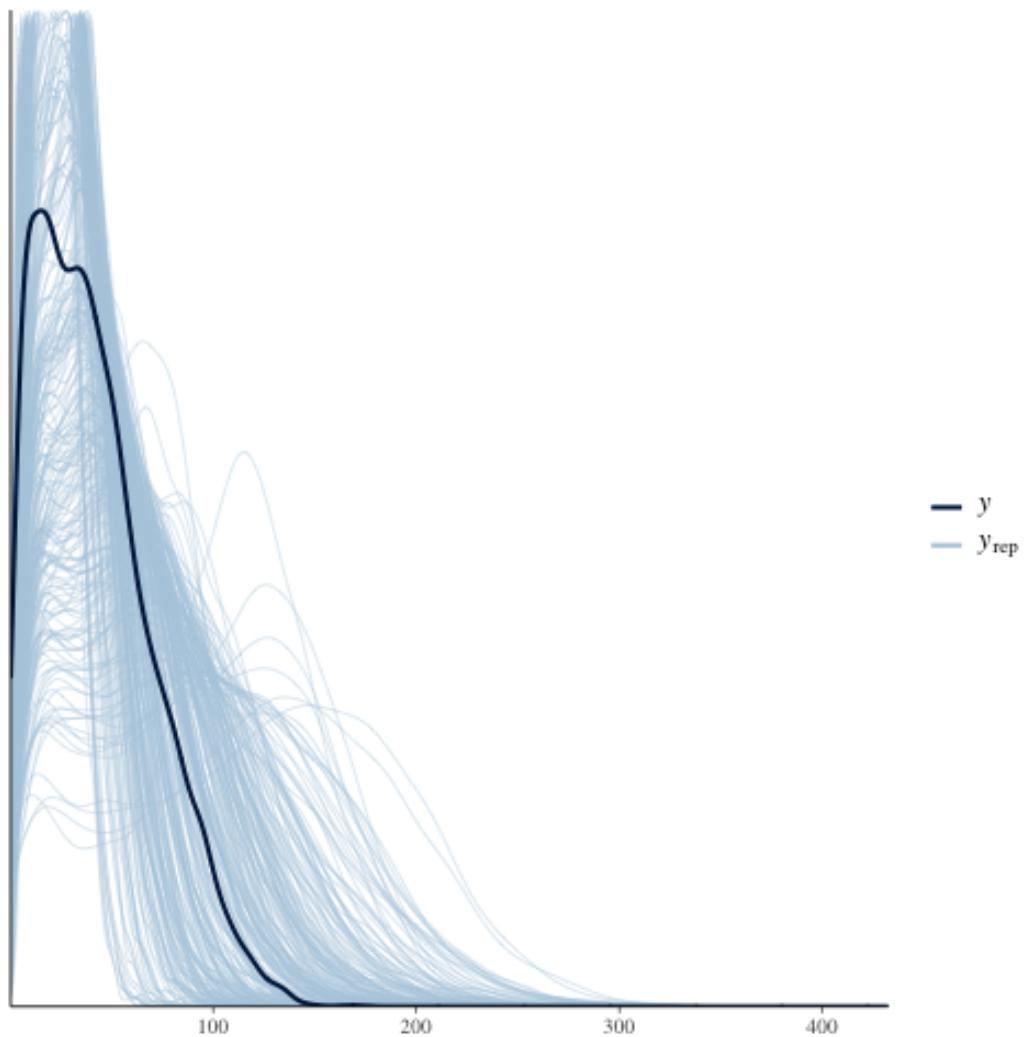
```
mod <- brm(form,  
data = d,  
prior = nlprior,  
family = Gamma("identity"),  
chains = 8, cores = 8, init_r = .3, iter = 2000)
```

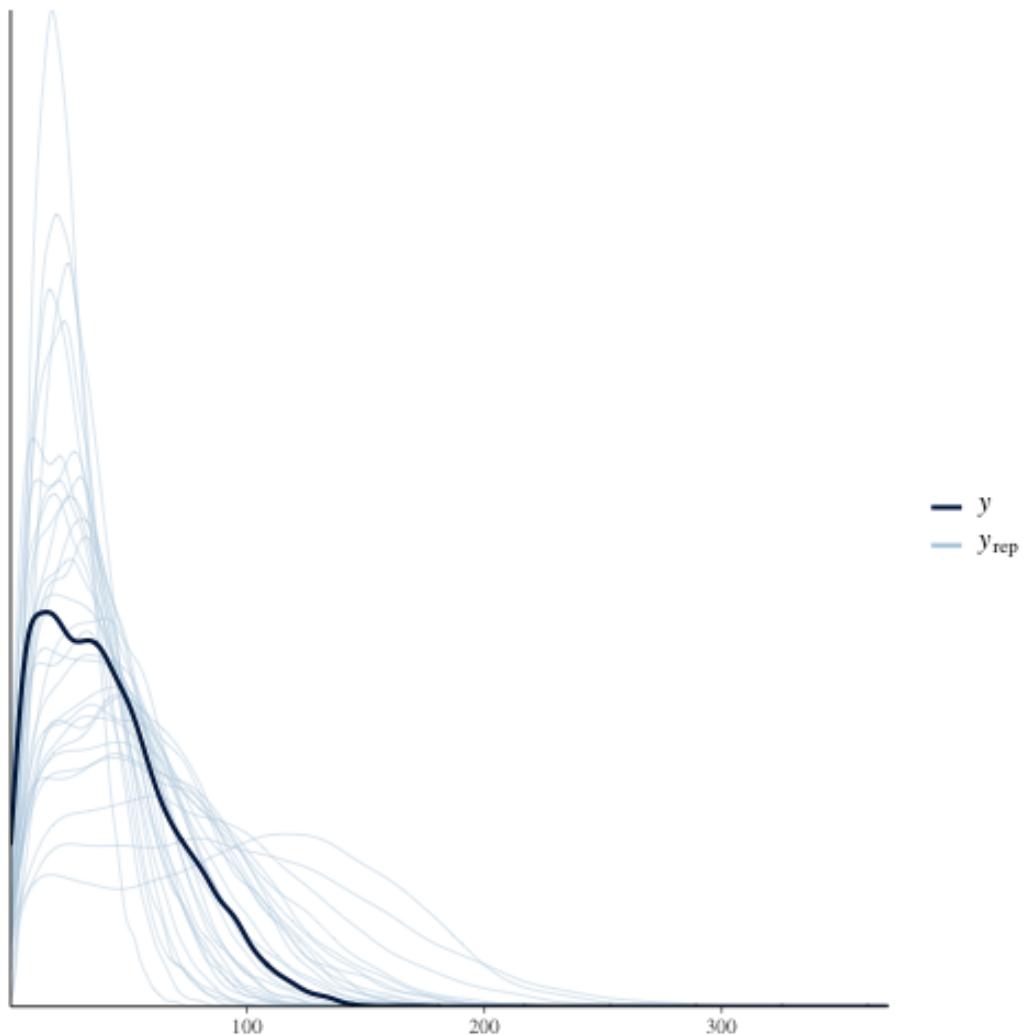
```
saveRDS(mod, paste0("../models/genus_",genus,"_species_",species,"_cities_", citi
```











- tangle C-c C-v t

- send to krusty

```
rsync -avz genus_none_species_single_cities_single_climate_none_hetero_no_family_ga
```

- 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_fam
exit
```

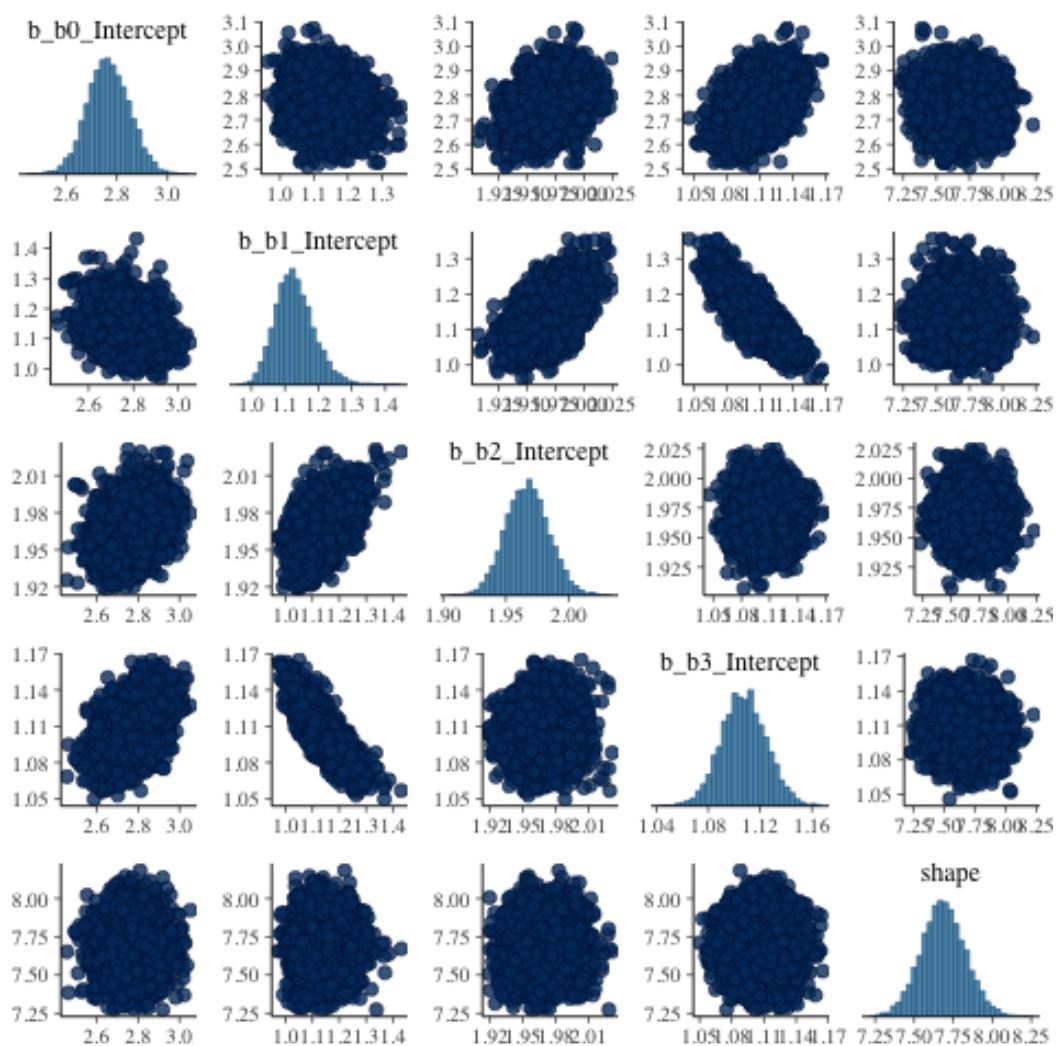
```
cat genus_none_species_single_cities_single_climate_none_hetero_no_family_gamma.Rou
```

- pull back from krusty

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

- diagnostics

```
mod_genus_none_species_single_cities_single_climate_none_hetero_no_family_Gamma <- r
mod <- mod_genus_none_species_single_cities_single_climate_none_hetero_no_family_Ga
```



```
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

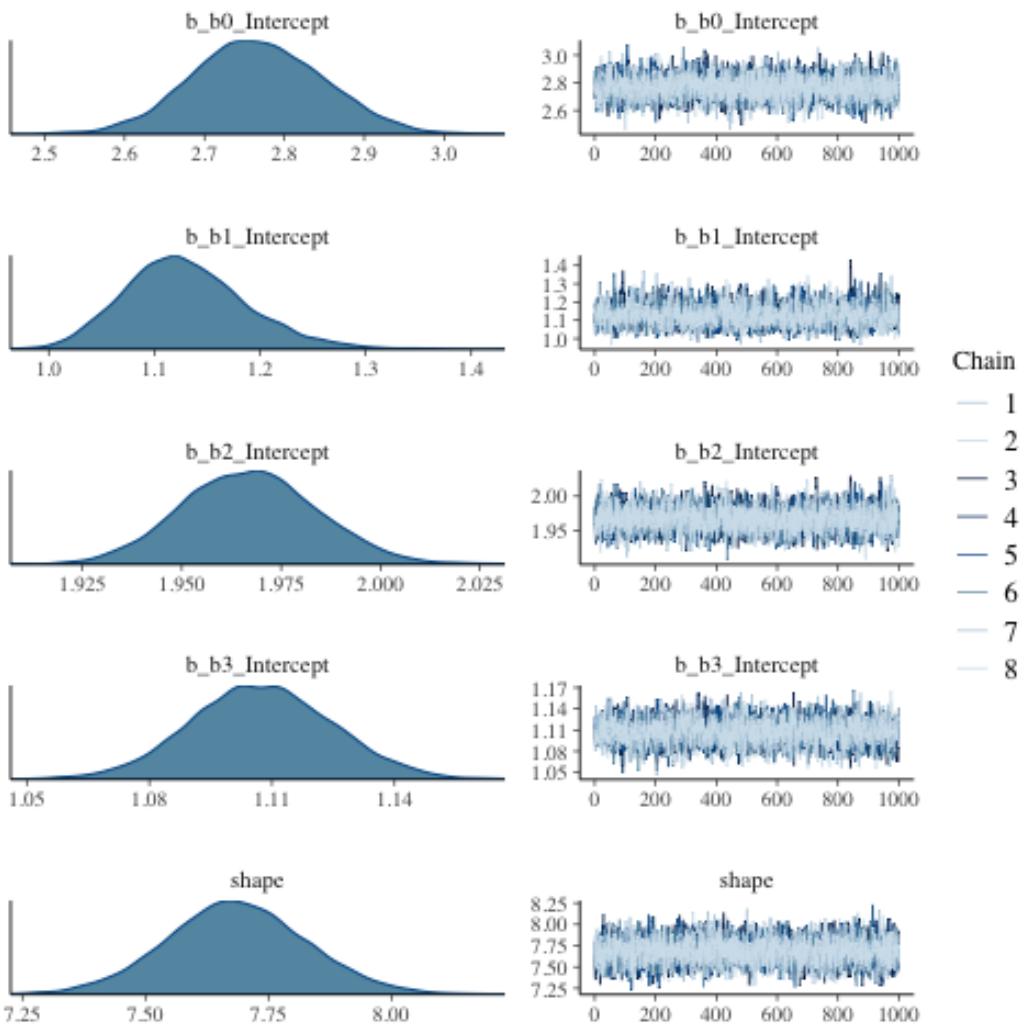
Population-Level Effects:

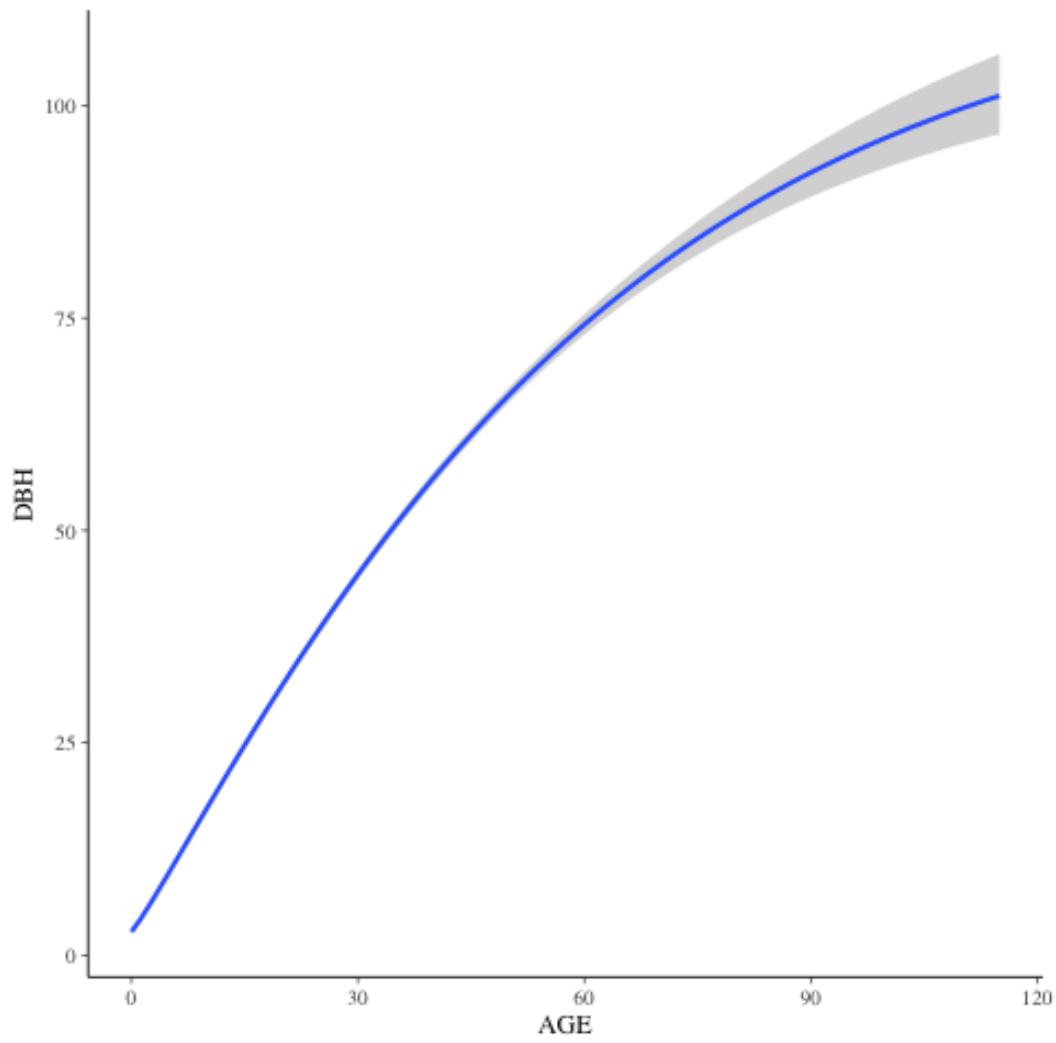
	Estimate	Est.Error	1-95% CI	u-95% CI	Eff.Sample	Rhat
b0_Intercept	2.77	0.08	2.61	2.93	3416	1.00
b1_Intercept	1.12	0.05	1.03	1.24	2694	1.00
b2_Intercept	1.08	0.04	1.00	1.17	3530	1.00
b3_Intercept	1.11	0.02	1.08	1.14	2806	1.00

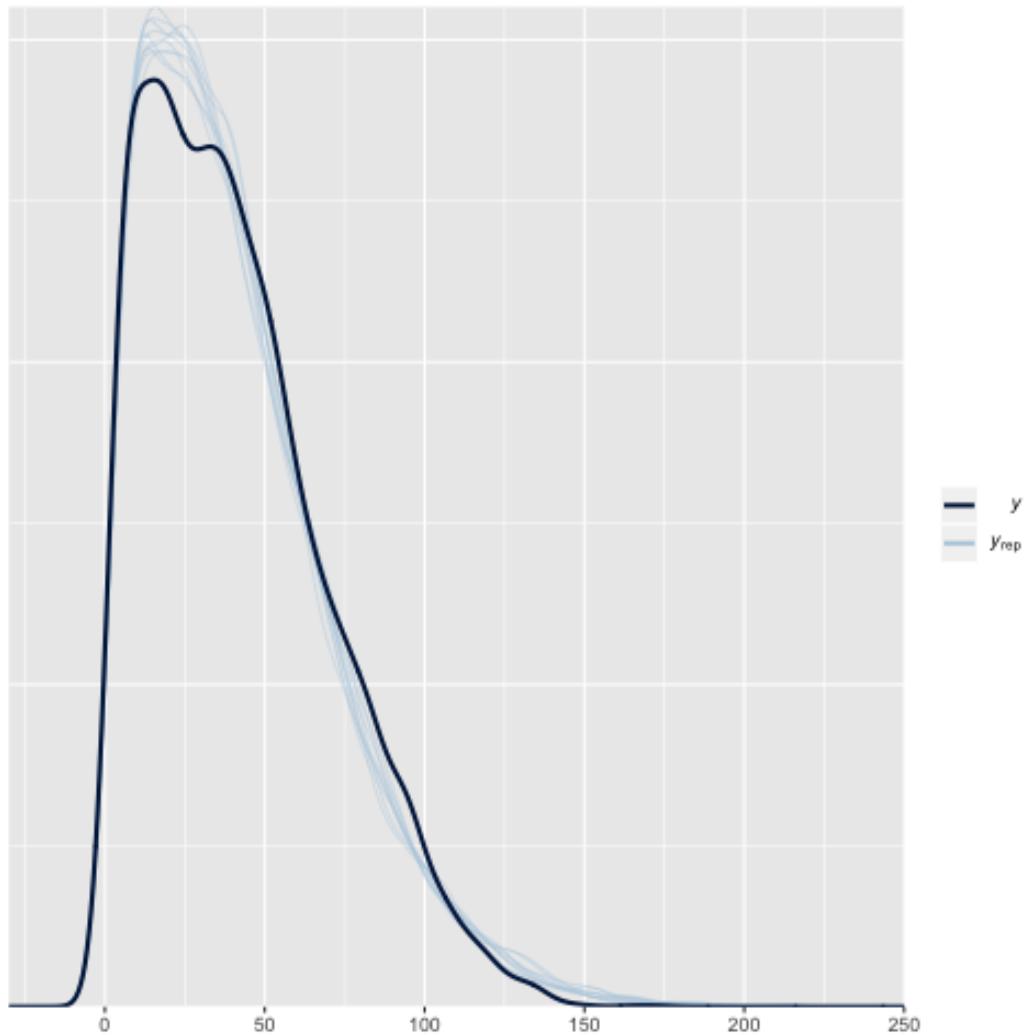
Family Specific Parameters:

	Estimate	Est.Error	1-95% 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).







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

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

```

- tangle
- send to krusty

```
rsync -avz genus_many_species_many_cities_many_climate_b1linint_hetero_no_family_Ga
```

- run on krusty

run from krusty terminal

```
ssh krusty
```

```
cd allo/code  
nohup R CMD BATCH genus_many_species_many_cities_many_climate_b1linint_hetero_no_fa  
exit  
  
cat genus_many_species_many_cities_many_climate_b1linint_hetero_no_family_Gamma.Rou
```

- pull back from krusty

```
rsync -avz erker@krusty:~/allo/models/genus_many_species_many_cities_many_climate_b1
```

- assess model

```
mod_genus_many_species_many_cities_many_climate_b1linint_hetero_no_family_Gamma <-  
mod <- mod_genus_many_species_many_cities_many_climate_b1linint_hetero_no_family_Ga
```

```
../figs/pairs_genus_many_species_many_cities_many_climate_b1linint_hetero_no_fa
```

```
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	l-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	l-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	l-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 l-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

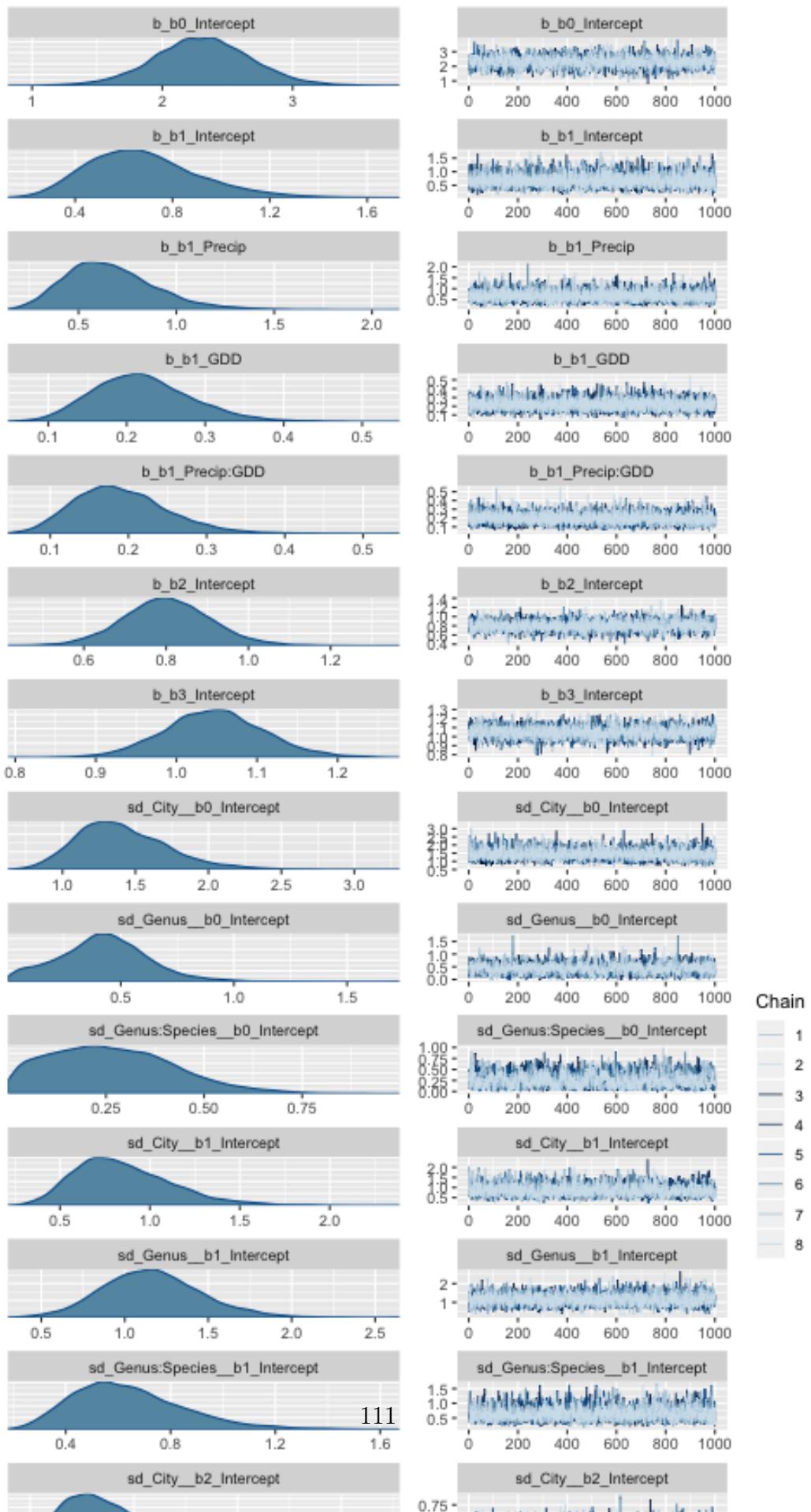
Family Specific Parameters:

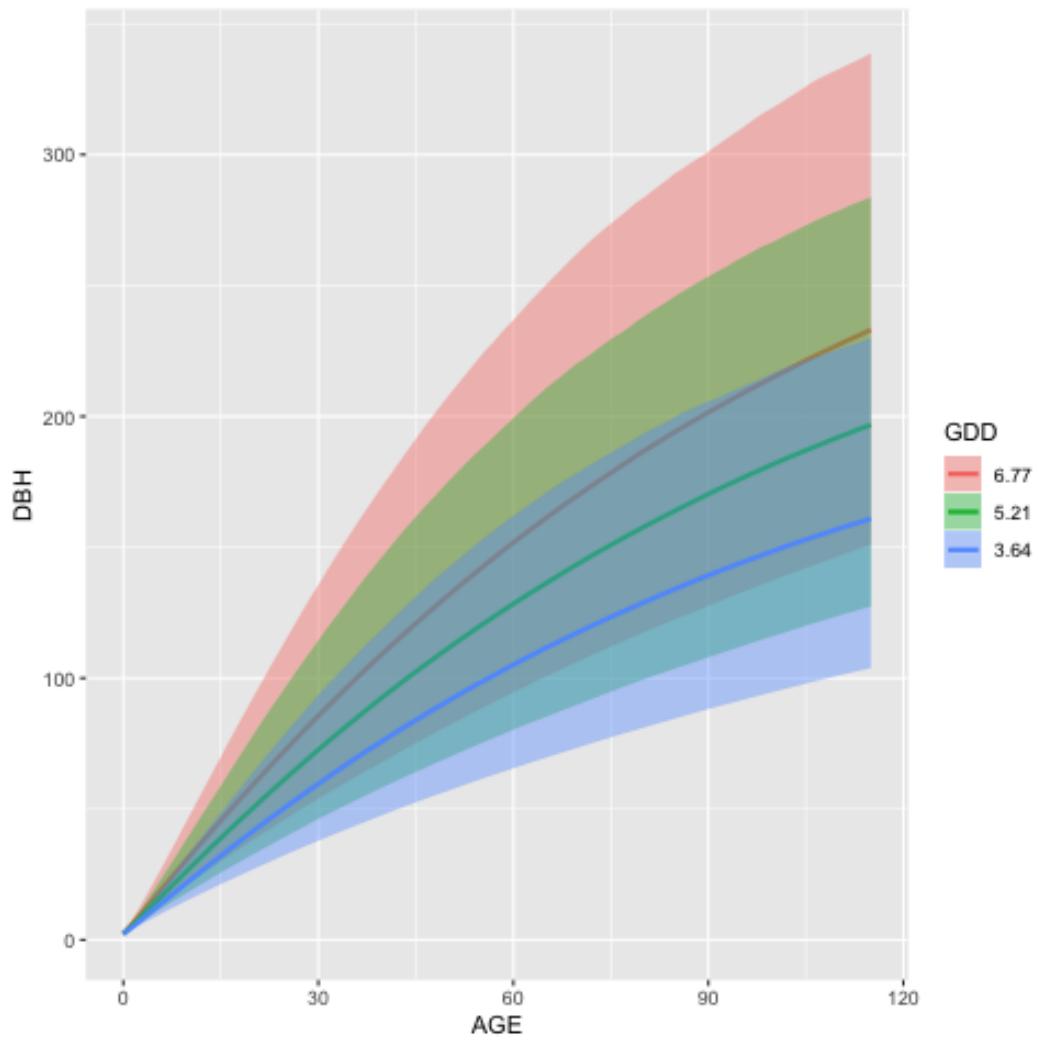
	Estimate	Est.Error	l-95% CI	u-95% CI	Eff.Sample	Rhat
shape	15.06	0.29	14.50	15.64	8000	1.00

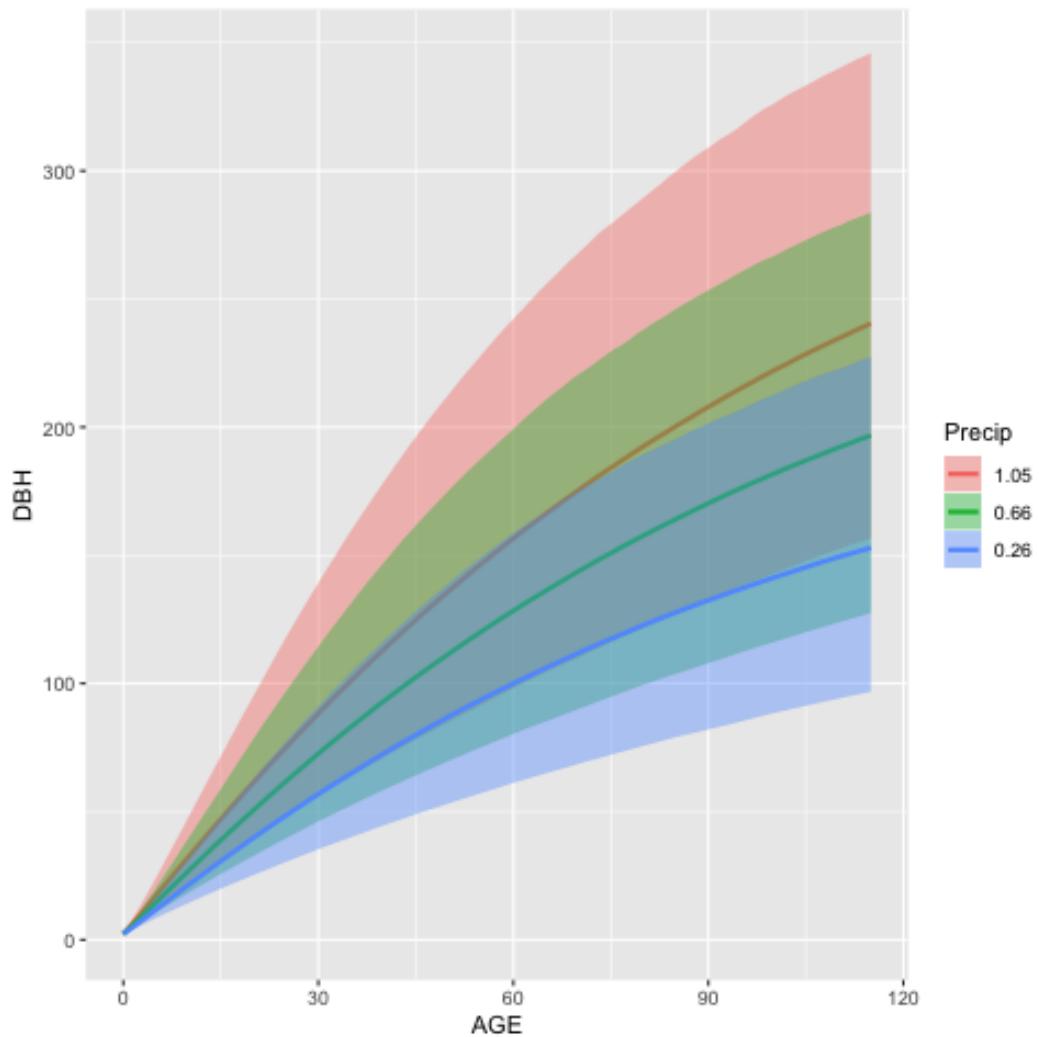
Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Warning message:

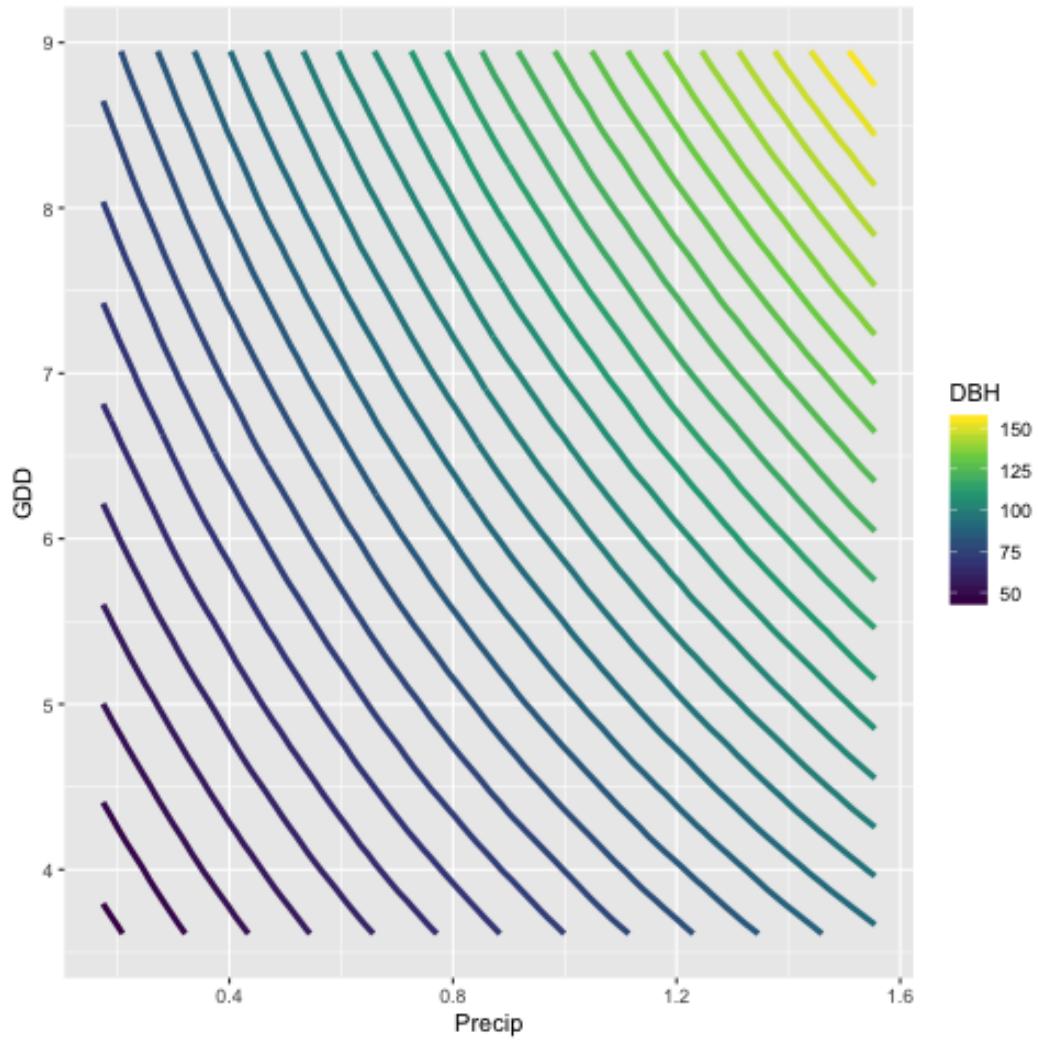
There were 3 divergent transitions after warmup. Increasing adapt_delta above 0.8 m
See <http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>

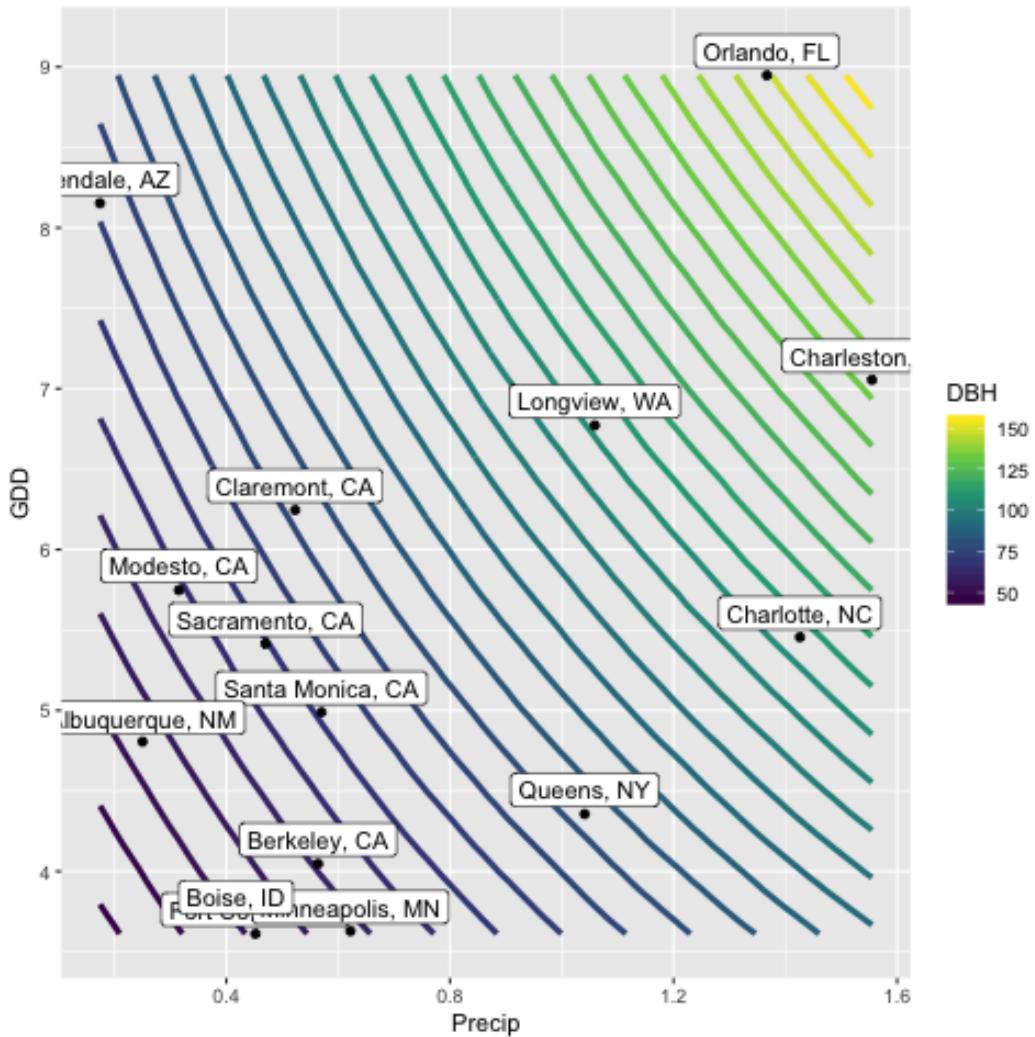






```
precip.gdd <- marginal_effects(mod, effects = "Precip:GDD", surface = T)
```

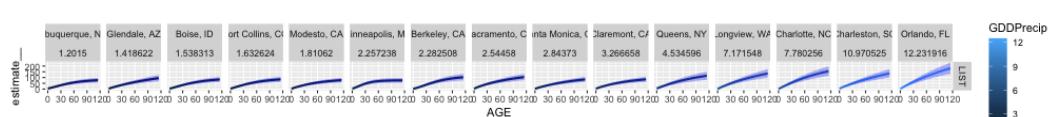


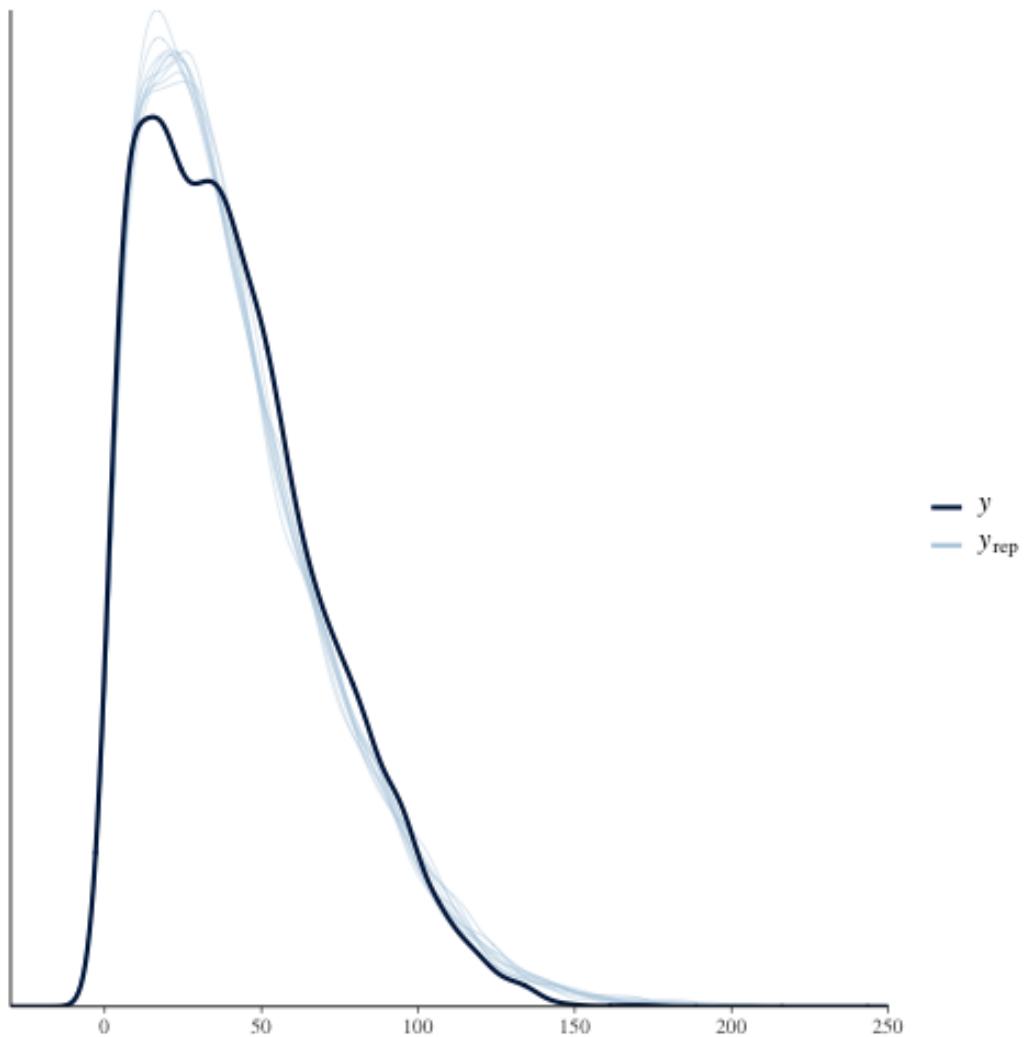


```

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 = N)
  
```





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

- 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))

```

- tangle
- send to krusty

```
rsync -avz genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gam
```

- 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_fa
exit

```

```
cat genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma.Rou
```

- pull back from krusty

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

- assess model

```
mod_genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gamma <-  
mod <- mod_genus_many_species_many_cities_many_climate_b3linint_hetero_no_family_Gam  
..../figs/pairs_genus_many_species_many_cities_many_climate_b3linint_hetero_no_fa
```

```
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	1-95% 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

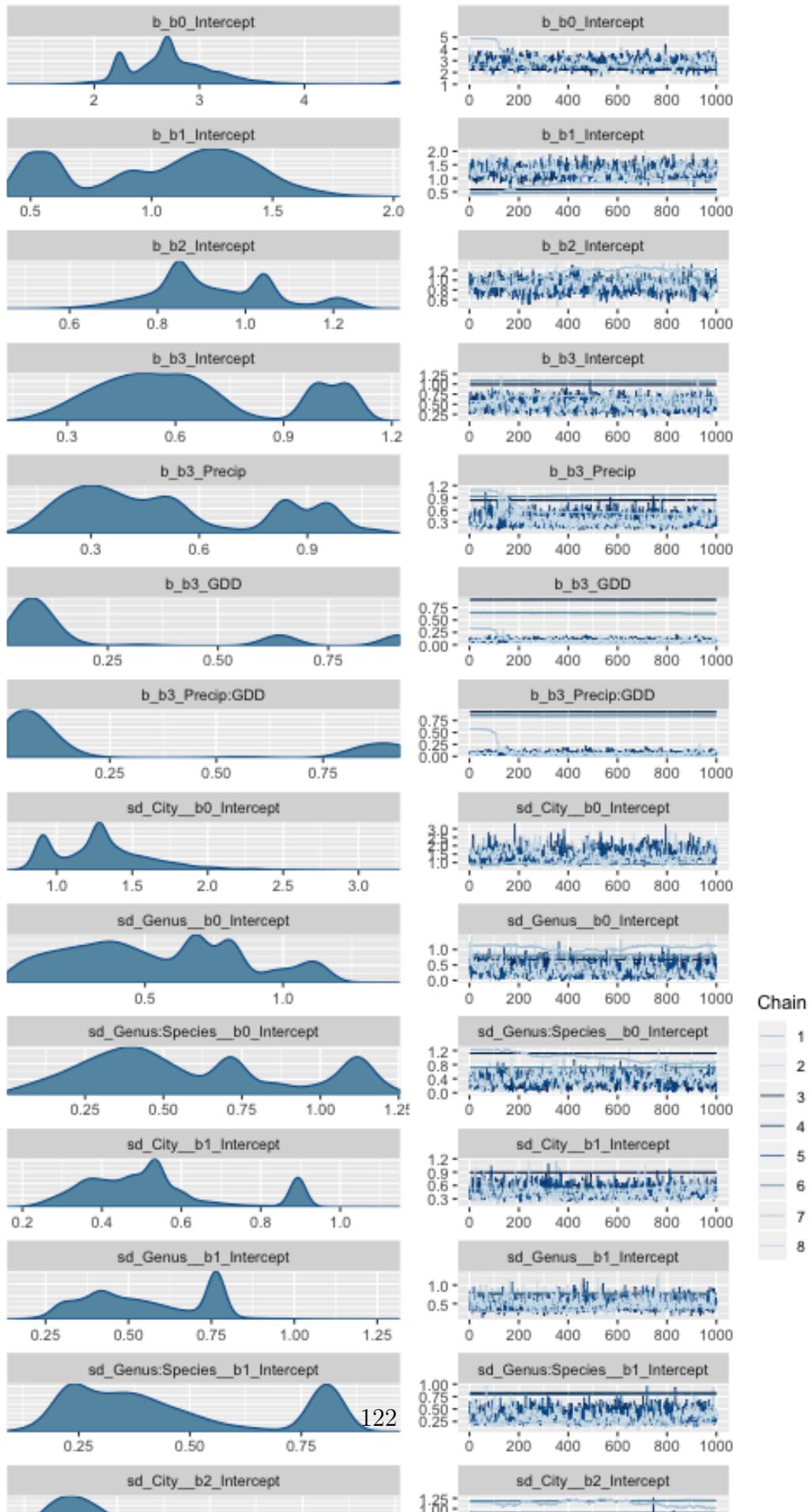
Family Specific Parameters:

	Estimate	Est.Error	1-95% 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
See <http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>



./figs/marginal_effects_AGEgDD_genus_many_species_many_cities_many_climate_b31

```
./figs/marginal_effects_AGEPrecip_genus_many_species_many_cities_many_climate_
```

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

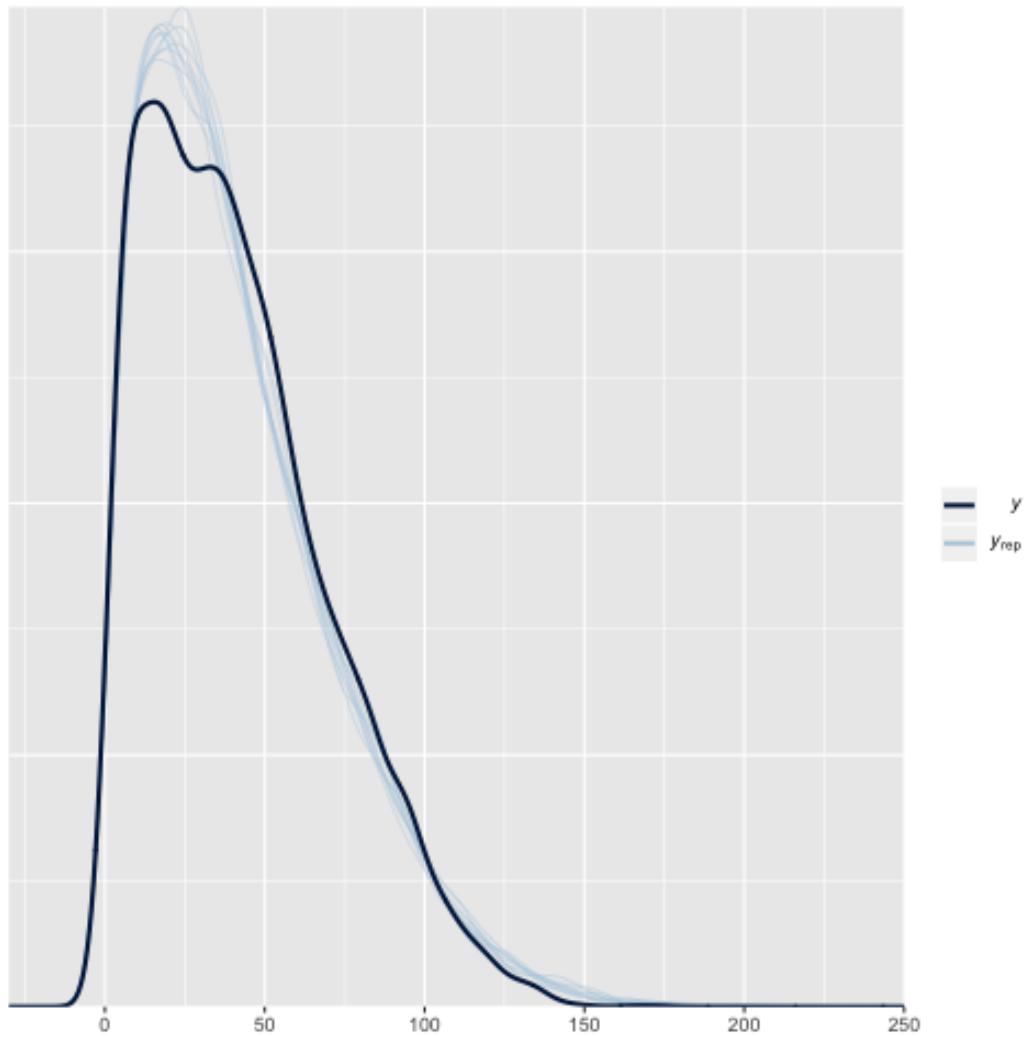
./figs/marginal_effects_surface_PrecipGDD_genus_many_species_many_cities_many_

```
../figs/marginal_effects_surface_PrecipCDD_genus_many_species_many_cities_many_
```

```
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 = N~
```

```
../figs/marginal_effects_BySpecies_Genus_many_species_many_cities_many_climate_
```



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)

for diagnostics

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

look at distributions

Priors

gamma alpha called shape beta called rate mean is alpha/beta; variance is alpha/beta²

