Multilevel Urban Tree Allometric equations

erker

October 12, 2018

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).

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.

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

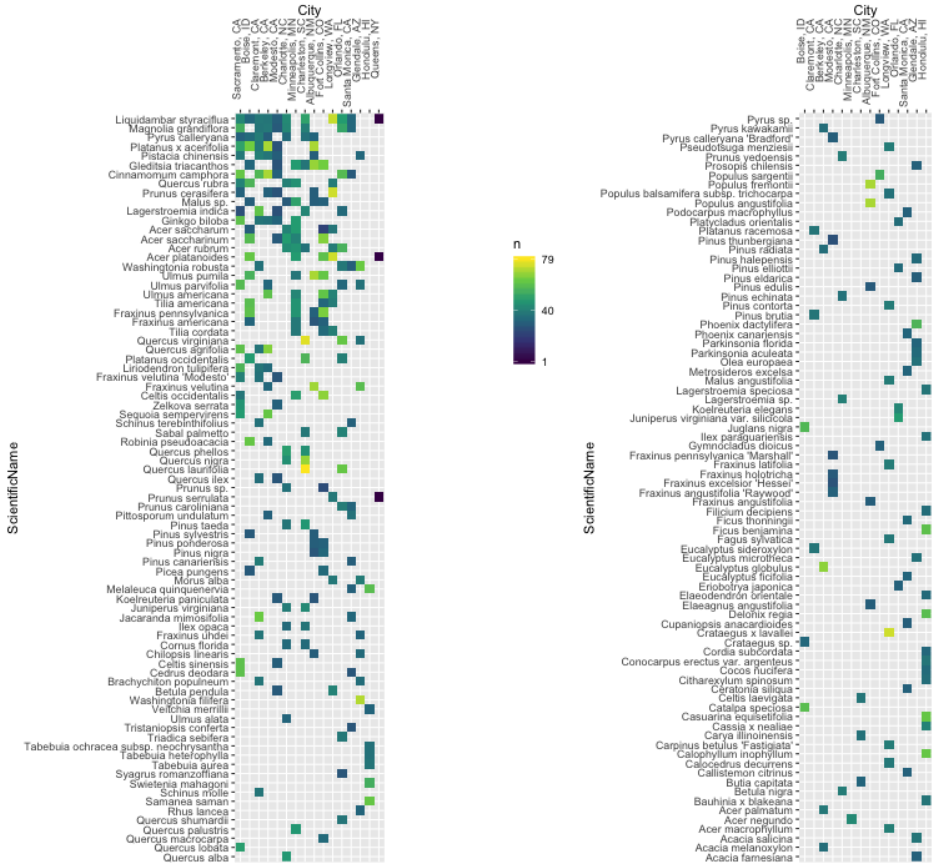


Figure 1: Number of trees sampled of each species and city combination in the urban tree database.

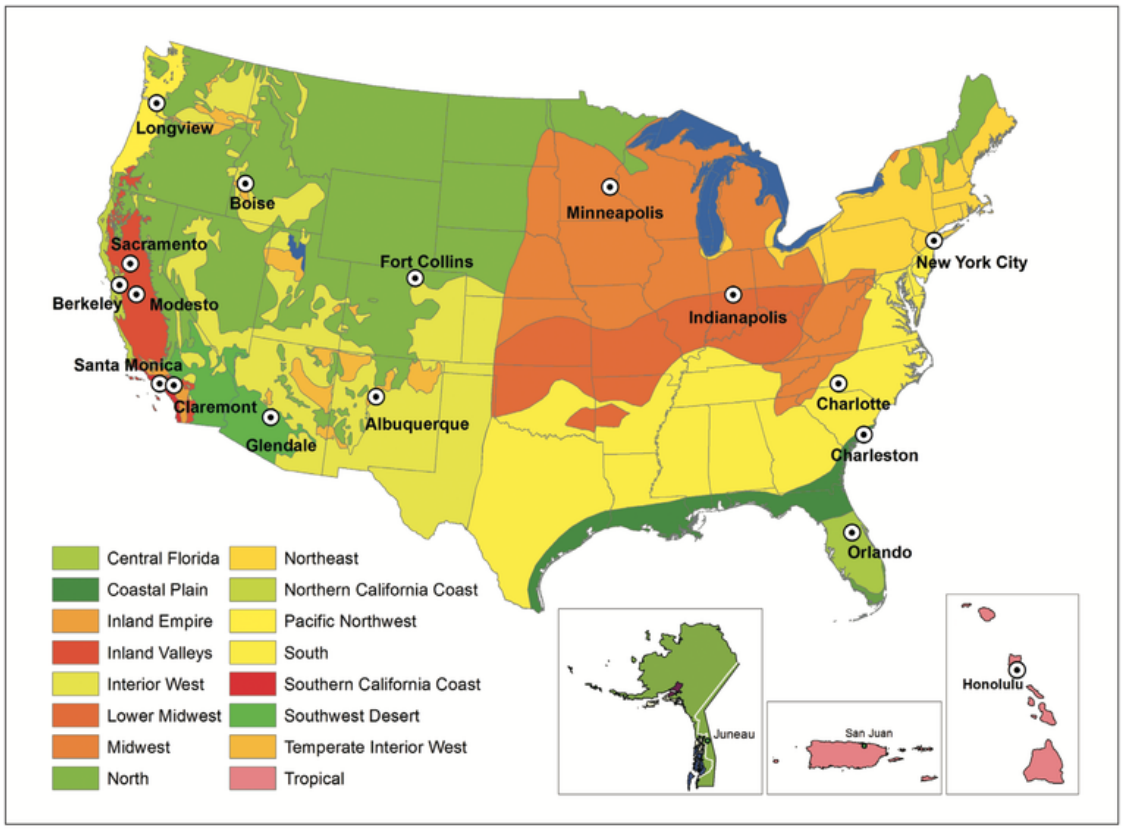


Figure 9—Climate zones were aggregated from 45 Sunset climate zones into 16 zones. Each zone has a reference city where tree growth data were collected. Sacramento, California, was added as a second reference city (with Modesto) to the Inland Valleys zone.

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

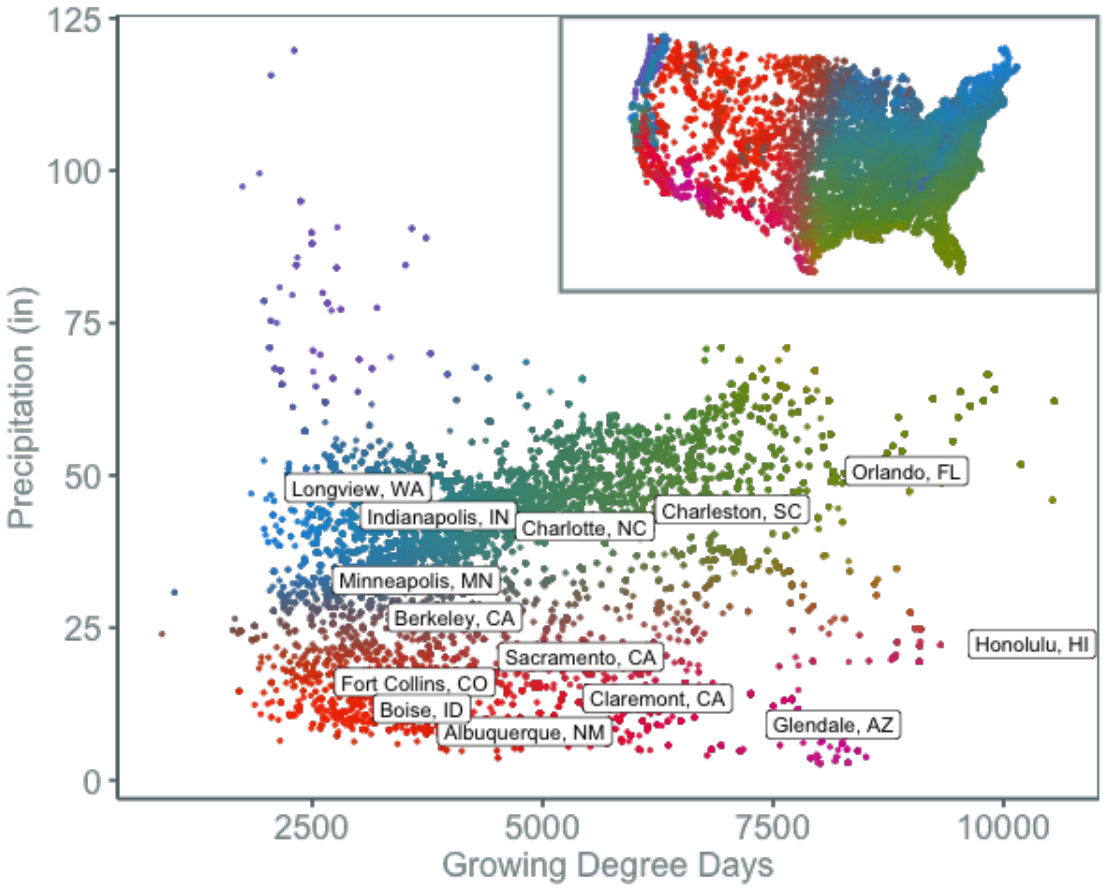


Figure 3: US census tract centroids with UTD reference cities overlaid in growing degree day (GDD) and precipitation climate space and matching color gradient in geographic space. The reference cities cover climate space well, and variation in precipitation and growing degree days is continuous.

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" it 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

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" "Mot
 $ 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" "Cinnamomu
 $ CommonName    : chr  "Silver maple" "European white birch" "Chinese hackberry" "Camph
 $ 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 ...
$ TreeOr        : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ CarShade      : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ LandUse       : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ Shape         : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ WireConf      : int  -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 ...
$ dbh3          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh4          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh5          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh6          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh7          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
$ dbh8          : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
- attr(*, "problems")=Classes ‘tbl_df’, ‘tbl’ and 'data.frame': 24255 obs. of  5 variab
 ..$ 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"
```

explanation of variables from metadata

DbaseID = Unique id number for each tree.

Region = 16 U.S. climate regions, abbreviations are used (see $1_{\text{Regionalinformation.csv}}$).

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 $2_{\text{Regionalspeciesandcounts.csv}}$ 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 cerasifer
d$ScientificName[d$ScientificName == "Pyrus calleryana cvs."] <- "Pyrus calleryana"
```

tidy a few names and select variables of interest here

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

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 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" "Cinnamomu
 - attr(*, "problems")=Classes ‘tbl_df’, ‘tbl’ and ’data.frame’: 24255 obs. of  5 variab
  ..$ 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"
```

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 = 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	Cupaniopsis anacardioides
39	Delonix regia
40	Elaeagnus angustifolia
41	Elaeodendron orientale
42	Eriobotrya japonica
43	Eucalyptus ficifolia
44	Eucalyptus globulus
45	Eucalyptus microtheca
46	Eucalyptus sideroxylon
47	Fagus sylvatica
48	Ficus benjamina
49	Ficus thonningii
50	Filicium decipiens
51	Fraxinus americana
52	Fraxinus angustifolia
53	Fraxinus angustifolia 'Raywood'
54	Fraxinus excelsior 'Hessei'
55	Fraxinus holotricha
56	Fraxinus latifolia
57	Fraxinus pennsylvanica
58	Fraxinus pennsylvanica 'Marshall'
59	Fraxinus uhdei
60	Fraxinus velutina
61	Fraxinus velutina 'Modesto'
62	Ginkgo biloba
63	Gleditsia triacanthos
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	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

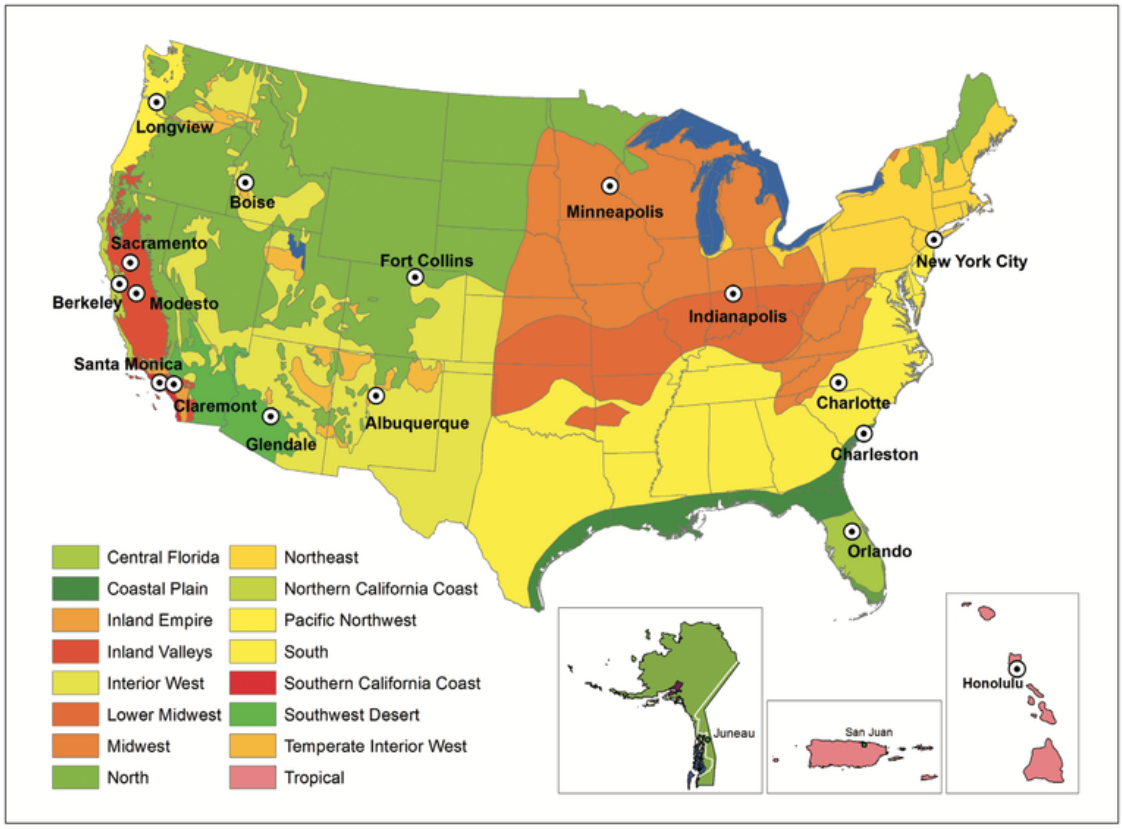


Figure 9—Climate zones were aggregated from 45 Sunset climate zones into 16 zones. Each zone has a reference city where tree growth data were collected. Sacramento, California, was added as a second reference city (with Modesto) to the Inland Valleys zone.

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 greeen 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... definitley 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/supplement
wget -O ../data/precip.txt https://www1.ncdc.noaa.gov/pub/data/normals/1981-2010/product
wget -O ../data/temp-station-info.txt https://www1.ncdc.noaa.gov/pub/data/normals/1981-2
```

```
  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(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)) %>%
      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)
```

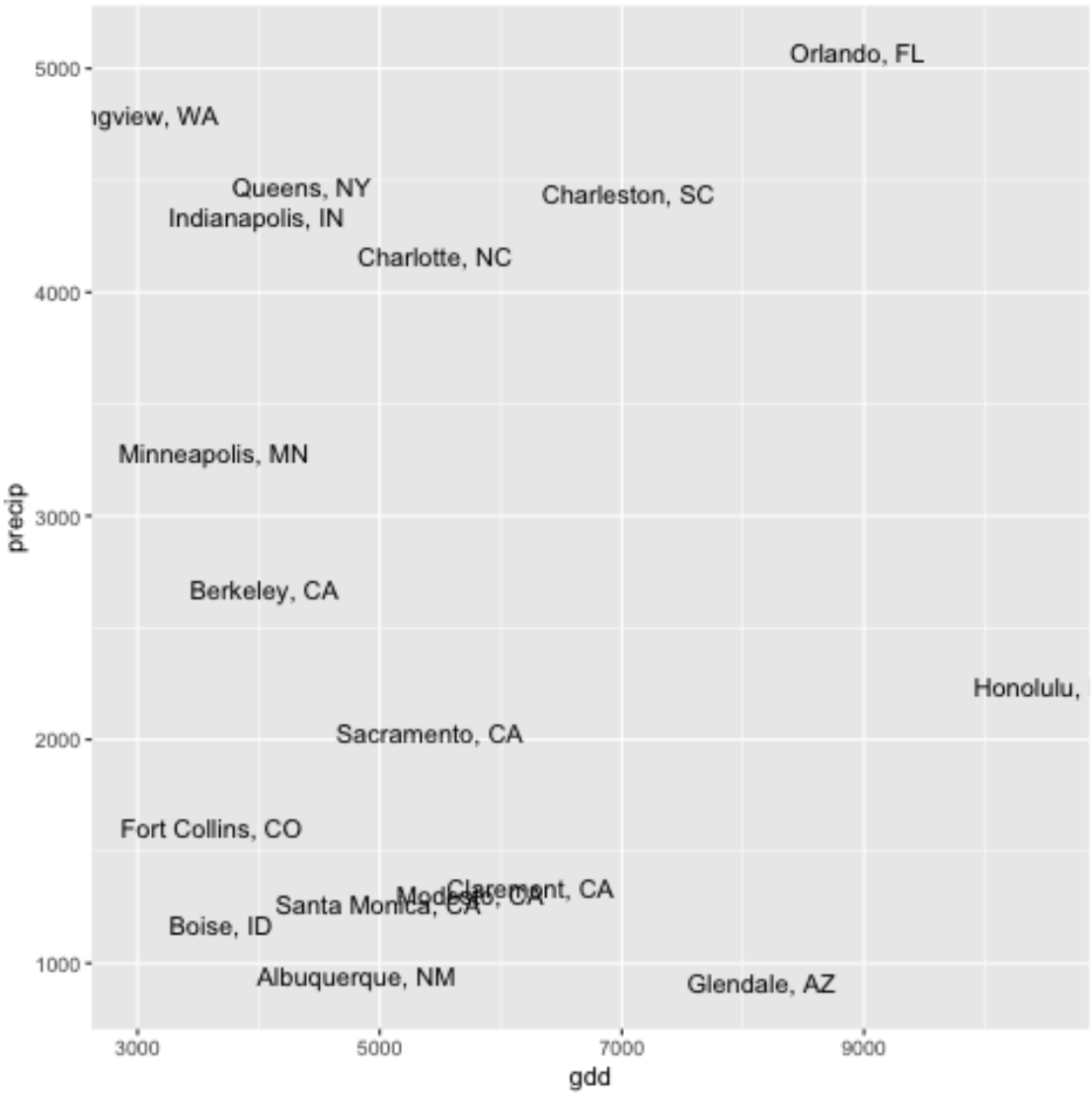
```
  dim(p)

  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
```

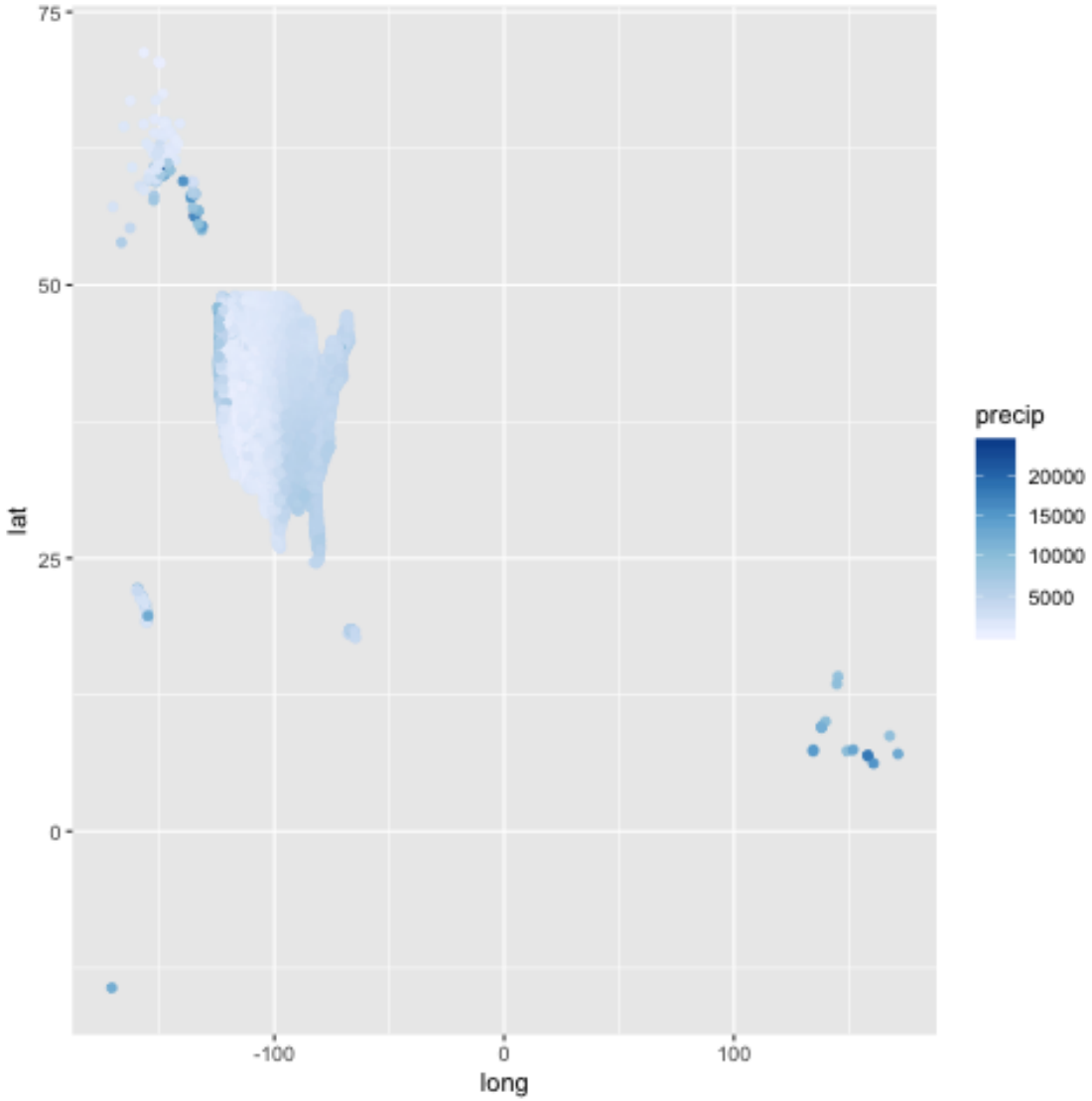
```
3 CAW00064757       3172Q   3172    Q
4 CQC00914080       8339P   8339    P
5 CQC00914801       9124R   9124    R
6 CQC00914855       6976P   6976    P
[1] 9307    4
[1] 7440    4
Joining, by = "station"
```

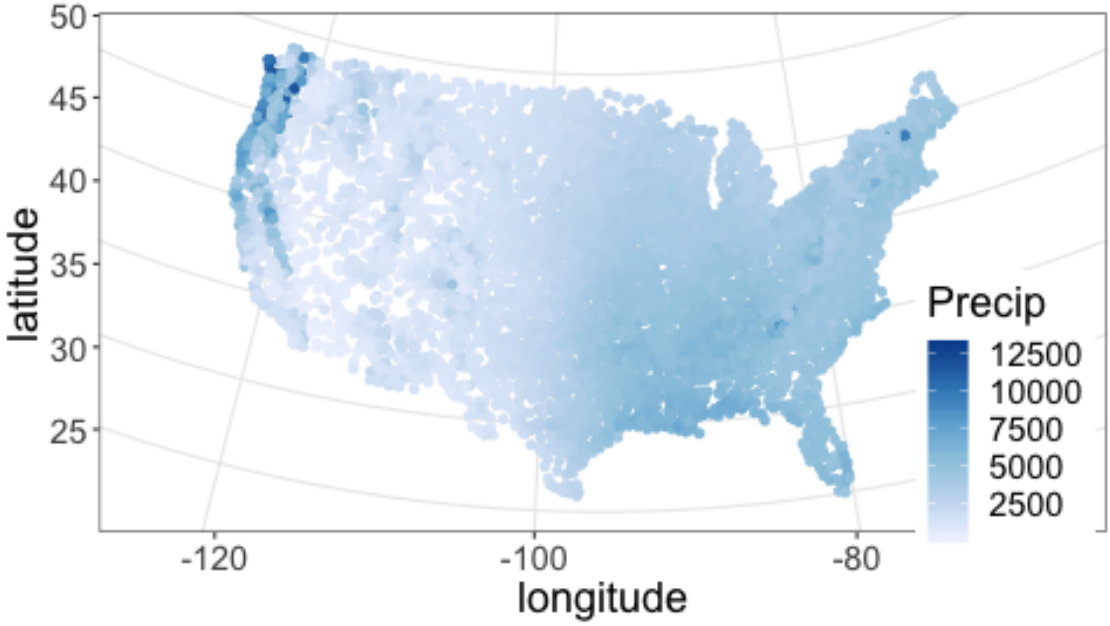
:var $\text{city}_{\text{climate}}$=$\text{city}_{\text{climate}}$

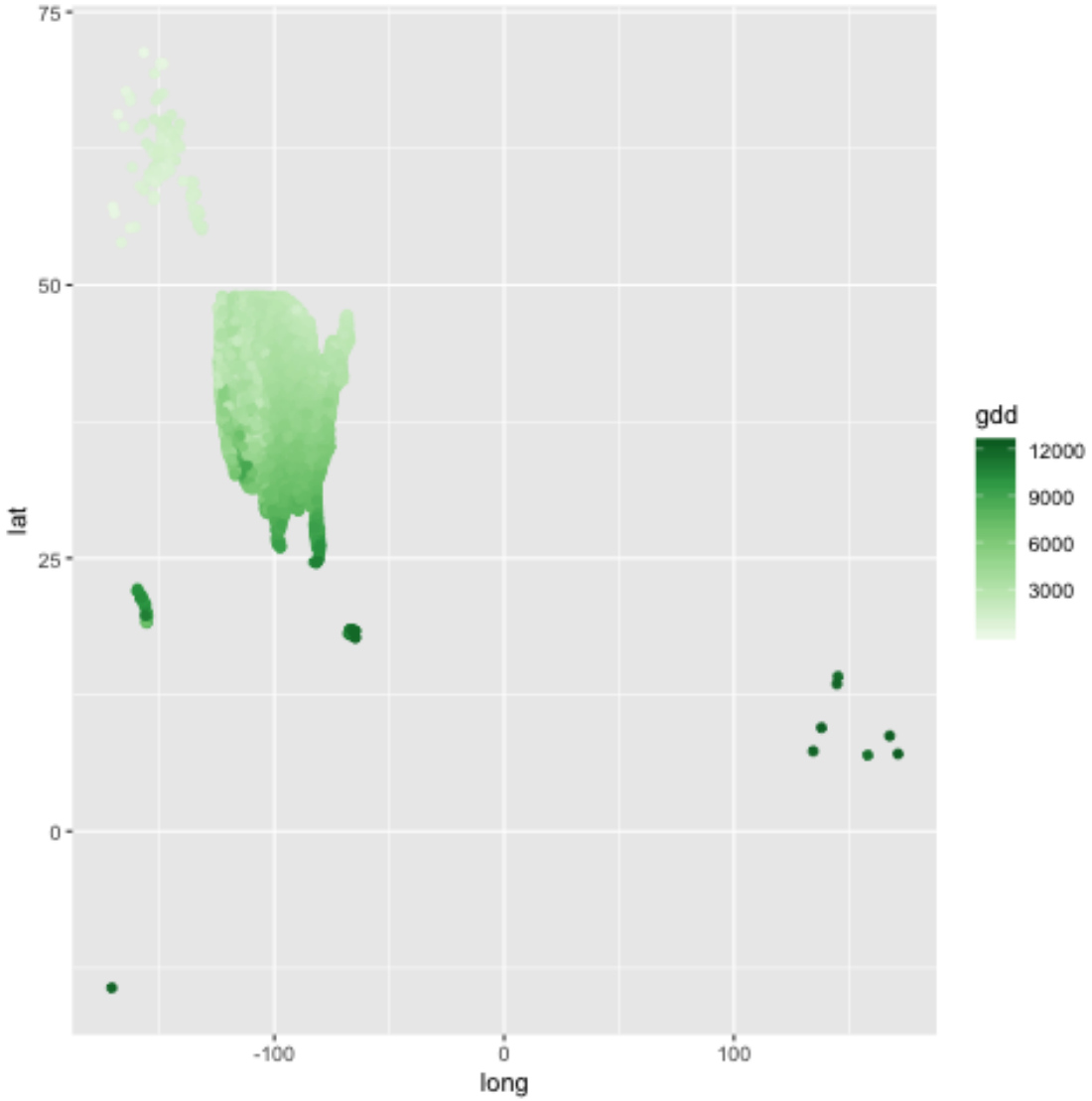
```
      library(sp)
        library(jsonlite)
      library(dplyr)

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])
```

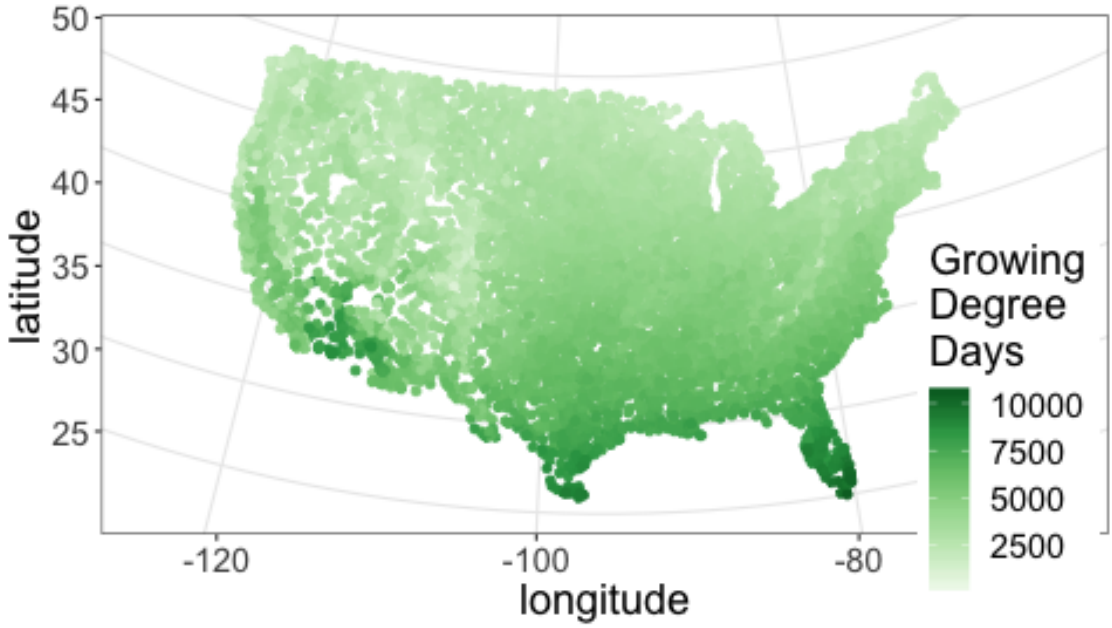
```
  gdd <- gdd[gdd.dists.min,]

  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")
```


Orlando, FL
ngview, WA
Queens, NY
Indianapolis, IN
Charleston, SC
Charlotte, NC
Minneapolis, MN
Berkeley, CA
Honolulu,
Sacramento, CA
Fort Collins, CO
Claremont, CA
Santa Monica, CA
Boise, ID
Albuquerque, NM
Glendale, AZ
precip
5000
4000
3000
2000
1000
3000
5000
7000
9000
gdd


75
50
25
0
lat
-100
0
100
long
precip
20000
15000
10000
5000


50
45
40
35
30
25
latitude
-120
-100
-80
longitude
Precip
12500
10000
7500
5000
2500


75
50
25
0
lat
-100
0
100
long
gdd
12000
9000
6000
3000



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)
```

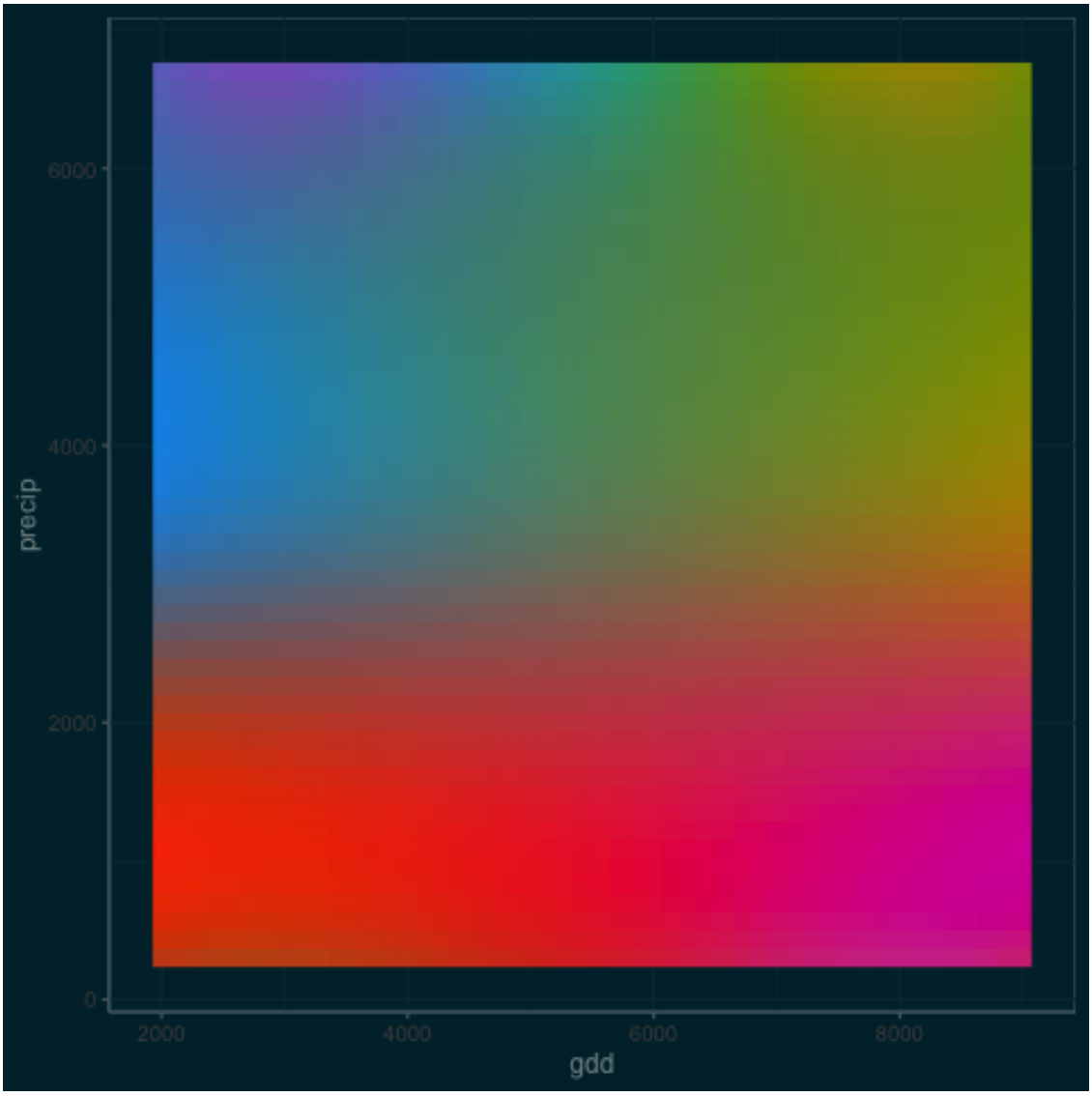
```
  library(dplyr)
  library(tidyr)
  library(ggplot2)
    library(reshape2)

    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")
```



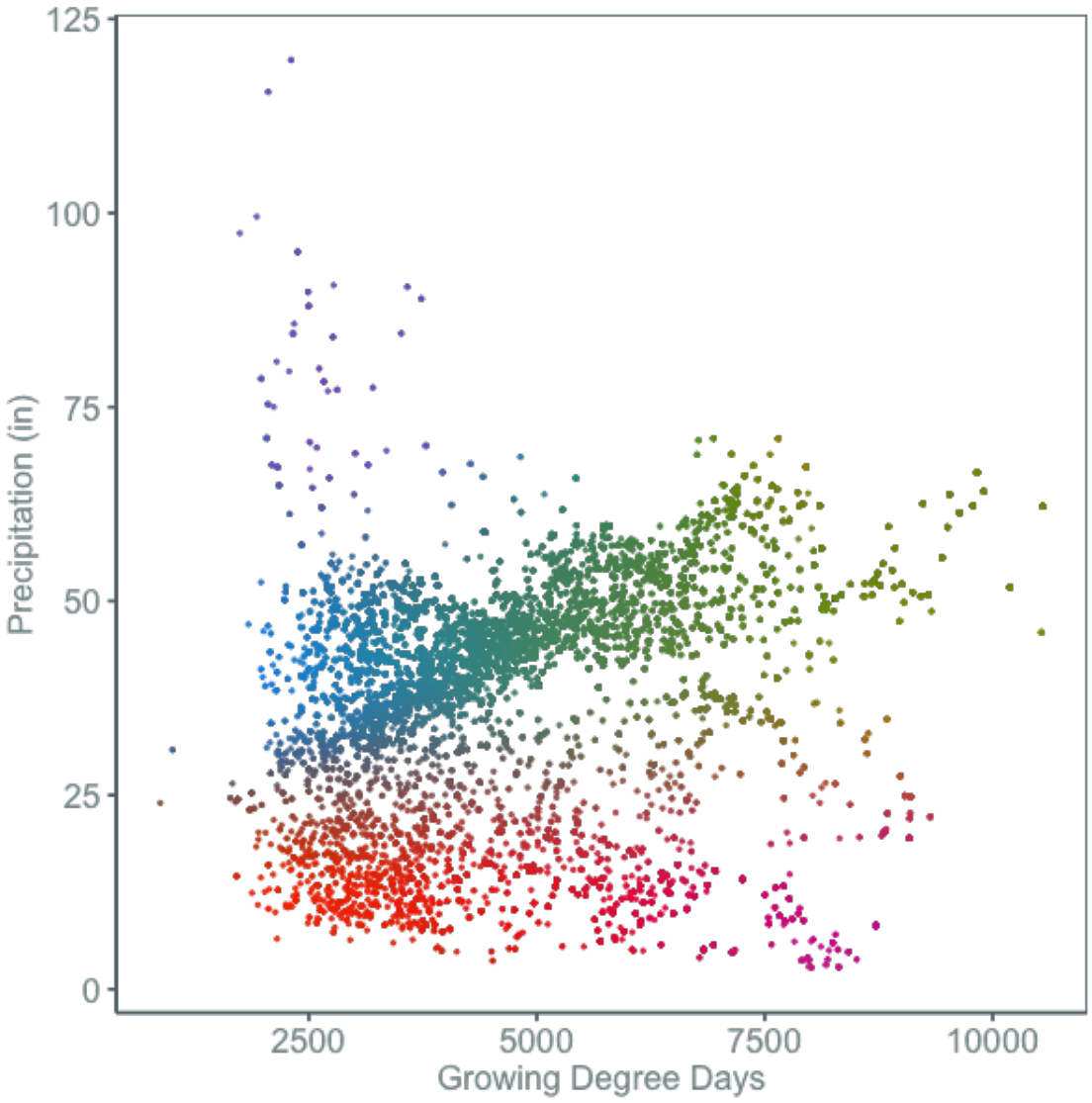

```
    library(sp)
```

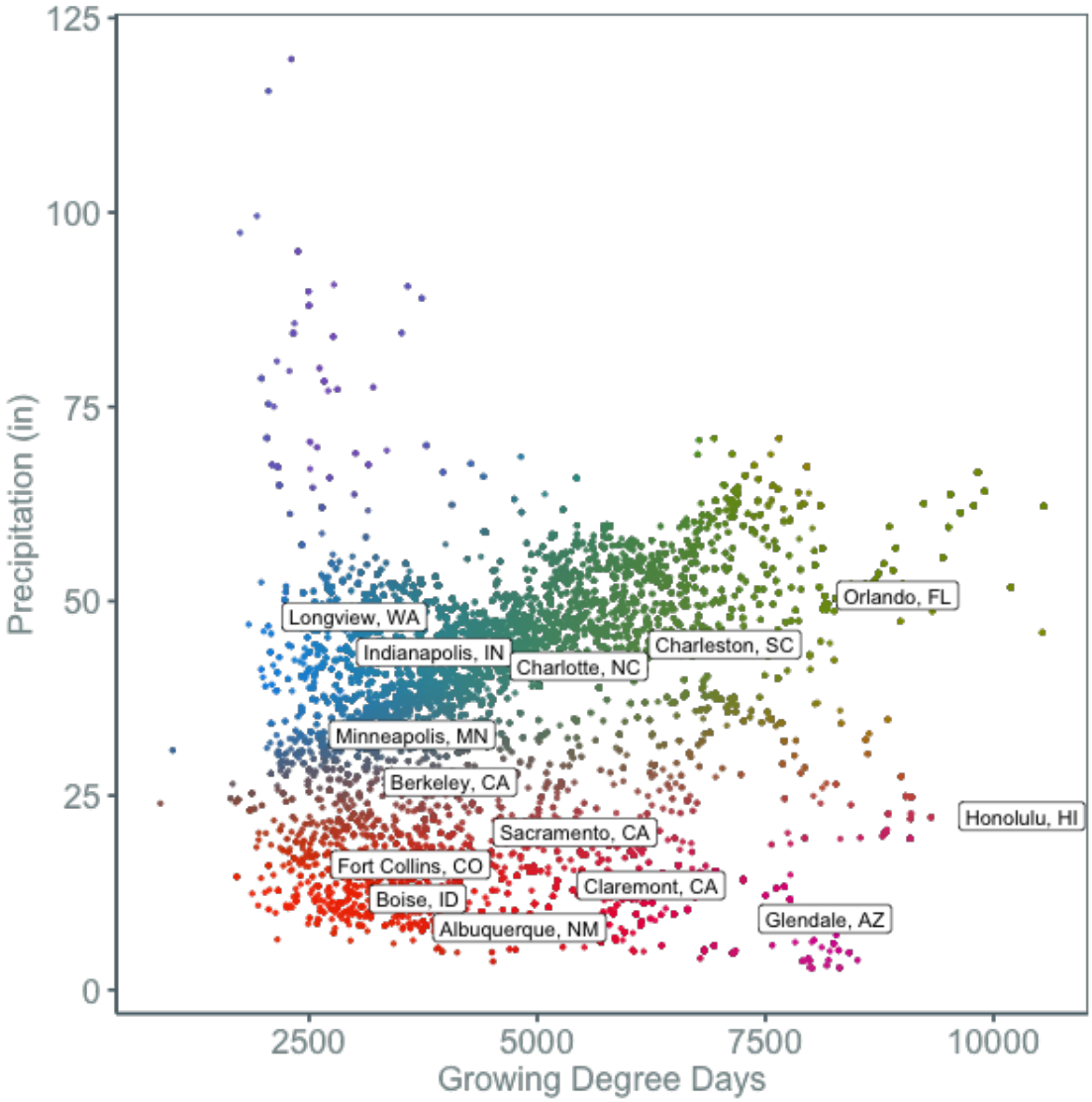
```
    library(reshape2)
  library(dplyr)
  library(tidyr)
  library(ggplot2)


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)
```

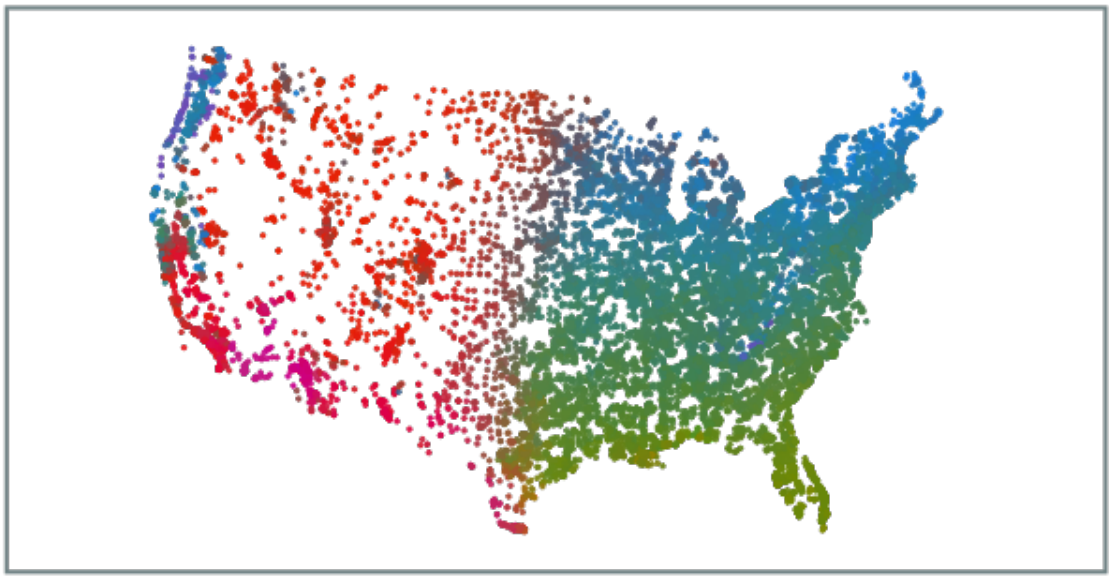
```
  ddn <- cbind(dn, precip_gdd_closest_value)
```

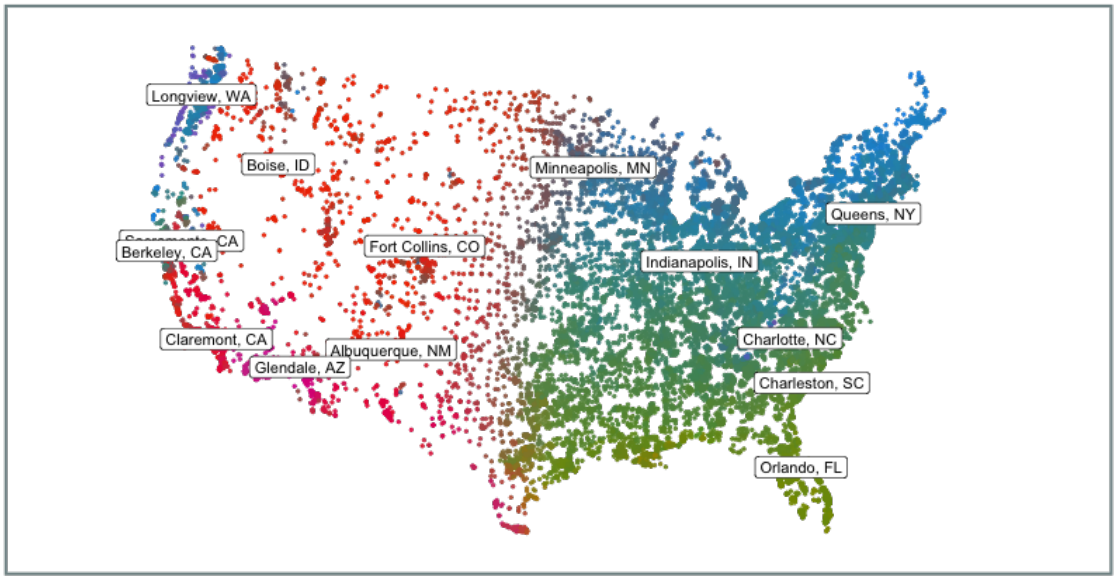
```
Joining, by = c("gdd_col", "precip_col")
[1] 38487     7
[1] 38487     2
```






```
## 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))
```

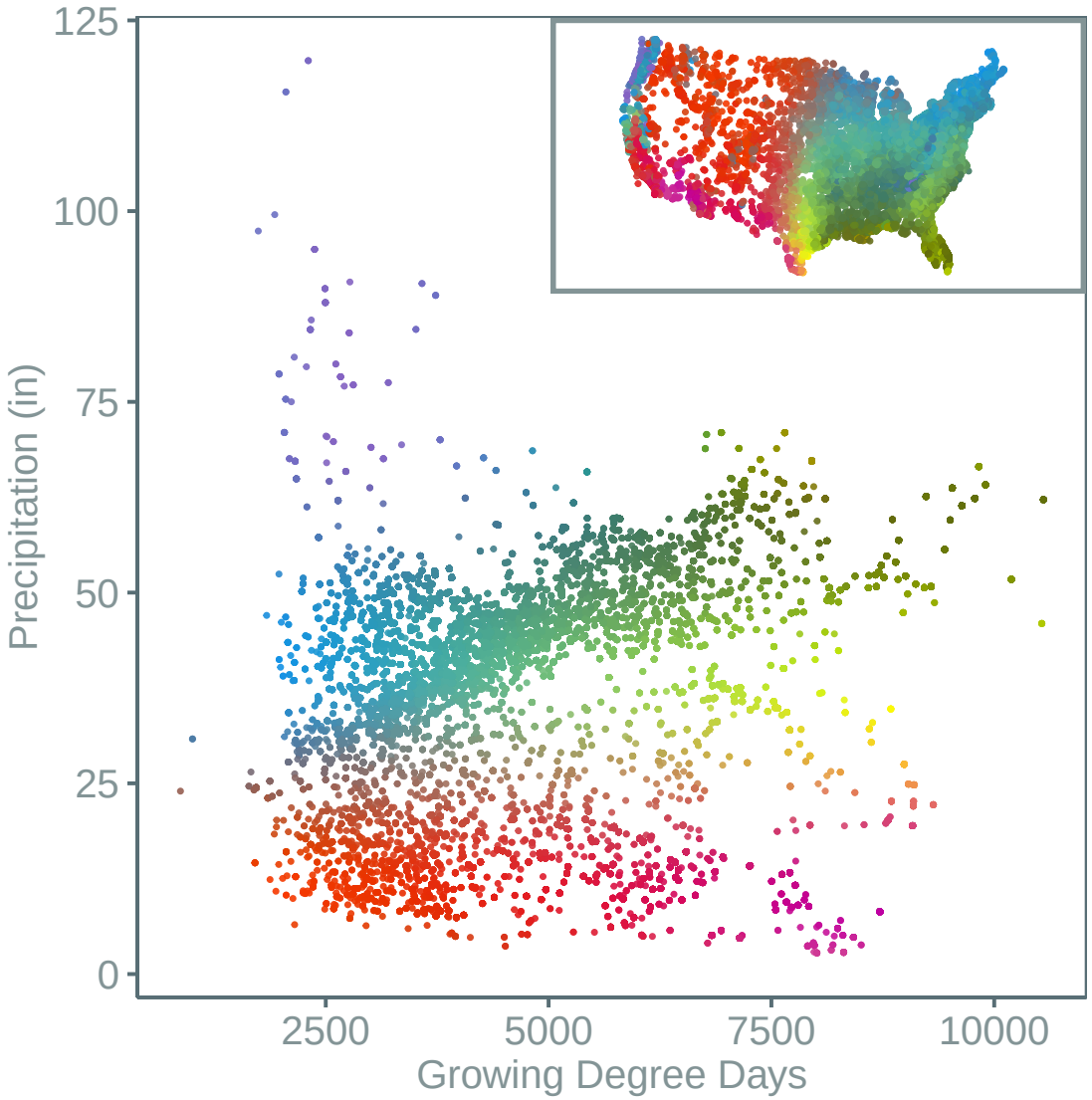





```
library(grid)

vp1 <- viewport(width = 0.5, height = 0.35, x = 0.74, y = 0.855)


#Just draw the plot twice
```

```
png("../figs/climate_space_wMap.png")
print(p)
print(mp, vp = vp1)
dev.off()
```

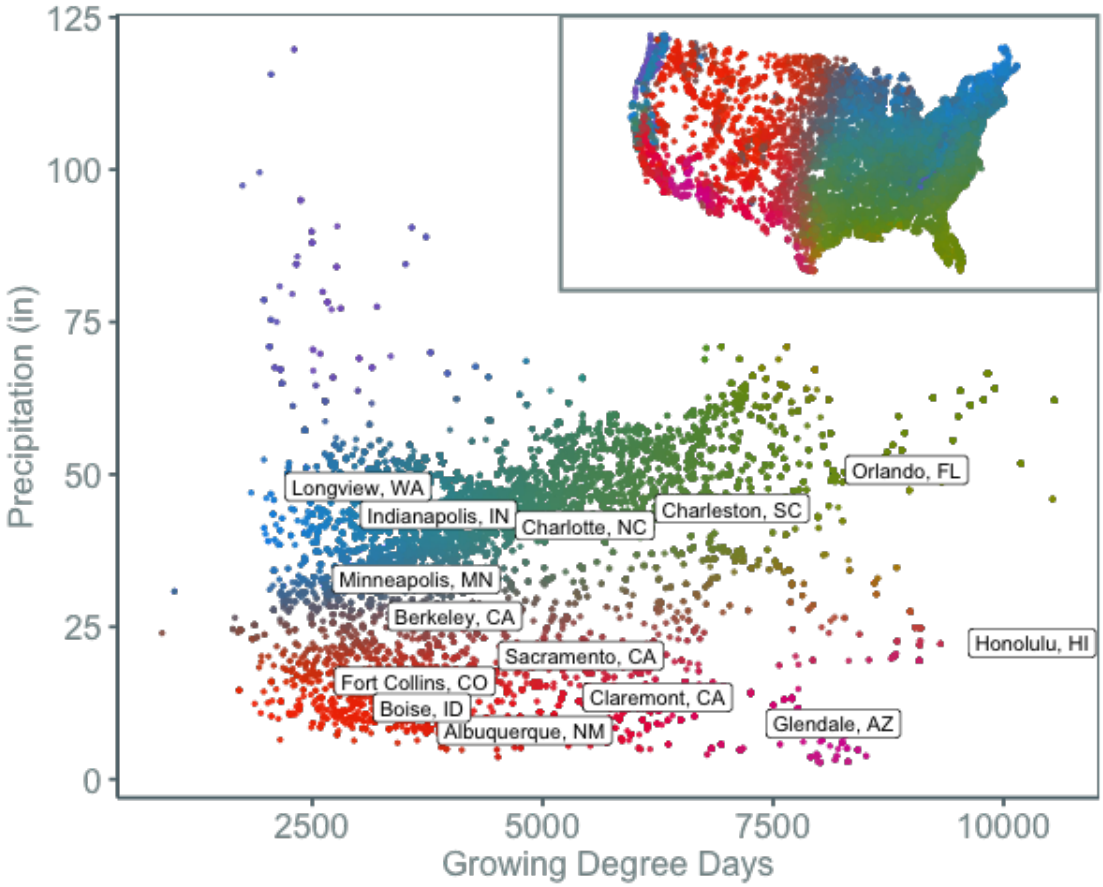
```
quartz
      2
```




```
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()
```

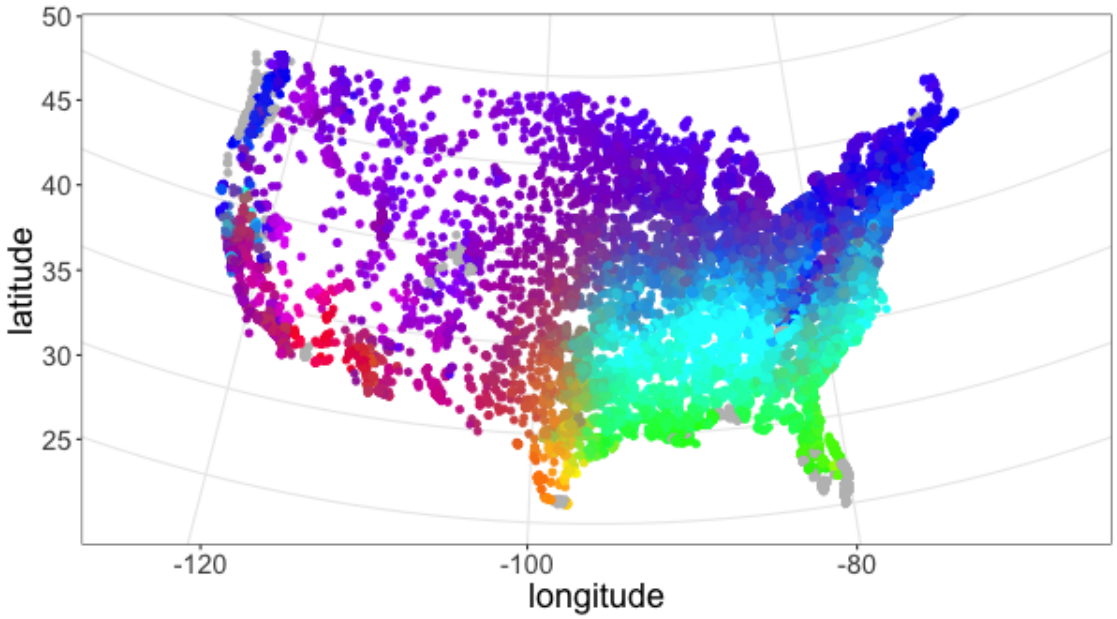
```
null device
          1
```



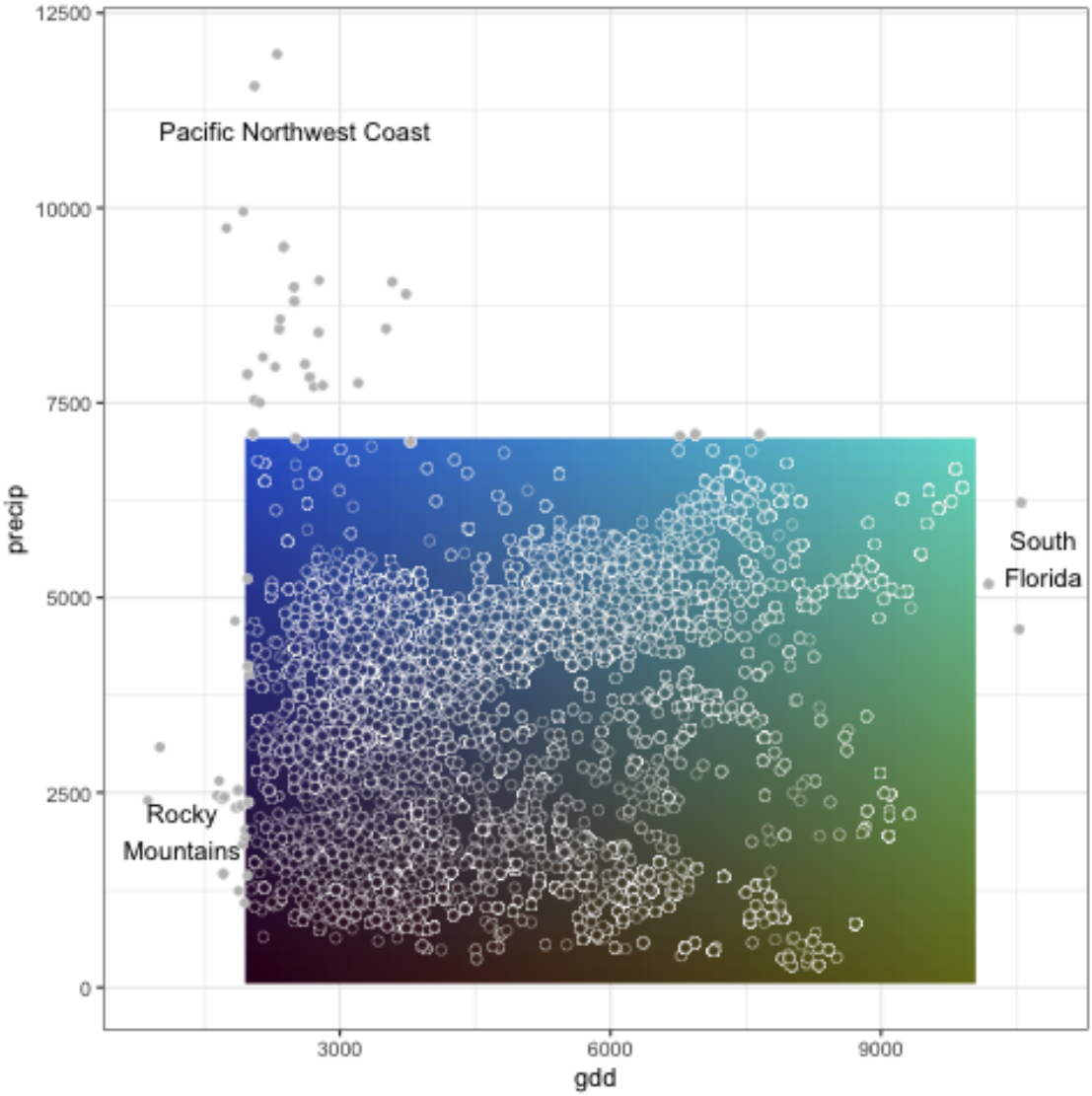
TODO: overlay the reference cities.

```
        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$precip_col[which.min(abs(x - cols$pre
```

```
  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
```

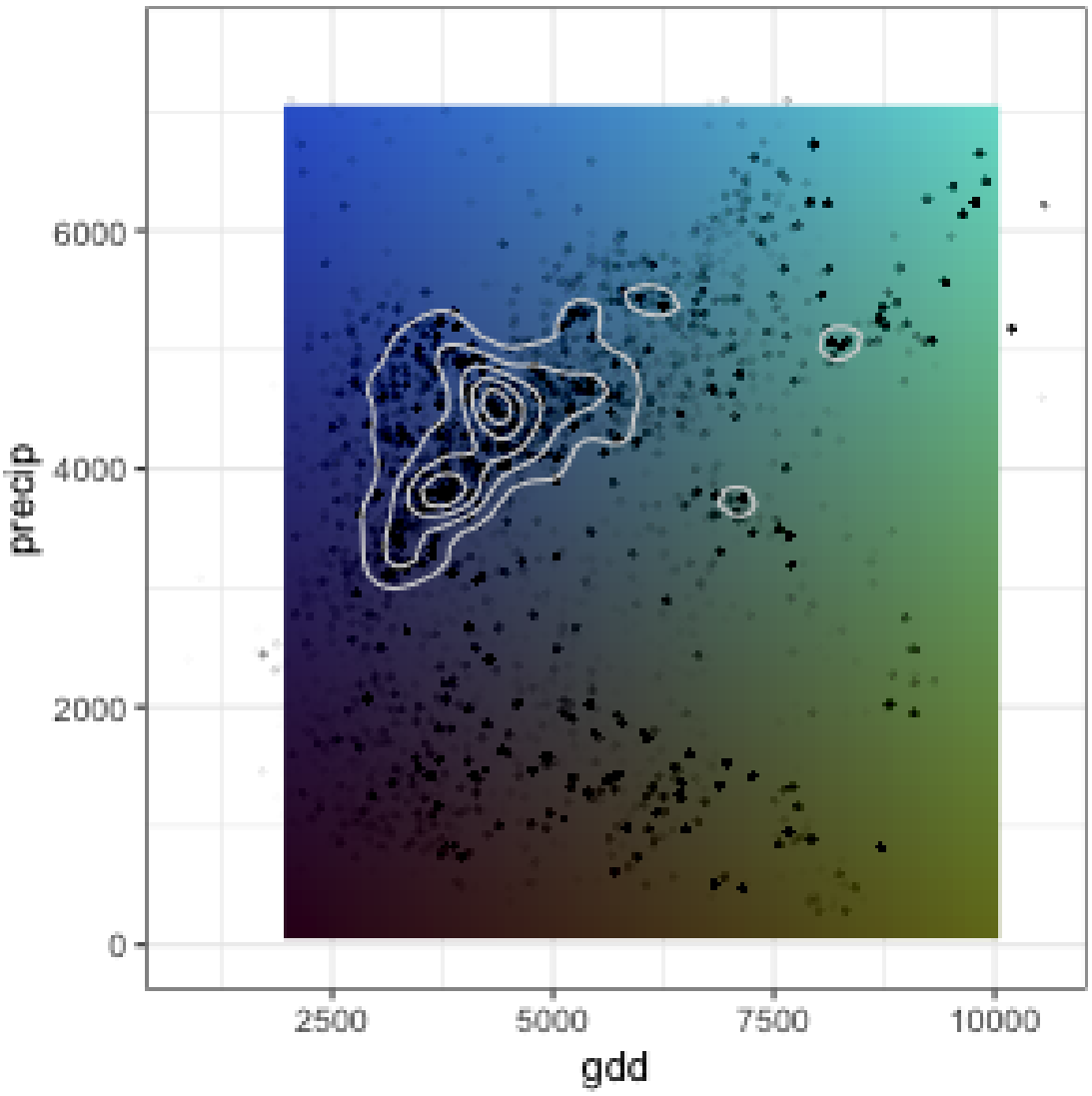


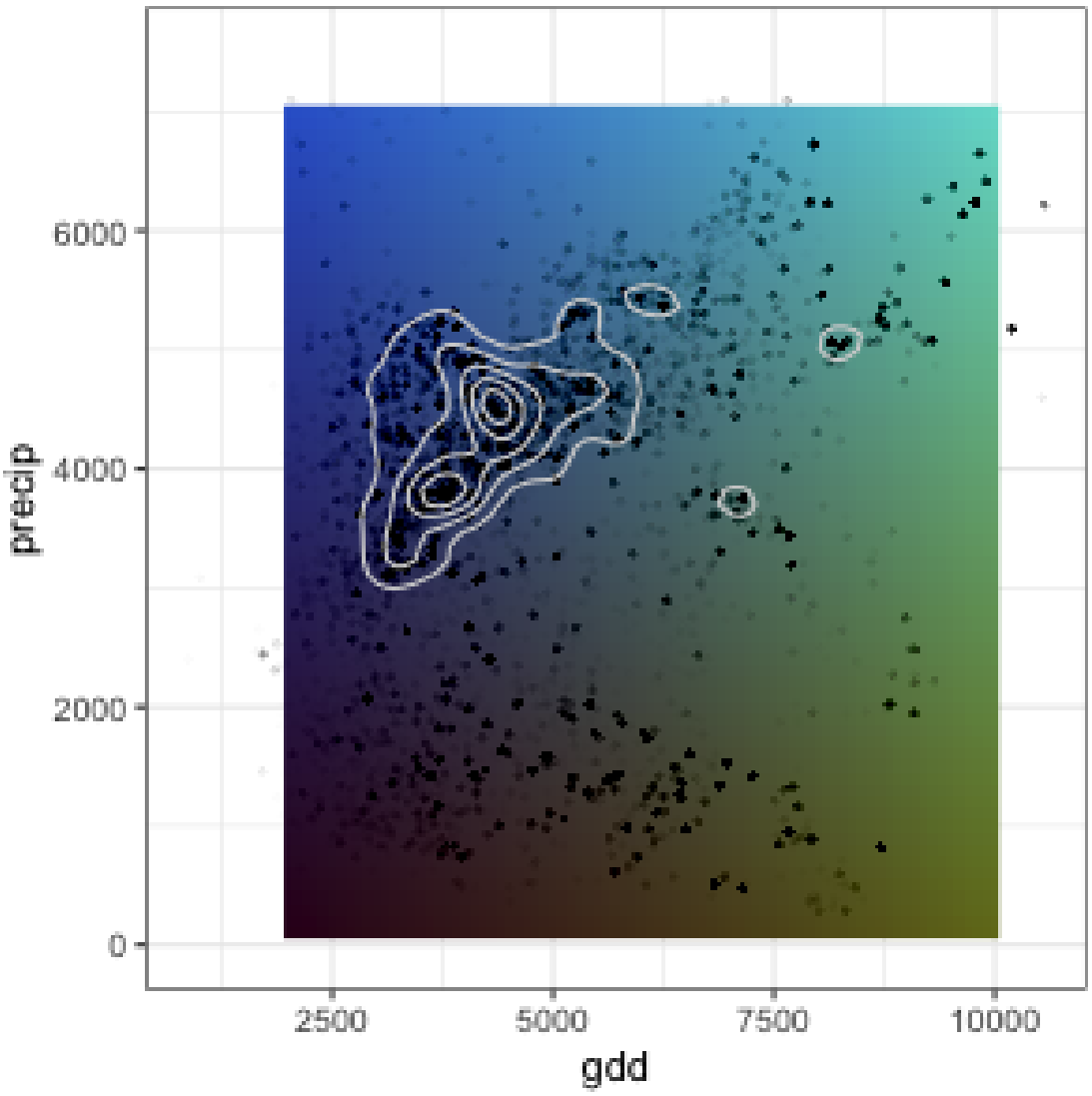
The dots should instead be a contour plot of the US population.

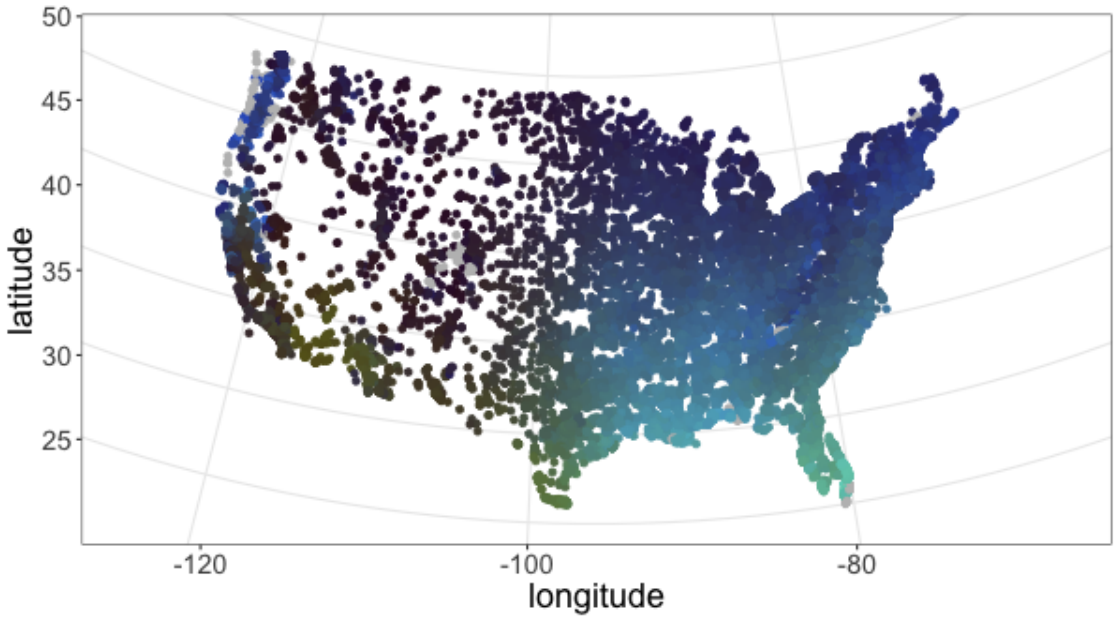
```
library(hdrcde)
```

```
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$

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


6000
4000
2000
0
precip
2500
5000
7500
10000
gdd


6000
4000
2000
0
precip
2500
5000
7500
10000
gdd

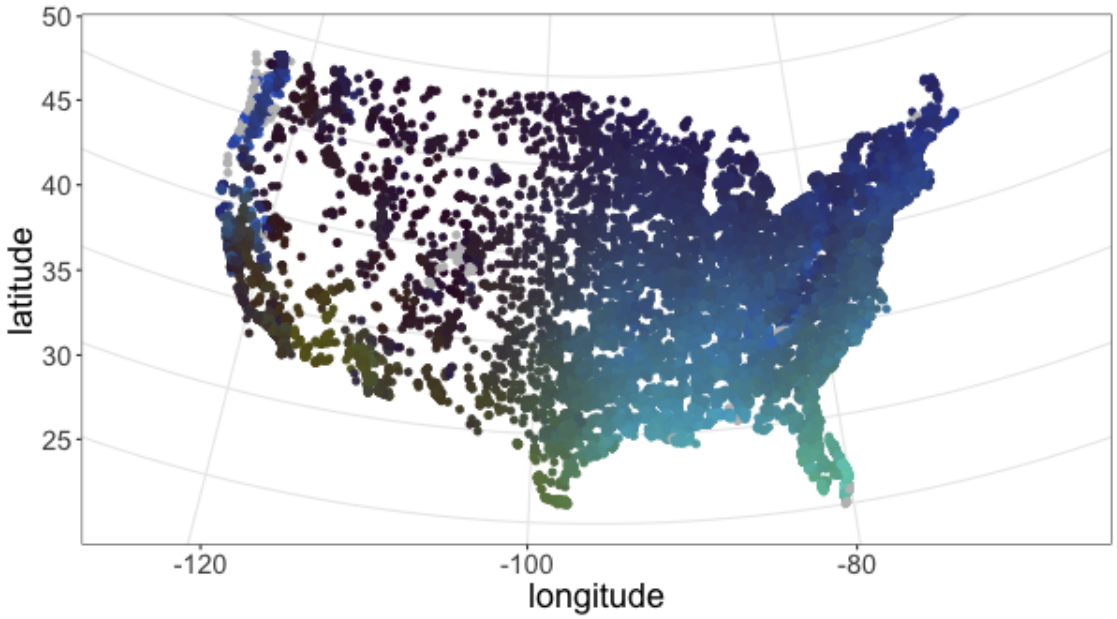


Trying to do log scale

```
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
```

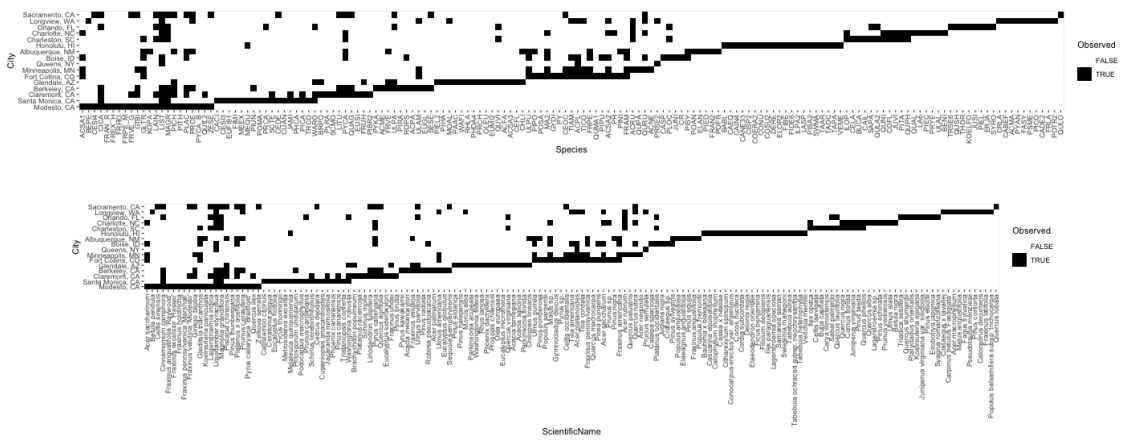
The dots should instead be a contour plot of the US population.



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)
```



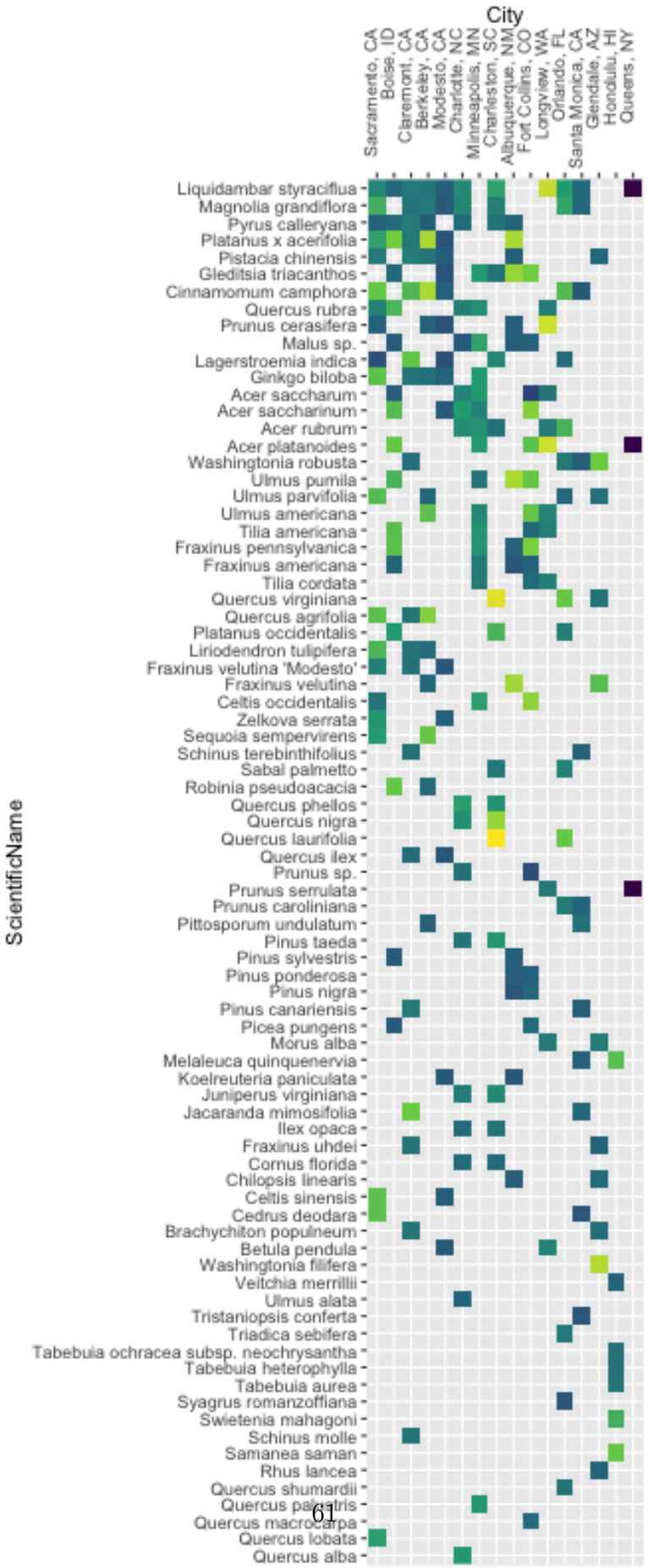
object 'cs_sp' not found

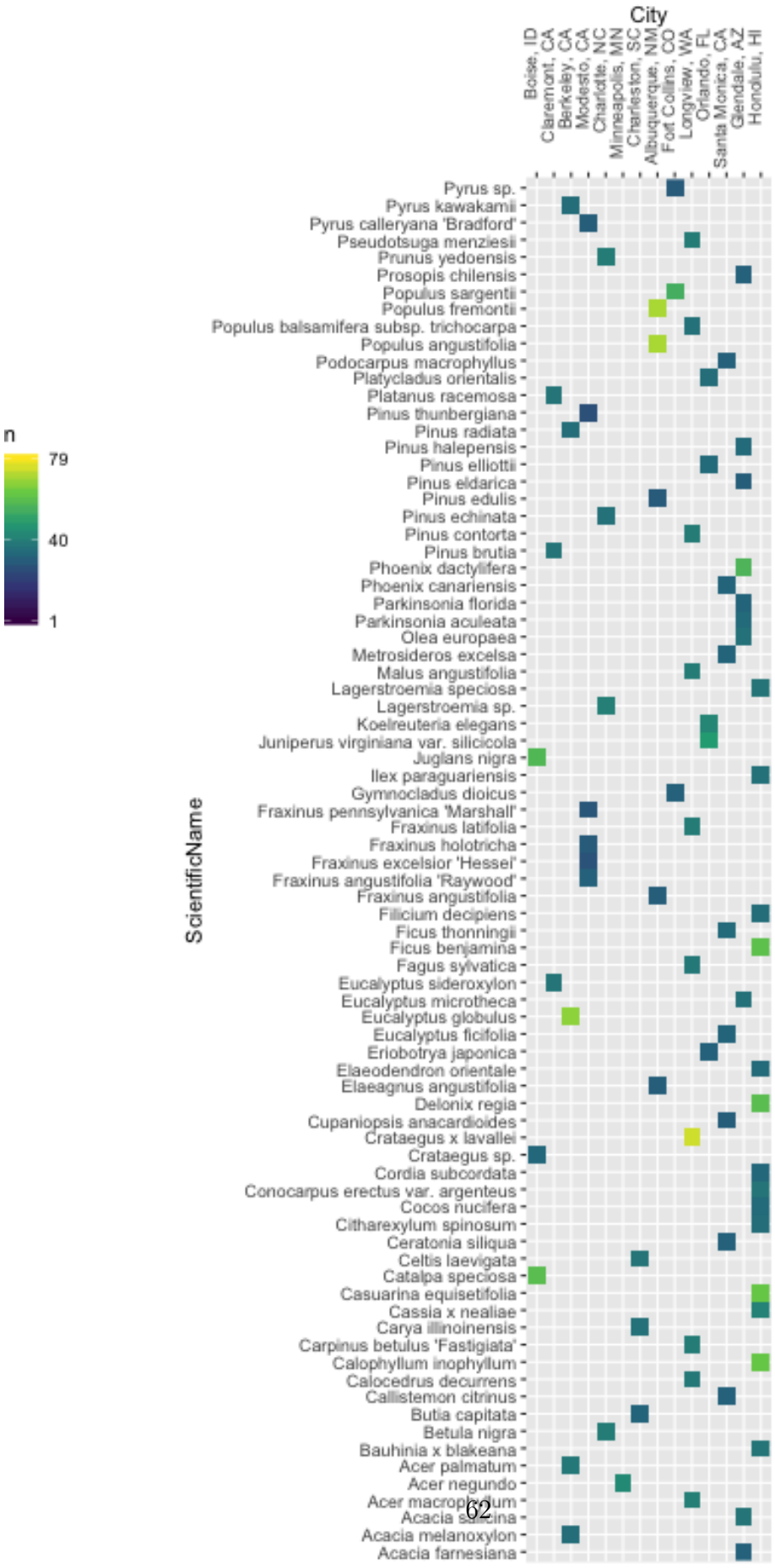
trees as fill. That is richer information.

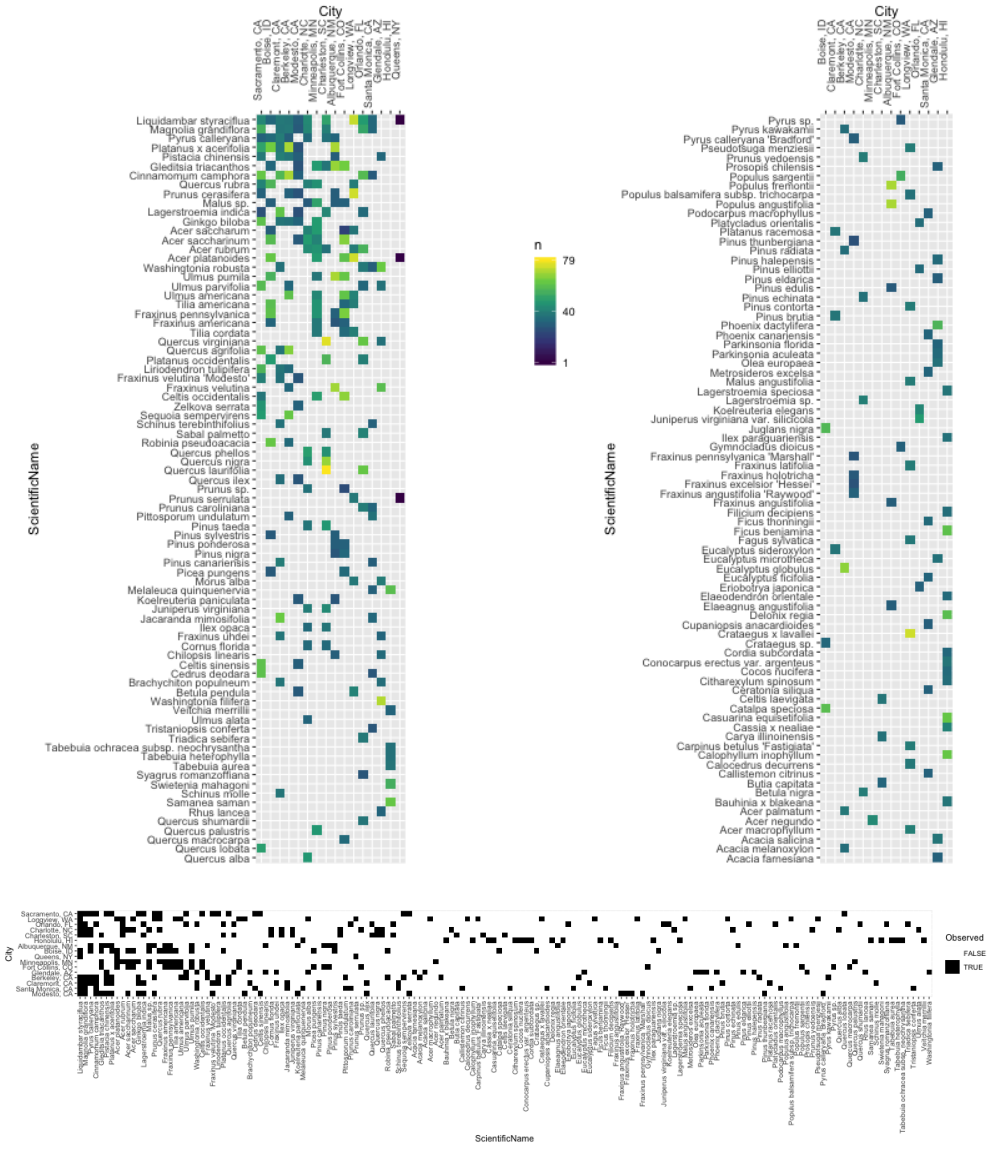
Also along the top I could have a N cities row, and along the right I could have a N species column. But that's a lot of work for what is readily apparent when looking.

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


City
Sacramento, CA
Boise, ID
Claremont, CA
Berkeley, CA
Modesto, CA
Charlotte, NC
Minneapolis, MN
Charleston, SC
Albuquerque, NM
Fort Collins, CO
Longview, WA
Orlando, FL
Santa Monica, CA
Glendale, AZ
Honolulu, HI
Queens, NY
ScientificName
Liquidambar styraciflua
Magnolia grandiflora
Pyrus calleryana
Platanus x acerifolia
Pistacia chinensis
Gleditsia triacanthos
Cinnamomum camphora
Quercus rubra
Prunus cerasifera
Malus sp.
Lagerstroemia indica
Ginkgo biloba
Acer saccharum
Acer saccharinum
Acer rubrum
Acer platanoides
Washingtonia robusta
Ulmus pumila
Ulmus parvifolia
Ulmus americana
Tilia americana
Fraxinus pennsylvanica
Fraxinus americana
Tilia cordata
Quercus virginiana
Quercus agrifolia
Platanus occidentalis
Liriodendron tulipifera
Fraxinus velutina 'Modesto'
Fraxinus velutina
Celtis occidentalis
Zelkova serrata
Sequoia sempervirens
Schinus terebinthifolius
Sabal palmetto
Robinia pseudoacacia
Quercus phellos
Quercus nigra
Quercus laurifolia
Quercus ilex
Prunus sp.
Prunus serrulata
Prunus caroliniana
Pittosporum undulatum
Pinus taeda
Pinus sylvestris
Pinus ponderosa
Pinus nigra
Pinus canariensis
Picea pungens
Morus alba
Melaleuca quinquenervia
Koelreuteria paniculata
Juniperus virginiana
Jacaranda mimosifolia
Ilex opaca
Fraxinus uhdei
Cornus florida
Chilopsis linearis
Celtis sinensis
Cedrus deodara
Brachychiton populneum
Betula pendula
Washingtonia filifera
Veitchia merrillii
Ulmus alata
Tristaniopsis conferta
Triadica sebifera
Tabebuia ochracea subsp. neochrysantha
Tabebuia heterophylla
Tabebuia aurea
Syagrus romanzoffiana
Swietenia mahagoni
Schinus molle
Samanea saman
Rhus lancea
Quercus shumardii
Quercus palustris
Quercus macrocarpa
Quercus lobata
Quercus alba
Pyrus sp.
Pyrus kawakamii
Pyrus calleryana 'Bradford'
Pseudotsuga menziesii
Prunus yedoensis
Prosopis chilensis
Populus sargentii
Populus fremontii
Populus balsamifera subsp. trichocarpa
Populus angustifolia
Podocarpus macrophyllus
Platycladus orientalis
Platanus racemosa
Pinus thunbergiana
Pinus radiata
Pinus halepensis
n
79
40
1

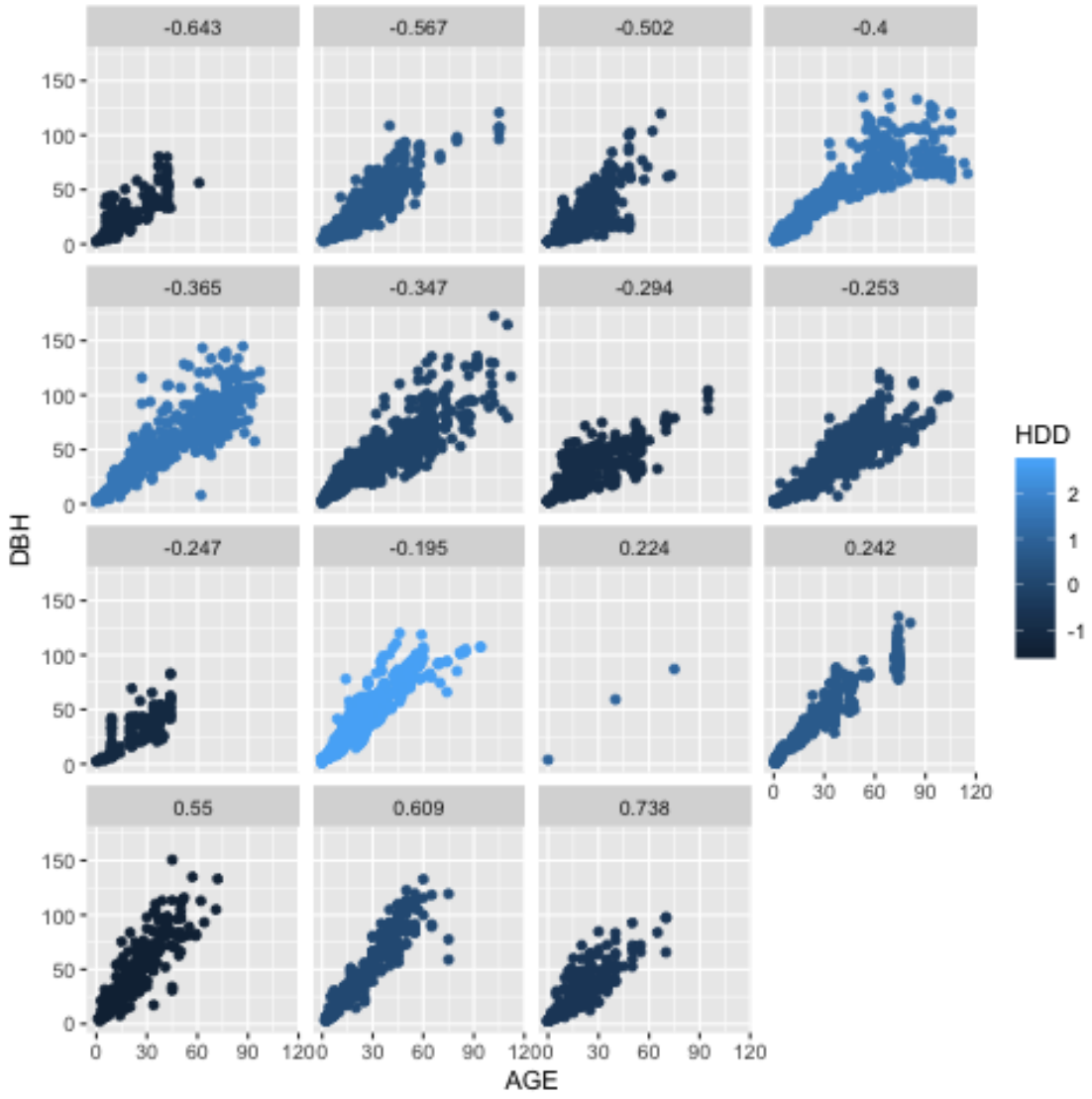

City
Sacramento, CA
Boise, ID
Claremont, CA
Berkeley, CA
Modesto, CA
Charlotte, NC
Minneapolis, MN
Charleston, SC
Albuquerque, NM
Fort Collins, CO
Longview, WA
Orlando, FL
Santa Monica, CA
Glendale, AZ
Honolulu, HI
Queens, NY
ScientificName
Liquidambar styraciflua
Magnolia grandiflora
Pyrus calleryana
Platanus x acerifolia
Pistacia chinensis
Gleditsia triacanthos
Cinnamomum camphora
Quercus rubra
Prunus cerasifera
Malus sp.
Lagerstroemia indica
Ginkgo biloba
Acer saccharum
Acer saccharinum
Acer rubrum
Acer platanoides
Washingtonia robusta
Ulmus pumila
Ulmus parvifolia
Ulmus americana
Tilia americana
Fraxinus pennsylvanica
Fraxinus americana
Tilia cordata
Quercus virginiana
Quercus agrifolia
Platanus occidentalis
Liriodendron tulipifera
Fraxinus velutina 'Modesto'
Fraxinus velutina
Celtis occidentalis
Zelkova serrata
Sequoia sempervirens
Schinus terebinthifolius
Sabal palmetto
Robinia pseudoacacia
Quercus phellos
Quercus nigra
Quercus laurifolia
Quercus ilex
Prunus sp.
Prunus serrulata
Prunus caroliniana
Pittosporum undulatum
Pinus taeda
Pinus sylvestris
Pinus ponderosa
Pinus nigra
Pinus canariensis
Picea pungens
Morus alba
Melaleuca quinquenervia
Koelreuteria paniculata
Juniperus virginiana
Jacaranda mimosifolia
Ilex opaca
Fraxinus uhdei
Cornus florida
Chilopsis linearis
Celtis sinensis
Cedrus deodara
Brachychiton populneum
Betula pendula
Washingtonia filifera
Veitchia merrillii
Ulmus alata
Tristaniopsis conferta
Triadica sebifera
Tabebuia ochracea subsp. neochrysantha
Tabebuia heterophylla
Tabebuia aurea
Syagrus romanzoffiana
Swietenia mahagoni
Schinus molle
Samanea saman
Rhus lancea
Quercus shumardii
Quercus palustris
Quercus macrocarpa
Quercus lobata
Quercus alba

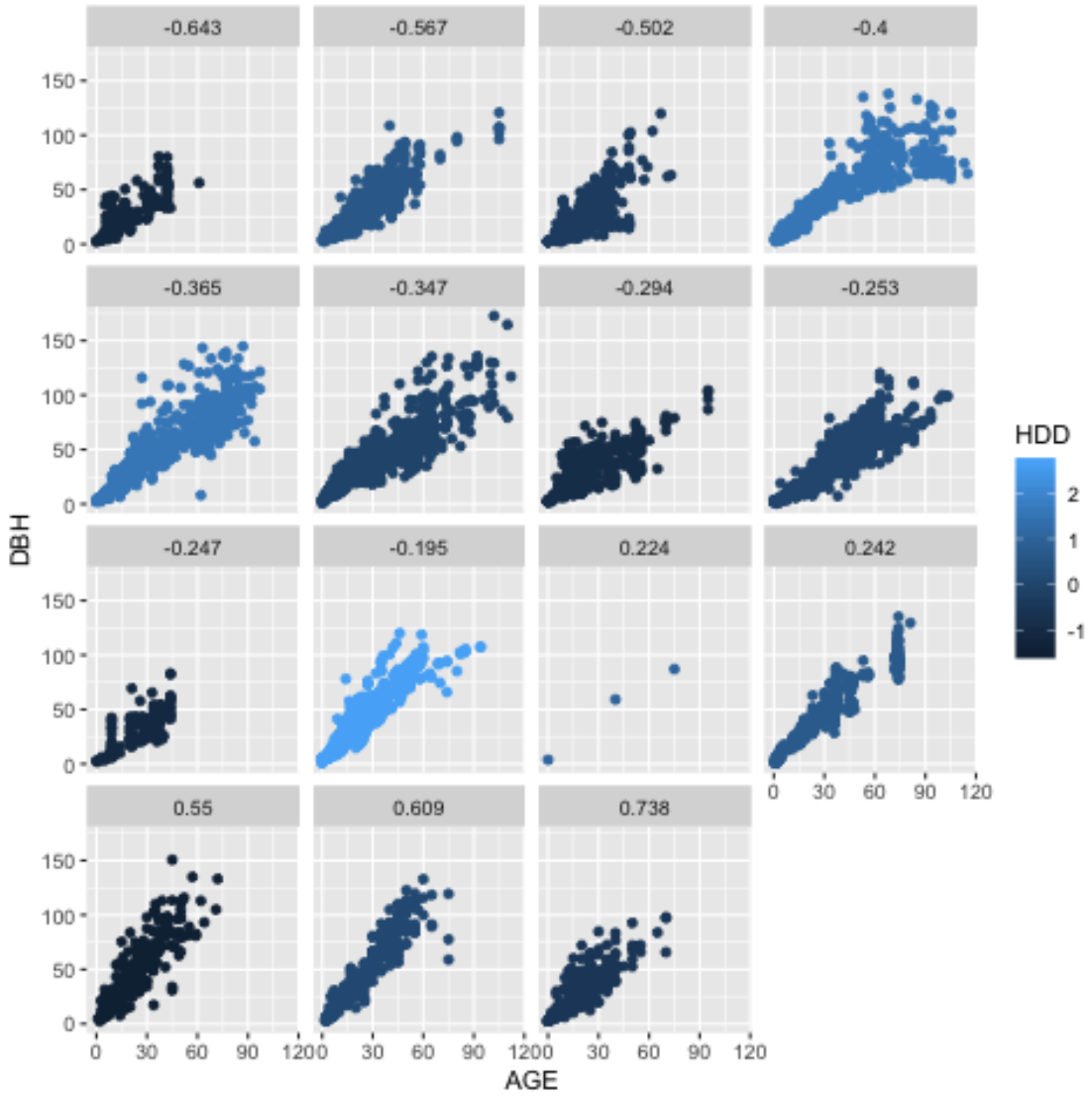

City
Boise, ID
Claremont, CA
Berkeley, CA
Modesto, CA
Charlotte, NC
Minneapolis, MN
Charleston, SC
Albuquerque, NM
Fort Collins, CO
Longview, WA
Orlando, FL
Santa Monica, CA
Glendale, AZ
Honolulu, HI
ScientificName
Pyrus sp.
Pyrus kawakamii
Pyrus calleryana 'Bradford'
Pseudotsuga menziesii
Prunus yedoensis
Prosopis chilensis
Populus sargentii
Populus fremontii
Populus balsamifera subsp. trichocarpa
Populus angustifolia
Podocarpus macrophyllus
Platycladus orientalis
Platanus racemosa
Pinus thunbergiana
Pinus radiata
Pinus halepensis
Pinus elliottii
Pinus eldarica
Pinus edulis
Pinus echinata
Pinus contorta
Pinus brutia
Phoenix dactylifera
Phoenix canariensis
Parkinsonia florida
Parkinsonia aculeata
Olea europaea
Metrosideros excelsa
Malus angustifolia
Lagerstroemia speciosa
Lagerstroemia sp.
Koelreuteria elegans
Juniperus virginiana var. silicicola
Juglans nigra
Ilex paraguariensis
Gymnocladus dioicus
Fraxinus pennsylvanica 'Marshall'
Fraxinus latifolia
Fraxinus holotricha
Fraxinus excelsior 'Hessei'
Fraxinus angustifolia 'Raywood'
Fraxinus angustifolia
Filicium decipiens
Ficus thonningii
Ficus benjamina
Fagus sylvatica
Eucalyptus sideroxylon
Eucalyptus microtheca
Eucalyptus globulus
Eucalyptus ficifolia
Eriobotrya japonica
Elaeodendron orientale
Elaeagnus angustifolia
Delonix regia
Cupaniopsis anacardioides
Crataegus x lavallei
Crataegus sp.
Cordia subcordata
Conocarpus erectus var. argenteus
Cocos nucifera
Citharexylum spinosum
Ceratonia siliqua
Celtis laevigata
Catalpa speciosa
Casuarina equisetifolia
Cassia x nealiae
Carya illinoinensis
Carpinus betulus 'Fastigiata'
Calophyllum inophyllum
Calocedrus decurrens
Callistemon citrinus
Butia capitata
Betula nigra
Bauhinia x blakeana
Acer palmatum
Acer negundo
Acer macrophyllum
Acacia saligna
Acacia melanoxylon
Acacia farnesiana
n
79
40
1



look at climate variables

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


-0.643
-0.567
-0.502
-0.4
-0.365
-0.347
-0.294
-0.253
-0.247
-0.195
0.224
0.242
0.55
0.609
0.738
DBH
AGE
HDD
2
1
0
-1
0
30
60
90
120
50
100
150



Is minneapolis an influential point?