Analysis of potential sample size

```
library(tidyverse)
library(dbplyr)
library(RSQLite)
library(terra)
library(stars)
```

Download FIA database locally

Only run once. Future - make a regular cron job to download the csv files that can be hosted on s3 (then arrow can be used to access)

```
download.file("https://apps.fs.usda.gov/fia/datamart/Databases/SQLite_FIADB_ENTIRE.zip",
              "SQLite_FIADB_ENTIRE.zip",
              method = "curl")
unzip("SQLite_FIADB_ENTIRE.zip")
```

Connect to database

```
fia_all <- DBI::dbConnect(RSQLite::SQLite(), "SQLite_FIADB_ENTIRE.db")
```

Access tree table

```
trees <- tbl(fia_all, "TREE")
```

Analyze sample size

Read in the table of focal species from Mike Bell

```
focal_species <- read_csv("focal_species.csv", show_col_types = FALSE)
```

Calculate the number of remeasurements for each unique tree in the database.

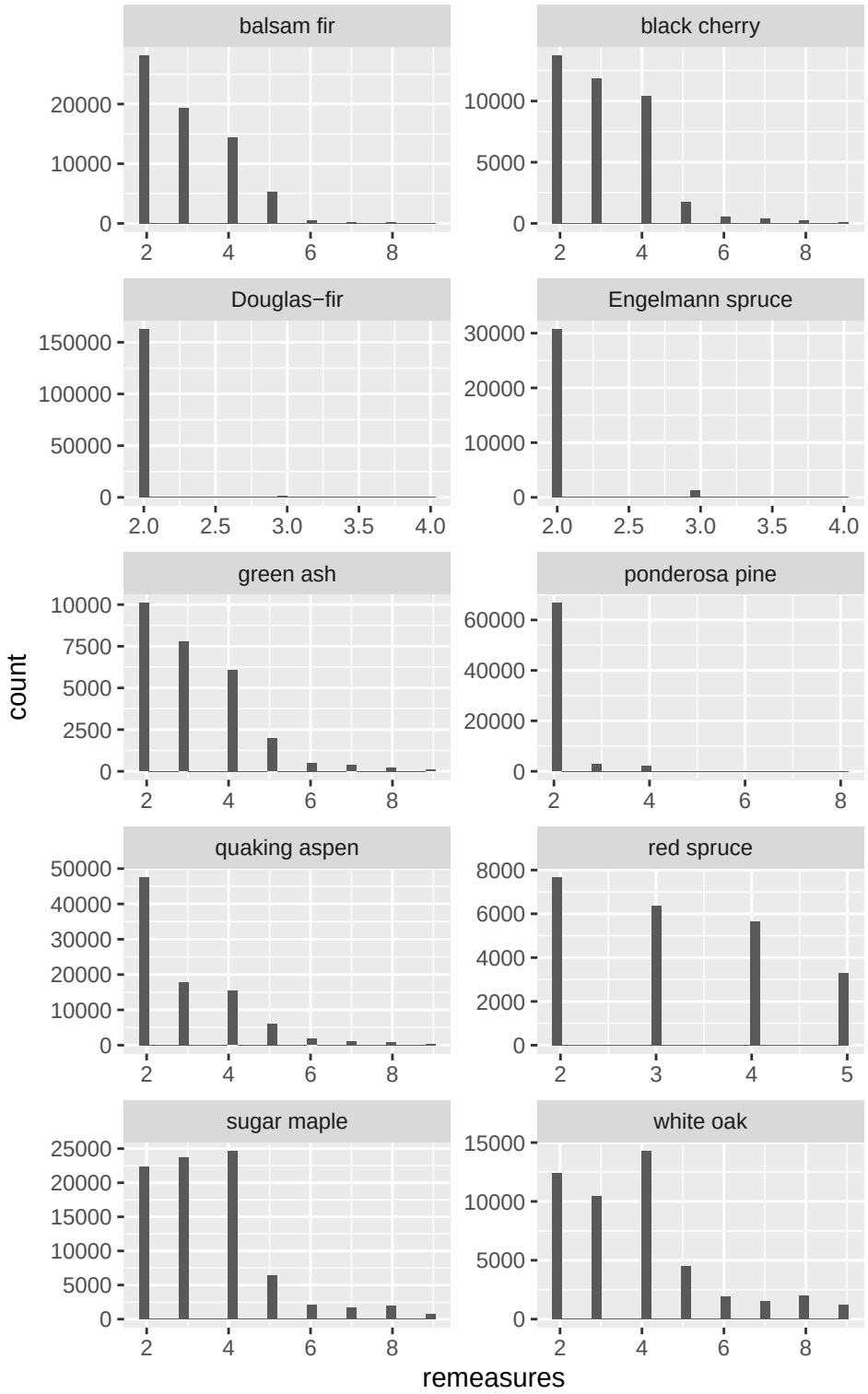
```
focal_SPCD <- focal_species$SPCD
tree_stats <- trees |>
  select(SUBP, TREE, PLOT, STATECD, SPCD, INVYR, DIA) |>
  filter(SPCD %in% focal_SPCD,
         INVYR > 1995,
         DIA > 5.0,
         SUBP <= 4) |>
  mutate(id = paste(STATECD, PLOT,SUBP,TREE, sep = "-")) |>
  summarise(remeasures = n(), .by = c("id", "SPCD")) |>
  filter(remeasures > 1 & remeasures < 10) |>
  collect()
```

Remeasurements by species

This shows the max number of remeasurements per tree for the different species.

```
tree_stats |>
  left_join(focal_species, by = "SPCD") |>
  ggplot(aes(x = remeasures)) +
  geom_histogram() +
  facet_wrap(~common_name, scales = "free", ncol = 2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


balsam fir
black cherry
Douglas−fir
Engelmann spruce
green ash
ponderosa pine
quaking aspen
red spruce
sugar maple
white oak
count
remeasures

Sample size by species

This table is the potential sample size per species. It is the number of trees that have been remeasured.

```
tree_stats |>
  summarise(sample_size = n(), .by = "SPCD") |>
  left_join(focal_species, by = "SPCD") |>
  select(common_name,sample_size) |>
  knitr::kable()
```

common_name	sample_size
white oak	48338
black cherry	39274
green ash	27108
sugar maple	83754
Douglas-fir	163998
quaking aspen	91257
Engelmann spruce	31978
ponderosa pine	72084
balsam fir	68246
red spruce	22990

Example trees

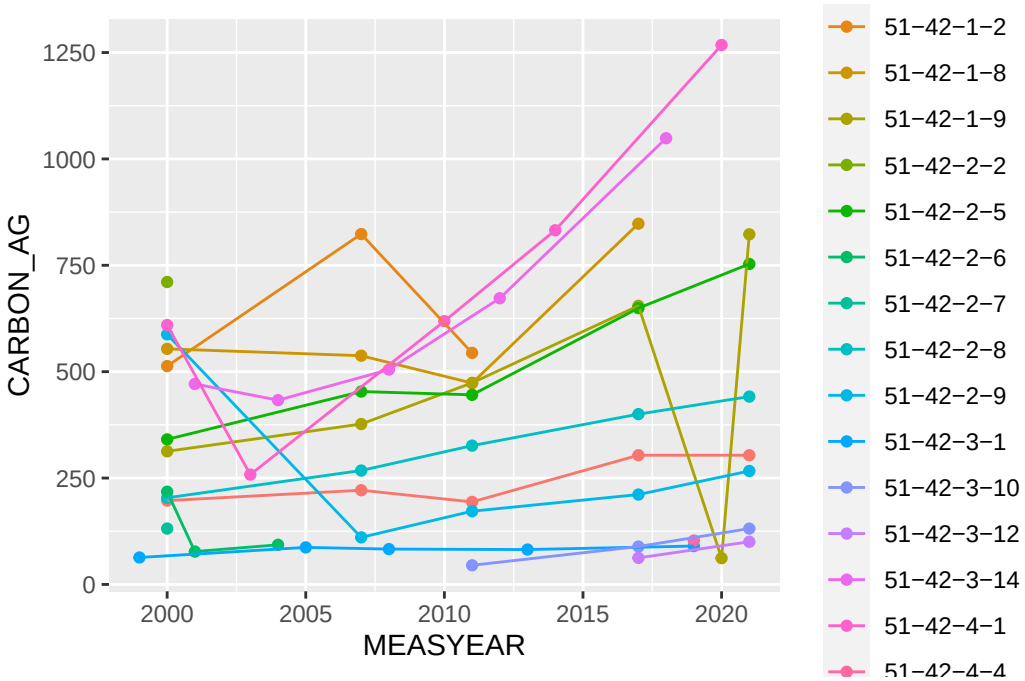
Here is a figure with sugar maples trees from a single plot in Virginia

```
example_trees <- trees %>%
  filter(STATECD == 51, SPCD == 318, PLOT == 42, SUBP <= 4) |>
  select(PLT_CN, SUBP, TREE, PLOT, STATECD, SPCD, CARBON_AG, INVYR, CYCLE, STATUSCD, CN, P
  mutate(id = paste(STATECD,PLOT,SUBP,TREE, sep = "-")) |>
  collect()

plot <- tbl(fia_all, "PLOT")
example_plot <- plot %>%
  filter(STATECD == 51, PLOT == 42) |>
  select(CN, MEASYEAR, MEASMON, MEASDAY, REMPER, LAT, LON, ELEV, PLOT) |>
  rename(PLT_CN = CN) |>
  collect()
```

```
example_trees <- left_join(example_trees, example_plot, by = "PLT_CN")
```

```
example_trees |>
  filter(STATUSCD == 1,
         DIA > 5.0) |>
ggplot(aes(x = MEASYEAR, y = CARBON_AG, color = factor(id))) +
  geom_point() +
  geom_line()
```



Linking to Ndep

```
n_dw_2000 <- rast("n_dw-2000/n_dw-2000.tif")
n_dw_2000
```

```
class       : SpatRaster
dimensions  : 727, 1158, 1  (nrow, ncol, nlyr)
resolution  : 4000, 4000  (x, y)
extent      : -2364000, 2268000, 266000, 3174000  (xmin, xmax, ymin, ymax)
coord. ref. : NAD83 / Conus Albers (EPSG:5070)
source      : n_dw-2000.tif
```

```
name         :  n_dw-2000
min value    :  0.3311314
max value    : 59.2533112
```

```
summary(n_dw_2000)
```

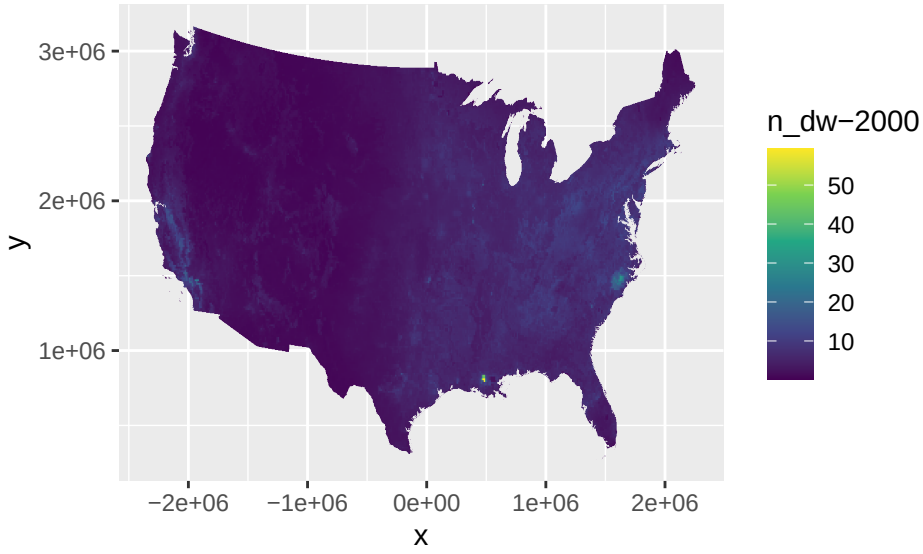
```
Warning: [summary] used a sample
```

```
   n_dw.2000
 Min.   : 0.41
 1st Qu.: 1.58
 Median : 3.47
 Mean   : 4.03
 3rd Qu.: 6.01
 Max.   :59.25
 NA's   :42325
```

```
n_dw_2000_df <- as.data.frame(n_dw_2000, xy = TRUE)
```

```
ggplot() +
    geom_raster(data = n_dw_2000_df , aes(x = x, y = y, fill = `n_dw-2000` )) +
    scale_fill_viridis_c()
```

```
Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
i Consider using `geom_tile()` instead.
```



Get Ndep for each tree

```
example_trees_sf <- example_trees |>
  distinct(id, LON, LAT) |>
  group_by(id) |>
  slice(1) |>
  ungroup() |>
  rename(longitude = LON,
         latitude = LAT) |>
  sf::st_as_sf(coords=c("longitude", "latitude"),
                          crs = 4326) |>
  tibble::rowid_to_column("FID")

n_dw_2000 <- read_stars("n_dw-2000/n_dw-2000.tif")
sf_plots <- st_transform(example_trees_sf, st_crs(n_dw_2000))
plot_ndep = st_extract(n_dw_2000, sf_plots)

plot_each_tree <- as.data.frame(plot_ndep) |>
  tibble::rowid_to_column("FID") |>
  right_join(example_trees_sf, by = "FID") |>
  rename(ndep = `n_dw-2000.tif`) |>
  mutate(year = 2000) |>
  select(id, year, ndep)
```