Multilevel Analysis of Tree Growth Rates in Northeastern US Forests

Rebecca Vithayathil

Abstract

This study examines tree growth rate variation across northeastern US forests using USFS Forest Inventory and Analysis (FIA) data from Connecticut, Massachusetts, and Rhode Island (2008-2020). The final model selected was a three-level hierarchical linear model with 88,645 growth observations from 42,905 trees across 1,531 plots. Model results revealed that 60.5% of growth variation occurs between individual trees, 17.0% between plots, and 4.2% between forest types. Stand age ($p<0.001$) and species identity (93 species analyzed) significantly predict annual diameter growth (square-root transformed). The intraclass correlation coefficients confirm substantial clustering at tree and plot levels, validating the multilevel modeling approach. These findings quantify how individual tree characteristics dominate growth patterns while plot-level conditions and forest composition contribute meaningful variation.

Introduction

Forest growth rates determine carbon sequestration capacity, timber productivity, and ecosystem services. Understanding variation in tree growth across spatial scales informs many real world problems such as forest management and climate change mitigation strategies. However, tree growth is inherently hierarchical: individual trees grow within plots, plots exist within forest types, and forest types vary across ecological regions.

The Forest Inventory and Analysis (FIA) program provides repeated measurements of permanent forest plots, enabling growth rate calculation through remeasurement intervals. This study leverages FIA's hierarchical structure to partition growth variation and identify factors explaining differences across scales.

Research Question: How do annual tree diameter growth rates vary across ecological subsections, states, forest types, plots, and individual trees in northeastern US forests, and what roles do stand age and species identity play in explaining this variation?

Method

Data Source and Processing

FIA data from Connecticut, Massachusetts, and Rhode Island were downloaded via the rFIA R package and combined into a unified dataset. Each state database contained 21 data tables. The data was filtered for 2008-2020 after EDA on the data table TREE showed that 2008 was

where the first remeasurements of the same trees occurred. The outcome variable was pulled from the TREE_GRM_COMPONENT data table which contained pre-calculated annual diameter growth rates (column ANN_DIA_GROWTH) standardized across varying remeasurement intervals (4-7 years). Data tables TREE, COND, and PLOTGEOM were used for grouping and predictor variables.

Table	Information Used
TREE_GRM_COMPONENT	Annual diameter growth (ANN_DIA_GROWTH); tree linkages; initial/final diameters
TREE	Tree identifiers (PLOT, SUBP, TREE); species code (SPCD); measurement year; tree status
COND	Forest type code (FORTYPCD); stand age (STDAGE)
PLOTGEOM	Ecological subsection (ECOSUBCD)

Missing Data

TREE was also heavily investigated for missing data. TREE is the dataset containing all TREES recorded while TREE_GRM_COMPONENT is prefiltered for only trees which have repeated measurements, so we used data table TREE in order to estimate how much data may have been excluded and for what reasons from TREE_GRM_COMPONENT. We found when looking at the TREE table nearly 60% were missing repeat measurements. First inventory year was investigated to see if that was a factor.

Missing Data Inventory Year:

- **1985-2007**: First measurements only (missing values in PREVDIA column - 0% have previous diameter with exception of a few in 1998)
- **2008-2020**: Includes remeasurements (years have values for 84-95% PREVDIA)

Based on this finding data limited to 2008 - 2020 and then reviewed for missingness in that time period. For this adjusted time frame the column DIA for current diameter was missing 14% and the PREVDIA column for previous diameter measurements was missing 11%. We wanted to investigate if these were measurement errors or something that was valid for our dataset. We reviewed the column STATUSCD (status code alive or dead) to see if missingness was due to tree death. There was some variation in status code recording strategies across years that was investigated.

What we found for our data:

Status Code	Description
0	Dead tree (gone)
1	Live tree
2	Dead tree (still standing)

Missing Previous Diameter (column PREVDIA) was found to in part be due to "ingrowth" trees. Ingrowth trees are too small to measure in the previous inventory (< 5" Diameter threshold). So some grew large enough by 2008 and later to be included in measurements but we can't calculate growth rates for these trees (no baseline). For these values it's valid that they don't have remeasurements and no imputation would be needed.

Missing Current Diameter (column DIA) Allmost all trees missing current diameter are dead or removed. You can't measure the diameter of a tree that's gone (STATUSCD=0) or too deteriorated and no longer growing (STATUSCD=2).

Missing Current Diameter (10,807 observations)

- Dead: 10,615 trees (98.22%) Expected - can't measure dead/removed trees
- Alive: 192 trees (1.78%) - Small number

The 192 live trees (1.78%) with missing current diameter might be measurement errors or trees that couldn't be accessed. These are values that could be imputed.

Missing Previous Diameter (8,595 observations)

- Dead: 400 trees (4.65%) - Some trees died before being remeasured
- Alive: 8,195 trees (95.35%) Many of these are likely ingrowth trees

The table below shows trees with measured diameters but no observed previous diameter compared to trees with remeasurements.

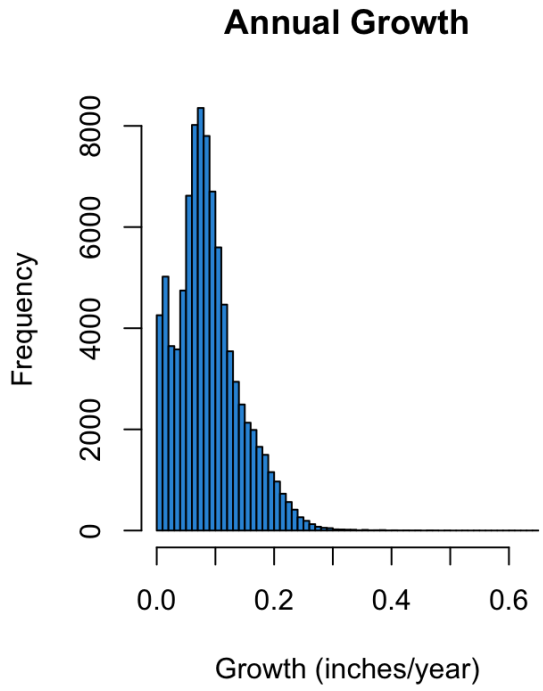
Tree Type	N	Mean (in)	Median (in)
Possible Ingrowth	8,195	6.92	5.6
Remeasured	49,011	9.65	8.8

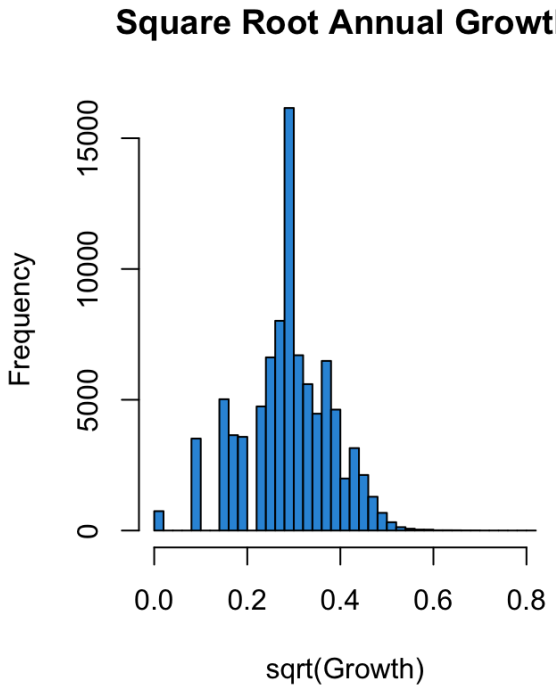
The possible ingrowth trees are smaller on average as a group than the remeasured trees. However, we also found some large diameter values that were too large to have been a sapling in previous collection cycles. These larger trees are likely trees that are being measured for the

first time. Based on reference material we could come up with a threshold diameter that would be impossible to have been less than 5 inches in diameter (minimum for measurement) 7 years before (our maximum collection interval). Using this threshold we could conclude which of the trees are truly ingrowth and which trees are new individuals that we could impute.

Variable Construction

Outcome variable: Annual diameter growth exhibited strong right skew. Square-root transformation was chosen to normalize over log to preserve zero values. Trees with no measurable growth could be ecologically meaningful indicators of stress or suppression. Some gaps are seen in the distribution due to transformation of discrete growth values (all rounded to the hundredths place in the original FIA table).





Hierarchical identifiers: Unique tree IDs were constructed by concatenating state, plot, subplot, and tree numbers (TREE_ID = STATE_PLOT_SUBP_TREE), enabling tracking of repeated measurements. Forest type codes (FORTYPCD, 47 levels) served as a grouping variable from data table COND.

Predictor variables: Stand age (STDAGE, continuous years) and species code (SPCD, 93 categorical levels) were joined from COND and TREE tables respectively. **Statistical Model**

Results

Model Development

Initial approach five-levels:

A five-level hierarchical model was initially specified to capture the full ecological hierarchy: Ecological Subsections → States → Forest Types → Plots → Trees. Our fixed effects were stand age and species. For our random intercept each group (tree, plot, forest type, state, subsection) gets its own baseline growth rate, but the effect of predictors (stand age, species) are assumed to be the same across all groups. This model asks the question “Do growth rates vary by stand age and species while accounting for clustering at multiple geographic and ecological levels?”.

Random Effects (5 nested levels):

Level	Group	Description	Number of Groups
5	Ecological subsections	Broad climate/geography regions	15
4	States	Connecticut (CT), Massachusetts (MA), Rhode Island (RI)	3
3	Forest types	Nested within states	45
2	Plots	Permanent monitoring locations	1,080
1	Trees	Individual trees measured repeatedly over time	36,574

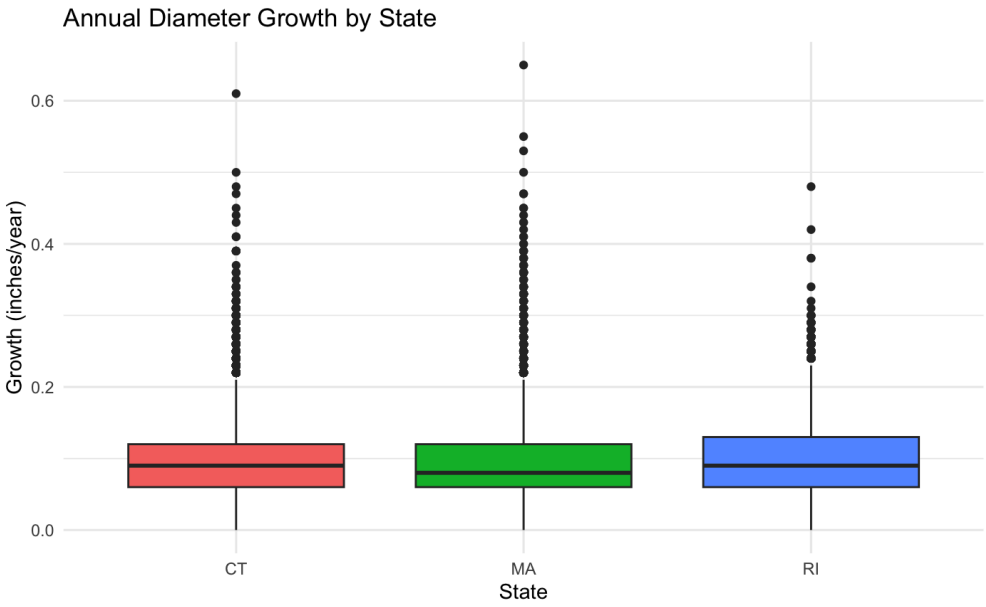
Variance and ICC of Random Effects:

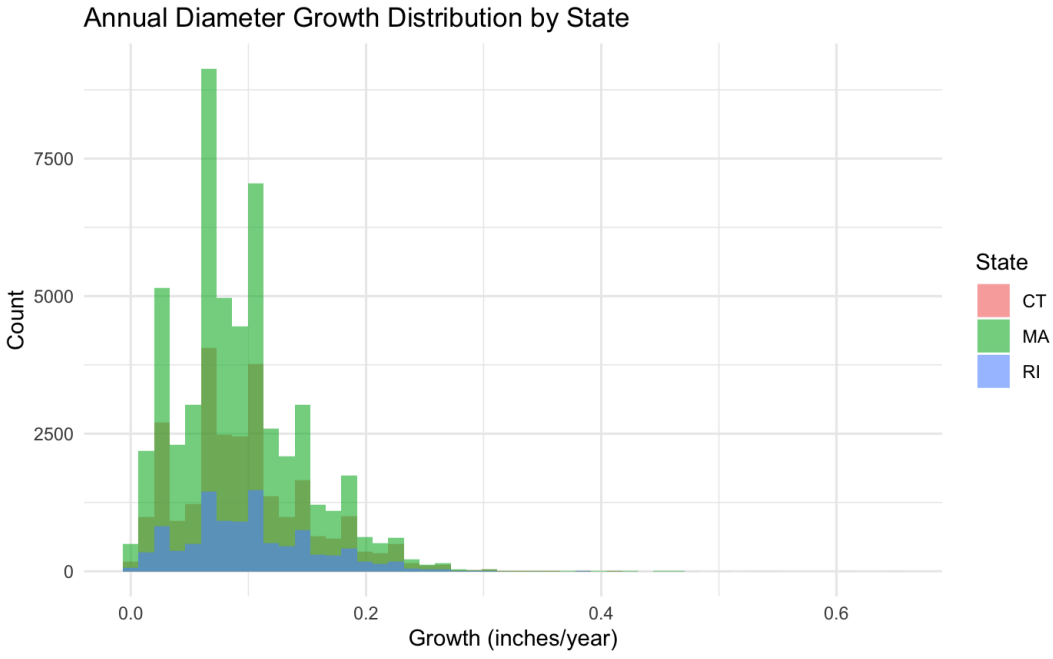
Level	Variance	ICC %	Interpretation
Ecosubcd (Level 5)	0.00009	1.2%	Small Ecological subsection variation
State (Level 4)	0.00006	0.8%	Very small state-level differences
Forest Type (Level 3)	0.00016	2.2%	Small Forest type differences
Plot (Level 2)	0.00117	16.0%	Good Plot-level variation
Tree (Level 1)	0.00447	61.2%	Largest – individual tree differences

Our residual variances was 18.5% this is the unexplained variation in our model. Examination of variance components for our groupings revealed problematic patterns. The ecological subsection level and the state level both had very small variances. Combined they accounted

for ~ 2% of variation. For ecological subsection it is likely that much of the climate and geographic variance is already being captured by the plot and forest type.

State is the worst grouping. When we look at the growth by state it's clear that this is the case and not due to any issue with model specification.





Growth per year follows a very similar pattern for each state. State lines are arbitrarily created with respect to biological phenomena so this makes sense. We also picked adjacent states so they will experience similar weather and have overlapping forest types. We will remove state and ecological subsection and rerun the model.

Updated approach three-levels:

Model Comparison:

Model	REML	Tree ICC	Plot ICC	Upper Levels ICC
5-level	-246618.9	61.3%	16.0%	4.1% (FT+State+Eco)
3-level	-246611.8	60.5%	17.0%	4.2% (FT only)

Comparing the 5-level and 3-level models showed that both had nearly identical fit, with very close REML values. In both models, variance was dominated by Tree (around 61%) and Plot (]16% and 17%) levels. While the 5-level model included additional levels for State and Ecological Subsection (Ecosubcd), these added complexity without noticeably improving model fit. Forest Type, however, remained important by explaining about 4.2% of the variance, indicating it is worth keeping. Based on these findings, the decision was made to use the simpler 3-level model.

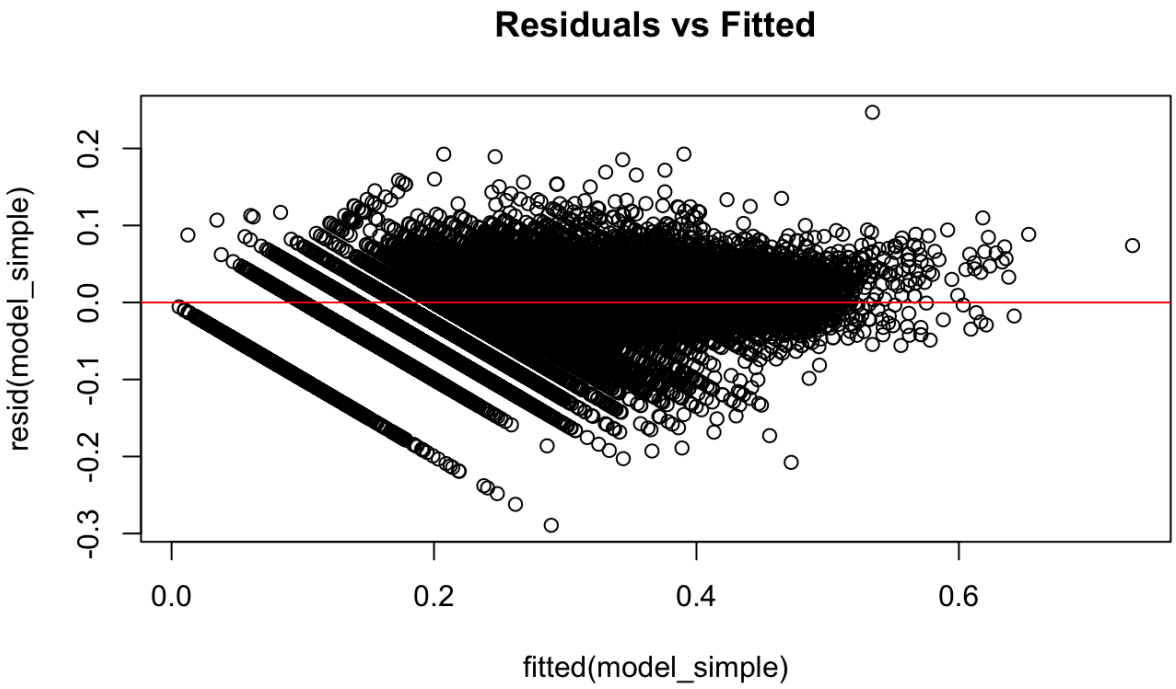
Individual tree differences account for the majority (60.5%) of growth variation, indicating that genetics, competitive status, and microsite conditions dominate growth patterns. Plot-level variation (17.0%) reflects soil quality, topography, and disturbance history. Forest type contributes modest variation (4.2%). When we test model performance with and without forest type it didn't change model performance so we should keep it. It is our only bioregion grouping in the model so retaining it adds ecological information we wouldn't have otherwise. We should review reference material to see if there is a better bioregion grouping we could use.

Evaluation of fixed effects showed that Stand Age (STDAGE) has a coefficient of -0.000826 and is highly significant ($p < 0.001$), indicating that older stands tend to exhibit slower growth. This negative effect is consistent with reference literature. This variable will be kept in the model. Regarding Species (SPCD), among the 93 species coefficients examined, several species (such as 315, 319, 391, 421, 701, 763, 816, 922, 923, 927, and others) demonstrated highly significant differences from the baseline species ($p < 0.001$). Additionally, many species showed significance ($p < 0.05$), including species 43, 68, 71, 91, 97, 130, among others. However, approximately 40 to 50 species did not differ significantly from the baseline ($p > 0.05$). Despite

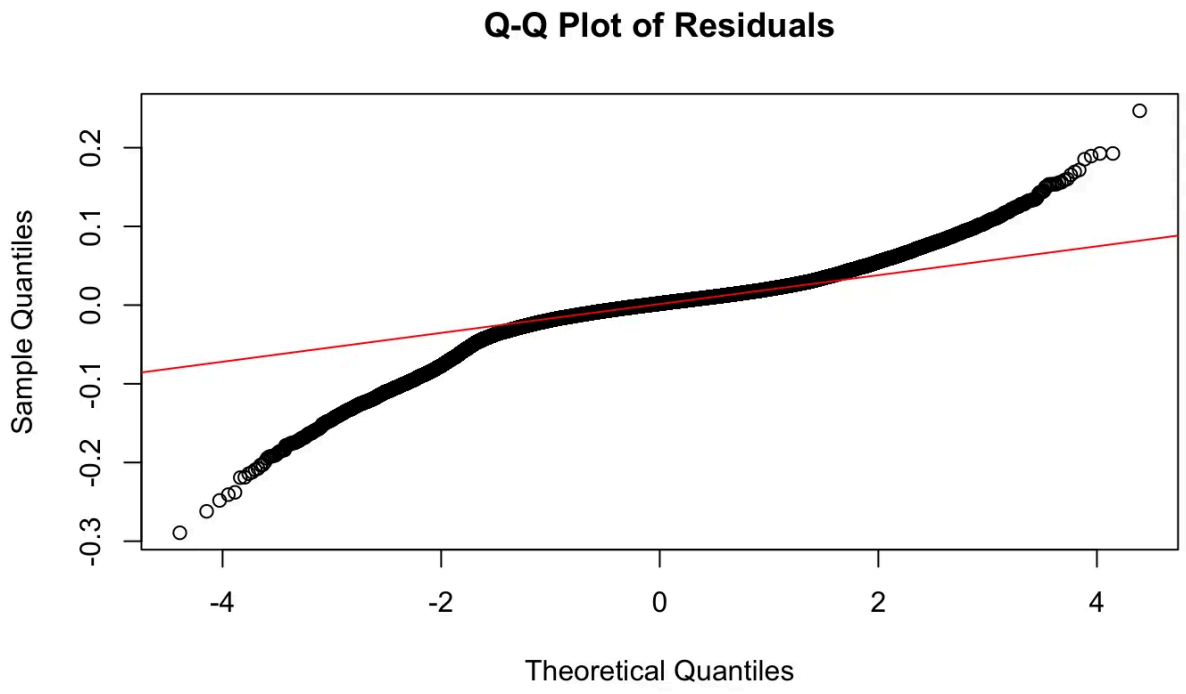
this variability, species should be retained in the model because many species exhibit significant effects and because species identity is biologically important for explaining growth differences.

Model Validation

When validating our fixed and random effects all looked good in qq and residual plots. However, we found some issues with our model residuals and qq plot.



The diagonal bands on the left are due to the discrete nature of the sqrt transformation of small values and aren't concerning. However, the spread is funnel shaped (larger on the left and get's smaller on the right). This indicates unequal variance.



The residuals have heavier tails than a normal distribution. There are more extreme values (both high and low) than expected. This means our standard errors and pvalues may be overly optimistic.

Discussion

This model indicates that tree growth in these three state's forests is predominantly determined at the individual tree level, with additional meaningful variation at the plot level. The 77.5% of variance explained by tree and plot clustering (ICCs of 60.5% and 17.0%) validates the multilevel modeling approach and reveals that ignoring hierarchical structure would produce severely biased standard errors. The negative effect of stand age indicates that growth declines as forests mature. Species-specific growth rates highlight the importance of forest composition for productivity forecasts.

Residual diagnostics reveal important model limitations. While residuals are centered at zero, heteroscedasticity is evident, with variance decreasing at higher fitted values. More concerning, the Q-Q plot shows substantial departures from normality, with heavy tails indicating far more extreme growth cases (both very slow and very fast) than the model predicts. This suggests stand age and species capture only part of the variation in tree growth, with substantial unexplained heterogeneity likely due to unmeasured factors. These violations indicate caution is warranted when interpreting standard errors and confidence intervals.

The multilevel modeling framework accounts for clustered observations and provides a foundation for incorporating additional predictors and extending scope. To be more confident in our results we need to address our model assumption issues.

Next Steps

Several extensions would strengthen this work:

- **Missing data imputation:** Implement multiple imputation for the 1.3% of live trees with missing diameter and 4.2% of trees with unexplained missing previous diameter to verify that complete-case analysis does not bias results.
- **Code Review Checks:** In order to create this dataset many joins between tables and based on various columns were created. Previous and current measurements per tree were grouped based on CN column identifiers. A unique tree ID was created and used to join across tables. These types of steps have the possibility of introducing issues in our dataset. At each step code checks should be added to verify that row counts are preserved.
- **Forest Type Grouping:** Forest type grouping was kept as it didn't change model performance and improves biological interpretability. However, we should look at other publications and the dataset to see if there is a better ecological grouping we can use.
- **Model Assumption Issues:** Heteroscedasticity was funnel shaped. As a first step other transformation methods should be investigated.
- **Additional predictors:** Incorporate tree size (initial diameter), competition indices (basal area of larger trees), and climatic variables (temperature, precipitation) to explain residual variation.
- **Geographic scope:** The analysis is limited to three small northeastern states. Expanding to Vermont, New Hampshire, Maine, and New York would improve generalizability and enable testing of broader ecological gradients.
- **Random Slopes:** There is a lot more work to be done before we can consider adding complexity with a random slope but if we get to a random intercept model we feel confident in, then we could consider investigating whether our fixed effects vary by forest type or plot.

External Validation

"Diameter Growth Models Using Minnesota Forest Inventory and Analysis Data" by Veronica C. Lessard, Ronald E. McRoberts, and Margaret R. Holdaway
https://research.fs.usda.gov/treesearch/12126

This article analyzes very similar data from Minnesota (FIA), but used a completely different model to estimate tree diameter growth. It doesn't provide support for the model we chose. It does agree in that they also found subject tree to be a significant predictor. They also found additional significant predictors we didn't use but have access to that would be good for us to try out in our model (initial diameter, physiographic class, latitude, longitude). We could also try out their transformation strategy to improve our heteroscedasticity. They used reweighted least squares to address heteroscedasticity.

Appendix

Data Analysis

Libraries

Data Load

```
# large ones causing timeout, limit scope to the 3 states we can load
# options(timeout = 600)
#me <- getFIA(states = "ME")
#vt <- getFIA(states = "VT")
#ny <- getFIA(states = "NY")
#nh <- getFIA(states = "NH")
```

```
# Initially Downloaded data using getFIA, Data then stored in offline folder

#Download separately to avoid timeout
#ct <- getFIA(states = "CT")
#ma <- getFIA(states = "MA")
#ri <- getFIA(states = "RI")
```

```
#Create Folder and save files there
#dir.create("FIA_data")
#saveRDS(ct, "FIA_data/CT_FIA.rds")
#saveRDS(ma, "FIA_data/MA_FIA.rds")
#saveRDS(ri, "FIA_data/RI_FIA.rds")
```

```
# Load data from folder into environment
ct <- readRDS("FIA_data/CT_FIA.rds")
ri <- readRDS("FIA_data/RI_FIA.rds")
ma <- readRDS("FIA_data/MA_FIA.rds")
```

```
# Combine them, each state object has multiple data tables within it
#all_states <- list(CT = ct, VT = vt, NH = nh, ME = me,
                   # MA = ma, NY = ny, RI = ri)
all_states <- list(CT = ct, MA = ma,RI = ri)
```

EDA and Data Cleaning

```
#View(all_states)
```

Connecticut

Start with one state first

Key Tables

Tables we are interested in for our model:

- PLOT - Plot information
- TREE - Individual tree measurements
- COND - Stand conditions and forest types
- PLOTGEOM - Geographic/ecological classifications
- TREE_GRM_COMPNENT - Pre-calculated growth rates
- SURVEY - Survey metadata (measurement intervals)

Tree Table

```
# Export the data frame
write_csv(ct$TREE, file = "TREE.csv")
```

Key columns to look at in TREE:

- DIA (current diameter)
- PREVDIA (previous diameter)
- SPCD (species code)
- STATUSCD (alive/dead)
- INVYR (inventory year)

Other Columns we may use:

- CN - unique identifier for a specific observation (tree measurement)
- PREV_TREE_CN - Links to the previous observation's CN
- SUBP - Subplot # within a Plot
- TREE - Tree # within subplot

After review of FIA material, for a unique identifier by tree we will use PLOT + SUBP + TREE = Permanent tree identifier

```
# Look at some examples with growth data
ct$TREE %>%
  filter(!is.na(DIA) & !is.na(PREVDIA)) %>%
  select(SPCD, DIA, PREVDIA, INVYR, STATUSCD) %>%
  head(20)
```

	SPCD	DIA	PREVDIA	INVYR	STATUSCD
1	832	5.6	5.6	1998	2
2	832	5.1	5.1	1998	2
3	372	5.4	5.2	1998	2
4	832	5.7	5.7	1998	2
5	316	6.2	6.2	1998	2
6	802	11.4	11.4	1998	2
7	261	5.9	5.8	1998	2
8	129	19.2	19.2	1998	2
9	261	6.7	6.7	1998	2
10	318	7.3	7.3	1998	2
11	261	6.5	6.5	1998	2
12	261	6.0	5.0	1998	2
13	833	11.6	11.6	1998	2
14	837	5.8	5.8	1998	2
15	833	5.3	5.3	1998	2
16	318	8.9	8.9	1998	2
17	832	14.2	14.2	1998	2
18	316	6.1	6.0	1998	2
19	316	5.0	5.0	1998	2
20	802	5.8	5.8	1998	2

```
# How many trees have both current and previous diameter?
ct$TREE %>%
  summarize(
    total_trees = n(),
```

```
    has_current_dia = sum(!is.na(DIA)),
    has_prev_dia = sum(!is.na(PREVDIA)),
    has_both = sum(!is.na(DIA) & !is.na(PREVDIA))
  )
```

```
  total_trees has_current_dia has_prev_dia has_both
1       54396           49922        23403    19480
```

```
#Calculate % missing per column
missing_summary <- ct$TREE %>%
  select(PLT_CN, TREE, SUBP, CONDID, STATUSCD, SPCD, DIA, PREVDIA,
         INVYR, TPA_UNADJ) %>%
  summarise(
    PLT_CN = 100 * sum(is.na(PLT_CN)) / n(),
    TREE = 100 * sum(is.na(TREE)) / n(),
    SUBP = 100 * sum(is.na(SUBP)) / n(),
    CONDID = 100 * sum(is.na(CONDID)) / n(),
    STATUSCD = 100 * sum(is.na(STATUSCD)) / n(),
    SPCD = 100 * sum(is.na(SPCD)) / n(),
    DIA = 100 * sum(is.na(DIA)) / n(),
    PREVDIA = 100 * sum(is.na(PREVDIA)) / n(),
    INVYR = 100 * sum(is.na(INVYR)) / n(),
    TPA_UNADJ = 100 * sum(is.na(TPA_UNADJ)) / n()
  )

# View Missing Summary Table
t(missing_summary)
```

```
               [,1]
PLT_CN     0.000000
TREE       0.000000
SUBP       0.000000
CONDID     0.000000
STATUSCD   0.000000
SPCD       0.000000
DIA        8.224869
PREVDIA   56.976616
INVYR      0.000000
TPA_UNADJ 11.552320
```

Missing Data Summary:

- **0% missing**: PLT_CN, TREE, SUBP, CONDID, STATUSCD, SPCD, INVYR (core identifiers)
- **8.2% missing**: DIA
- **57% missing**: PREVDIA (only remeasured trees have this)
- **11.6% missing**: TPA_UNADJ (trees per acre adjustment)

It seems this table is every tree recorded but there are many trees which didn't have more than one measurement. With nearly 60% missing repeat measurements imputation is not recommended.

Lets see if there are patterns in what is missing:

```
# Look at all inventory years
result <- ct$TREE %>%
  count(INVYR) %>%
  arrange(INVYR)

print(result)
```

```
   INVYR     n
1   1985  7840
2   1998 10986
3   2003  1449
4   2004  1456
5   2005  1543
6   2006  2285
7   2007  2465
8   2008  2149
9   2009  2196
10  2010  1915
11  2011  1679
12  2012  1807
13  2013  2004
14  2014  1336
15  2015  1477
16  2016  1423
17  2017  1073
18  2018  1399
19  2019  1282
20  2020  1382
```

```
21   2021   1374
22   2022   1502
23   2023   1338
24   2024   1036
```

```r
# See which years have remeasurements
ct$TREE %>%
  group_by(INVYR) %>%
  summarize(
    total = n(),
    has_prevdia = sum(!is.na(PREVDIA)),
    pct_with_prev = round(100 * has_prevdia / total, 1)
  ) %>%
  arrange(INVYR)
```

```
# A tibble: 24 x 4
   INVYR total has_prevdia pct_with_prev
   <int> <int>       <int>         <dbl>
 1  1985  7840           0           0
 2  1998 10986          93           0.8
 3  2003  1449           0           0
 4  2004  1456           0           0
 5  2005  1543           0           0
 6  2006  2285           0           0
 7  2007  2465           0           0
 8  2008  2149        2026          94.3
 9  2009  2196        1993          90.8
10  2010  1915        1653          86.3
# i 14 more rows
```

The Pattern:

- **1985-2007**: First measurements only (no PREVDIA - 0% have previous diameter with exception of a few in 1998)
- **2008-2020**: Includes remeasurements (84-95% have PREVDIA)

We should at least limit the scope of our dataset to 2008 - 2020. We may restrict it further based on missingness in the other two state datasets.

Missingness 2008 - 2020

```
# How many trees have both current and previous diameter (2008-2020)?
ct$TREE %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%  # Filter to remeasurement years
  summarize(
    total_trees = n(),
    has_current_dia = sum(!is.na(DIA)),
    has_prev_dia = sum(!is.na(PREVDIA)),
    has_both = sum(!is.na(DIA) & !is.na(PREVDIA))
  )
```

```
  total_trees has_current_dia has_prev_dia has_both
1       21122           18084        18815    15777
```

```
# Calculate % missing per column (2008-2020 only)
missing_summary_limited <- ct$TREE %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%  # Filter to remeasurement years
  select(PLT_CN, TREE, SUBP, CONDID, STATUSCD, SPCD, DIA, PREVDIA,
         INVYR, TPA_UNADJ) %>%
  summarise(
    PLT_CN = 100 * sum(is.na(PLT_CN)) / n(),
    TREE = 100 * sum(is.na(TREE)) / n(),
    SUBP = 100 * sum(is.na(SUBP)) / n(),
    CONDID = 100 * sum(is.na(CONDID)) / n(),
    STATUSCD = 100 * sum(is.na(STATUSCD)) / n(),
    SPCD = 100 * sum(is.na(SPCD)) / n(),
    DIA = 100 * sum(is.na(DIA)) / n(),
    PREVDIA = 100 * sum(is.na(PREVDIA)) / n(),
    INVYR = 100 * sum(is.na(INVYR)) / n(),
    TPA_UNADJ = 100 * sum(is.na(TPA_UNADJ)) / n()
  )

# View Missing Summary Table
t(missing_summary_limited)
```

```
               [,1]
PLT_CN      0.00000
TREE        0.00000
SUBP        0.00000
CONDID      0.00000
```

```
STATUSCD    0.00000
SPCD        0.00000
DIA        14.38311
PREVDIA    10.92226
INVYR       0.00000
TPA_UNADJ  14.38311
```

What's still missing?

Let's look at the STATUSCD (alive/dead) column to see if the missingness is due to tree death.

FIA Status Codes:

Let's Look at the codes in our data:

```
unique_statuscd <- ct$TREE %>%
  distinct(STATUSCD) %>%
  pull(STATUSCD)

print(unique_statuscd)
```

```
[1] 1 2 0 3
```

It Looks like there is an extra code "0" and "3"

Based other FIA online materials **STATUSCD = 3** typically means **"Removed/Cut"**, **STATUSCD = 0** means not tallied due to natural causes.

Let's check we have all the same codes for our date range:

```
unique_statuscd_2008_2020 <- ct$TREE %>%
  filter(INVYR >= 2008, INVYR <= 2020) %>%
  distinct(STATUSCD) %>%
  pull(STATUSCD)

unique_statuscd_2008_2020
```

```
[1] 1 2 0
```

0 remains but 3 is not used in recent years. Perhaps 0 has replaced 3 for removed/cut?

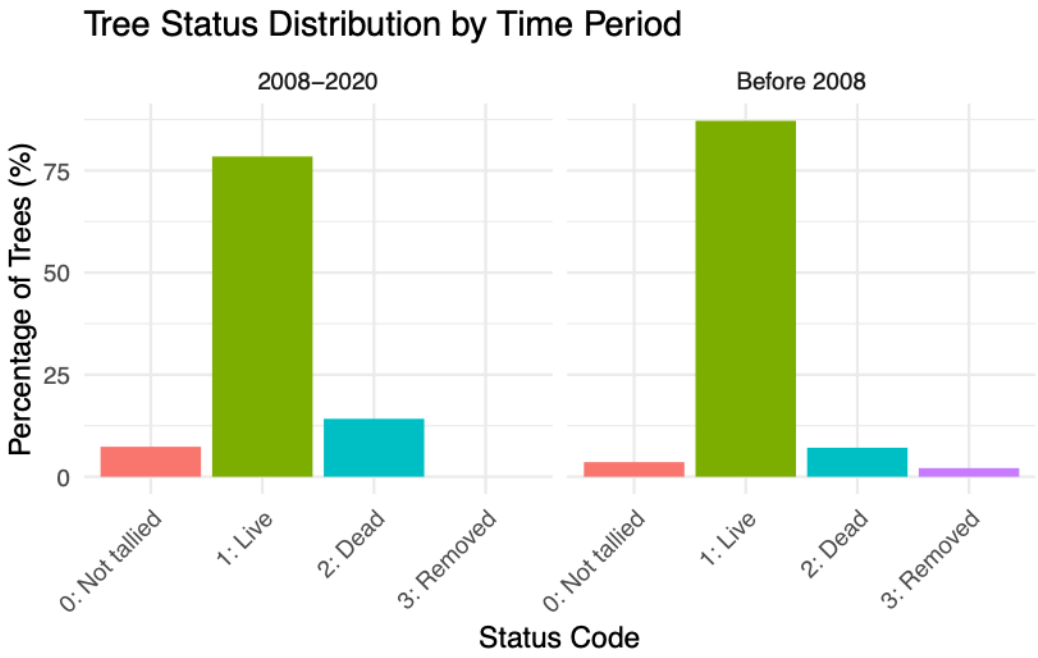
Let's check what codes were in earlier years and see if there were zeros

```r
# Create a dataset with status codes grouped by time period
status_comparison <- ct$TREE %>%
  mutate(
    time_period = case_when(
      INVYR < 2008 ~ "Before 2008",
      INVYR >= 2008 ~ "2008-2020"
    ),
    status_label = case_when(
      STATUSCD == 0 ~ "0: Not tallied",
      STATUSCD == 1 ~ "1: Live",
      STATUSCD == 2 ~ "2: Dead",
      STATUSCD == 3 ~ "3: Removed",
      TRUE ~ as.character(STATUSCD)
    )
  ) %>%
  count(time_period, status_label) %>%
  group_by(time_period) %>%
  mutate(
    pct = 100 * n / sum(n)
  )

# View the data
status_comparison
```

```
# A tibble: 7 x 4
# Groups:   time_period [2]
  time_period status_label       n   pct
  <chr>       <chr>          <int> <dbl>
1 2008-2020   0: Not tallied  1939  7.35
2 2008-2020   1: Live        20688 78.4
3 2008-2020   2: Dead         3745 14.2
4 Before 2008 0: Not tallied  1003  3.58
5 Before 2008 1: Live        24434 87.2
6 Before 2008 2: Dead         1992  7.11
7 Before 2008 3: Removed       595  2.12
```

```r
# Alternative: Faceted plot (separate panels)
ggplot(status_comparison, aes(x = status_label, y = pct, fill = status_label)) +
  geom_col() +
  facet_wrap(~time_period) +
  labs(
```

```
    title = "Tree Status Distribution by Time Period",
    x = "Status Code",
    y = "Percentage of Trees (%)"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "none"
  )
```



It loooks like there was less untallied prior to 2008 3.6% –> 7.4%

Also Removed trees before to after 2008 2.1% –> 0%

It's plausible that removed trees are now part of the 0 category but may not account for the entire category. Since online materials describe the 0 status code as untallied due to natural causes I think we can assume trees in this category are dead trees though whether they are dead due to removal or natural causes is not certain.

Our Data:

- **0** = Dead tree (gone)
- **1** = Live tree
- **2** = Dead tree (still standing)

Let's get back to the missing data. Now that we know that 0 and 2 are dead trees let's see if this relates to not having a measurement.

```
# Of the trees missing DIA, what % are dead/alive?
ct$TREE %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%
  filter(is.na(DIA)) %>%  # Only look at trees missing DIA
  summarize(
    total_missing_dia = n(),
    n_dead = sum(STATUSCD %in% c(0, 2), na.rm = TRUE),  # 0 or 2 = dead
    n_alive = sum(STATUSCD == 1, na.rm = TRUE),
    pct_dead = 100 * n_dead / total_missing_dia,
    pct_alive = 100 * n_alive / total_missing_dia
  ) %>%
print()
```

```
  total_missing_dia n_dead n_alive pct_dead pct_alive
1              3038   2999      39 98.71626  1.283739
```

```
# Of the trees missing PREVDIA, what % are dead/alive?
ct$TREE %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%
  filter(is.na(PREVDIA)) %>%  # Only look at trees missing PREVDIA
  summarize(
    total_missing_prevdia = n(),
    n_dead = sum(STATUSCD %in% c(0, 2), na.rm = TRUE),  # 0 or 2 = dead
    n_alive = sum(STATUSCD == 1, na.rm = TRUE),
    pct_dead = 100 * n_dead / total_missing_prevdia,
    pct_alive = 100 * n_alive / total_missing_prevdia
  ) %>%
print()
```

```
  total_missing_prevdia n_dead n_alive pct_dead pct_alive
1                  2307     97    2210 4.204595  95.79541
```

Missing PREVDIA (Previous Diameter)

- "ingrowth" trees
- **too small to measure in the previous inventory** (< 5" DBH threshold)
- grew large enough by 2008+ to be included in measurements
- can't calculate growth rates for these trees (no baseline)

Missing DIA (Current Diameter)

- Almost all trees missing current diameter are **dead or removed**
- You can't measure the diameter of a tree that's gone (STATUSCD=0) or too deteriorated (STATUSCD=2)
- The 39 live trees (1.3%) with missing DIA might be measurement errors or trees that couldn't be accessed

We understand the missing data problem now. Let's rerun for the combined TREE datasets across all states and review.

Missingness in full TREE dataset

```
# Combine TREE tables from all three states
combined_tree <- bind_rows(
  ct$TREE %>% mutate(STATE = "CT"),
  ma$TREE %>% mutate(STATE = "MA"),
  ri$TREE %>% mutate(STATE = "RI")
)

combined_tree %>%
  group_by(INVYR) %>%
  summarize(
    total = n(),
    has_prevdia = sum(!is.na(PREVDIA)),
    pct_with_prev = 100 * has_prevdia / total
  ) %>%
  arrange(INVYR)
```

```
# A tibble: 24 x 4
   INVYR total has_prevdia pct_with_prev
   <int> <int>       <int>         <dbl>
 1  1985 22452           0          0
 2  1998 39056         397          1.02
 3  2003  5282           0          0
 4  2004  4906           0          0
 5  2005  5075           0          0
 6  2006  7403           0          0
 7  2007  9120           0          0
 8  2008  7298        6849         93.8
 9  2009  7405        6809         92.0
10  2010  6615        5490         83.0
# i 14 more rows
```

```r
# Combine TREE tables from all three states
combined_tree <- bind_rows(
  ct$TREE %>% mutate(STATE = "CT"),
  ma$TREE %>% mutate(STATE = "MA"),
  ri$TREE %>% mutate(STATE = "RI")
)
```

```r
#When does missing data start for the three different states?

# Calculate % missing per column (combined data)
missing_summary_combined <- combined_tree %>%
  select(PLT_CN, TREE, SUBP, CONDID, STATUSCD, SPCD, DIA, PREVDIA,
         INVYR, TPA_UNADJ) %>%
  summarise(
    PLT_CN      = 100 * sum(is.na(PLT_CN)) / n(),
    TREE        = 100 * sum(is.na(TREE)) / n(),
    SUBP        = 100 * sum(is.na(SUBP)) / n(),
    CONDID      = 100 * sum(is.na(CONDID)) / n(),
    STATUSCD    = 100 * sum(is.na(STATUSCD)) / n(),
    SPCD        = 100 * sum(is.na(SPCD)) / n(),
    DIA         = 100 * sum(is.na(DIA)) / n(),
    PREVDIA     = 100 * sum(is.na(PREVDIA)) / n(),
    INVYR       = 100 * sum(is.na(INVYR)) / n(),
    TPA_UNADJ   = 100 * sum(is.na(TPA_UNADJ)) / n()
  )

# View Missing Summary Table (raw output)
missing_summary_combined %>% as.data.frame()
```

```
  PLT_CN TREE SUBP CONDID STATUSCD SPCD      DIA PREVDIA INVYR TPA_UNADJ
1      0    0    0      0        0    0 8.615559 56.2638     0  12.32039
```

```r
t(as.data.frame(missing_summary_combined))
```

```
              [,1]
PLT_CN    0.000000
TREE      0.000000
SUBP      0.000000
CONDID    0.000000
STATUSCD  0.000000
SPCD      0.000000
```

```
DIA        8.615559
PREVDIA   56.263796
INVYR      0.000000
TPA_UNADJ 12.320385
```

```r
# Remeasurement stats
combined_tree %>%
  group_by(INVYR) %>%
  summarize(
    total         = n(),
    has_prevdia   = sum(!is.na(PREVDIA)),
    pct_with_prev = 100 * has_prevdia / total
  ) %>%
  arrange(INVYR) %>%
  as.data.frame() %>%
  kable(caption = "Remeasurement Stats by Inventory Year")
```

Table 1: Remeasurement Stats by Inventory Year

INVYR	total	has_prevdia	pct_with_prev
1985	22452	0	0.000000
1998	39056	397	1.016489
2003	5282	0	0.000000
2004	4906	0	0.000000
2005	5075	0	0.000000
2006	7403	0	0.000000
2007	9120	0	0.000000
2008	7298	6849	93.847629
2009	7405	6809	91.951384
2010	6615	5490	82.993197
2011	6058	4999	82.518983
2012	6578	5548	84.341745
2013	6724	6024	89.589530
2014	4592	4164	90.679442
2015	5054	4408	87.218045
2016	4683	4252	90.796498
2017	4135	3563	86.166868
2018	4708	4155	88.254036
2019	4948	4472	90.379952
2020	4591	4061	88.455674
2021	4512	3825	84.773936

INVYR	total	has_prevdia	pct_with_prev
2022	4977	4338	87.160940
2023	4579	3969	86.678314
2024	4089	3519	86.060161

```
# Patterns by state
combined_tree %>%
  group_by(STATE, INVYR) %>%
  summarize(
    total         = n(),
    has_prevdia   = sum(!is.na(PREVDIA)),
    pct_with_prev = 100 * has_prevdia / total
  ) %>%
  arrange(STATE, INVYR) %>%
  as.data.frame() %>%
  kable(caption = "Patterns of Remeasurements by State and Inventory Year")
```

```
`summarise()` has grouped output by 'STATE'. You can override using the
`.groups` argument.
```

Table 2: Patterns of Remeasurements by State and Inventory Year

STATE	INVYR	total	has_prevdia	pct_with_prev
CT	1985	7840	0	0.0000000
CT	1998	10986	93	0.8465319
CT	2003	1449	0	0.0000000
CT	2004	1456	0	0.0000000
CT	2005	1543	0	0.0000000
CT	2006	2285	0	0.0000000
CT	2007	2465	0	0.0000000
CT	2008	2149	2026	94.2764076
CT	2009	2196	1993	90.7559199
CT	2010	1915	1653	86.3185379
CT	2011	1679	1399	83.3234068
CT	2012	1807	1512	83.6745988
CT	2013	2004	1724	86.0279441
CT	2014	1336	1197	89.5958084
CT	2015	1477	1297	87.8131347
CT	2016	1423	1348	94.7294448

STATE	INVYR	total	has_prevdia	pct_with_prev
CT	2017	1073	987	91.9850885
CT	2018	1399	1260	90.0643317
CT	2019	1282	1190	92.8237129
CT	2020	1382	1229	88.9290883
CT	2021	1374	1145	83.3333333
CT	2022	1502	1297	86.3515313
CT	2023	1338	1182	88.3408072
CT	2024	1036	871	84.0733591
MA	1985	11552	0	0.0000000
MA	1998	23588	264	1.1192132
MA	2003	3086	0	0.0000000
MA	2004	2856	0	0.0000000
MA	2005	3008	0	0.0000000
MA	2006	4370	0	0.0000000
MA	2007	5592	0	0.0000000
MA	2008	4141	3861	93.2383482
MA	2009	4430	4122	93.0474041
MA	2010	3866	3225	83.4195551
MA	2011	3603	3087	85.6786012
MA	2012	4009	3347	83.4871539
MA	2013	3842	3490	90.8381052
MA	2014	2670	2454	91.9101124
MA	2015	2995	2651	88.5141903
MA	2016	2691	2445	90.8584169
MA	2017	2471	2098	84.9048968
MA	2018	2698	2371	87.8799110
MA	2019	3141	2896	92.1999363
MA	2020	2575	2289	88.8932039
MA	2021	2582	2182	84.5081332
MA	2022	2879	2547	88.4682181
MA	2023	2675	2273	84.9719626
MA	2024	2417	2163	89.4911047
RI	1985	3060	0	0.0000000
RI	1998	4482	40	0.8924587
RI	2003	747	0	0.0000000
RI	2004	594	0	0.0000000
RI	2005	524	0	0.0000000
RI	2006	748	0	0.0000000
RI	2007	1063	0	0.0000000
RI	2008	1008	962	95.4365079
RI	2009	779	694	89.0885751

STATE	INVYR	total	has_prevdia	pct_with_prev
RI	2010	834	612	73.3812950
RI	2011	776	513	66.1082474
RI	2012	762	689	90.4199475
RI	2013	878	810	92.2551253
RI	2014	586	513	87.5426621
RI	2015	582	460	79.0378007
RI	2016	569	459	80.6678383
RI	2017	591	478	80.8798646
RI	2018	611	524	85.7610475
RI	2019	525	386	73.5238095
RI	2020	634	543	85.6466877
RI	2021	556	498	89.5683453
RI	2022	596	494	82.8859060
RI	2023	566	514	90.8127208
RI	2024	636	485	76.2578616

```r
# Compare missing data across states
combined_tree %>%
  group_by(STATE) %>%
  summarise(
    n_trees            = n(),
    pct_missing_DIA    = 100 * sum(is.na(DIA)) / n(),
    pct_missing_PREVDIA = 100 * sum(is.na(PREVDIA)) / n()
  ) %>%
  as.data.frame() %>%
  kable(caption = "Missing Data Percentages by State")
```

Table 3: Missing Data Percentages by State

STATE	n_trees	pct_missing_DIA	pct_missing_PREVDIA
CT	54396	8.224870	56.97662
MA	107737	8.598717	55.66518
RI	22707	9.631391	57.39640

All states start remeasurements in 2008.

Check status codes are the same:

```r
unique_statuscd_2008_2020_all_states <- combined_tree %>%
  filter(INVYR >= 2008, INVYR <= 2020) %>%
  distinct(STATUSCD) %>%
  pull(STATUSCD)

unique_statuscd_2008_2020_all_states
```

```
[1] 1 2 0
```

Status codes are the same for all states

```r
# Of the trees missing DIA, what % are dead/alive?
dia_summary <- combined_tree %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%
  filter(is.na(DIA)) %>%  # Only look at trees missing DIA
  summarize(
    total_missing_dia = n(),
    n_dead = sum(STATUSCD %in% c(0, 2), na.rm = TRUE),  # 0 or 2 = dead
    n_alive = sum(STATUSCD == 1, na.rm = TRUE),
    pct_dead = 100 * n_dead / total_missing_dia,
    pct_alive = 100 * n_alive / total_missing_dia
  )

kable(dia_summary, caption = "Percentage of Dead/Alive Trees Missing DIA")
```

Table 4: Percentage of Dead/Alive Trees Missing DIA

total_missing_dia	n_dead	n_alive	pct_dead	pct_alive
10807	10615	192	98.22337	1.776626

```r
# Of the trees missing PREVDIA, what % are dead/alive?
combined_tree %>%
  filter(INVYR >= 2008 & INVYR <= 2020) %>%
  filter(is.na(PREVDIA)) %>%  # Only look at trees missing PREVDIA
  summarize(
    total_missing_prevdia = n(),
    n_dead = sum(STATUSCD %in% c(0, 2), na.rm = TRUE),  # 0 or 2 = dead
    n_alive = sum(STATUSCD == 1, na.rm = TRUE),
    pct_dead = 100 * n_dead / total_missing_prevdia,
```

```
    pct_alive = 100 * n_alive / total_missing_prevdia
  ) %>%
print()
```

```
  total_missing_prevdia n_dead n_alive pct_dead pct_alive
1                  8595    400    8195 4.653869  95.34613
```

Missing DIA (10,807 trees)

- **Dead: 10,615 trees (98.22%)** Expected - can't measure dead/removed trees
- **Alive: 192 trees (1.78%)** - Small number of measurement errors

  (This is something we could impute but it is very small so if we don't have time for imputation it shouldn't strongly affect our overall conclusions)

Missing PREVDIA (8,595 trees)

- **Dead: 400 trees (4.65%)** - Some trees died before being remeasured
- **Alive: 8,195 trees (95.35%)** Expected - many of thease are likely **ingrowth trees**

```
# Strict comparison: Only NA PREVDIA vs valid PREVDIA > 0
diameter_comparison <- combined_tree %>%
  filter(
    INVYR >= 2008 & INVYR <= 2020,
    STATUSCD == 1,           # Live trees only
    !is.na(DIA)              # Has current diameter
  ) %>%
  mutate(
    tree_type = case_when(
      is.na(PREVDIA) ~ "Ingrowth",
      PREVDIA > 0 ~ "Remeasured",
      TRUE ~ NA_character_  # Exclude PREVDIA = 0
    )
  ) %>%
  filter(!is.na(tree_type))  # Remove PREVDIA = 0 trees

# Summary table
diameter_summary <- diameter_comparison %>%
  group_by(tree_type) %>%
  summarise(
    n = n(),
    mean_dia = round(mean(DIA, na.rm = TRUE), 2),
```

```
    median_dia = round(median(DIA, na.rm = TRUE), 2)
  ) %>%
  as.data.frame()

# Display table
kable(diameter_summary,
      caption = "Diameter Summary: Ingrowth vs Remeasured Trees",
      col.names = c("Tree Type", "N", "Mean (in)", "Median (in)"))
```

Table 5: Diameter Summary: Ingrowth vs Remeasured Trees

Tree Type	N	Mean (in)	Median (in)
Ingrowth	8195	6.92	5.6
Remeasured	49011	9.65	8.8

Smaller trees are in the ingrowth category indicating that these are likely trees which were too small to measure in previous years. We will move forward for now but a good next step for our analysis would be to investigate this category and get more certainty on the missing data. This may also be a category some data that would be a good candidate for imputation.

Remeasurement Checks

```
# intervals dataset
intervals <- combined_tree %>%
  filter(INVYR >= 2008) %>%
  left_join(
    combined_tree %>% select(CN, INVYR) %>% rename(PREV_INVYR = INVYR),
    by = c("PREV_TRE_CN" = "CN")
  ) %>%
  mutate(years_between = INVYR - PREV_INVYR) %>%
  filter(!is.na(years_between), years_between > 0)

# Check intervals by state and year
intervals %>%
  group_by(STATE, INVYR) %>%
  summarise(
    n = n(),
    mean_interval = round(mean(years_between), 1),
    median_interval = median(years_between),
    min_interval = min(years_between),
```

```
    max_interval = max(years_between)
  ) %>%
  arrange(STATE, INVYR) %>%
  as.data.frame() %>%
  kable(caption = "Intervals by State and Year") # add kable here
```

```
`summarise()` has grouped output by 'STATE'. You can override using the
`.groups` argument.
```

Table 6: Intervals by State and Year

STATE	INVYR	n	mean_interval	median_interval	min_interval	max_interval
CT	2008	2026	4.7	5	4	5
CT	2009	1993	4.4	4	4	5
CT	2010	1653	4.2	4	4	5
CT	2011	1399	4.6	5	4	5
CT	2012	1512	5.0	5	5	5
CT	2013	1724	5.0	5	5	5
CT	2014	1197	5.0	5	5	5
CT	2015	1297	5.5	6	5	6
CT	2016	1348	5.8	6	5	6
CT	2017	987	6.0	6	6	6
CT	2018	1260	6.2	6	6	7
CT	2019	1190	6.6	7	6	7
CT	2020	1229	7.0	7	7	7
CT	2021	1145	7.0	7	7	7
CT	2022	1297	7.0	7	7	7
CT	2023	1182	7.0	7	7	7
CT	2024	871	7.0	7	7	7
MA	2008	3861	4.8	5	4	5
MA	2009	4122	4.5	4	4	5
MA	2010	3225	4.2	4	4	5
MA	2011	3087	4.5	5	4	5
MA	2012	3347	5.0	5	5	5
MA	2013	3490	5.0	5	5	5
MA	2014	2454	5.0	5	5	5
MA	2015	2651	5.5	5	5	6
MA	2016	2445	5.8	6	5	6
MA	2017	2098	6.0	6	6	6
MA	2018	2371	6.3	6	6	7

STATE	INVYR	n	mean_interval	median_interval	min_interval	max_interval
MA	2019	2896	6.6	7	6	7
MA	2020	2289	7.0	7	7	7
MA	2021	2182	7.0	7	7	7
MA	2022	2547	7.0	7	7	7
MA	2023	2273	7.0	7	7	7
MA	2024	2163	7.0	7	7	7
RI	2008	962	4.8	5	4	5
RI	2009	694	4.5	5	4	5
RI	2010	612	4.3	4	4	5
RI	2011	513	4.5	5	4	5
RI	2012	689	5.0	5	5	5
RI	2013	810	5.0	5	5	5
RI	2014	513	5.0	5	5	5
RI	2015	460	5.3	5	5	6
RI	2016	459	5.9	6	5	6
RI	2017	478	6.0	6	6	6
RI	2018	524	6.2	6	6	7
RI	2019	386	6.4	6	6	7
RI	2020	543	7.0	7	7	7
RI	2021	498	7.0	7	7	7
RI	2022	494	7.0	7	7	7
RI	2023	514	7.0	7	7	7
RI	2024	485	7.0	7	7	7

```
# Are intervals consistent within each state?
intervals %>%
  group_by(STATE) %>%
  summarise(
    n = n(),
    mean_interval = round(mean(years_between), 1),
    sd_interval = round(sd(years_between), 1),
    most_common = median(years_between)
  ) %>%
  as.data.frame() %>%
  kable(caption = "Interval Consistency Statistics by State") # add kable here
```

Table 7: Interval Consistency Statistics by State

STATE	n	mean_interval	sd_interval	most_common
CT	23310	5.6	1.1	5
MA	47501	5.6	1.1	5
RI	9634	5.6	1.0	5

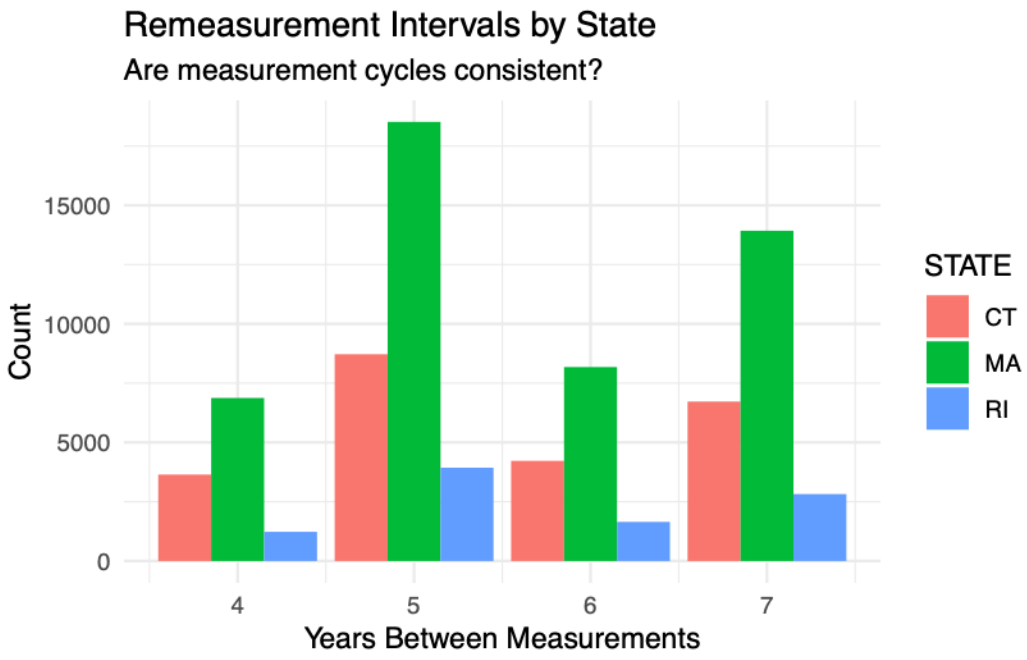
```
# Distribution of intervals by state
intervals %>%
  count(STATE, years_between) %>%
  group_by(STATE) %>%
  mutate(pct = round(100 * n / sum(n), 1)) %>%
  arrange(STATE, years_between) %>%
  as.data.frame() %>%
  kable(caption = "Distribution of Intervals by State") # add kable here
```

Table 8: Distribution of Intervals by State

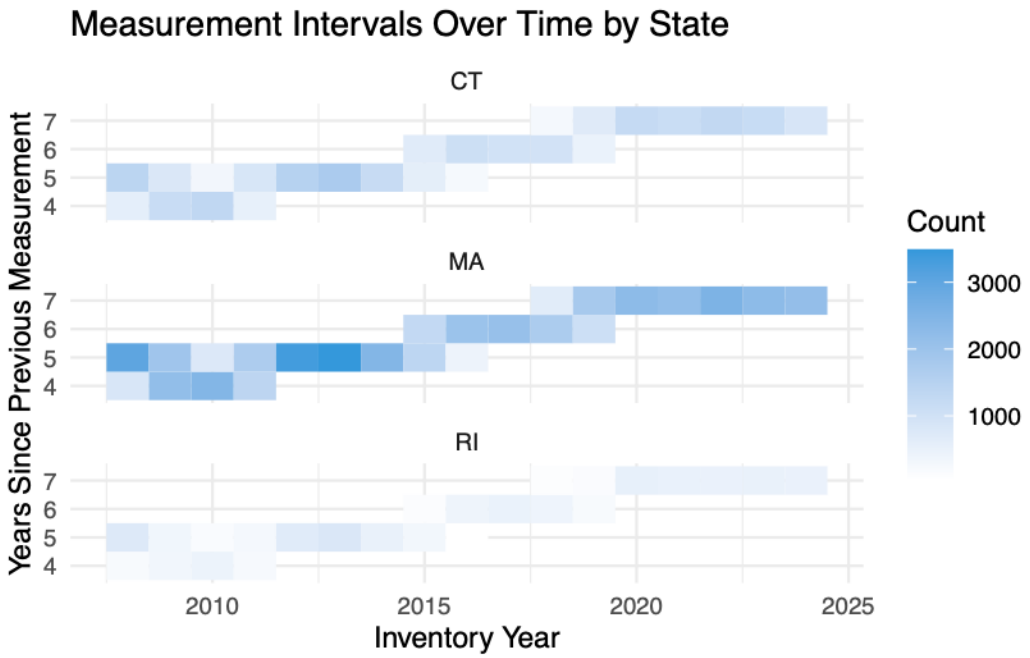
STATE	years_between	n	pct
CT	4	3644	15.6
CT	5	8719	37.4
CT	6	4223	18.1
CT	7	6724	28.8
MA	4	6875	14.5
MA	5	18520	39.0
MA	6	8178	17.2
MA	7	13928	29.3
RI	4	1229	12.8
RI	5	3934	40.8
RI	6	1649	17.1
RI	7	2822	29.3

```
# Visualization: Intervals by state
ggplot(intervals, aes(x = years_between, fill = STATE)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Remeasurement Intervals by State",
    subtitle = "Are measurement cycles consistent?",
    x = "Years Between Measurements",
    y = "Count"
```

```
  ) +
  theme_minimal()
```




```
# Heatmap: Which years have which intervals?
intervals %>%
  count(STATE, INVYR, years_between) %>%
  ggplot(aes(x = INVYR, y = as.factor(years_between), fill = n)) +
  geom_tile() +
  facet_wrap(~STATE, ncol = 1) +
  scale_fill_gradient(low = "white", high = "#3498DB") +
  labs(
    title = "Measurement Intervals Over Time by State",
    x = "Inventory Year",
    y = "Years Since Previous Measurement",
    fill = "Count"
  ) +
  theme_minimal()
```




```
# Show which previous years contribute to each current year
intervals %>%
  filter(INVYR %in% c(2018, 2019, 2020)) %>%  # Pick a few recent years
  count(STATE, INVYR, PREV_INVYR, years_between) %>%
  arrange(STATE, INVYR, years_between) %>%
  kable(caption = "Previous Measurement Years Contributing to 2018-2020")
```

Table 9: Previous Measurement Years Contributing to 2018-2020

STATE	INVYR	PREV_INVYR	years_between	n
CT	2018	2012	6	976
CT	2018	2011	7	284
CT	2019	2013	6	474
CT	2019	2012	7	716
CT	2020	2013	7	1229
MA	2018	2012	6	1704
MA	2018	2011	7	667
MA	2019	2013	6	1089
MA	2019	2012	7	1807
MA	2020	2013	7	2289
RI	2018	2012	6	402
RI	2018	2011	7	122

STATE	INVYR	PREV_INVYR	years_between	n
RI	2019	2013	6	220
RI	2019	2012	7	166
RI	2020	2013	7	543

```
# Summary: For each inventory year, what mix of intervals exists?
intervals %>%
  group_by(STATE, INVYR) %>%
  summarise(
    n_trees = n(),
    intervals_present = paste(sort(unique(years_between)), collapse = ", "),
    most_common = median(years_between),
    .groups = "drop"
  ) %>%
  kable(caption = "Mix of Measurement Intervals by State and Year")
```

Table 10: Mix of Measurement Intervals by State and Year

STATE	INVYR	n_trees	intervals_present	most_common
CT	2008	2026	4, 5	5
CT	2009	1993	4, 5	4
CT	2010	1653	4, 5	4
CT	2011	1399	4, 5	5
CT	2012	1512	5	5
CT	2013	1724	5	5
CT	2014	1197	5	5
CT	2015	1297	5, 6	6
CT	2016	1348	5, 6	6
CT	2017	987	6	6
CT	2018	1260	6, 7	6
CT	2019	1190	6, 7	7
CT	2020	1229	7	7
CT	2021	1145	7	7
CT	2022	1297	7	7
CT	2023	1182	7	7
CT	2024	871	7	7
MA	2008	3861	4, 5	5
MA	2009	4122	4, 5	4
MA	2010	3225	4, 5	4
MA	2011	3087	4, 5	5

STATE	INVYR	n_trees	intervals_present	most_common
MA	2012	3347	5	5
MA	2013	3490	5	5
MA	2014	2454	5	5
MA	2015	2651	5, 6	5
MA	2016	2445	5, 6	6
MA	2017	2098	6	6
MA	2018	2371	6, 7	6
MA	2019	2896	6, 7	7
MA	2020	2289	7	7
MA	2021	2182	7	7
MA	2022	2547	7	7
MA	2023	2273	7	7
MA	2024	2163	7	7
RI	2008	962	4, 5	5
RI	2009	694	4, 5	5
RI	2010	612	4, 5	4
RI	2011	513	4, 5	5
RI	2012	689	5	5
RI	2013	810	5	5
RI	2014	513	5	5
RI	2015	460	5, 6	5
RI	2016	459	5, 6	6
RI	2017	478	6	6
RI	2018	524	6, 7	6
RI	2019	386	6, 7	6
RI	2020	543	7	7
RI	2021	498	7	7
RI	2022	494	7	7
RI	2023	514	7	7
RI	2024	485	7	7

```
# Check if specific plots have consistent intervals
intervals %>%
  group_by(STATE, PLOT) %>%
  summarise(
    n_measurements = n(),
    intervals = paste(sort(unique(years_between)), collapse = ", "),
    sd_interval = round(sd(years_between), 2),
    .groups = "drop"
  ) %>%
```

```
  filter(n_measurements > 1) %>%  # Only plots with multiple remeasurements
  arrange(STATE, PLOT) %>%
  head(20) %>%
  kable(caption = "Measurement Interval Consistency Within Plots (First 20 plots)")
```

Table 11: Measurement Interval Consistency Within Plots (First 20 plots)

STATE	PLOT	n_measurements	intervals	sd_interval
CT	3	116	5, 7	0.99
CT	4	82	4, 6, 7	1.26
CT	5	11	4	0.00
CT	6	46	4, 7	1.52
CT	8	93	5, 7	0.93
CT	10	113	4, 6, 7	1.25
CT	15	104	4, 6, 7	1.25
CT	18	87	4, 5, 7	1.21
CT	19	90	4, 5, 7	1.25
CT	20	113	4, 5, 7	1.23
CT	21	85	4, 6, 7	1.22
CT	24	76	5, 7	0.92
CT	26	70	4, 6, 7	1.26
CT	32	16	7	0.00
CT	33	52	4, 7	1.51
CT	34	54	5, 7	0.97
CT	36	112	4, 6, 7	1.26
CT	41	21	5, 7	1.00
CT	44	9	7	0.00
CT	47	99	4, 5, 7	1.25

1. Multiple intervals exist within the same year

Looking at 2020 for example:

- **CT 2020:** Mix of 7-year intervals (last measured 2013)
- **MA 2020:** Mix of 7-year intervals (last measured 2013)
- **RI 2020:** Mix of 7-year intervals (last measured 2013)

But in 2018 and 2019, you see:

- **2018:** Both 6-year (from 2012) and 7-year (from 2011) intervals
- **2019:** Both 6-year (from 2013) and 7-year (from 2012) intervals

2. Individual plots ARE fairly consistent

The standard deviations are low (0.92 - 1.52), meaning:

- Most plots stick to similar intervals across remeasurements
- Example: Plot 51 has `sd = 0.00` - perfectly consistent 4-year intervals
- Example: Plot 32 has `sd = 0.00` - perfectly consistent 6-year intervals
- Most plots vary by only ~1 year between measurement cycles

3. Different plots have different "schedules"

- Some plots: Consistent 4-year cycles (Plot 51, 101, 201, etc.)
- Some plots: Consistent 5-7 year cycles
- A few plots: Mix of 4, 5, 6, and 7-year intervals

State Patterns:

1. **All three states follow similar patterns** - they're on the same measurement schedule
2. **2018:** Dominated by 6-year intervals (plots last measured in 2012)
3. **2019:** Mixed, shifting toward 7-year intervals (plots from 2012 and 2013)
4. **2020:** All 7-year intervals (plots last measured in 2013)

Interval Conclusion: Must include time interval as a variable unless we annualize the growth rate

Tree Growth Table

TREE_GRM_COMPENT

We also have the TREE_GRM_COMPNENT table. This should only include remeasured trees. We've analyzed the full TREE table with all the samples to understand what it is that's missing in the TREE_GRM_COMPENT table from the full tree dataset. Also this table is annualized so we won't need the interval as a predictor.

The TREE_GRM_COMPENT table does have **pre-calculated annual growth rates**

Key columns we need:

- **ANN_DIA_GROWTH** - Annual diameter growth
- **DIA_BEGIN** - Starting diameter
- **DIA_END** - Ending diameter
- **TRE_CN** - Tree identifier (to join back to TREE table

```
# Combine TREE_GRM_COMPONENT from all three states
combined_growth <- bind_rows(
  ct$TREE_GRM_COMPONENT %>% mutate(STATE = "CT"),
  ma$TREE_GRM_COMPONENT %>% mutate(STATE = "MA"),
  ri$TREE_GRM_COMPONENT %>% mutate(STATE = "RI")
)
nrow(combined_growth)
```

```
[1] 114569
```

```
# Look at the growth data
head(combined_growth %>%
     select(TRE_CN, PLT_CN, DIA_BEGIN, DIA_END, ANN_DIA_GROWTH, STATE)) %>%
  kable(caption = "First Few Rows of Combined Growth Data")
```

Table 12: First Few Rows of Combined Growth Data

TRE_CN	PLT_CN	DIA_BEGIN	DIA_END	ANN_DIA_GROWTH	STATE
3.522355e+14	1.68249e+14	10.3	10.9	0.09	CT
3.522355e+14	1.68249e+14	23.5	24.3	0.21	CT
3.522352e+14	1.68249e+14	14.0	14.4	0.17	CT
3.522355e+14	1.68249e+14	15.5	16.9	0.19	CT
3.522355e+14	1.68249e+14	5.9	6.0	0.07	CT
3.522355e+14	1.68249e+14	8.8	9.0	0.09	CT

```
# Summary statistics overall
summary(combined_growth$ANN_DIA_GROWTH)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.0600  0.0900  0.0931  0.1200  0.6500   24778
```

```r
# Summary by state
combined_growth %>%
  group_by(STATE) %>%
  summarise(
    n = n(),
    mean_growth = round(mean(ANN_DIA_GROWTH, na.rm = TRUE), 3),
    median_growth = round(median(ANN_DIA_GROWTH, na.rm = TRUE), 3),
    min_growth = round(min(ANN_DIA_GROWTH, na.rm = TRUE), 3),
    max_growth = round(max(ANN_DIA_GROWTH, na.rm = TRUE), 3),
    n_missing = sum(is.na(ANN_DIA_GROWTH))
  ) %>%
  kable(caption = "Annual Diameter Growth Summary by State")
```

Table 13: Annual Diameter Growth Summary by State

STATE	n	mean_growth	median_growth	min_growth	max_growth	n_missing
CT	33982	0.096	0.09	0	0.61	7278
MA	67183	0.091	0.08	0	0.65	14256
RI	13404	0.099	0.09	0	0.48	3244

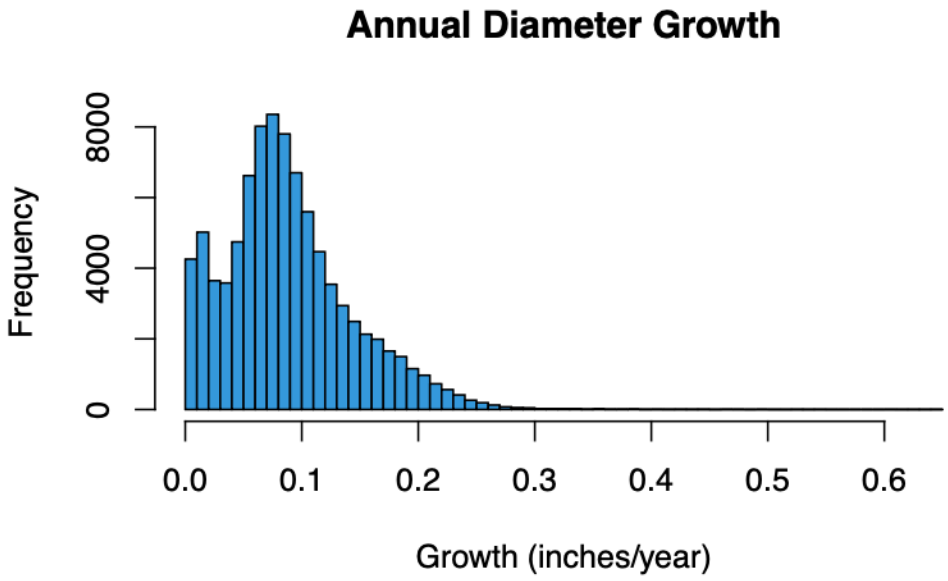
```r
# Remove any missing or weird values
growth_clean <- combined_growth %>%
  filter(!is.na(ANN_DIA_GROWTH),
    ANN_DIA_GROWTH >= 0)  # Positive growth only, negative doesn't make conceptual sense

# How many trees total and by state?
growth_clean %>%
  group_by(STATE) %>%
  summarise(n_trees = n()) %>%
  bind_rows(
    growth_clean %>% summarise(STATE = "TOTAL", n_trees = n())
  ) %>%
  kable(caption = "Number of Trees After Cleaning")
```

Table 14: Number of Trees After Cleaning

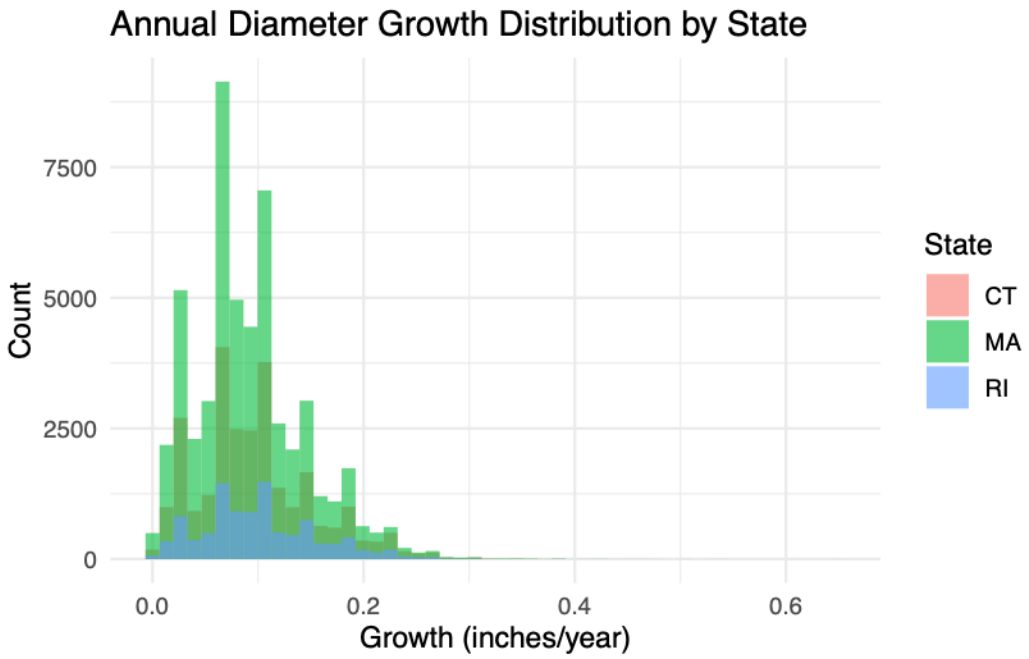
STATE	n_trees
CT	26704
MA	52927

STATE	n_trees
RI	10160
TOTAL	89791

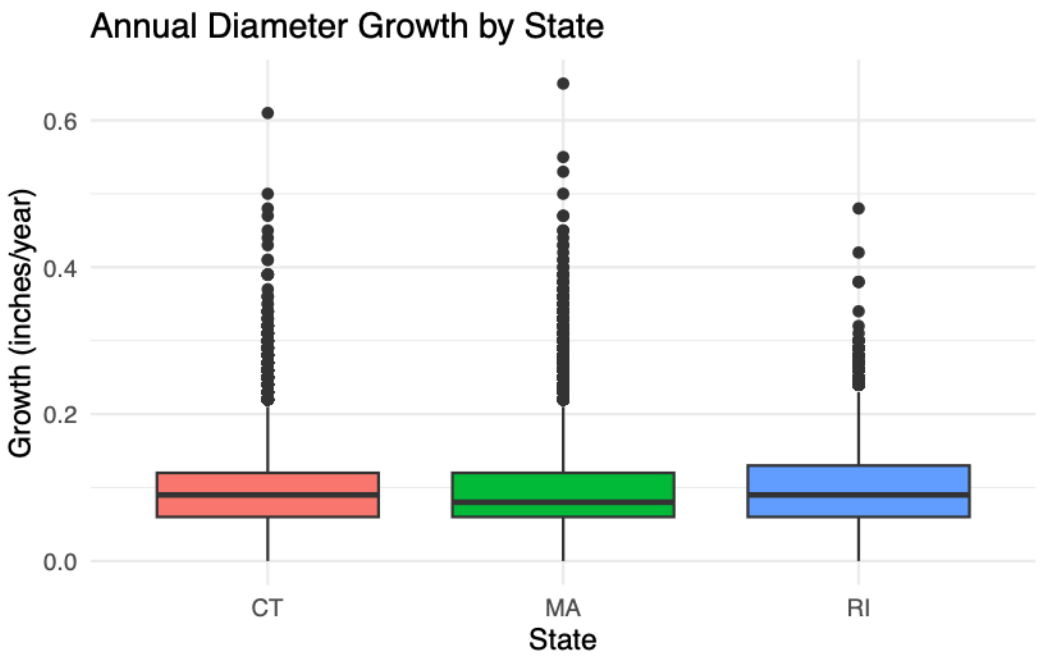
```
# Distribution - all states combined
hist(growth_clean$ANN_DIA_GROWTH, breaks = 50,
     main = "Annual Diameter Growth",
     xlab = "Growth (inches/year)",
     col = "#3498DB")
```




```
# Distribution by state (side-by-side)
ggplot(growth_clean, aes(x = ANN_DIA_GROWTH, fill = STATE)) +
  geom_histogram(bins = 50, alpha = 0.6, position = "identity") +
  labs(
    title = "Annual Diameter Growth Distribution by State",
    x = "Growth (inches/year)",
    y = "Count",
    fill = "State"
  ) +
  theme_minimal()
```

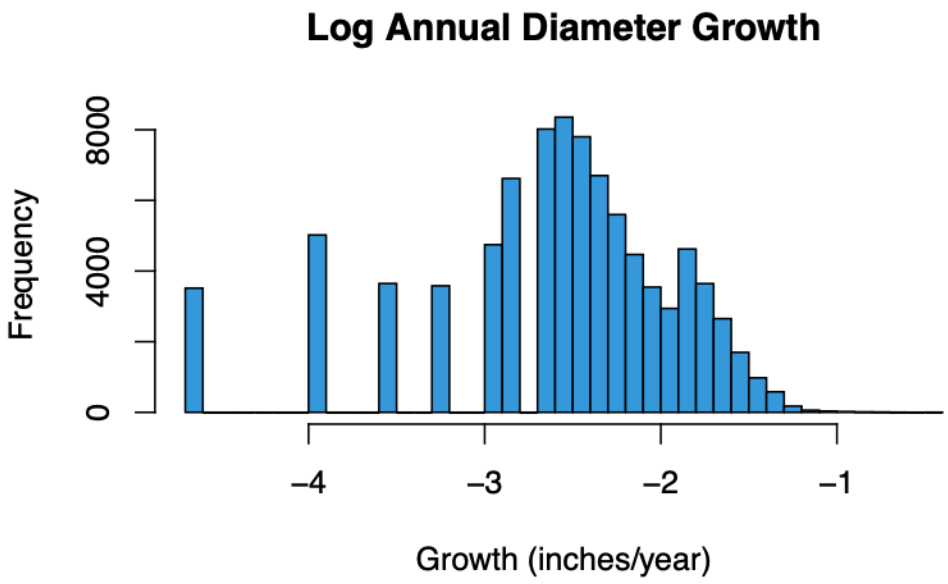



```
# Boxplot comparison
ggplot(growth_clean, aes(x = STATE, y = ANN_DIA_GROWTH, fill = STATE)) +
  geom_boxplot() +
  labs(
    title = "Annual Diameter Growth by State",
    x = "State",
    y = "Growth (inches/year)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



This is right skewed we should take the log of this:

```
# Add log-transformed growth
growth_clean <- combined_growth %>%
  mutate(log_growth = log(ANN_DIA_GROWTH))

# Distribution - all states combined
hist(growth_clean$log_growth, breaks = 50,
     main = "Log Annual Diameter Growth",
     xlab = "Growth (inches/year)",
     col = "#3498DB")
```



There are big gaps let's find out why:

```
# Check what's causing the gaps in log-transformed data
summary(combined_growth$ANN_DIA_GROWTH)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.0600  0.0900  0.0931  0.1200  0.6500   24778
```

```
# Are there zeros or negative values?
combined_growth %>%
  summarise(
    n_total = n(),
    n_zero = sum(ANN_DIA_GROWTH == 0, na.rm = TRUE),
    n_negative = sum(ANN_DIA_GROWTH < 0, na.rm = TRUE),
    n_positive = sum(ANN_DIA_GROWTH > 0, na.rm = TRUE),
    min_value = min(ANN_DIA_GROWTH, na.rm = TRUE),
    max_value = max(ANN_DIA_GROWTH, na.rm = TRUE)
  ) %>%
  kable()
```

n_total	n_zero	n_negative	n_positive	min_value	max_value
114569	741	0	89050	0	0.65

```
# Look at the distribution more carefully
combined_growth %>%
  filter(!is.na(ANN_DIA_GROWTH)) %>%
  count(ANN_DIA_GROWTH < 0) %>%
  kable(caption = "Are there negative growth values?")
```

Table 16: Are there negative growth values?

ANN_DIA_GROWTH < 0	n
FALSE	89791

```
# Check for very small values that might cause issues
combined_growth %>%
  filter(ANN_DIA_GROWTH > 0 & ANN_DIA_GROWTH < 0.01) %>%
  summarise(
    n_very_small = n(),
    examples = paste(head(ANN_DIA_GROWTH, 5), collapse = ", ")
  ) %>%
  kable()
```

n_very_small	examples
0	

Looking at this I realized I have zero growth values that will be lost with log.

Problem: 4,108 trees with no growth (ANN_DIA_GROWTH = 0)

- `log(0)` is **undefined** (negative infinity)
- R likely dropped these or created `-Inf` values

This doesn't account for there being gaps in different places though, but it's important to keep our 0 values as these are real cases of trees that didn't experience any annual growth.

Let's keep looking for the gaps

```r
# Look at the actual values in growth_clean
growth_clean %>%
  pull(ANN_DIA_GROWTH) %>%
  head(100)
```

```
  [1] 0.09 0.21 0.17 0.19 0.07 0.09 0.09 0.08 0.06 0.08 0.07 0.10 0.11 0.15 0.01
 [16] 0.19 0.06 0.04 0.10 0.08 0.21 0.09 0.07 0.09 0.08 0.08 0.19 0.09 0.07 0.10
 [31] 0.10 0.10 0.12 0.19 0.10 0.08   NA 0.07 0.02 0.01 0.05 0.08 0.20 0.22   NA
 [46] 0.12 0.09 0.15 0.12 0.10 0.03 0.10 0.14 0.09 0.08 0.15 0.14 0.04 0.17 0.15
 [61] 0.09 0.05 0.12 0.09 0.20 0.10 0.09 0.01 0.01 0.17 0.04 0.09 0.09 0.09   NA
 [76]   NA 0.06 0.15 0.16 0.18 0.09 0.11 0.20 0.24 0.02 0.05 0.13 0.10 0.09 0.15
 [91] 0.09 0.06 0.02 0.04 0.02 0.02 0.07 0.14 0.11 0.12
```

```r
# Are values discrete/rounded?
growth_clean %>%
  count(ANN_DIA_GROWTH) %>%
  arrange(desc(n)) %>%
  head(20) %>%
  kable(caption = "Most common growth values - Are they rounded?")
```

Table 18: Most common growth values - Are they rounded?

ANN_DIA_GROWTH	n
NA	24778
0.08	8354
0.07	8019
0.09	7801
0.10	6702
0.06	6620
0.11	5598
0.02	5020
0.05	4745
0.12	4467
0.03	3647
0.04	3582
0.13	3544
0.01	3516
0.14	2941
0.15	2492
0.16	2132

ANN_DIA_GROWTH	n
0.17	1989
0.18	1655
0.19	1497

```
# Check if growth values are multiples of 0.01
growth_clean %>%
  mutate(
    rounded_to_hundredth = round(ANN_DIA_GROWTH, 2) == ANN_DIA_GROWTH
  ) %>%
  count(rounded_to_hundredth) %>%
  kable()
```

rounded_to_hundredth	n
TRUE	89791
NA	24778

Gaps are due to transformation of discrete data.

Original data (rounded to 0.01):

- You have: 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08...
- These are discrete, evenly spaced values

After log transformation:

- You have: log(0.01) = -4.605, log(0.02) = -3.912, log(0.03) = -3.507, log(0.04) = -3.219, log(0.05) = -2.996, log(0.06) = -2.813, log(0.07) = -2.659, log(0.08) = -2.526

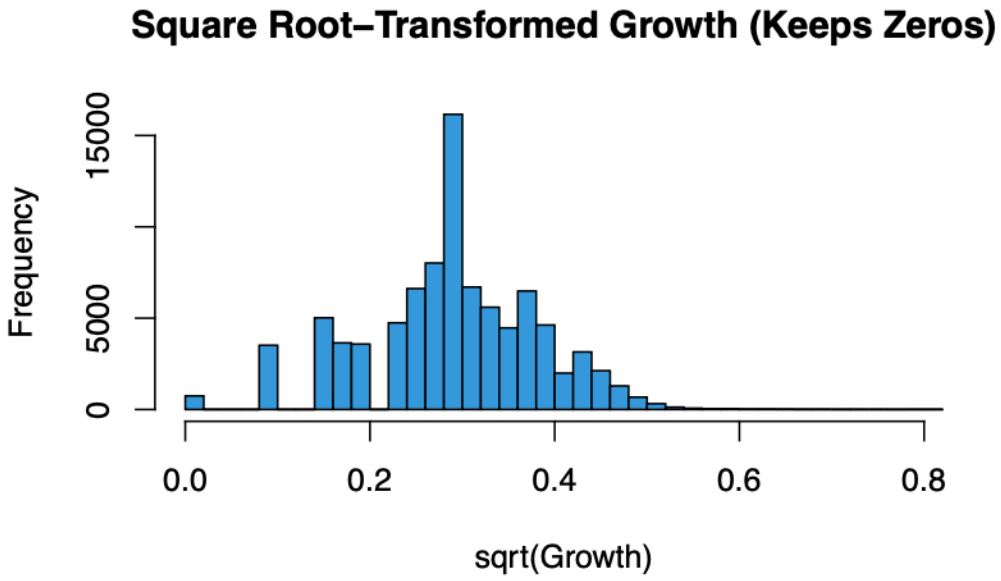
gaps between log values are not equal anymore:

- Gap between log(0.01) and log(0.02) = 0.693
- Gap between log(0.07) and log(0.08) = 0.133

The gaps will be less prominent with square root and we want to keep our zeros.

Let's try square root transformation instead since we want to keep our zero growth values:

```r
growth_clean_sqrt <- growth_clean %>%
  mutate(sqrt_growth = sqrt(ANN_DIA_GROWTH))

hist(growth_clean_sqrt$sqrt_growth, breaks = 50,
     main = "Square Root-Transformed Growth (Keeps Zeros)",
     xlab = "sqrt(Growth)", col = "#3498DB")
```




```r
# Create growth_with_ids by joining growth_clean with combined_tree
growth_with_ids <- growth_clean %>%
  left_join(
    combined_tree %>% select(CN, PLOT, SUBP, TREE, STATECD, INVYR),
    by = c("TRE_CN" = "CN")
  )

# Use the STATE column that's already in growth_clean
tree_measurements <- growth_with_ids %>%
  mutate(TREE_ID = paste(STATE, PLOT, SUBP, TREE, sep = "_")) %>%
  group_by(TREE_ID) %>%
  summarise(
    n_measurements = n(),
    years = paste(sort(unique(INVYR)), collapse = ", ")
  )
```

```r
# Summary table
tree_measurements %>%
  count(n_measurements) %>%
  mutate(pct = round(100 * n / sum(n), 1)) %>%
  kable(caption = "Number of Measurements Per Tree",
        col.names = c("Measurements", "Number of Trees", "Percent"))
```

Table 20: Number of Measurements Per Tree

Measurements	Number of Trees	Percent
1	15876	32.3
2	11110	22.6
3	13489	27.4
4	8226	16.7
5	233	0.5
6	140	0.3
7	78	0.2
8	35	0.1
9	15	0.0
10	8	0.0
11	4	0.0
12	1	0.0

Review Forest Type

Model Dataset Creation

```r
# Create model dataset (starting from growth_clean which has filtering done)
model_data <- growth_clean %>%
  # Join with combined_tree to get PLOT, SUBP, TREE identifiers AND SPCD
  left_join(
    combined_tree %>% select(CN, PLOT, SUBP, TREE, CONDID, INVYR, SPCD),  # Added SPCD here
    by = c("TRE_CN" = "CN")
  ) %>%
  # Create IDs
  mutate(
    TREE_ID = paste(STATECD, PLOT, SUBP, TREE, sep = "_"),
    PLOT_ID = paste(STATECD, PLOT, sep = "_"),
```

```
    sqrt_growth = sqrt(ANN_DIA_GROWTH)  # Add sqrt transformation
  ) %>%
  # Join with COND for forest type
  left_join(
    bind_rows(
      ct$COND %>% mutate(STATE_COND = "CT"),
      ma$COND %>% mutate(STATE_COND = "MA"),
      ri$COND %>% mutate(STATE_COND = "RI")
    ) %>% select(PLT_CN, CONDID, FORTYPCD, STDAGE),
    by = c("PLT_CN", "CONDID")
  ) %>%
  # Join with PLOTGEOM for ecological subsection
  left_join(
    bind_rows(
      ct$PLOTGEOM %>% mutate(STATE_PLOT = "CT"),
      ma$PLOTGEOM %>% mutate(STATE_PLOT = "MA"),
      ri$PLOTGEOM %>% mutate(STATE_PLOT = "RI")
    ) %>% select(CN, ECOSUBCD, LAT, LON),
    by = c("PLT_CN" = "CN")
  )

# Check hierarchical structure
model_data %>%
  summarise(
    Level_5_Ecosubcd = n_distinct(ECOSUBCD),
    Level_4_States = n_distinct(STATE),
    Level_3_ForTypes = n_distinct(FORTYPCD),
    Level_2_Plots = n_distinct(PLOT_ID),
    Level_1_Trees = n_distinct(TREE_ID),
    Total_Observations = n()
  ) %>%
  kable(caption = "5-Level Hierarchical Structure")
```

Table 21: 5-Level Hierarchical Structure

Level_5_Ecosubcd	Level_4_States	Level_3_ForTypes	Level_2_Plots	Level_1_Trees	Total_Observations
15	3	53	1278	49215	114569

```
# Distribution of measurements per tree
model_data %>%
  group_by(TREE_ID) %>%
```

```
  summarise(n_measurements = n()) %>%
  count(n_measurements) %>%
  mutate(pct = round(100 * n / sum(n), 1)) %>%
  kable(caption = "Measurements per Tree (Repeated Measures)")
```

Table 22: Measurements per Tree (Repeated Measures)

n_measurements	n	pct
1	15876	32.3
2	11110	22.6
3	13489	27.4
4	8226	16.7
5	233	0.5
6	140	0.3
7	78	0.2
8	35	0.1
9	15	0.0
10	8	0.0
11	4	0.0
12	1	0.0

```
# Save the dataset
saveRDS(model_data, "FIA_data/model_data_multilevel.rds")
```

Review TREE_ID as Unique Identifier

```
# Create example_trees by joining growth_clean with combined_tree
example_trees <- growth_clean %>%
  left_join(
    combined_tree %>% select(CN, PLOT, SUBP, TREE, STATE, STATE, INVYR),
    by = c("TRE_CN" = "CN")
  ) %>%
  mutate(
    TREE_ID = paste(STATECD, PLOT, SUBP, TREE, sep = "_"),
    PLOT_ID = paste(STATECD, PLOT, sep = "_")
  ) %>%
  group_by(TREE_ID) %>%
  filter(n() >= 3) %>%  # Trees with 3+ measurements
  arrange(TREE_ID, INVYR) %>%
```

```r
  select(TREE_ID, PLOT_ID, STATECD, INVYR, DIA_BEGIN, DIA_END, ANN_DIA_GROWTH) %>%
  ungroup()

# Pick one specific tree and show its full history
one_tree <- example_trees %>%
  filter(TREE_ID == first(TREE_ID)) %>%
  arrange(INVYR)

one_tree %>%
  kable(caption = "One Tree's Complete Measurement History")
```

Table 23: One Tree's Complete Measurement History

TREE_ID	PLOT_ID	STATECD	INVYR	DIA_BEGIN	DIA_END	ANN_DIA_GROWTH
25_1005_1_10	25_1005	25	2006	14.50	15.8	0.10
25_1005_1_10	25_1005	25	2010	15.80	16.7	0.11
25_1005_1_10	25_1005	25	2016	16.70	17.4	0.09
25_1005_1_10	25_1005	25	2023	16.89	17.6	0.10

```r
# Verify the tree is growing (DIA should increase over time)
model_data %>%
  group_by(TREE_ID) %>%
  filter(n() >= 2) %>%
  arrange(TREE_ID, INVYR) %>%
  mutate(
    diameter_increases = DIA_END > lag(DIA_END)
  ) %>%
  filter(!is.na(diameter_increases)) %>%
  summarise(
    pct_increasing = round(100 * sum(diameter_increases) / n(), 1)
  ) %>%
  ungroup() %>%
  summarise(
    mean_pct_increasing = mean(pct_increasing)
  ) %>%
  kable()
```

mean_pct_increasing
81.57261

Modeling

Fixed Effects:

- `STDAGE` = Stand age
- `SPCD` = Species

Random Effects (5 nested levels):

- Level 5: Ecological subsections (15) - broad climate/geography regions
- Level 4: States (3) - CT, MA, RI
- Level 3: Forest types (45) - nested within states
- Level 2: Plots (1,080) - permanent monitoring locations
- Level 1: Trees (36,574) - individual trees measured repeatedly over time

Random Intercept:

- Each group (tree, plot, forest type, state, subsection) gets its own baseline growth rate
- But the effect of predictors (FORTYPCD, STDAGE, SPCD) is assumed to be the same across all groups

Model Question: Do growth rates vary by forest type, stand age, and species while accounting for clustering at multiple geographic and ecological levels?

Model 1: 5 Levels

```
# Build the 5-level model
model <- lmer(
  sqrt_growth ~ STDAGE + factor(SPCD) +
    (1 | ECOSUBCD / STATE / FORTYPCD / PLOT_ID / TREE_ID),
  data = model_data
)

# Model summary
summary(model)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
sqrt_growth ~ STDAGE + factor(SPCD) + (1 | ECOSUBCD/STATE/FORTYPCD/PLOT_ID/TREE_ID)
   Data: model_data

REML criterion at convergence: -246618.9
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-7.8739 -0.2994  0.0549  0.3741  6.7133

Random effects:
 Groups                                    Name        Variance  Std.Dev.
 TREE_ID:PLOT_ID:FORTYPCD:STATE:ECOSUBCD (Intercept) 4.469e-03 0.066852
 PLOT_ID:FORTYPCD:STATE:ECOSUBCD         (Intercept) 1.165e-03 0.034135
 FORTYPCD:STATE:ECOSUBCD                 (Intercept) 1.605e-04 0.012669
 STATE:ECOSUBCD                          (Intercept) 6.063e-05 0.007787
 ECOSUBCD                                (Intercept) 8.918e-05 0.009444
 Residual                                            1.350e-03 0.036748
Number of obs: 88645, groups:
TREE_ID:PLOT_ID:FORTYPCD:STATE:ECOSUBCD, 42905; PLOT_ID:FORTYPCD:STATE:ECOSUBCD, 1531; FORTYP

Fixed effects:
                   Estimate Std. Error t value
(Intercept)       3.860e-01  4.508e-02   8.563
STDAGE           -8.248e-04  2.107e-05 -39.138
factor(SPCD)12   -1.514e-02  4.410e-02  -0.343
factor(SPCD)43   -1.056e-01  4.772e-02  -2.213
factor(SPCD)57    4.494e-03  9.539e-02   0.047
factor(SPCD)68   -1.266e-01  4.525e-02  -2.798
factor(SPCD)71   -1.094e-01  5.094e-02  -2.147
factor(SPCD)91   -1.353e-01  4.599e-02  -2.941
factor(SPCD)94    1.616e-02  8.960e-02   0.180
factor(SPCD)96   -8.958e-02  8.612e-02  -1.040
factor(SPCD)97   -1.027e-01  4.516e-02  -2.275
factor(SPCD)125   3.749e-02  4.698e-02   0.798
factor(SPCD)126  -8.029e-02  4.510e-02  -1.780
factor(SPCD)129   2.602e-02  4.495e-02   0.579
factor(SPCD)130  -1.431e-01  5.195e-02  -2.754
factor(SPCD)261  -2.413e-03  4.495e-02  -0.054
factor(SPCD)313  -8.910e-02  5.673e-02  -1.570
factor(SPCD)315  -2.056e-01  4.540e-02  -4.530
factor(SPCD)316  -5.860e-02  4.494e-02  -1.304
factor(SPCD)317  -4.537e-02  4.708e-02  -0.964
factor(SPCD)318  -7.342e-02  4.498e-02  -1.632
factor(SPCD)319  -2.543e-01  6.780e-02  -3.750
factor(SPCD)320  -8.880e-02  4.592e-02  -1.934
factor(SPCD)341  -1.125e-01  4.842e-02  -2.323
factor(SPCD)345  -5.794e-02  8.918e-02  -0.650
```

```
factor(SPCD)356  -1.908e-01  4.572e-02  -4.173
factor(SPCD)371  -9.661e-02  4.498e-02  -2.148
factor(SPCD)372  -8.768e-02  4.496e-02  -1.950
factor(SPCD)373   5.170e-02  8.958e-02   0.577
factor(SPCD)375  -9.819e-02  4.505e-02  -2.179
factor(SPCD)379  -1.257e-01  4.519e-02  -2.783
factor(SPCD)391  -2.070e-01  4.526e-02  -4.573
factor(SPCD)402   6.985e-03  4.577e-02   0.153
factor(SPCD)403   1.285e-02  4.505e-02   0.285
factor(SPCD)407   1.474e-02  4.519e-02   0.326
factor(SPCD)409   1.944e-02  4.549e-02   0.427
factor(SPCD)421  -2.146e-01  4.588e-02  -4.677
factor(SPCD)452  -6.819e-02  7.000e-02  -0.974
factor(SPCD)462  -1.189e-01  7.094e-02  -1.676
factor(SPCD)491  -1.883e-01  4.745e-02  -3.968
factor(SPCD)500  -1.973e-01  5.879e-02  -3.356
factor(SPCD)531  -8.583e-02  4.497e-02  -1.909
factor(SPCD)541  -3.919e-02  4.499e-02  -0.871
factor(SPCD)543  -7.262e-02  4.686e-02  -1.550
factor(SPCD)544  -7.117e-02  4.647e-02  -1.532
factor(SPCD)552  -6.448e-02  5.004e-02  -1.289
factor(SPCD)591  -1.185e-01  4.889e-02  -2.424
factor(SPCD)601  -6.364e-02  5.978e-02  -1.065
factor(SPCD)602  -7.249e-02  4.997e-02  -1.451
factor(SPCD)621   5.153e-02  4.570e-02   1.128
factor(SPCD)660  -1.364e-01  4.594e-02  -2.968
factor(SPCD)682  -1.165e-01  9.014e-02  -1.293
factor(SPCD)693  -1.351e-02  4.513e-02  -0.299
factor(SPCD)701  -1.505e-01  4.528e-02  -3.325
factor(SPCD)731  -6.332e-02  5.118e-02  -1.237
factor(SPCD)741  -1.423e-01  6.385e-02  -2.229
factor(SPCD)742   1.235e-02  4.752e-02   0.260
factor(SPCD)743   5.392e-03  4.519e-02   0.119
factor(SPCD)746  -3.755e-02  4.520e-02  -0.831
factor(SPCD)760  -1.815e-01  6.899e-02  -2.631
factor(SPCD)761  -1.490e-01  4.622e-02  -3.223
factor(SPCD)762  -1.119e-01  4.500e-02  -2.487
factor(SPCD)763  -2.773e-01  5.729e-02  -4.840
factor(SPCD)771  -9.297e-03  8.803e-02  -0.106
factor(SPCD)800  -2.232e-01  6.369e-02  -3.505
factor(SPCD)802  -5.234e-02  4.498e-02  -1.164
factor(SPCD)804  -5.659e-02  4.606e-02  -1.229
factor(SPCD)806   5.196e-02  4.498e-02   1.155
```

```
factor(SPCD)809   1.322e-02  5.499e-02   0.240
factor(SPCD)816  -2.018e-01  4.652e-02  -4.337
factor(SPCD)823  -3.733e-02  6.127e-02  -0.609
factor(SPCD)825   1.644e-02  8.327e-02   0.197
factor(SPCD)830   5.980e-02  6.039e-02   0.990
factor(SPCD)832  -2.186e-02  4.517e-02  -0.484
factor(SPCD)833   5.709e-02  4.495e-02   1.270
factor(SPCD)835  -3.723e-02  4.859e-02  -0.766
factor(SPCD)837   5.035e-02  4.497e-02   1.120
factor(SPCD)901  -8.407e-02  4.609e-02  -1.824
factor(SPCD)920  -8.849e-02  4.718e-02  -1.875
factor(SPCD)922  -1.681e-01  4.706e-02  -3.572
factor(SPCD)923  -3.131e-01  5.833e-02  -5.367
factor(SPCD)927  -1.781e-01  4.873e-02  -3.655
factor(SPCD)931  -1.259e-01  4.527e-02  -2.781
factor(SPCD)935  -2.085e-01  5.116e-02  -4.074
factor(SPCD)951  -6.189e-02  4.641e-02  -1.333
factor(SPCD)972  -1.257e-01  4.519e-02  -2.782
factor(SPCD)975  -1.362e-01  4.815e-02  -2.828
factor(SPCD)977  -6.379e-02  9.084e-02  -0.702
factor(SPCD)997  -1.851e-01  5.191e-02  -3.565
factor(SPCD)998  -1.996e-01  8.913e-02  -2.240
factor(SPCD)999  -1.189e-01  5.367e-02  -2.216
factor(SPCD)6918 -1.927e-01  4.809e-02  -4.008
factor(SPCD)6955 -1.645e-01  6.240e-02  -2.637
factor(SPCD)8813 -1.143e-02  8.920e-02  -0.128


Correlation matrix not shown by default, as p = 94 > 12.
Use print(x, correlation=TRUE)  or
    vcov(x)        if you need it
```

Model 1 Results:

Fixed Effects:

STDAGE (Stand Age):

- Coefficient: -0.000825 ($p < 0.001$)
- **Interpretation:** For each additional year of stand age, growth rate decreases by 0.000825 on the sqrt scale
- **Makes biological sense:** Older stands have slower-growing trees

SPCD (Species):

- Baseline species SPCD = 10
- **Significantly slower growers than baseline:**
  - Species 923: -0.313 ($p < 0.001$) - slowest
  - Species 763: -0.277 ($p < 0.001$)
  - Species 319: -0.254 ($p < 0.001$)
  - Species 421: -0.215 ($p < 0.001$)
- **Not significantly different from baseline:**
  - Species 12, 57, 94, 125, 621, 833, 837 ($p > 0.05$)

Random Effects (Variance Components):

Level	Variance	Interpretation
Tree (Level 1)	0.00447	Largest - individual tree differences
Plot (Level 2)	0.00117	Plot-level variation
Forest Type (Level 3)	0.00016	Forest type differences
State (Level 4)	0.00006	Small state-level differences
Ecosubcd (Level 5)	0.00009	Ecological subsection variation
Residual	0.00135	Unexplained variation

```r
# Calculate ICC for all levels
icc_results <- icc(model, by_group = TRUE)
print(icc_results)
```

```
# ICC by Group

Group                                    |   ICC
-----------------------------------------------
TREE_ID:PLOT_ID:FORTYPCD:STATE:ECOSUBCD | 0.613
PLOT_ID:FORTYPCD:STATE:ECOSUBCD         | 0.160
FORTYPCD:STATE:ECOSUBCD                 | 0.022
STATE:ECOSUBCD                          | 0.008
ECOSUBCD                                | 0.012
```

```r
# Manual calculation to understand it better
variance_components <- as.data.frame(VarCorr(model))
```

```r
variance_components %>%
  mutate(
    ICC = vcov / sum(vcov),
    Percent = round(100 * ICC, 2)
  ) %>%
  select(grp, vcov, ICC, Percent) %>%
  kable(caption = "Variance Partitioning (ICC) by Level",
        col.names = c("Level", "Variance", "ICC", "% of Total Variance"))
```

Table 26: Variance Partitioning (ICC) by Level

Level	Variance	ICC	% of Total Variance
TREE_ID:PLOT_ID:FORTYPCD:STATE:ECOSUBCD	0.0044801	0.6126216	61.26
PLOT_ID:FORTYPCD:STATE:ECOSUBCD	0.0011652	0.1597279	15.97
FORTYPCD:STATE:ECOSUBCD	0.0001605	0.0220005	2.20
STATE:ECOSUBCD	0.0000606	0.0083111	0.83
ECOSUBCD	0.0000892	0.0122252	1.22
Residual	0.0013504	0.1851138	18.51

```r
# Total variance
total_var <- sum(variance_components$vcov)

# Variance at each level
cat("\nVariance Decomposition:\n")
```

```
Variance Decomposition:
```

```r
cat("Tree level:", round(100 * variance_components$vcov[1] / total_var, 2), "%\n")
```

```
Tree level: 61.26 %
```

```r
cat("Plot level:", round(100 * variance_components$vcov[2] / total_var, 2), "%\n")
```

```
Plot level: 15.97 %
```

```r
cat("Forest Type:", round(100 * variance_components$vcov[3] / total_var, 2), "%\n")
```

```
Forest Type: 2.2 %
```

```r
cat("State level:", round(100 * variance_components$vcov[4] / total_var, 2), "%\n")
```

```
State level: 0.83 %
```

```r
cat("Ecosubcd level:", round(100 * variance_components$vcov[5] / total_var, 2), "%\n")
```

```
Ecosubcd level: 1.22 %
```

```r
cat("Residual:", round(100 * variance_components$vcov[6] / total_var, 2), "%\n")
```

```
Residual: 18.51 %
```

ICC Findings:

Tree Level (61.26%) - most important

- **61% of variation in growth is between individual trees**
- This confirms you NEED the multilevel structure with tree-level random effects
- Individual tree characteristics (genetics, microsite conditions, competition) matter most

Plot Level (15.97%) - important

- **16% of variation is between plots**
- Plot-level conditions (soil, topography, disturbance history) are meaningful
- Justifies including plots as a grouping level

Forest Type (2.20%) - small

- Forest type adds modest variation
- Could keep or remove this level

State (0.83%) & Ecosubcd (1.22%) - very small

- Combined ~2% of variation
- These highest levels add little beyond plot and tree
- **Could consider simplifying** to a 3-level model (Tree → Plot → Forest Type)

Residual (18.51%) - Unexplained variation

- Measurement error + unmeasured factors

Let's try removing state and ecosubcd.

Model 2: 3 Levels

```
# Simpler 3-level model
model_simple <- lmer(
  sqrt_growth ~ STDAGE + factor(SPCD) +
    (1 | FORTYPCD / PLOT_ID / TREE_ID),
  data = model_data
)

summary(model_simple)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: sqrt_growth ~ STDAGE + factor(SPCD) + (1 | FORTYPCD/PLOT_ID/TREE_ID)
   Data: model_data

REML criterion at convergence: -246611.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-7.8730 -0.2992  0.0544  0.3740  6.7189

Random effects:
 Groups                   Name        Variance  Std.Dev.
 TREE_ID:PLOT_ID:FORTYPCD (Intercept) 0.0044690 0.06685
 PLOT_ID:FORTYPCD         (Intercept) 0.0012537 0.03541
 FORTYPCD                 (Intercept) 0.0003091 0.01758
 Residual                             0.0013504 0.03675
Number of obs: 88645, groups:
```

```
TREE_ID:PLOT_ID:FORTYPCD, 42905; PLOT_ID:FORTYPCD, 1531; FORTYPCD, 47

Fixed effects:
                   Estimate Std. Error t value
(Intercept)       3.901e-01  4.506e-02   8.658
STDAGE           -8.257e-04  2.133e-05 -38.709
factor(SPCD)12   -1.580e-02  4.410e-02  -0.358
factor(SPCD)43   -1.077e-01  4.772e-02  -2.258
factor(SPCD)57   -2.098e-02  9.538e-02  -0.220
factor(SPCD)68   -1.273e-01  4.525e-02  -2.814
factor(SPCD)71   -1.099e-01  5.094e-02  -2.158
factor(SPCD)91   -1.341e-01  4.601e-02  -2.915
factor(SPCD)94    1.681e-02  8.960e-02   0.188
factor(SPCD)96   -8.512e-02  8.613e-02  -0.988
factor(SPCD)97   -1.034e-01  4.516e-02  -2.289
factor(SPCD)125   3.844e-02  4.700e-02   0.818
factor(SPCD)126  -7.984e-02  4.510e-02  -1.770
factor(SPCD)129   2.581e-02  4.495e-02   0.574
factor(SPCD)130  -1.424e-01  5.196e-02  -2.740
factor(SPCD)261  -3.097e-03  4.495e-02  -0.069
factor(SPCD)313  -7.845e-02  5.673e-02  -1.383
factor(SPCD)315  -2.061e-01  4.540e-02  -4.539
factor(SPCD)316  -5.941e-02  4.494e-02  -1.322
factor(SPCD)317  -5.010e-02  4.716e-02  -1.062
factor(SPCD)318  -7.392e-02  4.498e-02  -1.643
factor(SPCD)319  -2.549e-01  6.780e-02  -3.759
factor(SPCD)320  -8.948e-02  4.592e-02  -1.949
factor(SPCD)341  -9.786e-02  4.842e-02  -2.021
factor(SPCD)345  -5.782e-02  8.918e-02  -0.648
factor(SPCD)356  -1.911e-01  4.572e-02  -4.180
factor(SPCD)371  -9.736e-02  4.498e-02  -2.165
factor(SPCD)372  -8.815e-02  4.495e-02  -1.961
factor(SPCD)373   5.312e-02  8.958e-02   0.593
factor(SPCD)375  -9.901e-02  4.505e-02  -2.198
factor(SPCD)379  -1.265e-01  4.519e-02  -2.798
factor(SPCD)391  -2.076e-01  4.526e-02  -4.587
factor(SPCD)402   6.906e-03  4.577e-02   0.151
factor(SPCD)403   1.229e-02  4.505e-02   0.273
factor(SPCD)407   1.410e-02  4.519e-02   0.312
factor(SPCD)409   1.967e-02  4.549e-02   0.432
factor(SPCD)421  -2.157e-01  4.588e-02  -4.702
factor(SPCD)452  -6.157e-02  7.002e-02  -0.879
factor(SPCD)462  -1.166e-01  7.087e-02  -1.645
```

```
factor(SPCD)491  -1.882e-01  4.745e-02  -3.967
factor(SPCD)500  -1.982e-01  5.878e-02  -3.371
factor(SPCD)531  -8.641e-02  4.497e-02  -1.921
factor(SPCD)541  -4.025e-02  4.499e-02  -0.895
factor(SPCD)543  -7.379e-02  4.686e-02  -1.575
factor(SPCD)544  -7.244e-02  4.647e-02  -1.559
factor(SPCD)552  -6.637e-02  5.004e-02  -1.327
factor(SPCD)591  -1.196e-01  4.889e-02  -2.446
factor(SPCD)601  -6.823e-02  5.979e-02  -1.141
factor(SPCD)602  -7.584e-02  5.001e-02  -1.516
factor(SPCD)621   5.183e-02  4.570e-02   1.134
factor(SPCD)660  -1.381e-01  4.594e-02  -3.006
factor(SPCD)682  -1.238e-01  9.017e-02  -1.373
factor(SPCD)693  -1.458e-02  4.513e-02  -0.323
factor(SPCD)701  -1.515e-01  4.528e-02  -3.346
factor(SPCD)731  -7.034e-02  5.130e-02  -1.371
factor(SPCD)741  -1.418e-01  6.386e-02  -2.221
factor(SPCD)742   1.091e-02  4.754e-02   0.229
factor(SPCD)743   4.671e-03  4.519e-02   0.103
factor(SPCD)746  -3.830e-02  4.520e-02  -0.847
factor(SPCD)760  -1.803e-01  6.897e-02  -2.614
factor(SPCD)761  -1.501e-01  4.622e-02  -3.248
factor(SPCD)762  -1.126e-01  4.500e-02  -2.502
factor(SPCD)763  -2.782e-01  5.729e-02  -4.855
factor(SPCD)771  -1.745e-02  8.807e-02  -0.198
factor(SPCD)800  -2.289e-01  6.358e-02  -3.600
factor(SPCD)802  -5.279e-02  4.498e-02  -1.174
factor(SPCD)804  -5.739e-02  4.606e-02  -1.246
factor(SPCD)806   5.167e-02  4.498e-02   1.149
factor(SPCD)809   1.298e-02  5.499e-02   0.236
factor(SPCD)816  -2.041e-01  4.653e-02  -4.386
factor(SPCD)823  -3.857e-02  6.127e-02  -0.629
factor(SPCD)825   1.535e-02  8.327e-02   0.184
factor(SPCD)830   5.966e-02  6.039e-02   0.988
factor(SPCD)832  -2.236e-02  4.517e-02  -0.495
factor(SPCD)833   5.642e-02  4.495e-02   1.255
factor(SPCD)835  -4.136e-02  4.864e-02  -0.850
factor(SPCD)837   4.983e-02  4.497e-02   1.108
factor(SPCD)901  -8.648e-02  4.609e-02  -1.876
factor(SPCD)920  -8.808e-02  4.718e-02  -1.867
factor(SPCD)922  -1.677e-01  4.707e-02  -3.562
factor(SPCD)923  -3.196e-01  5.847e-02  -5.466
factor(SPCD)927  -1.846e-01  4.875e-02  -3.787
```

```
factor(SPCD)931  -1.266e-01  4.527e-02  -2.798
factor(SPCD)935  -2.093e-01  5.117e-02  -4.091
factor(SPCD)951  -6.203e-02  4.641e-02  -1.336
factor(SPCD)972  -1.272e-01  4.519e-02  -2.816
factor(SPCD)975  -1.383e-01  4.816e-02  -2.872
factor(SPCD)977  -6.074e-02  9.085e-02  -0.669
factor(SPCD)997  -1.883e-01  5.199e-02  -3.622
factor(SPCD)998  -1.986e-01  8.913e-02  -2.228
factor(SPCD)999  -1.195e-01  5.366e-02  -2.227
factor(SPCD)6918 -1.952e-01  4.809e-02  -4.059
factor(SPCD)6955 -1.624e-01  6.238e-02  -2.604
factor(SPCD)8813 -1.333e-02  8.920e-02  -0.149


Correlation matrix not shown by default, as p = 94 > 12.
Use print(x, correlation=TRUE)  or
    vcov(x)        if you need it
```

```
icc(model_simple, by_group = TRUE)
```

```
# ICC by Group

Group                     |   ICC
--------------------------------
TREE_ID:PLOT_ID:FORTYPCD | 0.605
PLOT_ID:FORTYPCD         | 0.170
FORTYPCD                 | 0.042
```

Comparison: 5-Level vs 3-Level Model

Model	REML	Tree ICC	Plot ICC	Upper Levels ICC
5-level	-246618.9	61.3%	16.0%	4.1% (FT+State+Eco)
3-level	-246611.8	60.5%	17.0%	4.2% (FT only)

Model 1 vs Model 2:

1. **Nearly identical fit** (REML values very close)
2. **Same conclusions** - Tree (61%) and Plot (17%) variance dominate
3. **Simpler is better** - State and Ecosubcd levels added complexity without improving fit
4. **Forest Type still matters** (4.2% variance) - worth keeping

Fixed Effects Evaluation:

STDAGE (Stand Age):

- Coefficient: -0.000826
- **p < 0.001*** (highly significant)
- Keep - evidence older stands have slower growth

SPCD (Species):

Looking at the 93 species coefficients:

Highly significant (p < 0.001):

- Species 315, 319, 391, 421, 701, 763, 816, 922, 923, 927, etc.
- Clear differences from baseline

Marginally significant (p < 0.05):

- Many more species (43, 68, 71, 91, 97, 130, etc.)

Not significant (p > 0.05):

- About 40-50 species show no difference from baseline

Should We Keep Species?

Yes, because:

Many species are significantly different

Species is biologically important for growth

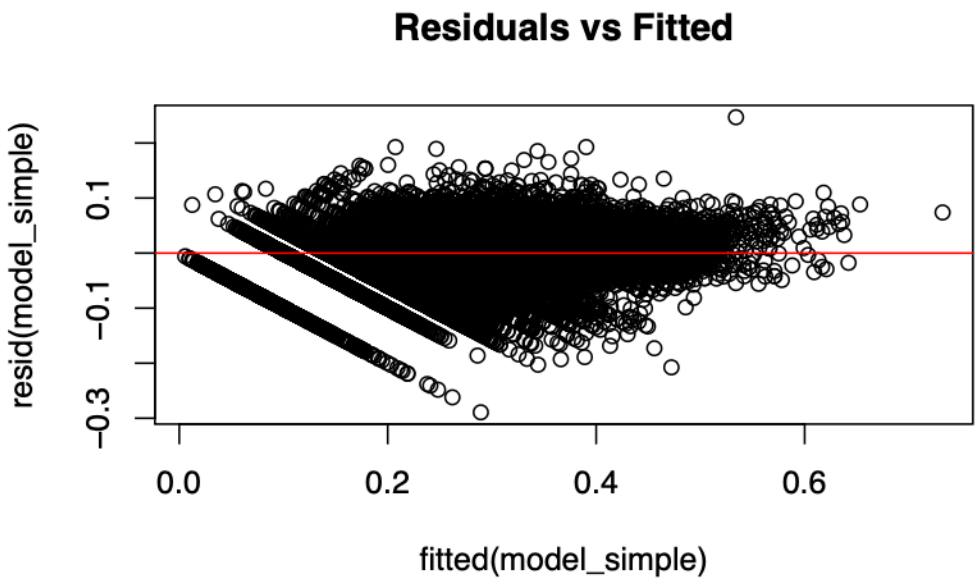
Forest Type?

```
# Full model with forest type random effect
model_full <- lmer(sqrt_growth ~ STDAGE + factor(SPCD) + (1 | FORTYPCD/PLOT_ID/TREE_ID), data

# Reduced model without forest type random effect
model_reduced <- lmer(sqrt_growth ~ STDAGE + factor(SPCD) + (1 | PLOT_ID/TREE_ID), data = mo

# Perform likelihood ratio test
anova(model_reduced, model_full)
```

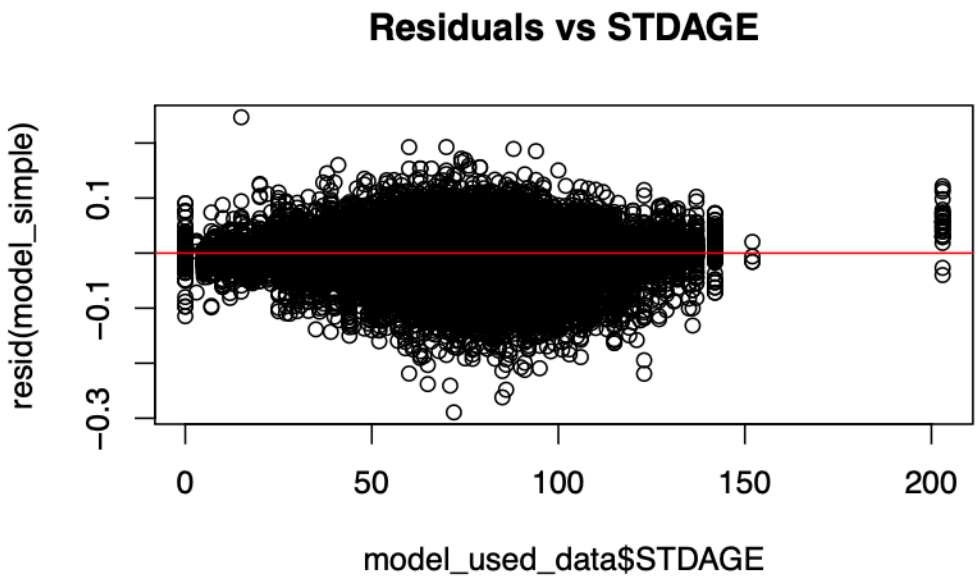
```
Data: model_data
Models:
model_reduced: sqrt_growth ~ STDAGE + factor(SPCD) + (1 | PLOT_ID/TREE_ID)
model_full: sqrt_growth ~ STDAGE + factor(SPCD) + (1 | FORTYPCD/PLOT_ID/TREE_ID)
              npar     AIC     BIC logLik -2*log(L) Chisq Df Pr(>Chisq)
model_reduced   97 -251538 -250627 125866   -251732
model_full      98 -247069 -246148 123632   -247265     0  1          1
```

Not important to the model but does change the model much. Let's keep it as it adds to biological interpretation. We can look for a better ecological grouping variable in the data.

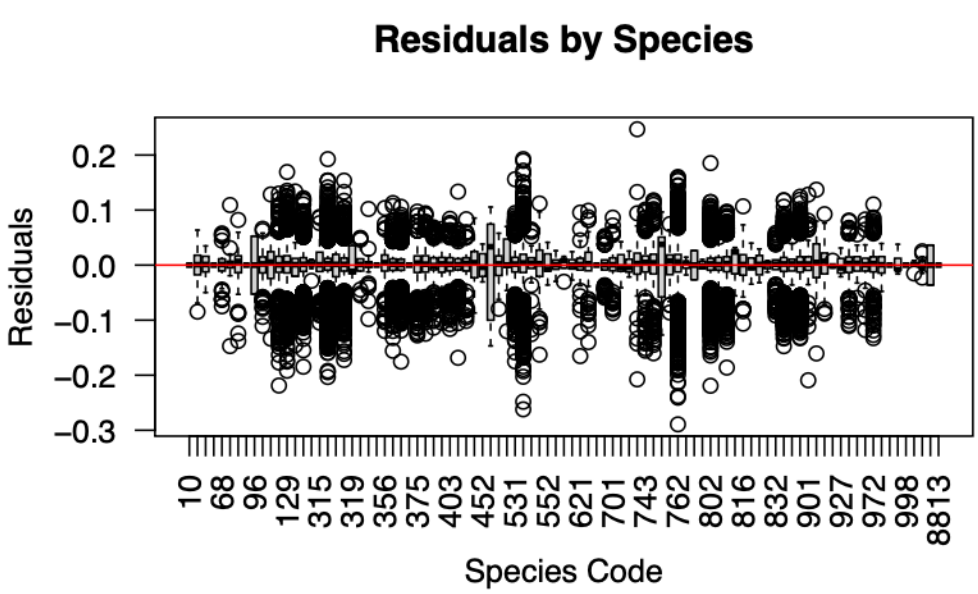
```r
#Residuals vs. Fitted values
plot(fitted(model_simple), resid(model_simple), main = "Residuals vs Fitted")
abline(h = 0, col = "red")
```



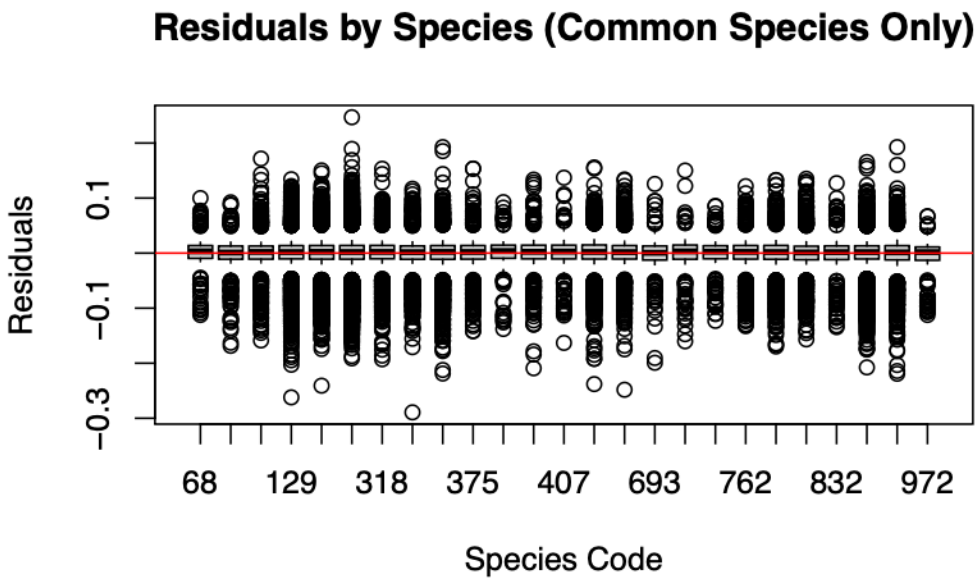

```r
#Residuals vs. STDAGE
model_used_data <- model_data[rownames(model_data) %in% names(resid(model_simple)), ]

plot(model_used_data$STDAGE, resid(model_simple), main = "Residuals vs STDAGE")
abline(h = 0, col = "red")
```




```r
#Boxplot vs Residuals SPCD (categorical)
# Find rows used in model
used_rows <- as.numeric(names(resid(model_simple)))

# Subset original data accordingly
species_used <- model_data$SPCD[used_rows]
residuals_used <- resid(model_simple)

boxplot(residuals_used ~ species_used,
        main = "Residuals by Species",
        xlab = "Species Code",
        ylab = "Residuals",
        las = 2)
abline(h = 0, col = "red")
```




```r
#simpler visualization with fewer species
# Show only common species (e.g., those with >500 observations)
common_species <- model_data %>%
  count(SPCD) %>%
  filter(n > 500) %>%
  pull(SPCD)

model_data_common <- model_data %>%
  filter(SPCD %in% common_species) %>%
  mutate(residuals = resid(model_simple)[SPCD %in% common_species])

boxplot(residuals ~ SPCD, data = model_data_common,
        main = "Residuals by Species (Common Species Only)",
        xlab = "Species Code",
        ylab = "Residuals")
abline(h = 0, col = "red")
```



Residuals vs Fitted

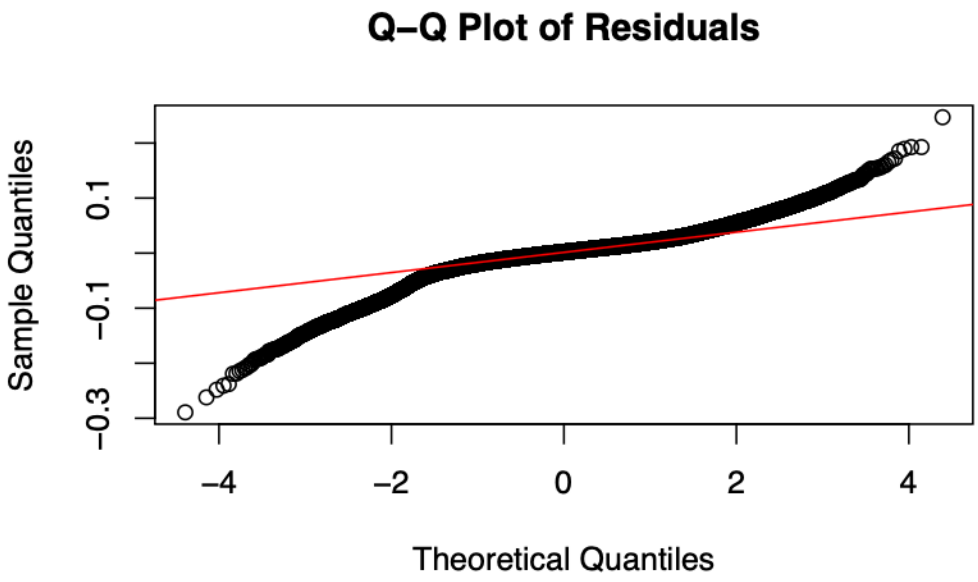
Good:

- Residuals centered around zero
- No clear pattern or trend

Issues:

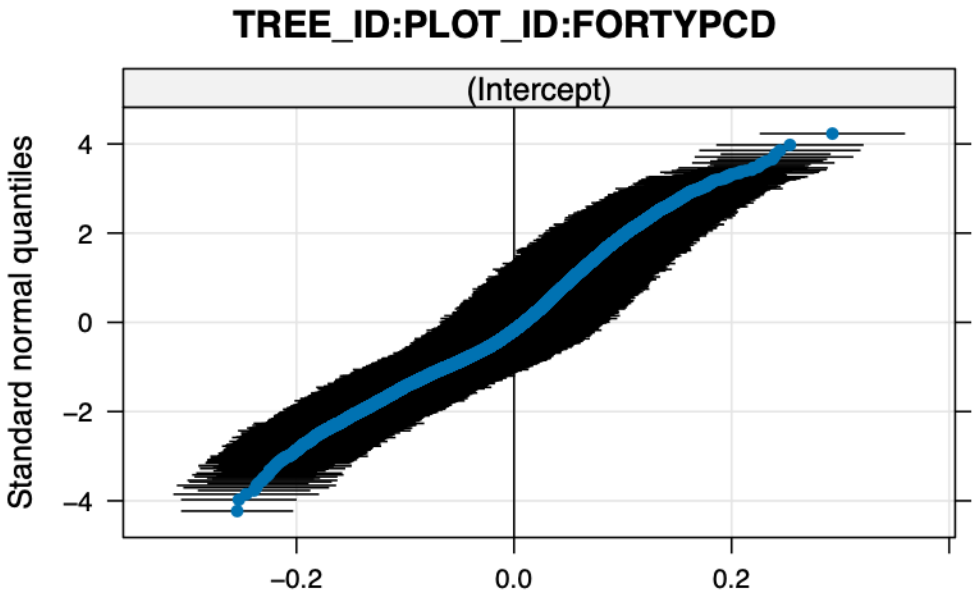
- **Funnel shape** - variance decreases as fitted values increase (slight heteroscedasticity)
- **Diagonal banding** - this is from the discrete nature of your growth data (rounded to 0.01 inches)

```
# Q-Q plot for normality of residuals
qqnorm(resid(model_simple), main = "Q-Q Plot of Residuals")
qqline(resid(model_simple), col = "red")
```

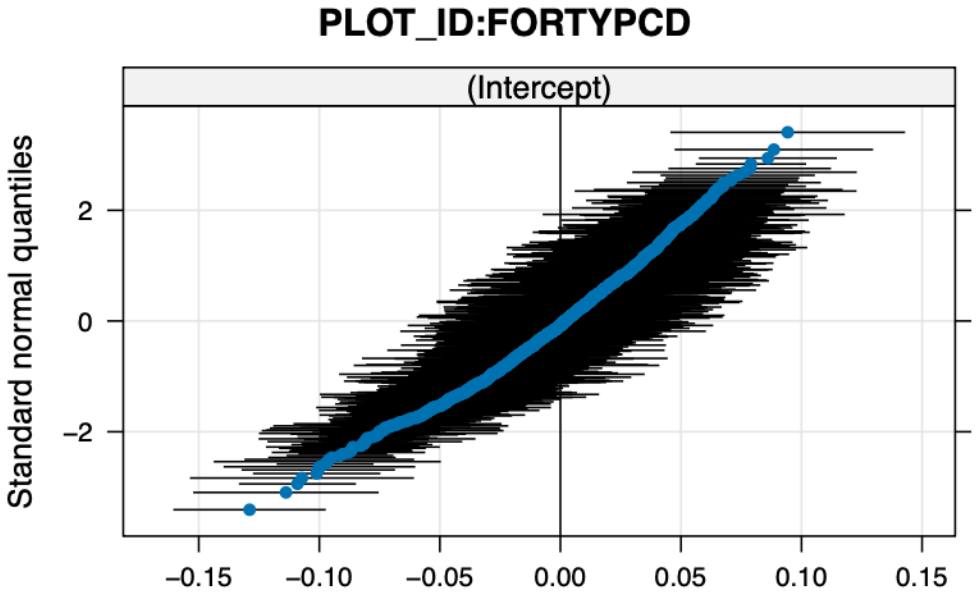



```
par(mfrow = c(2, 3))
# Check normality of random effects
library(lattice)

# Q-Q plots for each random effect level
qqmath(ranef(model_simple))
```

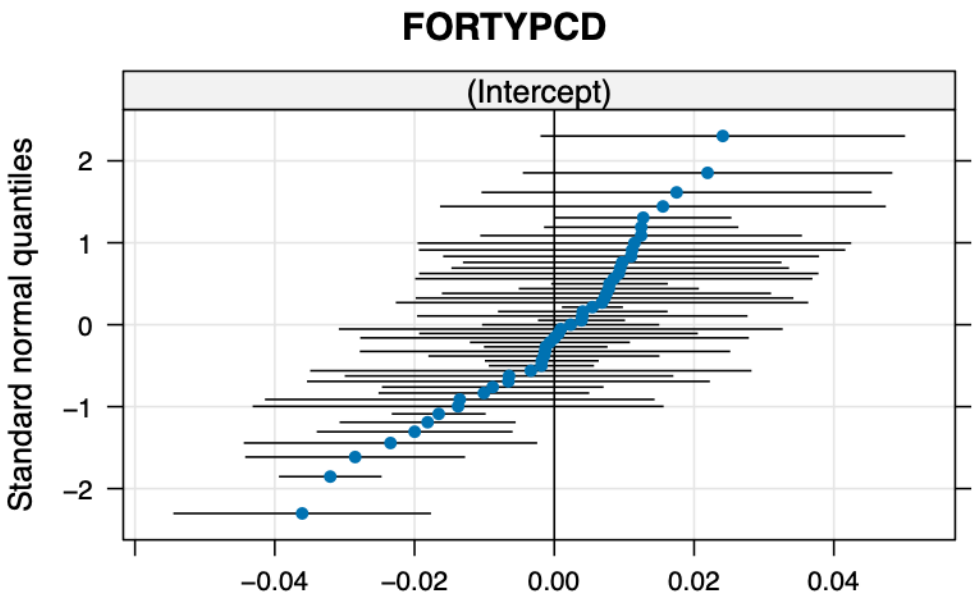
```
$`TREE_ID:PLOT_ID:FORTYPCD`
```



$`PLOT_ID:FORTYPCD`

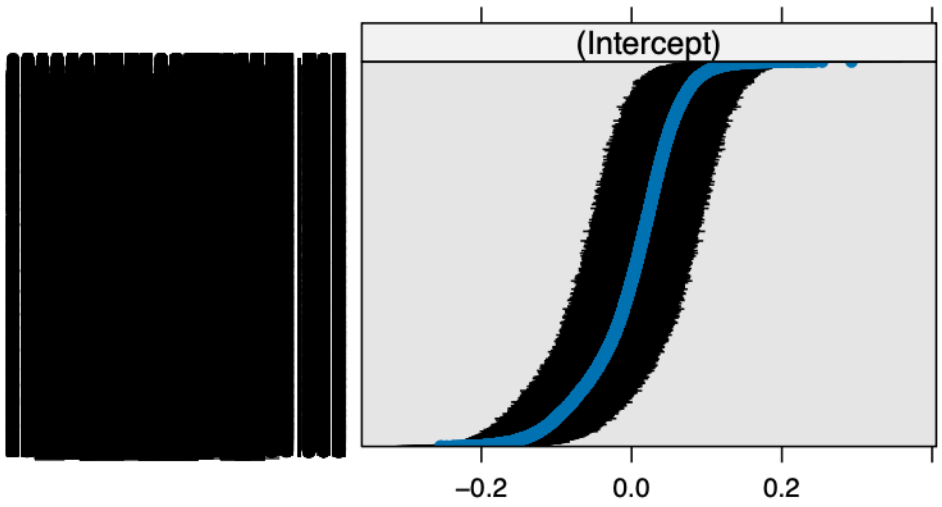


$FORTYPCD

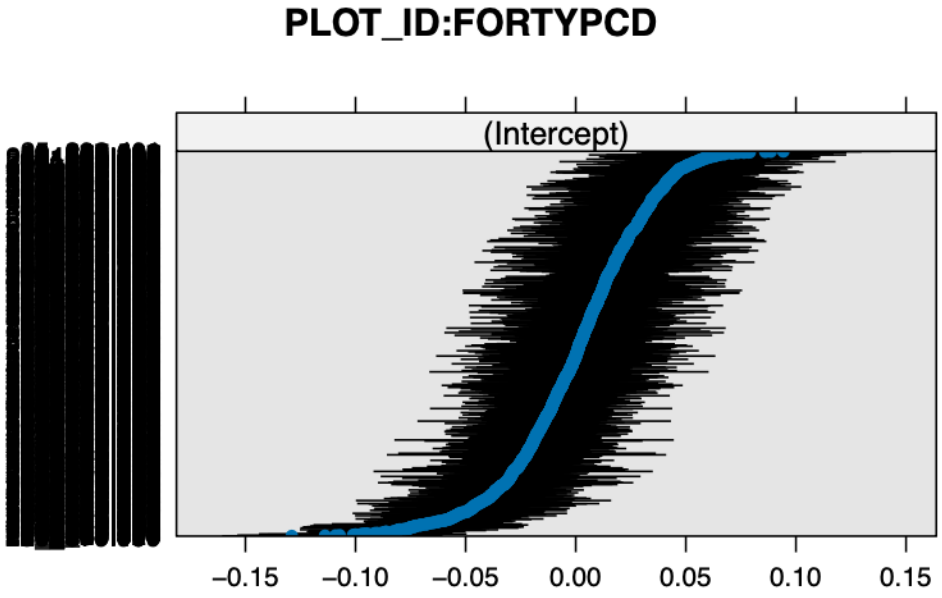



```
# Dotplot of random effects
dotplot(ranef(model_simple))
```

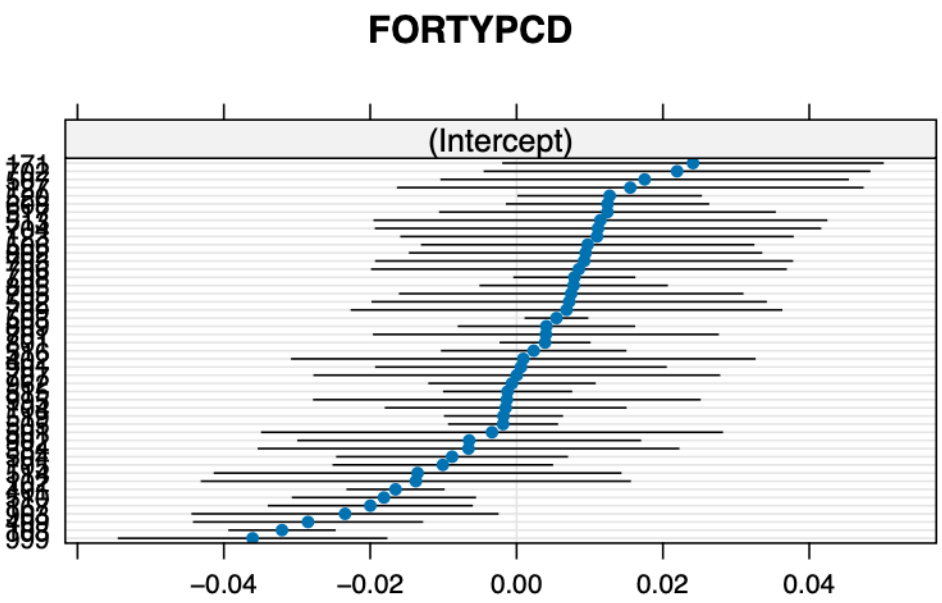
$`TREE_ID:PLOT_ID:FORTYPCD`




```
$`PLOT_ID:FORTYPCD`
```



$FORTYPCD




```
# Extract random effects
random_effects <- ranef(model_simple)
str(random_effects)
```

```
List of 3
 $ TREE_ID:PLOT_ID:FORTYPCD:'data.frame':   42905 obs. of  1 variable:
  ..$ (Intercept): num [1:42905] 0.0179 0.0102 0.0102 -0.0105 0.045 ...
  ..- attr(*, "postVar")= num [1, 1, 1:42905] 0.000501 0.000501 0.000501 0.000501 0.000501 .
 $ PLOT_ID:FORTYPCD        :'data.frame':   1531 obs. of  1 variable:
  ..$ (Intercept): num [1:1531] -0.0726 -0.02011 0.01717 -0.02865 -0.00958 ...
  ..- attr(*, "postVar")= num [1, 1, 1:1531] 0.00032 0.000128 0.000243 0.000133 0.000113 ...
 $ FORTYPCD                :'data.frame':   47 obs. of  1 variable:
  ..$ (Intercept): num [1:47] -0.01376 -0.03206 -0.00149 -0.01008 0.01099 ...
  ..- attr(*, "postVar")= num [1, 1, 1:47] 2.23e-04 1.35e-05 6.99e-05 5.81e-05 1.86e-04 ...
 - attr(*, "class")= chr "ranef.mer"
```

References

USFS Forest Inventory and Analysis (FIA) Program

- **Main site:** https://research.fs.usda.gov/programs/fia
- **Data Download:** https://apps.fs.usda.gov/fia/datamart/datamart.html
- **FIADB Population Estimation User Guide**: https://research.fs.usda.gov/understory/fiadb-population-estimation-user-guide
- **FIA Database Description and Users Manual:** https://www.fs.usda.gov/rm/pubs/rmrs_gtr245.pdf

Claude.ai: https://claude.ai/new

Supplementary Material

GitHub Link: https://github.com/rvithayathil/678-Final-Project

FIA Supplements:

Identifying tree characteristics:

Classification	Selection criteria
Live trees	TREE.STATUSCD = 1
Standing dead trees	TREE.STATUSCD = 2, TREE.STANDING_DEAD_CD = 1
Growing-stock trees	TREE.STATUSCD = 1, TREE.TREECLCD = 2

Identifying land classes (COND table):

Classification	Selection criteria
Forest land	COND_STATUS_CD = 1
Timberland	COND_STATUS_CD = 1, SITECLCD <7, RESERVCD = 0
Nonforest land	COND_STATUS_CD = 2
Reserved forest land	COND_STATUS_CD = 1, RESERVCD = 1
Unreserved forest land	COND_STATUS_CD = 1, RESERVCD = 0
Productive forest land	COND_STATUS_CD = 1, SITECLCD <7
Unproductive forest land	COND_STATUS_CD = 1, SITECLCD = 7

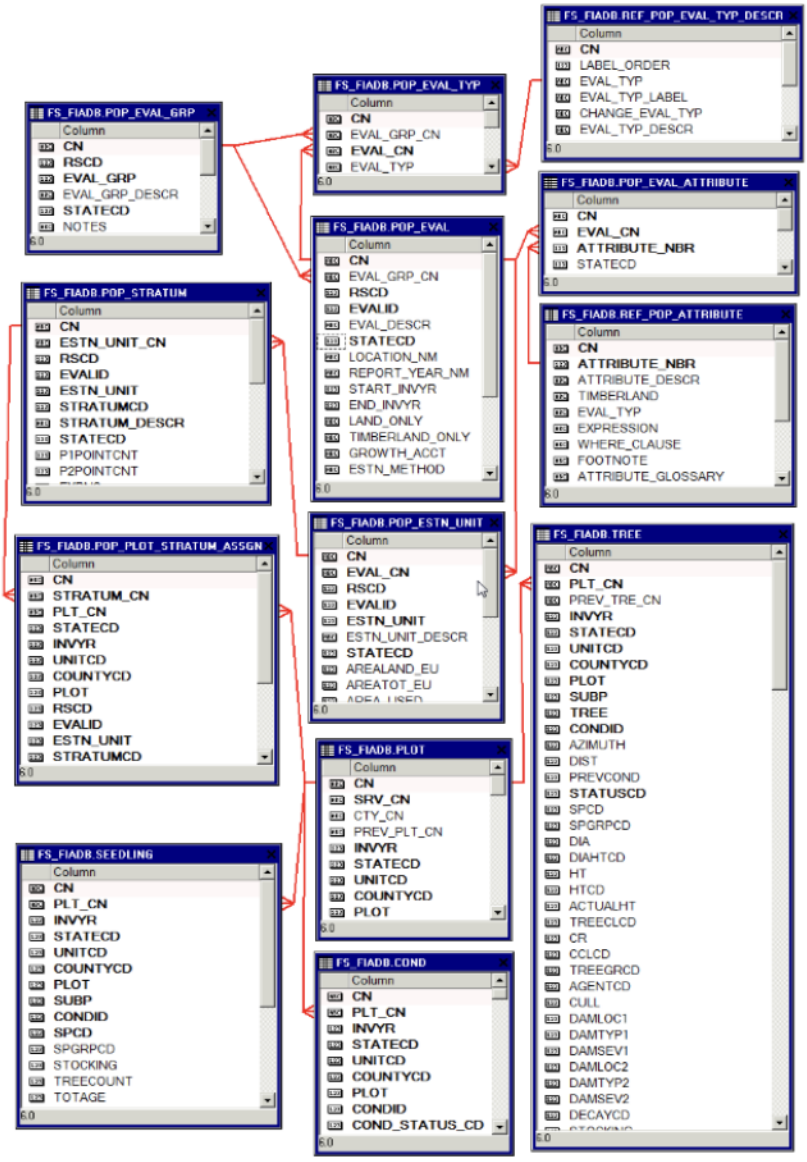


Figure 3-1: An abbreviated diagram of select FIADB tables. Note that there are more columns in each table than are shown.