Modeling Tree Diameter Growth

Michael Betancourt

January 2024

Table of contents

Because tree growth is sensitive to the surrounding environmental, in particular the ambient climate, continual measurements of tree growth are a powerful way to learn about the history of changes to that environment. In this exercise we'll explore a Bayesian approach to modeling and drawing inferences from tree diameter data measurements, with a focus on capturing the underlying data generating process. At the end I'll discuss how this exercise was integrated into a two-day interactive workshop.

1 Familiarizing Ourselves With The Data

Before building any models we need to understand is being modeled. In particular we need to develop our understanding of the data and its provenance.

1.1 Data Provenance

The data with which we'll be working consists of **diameter at breast height**, or **DBH**, measurements. These measurements are made by wrapping a measuring tape around the trunk of a tree at a height of 1.37 meters, approximating the breast height of an average human (Figure 1). This height is high enough to avoid the tree roots and forest floor but low enough to be accessible to researchers making the measurements. Diameter observations are given by dividing the recorded circumference by π.

When the diameter of a tree is first measured it is marked with a tag to indicate the breast height used. All future diameter measurements are made at the tag height to ensure consistency across time.

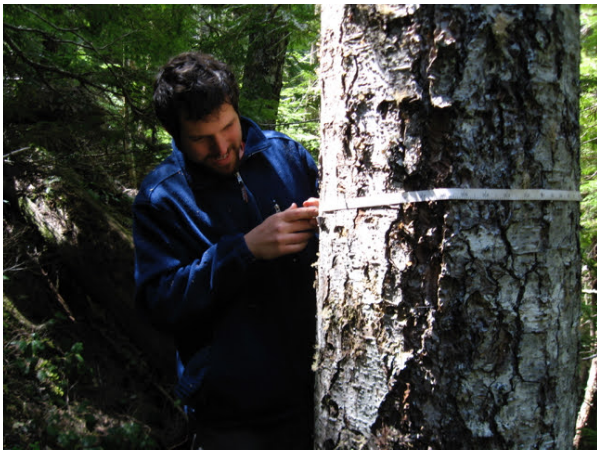


Figure 1: Tree diameters are measured by wrapping a measuring tape around the trunk of a tree 1.37 meters above the ground and then dividing the recorded circumference by π. Here Robert Ettinger is recording tree diameters in the Pacific Northwest. Photo :copyright: Ailene Ettinger, used with permission.

The H.J. Andrews Experimental Forest and Long Term Ecological Research Program has censused trees in forests across the Pacific Northwest every five or six years for decades. In addition to the tree diameter additional physiological attributes, such as tree vigor and the condition of the tree crown and stem are also recorded. If a tree is found to have died since it was last measured then a mortality assessment is also performed. For intimate detail on the measurement procedures and the structure of the resulting data see the Pacific Northwest Permanent Sample Plot Program Protocol for Tree Measurements and Mortality Assessments.

In this exercise we will be focusing on data collected from forests around Mount Rainier (Figure 2). These forests cover a wide range of elevations and other environmental conditions which make them particularly well-suited for studying various aspects of an evolving climate. That said here we will consider only trees that are isolated to a single stand of forest.

Figure 2: The forests around Mount Rainier span a diversity of environmental conditions. Photo courtesy of WikiMedia, https://commons.wikimedia.org/wiki/File:Mount_Rainier_panorama_2.jpg.

1.2 Code Environment Set Up

With the provenance established we're ready to take our first peak at the data itself. We'll start by configuring our local R environment for some reasonably aesthetic visualizations.

```
par(family="serif", las=1, bty="l",
    cex.axis=1, cex.lab=1, cex.main=1,
    xaxs="i", yaxs="i", mar = c(5, 5, 3, 5))

c_light <- c("#DCBCBC")
c_light_highlight <- c("#C79999")
c_mid <- c("#B97C7C")
c_mid_highlight <- c("#A25050")
c_dark <- c("#8F2727")
c_dark_highlight <- c("#7C0000")
```

```
c_light_teal <- c("#6B8E8E")
c_mid_teal <- c("#487575")
c_dark_teal <- c("#1D4F4F")
```

1.3 Exploratory Data Analysis

We now read in our data for stand AV06. The columns and their possible values are carefully documented in Pacific Northwest Permanent Sample Plot Program Protocol for Tree Measurements and Mortality Assessments.

```
stand_tree_data <- read.csv('data/TV01002_v17_AV06.csv', skip=5)
```

To ensure that visualizations fit nicely into this document I'm going to restrict consideration to only the first 22 trees in the stand.

```
tree_ids <-  unique(stand_tree_data$TREEID)

N_trees <- 22
tree_ids <- tree_ids[1:N_trees]
```

In this nominal format repeated measurements of a given tree are not necessarily organized next to each other. Because we are particularly interested in tree diameter growth it will be much more convenient to collect repeated measurements together and construct indices that allow us to readily isolate the data for a single tree.

Note how we're using our understanding of the data generating process to motivate how we format and visualize the data. Exploratory data analysis is always done in the context of as assumed, if conceptual, data generating process!

```
tree_N_obs <- c()
tree_start_idxs <- c()
tree_end_idxs <- c()
tree_years <- c()
tree_dbhs <- c()
tree_vigors <- c()
tree_species <- c()
tree_statuses <- c()
tree_notes <- c()

idx <- 1
```

```
for (id in tree_ids) {
  # Isolate observations for given tree
  tree_data <- stand_tree_data[stand_tree_data['TREEID'] == id,]

  years <- tree_data[,'YEAR']
  dbhs <- tree_data[,'DBH']
  vigors <- tree_data[,'TREE_VIGOR']
  species <- tree_data[,'SPECIES']
  statuses <- tree_data[,'TREE_STATUS']
  notes <- tree_data[,'CHECK_NOTES']

  # Append tree data
  N_obs <- length(years)
  tree_N_obs <- c(tree_N_obs, N_obs)

  start_idx <- idx
  end_idx <- idx + N_obs - 1
  idx <- idx + N_obs
  tree_start_idxs <- c(tree_start_idxs, start_idx)
  tree_end_idxs <- c(tree_end_idxs, end_idx)

  tree_years <- c(tree_years, years)
  tree_dbhs <- c(tree_dbhs, dbhs)
  tree_species <- c(tree_species, species)
  tree_vigors <- c(tree_vigors, vigors)
  tree_statuses <- c(tree_statuses, statuses)
  tree_notes <- c(tree_notes, notes)
}
```

Each tree diameter measurement is accompanied by many different assessments of the tree's condition. For our initial investigation I take a look at **tree vigor** which is a general quantification of health. We can communicate tree vigor in our visualizations by coloring each observation according to the five possible statuses.

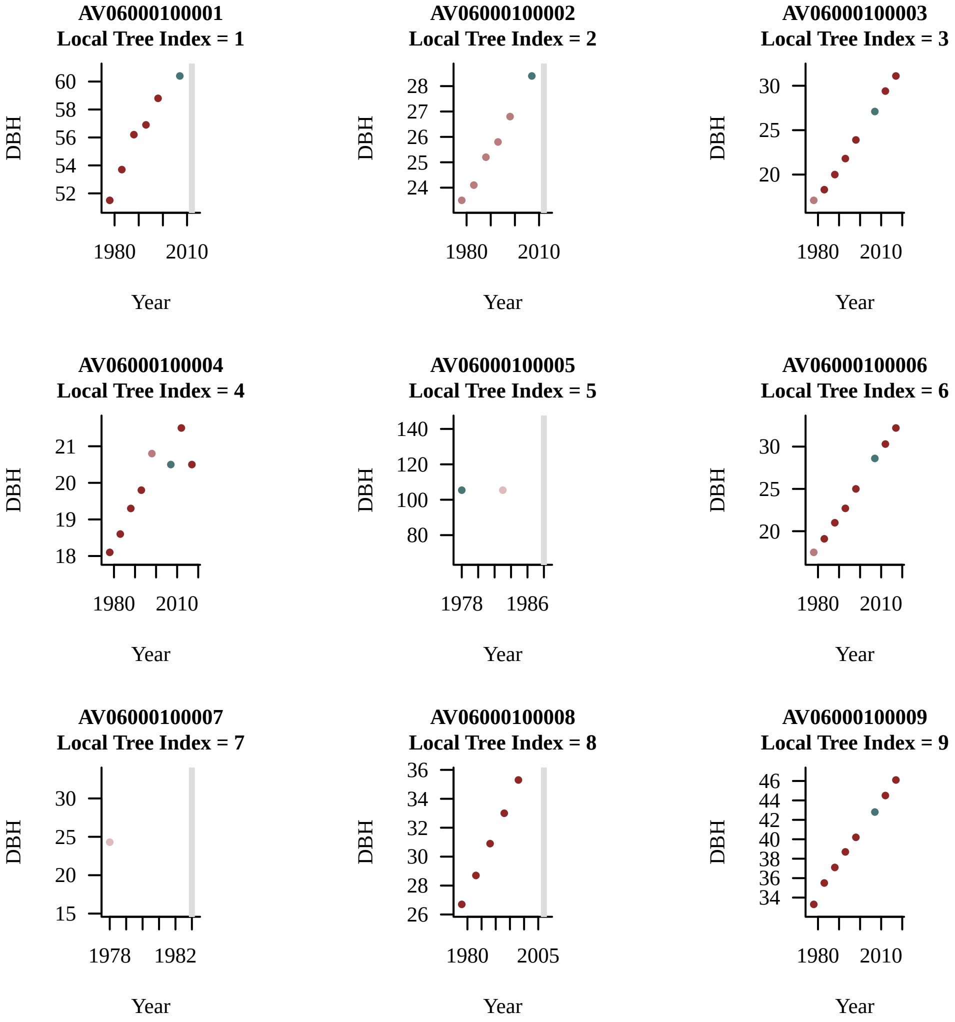
```
vigor_cols <- list("1" = c_dark, "2" = c_mid, "3" = c_light,
                   "U" = c_mid_teal, "M" = "black")
```

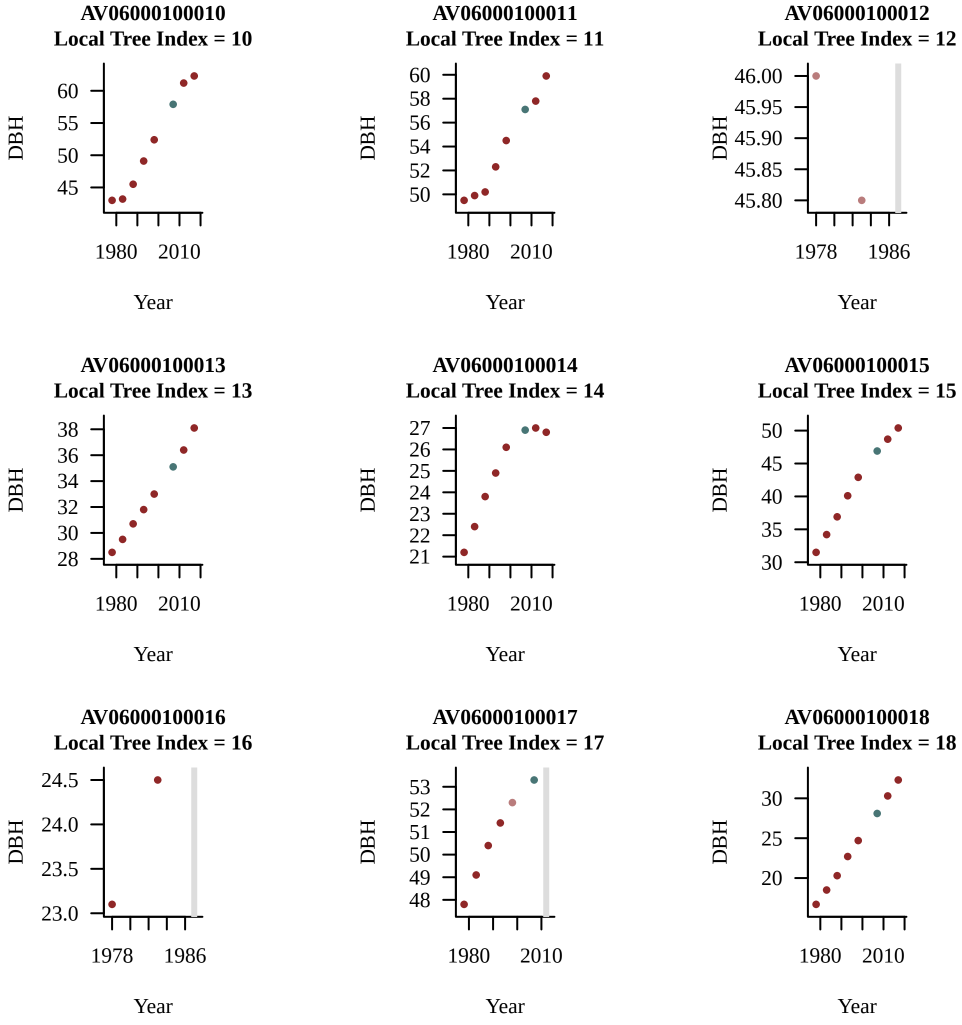
This reorganization of the data makes it straightforward to plot a time series of measurements for each tree. Each observation is colored according to the tree vigor. Observations of dead trees with `TREE_VIGOR = "M"` are given `NA` diameters values; these will be denoted with gray rectangles at the corresponding observation year.

```
plot_tree_data <- function(t) {
  tree_idxs <- tree_start_idxs[t]:tree_end_idxs[t]

  years <- tree_years[tree_idxs]
  dbhs <- tree_dbhs[tree_idxs]
  vigors <- tree_vigors[tree_idxs]
  cols = sapply(vigors, function(v) vigor_cols[[v]])

  xlims <- range(years)
  delta <- xlims[2] - xlims[1]
  xlims[1] <- xlims[1] - 0.1 * delta
  xlims[2] <- xlims[2] + 0.1 * delta

  ylims <- range(dbhs, na.rm=TRUE)
  delta <- ylims[2] - ylims[1]
  ylims[1] <- ylims[1] - 0.1 * delta
  ylims[2] <- ylims[2] + 0.1 * delta

  plot(years, dbhs,
       main=paste0(tree_ids[t], "\nLocal Tree Index = ", t),
       pch=16, cex=0.8, col=cols,
       xlab="Year", xlim=xlims, ylab="DBH", ylim=ylims)

  na_idxs <- which(is.na(dbhs))
  for (na_idx in na_idxs) {
    abline(v=years[na_idx], col='#DDDDDD', lwd=3)
  }
}
```

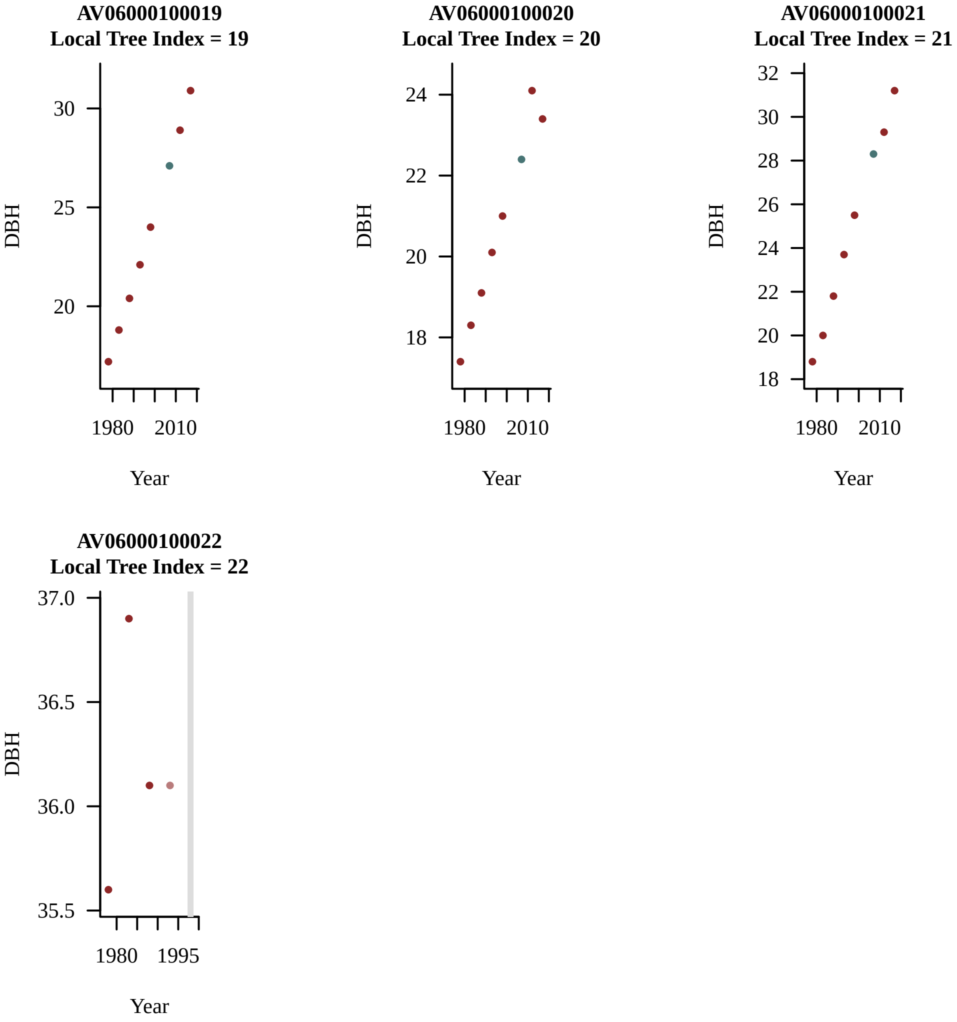
So how does our data look?

```
par(mfrow=c(3, 3))
for (t in 1:18) {
  plot_tree_data(t)
}
```


AV06000100001
Local Tree Index = 1
DBH
60
58
56
54
52
1980
2010
Year
AV06000100002
Local Tree Index = 2
DBH
28
27
26
25
24
1980
2010
Year
AV06000100003
Local Tree Index = 3
DBH
30
25
20
1980
2010
Year
AV06000100004
Local Tree Index = 4
DBH
21
20
19
18
1980
2010
Year
AV06000100005
Local Tree Index = 5
DBH
140
120
100
80
1978
1986
Year
AV06000100006
Local Tree Index = 6
DBH
30
25
20
1980
2010
Year
AV06000100007
Local Tree Index = 7
DBH
30
25
20
15
1978
1982
Year
AV06000100008
Local Tree Index = 8
DBH
36
34
32
30
28
26
1980
2005
Year
AV06000100009
Local Tree Index = 9
DBH
46
44
42
40
38
36
34
1980
2010
Year

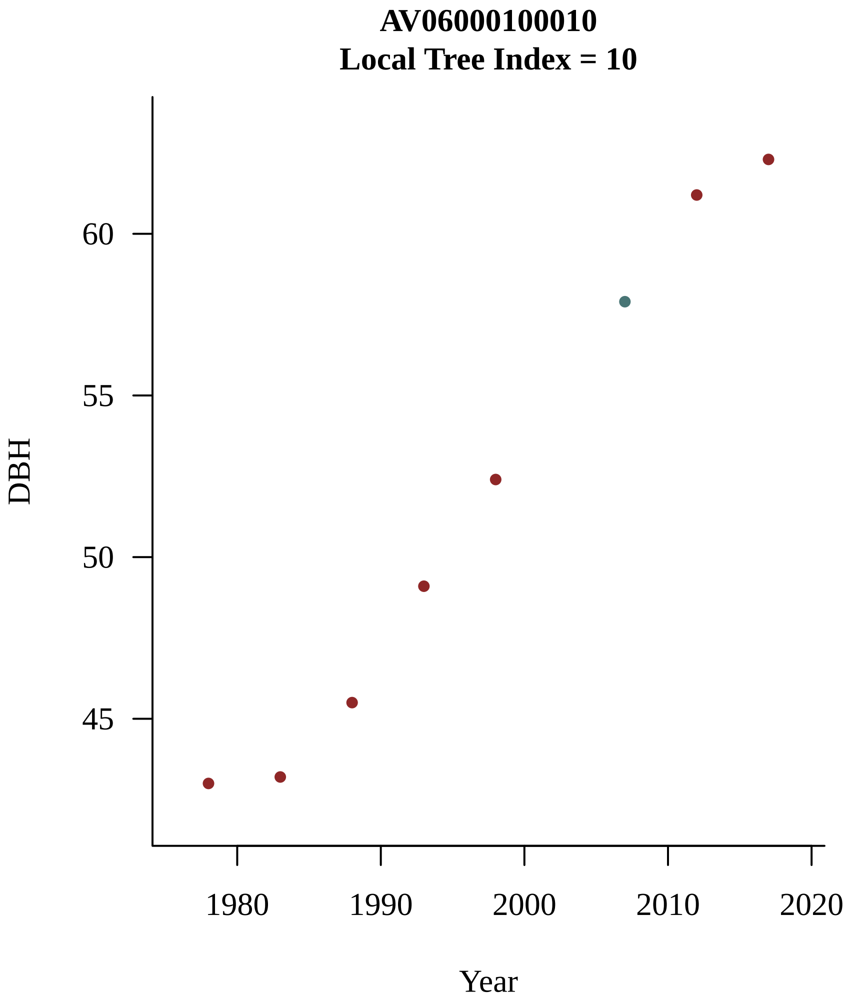


```r
par(mfrow=c(2, 3))
for (t in 19:N_trees) {
  plot_tree_data(t)
}
```



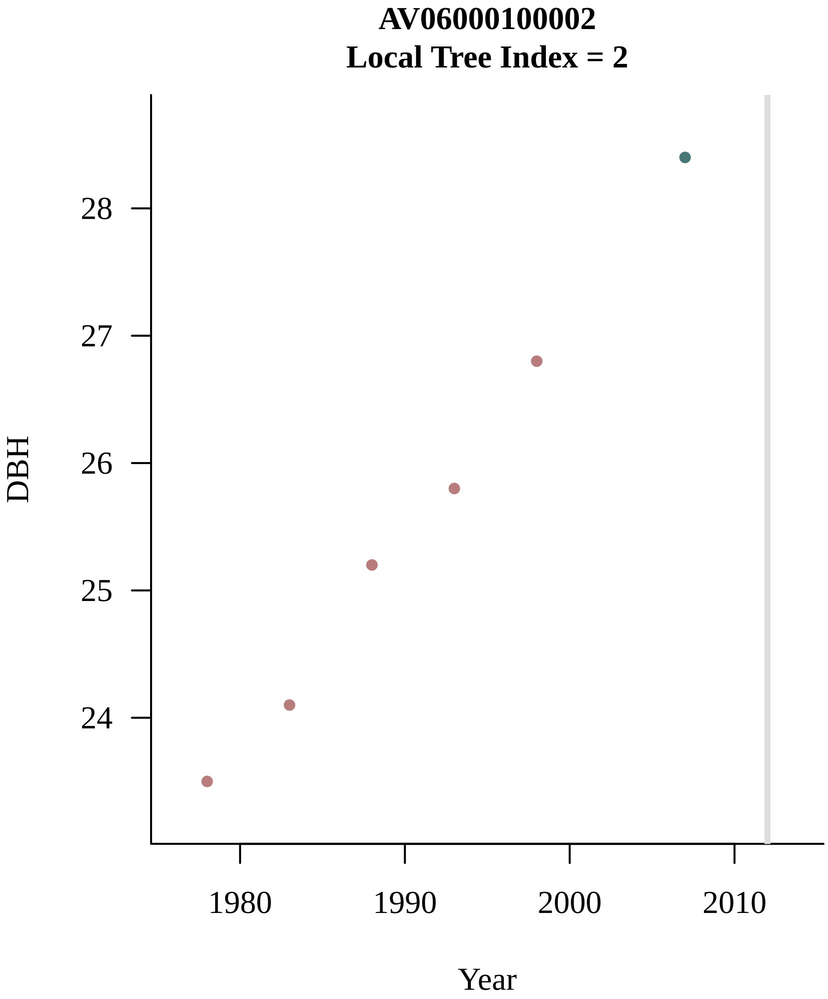
Most of the trees exhibit clean sequences of measurements that increase with time, consistent with persistent tree growth.

```
par(mfrow=c(1, 1))

plot_tree_data(10)
```



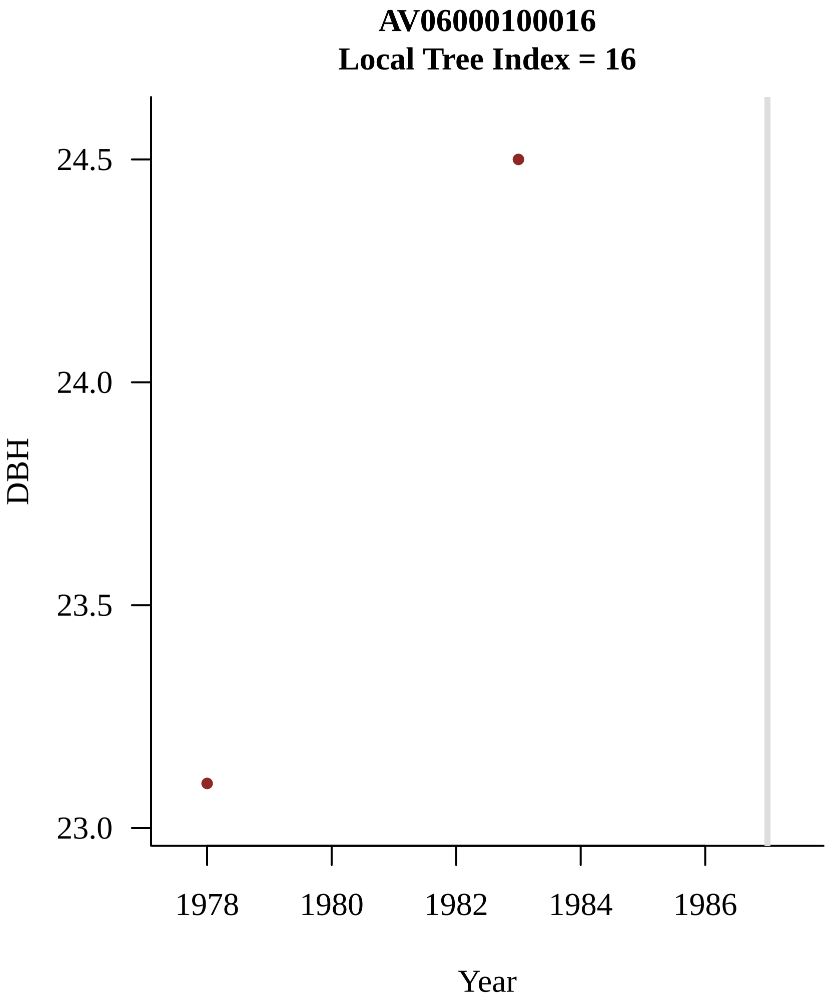
Some trees, however, have perished resulting in `NA` diameter measurements.

```
plot_tree_data(2)
```



Still others contain only a few measurements which limits how well we can constrain potential growth models.

```
plot_tree_data(16)
```



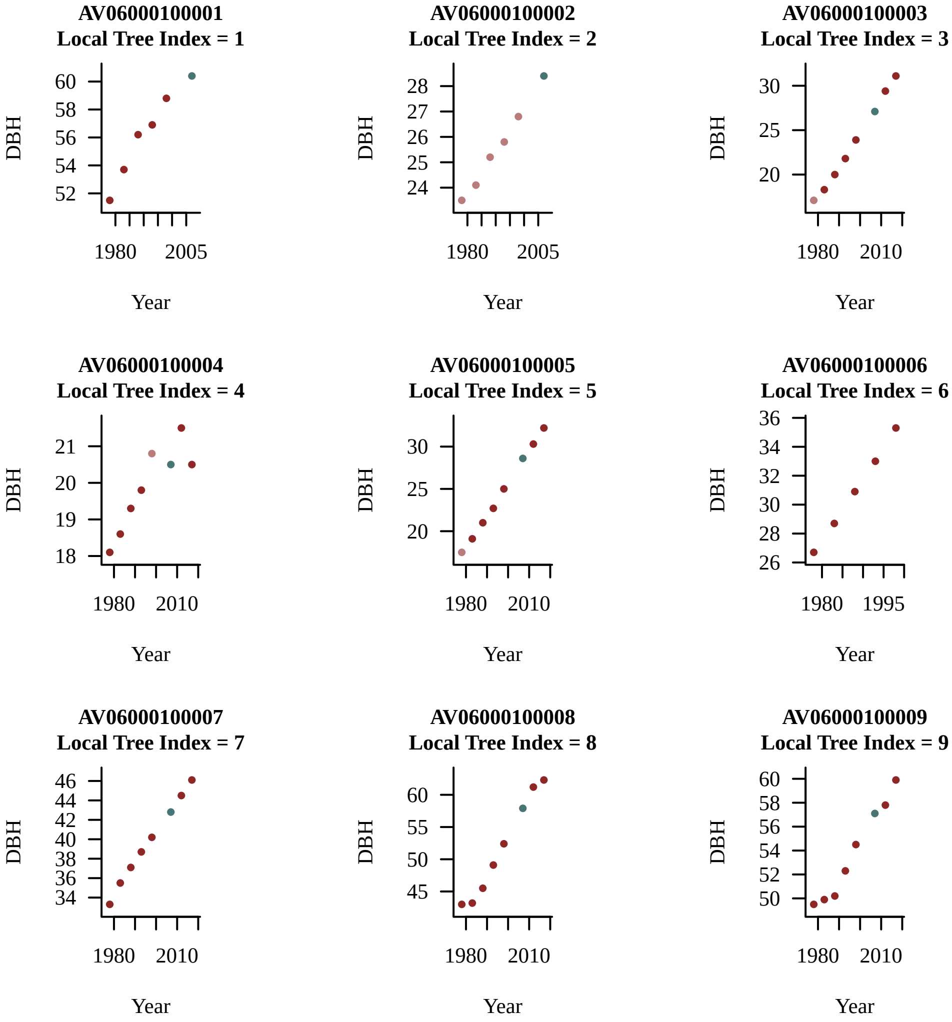
For our initial analysis let's clean up the data by removing observations of dead trees, `DBH = NA` and `TREE_VIGOR = "M"`. Moreover let's remove trees without at least four observations entirely. Why at least four observations? Eventually we will want to use four-parameter growth models that require at least four observations to be identified.

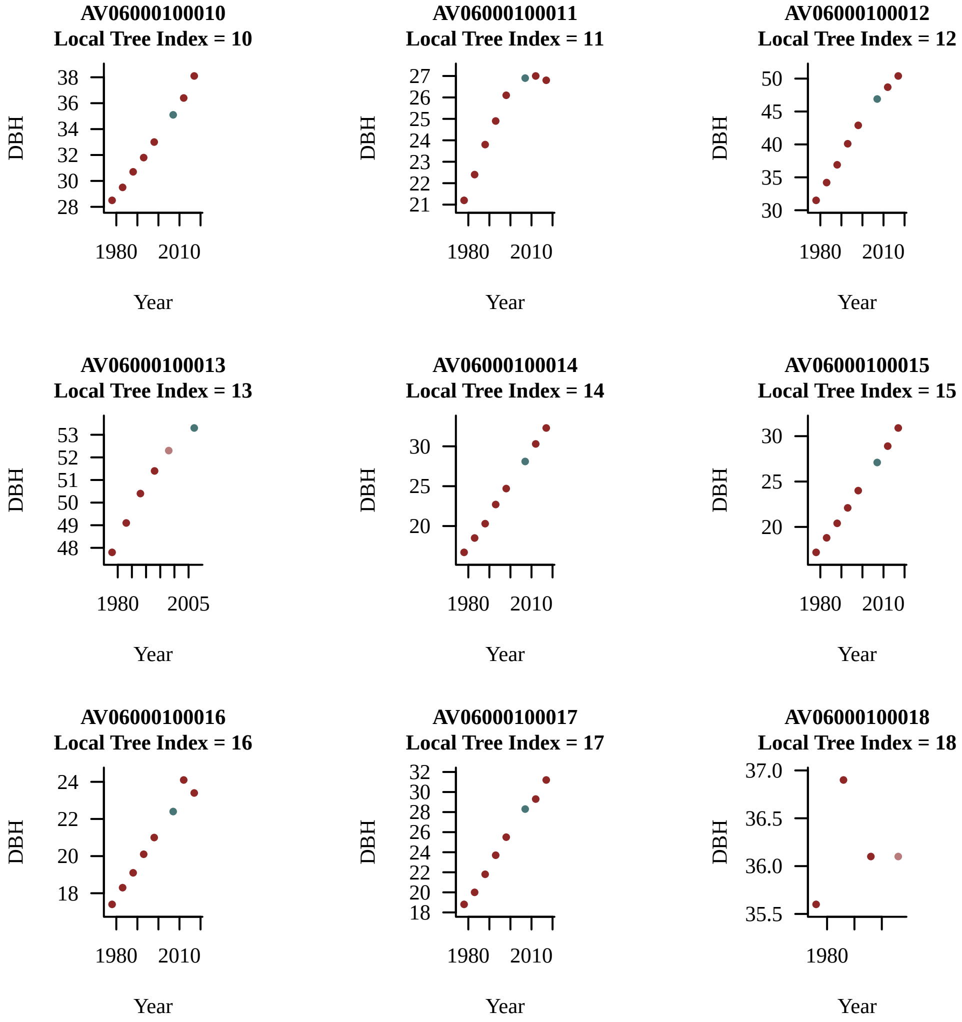
```
good_tree_ids <- c()
tree_N_obs <- c()
tree_start_idxs <- c()
tree_end_idxs <- c()
tree_years <- c()
tree_dbhs <- c()
tree_vigors <- c()
tree_species <- c()
```

```
tree_statuses <- c()
tree_notes <- c()

idx <- 1

for (id in tree_ids) {
  # Isolate observations for given tree
  tree_data <- stand_tree_data[stand_tree_data['TREEID'] == id,]

  years <- tree_data[,'YEAR']
  dbhs <- tree_data[,'DBH']
  vigors <- tree_data[,'TREE_VIGOR']
  species <- tree_data[,'SPECIES']
  statuses <- tree_data[,'TREE_STATUS']
  notes <- tree_data[,'CHECK_NOTES']

  # Remove post-mortality observations
  good_obs <- which(vigors != "M")
  years <- years[good_obs]
  dbhs <- dbhs[good_obs]
  vigors <- vigors[good_obs]
  species <- species[good_obs]
  statuses <- statuses[good_obs]
  notes <- notes[good_obs]

  # Drop tree if there are not at least four observations
  N_obs <- length(years)
  if (N_obs < 4) next

  # Append tree data
  good_tree_ids <- c(good_tree_ids, id)
  tree_N_obs <- c(tree_N_obs, N_obs)

  start_idx <- idx
  end_idx <- idx + N_obs - 1
  idx <- idx + N_obs
  tree_start_idxs <- c(tree_start_idxs, start_idx)
  tree_end_idxs <- c(tree_end_idxs, end_idx)

  tree_years <- c(tree_years, years)
  tree_dbhs <- c(tree_dbhs, dbhs)
  tree_vigors <- c(tree_vigors, vigors)
  tree_species <- c(tree_species, species)
  tree_statuses <- c(tree_statuses, statuses)
  tree_notes <- c(tree_notes, notes)
}

N <- length(tree_years)
N_trees <- length(good_tree_ids)
```

This leaves 25 trees with well-behaved enough observations to inform the common growth models that we will consider below.

```
par(mfrow=c(3, 3))

for (t in 1:N_trees) {
  plot_tree_data(t)
}
```


AV06000100001
Local Tree Index = 1
DBH
60
58
56
54
52
1980
2005
Year
AV06000100002
Local Tree Index = 2
DBH
28
27
26
25
24
1980
2005
Year
AV06000100003
Local Tree Index = 3
DBH
30
25
20
1980
2010
Year
AV06000100004
Local Tree Index = 4
DBH
21
20
19
18
1980
2010
Year
AV06000100005
Local Tree Index = 5
DBH
30
25
20
1980
2010
Year
AV06000100006
Local Tree Index = 6
DBH
36
34
32
30
28
26
1980
1995
Year
AV06000100007
Local Tree Index = 7
DBH
46
44
42
40
38
36
34
1980
2010
Year
AV06000100008
Local Tree Index = 8
DBH
60
55
50
45
1980
2010
Year
AV06000100009
Local Tree Index = 9
DBH
60
58
56
54
52
50
1980
2010
Year



2 Modeling Tree Growth

Our exploratory data analysis didn't reveal any behavior that we didn't already expected from our understanding of the data generating process. We are now ready to model that data generating process, from the growth of an individual tree to the imperfect diameter measurements.

2.1 Tree Growth Models

Tree growth is a complicated process. In theory growth is moderated by the ambient environment which can change drastically across time. For example a tree with full exposure to sunlight should grow faster than a tree under the shadow of neighboring trees. Likewise limited access to water or other nutrients can inhibit growth. Moreover the physiological processes that drive growth might be complicated in of themselves, for example varying with the age of a tree.

In this section we will review some relatively simple growth models that might be useful for initial analyses. The construction of these models does require some nontrivial mathematics, and it is absolutely okay if those mathematics are inaccessible to some readers. Successful analyses often require collaboration between people who contribute scientific expertise and people who contribute mathematical expertise!

2.1.1 Linear Diameter Growth

A mathematically simple model for tree growth assumes that the diameter at breast height increases linearly with time (Figure 3)

$$d(t) = d_0 + \beta \cdot (t - t_0).$$

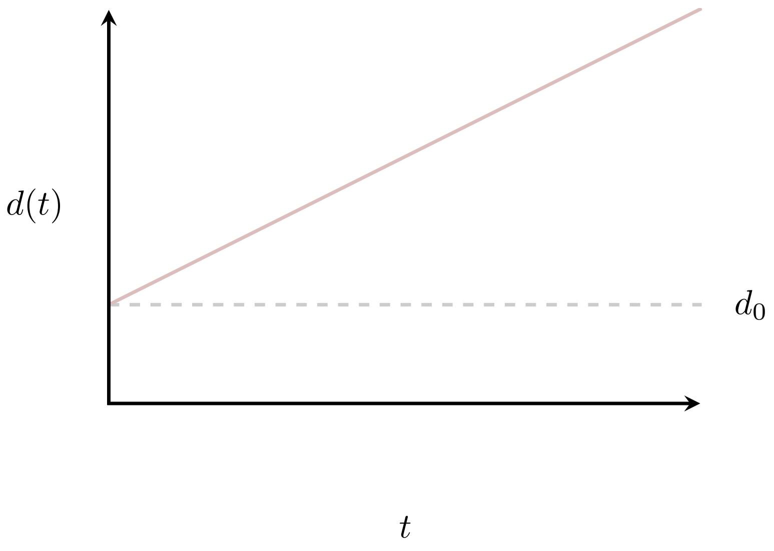


Figure 3: In a linear growth model a tree diameter grows linear with time.

Linear diameter growth over an extended time is often unrealistic, but it can be a reasonable approximation for more complicated growth dynamics over shorter intervals of time.

2.1.2 Linear Mass Growth

Linear diameter growth, however, also has awkward physiological consequences. As a tree grows a fixed increase in diameter requires larger and larger increases in overall mass.

If the energy inputs to a tree are fixed then an arguably more realistic model would assume that mass, and not diameter, increases linearly with time. To model this mathematically let's assume that we can approximate trees as cylinders with constant density so that the total mass is related to the diameter by

$$\begin{aligned} m &= \text{density} \cdot \text{volume} \\ &= \rho \cdot \pi \left(\frac{d}{2} \right)^2 h \\ &= \left(\frac{\pi}{4} \cdot \rho \cdot h \right) d^2 \\ &= C\, d^2. \end{aligned}$$

For convenience I've absorbed everything but the diameter into a single constant in that last step.

If the tree mass grows linearly with time and the height remaining constant then

$$\begin{aligned} \delta m \cdot t &= m(t) - m_0 \\ &= C\, d(t)^2 - C\, d_0^2, \end{aligned}$$

or upon solving for $d(t)$,

$$\begin{aligned} \delta m \cdot t &= C\, d(t)^2 - C\, d_0^2 \\ \frac{\delta m}{C} \cdot t &= d(t)^2 - d_0^2 \\ d(t)^2 &= d_0^2 + \frac{\delta m}{C} \cdot t \\ d(t) &= \sqrt{d_0^2 + \frac{\delta m}{C} \cdot t} \\ d(t) &= \sqrt{d_0^2 + \beta \cdot t}. \end{aligned}$$

Because of the square root function the diameter growth will decelerate as a tree ages, but never quite stop (Figure 4).

2.1.3 Gompertz Model

In practice both the linear and square root growth models will be too inflexible to adequately model tree diameter growth for the decades that our observations span. One particularly

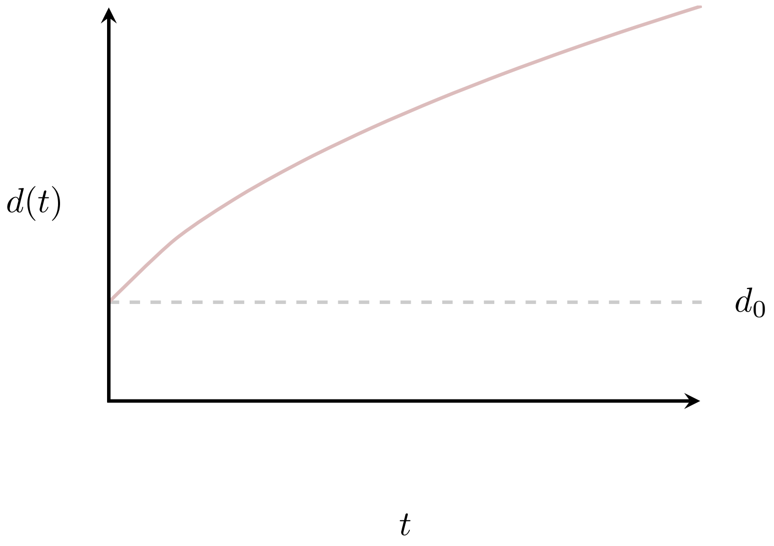


Figure 4: Under idealized assumptions a linear increase in the mass of a tree results in a square root increase in the diameter of the tree.

common class of models assume that the diameter growth follows a *sigmoidal curve*, allowing for accelerated growth at early ages, constant growth at intermediate ages, and decelerating growth at later ages. There are many mathematical curves that fit this qualitative shape, but a particularly common choice in many ecological applications is the **Gompertz** family of curves,

$$d(t) = d_{\text{max}} \, \exp(-b \exp(-c \, t)).$$

To ensure monotonically increasing growth all three parameters must be positive.

The interpretation of the parameter d_{max} is straightforward: it determines the asymptotic diameter that a tree will approach as it ages. On the other hand the interpretation of the other two parameters is less clear, which can frustrate a principled use of this model in a real analysis.

One way to make the Gompertz family of curves more interpretable is to find an alternative parameterization where each parameter directly corresponds to a meaningful behavior. For example we might parameterize the curves in terms of their linear behavior at intermediate times.

To that end let's introduce the intermediate time τ where a given Gompertz curve reached half of its asymptotic diameter,

$$\frac{d_{\max}}{2} = d(\tau) = d_{\max} \exp(-b \exp(-c\,\tau)),$$

or, solving for τ,

$$\tau = \frac{\log b - \log \log 2}{c}.$$

The rate of growth at τ is then given by

$$\beta = \frac{\mathrm{d}d}{\mathrm{d}t}(\tau) = \frac{\log 2}{2} d_{\max}\, c.$$

Conveniently the tree values $d_{\max}$, τ, and β completely determine a Gompertz curve. In terms of these parameters the Gompertz family of curves becomes (Figure 5)

$$d(t) = d_{\max} \exp\left(-\log 2 \cdot \exp\left(-\frac{2}{\log 2} \frac{\beta}{d_{\max}} \, (t - \tau) \right) \right)$$

which is fortunately straightforward to implement in practice.

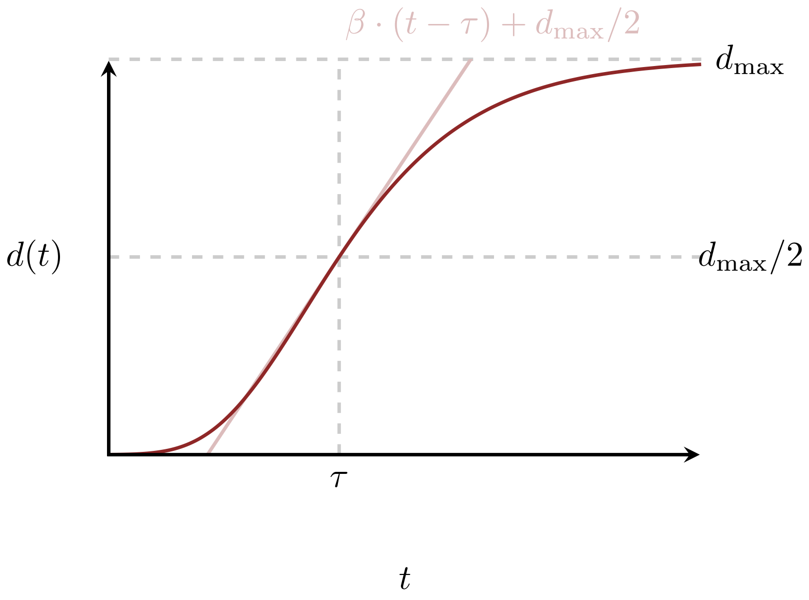


Figure 5: The Gompertz family of curves provides a sigmoidal model for tree diameter growth. Here we've parameterized each curve in terms of its asymptotic diameter, $d_{\max}$, the time at which curve reaches half of this value, τ, and the derivative at τ, β.

There are many ways of adding more flexible to the Gompertz model that could be useful when analyzing tree growth data. For example replacing $t - \tau$ with a polynomial would allow for more complicated, potentially even asymmetric, behavior at both young and old ages. Here we will consider a less sophisticated addition: a non-zero initial diameter,

$$d(t) = d_0 + \delta d \, \exp \left(- \log 2 \cdot \exp \left(- \frac{2}{\log 2} \, \frac{\beta}{\delta d} \, (t - \tau) \right) . \right)$$

This offset could, for instance, allow the model to accommodate different growth dynamics for very young trees before sigmoidal growth kicks in.

2.2 Measurement Model

Now that we have some candidate models for how tree diameter grows with time we need to consider how those diameters are actually measured.

The diameter at breast height measurement procedure is not infinitely precise. There can be errors in reading the measuring tape, misalignments of the measuring tape, non-uniforming in the diameter of the tree, inconsistencies from measurement to measurement, and more. A useful measurement model needs to account for not only how strongly these potential sources of variation can affect the diameter at breast height measurements but also what overall variation is consistent with our understanding of the procedure.

For example let's consider how much misalignment of the measuring tape can affect diameter measurements. As we did in the linear mass growth model let's assume that trees are well approximated by a cylinder of radius r, diameter $d = 2\,r$, and height h, with all lengths in units of centimeters.

In a perfect diameter at breast height measurement the measuring tape will form a circle with the same circumference as the tree at the tag height. Consequently we can recover the diameter of the tree by dividing the measured length by π,

$$d = \frac{C_{\text{flat}}}{\pi}.$$

When tape is tilted, however, it will form not a circle but rather an *ellipse*. If the maximum offset is given by δ then the shape of the ellipse will be determined by the semi-minor axis (Figure 6)

$$b = r = \frac{d}{2}$$

and the semi-major axis

$$a = \frac{1}{2} \sqrt{d^2 + \delta^2}.$$

As we would hope this reduces back to a circle with $a = b = r$ when $\delta \rightarrow 0$.

Unfortunately the perimeter of an ellipse does not admit a simple mathematical form; instead it is given by a complicated function known as an **elliptic integral**. For those interested in

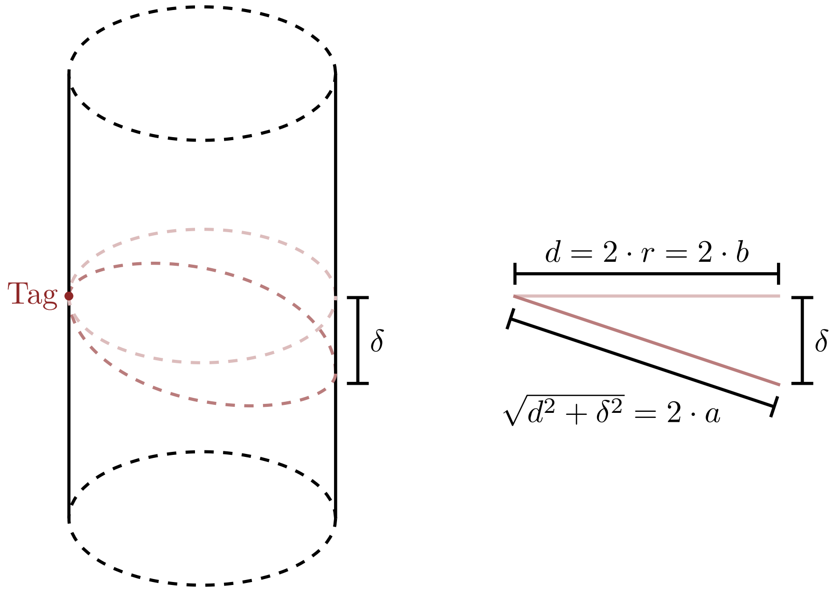

Tag
δ
$d = 2 \cdot r = 2 \cdot b$
δ
$\sqrt{d^2 + \delta^2} = 2 \cdot a$

Figure 6: When misaligned a measuring tape wrapped around a cylindrical tree defines an ellipse. In this case dividing the measured length by π does not give the tree diameter exactly.

learning more Wikipedia has a nice discussion at https://en.wikipedia.org/wiki/Ellipse#Circumference. We can, however, *approximate* the perimeter of an ellipse is a variety of different ways; again Wikipedia has a clear discussion, https://en.wikipedia.org/wiki/Ellipse#Arc_length.

For example the circumference of this tilted ellipse can be bounded above by

$$\begin{aligned}
C_{\text{tilt}} &\leq \sqrt{2}\,\pi\,\sqrt{b^2 + a^2} \\
&\leq \sqrt{2}\,\pi\,\sqrt{\left(\frac{d}{2}\right)^2 + \left(\frac{1}{2}\right)^2 \left(d^2 + \delta^2\right)} \\
&\leq \sqrt{2}\,\pi\,\frac{1}{2}\sqrt{d^2 + d^2 + \delta^2} \\
&\leq \sqrt{2}\,\pi\,\frac{1}{2}\sqrt{2d^2 + \delta^2} \\
&\leq \sqrt{2}\,\pi\,\frac{\sqrt{2}\,d}{2}\sqrt{1 + \left(\frac{\delta}{2\,d}\right)^2} \\
&\leq \pi\,d\,\sqrt{1 + \left(\frac{\delta}{2\,d}\right)^2}.
\end{aligned}$$

If the offset δ is much smaller than the actual tree diameter d then this upper bound will be a useful conservative approximation to the exact length of the tilted measuring tape,

$$C_{\text{tilt}} \approx \pi \, d \sqrt{1 + \left(\frac{\delta}{2 \, d} \right)^{2}}.$$

Using the perimeter of the tilted ellipse in place of the perimeter of a flat circle does not recover the diameter of the tree,

$$\frac{C_{\text{tilt}}}{\pi} = d \sqrt{1 + \left(\frac{\delta}{2 \, d} \right)^{2}} \geq d.$$

The larger perimeter always biases the recovered diameter to larger values. When δ is unknown this bias manifests as an asymmetric variability in the measurements.

Note that this measurement variability is not additive but rather *multiplicative*. Indeed multiplicative measurement variability is common when working with measurements of positive quantities. One way to recover additive measurement variability is to work with log diameters,

$$\log \frac{C_{\text{tilt}}}{\pi} = \log d + \frac{1}{2} \log \left(1 + \left(\frac{\delta}{2 \, d} \right)^{2} \right).$$

In practice we can sometimes approximate the multiplicative measurement variability with additive measurement variability,

$$\frac{C_{\text{tilt}}}{\pi} \approx d + d_{0} \left(\sqrt{1 + \left(\frac{\delta}{2 \, d} \right)^{2}} - 1 \right),$$

where d_0 is the baseline diameter at which we want this approximation to be best. This approximation is decent if all of our measurements are limited to a narrow range of tree diameters, but if our data spans measurements of very small and very large trees then the approximation can break down at the extremes.

Assuming that an additive measurement model is a good enough approximation what do these variabilities look like in practice? If the offsets never exceed ten percent of the actual tree diameter, which allows for some pretty strong misalignments, then

$$\frac{\delta}{d} \approx 0.1,$$

$$\frac{\delta}{2 \, d} \approx 0.05,$$

and

$$\sqrt{1 + \left(\frac{\delta}{2 \, d} \right)^{2}} \approx 1.001.$$

The deviation from the measured diameter to the true diameter is exceedingly small!

For example if $d_0 = 50\,\mathrm{cm}$ then the scale of the approximate additive measurement variability would be orders of magnitude smaller,

$$d_0 \left(\sqrt{1 + \left(\frac{\delta}{2\,d} \right)^2} - 1 \right) \approx 0.05\,\mathrm{cm},$$

which we can round up to about a millimeter. This may appear surprisingly small but it's all a consequence of the geometry: relatively large warpings of a circle don't change the diameter as much as we might expect!

Analysis of the other potential sources of variation gives similar results: multiplicative variation is more realistic but additive variation can be a reasonable approximation, and the scale of the linear variation is unlikely to exceed a millimeter or so. For this exercise we'll assume the simpler linear measurement model,

$$\hat{d}(t) \sim \mathrm{normal}(d(t), \sigma),$$

with our domain expertise excluding values of σ much larger than $0.1\,\mathrm{cm}$. To be particularly conservative we will allow for measurement variabilities as large as $0.25\,\mathrm{cm}$.

3 Demonstrative Analyses

Having brainstormed some possible models let's try to incorporate them into a Bayesian analysis using the probabilistic programming tool `Stan`, in particular it's `R` interface `RStan`.

3.1 Set Up

First and foremost we need to load the `rstan` package into our local `R` environment and configure it.

```r
library(rstan)
rstan_options(auto_write = TRUE)            # Cache compiled Stan programs
options(mc.cores = parallel::detectCores()) # Parallelize chains
parallel:::setDefaultClusterOptions(setup_strategy = "sequential")
```

Next we'll load my recommended Markov chain Monte Carlo diagnostic and analysis code. The code itself as well as documentation can be found at https://github.com/betanalpha/mcmc_diagnostics.

```
util <- new.env()
source('stan_utility_rstan.R', local=util)
```

3.2 Homogeneous Linear Model

Let's start as simply as possible: we'll assume a linear diameter growth model where all trees in the data set are allowed to have different initial diameters but all grow at the same rate. I will use normal containment prior models to softly constrain the initial heights below 150 cm, the growth rate below 2 cm per year in magnitude, and the measurement variability below 0.25 cm. For an introduction to normal containment prior models see https://betanalpha.github.io/assets/case_studies/prior_modeling.html#3_One-Dimensional_Expertise.

To facilitate the analysis of our Stan program we will also save the inferred diameter of each tree along a grid of times.

```
data <- mget(c("N", "N_trees", "tree_N_obs",
               "tree_start_idxs", "tree_end_idxs",
               "tree_years", "tree_dbhs"))

year_grid <- seq(1975, 2020, 1)
N_year_grid <- length(year_grid)

data$N_year_grid <- N_year_grid
data$year_grid <- year_grid
```

```
fit <- stan(file="stan_programs/homogeneous_linear_growth.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

The lack of diagnostic failures is consistent with Stan being able to accurately quantify our posterior uncertainties.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

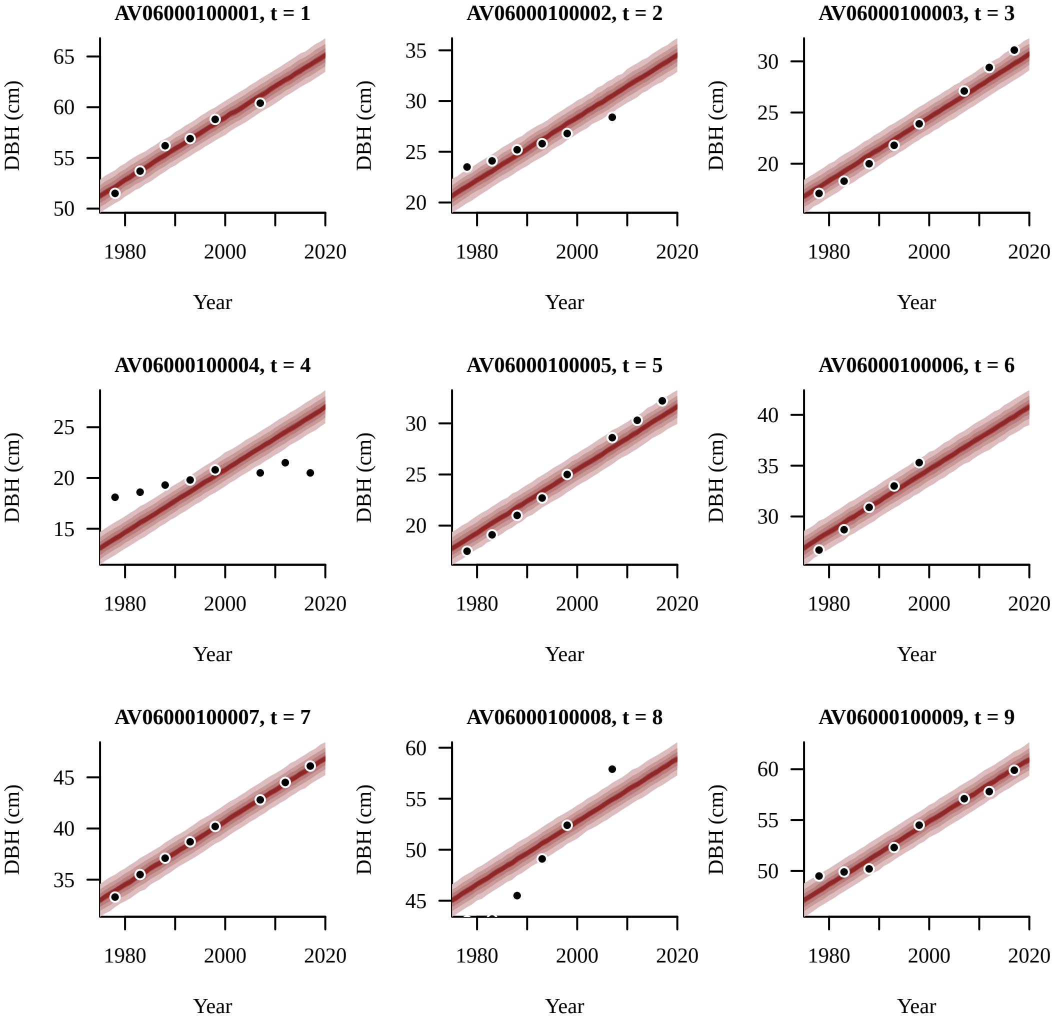
```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

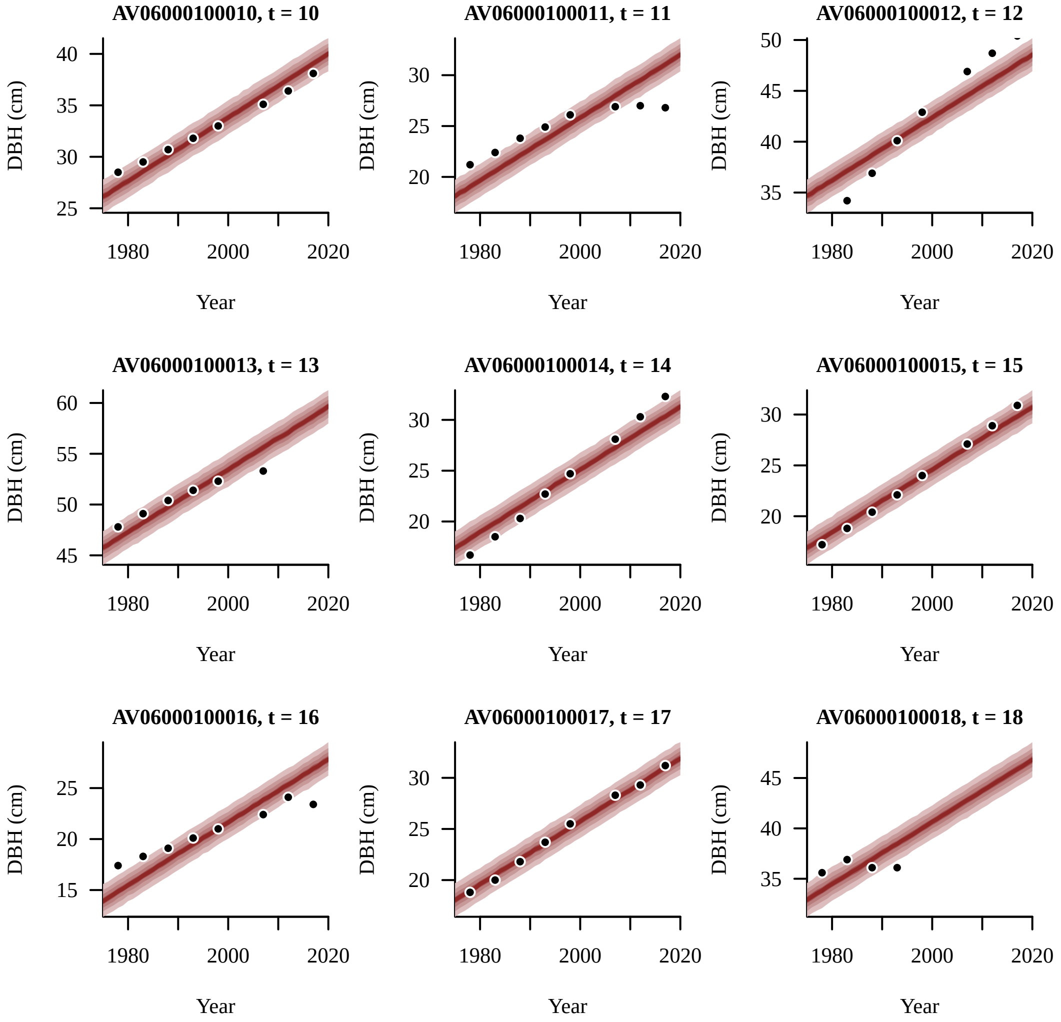
```
samples <- util$extract_expectands(fit)
base_samples <- util$filter_expectands(samples,
                                       c('d0', 'beta', 'sigma'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

Unfortunately posterior retrodictive checks show a poor fit for many of the trees in our data set. This suggests that different trees might be growing at different rates.

```
plot_cont_marginal_quantiles <- function(xs, preds,
                                         display_xlims=NA,
                                         display_ylims=NA,
                                         title="",
                                         x_name="", y_name="") {
  probs = c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
  cred <- sapply(preds, function(pred)
                        quantile(c(t(pred), recursive=TRUE),
                        probs=probs))

  if (is.na(display_xlims)) {
    xlims <- range(xs)
  } else {
    xlims <- display_xlims
  }

  if (is.na(display_ylims)) {
    ylims <- c(min(cred[1,]), max(cred[9,]))
  } else {
    ylims <- display_ylims
  }

  plot(1, type="n", main=title,
       xlim=xlims, xlab=x_name,
       ylim=ylims, ylab=y_name)

  polygon(c(xs, rev(xs)), c(cred[1,], rev(cred[9,])),
          col = c_light, border = NA)
  polygon(c(xs, rev(xs)), c(cred[2,], rev(cred[8,])),
          col = c_light_highlight, border = NA)
  polygon(c(xs, rev(xs)), c(cred[3,], rev(cred[7,])),
          col = c_mid, border = NA)
  polygon(c(xs, rev(xs)), c(cred[4,], rev(cred[6,])),
          col = c_mid_highlight, border = NA)
  lines(xs, cred[5,], col=c_dark, lwd=2)
}

par(mfrow=c(3, 3), mar = c(5, 4, 2, 1))

for (t in 1:data$N_trees) {
  idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
  years <- data$tree_years[idxs]
  dbhs <- data$tree_dbhs[idxs]

  pred_names <- sapply(1:data$N_year_grid,
                       function(n) paste0('dbh_grid_pred[',
                                          t, ',', n, ']'))
  preds <- samples[pred_names]
  plot_cont_marginal_quantiles(data$year_grid, preds,
                               title=paste0(tree_ids[t], ", t = ", t),
                               x_name="Year", y_name="DBH (cm)")
  points(years, dbhs, col="white", pch=16, cex=1.2)
  points(years, dbhs, col="black", pch=16, cex=0.8)
}
```

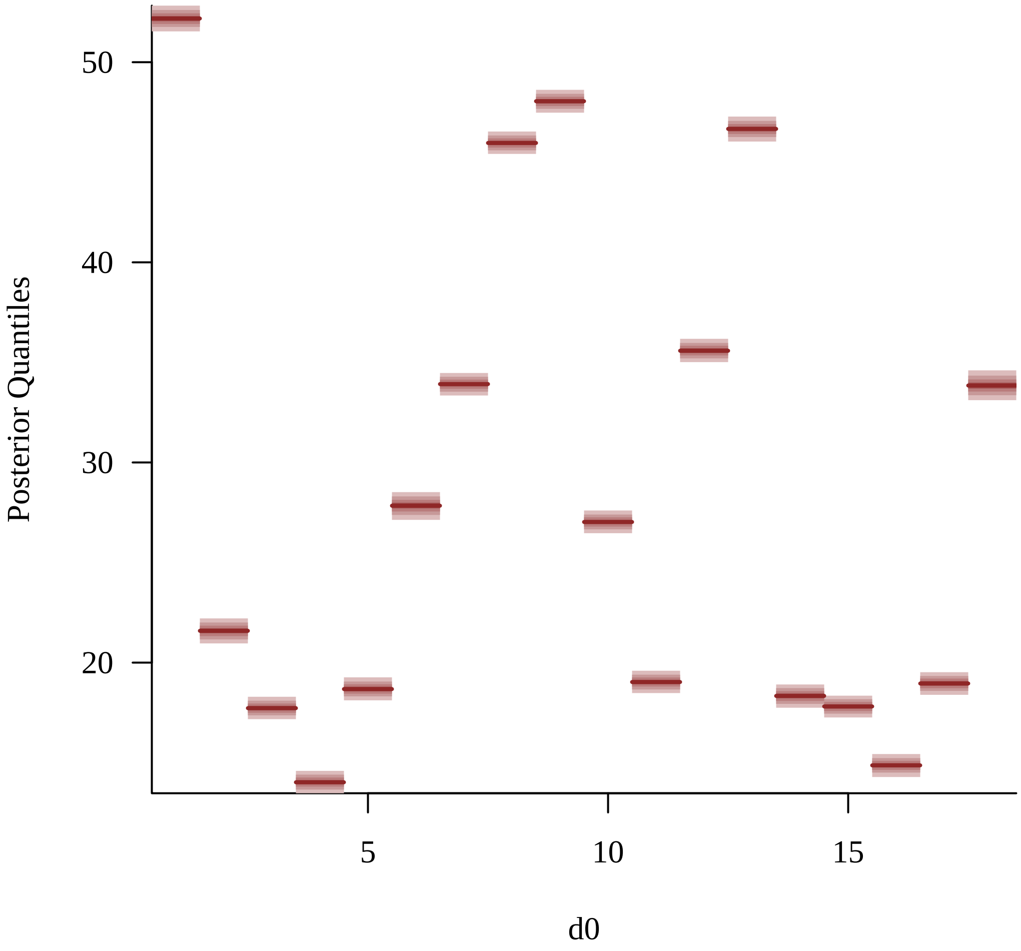

AV06000100001, t = 1
AV06000100002, t = 2
AV06000100003, t = 3
AV06000100004, t = 4
AV06000100005, t = 5
AV06000100006, t = 6
AV06000100007, t = 7
AV06000100008, t = 8
AV06000100009, t = 9
DBH (cm)
Year
1980
2000
2020
50
55
60
65
20
25
30
35
15
40
45



The inadequacy of this first model is also hinted at in the inferred posterior behavior. Marginal posterior distributions for all of the initial diameters appears to be reasonable.

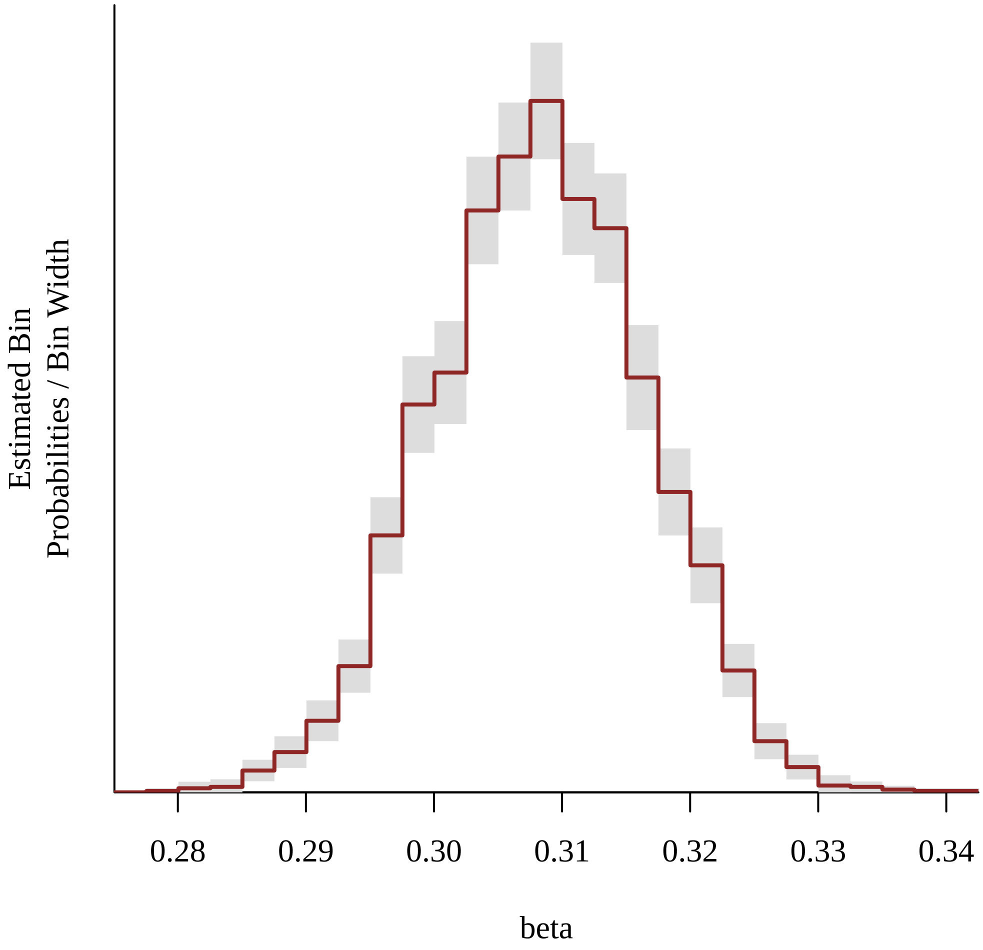
```
plot_disc_marginal_quantiles <- function(samples, names, x_name="") {
  N <- length(names)
  idx <- rep(1:N, each=2)
  xs <- sapply(1:length(idx), function(k)
                              if(k %% 2 == 0) idx[k] + 0.5 else idx[k] - 0.5)

  probs = c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)
  cred <- sapply(1:N, function(n)
                      quantile(c(t(samples[[names[n]]]), recursive=TRUE),
                               probs=probs))
  pad_cred <- do.call(cbind, lapply(idx, function(n) cred[1:9,n]))

  ylims <- c(min(cred[1,]), max(cred[9,]))

  plot(1, type="n",
       xlim=c(0.5, N + 0.5), xlab=x_name,
       ylim=ylims, ylab="Marginal\nPosterior Quantiles")

  polygon(c(xs, rev(xs)), c(pad_cred[1,], rev(pad_cred[9,])),
          col = c_light, border = NA)
  polygon(c(xs, rev(xs)), c(pad_cred[2,], rev(pad_cred[8,])),
          col = c_light_highlight, border = NA)
  polygon(c(xs, rev(xs)), c(pad_cred[3,], rev(pad_cred[7,])),
          col = c_mid, border = NA)
  polygon(c(xs, rev(xs)), c(pad_cred[4,], rev(pad_cred[6,])),
          col = c_mid_highlight, border = NA)
  for (n in 1:N) {
    lines(xs[(2 * n - 1):(2 * n)], pad_cred[5,(2 * n - 1):(2 * n)],
          col=c_dark, lwd=2)
  }
}
```

```
par(mfrow=c(1, 1), mar = c(5, 4, 2, 1))

d0_names <- sapply(1:data$N_trees, function(t) paste0('d0[', t, ']'))
plot_disc_marginal_quantiles(samples, d0_names, "d0")
```



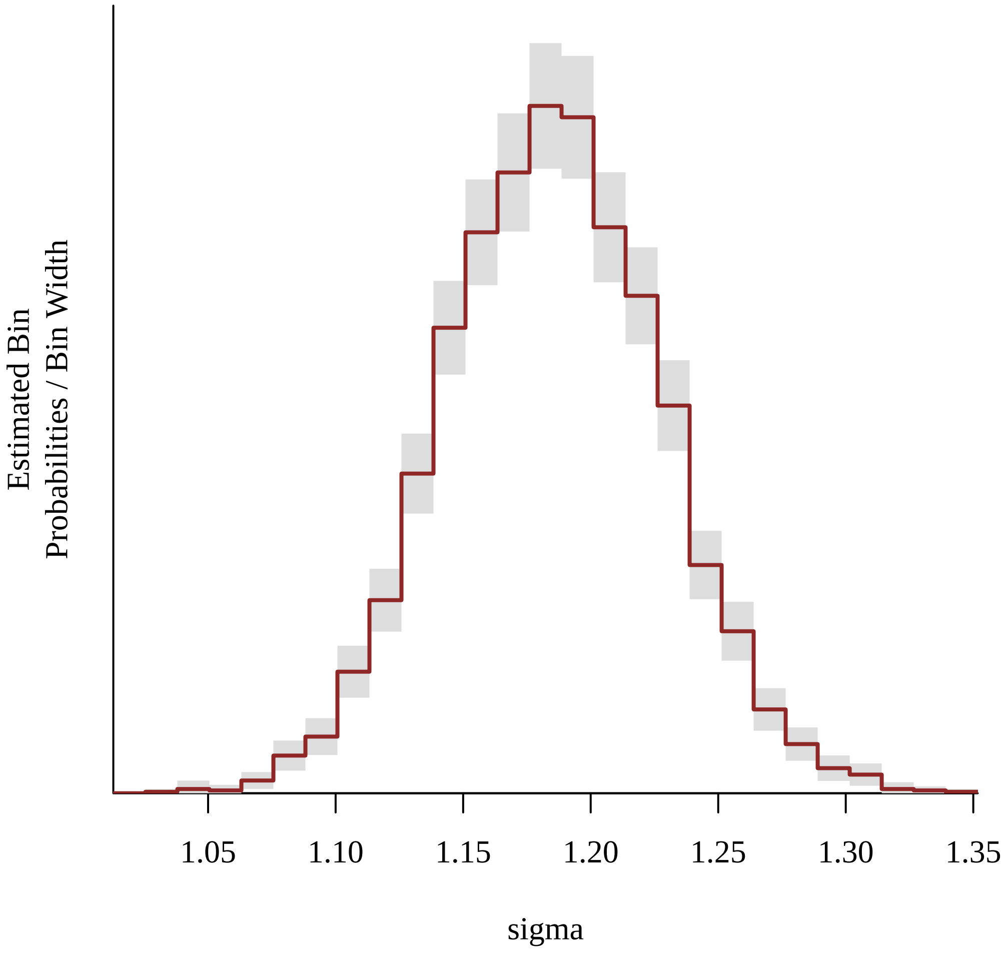
Similarly inferences for the shared growth rate don't appear to be out of order.

```
util$plot_expectand_pushforward(samples[["beta"]], 25,
                                display_name="beta")
```



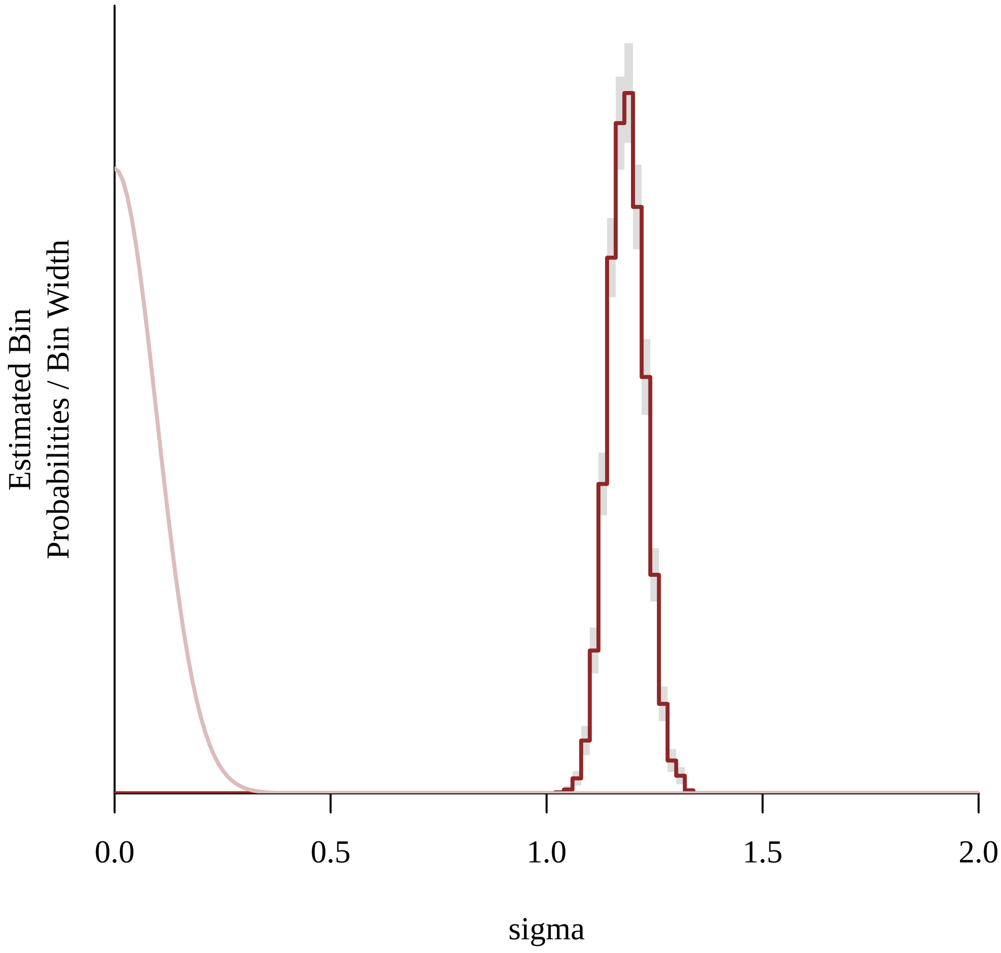
Inferences for the measurement variability, however, concentrate at oddly large values.

```
util$plot_expectand_pushforward(samples[["sigma"]], 25,
                                display_name="sigma")
```



Indeed they concentrate at values far above the soft threshold of 0.25 cm that our prior model tries to enforce. Note the factor of 2 when evaluating the prior probability densities because we're plotting a half-normal probability density function and not a regular normal probability density function.

```
util$plot_expectand_pushforward(samples[["sigma"]], 100,
                                display_name="sigma", flim=c(0, 2))
xs <- seq(0, 2, 0.01)
ys <- 2 * dnorm(xs, 0, 0.25 / 2.57)
lines(xs, ys, lwd=2, col=c_light)
```



What's happening here is that the posterior distribution tries to accommodate the data however it can. Because the growth model isn't flexible enough to accommodate the behavior of each individual tree the measurement model has to compensate by inflating σ so that the measurement variability envelopes as much of the residual deviation as possible.

3.3 Heterogeneous Linear Model

In general there could be many reasons why the posterior retrodictive performance of our initial model was poor. One of the most prominent is the assumption of a common growth rate amongst all of the trees. Let's see if allowing the growth rate to vary from tree to tree results in a better fit.

```
fit <- stan(file="stan_programs/heterogeneous_linear_growth.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

Fortunately there are no signs that our computed inferences might be untrustworthy.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

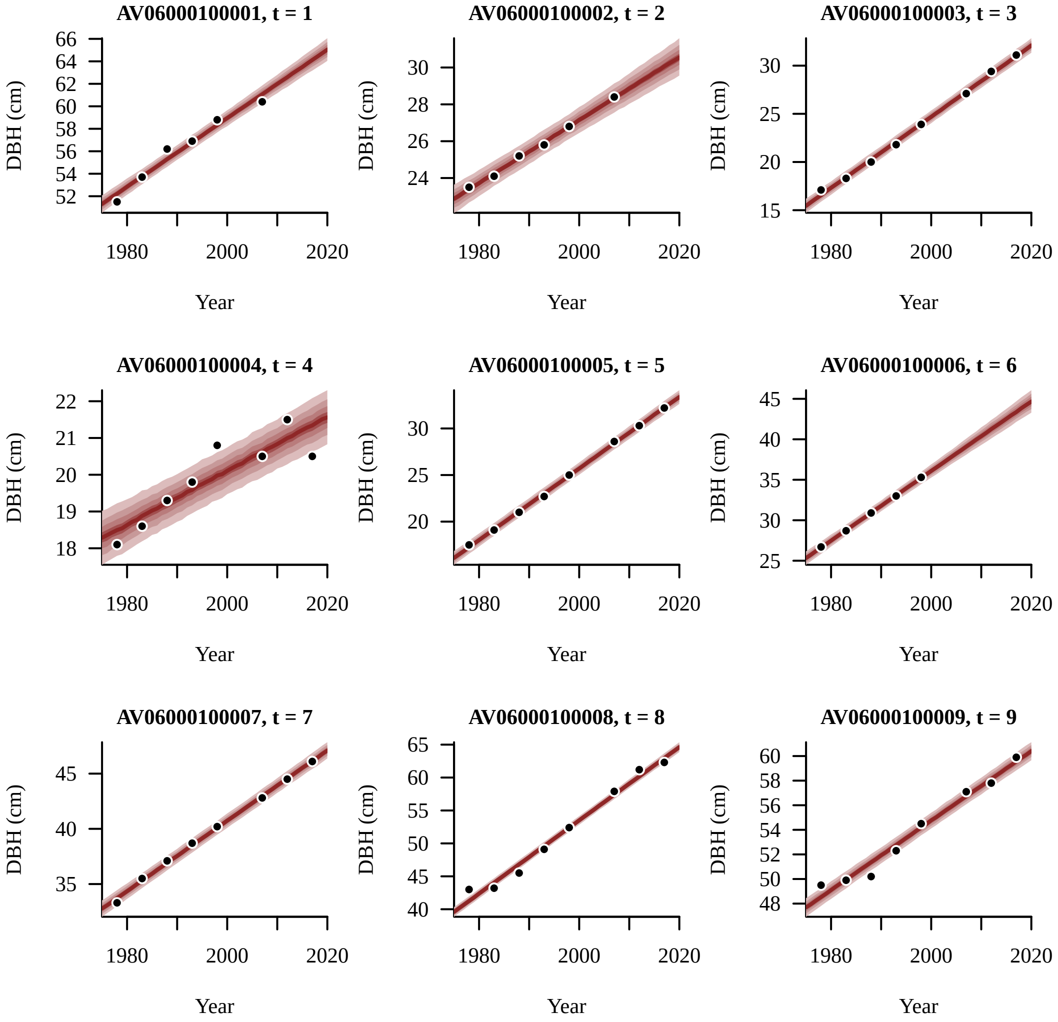
```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

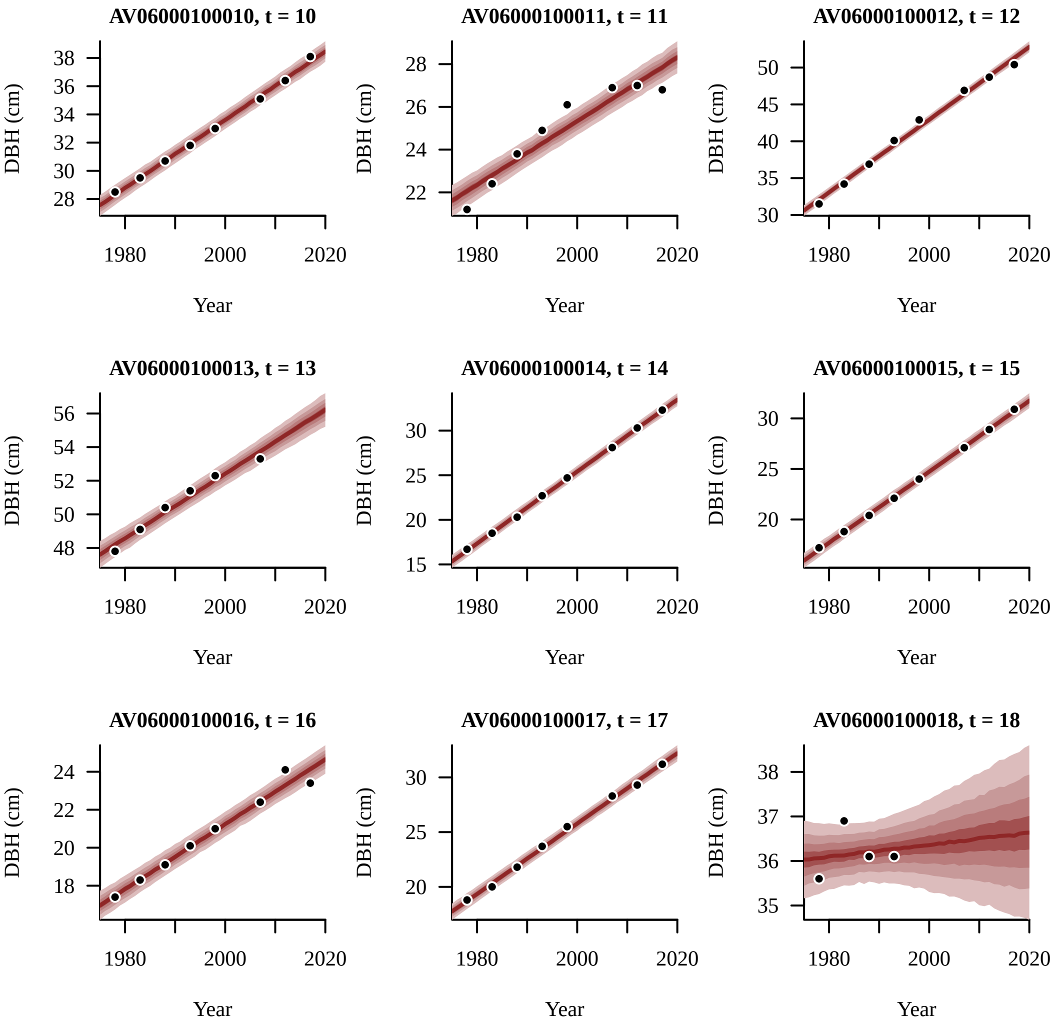
```
samples <- util$extract_expectands(fit)
base_samples <- util$filter_expectands(samples,
                                       c('d0', 'beta', 'sigma'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

Overall the posterior retrodictive performance is much improved.

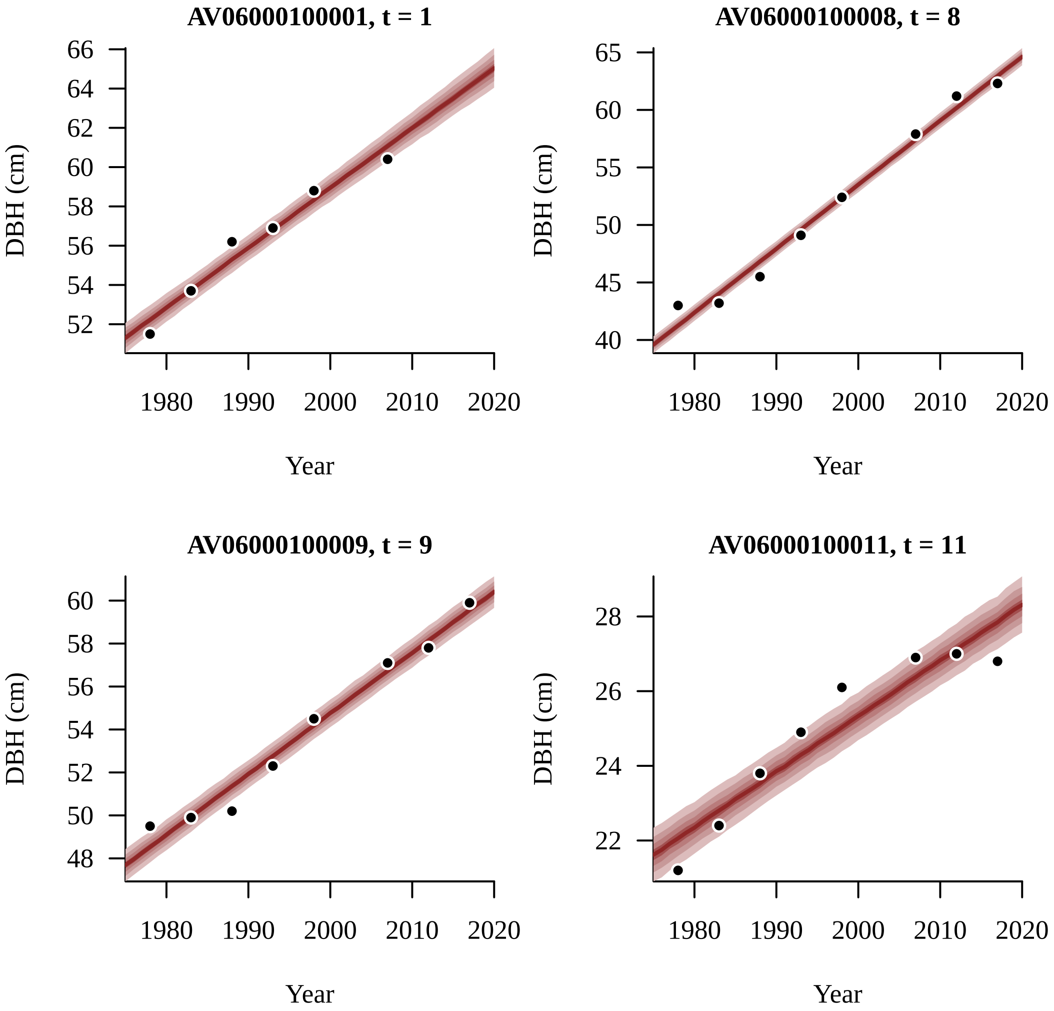
```
par(mfrow=c(3, 3), mar = c(5, 4, 2, 1))

for (t in 1:data$N_trees) {
  idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
  years <- data$tree_years[idxs]
  dbhs <- data$tree_dbhs[idxs]

  pred_names <- sapply(1:data$N_year_grid,
                       function(n) paste0('dbh_grid_pred[',
                                          t, ',', n, ']'))
  preds <- samples[pred_names]
  plot_cont_marginal_quantiles(data$year_grid, preds,
                               title=paste0(tree_ids[t], ", t = ", t),
                               x_name="Year", y_name="DBH (cm)")
  points(years, dbhs, col="white", pch=16, cex=1.2)
  points(years, dbhs, col="black", pch=16, cex=0.8)
}
```


AV06000100001, t = 1
DBH (cm)
66
64
62
60
58
56
54
52
1980
2000
2020
Year
AV06000100002, t = 2
DBH (cm)
30
28
26
24
1980
2000
2020
Year
AV06000100003, t = 3
DBH (cm)
30
25
20
15
1980
2000
2020
Year
AV06000100004, t = 4
DBH (cm)
22
21
20
19
18
1980
2000
2020
Year
AV06000100005, t = 5
DBH (cm)
30
25
20
1980
2000
2020
Year
AV06000100006, t = 6
DBH (cm)
45
40
35
30
25
1980
2000
2020
Year
AV06000100007, t = 7
DBH (cm)
45
40
35
1980
2000
2020
Year
AV06000100008, t = 8
DBH (cm)
65
60
55
50
45
40
1980
2000
2020
Year
AV06000100009, t = 9
DBH (cm)
60
58
56
54
52
50
48
1980
2000
2020
Year



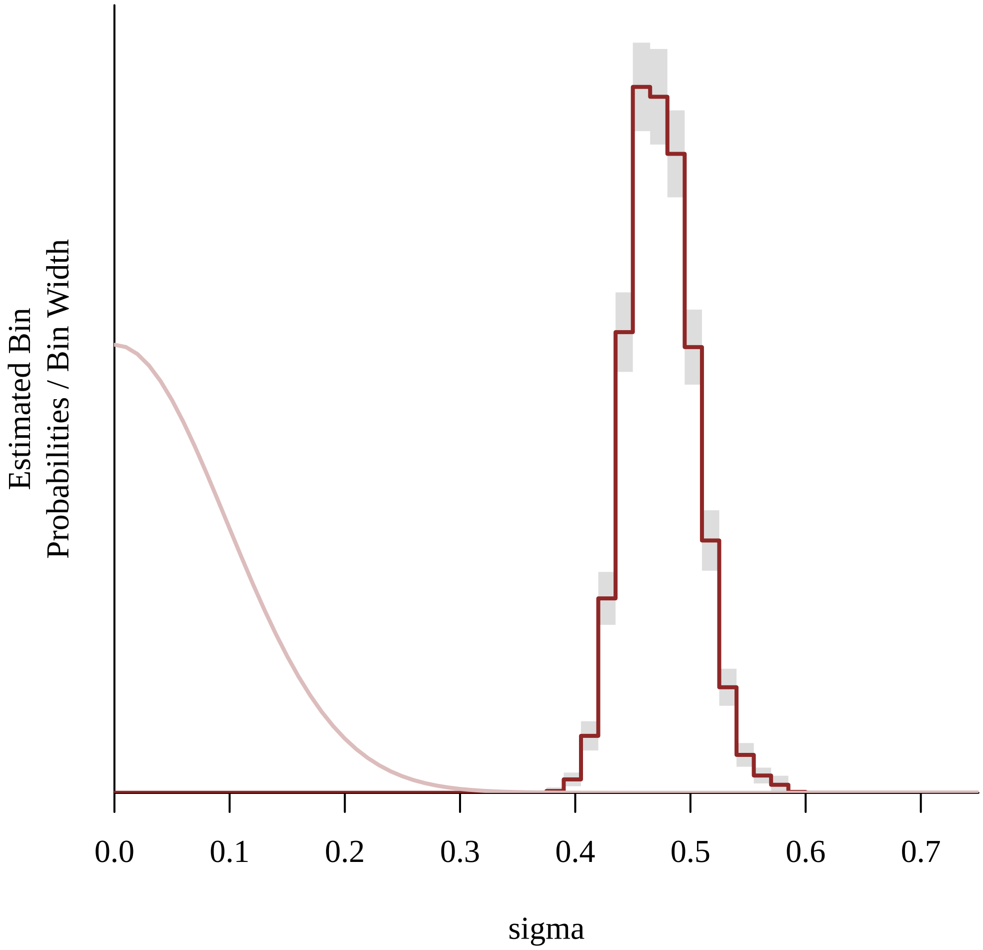
That some trees show signs of of accelerating growth in earlier years and decelerating growth in later years that our model cannot accommodate.

```
par(mfrow=c(2, 2), mar = c(5, 4, 2, 1))

for (t in c(1, 8, 9, 11)) {
  idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
  years <- data$tree_years[idxs]
  dbhs <- data$tree_dbhs[idxs]

  pred_names <- sapply(1:data$N_year_grid,
                       function(n) paste0('dbh_grid_pred[',
                                          t, ',', n, ']'))
  preds <- samples[pred_names]
  plot_cont_marginal_quantiles(data$year_grid, preds,
                               title=paste0(tree_ids[t], ", t = ", t),
                               x_name="Year", y_name="DBH (cm)")
  points(years, dbhs, col="white", pch=16, cex=1.2)
  points(years, dbhs, col="black", pch=16, cex=0.8)
}
```



The inadequacy of this heterogeneous linear growth model is also hinted at by the continued concentration of the marginal posterior distribution for σ at values above the soft threshold encoded in our prior model. That said the deviation is much weaker than it was before, suggesting that the inadequacy has been reduced.

```
par(mfrow=c(1, 1), mar = c(5, 4, 2, 1))

util$plot_expectand_pushforward(samples[["sigma"]], 50,
                                display_name="sigma", flim=c(0, 0.75))
xs <- seq(0, 2, 0.01)
ys <- 2 * dnorm(xs, 0, 0.25 / 2.57)
lines(xs, ys, lwd=2, col=c_light)
```



3.4 Heterogeneous Gompertz Model

A useful check for model adequacy doesn't just indicate that there might be a problem but also informs us of what the problem might be. The previous posterior retrodictive check suggested that we need a more flexible growth model that can accommodate early accelerated growth and late decelerated growth. Fortunately this is exactly what a Gompertz growth model can provide, especially with an offset initial diameter.

```
fit <- stan(file="stan_programs/heterogeneous_gompertz_growth.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

Unfortunately the diagnostics seem to be extremely upset indicating that our posterior computation is not to be trusted.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

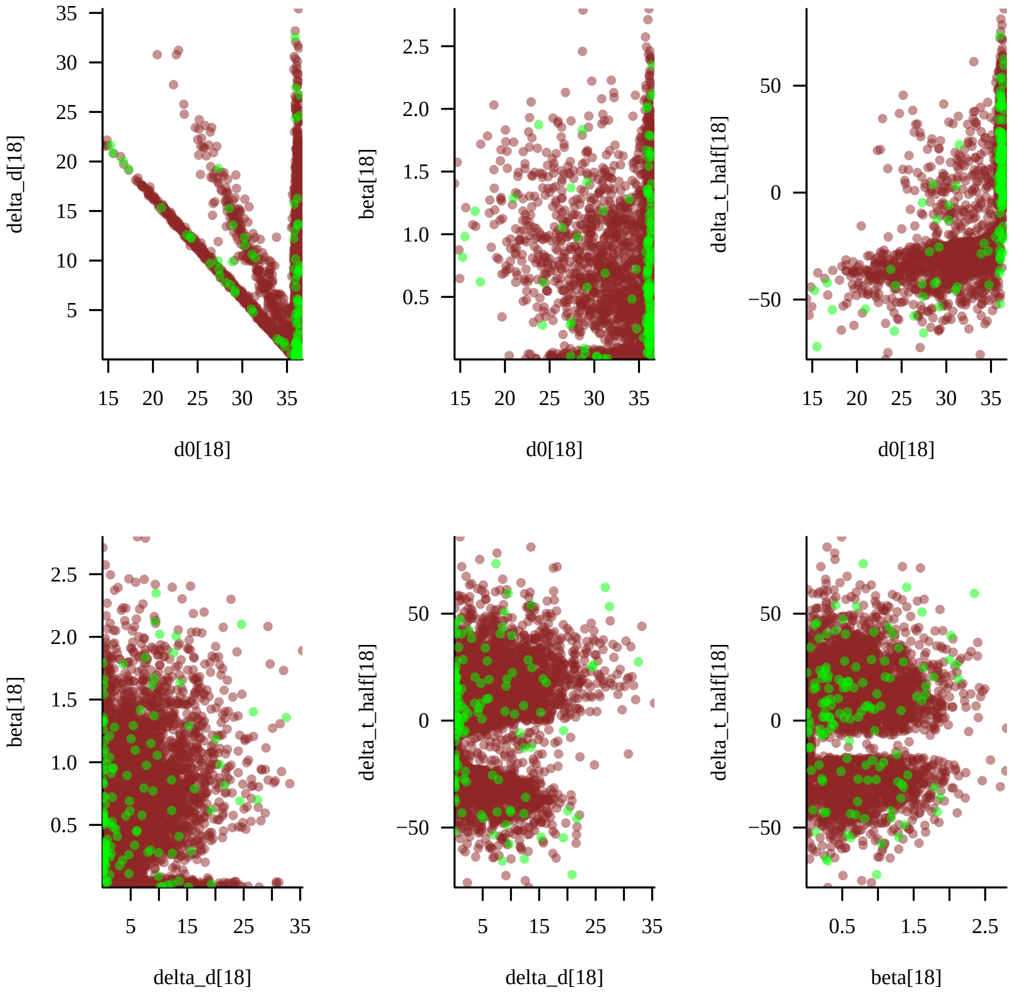
```
  Chain 1: 45 of 1024 transitions (4.4%) diverged.

  Chain 2: 31 of 1024 transitions (3.0%) diverged.
  Chain 2: 20 of 1024 transitions (1.953125%) saturated the maximum treedepth of 10.

  Chain 3: 34 of 1024 transitions (3.3%) diverged.

  Chain 4: 26 of 1024 transitions (2.5%) diverged.
  Chain 4: 24 of 1024 transitions (2.34375%) saturated the maximum treedepth of 10.

  Divergent Hamiltonian transitions result from unstable numerical
trajectories.  These instabilities are often due to degenerate target
geometry, especially "pinches".  If there are only a small number of
divergences then running with adept_delta larger than 0.801 may reduce
the instabilities at the cost of more expensive Hamiltonian
transitions.

  Numerical trajectories that saturate the maximum treedepth have
terminated prematurely.  Increasing max_depth above 10 should result in
more expensive, but more efficient, Hamiltonian transitions.
```

```
samples <- util$extract_expectands(fit)
base_samples <- util$filter_expectands(samples,
                                       c('d0', 'delta_d', 'beta',
                                         'delta_t_half', 'sigma'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
d0[18]:
  Chain 3: Left tail hat{xi} (1.610) exceeds 0.25!
  Chain 4: Left tail hat{xi} (1.670) exceeds 0.25!
  Chain 1: hat{ESS} (47.442) is smaller than desired (100)!
  Chain 2: hat{ESS} (58.902) is smaller than desired (100)!
  Chain 3: hat{ESS} (37.022) is smaller than desired (100)!
  Chain 4: hat{ESS} (60.659) is smaller than desired (100)!

delta_t_half[18]:
  Chain 1: hat{ESS} (24.466) is smaller than desired (100)!
  Chain 2: hat{ESS} (38.304) is smaller than desired (100)!
  Chain 3: hat{ESS} (32.900) is smaller than desired (100)!
  Chain 4: hat{ESS} (31.660) is smaller than desired (100)!


Large tail hat{xi}s suggest that the expectand might not be
sufficiently integrable.

Small empirical effective sample sizes indicate strong empirical
autocorrelations in the realized Markov chains. If the empirical
effective sample size is too small then Markov chain Monte Carlo
estimation may be unreliable even when a central limit theorem holds.
```

Notice that the expectand diagnostic failures manifest for only the parameters of the 18th tree in the data set. This suggests that those parameters might be the most probematic. Indeed the pairs plots of the Markov chain Monte Carlo output for these parameters show some wild posterior geometries.

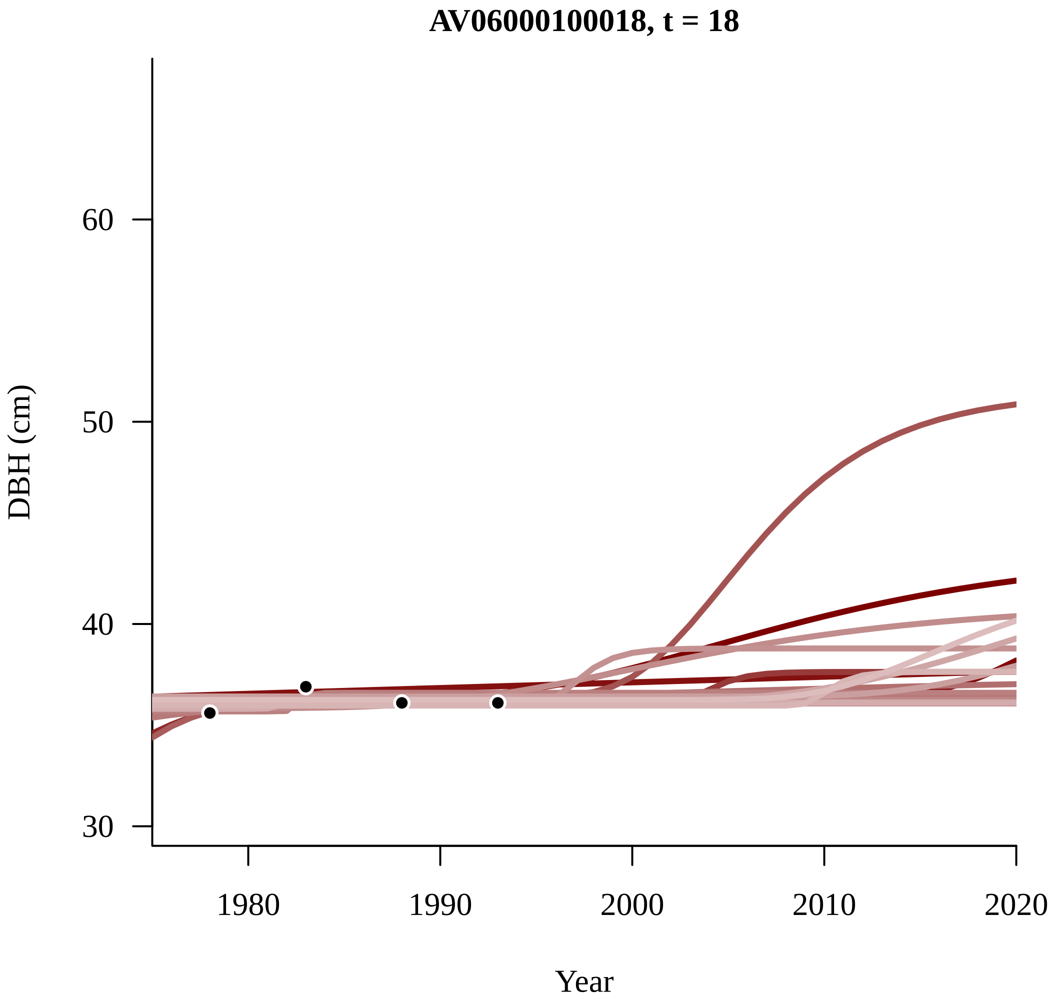
```r
names <- c('d0[18]', 'delta_d[18]', 'beta[18]', 'delta_t_half[18]')
util$plot_div_pairs(names, names, samples, diagnostics)
```



Why would this tree be so problematic? Well let's contrast the data to some posterior samples of the growth curve.

```
plot_realizations <- function(xs, fs, N=50,
                              x_name="", display_xlims=NA,
                              y_name="", display_ylims=NA,
                              title="") {
  I <- dim(fs)[2]
  J <- min(N, I)

  plot_idx <- sapply(1:J, function(j) (I %/% J) * (j - 1) + 1)

  nom_colors <- c("#DCBCBC", "#C79999", "#B97C7C",
                  "#A25050", "#8F2727", "#7C0000")
  line_colors <- colormap(colormap=nom_colors, nshades=J)

  if (is.na(display_xlims)) {
    xlims <- range(xs)
  } else {
    xlims <- display_xlims
  }

  if (is.na(display_ylims)) {
    ylims <- range(fs)
  } else {
    ylims <- display_ylims
  }

  plot(1, type="n", main=title,
       xlab=x_name, xlim=xlims,
       ylab=y_name, ylim=ylims)
  for (j in 1:J) {
    r_fs <- fs[, plot_idx[j]]
    lines(xs, r_fs, col=line_colors[j], lwd=3)
  }
}
```

```
par(mfrow=c(1, 1), mar = c(5, 4, 2, 1))

t <- 18
idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
years <- data$tree_years[idxs]
dbhs <- data$tree_dbhs[idxs]

fs <- t(sapply(1:data$N_year_grid,
               function(n) c(t(samples[[paste0('dbh_grid[',
                                               t, ',', n, ']')]]),
                             recursive=TRUE)))

plot_realizations(data$year_grid, fs, 25,
                  x_name="Year", y_name="DBH (cm)",
                  title=paste0(tree_ids[t], ", t = ", t),)
points(years, dbhs, col="white", pch=16, cex=1.2)
points(years, dbhs, col="black", pch=16, cex=0.8)
```



The measured diameters for this particular tree are consistent with no growth. An offset Gomertz curve can accommodate this flat behavior in multiple ways; for example we can shift the growth curve to the left so that the data fall into the upper asymptote or shift it to the right so that the data fall into the lower asymptote, with d_0 and δ_d compensating to give the appropriate diameters. All of these different possibilities result in complex posterior uncertainties that are computationally difficult for `Stan` to quantify.

Without more observations for this particular tree the only way to improve the posterior behavior would be to incorporate more information into our prior model, for example more carefully constraining the reasonable locations of the growth curves. For the moment let's just remove this tree entirely.

```r
active_tree_ids <- head(good_tree_ids, 17)

good_tree_ids <- c()
tree_N_obs <- c()
tree_start_idxs <- c()
tree_end_idxs <- c()
tree_years <- c()
tree_dbhs <- c()
tree_vigors <- c()
tree_species <- c()
tree_statuses <- c()
tree_notes <- c()

idx <- 1

for (id in active_tree_ids) {
  # Isolate tree observations
  tree_data <- stand_tree_data[stand_tree_data['TREEID'] == id,]

  years <- tree_data[,'YEAR']
  dbhs <- tree_data[,'DBH']
  vigors <- tree_data[,'TREE_VIGOR']
  species <- tree_data[,'SPECIES']
  statuses <- tree_data[,'TREE_STATUS']
  notes <- tree_data[,'CHECK_NOTES']

  # Remove post-mortality observations
  good_obs <- which(vigors != "M")
  years <- years[good_obs]
  dbhs <- dbhs[good_obs]
  vigors <- vigors[good_obs]
  species <- species[good_obs]
  statuses <- statuses[good_obs]
  notes <- notes[good_obs]

  # Drop tree if there are not at least four observations
  N_obs <- length(years)
  if (N_obs < 4) next

  # Append tree data to concatenated arrays
  good_tree_ids <- c(good_tree_ids, id)
```

```
    tree_N_obs <- c(tree_N_obs, N_obs)

    start_idx <- idx
    end_idx <- idx + N_obs - 1
    idx <- idx + N_obs
    tree_start_idxs <- c(tree_start_idxs, start_idx)
    tree_end_idxs <- c(tree_end_idxs, end_idx)

    tree_years <- c(tree_years, years)
    tree_dbhs <- c(tree_dbhs, dbhs)
    tree_vigors <- c(tree_vigors, vigors)
    tree_species <- c(tree_species, species)
    tree_statuses <- c(tree_statuses, statuses)
    tree_notes <- c(tree_notes, notes)
  }

  N <- length(tree_years)
  N_trees <- length(good_tree_ids)

  data <- mget(c("N", "N_trees", "tree_N_obs",
                 "tree_start_idxs", "tree_end_idxs",
                 "tree_years", "tree_dbhs"))

  year_grid <- seq(1975, 2020, 1)
  N_year_grid <- length(year_grid)

  data$N_year_grid <- N_year_grid
  data$year_grid <- year_grid
```

Will the fit be any better?

```
fit <- stan(file="stan_programs/heterogeneous_gompertz_growth.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

The diagnostics are still grumbling a bit.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  Chain 1: 5 of 1024 transitions (0.48828125%) saturated the maximum treedepth of 10.
  Chain 2: 2 of 1024 transitions (0.1953125%) saturated the maximum treedepth of 10.

  Chain 3: 13 of 1024 transitions (1.26953125%) saturated the maximum treedepth of 10.

  Chain 4: 1 of 1024 transitions (0.09765625%) saturated the maximum treedepth of 10.

  Numerical trajectories that saturate the maximum treedepth have
terminated prematurely.  Increasing max_depth above 10 should result in
more expensive, but more efficient, Hamiltonian transitions.
```

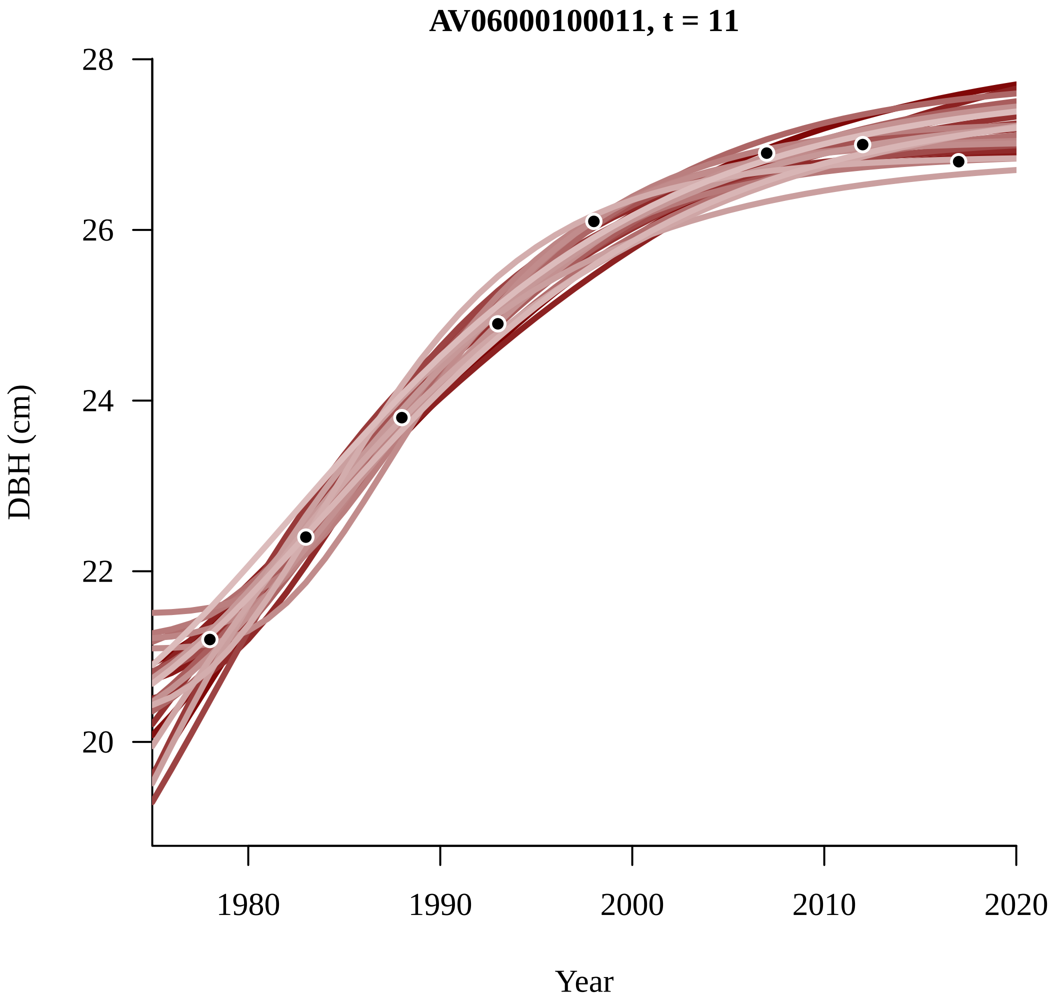
```r
samples <- util$extract_expectands(fit)
base_samples <- util$filter_expectands(samples,
                                       c('d0', 'delta_d', 'beta',
                                         'delta_t_half', 'sigma'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
d0[11]:
  Chain 2: Left tail hat{xi} (0.251) exceeds 0.25!
  Chain 2: hat{ESS} (62.145) is smaller than desired (100)!

delta_d[11]:
  Chain 2: Right tail hat{xi} (0.256) exceeds 0.25!
  Chain 2: hat{ESS} (58.650) is smaller than desired (100)!

beta[11]:
  Chain 2: Right tail hat{xi} (0.329) exceeds 0.25!
  Chain 2: hat{ESS} (79.760) is smaller than desired (100)!

delta_t_half[11]:
  Chain 2: hat{ESS} (74.431) is smaller than desired (100)!


Large tail hat{xi}s suggest that the expectand might not be
sufficiently integrable.

Small empirical effective sample sizes indicate strong empirical
autocorrelations in the realized Markov chains. If the empirical
effective sample size is too small then Markov chain Monte Carlo
estimation may be unreliable even when a central limit theorem holds.
```

The saturating tree depth and low emprical effective sample size warnings are consistent with strong posterior uncertainties. Fortunately these warnings indicate *inefficient* posterior computation but not *inaccurate* posterior computation. Similarly the $\hat{\xi}$ warnings indicate that the marginal posterior distribution for the 11th tree is heavy-tailed. Heavy-tailed posterior distributions require `Stan` to work harder but don't necessarily obstruct accurate computation.

The posterior realizations of the growth curve for the 11th tree don't seem to exhibit any particularly extreme behaviors.

```
par(mfrow=c(1, 1), mar = c(5, 4, 2, 1))

t <- 11
idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
years <- data$tree_years[idxs]
dbhs <- data$tree_dbhs[idxs]

fs <- t(sapply(1:data$N_year_grid,
               function(n) c(t(samples[[paste0('dbh_grid[',
                                               t, ',', n, ']')]]),
                             recursive=TRUE)))

plot_realizations(data$year_grid, fs, 25,
                  x_name="Year", y_name="DBH (cm)",
                  title=paste0(tree_ids[t], ", t = ", t),)
points(years, dbhs, col="white", pch=16, cex=1.2)
points(years, dbhs, col="black", pch=16, cex=0.8)
```

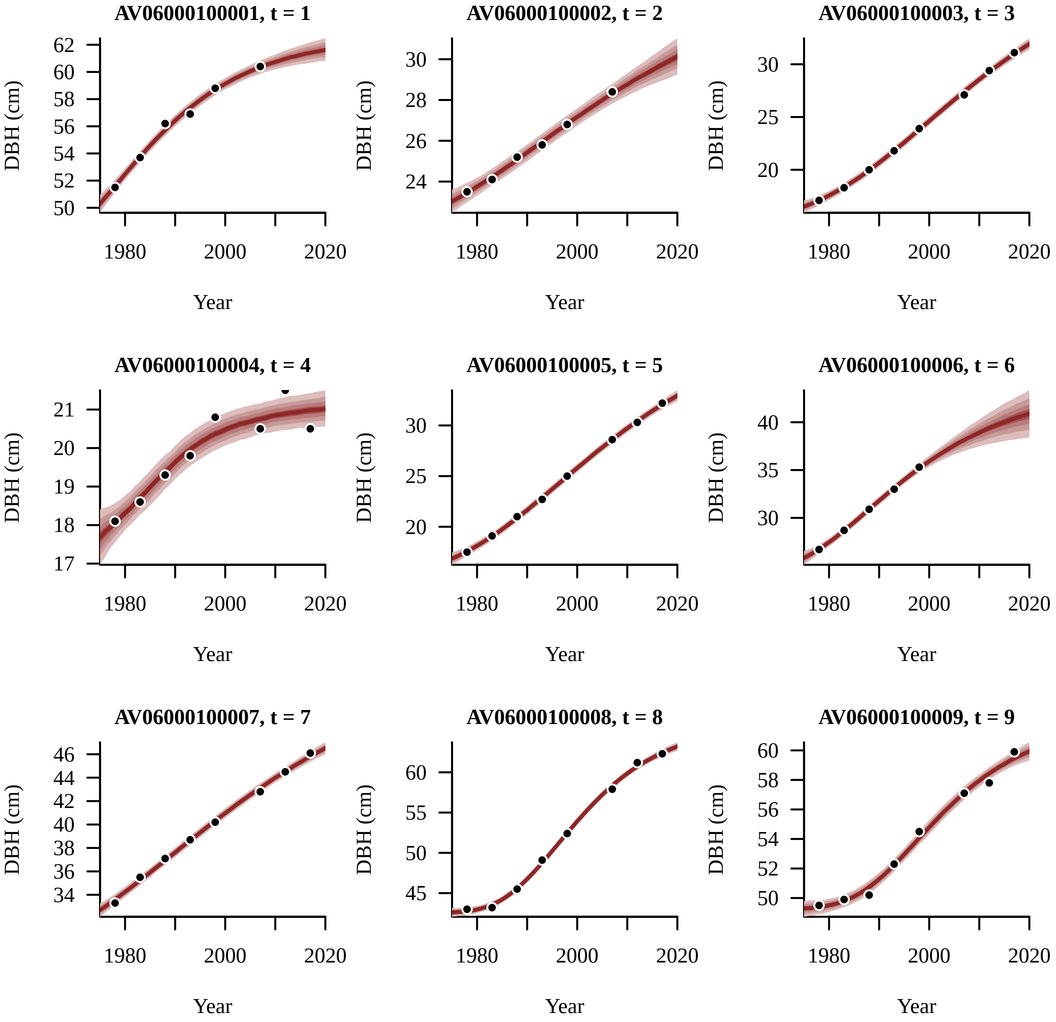


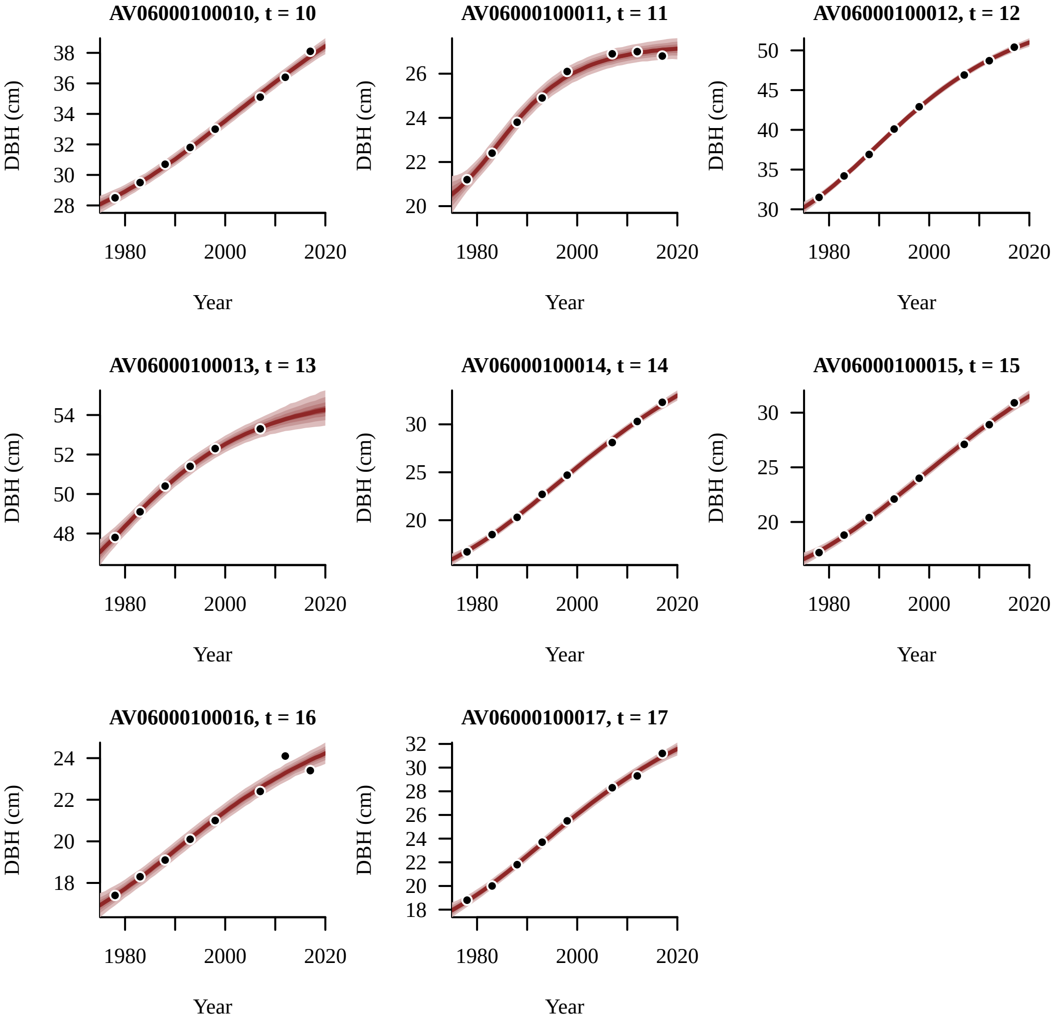
We have to be careful, however, because heavy tails result in more extreme but also more rare behaviors. This means that we often have to plot many realizations before we see the tail behaviors, and here we're plotting only 25 realizations.

Any ways, all of this discussion is to say that while our posterior distribution exhibits some awkward behaviors `Stan` should be giving us a faithful picture of those behaviors and we can continue on with our analysis.

The posterior retrodictive performance looks outright incredible compared to what have seen with the first two models.

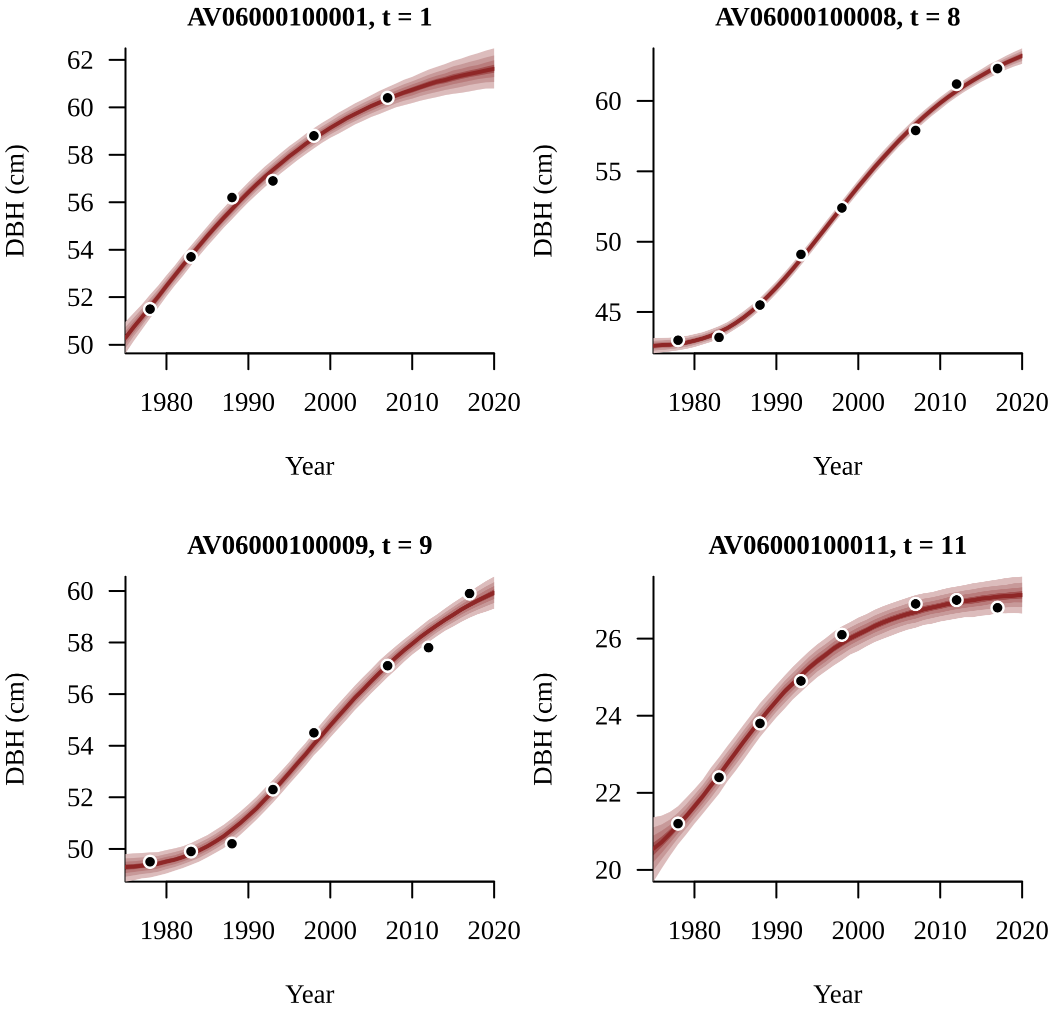
```
par(mfrow=c(3, 3), mar = c(5, 4, 2, 1))

for (t in 1:data$N_trees) {
  idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
  years <- data$tree_years[idxs]
  dbhs <- data$tree_dbhs[idxs]

  pred_names <- sapply(1:data$N_year_grid,
                       function(n) paste0('dbh_grid_pred[',
                                          t, ',', n, ']'))
  preds <- samples[pred_names]
  plot_cont_marginal_quantiles(data$year_grid, preds,
                               title=paste0(tree_ids[t], ", t = ", t),
                               x_name="Year", y_name="DBH (cm)")
  points(years, dbhs, col="white", pch=16, cex=1.2)
  points(years, dbhs, col="black", pch=16, cex=0.8)
}
```


AV06000100001, t = 1
AV06000100002, t = 2
AV06000100003, t = 3
AV06000100004, t = 4
AV06000100005, t = 5
AV06000100006, t = 6
AV06000100007, t = 7
AV06000100008, t = 8
AV06000100009, t = 9
DBH (cm)
Year
1980
2000
2020
62
60
58
56
54
52
50
30
28
26
24
30
25
20
21
20
19
18
17
30
25
20
40
35
30
46
44
42
40
38
36
34
60
55
50
45
60
58
56
54
52
50



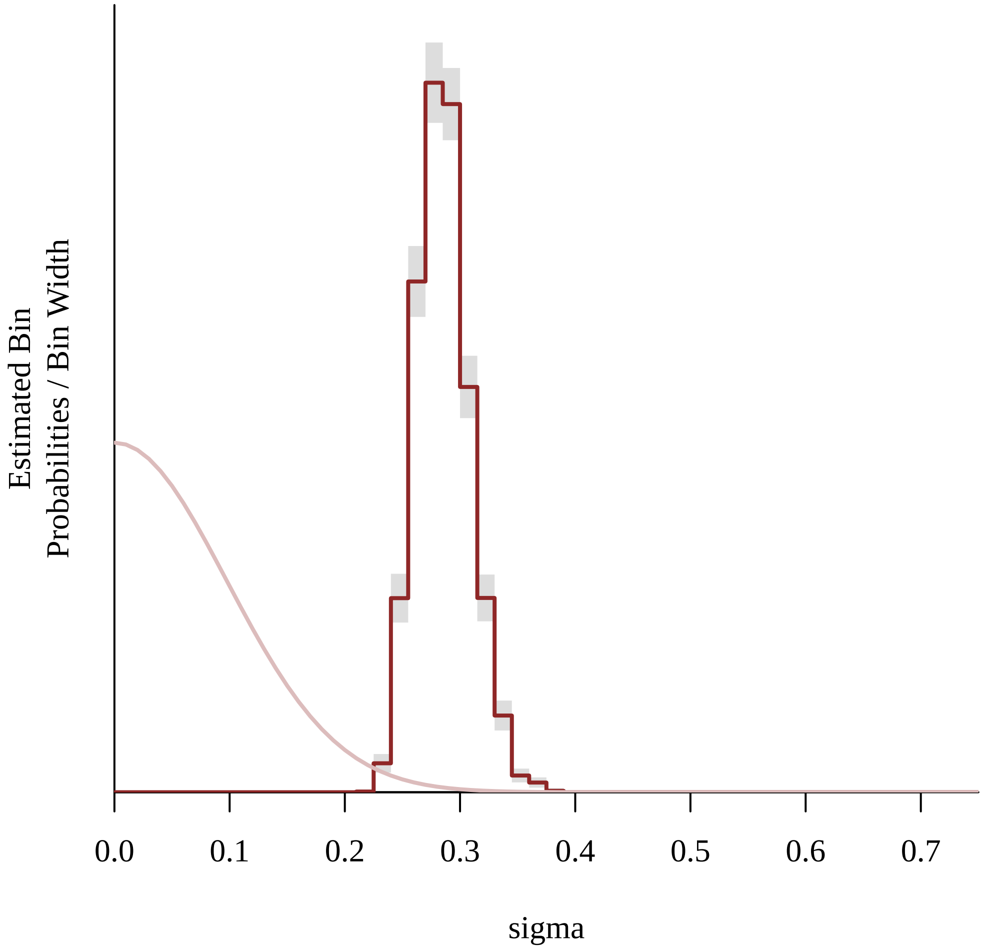
The offset Gompertz growth model is able to accommodate the late-stage saturation seen in many of the trees and is partially able to accommodate the early-stage turn on seen in other trees.

```
par(mfrow=c(2, 2), mar = c(5, 4, 2, 1))

for (t in c(1, 8, 9, 11)) {
  idxs <- data$tree_start_idxs[t]:data$tree_end_idxs[t]
  years <- data$tree_years[idxs]
  dbhs <- data$tree_dbhs[idxs]

  pred_names <- sapply(1:data$N_year_grid,
                       function(n) paste0('dbh_grid_pred[',
                                          t, ',', n, ']'))
  preds <- samples[pred_names]
  plot_cont_marginal_quantiles(data$year_grid, preds,
                               title=paste0(tree_ids[t], ", t = ", t),
                               x_name="Year", y_name="DBH (cm)")
  points(years, dbhs, col="white", pch=16, cex=1.2)
  points(years, dbhs, col="black", pch=16, cex=0.8)
}
```



Posterior inferences for the measurement variability are now finally corralled below the 0.25 cm threshold encoded in our prior model.

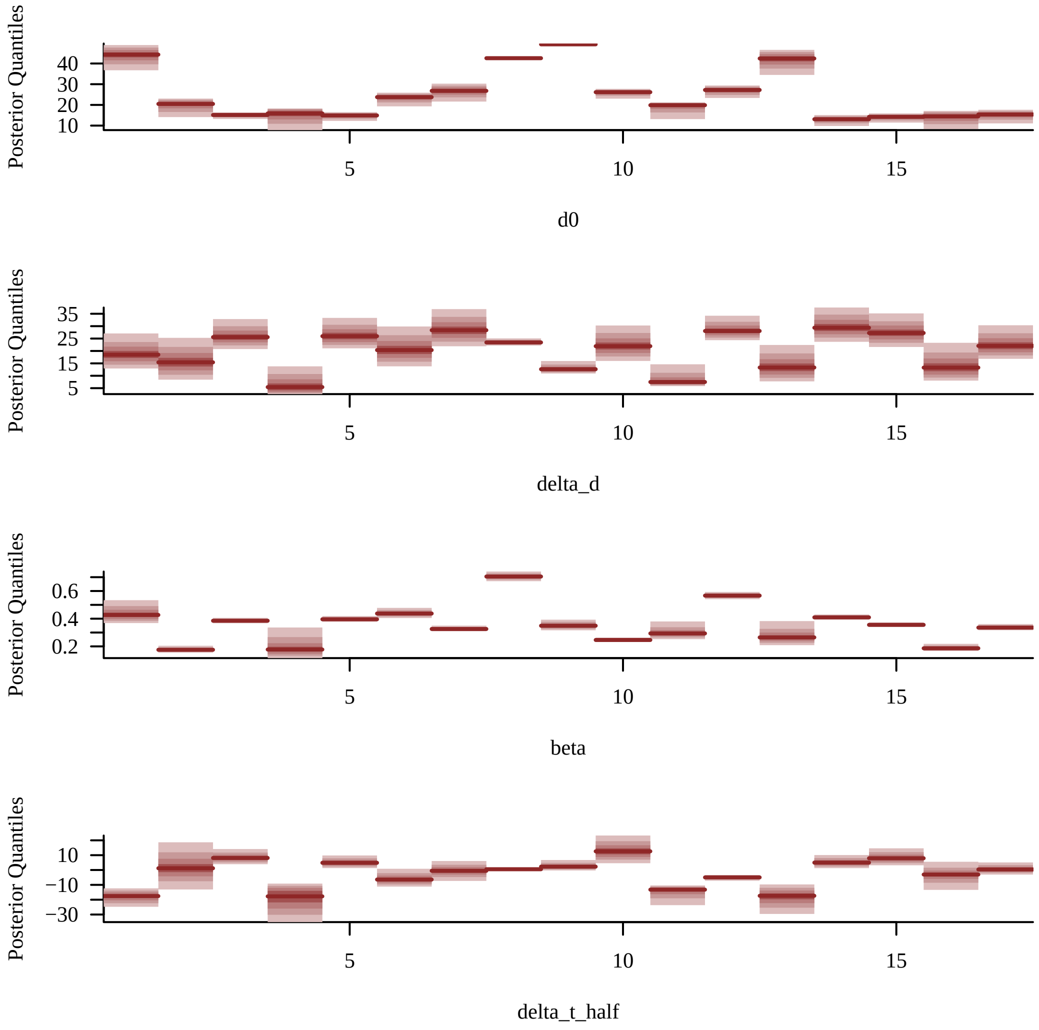
```
par(mfrow=c(1, 1), mar = c(5, 4, 2, 1))
util$plot_expectand_pushforward(samples[["sigma"]], 50,
                                display_name="sigma", flim=c(0, 0.75))
xs <- seq(0, 2, 0.01)
ys <- 2 * dnorm(xs, 0, 0.25 / 2.57)
lines(xs, ys, lwd=2, col=c_light)
```



Confident in the adequacy of our model we can study the posterior inferences for each individual tree. Any patterns in these inferences that correlate with tree health or environental conditions would suggest additional features that we could incorporate into the model.

```
par(mfrow=c(4, 1), mar = c(5, 4, 2, 1))
```

```r
names <- sapply(1:data$N_trees,
                function(t) paste0('d0[', t, ']'))
plot_disc_marginal_quantiles(samples, names, "d0")

names <- sapply(1:data$N_trees,
                function(t) paste0('delta_d[', t, ']'))
plot_disc_marginal_quantiles(samples, names, "delta_d")

names <- sapply(1:data$N_trees,
                function(t) paste0('beta[', t, ']'))
plot_disc_marginal_quantiles(samples, names, "beta")

names <- sapply(1:data$N_trees,
                function(t) paste0('delta_t_half[', t, ']'))
plot_disc_marginal_quantiles(samples, names, "delta_t_half")
```



4 Next Steps

By following the feedback from posterior retrodictive checks and leveraging our domain expertise we have been able to iteratively increase the sophistication of our model, building up to a reasonably accurate approximation of the true data generating process.

That said there are some lingerin issues with our last model and many possible extensions that could bring us closer to answering some of our scientific questions. In other words there are still many more steps that we could take to improve the analysis.

One immediate next step would be to follow up on the sampler diagnostics. We could for example increase the maximum tree depth in `Stan` to allow for more expensive, but more effective, Markov chain Monte Carlo computation. At the same time we could work on incorporating our domain expertise into a more informative prior model that might reduce the posterior uncertainties, and hence the computational difficulty in exploring those uncertainties.

We could also investigate alternative sigmoidal growth models that might offer a better fit to the data. For instance we might look at logistic models or models where the linear time argument is replaced by a higher-order polynomial to see if the added flexibility allows for an improved fit to the data. This can be especially important when we consider data from more trees, both in our initial tree stand and in other tree stands.

Once we trust the adequacy of our model and the faithfulness of our posterior computation we could then investigate any patterns in the tree behaviors, potentially correlating them with auxiliary tree status observations to suggest growth models sensitive to tree physiology and environmental factors such as tree species, tree health, and sunlight exposure. Any clustering of tree growth behavior by these categories could be incorporated into a hierarchical model that would model not only the individual trees but also the forest in which the trees belong.

Finally expanding our analysis with data from other tree stands at other locations around Mount Rainier would allow us to start considering the influence of different climates on tree growth. This is not only of general scientific interest but also might be useful for predicting growth behavior under an evolving climate, if not helping to infer that evolution itself.

5 Workshop Presentation

This exercise was originally developed for a two-day, interactive workshop held at the University of British Columbia in January 2024. The goal of the workshop was not to teach the audience of ecology undergraduate students, graduate students, and postdoctoral researchers any particular technique but rather to expose them to the possibilities of (narratively) generative modeling and Bayesian inference, stimulating their curiosity and perhaps motivating them to learn more in the future. In particular the audience was expected to be familiar with basic ecology and use of the `R` language but *not* to have any prior experience with the statistical analysis steps shown in Section 3.

The workshop itself was structured into three parts.

On the first half of the first day we introduced the provenance of the data discused in Section 1.1 before allowing the students to explore and visualize the data on their own. Throughout we encouraged the students to consider their understanding of the data provenance, and the conceptual data generating processes it suggested, to motivate the exploration. Students were free to use the `R` data manipulation tools with which they were most comfortable.

For the second half of the first day the students brainstormed and researched potential mathematical models for tree growth and diameter measurements. Interactively we also encouraged students to also consider reasonable values for the parameters in the models they identified, setting them up for principled prior modeling.

Finally at the beginning of the second day we gave the students the code in Section 1.2 and Section 1.3, in case they had any troubles in programming their own data exploration, and then the code in Section 3.1 and Section 3.2 to serve as a basic template for implementing a Bayesian analyses. We then directed the students to take the analysis in whichever direction they wanted.

At the end of the second day we gathered back together. We began our conclusion with the students sharing brief, improvised presentations of what they attempted and what they were able to achieve in such a short time. The teaching staff also took this opportunity to discuss the general challenges and opportunities that the student anaylyses demonstrated. Afterwards we reviewed my analysis in Section 3.2 and Section 3.3.

The room in which the workshop was held was organized into small, round ables that implicitly encouraged the students to work together in small groups. This facilitated the sharing of ecological, statistical, and programming expertise amongst the students, which was particularly productive given their diverse backgrounds.

Because of the interactive and open nature of the curriculum the workshop strongly benefited from a large and dedicated teaching staff. Myself, Elizabeth Wolkovich, Will Pearse, Harold Eyster, Andrew MacDonald, and Vianey Leos Barajas patrolled the room, answering individual questions about ecological modeling, Bayesian inference, Markov chain Monte Carlo, `R`, `Stan`, and more so that students were never left to struggle too long and become frustrated no matter what their prior experience may have been.

Acknowledgements

I thank Elizabeth Wolkovich for obtaining and researching the provenance of the data and sharing her extensive domain expertise, not to mention organizing the workshop that initially stimulated this exercise. Additionally I thank Ailene Ettinger for providing the diameter measurement photograph.

The tree diameter data were provided by the H.J. Andrews Experimental Forest and Long Term Ecological Research (LTER) program, administered cooperatively by Oregon State University, the USDA Forest Service Pacific Northwest Research Station, and the Willamette National Forest. This material is based upon work supported by the National Science Foundation under the grant LTER8 DEB-2025755.

License

Original Computing Environment

```
writeLines(readLines(file.path(Sys.getenv("HOME"), ".R/Makevars")))
```

```
CC=clang

CXXFLAGS=-O3 -mtune=native -march=native -Wno-unused-variable -Wno-unused-function -Wno-macro
CXX=clang++ -arch x86_64 -ftemplate-depth-256

CXX14FLAGS=-O3 -mtune=native -march=native -Wno-unused-variable -Wno-unused-function -Wno-mac
CXX14=clang++ -arch x86_64 -ftemplate-depth-256
```

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: x86_64-apple-darwin20 (64-bit)
Running under: macOS Sonoma 14.2

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

```
other attached packages:
[1] colormap_0.1.4      rstan_2.32.3        StanHeaders_2.26.28

loaded via a namespace (and not attached):
 [1] gtable_0.3.4       jsonlite_1.8.8     compiler_4.3.2     Rcpp_1.0.11
 [5] stringr_1.5.1      parallel_4.3.2     gridExtra_2.3      scales_1.3.0
 [9] yaml_2.3.8         fastmap_1.1.1      ggplot2_3.4.4      R6_2.5.1
[13] curl_5.2.0         knitr_1.45         tibble_3.2.1       munsell_0.5.0
[17] pillar_1.9.0       rlang_1.1.2        utf8_1.2.4         V8_4.4.1
[21] stringi_1.8.3      inline_0.3.19      xfun_0.41          RcppParallel_5.1.7
[25] cli_3.6.2          magrittr_2.0.3     digest_0.6.33      grid_4.3.2
[29] lifecycle_1.0.4    vctrs_0.6.5        evaluate_0.23      glue_1.6.2
[33] QuickJSR_1.0.8     codetools_0.2-19   stats4_4.3.2       pkgbuild_1.4.3
[37] fansi_1.0.6        colorspace_2.1-0   rmarkdown_2.25     matrixStats_1.2.0
[41] tools_4.3.2        loo_2.6.0          pkgconfig_2.0.3    htmltools_0.5.7
```

Stan Program 1 homogeneous_linear_growth.stan

```
data {
  int<lower=1> N;
  int<lower=1> N_trees;
  int<lower=1> tree_N_obs[N_trees];
  int<lower=1, upper=N> tree_start_idxs[N_trees];
  int<lower=1, upper=N> tree_end_idxs[N_trees];
  vector[N] tree_years;
  vector[N] tree_dbhs;

  int<lower=1> N_year_grid;
  vector[N_year_grid] year_grid;
}

parameters {
  real<lower=0> d0[N_trees]; // Diameter at year of first observation (cm)
  real beta;                 // Linear growth rate (cm / year)
  real<lower=0> sigma;       // Measurement variability (cm)
}

model {
  d0 ~ normal(0, 150 / 2.57);     // 99% prior mass between 0 and 150 cm
  beta ~ normal(0, 2 / 2.32);     // 99% prior mass between +/- 2 cm / year
  sigma ~ normal(0, 0.25 / 2.57); // 99% prior mass between 0 and 0.25 cm

  for (t in 1:N_trees) {
    int start_idx = tree_start_idxs[t];

    int end_idx = tree_end_idxs[t];
    vector[tree_N_obs[t]] years = tree_years[start_idx:end_idx];
    vector[tree_N_obs[t]] dbhs = tree_dbhs[start_idx:end_idx];

    dbhs ~ normal(d0[t] + beta * (years - years[1]), sigma);
  }
}

generated quantities {
  real dbh_grid_pred[N_trees, N_year_grid];
  for (t in 1:N_trees) {
    real y0 = tree_years[tree_start_idxs[t]];
    for (n in 1:N_year_grid) {
      real mu = d0[t] + beta * (year_grid[n] - y0);
      dbh_grid_pred[t, n] = normal_rng(mu, sigma);
    }
  }
}
```

Stan Program 2 heterogeneous_linear_growth.stan

```
data {
  int<lower=1> N;

  int<lower=1> N_trees;
  int<lower=1> tree_N_obs[N_trees];

  int<lower=1, upper=N> tree_start_idxs[N_trees];
  int<lower=1, upper=N> tree_end_idxs[N_trees];

  vector[N] tree_years;
  vector[N] tree_dbhs;


  int<lower=1> N_year_grid;
  vector[N_year_grid] year_grid;
}

parameters {
  real<lower=0> d0[N_trees]; // Diameter at year of first observation (cm)

  real beta[N_trees];        // Linear growth rate (cm / year)
  real<lower=0> sigma;       // Measurement variability (cm)

}


model {
  d0 ~ normal(0, 150 / 2.57);     // 99% prior mass between 0 and 150 cm

  beta ~ normal(0, 2 / 2.32);     // 99% prior mass between +/- 2 cm / year
  sigma ~ normal(0, 0.25 / 2.57); // 99% prior mass between 0 and 0.25 cm


  for (t in 1:N_trees) {
    int start_idx = tree_start_idxs[t];

    int end_idx = tree_end_idxs[t];
    vector[tree_N_obs[t]] years = tree_years[start_idx:end_idx];

    vector[tree_N_obs[t]] dbhs = tree_dbhs[start_idx:end_idx];


    dbhs ~ normal(d0[t] + beta[t] * (years - years[1]), sigma);
  }

}


generated quantities {
  real dbh_grid_pred[N_trees, N_year_grid];

  for (t in 1:N_trees) {
```

Stan Program 3 `homogeneous_linear_growth.stan`