
```
title: "Forest Mensuration in R"
author: "Jeff Atkins"
date: `r format(Sys.time(), "%B %d, %Y")'
output:
word_document: default
pdf_document: default
html_document: default
```

Fitting Diameter Distributions using ForestFit

`ForestFit` version 0.6.1 is available on CRAN

Here we show how to fit a diameter distribution model to forest inventory data from the `fortedata` package which contains data from a forest manipulation experiment in northern, lower Michigan.

Installation of ForestFit

```
# install the package if not already installed and call it via library()
if(!require(ForestFit)){install.packages("ForestFit")}

## Loading required package: ForestFit

## Package 'ForestFit' version 0.7.1
## Type 'citation("ForestFit")' for citing this R package in publications.

library(ForestFit)

library(fortedata)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.5      v purrr   0.3.4
## v tibble   3.1.3      v dplyr    1.0.7
## v tidyverse 1.1.3     v stringr  1.4.0
## v readr    2.0.0      vforcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

Importing data from the FoRTE project

FoRTE, the Forest Resilience Threshold Experiment, is a stem-girdling manipulation experiment targeting various levels of disturbance severity in order to test the effect of different severity levels on carbon cycling processes. The disturbance treatment was enacted in 2019. Baseline data were collected in 2018, before the manipulation. We will be using a subset of these 2018 data to test the `ForestFit` stem distribution modeling functions.

We import data from `fortedata` by calling the `fd_inventory()` function. See Atkins et al. 2020 for a thorough description.

<https://arxiv.org/pdf/1911.11002.pdf>

```
inv <- fortedata::fd_inventory()

## These data are unpublished. Please contact Jeff Atkins to ask about using

## Data citation: ESSD

## Contact person: Jeff Atkins
```

We want to focus our analysis on the two most different areas of the forest—the moraine and outwash plain areas. The moraine is a nutrient rich area of the forest dominated by tall aspen and white birch species while the outwash plain is a nutrient poor area of the forest dominated by red maple, northern red oak, and beech.

```
# filter to plot
inv %>%
  filter(subplot_id == "AO4W" & health_status == "L") %>%
  data.frame() -> moraine

# filter to plot
inv %>%
  filter(subplot_id == "D03E" & health_status == "L") %>%
  data.frame() -> outwash.plain
```

Model Selection

On our sorted data we used the `fitmixture()` function in `ForestFit`, to model the diameter distribution of our data. The package includes multiple finite mixture distributions, designated as `family = ...` in the function. Options include: “birnbaum-saunders”, “burrxii”, “chen”, “f”, “Frechet”, “gamma”, “gompetrz”, “log-normal”, “log-logistic”, “lomax”, “skew-normal”, and “weibull”. For our analysis, we opted for “log-normal”. The output from this function includes vectors of the estimated weight, shape, and scale of model parameters. Then, further, it provides a sequence of goodness-of-fit measures : Akaike Information Criterion (AIC), Consistent Akaike Information Criterion (CAIC), Bayesian Information Criterion (BIC), Hannan-Quinn information criterion (HQIC), Anderson-Darling (AD), CramerVon Misses (CVM), Kolmogorov-Smirnov (KS), and log-likelihood (log-likelihood) statistics. We tested four distribution families for each of the moraine and outwash plain plots, with the number of components, K, set to 3.

```
“{ r fit-mixture} #fit mixture fitmixture(morainedbh_cm, "log - normal", 3)fitmixture(morainedbh_cm,
"log-logistic", 3) fitmixture(morainedbh_cm, "weibull", 3)fitmixture(morainedbh_cm, "gamma", 3)

#fit mixture fitmixture(outwash.plaindbh_cm, "log - normal", 3)fitmixture(outwash.plaindbh_cm, "log-
logistic", 3) fitmixture(outwash.plaindbh_cm, "weibull", 3)fitmixture(outwash.plaindbh_cm, "gamma", 3)
```

Model Fitting

We fit our model using the defined parameters from the Model Selection step, choosing the best model for

```
““{ r model-param}
```

```

# parameters from model moraine
weight <- c(0.165, 0.434, 0.399)
alpha <- c(29.525, 22.531, 36.454)
beta <- c(1.471, 0.583, 0.651)
param <- c(weight, alpha, beta)
XX <- seq(0, max(moraine$dbh_cm), 0.01)
pdf <- dmixture(XX, "gamma", 3, param)

# parameters from model outwash plain
weight2 <- c(0.233, 0.478, 0.287)
alpha2 <- c(16.078, 7.435, 12.103)
beta2 <- c(30.396, 10.019, 21.199)
param2 <- c(weight2, alpha2, beta2)
XX2 <- seq(0, max(outwash.plain$dbh_cm), 0.01)
pdf2 <- dmixture(XX2, "log-logistic", 3, param2)

hist(moraine$dbh_cm, freq=FALSE, breaks = 20, xlim = c(0, 40), ylim = c(0, 0.12),
      col = rgb(255,192,203, max = 255, alpha = 200), xlab="DBH [cm]", main = "")
hist(outwash.plain$dbh_cm, freq=FALSE, breaks = 20, xlim = c(0, 40), ylim = c(0, 0.12),
      col = rgb(173,216,230, max = 255, alpha = 200), add = TRUE,
      xlab="DBH [cm]", main = "")
lines(XX, pdf, col = "red", lwd = 3)
lines(XX2, pdf2, col = "blue4", lwd = 3)
legend("topright", c("Moraine", "Outwash Plain"), fill=c("red", "blue"))

```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.