
```
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## These data are unpublished.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 == "A04W" & 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”, “burxii”, “chen”, “f”, “Frechet”, “gamma”, “gom-petrz”, “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.

```
#fit mixture
fitmixture(moraine$dbh_cm, "log-normal", 3)

## $estimate
##      weight     alpha      beta
## [1,] 0.3940228 3.153556 0.1642019
## [2,] 0.4413798 2.554067 0.2160966
## [3,] 0.1645974 3.754315 0.1837698
##
## $measures
```

```

##          AIC      CAIC      BIC      HQIC         AD        CVM        KS
## [1,] 622.7805 624.7265 642.1313 630.5546 0.3719174 0.0485663 0.06003812
##      log.likelihood
## [1,]      -303.3903
##
## $cluster
##  [1] 2 1 2 3 2 3 1 1 3 1 1 1 1 2 2 2 1 1 2 2 2 1 2 2 2 1 2 1 2 2 2 3 3 3 2 2 1
## [39] 1 1 1 3 1 2 2 1 1 2 1 1 1 3 1 1 1 2 1 1 2 2 2 2 3 2 1 2 3 3 2 1 2 2 2 1
## [77] 2 2 1 1 3 3 1

fitixture(moraine$dbh_cm, "log-logistic", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.1395332 14.839186 34.12903
## [2,] 0.0573485 38.556686 53.02725
## [3,] 0.8031183 5.229314 16.52340
##
## $measures
##          AIC      CAIC      BIC      HQIC         AD        CVM        KS
## [1,] 617.0514 618.9974 636.4022 624.8255 0.4713452 0.09473043 0.09281535
##      log.likelihood
## [1,]      -300.5257
##
## $cluster
##  [1] 3 3 3 1 3 1 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 2 2 2 3 3 3
## [39] 3 1 1 1 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 2 3 3 3 1 2 3 3 3 3 3 3
## [77] 3 3 3 3 1 1 3

fitixture(moraine$dbh_cm, "weibull", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.4318083 6.040652 25.50254
## [2,] 0.1704540 5.937891 46.87494
## [3,] 0.3977377 5.801287 14.22988
##
## $measures
##          AIC      CAIC      BIC      HQIC         AD        CVM        KS
## [1,] 627.2794 629.2253 646.6301 635.0534 0.4791575 0.08189596 0.0975501
##      log.likelihood
## [1,]      -305.6397
##
## $cluster
##  [1] 3 1 3 2 3 2 1 1 2 1 1 1 3 3 3 1 1 3 3 3 1 3 3 3 1 3 3 3 2 2 2 3 3 1
## [39] 1 1 1 2 1 3 3 1 1 3 1 1 1 2 1 1 1 3 1 1 3 3 3 2 3 1 3 2 2 3 1 1 3 3 1
## [77] 3 3 1 1 2 2 1

fitixture(moraine$dbh_cm, "gamma", 3)

## $estimate
##      weight     alpha     beta
```

```

## [1,] 0.3991788 36.45396 0.6513454
## [2,] 0.4349558 22.53153 0.5836266
## [3,] 0.1658654 29.52569 1.4712090
##
## $measures
##      AIC     CAIC      BIC     HQIC      AD      CVM      KS
## [1,] 622.9066 624.8526 642.2573 630.6806 0.3464573 0.04625958 0.06669493
##      log.likelihood
## [1,] -303.4533
##
## $cluster
## [1] 2 1 2 3 2 3 1 1 3 1 1 1 1 2 2 2 1 1 2 2 2 1 2 2 2 1 2 1 2 2 2 3 3 3 2 2 1
## [39] 1 1 1 3 1 2 2 1 1 2 1 1 1 3 1 1 1 2 1 1 2 2 2 2 3 2 1 2 3 3 2 1 2 2 2 1
## [77] 2 2 1 1 3 3 1

#fit mixture
fitmixture(outwash.plain$dbh_cm, "log-normal", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.2805346 3.044752 0.1386829
## [2,] 0.4833381 2.304371 0.2514775
## [3,] 0.2361273 3.418418 0.1036404
##
## $measures
##      AIC     CAIC      BIC     HQIC      AD      CVM      KS
## [1,] 671.9424 673.5976 692.4572 680.2348 0.4589103 0.04682464 0.06884811
##      log.likelihood
## [1,] -327.9712
##
## $cluster
## [1] 2 1 2 2 2 2 1 3 2 2 2 3 1 2 3 2 2 3 3 2 2 2 1 2 3 3 3 1 2 3 3 2 2 3 3 3 3
## [39] 2 2 2 2 1 1 3 2 2 2 1 3 2 3 2 1 2 1 1 1 1 1 3 1 2 2 2 1 2 2 2 2 1 2 2 1
## [77] 2 3 2 1 3 3 1 1 1 2 2 1 1 2 2 2 1 3 3 1

fitmixture(outwash.plain$dbh_cm, "log-logistic", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.4787232 7.435686 10.01904
## [2,] 0.2875163 12.103871 21.19974
## [3,] 0.2337605 16.078028 30.39646
##
## $measures
##      AIC     CAIC      BIC     HQIC      AD      CVM      KS
## [1,] 668.8934 670.5485 689.4082 677.1858 0.3522447 0.04250903 0.06184225
##      log.likelihood
## [1,] -326.4467
##
## $cluster
## [1] 1 2 1 1 1 1 2 3 1 1 1 1 3 2 1 3 1 1 3 3 1 1 1 2 1 3 3 3 2 1 3 3 1 1 3 3 3 3
## [39] 1 1 1 1 2 2 3 1 1 1 2 3 1 3 1 2 1 2 2 2 2 2 3 2 1 1 2 1 1 1 2 1 1 2 1 1 2
## [77] 1 3 1 2 3 3 2 2 2 1 1 2 2 1 1 1 2 3 3 2
```

```

fitmixture(outwash.plain$dbh_cm, "weibull", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.2550410 9.777267 32.18260
## [2,] 0.2818059 8.793878 22.43830
## [3,] 0.4631531 4.634159 11.26931
##
## $measures
##      AIC     CAIC     BIC     HQIC      AD      CVM      KS
## [1,] 672.6846 674.3398 693.1994 680.977 0.5984044 0.07650118 0.07490253
##      log.likelihood
## [1,] -328.3423
##
## $cluster
## [1] 3 2 3 3 3 3 2 1 3 3 3 3 1 2 3 1 3 3 1 1 3 3 3 2 3 1 1 1 2 3 1 1 3 3 1 1 1 1
## [39] 3 3 3 3 2 2 1 3 3 3 2 1 3 1 3 2 3 2 2 2 2 2 2 1 2 3 3 3 2 3 3 3 3 2 3 3 2
## [77] 3 1 3 2 1 1 2 2 2 3 3 2 2 3 3 3 2 1 1 2

fitmixture(outwash.plain$dbh_cm, "gamma", 3)

## $estimate
##      weight     alpha     beta
## [1,] 0.2375431 92.41903 0.3320415
## [2,] 0.4776012 17.11540 0.6028359
## [3,] 0.2848557 53.25932 0.3981195
##
## $measures
##      AIC     CAIC     BIC     HQIC      AD      CVM      KS
## [1,] 669.8963 671.5514 690.4111 678.1887 0.4222233 0.04673506 0.06362467
##      log.likelihood
## [1,] -326.9481
##
## $cluster
## [1] 2 3 2 2 2 2 3 1 2 2 2 2 1 3 2 1 2 2 1 1 2 2 2 3 2 1 1 1 3 2 1 1 2 2 1 1 1 1
## [39] 2 2 2 2 3 3 1 2 2 2 3 1 2 1 2 3 2 3 3 3 3 3 1 3 2 2 2 3 2 2 2 3 2 2 3
## [77] 2 1 2 3 1 1 3 3 3 2 2 3 3 2 2 2 3 1 1 3

```

Model Fitting

We fit our model using the defined parameters from the Model Selection step, choosing the best model for each data set: “gamma” for the moraine and “log-logistic” for the outwash plain.

```

# 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"))

```

