

Package ‘FitUltD’

May 7, 2026

Type Package

Title Fit Univariate Mixed and Usual Distributions

Version 3.1.0

Author José Carlos Del Valle <jcval194@gmail.com>

Maintainer José Carlos Del Valle <jcval194@gmail.com>

Description Extends the fitdist() (from 'fitdistrplus') adding the Anderson-Darling ad.test() (from 'AD-GofTest') and Kolmogorov Smirnov Test ks.test() inside, trying the distributions from 'stats' package by default and offering a second function which uses mixed distributions to fit, this distributions are split with unsupervised learning, with Mclust() function (from 'mclust').

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.2.0), mclust

Imports ADGofTest, fitdistrplus, assertthat, MASS, purrr, ggplot2, cowplot, methods, stats

RoxygenNote 6.1.1

URL <https://github.com/jcval194/FitUltD>

BugReports <https://github.com/jcval194/FitUltD/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2019-09-11 13:30:02 UTC

Contents

FDist	2
FDistUlt	3
Index	5

FDist	<i>Fit of univariate distributions with censored data ignored by default or can be inputed.</i>
-------	---

Description

Fit of univariate distributions with censored data ignored by default or can be inputed.

Usage

```
FDist(X, gen = 1, Cont = TRUE, inputNA, plot = FALSE,
      p.val_min = 0.05, crit = 2, DPQR = TRUE)
```

Arguments

X	A random sample to be fitted.
gen	A positive integer, indicates the sample length to be generated by the fit, 1 by default.
Cont	TRUE, by default the distribution is considered as continuous.
inputNA	A number to replace censored values, if is missing, only non-censored values will be evaluated.
plot	FALSE. If TRUE, a plot showing the data distribution will be given.
p.val_min	0.05, minimum p.value for Anderson Darling and KS Test to non-reject the null hypothesis and continue with the process.
crit	A positive integer to define which test will use. If 1, show the distributions which were non-rejected by the Anderson Darling or Kolmogorov Smirnov tests, in other cases the criterion is that they mustn't be rejected by both tests.
DPQR	TRUE, creates the distribution function, density and quantile function with the names dfit, pfit and qfit.

Value

Calculate the distribution name with parameters, a function to reproduce random values from that distribution, a numeric vector of random numbers from that function, Anderson Darling and KS p.values, a plot showing the distribution difference between the real sample and the generated values and a list with the random deviates generator, the distribution function, density and quantile function

Examples

```
set.seed(31109)
FIT1<-FDist(rnorm(1000,10),p.val_min=.03,crit=1,plot=TRUE)

#Random Variable
FIT1[[1]]

#Random numbers generator
```

```

FIT1[[2]]()

#Random sample
FIT1[[3]]

#Goodness of fit tests results
FIT1[[4]]

#Plot
FIT1[[5]]

#Functions r, p, d, q
FIT1[[6]]

```

FDistUlt	<i>Fits a set of observations (random variable) to test whether is drawn from a certain distribution</i>
----------	--

Description

Fits a set of observations (random variable) to test whether is drawn from a certain distribution

Usage

```
FDistUlt(X, n.obs = length(X), ref = "OP", crt = 1, plot = FALSE,
         subplot = FALSE, p.val_min = 0.05)
```

Arguments

X	A random sample to be fitted.
n.obs	A positive integer, is the length of the random sample to be generated
ref	Aumber of clusters to use by the kmeans function to split the distribution, if isn't a number, uses mclust classification by default.
crt	Criteria to be given to FDist() function
plot	FALSE. If TRUE, generates a plot of the density function.
subplot	FALSE. If TRUE, generates the plot of the mixed density function's partitions.
p.val_min	Minimum p.value to be given to non-reject the null hypothesis.

Value

A list with the density functions, a random sample, a data frame with the KS and AD p.values results, the corresponding plots an the random numbers generator functions

Examples

```
set.seed(31109)
X<-c(rnorm(193,189,12),rweibull(182,401,87),rgamma(190,40,19))

A_X<-FDistUlt(X,plot=TRUE,subplot=TRUE)

A_X<-FDistUlt(X,plot=TRUE,subplot=TRUE,p.val_min=.005)

# Functions generated
A_X[[1]][[1]]()
# Random sample
A_X[[2]]

#Distributions
A_X[[3]]

# Plots
par(mfrow=c(1,2))
A_X[[4]][[1]]
A_X[[4]][[2]]

# More functions
A_X[[5]][[1]]()
```

Index

[FDist](#), [2](#)

[FDistUlt](#), [3](#)