

# Package: MixedTS (via r-universe)

September 14, 2024

**Type** Package  
**Title** Mixed Tempered Stable Distribution  
**Version** 1.0.4  
**Date** 2015-10-22  
**Depends** methods, stats, graphics, stats4, MASS  
**Author** Lorenzo Mercuri, Edit Roji  
**Maintainer** Lorenzo Mercuri <lorenzo.mercuri@unimi.it>  
**Description** We provide detailed functions for univariate Mixed Tempered Stable distribution.  
**License** GPL (>= 2)  
**Repository/R-Forge/Project** mixedts  
**Repository/R-Forge/Revision** 15  
**Repository/R-Forge/DateTimeStamp** 2015-10-22 16:15:11  
**Date/Publication** 2015-10-25 17:21:21  
**NeedsCompilation** no  
**Repository** <https://lore140378.r-universe.dev>  
**RemoteUrl** <https://github.com/cran/MixedTS>  
**RemoteRef** HEAD  
**RemoteSha** 420550731ce101425f99b4412205947cccf80093

## Contents

MixedTS-package . . . . .	2
dMixedTS-methods . . . . .	3
MixedTS-class . . . . .	4
MixedTS.qmle-class . . . . .	5
mle.MixedTS . . . . .	6
param.MixedTS-class . . . . .	7
pMixedTS-methods . . . . .	8
qMixedTS-methods . . . . .	9
rMixedTS-methods . . . . .	9
setMixedTS.param . . . . .	10

---

MixedTS-package	<i>Mixed Tempered Stable Distribution</i>
-----------------	---

---

### Description

This package provides detailed functions for univariate Mixed Tempered Stable distribution distribution with Gamma density. This distribution encompasses, Variance Gamma and Symmetric Geo-Stable as special cases. The package contains routine for mle estimation, for the computation of density, probability, quantile and random numbers

### Details

Package: MixedTS  
Type: Package  
License: GPL ( $\geq 2$ )

### Author(s)

Lorenzo Mercuri, Edit Rroji

Maintainer: Lorenzo Mercuri <lorenzo.mercuri@unimi.it>

### References

- Barndorff-Nielsen, O.E., Kent, J. and Sorensen, M. (1982): Normal variance-mean mixtures and z-distributions, *International Statistical Review*, 50, 145-159.
- Kuchler, U. and Tappe, S. (2014): Exponential stockmodels driven by tempered stable processes. *Journal of Econometrics*, 181 (1), 53-63.
- Madan, D.B. and Seneta E. (1990): The variance gamma (V.G.) model for share market returns, *Journal of Business*, 63, 511-524
- Rroji, E and Mercuri, L.(2014): Mixed Tempered Stable distribution *UNIMI-Research Papers in Economics, Business, and Statistics*, 64.

**Description**

This Method returns the density of a Mixed Tempered Stable

**Methods**

signature(object = "param.MixedTS", x = numeric(), setSup=NULL, setInf=NULL, N=2^10)

This method returns an object of class MixedTS where the slot dens contains the value of the density evaluated on the x. setSup and setInf are used to choose + infinity and - infinity. N is the number of point used for discretization in fft algorithm.

**Examples**

```
# First Example

# Density of MixedTS with Gamma

ParamEx1<-setMixedTS.param(mu0=0, mu=0, sigma=0.4, a=1.5,
                           alpha=0.8, lambda_p=4, lambda_m=1,
                           Mixing="Gamma")

# support

x<-seq(-3,1,length=100)

dens1<-dMixedTS(x=x,object=ParamEx1,setSup=10,setInf=-10,N=2^7)

plot(dens1)

# Density of MixedTS with IG

Mix<-"User"

logmgf<-("lamb/mu1*(1-sqrt(1-2*mu1^2/lamb*u))")

parMix<-list(lamb=1,mu1=1)

ParamEx2<-setMixedTS.param(mu0=0, mu=0, sigma=0.4, a=logmgf,
                           alpha=0.8, lambda_p=4, lambda_m=1,
                           Mixing=Mix,paramMixing=parMix)

x<-seq(-3,1,length=100)

dens2<-dMixedTS(x=x,object=ParamEx2,setSup=10,setInf=-10,N=2^7)

plot(dens2)
```

---

MixedTS-class	<i>"MixedTS": A class for informations about Mixed Tempered Stable</i>
---------------	--

---

### Description

Mathematical description of the Mixed Tempered Stable distribution.

This class inherits from the class `param.MixedTS` and is a superclass for `MixedTS.qmle-class`.

### Objects from the Class

This object is built by the following methods:

`dMixedTS`, `pMixedTS`, `qMixedTS`, `rMixedTS`.

### Slots

**Data:** Object of class "numeric" containing a random number. This slot is filled when the method `rMixedTS` is used.

**dens:** Object of class "numeric" that contains the density of the MixedTS. This slot is filled by `dMixedTS`.

**prob:** Object of class "numeric" that contains the probability of the MixedTS. This slot is filled by `pMixedTS` and `pMixedTS`.

**xMixedTS:** Object of class "numeric" that contains the support for the density and probability.

**quantile:** Object of class "logical". If TRUE the object is built by the method `qMixedTS`. If FALSE the object is built by the method `qMixedTS`.

**mu0:** Object of class "numeric". See `param.MixedTS`.

**mu:** Object of class "numeric". See `param.MixedTS`.

**sigma:** Object of class "numeric". See `param.MixedTS`.

**a:** Object of class "vector". See `param.MixedTS`.

**alpha:** Object of class "numeric". See `param.MixedTS`.

**lambda\_p:** Object of class "numeric". See `param.MixedTS`.

**lambda\_m:** Object of class "numeric". See `param.MixedTS`.

**Mixing:** Object of class "character". See `param.MixedTS`.

**paramMixing:** Object of class "list". See `param.MixedTS`.

**MixingLogMGF:** Object of class "OptionalFunction". See `param.MixedTS`.

### Extends

Class "`param.MixedTS`", directly.

### Methods

**plot** signature(`x = "MixedTS"`, ...)

---

MixedTS.qmle-class	MixedTS.qmle: <i>a class for Maximum Likelihood of Mixed Tempered Stable</i>
--------------------	--

---

### Description

This class is constructed by function `MixedTS.qmle`. It is a subclass for the `MixedTS`-class

### Objects from the Class

Objects can be created by function `MixedTS.qmle`.

### Slots

`time`: Object of class "numeric". Computational Time.  
`coef`: Object of class "numeric". Estimated parameters.  
`vcov`: Object of class "matrix". Approximate variance-covariance matrix.  
`min`: Object of class "numeric". Minimum value of objective function.  
`details`: Object of class "list". A list as returned from `constrOptim`  
`nobs`: Object of class "integer". Number of observation.  
`method`: Object of class "character". The optimization method used.  
`Data`: Object of class "numeric". See `MixedTS`-class.  
`dens`: Object of class "numeric". See `MixedTS`-class.  
`prob`: Object of class "numeric". See `MixedTS`-class.  
`xMixedTS`: Object of class "numeric". See `MixedTS`-class.  
`quantile`: Object of class "logical". See `MixedTS`-class.  
`mu0`: Object of class "numeric". See `MixedTS`-class.  
`mu`: Object of class "numeric". See `MixedTS`-class.  
`sigma`: Object of class "numeric". See `MixedTS`-class.  
`a`: Object of class "vector". See `MixedTS`-class.  
`alpha`: Object of class "numeric". See `MixedTS`-class.  
`lambda_p`: Object of class "numeric". See `MixedTS`-class.  
`lambda_m`: Object of class "numeric". See `MixedTS`-class.  
`Mixing`: Object of class "character". See `MixedTS`-class.  
`paramMixing`: Object of class "list". See `MixedTS`-class.  
`MixingLogMGF`: Object of class "OptionalFunction". See `MixedTS`-class.

### Extends

Class "`MixedTS`", directly. Class "`param.MixedTS`", by class "`MixedTS`", distance 2.

**Methods**

```

summary signature(.Object = "MixedTS.qmle")
coef signature(.Object = "MixedTS.qmle")
vcov signature(.Object = "MixedTS.qmle")
logLik signature(.Object = "MixedTS.qmle")
BIC signature(.Object = "MixedTS.qmle")
AIC signature(.Object = "MixedTS.qmle")

```

---

mle.MixedTS

*Maximum Likelihood Estimation for MixedTS distribution*


---

**Description**

Estimate MixedTS parameters using the Maximum Likelihood Estimation procedure.

**Usage**

```

mle.MixedTS(object, start = list(), Data = NULL,
             method = "L-BFGS-B", fixed.param = NULL,
             lower.param = NULL, upper.param = NULL,
             setSup = NULL, setInf = NULL, N = 2^10)

```

**Arguments**

<code>object</code>	an object of class <code>param.MixedTS</code> that contains informations about the model.
<code>start</code>	a list of parameter for the mle.
<code>Data</code>	a numeric object containing the dataset.
<code>method</code>	methods for optimization routine. See <code>optim</code> for more details.
<code>fixed.param</code>	a list of the model parameter that must be fix during optimization routine. Choosing <code>alpha=2</code> the function returns the estimate parameters for the Normal Variance Mean Mixture distribution.
<code>lower.param</code>	a list containing the lower bound for the parameters.
<code>upper.param</code>	a list containing the upper bound for the parameters.
<code>setSup</code>	Internal parameter. see documentation for <code>dMixedTS</code> for more details.
<code>setInf</code>	Internal parameter. see documentation for <code>dMixedTS</code> for more details.
<code>N</code>	Internal parameter. see documentation for <code>dMixedTS</code> for more details.

**Value**

The function returns an object of class `MixedTS.qmle`.

## Examples

```
# First Example:
# We define the Mixed Tempered Stable using the function setMixedTS.param

ParamEx1<-setMixedTS.param(mu0=0, mu=0, sigma=0.4, a=1.5,
                           alpha=0.8, lambda_p=4, lambda_m=1, Mixing="Gamma")

# We generate a sample using the rMixedTS method
set.seed(100)
Rand1 <- rMixedTS(x=5000,object=ParamEx1, setSup=10,setInf=-10,N=2^9)

# Estimate procedure
## Not run:
est1<-mle.MixedTS(object=Rand1 , setSup=10,setInf=-10,N=2^9)
# Show results

summary(est1)

## End(Not run)
```

---

```
param.MixedTS-class  "param.MixedTS": A mathematical Description of the Mixed Tem-
                    pered Stable
```

---

## Description

Main class of the package MixedTS.

## Objects from the Class

Objects can be created by calls of the form setMixedTS.

## Slots

**mu0**: a numeric object. mu0 parameter belongs to the real axis.

**mu**: a numeric object. mu parameter belongs to the real axis

**sigma** a numeric object. sigma parameter assumes value from zero to infinity.

**a** a vector object. If numeric, the mixing density  $V$  is a Gamma and  $a$  is the value of the shape parameter. If string,  $a$  is the log of the moment generating function of the mixing density  $V$ .

**alpha** a numeric object that takes value from 0 to 2. If alpha is fixed to 2, the Mixed Tempered Stable becomes the Normal Variance Mean mixture.

**lambda\_p** a positive numeric object. It is the right tempering parameter of the random variable  $X$ .

**lambda\_m** a positive numeric object. It is the left tempering parameter of the random variable  $X$

**Mixing** a string object indicating the nature of the mixing density  $V$ . If `Mixing="Gamma"` (default value), the  $V$  random variable is a Gamma. If `Mixing="User"`, the user has to specify the log of the moment generating function of the  $V$  random variable.

**paramMixing** a list object. It is an empty list when `Mixing="Gamma"`. If `Mixing="User"`, it is used to pass the values of the Mixing density parameters defined by the User through slot `a`.

**MixingLogMGF**: This slot contains a function that returns the logarithm of mgf for the Mixing density. The function is built internally using the information contained into the slots `a`, `paramMixing`.

**Parametrization**: String that indicates the parametrization used by user for the MixedTS

## Methods

**dMixedTS** signature(object = "param.MixedTS"): Method for computing density of MixedTS. See "[dMixedTS-methods](#)" for more details.

**pMixedTS** signature(object = "param.MixedTS"): Method for computing probability of MixedTS. See "[pMixedTS-methods](#)" for more details.

**qMixedTS** signature(object = "param.MixedTS"): Method for computing quantile of MixedTS. See "[qMixedTS-methods](#)" for more details.

**rMixedTS** signature(object = "param.MixedTS"): Method for computing random numbers of MixedTS. See "[rMixedTS-methods](#)" for more details.

**initialize** signature(object = "param.MixedTS").

**Qparam.MixedTS** signature(object = "param.MixedTS").

---

pMixedTS-methods

*Probability of Mixed Tempered Stable distribution*

---

## Description

This Method returns the cdf of a Mixed Tempered Stable

## Methods

signature(object = "param.MixedTS", x = numeric(), setSup=NULL, setInf=NULL, N=2^10)

This method returns an object of class MixedTS where the slot `prob` contains the value of the probability evaluated on the `x`. `setSup` and `setInf` are used to choose + infinity and - infinity. `N` is the number of points used for discretization in the FFT algorithm.

## Examples

```
# First Example

# Density of MixedTS with Gamma

ParamEx1<-setMixedTS.param(mu0=0, mu=0, sigma=0.4, a=1.5,
                           alpha=0.8, lambda_p=4, lambda_m=1,
                           Mixing="Gamma")

# support
```



```

x<-seq(-3,1,length=100)

prob1<-pMixedTS(x=x,object=ParamEx1,setSup=10,setInf=-10,N=2^7)

plot(prob1)

# Prob of MixedTS with IG

Mix<-"User"

parMix<-list(lamb=1,mu1=1)

logmgf<-("lamb/mu1*(1-sqrt(1-2*mu1^2/lamb*u))")

ParamEx2<-setMixedTS.param(mu0=0, mu=0, sigma=0.4, a=logmgf,
                           alpha=0.8, lambda_p=4, lambda_m=1,
                           Mixing=Mix,paramMixing=parMix)

x<-seq(-3,1,length=100)

prob2<-pMixedTS(x=x,object=ParamEx2,setSup=10,setInf=-10,N=2^7)

plot(prob2)

```

---

qMixedTS-methods

*Quantile of Mixed Tempered Stable distribution*


---

### Description

This Method returns the quantile of a Mixed Tempered Stable.

### Methods

```
signature(object = "param.MixedTS", x = numeric(), setSup=NULL, setInf=NULL, N=2^10)
```

This method returns an object of class MixedTS where the slot prob contains the value of the quantile evaluated on the x (x is the probability). setSup and setInf are used to choose + infinity and - infinity. N is the number of point used for discretization in fft algorithm.

---

rMixedTS-methods

*Random number of Mixed Tempered Stable distribution*


---

### Description

This Method returns the quantile of a Mixed Tempered Stable.

## Methods

```
signature(object = "param.MixedTS", x = numeric(), setSup=NULL, setInf=NULL, N=2^10)
```

This method returns an object of class MixedTS where the slot Data contains a set of size  $x$  of random numbers. setSup and setInf are used to choose + infinity and - infinity.  $N$  is the number of point used for discretization in fft algorithm.

---

setMixedTS.param	<i>Mixed Tempered Stable distribution</i>
------------------	---

---

## Description

setMixedTS describes the Mixed Tempered Stable distribution introduced in Rroji and Mercuri (2014):

### Definition

We say that a continuous random variable  $Y$  follows a Mixed Tempered Stable distribution if:

$$Y = \mu_0 + \mu * V + \sigma * \sqrt{V} * Z$$

The conditional distribution of random variable given  $V=v$  is a standardized Tempered Stable with parameters  $(\alpha, \lambda_p * \sqrt{v}, \lambda_m)$  (see Kuchler, U. and Tappe, S. 2014). The distribution of  $V$  is infinitely divisible defined on the positive axis.

## Usage

```
setMixedTS.param(mu0 = numeric(), mu = numeric(),
  sigma = numeric(), a, alpha = numeric(),
  lambda_p = numeric(), lambda_m = numeric(),
  param = numeric(), Mixing = "Gamma", paramMixing = list(), Parametrization = "A")
```

## Arguments

mu0	a numeric object. mu0 parameter belongs to the real axis.
mu	a numeric object. mu parameter belongs to the real axis
sigma	a numeric object. sigma parameter assumes value from zero to infinity.
a	a vector object. If numeric, the mixing density $V$ is a Gamma and $a$ is the value of the shape parameter. If string, $a$ is the log of the moment generating function of the mixing density $V$ .
alpha	a numeric object that takes value from 0 to 2. If alpha is fixed to 2, the Mixed Tempered Stable becomes the Normal Variance Mean mixture.
lambda_p	a positive numeric object. It is the right tempering parameter of the random variable $X$ .
lambda_m	a positive numeric object. It is the left tempering parameter of the random variable $X$
param	a numeric object containing the Mixed Tempered Stable parameters. It is not necessary if we use the previous inputs for defining the distribution. See documentation for more details.





# Index

dMixedTS (dMixedTS-methods), 3  
dMixedTS, param.MixedTS-method  
(dMixedTS-methods), 3  
dMixedTS-methods, 3

initialize, MixedTS-method  
(MixedTS-class), 4  
initialize, MixedTS.qmle-method  
(MixedTS.qmle-class), 5

Mixed Tempered Stable distribution  
(mle.MixedTS), 6  
MixedTS, 5  
MixedTS (MixedTS-package), 2  
MixedTS-class, 4  
MixedTS-package, 2  
MixedTS-parameters (setMixedTS.param),  
10  
MixedTS.qmle-class, 5  
mle (mle.MixedTS), 6  
mle.MixedTS, 6

Normal Variance Mean Mixture  
(mle.MixedTS), 6

param.MixedTS, 4, 5, 11  
param.MixedTS (param.MixedTS-class), 7  
param.MixedTS-class, 7  
plot, MixedTS, ANY-method  
(MixedTS-class), 4  
pMixedTS (pMixedTS-methods), 8  
pMixedTS, param.MixedTS-method  
(pMixedTS-methods), 8  
pMixedTS-methods, 8

qMixedTS (qMixedTS-methods), 9  
qMixedTS, param.MixedTS-method  
(qMixedTS-methods), 9  
qMixedTS-methods, 9

rMixedTS (rMixedTS-methods), 9  
rMixedTS, param.MixedTS-method  
(rMixedTS-methods), 9  
rMixedTS-methods, 9  
setMixedTS.param, 10