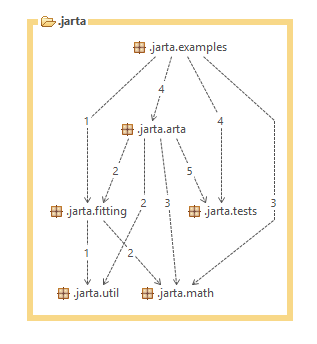
Recherche JARTA

# Jarta



## Allgemein

Wir wollen die Funktionalität von JARTA in eine C# Bibliothek transverieren. Dieses Dokument dient der Analyse von JARTA und der Vorbereitung JARTA in c# zu implementieren.

Jarta besteht aus sechs Packages und referenziert diverse externe Bibliotheken. Wir werden hier genauer auf diese eingehen, um jeweilige Problemstellung die uns erwarten im Vorhinein zu vermindern und um gleichwertige C# Biblitotheken zu finden.

Die Struktur von Jarta soll möglichst übernommen werden.

## Begriffe

### Mersenne Twister

* <https://de.wikipedia.org/wiki/Mersenne-Twister>
* <https://en.wikipedia.org/wiki/Mersenne_Twister>
* <https://www.programmingalgorithms.com/algorithm/mersenne-twister>
* <https://www.codeproject.com/articles/5147/a-c-mersenne-twister-class>

### Empirical Distribution

* <https://en.wikipedia.org/wiki/Empirical_distribution_function>
* <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/ecdf.html>

### Normal Distribution

* <https://en.wikipedia.org/wiki/Normal_distribution>
* <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Normal.html>

### Real Distribution

* <http://en.cppreference.com/w/cpp/numeric/random/uniform_real_distribution>
* <https://msdn.microsoft.com/de-de/library/ee462299.aspx>

### Uniform Real Distribution

* <http://en.cppreference.com/w/cpp/numeric/random/uniform_real_distribution>
* <https://msdn.microsoft.com/de-de/library/ee462299.aspx>

### Cholesky Decomposition

* <https://en.wikipedia.org/wiki/Cholesky_decomposition>
* <https://de.wikipedia.org/wiki/Cholesky-Zerlegung>
* <http://rosettacode.org/wiki/Cholesky_decomposition>

## Checked Exceptions

In C# existieren keine Checked Exceptions. Hier wird definiert wie diese Exceptions gelöst werden.

## Packages

### JARTA.arta

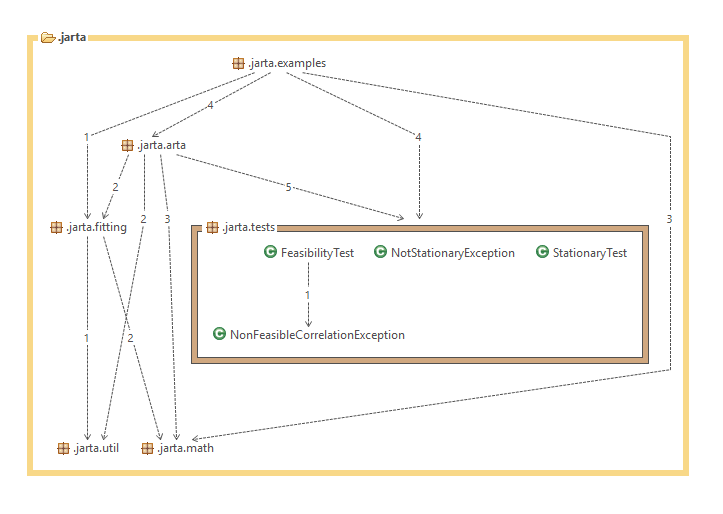
#### ArtaProcessFactory

Mittels des Factory Pattern werden hier die verschiedenen Arta Prozesse erzeugt. Hier kann zwischen drei Distributionen unterschieden werden:

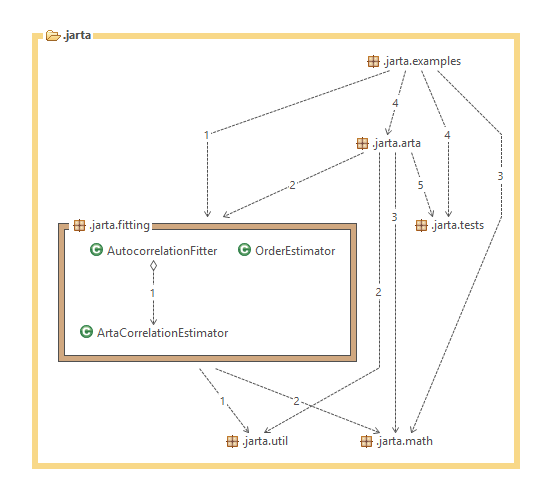
* Normal Distribution
* Real Distribution
* Uniform Real Distribution

#### ArProcessFactory

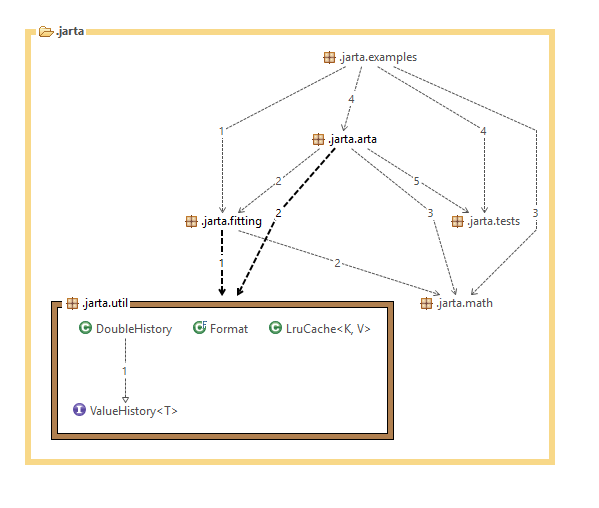
### JARTA.tests



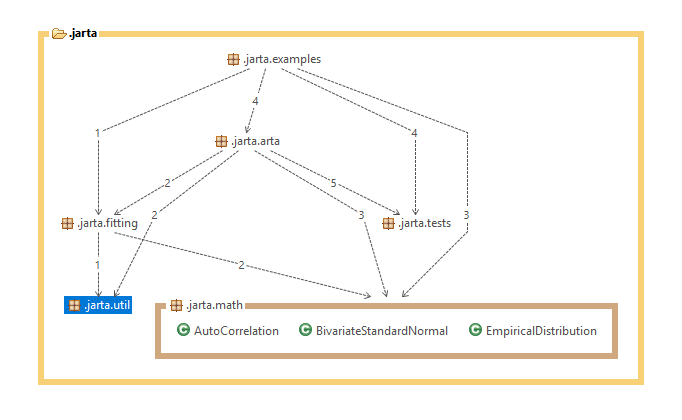
### JARTA.fitting



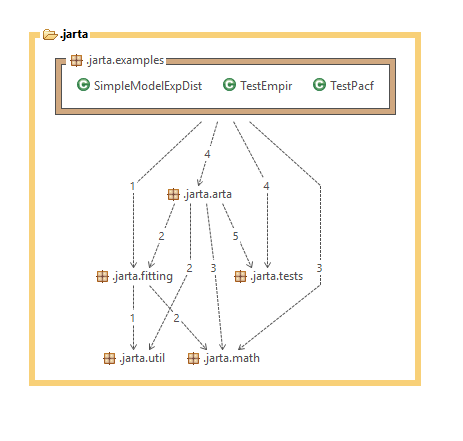
### JARTA.util



### JARTA.math



### JARTA.examples



## Bibliotheken

### Java

#### Org.apache.commons.math3.distribution.NormalDistribution

* sample()
* getNumericalVariance()
* NormalDistribution(x,y,z,u)
* NumerivalMean()
* getcumulativeProbability()

#### Org.apache.commons.math3.random.MersenneTwister.MersenneTwister()

* new MersenneTwister()

#### Org.apache.commons.random.RandomGenerator

#### Org.apache.commons.linear.RealMatrix

* new Array2DRowRealMatrix(double[]).transpose()
* multiply

#### Org.apache.commons.linear.CholeskyDecomposition

* getSolver()
* getInverse()

#### Org.apache.commons.random.RandomAdaptor

#### Org.apache.commons.math3.distribution.RealDistribution

* inverseCumulativeProbability
* getClass().getSimpleName()

#### Org.apache.commons.math3.distribution.UniformRealDistribution

* getSupportLowerBound()
* getSupportUpperBound()

#### Org.apache.commons.math3.distribution.ExponentialDistribution

### C#

Hier werden für die oben erwähnten Java Bibliotheken alternativen aufgeführt

* <https://stackoverflow.com/questions/876597/is-there-a-standard-c-sharp-lib-like-apache-commons-for-java>
* <https://github.com/Fylax/Apache-Commons-Math3-C->