

# Package: rjd3highfreq (via r-universe)

October 10, 2024

**Type** Package

**Title** Seasonal Adjustment of High Frequency Data with 'JDemetra+ 3.x'

**Version** 2.1.1

**Description** R Interface to 'JDemetra+ 3.x'

(<https://github.com/jdemetra>) time series analysis software.

It provides functions for seasonal adjustment of high-frequency data displaying multiple, non integer periodicities.

Pre-adjustment with extended airline model and Arima Model Based decomposition.

**Depends** R (>= 4.1.0)

**Imports** rJava (>= 1.0-6), RProtoBuf (>= 0.4.17), rjd3toolkit (>= 3.2.2), rjd3sts (>= 2.1.0), checkmate, methods

**Remotes** github::rjdverse/rjd3toolkit, github::rjdverse/rjd3sts

**SystemRequirements** Java (>= 17)

**License** EUPL

**URL** <https://github.com/rjdverse/rjd3highfreq>,  
<https://rjdverse.github.io/rjd3highfreq/>

**LazyData** TRUE

**Suggests** knitr, rmarkdown

**RoxygenNote** 7.3.1

**BugReports** <https://github.com/rjdverse/rjd3highfreq/issues>

**Encoding** UTF-8

**Collate** 'utils.R' 'jd3\_extendedairline.R' 'jd3\_fractionalairline.R'  
'plot.R' 'print.R' 'zzz.R'

**Repository** <https://rjdverse.r-universe.dev>

**RemoteUrl** <https://github.com/rjdverse/rjd3highfreq>

**RemoteRef** v2.1.1

**RemoteSha** 01acfc49692286761ff204931df1e986e7580e11

## Contents

.arima_extract . . . . .	2
.extendedairline_estimation . . . . .	3
.extendedairline_regarima . . . . .	3
.extendedairline_spec . . . . .	4
.extended_airline_loglevel . . . . .	5
.extended_airline_outliers . . . . .	5
.plot_jd . . . . .	6
.ucm_extract . . . . .	6
fractionalAirlineDecomposition . . . . .	7
fractionalAirlineDecomposition_raw . . . . .	7
fractionalAirlineDecomposition_ssf . . . . .	8
fractionalAirlineEstimation . . . . .	8
jd2r_fractionalAirlineDecomposition . . . . .	9
jd2r_multiAirlineDecomposition . . . . .	10
multiAirlineDecomposition . . . . .	10
multiAirlineDecomposition_raw . . . . .	11
multiAirlineDecomposition_ssf . . . . .	12
plot.JDFractionalAirlineDecomposition . . . . .	12
plot.JDFractionalAirlineEstimation . . . . .	13
print.JDFractionalAirlineDecomposition . . . . .	13
print.JDFractionalAirlineEstimation . . . . .	14
<b>Index</b>	<b>15</b>

---

.arima_extract	<i>Title</i>
----------------	--------------

---

### Description

Title

### Usage

```
.arima_extract(jrslt, path)
```

### Arguments

path

---

.extendedairline\_estimation  
*Title*

---

**Description**

Title

**Usage**

```
.extendedairline_estimation(  
  jregarima,  
  jspec,  
  eps = 1e-09,  
  exactHessian = FALSE  
)
```

**Arguments**

exactHessian

---

.extendedairline\_regarima  
*Creates a java RegArima models based on an extended airline spec*

---

**Description**

Creates a java RegArima models based on an extended airline spec

**Usage**

```
.extendedairline_regarima(y, jspec, mean = FALSE, X = NULL)
```

**Arguments**

y	y
jspec	Java spec
mean	Mean correction (to be avoided)
X	Regression variables

**Value**

A Java RegArima model

## Examples

```
jspec<- .extendedairline_spec(c(12))  
.extendedairline_regarima(rjd3toolkit::ABS$X0.2.09.10.M, jspec)
```

---

.extendedairline\_spec *Internal routine to create an ExtendedAirlineSpec*

---

## Description

Internal routine to create an ExtendedAirlineSpec

## Usage

```
.extendedairline_spec(  
  periodicities,  
  differencing = -1,  
  ar = FALSE,  
  toint = FALSE  
)
```

## Arguments

periodicities	Periodicities
differencing	Differencing order. -1 for automatic computation
ar	Use of an AR regular stationary polynomial instead of a MA polynomial
toint	Round periodicities to integers

## Value

A Java ExtendedAirlineSpec object

## Examples

```
.extendedairline_spec(c(7, 365.25))
```

---

*.extended\_airline\_loglevel*  
*Title*

---

**Description**

Title

**Usage**

```
.extended_airline_loglevel(jregarima, jspec, precision = 1e-05)
```

**Arguments**

precision

---

*.extended\_airline\_outliers*  
*Title*

---

**Description**

Title

**Usage**

```
.extended_airline_outliers(  
  jregarima,  
  jspec,  
  types = c("ao"),  
  start = 0,  
  end = 0,  
  critical_value = 0,  
  max_outliers = 30,  
  max_round = 30  
)
```

**Arguments**

max\_round

---

.plot_jd	<i>Custom Plot Function on JD+ template</i>
----------	---

---

### Description

This function creates a customized plot in the same template as JD+ GUI color and forms.

### Usage

```
.plot_jd(x, y, col, legend_txt = NULL, ...)
```

### Arguments

x	Numeric vector, x-axis values.
y	List of numeric vectors, y-axis values for different series.
col	Vector of colors for different series.
legend_txt	Character vector of legend labels for different series.
...	Additional graphical parameters.

### Value

'NULL' (invisible).

---

.ucm_extract	<i>Title</i>
--------------	--------------

---

### Description

Title

### Usage

```
.ucm_extract(jrslt, cmp)
```

### Arguments

cmp

---

fractionalAirlineDecomposition

*Perform an Arima Model Based (AMB) decomposition*


---

**Description**

Perform an Arima Model Based (AMB) decomposition

**Usage**

```
fractionalAirlineDecomposition(
  y,
  period,
  sn = FALSE,
  stde = FALSE,
  nbcasts = 0,
  nfcasts = 0,
  log = FALSE,
  y_time = NULL
)
```

**Arguments**

y	input time series.
period	period of the seasonal component, any positive real number.
sn	decomposition into signal and noise (2 components only). The signal is the seasonally adjusted series and the noise the seasonal component.
stde	Boolean: TRUE: compute standard deviations of the components. In some cases (memory limits), it is currently not possible to compute them
nbcasts	number of backcasts.
nfcasts	number of forecasts.
y_time	vector of times at which 'y' is indexed
adjust	Boolean: TRUE: actual fractional airline model is to be used, FALSE: the period is rounded to the nearest integer.

---

fractionalAirlineDecomposition\_raw

*Title*


---

**Description**

Title

**Usage**

```
fractionalAirlineDecomposition_raw(  
  y,  
  period,  
  sn = FALSE,  
  stde = FALSE,  
  nbcasts = 0,  
  nfcasts = 0  
)
```

**Arguments**

nfcasts

---

```
fractionalAirlineDecomposition_ssf  
  Title
```

---

**Description**

Title

**Usage**

```
fractionalAirlineDecomposition_ssf(jdecomp)
```

**Arguments**

jdecomp

---

```
fractionalAirlineEstimation  
  Linearize the series with a fractional airline model
```

---

**Description**

Linearize the series with a fractional airline model



**Usage**

```
fractionalAirlineEstimation(
  y,
  periods,
  x = NULL,
  ndiff = 2,
  ar = FALSE,
  outliers = NULL,
  criticalValue = 6,
  precision = 1e-12,
  approximateHessian = FALSE,
  nfcasts = 0,
  log = FALSE,
  y_time = NULL
)
```

**Arguments**

y	input time series.
periods	vector of periods values of the seasonal component, any positive real numbers.
x	matrix of user-defined regression variables (see rjd3toolkit for building calendar regressors).
outliers	type of outliers sub vector of c("AO", "LS", "WO")
criticalValue	Critical value for automatic outlier detection
precision	Precision of the likelihood
approximateHessian	Compute approximate hessian (based on the optimizing procedure)
nfcasts	Number of forecasts
log	a logical
y_time	vector of times at which 'y' is indexed

---

jd2r\_fractionalAirlineDecomposition  
*Title*

---

**Description**

Title

**Usage**

```
jd2r_fractionalAirlineDecomposition(
  jrslt,
  sn = FALSE,
  stde = FALSE,
  period,
  log = FALSE,
  y_time = NULL
)
```

**Arguments**

y\_time                vector of times at which the time series is indexed

---

jd2r\_multiAirlineDecomposition  
*Title*

---

**Description**

Title

**Usage**

```
jd2r_multiAirlineDecomposition(
  jrslt,
  stde = FALSE,
  periods,
  log = FALSE,
  y_time = NULL
)
```

**Arguments**

y\_time                vector of times at which the time series is indexed

---

multiAirlineDecomposition  
*Perform an Arima Model Based (AMB) decomposition on several pe-  
riodicities at once*

---

**Description**

Perform an Arima Model Based (AMB) decomposition on several periodicities at once

**Usage**

```
multiAirlineDecomposition(
  y,
  periods,
  ndiff = 2,
  ar = FALSE,
  stde = FALSE,
  nbcasts = 0,
  nfcasts = 0,
  log = FALSE,
  y_time = NULL
)
```

**Arguments**

y	input time series.
periods	vector of periods values of the seasonal component, any positive real numbers.
stde	Boolean: TRUE: compute standard deviations of the components. In some cases (memory limits), it is currently not possible to compute them
nbcasts	number of backcasts.
nfcasts	number of forecasts.
y_time	vector of times at which 'y' is indexed
adjust	Boolean: TRUE: actual fractional airline model is to be used, FALSE: the period is rounded to the nearest integer.
sn	decomposition into signal and noise (2 components only). The signal is the seasonally adjusted series and the noise the seasonal component.

---

```
multiAirlineDecomposition_raw
  Title
```

---

**Description**

Title

**Usage**

```
multiAirlineDecomposition_raw(
  y,
  periods,
  ndiff = 2,
  ar = FALSE,
  stde = FALSE,
  nbcasts = 0,
  nfcasts = 0
)
```

**Arguments**

nfcasts

---

multiAirlineDecomposition\_ssf  
*Title*

---

**Description**

Title

**Usage**

```
multiAirlineDecomposition_ssf(jdecomp)
```

**Arguments**

jdecomp

---

plot.JDFractionalAirlineDecomposition  
*Plot Function for JDFractionalAirlineDecomposition Objects*

---

**Description**

This function creates a plot for the result of an Arima Model Based (AMB) decomposition of one or several frequencies (class 'JDFractionalAirlineDecomposition'). It shows the decomposition and the component of the model.

**Usage**

```
## S3 method for class 'JDFractionalAirlineDecomposition'
plot(x, from, to, type_chart = c("y-sa-trend", "cal-seas-irr"), ...)
```

**Arguments**

x	An object of class 'JDFractionalAirlineDecomposition'.
from	'Date' or 'POSIXt' object, optional starting point for x-axis.
to	'Date' or 'POSIXt' object, optional ending point for x-axis.
type_chart	Character vector specifying the type of chart to plot ("y-sa-trend", "cal-seas-irr").
...	Additional graphical parameters.

**Value**

'NULL' (invisible).

---

```
plot.JDFractionalAirlineEstimation
    Plot Function for JDFractionalAirlineEstimation Objects
```

---

**Description**

This function creates a plot for the result of fractional airline model (class ‘JDFractionalAirlineEstimation’). It shows the raw data and linearized series.

**Usage**

```
## S3 method for class 'JDFractionalAirlineEstimation'
plot(x, from, to, ...)
```

**Arguments**

x	An object of class ‘JDFractionalAirlineEstimation’.
from	‘Date’ or ‘POSIXt’ object, optional starting point for x-axis.
to	‘Date’ or ‘POSIXt’ object, optional ending point for x-axis.
...	Additional graphical parameters.

**Value**

‘NULL’ (invisible).

---

```
print.JDFractionalAirlineDecomposition
    Print method for 'JDFractionalAirlineDecomposition' objects
```

---

**Description**

This function prints informations on the result of a Fractional Airline model (classe JDFractionalAirlineDecomposition).

**Usage**

```
## S3 method for class 'JDFractionalAirlineDecomposition'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

x	An object of class ‘JDFractionalAirlineDecomposition’.
digits	Number of digits to round numerical values (default is 3 or digits - 3 from options).

**Value**

The original object 'x'.

---

```
print.JDFractionalAirlineEstimation
```

*Print method for 'JDFractionalAirlineEstimation' objects*

---

**Description**

This function prints informations on the result of a Fractional Airline model (classe JDFractionalAirlineEstimation).

**Usage**

```
## S3 method for class 'JDFractionalAirlineEstimation'  
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

x	An object of class 'JDFractionalAirlineEstimation'.
digits	Number of digits to round numerical values (default is 3 or digits - 3 from options).

**Value**

The original object 'x'.

# Index

.arima\_extract, [2](#)  
.extended\_airline\_loglevel, [5](#)  
.extended\_airline\_outliers, [5](#)  
.extendedairline\_estimation, [3](#)  
.extendedairline\_regarima, [3](#)  
.extendedairline\_spec, [4](#)  
.plot\_jd, [6](#)  
.ucm\_extract, [6](#)

fractionalAirlineDecomposition, [7](#)  
fractionalAirlineDecomposition\_raw, [7](#)  
fractionalAirlineDecomposition\_ssf, [8](#)  
fractionalAirlineEstimation, [8](#)

jd2r\_fractionalAirlineDecomposition, [9](#)  
jd2r\_multiAirlineDecomposition, [10](#)

multiAirlineDecomposition, [10](#)  
multiAirlineDecomposition\_raw, [11](#)  
multiAirlineDecomposition\_ssf, [12](#)

plot.JDFractionalAirlineDecomposition,  
[12](#)  
plot.JDFractionalAirlineEstimation, [13](#)  
print.JDFractionalAirlineDecomposition,  
[13](#)  
print.JDFractionalAirlineEstimation,  
[14](#)