

# Package: MonoInc (via r-universe)

September 18, 2024

**Type** Package

**Title** Monotonic Increasing

**Version** 1.1

**Date** 2016-05-19

**Author** Melyssa Minto, Michele Josey, and ClarLynda Williams-DeVane

**Maintainer** Michele Josey <mjosey@nccu.edu>

**Description** Various imputation methods are utilized in this package, where one can flag and impute non-monotonic data that is outside of a prespecified range.

**License** GPL-3

**Encoding** UTF-8

**Depends** compare, doParallel, foreach, iterators, parallel

**Imports** sitar

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-05-20 22:36:53

## Contents

MonoInc-package . . . . .	2
data.r . . . . .	2
decData.r . . . . .	3
mono.flag . . . . .	4
mono.range . . . . .	5
MonoInc . . . . .	6
monotonic . . . . .	7
simDEC_data . . . . .	8
simulated_data . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

MonoInc-package      *Cleans and imputes on monotonic data.*

---

### Description

The MonoInc package in R seeks to clean data so that erroneous values are less effective statistically. Given a prespecified range, MonoInc will determine if an observation is “unusual”, and then replace the value at the user’s will. MonoInc will impute on participant data individually, so that the number of time points need not be the same. MonoInc will also remove duplicate rows.

### Details

Package: MonoInc  
Type: Package  
Version: 1.1  
Date: 2016-05-19  
License: GPL-3

### Author(s)

Melyssa Minto, Michele Josey  
Maintainer: Michele Josey [mjosey@nccu.edu](mailto:mjosey@nccu.edu)

---

data.r      *Data range*

---

### Description

CDC growth chart of heights of female children aged 0 to 120 months.

### Usage

```
data("data.r")
```

### Format

A data frame with 121 observations on the following 3 variables.

Age a numeric vector

Per\_5 a numeric vector

Per\_95 a numeric vector

**Details**

Range data needed for the simulated data.

**Source**

[http://www.cdc.gov/growthcharts/clinical\\_charts.htm](http://www.cdc.gov/growthcharts/clinical_charts.htm)

**Examples**

```
data(data.r)
## plot Range boundary lines
tol <- 3
plot(data.r$Age, data.r$Per_5, type="l", lty=2, col=2)
lines(data.r$Age, data.r$Per_95, type="l", lty=2, col=2)
lines(data.r$Age, data.r$Per_5 - tol, type="l", lty=2, col=4)
lines(data.r$Age, data.r$Per_95 + tol, type="l", lty=2, col=4)
```

---

decData.r

*Data range(decreasing)*

---

**Description**

Chart of measurements of children aged 0 to 120 months

**Usage**

```
data("decData.r")
```

**Format**

A data frame with 121 observations on the following 3 variables.

Age a numeric vector

L.bound a numeric vector

U.bound a numeric vector

**Details**

Range data needed for the simulated decreasing data.

**Examples**

```
data(decData.r)

## plot Range boundary lines
tol <- 3
plot(decData.r[,1], decData.r[,2], type="l", lty=2, col=2)
lines(decData.r[,1], decData.r[,3], type="l", lty=2, col=2)
lines(decData.r[,1], decData.r[,2] - tol, type="l", lty=2, col=4)
lines(decData.r[,1], decData.r[,3] + tol, type="l", lty=2, col=4)
```

---

 mono.flag

*Flag*


---

### Description

This function flags data that is outside the prespecified range and that is not monotonic.

### Usage

```
mono.flag(data, id.col, x.col, y.col, min, max, data.r = NULL, tol = 0, direction)
```

### Arguments

data	a data.frame or matrix of measurement data
id.col	column where the id's are stored
x.col	column where x values, or time variable is stored
y.col	column where y values, or measurements are stored
min	lowest acceptable value for measurement; does not have to be a number in ycol
max	highest acceptable value for measurement; does not have to be a number in ycol
data.r	prespecified range for y values; must have three columns: 1 - must match values in xcol, 2 - lower range values, 3 - upper range values
tol	tolerance; how much outside of the range (data.r) is acceptable; same units as data in ycol
direction	the direction of the function a choice between increasing 'inc', and decreasing 'dec'

### Details

The data range (data.r) does not need to have the same number of rows as data; it only needs to include the exact time increments as xcol.

### Value

Returns the data matrix with two additional columns. "Decreasing" is a logical vector that is TRUE if the observation decreases, or causes the ID to be non-monotonic. "Outside.Range" is a logical vector that returns TRUE if the observation is outside of the data.r +/- tol range. Any duplicate rows are removed.

### Author(s)

Michele Josey [mjosey@ncu.edu](mailto:mjosey@ncu.edu) Melyssa Minto [mminto@ncu.edu](mailto:mminto@ncu.edu)

**Examples**

```

data(simulated_data)
simulated_data <- simulated_data[1:1000,]
data(data.r)
## run mono.flag function
test <- mono.flag(simulated_data, 1, 2, 3, 30, 175, data.r=data.r, direction='inc')
head(test)

```

---

mono.range	<i>Proportion in Range</i>
------------	----------------------------

---

**Description**

This function reports the proportion of entries that fall inside of the prespecified range.

**Usage**

```
mono.range(data, data.r, tol, xr.col, x.col, y.col)
```

**Arguments**

data	a data.frame or matrix of measurement data
data.r	range for y values; must have three columns: 1 - must match values in x.col, 2 - lower range values, 3 - upper range values
tol	tolerance; how much outside of the range (data.r) is acceptable; same units as data in y.col
xr.col	column where x values, or time variable is stored in data.r
x.col	column where x values, or time variable is stored in data
y.col	column where y values, or measurements are stored in data

**Value**

Returns the proportion of y values that fall inside the prespecified range

**Author(s)**

Michele Josey [mjosey@nccu.edu](mailto:mjosey@nccu.edu) Melyssa Minto [mminto@nccu.edu](mailto:mminto@nccu.edu)

**Examples**

```

data(simulated_data)
data(data.r)

mono.range(simulated_data, data.r, tol=4, xr.col=1 ,x.col=2, y.col=3)

```

---

MonoInc *Monotonic Increasing*

---

### Description

Combines many of the functions in the MonoInc package. Given a data range, weights, and imputation methods of choice, MonoInc will impute flagged values using either one or a combination of two imputation methods. It can also perform all single imputation methods for comparison.

### Usage

```
MonoInc(data, id.col, x.col, y.col, data.r = NULL, tol = 0, direction = "inc", w1 = 0.5,
        min, max, impType1 = "nn", impType2 = "reg", sum = FALSE)
```

### Arguments

data	a data.frame or matrix of measurement data
id.col	column where the id's are stored
x.col	column where x values, or time variable is stored
y.col	column where y values, or measurements are stored
data.r	range for y values; must have three columns: 1 - must match values in xcol, 2 - lower range values, 3 - upper range values
tol	tolerance; how much outside of the range (data.r) is acceptable; same units as data in ycol
direction	the direction of the function a choice between increasing 'inc', and decreasing 'dec'
w1	weight of imputation type 1 (impType1); default is 0.50
min	lowest acceptable value for measurement; does not have to be a number in ycol
max	highest acceptable value for measurement; does not have to be a number in ycol
impType1	imputation method 1, a choice between Nearest Neighbor "nn", Regression "reg", Fractional Regression "fr", Last Observation Carried Forward "locf", or Last & Next "ln"; default is "nn"
impType2	imputation method 2; default is "reg"
sum	if true the function will return a matrix of all imputation methods in the package

### Details

If two imputation methods are chosen, MonoInc will take a weighted average of the output of the imputed values. User must chose one or two imputation methods or sum=TRUE for a comparison. If there are not enough values available to impute missing or erroneous values, MonoInc will return an NA. Advice: Do NOT overwrite original data using this function! Use parallel processing if available on your device.

**Value**

Returns the data matrix with additional columns for the selected imputation method. If `sum=TRUE`, it will return a column for each single imputation method. The Y column will have NAs, indicating that this observation was flagged and imputed, for summary only. Duplicate rows are removed.

**Author(s)**

Michele Josey [mjosey@ncu.edu](mailto:mjosey@ncu.edu) Melyssa Minto [mminto@ncu.edu](mailto:mminto@ncu.edu)

**Examples**

```
data(simulated_data)
simulated_data <- simulated_data[1:1000,]
data(data.r)
library(sitar)

## Run MonoInc
sum <- MonoInc(simulated_data, 1,2,3, data.r,5,direction='inc', w1=0.3, min=30, max=175,
  impType1=NULL, impType2=NULL, sum=TRUE)
head(sum)
test <- MonoInc(simulated_data, 1,2,3, data.r,5,direction='inc', w1=0.3, min=30, max=175,
  impType1="nn", impType2="fr")
head(test)

## plot longitudinal height for each id
mplot(x=X, y=Nn.Fr, data=test)
tol <- 5
lines(data.r[,1], data.r[,2]-tol, col=2, lty=2)
lines(data.r[,1], data.r[,3]+tol, col=2, lty=2)
```

---

monotonic

*Monotonic Check*

---

**Description**

This function can check the monotonicity of a single vector, matrix, or data.frame that has multiple IDs within the matrix or data.frame.

**Usage**

```
monotonic(data, id.col=NULL, y.col=NULL, direction)
```

**Arguments**

<code>data</code>	a data.frame or matrix or vector of measurement data
<code>id.col</code>	column where the id's are stored; default is NULL
<code>y.col</code>	column where y values, or measurements are stored; default is NULL
<code>direction</code>	the direction of the function a choice between increasing 'inc', and decreasing 'dec'

**Value**

If the user enters a vector, the function returns TRUE or FALSE as to where that particular vector is monotonic increasing or not, it returns NA if the vector has missing values. If the user enters a matrix or data frame, the function returns a matrix with 2 columns. The first column as the id. The second column as a 0 for FALSE and 1 for TRUE as to where the data in that particular id is monotonic increasing or not, or NA if the y column has missing values in that particular id.

**Author(s)**

Michele Josey [mjosey@nccu.edu](mailto:mjosey@nccu.edu) Melyssa Minto [mminto@nccu.edu](mailto:mminto@nccu.edu)

**Examples**

```
data(simulated_data)
## Run monotonic
test <- monotonic(simulated_data, 1,3, direction='inc')

## look at the number of ids that are non-monotonic
table(as.logical(test[,2]))

##to ignore NA values
x<-c(1,2,3,5,NA,7,8)
monotonic(na.omit(x), direction='inc')
```

---

simDEC\_data

*Simulated Decreasing Data*

---

**Description**

This data was simulated to be monotonically decreasing. There are 500 individuals, with a random number of data points. Each individual has a two-level random effect (intercept and slope), a common intercept, and a random error term. The ages range from 0 to 10 years, which is given in months.

**Usage**

```
data("simDEC_data")
```

**Format**

A data frame with 5505 observations on the following 3 variables.

id a numeric vector of the identification number of each individual

age a numeric vector of the age in months

y a numeric vector of measurements



**References**

<http://blog.stata.com/2014/07/18/how-to-simulate-multilevellongitudinal-data/>

**Examples**

```
data(simDEC_data)
library(sitar)

mplot(x=age, y=y, id=id, data=simDEC_data, col=id, main="Individual Measurement Curves")
```

---

simulated_data	<i>Simulated Data</i>
----------------	-----------------------

---

**Description**

This data was simulated to imitate height growth of female children in electronic medical records. There are 500 individuals, with a random number of data points. Based on the CDC growth curve, each individual has a two-level random effect (intercept and slope), a common intercept, and a random error term. The ages range from 0 to 10 years, which is given in months.

**Usage**

```
data("simulated_data")
```

**Format**

A data frame with 5673 observations on the following 3 variables.

nestid a numeric vector of the identification number of each individual

age a numeric vector of the age in months

height a numeric vector of the height in centimeters

**References**

<http://blog.stata.com/2014/07/18/how-to-simulate-multilevellongitudinal-data/>

**Examples**

```
data(simulated_data)
library(sitar)

## plot each individual growth curve
mplot(x=age, y=height, id=nestid, data=simulated_data, col=nestid, main="Growth Curves")
```

# Index

- \* **CDC**
    - data.r, 2
  - \* **MonoInc**
    - MonoInc-package, 2
  - \* **datasets**
    - data.r, 2
    - decData.r, 3
    - simDEC\_data, 8
    - simulated\_data, 9
  - \* **flag**
    - mono.flag, 4
  - \* **monotonic**
    - MonoInc, 6
    - MonoInc-package, 2
  - \* **mono**
    - MonoInc, 6
    - MonoInc-package, 2
    - monotonic, 7
  - \* **package**
    - MonoInc-package, 2
  - \* **range**
    - mono.range, 5
  - \* **simulated**
    - simulated\_data, 9
- data.r, 2  
decData.r, 3
- mono.flag, 4  
mono.range, 5  
MonoInc, 6  
MonoInc-package, 2  
monotonic, 7
- simDEC\_data, 8  
simulated\_data, 9