

# Package ‘AGD’

May 6, 2026

**Type** Package

**Version** 0.45.0

**Title** Analysis of Growth Data

**Date** 2024-05-03

**Maintainer** Stef van Buuren <stef.vanbuuren@tno.nl>

**Depends** R (>= 2.10.0)

**Imports** gamlss, gamlss.dist

**Description** Tools for the analysis of growth data: to extract an LMS table from a gamlss object, to calculate the standard deviation scores and its inverse, and to superpose two wormplots from different models. The package contains a some varieties of reference tables, especially for The Netherlands.

**License** GPL-2 | GPL-3

**LazyLoad** yes

**LazyData** yes

**URL** <https://stefvanbuuren.name/AGD/>

**BugReports** <https://github.com/stefvanbuuren/AGD/issues>

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Stef van Buuren [aut, cre]

**Repository** CRAN

**Date/Publication** 2024-05-03 15:00:05 UTC

## Contents

ageGrid . . . . .	2
boys7482 . . . . .	3
extractLMS . . . . .	4

References CDC . . . . .	5
References NL3 . . . . .	7
References NL4 . . . . .	8
References WHO . . . . .	9
wp.twin . . . . .	10
y2z . . . . .	12
z2y . . . . .	15
<b>Index</b>	<b>19</b>

---

ageGrid	<i>Creates an age grid according to a specified format.</i>
---------	---

---

## Description

Creates an age grid according to a specified format.

## Usage

```
ageGrid(grid = "compact")
```

## Arguments

grid	A character string specifying one of the following: "compact", "classic", "extensive", "0-104w", "0-24m", "0-21y", "0-21yd" or "0-21yc". The default is "compact", which produces an age grid between 0 and 21 years with 95 points.
------	--

## Value

A list with five components: format, year, month, week and day containing the age grid in different units.

## Author(s)

Stef van Buuren, 2010

## Examples

```
age <- ageGrid("classic")$year
```

---

boys7482

*Growth of Dutch boys*

---

### Description

Height, weight, head circumference and puberty of 7482 Dutch boys.

### Format

A data frame with 7482 rows on the following 9 variables:

**age** Decimal age (0-21 years)

**hgt** Height (cm)

**wgt** Weight (kg)

**bmi** Body mass index

**hc** Head circumference (cm)

**gen** Genital Tanner stage (G1-G5)

**phb** Pubic hair (Tanner P1-P6)

**tv** Testicular volume (ml)

**reg** Region (north, east, west, south, city)

### Details

The complete sample of cross-sectional data from boys 0-21 years used to construct the Dutch growth references 1997. Variables `gen` and `phb` are ordered factors. `reg` is a factor. Note: A 10% sample from this data is available in data set `boys` in the `mice` package.

### Author(s)

Stef van Buuren, 2012

### Source

Fredriks, A.M., van Buuren, S., Burgmeijer, R.J., Meulmeester JF, Beuker, R.J., Brugman, E., Roede, M.J., Verloove-Vanhorick, S.P., Wit, J.M. (2000) Continuing positive secular growth change in The Netherlands 1955-1997. *Pediatric Research*, **47**, 316-323.

Fredriks, A.M., van Buuren, S., Wit, J.M., Verloove-Vanhorick, S.P. (2000). Body index measurements in 1996-7 compared with 1980. *Archives of Disease in Childhood*, **82**, 107-112.

---

 extractLMS

*Extracts LMS values from a gamlss object.*


---

### Description

Extract LMS values from a gamlss object for solutions that transform the age axis according to the M-curve.

### Usage

```
extractLMS(
  fit,
  data,
  sex = "M",
  grid = "classic",
  decimals = c(4, 4, 4),
  flatAge = NULL
)
```

### Arguments

<code>fit</code>	A gamlss object containing the final fit on transformed age, <code>t.age</code> .
<code>data</code>	A data frame containing the original data, with both age and <code>t.age</code>
<code>sex</code>	A character vector indicating whether the fit applied to males <code>sex="M"</code> or females <code>sex="F"</code> . The default is <code>sex="M"</code> .
<code>grid</code>	A character vector indicating the desired age grid. See <code>ageGrid()</code> for possible options. The default is a <code>grid="classic"</code> , a grid of 59 age points.
<code>decimals</code>	A numerical vector of length 3 indicating the number of significant digits for rounding of the L, M and S curves, respectively.
<code>flatAge</code>	A scalar indicating the age beyond which the L, M and S values should be constant. The default (NULL) is not to flatten the curves.

### Details

It is crucial that `t.age` in `data` correspond to exactly the same age transformation as used to fit the gamlss object. Age grid values beyond the range of `data$age` produce NA in the L, M and S values. Parameter `flatAge` should be one of the values of the age grid.

### Value

A data frame with rows corresponding to time points, and with the following columns: `sex,x,L,M,S`.

### Author(s)

Stef van Buuren, 2010

**Examples**

```
## Not run:
#
library(gamlss)
boys <- boys7482

# calculate initial M curve
data <- na.omit(boys[,1:2])
f0154 <- gamlss(hgt~cs(age,df=15,c.spar=c(-1.5,2.5)),
               sigma.formula=~cs(age,df=4,c.spar=c(-1.5,2.5)),
               data=data,family=NO,
               control=gamlss.control(n.cyc=3))

# calculate transformed age
t.age <- fitted(lm(data$age~fitted(f0154)))
t.age <- t.age - min(t.age)
data.t <- data.frame(data,t.age=t.age)

# calculate final solution
f0106r <- gamlss(hgt~cs(t.age,df=10,c.spar=c(-1.5,2.5)),
                sigma.formula=~cs(t.age,df=6,c.spar=c(-1.5,2.5)),
                data=data.t,family=NO,
                control=gamlss.control(n.cyc=3))

# extract the LMS reference table in the 'classic' age grid
n14.hgt.boys <- extractLMS(fit = f0106r, data=data.t, grid="compact",
                          dec = c(0,2,5))
n14.hgt.boys

# flatten the reference beyond age 20Y (not very useful in this data)
n14.hgt.boys.flat <- extractLMS(fit = f0106r, data=data.t, flatAge=20)
n14.hgt.boys.flat

# use log age transformation
data.t <- data.frame(data, t.age = log(data$age))
f0106rlog <- gamlss(hgt~cs(t.age,df=10,c.spar=c(-1.5,2.5)),
                  sigma.formula=~cs(t.age,df=6,c.spar=c(-1.5,2.5)),
                  data=data.t,family=NO,
                  control=gamlss.control(n.cyc=1))

n14.hgt.boys.log <- extractLMS(fit = f0106rlog, data=data.t)
n14.hgt.boys.log

## End(Not run)
```

**Description**

Reference tables from CDC 2000

**Format**

A data frame with seven variables:

**list("pop")** Study Population

**list("sub")** Subpopulation

**list("sex")** Sex (M,F)

**list("x")** Decimal age (0-5 years)

**list("L")** Lambda (skewness) curve

**list("M")** Median curve

**list("S")** Coefficient of Variation curve

**Details**

The models were fitted by the LMS model. Parameters are stored as type LMS. Tabulated values are point ages.

The naming conventions are as follows:

**list("cdc.hgt")** Combined length/height (cm) for Age, 0-20 years. Measures <2 years apply to length (lying), while ages  $\geq 2$  years apply to height, or stature (standing).

**list("cdc.wgt")** Weight (kg) for Age, 0-20 years.

**list("cdc.bmi")** Body Mass Index (kg/m<sup>2</sup>) for Age, 2-20 years.

**Source**

Kuczmariski RJ, Ogden CL, Guo SS, Grummer-Strawn LM, Flegal KM, Mei Z, Wei R, Curtin LR, Roche AF, Johnson CL. 2000 CDC growth charts for the United States: methods and development. *Vital Health Stat*, 2002, **11**, 246, 1-190.

**See Also**

[n14.wgt](#), [n14.hgt](#), [n14.bmi](#), [who.wgt](#)

**Description**

Reference table from the Third Dutch Growth Study 1980

**Format**

A data frame with seven variables:

**list("pop")** Study Population

**list("sub")** Subpopulation, e.g. ethnicity or age group (for nl4.wfh)

**list("sex")** Sex (M,F)

**list("x")** Decimal age (0-21 years) or Height (for nl4.wfh)

**list("L")** Lambda (skewness) curve

**list("M")** Median curve

**list("S")** Coefficient of Variation curve

**Details**

The model was fitted by the LMS model. Parameters are stored as type LMS. Tabulated values are point ages.

Height follows a normal distribution, with all lambda parameters set equal to 1. The standard deviation (in cm) is obtained as  $S \cdot M$ .

The naming conventions are as follows:

**list("nl4.hgt")** Length/Height (cm) for Age

**list("nl4.wgt")** Weight (kg) for Age

**list("nl4.wfh")** Weight (kg) for Height (cm)

**list("nl4.bmi")** Head circumference (cm) for Age

**list("nl4.lgl")** Leg Length (cm) for Age

**list("nl4.hip")** Hip circumference (cm) for Age

**list("nl4.wst")** Waist circumference (cm) for Age

**list("nl4.whr")** Waist/Hip ratio for Age

**list("nl4.sit")** Sitting Height for Age

**list("nl4.shh")** Sitting Height/Height ratio for Age

**Source**

Fredriks, A.M., van Buuren, S., Burgmeijer, R.J., Meulmeester JF, Beuker, R.J., Brugman, E., Roede, M.J., Verloove-Vanhorick, S.P., Wit, J.M. (2000) Continuing positive secular growth change in The Netherlands 1955-1997. *Pediatric Research*, **47**, 316-323.

Fredriks, A.M., van Buuren, S., Wit, J.M., Verloove-Vanhorick, S.P. (2000). Body index measurements in 1996-7 compared with 1980. *Archives of Disease in Childhood*, **82**, 107-112.

**See Also**

[cdc.wgt](#), [who.wgt](#)

---

 References NL4

*Reference tables from Fourth Dutch Growth Study 1997*


---

**Description**

Reference table from the Fourth Dutch Growth Study 1997

**Format**

A data frame with seven variables:

**list("pop")** Study Population

**list("sub")** Subpopulation, e.g. ethnicity or age group (for `nl4.wfh`)

**list("sex")** Sex (M,F)

**list("x")** Decimal age (0-21 years) or Height (for `nl4.wfh`)

**list("L")** Lambda (skewness) curve

**list("M")** Median curve

**list("S")** Coefficient of Variation curve

**Details**

The model was fitted by the LMS model. Parameters are stored as type LMS. Tabulated values are point ages.

Height follows a normal distribution, with all lambda parameters set equal to 1. The standard deviation (in cm) is obtained as  $S \cdot M$ .

The naming conventions are as follows:

**list("nl4.hgt")** Length/Height (cm) for Age

**list("nl4.wgt")** Weight (kg) for Age

**list("nl4.wfh")** Weight (kg) for Height (cm)

**list("nl4.bmi")** Head circumference (cm) for Age

**list("nl4.lgl")** Leg Length (cm) for Age

**list("nl4.hip")** Hip circumference (cm) for Age

**list("nl4.wst")** Waist circumference (cm) for Age

**list("nl4.whr")** Waist/Hip ratio for Age

**list("nl4.sit")** Sitting Height for Age

**list("nl4.shh")** Sitting Height/Height ratio for Age

**Source**

Fredriks, A.M., van Buuren, S., Burgmeijer, R.J., Meulmeester JF, Beuker, R.J., Brugman, E., Roede, M.J., Verloove-Vanhorick, S.P., Wit, J.M. (2000) Continuing positive secular growth change in The Netherlands 1955-1997. *Pediatric Research*, **47**, 316-323.

Fredriks, A.M., van Buuren, S., Wit, J.M., Verloove-Vanhorick, S.P. (2000). Body index measurements in 1996-7 compared with 1980. *Archives of Disease in Childhood*, **82**, 107-112.

**See Also**

[cdc.wgt](#), [who.wgt](#)

---

References WHO

*References WHO*

---

**Description**

Reference tables, combined from the WHO Multicentre Growth Reference Study (MGRS) (ages 0-5 years) and the WHO 2007 reference (5-19 years).

**Format**

A data frame with seven variables:

**pop** Study Population (always "who")

**sub** Subpopulation (always "N")

**sex** Sex (M, F)

**x** Decimal age, height (cm) or length(cm)

**L** Lambda (skewness) curve

**M** Median curve

**S** Coefficient of variation

**Details**

The data were fitted by the LMS model. Parameters are stored as type LMS. Tabulated values are point ages.

The naming conventions are as follows:

**who.hgt** Length (cm, 0-2 Yrs) or height (cm, 2-19 years)

**who.wgt** Weight (kg) for age (0-10 years)

**who.bmi** BMI (kg/m<sup>2</sup>) for age (0-19 years)

**who.hdc** Head circumference (cm) for age (0-5 years)

**who.wfh** Weight (kg) for height (65-120 cm)

**who.wfl** Weight (kg) for length (45-110 cm)

**Source**

WHO Multicentre Growth Reference Study Group. WHO Child Growth Standards based on length/height, weight and age. *Acta Paediatr*, Suppl. 2006, 450, 76-85.

de Onis M, Onyango AW, Borghi E, Siyam A, Nishida C, Siekmann J. Development of a WHO growth reference for school-aged children and adolescents *Bulletin of the World Health Organization*, 2007;85:660-7.

**See Also**

[nl4.wgt](#), [cdc.wgt](#)

---

wp.twin

*Superposes two worm plots*

---

**Description**

Superposes two worm plots from GAMLSS fitted objects. This is a diagnostic tool for comparing two solutions.

**Usage**

```
wp.twin(
  obj1,
  obj2 = NULL,
  xvar = NULL,
  xvar.column = 2,
  n.inter = 16,
  show.given = FALSE,
  ylim.worm = 0.5,
  line = FALSE,
  cex = 1,
  col1 = "black",
  col2 = "orange",
  warnings = FALSE,
  ...
)
```

**Arguments**

obj1	a GAMLSS fitted object
obj2	an optional second GAMLSS fitted object
xvar	the explanatory variable against which the worm plots will be plotted
xvar.column	the number referring to the column of $\text{obj1}\$mu.x$ and $\text{obj2}\$mu.x$ . If $xvar=NULL$ then the explanatory variable is set to $xvar=\text{obj1}\$mu.x[,xvar.column]$ respectively $xvar=\text{obj2}\$mu.x[,xvar.column]$ . The default is $xvar.column=2$ , which selects the variable following the intercept (which is typically age in most applications).

n.inter	the number of intervals in which the explanatory variable xvar will be cut. The default is 16.
show.given	whether to show the x-variable intervals in the top of the graph, default is show.given=FALSE
ylim.worm	for multiple plots, this values is the y-variable limit, default value is ylim.worm=0.5
line	whether to plot the polynomial line in the worm plot, default value is line=FALSE
cex	the cex plotting parameter with default cex=1
col1	the color for the points of obj1. The default col="black"
col2	the color for the points of obj2. The default col="orange"
warnings	a logical indicating whether warnings should be produced. The default warnings=FALSE
...	for extra arguments, overlap, xlim.worm or pch

### Details

This function is a customized version of the `wp()` function found in the `gamlss` package. Function `wp.twin()` allows overplotting of two worm plots, each in its own color. The points of `obj1` are plotted first, the points of `obj2` are superposed. This twin worm plot provide a visual assessment of the differences between the solutions. Extra arguments can be specified (e.g. `xvar`) that are passed down to the `wp()` function of `gamlss` if specified. The worm plot is a detrended normal QQ-plot that highlight departures from normality.

Argument `xvar` takes priority over `xvar.column`. The `xvar` variable is cut into `n.inter` intervals with an equal number observations and detrended normal QQ (i.e. worm) plots for each interval are plotted. This is a way of highlighting failures of the model within different ranges of the explanatory variable.

If `line=TRUE` and `n.inter>1`, the fitted coefficients from fitting cubic polynomials to the residuals (within each x-variable interval) can be obtain by e.g. `coeffs<-wp.twin(model1, xvar=x, n.inter=9)`. van Buuren *et al.* (2001) used these residuals to identify regions (intervals) of the explanatory variable within which the model does not fit adequately the data (called "model violation")

### Value

For multiple plots the `xvar` intervals and the coefficients of the fitted cubic polynomials to the residuals (within each `xvar` interval) are returned.

### Author(s)

Stef van Buuren, using R code of Mikis Stasinopoulos and Bob Rigby

### References

- Stasinopoulos D. M. Rigby R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, Vol. **23**, Issue 7, Dec 2007, <https://www.jstatsoft.org/v23/i07>.
- van Buuren and Fredriks M. (2001) Worm plot: simple diagnostic device for modelling growth reference curves. *Statistics in Medicine*, **20**, 1259–1277.
- van Buuren and Fredriks M. (2007) Worm plot to diagnose fit in quantile regression. *Statistical Modelling*, **7**, 4, 363–376.

## Examples

```
## Not run:
library(gamlss)
data(abdom)
a <- gamlss(y~cs(x,df=1),sigma.fo=~cs(x,0),family=L0,data=abdom)
b <- gamlss(y~cs(x,df=3),sigma.fo=~cs(x,1),family=L0,data=abdom)
coeff1 <- wp.twin(a,b,line=TRUE)

## End(Not run)
```

---

y2z

*Converts measurements to standard deviation scores (SDS)*


---

## Description

Converts measurements into age- and sex-conditional standard deviation score (SDS) using an external reference.

## Usage

```
y2z(
  y = c(75, 80, 85),
  x = 1,
  sex = "M",
  sub = "N",
  ref = get("n14.hgt"),
  dist = "LMS",
  dec = 3,
  sex.fallback = NA,
  sub.fallback = NA,
  tail.adjust = FALSE
)
```

## Arguments

y	A numerical vector containing the outcome measurements. The length <code>length(y)</code> determines the size of the output vector.
x	A vector containing the values of the numerical covariate (typically decimal age or height) at which conversion is desired. Values are replicated to match <code>length(y)</code> .
sex	A character vector indicating whether the male ("M") or female ("F") reference should be used. Values are replicated to match <code>length(y)</code> .
sub	A character vector indicating the level of the sub field of the reference standard defined in <code>ref</code>
ref	A data frame containing a factor <code>sex</code> , a numerical variable <code>age</code> containing the tabulated decimal point ages, and two or more numerical variables with reference values. See details.

<code>dist</code>	A string identifying the type of distribution. Values values are: "NO", "BCCG", "LMS", "BCPE" and "BCT". The default is "LMS".
<code>dec</code>	A scalar value indicating the number of decimals used to round the value.
<code>sex.fallback</code>	The level of the sex field used when no match is found. The default <code>sex.fallback=NA</code> specifies that unmatched entries should receive a NA value.
<code>sub.fallback</code>	The level of the sub field used when no match is found. The default <code>sub.fallback=NA</code> specifies that unmatched entries should receive a NA value.
<code>tail.adjust</code>	Logical. If TRUE then the WHO method for tail adjustment is applied. The default is FALSE.

### Details

Functions `z2y()` and `y2z()` are the inverse of each other.

The argument `dist` determines the statistical distribution. The possibilities are as follows:

**list("\NO\")** `ref` should contain columns `mean` and `sd`, containing the mean and the standard deviation in the external reference population.

**list("\LMS\")** `ref` should contain columns `L`, `S` and `M` containing the LMS parameters.

**list("\BCCG\")** `ref` should contain columns `mu`, `sigma` and `nu` containing the Box-Cox Cole-Green parameters.

**list("\BCPE\")** `ref` should contain columns `mu`, `sigma`, `nu` and `tau` containing the Box-Cox Power Exponential parameters.

**list("\BCT\")** `ref` should contain columns `mu`, `sigma`, `nu` and `tau` containing the Box-Cox T distribution parameters.

### Value

For `y2z()`: A vector with `length(y)` elements containing the standard deviation score. For `z2y()`: A vector with `length(z)` elements containing quantiles.

### Author(s)

Stef van Buuren, 2010

### See Also

[z2y](#)

### Examples

```
boys <- boys7482

# SDS of height 115 cm at age 5 years,
# relative to Dutch boys reference
y2z(y=115, x=5)

# same relative to Dutch girls
```

```

y2z(y=115, x=5, sex="F")

# SDS of IOTF BMI cut-off value for overweight (boys 2-18)
# relative to Dutch boys reference
cutoff <- c(
18.41, 18.15, 17.89, 17.72, 17.55, 17.49, 17.42, 17.49, 17.55, 17.74,
17.92, 18.18, 18.44, 18.77, 19.10, 19.47, 19.84, 20.20, 20.55, 20.89,
21.22, 21.57, 21.91, 22.27, 22.62, 22.96, 23.29, 23.60, 23.90, 24.18,
24.46, 24.73, 25.00)
age <- seq(2, 18, by=0.5)
(z <- y2z(y=cutoff, x=age, sex="M", ref=nl4.bmi))

# apply inverse transformation to check calculations
round(z2y(z, age, ref=nl4.bmi), 2)
cutoff

# calculate percentiles of weight 12 kg at 2 years (boys, girls)
100*round(pnorm(y2z(y=c(12,12), x=2, sex=c("M","F"), ref=nl4.wgt)),2)

# # percentage of children lighter than 15kg at ages 2-5
e <- expand.grid(age=2:5, sex=c("M","F"))
z <- y2z(y=rep(15,nrow(e)), x=e$age, sex=e$sex, ref=nl4.wgt)
w <- matrix(100*round(pnorm(z),2), nrow=2, byrow=TRUE)
dimnames(w) <- list(c("boys","girls"),2:5)
w

# analysis in Z scale
hgt.z <- y2z(y=boys$hgt, x=boys$age, sex="M", ref=nl4.hgt)
wgt.z <- y2z(y=boys$wgt, x=boys$age, sex="M", ref=nl4.wgt)
plot(hgt.z, wgt.z, col="blue")

# z2y

# quantile at SD=0 of age 2 years,
# height Dutch boys
z2y(z=0, x=2)

# same for Dutch girls
z2y(z=0, x=2, sex="F")

# quantile at SD=c(-1,0,1) of age 2 years, BMI Dutch boys
z2y(z=c(-1,0,+1), x=2, ref=nl4.bmi)

# 0SD line (P50) in kg of weight for age in 5-10 year, Dutch boys
z2y(z=rep(0,6), x=5:10, ref=nl4.wgt)

# 95th percentile (P95), age 10 years, wfa, Dutch boys
z2y(z=qnorm(0.95), x=10, ref=nl4.wgt)

# table of P3, P10, P50, P90, P97 of weight for 5-10 year old dutch boys
# age per year
age <- 5:10

```

```

p <- c(0.03,0.1,0.5,0.9,0.97)
z <- rep(qnorm(p), length(age))
x <- rep(age, each=length(p))
w <- matrix(z2y(z, x=x, sex="M", ref=n14.wgt), ncol=length(p),
  byrow=TRUE)
dimnames(w) <- list(age, p)
round(w,1)

# standard set of Z-scores of weight for all tabulated ages, boys & girls
# and three ethnicities
sds <- c(-2.5, -2, -1, 0, 1, 2, 2.5)
age <- n14.wgt$x
z <- rep(sds, times=length(age))
x <- rep(age, each=length(sds))
sex <- rep(c("M","F"), each=length(z)/2)
w <- z2y(z=z, x=x, sex=sex, ref=n14.wgt)
w <- matrix(w, ncol=length(sds), byrow=TRUE)
dimnames(w) <- list(age, sds)
data.frame(sub=n14.wgt$sub,sex=n14.wgt$sex,round(w,2), row.names=NULL)

# P85 of BMI in 5-8 year old Dutch boys and girls
e <- expand.grid(age=5:8, sex=c("M","F"))
w <- z2y(z=rep(qnorm(0.85),nrow(e)), x=e$age, sex=e$sex, ref=n14.bmi)
w <- matrix(w, nrow=2, byrow=TRUE)
dimnames(w) <- list(c("boys","girls"),5:8)
w

# data transformation of height z-scores to cm-scale
z <- c(-1.83, 0.09, 2.33, 0.81, -1.20)
x <- c(8.33, 0.23, 19.2, 24.3, 10)
sex <- c("M", "M", "F", "M", "F")
round(z2y(z=z, x=x, sex=sex, ref=n14.hgt), 1)

# interpolate published height standard
# to daily values, days 0-31, boys
# on centiles -2SD, 0SD and +2SD
days <- 0:31
sds <- c(-2, 0, +2)
z <- rep(sds, length(days))
x <- rep(round(days/365.25,4), each=length(sds))
w <- z2y(z, x, sex="M", ref=n14.hgt)
w <- matrix(w, ncol=length(sds), byrow=TRUE)
dimnames(w) <- list(days, sds)
w

```

**Description**

Converts standard deviation score (SDS) into measurements using an age- and sex-conditional external reference.

**Usage**

```
z2y(
  z = c(-2, 0, 2),
  x = 1,
  sex = "M",
  sub = "N",
  ref = get("n14.hgt"),
  dist = "LMS",
  dec = 3,
  sex.fallback = NA,
  sub.fallback = NA
)
```

**Arguments**

z	A numerical vector containing standard deviation scores that are to be converted. The length <code>length(z)</code> determines the size of the output vector.
x	A vector containing the values of the numerical covariate (typically decimal age or height) at which conversion is desired. Values are replicated to match <code>length(y)</code> .
sex	A character vector indicating whether the male ("M") or female ("F") reference should be used. Values are replicated to match <code>length(y)</code> .
sub	A character vector indicating the level of the sub field of the reference standard defined in <code>ref</code>
ref	A data frame containing a factor <code>sex</code> , a numerical variable <code>age</code> containing the tabulated decimal point ages, and two or more numerical variables with reference values. See details.
dist	A string identifying the type of distribution. Values values are: "NO", "BCCG", "LMS", "BCPE" and "BCT". The default is "LMS".
dec	A scalar value indicating the number of decimals used to round the value.
sex.fallback	The level of the sex field used when no match is found. The default <code>sex.fallback=NA</code> specifies that unmatched entries should receive a NA value.
sub.fallback	The level of the sub field used when no match is found. The default <code>sub.fallback=NA</code> specifies that unmatched entries should receive a NA value.

**Details**

Functions `z2y()` and `y2z()` are the inverse of each other.

The argument `dist` determines the statistical distribution. The possibilities are as follows:

**list("\NO\")** `ref` should contain columns `mean` and `sd`, containing the mean and the standard deviation in the external reference population.

- list("\LMS\")** ref should contain columns L, S and M containing the LMS parameters.
- list("\BCCG\")** ref should contain columns mu, sigma and nu containing the Box-Cox Cole-Green parameters.
- list("\BCPE\")** ref should contain columns mu, sigma, nu and tau containing the Box-Cox Power Exponential parameters.
- list("\BCT\")** ref should contain columns mu, sigma, nu and tau containing the Box-Cox T distribution parameters.

### Value

For `y2z()`: A vector with `length(y)` elements containing the standard deviation score. For `z2y()`: A vector with `length(z)` elements containing quantiles.

### Author(s)

Stef van Buuren, 2010

### See Also

[y2z](#)

### Examples

```

boys <- boys7482

# quantile at SD=0 of age 2 years,
# height Dutch boys
z2y(z=0, x=2)

# same for Dutch girls
z2y(z=0, x=2, sex="F")

# quantile at SD=c(-1,0,1) of age 2 years, BMI Dutch boys
z2y(z=c(-1,0,+1), x=2, ref=n14.bmi)

# 0SD line (P50) in kg of weight for age in 5-10 year, Dutch boys
z2y(z=rep(0,6), x=5:10, ref=n14.wgt)

# 95th percentile (P95), age 10 years, wfa, Dutch boys
z2y(z=qnorm(0.95), x=10, ref=n14.wgt)

# table of P3, P10, P50, P90, P97 of weight for 5-10 year old dutch boys
# age per year
age <- 5:10
p <- c(0.03,0.1,0.5,0.9,0.97)
z <- rep(qnorm(p), length(age))
x <- rep(age, each=length(p))
w <- matrix(z2y(z, x=x, sex="M", ref=n14.wgt), ncol=length(p),
  byrow=TRUE)
dimnames(w) <- list(age, p)

```

```

round(w,1)

# standard set of Z-scores of weight for all tabulated ages, boys & girls
# and three ethnicities
sds <- c(-2.5, -2, -1, 0, 1, 2, 2.5)
age <- nl4.wgt$x
z <- rep(sds, times=length(age))
x <- rep(age, each=length(sds))
sex <- rep(c("M","F"), each=length(z)/2)
w <- z2y(z=z, x=x, sex=sex, ref=nl4.wgt)
w <- matrix(w, ncol=length(sds), byrow=TRUE)
dimnames(w) <- list(age, sds)
data.frame(sub=nl4.wgt$sub,sex=nl4.wgt$sex,round(w,2), row.names=NULL)

# P85 of BMI in 5-8 year old Dutch boys and girls
e <- expand.grid(age=5:8, sex=c("M","F"))
w <- z2y(z=rep(qnorm(0.85),nrow(e)), x=e$age, sex=e$sex, ref=nl4.bmi)
w <- matrix(w, nrow=2, byrow=TRUE)
dimnames(w) <- list(c("boys","girls"),5:8)
w

# data transformation of height z-scores to cm-scale
z <- c(-1.83, 0.09, 2.33, 0.81, -1.20)
x <- c(8.33, 0.23, 19.2, 24.3, 10)
sex <- c("M", "M", "F", "M", "F")
round(z2y(z=z, x=x, sex=sex, ref=nl4.hgt), 1)

# interpolate published height standard
# to daily values, days 0-31, boys
# on centiles -2SD, 0SD and +2SD
days <- 0:31
sds <- c(-2, 0, +2)
z <- rep(sds, length(days))
x <- rep(round(days/365.25,4), each=length(sds))
w <- z2y(z, x, sex="M", ref=nl4.hgt)
w <- matrix(w, ncol=length(sds), byrow=TRUE)
dimnames(w) <- list(days, sds)
w

```

# Index

## \* datasets

boys7482, [3](#)  
References CDC, [5](#)  
References NL3, [7](#)  
References NL4, [8](#)  
References WHO, [9](#)

## \* distribution

ageGrid, [2](#)  
extractLMS, [4](#)  
y2z, [12](#)  
z2y, [15](#)

## \* smooth

wp.twin, [10](#)

ageGrid, [2](#)

boys7482, [3](#)

cdc.bmi (References CDC), [5](#)  
cdc.hgt (References CDC), [5](#)  
cdc.wgt, [8–10](#)  
cdc.wgt (References CDC), [5](#)

extractLMS, [4](#)

n13.bmi (References NL3), [7](#)  
n14.bmi, [6](#)  
n14.bmi (References NL4), [8](#)  
n14.hdc (References NL4), [8](#)  
n14.hgt, [6](#)  
n14.hgt (References NL4), [8](#)  
n14.hip (References NL4), [8](#)  
n14.lgl (References NL4), [8](#)  
n14.shh (References NL4), [8](#)  
n14.sit (References NL4), [8](#)  
n14.wfh (References NL4), [8](#)  
n14.wgt, [6, 10](#)  
n14.wgt (References NL4), [8](#)  
n14.whr (References NL4), [8](#)  
n14.wst (References NL4), [8](#)

References CDC, [5](#)  
References NL3, [7](#)  
References NL4, [8](#)  
References WHO, [9](#)

who.bmi (References WHO), [9](#)  
who.hdc (References WHO), [9](#)  
who.hgt (References WHO), [9](#)  
who.wfh (References WHO), [9](#)  
who.wfl (References WHO), [9](#)  
who.wgt, [6, 8, 9](#)  
who.wgt (References WHO), [9](#)  
wp.twin, [10](#)

y2z, [12, 17](#)

z2y, [13, 15](#)