

# Package ‘zlog’

July 21, 2025

**Title** Z(log) Transformation for Laboratory Measurements

**Version** 1.0.2

**Date** 2022-12-28

**Description** Transformation of laboratory measurements into z or z(log)-value based on given or empirical reference limits as proposed in Hoffmann et al. 2017 <[doi:10.1515/labmed-2016-0087](https://doi.org/10.1515/labmed-2016-0087)>.

**Maintainer** Sebastian Gibb <[mail@sebastiangibb.de](mailto:mail@sebastiangibb.de)>

**Depends** R (>= 3.6.0)

**Suggests** kableExtra, knitr, rmarkdown, survival, testthat

**License** GPL (>= 3)

**RoxygenNote** 7.2.3

**URL** <https://github.com/ampel-leipzig/zlog/>

**BugReports** <https://github.com/ampel-leipzig/zlog/issues/>

**VignetteBuilder** knitr

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Sebastian Gibb [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-7406-4443>>)

**Repository** CRAN

**Date/Publication** 2023-01-05 17:40:02 UTC

## Contents

impute_df . . . . .	2
iz . . . . .	3
lookup_limits . . . . .	4
reference_limits . . . . .	6
set_missing_limits . . . . .	6
z . . . . .	7
zcol . . . . .	9
z_df . . . . .	10

**Index****12**

impute\_df

*Imputation***Description**

Impute NA values with the logmean, mean, minimal or maximum reference value.

**Usage**

```
impute_df(x, limits, method = c("logmean", "mean", "min", "max"))
```

**Arguments**

x	data.frame, with the columns: "age", numeric, "sex", factor and more user defined numeric columns that should be imputed.
limits	data.frame, reference table, has to have the columns: "age", numeric (same units as in age, e.g. days or years, age of 0 matches all ages), "sex", factor (same levels for male and female as sex and a special level "both"), "param", character with the laboratory parameter name that have to match the column name in x, "lower" and "upper", numeric for the lower and upper reference limits.
method	character, imputation method. method = "logmean" (default) replaces all NA with its corresponding logged mean values for the reference table limits (for subsequent use of the <i>zlog</i> score, use method = "mean" for *z* score calculation). For method = "min" or method = "max" the lower or the upper limits are used.

**Value**

data.frame, the same as x but missing values are replaced by the corresponding logmean, mean, minimal or maximal reference values depending on the chosen method.

**Note**

Imputation should be done prior to `z()`/`zlog()` transformation. Afterwards the NA could be replaced by zero (for mean-imputation) via `dis.na(d) <- 0`.

**Author(s)**

Sebastian Gibb

**Examples**

```

l <- data.frame(
  param = c("alb", "bili"),
  age = c(0, 0),
  sex = c("both", "both"),
  units = c("mg/l", "µmol/l"),
  lower = c(35, 2),
  upper = c(52, 21)
)
x <- data.frame(
  age = 40:48,
  sex = rep(c("female", "male"), c(5, 4)),
  # from Hoffmann et al. 2017
  alb = c(42, NA, 38, NA, 50, 42, 27, 31, 24),
  bili = c(11, 9, NA, NA, 22, 42, NA, 200, 20)
)
impute_df(x, l)
impute_df(x, l, method = "min")
zlog_df(impute_df(x, l), l)

```

iz

*Calculate Laboratory Measurements from z/zlog Values***Description**

Inverse function to z or z(log) for laboratory measurement standardisation as proposed in Hoffmann 2017 et al.

**Usage**

```
iz(x, limits, probs = c(0.025, 0.975))
```

```
izlog(x, limits, probs = c(0.025, 0.975))
```

**Arguments**

x	numeric, z/zlog values.
limits	numeric or matrix, lower and upper reference limits. Has to be of length 2 for numeric or a two-column matrix with as many rows as elements in x.
probs	numeric, probabilities of the lower and upper reference limit, default: c(0.025, 0.975) (spanning 95 %). Has to be of length 2 for numeric or a two-column matrix with as many rows as elements in x.

**Details**

The inverse z value is calculated as follows (assuming that the limits where 0.025 and 0.975 quantiles):  $x = z * (limits_2 - limits_1) / 3.92 + (limits_1 + limits_2) / 2$

The inverse z(log) value is calculated as follows (assuming that the limits where 0.025 and 0.975 quantiles):  $x = \exp(z * (\log(limits_2) - \log(limits_1)) / 3.92 + (\log(limits_1) + \log(limits_2)) / 2)$

**Value**

numeric, laboratory measurements.

**Author(s)**

Sebastian Gibb

**References**

Georg Hoffmann, Frank Klawonn, Ralf Lichtinghagen, and Matthias Orth. 2017. "The Zlog-Value as Basis for the Standardization of Laboratory Results." *LaboratoriumsMedizin* 41 (1): 23–32. doi:10.1515/labmed20160087.

**See Also**

[zlog\(\)](#)

**Examples**

```
iz(z(1:10, limits = c(2, 8)), limits = c(2, 8))

# from Hoffmann et al. 2017
albuminzlog <- c(-0.15, -2.25, -1.15, 0.08, 1.57, -0.15, -4.53, -3.16, -5.70)
izlog(albuminzlog, limits = c(35, 52))

bilirubinzlog <- c(0.85, 0.57, -1.96, -0.43, 2.04, 3.12, 2.90, 5.72, 1.88)

limits <- cbind(
  lower = rep(c(35, 2), c(length(albuminzlog), length(bilirubinzlog))),
  upper = rep(c(52, 21), c(length(albuminzlog), length(bilirubinzlog)))
)
izlog(c(albuminzlog, bilirubinzlog), limits = limits)
```

---

lookup\_limits

*Lookup Limits in Reference Tables*

---

**Description**

Reference limits are specific for age and sex. Each laboratory institute has its own reference limits. This function is useful to query a dataset against this database.

**Usage**

```
lookup_limits(age, sex, table)
```

**Arguments**

age	numeric, patient age.
sex	character/factor, patient sex, has to be the same length as age.
table	data.frame, reference table, has to have the columns: "age", numeric (same units as in age, e.g. days or years, age of 0 matches all ages), "sex", factor (same levels for male and female as sex and a special level "both"), "lower" and "upper", numeric for the lower and upper reference limits. If an "param" column is given the "lower" and "upper" limits for all different values in "param" is returned. Additional columns are allowed (and ignored).

**Value**

matrix, with 2 columns ("lower", "upper") and as many rows as length(age).

**Author(s)**

Sebastian Gibb

**Examples**

```
reference <- data.frame(
  param = c("albumin", rep("bilirubin", 4)),
  age = c(0, 1, 2, 3, 7),      # days
  sex = "both",
  units = c("g/l", rep("µmol/l", 4)), # ignored
  lower = c(35, rep(NA, 4)), # no real reference values
  upper = c(52, 5, 8, 13, 18) # no real reference values
)

# lookup albumin reference values for 18 year old woman
lookup_limits(
  age = 18 * 365.25,
  sex = "female",
  table = reference[reference$param == "albumin",]
)

# lookup albumin and bilirubin values for 18 year old woman
lookup_limits(
  age = 18 * 365.25,
  sex = "female",
  table = reference
)

# lookup bilirubin referenc values for infants
lookup_limits(
  age = 0:8,
  sex = rep(c("female", "male"), 5:4),
  table = reference[reference$param == "bilirubin",]
)
```

---

reference\_limits      *Calculate Reference Limits*

---

**Description**

Calculates the lower and upper reference limit for given probabilities.

**Usage**

```
reference_limits(x, probs = c(0.025, 0.975), na.rm = TRUE)
```

**Arguments**

x	numeric, laboratory values
probs	numeric, probabilities of the lower and upper reference limit, default: c(0.025, 0.975) (spanning 95 %).
na.rm	logical, if TRUE (default) NA values are removed before the reference limits are calculated.

**Value**

numeric of length 2 with the lower and upper limit.

**Author(s)**

Sebastian Gibb

**Examples**

```
reference_limits(1:10)
```

---

set\_missing\_limits      *Set Missing Limits in Reference Tables*

---

**Description**

Sometimes reference limits are not specified. That is often the case for biomarkers that are related to infection or cancer. Using zero as lower boundary results in skewed distributions (Hoffmann et al. 2017; fig. 7). Haeckel et al. 2015 suggested to set the lower reference limit to 0.15 of the upper one.

**Usage**

```
set_missing_limits(x, fraction = c(0.15, 20/3))
```

**Arguments**

`x` data.frame, reference table, has to have the columns: "lower" and "upper", numeric for the lower and upper reference limits. Additional columns are allowed (and ignored).

`fraction` numeric(2), targeted fraction of the lower to the upper and the upper to the lower limit. Haeckel et al. 2015 suggested to set the lower limit to 0.15 of the upper one. We choose 20/3 (the reciprocal of 0.15) for the upper to the lower one.

**Value**

data.frame, the same as `x` but the "lower" and "upper" columns are modified if there were NA before.

**Author(s)**

Sebastian Gibb

**References**

Georg Hoffmann, Frank Klawonn, Ralf Lichtinghagen, and Matthias Orth. 2017. "The Zlog-Value as Basis for the Standardization of Laboratory Results." *LaboratoriumsMedizin* 41 (1): 23–32. doi:10.1515/labmed20160087.

Rainer Haeckel, Werner Wosniok, Ebrhard Gurr and Burkhard Peil. 2015. "Permissible limits for uncertainty of measurement in laboratory medicine" *Clinical Chemistry and Laboratory Medicine* 53 (8): 1161-1171. doi:10.1515/cclm20140874.

**Examples**

```
reference <- data.frame(
  param = c("albumin", rep("bilirubin", 4)),
  age = c(0, 1, 2, 3, 7), # ignored
  sex = "both", # ignored
  units = c("g/l", rep("µmol/l", 4)), # ignored
  lower = c(35, rep(NA, 4)), # no real reference values
  upper = c(52, 5, 8, 13, 18) # no real reference values
)
set_missing_limits(reference)
set_missing_limits(reference, fraction = c(0.2, 5))
```

**Description**

Calculates the z or z(log) values for laboratory measurement standardisation as proposed in Hoffmann 2017 et al.

**Usage**

```
z(x, limits, probs = c(0.025, 0.975), log = FALSE)
```

```
zlog(x, limits, probs = c(0.025, 0.975))
```

**Arguments**

x	numeric, laboratory values.
limits	numeric or matrix, lower and upper reference limits. Has to be of length 2 for numeric or a two-column matrix with as many rows as elements in x.
probs	numeric, probabilities of the lower and upper reference limit, default: c(0.025, 0.975) (spanning 95 %). Has to be of length 2 for numeric or a two-column matrix with as many rows as elements in x.
log	logical, should z (log = FALSE, default) or z(log) (log = TRUE) calculated?

**Details**

The z value is calculated as follows (assuming that the limits where 0.025 and 0.975 quantiles):  

$$z = (x - (limits_1 + limits_2)/2) * 3.92 / (limits_2 - limits_1).$$

The z(log) value is calculated as follows (assuming that the limits where 0.025 and 0.975 quantiles):  

$$z = (\log(x) - (\log(limits_1) + \log(limits_2))/2) * 3.92 / (\log(limits_2) - \log(limits_1)).$$

zlog is an alias for z(..., log = TRUE).

**Value**

numeric, z or z(log) values.

**Author(s)**

Sebastian Gibb

**References**

Georg Hoffmann, Frank Klawonn, Ralf Lichtinghagen, and Matthias Orth. 2017. "The Zlog-Value as Basis for the Standardization of Laboratory Results." *LaboratoriumsMedizin* 41 (1): 23–32. doi:[10.1515/labmed20160087](https://doi.org/10.1515/labmed20160087).

**See Also**

[izlog\(\)](#)

**Examples**

```
z(1:10, limits = c(2, 8))

# from Hoffmann et al. 2017
albumin <- c(42, 34, 38, 43, 50, 42, 27, 31, 24)
zlog(albumin, limits = c(35, 52))
```



```
bilirubin <- c(11, 9, 2, 5, 22, 42, 37, 200, 20)

limits <- cbind(
  lower = rep(c(35, 2), c(length(albumin), length(bilirubin))),
  upper = rep(c(52, 21), c(length(albumin), length(bilirubin)))
)
zlog(c(albumin, bilirubin), limits = limits)
```

---

zcol

*Z(log) Depending Color*

---

### Description

This function provides a color gradient depending on the zlog value as described in Hoffmann et al. 2017. The colours are only roughly equal to the one found in the article.

### Usage

```
zcol(x)
```

### Arguments

x                    numeric, z(log) value.

### Value

character, of length(x) with the corresponding color hex code.

### Author(s)

Sebastian Gibb

### References

Hoffmann, Georg, Frank Klawonn, Ralf Lichtinghagen, and Matthias Orth. 2017. "The Zlog-Value as Basis for the Standardization of Laboratory Results." *LaboratoriumsMedizin* 41 (1): 23–32. doi:10.1515/labmed20160087.

### Examples

```
z <- -10:10
image(matrix(z, ncol = 1), col = zcol(z), xaxt = "n", yaxt = "n")
text(seq(0, 1, length.out=length(z)), 0, label = z)
```

z\_df

*Calculate z/zlog Values for a data.frame***Description**

Calculates the z or z(log) values for laboratory measurement standardisation as proposed in Hoffmann 2017 et al. for a complete data.frame.

**Usage**

```
z_df(x, limits, probs = c(0.025, 0.975), log = FALSE)
```

```
zlog_df(x, limits, probs = c(0.025, 0.975))
```

**Arguments**

x	data.frame, with the columns: "age", numeric, "sex", factor and more user defined numeric columns that should be z/z(log) transformed.
limits	data.frame, reference table, has to have the columns: "age", numeric (same units as in age, e.g. days or years, age of 0 matches all ages), "sex", factor (same levels for male and female as sex and a special level "both"), "param", character with the laboratory parameter name that have to match the column name in x, "lower" and "upper", numeric for the lower and upper reference limits.
probs	numeric, probabilities of the lower and upper reference limit, default: c(0.025, 0.975) (spanning 95 %). Has to be of length 2 for numeric or a two-column matrix with as many rows as elements in x.
log	logical, should z (log = FALSE, default) or z(log) (log = TRUE) calculated?

**Details**

This is a wrapper function for `z()` and `lookup_limits()`. Please find the details for the z/z(log) calculation at `z()`.

`zlog_df` is an alias for `z_df(..., log = TRUE)`.

**Value**

data.frame, with the columns: "age", "sex" and all numeric columns z/zlog transformed. If a column name is missing in `limits$param` a warning is thrown and the column is set to NA.

**Author(s)**

Sebastian Gibb

## References

Georg Hoffmann, Frank Klawonn, Ralf Lichtinghagen, and Matthias Orth. 2017. "The Zlog-Value as Basis for the Standardization of Laboratory Results." *LaboratoriumsMedizin* 41 (1): 23–32. doi:10.1515/labmed20160087.

## See Also

[zlog\(\)](#)

## Examples

```
l <- data.frame(
  param = c("alb", "bili"),
  age = c(0, 0),
  sex = c("both", "both"),
  units = c("mg/l", "µmol/l"),
  lower = c(35, 2),
  upper = c(52, 21)
)
x <- data.frame(
  age = 40:48,
  sex = rep(c("female", "male"), c(5, 4)),
  # from Hoffmann et al. 2017
  alb = c(42, 34, 38, 43, 50, 42, 27, 31, 24),
  bili = c(11, 9, 2, 5, 22, 42, 37, 200, 20)
)
z_df(x, l)

zlog_df(x, l)
```

# Index

`impute_df`, 2

`iz`, 3

`izlog(iz)`, 3

`izlog()`, 8

`lookup_limits`, 4

`lookup_limits()`, 10

`reference_limits`, 6

`set_missing_limits`, 6

`z`, 7

`z()`, 2, 10

`z_df`, 10

`zcol`, 9

`zlog(z)`, 7

`zlog()`, 2, 4, 11

`zlog_df(z_df)`, 10