Package ‘lucid’
August 24, 2018

Version 1.6
Title Printing Floating Point Numbers in a Human-Friendly Format
Description Print vectors (and data frames) of floating point numbers using a non-scientific format optimized for human readers. Vectors of numbers are rounded using significant digits, aligned at the decimal point, and all zeros trailing the decimal point are dropped.

LazyData yes
License GPL-3
URL https://github.com/kwstat/lucid
BugReports https://github.com/kwstat/lucid/issues
VignetteBuilder knitr
Imports nlme
Suggests broom, dplyr, knitr, lattice, lme4, rjags, testthat
Encoding UTF-8
RoxygenNote 6.1.0
NeedsCompilation no
Author Kevin Wright [aut, cre] (<https://orcid.org/0000-0002-0617-8673>)
Maintainer Kevin Wright <kw.stat@gmail.com>
Repository CRAN
Date/Publication 2018-08-24 21:02:17 UTC

R topics documented:

antibiotic ................................................................. 2
lucid ................................................................. 3
vc ................................................................. 5

Index 7
Effectiveness of 3 antibiotics against 16 bacterial species.

**Description**

Effectiveness of 3 antibiotics against 16 bacterial species.

**Format**

A data frame with 16 observations on the following 5 variables.

- **bacteria**: bacterial species, 16 levels
- **penicillin**: MIC for penicillin
- **streptomycin**: MIC for streptomycin
- **neomycin**: MIC for neomycin
- **gramstain**: Gram staining (positive or negative)

**Details**

The values reported are the minimum inhibitory concentration (MIC) in micrograms/milliliter, which represents the concentration of antibiotic required to prevent growth in vitro.

**Source**


**References**


**Examples**

```r
data(antibiotic)
lucid(antibiotic)

## Not run:
# Plot the data similar to Fig 2.14 of Wainer's book, "Medical Illuminations"

require(lattice)
require(reshape2)

# Use log10 transform
```
dat <- transform(antibiotic,  
penicillin = log10(penicillin),  
streptomycin = log10(streptomycin),  
neomycin = log10(neomycin))  
dat <- transform(dat, sgn = ifelse(dat$gramstain == "neg", "-", ""))  
dat <- transform(dat,  
bacteria = paste(bacteria, sgn))  
dat <- transform(dat, bacteria = reorder(bacteria, -penicillin))  

dat <- melt(dat)  
op <- tpg <- trellis.par.get()  
tpg$superpose.symbol$pch <- toupper(substring(levels(dat$variable), 1, 1))  
tpg$superpose.symbol$col <- c("darkgreen", "purple", "orange")  
trellis.par.set(tpg)  
dotplot(bacteria ~ value, data = dat, group = variable,  
cex = 2,  
scales = list(x = list(at = -3:3,  
labels = c('.001', '.01', '.1', '1', '10', '100', '1000'))),  
main = "Bacterial response to Neomycin, Streptomycin, and Penicillin",  
  xlab = "Minimum Inhibitory Concentration (mg/L)"")  

trellis.par.set(op)  

## End (Not run)  

---  

**lucid**  

**Lucid printing**  

**Description**  

Format a column of numbers in a way to make it easy to understand.  

**Usage**  

lucid(x, dig = 3, na.value = NULL, ...)  

## Default S3 method:  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'numeric'  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'data.frame'  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'matrix'  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'array'  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'list'  
lucid(x, dig = 3, na.value = NULL, ...)  

## S3 method for class 'data.class'  
lucid(x, dig = 3, na.value = NULL, ...)
## lucid

### Arguments

- **x**: Object to format.
- **dig**: Number of significant digits to use in printing.
- **na.value**: Character string to use instead of `NA` for numeric missing values. Default is `NULL`, which does nothing.
- **...**: Additional arguments passed to the `data.frame` method.

### Details

Output from R is often in scientific notation, which makes it difficult to quickly glance at numbers and gain an understanding of the relative values. This function formats the numbers in a way that makes interpretation of the numbers immediately apparent.

The sequence of steps in formatting the output is: (1) zap to zero (2) use significant digits (3) drop trailing zeros after decimal (4) align decimals.

### Value

Text, formatted in a human-readable way. Standard R methods are used to print the value.

### See Also

- `signif`

### Examples

```r
x0 <- c(123, 12.3, 1.23, .123456) # From Finney, page 352
print(x0)
lucid(x0, dig=2)

x1 <- c(123, NA, 1.23, NA)
lucid(x1, na.value="--")

signif(mtcars[15:20,])
lucid(mtcars[15:20,])

x2 <- c(1/3, 5/3, 1, 1.5, 2, 11/6, 5/6, 8.43215652105343e-17)
print(x2)
lucid(x2)
```
Extract variance components from mixed models

Description

Extract the variance components from a fitted model. Currently supports asreml, lme4, nlme and mcmc.list objects.

Usage

vc(object, ...)

## Default S3 method:
vc(object, ...)

## S3 method for class 'asreml'
vc(object, gamma = FALSE, ...)

## S3 method for class 'lme'
vc(object, ...)

## S3 method for class 'glmerMod'
vc(object, ...)

## S3 method for class 'lmerMod'
vc(object, ...)

## S3 method for class 'mcmc.list'
vc(object, quantiles = c(0.025, 0.5, 0.975), ...)

# Which coef is 0? How large is the intercept?
df1 <- data.frame(effect=c(-13.5, 4.5, 24.5, 6.927792e-14, -1.75, 16.5, 113.5000))
rownames(df1) <- c("A", "B", "C", "C1", "C2", "D", "(Intercept)")
pdf1
lucid(df1)

# Which are smallest/largest/significant variance components
df2 <- data.frame(effect=c("hyb","region","region:loc","hyb:region","yr","hyb:yr","region:yr","R!variance"),
  component=c(10.9,277.493,1.30E-04,126,22.3,481,268),
  std.error=c(4.40,166,26.1,1.58E-06,119,4.50,108,3.25),
  z.ratio=c(2.471,1.669,18.899,82.242,1.060,4.951,4.442,82.242),
  constraint=c("pos","pos","pos","pos","pos","pos","pos","pos")
pdf2
lucid(df2)
Arguments

object  A fitted model object
...  Not used. Extra arguments.
gamma  If gamma=FALSE, then the 'gamma' column is omitted from the results from asreml
quantiles  The quantiles to use for printing mcmc.list objects

Details

The extracted variance components are stored in a data frame with an additional 'vc.xxx' class that has an associated print method.

Value

A data frame or other object.

Examples

## Not run:

require("nlme")
data(Rail)
m3 <- lme(travel~1, random=~1|Rail, data=Rail)
vc(m3)
##    effect variance stddev
## (Intercept)  615.3  24.81
## Residual      16.17  4.021

require("lme4")
m4 <- lmer(travel~1 + (1|Rail), data=Rail)
vc(m4)
##    grp  var1  var2 vcov sdcor
## Rail (Intercept) <NA> 615.3  24.81
## Residual  <NA> <NA> 16.17  4.021

require("asreml")
ma <- asreml(travel~1, random=~Rail, data=Rail)
vc(ma)
##    effect component std.error  z.ratio constr
## Rail!Rail.var    615.3     392.6    1.6    pos
## !variance        16.17      6.6    2.4    pos

# See vignette for rjags example

# To change the number of digits, use the print function.
print(vc(m3), dig=5)

## End(Not run)
Index

*Topic datasets  
  antibiotic, 2

antibiotic, 2
lucid, 3
signif, 4
vc, 5