Package 'lin'

November 13, 2020

Type Package

Title Jennifer Lin's Collection of Functions

Version 0.1.0

URL https://github.com/lin-jennifer/lin

Description These are functions that I use often in my work. May not be useful for general use but may be necessary to install if you are collaborating with me or interacting my R code.

License MIT

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat

Imports ggplot2

Depends R (>= 2.10)

R topics documented:

`ggLin` *Jennifer's ggplot2 Theme*

Description

I find that I am copying and psting the same long command for my favorite theme settings, so I figure I'd write that into this package to simplify the process.

Usage

```
theme_lin(...)
```

Arguments

...	Passed to [ggplot2::theme()]

Details

The specifications of this theme is as follows:
theme_classic()+
theme(
plot.title = element_text(hjust = 0.5, size = 24, colour="black"),
plot.subtitle = element_text(hjust = 0.5, size = 18, colour="black"),
legend.title = element_text(hjust = 0.5, size = 16, colour="black"),
plot.caption = element_text(size = 12, colour="black"),
axis.title = element_text(size = 16, colour="black"),
axis.text.x = element_text(size = 14, colour="black"),
axis.text.y = element_text(size = 14, colour="black"),
legend.title.align = 0.5)

Examples

```
library(ggplot2)
ggplot(mtcars, aes(y=mpg, x=disp, color=cyl)) +
  geom_point() +
  theme_lin()
```

Operators *Operators*

Description

Some useful operators outside of the standard ones in R.

Usage

```
x %nin% y

x %NIN% y

x %IN% y
```

Arguments

x	a vector
y	a vector to match

Value

logical vecotor of items in x not in y

logical vecotor of items in x not in y, omits NAs

logical vecotor of items in x in y, omits NAs

Examples

```
y = c(3, 4, 5, NA)

# Not In -- and omits NA
y %nin% 3    # FALSE  TRUE  TRUE  TRUE
y %NIN% 3    # FALSE  TRUE  TRUE    NA

# IN -- Omits NA
y %in% 3     # TRUE FALSE FALSE FALSE
y %IN% 3     # TRUE FALSE FALSE    NA
```

ReverseCode *Reverse Coding Variables*

Description

I never actually know how to do them, so I google this every time. Perhaps its time to settle this once and for all.

Usage

```
reverse_code(var)
```

Arguments

var	a numeric variable

Details

Thanks goes to James Martherus.

Source

https://github.com/jamesmartherus/martherus

Examples

```
x1 <- c(1, 2, 3, 4, NA, 5)
reverse_code(x1)   #c(5, 4, 3, 2, NA, 1)

x2 <- c(0, 1, 2, NA, 4, 7)
reverse_code(x2)    #c(7, 6, 5, NA, 3, 0)
```

SummaryStats *Summary Statistics Calculations*

Description

Common functions for calculating central tendancies but with NA parameters set to TRUE unlike the defaults.

Usage

```
modeNA(x)

meanNA(x)

wMeanNA(x, w)

medianNA(x)

minNA(x)

maxNA(x)

rangeNA(x)

sdNA(x)

sumNA(x)

varNA(x)
```

Arguments

x	a vector
w	a weight variable

Details

Credits to John Bullock for the inspiration. Some of this is from his Bullock package, but others are my own.

Source

https://github.com/jbullock35/Bullock

Examples

```
x <- c(1, 1, 2, 3, 5, 8, 13, 21, NA, NA, NA)
w <- c(0, 0, 0, 1, 1, 2, 2, 2, 0, 0, 0)

# Mode
modeNA(x)      # 1
```

```
# Mean and Weighted Mean
meanNA(x)      # 6.75
wMeanNA(x, w)  # 11.5

# Median
medianNA(x)    # 4

# Range
minNA(x)       # 1
maxNA(x)       # 21
rangeNA(x)     # c(1, 21)

# Sum
sumNA(x)       # 54

# Variance and Standard Deviation
varNA(x)
sdNA(x)
```

`tscoreCI` *Confidence Interval from a t Distribution*

Description

Helpful functions to calculate a confidence interval using a t-distribution

Usage

```
tCIupper(x, se, ci, n)

tCIlower(x, se, ci, n)

tCI(x, se, ci, n)

tMOE(se, ci, n)
```

Arguments

`x`	x-bar for observed mean
`se`	Standard Error – see se() from zscoreCI section
`ci`	Confidence Interval Level
`n`	Number of participants

Details

NOTE: This is not NOT intended from a z-distribution. See relevant zscoreCI file for those calculations

See Also

zCI(), zCIupper(), zCIlower()

Examples

```
x = 3.5
sd = 2
n = 25

se <- se(sd, n)  # 0.4

ci = 0.95  # for 95% Confidence Interval

tCIupper(x, se, ci, n) # 4.325559
tCIlower(x, se, ci, n) # 2.674441
tMOE(se, ci, n)        # 0.8255594
```

`zscore` *z-score Calculations*

Description

Calculating a standard score in Base R can be hard.

Usage

```
zscore(x, mean, sd)
```

Arguments

`x`	the observation
`mean`	mean of interest – can be sample or population depending on zscore interest
`sd`	standard deviation or standard error, depending on context

Examples

```
zscore(10, 15, 2)
```

`zscoreCI` *Confidence Interval from a Normal Distribution*

Description

Helpful functions to calculate a confidence interval using a z-distribution

Usage

```
se(sd, n)

zCIupper(x, se, ci)

zCIlower(x, se, ci)

zCI(x, se, ci)

zMOE(se, ci)
```

Arguments

sd	Standard Deviation – reference sdNA()
n	Number of participants
x	x-bar for observed mean
se	Standard Error – see se()
ci	Confidence Interval Level

Details

NOTE: This is not NOT intended from a t-distribution. See relavant tscoreCI file for those calculations

See Also

sdNA()

Examples

```
x = 3.5
sd = 2
n = 25

se <- se(sd, n)  # 0.4

ci = 0.95  # for 95% Confidence Interval

zCIupper(x, se, ci) # 4.283986
zCIlower(x, se, ci) # 2.716014
zMOE(se, ci)        # 0.7839856
```

Index