

# Package ‘interpretCI’

July 22, 2025

**Type** Package

**Title** Estimate the Confidence Interval and Interpret Step by Step

**Version** 0.1.1

**URL** <https://github.com/cardiomoon/interpretCI>,  
<https://cardiomoon.github.io/interpretCI/>

**Description** Estimate confidence intervals for mean, proportion, mean difference for unpaired and paired samples and proportion difference. Plot the confidence intervals. Generate documents explaining the statistical result step by step.

**License** GPL-3

**Encoding** UTF-8

**Imports** dplyr, purrr, tidyr, rlang, ggplot2, scales, ggbeeswarm,  
patchwork, aplot, rstudioapi, rmarkdown, flextable, officer,  
english, RColorBrewer, moonBook

**Suggests** knitr, PairedData, glue

**RoxygenNote** 7.1.2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Keon-Woong Moon [aut, cre]

**Maintainer** Keon-Woong Moon <cardiomoon@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-28 08:50:02 UTC

## Contents

|                           |   |
|---------------------------|---|
| acs . . . . .             | 2 |
| draw_n . . . . .          | 3 |
| draw_t . . . . .          | 3 |
| English . . . . .         | 4 |
| english2 . . . . .        | 5 |
| estimationPlot1 . . . . . | 5 |

|                             |           |
|-----------------------------|-----------|
| interpret . . . . .         | 6         |
| is.mynumeric . . . . .      | 7         |
| isProvided . . . . .        | 7         |
| meanCI . . . . .            | 8         |
| meanCI.data.frame . . . . . | 9         |
| meanCI.default . . . . .    | 10        |
| meanCI2df . . . . .         | 11        |
| meanCI2df1 . . . . .        | 12        |
| meanCIplot1 . . . . .       | 12        |
| pairPlot . . . . .          | 13        |
| pairPlot1 . . . . .         | 14        |
| palette2colors . . . . .    | 14        |
| plot.meanCI . . . . .       | 15        |
| print.meanCI . . . . .      | 16        |
| print.plotCI . . . . .      | 17        |
| propCI . . . . .            | 17        |
| propCI_sub . . . . .        | 18        |
| rstudio_viewer . . . . .    | 19        |
| show_t_table . . . . .      | 20        |
| show_z_table . . . . .      | 20        |
| textBox . . . . .           | 21        |
| <b>Index</b>                | <b>23</b> |

---

|     |                                                  |
|-----|--------------------------------------------------|
| acs | <i>Demographic data of 857 patients with ACS</i> |
|-----|--------------------------------------------------|

---

**Description**

A dataset containing demographic data and laboratory data of 857 patients with acute coronary syndrome(ACS).

**Usage**

acs

**Format**

An object of class data.frame with 857 rows and 17 columns.

**Examples**

interpretCI::acs

---

|        |                                       |
|--------|---------------------------------------|
| draw_n | <i>Draw normal distribution curve</i> |
|--------|---------------------------------------|

---

**Description**

Draw normal distribution curve

**Usage**

```
draw_n(mean = 0, sd = 1, z = NULL, p = 0.05, alternative = "two.sided")
```

**Arguments**

|             |                                                                                                                      |
|-------------|----------------------------------------------------------------------------------------------------------------------|
| mean        | vector of means                                                                                                      |
| sd          | vector of standard deviations                                                                                        |
| z           | vector of quantiles                                                                                                  |
| p           | vector of probabilities                                                                                              |
| alternative | a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". |

**Value**

A ggplot

**Examples**

```
draw_n()  
draw_n(alternative="less")  
draw_n(alternative="greater")  
draw_n(z=-1.75)  
draw_n(z=-1.75,alternative="greater")  
draw_n(z=-1.75,alternative="less")
```

---

|        |                                  |
|--------|----------------------------------|
| draw_t | <i>Draw t distribution curve</i> |
|--------|----------------------------------|

---

**Description**

Draw t distribution curve

**Usage**

```
draw_t(DF = 50, t = NULL, p = 0.05, alternative = "two.sided")
```

**Arguments**

|             |                                                                                                                      |
|-------------|----------------------------------------------------------------------------------------------------------------------|
| DF          | numeric degree of freedom                                                                                            |
| t           | numeric t value                                                                                                      |
| p           | numeric p value                                                                                                      |
| alternative | a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". |

**Value**

A ggplot

**Examples**

```
draw_t(DF=30)
draw_t(DF=20,t=2.5)
draw_t(DF=49,t=1.77)
draw_t(DF=49,p=0.005)
draw_t(DF=19,t=-0.894,alternative="less")
draw_t(DF=146,t=0.67,alternative="greater")
```

---

English

---

*Convert numeric to string with uppercase first letter*


---

**Description**

Convert numeric to string with uppercase first letter

**Usage**

```
English(x, digits = 2)
```

**Arguments**

|        |                                                 |
|--------|-------------------------------------------------|
| x      | A numeric                                       |
| digits | integer indicating the number of decimal places |

**Value**

A string

**Examples**

```
English(40)
English(13.1)
```

---

|          |                                  |
|----------|----------------------------------|
| english2 | <i>Convert numeric to string</i> |
|----------|----------------------------------|

---

**Description**

Convert numeric to string

**Usage**

```
english2(x, digits = 2)
```

**Arguments**

|        |                                                 |
|--------|-------------------------------------------------|
| x      | A numeric                                       |
| digits | integer indicating the number of decimal places |

**Value**

A character string

**Examples**

```
english2(45)  
english2(12.34)
```

---

|                 |                              |
|-----------------|------------------------------|
| estimationPlot1 | <i>Draw estimation plot1</i> |
|-----------------|------------------------------|

---

**Description**

Draw estimation plot1

**Usage**

```
estimationPlot1(x, palette = NULL)
```

**Arguments**

|         |                                                             |
|---------|-------------------------------------------------------------|
| x       | An object of class meanCI                                   |
| palette | The name of color palette from RColorBrewer package or NULL |

**Value**

A ggplot

**Examples**

```
x=meanCI(iris,Species,Sepal.Length)
estimationPlot1(x)
```

interpret

*Interpret an object of meanCI***Description**

Interpret an object of meanCI. Render appropriate rmarkdown file to html file and show RStudio viewer or browser.

**Usage**

```
interpret(x, viewer = "rstudio")
```

**Arguments**

|        |                                         |
|--------|-----------------------------------------|
| x      | An object of class "meanCI"             |
| viewer | Character One of c("rstudio","browser") |

**Value**

No return value, called for side effect

**Examples**

```
x=meanCI(mtcars$mpg)
x=meanCI(mtcars,mpg,mu=23)
x=meanCI(n=150,m=115,s=10,alpha=0.01)
x=meanCI(n=50,m=295,s=20,mu=300)
x= meanCI(n=20,m=108,s=10,mu=110,alpha=0.01,alternative="less")
x=meanCI(n1=500,n2=1000,m1=20,s1=3,m2=15,s2=2,alpha=0.01)
x=meanCI(n1=15,n2=20,m1=1000,s1=100,m2=950,s2=90,alpha=0.1)
x=meanCI(n1=30,n2=25,m1=78,s1=10,m2=85,s2=15,mu=0,alpha=0.10)
x=meanCI(n1=100,n2=100,m1=200,s1=40,m2=190,s2=20,mu=7,alpha=0.05,alternative="greater")
x1=c(95,89,76,92,91,53,67,88,75,85,90,85,87,85,85,68,81,84,71,46,75,80)
y1=c(90,85,73,90,90,53,68,90,78,89,95,83,83,83,82,65,79,83,60,47,77,83)
x=meanCI(x=x1,y=y1,paired=TRUE,alpha=0.1,mu=0)
x=propCI(n=1600,p=0.4,alpha=0.01)
x=propCI(n=100,p=0.73,P=0.8,alpha=0.01)
x=propCI(n=100,p=0.73,P=0.8,alpha=0.05,alternative="greater")
x=propCI(n1=100,n2=200,p1=0.38,p2=0.51,alpha=0.01)
x=propCI(n1=150,n2=100,p1=0.71,p2=0.63,P=0,alternative="greater")
## Not run:
interpret(x)
interpret(x,"browser")

## End(Not run)
```

---

`is.mynumeric`*Decide whether a vector can be treated as a numeric variable*

---

**Description**

Decide whether a vector can be treated as a numeric variable

**Usage**

```
is.mynumeric(x, maxy.lev = 5)
```

**Arguments**

|                       |                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------|
| <code>x</code>        | A vector                                                                                                             |
| <code>maxy.lev</code> | An integer indicating the maximum number of unique values of a numeric variable be treated as a categorical variable |

**Value**

logical

**Examples**

```
x=1:5
is.mynumeric(x)
x=1:13
is.mynumeric(x)
```

---

`isProvided`*Whether the arg is provided in function call*

---

**Description**

Whether the arg is provided in function call

**Usage**

```
isProvided(x, seek = "mu")
```

**Arguments**

|                   |                                                                  |
|-------------------|------------------------------------------------------------------|
| <code>x</code>    | An object of class "meanCI" or function call or character string |
| <code>seek</code> | character. Default="mu"                                          |

**Value**

logical

**Examples**

```
x=meanCI(mtcars,am,mpg)
isProvided(x)
```

meanCI

*Calculate confidence intervals of mean or difference between means***Description**

Calculate confidence intervals of mean or difference between means

**Usage**

```
meanCI(x, ...)
```

**Arguments**

x                    An object of class data.frame or vector  
...                   Further arguments

**Value**

An object of class "meanCI" which is a list containing at least the following components:

**data** A tibble containing raw data or a list of numeric vector

**result** A data.frame consist of summary statistics

**call** the matched call

**attr(\*,"measure")** character. One of c("mean","unpaired","paired")

**Examples**

```
meanCI(mtcars$mpg)
meanCI(n=150,m=115,s=10,alpha=0.01)
meanCI(n=50,m=295,s=20,mu=300)
meanCI(n=20,m=108,s=10,mu=110,alpha=0.01,alternative="less")
meanCI(mtcars,am,mpg)
meanCI(n1=15,n2=20,m1=1000,s1=100,m2=950,s2=90,alpha=0.1)
meanCI(n1=500,n2=1000,m1=20,s1=3,m2=15,s2=2,alpha=0.01)
meanCI(n1=30,n2=25,m1=78,s1=10,m2=85,s2=15,alpha=0.10)
meanCI(n1=100,n2=100,m1=200,s1=40,m2=190,s2=20,mu=7,alpha=0.05,alternative="greater")
x=c(3.04,2.92,2.86,1.71,3.60,3.49,3.30,2.28,3.11,2.88,2.82,2.13,2.11,3.03,3.27,2.60,3.13)
y=c(2.56,3.47,2.65,2.77,3.26,3.00,2.70,3.20,3.39,3.00,3.19,2.58,2.98)
meanCI(x=x,y=y)
x=c(95,89,76,92,91,53,67,88,75,85,90,85,87,85,85,68,81,84,71,46,75,80)
y=c(90,85,73,90,90,53,68,90,78,89,95,83,83,83,82,65,79,83,60,47,77,83)
meanCI(x=x,y=y,paired=TRUE,alpha=0.1)
meanCI(10:30,1:15)
meanCI(acs,sex,age)
meanCI(iris$Sepal.Width,iris$Sepal.Length)
meanCI(iris$Sepal.Width,iris$Sepal.Length,paired=TRUE)
```

---

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| meanCI.data.frame | <i>Calculate confidence intervals of mean or difference between means in a data.frame</i> |
|-------------------|-------------------------------------------------------------------------------------------|

---

## Description

Calculate confidence intervals of mean or difference between means in a data.frame

## Usage

```
## S3 method for class 'data.frame'
meanCI(x, ...)

meanCI_sub(data = data, x, y, group, paired = FALSE, idx = NULL, ...)
```

## Arguments

|        |                                                                                                                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x      | Name of a categorical or numeric column. If !missing(y), name of continuous variable                                                                                                                                                            |
| ...    | Further arguments to be passed to meanCI                                                                                                                                                                                                        |
| data   | A data.frame                                                                                                                                                                                                                                    |
| y      | Name of a numeric column                                                                                                                                                                                                                        |
| group  | Name of categorical column                                                                                                                                                                                                                      |
| paired | logical                                                                                                                                                                                                                                         |
| idx    | A vector containing factors or strings in the x columns. These must be quoted (ie. surrounded by quotation marks). The first element will be the control group, so all differences will be computed for every other group and this first group. |

## Value

An object of class "meanCI" which is a list containing at least the following components:

**data** A tibble containing raw data or a list of numeric vector

**result** A data.frame consists of summary statistics

**call** the matched call

**attr(\*,"measure")** character. One of c("mean","unpaired","paired")

## Methods (by generic)

- meanCI: S3 method for data.frame

**Examples**

```

meanCI(acs,age)
meanCI(acs,sex,age)
meanCI(acs,Dx,age)
acs %>% select(age) %>% meanCI()
acs %>% select(sex,age) %>% meanCI()
meanCI(iris,Species,Sepal.Length)
meanCI(iris,Sepal.Width,Sepal.Length,paired=TRUE)
meanCI(iris,Sepal.Length,Sepal.Width)
iris %>% select(starts_with("Petal")) %>% meanCI(paired=TRUE)
iris %>% meanCI(paired=TRUE)
meanCI(acs,sex,age,Dx,mu=10)
acs %>% select(sex,TC,TG,HDL) %>% meanCI(group=sex)
acs %>% select(sex,TC,TG,HDL) %>% meanCI(sex)
iris %>% select(Species,starts_with("Sepal")) %>% meanCI(Species)
iris %>% select(Species,starts_with("Sepal")) %>% meanCI(group=Species)

```

---

|                |                                                                           |
|----------------|---------------------------------------------------------------------------|
| meanCI.default | <i>Calculate confidence intervals of mean or difference between means</i> |
|----------------|---------------------------------------------------------------------------|

---

**Description**

Calculate confidence intervals of mean or difference between means

**Usage**

```

## Default S3 method:
meanCI(x, ...)

meanCI2(
  x,
  y,
  n,
  m,
  s,
  n1,
  n2,
  m1,
  m2,
  s1,
  s2,
  mu = 0,
  paired = FALSE,
  var.equal = FALSE,
  alpha = 0.05,
  digits = 2,
  alternative = "two.sided"
)

```

**Arguments**

|                          |                                                                                                                      |
|--------------------------|----------------------------------------------------------------------------------------------------------------------|
| <code>x</code>           | A vector                                                                                                             |
| <code>...</code>         | Further arguments to be passed to <code>meanCI2</code>                                                               |
| <code>y</code>           | A vector                                                                                                             |
| <code>n, n1, n2</code>   | integer sample(s) size                                                                                               |
| <code>m, m1, m2</code>   | Numeric mean value of sample(s)                                                                                      |
| <code>s, s1, s2</code>   | Numeric standard deviation of sample(s)                                                                              |
| <code>mu</code>          | numeric hypothesized true value of mean or mean difference                                                           |
| <code>paired</code>      | logical If true, difference between paired sample calculated                                                         |
| <code>var.equal</code>   | logical If true, pooled standard deviation is used                                                                   |
| <code>alpha</code>       | Numeric Confidence level                                                                                             |
| <code>digits</code>      | integer indicating the number of decimal places                                                                      |
| <code>alternative</code> | A character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". |

**Value**

An object of class "meanCI" which is a list containing at least the following components:

**data** A tibble containing raw data or a list of numeric vector

**result** A data.frame containing of summary statistics

**call** the matched call

**attr(\*,"measure")** character. One of `c("mean","unpaired","paired")`

**Methods (by generic)**

- `meanCI`: Default S3 method

---

`meanCI2df`
*Prepare data to plot from an object of class meanCI*


---

**Description**

Prepare data to plot from an object of class `meanCI`

**Usage**

```
meanCI2df(x)
```

**Arguments**

|                |                                        |
|----------------|----------------------------------------|
| <code>x</code> | An object of class <code>meanCI</code> |
|----------------|----------------------------------------|

**Value**

A data.frame

**Examples**

```
x=meanCI(acs,sex,age)
meanCI2df(x)
```

---

meanCI2df1

*Extract data from an object of class meanCI*

---

**Description**

Extract data from an object of class meanCI

**Usage**

```
meanCI2df1(x)
```

**Arguments**

x                      An object of class meanCI

**Value**

A data.frame summarizing mean and confidence interval

**Examples**

```
x=meanCI(acs,sex,age)
meanCI2df1(x)
```

---

meanCIplot1

*Draw meanCI plot for data with single vector*

---

**Description**

Draw meanCI plot for data with single vector

**Usage**

```
meanCIplot1(x)
```

**Arguments**

x                      An object of class "meanCI" with attr(x,"measure")==="mean"

**Value**

A ggplot

**Examples**

```
x=meanCI(mtcars,mpg)
meanCIplot1(x)
```

---

pairPlot

*Draw a pair plot with an object of class meanCI*

---

**Description**

Draw a pair plot with an object of class meanCI

**Usage**

```
pairPlot(x, palette = NULL)
```

**Arguments**

|         |                                                               |
|---------|---------------------------------------------------------------|
| x       | An object of class "meanCI" with attr(x,"measure")==="paired" |
| palette | The name of color palette from RColorBrewer package or NULL   |

**Value**

A ggplot

**Examples**

```
x=meanCI(iris,paired=TRUE)
pairPlot(x)
x=meanCI(iris,Petal.Width, Petal.Length,paired=TRUE)
pairPlot(x)
```

---

|           |                         |
|-----------|-------------------------|
| pairPlot1 | <i>Draw a pair plot</i> |
|-----------|-------------------------|

---

**Description**

Draw a pair plot

**Usage**

```
pairPlot1(data, ref = NULL, palette = NULL)
```

**Arguments**

|         |                                                             |
|---------|-------------------------------------------------------------|
| data    | a data.frame                                                |
| ref     | Numeric or NULL                                             |
| palette | The name of color palette from RColorBrewer package or NULL |

**Value**

A ggplot

**Examples**

```
x=meanCI(mtcars,paired=TRUE)
pairPlot1(x$data)
pairPlot1(x$data,ref=c(1,4,6))
pairPlot1(x$data,ref=c(1,3))
```

---

|                |                                                        |
|----------------|--------------------------------------------------------|
| palette2colors | <i>Extract hexadecimal colors from a color palette</i> |
|----------------|--------------------------------------------------------|

---

**Description**

Extract hexadecimal colors from a color palette

**Usage**

```
palette2colors(name, reverse = FALSE)
```

**Arguments**

|         |                                                     |
|---------|-----------------------------------------------------|
| name    | The name of color palette from RColorBrewer package |
| reverse | Whether or not reverse the order of colors          |

**Value**

hexadecimal colors

**Examples**

```
palette2colors("Reds")
```

---

|             |                                                  |
|-------------|--------------------------------------------------|
| plot.meanCI | <i>S3 method for an object of class "meanCI"</i> |
|-------------|--------------------------------------------------|

---

**Description**

S3 method for an object of class "meanCI"

**Usage**

```
## S3 method for class 'meanCI'
plot(x, ref = "control", side = NULL, palette = NULL, ...)
```

**Arguments**

|         |                                                             |
|---------|-------------------------------------------------------------|
| x       | an object of class "meanCI"                                 |
| ref     | string One of c("test","control").                          |
| side    | logical or NULL If true draw side by side plot              |
| palette | The name of color palette from RColorBrewer package or NULL |
| ...     | Further arguments to be passed                              |

**Value**

A ggplot or an object of class "plotCI" containing at least the following components: '

**p1** A ggplot

**p2** A ggplot

**side** logical

**Examples**

```
meanCI(mtcars,mpg) %>% plot()
meanCI(mtcars,am,mpg) %>% plot()
meanCI(iris,Sepal.Width) %>% plot()
meanCI(iris,Sepal.Width,Sepal.Length) %>% plot()

meanCI(iris,Sepal.Width,Sepal.Length,paired=TRUE) %>% plot(palette="Dark2")
meanCI(iris,Sepal.Width,Sepal.Length) %>% plot()
meanCI(iris,Species,Sepal.Width) %>% plot(side=TRUE)
meanCI(iris,Species,Sepal.Width,mu=0.5,alternative="less") %>% plot(ref="test")
meanCI(acs,age) %>% plot()
```

```

meanCI(acs,sex,age) %>% plot()
meanCI(acs,smoking,age) %>% plot(palette="Set1")
meanCI(acs,Dx,age) %>% plot()
meanCI(acs,Dx,age,sex,mu=0) %>% plot(palette="Dark2")
x=c(95,89,76,92,91,53,67,88,75,85,90,85,87,85,85,68,81,84,71,46,75,80)
y=c(90,85,73,90,90,53,68,90,78,89,95,83,83,83,82,65,79,83,60,47,77,83)
meanCI(x=x,y=y,paired=TRUE,alpha=0.1) %>% plot()
meanCI(10:30,1:15) %>% plot()
iris %>% meanCI() %>% plot(side=TRUE)
meanCI(n=150,m=115,s=10,alpha=0.01) %>% plot()
meanCI(n1=30,n2=25,m1=78,s1=10,m2=85,s2=15,alpha=0.10) %>% plot()
data(anscombe2,package="PairedData")
meanCI(anscombe2,idx=list(c("X1","Y1"),c("X4","Y4"),c("X3","Y3"),c("X2","Y2")),
paired=TRUE,mu=0) %>% plot()
x=meanCI(anscombe2,idx=list(c("X1","X2","X3","X4"),c("Y1","Y2","Y3","Y4")),paired=TRUE,mu=0)
plot(x)
longdf=tidyr::pivot_longer(anscombe2,cols=X1:Y4)
x=meanCI(longdf,name,value,idx=list(c("X1","X2","X3","X4"),c("Y1","Y2","Y3","Y4")),paired=TRUE,mu=0)
plot(x)
acs %>% select(sex,TC,TG,HDL) %>% meanCI(group=sex) %>% plot()
acs %>% select(sex,TC,TG,HDL) %>% meanCI(sex) %>% plot()

```

---

print.meanCI

*S3 method "print" for class "meanCI"*


---

## Description

S3 method "print" for class "meanCI"

## Usage

```
## S3 method for class 'meanCI'
print(x, ...)
```

## Arguments

|     |                             |
|-----|-----------------------------|
| x   | An object of class "meanCI" |
| ... | Further arguments           |

## Value

No return value, called for side effect

---

|              |                                   |
|--------------|-----------------------------------|
| print.plotCI | <i>S3 method for class plotCI</i> |
|--------------|-----------------------------------|

---

**Description**

S3 method for class plotCI

**Usage**

```
## S3 method for class 'plotCI'  
print(x, ...)
```

**Arguments**

|     |                           |
|-----|---------------------------|
| x   | An object of class plotCI |
| ... | Further arguments         |

**Value**

No return value, called for side effect

---

|        |                                                                                       |
|--------|---------------------------------------------------------------------------------------|
| propCI | <i>Calculate confidence intervals of proportion or difference between proportions</i> |
|--------|---------------------------------------------------------------------------------------|

---

**Description**

Calculate confidence intervals of proportion or difference between proportions

**Usage**

```
propCI(  
  x,  
  y,  
  n,  
  p,  
  n1,  
  n2,  
  p1,  
  p2,  
  P = 0,  
  alpha = 0.05,  
  digits = 2,  
  alternative = "two.sided"  
)
```

**Arguments**

|              |                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| x            | A vector                                                                                                             |
| y            | A vector                                                                                                             |
| n, n1, n2    | integer sample size                                                                                                  |
| p, p1, p2, P | Numeric proportion                                                                                                   |
| alpha        | numeric confidence level                                                                                             |
| digits       | integer indicating the number of decimal places                                                                      |
| alternative  | A character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". |

**Value**

A list containing at least the following components:

**data** A tibble containing raw data or a list of numeric vector

**result** A data.frame consists of summary statistics

**call** the matched call

**attr(\*,"measure")** character. One of c("prop","propdiff")

#' @examples propCI(acs\$sex) propCI(acs\$sex,acs\$DM) propCI(n=1600,p=0.4,alpha=0.01) propCI(n=100,p=0.73,P=0.8,alpha=0.01)  
propCI(n1=400,n2=300,p1=0.4,p2=0.3,alpha=0.1) propCI(n1=100,n2=200,p1=0.38,p2=0.51,alpha=0.01)  
propCI(n1=100,n2=200,p1=0.38,p2=0.51,alpha=0.01,alternative="less")

---

|            |                                                                                                       |
|------------|-------------------------------------------------------------------------------------------------------|
| propCI_sub | <i>Calculate confidence intervals of proportion or difference between proportions in a data.frame</i> |
|------------|-------------------------------------------------------------------------------------------------------|

---

**Description**

Calculate confidence intervals of proportion or difference between proportions in a data.frame

**Usage**

```
propCI_sub(data, x, y = NULL)
```

**Arguments**

|      |                                                        |
|------|--------------------------------------------------------|
| data | A data.frame                                           |
| x    | Character Name of a categorical column                 |
| y    | Character Optional. Name of another categorical column |

**Value**

A list containing at least the following components:

**data** A tibble containing raw data or a list of numeric vector

**result** A data.frame consists of summary statistics

**call** the matched call

**attr(\*,"measure")** character. One of c("prop","proptdiff")

**Examples**

```
propCI_sub(acs,"sex")
propCI_sub(acs,"sex","HBP")
```

---

rstudio\_viewer

*Show html file in RStudio viewer or browser*

---

**Description**

Show html file in RStudio viewer or browser

**Usage**

```
rstudio_viewer(file_name, file_path = NULL, viewer = "rstudio")
```

**Arguments**

|           |                                         |
|-----------|-----------------------------------------|
| file_name | character file name                     |
| file_path | character file path                     |
| viewer    | Character One of c("rstudio","browser") |

**Value**

No return value, called for side effect

---

|              |                    |
|--------------|--------------------|
| show_t_table | Show t-value table |
|--------------|--------------------|

---

**Description**

Show t-value table

**Usage**

```
show_t_table(DF = 20, p = 0.05, alternative = "two.sided")
```

**Arguments**

|             |                                                    |
|-------------|----------------------------------------------------|
| DF          | Numeric degree of freedom                          |
| p           | Numeric probability                                |
| alternative | Character One of c("two.sided", "greater", "less") |

**Value**

An object of class "flextable"

**Examples**

```
show_t_table()
```

---

|              |                    |
|--------------|--------------------|
| show_z_table | Show z-value table |
|--------------|--------------------|

---

**Description**

Show z-value table

**Usage**

```
show_z_table(p = 0.05, alternative = "two.sided")
```

**Arguments**

|             |                                                    |
|-------------|----------------------------------------------------|
| p           | Numeric probability                                |
| alternative | Character One of c("two.sided", "greater", "less") |

**Value**

An object of class "flextable"

Examples

```
show_z_table()
show_z_table(p=0.01)
```

---

|         |                     |
|---------|---------------------|
| textBox | <i>Draw textbox</i> |
|---------|---------------------|

---

Description

Draw textbox

Usage

```
textBox(
  string,
  color = "black",
  lcolor = "red",
  bg = "cornsilk",
  lwd = 1,
  width = 10,
  bold = FALSE,
  italic = FALSE,
  fontsize = 11,
  space = 1.5,
  fontname
)
```

Arguments

|              |                     |
|--------------|---------------------|
| string       | string              |
| color        | font color          |
| lcolor       | line color          |
| bg           | background color    |
| lwd          | numeric line width  |
| width        | numeric box width   |
| bold, italic | logical             |
| fontsize     | numeric font size   |
| space        | space between lines |
| fontname     | name of font        |

Value

A flextable

**Examples**

```
string="Good Morning!"  
textBox(string,italic=TRUE)
```

# Index

## \* datasets

acs, [2](#)

acs, [2](#)

draw\_n, [3](#)

draw\_t, [3](#)

English, [4](#)

english2, [5](#)

estimationPlot1, [5](#)

interpret, [6](#)

is.mynumeric, [7](#)

isProvided, [7](#)

meanCI, [8](#)

meanCI.data.frame, [9](#)

meanCI.default, [10](#)

meanCI2(meanCI.default), [10](#)

meanCI2df, [11](#)

meanCI2df1, [12](#)

meanCI\_sub(meanCI.data.frame), [9](#)

meanCIplot1, [12](#)

pairPlot, [13](#)

pairPlot1, [14](#)

palette2colors, [14](#)

plot.meanCI, [15](#)

print.meanCI, [16](#)

print.plotCI, [17](#)

propCI, [17](#)

propCI\_sub, [18](#)

rstudio\_viewer, [19](#)

show\_t\_table, [20](#)

show\_z\_table, [20](#)

textBox, [21](#)