

Operators

=	Assigns a value to an object
<-	
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
==	exactly equal to
!=	not equal to
!x	not x
x y	x OR y
x & y	x AND y

Basic R Functions

Access a function's help file

```
help(function name)
```

Load a csv file

```
read.csv( "filename.csv", header = TRUE )
```

Install a library

```
install.packages("library name")
```

Load an installed library

```
library(library name)
```

Resize images in Jupyter and Google Collab

```
options(repr.plot.width = x, repr.plot.height = y)
```

Return the amount of values in x

```
length(x)
```

Return the absolute value(s) in x

```
abs(x)
```



By **Jeffrey M. Pisklak**
(non_human_entity)
cheatography.com/non-human-entity/

Not published yet.
Last updated 16th October, 2020.
Page 1 of 5.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Basic R Functions (cont)

Return the sum of all the values in x

```
sum(x)
```

Return the square-root of the value(s) in x

```
sqrt(x)
```

Return the mean of the values in x with optional arguments for trimming and removing NAs

```
mean(x, tr = 0, na.rm = FALSE)
```

Return the median of the values in x with optional arguments removing NAs

```
median(x, na.rm = FALSE)
```

Return the sample standard deviation of values in x with optional argument for removing NAs

```
sd(x, na.rm = FALSE)
```

Return the sample variance of values in x with optional argument for removing NAs

```
var(x, na.rm = FALSE)
```

Return the quartiles for x with optional argument for removing NAs

```
quantile(x, na.rm = FALSE)
```

Sort the values of x into ascending order

```
sort(x)
```

Compute the median absolute deviation of x with optional argument to remove NAs

```
mad(x, na.rm = FALSE)
```

Find NA values in x (returns TRUE/FALSE)

```
is.na(x)
```

Compute the classic one and two-sample t-tests and Welch's t-test

```
t.test(y~x,  
       alternative = c("two.sided"),  
       mu = 0,  
       var.equal = FALSE,  
       conf.level = 0.95)
```



Data Frames

Create a new data frame

```
Column_1 = c("A", "B", "C")
Column_2 = c(21, 22, NA)
new_df = data.frame(Column_1, Column_2)
```

Add a column

```
new_df$Column_3 = c(51, 52, 53)
```

Select a specific value (e.g., 52 = row 2, column 3)

```
new_df[2, 3]
```

Select a series of values (e.g., all of row 2)

```
new_df[2, c(1,2,3)]
or
new_df[2, ]
```

Select an entire column (e.g., column 2)

```
new_df$Column_2
or
new_df[, 2]
```

Isolate values that are not NAs

```
new_df$Column_2[!is.na(new_df$Column_2)]
```

Subset Function

Used to select specific observations from a dataframe according to a rule you specify.

```
subset(dataframe, subset rule, select = ("columns to keep"))
```

Example:

```
outliers = subset(heightData, Father < 60.1 | Father > 75.3, select = c("Father"))
```

Library Functions

library(effsize)

Cohen's d and Hedges g

```
cohen.d(y~x, data, hedges.correction = FALSE)
```

library(plyr)



By **Jeffrey M. Pisklak**
(non_human_entity)
cheatography.com/non-human-entity/

Not published yet.
Last updated 16th October, 2020.
Page 3 of 5.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Library Functions (cont)

Contains the `ddply` function to easily aggregate data sets into a new dataframe. For example ...

```
new_df = ddply(dataframe, c("Group_Column"), summarise,
  n = length(Score_Column),
  Means = mean(Score_Column)
)
```

`library(rcompanion)`

Calculates lambda for Tukey's ladder of powers

```
transformTukey(x, plotit = FALSE, returnLambda = TRUE)
```

`library(WRS2)`

Winsorized *variance* of x

```
winvar(x, tr = .2)
```

Yuen's two sample t-test for trimmed means

```
yuen(y ~ x, tr = .2)
```

Distribution Functions

Normal Distribution

Return the the corresponding quantile for a given probability.

```
qnorm(probability, mean, sd, lower.tail = TRUE)
```

Return the the corresponding probability for a given quantile.

```
pnorm(quantile, mean, sd, lower.tail = TRUE)
```

T Distribution

Return the the corresponding quantile for a given probability.

```
qt(probability, df, lower.tail = TRUE)
```

Return the the corresponding probability for a given quantile.

```
pt(quantile, df, lower.tail = TRUE)
```

Note:

- z-scores and t-scores (e.g. critical T and test statistics) are types of quantiles.

- The calculations are all performed from left to right by default unless you specify `lower.tail = FALSE`).



By **Jeffrey M. Pisklak**
(non_human_entity)
cheatography.com/non-human-entity/

Not published yet.
Last updated 16th October, 2020.
Page 4 of 5.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Plotting: library(ggplot2)

Histogram

```
ggplot(dataFrame, aes(x = Dep_Var)) +
  geom_histogram(colour = "black",
                 fill = "white")
```

Density Plot

```
ggplot(dataFrame, aes(x = Dep_Var)) +
  geom_density(colour = "black",
               fill = "pink",
               adjust = 1)
```

Boxplot - for one sample

```
ggplot(dataFrame, aes(y = Dep_Var,)) +
  geom_boxplot()
```

Boxplot - for two or more samples

```
ggplot(dataFrame, aes(x = Indep_Var, y = Dep_Var)) +
  geom_boxplot()
```

Barplot with errorbars

```
ggplot(plotData, aes(x = Indep_Var, y = Dep_Var,
                     fill = Indep_Var)) +
  geom_bar(stat = "identity", colour = "black") +
  geom_errorbar(aes(ymin = bottom_value,
                   ymax = top_value),
               width = .25)
```

Q-Q Plot For two independent samples

Remove + facet_wrap() for a single sample

```
ggplot(dataFrame, aes(sample = Dep_Var)) +
  stat_qq() +
  stat_qq_line() +
  facet_wrap(~ Indep_Var)
```

Note:

Indep_Var = Independent Variable

Dep_Var = Dependent Variable

plotData = Dataframe of aggregated values



By **Jeffrey M. Pisklak**
(non_human_entity)
cheatography.com/non-human-entity/

Not published yet.
Last updated 16th October, 2020.
Page 5 of 5.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>