R Cheat Sheet Cheat Sheet

by Jeffrey M. Pisklak (non_human_entity) via cheatography.com/127433/cs/24819/

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
`%>%`	Sends something (e.g., dataframe or function output) to a tidyverse function. i.e., it is an elegant way to nest tidyverse functions.

Example:

```
library(tidyverse)
msleep %>%
   filter (conservation == 'domesticated') %>%
     summarise(m = mean(brainwt, na.rm = TRUE),
              s = sd(brainwt, na.rm = TRUE),
              med = median (brainwt, na.rm = TRUE)
              )
```

Note that `summarise()` is nested within `filter()`, which is using info from the `msleep` dataframe.

Basic R Functions

Access a function's help file

```
?[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 Google Collab

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

Return the amount of values in x

```
length(x)
```

Return the number of rows in a dataframe

```
nrow(data frame)
```

Return the absolute value(s) in x

```
abs(x)
```

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)
```

Basic R Functions (cont)

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

```
quanti le(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)
```

Paste things together into a single string

```
paste(x, y, z, sep = "  ")
```

Create a table of counts

Examples:

```
table(x)
table(x, y)
```

Data Frames

Create a new data frame

```
Column_1 <- c("A ", " B", " C")
Column_2 <- c(21, 22, NA)
new_df <- data.f ram e(C olu mn_1, Column_2)
```

Add a column

```
new_df $Co lumn_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 $Co lumn_2
```

or

```
new_df[ , 2]
```

Isolate values that are not NAs

```
new_df $Co lum n_2 [!i s.n a(n ew_ df$ Col umn_2)]
```

R Cheat Sheet Cheat Sheet

by Jeffrey M. Pisklak (non_human_entity) via cheatography.com/127433/cs/24819/

Filter Function

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

```
filter (dataframe, subset rule)
```

Example 1:

```
filter (heightData, Father < 60.1 | Father > 75.3)
```

Example 2:

```
heightData %>% filter (Father < 60.1 | Father > 75.3)
```

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(tidyverse) or **library(dplyr)**

Aggregate data sets into a new dataframe.

For example . . .

```
msleep %>%
  group_by(vore, conservation) %>%
  summarise(m = mean(brainwt, na.rm = TRUE),
            s = sd(brainwt, na.rm = TRUE)
            )
```

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)
```

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

Published 28th December, 2023.
Last updated 28th December, 2023.

Distribution Functions

Return the the corresponding *quantile* for a given probability

Normal Distribution

```
qnorm(probability, mean, sd)
```

T Distribution

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

F Distribution

```
qf(probability, df1, df2, lower.tail)
```

Chi-Square Distribution

```
qchisq(probability, df, lower.tail)
```

Return the the corresponding *probability* for a given quantile.

Normal Distribution

```
pnorm(quantile, mean, sd)
```

T Distribution

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

F Distribution

```
pf(quantile, df1, df2, lower.tail)
```

Chi-Square Distribution

```
pchisq(quantile, df, lower.tail)
```

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).

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)
```

Plotting: library(ggplot2) (cont)

Boxplot - for one sample

```
ggplot (da taF rame, aes(y = Dep_Var,)) +
            geo m_b oxp lot()
```

Boxplot - for two or more samples

```
ggplot (da taF rame, aes(x = Indep_Var, y = Dep_Var)) +
            geo m_b oxp lot()
```

Barplot with errorbars

```
ggplot (pl otData, aes(x = Indep_Var, y = Dep_Var,
                                   fill = Indep_ Var)) +
    geom_b ar(stat = " ide nti ty", colour = " bla ck") +
    geom_e rro rba r(a es(ymin = bottom _va lues,
                                         ymax = top_va lues),
                                  width = .25)
```

Q-Q Plot For two independent samples

Remove + facet_wrap() for a single sample

```
ggplot (da taF rame, aes(sample = Dep_Var)) +
                 stat_qq() +
                 stat_q q_l ine() +
              facet_ wrap(~ Indep_Var)
```

Line Plot of Means with Two Predictors

```
ggplot (pl otData, aes(x = Predic torA, y = Means, group = Predic torB, colour = Predic torB)) +
     geom_l ine (po sition = positi on_ dod ge( width = 0.4)) +
     geom_p oin t(p osition = positi on_ dod ge( width = 0.4))
```

Scatterplot with Regression Line

```
ggplot (da taf rame, aes(x = predictor, y = response)) +
     geom_p oint() +
     geom_a bli ne( int ercept = b0, slope = b1)
```

Note:

Indep_Var = Independent Variable

Dep_Var = Dependent Variable

plotData = Dataframe of aggregated values

R Style Guide (from the Tidyverse)

https://style.tidyverse.org/