

# Learners' Reference

Cheat sheet of functions used in the lessons

## Lesson 1 – Introduction to R

- `sqrt()` # calculate the square root
- `round()` # round a number
- `args()` # find what arguments a function takes
- `length()` # how many elements are in a particular vector
- `class()` # the class (the type of element) of an object
- `str()` # an overview of the object and the elements it contains
- `c()` # create vector; add elements to vector
- `[ ]` # extract and subset vector
- `%in%` # to test if a value is found in a vector
- `is.na()` # test if there are missing values
- `na.omit()` # Returns the object with incomplete cases removed
- `complete.cases()` # elements which are complete cases

## Lesson 2 – Starting with data

- `download.file()` # download files from the internet to your computer
- `read_csv()` # load CSV file into R memory
- `head()` # shows the first 6 rows
- `view()` # invoke a spreadsheet-style data viewer
- `read_delim()` # load a file in table format into R memory
- `str()` # check structure of the object and information about the class, length and content of each column
- `dim()` # check dimension of data frame
- `nrow()` # returns the number of rows
- `ncol()` # returns the number of columns
- `tail()` # shows the last 6 rows
- `names()` # returns the column names (synonym of `colnames()` for data frame objects)
- `rownames()` # returns the row names
- `summary()` # summary statistics for each column
- `factor()` # create factors
- `levels()` # check levels of a factor
- `nlevels()` # check number of levels of a factor
- `as.character()` # convert an object to a character vector
- `as.numeric()` # convert an object to a numeric vector
- `as.numeric(as.character(x))` # convert factors where the levels appear as characters to a numeric vector
- `as.numeric(levels(x))[x]` # convert factors where the levels appear as numbers to a numeric vector
- `plot()` # plot an object

- `addNA()` # convert NA into a factor level
- `data.frame()` # create a data.frame object
- `ymd()` # convert a vector representing year, month, and day to a Date vector
- `paste()` # concatenate vectors after converting to character

### Lesson 3 – Manipulating, analyzing and exporting data with tidyverse

- `str()` # check structure of the object and information about the class, length and content of each column
- `view()` # invoke a spreadsheet-style data viewer
- `select()` # select columns of a data frame
- `filter()` # allows you to select a subset of rows in a data frame
- `%>%` # pipes to select and filter at the same time
- `mutate()` # create new columns based on the values in existing columns
- `head()` # shows the first 6 rows
- `group_by()` # split the data into groups, apply some analysis to each group, and then combine the results.
- `summarize()` # collapses each group into a single-row summary of that group
- `mean()` # calculate the mean value of a vector
- `!is.na()` # test if there are no missing values
- `print()` # print values to the console
- `min()` # return the minimum value of a vector
- `arrange()` # arrange rows by variables
- `desc()` # transform a vector into a format that will be sorted in descending order
- `count()` # counts the total number of records for each category
- `spread()` # reshape a data frame by a key-value pair across multiple columns
- `gather()` # reshape a data frame by collapsing into a key-value pair
- `n_distinct()` # get a count of unique values
- `write_csv()` # save to a csv formatted file

### Lesson 4 – Data visualization with ggplot2

- `read_csv()` # load a csv formatted file into R memory
- `ggplot2(data= , aes(x= , y= )) + geom_point( ) + facet_wrap( ) + theme_bw() + theme()`
- `aes()` # by selecting the variables to be plotted and the variables to define the presentation such as plotting size, shape color, etc.
- `geom_` # graphical representation of the data in the plot (points, lines, bars). To add a geom to the plot use + operator
- `facet_wrap()` # allows to split one plot into multiple plots based on a factor included in the dataset

- `labs()` # set labels to plot
- `theme_bw()` # set the background to white
- `theme()` # used to locally modify one or more theme elements in a specific ggplot object
- `+` # arrange ggplots horizontally
- `/` # arrange ggplots vertically
- `plot_layout()` # set width and height of individual plots in a patchwork of plots
- `ggsave()` # save a ggplot