# R Training Cheat Sheet

| Topic | Statement | Explanation and Sample |
| --- | --- | --- |
| Help | ?mean | Get help for the mean function  Script: R Base 01.R line #3 |
| Help | help.search('weighted mean') | Search the help files for the word within quotes.  Script: R Base 01.R line #5 |
| Help | help.package('dplyr') | Get help from the R Package within quotes.  Script: R Base 01.R line #7 |
| Libraries & Packages | install.packages('dplyr') | Download and install the R package within quotes. It downloads the package from CRAN (Comprehensive R Archive Network) that is a network of FTP servers in the web to store code, libraries and documentations related to R Language.  Script: R Base 01.R line #12 |
| Libraries & Packages | library(dplyr) | Load the package into the session (program) making available all functions to use.  Script: R Base 01.R line #14 |
| Libraries & Packages | dplyr::select | Use a particular function for a package. This reference is only required with there are more that a single package in usage that has duplicated functions name, else the function name (e.g. select) can be used directly.  Script: R Base 01.R line #16 |
| Libraries & Packages | data(iris) | Load a R Built-in Dataset, normally used for training. Iris is a classic database with flours details those are used for R Training. These databases are stored on PROMISE, meaning they are part of R installation.  Script: R Base 01.R line #18 |
| Directory | getwd() | Get the current project (working) directory.  Script: R Base 01.R line #23 |
| Directory | setwd('C://file\_path') | Set the project (working) directory.  Script: R Base 01.R line #25 |
| Variables | var1 <- “hello” | Assign the string “hello” to variable (var1)  Script: R Base 01.R line #30 |
| Variables | var2 <- 10 | Assign the value 10 to variable (var2)  Script: R Base 01.R line #32 |
| Variables | var3 <- var2 + 30 | Assign the sum of variable (var2) + 30  Script: R Base 01.R line #34 |
| Variables | var3 | Display the variable (var3) data  Script: R Base 01.R line #36 |
| Vectors | ve1 <- c(2,4,6) | Creates the vector (ve1) with number 2, 4 and 6. (e.g. 2,4,6)  Script: R Base 01.R line #41 |
| Vectors | ve2 <- c("A",4,5) | Creates the vector (ve2) with characters “A”, “4” and “5”. See that different from the previous sample, the fact that one of the elements is a character, converts automatically the other ones to character. (e.g. “A”,”4”,”5”)  Script: R Base 01.R line #45 |
| Vectors | ve3 <- 2:4 | Creates the vector (ve3) with integers from 2 to 4 (e.g. 2,3,4)  Script: R Base 01.R line #49 |
| Vectors | ve4 <- seq(2,5, by=0.5) | Creates the vector (ve4) with a numerical sequency from 2 to 5 with steps of 0.5 (e.g. 2.0,2.5,3.0,3.5,4.0,4.5,5.0)  Script: R Base 01.R line #53 |
| Vectors | ve5 <- rep(1:2, times=3) | Creates the vector (ve5) repeating the sequence 1,2 by 3 times. (e.g. 1,2,1,2,1,2)  Script: R Base 01.R line #57 |
| Vectors | ve6 <- rep(1:2, each=3) | Creates the vector (ve6) repeating 3 times each number in the sequence. (e.g. 1,1,1,2,2,2)  Script: R Base 01.R line #61 |
| Vectors | ve6 | Display the vector (ve6) data  Script: R Base 01.R line #63 |
| Vectors Functions | sort(vf1) | Sorts the vector (vf1)  Script: R Base 01.R line #69 |
| Vectors Functions | rev(vf1) | Reverse the order of vector (vf1)  Script: R Base 01.R line #71 |
| Vectors Functions | table(vf1) | Makes a table and count elements of vector (vf1)  Script: R Base 01.R line #73 |
| Vectors Functions | unique(vf1) | Remove the duplicates (keep only unique values) of vector (vf1)  Script: R Base 01.R line #75 |
| Vectors Elements Selection |  |  |