R Cheat Sheet

MSI’s Data Science Team

# Key Terms & Concepts

**R**

R is a programming language and software environment for statistical computing and graphics. It is widely used in data analysis, statistical modeling, and visualization.

**Posit Cloud**

An online platform for running R without installing it on your computer.

**RStudio**

A user-friendly interface for working with R, making it easier to write code, manage files, and visualize data.

**Git**

A version control system used for tracking changes in code. RStudio integrates with Git to help manage project versions.

**Terminal**

The terminal in RStudio provides a command-line interface where users can run shell commands directly within the RStudio environment.

**Environment**

The environment in RStudio is where all your variables, datasets, and functions are stored during your session.

* You can view it in the *Environment* tab in RStudio.

**Script**

A script is a file where you write and save your R code. You can execute parts of it instead of typing directly in the console.

* Shortcut: Ctrl + Shift + N (Windows) to open a new script.

**Console**

The console is where you type and run R commands.

* Example: 2 + 2 will return [1] 4.

**Packages**

Packages in R are essential tools for expanding the capabilities of R by adding new functions, datasets, and even compiled code. They are a fundamental part of the R ecosystem, allowing users to perform a wide range of tasks, from data manipulation to statistical analysis and graphical representation.

**Comments**

To add comments to an R script, use # followed by your comment.

# This is a comment in R

Comments help explain code and are ignored when the script runs.

**Creating an Object**

In R, objects store values, data, or functions. To create an object, use <-.

x <- 10 # Assigns the value 10 to x

Objects allow you to reuse and manipulate stored data.

**Piping Operator (%>%)**

The piping operator (%>%) is used to pass the output of one function into another, improving code readability.

data %>% filter(age > 30) %>% select(name, age)

This allows functions to be chained together in a clear and structured way.

**c() Function**

In R, the c() function is used to combine or concatenate values into a vector. It's a fundamental function for creating vectors of various types, such as numeric, character, or logical vectors.

numbers <- c(1, 2, 3, 4, 5) # Creates a numeric vector

names <- c("Alice", "Bob", "Charlie") # Creates a character vector

Vectors are the basic data structure in R and are used extensively in data analysis.

# Basic Commands

**Setting and Getting the Working Directory**

The working directory is where R looks for files and saves outputs.

* Set the working directory:

setwd("C:/Users/YourName/Documents/MyProject")

* Get the current working directory:

getwd()

**Installing and Loading Packages**

* Installed using:

install.packages("tidyverse")

* Loaded into R using:

library(tidyverse)

**Reading Data**

**CSV (Comma-Separated Values)**

data <- read.csv("datafile.csv")

Loads a CSV file into R, storing it as an object named data.

**SPSS (.sav)**

library(haven)

data <- read\_sav("datafile.sav")

Uses the haven package to load an SPSS dataset.

**Stata (.dta)**

library(haven)

data <- read\_dta("datafile.dta")

Loads a Stata dataset into R.

**R’s Internal Format (.rds)**

data <- readRDS("datafile.rds")

Loads an .rds file, which preserves R-specific data structures.

**Using External Scripts (source())**

Executes an external R script.

source("script.R")

Runs script.R, which can contain additional functions or data processing steps.

# Exploring Data

**Reviewing Variable Names**

names(data)

Lists column names of a dataset.

**View Dataset Structure**

str(data) # Shows structure of dataset

head(data) # Shows first few rows

* str() provides an overview of data types and structures.
* head() displays the first few rows.

**Filtering Data (filter())**

Filters rows based on a given condition.

library(dplyr)

data\_filtered <- filter(data, age > 30)

* filter(data, age > 30): Keeps rows where age is greater than 30.
* data\_filtered <-: Stores the filtered dataset.

**Selecting Data (select())**

Extracts specific columns.

data\_selected <- select(data, name, age)

Keeps only name and age columns from data.

**Summarizing Data (summarise())**

Aggregates data into summary statistics.

data\_summary <- summarise(data, average\_age = mean(age, na.rm = TRUE))

Computes the mean of age, ignoring missing values (na.rm = TRUE).