Day 3 Cheatsheet

Subsetting Data in R

Functions

Library/Package	Piece of code	Example of usage	What it does
Base R	`nrow(x); ncol(x)`	`nrow(x); ncol(x)`	Get the number of rows and the number of columns in an object `x`, respectively.
Base R	`dim(x)`	`dim(x)`	Get the number of rows *and* number of columns in an object `x`
`dplyr`	`glimpse(x)`	`glimpse(mtcars)`	Get an overview of data frame `x`
`dplyr`	`slice_sample(x)`	`slice_sample(mtcars)`	See a random subset of the rows of `x`
Base R	`data.frame()`	`df <- data.frame(1:3)`	Creates a data frame where the named arguments will be the same length.
Base R	`tibble()`	`tibble(mtcars)`	Creates a tibble from a data.frame or matrix.
`tibble`	`column_to_rownames()`	`df <- df %>% column_to_rownames('existing_variable_name')`	Transforms an existing column variable's name string into the rownames.
`tibble`	`rownames_to_column()`	`df <- df %>% column_to_rownames('new_variable_name')`	Transforms the rownames of a data frame into a column (which is added to the start of the data frame). The string supplied as an argument will be the name of the new column.
`dplyr`	`rename()`	`df <- dplyr::rename(df, MPG = mpg)`	Renames designated columns while keeping all variables of the data.frame
`dplyr`	`pull()`	`pull(df, 'existing_variable_name')`	Extract a column as a vector
`dplyr`	`select()`	`select(df, 'existing_variable_name')`	Selects columns that match the specified argument
`dplyr`	`filter()`	`filter(df, mpg > 20)`	Returns a subset of rows matching the conditions of the specified logical argument

Library/Package	Piece of code	Example of usage	What it does
Base `R`	`==, <=, >=, !=`	`filter(df, mpg > 20)`	These are binary operators which allow for the comparison of values in an object. They are handy for use with `dplyr::filter()`
Base `R`	`%in%`	`filter(df, mpg %in% c(20,21,22))`	Checks if the given value(s) on the left side of the operator are in the vector or other R object defined on the right side of the operator. It returns a logical `TRUE` or `FALSE` statement.
`dplyr`	`%>%`	`df <- df %>% select('new_variable_name')`	Funnels a data.frame through tidyverse operations
`dplyr`	`mutate()`	`df <- mutate(df, newcol = wt/2.2)`	Adds a new column that is a function of existing columns
`dplyr`	`relocate()`	`df_carb <- relocate(.data = df, wt, .before = mpg)`	Reorder columns in a data frame or tibble
`dplyr`	`arrange()`	`df <- arrange(df, mpg)`	Reorders rows in ascending order. `arrange(desc())` would reorder rows in descending order.
`dplyr`	`case_when()`	`df <- arrange(df, mpg)`	This function allows you to vectorise multiple `if_else()` statements. If no cases match, NA is returned.
Base `R`	`colnames()`	`colnames(df)`	Gets or sets the column names of a matrix or data frame.

- See tidyselect helpers for handy things to use with `select()`.

* This format was adapted from the cheatsheet format from AlexsLemonade.