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)	<pre>nrow(x); ncol(x)</pre>	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)	<pre>glimpse(mtcars)</pre>	Get an overview of data frame x
dplyr	<pre>slice_sample(x)</pre>	<pre>slice_sample(mtcars)</pre>	See a random subset of the rows of x
Base R	data.frame()	<pre>df <- data.frame(1:3)</pre>	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	<pre>column_to_rownames()</pre>	df <- df %>%	Transforms an existing
		column_to_rownames('exis	tingunartable lname') string into the rownames.
tibble	rownames_to_column()	<pre>df <- df %>% column_to_rownames('new_')</pre>	Transforms the varouable safe and 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()	<pre>df <- rename(df, MPG = mpg)</pre>	Renames designated columns while keeping all variables of the data.frame
dplyr	pull()	<pre>pull(df, 'existing_variable_name';</pre>	Extract a column as a vector
dplyr	select()	select(df, 'existing_variable_name'	Selects columns that
dplyr	filter()	<pre>filter(df, mpg > 20)</pre>	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 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	%>%	<pre>df <- df %>% select('new_variable_name</pre>	Funnels a data frame nethrough tidyverse operations
dplyr	<pre>mutate()</pre>	<pre>df <- mutate(df, newcol = wt/2.2)</pre>	Adds a new column that is a function of existing columns
dplyr	relocate()	<pre>df_carb <- relocate(.data = df, wt, .before = mpg)</pre>	Reorder columns in a data frame or tibble
dplyr	arrange()	<pre>df <- arrange(df, mpg)</pre>	Reorders rows in ascending order. arrange(desc()) would reorder rows in descending order.
dplyr	<pre><- select(df, starts_with("B"), year) This function allows you to select columns based on a particular column name pattern. BaseR </pre>	names of a matrix or data frame.	t.r-lib.org/reference/starts_with.h

[•] See tidyselect helpers for handy things to use with select().

 $^{\ ^*}$ This format was adapted from the cheat sheet format from AlexsLemonade.