The Weekend Data Course: Essential R Functions

Gregory Sward

January 30, 2018

1 Help Functions

Function	Package	Description
`?`	**base**	Opens help document for a function.
`??`	**base**	Searches help pages.
`citation()`	**utils**	Returns info on how to cite an R library.
`example()`	**utils**	Runs example code for a given function.

2 Environment Functions

Function	Package	Description
`getwd()`	**base**	Returns filepath of current working directory.
`ls()`	**base**	Returns a list of the objects in the environment.
`rm()`	**base**	Remove an object from the environment.
`setwd()`	**base**	Sets the filepath of the working directory.

3 Importing Data & Exporting Data

Function	Package	Description
`dev.off()`	**grDevices**	Shuts off the current graphics device.
`odbcConnect()`	**RODBC**	Opens a connection to ODBC databases.
`pdf()`	**grDevices**	Starts graphics driver for producing PDF graphics.
`read.csv()`	**utils**	Creates a data frame from a CSV file.
`readRDS()`	**base**	Restores a single R object.
`read.table()`	**utils**	Creates a data frame from a file.
`saveRDS()`	**base**	Saves a single R object.
`sqlFetch()`	**RODBC**	Read data from ODBC database.
`write.csv()`	**utils**	Saves R object to CSV file.

4 Creating Data Objects

Function	Package	Description
`c()`	**base**	Combines arguments to form vector.
`cbind()`	**base**	Combines data objects by columns.
`data.frame()`	**base**	Creates a data frame object.
`list()`	**base**	Combines arguments to form list.
`rbind()`	**base**	Combines data objects by rows.
`rep()`	**base**	Replicates the values in a vector.
`seq()`	**base**	Creates regular sequences.
`tibble()`	**tibble**	Creates a tibble data frame.

5 Examining Data Structure

Function	Package	Description
`class()`	**base**	Returns the class of an object.
`dim()`	**base**	Returns or sets the dimensions of an object.
`head()`	**utils**	Returns the observations of an object.
`levels()`	**base**	Get or set the levels of a factor.
`length()`	**base**	Get or set the length of vector or list.
`mode()`	**base**	Returns the mode of an object.
`str()`	**base**	Displays the structure of an object.
`summary()`	**base**	Displays summary information for object.

6 Logical Operators and Set Functions

Function	Package	Description
`x>y`	**base**	Is x greater than y?
`x>=y`	**base**	Is x greater than or equal to y?
`x<y`	**utils**	Is x less than y?
`x<=y`	**base**	Is x less than or equal to y?
`x==y`	**base**	Is x equal to y?
`x!=y`	**base**	Is x not equal to y?
`x|y`	**base**	Is x or y TRUE?
`x&y`	**base**	Is x and y TRUE?
`any(x`	**base**	Is any element of x TRUE?
`all(x)`	**base**	Are all elements of x TRUE?
`union(x,y)`	**base**	Returns the combined set of values.
`intersect(x,y)`	**base**	Returns the set of common values.
`setdiff(x,y)`	**base**	Returns values of x that are not found in y.
`setequal(x,y)`	**base**	Are the sets equal?
`is.element(x,y)`	**base**	Returns a boolean vector whether an element of x is found in y.

7 Loops and Control Statements

Function	Package	Description
`for(var in seq)`	**base**	Loop over a sequence of values.
`while(cond)`	**base**	Loop while the lopp condition is TRUE.
`break`	**base**	Exit loop.
`next`	**base**	Stops current iterations and skips to next iteration.
`if(cond)`	**base**	Evaluates block of code if condition is true.
`ifelse()`	**base**	Vectorized if-else statement.

8 Data Tidying and Data Manipulation

Function	Package	Description
`gather()`	**tidyr**	Collapses multiple columns into key-value pairs.
`spread()`	**tidyr**	Spread a key-value pair across multiple columns.
`arrange()`	**dplyr**	Sorts rows by variables.
`filter()`	**dplyr**	Returns rows/cases where given conditions are true.
`group_by()`	**dplyr**	Creates a grouped table. Further operations are performed "by group".
`mutate()`	**dplyr**	Returns the class of an object.
`select()`	**dplyr**	Returns the class of an object.
`summarize()`	**dplyr**	Creates a new data frame with specified summary variables.
`ungroup()`	**dplyr**	Removes a table's grouping.