Switching from Base R to tidyverse

October 5, 2017 in Coding

One of the most transformative changes in my coding has been switching to tidyverse packages from base R. Tidy code is easier to write, read, maintain and almost always faster than the base R counterparts. While there has been some debate on whether base R should be taught to newcomers first or should they jump right into tidyverse, I havent heard anyone deny that ultimately, everyone should be be using tidyverse. A lot of online courses continue to use base R to teach students R programming. Hopefully the table below helps you switch from base R to their equivalent tidyverse commands when you are ready.

Couple of notes before we start. The list below is not exhaustive (best to read package documentation for that). For instance, it doesnt cover lubridate (which covers date/time related functions), forcats (which covers everything you would want to do to factors), broom (which tidies up messy R objects), modelr (which has helper functions for creating models) or ggplot. I also use data frame and tibble interchangeably, although they are obviously different.

Base R command	Tidyverse Command	What it does and why you should use the tidyverse version	Comment
read.csv()	read_csv()	reads in a csv file, but its much faster, shows progress bar for large files, can automatically parse data types	also see read_delim(), read_tsv() and readxl::read_xlsx()

sort(), order()	arrange()	sort column(n) within a data frame	see also order_by()
mtcars$mpg = ...	mutate()	modify a column	see also transmute() which drops existing variables
mtcars[,c("mpg", "am")], subset()	select(), rename()	select or rename columns	see also pull()
mtcars[mtcars$am == 1,], subset()	filter()	select rows based on a criterion	
Base R command	**Tidyverse Command**	**What it does and why you should use the tidyverse version**	**Comment**
aggregate()	summarise(), summarize(), do()	reduce grouped values to a single value	see also varaints like summarize_if()
ifelse()	if_else(), case_when()	standand vectorized if else, but stricter than base version	see also near()
unique()	distinct()	finds unique rows in a data frame, but its much, faster	
length(unique())	n_distinct()	count the number of distinct values in a vector, faster	

sample(), sample.int()	sample_n(), sample_frac()	sample n rows or a fraction of rows from a dataframe	
all.equal()	all_equal()	checks if two vectors are the same	
merge()	inner_join(), left_join()	perform joins, much faster, verbose, and row order is maintain	see also right_join(), full_join(), semi_join(), anti_join()
Base R command	**Tidyverse Command**	**What it does and why you should use the tidyverse version**	**Comment**
rbind(), cbind()	bind_rows(), bind_cols()	concatenate two dataframes along rows or columns, much faster	
x >= left & x <= right	between()	easier to read and faster implementation for larege datasets	see also near()
nrow(), sum()	tally(), count(), add_tally(), add_count()	count or sum up rows	
c()	combine()	combine into a vector	
extends base R	cumall(), cumany(),	extends base R collection of cumsum(),	

	cummean()	cumprod() etc	
mtcars$mpg[1,] etc	first(), last(), n(), top_n()	works within groups, allows you to order by another column(s) and provide defaults for missing values	
split(), aggregate()	group_by()	create a grouped data frame (tibble) to perform operations on groups	see also ungroup()
Base R command	**Tidyverse Command**	**What it does and why you should use the tidyverse version**	**Comment**
intersect(), union()	intersect(), union()	set operations, but dplyr works on data frames as well	
mtcars\(mpg2 = c(NA, mtcars\)mpg[1:nrow(mtcars)-1])	lead(), lag()	No equivalent command in base R, easier to read	
ifelse(..., NA)	na_if()	convert a value to NA	
switch()	recode()	change certain values in your vector	see also forcats package when dealing with factors
		select rows	

mtcars[3:5,]	slice()	bases on row numbers	
seq_along(), quantile()	row_number(), ntile(), min_ran() etc	add rankings in various ways, much richer set of rankings supported than base r	
no easy way	complete(), expand()	expands the dataframe so that supplied columns are completely filled out	often used with nesting(), see also full_seq()
Base R command	**Tidyverse Command**	**What it does and why you should use the tidyverse version**	**Comment**
expand.grid()	crossing()	create a data frame of all possible combinations of supplied vectors	
ifelse(is.na(...), ...)	drop_na(), replace_na()	drop rows with missing values or convert NAs to supplied values	see also fill(), coalesce()
some mix of paste/strsplit	separate(), unite()	separate two columns based on regex or combine two columns into one	

reshape2::dcast()	spread()	convert long (tidy) data into wide (untidy) format
reshape2::melt()	gather()	convert wide (untidy) data into long(tidy) format
replicate()	rerun()	run an expression n number of times
unlist(lapply(x, `[[`, n))	pluck()	extract elements out of a list
lapply(), sapply()	map(), map2()	apply a function to a set of values, working with lists
paste0()	glue()	combine two strings together, but much more powerful because it allows for expressions

Base R command	Tidyverse Command	What it does and why you should use the tidyverse version	Comment

TAGGED IN

R tidyverse