My favorite R packages for data visualization and munging

| Package | Category | Description | Sample Use | Author |
| --- | --- | --- | --- | --- |
| [devtools](https://github.com/hadley/devtools) | package development, package installation | While devtools is aimed at helping you create your own R packages, it's also essential if you want to easily install other packages from GitHub. Install it! Requires [Rtools](http://cran.r-project.org/bin/windows/Rtools/) on Windows and [XCode](https://developer.apple.com/xcode/downloads/) on a Mac. On CRAN. | install\_github("rstudio/leaflet") | Hadley Wickham & others |
| [remotes](https://github.com/r-pkgs/remotes) | package installation | If all you want is to install packages from GitHub, devtools may be a bit of a heavyweight. remotes will easily install from GitHub as well as Bitbucket and some others. On CRAN. ([ghit](https://cran.r-project.org/web/packages/ghit/) is another option, but is GitHub-only.) | remotes::install\_github("mangothecat/franc") | Gabor Csardi & others |
| [installr](https://github.com/talgalili/installr/) | misc | Windows only: Update your installed version of R from within R. On CRAN. | updateR() | Tal Galili & others |
| [reinstallr](https://github.com/calligross/reinstallr) | misc | Seeks to find packages that had previously been installed on your system and need to be re-installed after upgrading R. CRAN. | reinstallr() | Calli Gross |
| [readxl](https://github.com/hadley/readxl/) | data import | Fast way to read Excel files in R, without dependencies such as Java. CRAN. | read\_excel("my-spreadsheet.xls", sheet = 1) | Hadley Wickham |
| [googlesheets](https://github.com/jennybc/googlesheets) | data import, data export | Easily read data into R from Google Sheets. CRAN. | mysheet <- gs\_title("Google Spreadsheet Title") mydata <- mydata <- gs\_read(mysheet, ws = “WorksheetTitle”) | Jennifer Bryan |
| [readr](https://github.com/hadley/readr) | data import | Base R handles most of these functions; but if you have huge files, this is a speedy and standardized way to read tabular files such as CSVs into R data frames, as well as plain text files into character strings with read\_file. CRAN. | read\_csv(myfile.csv) | Hadley Wickham |
| [rio](http://cran.r-project.org/web/packages/rio/vignettes/rio.html) | data import, data export | rio has a good idea: Pull a lot of separate data-reading packages into one, so you just need to remember 2 functions: import and export. CRAN. | import("myfile") | Thomas J. Leeper & others |
| [Hmisc](http://biostat.mc.vanderbilt.edu/wiki/Main/HmiscNew) | data analysis | There are a number of useful functions in here. Two of my favorites: describe, a more robust summary function, and Cs, which creates a vector of quoted character strings from unquoted comma-separated text. Cs(so, it, goes) creates c("so", "it", "goes"). CRAN. | describe(mydf) Cs(so, it, goes) | Frank E Harrell Jr & others |
| [datapasta](https://github.com/MilesMcBain/datapasta) | data import | Data copy and paste: Meet reproducible research. If you've copied data from the Web, a spreadsheet, or other source into your clipboard, datapasta lets you paste it into R as an R object, with the code to reproduce it. It includes RStudio add-ins as well as command-line functions for transposing data, turning it into markdown format, and more. CRAN. | df\_paste() to create a data frame, vector\_paste() to create a vector. | Miles McBain |
| [sqldf](https://github.com/ggrothendieck/sqldf) | data wrangling, data analysis | Do you know a great SQL query you'd use if your R data frame were in a SQL database? Run SQL queries on your data frame with sqldf. CRAN. | sqldf("select \* from mydf where mycol > 4") | G. Grothendieck |
| [jsonlite](https://www.opencpu.org/posts/jsonlite-a-smarter-json-encoder/) | data import, data wrangling | Parse json within R or turn R data frames into json. CRAN. | myjson <- toJSON(mydf, pretty=TRUE) mydf2 <- fromJSON(myjson) | Jeroen Ooms & others |
| [XML](http://cran.r-project.org/web/packages/XML/) | data import, data wrangling | Many functions for elegantly dealing with XML and HTML, such as readHTMLTable. CRAN. | mytables <- readHTMLTable(myurl) | Duncan Temple Lang |
| [httr](https://github.com/hadley/httr) | data import, data wrangling | An R interface to http protocols; useful for pulling data from APIs. See the [httr quickstart guide](https://cran.r-project.org/web/packages/httr/vignettes/quickstart.html). CRAN. | r <- GET("http://httpbin.org/get") content(r, "text") | Hadley Wickham |
| [quantmod](http://www.quantmod.com/examples/intro/) | data import, data visualization, data analysis | Even if you're not interested in analyzing and graphing financial investment data, quantmod has easy-to-use functions for importing economic as well as financial data from sources like the Federal Reserve. CRAN. | getSymbols("AITINO", src="FRED") | Jeffrey A. Ryan |
| [tidyquant](https://business-science.github.io/tidyquant/) | data import, data visualization, data analysis | Another financial package that's useful for importing, analyzing and visualizing data, integrating aspects of other popular finance packages as well as tidyverse tools. With thorough documentation. CRAN. | aapl\_key\_ratios <- tq\_get("AAPL", get = "key.ratios") | Matt Dancho |
| [rvest](http://cran.r-project.org/web/packages/rvest/vignettes/selectorgadget.html) | data import, web scraping | Web scraping: Extract data from HTML pages. Inspired by Python's Beautiful Soup. Works well with Selectorgadget. CRAN. | [See the package vignette](http://cran.r-project.org/web/packages/rvest/vignettes/selectorgadget.html) | Hadley Wickham |
| [dplyr](http://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html) | data wrangling, data analysis | The essential data-munging R package when working with data frames. Especially useful for operating on data by categories. CRAN. | [See the intro vignette](http://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html) | Hadley Wickham |
| [purrr](http://purrr.tidyverse.org/) | data wrangling | purrr is a relatively new package aimed at replacing [plyr](http://plyr.had.co.nz/) and some base R apply functions for doing operations for running . It's more complex to learn but also has more functionality. CRAN. | map\_df(mylist, myfunction) More: [Charlotte Wickham's purr tutorial video](https://www.rstudio.com/resources/videos/happy-r-users-purrr-tutorial/), the [purrr cheat sheet](https://github.com/rstudio/cheatsheets/raw/master/purrr.pdf) PDF download. | Hadley Wickham |
| [reshape2](https://github.com/hadley/reshape) | data wrangling | Change data row and column formats from "wide" to "long"; turn variables into column names or column names into variables and more. The tidyr package is a newer, more focused option, but I still use reshape2. CRAN. | [See my tutorial](http://www.computerworld.com/article/2486425/business-intelligence-4-data-wrangling-tasks-in-r-for-advanced-beginners.html?page=6#reshaping) | Hadley Wickham |
| [tidyr](https://github.com/hadley/tidyr) | data wrangling | While I still prefer reshape2 for general re-arranging, tidy won me over with specialized functions like fill (fill in missing columns from data above) and replace\_na. CRAN. | See examples in [this blog post](http://blog.rstudio.org/2015/09/13/tidyr-0-3-0/). | Hadley Wickham |
| [magrittr](https://github.com/tidyverse/magrittr) | data wrangling | This package gave us the %>% symbol for chaining R operations, but it's got other useful operators such as %<>% for mutating a data frame in place and and . as a placeholder for the original object being operated upon. CRAN. | mydf %<>% mutate(newcol = myfun(colname)) | Stefan Milton Bache & Hadley Wickham |
| [validate](https://github.com/data-cleaning/validate) | data wrangling | Intuitive data validation based on rules you can define, save and re-use. CRAN. | See the [introductory vignette](https://cran.r-project.org/web/packages/validate/vignettes/intro.html). | Mark van der Loo & Edwin de Jonge |
| [testthat](https://github.com/hadley/testthat) | programming | Package that makes it easy to write unit tests for your R code. CRAN. | See the [testing chapter](http://r-pkgs.had.co.nz/tests.html) of Hadley Wickham's book on R packages. | Hadley Wickham |
| [data.table](https://github.com/Rdatatable/data.table/wiki/Getting-started) | data wrangling, data analysis | Popular package for heavy-duty data wrangling. While I typically prefer dplyr, data.table has many fans for its speed with large data sets. CRAN. | [Useful tutorial](http://blog.yhathq.com/posts/fast-summary-statistics-with-data-dot-table.html) | Matt Dowle & others |
| [stringr](https://github.com/hadley/stringr) | data wrangling | Numerous functions for text manipulation. Some are similar to existing base R functions but in a more standard format, including working with regular expressions. Some of my favorites: str\_pad and str\_trim. CRAN. | str\_pad(myzipcodevector, 5, "left", "0") | Hadley Wickham |
| [lubridate](https://github.com/hadley/lubridate) | data wrangling | Everything you ever wanted to do with date arithmetic, although understanding & using available functionality can be somewhat complex. CRAN. | mdy("05/06/2015") + months(1) [More examples in the package vignette](http://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html) | Garrett Grolemund, Hadley Wickham & others |
| [zoo](http://cran.r-project.org/web/packages/zoo/) | data wrangling, data analysis | Robust package with a slew of functions for dealing with time series data; I like the handy rollmean function with its align=right and fill=NA options for calculating moving averages. CRAN. | rollmean(mydf, 7) | Achim Zeileis & others |
| [editR](https://github.com/swarm-lab/editR) | data display | Interactive editor for [R Markdown](http://rmarkdown.rstudio.com/) documents. Note that [R Markdown Notebooks](http://rmarkdown.rstudio.com/r_notebooks.html) are another useful way to generate Markdown interactively. editR is on GitHub. | editR("path/to/myfile.Rmd") | Simon Garnier |
| [knitr](http://yihui.name/knitr/) | data display | Add R to a markdown document and easily generate reports in HTML, Word and other formats. A must-have if you're interested in reproducible research and automating the journey from data analysis to report creation. CRAN. | See the [Minimal Examples](http://yihui.name/knitr/demo/minimal/) page. | Yihui Xie & others |
| [officeR](https://github.com/davidgohel/officer) | data display | Import and edit Microsoft Word and PowerPoint documents, making it easy to add R-generated analysis and visualizations to existing as well as new reports and presentations. CRAN. | my\_doc <- read\_docx() %>%  body\_add\_img(src = myplot) [The package website](https://davidgohel.github.io/officer/index.html) has many more examples. | David Gohel |
| [listviewer](http://www.buildingwidgets.com/blog/2015/4/14/week-15-listviewer) | data display, data wrangling | While RStudio has since added a list-viewing option, this HTML widget still offers an elegant way to view complex nested lists within R. GitHub timelyportfolio/listviewer. | jsonedit(mylist) | Kent Russell |
| [DT](https://rstudio.github.io/DT/) | data display | Create a sortable, searchable table in one line of code with this R interface to the jQuery DataTables plug-in. GitHub rstudio/DT. | datatable(mydf) | RStudio |
| [ggplot2](http://ggplot2.org/) | data visualization | Powerful, flexible and well-thought-out dataviz package following 'grammar of graphics' syntax to create static graphics, but be prepared for a steep learning curve. CRAN. | qplot(factor(myfactor), data=mydf, geom="bar", fill=factor(myfactor)) See my [searchable ggplot2 cheat sheet](http://www.computerworld.com/article/2935394/business-intelligence/my-ggplot2-cheat-sheet-search-by-task.html) and [time-saving code snippets](http://www.computerworld.com/article/2936729/data-analytics/free-download-save-r-data-visualization-time-with-these-ggplot2-code-snippets.html). | Hadley Wickham |
| [patchwork](https://github.com/thomasp85/patchwork) | data visualization | Easily combine ggplot2 plots and keep the new, merged plot a ggplot2 object. plot\_layout() adds ability to set columns, rows, and relative sizes of each component graphic. GitHub. | plot1 + plot2 + plot\_layout(ncol=1) | Thomas Lin Pedersen |
| [ggiraph](http://davidgohel.github.io/ggiraph/) | data visualization | Make ggplot2 plots interactive with this extension's new geom functions such geom\_bar\_interactive and arguments for tooltips and JavaScript onclicks. CRAN. | g <- ggplot(mpg, aes( x = displ, y = cty, color = drv) ) my\_gg <- g + geom\_point\_interactive(aes(tooltip = model), size = 2) ggiraph(code = print(my\_gg), width = .7). | David Gohel |
| [dygraphs](https://rstudio.github.io/dygraphs/index.html) | data visualization | Create HTML/JavaScript graphs of time series - one-line command if your data is an xts object. CRAN. | dygraph(myxtsobject) | JJ Allaire & RStudio |
| [googleVis](https://github.com/mages/googleVis) | data visualization | Tap into the Google Charts API using R. CRAN. | mychart <- gvisColumnChart(mydata) plot(Column) [Numerous examples here](http://cran.r-project.org/web/packages/googleVis/vignettes/googleVis_examples.html) | Markus Gesmann & others |
| [metricsgraphics](http://hrbrmstr.github.io/metricsgraphics/) | data visualization | R interface to the metricsgraphics JavaScript library for bare-bones line, scatterplot and bar charts. GitHub hrbrmstr/metricsgraphics. | [See package intro](http://hrbrmstr.github.io/metricsgraphics/) | Bob Rudis |
| [RColorBrewer](http://cran.r-project.org/web/packages/RColorBrewer/index.html) | data visualization | Not a designer? RColorBrewer helps you select color pallettes for your visualizations. CRAN. | [See Jennifer Bryan's tutorial](http://www.stat.ubc.ca/~jenny/STAT545A/block14_colors.html#rcolorbrewer) | Erich Neuwirth |
| [sf](https://github.com/r-spatial/sf/) | mapping, data wrangling | This package makes it much easier to do GIS work in R. Simple features protocols make geospatial data look a lot like regular data frames, while various functions allow for analysis such as determining whether points are in a polygons. A GIS game-changer for R. CRAN. | See the package vignettes, starting with the introduction, [Simple Features for R](https://r-spatial.github.io/sf/articles/sf1.html). | Edzer Pebesma & others |
| [leaflet](http://rstudio.github.io/leaflet/) | mapping | Map data using the Leaflet JavaScript library within R. GitHub rstudio/leaflet. | [See my tutorial](http://www.computerworld.com/article/2893271/5-data-visualizations-in-5-minutes-each-in-5-lines-or-less-of-r.html) | RStudio |
| [ggmap](https://github.com/dkahle/ggmap) | mapping | Although I don't use this package often for its main purpose of pulling down background map tiles, it's my go-to for geocoding up to 2,500 addresses with the Google Maps API with its geocode and mutate\_geocode functions. CRAN. | geocode("492 Old Connecticut Path, Framingham, MA") | David Kahle &Hadley Wickham |
| [tmap & tmaptools](http://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html) | mapping | These package offer an easy way to read in shape files and join data files with geographic info, as well as do some exploratory mapping. Recent functionality adds support for simple features, interactive maps and creating leaflet objects. Plus, tmaptools::palette\_explorer() is a great tool for picking ColorBrewer palettes. CRAN. | [See the package vignette](http://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html) or my [mapping in R tutorial](http://www.computerworld.com/article/3038270/data-analytics/create-maps-in-r-in-10-fairly-easy-steps.html) | Martijn Tennekes |
| [mapsapi](https://cran.r-project.org/web/packages/mapsapi/vignettes/intro.html) | mapping, data wrangling | This interface to the Google Maps Direction and Distance Matrix APIs let you analyze and map distances and driving routes. CRAN. | google\_directions( origin = c(my\_longitude, my\_latitude), destination = c(my\_address), alternatives = TRUE Also [see the vignette](https://cran.r-project.org/web/packages/mapsapi/vignettes/intro.html) | Michael Dorman |
| [tidycensus](https://github.com/walkerke/tidycensus) | mapping, data wrangling | Want to analyze and map U.S. Census Bureau data from 5-year American Community Surveys or 10-year censuses? This makes it easy to download numerical and geospatial info in R-ready format. CRAN. | See [Basic usage of tidycensus](https://walkerke.github.io/tidycensus/articles/basic-usage.html). | Kyle E. Walker |
| [glue](https://github.com/tidyverse/glue) | data wrangling | Main function, also glue, evaluates variables and R expressions within a quoted string, as long as they're enclosed by {} braces. This makes for an elegant paste() replacement. CRAN. | glue("Today is {Sys.Date()}") | Jim Hester |
| [rga](https://github.com/skardhamar/rga) | Web analytics | Use Google Analytics with R. GitHub skardhamar/rga. | [See package README file](https://github.com/skardhamar/rga) and [my tutorial](http://www.computerworld.com/article/2486018/business-intelligence-how-to-extract-custom-data-from-the-google-analytics-api.html) | Bror Skardhamar |
| [RSiteCatalyst](http://randyzwitch.com/rsitecatalyst/) | Web analytics | Use Adobe Analytics with R. GitHub randyzwitch/RSiteCatalyst. | [See intro video](http://randyzwitch.com/rsitecatalyst/) | Randy Zwitch |
| [roxygen2](http://roxygen.org/#documentation) | package development | Useful tools for documenting functions within R packages. CRAN. | [See this short, easy-to-read blog post on writing R packages](http://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/) | Hadley Wickham & others |
| [shiny](http://shiny.rstudio.com/) | data visualization | Turn R data into interactive Web applications. I've seen some nice (if sometimes sluggish) apps and it's got many enthusiasts. CRAN. | [See the tutorial](http://shiny.rstudio.com/tutorial/) | RStudio |
| [flexdashboard](http://rmarkdown.rstudio.com/flexdashboard/) | data visualization | If Shiny is too complex and involved for your needs, this package offers a simpler (if somewhat less robust) solution based on R Markdown. CRAN. | More info in [Using flexdashboard](http://rmarkdown.rstudio.com/flexdashboard/using.html) | JJ Allaire, RStudio & others |
| [openxlsx](https://github.com/awalker89/openxlsx) | misc | If you need to write to an Excel file as well as read, this package is easy to use. CRAN. | write.xlsx(mydf, "myfile.xlsx") | Alexander Walker |
| [gmodels](http://cran.r-project.org/web/packages/gmodels/gmodels.pdf) | data wrangling, data analysis | There are several functions for modeling data here, but the one I use, CrossTable, simply creates cross-tabs with loads of options -- totals, proprotions and several statistical tests. CRAN. | CrossTable(myxvector, myyvector, prop.t=FALSE, prop.chisq = FALSE) | Gregory R. Warnes |
| [janitor](https://github.com/sfirke/janitor) | data wrangling, data analysis | Basic data cleaning made easy, such as finding duplicates by multiple columns, making R-friendly column names and removing empty columns. It also has some nice tabulating tools, like adding a total row and doing tables with percentages and easy crosstabs. CRAN. | tabyl(mydf, sort = TRUE) %>% adorn\_totals("row") | Samuel Firke |
| [car](https://mran.microsoft.com/package/car/) | data wrangling | car's recode function makes it easy to bin continuous numerical data into categories or factors. While base R's cut accomplishes the same task, I find recode's syntax to be more intuitive - just remember to put the entire recoding formula within double quotation marks. dplyr's [case\_when() function](http://dplyr.tidyverse.org/reference/case_when.html) is another option worth considering. CRAN. | recode(x, "1:3='Low'; 4:7='Mid'; 8:hi='High'") | John Fox & others |
| [rcdimple](https://github.com/timelyportfolio/rcdimple) | data visualization | R interface to the dimple JavaScript library with numerous customization options. Good choice for JavaScript bar charts, among others. GitHub timelyportfolio/rcdimple. | dimple(mtcars, mpg ~ cyl, type = "bar") | Kent Russell |
| [foreach](https://cran.r-project.org/web/packages/foreach/) | data wrangling | Efficient - and intuitive if you come from another programming language - for loops in R. CRAN. | foreach(i=1:3) %do% sqrt(i) Also see [The Wonders of foreach](http://www.exegetic.biz/blog/2013/08/the-wonders-of-foreach/) | Revolution Analytics, Steve Weston |
| [scales](https://cran.r-project.org/web/packages/scales/) | data wrangling | While this package has many more sophisticated ways to help you format data for graphing, it's worth a download just for the comma(), percent() and dollar() functions. CRAN. | comma(mynumvec) | Hadley Wickham |
| [plotly](https://plot.ly/r/) | data visualization | R interface to the Plotly JavaScript library that was open-sourced in late 2015. Basic graphs have a distinctive look which may not be for everyone, but it's full-featured, relatively easy to learn (especially if you know ggplot2) and includes a ggplotly() function to turn graphs created with ggplot2 interactive. CRAN. | d <- diamonds[sample(nrow(diamonds), 1000), ]  plot\_ly(d, x = carat, y = price, text = paste("Clarity: ", clarity), mode = "markers", color = carat, size = carat) | Carson Sievert & others |
| [highcharter](http://jkunst.com/highcharter/) | data visualization | R wrapper for the robust and well documented Highcharts JavaScript library, one of my favorite choices for presentation-quality interactive graphics. The package uses ggplot2-like syntax, including options for handling both long and wide data, and comes with plenty of examples. Note that a [paid Highcharts license](https://shop.highsoft.com/highcharts) is needed to use this for commercial or government work (it's free for personal and non-profit projects). CRAN. . CRAN. | hchart(mydf, "charttype", hcaes(x = xcol, y = ycol, group = groupbycol)) | Joshua Kunst & others |
| [profvis](https://rstudio.github.io/profvis/) | programming | Is your R code sluggish? This package gives you a visual representative of your code line by line so you can find the speed bottlenecks. CRAN. | profvis({ your code here }) | Winston Chang & others |
| [tidytext](https://github.com/juliasilge/tidytext) | text mining | Elegant implementation of text mining functions using Hadley Wickham's "tidy data" principles. CRAN. | See [tidytextmining.com](http://tidytextmining.com/) for numerous examples. | Julia Silge & David Robinson |
| [diffobj](https://cran.r-project.org/web/packages/diffobj/vignettes/diffobj.html) | data analysis | Base R's identical() function tells you whether or not two objects are the same; but if they're not, it won't tell you why. diffobj gives you a visual representation of how two R objects differ. CRAN. | diffObj(x,y) | Brodie Gaslam & Michael B. Allen |
| [Prophet](https://facebookincubator.github.io/prophet/) | forecasting | I don't do much forecasting analysis; but if I did, I'd start with this package. CRAN. | See the [Quick start guide](https://facebookincubator.github.io/prophet/docs/quick_start.html#r-api). | Sean Taylor & Ben Letham at Facebook |
| [feather](https://blog.rstudio.org/2016/03/29/feather/) | data import, data export | This binary data-file format can be read by both Python and R, making data interchange easier between the two languages. It's also built for I/O speed. CRAN. | write\_feather(mydf, "myfile") | Wes McKinney & Hadley Wickham |
| [fst](http://www.fstpackage.org/) | data import, data export | Another alternative for binary file storage (R-only), fst was built for fast storage and retrieval, with access speeds above 1 GB/sec. It also offers compression that doesn't slow data access too much, as well as the ability to import a specific range of rows (by row number). CRAN. | write.fst(mydf, "myfile.fst", 100) | Mark Klik |
| [googleAuthR](http://code.markedmondson.me/googleAuthR/) | data import | If you want to use data from a Google API in an R project and there's not yet a specific package for that API, this is the place to turn for authenticating CRAN. | See examples on [the package website](http://code.markedmondson.me/googleAuthR/) and [this gist](https://gist.github.com/MarkEdmondson1234/0198e283cd228565f9313cf36f35c7ab) for use with Google Calendars. CRAN. | Mark Edmondson |
| [here](https://github.com/krlmlr/here) | misc | This package has one function with a single, useful purpose: find your project's working directory. Surprisingly helpful if you want your code to run on more than one system. CRAN. | my\_project\_directory <- here() | Kirill Müller |
| [cloudyR project](https://cloudyr.github.io/) | data import, data export | This is a collection of packages aimed at making it easier for R to work with cloud platforms such as Amazon Web Services, Google and Travis-CI. Some are already on CRAN, some can be found on GitHub. |  |  |