**Recommended Packages**

Many useful R function come in packages, free libraries of code written by R's active user community. To install an R package, open an R session and type at the command line

install.packages("<the package's name>")

R will download the package from CRAN, so you'll need to be connected to the internet. Once you have a package installed, you can make its contents available to use in your current R session by running

library("<the package's name>")

There are thousands of helpful R packages for you to use, but navigating them all can be a challenge. To help you out, we've compiled this guide to some of the best. We've used each of these, and found them to be outstanding – we've even written some of them. But you don't have to take our word for it, these packages are also some of the top most downloaded R packages.

To load data

[DBI](https://db.rstudio.com/dbi/) - The standard for for communication between R and relational database management systems. Packages that connect R to databases depend on the DBI package.

[odbc](https://db.rstudio.com/odbc/) - Use any ODBC driver with the odbc package to connect R to your database. Note: RStudio professional products come with [professional drivers](https://www.rstudio.com/products/drivers/) for some of the most popular databases.

[RMySQL](http://www.rdocumentation.org/packages/RMySQL/functions/RMySQL-package), [RPostgresSQL](http://www.rdocumentation.org/packages/RPostgresSQL" \o "Link: http://www.rdocumentation.org/packages/RPostgresSQL), [RSQLite](http://www.rdocumentation.org/packages/RSQLite" \o "Link: http://www.rdocumentation.org/packages/RSQLite) - If you'd like to read in data from a database, these packages are a good place to start. Choose the package that fits your type of database.

[XLConnect](https://cran.rstudio.com/web/packages/XLConnect), [xlsx](https://cran.rstudio.com/web/packages/xlsx) - These packages help you read and write Micorsoft Excel files from R. You can also just export your spreadsheets from Excel as .csv's.

[foreign](http://www.rdocumentation.org/packages/foreign) - Want to read a SAS data set into R? Or an SPSS data set? Foreign provides functions that help you load data files from other programs into R.

[haven](https://haven.tidyverse.org/) - Enables R to read and write data from SAS, SPSS, and Stata.

R can handle plain text files – no package required. Just use the functions read.csv, read.table, and read.fwf. If you have even more exotic data, consult the CRAN [guide](https://cran.rstudio.com/doc/manuals/R-data.pdf) to data import and export.

*For more information about using R with databases see*[*db.rstudio.com*](https://db.rstudio.com/)*.*

To manipulate data

[dplyr](http://blog.rstudio.org/2014/01/17/introducing-dplyr/) - Essential shortcuts for subsetting, summarizing, rearranging, and joining together data sets. dplyr is our go to package for fast data manipulation.

[tidyr](http://blog.rstudio.org/2014/07/22/introducing-tidyr/) - Tools for changing the layout of your data sets. Use the gather and spread functions to convert your data into the [tidy format](http://www.jstatsoft.org/v59/i10/paper), the layout R likes best.

[stringr](http://journal.r-project.org/archive/2010-2/RJournal_2010-2_Wickham.pdf) - Easy to learn tools for regular expressions and character strings.

[lubridate](http://www.r-statistics.com/2012/03/do-more-with-dates-and-times-in-r-with-lubridate-1-1-0/) - Tools that make working with dates and times easier.

To visualize data

[ggplot2](http://docs.ggplot2.org/current/) - R's famous package for making beautiful graphics. ggplot2 lets you use the [grammar of graphics](http://vita.had.co.nz/papers/layered-grammar.pdf) to build layered, customizable plots.

[ggvis](http://ggvis.rstudio.com/) - Interactive, web based graphics built with the grammar of graphics.

[rgl](http://rgl.neoscientists.org/about.shtml) - Interactive 3D visualizations with R

[htmlwidgets](http://www.htmlwidgets.org/) - A fast way to build interactive (javascript based) visualizations with R. Packages that implement htmlwidgets include:

* [leaflet](http://rstudio.github.io/leaflet/) (maps)
* [dygraphs](http://rstudio.github.io/dygraphs) (time series)
* [DT](http://rstudio.github.io/DT/) (tables)
* [diagrammeR](http://rich-iannone.github.io/DiagrammeR/) (diagrams)
* [network3D](http://christophergandrud.github.io/networkD3/) (network graphs)
* [threeJS](https://github.com/bwlewis/rthreejs) (3D scatterplots and globes).

[googleVis](https://cran.rstudio.com/web/packages/googleVis) - Let's you use Google Chart tools to visualize data in R. Google Chart tools used to be called Gapminder, the graphing software Hans Rosling made famous in hie TED talk.

To model data

[car](http://www.rdocumentation.org/packages/car) - car's [Anova](http://www.rdocumentation.org/packages/car/functions/Anova) function is popular for making type II and type III Anova tables.

[mgcv](http://www.rdocumentation.org/packages/mgcv/functions/mgcv-package) - Generalized Additive Models

[lme4](http://www.rdocumentation.org/packages/lme4/functions/lme4-package)/[nlme](http://www.rdocumentation.org/packages/nlme/functions/nlme) - Linear and Non-linear mixed effects models

[randomForest](http://www.rdocumentation.org/packages/randomForest/functions/randomForest) - Random forest methods from machine learning

[multcomp](http://www.rdocumentation.org/packages/multcomp) - Tools for multiple comparison testing

[vcd](http://www.rdocumentation.org/packages/vcd) - Visualization tools and tests for categorical data

[glmnet](http://www.rdocumentation.org/packages/glmnet/functions/glmnet) - Lasso and elastic-net regression methods with cross validation

[survival](http://www.rdocumentation.org/packages/survival) - Tools for survival analysis

[caret](https://cran.rstudio.com/web/packages/caret) - Tools for training regression and classification models

To report results

[shiny](http://shiny.rstudio.com/) - Easily make interactive, web apps with R. A perfect way to explore data and share findings with non-programmers.

[R Markdown](https://rmarkdown.rstudio.com/) - The perfect workflow for reproducible reporting. Write R code in your [markdown](http://daringfireball.net/projects/markdown/) reports. When you run render, R Markdown will replace the code with its results and then export your report as an HTML, pdf, or MS Word document, or a HTML or pdf slideshow. The result? Automated reporting. R Markdown is integrated straight into RStudio.

[xtable](http://www.rdocumentation.org/packages/xtable/functions/xtable) - The [xtable](http://www.rdocumentation.org/packages/xtable/functions/xtable) function takes an R object (like a data frame) and returns the latex or HTML code you need to paste a pretty version of the object into your documents. Copy and paste, or pair up with R Markdown.

For Spatial data

[sp](https://cran.rstudio.com/web/packages/sp), [maptools](http://www.rdocumentation.org/packages/maptools) - Tools for loading and using spatial data including shapefiles.

[maps](http://www.rdocumentation.org/packages/maps) - Easy to use map polygons for plots.

[ggmap](http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf) - Download street maps straight from Google maps and use them as a background in your ggplots.

For Time Series and Financial data

[zoo](https://cran.rstudio.com/web/packages/zoo) - Provides the most popular format for saving time series objects in R.

[xts](https://cran.rstudio.com/web/packages/xts) - Very flexible tools for manipulating time series data sets.

[quantmod](http://www.quantmod.com/) - Tools for downloading financial data, plotting common charts, and doing technical analysis.

To write high performance R code

[Rcpp](http://dirk.eddelbuettel.com/code/rcpp.html) - Write R functions that call C++ code for lightning fast speed.

[data.table](http://www.rdocumentation.org/packages/data.table) - An alternative way to organize data sets for very, very fast operations. Useful for big data.

[parallel](http://www.rdocumentation.org/packages/parallel/functions/parallel-package) - Use parallel processing in R to speed up your code or to crunch large data sets.

To work with the web

[XML](http://www.omegahat.org/RSXML/shortIntro.pdf) - Read and create XML documents with R

[jsonlite](https://cran.rstudio.com/web/packages/jsonlite) - Read and create JSON data tables with R

[httr](http://www.rdocumentation.org/packages/httr/functions/httr) - A set of useful tools for working with http connections

To write your own R packages

[devtools](https://www.rstudio.com/projects/devtools/) - An essential suite of tools for turning your code into an R package.

[testthat](http://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf) - testthat provides an easy way to write unit tests for your code projects.

[roxygen2](http://www.rdocumentation.org/packages/roxygen2) - A quick way to document your R packages. roxygen2 turns inline code comments into documentation pages and builds a package namespace.

You can also read about the entire package development process online in Hadley Wickham's [R Packages](http://r-pkgs.had.co.nz/) book