

Hands-On Data Science with R

Pulling it Together into a Package

Graham.Williams@togaware.com

1st June 2016

Visit <http://HandsOnDataScience.com/> for more Chapters.

In this module we review the `package.skeleton()` function of R R-base (?).

As we work through this chapter, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the `?` command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the `help=` option of `library()`:

```
library(help=rattle)
```

This chapter is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

Copyright © 2013-2015 Graham Williams. You can freely copy, distribute, or adapt this material, as long as the attribution is retained and derivative work is provided under the same license.



1 Functions to Package

We will create a package containing the following functions. The Functions module covered the development of these functions and more. The aim is to get started with a package rather than to produce a fully developed package. Thus we include two quite simple functions to illustrate the process.

```
varWeights <- function(formula, data)
{
  vars <- as.character(attr(terms(model.frame(formula, data)),
                           "variables"))[-1L]

  target <- vars[[1]]
  inputs <- vars[-1]
  abscor <- function(x)
    abs(suppressWarnings(cor(as.numeric(data[[x]]),
                           as.numeric(data[[target]]),
                           use="pairwise.complete.obs")))

  correlation <- sapply(inputs, abscor)

  correlation[is.na(correlation)] <- 0
  correlation[correlation == 1] <- 0

  return(correlation/sum(correlation))
}

selectVars <- function(formula, data, n)
{
  pvar <- varWeights(formula, data)
  draws = sample(names(pvar), size=n, replace=FALSE, prob=pvar)
  return(draws)
}
```

2 Package Skeleton

Having defined the above functions in an R session we can then begin creating the package to include just these two functions, to start with. We'll call the package `vw`.

```
package.skeleton(name="vw", list=c("varWeights", "selectVars"))  
## Creating directories ...  
## Creating DESCRIPTION ...  
## Creating NAMESPACE ...  
## Creating Read-and-delete-me ...  
## Saving functions and data ...  
## Making help files ...  
## Done.  
## Further steps are described in './vw/Read-and-delete-me'.
```

This creates a folder with the name `vw`. The folder contains three files. One is called `DESCRIPTION` and should be edited to provide details of the package. Another is called `NAMESPACE` and will not need to be changed for now. The third is called `Read-me-and-delete` which, as the name suggests, is worth having a read of, but then deleting.

Two folders are also created, named `R` and `man`. The `R` folder contains several `.R` files, one for each of the functions forming the package, containing the actual function definitions. The `man` folder contains documentation (manual pages), again a single file for each function, plus an additional file for documenting the package in general. Each file has the `.Rd` extension, for R documentation.

3 Check Package

After editing each of the .Rd files to describe the functionality, we can check that the package has integrity using:

```
$ R CMD check vw
```

Draft Only

4 Build Package

If all is okay, we can then build the package:

```
$ R CMD build vw
```

5 Install Package

Finally, install the package:

```
$ R CMD INSTALL vw_1.0.tar.gz
```

The package file `vw_1.0.tar.gz` is the source code and documentation, stored in a `.tar` archive format, compressed using `gzip` to provide a `.gz` file for distribution. This is a so-called source package. It can be installed on GNU/Linux systems such as Ubuntu. Extra work is required to build a binary package for MS/Windows or a package for Mac/OSX.

Once the package is installed we can load it into R:

```
library(vw)
```

Draft Only

6 Cleanup

7 Submitting to CRAN

The CRAN team are volunteers dedicated to maintaining the quality of R and the suite of packages available to R. Kurt Hornik and Uwe Ligges, in particular, work tirelessly managing the uploading of packages across the different platforms. Thus the onus needs to be on us, the package contributors, to minimise their effort by ensuring our packages have been thoroughly checked before an upload to CRAN.

Draft Only

8 Further Reading

9 References

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Williams GJ (2009). “Rattle: A Data Mining GUI for R.” *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer, New York.

This document, sourced from PackageO.Rnw bitbucket revision 148, was processed by KnitR version 1.12.3 of 2016-01-22 and took 1.2 seconds to process. It was generated by gjw on theano running Ubuntu 14.04.4 LTS with Intel(R) Core(TM) i7-3517U CPU @ 1.90GHz having 4 cores and 3.9GB of RAM. It completed the processing 2016-06-01 19:43:15.

Draft Only