Data Science with R
The Basics of R

Graham.Williams@togaware.com

16th August 2014

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

In this chapter we introduce some of the basic commands we often use in interacting with R as a Data Scientist. We don't aim to be comprehensive but rather to provide basic familiarity.

The required packages for this module include:

```
library(rattle)
library(scales)
```

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.

1 Data Types: Numeric

2 Data Types: Integer

3 Data Types: Complex

4 Data Types: Logical

5 Data Types: Character

The class of objects that we generally call strings in character.

```
class("abc")

## [1] "character"
```

We can convert other data types to class character using `as.character()`:

```
n <- 42.134
class(n)

## [1] "numeric"

as.character(n)

## [1] "42.134"
```

Concatenate strings:

```
paste("abc", "def")

## [1] "abc def"

paste("abc", "def", sep="")

## [1] "abcdef"

paste0("abc", "def")

## [1] "abcdef"
```

String formatting:

```
s <- "abc"
sprintf("The length of %s is %d.", s, length(s))

## [1] "The length of abc is 1."
```

Substrings:

```
s <- "Vulpes celeris et fluva salit super ignavum canem."
substr(s, start=8, stop=12)

## [1] "celer"
```

Substitute:

```
s <- "Vulpes celeris et fluva salit super ignavum canem."
sub("Vulpes", "The Fox", s)

## [1] "The Fox celeris et fluva salit super ignavum canem."
```

6 Data Structure: Vector

7 Data Structure: Matrix

8 Data Structure: Data Frame

8.1 Sort a Data Frame

From Stack Overflow.

```
sort.data.frame <- function(x, decreasing=FALSE, by=1, ... )
{
  f <- function(...) order(..., decreasing=decreasing)
  i <- do.call(f, x[by])
  x[i,,drop=FALSE]
}
sort(weather, by="MinTemp")

##             Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 293 2008-08-19 Canberra    -5.3    13.1      0.0         2.2      7.9
## 298 2008-08-24 Canberra    -3.7    14.4      0.0         2.6     10.4
## 314 2008-09-09 Canberra    -3.7    14.7      0.0         3.4     10.9
....

sort(weather, by="MaxTemp")

##             Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 284 2008-08-10 Canberra    -3.5     7.6      0.4         2.4      4.7
## 254 2008-07-11 Canberra     2.9     8.4      1.6         1.4      7.7
## 253 2008-07-10 Canberra     1.8     8.7      0.0         1.8      1.2
....

sort(weather, by=c("MinTemp", "MaxTemp"))

##             Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 293 2008-08-19 Canberra    -5.3    13.1      0.0         2.2      7.9
## 298 2008-08-24 Canberra    -3.7    14.4      0.0         2.6     10.4
## 314 2008-09-09 Canberra    -3.7    14.7      0.0         3.4     10.9
....

sort(weather, decreasing=TRUE, by=c("MinTemp", "MaxTemp"))

##             Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 73  2008-01-12 Canberra    20.9    35.7      0.0        13.8      6.9
## 52  2007-12-22 Canberra    19.9    22.0     11.0         4.4      5.9
## 96  2008-02-04 Canberra    18.2    22.6      1.8         8.0      0.0
....

sort(weather, by=3:4)

##             Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine
## 293 2008-08-19 Canberra    -5.3    13.1      0.0         2.2      7.9
## 298 2008-08-24 Canberra    -3.7    14.4      0.0         2.6     10.4
## 314 2008-09-09 Canberra    -3.7    14.7      0.0         3.4     10.9
....
```

9 Data Structure: List

10 Presentation: Display Large Numbers

By default, large numbers are written in scientific notation if they are too large (have too many digits).

```
22000 * 2500

## [1] 5.5e+07
```

We can increase the tolerance for large numbers using the scipen= argument of options(). Many readers find the scientific notation difficult and time consuming to read. So let's write out the number in full:

```
old.opts <- options(scipen=10)
22000 * 2500

## [1] 55000000
```

Such numbers remain hard to read though. Is that 5.5 million, 55 million, or 550 million. We have to look carefully to decide. Commas **always** assist, and the simplest way to get commas into the output we produce is to use comma() from scales (Wickham, 2014).

```
comma(22000 * 2500)

## [1] "55,000,000"
```

We can restore the original options, if we saved them as above:

```
options(old.opts)
22000 * 2500

## [1] 5.5e+07
```

11 Data Frames: Creating

Create empty data frames:

```
data.frame(a=integer(), b=numeric())
```

```
## [1] a b
## <0 rows> (or 0-length row.names)
```

```
data.frame(c=character(), d=factor(levels=c("y","n")), stringsAsFactors=FALSE)
```

```
## [1] c d
## <0 rows> (or 0-length row.names)
```

12 Data Frames: Indexing

13 Data Frames: Subsets Using subset()

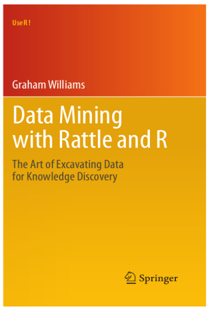
The function `subset()` was introduced by Peter Dalgaard as a convenience for subsetting data frames rather than using indexing directly. Many now argue that it is the most natural way of subsetting data frames. Brian Ripley noted that he thinks this convenience function is a mistake (R-Devel mailing list 21 October 2013) as people start to make use of them in functions and packages which can lead to issues. As the help page notes, this should only be used as an interactive convenience function, not for programming.

14 Data Frames: Saving Data

A compressed binary data file can be saved, containing one or more R objects. Here we save two R objects:

```
wfname <- "weather.RData"
save(weather, weatherAUS, file=wfname)
```

15 Further Reading and Acknowledgements


Use R!
Graham Williams
Data Mining with Rattle and R
The Art of Excavating Data for Knowledge Discovery
Springer

The Rattle Book, published by Springer, provides a comprehensive introduction to data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from http://datamining.togaware.com, including the Datamining Desktop Survival Guide.

This chapter is one of many chapters available from http://HandsOnDataScience.com. In particular follow the links on the website with a * which indicates the generally more developed chapters.

Other resources include:

- http://www.nzdl.org/Books/Books/realistic-books-svn/books/R_ByExample/
- Chi Yau's R Tutorial is a good place to start with R too.

16 References

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Wickham H (2014). *scales: Scale functions for graphics*. R package version 0.2.4, URL http://CRAN.R-project.org/package=scales.

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. URL http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

This document, sourced from BasicRO.Rnw revision 484, was processed by KnitR version 1.6 of 2014-05-24 and took 3.5 seconds to process. It was generated by gjw on nyx running Ubuntu 14.04.1 LTS with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-08-16 17:45:28.