JOINing data in R using data.table

Ronald Stalder

23-12-2014

Tutorial on how to perform the different types

of SQL-JOINs in R using data.table

Synopsis

- This tutorial is **based on the following articles**:
  - basic overview of SQL JOINs by Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)
  - the Wikipedia entry on Join (SQL) (http://en.wikipedia.org/wiki/Join_(SQL))
  - more elaborate: The joy of joining data.tables (http://www.magesblog.com/2014/06/the-joy-of-joining-datatables.html)
- For **joining data.tables, the basics are**:
  - the `ON` or `USING` clause is defined by setting the keys on the tables with `setkey()`
  - without anything else, `TABLE_X[TABLE_Y]` returns a **right outer join**; setting `nomatch=0` it returns a **inner join**
- The **source of this tutorial**, with the example datasets, is available here on GitHub (https://github.com/ronasta/JOINing-Data-with-R-data.table)

Summary

JOIN type	DT syntax	data.table::merge() syntax
INNER	X[Y, nomatch=0]	merge(X, Y, all=FALSE)
LEFT OUTER	Y[X]	merge(X, Y, all.x=TRUE)
RIGHT OUTER	X[Y]	merge(X, Y, all.y=TRUE)
FULL OUTER	-	merge(X, Y, all=TRUE)
FULL OUTER WHERE NULL (NOT INNER)	-	merge(X, Y, all=TRUE), subset NA
CROSS (cartesian)	-	- (see below)

Example Data

The example data consists of the following two tables:

table Employees

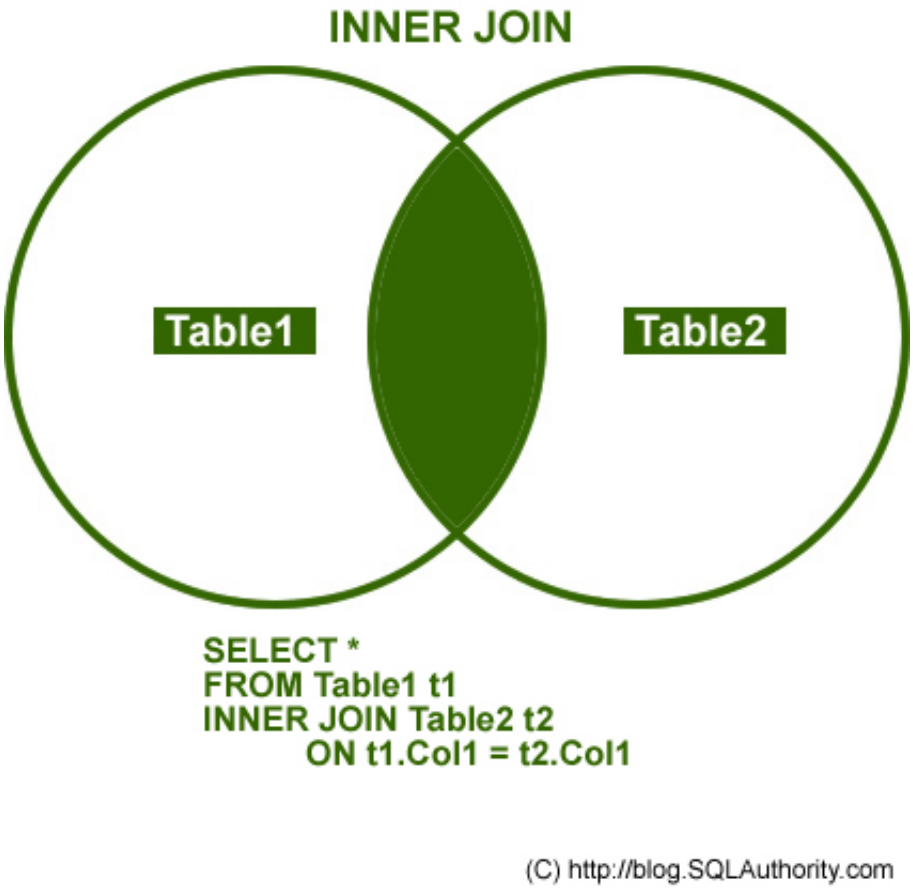
Employee	EmployeeName	Department	Salary
1	Alice	11	800
2	Bob	11	600
3	Carla	12	900
4	Daniel	12	1000
5	Evelyn	13	800
6	Ferdinand	21	700

table Departments

Department	DepartmentName	Manager
11	Production	1
12	Sales	4
13	Marketing	5
14	Research	NA

Inner Join

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



the *INNER JOIN* returns the rows with a match in both tables

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# perform the join, eliminating not matched rows from Right
Result <- Employees[Departments, nomatch=0]
```

Inner Join

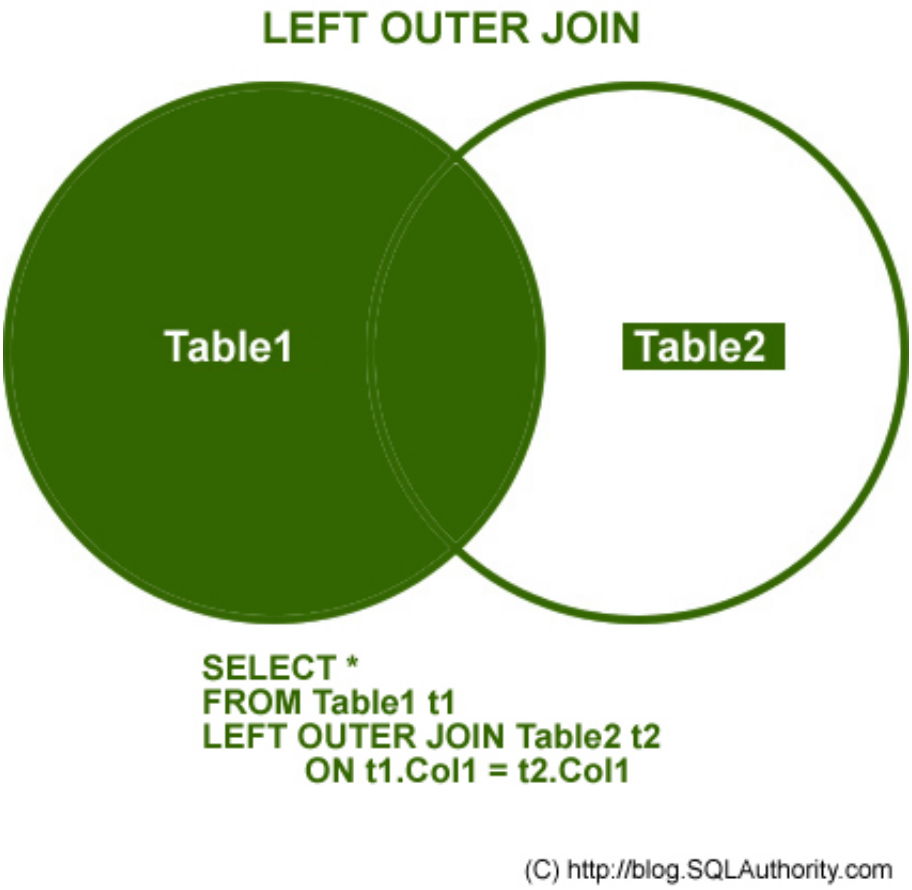
Employee	EmployeeName	Department	Salary	DepartmentName	Manager
1	Alice	11	800	Production	1
2	Bob	11	600	Production	1
3	Carla	12	900	Sales	4
4	Daniel	12	1000	Sales	4
5	Evelyn	13	800	Marketing	5

note:

> Employee #6, Ferdinand, has not been returned, as he's in a yet to be created department 21.
> Neither has department #14, Research, as there are no employees.

Left Outer Join

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



the *LEFT OUTER JOIN* returns all the rows from the left table, filling in matched columns (or NA) from the right table

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# perform the join using the merge function
Result <- merge(Employees,Departments, all.x=TRUE)
```

Left Outer Join

Department	Employee	EmployeeName	Salary	DepartmentName	Manager
11	1	Alice	800	Production	1
11	2	Bob	600	Production	1
12	3	Carla	900	Sales	4
12	4	Daniel	1000	Sales	4
13	5	Evelyn	800	Marketing	5
21	6	Ferdinand	700	NA	NA

note:

> Employee #6, Ferdinand, has been returned with department name as NA.
> Department #14, Research, has not been returned.
> If the column order Left → Right has to be preserved, we need to elaborate:

```
# get the columns of the tables:
leftCols <- colnames(Employees)
rightCols <- colnames(Departments)
# remove the match key of the Right table
rightCols <- setdiff(rightCols,key(Departments))
# set the column order
setcolorder(Result,c(leftCols,rightCols))
```

Left Outer Join, preserved column order

Employee	EmployeeName	Department	Salary	DepartmentName	Manager
1	Alice	11	800	Production	1
2	Bob	11	600	Production	1
3	Carla	12	900	Sales	4
4	Daniel	12	1000	Sales	4
5	Evelyn	13	800	Marketing	5
6	Ferdinand	21	700	NA	NA

A typical use case is to match in labels, e.g. in our Employees table substitute the department number by its name:

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# defining the Result columns, substitute Department by DepartmentName
leftCols <- colnames(Employees)
leftCols <- sub("Department","DepartmentName",leftCols)

# perform the join, inverting the tables, return defined columns
Result <- Departments[Employees][, leftCols, with=FALSE]

# -- or --
# Result <- merge(Employees, Departments, all.x=TRUE)
# Result <- Result[, setCols, with=FALSE]
```

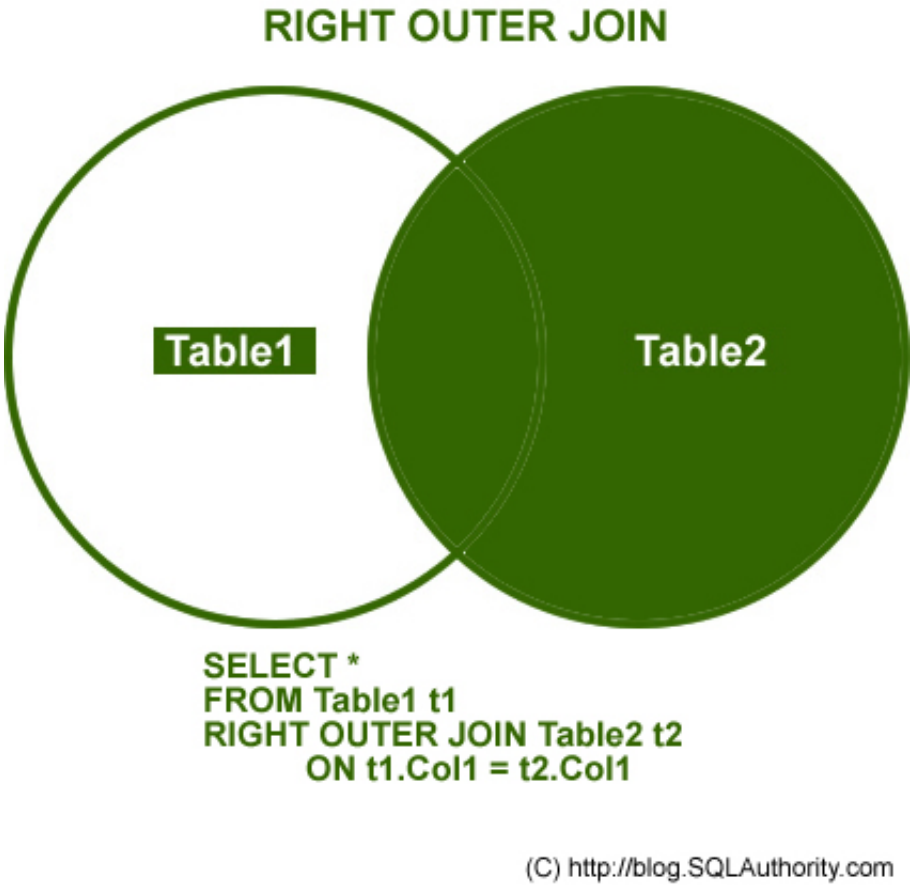
Left Outer Join - matching in DepartmentName

Employee	EmployeeName	DepartmentName	Salary
1	Alice	Production	800
2	Bob	Production	600
3	Carla	Sales	900
4	Daniel	Sales	1000
5	Evelyn	Marketing	800

6	Ferdinand	NA	700

Right Outer Join

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



the *RIGHT OUTER JOIN* returns all the rows from the right table, filling in matched columns (or NA) from the left table

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# perform the join - this is the basic join for data.table
Result <- Employees[Departments]
# this corresponds to
# Result <- merge(Employees,Departments, all.y=TRUE)
```

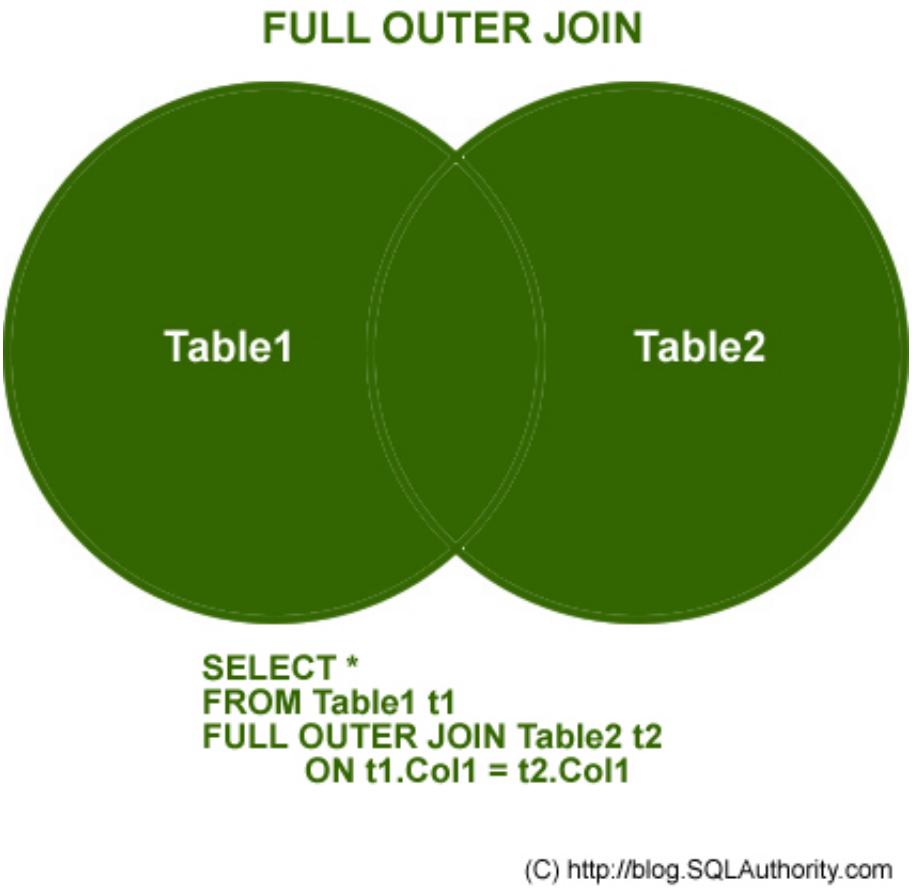
Right Outer Join

Employee	EmployeeName	Department	Salary	DepartmentName	Manager
1	Alice	11	800	Production	1
2	Bob	11	600	Production	1
3	Carla	12	900	Sales	4

4	Daniel	12	1000	Sales	4
5	Evelyn	13	800	Marketing	5
NA	NA	14	NA	Research	NA

Full Outer Join

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



the *FULL OUTER JOIN* returns all the rows from both tables, filling in matched columns (or NA)

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# perform the join
Result <- merge(Employees,Departments, all=TRUE)
```

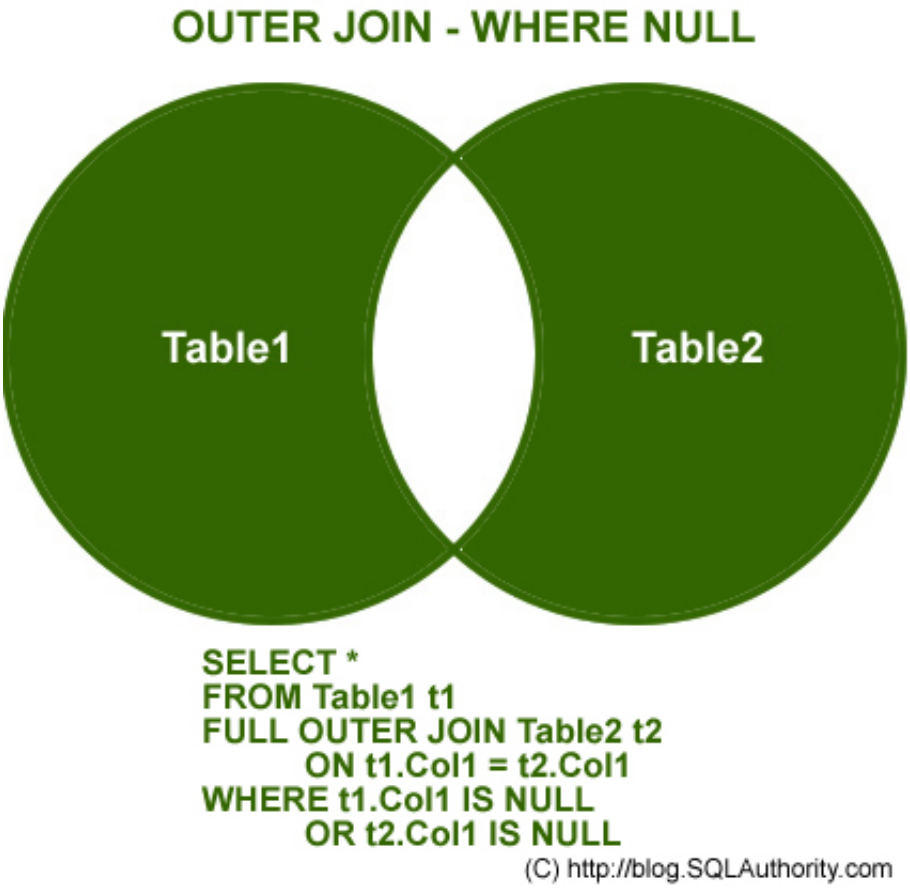
Full Outer Join

Department	Employee	EmployeeName	Salary	DepartmentName	Manager
11	1	Alice	800	Production	1
11	2	Bob	600	Production	1
12	3	Carla	900	Sales	4
12	4	Daniel	1000	Sales	4

13	5	Evelyn	800	Marketing	5
14	NA	NA	NA	Research	NA
21	6	Ferdinand	700	NA	NA

Full Outer Join Where NULL - a.k.a “NOT INNER join”

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



the *NOT INNER JOIN* returns all the rows from both tables, where no match was obtained

```
# set the ON clause as keys of the tables:
setkey(Employees,Department)
setkey(Departments,Department)

# perform the join, retain only NA from matched cols on both side
Result <- merge(Employees,Departments, all=TRUE)
Result <- Result[is.na(EmployeeName) | is.na(DepartmentName)]
```

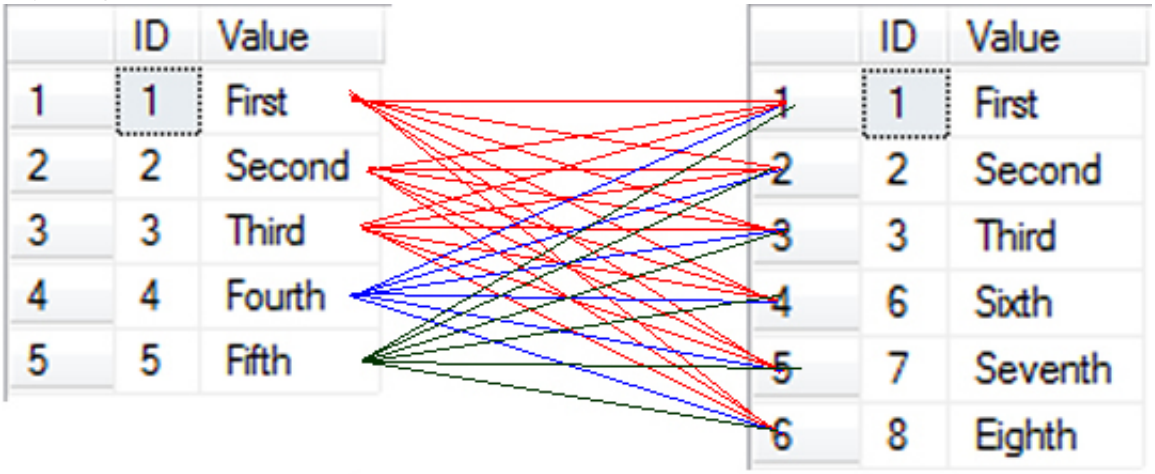
Full Outer Join Where Null

Department	Employee	EmployeeName	Salary	DepartmentName	Manager
14	NA	NA	NA	Research	NA

21	6	Ferdinand	700	NA	NA

Cross Join

Figure linked from Pinal Dave (http://blog.sqlauthority.com/2009/04/13/sql-server-introduction-to-joins-basic-of-joins/)



	ID	Value	ID	Value
1	1	First	1	First
2	1	First	2	Second
3	1	First	3	Third
4	1	First	6	Sixth
5	1	First	7	Seventh
6	1	First	8	Eighth
7	2	Second	1	First
8	2	Second	2	Second
9	2	Second	3	Third
10	2	Second	6	Sixth
11	2	Second	7	Seventh
12	2	Second	8	Eighth

the *CROSS JOIN* returns all the rows from one table combined with all the rows from the other

As the documentation `?data.table` states, this ***"is (deliberately) difficult to achieve in data.table"***. So, I'll leave this as an excersise to the reader :-). In the context of the examples here, I can't see the usefulness of a cross join, anyway.

System and R-Packages Information

```
if (Sys.which("uname") != "") system("uname -srpi", intern=TRUE)
if (Sys.which("lsb_release") != "")
    print(paste("Ubuntu:",system("lsb_release -rc", intern=TRUE)[1]))
#print(paste("Rstudio version:", rstudio::versionInfo()$version))  # does not work in Rmd
```

```
## [1] "Linux 3.13.0-43-generic x86_64 x86_64"
## [1] "Ubuntu: Release:\t14.04"
```

```
sessionInfo()
```

```
## R version 3.1.2 (2014-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=pt_BR.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=pt_BR.UTF-8        LC_COLLATE=pt_BR.UTF-8
##  [5] LC_MONETARY=pt_BR.UTF-8    LC_MESSAGES=pt_BR.UTF-8
##  [7] LC_PAPER=pt_BR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=pt_BR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] data.table_1.9.4
##
## loaded via a namespace (and not attached):
##  [1] chron_2.3-45     digest_0.6.4     evaluate_0.5.5   formatR_1.0
##  [5] highr_0.4        htmltools_0.2.6  knitr_1.8        plyr_1.8.1
##  [9] Rcpp_0.11.3      reshape2_1.4.1   rmarkdown_0.3.10 stringr_0.6.2
## [13] tools_3.1.2      yaml_2.1.13
```

The report was produced using `RStudio/knittr`
on 2014-12-23 at 14:07:00 (BRT, GMT-0300)