

# irutils: An R Package for Institutional Research

Jason M. Bryer  
jbryer@expcelsior.edu  
Excelsior College

February 14, 2012

## Abstract

This document specifies the functions available in the `irutils` package. This package is designed to accompany the *Introduction to R and L<sup>A</sup>T<sub>E</sub>X* document. The package provides many functions useful for typical reporting requirements of Institutional Research offices.

Keywords: *institutional research, database, higher education, ipeds*

## 1 Introduction

The `irutils` package is hosted at <http://github.com/jbryer/irutils>. The `devtools` (Wickham, 2011) package provides functions to assist in R package development including a function to install directly from github. The `install_github` will install the latest development version of the `irutils` package.

```
> library(devtools)
> install_github('devtools', 'jbryer')
```

The currently available functions are:

```
> library(irutils)
> ls('package:irutils')

[1] "adjustedTableLaTeX"      "cacheQuery"
[3] "cohortRetention"         "cohortSummary"
[5] "cohortSummaryLaTeX"      "createLocalRepos"
[7] "execQuery"               "fySummary"
[9] "fySummaryLaTeX"          "geocode.google"
[11] "geocode.ips"              "geocode.yahoo"
[13] "geolite.delete"           "geolite.load"
[15] "geolite.update"           "geolite.version"
[17] "getAge"                   "getAgeGroups"
[19] "getAgeGroupsIPEDS"        "getParameters"
```

[21]	"getQueries"	"getQuery"
[23]	"getQueryDesc"	"getSQLRepos"
[25]	"local.available.packages"	"local.install.packages"
[27]	"plotBarchartTable"	"plotCohortRetention"
[29]	"plotHeatmapTable"	"plotPersistence"
[31]	"plotRetention"	"plotRetentionOverall"
[33]	"retention"	"retentionSummary"
[35]	"retentionSummaryLaTeX"	"saveIPEDStoDB"
[37]	"setSQLRepos"	"updateLocalRepos"

## 2 Database Access

For many Institutional Research offices the institutions student information system (SIS) is the most common source of data. Since virtually all SIS systems are backed by a database, extracting data requires extracting data using queries. Typically the language used to extract data is called structured query language (SQL) regardless if the database is provided by Oracle, Microsoft, or an open source options such as MySQL and PostgreSQL. There are a number of functions in this package that will facilitate extracting data from these databases directly into R.

The database access functions provide an interface to a directory of SQL scripts. SQL scripts are simply a plain text file containing the query. The directory containing these files can be determined or set using the `getSQLRepos` and `setSQLRepos` functions, respectively.

```
> getSQLRepos()
```

```
[1] "0:/Resources/R/library/irutils/data"
```

By convention, all SQL files must use a `.sql` file extension. The `getQueries` function will return a list of all the queries available in the current repository.

```
> getQueries()
```

```
[1] "TestQuery"
```

The `getQueryDesc` and `getParameters` functions will provide some details about the query in question. In particular, the latter will return the parameters that are required for the query to execute.

```
> getQueryDesc('TestQuery')
```

```
[1] ""
```

```
> getParameters('TestQuery')
```

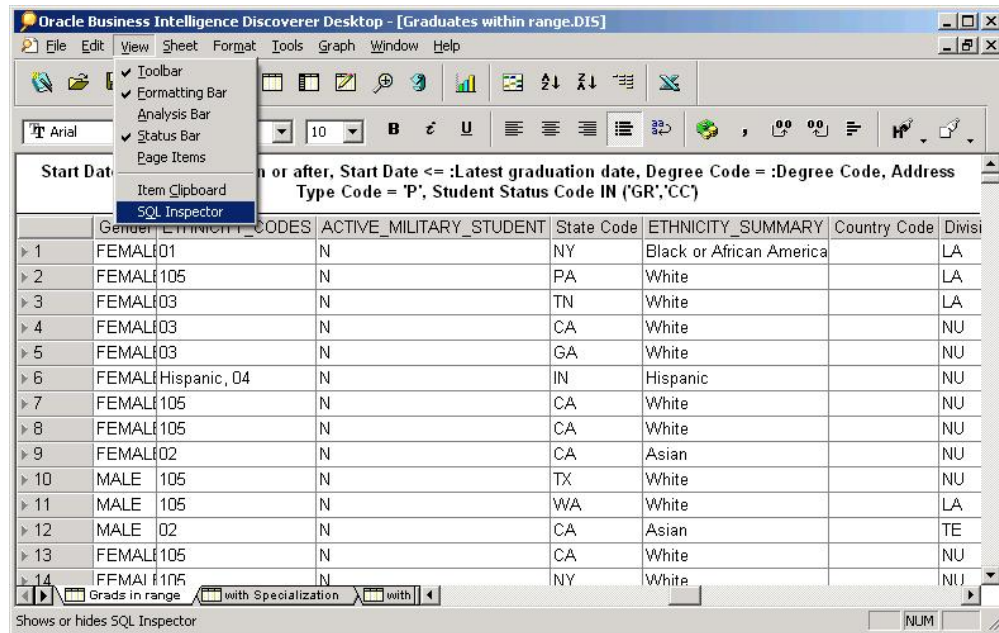
```
[1] "ContactID"
```

There are two functions available for executing the query. The `execQuery` will execute the query and return a data frame. The `cacheQuery` however, will first look in the specified directory (by default the `dir` parameter is set to `getwd()`) for a CSV file that matches the currently request query. That is, the file name (which is returned when this function is executed) is built using

a combination of the query name and parameters to uniquely identify it. This is useful when using Sweave and L<sup>A</sup>T<sub>E</sub>X for document preparation where the function may be executed multiple times but the data does not change. It is considerably faster to read data from a flat file than it is to query the database each time.

## 2.1 Creating Your Own Query

To create your own query, simply place the SQL statement in its own text file ending with .sql. Comments can be placed in the file using the # symbol. Placing informative comments at the beginning of the file will be useful since the `getQueryDesc` function will return these comments to the user. Parameters can be placed anywhere in the file and must be enclosed within colons (i.e. the : character). Additionally, parameter names must begin with a letter, contain only letters and numbers, and cannot have any spaces.



The screenshot shows the Oracle Business Intelligence Discoverer Desktop interface. A menu is open over the 'SQL Inspector' option. The main window displays a table with the following data:

	Gender	ETHNICITY_CODES	ACTIVE_MILITARY_STUDENT	State Code	ETHNICITY_SUMMARY	Country Code	Divisi
1	FEMALE	01	N	NY	Black or African America		LA
2	FEMALE	105	N	PA	White		LA
3	FEMALE	03	N	TN	White		LA
4	FEMALE	03	N	CA	White		NU
5	FEMALE	03	N	GA	White		NU
6	FEMALE	Hispanic, 04	N	IN	Hispanic		NU
7	FEMALE	105	N	CA	White		NU
8	FEMALE	105	N	CA	White		NU
9	FEMALE	02	N	CA	Asian		NU
10	MALE	105	N	TX	White		NU
11	MALE	105	N	WA	White		LA
12	MALE	02	N	CA	Asian		TE
13	FEMALE	105	N	CA	White		NU
14	FEMALE	105	N	NY	White		NU

Figure 1: Beginning 72-Month Completion Rates by Military Status

## 3 Visualizing Likert Response Items

```
> data(pisa)
> items28 = pisa[,substr(names(pisa), 1,5) == 'ST24Q']
> names(items28) = c("I read only if I have to.",
+   "Reading is one of my favorite hobbies.",
```

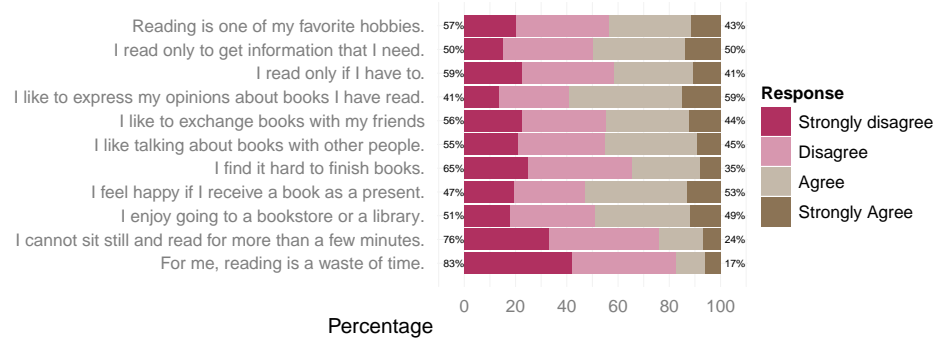


Figure 2: Barchart Table Graphic

```

+   "I like talking about books with other people.",
+   "I find it hard to finish books.",
+   "I feel happy if I receive a book as a present.",
+   "For me, reading is a waste of time.",
+   "I enjoy going to a bookstore or a library.",
+   "I read only to get information that I need.",
+   "I cannot sit still and read for more than a few minutes.",
+   "I like to express my opinions about books I have read.",
+   "I like to exchange books with my friends")
> for(i in 1:ncol(items28)) {
+   items28[,i] = factor(items28[,i], levels=1:4,
+   labels=c('Strongly disagree', 'Disagree', 'Agree', 'Strongly Agree'),
+   ordered=TRUE)
+ }

```

## 4 Retention and Completion Rates

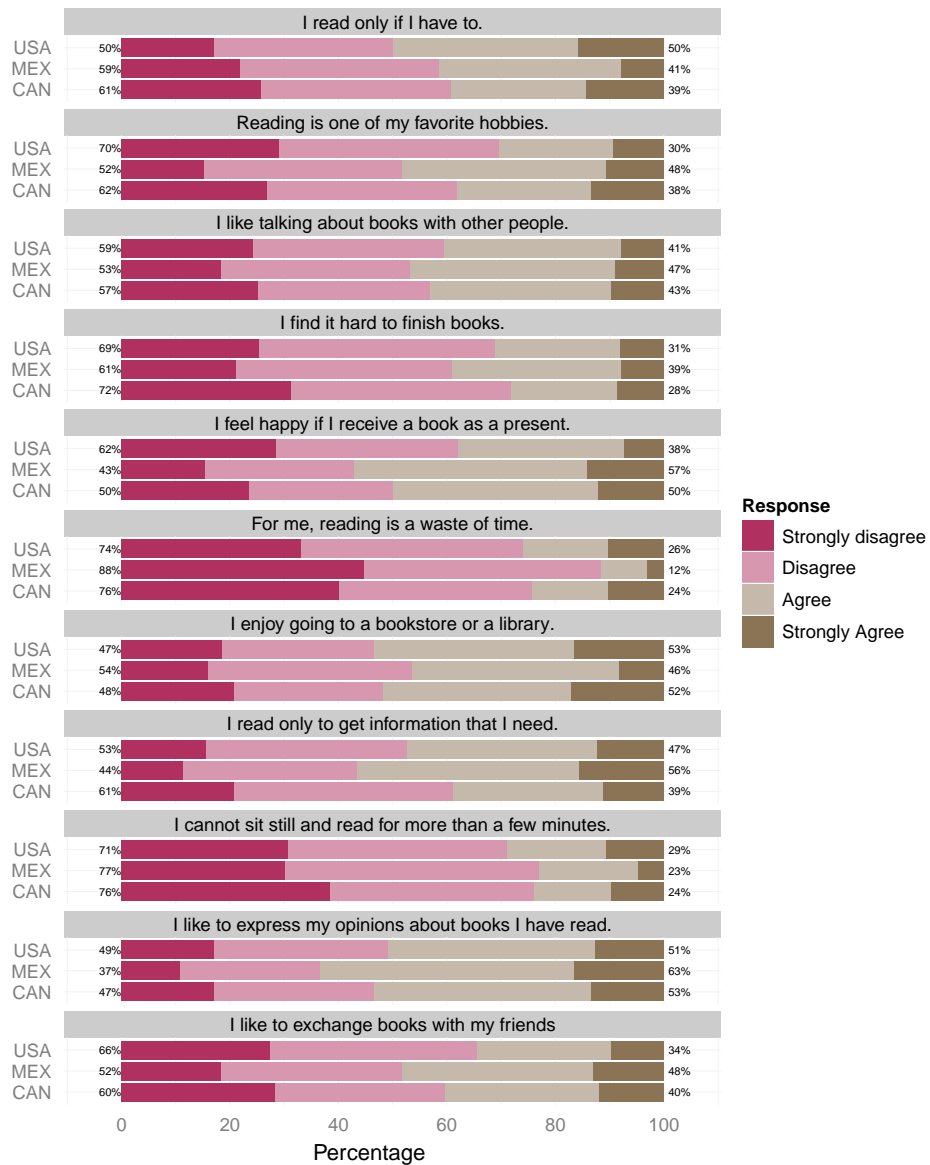


Figure 3: Barchart Table Graphic Grouped by Country

## References

Wickham, H. (2011). devtools: Tools to make developing r code easier [Computer software manual]. Available from <http://CRAN.R-project.org/package=devtools> (R package version 0.4)