Collecting R Package Usage Information

by Zhiyong Zhang

Abstract Usage information about a package can help its developers, and ultimately its users, in many ways. However, such information is rarely available. An R package **rstats** is developed in the hope to stimulate more discussion on how to better collect and utilize package usage information. The package **rstats** makes easily possible rating a package and viewing package usage information within R.

As of April 2013, there are almost 4,500 R packages available on CRAN and the number is growing. Contribution of R packages from all around the world has kept R more sophisticated than the rest of statistical software and has also largely advanced statistical data analysis. Usage information on a package can help the developers understand how a package is used, track potential bugs, and improve the package in a future release, to say the least. In some cases, such information can also be used to make high-stake decision on the developers, e.g., by their supervisors.

As a developer of several R packages myself, however, I found that usage information of R packages was rare. There are certainly good reasons for the lack of usage information. CRAN has many mirrors that appear and disappear over time. Therefore, it is generally difficult to track the usage information. For example, although the site `http://neolab.stat.ucla.edu/cranstats/` provides information about package usage, the information is limited to the packages downloaded from the UCLA CRAN mirror. Furthermore, it only provides basic information on the downloads of a given package. The site `http://crantastic.org/` has been recently started by Dr. Hadley Wickham to collect more information on R usage. However, the users of R seemed to lack the enthusiasm to use the site, I think, largely because of the lack of a direct connection between R and the website.

In this paper, I propose a simple method that can be used to more actively collect information on the usage of R packages. This method, utilizing a simple R package **rstats** (with only two R functions) and a web server (with only two PHP script files), allows users to provide usage information of a package within R. The collected information can be viewed within R directly. The method can be easily integrated to CRAN or crantastic. The package can also be used in other scenarios as we will point out.

In the following, I will first demonstrate how to use the R package **rstats**. Then I will explain how the package works. Finally, I discuss how to improve the package in the future. Because of the simplicity of the package, I will list the entire source code in the appendix so that interesting users can better utilize the package.

Use of rstats

The package **rstats** is currently available on github and can be installed using `install_github('rstats','johnnyzhz')`. After installing the package, a user can start to provide download information, rate, and comment on any R package currently installed by the user using the following function.

```
rate(package="base", download=NULL, like=NULL, rating=NULL, comment=NULL,
                  meta=TRUE, lib.loc = NULL)
```

In the `rate` function, the argument `package` asks for the name of the package to rate. By default, the R base is used. A user can only provide usage information on a package that he or she actually installed, hopefully used. The function will prompt an error if the package provided is not installed. Setting the argument `download` to anything other than `NULL` such as `download=TRUE` or `download=1` will indicate a user has downloaded a package. If a user wants to vote to like a package, he/she can provide a non-null value to the argument `like`. A user can also rate a package with scores 1, 2, 3, 4, 5 indicating Bad, OK, Good, Very Good, Excellent ratings through the argument `rating`. Finally, a user can also comment on an package through the argument `comment`. The argument `meta` determines whether to provide meta information of a package. If it is set to `TRUE`, information regarding the version number, how the package is built, and the maintainer of the package will be collected.

For example, the following code suggests that a user downloaded the package **rsem** and rated the package as "Excellent". The user also provided a comment on the usefulness of the package. Currently, a comment is limited to 1,000 characters.

```
rate(package="rsem", download=1, rating=5,
     comment="I found the package was useful for analyzing my non-normal missing data.")
```

All information provided by a user is saved on a web server at http://rstats.psychstat.org. The usage information of a package can be accessed within R using the function below,

```
view(package = "base",comment = FALSE,ncomment = 1:5,lib.loc = NULL)
```

Note that a user does not have to install a package to view its usage information. By default, the function only provides the information on download, like, and rating. If a user wants to view the comments of a package, he/she needs to set `comment = TRUE`. By default, only the 5 latest comments will be retrieved. However, one can change the argument `ncomment` to view more comments or select a specific subset of comments.

The following example displays the usage information of the R package **rsem**. Basically, the output says that the package has been downloaded 6 times. This is not the actually download time but the number of times that users chose to report their downloads. One user voted liking the package and the overall rating for this package is 4.83 (between Very Good and Excellent). There are also four pieces of comments on the package. Note that the example is made by the author, not really from end users, to illustrate how to use the package **rstats**.

```
> view('rsem', comment=T)
Download: 6
Like: 1
Rating: 4.83

There are  4  comments in total.
1  I found the package was useful for analyzing my non-normal missing data.
2  Does the package support categorical data analysis?
3  This package can be used to conduct robust SEM analysis.
4  I like the recent added support of lavaan because I don't have EQS.
```

How does rstats work

The package **rstats** utilizes the R package **RCurl** (Lang, 2013) to communicate between R and a web server (http://rstats.psychstat.org). The function `rate` calls the PHP script file `rate.php` on the web server to save usage information of a package provided by a user. The information is saved into a MySQL database on the server and the data structure of the MySQL database in provided in the appendix B. The whole process takes place within R without actually opening a web browser. However, because the information needs to be uploaded to a web server, Internet connection is necessary. To retrieve information regarding a package, the R function `view` calls the script file `view.php` on the web server. The retrieved information is then organized in the way as displayed in the previous example. The source code for both the R functions and PHP files is listed in the appendices.

Future development

Every time when the function `rate` is called, a piece of information will be saved to the database. Therefore, a user can rate the same package multiple times. Actually, a simple R loop will do this very efficiently and can easily overload the web server for saving usage information. This can also render the usage information meaningless. A simple way to improve the quality of collected data is to record the IP address and the time of response. Then on the server, one can restrict a user can provide feedback only once within certain period of time for a given IP address. Certain authentication process can also be deployed in the future so that only verified users can provide feedback information.

Many times, it will be a good idea to know which version of a package is used on what kind of operating systems. Such information is saved if a user chooses to provide it. However, currently, I have not utilized the information in reporting the usage statistics.

Ideally, R base can incorporate certain method in evaluating the usage of a package. For example, if the `rate` function can be incorporated into the function `install.packages`, the download information can be very accurate and one can also get more information such as which repository a package is downloaded. If the `rate` function is incorporated into the function `library`, one can get the information each time a package is used. Certainly, this may raise concerns on privacy.

Currently, the package **rstats** is simply used to collect information. However, this can be developed into an interactive platform between the developers and the users. For example, when a user provides a comment, the comment can be automatically delivered to the developers. The developers can also answer the comment directly within R. The communication can be archived to benefit the other users.

The developers of R packages can also incorporate **rstats** in their own packages. For example, it is often useful if the developers understand the possible error information of a package. In addition to display the error information to end users, one can also let the users choose to send the error information to the developers directly as a comment.

Closing remarks

Admittedly, the package **rstats** and the described method for collecting usage information are sill in its early developmental stage. However, it clearly demonstrates the possibility to better collect and utilize R package usage information. I hope this will stimulate more discussion among R developers and users to develop better methods in the future.

Appendix A. R code

```
rate<-function(package="base", download=NULL, like=NULL, rating=NULL, comment=NULL, meta=TRUE, lib.
    loc = NULL){
        dir <- system.file(package = package, lib.loc = lib.loc)
    if (dir == "") {
                gettextf("You have not installed the package %s. You can only rate a package you
                    have installed. Thanks. ", sQuote(package))
        }else{
                if (meta){
                        meta <- packageDescription(pkg = package)
                        meta <- paste(meta$Version, ";", meta$Built, ";", meta$Maintainer, ";",
                            meta$Repository)
                }else{
                        meta <- "Not provided"
                }
                if (!is.null(download)) download <- '1'
                if (!is.null(like)) like <- '1'
                if (!is.null(rating)){
                        if (!(rating %in% 1:5)) stop('The rating has to be 1 from 5.')
                        rating <- as.character(rating)
                }

                library('RCurl')
                postForm("http://rstats.psychstat.org/rate.php", name=package, download=download,
                    like=like, rating=rating, comment=comment, meta=meta)
                cat('Thanks for your feedback!\n')
        }
}

view<-function(package="base",comment=FALSE, ncomment=1:5, lib.loc = NULL){
        library('RCurl')
        rating<-getURL(paste('http://rstats.psychstat.org/view.php?name=',  package,  sep=''))
        rate<-strsplit(rating, "\n")[[1]]
        nrate<-length(rate)
        if (nrate>2){
                for (i in 1:3) cat(rate[i], "\n")
                cat("\n")
                if (comment){
                        if (nrate==3) stop("No comment available yet")
                        totalcomment<-nrate-3
                        cat("There are ", totalcomment, " comments in total.\n")
                        if (max(ncomment)>totalcomment) ncomment<-1:totalcomment
                                for (i in (ncomment+3)) cat(rate[i], "\n")
                        }
                }
}
```

Appendix B. PHP script files

On the webserver, three PHP script files are used. The first one is `config.php` that specifies the information on the MySQL database. The second one is `rate.php` that accepts rating from R. The third one is `view.php` that generates usage information for R.

config.php file

The content for this file is given below. Note that "XXXX" should be changed to match the server username and password for MySQL.

```
<?php
define ("DB_HOST", "localhost");
define ("DB_USER", "XXXX");
define ("DB_PASS","XXXX");
define ("DB_NAME","rstats");

$connection = mysql_connect(DB_HOST, DB_USER, DB_PASS) or die("Couldn't make connection.");
$db = mysql_select_db(DB_NAME, $connection) or die("Couldn't select database");
?>
```

The structure of the database `rstats` is given below.

```
CREATE TABLE IF NOT EXISTS `comment` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `rating` varchar(1) NOT NULL,
  `comment` text NOT NULL,
  `name` varchar(254) NOT NULL,
  PRIMARY KEY (`id`)
) AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `info` (
  `id` int(10) NOT NULL AUTO_INCREMENT,
  `name` varchar(254) NOT NULL,
  `download` int(1) NOT NULL,
  `like` int(1) NOT NULL,
  `rating` int(1) NOT NULL,
  `comment` varchar(1000) NOT NULL,
  `meta` varchar(254) NOT NULL,
  `ip` varchar(50) NOT NULL,
  `os` varchar(200) NOT NULL,
  `time` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `package` (
  `id` int(5) NOT NULL AUTO_INCREMENT,
  `name` varchar(254) NOT NULL,
  PRIMARY KEY (`id`)
) AUTO_INCREMENT=1 ;

CREATE TABLE IF NOT EXISTS `rating` (
  `name` varchar(254) NOT NULL,
  `download` int(10) NOT NULL DEFAULT '0',
  `like` int(10) NOT NULL DEFAULT '0',
  `rating` varchar(4) NOT NULL,
  `norate` int(10) NOT NULL,
  UNIQUE KEY `name` (`name`)
);
```

rate.php file

```
<?php
include('config.php');
// Submitted information from R
$name = $_POST['name'];
$rating = $_POST['rating'];
$comment = $_POST['comment'])
$download = $_POST['download'];
$like = $_POST['like'];
$meta = $_POST['meta'])
// ip information
$ip = $_SERVER["REMOTE_ADDR"];
$os = $_SERVER["HTTP_USER_AGENT"];
$time = time();

// save data into database
if ($name==''){
        echo "No R package is identified!";
```

```
}else{
        // first check whether the package is already in the database, if not, add it
        $query = "select `id` from package where `name`='$name'";
        $results = mysql_query($query) or die("Cannot connect the table package.");
        $nrow = mysql_num_rows($results);
        if ($nrow==0){
                // add the package name into the data base
                $query = "INSERT into `package` (`name`) VALUES ('$name')";
                $results = mysql_query($query) or die("Cannot insert data into package table.");
                $query = "INSERT into `rating` (`name`) VALUES ('$name')";
                $results = mysql_query($query) or die("Cannot insert data into rating table.");
        }

        // update usage information about a package
        $query = "select `download`,`like`,`rating`,`norate` from rating where `name`='$name'";
        $results = mysql_query($query) or die("Data error");
        $stats = mysql_fetch_array($results);

        if ($download==1) $download = $stats['download']+1;
        if ($like==1) $like = $stats['like']+1;
        if ($rating!=0){
                $norate = $stats['norate'];
                $rate = $stats['rating']*$norate+$rating;
                $norate = $norate + 1;
                $rate = $rate/$norate;
        }

        $query = "UPDATE rating SET `download` = '$download', `like` = '$like', `rating` = '$rate',
            `norate`='$norate' WHERE `name`='$name'";
        $results = mysql_query($query) or die("Cannot insert data into database.");
        $query = "INSERT into `info` (`name`,`download`, `like`, `rating`, `comment`, `ip`, `os`, `
            time`, `meta`) VALUES ('$name','$download','$like','$rating','$comment','$ip','$os','
            $time', '$meta')";
        $results = mysql_query($query) or die("Cannot insert data into rating table.");
        if ($comment!=""){
                $comment = urldecode($comment);
                $query = "INSERT into `comment` (`comment`,`name`) VALUES ('$comment','$name')";
                $results = mysql_query($query) or die("Cannot insert data into comment table.");
        }
        echo "Thanks for your feedback!";
}
?>
```

view.php file

```
<?php
include('config.php');
$name = $_GET['name'];
$query="SELECT * FROM rating where `name`='$name'";
$results=mysql_query($query) or die("Cannot select the table share");
$row = mysql_fetch_array( $results );

if ($row>0){
        echo "Download: ".$row['download']."\nLike: ".$row['like']."\nRating: ".$row['rating']."\n";
        $query="SELECT * FROM comment where `name`='$name'";
        $results=mysql_query($query) or die("Cannot select the table share");
        $nrow = mysql_num_rows($results);
        if ($nrow==0){
                echo "No comments available!\n";
        }else{
                $id = 1;
                while($row = mysql_fetch_array( $results )) {
                        echo $id."  ".$row['comment']."\n";  $id++;
                }
        }
}else{
        echo "No usage information on this package yet! You can be the first one to rate it!\n";
}
?>
```

Bibliography

D. T. Lang. *RCurl: General network (HTTP/FTP/...) client interface for R*, 2013. URL http://CRAN.R-project.org/package=RCurl. R package version 1.95-4.1. [p2]

Zhiyong Zhang
Department of Psychology
University of Notre Dame
USA zzhang4@nd.edu