Searching help pages of R packages

by Spencer Graves, Sundar Dorai-Raj, and Romain Francois

The sos package provides a means to quickly and flexibly search the help pages of contributed packages, finding functions and datasets in seconds or minutes that could not be found in hours or days by any other means we know.

The main capability of this package is the `findFn` function, which queries only the "function" help pages in Jonathan Baron's RSiteSearch data base and returns the results in a `data.frame` of class `findFn`. The corresponding `print` method displays the results as a table in a web browser with links to the individual help pages, sorted by package displaying the one with the most matches first. Other sos functions provide a summary with one line for each package, support the union and intersection of `findFn` objects, and write the results to an Excel file with three sheets: (1) `PackageSum2`, which provides an enhanced summary of the packages with matches, (2) the `findFn` table itself, and (3) the `call` used to produce it.

Other R functions can then be used to quickly find what you want among possibly hundreds of matches.

Two examples are considered below: First we find a dataset containing a variable `Petal.Length`, used without indicating the source by

(**?**, pp. 282-283). Second, we study R capabilities for splines, including looking for a function named `spline`.

Petal.Length

Chambers (2009, p. 282-283) uses a variable `Petal.Length` from a famous Fisher data set but without naming the dataset nor indicating where it can be found nor even if it exists in R. The sample code he provides does not work by itself. To reproduce his Figure 7.2, we must first obtain a copy of this famous data set in a format compatible with Chambers' code.

How to add Bibliography ? Chambers, John (2009) Software for Data Analysis (Springer)

Some users might try the following:

```
> help.search('Petal.Length')
No help files found ...
```

When this failed, many users might then try `RSiteSearch('Petal.Length')`. This produced 80 hits. `RSiteSearch('Petal.Length', 'function')` will identify only the help pages on this list, but we can get something similar and more useful as follows:

```
> library(sos)
> PL <- findFn('Petal.Length')
```

`PL` is a `data.frame` of class `findFn` identifying all the help pages in Jonathan Baron's data base matching the search term.

The `summary` method for such an object returns the number of matches with a table giving for each `Package` the `Count` (number of matches), `MaxScore` (max of the `Score`), `TotalScore` (sum of `Score`), and `Date`, sorted like a Pareto chart to place the `Package` with the most help pages first:

```
> summary(PL)

Total number of hits: 23
Number of links downloaded: 23

Packages with at least 1 hit
using search pattern 'Petal.Length':
          Count MaxScore TotalScore
yaImpute      8        1          8
<...>
datasets      1        2          2
<...>
```

REDO to add the Date

One of the listed packages is `datasets`. Since it's part of the default R distribution, we decide to look there first. We can select that row of PL just like we would select a row from any other data.frame:

```
> PL[PL$Package=='datasets', 'Function']
[1] iris
```

The `print` method for an object of class `findFn` opens the result in a browser with the last column being linked to the associated help page.

Problem solved in less than a minute! Any other method known to the present authors would have taken substantially more time.

spline

Three years ago, I decided I wanted to learn more about splines. I started my literature search as follows:

```
RSiteSearch('spline')
```

While preparing this manuscript, this command identified 1526 documents. That is too much, so I restricted it to functions:

```
RSiteSearch('spline', 'fun')
```

This identified only 631. That's an improvement over 1526 but is still too much. To get a quick overview of these 631, we can proceed as follows:

```
splinePacs <- findFn('spline')
```

This downloaded a summary of the 400 highest-scoring help pages in the 'RSiteSearch' data base in roughly 5-15 seconds, depending on the speed of the Internet connection. To get all 631 hits, increase `maxPages`:

```
splineAll <- findFn('spline', maxPages=999)
```

As noted above, the `print` method will open the result in a web browser.

However, a table with 631 rows is rather large to digest easily. We could try the
summary method, but that produces a table with
HOW MANY?
rows. The simplest thing to do from here is to create an Excel file as follows:

```
writeFindFn2xls(splineAll)
```

This produces an Excel file (which can be opened with Open Office Calc
citation for Open Office Calc?
asdf, which can be useful for people who do not have Excel) with three sheets:

To find a function named `spline` from this, we can proceed as follows:

```
selSpl <- (splineAll[,'Function']=='spline')
splineAll[selSpl, ]
```

This has 0 rows, because there is no help page named `spline`.

We can expand this to include any help page containing `spline` in the name using `grepFn`:

```
> grepFn('spline', splineAll, ignore.case=TRUE)
```

This returned a `findFn` object identifying 66 help pages. The `print` method for an object of class `findFn` presents the result in a web browser,

asdf, the first of which is 'lspline' in the 'assist' package. The `RSiteSearch` engine assigned it a `Score` of 1. Evidently, that search engine found only minimal evidence of its relevance to the requested search `string`. It appeared at the top of this list, because the `assist` package had 34 help pages identifed as potentially relevant to that search `string`, none of which had a `Score` exceeding 1.

To establish priorities among different packages for further study, it might be nice to have a Pareto chart showing the 10 packages with the most help pages relevant to our search `string`. We can get this as follows:

```
> spSm <- attr(splineAll,'PackageSummary')
> spSm[1:10,'Count']
    assist     fda           gss       mgcv
        34      30            25         22
      VGAM kernlab DierckxSpline bayesSurv
        17      17            16         16
smoothSurv splines
        15      14
```

To obtain a similar Pareto by 'TotalScore' requires a little more effort:

```
> o <- rev(order(spSm[, 'TotalScore']))
> splineSum[o, ][1:10, ]
              Count MaxScore TotalScore
gss              25       35        448
splines          14       45        354
fda              30       48        275
<...>
```

This analysis gave us in seconds a very informative overview of `spline` capabilities in contributed R packages in a way that can help establish priorities for further study of the different packages and functions.

HTML

The `HTML` function writes an `RSiteSearch` object to a file in HTML format and opens it in a browser from which a mouse click will open a desired help page.

The power of this can be seen by applying this function to the `grep'ed` subset of help pages with names including the phrase `spline`:

```
HTML(splineAll[select, ])
```

Of the 631 help pages containing `spline`, this displayed only those whose name included the phrase `spline`. Similar analyses could display any desired subset of an `RSiteSearch` object created from merging several calls to `RSiteSearch.function`.

Summary

In sum, we have found `RSiteSearch.function` in the `RSiteSearch` package to be a very quick and efficient method for finding things in contributed packages.

Acknowledgments

The `RSiteSearch` capabilities here extend the power of the `RSiteSearch` search engine maintained by Jonathan Baron. Without Prof. Baron's support, it would not have been feasible to develop the features described here.

Spencer Graves
President and Chief Operating Officer
Structure Inspection and Monitoring
San Jose, CA
email: `spencer.graves@prodsyse.com`

Sundar Dorai-Raj
Google
Mountain View, CA
email: `sdorairaj@google.com`

Romain Francois

email: `romain.francois@dbmail.com`