

RjpWiki アーカイブス

【Rのヘルプ機能・必要な関数を探すコツ (06.06.24)】

Rの魅力の一つは充実した(オンライン)ヘルプ機能です。このチップスではそうした機能を紹介していきます。

1 基本 help(foo), ?foo

1.1 オブジェクト foo の man 形式のオンラインヘルプ

```
help(foo) #または簡略形 ?foo
```

1.2 パッケージ MASS 中のオブジェクト parcoord のヘルプ

パッケージ MASS が既にインストールされていることが前提.

```
help(parcoord, package=MASS)
```

1.3 演算子のヘルプ

演算子等は二重引用符で囲む必要がある.

```
help("for") # もしくは ?"for"  
?"+"
```

1.4 パッケージ "stepfun" 中のオブジェクト一覧

```
help(package="stepfun") # パッケージ "stepfun" 中のオブジェクト一覧 (パッケージ未ロードでも可)
```

1.5 データセットの解説

```
data() # 現在利用可能なすべてのデータセットの一覧  
?women # データセットの解説も同様に得られる
```

1.6 パッケージにあるデータのヘルプ

```
data(package = "base")      # パッケージ base で使用可能なデータセットを表示
data(USArrests, "VADeaths") # データセット 'USArrests' and 'VADeaths' をロード
help(USArrests)            # データセット 'USArrests' に関するヘルプ
```

R のヘルプは英語なので日本人には読みにくい場合がある。手っ取り早く使い方が見たいという場合、例えば `solve` の使い方が知りたい場合は以下のようにすれば例を出力してくれる。

```
example(solve)
```

ただし、グラフ出力がある場合で、グラフをいくつか含む場合は、単に `example` をやってしまうと数枚のグラフが一瞬で表示されきってしまう。動体視力に自信のない方は、以下を定義しておくだけでグラフの描画ごとに逐一静止してくれる。

```
par(ask = TRUE)
```

2 キーワードによるオブジェクトの検索 `help.search("foo")`

キーワードの一部で良い。現在インストール済みでパッケージ検索パス中にある全パッケージが対象。オブジェクトドキュメントのタイトル、別名、キーワード等からの検索を指定できる。キーワードは文字列なので二重括弧で囲む必要。キーワードの一部で良い。現在インストール済みでパッケージ検索パス中にある全パッケージが対象。オブジェクトドキュメントのタイトル、別名、キーワード等からの検索を指定できる。

```
help.search("linear models") # 説明に単語 'linear models' を含むドキュメント
                             # の検索 (大小文字を区別せずに検索)
help.search("print")         # トピックやタイトルが 'print' を含むドキュメント
help.search(apos = "print") # 上と同じ
help.search(keyword = "hplot") # キーワード "hplot" を持つドキュメント
help.search("\\btry", fields = "alias") # 別名の先頭が "try" で始まる
                                     # オブジェクト (正規表現使用)
```

3 latex 形式のヘルプファイルの見方

R およびアドオンパッケージをインストールすると適当なディレクトリに膨大なヘルプ文章が登録される。一つは `help()` 関数で表示される `man` 形式のドキュメントであるが、`latex` 形式のドキュメントも一揃い用意されている。これらは `Rd` 書式という独自のフォーマットで記述されたヘルプドキュメントから適当な Perl スクリプトでインストール時に生成される。`latex` 形式のドキュメントは最終的に `ps` ファイルに変換され印刷されることを想定している。

ソースから R システムを構築するとき `make install-dvi` とすれば `dvi` ファイルが一挙にできる。ソースファイル中にあるヘルプファイルの原型の `xxx.Rd` ファイルを個別に `R CMD Rd2dvi xxx.Rd` と処理すればやはり `dvi` ファイルが得られる。`help` 関数には `tex` 形式のヘルプファイ

ルを要求に応じてその場で処理し、ps ファイルを生成するオプションがある。ヘルプ関数は途中で出来た dvi ファイルは抹消する仕様になっている。シェル環境変数 'R.LATEXCMD' と 'R.DVIPSCMD' を指定しても良い。

```
options(latexcmd="/usr/bin/latex") # 環境変数
options(dvipscmd="/usr/bin/dvips")
help(optim, offline=TRUE)
```

Saving help page to 'optim.ps' # optim.tex から optim.ps が作られた

さらに次のようにすると xdvi によるオンライン表示が出来る (コンパイルに結構時間がかかるのが難)。

```
options(dvipscmd="/usr/bin/xdvi")
```

さらに上で dvips を dvipdfm に変えると pdf が作成 (/tmp 下) される。さらに dvipscmd をだましてやると、dvi ファイルも保存できるが、dvi ファイルは中間ファイルで名前が変。例えば

```
cp ./*.dvi /tmp
```

というスクリプトファイル (名前を仮に makedvi) に実行属性をつけ (*unix OS の場合) た上で

```
options(dvipscmd="./makedvi")
```

としてやる。

4 必要な関数を探すコツ・自分の使用環境で探す

4.1 キーワード検索関数 help.search()

R の多様な関数を使いこなすには、使えそうな関数を要領よく探すコツが要ります。例えば「ベクトル場のグラフィックスを描きたい」とします。平面上の点から矢印を描けば良いわけですが、それに使えるどんな関数があるのか分からないとします。矢印は arrow ですから、先ず R のキーワード検索関数 help.search() を使い

```
> help.search("arrow")
```

とします。キーワードは二重引用符で囲む必要があります¹。すると

```
Help files with alias or title matching 'arrow' using fuzzy matching:
```

```
arrows(base)          Add Arrows to a Plot
```

¹結構面倒なので、?hoge でヒットない場合は help.search("hoge") を提案してくれるので、それをコピーペーストすると楽です。

```
Type 'help(FOO, package = PKG)' to inspect entry 'FOO(PKG) TITLE'.
```

と表示されます。これはキーワードをふくむ関数が base パッケージに一つあったことを意味しています。(検索は現在使用中の機械にインストールされているすべてのパッケージ中から行なわれますから、こういう時のためにとりあえず使う予定のない追加パッケージも手あたり次第インストールしておくともよいかも。) 次に関数 `arrows()` はどのようなものか感触を得るために、

```
> example(arrows)
```

で例示用コードを実行します。`par(ask=TRUE)` を先に一度実行しておくこと、複数のグラフィックス出力画面が一気に流れ去ることを防ぎ、画面を順にゆっくり点検することができます。使えそうだなとなったら

```
> ?arrows          # または help(arrows)   (二重引用符は不要)
```

で関数の説明と、最後に付いている例示用コードを眺めます。すると、`arrows()` は直前の高機能作図関数の出力に矢印を加える関数であること、引数としてベクトルが使えるので一つ一つ描く必要が無いことなどが分かります。また、同様の関数として矢印の代わりに線分を描く関数 `segments()` があることも分かります。

4.2 help.seach()

`help.seach()` は正規表現による曖昧な検索も受け付けます

```
> help.search("ab*cd")    # 前後にそれぞれ文字列 ab, cd を含む文字列を検索
```

4.3 オブジェクトのキーワード検索関数 `apropos()`, `find()`

`help.search()` に似ている関数に、ほとんど同じ機能をもつ `apropos()`, `find()` があります。これはキーワードを含むオブジェクトを検索します。キーワードは二重引用符で囲んでも、囲まなくても構いません。正規表現による検索も可能です (この際は二重引用符が必要)。違いは、`apropos()` は現在ロードされているライブラリ中から、キーワードを含むオブジェクトだけを検索対象とすることです。

```
> apropos("lm")          # lm を含むオブジェクトを検索
> apropos(ls)            # ls を含むオブジェクトを検索
> length(apropos("."))   # すべてのオブジェクトの総数
> apropos("^pr")         # pr で始まるすべてのオブジェクトの検索
> apropos("^..?$")       # 2文字のオブジェクトを検索
> apropos("pr$")         # pr で終るオブジェクトを検索
> apropos("ub.{0,}a")    # ub と a の間に零個以上の文字列を含むパターン
> apropos("ub.{1,}a")    # ub と a の間に1個以上の文字列を含むパターン
```

4.4 組み込み関数の一覧を得る `builtins()`

R 2.3.0 版で 1091 種類

```
> builtins()
 [1] "zip.file.extract"          "zapsmall"
 [3] "xpdrows.data.frame"      "xor"
 [5] "writeLines"              "writeChar"
 (途中省略)
[1087] "!="                       "!"
[1089] "-.POSIXt"                "-.Date"
[1091] "-"
```

4.5 インストールされているパッケージの一覧を得る `search()`

現在使用中の計算機にインストールされているパッケージの一覧を得るためには、次の命令を使います。

```
> search()
> searchpaths() # インストール場所の絶対パスを得る
```

4.6 あるライブラリ中のオブジェクトの一覧を見る `library(help=パッケージ名)`

もちろんそのライブラリが既にインストールされている必要があります。

```
> library(gstat) # ライブラリ gstat をロード
> library(help=gstat) # gstat 中のオブジェクト (関数、データセット) の一覧を得る
```

5 必要な関数を探すコツ・検索サイトで探す

5.1 R site search でキーワード検索

R site search : <http://finzi.psych.upenn.edu/search.html>

R 関連ドキュメントからの最大の全文検索システム。というコマンドもあります。

5.2 `RSiteSearch?()`

```
> RSiteSearch("BUGS") # ブラウザが起動し BUGS の検索結果を表示
```

いろいろなオプションがあるので、ヘルプを参照。ブラウザを変更するには、環境変数 `R_BROWSER` を変更する。

5.3 S-Plus/R の関数のカテゴリ化されたリスト

S-Plus/R Function Finder : <http://biostat.mc.vanderbilt.edu/s/finder/finder.html>

5.4 インターネット (Google) で検索する

R のマニュアルや公式メイリングリストの過去記事からキーワード検索するには、

- CRAN のホームページにある検索エンジンを使う
- google で検索先サイトを指定して検索

例えば google でキーワード `rpart`, `NA` を含む文書、記事を検索するには次のようなキーワードを使います。

```
site:r-project.org rpart NA
```

また用例を探すには R のファイル形式を指定します。例えば `plot` の場合次のように google に入力します。拡張子が `r` という名前のファイルをネットで検索できます。

```
filetype:r plot
```

5.5 中間さんの CRAN code search by gonzui を試す

キーワードを含む R 関係のコードを検索してくれます。

CRAN code search by gonzui : <http://rgonzui.nakama.ne.jp/CRAN>