

# Package ‘RWsearch’

August 19, 2019

**Title** Lazy Search in R Packages, Task Views, CRAN, the Web. All-in-One Download

**Description** Search by keywords in R packages, task views, CRAN, the web and display the results in console, txt, html or pdf pages. Download the whole documentation (html index, pdf manual, vignettes, source code, etc) with a single instruction. Visualize the package dependencies. Several functions for task view maintenance and exploration of CRAN archive. Quick links to more than 70 web search engines. Lazy evaluation of non-standard content is available throughout the package and eases the use of many functions.

**Version** 4.6.2

**Date** 2019-08-19

**Depends** R (>= 3.4.0)

**Imports** brew, latexpdf, networkD3, sig, sos, XML

**Suggests** ctv, cranly, findR, foghorn, knitr, pacman, pkgnet, rmarkdown

**License** GPL-2

**Maintainer** Patrice Kiener <fattailsr@inmodelia.com>

**Author** Patrice Kiener [aut, cre] (<<https://orcid.org/0000-0002-0505-9920>>)

**Encoding** UTF-8

**NeedsCompilation** no

**LazyData** false

**Language** en-GB

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**Repository** CRAN

**Date/Publication** 2019-08-19 19:00:09 UTC

## R topics documented:

RWsearch-package . . . . .	2
archivedb . . . . .	4
cncs . . . . .	6

crandb . . . . .	7
e_check . . . . .	9
funmaintext . . . . .	10
f_args . . . . .	11
f_pdf . . . . .	12
h_direct . . . . .	12
h_engine . . . . .	13
h_R . . . . .	17
h_ttp . . . . .	18
n_graph . . . . .	19
p_deps . . . . .	20
p_display . . . . .	21
p_down . . . . .	22
p_html . . . . .	24
p_inun . . . . .	25
p_table2pdf . . . . .	26
p_text2pdf . . . . .	28
s_crandb . . . . .	30
s_crandb_tvdb . . . . .	32
s_hs . . . . .	34
s_sos . . . . .	35
s_tvdb . . . . .	36
tvdb . . . . .	37
zcrandb . . . . .	38
ztvdb . . . . .	38
<b>Index</b>	<b>40</b>

---

RWsearch-package	<i>Package RWsearch</i>
------------------	-------------------------

---

## Description

Search by keywords in R packages, task views, CRAN, the web and display the results in console, txt, html or pdf pages. Download the whole documentation (html index, pdf manual, readme, vignettes, source code, etc) with a single instruction, either in a flat format or in subdirectories defined by the keywords. Visualize the package dependencies. Several functions for task view maintenance and exploration of CRAN archive. Quick links to more than 70 web search engines. Lazy evaluation of non-standard content is available throughout the package and eases the use of many functions. Packages RWsearch and pacman share the same syntax and complement each other. Inspired by packages ctv, fohorn, latexpdf, pacman, sos.

## Examples

```
### NON-STANDARD CONTENT - NON-STANDARD EVALUATION
## Non-standard content (nsc1, nsc2), standard content ("stc3", "double word4")
## and regular object (obj) stored in .GlobalEnv can be merged with cnsc()
obj <- c("obj5", "obj6")
```

```
cnsc(nsc1, nsc2, "stc3", "double word4", obj)

### DOWNLOAD CRANDB
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

### SEARCH IN CRANDB
## Search in crandb. Use standard or non-standard content.
## Display the results in a vector or in a list.
s_crandb(search, find, cran, web)
s_crandb(search, find, cran, web, select = "PD", mode = "relax")
s_crandb(c("thermodynamic", "chemical reaction"))
(lst <- s_crandb_list(thermodynamic, "chemical reaction"))

### DISPLAY THE RESULTS
## in the console, in (txt, md, pdf) files or in the browser.
p_table2(lst)
p_table7pdf(lst, dir = file.path(tempdir(), "ptable"), cleantex = FALSE, openpdf = TRUE)
p_text(lst, dir = file.path(tempdir(), "ptext2"), editor = TRUE,
       repos = "https://cran.univ-paris1.fr")
p_text2pdf(lst, dir = file.path(tempdir(), "ptext2"), cleantex = FALSE,
          openpdf = TRUE, repos = "https://cran.univ-paris1.fr")
p_display(lst, dir = tempdir())

### VISUALIZE THE DOCUMENTATION
## from installed packages or U. Pennsylvania in the browser
p_html(brew, sig)
p_htmlweb(foghorn)
p_pdfweb(sos, repos = "https://cran.univ-paris1.fr")

### DOWNLOAD THE DOCUMENTATION
## Vector => download in the "docpkgs" directory ( "." is for current directory)
## List => download in subdirectories named after the keywords
## (non-standard content is accepted)
p_down(pacman, pdfsearch, sos, dir = file.path(tempdir(), "pdown"),
       repos = "https://cran.univ-paris1.fr")
p_down(lst, dir = file.path(tempdir(), "pdown"), repos = "https://cran.univ-paris1.fr")

### SEARCH WITH sos (U. PENNSYLVANIA)
s_sos("chemical reaction")
res <- s_sos(distillation)
as.data.frame(res)
res

### LAUNCH WEBSITES AND SEARCH ENGINES
h_ftp("www.r-project.org")
h_cranbydate(repos = "https://cran.univ-paris1.fr")
h_yt("Serge Gainsbourg Ne dis rien")
h_so(R, deep, neural, network)
h_osm("La Ferriere sous Jougne")
h_mw(recension)
```

```

h_lexilogos()

### TASK VIEW MAINTENANCE
## In real life, download crandb and tvdb from CRAN or load them from your directory
## with functions crandb_down(), crandb_load(), tvdb_down(), tvdb_load().
## In this example, we use small files.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
tvdb_load(system.file("data", "ztvdb.rda", package = "RWsearch"))

## List the task views
tvdb_vec()
tvdb_pkgs(gR, Genetics, Robust)

## Search for some packages in the task views
s_tvdb(actuar, FatTailsR, MASS, zoo, nopackage)

## Search for the recent packages in crandb that contain the keyword
## and verify if the packages are already refereed in the task view.
## from = "2017-01-01" and "2018-01-01" are selected for this small example.
s_crandb_tvdb("distribution", tv = "Distributions", from = "2017-01-01")
s_crandb_tvdb("distribution", tv = "Distributions", from = "2018-01-01")

### EXPLORE CRAN ARCHIVE AND DOWNLOAD OLD tar.gz FILES
## In real life, download archivedb and crandb from CRAN
## with the functions archivedb_down() and crandb_down().
## In this example, we load two small files (50 and 43 packages).
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
archivedb_load(system.file("zarchive", "zCRAN-archive.html", package = "RWsearch"))
archivedb_npkgs()
lapply(archivedb_list(), tail)

## Download the latest tar.gz version from CRAN archive
## (this works for both both existing and removed packages).
p_downarch(fitur, zmatrix, dir = file.path(tempdir(), "pdownarch"))

```

---

archivedb

*CRAN archive (CRAN-archive.html + archivedb)*


---

## Description

The following functions deal with the packages archived in CRAN. The html file downloaded from CRAN contains the regular packages that have been updated once and the packages that have been removed from CRAN by CRAN administrators. It does not contain the first version of the packages uploaded to CRAN and never updated. These files and the files removed from CRAN index can be guessed through a comparison with crandb.

archivedb\_down downloads from CRAN the html file of the archived packages, saves it on the disk under the name filename, extracts from it and loads in .GlobalEnv a data.frame named archivedb.

`archivedb_load` reads the html file filename saved on the disk, extracts from it and loads in `.GlobalEnv` a data.frame named `archivedb`.

`archivedb_npkgs` returns the number of packages listed each category: number of packages in `crandb`, in `archivedb`, at first version, at subsequent version and removed from `crandb` (CRAN index).

`archivedb_pkgs` returns the packages listed in CRAN archive (= `archivedb`).

`archivedb_rempkgs` returns the archived packages removed from CRAN regular index.

`archivedb_list` compares the data.frame `archivedb` and `crandb` and returns a list with the following items:

- `pkgs_crandb`: the packages listed in `crandb`.
- `pkgs_archivedb`: the packages listed in `archivedb`.
- `pkgs_first`: the packages in first version in `crandb`.
- `pkgs_updated`: the packages with more than one version in `crandb`.
- `pkgs_removed`: the archived packages removed from CRAN regular index, i.e. not listed in `crandb`.
- `dfr_crandb`: data.frame `pkgs_crandb` + Published date.
- `dfr_archivedb`: data.frame `pkgs_archivedb` + Archived date.
- `dfr_first`: data.frame `pkgs_first` + Published date.
- `dfr_updated`: data.frame `pkgs_updated` + Published date.
- `dfr_removed`: data.frame `pkgs_removed`+ Archived date.
- `npkgs`: the number of packages in each category.

`p_downarch` downloads the latest tar.gz version of the package(s) listed in CRAN archive.

### Usage

```
archivedb_down(filename = "CRAN-archive.html", dir = ".",  
url = "https://cran.r-project.org/src/contrib/Archive")
```

```
archivedb_load(filename = "CRAN-archive.html")
```

```
archivedb_npkgs(archivedb = get("archivedb", envir = .GlobalEnv),  
crandb = get("crandb", envir = .GlobalEnv))
```

```
archivedb_pkgs(archivedb = get("archivedb", envir = .GlobalEnv))
```

```
archivedb_rempkgs(archivedb = get("archivedb", envir = .GlobalEnv),  
crandb = get("crandb", envir = .GlobalEnv))
```

```
archivedb_list(archivedb = get("archivedb", envir = .GlobalEnv),  
crandb = get("crandb", envir = .GlobalEnv))
```

```
p_downarch(..., char = NULL, dir = ".", archivedb = get("archivedb",  
envir = .GlobalEnv),  
url = "https://cran.r-project.org/src/contrib/Archive")
```

## Arguments

filename	character. The path to file "CRAN-archive.html" (or equivalent).
dir	character. The directory where filename or tar.gz files are saved. Default value "." is the current directory.
url	character. The url address of CRAN archive html file.
archivedb	data.frame archivedb. The archivedb data.frame format loaded in memory by archivedb_down or archivedb_load.
crandb	data.frame crandb. The data.frame of CRAN packages.
...	any format recognized by <a href="#">cncs</a> , except list. A vector of packages.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

## Examples

```
### DOWNLOAD archivedb AND COMPARE IT WITH crandb.
## In real life, download archivedb and crandb from CRAN
## with the functions archivedb_down() and crandb_down().
## In this example, we load two small files (50 and 43 packages).

crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
archivedb_load(system.file("aabb", "zCRAN-archive.html", package = "RWsearch"))
archivedb_npkgs()

lst <- archivedb_list()
lapply(lst, head)
lapply(lst, tail)
range(lst$dfr_removed$Archived)
hist(lst$dfr_removed$Archived, breaks = 15, freq = TRUE, las = 1)

## Download the latest tar.gz version from CRAN archive
## (this works for both both existing and removed packages).
p_downarch(fitur, zmatrix, dir = file.path(tempdir(), "pdownarch"))
```

## Description

cncs converts standard content and non-standard content ... into a character chain. Standard content, including *calls*, is evaluated. Non-standard content and non-existing objects (in `.GlobalEnv`) are converted into character chains. Regular lists are kept unchanged.

cncsfun is intended to be used inside a function. It should not be used directly.

**Usage**

```
cncs(...)

cncscinfun()

cncscinfun2(...)
```

**Arguments**

```
...          Character vectors, standard or non-standard, existing or non-existing R objects,
             regular call. Examples : "word1"; c("word1 word2"); c("word1", "word2");
             "word1", "word2", "word3"; word1, word2, c("word3", "word4").
```

**Examples**

```
### cncs
## Non-standard content (nsc1, nsc2), standard content ("stc3", "double word4")
## and regular object (vec) stored in .GlobalEnv are merged.
vec <- c("obj5", "obj6")
cncs(nsc1, nsc2, "stc3", "double word4", vec)

## Lists, either name in .GlobalEnv or call, are evaluated.
lst <- list(A = c("txt1", "txt2", "txt3"), B = c("txt4", "txt5"))
cncs(lst)
cncs(list(C = c("pkg1", "pkg2", "pkg3"), D = c("pkg4", "pkg5")))

### cncscinfun
fun <- function(...) cncscinfun()
fun(nsc1, nsc2, "stc3", "double word4", vec)
fun(lst)
```

---

crandb

*CRAN Packages (crandb.rda)*


---

**Description**

crandb\_down downloads from CRAN the file *packages.rds*, a file refreshed everyday that describes the packages available in CRAN for this day, rename (with `make.names`) the column names that are syntactically invalid, removes the duplicated lines located at the end of the file, cleans some bad characters in the Description column, loads the resulting data.frame in `.GlobalEnv` under the name *crandb* and saves it in the current directory with the name *crandb.rda*. If `oldfile` is defined, the vector of packages between the two files is compared.

crandb\_load loads the file `filename` in `.GlobalEnv` under the name *crandb*. Equivalent to `load("crandb.rda")`. Use this function if you are not connected to internet or do not want to refresh your file.

crandb\_pkgs displays all packages listed in *crandb*. The number of packages is larger than the number obtained with `nrow(available.packages())` since packages for all OSes are counted.

crandb\_fromto displays the packages published in CRAN between two dates.

**Usage**

```

crandb_down(dir = ".", oldfile = "crandb.rda", verbose = TRUE,
  repos = getOption("repos")[1])

crandb_load(filename = "crandb.rda")

crandb_comp(filename = "crandb.rda", oldfile = "crandb-old.rda",
  addtxt = "")

crandb_pkgs(bydate = FALSE, rev = FALSE, crandb = get("crandb", envir
  = .GlobalEnv))

crandb_fromto(from = -10, to = Sys.Date(), crandb = get("crandb",
  envir = .GlobalEnv))

```

**Arguments**

<code>dir</code>	character. The directory where "crandb.rda" is saved and the old "crandb.rda" is read. Default value "." is the current directory.
<code>oldfile</code>	character or NULL. The (path to an) old file that will be compared to a freshly downloaded version of "crandb.rda" or to filename. Set to NULL if no comparison is required.
<code>verbose</code>	logical. TRUE prints the result. FALSE keeps it invisible.
<code>repos</code>	character. The address of your local CRAN.
<code>filename</code>	character. The (path to a) file "crandb.rda" or an equivalent.
<code>addtxt</code>	character. Internal use.
<code>bydate</code>	logical. List the package by date of publication rather than by alphabetical order.
<code>rev</code>	logical. Print in reverse order.
<code>crandb</code>	data.frame crandb. The data.frame of CRAN packages.
<code>from</code>	Negative integer or character representing a date. The number of days preceding to or a date before to.
<code>to</code>	date. The upper date in the search.

**Examples**

```

### In this first example, we use a small file.
## List the 50 packages of this file, the ones uploaded since 2019-01-01
## and those uploaded in the last 15 days before the last date (2019-02-10)
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
crandb_pkgs()
dim(crandb)
colnames(crandb)
crandb$Published
crandb_fromto(from = "2019-01-01", to = Sys.Date())
crandb_fromto(from = -15, to = max(crandb$Published))

```



```

### Now, we use the big file (7 MB) that contains the list of all packages.
## Download from your local CRAN a fresh version of crandb (5-20 seconds)
## and query the packages of the last 2 days
crandb_down(dir = tempdir(), repos = "https://cran.univ-paris1.fr")
ls()
pkgs <- crandb_fromto(-2) ; pkgs
p_table2(pkgs) # Print in the console (better if full width)
p_display7(pkgs, dir = tempdir()) # Display in the browser

```

---

e\_check

*Package Check Results By Email Address*


---

### Description

e\_check opens the browser and returns the "CRAN Check Results" page(s) of the packages maintained by one or several maintainers identified by their regular email addresses (but not the orphaned ones). An internet connection is required.

This function is a simplified version of the functions proposed in package *foghorn*.

### Usage

```
e_check(..., char = NULL, repos = getOption("repos")[1])
```

### Arguments

... any format recognized by [cnsC](#), except list. A vector of quoted "e-mail addresses".

char (name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

repos character. The address of your local CRAN.

### Examples

```
e_check(c("FatTailsR@inmodelia.com", "francois.michonneau@gmail.com"),
        repos = "https://cran.univ-paris1.fr")
```

---

 funmaintext

---

*Modify the Main Text and the Markdown Header in p\_text Function*


---

### Description

Use funmaintext or funmaintext2 to select the function that displays the main text. Usage is `f_maintext = funmaintext` (without curly braces).

Use funheadermd to insert markdown header in function `sep1 = funheadermd()` (with curly braces).

See the example in `p_text`. To create you own functions, use these functions as a pattern. The five parameters in `f_maintext`, `funmaintext` are mandatory. `funheadermd` can be freely modified.

### Usage

```
funmaintext(pkg, sep1, sep2, eol, crandb, repos)
```

```
funmaintex(pkg, sep1, sep2, eol, crandb, repos)
```

```
funheadermd(title = "TITLE", author = "AUTHOR", date = Sys.Date(),
  keep_tex = "false", toc = "false", number_sections = "true",
  fontsize = "10pt", papersize = "a4paper", margin = "1in")
```

```
funheadertex(fontsize = "10pt", papersize = "a4paper",
  margin = "1in")
```

```
funfootertex()
```

### Arguments

<code>pkg</code>	character. The package name.
<code>sep1</code>	character. The symbols written just before each package name.
<code>sep2</code>	character. The symbols written just after each package name.
<code>eol</code>	character. The end of line for the main text (but not for the header and the footer). <code>"\n"</code> for text, <code>" \n"</code> for rmarkdown, <code>" \\ \n"</code> for latex.
<code>crandb</code>	data.frame <code>crandb</code> . The data.frame of CRAN packages.
<code>repos</code>	character. The address of your local CRAN.
<code>title</code>	character. The title of the .md document (and then in the .pdf file).
<code>author</code>	character. The author of the .md document.
<code>date</code>	character. The date of the document. Any text format is accepted.
<code>keep_tex</code>	character. "true" or "false".
<code>toc</code>	character. "true" or "false".
<code>number_sections</code>	character. "true" or "false".
<code>fontsize</code>	character. Usually "10pt", "11pt", "12pt".

papersize	character. The usual tex format. Example: "a4paper".
margin	character. In inches, cm or mm. Example: "0.5in", "1.5cm", "25mm".

---

f\_args

*Names and Arguments of Functions*


---

### Description

f\_args is a wrapper of the base function `args`.

f\_sig prints the name and arguments of one or several functions in a readable style. It wraps the function `sig::sig`.

p\_sig prints the name and arguments of the functions exported by one or several packages. It wraps the function `sig::list_sigs`.

### Usage

```
f_args(..., char = NULL)
```

```
f_sig(..., char = NULL)
```

```
p_sig(..., char = NULL, pattern = NULL)
```

### Arguments

... any format recognized by `cnsc`, except list. A vector of unquoted "functions" or characters.

char (name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

pattern a regular expression. See the example.

### Examples

```
f_args(mean, p_display)
f_sig(mean, p_display)
```

```
library(sos)
p_sig("sos")
p_sig(RWsearch, sos, pattern = "^f")
```

---

f\_pdf

*PDF Pages of Functions*


---

### Description

f\_pdf generates in the current directory the pdf pages of one or several functions. The pdf pages are printed but not opened. Miktex or Texlive is required. See also packages *document* and *sinew*.

### Usage

```
f_pdf(..., char = NULL)
```

### Arguments

... any format recognized by `cns`, except list. A vector of quoted "package::function".

char (name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

### Examples

```
f_pdf(c("RWsearch::cns", "RWsearch::p_inun"))
```

---

h\_direct

*Open a Web Page in the Browser*


---

### Description

h\_academie gives access to the Academie francaise's dictionary.

h\_framasoft gives access to several free web services (as in speech and in half pint of beer) that are good alternatives to GAFSA services.

h\_lexilogos gives access to hundreds of dictionaries in many languages.

h\_linguee gives access to a translation engine in many languages. Many examples with long sentences.

h\_reverso gives access to a translation engine and dictionnaires in many languages.

h\_tad and h\_tadsm gives access to a website dedicated to date and time conversion plus timezone management.

h\_yacy is a decentralized peer-to-peer web search software.

Using the regular R format "w1 w2 w3" rather than w1, w2, w3 makes sense as most engines collapse the words into character chains "w1 w2 w3", "w1+w2+w3" or "w1-w2-w3".

**Usage**

```
h_academie()  
  
h_framasoft()  
  
h_framasoft0()  
  
h_lexilogos()  
  
h_linguee()  
  
h_reverso()  
  
h_tad()  
  
h_tadsm()  
  
h_yacy()
```

**Examples**

```
h_linguee()  
h_lexilogos()
```

---

h\_engine

*Explore the Web with Various Search Engines*

---

**Description**

Launch the default browser and search in: 1bis Map (BottinCarto), ABC Bourse (short stock names), arXiv (vectorized), Ask, Baidu, Blackle, Bing, Bing Map (bmap), Boursorama (short stocknames), CNRTL (French dictionary), Crossref (DOI and bibliographic metadata), Daum, DailyMotion (dm), DuckDuckGo (ddg), Ecosia, Egerin, Evene (citations), Exalead, Excite, Gigablast, GitHub, GitLab, Google Map (gmap), Google, Google Scholar (gscholar), Info, Khoj, Les Echos, La Tribune (lt), Lilo, Lycos, Mappy Map, Merriam-Webster (mw, English dictionary), Nabble, Nate, Naver (see N2H4 package), Orcid, Open Street Map, OSM Nominatim, Parsijoo, PeerTube, Peru, Pipilika, Qwant (qw + qwfr), R-bloggers, Rdocumentation (rdoc), Rdocumentation task views (rdoctv), Rdr, Reverso dictionary, Rseek, Sapo, Searx, Sogou, SSRN and SSRN Author (vectorized), Stackoverflow (so), Startpage (ex-Ixquick), Twitter (+ twfr), L'Usine Nouvelle (un), ViaMichelin Map and Routes (via), Les Verbes, Vimeo, Wego (Here maps), Wikipedia (wp + wpfr), Yahoo, Yahoo Finance, Yandex, Yooz, Youtube (yt).

Using the regular R format "w1 w2 w3" rather than w1, w2, w3 makes sense as most functions collapse the words into character chains "w1 w2 w3", "w1+w2+w3" or "w1-w2-w3".

Visit [https://en.wikipedia.org/wiki/Web\\_search\\_engine](https://en.wikipedia.org/wiki/Web_search_engine) for a list of web search engines.

**Usage**

h\_1bis(..., char = NULL)  
h\_abcbourse(..., char = NULL)  
h\_ask(..., char = NULL)  
h\_arxiv(..., char = NULL)  
h\_arxivpdf(..., char = NULL)  
h\_baidu(..., char = NULL)  
h\_blackle(..., char = NULL)  
h\_bing(..., char = NULL)  
h\_bmap(..., char = NULL)  
h\_boursorama(..., char = NULL)  
h\_cnrtl(..., char = NULL)  
h\_crossref(..., char = NULL)  
h\_daum(..., char = NULL)  
h\_dm(..., char = NULL)  
h\_ddg(..., char = NULL)  
h\_ecosia(..., char = NULL)  
h\_egerin(..., char = NULL)  
h\_estrep(..., char = NULL)  
h\_evene(..., char = NULL)  
h\_exalead(..., char = NULL)  
h\_excite(..., char = NULL)  
h\_framabee(..., char = NULL)  
h\_gigablast(..., char = NULL)  
h\_github(..., char = NULL)

```
h_gitlab(..., char = NULL)
h_gmap(..., char = NULL)
h_google(..., char = NULL)
h_gschohar(..., char = NULL)
h_info(..., char = NULL)
h_ixquick(..., char = NULL)
h_khoj(..., char = NULL)
h_lesechos(..., char = NULL)
h_lilo(..., char = NULL)
h_lt(..., char = NULL)
h_lycos(..., char = NULL)
h_mappy(..., char = NULL)
h_mw(..., char = NULL)
h_nate(..., char = NULL)
h_naver(..., char = NULL)
h_orcid(..., char = NULL)
h_osm(..., char = NULL)
h_osmn(..., char = NULL)
h_parsijoo(..., char = NULL)
h_peertube(..., char = NULL)
h_peru(..., char = NULL)
h_pipilika(..., char = NULL)
h_qwant(..., char = NULL, lang = "en")
h_qwfr(..., char = NULL)
```

```
h_reverso_d(..., char = NULL)
h_sapo(..., char = NULL)
h_searx(..., char = NULL)
h_so(..., char = NULL)
h_sogou(..., char = NULL)
h_ssrn(..., char = NULL)
h_ssrnauth(..., char = NULL)
h_startpage(..., char = NULL)
h_twfr(..., char = NULL)
h_twitter(..., char = NULL, lang = "en")
h_un(..., char = NULL)
h_verbes(..., char = NULL)
h_via(..., char = NULL)
h_vimeo(..., char = NULL)
h_wego(..., char = NULL)
h_wp(..., char = NULL, lang = "en")
h_wpfr(..., char = NULL)
h_yahoo(..., char = NULL, lang = "en")
h_yahoofin(..., char = NULL, lang = "en")
h_yandex(..., char = NULL)
h_yooz(..., char = NULL)
h_yt(..., char = NULL)
```

### Arguments

... any format recognized by [cnsr](#), except list. A vector of packages.



char	(name to) a character vector. Use this argument if . . . fails or if you call the function from another function.
lang	character. The language accepted by the search engine, usually "en", "de", "es", "fr", "jp", etc.

### Examples

```
h_yt("Serge Gainsbourg Ne dis rien")
h_so(R, deep, neural, network)
h_osm("Le Chateau d'Oleron")
h_mw(recension)
h_arxiv(c(1212.4320, 1605.08732))
```

---

h\_R

*Open a Web Page in the Browser*

---

### Description

h\_R opens the page <https://www.r-project.org>.

h\_cran opens the page of you local CRAN.

h\_crandate opens the page of CRAN packages sorted by date of publication.

h\_crantv opens the page of CRAN task views.

h\_cranberries, h\_nabble, h\_rbloggers, h\_rdoc, h\_rdoctv (RDocumentation), h\_rdr, h\_rseek open the pages of web sites related to R.

h\_gepuro lists all (most) R packages available on GitHub.

### Usage

```
h_R()
```

```
h_cran(repos = getOption("repos")[1])
```

```
h_cranbydate(repos = getOption("repos")[1])
```

```
h_cranbyname(repos = getOption("repos")[1])
```

```
h_crantv(repos = getOption("repos")[1])
```

```
h_cranstatus()
```

```
h_cranberries()
```

```
h_gepuro()
```

```
h_nabble(..., char = NULL)
```

```
h_rbloggers(..., char = NULL)
```

```
h_rdoc(..., char = NULL)
```

```
h_rdoctv(..., char = NULL)
```

```
h_rdrd(..., char = NULL)
```

```
h_rseek(..., char = NULL)
```

### Arguments

repos	character. The address of your local CRAN.
...	any format recognized by <code>cnsc</code> , except list. A regular web address.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function.

### Examples

```
h_cranbydate()
h_cranberries()
```

---

h\_ftp

*Open a Web Page in the Browser*

---

### Description

h\_ftp opens the page corresponding to the mentioned address in the default browser.

### Usage

```
h_ftp(..., char = NULL, https = TRUE, www = FALSE)
```

### Arguments

...	any format recognized by <code>cnsc</code> , except list. A regular web address.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function.
https	logical. Use https or http.
www	logical. Add www. to the address.

**Examples**

```
h_ttp("www.r-project.org")
```

---

n\_graph

*Graphs of Dependencies*


---

**Description**

These two functions display in the default browser a network of the package dependencies as a standard graph `n_graphF` (F for force in networkD3 catalogue) or as a Sankey graph `n_graphS`. The graphs are built (and can be saved) with the *networkD3* package.

Remember that the option `p_network(exclpkgs = ...)` in `p_network`, whose default value TRUE is equivalent to `exclpkgs = c("graphics", "grDevices", "methods", "stats", "tools", "utils")`, can substantially modify the aspect of the graph, especially for `reverse = FALSE`.

**Usage**

```
n_graphF(netw, group = 2, fontFamily = "serif", fontSize = 11,
  linkDistance = 50, charge = -100)
```

```
n_graphS(netw, group = 2, fontFamily = "serif", fontSize = 14,
  nodeWidth = 30, nodePadding = 10)
```

**Arguments**

<code>netw</code>	a list of class "pkgsnetwork" produced by <code>p_network</code> that describes with nodes and links the dependencies of one or several packages (a network).
<code>group</code>	integer, currently 1, 2 or 3. The suffix of the "NGroup" column in <i>netw</i> . Define a scheme for colouring the nodes.
<code>fontFamily</code>	character. Either "serif" or "sans-serif".
<code>fontSize</code>	integer. The size of the font.
<code>linkDistance</code>	integer. The minimal distance of a link between two nodes.
<code>charge</code>	integer. A repulsive value between two nodes.
<code>nodeWidth</code>	integer. The width of the rectangular nodes in the Sankey graph.
<code>nodePadding</code>	integer. The vertical space between two nodes in the same column of a Sankey graph.

## Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
netw <- p_network(stringr, methods, parallel, stats, utils, reverse = TRUE)
n_graphF(netw)
n_graphS(netw)

n_graphF(p_network(canprot, FatTailsR, actuar, exclpkgs = FALSE))
```

---

p\_deps

*Dependencies and Reverse Dependencies of Packages*

---

## Description

p\_deps returns the (reverse) dependencies of a (vector of) package(s). It is a wrapper of the `tools::package_dependencies` function. A warning is issued for packages that are not in `crandb + .libPaths()` (for instance in BioConductor).

p\_depsrev returns the reverse dependencies.

p\_network returns the package dependencies as a network of nodes and links. It is called by [n\\_graphF](#) and [n\\_graphS](#).

## Usage

```
p_deps(..., char = NULL, which = "DIL", recursive = FALSE,
       reverse = FALSE, verbose = getOption("verbose"),
       crandb = get("crandb", envir = .GlobalEnv))
```

```
p_depsrev(..., char = NULL, which = "DIL", recursive = FALSE,
          reverse = FALSE, verbose = getOption("verbose"),
          crandb = get("crandb", envir = .GlobalEnv))
```

```
p_network(..., char = NULL, which = "DIL", reverse = FALSE,
          exclpkgs = TRUE, crandb = get("crandb", envir = .GlobalEnv))
```

## Arguments

...	any format recognized by <a href="#">cns</a> , excluding list. A package or a vector of packages listed in <code>crandb</code> or in <code>installed.packages()</code> .
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

which	character vector. A sub-vector of <code>c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances")</code> . The short forms "D", "I", "L", "S", "N", "DL", "DI", "DIL", "DILS", "DILN", "DILSN", "SN" are accepted. "N" is for "Enhances" as the single letter "E" is used by R as a shortcut to <code>EXPR</code> , a reserved word.
recursive	logical. Search for (reverse) dependencies of (reverse) dependencies.
reverse	logical. Search for reverse dependencies.
verbose	logical. Returns additional information about the search.
crandb	<code>data.frame</code> <code>crandb</code> . Also accepted is <code>NULL</code> which will search in the local <code>installed.packages()</code> . This later form allows (private) packages that are not listed in <code>crandb</code> .
exclpkgs	logical or character vector. <code>TRUE</code> excludes from the network of nodes and links the dependencies <code>c("graphics", "grDevices", "methods", "stats", "tools", "utils")</code> . <code>FALSE</code> includes them. You can provide your own vector of packages to exclude them from the network of nodes and links.

### Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

p_deps(canprot, FatTailsR)
p_deps(canprot, FatTailsR, recursive = TRUE)
p_deps(canprot, FatTailsR, recursive = TRUE, which = "DIL")
p_deps(actuar, reverse = TRUE, which = "DILSN")

p_network(canprot, FatTailsR, exclpkgs = FALSE)
```

---

p\_display

*Display Package Information in HTML Pages*

---

### Description

`p_display`, `p_display5` and `p_display7` open the default browser and display the results of `p_table`, `p_table5` and `p_table7` in one or several html pages. If `...` (or `char`) is a list, several pages are opened.

### Usage

```
p_display(..., char = NULL, columns = c("Package", "Title",
    "Description"), dir = tempdir(), verbose = FALSE,
    crandb = get("crandb", envir = .GlobalEnv))

p_display5(..., char = NULL, dir = tempdir(), verbose = FALSE,
    crandb = get("crandb", envir = .GlobalEnv))

p_display7(..., char = NULL, dir = tempdir(), verbose = FALSE,
    crandb = get("crandb", envir = .GlobalEnv))
```

**Arguments**

...	any format recognized by <code>cnsc</code> , including list. A vector or a list of packages. Or a vector or a list of data.frame produced by <code>p_table</code> .
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
columns	character vector. A sub-vector of <code>colnames(crandb)</code> . The short form "P", "T", "D", "PT", "PD", "TD", "PTD", "A", "M", "AM" describing the Package name, Title, Description, Author, Maintainer or a combination of them is accepted.
dir	character. The directory in which the html file(s) is (are) saved. <code>tempdir()</code> or <code>getwd()</code> are common paths.
verbose	logical. List the generated html file(s).
crandb	data.frame crandb. The data.frame of CRAN packages.

**Examples**

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

## Vector => 1 page
p_display(pacman, pdfsearch, sos, dir = tempdir())

## List with 3 items => 3 pages
## No package has the 'distillation' keyword. An empty table is returned.
(lst <- s_crandb_list("thermodynamic", "chemical reaction", "distillation"))
p_display5(lst, dir = tempdir())
```

---

p\_down

*Download Package Documentation in One Directory or Several Sub-directories*

---

**Description**

If `pkgs` is a vector of packages obtained from `s_crandb`, `p_down` downloads from CRAN and saves in the `dir` directory (by default the current directory) the index page, the manual, the vignettes, the README, NEWS, ChangeLog, CRAN checks files, the source code in `pkg_ver.tar.gz` format and a minimal R-script of each package. The files that do not exist in CRAN are ignored, with no warning.

If `pkgs` is a list of packages obtained from `s_crandb_list`, `p_down` saves the downloaded files in subdirectories named after the names of the list, e.g. the keywords used at the search step. The names are eventually modified with `gsub(".", "_", make.names(pkg), fixed = TRUE)` to cope with Unix and Windows directory names.

p\_down0 calls p\_down with different values for each argument. With the default configuration, this function downloads nothing. It is mostly used to download one specific item which has not been previously downloaded.

Visit [p\\_downarch](#) to download tar.gz file(s) from CRAN archive.

## Usage

```
p_down(..., char = NULL, index = TRUE, manual = TRUE,
  vignettes = TRUE, README = TRUE, NEWS = FALSE, ChangeLog = FALSE,
  checks = FALSE, targz = FALSE, script = FALSE, dir = ".",
  crandb = get("crandb", envir = .GlobalEnv),
  repos = getOption("repos")[1])
```

```
p_down0(..., char = NULL, index = FALSE, manual = FALSE,
  vignettes = FALSE, README = FALSE, NEWS = FALSE,
  ChangeLog = FALSE, checks = FALSE, targz = FALSE, script = FALSE,
  dir = ".", crandb = get("crandb", envir = .GlobalEnv),
  repos = getOption("repos")[1])
```

## Arguments

...	any format recognized by <a href="#">cnsr</a> , including list. A vector or packages or a named list of packages (with names being the keywords).
char	(name to) a character vector or a list. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
index	logical. Download the html index page of each package.
manual	logical. Download the pdf manual.
vignettes	logical. Download the html and pdf vignettes, if they exist.
README	logical. Download the README file, if it exists.
NEWS	logical. Download the NEWS file, if it exists.
ChangeLog	logical. Download the ChangeLog file, if it exists.
checks	logical. Download the CRAN checks file.
targz	logical. Download the *.tar.gz source file.
script	logical. Create a mini-script to test the package.
dir	character. The directory in which the files are saved. Default value "." is the current directory.
crandb	data.frame crandb. The data.frame of CRAN packages.
repos	character. The address of your local CRAN.

## Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
```

```
## Download the documentation in the "dirpkgs" directory. Flat representation.
p_down(pacman, pdfsearch, sos, dir = "dirpkgs", repos = "https://cran.univ-paris1.fr")

## Download the documentation in subdirectories named after the keywords.
(lst <- s_crandb_list(thermodynamic, "chemical reaction"))
p_down(lst, dir = "dirpkgslist", repos = "https://cran.univ-paris1.fr")
```

---

p\_html

*HTML Help Page, PDF Manual and Vignettes*


---

## Description

p\_page opens the default browser, connects to your local CRAN and displays the home page of the package(s). An internet connexion is required.

p\_html and p\_html2 open the default browser and display the html help page of the package, if it is installed. On Windows, p\_html returns a local server address *http://127.0.0.1:\*:html* and subfunctions listed in the page can be explored whereas p\_html2 returns a file address *file:///C:/:\*:html* with no links to the subfunctions.

p\_htmlweb opens the default browser and displays the html help page of the package housed by the University of Pennsylvania. An internet connexion is required.

p\_pdf displays in a pdf reader the pdf manual of the package, or generates it on the fly in the current directory if the package is installed. Miktex or Texlive is required. This is a very fast function if the files already exist (and `overwrite=FALSE`) and a (relatively) slow function if the files needs to be generated, usually much slower than:

p\_pdfweb downloads from you local CRAN the pdf manual of the package, saves it in the current directory and opens it in the pdf application of your browser. An internet connexion is required.

p\_vig is a wrapper of `utils::browseVignettes`. It opens the default browser and displays a list of the vignettes related to a package, if they exist.

p\_vig\_all wraps `utils::browseVignettes(NULL)`. It opens the default browser and displays all vignettes available in the computer. This can be a very large html file.

p\_check opens the default browser, connects to your local CRAN and displays the CRAN Package Check Results page for the package(s). An internet connexion is required.

p\_archive opens the default browser and displays the package archives. An internet connexion is required.

## Usage

```
p_page(..., char = NULL, repos = getOption("repos")[1])
```

```
p_html(..., char = NULL)
```

```
p_html2(..., char = NULL)
```



```

p_htmlweb(..., char = NULL)

p_pdf(..., char = NULL, overwrite = FALSE, dir = ".")

p_pdfweb(..., char = NULL, repos = getOption("repos")[1])

p_vig(..., char = NULL)

p_vig_all()

p_check(..., char = NULL, repos = getOption("repos")[1])

p_archive(..., char = NULL)

```

### Arguments

...	any format recognized by <a href="#">cns</a> , except list. A vector of packages.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
repos	character. The address of your local CRAN.
overwrite	logical. Overwrite already existing file (and use LaTeX intensively).
dir	character. The directory in which the files are read or written. Default value "." is the current directory.

### Examples

```

p_page(pacman, sos, repos = "https://cran.univ-paris1.fr")
p_html(pacman, sos)
p_htmlweb(pacman)
p_check(pacman, repos = "https://cran.univ-paris1.fr")
p_archive(pacman)
p_vig(pacman)
p_pdfweb(sos, repos = "https://cran.univ-paris1.fr")
p_pdf(sos, dir = file.path(tempdir(), "ppdf"))

```

---

p\_inun

*List of Installed, Uninstalled and Non-Existing Packages*

---

### Description

p\_incrandb returns TRUE if all packages are listed in crandb and a vector of FALSE with the names of the packages not listed in crandb.

p\_inun returns a list of packages installed or not installed in the computer.

In addition, p\_inun\_crandb checks if the packages exist or do not exist in crandb. This latest information reminds you about all your private unpublished packages.

**Usage**

```
p_incrandb(..., char = NULL, crandb = get("crandb", envir =
  .GlobalEnv))
```

```
p_inun(..., char = NULL)
```

```
p_inun_crandb(..., char = NULL, crandb = get("crandb", envir =
  .GlobalEnv))
```

**Arguments**

...	any format recognized by <code>cnsc</code> , including list. A vector or a list of packages.
char	(name to) a character vector or a list. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
crandb	data.frame crandb.

**Examples**

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

## Check if packages are installed or not, and exist or not in crandb
p_incrandb(RWsearch, NotAPkg1, pacman, NotAPkg2, sos)
p_inun(RWsearch, NotAPkg1, pacman, NotAPkg2, sos)
p_inun_crandb(RWsearch, NotAPkg1, pacman, NotAPkg2, sos)
```

---

p\_table2pdf

*Package Information in Console and PDF Files*


---

**Description**

p\_table returns a subset of crandb for the given packages and the selected columns, by default the Package name, the Title and the Description.

p\_table2 has a preset value to 2 columns: "Package", "Title" and prints the results in the console with a left alignment.

p\_table5 has a preset value to 5 columns: "Package", "Title", "Description", "Author", "Maintainer".

p\_table7 has a preset value to 7 columns: "Package", "Version", "Published", "Title", "Description", "Author", "Maintainer".

table\_pdf prints the results of p\_table, p\_table5 or p\_table7 in pdf file(s). Miktex or Texlive is required.

p\_table2pdf, p\_table3pdf, p\_table5pdf, p\_table7pdf combine the above functions.

**Usage**

```

p_table(..., char = NULL, columns = c("Package", "Title",
  "Description"), crandb = get("crandb", envir = .GlobalEnv))

p_table2(..., char = NULL, crandb = get("crandb", envir = .GlobalEnv))

p_table5(..., char = NULL, crandb = get("crandb", envir = .GlobalEnv))

p_table7(..., char = NULL, crandb = get("crandb", envir = .GlobalEnv))

table_pdf(x, filename = "SelectedPkgs.tex", dir = ".",
  texops = "a4paper,landscape,10pt", pdf = TRUE, cleantex = TRUE,
  openpdf = TRUE, verbose = TRUE)

p_table2pdf(..., char = NULL, filename = "Selectedpkgs", dir = ".",
  texops = "a4paper,landscape,10pt", pdf = TRUE, cleantex = TRUE,
  openpdf = TRUE, verbose = TRUE, crandb = get("crandb", envir =
  .GlobalEnv))

p_table3pdf(..., char = NULL, filename = "Selectedpkgs", dir = ".",
  texops = "a4paper,landscape,10pt", pdf = TRUE, cleantex = TRUE,
  openpdf = TRUE, verbose = TRUE, crandb = get("crandb", envir =
  .GlobalEnv))

p_table5pdf(..., char = NULL, filename = "Selectedpkgs", dir = ".",
  texops = "a4paper,landscape,10pt", pdf = TRUE, cleantex = TRUE,
  openpdf = TRUE, verbose = TRUE, crandb = get("crandb", envir =
  .GlobalEnv))

p_table7pdf(..., char = NULL, filename = "Selectedpkgs", dir = ".",
  texops = "a4paper,landscape,10pt", pdf = TRUE, cleantex = TRUE,
  openpdf = TRUE, verbose = TRUE, crandb = get("crandb", envir =
  .GlobalEnv))

```

**Arguments**

...	any format recognized by <a href="#">cns</a> , including list. A vector or a list of packages.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
columns	character vector. A sub-vector of <code>colnames(crandb)</code> . The short form "P", "T", "D", "PT", "PD", "TD", "PTD", "A", "M", "AM" describing the Package name, Title, Description, Author, Maintainer or a combination of them is accepted.
crandb	data.frame crandb. The data.frame of CRAN packages.
x	(list of) data.frame produced by <code>p_table</code> (with 3 columns), <code>p_table5</code> (5 columns) or <code>p_table7</code> (7 columns). If x is a list, the names of the list will be appended to filename.
filename	character. The file name (with or without extension).

dir	character. The directory in which the files are read or written. Default value "." is the current directory.
texops	character vector. Options passed to instruction documentclass in *.tex file.
pdf	logical. FALSE generates the *.tex file. TRUE generates both the *.tex and *.pdf files.
cleantex	logical. Remove the .tex file(s) (only if pdf = TRUE).
openpdf	logical. Open the generated *.pdf file(s) in a pdf viewer (only if pdf = TRUE).
verbose	logical. Print the path(s) to the generated file(s).

### Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

## Use a large console (useful for p_table2())
p_table2(pacman, pdfsearch, sos)
(lst <- s_crandb_list("thermodynamic", "chemical reaction", "distillation"))
p_table2(lst)

## print the table(s) in pdf file(s) and open it (them) in a pdf viewer.
p_table5pdf(pacman, pdfsearch, sos, dir = file.path(tempdir(), "ptable"))
p_table7pdf(lst, dir = file.path(tempdir(), "ptable"), cleantex = FALSE, openpdf = TRUE)
```

---

p\_text2pdf

*Download Package Documentation in Text Files*

---

### Description

p\_text extracts from CRAN the most relevant information related to one or several packages and print them in a text file which can be tailored to various formats: \*.txt, \*.md, \*.tex for further treatment.

p\_text2md has preset values for markdown files.

p\_text2tex has preset values for latex files.

p\_text2pdf has preset values for pdf files.

### Usage

```
p_text(..., char = NULL, filename = "txtpkgs.txt", dir = ".",
  beforetext = "", f_maintext = funmaintext, sep1 = "==" ,
  sep2 = " ==", eol = "\n", README = TRUE, NEWS = TRUE,
  vignettes = TRUE, aftertext = "", editor = FALSE, pager = FALSE,
  verbose = TRUE, crandb = get("crandb", envir = .GlobalEnv),
```

```

    repos = getOption("repos")[1])

p_text2md(..., char = NULL, filename = "mdpks.md", dir = ".",
  beforetext = funheadermd(), f_maintext = funmaintext, sep1 = "# ",
  sep2 = " ", eol = " \n", README = TRUE, NEWS = TRUE,
  vignettes = TRUE, aftertext = "", editor = FALSE, pager = FALSE,
  verbose = TRUE, crandb = get("crandb", envir = .GlobalEnv),
  repos = getOption("repos")[1])

p_text2tex(..., char = NULL, filename = "texpks.tex", dir = ".",
  beforetext = funheadertex(), f_maintext = funmaintext,
  sep1 = "\\section{", sep2 = "}", eol = " \\\\n",
  README = TRUE, NEWS = TRUE, vignettes = TRUE,
  aftertext = funfootertex(), editor = FALSE, pager = FALSE,
  verbose = TRUE, crandb = get("crandb", envir = .GlobalEnv),
  repos = getOption("repos")[1])

p_text2pdf(..., char = NULL, filename = "pdfpks.pdf", dir = ".",
  beforetext = funheadertex(), f_maintext = funmaintext,
  sep1 = "\\section{", sep2 = "}", eol = " \\\\n",
  README = TRUE, NEWS = TRUE, vignettes = TRUE,
  aftertext = funfootertex(), cleantex = TRUE, openpdf = FALSE,
  verbose = TRUE, crandb = get("crandb", envir = .GlobalEnv),
  repos = getOption("repos")[1])

```

## Arguments

...	any format recognized by <a href="#">cns</a> , including list. A vector or packages or a named list of packages (with names being the keywords).
char	(name to) a character vector or a list. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
filename	character. The file name with extension. If ... (or ...) is a list, the names of the list will be appended to filename.
dir	character. The directory in which the files are read or written. Default value "." is the current directory.
beforetext	character. The text written at the beginning of the file.
f_maintext	function name. The function used to extract the main text from crandb (supplied with no parenthesis).
sep1	character. The symbols written just before each package name.
sep2	character. The symbols written just after each package name. If used with with markdown, add two blank characters at the end to force a new line.
eol	character. The end of line for the main text (but not for the header and the footer). "\n" for text, "\n" for rmarkdown, "\\n" for latex.
README	logical. Write the line related to the README page, if it exists.
NEWS	logical. Write the line related to the NEWS page, if it exists.
vignettes	logical. Write the lines related to the vignette(s), if they exist.

aftertext	character. The text written at the end of the file.
editor	logical. Open the text file with editor.
pager	logical. Open the text file with pager.
verbose	logical. List the generated file(s).
crandb	data.frame crandb. The data.frame of CRAN packages.
repos	character. The address of your local CRAN.
cleantex	logical. Remove the .tex file(s).
openpdf	logical. Open the pdf files in the default pdf viewer.

### Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

## Search in crandb
vec <- s_crandb(search, find, select = "PT")
lst <- s_crandb_list(thermodynamic, "chemical reaction")

## Generate a txt file
p_text(vec, filename = "SearchFind.txt", dir = file.path(tempdir(), "ptext"),
       repos = "https://cran.univ-paris1.fr")

## Generate a markdown file (and use rmarkdown::render() for further treatment)
## In sep1, replace the '=' sign by the sharp sign (rejected by R CMD check)
p_text(char = lst, filename = "Chemistry.md", beforetext = funheadermd(),
       f_maintext = funmaintext, sep1 = "= ", sep2 = " ",
       dir = file.path(tempdir(), "ptext"), repos = "https://cran.univ-paris1.fr")

## Generate 4 tex + 4 pdf files (40-60 seconds)
p_text2pdf(lst, dir = file.path(tempdir(), "ptext"), cleantex = FALSE,
          openpdf = FALSE, repos = "https://cran.univ-paris1.fr")
```

---

s\_crandb

*Search Packages by Keywords in data.frame crandb*


---

### Description

The most important functions in this package along with [p\\_down](#).

Search packages in data.frame crandb that contain one or several keywords in the columns "Package", "Title", "Description", "Author" or "Maintainer".

s\_crandb returns a vector of the packages that contain the keywords.

s\_crandb\_list returns a list where each element of the list is one of the keywords.

s\_crandb\_PTD returns a list split by results in columns "Package", "Title" and "Description". Option mode = "and", "relax" is ignored.

s\_crandb\_AM returns a list split by results in columns "Author" and "Maintainer". Option mode = "and", "relax" is ignored.

Use [p\\_table2](#) to print the results of s\_crandb and s\_crandb\_list in the console. Use [p\\_text](#) to send the results in txt, md or pdf files. Use [p\\_display](#) to visualize the results in html pages in the browser.

### Usage

```
s_crandb(..., char = NULL, select = "PTD", mode = "or",
  sensitive = FALSE, perl = FALSE, fixed = FALSE, agrep = FALSE,
  max.distance = 0.1, costs = NULL, crandb = get("crandb", envir =
  .GlobalEnv))
```

```
s_crandb_list(..., char = NULL, select = "PTD", mode = "or",
  sensitive = FALSE, perl = FALSE, fixed = FALSE, agrep = FALSE,
  max.distance = 0.1, costs = NULL, crandb = get("crandb", envir =
  .GlobalEnv))
```

```
s_crandb_PTD(..., char = NULL, mode = "or", sensitive = FALSE,
  perl = FALSE, fixed = FALSE, agrep = FALSE, max.distance = 0.1,
  costs = NULL, crandb = get("crandb", envir = .GlobalEnv))
```

```
s_crandb_AM(..., char = NULL, mode = "or", sensitive = FALSE,
  perl = FALSE, fixed = FALSE, agrep = FALSE, max.distance = 0.1,
  costs = NULL, crandb = get("crandb", envir = .GlobalEnv))
```

### Arguments

...	any format recognized by <a href="#">cns</a> , except list. One or several keywords.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
select	character vector. A sub-vector of colnames(crandb). The short form "P", "T", "D", "PT", "PD", "TD", "PTD", "A", "M", "AM" describing the Package name, Title, Description, Author, Maintainer or a combination of them is accepted.
mode	character among "or", "and", "relax". The search mode. "relax" is for 3 words and more. It is an intermediate between "or" and "and" as it requires just 2 matching words: ("word1" AND "word2") OR ("word1" AND "word3") OR ("word1" AND "word3").
sensitive	logical. TRUE forces the search to be case sensitive.
perl	logical. Used only if fixed = FALSE. TRUE uses Perl-compatible regex. FALSE uses default regexps.
fixed	logical. TRUE matches the keywords as is (and sensitive is forced to TRUE). FALSE allows grep or Perl regexps. See <a href="#">grep</a> . Not used by agrep.
agrep	logical. For approximate matching, use <a href="#">agrep</a> function rather than grep.

max.distance    integer or numeric. See [agrep](#).  
 costs            NULL or list. See [agrep](#).  
 crandb           data.frame crandb.

### Examples

```
## In real life, download crandb from CRAN or load it from your directory
## with functions crandb_down() or crandb_load().
## In this example, we use a small file.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))

## Search using standard or non-standard content
s_crandb(c("thermodynamic", "chemical reaction", "distillation"))
s_crandb_list(thermodynamic, "chemical reaction", distillation)

## Search using the various options
s_crandb(pH, sensitive = TRUE)
s_crandb_PTD(pH, sensitive = TRUE)
s_crandb_PTD("C++", fixed = TRUE)
s_crandb(search, find, cran, web, select = "PD", mode = "and")
s_crandb(search, find, cran, web, select = "PD", mode = "relax")
s_crandb(search, find, cran, web, select = "PD", mode = "or")

## Search for some authors using the various options
s_crandb_AM(Kiener, Dutang, ORPHANED)

## Non-standard content can be unquoted words or objects in .GlobalEnv
## They are transformed into character or are evaluated
## Here, the searched keywords are "find" and "search".
OTHER <- "search"
(lst <- s_crandb_list(find, OTHER, select = "P", sensitive = TRUE))

## Display in the browser this list of packages
p_display5(lst, dir = tempdir())
```

---

s\_crandb\_tvdb

*Search For Recent Packages In crandb And In Task View*

---

### Description

This is a function for task view maintenance.

Search packages in a subset of crandb within dates from and to that contain one or several keywords in the columns "Package", "Title", "Description", "Author" or "Maintainer", then verify if these packages are already refereed in one of the task views stored in tvdb.



**Usage**

```
s_crandb_tvdb(..., char = NULL, tv = "Distributions", from = -10,
  to = Sys.Date(), select = "PTD", mode = "or", sensitive = FALSE,
  perl = FALSE, fixed = FALSE, agrep = FALSE, max.distance = 0.1,
  costs = NULL, crandb = get("crandb", envir = .GlobalEnv),
  tvdb = get("tvdb", envir = .GlobalEnv))
```

**Arguments**

...	any format recognized by <a href="#">cns</a> , except list. One or several keywords.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
tv	character. One task view among those listed in tvdb.
from	character representing a date earlier than date to. Or a negative integer representing the number of days preceding the date to.
to	date. The upper date in the search.
select	character vector. A sub-vector of colnames(crandb). The short form "P", "T", "D", "PT", "PD", "TD", "PTD", "A", "M", "AM" describing the Package name, Title, Description, Author, Maintainer or a combination of them is accepted.
mode	character among "or", "and", "relax". The search mode. "relax" is for 3 words and more. It is an intermediate between "or" and "and" as it requires just 2 matching words: ("word1" AND "word2") OR ("word1" AND "word3") OR ("word1" AND "word3").
sensitive	logical. TRUE forces the search to be case sensitive.
perl	logical. Used only if fixed = FALSE. TRUE uses Perl-compatible regex. FALSE uses default regexps.
fixed	logical. TRUE matches the keywords as is (and sensitive is forced to TRUE). FALSE allows grep or Perl regexps. See <a href="#">grep</a> . Not used by <a href="#">agrep</a> .
agrep	logical. For approximate matching, use <a href="#">agrep</a> function rather than <a href="#">grep</a> .
max.distance	integer or numeric. See <a href="#">agrep</a> .
costs	NULL or list. See <a href="#">agrep</a> .
crandb	data.frame crandb.
tvdb	list. The list of the task views.

**Value**

A list with the following vectors:

- spkgs: The selected packages that contain the keyword(s).
- inTV: The packages that contain the keyword(s) already refereed in the task view.
- notinTV: The packages that contain the keyword(s) not (yet) refereed in the task view.
- inTV\_in: Among the packages available in the task view, those installed in the computer.
- inTV\_un: Among the packages available in the task view, those not installed in the computer.

- `notinTV_in`: Among the packages not refereed in the task view, those installed in the computer.
- `notinTV_un`: Among the packages not refereed in the task view, those not installed in the computer.

### Examples

```
### TASK VIEW MAINTENANCE (tvdb + crandb)
## In real life, download crandb and tvdb from CRAN or load them from your directory
## with functions crandb_down(), crandb_load(), tvdb_down(), tvdb_load().
## In this example, we use small files.
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
tvdb_load(system.file("data", "ztvdb.rda", package = "RWsearch"))

## List the task views
tvdb_vec()

## Search for the recent packages in crandb that contain the keyword
## and verify the packages already refereed in the task view.
## With ztvdb.rda, from = "2018-01-01" is selected.
(lst <- s_crandb_tvdb("distribution", tv = "Distributions", from = "2018-01-01"))

## Display in the browser some information about the selected packages.
p_display7(lst[c("inTV", "notinTV")], dir = tempdir())
```

---

s\_hs

*Search Packages and Functions in Installed Packages*

---

### Description

`s_hs` is a wrapper of the well known function `??` and its parent function `help.search`. Visit the help page [help.search](#) for details on the various arguments.

### Usage

```
s_hs(..., char = NULL, fields = c("alias", "concept", "title"),
      apropos, keyword, whatis, ignore.case = TRUE, package = NULL,
      agrep = NULL, use_UTF8 = FALSE)
```

### Arguments

<code>...</code>	one single character string recognized by <a href="#">cnsc</a> . One and only one pattern
<code>char</code>	(name to) a single character string. Use this argument if <code>...</code> fails or if you call the function from another function. If used, argument <code>...</code> is ignored.
<code>fields</code>	See <code>help.search</code> .
<code>apropos</code>	See <code>help.search</code> .

keyword	See help.search.
whatis	See help.search.
ignore.case	See help.search.
package	See help.search.
agrep	See help.search.
use_UTF8	See help.search.

## Examples

```
s_hs("neural network")
```

---

s_sos	<i>Search Packages and Functions in U. of Pennsylvania and RDocumentation</i>
-------	---

---

## Description

s\_sos searches in all R documentation packages and functions that contain one or several keywords, open the default browser and display the results in a html page. For one or two keywords, s\_sos may find more results than [s\\_crandb](#) as it goes deeper in the documentation, down to the function level. An internet connection is required to reach the website maintained by the University of Pennsylvania.

s\_sos is a minimal wrapper of the function *sos::findFn*. Use directly the package *sos* and read its vignette for advanced search options.

## Usage

```
s_sos(..., char = NULL)
```

## Arguments

...	any format recognized by <a href="#">cnsc</a> , except list. One or several keywords.
char	(name to) a character vector. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.

## See Also

<http://finzi.psych.upenn.edu>,

<https://CRAN.R-project.org/package=sos> (index and vignette).

**Examples**

```
## Search using standard or non-standard content
## and display the results in a browser.
s_sos(distillation)

res <- s_sos("chemical reaction")
as.data.frame(res)
res
```

---

s\_tvdb

*Search Packages in Task Views*


---

**Description**

s\_tvdb searches if one or several package(s) are referred in some task views and lists these task views.

**Usage**

```
s_tvdb(..., char = NULL, tvdb = get("tvdb", envir = .GlobalEnv))
```

**Arguments**

...	any format recognized by <a href="#">cnsc</a> , except list. The names of one or several task views.
char	(name to) a character vector or a list. Use this argument if ... fails or if you call the function from another function. If used, argument ... is ignored.
tvdb	list. The list of the task views.

**Examples**

```
## In real life, download tvdb from CRAN or load it from your directory
## with functions tvdb_down() or tvdb_load().
## In this example, we use a small file.
tvdb_load(system.file("data", "ztvdb.rda", package = "RWsearch"))
s_tvdb(actuar, FatTailsR, MASS, zoo, nopackage)
```

---

tvdb	<i>Task Views (tvdb.rda)</i>
------	------------------------------

---

## Description

`tvdb_down` downloads from CRAN the file "Views.rds", a file refreshed every day that describes the task views available in CRAN for this day, rearranges the list in an alphabetical order and gives names to the list names, then loads in `.GlobalEnv` this list (of class `ctvlist`) under the name `tvdb` and saves it with the filename `tvdb.rda`.

`tvdb_load` loads the file `filename` in `.GlobalEnv` under the name `tvdb`. Equivalent to `load("tvdb.rda")`.

`tvdb_vec` displays the list of the task views. There are 36 task views in August 2018.

`tvdb_dfr` extracts from `tvdb` a data.frame *version, name, topic* of the task views.

`tvdb_list` extracts from `tvdb` the list of the task views and the referenced packages.

`tvdb_pkgs` displays the packages referenced by one or several task views.

Visit [s\\_crandb\\_tvdb](#) to conduct task view maintenance.

## Usage

```
tvdb_down(dir = ".", repos = getOption("repos")[1])

tvdb_load(filename = "tvdb.rda")

tvdb_vec(tvdb = get("tvdb", envir = .GlobalEnv))

tvdb_dfr(tvdb = get("tvdb", envir = .GlobalEnv))

tvdb_list(tvdb = get("tvdb", envir = .GlobalEnv))

tvdb_pkgs(..., char = NULL, tvdb = get("tvdb", envir = .GlobalEnv))
```

## Arguments

<code>dir</code>	character. The directory where "tvdb.rda" is saved. Default value "." is the current directory.
<code>repos</code>	character. The address of your local CRAN.
<code>filename</code>	character. The path to file "tvdb.rda". The default is to read in the current directory.
<code>tvdb</code>	list. The list of the task views.
<code>...</code>	any format recognized by <a href="#">cnsc</a> , except list. The names of one or several task views.
<code>char</code>	(name to) a character vector or a list. Use this argument if <code>...</code> fails or if you call the function from another function. If used, argument <code>...</code> is ignored.

## Examples

```
### DOWNLOAD AND VISUALIZE THE TASK VIEWS (tvdb)
## In real life, download tvdb from CRAN or load it from your directory
## with functions tvdb_down() or tvdb_load().
## In this example, we use a small file.
tvdb_load(system.file("data", "ztvdb.rda", package = "RWsearch"))
length(tvdb)

## List the task views
tvdb_vec()
tvdb_dfr()
tvdb_list()[1:3]
tvdb_pkgs("Distributions")
lst <- tvdb_pkgs(gR, Genetics) ; lst

## Display in the browser the referred packages.
## A full version of crandb is required to fetch all listed packages.
## Run crandb_down() (5-20 seconds) if it is not already in .GlobalEnv.
crandb_down(dir = tempdir(), repos = "https://cran.univ-paris1.fr")
ls()
p_display7(lst, dir = file.path(tempdir(), "pdisp"), verbose = TRUE)
```

---

zcrandb

*File zcrandb.rda: A Subset of crandb Dataset*


---

## Description

File *zcrandb.rda* loads in `.Globalenv` as `crandb`, a data.frame of dim 50 x 65. It contains 50 packages that match the keywords used in the examples of this package.

File *zcrandb.rda*, 22 ko, acts as a replacement of the original but large file *crandb.rda* to be downloaded from CRAN. The weight of *crandb.rda* was 4.3 Mo with 13001 packages on August 31, 2018 and 6.7 Mo with 13902 packages on March 17, 2019 . The use of *zcrandb.rda* avoids inappropriate connections to CRAN and increases the speed in the examples.

## Examples

```
crandb_load(system.file("data", "zcrandb.rda", package = "RWsearch"))
```

---

ztvdb

*File ztvdb.rda: A Subset of tvdb Dataset*


---

## Description

File *ztvdb.rda* is a small file of 5 ko that contains 6 task views and acts as a replacement of the large file *tvdb.rda* downloaded from CRAN that contains 36 task views. It loads in `.GlobalEnv` as `tvdb`. The use of *ztvdb.rda* avoids inappropriate connections to CRAN and increases the speed in the examples.

**Examples**

```
ztdb_load(system.file("data", "ztdb.rda", package = "RWsearch"))
```

# Index

## \*Topic **datasets**

- zcrandb, 38
- ztvdb, 38
  
- agrep, 31–33
- archivedb, 4
- archivedb\_down (archivedb), 4
- archivedb\_list (archivedb), 4
- archivedb\_load (archivedb), 4
- archivedb\_npkgs (archivedb), 4
- archivedb\_pkgs (archivedb), 4
- archivedb\_rempkgs (archivedb), 4
- args, 11
  
- cncs, 6, 6, 9, 11, 12, 16, 18, 20, 22, 23, 25–27, 29, 31, 33–37
- cncsinfun (cncs), 6
- cncsinfun2 (cncs), 6
- crandb, 7
- crandb\_comp (crandb), 7
- crandb\_down (crandb), 7
- crandb\_fromto (crandb), 7
- crandb\_load (crandb), 7
- crandb\_pkgs (crandb), 7
  
- e\_check, 9
  
- f\_args, 11
- f\_pdf, 12
- f\_sig (f\_args), 11
- funfootertex (funmaintext), 10
- funheadermd (funmaintext), 10
- funheadertex (funmaintext), 10
- funmaintex (funmaintext), 10
- funmaintext, 10
  
- grep, 31, 33
  
- h\_1bis (h\_engine), 13
- h\_abcbourse (h\_engine), 13
- h\_academie (h\_direct), 12
- h\_arxiv (h\_engine), 13
- h\_arxivpdf (h\_engine), 13
- h\_ask (h\_engine), 13
- h\_baidu (h\_engine), 13
- h\_bing (h\_engine), 13
- h\_blackle (h\_engine), 13
- h\_bmap (h\_engine), 13
- h\_boursorama (h\_engine), 13
- h\_cnrtl (h\_engine), 13
- h\_cran (h\_R), 17
- h\_cranberries (h\_R), 17
- h\_cranbydate (h\_R), 17
- h\_cranbyname (h\_R), 17
- h\_cranstatus (h\_R), 17
- h\_crantv (h\_R), 17
- h\_crossref (h\_engine), 13
- h\_daum (h\_engine), 13
- h\_ddg (h\_engine), 13
- h\_direct, 12
- h\_dm (h\_engine), 13
- h\_ecosia (h\_engine), 13
- h\_egerin (h\_engine), 13
- h\_engine, 13
- h\_estrep (h\_engine), 13
- h\_evene (h\_engine), 13
- h\_exalead (h\_engine), 13
- h\_excite (h\_engine), 13
- h\_framabee (h\_engine), 13
- h\_framasoft (h\_direct), 12
- h\_framasoft0 (h\_direct), 12
- h\_gepuro (h\_R), 17
- h\_gigablast (h\_engine), 13
- h\_github (h\_engine), 13
- h\_gitlab (h\_engine), 13
- h\_gmap (h\_engine), 13
- h\_google (h\_engine), 13
- h\_gscolar (h\_engine), 13
- h\_info (h\_engine), 13
- h\_ixquick (h\_engine), 13



- h\_khoj (h\_engine), 13
- h\_lesechos (h\_engine), 13
- h\_lexilogos (h\_direct), 12
- h\_lilo (h\_engine), 13
- h\_linguee (h\_direct), 12
- h\_lt (h\_engine), 13
- h\_lycos (h\_engine), 13
- h\_mappy (h\_engine), 13
- h\_mw (h\_engine), 13
- h\_nabble (h\_R), 17
- h\_nate (h\_engine), 13
- h\_naver (h\_engine), 13
- h\_orcid (h\_engine), 13
- h\_osm (h\_engine), 13
- h\_osmn (h\_engine), 13
- h\_parsijoo (h\_engine), 13
- h\_peertube (h\_engine), 13
- h\_peru (h\_engine), 13
- h\_pipilika (h\_engine), 13
- h\_qwant (h\_engine), 13
- h\_qwfr (h\_engine), 13
- h\_R, 17
- h\_rbloggers (h\_R), 17
- h\_rdoc (h\_R), 17
- h\_rdoctv (h\_R), 17
- h\_rdr (h\_R), 17
- h\_reverso (h\_direct), 12
- h\_reverso\_d (h\_engine), 13
- h\_rseek (h\_R), 17
- h\_sapo (h\_engine), 13
- h\_searx (h\_engine), 13
- h\_so (h\_engine), 13
- h\_sogou (h\_engine), 13
- h\_ssrn (h\_engine), 13
- h\_ssrnauth (h\_engine), 13
- h\_startpage (h\_engine), 13
- h\_tad (h\_direct), 12
- h\_tadsm (h\_direct), 12
- h\_ttp, 18
- h\_twfr (h\_engine), 13
- h\_twitter (h\_engine), 13
- h\_un (h\_engine), 13
- h\_verbes (h\_engine), 13
- h\_via (h\_engine), 13
- h\_vimeo (h\_engine), 13
- h\_wego (h\_engine), 13
- h\_wp (h\_engine), 13
- h\_wpfr (h\_engine), 13
- h\_yacy (h\_direct), 12
- h\_yahoo (h\_engine), 13
- h\_yahoofin (h\_engine), 13
- h\_yandex (h\_engine), 13
- h\_yooz (h\_engine), 13
- h\_yt (h\_engine), 13
- help.search, 34
- n\_graph, 19
- n\_graphF, 20
- n\_graphF (n\_graph), 19
- n\_graphS, 20
- n\_graphS (n\_graph), 19
- p\_archive (p\_html), 24
- p\_check (p\_html), 24
- p\_deps, 20
- p\_depsrev (p\_deps), 20
- p\_display, 21, 31
- p\_display5 (p\_display), 21
- p\_display7 (p\_display), 21
- p\_down, 22, 30
- p\_down0 (p\_down), 22
- p\_downarch, 23
- p\_downarch (archivedb), 4
- p\_html, 24
- p\_html2 (p\_html), 24
- p\_htmlweb (p\_html), 24
- p\_incrandb (p\_inun), 25
- p\_inun, 25
- p\_inun\_crandb (p\_inun), 25
- p\_network, 19
- p\_network (p\_deps), 20
- p\_page (p\_html), 24
- p\_pdf (p\_html), 24
- p\_pdfweb (p\_html), 24
- p\_sig (f\_args), 11
- p\_table (p\_table2pdf), 26
- p\_table2, 31
- p\_table2 (p\_table2pdf), 26
- p\_table2pdf, 26
- p\_table3pdf (p\_table2pdf), 26
- p\_table5 (p\_table2pdf), 26
- p\_table5pdf (p\_table2pdf), 26
- p\_table7 (p\_table2pdf), 26
- p\_table7pdf (p\_table2pdf), 26
- p\_text, 31
- p\_text (p\_text2pdf), 28
- p\_text2md (p\_text2pdf), 28

`p_text2pdf`, 28  
`p_text2tex` (`p_text2pdf`), 28  
`p_vig` (`p_html`), 24  
`p_vig_all` (`p_html`), 24

RWsearch (RWsearch-package), 2  
RWsearch-package, 2

`s_crandb`, 22, 30, 35  
`s_crandb_AM` (`s_crandb`), 30  
`s_crandb_list`, 22  
`s_crandb_list` (`s_crandb`), 30  
`s_crandb_PTD` (`s_crandb`), 30  
`s_crandb_tvdb`, 32, 37  
`s_hs`, 34  
`s_sos`, 35  
`s_tvdb`, 36

`table_pdf` (`p_table2pdf`), 26  
`tvdb`, 37  
`tvdb_dfr` (`tvdb`), 37  
`tvdb_down` (`tvdb`), 37  
`tvdb_list` (`tvdb`), 37  
`tvdb_load` (`tvdb`), 37  
`tvdb_pkgs` (`tvdb`), 37  
`tvdb_vec` (`tvdb`), 37

`zcrandb`, 38  
`ztvdb`, 38