

# Package ‘rscorecard’

May 9, 2026

**Title** A Method to Download Department of Education College Scorecard Data

**Version** 0.33.0

**Maintainer** Benjamin Skinner <ben@btskinner.io>

**URL** <https://www.btskinner.io/rscorecard/>

**BugReports** <https://github.com/btskinner/rscorecard/issues>

**Description** A method to download Department of Education College Scorecard data using the public API <<https://collegescorecard.ed.gov/data/data-documentation/>>. It is based on the 'dplyr' model of piped commands to select and filter data in a single chained function call. An API key from the U.S. Department of Education is required.

**Depends** R (>= 4.1.0)

**License** MIT + file LICENSE

**Imports** dplyr, httr, jsonlite, lazyeval, tibble, tidyselect, tidyr, purrr

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**Suggests** knitr, rmarkdown, testthat

**NeedsCompilation** no

**Author** Benjamin Skinner [aut, cre] (ORCID: <<https://orcid.org/0000-0002-0337-7415>>)

**Repository** CRAN

**Date/Publication** 2026-03-31 18:10:02 UTC

## Contents

sc_dict . . . . .	2
sc_filter . . . . .	3
sc_get . . . . .	4

sc_init . . . . .	5
sc_key . . . . .	6
sc_select . . . . .	6
sc_year . . . . .	7
sc_zip . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

sc_dict	<i>Search data dictionary.</i>
---------	--------------------------------

---

## Description

This function is used to search the College Scorecard data dictionary.

## Usage

```
sc_dict(
  search_string,
  search_col = c("all", "description", "varname", "dev_friendly_name", "dev_category",
    "label", "source"),
  ignore_case = TRUE,
  limit = 10,
  confirm = FALSE,
  print_dev = FALSE,
  print_notes = FALSE,
  return_df = FALSE,
  print_off = FALSE,
  can_filter = FALSE,
  filter_vars = FALSE
)
```

## Arguments

search_string	Character string for search. Can use regular expression for search. Must escape special characters, . \   ( ) [ { ^ \$ * + ?, with a doublebackslash \\.
search_col	Column to search. The default is to search all columns. Other options include: "varname", "dev_friendly_name", "dev_category", "label".
ignore_case	Search is case insensitive by default. Change to FALSE to restrict search to exact case matches.
limit	Only the first 10 dictionary items are returned by default. Increase to return more values. Set to Inf to return all items matched in search'
confirm	Use to confirm status of variable name in dictionary. Returns TRUE or FALSE.
print_dev	Set to TRUE if you want to see the developer friendly name and category used in the API call.
print_notes	Set to TRUE if you want to see the notes included in the data dictionary (if any).

return_df	Return a tibble of the subset data dictionary.
print_off	Do not print to console; useful if you only want to return a tibble of dictionary values.
can_filter	Use to confirm that a variable can be used as a filtering variable. Returns TRUE or FALSE
filter_vars	Use to print variables that can be used to filter calls. Use with argument return_df = TRUE to return a tibble of these variables in addition to console output.

### Examples

```
## simple search for 'state' in any part of the dictionary
sc_dict('state')

## variable names starting with 'st'
sc_dict('^st', search_col = 'varname')

## return full dictionary (only recommended if not printing and
## storing in object)
df <- sc_dict('.', limit = Inf, print_off = TRUE, return_df = TRUE)

## print list of variables that can be used to filter
df <- sc_dict('.', filter_vars = TRUE, return_df = TRUE)
```

---

sc_filter	<i>Filter scorecard data by variable values.</i>
-----------	--

---

### Description

This function is used to filter the downloaded scorecard data. It converts idiomatic R into the format required by the API call.

### Usage

```
sc_filter(sccall, ...)

sc_filter_(sccall, filter_string)
```

### Arguments

sccall	Current list of parameters carried forward from prior functions in the chain (ignore)
...	Expressions to evaluate
filter_string	Filter as character string or vector of filters as character strings

### Functions

- `sc_filter_()`: Standard evaluation version of `sc_filter` (`filter_string` must be a string or vector of strings when using this version)

**Examples**

```
## Not run:
sc_filter(region == 1) # New England institutions
sc_filter(stabbr == c("TN","KY")) # institutions in Tennessee and Kentucky
sc_filter(control != 3) # exclude private, for-profit institutions
sc_filter(control == c(1,2)) # same as above
sc_filter(control == 1:2) # same as above
sc_filter(stabbr == "TN", control == 1, locale == 41:43) # TN rural publics

## End(Not run)
## Not run:
sc_filter_("region == 1")
sc_filter_("control != 3")

## With internal strings, you must either use both double and single quotes
## or escape internal quotes
sc_filter_("stabbr == c('TN','KY')")
sc_filter_('stabbr == c(\'TN\',\'KY\')')

## stored in object
filters <- c("control == 1", "locale == 41:43")
sc_filter_(filters)

## End(Not run)
```

---

sc\_get

*Get scorecard data.*


---

**Description**

This function gets the College Scorecard data by compiling and converting all the previous piped output into a single URL string that is used to get the data.

**Usage**

```
sc_get(
  sccall,
  api_key,
  debug = FALSE,
  print_key_debug = FALSE,
  return_json = FALSE
)
```

**Arguments**

sccall            Current list of parameters carried forward from prior functions in the chain (ignore)

api_key	Personal API key requested from <a href="https://api.data.gov/signup">https://api.data.gov/signup</a> stored in a string. If you first set your key using <code>sc_key</code> , then you may omit this parameter. A key set here will take precedence over any set in the environment ( <code>DATAGOV_API_KEY</code> ).
debug	Set to true to print and return API call (URL string) rather than make actual request. Should only be used when debugging calls.
print_key_debug	Only used when <code>debug == TRUE</code> . Default masks the <code>api_key</code> value. Set to <code>TRUE</code> to print the full API call string with the <code>api_key</code> unmasked.
return_json	Return data in JSON format rather than as a tibble.

### Obtain a key

To obtain an API key, visit <https://api.data.gov/signup>

### Examples

```
## Not run:
sc_get("<API KEY IN STRING>")
key <- "<API KEY IN STRING>"
sc_get(key)

## End(Not run)
```

---

sc_init	<i>Initialize chained request.</i>
---------	------------------------------------

---

### Description

This function initializes the data request. It should always be the first in the series of piped functions.

### Usage

```
sc_init(dfvars = FALSE)
```

### Arguments

dfvars	Set to <code>TRUE</code> if you would rather use the developer-friendly variable names used in actual API call.
--------	---

### Examples

```
## Not run:
sc_init()
sc_init(dfvars = TRUE)

## End(Not run)
```

---

sc_key	<i>Store Data.gov API key in system environment.</i>
--------	--

---

### Description

This function stores your data.gov API key in the system environment so that you only have to load it once at the start of the session. If you set your key using `sc_key`, then you may omit `api_key` parameter in the `sc_get` function.

### Usage

```
sc_key(api_key)
```

### Arguments

api_key	Personal API key requested from <a href="https://api.data.gov/signup">https://api.data.gov/signup</a> stored in a string.
---------	---

### Obtain a key

To obtain an API key, visit <https://api.data.gov/signup>.

### Examples

```
## Not run:  
sc_key('<API KEY IN STRING>')  
  
## End(Not run)
```

---

sc_select	<i>Select scorecard data variables.</i>
-----------	---

---

### Description

This function is used to select the variables returned in the final dataset.

### Usage

```
sc_select(sccall, ...)  
  
sc_select_(sccall, vars)
```

**Arguments**

sccall	Current list of parameters carried forward from prior functions in the chain (ignore)
...	Desired variable names separated by commas (not case sensitive)
vars	Character string of variable name or vector of character string variable names

**Functions**

- `sc_select_()`: Standard evaluation version of `sc_select` (vars must be string or vector of strings when using this version)

**Examples**

```
## Not run:
sc_select(UNITID)
sc_select(UNITID, INSTNM)
sc_select(unitid, instnm)

## End(Not run)
## Not run:
sc_select_("UNITID")
sc_select_(c("UNITID", "INSTNM"))
sc_select_(c("unitid", "instnm"))

## stored in object
vars_to_pull <- c("unitid", "instnm")
sc_select(vars_to_pull)

## End(Not run)
```

---

sc_year	<i>Select scorecard data year.</i>
---------	------------------------------------

---

**Description**

This function is used to select the year of the data.

**Usage**

```
sc_year(sccall, year)
```

**Arguments**

sccall	Current list of parameters carried forward from prior functions in the chain (ignore)
year	Four-digit year or string latest for latest data.

**Important notes**

1. Not all variables have a year option.
2. At this time, only one year at a time is allowed.
3. The year selected is not necessarily the year the data were produced. It may be the year the data were collected. For data collected over split years (fall to spring), it is likely the year represents the fall data (*e.g.*, 2011 for 2011/2012 data).

Be sure to check with the College Scorecard [data documentation report](#) when choosing the year.

**Examples**

```
## Not run:
sc_year() # latest
sc_year("latest")
sc_year(2012)

## End(Not run)
```

---

sc\_zip

*Subset results to those within specified area around zip code.*

---

**Description**

Subset results to those within specified area around zip code.

**Usage**

```
sc_zip(sccall, zip, distance = 25, km = FALSE)
```

**Arguments**

sccall	Current list of parameters carried forward from prior functions in the chain (ignore)
zip	A 5-digit zipcode
distance	An integer distance in miles or kilometers
km	A boolean value set to TRUE if distance should be in kilometers (default is FALSE for miles)

**Note**

Zip codes with leading zeros (Northeast) can be called either using a string ("02111") or as a numeric (02111). R will drop the leading zero from the second version, but `sc_zip()` will add it back before the call. The shortened version without the leading zero may also be used (2111 and "2111" both become "02111"), but is not recommended for clarity.

**Examples**

```
## Not run:
sc_zip(37203)
sc_zip(37203, 50)
sc_zip(37203, 50, km = TRUE)
sc_zip("02111")           # 1. Using string
sc_zip(02111)             # 2. Dropped leading zero will be added
sc_zip(2111)              # 3. Will become "02111" (not recommended)

## End(Not run)
```

# Index

sc\_dict, [2](#)  
sc\_filter, [3](#), [3](#)  
sc\_filter\_(sc\_filter), [3](#)  
sc\_get, [4](#)  
sc\_init, [5](#)  
sc\_key, [6](#)  
sc\_select, [6](#), [7](#)  
sc\_select\_(sc\_select), [6](#)  
sc\_year, [7](#)  
sc\_zip, [8](#)