

# Package ‘rde’

July 2, 2018

**Title** Reproducible Data Embedding

**Version** 0.1.0

**Description** Allows caching of raw data directly in R code. This allows R scripts and R Notebooks to be shared and re-run on a machine without access to the original data. Cached data is encoded into an ASCII string that can be pasted into R code. When the code is run, the data is automatically loaded from the cached version if the original data file is unavailable. Works best for small datasets (a few hundred observations).

**Depends** R (>= 3.1)

**Imports** clipr

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat, knitr, lintr, rmarkdown, spelling

**VignetteBuilder** knitr

**RoxygenNote** 6.0.1

**Language** en-US

**SystemRequirements** xclip (<https://github.com/astrand/xclip>) or xsel (<http://www.vergenet.net/~conrad/software/xsel/>) for accessing the X11 clipboard

**URL** <https://github.com/kloppen/rde>

**BugReports** <https://github.com/kloppen/rde/issues>

**NeedsCompilation** no

**Author** Stefan Kloppenborg [aut, cre]

**Maintainer** Stefan Kloppenborg <[stefan@kloppenborg.ca](mailto:stefan@kloppenborg.ca)>

**Repository** CRAN

**Date/Publication** 2018-07-02 08:50:06 UTC

## R topics documented:

copy_rde_var	2
load_rde_var	3
<b>Index</b>	<b>5</b>

---

copy_rde_var	<i>Encode, compress and copy data into the clipboard</i>
--------------	--

---

### Description

copy\_rde\_var is intended to work with [load\\_rde\\_var](#). The normal workflow would use copy\_rde\_var to copy a variable to the clipboard and then paste it in to the third argument of load\_rde\_var.

### Usage

```
copy_rde_var(var, line.width = 80L, no.clipboard = FALSE)
```

### Arguments

var	the variable to copy
line.width	the desired width of lines of text (-1 for no line breaks)
no.clipboard	the default is FALSE. Indicates that you want the function to return the string that would have been copied to the clipboard without actually copying to the clipboard. This option is mainly used for testing purposes. Normal users will not use it.

### Details

The variable in the argument var is first saved using [saveRDS](#). Then the saved variable is compressed using bzip2 compression. Next, the compressed data is base 64 encoded into a character string. Next, that string is prepended with a code that indicates the version of this package. The prepended code (currently 'rde1' allows for future changes while providing backwards compatibility). Finally, the string is optionally broken up into lines of width line.width. Whitespace and line breaks are ignored by [load\\_rde\\_var](#).

On X11 systems (e.g. Linux), external software is required in order to access the clipboard. Either 'xsel' or 'xclip' is required. Installation of this software will depend on the installation that you use, but on Ubuntu/Debian, 'sudo apt-get install xsel' will probably work.

On Windows and OSX, no additional software is required.

### Value

None (or string if no.clipboard=TRUE)

## Examples

```
copy_rde_var(iris)
```

---

load_rde_var	<i>Load data in a reproducible way that allows for exchange of code</i>
--------------	---

---

## Description

load\_rde\_var attempts to execute the code in load.fcn. If that succeeds, then the return value of that code is returned by load\_rde\_var. Otherwise, the value stored in cache is returned. cache must contain an encoded copy of the value produced by the function [copy\\_rde\\_var](#). Optionally, you can force the use of the cached data by setting use.cache = TRUE.

## Usage

```
load_rde_var(use.cache = FALSE, load.fcn, cache)
```

## Arguments

use.cache	boolean to force the use of cached data
load.fcn	code to load the data from its original source
cache	a cached copy of the data

## Details

This package is intended for small datasets. A copy of the data is encoded as a string (using base64 encoding, after compressing the data) and that string is copied into your code. Even though the data is compressed, the encoded string can still be quite long. If your data is more than a few hundred observations, this package probably isn't for you.

load.fcn must contain executable R code. Unless that code is a single expression, normally it would be enclosed in a pair of braces.

cache must be a string that was originally produced by [copy\\_rde\\_var](#). See the documentation for that function for more details about the format of this string.

If the code in load.fcn fails, then a message is produced to indicate that the failure and the data encoded in cache is returned instead. This would occur if you share your code with someone who does not have access to the data that you're loading in your code.

If use.cache = TRUE, the code in load.fcn is ignored and the data is loaded from the encoded string cache. This can be useful if it takes a very long time to obtain the data and you re-run your code often.

If the value produced by the code in load.fcn does not match the value encoded in cache, then a warning is produced to indicate that there is a mismatch.

**Value**

The data, either loaded using `load.fcn`, if possible, or from cache if that fails.

**See Also**

[copy\\_rde\\_var](#)

**Examples**

```
load_rde_var(use.cache = FALSE, {
  head(iris, 3)
}, "
rde1Q1po0TFBWSZTWbG0254AAKT/5P//XAAAAQAaAwARiWc/n3YBAAAaWACYFABAA7IhKIm
lPaU3oaRqekyaDTQNPJP1MhDAaA0AAGmg0A0AaBIokNGgAAAAANMYUzuJyxRYUJWNnsC1
tgiccpLFvZTHhARK1KFQ1z25bNBCC+0pWKgEnGEzpxVaihSiTBL2j6RRFchjamGBFpBMwN
bAHwgEGosCEGYBoztHPFUVjGcDz3qu9p4cb8rVyVfHmR5S3bWxfDDnTnyJDJh0iMipionY
lfq1FwK/IvzsuBs0muZNGtpp7oWrW4upNNGDiL2E9T6iY2RFqab09/r9oiN6p/YIdV1FPP
ISLqVP4u5IpwoSFjHbc8A=
")
```

# Index

`copy_rde_var`, [2](#), [3](#), [4](#)

`load_rde_var`, [2](#), [3](#)

`saveRDS`, [2](#)