

# Package ‘datasailr’

November 26, 2020

**Type** Package

**Title** Row by Row Data Processing Tool, Using 'DataSailr' Script

**Version** 0.8.7

**Date** 2020-11-26

**Maintainer** Toshihiro Umehara <toshi@niceume.com>

**Description** A row by row data processing tool. You can write data processing code in 'DataSailr' script which is specially intended for data manipulation. The package uses 'lib-sailr' (C/C++ library) for its 'DataSailr' code parsing and its execution.

**Encoding** UTF-8

**License** GPL (>= 3)

**Biarch** true

**Imports** Rcpp (>= 0.12.18)

**Suggests** RUnit

**LinkingTo** Rcpp

**SystemRequirements** GNU make, C++11

**URL** <https://datasailr.io>

**BugReports** <https://github.com/niceume/datasailr>

**NeedsCompilation** yes

**Author** Toshihiro Umehara [aut, cre],  
Troy Hanson [cph, ctb] (uthash),  
Howard Hinnant [cph, ctb] (date.h),  
Adrian Colomitchi [cph, ctb] (date.h),  
Florian Dang [cph, ctb] (date.h),  
Paul Thompson [cph, ctb] (date.h),  
Tomasz Kamiński [cph, ctb] (date.h),  
Nemanja Trifunovic [cph, ctb] (utfcpp),  
Kim Grasman [cph, ctb] (getopt\_port),  
Jon Clayden [cph, ctb] (ore package),  
K.Kosako [cph, ctb] (onigmo author),  
K.Takata [cph, ctb] (onigmo author),

Byte [cph, ctb] (onigmo contributor),  
 KUBO Takehiro [cph, ctb] (onigmo contributor),  
 Free Software Foundation, Inc [cph],  
 X Consortium [cph]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2020-11-26 12:40:02 UTC

## R topics documented:

datasailr-package . . . . .	2
adult_total . . . . .	3
author . . . . .	4
sail . . . . .	4
test_sail . . . . .	6

**Index** 7

---

datasailr-package *Row by Row Data Processing Tool, Using 'DataSailr' Script*

---

### Description

A row by row data processing tool. You can write data manipulation code in 'DataSailr' script which is specially intended for the data manipulation. The package uses 'libsailr' (C/C++ library) for its 'DataSailr' code parsing and its execution.

### Details

The main function, `datasailr::sail()`, takes dataframe as its 1st argument and 'DataSailr' script as its 2nd argument, and returns the processing results as dataframe. This package works with 'libsailr' library, which conducts arithmetic calculations and string/character manipulations following 'Sailr' script (which is a base version of 'DataSailr' script). 'DataSailr' package passes both 'DataSailr' script and values of each row of dataframe to the 'libsailr', and 'libsailr' processes those values of each row. The processing results are stored and are returned as dataframe.

### Author(s)

Toshi Umehara, [toshi@niceume.com](mailto:toshi@niceume.com). Maintainer: Toshi Umehara <[toshi@niceume.com](mailto:toshi@niceume.com)>

### See Also

[sail](#).

**Examples**

```

library(datasailr)
data(mtcars)

code_example = 'powerful = .
  if( 145 >= hp && hp > 0 ){
    powerful = 0
  }else if( hp > 145 ){
    powerful = 1
  }else{
    powerful = 2
  }

  germany = re/^(^Merc|^Porsche|^Volvo)/
  japan = re/^(^Mazda|^Honda|^Toyota)/

  if ( _rowname_ =~ germany ) {
    country = "Germany"
  }else if( _rowname_ =~ japan ){
    country = "Japan"
  }else {
    country = "Other"
  }
  company = rexp_matched(1)
'

sail(mtcars, code_example)

```

---

adult\_total

*Census income dataset*


---

**Description**

"Census Income" dataset from UCI machine learning repository.

**Usage**

```
data(adult_total)
```

**Format**

>50K, <=50K. age: continuous. workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. fnlwgt: continuous. education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. education-num: continuous. marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty,

Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. sex: Female, Male. capital-gain: continuous. capital-loss: continuous. hours-per-week: continuous. native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holand-Netherlands.

### Source

<https://archive.ics.uci.edu/ml/datasets/Adult>

### References

<https://archive.ics.uci.edu/ml/datasets/Adult>

---

author	<i>show author of this package</i>
--------	------------------------------------

---

### Description

`datasailr::author()` shows author information of this package.

### Usage

```
author()
```

---

sail	<i>Run 'DataSailr' script and manipulate dataframe</i>
------	--

---

### Description

`datasailr::sail()` is the main function of this 'DataSailr' package. This function takes `data.frame` as the 1st argument and 'DataSailr' script as the 2nd argument.

### Usage

```
sail(df, code, fullData = TRUE, rowname = "_rowname_", stringsAsFactors = FALSE)
```

**Arguments**

<code>df</code>	data.frame to be processed.
<code>code</code>	'DataSailr' script that instructs how to manipulate values and strings.
<code>fullData</code>	When <code>fullData</code> is set FALSE, only the columns corresponding to left-hand-side(LHS) variables in 'DataSailr' scripts are returned. (e.g. In 'DataSailr' script, <code>bmi = weight / height / height</code> , the variable <code>bmi</code> is the LHS.) If set TRUE, all the columns are returned.
<code>rowname</code>	Dataframe's row names are accessed via the variable name specified by this option. The default variable name is <code>_rowname_</code> , meaning that <code>_rowname_</code> represents dataframe's row names in 'DataSailr' script, though this does not allow assignment. Even if you assign some value to this variable, they are not reflected in the row names of the resultant dataframe.
<code>stringsAsFactors</code>	Columns of character vectors (= string vectors) that appear as left-hand-side(LHS) variables in 'DataSailr' scripts can be returned as either character vectors or factor vectors. If <code>stringsAsFactors</code> is set to FALSE(default), character columns are returned as character vectors. If <code>stringsAsFactors</code> is set to TRUE, character columns are returned as factor vectors.

**Value**

If `fullData` argument is set FALSE, return value is a data.frame. If `fullData` is TRUE, the result is created by `cbind()`, which concatenate the original input and newly created dataframe. For example, if the original input is 'data.table', the return value is 'data.table'.

**Examples**

```
library(datasailr)
data(iris)

iris_code = '
iris_type = .
if ( Species == "setosa" ) {
  type = 1
  type_char = "A"
}else if( Species == "versicolor" ) {
  type = 2
  type_char = "B"
}else if(Species == "virginica" ) {
  type = 3
  type_char = "C"
}
'
sail(iris, iris_code)
```

---

test_sail	<i>run DataSailr test scripts</i>
-----------	-----------------------------------

---

**Description**

datasailr::test\_sail() run test scripts for 'DataSailr' package.

**Usage**

```
test_sail()
```

**Arguments**

No arguments required.

**Examples**

```
library(datasailr)
test_sail()
```

# Index

\* **datasets**

adult\_total, 3

\* **package**

datasailr-package, 2

adult\_total, 3

author, 4

datasailr-package, 2

sail, 2, 4

test\_sail, 6