Package: SEER2R (via r-universe)

November 15, 2026

Type Package

Title reading and writing SEER*STAT data files

Version 1.0.9001

Date 2011-09-28

Author Jun Luo [aut, cre]

Maintainer Jun Luo <rpackages@gmail.com>

Description read and write SEER*STAT data files Compatibility updates by Renova.

License GPL (>= 2)

LazyLoad yes

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-15 20:37:03 UTC

RemoteUrl https://github.com/renova-r/SEER2R

RemoteRef renova-1.0.9001

RemoteSha 119da3ac970c23a936bf696ad9215694326201a0

Contents

SEER2R-package 2
read.SeerStat 3
SEER2RTestdata 5
write.SeerStat 7

Index **9**

SEER2R-package *read from and write to SEER*STAT export files*

Description

The Surveillance, Epidemiology, and End Results (SEER) Program of the National Cancer Institute (NCI) is an authoritative source of information on cancer incidence and survival in the United States. The SEER*Stat software developed by NCI provides a convenient, intuitive mechanism for the analysis of SEER and other cancer-related databases. SEER*STAT presents results in matrix session and users can export results into data file (either in plain text format or gz format) and dictionary information into dic file. read.SeerStat reads data and dictionary information from SEER*STAT export files. write.SeerStat writes data and dictionary information to files in the formats of SEER*STAT export files

Details

Package:	SEER2R
Type:	Package
Version:	1.0
Date:	2011-09-28
License:	
LazyLoad:	yes

Author(s)

Maintainer: Jun Luo <rpackages@gmail.com>

References

Jun Luo and Binbing Yu, 'SEER2R: An interface between SEER cancer registry data and R'

See Also

read.SeerStat, write.SeerStat

Examples

```
#load testing data: there are three data objects, i.e. SEER2RTestData1,SEER2RTestData2,SEER2RTestData3
data("SEER2RTestdata");

#create one SEER*STAT export dic and the associated text data file for testing purpose
dicinfoused = write.SeerStat(SEER2RTestData2, DICfileName = "testrun1.dic",UseVarLabelsInTxtFile=FALSE);

#usage of read.SeerStat
```

```
mydata = read.SeerStat("testrun1.dic",UseVarLabelsInData=FALSE);
#get informatin inside the dic file
DICInfo = attr(mydata, "DICInfo");

#change names of columns whose names contains strings "site" or "sex";
#the order of strings does not matter
testdatanewnames = attr(mydata,"assignColNames")(mydata,c("sex","site"));

#extract columns whose names contains strings "site" or "sex";
testdata = attr(mydata,"getSubDataByVarName")(mydata,c("site","sex"));

#usage of write.SeerStat
dicinfoused = write.SeerStat(mydata, DICfileName = "testrun2.dic", UseVarLabelsInTxtFile = FALSE);
```

`read.SeerStat` *read from SEER*STAT export files*

Description

SEER*STAT presents results in matrix session and users can export results into data file (either in plain text format or gz format) and dictionary information into dic file. read.SeerStat reads data and dictionary information from SEER*STAT export files.

Usage

```
read.SeerStat(DICfileName, TXTfileName = NULL, UseVarLabelsInData = FALSE,ReadHeaderOnly=FALSE,...)
```

Arguments

`DICfileName`	: filename of the dic file. The default extention is 'dic'. If 'DICfileName' does not contain at the end a string '.dic' (letter case does not matter), then '.dic' will be added.
`TXTfileName`	: filename of the associated data file. If TXTfileName = NULL, then a string of DICfileName with extention substituted by 'txt' (for uncompressed data file) or 'gz' (for gzip compressed data file) will be used (whether 'txt' or 'gz' depends on information in the dic file).
`UseVarLabelsInData`	: a logic value. If true, then variable labels read from the dic file will replace associated numeric values in the data.frame object, which stores data from the associated data file and is returned by this function read.SeerStat. If false, then data read from the associated data file won't be changed.
`ReadHeaderOnly`	: a logic value. If true, then only the list storing the information read from the dic file will be returned. Otherwise, A data frame object containing a representation of the data in the associated data file will be returned.
`...`	: Arguments to be passed to `read.table` for reading from the associated data file.

Details

"read.SeerStat" reads data from a SEER*Stat data file into an object of data.frame in R and stores information from the associated dictionary file in an attribute variable (named "DICInfo") of the data.frame object. The variables of the SEER*Stat data file are stored in columns of the data.frame object. The column names of the data.frame object are based on the variable names in the associated SEER*Stat dic file, with special characters ",:()<>={}!@#$ to a single '_'. For example, the column name will be "Example_Variable_1" if the variable name in the SEER*Stat dic file is "Example* (Variable 1)".

Value

A data frame (data.frame) containing a representation of the data in the associated data file and having three attributes ('DICInfo': a list storing dictionary information read from the dic file; 'assignColNames': a function that assigns new names (they are substrings of names of those columns that will be replaced) to asscociated columns of a data.frame object; 'getSubDataByVarName': a function that extracts columns of a data.frame object given substrings of names of columns that will be extracted).

or

a list storing dictionary information read from the dic file if ReadHeaderOnly is true

Author(s)

Jun Luo

Maintainer: Jun Luo <rpackages@gmail.com>

References

Jun Luo and Binbing Yu, 'SEER2R: An interface between SEER cancer registry data and R'

See Also

write.SeerStat, SEER2R

Examples

```
#load testing data: there are three data objects, i.e. SEER2RTestData1,SEER2RTestData2,SEER2RTestData3
data("SEER2RTestdata");

#create one SEER*STAT export dic and the associated text data file for testing purpose
dicinfoused = write.SeerStat(SEER2RTestData2, DICfileName = "testrun1.dic",UseVarLabelsInTxtFile=FALSE);

#usage of read.SeerStat
mydata = read.SeerStat("testrun1.dic",UseVarLabelsInData=FALSE);
#get informatin inside the dic file
DICInfo = attr(mydata, "DICInfo");

#change names of columns whose names contains strings "site" or "sex";
#the order of strings does not matter
testdatanewnames = attr(mydata,"assignColNames")(mydata,c("sex","site"));
```

```
#extract columns whose names contains strings "site" or "sex";
testdata = attr(mydata,"getSubDataByVarName")(mydata,c("site","sex"));

#usage of write.SeerStat
dicinfoused = write.SeerStat(mydata, DICfileName = "testrun2.dic", UseVarLabelsInTxtFile = FALSE);
```

SEER2RTestdata *Testing data for the package SEER2R*

Description

There are three data objects, i.e. SEER2RTestData1,SEER2RTestData2,SEER2RTestData3, contained in SEER2RTestData.rda.

Format

SEER2RTestData1: A data frame with 495 observations on the following 7 variables.

`Statecounty_Test` a factor with levels `CA: Alameda County (06001)` `CA: Contra Costa County (06013)` `CA: Marin County (06041)` `CA: San Francisco County (06075)` `CA: San Mateo County (06081)`

`Sex` a factor with levels `Female` `Male` `Male and female`

`19982008` a numeric vector

`Site_Test_` a factor with levels `Hodgkin - Extranodal` `NHL - Extranodal` `NHL - Nodal`

`AgeAdjusted_Rate` a numeric vector

`Count` a numeric vector

`Population` a numeric vector

SEER2RTestData2: A data frame with 628 observations on the following 20 variables.

`Page_type` a numeric vector

`Sites_LB` a numeric vector

`Year_of_diagnosis_7507_individual` a numeric vector

`Histologic_Type_ICDO3_LungSCNSCOth` a numeric vector

`Sex` a numeric vector

`Age_7584` a numeric vector

`Interval` a numeric vector

`Alive_at_Start` a numeric vector

`Died` a numeric vector

`Lost_to_Followup` a numeric vector

`Observed_Survival_Interval` a numeric vector

`Observed_Survival_Cum` a numeric vector
`Expected_Survival_Interval` a numeric vector
`Expected_Survival_Cum` a numeric vector
`Relative_Survival_Interval` a numeric vector
`Relative_Survival_Cum` a numeric vector
`Observed_SE_Interval` a numeric vector
`Observed_SE_Cum` a numeric vector
`Relative_SE_Interval` a numeric vector
`Relative_SE_Cum` a numeric vector

SEER2RTestData3: A data frame with 628 observations on the following 20 variables.

`Page_type` a character vector
`Sites_LB` a character vector
`Year_of_diagnosis_7507_individual` a character vector
`Histologic_Type_ICDO3_LungSCNSCOth` a character vector
`Sex` a character vector
`Age_7584` a character vector
`Interval` a character vector
`Alive_at_Start` a numeric vector
`Died` a numeric vector
`Lost_to_Followup` a numeric vector
`Observed_Survival_Interval` a numeric vector
`Observed_Survival_Cum` a numeric vector
`Expected_Survival_Interval` a numeric vector
`Expected_Survival_Cum` a numeric vector
`Relative_Survival_Interval` a numeric vector
`Relative_Survival_Cum` a numeric vector
`Observed_SE_Interval` a numeric vector
`Observed_SE_Cum` a numeric vector
`Relative_SE_Interval` a numeric vector
`Relative_SE_Cum` a numeric vector

Details

These data are extracted from SEER*STAT, which is available for general public.

References

Jun Luo and Binbing Yu, 'SEER2R: An interface between SEER cancer registry data and R'

Examples

```
data(SEER2RTestdata)
```

`write.SeerStat` *write data and dictionary information to files*

Description

write.SeerStat writes data and dictionary information to files in the format of SEER*STAT export files

Usage

```
write.SeerStat(myData, DICfileName, TXTfileName = NULL, UseVarLabelsInTxtFile = TRUE, LabVarsNames = NU
```

Arguments

`myData`	: A data.frame object to be output. It may have an attribute 'DICInfo'. If no 'DICInfo' defined, then a default 'DICInfo' will be created and used for generating the dic file.
`DICfileName`	: filename of the dic file. The default extention is 'dic'. If 'DICfileName' does not contain at the end a string '.dic' (letter case does not matter), then '.dic' will be added.
`TXTfileName`	: filename of the associated data file. If TXTfileName = NULL, then a string of DICfileName with extention substituted by 'txt' (for uncompressed data file) or 'gz' (for gzip compressed data file) will be used (whether 'txt' or 'gz' depends on the attribute variable 'DICInfo' of the data.frame object 'myData').
`UseVarLabelsInTxtFile`	: a logic value. If true, then variable labels will be output into the associated data file. If false, then the numeric values will be output. The variables, labels of which are to be output, are all character or factor columns of 'myData' if LabVarsNames = NULL, otherwise they are defined by 'LabVarsNames'.
`LabVarsNames`	: a name list of variables in the data.frame 'myData', labels of which will be output into the associated data file, instead of numeric values.
`...`	: Arguments to be passed to `write.table` for writing to the associated data file.

Details

"write.SeerStat" writes an object of data.frame to a SEER*Stat data file and creates the associated dictionary file that is in the format of SEER*Stat dictionary files. The variable names of the SEER*Stat data file are the same as the column names of the data.frame object. The contents of sections of dictionary file are based on the attribute variable "DICInfo" of the data.frame object. If users do not define an attribute variable "DICInfo" of the data.frame object, then default values will be used.

Value

The 'DICInfo' actually used for writing to the dictionary file and the associated data file will be returned.

Author(s)

Jun Luo

Maintainer: Jun Luo <rpackages@gmail.com>

References

Jun Luo and Binbing Yu, 'SEER2R: An interface between SEER cancer registry data and R'

See Also

read.SeerStat, SEER2R

Examples

```
#load testing data: there are three data objects, i.e. SEER2RTestData1,SEER2RTestData2,SEER2RTestData3
data("SEER2RTestdata");

#create one SEER*STAT export dic and the associated text data file for testing purpose
dicinfoused = write.SeerStat(SEER2RTestData2, DICfileName = "testrun1.dic",UseVarLabelsInTxtFile=FALSE);

#usage of read.SeerStat
mydata = read.SeerStat("testrun1.dic",UseVarLabelsInData=FALSE);
#get informatin inside the dic file
DICInfo = attr(mydata, "DICInfo");

#change names of columns whose names contains strings "site" or "sex";
#the order of strings does not matter
testdatanewnames = attr(mydata,"assignColNames")(mydata,c("sex","site"));

#extract columns whose names contains strings "site" or "sex";
testdata = attr(mydata,"getSubDataByVarName")(mydata,c("site","sex"));

#usage of write.SeerStat
dicinfoused = write.SeerStat(mydata, DICfileName = "testrun2.dic", UseVarLabelsInTxtFile = FALSE);
```

Index

∗ **Cancer registry**
    read.SeerStat, 3
    SEER2R-package, 2
    write.SeerStat, 7
∗ **SEER*STAT**
    read.SeerStat, 3
    SEER2R-package, 2
    write.SeerStat, 7
∗ **SEER2R**
    read.SeerStat, 3
    SEER2R-package, 2
    write.SeerStat, 7
∗ **datasets**
    SEER2RTestdata, 5
∗ **package**
    SEER2R-package, 2

read.SeerStat, *2*, 3, *8*
read.table, *3*

SEER2R, *4*, *8*
SEER2R *(*SEER2R-package*)*, 2
SEER2R-package, 2
SEER2RTestdata, 5
SEER2RTestData1 *(*SEER2RTestdata*)*, 5
SEER2RTestData2 *(*SEER2RTestdata*)*, 5
SEER2RTestData3 *(*SEER2RTestdata*)*, 5

write.SeerStat, *2*, *4*, 7
write.table, *7*