Package 'rSPARCS'

February 25, 2019

Type Package

Title Statistical Package for Analysis Related Cleaning Support

Version 0.0.4

Author Wangjian Zhang, Zhicheng Du, Ziqiang Lin, Jijin Yao, Yanan Jin, Wayne R. Lawrence, Yuantao Hao

Maintainer Wangjian Zhang <wzhang27@albany.edu>

Description To clean and analyze hospital data, and generate sets for statistical modeling.

License GPL-3

Encoding UTF-8

LazyData true

Imports foreign,data.table,spatialEco,geosphere,tigris,raster,sp,plyr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-25 15:30:03 UTC

R topics documented:

`case.series` *Generate the Case Series*

Description

Estimates the daily number of cases reported by multiple grouping factors.

Usage

```
case.series(data,ICD,diagnosis,date,start,end,by1,by2,by3,by4,by5)
```

Arguments

`data`	a data.frame containing with each row representing a case, and each column representing the patient characteristics such as gender, age, admission date, and discharge date, etc.
`ICD`	a vector of ICD 9, or 10 codes, or a mix of them, which users are willing to calculate the daily numbers for; can be of length 3-6.
`diagnosis`	the name of the variable in the data containing the diagnostic code upon admission.
`date`	the name of the variable in the data showing the admission date, either in the format like "20181129" or "2018/11/29".
`start,end`	the start and end date for the case series to be generated.
`by1,by2,by3,by4,by5`	the name of the variable in the data used as grouping variables.

Details

Not limited to hospital data, but also applicable to other surveillance data.

Value

`dataset`	A case series will be generated for time series analysis, trend analysis and displaying, with following variables:
`date`	from the start date to the end date as user specified, with 1 day bin.
`case`	the daily number of cases diagnosed with diseases of user specified ICD codes.
`others`	grouping variables.

Note

When applied to other medical data without ICD code, users may arbitrarily set a ICD code, meanwhile, define the diagnosis variable in the data to the same ICD code.

Examples

```
# similated data
set.seed(2018)

data=data.frame(
  patient=1:10000,
  primdiag=sample(390:398,10000,replace=TRUE),
  onset=sample(seq.Date(as.Date("2015/2/1"),
    as.Date("2016/2/1"),"1 day"),10000,replace=TRUE),
  sex=sample(c("M","F"),10000,replace=TRUE),
  county=sample(c("Albany","New York"),10000,replace=TRUE)
)

output.series=case.series(
  data,ICD=392:396,diagnosis="primdiag",
  date="onset",start="2015/1/1",end="2016/12/31",by1="sex")

head(output.series)
```

CXover.data *Generate the Dataset for Case Crossover Analysis*

Description

Generate the dataset for case crossover analysis.

Usage

```
CXover.data(data,date,ID,direction)
```

Arguments

data	a data.frame containing the date of each case.
date	the name of the variable in the data indicating the date of each case reported to the database.
ID	the name of the variable in the data indicating case ID, if not specified, it will automatically generated starting from 1.
direction	"pre4"or"month4". With "pre4", each case day will be matched with same weekdays in previous 4 weeks. With "month4", each case day will be matched with same weekdays in the same month.

Details

Not limited to hospital data, but also applicable to other surveillance data.

Value

dataset	A data.frame ready for the case crossover analysis, with following variables:
ID	same ID represents the same patient.
Date	one case day is matched with 3-4 control days.
status	indicating whether it is a case day or a control day.

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

References

Zhang W, Lin S, Hopke PK, et al. Triggering of cardiovascular hospital admissions by fine particle concentrations in New York state: Before, during, and after implementation of multiple environmental policies and a recession. Environ. Pollut. 2018;242:1404–1416.

Examples

```
# similated data
set.seed(2018)
dataset=data.frame(
  patient=1:1000,
  primdiag=sample(390:398,1000,replace=TRUE),
  onset=sample(seq.Date(as.Date("2015/2/1"),as.Date("2016/2/1"),"1 day"),1000,replace=TRUE),
  sex=sample(c("M","F"),1000,replace=TRUE),
  county=sample(c("Albany","New York"),1000,replace=TRUE))

out.data=CXover.data(data=dataset,date="onset",ID="patient",direction="pre4")
head(out.data)
```

DBFgeocode *Create a dbf File for Geocoding*

Description

Generate address variables and output the data as a dbf file for geocoding.

Usage

```
DBFgeocode(data,cityname,roadaddress,mailbox,ZIP,output)
```

Arguments

data	a data.frame containing address variables that are necessry for geocoding.
cityname	the name of the variable in the data indicating city or county names.
roadaddress	the name of the variable in the data indicating home addresses.
mailbox	optional address information such as the number of mailbox and the number of floor.
ZIP	the name of the variable in the data indicating ZIP codes.
output	specify the name of the dbf file (.dbf) and the directory for output.

Details

The suffix of the output argument should be ".csv" instead of ".dbf".

Value

A dbf file will be automatically output to the computer.

Note

In the dbf file, a variable named "singleline" will be used in the second step of geocoding, while variables roadaddress,cityname and ZIP will be seperately used in the first step, and the variable ZIP for the last step.

Examples

```
# similated data
datatest=data.frame(county=c("Albany","Albany","Albany"),
                address1=c("1 Lincoln ave","2 Lincoln ave","489 Washinton ave"),
                address2=c("1st floor","1st floor","2nd floor"),
                zip=12206
                )
DBFgeocode(data=datatest,cityname="county",roadaddress="address1",
mailbox="address2",ZIP="zip",output="data_output.csv")
```

---

desc.comp *Generate a Descriptive Table*

Description

Generate a comprehensive descriptive table with intergroup comparison.

Usage

```
desc.comp(data,variables,by,margin,avg.num,test.num)
```

Arguments

data	a data.frame containing the variables to be described and a group variable
variables	a numeric variable indicating the columns of variables to be described.
by	a number indicating the column of the group variable
margin	calculate the proportion for categorical variables by 1 (row) or 2 (column).
avg.num	"mean", describe continuous variables with mean and standard deviation; "median", describe continuous variables with median and interquantile range; otherwise, normal distribution test will be conducted, for normal distributed variables, "mean" will be used, otherwise, "median" will be used.
test.num	"metric", t test or anova will be used for intergroup comparison; "nonmetric", Wilcoxon rank sum test or Kruskal-Wallis test will be used; otherwise, normal distribution test will be conducted, for normal distributed variables, "metric" will be used, otherwise, "nonmetric" will be used.

Details

Not limited to hospital data, but also applicable to other surveillance data.

Value

A comprehensive descriptive table with statistics and P value for intergroup comparisons.

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

Examples

```
desc.comp(CO2,variables=2:5,by=1,margin=1)
```

---

dupl.readm *Identify Duplicates and Re-admissions*

Description

Identify the duplicates and re-admissions in hospital data with subject identifications.

Usage

```
dupl.readm(data,UniqueID,date,period)
```

Arguments

data	a data.frame containing "UniqueID" and "date"
UniqueID	the name of the variable in the data indicating case ID.
date	the name of the variable in the data indicating the admission/onset date.
period	the time period used to define an re-admission; period=365 by default.

Details

Not limited to hospital data, but also applicable to other surveillance data with "UniqueID" and "date".

Value

`id.dupl`	indicating whether it is a duplicated record with exactly the same "UniqueID" and "date" as a previous record. In some hospital data,some patients may be reported twice or even more due to insurance issues. For most studies, researchers may remove this kind of duplicates to avoid potential overcounting problems.
`onlyone`	indicating whether this is the only record with this ID.
`Period`	the time period between the current visit and the previous one for a patient; 0 for the 1st visit; and NA for those with only one record.
`Nadmission`	indicating the times of admission, e.g. 1st, 2nd admission; a patient may have more than one 1st admissions if some periods between two visits are greater than e.g. 365 days.

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

Examples

```
dataset=data.frame(
  ID=c(1,3,4,2,4,6,3,5,7,1),
  onset=c("2015/1/1","2016/1/2","2015/5/9",
          "2015/12/1","2016/8/2","2015/5/9",
  "2015/11/1","2016/3/2","2016/5/9","2015/9/9")
)

out.data=dupl.readm(data=dataset,
                    UniqueID="ID",date="onset",period=365)
head(out.data)
```

FIPS.name *Add county/census tract names or FIPS code*

Description

Identify the residential census tracts for each case, and add county/census tract names or FIPS code.

Usage

```
FIPS.name(data,patco,level,add,addfrom,state,county,map,long.case,lat.case,censusFIPS)
```

Arguments

data	a data.frame containing coordinates of cases for level="census"
patco	the name of variable in the data indicating the county code.
level	"county" or "census", indicating the study level.
add	"name" or "FIPS", or a vector containing both, to specify what vairables to be added.
addfrom	a data.frame containing "COUNTY" (county names),"FIPS" (county FIPS code) and"CODE" (county code, should match those in the health data).
state	State FIPS code, e,g, "36" for the New York State.
county	County FIPS code, e,g, "36001" for Albany, we use "001" for Albany here.
map	A map for a region outside the U.S. can also be imported as a "spatialpolygons-dataframe" object.
long.case	the name of variable in the data indicating the longitude of cases.
lat.case	the name of variable in the data indicating the latitude of cases.
censusFIPS	the name of variable in the map indicating the FIPS for census tracts. Use the default if the study is within the U.S.

Details

Not limited to hospital data, but also applicable to other surveillance data.

Value

FIPS	the FIPS code at county or census tract level depending on the arguement "level".
county	the name of counties.

Note

If you are working on the NY SPARCS data,no "addfrom" is required as this package has include a public information from https://www.health.ny.gov/statistics/sparcs/sysdoc/appf.htm

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

Examples

```
dataset=data.frame(Patient=1:10,county=5:14)
data.out=FIPS.name(data=dataset,patco="county",level="county")

#set.seed(2018)
#dataset=data.frame(Patient=1:10,lat=rnorm(10,42,0.5),long=rnorm(10,-76,1))
#data.out=FIPS.name(data=dataset,level="census",state="36",
#long.case="long",lat.case="lat",censusFIPS="GEOID")

head(data.out)
```

`pick.cases` *Select cases within certain distance around a site*

Description

Identify the closest site (e.g. monitoring sites) for each case, and select cases within certain distance around a site, e.g. 15 miles buffer.

Usage

```
pick.cases(data,long.case,lat.case,long.sites,lat.sites,radius)
```

Arguments

`data`	a data.frame containing the coordinates of cases.
`long.case`	the name of variable in the data indicating the longitude of cases.
`lat.case`	the name of variable in the data indicating the latitude of cases.
`long.sites`	a numeric vector containing the longitude of sites.
`lat.sites`	a numeric vector containing the latitude of sites.
`radius`	radius of the buffer, e.g."15 miles", "20 kms".

Details

Not limited to hospital data, but also applicable to other surveillance data.

Value

`which.site`	the closest site to the case.
`minDIST`	the distance of the case to the closest site; in the same unit as "radius".
`Select`	an indicator of whether a case was within the buffer.

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

References

Zhang W, Lin S, Hopke PK, et al. Triggering of cardiovascular hospital admissions by fine particle concentrations in New York state: Before, during, and after implementation of multiple environmental policies and a recession. Environ. Pollut. [electronic article]. 2018;242:1404–1416.

Examples

```
set.seed(2018)
data=data.frame(Patient=1:100,lat=rnorm(100,41,0.5),long=rnorm(100,-76,1))

long.monitor=c(-73.75464,-78.80953,-73.902,-73.82153,-77.54817)
lat.monitor=c(42.64225,42.87691,40.81618,40.73614,43.14618)

data.out=pick.cases(data,long.case="long",lat.case="lat",
long.sites=long.monitor,lat.sites=lat.monitor,radius="30 miles")
data.out
```

---

raster_extract *Extract Values from a Raster Map*

Description

Crop the raster with the boundary of areas of your interest, and extract the values from the raster to each of these areas.

Usage

```
raster_extract(rastermap,refmap,ID.var,ID.code,cutpoint)
```

Arguments

rastermap	a raster map containing the information you need, such as the National Land Cover Database 2011.
refmap	"SpatialPolygonsDataFrame" object. A reference map containing the boundary information of your study areas.
ID.var	the name of variable in the refmap indicating the unique ID for each of your study areas.
ID.code	a character vector containing the unique ID for areas that you want to extract the values to. ID.code=ALL" by default where all areas in the reference map are of interest.
cutpoint	a number to dichotomize the values in the raster; specified ONLY when those values are continuous.

Details

Usually for extracting data which are available as rasters such as the land coverage or land usage data.

Value

ID.code	the column indicating the unique ID for each area, followed by the number of cells for each category/colour within that area.
Total cells	the total number of cells within each area.

Author(s)

Wang-Jian Zhang (wzhang27@albany.edu)

Examples

```
#library(raster)
#set.seed(4715)
#rast=raster(matrix(rnorm(500),100,100))
#extent(rast)=c(50,100,10,60)
#crs(rast)=CRS("+proj=longlat +datum=WGS84")

#ref=cbind(x=c(60,80,80,70), y=c(20,25,40,30))
#p=Polygon(ref)
#ps=Polygons(list(p),ID="ID")
#ref=SpatialPolygons(list(ps))
#data=data.frame(value=1, ID="10086",row.names="ID")
#ref=SpatialPolygonsDataFrame(ref,data)
#proj4string(ref)=CRS("+proj=longlat +datum=WGS84")

#raster_extract(rastermap=rast,refmap=SPDF,ID.var="ID",ID.code="ALL",cutpoint=0.5)
```

Index