Package 'revengc'

August 18, 2017

Type Package

Title Reverse Engineering Censored, Decoupled Residential Data for Population Density Estimation

Version 1.0.0

Author Samantha Duchscherer [aut, cre], UT-Battelle, LLC [cph]

Maintainer Samantha Duchscherer <sam.duchscherer@gmail.com>

Description A wealth of open source information is available that points to building usage including floor area of building size and likely occupancy. In the case of residential structures, census data provides a number of attributes including two values important to interior density estimation: household size (hhs) and area. If a national census revealed the raw data or provided a full uncensored contingency table (hhs x area), computing interior density as people/area would be straightforward. However, agencies rarely report this contingency table. Rather hhs and area are often decoupled and reported as separate univariate frequency tables, average values, or a combination of the two. In addition, the decoupled or contingency tables provided are typically left (<, <=), right (>, >=), and interval (-) censored. This type of information becomes problematic in estimating interior residential occupancy for numerous reasons. How can the people/area ratio be calculated when no affiliation between the variables exist? If a census reports a hhs average of 5.3, then how many houses are there with 1 person, 2 people,..., 10 people? If a census reports that there are 100 houses in an area of 26-50 square meters, then how many houses are in 26, 27,..., 50 square meters? The challenge therefore is to infer the people/area ratio when given decoupled and summarized data. The statistical package 'revengc' was designed to reverse engineer censored, decoupled census data into a likely hhs x area uncensored contingency table for estimating interior residential occupancy.

URL https://github.com/GIST-ORNL/revengc

Depends R (>= 2.14)

License MIT + file LICENSE

LazyData TRUE

Imports stringr

Suggests R.rsp

VignetteBuilder R.rsp

RoxygenNote 6.0.1.9000

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-18 18:59:00 UTC

R topics documented:

`hongkong_area` *Area frequency table (meters squared) for Hong Kong*

Description

A 2016 censored, univariate frequency table containing area (meters squared) for 87600 sub-divided units in Hong Kong. This table is formatted to look the same as a read in csv file [read.csv("filename.csv", row.names = NULL, header= FALSE, check.names=FALSE)].

Usage

```
hongkong_area
```

Source

Thematic Household Survey Report - Report No. 60 - Housing conditions of sub-divided units in Hong Kong, 2016, page 23.

References

Census and Statistics Department of Hong Kong Special Administrative Region . (2016). *Thematic Household Survey Report - Report No. 60 - Housing conditions of sub-divided units in Hong Kong*. Retrieved from: http://www.censtatd.gov.hk/hkstat/sub/sp100.jsp?productCode=C0000091

hongkong_hhs *Household size frequency table for Hong Kong*

Description

A 2016 censored, univariate frequency table containing hhs for 87600 sub-divided units in Hong Kong. This table is formatted to look the same as a read in csv file [read.csv("filename.csv", row.names = NULL, header= FALSE, check.names=FALSE)].

Usage

```
hongkong_hhs
```

Source

Thematic Household Survey Report - Report No. 60 - Housing conditions of sub-divided units in Hong Kong, 2016, page 31.

References

Census and Statistics Department of Hong Kong Special Administrative Region . (2016). *Thematic Household Survey Report - Report No. 60 - Housing conditions of sub-divided units in Hong Kong*. Retrieved from: http://www.censtatd.gov.hk/hkstat/sub/sp100.jsp?productCode=C0000091

indonesia_contingency *Contingency table of household size x area (meters squared) for Indonesia's rural Aceh Province*

Description

A 2010 censored contingency table containing household size and area (meters squared) in Indonesia's rural Aceh Province. This table is formatted to look the same as a read in csv file [read.csv("filename.csv", row.names = NULL, header= TRUE, check.names=FALSE)].

Usage

```
indonesia_contingency
```

Source

Population Census Data - Statistics Indonesia, 2010

References

Population Census Data - Statistics Indonesia. (2010). *Household by Floor Area of Dwelling Unit and Households Member Size*. Retrieved from: http://sp2010.bps.go.id/index.php/site/tabel?wid=1100000000&tid=334&fi1=

iran_hhs — *Household size frequency table for East Azerbayejan (Azerbaijan), Iran*

Description

A 2011 censored, univariate frequency table containing household size in East Azerbayejan (Azerbaijan), Iran. This table is formatted to look the same as a read in csv file [read.csv("filename.csv", row.names = NULL, header= FALSE, check.names=FALSE)].

Usage

```
iran_hhs
```

Source

Selected Findings of National Population and Housing Census, 2011, page 54.

References

The Statistical Centre of Iran. (2011). *Selected Findings of National Population and Housing Census*. Retrieved from: https://www.amar.org.ir/Portals/1/Iran/90.pdf

nepal_hhs — *Household size frequency table for urban Nepal*

Description

A 2011 censored, univariate frequency table containing household size in urban Nepal. This table is formatted to look the same as a read in csv file [read.csv("filename.csv", row.names = NULL, header= FALSE, check.names=FALSE)].

Usage

```
nepal_hhs
```

Source

The Nepal Living Standards Survey, 2011, page 28.

References

National Planning Commissions Secretariat, Government of Nepal. (2011). *Nepal Living Standards Survey*. Retrieved from: http://siteresources.worldbank.org/INTLSMS/Resources/3358986-1181743055198/3877319-1329489437402/Statistical_Report_Vol1.pdf

rec *Reverse Engineering Census (rec)*

Description

A wealth of open source information is available that points to building usage including floor area of building size and likely occupancy. In the case of residential structures, census data provides a number of attributes including two values important to interior density estimation: household size (hhs) and area. If a national census revealed the raw data or provided a full uncensored contingency table (hhs x area), computing interior density as people/area would be straightforward. However, agencies rarely report this contingency table. Rather hhs and area are often decoupled and reported as separate univariate frequency tables, average values, or a combination of the two. In addition, the decoupled or contingency tables provided are typically left (<, <=), right (>, >=), and interval (-) censored. This type of information becomes problematic in estimating interior residential occupancy for numerous reasons. How can the people/area ratio be calculated when no affiliation between the variables exist? If a census reports a hhs average of 5.3, then how many houses are there with 1 person, 2 people,..., 10 people? If a census reports that there are 100 houses in an area of 26-50 square meters, then how many houses are in 26, 27,..., 50 square meters? The challenge therefore is to infer the people/area ratio when given decoupled and summarized data. The statistical package revengc was designed to reverse engineer censored, decoupled census data into a likely hhs x area uncensored contingency table for estimating interior residential occupancy.

Usage

```
rec(hhsdata, areadata, hhslowerbound, hhsupperbound, arealowerbound,
  areaupperbound)
```

Arguments

hhsdata	This household size value can be a univariate frequency table or numeric value that represents an average. This input could also be a contingency table, but only if the areadata = 0.
areadata	This area (size of house) value can be a univariate frequency table or numeric value that represents an average. This input could also be a contingency table, but only if the hhsdata = 0. The areadata can be any unit of measure.
hhslowerbound	This is a numeric value to represent the household size lower bound. This lower bound variable needs to be numeric value >=0.
hhsupperbound	This is a numeric value to represent the household size upper bound. This upper bound variable cannot be less than the highest category value (e.g. if a table has '>100' then the upper bound cannot be 90).
arealowerbound	This is a numeric value to represent the area lower bound. This lower bound variable needs to be numeric values >=0.
areaupperbound	This is a numeric value to represent the area upper bound. This upper bound variable cannot be less than the highest category value (e.g. if a table has '>100' then the upper bound cannot be 90).

Details

This rec function will handle four different types of input:

- Case I. hhs average, area average, hhs lower bound, hhs upper bound, area lower bound, and area upper bound
- Case II. hhs frequency table, area frequency table, hhs lower bound, hhs upper bound, area lower bound, and area upper bound
- Case III. hhs average or frequency table, area average or frequency table, hhs lower bound, hhs upper bound, area lower bound and area upper bound
- Case IV. contingency table (hhs, area) or (area, hhs), hhs lower bound, hhs upper bound, area lower bound, and area upper bound

Bounds:

Ideally, the four bounds should be chosen based off prior knowledge and expert elicitation, but they can also be selected intuitively with a brute force method. If the reverse engineering tool outputs a final contingency table with higher probabilities near the edge(s) of the table, then it would make sense to increase the range of the bound(s). For both the hhs and area variables, this would just involve making the lower bound less, making the upper bound more, or doing a combination of the two. The opposite holds true as well. If the final contingency table has very low probabilities near the edge(s) of the table, then a person should decrease the range of the particular bound(s).

Tables:

The table(s) for Case II and III has restrictions. The frequency table must be formatted where there are two columns with n number of rows. The categories must be in the first column and the frequencies in the second column. Row names should never be placed in this table, the default name should always be 1:n where n is number of rows in the table. Both columns should not have a header (header=FALSE). No words are allowed for censoring. The only censoring symbols accepted are < and <= (left censoring), - (interval censoring), > and >= and + (right censoring). A formatted example is below.

<=6	11800
7-12	57100
13-19	14800
20+	3900

The table for Case IV also has restrictions. Again, no words are allowed for censoring. Only the censored values of <, <=, -, >, >=, and + are allowed. This table works when there is a column header present or absent. However, the only column header that is allowed has to be the hhs or area category values. Row names should never be placed in this table, the default name should always be 1:n where n is number of rows in the table. The inside of this table is the cross tabulation of hhs x area which are either positive frequency values or percentages. The row and column total marginals have to be in this table. The top left, top right, and bottom left corners of this table have to be NA or blank, but the bottom right corner can be a total sum value, NA, or blank. This code will transpose a contingency table if given a table with area=rows and hhs=columns, but the output will always be hhs=rows and area=columns. This transpose will only occur under the assumption that the sum of area category value is greater than the sum of household size category value. Below is a formatted

example with percentages as the cross-tabulations, the bottom right corner as a total sum, and the column header as the area category values.

NA	<20	20-30	>30	NA
<5	0.18	0.19	0.08	0.45
5-9	0.13	0.08	0.12	0.33
>=10	0.06	0.05	0.10	0.21
NA	0.38	0.32	0.31	1.00

Value

This function produces an uncensored contingency table with household size as the rows and area as the columns. The rows will range from the household size lower bound to the household size upper bound. The columns will range from the area lower bound to the area upper bound. If you uploaded a censored contingency table (Case IV), then these cross tabulations are probabilities calculated from a bivariate Poisson distribution. For the rest of the entries (Case I, II, and III), the probabilities are calculated using the IPF algorithm.

Examples

```
#Example found in the Nepal Living Standards Survey (2011)
#An average household size and an average area (square feet) are provided for urban Nepal

  nepal_results1<-rec(4.4,571.3,1,20,520,620)

  #A table for urban Nepal is also listed in this census
  #Would read in the a csv file as
  #nepal_hhs<-read.csv("filename.csv",row.names=NULL,header=FALSE,check.names=FALSE)
  #Using a read csv file for urban Nepal look like the following
  nepal_results2<-rec(nepal_hhs,571.3,1,20,520,620)



#Example found in the Census and Statistics Department of Hong Kong (2016)
#Focusing on 87,600 sub-divided Hong Kong units
#A table for household size and a table for area (square meters) are provided

  #Would create tables in R by
  hhs_hongkong<-cbind(as.character(c("1", "2", "3", ">3")), c(27600,25600,20900,13500))
 area_hongkong<-cbind(as.character(c("<7", "7-12", "13-19", ">19")), c(11800,57100,14800,3900))
  hongkong_results<-rec(hhs_hongkong,area_hongkong,1,15,1,30)

  #Would read in the two csv file as
  #hongkong_hhs<-read.csv("filename.csv",row.names=NULL,header=FALSE,check.names=FALSE)
  #hongkong_area<-read.csv("filename.csv",row.names=NULL,header=FALSE,check.names=FALSE)
  #Using a read csv file for Hong Kong looks like the following
  hongkong_results<-rec(hongkong_hhs,hongkong_area,1,15,1,30)



#Example found in the Statistical Centre of Iran (2011)
```

```
#Focusing on East Azerbayejan (Azerbaijan), Iran
#A table for household size and an average for area (square meters) area are provided

  #Would create table in R by
  hhs_iran<-cbind(as.character(c("1", "2", "3", "4", ">=5")), c(7.08,18.29,29.64,27.95,17.04))
  iran_results<-rec(hhs_iran,100.5,1,10,80,130)

  #Would read in a csv file as
  #iran_hhs<-read.csv("filename.csv",row.names=NULL,header=FALSE,check.names=FALSE)
  #Using a read csv file for East Azerbayejan looks like the following
  iran_results<-rec(iran_hhs,100.5,1,10,80,130)



#Example found in the Population Census Data - Statistics Indonesia (2010)
#A censored contingency table is provided for rural Aceh Province, Indonesia (square meters)

  #Would create table in R by
  contingencytable<-matrix(c(6185,9797,16809,11126,6156,3637,908,147,69,4,
                             5408,12748,26506,21486,14018,9165,2658,567,196,78,
                             7403,20444,44370,36285,23576,15750,4715,994,364,136,
                             4793,17376,44065,40751,28900,20404,6557,1296,555,228,
                             2354,11143,32837,33910,26203,19301,6835,1438,618,245,
                             1060,6038,19256,21298,17774,13864,4656,1039,430,178,
                             273,2521,9110,11188,9626,7433,2608,578,196,112,
                             119,1130,4183,5566,5053,3938,1367,318,119,66,
                             33,388,1707,2367,2328,1972,719,171,68,37,
                             38,178,1047,1672,1740,1666,757,193,158,164),
                             nrow=10,ncol=10, byrow=TRUE)
  rowmarginal<-apply(contingencytable,1,sum)
  contingencytable<-cbind(contingencytable, rowmarginal)
  colmarginal<-apply(contingencytable,2,sum)
  contingencytable<-rbind(contingencytable, colmarginal)
  row.names(contingencytable)[row.names(contingencytable)=="colmarginal"]<-""
  contingencytable<-data.frame(c("1","2","3","4","5","6", "7", "8","9","10+", NA), contingencytable)
  colnames(contingencytable)<-c(NA,"<20","20-29","30-39","40-49","50-69","70-99",
                                "100-149","150-199","200-299","300+", NA)


  #Note rec(contingencytable,0,1,15,10,310) == rec(0,contingencytable,1,15,10,310)
  indonesia_results<-rec(contingencytable,0,1,15,10,310)

## Not run:
  #Would read in a csv file as
  #indonesia_contingency<-read.csv("filename.csv",row.names=NULL,header=TRUE,check.names=FALSE)
  #Using a read csv for rural Aceh Province, Indonesia looks like the following
  indonesia_results<-rec(indonesia_contingency,0,1,15,10,310)

## End(Not run)
```

Index