

Package: rcrimeanalysis (via r-universe)

October 11, 2024

Type Package

Title An Implementation of Crime Analysis Methods

Version 0.5.0

Author Jamie Spaulding and Keith Morris

Maintainer Jamie Spaulding <jspaulding02@hamline.edu>

Description An implementation of functions for the analysis of crime incident or records management system data. The package implements analysis algorithms scaled for city or regional crime analysis units. The package provides functions for kernel density estimation for crime heat maps, geocoding using the 'Google Maps' API, identification of repeat crime incidents, spatio-temporal map comparison across time intervals, time series analysis (forecasting and decomposition), detection of optimal parameters for the identification of near repeat incidents, and near repeat analysis with crime network linkage.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports forecast, ggmap, htmltools, igraph, leaflet, leafsync, lubridate, KernSmooth, pals, raster, sp, stats, terra

RoxygenNote 7.2.3

Suggests dplyr, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://jsspaulding.r-universe.dev>

RemoteUrl <https://github.com/jsspaulding/rcrimeanalysis>

RemoteRef HEAD

RemoteSha da66ddb9ef3d240cba54f429e69549ccf444968b

Contents

crimes	2
geocode_address	3
id_repeat	4
kde_int_comp	5
kde_map	6
near_repeat_analysis	7
near_repeat_eval	8
ts_daily_decomp	9
ts_forecast	10
ts_month_decomp	11
Index	12

crimes	<i>Example data from the Chicago Data Portal</i>
--------	--

Description

A sample dataset of crime incidents in Chicago, IL from 2017-2019.

Usage

crimes

Format

A data frame with 25000 rows and 22 variables.

id Unique identifier for the record.

case_number The Chicago Police Department Records Division Number, which is unique to the incident.

date Date when the incident occurred.

block Partially redacted address where the incident occurred.

iucr Illinois Unifrom Crime Reporting code (directly linked to primary_type and description)

primary_type The primary description of the IUCR code.

description The secondary description of the IUCR code, a subcategory of the primary description.

location_description Description of the location where the incident occurred.

arrest Indicates whether an arrest was made.

domestic Indicates whether the incident was domestic-related as defined by the Illinois Domestic Violence Act.

beat Indicates the police beat where the incident occurred.

district Indicates the police district where the incident occurred.

ward The ward (City Council district) where the incident occurred.

community_area Indicates the community area where the incident occurred.

fbi_code Indicates the National Incident-Based Reporting System (NIBRS) crime classification.

x_coordinate X coordinate of the incident location (State Plane Illinois East NAD 1983 projection).

y_coordinate Y coordinate of the incident location (State Plane Illinois East NAD 1983 projection).

year Year the incident occurred.

updated_on Date and time the record was last updated.

latitude The latitude of the location where the incident occurred.

longitude The longitude of the location where the incident occurred.

location Concatenation of latitude and longitude.

Source

<https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data>

geocode_address

Batch Geocoding of Physical Addresses using the Google Maps API

Description

Geocodes a location (determines latitude and longitude from physical address) using the Google Maps API. Note that the Google Maps API requires registered credentials (Google Cloud Platform), see the ggmap package for more details at <https://github.com/dkahle/ggmap>. Note that when using this function you are agreeing to the Google Maps API Terms of Service at <https://cloud.google.com/maps-platform/terms/>.

Usage

```
geocode_address(location)
```

Arguments

location	a character vector of physical addresses (e.g. 1600 University Ave., Morgantown, WV)
----------	--

Value

Returns a two column matrix with the latitude and longitude of each location queried.

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
library(ggmap) #needed to register Google Cloud Credentials
register_google("**Google Cloud Credentials Here**")
addresses <- c("Milan Puskar Stadium, Morgantown, WV", "Woodburn Hall, Morgantown, WV")
geocode_address(addresses)
```

id_repeat	<i>Identify Repeat Crime Incidents</i>
-----------	--

Description

This function identifies crime incidents which occur at the same location and returns a list of such incidents where each data frame in the list contains the RMS data for the repeat crime incidents. The data is based on the Chicago Police Department RMS structure.

Usage

```
id_repeat(data)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
------	---

Value

A list where each data frame contains repeat crime incidents for a given location.

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
#Using provided dataset from Chicago Data Portal:
data(crimes)
crimes <- head(crimes, n = 1000)
out <- id_repeat(crimes)
```

kde_map	<i>Kernel Density Estimation and Heat Map Generation for Crime Incidents</i>
---------	--

Description

This function computes a kernel density estimate of crime incident locations and returns a 'Leaflet' map of the incidents. The data is based on the Chicago Police Department RMS structure and populates pop-up windows with the incident location for each incident.

Usage

```
kde_map(data, pts = NULL)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
pts	Either true or false. Dictates whether the incident points will be plotted on the map widget. If NULL, the default value is TRUE.

Value

A **Leaflet** map with three layers: an 'ESRI' base-map, all crime incidents plotted (with incident info pop-up windows), and a kernel density estimate of those points.

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
#Using provided dataset from Chicago Data Portal:
data(crimes)
crimes <- head(crimes, 1000)
library('leaflet') # needed to install basemap providers
kde_map(crimes)
```

near_repeat_analysis *Near Repeat Analysis of Crime Incidents with Crime Linkage Output*

Description

This function performs near repeat analysis for a set of incident locations. The user specifies distance and time thresholds which are utilized to search all other incidents and find other near repeat incidents. From this an adjacency matrix is created for incidents which are related under the thresholds. The adjacency matrix is then used to create an igraph graph which illustrates potentially related or linked incidents (under the near repeat thresholds).

Usage

```
near_repeat_analysis(  
  data,  
  epsg,  
  dist_thresh = NULL,  
  time_thresh = NULL,  
  tz = NULL  
)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
epsg	The EPSG Geodetic Parameter code for the area being considered. The EPSG code is used for identifying projections and performing coordinate transformations. If needed, the EPSG for an area can be found at https://spatialreference.org .
dist_thresh	The spatial distance (in meters) which defines a near repeat incident. By default this value is set to 1000 meters.
time_thresh	The temporal distance (in days) which defines a near repeat incident. By default this value is set to 7 days.
tz	Time zone for which the area being examined. By default this value is assigned as the same time zone of the system. For more information about time zones within R, see https://www.rdocumentation.org/packages/base/versions/3.6.1/topics/timezones .

Value

Returns a list of all near repeat series identified within the input data as **igraph** graph objects. This list can be used to generate plots of each series and to discern the near repeat linkages between the crime incidents.

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
data(crimes)
nr_data <- head(crimes, n = 1000) #truncate dataset for near repeat analysis
out <- near_repeat_analysis(data=nr_data,tz="America/Chicago",epsg="32616")
```

near_repeat_eval	<i>Identification of Optimal Time and Distance Parameters for Near Repeat Analysis</i>
------------------	--

Description

This function performs an evaluation of given crime incidents to recommend parameters for near repeat analysis. A series of time and distance parameters are tested using a full factorial design using the set of incident locations to determine the frequency of occurrence given each set of parameters. The results of the full factorial assessment are then modeled through interpolation and the second derivative is calculated to determine the inflection point. The inflection point represents the change in frequency of detected incidents which near repeat. Determination of the inflection point is completed for both the time and distance domains.

Usage

```
near_repeat_eval(data, epsg, tz = NULL)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
epsg	The EPSG Geodetic Parameter code for the area being considered. The EPSG code is used for identifying projections and performing coordinate transformations. If needed, the EPSG for an area can be found at https://spatialreference.org .
tz	Time zone for which the area being examined. By default this value is assigned as the same time zone of the system. For more information about time zones within R, see https://www.rdocumentation.org/packages/base/versions/3.6.1/topics/timezones .

Value

Returns a data frame with one instance (row) of two fields (columns). The fields are: distance and time. The instance indicates the optimal near repeat parameters for each. Note that distance is given in meters and time is given as days.

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
data(crimes)
nr_dat <- subset(crimes, crimes$primary_type == "BURGLARY")
pars <- near_repeat_eval(data=nr_dat, tz="America/Chicago", epsg="32616")
pars
```

ts_daily_decomp

*Time Series Forecast and Decomposition for Daily Crime Data***Description**

This function transforms daily crime count data and plots the resultant components of a time series which has been decomposed into seasonal, trend, and irregular components using Loess smoothing. Holt Winters exponential smoothing is also performed for improved trend resolution since data is in a daily format.

Usage

```
ts_daily_decomp(data, start)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
start	Start date for the time series being analyzed. The format is as follows: c('year', 'month', 'day'). See example below for reference.

Value

Returns an object of class "stl" with the following components:

- time.series: a multiple time series with columns seasonal, trend and remainder.
- weights: the final robust weights (all one if fitting is not done robustly).
- call: the matched call.
- win: integer (length 3 vector) with the spans used for the "s", "t", and "l" smoothers.
- deg: integer (length 3) vector with the polynomial degrees for these smoothers.
- jump: integer (length 3) vector with the 'jumps' (skips) used for these smoothers.
- inner: number of inner iterations

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
#Using provided dataset from Chicago Data Portal:
data(crimes)
test <- ts_daily_decomp(data = crimes, start = c(2017, 1, 1))
plot(test)
```

ts_forecast

Time Series Forecast for Daily Crime Data

Description

This function transforms traditional crime data into a time series and forecasts future incident counts based on the input data over a specified duration. The forecast is computed using simple exponential smoothing with additive errors. Returned is a plot of the time series, trend, and the upper and lower prediction limits for the forecast.

Usage

```
ts_forecast(data, start, duration = NULL)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
start	Start date for the time series being analyzed. The format is as follows: c('year', 'month', 'day'). See example below for reference.
duration	Number of days for the forecast. If NULL, the default duration for the forecast is 365 days.

Value

Returns a plot of the time series entered (black), a forecast over the specified duration (blue), the exponentially smoothed trend for both the input data (red) and forecast (orange), and the upper and lower bounds for the prediction interval (grey).

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
#Using provided dataset from Chicago Data Portal:
data(crimes)
ts_forecast(crimes, start = c(2017, 1, 1))
```

ts_month_decomp*Time Series Decomposition for Monthly Crime Data*

Description

This function transforms traditional crime data and plots the resultant components of a time series which has been decomposed into seasonal, trend and irregular components using Loess smoothing.

Usage

```
ts_month_decomp(data, start)
```

Arguments

data	Data frame of crime or RMS data. See provided Chicago Data Portal example for reference
start	The year in which the time series data starts. The time series is assumed to be composed of solely monthly count data

Value

Returns an object of class "stl" with the following components:

- time.series: a multiple time series with columns seasonal, trend and remainder.
- weights: the final robust weights (all one if fitting is not done robustly).
- call: the matched call.
- win: integer (length 3 vector) with the spans used for the "s", "t", and "l" smoothers.
- deg: integer (length 3) vector with the polynomial degrees for these smoothers.
- jump: integer (length 3) vector with the 'jumps' (skips) used for these smoothers.
- inner: number of inner iterations

Author(s)

Jamie Spaulding, Keith Morris

Examples

```
#Using provided dataset from Chicago Data Portal:  
data(crimes)  
test <- ts_month_decomp(crimes, 2017)  
plot(test)
```

Index

- * **datasets**
 - crimes, [2](#)
- * **dynamic**
 - kde_int_comp, [5](#)
 - kde_map, [6](#)
- * **hplot**
 - kde_int_comp, [5](#)
 - kde_map, [6](#)
- * **methods**
 - kde_int_comp, [5](#)
 - kde_map, [6](#)
- * **spatial**
 - geocode_address, [3](#)
 - id_repeat, [4](#)
 - kde_int_comp, [5](#)
 - kde_map, [6](#)
 - near_repeat_analysis, [7](#)
 - near_repeat_eval, [8](#)
- * **ts**
 - ts_daily_decomp, [9](#)
 - ts_forecast, [10](#)
 - ts_month_decomp, [11](#)

crimes, [2](#)

geocode_address, [3](#)

id_repeat, [4](#)

kde_int_comp, [5](#)

kde_map, [6](#)

near_repeat_analysis, [7](#)

near_repeat_eval, [8](#)

ts_daily_decomp, [9](#)

ts_forecast, [10](#)

ts_month_decomp, [11](#)