crime-hotspots-uk Documentation

Release unknown

George Sykes

Jun 12, 2021

CONTENTS

Hello, This is the official documentation for the Crime Hotspots UK anausis software.

The software is programed by [Reclaim The Night Leeds] but is open for use by anyone.

CHAPTER

ONE

CONTENTS

1.1 Crime Hotspots UK

A python module that identifies and produces graphs of assult hotspots within the UK.

For full documentation click here

1.1.1 About

This module has been programmed by organisers of Reclaim The Night Leeds. We hope it will be a useful tool for feminist and political activism groups in the UK. We do not necessarily any groups who have used this software.

1.1.2 How to use

View the demo.ipynb Jupyter Notebook file for demonstration of how to use this module. We advise using the anaconda toolkit to install and run Jupyter Notebook.

You can also view the notebook online here however then you cannot change any of the variables.

1.1.3 Contribute

We welcom pull requests, issues and feature suggestions. To colloborate with us on publishing data you have produced using this or any other tool email us at RTNLeeds[at]gmail.com.

1.2 Contributors

- George Sykes el18gs@leeds.ac.uk

1.3 Changelog

1.3.1 Version 0.1.2 Mary Wollstonecraft the Third

Major Features

- [devel~22] Implemented API method for acquiring stop and search data. This will be improved on in later versions with the ability to compute diversity statitistics for the data.
- [devel~5] Added caching capability for all forms of data. This means you only hae to get the data for a reigion and crime type once and then can wor offline with it. Be warned with large requests this could take up a large ammount of space.
- [devel~8] Added function to export data to csv (for general importing) or sav (for importing to SPSS)
- [devel~12] Updated location fixing algorithim to only calculate snap points once. This vastly improves the speed at which the location points can be fixed especially with large datasets.

Minor improvements and bug fixes

- [devel~22] Added complete list of non descriptive place names
- [devel~3] Updated demo document to use new features
- [devel~4] Updated license document to include title block for RST compatibility
- [devel~10] added exceptions for http error codes
- [devel-11] changed date range to be calculated automatically as the past three years
- [devel~13] Remove on or near part from all the place names

Backend and API changes

- [devel~6] Added child classes for different types of data
- [devel~7] Renamed original Reclaim class to Root for easier understanding
- [devel~23] Renamed columns in the constituincy class to follow standard format

1.3.2 Version 0.1.1 Mary Wollstonecraft the seccond

Features

- Can now filter to only show crimes that happened near a certain kind of area
- Added exporting of location data
- Now able to search for constituincies instead of passing their name

Development workflow

- Added precommit hooks functionality, includeing Black and Flake8 code checkers

Backend

- Refactored locations to be their own submodule, in future new location types can be added more easily
- Implemented constituincy location types to use the PyParliment module

1.3.3 Version 0.1.0 Mary Wollstonecraft

- Added documentation for all existing classes
- Ported documentation to read the docs
- Removed fishnet algorithim so all data for each constituincy is got in one batch, paves the way to ensure program works for all constituincies
- Added a constants file to enable easier changing of widely used constant variables
- Minor speed improvements
- Assorted bug fixes

1.4 crime_hotspots_uk

1.4.1 crime_hotspots_uk package

Subpackages

crime_hotspots_uk.locations package

Submodules

crime_hotspots_uk.locations.constituincy module

class crime_hotspots_uk.locations.constituincy.**Constituincy**(*names*, *title*)
Bases: *crime_hotspots_uk.locations.generic.Locations*

This class is used to hold a dataframe of constituincy boundaries and any relevant data. Any data pertaining to a perticular constituincy including demographics or political representation should be implemented here.

__init__(*names*, *title*)
Initialise the class and import data

Parameters

- **names** – A list of strings containing names of constituincies to search for
- **title** (*string*) – A string representing what area the constiuincies represent. For instance *London Mayoral Constituincies*

The init function will get all constituincies with names starting with any of the items in names.

crime_hotspots_uk.locations.generic module

class crime_hotspots_uk.locations.generic.**Locations**(*name*)
Bases: object

A template class that provide a generic structure for how location should be loaded in. The implementations of the functions should be overwritten or extended by each implementation but all implementations must follow this format.

This class is handler to a python dataframe that contains rows of shapely geometries that define individual areas.

__init__(*name*)
This function is called when the class is initialised. It will set variables that will describe the data and then call the import function.

Parameters **name** (*string*) – What the name of collection of locations is called. For instance Leeds constituincies. This should be human readable and provide a concise explanation of what the boundaries represent.

export(*file_name='DEADBEEF'*)
Export the current dataframe of locations to a .csv file. If no filename is passed the file will be exported to a cache directory depending on the name and type of the dataframe it comes from

Parameters **file_name** (*string*) – What to save the file as

get_area(*location_name*)
Returns a list of lists containing lon/lat coordinate pairs defining the boundaries of an area (it is a list of lists as some areas have islands and as such have to be defined by multiple boundaries) that has the name passed to this function

Parameters **location_name** (*string*) – Name of the location to get the data for instance *Leeds Central* for the leed central constituincy.

exception crime_hotspots_uk.locations.generic.**import_not_overwritten**(*message='Import function has not been overwritten'*)
Bases: Exception

Exception raised when the generic import function is called. This normally means we haven't yet finished implementing this particular location type

__init__(*message='Import function has not been overwritten'*)
Initialize self. See help(type(self)) for accurate signature.

exception crime_hotspots_uk.locations.generic.**location_not_found**(*name*, *collection_name*)
Bases: Exception

Exception raised when the class is asked to find a location that is not in its datafreame

Parameters

- **name** (*string*) – Name of the location it was trying to find (For instance *Leeds North West*)
- **collection_name** (*string*) – Name of the class it was searching in (for instance *Leeds*)

__init__(*name*, *collection_name*)
Initialize self. See help(type(self)) for accurate signature.

crime_hotspots_uk.locations.police module

Module contents

Submodules

crime_hotspots_uk.constants module

crime_hotspots_uk.crimes module

class crime_hotspots_uk.crimes.**Data**(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*)

Bases: *crime_hotspots_uk.data.Root*

This class is used to hold a dataframe of constituincy boundaries and any relevant data. Any data pertaining to a perticular constituincy including demographics or political representation should be implemented here.

__init__(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*)

This function initiates the class

It does this by downloading the location boundaries and crime type options from the UK Police API.

Parameters **usage** (*string, optional*) – Wether to get crime data or stop and search data from the police API. If 'crime' is passed the class will use the Street level crimes method, if search is passed it will use the Stop and searches by area method.

Raises **AssertionError** – This error is raised if the string passed to usage is not 'crime' or 'search'.

crime_hotspots_uk.data module

This module is used to download and analyse data from the data.police.uk API.

class crime_hotspots_uk.data.**Root**(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*, *usage='crime'*)

Bases: object

This class handles all downloading and processing of the data.

__init__(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*, *usage='crime'*)

This function initiates the class

It does this by downloading the location boundaries and crime type options from the UK Police API.

Parameters **usage** (*string, optional*) – Wether to get crime data or stop and search data from the police API. If 'crime' is passed the class will use the Street level crimes method, if search is passed it will use the Stop and searches by area method.

Raises **AssertionError** – This error is raised if the string passed to usage is not 'crime' or 'search'.

cache_data()

create_mappings()

export(*name*, *file_type*)

fishnet(*geometry*, *threshold*)
Divide a shapely geometry into small sections

Note: This function is not currently used and is not doccumented

fix_locations()
Fix locations in the self.all_crimes dataframe

This is needed because some of the location names used by the police are used for multiple locations. For instance *On or near bus stop* doesn't tell us which bus stop it was near. This function takes the provided latitude and longitude coordinates and identifies which locale with a definitive name in the local area is closest.

Raises AssertionError – This error is raised if a location name can't be correctly mapped to a street because there was no points close enough.

fix_polygons(*polygon*)

get_crimes(*coords*, *name*)
Download all crimes of a specific type within a boundary

Parameters

- **coords** (*string*) – A two deep list containing latitude and longitude coordinate pairs
- **name** – The name of the area the data is for, this name will be appended as a column to the output dataframe to ensure that each area can be selected individualy

Returns Returns either a pandas dataframe if the data retireval was successfull or NONE if it wasn't

Return type pandas.dataframe

get_data(*crime_type*)
Download data for a specified crime type.

This is also used to download stop and search data. To do so make sure self.usage has been set previously.

Parameters **crime_type** (*string, required*) – The crime type to download the data for. It must be one of the types listed in self.crime_types, it should be the readable name (without any -/_) The full explanation of what each category is can be infered from the *Police website <https://www.police.uk/pu/contact-the-police/what-and-how-to-report/what-report/>_* # noqa e501

Returns Will return true if it managed to successfully download and validate the data. If it fails to it will return false.

Return type bool

hotspots_graph(*top*, *location*, *location_type=['All']*)
Draw a bargraph of the rates of assult at the top hotspots

Parameters

- **top** (*int*) – how many hotspots to plot, for instance 10 would show the top 10 hotspots. IF this is set to none all hotspots will be graphed.
- **location** (*string*) – Where the title of the graph should say the data is from

- **location_type** (*list (optional)*) – Type of location to make the graph for, must be a list of location types, each entry must be either *Street* or value in the ignore list in constants.py. You can also pass *All* to select all crimes. The default value is *All*

import_cache(*location_type*, *area*, *month*, *category=None*)

url_gen(*location*, *date*)

Generate the url for API requests

Parameters

- **location** (*String*) – String of Lat/Lon coordinates marking out a boundary
- **date** – The month to get the data for in format yyyy-mm

Type date String

exception crime_hotspots_uk.data.**http_error_code**(*code*, *url*)

Bases: Exception

Exception raised when a function that should only be run after the crime data location data has been fixed to ensure readable place names are used instead of generic identifiers.

__init__(*code*, *url*)

Initialize self. See help(type(self)) for accurate signature.

exception crime_hotspots_uk.data.**locations_not_fixed_yet**(*message='Locations have not been fixed yet'*)

Bases: Exception

Exception raised when a function that should only be run after the crime data location data has been fixed to ensure readable place names are used instead of generic identifiers.

__init__(*message='Locations have not been fixed yet'*)

Initialize self. See help(type(self)) for accurate signature.

crime_hotspots_uk.searches module

class crime_hotspots_uk.searches.**Data**(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*)

Bases: *crime_hotspots_uk.data.Root*

This class is used to hold a dataframe of constituincy boundaries and any relevant data. Any data pertaining to a perticular constituincy including demographics or political representation should be implemented here.

__init__(*name*, *location_names*, *location_type=<class 'crime_hotspots_uk.locations.constituincy.Constituincy'>*)

This function initiates the class

It does this by downloading the location boundaries and crime type options from the UK Police API.

Parameters **usage** (*string, optional*) – Wether to get crime data or stop and search data from the police API. If 'crime' is passed the class will use the Street level crimes method, if search is passed it will use the Stop and searches by area method.

Raises **AssertionError** – This error is raised if the string passed to usage is not 'crime' or 'search'.

cache_data()

`get_data()`
Download data for a specified crime type.

This is also used to download stop and search data. To do so make sure self.usage has been set previously.

Parameters `crime_type` (*string, required*) – The crime type to download the data for. It must be one of the types listed in self.crime_types, it should be the readable name (without any -/_) The full explanation of what each category is can be infered from the *Police website <https://www.police.uk/pu/contact-the-police/what-and-how-to-report/what-report/>_* # noqa e501

Returns Will return true if it managed to successfully download and validate the data. If it fails to it will return false.

Return type bool

`import_cache`(*location_type*, *area*, *month*, *category=None*)

Module contents

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

[Reclaim The Night Leeds] [Sphinx]: http://www.sphinx-doc.org/ [Markdown]: https://daringfireball.net/projects/markdown/ [reStructuredText]: http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html [recommonmark]: https://recommonmark.readthedocs.io/en/latest [autostructify]: https://recommonmark.readthedocs.io/en/latest/auto_structify.html

PYTHON MODULE INDEX

C

INDEX

Symbols

C

D

E

F

G

H

I

L

M

R

U