Package 'AOI'

August 23, 2018

Type Package

Title Areas of Interest

Version 0.1.9000

Maintainer Mike Johnson <jmj00@ucsb.edu>

BugReports https://github.com/mikejohnson51/AOI/issues

Description An area of interest (AOI) is a geographic extent. AOIs help confine and formalize a unit of work to a geographic area and define research and sub setting efforts while improving reproducibility.
They are built around concrete spatial attributes but often are discussed in a more colloquial way. This package lets users define regions through a common query to achieve spatial geometries.
Tools are provided to help define, describe, and convert points, boundaries, and features to usable forms including strings and political boundaries (for USA states and counties). In addition, this package provides geocoding
and reverse geocoding functions through the Google Maps, OpenStreetMaps and ESRI Webservices. This package is provided to support the bedth of spatial packages in the R ecosystems

Depends R(>= 3.3.0),
leaflet

Imports jsonlite,
magrittr,
sf(>= 0.6-0),
utils,
xml2

Suggests testthat

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

URL https://github.com/mikejohnson51/AOI/

R topics documented:

`aoiProj` *AOI Projection*

Description

The projection used for all AOI calls: *EPSG:4269*

Usage

```
aoiProj
```

Format

An object of class `character` of length 1.

Author(s)

Mike Johnson

`bbox_sp` *Convert bounding box string to geometry*

Description

Convert a vector, dataframe, or bb object to a `SpatialPolygon`

Usage

```
bbox_sp(bbox_st, sf = FALSE)
```

Arguments

`bbox_st`	a bounding box string or vector in the order ("xmin","xmax", "ymin", "ymax")
`sf`	logical. Should returned feature be of class sf (default = FALSE)

Value

a bounding box geometry

Author(s)

Mike Johnson

Examples

```
## Not run:
CO = getAOI(state = 'CO') %>% bbox_st()
CO.1 = CO %>% bbox_sp()

## End(Not run)
```

---

bbox_st *Convert bounding box geometry to string*

Description

Convert an AOI or spatial object to a data.frame of (xmin, xmax, ymin, ymax)

Usage

```
bbox_st(AOI)
```

Arguments

AOI	an AOI obtained using getAOI (or any sp/sf object).

Value

a bounding box data.frame

Author(s)

Mike Johnson

Examples

```
## Not run:
#Get a bounding box data.frame for AOI
   AOI = getAOI(clip = list("UCSB", 10, 10))
   bb = bbox_st(AOI)
print(bb)

# Chain to AOI calls:
AOI = getAOI(clip = list("UCSB", 10, 10)) %>% bbox_st()
print(AOI)

## End(Not run)
```

buffer *Buffer AOI*

Description

buffer add a uniform buffer to an AOI or spatial obejct in either miles or kilometers.

Usage

```
buffer(AOI, d, km = FALSE)
```

Arguments

AOI	a spatial of simple features object
d	the distance to add to each edge
km	is the distance in kilometers? Default is FALSE eg miles

Value

a spatial geometry of the same class as the input AOI

Author(s)

Mike Johnson

Examples

```
## Not run:
 # get an AOI of garden of the gods and add a 2 mile buffer
    getAOI("Garden of the Gods") %>% buffer(10)
 # get an AOI of garden of the gods and add a 2 kilometer buffer
    getAOI("Garden of the Gods") %>% buffer(10, km = TRUE)

## End(Not run)
```

check *Vizualize AOIs*

Description

Generate an interactive leaflet map for defining, checking, and refining AOI queries. Can be chained to getAOI via ‘ an approapriate AOI.

Usage

```
check(AOI = NULL)
```

Arguments

AOI	an AOI obtained using getAOI or any/sf. Can be left NULL

Value

a list of AOI and `leaflet` html object

Author(s)

Mike Johnson

Examples

```
## Not run:
# Generate an empty map:
check()

# Check a defined AOI:
AOI = getAOI(clip = list("UCSB", 10, 10))
check(AOI)

# Chain to AOI calls:
getAOI(clip = list("UCSB", 10, 10)) %>% check()

## End(Not run)
```

`counties` *USA Counites*

Description

Dataset containing `SpatialPolygons` of USA Counties. Data is initalized from the USAboundaries and USAboundariesData package, converted to spatial `sp` objects, and cleaned-up for this package. The primary reason for doing this is to limit the challenges associated with using the USAboundariesData (not on CRAN) as a dependency for this, and other packages, while also providing a more minmualistic dataset.

Usage

```
counties
```

Format

a `SpatialPolygonsDataFrame`, 3220 observations of 7 variables

- 'statefp': A `character` State 2-digit FederalInformationProcessingStandards (FIPS) code
- 'countyfp': A `character` County 3-digit FederalInformationProcessingStandards (FIPS) code
- 'affgeoid': A `character` AFF Summary Level Code
- 'geoid': A `character` Concatinates state and county FIP code
- 'name': A `character` County name
- 'state_name': A `character` State name
- 'state_abbr': A `character` State Abbriviation

Source

USAboundaries

Examples

```
## Not run:
 counties = AOI::counties

## End(Not run)
```

`describe` *Describe an AOI*

Description

Convert an AOI object to a data.frame of describing factors. Can be useful for sharing, documenting and repaeating AOI calls.

Usage

```
describe(AOI)
```

Arguments

`AOI`	an AOI obtained using getAOI, or any sp/sf object.

Value

a data.frame of AOI descriptors including

latCent the AOI center latitude

lngCent the AOI center longitude

height height in (miles)

width width in(miles)

origin AOI origin

name Most descriptive geocoded name from `revgeocode`

Author(s)

Mike Johnson

Examples

```
## Not run:
#Get an AOI
AOI = getAOI(clip = list("UCSB", 10, 10))
describe(AOI)

# Chain to AOI calls:
AOI = getAOI(clip = list("UCSB", 10, 10)) %>% describe()

## End(Not run)
```

geocode *Geocoding*

Description

A wrapper around the Google and OpenSteetMpa geocoding web-services. Users can request a lat/long pair, spatail points with geocoded metadata, and/or a bounding box geometry. A single or mulitple locations can be input. If a single point is given ‘geocode‘ with provide a matix of lat lon a spatial point an the geocode derived bounding box (if requested). If multiple points are given the retured objects will be a matrix with columns for input name:lat:lon; a collection of spatial points; and a minimum bounding box of the input locations.

Usage

```
geocode(location = NULL, pt = FALSE, bb = FALSE, server = "google")
```

Arguments

location	place name(s)
pt	logical. Should the function return a spatial feature(s) of the location
bb	return bb Should a bounding box geometery be returned with the object.
server	what server should be prioritized. Options inlcude "google" or "OSM" (default = ’google)

Value

at minimum a matrix of lat/long coordinates

Author(s)

Mike Johnson

Examples

```
## Not run:
 #request a single location
     geocode("UCSB")
 #request a spatial point of location
     geocode("UCSB", pt = TRUE)
 #request a geocode derived bounding box of location
     geocode("UCSB", bb = TRUE)
 #request multiple locations
     geocode(c("UCSB", "Goleta", "Sterns Warf"))
 #request a minimum bounding box of all requested points
     geocode(c("UCSB", "Goleta", "Sterns Warf"), bb = T, pt= T)

## End(Not run)
```

getAOI *Get Area of Interest (AOI) geometry*

Description

Get a Spatial* representation of an AOI defined by:

1. a US state name(s)
2. a US state, county pair(s)
3. a user Spatial, sf or raster object or
4. a clipping unit (see details)

`getAOI` wraps `getFiat` and `getClip` into a single function.

Usage

```
getAOI(clip = NULL, state = NULL, county = NULL, sf = FALSE,
  km = FALSE, bb = FALSE)
```

Arguments

`clip`	`Spatial` object, a `Raster` object, or a `list` (see details and `getClip`)
`state`	`character`. Full name or two character abbriviation. Not case senstive
`county`	`character`. County name(s). Requires `state` input. Not case senstive
`sf`	`logical`. If `TRUE` object returned is of class sf, default is `FALSE` and returns class SpatialPolygons
`km`	`logical`. If `TRUE` distance are in kilometers, default is `FALSE` and with distances in miles
`bb`	`logical`. If `TRUE` then the bounding geometry of state/county is returned, default is `FALSE` and returns fiat geometries

Details

If `clip` is a list, a clip unit requires a minimum of 3 inputs:

1. A point:
   - 'location name' ex: "UCSB"
   - lat/lon pair: ex: '-36, -120'
2. A bounding box height
   - in miles ex: 10
3. A bounding box width
   - in miles ex: 10

The bounding box is always drawn in relation to the location. By default the point is treated as the center of the box. To define the realtive location of the point to the bounding box a fourth input can be used:

1. Origin

- 'center' (default)
- 'upperleft'
- 'upperright'
- 'lowerleft'
- 'lowerright'

3 to 5 elements can be used to paramaterize the `clip` element but **ORDER MATTERS** (point, height, width, origin). Acceptable variations include:

- 3 members: (1) location name, (2) height, (3) width
  - *list("UCSB", 10, 10)*
- 4 members: (1) latitude, (2) longitude, (3) height, (4) width
  - *list(36, -120, 10, 10)*
- 4 members: (1) location name, (2) height, (3) width, (4) origin
  - *list("UCSB", 10, 10, "lowerright)*
- 5 members: (1) lat, (2) long, (3) height, (4) width, (5) origin
  - *list(36,-120, 10, 10, "upperright)*

Value

a geometry projected to *EPSG:4269*.

Author(s)

Mike Johnson

Examples

```
## Not run:
# Get AOI defined by a state(s)
    getAOI(state = 'CA')
    getAOI(state = c('CA', 'nevada'))

# Get AOI defined by state & county pair(s)
    getAOI(state = 'California', county = 'Santa Barbara')
    getAOI(state = 'CA', county = c('Santa Barbara', 'ventura'))

# Get AOI defined by external spatial file:
    getAOI(clip = rgdal::readOGR('la_metro.shp'))
    getAOI(clip = raster('AOI.tif'))

# Get AOI defined by 10 mile bounding box using lat/lon
    getAOI(clip = c(35, -119, 10, 10))

# Get AOI defined by 10 mile2 bounding box using the 'KMART near UCSB' as lower left corner
    getAOI(clip = list('KMART near UCSB', 10, 10, 'lowerleft'))

## End(Not run)
```

getBoundingBox *Get mimimum bounding box of spatial Objects*

Description

Returns a minimum bounding box for a set or Spatial*, raster or sf object(s)

Usage

```
getBoundingBox(x, sf = FALSE)
```

Arguments

x	a data.frame with a lat and long column, a Raster, sf, or Spatial Objects
sf	logical. If TRUE object returned is of class sf default is FALSE and returns class SpatialPolygon

Author(s)

Mike Johnson

revgeocode *Reverse Geocoding*

Description

Describe a location using the ERSI and OSM reverse geocoding web-services.

Usage

```
revgeocode(point)
```

Arguments

point	a point provided by lat,long or place name

Value

a data.frame of descriptive features

Author(s)

Mike Johnson

Examples

```
## Not run:
pt1 = revgeocode("UCSB")

pt2 = revgeocode(c(38,-115))

## End(Not run)
```

`states` *USA States*

Description

Dataset containing `SpatialPolygons` of USA States. Data is initalized from the USAboundaries and USAboundariesData package, converted to spatial `sp` objects, and cleaned-up for this package. The primary reason for doing this is to limit the challenges associated with using the USAboundariesData (not on CRAN) as a dependency for this, and other packages, while also providing a more minmualistic dataset.

Usage

```
states
```

Format

a `SpatialPolygonsDataFrame`, 52 observations of 5 variables

- 'statefp': A `character` State 2-digit FederalInformationProcessingStandards (FIPS) code
- 'statens': A `character` American National Standards Institute (ANSI) code
- 'affgeoid': A `character` AFF Summary Level Code
- 'state_name': A `character` State Name
- 'state_abbr': A `character` State Abbriviation

Source

USAboundaries

Examples

```
## Not run:
 states = AOI::states

## End(Not run)
```

Index