# Package ‘RCzechia’

May 4, 2020

**Type** Package  
**Title** Spatial Objects of the Czech Republic  
**Version** 1.5.0  
**Date** 2020-05-03  
**Author** Jindra Lacko  
**Maintainer** Jindra Lacko <jindra.lacko@gmail.com>  
**Description** Administrative regions and other spatial objects of the Czech Republic.  
**URL** https://github.com/jlacko/RCzechia  
**BugReports** https://github.com/jlacko/RCzechia/issues  
**License** MIT + file LICENSE  
**Encoding** UTF-8  
**LazyData** true  
**Depends** R (>= 3.3.0), sf  
**Imports** curl, httr, jsonlite, magrittr, dplyr, rlang  
**Suggests** raster, readxl, units, ggplot2, scales, knitr, testthat, roxygen2, devtools, rmarkdown, lwgeom  
**RoxygenNote** 7.1.0  
**VignetteBuilder** knitr  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2020-05-04 11:40:12 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCzechia-package</td>
<td>2</td>
</tr>
<tr>
<td>casti</td>
<td>3</td>
</tr>
<tr>
<td>chr_uzemi</td>
<td>4</td>
</tr>
<tr>
<td>geocode</td>
<td>5</td>
</tr>
<tr>
<td>KFME_grid</td>
<td>6</td>
</tr>
</tbody>
</table>
RCzechia-package

Description

A selection of spatial objects relevant to the Czech Republic. Due to CRAN package size requirements (5 MB) the objects are stored externally (on Amazon S3) - and therefore could not be implemented as datasets. They are functions returning data frames instead.

Details

To save time (and bandwidth) the downloaded objects are saved locally in ‘tempdir’ directory when requested, and downloaded at most once per R session; out of respect to CRAN Repository Policy a more permanent caching on user’s side is not attempted.

This means that:

• a working internet connection is required to use the full resolution objects
• all objects need to be called with (possibly empty) braces

For the most frequently used objects - republika, kraje and okresy - a low resolution version is also implemented. The low resolution data sets are stored locally (and working internet connection is not necessary to use them).

All objects are implemented as sf data frames.

Administrative regions

• republika - borders of the Czech Republic
• kraje - regions / NUTS3 units
• okresy - districts / LAU1 units
• orp_polygony - municipalities with extended powers (obce s rozšířenou působností)
• obce_polygony - municipalities as polygons
• obce_body - municipalities as centroids (points)
• casti - city districts (for cities that implement them)

Natural objects
• reky - rivers
• plochy - water bodies
• lesy - woodland areas (more than 30 ha in area)

Other objects
• silnice - roads
• zeleznice - railroads
• KFME_grid - grid cells (faunistické čtverce) according to Kartierung der Flora Mitteleuropas methodology

Utility functions
In addition three utility functions are implemented to support spatial workflow:
• union_sf - merging polygons based on a key value
• geocode - geocoding (from address to coordinates)
• revgeo - reverse gecoding (from coordinates to address)

---

casti | City Parts
---

Description
Function taking no parameters and returning data frame of districts of Prague and other major cities as sf polygons.

Usage
casti()

Format
sf data frame with 142 rows of 4 variables + geometry

KOD  Code of the city part / kod mestske casti
NAZEV  Name of the city part / nazev mestske casti
KOD_OBEC  Code of the city
NAZ_OBEC  Name of the city
Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 593.6 KB.

Source


chr_uzemi

Protected Natural Areas

Description

Function returning data frame of protected natural areas (Chráněná území) of the Czech Republic as sf polygons. It has no obligatory parameters.

Usage

chr_uzemi()

Format

sf data frame with 36 rows of 2 variables + geometry

TYP  Type of protected area: Národní park, Chráněná krajinná oblast
NAZEV Name, with Czech accents

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 114 KB.

Source

**Geocode a Czech Address**

**Description**
This function connects to Czech State Administration of Land Surveying and Cadastre (https://www.cuzk.cz/en) API to geocode an address. As consequence it is implemented only for Czech addresses.

**Usage**

```r
gencode(address, crs = 4326)
```

**Arguments**

- `address`: point to be geocoded, as character (vector)
- `crs`: coordinate reference system of output; default = WGS84

**Format**

- `sf` data frame with 3 variables + geometry
  - `target`: the address searched (address input)
  - `typ`: type of record matched by API
  - `address`: address as recorded by RÚIAN
  - `geometry`: hidden column with spatial point data

**Details**

Input of the function are an address to geocode (or a vector of addresses) and expected Coordinate Reference System of output (default is WGS84 = EPSG:4326, but in some use cases inž. Křovák = EPSG:5514 may be more relevant).

Output is a `sf` data frame of spatial points.

Depending on the outcome of matching the address to RÚIAN data there is a number of possible outcomes:

- All items were *matched exactly*: the returned `sf` data frame has the same number of rows as there were elements in vector to be geocoded. The field `target` will have zero duplicates.
- Some items had *multiple matches*: the returned `sf` data frame has more rows than the there were elements in vector to be geocoded. In the field `target` will be duplicate values. Note that the RÚIAN API limits multiple matches to 10.
- Some (but not all) items had *no match* in RÚIAN data: the returned `sf` data frame will have fewer rows than the vector sent. to be geocoded elements. Some values will be missing from field `target`.
- No items were matched at all: the function returns NA.
Note that character encoding is heavily platform dependent, and you may need to convert to UTF-8, e.g. by running `address <- iconv(address, from = "windows-1250", to = "UTF-8")` before calling the function.


**Examples**

```r
asdf <- geocode("Gogolova 212, Praha 1")
print(asdf)
```

---

**KFME_grid**  
**KFME grid cells (faunistické čtverce) of the Czech Republic**

---

**Description**

Function returning grid covering the Czech Republic according to the Kartierung der Flora Mitteleuropas methodology.

**Usage**

```r
KFME_grid(resolution = "low")
```

**Arguments**

- `resolution` Should the function return high or low resolution shapefile? Allowed values are "low" and "high". Default is "low".

**Format**

- `sf` data frame with 1092 rows in low resolution and 4368 rows in high resolution
- `ctverec` KFME code of the grid cell; depending on value of 'resolution' parameter either 4 digits, or 4 digits + 1 letter

**Details**

The function returns a `sf` data frame of grid cells. Depending on the value of parameter ‘resolution’ either low resolution (26×42 cells - default) with labels in 4 digit format (e.g. Hrčava = 6479) or high resolution (104×168 cells) with labels in 4 digit + 1 letter format (e.g Hrčava = 6479c).

**Examples**

```r
library(ggplot2)

ggplot() +
  geom_sf(data = republika("low")) +
  geom_sf(data = KFME_grid("low"), fill = NA)
```
Description

Function returning data frame of NUTS3 administrative units for the Czech Republic as sf polygons. It takes a single parameter resolution - high res (default) or low res polygons.

Usage

kraje(resolution = "high")

Arguments

resolution Should the function return high or low resolution shapefile? Allowed values are "high" (default) and "low". This parameter affects only the geometry column, all other fields remain the same.

Format

sf data frame with 14 rows of 3 variables + geometry

KOD_KRAJ Code of the region, primary key. Use this as key to add other data items.
KOD_CZNUTS3 Code of the region as NUTS3 (kraj).
NAZ_CZNUTS3 Name of the region as NUTS3 (kraj).

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size of high resolution shapefile is <1 MB.

Source

https://vdp.cuzk.cz/

Examples

library(sf)

hranice <- kraje()  
plot(hranice, col = "white", max.plot = 1)
### lesy

**Woodland Areas**

**Description**
Function returning data frame of woodland areas (lesy) of more than 30 hectares in area of the Czech Republic as sf polygons. It has no obligatory parameters.

**Usage**
```
lesy()
```

**Format**
- `sf` data frame with 2.366 rows of geometry variable only

**Details**
Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 2.1 MB.

**Source**

### obce_body

**Municipalities / communes (obce) as centerpoints**

**Description**
Function returning data frame of LAU2 administrative units for the Czech Republic as sf points. It takes no parameters.

**Usage**
```
obce_body()
obce_body()
```

**Format**
- `sf` data frame with 6.258 rows of 14 variables + geometry

- `KOD_OBEC` Code of the level I commune (obec).
- `NAZ_OBEC` Name of the level I commune (obec).
- `KOD_ZUJ` Code of the basic administrative unit (ICZUJ).
- `NAZ_ZUJ` Name of the basic administrative unit (ICZUJ).
KOD_POU  Code of the level II commune (obec s poverenym uradem).
NAZ_POU  Name of the level II commune (obec s poverenym uradem)).
KOD_ORP  Code of the level III commune (obec s rozsirenou pusobnosti).
NAZ_ORP  Name of the level III commune (obec s rozsirenou pusobnosti).
KOD_OKRES  Code of the district (okres).
KOD_LAU1  Code of the LAU1 administrative unit (okres).
NAZ_LAU1  Name of the LAU1 administrative unit (okres).
KOD_KRAJ  Code of the region (kraj).
KOD_CZNUTS3  Code of the NUTS3 unit (kraj)
NAZ_CZNUTS3  Name of the NUTS3 unit (kraj)

Details
Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 270 KB.

Source
https://vdp.cuzk.cz/

Description
Function returning data frame of LAU2 administrative units for the Czech Republic as sf polygons. It takes no parameters.

Usage
obce_polygony()

Format
sf data frame with 6.258 rows of 14 variables + geometry

KOD_OBEC  Code of the level I commune (obec).
NAZ_OBEC  Name of the level I commune (obec).
KOD_ZUJ  Code of the basic administrative unit (ICZUJ).
NAZ_ZUJ  Name of the basic administrative unit (ICZUJ).
KOD_POU  Code of the level II commune (obec s poverenym uradem).
NAZ_POU  Name of the level II commune (obec s poverenym uradem)).
KOD_ORP  Code of the level III commune (obec s rozsirenou pusobnosti).
**NAZ_OKR** Name of the level III commune (obec s rozsirenovou psoobnosti).

**KOD_OKRES** Code of the district (okres).

**KOD_LAU1** Code of the LAU1 administrative unit (okres).

**NAZ_LAU1** Name of the LAU1 administrative unit (okres).

**KOD_KRAJ** Code of the region (kraj).

**KOD_CZNUTS3** Code of the NUTS3 unit (kraj)

**NAZ_CZNUTS3** Name of the NUTS3 unit (kraj)

### Details
Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 13.3 MB (so use with caution, and patience).

### Source
https://vdp.cuzk.cz/

### Examples
```r
library(sf)
library(dplyr)

praha <- obce_polygony() %>%
  filter(NAZ_LAU1 == "Praha")

plot(praha, max.plot = 1)
```

---

### Okresy

**Districts (okresy)**

### Description
Function returning data frame of LAU1 administrative units for the Czech Republic as `sf` polygons. It takes a single parameter resolution - high res (default) or low res polygons.

### Usage
```r
okresy(resolution = "high")
```

### Arguments
```
resolution Should the function return high or low resolution shapefile? Allowed values are "high" (default) and "low". This parameter affects only the geometry column, all other fields remain the same.
```
Format

sf data frame with 77 rows of 6 variables + geometry

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KOD_OKRES</td>
<td>Code of the district (okres).</td>
</tr>
<tr>
<td>KOD_LAU1</td>
<td>Code of the district as LAU1 unit (okres), primary key. Use this as key to add other data items.</td>
</tr>
<tr>
<td>NAZ_LAU1</td>
<td>Name of the district as LAU1 unit (okres).</td>
</tr>
<tr>
<td>KOD_KRAJ</td>
<td>Code of the region.</td>
</tr>
<tr>
<td>KOD_CZNUTS3</td>
<td>Code of the region as NUTS3 (kraj).</td>
</tr>
<tr>
<td>NAZ_CZNUTS3</td>
<td>Name of the region (kraj).</td>
</tr>
</tbody>
</table>

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size of high resolution shapefile is 1 MB (so use with caution, and patience).

Source

https://vdp.cuzk.cz/

Examples

library(sf)

hranice <- okresy()
plot(hranice, col = "white", max.plot = 1)

object.size(okresy("low"))
object.size(okresy("high"))

orp_polygony  

Obce s rozsirenom pousobnosti

Description

Function returning data frame of municipalities with extended powers (obce s rozsirenou pousobností) as sf polygons. It takes no parameters.

Usage

orp_polygony()
Format

sf data frame with 206 rows of 10 variables + geometry

**KOD_ORP** Code of the level III commune (obec s rozsirenou pusbobnosti).

**NAZ_ZKR_ORP** Short name of the level III commune (obec s rozsirenou pusbobnosti).

**NAZ_ORP** Full name of the level III commune (obec s rozsirenou pusbobnosti).

**KOD_RUIAN** RUIAN (Registro uzemni identifikace, adres a nemovitost) code.

**KOD_OKRES** Code of the district (okres).

**KOD_LAU1** Code of the LAU1 administrative unit (okres).

**NAZ_LAU1** Name of the LAU1 administrative unit (okres).

**KOD_KRAJ** Code of the region (kraj).

**KOD_KRAJ** Code of the region (kraj).

**KOD_CZNUTS2** Code of the NUTS3 unit (kraj)

**NAZ_CZNUTS2** Name of the NUTS3 unit (kraj)

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 1.2 MB (so use with caution, and patience).

Source

https://vdp.cuzk.cz/

---

**plochy**  |  *Water Bodies*

Description

Function returning data frame of water bodies of the Czech Republic as sf polygons. It takes no parameters.

Usage

plochy()

Format

sf data frame with 480 rows of 5 variables + geometry

**TYP** Type of water body: 1 = dam, 2 = pond, 3 = lake

**NAZEV** Name, with Czech accents

**NAZEV_ASCII** Name, without Czech accents

**VYSKA** water level, meters above sea level

**Major** Boolean indicating major water bodies
Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 118.6 KB.

Source


reky Riverside

Description

Function returning data frame of rivers of the Czech Republic as sf lines. It takes a single parameter scope with default ”global”.

Usage

reky(scope = ”global”)

Arguments

scope Should the function return all rivers, or just Vltava in Prague / Svitava & Svratka in Brno?

Format

sf data frame with 6.198 rows of 4 variables + geometry:

TYP Type of river: 1 = natural, 2 = man-made, 3 = fictional
NAZEV Name, with Czech accents
NAZEV_ASCII Name, without Czech accents
Major Boolean indicating one of the major rivers.

Details

Two special case scopes are defined: Praha (returning the part of Vltava in and around Prague) and Brno (returning Svitava and Svratka near and around Brno).

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 1 MB.

Source

Examples

library(sf)

plot(st_geometry(subset(okresy(), KOD_LAU1 == "CZ0642"))) # Brno city
plot(reky("Brno"), add = TRUE) # Svitava & Svratka added to Brno my city plot

---

republika  

Republika

Description

Boundaries of the Czech Republic as sf polygon.

Usage

republika(resolution = "high")

Arguments

resolution Should the function return high or low resolution shapefile? Allowed values are "high" (default) and "low". This parameter affects only the geometry column, all other fields remain the same.

Format

sf data frame with 1 row of 1 variable + geometry:

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size of high resolution shapefile is <1 MB.

Source

https://vdp.cuzk.cz/

Examples

library(sf)

hranice <- republika()
plot(hranice, col = "white")
**revgeo**

*Reversely Geocode a Czech Address*

**Description**

This function connects to Czech State Administration of Land Surveying and Cadastre ([https://www.cuzk.cz/en](https://www.cuzk.cz/en)) API to reversely geocode an address. As consequence it is implemented only for Czech addresses.

**Usage**

```r
textwrap::text_justify(revgeo(coords))
```

**Arguments**

- **coords** coordinates to be reverse geocoded; expected as `sf` data frame of spatial points

**Details**

Input of the function is a `sf` data frame of spatial points, and output a vector of characters.

The function returns the same `sf` data frame as input, with added field `revgeocoded`; it contains the result of operation. If the data frame contained a column named `revgeocoded` it gets overwritten.

In case of reverse geocoding failures (e.g. coordinates outside of the Czech Republic and therefore scope of ČÚZK) `NA` is returned.


**Examples**

```r
textwrap::text_justify(library(dplyr))
library(sf)

brno <- obce_polygony() # shapefile of Brno
filter(NAZ_OBEC == "Brno") #
   st_transform(5514) # planar CRS (eastings & northings)

pupek_brna <- st_centroid(brno) # calculate centroid

adresa_pupku <- revgeo(pupek_brna)$revgeocoded # address of the center
```
silnica

Road Network

Description

Function returning data frame of roads of the Czech Republic as sf lines. It has no obligatory parameters.

Usage

silnica()

Format

sf data frame with 18,979 rows of 4 variables + geometry:

TRIDA Class of the road: highway = dálnice, speedway = rychlostní silnice, 1st clas road = silnice I. třídy, 2nd class road = silnice II. třídy, 3rd class road = silnice III. třídy, other road = nevidovaná silnice

CISLO_SILNICE Local road code

MEZINARODNI_OZNACENI International road code

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 1.5 MB.

Source


union_sf

Aggregate Polygons in a sf Object

Description

The function aggregates polygons of geometry column of a sf data frame according to values of a single data column. It has outcome comparable to unionSpatialPolygons from maptools package, except that it works on sf and not sp objects.

Usage

union_sf(data, key, tolerance = 1, planar_CRS = 5514)
Arguments

data sf data frame to be aggregated
key name of a single column to define the output objects
tolerance buffer size for avoiding artefacts (slivers); default is one meter
planar_CRS planar CRS for avoiding artefacts (slivers); default is EPSG:5514 = ing. Křovák

Details

The function has data frame as the first argument, so it is pipe friendly. It retains only geometry and key value, dropping all other columns (they are easy to re-attach using tidyverse/dplyr workflow if required).

During processing the sf data frame is temporarily transformed to planar coordinates and - to avoid artefacts (slivers) at the place of former boundaries - buffered; the default values of planar_CRS and tolerance should cover most situations in the Czech Republic and near abroad.

Examples

# library(sf)
NUTS3 <- union_sf(okresy(), "KOD_CZNUTS3")
# assembles NUTS3 regions from LAU1 regions of Czech Republic = equivalent to kraje() in geometry
plot(NUTS3)

vyskopis Vyskopis

Description

Terrain of the Czech Republic as a raster package object.

Usage

vyskopis(format = "rayshaded")

Arguments

format Should the function return actual relief (meters above sea level) or shaded relief (rayshaded). Allowed values are "actual" and "rayshaded".

Format

raster package RasterLayer.
Details

The function returns a raster file of either actual relief (values are meters above sea level) or rayshaded relief (created via highly recommended rayshader package).

The raster is 5084 by 3403 cells, meaning each pixel is about 90 × 90 meters. It works the best at level of country or regions, at the level of a city or lower it may be somewhat grainy.

Due to package size constraints both versions are stored externally (and a working internet connection is required to use the package). Downloaded size of the rayshaded raster is 8.4 MB, actual raster is 31.4 MB.

Source


Examples

library(raster)
relief <- vyskopis("rayshaded")
plot(relief, col = gray.colors(16))

zeleznice

Railroad Network

Description

Function returning data frame of railroads of the Czech Republic as sf lines. It has no obligatory parameters.

Usage

zeleznice()

Format

sf data frame with 3.525 rows of 4 variables + geometry:

**ELEKTRIFIKACE** is the railroad electrified?: yes = ano, no = ne

**KATEGORIE** category: international = mezinárodní, local = vnitrostátní

**KOLEJNOST** track: single = jednokolejní, double = dvojkolejní, more = tří a vícekolejní

**ROZCHODNOST** gauge: standard = normální, narrow = úzkokolejka

Details

Due to package size constraints the data are stored externally (and a working internet connection is required to use the package). Downloaded size is 285 KB.
zeleznice

Source

Index

casti, 3
chr_uzemi, 4
geocode, 5
KFME_grid, 6
kraje, 7
lesy, 8
obce_body, 8
obce_polygony, 9
okresy, 10
orp_polygony, 11
plochy, 12
RCzechia-package, 2
reky, 13
republika, 14
revgeo, 15
silnice, 16
union_sf, 16
vyskopis, 17
zeleznice, 18