

Package ‘osmclass’

May 19, 2026

Type Package

Title Classify Open Street Map Features

Version 0.1.5

Description Classify Open Street Map (OSM) features into meaningful functional or analytical categories. Designed for OSM PBF files, e.g. from <https://download.geofabrik.de/> imported as spatial data frames. A classification consists of a list of categories that are related to certain OSM tags and values. Given a layer from an OSM PBF file and a classification, the main `osm_classify()` function returns a classification data table giving, for each feature, the primary and alternative categories (if there is overlap) assigned, and the tag(s) and value(s) matched on. The package also contains a classification of OSM features by economic function/significance, following Krantz (2023) <https://www.ssrn.com/abstract=4537867>.

License GPL-3

Encoding UTF-8

Depends R (>= 3.3.0)

Imports collapse (>= 1.9.6), data.table, stringi

RoxygenNote 7.3.3

LazyData true

URL <https://sebkrantz.github.io/osmclass/>

NeedsCompilation no

Author Sebastian Krantz [aut, cre]

Maintainer Sebastian Krantz <sebastian.krantz@graduateinstitute.ch>

Repository CRAN

Date/Publication 2026-05-19 21:20:07 UTC

Contents

osmclass-package	2
classifications	4
djibouti_points	5

osm_classify	6
osm_other_tags_list	7
osm_other_tags_str	8
osm_tags_df	9

Index	11
--------------	-----------

osmclass-package	<i>Classify Open Street Map Features</i>
------------------	--

Description

An R package to classify Open Street Map (OSM) features into meaningful functional or analytical categories. It expects OSM PBF data, e.g. from <https://download.geofabrik.de/>, imported as data frames (e.g. using *sf*), and is well optimized to deal with large quantities of OSM data.

Functions

Main Function to Classify OSM Features

[osm_classify\(\)](#)

Auxiliary Functions to Extract Information (Tags) from OSM PBF Layers

[osm_other_tags_list\(\)](#)

[osm_other_tags_str\(\)](#)

[osm_tags_df\(\)](#)

Classifications

A Classification of OSM Features by Economic Function, developed for the Africa OSM following Krantz (2023)

[osm_point_polygon_class](#)

[osm_line_class](#)

[osm_line_info_tags](#)

References

Krantz, Sebastian, Mapping Africa's Infrastructure Potential with Geospatial Big Data, Causal ML, and XAI (August 10, 2023). Available at SSRN: <https://www.ssrn.com/abstract=4537867>

Examples

```

## Not run:
# Download OSM PBF file for Djibouti
download.file("https://download.geofabrik.de/africa/djibouti-latest.osm.pbf",
             destfile = "djibouti-latest.osm.pbf", mode = "wb")

# Import OSM data for Djibouti
library(sf)
st_layers("djibouti-latest.osm.pbf")
points <- st_read("djibouti-latest.osm.pbf", "points")
lines <- st_read("djibouti-latest.osm.pbf", "lines")
polygons <- st_read("djibouti-latest.osm.pbf", "multipolygons")

# Classify features
library(osmclass)
points_class <- osm_classify(points, osm_point_polygon_class)
polygons_class <- osm_classify(polygons, osm_point_polygon_class)
lines_class <- osm_classify(lines, osm_line_class)

# See what proportion of the data we have classified
sum(points_class$classified)/nrow(points)
sum(polygons_class$classified)/nrow(polygons)
sum(lines_class$classified)/nrow(lines)

# Get some additional info for lines
library(collapse)
lines_info <- lines |> ss(lines_class$classified) |>
  rsplit(lines_class$main_cat[lines_class$classified]) |>
  get_vars(names(osm_line_info_tags), regex = TRUE)

lines_info <- Map(osm_tags_df, lines_info, osm_line_info_tags[names(lines_info)])
str(lines_info)

# Get 'other_tags' of points layer as list
other_point_tags <- osm_other_tags_list(points$other_tags, values = TRUE)
str(other_point_tags)

# TIP: For larger OSM files, importing layers (esp. lines and polygons) at once
# may not be feasible memory-wise. In this case, translating to GPKG and using
# an SQL query for stepwise processing is helpful:

library(fastverse)
library(sf)

# Get all Africa OSM (6 Gb)
opt <- options(timeout = 6000)
download.file("https://download.geofabrik.de/africa-latest.osm.pbf",
             destfile = "africa-latest.osm.pbf", mode = "wb")

# GPKG is large (> 40 Gb)

```

```

gdal_utils("vectortranslate", "africa-latest.osm.pbf", "africa-latest.gpkg")

# Get map layers: shows how many features per layer
layers <- st_layers("africa-latest.gpkg")
print(layers)

# Example: stepwise classifying lines, 1M features at a time
N <- layers$features[layers$name == "lines"]
int <- seq(0L, N, 1e6L)
lines_class <- vector("list", length(int))

for (i in seq_len(length(int))) {
  cat("\nReading Lines Chunk:", i, "\n")
  temp = st_read("africa-latest.gpkg",
                query = paste("SELECT * FROM lines LIMIT 1000000 OFFSET", int[i]))
  # Some pre-selection: removing residential roads
  temp %<>% fsubset(is.na(highway) | highway %chin% osm_line_class$road$highway)
  # Classifying
  temp_class <- osm_classify(temp, osm_line_class)
  lines_class[[i]] <- ss(temp_class, temp_class$classified, check = FALSE)
}

# Combining
lines_class <- rbindlist(lines_class)
options(opt)

## End(Not run)

```

classifications

Classification of OSM Features by Economic Function

Description

A classification of OSM features by economic function. A newer detailed version is available for points and polygons.

Usage

```
osm_point_polygon_class
```

```
osm_point_polygon_class_det
```

```
osm_line_class
```

```
osm_line_info_tags
```

Format

An object of class `list` of length 34.

An object of class `list` of length 55.

An object of class `list` of length 11.

An object of class `list` of length 11.

Details

The simple classification, developed for Krantz (2023), aims to classify OSM features into meaningful and specific economic categories such as 'education', 'health', 'tourism', 'financial', 'shopping', 'transport', 'communications', 'industrial', 'residential', 'road', 'railway', 'pipeline', 'power', 'waterway' etc. Separate classifications are developed for points and polygons (buildings) (33 categories), and lines (11 categories), which should be applied to the respective layers of OSM PBF files, see [osmclass-package](#) for an example. The classification is optimized (in terms of tag choice and order of categories) to assign the most sensible primary category to most features in the Africa OSM.

In spring 2025, I added a more detailed classification with 55 categories (`osm_point_polygon_class_det`) for point and polygon-based features, approximately reflecting an extended database used in the November 2024 update of the article where OSM features are combined with other geospatial data sources.

References

Krantz, Sebastian, Mapping Africa's Infrastructure Potential with Geospatial Big Data, Causal ML, and XAI (August 10, 2023). Available at SSRN: <https://www.ssrn.com/abstract=4537867>

See Also

[osmclass-package](#)

Examples

```
collapse::unlist2d(osm_point_polygon_class, idcols = c("category", "tag"))
collapse::unlist2d(osm_line_class, idcols = c("category", "tag"))
# This list contains additional tags with information about lines (e.g. roads and railways)
collapse::unlist2d(osm_line_info_tags, idcols = c("category", "tag"))
```

`djibouti_points`

OSM Points Layer for Djibouti, August 2023

Description

A data table of all 8608 OSM points in Djibouti as of August 2023.

Usage

```
djibouti_points
```

Format

A data table with 8608 rows and 10 columns. The first column contains the OSM id of each point. Other columns give the values of frequent OSM tags for point features. The last column is called 'other_tags' and contains all remaining (less frequent) tags. Please consult the [OSM Feature Documentation](#) for the exact meaning and frequently used values of these tags.

Source

Geofabrik download server (<https://download.geofabrik.de/>). See [osmclass-package](#) for how to download it.

See Also

[osmclass-package](#)

Examples

```
data(djibouti_points)
```

osm_classify

Classify OSM Features

Description

Classifies OSM features into meaningful functional or analytical categories, according to a supplied classification.

Usage

```
osm_classify(data, classification)
```

Arguments

data imported layer from an OSM PBF file. Usually an 'sf' data frame, but the geometry column is unnecessary. Importantly, the data frame should have an 'other_tags' column with OSM PBF formatting.

classification a 2-level nested list providing a classification. The layers of the list are:

categories a list of tags and matched values that constitute a feature category.

tags a character vector of tag values to match, or "" to match all possible values. It is also possible to match all ex...

See [osm_point_polygon_class](#) and [osm_line_class](#) for example classifications.

Value

a *data.table* with rows matching the input frame and columns

classified	logical. Whether the feature was classified i.e. matched by any tag-value in the classification.
main_cat	character. The first category the feature was assigned to, depending on the order of categories in the classification.
main_tag	character. The tag matched for the main category.
main_tag_value	character. The value matched on.
alt_cats	character. Alternative (secondary) categories assigned, comma-separated if multiple.
alt_tags_values	character. The tags and double-quoted values matched for secondary categories, comma-separated if multiple.

Note

It is not necessary to expand the 'other_tags' column, e.g. using `osm_tags_df()`. `osm_classify()` efficiently searches the content of that column without expanding it.

See Also

[osmclass-package](#)

Examples

```
# See Examples at ?osmclass for a full examples

# Classify OSM Points in Djibouti
djibouti_points_class <- osm_classify(djibouti_points, osm_point_polygon_class)
head(djibouti_points_class)
collapse::descr(djibouti_points_class)
```

osm_other_tags_list *Generate a List from the 'other_tags' Column in OSM PBF Data*

Description

Generate a List from the 'other_tags' Column in OSM PBF Data

Usage

```
osm_other_tags_list(x, values = FALSE, split = "\",\"|\"=>\"", ...)
```

Arguments

`x` character. The 'other_tags' column of an imported osm.pbf file.
`values` logical. TRUE also includes the values of tags.
`split` character. Pattern passed to `strsplit` to split up `x`.
`...` further arguments to `strsplit`.

Value

a list of tags as character vectors, or a nested list of tags and values if `values = TRUE`.

See Also

[osmclass-package](#)

Examples

```
# See Examples at ?osmclass for full examples

# Extract 'other_tags' as list
other_tags <- osm_other_tags_list(djibouti_points$other_tags)
other_tags[1:10]

# Count frequency (showing top 10)
sort(table(unlist(other_tags)), decreasing = TRUE)[1:10]

# Also include values
other_tags_values <- osm_other_tags_list(djibouti_points$other_tags, values = TRUE)
other_tags_values[1:10]
```

`osm_other_tags_str` *Extract Remaining 'other_tags' as Comma-Separated Strings*

Description

Parses the 'other_tags' column from an imported OSM PBF file and reformats it as a comma-separated key: "value" string per feature, optionally dropping a set of tag keys. The intended use is to preserve OSM tags not already captured by `osm_classify()`: pass the classification's tag keys via `exclude` so they are not duplicated in `main_tag`, `main_tag_value`, and `alt_tags_values`.

Usage

```
osm_other_tags_str(x, exclude = character(0), split = "\\", "\\|\\\"=>\\\"", ...)
```

Arguments

x	character. The 'other_tags' column of an imported osm.pbf file.
exclude	character. Tag keys to drop from the output. Default character(0).
split	character. Pattern passed to strsplit to split up x.
...	further arguments to strsplit .

Value

a character vector of the same length as x, with each element a comma-separated string of key: "value" pairs, or NA_character_ if no tags remain after exclusion (or the input was missing/empty).

See Also

[osm_other_tags_list](#), [osm_classify](#), [osmclass-package](#)

Examples

```
# See Examples at ?osmclass for full examples

# Reformat all other_tags as key:"value" strings
head(osm_other_tags_str(djibouti_points$other_tags))

# Drop tag keys used by the classification (already captured in
# 'main_tag'/'alt_tags_values' returned by osm_classify())
used_tags <- unique(unlist(lapply(osm_point_polygon_class, names)))
head(osm_other_tags_str(djibouti_points$other_tags, exclude = used_tags))
```

osm_tags_df

Extract Tags as Columns from an OSM PBF Layer

Description

Extract Tags as Columns from an OSM PBF Layer

Usage

```
osm_tags_df(data, tags, na.prop = 0)
```

Arguments

data	an imported layer from an OSM PBF file. Usually has a few important tags already expanded as columns, and an 'other_tags' column which compounds less frequent tags as character strings.
tags	character. A vector of tags to extract as columns.
na.prop	double. Proportion of features having a tag in order to keep the column.

Value

a *data.table* with the supplied tags as columns, and the same number of rows as the input frame.

See Also

[osmclass-package](#)

Examples

```
# See Examples at ?osmclass for full examples

# Extracting tags of interest (some of which are inside 'other_tags')
tags <- c("osm_id", "highway", "man_made", "name", "alt_name",
         "description", "wikidata", "amenity", "tourism")
head(osm_tags_df(djibouti_points, tags))

# Only keeping tags with at least 5% non-missing
head(osm_tags_df(djibouti_points, tags, na.prop = 0.05))
```

Index

- * **datasets**
 - classifications, [4](#)
 - djibouti_points, [5](#)

- classifications, [4](#)

- djibouti_points, [5](#)

- osm_classify, [6, 9](#)
- osm_classify(), [2, 8](#)
- osm_line_class, [2, 6](#)
- osm_line_class(classifications), [4](#)
- osm_line_info_tags, [2](#)
- osm_line_info_tags(classifications), [4](#)
- osm_other_tags_list, [7, 9](#)
- osm_other_tags_list(), [2](#)
- osm_other_tags_str, [8](#)
- osm_other_tags_str(), [2](#)
- osm_point_polygon_class, [2, 6](#)
- osm_point_polygon_class
 (classifications), [4](#)
- osm_point_polygon_class_det
 (classifications), [4](#)
- osm_tags_df, [9](#)
- osm_tags_df(), [2, 7](#)
- osmclass (osmclass-package), [2](#)
- osmclass-package, [2, 5–10](#)

- strsplit, [8, 9](#)