

Package ‘mongolmaps’

October 25, 2026

Title Maps and Administrative Boundaries of Mongolia and Ulaanbaatar

Version 0.2.0

Description Ready-to-use 'sf' maps of Mongolia at every openly available administrative level: country, economic regions, aimags (provinces), soums and Ulaanbaatar districts, and Ulaanbaatar khoroots (subdistricts), with codes and names for every bag. Provides a crosswalk between English and Cyrillic names, common spelling variants, ISO 3166-2 codes, humanitarian P-codes and National Statistics Office codes; helpers to join statistics to maps; thematic layers (settlements, rivers, lakes, roads, railways, airports, elevation, land cover, population and protected areas); and quick 'ggplot2' and 'leaflet' maps. Boundaries derive from the Common Operational Dataset published by the National Statistics Office of Mongolia and the United Nations Office for the Coordination of Humanitarian Affairs <<https://data.humdata.org/dataset/cod-ab-mng>>.

License MIT + file LICENSE

Copyright See file inst/COPYRIGHTS for data sources and their licences.

URL <https://github.com/temuulene/mongolmaps>,
<https://temuulene.github.io/mongolmaps/>

BugReports <https://github.com/temuulene/mongolmaps/issues>

Depends R (>= 4.1.0)

Imports cli (>= 3.6.0), curl, dplyr (>= 1.1.1), grDevices, httr2 (>= 1.0.0), openssl, purrr (>= 1.0.0), rlang (>= 1.1.0), sf (>= 1.0-12), stats, stringi, tibble, tools, utils

Suggests cartogram, covr, geofacet, ggplot2 (>= 3.5.0), ggrepel, knitr, leaflet, mongolstats, patchwork, rmarkdown, s2, scales, terra (>= 1.7-0), testthat (>= 3.2.0), vdiff, withr, zip

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 8.0.0

NeedsCompilation no

Author Temuulen Enebish [aut, cre, cph]

Maintainer Temuulen Enebish <temuulen@gmail.com>

Repository CRAN

Date/Publication 2026-10-25 14:20:09 UTC

Contents

mn_admin	3
mn_aimags	4
mn_aimag_grid	5
mn_bags	6
mn_cache_dir	8
mn_codes	9
mn_country	10
mn_crs	11
mn_elevation	12
mn_erase_water	13
mn_example_population	14
mn_inset_ub	15
mn_join	16
mn_join_audit	18
mn_label_points	19
mn_landcover	20
mn_leaflet	21
mn_map	22
mn_match	24
mn_neighbours	25
mn_population	26
mn_protected_areas	27
mn_provenance	28
mn_roads	29
mn_settlements	30
mn_soums	31
mn_sources	33
mn_translit	34
mn_ub	35
mn_ub_grid	36
mn_zonal	37
theme_mn	38

Index	39
--------------	-----------

mn_admin

*Administrative boundaries of Mongolia at any level***Description**

The engine behind [mn_country\(\)](#), [mn_regions\(\)](#), [mn_aimags\(\)](#), [mn_soums\(\)](#), [mn_khoroos\(\)](#) and [mn_bags\(\)](#). Every function returns the same columns, so maps from different levels can be combined and joined in the same way.

Usage

```
mn_admin(
  level = "aimag",
  within = NULL,
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

Arguments

level	The level to return: • "country": Mongolia. • "region": the five economic regions used by NSO (Western, Khangai, Central, Eastern, Ulaanbaatar). • "aimag": the 21 aimags (provinces) and the capital, Ulaanbaatar. • "suum": the 330 soums and the 9 districts (duureg) of Ulaanbaatar. • "bag": the 204 khoroos of Ulaanbaatar, plus rural bags if you have registered a bag boundary file (see mn_bags()). Synonyms such as "province", "district", "adm1" or "khoroo" also work.
within	Keep only units inside these places, or these places themselves: names or codes of any level, such as "Khovd", "MN84" or "Western region".
resolution	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see mn_cache_dir() .
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
boundary_version	An available data revision, currently only "data-v1". NULL uses that revision. See mn_vintages() ; statistical years do not imply matching historical boundaries.

Value

An sf tibble with one row per unit and the columns:

- pcode** Unique code (see [mn_codes\(\)](#)).
- name** Name in the language chosen with lang.
- name_en, name_mn, name_mns** English, Cyrillic and MNS Latin names.
- level, type** Level ("aimag", "soum", ...) and type ("capital", "district", "khoroo", ...).
- number** Bag or khoroo number within its soum or district.
- iso_code** ISO 3166-2 code (aimags only).
- nso_code** Code used in NSO statistical tables.
- parent_pcode, region_pcode, aimag_pcode, soum_pcode** Codes of the units that contain this one.
- area_km2** Area in square kilometres, computed from the full-resolution boundary.
- geometry** Multipolygon boundary.

See Also

Other admin boundaries: [mn_aimags\(\)](#), [mn_bags\(\)](#), [mn_country\(\)](#), [mn_soums\(\)](#), [mn_ub\(\)](#)

Examples

```
mn_admin("aimag")
mn_admin("soum", within = "Khovd")
mn_admin("aimag", within = c("Khovd", "Uvs"), lang = "mn")
```

<code>mn_aimags</code>	<i>Aimags (provinces) of Mongolia</i>
------------------------	---------------------------------------

Description

Returns the 21 aimags and the capital, Ulaanbaatar, which has aimag status (type = "capital").

Usage

```
mn_aimags(
  region = NULL,
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

Arguments

region	Keep only aimags in these economic regions, such as "Western" or "Khangai region". You can also name aimags here to select them directly.
resolution	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see mn_cache_dir() .
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
boundary_version	An available data revision, currently only "data-v1". NULL uses that revision. See mn_vintages() ; statistical years do not imply matching historical boundaries.

Value

An sf tibble; see [mn_admin\(\)](#) for the columns.

See Also

Other admin boundaries: [mn_admin\(\)](#), [mn_bags\(\)](#), [mn_country\(\)](#), [mn_soums\(\)](#), [mn_ub\(\)](#)

Examples

```
aimags <- mn_aimags()
aimags
mn_aimags(region = "Western")
mn_aimags(lang = "mn")
```

mn_aimag_grid

A grid layout of the aimags for small-multiple maps

Description

Positions of the 21 aimags and Ulaanbaatar on a 4-by-9 grid that keeps their rough geographic arrangement, in the format used by the **geofacet** package.

Usage

```
mn_aimag_grid
```

Format

A data frame with 22 rows and 4 columns: row, col, code (the aimag pcode) and name (English name).

See Also

Other mapping helpers: [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_label_points\(\)](#), [mn_leaflet\(\)](#), [mn_map\(\)](#), [mn_ub_grid](#), [theme_mn\(\)](#)

Examples

```
library(ggplot2)
pop <- mn_example_population[nchar(mn_example_population$Region) == 3, ]
pop$code <- mn_match(pop$Region)
ggplot(pop, aes(Year, value / 1000)) +
  geom_line() +
  geofacet::facet_geo(~code, grid = mn_aimag_grid, label = "name") +
  labs(y = "Population (thousands)")
```

mn_bags

Bags: the smallest rural units

Description

Bags (baga) are the subdivisions of soums; there are about 1,650 of them. Their codes and names are built in (see [mn_codes\("bag"\)](#)), but **their boundaries are not openly published**: the National Statistics Office of Mongolia (NSO) shares them on request. Once you have a bag boundary file, [mn_bags\(\)](#) and every other function in the package can use it.

Usage

```
mn_bags(
  aimag = NULL,
  soum = NULL,
  path = getOption("mongolmaps.bags_path"),
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL
)

mn_read_bags(
  path,
  code_col = NULL,
  check_nesting = TRUE,
  nesting_tolerance = 0.001
)
```

Arguments

aimag, soum	Keep only bags in these aimags or soums (names or codes).
path	Path to a bag boundary file readable by sf::st_read() . Defaults to <code>getOption("mongolmaps.bags_path")</code> .

resolution	Validated for consistency with other getters; the supplied bag geometry is retained at both settings.
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see <code>mn_crs()</code>), or any value accepted by <code>sf::st_crs()</code> .
code_col	Name of the column holding bag codes: 7-digit NSO codes ("1830151") or P-codes ("MN830151"). NULL detects it.
check_nesting	If TRUE (default), stop when a bag lies outside its soum. Up to <code>nesting_tolerance</code> of its area may fall outside the simplified parent boundary.
nesting_tolerance	Maximum fraction of bag area outside its parent, from 0 to 1. Defaults to 0.001 (0.1%). Duplicate codes are dissolved.

Value

An sf tibble; see `mn_admin()` for the columns.

Getting bag boundaries

Write to NSO (international@nso.mn) and ask for the bag boundaries as a shapefile or GeoPackage, with each bag's NSO code. Then either pass the file to `mn_bags(path = ...)`, or register it once per session with `options(mongolmaps.bags_path = "path/to/bags.gpkg")` (put that line in your .Rprofile to make it permanent). Registered bags are also returned by `mn_admin("bag")` and used by `mn_join()`.

`mn_read_bags()` checks the file: every bag must have a known code and sit inside its soum.

See Also

Other admin boundaries: `mn_admin()`, `mn_aimags()`, `mn_country()`, `mn_soums()`, `mn_ub()`

Examples

```
# Codes and names of all bags, without boundaries:
mn_codes("bag", within = "Khovd")

# Without a bag boundary file, mn_bags() explains how to get one:
try(mn_bags())

# With the file from NSO (not run, as the file is not public):
## Not run:
mn_bags(aimag = "Khovd", path = "bags_from_nso.gpkg")

## End(Not run)
```

mn_cache_dir	<i>Manage downloaded map data</i>
--------------	-----------------------------------

Description

Full-resolution boundaries and the larger thematic layers are downloaded the first time you use them and kept in a cache folder, so later calls work offline.

Usage

```
mn_cache_dir()

mn_cache_list()

mn_cache_clear(layers = NULL, old_versions = FALSE)

mn_download(layers = "all")

mn_assets()

mn_cache_verify(layers = NULL)
```

Arguments

layers	Layer ids or groups (see mn_sources()), such as "admin_high" or "osm". NULL (the default for <code>mn_cache_clear()</code>) means all; "all" (the default for <code>mn_download()</code>) means every fixed-file downloadable layer. WorldPop is opt-in: use "worldpop" for every year/resolution, or an id such as "worldpop_2025_1km".
old_versions	If TRUE, <code>mn_cache_clear()</code> removes only data left by older versions of the package.

Details

- `mn_cache_dir()` returns the cache folder. Change it with `options(mongolmaps.cache_dir = "path")`.
- `mn_cache_list()` lists what is cached.
- `mn_cache_clear()` deletes cached files. Old revisions are retained unless explicitly removed with `old_versions = TRUE`.
- `mn_cache_verify()` hashes downloaded assets to check their integrity.
- `mn_assets()` lists concrete assets, including WorldPop years/resolutions.
- `mn_download()` downloads layers now, for example before fieldwork without internet. See [mn_sources\(\)](#) for the layer ids.

Value

`mn_cache_dir()`: a path. `mn_cache_list()`: a tibble with one row per cached file. `mn_cache_clear()` and `mn_download()`: the affected paths, invisibly.

See Also

Other data sources and cache: [mn_provenance\(\)](#), [mn_sources\(\)](#)

Examples

```
mn_cache_dir()
mn_cache_list()

mn_download("admin_high")
```

mn_codes	<i>Codes and names of Mongolian administrative units</i>
----------	--

Description

Returns the table behind every map in the package: one row per unit, from the country down to bags and khoroos, with its codes, names and place in the hierarchy. Use it to look up codes, to build your own crosswalks, or to see which units have a boundary.

Usage

```
mn_codes(level = NULL, within = NULL, aliases = FALSE, lang = NULL)
```

Arguments

level	Levels to return (see mn_match()); NULL returns all.
within	Keep only units inside this parent (a name or code).
aliases	If TRUE, return one row per known spelling of each unit (the table used by mn_match() instead of one row per unit.
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .

Details

Units are identified by pcode, a code that is unique across levels:

- "MN" for the country, "MNR1" to "MNR5" for the economic regions (Western, Khangai, Central, Eastern, Ulaanbaatar);
- "MN" + the two-digit NSO aimag code for aimags ("MN84" Khovd, "MN11" Ulaanbaatar);
- two more digits for soums and Ulaanbaatar districts ("MN8401"), and two more for bags and khoroos ("MN110751").

These match the P-codes of the humanitarian Common Operational Dataset and are the NSO statistical codes without their leading region digit (nso_code, as used in NSO PXWeb tables).

The type column separates units that share a level: "capital" (Ulaanbaatar) among aimags; "district" and "village" among soums; "khoroo" among bags. Villages (tosgon) and rural bags have codes and statistics but no boundary (has_geometry is FALSE).

Value

A tibble. With `aliases = FALSE`: the columns of every admin map (see `mn_admin()`) plus `has_geometry`.
 With `aliases = TRUE`: `pcode`, `level`, `name_en`, `alias` and `source`.

See Also

Other names and codes: `mn_match()`, `mn_translit()`

Examples

```
mn_codes("aimag")
mn_codes("soum", within = "Khovd")
mn_codes("aimag", aliases = TRUE)
```

mn_country	<i>Mongolia and its economic regions</i>
------------	--

Description

`mn_country()` returns the outline of Mongolia. `mn_regions()` returns the five economic regions used in NSO statistics: Western, Khangai, Central, Eastern and Ulaanbaatar.

Usage

```
mn_country(
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)

mn_regions(
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

Arguments

resolution	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see <code>mn_cache_dir()</code> .
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .

crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
boundary_version	An available data revision, currently only "data-v1". NULL uses that revision. See mn_vintages() ; statistical years do not imply matching historical boundaries.

Value

An sf tibble; see [mn_admin\(\)](#) for the columns.

See Also

Other admin boundaries: [mn_admin\(\)](#), [mn_aimags\(\)](#), [mn_bags\(\)](#), [mn_soums\(\)](#), [mn_ub\(\)](#)

Examples

```
mn_country()
mn_regions()
```

mn_crs	<i>Coordinate reference systems suited to Mongolia</i>
--------	--

Description

Returns an equal-area or conformal projection suited to Mongolia. Maps of the whole country can use Albers equal-area or Lambert conformal conic projection centred on Mongolia; maps of Ulaanbaatar look best in UTM zone 48N.

Usage

```
mn_crs(type = c("albers", "lcc", "utm", "wgs84"))
```

Arguments

type	One of: "albers": Albers equal-area conic centred on 104E, 47N, with standard parallels 43.5N and 50.5N. Best for national choropleths because areas are true. "lcc": Lambert conformal conic with the same parameters. Keeps local shapes true. "utm": UTM zone 48N (EPSG:32648), the usual choice for Ulaanbaatar and central Mongolia. "wgs84": plain longitude and latitude (EPSG:4326).
------	---

Value

An `sf::st_crs()` object.

See Also

Other mapping helpers: `mn_aimag_grid`, `mn_erase_water()`, `mn_inset_ub()`, `mn_label_points()`, `mn_leaflet()`, `mn_map()`, `mn_ub_grid`, `theme_mn()`

Examples

```
mn_crs()
sf::st_transform(mn_aimags(), mn_crs("albers"))
```

<code>mn_elevation</code>	<i>Elevation and hillshade</i>
---------------------------	--------------------------------

Description

Terrain rasters for Mongolia from the Copernicus DEM GLO-90.

Usage

```
mn_elevation(
  resolution = c("1km", "90m"),
  within = NULL,
  mask = TRUE,
  crs = NULL
)

mn_hillshade(
  resolution = c("1km", "90m"),
  within = NULL,
  mask = TRUE,
  angle = 40,
  direction = 315,
  crs = NULL
)
```

Arguments

- `resolution` "1km" or "90m".
- `within` Area to return (names or codes). Required for the online resolutions: "90m" elevation and "10m" land cover.
- `mask` If TRUE (default), cells outside Mongolia (or outside `within`) are set to NA.
- `crs` Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see `mn_crs()`), or any value accepted by `sf::st_crs()`.

angle, direction

Sun elevation and direction (degrees) for the hillshade.

Details

- resolution = "1km" (default): a national grid in the Albers projection, downloaded once (about 5 MB) and cached.
- resolution = "90m": the original 90 m data, read directly from the cloud for the area in within (an aimag or smaller). Nothing is cached.

mn_hillshade() computes shaded relief from the elevation, for a background under other layers.

Value

A terra SpatRaster: elevation in metres, or hillshade values between 0 and 1.

Source

Copernicus DEM GLO-90, (c) DLR e.V. 2010-2014 and (c) Airbus Defence and Space GmbH 2014-2018, provided under COPERNICUS by the European Union and ESA. Free to use with attribution.

See Also

Other raster layers: [mn_landcover\(\)](#), [mn_population\(\)](#), [mn_zonal\(\)](#)

Examples

```
elev <- mn_elevation()
terra::plot(elev)
ub <- mn_elevation("90m", within = "Ulaanbaatar")
terra::plot(mn_hillshade(within = "Khovd"), col = grey.colors(100), legend = FALSE)
```

mn_erase_water

Remove water from display geometry

Description

Return a new polygon layer with water removed. The input object and its area_km2 field are unchanged; land_area_km2 describes the new geometry. Land area is not inhabited area. Water source/resolution affects this cartographic operation and any land-only density interpretation.

Usage

```
mn_erase_water(x, water = NULL, min_area_km2 = 0)
```

Arguments

<code>x</code>	An sf polygon object.
<code>water</code>	An sf polygon water layer, defaulting to <code>mn_lakes()</code> .
<code>min_area_km2</code>	Ignore water polygons smaller than this area.

Value

An sf object with `land_area_km2`, preserving one row per input unit, including empty geometry for entirely water-covered units.

See Also

Other mapping helpers: `mn_aimag_grid`, `mn_crs()`, `mn_inset_ub()`, `mn_label_points()`, `mn_leaflet()`, `mn_map()`, `mn_ub_grid`, `theme_mn()`

Examples

```
land <- mn_erase_water(mn_aimags())
head(land$land_area_km2)
```

`mn_example_population` *Mid-year resident population of Mongolia (example data)*

Description

Population by administrative unit for 2015, 2020 and 2025, exactly as returned by the National Statistics Office (NSO) table DT_NSO_0300_002V4 via the **mongolstats** package. The Region column mixes levels (national total, regions, aimags, soums, bags and khoros), as NSO tables do, which makes it a good example for `mn_join()`.

Usage

```
mn_example_population
```

Format

A data frame with 6,672 rows and 4 columns:

Region NSO unit code, such as "0" (Mongolia), "183" (Bayan-Ulgii), "18301" (Ulgii soum) or "5110751" (Bayangol district, 1st khoroo).

Region_en English label of the unit.

Year Year.

value Mid-year resident population.

Source

National Statistics Office of Mongolia, <https://data.1212.mn/pxweb/>, retrieved 2026-09-29.

See Also

Other joining data: [mn_join\(\)](#), [mn_join_audit\(\)](#)

Examples

```
head(mn_example_population)
```

mn_inset_ub

National and urban maps with Ulaanbaatar inset panels

Description

Compose ordinary ggplots without shifting any analytical coordinates. All panels share fill limits and breaks. When supplying different data to the national and urban panels, their fill expressions must have compatible units. Insets have independent map scales.

Usage

```
mn_inset_ub(
  x = NULL,
  ub = NULL,
  fill = NULL,
  label = FALSE,
  title = NULL,
  limits = NULL,
  breaks = NULL,
  ...
)

mn_map_ub(
  x = NULL,
  fill = NULL,
  label = FALSE,
  title = NULL,
  limits = NULL,
  breaks = NULL,
  ...
)
```

Arguments

x	A national map for <code>mn_inset_ub()</code> , or a UB district/khoroo map for <code>mn_map_ub()</code> . Defaults to aimags or UB districts respectively.
ub	Optional urban data for the inset. By default, select Ulaanbaatar units from x; if none exist, stop rather than invent data.
fill	An unquoted fill column/expression, as in mn_map() .

label	Label units in each panel.
title	Overall title.
limits, breaks	Shared fill scale limits and breaks.
...	Additional arguments to <code>mn_map()</code> , such as lang, trans, bins or midpoint.

Value

A patchwork object; requires the optional patchwork package. No input geometry is modified.

See Also

Other mapping helpers: `mn_aimag_grid`, `mn_crs()`, `mn_erase_water()`, `mn_label_points()`, `mn_leaflet()`, `mn_map()`, `mn_ub_grid`, `theme_mn()`

Examples

```
mn_inset_ub(fill = area_km2)
mn_map_ub(fill = area_km2)
```

mn_join

Join your data to a map of Mongolia

Description

Attaches a data frame to boundaries in one step. The column in by can hold names in any common spelling, Cyrillic names, NSO codes, ISO codes or P-codes; they are matched with `mn_match()`. Every unit of the level is kept, so places without data still appear on the map (with NA).

Usage

```
mn_join(
  data,
  by,
  level = NULL,
  by_parent = NULL,
  within = NULL,
  drop_other_levels = TRUE,
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  fuzzy = TRUE,
  max_dist = NULL,
  strict = FALSE,
  strata = NULL,
  value_cols = NULL,
  ub_policy = c("prefer_city", "prefer_region", "error", "none"),
  duplicates = c("allow", "warn", "error")
)
```

Arguments

data	A data frame.
by	The column of data with place names or codes, as a string or a bare column name.
level	The level of the places in by. NULL detects it.
by_parent	Optional column of data naming each row's parent (for example the aimag of each soum). Needed when soum or bag names repeat across the country.
within	Keep only units inside these places, or these places themselves: names or codes of any level, such as "Khovd", "MN84" or "Western region".
drop_other_levels	If TRUE (default), rows for units at other levels, such as national or aimag totals in a soum table, are dropped with a single message. If FALSE they are reported as unmatched.
resolution	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see mn_cache_dir() .
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
fuzzy, max_dist	Passed to mn_match() .
strict	If TRUE, reject missing, unmatched, ambiguous or fuzzy keys, missing boundaries, duplicate statistical keys and UB conflicts. Deliberately excluded levels and geographic areas are allowed.
strata	Character vector of statistical key columns, such as <code>c("Year", "Sex")</code> . NULL recognizes common dimension names. Supply this explicitly for other dimensions or repeated records.
value_cols	Columns to reconcile for Ulaanbaatar. NULL uses numeric columns except keys and recognized dimensions.
ub_policy	"prefer_city" (default), "prefer_region", "error" on conflicting values, or "none" to disable reconciliation.
duplicates	"allow" (default), "warn" or "error" for repeated pcode/stratum keys. Strict mode always rejects duplicates.

Details

Level. When level is NULL, the level with the most matches is used. NSO tables often list several levels in one column (the national total, regions, aimags, soums ...); rows at other levels are dropped with a message (see `drop_other_levels`). Join NSO tables by their code column (such as Region) rather than the label column: labels such as "Ulaanbaatar" name both a region and an aimag.

Ulaanbaatar. The Ulaanbaatar region (NSO code 5) and the capital (511) cover the same area. Many NSO tables, health tables in particular, give the capital's figures only on the region row and

leave 511 empty or out. At the aimag level, missing capital values are filled from the region within each statistical stratum. Conflicts are reported; ub_policy controls which source wins. A few tables use 511 for something else (for example "Other"); check their metadata first. Use ub_policy = "none" for tables where this equivalence is invalid.

Several rows per unit. Data with several rows per place (for example one per year) give several copies of that place's polygon, ready for ggplot2::facet_wrap().

Places without boundaries. Villages (tosgon) and rural bags have NSO codes but no boundary; their rows are reported and left out.

Value

An sf tibble: the boundaries of every unit at the level, with the columns of data added. Columns of data whose names clash with boundary columns get the suffix .data. `mn_join_audit()` returns one diagnostic record per original input row.

See Also

Other joining data: `mn_example_population`, `mn_join_audit()`

Examples

```
# Mid-year population from NSO, by aimag and year
pop <- mn_example_population[mn_example_population$Year == 2025, ]
mn_join(pop, by = "Region", level = "aimag")

# Khoros of Ulaanbaatar, from the same table
mn_join(pop, by = "Region", level = "bag", within = "Ulaanbaatar")

# Names in any spelling work
df <- data.frame(aimag = c("Khovsgol", "Hovd", "\\u0423\\u0432\\u0441"), value = 1:3)
mn_join(df, aimag)
```

mn_join_audit

Inspect the input records behind a map join

Description

Inspect the input records behind a map join

Usage

```
mn_join_audit(x)
```

Arguments

x An object returned by `mn_join()`.

Value

A tibble with one row per original input record. `action` distinguishes joined, coalesced, superseded, outside-scope, other-level, missing, unmatched, ambiguous and boundaryless rows. Original row numbers refer to the original input, even after filtering the map. The `audit` is an attribute; rebuilding an object can remove it.

See Also

Other joining data: [mn_example_population](#), [mn_join\(\)](#)

Examples

```
x <- mn_join(data.frame(place = c("Khovd", "Uvs"), value = 1:2), "place")
mn_join_audit(x)
```

mn_label_points	<i>Label points for map units</i>
-----------------	-----------------------------------

Description

Returns one point per unit, placed well inside its polygon (unlike a centroid, which can fall outside curved shapes). Points for the maps in this package are precomputed; other polygons use [sf::st_point_on_surface\(\)](#).

Usage

```
mn_label_points(x)
```

Arguments

`x` An sf object of polygons.

Value

An sf object of points with the attributes of `x`.

See Also

Other mapping helpers: [mn_aimag_grid](#), [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_leaflet\(\)](#), [mn_map\(\)](#), [mn_ub_grid](#), [theme_mn\(\)](#)

Examples

```
mn_label_points(mn_aimags())
```

mn_landcover

*Land cover***Description**

Land cover from ESA WorldCover 2021, in 11 classes such as grassland, bare ground, cropland and built-up areas. The result is a categorical raster with class names and the official colours, so `terra::plot()` draws it with a legend.

Usage

```
mn_landcover(
  resolution = c("1km", "10m"),
  within = NULL,
  mask = TRUE,
  crs = NULL
)
```

Arguments

<code>resolution</code>	"1km" or "10m".
<code>within</code>	Area to return (names or codes). Required for the online resolutions: "90m" elevation and "10m" land cover.
<code>mask</code>	If TRUE (default), cells outside Mongolia (or outside <code>within</code>) are set to NA.
<code>crs</code>	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see <code>mn_crs()</code>), or any value accepted by <code>sf::st_crs()</code> .

Details

- `resolution = "1km"` (default): national grid generalized from source overviews using modal resampling, downloaded once (about 2 MB). It is not guaranteed to be the native 10 m majority of each square kilometre.
- `resolution = "10m"`: the original 10 m data for the area in `within` (a soum, district or smaller aimag), read from the cloud.

Value

A categorical terra `SpatRaster`.

Source

ESA WorldCover 10 m 2021 v200, (c) ESA WorldCover project / contains modified Copernicus Sentinel data (2021) processed by the ESA WorldCover consortium. CC BY 4.0.

See Also

Other raster layers: `mn_elevation()`, `mn_population()`, `mn_zonal()`

Examples

```
lc <- mn_landcover()
terra::plot(lc)
terra::plot(mn_landcover("10m", within = "Nalaikh"))
```

mn_leaflet

*Interactive maps with leaflet***Description**

Shows any map from this package, or your joined data, on an interactive web map with a basemap, hover labels and click pop-ups.

Usage

```
mn_leaflet(
  x = NULL,
  fill = NULL,
  popup = NULL,
  lang = NULL,
  tiles = "CartoDB.Positron",
  opacity = 0.7
)
```

Arguments

<code>x</code>	An sf object; defaults to the aimags.
<code>fill</code>	Column to colour by (unquoted), or a single colour.
<code>popup</code>	Column with pop-up text (unquoted). By default the pop-up shows the English and Mongolian names, the code and the fill value.
<code>lang</code>	Language of the hover labels: "en", "mn" or "mns".
<code>tiles</code>	A basemap from leaflet::providers , such as "CartoDB.Positron" (default) or "OpenStreetMap".
<code>opacity</code>	Fill opacity between 0 and 1.

Value

A leaflet htmlwidget.

See Also

Other mapping helpers: [mn_aimag_grid](#), [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_label_points\(\)](#), [mn_map\(\)](#), [mn_ub_grid](#), [theme_mn\(\)](#)

Examples

```
mn_leaflet()
mn_leaflet(mn_khoroos(), fill = area_km2)
```

mn_map

*Quick maps of Mongolia with ggplot2***Description**

Draws any map from this package, or a map joined to your data, with sensible defaults: a projection suited to Mongolia, a colour-blind friendly palette, grey for missing values, optional labels and surroundings, and the data attribution as a caption. The result is a normal ggplot, so you can add layers, scales and themes to it.

Usage

```
mn_map(
  x = NULL,
  fill = NULL,
  trans = "identity",
  label = FALSE,
  context = FALSE,
  crs = NULL,
  lang = NULL,
  caption = TRUE,
  title = NULL,
  label_size = 2.6,
  limits = NULL,
  breaks = NULL,
  bins = NULL,
  midpoint = NULL,
  na_label = "No data"
)
```

Arguments

x	An sf object, typically from mn_aimags() , mn_soums() , mn_khoroos() or mn_join() , or a terra raster such as mn_elevation() , mn_landcover() or mn_population() . Defaults to the aimags.
fill	Column to colour the polygons by (unquoted), or a single colour such as "steelblue". Numbers get a continuous viridis scale; text and factors get a discrete one.

trans	Transformation of a numeric fill scale, such as "log10" or "sqrt". Useful when Ulaanbaatar dwarfs everything else.
label	TRUE to label each unit with its name (khoroos with their number), or an unquoted column to label with.
context	If TRUE, draws neighbouring countries, major rivers and lakes around the map.
crs	Projection. NULL picks one: Albers equal-area for the country or large parts of it, UTM zone 48N for Ulaanbaatar and other small areas in central Mongolia. See mn_crs() .
lang	Language of labels when label = TRUE: "en", "mn" or "mns".
caption	TRUE adds the data attribution from mn_citation() ; FALSE adds none; a string is used as is.
title	Optional plot title.
label_size	Text size of labels.
limits, breaks	Shared fill scale limits and legend breaks. Useful for comparing years or inset panels.
bins	For polygons, either a number of intervals or increasing numeric cut points. Values are binned before plotting; trans applies only to continuous scales.
midpoint	For continuous polygon values, use a diverging scale around this value (for example zero for differences).
na_label	Caption key for missing polygon values. NULL disables it.

Value

A ggplot object.

See Also

Other mapping helpers: [mn_aimag_grid](#), [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_label_points\(\)](#), [mn_leaflet\(\)](#), [mn_ub_grid](#), [theme_mn\(\)](#)

Examples

```
mn_map()
mn_map(mn_aimags(), fill = area_km2, label = TRUE)
mn_map(mn_khoroos(district = "Bayangol"), label = TRUE)

pop <- mn_example_population[mn_example_population$Year == 2025, ]
pop_map <- mn_join(pop, "Region", level = "aimag")
mn_map(pop_map, fill = value, trans = "log10", title = "Population, 2025")

mn_map(mn_landcover())
mn_map(mn_elevation()) + ggplot2::geom_sf(data = mn_aimags(), fill = NA, colour = "white")
```

mn_match

*Match place names and codes to Mongolian administrative units***Description**

Converts any common way of writing a Mongolian place into its code: English spellings ("Khuvsgul", "Khovsgol", "Hovsgol"), Cyrillic, NSO statistical codes, ISO 3166-2 codes and P-codes. Names are compared through a normalised key, so spelling variants, case, punctuation and words such as "aimag", "province", "soum" or "district" do not matter. Values that still do not match are tried with a small edit distance.

Usage

```
mn_match(
  x,
  level = NULL,
  within = NULL,
  to = c("pcode", "name_en", "name_mn", "name_mns", "iso_code", "nso_code", "level",
        "type"),
  fuzzy = TRUE,
  max_dist = NULL,
  quiet = FALSE,
  details = FALSE
)
```

Arguments

x	A character vector of names or codes (numbers and factors are converted to character).
level	The level(s) to search: "country", "region", "aimag", "soum" (includes Ulaanbaatar districts) or "bag" (includes khoroos). NULL (the default) searches every level, preferring aimags, then soums, regions, the country and bags.
within	Restricts the search to units inside a parent, given as a name or code. Use it to separate soums that share a name. Either a single value or one value per element of x.
to	What to return: "pcode" (default), "name_en", "name_mn", "name_mns", "iso_code", "nso_code", "level" or "type".
fuzzy	If TRUE (default), values without an exact match are matched by edit distance when a single close candidate exists. Each fuzzy match is reported in a message so you can check it.
max_dist	Maximum edit distance for fuzzy matching. NULL scales it with the length of the name (0 for 3 letters or fewer, up to 3 for long names).
quiet	If TRUE, suppresses the message listing fuzzy matches. Warnings about ambiguous or unmatched values are always shown.
details	If TRUE, return a diagnostic tibble with the input, normalized key, parent, matched code, method, edit distance and candidates.

Value

A character vector the same length as `x`. Values that cannot be matched, or that match several units, are NA and are listed in a warning.

See Also

Other names and codes: `mn_codes()`, `mn_translit()`

Examples

```
mn_match(c("Khuvs gul", "Hovsgol", "\\u0425\\u04e9\\u0432\\u0441\\u0433\\u04e9\\u043b", "MN-041"))

# NSO statistical codes work too
mn_match(c("183", "511", "51107"), to = "name_en")

# Many soums share a name: say which aimag you mean
mn_match("Bayan-Uul", level = "suum", within = "Dornod")

# Khorooos are numbered within their district
mn_match("15-r khoroo", level = "bag", within = "Bayangol")
```

mn_neighbours

Map context: neighbouring countries, rivers and lakes

Description

Layers that give a map of Mongolia its surroundings and water.

Usage

```
mn_neighbours(crs = NULL)

mn_rivers(detail = c("major", "all"), within = NULL, crs = NULL)

mn_lakes(detail = c("major", "all"), within = NULL, crs = NULL)
```

Arguments

<code>crs</code>	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see <code>mn_crs()</code>), or any value accepted by <code>sf::st_crs()</code> .
<code>detail</code>	"major" for the main rivers and lakes (bundled), or "all" for everything in OpenStreetMap (downloaded).
<code>within</code>	Keep only features inside these places (names or codes).

Details

- `mn_neighbours()`: the neighbouring parts of Russia, China and Kazakhstan (Natural Earth, public domain), clipped to a frame 300 km around Mongolia.
- `mn_rivers()` and `mn_lakes()`: with `detail = "major"` (default), major rivers and lakes from Natural Earth, bundled with the package. With `detail = "all"`, every river, stream, canal and water body mapped in OpenStreetMap (ODbL), downloaded once (about 20 MB) and cached.

Value

An sf tibble. Natural Earth layers have `name_en` and `scalerank` (lower is more important). OpenStreetMap layers have `osm_id`, `name`, `name_en`, `name_mn` and `class` (for example "river", "stream", "lake" or "reservoir").

See Also

Other thematic layers: `mn_protected_areas()`, `mn_roads()`, `mn_settlements()`

Examples

```
mn_neighbours()
mn_rivers()
mn_lakes(within = "Khuvsigul")

mn_rivers(detail = "all", within = "Ulaanbaatar")
```

mn_population

Population grids

Description

Estimated population per grid cell from WorldPop (2015 to 2030, constrained to built-up areas), at 1 km or 100 m. The national file for the year is downloaded once from WorldPop (0.4 MB at 1 km, 11 MB at 100 m) and cached. Use `mn_zonal()` to sum it over aimags, soums or khoros.

Usage

```
mn_population(
  year = 2025,
  resolution = c("1km", "100m"),
  within = NULL,
  mask = TRUE,
  crs = NULL
)
```

Arguments

year	A year from 2015 to 2030 (years after the last census are projections).
resolution	"1km" or "100m".
within	Area to return (names or codes). Required for the online resolutions: "90m" elevation and "10m" land cover.
mask	If TRUE (default), cells outside Mongolia (or outside within) are set to NA.
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .

Value

A terra SpatRaster of people per cell. Reprojection uses weighted-sum remapping normalized to the retained native-grid total. This preserves that total but assumes a new local allocation. The `mongolmaps_remapping` attribute records the adjustment. Prefer extraction in the native grid for analysis; clipping and changing grids introduce allocation assumptions. Validate totals for the target extent.

Source

WorldPop (www.worldpop.org), Global 2015-2030 constrained population counts, release R2025A. CC BY 4.0.

See Also

Other raster layers: [mn_elevation\(\)](#), [mn_landcover\(\)](#), [mn_zonal\(\)](#)

Examples

```
pop <- mn_population(2025)
terra::plot(log10(pop))
mn_zonal(pop, mn_aimags())
```

mn_protected_areas	<i>Protected areas</i>
--------------------	------------------------

Description

National parks, nature reserves and other protected and conserved areas of Mongolia from the World Database on Protected Areas (WDPA). The data are downloaded from UNEP-WCMC the first time and refreshed after 90 days; they may not be redistributed, so they never ship with the package.

Usage

```
mn_protected_areas(within = NULL, refresh = FALSE, crs = NULL)
```

Arguments

within	Keep only areas inside these places (names or codes).
refresh	If TRUE, download again now.
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see <code>mn_crs()</code>), or any value accepted by <code>sf::st_crs()</code> .

Value

An sf tibble with the WDPa attributes, including name_eng, desig_eng (designation), iucn_cat and status_yr.

Terms of use

UNEP-WCMC and IUCN, Protected Planet: The World Database on Protected Areas (WDPA), Cambridge, UK. See <https://www.protectedplanet.net/en/legal>. The WDPA may be used for non-commercial purposes with attribution; it may not be redistributed.

See Also

Other thematic layers: `mn_neighbours()`, `mn_roads()`, `mn_settlements()`

Examples

```
pa <- mn_protected_areas()
mn_map(pa, fill = desig_eng)
```

mn_provenance

Inspect source provenance and available boundary revisions

Description

Provenance records distinguish the source snapshot from its effective date, which is unknown for several layers. A high resolution or recent download does not make an unofficial source authoritative.

Usage

```
mn_provenance(x)

mn_vintages(layer = NULL)
```

Arguments

x	An object returned by mongolmaps.
layer	Optional source layer ids, as in <code>mn_sources()</code> .

Value

`mn_provenance()`: a tibble of source metadata attached to the object, or an empty tibble for an untagged object. `mn_vintages()`: a catalog of currently available administrative source snapshots. `effective_date` is missing unless independently established. Metadata are attributes, so reconstructing objects may remove them.

See Also

Other data sources and cache: [mn_cache_dir\(\)](#), [mn_sources\(\)](#)

Examples

```
mn_provenance(mn_aimags())
mn_vintages()
```

mn_roads	<i>Roads, railways, airports and places from OpenStreetMap</i>
----------	--

Description

Transport and settlement layers from OpenStreetMap, prepared from the Humanitarian OpenStreetMap Team exports. Each layer is downloaded once (roads about 30 MB, the others smaller) and cached; see [mn_download\(\)](#).

Usage

```
mn_roads(class = "main", within = NULL, crs = NULL)

mn_railways(stations = FALSE, within = NULL, crs = NULL)

mn_airports(within = NULL, crs = NULL)

mn_places(
  type = c("city", "town", "village"),
  within = NULL,
  lang = NULL,
  crs = NULL
)
```

Arguments

<code>class</code>	Road classes to keep: "main" (motorways to tertiary roads, the default), "minor" (residential, service and unclassified streets), "track", "path", or "all".
<code>within</code>	Keep only features inside these places (names or codes of any level with a boundary, such as "Ulaanbaatar" or "Khovd").

crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
stations	If TRUE, mn_railways() returns railway stations as points instead of the lines.
type	Place types for mn_places() : any of "city", "town", "village", "hamlet", "suburb" and "isolated_dwelling".
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .

Details

Use `within` to get a layer for one place: only the features in that place's bounding box are read from disk, then lines are cut at its border.

Value

An `sf` tibble with `osm_id`, `name` (in the chosen language, where OpenStreetMap has it), `name_en`, `name_mn` and layer-specific columns: `highway`, `class` and `surface` for roads; `railway` for railways; `place` and `population` for places.

Licence

OpenStreetMap data are available under the Open Database Licence (ODbL). Credit "(c) OpenStreetMap contributors" when you publish maps; `mn_citation("osm_roads")` gives the text.

See Also

Other thematic layers: [mn_neighbours\(\)](#), [mn_protected_areas\(\)](#), [mn_settlements\(\)](#)

Examples

```
ub_roads <- mn_roads(class = c("main", "minor"), within = "Ulaanbaatar")
mn_railways()
mn_airports()
mn_places(within = "Khovd")
```

mn_settlements

Settlements: the capital, aimag centres and soum centres

Description

Points for Ulaanbaatar, the 21 aimag centres and the soum centres, handy for labelling maps. Locations come from Wikidata; a few soums without coordinates there use a point inside the soum (see `location_source`).

Usage

```
mn_settlements(
  type = c("capital", "aimag_centre", "soum_centre"),
  within = NULL,
  lang = NULL,
  crs = NULL
)
```

Arguments

type	Which settlements to return: any of "capital", "aimag_centre" and "soum_centre". By default, all.
within	Keep only settlements inside these places (names or codes).
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .

Value

An sf tibble of points with columns admin_pcode (the aimag or soum the settlement is the centre of), soum_pcode, aimag_pcode, type, name, name_en, name_mn, name_mns and location_source.

See Also

Other thematic layers: [mn_neighbours\(\)](#), [mn_protected_areas\(\)](#), [mn_roads\(\)](#)

Examples

```
mn_settlements(type = c("capital", "aimag_centre"))
mn_settlements(within = "Khovd")
```

mn_soums

Soums of Mongolia and districts of Ulaanbaatar

Description

Returns the 330 soums (rural districts) together with the 9 districts (duureg) of Ulaanbaatar, which sit at the same level (type is "soum" or "district").

Usage

```
mn_soums(
  aimag = NULL,
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

Arguments

<code>aimag</code>	Keep only soums in these aimags (names or codes).
<code>resolution</code>	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see mn_cache_dir() .
<code>lang</code>	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
<code>crs</code>	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see mn_crs()), or any value accepted by sf::st_crs() .
<code>boundary_version</code>	An available data revision, currently only "data-v1". NULL uses that revision. See mn_vintages() ; statistical years do not imply matching historical boundaries.

Value

An sf tibble; see [mn_admin\(\)](#) for the columns.

See Also

Other admin boundaries: [mn_admin\(\)](#), [mn_aimags\(\)](#), [mn_bags\(\)](#), [mn_country\(\)](#), [mn_ub\(\)](#)

Examples

```
mn_soums()
mn_soums(aimag = "Khovd")
mn_soums(aimag = c("Uvs", "Zavkhan"), lang = "mn")
```

Description

`mn_sources()` lists every data layer in the package: who provides it, under which licence, whether it ships with the package or is downloaded, and how current it is. `mn_citation()` returns the attribution text to put under a map; `mn_map()` adds it as a caption automatically.

Usage

```
mn_sources(layer = NULL)

mn_citation(layers = "admin")
```

Arguments

`layer, layers` Layer ids, such as "admin", "khorroos" or "settlements". NULL lists all layers. `mn_citation()` also accepts a tagged map or raster, using its attached source provenance.

Details

Please credit the data providers when you publish maps. Most layers are under licences that require attribution (CC BY, ODbL).

Value

`mn_sources()`: a tibble with columns `id`, `title`, `group`, `delivery` ("bundled", "release" or "upstream"), `provider`, `license`, `license_url`, `valid_on`, `source_url`, `attribution` and `notes`, plus source revision, code/geometry vintages, authority and effective date where known. `mn_citation()`: a single string.

See Also

Other data sources and cache: `mn_cache_dir()`, `mn_provenance()`

Examples

```
mn_sources()
mn_citation(c("admin", "khorroos"))
```

mn_translit

*Transliterate Mongolian Cyrillic to Latin script***Description**

Converts Mongolian Cyrillic text to Latin script with a fixed, package-owned table, so results never change between systems. Latin characters, digits and punctuation pass through unchanged.

Usage

```
mn_translit(x, to = c("mns", "nso"))
```

Arguments

- | | |
|----|---|
| x | A character vector (factors are converted to character). |
| to | The romanisation scheme: <ul style="list-style-type: none"> • "mns" (default): the Mongolian national standard MNS 5217:2012, which writes the front vowels as o-umlaut and u-umlaut, for example "Khovsgol" with umlauts. • "nso": the plain-ASCII spelling used in English tables of the National Statistics Office, for example "Khuvs gul". |

Value

A character vector the same length as x.

See Also

Other names and codes: [mn_codes\(\)](#), [mn_match\(\)](#)

Examples

```
mn_translit("\u0425\u04e9\u0432\u0441\u0433\u04e9\u043b")
mn_translit("\u0425\u04e9\u0432\u0441\u0433\u04e9\u043b", to = "nso")
mn_translit("\u0423\u043b\u0430\u0430\u043d\u043d\u0431\u0430\u0442\u0430\u0440")
```

mn_ub

*Ulaanbaatar: city, districts and khoros***Description**

Maps of the capital at three levels:

- `mn_ub()`: the city boundary;
- `mn_ub_districts()`: its 9 districts (duureg);
- `mn_khoros()`: its 204 khoros (subdistricts).

Usage

```
mn_ub(
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

```
mn_ub_districts(
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

```
mn_khoros(
  district = NULL,
  resolution = c("low", "high"),
  lang = NULL,
  crs = NULL,
  boundary_version = NULL
)
```

Arguments

resolution	"low" (default) uses simplified boundaries that ship with the package and suit most maps. "high" uses full-resolution boundaries, downloaded once (about 10 MB) and cached; see <code>mn_cache_dir()</code> .
lang	Language of the name column: "en" (English, as in NSO tables), "mn" (Cyrillic) or "mns" (Latin with diacritics, MNS 5217). Defaults to <code>getOption("mongolmaps.lang", "en")</code> .
crs	Coordinate reference system of the result. NULL (default) keeps longitude/latitude (EPSG:4326). Use "albers", "lcc" or "utm" for a projection suited to Mongolia (see <code>mn_crs()</code>), or any value accepted by <code>sf::st_crs()</code> .

boundary_version	An available data revision, currently only "data-v1". NULL uses that revision. See <code>mn_vintages()</code> ; statistical years do not imply matching historical boundaries.
district	Keep only khoroos in these districts (names or codes, such as "Bayangol", "BGD" or "MN1107").

Details

The three layers share borders exactly: khoroos tile their districts and districts tile the city.

Value

An sf tibble; see `mn_admin()` for the columns. For khoroos, number holds the khoroo number within its district.

Data source

Khoroo boundaries come from the open khoroo-map project (<https://github.com/Tuvshin-Level/khoroo-map>, 0BSD licence), which does not state where its data come from, so treat them as **un-official**. They were fitted to the official Ulaanbaatar outline, with small gaps and overlaps between khoroos resolved: 0.6% of the city area was reassigned, and 97% of khoroos changed area by less than 1%. Khoroo codes and names follow NSO.

The district lines are the unions of their khoroos. They follow the source layout. `mn_soums()` also uses these derived district lines; independent national/urban simplification can create small differences. Neither getter is an official historical district snapshot.

See Also

Other admin boundaries: `mn_admin()`, `mn_aimags()`, `mn_bags()`, `mn_country()`, `mn_soums()`

Examples

```
mn_ub()
mn_ub_districts()
mn_khoroos(district = "Bayangol")
mn_khoroos(lang = "mn", crs = "utm")
```

mn_ub_grid

A geographic small-multiple grid for Ulaanbaatar districts

Description

Approximate positions of the nine districts, suitable for geofacet. This layout is a cartographic arrangement, not a coordinate system.

Usage

```
mn_ub_grid
```

Format

A data frame with nine rows and columns row, col, code (district pcode) and name (English name).

See Also

Other mapping helpers: [mn_aimag_grid](#), [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_label_points\(\)](#), [mn_leaflet\(\)](#), [mn_map\(\)](#), [theme_mn\(\)](#)

Examples

```
data <- mn_ub_grid
data$value <- seq_len(nrow(data))
ggplot2::ggplot(data, ggplot2::aes(1, value)) +
  ggplot2::geom_col() +
  geofacet::facet_geo(~code, grid = mn_ub_grid, label = "name")
```

mn_zonal	<i>Summarise a raster over map units</i>
----------	--

Description

Adds a column to x with a summary of the raster cells in each unit, for example the total population of every soum or the mean elevation of every aimag.

Usage

```
mn_zonal(
  r,
  x = NULL,
  fun = c("sum", "mean", "min", "max", "median"),
  name = "value",
  exact = TRUE
)
```

Arguments

r	A terra SpatRaster (one layer), such as from mn_population() .
x	An sf object of polygons, such as mn_soums() . Defaults to the aimags.
fun	Summary function name: "sum" (default), "mean", "min", "max" or "median".
name	Name of the new column.
exact	If TRUE (default), weight sums and means by the fraction of each cell covered by the polygon. This assumes uniform values within cells. Means weight cell fractions, not physical cell areas. Median requires exact = FALSE. Missing coverage returns NA, not zero.

Value

x with a new column.

See Also

Other raster layers: [mn_elevation\(\)](#), [mn_landcover\(\)](#), [mn_population\(\)](#)

Examples

```
soum_pop <- mn_zonal(mn_population(2025), mn_soums(), name = "population")
mn_map(soum_pop, fill = population / area_km2, trans = "log10")
```

theme_mn

A clean map theme

Description

A minimal ggplot2 theme for maps: no axes or grid, legend at the right, small grey caption.

Usage

```
theme_mn(base_size = 11, base_family = "")
```

Arguments

base_size	Base font size.
base_family	Base font family.

Value

A ggplot2 theme.

See Also

Other mapping helpers: [mn_aimag_grid](#), [mn_crs\(\)](#), [mn_erase_water\(\)](#), [mn_inset_ub\(\)](#), [mn_label_points\(\)](#), [mn_leaflet\(\)](#), [mn_map\(\)](#), [mn_ub_grid](#)

Examples

```
library(ggplot2)
ggplot(mn_aimags()) + geom_sf() + theme_mn()
```

Index

- * **admin boundaries**
 - `mn_admin`, 3
 - `mn_aimags`, 4
 - `mn_bags`, 6
 - `mn_country`, 10
 - `mn_soums`, 31
 - `mn_ub`, 35
- * **data sources and cache**
 - `mn_cache_dir`, 8
 - `mn_provenance`, 28
 - `mn_sources`, 33
- * **datasets**
 - `mn_aimag_grid`, 5
 - `mn_example_population`, 14
 - `mn_ub_grid`, 36
- * **joining data**
 - `mn_example_population`, 14
 - `mn_join`, 16
 - `mn_join_audit`, 18
- * **mapping helpers**
 - `mn_aimag_grid`, 5
 - `mn_crs`, 11
 - `mn_erase_water`, 13
 - `mn_inset_ub`, 15
 - `mn_label_points`, 19
 - `mn_leaflet`, 21
 - `mn_map`, 22
 - `mn_ub_grid`, 36
 - `theme_mn`, 38
- * **names and codes**
 - `mn_codes`, 9
 - `mn_match`, 24
 - `mn_translit`, 34
- * **raster layers**
 - `mn_elevation`, 12
 - `mn_landcover`, 20
 - `mn_population`, 26
 - `mn_zonal`, 37
- * **thematic layers**
 - `mn_neighbours`, 25
 - `mn_protected_areas`, 27
 - `mn_roads`, 29
 - `mn_settlements`, 30
- `leaflet::providers`, 21
- `mn_admin`, 3
- `mn_admin()`, 5, 7, 10, 11, 32, 36
- `mn_aimag_grid`, 5, 12, 14, 16, 19, 22, 23, 37, 38
- `mn_aimags`, 4
- `mn_aimags()`, 3, 4, 7, 11, 22, 32, 36
- `mn_airports (mn_roads)`, 29
- `mn_assets (mn_cache_dir)`, 8
- `mn_bags`, 6
- `mn_bags()`, 3–5, 11, 32, 36
- `mn_cache_clear (mn_cache_dir)`, 8
- `mn_cache_dir`, 8
- `mn_cache_dir()`, 3, 5, 10, 17, 29, 32, 33, 35
- `mn_cache_list (mn_cache_dir)`, 8
- `mn_cache_verify (mn_cache_dir)`, 8
- `mn_citation (mn_sources)`, 33
- `mn_citation()`, 23
- `mn_codes`, 9
- `mn_codes()`, 4, 25, 34
- `mn_country`, 10
- `mn_country()`, 3–5, 7, 32, 36
- `mn_crs`, 11
- `mn_crs()`, 3, 5–7, 11, 12, 14, 16, 17, 19, 20, 22, 23, 25, 27, 28, 30–32, 35, 37, 38
- `mn_download (mn_cache_dir)`, 8
- `mn_download()`, 29
- `mn_elevation`, 12
- `mn_elevation()`, 21, 22, 27, 38
- `mn_erase_water`, 13
- `mn_erase_water()`, 6, 12, 16, 19, 22, 23, 37, 38
- `mn_example_population`, 14, 18, 19
- `mn_hillshade (mn_elevation)`, 12

mn_inset_ub, 15
 mn_inset_ub(), 6, 12, 14, 19, 22, 23, 37, 38
 mn_join, 16
 mn_join(), 7, 14, 15, 18, 19, 22
 mn_join_audit, 18
 mn_join_audit(), 15, 18
 mn_khoroos (mn_ub), 35
 mn_khoroos(), 3, 22
 mn_label_points, 19
 mn_label_points(), 6, 12, 14, 16, 22, 23, 37, 38
 mn_lakes (mn_neighbours), 25
 mn_lakes(), 14
 mn_landcover, 20
 mn_landcover(), 13, 22, 27, 38
 mn_leaflet, 21
 mn_leaflet(), 6, 12, 14, 16, 19, 23, 37, 38
 mn_map, 22
 mn_map(), 6, 12, 14–16, 19, 22, 33, 37, 38
 mn_map_ub (mn_inset_ub), 15
 mn_match, 24
 mn_match(), 9, 10, 16, 17, 34
 mn_neighbours, 25
 mn_neighbours(), 28, 30, 31
 mn_places (mn_roads), 29
 mn_population, 26
 mn_population(), 13, 21, 22, 37, 38
 mn_protected_areas, 27
 mn_protected_areas(), 26, 30, 31
 mn_provenance, 28
 mn_provenance(), 9, 33
 mn_railways (mn_roads), 29
 mn_read_bags (mn_bags), 6
 mn_regions (mn_country), 10
 mn_regions(), 3
 mn_rivers (mn_neighbours), 25
 mn_roads, 29
 mn_roads(), 26, 28, 31
 mn_settlements, 30
 mn_settlements(), 26, 28, 30
 mn_soums, 31
 mn_soums(), 3–5, 7, 11, 22, 36, 37
 mn_sources, 33
 mn_sources(), 8, 9, 28, 29
 mn_translit, 34
 mn_translit(), 10, 25
 mn_ub, 35
 mn_ub(), 4, 5, 7, 11, 32
 mn_ub_districts (mn_ub), 35
 mn_ub_grid, 6, 12, 14, 16, 19, 22, 23, 36, 38
 mn_vintages (mn_provenance), 28
 mn_vintages(), 3, 5, 11, 32, 36
 mn_zonal, 37
 mn_zonal(), 13, 21, 26, 27
 sf::st_crs(), 3, 5, 7, 11, 12, 17, 20, 25, 27, 28, 30–32, 35
 sf::st_point_on_surface(), 19
 sf::st_read(), 6
 theme_mn, 38
 theme_mn(), 6, 12, 14, 16, 19, 22, 23, 37