Package: nycbuildxwalks (via r-universe)

May 19, 2026

Title Build Geographic Crosswalk Tables for New York City Boundaries

Version 0.0.0.9000

Description Downloads official New York City geographic boundary data from NYC Open Data, the Department of City Planning, and other sources, then generates comprehensive crosswalk tables showing how administrative and spatial boundaries overlap. Produces wide-format and long-format CSV crosswalks with intersection area and percentage calculations. Based on the Python tool by Nathan Storey at MODA-NYC (<https://github.com/MODA-NYC/nyc-geography-crosswalks>), which builds on earlier work by BetaNYC (<https://github.com/BetaNYC/nyc-boundaries>).

License MIT + file LICENSE

URL https://kjhealy.github.io/nycbuildxwalks/, https://github.com/kjhealy/nycbuildxwalks

BugReports https://github.com/kjhealy/nycbuildxwalks/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports cli, dplyr, httr2, jsonlite, purrr, readr, rlang, sf, stringr, withr

Suggests testthat (>= 3.0.0), tibble

Config/testthat/edition 3

LazyData true

LazyDataCompression xz

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository https://kjhealy.r-universe.dev

Date/Publication 2026-04-19 14:03:10 UTC

RemoteUrl https://github.com/kjhealy/nycbuildxwalks

RemoteRef HEAD

RemoteSha 39754bd941edec362779301d208ad7142d6a6ee6

Contents

`build_crosswalks` *Build all crosswalk tables from boundary data*

Description

Reads a combined boundaries GeoJSON file, reprojects to EPSG:2263 (NY State Plane, feet) for accurate area calculations, and builds both long-form and wide-format crosswalk CSVs for each primary geography.

Usage

```
build_crosswalks(
  boundaries_path,
  run_dir,
  buffer_feet = -50,
  min_area_final = 100,
  epsilon = 1e-06,
  exclude_ids = "cc_upcoming",
  primary_only = NULL,
  targets = NULL,
  max_primaries = NULL
)
```

Arguments

boundaries_path	Path to all_boundaries.geojson.
run_dir	Output directory for crosswalk CSVs and metadata.
buffer_feet	Negative buffer in feet for intersection de-noising. Default -50.
min_area_final	Minimum intersection area in square feet. Default 100.
epsilon	Tiny area to suppress numerical noise. Default 1e-6.
exclude_ids	Character vector of geography IDs to exclude. Default "cc_upcoming".
primary_only	Optional character vector to restrict which primary geography IDs are processed.
targets	Optional character vector to restrict target geography IDs.
max_primaries	Optional integer to limit features per primary (for testing).

Value

The path to run_dir (invisibly).

---

build_longform_for_primary

Build long-form crosswalk for a single primary geography

Description

For each feature of the primary geography, computes the intersection area with every feature of each target geography. Returns a tibble with one row per significant pairwise intersection, including area and percentage overlap.

Usage

```
build_longform_for_primary(
  all_gdf,
  primary_id,
  other_ids,
  buffer_feet = -50,
  min_area = 100,
  epsilon = 1e-06,
  max_primaries = NULL
)
```

Arguments

all_gdf	An sf object containing all boundaries, projected to EPSG:2263 (NY State Plane, feet).
primary_id	The geography ID to use as the primary (e.g., "cd").
other_ids	Character vector of target geography IDs.

buffer_feet	Negative buffer (in feet) applied during intersection to suppress edge artifacts. Default -50.
min_area	Minimum intersection area in square feet. Default 100.
epsilon	Tiny area threshold to suppress numerical noise. Default 1e-6.
max_primaries	Optional integer to limit the number of primary features processed (useful for testing).

Value

A tibble with columns primary_geo_id, primary_geo_name, other_geo_id, other_geo_name, primary_area_sqft, intersection_area_sqft, and pct_overlap.

`build_wide_for_primary` *Build wide-format crosswalk for a single primary geography*

Description

For each feature of the primary geography, finds overlapping features from all target geographies and returns a tibble with one row per primary feature and one column per target geography (values are semicolon-separated names of overlapping features).

Usage

```
build_wide_for_primary(
  all_gdf,
  primary_id,
  other_ids,
  buffer_feet = -50,
  min_area = 100,
  epsilon = 1e-06,
  max_primaries = NULL
)
```

Arguments

all_gdf	An sf object containing all boundaries, projected to EPSG:2263 (NY State Plane, feet).
primary_id	The geography ID to use as the primary (e.g., "cd").
other_ids	Character vector of target geography IDs.
buffer_feet	Negative buffer (in feet) applied during intersection to suppress edge artifacts. Default -50.
min_area	Minimum intersection area in square feet. Default 100.
epsilon	Tiny area threshold to suppress numerical noise. Default 1e-6.
max_primaries	Optional integer to limit the number of primary features processed (useful for testing).

Value

A tibble with one row per primary feature. The first column is named after `primary_id` and contains the primary feature names. Remaining columns are named after each target geography ID and contain semicolon-separated overlapping feature names.

`dissolve_by_name` *Dissolve features by name column*

Description

Groups features by `name_col` and unions their geometries, producing one feature per unique name. This prevents duplicate rows for multipart features (e.g., some MODZCTAs).

Usage

```
dissolve_by_name(gdf)
```

Arguments

`gdf`	An sf object with columns `id`, `name_col`, and `geometry`.

Value

An sf object with one row per unique `name_col` value, with columns `id`, `name_col`, and dissolved `geometry`.

`download_all_boundaries`
Download and combine all NYC boundary datasets

Description

Iterates over the datasets defined in nyc_datasets, downloads each one via `process_dataset()`, combines them into a single sf object, fixes invalid geometries, and writes the results to a timestamped output directory.

Usage

```
download_all_boundaries(
  output_dir = "outputs",
  auto_detect_latest = TRUE,
  preferred_cycle = NULL,
  external_data_dir = "data/external",
  datasets = NULL
)
```

Arguments

output_dir	Base directory for timestamped run outputs. Default "outputs".
auto_detect_latest	Logical. Attempt to auto-detect the latest DCP cycle letter. Default TRUE.
preferred_cycle	Optional single letter to pin a DCP cycle.
external_data_dir	Path to local fallback files. Default "data/external".
datasets	A tibble of dataset definitions. Defaults to nyc_datasets.

Value

The path to the timestamped run directory (invisibly).

---

make_run *Run the full crosswalk generation pipeline*

Description

Orchestrates the complete workflow: downloads all NYC boundary datasets via download_all_boundaries(), then builds crosswalk tables via build_crosswalks(). Optionally creates ZIP archives of the outputs.

Usage

```
make_run(
  output_dir = "outputs",
  auto_detect_latest = TRUE,
  preferred_cycle = NULL,
  external_data_dir = "data/external",
  buffer_feet = -50,
  min_area_final = 100,
  epsilon = 1e-06,
  exclude_ids = "cc_upcoming",
  primary_only = NULL,
  targets = NULL,
  max_primaries = NULL,
  zip_artifacts = FALSE
)
```

Arguments

output_dir	Base directory for outputs. Default "outputs".
auto_detect_latest	Logical. Auto-detect latest DCP cycle. Default TRUE.

`preferred_cycle`
: Optional DCP cycle letter to pin.

`external_data_dir`
: Path to local fallback files. Default `"data/external"`.

`buffer_feet`
: Negative buffer for intersection de-noising. Default `-50`.

`min_area_final`
: Minimum intersection area (sq ft). Default `100`.

`epsilon`
: Numerical noise threshold. Default `1e-6`.

`exclude_ids`
: Geography IDs to exclude. Default `"cc_upcoming"`.

`primary_only`
: Optional character vector of primary IDs to build.

`targets`
: Optional character vector of target IDs.

`max_primaries`
: Optional limit on features per primary (for testing).

`zip_artifacts`
: Logical. If `TRUE`, create ZIP archives of the outputs. Default `FALSE`.

Value

The path to the timestamped run directory (invisibly).

`nyc_datasets` *NYC geographic dataset definitions*

Description

A tibble containing metadata for the 15 NYC geographic boundary datasets used to build crosswalk tables. Each row defines a dataset's source URL, the column used for feature names, and the source type.

Usage

```
nyc_datasets
```

Format

`nyc_datasets`**:**

A tibble with 15 rows and 6 columns:

id Short identifier for the geography (e.g., "cd", "pp", "nta")

dataset_name Human-readable name of the geography

url Source URL for downloading the boundary data

name_col Name of the column in the source data that contains feature names

name_alt Optional alternate name column, or `NA` if none

source_type One of "dcp_zip" (DCP shapefile zip with cycle detection), "opendata_shapefile" (NYC Open Data shapefile export), "opendata_geojson" (NYC Open Data GeoJSON), or "edc_zip" (EDC shapefile zip)

Source

NYC Department of City Planning (DCP), NYC Open Data, and the NYC Economic Development Corporation (EDC). Dataset definitions adapted from the Python tool by Nathan Storey at MODA-NYC (https://github.com/MODA-NYC/nyc-geography-crosswalks).

`process_dataset` *Download and standardize a single NYC boundary dataset*

Description

Downloads a geographic boundary dataset, reads it into an sf object, reprojects to EPSG:4326, and standardizes the columns to a common schema (`id`, `name_col`, optionally `name_alt`, and `geometry`).

Usage

```
process_dataset(
  dataset_info,
  auto_detect_latest = TRUE,
  preferred_cycle = NULL,
  external_data_dir = "data/external"
)
```

Arguments

`dataset_info`	A single-row tibble or list with elements `id`, `dataset_name`, `url`, `name_col`, `name_alt`, and `source_type`.
`auto_detect_latest`	Logical. Passed to `resolve_dcp_cycle()` for DCP zip sources. Default `TRUE`.
`preferred_cycle`	Optional cycle letter passed to `resolve_dcp_cycle()`.
`external_data_dir`	Path to a directory containing local fallback files (e.g., `ibz.zip`). Default `"data/external"`.

Value

A list with components:

- `gdf`: An sf object with standardized columns, or `NULL` on failure.
- `meta`: A list with `id`, `original_url`, `resolved_url`, `cycle`, `auto_detected`, `status`, and `error`.

resolve_dcp_cycle *Resolve the latest available DCP cycle for a URL*

Description

DCP boundary files are versioned with a cycle suffix (e.g., `_25a.zip`). This function probes for newer cycles by sending HTTP HEAD requests with ascending letters (b, c, d, ...) and returns the URL for the highest available cycle.

Usage

```
resolve_dcp_cycle(url, auto_detect = TRUE, preferred_cycle = NULL)
```

Arguments

`url`	A DCP boundary URL containing a cycle suffix like `_25a.zip`.
`auto_detect`	Logical. If `TRUE` (default), probe for newer cycles.
`preferred_cycle`	Optional single lowercase letter to pin a specific cycle (e.g., `"d"`). If set and available, overrides auto-detection.

Value

A list with components:

- `resolved_url`: The URL with the best available cycle.
- `meta`: A list with `cycle_source`, `cycle_resolved`, `auto_detected`, and `probes`.

union_by_name *Union features by name column*

Description

Groups features by `name_col` and unions each group's geometry into a single geometry. Returns a tibble of name-geometry pairs rather than an sf object, for use in intersection calculations.

Usage

```
union_by_name(gdf)
```

Arguments

`gdf`	An sf object with a `name_col` column.

Value

A tibble with columns `name` (character) and `geometry` (sfc_GEOMETRY).

Index