Open Zoning Feed Specification

Data standards for describing zoning regulations, parcel geometry, and building characteristics

# Abstract

Efforts to conduct detailed analysis of the variation in and consequences of zoning regulations have been hampered by the lack of standardized, machine-readable data on the requirements described in the texts of municipal zoning codes. Existing efforts to assemble data on zoning regulations have relied on database structures that are not well-suited to the complexity of zoning regulations. In this paper, we introduce and describe the Open Zoning Feed Specification (OZFS) standard as a recommended format for describing zoning regulations, parcel geometry, and building characteristics. We introduce a sample dataset that uses the OZFS standard to describe zoning regulations and parcel geometries for a set of 162 municipalities in the Dallas-Fort Worth region of Texas. The OZFS data standard offers a foundation on which software can be developed to analyze zoning regulations in greater detail than has previously been possible to do at a regional scale.

# Introduction

Scholars and advocates in the United States have shown an increasing interest in assembling data on zoning regulations across municipalities on order to document the degree to which zoning constrains the production of housing. These efforts are hampered by the absence of a flexible and generalizeable data standard that can describe complex zoning regulations in a way that can be consistently parsed by software. The purpose of this paper is to introduce the Open Zoning Feed Specification (OZFS) and offer it as a solution that can describe zoning regulations in a more a complete and consistent manner than the existing state of practice.

## Prior work

An example of the type of research and analysis that is stymied by the lack of standardized, machine-readable zoning data is a study by Conway (2021), which addresses the question of how statewide proposals to relax zoning restrictions in California might impact traffic congestion. That analysis required strong assumptions with regards to allowable setbacks, since these data were not available in a consistent format for municipalities across the state. These assumptions (derived from existing building footprints) were not unreasonable. The method probably overestimates allowable building footprints as often as it underestimated them, so the aggregated results about the overall increases in traffic congestion were probably valid. However, this approach would be less likely to yield useful results about the effects of the proposed policies on specific road segments or within small areas.

The largest existing effort to compile zoning regulations for municipalities throughout the United States into a comprehensive, nationwide database is the National Zoning Atlas (NZA) (National Zoning Atlas 2025f), a project led and founded by Sara Bronin (National Zoning Atlas 2025d). NZA coordinates among local teams to encode zoning regulations. Early in the NZA data collection program, contributors would enter information from the zoning codes into a spreadsheet as tabular (rectangular) data, with a sheet for each municipality, a row for each zoning district, and a column for each of a large number of variables intended to comprehensively represent the regulations that could be represented in a zoning code (Bronin and Ilyankou 2021). NZA subsequently changed its data collection process to replace the use of spreadsheets with an online editor tool that was accessible to those who had registered with NZA (Bronin et al. 2023). The editor tool may have recorded data in a similar format to the spreadsheets and geospatial files that were used previously, although since these data sets are not shared publicly, NZA is free to modify its data structures from time to time according to its internal needs. The current data collection methodology is not publicly available (National Zoning Atlas 2025b). Individual contributors are free to store and share the data they collect in any format, in addition to sharing it with NZA using their current data-entry tools. Tabular data formats seem to have been the most common.

The Mercatus Center at George Mason University has provided funding for local NZA data compilation efforts in nine states and regions (Mercatus Center 2024). Data collected by the teams in those regions are available on the Mercatus website as spreadsheets (csv or Excel) with accompanying geojson files. The Mercatus website includes a data dictionary (Kahn 2024) that notes in its introduction that some numeric variables are commonly encoded as strings, which can be an error in the way they were entered, or it may be necessary in some cases because “it is very common for numeric variables in these datasets to have no single numeric value due to complex and context-dependent regulations within districts” (Kahn 2024, 1).

The Metropolitan Area Planning Council (MAPC), which is the regional planning agency for the Boston metropolitan area, maintains an online zoning atlas for the 101 cities and towns it serves (Metropolitan Area Planning Council 2025). MAPC began collecting data for its zoning atlas in 2012 and published its initial version in 2020 (Reardon, Partridge, and Perkins-High 2020). The MAPC Zoning Atlas effort has since been incorporated into NZA (National Zoning Atlas 2025c). Their compiled zoning data are viewable through an interactive map and downloadable either as ESRI shapefiles or as csv files (both using a tabular data format).

NZA is tightly controlled, both in terms of who has access to the data and who can contribute. The format they currently use to store zoning data is not public. Contributors must be approved by NZA and, although the compiled data is available through an interactive map (National Zoning Atlas 2025a), it is not downloadable for incorporation into third-party software or detailed analysis by researchers (individual contributors to NZA can choose to publish or share the data they collect). It is not open data, in the sense that its terms of use (National Zoning Atlas 2025e) does not meet the Open Knowledge Foundation’s definition of Openness: “Knowledge is open if anyone is free to access, use, modify, and share it – subject, at most, to measures that preserve provenance and openness” (Open Knowledge Foundation 2015). This is not necessarily a weakness of the project. There are many advantages to a more closed approach to data collection and dissemination, including better control of data quality and an assurance that the data will not be used for purposes that are not aligned with an organization’s values and priorities. On the other hand, this approach does create a barrier to data use and to the adoption of data collection methodologies and formats.

The Center for Spatial Planning Analytics and Visualization (CSPAV) at Georgia Institute of Technology has digitized has created an interactive dashboard for exploring zoning regulations in Atlanta (Schwabacher 2022). Like NZA, CSPAV does not allow users to download the data powering its interactive map, and its data format is not apparent from the dashboard itself.

Other efforts at compiling zoning regulations in a machine-readable format include Envelope (Envelope 2022) and Zonenomics (Zoneomics 2024). These are efforts by private firms to compile data that can be sold directly to users, or a that can be used in analyses that the firms do on behalf of customers (and are thus somewhat less open than the NZA effort). Envelope is no longer operating, and detailed information on their data format is not available. Zoneomics offers data through a paid API subscription. API outputs are in a json or geojson format, which offers the potential to overcome a weakness of a tabular data format: variables do not necessarily need to be stored for any districts they are not defined for. As noted by Kahn (2024) for NZA data published by the Mercatus Center, some variable values are necessarily encoded by non-standardized text strings that describe constraints that may depend on land use or on characteristics of the building of the parcel.

A limitation of the data structures used by all of the efforts described above is that they do not offer a consistent method for encoding constraints on the building form that are dependent on features of the building (e.g. when the required setback depends on the building height) or on features of the parcel (e.g. when the required setback depends on the the parcel width). Moreover, none of them differentiate between definitions of terms that may be used differently in different jurisdictions. For example, one jurisdiction might define height as the top of the roof for any structure, but another might define height (for particular roof types) as the midpoint between the top of the roof and the eave. As another example, zoning codes may have different setback requirements for multifamily homes and for townhomes, but a building that is defined as a multifamily building based on the definitions used in one jurisdiction might be defined as a row of townhomes in another. Another limitation is that, with exception of the Zoneomics API, all the efforts described above use a tabular data structure, which would require that any variable defined for any district in the dataset (or in the entire nation, if the goal is to compile a harmonized national dataset) must be defined (at least with a null value) for all districts.

## Contribution

The aim of the OZFS data standard is to offer a scaleable, extendable foundation that can serve as the basis for analytic methods and software that can improve:

* developers’ understanding of where there may be opportunities to build new housing,
* policy-makers’ understanding of where there may be the greatest opportunity to increase opportunities for housing development through targeted changes to zoning regulations, and
* researchers’ understanding of how variations in zoning regulations might relate to variations in outcomes of interest such as housing affordability, transportation accessibility, or economic development.

The OZFS data standard uses a geojson data format to describe zoning regulations, which offers the flexibility to define only those variables that are defined for a particular zoning district and to add additional key-value pairs for new variables as needed. We incorporate the use of expressions (in the Python syntax) to define constraints that are dependent on land use or on characteristics of buildings and parcels. We also use expressions to well as to define municipality-level definitions relating to building characteristics. To ensure that buildings and parcels are described in a way that is interpretable by these expressions, the OZFS data standard also includes standards for encoding building and parcel characteristics.

We aim to maximize the usefulness of this effort by publishing this data standard under a creative commons license. We hope it will prove useful to:

* contributors who want to create OZFS data by manually translating the texts of zoning codes into the OZFS data format,
* data scientists who want to develop applications to automate the creation of OZFS data from zoning codes and geospatial parcel data files, and
* software developers who want to create tools for housing developers and policy-makers seeking to increase the supply of housing.

We were directly inspired by OpenStreetMap (OpenStreetMap contributors 2024), which has defined a flexible data standard for encoding geographic data, allowing volunteer contributors their data collection efforts toward the effort of assembling a global digital map of the world. Noteably, the OpenStreetMap data standard itself has changed over time, as contributors have coordinated to define additional features. It is our vision that OZFS will likewise be revised by contributors over time to better describe the full range of and complexity of zoning code features.

# OZFS Data Structures

The OZFS data standard includes three files:

* a file with a \*.zoning extension to describe the zoning regulations for a particular municipality;
* a file with a \*.parcel extension to describe the geometry of all parcels (or all parcels of interest) within a municipality; and
* a file with a \*.building extension to describe the geometry of a proposed building.

## Zoning regulations

Zoning data are encoded in a separate \*.zoning file for each municipality. The \*.zoning file is formatted as a geojson file, which offers the advantages of a nested data structure. The specific structure of the \*.zoning file is illustrated in [Figure 1](#fig-zoning-structure).

|  |
| --- |
| Figure 1: Structure of the zoning file |

The top level of the file is an array with six key-value pairs:

* Type: As for geojson files, the value for this key should be “FeatureCollection.”
* version: The version of the OZFS data standard used in this file. The current version of the standard is 0.5.0.
* muni\_name: The name of the municipality this zoning code refers to. This is a required value.
* date: The most recent date on which the zoning regulations are known to have been in effect. This is a required value.
* definitions is an array of definitions of terms (in the current version of the standard, height and residential building types) that may very from one municipality to the next and are defined in the text of the zoning code.
* features contains information on each zoning district, where each district represented by an array of three elements: type, properties, and geometry. As for geojson files, the value for the type key will be “Feature” and the geometry key takes an array of coordinates describing the geometry of the feature (in this case, the district boundary). The value for the properties key is an of key-value pairs that may include the following:
  + dist\_name is the name of district. This value is required if there is no value for dist\_name and optional otherwise.
  + dist\_abbr is the abbreviated name of district. This value is required if there is no value for dist\_name and optional otherwise.
  + planned\_dev is a binary value indicating wheter this is a planned development district (where the entire district will be developed by a single developer who negotiates constraints directly with the municipality). This value is optional and is assumed to have a value of false if it is missing.
  + overlay is a binary value indicating whether this district is an overlay district (a district that modifies the requirements of any base districts it overlaps). This value is optional and is assumed to have a value of false if it is missing.
  + res\_types\_allowd is a list of residential land uses that are allowed in the district. All values in the list must also appear in the definitions array. This list of values is required if neither the value of the planned\_dev key nor the value of the overlay key is true.
  + constraints is an array of constraints that define allowable building characteristics. The constraints key must be present if neither the value of the planned\_dev key nor the value of the overlay key is true.

The definitions array (defined for the municipality as a whole) and the constraints array (defined for each district in the features array that is not a planned development district or an overlay district) are described in greater detail below.

### Constraints

Constraints are described by numeric values or expressions and are stored in the structure illustrated in [Figure 2](#fig-constraints). Possible constraints include setback\_front (the front setback), far (the floor area ratio), and lot\_cov\_bldg (the lot coverage). [Appendix A](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-a.html) includes the constraints that have been defined for the \*.zoning file, together with their descriptions. For each constraint that is included in the constraints array, a minimum value min\_val and/or a maximum value max\_val must be given.

|  |
| --- |
| Figure 2: Data structure for constraints in the \*.zoning file |

Minimum and maximum values for constraints are stored as arrays including the following key-value pairs:

* condition: The condition under which the minimum or maximum value applies. This key is required if the number of elements in the min\_value (or max\_value) array is greater than one. The condition can be a logical expression (in Python syntax) defining the condition under which the minimum (or maximum) value applies. [Appendix B](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-b.html) includes a full list of the of the variables that can be used in constraint and condition expressions, along with a description of each variable. If the condition under which the value applies cannot be described as a logical expression within with one or more of those variables, it may be described in a text string (which will limit machine-readability).
* expression: These can either be constant numeric values or equations (in Python syntax) referring to variables listed in [Appendix B](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-b.html).
* min\_max: This key is required if the list of expressions has a more than one element in it. It is a character string that can take one of two values: min or max. A value of min indicates that the governing constraint is the minimum of the possible values listed in the expression key. A value of maximum indicates that the governing constraint is the maximum of the possible values listed in the expression key.

### Definitions

There may be terms that are used in many different zoning codes, but with definitions that vary across municipalities. The current version of the OZFS standard (version 0.5.0) requires definitions for height and for types of residential buildings. Other definitions may be added to future extensions of the standard.

For each definition, one or more arrays comprising conditions and expressions can be defined. The value of the condition key defines the circumstance under which the value of the expression key applies and should be a logical statement (one that returns a value of true or false) in Python syntax, referencing any of the variable names listed in [Appendix B](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-b.html). The value of the expression key should be an equation (in Python syntax) referencing any of the variable names listed in [Appendix B](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-b.html). As an example, if the height of a building is defined as the top of the highest wall plate for buildings with a flat roof type and the mid-point between the top of the roof and the eave for all roof types except a flat roof (see [Appendix C](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-c.html) for an illustration of various roof types), the height definition would be coded as illustrated in [Figure 3](#fig-height-def-ex).

|  |
| --- |
| Figure 3: Illustrative example of a height definition |

Conditions do not necessarily need to be mutually exclusive. When they are not, they are applied in the order in which they appear. For example, if the residential building type (res\_type) of a building with three or more units units is defined as multifamily building unless all units have outside entrances on the ground level, in which case it is defined as a townhouse, this could be encoded as illustrated in [Figure 4](#fig-res-type-def-ex). In that example, all buildings with only one dwelling unit would be defined as single-family. Of the remaining buildings, all buildings with two units would be defined as duplexes. Of the remaining buildings (all of which would have three or more dwelling units), those in which all units have an outside, ground-level entrances would be classified as townhouses, and all other buildings with three or more units would be classified as multifamily buildings.

|  |
| --- |
| Figure 4: Illustrative example of a residential type definition |

## Parcel geometry

Parcel geometry data representing parcel boundaries as polygons are commonly available in GIS files from state, county, or municipal open-data portals. These require pre-processing for zoning analysis because applying required setbacks to determine the buildable area of a parcel requires information not only about the shape and location of the parcel, but also about its orientation with respect to the street, since zoning codes may specify difference setbacks for the front, sides, and rear of a parcel, respectively.

In the OZFS data standard, parcels must be represented in a geojson file that includes, for each parcel, line strings representing each parcel edge (front, back, and side(s) and a point representing the parcel centroid. All features have a parcel\_id key with a value that uniquely identifies which parcel each edge or centroid is associated with.

Each feature in the parcel dataset will also have a key side that can take one of six values:

* front indicates that this is a line string representing the front of a parcel.
* rear indicates that this is line string representing the rear of a parcel.
* interior side indicates that this is a line string representing the interior side of a parcel (the side adjacent to another parcel).
* exterior side indicates that this is a line string representing the exterior side of a parcel. Only a corner lot can have an exterior side. This is the side of a corner lot that is adjacent (and approximately parallel) to the street that is not indicated in the parcel’s address.
* unknown indicates that this is a line string representing the side of a parcel that has not been classified into one of the above categories (for irregular parcel geometries and/or parcels where the relationship to the adjacent street network is unclear).
* centroid indicates that this is a point representing the parcel’s centroid.

Parcel centroids have three additional key/value pairs. lot\_width indicates the width of the parcel in feet. lot\_depth indicates the depth of the parcel in feet. lot\_area indicates the area of the parcel in acres.

In addition to parcel geometry features, the \*.parcel file must also include a version key indicating what version of the OZFS standard the file is consistent with. The version described in this paper is 0.5.0.

## Building characteristics

Building characteristics for a single building are stored in json file with the structure illustrated in [Figure 5](#fig-bldg-data). The file includes three arrays:

* bldg\_info includes information on the characteristics of the overall building (building dimensions and number of parking spaces within the structure).
* unit\_info includes information in each type of unit within the building,
* level\_info contains information on each level within the building, and
* parking\_info optionally contains information about each on-site parking area.

|  |
| --- |
| Figure 5: Structure of \*.bldg file |

### Building dimensions

All building dimensions are in feet. The building information array includes the height from the ground to the top of the building (height\_top), from the ground to the highest wall plate (height\_plate), as well as the building width, the building depth, and the building’s roof type (roof\_type). Refer to [Appendix C](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-c.html) for an illustration of roof types that are defined for use in OZFS.

As noted in the section on zoning constraints, there are differences among zoning codes with regards to how a building’s height is defined for various roof types. For roof types other than a flat roof, the eave height must be specified in the height\_eave key. For Mansard roofs, height\_deck must be specified as well.

If the building includes towers, chimneys, antennas, or mechanical structures, the (maximum) height of these (from the roof) can optionally be specified with the height\_tower key.

The sep\_platted key is used to indicate whether each until in the building would be on a separately platted parcel (this may be used to determine whether the building meet’s a municipality’s definition of a townhome). There is also an optional parking key to indicate the number of parking spaces contained within the building’s structure (i.e. in a garage).

### Level information

The level array includes, for each level of the building, a two-element array with the level number and the gross floor area (in square feet) of that level. Above-ground levels are numbered with positive sequential numbers beginning at one (for the lowest above-ground level), and below-ground level are numbered with numbers decreasing from negative one (for the level closest to the ground).

### Unit information

The unit array includes an array specifying the following characteristics for each unit type, where units are classified as being of the same type if they have the same values for each characteristic below:

* fl\_area: The floor area of the unit in square feet.
* bedrooms: The number of bedrooms in the unit, expressed as a whole number with a minimum value of zero (for a studio unit).
* entry\_level: The level number the entrance to the unit is on.
* outside\_entry: A binary value indicating whether the entrance to the unit is directly from the outside of the building.

In addition to the four characteristics above, there is also a qty key to indicate how many units of each type are in the building.

# Sample Dataset

We have compiled a sample dataset with zoning regulations and parcel geometry for a set of 162 municipalities in the Dallas/Forth Worth region of Texas, as well as sample building characteristic data for four hypothetical residential buildings.

The \*zoning files were created using NZA data published by the Mercatus Center at George Mason University (Mercatus Center 2024) as a starting point. We converted that data for each of 161 cites in the Dallas-Forth Worth region into a \*.zoning file consistent with the OZFS data standard. Textual notes in NZA fields were used to create key:value tags that were not represented by any NZA field and/or to formulate expressions to represent context-dependent constraints. [Appendix D](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-d.html) offers a detailed description of process for converting data from the NZA data data format (as represented by the data published by Mercatus Center) to the OZFS data format. NZA data does not include municipality specific definitions for residential building types or for building heights. For XX cities[[1]](#footnote-57), we added these definitions directly from their respective municipal zoning codes. For the remaining YYY cities, we used the residential building type definitions that are implied by the NZA method that was current at the time the data were collected, as described in [Appendix D](https://vibe-lab-gsd.github.io/ozfs-standard/appendices/appendix-d.html).

The sample set of zoning regulations for 162 municipalities in the Dallas-Forth Worth region in available at {{{{dataverse}}}}} (cite dataverse).

## Parcel geometry

The \*.parcel files were derived from the 2024 Land Parcels page of the Texas Geographic Information Office (TxGIO) data hub parcel data (Various Appraisal Districts 2024) and from the road centerline files from the 2024 United States Census Bureau TIGER/Line Shapefiles (U.S. Census Bureau, Geography Division 2024). Appendix E contains details on how the \*.parcel files were assembled from these sources. and the \*.parcel files themselves are available at {{{{dataverse}}}}} (cite dataverse).

## Building characteristics

We have also created sample \*.bldg files for each of four hypothetical buildings: One two-unit building; two different four-unit buildings, and a twelve-unit building. These can be used as an example to guide the creation of \*.bldg files for other proposed buildings. They can be also be used to test applications intended to check whether a proposed building is allowable under existing zoning regulations on a particular parcel. These sample building files are available at {{{{dataverse}}}}} (cite dataverse).

# Opportunities for future development

This paper, including the accompanying appendices, represents a complete description of the data standard. The full text of the this paper (with appendices) can be found at: {{{{{}}}}}}} We welcome potential users and other interested parties to submit issues and/or pull requests with comments and suggestions to improve the usefulness of the data standard, or to note inaccuracies in the sample dataset.

The most immediate opportunities for improvement that we have identified are the treatments of overlay districts, and site grades.

## Overlay districts

The current version of the \*.zoning file structure indicates whether a district is an overlay district, but does not require any information about the regulations within overlay districts. Future extensions of the data standard could include a sensible and useful scheme for storing data on the regulations in overlay district in a way that software can easily determine how to resolve the interactions between the requirements in overlay districts and those in the base districts that they overlap.

## Site grade

The \*.parcel file currently includes information about the two-dimensional geometry of each parcel, but does not include any information about grades and elevations. This information could be relevant for some zoning analysis where the grade might have an effect, for example, on how the height of a building is measured. Future extensions of the data standard could incorporate information about grades and elevations in the \*.parcel file. Keys could also be added to the \*.zoning file to specify how site grade might influence the definitions of building heights in each municipality. Our initial test cases were in the Dallas-Forth Worth region, where there is minimal variation in elevation.

# Conclusion

This data standard represents an important step towards the development of scaleable, automated methods that can facilitate strategic increases in the supply of housing in the United States through their direct use by developers and policy-makers. They also offer researchers new opportunities to uncover the ways in which zoning regulations vary across the United States.

# References

Bronin, Sara C., and Ilya Ilyankou. 2021. “How to Make a Zoning Atlas: A Methodology for Translating and Standardizing District-Specific Regulations.” Rochester, NY. <https://doi.org/10.2139/ssrn.3996609>.

Bronin, Sara C., Scott Markley, Aline Fader, and Evan Derickson. 2023. “How to Make a Zoning Atlas 2.0: The Official Methodology of the National Zoning Atlas.” Rochester, NY: Social Science Research Network. <https://doi.org/10.2139/ssrn.4476927>.

Conway, Matthew Wigginton. 2021. “If You Zone It, Who Will Come, and How Will They Travel? The Effects of Relaxed Zoning Regulations on Travel Behavior.” PhD thesis, United States – Arizona: Arizona State University. <https://www.proquest.com/docview/2533144762/abstract/3AA6C787EA7A466DPQ/1>.

Envelope. 2022. “About Envelope City. About Envelope City.” 2022. <https://medium.com/envelopecity>.

Kahn, Eli. 2024. “NZA Data Dictionary.” Fairfax, CA: Mercatus Center, George Mason University. <https://www.mercatus.org/sites/default/files/2024-09/zoning_atlas_data_dictionary_1.pdf>.

Mercatus Center. 2024. “State and Regional Zoning Atlas Datasets Mercatus Center.” August 2024. <https://www.mercatus.org/state-and-regional-zoning-atlas-datasets>.

Metropolitan Area Planning Council. 2025. “MAPC Zoning Atlas MAPC Zoning Atlas.” 2025. <https://zoningatlas.mapc.org/>.

National Zoning Atlas. 2025a. “Explore the Atlas. National Zoning Atlas.” 2025. <https://www.zoningatlas.org/atlas>.

———. 2025b. “How the National Zoning Atlas Is Made. National Zoning Atlas.” 2025. <https://www.zoningatlas.org/how>.

———. 2025c. “Massachusetts Zoning Atlas. National Zoning Atlas.” 2025. <https://www.zoningatlas.org/massachusetts>.

———. 2025d. “National Zoning Atlas Team. National Zoning Atlas.” 2025. <https://www.zoningatlas.org/team>.

———. 2025e. “Terms of Use. National Zoning Atlas.” 2025. <https://www.zoningatlas.org/terms>.

———. 2025f. “National Zoning Atlas. National Zoning Atlas.” May 12, 2025. <https://www.zoningatlas.org>.

Open Knowledge Foundation. 2015. “Open Definition 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge.” November 2015. <https://opendefinition.org/od/2.1/en/>.

OpenStreetMap contributors. 2024. “OpenStreetMap.” <https://www.openstreetmap.org>.

Reardon, Tim, Jessie Partridge, and Lily Perkins-High. 2020. “Introduction to the Zoning Atlas.” Boston: Metropolitan Area Planning Councul. <https://zoningatlas.mapc.org/reports/1>.

Schwabacher, Mark. 2022. “ATL Zoning Code Explorer.” 2022. <https://sites.gatech.edu/atlzoningexplorer/atl-zoning-code-explorer/>.

U.S. Census Bureau, Geography Division. 2024. “TIGER/Line Shapefiles: Roads Layer (Äll Roads/̈ Linear Features).” Shapefile (GIS vector), downloaded from the 2024 TIGER/Line Shapefiles. <https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html>.

Various Appraisal Districts. 2024. “Land Parcels.” Web. <https://data.geographic.texas.gov/collection/?c=d0f7da13-ab09-4994-a16f-d52589e2476e>.

Zoneomics. 2024. “Zoneomics Find Useful Data on Zoning and Land Use.” 2024. <https://www.zoneomics.com/>.

1. Cockrell Hill, Dallas, Garland, Highland Park, Irving, Rowlett, and University Park. [↑](#footnote-ref-57)