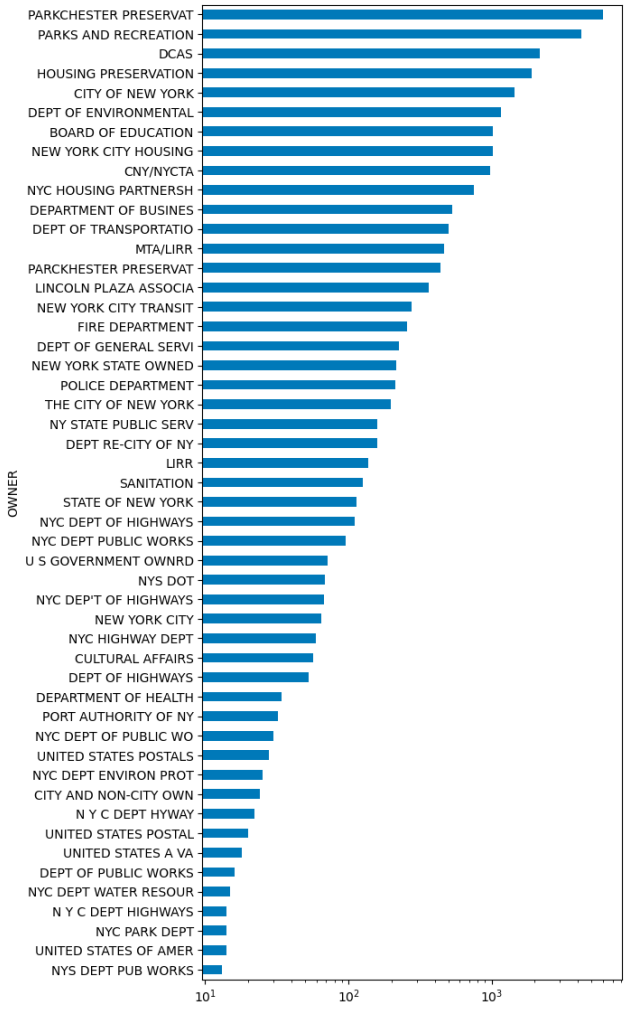
Describe Cleaning and Variables

1. Remove Exclusions
    a. Initially, the dataset consists of a specific number of records (numrecords_orig). This count represents the total dataset size before any exclusions are applied.
    b. Removing Government Easements
        i. The first exclusion step targets records with an "EASEMENT" marked as "U", indicating government use. These records are removed because they are not relevant to the analysis focusing on non-governmental property transactions. The difference between the original record count and the count after this removal provides the number of records excluded in this step, which is stored and printed as numremoved. The number of records removed is 1.
    c. Identifying Owners for Exclusion
        i. A list of keywords (gov_list = ['DEPT ', 'DEPARTMENT', 'UNITED STATES','GOVERNMENT',' GOVT ', 'CEMETERY']) that typically signify government or cemetery ownership is compiled. These keywords are used to filter out owners from the dataset. The script checks each owner name against this list, excluding names that contain any of the keywords but ensuring names with 'STORES' are not incorrectly removed to avoid excluding commercial entities erroneously. Here, the total owner number before removing is 863347.
    d. Analyzing Frequent Owners
        i. The script identifies the top 20 most frequently occurring owner names in the dataset. This step aims to uncover any large entities or frequently appearing names that might skew the analysis if not excluded. Additional names known to represent government entities are manually appended to the exclusion list such as ‘THE CITY OF NEW YORK’ , ‘NYS URBAN DEVELOPMENT’...
    e. Refining and Applying Exclusions
        i. Further refinement of the owner exclusion list is undertaken by manually reviewing and removing any names that should not be excluded, such as those mistakenly identified or irrelevant to government or cemetery associations. The comprehensive exclusion list is then applied to the dataset, removing all records associated with these owners.
    f. For now, we have removed numbers of records are 26501.

g. After applying the exclusions, a bar chart is generated to visually inspect the frequencies of the remaining owner names. This visual confirmation helps ensure that no significant entities that could bias the analysis remain. The final count of records removed through these exclusions is calculated by comparing the adjusted dataset size to the original, confirming the extent of data reduction.



h. After thoroughly executing these exclusion steps, the dataset is significantly refined, ensuring that it primarily includes records pertinent to the analysis. By meticulously removing entries associated with government-owned properties and cemeteries, we have enhanced the dataset's relevance and quality for detecting anomalies in non-governmental property transactions. This careful preparation sets a robust foundation for the subsequent phases of data cleaning, including field imputation and detailed anomaly detection

analysis. This meticulous approach ensures that the analysis will be based on the most relevant and accurate data available, maximizing the potential for insightful and actionable findings.

2. Fill in missing ZIP
    a. Initial Identification of Missing ZIP Codes
        i. Initially, the process begins with identifying the total number of missing ZIP codes in the dataset. Using a condition to find null values in the 'ZIP' column, it is determined that there are 20,431 missing ZIP codes. This step establishes the baseline for the imputation work that follows.
    b. Verification of Related Column Integrity
        i. Before proceeding with ZIP code imputation, it's essential to check the integrity of related columns that will be used in constructing unique identifiers. It's found that there are no missing values in the 'BORO' column, but there are 364 missing entries in the 'STADDR' column. Ensuring minimal missing data in these supporting columns is crucial for the accuracy of the subsequent imputation steps.
    c. Creation of Composite Address Identifier
        i. To facilitate a more accurate ZIP code mapping, a new column named 'staddr_boro' is created by concatenating the 'STADDR' (street address) and 'BORO' (borough) columns. This composite identifier uniquely represents each address across boroughs, which aids in associating missing ZIP codes with their correct locations based on address data.
    d. Initial ZIP Code Imputation Using Address Mapping
        i. A dictionary is constructed to map these unique 'staddr_boro' identifiers to known ZIP codes, capturing the relationships where available. This dictionary is then used to fill in missing ZIP codes by mapping back to the 'staddr_boro' in the data. This method successfully imputes 2,832 ZIP codes, reducing the total number of missing ZIPs to 17,599.
    e. Sequential ZIP Code Consistency Check
        i. Leveraging the orderliness of the dataset, a method is employed where missing ZIP codes are filled based on the consistency of neighboring ZIP values. If a ZIP code is absent and the ZIP codes of the immediate preceding and following records are the same, that

consistent ZIP code is used to fill the gap. This step fills in 9,491 more ZIP codes, bringing the number of still-missing ZIPs down to 8,108.

f. Final ZIP Code Filling Using Previous Record Values
    i. For the remaining missing ZIP codes, a straightforward approach is adopted. Each missing ZIP is filled with the ZIP code from the previous record, assuming geographical proximity implies similar ZIPs. This process fills all the remaining 8,108 missing ZIPs, resulting in no missing ZIP codes left in the dataset.
g. Data Cleanup Post-Imputation
    i. After all missing ZIP codes are imputed, the temporary 'staddr_boro' column, used solely for aiding the imputation process, is removed from the dataset.
h. The process of filling missing ZIP codes in the dataset has been successfully completed, reducing the initial 20,431 missing entries to zero through a structured approach involving address mapping, sequential consistency checks, and direct carryovers. This has significantly enhanced the dataset's completeness and accuracy, ensuring it is well-prepared for further analysis or modeling

3. FULLVAL, AVLAND, AVTOT
    a. FULLVAL
        i. Initially, the number of properties listed with a FULLVAL of zero was identified to be 10,025. These entries potentially represent incomplete or unassessed properties.
        ii. To facilitate imputation, all zero values in the FULLVAL field were converted to NaN (null values), allowing for more standardized filling methods. After this conversion, the total number of missing values in FULLVAL remained 10,025, as no previously missing values existed in the dataset.
        iii. The dataset was grouped by TAXCLASS, BORO, and BLDGCL (building class), and the missing values in FULLVAL were filled using the mean FULLVAL of each group. After this step, the number of missing values was reduced to 7,307, indicating that not all groups had sufficient data to compute a mean.
        iv. To address the remaining missing values, a less granular grouping was used, this time only by TAXCLASS and BORO. This reduced the number of missing values further to 386, showing an improvement but still leaving some entries unfilled.

v. The last step involved grouping by TAXCLASS alone, ensuring that all properties at least had a fallback mean value based on their tax classification. This step successfully filled all remaining missing values, resulting in zero missing values in the FULLVAL field.

b. AVLAND
    i. Initially, the dataset was examined to identify how many properties had an AVLAND value of zero, indicating potentially unassessed or incorrectly recorded entries. The total count of properties with a zero value was found to be 10,027.
    ii. To facilitate imputation, all zero values in the AVLAND field were replaced with NaN (null values). This conversion was necessary to standardize missing values and prepare for data imputation. After replacing zeros with NaNs, the total number of missing values in AVLAND remained at 10,027.
    iii. The first imputation attempt involved grouping the data by TAXCLASS, BORO, and BLDGCL (building class). Within each group, missing AVLAND values were filled using the mean AVLAND value of the group. This step aimed to preserve the valuation characteristics that are specific to the property type and location. However, after this step, 7,307 values remained unfilled, indicating incomplete groups or those without sufficient data to calculate a mean.
    iv. To address the still-missing values, a less granular approach was employed by grouping the properties only by TAXCLASS and BORO. The missing values were again filled using the mean of these new groups. This reduced the number of missing values to 386, showing that broader group averages could provide a fallback for more properties.
    v. In the final imputation step, the dataset was grouped solely by TAXCLASS. This grouping ensured that every missing AVLAND value had at least a tax class-specific mean value for filling, regardless of borough or building class distinctions. This step successfully filled all remaining missing values, bringing the total number of missing values in the AVLAND field to 0.
    vi. This detailed, tiered approach to imputing missing values in the AVLAND field ensures that each property's land valuation is estimated as accurately as possible with the available data. By progressively broadening the grouping criteria, the imputation process effectively minimized the number of properties with undefined actual land

values, enhancing the dataset's overall quality and usability for further analysis.

c. AVTOT
    i. The process starts by identifying properties in the dataset where AVTOT (total assessed value) is recorded as zero. A total of 10,025 properties initially had a zero value, indicating incomplete or potentially incorrect assessments.
    ii. All zero values in the AVTOT field were replaced with NaN to standardize the handling of missing data. This conversion facilitates more uniform data imputation strategies. After this step, there were 10,025 missing values in the AVTOT field.
    iii. The initial imputation strategy grouped properties by TAXCLASS, BORO, and BLDGCL (building class). The mean AVTOT of each group was used to fill missing values. This approach respects the property classification and location, ensuring that the imputed values are as realistic as possible. However, after this step, 7,307 values remained unfilled, indicating that some groups lacked enough data to calculate a meaningful average.
    iv. To further reduce the number of missing values, the dataset was grouped by just TAXCLASS and BORO. The missing AVTOT values were again filled using the mean of these broader groups. This step reduced the number of missing values to 386, demonstrating that a less detailed grouping could still provide useful imputation values for many properties.
    v. The last imputation step involved grouping the properties solely by TAXCLASS. Every missing AVTOT value was filled using the mean AVTOT of the respective tax class. This final step successfully filled all remaining missing values, resulting in zero missing values in the AVTOT field.
    vi. This hierarchical, step-by-step imputation process has ensured that all missing or incorrect entries in the AVTOT field were addressed comprehensively.

4. Fill in the missing STORIES
    a. The first step was to assess the extent of missing data in the STORIES field. It was found that 42,030 entries lacked this information, representing a considerable portion of the dataset that required attention for accurate property analysis.

b. To address these missing values, the most common number of stories (mode) for buildings was calculated within groups defined by their BORO and BLDGCL (building class). This grouping strategy was chosen because building heights can vary significantly across different boroughs and types of buildings. The mode represents the most frequently occurring value in the dataset, making it a realistic choice for filling in missing STORIES. After applying this imputation method, the number of missing entries was reduced to 37,922.
c. For the remaining missing values, a second imputation step was employed, grouping buildings by TAXCLASS. Within each tax class group, missing STORIES values were filled using the mean number of stories. This step assumes that properties within the same tax class will have similar structural characteristics, including the number of floors. This method successfully filled all remaining missing values, reducing the count of missing STORIES to 0.
d. After completing the imputation steps, a verification showed that there were no remaining missing values in the STORIES field. The dataset's head was displayed to ensure that the STORIES values were appropriately filled and to confirm the overall integrity and consistency of the data following the imputation process.
e. This methodical approach to filling in missing STORIES ensured that each property's record was completed in a manner consistent with its characteristics and those of similar properties.

5. Fill in LTFRONT, LTDEPTH, BLDDEPTH, BLDFRONT with averages by TAXCLASS
    a. Firstly, it was identified that LTFRONT, LTDEPTH, BLDDEPTH, and BLDFRONT fields had zero values, which are invalid for dimensional measurements. These zero values needed to be treated as missing data to accurately represent property dimensions.
    b. LTFRONT
        i. Initial Missing Values: Initially, 160,565 entries for LTFRONT were identified as missing or zero and converted to NaNs for accurate processing.
        ii. Mean values for LTFRONT were calculated and applied within each group formed by TAXCLASS and BORO. This reduced the NaN count significantly, but some values remained NaN, particularly in groups lacking sufficient data.

iii. Action: A broader grouping by TAXCLASS alone was used to fill remaining NaNs using the mean of this group. This step successfully filled all remaining missing values, reducing the NaN count to zero.

c. LTDEPTH
    i. There were 161,656 missing or zero values for LTDEPTH.
    ii. The dataset was grouped similarly by TAXCLASS and BORO, and means were applied. Some entries still lacked data due to insufficient group data.
    iii. Remaining NaNs were filled using the mean values calculated from grouping by TAXCLASS alone.
    iv. All missing values were successfully filled, with the final NaN count at zero.
d. BLDFRONT
    i. Initially, there were no missing values identified explicitly before the grouping imputation.
    ii. Avenues included grouping by combinations of TAXCLASS, BORO, and BLDGCL, then by TAXCLASS and BORO, and finally by TAXCLASS alone, applying mean values accordingly.
    iii. Each step confirmed no remaining NaNs, effectively handling any potential missing data thoroughly through the grouping strategies.
e. BLDDEPTH
    i. Similar to BLDFRONT, no missing values were mentioned before processing.
    ii. The same tiered grouping strategy as BLDFRONT was applied.
    iii. Post-imputation checks confirmed no missing values, indicating a comprehensive coverage and filling of any potential gaps.
f. This detailed and hierarchical imputation process for building and lot dimensions ensured that all properties in the dataset had realistic and consistent dimensional values.

6. Conversion of ZIP Code Data Type:
    a. The ZIP field, originally stored as a float (which could lead to incorrect ZIP code formats due to rounding or dropping of leading zeros), is converted to a string format. This conversion is crucial for maintaining the accuracy of ZIP codes, particularly when they are used in analyses that require precise geographical identification.
    b. The conversion is done using the astype(str) method, which transforms each ZIP code entry into a string, preserving all characters and avoiding any data loss that comes from numerical representation.

7. Extraction of the First Three Digits of ZIP Codes
    a. A new column zip3 is created to store only the first three digits of each ZIP code. The first three digits of a ZIP code are often used to broadly categorize geographical areas, making this reduction useful for analyses that require a more general location indicator without the need for precise ZIP code data. This is achieved by slicing the first three characters from the ZIP string.
8. Summary
    a. All columns representing dimensional data (like LTFRONT, LTDEPTH, BLDDEPTH, BLDFRONT) had zero values converted to NaNs, recognizing these as missing data and preparing them for accurate imputation.
    b. Missing data for dimensions and other important property characteristics were imputed using a hierarchical strategy based on TAXCLASS and sometimes additional characteristics like BORO or BLDGCL.
    c. The ZIP codes were normalized by converting them into string format to preserve leading zeros and extracting the first three digits for broader geographical categorization, useful in macro-level analyses.

9. The logic for all variables
    a. Creation of Derived Variables
        i. Lot Size Calculation (ltsize): Computes the product of lot frontage (LTFRONT) and lot depth (LTDEPTH) to get the total lot area, adding a small number (epsilon) to avoid division by zero in subsequent calculations.
        ii. Building Size Calculation (bldsize): Calculates the building area by multiplying building front (BLDFRONT) by building depth (BLDDEPTH), also adding epsilon.
        iii. Building Volume Calculation (bldvol): Multiplies the building area (bldsize) by the number of stories (STORIES) to estimate the total volume of the building, including epsilon.
    b. Ratio Variables
        i. R1-R9 represent different ratios used to compare property values to physical dimensions:
            1. r1, r2, r3: These ratios relate the total property value (FULLVAL) to the lot size, building size, and building volume, respectively.
            2. r4, r5, r6: Similar to the above, but using land value (AVLAND) instead.
            3. r7, r8, r9: Use the total property assessment value (AVTOT) for comparison.

c. Scaling of Ratio Variables
    i. Each of the nine ratio variables is then scaled by dividing by the median of that variable, normalizing them to a consistent scale and making them easier to compare across the dataset.
d. Inversion of Ratio Variables
    i. To facilitate the detection of very low outliers (which are close to zero and thus not many standard deviations below the mean), the inverse of each ratio variable is calculated. This transformation turns small values into large outliers, which can be more easily detected.
e. Selection of Max Values
    i. For each property, the maximum value between the original and the inverse ratio is retained. This method ensures that both extremely high and extremely low values are captured as potential outliers.
f. Removal of Inverse Columns
    i. Once the maximum values are determined, the inverse columns are dropped since they are no longer needed, simplifying the dataset.
g. Group Normalization
    i. Additional derived variables are standardized by grouping by logical categories such as ZIP code and tax class. This step involves:
    ii. Calculating the mean of each ratio variable for each group (zip5_mean and taxclass_mean).
    iii. Creating new variables for each ratio that represent the ratio divided by the group mean, allowing for comparisons of each property against its group's typical values.
h. Additional Derived Variables
    i. value_ratio: A new variable combining the full property value with land and total values, normalized across the dataset.
    ii. Handling of Extremely Low Values: For value_ratio, where the variable is below one, the inverse is used if it is larger, ensuring that low outliers are emphasized.
i. Summary
    i. This thorough approach ensures that each property's values are not only compared to its physical dimensions but also to its peers within the same ZIP code or tax class, enhancing the detection of anomalies. The final dataset consists of these refined variables, now ready for unsupervised modeling to identify potentially fraudulent or unusual properties. This preprocessing prepares the data effectively by highlighting extremes in property characteristics relative to assessed

values and physical dimensions, crucial for spotting anomalies in a real estate dataset.

10. Variable list:

Description	# Variables_Created
Ratios based on property dimensions and valuation metrics R1-r9: These ratios compare different valuation metrics (FULLVAL, AVLAND, AVTOT) to property size dimensions (ltsize, bldsize, bldvol).	9
Normalized ratios by ZIP code r1_zip5 to r9_zip5: These variables are standardized versions of r1 to r9 ratios, normalized by the average values for their respective groups defined by ZIP code (ZIP)	9
Normalized ratios by tax class r1_taxclass to r9_taxclass: These variables are standardized versions of r1 to r9 ratios, normalized by the average values for their respective groups defined by tax classification (TAXCLASS).	9
Value ratio A composite metric that compares the total property value (FULLVAL) to the sum of the land (AVLAND) and total assessed values (AVTOT). This ratio is then normalized across the dataset to identify properties whose valuation ratios significantly deviate from the norm.	1
Size ratio This ratio measures the building size relative to the lot size, providing a metric of how much of the lot is covered by the building. This can indicate properties that are unusually developed relative to their lot	1

size, which might be of interest in certain urban planning or zoning analyses.	