Lab 2 - Data Wrangling

- **Due** Nov 30, 2023 by 11:59pm

- **Points** 100

- **Submitting** a file upload

TL;DR: For lab 2, you will begin with a dataset with nontrivial data quality issues. You need to profile the dataset to identify the specific data quality issues and then use Tableau Prep to clean and transform the raw data into a format you can load into a provided relational database schema.

Updates:

Updates or clarifications made after the original lab is published will be listed here. All modifications in the text below that differ from the original assignment spec are flagged by green text in the document below.

- *Thursday, Nov 23 - lab due date extended by two days*
- *Wednesday, Nov 15 - uploaded revised empty destination database structure that adds AnnualSalesByNeighborhood.BoroughName to the primary key for that table.*
  - New database shell -MDM M2-23 - Lab 2 NYC Real Estate DB - empty (v1_1).dbDownload MDM M2-23 - Lab 2 NYC Real Estate DB - empty (v1_1).db

Assignment Overview:

In this lab, you will wrangle and clean data from multiple sources to create an analytic dataset that analysts at your real estate investment firm can use to understand market trends and price estimates for real estate parcels in New York City.

Your primary data source is a collection of data about every real estate transaction recorded in New York City from Q3-2019 through Q3-2023. That transactional data is contained in the following Excel workbook:

NYC Real Estate Combined Dataset 2019-2023.xlsxDownload NYC Real Estate Combined Dataset 2019-2023.xlsx

This dataset is derived from public real estate transaction records published by the New York City government at

http://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page

To better understand what this file contains, review the information available at this site, especially the descriptions and details about the data at the following two links:

-
    - Glossary and Excel File Use Information
    - NYC Building Class Code Descriptions

You are welcome to pull down additional data from the site if you would like to. This four-year data set should be sufficient to give you a strong baseline for analysis while still being a manageable size for data wrangling and loading into a relational database.

In addition to the core transactions dataset, you will also need to integrate data from the following sources to create your final database:

- NYC Building Class Code Descriptions - You will need to do some screen-scraping to recreate the table mapping building classification codes to classification descriptions. I suggest a simple copy-and-paste operation from the web into MS Excel, then cleaning the HTML data by hand to make a simple table readily imported into your target SQLite database.
- Vacant storefront dataLinks to an external site.. New York City's government tracks vacant storefronts through property owner filings and makes historical data about vacant storefront filings publicly available at:
    - https://data.cityofnewyork.us/City-Government/Storefronts-Reported-Vacant-or-Not/92iy-9c3nLinks to an external site. .
    - Use this CSV file containing vacant storefront filing data from 2019 through 2022 as your source data to clean and load into the SQLite database. This data was downloaded from the above website.
        - Storefronts_Reported_Vacant_or_Not.csvDownload Storefronts_Reported_Vacant_or_Not.csv
    - The following data dictionary describes this dataset (again, from the website listed above).
        - Vacant Storefronts Data Dictionary.xlsxDownload Vacant Storefronts Data Dictionary.xlsx

Target database structure:

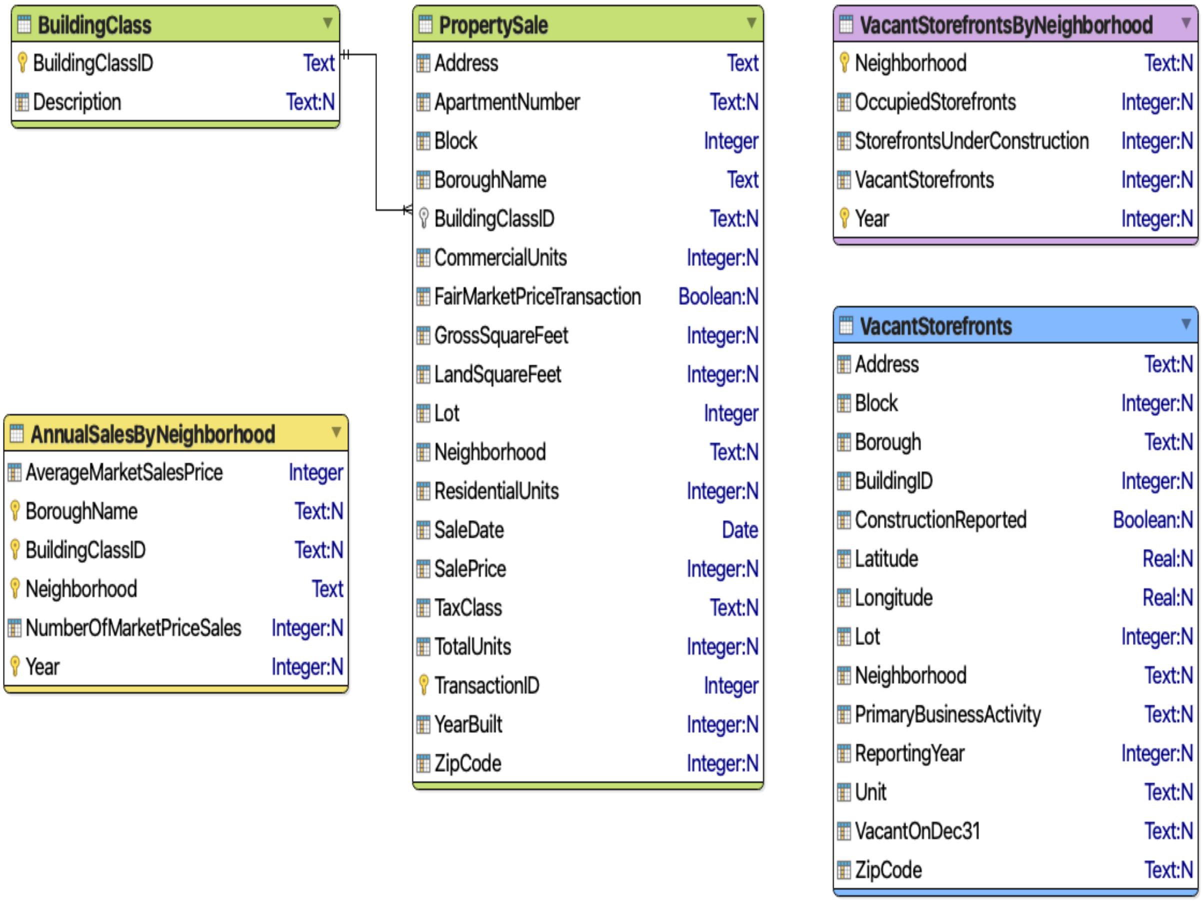
You need to clean and shape/transform the provided data using Tableau Data Prep and then load it into the SQLite relational database provided below. Note that a SQLite database runs locally on your laptop and can be quickly packaged as a single file to be submitted through Canvas for grading. I recommend setting up a folder on your laptop to store the Excel workbook with the source data, the SQLite database file, your Tableau Prep flow file (.tfl), and all other files and data sources you require for the lab.

~~DMF F23 - Lab 5 NYC Real Estate DB - empty (v1_0).db~~

MDM M2-23 - Lab 2 NYC Real Estate DB - empty (v1_1).dbDownload MDM M2-23 - Lab 2 NYC Real Estate DB - empty (v1_1).db

The provided database is an empty shell containing the tables you need to fill with your appropriately cleaned and wrangled data. Assume that you cannot modify the target database's structure. As you will often find in corporate data management situations, assume the target database structure is used by applications running the business day-to-day and thus not something that is readily modified for data analytics purposes. You will typically need to adapt the structure of the data you are loading to match the target database structure rather than vice-versa. Do that here as well.

ER Diagram for target SQLite database:



BuildingClass	
🔑 BuildingClassID	Text
Description	Text:N

PropertySale	
Address	Text
ApartmentNumber	Text:N
Block	Integer
BoroughName	Text
🔑 BuildingClassID	Text:N
CommercialUnits	Integer:N
FairMarketPriceTransaction	Boolean:N
GrossSquareFeet	Integer:N
LandSquareFeet	Integer:N
Lot	Integer
Neighborhood	Text:N
ResidentialUnits	Integer:N
SaleDate	Date
SalePrice	Integer:N
TaxClass	Text:N
TotalUnits	Integer:N
🔑 TransactionID	Integer
YearBuilt	Integer:N
ZipCode	Integer:N

VacantStorefrontsByNeighborhood	
🔑 Neighborhood	Text:N
OccupiedStorefronts	Integer:N
StorefrontsUnderConstruction	Integer:N
VacantStorefronts	Integer:N
🔑 Year	Integer:N

AnnualSalesByNeighborhood	
AverageMarketSalesPrice	Integer
🔑 BoroughName	Text:N
🔑 BuildingClassID	Text:N
🔑 Neighborhood	Text
NumberOfMarketPriceSales	Integer:N
🔑 Year	Integer:N

VacantStorefronts	
Address	Text:N
Block	Integer:N
Borough	Text:N
BuildingID	Integer:N
ConstructionReported	Boolean:N
Latitude	Real:N
Longitude	Real:N
Lot	Integer:N
Neighborhood	Text:N
PrimaryBusinessActivity	Text:N
ReportingYear	Integer:N
Unit	Text:N
VacantOnDec31	Text:N
ZipCode	Text:N

Required data cleaning and shaping steps:

You must complete the following tasks to integrate, clean, shape, and load the requested data into the target database structure. Instructions are given for each of the tables in the target database.

BuildingClass table: To fill the BuildingClass table, you need to retrieve the data contained on the website NYC Building Class Code Descriptions so that you can create an output file named BuildingClass.csv that has the following structure - exactly these heading names, in exactly this order:

```
BuildingClassID, Description
```

PropertySale table: You will find that the property transactions dataset is far from perfect. The data is scattered across multiple Excel worksheets. There are records with missing data, wildly incorrect values, and contradictory values scattered throughout. Figuring out how to integrate this data from multiple source worksheets, format it, clean it, deal with missing or seemingly incorrect data, etc., reasonably and consistently is ***the*** core task of this assignment.

You need to complete *at least* the following data shaping and cleansing steps to load data into the PropertySale table:

- Deal appropriately with duplicate records. This includes both exact duplicates and near duplicates.
- Handle data quality problems with missing fields or fields that are so far out of their expected value range that they are likely incorrect. This includes but is not limited to issues with unrealistically low (or high) transaction prices.
    - *Note - there are many transaction records for transactions with 0 gross square feet and/or 0 land square feet. This should appear to likely be data errors. These are not, however, necessarily errors. Property ownership in New York City has some unusual legal structures, such that a person purchasing a condominium or a co-op apartment may not actually have any legal right to the specific building and/or the land on which the building sits. They are just purchasing the right to live in a specific unit within the building. Those transactions are then recorded as having 0 sq feet of land or building changing hands. Since these are very common real estate holdings in New York City, many transactions show 0 for these fields. Further, the data is not consistently recorded across the years. For the 2021 sales, these fields are null instead of 0, for 2018 and 2019 they are set to 0. I recommend setting all 0 sq feet sales fields (land or gross) to null and assuming they are not data errors. Likewise, many sales in 2021 seem to have a null value for the number of commercial, residential, or total units recorded in the sale.*

    - *You should choose how to handle these values in a way that will be as consistent as possible across the years.*
- Develop one or more rules to flag transactions that are likely not indicative of the fair market price for the property. These are typically transactions with very low (or zero) prices. Please review the data dictionary carefully for an explanation of what transactions recorded as $0 mean. You should decide on a rule, or set of rules, to identify transactions that likely do not represent the true market value of a property. You should add a flag for each transaction that indicates whether you believe it to be a fair market value transaction or a non-market value transaction. Flag non-market value transactions by setting the PropertySale.FairMarketPrice field to false.
    - *Note - there is no One Right Rule to determine which transactions are fair market value and which are not. It is clear that transactions with a $0 sales price are not at fair market value. What you need to do with your rule is to try to find a heuristic (rule) that makes a reasonable guess about whether a property is transferred at fair market value. I don't expect you to come up with a precise and 100% correct rule. Just one that flags most values that are likely well below market rate as such. The exercise here is about trying to look at the available data, make a decision about what might be a reasonable metric, make a (hopefully simple) rule to implement that heuristic, then test that rule to see how many (likely) false positives and false negatives it creates. You should settle on a rule that seems to minimize each of those errors, but you are unlikely to create a single simple rule that eliminates all of them. Developing good judgment about when you've cleaned the data to an appropriate level of quality for your analytic and record-keeping purposes (even though some errors likely still exist in the dataset) is an important part of developing your skills as an analyst and data scientist.*

- Remove fields that add no information (ie are null for all records) or are not required in the target database structure.
- Convert the borough identifier for each property sales record (stored as an integer) into the appropriate name for that borough. There are five boroughs in New York City - Manhattan, Bronx, Brooklyn, Queens, and Staten Island. You need to figure out which number corresponds to which borough (check the data dictionary) and add a step in your Tableau Data Prep flow to convert the id to a borough name.
- Decide which fields to use to come up with a final tax class and building class for each transaction.
    - There should be only one value for building class and tax class for each transaction in the target database; figure out which one makes the most sense for the transactions and eliminate the others.
- Properly separate street addresses and apartment numbers. You will likely find that some PropertySale records have the apartment number appended to the end of the street address field and others have it properly listed in the apartment_number column. To the extent possible, you should ensure

that all transactions with an apartment number have that apartment number listed in the apartment_number column and that they do not have a redundant apartment number appended to the address field.

The result of your Tableau Flow for working with the transactions dataset should be to create an output file called PropertySales.csv that contains the cleaned, integrated data ready to be loaded into your PropertySales table in SQLite. The output file needs to be properly formatted in Comma Separated Value (.csv) format using exactly the following field headers, listed with exactly these heading names, in exactly this order:

```
Address, ApartmentNumber, Block, BoroughName, BuildingClassID,
  CommercialUnits, FairMarketPriceTransaction, GrossSquareFeet, LandSquareFeet, Lo
t,
  Neighborhood, ResidentialUnits, SaleDate, SalePrice, TaxClass, TotalUnits,
  YearBuilt, ZipCode
```

You do not need to provide a TransactionID field for each property sale. There are no TransactionID's provided in the source dataset so the skeletal database schema is configured to automatically generate a unique transaction ID for each sale record as it is INSERT'ed into the table.

AnnualSalesByNeighborhood table: You need to create a table to store aggregated summary data about market conditions in various neighborhoods for each year of data in the source dataset. To do so, you must filter out all non-market price transactions and then build pipeline steps to calculate the total number of market-price transactions organized by year, neighborhood and building classification, along with the average transaction price for every one of those (Year, Neighborhood, BuildingClassID) combination.

Your AnnualSalesByNeighborhood.csv file needs to be properly formatted in Comma Separated Value (.csv) format. Your file should have the following field headers, listed with exactly these heading names, in exactly this order:

```
Year, BoroughName, Neighborhood, BuildingClassID, NumberOfMarketPriceSales,
AverageMarketSalesPrice
```

VacantStorefronts table: You need to use Tableau Prep to clean and shape the data about vacant storefronts in NYC from the city's vacant storefront data websiteLinks to an external site. to prepare it for loading into the VacantStorefronts table of the target database. Your flow needs to create an output file called VacantStorefronts.csv that can be used to directly load the data into the target database's VacantStorefronts table using the following file structure - exactly these heading names, in exactly this order:

```
Address, Block, Borough, BuildingID, ConstructionReported, Latitude, Longitude, Lo
t, Neighborhood, PrimaryBusinessActivity, ReportingYear, Unit, VacantOnDec31, ZipC
ode
```

Use the following guidelines to shape and clean the source data for vacant storefronts:

- Extract the Block and Lot fields from the BOROUGH-BLOCK-LOT or BBL fields in the source data file. (These appear to be redundant columns). The data dictionary and other online resources explain how Borough, Block, and Lot are encoded into a single numeric value in each field. Use the Google...
- You need to convert the ReportingYear field from a two-year range in the source data file into a single integer value for use in the target database (e.g. use 2019 instead of "2019 and 2020"). You can read through the provided data dictionary and other online resources to figure out how they are using these fields, but for our purposes, you can simply extract the earlier of each pair of years listed. For example, if the source's Reporting Years field is "2019 and 2020", you should convert that to the integer 2019 for the target ReportingYear field.
- For Borough, use the Borough name with normal case (e.g. Manhattan, Brooklyn) instead of using all UPPERCASE letters.
- Use the BIN field as your BuildingID
- For the ConstructionReported field, set the values to YES if it is reported and NO if it is listed as NO or it has a null or empty value.
- Remove duplicates. Deciding what constitutes duplicate rows in this dataset is a bit trickier than in the PropertySale dataset, so dig through the dataset to decide which fields must match exactly to conclude that a pair (or set) of rows are duplicates. This is mainly because it appears that a single property location (according to city property records) may have multiple businesses or storefronts operating out of it. There is not one obviously correct way to do de-duping on this dataset, so experiment a bit with different alternatives to see which of them seem to flag too many "false" duplicates, which identify too few "real" duplicates, and what balance of matching criteria seems to leave you with the cleanest dataset.

VacantStorefrontsByNeighborhood table: Use the data from the VacantStorefronts dataset to generate a summary count of storefronts that are vacant, occupied, and under construction in each neighborhood for each of the years covered by the dataset. To do so, use an Aggregation step in Tableau Prep to count the storefronts listed as vacant, not vacant (occupied), and under construction for each combination of neighborhood and year present in your cleaned and shaped VacantStorefronts data. Make sure your summary aggregation step occurs after all previous cleaning and shaping steps for the source data, including de-duping. Doing so ensures that your summary table is built from high-quality source data.

When doing the summary calculations, count a storefront as "Vacant" if it is listed as vacant on Dec. 31st of the listed year. Count it as an occupied storefront if it is not specified as vacant on that date. The count of storefronts under construction should be independent of whether the storefront is vacant or occupied, as construction sometimes requires closing a store and other times the store will keep operating while it is under construction or undergoing renovations.

Your flow needs to create an output file called VacantStorefrontsByNeighborhood.csv that can be used to directly load the data into the target database's VacantStorefrontsByNeighborhood table using the following file structure - exactly these heading names, in precisely this order:

```
Neighborhood, OccupiedStorefronts, StorefrontsUnderConstruction,
VacantStorefronts, Year
```

This list includes some of the most obvious data quality issues with the dataset, but it does not identify all possible data quality issues with the dataset, nor all of the transformations you may need to do to fill your SQLite database with high-quality data in the format requested. As you identify additional data quality concerns, you should decide how best to handle them.

Deliverables:

You need to submit the following for your lab:

- Your Tableau Data Prep flow file (.tfl) cleans and shapes the provided datasets and creates clean .csv files ready for loading into your SQLite database. I will test your .tlf file by downloading it from Canvas and connecting it to the original Excel and .csv files provided as source data. As a result, any data cleansing work that you have done in Excel will not be visible in the flow that is being graded. So... do all of your data wrangling and cleaning in your Tableau Prep flow, not in the source Excel or .csv files.
- Load your cleaned and transformed data into the required tables in your individual SQLite database. Submit the .sqlite database file containing your cleaned data to Canvas.
- If you are working in a group, ensure all group members are appropriately added to your Canvas lab group.

Groups: This lab may be completed individually or in groups of two or three students. There is a strict limit of three students per group. If you are working as a group, you should work with the same students you worked with on previous labs. If, for some reason, you are unable to work with the same partners for this lab, please contact me (Prof. Monroe) to explain the situation and what your new group situation will be.