CS513: Theory & Practice of Data Cleaning
Project Phase - II

Sim Bhamra (sbhamra2@illinois.edu)
Albert Yeh (ayeh2@illinois.edu)

Introduction

The goal of this project is to apply the data cleaning concepts and practices we learned during class to crime data from the city of Denver, Colorado. The Denver City crime data is part of an open data catalog provided by the city government and collected by the Denver Police Department. The data can be found at this location: https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-crime.

For this project we will be using two of the tables provided, the first is the crime table which contains all criminal offenses in the city and county of Denver from January 2017- June 2022. The second is the offense code table, which can be used to cross-reference the offense code given in table 1 to find the actual crime name according to the National Incident-Based Reporting System.

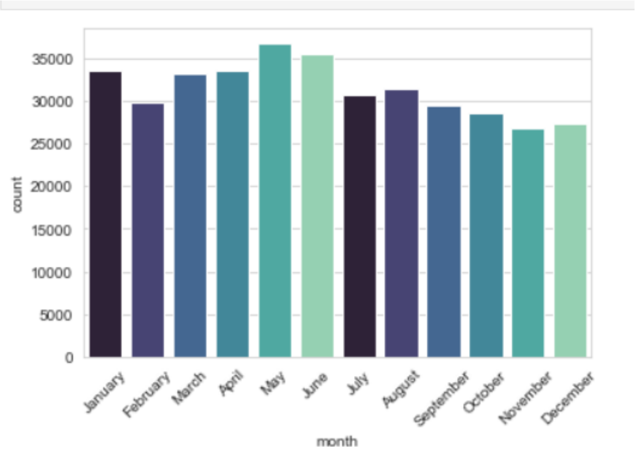
Data Cleaning

Import Datasets

The first step in our data cleaning workflow is to import both our data sets and begin to change the data to the correct types. We do this with the crimes table first using some of the data we collected during our data exploration in phase I. Our dirty datasets are crime.csv and offense_codes.csv both in our supplementary materials folder.

Step 1

First the FIRST_OCCURRENCE_DATE and REPORTED_DATE columns are updated to the correct type as planned in phase I. Initially both are of type object, we chose to convert them to pandas datetime objects and ISO format as both could be useful. Further we can see that 376812 values of FIRST_OCCURRENCE_DATE and REPORTED_DATE are changed here. Once the type has been changed, FIRST_OCCURRENCE_DATE with datetime formatting is saved as first_occurrence_date_dt and ISO formatting is saved as first_occurrence_date_isoformat and added to the crimes dataframe as new columns. Similarly with REPORTED_DATE the columns are saved as reported_date_dt and reported_date_isoformat respectively. This is done for the purpose of one of our other use cases to help determine the most and least common crimes in different time periods.



This is one of the tables that can be found in our jupyter notebook, but it shows that crimes do peak in the summer as most crime data hypothesizes. Whether due to the temperature or other reasons, there is usually less crime near the holiday seasons. By putting the dates that were initially strings, and turning them into datetime objects, we were able to plot the data into a histogram with ease.

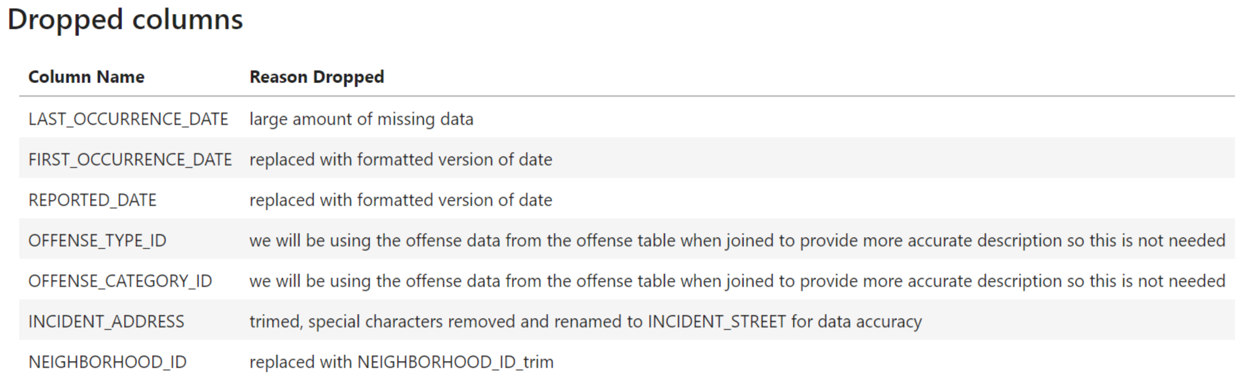
Step 2

Using .strip() the whitespace before and after the following columns: OFFENSE_TYPE_ID, OFFENSE_CATEGORY_ID, INCIDENT_ADDRESS, NEIGHBORHOOD_ID. These are columns that include strings which in pandas dataframes are converted to objects. Before performing the trim we see below that the columns don't have any leading or trailing whitespace, which we did not expect. We still run the trim function here and check for whitespaces again to confirm there are no whitespaces present.

In addition, the special characters are removed from the INCIDENT_ADDRESS column. The columns are also renamed to OFFENSE_TYPE_ID_trim, OFFENSE_CATEGORY_ID_trim, INCIDENT_ADDRESS_trim, and NEIGHBORHOOD_ID_trim. Thus far all of the steps have been planned for and executed from phase I, however the next step listed in phase I : "adjust address column name or add city, state,and zip to make a full address" is skipped. This is because, in order to make a full address with zip and city we would need to query an outside API which would go beyond the scope of this project. Instead, this column is renamed to INCIDENT_STREET which more accurately describes the data.

Step 3

Initially mentioned later in the phase 1 proposal, a few columns are dropped in this step. Columns are dropped here as well as when joining the tables as planned because many columns are now duplicated versions we have made in conversions, some columns are not needed, and it makes the dataframe easier to work with as there are fewer columns to keep track of.



Dropped columns

Column Name	Reason Dropped
LAST_OCCURRENCE_DATE	large amount of missing data
FIRST_OCCURRENCE_DATE	replaced with formatted version of date
REPORTED_DATE	replaced with formatted version of date
OFFENSE_TYPE_ID	we will be using the offense data from the offense table when joined to provide more accurate description so this is not needed
OFFENSE_CATEGORY_ID	we will be using the offense data from the offense table when joined to provide more accurate description so this is not needed
INCIDENT_ADDRESS	trimed, special characters removed and renamed to INCIDENT_STREET for data accuracy
NEIGHBORHOOD_ID	replaced with NEIGHBORHOOD_ID_trim

Step 4

Next NEIGHBORHOOD_ID_trim must be cleaned. In phase I, one of the steps was to consolidate the neighborhood ID column and remove garbage values. This ended up requiring more work than expected because of all the missing values. Since this is an important column, we calculate the haversine distance / great circle distance between each incident location and all other average neighborhood locations. Before this we ensure that columns are named correctly, check the uniqueness, and calculate the NA amount.

Source: *https://en.wikipedia.org/wiki/Haversine_formula*

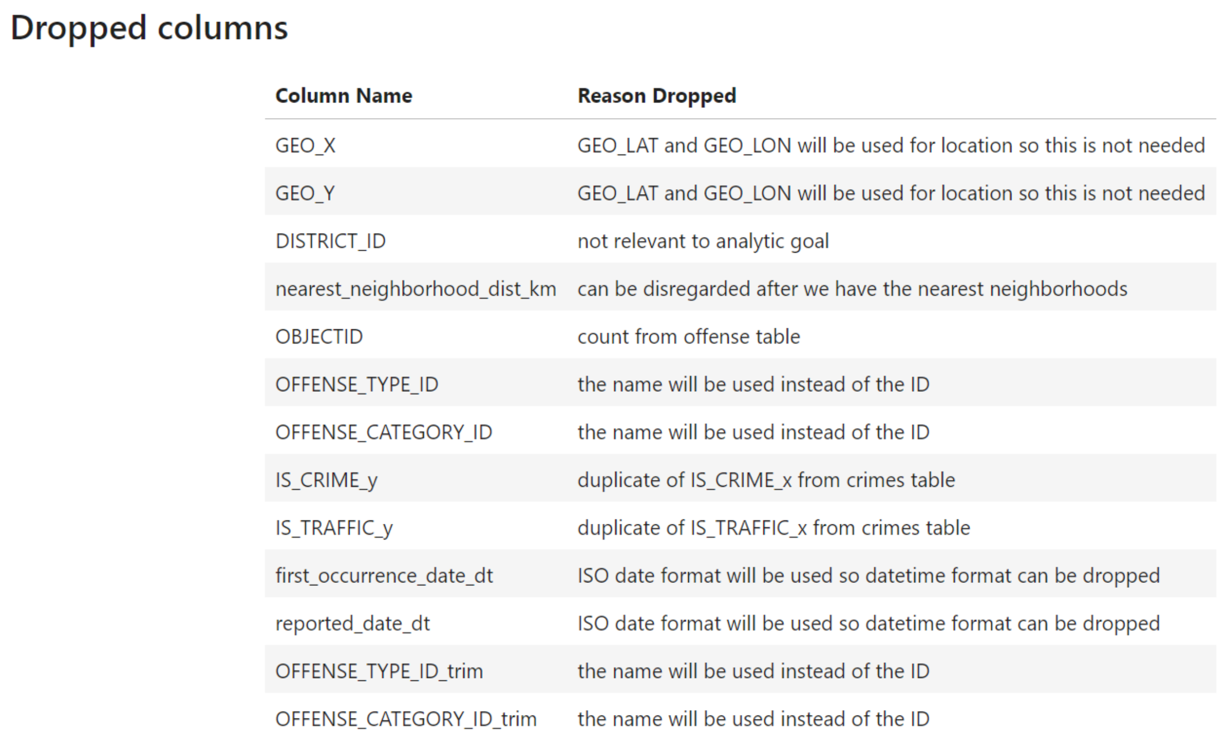
https://stackoverflow.com/questions/29545704/fast-haversine-approximation-python-pandas/29546836#29546836

Step 5

With sufficiently cleaned crimes data the offense data frame can now be joined, note we did not need to apply any cleaning to the offense data as it came already cleaned. The tables are joined to the OFFENSE_CODE and OFFENSE_CODE_EXTENSION using a left join into a new table called dv_crime_data. To ensure there are no duplicates when joining the data we check that IS_CRIME_x', 'IS_TRAFFIC_x', is equal to 'IS_CRIME_y','IS_TRAFFIC_y', the 'X' version coming from the crimes data and 'Y' version from offense codes. This step's direct purpose is to allow us to show the change in each type of crime according to the NIBRS definition in the last 5 years in the city and county of Denver (Use case 1).

Step 6

With the joined tables there are more redundant columns to drop. After dropping the columns, the column names are standardized. Additionally, the columns are renamed to give more accurate descriptions, and remove readability.



Dropped columns

Column Name	Reason Dropped
GEO_X	GEO_LAT and GEO_LON will be used for location so this is not needed
GEO_Y	GEO_LAT and GEO_LON will be used for location so this is not needed
DISTRICT_ID	not relevant to analytic goal
nearest_neighborhood_dist_km	can be disregarded after we have the nearest neighborhoods
OBJECTID	count from offense table
OFFENSE_TYPE_ID	the name will be used instead of the ID
OFFENSE_CATEGORY_ID	the name will be used instead of the ID
IS_CRIME_y	duplicate of IS_CRIME_x from crimes table
IS_TRAFFIC_y	duplicate of IS_TRAFFIC_x from crimes table
first_occurrence_date_dt	ISO date format will be used so datetime format can be dropped
reported_date_dt	ISO date format will be used so datetime format can be dropped
OFFENSE_TYPE_ID_trim	the name will be used instead of the ID
OFFENSE_CATEGORY_ID_trim	the name will be used instead of the ID

Step 7

As planned for phase 1, integrity constraints are checked. First, we check that there are no primary key violations, meaning that there will be no duplicates. Here incident_id and offense_id are used as the primary key, since a single incident can have multiple offenses occur.

With no primary key violations, the next violation to check is not null. We do not need to check for unique constraints as for the remainder of the columns values can be repeated, for example crimes can happen on the same day, or in the same neighborhood.

Referring to the code above we see that around 1.5% of the data has NA values. Looking at the table these NA values are on the columns geo_lon, geo_lat, and incident_street. This makes sense as geo_lon, geo_lat are dependent on incident_street. As the amount of NA is so low and all of the crimes are sexula assualt these we have decided to keep this data. Sexual assault is historically under reported therefore we think it important to leave this data in so we can represent the crime as well as we can. Moreover it is likely because of the nature of the crime and protection of the victim that the street address may not be reported for these crimes.

Step 8

Next the clean data is exported to dv_crime_data. csv. With the clean data the next steps include filtering and plotting the data for analysis.

Document Data Quality

Description	Which Columns Changed	Changes
Import dataset	All	376812 lines imported
Convert to date time format then to ISO	REPORTED_DATE	376812 lines changed
Conversions for reported date to ISO	FIRST_OCCURRENCE_DATE	376812 lines changed
Stripping white space from multiple cols	OFFENSE_TYPE_ID, OFFENSE_CATEGORY_ID, INCIDENT_ADDRESS, NEIGHBORHOOD_ID	0 lines changed
Remove special characters from address	INCIDENT_ADDRESS	32305 lines changed
Dropped columns that will not be used or were updated	Previously changed cols that were updated	7 columns dropped
Manually replace abbreviated neighborhood names	NEIGHBORHOOD_ID	1 unique name replaced
Find number of missing neighborhoods	NEIGHBORHOOD_ID	584 missing neighborhoods
Calculated haversine to find closest neighborhoods	CREATED nearest_neighborhood	582 added nearest neighborhoods (2 are missing longitude/latitude

Dropped columns no longer used or not relevant	GEO_X GEO_Y DISTRICT_ID nearest_neighborhood_dist_km OBJECTID OFFENSE_TYPE_ID OFFENSE_CATEGORY_ID IS_CRIME_y IS_TRAFFIC_y first_occurrence_date_dt reported_date_dt OFFENSE_TYPE_ID_trim OFFENSE_CATEGORY_ID_trim	

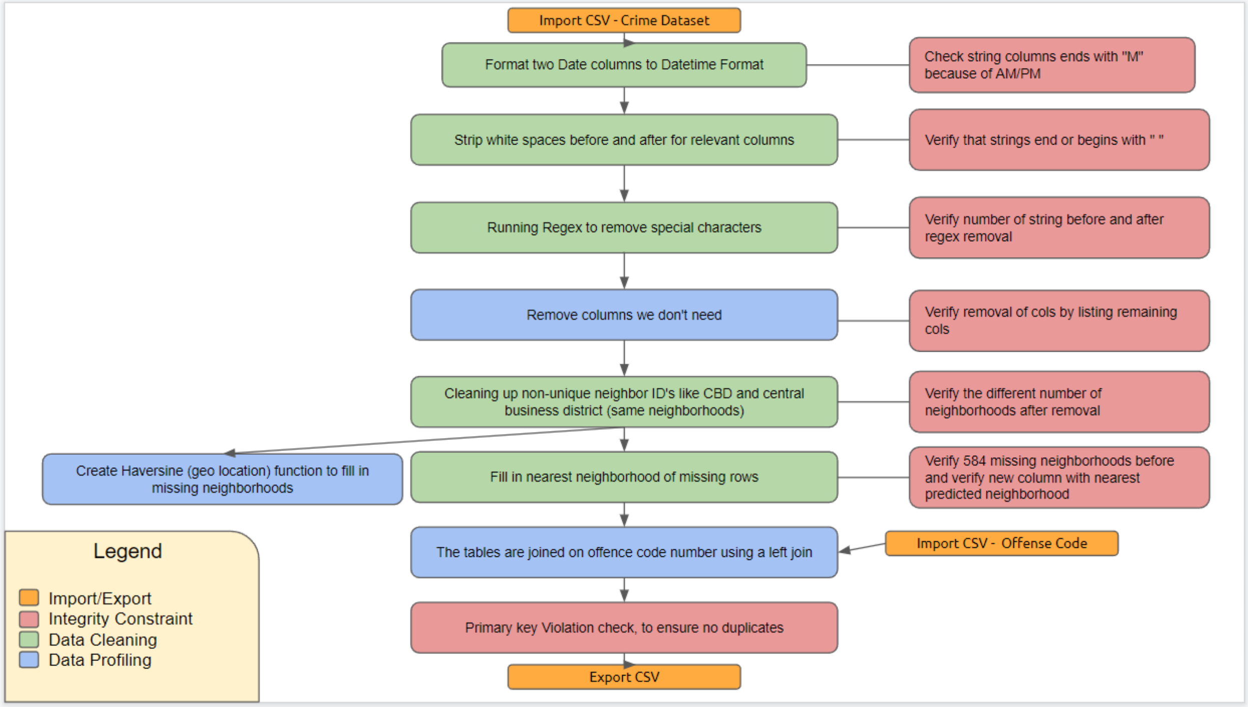
NOTE:

The quality of the data has been verified in the supplementary materials. Please see the jupyter notebook entitled: "phase_2_code_Bhamra_Yeh.ipynb"

We demonstrate that the data quality has been improved with many different before and after queries.

Workflow Model

Outer Workflow

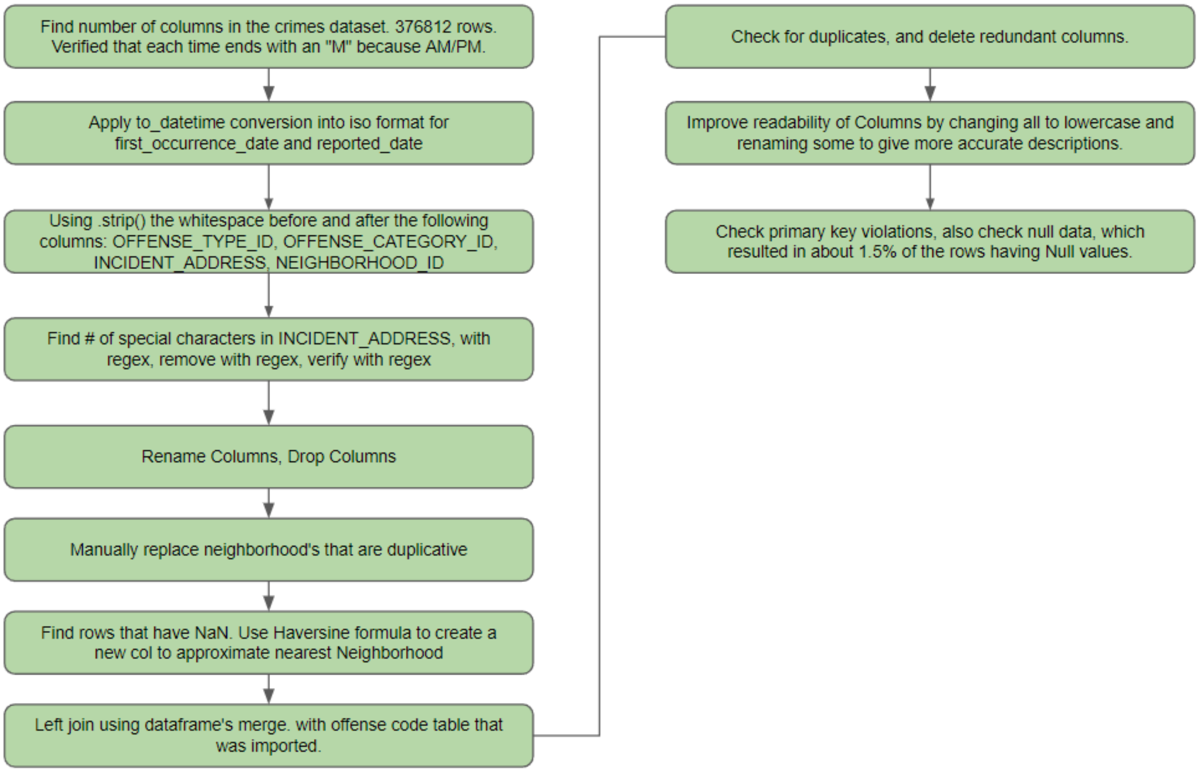

Import CSV - Crime Dataset
Format two Date columns to Datetime Format
Check string columns ends with "M" because of AM/PM
Strip white spaces before and after for relevant columns
Verify that strings end or begins with " "
Running Regex to remove special characters
Verify number of string before and after regex removal
Remove columns we don't need
Verify removal of cols by listing remaining cols
Cleaning up non-unique neighbor ID's like CBD and central business district (same neighborhoods)
Verify the different number of neighborhoods after removal
Create Haversine (geo location) function to fill in missing neighborhoods
Fill in nearest neighborhood of missing rows
Verify 584 missing neighborhoods before and verify new column with nearest predicted neighborhood
The tables are joined on offence code number using a left join
Import CSV - Offense Code
Primary key Violation check, to ensure no duplicates
Export CSV
Legend
🟧 Import/Export
🟥 Integrity Constraint
🟩 Data Cleaning
🟦 Data Profiling

We did the entire profiling, cleaning, integrity checks, import and export of the data in a jupyter notebook. We chose to use jupyter as our tool because it allowed us to primarily use python to explore the dataset, to quickly see, verify changes, ensure integrity, use regex, built-in python functions, to do the heavy lifting of the cleaning portion.

Firstly we chose python, because it contains important libraries that would allow us to manipulate the data. Python has many functions that allow us to check the integrity of our cleaning procedures. We chose the pandas library because it is well integrated into python and allowed us to import and export the csv file with ease. We used the numpy library to allow us to use their radians/sine/cosine functions for the haversine function. We used datetime library to allow us to convert native strings in a time format into a specific datetime object to allow us to analyze the data according to one of our tertiary use cases, to verify the type of crimes and when they are committed.

Our workflow model was built using a generic spreadsheet tool. We were able to show the outer workflow, which identifies our key inputs which are two different tables, one a crime data from Denver 2017-2021 to an offense code table used by the National Incident-Based Reporting System. They are joined in the end using a left join, (a merge in dataframes).

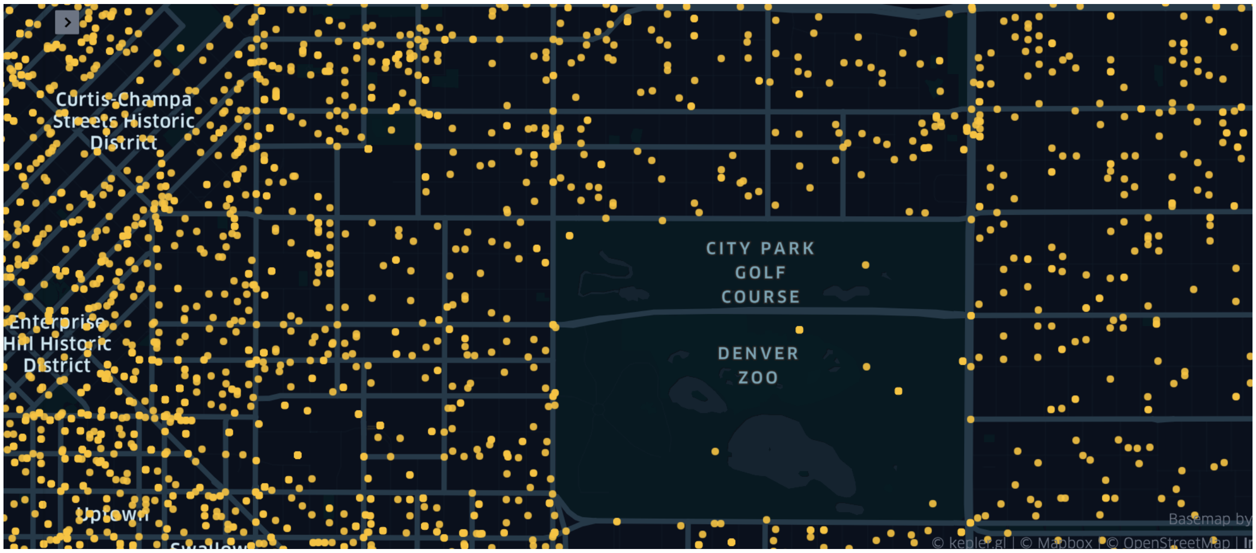
Inner Data Cleaning Workflow


Find number of columns in the crimes dataset. 376812 rows. Verified that each time ends with an "M" because AM/PM.
Apply to_datetime conversion into iso format for first_occurrence_date and reported_date
Using .strip() the whitespace before and after the following columns: OFFENSE_TYPE_ID, OFFENSE_CATEGORY_ID, INCIDENT_ADDRESS, NEIGHBORHOOD_ID
Find # of special characters in INCIDENT_ADDRESS, with regex, remove with regex, verify with regex
Rename Columns, Drop Columns
Manually replace neighborhood's that are duplicative
Find rows that have NaN. Use Haversine formula to create a new col to approximate nearest Neighborhood
Left join using dataframe's merge. with offense code table that was imported.
Check for duplicates, and delete redundant columns.
Improve readability of Columns by changing all to lowercase and renaming some to give more accurate descriptions.
Check primary key violations, also check null data, which resulted in about 1.5% of the rows having Null values.

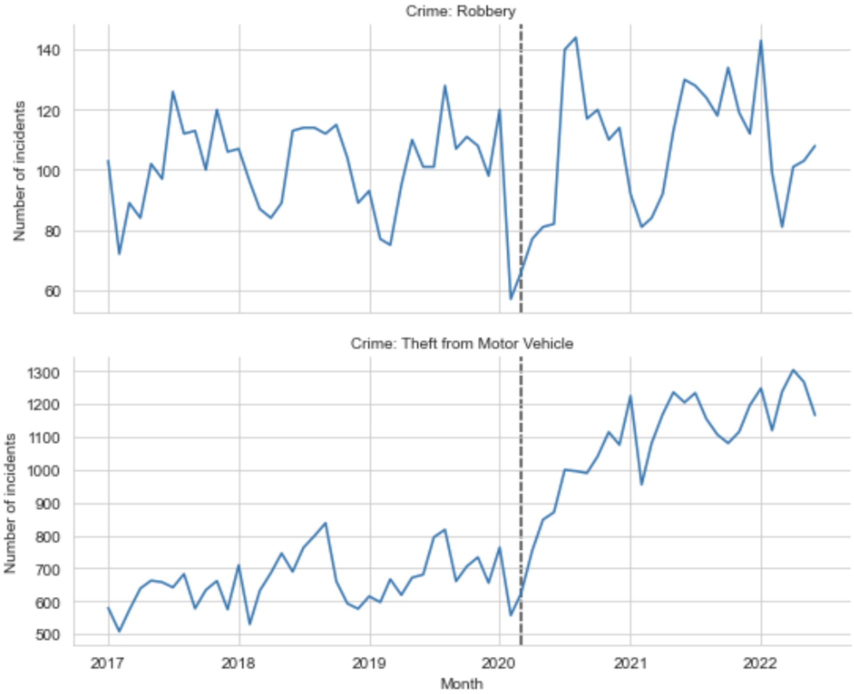
The inner data cleaning workflow shows the steps that were taking that altered the data in any way shape or form. If the columns, rows, or data were added, removed, or edited, it is captured here in the workflow. The reasons are captured in the detailed summaries in the data cleaning phase above.

Conclusions and Summary

In phase I, our goal for this project was to apply data cleaning concepts to the Denver city crime data to analyze crime trends. We believe this goal has been achieved. We performed simple summary statistics and explored locational and temporal crime trends with the cleaned data. With the location data from our cleaned data we created an interactive geoplot for each year as seen below for the year 2017.


Curtis-Champa Streets Historic District
Enterprise Hill Historic District
Uptown
Swallow
CITY PARK GOLF COURSE
DENVER ZOO
Basemap by
© kepler.gl | © Mapbox | © OpenStreetMap | In

Furthermore, the cleaned data allowed us to make clear frequency graphs for crimes per year, per month, and per month for each year. Temporal trends are seen a lot easier here than compared to the initial data, because after the cleaning process we corrected time and date formatting which allowed for easier categorizing and visualization. An additional use case was exploring the temporal crime data trends before and after covid. Example graphs for the crimes of robbery and theft from a motor vehicle are shown below. In addition, all other graphs created can be seen in the Jupyter notebook analysis section.



There were a few lessons learned throughout the project. We learned that checking data integrity and seeing if it makes common sense can save time and improve the quality of a data cleaning process. For example when using the haversine function to identify nearby neighborhoods for records where that information was missing, it was helpful to check the coordinates of each neighborhood location to ensure that they were all within the Denver area. We also learned to be careful of duplicating data during joins. We were able to mitigate this risk by double checking and removing duplicates, and performing integrity checks after each processing step to ensure no data was changed unintentionally. We realized after we started cleaning a lot of the columns that the data was irrelevant, so an earlier lesson would be to delete columns before even cleaning so that there would be no additional or duplicative work. Lastly, we learned that it was important to go back and to validate our data as part of integrity and correctness, one example was that when there were two less rows without the nearest neighborhood, only then did we find out that there were two missing rows that did not have longitude and latitude data.

If we were to complete this project again or have more time to further our data we would narrow down our scope more and focus on only certain crimes or even smaller areas. In addition we could try to add previous years statistics to see bigger picture trends. For example we could focus in on one neighborhood and one type of offense category such as larceny. In addition we could use different visualization tools such as Tableau to create advanced dashboards. This project allowed us to explore different ways to clean data to make it suitable for our use cases that we had not had much practice with before. We learned the value that data cleaning holds as it allowed for our graph creation and further analysis to be completed with ease.

Team Contributions

This project was a collaborative effort split between both members equally. Sim’s contributions were primarily focussed on granular data cleaning processes, including imputing missing values for neighborhoods. Albert performed the buik of the workflow modeling, summarizing changes carried out during the data cleaning exercise. Both team members discussed and collaborated on developing the data cleaning pipeline, visualizations, analysis and final report.