1.1 Exploratory Data Analysis

Exploratory data analysis was done prior to data cleaning. As a result, many of the figures were initially difficult to comprehend and required additional visualizations. The statistics and visualizations are saved in the plots directory and the eda.ipynb file.

1.2 Data Cleaning and Imputing Missing Values

Missing Value	**Why & how the missing values were filled**
Age	This column had over 200,000 rows of missing data in the dataset. As age is a critical factor to determine the outcome of COVID-19 patients, it was important to fill missing age values. The missing values were determined by creating a memo of all the average ages for each given province and then referencing the province of each missing age to impute the missing value. If the average age for a province was unavailable, the average age of a country was used, and if neither was available, the median of all the existing ages was used to fill missing values.
Sex	Similar to age, this column also had over 200,000 rows of missing values. In the training data set, the outcome column was used to determine the percentage of males and females who had 'recovered', 'hospitalized', etc. as their outcome. For each potential outcome, the sex that had a bigger percentage of receiving that outcome was used to fill in the missing values. Because the test dataset had no outcome column, a dictionary containing the larger gender majority percentage for each province or country generated from the training dataset was used to fill missing values.
Province	There were around over 4,000 missing values for the province which contained valuable information for determining if the location was a contributing factor to COVID-19 outcomes. Geopy was used to compute the missing provinces by inputting the given latitude/longitudes.
Country	The 18 missing values for the country column were filled to have a more detailed and complete dataset for milestone 2. Similar to the province, the country was determined using geopy by inputting the latitude and longitude of each missing data row to determine the missing country.
Date Confirmation	The 300 missing values for the date confirmation were filled in to have a full and complete dataset for milestone 2. After clearing the odd formats for the dates, the empty cells for date confirmation were filled by the most occurring date out of all the existing date confirmations.

1.3 Dealing with Outliers

There were two attribute groups that contained outliers.

The first attribute is age because some ages reached ranges way above the average life expectancy. These numerical inconsistencies were automatically detected using a normal distribution and setting z-index boundaries. The possible age ranges were plotted in the table ./plots/Age_plot.png which revealed a normal distribution in the data as displayed by the bell curve plotted in ./plots/Age_normal.png. A z-index range from -4 to 4 was determined to be the bounds such that any ages with z-indexes that resided outside of the bounds were determined to be outliers. For this dataset, ages that were greater than or equal to 100 were classified to be outliers.

Similarly, some latitude and longitude attributes were considered outliers because after filling in missing provinces from 1.2, some remaining missing values remained due to geopy unrecognizing the coordinates that point to undefined locations. Due to the improbable case that patients from an undefined region were tested for COVID-19, latitudes/longitudes that lead to lingering missing provinces were considered outliers and removed.

1.4 Transformation

US rows were filtered and grouped by 'Province_State'. The following aggregations were used:

Column	Aggregation Method
Last_Update	The mode of all rows belonging to the state. This would make the aggregated Last_Update value reflect the most frequently seen date.
Lat / Long_	The average of Lat and Long_ values was used. Averaging these would bring the aggregated coordinate to roughly the middle of all the individual locations.
Confirmed / Deaths / Recovered / Active	The sum aggregation was used for these columns. It makes sense to count all values amongst a state's various counties/areas to get a total for the state.
Combined_Key	The mode was used with the first value (i.e. the county/area) removed. This made the Combined_Key value only show '<state>, US'.
Incidence_Rate	The incidence rate represents cases per 100,000 people, so the value is already normalized to have a consistent base (100,000). Therefore, the values of the individual counties/areas were averaged to get a representative rate for the state.
Case-Fatality_Ratio	Since this ratio is the number of deaths / number of cases, a weighted average of the state's counties' ratios was used. Taking Alabama for example, its Case-Fatality_Ratio = Σ[(Case-Fatality_Ratio for county) * (Confirmed # for county) / (Confirmed # for Alabama)] for all of Alabama's counties. Using the weighted sum ensured that the ratio was correctly weighted by the number of confirmed cases per county.

1.5 Joining the Cases and Location Dataset

The 'province, country' key was used to join the cases datasets to the location dataset. This strategy was used because it was easy to implement and check matching values after imputing the missing provinces/states in part 1.2.

There were notably some mismatches between the datasets; for example, Puerto Rico was noted as a country in the cases dataset but as a province/state in the locations dataset. An exploratory pass of the datasets was done to find these mismatches and helper functions were created to align them and create a Combined_Key for the cases datasets.

The output (*cases_train_processed.csv* and *cases_test_processed.csv*) can be found in the results directory. Note there were some rows (~8%) with NaN values for countries such as Peru and Russia because they had no matching 'Province_States' in the location dataset. The values for these rows were filled with the mean values of the aggregation of the countries in location.csv.

1.6 Outcome Labels

The different outcome labels in the cases dataset are [deceased, hospitalized, nonhospitalized, recovered]. They represent the COVID-19 statuses of the individuals whose data is recorded in the dataset (e.g. "deceased" means the person had COVID-19 and died). Using such data to predict the outcome label falls under the data mining task of classification.