CS520-Team project

3#1 Chengzhe Li A20479782
3#2 Feng Zhang A20478597
3#3 Dingyi Zhao A20501339

Introduce the project dataset.

Data set source address:https://data.cityofchicago.org/Health-Human-Services/COVID-19-Vaccination-Coverage-ZIP-Code/2ani-ic5x

The name of the dataset is "COVID-19 Vaccination Coverage, ZIP Code." This dataset compiles the number of individuals vaccinated against COVID-19 and the percentage of the population vaccinated in various age groups in Illinois.

Column Name	Description	Type
ZIP Code	Home ZIP Code of the person vaccinated.	Plain Text
Week End	End date of the week. Dates are based on standard MMWR weeks as defined by the U.S. Centers for Disease Control and Prevention.	Date & Time
Season	Annually recurring reporting period for which estimates are calculated.	Plain Text
Measure	Vaccination status	Plain Text
Age Group	Values used are: 0-17, 18-64, 65+, 18+, Unknown, All Ages	Plain Text
Population Size	Number of people in the group based on the specified combination of ZIP Code and age.	Number
Count	Number of people in the specified group that achieved the vaccination status.	Number
Percent	Percent of people in the specified group that achieved the vaccination status.	Number
ZIP Code Centroid	A point that falls inside the ZIP Code, usually the centroid but adjusted if the centroid would fall outside the Zip. It is present to facilitate geographic analysis.	Point

Data Quality

Subsequent steps will primarily address formatting and missing data issues.

- There are redundant fields
- There are issues with field formats
- There are missing fields
- There are no duplicate rows
- The data range is normal

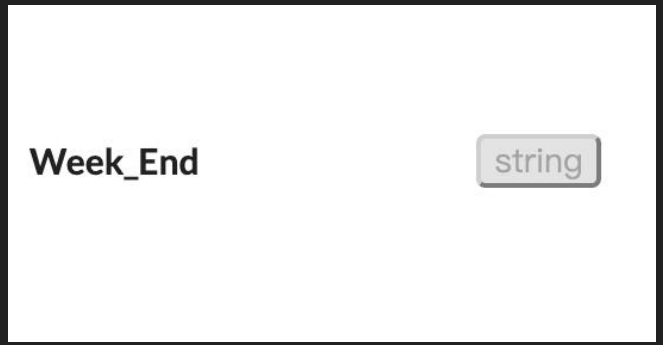
Redundant fields

The original dataset downloaded contains five additional fields compared to what is described in the dataset introduction, and there is no further information available to interpret or explain the function of these fields. Therefore, these five redundant fields should be deleted.

```
#The data source field description and related descriptions provide a slight explanation of the meaning of the last five f
#And rename the deleted dataset

import pandas as pd

# Load the dataset
file_path = '/vizier.db/COVID-19_Vaccination_Coverage__ZIP_Code_20231120.csv'
data = pd.read_csv(file_path)

# Columns to be removed
columns_to_remove = ['Boundaries_ZIP_Codes', 'Community_Areas', 'Zip_Codes', 'Census_Tracts', 'Wards_2023_']

# Remove the specified columns
data_cleaned = data.drop(columns=columns_to_remove, errors='ignore')

# Save the cleaned dataset to a new CSV file with the original file name
cleaned_file_path = '/vizier.db/COVID-19_Vaccination_Coverage__ZIP_Code_1.csv'
data_cleaned.to_csv(cleaned_file_path, index=False)

cleaned_file_path
```

Field format issues

The 'Week End' field is described as a date format on the page, but the downloaded data shows the field in String format.



Missing fields

Population Size: 90 missing entries. Percent: 90 missing entries. ZIP Code Centroid: 45 missing entries.

```
SELECT
    (COUNT(*) - COUNT(`ZIP_Code`)) AS missing_zip_code,
    (COUNT(*) - COUNT(`Week_End`)) AS missing_week_end,
    (COUNT(*) - COUNT(`Season`)) AS missing_season,
    (COUNT(*) - COUNT(`Measure`)) AS missing_measure,
    (COUNT(*) - COUNT(`Age_Group`)) AS missing_age_group,
    (COUNT(*) - COUNT(`Population_Size`)) AS missing_populatior
    (COUNT(*) - COUNT(`Count`)) AS missing_count,
    (COUNT(*) - COUNT(`Percent`)) AS missing_percent,
    (COUNT(*) - COUNT(`ZIP_Code_Centroid`)) AS missing_zip_code
FROM
    `covid19`;
```

	missing_zip_code (long)	missing_week_end (long)	missing_season (long)
0	45	0	0

missing_measure (long)	missing_age_group (long)	missing_population_size (long)
0	0	90

missing_count (long)	missing_percent (long)	missing_zip_code_centroid (long)
0	90	45

Check for duplicates

There are no duplicate rows in the data.

```
SELECT
    `ZIP_Code`,
    `Week_End`,
    `Season`,
    `Measure`,
    `Age_Group`,
    `Population_Size`,
    `Count`,
    `Percent`,
    `ZIP_Code_Centroid`,
    COUNT(*) AS duplicate_count
FROM
    `covid19`
GROUP BY
    `ZIP_Code`,
    `Week_End`,
    `Season`,
    `Measure`,
    `Age_Group`,
    `Population_Size`,
    `Count`,
    `Percent`,
    `ZIP_Code_Centroid`
HAVING
    COUNT(*) > 1;
```

Check data range

No out-of-range data for percentage and population fields.

```
SELECT
    *
FROM
    `covid19`
WHERE
    `Percent` < 0 OR `Percent` > 1;
```

Console | Timing | Datasets | Charts

temporary_dataset (0 rows)

Views

ZIP_Code (int)	Week_End (string)	Season (string)	Measure (string)	Age_Group (string)	Population_Size (float)	Count (short)	Percent (float)	ZIP_Code_Centroid (string)

```
SELECT
    *
FROM
    `covid19`
WHERE
    `Population_Size` < 0;
```

Console | Timing | Datasets | Charts

temporary_dataset (0 rows)

Views

ZIP_Code (int)	Week_End (string)	Season (string)	Measure (string)	Age_Group (string)	Population_Size (float)	Count (short)	Percent (float)	ZIP_Code_Centroid (string)

Resolving data quality issues

Week End field type error

Correct field types

```
import pandas as pd

# Read the CSV file
df = pd.read_csv('/vizier.db/COVID-19_Vaccination_Coverage__ZIP_Code_1.csv')

# Since the date format is MM/DD/YYYY, we need to change the format parameter accordingly
df['Week End'] = pd.to_datetime(df['Week End'], format='%m/%d/%Y')

# If the conversion is successful, save the transformed DataFrame back to the CSV file
df.to_csv('/vizier.db/COVID-19_Vaccination_Coverage__ZIP_Code_1.csv', index=False)

# Print DataFrame information to confirm the conversion
df.info()
```

Verify the results of the correction

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2700 entries, 0 to 2699
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ZIP Code           2700 non-null   object
 1   Week End           2700 non-null   datetime64[ns]
 2   Season             2700 non-null   object
 3   Measure            2700 non-null   object
 4   Age Group          2700 non-null   object
 5   Population Size    2610 non-null   float64
 6   Count              2700 non-null   int64
 7   Percent            2610 non-null   float64
 8   ZIP Code Centroid  2655 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 190.0+ KB
```

Solving data missing issues

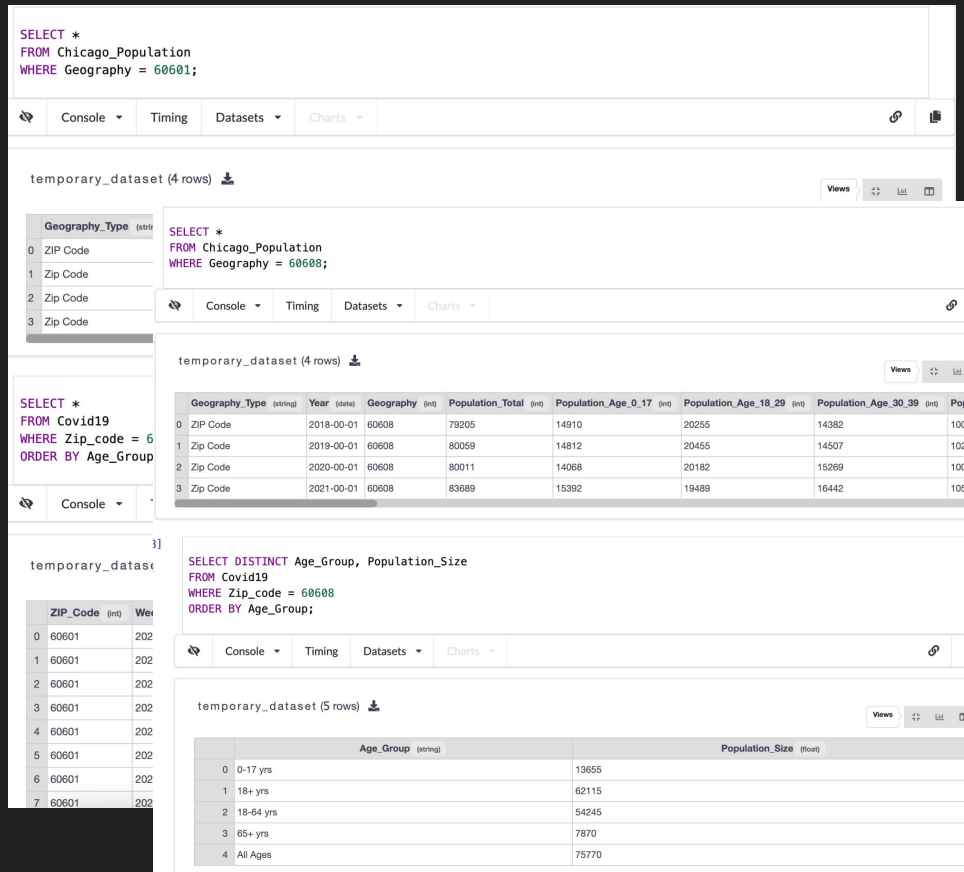
Find the appropriate dataset

1. Chicago Population Counts https://data.cityofchicago.org/Health-Human-Services/Chicago-Population-Counts/85cm-7uqa
2. Data Sources: U.S. Census Bureau American Community Survey (ACS) 5-year estimates (ZIP Code) and 1-year estimates (Citywide). The U.S. Census Bureau did not release standard 1-year estimates from the 2020 ACS. In 2020 only, 5-year estimates were used for the Citywide estimates.

Column Name	Description	Type
Geography Type	The type of geographic unit for the population count.	Plain Text T
Year	The year to which the population count applies.	Number #
Geography	The specific geographic unit.	Plain Text T
Population - Total		Number #
Population - Age 0-17		Number #
Population - Age 18-29		Number #
Population - Age 30-39		Number #
Population - Age 40-49		Number #
Population - Age 50-59		Number #
Population - Age 60-69		Number #
Population - Age 70-79		Number #
Population - Age 80+		Number #
Population - Age 0-4	Note that this column is provided as a convenience and overl...	Number #
Population - Age 5-11	Note that this column is provided as a convenience and overl...	Number #
Population - Age 12-17	Note that this column is provided as a convenience and overl...	Number #
Population - Age 5+	Note that this column is provided as a convenience and overl...	Number #
Population - Age 18+	Note that this column is provided as a convenience and overl...	Number #
Population - Age 65+	Note that this column is provided as a convenience and overl...	Number #
Population - Female		Number #
Population - Male		Number #
Population - Latinx		Number #
Population - Asian Non-Latinx		Number #
Population - Black Non-Latinx		Number #
Population - White Non-Latinx		Number #
Population - Other Race Non-Latinx		Number #
Record ID	A unique ID for the record.	Plain Text T

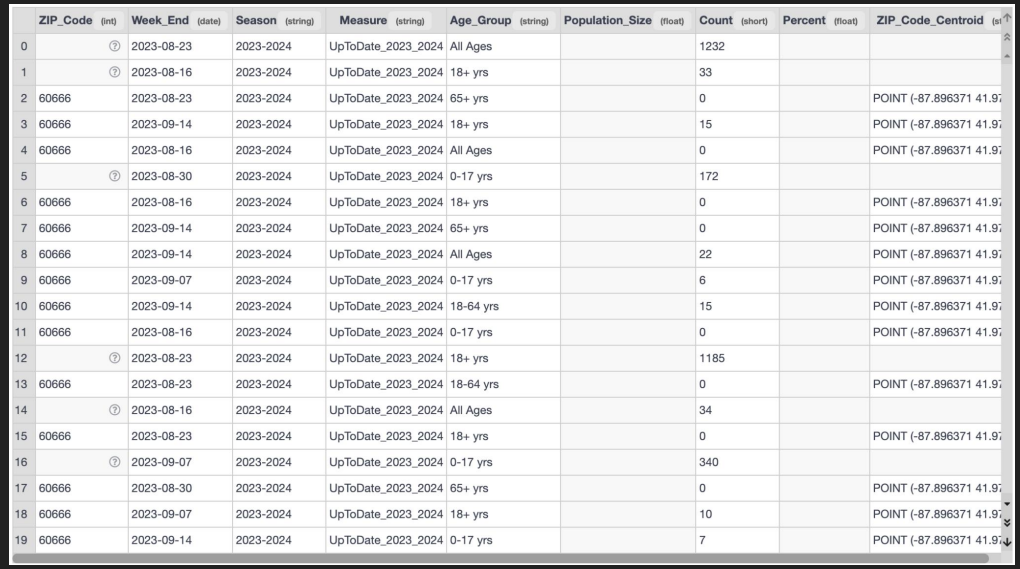
Verify the compatibility of the dataset

1. Randomly comparing population data under different postal codes
2. Confirm the vacancy data situation of the current target data set. There are two types of vacancy data situations: one is for the region with a postal code of 60666, with missing population data, and the other is for the unknown region, with missing data confirmation
3. Confirm using this dataset to supplement the target dataset data based on the data situation



Verify the compatibility of the dataset

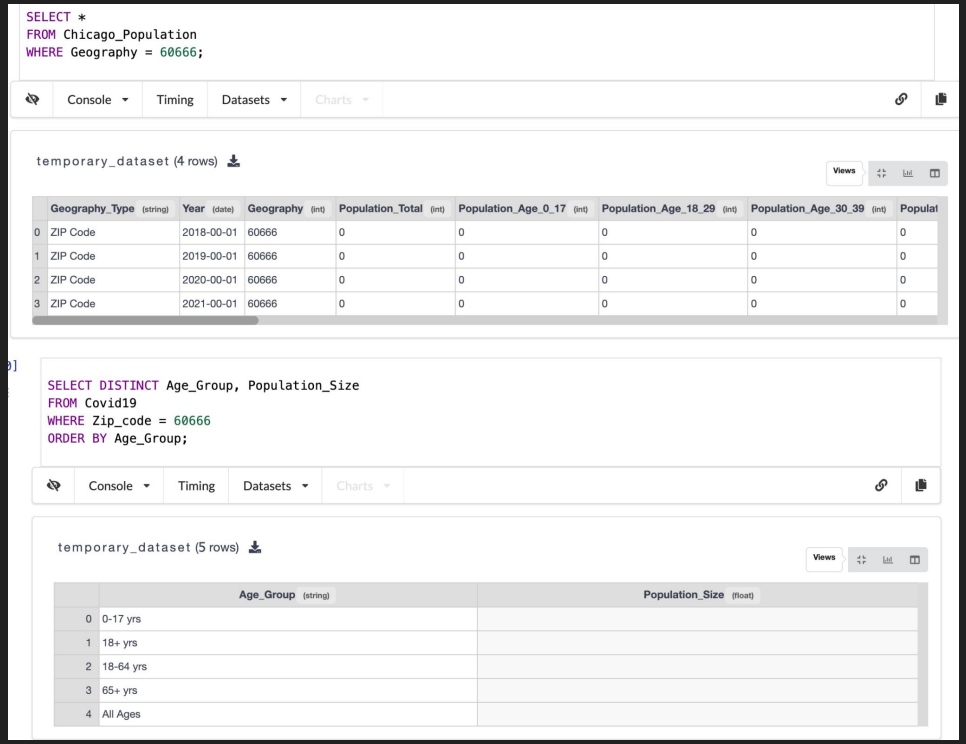
1. Randomly comparing population data under different postal codes
2. Confirm the vacancy data situation of the current target data set. There are two types of vacancy data situations: one is for the region with a postal code of 60666, with missing population data, and the other is for the unknown region, with missing data confirmation
3. Confirm using this dataset to supplement the target dataset data based on the data situation



	ZIP_Code (int)	Week_End (date)	Season (string)	Measure (string)	Age_Group (string)	Population_Size (float)	Count (short)	Percent (float)	ZIP_Code_Centroid (st
0		2023-08-23	2023-2024	UpToDate_2023_2024	All Ages		1232		
1		2023-08-16	2023-2024	UpToDate_2023_2024	18+ yrs		33		
2	60666	2023-08-23	2023-2024	UpToDate_2023_2024	65+ yrs		0		POINT (-87.896371 41.97
3	60666	2023-09-14	2023-2024	UpToDate_2023_2024	18+ yrs		15		POINT (-87.896371 41.97
4	60666	2023-08-16	2023-2024	UpToDate_2023_2024	All Ages		0		POINT (-87.896371 41.97
5		2023-08-30	2023-2024	UpToDate_2023_2024	0-17 yrs		172		
6	60666	2023-08-16	2023-2024	UpToDate_2023_2024	18+ yrs		0		POINT (-87.896371 41.97
7	60666	2023-09-14	2023-2024	UpToDate_2023_2024	65+ yrs		0		POINT (-87.896371 41.97
8	60666	2023-09-14	2023-2024	UpToDate_2023_2024	All Ages		22		POINT (-87.896371 41.97
9	60666	2023-09-07	2023-2024	UpToDate_2023_2024	0-17 yrs		6		POINT (-87.896371 41.97
10	60666	2023-09-14	2023-2024	UpToDate_2023_2024	18-64 yrs		15		POINT (-87.896371 41.97
11	60666	2023-08-16	2023-2024	UpToDate_2023_2024	0-17 yrs		0		POINT (-87.896371 41.97
12		2023-08-23	2023-2024	UpToDate_2023_2024	18+ yrs		1185		
13	60666	2023-08-23	2023-2024	UpToDate_2023_2024	18-64 yrs		0		POINT (-87.896371 41.97
14		2023-08-16	2023-2024	UpToDate_2023_2024	All Ages		34		
15	60666	2023-08-23	2023-2024	UpToDate_2023_2024	18+ yrs		0		POINT (-87.896371 41.97
16		2023-09-07	2023-2024	UpToDate_2023_2024	0-17 yrs		340		
17	60666	2023-08-30	2023-2024	UpToDate_2023_2024	65+ yrs		0		POINT (-87.896371 41.97
18	60666	2023-09-07	2023-2024	UpToDate_2023_2024	18+ yrs		10		POINT (-87.896371 41.97
19	60666	2023-09-14	2023-2024	UpToDate_2023_2024	0-17 yrs		7		POINT (-87.896371 41.97

Verify the compatibility of the dataset

1. Randomly comparing population data under different postal codes
2. Confirm the vacancy data situation of the current target data set. There are two types of vacancy data situations: one is for the region with a postal code of 60666, with missing population data, and the other is for the unknown region, with missing data confirmation
3. Confirm using this dataset to supplement the target dataset data based on the data situation



```
SELECT *
FROM Chicago_Population
WHERE Geography = 60666;
```

Console | Timing | Datasets | Charts

temporary_dataset (4 rows)

Views

	Geography_Type (string)	Year (date)	Geography (int)	Population_Total (int)	Population_Age_0_17 (int)	Population_Age_18_29 (int)	Population_Age_30_39 (int)	Populat
0	ZIP Code	2018-00-01	60666	0	0	0	0	0
1	ZIP Code	2019-00-01	60666	0	0	0	0	0
2	ZIP Code	2020-00-01	60666	0	0	0	0	0
3	ZIP Code	2021-00-01	60666	0	0	0	0	0

```
SELECT DISTINCT Age_Group, Population_Size
FROM Covid19
WHERE Zip_code = 60666
ORDER BY Age_Group;
```

Console | Timing | Datasets | Charts

temporary_dataset (5 rows)

Views

	Age_Group (string)	Population_Size (float)
0	0-17 yrs	
1	18+ yrs	
2	18-64 yrs	
3	65+ yrs	
4	All Ages	

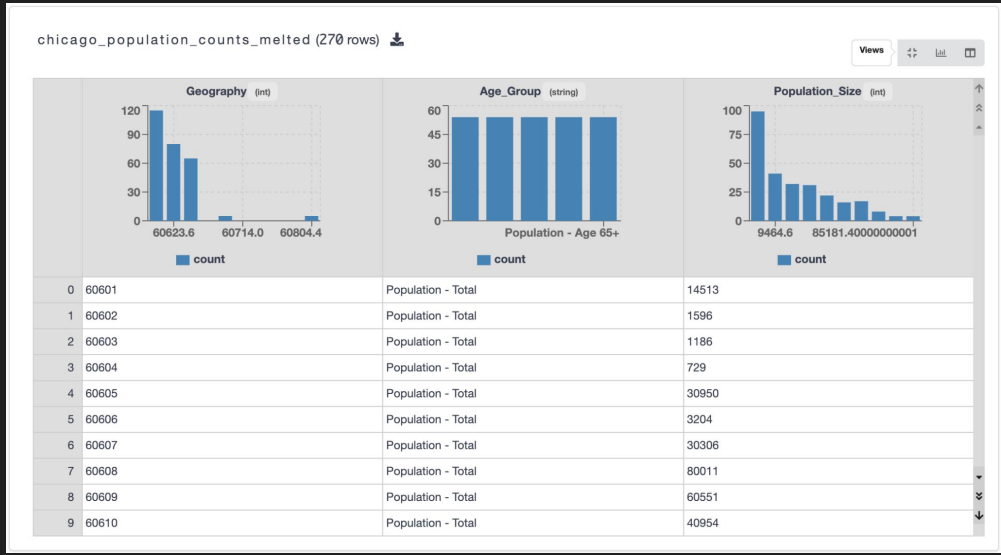
Dataset processing

1. Perform data row to column conversion
2. Obtain matching columns through calculation
3. Match data field connections with the target table
4. Make a left join to the target table, connect the supplementary dataset to the target table, and perform secondary confirmation
5. Add supplementary field data to the target table

```python
import pandas as pd

# Set Pandas display options
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 20)
pd.set_option('display.width', 1000)

# Load the CSV file
population_data_path = '/vizier.db/Chicago_Population_Counts_20231115.csv'
population_data = pd.read_csv(population_data_path)

# Filter for data where 'Geography Type' is either 'ZIP Code' or 'Zip Code'
population_data = population_data[(population_data['Geography Type'] == 'ZIP Code') | (population_data['Geography Type'] ==

# Define the target year
target_date = '2020'

# Filter for rows where 'Year' column contains '2020', and keep only necessary columns
population_data_filtered = population_data[population_data['Year'].astype(str).str.contains(target_date)]
population_data_filtered = population_data_filtered[['Geography', 'Population - Total', 'Population - Age 0-17', 'Populatio

# Calculate a new column 'Age18-64'
population_data_filtered['Age18-64'] = population_data_filtered['Population - Age 18+'] - population_data_filtered['Popula

# Pivot the Population related columns
melted_population_data = population_data_filtered.melt(id_vars=['Geography'],
                                                       value_vars=['Population - Total', 'Population - Age 0-17', 'Populat
                                                       var_name='Age_Group2',
                                                       value_name='Population_Size2')

# Save the final results to a new table (CSV file)
output_file_path = '/vizier.db/Chicago_Population_Counts_melted.csv'
melted_population_data.to_csv(output_file_path, index=False)

# Notify about the saving result
print(f"Data saved to: {output_file_path}")
```

Dataset processing

1. Perform data row to column conversion
2. Obtain matching columns through calculation
3. Match data field connections with the target table
4. Make a left join to the target table, connect the supplementary dataset to the target table, and perform secondary confirmation
5. Add supplementary field data to the target table



	Geography (int)	Age_Group (string)	Population_Size (int)
0	60601	Population - Total	14513
1	60602	Population - Total	1596
2	60603	Population - Total	1186
3	60604	Population - Total	729
4	60605	Population - Total	30950
5	60606	Population - Total	3204
6	60607	Population - Total	30306
7	60608	Population - Total	80011
8	60609	Population - Total	60551
9	60610	Population - Total	40954

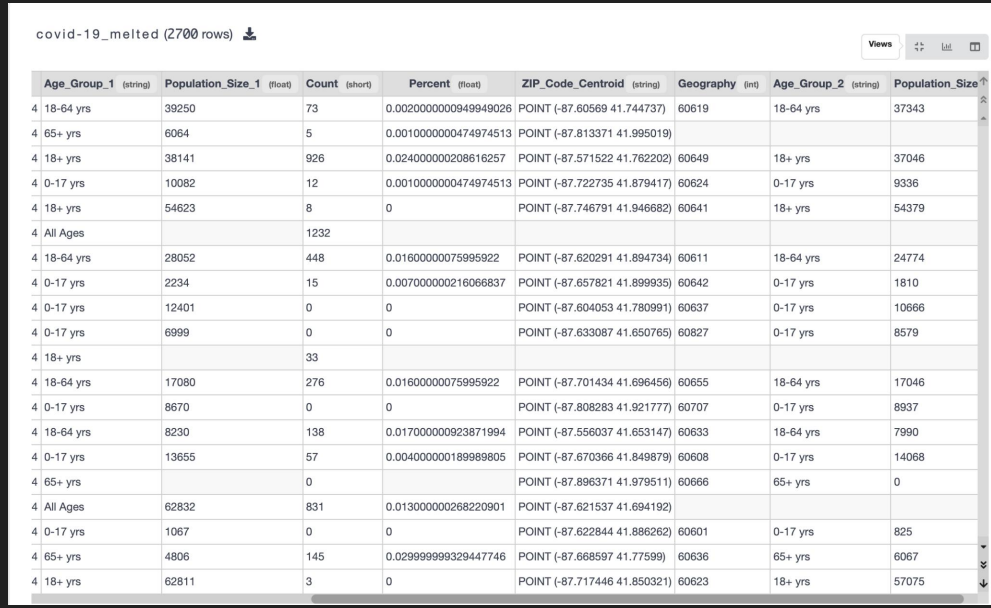
Dataset processing

1. Perform data row to column conversion
2. Obtain matching columns through calculation
3. Match data field connections with the target table
4. Make a left join to the target table, connect the supplementary dataset to the target table, and perform secondary confirmation
5. Add supplementary field data to the target table

```python
import pandas as pd

# Load two CSV files
covid_data_path = '/vizier.db/COVID-19_Vaccination_Coverage__ZIP_Code_1.csv'
population_data_path = '/vizier.db/Chicago_Population_Counts_melted.csv'

covid_data = pd.read_csv(covid_data_path)
population_data = pd.read_csv(population_data_path)

# Ensure that the merging key columns are of string type
covid_data['ZIP Code'] = covid_data['ZIP Code'].astype(str)
covid_data['Age Group'] = covid_data['Age Group'].astype(str)
population_data['Geography'] = population_data['Geography'].astype(str)
population_data['Age_Group2'] = population_data['Age_Group2'].astype(str)

# Create a mapping dictionary
age_group_mapping = {
    'Age18-64': '18-64 yrs',
    'Population - Age 65+': '65+ yrs',
    'Population - Age 18+': '18+ yrs',
    'Population - Age 0-17': '0-17 yrs',
    'Population - Total': 'All Ages'
}

# Apply the mapping dictionary to 'Age_Group2'
population_data['Age_Group2'] = population_data['Age_Group2'].map(age_group_mapping)

# Merge data using a left join
merged_data = pd.merge(covid_data, population_data,
                       how='left',
                       left_on=['ZIP Code', 'Age Group'],
                       right_on=['Geography', 'Age_Group2'])

# Save the merged data
output_file_path = '/vizier.db/COVID-19_melted.csv'
merged_data.to_csv(output_file_path, index=False)

print("Merged data saved to:", output_file_path)
```

Dataset processing

1. Perform data row to column conversion
2. Obtain matching columns through calculation
3. Match data field connections with the target table
4. Make a left join to the target table, connect the supplementary dataset to the target table, and perform secondary confirmation
5. Add supplementary field data to the target table



covid-19_melted (2700 rows)

Views

	Age_Group_1 (string)	Population_Size_1 (float)	Count (short)	Percent (float)	ZIP_Code_Centroid (string)	Geography (int)	Age_Group_2 (string)	Population_Size
4	18-64 yrs	39250	73	0.0020000000949949026	POINT (-87.60569 41.744737)	60619	18-64 yrs	37343
4	65+ yrs	6064	5	0.0010000000474974513	POINT (-87.813371 41.995019)			
4	18+ yrs	38141	926	0.024000000208616257	POINT (-87.571522 41.762202)	60649	18+ yrs	37046
4	0-17 yrs	10082	12	0.0010000000474974513	POINT (-87.722735 41.879417)	60624	0-17 yrs	9336
4	18+ yrs	54623	8	0	POINT (-87.746791 41.946682)	60641	18+ yrs	54379
4	All Ages		1232					
4	18-64 yrs	28052	448	0.01600000075995922	POINT (-87.620291 41.894734)	60611	18-64 yrs	24774
4	0-17 yrs	2234	15	0.007000000216066837	POINT (-87.657821 41.899935)	60642	0-17 yrs	1810
4	0-17 yrs	12401	0	0	POINT (-87.604053 41.780991)	60637	0-17 yrs	10666
4	0-17 yrs	6999	0	0	POINT (-87.633087 41.650765)	60827	0-17 yrs	8579
4	18+ yrs		33					
4	18-64 yrs	17080	276	0.01600000075995922	POINT (-87.701434 41.696456)	60655	18-64 yrs	17046
4	0-17 yrs	8670	0	0	POINT (-87.808283 41.921777)	60707	0-17 yrs	8937
4	18-64 yrs	8230	138	0.017000000923871994	POINT (-87.556037 41.653147)	60633	18-64 yrs	7990
4	0-17 yrs	13655	57	0.004000000189989805	POINT (-87.670366 41.849879)	60608	0-17 yrs	14068
4	65+ yrs		0		POINT (-87.896371 41.979511)	60666	65+ yrs	0
4	All Ages	62832	831	0.013000000268220901	POINT (-87.621537 41.694192)			
4	0-17 yrs	1067	0	0	POINT (-87.622844 41.886262)	60601	0-17 yrs	825
4	65+ yrs	4806	145	0.029999999329447746	POINT (-87.668597 41.77599)	60636	65+ yrs	6067
4	18+ yrs	62811	3	0	POINT (-87.717446 41.850321)	60623	18+ yrs	57075

Dataset processing

1. Perform data row to column conversion
2. Obtain matching columns through calculation
3. Match data field connections with the target table
4. Make a left join to the target table, connect the supplementary dataset to the target table, and perform secondary confirmation
5. Add supplementary field data to the target table

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('/vizier.db/COVID-19_melted.csv')  # Load the dataset from the specified location

# Fill missing values in 'Population Size' with values from 'Population_Size2'
df['Population Size'].fillna(df['Population_Size2'], inplace=True)

# Drop the columns 'Geography', 'Age_Group2', and 'Population_Size2'
df.drop(columns=['Geography', 'Age_Group2', 'Population_Size2'], inplace=True)

# Print the first few rows of the modified dataframe
print("Modified DataFrame (first few rows):")
print(df.head())

# Save the modified dataframe to a new file
output_file_path = '/vizier.db/COVID19_final.csv'
df.to_csv(output_file_path, index=False)

print("\nModified data saved to:", output_file_path)
```

Console ▾ | Timing | Datasets ▾ | Charts ▾

```
Modified DataFrame (first few rows):
ZIP Code    Week End     Season  ... Count Percent            ZIP Code Centroid
0    60619  2023-09-23  2023-2024  ...    73   0.002   POINT (-87.60569 41.744737)
1    60631  2023-09-16  2023-2024  ...     5   0.001  POINT (-87.813371 41.995019)
2    60649  2023-10-14  2023-2024  ...   926   0.024  POINT (-87.571522 41.762202)
3    60624  2023-10-14  2023-2024  ...    12   0.001  POINT (-87.722735 41.879417)
4    60641  2023-09-16  2023-2024  ...     8   0.000  POINT (-87.746791 41.946682)

[5 rows x 9 columns]

Modified data saved to:
/vizier.db/COVID19_final.csv
```

Result after processing

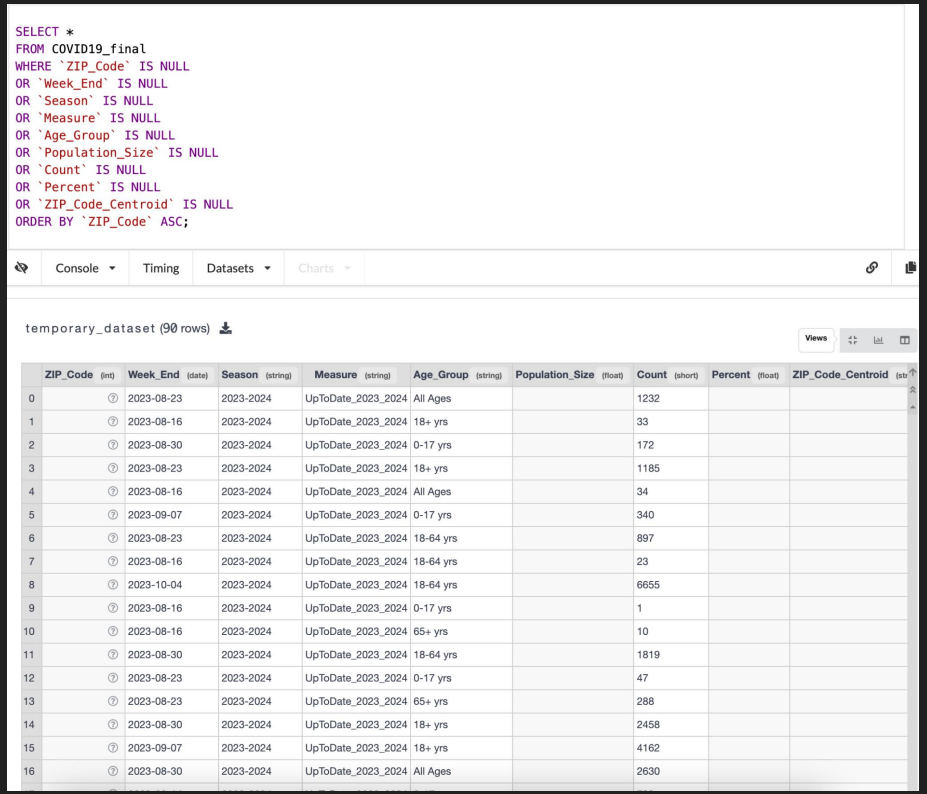
Summarize the status after processing

There are 90 rows with missing values left.

Of these, 45 rows have missing values in the 'Percent' field for the area with ZipCode 60666.

This is because the permanent population of this area is 0 according to census data, making it impossible to calculate percentages with 0 as the denominator.

The other 45 rows contain vaccination data for people from unknown areas, for which no supplemental information could be found. Therefore, these 45 rows of data will be deleted.



```
SELECT *
FROM COVID19_final
WHERE `ZIP_Code` IS NULL
OR `Week_End` IS NULL
OR `Season` IS NULL
OR `Measure` IS NULL
OR `Age_Group` IS NULL
OR `Population_Size` IS NULL
OR `Count` IS NULL
OR `Percent` IS NULL
OR `ZIP_Code_Centroid` IS NULL
ORDER BY `ZIP_Code` ASC;
```

Console | Timing | Datasets | Charts

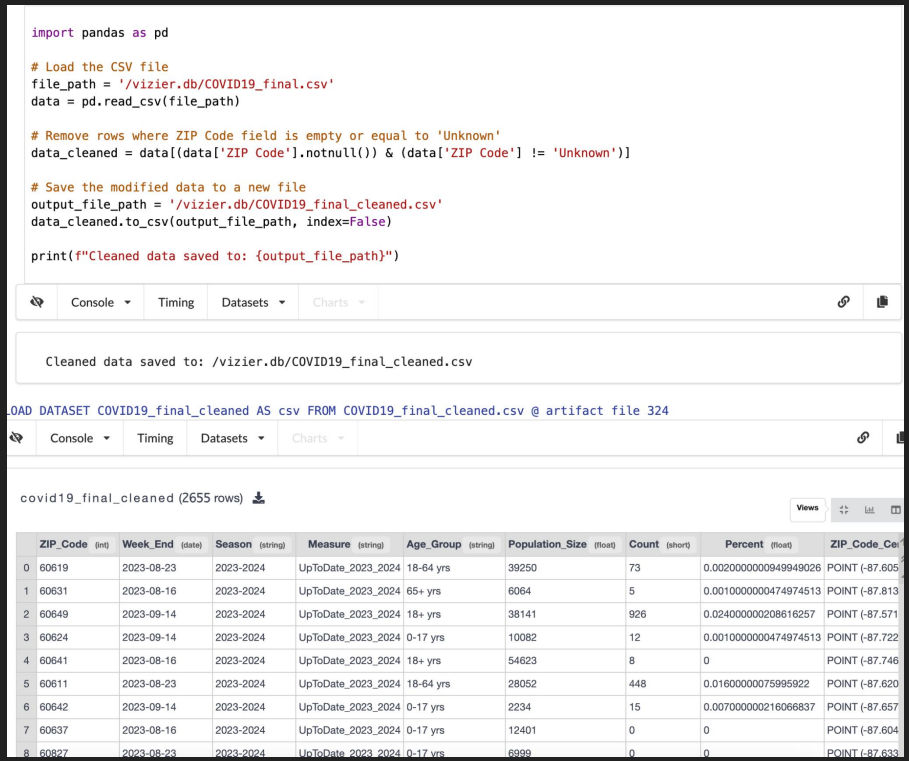
temporary_dataset (90 rows)

Views

	ZIP_Code (int)	Week_End (date)	Season (string)	Measure (string)	Age_Group (string)	Population_Size (float)	Count (short)	Percent (float)	ZIP_Code_Centroid (str
0		2023-08-23	2023-2024	UpToDate_2023_2024	All Ages		1232		
1		2023-08-16	2023-2024	UpToDate_2023_2024	18+ yrs		33		
2		2023-08-30	2023-2024	UpToDate_2023_2024	0-17 yrs		172		
3		2023-08-23	2023-2024	UpToDate_2023_2024	18+ yrs		1185		
4		2023-08-16	2023-2024	UpToDate_2023_2024	All Ages		34		
5		2023-09-07	2023-2024	UpToDate_2023_2024	0-17 yrs		340		
6		2023-08-23	2023-2024	UpToDate_2023_2024	18-64 yrs		897		
7		2023-08-16	2023-2024	UpToDate_2023_2024	18-64 yrs		23		
8		2023-10-04	2023-2024	UpToDate_2023_2024	18-64 yrs		6655		
9		2023-08-16	2023-2024	UpToDate_2023_2024	0-17 yrs		1		
10		2023-08-16	2023-2024	UpToDate_2023_2024	65+ yrs		10		
11		2023-08-30	2023-2024	UpToDate_2023_2024	18-64 yrs		1819		
12		2023-08-23	2023-2024	UpToDate_2023_2024	0-17 yrs		47		
13		2023-08-23	2023-2024	UpToDate_2023_2024	65+ yrs		288		
14		2023-08-30	2023-2024	UpToDate_2023_2024	18+ yrs		2458		
15		2023-09-07	2023-2024	UpToDate_2023_2024	18+ yrs		4162		
16		2023-08-30	2023-2024	UpToDate_2023_2024	All Ages		2630		

Delete 45 unknown data items

The other 45 rows contain vaccination data for people from unknown areas, for which no supplemental information could be found. Therefore, these 45 rows of data will be deleted.



```
import pandas as pd

# Load the CSV file
file_path = '/vizier.db/COVID19_final.csv'
data = pd.read_csv(file_path)

# Remove rows where ZIP Code field is empty or equal to 'Unknown'
data_cleaned = data[(data['ZIP Code'].notnull()) & (data['ZIP Code'] != 'Unknown')]

# Save the modified data to a new file
output_file_path = '/vizier.db/COVID19_final_cleaned.csv'
data_cleaned.to_csv(output_file_path, index=False)

print(f"Cleaned data saved to: {output_file_path}")
```

Console | Timing | Datasets | Charts

```
Cleaned data saved to: /vizier.db/COVID19_final_cleaned.csv
```

```
LOAD DATASET COVID19_final_cleaned AS csv FROM COVID19_final_cleaned.csv @ artifact file 324
```

Console | Timing | Datasets | Charts

covid19_final_cleaned (2655 rows)

Views

	ZIP_Code (int)	Week_End (date)	Season (string)	Measure (string)	Age_Group (string)	Population_Size (float)	Count (short)	Percent (float)	ZIP_Code_Ce
0	60619	2023-08-23	2023-2024	UpToDate_2023_2024	18-64 yrs	39250	73	0.0020000000949949026	POINT (-87.605
1	60631	2023-08-16	2023-2024	UpToDate_2023_2024	65+ yrs	6064	5	0.0010000000474974513	POINT (-87.813
2	60649	2023-09-14	2023-2024	UpToDate_2023_2024	18+ yrs	38141	926	0.024000000208616257	POINT (-87.571
3	60624	2023-09-14	2023-2024	UpToDate_2023_2024	0-17 yrs	10082	12	0.0010000000474974513	POINT (-87.722
4	60641	2023-08-16	2023-2024	UpToDate_2023_2024	18+ yrs	54623	8	0	POINT (-87.746
5	60611	2023-08-23	2023-2024	UpToDate_2023_2024	18-64 yrs	28052	448	0.01600000075995922	POINT (-87.620
6	60642	2023-09-14	2023-2024	UpToDate_2023_2024	0-17 yrs	2234	15	0.007000000216066837	POINT (-87.657
7	60637	2023-08-16	2023-2024	UpToDate_2023_2024	0-17 yrs	12401	0	0	POINT (-87.604
8	60827	2023-08-23	2023-2024	UpToDate_2023_2024	0-17 yrs	6999	0	0	POINT (-87.633

Relayout table

- Use SQL to query the population of different categories (0-17 yrs, 18+ yrs, 65+ yrs, All Ages)

[42]

```
Select
y.ZIP_Code,
to_date('2023-01-01','yyyy-MM-dd') as Year,
a.Population_Size as Population_Total,
y.Population_Size as Population_Age_0_17,
g.Population_Size as Population_Age_18_,
o.Population_Size as Population_Age_65_

From COVID19_final_0_17_yrs_Population  y
join COVID19_final_18_Population g on y.ZIP_Code = g.ZIP_Code
join COVID19_final_65_Population o on y.ZIP_Code =o.ZIP_Code
join COVID19_final_All_Ages_Population a on y.ZIP_Code =a.ZIP_Code
--COVID19_final_All_Population
```

Console | Timing | Datasets | Charts

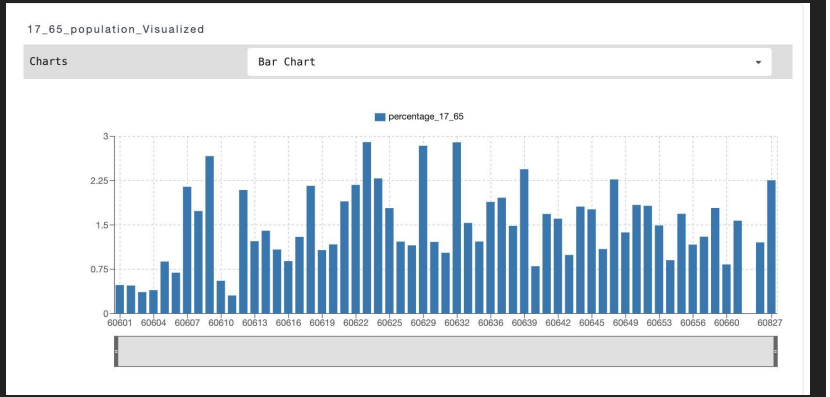
covid19_final_all_population (59 rows)

Views

	ZIP_Code (int)	Year (date)	Population_Total (float)	Population_Age_0_17 (float)	Population_Age_18_ (float)	Population_Age_65_ (float)
0	60637	2023-00-01	53555	12401	41154	6324
1	60626	2023-00-01	50548	7174	43374	5889
2	60647	2023-00-01	85631	14291	71340	6295
3	60624	2023-00-01	37547	10082	27465	4407
4	60633	2023-00-01	13359	3105	10254	2024
5	60625	2023-00-01	76525	14903	61622	8349
6	60618	2023-00-01	90316	18465	71851	8540
7	60602	2023-00-01	1261	111	1150	234
8	60607	2023-00-01	31816	3718	28098	1733
9	60652	2023-00-01	40898	9685	31213	5311
10	60646	2023-00-01	28383	6286	22097	5754
11	60634	2023-00-01	75694	15258	60436	12512
12	60631	2023-00-01	29691	6242	23449	6064
13	60640	2023-00-01	65941	7261	58680	9067
14	60608	2023-00-01	75770	13655	62115	7870
15	60639	2023-00-01	89543	22969	66574	9403
16	60614	2023-00-01	73173	10082	63091	7191
17	60616	2023-00-01	54013	8049	45964	9067
18	60666	2023-00-01	0	0	0	0
19	60611	2023-00-01	38456	2438	36018	7966

Ratio of young population to old population

- 0-17 yrs as a young age.
- 65+ yrs as a old age.
- The higher the percentage, the more young and energetic the community is.



[47]

```
Select o.ZIP_Code, y.Population_Size/o.Population_Size as percentage_17_65,
y.Population_Size as 17_Population_Size,
o.Population_Size as 65_Population_Size
from COVID19_final_0_17_yrs_Population  y
join COVID19_final_65_yrs_Population o on y.ZIP_Code=o.ZIP_Code
--17_65_population
```

Console | Timing | Datasets | Charts

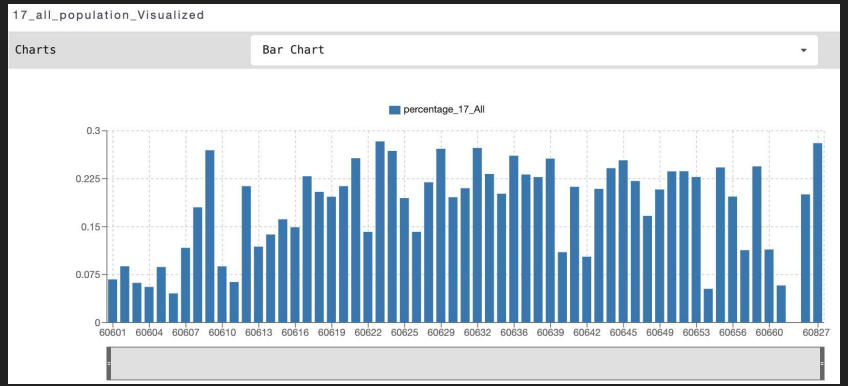
17_65_population (59 rows)

Views

	ZIP_Code (int)	percentage_17_65 (real)	17_Population_Size (float)	65_Population_Size (float)
0	60637	1.9609424414927261	12401	6324
1	60626	1.21820343012396	7174	5889
2	60647	2.2702144559173947	14291	6295
3	60624	2.287724075334695	10082	4407
4	60633	1.5340909090909092	3105	2024
5	60625	1.7850041921188167	14903	8349
6	60618	2.1621779859484778	18465	8540
7	60602	0.47435897435897434	111	234
8	60607	2.145412579342181	3718	1733
9	60652	1.8235737149312747	9685	5311
10	60646	1.0924574209245743	6286	5754
11	60634	1.2194693094629157	15258	12512
12	60631	1.029353562005277	6242	6064
13	60640	0.8008161464652035	7261	9067
14	60608	1.7350698856416773	13655	7870
15	60639	2.4427310432840583	22969	9403
16	60614	1.4020303156723681	10082	7191
17	60616	0.8877247160030881	8049	9067
18	60666		0	0
19	60611	0.30605071554104946	2438	7966

Ratio of young population to All population

- Proportion of young people in total community population.
- The higher the percentage, the more young and energetic the community is.



[48]

```
Select o.ZIP_Code, y.Population_Size/o.Population_Size as percentage_17_All,
y.Population_Size as 17_Population_Size,
o.Population_Size as All_Population_Size
from COVID19_final_0_17_yrs_Population  y
join COVID19_final_All_Ages_Population o on y.ZIP_Code=o.ZIP_Code
```

Console | Timing | Datasets | Charts

17_all_population (59 rows)

Views

	ZIP_Code (int)	percentage_17_All (real)	17_Population_Size (float)	All_Population_Size (float)
0	60637	0.2315563439454766	12401	53555
1	60626	0.14192450739890797	7174	50548
2	60647	0.1668904952645654	14291	85631
3	60624	0.2685167922870003	10082	37547
4	60633	0.23242757691443972	3105	13359
5	60625	0.1947468147664162	14903	76525
6	60618	0.2044488241286151	18465	90316
7	60602	0.088025376685170 5	111	1261
8	60607	0.11685944179029419	3718	31816
9	60652	0.236808645899555	9685	40898
10	60646	0.22147059859775217	6286	28383
11	60634	0.20157476153988427	15258	75694
12	60631	0.210232056852244 78	6242	29691
13	60640	0.11011358638783154	7261	65941
14	60608	0.18021644450310148	13655	75770
15	60639	0.2565136303228616	22969	89543
16	60614	0.1377830620584095	10082	73173
17	60616	0.1490196804472997	8049	54013
18	60666		0	0
19	60611	0.06339712918660287	2438	38456

Calculate the average population

- There are 4 years of data in Chicago_Population. Use this data to calculate the average population.

```
Select
Zip_Code,
avg(Population_Total) as Population_Total,
avg(Population_Age_0_17) as Population_Age_0_17,
avg(Population_Age_18_) as Population_Age_18_,
avg(Population_Age_65_) as Population_Age_65_
from
Chicago_Sum_Population
group by Zip_Code
order by Zip_Code;
--Chicago_AVG_Population
```

Console | Timing | Datasets | Charts

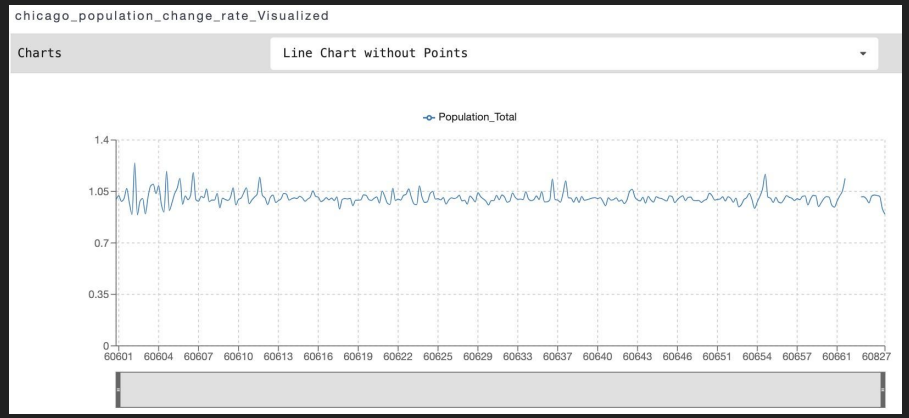
chicago_avg_population (59 rows)

Views

	Zip_Code (int)	Population_Total (real)	Population_Age_0_17 (real)	Population_Age_18_ (real)	Population_Age_65_ (real)
0	60601	14768.75	881	13887.75	2188
1	60602	1281.75	133	1148.75	9.75
2	60603	1171.75	30	1141.75	163.5
3	60604	755.25	21.25	734	75.25
4	60605	29901.5	2530	27371.5	2947.25
5	60606	3222.5	109.25	3113.25	341.5
6	60607	29795	3288.25	26506.75	1587.75
7	60608	80741	14795.5	65945.5	8262
8	60609	61211.5	15738.5	45473	7449.25
9	60610	40735.75	3620.75	37115	6273
10	60611	33537.25	1928.5	31608.75	7149
11	60612	33548.5	7569.75	25978.75	3323.75
12	60613	51388.75	6135.5	45253.25	4690
13	60614	71943	10316.75	61626.25	6584
14	60615	41279.5	6275.75	35003.75	6112.25
15	60616	53575.75	7562.5	46013.25	8700.25
16	60617	83095.25	21497.5	61597.75	13054.75
17	60618	94756.5	20679	74077.5	8440.25
18	60619	61829.5	13029.75	48799.75	10831
19	60620	68481.25	15655.25	52826	12189.25

Merge data to calculate change rate of population

The population change rate is calculated based on the total population and average population, and is within a reasonable range based on the data shown in the chart.




```
Select * from
Chicago_Sum_Population
union all
select * from
COVID19_final_All_Population
order by Zip_Code;
```

Console | Timing | Datasets | Charts

chicago_final_population (290 rows)

Views

	Zip_Code (int)	Year (date)	Population_Total (float)	Population_Age_0_17 (float)	Population_Age_18_ (float)	Population_Age_65_ (float)
0	60601	2018-00-01	14675	820	13855	2075
1	60601	2019-00-01	15083	880	14203	2109
2	60601	2020-00-01	14513	825	13688	2605
3	60601	2021-00-01	14804	999	13805	1963
4	60601	2023-00-01	15814	1067	14747	2211
5	60602	2018-00-01	1244	149	1095	5
6	60602	2019-00-01	1145	149	996	6
7	60602	2020-00-01	1596	115	1481	4
8	60602	2021-00-01	1142	119	1023	24
9	60602	2023-00-01	1261	111	1150	234
10	60603	2018-00-01	1174	56	1118	112
11	60603	2019-00-01	1052	16	1036	120
12	60603	2020-00-01	1186	15	1171	198
13	60603	2021-00-01	1275	33	1242	224
14	60603	2023-00-01	1288	80	1208	220
15	60604	2018-00-01	782	38	744	93
16	60604	2019-00-01	823	36	787	72
17	60604	2020-00-01	729	5	724	65
18	60604	2021-00-01	687	6	681	71
19	60604	2023-00-01	897	50	847	126

Thank You!