Analyzing a real world data-set with SQL and Python

Estimated time needed: **15** minutes

Objectives

After completing this lab you will be able to:

- Understand a dataset of selected socioeconomic indicators in Chicago
- Learn how to store data in an SQLite database.
- Solve example problems to practice your SQL skills

Selected Socioeconomic Indicators in Chicago

The city of Chicago released a dataset of socioeconomic data to the Chicago City Portal. This dataset contains a selection of six socioeconomic indicators of public health significance and a “hardship index,” for each Chicago community area, for the years 2008 – 2012.

Scores on the hardship index can range from 1 to 100, with a higher index number representing a greater level of hardship.

A detailed description of the dataset can be found on the city of Chicago's website, but to summarize, the dataset has the following variables:

- **Community Area Number** (`ca`): Used to uniquely identify each row of the dataset

- **Community Area Name** (`community_area_name`): The name of the region in the city of Chicago

- **Percent of Housing Crowded** (`percent_of_housing_crowded`): Percent of occupied housing units with more than one person per room

- **Percent Households Below Poverty** (`percent_households_below_poverty`): Percent of households living below the federal poverty line

- **Percent Aged 16+ Unemployed** (`percent_aged_16_unemployed`): Percent of persons over the age of 16 years that are unemployed

- **Percent Aged 25+ without High School Diploma** (`percent_aged_25_without_high_school_diploma`): Percent of persons over the age of 25 years without a high school education

- **Percent Aged Under** 18 or Over 64:Percent of population under 18 or over 64 years of age (`percent_aged_under_18_or_over_64`): (ie. dependents)

- **Per Capita Income** (`per_capita_income_`): Community Area per capita income is estimated as the sum of tract-level aggragate incomes divided by the total population

- **Hardship Index** (`hardship_index`): Score that incorporates each of the six selected socioeconomic indicators

In this Lab, we'll take a look at the variables in the socioeconomic indicators dataset and do some basic analysis with Python.

Connect to the database

Let us first load the SQL extension and establish a connection with the database

The syntax for connecting to magic sql using sqllite is

%sql sqlite://DatabaseName

where DatabaseName will be your **.db** file

In [ ]:

```
!pip install ipython-sql
%load_ext sql
```

In [ ]:

```
import csv, sqlite3

con = sqlite3.connect("socioeconomic.db")
cur = con.cursor()
!pip install pandas
```

In [5]:

```
%sql sqlite:///socioeconomic.db
```

Store the dataset in a Table

In many cases the dataset to be analyzed is available as a .CSV (comma separated values) file, perhaps on the internet. To analyze the data using SQL, it first needs to be stored in the database.

We will first read the csv files from the given url into pandas dataframes

Next we will be using the df.to_sql() function to convert each csv file to a table in sqlite with the csv data loaded in it.

In [6]:

```
import pandas
df = pandas.read_csv('https://data.cityofchicago.org/resource/jcxq-k9xf.csv')
df.to_sql("chicago_socioeconomic_data", con, if_exists='replace', index=False,method="multi")
```

```
Out[6]:  78
```

```
In [ ]:  # Install the 'ipython-sql' and 'prettytable' libraries using pip
         !pip install ipython-sql prettytable

         # Import the 'prettytable' library, which is used to display data in a formatted table
         import prettytable

         # Set the default display format for prettytable to 'DEFAULT' (i.e., a simple table format)
         prettytable.DEFAULT = 'DEFAULT'
```

You can verify that the table creation was successful by making a basic query like:

```
In [8]:  %sql SELECT * FROM chicago_socioeconomic_data limit 5;

          * sqlite:///socioeconomic.db
         Done.
```

Out[8]:

ca	community_area_name	percent_of_housing_crowded	percent_households_below_poverty	percent_aged_16_unemployed	percent_aged_25_without_high_school_diploma
1.0	Rogers Park	7.7	23.6	8.7	18.2
2.0	West Ridge	7.8	17.2	8.8	20.8
3.0	Uptown	3.8	24.0	8.9	11.8
4.0	Lincoln Square	3.4	10.9	8.2	13.4
5.0	North Center	0.3	7.5	5.2	4.5

Problems

Problem 1

How many rows are in the dataset?

```
In [9]:  %sql select count(*) from chicago_socioeconomic_data

          * sqlite:///socioeconomic.db
         Done.
```

Out[9]:

count(*)
78

▸ Click here for the solution

Problem 2

How many community areas in Chicago have a hardship index greater than 50.0?

```
In [11]:  %sql select count(*) from chicago_socioeconomic_data where hardship_index > 50.0

           * sqlite:///socioeconomic.db
          Done.
```

Out[11]:

count(*)
38

▸ Click here for the solution

Problem 3

What is the maximum value of hardship index in this dataset?

```
In [12]:  %sql select max(hardship_index) from chicago_socioeconomic_data

           * sqlite:///socioeconomic.db
          Done.
```

Out[12]:

max(hardship_index)
98.0

▸ Click here for the solution

Problem 4

Which community area which has the highest hardship index?

```
In [13]:  %sql select community_area_name from chicago_socioeconomic_data where hardship_index = (select max(hardship_index) from chicago_socioeconomic_data)

           * sqlite:///socioeconomic.db
          Done.
```

Out[13]:

community_area_name
Riverdale

▸ Click here for the solution

Problem 5

Which Chicago community areas have per-capita incomes greater than $60,000?

```
In [14]: %sql select community_area_name from chicago_socioeconomic_data where per_capita_income_ > 60000
```

```
 * sqlite:///socioeconomic.db
Done.
```

Out[14]:

community_area_name
Lake View
Lincoln Park
Near North Side
Loop

▸ Click here for the solution

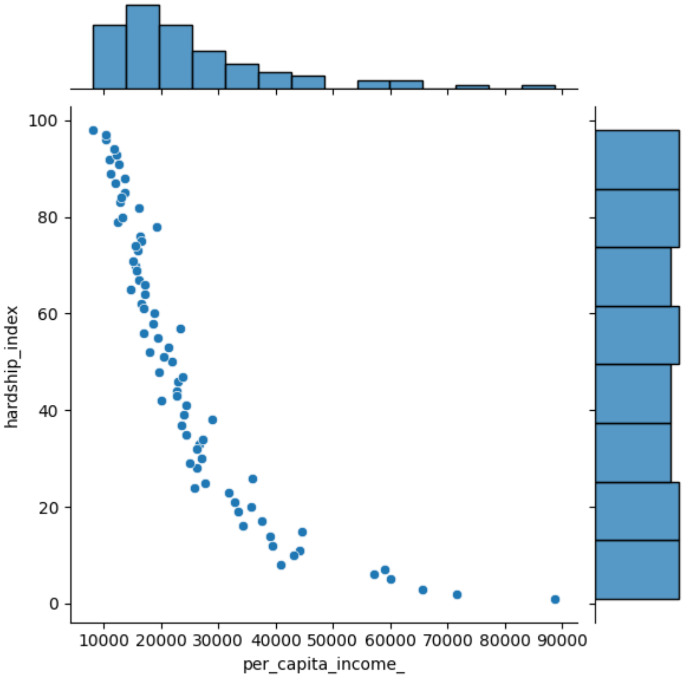
Problem 6

Create a scatter plot using the variables `per_capita_income_` and `hardship_index`. Explain the correlation between the two variables.

```
In [18]: import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns

         income_vs_hardship = %sql SELECT per_capita_income_, hardship_index FROM chicago_socioeconomic_data;
         plot = sns.jointplot(x='per_capita_income_',y='hardship_index', data=income_vs_hardship.DataFrame())
```

```
 * sqlite:///socioeconomic.db
Done.
```



▸ Click here for the solution

Conclusion

Now that you know how to do basic exploratory data analysis using SQL and python visualization tools, you can further explore this dataset to see how the variable `per_capita_income_` is related to `percent_households_below_poverty` and `percent_aged_16_unemployed`. Try to create interesting visualizations!

Summary

In this lab you learned how to store a real world data set from the internet in a database, gain insights into data using SQL queries. You also visualized a portion of the data in the database to see what story it tells.

Author

Rav Ahuja