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

```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

```
import csv, sqlite3

con = sqlite3.connect("socioeconomic.db")
cur = con.cursor()
!pip install -q pandas==1.1.5

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

```
'Connected: @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.

```
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")
```

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

```
%sql SELECT * FROM chicago_socioeconomic_data limit 5
```

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

```
[(1.0, 'Rogers Park', 7.7, 23.6, 8.7, 18.2, 27.5, 23939, 39.0),
 (2.0, 'West Ridge', 7.8, 17.2, 8.8, 20.8, 38.5, 23040, 46.0),
 (3.0, 'Uptown', 3.8, 24.0, 8.9, 11.8, 22.2, 35787, 20.0),
 (4.0, 'Lincoln Square', 3.4, 10.9, 8.2, 13.4, 25.5, 37524, 17.0),
 (5.0, 'North Center', 0.3, 7.5, 5.2, 4.5, 26.2, 57123, 6.0)]
```

Problems

Problem 1

How many rows are in the dataset?

```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data
#Correct answer: 78
```

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

```
[(78,)]
```

Problem 2

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

```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data WHERE
hardship_index > 50.0
#Correct answer: 38
```

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

```
[(38,)]
```

Problem 3

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

```
%sql SELECT MAX(hardship_index) FROM chicago_socioeconomic_data
#Correct answer: 98.0
```

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

```
[(98.0,)]
```

Problem 4

Which community area which has the highest hardship index?

```
#We can use the result of the last query to as an input to this query:
%sql SELECT community_area_name FROM chicago_socioeconomic_data where
hardship_index=98.0

#or another option:
%sql SELECT community_area_name FROM chicago_socioeconomic_data ORDER
BY hardship_index DESC LIMIT 1

#or you can use a sub-query to determine the max hardship index:
%sql select community_area_name from chicago_socioeconomic_data where
hardship_index = (select max(hardship_index) from
chicago_socioeconomic_data)

#Correct answer: 'Riverdale'
```

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

[('Riverdale',)]
```

Problem 5

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

```
%sql SELECT community_area_name FROM chicago_socioeconomic_data WHERE
per_capita_income_ > 60000
#Correct answer:Lake View,Lincoln Park, Near North Side, Loop
```

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

[('Lake View',), ('Lincoln Park',), ('Near North Side',), ('Loop',)]
```

Problem 6

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

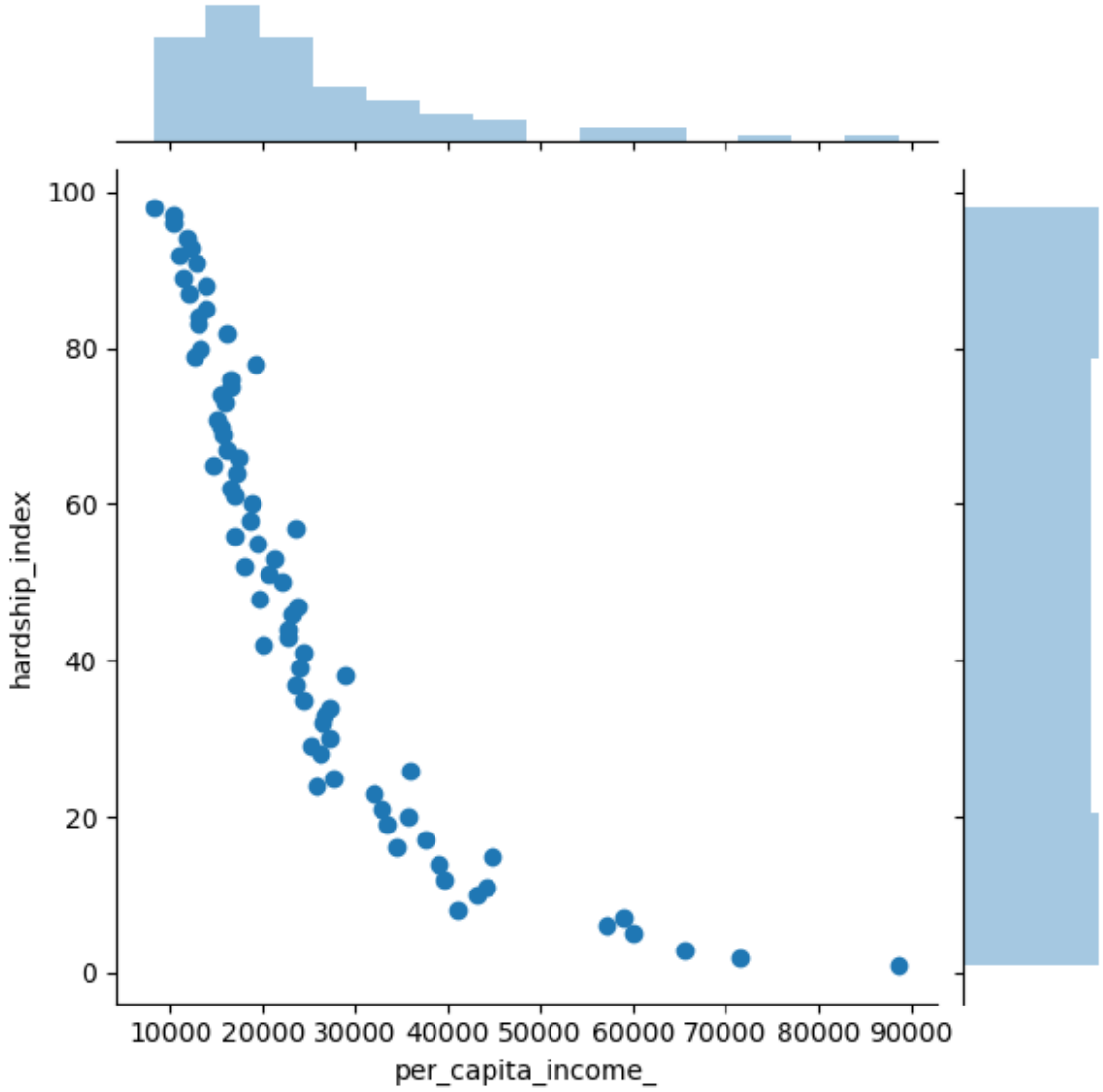
```
# if the import command gives ModuleNotFoundError: No module named
'seaborn'
# then uncomment the following line i.e. delete the # to install the
seaborn package
# !pip install seaborn
```

```
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())

#Correct answer:You can see that as Per Capita Income rises as the
Hardship Index decreases.
#We see that the points on the scatter plot are somewhat closer to a
straight line in the negative direction,
#so we have a negative correlation between the two variables.
```

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


hardship_index
per_capita_income_
0
20
40
60
80
100
10000
20000
30000
40000
50000
60000
70000
80000
90000

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

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-03-04	2.3	Lakshmi Holla	Made changes in markdown cells
2021-07-09	2.2	Malika	Updated connection string
2021-05-06	2.1	Malika Singla	Added libraries
2020-08-28	2.0	Lavanya	Moved lab to course repo in GitLab