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 Db2 database on IBM Cloud instance
- 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 following required modules are pre-installed in the Skills Network Labs environment. However if you run this notebook commands in a different Jupyter environment (e.g. Watson Studio or Ananconda) you may need to install these libraries by removing the `#` sign before `!pip` in the code cell below.

In [9]:
```
# These libraries are pre-installed in SN Labs. If running in another environment p
# !pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.3
# Ensure we don't load_ext with sqlalchemy>=1.4 (incompadible)
# !pip uninstall sqlalchemy==1.4 -y && pip install sqlalchemy==1.3.24
# !pip install ipython-sql
```

In [10]:
```
%load_ext sql
```

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

In [11]:
```
# Remember the connection string is of the format:
# %sql ibm_db_sa://my-username:my-password@hostname:port/BLUDB?security=SSL
# Enter the connection string for your Db2 on Cloud database instance below
# i.e. copy after db2:// from the URI string in Service Credentials of your Db2 ins
%sql ibm_db_sa://jwg19774:RsgH6FXWZ3OsrUD9@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n
```

Out[11]:
```
'Connected: jwg19774@bludb'
```

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 dataset source .CSV from the internet into pandas dataframe

Then we need to create a table in our Db2 database to store the dataset. The PERSIST command in SQL "magic" simplifies the process of table creation and writing the data from a `pandas` dataframe into the table

In [12]:
```
import pandas
chicago_socioeconomic_data = pandas.read_csv('https://data.cityofchicago.org/resour
%sql PERSIST chicago_socioeconomic_data
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
```

Out[12]: 'Persisted chicago_socioeconomic_data'

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

Problems

Problem 1

How many rows are in the dataset?

In [13]:
```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
Done.
```

Out[13]:

1
78

▶ Click here for the solution

Problem 2

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

In [14]:
```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data WHERE hardship_index > 50.0;
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
Done.
```

Out[14]:

1
38

▶ Click here for the solution

Problem 3

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

In [15]:
```
%sql SELECT MAX(hardship_index) FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
Done.
```

Out[15]:

1
98.0

▶ Click here for the solution

Problem 4

Which community area which has the highest hardship index?

In [16]:
```
%sql select community_area_name from chicago_socioeconomic_data where hardship_inde
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
Done.
```

Out[16]:

community_area_name
Riverdale

▶ Click here for the solution

Problem 5

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

In [17]:
```
%sql SELECT community_area_name FROM chicago_socioeconomic_data WHERE per_capita_in
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
Done.
```

Out[17]:

community_area_name
Lake View
Lincoln Park
Near North Side
Loop

▶ Click here for the solution

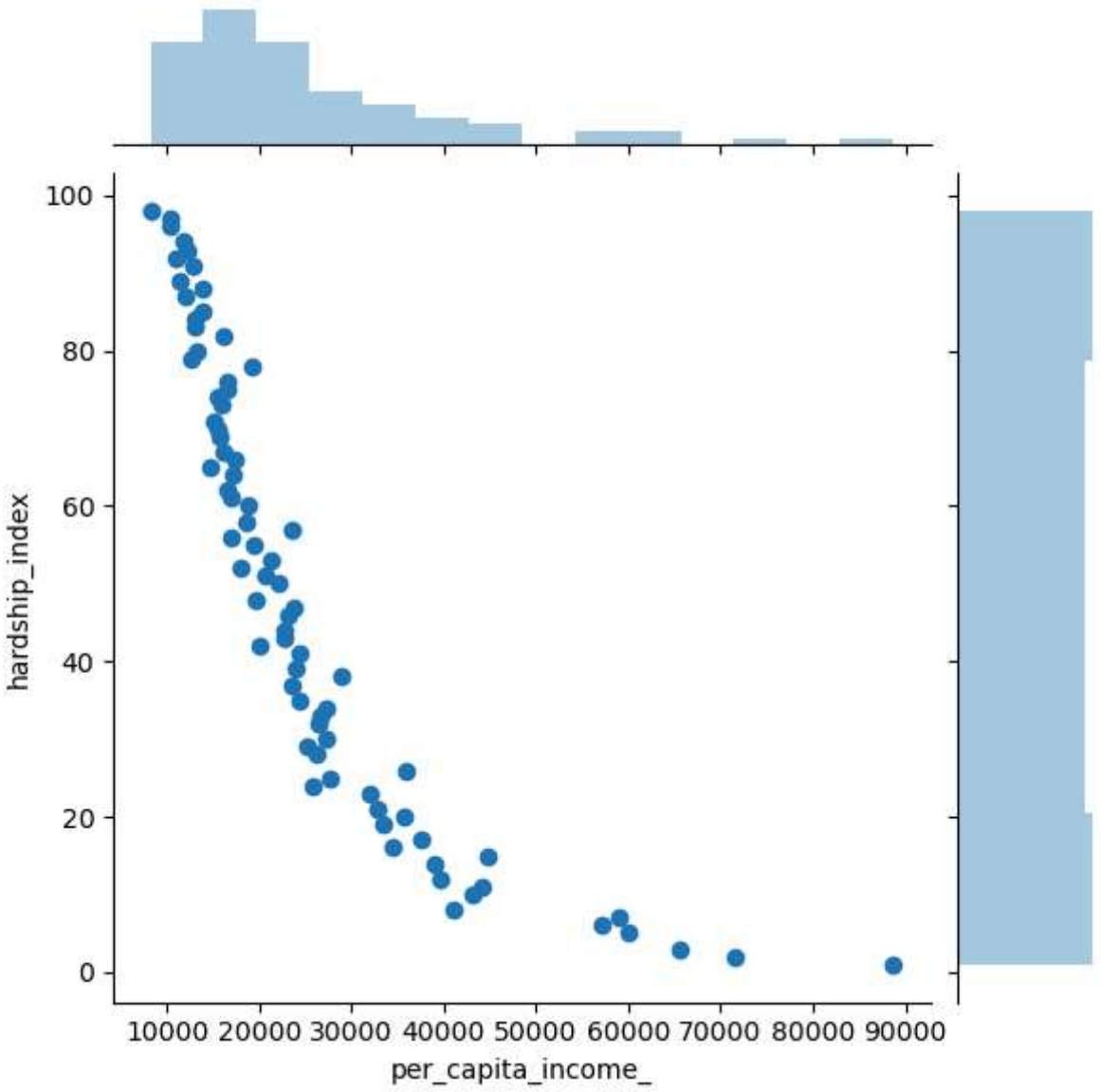
Problem 6

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_so
plot = sns.jointplot(x='per_capita_income_',y='hardship_index', data=income_vs_hard
```

```
 * ibm_db_sa://jwg19774:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98
g.databases.appdomain.cloud:32286/bludb
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 (Db2 on IBM Cloud), 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
2021-11-17	2.3	Lakshmi	Updated library
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