**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](https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDB0201ENSkillsNetwork20127838-2021-01-01), 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

[1]:



**!**pip install sqlalchemy**==**1.3.9

**!**pip install ibm\_db\_sa

Requirement already satisfied: sqlalchemy==1.3.9 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.3.9)

Requirement already satisfied: ibm\_db\_sa in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.3.3)

Requirement already satisfied: sqlalchemy>=0.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ibm\_db\_sa) (1.3.9)

[2]:



**%**load\_ext sql

[3]:



*# 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 instance. Remove the double quotes at the end.*

**%**sql ibm\_db\_sa:**//**kvn77701:INISp92TCJkh15dQ**@**54a2f15b**-**5c0f**-**46df**-**8954**-**7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733**/**BLUDB**?**security**=**SSL

[3]:

'Connected: kvn77701@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**

[4]:



**import** pandas

chicago\_socioeconomic\_data **=** pandas.read\_csv('https://data.cityofchicago.org/resource/jcxq-k9xf.csv')

**%**sql PERSIST chicago\_socioeconomic\_data

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

[4]:

'Persisted chicago\_socioeconomic\_data'

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

[5]:



**%**sql **SELECT** **\*** **FROM** chicago\_socioeconomic\_data **limit** 5;

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

Done.

[5]:

| **index** | **ca** | **community\_area\_name** | **percent\_of\_housing\_crowded** | **percent\_households\_below\_poverty** | **percent\_aged\_16\_unemployed** | **percent\_aged\_25\_without\_high\_school\_diploma** | **percent\_aged\_under\_18\_or\_over\_64** | **per\_capita\_income\_** | **hardship\_index** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1.0 | Rogers Park | 7.7 | 23.6 | 8.7 | 18.2 | 27.5 | 23939 | 39.0 |
| 1 | 2.0 | West Ridge | 7.8 | 17.2 | 8.8 | 20.8 | 38.5 | 23040 | 46.0 |
| 2 | 3.0 | Uptown | 3.8 | 24.0 | 8.9 | 11.8 | 22.2 | 35787 | 20.0 |
| 3 | 4.0 | Lincoln Square | 3.4 | 10.9 | 8.2 | 13.4 | 25.5 | 37524 | 17.0 |
| 4 | 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?**

[6]:



**%**sql **SELECT** **COUNT**(**\***) **FROM** chicago\_socioeconomic\_data;

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

Done.

[6]:

| **1** |
| --- |
| 78 |

Click here for the solution

**Problem 2**

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

[7]:



**%**sql **SELECT** **COUNT**(**\***) **FROM** chicago\_socioeconomic\_data **WHERE** hardship\_index **>** 50.0;

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

Done.

[7]:

| **1** |
| --- |
| 38 |

Click here for the solution

**Problem 3**

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

**Did you know? IBM Watson Studio lets you build and deploy an AI solution, using the best of open source and IBM software and giving your team a single environment to work in.**[**Learn more here.**](https://cocl.us/ibm_watson_studio_infobox)

[8]:



**%**sql **SELECT** MAX(hardship\_index) **FROM** chicago\_socioeconomic\_data;

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

Done.

[8]:

| **1** |
| --- |
| 98.0 |

Click here for the solution

**Problem 4**

**Which community area which has the highest hardship index?**

[ ]:



​

Click here for the solution

*#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 NULLS LAST FETCH FIRST ROW ONLY;

*#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'

**Problem 5**

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

[9]:



**%**sql **SELECT** community\_area\_name **FROM** chicago\_socioeconomic\_data **WHERE** per\_capita\_income\_ **>** 60000;

\* ibm\_db\_sa://kvn77701:\*\*\*@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB

Done.

[9]:

| **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.**

[ ]:



​

Click here for the solution

*# 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.

**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.**