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?cm_mmc=Email_Newsletter-_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork-20127838&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newslette_-Developer_Ed%2BTech-_-WW_WW-_-SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork-20127838&cm_mmca1=000026UJ&cm_mmca2=10006555&cm_mmca3=M12345678&cvosrc=email.Newslette 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.

In [ ]:

```
```

Connect to the database

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

In [1]:

```
%load_ext sql
```

In [2]:

```
# Remember the connection string is of the format:
# %sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name
# 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 instanc
e. Remove the double quotes at the end.
%sql ibm_db_sa://tnm91075:krrsj3j7zjn%40nphp@dashdb-txn-sbox-yp-dal09-12.services.dal.b
luemix.net:50000/BLUDB
```

Out[2]:

```
'Connected: tnm91075@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 [4]:

```
import pandas as pd
chicago_socioeconomic_data = pd.read_csv('https://data.cityofchicago.org/resource/jcxq-
k9xf.csv')
%sql PERSIST chicago_socioeconomic_data
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
```

Out[4]:

```
'Persisted chicago_socioeconomic_data'
```

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

In [7]:

```
%sql SELECT * FROM chicago_socioeconomic_data limit 3;
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[7]:

index	ca	community_area_name	percent_of_housing_crowded	percent_households_below_p
0	1.0	Rogers Park	7.7	
1	2.0	West Ridge	7.8	
2	3.0	Uptown	3.8	

Problems

Problem 1

How many rows are in the dataset?

In [40]:

```
#The SQL COUNT() function returns the number of rows in a table
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[40]:

1
78

▶ Click here for the solution

Problem 2

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

In [21]:

```
%sql SELECT COUNT(ca) as ca_greater_50 FROM chicago_socioeconomic_data WHERE hardship_i
ndex > 50.0;
#or
#%sql SELECT COUNT(*) as ca_greater_50 FROM chicago_socioeconomic_data WHERE hardship_i
ndex > 50.0;
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[21]:

ca_greater_50
38

▶ Click here for the solution

Problem 3

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

In [26]:

```
%sql SELECT MAX(hardship_index) as max_value FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[26]:

max_value
98.0

▶ Click here for the solution

Problem 4

Which community area which has the highest hardship index?

In [39]:

```
%sql SELECT community_area_name FROM chicago_socioeconomic_data where hardship_index=9
8.0
#or
%sql select community_area_name from chicago_socioeconomic_data where hardship_index =
 (select max(hardship_index) from chicago_socioeconomic_data)
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[39]:

community_area_name
Riverdale

▶ Click here for the solution

Problem 5

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

In [45]:

```
%sql SELECT community_area_name FROM chicago_socioeconomic_data WHERE per_capita_income
_ > 60000;
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/BLUDB
Done.
```

Out[45]:

community_area_name
Lake View
Lincoln Park
Near North Side
Loop

▶ Click here for the solution

Problem 6

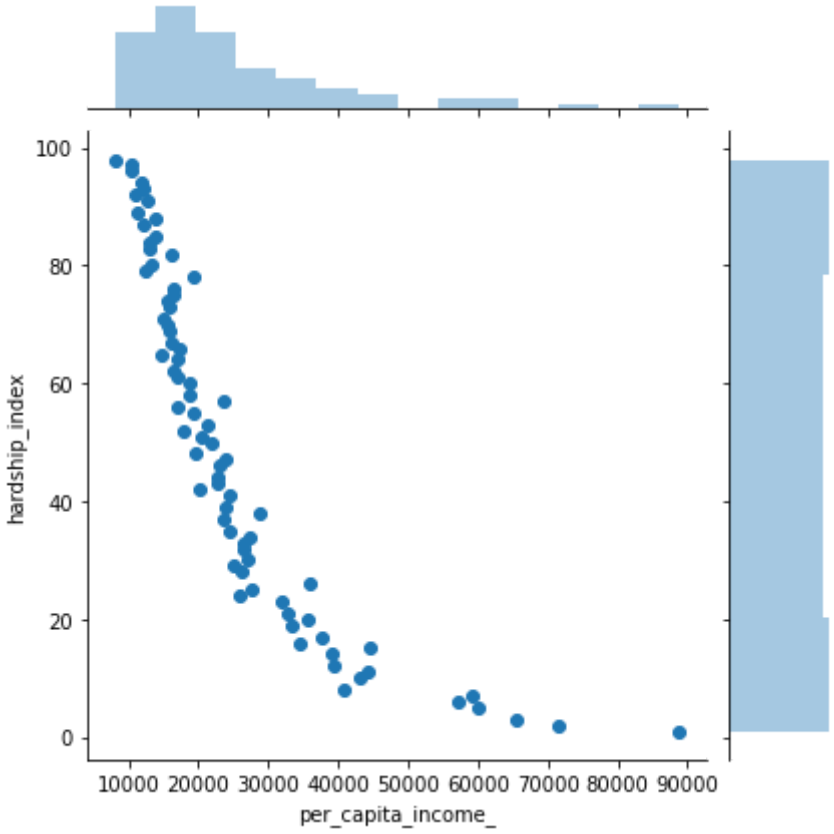
Create a scatter plot using the variables `per_capita_income_` and `hardship_index`. Explain the correlation between the two variables.

In [50]:

```
# 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
#we first collect the data under the title of income_vs_hardship
income_vs_hardship = %sql SELECT per_capita_income_, hardship_index FROM chicago_socioe
conomic_data;
#jointplot() to Draw a plot of two variables with bivariate and univariate graphs.
plot = sns.jointplot(x='per_capita_income_',y='hardship_index', data=income_vs_hardship
.DataFrame())
```

```
 * ibm_db_sa://tnm91075:***@dashdb-txn-sbox-yp-dal09-12.services.dal.bluem
ix.net:50000/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 storv it tells.

Author

Rav Ahuja (https://www.linkedin.com/in/ravahuja/)

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-08-28	2.0	Lavanya	Moved lab to course repo in GitLab