DB0201EN-Week3-1-4-Analyzing-v5-py

February 13, 2019

Lab: Analyzing a real world data-set with SQL and Python

1 Introduction

This notebook shows how to store a dataset into a database using and analyze data using SQL and Python. In this lab you will: 1. Understand a dataset of selected socioeconomic indicators in Chicago 1. Learn how to store data in an Db2 database on IBM Cloud instance 1. Solve example problems to practice your SQL skills

1.1 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.

1.1.1 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 instance
        %sql ibm_db_sa://ttk07945:kk41nf3cg7lr9s-7@dashdb-txn-sbox-yp-dal09-04.services.dal.blue
```

```
Out[2]: 'Connected: ttk07945@BLUDB'
```

1.1.2 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 [3]: import pandas
        chicago_socioeconomic_data = pandas.read_csv('https://data.cityofchicago.org/resource/jc
        %sql PERSIST chicago_socioeconomic_data
```

```
 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
```

```
Out[3]: 'Persisted chicago_socioeconomic_data'
```

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

```
In [4]: %sql SELECT * FROM chicago_socioeconomic_data limit 5;

 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.


Out[4]: [(0, 1.0, 'Rogers Park', 39.0, 23939, 8.7, 18.2, 27.5, 23.6, 7.7),
         (1, 2.0, 'West Ridge', 46.0, 23040, 8.8, 20.8, 38.5, 17.2, 7.8),
         (2, 3.0, 'Uptown', 20.0, 35787, 8.9, 11.8, 22.2, 24.0, 3.8),
         (3, 4.0, 'Lincoln Square', 17.0, 37524, 8.2, 13.4, 25.5, 10.9, 3.4),
         (4, 5.0, 'North Center', 6.0, 57123, 5.2, 4.5, 26.2, 7.5, 0.3)]

In [6]: chicago_socioeconomic_data.head()

Out[6]:     ca community_area_name  hardship_index  per_capita_income_  \
        0  1.0         Rogers Park            39.0               23939
        1  2.0          West Ridge            46.0               23040
        2  3.0              Uptown            20.0               35787
        3  4.0      Lincoln Square            17.0               37524
        4  5.0        North Center             6.0               57123

           percent_aged_16_unemployed  percent_aged_25_without_high_school_diploma  \
        0                         8.7                                        18.2
        1                         8.8                                        20.8
        2                         8.9                                        11.8
        3                         8.2                                        13.4
        4                         5.2                                         4.5

           percent_aged_under_18_or_over_64  percent_households_below_poverty  \
        0                              27.5                              23.6
        1                              38.5                              17.2
        2                              22.2                              24.0
        3                              25.5                              10.9
        4                              26.2                               7.5

           percent_of_housing_crowded
        0                         7.7
        1                         7.8
        2                         3.8
        3                         3.4
        4                         0.3
```

1.2 Problems

1.2.1 Problem 1

How many rows are in the dataset?

```
In [9]: %sql SELECT COUNT(*) FROM chicago_socioeconomic_data;

 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.


Out[9]: [(Decimal('78'),)]
```

Double-click **here** for the solution.

1.2.2 Problem 2

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

```
In [11]: %sql select community_area_name,hardship_index from chicago_socioeconomic_data where (h

 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.


Out[11]: [('Albany Park', 53.0),
          ('Belmont Cragin', 70.0),
          ('Hermosa', 71.0),
          ('Humboldt park', 85.0),
          ('Austin', 73.0),
          ('West Garfield Park', 92.0),
          ('East Garfield Park', 83.0),
          ('North Lawndale', 87.0),
          ('South Lawndale', 96.0),
          ('Lower West Side', 76.0),
          ('Armour Square', 82.0),
          ('Oakland', 78.0),
          ('Fuller Park', 97.0),
          ('Grand Boulevard', 57.0),
          ('Washington Park', 88.0),
          ('Woodlawn', 58.0),
          ('South Shore', 55.0),
          ('Chatham', 60.0),
          ('South Chicago', 75.0),
          ('Burnside', 79.0),
          ('Roseland', 52.0),
          ('Pullman', 51.0),
          ('South Deering', 65.0),
          ('East Side', 64.0),
          ('West Pullman', 62.0),
          ('Riverdale', 98.0),
          ('Archer Heights', 67.0),
          ('Brighton Park', 84.0),
          ('McKinley Park', 61.0),
```

```
          ('New City', 91.0),
          ('West Elsdon', 69.0),
          ('Gage Park', 93.0),
          ('West Lawn', 56.0),
          ('Chicago Lawn', 80.0),
          ('West Englewood', 89.0),
          ('Englewood', 94.0),
          ('Greater Grand Crossing', 66.0),
          ('Auburn Gresham', 74.0)]
```

```
In [12]: %sql SELECT COUNT(*) FROM chicago_socioeconomic_data WHERE hardship_index > 50.0;
```

```
 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.
```

```
Out[12]: [(Decimal('38'),)]
```

Double-click **here** for the solution.

1.2.3 Problem 3

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

```
In [26]: #%sql select max(hardship_index) from chicago_socioeconomic_data;
         %sql select community_area_name, hardship_index from chicago_socioeconomic_data where h
```

```
 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.
```

```
Out[26]: [('Riverdale', 98.0)]
```

Double-click **here** for the solution.

1.2.4 Problem 4

Which community area which has the highest hardship index?

```
In [17]: %sql select community_area_name from chicago_socioeconomic_data where hardship_index =
```

```
 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.
```

```
Out[17]: [('Riverdale',)]
```

Double-click **here** for the solution.

1.2.5 Problem 5

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

```
In [19]: %sql select community_area_name,per_capita_income_ from chicago_socioeconomic_data wher

 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.


Out[19]: [('Lake View', 60058),
          ('Lincoln Park', 71551),
          ('Near North Side', 88669),
          ('Loop', 65526)]
```

Double-click **here** for the solution.

1.2.6 Problem 6

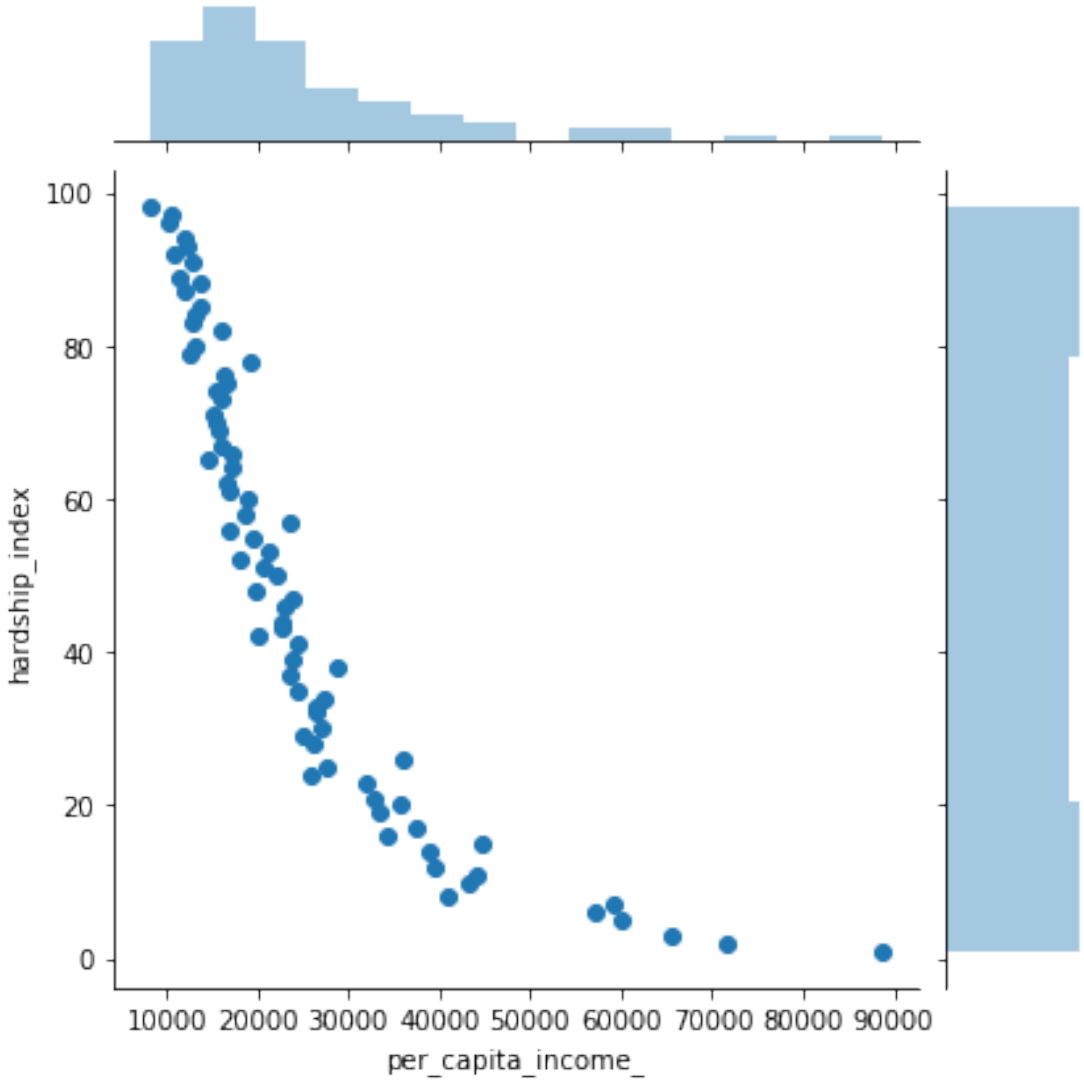
Create a scatter plot using the variables `per_capita_income_` **and** `hardship_index`**. Explain the correlation between the two variables.**

```
In [24]: #!pip install seaborn
         import matplotlib.pyplot as plt
         %matplotlib inline
         import seaborn as sns

         income_and_hardship_index = %sql Select per_capita_income_, hardship_index from chicago
         plot = sns.jointplot(x='per_capita_income_',y='hardship_index', data=income_and_hardshi

 * ibm_db_sa://ttk07945:***@dashdb-txn-sbox-yp-dal09-04.services.dal.bluemix.net:50000/BLUDB
Done.


/home/jupyterlab/conda/lib/python3.6/site-packages/scipy/stats/stats.py:1713: FutureWarning: Usi
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Double-click **here** for the solution.

1.2.7 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!**

1.3 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.