DB0201EN-Week3-1-4-Analyzing-Solved

July 1, 2022

1 Analyzing a real world data-set with SQL and Python

Estimated time needed: **15** minutes

1.1 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

1.2 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.

1.2.1 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.

[1]:
```
# These libraries are pre-installed in SN Labs. If running in another␣
 ↪environment please uncomment lines below to install them:
# !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
```

[2]:
```
%load_ext sql
```

[3]:
```
%sql ibm_db_sa://lqh43420:lb76AfXMQnkkX6lw@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.
 ↪bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB?security=SSL
```

1.2.2 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

[4]:
```
import pandas
chicago_socioeconomic_data = pandas.read_csv('https://data.cityofchicago.org/
 ↪resource/jcxq-k9xf.csv')
chicago_socioeconomic_data
```

[4]:

	ca	community_area_name	percent_of_housing_crowded	\
0	1.0	Rogers Park	7.7	
1	2.0	West Ridge	7.8	

```
2    3.0             Uptown                          3.8
3    4.0      Lincoln Square                         3.4
4    5.0        North Center                         0.3
..   …                 …                           …
73  74.0     Mount Greenwood                         1.0
74  75.0         Morgan Park                         0.8
75  76.0              O'Hare                         3.6
76  77.0           Edgewater                         4.1
77   NaN             CHICAGO                         4.7

    percent_households_below_poverty  percent_aged_16_unemployed  \
0                               23.6                        8.7
1                               17.2                        8.8
2                               24.0                        8.9
3                               10.9                        8.2
4                                7.5                        5.2
..                               …                          …
73                               3.4                        8.7
74                              13.2                       15.0
75                              15.4                        7.1
76                              18.2                        9.2
77                              19.7                       12.9

    percent_aged_25_without_high_school_diploma  \
0                                          18.2
1                                          20.8
2                                          11.8
3                                          13.4
4                                           4.5
..                                          …
73                                          4.3
74                                         10.8
75                                         10.9
76                                          9.7
77                                         19.5

    percent_aged_under_18_or_over_64  per_capita_income_  hardship_index
0                               27.5              23939            39.0
1                               38.5              23040            46.0
2                               22.2              35787            20.0
3                               25.5              37524            17.0
4                               26.2              57123             6.0
..                               …                …              …
73                              36.8              34381            16.0
74                              40.3              27149            30.0
75                              30.3              25828            24.0
76                              23.8              33385            19.0
```

```
77                                33.5                28202                NaN

[78 rows x 9 columns]
```

[5]:
```
%%sql
SELECT * FROM chicago_socioeconomic_data
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[5]:
```
[(1, 'Rogers Park', Decimal('7.7'), Decimal('23.6'), Decimal('8.7'),
Decimal('18.2'), Decimal('27.5'), 23939, 39),
 (2, 'West Ridge', Decimal('7.8'), Decimal('17.2'), Decimal('8.8'),
Decimal('20.8'), Decimal('38.5'), 23040, 46),
 (3, 'Uptown', Decimal('3.8'), Decimal('24.0'), Decimal('8.9'), Decimal('11.8'),
Decimal('22.2'), 35787, 20),
 (4, 'Lincoln Square', Decimal('3.4'), Decimal('10.9'), Decimal('8.2'),
Decimal('13.4'), Decimal('25.5'), 37524, 17),
 (5, 'North Center', Decimal('0.3'), Decimal('7.5'), Decimal('5.2'),
Decimal('4.5'), Decimal('26.2'), 57123, 6),
 (6, 'Lake View', Decimal('1.1'), Decimal('11.4'), Decimal('4.7'),
Decimal('2.6'), Decimal('17.0'), 60058, 5),
 (7, 'Lincoln Park', Decimal('0.8'), Decimal('12.3'), Decimal('5.1'),
Decimal('3.6'), Decimal('21.5'), 71551, 2),
 (8, 'Near North Side', Decimal('1.9'), Decimal('12.9'), Decimal('7.0'),
Decimal('2.5'), Decimal('22.6'), 88669, 1),
 (9, 'Edison Park', Decimal('1.1'), Decimal('3.3'), Decimal('6.5'),
Decimal('7.4'), Decimal('35.3'), 40959, 8),
 (10, 'Norwood Park', Decimal('2.0'), Decimal('5.4'), Decimal('9.0'),
Decimal('11.5'), Decimal('39.5'), 32875, 21),
 (11, 'Jefferson Park', Decimal('2.7'), Decimal('8.6'), Decimal('12.4'),
Decimal('13.4'), Decimal('35.5'), 27751, 25),
 (12, 'Forest Glen', Decimal('1.1'), Decimal('7.5'), Decimal('6.8'),
Decimal('4.9'), Decimal('40.5'), 44164, 11),
 (13, 'North Park', Decimal('3.9'), Decimal('13.2'), Decimal('9.9'),
Decimal('14.4'), Decimal('39.0'), 26576, 33),
 (14, 'Albany Park', Decimal('11.3'), Decimal('19.2'), Decimal('10.0'),
Decimal('32.9'), Decimal('32.0'), 21323, 53),
 (15, 'Portage Park', Decimal('4.1'), Decimal('11.6'), Decimal('12.6'),
Decimal('19.3'), Decimal('34.0'), 24336, 35),
 (16, 'Irving Park', Decimal('6.3'), Decimal('13.1'), Decimal('10.0'),
Decimal('22.4'), Decimal('31.6'), 27249, 34),
 (17, 'Dunning', Decimal('5.2'), Decimal('10.6'), Decimal('10.0'),
Decimal('16.2'), Decimal('33.6'), 26282, 28),
 (18, 'Montclaire', Decimal('8.1'), Decimal('15.3'), Decimal('13.8'),
Decimal('23.5'), Decimal('38.6'), 22014, 50),
```

```
 (19, 'Belmont Cragin', Decimal('10.8'), Decimal('18.7'), Decimal('14.6'),
Decimal('37.3'), Decimal('37.3'), 15461, 70),
 (20, 'Hermosa', Decimal('6.9'), Decimal('20.5'), Decimal('13.1'),
Decimal('41.6'), Decimal('36.4'), 15089, 71),
 (21, 'Avondale', Decimal('6.0'), Decimal('15.3'), Decimal('9.2'),
Decimal('24.7'), Decimal('31.0'), 20039, 42),
 (22, 'Logan Square', Decimal('3.2'), Decimal('16.8'), Decimal('8.2'),
Decimal('14.8'), Decimal('26.2'), 31908, 23),
 (23, 'Humboldt park', Decimal('14.8'), Decimal('33.9'), Decimal('17.3'),
Decimal('35.4'), Decimal('38.0'), 13781, 85),
 (24, 'West Town', Decimal('2.3'), Decimal('14.7'), Decimal('6.6'),
Decimal('12.9'), Decimal('21.7'), 43198, 10),
 (25, 'Austin', Decimal('6.3'), Decimal('28.6'), Decimal('22.6'),
Decimal('24.4'), Decimal('37.9'), 15957, 73),
 (26, 'West Garfield Park', Decimal('9.4'), Decimal('41.7'), Decimal('25.8'),
Decimal('24.5'), Decimal('43.6'), 10934, 92),
 (27, 'East Garfield Park', Decimal('8.2'), Decimal('42.4'), Decimal('19.6'),
Decimal('21.3'), Decimal('43.2'), 12961, 83),
 (28, 'Near West Side', Decimal('3.8'), Decimal('20.6'), Decimal('10.7'),
Decimal('9.6'), Decimal('22.2'), 44689, 15),
 (29, 'North Lawndale', Decimal('7.4'), Decimal('43.1'), Decimal('21.2'),
Decimal('27.6'), Decimal('42.7'), 12034, 87),
 (30, 'South Lawndale', Decimal('15.2'), Decimal('30.7'), Decimal('15.8'),
Decimal('54.8'), Decimal('33.8'), 10402, 96),
 (31, 'Lower West Side', Decimal('9.6'), Decimal('25.8'), Decimal('15.8'),
Decimal('40.7'), Decimal('32.6'), 16444, 76),
 (32, 'Loop', Decimal('1.5'), Decimal('14.7'), Decimal('5.7'), Decimal('3.1'),
Decimal('13.5'), 65526, 3),
 (33, 'Near South Side', Decimal('1.3'), Decimal('13.8'), Decimal('4.9'),
Decimal('7.4'), Decimal('21.8'), 59077, 7),
 (34, 'Armour Square', Decimal('5.7'), Decimal('40.1'), Decimal('16.7'),
Decimal('34.5'), Decimal('38.3'), 16148, 82),
 (35, 'Douglas', Decimal('1.8'), Decimal('29.6'), Decimal('18.2'),
Decimal('14.3'), Decimal('30.7'), 23791, 47),
 (36, 'Oakland', Decimal('1.3'), Decimal('39.7'), Decimal('28.7'),
Decimal('18.4'), Decimal('40.4'), 19252, 78),
 (37, 'Fuller Park', Decimal('3.2'), Decimal('51.2'), Decimal('33.9'),
Decimal('26.6'), Decimal('44.9'), 10432, 97),
 (38, 'Grand Boulevard', Decimal('3.3'), Decimal('29.3'), Decimal('24.3'),
Decimal('15.9'), Decimal('39.5'), 23472, 57),
 (39, 'Kenwood', Decimal('2.4'), Decimal('21.7'), Decimal('15.7'),
Decimal('11.3'), Decimal('35.4'), 35911, 26),
 (40, 'Washington Park', Decimal('5.6'), Decimal('42.1'), Decimal('28.6'),
Decimal('25.4'), Decimal('42.8'), 13785, 88),
 (41, 'Hyde Park', Decimal('1.5'), Decimal('18.4'), Decimal('8.4'),
Decimal('4.3'), Decimal('26.2'), 39056, 14),
 (42, 'Woodlawn', Decimal('2.9'), Decimal('30.7'), Decimal('23.4'),
```

```
Decimal('16.5'), Decimal('36.1'), 18672, 58),
 (43, 'South Shore', Decimal('2.8'), Decimal('31.1'), Decimal('20.0'),
Decimal('14.0'), Decimal('35.7'), 19398, 55),
 (44, 'Chatham', Decimal('3.3'), Decimal('27.8'), Decimal('24.0'),
Decimal('14.5'), Decimal('40.3'), 18881, 60),
 (45, 'Avalon Park', Decimal('1.4'), Decimal('17.2'), Decimal('21.1'),
Decimal('10.6'), Decimal('39.3'), 24454, 41),
 (46, 'South Chicago', Decimal('4.7'), Decimal('29.8'), Decimal('19.7'),
Decimal('26.6'), Decimal('41.1'), 16579, 75),
 (47, 'Burnside', Decimal('6.8'), Decimal('33.0'), Decimal('18.6'),
Decimal('19.3'), Decimal('42.7'), 12515, 79),
 (48, 'Calumet Heights', Decimal('2.1'), Decimal('11.5'), Decimal('20.0'),
Decimal('11.0'), Decimal('44.0'), 28887, 38),
 (49, 'Roseland', Decimal('2.5'), Decimal('19.8'), Decimal('20.3'),
Decimal('16.9'), Decimal('41.2'), 17949, 52),
 (50, 'Pullman', Decimal('1.5'), Decimal('21.6'), Decimal('22.8'),
Decimal('13.1'), Decimal('38.6'), 20588, 51),
 (51, 'South Deering', Decimal('4.0'), Decimal('29.2'), Decimal('16.3'),
Decimal('21.0'), Decimal('39.5'), 14685, 65),
 (52, 'East Side', Decimal('6.8'), Decimal('19.2'), Decimal('12.1'),
Decimal('31.9'), Decimal('42.8'), 17104, 64),
 (53, 'West Pullman', Decimal('3.3'), Decimal('25.9'), Decimal('19.4'),
Decimal('20.5'), Decimal('42.1'), 16563, 62),
 (54, 'Riverdale', Decimal('5.8'), Decimal('56.5'), Decimal('34.6'),
Decimal('27.5'), Decimal('51.5'), 8201, 98),
 (55, 'Hegewisch', Decimal('3.3'), Decimal('17.1'), Decimal('9.6'),
Decimal('19.2'), Decimal('42.9'), 22677, 44),
 (56, 'Garfield Ridge', Decimal('2.6'), Decimal('8.8'), Decimal('11.3'),
Decimal('19.3'), Decimal('38.1'), 26353, 32),
 (57, 'Archer Heights', Decimal('8.5'), Decimal('14.1'), Decimal('16.5'),
Decimal('35.9'), Decimal('39.2'), 16134, 67),
 (58, 'Brighton Park', Decimal('14.4'), Decimal('23.6'), Decimal('13.9'),
Decimal('45.1'), Decimal('39.3'), 13089, 84),
 (59, 'McKinley Park', Decimal('7.2'), Decimal('18.7'), Decimal('13.4'),
Decimal('32.9'), Decimal('35.6'), 16954, 61),
 (60, 'Bridgeport', Decimal('4.5'), Decimal('18.9'), Decimal('13.7'),
Decimal('22.2'), Decimal('31.3'), 22694, 43),
 (61, 'New City', Decimal('11.9'), Decimal('29.0'), Decimal('23.0'),
Decimal('41.5'), Decimal('38.9'), 12765, 91),
 (62, 'West Elsdon', Decimal('11.1'), Decimal('15.6'), Decimal('16.7'),
Decimal('37.0'), Decimal('37.7'), 15754, 69),
 (63, 'Gage Park', Decimal('15.8'), Decimal('23.4'), Decimal('18.2'),
Decimal('51.5'), Decimal('38.8'), 12171, 93),
 (64, 'Clearing', Decimal('2.7'), Decimal('8.9'), Decimal('9.5'),
Decimal('18.8'), Decimal('37.6'), 25113, 29),
 (65, 'West Lawn', Decimal('5.8'), Decimal('14.9'), Decimal('9.6'),
Decimal('33.6'), Decimal('39.6'), 16907, 56),
```

```
 (66, 'Chicago Lawn', Decimal('7.6'), Decimal('27.9'), Decimal('17.1'),
Decimal('31.2'), Decimal('40.6'), 13231, 80),
 (67, 'West Englewood', Decimal('4.8'), Decimal('34.4'), Decimal('35.9'),
Decimal('26.3'), Decimal('40.7'), 11317, 89),
 (68, 'Englewood', Decimal('3.8'), Decimal('46.6'), Decimal('28.0'),
Decimal('28.5'), Decimal('42.5'), 11888, 94),
 (69, 'Greater Grand Crossing', Decimal('3.6'), Decimal('29.6'),
Decimal('23.0'), Decimal('16.5'), Decimal('41.0'), 17285, 66),
 (70, 'Ashburn', Decimal('4.0'), Decimal('10.4'), Decimal('11.7'),
Decimal('17.7'), Decimal('36.9'), 23482, 37),
 (71, 'Auburn Gresham', Decimal('4.0'), Decimal('27.6'), Decimal('28.3'),
Decimal('18.5'), Decimal('41.9'), 15528, 74),
 (72, 'Beverly', Decimal('0.9'), Decimal('5.1'), Decimal('8.0'), Decimal('3.7'),
Decimal('40.5'), 39523, 12),
 (73, 'Washington Height', Decimal('1.1'), Decimal('16.9'), Decimal('20.8'),
Decimal('13.7'), Decimal('42.6'), 19713, 48),
 (74, 'Mount Greenwood', Decimal('1.0'), Decimal('3.4'), Decimal('8.7'),
Decimal('4.3'), Decimal('36.8'), 34381, 16),
 (75, 'Morgan Park', Decimal('0.8'), Decimal('13.2'), Decimal('15.0'),
Decimal('10.8'), Decimal('40.3'), 27149, 30),
 (76, "O'Hare", Decimal('3.6'), Decimal('15.4'), Decimal('7.1'),
Decimal('10.9'), Decimal('30.3'), 25828, 24),
 (77, 'Edgewater', Decimal('4.1'), Decimal('18.2'), Decimal('9.2'),
Decimal('9.7'), Decimal('23.8'), 33385, 19),
 (None, 'CHICAGO', Decimal('4.7'), Decimal('19.7'), Decimal('12.9'),
Decimal('19.5'), Decimal('33.5'), 28202, None)]
```

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

1.3 Problems

1.3.1 Problem 1

How many rows are in the dataset?

[6]:
```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[6]:
```
[(78,)]
```

Click here for the solution

```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data;

Correct answer: 78
```

1.3.2 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://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[7]:
```
[(38,)]
```

Click here for the solution

```
%sql SELECT COUNT(*) FROM chicago_socioeconomic_data WHERE hardship_index > 50.0;

Correct answer: 38
```

1.3.3 Problem 3

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

[8]:
```
%sql SELECT MAX(hardship_index) FROM chicago_socioeconomic_data;
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[8]:
```
[(98,)]
```

Click here for the solution

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

Correct answer: 98.0
```

1.3.4 Problem 4

Which community area which has the highest hardship index?

[9]:
```
%sql select community_area_name from chicago_socioeconomic_data where␣
↪hardship_index = ( select max(hardship_index) from␣
↪chicago_socioeconomic_data )
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[9]:
```
[('Riverdale',)]
```

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 NU

#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 = ( selec

Correct answer: 'Riverdale'
```

1.3.5 Problem 5

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

[10]:
```
%sql SELECT community_area_name FROM chicago_socioeconomic_data WHERE␣
↪per_capita_income_ > 60000;
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

[10]:
```
[('Lake View',), ('Lincoln Park',), ('Near North Side',), ('Loop',)]
```

Click here for the solution

```
%sql SELECT community_area_name FROM chicago_socioeconomic_data WHERE per_capita_income_ > 600

Correct answer:Lake View,Lincoln Park, Near North Side, Loop
```

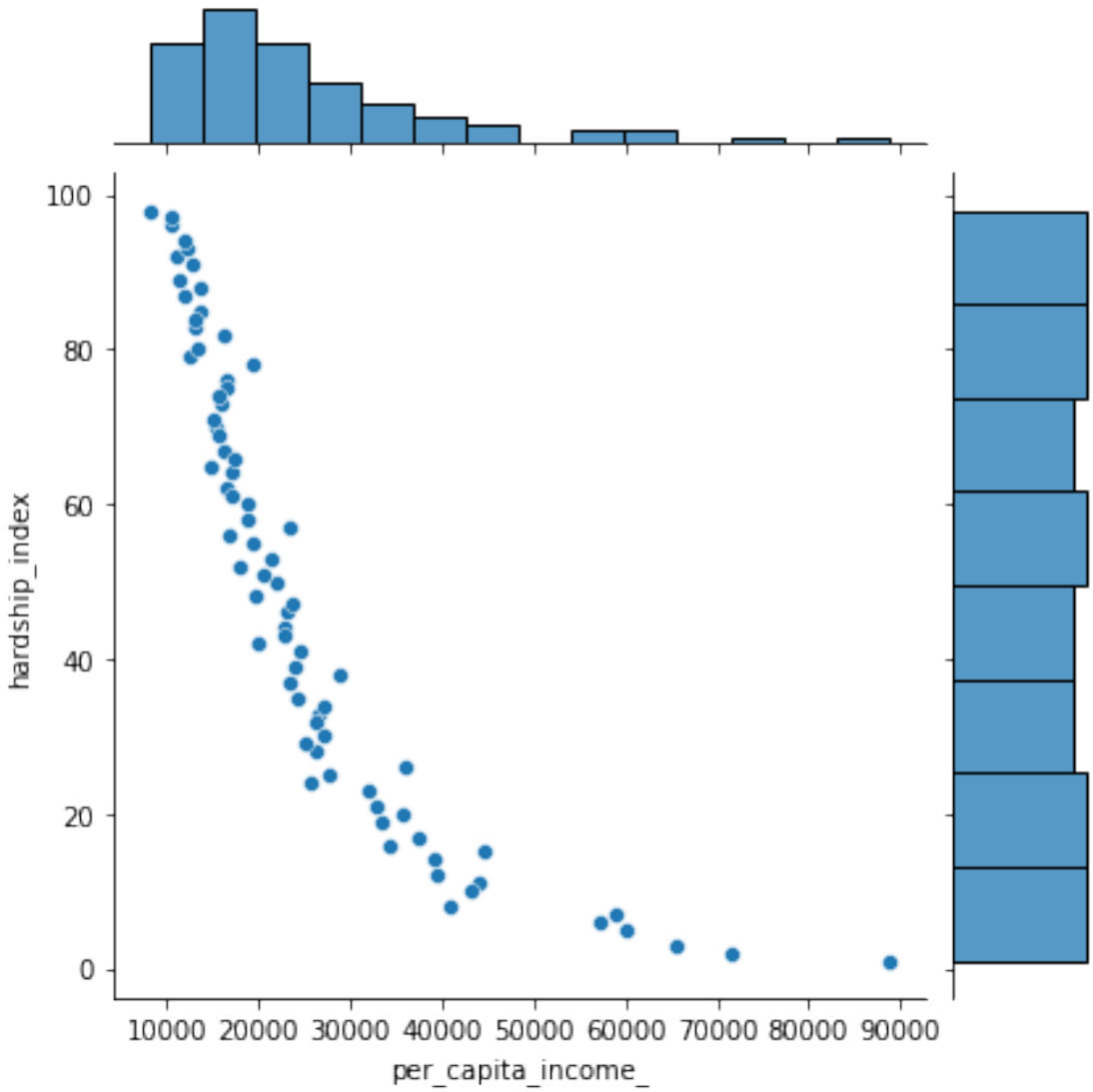
1.3.6 Problem 6

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

[11]:
```
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())
```

```
 * ibm_db_sa://lqh43420:***@ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```



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==0.9.0

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

income_vs_hardship = %sql SELECT per_capita_income_, hardship_index FROM chicago_socioeconomic
plot = sns.jointplot(x='per_capita_income_',y='hardship_index', data=income_vs_hardship.DataFra

Correct answer:You can see that as Per Capita Income rises as the Hardship Index decreases. We
```

1.3.7 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!

1.4 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.

1.5 Author

Rav Ahuja

1.6 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

##