Devoted Data Science Challenge

The following exercise is designed to both evaluate and give you a sense for some of the data science skills we're currently looking for at Devoted Health. There are three parts to the challenge: SQL skills, cleaning messy data, and scraping data from a web page.

Background

Health plans are required to maintain a provider directory so members can know which doctors are in their network and available to see. This data challenge focuses on a toy example of some of the data that might be used to build a plan's directory. Because providers periodically move, retire, or change practices, plans tend to rely on several data sources for information about who is in their network, where they practice, and how they can be reached. Here we'll look at a couple examples of ingesting and working with this type of data.

SQL Skills

This component asks you to execute a few queries against a small postgresql database. You'll receive the credentials for connecting to the database server separately and are free to use any database software of your choosing (if you don't have one off-hand, dbeaver is a fine option).

A brief data dictionary is below (all of the tables live in the **directory** schema):

Table	Column	Description
providers	npi	National provider identifier
providers	full_name	Full provider name
providers	is_pcp	Indicator for primary care providers
medical_groups	medical_group_id	Medical Group Identifier
medical_groups	medical_group_name	Name of the group
provider_groups	npi	National provider identifier
provider_groups	medical_group_id	Medical Group identifier
provider_contact_info	npi	National provider identifier
provider_contact_info	address1	Street address
provider_contact_info	address2	Unit/Suite Number
provider_contact_info	city	City name
provider_contact_info	state	State code (2-letter)
provider_contact_info	zip	Zip code (5 digits)
provider_contact_info	phone	Phone number
provider_contact_info	fax	Fax number
provider_contact_info	data_source	Source of contact information
provider_contact_info	confidence_score	Quality score for contact record

Table	Column	Description
provider_contact_info	update_date	Date contact info was updated

In particular, note that the `provider_contact_info` contains the full history of data updates each of a variety of sources, rather than just the most current information. That is, each time new information is received from a given source, it is appended to this table (so a given provider/source combination may have multiple records).

For each question below, please include both your SQL query and requested result (**Please use only SQL to perform each analysis for this section**):

1. How many providers does each group have? How many primary care providers?

2. Which providers do not have associated contact info (order the list by NPI and include the top 20 results)?

3. The `provider_contact_info` table contains contact information from many different data sources with different levels of confidence and recency. Using these data:

   (a) For each provider, find the record associated with the most recent update (order the list by NPI and include the top 20 results)
   (b) For each provider, find the record associated with the highest quality source that has been updated in the last 60 days (order the list by NPI and include the top 20 results)
   (c) For each provider/data source combination, find the current phone number, previous phone number, and flag whether the number has changed (order the list by provider then data source and include the top 20 results)

Cleaning Data

Next, suppose you received the file `new_phone_records.csv` with data from a new source.

1. Using python, clean these phone numbers to prepare them for upload into the database as a 10-digit string. Attach your code for cleaning the data as well as a CSV with your resulting file for upload sorted by NPI then phone number.

2. Comment on your observations from cleaning the data. Did you identify any issues with the data while doing so? If so, give a few examples.

3. Compare the new phone numbers to the existing ones in the database (you can do so either in SQL or python environment, but please include your code for answering each question):

   (a) How many providers who did not previously have a phone number in the database have one in the new data source?

   (b) How many of the records in the new data source match a phone number we already had associated with that provider

   (c) How many records in the new data source conflict with a phone number we already had associated with that provider?

Scraping Data

Finally, you might want to make use of data publically available on the web. Here, we'd like to include scores from the popular provider rating site DJ's Docstars.

1. Using python, scrape the ratings from this site and export them to a CSV for upload to the database. Attach your code for scraping the ratings as well as the data in a CSV sorted by NPI.

2. What is the average rating for providers in the plan's network (that is, the providers in the SQL database you looked at above)? What about for the plan's Primary Care providers? (you can do this either in SQL or python, but please include your code along with your answer)