Tapestry Bridge Program for QCEW postgreSQL databases within pgAdmin4

Fauwial Khan

Program Brief

This is an R program that connects the unsuppressed QCEW 'year' from pgAdmin4 (PostgreSQL Client) to 'R'. It will create a queryable data frame in R for Total ownership of all counties at the NAICS6 Level.

You will require:

- pgAdmin4
- PostgreSQL
- Rstudio
- Login Credentials for the PgAdmin4 server (NKN_Taprestry_DataWarehouse)

Install and Load DBI and RPostgreSQL packages.

Once you have installed the required software you can proceed to the following steps to load the packages below.

DBI: DBI separates the connectivity to the DBMS into a "front-end" and a "back-end". Applications use only the exposed front-end API. The back-end facilities that communicate with specific DBMSs are provided by *drivers* that get called automatically. In our case we have PostgreSQL as the DBMS.

RPostgreSQL: RPostgreSQL provides a Database Interface 'DBI' compliant *driver* for 'R' to access our 'PostgreSQL' QCEW database system.

In case of you not having them, enter **install.packages("package_name")** and replace package_name with **DBI** or **RPostgreSQL**, depending on what is missing to successfully install them on your machine before running the below code.

```
library(DBI)
library(RPostgreSQL)
```

Set up the connection to NKN server

The Northwest Knowledge Network (NKN) is a unit within the University of Idaho (UI) Office of Research and Development (ORED) that provides research data management and computing support for UI researchers and their regional, national, and international collaborators. NKN operates as a partnership between ORED, the UI Information Technology Services Unit, and the UI Library.

More information can be accessed on https://vivo.nkn.uidaho.edu/vivo/display/n19885

The NKN server is the host to which the pgAdmin4 client establishes a connection with. The unsuppresed QCEW 'year' PostgreSQL database is within this server. It is accessed during server setup and inputting login credentials in pgAdmin4. The code chunk below demonstrates defining the variables used registering and logging onto the NKN server.

```
host <- "db.nkn.uidaho.edu"
port <- 5434
dbname <- "qcew_2002" #enter database name e.g. :qcew_2001
user <- "fauwial"     #enter your username for pgAdmin4
password <- "nknpassword"
```

Variable Definitions:

- **host** : The hostname/address that is being connected to.
- **port** : The port defines where network connections start and end.it is **5434** for our case.
- **dbname** : The QCEW database name within the server in the pgAdmin4 client. It is named exactly as is from pgAdmin4 to ensure the program picks the correct database.
- **user** : This is the username credential that is given to you by the administrator of the pgAdmin4 server.
- **password** : This is the password credential given to you by the administrator, It is standard throughout as **nknpassword**.

Create the Connection Object

Now that the connection variables have been defined and setup the next step is to create a connection object. This object will namely link the PostgreSQL database to your R session and our previously defined variables. The object will be used to communicate with the database engine. The code chunk below does exactly that and keeps the naming schema simple as to avoid confusion for any new user.

The *con* variable is set to perform this by using the dbConnect function to connect to our PostgreSQL DBMS and assigning its set authentication parameters as defined by the user.

```
con <- dbConnect(drv = PostgreSQL(),
                 dbname = dbname,
                 host = host,
                 port = port,
                 user = user,
                 password = password)
```

Runing a Query within a view from pgAdmin4

pgAdmin4 has a native SQL query functionality allowing for schema and subsequent view creation of user queries. The following code queries the QCEW_2002 year PostgreSQL database within NKN Tapestry Server and creates a view for all ownership levels across all counties across the NAICS6 tier.

```
schema <- "qcew" #enter specific schema under the pgAdmin4 database name
view <- "fact_qcew_byowner_bycnty_naics6"  #enter view name in underlying  pgAdmin4 database
```

As the schema and view have been defined in the above chunk, the next step becomes writing the query to be performed to a separate variable (**query**).

```
query <- paste0("SELECT * FROM ", schema, ".", view)
```

Once the above line is run, we use the dbSendQuery function to interface with back-end of the QCEW_2001 database to submit and execute the *query* we have defined to return resulting object. The

```
rs <- dbSendQuery(con, query)
```

The final step is creating the dataframe of the resulting query in our R Session's global environment as the *df* variable. This not only improves efficiency but also allocates lesser memory to finish the task.

```
df <- fetch(rs, n = -1)
```
