Database Connection Script

MMM Arachchi

2025-09-21

Purpose

The purpose of this script is to connect to a PostgreSQL database, verify the connection, and provide helper functions to run queries safely from R.

Pre-requisites:

- R and RStudio installed
- PostgreSQL database already created
  - Example: TCSI_Extract_DB
  - Ensure the database user (e.g., postgres) has the correct permissions to connect and run queries
- Required R packages: DBI, RPostgres, dplyr, readr, writexl Install them once in your R console using:

```
install.packages(c("DBI", "RPostgres", "dplyr", "readr", "writexl"),
                 repos = "https://cloud.r-project.org/")
```

- Optional: Ensure network/firewall rules allow access to the PostgreSQL server if it's remote.

Environment Variables

Database credentials should be stored securely in a .Renviron file to avoid hardcoding sensitive information. See the "Setup_Guide.pdf" for detailed instructions on creating this file for Windows, Mac, and Linux.

Create a `.Renviron` file with the following contents:

```
DB_HOST=localhost           # Database server IP or hostname
DB_PORT=5432                # Default PostgreSQL port
DB_NAME=TCSI_Extract_DB     # Database name
DB_USER=postgres            # Username
DB_PASSWORD=<your_password> # Database password
```

Notes:

- Replace `<your_password>` with your actual password
- On Windows, save it in `C:/Users/<username>/Documents/.Renviron` (ensure "All files" type).
- On Mac/Linux, save it in your home directory `~/.Renviron`.

Verify in R:

```r
Sys.getenv("DB_HOST")
Sys.getenv("DB_PASSWORD")
```

If the values print correctly, `.Renviron` is working.

Setup: Load Libraries and DB Parameters

Load the installed libraries into the R session so their functions are available for database operations, data manipulation, and file handling.

```r
library(DBI)
library(RPostgres)
library(dplyr)
library(readr)
library(writexl)
```

Define the database connection parameters by reading the values stored in the `.Renviron` file. These variables will be used to securely connect to the PostgreSQL database.

```r
db_config <- list(
  host = Sys.getenv("DB_HOST"),
  port = as.integer(Sys.getenv("DB_PORT")),
  dbname = Sys.getenv("DB_NAME"),
  user = Sys.getenv("DB_USER"),
  password = Sys.getenv("DB_PASSWORD")
)
```

Database Connection Function

Define a function to establish a connection to the PostgreSQL database with informative messages if the connection fails:

```r
create_db_connection <- function() {
  tryCatch({
    con <- dbConnect(
      RPostgres::Postgres(),
      host = db_config$host,
      port = db_config$port,
      dbname = db_config$dbname,
      user = db_config$user,
      password = db_config$password
    )
    message("Database connection successful!")
    return(con)
  }, error = function(e) {
    stop("Connection failed: ", e$message)
  })
}
```

Helper Function: `run_query`

Define a helper function that executes SQL queries against the database. Execute SQL queries safely, automatically opening and closing the connection:

```
run_query <- function(query) {
  con <- create_db_connection()
  on.exit(dbDisconnect(con), add = TRUE)
  dbGetQuery(con, query)
}
```

Test Connection & Examples

Test your connection and the `run_query()` function. Query outputs are hidden in this document to avoid exposing real data. You can run them in your R session to view results.

```
# Create a connection and list all tables
db_conn <- create_db_connection()
dbListTables(db_conn)
dbDisconnect(db_conn)

# Example query using run_query
run_query("SELECT * FROM students LIMIT 5;")
```

Notes & Recommendations

- Always ensure `.Renviron` is correctly configured before running queries.
- `run_query()` automatically closes connections to avoid leaving open sessions.
- For more advanced queries, you may use the `query_table()` function (see `query_table_doc.Rmd`) which supports filtering, aggregation, ordering, and limiting results.

Session Info

```
sessionInfo()
```

```
## R version 4.4.3 (2025-02-28 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22621)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=English_Australia.utf8
## [3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Australia.utf8
##
## time zone: Australia/Perth
## tzcode source: internal
##
```

```
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] writexl_1.5.4   readr_2.1.5     dplyr_1.1.4     RPostgres_1.4.8
## [5] DBI_1.2.3
##
## loaded via a namespace (and not attached):
##  [1] vctrs_0.6.5       cli_3.6.4         knitr_1.50        rlang_1.1.5
##  [5] xfun_0.51         generics_0.1.3    glue_1.8.0        bit_4.6.0
##  [9] htmltools_0.5.8.1 hms_1.1.3         rmarkdown_2.29    evaluate_1.0.3
## [13] tibble_3.2.1      tzdb_0.5.0        fastmap_1.2.0     yaml_2.3.10
## [17] lifecycle_1.0.4   compiler_4.4.3    blob_1.2.4        timechange_0.3.0
## [21] pkgconfig_2.0.3   rstudioapi_0.17.1 digest_0.6.37     R6_2.6.1
## [25] tidyselect_1.2.1  pillar_1.10.1     magrittr_2.0.3    tools_4.4.3
## [29] bit64_4.6.0-1     lubridate_1.9.4
```