**Assignment 4**

It is advisable to read over the whole assignment before making a start.
When submitting screenshots, please resize them to keep the total submission file size down to below 8 MB, the Hub's upload size limit (the smaller the better, as long as the screenshots are readable).

Loading and analysing CSV files is a very common task in data science. CSV (comma-separated value) files are a simple and useful format for sharing datasets; they are often made available for download. CSV files may use other delimiters (such as tab) instead of the comma.

**Theory: psql options**

You can supply various options when calling psql, including:

| | |
|---|---|
| -h *host* | set the host (such as localhost or the AWS server) |
| -p *port* | set the port (default 5432) |
| -U *user* | connect to Postgres with a particular Postgres username |
| -f *file* | run an SQL file (do queries or load data) |
| -E | show extra information about psql operations |
| -d *database* | connect to a particular database |

Once you have connected to a database server, you can run any SQL statement (queries, inserts, deletes, etc).

You can also use these specific psql commands:

| | |
|---|---|
| help | show help |
| \q | quit |
| \l | list databases |
| \c *database* | connect to a particular database |
| \dt | list tables in current database |
| \lv | list views |
| \du | list database users |

When working with Postgres, there are often two types of account called **postgres**:

- When you install Postgres it creates its own user account under Windows or OS X; this is typically called **postgres**. You should not have to log into this account via the OS.
- The Postgres server also has a default account with username **postgres**. This is the account we will log in to via psql.

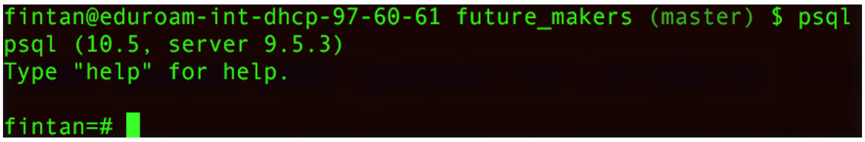
Postgres stores its data in a data directory. This is usually inside the postgres install folder, wherever that has been installed.

## Question 1

In the last coursework we made sure that we could launch psql directly from the command line. We will now look at connecting to the server.

Use psql to connect to the postgres server.

You will know you are connected once you see the psql prompt:



```
fintan@eduroam-int-dhcp-97-60-61 future_makers (master) $ psql
psql (10.5, server 9.5.3)
Type "help" for help.

fintan=#
```

If you are having trouble connecting, try using the psql options to supply the postgres username (psql may try to connect with your OS X / Windows account name by default.

The host localhost is already supplied by default.

The default password for database user **postgres** is empty. You may have set a different password during install.

If you are still having trouble connecting, make sure postgres is started: research the pg_ctl command and how to use it to start and stop the server. pg_ctl is installed in the same folder as psql, so that folder is already on your PATH you will be able to run pg_ctl first. In order to work, pg_ctl may need to be supplied with the location of your postgres data folder as an argument; look this up.

If you have password issues, try configuring the postgres server to not require a password. Look this up; it involves editing the **pg_hba.conf** file and changing the md5 authentication method to trust in order to not require a password. We recommend using Sublime Text to edit the configuration file rather than vi or emacs.

If you are still having trouble connecting, contact the instructor.

Once you are connected, take a screenshot.

## Question 2

We will now download a CSV file from the Web for analysis. Locate an interesting CSV file online, containing a dataset of your choice.

The dataset should have at least 1000 rows and at least 5 attributes (columns).

Briefly describe the CSV file you have chosen.

**Question 3**

Connect to the database server with psql and make a new database (with a name of your choice) to hold the CSV data. Research the **CREATE DATABASE** command.

Take a screenshot.

**Question 4**

Connect to the database you just created. You can either use psql's **-d** option on launch or the **\c** command inside psql.

Take a screenshot.

**Question 5**

Choose a datatype for each column of the CSV, then prepare the **CREATE TABLE** statement to make a table in which to store the CSV data. Show the **CREATE TABLE** statement here.

**Question 6**

Run the **CREATE TABLE** statement; take a screenshot.

**Question** 7

Load the CSV file into the database table you created, using psql.

Once logged in with psql, use the **COPY** command; research this command and its syntax, and check some examples.

Make sure you understand what the **DELIMITER** and **HEADER** commands do. You may need to supply the full path (from the root of your filesystem) to your CSV file.

**Question 8**

In psql, select everything from the table to check it has been imported; take a screenshot.

**Question 9**

Devise three interesting and useful queries to run on your dataset.

For each query, describe the analysis you would like to perform in English, run the query to check it works, then give the query text. Do not submit a query which you have not verified to work.

Use each of the following keywords at least once across the three queries: **WHERE**, **GROUP BY**, and **OVER**.

## Question 10

Devise a query which returns a table as a result rather than a single value. Describe the analysis it performs in English, then give the query text.

Now save the results of this query as a new table. To do this, use the **CREATE TABLE AS** command; research this command and look at some examples.

Select everything from this new table and take a screenshot.

## Question 11

Export the new table you just created as a CSV file. You can also do this with the **COPY** command, which can write CSV files as well as reading them; look at some examples.

Open the CSV file in a text editor and take a screenshot.