**Assessment task 2: Collaborative Development of an end to end project using Centralized Code Repositories + Github usage analysis and reflection of the project**

**Intent:**Is to tie all the pieces taught in this Subject together. Do an end to end analysis and collaborate as a team using Github. To analyse the github usage of the entire group and reflection of your own project in terms of what worked and what didn’t work.

**Objective(s):**

This task addresses the following subject learning objectives: 1 and 2

This assessment task contributes to the development of course intended learning outcome(s): 2.1, 2.2, 2.4, 3.3, 4.1 and 5.1

**Type:**Report/Code

**Groupwork:**Group, group assessed and Individual

**Weight: 30**% + 10%

**Criteria:**

1. Research on the effective data stores. And designing the datawarehouse appropriately. (Group), 20%
2. Using SQL/R/Python to do basic analysis. (Group), 20%
3. Clarity and why a certain programming language was chosen. (Group), 10%
4. Appropriateness of commits and branches to collaborate within a team using Git, adhering to one of the documented workflows. (Group), 10%
5. Clarity and efficiency of content review and change negotiation using Pull Requests, and successful incorporation of individual changes into the team’s master branch. (Group), 20%
6. Presenting and clearly communicating your findings as a report. (Group), 10%
7. Clarity on highlighting the individual and teams efforts on github usage.(Individual), 5%
8. Articulating what worked and didn’t work in during the course of the project. (Individual), 5%

# Introduction

Our project mainly focus on solar energy: consumption, installation

# Background

Solar power in Australia is a fast-growing industry. As of June 2020, Australia's over 2.46 million solar PV installations had a combined capacity of 17,616 MW photovoltaic (PV) solar power, of which 4,148 MW were installed in the preceding 12 months. In 2019, 59 solar PV projects with a combined capacity of 2,881 MW were either under construction, constructed or due to start construction having reached financial closure. Solar accounted for 5.2% (or 11.7 TWh) of Australia's total electrical energy production (227.8 TWh) in 2018.

The sudden rise in Solar PV installations in Australia since 2018 dramatically propelled the country from being considered a relative laggard to a strong contender in Solar PV development by mid 2019. With an installed photovoltaic capacity of 16.3 GW at the end of 2019, Australia has the highest per capita solar capacity at 600 watts per capita, overtaking Germany with 580 watts per capita.

# Data collected

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data File** | **Description** | **Data Frequency** | **Data Format** | **Source** |
| TOTAL NUMBER OF ACCREDITED INSTALLERS AND DESIGNERS.csv | Number of solar system installers accredited by state by year | Annual | CSV | Clean Energy Council |
| IDCJAC0003\_<weatherStation>\_Data1.csv | Daily solar exposure as measured at 20 weather stations across all states in Australia | Daily | CSV | BOM |
| ANNUAL SOLAR PV INSTALLATIONS8.csv | Number of solar system installed by state by year | Annual | CSV | Clean Energy Council |
| ANNUAL INSTALLED CAPACITY OF SOLAR PV (MW).csv | Total capacity of solar systems generated by state by year | Annual | CSV | Clean Energy Council |
| Energy\_consumption\_data.csv | Energy consumed and number of customers by council and by postcode in QLD, by month since 2009 | Monthly | CSV | Energex |
| Postcode data for small-scale installations - sgu - solar panel.csv | Small solar systems installed by postcode, for months in 2019 and 2020 | Monthly | CSV | Energex |
| Solar Panel Technical Specs.csv | Technical specifications of 16 solar panels in the market by 2 manufacturers (Jinko and LG) | Ad-hoc | CSV | Solar Reviews |
| Solar PV system cost vs output.csv | Reference costs of solar systems as of August 2020, with average daily system output | Ad-hoc | CSV | Solar Quotes |
| solar system cost.csv | Reference minimum and maximum costs of solar systems | Ad-hoc | CSV | Solar Choice |
| Solar\_Installation\_Data.csv | Count of solar system installations by postcode in QLD | Monthly | CSV | Energex |
| PV\_Power\_-33.86882\_151.209295\_<date>-<time>.json | Estimated power generated by a fictitious solar panel at a particular location since 19/10/2020 every 30 minutes | Every 30 minutes | Solcast API in json | Solcast |
| Radiation\_-33.86882\_151.209295\_<date>-<time>.json | Solar energy received at a particular location since 19/10/2020 every 30 minutes | Every 30 minutes | Solcast API in json | Solcast |
| Summary of Benchmarks - postcodes updated March 2018.csv | Benchmarks of electricity consumption for each season, by household size, state, and climate zone | Ad-hoc | CSV | AER |
| Ausgrid Customer Energy Data | Various energy consumed in 2018 and 2019 by Ausgrid customers | Monthly | CSV | Ausgrid |

## The 5 Aspects of Big Data

To analyse the solar energy production and consumption in Australia, the team collected multiple datasets from different sources. Our final implementation transforms all these datasets into a relational database in MySQL on AWS RDS following the Dimensional modelling. This design is justified when we examine our datasets according to the following 5 aspects of big data.

### Volume

The datasets collected are not very large at all, with the size ranging from a few KB to about 2MB. The total size of all the data files is less than 20MB. This will not post difficulty on any professional database engine like MySQL.

### Velocity

### Most of the datasets are static in nature. That mean they were downloaded from particular websites. Since the kind of analysis we are attempting does not require real-time data, this approach will be sufficient. We do have 2 renewable datasets that we extracted from the Solcast website using their APIs, one on the amount of solar exposure a particular geographic location receives and the other on the estimated power generated by a fictitious solar panel at the same location. The dataset returned is not real-time but the values at 30 minutes interval starting from 19/10/2020. Thus, the velocity of our datasets is low and fully within the capability of MySQL which handles multiple concurrent sessions well.

### Variety

All CSV datasets are structured while the API result set is in JSON which is semi-structured. Although MySQL can handle JSON data type natively, in order to simplify the use of this data by developers, the tabular structure inside JSON was extracted with R script before storing it in normal relational tables.

### Value

They collectively will provide the team with the statistics of solar energy produced and consumed in Australia.

### Veracity

The datasets collected were all sourced from trustworthy providers (refer to table above) which ensure the accuracy and value of the data. The providers are:

* + government agencies – Australian Energy Regulator (AER) and Bureau of Meteorology (BOM)
  + operators – Ausgrid, and Energex
  + solar industry entities – Solcast (<https://solcast.com>, operating since 2016), Solar Reviews (<https://www.solarreviews.com/>, operating since 2012),

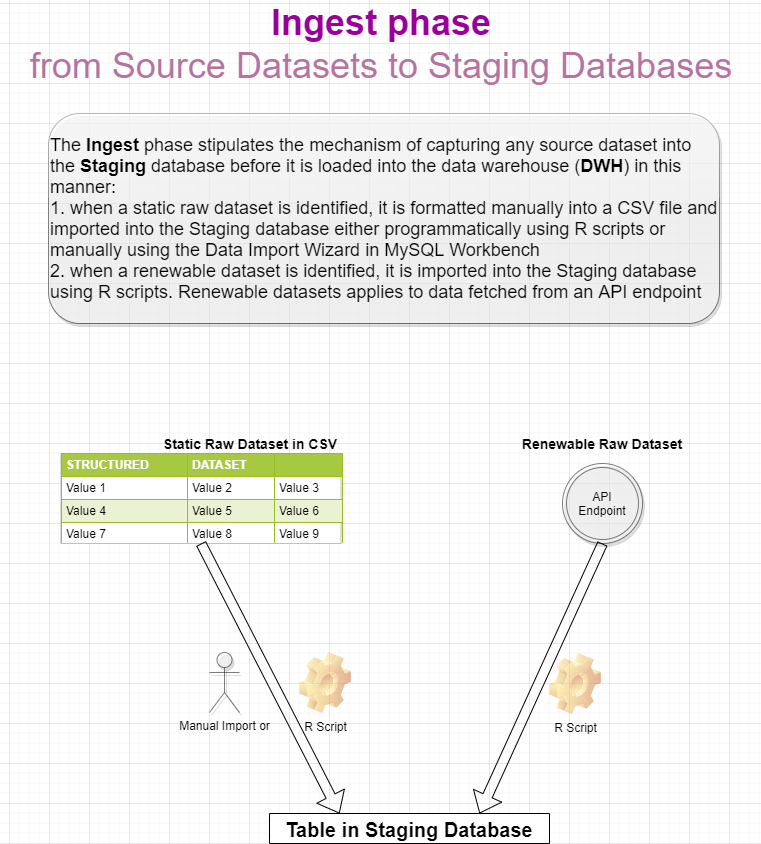
Solar Choice (<https://www.solarchoice.net.au/>, operating since 2008),

Clean Energy Council (<https://www.cleanenergycouncil.org.au/>, operating since 2007), and Solar Quotes (<https://www.solarquotes.com.au/>, operating since 2009)

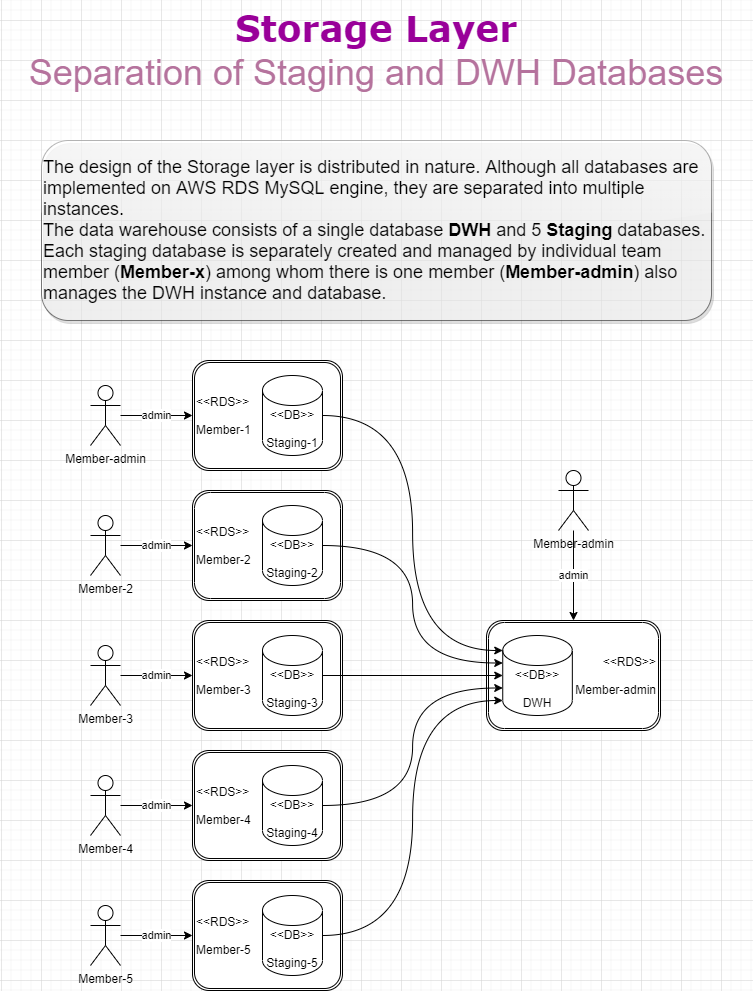
All of them provide good documentation on the data or APIs provided by them.

# Data Analytic Platform Architecture Design

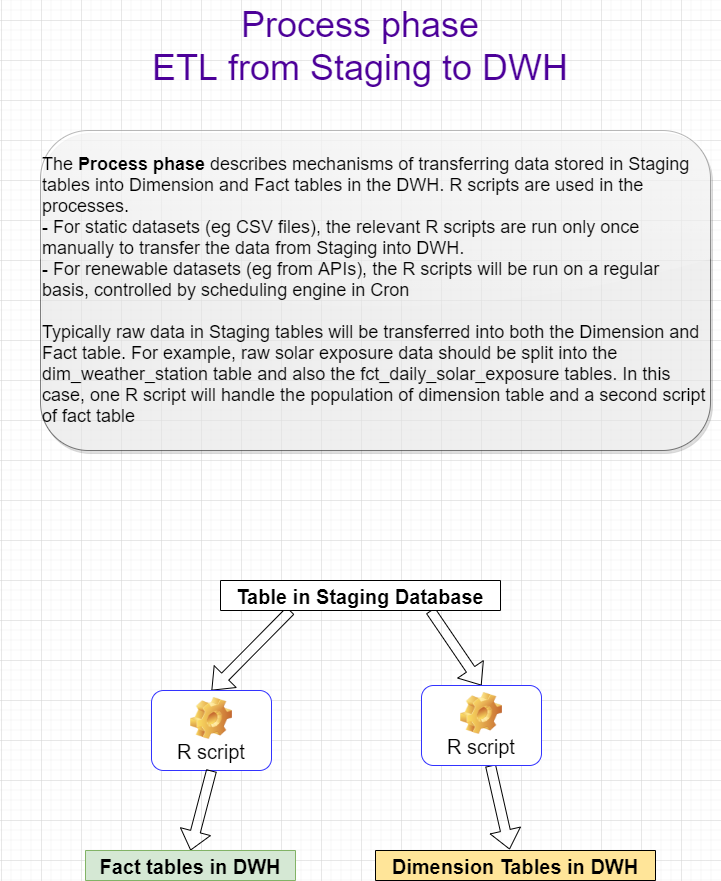
## Ingest Phase



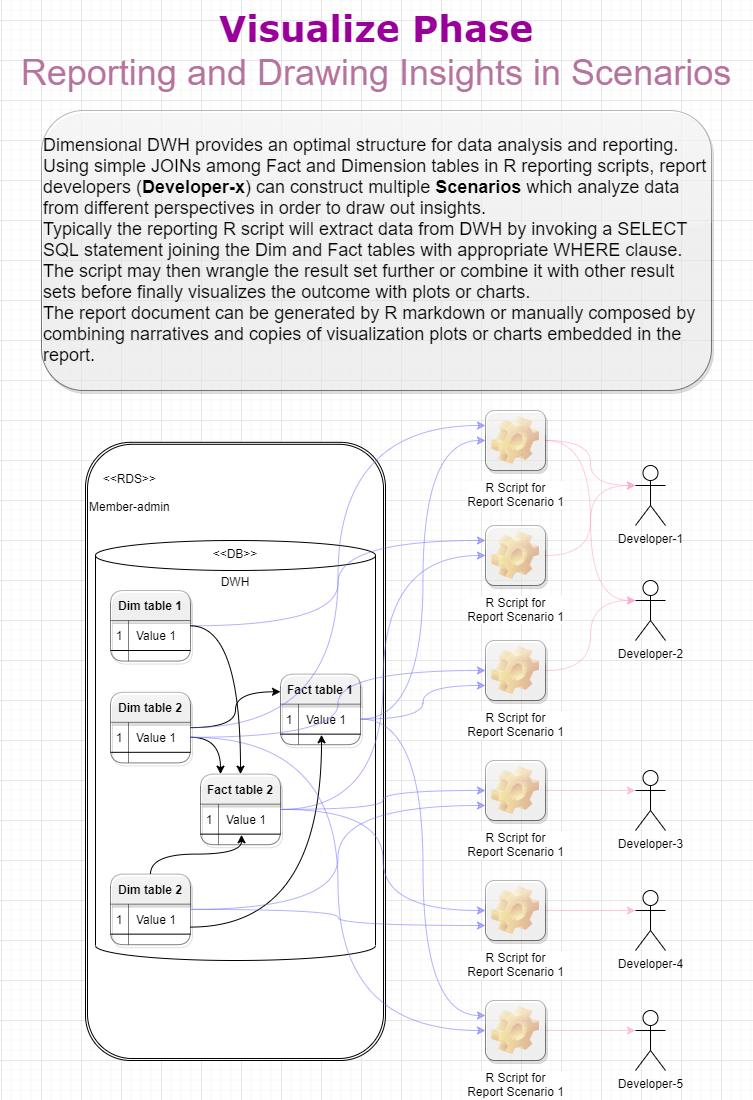
## Storage Layer



## Process Phase



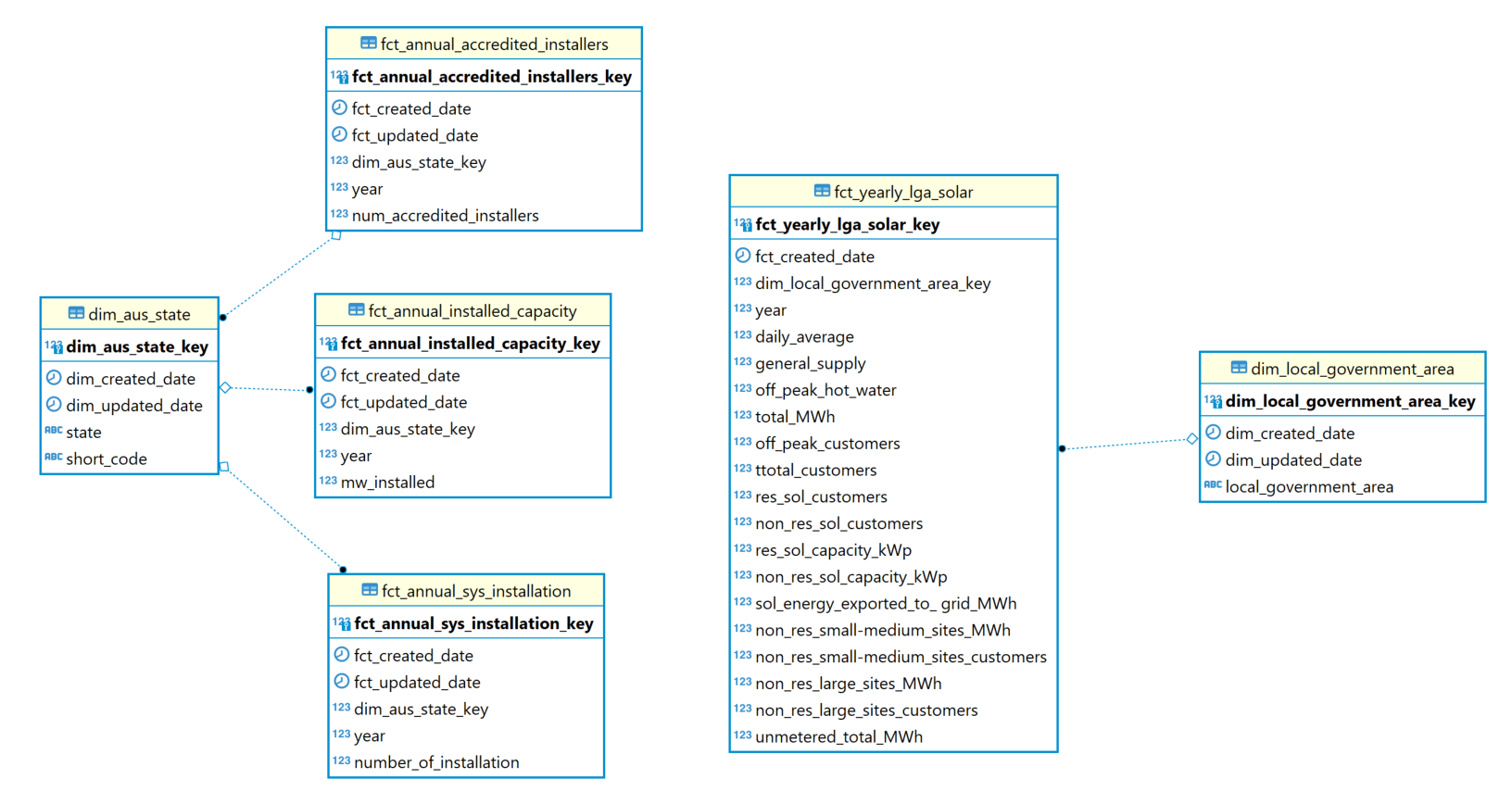
## Visualize Phase



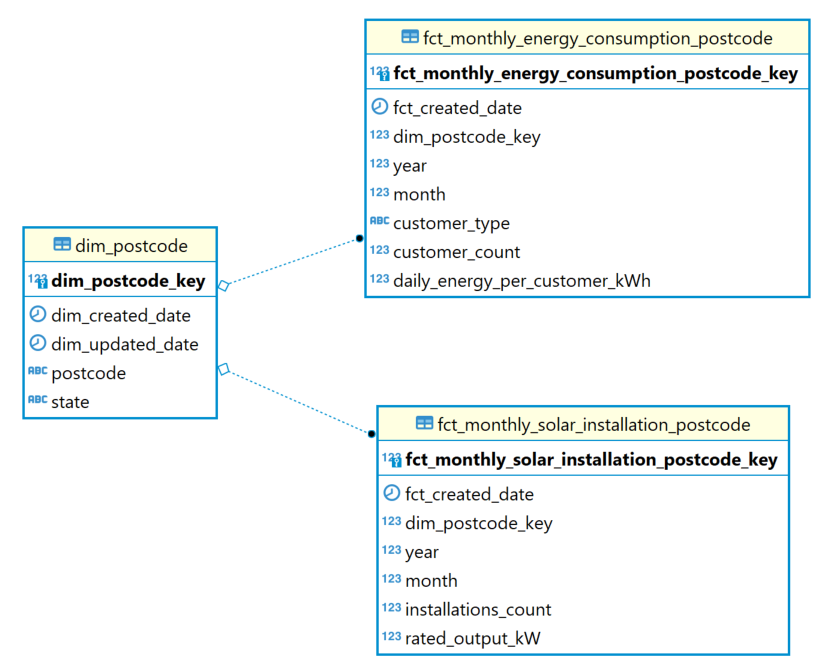
# Data warehouse design

## Dimensional Modelling and ERD

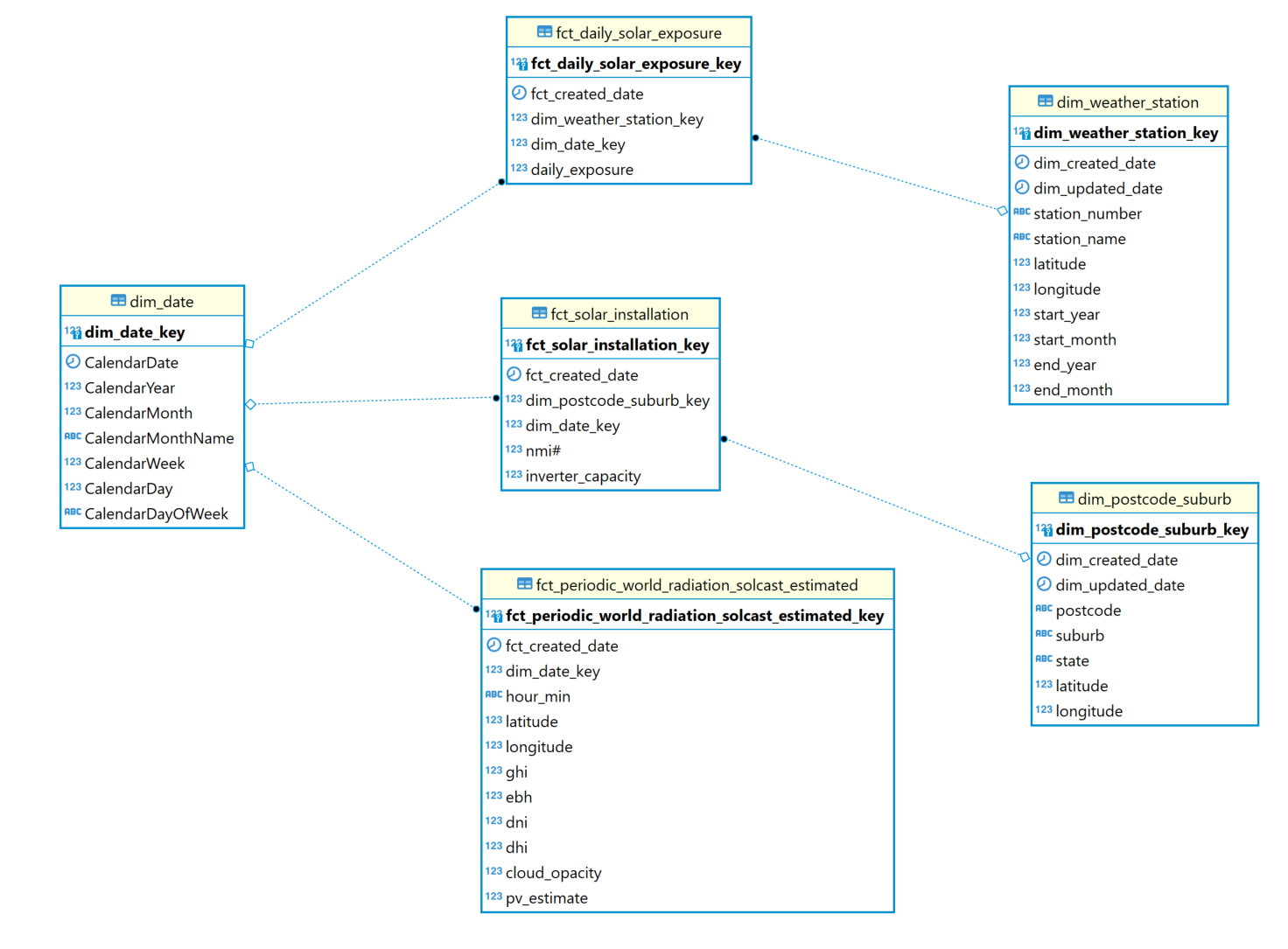
### Annual Facts



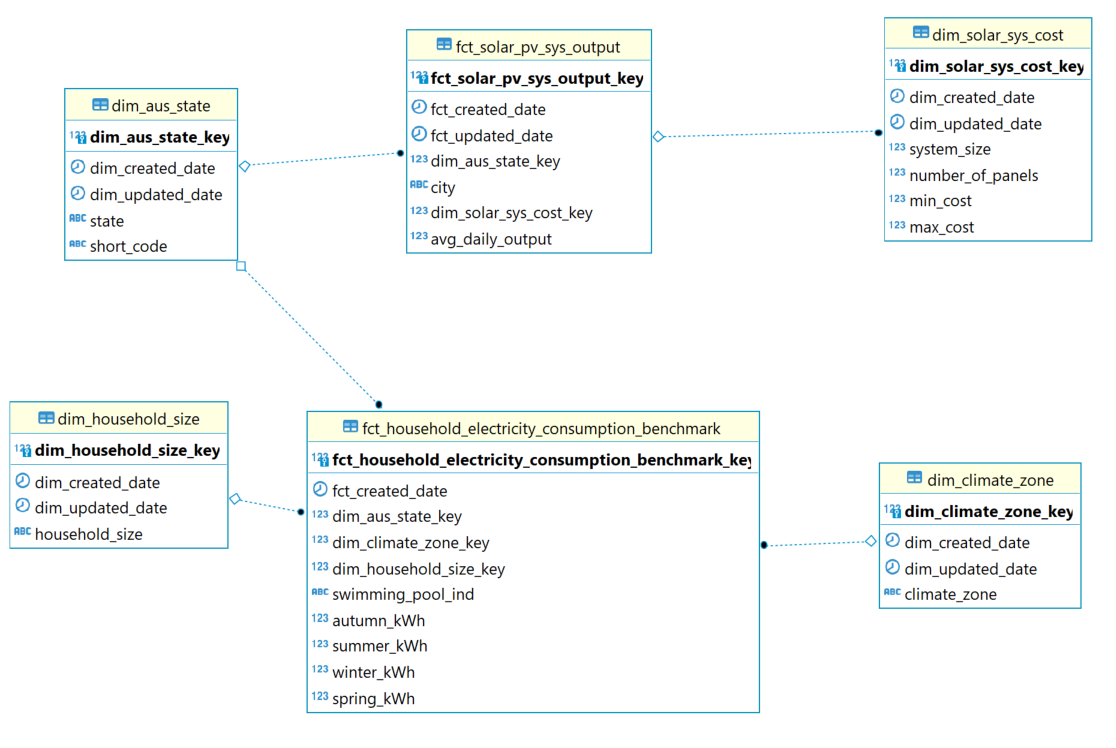
### Monthly Facts



### Daily Facts



### Ad-hoc (Snapshots)



## Physical Implementation

The implementation of the dimensional data warehouse on MySQL running on AWS RDS was based on these considerations:

* + 1. Each team member needed to subscribe to the free tier of AWS and created a running MySQL database on RDS. These individual databases are used as Staging databases to store CSV files as tables.
    2. Each member also needed to use different database tools to manage and access her own database(s) with DBvisualizer, MySQL Workbench, or DBeaver.
    3. Each member was responsible to source raw datasets and migrate them inside his or her own staging tables. Templates were provided on CREATE TABLE in SQL, and sample R scripts on migrating the files programmatically into MySQL.
    4. A centralize database was used as the main data warehouse. SQL scripts were used to create all the dim and fact tables. The population of the dim\_date table was also done in pure SQL as it can be generated automatically using CTEs. R scripts were developed to perform ETL from various staging databases to the central data warehouse.
    5. Taking this distributed approach to operate databases proves to be cost effective in this project since it allows us to spread the costs of running these databases in AWS. It also proves it was a flexible approach since it allows us to relocate the main DWH database to a different instance. The only change we needed to make was the host name of RDS in our scripts.
    6. This setup reflects the normal practice in a commercial firm to keep staging tables in databases separate from the data warehouse database.

# Data visualisation and analysis

## Data story 1:

Maximization on saving on electricity bill (domestic) / pay-off period

Solar energy Panel Generation - energy (MJ) by month, postcode / suburb / region

Normal electricity consumption

Government installation subsidy (one-off)

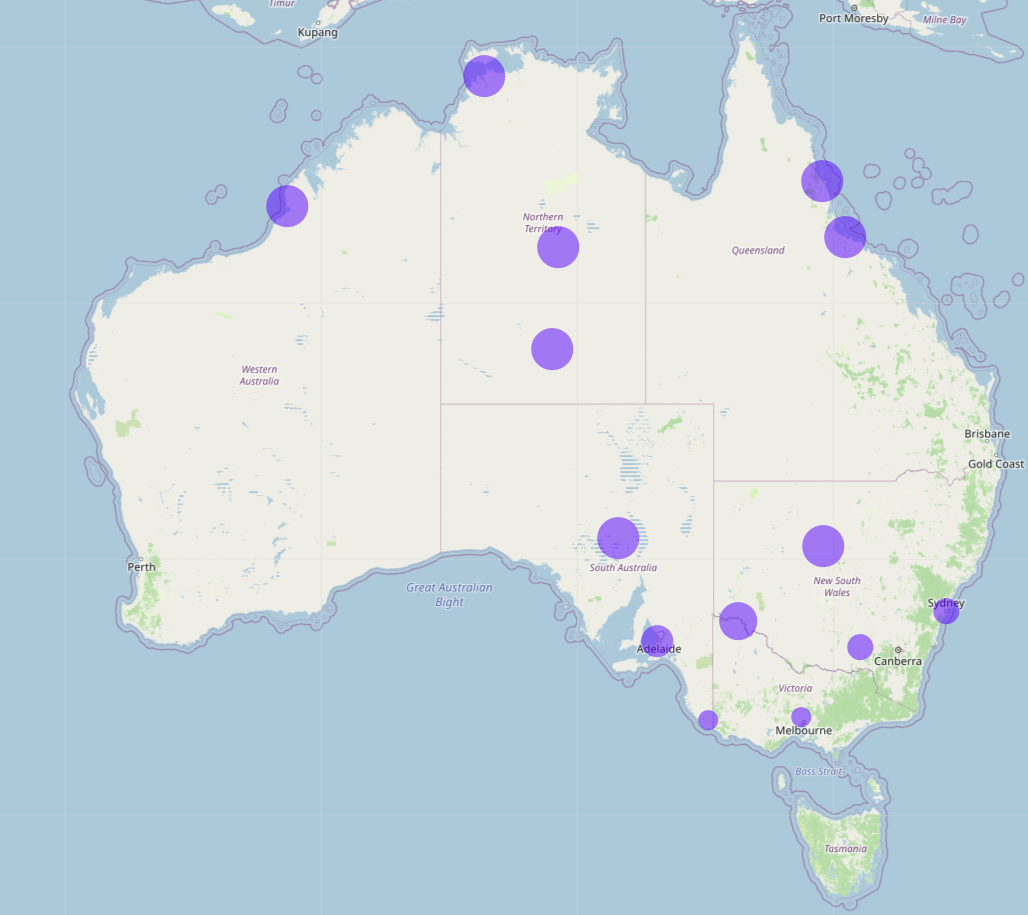
Feed-in tariff

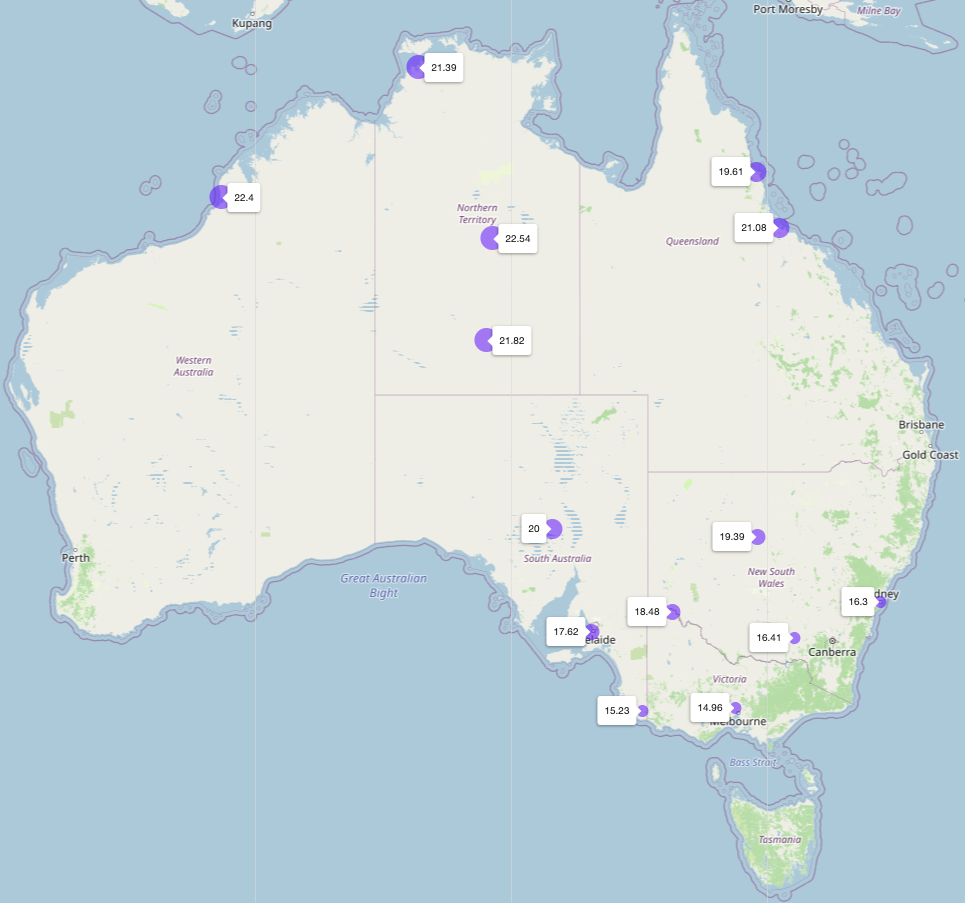
## Data story 2:

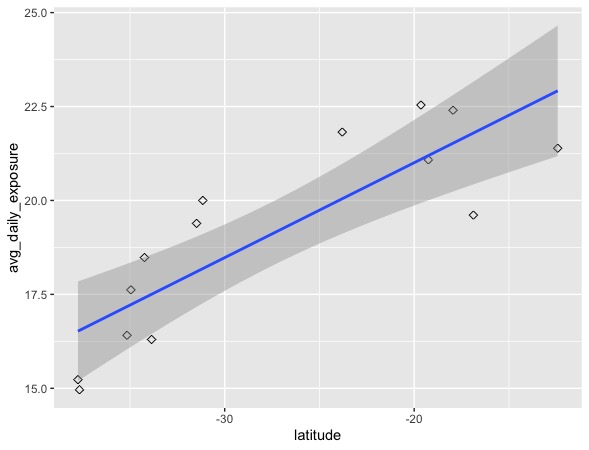
Is it possible to pay $0 in electricity bill?

## Data story 3:

How latitude/climate influence solar radiation in Australia?

I used leaflet package to draw a plot for solar exposure per station across Australia.  
Please feel free to let me know your thoughts and I'm also trying to add how much solar exposure per station on the map to make the map make more sense to the audience.  
  
I have worked out numbers that can show on the map and draw a plot reflect that, please see below screenshot:





## Data story 4:

Which size of solar system do I need to run my house entirely?

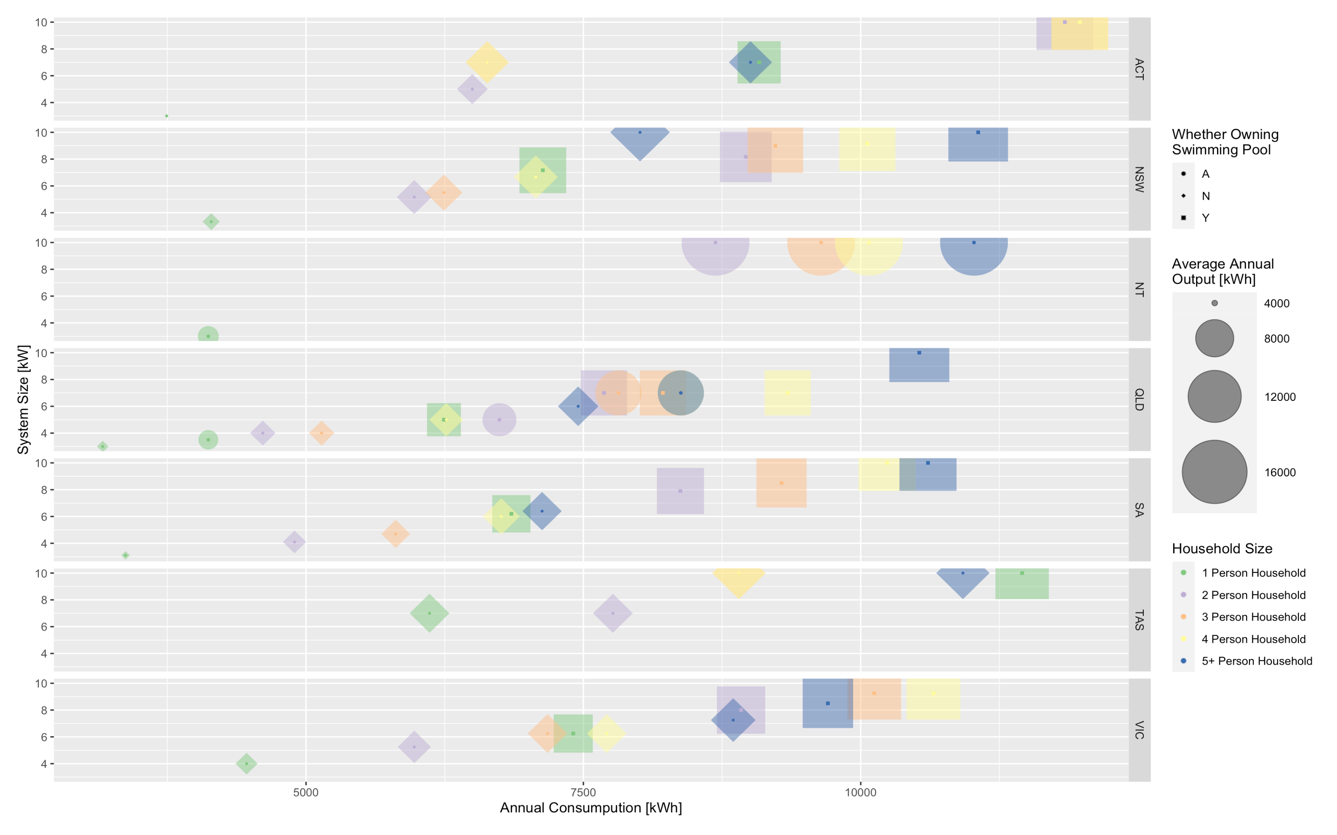


Figure.

## Data story 5:

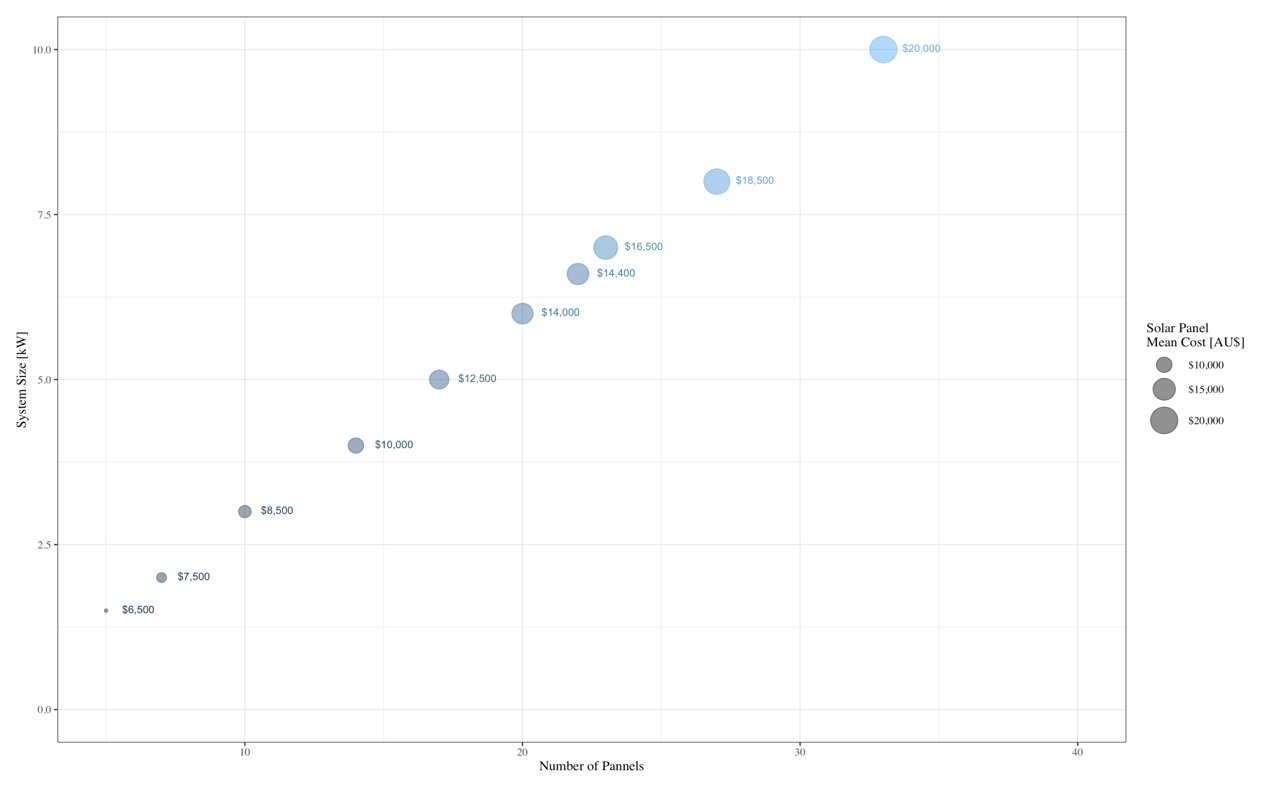
How much do you need to pay for solar system installation?

Figure.

## Data story 6:

### Scenario 6a

How many kWh per standard PV module (residential panel 65’’ by 39’’ and commercial panel 78’’ by 39’’) could have been generated at the BOM station SYDNEY (OBSERVATORY HILL) at ideal conditions (angled tilt)?

### Scenario 6b

How many kWh could have been produced by all modules installed in the Ausgrid territory and all were at those ideal conditions?

### Scenario 6c

How many kWh were actually produced?

### Scenario 6d

How much sustainable energy could be generated additionally if the density of solar panel users would be increase by e.g. 10%, 20% or 50%?

## Data story 7:

The annual change of solar PV installations and accredited installers across each state

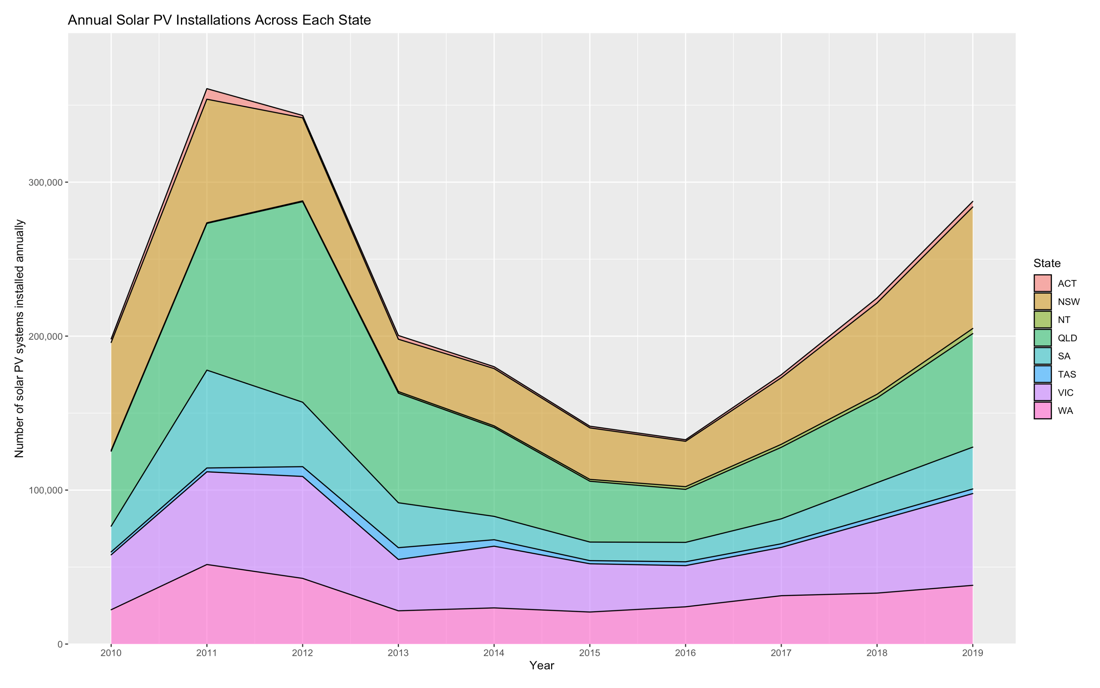


Figure.

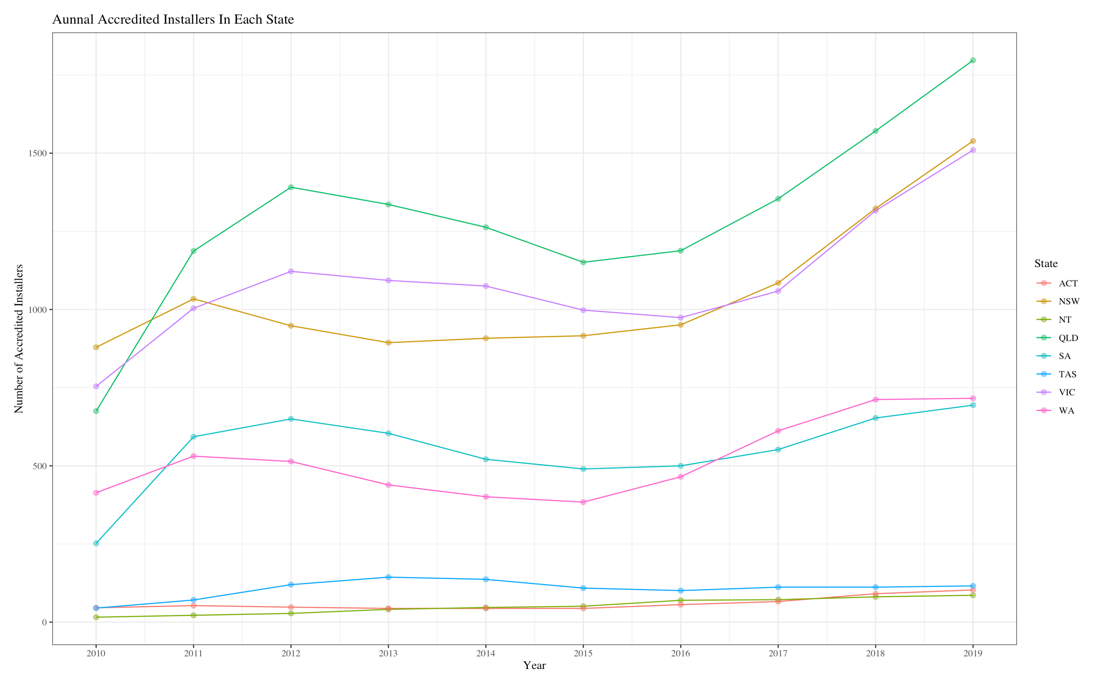


Figure.

Data story 8:

Has there been any changes in choosing energy sources for QLD residents in the last decade?

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

FIGURE 8-1

The line graphs above compare the daily energy consumption per customer in five local government areas(LGAs) by choosing Electricity and Solar Energy over a period of 10 years. It is clear that the solar energy consumption increases while Electricity consumption decreases in each LGA between 2009 and 2018. Central Queensland area is expected to see the most changes in solar energy consumption and almost equivalent to the electricity energy consumed by 2018. Although there is a noticeable decrease in 2010, the ‘Daily Solar Energy Consumption Trend’ showed an upward trend in choosing Solar Energy.

Chart, line chart

Description automatically generated

FIGURE 8-2

According to Australian Energy Council Solar Report, the total installed capacity of rooftop solar PV in Australia at 30 June 2019 is approximately exceed 9.0 GW, which means they are more than 2.1 million installations (2019). Queensland has the highest amount of rooftop solar capacity and the highest number of installations in the country as shown in Figure ‘Annual Accredited installers in Each States’ from Data Story 7. Analysis from the monthly installation data in the above figure 8-2 also allowed us to estimate the amount and capacity of solar PV installed in QLD.

Much of the growth of the installation and solar energy consumption, which may expect that in the future more households will add solar energy systems to their rooftops.

# GitHub Repository Operation

## Introduction to GitHub

GitHub is a web-based graphical interface and the heart of it is Git – an Open Source Distributed Version Control System. GitHub provides the open source version control service and source code management (SCM) functionality of Git and adds many of its own features which allow multiple developers to collaborate and make contributions to the whole community.

In our project, we use the platform to store a variety of files, including R scripts, SQL codes, CSV files and documents, etc. Besides, each member consistently uploads new codes and new files towards the repository.

In order to resolve conflicts as many as possible, we also applied “.gitignore” file to tell Git which files or folders to ignore in this project such as .DS\_Store and database configuration file. The former one is a hidden file from MacOS Operating System and the latter one contains our AWS RDS configuration information which we do not want to be exposed to the public.

## Repository Structure

.

├── DSP\_Data\_Geek\_Solar\_Energy.Rproj

├── README.md

├── code

│ ├── build

│ ├── common

│ ├── config

│ ├── sql

│ └── src

├── data

│ ├── processed\_datasets

│ ├── raw\_datasets

├── doc

├── .gitignore

└── temp

## Git Commands

3.1 Common Git Command

|  |  |
| --- | --- |
| **Command Line** | **Description** |
| git clone <remote\_URL> | Create local copy of an existing remote repository; to get our project you shall use 'git clone <https://github.com/YUECHEN0830/>DSP\_Data\_Geek\_Solar\_Energy.git' |
| git pull | Get the latest version of a repository |
| git status | Check the current state of the local repository compared with remote one |
| git add <file or directory name> | Add a specific file or folder (including files) to the staging area for git |
| git add . | Add all file in the staging area |
| git commit -m <commit message in quotes> | Record the changes made to files to a local repository |
| git checkout <file\_path\_in\_project> | Overwrite the file from remote repository |
| git rm -f <file\_name> | Delete a file |
| git reset <file or directory name> | Remove a file from the staging area |
| git push | Send local commits to the remote repository |

3.2 Branching and Merging

|  |  |
| --- | --- |
| **Command Line** | **Description** |
| git checkout -b <branch\_name> | Create a new branch and switch to it at the same time  (this is shorthand for:  git branch <new\_branch\_name>: create a new branch  git checkout <new\_branch\_name>: switch to the new branch) |
| git branch | List branches |
| git checkout <branch\_name> | Switch branch |
| git branch -d <branch\_name> | Delete branch |
| git push origin <branch\_name> | Push a branch to the remote repository -- here is 'origin'; the command always been used when you want to create a new branch |
| git push -u origin <branch\_name> | Using "-u" option for upstream |
| git merge <branch\_name> | Merge <branch\_name> (remote branch) to your current branch |
| git reset --hard <commit-hash> | Undo a merge |
| git reset --hard HEAD~1 | Undo a merge |
| git revert -m 1 <merge-commit-hash> | Undo a pushed merge |

## GitHub Workflow

The GitHub workflow can be summarised by the “commit-pull-push” stream.

1. Commit

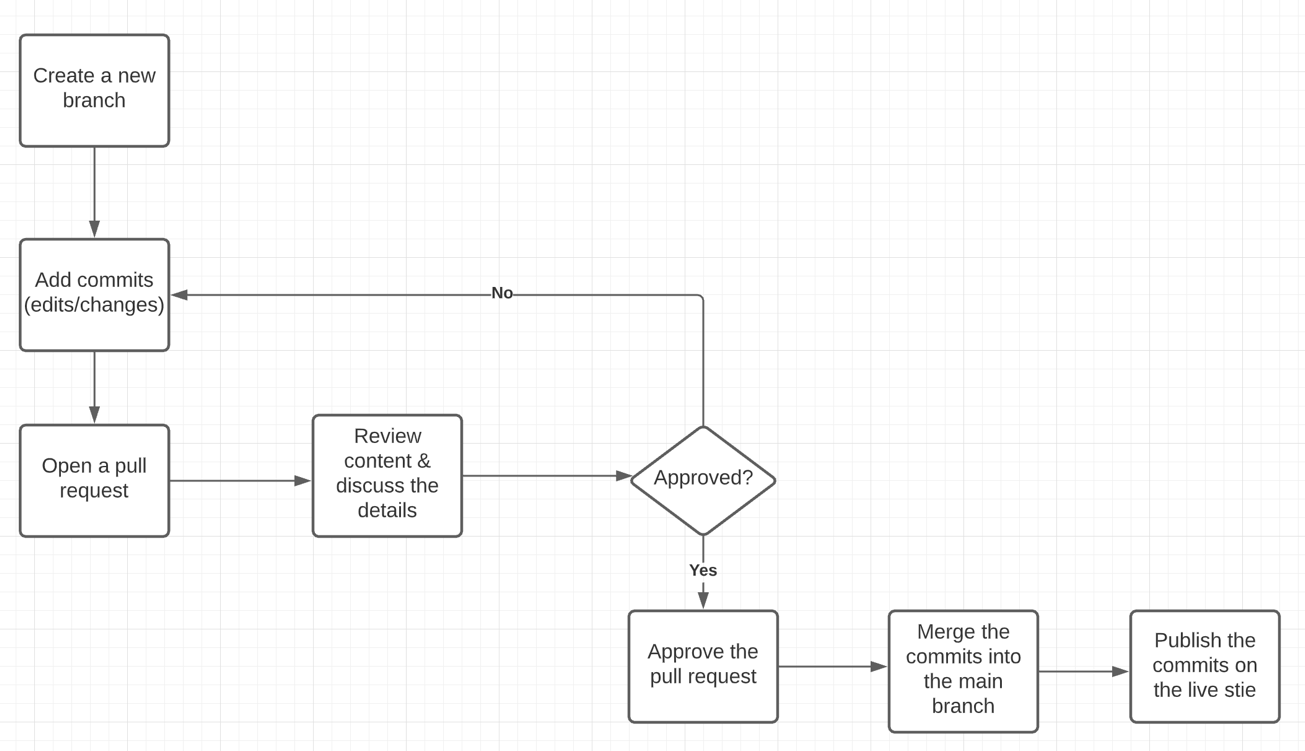
· Once you have saved your files, your need to commit them – this means the changes you have made to files in your repository will be saved as a version of the repo, and your changes are now ready to go up on GitHub

2. Pull

· Before you send your changes to GitHub, you need to pull the latest version from remote repo (i.e. you are completely up to date with the latest version of the files)

3. Push

· Once you are up to date, you can push your changes – at this point in time your local copy and the online copy of the files will be the same.



# Discussion and Conclusion

Because solar data was available from commercial providers, we had to rely on BOM data which could be downloaded complimentary.

A data warehouse compared to a collection of individual csv files has numerous advantages such as data can be updated incrementally and the script runs always on the latest data, version control is easy, dimension data is reusable hence systematic changes can be easily implemented. New queries can be drafted quickly.

For a small project like our assessment the efficiency advantages are not visible. However, on and industrial scale the solution we have developed would be effective because the solution running sql out of an R script, we achieved the following advantages:

* Data frames would not need to be saved and copied, we could assign them to objects.
* With the WHERE clause in sql we could limit the data downloaded. This would reduce data traffic (i.e. costs) and increase speed.
* When we are executing sql we are leveraging computational resources by running operation on the AWS server.
* By running sql for wrangling, we are getting the advantage of the data base engine optimiser this way.
* R was chosen because of personal preference and the packages available for data handling and visualisation are strong
* Python was not chosen because the strength in analysing and modelling (i.e. machine learning) was not needed for the focus of our project.

Data base:

We used 5 different staging data bases for individual learning purpose. In a professional context multiple developer would have shared fewer staging data bases.

For larger databases, we may need to resort to Redshift kind of database engine which offers columnar table for efficient storing and retrieval of data.

Database authentication should be tightened. Rather than just using username and password, proper connection to database should be encrypted using something like IAM from AWS.

GitHub:

We will improve on reading error messages from the terminal and using other git commands to resolve errors. Conflicts management is the one we were struglling with most, as the Git reminds us it cannot auto-merge the local repository from the remote one and we need to fix it first. One of the recommendations is following up the stream of git workflow and pulling latest version from remote repository before committing and pushing local changes. In addition, we will put more comments for each other’s codes and issues, and communicate more on GitHub rather than ‘Teams’.

Conclusion

* A good hands-on exercise for every member in the team to have a taste of a real-life data science project in a commercial environment
* The Ingest – Storage – Process – Visualize is a good framework to structure data science projects
* This project provides us with a good approach to work collaboratively as a team. Each one of us has different skill sets and we can contribute to the project with our experiences in databases and github, for example. At the same time, all of us have opportunities to learn something we don’t normally have to do in a project.
* In real life, we would be getting a lot of data from different departments. Those datasets may be more targeted and specific. It will require continuous innovation in designing the data warehouse or programmes to handle emerging requirements