Research topic

Solar energy generation

Utilization of solar in non-domestic entities

Solar panel imports

Amount of energy export (to Singapore?)

Number of households having panel installed

Amount of electricity generated

Amount of electricity consumed

Solar power stat in AUS

Solar pricing

Research question options

Electricity price throughout day, season, location

Peak price optimization

Maximization on saving on electricity bill

Analysis

Model

Data warehouse setup and justification

Github repository setup

Git branching strategy

Git Feature Branch Workflow (<https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>)

1. git checkout master git fetch origin git reset --hard origin/master
2. git checkout -b new-feature
3. git status git add git commit
4. git push -u origin new-feature

This command pushes new-feature to the central repository (origin), and the -u flag adds it as a remote tracking branch. After setting up the tracking branch, git push can be invoked without any parameters to automatically push the new-feature branch to the central repository. To get feedback on the new feature branch, create a pull request in a repository management solution like [Bitbucket Cloud](https://bitbucket.org/product) or [Bitbucket Server](https://www.atlassian.com/software/bitbucket/server). From there, you can add reviewers and make sure everything is good to go before merging.

### Resolve feedback

1. Now teammates comment and approve the pushed commits. Resolve their comments locally, commit, and push the suggested changes to Bitbucket. Your updates appear in the pull request.

### Merge your pull request

1. Before you merge, you may have to resolve merge conflicts if others have made changes to the repo. When your pull request is approved and conflict-free, you can add your code to the master branch. Merge from the pull request in Bitbucket.

## **Pull requests**

Aside from isolating feature development, branches make it possible to discuss changes via pull requests. Once someone completes a feature, they don’t immediately merge it into master. Instead, they push the feature branch to the central server and file a pull request asking to merge their additions into master. This gives other developers an opportunity to review the changes before they become a part of the main codebase.

Code review is a major benefit of pull requests, but they’re actually designed to be a generic way to talk about code. You can think of pull requests as a discussion dedicated to a particular branch. This means that they can also be used much earlier in the development process. For example, if a developer needs help with a particular feature, all they have to do is file a pull request. Interested parties will be notified automatically, and they’ll be able to see the question right next to the relevant commits.

Once a pull request is accepted, the actual act of publishing a feature is much the same as in the [Centralized Workflow](https://www.atlassian.com/git/tutorials/comparing-workflows). First, you need to make sure your local master is synchronized with the upstream master. Then, you merge the feature branch into master and push the updated master back to the central repository.

Pull requests can be facilitated by product repository management solutions like Bitbucket Cloud or Bitbucket Server. View the Bitbucket Server pull requests documentation for an example.

Timeline:

1st week

Github repos , research questions , AWS account

2nd week

3rd week

4th week

Submit report + code + individual report

Action Points (29/09)

* Marco to enquire BOM for free access to dataset (gridded data of solar)
* Rato to explore Redshift setup
* Rato to sign up to Solcast
* Cecilia to enquire Duhita and Shibani regarding the roles of github admin / contributors / external
* All to continue to research into the topic