Template for Transparent Research

Introduction

This repository is a template for using Rmarkdown, Docker, and GNU Make to create and communicate transparent, reproducible, and error-free research. These tools automate the minutiae of managing dependencies, updating calculations with new data, and transcribing results, leaving you to focus on the important parts of research.

R-Markdown allows you to integrate the results of your analysis directly into your research reports. GNU Make ensures that changes to one part of your code are carried through the entire analysis process, so figures downstream are always re-calculated when the data and code they depend on are updated. Docker ensures that your research runs the same on any machine, allowing others to easily replicate and contribute to your research.

Managing Dependencies with GNU Make

GNU Make automates the process of re-running analysis when "upstream" code or data on which it depends changes. This frees you up to focus on the important parts of research, and gives you confidence that your calculations always incorporate the most up-to-date figures from previous analysis steps.

Make achieves this by expressing the network of file dependencies in a project with a system of rules. As an example, the files in this example repository depend on each other in the following way:

Dependency Chart for the Example Project:

`document.pdf` depends on both `document.rmd` and `cars.csv`, so when either of these files are changed, `document.pdf` should be re-rendered. Similarly, `cars.csv` depends on `/01_example_script.R`, so any time the example script changes, it should be re-run to update `cars.csv`.

Make encodes these dependencies with the following syntax:

```
# document.pdf depends on document.Rmd and ./data/cars.csv.
# When either of the dependencies change, re-render the document
document.pdf: document.Rmd ./data/cars.csv
    R -e "require(rmarkdown); render('document.Rmd')"

# ./data/cars.csv depends on ./code/01_example_script.R.
# When 01_example_script.R changes, rerun it with R.
./data/cars.csv: ./code/01_example_script.R
    cd code; R CMD BATCH --vanilla 01_example_script.R
```

These resources give more information and tips on using make to manage data analysis

1. The Plain Person's Guide to Plain Text Social Science by Kieran Healy
2. Minimal Make by Karl Broman
3. Using GNU Make to Manage the Workflow of Data Analysis Projects by Peter Baker

Replication With Docker

First, build the docker container. This will download and install all of the required dependencies.

```
docker build -t reproduce .
```

Then, depending on your operating system, start the docker container in one of two ways:

Using Linux / Mac / Windows Power Shell:

```
docker run --rm -v ${PWD}:/opt/reproduce reproduce
```

Using Windows Command Line:

```
docker run --rm -v  %cd%:/opt/reproduce reproduce
```

Once the container is up and running, any changes you make to the code or data will be propagated through to the final report.