Data management procedures for reproducible research pipelines

Ivan C. Hanigan

Abstract

This working paper sets out key data management and analysis principles that have been found to be effective for the reproducible synthesis and integration of heterogeneous datasets for analysis and reporting. The draft was last updated April 17, 2016. The most recent version is available on the Github repository: https://github.com/swish-climate-impact-assessment/swish_data_management_procedures.

Contents

1 Introduction

There is a need for developing an evidence-based set of best practice guidelines for data management procedures that support reproducibility in all fields of computational data analysis (Long, 2008; Noble, 2009; Peng, 2015). Reproducibility is the ability to recompute the results of a data analysis with the original data (as distinct from replication which involves analysing independently collected data (Peng, 2011)). The examples drawn together in this report come from experiences and use-cases found from implementing reproducible research pipelines in an eco-social epidemiologic research context. This emerging paradigm mixes environmental and social epidemiology and is inherently concerned with complex systems. To do this work integration of heterogenous data sources, and synthesising new datasets, is required. Then analyses that aim to recognise subtle and complicated patterns in the environmental and social determinants of health must be rigorously and transparently conducted (McMichael, 2013). This document outlines a suite of data management procedures that have been found to effectively assist the development of reproducible research pipelines in this context.

1.1 The 'reproducibility crisis'

It is possible to have analyses that are reproducible with varying degrees of difficulty. A data analysis might be reproducible but require thousands of hours of work. A primary challenge for reproducible data analysis is to make analyses that are *easy* to reproduce.

In essence this requires attention to be turned to the issue of how the data and analytical steps amassed – toward a reality where this is archived and there is a good understanding all round as to how the study were set up and conducted. Different assumptions or different treatment of the data could conceivably lead to different inferences and conclusions being drawn, such as in the example shown by Silberzahn and Uhlmann (2015) in which 29 research teams were given the same dataset but reached a wide variety of conclusions using different methods on the same dataset to answer the same question.

This is partly because of an underlying complexity in the information drawn from complex systems involving multi-causality, and partly because of different assumptions and different backgrounds and viewpoints. A finding that a variable does or does not cause a disease, might be drawn honestly from the same set of data.

1.2 A common (flawed) approach for generating statistical reports

A common approach with inherent flaws that make it error-prone was identified by Scott (2010), and the examples are paraphrased here. First, the data entry, cleaning, preparation and possibly statistical analyses are conducted by 'point-and-click' procedures using software such as Microsoft (MS) Excel. This introduces well-known issues with handling of missing data, poor algorithms and unreliable results (McCullough and Heiser, 2008). In cases where data are imported to a program such as STATA or SPSS, further point-and-click data preparation and statistical analyses often occur. Spreadsheet software such as MS Excel is regularly used to record or format the desired results, and generate figures. Finally, the results (text, tables and figures) from the data analysis system are inserted into a word processor (eg, MS Word) using 'copy-and-paste' procedures (or typed by hand).

Problems with this common, flawed approach according to Scott (2010) are:

- You sit down to finish writing your manuscript. You realize that you need to clarify one result by running an additional analysis. You first re-run the primary analysis. Major problem: the primary results don't match what you have in your paper.
- When you go to your project folder to run the additional analysis, you find multiple data files, multiple analysis files, and multiple results files. You can't remember which ones are pertinent.
- Youv́e just spent the week running your analysis and creating a results report (including tables and figures) to present to your collaborators. You then receive an email from your PI asking you to regenerate the report based on a subset of the original data set and including an additional set of analyses – she would like it by tomorrowś meeting.
- With point and click programs (eg, MS Excel or not using SPSS's log), no way to record/save the steps performed that generated the documented results.
- Common to keep analysis code, results, and reports as separate files and to save various versions of each of these as separate files. After several modifications of one or more of the files involved, becomes unclear which version of the files exactly correspond to the desired analysis and results.
- Every time analyses and/or results change, have to regenerate the results report by hand – very time consuming.
- Very easy for human error to creep into results report (eg, typing in results by hand, copying/pasting the wrong tables/figures).

1.3 Reproducible research reports: A better alternative

It is widely recommended that a better approach is to create Reproducible Research Reports (RRR) (Healy, 2013). This embeds the analysis into the report so that the code to clean and prepare the data or to perform the desired statistical analysis is included in the document that contains the documentation and text of the report. Solutions have been developed that combine both the data analysis code and the descriptive prose that constitutes the publishable report into a compendium (Gentleman and Temple Lang, 2004; Schulte et al., 2012).

Reproducible research reports are written using a scripting language for statistical computing and graphics. The report is made up of ordinary text written in a suitable format that enables the computational process to recognise it as text. An example is the Rmarkdown format which is very similar to text used when authoring word processor documents (http://rmarkdown.rstudio.com). There are also chunks of pure statistical programming code (such as R codes) that perform data manipulations and analyses when the document is 'evaluated'. When the processing stage is run a report document is generated that includes both content as well as the output of any embedded computer code 'chunks' within the document. These are distinguished from the regular text by a special delimiter at their beginning and end. An example using the R language is presented below:

~~~
---
title: "Reproducible report example"
author: "Ivan C. Hanigan"
output: pdf_document
---

# Some exploratory analysis
In this section we do some exploratory analysis of the NMMAPS data for
deaths in Chicago 1987-2000.  The code, messages and intermediary
results are hidden in the resulting report document.
```{r, echo = FALSE, message = FALSE}
## We begin  by reading in the data file:
## If using our own data we would use 'read.csv' or a similar tool to import data to R
# my.data <- read.csv('data/sampledata.csv',header=TRUE)
## for this example use data that are included in the dlnm package
library(dlnm)
# look at the structure of the data
# str(chicagoNMMAPS)
# summary(chicagoNMMAPS)
~~~
We made a simple scatter plot shown below
```{r, echo = FALSE, message = FALSE}
## make some plots. first by day
# with(chicagoNMMAPS, plot(date, cvd, type = "l"))
# we suspect a relationship between temperature and deaths
with(chicagoNMMAPS, plot(temp, cvd, pch = 16, cex = .6))
title(main = "A scatter plot of daily temperatures against deaths")
```
We ran some exploratory models. A Poisson GAM with smooth functions
on temperature and time was compared to a linear fit on temperature.
```{r, echo = FALSE, message = FALSE}
library(mgcv)
fit1 <- gam(cvd ~ s(temp) + s(time), data=chicagoNMMAPS, family = "poisson")
# we can access post-estimation summary statistics
# summary(fit1)
# or just plot the exposure-response function
plot(fit1, select = 1)
title(main = "The exposure-response function estimated using MGCV")
aic1 <- AIC(fit1)
# compare this to a linear term for temperature
aic0 <- AIC(gam(cvd ~ temp + s(time), data=chicagoNMMAPS, family = "poisson"))
# calculate the delta aic
aici <- aic1 - aic0
```
The result can be automatically inserted to the text. This model has
a delta AIC of `r round(aici,1)` (smoothed minus linear term).
```

In this example the Rmarkdown engine is used to construct the final report by ’weaving’ or ’knitting’
```

together the prose and the code. The prose is written in 'markdown' which is a simple way to use 'markup' commands to tell the program to do formatting on the inputs. For example the first 'hash' symbol tells the program that this line should be written in the style of heading-1. The three 'backtick' marks tell the program that the following text should be interpreted as R code. Inside these 'chunks' the 'hash' symbol is interpreted as a comment and the line is not executed.

The resulting report can be viewed at this website: https://github.com/ivanhanigan/ReproducibleResearchPipelineTemplate-results/tree/master/2016-01-30-RRRexample.

If an analysis is published as a reproducible report, readers can have greater confidence in the work that was done, and verify this for themselves if questions remain. However, it has also been recognised that reproducible research can still be wrong and a prevention approach has been recommended that incorporates evidence-based data analysis tools and techniques into such a pipeline (Leek and Peng, 2015a).

## 1.4 Reproducible Research Pipelines (RRP) defined

The techniques of pipelines described here are targeting the integrity of the process of data selection, the robustness and suitableness of the methods used, a commonsense and well-argued selection of health outcomes and environmental or social exposures, and the clarity and transparency of the methods used.

To achieve this, a guiding principle is that analysts should effectively implement 'pipelines' of method steps and tools. Standardised and evidence-based methods based on conventions developed from many data analysts approaching the problems in a similar way should be used, rather than each analyst configuring a pipeline to suit particular individual or domain-specific preferences (Borer et al., 2009; White et al., 2013).

Noble (2009) points out that 'the principles behind organizing and documenting computational experiments are often learned on the fly, and this learning is strongly influenced by personal predilections'. Leek and Peng (2015b) describe this as data analysis being 'taught through an apprenticeship model, and different disciplines develop their own analysis subcultures'. By codifying what an appropriate pipeline would contain, data analysis will be more robust. According to Peng (2015), there should not be a 'lonely data analyst' coming up with their own method. If a researcher conducted an analysis using an evidence-based reproducible research pipeline 'you could at least have a sense that something reasonable was done' Peng (2015) and be confident that you could easily check what had been done if you needed to.

## 1.5 The core components of a pipeline

The core concepts and flow of steps in the method are shown in Figure 1 (after Peng et al. (2006) and Sólymos and Fehér (2008)). In this model there are two main actors: the author and the reader. The author moves from left to right, from initial hypothesis and study design, through data collection and pre-processing, to analysis and reporting. The aim is to conduct all steps of the analysis work in such a way that the key dataset and code script can be sent through the distribution mechanism and the reader can easily move from right to left. Thus the reader can start with the published results and then dig deeper by assessing the analysis code and analytic data to gain full understanding of the methods steps.

Peng et al. (2006) distilled a core set of components for reproducibility from earlier work including that of Schwab et al. (2000). These are:

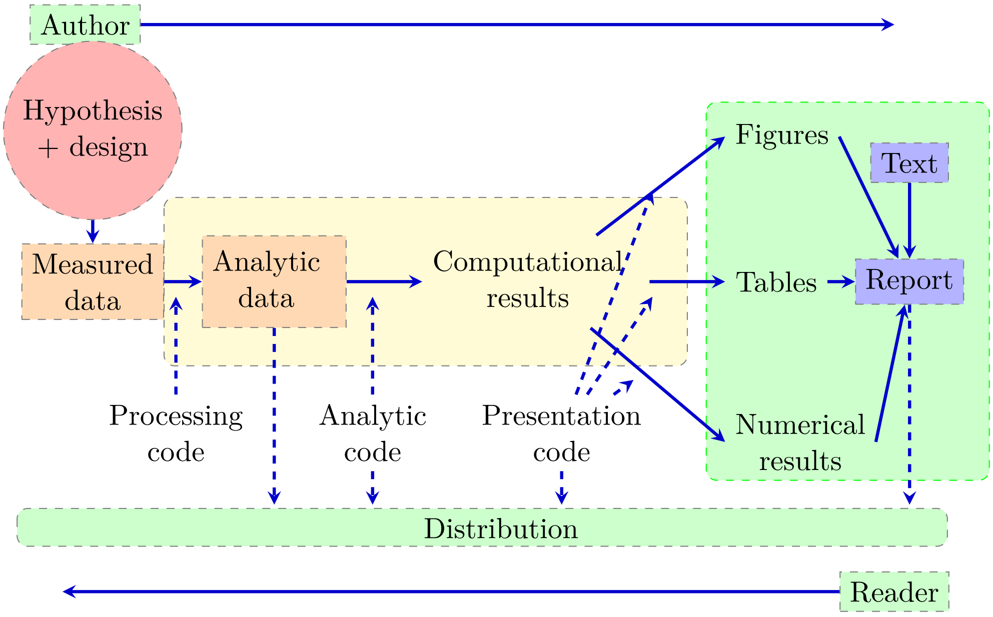

Figure 1: A schematic diagram representing the reproducible research pipeline

- Hypothesis and design
- Data (measurement, pre-processing, analytic)
- Analysis Methods
- Documentation (of all steps)
- Distribution (of the paper, data and code).

### 1.5.1 Hypothesis and design

The first stage of the pipeline is hypothesis generation and study design. In this stage documentation should explain the literature base supporting the study, the decisions made in selection of explanatory factors for inclusion, decisions made such as the experimental unit, observational unit, measurement method, as well as spatial or temporal extent. This information will also be needed for ethical review and approval.

### 1.5.2 Data

The data that were measured should be well managed, however the requirements for accessing the original raw data are less important than for the analytical dataset. Descriptions of how the measured data were transformed into the analytic data should be available. Public data repositories or institutional services such as university libraries should be used to ensure longevity of the data storage.

### 1.5.3 Methods

The software code underlying the principal results needs to be made available. In addition, the computer environment necessary to execute that code should be described adequately to 'deploy' a new computer set-up that can reproduce the computations needed.

### 1.5.4 Documentation

Adequate documentation of the code and data should be available to enable others to repeat the analyses and to conduct other similar ones. This can take the form of metadata, reports, journal papers or even books (Peng and Dominici, 2008). Indeed textbooks on statistical methods can benefit greatly from being accompanied by data and analytical code to enhance their pedagogic functions (Barnett et al., 2014; Barnett and Dobson, 2010).

Misuse may be due either to unintended user misunderstandings about data attributes (no dataset is perfect and self-explanatory, see Michener et al. (1997)) or intentional mis-use for malicious or selfish reasons (for example the misuse of data by Bjorn Lomborg to support the argument that environmental health conditions are actually improving. See Bodnar et al. (2004) for a discussion on Lomborgś misuse of data. There have also been notable examples of mistakes in data analyses used for climate change science. See Cai et al. (2010) for a discussion of one such case. The careful storage and curation of datasets is also critical because data from many studies are lost (Pullin and Salafsky, 2010; Vines et al., 2014).

An important underpinning to reproducible research is the reproducible report. This is the ultimate form of documentation because the information that represents the outputs of the research is written alongside the code that performs the computations that are being described. There has been many recent advances made in terms of tools for reproducible reports such as R markdown and knitr (Xie, 2014).

Metadata should be created and maintained as a priority task at all stages of the data analysis process. An international standard should be preferred over selectively choosing what information one collects and what fieldnames one uses to describe each item of documentation. Ecological Metadata Language (EML) and the Data Documentation Inititative (DDI) are two such standards that offer useful semantic constructs for describing epidemiological data.

Time and effort may be saved by considering metadata requirements at the commencement of a study, rather than trying to recall all the details later. If metadata adheres to a standard schema, it can be used in catalogues to enable fast searching and retrieval, or in machine-to-machine data queries that assist data access and use.

### 1.5.5 Distribution

Distribution or dissemination of the material needs to use a standard method if they are to be used by others. It is not enough just to provide access to the software and data, but also adequate documentation is required to explain and potentially assist downstream users to piece these together.

## 2 Principles for organising projects, datasets and files

For data to be reused in the future, files and related documents need to be carefully managed to allow future users (including the original collector) to find and understand them. A formalised approach to data management should be developed, and following a widely agreed ’standard’ if possible. This example shows the use of the Ecological Metadata Language (EML) concepts of Projects, Datasets and Entities. EML was developed primarily for creating metadata and allows sufficient detail to describe the collection process and record decisions that were made during the creation of the data. In EML the elements of any dataset can be seen as a nested hierarchy at three levels shown in figure 2.

1. The Project level: this is an overarching grouping of data. It might be indicative of the principal investigator or organisation who provided the data, or a programme of research studies (sub-projects).
2. The Dataset level: this is a distinct grouping of data that might be organised around a particular time period or geographical region.
3. The Entity level: This grouping of data includes data files (such as tables in CSV or Excel, shapefiles and raster images) or documents (such as metadata descriptions or related publications).

This conceptual framework can be very useful for the organisation of the work constituting a single pipeline, as well as when working with multiple pipelines within several projects.

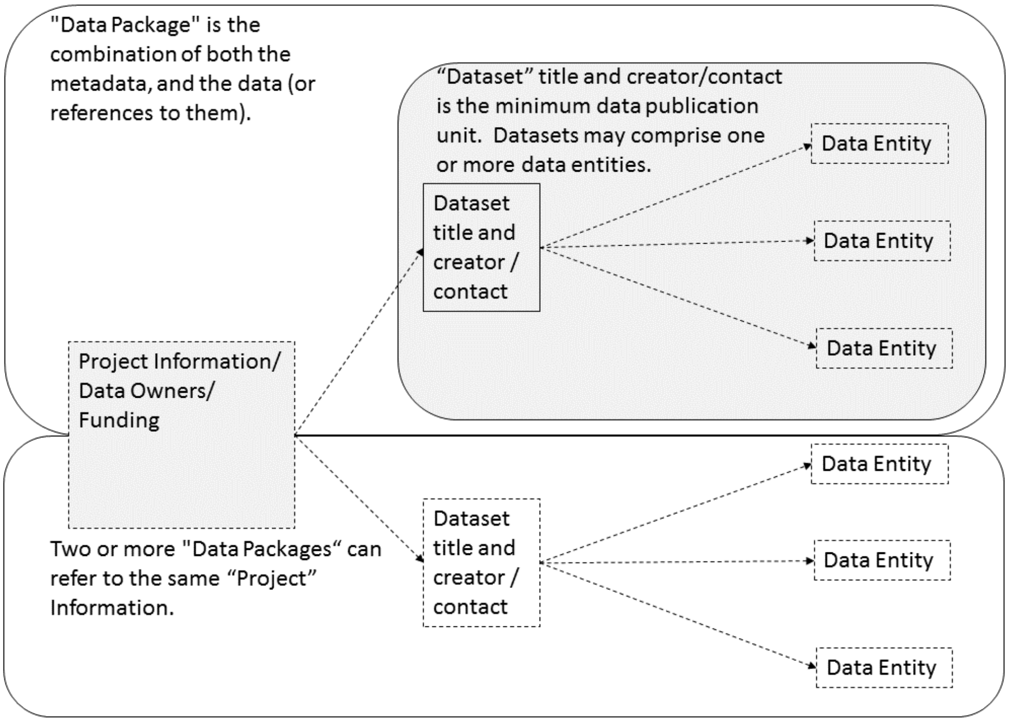
"Data Package" is the combination of both the metadata, and the data (or references to them).
"Dataset" title and creator/contact is the minimum data publication unit. Datasets may comprise one or more data entities.
Data Entity
Dataset title and creator / contact
Data Entity
Data Entity
Project Information/ Data Owners/ Funding
Data Entity
Dataset title and creator / contact
Data Entity
Two or more "Data Packages" can refer to the same "Project" Information.
Data Entity

Figure 2: The EML approach to managing Projects, Datasets and Entities

## 2.1 Procedures when conducting a reproducible research analysis

Having defined above the principle components for a pipeline there are procedural questions about how to go about compiling those. The key steps include:

- Data Management Plans and Data Inventories
- Planning and implementing a pipeline
- Tracking method steps

These three topics will be explored in each of the next three sections of this report.