

Introduction

The aim of this exercise is to highlight some elements of good coding practice. This is a key skill as a Health Data Scientist, especially when writing code that others will need to read, review, edit or run.

To complete this exercise, you need to:

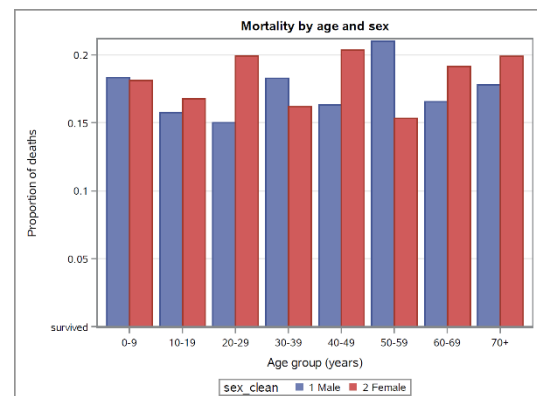
1. Download some SAS data and code from the cloud.
2. Update the code to indicate where the files are stored on your computer.
3. Run the code on your computer to reproduce some very basic results.
4. Answer some questions about the code.
5. Add an additional program to the analysis

This process should illustrate a key principle for analysis: given the raw data and your code, anyone should be able to reproduce your analysis.

The analysis uses two mock datasets: **survey.sas7bdat** and **hospital.sas7bdat**. The code links the two datasets to identify survey participants who had died in hospital during the follow up period. Three key variables (sex, age group and death) are summarised, and the proportion of deaths by sex and age group is plotted.

The details of the analysis are less important than the structure of the code used to (re)produce the results. The code files are presented as an ordered sequence of programs. Each program in the sequence performs a discrete function with a clear input and output.

The entire analysis can be reproduced by running the master program **0.MasterFile.sas**, although first you must edit this file to indicate where you have stored the data and code on your computer.



Remember the principle:

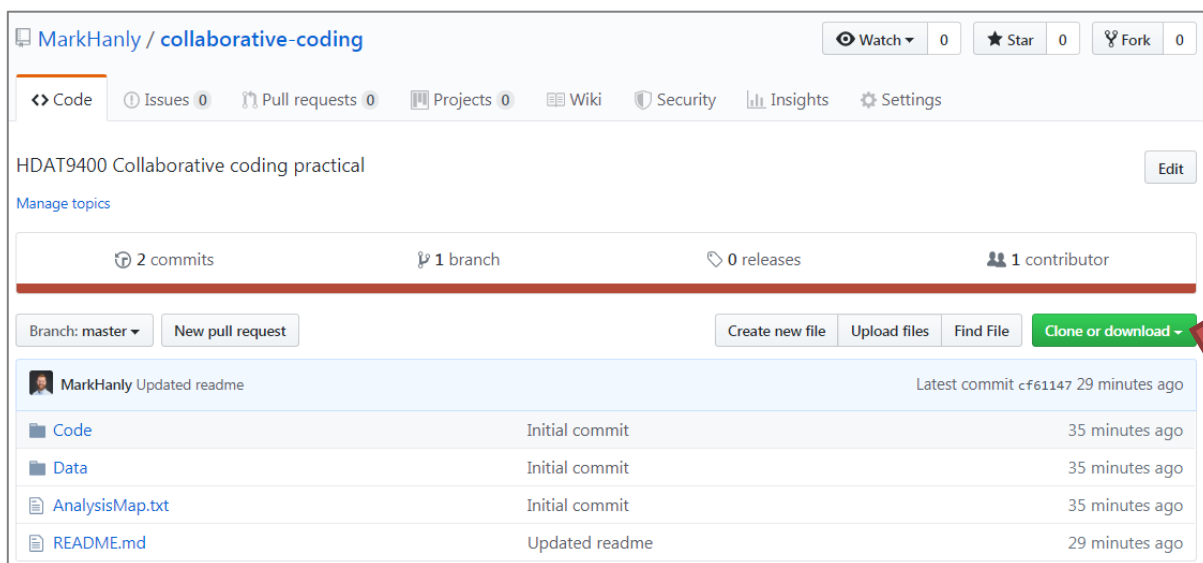
Raw Data + Code = Results

In this exercise you have been provided with the data and code, so reproducing the results should be relatively straightforward.

As a Health Data Scientist your finalised analyses should always follow this rule. This makes sure your work is fully reproducible, and makes it easier for others to read and adapt your code.

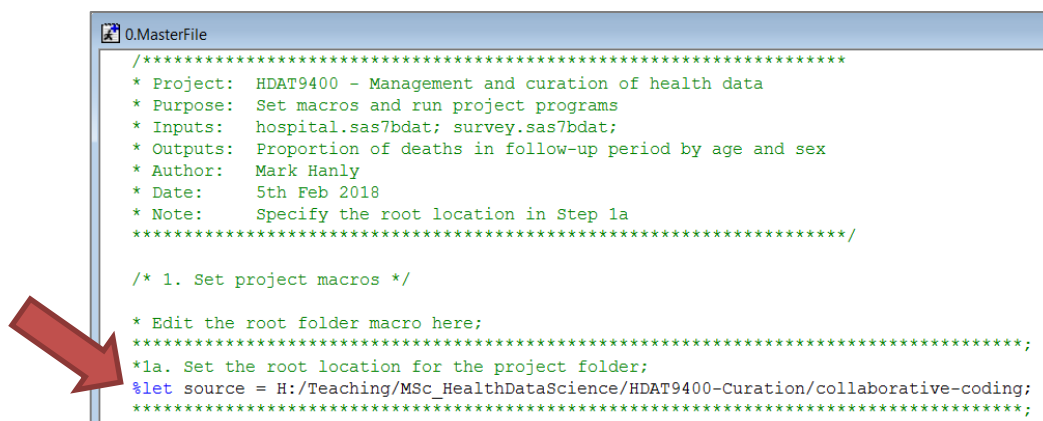
Step 1. Download the code

- Go to <https://github.com/MarkHanly/collaborative-coding>
- Select Clone or download
- If you are comfortable using Git**
 - Clone the repository using your preferred Git UI (e.g. Git Bash/GitHub Desktop)
- If you are NOT comfortable using Git**
 - Select Download ZIP
 - Unzip and save the folder “collaborative-coding”, and all its contents, somewhere on your computer
- Note the location where you have downloaded the code:



Step 2. Specify the location where you have stored the project folder in the program 0.MasterFile.sas

- Open the SAS program file **collaborative-coding/Code/0.MasterFile.sas**
- At the top of the program, edit the line beginning **%let source =** to indicate where you have stored the folder collaborative-coding



Step 3. Examine the program 0.MasterFile.sas to understand what it is intended to do

- a. Once you update the source location, this string is used to identify three new locations: a Code folder, a Data folder and a Results folder

```
*1b. Automatically update the location of the code folder;
%let code = &source/Code;

*1c. Automatically update the location of the data folder;
%let data = &source/Data;

*1d. Automatically update the location of the results folder;
%let output = &source/Results;
```

- b. The next bit of code stores today's date in the macro **date**. This is useful for labelling outputs from the analysis

```
* Today's date automatically set here;
%let date = %sysfunc(putn(%sysfunc(today()), yymmddn8.));
```

- c. The next bit of code uses the **libname** statement to create a new folder in your file system. This new folder is named Results.

```
/* 2. Use the libname statement to create a folder for results */
options dlcreatedir;
libname Results "&source/Results";
```

- d. The next bit of code defines SAS libraries based on the locations


```
/* 3. Define libraries based on the above locations */
libname data "&data";
libname output "&output";
libname source "&source";
```

- e. The final bit of code uses the %INCLUDE statement to run the other SAS program files in the appropriate order

```
*/ 4. Reproduce the analysis */

* Run the project programs in order;
%INCLUDE "&code/1.PrepData.sas";
%INCLUDE "&code/2.SummVars.sas";
%INCLUDE "&code/3.PrintBarChart.sas";
```

Step 4. Run the program 0.MasterFile.sas

- Highlight all the code and select **Submit** 
- The analysis should take about 30 seconds to run. If it runs smoothly, two pdf files with results should be generated and stored in the newly created Results folder.

Step 5. Examine the code and results files to answer the following questions:

- 882 individuals in the analysis were aged _____ years at the time of the survey [HINT: check output *variable_summary.pdf*]
- Roughly what proportion of woman aged ≥ 70 died during the follow up period? _____% [HINT: check output *risk_by_age_sex.pdf*]
- What dataset was used to summarise the variables? [HINT: check code]

- If you reran this analysis tomorrow, what would the histogram pdf file be named? [HINT: check code]

Step 6. Add a new program to the analysis

Create a new program to test whether there was a statistically significant association between mortality and sex and save the results to a pdf file.

Remember to:

- Include header information with clear inputs and outputs
- Comment your code
- Name your program appropriately
- Add your new program to the running sequence in **0.MasterFile.sas**

Hints for Step 6.

- This analysis can be performed using the same dataset that was used to create the histogram
- The SAS code to run the chi squared test of association between mortality (died) and sex (sex_clean) is below:

```
proc freq data=data.mort_by_age_sex;  
tables sex_clean*died /chisq nopercnt nocol;  
run;
```

- To print to pdf, your code should be enclosed between two ODS PDF statements. You can copy the structure from 2.SummVars.sas and/or 3.printBarChart.sas

Step 7. Test your new program

- Rerun the master program **0.MasterFile.sas** to make sure your new program runs smoothly
- Make sure the results of the test of association were saved successfully to the correct destination