Laura Pemberton

THe University of Manchester

GEOG5995: Assignment 2

Programming for Social Scientists: Core Skills

Laura Pemberton

Manchester (9469353) & Leeds (201389071)

Friday 13th December 2019

# Intention

This software was developed to be submitted as part of the assignment requirements for module GEOG5995 (Programming for Social Scientists).

The program itself is intended to be used as a tool for data visualisation. The program is designed to take a large datasets, select a number of variables and display these in various forms, to allow for quicker and easier analysis of a large dataset.

It is also intended that the features of this program be used repetitively across other datasets throughout my career as a PhD student. Therefore, the software is well labelled and documented to allow for future reuse of code.

# Issues during development – how these were resolved

The first issue during this assignment was the consideration of what kind of program to create. The course allowed for reuse of code from the previous assignment for submission of a similar assignment for part 2. However, I felt that it was too difficult to rewrite a pre-existing program to be used again and decided to focus my time and energy on learning a functionality within python, such as data analysis.

A number of bugs were found during the development of the program. Any bugs and issues found during development of the program were resolved using either the text books mentioned below or reference to stack overflow.

# Sources used/references

A number of sources were used to guide the development of this program. This includes: *Python for Data Analysis: Data Wrangling with Pandas, Numpy and IPython* (McKinney, 2018), *Python for Data Science: For Dummies* (Mueller & Massaron, 2015) and *Mastering Python Data Analysis* (Persson & Martins, 2016).

A number of online sources were also used to support the development of the program. This includes a site titled *Pie Chart in MatPlotLib* (Albon, 2017).

# Software Development Process

The program was developed in an agile format, leaning on the CRISP-DM methodology which I have prior experience with (figure 1). Each section of the code was reviewed as each new plot was introduced. The benefits of using a CRISP-DM methodology meant that as I was undertaking the data preparation phase of the project I could work on this iteratively as my data understanding increased. Fortunately, the dataset included data dictionary[[1]](#footnote-1) which was used to guide how the variables were used for analysis.

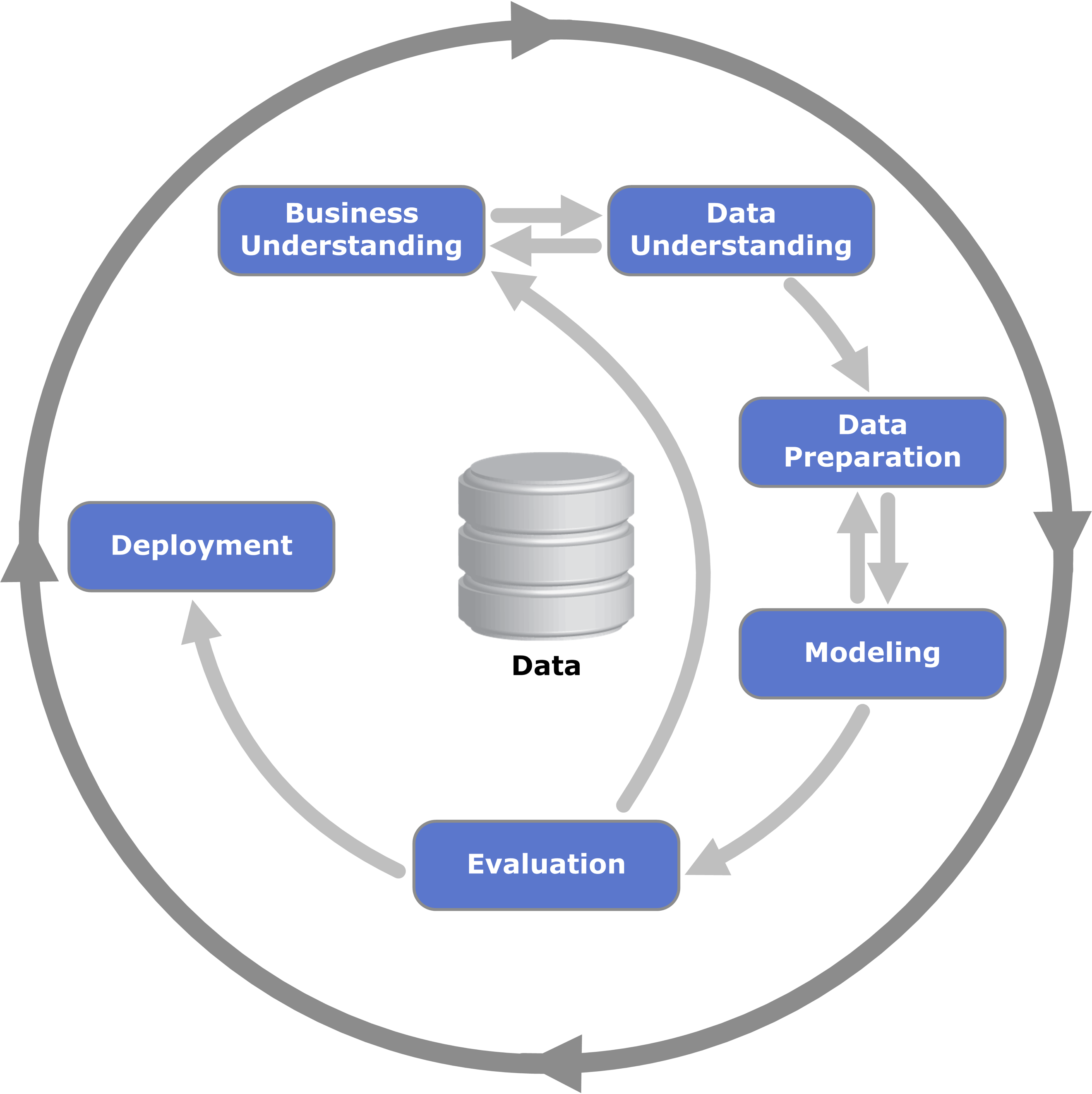


Figure 1: CRISP-DM Data Science Methodology (Vorhies, 2016)

The software was developed to be used as a tool for data visualisation. The program started as a means to meet the assignment goals of the GEOG5995 module, however, has development of the program commenced I felt that the code developed here could be of future benefit to me. Therefore, the software has been designed to be easily picked up again in the future and applied to a different context.

The program started as a means to complete statistical analysis, however as the capabilities of python allow for the development of plots that allow for visualisation of data it evolved to utilise more the matplotlib library within python.

# Known Issues

One known issue with the program is when the plots are run individually the program will produce two figure windows. One window contains the plot and one window is empty. This is due to the ‘.figure()’ line within each plot. Without this line of code, the program would map the plots on top of each other, which was not a desired output.

# Testing

The program was tested continuously through development. The evidence of this has been left within the code in the form of print statements that were used to confirm that the output and variables generated by the program were expected.

# UML diagram

A UML flow diagram was created to go alongside the use of this program. The UML diagram offers a diagrammatic method to explain the functionality and flow of the program more easily.

The UML diagram to go with this program can be seen in Figure 2.

A screenshot of a cell phone

Description automatically generated

Figure 2: UML Flow Diagram created using LucidChart

# Future of the code

The code can easily be developed or evolved in the future to either incorporate a different data set or additional plots. The code is divided into clearly marked sections that can easily be identified if a person wished to target a specific aspect to change.

If the program were to be developed to complete a larger amount of analysis it is assumed that more functions would be required to run the data cleaning aspect of the program.

In addition, a plotting class may be developed in the future that could be used to create and design all the plots created so they reflected more consistency.

1. <https://github.com/LauraIsCool/Assignment_2_Python/blob/master/elsa_sampler_UKDA_Data_Dictionary.rtf> [↑](#footnote-ref-1)