Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

MRP-Kit A 'grammar' of multilevel regression with post-stratification and implementation in R

Lauren Kennedy
Monash University

Mitzi Morris
Columbia University

Jonah Gabry
Columbia University

Rohan Alexander
University of Toronto

Abstract

In this paper we define a 'grammar' for multilevel regression with post-stratification (MRP) and implement this grammar in the R Package **MRPKit**. Our grammar is centered around the following verbs: add, remove, delete, and replace. These act on survey objects. Our grammar, and its implementation, provides a detailed workflow when conducting MRP that will be useful for researchers and in teaching.

Keywords: multilevel regression with post-stratification, R, reproducibility, statistics, political science.

1. Introduction

Multilevel regression with post-stratification (MRP) is a statistical approach in which surveys are related to each other using a statistical model. This is important because known biases in one survey, can be adjusted for by another in a statistically reasonable way. This enables better use of non-representative surveys, additional information, and propagation of uncertainty. However it can be difficult to use MRP due to this need to related two different datasets. This package defines a grammar, or list of underlying rules, of MRP and then describes an R package, MRPKit, that implements this grammar.

At its core, MRP is a mapping between a survey object and a population object. It is from this mapping that the power of MRP exists, but it also establishing this mapping that is the difficult part of implementing MRP models. Making this implementation easier and more reproducible is important as interest in, and the use of, MRP increases. It can be difficult even for those experienced with MRP to ensure there are no mistakes in this mapping and easing this is an important contribution to enhancing the reproducibility of MRP analysis.

We first define a grammar of MRP, which we define as the underlying rules and principles that

are common to every analysis based on MRP. This grammar is based around the following verbs: 'add', 'remove', 'delete', and 'replace'. These verbs are applied to a survey object, a post-stratification object, and survey_map object. These three objects come together to create a mapping object, which is what processes such as regression act on. Finally, common diagnostics and graphs are implemented. In this way, our grammar and package implement an entire statistical workflow for conducting MRP.

The survey object would typically be a regular survey, such as a political poll of 1,000 respondents, but it could also be a larger survey, such as the Canadian Election Survey, or similar. The post-stratification object would typically be a larger survey, such as, in the case of the US, the (INSERT THE USUAL ONE), or a census.

Our grammar and package complements existing packages such as survey (Lumley 2020, Lumley (2004), Lumley (2010)), and DeclareDesign (Blair, Cooper, Coppock, and Humphreys 2019). These packages are focused on the designing, implementing, and simulating from, surveys. Instead, ours is focused on what is needed for MRP.

The remainder of this paper is structured as follows: Section 2 reviews similar packages and contributions and places ours within that context. Section 3 discusses the grammar and the core aspects of MRP as implemented in MRPKit. Section 4 discusses some of the implementation issues and technical notes related to the decisions that were made. Section 5 provides two examples of the package in use, one using SOMETHING, and the other using SOMETHING ELSE. Finally, Section 6 provides a summary discussion, some cautions and weaknesses, as well as notes about next steps.

2. Review of the other packages

The most common alternative at the moment to this package is for users to do all aspects themselves. While there is nothing inherently difficult about MRP, the implementation can be difficult. In particular, preparing and matching different levels between surveys can be time consuming and potentially introduce undocumented errors.

The survey package (Lumley 2020, Lumley (2004), Lumley (2010)) ...

The DeclareDesign package (Blair *et al.* 2019)...

There is also an existing package called MRP available here: `https://github.com/gelman/mrp`. However it does not appear to have been updated in some time.

The main alternative approach is for MRP to be conducted on a case-specific basis. That is a researcher interested in MRP estimates, writes the code needed to obtain, clean, prepare, analyze and ultimately interpret the estimates. Again, while there is nothing inherently wrong with this approach, successfully conducting MRP requires dealing with a large number of small issues. Each of these is small in their own right, but getting them wrong can have large effects that are potentially unnoticed on the estimates.

3. Components and grammar

The MRPKit package has the following key components: survey objects, which for most users will be a collection of two surveys where one is larger than the other; a survey map, that relates the survey objects and then an MRP object, which is created once the survey map is

applied to the survey objects. Regression acts on the MRP object. These objects are subject to the following verbs: 'add', 'delete', 'new', and 'replace'. For instance: SurveyMap$add, SurveyOb$delete, SurveyMap$new, and SurveyMap$replace.

Survey objects

Survey objects are the surveys that MRP will bring together. A user can add a survey object by providing a CSV file location. The user then needs to identify the data types of each column. Typically, there will need to be two survey objects, where one will be the survey that is of interest for its response variables, such as political opinion, and another will be a survey that is to be used for post-stratification.

New is when you have a new survey object and you are bringing it in. e.g.

What is the difference between add and new? What is replace doing? URGH, I just need to look at the repo.

Survey map

A `SurveyMap` object holds the mapping between a set of items in a survey and a population dataset. The label is the item label in each dataset and the values is a list of all possible values. The values for the survey and population must be aligned, i.e., the lists must have the same number of elements and the values at index i in each list are equivalent. If there is a meaningful ordering over the values, they should be listed in that order, either descending or ascending.

One of the fundamental issues when conducting MRP is to ensure that

MRP object

An MRP object contains survey objects, and a survey map. At the point at which the survey objects are put into the MRP object they become immutable but the survey map object is.

The MRP object outputs an analysis. There would be a MRP model fit class that would

Many objects on the same class. There are two

4. Implementation and technical notes

5. Vignette

Simulated data

Using CCES data

Using your own data

6. Summary and discussion

Next steps and cautions

Acknowledgments

References

Blair G, Cooper J, Coppock A, Humphreys M (2019). "Declaring and Diagnosing Research Designs." *American Political Science Review*, **113**, 838–859. URL https://declaredesign.org/paper.pdf.

Lumley T (2004). "Analysis of Complex Survey Samples." *Journal of Statistical Software*, **9**(1), 1–19. R package verson 2.2.

Lumley T (2010). *Complex Surveys: A Guide to Analysis Using R: A Guide to Analysis Using R*. John Wiley and Sons.

Lumley T (2020). "survey: analysis of complex survey samples." R package version 4.0.

Affiliation:

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd