

Report

Drunken Master 2

Due 5 November 2018

Contents

Introduction	1
R	2
**The function <i>lmBoot</i> *	2
<i>Changes made to lmBoot</i>	2
<i>Example analysis using lmBoot</i>	2
SAS	3
**The program <i>newboot2</i> *	3
<i>Changes made to newboot2</i>	3
<i>Example analysis using newboot2</i>	4
References	5
Appendix	6
A.### Code to measure program runtime in SAS	6

Introduction

Bootstrapping is usually implimented using simple Monte Carlo simulations whereby a certain calculation is repeated a large number of times with random sampling (ref?). Repeating calculations a large number of times, say 1,000,000 times, can become slow to compute. Therefore, the use of efficient and fast code is essential.

This project aimed to improve and produce two fast and efficient bootstrap functions using R 3.5.1 (R, 2018) and SAS 9.4 (SAS Institute, Cary NC).

R

****The function lmBoot***

Changes made to lmBoot

Example analysis using lmBoot

#Include plots and interpretation

SAS

****The program newboot2***

The macro program “newboot2” uses bootstrap sampling to calculate estimates for the means and confidence intervals of the slope and intercept parameters produced by a linear regression.

It takes in four arguments:

- NumberOfLoops: the number of bootstrap iterations.
- DataSet: A SAS dataset containing the response and covariate.
- XVariable: The covariate for our regression model (gen. continuous numeric).
- YVariable: The response variable for our regression model (gen. continuous numeric).

The program then outputs:

- ResultHolder: A SAS dataset with the number of rows equivalent to the NumberOfLoops argument and two columns; RandomIntercept and RandomSlope.
- An RTF file containing 95% confidence intervals for the mean, the mean estimate for each parameter and plots of the distributions of the bootstrap parameters.

The function makes use of

- MACRO statements to create a flexible program with input arguments.
- PROC SURVEYSELECT which allows the use of random sampling to generate random samples from a selected or inputted dataset.
- PROC REG to perform a linear regression.

Changes made to newboot2

1. The %do% loop was first removed and the following simple code was added to PROC SURVEYSELECT:
samprate = 1
outhits
rep = %NumberOfLoops

which ensures that NumberOfLoops samples of the same size as the original data set are produced recorded.

2. A linear regression using PROC REG was improved by introducing the by-variable REPLICATE. This variable is automatically produced from PROC SURVEYSELECT to keep track of each new bootstrap sample, then ensuring that the linear regression is run on each sample. Thus, only the Result Holder Dataset was necessary, and there was no need to generate the Temp Dataset.
3. The SASFILE statement was included to upload the dataset to RAM rather than the hard drive before any sampling was carried out so that the dataset does not have to be read in every time a resample needs to be done.

(4. Replaced noprint and ODS listing close - still working on this)

Table ### displays the runtime (in seconds) for 100 loops. The code used to measure the run time of the newboot2 program can be found in Appendix A.### (H, 2012).

Initial	Removed %do% Loop	Loading Dataset to RAM
37.5370	0.2190	0.1880

- Initial runtime for 10 loops: 3.3390s

Example analysis using newboot2

#Include plots and interpretation

References

Cassell, D. (2018). Don't Be Loopy: Re-Sampling and Simulation the SAS® Way. [online]. Available at: <http://www2.sas.com/proceedings/forum2007/183-2007.pdf> [Accessed 26 Oct. 2018].

H, J. (2012). To calculate SAS program run time. [online]. Available at: <http://sashowto.blogspot.com/2012/06/to-calculate-sas-program-run-time.html> [Accessed 26 Oct. 2018].

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Available at: <https://www.R-project.org/>.

SAS 9.4, SAS Institute Inc., Cary, NC, USA.

SAS Institute Inc. 2004. Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference. Cary, NC: SAS Institute Inc.

Appendix

A.### Code to measure program runtime in SAS

```
%let __sdtm=%sysfunc(datetime());
```

Program of interest to be timed

```
%let __edtm=%sysfunc(datetime());
```

```
%let __runtm=%sysfunc(putn(&__edtm - &__sdtm, 12.4));
```

```
%put It took &__runtm seconds to run the program;
```