rwolf2 Implementation and Flexible Syntax

Damian Clarke*

July 8, 2021

Abstract

This is a short document that details the updated more flexible syntax of the rwolf2 Stata command. It lays out the syntax, shows how this syntax is used to replicate the previous implementations in the original rwolf command, and describes cases where it allows for a much easier implementation of the Romano & Wolf multiple hypothesis correction in Stata.

Introduction

rwolf2 is a new flexible implementation of the Romano and Wolf (2005a,b, 2016) multiple hypothesis correction written for Stata. A previous implementation of this procedure is available as rwolf (Clarke, 2016). Both are available for download from Stata's SSC using:

```
ssc install ado
```

where ado should be replaced with rwolf or rwolf2. Both return the same results (provided the same seed is set) when the same multiple hypothesis correction is implemented. However rwolf2 has a much more flexible syntax (in line with Stata's sureg syntax) implying that any type of estimation procedures can be used, and these estimation procedures can be mixed within a single multiple correction procedure, provided that they return a point estimate and standard error. This includes standard Stata estimation procedures such as reg, xtreg, areg and so forth, but also other user-written libraries such as rdrobust (Calonico et al., 2017), ivreg2 (Baum et al., 2002) and reghdfe (Correia, 2014) among others.

In this document I provide a brief description of the new more flexible syntax, document its implementation, and show that it replicates exactly the results of rwolf from examples shown in Clarke, Romano,

*Associate Professor of Economics, University of Chile. Contact email: dclarke@fen.uchile.cl. I thank David McKenzie for suggestions on the alternative syntax implemented here.

and Wolf (2020). I note the things which can now be implemented very easily (in one line) which were previously complicated to implement in `rwolf`.

New Syntax

The syntax in `rwolf2` allows for the concatenation of all multiple tests to be implemented as a series of estimation commands, and requries that an option indicates the indepedent variables of interest (for which the correction should be implemented). Specifically, the syntax is:

```
rwolf2 (model1 depvar1 varlist1 [if] [in] [weight], [options1])
       (model2 depvar2 varlist2 [if] [in] [weight], [options2])
       ...
       (modelN depvarN varlistN [if] [in] [weight], [optionsN]),
indepvars(vars1, vars2, ..., varsN) [other_options]
```

Above model1, model2, …, modelN refer to estimation commands, depvar1, depvar2, …, depvarN refer to dependent variables and varlist1, varlist2, …, varlistN refer to independent variables in each model. These independent variables can be syntax structures such as `(endog=iv) control` in the case that the model is `ivregress`. The indepvars() option specifies lists of independent variables in each model for which the Romano-Wolf correction is required. The help file of the command provides a full description of its syntax and implementation. As an example, a simple implementation of the command may be:

```
rwolf2 (reg y1 Treat1 Treat2, robust)
       (reg y2 Treat1 Treat2, robust),
indepvars(Treat1 Treat2, Treat1 Treat2)
```

where corrections for four tests are made (two treatment variables with two regression outcomes). However, more complicated cases can be easily incorporated such as:

```
rwolf2 (regress y1 Treat1 control1a control1b, robust)
       (ivregress 2sls y2 (Treat1=Assignment1) control2, robust),
indepvars(Treat1, Treat1)
```

In this second option, a correction is made for two multiple hypothesis tests (on the variable Treat1 in both cases), but note that one model is estimated by IV, while the other is estimated by linear regression. Any

other combinations can be easily implemented. A complicated example may include regression discontinuity models, IV models high-dimensional fixed-effect models, and so forth. For example:

```
rwolf2 (regress y1 Treat1 control1a control1b, robust)
       (ivreg2 y2 (Treat1=Assignment1) control2, robust),
       (rdrobust y3 runningvar, covs(control3)),
       (reghdfe y4 Treat1 Treat2, absorb(state year)),
indepvars(Treat1, Treat1, RD_Estimate, Treat1 Treat2)
```

where an OLS, IV, regression discontinuity and high-dimensional fixed effect models are all implemented, and multiple hypothesis corrections are implemented for 5 variables of interest (specified in the indepvars option). This multiple hypothesis correction was previously complicated to implement with `rwolf`, but is very easy with `rwolf2`.

Changes

The `rwolf2` command makes it much easier than in `rwolf` to:

- Include any estimation commands for underlying models, as well as different estimation commands between models, including user written commands like reghdfe, ivreg2, rdrobust and so forth.
- Adjust for multiple and differing treatment variables within and across models
- Include different controls in each model (or estimate with different weighting schemes, estimation options, sub-samples, etc.)

Consistenty Across Versions

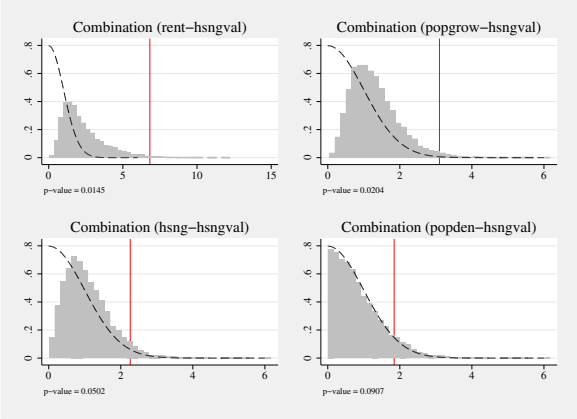
In cases where `rwolf2` and `rwolf` can be implemented to conduct the same multiple hypothesis test, they will return exactly the same results, provided that the same seed is set. For example, see below for the IV regression example from Clarke et al. (2020). The first implementation uses `rwolf2` with the new syntax, while the second version implements the same tests with the old syntax of `rwolf`. The output is the same in each case.

```
. use "http://www.stata-press.com/data/r13/hsng", clear
(1980 Census housing data)

. #delimit ;
delimiter now ;
. rwolf2 (ivregress 2sls rent (hsngval=faminc i.region) pcturban)
>        (ivregress 2sls popden (hsngval=faminc i.region) pcturban)
>        (ivregress 2sls popgrow (hsngval=faminc i.region) pcturban)
>        (ivregress 2sls hsng (hsngval=faminc i.region) pcturban), graph
> indepvars(hsngval, hsngval, hsngval, hsngval) reps(10000) seed(2) nodots;
Bootstrap replications (10000). This may take some time.


Romano-Wolf step-down adjusted p-values
Number of resamples: 10000


-------------------------------------------------------------------------------
                | Model p-value    Resample p-value    Romano-Wolf p-value
----------------+--------------------------------------------------------------
rent            |
        hsngval |     0.0000             0.0169              0.0169
----------------+--------------------------------------------------------------
popden          |
        hsngval |     0.0654             0.0845              0.0845
----------------+--------------------------------------------------------------
popgrow         |
        hsngval |     0.0019             0.0107              0.0169
----------------+--------------------------------------------------------------
hsng            |
        hsngval |     0.0236             0.0106              0.0460
-------------------------------------------------------------------------------



. rwolf rent popden popgrow hsng, indepvar(hsngval) method(ivregress)
> iv(faminc i.region) reps(10000) graph nodots controls(pcturban) seed(2);
Bootstrap replications (10000). This may take some time.




Romano-Wolf step-down adjusted p-values


Independent variable:  hsngval
Outcome variables:   rent popden popgrow hsng
```

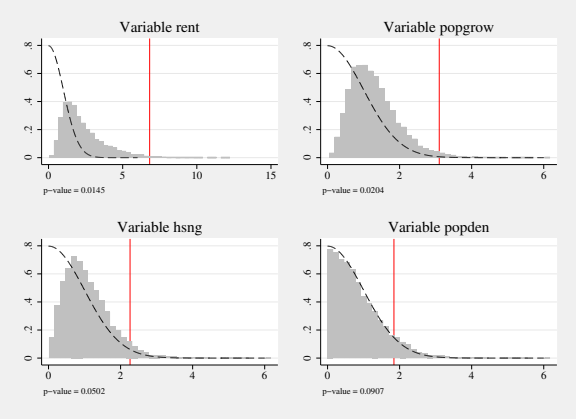
```
Number of resamples: 10000

------------------------------------------------------------------------------
   Outcome Variable | Model p-value    Resample p-value    Romano-Wolf p-value
--------------------+---------------------------------------------------------
               rent |    0.0000            0.0169              0.0169
             popden |    0.0654            0.0845              0.0845
            popgrow |    0.0019            0.0107              0.0169
               hsng |    0.0236            0.0106              0.0460
------------------------------------------------------------------------------
```

The only change is in terms of graphical output if the `graph` option is used. While they are identical in terms of the displayed results, the new `rwolf2` output renames null distribution graphs more clearly for cases where dependent variables may be constant across models, so that both dependent and independent variable names are uesd to caption graphs.

Figure 1: Graphical Output from `rwolf2` and `rwolf`



(a) Output from `rwolf2`



(b) Output from `rwolf`

References

C. F. Baum, M. E. Schaffer, and S. Stillman. IVREG2: Stata module for extended instrumental variables/2SLS and GMM estimation. Statistical Software Components, Boston College Department of Economics, Apr. 2002.

S. Calonico, M. D. Cattaneo, M. H. Farrell, and R. Titiunik. rdrobust: Software for regression-discontinuity designs. *Stata Journal*, 17(2):372–404(33), 2017.

D. Clarke. RWOLF: Stata module to calculate Romano-Wolf stepdown p-values for multiple hypothesis testing. Statistical Software Components, Boston College Department of Economics, Dec. 2016. URL `https://ideas.repec.org/c/boc/bocode/s458276.html`.

D. Clarke, J. P. Romano, and M. Wolf. The romano–wolf multiple-hypothesis correction in stata. *The Stata Journal*, 20(4):812–843, 2020. doi: 10.1177/1536867X20976314.

S. Correia. REGHDFE: Stata module to perform linear or instrumental-variable regression absorbing any number of high-dimensional fixed effects. Statistical Software Components, Boston College Department of Economics, July 2014. URL `https://ideas.repec.org/c/boc/bocode/s457874.html`.

J. P. Romano and M. Wolf. Stepwise Multiple Testing as Formalized Data Snooping. *Econometrica*, 73(4): 1237–1282, 2005a.

J. P. Romano and M. Wolf. Exact and Approximate Stepdown Methods for Multiple Hypothesis Testing. *Journal of the American Statistical Association*, 100(469):94–108, 2005b.

J. P. Romano and M. Wolf. Efficient computation of adjusted p-values for resampling-based stepdown multiple testing. *Statistics and Probability Letters*, 113:38–40, 2016.