Instructions

This code implements the WLS and ALS estimators of Romano and Wolf (2017). Basic familiarity with the software `R` is assumed. To load the code (and associated data sets) into your `R` working session place the file `wls.als.RData` into your working directory and then use the command

```
> load("wls.als.RData")
```

1 Fit your model by OLS

You need to use the function `lm` with the options `x = T` and `y = T`. To give an example, say you have a data frame `wages.fr` that contains log-wages (`log.wages`), years of schooling (`schooling`), and years of experience (`experience`). Then you would fit a regression model as follows:

```
> fit.wages.ols = lm(log.wages ~ schooling + experience, x = T, y = T, data = wages.fr)
```

2 Fit your model by WLS and ALS

Use the function `lm.wls`. Its main input is a model fitted by OLS using the function `lm` with the options `x = T` and `y = T`. The function is set up as follows:

```
> lm.wls.als
function(fit.ols, spec.wool = FALSE, delta = 0.1, alpha.als = 0.1, frac.up = 0.05){
  # INPUTS:
  #
  # fit.ols = fitted univariate regression model using OLS; must be lm object
  # spec.wool = binary variable whether to use the Wooldridge specification (3.7)
  #             or 'our' specification (3.4); default is (3.4)
  #             Wooldridge specification is based too on absolute values of regressors
  # delta = the constant for lower truncation in (3.6); the default is 0.1
  # alpha.als = the significance level for the ALS pretest; the default is 0.1
  # frac.up = upper bound on the fraction of the 'truncated' squared residuals;
  #           if exceeded, 'blow' up the model; the default is 0.05
  #
  # OUTPUTS:
  #
  # fit.wls = fitted model using WLS
  # fit.als = fitted model using ALS
```

In a typical application, one only needs to supply the first argument (that is, the model fitted by OLS) and then choose the specification of the skedastic function via the second argument: the default is 'our' specification (3.4); if the Wooldridge specification (3.7) is desired instead, make the second input `spec.wool = TRUE` instead.[1] So, in the example above, if you want to use specification (3.4) for the skedastic function, use the command

```
> fit.wages.wls.als = lm.wls.als(fit.wages.ols)
```

and otherwise use the command

```
> fit.wages.wls.als = lm.wls.als(fit.wages.ols, spec.wool = TRUE)
```

The two outputs of the function, as elements of a returned list, are the two fitted models: one by WLS (`fit.wls`) and one by ALS (`fit.als`). You can 'access' each fitted model with the usual dollar-sign syntax:

```
> fit.wages.wls = fit.wages.wls.als$fit.wls

> fit.wages.als = fit.wages.wls.als$fit.als
```

Remark 2.1. Strictly speaking, specification (3.4) of the skedastic function cannot be used when some of the regressors can take on the value zero, such as dummy variables, or can take on valuses very close to zero, such as stock returns, because of taking logs. Therefore, the function `lm.wls.als` automatically checks for such occurences and adds a small number `delta` (whose default value is 0.1) to a regressor after taking absolute values, if necessary. No action by the user is required.

Remark 2.2. In the estimation of the skedastic function, whether based on specification (3.4) or specification (3.7), the squared OLS residuals are truncated at a lower value δ^2, if necessary, before taking logs. In some applications, depending on the units used, the values of the squared OLS residuals can be very close to zero, in which case most (or even all) squared OLS residuals get truncated, which is undesirable. Therefore, the function `lm.wls.als` automatically checks for such occurences and 'blows up' the regression for the estimation of the skedastic function until the fraction of squared OLS residuals that are below δ^2 is no larger than `frac.up` (whose default value is 0.05), if necessary. No action by the user is required.

3 Inference

In the presence of conditional heteroskedasticity, inference needs to based on robust standard errors, whether the model has been fitted by OLS, WLS, or ALS; in particular, we recommend the use of HC3 standard errors. For the analysis in `R`, one needs to install (once) the two packages `lmtest` and `sandwich`; these packages automatically are 'required' by the function `lm.wls.als` and, therefore, do not need to be activated manually at the start of an `R` session by use of the `library()` command. Then the software output can be obtained as follows:

```
> coeftest(fit.wages.wls, vcov = vcovHC(fit.wages.wls, "HC3"))
```

[1] We also take absolute values of the regressors in the Wooldridge specification in our code, analogously to specification (3.4).

4 An example

This cross-sectional data set from 1970 contains 506 observations from communities in the Boston area. The aim is to explain the median housing price in a community by means of the level of air pollution, the average number of rooms per house, and some other community characteristics. The variables (one response and four explanatory) used in the regression model under consideration are as follows:

$\log(price)$:	log of median housing price (in US$)
$\log(nox)$:	log of nitrogen oxide in the air (in parts per million)
$\log(dist)$:	log of weighted distance from five employment centers (in miles)
rooms:	average number of rooms per house
stratio:	average student-teacher ratio

The model follows an example from Wooldridge (2012, p. 132). The results from the OLS estimation are presented in the upper part of Table 1. All the estimated coefficients have the expected sign and are significant at the 1% level.

The lower part of Table 1 presents the WLS results, which are based on specification (3.4) of the skedastic function. The WLS estimates do not substantially differ from the OLS estimates. However, the HC standard errors are always smaller for WLS compared to OLS and generally noticeably so, with the ratios ranging from 0.90 to 0.52. As for OLS, all estimated coefficients are significant at the 1% level. But the corresponding confidence intervals based on WLS are shorter compared to OLS due to the smaller standard errors, which results in more informative inference. For example, a 95% confidence interval for the coefficient on *rooms* is given by $[0.299, 0.351]$ based on WLS and by $[0.258, 0.356]$ based on OLS. Needless to say, the smaller standard errors for WLS compared to OLS would also result in more powerful hypothesis tests (that is, smaller corresponding p-values) concerning the various regression coefficients.

To check your correct use of the code, the data set and the fitted models are provided as follows:

- `hprice.data`: the data set
- `fit.hprice.ols`: the model fitted by OLS
- `fit.hprice.wls.als`: the model fitted by WLS and ALS, using specification (3.4)

References

Romano, J. P. and Wolf, M. (2017). Resurrecting weighted least squares. *Journal of Econometrics*, 197:1–19.

Wooldridge, J. M. (2012). *Introductory Econometrics*. South-Western, Mason, Ohio, fifth edition.

Response Variable: log(*price*)				
		OLS		
Coefficient	Estimate	SE (HC)	t-stat	
constant	11.084	0.383	28.98	
log(*nox*)	−0.954	0.128	−7.44	
log(*dist*)	−0.134	0.054	−2.48	
rooms	0.255	0.025	10.10	
stratio	−0.052	0.005	−11.26	
		WLS		
Coefficient	Estimate	SE (HC)	t-stat	WLS/OLS
constant	9.958	0.248	40.23	0.65
log(*nox*)	−0.778	0.094	−8.27	0.73
log(*dist*)	−0.139	0.034	−4.25	0.60
rooms	0.325	0.013	24.57	0.52
stratio	−0.030	0.004	−7.26	0.90

Table 1: OLS and WLS results for the housing prices data set. WLS/OLS denotes the ratio of the WLS-HC standard error to the OLS-HC standard error. For this data set, ALS coincides with WLS.