

Documentation

Authors:

Chung Shun Man (20-621-587)

Matthias Lukosch (20-601-050)

OLS Program Description

The function 'OLSmodel.m' can be used to perform ordinary least square (OLS) regressions. It is also capable of conducting different kinds of diagnostics to indicate whether certain OLS assumptions are violated.

Usage

```
OLSmodel(y, covariates, 'h_robust', true, 'h_diagnostic', true,
        'resid_diagnostic', true, 'spec_diagnostic', true);
```

Input Arguments

Table 1: *Details Input Arguments.*

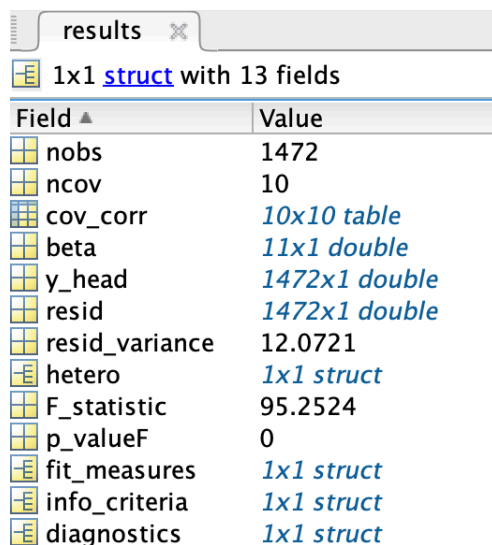
Argument	Description	Type	Data Type
y	Dependent variable	required	table
covariates	Independent variables	required	table
h_robust	Specifies whether to use heteroskedasticity robust standard errors (default = true)	optional	logical
h_diagnostic	Specifies whether to perform heteroskedasticity diagnostics (visual & formal test) (default = false)	optional	logical
resid_diagnostic	Specifies whether to perform further OLS residuals diagnostics (visual & formal tests) (default = false)	optional	logical
spec_diagnostic	Specifies whether to perform a Regression Specification Error Test (RESET) (default = false)	optional	logical

Output Arguments

The function creates a structure array that contains all the results of the OLS regression and, if specified, substructures with the results of the diagnostic tests. Figure 1 presents an example for

the returned structure. The specific fields of the structure are described in Table 2.

Figure 1: Output Structure Example.



The screenshot shows a MATLAB variable named 'results' which is a 1x1 struct with 13 fields. The fields and their values are as follows:

Field	Value
nobs	1472
ncov	10
cov_corr	10x10 table
beta	11x1 double
y_head	1472x1 double
resid	1472x1 double
resid_variance	12.0721
hetero	1x1 struct
F_statistic	95.2524
p_valueF	0
fit_measures	1x1 struct
info_criteria	1x1 struct
diagnostics	1x1 struct

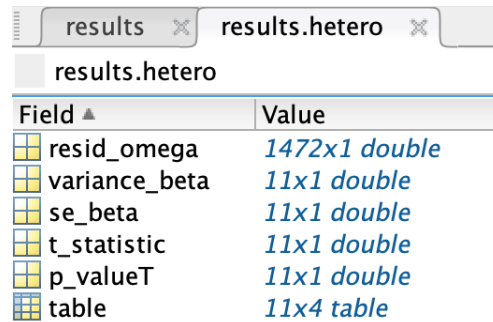
Details parental structure:

Table 2: Details about the fields contained in the output structure.

Field	Description	Type
nobs	Number of observations	double
ncov	Number of covariates	double
cov_corr	Pearson correlation matrix of the covariates	table
beta	OLS coefficients	column vector
y_head	Fitted values of the dependent variable	column vector
resid	OLS residuals	column vector
resid_variance	Variance of OLS residuals	double
hetero	Contains results related to heteroskedasticity	structure array
F_statistic	F statistic	double
p_valueF	p-Value of the F-statistic	double
fit_measures	Contains several measures of fit	structure array
info_criteria	Contains information criteria	structure array
diagnostics	Contains results of diagnostic tests	structure array

Substructure 'hetero':

Figure 2: Example: daughter structure 'hetero'.



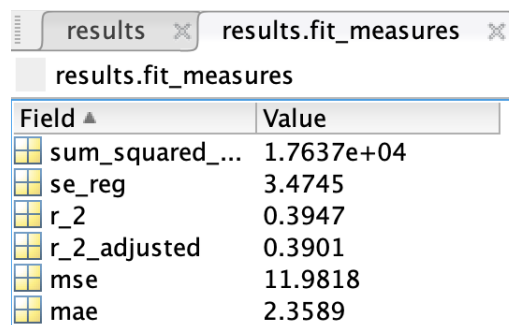
Field ▲	Value
resid_omega	1472x1 double
variance_beta	11x1 double
se_beta	11x1 double
t_statistic	11x1 double
p_valueT	11x1 double
table	11x4 table

Table 3: Details: daughter structure 'hetero'.

Field	Description	Type
resid_omega	Matrix containing variances of OLS residuals in the diagonal	column vector
variance_beta	Heteroskedasticity robust variances of OLS coefficients	column vector
se_beta	Heteroskedasticity robust standard errors of OLS coefficients	column vector
t_statistic	t-test statistics	column vector
p_valueT	p-values corresponding to t-test	column vector
table	Results table (formatted) for further use in LaTeX	table

Substructure 'fit_measures':

Figure 3: Example: daughter structure 'fit_measures'.



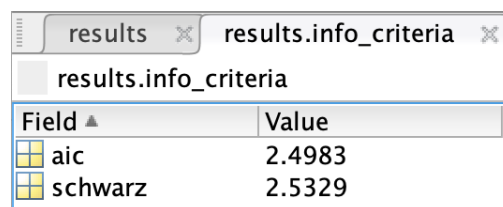
Field ▲	Value
sum_squared_...	1.7637e+04
se_reg	3.4745
r_2	0.3947
r_2_adjusted	0.3901
mse	11.9818
mae	2.3589

Table 4: Details: daughter structure 'fit_measures'.

Field	Description	Type
sum_squared_resid	Sum of squared residuals	double
se_reg	Standard error of regression	double
r_2	Multiple R^2	double
r_2_adjusted	Adjusted R^2	double
mse	Mean Squared Error	double
mae	Mean Absolute Error	double

Substructure 'info_criteria':

Figure 4: Example: daughter structure 'info_criteria'.



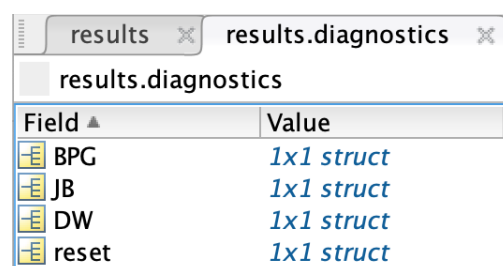
Field ▲	Value
aic	2.4983
schwarz	2.5329

Table 5: Details: daughter structure 'info_criteria'.

Field	Description	Type
aic	Akaike's information criterion	double
schwarz	Schwarz's information criterion	double

Substructure 'diagnostics':

Figure 5: Example: daughter structure 'diagnostics'.



Field ▲	Value
BPG	1x1 struct
JB	1x1 struct
DW	1x1 struct
reset	1x1 struct

Table 6: Details: daughter structure 'diagnostics'.

Field	Description	Type
BPG	Breusch-Pagan-Godfrey Test	structure
JB	Jarque-Bera Test	structure
DW	Durbin-Watson Test statistic	structure
reset	Regression Specification Error Test	structure

Additional information about the substructure 'diagnostics' is given in the part '*Diagnostics*' of this documentation.

Data inspection

The function is capable of conducting several input validations. First, the type and number of arguments that can be passed to the function is restricted. Please see the part '*Input Arguments*' for details about which arguments are required and which data types are allowed. Second, the function checks whether the dependent variable and the covariates have the same number of observations, if this is not the case, the function returns an input error. Third, the function validates if there are at least as many observations as covariates, if this is not the case, the function returns an input error. Fourth, the function tests whether there are any missing values in the table of the dependent variable or in the covariate table and returns an input error if there are any.

Regression diagnostics

The function is engineered such that it returns several diagnostics that can be used to evaluate if the OLS assumptions are satisfied.

Multicollinearity:

The function is able to detect perfect multicollinearity and returns an error in this case. In addition, the function calculates a Pearson correlation matrix of the regressors that the user can use to evaluate if there is imperfect multicollinearity between the regressors. This multicollinearity diagnostic is always performed and does not need to be activated.

Heteroskedasticity:

If the argument 'h_diagnostic' is set to 'true', the function generates two heteroskedasticity diagnostics. First, it plots the OLS residuals against the fitted values of the dependent variable (if regression with multiple regressors) or against the regressor (if regression with only one regressor). This figure can be used to graphically evaluate whether the OLS residuals are heteroskedastic. Second, the function conducts a Breusch-Pagan-Godfrey Test testing the null hypothesis of homoskedasticity at the 5 % significance level. The results of the test are stored in the substructure 'diagnostics.BPG' and the function prints the result also in the command window.

Further OLS residuals diagnostics:

Besides testing for heteroskedasticity, the function is capable of performing additional tests regarding the distribution and autocorrelation of the OLS residuals.

If the argument 'resid_diagnostic' is set to 'true', the function generates two diagnostics regarding the distribution of the OLS residuals. First, it plots a histogram of the OLS residuals that can be used to graphically evaluate the distribution of the OLS residuals. Second, the function conducts a Jarque-Bera Test testing the null hypothesis of normally distributed OLS residuals at the 5 % significance level. The results of the test are stored in the substructure 'diagnostic.JB' and are also printed in the command window.

If the argument 'resid_diagnostic' is set to 'true', the function also generates two diagnostics regarding the autocorrelation of the OLS residuals. First, it plots the OLS residuals against one period lagged OLS residuals. This figure can be used to graphically evaluate if there is any autocorrelation pattern at lag 1. Second, the function calculates the Durbin-Watson Test statistic and an estimate of the autocorrelation coefficient at lag 1. The results are stored in the substructure 'diagnostic.DW' and are also printed in the command window.

Regression Specification:

If the argument 'spec_diagnostic' is set to 'true', the function conducts a Regression Specification Error Tests (RESET) testing the null hypothesis of linearity at the 5% significance level. The results are stored in the substructure 'diagnostic.reset' and are also printed in the command window.

User guide

The user should take the following steps to set up the environment. To start with, the data set should be either selected from the current folder or imported manually. Its file name will also become the table name in the Matlab program unless specified otherwise. After importing the data set, the user should locate the OLS model and the data cleaning programs to set the directory by typing `cd` and the program location on the command window.

After completing the set up, the user should begin the data cleaning process to create variables of interest. The data cleaning programs are written outside the OLS model program as options for the users. They include log, square, dummy and cross-term transformation. For instance, if the user is interested in estimating the percentage change of one variable in response to another, then log-transformation of the variable can be conducted in the program by typing `my_log` and inserting the input. Then, it will return the variable in the log form, which can be merged with the data set. More details of the cleaning functions are written in the program files.

Subsequently, the user can select the variables of interest from the data table for the OLS regression. First, the type of the input variables needs to be table in the `OLSmodel()`. It is because a table saves both the observed values and the names of the variables. It is done by typing a round bracket i.e. `data(:, variable_name)`. Notice a curly bracket following `data` only gives a vector or a matrix which forgoes the variable name and it becomes insufficient for the OLS model.

Matlab computational methods

The OLS function algorithms consist of various estimates of parameters, test-statistics, and diagnostics tests. Most of the estimates use matrix multiplication. For instance, the covariate coefficients are calculated as in the equation below:

$$\beta = (X'X)^{-1}(X'Y)$$

where β is a column vector of coefficients including an intercept, X is a matrix of the observed values of covariates containing a constant of one, and Y is a column vector of the observed values of the dependent variable. In Matlab, this matrix multiplication can take advantage of the right divide operator ' \backslash ' to simultaneously invert and multiply a matrix with another matrix. Thus, the equation can also take the form: $(X'X)\backslash(X'Y)$.

Another useful tool of the Matlab function program is nested functions. The OLS function relies on complex mathematical equations, such as integrals and gamma distributions, that need to be reused in the program. Therefore, in the beginning of the OLS program, the complex functions are defined. For instance, the p-value of the F-statistics is based on the F-distribution function, the critical value and the degree of freedom. The nested function defines the cumulative density function, which embeds a probability density function. Then, when calculating the p-value, simply summon the function and insert the input, such as the test-statistics and the degrees of freedom. Eventually, the nested functions will return the p-value. Thus, it saves time and avoids computational mistakes from writing the same equation. In addition, the code is highly modular such that the nested functions could also be used in other applications.

Finally, the Matlab program utilises basic elements such as for-loops and conditions. The for-loop is mainly used in the warnings part to detect anomalies in the input data. Moreover, a condition is usually implemented in both the diagnostics and warnings part to execute one of the if/else command. The diagnostics test is only executed if the condition is true.

Economics Problem and Findings

The BWAGES file contains 1472 observations from the 1994 wave of the Belgian part of the European Community Household Panel. The dataset contains cross-sectional variables of individuals such as wage, level of education, experience and gender. The data also gives the natural logarithm of level of education, experience and wage. The summary statistics are shown in table 7. The range of the dependent variable wage seems large enough for a linear OLS model. There is also an approximate balance between male and female respondents, however males are slightly over represented which might lead to a selection bias.

Table 7: Summary Statistics

	Mean	Var	Std	Max	Min
WAGE	11.0506	19.8071	4.4505	47.5755	2.191
LNWAGE	2.3344	0.13143	0.36253	3.8623	0.78435
EDUC	3.3784	1.4509	1.2045	5	1
EXPER	17.2174	103.3613	10.1667	47	0
LNEXPER	2.6907	0.53176	0.72922	3.8712	0
LNEDUC	1.1365	0.18833	0.43397	1.6094	0
MALE	0.60666	0.23879	0.48866	1	0

But there are many ways to dig deeper into the data and study the economic problems. For example, one can estimate the effect of higher education and work experience on wages. The OLS regression model is run on various combinations of the outcome variables and covariates and the results are discussed and shown below.

a) Regression of *wage* on *male* or/and *female*:

The *female* dummy variable can be created by subtracting 1 from the *male* dummy variable. If we add both dummies as regressors, there is a perfect multicollinearity issue as the sum of the dummies is equal to 1 for all observations, thus the intercept can be represented by a linear combination of the dummies. This situation is also known as the dummy variable trap and can be avoided by dropping one of the dummies. The intuition behind this is that one cannot estimate the ceteris paribus effect of a change of, for example, the *male* dummy on the dependent variable in this scenario since the change of the *male* dummy would also imply a change of the *female* dummy. Our OLS function embeds an algorithm to detect perfect multicollinearity by checking whether the covariate matrix has full rank. If it does not have full rank, the function returns an error message to the users, illustrated in figure 6.

Figure 6: Perfect multicollinearity warning

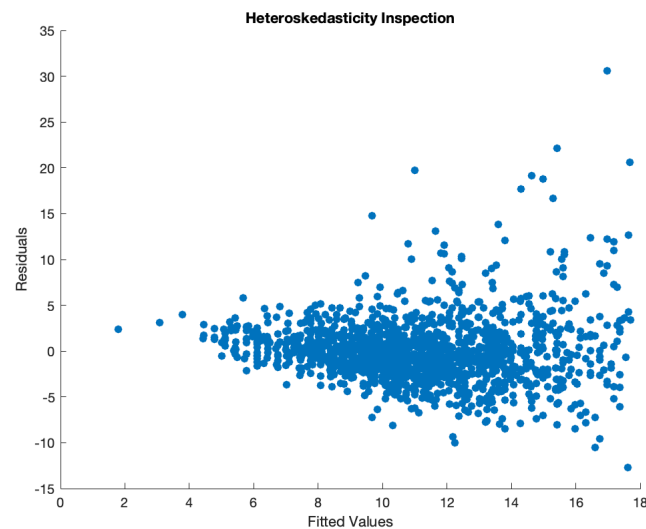
```
Error using OLSmodel (line 336)
Perfect multicollinearity detected

Error in OLS_fct_applied (line 24)
results = OLSmodel(data1(:, 'WAGE'), data1(:, {'MALE', 'FEMALE'}),...
```

b) Regression of *wage* on *male*, *educ*, *exper*, *exper*²:

The coefficients on the covariates are statistically significant. But it would be useful to detect issues with the regression by checking the pattern of the residual and fitted outcome values from a scatter plot. To visualise it, simply set the *resid_diagnostic* argument in the function to be true. Then the fitted values of the outcome variable and the residuals are visualised. Figure 7 depicts that the variance of the residuals increases as the fitted outcome values rise. This systemic pattern is called heteroskedasticity.

Figure 7: Heteroskedasticity check



c) Regression of $\log(wage)$ on $male$, $\log(educ)$, $\log(exper)$, $\log^2(exper^2)$:

By setting the 'h_diagnostic' argument in the *OLSmodel* function to be 'true', the function again plots the OLS residuals against the fitted outcome values. The regression bears the same heteroskedasticity issue as the one without logging, as the residual scatter plot again shows a heteroskedastic pattern. Table ?? shows a Breusch-Pagan-Godfrey test testing the null hypothesis of homoskedasticity that can be rejected at the 5 % significance level. Hence, one should use heteroskedasticity robust standard errors, by setting the 'h_robust' argument to 'true'. In addition, a Jarque-Bera Test testing the null hypothesis of normally distributed OLS residuals leads to the conclusion that the latter null hypothesis can also be rejected at the 5 % significance level. As the assumption of normally distributed OLS residuals is relevant to hypothesis testing, we take a look at the histogram of the OLS residuals that is also created by the function. It seems that the normal distribution is nonetheless a good approximation of the residual distribution. The latter is also implied by the Central-limit theorem. Furthermore, to evaluate whether the OLS residuals are autocorrelated, the function returns the Durbin-Watson Test statistic that is equal to $DW = 1.8863$ in this case. As the test statistic is smaller than 4 minus the upper bound of the critical value at the 5% significance level ($d = 1.919$), we cannot reject the null hypothesis of the autocorrelation coefficient being larger or equal to zero at the 5 % significance level. To conclude, the optional diagnostics of the *OLSmodel* function are very helpful to check the validity of the OLS assumptions.

Figure 8: Diagnostic tests

```
#####  
#####  
Heteroskedasticity Diagnostic:  
Visual inspection: Please see the created figure.  
Formal Test: Breusch-Pagan-Godfrey Test.  
  
Breusch-Pagan-Godfrey Test:  
Test statistics: 3.286894e+01  
P-value: 1.270601e-06  
The null hypothesis of homoskedasticity can be rejected at the 5 % significance level.  
  
#####  
  
Further OLS Residuals Diagnostics:  
Visual inspection: Please see the created figure.  
Formal Tests:  
  
Regarding distribution of OLS residuals:  
Jarque-Bera Test:  
The null hypothesis of normally distributed residuals can be rejected at the 5 % significance level.  
  
Regarding autocorrelation (lag = 1):  
Please use the DW statistic that is given below to perform DW tests.  
Durbin-Watson Test statistic: 1.8863  
Estimate autocorrelation (l=1): 0.056848
```

- d) Discussion of *educ* and regression of wage on covariates, dummy variables and cross terms:
The *EDUC* is an ordinal variable where the size of a interval is unclear. For instance, it cannot be said that a one unit increase from 1 to 2 is the same as that from 2 to 3, as the intervals can be ranging between different educational quality i.e. from middle school to high school and from high school to university. So the impact of an one-unit rise is often non-linear. It invokes an issue in OLS since it always assumes a linear correlation between the outcome variable and the covariates. One can transform the *EDUC* variable into multiple dummies to remedy it. Table 8 underlines that the coefficients of the education dummy variables are increasing exponentially as the level rises. So a dummy-transformation allows a flexible rate of change and mitigate the ordinal issue.
Furthermore, our function is capable of conducting a Regression Specification Error Test by setting the 'spec_diagnostic' argument to 'true'. In this particular regression specification from above, the null hypothesis of a linear specification cannot be rejected at the 5 % significance level. Thus, it seems that our linear specification can be used in this application.

Table 8: OLS regression of WAGE on MALE, EDUC, EXPER, and EXPER_2

	Coefficient	SE	t-Statistic	p-Value
Intercept	6.1254	0.48068	12.7433	0
MALE	1.2387	0.18453	6.7126	0
EXPER	0.052292	0.017403	3.0048	0.002703
EDUC2	-0.1922	0.58158	-0.33048	0.74109
EDUC3	0.72241	0.54066	1.3362	0.18171
EDUC4	1.3418	0.53546	2.5058	0.012324
EDUC5	2.6652	0.6067	4.3929	1.2e-05
EDUC2_EXPER	0.062277	0.025798	2.414	0.015899
EDUC3_EXPER	0.10009	0.025322	3.9527	8.1e-05
EDUC4_EXPER	0.17495	0.02579	6.7835	0
EDUC5_EXPER	0.25166	0.035835	7.0226	0

Conclusion

The OLS model is capable of conducting statistical tests, predicting the outcome, and checking for systemic issues. Users can also use other Matlab functions such as *fitlm* to get the regression. Other functions and tests can also be included in our function for future use.