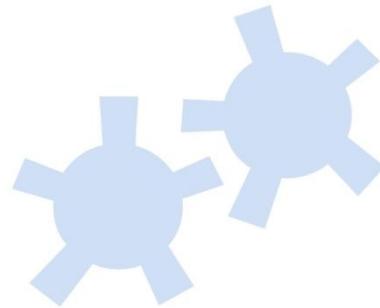


VIII ITALIAN STATA USERS MEETING
Isola di San Servolo, Venezia November 17-18, 2011



Sar: Automatic Generation of Statistical Reports
Using Stata and Microsoft Word for Windows

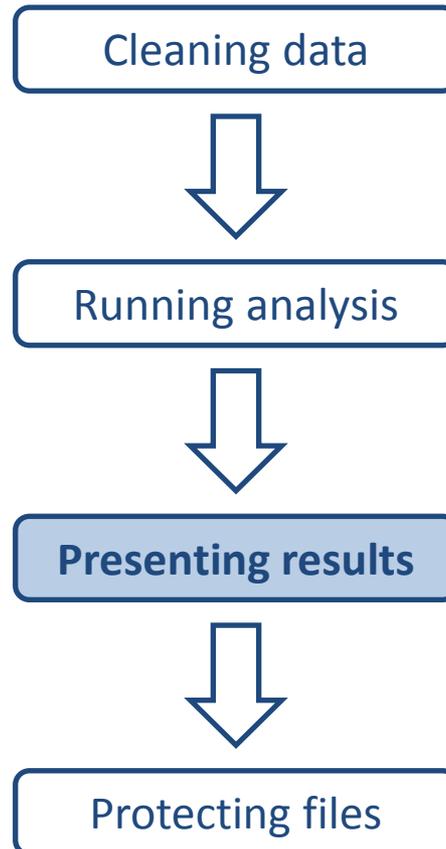
Giovanni Luca Lo Magno

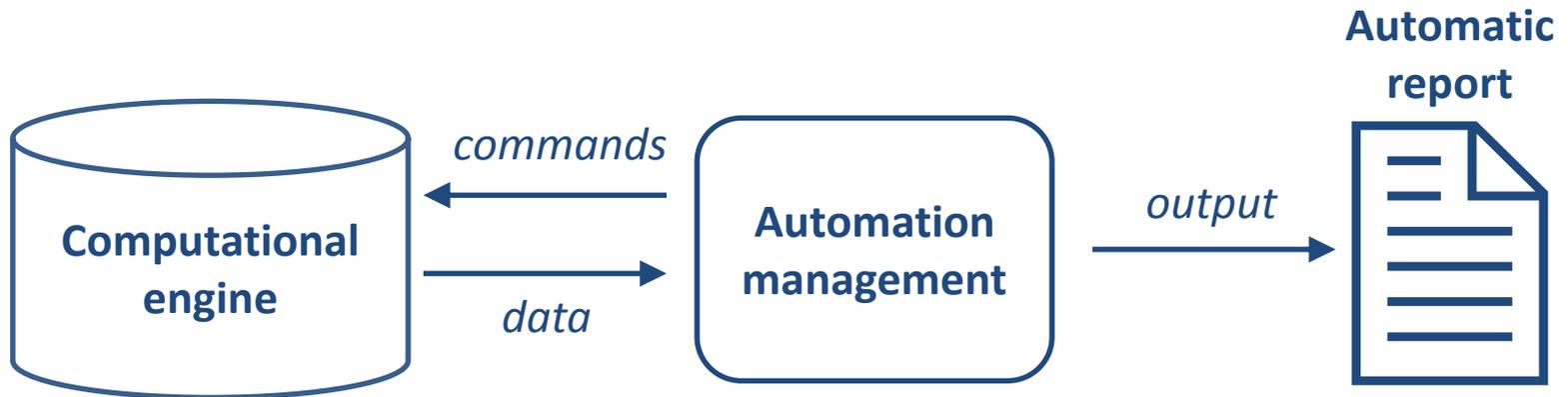
lomagno.gl@virgilio.it

Department of Economics, Business and Finance

University of Palermo

Currently under review by the Stata Journal







- `listtex` by Newson (2003)
- `textab` by Hardin (1995)
- `estout` by Jahn (2005)
- `estab` by Jann (2007)
- `outreg` by Gallup (1998)
- Automatic generation of documents, discussed in Gini e Pasquini (2006)



General limits:

- Tex/Latex oriented
- Not easy to learn
- Not "what you see is what you get approach" (WYSIWYG)
- Not complete solutions



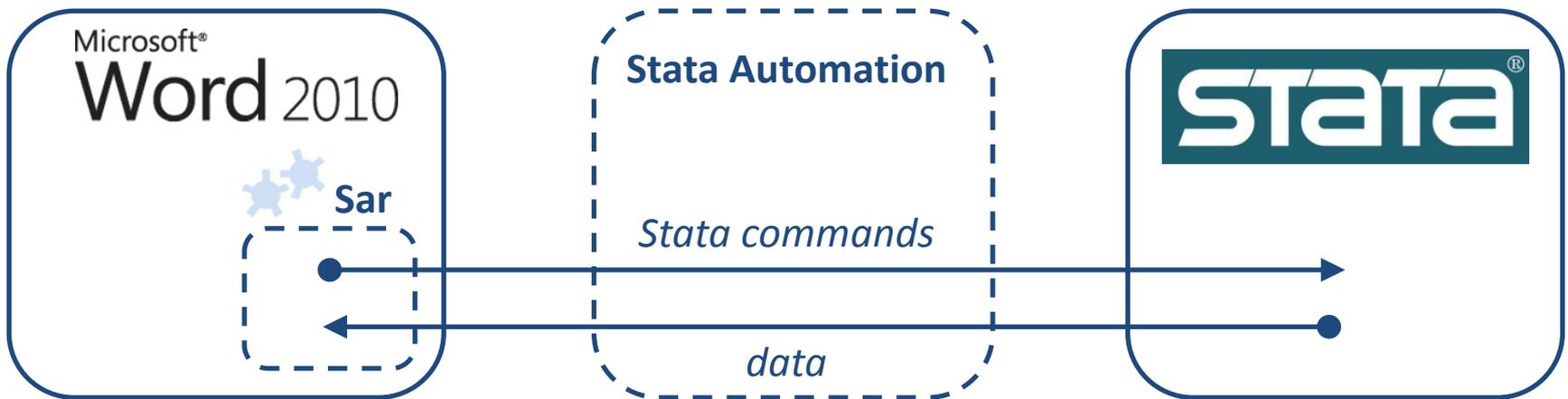
Sar is a software which allows you to automatically obtain numerical results from Stata in Word, making the formatting of statistical results easier



Sar is not a Stata command, but a macro for Microsoft Word written in the Visual Basic for Applications (VBA) programming language



Current version of Sar is 1.0

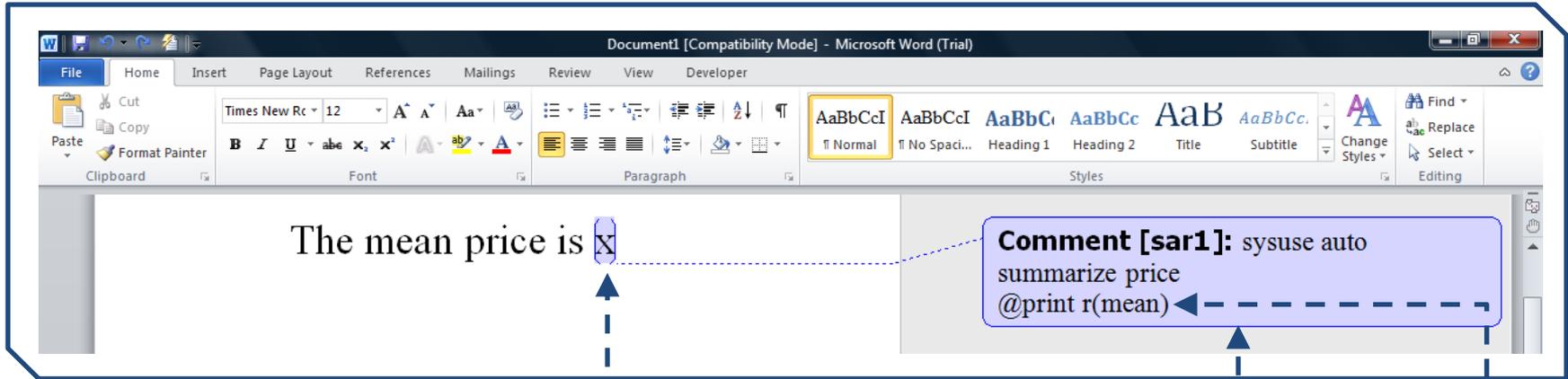


Stata Automation is a communication mechanism between Stata and Microsoft Windows applications
(read www.stata.com/automation for details)

A quick look to Sar at work

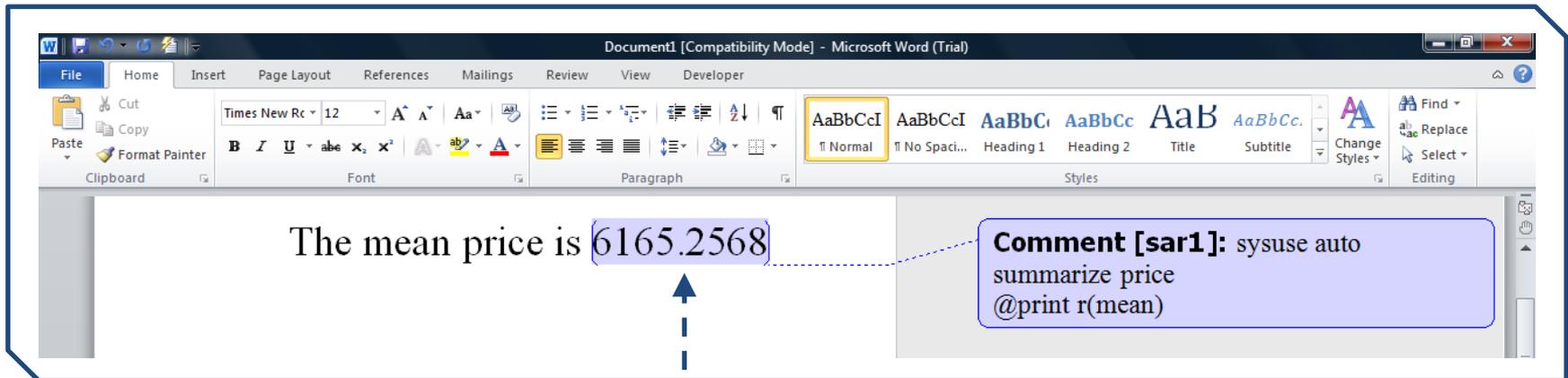


Before executing Sar



- Temporary text placeholder
 - Only comments with initials "sar" are processed by sar
 - The @print command replaces the temporary text placeholder with data retrieved from Stata
- note: all Sar commands begin with @*

After executing Sar



Data retrieved from Stata



Wrong!

```
Comment [sar1]: use "C:\mydata.dta"  
regress y x  
matrix beta = e(b)'
```



Correct!

```
Comment [sar1]: use "C:\mydata.dta"  
regress y x  
matrix beta = e(b)'
```

Hint: press Control+Z after you typed the wrong quotation mark or apostrophe in Word



Software requirements

- Microsoft Word for Windows (Stata Automation only runs on Windows)
- Stata (of course!)



Manual settings

- You have to install the Stata Automation object
- You have to copy the Sar macro ("Stata automatic report 1.0.dotm") in the Word startup folder
- Optionally, you can customize the Word quick access toolbar creating a button to easily execute the Sar macro
- You have to set user's initials of Word comments to "sar"



1. Right-click on the Stata executable (on my machine the executable file is *westata.exe* and it is located in the *C:\Stata10* folder)
2. Choose "Create shortcut" (I suggest you to rename the shortcut to "*stata automation*")
3. Right-click on the just created shortcut, choose "Property" and change Target from "*C:\Stata10\westata.exe*" to "*C:\Stata10\westata.exe /Register*" (please check the correctness of your Stata path and its executable)
4. Right-click on the shortcut and choose "Run as administrator"

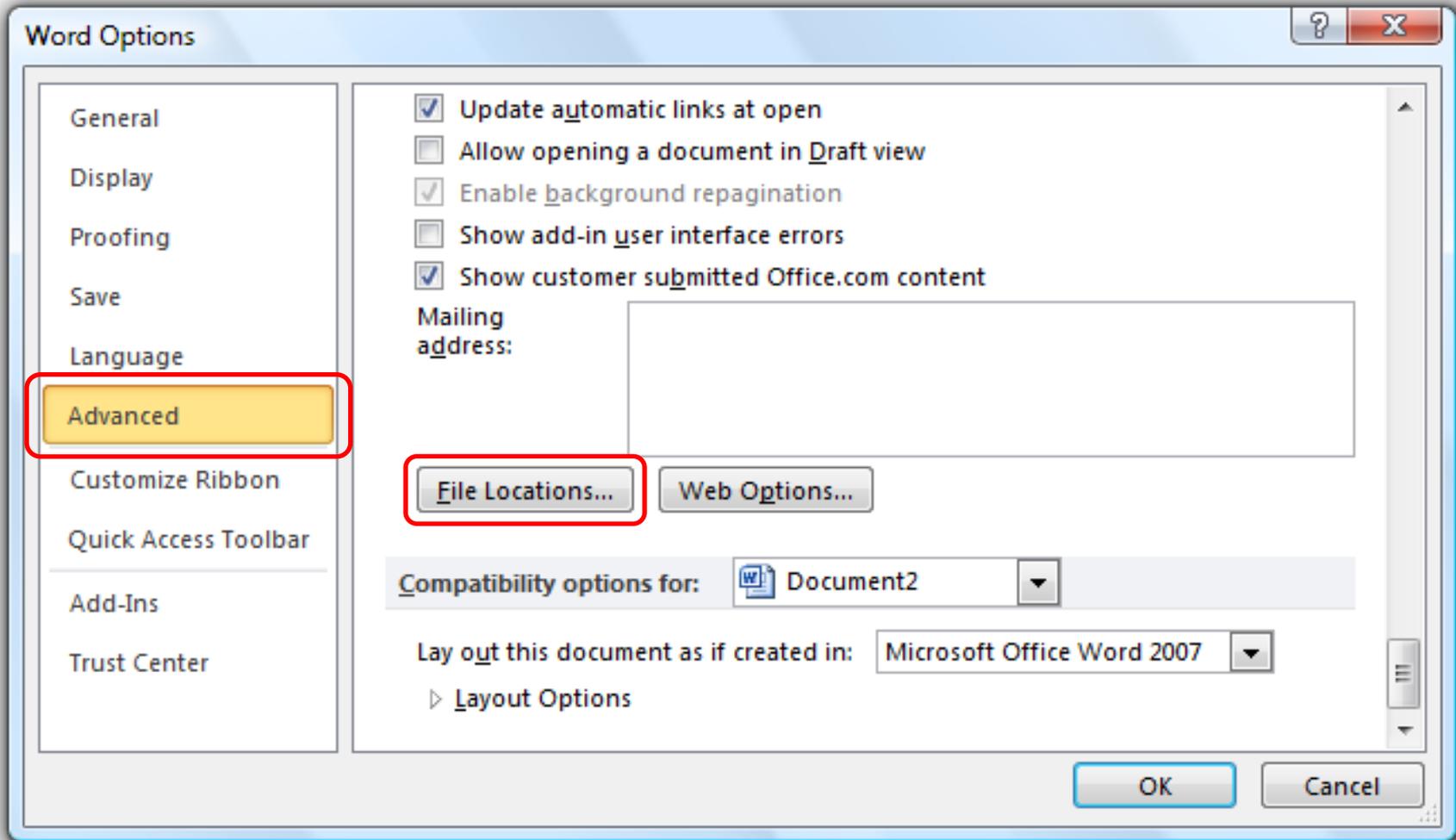
(read www.stata.com/automation for more details and informations about how to install the Stata Automation object on a Windows non-Vista machine)

Copying the Sar macro in the Word startup folder



In Microsoft Word:

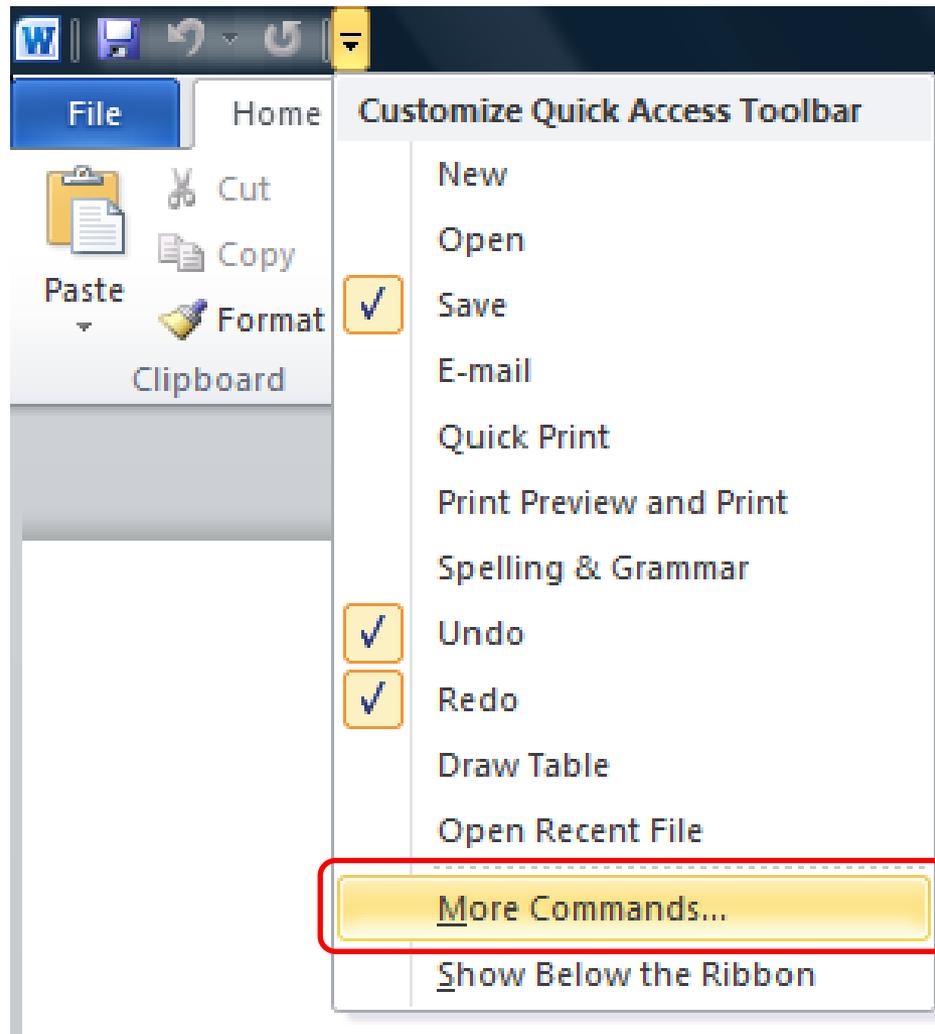
File ⇒ Options ⇒ Advanced ⇒ File Locations...



Click on the "File Locations..." button to find out where the Word startup folder is and copy the "Stata automatic report 1.0.dotm" macro file in it



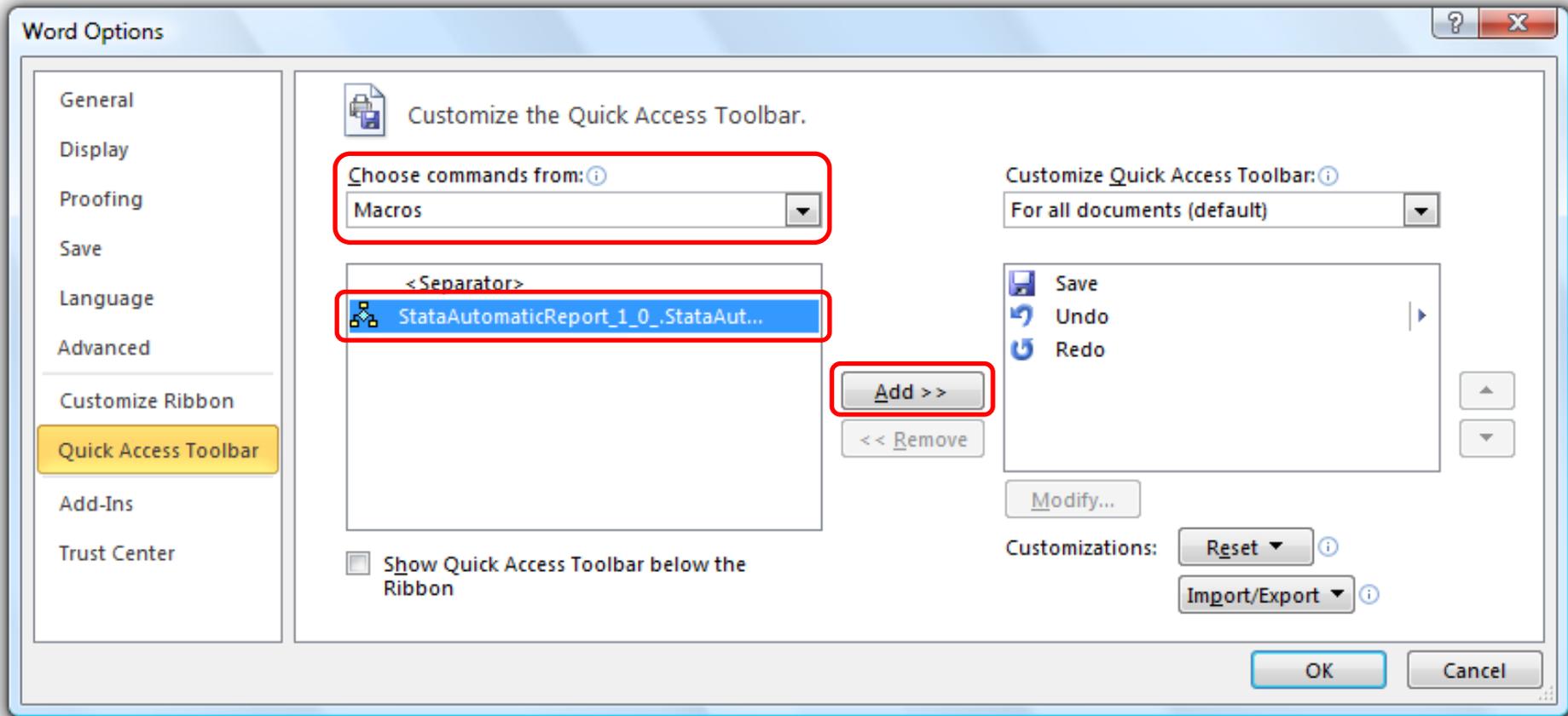
In Microsoft Word:
Customize Quick Access Toolbar ⇨ More Commands...



Customize the Word quick access toolbar (Step 2 of 4)



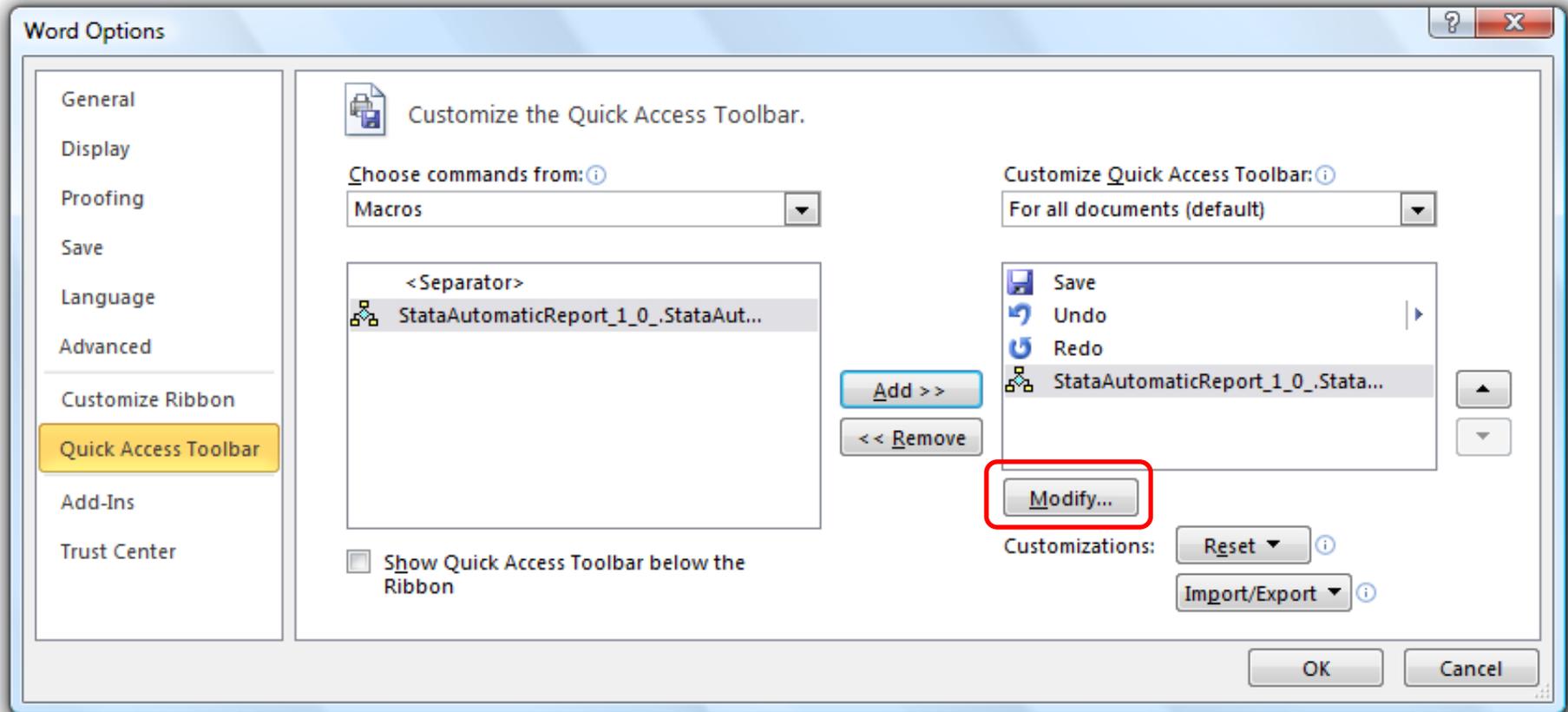
In Microsoft Word, after Step 1:
Choose "Macros" from the "Choose commands from" list ⇒ Select the Stata automatic report macro ⇒ Click on the "Add" button





In Microsoft Word, after Step 2:

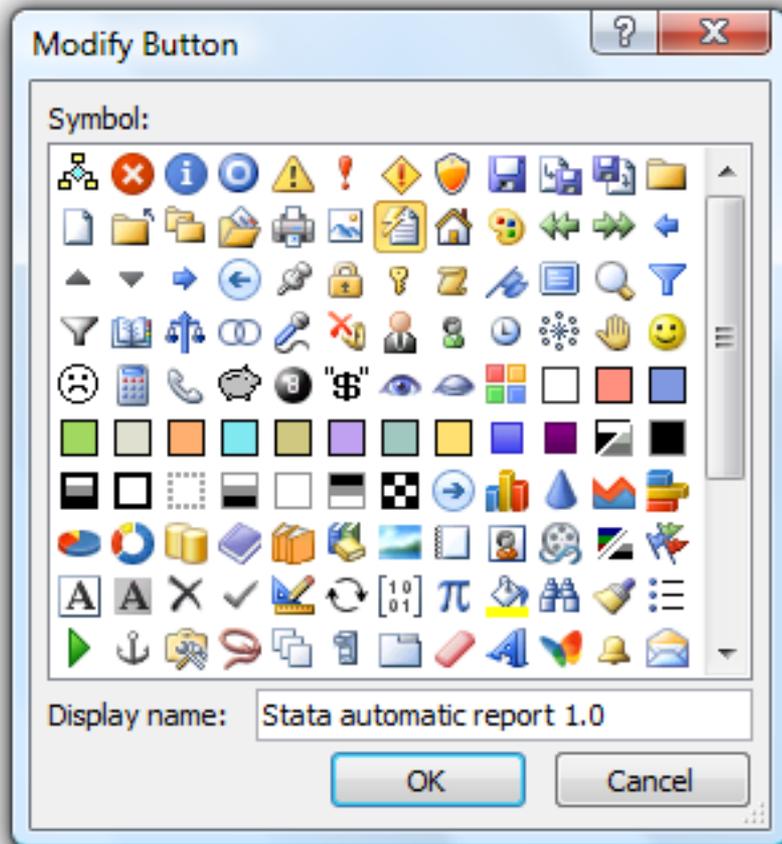
Select the Sar macro from the right list ⇒ Click on the "Modify..." button to choose an icon for the button which will be added to the quick access toolbar



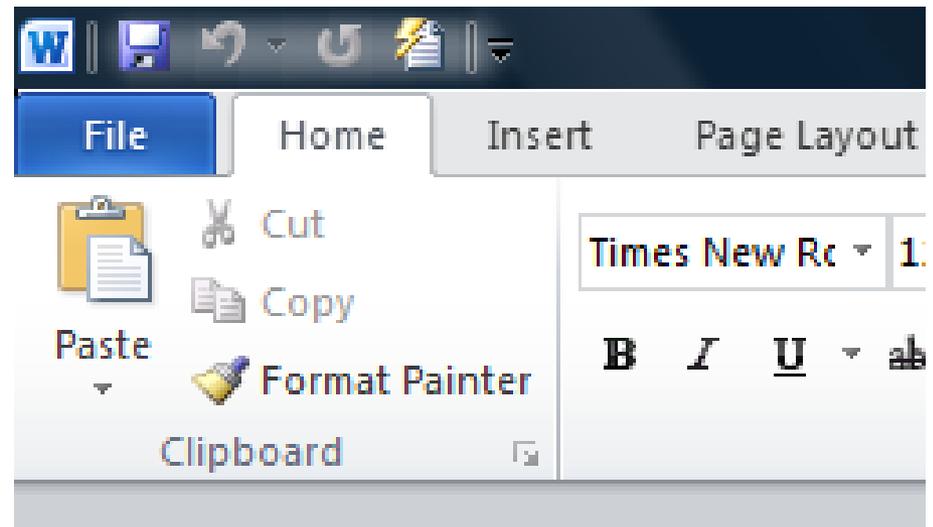
Customize the Word quick access toolbar (Step 4 of 4)



In Microsoft Word, after Step 3:
Choose your favorite icon and change the display name to "Stata automatic report 1.0"



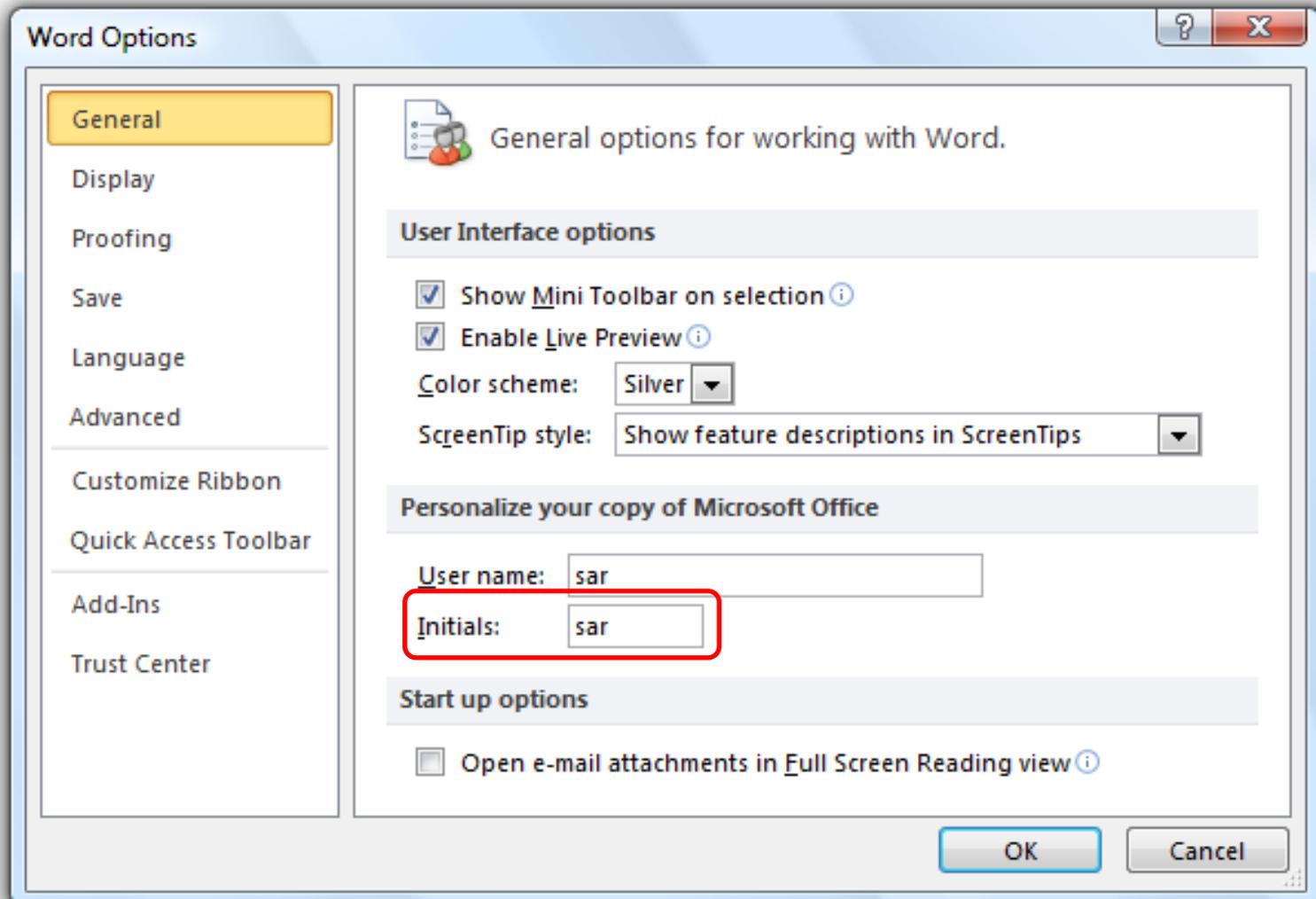
This is your Sar button on the quick access toolbar. You can launch the Sar macro by clicking on it



Setting user's initials of Word comments to "sar"



In Microsoft Word:
Word Options ⇒ General ⇒ Set user's initials to "sar"





Syntax

```
@print StataData
```

```
@format %StataNumericalFormat
```

My beautiful report

In this report I comment some summary statistics from the auto.dta dataset.

The mean price is 6165.3, the number of observations is 74 and there are 22 foreign cars.

Comment [sar1]: sysuse auto
summarize price

Comment [sar2]: @format %5.1f
@print r(mean)

Comment [sar3]: @format %2.0f
@print r(N)

Comment [sar4]: count if foreign==1
@print r(N)

Note: no leading spaces are added to the numerical output even if they are expected according to the Stata formatting rules



Syntax

```
@filltable StataData startingRow startingCol [rowStep colStep]
@matrixrownames StataMatrix startingRow startingCol [rowStep]
```

My regression analysis

Variable	Coefficient
<i>weight</i>	4.7
<i>length</i>	-98.0
<i>cons</i>	10386.5

Comment [sar1]: sysuse auto
regress price weight length
matrix beta = e(b)'

Comment [sar2]: @format %5.1f
@filltable beta 2 2
@matrixrownames beta 2 1



Syntax

```
@filltable StataData startingRow startingCol [rowStep colStep]
@matrixrownames StataMatrix startingRow startingCol [rowStep]
@beginstring #string#
@endstring #string#
```

My regression analysis

Variable	Coefficient
<i>weight</i>	4.7 (1.1)
<i>length</i>	-98.0 (39.2)
<i>cons</i>	10386.5 (4308.2)

Note: standard errors in parentheses

Comment [sar1]: sysuse auto
regress price weight length
matrix beta = e(b)
mata: V = st_matrix("e(V)")
mata: st_matrix("sd", sqrt(diagonal(V)))

Comment [sar2]: @format %5.1f
@filltable beta 2 2 1 0
@beginstring #(#
@endstring #)#
@filltable sd 3 2 1 0
@matrixrownames beta 2 1 1



Roughly speaking, a Sar program is a list of Sar and Stata commands:

Syntax

```
@program myprog [arg1 arg2 ... argN]  
[...]  
[my Sar and Stata commands]  
[...]  
@end
```

It can be defined:

- in a Word comment
- in a plain text file (called library in the Sar jargon)

It can be executed using the @do command:

Syntax

```
@do myprog [arg1 arg2 ... argN]
```



Program definition

```
program name --- @program outmatrix matrix --- argument
@matrixrownames §matrix§ 2 1
@matrixcolnames §matrix§ 1 2
@format %4.3f
@filltable §matrix§ 2 2
@end
--- callbacks
```

Example of usage

```
correlate price weight length
@do outmatrix r(C)
```



The compiled program

```
@program outmatrix r(C)
@matrixrownames r(C) 2 1
@matrixcolnames r(C) 1 2
@format %4.3f
@filltable r(C) 2 2
@end
```

**A correlation matrix**

	<i>price</i>	<i>weight</i>	<i>length</i>
<i>price</i>	1.000	0.539	0.432
<i>weight</i>	0.539	1.000	0.946
<i>length</i>	0.432	0.946	1.000

Comment [sar1]: @program outmatrix
matrix

@matrixrownames §matrix§ 2 1

@matrixcolnames §matrix§ 1 2

@format %4.3f

@filltable §matrix§ 2 2

@end

Comment [sar2]: sysuse auto

correlate price weight length

@do outmatrix r(C)

You can use the just defined outmatrix program how many times you want in your Word document



Syntax

```
@loadlibrary "pathOfTheLibraryFile"
```

Plain text file: *c:\sar libraries\mylibrary.txt*

```
@program outmatrix matrix
@matrixrownames §matrix§ 2 1
@matrixcolnames §matrix§ 1 2
@format %4.3f
@filltable §matrix§ 2 2
@end
```

A correlation matrix

	<i>price</i>	<i>weight</i>	<i>length</i>
<i>price</i>	1.000	0.539	0.432
<i>weight</i>	0.539	1.000	0.946
<i>length</i>	0.432	0.946	1.000

Comment [sar1]: @loadlibrary "c:\sar libraries\mylibrary.txt"

Comment [sar2]: sysuse auto
correlate price weight length
@do outmatrix r(C)



Syntax

@resetstring

(no arguments are required)

Plain text file: *c:\sar libraries\mylibrary.txt*

```
@program regressout
matrix beta = e(b)'
mata: V = st_matrix("e(V)")
mata: V = st_matrix("sd", sqrt(diagonal(V)))
@format %10.1f
@filltable beta 2 2 1 0
@matrixrownames beta 2 1 1
@beginstring #(#
@endstring #)#
@filltable sd 3 2 1 0
@resetstring
@format %3.0f
@filltable e(N) -2 2
@format %4.3f
@filltable e(r2) -1 2
@end
```

Note:

When arguments of the @filltable command are -1, -2, etc... they indicate the last row/column, the second last row/column and so on.

Second last row

Last row



My regression analysis

Variable	Coefficient
<i>weight</i>	4.7 (1.1)
<i>length</i>	-98.0 (39.2)
<i>_cons</i>	10386.5 (4308.2)
<i>N. of obs.</i>	74
<i>R²</i>	0.348

Note: standard errors in parentheses

Comment [sar1]: sysuse auto
@loadlibrary "c:\sar libraries\mylibrary.txt"

Comment [sar2]: regress price weight
length
@do regressout

Note: Sar is not verbose!





Syntax `@interact`

(no arguments are required)

Example: our goal is to create the well-known $(X'X)^{-1}$ matrix

The execution of Sar will halt here, allowing the user to interact with Stata

A tricky matrix


```
Comment [sar1]: @interact  
@filltable mymatrix 1 1
```

*This matrix will be created
by the user in Stata*



```
*****
*
*           Stata session called from Sar
*
* Warning: don't close this Stata window from here, it will cause the crash of Sar
*           Close this Stata window from the dialog window appeared in Word
*
*****
sysuse auto
(1978 Automobile Data)
. * suppose you don't remember how to use the mkmatrix command
. help mkmatrix
. mkmatrix mpg weight, matrix(X)
. count
  74
. matrix one = j(74, 1, 1)
. matrix X = X, one
. matrix mymatrix = invsym(X' * X)
```

This is our final matrix

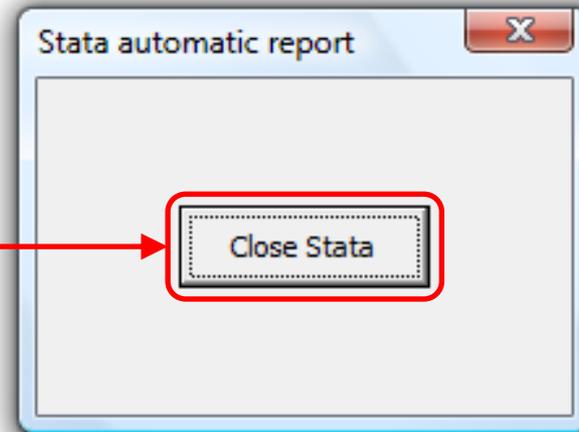
Commands typed by the user in Stata

This Stata window has to be closed from Word (see Step 3)



This dialog window will be opened in Word after the execution of @interact

Don't forget to click here after interacting with Stata!



This is the final output:

A tricky matrix

.00117444	7.057e-06	-.04632027
7.057e-06	6.508e-08	-.0003468
-.04632027	-.0003468	2.047163

Comment [sar1]: @interact
@filltable mymatrix 1 1

Probably you will delete this Sar comment (it's useless)



Do file: *c:\mydofile.do*

```
sysuse auto
summarize price
global mean: display %5.1f r(mean)
global nObs: display %2.0f r(N)
count if foreign==1
global nObsForeign: display %2.0f r(N)
```

My beautiful report

In this report I comment some summary statistics from the auto.dta dataset.

The mean price is 6165.3, the number of observations is 74 and there are 22 foreign cars.

Comment [sar1]: do "c:\mydofile.do"

Comment [sar2]: @print mean

Comment [sar3]: @print nObs

Comment [sar4]: @print nObsForeign

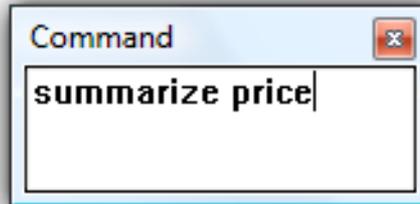


Creating automatic reports



- There is no need to edit your report if data have changed
- The report is well documented

Using Sar in interactive mode



You can obtain data from Stata on the fly

Calling do files from Sar



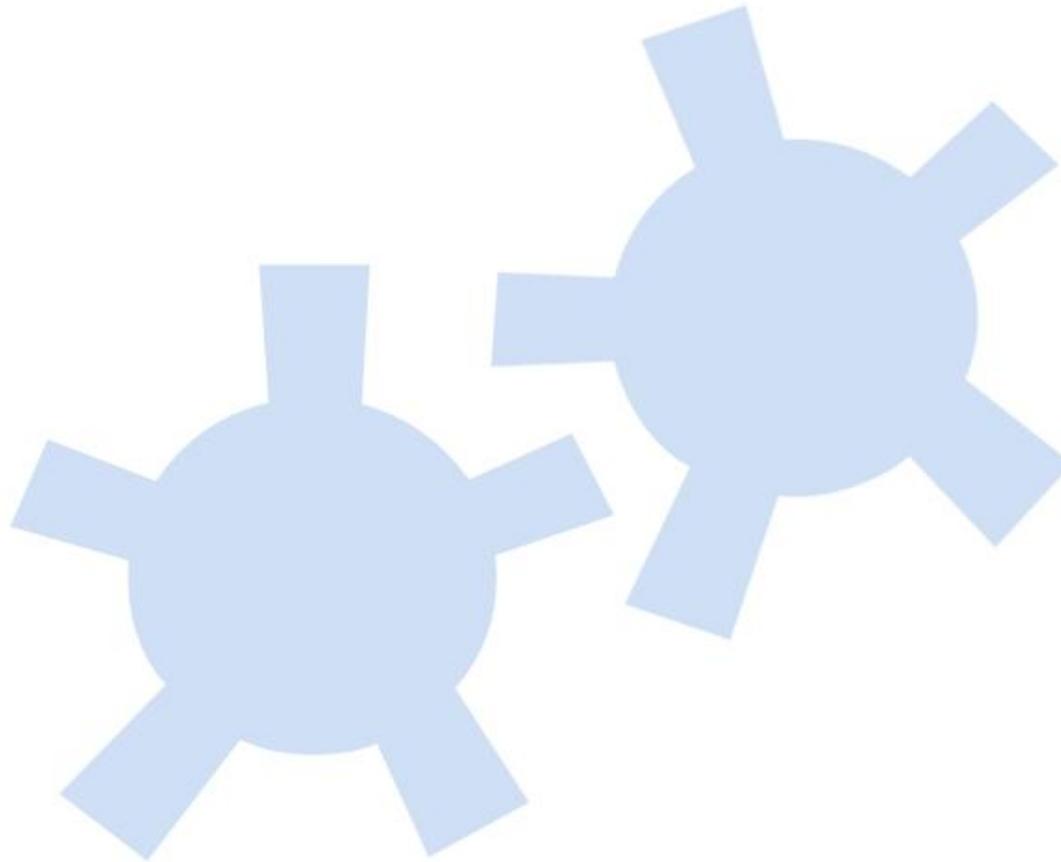
- You can test your do file in Stata
- You can store your statistical analysis in a do file



- Automatic reports: documents which can auto-update themselves if data have changed
- Self-explaining data analysis
- WYSIWYG approach exploiting all the functions of Word
- Ease of learning
- Only 15 keywords
- Sar documents are not verbose
- Extensibility through Sar programs



- Sar only works in Windows
- Lack of "undo" function to erase all changes made by Sar on the document
- The following Stata commands can not be used: `program`, `define`, `while`, `forvalues`, `foreach` and `input` (but they can be used in do files)
- The `@print` command can not be used inside a Word table
- Setting of global and local macros has no effect in Sar
- Word comments with Sar commands can not refer to the same portion of a text
- You have to avoid to use the macro names "stataAutomaticReportValue" and "stataAutomaticReportMatrix", because they are internally used by Sar

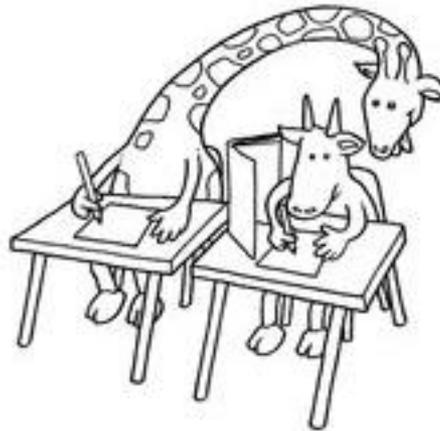


Thank you for your attention



The problem

Students like copying from their schoolmates during statistics classwork



The solution

An automatic document created with Sar where numerical values of the exercises are randomly sampled from a dataset. A code-seed is uniquely assigned to each student. The teacher can use the code-seed to quickly reproduce the sampled dataset and mark the schoolwork.



Schoolwork

Student's name: _____

Student's ID: **21**

Exercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight
VW_Diesel	5397	2040
Buick_Skylark	4082	3400
Merc_XR-7	6303	4130
Toyota_Corona	5719	2670
Buick_Century	4816	3250
Subaru	3798	2050
Chev_Malibu	4504	3180

Comment [sar1]: sysuse auto

```
set seed 21
sample 7, count
encode make, generate(makeNumeric)
mkmat makeNumeric, matrix(make)
rownames(make)
mkmat price, matrix(price)
mkmat weight, matrix(weight)
```

You should manually change the seed in the comment and in the document, launch Sar and print the document: that's very boring if you have many students. It's better to use the mail-merge functions by Word.

Comment [sar2]: @matrixrownames

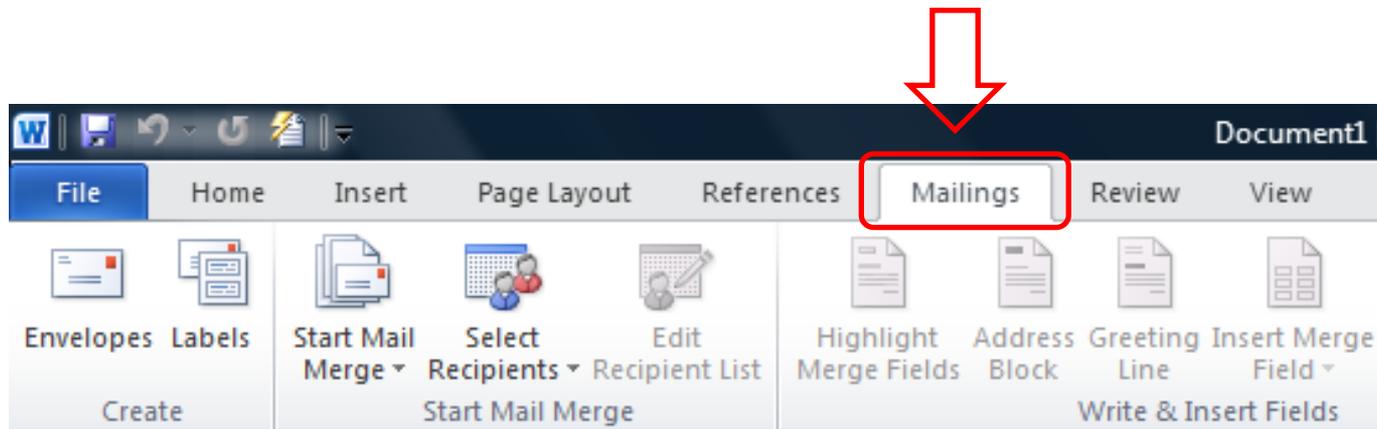
```
make 2 1
@filltable price 2 2
@filltable weight 2 3
```



What is *mail-merge* is?

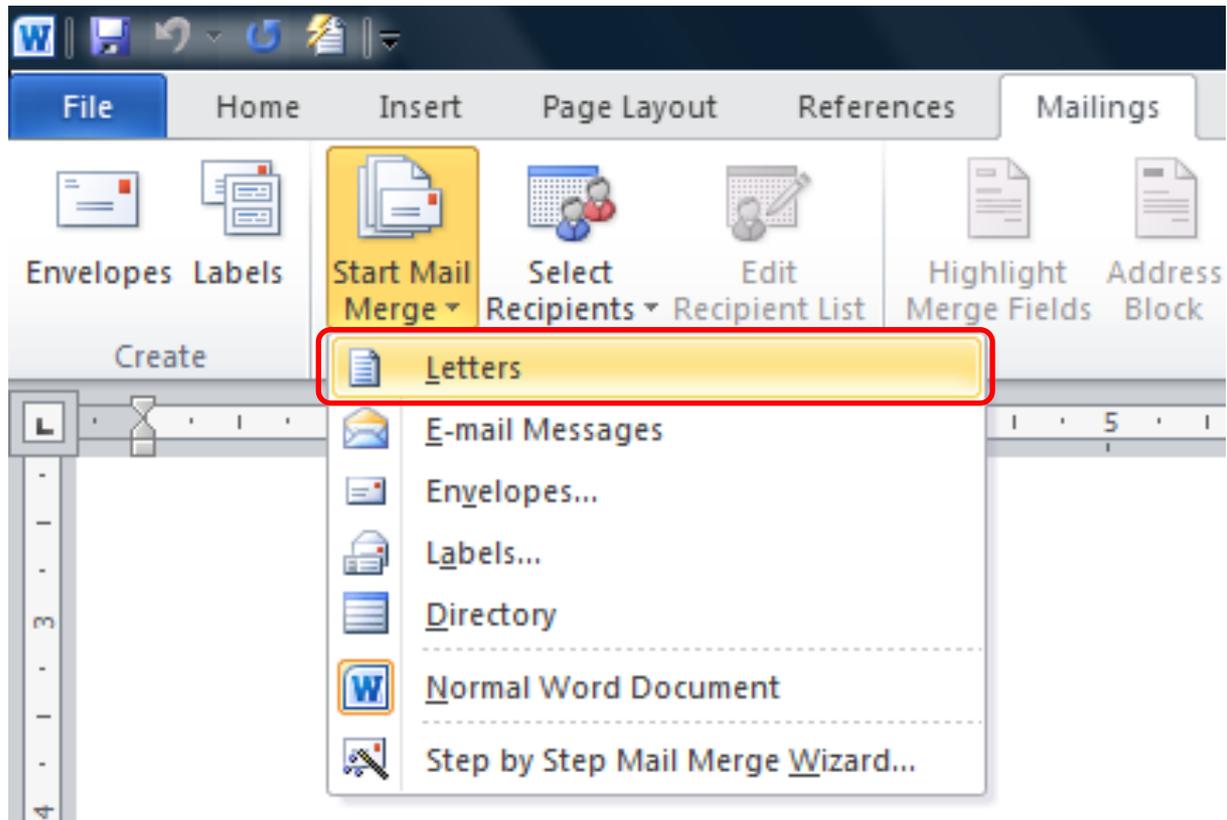
It is a software function which allows you to create multiple documents from a template

How you can access the mail-merge functions of Word:



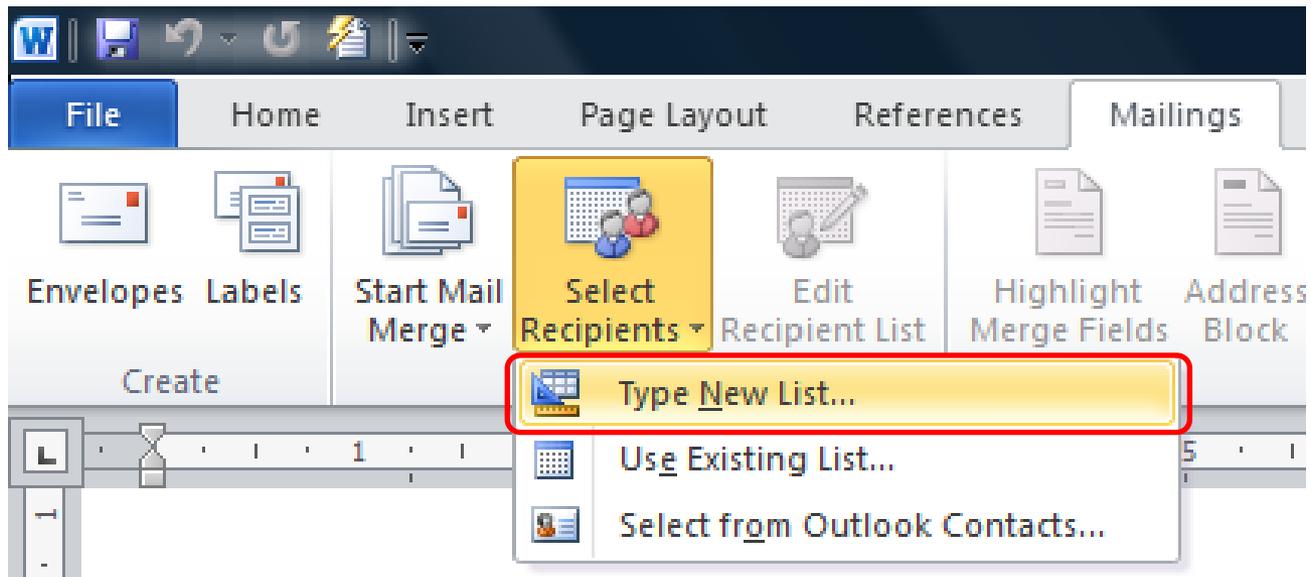


Step 1: Select the «letters» document type





Step 2: Create a new database





Step 3: Fill the database with your students' names and a univocal ID

New Address List

Type recipient information in the table. To add more entries, click New Entry.

Title	ID	First Name	Last Name	Company Name
	1	Christine	Giraffe	
	2	John	Goat	
	3	Carmen	Donkey	
	4	Vincent	Bird	
	5	Andrew	Hedgehog	
	6	Francis	Galton	
	7	Andrej	Kolmogorov	
	8	William	Cochran	
	9	Abraham	Wald	
	10	David	Kendall	
	

New Entry Find... Delete Entry Customize Columns... OK Cancel

Click here to add the «ID» column



Step 4: Create the template

Schoolwork

Student's name:

Student's ID:

Exercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight

```
Comment [sar1]: sysuse auto, clear
set seed
sample 7, count
encode make, generate(makeNumeric)
mkmat makeNumeric, matrix(make)
rownames(make)
mkmat price, matrix(price)
mkmat weight, matrix(weight)
```

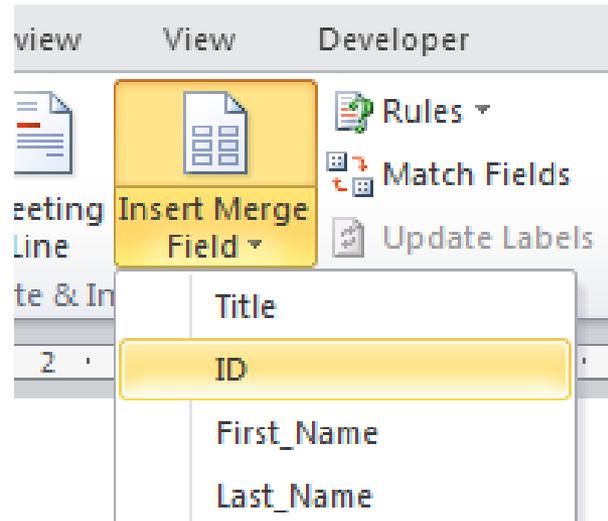
Don't worry about this incomplete command (seed number is missing)

Don't forget to use the «clear» option

```
Comment [sar2]: @matrixrownames
make 2 1
@filltable price 2 2
@filltable weight 2 3
```



Step 5: Insert merge fields



Schoolwork

Student's name: «First_Name»

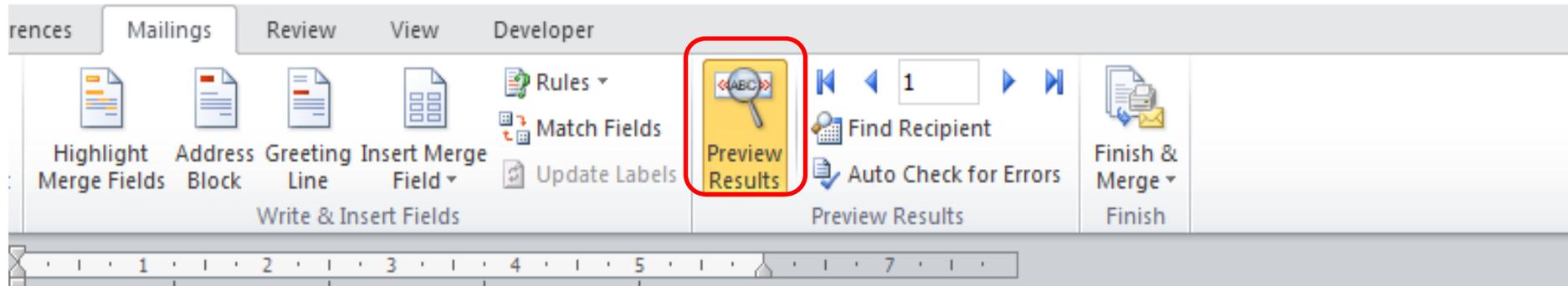
«Last_Name»

Student's ID: «ID»

Comment [sar1]: sysuse auto, clear
 set seed «ID»
 sample 7, count
 encode make, generate(makeNumeric)
 mkmat makeNumeric, matrix(make)
 rownames(make)
 mkmat price, matrix(price)
 mkmat weight, matrix(weight)



Step 6: Preview results



Schoolwork

Student's name: Christine Giraffe

Student's ID: 1

Exercise 1

Calculate the Pearson correlation coefficient between the price and

```
Comment [sar1]: sysuse auto, clear
set seed 1
sample 7, count
encode make, generate(makeNumeric)
mkmat makeNumeric, matrix(make)
rownames(make)
mkmat price, matrix(price)
mkmat weight, matrix(weight)
```



Step 7: Add a page break at the end of the document

Schoolwork

Student's name: Christine Giraffe

Student's ID: 1

Exercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight

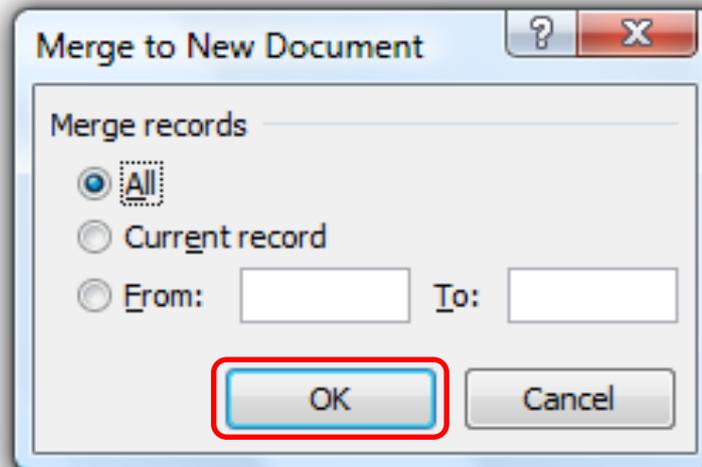
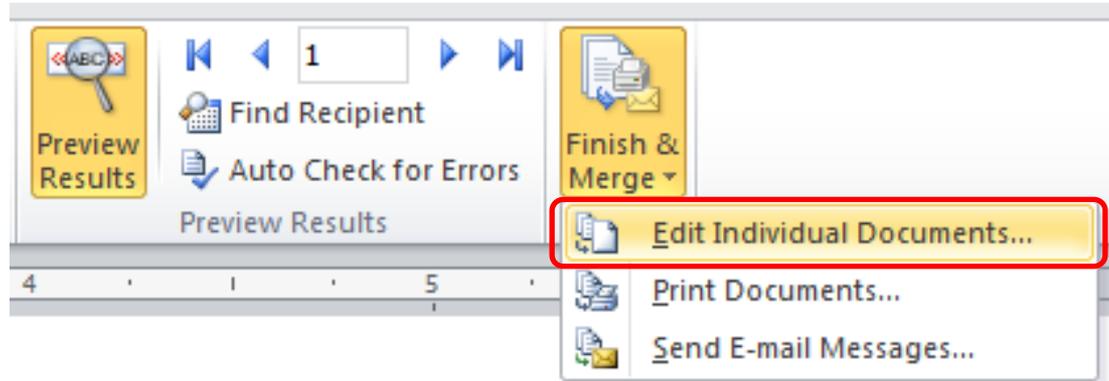
Page Break

```
Comment [sar1]: sysuse auto, clear
set seed 1
sample 7, count
encode make, generate(makeNumeric)
mkmat makeNumeric, matrix(make)
rownames(make)
mkmat price, matrix(price)
mkmat weight, matrix(weight)
```

```
Comment [sar2]: @matrix rownames
make 2 1
@filltable price 2 2
@filltable weight 2 3
```

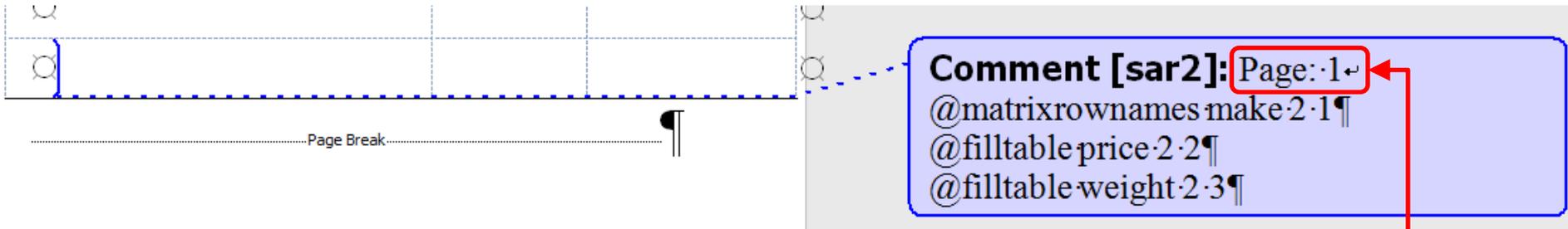


Step 8: Merge to a new document





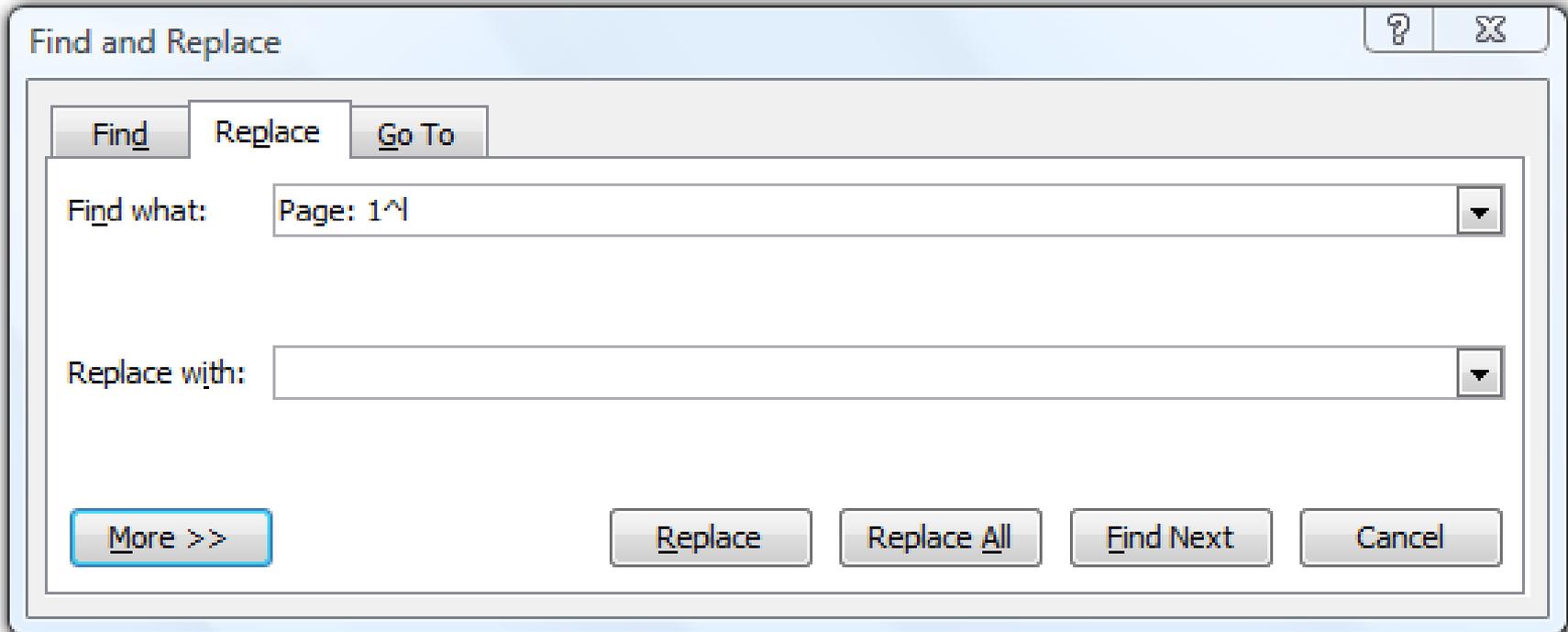
Problem: an artifact is generated by Word in the «mail-merge» document



This artifact is automatically generated by Word in each comment of the new «mail-merge» document. It will generate an error if Sar tries to execute it.



Step 9: Correct the generated by Word artifact in the «mail-merge» document



Notes:

- The string you have to replace is «Page: 1 ^l», where «^l» is a special character for «manual line break»
- Leave the «replace with» field void



Step 10: Launch Sar from the «mail-merge» document and print the schoolwork for your students

Schoolwork

Student's name: Christine Giraffe
Student's ID: 1

Excercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight
Dodge_Colt	3984	2120
Chev__Nova	3955	3430
Chev__Chevette	3299	2110
Linc__Mark_V	13594	4720
Datsun_510	5079	2280
Plym__Horizon	4482	2200
Toyota_Corolla	3748	2200

Every student has got a different schoolwork

Schoolwork

Student's name: John Goat
Student's ID: 2

Excercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight
Volvo_260	11995	3170
VW_Dasher	7140	2160
VW_Scirocco	6850	1990
Olds_Cutl_Supr	5172	3310
Plym__Sapporo	6486	2520
Honda_Civic	4499	1760
AMC_Pacer	4749	3350

Schoolwork

Student's name: Carmen Donkey
Student's ID: 3

Excercise 1

Calculate the Pearson correlation coefficient between the price and the weight of the following 7 cars:

Make	Price	Weight
Chev__Impala	5705	3690
Volvo_260	11995	3170
BMW_320i	9735	2650
Plym__Arrow	4647	3260
AMC_Concord	4099	2930
Audi_5000	9690	2830
Buick_Electra	7827	4080



Step 11: Create a Sar command (in a library) to mark the schoolwork

Plain text file: *c:\sar libraries\checkcompute.txt*

```
@program checkcompute seed
sysuse auto, clear
set seed `seed`
sample 7, count
encode make, generate(makeNumeric)
mkmat makeNumeric, matrix(make) rownames(make)
mkmat price, matrix(price)
mkmat weight, matrix(weight)
generate xy = price * weight
mkmat xy, matrix(xy)
generate xQuad = price ^ 2
mkmat xQuad, matrix(xQuad)
generate yQuad = weight ^ 2
mkmat yQuad, matrix(yQuad)
summarize price
scalar sumPrice = r(sum)
summarize weight
scalar sumWeight = r(sum)
summarize xy
scalar sumXy = r(sum)
summarize xQuad
scalar sumXQuad = r(sum)
summarize yQuad
scalar sumYQuad = r(sum)
correlate price weight
scalar correlation = r(rho)
@end
```

Appendix A – Automatic classwork



Step 12: Create a checker template which is linked to the students' database

Checker

Student's name: «First_Name» «Last_Name»

Student's ID: «ID»

Comment [sar1]: @loadlibrary "c:\sar
libraries\checkcompute.txt"
@do checkcompute «ID»

Make	x_i (Price)	y_i (Weight)	$x_i y_i$	x_i^2	y_i^2
Volvo_260					
VW_Dasher					
VW_Scirocco					
Olds_Cutl_Supr					
Plym_Sapporo					
Honda_Civic					
AMC_Pacer					
Total					

Comment [sar2]: @matrixrownames
make 2 1
@format %14.0f
@filltable price 2 2
@filltable weight 2 3
@filltable xy 2 4
@filltable xQuad 2 5
@filltable yQuad 2 6
@filltable sumPrice 9 2
@filltable sumWeight 9 3
@filltable sumXy 9 4
@filltable sumXQuad 9 5
@filltable sumYQuad 9 6

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} =$$

$$= \frac{7(\infty) - (\infty)(\infty)}{\sqrt{n(\infty) - (\infty)^2} \sqrt{n(\infty) - (\infty)^2}} = x$$

Comment [sar3]: @print sumXy

Comment [sar4]: @print sumPrice

Comment [sar5]: @print sumWeight

Comment [sar6]: @print sumXQuad

Comment [sar7]: @print sumPrice

Comment [sar8]: @print sumYQuad

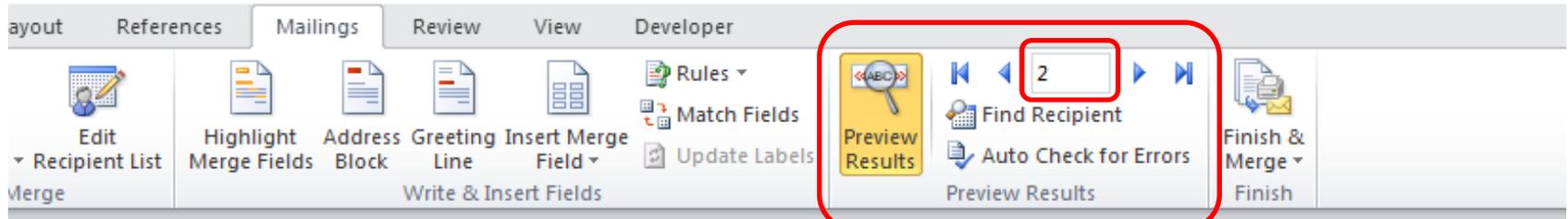
Comment [sar9]: @print sumWeight

Comment [sar10]: @format %5.4f
@print correlation

Appendix A – Automatic classwork



Step 13: Select the student's ID and launch Sar



Checker

Student's name: John Goat

Student's ID: 2

Make	x_i (Price)	y_i (Weight)	$x_i y_i$	x_i^2	y_i^2
Volvo_260	11995	3170	38024152	143880032	10048900
VW_Dasher	7140	2160	15422400	50979600	4665600
VW_Scirocco	6850	1990	13631500	46922500	3960100
Olds_Cutl_Supr	5172	3310	17119320	26749584	10956100
Plym__Sapporo	6486	2520	16344720	42068196	6350400
Honda_Civic	4499	1760	7918240	20241000	3097600
AMC_Pacer	4749	3350	15909150	22553000	11222500
Total	46891	18260	124369482	353393912	50301200

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} =$$

$$= \frac{7(124369482) - (46891)(18260)}{\sqrt{7(353393912) - (46891)^2} \sqrt{7(50301200) - (18260)^2}} = 0.2003$$



Conclusions about *automatic schoolwork*:

- You can discourage students from copying during classwork
- By using «mailings» functions of Word, you can send by e-mail to your students:
 - automatic homework
 - automatic solutions with calculations and formulas
- You can manage exercises in the classroom encouraging students to work on their own



The problem

If the number of observations on which an estimate is based falls below a minimum criterion, a warning should accompany the estimate

X	Y
A	1.3
B	4.5*
C	3.2*

* The value is statistically unreliable given the small sample size

The solution

An automatic document created with Sar where an asterisk is added to the reported estimates which are based on a low number of observations



Example:

A report with mean wage by industry from the nlsw88.dta dataset. We want to highlight estimates which are based on less than **30** observations

```
. sysuse nlsw88
(NLSW, 1988 extract)

. mean wage, over(industry)
(output suppressed)

. matrix N = e(_N)

. matrix list N
```

This matrix contains the number of observations used in estimating mean wages by industry



N[12,1]

	r1	
wage:_subpop_1	17	<30
wage:Mining	4	<30
wage:Construction	29	<30
wage:Manufacturing	367	
wage:_subpop_5	90	
wage:_subpop_6	333	
wage:_subpop_7	192	
wage:_subpop_8	86	
wage:_subpop_9	97	
wage:_subpop_10	17	<30
wage:_subpop_11	824	
wage:_subpop_12	176	



Step 1: create the `genlownumbermatrix` program and put it in a valid ADO path
(See the next slide to know how the `genlownumbermatrix` program works)

```
program genlownumbermatrix
  syntax namelist(min=2 max=2)
  local inputMatrix: word 1 of `namelist'
  local outputMatrix: word 2 of `namelist'
  confirm matrix `inputMatrix'
  local nRowsOfInputMatrix = rowsof(`inputMatrix')
  matrix `outputMatrix' = J(`nRowsOfInputMatrix', 1, .)
  forvalues i = 1/`nRowsOfInputMatrix' {
    if `inputMatrix'[`i', 1] < 30 {
      local rowNames ``rowNames' "*" "'
    }
    else {
      // void row name
      local rowNames ``rowNames' " " "'
    }
  }
  matrix rownames `outputMatrix' = `rowNames'
end
```



How the `genLowNumberMatrix` works

Syntax

```
genLowNumberMatrix inputMatrix outputMatrix
```

Description

The `genLowNumberMatrix` program creates an output matrix in which the matrix row names are asterisks if the corresponding row value in the input matrix is less than 30

Example

```
. matrix N = (12 \ 32 \ 8)
. genLowNumberMatrix N lowN
. matrix list lowN
```

```
lowN[3,1]
      c1
```

```
asterisk → *
void      → .
asterisk → *
          .
```



Step 2: create a Sar automatic report which calls the genlownumbermatrix program

Mean wage by industry

Industry	Wage
Ag/Forestry/Fisheries	5.6*
Mining	15.3*
Construction	7.6*
Manufacturing	7.5
Transport/Comm/Utility	11.4
Wholesale/Retail Trade	6.1
Finance/Ins/Real Estate	9.8
Business/Repair Svc	7.5
Personal Services	4.4
Entertainment/Rec Svc	6.7*
Professional Services	7.9
Public Administration	9.1

(*) number of observation is less than 30

Comment [sar1]: sysuse nlsw88
 mean wage, over(industry)
 matrix wage = e(b)'
 matrix N = e(_N)'
 genlownumbermatrix N lowN

Comment [sar2]: @format %3.1f
 @filltable wage 2 2
 @matrixrownames lowN 2 3



Conclusions about highlighting subgroups with low number of observations:

- Warning about statistics which are based on a small sample is a good practice
- Sar can be used to automate numbers and text as well (asterisks for example)
- You can improve the `genLownumbermatrix` program by:
 - adding an argument which represents the threshold (a fixed threshold of 30 was used in the example)
 - adding an argument which represents an alternative symbol to asterisk



Sar notifies you when an error occurred



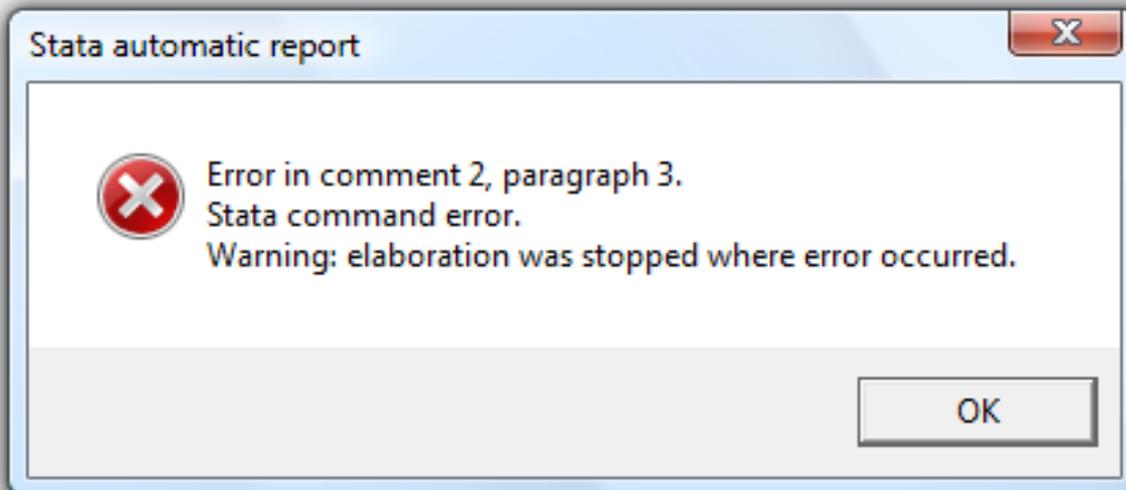


My beautiful report

In this report I comment some summary statistics from the auto.dta dataset.
The mean price is ~~X~~.

Comment [sar1]: sysuse auto

Comment [sar2]: summarize price
@format %5.1f
@printtt r(mean)



The command in the third paragraph of comment 2 is wrong («@printtt» was typed instead of «@print»)



Sar halts the execution of the commands where the error occurs:
you can open the Stata window to debug your Sar session

```
. count
  0
* Sar internally uses the previous command "count" to synchronize with Stata.
* Don't worry about it.
sysuse auto
(1978 Automobile Data)
summarize price
```

Variable	Obs	Mean	Std. Dev.	Min	Max
price	74	6165.257	2949.496	3291	15906

```
confirm numeric format %5.1f
@printtt r(mean)
unrecognized command: @ invalid command name
```



Tip: use `@viewlog` and `@interact` to debug



- **`@viewlog`**
When used (it does not matter in which Sar comment) it leaves the Stata window open after Sar is executed, so you can see the log of your session
- **`@interact`**
It halts Sar execution and makes Stata at your disposal



@beginstring

Syntax

@beginstring *#string#*

Description

The @beginstring command sets the string of characters you want to put before the numerical outputs of the @filltable command.

The string must be specified between two sharps (#).

See also the @endstring command.

@cleartable

Syntax

@cleartable

Description

The @cleartable command clears the table associated with the comment where the command is written. It can only be used within Word comments associated with a single table.

The command has no arguments.

@do

Syntax

@do *SarProgram*

Description

The @do command executes a program previously loaded by the @loadlibrary command or defined in a Word comment through the @program/@end paradigm.

The *SarProgram* argument specifies the program which has to be executed.



@endstring

Syntax

`@endstring #string#`

Description

The `@endstring` command sets the string of characters you want to place after the numerical outputs of the `@filltable` command.

The string must be specified between two sharps (#).

See also the `@beginstring` command syntax and description.

@filltable

Syntax

`@filltable StataData startingRow startingCol [rowStep colStep]`

Description

The `@filltable` command inserts values from matrices in a table, Stata results, scalars and macros given by the `StataData` argument in a Word table. It can be used only in Word comments associated with a single table.

`StataData` is the data retrieved from the Stata environment used by the command to fill the table. It can be a matrix, a Stata result, a scalar or a macro.

`startingRow` and `startingCol` indicate, respectively, the row and the column of the table cell from which `StataData` begins to be printed. They have to be nonzero integers. If these values are negative, -1 means last row/column, -2 means second-last row/second-last column and so on.

`rowStep` and `colStep` indicate, respectively, how many rows (columns) have to be skipped, between a row (column) and the next row (column), filling the table. When `rowStep/colStep` equals 0, no blank row/column is left between printed rows/columns. When `rowStep/colStep` equals 1, a blank row/column is left between printed rows/columns. Generally, if `rowStep/colStep` equals n , then n blank rows/columns are left between printed rows/columns. These arguments are optional and they have to be non-negative integers.



@format

Syntax

```
@format %fmt
```

Description

The @format command sets the numerical format of the output obtained by @print and @filltable commands. The set numerical format is preserved for the following @print and @filltable commands. The *%fmt* argument has to be a numerical format written using the same rules used in the Stata format command (See help format in Stata).

@interact

Syntax

```
@interact
```

Description

The @interact command haltes the execution of Sar to make Stata at your disposal. So you can use Stata, interact with it and create data objects (like scalars or matrices) that will be available in the Sar environment after your Stata session has been closed. Remember to not manually close the Stata window: this will cause the crash of Sar. You have to return to Word, where you will find a dialog window with a button to close Stata. The command has no arguments.

@loadlibrary

Syntax

```
@loadlibrary "pathOfTheLibraryFile"
```

Description

The @loadlibrary command loads programs defined in a Sar library file. The path of the Sar library file has to be specified in the *pathOfTheLibraryFile* argument.



@matrixcolnames and @matrixrownames

Syntax

```
@matrixcolnames StataMatrix startingRow startingCol [colStep]  
@matrixrownames StataMatrix startingRow startingCol [rowStep]
```

Description

The @matrixcolnames and @matrixrownames commands fill a Word table with, respectively, row-names and column-names of a Stata matrix. They can be used only in Word comments associated with a single table.

StataData is the matrix retrieved from the Stata environment whose matrix row-names are printed by @matrixrownames and whose matrix column-names are printed by @matrixcolnames. This argument has to be a matrix.

startingRow and *startingCol* indicate, respectively, the row and the column of the table cell from which the row-names/column-names of *StataMatrix* begin to be printed. They have to be nonzero integers. If these values are negative, -1 will indicate the last row/column, -2 will indicate the second-last row/second-last column and so on.

colStep is an optional argument for @matrixcolnames. It indicates the column step according to the table is filled. The default value is 0. It has to be a non-negative integer.

rowStep is an optional argument of @matrixrownames. It indicate the row step according to the table is filled. The default value is 0. It has to be a non-negative integer.

@print

Syntax

```
@print StataValue
```

Description

The @print command, launched from a Word comment associated with a portion of text (a temporary text placeholder in the Sar jargon), replaces its placeholder with the value of a Stata result, a scalar or a macro retrieved from the Stata environment. The @print command can not be used in a Word comment associated with a table.

The *StataValue* argument must be a Stata result, a scalar or a macro.



@program/@end paradigm

Syntax

```
@program programName [arg1 arg2 ... argN]  
[...]  
[Sar and Stata commands]  
[...]  
@end
```

Description

The @program/@end paradigm is used to define a Sar program. This paradigm can be used in a Word comment or in a Sar library. Sar programs are, roughly speaking, a list of Sar and Stata commands. This list of commands is defined between the @program and the @do commands. After the commands are loaded in the Sar environment, they can be executed through the @do command.

The *programName* argument is used to set the name of the program.

The optional arguments *arg1*, *arg2*, ..., *argN* specify the arguments of the program defined by the @program/@end paradigm. When you want to use the values passed as arguments in your program, you have to use the *§arg1§*, *§arg2§*, ..., *§argN§* callbacks inside your program code: before executing the program Sar replaces every callback with the corresponding values of arguments.

The @end command closes a program definition. It has no arguments.

The following commands can not be used in a Sar program: @do, @loadlibrary, @interact and the @program/@end paradigm.



@resetstring

Syntax

```
@resetstring
```

Description

The @resetstring command sets to an empty string the string of characters which is putted before and after the numerical outputs of the @print and @filltable commands: when the @resetstring command is used no characters are added before or after the numerical output. It's equivalent to the couple of command @beginstring ## and @endstring ##.

The command has no arguments.

See also @beginstring and @endstring syntax and description.

@viewlog

Syntax

```
@viewlog
```

Description

The @viewlog command asks Sar to leave the Stata window open after the Sar macro was executed. This can be useful to look at the log created by Stata computations. When @viewlog is used, in whatever word comment, a dialog window is opened after the execution of the Sar macro, allowing you to close the Stata Window and terminate the Sar macro.

The command has no arguments.



The user is the only responsible for the accuracy of the statistical analysis and for possible damages caused by Sar. It's strongly recommended to save the Word document before you launch Sar.