经济学研究方法

Lecture 1: Introduction to STATA

Zhiyong Huang

Outline

- A review of data management & analysis: tools and ethics
- STATA: Basic commands and syntax
- File management (do files, log files, data files, etc)
- Data Manipulation I
- Self-learning resources

Data Management Tools

- Database
  - Excel
  - Access
  - …
- Data management & analysis
  - STATA
  - SPSS
  - R
  - SAS
  - Eview
  - Splus
  - Python
  - …

Data Management

- Readable
- Reproducible
- Interoperability

More coding tips

- Organized
- Indenting
- Notes

Installing Stata

- Windows
- IOS

Types of Files

- Data files

xx.dta

auto.dta

Stata can also read other data types

xx.txt

xx.csv

- Do files

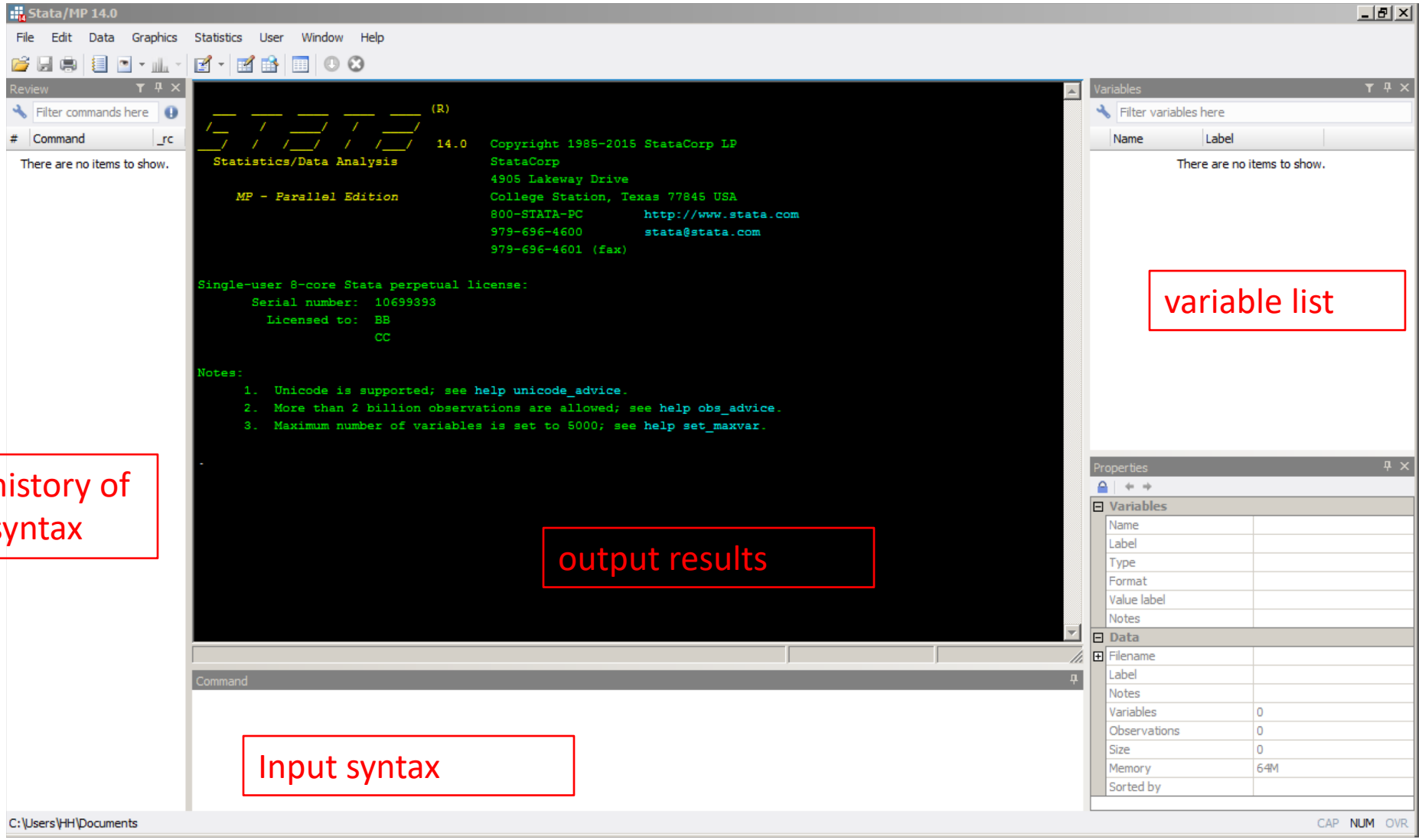
xx.do

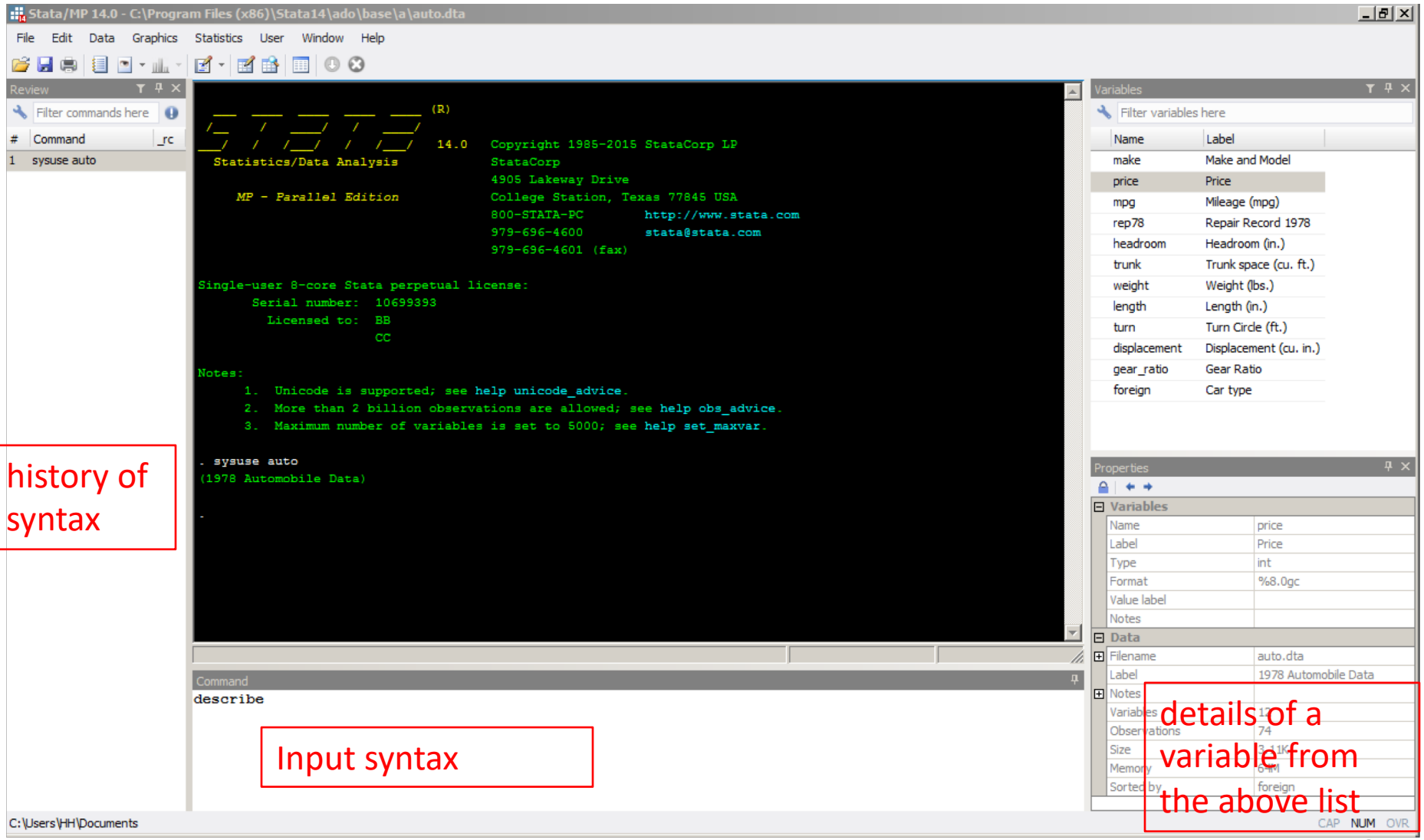
- Results files

xx.log

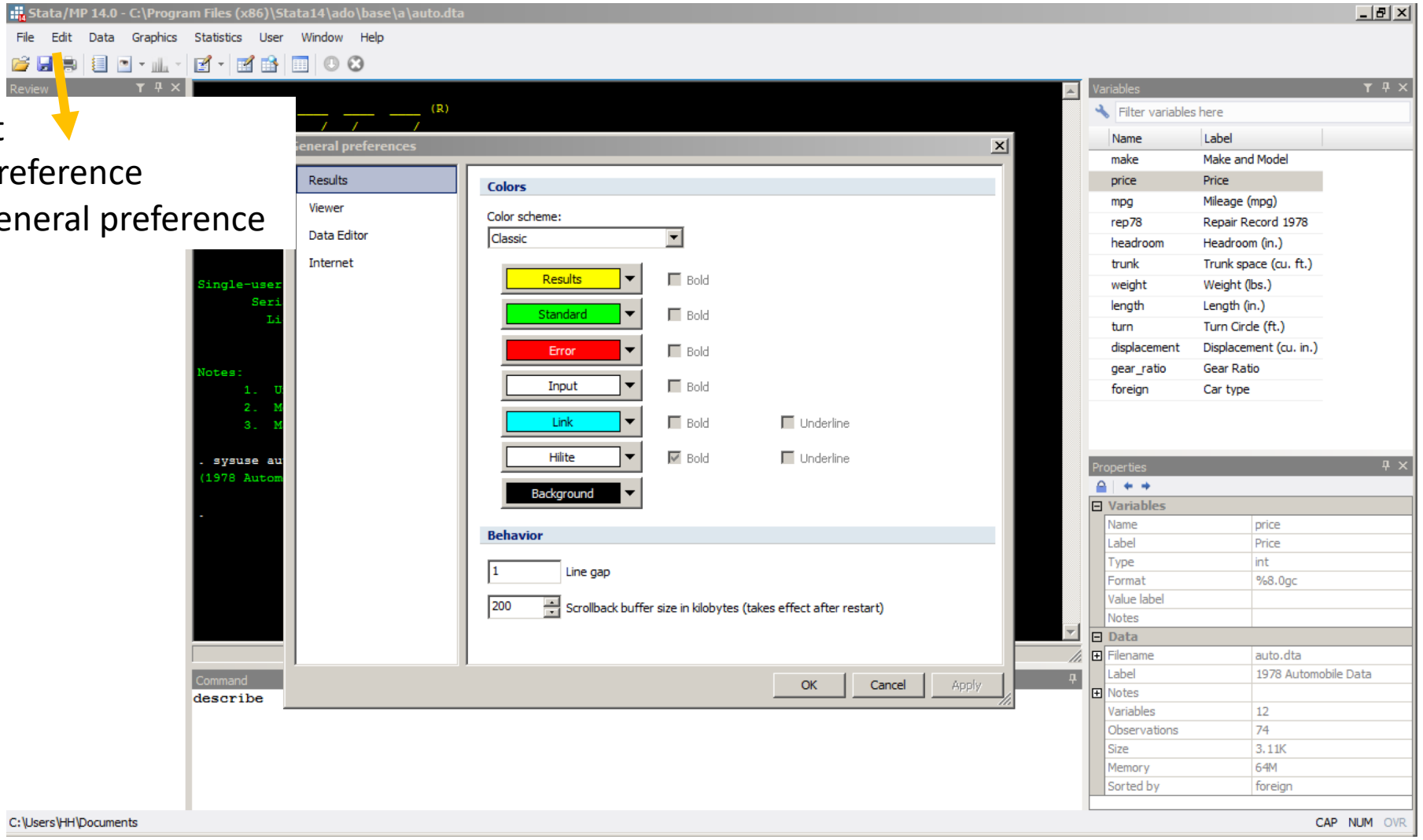
xx.smcl

Open Stata

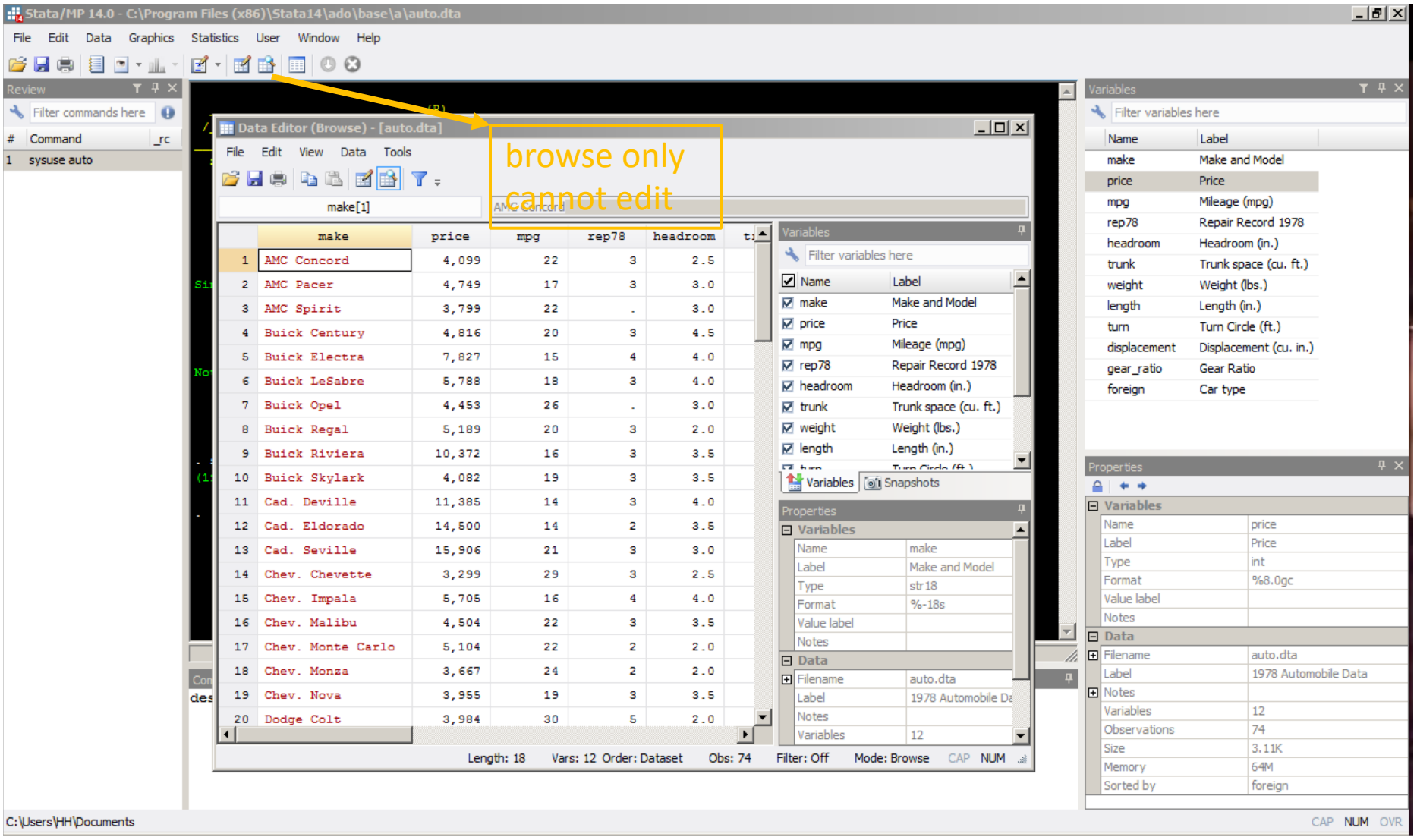

Stata/MP 14.0
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Command _rc
There are no items to show.
(R)
14.0 Copyright 1985-2015 StataCorp LP
Statistics/Data Analysis StataCorp
4905 Lakeway Drive
MP - Parallel Edition College Station, Texas 77845 USA
800-STATA-PC http://www.stata.com
979-696-4600 stata@stata.com
979-696-4601 (fax)
Single-user 8-core Stata perpetual license:
Serial number: 10699393
Licensed to: BB
CC
Notes:
1. Unicode is supported; see help unicode_advice.
2. More than 2 billion observations are allowed; see help obs_advice.
3. Maximum number of variables is set to 5000; see help set_maxvar.
Variables
Filter variables here
Name Label
There are no items to show.
variable list
history of syntax
output results
Command
Input syntax
Properties
Variables
Name
Label
Type
Format
Value label
Notes
Data
Filename
Label
Notes
Variables 0
Observations 0
Size 0
Memory 64M
Sorted by
C:\Users\HH\Documents
CAP NUM OVR

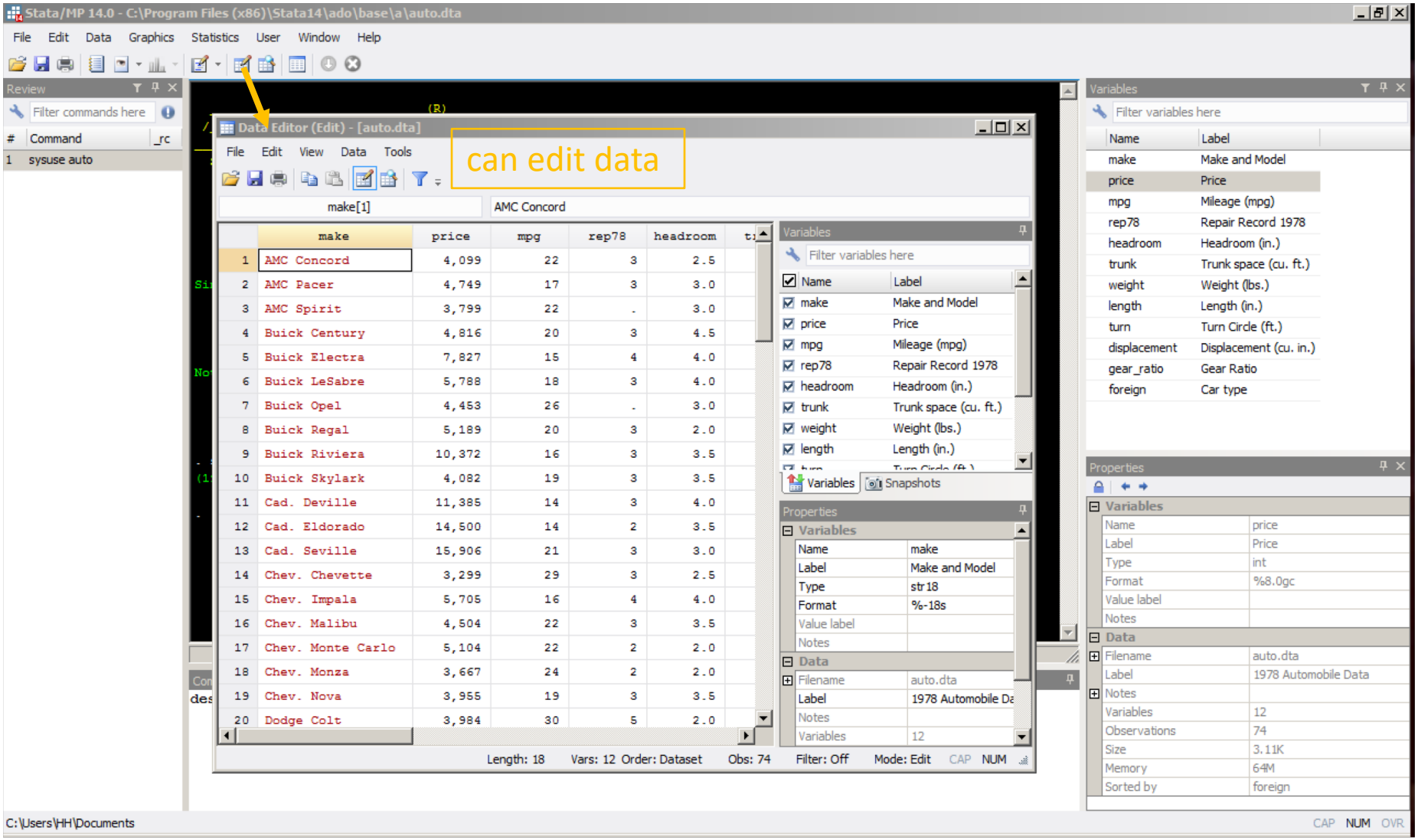

Stata/MP 14.0 - C:\Program Files (x86)\Stata14\ado\base\a\auto.dta
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Command _rc
1 sysuse auto
(R)
14.0 Copyright 1985-2015 StataCorp LP
Statistics/Data Analysis StataCorp
4905 Lakeway Drive
MP - Parallel Edition College Station, Texas 77845 USA
800-STATA-PC http://www.stata.com
979-696-4600 stata@stata.com
979-696-4601 (fax)
Single-user 8-core Stata perpetual license:
Serial number: 10699393
Licensed to: BB
CC
Notes:
1. Unicode is supported; see help unicode_advice.
2. More than 2 billion observations are allowed; see help obs_advice.
3. Maximum number of variables is set to 5000; see help set_maxvar.
. sysuse auto
(1978 Automobile Data)
.
Variables
Filter variables here
Name Label
make Make and Model
price Price
mpg Mileage (mpg)
rep78 Repair Record 1978
headroom Headroom (in.)
trunk Trunk space (cu. ft.)
weight Weight (lbs.)
length Length (in.)
turn Turn Circle (ft.)
displacement Displacement (cu. in.)
gear_ratio Gear Ratio
foreign Car type
Properties
Variables
Name price
Label Price
Type int
Format %8.0gc
Value label
Notes
Data
Filename auto.dta
Label 1978 Automobile Data
Notes
Variables
Observations 74
Size
Memory
Sorted by foreign
Command
describe
history of syntax
Input syntax
details of a variable from the above list
C:\Users\HH\Documents
CAP NUM OVR

Change the Color Settings

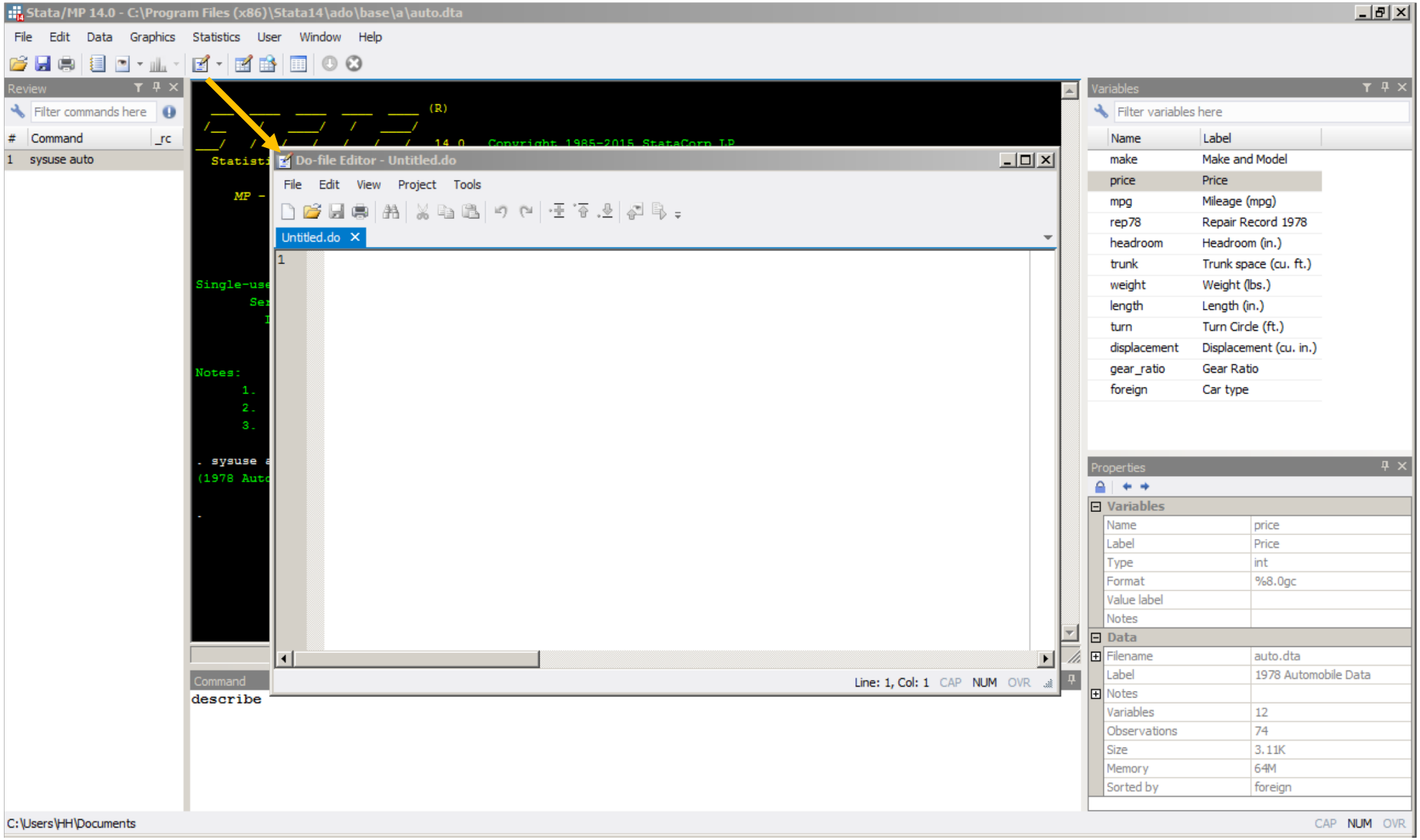


Where's the data




Stata/MP 14.0 - C:\Program Files (x86)\Stata14\ado\base\a\auto.dta
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Command _rc
1 sysuse auto
Data Editor (Edit) - [auto.dta]
File Edit View Data Tools
can edit data
make[1]
AMC Concord
make price mpg rep78 headroom
1 AMC Concord 4,099 22 3 2.5
2 AMC Pacer 4,749 17 3 3.0
3 AMC Spirit 3,799 22 . 3.0
4 Buick Century 4,816 20 3 4.5
5 Buick Electra 7,827 15 4 4.0
6 Buick LeSabre 5,788 18 3 4.0
7 Buick Opel 4,453 26 . 3.0
8 Buick Regal 5,189 20 3 2.0
9 Buick Riviera 10,372 16 3 3.5
10 Buick Skylark 4,082 19 3 3.5
11 Cad. Deville 11,385 14 3 4.0
12 Cad. Eldorado 14,500 14 2 3.5
13 Cad. Seville 15,906 21 3 3.0
14 Chev. Chevette 3,299 29 3 2.5
15 Chev. Impala 5,705 16 4 4.0
16 Chev. Malibu 4,504 22 3 3.5
17 Chev. Monte Carlo 5,104 22 2 2.0
18 Chev. Monza 3,667 24 2 2.0
19 Chev. Nova 3,955 19 3 3.5
20 Dodge Colt 3,984 30 5 2.0
Length: 18 Vars: 12 Order: Dataset Obs: 74 Filter: Off Mode: Edit CAP NUM
Variables
Filter variables here
Name Label
make Make and Model
price Price
mpg Mileage (mpg)
rep78 Repair Record 1978
headroom Headroom (in.)
trunk Trunk space (cu. ft.)
weight Weight (lbs.)
length Length (in.)
Variables Snapshots
Properties
Variables
Name make
Label Make and Model
Type str18
Format %-18s
Value label
Notes
Data
Filename auto.dta
Label 1978 Automobile Da
Notes
Variables 12
Variables
Filter variables here
Name Label
make Make and Model
price Price
mpg Mileage (mpg)
rep78 Repair Record 1978
headroom Headroom (in.)
trunk Trunk space (cu. ft.)
weight Weight (lbs.)
displacement Displacement (cu. in.)
gear_ratio Gear Ratio
foreign Car type
Properties
Variables
Name price
Label Price
Type int
Format %8.0gc
Value label
Notes
Data
Filename auto.dta
Label 1978 Automobile Data
Notes
Variables 12
Observations 74
Size 3.11K
Memory 64M
Sorted by foreign
C:\Users\HH\Documents
CAP NUM OVR

Do File


Stata/MP 14.0 - C:\Program Files (x86)\Stata14\ado\base\a\auto.dta
File Edit Data Graphics Statistics User Window Help
Review
Filter commands here
Command _rc
1 sysuse auto
Do-file Editor - Untitled.do
File Edit View Project Tools
Untitled.do
Line: 1, Col: 1 CAP NUM OVR
Command
describe
Variables
Filter variables here
Name Label
make Make and Model
price Price
mpg Mileage (mpg)
rep78 Repair Record 1978
headroom Headroom (in.)
trunk Trunk space (cu. ft.)
weight Weight (lbs.)
length Length (in.)
turn Turn Circle (ft.)
displacement Displacement (cu. in.)
gear_ratio Gear Ratio
foreign Car type
Properties
Variables
Name price
Label Price
Type int
Format %8.0gc
Value label
Notes
Data
Filename auto.dta
Label 1978 Automobile Data
Notes
Variables 12
Observations 74
Size 3.11K
Memory 64M
Sorted by foreign
C:\Users\HH\Documents
CAP NUM OVR

Some Basic Codes

use

- `use` -- Load Stata dataset

- Load Stata-format dataset

```
use filename [, clear nolabel]

use auto.dta,clear

use http://www.stata-press.com/data/r14/auto

sysuse auto, clear
```

clear

- `clear`- Clear memory,

```
clear //clear all variables & labels

clear [ mata | results | matrix | programs | ado ]

clear [ all | * ]
```

cd

- `cd` -- Change directory

- Stata for Windows:

```
. cd // to know the current directory
. cd \data\city // change the directory
. cd d:
```

- Stata for Mac and Stata for Unix:

```
. pwd
. cd ~/data/city
```

log

- log -- Echo copy of session to file

- Report status of log file

```
log
log query [logname | _all]
```

- Open log file

```
log using filename [, append replace [text|smcl]
name(logname)]
```

- Close log

```
log close [logname | _all]
```

- Temporarily suspend logging or resume logging

```
log {off|on} [logname]
```

describe / des

- `describe` -- Describe data in memory or in file

- Describe all variables in the dataset

```
describe
des
d
des, short
```

- Describe some variables

```
des make price
```

summarize / sum

- summarize -- Summary statistics

```
summarize
sum
su

summarize mpg weigh
summarize mpg weigh, detail

summarize rep78
summarize i.rep78
```

edit / browse

- edit -- Browse or edit data with Data Editor
- Edit using Data Editor

edit [varlist] [if] [in] [, nolabel]
ed

- Browse using Data Editor

browse [varlist] [if] [in] [, nolabel]
brow
br

list

- list -- List values of variables

list
li
l

list in 1/10
list mpg weight
list mpg weight in 1/20

codebook

- codebook -- Describe data contents

- Display codebook for all variables in dataset

```
. codebook
. codebook _all
```

- Same as above command, but print dataset name, date last saved, dataset Display codebook for rep78 variable

```
. codebook rep78
```

- Display codebook for rep78 variable, including notes attached to rep78

```
. codebook rep78, notes
```

save

- save -- Save Stata dataset

save newfile.dta, replace

Be cautious of using replace, always save a raw file

saveold newfile

saveold newfile, version(12)

help

- help -- Display help in Stata
- Most useful resource of learning!!

help sum

help des

help list

Add Notes

- 1: an asterisk,

*, at the beginning of a line

- 2: slash + *,

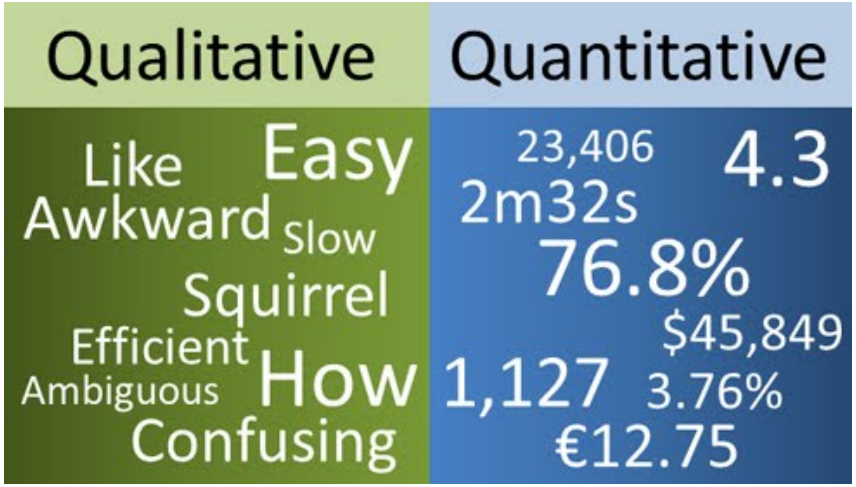
/* … */, for a paragraph

- 3: double slashes,

//, at the beginning of a line or the end of a line

Data

- Definition from Wiki
  - **Data** (数据) is a set of values of qualitative or quantitative variables (变量).
  - Structured vs unstructured



https://sites.google.com/a/staff.lisd.net/mr-nic-s---8th-grade-pre-ap-science/unit-1---forces-motion/qualitative-data-vs-quantitative-data

Variable

- Definitions: from Wiki
  - Variable and attribute (research)
    - In science and research, attribute (属性) is a characteristic of an object (person, thing, etc.). A variable is a logical set of attributes. Variables can "vary" - for example, be high or low. (Babbie, 2009)
    - the variable is the operationalized (操作化) way in which the attribute is represented for further data processing.
  - Variable (computer science)
    - In computer programming, a variable or scalar is a storage location paired with an associated symbolic name (an identifier), which contains some known or unknown quantity of information referred to as a value.

Data Types: Quantitative-1

- Continuous variables (连续变量)
  - Can assume any value in some (possibly unbounded) interval of real numbers.
  - Interval variables (narrow)，等距数据
    - has the same unit, but no absolute zero
    - e.g., temperature, IQ
  - Ratio variables，比率数据
    - Amount & unit, and absolute zero
    - e.g., length, weight,
  - Usually belong to measurement data (测量数据)

Data Types: Quantitative-2

- Discrete variables (离散变量)
  - Assume only isolated values
  - Ranked (ordinal) variables, 顺序/定序变量
    - Not measured, no natural ordering, no same units, no absolute zero
    - e.g., quality rank, satisfaction, school rank
  - Categorical (nominal) variables,分类/称名变量
    - To notify different groups on one attribute by integers
    - e.g., gender/sex, color, ethnicity, vote
  - What about count data (计数数据)?
    - e.g., population number

Data Types: Qualitative

- Written document (texts文字, string字母串)
  - Some structure
    - Survey: Generic & general name of drug, other(specify)
    - Records: Prescription, diagnosis
  - No structure, usually exiting :
    - newspapers, magazines, books, websites, memos, annual reports
- In-Depth Interviews
  - individual interviews (e.g., one-on-one) as well as "group" interviews (including focus groups).
  - stenography, audio recording, video recording or written notes.
- Direct Observation
  - same ways as interviews (stenography, audio, video) and through pictures, photos or drawings (e.g., those courtroom drawings of witnesses are a form of direct observation).

http://www.socialresearchmethods.net/kb/qualdata.php

Display

- display -- Display strings and values of scalar expressions

- As a hand calculator:

```
. display 2 + 2
```

- As might be used in do-files and programs:

```
. sysuse auto
. summarize mpg
. display "mean of mpg = " as result
r(mean)
. disp "age" _col(20) r(mean)
. disp "age" _col(40) r(mean)
```

Operators操作符

Syntax

Arithmetic		Logical		Relational (numeric and string)	
+	addition	&	and	>	greater than
-	subtraction	|	or	<	less than
*	multiplication	!	not	>=	> or equal
/	division	~	not	<=	< or equal
^	power			==	equal
-	negation			!=	not equal
+	string concatenation			~=	not equal

- A double equal sign (==) is used for equality testing.
- The order of evaluation (from first to last) of all operators is ! (or ~), ^, - (negation), /, *, - (subtraction), +, != (or ~=), >, <, <=, >=, ==, &, and |.

Operators操作符

- Another
  `if`
- Examples

```
. sysuse auto
. generate weight2 = weight^2
. count if rep78 > 4
. count if rep78 > 4 & weight < 3000
. list make if rep78 == 5 | mpg > 25
. list make if rep78 != 1 | mpg <=50
```

.do files

- Simple .do file:

```
log using class2.log
use auto.dta

log close
```

- *Every* line of code necessary for a project should appear in a .do file

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course // optional
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

.do 文件包含的内容

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

[everything else]

log close
```

Other things to include in .do files

```
capture log close
cd C:\data\stata_course
log using class2.log
//The purpose of this .do file is...

version 14
clear all
macro drop _all
set more off
set linesize 255

[everything else]

log close
```

Other things to include in .do files

```
capture log close
log using class2.log
//The purpose of this .do file is...

version 14
clear all
macro drop _all
set more off
set linesize 255

[everything else]

log close
```

Other things to include in .do files

```
capture log close
log using class2.log
//The purpose of this .do file is...

version 14
clear all
macro drop _all
set more off
set linesize 255

[everything else]

log close
```

Other things to include in .do files

```
capture log close
log using class2.log
//The purpose of this .do file is...

version 14
clear all
macro drop _all
set more off // set more off, permanently
set linesize 255

[everything else]

log close
```

Other things to include in .do files

```
capture log close
log using class2.log
//The purpose of this .do file is...

version 14
clear all
macro drop _all
set more off
set linesize 255


[everything else]

log close
```

More Loading & Saving Files

Loading a file into memory

More complex:

```
use [file] [if] [in]
use [varlist] [if] [in] using [file]
```

Loading a file into memory

Examples:

```
use age bmi using transplants
use transplants in 1/50, clear
use transplants in 44
use transplants if age>50
use a* p* using transplants in 1/100
```

usespss

- usespss -- Use SPSS dataset

. usespss using "myfile.sav",

. usespss using "myfile.sav", clear

. usespss using "myfile.sav", clear saving("myfile.dta")

import delimited

```
copy http://www.stata.com/examples/auto.csv auto.csv
import delimited auto
import delimited mydata.txt, delimiters(";")
```

Other commands for importing data

See help files for details, or `help import` for an overview.

- `infile`
- `infix`
- `import excel using file.xlsx`
- `import sasxport using file.sas7bdat`

…

save

Save the data in memory

- `save newfile`
- `save newfile, replace`
- `save, replace  //careful!`
- `saveold auto_v12,ver(12)`

Other commands for exporting data

See help files for details, or `help export` for an overview.

- `export excel using file.xlsx`
- `export sasxport using file.sas7bdat`

…

exit

`exit` -- Exit Stata

```
exit
```

```
exit, clear
```

Manipulating Data I

generate / gen

generate -- Create or change contents of variable

Simple:

```
gen price_center=price-6165.257
gen cheap=(price<6165.257)
gen cheap=1 if price<6165.257
  replace cheap=0 if price>=6165.257
list
gen lightcheap=(price<4000)&(weight<3000)
```

A word on variable types

Variables have a *type* – one of the following:

- byte: integer range -127 to 100
- int: integer range to -32767 to 32740
- long: integer range +/- 2 billion
- float: decimal, range +/- 10^{38}
- double: decimal, range +/- 10^{307}, more precise
- strings, e.g. str5 = string of length 5

generate / gen

- A bit more complex:
- `gen byte price=(price<6000)`
- `gen cheap = price if rep78==5`
- `gen price_spline=(price>6000)*(age-6000)`

_n and _N

- _n represents the current record number. So 1 for the first record in memory, 2 for the second record, etc.
- _N represents the number of records in the dataset
- `gen new_id = _n`
- `gen total_records = _N`
- `gen percentile = 100*_n/_N`

`drop / keep` (1)

Delete some variables from the database

- `keep price rep78 mpg*`
- `drop foreign`

Remember: to load the original dataset again, do

`sysuse auto, clear`

drop / keep (2)

Delete some observations from the database

- `keep in 1/100`
- `drop if price < 6000`
- `keep if rep78==4`
- `drop if price<5000 & rep78==4`

load the original dataset

```
use auto, clear
```

replace

Change the value of a variable

- `replace price=0`
- `replace rep78=2 if rep78>2`
- `replace price=price+1`
- `replace price = . if !inrange(price, 5000,6000)`

load the original dataset

```
use auto, clear
```

rename

Change the value of a variable

- `rename price price_old`

`sort` and `gsort`

Sort the records

- `sort price`
- `sort light weight`

gsort: allows sorting in descending order

- `gsort price`
- `gsort light weight`

recode

Do a bunch of replacing at once

- `recode rep78(1 2 3 4 = 0)`
- `recode price (0/6165=1)(6166/16000=0), gen(cheap1)`
- `recode cheap1 (0=1) (1=0), gen(cheap2)`
- `recode price (0/6165=1)(6166/15000=0), gen(cheap3)`

Labels

Give descriptive names to variables, and category values

```
describe price
label var price “Car price"
```

Labels

Give descriptive names to variables, and category values

```
label var cheap “车价分类”
tab cheap //is 1 cheap or not cheap?
label define cheap_la 0 "0=not cheap“ ///
                      1 "1=cheap"
label values cheap cheap_la
tab cheap //much better
tab cheap , nolabel //don't show label
```

`preserve` and `restore`

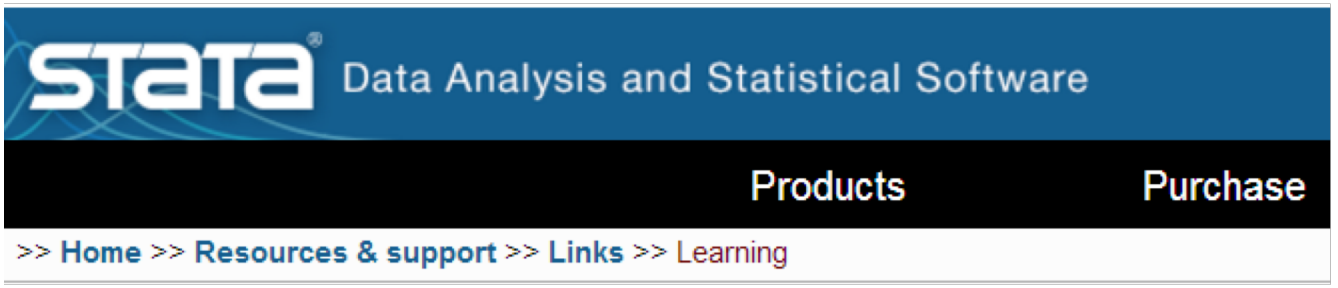
If you type `preserve` and later type `restore`, the data will go back to the way it was when you typed `preserve`

```
sum price //mean = 6165 usd
preserve
drop if price < 6165
sum price //mean = 9656 usd
restore
sum price //mean = 6165 usd
```

Other Self-Learning Resources

- http://www.ats.ucla.edu/stata/

- http://www.stata.com/links/resources-for-learning-stata/



- http://bbs.pinggu.org/

  经管之家