Stata 简明讲义

王 非
中国经济研究中心
ebwf@163.com

〇、写在前面的话

关于学习 Stata 的意义，大家只需知道：目前，Stata 是计量经济学、特别是微观计量经济学的主流软件。因此，Stata 很重要、很有用，而大家也会在使用 Stata 的过程中慢慢体会到它的特点。

本讲义取名为“Stata 简明讲义”，意在突出“简”和“明”两个字。虽然讲义长达五十多页，但相比 Stata 的完全手册来说，还不及九牛之一毛，故为“简”。实际上，完全手册中的很多内容都鲜有人（特别是计量经济学者）问津，而本讲义列出的内容则是大家经常用到的操作；所以，“简”也有“简”的好处。即便如此，掌握这份讲义也并非易事。所谓“明”，是明晰的意思。本讲义本着“手把手教”的精神，力求把每项操作都说得具体明晰，以方便初学者（特别是没有程序操作经历的初学者）尽快上手。至于本讲义在“简明”上做得怎么样，还需要各位读者来评判。

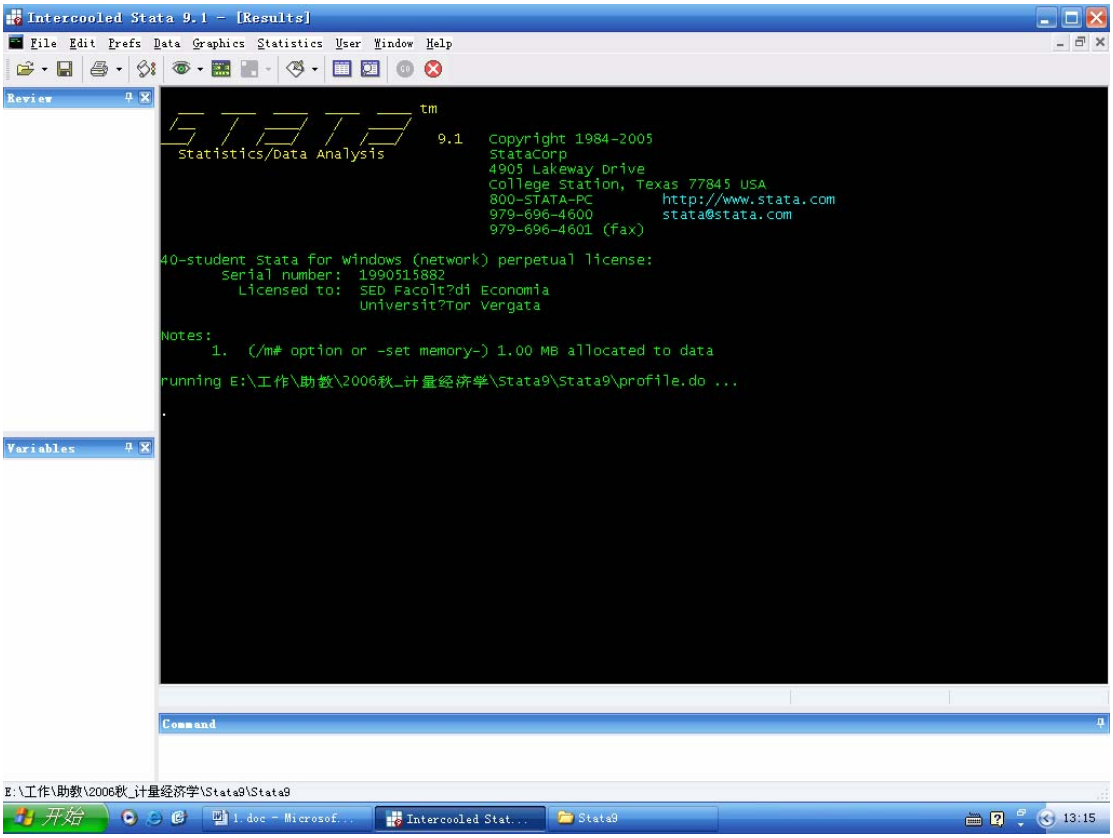
中心的一位学长邹传伟，曾经写过一份“Stata 介绍”，在网上可以下载。那份讲义比较全面，但不够具体明晰。本讲义参照那份讲义，在框架上查漏补缺，并进一步地明晰化。本讲义第二部分的“do 文件”和第七部分的“残差分析”的相关内容均来自于中心的沈艳老师的相关讲义，而沈老师对于本讲义的成形给予了细致的指导。本讲义附带了一些数据文件，其中“WAGE1.dta”和“WAGEPRC.dta”均来自 Wooldridge 的中级计量教材的数据集，而其他数据则为作者自己的杜撰。尽管从别人那里拿来了许多好东西，但本讲义的任何错误仍源于作者自己的疏忽。

本讲义是这样安排的：第一部分讲 Stata 的界面，第二部分讲 do 文件，第三部分讲怎样把数据导入 Stata，第四部分专门讲 help 和 search 命令以及帮助文件的阅读方法，第五部分讲数据的描述及管理，第六部分讲如何画图，第七部分讲初步的回归分析。

祝各位学习愉快。

一、Stata 长什么样？

首先，让我们看看 Stata 长什么样。我们以 Stata 9.1（以下简称 Stata）为例。点击可执行文件“wstata.exe”，即可看到 Stata 的基本界面：



中间黑色背景的区域就是 Stata 的基本显示界面，数据分析的结果一般显示在这一区域中。在我看来，黑色的背景有助于减缓视觉疲劳。如果你不喜欢这种显示方式，可以把鼠标放在这一区域中，点击鼠标右键，进而设定自己喜欢的显示方式。

围绕着黑色区域，有三个白色背景的框，左边两个，下边一个。左下角的框的标

题是“Variables”，这里会显示数据中所有变量的名称。下边的框的标题是“Command”，这里用来输入各种操作命令，命令操作的结果一般会显示在黑色区域中。左上角的框的标题是“Review”，这里会显示你曾经操作过的所有命令。在三个框的右上角，均有这样一个小图标：  。点击它会使框隐藏起来，其效果类似于把 QQ 拖到屏幕的边上；再次点击会恢复原状。而点击右上角另外一个图标  ，会把相应的框关掉；如果想再次打开，可以点击菜单栏的“Window”，并选择相应的框。

黑色屏幕上方的菜单栏和图标栏，下文会逐步涉及。

二、良好的习惯从 do 文件开始

上文提到，Stata 的 Command 框可以输入各种操作命令。实际上，绝大多数初学者（甚至很多长时间使用 Stata 的人）都只是通过 Command 框一条一条地输入命令，边走边看。这种做法的缺点在于：进行命令操作的时候具有盲目性，不易厘清自己将要做什么；而命令操作过后缺乏系统性，忘了自己曾经做过什么，而且别人也无法了解你的操作过程；另外，如果你想再次进行类似的操作，得重新输入曾经运行过的命令，比较繁琐。因此，大家最好从一开始就养成一个良好的习惯：在进行任何程序操作之前，都要事先写好完整的操作计划书；这一操作计划书在 Stata 里叫做 do 文件，而 Stata 会自动运行 do 文件中计划好的所有操作。

下面是一个 do 文件的一部分（选自沈艳老师的相关讲义），我们借此看一下 do 文件是个怎样的东西。

```
/* regreport.do
 * The purpose of this file is to fully understand the machanism of OLS*/

set more off
cap log close

log using "D:\teaching\Ieconometrics\fal2006\homework\regreport.log", replace
cd D:\teaching\Ieconometrics\fal2006\data
use wage1.dta, clear
reg  wage educ exper tenure
/* generate variables equaling the estimated coefficients for future use*/
gen eeduc=_b[educ]
gen eexper=_b[exper]
gen etenure=_b[tenure]

/* generate the total sum of square */
egen mwage=mean(wage)
sum mwage
egen sst=sum((wage-mwage)^2)
disp sst

/* generate the explained sum of square */
predict yhat, xb
egen sse=sum((yhat-mwage)^2)
disp sse

gen ssr=sst-sse
disp ssr

/* mean square error, sigmahat square*/
gen n=526
gen k=3
gen sigmahat2=ssr/(n-k-1)
disp sigmahat2
gen rmse = sqrt(sigmahat2)
disp rmse
```

第一行是这个 do 文件的名称，do 文件的后缀名是“do”。第二行是这个 do 文件的作用，即你要通过这份操作计划书做什么事情。这两行不是操作的内容，而是对操作的注释。在 do 文件中，注释的部分用“/*”和“*/”包裹起来。有编程经验的人都知道，注释在程序里非常重要。从上面的 do 文件可以看出，注释不仅出现在开头，而且出现在每一段命令之前。注释虽然不直接参与程序的运行，但却可以帮助你清晰地规划将要做的事情，也可以帮助你在事后准确地回忆起曾经做过的事情，还可以帮助他人较快地读懂你的操作计划。一个好的注释必须简洁、清晰，能用短短几个词就准确地描述你要做的事情。

接下来，是“set more off”命令。在程序的运行过程中，如果显示结果很长（如一屏显示不完），屏幕下方就会出现“—more—”的标记；这时，Stata 会暂停显示；只有按任意键，结果才能继续显示下去。而“set more off”就是把“—more—”

标记去掉的命令；这样，do 文件在运行的过程中就不会因为某项操作的显示结果太长而暂停运行。

接下来，是“cap log close”命令。要明白这条命令，得先明白什么是 log 文件。打个比方，开大会的时候，需要录像机全程录像，以备事后查用。同样，运行操作程序的时候，也需要全程记录所有的操作命令和操作结果，以备事后查用。log 文件就是 Stata 中的录像带，用来忠实记录整个操作过程。如果准备拍新录像的时候，发现一盘旧录像带还在录像机里放着，那么就要先取出旧录像带，以便放入新录像带。同样，如果在进行新的程序操作之前，Stata 还在运行着某个以前的 log 文件，那么就得先把这个 log 文件关掉，以便开启新的 log 文件进行记录。“log close”就是关闭旧的 log 文件的命令。在“log close”前面加“cap”的原因是：如果原来没有 log 文件在运行，那么运行“log close”命令会产生错误信息，Stata 也就会因此中止 do 文件的运行；而前面的“cap”可以阻止在“log close”命令运行过程中的任何错误信息的出现，从而保证 do 文件的运行不会因此中止。

接下来便是开启新的 log 文件的命令。“log using”命令用来开启一个 log 文件，命令后面是 log 文件的路径和名称。值得强调的是，Stata 命令中出现文件的路径和名称时，可以用双引号包裹起来，也可以不用。命令最后“replace”的意思是：如果在那个路径下已经存在一个这样的文件，那么新开启的 log 文件把原文件完全覆盖；如果你想在已经存在的 log 文件后面继续记录，可以把“replace”换成“append”。log 文件的操作还有其他常用命令。直接运行“log”命令可以查询当前 log 文件的工作状态；“log off”命令可以暂停 log 文件的运行，就像把录像机暂时关掉；“log on”可以重新开始 log 文件的运行，就像重新开动录像机；如果想查看 log 文件记录的内容，可以在“view”命令后面加上 log 文件的路径和名称。此外，点击图标栏的  图标，也可以对 log 文件进行一系列的操作。

接下来就是导入数据、进行各种操作了。后文会逐步介绍相关的内容。

那么，怎样写这样一个 do 文件呢？主要有两种方法：一、打开一个记事本文件，

直接在里面编辑，编辑好之后另存为后缀名为“do”的文件即可。二、用 Stata 自身附带的 do 文件的编辑器进行编辑。在 Command 框中输入“doed”，就可以打开 do 文件编辑器。如果想编辑已经存在的 do 文件，需要在“doed”后面加上 do 文件的路径和名称。另外，也可以直接点击 Stata 图标栏里的  图标来编辑 do 文件。编辑 do 文件的过程中，别忘了点击编辑器图标栏上的  图标来保存编辑的成果。

如果用 do 文件编辑器编辑 do 文件，可以点击编辑器图标栏里的  图标来试运行 do 文件（也可以运行“run”命令加 do 文件的路径和名称）。试运行只会反馈 do 文件中的错误，而不会显示 do 文件的运行结果，这便于对 do 文件的调试。当 do 文件顺利通过试运行之后，便可以点击编辑器图标栏里的  来正式运行（也可以运行“do”命令加 do 文件的路径和名称）。正式运行会显示所有的运行结果。此外，还可以通过 Stata 菜单栏中的“File → Do…”来运行一个 do 文件。

上面所讲的内容恐怕不易在短时间内被 Stata 的初学者（尤其是没有程序操作经历的初学者）完全接受。但是大家应该试着从一开始就养成写 do 文件的好习惯，并在实践的过程中慢慢体会 do 文件的好处及其所涉及的各种操作。

三、怎样把数据导入 Stata?

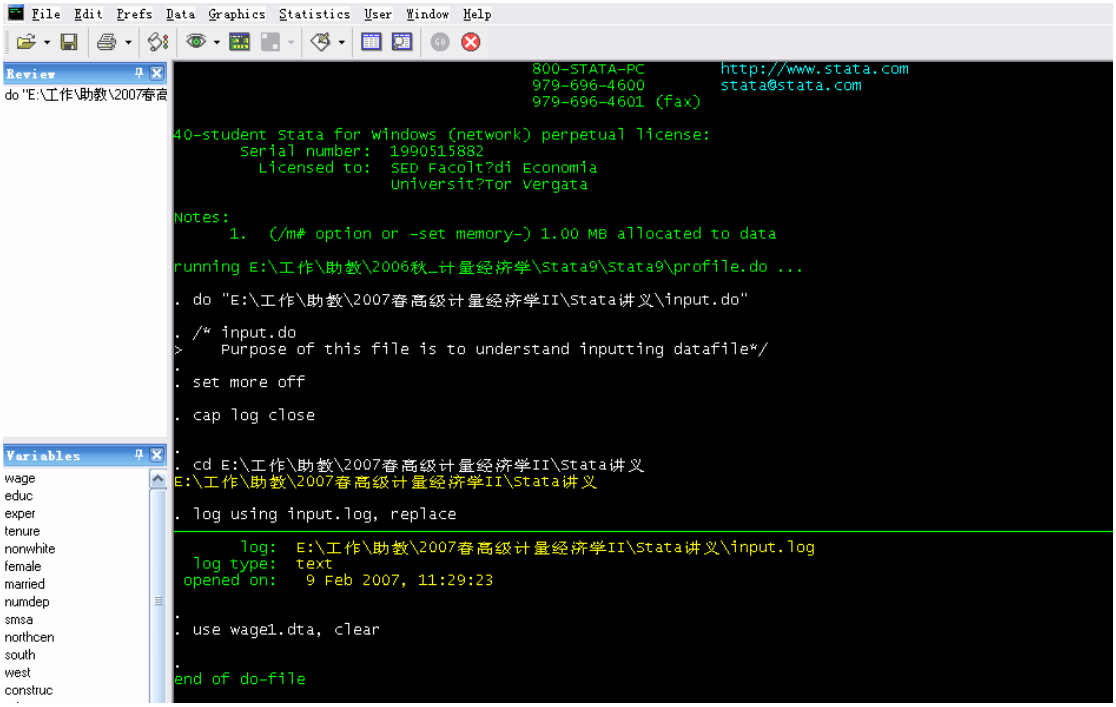
想炒菜，得先把菜倒进锅里；同样，用 Stata 分析数据，得先把数据导入 Stata。

Stata 默认的数据文件是后缀名为“dta”的文件。讲义附带的一个数据文件是“WAGE1.dta”，它可以直接用 Stata 打开。打开的方式无非有以下几种：1、运行“use”命令加数据的路径和名称。2、像上文中列出的 do 文件那样，先用“cd”命令进入数据所在的目录，然后用“use”命令直接加数据的名称来导入数据。当要操作的许多的数据文件都在同一个目录下面时，用这样的方法导入数据比较

方便——导入新数据时，只需改变“use”命令后的文件名即可，而不需改变数据的路径。3、在菜单栏中选择“File → Open”，并选择数据所在的路径；4、点击图标栏的  ，并选择数据所在的路径。

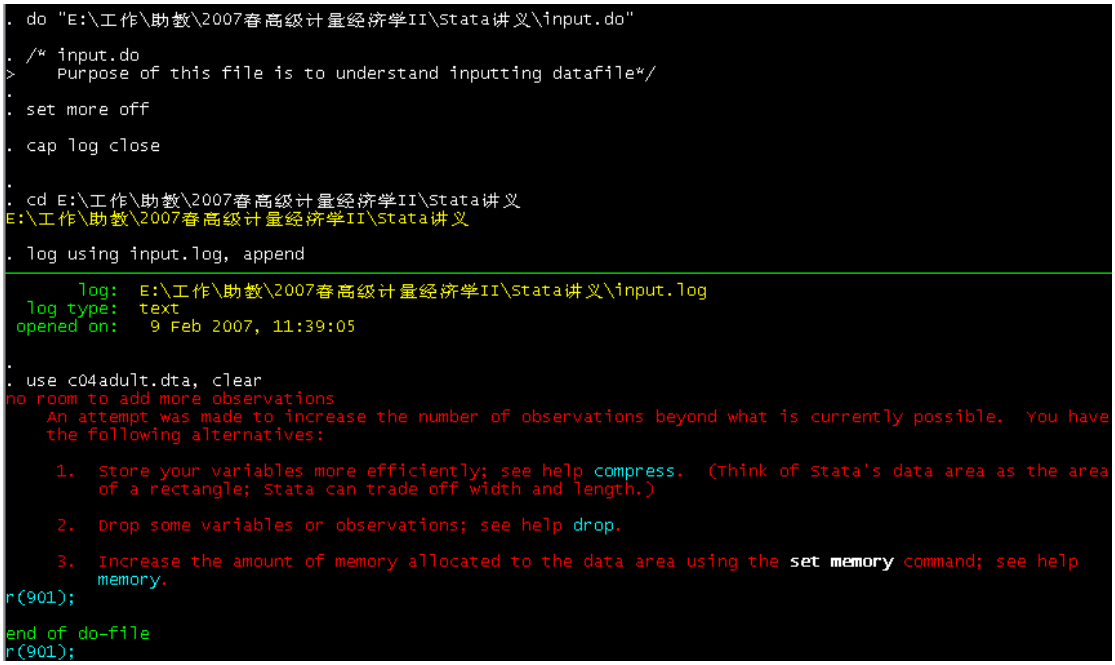
在前两种方式中，“use”命令后面，往往需要加一个“clear”。打个比方，想炒一锅新菜，得把原来的一锅菜倒出去；同样，想导入一个新数据，得把原来的数据清理出 Stata。“clear”就是把原来的数据清理出 Stata 的命令。

考虑到编辑 do 文件的需要，大家应掌握前两种数据导入方式。通过一个 do 文件导入数据后，会看到如下的界面：



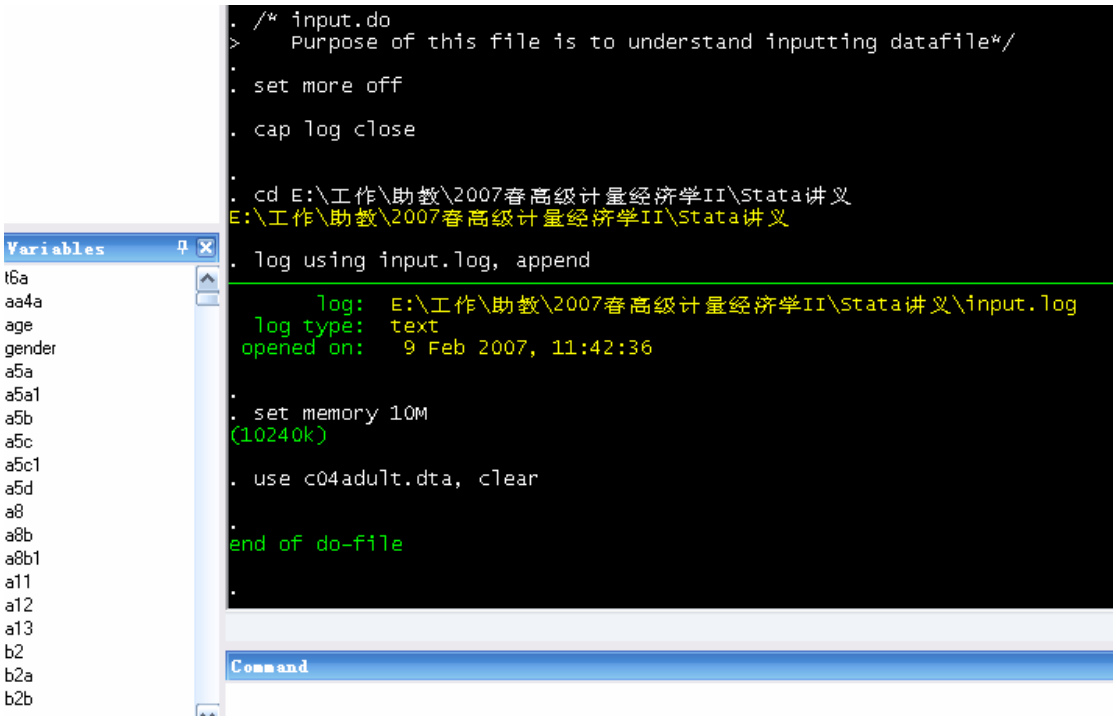
我们看到，黑色区域显示出了 do 文件的所有操作及其结果。Review 框中显示出了曾经运行过的命令（打开 do 文件的命令）；如果你想再次操作曾经操作过的命令，直接双击 Review 框中的相应命令即可，就不必重新输入一遍了。Variable 框中显示的就是“WAGE1.dta”所包含的变量，有工资（wage），教育水平（educ）和工作经验（exper）等。

但是，当按照上述方法打开另一个数据文件“c04adult.dta”时（该文件为中国健康和营养调查的 2004 年的成人数据。因为太大，没有和讲义附在一起），显示界面中出现了红色的错误信息：



这几行提示告诉我们，没有足够的空间容纳数据；此外，还给出了三种可行的方案：1、更有效地存储和压缩数据，2、删掉某些变量或观测值，3、增大分配给该数据的空间。一般来说，我们选择第 3 种方案。毕竟，许多人不愿意“委屈”菜的质量和分量，那就换口大锅吧。

“换大锅”的命令是：set memory xxM。其中的“xx”为一个数字，代表分配给数据多大的空间；“M”为存储容量的单位，即兆字节。Stata 默认的分配空间是 1M。接下来，我分配给这个大数据 10M 的空间。分配完毕后，就可以顺利打开数据了：

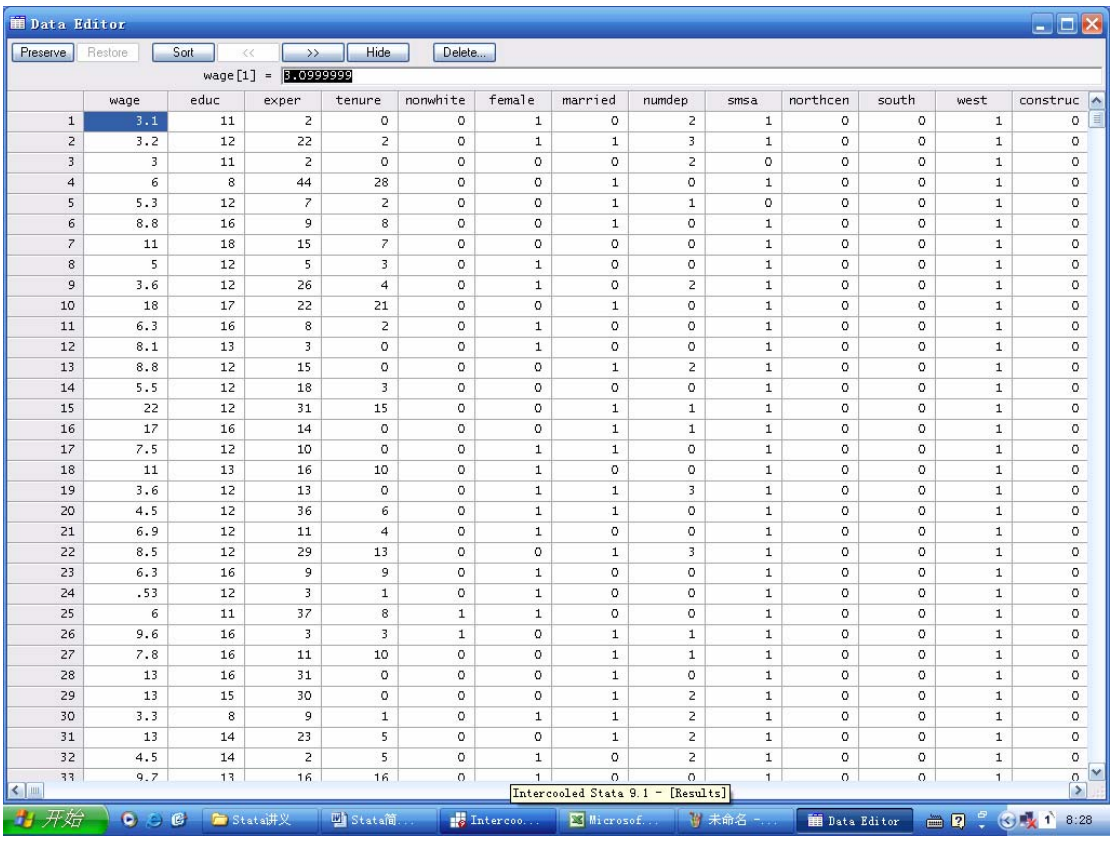


到此为止，我们已经知道怎样把 dta 格式的数据文件导入 Stata 了。但是，对于许多不是 dta 格式的数据文件，我们应该怎么办？

对于后缀名是“raw”，“txt”和“csv”的数据，Stata 可以用“insheet using”命令导入。这个命令的用法和“use”类似。对于那些可以另存为这三种格式的数据，可以先把它们转化成这三种格式，然后用“insheet using”命令导入。讲义附带着一个 Excel 文件“wage1_sub.xls”，是“WAGE1.dta”的子样本。打开这个 Excel 文件，另存为“txt”（制表符分隔的文本文件）或“csv”格式后，可以用“insheet using”命令导入。一般来说，在导入非“dta”格式的数据后，要把这些数据另存为“dta”格式。因此，在 do 文件的“insheet using”命令的下一行，最好紧跟“save”命令。一般的命令是“save *.dta”，“*”代表另存为的文件名。如果不加“.dta”，Stata 默认的存储格式为“dta”。此外，如果想保存对数据的任何改动，也要使用“save”命令。

当源数据是 Excel 数据的时候，我们还可以将其直接复制粘贴到 Stata 里。以“WAGE1_sub.xls”为例：1、打开“WAGE1_sub.xls”，用鼠标选定包括变量名

在内的全部数据内容，并复制；2、用 clear 命令清空 Stata 内的原有数据，然后点击菜单栏中的图标，将光标放在左上角的那个格子中，然后粘贴。操作完毕后，我们会看到：



	wage	educ	exper	tenure	nonwhite	female	married	numdep	smsa	northcen	south	west	construc
1	3.1	11	2	0	0	1	0	2	1	0	0	1	0
2	3.2	12	22	2	0	1	1	3	1	0	0	1	0
3	3	11	2	0	0	0	0	2	0	0	0	1	0
4	6	8	44	28	0	0	1	0	1	0	0	1	0
5	5.3	12	7	2	0	0	1	1	0	0	0	1	0
6	8.8	16	9	8	0	0	1	0	1	0	0	1	0
7	11	18	15	7	0	0	0	0	1	0	0	1	0
8	5	12	5	3	0	1	0	0	1	0	0	1	0
9	3.6	12	26	4	0	1	0	2	1	0	0	1	0
10	18	17	22	21	0	0	1	0	1	0	0	1	0
11	6.3	16	8	2	0	1	0	0	1	0	0	1	0
12	8.1	13	3	0	0	1	0	0	1	0	0	1	0
13	8.8	12	15	0	0	0	1	2	1	0	0	1	0
14	5.5	12	18	3	0	0	0	0	1	0	0	1	0
15	22	12	31	15	0	0	1	1	1	0	0	1	0
16	17	16	14	0	0	0	1	1	1	0	0	1	0
17	7.5	12	10	0	0	1	1	0	1	0	0	1	0
18	11	13	16	10	0	1	0	0	1	0	0	1	0
19	3.6	12	13	0	0	1	1	3	1	0	0	1	0
20	4.5	12	36	6	0	1	1	0	1	0	0	1	0
21	6.9	12	11	4	0	1	0	0	1	0	0	1	0
22	8.5	12	29	13	0	0	1	3	1	0	0	1	0
23	6.3	16	9	9	0	1	0	0	1	0	0	1	0
24	.53	12	3	1	0	1	0	0	1	0	0	1	0
25	6	11	37	8	1	1	0	0	1	0	0	1	0
26	9.6	16	3	3	1	0	1	1	1	0	0	1	0
27	7.8	16	11	10	0	0	1	1	1	0	0	1	0
28	13	16	31	0	0	0	1	0	1	0	0	1	0
29	13	15	30	0	0	0	1	2	1	0	0	1	0
30	3.3	8	9	1	0	1	1	2	1	0	0	1	0
31	13	14	23	5	0	0	1	2	1	0	0	1	0
32	4.5	14	2	5	0	1	0	2	1	0	0	1	0
33	9.7	13	16	16	0	1	0	0	1	0	0	1	0

这个大框是数据编辑框，除了点击图标之外，还可以直接输入 edit 命令打开。中间的白色区域就是源文件中的所有数据，每一行为一个观测值（一个人），每一列为一个变量；白色区域左边是观测值的序号，从 1 到 100；白色区域的上边是变量名称。我们发现，在复制粘贴的过程中，原来 Excel 文件中的变量名称自动跑到了数据编辑框中它应当所在的位置。要达到这样的效果，变量名就不能用汉字表示。

此外，还可以用专门的数据格式转化软件（比如 StatTransfer）将其他格式的数据转化成可以直接导入 Stata 的格式。StatTransfer 可以转化的数据格式有许多：Access 数据、ASCII 数据、Excel 数据、Gauss 数据、Matlab 数据、Minitab 数据、SAS 数据、SPSS 数据、Statistica 数据和 Stata 数据等，基本上涵盖了常见的数据

格式。该软件的使用比较简单，这里就不再赘述了。大家可以试着将“WAGE1_sub.xls”转化成 dta 格式。

最后，如果要一条一条地录入数据，可以直接在数据编辑框中录入（但我建议先用 Excel 等比较方便的软件录入，然后再导入 Stata）。

在数据导入的过程中，还可能出现其他问题，这需要大家开动脑筋，灵活解决。在数据导入之后，请大家务必要对照一下源数据，看看导入的数据和源数据是否一致，是否出现了诸如串行或串列的问题。

四、Stata 命令中的倚天剑和屠龙刀

把数据导入 Stata 之后，接下来就是复杂的数据分析和处理工作了。这一工作需要浩如烟海、变化多端的命令，而这往往使初学者畏而却步。其实，毫不夸张地说，只要掌握了 Stata 命令中的倚天剑和屠龙刀，就可以畅行江湖了。这两条最重要的命令是：help 和 search。

当你确切地知道某个命令的名称，但却不太清楚它的用法时，就用 help 命令；当你只是模糊地知道某个命令时，就用 search 命令。下面以简单的线性回归命令为例。

最简单的线性回归命令是 regress。如果你知道 regress 这个命令，但是不知道它的用法，可以输入如下命令：help regress。然后，Stata 会跳出一个框，对这个命令进行详细说明（即该命令的帮助文件）。如果你不知道这个命令，但是你知道你想做 OLS，可以输入如下命令：search ols。然后，Stata 会给出它认为和“ols”这个关键词比较相关的命令及简要的说明：

```
. search ols

Keyword search

        Keywords:  ols
          Search:  (1) Official help files, FAQs, Examples, SJs, and STBs

Search of official help files, FAQs, Examples, SJs, and STBs


[R]     anova . . . . . . . . . . . . . .  Analysis of variance and covariance
        (help anova)

[R]     areg  . . . . . . . .  Linear regression with a large dummy-variable set
        (help areg)

[R]     cnsreg  . . . . . . . . . . . . . . . .  Constrained linear regression
        (help cnsreg)

[R]     eivreg  . . . . . . . . . . . . . . . . Errors-in-variables regression
        (help eivreg)

[R]     loneway . . . . . Large one-way ANOVA, random effects, and reliability
        (help loneway)

[R]     mvreg . . . . . . . . . . . . . . . . . . . .  Multivariate regression
        (help mvreg)

[R]     regress . . . . . . . . . . . . . . . . . . . . . .  Linear regression
        (help regress)

[R]     regress postestimation  . . . . . . . Postestimation tools for regress
        (help regress postestimation)

[R]     sureg . . . . . . . . . . . . . Zellner's seemingly unrelated regression
        (help sureg)
```

我们发现，其中对 regress 命令的简要说明是“Linear regression”，比较符合我们的需要。用鼠标点击蓝色的命令，就可以得到该命令的帮助文件。寻找命令需要很多搜索技巧，大家要灵活掌握。

拿到了倚天剑和屠龙刀，但是不会玩，也是白搭。同样，找到了需要的命令，但是看不懂帮助文件，也是没用。下面详细解说一下 regress 命令的帮助文件，以求大家对帮助文件有些感觉。

打开 regress 的帮助文件，我们看到：

```
help regress                                          dialog:  regress
                                                    also see:  regress postestimation
                                                               regress postestimation ts
```

Title

[R] regress — Linear regression

Syntax

regress *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	description
Model	
noconstant	suppress constant term
hascons	has user-supplied constant
tsscons	compute total sum of squares with constant; seldom used
SE/Robust	
vce(*vcetype***)**	*vcetype* may be **robust**, **bootstrap**, or **jackknife**
robust	synonym for **vce(robust)**
cluster(*varname***)**	adjust standard errors for intragroup correlation
mse1	force mean squared error to 1
hc2	use u^2_j/(1-h_jj) as observation's variance
hc3	use u^2_j/(1-h_jj)^2 as observation's variance
Reporting	
level(#)	set confidence level; default is **level(95)**
beta	report standardized beta coefficients
eform(*string***)**	report exponentiated coefficients and label as *string*
noheader	suppress the table header
plus	make table extendable
depname(*varname***)**	substitute dependent variable name; programmer's option

depvar and *indepvars* may contain time-series operators; see varlist.
bootstrap, **by**, **jackknife**, **rolling**, **statsby**, **stepwise**, **svy**, and **xi** are allowed; see prefix.
aweights, **fweights**, **iweights**, and **pweights** are allowed; see weight.
See regress postestimation for additional capabilities and estimation commands.

右上角有一行是“dialog：regress”。用鼠标点击后，跳出一个对话框，根据对话框的提示，可以完成线性回归的操作。前文提到过，Stata 中许多常见的操作，既可以用程序命令来实现，又可以通过菜单栏和图标栏来实现（如 use、edit 和 regress 等）。使用对话框等工具，固然会比较简便，但却无法用 do 文件进行统一操作，也无法留下操作记录。因此，建议大家努力掌握相关命令的语法，即使它非常复杂。

右上角还有一个“also see”。在这里面，Stata 列出了它认为的和 regress 命令比较相关的其他操作命令。此外，在帮助文件的末尾，有“also see”的更详细的列表。

往下走，是 Title 和 Syntax。Title 就是命令的名称及简要说明。Syntax 是帮助文

件的核心内容，它简要介绍了命令的结构和选项。Syntax 的第一行是命令的结构。“regress”表示回归的命令，“depvar”表示回归的因变量，“indepvars”表示回归的自变量，“if”表示回归的条件，“in”表示回归的范围，“weight”表示回归的权重，“options”表示回归的选项。“regress”的前三个字母的下方有一条线，表示在输入 regress 这个命令的时候，只要输入“reg”即可。帮助文件中其他命令或选项的下划线均表示这种含义。命令结构中斜体的部分表示在实际操作中，大家需要将其替换成相应的内容（如将 depvar 替换成真正的因变量）；而粗体且非斜体的部分（比如 regress）表示在实际操作中，大家得将其照搬。帮助文件中的斜体或粗体均表示这种含义。用方括号括起来的部分在实际操作中可有可无，没有括起来的部分必须出现在命令中（比如，reg 命令至少要包含 reg 和 depvar 两项。如果是“reg depvar”，就表示拿因变量对常数项作回归）。所有蓝色的部分均可用鼠标点击，从而出现进一步的说明。此外，不要忘了“options”之前还得加个逗号。

命令结构的下面，是命令的选项，即列出什么东西可以作为“options”。选项有三类。第一类是“Model”，第二类是“SE/Robust”，第三类是“Reporting”。“Model”表示需要怎样的模型，比如要不要常数项；“SE/Robust”表示需要怎样的标准误，比如要不要稳健的标准误；“Reporting”表示需要报告怎样的结果，比如需要报告怎样的置信区间。在选项的右侧，有相应的简要说明。如果需要在命令中出现多个选项（如既不要常数项，又要报告稳健标准误），则选项之间用空格间隔。

在选项说明的下方，有几行字，是 regress 命令的补充说明：比如在该命令中，可以加前缀，变量可以包含滞后算子等。

到此为止，大家就把 regress 这个命令的“浓缩版”帮助文件看完了。后面还有三大部分：“Description”，“Options”和“Examples”。 这三部分是对命令的进一步说明。“Description”详细说明了该命令是干什么的，“Options”对命令选项作了更细致的说明，而“Examples”给出了许多应用该命令的例子。如果大家没有看懂前面“浓缩版”的帮助文件，就需要沉住气，慢慢地品味后面三部分的内

容。

以“WAGE1.dta”为例，运行如下的命令：

```
reg wage educ exper if female= =1 in 1/100 [aweight=educ], noc l(90)
```

可以得到：

```
. reg wage educ exper if female==1 in 1/100 [aweight=educ], noc l(90)
(sum of wgt is   6.5500e+02)

      Source |       SS       df       MS              Number of obs =      52
-------------+------------------------------           F(  2,    50) =   78.76
       Model |  1731.96407     2  865.982037            Prob > F      =  0.0000
    Residual |  549.726881    50  10.9945376            R-squared     =  0.7591
-------------+------------------------------           Adj R-squared =  0.7494
       Total |  2281.69096    52  43.8786722            Root MSE      =  3.3158

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .4315302   .0513144     8.41   0.000     .3455321    .5175283
       exper |   .0125829   .0406296     0.31   0.758    -.0555085    .0806743
------------------------------------------------------------------------------
```

这条命令是拿工资对教育水平和工作经验作回归，回归的样本限定为序号从 1 到 100 中的女性；考虑到可能存在的异方差，用教育水平作权重；此外，回归不包括常数项，且需要报告 90%水平上的置信区间。至于回归结果，后文再进行解释。需要注意的是，条件语句中的等号为两个“=”，赋值语句中的等号为一个“=”，这符合一般的编程原则。

其他的运算符号还有：大于（>），小于（<），大于或等于（>=），小于或等于（<=），不等于（!=），和（&），或（|）等。圆括号可以改变运算的先后顺序。详情请参照 operator 的帮助文件。

下文提到的所有命令，大家都可以通过帮助文件来掌握它的详细语法。值得强调的是，看帮助文件、自学命令的使用方法是项辛苦的工作，需要大家的信心、耐心和细心。

四、“看”数据与“管”数据

现在，我们正式开始数据分析和处理的万里长征。第一步应该怎么走？许多人一拿到数据，就迫不及待地做回归——就如同猪八戒吃人参果，不观其色，不嗅其香，不品其味，不求其疵，吃到肚子里都不知怎么回事。倘若是好果子，这么吃有些可惜；倘若是坏果子，这么吃有伤身体。同样，一个数据，有它的魅力，也有它的瑕疵；在回归之前，最好用各种方法“看”数据：看它的基本含义、基本特征乃至基本规律，从而掌握数据的基本面貌。“看”完数据之后，难免会发现一些需要修剪的地方（或者是数据本身存在一些不足，或者是数据的状态不符合自己的分析习惯等），这就需要对数据进行初步的管理。“看”数据才能知道如何“管”数据，而“管”数据是为了更好地“看”数据，两者浑然一体。所以，下文将“看”数据和“管”数据的操作穿插在一起，并以“WAGE1.dta”为例进行说明。

首先，使用“list”等命令可以浏览整个数据。此外，点击菜单栏的  图标（或运行“browse”），也可以看到整个数据。数据的界面和上文提到的点击  图标（或运行“edit”）所得到的界面类似；不同之处在于，前者只能浏览数据，而后者可以编辑数据。

其次，你需要了解各个变量的特征和含义。运行“describe”命令，你可以看到：

```
. describe

Contains data from E:\工作\助教\2007春高级计量经济学II\Stata讲义\WAGE1.DTA
  obs:           526
 vars:            24                          16 Sep 1996 15:52
 size:        18,936 (98.2% of memory free)

              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
wage            float  %8.2g                  average hourly earnings
educ            byte   %8.0g                  years of education
exper           byte   %8.0g                  years potential experience
tenure          byte   %8.0g                  years with current employer
nonwhite        byte   %8.0g                  =1 if nonwhite
female          byte   %8.0g                  =1 if female
married         byte   %8.0g                  =1 if married
numdep          byte   %8.0g                  number of dependents
smsa            byte   %8.0g                  =1 if live in SMSA
northcen        byte   %8.0g                  =1 if live in north central U.S
south           byte   %8.0g                  =1 if live in southern region
west            byte   %8.0g                  =1 if live in western region
construc        byte   %8.0g                  =1 if work in construc. indus.
ndurman         byte   %8.0g                  =1 if in nondur. manuf. indus.
trcommpu        byte   %8.0g                  =1 if in trans, commun, pub ut
trade           byte   %8.0g                  =1 if in wholesale or retail
services        byte   %8.0g                  =1 if in services indus.
profserv        byte   %8.0g                  =1 if in prof. serv. indus.
profocc         byte   %8.0g                  =1 if in profess. occupation
clerocc         byte   %8.0g                  =1 if in clerical occupation
servocc         byte   %8.0g                  =1 if in service occupation
lwage           float  %9.0g                  log(wage)
expersq         int    %9.0g                  exper^2
tenursq         int    %9.0g                  tenure^2
------------------------------------------------------------------------------
Sorted by:
```

“describe”命令是对数据和变量作基本描述的命令。运行该命令，我们可以得到样本容量、变量个数、变量名称、变量存储类型和格式以及变量的基本含义等信息。如果想修改变量名称、变量格式或变量含义等内容，可以分别使用“rename”，“format”和“label”等命令，也可以在数据编辑框中双击相应变量而直接修改。值得一提的是，在由大型问卷整理而来的数据中，由于变量众多，所以很难像“WAGE1.dta”这样给每个变量都起上非常直观的名字；通常的做法是把问卷问题的编号作为相应变量的名称。此时，大家需要对照问卷来了解各个变量的含义。

如果你想在既有变量的基础上定义新变量，可以用“generate”命令。比如，我想定义一个新的虚拟变量“male”。当该变量取值为 0 时，个体为女性；取值为 1 时，个体为男性。那么我可以运行下面的命令：

```
generate male = abs(female - 1)
```

其中，female 是已经存在的变量——其取 1 时，个体为女性；取 0 时，个体为男性。abs(.)是绝对值函数（Stata 中常用的函数或数学函数可以通过 help functions 或 help math functions 获得。但是，并不是所有的函数都可以用于 generate 命令；如果遇到这种情况，可以尝试 generate 的扩展命令——egen）。那么，一列名为“male”的新变量就生成了。我还可以通过如下的方式来生成：

```
generate male = 0 if female = = 1
replace male = 1 if male = = .
```

第一行命令是对所有的女性样本定义变量 male 的取值。如果只运行第一行命令，会发现：对于女性样本，male 取值为 0；而对于男性样本，male 的取值为“.”。这个小圆点代表 missing value，即对于男性样本，变量 male 的值缺漏了。为了把这些缺漏的值补上，我们加上第二行命令。“replace”命令用来修改变量的取值。第二行命令的含义是：将变量 male 所有缺漏的值更改为 1。如果不能用一个简单的数学函数生成整个变量，我们往往就要使用这种分步生成变量的方法。此外，在定义新变量的时候，写明变量的基本含义（即 variable label）是一个良好的习惯，既方便别人了解数据，又防止自己遗忘变量的含义。如果想删除这个刚定义的变量，可以使用“drop”命令（“keep”命令是与其恰好相反的命令）。

如果想进一步了解每个变量取值的基本特征，可以运行“summarize”命令，从而得到：

```
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
wage	526	5.896103	3.693086	.53	24.98
educ	526	12.56274	2.769022	0	18
exper	526	17.01711	13.57216	1	51
tenure	526	5.104563	7.224462	0	44
nonwhite	526	.1026616	.3038053	0	1
female	526	.4790875	.500038	0	1
married	526	.608365	.4885804	0	1
numdep	526	1.043726	1.261891	0	6
smsa	526	.7224335	.4482246	0	1
northcen	526	.2509506	.4339728	0	1
south	526	.3555133	.4791242	0	1
west	526	.1692015	.3752867	0	1
construc	526	.0456274	.2088743	0	1
ndurman	526	.1140684	.318197	0	1
trcommpu	526	.0437262	.20468	0	1
trade	526	.2870722	.4528262	0	1
services	526	.1007605	.3012978	0	1
profserv	526	.2585551	.4382574	0	1
profocc	526	.3669202	.4824233	0	1
clerocc	526	.1673004	.3735991	0	1
servocc	526	.1406844	.3480267	0	1
lwage	526	1.623268	.5315382	-.6348783	3.218076
expersq	526	473.4354	616.0448	1	2601
tenursq	526	78.15019	199.4347	0	1936

其中包括与每个变量对应的样本容量、算术平均数、标准差、最小值和最大值等信息。从而，我们可以了解每个变量基本的取值特征和取值范围。此外，我们还可以初步地检查变量的取值有无技术性错误。比如，wage, educ 和 exper 等变量都是非负数；当你看到这些变量的最小值是负数时，就要怀疑数据的录入是否正确（但有时，一些调查问卷会用负数来代表缺漏值）。再比如，变量“married”的含义是“婚否”，只取 0 和 1 两个值；如果看到其最大值是 2，就很可能存在数据的录入错误。如果没有剔除这些“问题数据”而直接进行回归等分析，其结果就可能存在偏差。

此外，诸如 ameans, centile, correlate, count, inspect, tabstat, tabulate 等命令都可以从某个角度对变量及其数据进行粗略的描述。大家可以对照帮助文件掌握它们的语法。

有时，把样本个体按照某个变量的取值排序，会有助于对数据的清晰观察。运行

“sort educ”命令，就可以把整个样本按照教育水平从低到高排序，从而就可以看看工资随着教育水平的增加会怎样变化。但是，sort 命令只能升序排列；如果还想降序排列，可以使用“gsort”命令。

有时，为了处理的方便，常常需要把一个截面数据转化成面板数据，也需要把一个面板数据转化成截面数据，这需要用到“reshape”命令。打开讲义附带的数据“consume2.dta”。这个数据是截面数据，变量“hhid”可以唯一地确定观测值。数据中的“consum1”,“consum2”和“consum3”分别表示某家庭在 1 月、2 月和 3 月的消费。现在，我想把这个数据转化成面板数据，即每一个观测值由两个维度的变量（一个表示家庭，一个表示时间）来确定。以下是相关的 do 文件的语句。

```
/* input the wide data and view */
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use consume2, clear
list

/* reshape the wide data into long data */
reshape long consum, i(hhid) j(month)
list

/* reshape back */
reshape wide consum, i(hhid) j(month)
list

/* rename variables */
rename consum1 consum01
rename consum2 consum02
rename consum3 consum03


/* reshape the wide data into long datg with string j-variable */
reshape long consum, i(hhid) j(month) string
list

/* reshape back */
reshape wide
list
```

第一段语句表示导入数据并浏览，可得数据的界面：

	hhid	consum1	consum2	consum3
1.	01	1000	1100	1300
2.	02	1600	1500	1400
3.	03	900	1000	1000
4.	04	2700	2500	2500
5.	05	1500	1600	1500
6.	06	2200	2400	2500
7.	07	3000	3000	2800
8.	08	1400	1500	1300
9.	09	2500	2500	2400
10.	10	900	1000	850

第二段语句表示将该数据转化成面板数据。转化的结果是：

	hhid	month	consum
1.	01	1	1000
2.	01	2	1100
3.	01	3	1300
4.	02	1	1600
5.	02	2	1500
6.	02	3	1400
7.	03	1	900
8.	03	2	1000
9.	03	3	1000
10.	04	1	2700
11.	04	2	2500
12.	04	3	2500
13.	05	1	1500
14.	05	2	1600
15.	05	3	1500
16.	06	1	2200
17.	06	2	2400
18.	06	3	2500
19.	07	1	3000
20.	07	2	3000
21.	07	3	2800
22.	08	1	1400
23.	08	2	1500
24.	08	3	1300
25.	09	1	2500
26.	09	2	2500
27.	09	3	2400
28.	10	1	900
29.	10	2	1000
30.	10	3	850

在“reshape”命令中，截面数据被形象地称作宽（wide）数据，而面板数据被形象地称作长（long）数据。所以，将截面数据转化成面板数据的命令是“reshape long”，把面板数据转化成截面数据的命令是“reshape wide”。在第二段语句中，“reshape long consum”表示：在即将转化成的面板数据中，变量“consum”将用来表示月消费。在面板数据中，每一个观测值中的月消费需要由两个维度的变量来确定，也即$consum_{ij}$。其中，i变量为家庭，j变量为月份；体现在命令中，就是“i(hhid) j(month)”。转化之后，便得到上面的结果。

同理，第三段语句将面板数据再转化回截面数据。到目前为止，大家需要注意以下问题：1、面板数据中表示时间的变量（月份）的名称是自己定义的，数据的相互转化伴随着时间变量的增删。2、不管在截面数据中，还是在面板数据中，关键变量的取名要统一，以方便程序的自动转化。比如在面板数据中，消费的变量取名为“consum”，是因为截面数据中消费变量名称的字母部分就是“consum”。如果面板数据中的消费取名为“consumption”，Stata 就无法进行转化工作。3、在转化的过程中，Stata 自动分拆“consum1”，“consum2”和“consum3”的字母部分和数字部分，并将数字部分赋值给新生成的时间变量。

在上面的情况下，时间变量的取值是 1、2 和 3，都是数字格式。但在很多情况下，时间变量会取诸如“01”，“02”和“03”的字符串格式。这种情况下，就需要在 reshape 的选项部分中加入“string”。第四段和第五段语句就说明了这个问题。

最后，在反复的转化过程中，如果想转化回去，直接运行“reshape long”或“reshape wide”即可，如第六段语句所示。更详细的内容请参见“reshape”命令的帮助文件。

有时，想观察的数据位于两个或多个不同的数据文件中，我们最好先将这些数据文件合并起来。数据的合并有三种基本情况。第一种是简单合并（simple merge）。比如，有两个数据文件。第一个数据记录了 100 名工人的性别和年龄，第二个数

据记录了这 100 名工人的工资和消费。我想把这两个数据合并起来，让所有工人的性别、年龄、工资和消费出现在同一个文件中。如果两个数据中相同的观测值序号代表相同的工人，那么我们只需根据观测值的序号将两个数据简单合并在一起即可。相关的命令是：

use *firstdata*

merge using *seconddata*

这两条命令的含义是：首先，导入第一个数据；然后，将第二个数据合并进来。在 Stata 的合并操作中，先导入的数据叫 master data 或 data in memory，合并进来的数据叫 using data 或 data on disk。

第二种是附加合并（append）。比如，有两个数据文件。第一个数据记录了 50 名工人的性别、年龄、工资和消费，第二个数据记录了另外 50 名工人的性别、年龄、工资和消费。如果我想把这 100 名工人的数据合并到同一个文件中，只需把一个数据附加到另一个数据的末尾即可。相关的命令是：

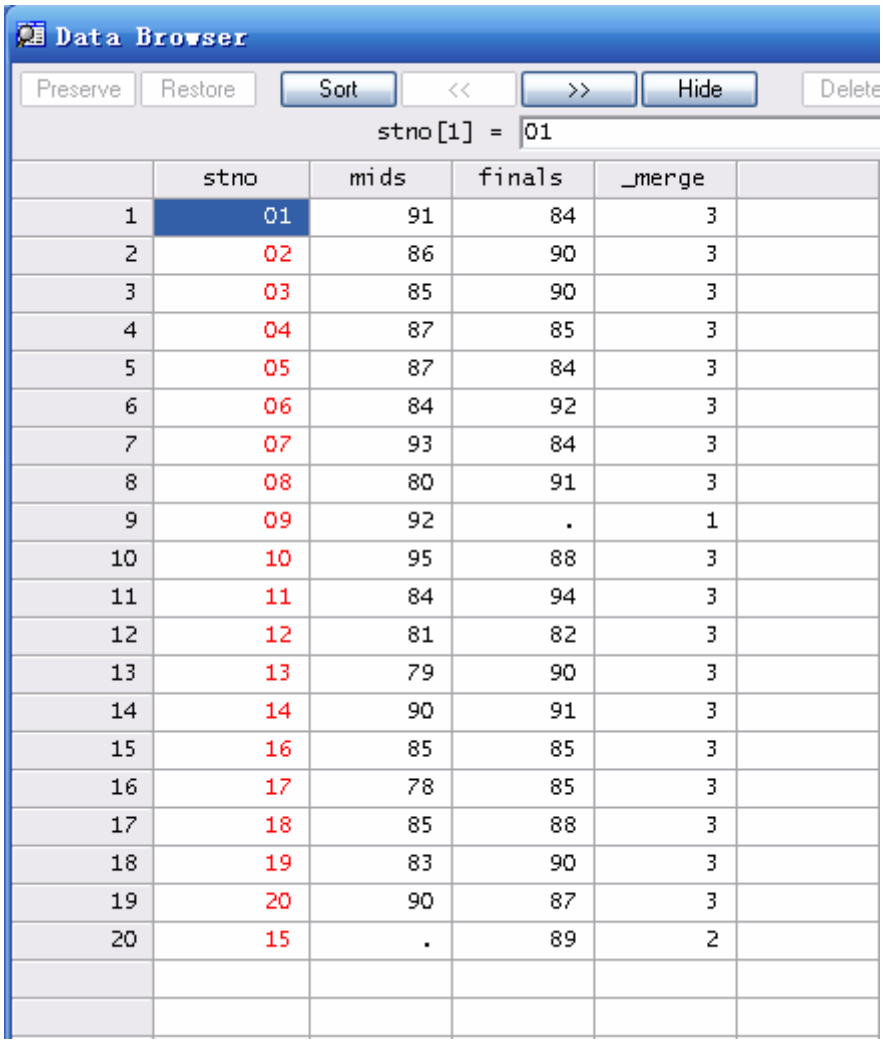
use *firstdata*

append using *seconddata*

针对以上两种数据合并的情况，大家可以自己杜撰数据，来尝试一下。下面重点讲第三种合并：匹配合并（match merge）。随讲义有两个文件，一个叫“mid.dta”，一个叫“final.dta”。前者记录着一个班的期中考试成绩，后者记录着该班的期末考试成绩。打开“mid.dta”，我们会看到学号为从 01 到 20 的学生的期中成绩（学号为 15 的学生没有成绩）；打开“final.dta”，我们会看到这些学生的期末成绩（学号为 09 的学生没有成绩）。我们想把这些学生的期中成绩和期末成绩合并在同一文件里，以方便比较和统计。在合并的过程中，我们需要把学生的学号作为合并的纽带；通过共同的学号，两个数据可以匹配在一起。下面是 do 文件的相关语句：

```
/* sort and save the using data */
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use final, clear
sort stno
save, replace

/* sort the master data */
use mid, clear
sort stno

/* merge based on "stno" and save */
merge stno using final
save all, replace
```

在匹配合并之前，我们必须对作为纽带的变量进行排序，并且排序要在所有的待合并数据中进行。排序之后，匹配合并才能继续。在上面的 do 文件中，第一段代码表示在 using data（“final.dta”）中对学号排序。第二段代码表示导入 master data，并对学号排序。第三段代码表示通过学号将两个数据合并起来，并且将合并好的数据存储为“all.dta”。合并后，得到如下数据：


Data Browser

Preserve Restore Sort << >> Hide Delete

stno[1] = 01

	stno	mids	finals	_merge
1	01	91	84	3
2	02	86	90	3
3	03	85	90	3
4	04	87	85	3
5	05	87	84	3
6	06	84	92	3
7	07	93	84	3
8	08	80	91	3
9	09	92	.	1
10	10	95	88	3
11	11	84	94	3
12	12	81	82	3
13	13	79	90	3
14	14	90	91	3
15	16	85	85	3
16	17	78	85	3
17	18	85	88	3
18	19	83	90	3
19	20	90	87	3
20	15	.	89	2

在这个数据中，有四个变量。其中，“stno”，“mids”，“finals”分别表示学号、期中成绩和期末成绩。“_merge”表示相应的观测值来源于哪个数据文件。该变量等于 1，则相应的观测值来源于 master data；该变量等于 2，则相应的观测值来源于 using data；该变量等于 3，则相应的观测值来源于两个数据文件（关于“_merge”的更多的涵义请参见 merge 的帮助文件）。通过观察合并后的数据，我们可以看出 Stata 合并的算法：1、把两个文件中的数据分别按照学号排序。2、以 master data 中的学号为根本，将 using data 中具有相同学号的数据一条一条并进来；如果某学号在 using data 中没有记录，则在相应的变量中用小圆点代替（如 09 号学生）。3、如果某学号在 master data 中无记录、但在 using data 中有记录（如 15 号学生），则暂不合并；待所有在 master data 中存在的学号都合并完之后，再考虑那些只在 using data 中出现的学号。

匹配合并不仅能合并两个数据，还能合并多个数据。在讲义的附带数据里，除了“mid.dta”和“final.dta”之外，还有“gender.dta”。这个文件记录了这 20 名学生的性别。现在，我想把这三个数据合并在一起。下面是 do 文件的相关命令：

```
/* sort and save the using data */
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use mid, clear
sort stno
save, replace

use final, clear
sort stno
save, replace

/* sort the master data */
use gender, clear
sort stno

/* merge based on "stno" and save */
merge stno using mid final
save all_three, replace
```

这几段代码与前面的几段代码大同小异：首先，先对所有的 using data 以及 master data 排序，然后进行匹配合并。

以上的两个例子有个共同点：不管在 master data 中，还是在 using data 中，作为纽带的变量（“stno”）都可以唯一地确定观测值（即不存在两个观测值，它们拥有相同的学号）。如果这个“唯一性”不满足，又该怎么办呢？在讲义的附带数据里，有“consume.dta”和“earning.dta”两个文件。前者记录了家庭层面的消费（变量包括家庭编号和家庭月消费），而后者记录了个人层面的工资收入（包括家庭编号、个人编号和个人工资）。如果想把这两个数据合并在一起，得依靠变量“hhid”（家庭编号）作纽带。“hhid”在“consume.dta”中可以唯一确定观测值，但在“earning.dta”中却不可——因为在这个数据中，同一个家庭内，还有不同的个人及其工资。下面是进行此操作的 do 文件的相关命令。

```
/* sort and save the using data */
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use consume, clear
sort hhid
save, replace

/* sort the master data */
use earning, clear
sort hhid pid

/* merge based on "hhid" and save */
merge hhid using consume, uniqusing
save e_c, replace
```

关键的改动位于倒数第二行——在 merge 命令中，加入“uniqusing”的选项。“uniqusing”表示“hhid”在 using data（“consume.dta”）中可以唯一确定观测值，而在 master data 中不可以。同样，当合并的纽带变量只能在 master data 中唯一确定观测值时，选项就变为“uniqmaster”。在大型的微观层面的调查中，数据往往被划分为不同的层面——个人层面、家庭层面、社区层面等等。要想把不同层面的数据合并在一起，就要用到上述的操作。

另一种“唯一性”不满足的情况就是截面数据和面板数据的合并。讲义附带的“gpa.dta”记录了学号从 01 到 20 的学生在两个学期中的 gpa，是一个简单的面板数据。另一个数据“gender.dta”则记录了这 20 位学生的性别，是一个截面数据。我们现在想把这两个数据合并到一起。在合并的过程中，学号（stno）在截面数据中具有唯一性，但在面板数据中没有（面板数据中，学号和学期共同确定一个观测值）。同理，相应的 do 文件命令为：

```
/* input data and sort */
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use gender, clear
sort stno
save,replace

use gpa, clear
sort stno smst

/* merge */
merge stno using gender, uniqusing
save gpa_gender, replace
```

我们看到，原来截面数据中的变量（female）被合并到面板数据中之后，被视作不随时间变化的变量。

在合并操作之后，一定要检查一下合并好的数据，看看是否与自己的合并初衷相符。“merge”命令非常重要，需要大家在实践中熟练掌握。

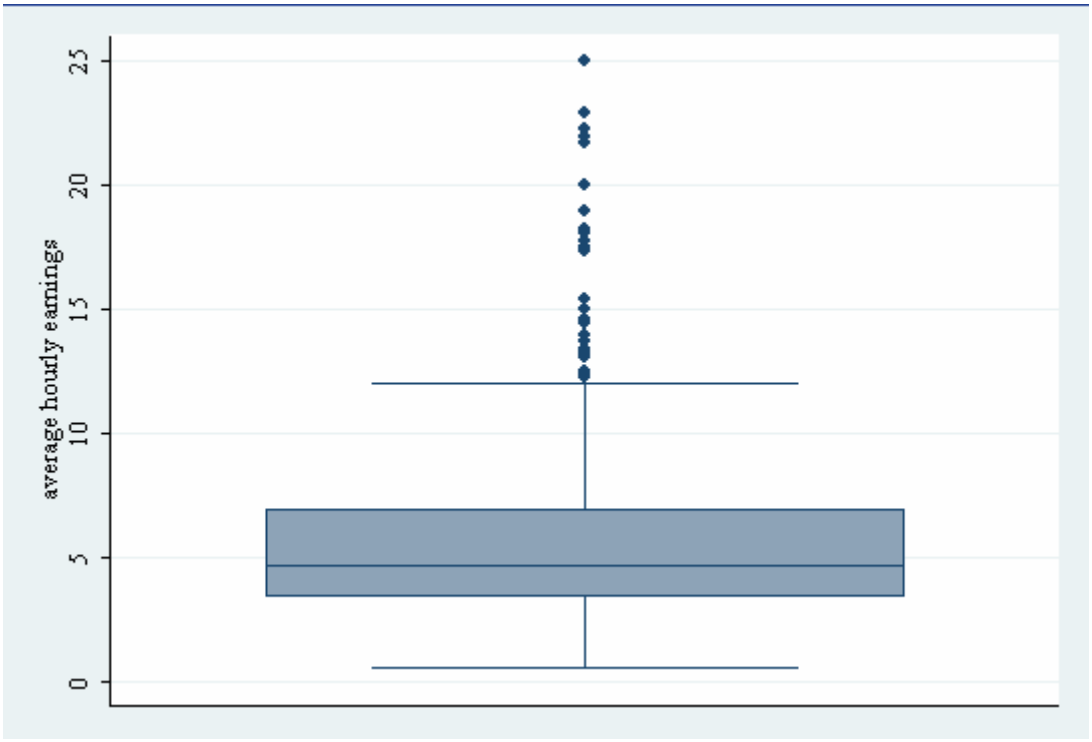
五、画图

上面一部分，我们提到了“看”数据的许多方法。除了冷冰冰的数字，我们还可以通过鲜活形象的图来了解数据。因为画图的内容较多，故单独列为一个部分。

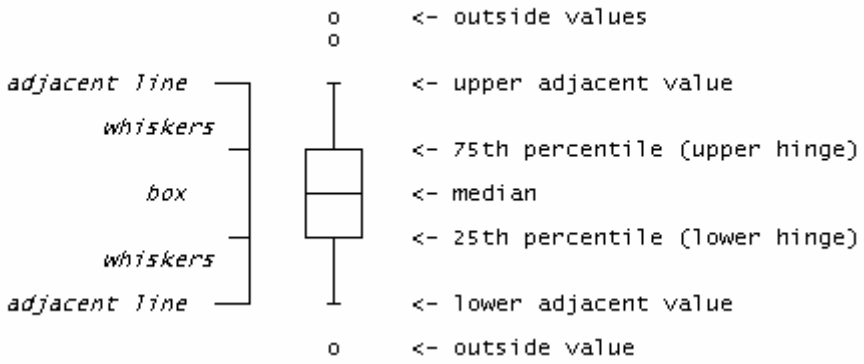
输入“help graph”，我们可以看到几乎所有和画图有关的命令。我们对画图的学习就从这里开始。下面仍以“WAGE1.dta”为主进行说明。

首先，我们介绍对单个变量如何画图。对于单个变量，我们最关心的可能就是这个变量取值的特征了。

第一个常用的命令是 graph box，即画盒状图的命令。运行“graph box wage”，可得：



下面是对盒状图结构的说明，摘自其帮助文件。

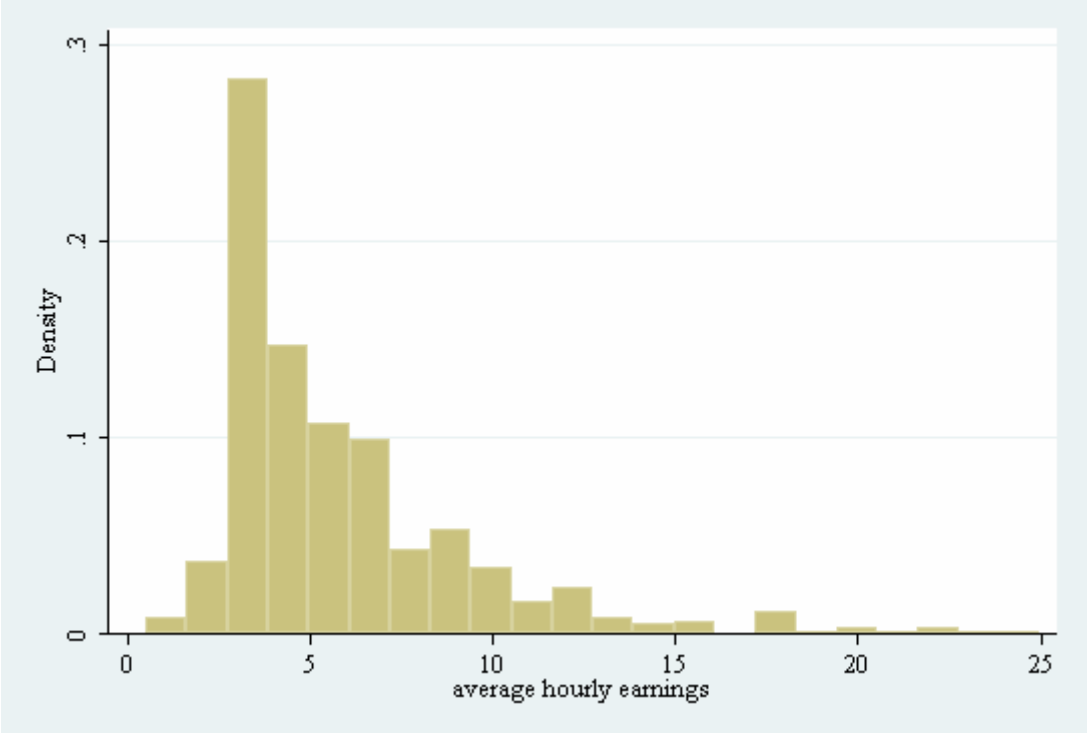


其中，upper adjacent value = 75th percentile + (75th percentile – 25th percentile) * 1.5, lower adjacent value = 25th percentile – (75th percentile – 25th percentile) * 1.5。实际操作中，如果这样计算出的upper adjacent value比样本的最大值还大，那么upper adjacent value就取样本的最大值；同样，如果lower adjacent value比样本的最小值还小，那么lower adjacent value就取样本的最小值。

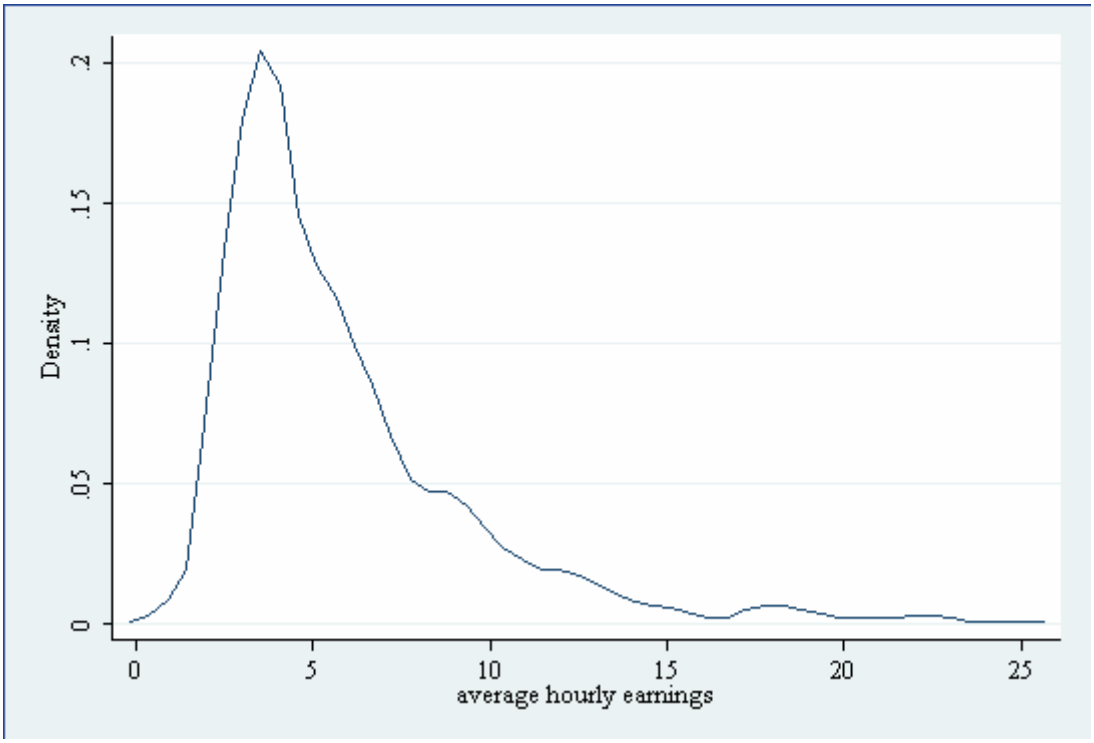
一般来说，比 upper adjacent value 大或比 lower adjacent value 小的值都是奇异值

（outlier）。因为是否包含奇异值会对数据分析的结果产生比较显著的影响，所以对数据做深入分析之前，有必要通过盒状图等图来发现和处理奇异值。

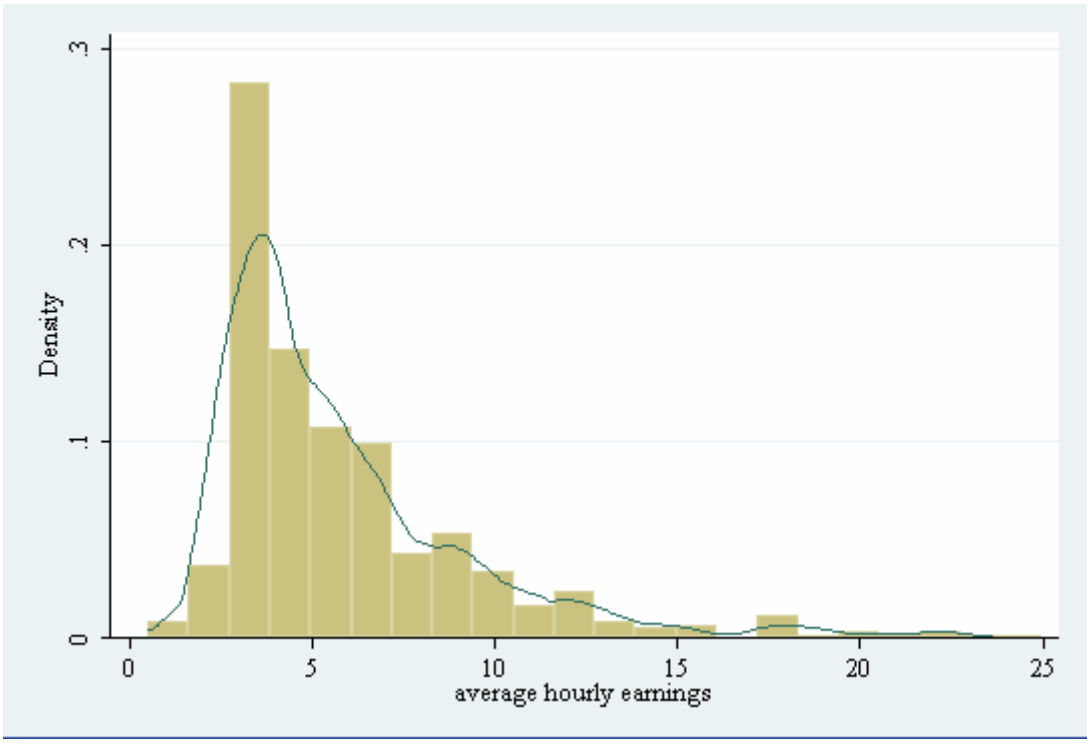
进一步，如果想形象地知道单个变量取值的分布情况，可以用 histogram 命令，这个命令用来画某变量取值的柱状图。图的横坐标是该变量的取值，纵坐标是该变量取各个值的频率。下图是运行“histogram wage”的结果。



如果想将工资的分布拟合成光滑的曲线，可以运行“kdensity wage”，得到：



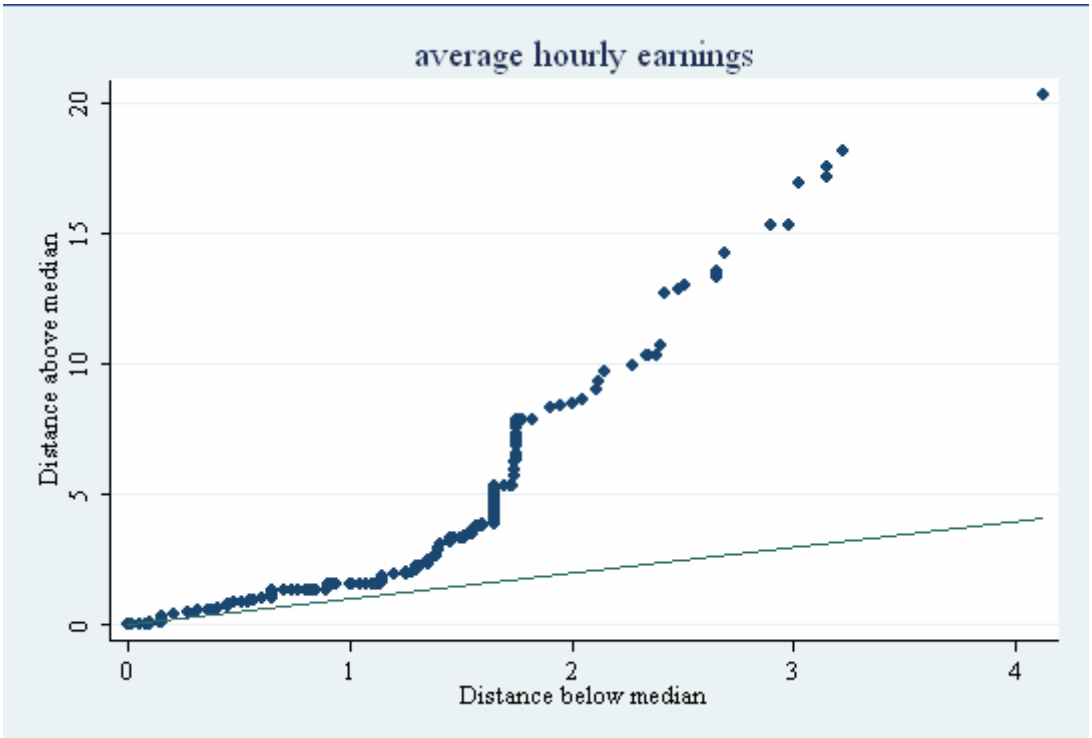
如果想将这两个图叠加，可以运行“histogram wage, kden”，得到：



从以上三幅图可以看出，工资的分布是左偏的。此外，spikeplot, dotplot 等命令

都可以形象地画出单个变量取值的分布。

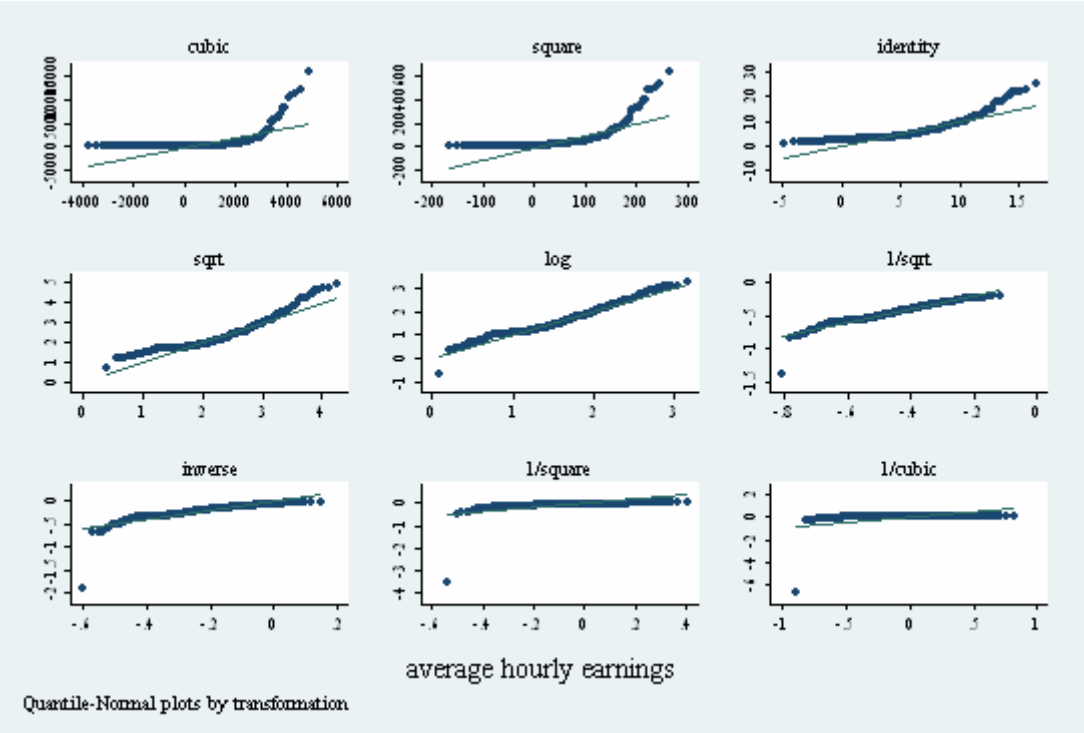
有时，人们还需要知道某个变量的分布距离一些常见分布（比如正态分布）有多远。symplot, quantile, qnorm, pnorm, qchi, pchi 等命令就可以实现这些想法。symplot 检验某变量的分布距离对称分布有多远。运行“symplot wage”，可得：



其中，直线为 45 度线。散点越接近那条直线，该变量的分布就越接近于对称分布。由这个图可以看出，工资的分布远远不是对称分布。

同理，quantile 用于检验变量的分布距均匀分布有多远，qnorm 和 pnorm 用于检验变量的分布距正态分布有多远，qchi 和 pchi 用于检验变量的分布距卡方分布有多远。这些图中都有一条 45 度线，图中的散点越接近这条线，变量的分布就越接近于相应的常见分布。

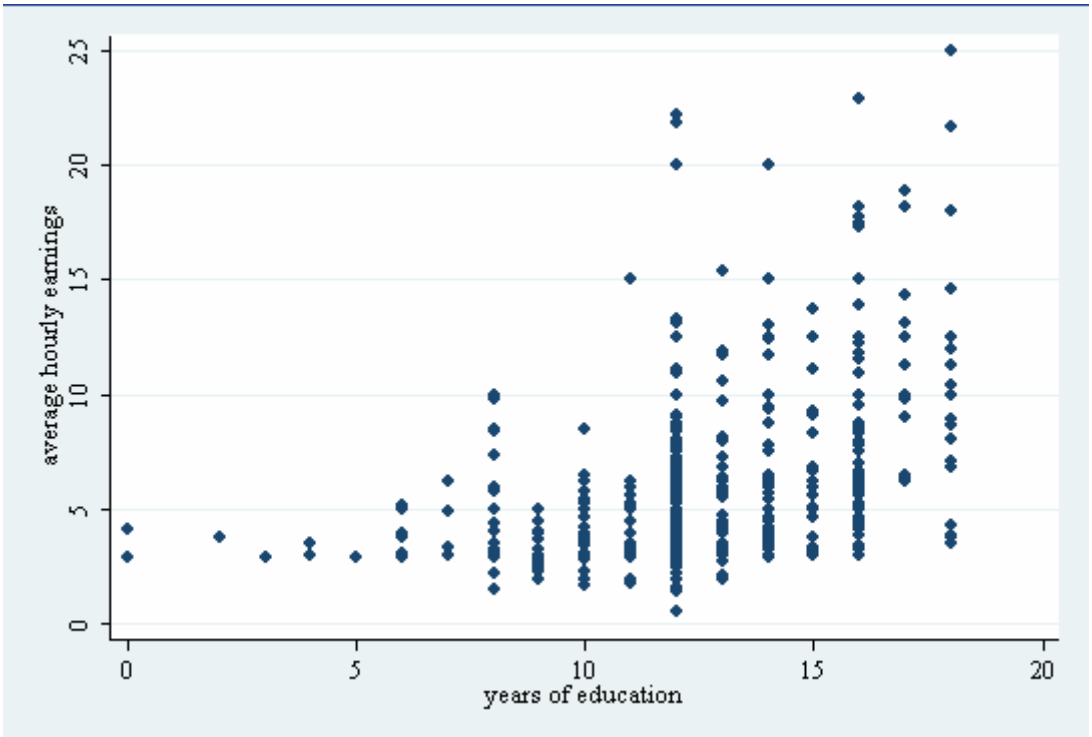
需要补充的一个相关命令是 qladder。运行“qladder wage”，可得：



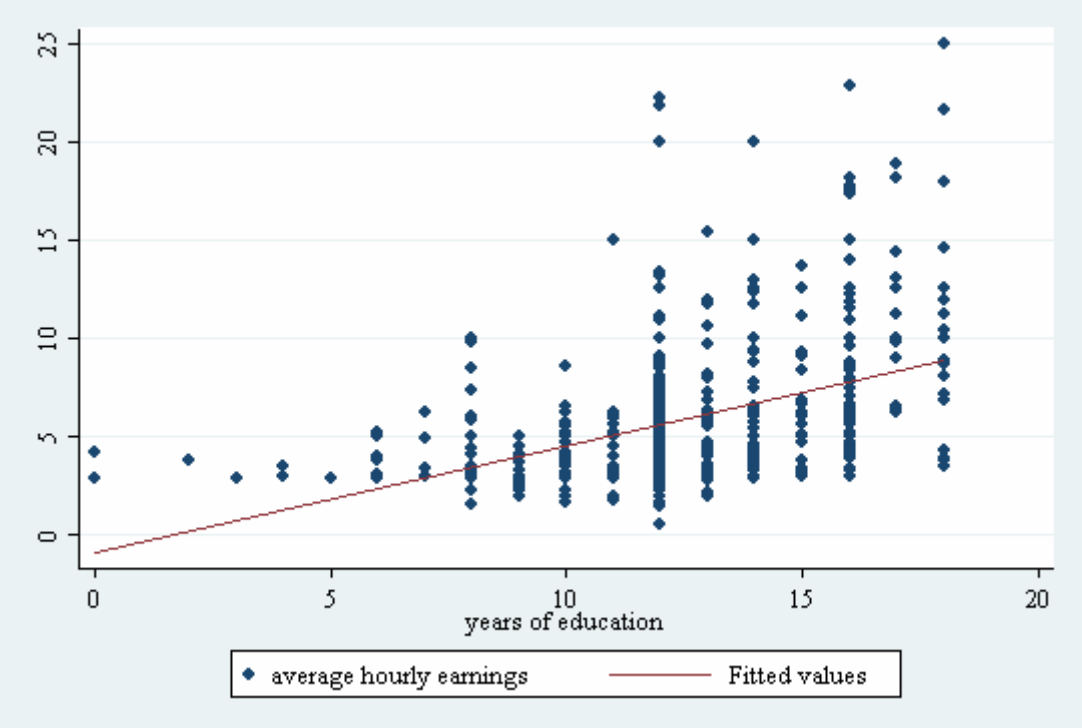
这九幅图用来检验工资及其幂、对数等函数距正态分布有多远，相当于对工资及其幂、对数等函数用 qnorm 命令。从中我们可以看出，工资的对数形式（最中间的那幅图）比工资本身（右上角的那幅图）更接近正态分布。

接下来，我们介绍对两个变量如何画图。对于两个变量，我们最关心的大概就是它们之间的关系了。

对于两个变量，最重要的一类画图命令是 graph twoway。所谓“twoway”，就是两个变量构成的平面坐标系。在 graph twoway 这一大类命令中，还有许多小类，比如两个变量之间的关系用点表示还是用线表示等等。运行“graph twoway scatter wage educ”，可得：

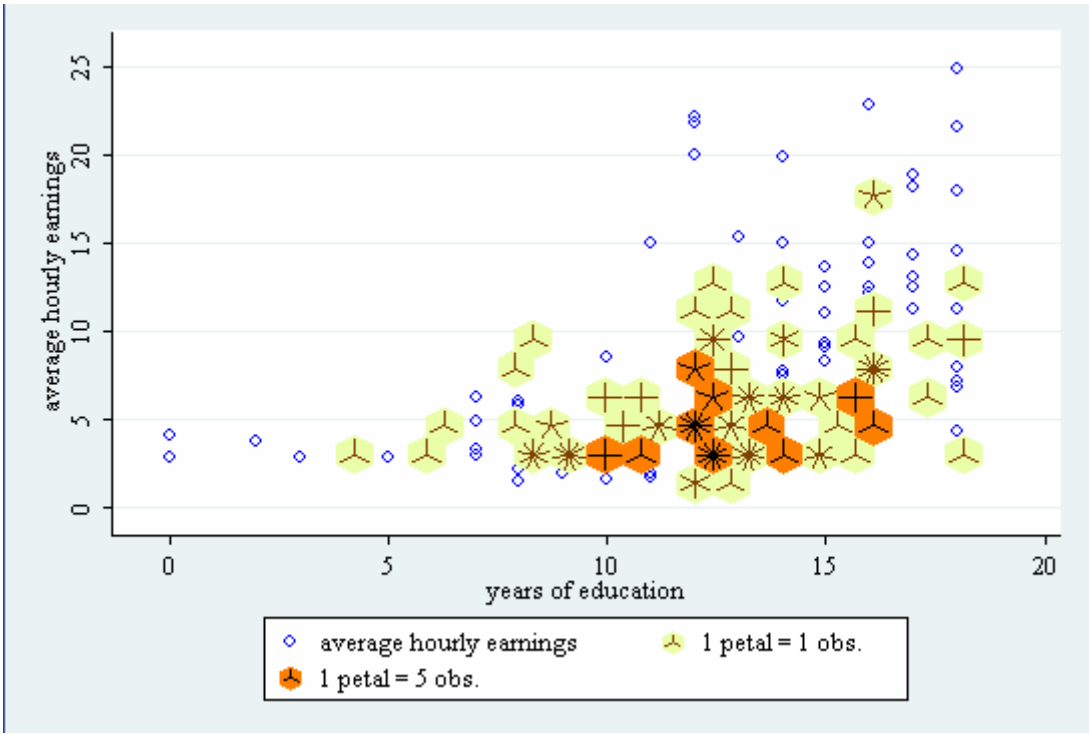


这是用点表示工资和教育水平之间关系的命令。对应于每个教育水平，都有与之相对应的工资水平。从图中，我们大概可以看出，随着教育水平的提高，工资的平均水平是上升的。运行“graph twoway scatter wage educ || lfit wage educ ”，可得：



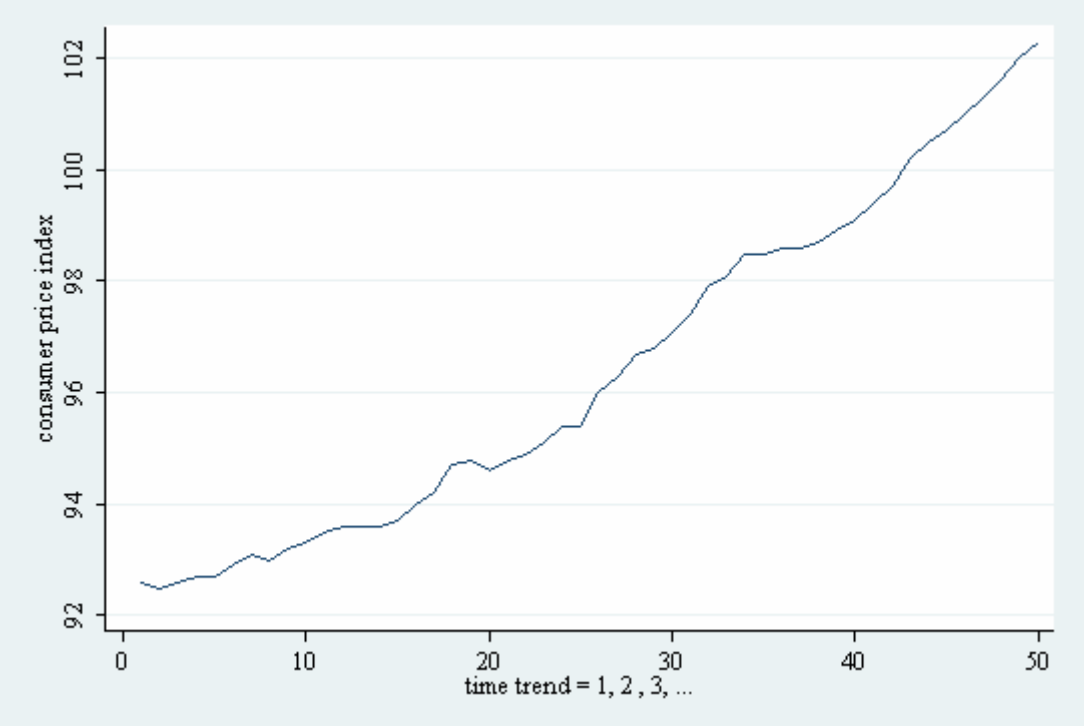
其中的红色直线是工资对教育水平的拟合线。从这幅图中，我们就可以清晰地看出工资随教育水平上升的趋势。如果你想用曲线来拟合二者之间的关系，可以将上述命令中的“lfit”换成“qfit”或“fpfit”等，或直接运行“lowess wage educ”。

从上面的散点图可以看出，对于某个教育水平，散点有时稠密，有时稀疏。稠密意味着工资取相应值的点较多，稀疏意味着工资取相应值的点较少。下面介绍的 sunflower 命令会使“稠密”和“稀疏”变得更明显。运行“sunflower wage educ”，可得：



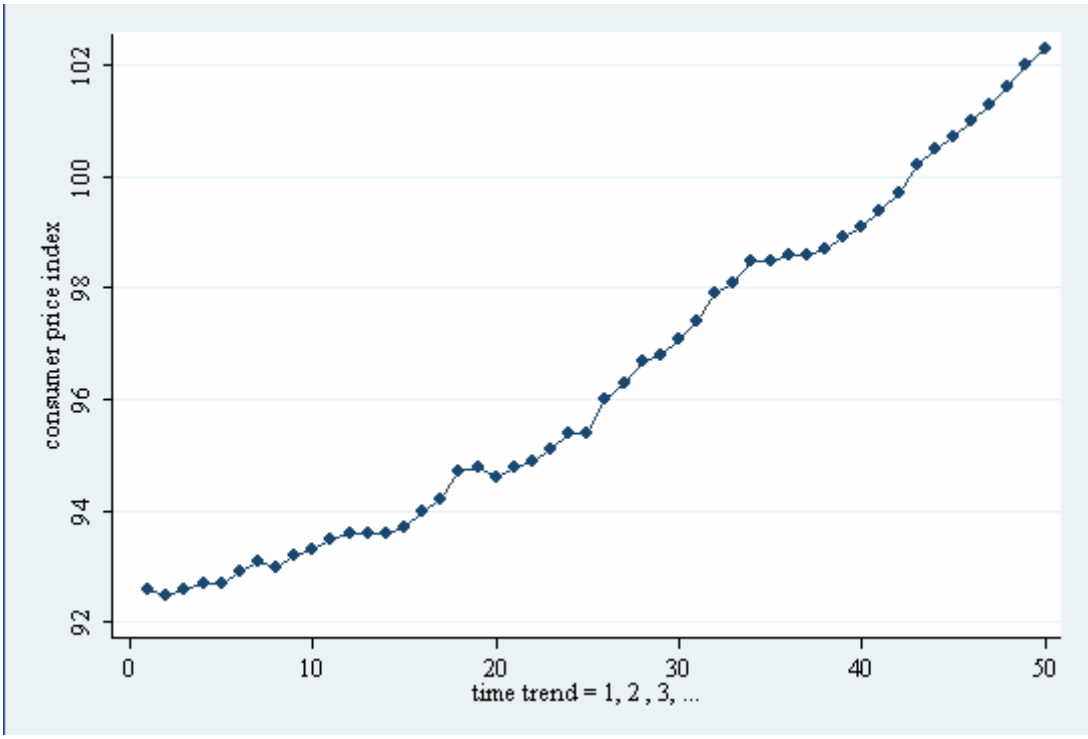
其中，蓝色小圆圈代表一个观测值，而一朵绿色或橘红色的“花”代表许多观测值。对于绿色花来说，一片花瓣代表一个观测值；对于橘红色花来说，一片花瓣代表 5 个观测值（一片花瓣代表多少观测值可以人为设定）。因此，看花的颜色以及花瓣的数量就可以判断出工资在相应值附近的取值是稠密还是稀疏。

下面以“WAGEPRC.dta”为例，说明用曲线表示两个变量之间关系的命令。打开这个数据，运行“graph twoway line price t if t<=50”，可得：



这幅图画出了在前 50 期，消费价格指数（price）随时间（t）的变化趋势。如果想用直线或平滑曲线拟合，可以仿照画散点图的命令，在后面加上 lfit, qfit, fpfit 等命令。

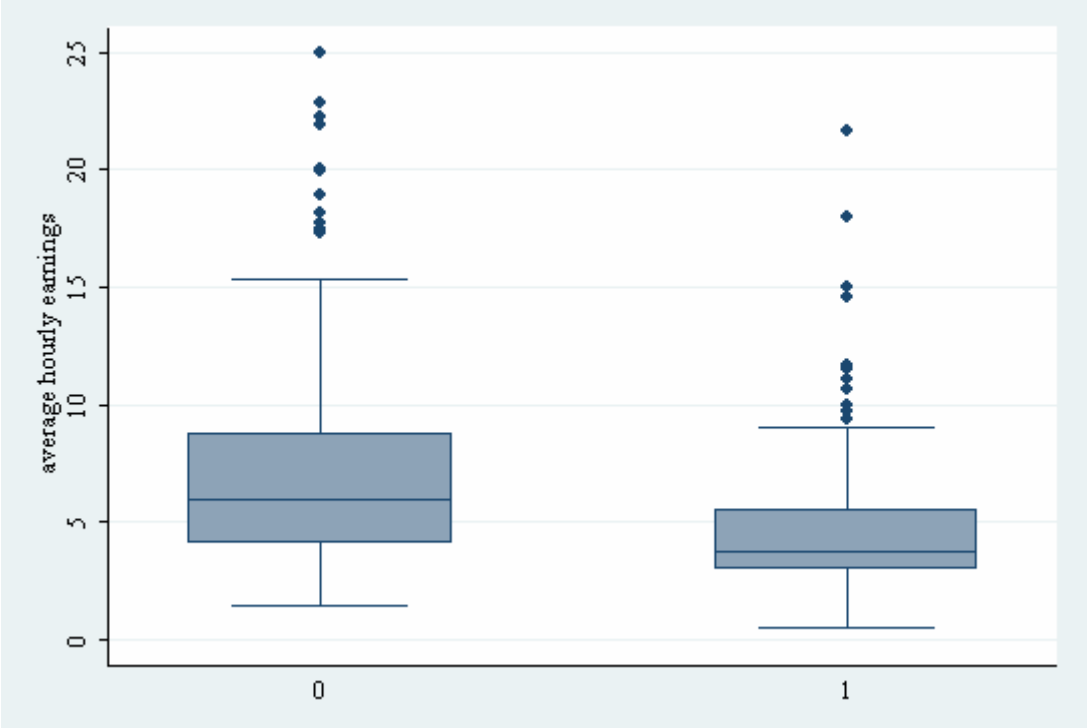
此外，运行“graph twoway connected price t if t<=50”，可得：



这个命令用圆点标出了曲线的每个转折点。

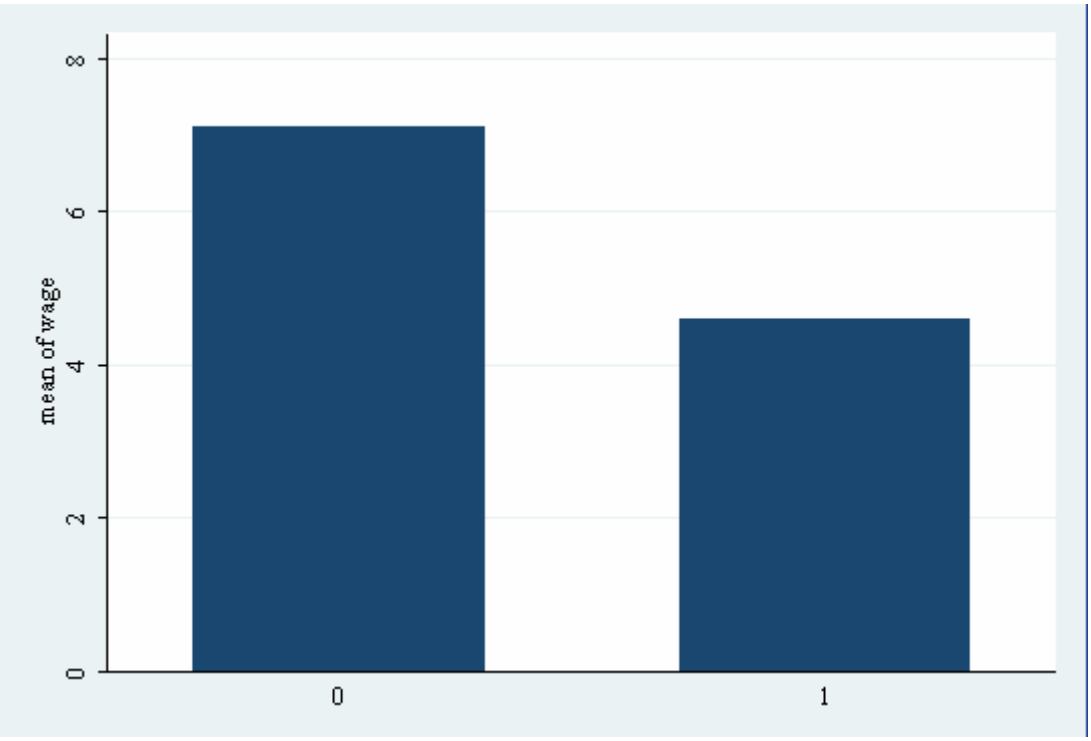
除了散点和曲线以外，还可以用面积图、柱状图、针状图等许多其他形式来勾画两个变量之间的关系，只需将上述命令中的”scatter”, “line”或”connected”换成”area”, “bar”或”spike”等命令即可。详情请参考 graph twoway 的帮助文件。

上文提到的 graph box 命令，还可以表示两个变量之间的关系。比如，回到“WAGE1.dta”，我想看工资和性别的关系，运行“graph box wage, over(female)”，可得：



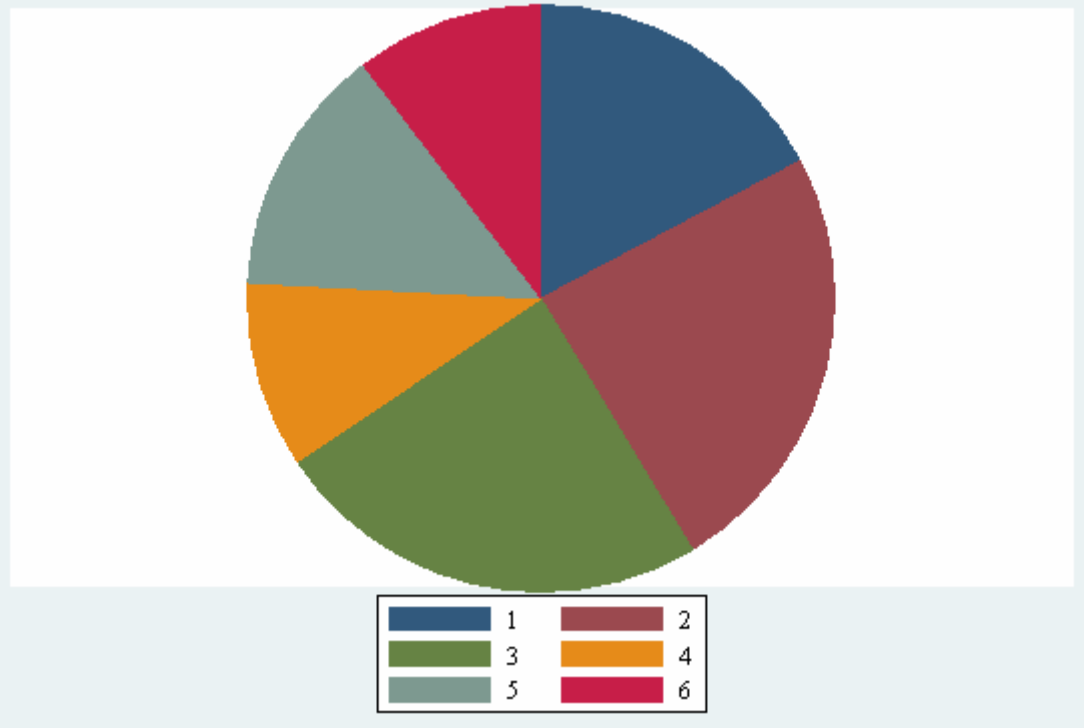
这条命令的语法和 twoway 命令的语法略有不同：twoway 把两个变量直接罗列在一起；而 graph box 拿其中的一个变量（female）做分组或分类变量，从而画出工资在两个性别上的对比情况，也就反映出了工资和性别的某种关系。从这个图中可以看出，男性工资的平均水平（左边一组）要比女性工资的平均水平（右边一组）高一些。

有时，我们想直接画出不同性别人群的平均工资，可以运行“graph bar wage, over(female)”或“graph dot wage, over(female)”。前者是柱状图，后者是点状图。下图为运行第一条命令画出的柱状图：



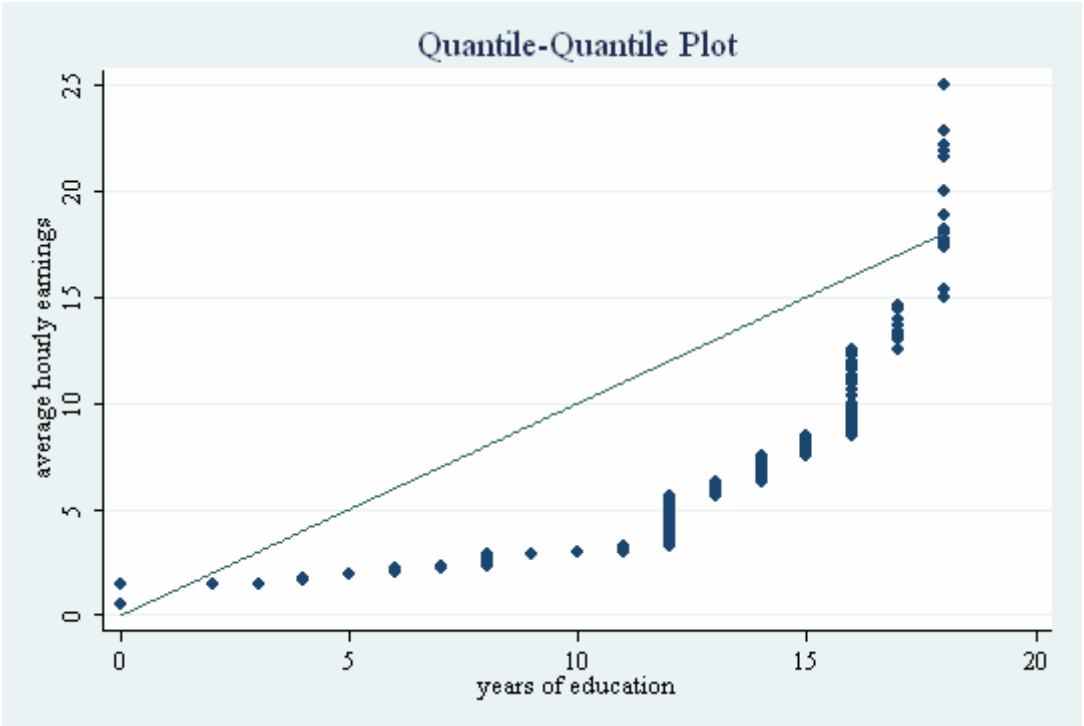
从图中可以明确看出，男性的平均工资比女性的平均工资高。如果想让柱状图旋转 90 度，可以使用 graph hbar 命令。

下面介绍饼状图的画法。打开“revenue.dta”。该数据记录的是一个连锁超市在六个地区的年度收益情况。变量 district 是地区的编号，1 号到 5 号的地区都有两家超市分店，而 6 号地区只有一家分店；变量 revenue 是相应分店的收益（百万元）。现在，总店经理想对比这六个地区的收益情况，即想了解收益和地区的关系。运行“graph pie revenue, over(district)”，可得：



这条命令把每个地区所有分店的收益相加，然后列出这六个地区各自的总收益。

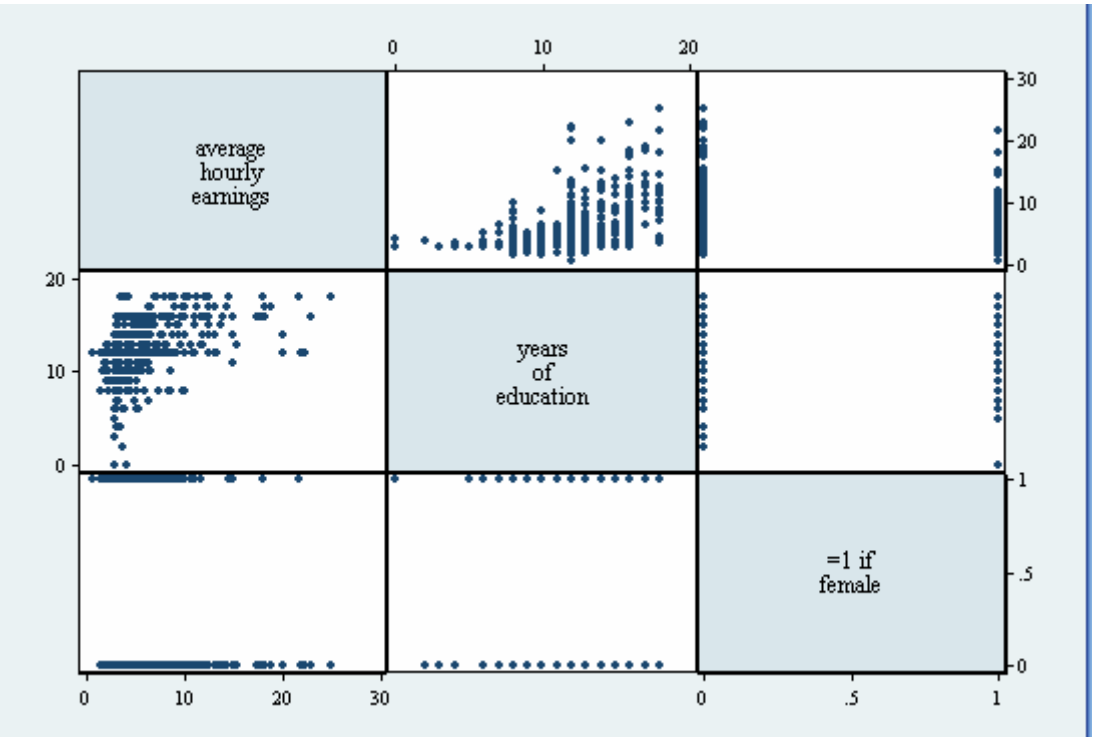
有时，我们想比较两个变量的分布之间有多大差距。回到“WAGE1.dta”，运行“qqplot wage educ”，可得：



这些散点越接近于 45 度线，两个变量的分布就越接近。因此，上图告诉我们，工资和教育的取值分布相差比较大。

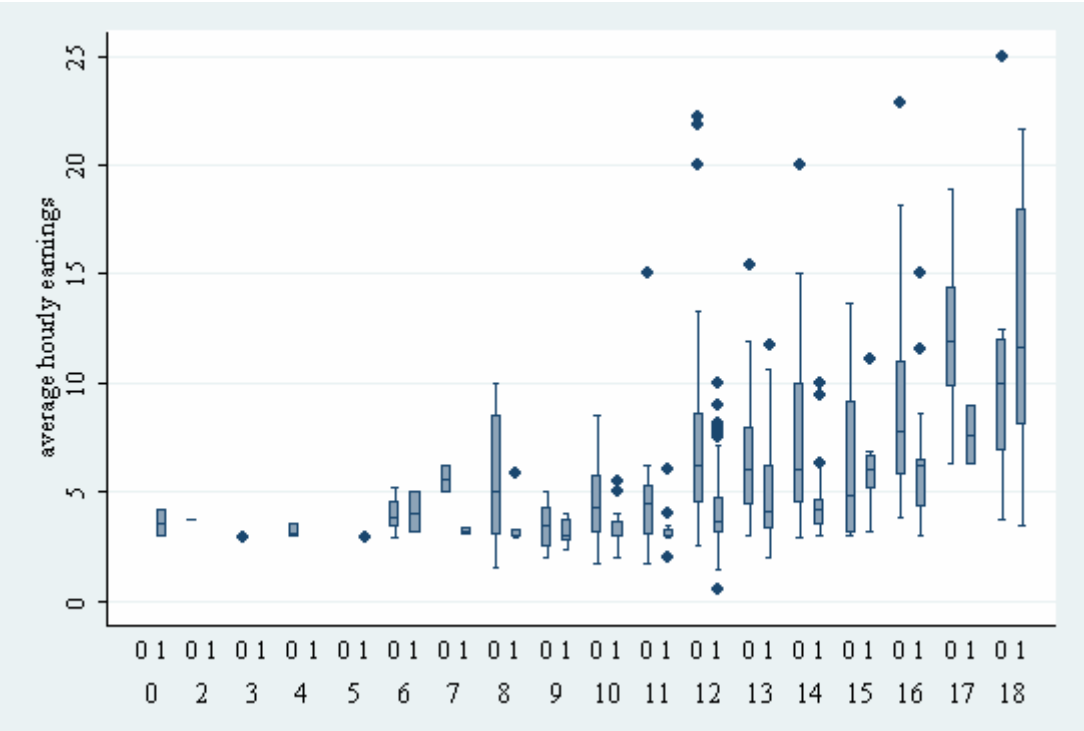
接下来，我们介绍对多个变量如何画图。

有时，我们对多个变量之间的相互关系很感兴趣，想同时画出它们之间的关系。比如，我们对工资、教育和性别三者之间的关系很感兴趣，运行“graph matrix wage educ female”，可得：



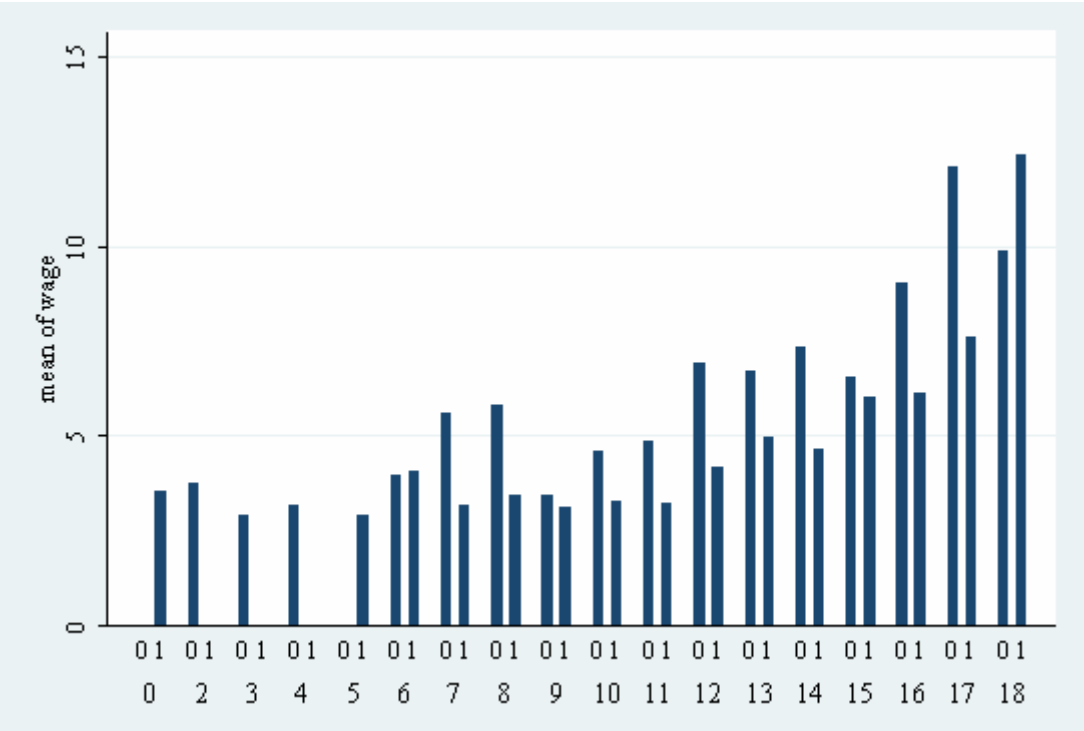
graph matrix 命令用矩阵的形式同时画出多个变量之间的相互关系。第 1 行第 2 列和第 2 行第 1 列的图画出了工资和教育的关系，第 1 行第 3 列和第 3 行第 1 列画出了工资和性别的关系，第 2 行第 3 列和第 3 行第 2 列画出了教育和性别的关系。任意两个变量之间的关系都用两幅图表示，只不过横纵坐标恰好颠倒。

上文介绍的 graph box, graph bar, graph dot 等命令都可以画多个变量之间的关系。运行“graph box wage, over(female) over(educ)”，可得：



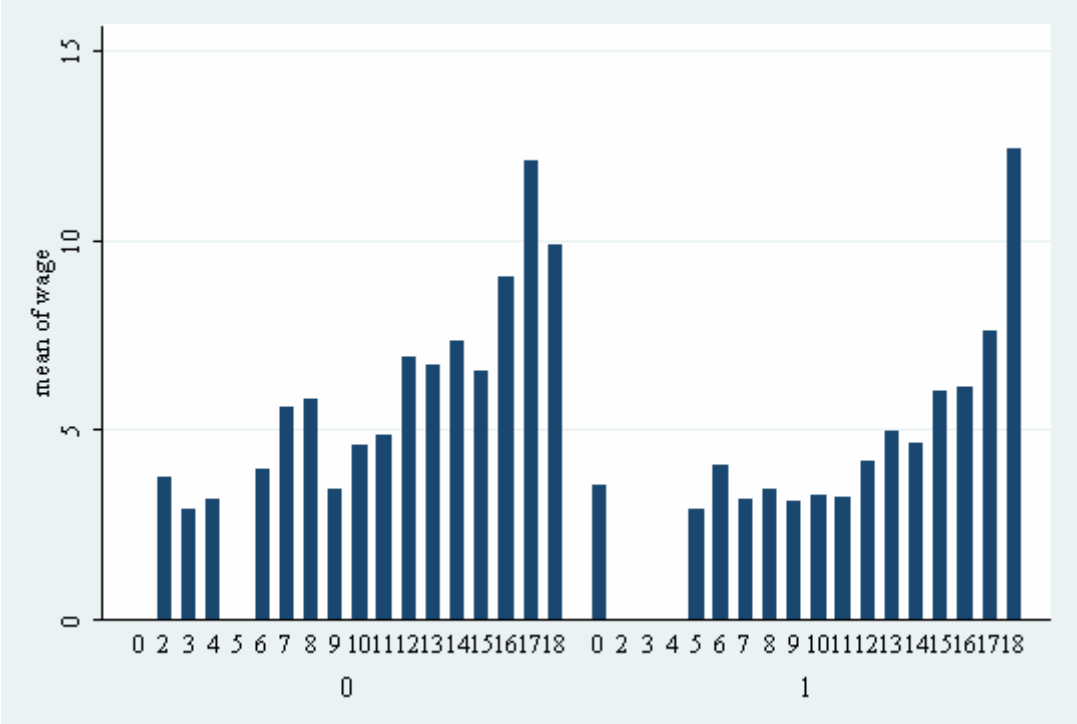
这幅图画出了在任何教育水平上，工资的性别差异。纵坐标是工资，横坐标有两行：底下一行为教育水平，上面一行为（在每个教育水平上的）性别。从这幅图中，我们不仅可以看出，在绝大多数教育水平上，男性平均工资都高于女性；而且可以看出，不管是男性还是女性，随着教育水平的提高，工资的平均水平都有所提高。

运行“graph bar wage, over(female) over(educ)”，可得：



这一幅图与上一幅图类似，只不过画出了每一类人的平均工资。其揭示的规律也与上一幅图类似。

命令后的“over”表示按什么变量分组；有几个“over”，就分几层。此外，“over”的前后顺序不同，画出的图就不同。运行“graph bar wage, over(educ) over(female)”，可得：

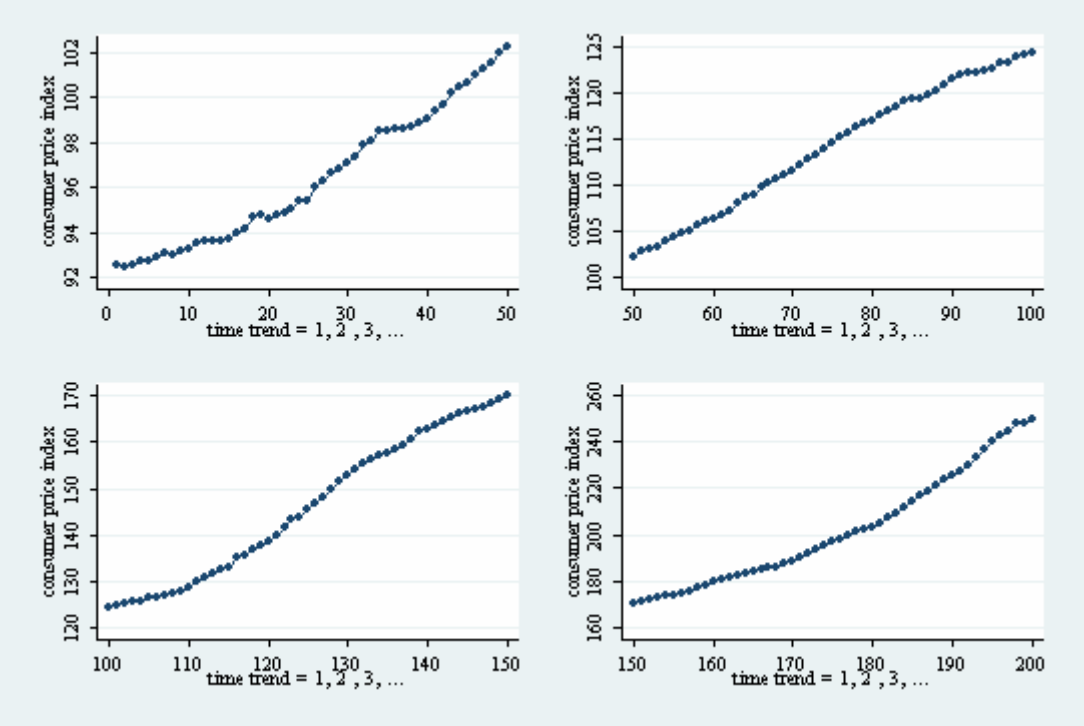


这幅图横坐标的分层次序发生了改变，表示对于每一个性别，工资如何随教育水平变化。如何将横坐标分层，取决于大家希望通过图传达怎样的信息。graph dot 命令也可以画出类似的图，不再赘述。

画图的主要命令就介绍到这，下面简单介绍图的设定。只用上面所讲的简单命令画出的图可能不会令你满意。比如，你想修改图的整体风格、图的颜色或大小、横纵坐标的标识、图的标题、图例的格式、图的背景等等。一般来说，每一个画图命令都有非常详细的“options”，用来设定图的特征。大家可以研读帮助文件去掌握相应的命令。此外，还可以用对话框操作。点击菜单栏中的“Graphics”，你可以看到常用的画图命令以及对图的各种设定；点击菜单栏中的“Prefs → Graph Preferences…”，就可以对图片的风格和字体等特征进行设定（其实，直接在图片上点击鼠标右键，选择“Preference…”，就可以完成这一操作）。

最后，简单介绍一下图片的存储和合并。存储图片可以用“graph save”命令，也可以在图上点击鼠标右键，选择“Save Graph…”。图片的存储格式有许多，可以自行选择。如果需要转换图片的存储格式，可以使用专门的图片处理软件。

如果想把几幅画好的图合并成一幅图，可以使用 graph combine 命令。下图是“WAGEPRC.dta”文件中，消费价格指数随时间的变化的四个图（时间共 200 期，每图 50 期）合并成一幅图的情形：



下面是 do 文件的相应语句。

```
/*input data*/
cd E:\工作\助教\2007春高级计量经济学II\Stata讲义
use wageprc, clear

/*graph and rename*/

cap graph drop _all

twoway connected price t if t<=50
graph rename cpi1

twoway connected price t if t>=50 & t<=100
graph rename cpi2

twoway connected price t if t>=100 & t<=150
graph rename cpi3

twoway connected price t if t>=150 & t<=200
graph rename cpi4

/*combine graphs*/
graph combine cpi1 cpi2 cpi3 cpi4
graph drop cpi1 cpi2 cpi3 cpi4
graph save cpi, replace
```

另外，关于图片的导入、查询、展示、描述、改名和删除等操作，请参照 graph 命令的帮助文件，这里不再赘述。画图命令是一大类命令，以上所述也只是皮毛，更多的知识需要大家在实践中慢慢积累。

六、回归分析初探

回归分析是计量经济学的核心。在做完详细的数据描述和处理之后，就可以根据研究需要，做初步的回归分析了。本部分先以“WAGE1.dta”为例，介绍了和回归分析相关的一些操作；然后以“WAGEPRC.dta”为例，补充说明了时间序列回归中的一些操作。

在“WAGE1.dta”中，我们想研究工资取决于怎样的因素。经典的工资方程是：

$$wage_i = \beta_0 + \beta_1 educ_i + \beta_2 \exp er_i + u_i$$

wage 是工资，educ 是教育水平，exper 是工作经验，u 是误差项，下标 i 表示个人。运行“reg wage educ exper”，可得：

```
. reg wage educ exper

      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  2,   523) =   75.99
       Model |  1612.2545      2  806.127251            Prob > F      =  0.0000
    Residual |  5548.15979   523  10.6083361            R-squared     =  0.2252
-------------+------------------------------           Adj R-squared =  0.2222
       Total |  7160.41429   525  13.6388844            Root MSE      =   3.257

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .6442721   .0538061    11.97   0.000     .5385695    .7499747
       exper |   .0700954   .0109776     6.39   0.000     .0485297    .0916611
       _cons |  -3.390539   .7665661    -4.42   0.000    -4.896466   -1.884613
------------------------------------------------------------------------------
```

回归结果分为三部分：左上角的部分为各种平方和，右上角的部分为回归的某些整体参数，下面的部分为系数及其相关的参数。我们分别介绍。

先看左上角的表格。与“SS”相对的那一列为各种平方和。其中，与“Model”相对的是解释平方和，与“Residual”相对的是残差平方和，与“Total”相对的是总平方和。显然，总平方和 = 解释平方和 + 残差平方和。与“df”相对的那一列是各种平方和的自由度，而与“MS”相对的那一列是各种平方和除以相应的自由度。

再看右上角的部分。第一行为样本容量。第二行是检验联合显著性的 F 统计量的值。第三行告诉我们，回归是联合显著的。第四行和第五行分别是 R 平方和调整的 R 平方。在 Gauss-Markov 假设中，$E(u^2 \mid x) = \sigma^2$。第六行的“Root MSE”就是对 σ 的估计 $\hat{\sigma}$，也就是左上角“Residual”那一行、“MS”那一列的数字（10.6083361）的算术平方根。

最后看下面的部分。三行数据分别是 educ 的系数，exper 的系数和常数项的估计值及相关的参数。第一列为系数的估计值；第二列为系数的标准误；第三列为各系数的 t 值（第一列除以第二列）；第四列为 p 值，反映系数的显著程度；第五、

六列为系数的置信区间。可以看出，教育水平和工作经验与工资都有显著的正相关关系。

估计完回归方程之后，还需要进行检验、预测等工作。本部分着重讲解检验，预测放在时间序列部分中。所谓检验，既有在既定模型下对系数的检验，又有对模型本身的检验。下面分别说明。

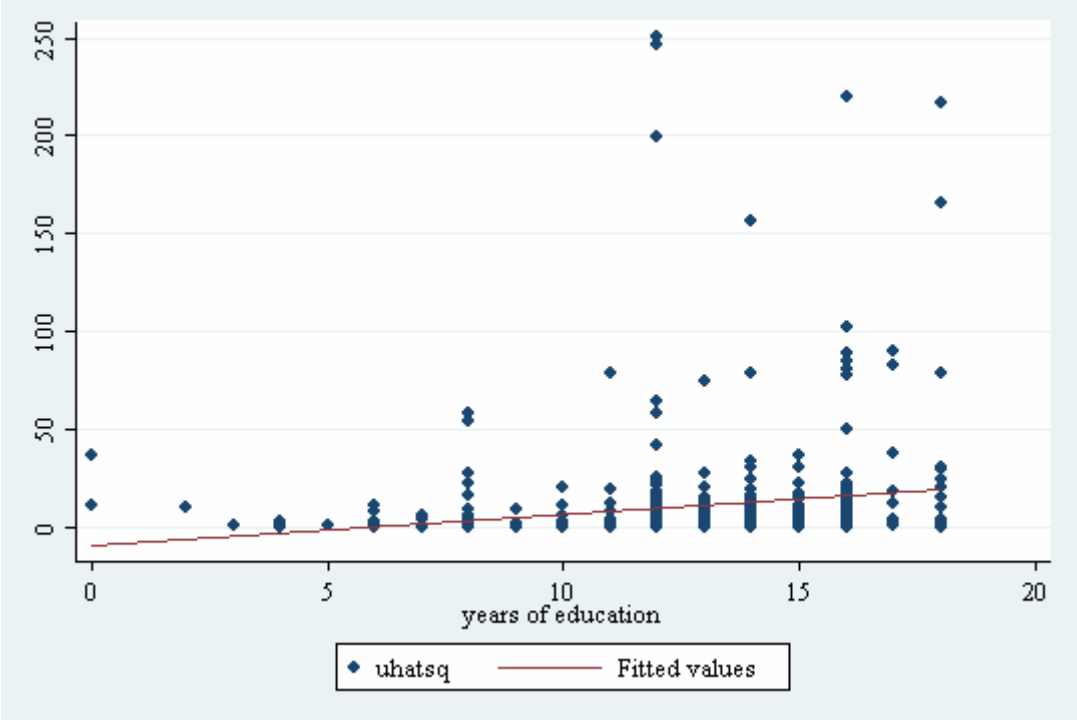
对系数的检验，无非包括对单变量系数的检验和对多变量系数的检验。首先讲单变量检验。如果想检验某变量系数的显著性，可以直接看回归结果中的 p 值，也可以用 test 命令。例如，想检验 educ 的系数是否显著，可以运行“test educ”。与回归表格中给出的 t 统计量不同，test 命令给出的是 F 统计量。如果想检验 educ 的系数是否等于一个常数（比如 1），可以运行“test educ = 1”或者“test _b[educ] = 1”。在 Stata 里，_b[educ]返回的是 educ 的系数估计值，并且可以直接作为一个数字参加运算。比如，我想计算 educ 的系数和 exper 的系数相加等于多少，可以运行“display _b[educ] + _b[exper]”，屏幕上会直接显示运算结果。接下来是多变量检验。如果想检验系数的联合显著性，可以直接看回归结果中的 F 统计量的值，也可以运行“test educ exper”。如果想检验“educ 的系数是否等于 0.5，并且 exper 的系数是否等于 0.1”，可以运行“test (educ = 0.5) (exper = 0.1)”。此外，诸如“test educ = exper”，“test educ = exper = 1”，“test educ + exper = 1”等命令可以用来检验“educ 和 exper 的系数是否相等”、“educ 和 exper 的系数是否都等于 1”和“educ 的系数和 exper 的系数的和是否为 1”等各种原假设。详情请看 test 命令的帮助文件。

对模型本身的检验有许多。如果你想检验模型是否遗漏了高次项（比如方程中是否应当包括工作经验的平方项），可以运行“ovtest”。 这条命令使用的是 RESET 检验。如果想检验模型是否存在异方差，可以运行“hettest”。这条命令使用的是 Breusch-Pagan 检验和 Cook-Weisberg 检验。至于序列相关检验，在下面的时间序列部分中会提到。如果怀疑模型存在异方差和序列相关等问题，可以命令 Stata 算出稳健的标准误，在回归命令的选项中加入“r”即可。

有时，做完回归后，对残差进行分析很有必要。首先，你要生成残差那一列数据。在运行完回归后，运行“predict uhat, r”，即可生成残差，且残差的变量名叫“uhat”。如果你想通过画图来看看是否存在异方差现象（比如想看看误差项的方差是否是 educ 的函数），可以运行下面的命令：

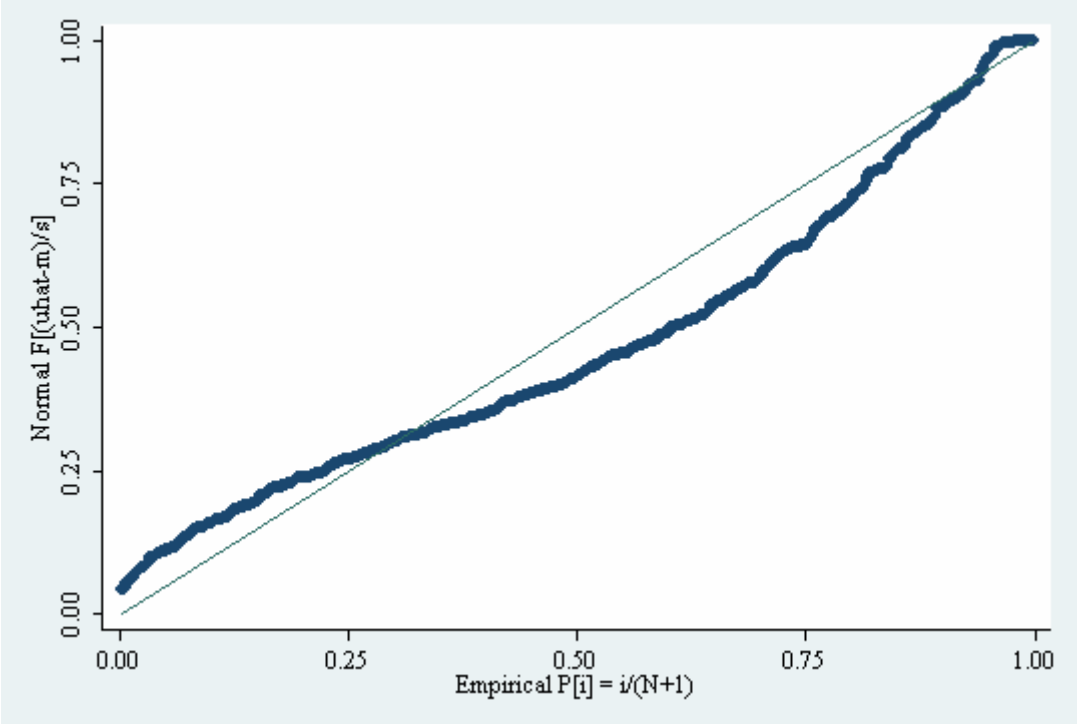
```
gen uhatsq = uhat^2
twoway scatter uhatsq educ || lfit uhatsq educ
```

第一条命令生成残差的平方项这个变量；第二条命令画出残差的平方和教育的关系，并作线性拟合。得到：



可以看出，残差的平方与教育水平之间可能存在某种关系，进而可能存在异方差现象。

如果想看看误差项的正态分布假设是否成立，可以通过残差来近似地看。运行“pnorm uhat”，得到：



散点连成的曲线越接近于 45 度线，就说明残差的分布越接近于正态分布。由上图看出，正态分布的假设有些牵强。

最后，讲解一下虚拟变量作为自变量时的相关命令。比如，我认为在控制了教育水平和工作经验的情况下，工资还会受性别的影响，因此我在原来的工资方程右边再加入表示性别的虚拟变量（female）。在 Stata 操作时，并无不同，直接加入这个变量即可（即运行“reg wage educ exper female”）。如果我认为教育对工资的回报本身还存在性别差异，那么需要在工资方程右边再加入教育水平和性别的交叉相乘项（educ*female）。在 Stata 处理时，可以生成一列新变量（比如叫 edufem），等于 educ*female，然后再把这个新变量加入自变量中（即运行“reg wage educ exper female edufem”）。另外一种方法是运行“xi: reg wage exper i.female*educ”，可得：

```
. xi: reg wage exper i.female*educ
i.female          _Ifemale_0-1        (naturally coded; _Ifemale_0 omitted)
i.female*educ     _IfemXeduc_#        (coded as above)

      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  4,   521) =   58.78
       Model |  2226.52557     4  556.631393            Prob > F      =  0.0000
    Residual |  4933.88872   521  9.47003593            R-squared     =  0.3109
-------------+------------------------------           Adj R-squared =  0.3057
       Total |  7160.41429   525  13.6388844            Root MSE      =  3.0773

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       exper |   .0647214   .0104068     6.22   0.000      .044277    .0851658
  _Ifemale_1 |  -.7580393    1.28162    -0.59   0.554    -3.275817    1.759739
       educ |   .6462078    .064354    10.04   0.000     .5197825     .772633
_IfemXeduc_1 |  -.1117433   .1001751    -1.12   0.265      -.30854    .0850535
       _cons |  -2.300828   .9085445    -2.53   0.012    -4.085688   -.5159667
------------------------------------------------------------------------------
```

“i.female*educ”这一项就表示把 educ, female 和 educ*female 这三个变量加入回归方程；所以有了这一项，就不必在 Stata 命令中单独加入 female 和 educ 了。“i”是和分类变量相关的运算符，需要放在分类变量（female）的前面。“xi”是一种前缀，表示后面要使用“i”这个运算符；前缀和主命令之间用“:”分隔。Stata 中有许多特定的前缀，除了“xi”之外，还有“by”,“cap”等等；详情请参见 prefix 的帮助文件。接下来看回归结果怎样解释。“_Ifemale_1”的系数是 -0.7580393，这表明在其他条件都不变的情况下，female 这个变量取 1 的群体的平均工资比 female 取 0 的群体的平均工资大约低 0.758 个单位（也即在其他条件都不变的情况下，样本中女性的平均工资比男性的平均工资低大约 0.758 个单位）。类似，“_Ifem*educ_1”的系数的含义是，在其他条件都不变时，样本中女性多受一年教育所得的回报比男性少大约 0.112 个单位。但是，回归结果显示，这两个变量的系数都不显著，一个可能的原因是多重共线性（用 corr 命令，得到这两个变量的相关系数超过了 0.9），因此需要去掉一个变量。如果想去掉交叉相乘项，则上文已给出了相应的命令；如果想去掉 female 这一项但是保留交叉相乘项，可以运行“xi: reg wage exper i.female|educ”，可得：

```
. xi: reg wage exper i.female|educ
i.female          _Ifemale_0-1      (naturally coded; _Ifemale_0 omitted)
i.female|educ     _IfemXeduc_#      (coded as above)

      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  3,   522) =   78.35
       Model |  2223.21261     3   741.07087            Prob > F      =  0.0000
    Residual |  4937.20168   522  9.45824077            R-squared     =  0.3105
-------------+------------------------------           Adj R-squared =  0.3065
       Total |  7160.41429   525  13.6388844            Root MSE      =  3.0754

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       exper |   .0650616   .0103844     6.27   0.000     .0446612    .0854619
        educ |   .6694726   .0509024    13.15   0.000     .5694738    .7694713
_IfemXeduc_1 |  -.1696616   .0211098    -8.04   0.000    -.2111321   -.1281911
       _cons |  -2.620268   .7301383    -3.59   0.000    -4.054639   -1.185898
------------------------------------------------------------------------------
```

从而，系数都显著了。

重新回到最经典的工资方程。如果我认为在控制了教育和工作经验的情况下，工资还有地域性差异，那么就有必要加入表示地域的变量。原数据中有三个虚拟变量表示地域：northcen 表示是否在北部，south 表示是否在南部，west 表示是否在西部；而这三个虚拟变量都为 0，则表明在东部。运行“reg wage educ exper northcen south west”即可。数据中还有一个变量，叫“region”。它取四个值（1、2、3、4），分别表示在北部、南部、西部和东部。这个变量是和表示地域的三个虚拟变量相对应的；原数据中并没有，是我为了说明问题而额外生成的。我的问题是：如果原数据中没有那三个虚拟变量，你如何控制地域这个因素？第一种答案是直接把 region 这个变量加到回归式里。这个答案不妥。因为 region 的取值只是表示分类，并没有实际的含义。如果直接把 region 加入回归式，其系数必然无法解释。第二种答案是根据 region 这个变量生成三个表示地域的虚拟变量；可以，但有些麻烦。第三种答案是直接运行“xi: reg wage educ exper i.region”，可得：

```
. xi: reg  wage educ exper i.region
i.region          _Iregion_1-4        (naturally coded; _Iregion_1 omitted)

      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  5,   520) =   32.60
       Model |  1708.72096     5  341.744191            Prob > F      =  0.0000
    Residual |  5451.69333   520  10.4840256            R-squared     =  0.2386
-------------+------------------------------           Adj R-squared =  0.2313
       Total |  7160.41429   525  13.6388844            Root MSE      =  3.2379

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .6392048    .053819    11.88   0.000     .5334755    .7449341
       exper |   .0714109   .0109248     6.54   0.000     .0499487    .0928732
  _Iregion_2 |   .0937112   .3700084     0.25   0.800    -.6331837    .8206062
  _Iregion_3 |   1.081413   .4443572     2.43   0.015     .2084572    1.954369
  _Iregion_4 |   .7781327   .4105822     1.90   0.059    -.0284711    1.584736
       _cons |  -3.740122   .8192093    -4.57   0.000    -5.349489   -2.130756
------------------------------------------------------------------------------
```

“i.region”表示把根据 region 生成的虚拟变量都加入回归式。因为升成的四个虚拟变量都加入会产生完全共线性，所以系统自动忽略了其中的一个虚拟变量（即与 region 取 1 相对应的虚拟变量）。

进一步，如果我认为不同地域的工资性别歧视的程度也不同，就有必要加入性别和地域的交叉相乘项。运行“xi: reg wage educ exper i.region*i.female”，可得：

```
. xi: reg wage educ exper i.region*i.female
i.region          _Iregion_1-4        (naturally coded; _Iregion_1 omitted)
i.female          _Ifemale_0-1        (naturally coded; _Ifemale_0 omitted)
i.reg~n*i.fem~e   _IregXfem_#_#       (coded as above)

      Source |       SS       df       MS              Number of obs =     526
-------------+------------------------------           F(  9,   516) =   28.23
       Model |  2362.46519     9  262.496133            Prob > F      =  0.0000
    Residual |   4797.9491   516  9.29835097            R-squared     =  0.3299
-------------+------------------------------           Adj R-squared =  0.3182
       Total |  7160.41429   525  13.6388844            Root MSE      =  3.0493

------------------------------------------------------------------------------
        wage |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        educ |   .5904162   .0511095    11.55   0.000     .4900079    .6908246
       exper |   .0661145   .0103542     6.39   0.000     .0457729     .086456
  _Iregion_2 |  -.0898068   .4819704    -0.19   0.852    -1.036672    .8570587
  _Iregion_3 |   1.648303   .6060545     2.72   0.007     .4576657    2.838941
  _Iregion_4 |    1.12201   .5367421     2.09   0.037     .0675418    2.176479
  _Ifemale_1 |  -1.906438    .534858    -3.56   0.000    -2.957205   -.8556713
_IregXfe~2_1 |   .1539577   .6972731     0.22   0.825    -1.215885    1.523801
_IregXfe~3_1 |  -.9103599   .8407124    -1.08   0.279       -2.562    .7412801
_IregXfe~4_1 |   -.864374   .7752502    -1.11   0.265    -2.387409    .6586608
       _cons |  -2.082672   .8446005    -2.47   0.014     -3.74195   -.4233932
------------------------------------------------------------------------------
```

在 region 和 female 前面都加“i”，是因为两者都是分类变量。“i.region*i.female”意味着加入表示地域的虚拟变量以及地域和性别的交叉相乘项；如果是

“i.female*i.region”，意味着加入表示性别的虚拟变量以及地域和性别的交叉相乘项。在地域和性别的交叉相乘项存在的情况下，不能同时出现表示地域的虚拟变量和表示性别的虚拟变量，因为存在完全共线性。

以上所述的基于截面数据的绝大部分操作都可以用于对时间序列数据的操作，下面再补充讲述时间序列数据处理过程中的一些常见操作。以“WAGEPRC.dta”为例。

一个数据导入 Stata 后，Stata 并不知道这个数据是截面数据还是时间序列数据。因此，在应用许多时间序列特有的操作前，必须告诉 Stata——这个数据是时间序列数据。导入“WAGEPRC.dta”后，运行“tsset t”，可得：

```
. tsset t
        time variable:  t, 1 to 286
```

tsset 是“声名这个数据是时间序列数据”的命令，而 t 是时间序列数据中表示时间的变量。可见，这个时间序列数据共有 286 期。

Stata 允许变量使用下标，这对时间序列来说非常方便。比如，我想生成一列新变量，是滞后一期的消费价格指数（price），可以运行“gen price_1 = price[_n-1]”。如果滞后 t 期，将下标改为“_n-t”即可。

Stata 还允许使用滞后算子“L.”。因此，运行“gen price_1 = L.price”也可以生成滞后一期的消费价格指数。如果滞后 t 期，前面的算子就需变成“Lt.”。其他的算子（超前算子、差分算子等）请见 varlist 的帮助文件。

接下来介绍时间序列数据的回归。时间序列有许多特有的回归模型，下面主要以简单的 ARMA(1, 1)模型为例。至于时间序列的其他模型（ARCH 系列模型，VAR 模型等）请参见 time 的帮助文件。

运行“arima price if t<=100, ar(1) ma(1)”，可以在前 100 期中估计关于 price 的 ARMA(1, 1)模型：

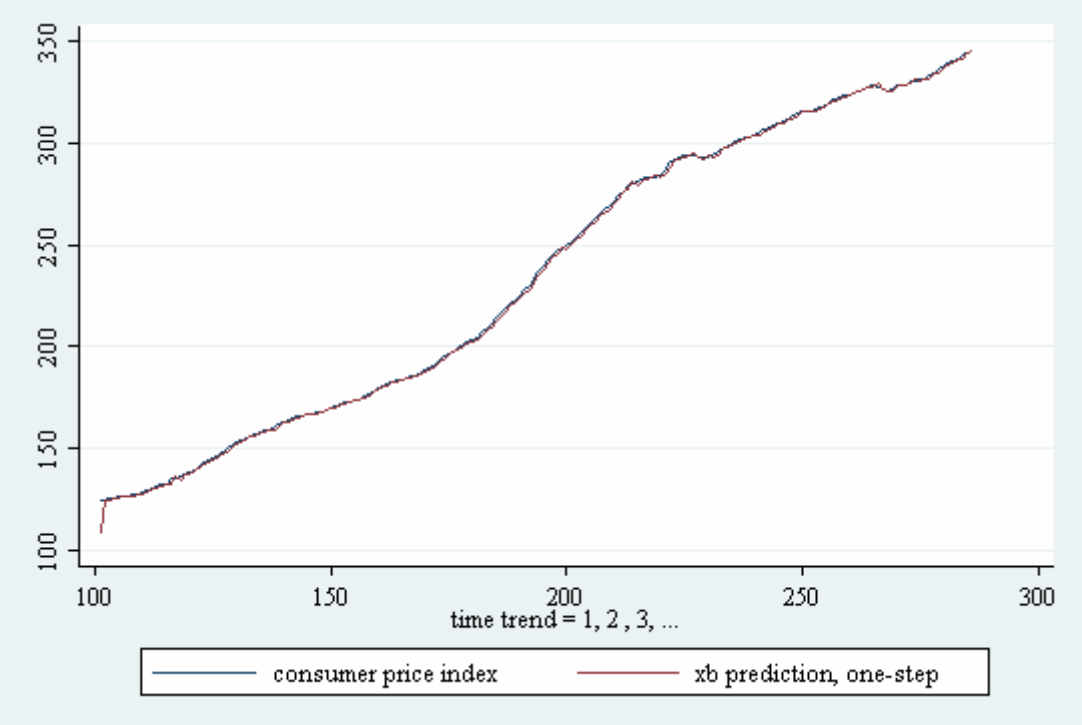
```
ARIMA regression

Sample:  1 to 100                               Number of obs      =       100
                                                Wald chi2(2)       =  42340.04
Log likelihood =  -24.3142                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
       price |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
price        |
       _cons |   108.5279   15.61704     6.95   0.000     77.91906    139.1367
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .9995704   .0048726   205.14   0.000     .9900203     1.00912
          ma |
         L1. |   .5596707   .0977759     5.72   0.000     .3680336    .7513079
-------------+----------------------------------------------------------------
       /sigma |   .2959937   .0245136    12.07   0.000      .247948    .3440395
------------------------------------------------------------------------------
```

ARIMA（Autoregressive Integrated Moving Average）模型是一个变化很多的模型，ARMA 模型可以算作它的特例；所以 Stata 中没有专门的 ARMA 模型的命令，而是将其“嵌套”在 ARIMA 模型的命令中。命令逗号后面的“ar(.)”和“ma(.)”用来表示 ARMA 模型的阶数。

预测是时间序列中的重要内容。上面对前 100 期的 price 进行了回归。基于前 100 期的回归系数能否很好地预测后面各期的 price 的值呢？运行下面的命令：

```
predict pricehat if t>100, xb
twoway line price t if t>100 || line pricehat t if t>100
```

第一条命令是对因变量作预测的命令。前文讲过，得到残差也是用 predict 命令，只不过后面的选项为“r”；而要得到因变量的拟合值，后面的选项变为“xb”。第二条命令是将后面各期的 price 的真实值和预测值画在同一幅图中进行比较。如果要在同一幅图中画多条线，中间可以用“||”间隔，也可以将画每一条线的命令用圆括号括起来。下图即为得到的图：



可见，ARMR(1, 1)模型虽然简单，但预测的效果很好。需要说明的是，预测的相关操作在截面数据中也是经常使用的。

时间序列回归中还有一类特别的问题——序列相关。首先运行“reg price L.price”，对 price 做一阶自回归（AR(1)模型），然后检验这个模型中是否存在序列相关的问题。运行“dwstat”，可以用 Durbin-Watson d 统计量去检验序列相关。还可以运行“durbina”，“bgodfrey”等命令，即利用 Dubin 替代检验、Breusch-Godfrey 检验等方法检验序列相关。其他类的检验（如单位根检验、条件异方差检验等）请参照 time 的帮助文件。

处理序列相关的办法有许多，如果想用准差分法修正，可以运行“prais price L.price”，得到：

```
. prais price L.price

Iteration 0:  rho = 0.0000
Iteration 1:  rho = 0.6917
Iteration 2:  rho = 0.6919
Iteration 3:  rho = 0.6919

Prais-Winsten AR(1) regression -- iterated estimates

      Source |       SS       df       MS              Number of obs =     285
-------------+------------------------------           F(  1,   283) =       .
       Model |  197738.039     1  197738.039            Prob > F      =  0.0000
    Residual |  79.3222965   283    .2802908            R-squared     =  0.9996
-------------+------------------------------           Adj R-squared =  0.9996
       Total |  197817.362   284  696.540006            Root MSE      =  .52942

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |
         L1. |   1.003408   .0011797   850.56   0.000     1.001085     1.00573
       _cons |   .2332372   .2453489     0.95   0.343     -.249703    .7161775
-------------+----------------------------------------------------------------
         rho |   .6918516
------------------------------------------------------------------------------
Durbin-Watson statistic (original)    0.614682
Durbin-Watson statistic (transformed) 2.140098
```

最前面几行是用迭代的方法得到的误差项一阶自相关的系数，即所谓的 $\hat{\rho}$。中间一大部分即为修正后的回归结果。最下面两行列出了修正前后的 Durbin-Watson d 统计量的值。可见，修正之后，序列相关问题被大大减弱了。

回归分析是计量经济学的核心，而本部分所讲只是一点皮毛。更艰深的内容（如和 limited dependent variable 相关的回归模型、2SLS、面板数据的回归模型等）暂时按下不表。