Lab4: Data Management(II)

Introduction to Econometrics,Fall 2020

Yi Wang

Nanjing University

14/10/2020

Section 1

A Brief Review

A Brief Review

- stata系统目录的设定
  - 显示当前系统目录的设定:*-sysdir-*
  - 设定外部命令的存放地址:

```
sysdir set PLUS "D:\stata16\ado\plus"
```

  - Dos命令：*-mkdir-,-dir-,-rm-,-erase-,-!del-*
- Stata外部命令
  - 下载安装:*-help-,-ssc install-,-findit-*
  - 查询:*-ado-,-which-*

A Brief Review

- stata工作路径设置
  - 查看当前工作路径：*-pwd-*,*-cd-*
  - 进入（修改）指定文件夹:

```
cd D:\Teaching\Stata\lab4
```

  - 打开当前文件夹:*-cdout-*

A Brief Review

- 数据导入导出
  - 导入：*-import-*
  - 导出：*-export-*
  - 存储：*-save-*

A Brief Review

- 变量
  - 变量名称基本规则
  - 查看数据结构：*-des-*
  - 变量的存储类型：
    - ★ 字符型数据
    - ★ 数值型数据（整数、小数）
    - ★ 缺失数据
  - 更改变量的存储类型：*-recast-*
  - 定义变量的显示格式：*-format-*

A Brief Review

- 变量
  - 样本标签:

```
label data "这是一份汽车价格资料"
```

  - 变量标签:

```
label var price "汽车价格"
```

  - 值标签:

```
label define repair 1 "好" 2 "较好" 3 "中" 4 "较
差" 5 "差"
label values rep78 repair
```

  - 管理值标签:*-labelbook-*,*-label list-*,*-label drop-*

A Brief Review

- 基本统计量
  - *-summarize-*,*-codebook-*
  - 列表统计:*-table-*, *-tabulate-*,*-tabstat-*
  - 基本图形分析:
    直方图:*-histogram-*
    密度函数图:*-kdensity-*
    散点图:*-scatter-*
    相关系数矩阵:*-graph matrix-*

A Brief Review

- Do Files
  - 打开
  - 新建
  - 执行
  - 注释（三种方式）
  - 断行（三种方式）
- Log File
  - 新建：*-log using-*
  - 暂停录制：*-log off-*
  - 继续录制：*-log on-*
  - 结束录制：*-log close-*

A Brief Review

- 数据清理
  - 建立新变量：*-gen-*
  - 变量重命名：*-rename-*
  - 修改观察值：*-replace-*(*if*/*in*限定样本范围)
  - 变量与样本的保存：*-keep-*
  - 变量与样本的删除：*-drop-*
  - 变量的移动：*-order-*
  - 变量的克隆：*-clonevar-*
  - 样本的排序：*-sort-*

A Brief Review

- More About Creating Variables
  - _n和_N
  - 生成虚拟变量：*-replace-*，*-tab-*，特定函数
  - 将连续变量转为类别变量：等分样本group()函数，指定区间*-recode-*
  - 交叉类别变量的生成：*-xgroup-*
  - *-egen-*命令

A Brief Review

- 分组统计
  - 单维分组，二维、三维分组，多维分组
  - 转换数据为分组统计量：*-collapse-*
  - 分组统计图：
    柱状图：*-graph bar-*
    箱形图：*-graph box-*

A Brief Review

- Duplicate Observations
  - 检查重复样本：*-duplicates list-*，*-duplicates report-*
  - 标记重复样本：*-duplicates tag-*
  - 删除重复样本：*-duplicates drop-*
- Missing Values
  - 查找缺漏值：*-misstable-*
  - 删除缺漏值：*drop if missing(. . .)*
- Outliers
  - 离群值的影响
  - 查找离群值：*-adjacent-*,箱型图
  - 删除离群值
  - 处理：取对数，缩尾(*-winsor-*)，截尾(*-winsor2-*)

A Brief Review

- Database Append and Merge
  - 数据纵向合并：*-append-*
  - 数据横向合并：*-merge-*(1:1合并,1:m合并,m:1合并)
  - 配对合并：*-joinby-*

Section 2

Data Management(Supplement)

Subsection 1

Missing Values

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 数值型缺漏值-*mvdecode*-

```
. cd D:\Teaching\Stata\lab4
. shellout d_m1.txt

. insheet using d_m1.txt, clear
(3 vars, 4 obs)
. list
```

	y	x1	x2
1.	2	-97	-999
2.	6	2	3
3.	3	5	-999
4.	5	-97	0

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 数值型缺漏值-*mvdecode*-

```
. mvdecode x1, mv(-97)   //重新定义某个变量的缺漏值
         x1: 2 missing values generated
. list
```

	y	x1	x2
1.	2	.	-999
2.	6	2	3
3.	3	5	-999
4.	5	.	0

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 数值型缺漏值-*mvdecode*-

```
. insheet using d_m1.txt, clear
(3 vars, 4 obs)
. mvdecode _all, mv(-97 -999)  //重新定义全部变量的缺漏值
         x1: 2 missing values generated
         x2: 2 missing values generated
. list
```

	y	x1	x2
1.	2	.	.
2.	6	2	3
3.	3	5	.
4.	5	.	0

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 文字型缺漏值

```
. shellout d_m2.txt

. insheet using d_m2.txt, clear
(3 vars, 4 obs)
. list
```

	y	x1	x2
1.	2	N/A	.
2.	6	2	3
3.	3	5	.
4.	5	N/A	0

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 文字型缺漏值

```
. cap replace x1 = . if x1== "N/A"  //错误方式
. replace x1 ="." if x1== "N/A"
(2 real changes made)


. des
Contains data
  obs:             4
 vars:             3
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
y                 byte    %8.0g
x1                str3    %9s
x2                byte    %8.0g
-------------------------------------------------------------------------------
Sorted by:
     Note: Dataset has changed since last saved.
```

Data Management(Supplement)

- Missing Values
  - 1.缺漏值的标记
    - ★ 文字型缺漏值

```
. gen x1_new = real(x1)
(2 missing values generated)
. list
```

	y	x1	x2	x1_new
1.	2	.	.	.
2.	6	2	3	2
3.	3	5	.	5
4.	5	.	0	.

Data Management(Supplement)

- Missing Values
  - 2.缺失值产生的原因
    - ★ **机械原因**：由于机械原因导致的数据收集或保存失败造成的数据缺失。
    - ★ **人为原因**：由于人的主观失误、历史局限或有意隐瞒造成的数据缺失。

      1)在市场调查中被访人拒绝透露相关问题的答案。
      2)回答的问题是无效的。
      3)数据录入人员失误漏录了数据。

Data Management(Supplement)

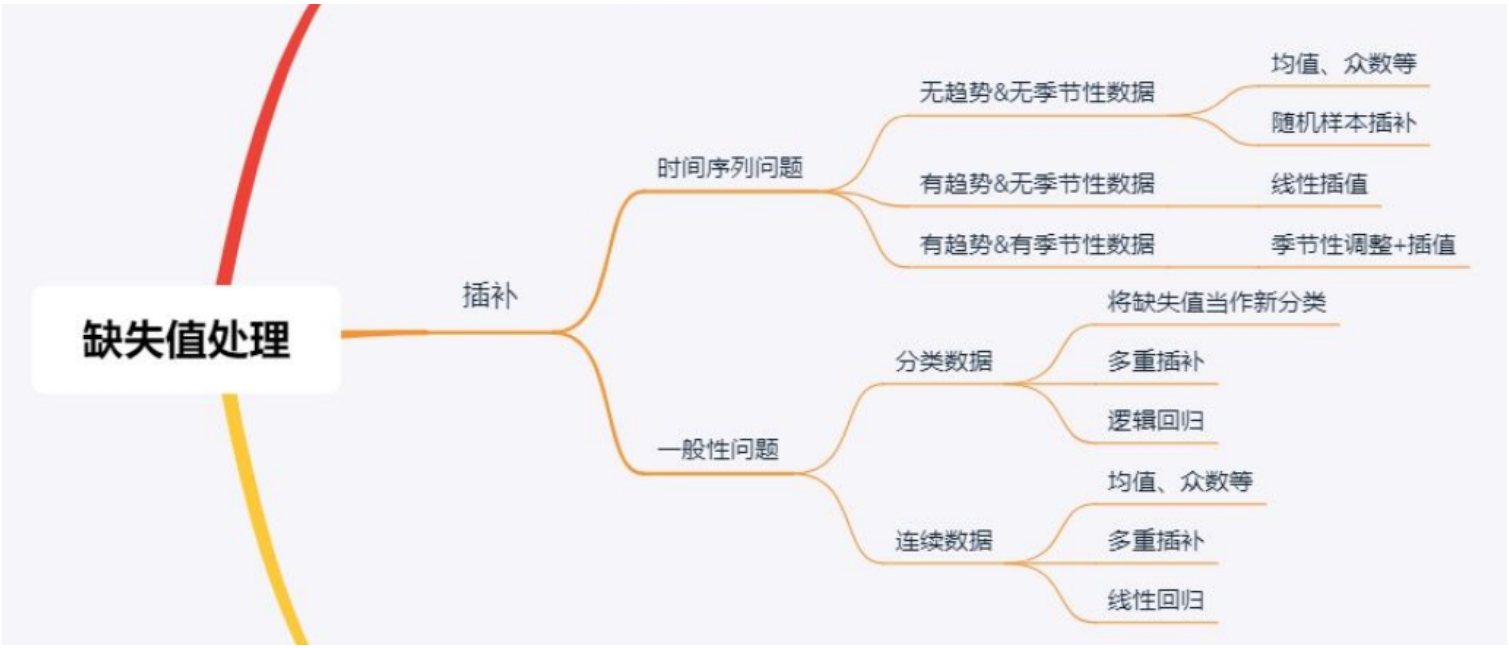
- Missing Values
  - 2.缺失值的类型
    - ★ **MCAR(Missing Completely at Random)**–完全随机缺失
      数据的缺失是随机的，数据的缺失**不依赖于任何**不完全变量或完全变量。不影响样本的无偏性。
    - ★ **MAR(Missing at Random)**–随机缺失
      数据的缺失不是完全随机的，即该类数据的缺失依赖于**其他完全变量**。
    - ★ **MNAR(Missing Not at Random)**–完全非随机缺失
      数据的缺失依赖于**不完全变量自身**。

      分两种情况：
      1)缺失值取决于其假设值（例如，高收入人群通常不希望在调查中透露他们的收入）；
      2)缺失值取决于其他变量值（假设女性通常不想透露她们的年龄，则年龄变量缺失值受性别变量的影响）。

Data Management(Supplement)

- Missing Values
  - 3.缺失值的处理方法
    - ★ 总体上来，分为**删除**存在缺失值的变量、样本和**缺失值插补**。
    - ★ **MCAR和MAR:**
      二者都可以根据缺失值出现情况删除缺失值的数据，MAR可以通过已知变量对缺失值进行估计。
      MNAR:
      删除包含缺失值的数据可能会导致模型出现偏差，对数据进行填充也需要格外谨慎。
    - ★ 当**缺失数据所占比例较大**，特别当**数据MNAR**时，**直接删除**可能导致丢失大量数据或引出错误结论。

- Missing Values
  - 4.缺漏值的填补
    - ★ 使用可能值插补缺失值：
      思想来源是以最可能的值来插补缺失值比全部删除不完全样本所产生的信息丢失要少。



Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 对于Panel Data或一些特殊的资料，可采用前向或后向非缺漏值填补。
    - ★ 单一缺漏值的填补

```
. use d_m3.dta, clear
. list
```

	t	x
1.	1	42
2.	2	.
3.	3	56
4.	4	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 单一缺漏值的填补

```
. sort t                                  //排序
. replace x = x[_n-1] if x==.             //使用该列的前一个数值填补
(1 real change made)
. list
```

	t	x
1.	1	42
2.	2	42
3.	3	56
4.	4	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 单一缺漏值的填补

```
. use d_m3.dta, clear
. list
```

	t	x
1.	1	42
2.	2	.
3.	3	56
4.	4	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 单一缺漏值的填补

```
. sort t                                    //排序
. replace x = x[_n+1] if missing(x) //使用该列的后一个数值填补
(1 real change made)
. list
```

	t	x
1.	1	42
2.	2	56
3.	3	56
4.	4	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 多个缺漏值的填补<一>

```
. use d_m4.dta, clear
. list
```

	t	x
1.	1	42
2.	2	.
3.	3	.
4.	4	56
5.	5	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 多个缺漏值的填补<一>

```
. sort t
. replace x = x[_n-1] if mi(x)  //依次向前替换
(2 real changes made)
. list
```

	t	x
1.	1	42
2.	2	42
3.	3	42
4.	4	56
5.	5	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 多个缺漏值的填补<二>

```
. use d_m4.dta, clear
. list
```

	t	x
1.	1	42
2.	2	.
3.	3	.
4.	4	56
5.	5	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 多个缺漏值的填补<二>

```
. gsort -t
. list
```

	t	x
1.	5	67
2.	4	56
3.	3	.
4.	2	.
5.	1	42

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ 多个缺漏值的填补<二>

```
. replace x = x[_n-1] if mi(x)  //依次向前替换
(2 real changes made)
. sort t
. list
```

	t	x
1.	1	42
2.	2	56
3.	3	56
4.	4	56
5.	5	67

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ Panel Data缺漏值的填补

```
. use d_m5.dta, clear
. list , sep(4)
```

	id	year	x
1.	1	2000	34
2.	1	2001	43
3.	1	2002	.
4.	1	2003	60
5.	2	2000	12
6.	2	2001	.
7.	2	2002	.
8.	2	2003	40

Data Management(Supplement)

- Missing Values
  - 4.缺漏值的填补
    - ★ Panel Data缺漏值的填补

```
. xtset id year
       panel variable:  id (strongly balanced)
        time variable:  year, 2000 to 2003
                delta:  1 unit
. by id: replace x = L.x if mi(x)
(3 real changes made)
. list, sep(4)
```

	id	year	x
1.	1	2000	34
2.	1	2001	43
3.	1	2002	43
4.	1	2003	60
5.	2	2000	12
6.	2	2001	12
7.	2	2002	12
8.	2	2003	40

Subsection 2

Transpose of the data:-*xpose*-

Data Management(Supplement)

- 样本的转置*-xpose-*

```
. use d_xp.dta, clear
. list v1-v5
```

	v1	v2	v3	v4	v5
1.	20010102	20010103	20010104	20010105	20010108
2.	1320.28	1283.27	1347.5601	1333.34	1298.35
3.	1283.27	1347.5601	1333.34	1298.35	1295.86

Data Management(Supplement)

- 样本的转置*-xpose-*

```
. xpose, clear // clear必须加
```

```
. rename v1 date
. rename v2 open
. rename v3 close
```

Data Management(Supplement)

- 样本的转置*-xpose-*

```
. list
```

	date	open	close
1.	2.00e+07	1320.28	1283.27
2.	2.00e+07	1283.27	1347.56
3.	2.00e+07	1347.56	1333.34
4.	2.00e+07	1333.34	1298.35
5.	2.00e+07	1298.35	1295.86
6.	2.00e+07	1295.86	1300.8
7.	2.00e+07	1300.8	1313.27
8.	2.00e+07	1313.27	1326.82
9.	2.00e+07	1326.82	1318.55
10.	2.00e+07	1318.32	1326.65
11.	2.00e+07	1326.65	1329.47
12.	2.00e+07	1329.89	1347.97
13.	2.00e+07	1347.97	1342.54
14.	2.00e+07	1342.54	1342.9
15.	2.00e+07	1342.9	1360.4

Data Management(Supplement)

- 样本的转置*-xpose-*

```
. xpose , clear varname   //varname增加一列变量名
. save d_xp1.dta, replace //另存一份数据，因为原始数据已被修改
file d_xp1.dta saved


. list v1-v5 _varname
```

	v1	v2	v3	v4	v5	_varname
1.	2.00e+07	2.00e+07	2.00e+07	2.00e+07	2.00e+07	date
2.	1320.28	1283.27	1347.56	1333.34	1298.35	open
3.	1283.27	1347.56	1333.34	1298.35	1295.86	close

Subsection 3

Transformations of the data:-*reshape*-

Data Management(Supplement)

- 数据的横纵变换-*reshape*-
  - 原始资料 -wide-型数据

		income			unemployee		
id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1	0	5000	5500	6000	0	1	0
2	1	2000	2200	3300	1	0	0
3	0	3000	2000	1000	0	0	1

- 数据的横纵变换-*reshape*-
  - 最终需要的资料(panel data) -long-型数据

id	year	sex	inc	ue
1	1980	0	5000	0
1	1981	0	5500	1
1	1982	0	6000	0
2	1980	1	2000	1
2	1981	1	2200	0
2	1982	1	3300	0
3	1980	0	3000	0
3	1981	0	2000	0
3	1982	0	1000	1

Data Management(Supplement)

- 数据的横纵变换-reshape-
  - wide –> long

```
. use d_rs.dta, clear
. list
```

	id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1.	1	0	5000	5500	6000	0	1	0
2.	2	1	2000	2200	3300	1	0	0
3.	3	0	3000	2000	1000	0	0	1

Data Management(Supplement)

- 数据的横纵变换-reshape-
  - wide –> long

```
.   reshape long inc ue, i(id) j(year)
(note: j = 80 81 82)
Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                        3   ->      9
Number of variables                   8   ->      5
j variable (3 values)                     ->   year
xij variables:
                     inc80 inc81 inc82   ->   inc
                        ue80 ue81 ue82   ->   ue
-----------------------------------------------------------------------------
. //sex不发生变化，无需转换
. //i()中选取ID变量，j() 中填写新的变量名称
```

Data Management(Supplement)

- 数据的横纵变换-reshape-
  - wide –> long

```
. list, sepby(id)
```

	id	year	sex	inc	ue
1.	1	80	0	5000	0
2.	1	81	0	5500	1
3.	1	82	0	6000	0
4.	2	80	1	2000	1
5.	2	81	1	2200	0
6.	2	82	1	3300	0
7.	3	80	0	3000	0
8.	3	81	0	2000	0
9.	3	82	0	1000	1

Data Management(Supplement)

- 数据的横纵变换-reshape-
  - wide –> long

```
. replace year = real("19" + string(year))
variable year was byte now int
(9 real changes made)
. list, sepby(id)
```

	id	year	sex	inc	ue
1.	1	1980	0	5000	0
2.	1	1981	0	5500	1
3.	1	1982	0	6000	0
4.	2	1980	1	2000	1
5.	2	1981	1	2200	0
6.	2	1982	1	3300	0
7.	3	1980	0	3000	0
8.	3	1981	0	2000	0
9.	3	1982	0	1000	1

Data Management(Supplement)

- 数据的横纵变换-reshape-
  - long –> wide

```
. reshape wide inc u@e, i(id) j(year)
(note: j = 1980 1981 1982)
Data                                long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                         9   ->      3
Number of variables                    5   ->      8
j variable (3 values)               year   ->   (dropped)
xij variables:
                                     inc   ->   inc1980 inc1981 inc1982
                                      ue   ->   u1980e u1981e u1982e
-----------------------------------------------------------------------------
```

```
. //j() 中填写已经存在的变量名称
. //变量名默认变为"Xj"
. //包含"@",表示"j"可能出现在变量名中的位置
. list
```

	id	inc1980	u1980e	inc1981	u1981e	inc1982	u1982e	sex
1.	1	5000	0	5500	1	6000	0	0
2.	2	2000	1	2200	0	3300	0	1
3.	3	3000	0	2000	0	1000	1	0

Subsection 4

Sample stacks: *-stack-*

Data Management(Supplement)

- 样本的堆砌-*stack*-

```
. use d_st.dta, clear
. list
```

	a	b	c	d
1.	1	2	3	4
2.	5	6	7	8

```
. stack a b c d, into(x) clear // 堆砌成一列,指定生成的新变量名X
. list, sepby(_stack)
```

	_stack	x
1.	1	1
2.	1	5
3.	2	2
4.	2	6
5.	3	3
6.	3	7
7.	4	4
8.	4	8

Data Management(Supplement)

- 样本的堆砌-*stack*-

```
. use d_st.dta, clear
. list
```

	a	b	c	d
1.	1	2	3	4
2.	5	6	7	8

```
. stack a b c d, into(x1 x2) clear // 堆砌成两列,指定生成的新变量名X1和X2
. list, sepby(_stack)
```

	_stack	x1	x2
1.	1	1	2
2.	1	5	6
3.	2	3	4
4.	2	7	8

Data Management(Supplement)

- 样本的堆砌*-stack-*
  - 一个实用案例

```
. *-原始样本
. use ex_stack.dta,clear


. *-堆砌样本
. stack china-unitedstates, into(lnexp) clear
```

Data Management(Supplement)

- 样本的堆砌*-stack-*
  - 一个实用案例

```
. *增加年度变量
. rename _stack id
. egen year = seq(), from(1998) to(2007)
. order id year
. xtset id year            //声明面板数据
       panel variable:  id (strongly balanced)
        time variable:  year, 1998 to 2007
                delta:  1 unit
. save temp1, replace
file temp1.dta saved
```

Data Management(Supplement)

- 样本的堆砌-*stack*-
  - 一个实用案例

```
. *取出国名
. use ex_stack.dta,clear
. drop lnexp
. mkmat _all, mat(a)    //矩阵的列名(国名)就是我们需要的
. *mat list a
. global vn: colnames a // 将国名存储于暂元 vn 中
. dis "$vn"
china austria czechrepublic denmark finland france greece germany hungary ireland japan
> poland portugal netherlands romania slovakia slovenia spain sweden turkey unitedkingd
> dstatesofamerica
```

Data Management(Supplement)

- 样本的堆砌*-stack-*
  - 一个实用案例

```
. *处理国名
. use temp1, clear
. gen country = ""
(230 missing values generated)
. local i = 1
. foreach nn of global vn{           // 1 --> "china"
  2. qui replace country="`nn´" if id==`i++´
  3. }
```

Data Management(Supplement)

- 样本的堆砌-*stack*-
  - 一个实用案例

```
. sencode country, replace //country是一个文字变量，现转化为数值变量
. labelbook
```

```
value label country
```

```
      values                                  labels
       range:  [1,23]                  string length:  [5,21]
           N:  23                 unique at full length:  yes
        gaps:  no                   unique at length 12:  yes
  missing .*:  0                             null string:  no
                                 leading/trailing blanks:  no
                                      numeric -> numeric:  no

  definition
           1   china
           2   austria
           3   czechrepublic
           4   denmark
           5   finland
           6   france
           7   greece
           8   germany
           9   hungary
          10   ireland
```

Data Management(Supplement)

- 样本的堆砌-*stack*-
  - 一个实用案例

```
. order id year country
. xtset id year
       panel variable:  id (strongly balanced)
        time variable:  year, 1998 to 2007
                delta:  1 unit


. save temp2.dta, replace
file temp2.dta saved
```

Subsection 5

Text Variables Management

- 文字变量的处理
  - 1.字符型转换成数值型-*destring*-

```
. use str1.dta, clear
. list
```

	code	year	date	size	leverage	gov
1.	600001	2004-12	2004/12/31	980.32	24%	国有
2.	600 002	2004-12	2004/9/30	143,2.23	46%	国有
3.	600 003	2004-12	2004/6/30	230,9.02	36%	国有
4.	600004	2004-12	2004/3/31	145.98	61%	民营

  - code变量由数字组成，但其类型为 str7,即为文字型变量。leverage, size, date 都存在类似的问题。

Data Management(Supplement)

- 文字变量的处理
  - 1.字符型转换成数值型*-destring-*

```
. destring code, gen(code1) ignore(" ")
code: character space removed; code1 generated as long


. destring leverage, gen(lev) percent
leverage: character % removed; lev generated as double


. destring year date size lev, replace ignore("-/,%")     //同时转换多个变量
year: character - removed; replaced as long
date: character / removed; replaced as long
size: character , removed; replaced as double
lev already numeric; no replace
```

- 文字变量的处理
  - 1.字符型转换成数值型-*destring*-

```
. list
```

	code	code1	year	date	size	leverage	lev	gov
1.	600001	600001	200412	20041231	980.32	24%	.24	国有
2.	600 002	600002	200412	2004930	1432.23	46%	.46	国有
3.	600 003	600003	200412	2004630	2309.02	36%	.36	国有
4.	600004	600004	200412	2004331	145.98	61%	.61	民营

  - *real()*函数，同样可以实现1.字符型转换成数值型的功能。

Data Management(Supplement)

- 文字变量的处理
  - 2.字符型转换类别变量*-encode-*
  - *-encode-*会自动根据文字类别编号,并设定相应的“数字-文字对应表”。

```
. encode gov, gen(gov1)
. labelbook gov1
```

```
value label gov1
```

```
      values                                  labels
       range:  [1,2]                  string length:  [6,6]
           N:  2                unique at full length:  yes
        gaps:  no                 unique at length 12:  yes
 missing .*:  0                           null string:  no
                                leading/trailing blanks:  no
                                    numeric -> numeric:  no

 definition
           1   国有
           2   民营
  variables:  gov1
```

Data Management(Supplement)

- 文字变量的处理
  - 2.字符型转换类别变量-*sencode*-
  - -*sencode*-附加replace 选项

```
. sencode gov, replace
. labelbook gov
```

```
value label gov
```

```
      values                                  labels
       range:  [1,2]                  string length:  [6,6]
           N:  2                unique at full length:  yes
        gaps:  no                 unique at length 12:  yes
  missing .*:  0                           null string:  no
                                leading/trailing blanks:  no
                                     numeric -> numeric:  no
  definition
           1    国有
           2    民营
   variables:  gov
```

Data Management(Supplement)

- 文字变量的处理
  - 3.数值型转换成字符型-*tostring*-
  - 某些情况下，先把数字转换成文字，再利用处理文字的函数进行处理比较方便。
  - 举例：年月日组合

```
. use str2.dta, clear
. tostring year day, replace
year was float now str4
day was float now str2


. gen date = year + "-" + month + "-" + day
. gen edate = date(date, "YMD")  //(days since 01jan1960)
. format edate %td
. list in 1/3
```

	id	month	day	year	date	edate
1.	123456789	jan	10	2001	2001-jan-10	10jan2001
2.	123456710	mar	20	2001	2001-mar-20	20mar2001
3.	123456711	may	30	2001	2001-may-30	30may2001

Data Management(Supplement)

- 文字变量的处理
  - 3.数值型转换成字符型-*tostring*-
  - 举例：年月日分离

```
. use str3.dta, clear
. tostring date_pub, gen(date1)
date1 generated as str8


. gen year = substr(date1, 1, 4)
. gen month = substr(date1, 5, 2)
. gen day = substr(date1, 7, 2)
```

Data Management(Supplement)

- 文字变量的处理
  - 3.数值型转换成字符型-*tostring*-
  - 举例：年月日分离

```
. destring year month day, replace
year: all characters numeric; replaced as int
month: all characters numeric; replaced as byte
day: all characters numeric; replaced as byte


. list
```

	date_pub	date1	div	year	month	day
1.	20070930	20070930	.5	2007	9	30
2.	20080823	20080823	.3	2008	8	23
3.	20090102	20090102	.7	2009	1	2
4.	20081130	20081130	.3	2008	11	30
5.	20061204	20061204	.4	2006	12	4

Data Management(Supplement)

- 文字变量的处理
  - 3.数值型转换成字符型-*decode*-

```
. use str4,clear
. decode gender,generate(gender1)  //根据gender变量重新生成一个字符型变量gender2
. codebook gender1
```

```
gender1                                                                    (un
-------------------------------------------------------------------------------

                  type:  string (str6)
        unique values:  2                        missing "":  2/1,130
           tabulation:  Freq.  Value
                            2  ""
                          433  "female"
                          695  "male"
```

Data Management(Supplement)

- 文字变量的处理
  - 4.字符型转换类别变量-*sdecode*-

-*sdecode*-附加replace 选项

```
. sdecode gender, replace
. codebook gender
-------------------------------------------------------------------------------
gender                                                                    (ur
-------------------------------------------------------------------------------

                  type:  string (str6)
        unique values:  2                        missing "":  2/1,130
           tabulation:  Freq.  Value
                            2  ""
                          433  "female"
                          695  "male"
```

- 文字变量的处理
  - 5.字符型样本值的分解-*split*-

```
. use str1.dta, clear
. list
```

	code	year	date	size	leverage	gov
1.	600001	2004-12	2004/12/31	980.32	24%	国有
2.	600 002	2004-12	2004/9/30	143,2.23	46%	国有
3.	600 003	2004-12	2004/6/30	230,9.02	36%	国有
4.	600004	2004-12	2004/3/31	145.98	61%	民营

Data Management(Supplement)

- 文字变量的处理
  - 5.字符型样本值的分解-*split*-
  - 提取年份from变量year

```
. split year, parse(-)
variables created as string:
year1  year2
. order year year1 year2
. list year*
```

	year	year1	year2
1.	2004-12	2004	12
2.	2004-12	2004	12
3.	2004-12	2004	12
4.	2004-12	2004	12

```
. gen year3 = real(year1)  //字符型转数值型存储数据
```

Data Management(Supplement)

- 文字变量的处理
  - 5.字符型样本值的分解-*split*-
  - 提取年、月、日from变量date，并转化为数值型

```
. split date,parse(/) destring ignore("/")
variables born as string:
date1  date2  date3
date1: all characters numeric; replaced as int
date2: all characters numeric; replaced as byte
date3: all characters numeric; replaced as byte
. order date date*
. list  date*
```

	date	date1	date2	date3
1.	2004/12/31	2004	12	31
2.	2004/9/30	2004	9	30
3.	2004/6/30	2004	6	30
4.	2004/3/31	2004	3	31

Data Management(Supplement)

- 文字变量的处理
  - 6.常用函数

```
help string functions
```

```
. dis length("price weight length mpg")
23
. dis wordcount("price weight length mpg") //统计变量的个数
4


. dis lower("AbCDef")
abcdef
. dis proper("mR. joHn a. sMitH") // 规整人名
Mr. John A. Smith


. dis strmatch("C51", "C")
0
. dis strmatch("C51", "C*")
1
```

- 文字变量的处理
  - 6.常用函数

```
. dis trim(" Hello World! ") // 去掉两端的空格
Hello World!

. dis ltrim(" Hello World! ") // 去掉左边的空格
Hello World!

. dis rtrim(" Hello World! ") // 去掉右边的空格
 Hello World!
```

Data Management(Supplement)

- 文字变量的处理
  - 6.常用函数

```
. dis subinstr("内 蒙 古 自治区", " ", "", .)
内蒙古自治区

  * subinstr(s, s1, s2, n)
  *    s原始字符串
  *    s1被替换的字符串
  *    s2替换成的字符串
  *    n前n个出现的目标字符，若为``.''则表示全部替换

. dis subinstr("内 蒙 古 自治区", " ", "", 1)
内蒙 古 自治区
. dis subinstr("内 蒙 古 自治区", " ", "", 2)
内蒙古 自治区
```

- 文字变量的处理
  - 7.实用案例<一>

```
. use str5,clear
. list
```

	date	sic	location
1.	20010123	A01	湖北 荆州
2.	20021231	A0599	陕西 西安
3.	20030922	C0501	广东 广州
4.	20040101	C11	河北 石家庄
5.	20050630	C4310	内蒙古 呼和浩特

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<一>
  - 处理date，分离出年月日

```
. gen year = int(date/10000)
. tostring date, gen(date1)
date1 generated as str8


. gen year1 = substr(date1,1,4)
. gen year2 = real(year1)


. gen month1 = substr(date1,5,2)
. gen month2= real(month)


. gen day1 = substr(date1,7,2)
. gen day2 = real(day)
```

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<一>
  - 处理date，分离出年月日

```
. *或等价于

. gen year3 = real(substr(date1, 1, 4))
. gen month3 = real(substr(date1, 5, 2))
. gen day3 = real(substr(date1, 7, 2))
. list year* month* day*
```

year	year1	year2	year3	month1	month2	month3	day1	day2	day3
2001	2001	2001	2001	01	1	1	23	23	23
2002	2002	2002	2002	12	12	12	31	31	31
2003	2003	2003	2003	09	9	9	22	22	22
2004	2004	2004	2004	01	1	1	01	1	1
2005	2005	2005	2005	06	6	6	30	30	30

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<一>
  - 处理sic，行业分类

```
. use str5,clear


. gen sic_c = substr(sic,1,1)
. encode sic_c,gen(sic_co)


. tab  sic_co
```

sic_co	Freq.	Percent	Cum.
A	2	40.00	40.00
C	3	60.00	100.00
Total	5	100.00	

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<一>
  - 处理location，分离城市

```
. use str5,clear
. gen province = word(location, 1)
. gen city = word(location, 2)
. list location province city
```

	location	province	city
1.	湖北 荆州	湖北	荆州
2.	陕西 西安	陕西	西安
3.	广东 广州	广东	广州
4.	河北 石家庄	河北	石家庄
5.	内蒙古 呼和浩特	内蒙古	呼和浩特

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<二>
  - 提取总行名称

```
. use str6,clear
. list in 1/10
```

	id	year	objnm
1.	890	2007	中国农业银行深圳光明支行
2.	600307	2008	招商银行股份有限公司兰州分行
3.	600589	2005	光大银行深圳罗湖支行
4.	600397	2006	中国工商银行股份有限公司萍乡分行
5.	917	2006	中国银行吉首支行
6.	31	2006	中国银行股份有限公司深圳市分行
7.	600530	2008	北京银行上海分行
8.	600198	2005	中国银行
9.	21	2006	兴业银行深圳科技支行
10.	600719	2006	中国农业银行大连市分行

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<二>
  - 提取总行名称

```
. gen bank = objnm
. replace bank="中国农业银行" if strmatch(bank,"*农业银行*")
(13 real changes made)
. replace bank="招商银行" if strmatch(bank,"*招商*")
(26 real changes made)
. replace bank="中国银行" if strmatch(bank,"*中国银行*")
(29 real changes made)
. replace bank="中国工商银行" if strmatch(bank,"*工商*")
(11 real changes made)
. replace bank="兴业银行" if strmatch(bank,"*兴业*")
(11 real changes made)
. replace bank="光大银行" if strmatch(bank,"*光大*")
(10 real changes made)
. replace bank="交通银行" if strmatch(bank,"*交通*")
(17 real changes made)
. replace bank="北京银行" if strmatch(bank,"*北京*")
(11 real changes made)
. compress
```

Data Management(Supplement)

- 文字变量的处理
  - 7.实用案例<二>
  - 提取总行名称

```
. list in 1/10
```

	id	year	objnm	bank
1.	890	2007	中国农业银行深圳光明支行	中国农业银行
2.	600307	2008	招商银行股份有限公司兰州分行	招商银行
3.	600589	2005	光大银行深圳罗湖支行	光大银行
4.	600397	2006	中国工商银行股份有限公司萍乡分行	中国工商银行
5.	917	2006	中国银行吉首支行	中国银行
6.	31	2006	中国银行股份有限公司深圳市分行	中国银行
7.	600530	2008	北京银行上海分行	北京银行
8.	600198	2005	中国银行	中国银行
9.	21	2006	兴业银行深圳科技支行	兴业银行
10.	600719	2006	中国农业银行大连市分行	中国农业银行

Section 3

Scalar and Macro

Subsection 1

Scalar

Scalar and Macro

- 单值(scalar)
  - 1.存放数值

```
. scalar a = 3
. dis a
3


. scalar b = sin(2/_pi) + (9^5)/exp(2)
. dis b
中国农业银行
```

Scalar and Macro

- 单值(scalar)
  - 2.存放字符串

```
. scalar c = "hello World! "
. dis c
hello World!
```

Scalar and Macro

- 单值(scalar)
  - 3.标示变量的特定观察值

```
. sysuse auto,clear
(1978 Automobile Data)
. dis price[5]
7827
```

Scalar and Macro

- 单值(scalar)
  - 4.标示执行命令后的单值结果

```
. sum price
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
       price |         74    6165.257    2949.496       3291      15906
. return list   //返回留存在内存中的结果
scalars:
                  r(N) =  74
              r(sum_w) =  74
               r(mean) =  6165.256756756757
                r(Var) =  8699525.974268788
                 r(sd) =  2949.495884768919
                r(min) =  3291
                r(max) =  15906
                r(sum) =  456229
. dis r(N)
74


. scalar sd = r(sd)
. dis sd
2949.4959
```

Scalar and Macro

- 单值(scalar)
  - 4.标示执行命令后的单值结果

```
. *求 mean(price) 的标准误
. *公式: s.e.(mean(price)) = s.d.(price)/sqrt(N)
. scalar se_price = r(sd)/sqrt(r(N))
. disp se_price
342.87193
```

Scalar and Macro

- 单值(scalar)
  - 4.标示执行命令后的单值结果
  - 应用：t检验

```
. scalar t_value = r(mean) / se_price
. dis t_value
17.981223
```

- 使用stata命令验证：

```
. ttest price=0
One-sample t test
```

Variable	Obs	Mean	Std. Err.	Std. Dev.	[95% Conf.	Interval]
price	74	6165.257	342.8719	2949.496	5481.914	6848.6

```
    mean = mean(price)                                     t =  17.9812
Ho: mean = 0                             degrees of freedom =       73

    Ha: mean < 0                 Ha: mean != 0                 Ha: mean > 0
 Pr(T < t) = 1.0000         Pr(|T| > |t|) = 0.0000          Pr(T > t) = 0.0000
```

Scalar and Macro

- 单值(scalar)
  - 5.单值管理

```
scalar dir
scalar list
scalar drop a
scalar drop _all
```

Subsection 2

Macro

Scalar and Macro

- 暂元(Macro)
  - 1.存放数值

```
. local a = 5
. dis `a´
5
```

Scalar and Macro

- 暂元(Macro)
  - 2.存放字符串

```
. local b "Hello!"
. dis "`b´"
Hello!
```

Scalar and Macro

- 暂元(Macro)
  - 3.存放变量名称

```
. sysuse auto, clear
(1978 Automobile Data)
. local var price weight rep78 length
. sum `var´
```

Variable	Obs	Mean	Std. Dev.	Min	Max
price	74	6165.257	2949.496	3291	15906
weight	74	3019.459	777.1936	1760	4840
rep78	69	3.405797	.9899323	1	5
length	74	187.9324	22.26634	142	233

Scalar and Macro

- 暂元(Macro)
  - 4.数学运算

```
. local c "2+2"
. dis `c´
4
. dis "`c´"
2+2


. local d = 2+2
. dis `d´
4
. dis "`d´"
4
```

Scalar and Macro

- 暂元(Macro)
  - 5.复合双引号:

`" "'--文字中包含 `' 和 ""时会用到。

```
. local e John´s "cat"
. dis `"`e´ "´
John´s "cat"
```

Scalar and Macro

- 暂元(Macro)
  - 6.暂元中的暂元：

```
. local a1 = 2
. dis `a1´
2


. local a2 "var"
. dis "`a2´"
var


. local a3 = 2*`a1´
. dis `a3´
4


. local a4 `a`a1´´
. dis "`a4´"
var
```

Scalar and Macro

- 暂元(Macro)
  - 6.暂元中的暂元：

```
. local `a2´`a1´ = 2*`a3´
. dis ``a2´`a1´´
8


. local `a`a3´´ "``a`a1´´2´"
. dis "``a`a3´´´"       //从第一个完整的 `` 开始分析
8
```

Scalar and Macro

- 暂元(Macro)
  - 7.全局暂元的定义和引用

```
. global aa "This is Stata Lab4!"
. dis "$aa"
This is Stata Lab4!


. global x1 = 5
. global x2 = 2^$x1
. dis $x2
32
```

Scalar and Macro

- 暂元(Macro)
  - 8.暂元的管理

```
macro list
macro dir
macro dir aa
macro drop x2
```