

AULA 3

Análise de Dados Legislativos e Eleitorais Utilizando o Programa Stata

Professor: Ernesto Friedrich de Lima Amaral

Email: eflamaral@gmail.com

Site do curso: www.ernestoamaral.com/stata20091.html

Data: 21/05/2009

Horário: 18:00–21:00

Questionário

Somente para a aula não ficar muito abstrata, vamos olhar um pouco o questionário da Argentina. Durante a aula, é interessante olhar o questionário para entender as variáveis.

Gráficos (continuação da Aula 2)

De uma forma geral, é bom olhar o menu *Graphics* para explorar os diversos tipos de gráficos elaborados pelo Stata. Aqui vou colocar alguns exemplos:

– Histogramas:

```
gen menem=p1406
replace menem=. if p1406==99

histogram menem, ///
discrete frequency ytitle(Frequência) xtitle(Carlos Menem) ///
title("Histograma da escala esquerda/direita de Carlos Menem," ///
"segundo legisladores", size(medsmall))
graph export "C:\cursodcp\graf\histmenem.wmf", replace
```

– Gráfico de caixa (box-plot):

```
gen pratica=p60b
replace pratica=. if p60b==99
label variable pratica "Prática religiosa"

graph hbox pratica, ///
title("Distribuição dos legisladores segundo prática religiosa" ///
, size (medsmall))
graph export "C:\cursodcp\graf\boxpratrel.wmf", replace
```

– Gráfico de dispersão (scatterplot):

```
gen idade=p63
replace idade=. if p63==99
label variable idade "Idade"

tway scatter pratica idade, ///
title("Gráfico de dispersão da idade do legislador pela prática religiosa" ///
, size (medsmall))
graph export "C:\cursodcp\graf\scatrelid.wmf", replace
```

Reorganização do arranjo (formato) dos bancos de dados

O comando "reshape" muda o formato dos bancos de dados, segundo a necessidade do estudo em questão. Este é o exemplo da ajuda do Stata:

```
(wide form)
i          ..... x_ij .....
id sex   inc80   inc81   inc82
-----
1    0    5000    5500    6000
2    1    2000    2200    3300
```

```
(long form)
i   j          x_ij
id year   sex   inc
-----
1   80     0   5000
1   81     0   5500
1   82     0   6000
2   80     1   2000
2   81     1   2200
2   82     1   3300
```

O comando para mudar o formato dos bancos de "wide" (amplo) para "long" (dados em painel) é:
`reshape long inc, i(id) j(year)`

O comando para mudar o formato dos bancos de "long" (dados em painel) para "wide" (amplo) é:
`reshape wide inc, i(id) j(year)`

Suponha que queremos saber qual a média geral que os partidos argentinos receberam na escala de esquerda (1) à direita (10) por parte dos legisladores. As variáveis que indicam as escalas esquerda/direita dos partidos vão de "p1301" a "p1305". Vamos transformar essas seis variáveis em uma variável única (p), repetindo os deputados nas linhas:

```
keep cuesti p1301-p1305
reshape long p, i(cuesti) j(partido)
replace p=. if p==98 | p==99
```

O número de observações no novo banco é igual ao número de observações originais (105) multiplicado por cinco partidos:

```
display _N
display 105*5
```

Vamos estimar a média das escalas esquerda/direita de cada partido:

```
tab partido, sum(p) mean
```

partido	Summary of p Mean
1301	6.6041667
1302	5.8157895
1303	3.4042553
1304	8.2
1305	3.9569892
Total	5.5158151

Agora queremos colocar os valores da tabela acima em nosso banco como uma nova variável:

```
sort partido
by partido: egen esqdir=mean(p)
```

Lembrem-se que usando o comando "egen" podemos calcular a média de uma série de variáveis com a opção "rowmean" (média na linha). No caso acima, o "egen" foi utilizado para estimar a média (mean) de uma única variável (p) por categorias de outra variável (partido).

Somente como exercício, vamos retornar esse banco para o formato "wide":

```
drop esqdir
reshape wide p, i(cuesti esqdir) j(partido)
```

Utilização do comando foreach para recodificação e criação de grupos de variáveis

As variáveis "p2901" a "p2910" indicam a opinião do legislador sobre o grau de intervenção que o Estado deveria assumir frente a uma série de funções tradicionais. Com o intuito de reclassificar as respostas "não sabe" (8) e "não respondeu" (9) para "missing" (.), poderíamos usar todas essas linhas de comando:

```
replace p2901=. if p2901==8 | p2901==9
replace p2902=. if p2902==8 | p2902==9
replace p2903=. if p2903==8 | p2903==9
replace p2904=. if p2904==8 | p2904==9
replace p2905=. if p2905==8 | p2905==9
replace p2906=. if p2906==8 | p2906==9
replace p2907=. if p2907==8 | p2907==9
replace p2908=. if p2908==8 | p2908==9
replace p2909=. if p2909==8 | p2909==9
replace p2910=. if p2910==8 | p2910==9
```

No entanto, podemos simplesmente escrever:

```
foreach x of varlist p29?? {
  replace `x'=. if `x'==8 | `x'==9
}
```

Depois de recodificar os valores "não sabe" e "não respondeu" das variáveis p2901 a p2910, podemos gerar novas variáveis para informar as médias das opiniões dos legisladores (sobre o grau de intervenção do Estado em certas atividades) por partido:

```
sort partido
foreach x of varlist p2901-p2910 {
  by partido: egen med`x' = mean(`x')
}

browse partido p2901-p2910 med*
browse partido p2901-p2910 med*, nolabel
```

Criação de bancos de dados agrupados por categorias de variáveis

Vamos criar e salvar um banco de dados que vai indicar o número de deputados por departamento e partido político. Abra o banco de dados original antes de rodar os comandos abaixo:

```
gen ndeputado=1
sort departam partido
collapse (count) ndeputado, by(departam partido)
save "C:\cursodcp\dados\ndep.dta", replace
browse departam partido ndeputado
```

A opção peso pode ser usada em uma tabela simples somente para certificar que o número de deputados foi calculado corretamente:

```
tab partido [fweight=ndeputado]
```

Em seguida, vamos abrir o banco de dados original novamente para calcular as médias das variáveis p2901–p2910 (opiniões sobre grau de intervenção do Estado) e p3101–p3114 (opiniões sobre relevância de problemas na Argentina) por departamento e partido. Antes de tudo, temos que reclassificar os "não sabe" e "não respondeu":

```
foreach x of varlist p29?? p31?? {
  replace `x'=. if `x'==8 | `x'==9
}

sort departam partido
collapse (mean) p29?? p31??, by(departam partido)
save "C:\cursodcp\dados\p29p31.dta", replace

merge departam partido using "C:\cursodcp\dados\ndep.dta"
tab _merge
drop _merge
save "C:\cursodcp\dados\p29p31ndep.dta", replace
```

Alocação de informações entre registros

Vamos supor que queremos colocar algumas respostas do deputado mais velho de cada partido no registro dos outros deputados daquele mesmo partido. O pressuposto é de que o deputado mais velho teria influência sobre os demais deputados de seu partido em temas polêmicos. A informação de deputado mais velho poderia ser substituída por deputado mais rico, deputado líder do partido, e assim por diante (dependendo da disponibilidade dos dados e de suas hipóteses).

Esse tipo de exercício é muito utilizado em estudos que alocam informações dos pais (renda, educação e outros) para as crianças no mesmo domicílio.

Primeiramente vamos recodificar a variável idade e ordenar o banco de dados original de forma crescente por partido e decrescente por idade (p63) com o comando "gsort":

```
use "C:\cursodcp\dados\Argentina51.dta", clear
replace p63=. if p63==99
gsort partido -p63
browse partido p63
```

Depois vamos criar uma variável dicotômica que terá valor igual a "1" para os deputados mais velhos em cada partido:

```
egen depvel=tag(partido)
browse partido p63 depvel
```

Vamos ordenar o banco por partido, já que mais adiante iremos juntá-lo a outro banco. É bom salvar esse banco com outro nome:

```
sort partido
save "C:\cursodcp\dados\argoriginal.dta", replace
```

Agora vamos criar um pequeno banco que vai ter informações somente dos deputados mais velhos de cada partido ("depvel" igual a 1). Além disso, queremos somente as variáveis "p34" (opinião sobre imposto), "p35" (opinião sobre privatizações) e "p36" (opinião sobre Mercosul):

```
keep if depvel==1
keep partido p34 p35 p36
```

Como iremos juntar esse novo banco ao banco com todos os deputados, precisamos renomear as variáveis para que elas não sejam gravadas por cima das outras:

```
rename p34 p34vel
rename p35 p35vel
rename p36 p36vel
browse
```

Lembre-se de ordenar o banco por partido e salvá-lo com outro nome:

```
sort partido
save "C:\cursodcp\dados\argvelho.dta", replace
```

Temos que abrir novamente o banco que salvamos anteriormente e juntá-lo (merge) ao banco dos deputados mais velhos:

```
use "C:\cursodcp\dados\argoriginal.dta", clear
merge partido using "C:\cursodcp\dados\argvelho.dta"
tab _merge
drop _merge
```

Por fim, seria interessante colocar valores "missing" para as novas variáveis ("p34vel", "p35vel" e "p36vel") quando se trata do deputado mais velho. Isso é feito porque as novas variáveis só fazem sentido de serem comparadas com as antigas ("p34", "p35" e "p36"), se o registro for dos demais deputados:

```
replace p34vel=. if depvel==1
replace p35vel=. if depvel==1
replace p36vel=. if depvel==1

save "C:\cursodcp\dados\argalocado.dta", replace
```