AULA 1
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/stata20092ufpe.html
Data: 21 e 24/11/2009

Entendendo o Stata

O Stata possibilita gerenciamento de dados, análise estatística e elaboração de gráficos. Existem programas para tornar o Stata mais amigável para aqueles que não gostam de trabalhar com comandos. O próprio Stata possui menus e janelas que visam facilitar seu uso. No entanto, o curso será baseado no estilo padrão de comandos.

Principais janelas do Stata: *Review*, *Variables*, *Results*, *Command* e *Do-file Editor*.

Bancos de dados em Stata possuem extensão ".dta", e programas (*syntax*) possuem extensão ".do".

O Stata trabalha com os dados copiando-os na memória RAM. Em virtude disso, bancos de dados grandes podem ser de difícil gerenciamento no Stata. Quando um banco é aberto, nenhuma mudança é realizada até que você salve o banco. O fato de usar uma cópia dos dados é importante porque:

– Quando se utiliza o comando "use arquivo", os dados são copiados para a memória do computador, e o arquivo original é fechado.

– Você pode fazer o que quiser com os dados na memória, e a cópia permanente continuará a mesma em seu disco.

– A única forma de mudar uma cópia permanente dos dados é utilizando o comando "save".

– Além disso, se algum erro é reportado, nenhuma mudança é realizada no banco que se encontra na memória.

Recursos disponíveis online

– Stata:
http://www.stata.com/links

– Centro de População da Carolina (CPC) da Universidade da Carolina do Norte de Chapel Hill (UNC):
http://www.cpc.unc.edu/services/computer/presentations/statatutorial

– Serviços de Tecnologia Acadêmica (ATS) da Universidade da Califórnia de Los Angeles (UCLA):
http://www.ats.ucla.edu/stat/stata/sk

– Portal de Estatística Computacional da Universidade da Califórnia de Los Angeles (UCLA):
http://statcomp.ats.ucla.edu/mlm/default.htm

– Site com explicações e exemplificações de comandos diversos de inferência estatística:
http://www.ats.ucla.edu/stat/stata/whatstat/whatstat.htm

– "Generalized Linear Latent And Mixed Models (GLLAMM)":
http://www.gllamm.org

– Instalação do "Generalized Linear Latent And Mixed Models (GLLAMM)":
http://www.gllamm.org/install.html

Semelhanças nas janelas do SPSS e Stata

SPSS e Stata possuem janelas para resultados (*Results*), variáveis (*Variables*), visualização dos dados (*Data Browser*, *Data Editor*), edição dos comandos (*Do-file Editor*).

O Stata ainda tem uma janela para digitar comandos rapidamente (*Command*) e comandos que foram digitados desde que o programa foi aberto (*Review*).

Organizando os diretórios para o curso

Antes de tudo vamos organizar os diretórios da aula na unidade C:\, utilizando o Windows Explorer:

C:\cursoufpe	Diretório principal
C:\cursoufpe\dados	Diretório para banco de dados
C:\cursoufpe\doc	Diretório para documentos diversos
C:\cursoufpe\graf	Diretório para gráficos
C:\cursoufpe\log	Diretório para resultados dos comandos
C:\cursoufpe\prog	Diretório para programas ".do"

Dados, questionários e documentação técnica

Os bancos de dados, questionários e documentação técnica deste curso estão disponíveis no site do “Observatorio de Élites Parlamentarias en América Latina” (Élites):

http://americo.usal.es/oir/Elites/bases_de_datos.htm

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.

Inserindo comentários

Para inserir comentários no Stata, simplesmente utilize asterisco (*) antes do texto:

```
*Isso é um tido como um comentário no Stata
```

Ou utilize os símbolos "/*" e "*/", antes e depois do comentário:

```
/*Essa é uma outra forma de
inserir comentário no Stata*/
```

Comandos que todos devem saber

O Stata tem por volta de 800 comandos. No quadro abaixo, estão listados alguns comandos que todos devem saber do que se tratam:

Categoria	Comandos no Stata
Obtendo ajuda	**search**, **findit**, **help**
Atualizando o Stata pela internet	**update**, **net**, **ado**, **news**
Começando, salvando e terminando o trabalho	**clear, set mem, set more, log**, **notes, exit**
Importando dados para o Stata	**infix, input, infile, insheet**
Usando e salvando dados do disco	**use**, **save**, **append**, **merge**, **compress**
Reportando dados básicos	**describe**, **codebook**, **list**, **browse**, **edit, count**, **inspect**, **summarize**, **table**, **tabulate**
Manipulação de dados	**generate**, **replace**, **egen**, **rename**, **drop**, **keep**, **sort**, **encode**, **decode**, **order**, **by**, **reshape**
Formatando	**format**, **label**
Conveniência	**display**

Obtendo ajuda

O comando "**findit**" procura informações de uma determinada palavra-chave. O "findit" faz uma procura completa, incluindo as procuras de "**help**" que procura por comando existente no Stata; "**search**" que procura pela palavra-chave na internet; e "**net search**" que procura por pacotes para instalação no site www.stata.com.

Atualizando o Stata pela internet

Para fazer atualização de comandos e procedimentos utilize o comando "**update**".

Para mostrar uma breve lista de recentes notícias e informações sobre o Stata, provenientes do site www.stata.com, digite "**news**".

Instalar um pacote de comandos:

```
net install nomepacote, from(diretório_ou_url)]
```

Utilize o comando "**ado**" para listar pacotes instalados.

Para descrever pacotes instalados, digite:

```
ado describe
```

Começando, salvando e terminando o trabalho

Geralmente o comando "clear" inicia um programa ".do" para limpar a memória do Stata:

```
clear
```

Estabelecendo a quantidade de memória alocada para o Stata:

```
set mem 100m
```

Para que essa quantidade de memória seja permanente toda vez que abrir o Stata:

```
set mem 100m, perm
```

Se grandes tabelas ou regressões forem ser geradas pelos seus comandos, é bom digitar o comando abaixo para que o programa não paralise a tela:

```
set more off
```

Abrindo um arquivo ".log" para salvar o trabalho. O ideal é escrever esse comando no começo do arquivo ".do":

```
log using "C:\cursoufpe\log\aula1.log", text replace
```

Salvando os comandos e tabelas geradas. Escreva esse comando no final do arquivo ".do":

```
log close
```

No final do trabalho, rode todo o programa novamente para salvar o "log" completo, usando os comandos “log using” e “log close”.

Para salvar somente os comandos, fazer um arquivo ".do" no "Do-file Editor". Se a janela "Review" tiver sido usada, clique com o botão direito do mouse para copiar o conteúdo e colar em um arquivo ".do".

Colocando avisos no banco de dados:

```
notes: criar rótulos em português para variáveis p501-p511
```

e

```
notes p201: verificar se variável foi codificada corretamente
```

Listar todos avisos criados no banco de dados:

```
notes
```

Encerrar o Stata:

```
exit
```

Se houver um banco de dados aberto no Stata, o ideal é digitar o seguinte comando para encerrar o programa sem salvar os dados:

```
exit, clear
```

Set matsize, set maxvar

Aprendemos a usar o "set mem" para informar o quanto de memória RAM deve ser disponibilizada pelo computador para que o Stata possa trabalhar:

```
set mem 100m, perm
```

Há ainda o comando "set matsize" que indica ao Stata o número máximo de variáveis que podem ser incluídas nos comandos do Stata. Por exemplo, esse número indica a quantidade máxima de variáveis que podem ser usadas em uma regressão.

O tamanho padrão no Stata/MP e Stata/SE é de 400 variáveis, podendo ser mudado para baixo ou para cima, com limite máximo de 11.000 variáveis. No Stata/IC, o valor inicial é de 200, com limite máximo de 800.

Por exemplo, podemos mudar o número máximo de variáveis nos comandos de estimação para 500:

```
set matsize 500
```

ou

```
set matsize 500, permanently
```

Além disso, o Stata/MP e Stata/SE permitem mudar o número máximo de variáveis no banco de dados com o comando "set maxvar". Isso não é permitido no Stata/IC.

```
set maxvar 5000
```

Importando dados para o Stata

Importando dados de um arquivo texto que possui formato fixo para as colunas. Exemplos da "Demographic Health Survey" e do Censo:

```
infix v005 038-045 v012 062-063 v013 064 using "C:\DHS96\brir31fl.dat"
```

ou

```
infix v0102 001-002 v0103 012-018 v0401 069 using "C:\Censo\2000\pes52.txt"
```

Importando dados manualmente para o Stata:

```
input nestu cuesti pais legis partido entrev
51 1 51 307 4 1
51 2 51 307 4 2
51 3 51 307 4 2
51 4 51 307 4 2
51 5 51 307 4 1
51 6 51 307 4 2
51 7 51 307 4 1
51 8 51 307 4 1
51 9 51 307 4 1
51 10 51 307 4 1
end
```

Usando e salvando dados do disco

Com o comando "use", você abre um banco de dados no Stata, mas não muda o diretório:

```
use c:\cursoufpe\dados\Argentina51.dta
```

Você pode primeiramente mudar para o diretório c:\cursoufpe\dados:

```
cd c:\cursoufpe\dados
```

Depois, simplesmente digite:

```
use Argentina51.dta
```

Já que os bancos de dados em Stata usam a extensão ".dta", você pode abrir o banco sem digitar a extensão:

```
use Argentina51
```

Se algum outro banco já estiver aberto, é preciso utilizar a opção "clear" para limpar a memória do Stata:

```
use c:\cursoufpe\dados\Argentina51.dta, clear
```

Como vimos, o comando para abrir um banco é "use". Se não houver nenhum banco aberto, utiliza-se:

```
use arquivo.dta
```

Se o desejo for descartar tudo que estiver na memória, utiliza-se:

```
use arquivo.dta, clear
```

Para salvar um banco pela primeira vez, utiliza-se:

```
save arquivo.dta
```

Se o arquivo já existir, e você quiser gravar o banco por cima do anterior:

```
save arquivo.dta, replace
```

Ou seja, um banco é salvo somente com o comando "save", tornando difícil perder os dados originais. Mesmo se o comando "save" não for usado intencionalmente, o Stata recusará gravar o banco por cima do original, se a opção "replace" não for colocada.

Utilize o comando “saveold” para salvar na versão anterior do Stata (versões 8 e 9) para que não haja problemas quando for usar o Stat Transfer ou o Stata antigo:

```
saveold arquivo.dta
```

Reportando dados básicos

Antes de tudo, é importante saber alguns sinais no Stata:

== igual
!= diferente
> maior
>= maior/igual
< menor
<= menor/igual
& E
| OU

Para mostrar o sumário do banco de dados, com nome, tipo e rótulo das variáveis:

```
describe
```

e

```
describe p501-p511
```

Para mostrar o sumário mais detalhado das variáveis do banco:

```
codebook
```

Outra forma de mostrar informações sobre as variáveis do banco, com ilustração de quantidade de números negativos, positivos e "missings", além de um pequeno gráfico de ramos e folhas (com distribuição da variável entre os seus valores):

```
inspect
```

Para contar quantos legisladores pensam que os riscos para consolidação da democracia são muito altos em decorrência da crise econômica (p502=4) e da dívida externa (p506=4):

```
count if p502==4 & p506==4
```

Para mostrar o banco na tela de resultados do Stata, utilize o comando "list". Para mostrar as variáveis que indicam a opinião dos legisladores do partido UCR (partido=2) sobre as possíveis vantagens de um regime democrático (p201 e p202 originárias da pergunta P2 na página 1 do questionário), digite o comando:

```
list p201 p202 if partido==2
```

```
     |     p201       p202 |
     |---------------------|
 21. | la posib   la posib |
 22. | la prote   la resol |
 23. | la prote   la posib |
 24. | la prote   la posib |
 25. | la prote   el respe |
     |---------------------|
 26. | la prote   la resol |
 27. | la prote   la posib |
 28. | la prote   el respe |
 29. | la prote   la mayor |
 30. | la prote   la posib |
     |---------------------|
 31. | la prote   la mayor |
 32. | el creci   la posib |
 33. | la prote   la posib |
 34. | la prote   la posib |
 35. | la posib   la prote |
     |---------------------|
 36. | la prote   la posib |
 37. | la mayor   el creci |
 97. | la resol   la mayor |
 98. | el respe   el creci |
 99. | el respe   la posib |
     |---------------------|
100. | la posib   la prote |
101. | la posib   la resol |
102. | la prote   la posib |
103. | la prote   el respe |
     +---------------------+
```

Se o rótulo da pergunta dificultar a visualização, utilize a opção "nolabel":

```
list p201 p202 if partido==2, nolabel
```

```
     | p201   p202 |
     |-------------|
 21. |    9      3 |
 22. |    2      8 |
 23. |    2      9 |
 24. |    2      6 |
 25. |    2      5 |
     |-------------|
 26. |    2      8 |
 27. |    2      3 |
 28. |    2      5 |
 29. |    2      4 |
 30. |    2      9 |
     |-------------|
 31. |    2      4 |
 32. |    1      9 |
 33. |    2      9 |
 34. |    2      9 |
 35. |    9      2 |
     |-------------|
 36. |    2      9 |
 37. |    4      1 |
 97. |    8      4 |
 98. |    5      1 |
 99. |    5      6 |
     |-------------|
100. |    3      2 |
101. |    6      8 |
102. |    2      6 |
103. |    2      5 |
     +-------------+
```

Para mostrar o banco em uma tela separada, utilize o comando "browse". Como no exemplo anterior:

```
browse p201 p202 if partido==2
browse p201 p202 if partido==2, nolabel
```

Para visualizar todo o banco, simplesmente digite:

```
browse
```

Para editar um banco, utilizasse o comando "edit" da mesma forma que o "list" e "browse". O comando "edit" pode ser acessado com o ícone "Data Editor" da barra de ferramentas.

Sem a utilização da opção "nolabel", as variáveis que aparecem em preto não possuem rótulos, aquelas que aparecem em azul possuem o rótulo visualizado, e as que apresentam a cor vermelha são variáveis nominais (*string* ou *character*).

É possível obter estatísticas básicas de variáveis com o comando "summarize" que é o mesmo que "sum". Podemos analisar as respostas quanto aos temas que podem representar uma ameaça ao risco de consolidação da democracia (p501 a p511):

```
summarize p501-p511
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        p501 |       104    1.596154    .6464125          1          4
        p502 |       104    3.144231    .8409226          1          4
        p503 |       104    2.048077    .9791993          1          4
        p504 |       103    3.048544    .8561043          1          4
        p505 |       104    2.509615    .9552633          1          4
-------------+--------------------------------------------------------
        p506 |       103           3    .9801961          1          4
        p507 |       104    3.086538    .7896242          1          4
        p508 |       103    2.203883    .7965517          1          4
        p509 |       103     3.38835    .8311699          1          4
        p510 |       103     2.84466    .9472073          1          4
-------------+--------------------------------------------------------
        p511 |       103           2    .8631906          1          4
```

Note acima que essas variáveis variam de 1 (Nada) a 4 (Muito). É bom estar ciente que poderia haver casos iguais a 8 (Não sabe=NS) e 9 (Não respondeu=NC), o que enviesaria a análise.

Uma análise de percentil poderia ser feita com a opção "detail" que é o mesmo que "d":

```
summarize p501-p511, detail
```

Variáveis nominais são automaticamente retiradas do sumário pelo Stata.

Para ordenar um banco de dados por uma variável ou conjunto de variáveis, utilize a opção "sort":

```
sort partido
```

Para realizar uma tabela de uma variável, utilize o comando "tabulate", que é o mesmo que "tab". As opções de "nolabel", "if" e "missing" também podem ser utilizadas:

```
tab p501 if p501!=1, nolabel missing
```

Para realizar um cruzamento entre partido político e a variável p501:

```
tab partido p501
```

Para gerar tabelas simples para cada uma das variáveis listadas:

```
tab1 partido p501-p511
```

Para gerar tabelas com cruzamentos entre duas variáveis para todas combinações possíveis:

```
tab2 partido p501 p502
```

Para obter a média, desvio padrão e freqüência da variável p501 em cada partido:

```
tab partido, summarize(p501)
```

Manipulação de dados

Transformar variável numérica em variável nominal (*string*), utilize o comando "decode":

```
decode partido, generate(nomepart)
```

Transformar variável nominal em variável numérica, utilize o comando "encode":

```
encode nomepart, generate(numpart)
```

Verificando o que foi feito:

```
browse partido nomepart numpart
```

O comando "tostring" não exige a criação de uma nova variável para transformar uma variável numérica para nominal. Isso só funciona se a variável não tiver rótulo (*label*):

```
tostring nestu, replace
```

O comando "destring" pode converter todas variáveis nominais para numéricas em um único comando:

```
destring, replace
```

Alocar determinadas variáveis para o começo do banco:

```
order p501-p511
```

Mover uma variável para uma posição anterior à segunda variável indicada no comando:

```
move entrev pais
```

Colocar em ordem alfabética as variáveis listadas e movê-las para o começo do banco:

```
aorder legisbis partido departam
```

Para excluir uma variável ou conjunto de variáveis do banco, utilize a opção "drop". Foi escolhido o número do estudo (nestu), pois ele não varia no banco de dados:

```
drop nestu
```

Note que só excluímos a variável da cópia do banco que está na memória RAM, e não a que está originalmente no disco.

Para remover observações do banco, utilize o comando "drop" com indicação da primeira e última observações a serem retiradas. Por exemplo, vamos remover as observações de 5 a 10:

```
drop in 5/10
```

Essa remoção pode também ser feita de outras formas. Por exemplo, vamos remover aqueles que consideram que as relações entre as forças armadas e o governo não apresentam risco para a consolidação da democracia (p501=1):

```
drop if p501==1
```

ou

```
drop if p501!=2 & p501!=3 & p501!=4 & p501!=.
```

ou

```
drop if p501<2
```

ou

```
drop if p501<2 & p501>=1
```

No Stata, o "missing" é o maior número (ao contrário do SAS), por isso o comando "drop if p501<2" não exclui os valores de p501 iguais a "missing".

É possível também escolher por manter determinadas variáveis no banco com o comando "keep":

```
keep if p501!=1
```

ou

```
keep if p501==2 | p501==3 | p501==4 | p501==.
```

Para gerar uma variável, utilize o comando "generate", que é o mesmo que "gen" ou "g":

```
gen p500=p501+p502+p503+p504+p505+p506+p507+p508+p509+p510+p511
```

ou

```
egen p500=rowtotal(p501 p502 p503 p504 p505 p506 p507 p508 p509 p510 p511)
```

Note que a opção "rowtotal" considera os valores "missings" iguais a zero. Como o comando "gen" simplesmente retira os casos que têm "missings" em algumas variáveis, o resultado é diferente entre os dois comandos acima. Para gerar o mesmo resultado, poderíamos fazer os seguintes comandos:

```
drop if p501==. | p502==. | p503==. | p504==. | p505==. | p506==. | p507==. |
p508==. | p509==. | p510==. | p511==.
```

e

```
gen p500=p501+p502+p503+p504+p505+p506+p507+p508+p509+p510+p511
tab p500
```

ou

```
egen p500=rowtotal(p501 p502 p503 p504 p505 p506 p507 p508 p509 p510 p511)
tab p500
```

Construindo uma variável que agrega informações de partido e departamento do legislador:

```
sort partido departam
egen pardep=group(partido departam)
browse partido departam pardep
```

Calculando o número de partidos por departamento:

```
sort departam
by departam: egen npartido=max(partido)
browse partido departam npartido
```

Quantidade de legisladores em cada partido:

```
tab partido
```

partido político	Freq.	Percent	Cum.
pdc	18	20.45	20.45
rn	16	18.18	38.64
udi	25	28.41	67.05
ppd	15	17.05	84.09
ps	9	10.23	94.32
prsd	5	5.68	100.00
Total	88	100.00	

A informação da tabela acima (número de deputados por partido) pode ser inserida no banco (como uma variável), seguindo os seguintes passos:

```
gen deputado=1
sort partido
by partido: egen ndeputado=count(deputado)
browse partido deputado ndeputado
tab ndeputado
```

ndeputado	Freq.	Percent	Cum.
5	5	5.68	5.68
9	9	10.23	15.91
15	15	17.05	32.95
16	16	18.18	51.14
18	18	20.45	71.59
25	25	28.41	100.00
Total	88	100.00	

Para renomear uma variável, digite "rename", o nome atual da variável e o nome que deseja. Vamos renomear a variável que informa o número do questionário:

```
rename cuesti quest
```

Para criar uma nova categorização da variável que informa sobre o grau de confiança que o legislador tem sobre o poder judiciário (p701) na vida pública argentina, podemos agrupar os valores de "muito" (p701=4) e "bastante" (p701=3) em uma categoria, e os valores de "pouco" (p701=2) e "nenhuma" (p701=1) em outro grupo.

Primeiramente, vamos saber se essa variável tem valores de "não sabe", "não respondeu" ou "missing":

```
tab p701, missing
```

ou

```
tab p701, missing nolabel
```

Posteriormente, criamos uma variável em que todos valores são missing:

```
gen p701g=.
```

Depois, fazemos as substituições em "p701g", conforme os valores de "p701":

```
replace p701g=0 if p701==1 | p701==2
replace p701g=1 if p701==3 | p701==4
```

Verificar se a nova variável foi criada corretamente:

```
tab p701 p701g, missing
```

Formatando

O comando "format" estabelece o formato de determinadas variáveis no banco. Use "help format" para ver as diferentes opções.

Mudando o ponto da variável "peso" para vírgula:

```
format peso %9,6fc
```

Colocando ou mudando o rótulo de uma variável:

```
label variable cuesti "Número do questionário"
```

e

```
label variable p701g "Poder judiciário (agrupado)"
```

Criando rótulos de categorias de variáveis:

```
label define escala 1 "Nada" 2 "Pouco" 3 "Bastante" 4 "Muito"
```

e

```
label define escalag 0 "Nada/Pouco" 1 "Bastante/Muito"
```

Colocando os rótulos das categorias para determinada variável:

```
label values p501-p511 escala
```

e

```
label values p701g escalag
```

Conveniência

"Display" mostra valores nominais e numéricos de expressões escalares.

Número de observações no banco de dados:

```
display _N
```

Fazendo cálculos manuais:

```
display 439/23
```

Utilizando os comandos "summarize" e "display" em conjunto:

```
summarize p501
display as text "Média de p501 = " as result r(mean)
```

Quebra de linha

No caso de comandos muito longos (tais como os utilizados para gerar gráficos e regressões), é bom inserir quebras de linha que indicam ao Stata que o comando não foi finalizado na linha atual, e que continua na linha abaixo. Isso ajuda na organização da programação. Podemos usar três barras (///) no final da linha para fazer essa indicação ao Stata:

```
drop if p501==. | p502==. | p503==. | p504==. | ///
        p505==. | p506==. | p507==. | p508==. | ///
        p509==. | p510==. | p511==.
```