

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
FACULDADE DE FILOSOFIA E CIÊNCIAS HUMANAS  
DEPARTAMENTO DE CIÊNCIA POLÍTICA**

**Curso de Introdução ao Stata – Aula 1**

**Professor**

Ernesto Friedrich de Lima Amaral

Email: [eflamaral@gmail.com](mailto:eflamaral@gmail.com)

Site do curso: [www.ernestoamaral.com/stata2012.html](http://www.ernestoamaral.com/stata2012.html)

**Antes do Stata**

Uma forma fácil de converter bancos de dados de um programa para outro é com o Stat Transfer. É possível converter bancos que estão em outros formatos para o Stata e vice-versa:

– Por isso, é possível utilizar o Excel para organizar bancos de dados secundários, já que este é um programa mais acessível. O Excel permite manipular bancos que tenham no máximo 1.048.576 observações (linhas) por 16.384 variáveis (colunas). Após a organização dos dados, o Stat Transfer pode ser utilizado para transferir os dados para o Stata, permitindo fazer análises estatísticas mais sofisticadas.

– No caso de dados primários (coletados com questionários próprios), é possível utilizar o Data Entry para realizar formulários de entrada dos dados. Após a digitalização dos questionários, é feita a transferência do banco para o Stata.

– Neste curso, será utilizado um banco de dados proveniente do SPSS.

**Dados, questionários e documentação técnica**

Os bancos de dados, questionários e documentação técnica deste curso estão disponíveis no site do Consórcio de Informações Sociais (CIS) do Núcleo de Apoio à Pesquisa sobre Democratização e Desenvolvimento da Universidade de São Paulo (NADD-USP) e da Associação Nacional de Pós-Graduação e Pesquisa em Ciências Sociais (ANPOCS):

<http://www.nadd.prp.usp.br/cis/index.aspx>

Vamos utilizar os dados da Pesquisa Nacional por Amostra de Domicílios (PNAD) de 2007 para Minas Gerais. Estes dados estão disponíveis em SPSS no CIS. Somente para a aula não ficar muito abstrata, vamos olhar um pouco o questionário, livro de códigos e demais arquivos. Durante a aula, é interessante olhar estes documentos para entender as variáveis.

## Entendendo o Stata

O Stata possibilita a criação de bancos de dados, gerenciamento destes dados, análise estatística e elaboração de gráficos.

É preciso utilizar comandos para realizar tais tarefas. Há por volta de 800 comandos no Stata. Durante o curso, iremos aprender alguns deles.

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. Estes menus podem ser usados quando não se sabe os comandos, e a *syntax* pode ser salva em seguida.

Principais janelas do Stata:

- 1) *Variables*: lista das variáveis no banco. Ao clicar no nome da variável, ela é enviada para a janela de comandos;
- 2) *Command*: digitação rápida de comandos;
- 3) *Results*: resultados obtidos;
- 4) *Review*: comandos que foram digitados desde que o programa foi aberto. É possível copiar os comandos aqui presentes para a janela de edição com o botão direito do mouse;
- 5) *Do-file Editor*: edição dos comandos;
- 6) *Data Browser* e *Data Editor*: visualização e edição dos dados.

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

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>

## Organizando os diretórios para o curso

Os diretórios da aula estão organizados da seguinte maneira na unidade C:\

C:\curso	Diretório principal
C:\curso\aulas	Diretório com roteiro das aulas
C:\curso\dados	Diretório para banco de dados
C:\curso\docs	Diretório para documentos diversos
C:\curso\grafs	Diretório para gráficos
C:\curso\modulos	Diretório com módulos extras
C:\curso\progs	Diretório para programas e resultados
C:\curso\tabelas	Diretório para tabelas

## 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 módulos para instalação no site [www.stata.com](http://www.stata.com).

É possível utilizar abreviações de comandos. Observe no "help" de determinado comando, a abreviação disponível.

## 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 memory 100m
```

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

```
set mem 100m, permanently
```

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:\curso\progs\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 para todas variáveis
```

e

```
notes v8005: recodificar a variável de idade em grupos quinquenais
```

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
```

## 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*/
```

## 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:

```
infix v0101 001-004 uf 005-006 v0201 022 using "C:\curso\dados\IBGE\DOM2007.txt"
```

Importando dados manualmente para o Stata:

```
input v0101 uf str8 (v0102) v0103 v0104 v0105 v0106
2007 11 11000015 001 09 . .
2007 11 11000015 002 01 3 2
2007 11 11000015 003 05 . .
2007 11 11000015 004 01 3 3
2007 11 11000015 005 01 4 2
2007 11 11000015 006 01 4 4
2007 11 11000015 007 01 4 3
2007 11 11000015 008 01 3 2
2007 11 11000015 009 01 4 2
2007 11 11000015 010 01 2 2
end
```

Como foi visto, é ainda possível utilizar o Stat Transfer para converter bancos que estão em outros formatos para o Stata e vice-versa.

## 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:\curso\dados\pes2007MG.dta
```

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

```
cd c:\curso\dados
```

Depois, simplesmente digite:

```
use pes2007MG.dta
```

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

```
use pes2007MG
```

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

```
use c:\curso\dados\pes2007MG.dta, clear
```

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

```
save c:\curso\dados\arquivo.dta
```

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

```
save c:\curso\dados\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 a versão antiga do Stata:

```
saveold c:\curso\dados\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 v0302 v0404 v0601-v0611
```

Para mostrar o sumário mais detalhado das variáveis do banco:

```
codebook v0302 v0404 v0601
```

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

```
inspect v0302 v0404 v0601
```

Para contar quantas mulheres (v0302=4) estão entre 15 e 49 anos de idade (v8005):

```
count if v0302==4 & (v8005>=15 & v8005<=49)
```

Para mostrar o total de observações (linhas) no banco de dados:

```
count
```

Para mostrar o banco na tela de resultados do Stata, utilize o comando "list". Para mostrar as variáveis que indicam a idade (v8005) e a cor/raça da pessoa (v0404) para aqueles que não sabem ler e escrever (v0601=3) e somente para os 100 primeiros casos do banco de dados:

```
list v8005 v0404 if v0601==3 in 1/100
```

	v8005	v0404
10.	3	2
11.	0	2
14.	5	8
16.	72	8
21.	3	2
32.	2	8
33.	83	2
35.	75	8
36.	50	8
45.	6	2
50.	86	8
52.	68	8
55.	94	8
61.	39	8
73.	43	2
77.	58	2

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

```
browse v8005 v0404 if v0601==3 in 1/100
```

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.

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 rendimentos dos moradores de 10 anos ou mais de idade:

- Se recebia rendimento mensal em dinheiro no trabalho principal (v9531);
- Valor do rendimento mensal em dinheiro no trabalho principal (v9532);
- Se recebia rendimento mensal em produtos ou mercadorias no trabalho principal (v9534);
- Valor do rendimento mensal em produtos ou mercadorias no trabalho principal (v9535).

```
summarize v9531 v9532 v9534 v9535
```

Variable	Obs	Mean	Std. Dev.	Min	Max
v9531	16380	1	0	1	1
v9532	16380	5.43e+09	7.35e+10	12	1.00e+12
v9534	95	3	0	3	3
v9535	95	922.7684	1695.816	20	9200

Note acima que a variável v9532 possui valores iguais a 999.999.999.999 (sem declaração). Além disso, todas variáveis acima possuem valores *missing*, já que na realidade temos um total de 36.320 observações no banco. Isso tudo envia a análise, tornando necessário recodificar as variáveis.

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

```
summarize v9532 v9535, detail
```

Variáveis nominais são automaticamente retiradas do sumário pelo Stata.

Outro comando para reportar estatísticas básicas é o "tabstat". Ele pode ser aplicado, por exemplo, a algumas variáveis de rendimento:

- Rendimento mensal no trabalho principal (v4718)
- Rendimento mensal de todos trabalhos (v4719)
- Rendimento mensal de todas fontes (v4720)

```
tabstat v4718-v4720, s(count min max mean sd p10 p25 median p75 p90)
```

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

```
*Ordenando o banco pela idade dos indivíduos
sort v8005
```

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

```
*Tabela da variável cor ou raça (v0404) somente para mulheres (v0302=4)
tab v0404 if v0302==4, missing
```

É possível selecionar as tabelas que aparecem na janela de resultados, copiar como texto, tabela, HTML ou figura e transferir para o Word ou Excel.

Para realizar um cruzamento entre anos de estudo (v4803) e cor ou raça (v0404):

```
tab v4803 v0404
```

Para facilitar a visualização, podemos retirar a frequência ("nofreq"), colocar a porcentagem na coluna ("col") e/ou colocar a porcentagem na linha ("row"):

```
tab v4803 v0404, nofreq col
```



Para gerar tabelas simples para cada uma das variáveis listadas:

```
tab1 v4803 v0404
```

Para gerar tabelas com cruzamentos entre duas variáveis para todas combinações possíveis entre grupos de anos de estudo (v4838), cor ou raça (v0404) e sexo (v0302):

```
tab2 v4838 v0404 v0302, nofreq col
```

Para obter a média, desvio padrão e frequência de anos de estudo (v4803) para cada cor/raça (v0404):

```
tab v0404, summarize(v4803)
```

## Manipulação de dados

O comando "tostring" transforma uma variável numérica em nominal. Isso só funciona se a variável não tiver rótulo (*label*):

```
tostring v0102, generate(strcontrole)
```

Verificando o que foi feito:

```
browse v0102 strcontrole
```

O comando "destring" converte variáveis nominais em numéricas:

```
destring strcontrole, generate(numcontrole)
```

Verificando o que foi feito:

```
browse v0102 strcontrole numcontrole
```

Com a opção "replace" os comandos "tostring" e "destring" salvam a nova variável sobre a variável já existente. É possível realizar essas transformações para todas variáveis do banco em um único comando:

```
destring, replace
```

É possível alocar as seguintes variáveis para o começo do banco:

- Sexo (v0302)
  - Cor ou raça (v0404)
  - Anos de estudo (v4803)
  - Grupos de anos de estudo (v4838)
  - Rendimento mensal no trabalho principal (v4718)
  - Rendimento mensal de todos trabalhos (v4719)
  - Rendimento mensal de todas fontes (v4720)
- ```
order v0302 v0404 v4803 v4838 v4718-v4720
```

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

```
move v0404 v0302
```

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

```
aorder v0302 v0404 v4803 v4838 v4718-v4720
```

Para colocar em ordem alfabética todas variáveis do banco:

```
aorder
```

Para excluir uma variável ou conjunto de variáveis do banco, utilize a opção "drop". Foi escolhido o ano de referência (v0101), pois ele não varia no banco de dados:

```
drop v0101
```

Note que só excluimos 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 as pessoas que não possuem declaração no rendimento mensal no trabalho principal (v4718=999.999.999.999):

```
drop if v4718==999999999999
```

ou

```
drop if v4718>=999999999999
```

No Stata, o "missing" é o maior número, ao contrário do SAS. Por isso, o comando "drop if v4718>=999999999999" exclui os valores de v4718 iguais a "missing".

Também é possível utilizar o comando "keep" para indicar quais variáveis se deseja manter no banco. É possível ainda escolher por manter determinadas categorias de variáveis no banco com o comando "keep":

```
keep if v4718<999999999999 | v4718==.
```

ou

```
keep if (v4718>=0 & v4718<999999999999) | v4718==.
```

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

- Número de filhos tidos, do sexo masculino, que moravam no domicílio (v1141)
- Número de filhos tidos, do sexo feminino, que moravam no domicílio (v1142)
- Número de filhos tidos, do sexo masculino, ainda vivos que moravam em outro local qualquer (v1151)
- Número de filhos tidos, do sexo feminino, que moravam no domicílio (v1152)
- Número de filhos tidos, do sexo masculino, que morreram (v1161)
- Número de filhos tidos, do sexo feminino, que morreram (v1162)

```
gen filhos1 = v1141 + v1142 + v1151 + v1152 + v1161 + v1162
```

ou

```
egen filhos2 = rowtotal(v1141 v1142 v1151 v1152 v1161 v1162)
```

A opção "rowtotal" considera os valores "missings" iguais a zero. O comando "gen" gera "missings" para aquelas observações que possuem algum valor em branco. Por isso, o resultado é diferente entre os dois comandos acima:

```
tab1 filhos1 filhos2, missing
```

Construindo uma variável que agrega informações de sexo (v0302) e cor/raça (v0404) da pessoa:

```
sort v0302 v0404
```

```
egen sexcor=group(v0302 v0404)
```

```
browse v0302 v0404 sexcor
```

É importante explorar o comando "egen" para entender diferentes formas de gerar variáveis.

É possível calcular a média de filhos tidos (filhos2) por cor/raça da mãe (v0404) e incluir no banco de dados como uma nova variável:

```
tab v0404, sum(filhos2)

sort v0404
by v0404: egen medfilho=mean(filhos2)
browse v0404 filhos2 medfilho
```

A distribuição de pessoas por cor/raça (v0404) pode ser inserida no banco, seguindo os seguintes passos:

```
tab v0404

gen n=1
sort v0404
by v0404: egen ncor=count(n)
browse v0404 ncor
```

O comando "tabulate" pode ser utilizado em conjunto com o comando "generate" para criar variáveis dicotômicas automaticamente, levando em consideração as categorias de uma determinada variável. No caso abaixo, serão criadas seis variáveis ("cor1", "cor2", "cor3", "cor4", "cor5" e "cor6"):

```
tab v0404, gen(cor)
```

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

```
rename v8005 idade
```

Para criar uma nova categorização da variável que informa sobre a cor/raça da pessoa (v0404), podemos agrupar as categorias "preta" (v0404=4) e "parda" (v0404=8), colocar a categoria "branca" (v0404=2) separada, além de deixar em branco as categorias "amarela" (v0404=6), "indígena" (v0404=0) e "sem declaração" (v0404=9).

Primeiramente, vamos saber se essa variável tem valores "sem declaração" ou "missing":

```
tab v0404, missing
```

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

```
gen branca=.
```

Depois, fazemos as substituições em "branca", conforme os valores de "v0404":

```
replace branca=0 if v0404==4 | v0404==8
replace branca=1 if v0404==2
```

Verificar se a nova variável foi criada corretamente:

```
tab v0404 branca, missing
```

A recodificação da idade (v8005) em grupos quinquenais de idade seria realizada assim:

```
gen gridade = .
replace gridade = 00 if v8005 >= 0 & v8005 <= 4
replace gridade = 05 if v8005 >= 5 & v8005 <= 9
replace gridade = 10 if v8005 >= 10 & v8005 <= 14
replace gridade = 15 if v8005 >= 15 & v8005 <= 19
replace gridade = 20 if v8005 >= 20 & v8005 <= 24
replace gridade = 25 if v8005 >= 25 & v8005 <= 29
replace gridade = 30 if v8005 >= 30 & v8005 <= 34
replace gridade = 35 if v8005 >= 35 & v8005 <= 39
replace gridade = 40 if v8005 >= 40 & v8005 <= 44
replace gridade = 45 if v8005 >= 45 & v8005 <= 49
replace gridade = 50 if v8005 >= 50 & v8005 <= 54
replace gridade = 55 if v8005 >= 55 & v8005 <= 59
replace gridade = 60 if v8005 >= 60 & v8005 <= 64
replace gridade = 65 if v8005 >= 65 & v8005 <= 69
replace gridade = 70 if v8005 >= 70 & v8005 <= 74
replace gridade = 75 if v8005 >= 75 & v8005 <= 79
replace gridade = 80 if v8005 >= 80 & v8005 <= 84
replace gridade = 85 if v8005 >= 85 & v8005 <= 120
```

Verificar o resultado:

```
tab v8005 gridade, missing
```

## Formatando

Colocando ou mudando o rótulo de uma variável:

```
label variable branca "Indicador de cor/raça branca"
```

Os rótulos das variáveis originais do banco já haviam sido inseridos (ver arquivos "do"). Todas variáveis também haviam sido transformadas em numéricas ("dstring, replace").

Criando rótulos de categorias de variáveis:

```
label define cor 2 "Branca" 4 "Preta" 6 "Amarela" 8 "Parda" 0 "Indígena" 9 "Sem
declaração"
```

e

```
label define branca 0 "Preta/Parda" 1 "Branca"
```

Colocando os rótulos das categorias para determinada variável:

```
label values v0404 cor
```

e

```
label values branca branca
```

Verificando o que foi feito:

```
tab v0404 branca, missing
```

## 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
```

## 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:

```
label define cor 2 "Branca" 4 "Preta" 6 "Amarela" ///  
8 "Parda" 0 "Indígena" 9 "Sem declaração"
```