

3 POR QUE USAR O DO FILE?

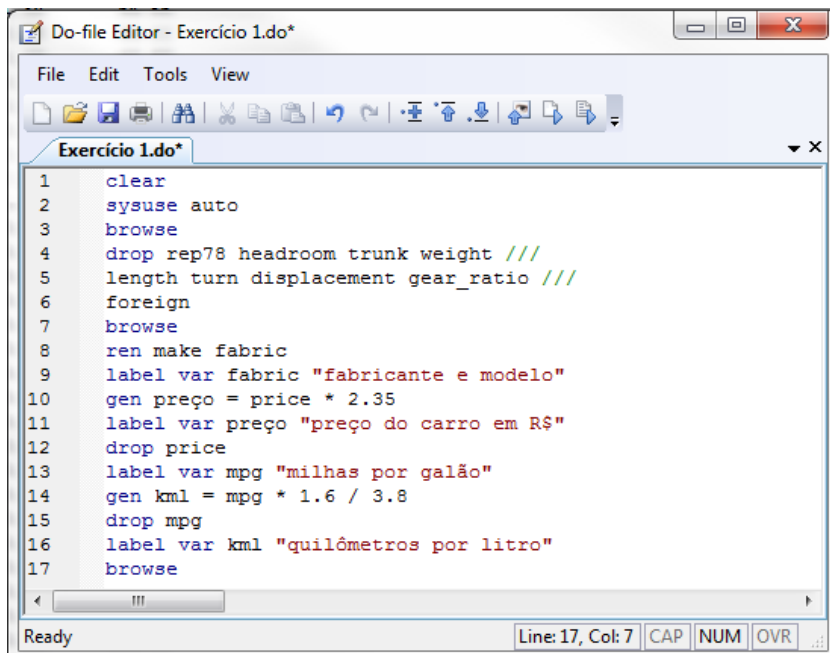
Imagine a seguinte situação: Você está escrevendo um artigo com uma colega, sua co-autora, e precisa pedir para ela executar os mesmos comandos que você executou no Stata. O que é mais simples: enviar um e-mail para ela com um arquivo de texto contendo o passo a passo; ligar para ela e dizer os comandos um a uma para ela digitar na linha “command”; ou enviar um arquivo organizado e com comentários para ela executar diretamente no Stata? A última opção parece a mais viável, não é mesmo?

Vamos pensar em mais um exemplo real. Você escreveu um artigo para uma disciplina do mestrado e, após deixá-lo de lado por um ano, deseja melhorá-lo. Será que você vai lembrar o que cada código que você escreveu faz depois de tanto tempo?

E é justamente por isso que usamos o do-file. Ele oferece diversas vantagens sobre qualquer outra forma de trabalhar, seja sozinho ou em conjunto, já que possibilita a inserção de comentários e comandos que podem ser executados de uma vez só, sem ter que digitar cada linha de comandos. Além disso, você pode comentar cada código, explicar por que está usando aquele código e para quê ele serve. A seguir, veremos um exemplo.

4 UM MODELO DE DO-FILE

A figura abaixo mostra um do-file (sem comentários). Você é capaz de entender o que cada código significa e o que ele faz? (Tente copiá-lo no seu Stata e veja o que acontece. Não esqueça de usar um do-file!)



```
1 clear
2 sysuse auto
3 browse
4 drop rep78 headroom trunk weight ///
5 length turn displacement gear_ratio ///
6 foreign
7 browse
8 ren make fabric
9 label var fabric "fabricante e modelo"
10 gen preço = price * 2.35
11 label var preço "preço do carro em R$"
12 drop price
13 label var mpg "milhas por galão"
14 gen kml = mpg * 1.6 / 3.8
15 drop mpg
16 label var kml "quilômetros por litro"
17 browse
```

Difícil, não é mesmo? Agora, vou explicar o que fiz em palavras, e depois mostrar o mesmo código, só que com comentários.

*Basicamente, usei um banco de dados que já vem no Stata, o **auto**. Esse banco de dados contém dados de diversas variáveis relativas a automóveis nos EUA no ano de 1978. Como este é um exercício simples, eu só quis manter as três primeiras variáveis e eliminar as demais, para não ficar muito complexo de entender. Também para simplificar, renomeei as variáveis em inglês, traduzindo-as para o português (ou abreviações em português). Fiz o mesmo para a descrição de cada variável, logo depois de renomeá-las/criá-las.*

Como o preço do carro estava em dólar, criei uma nova variável chamada **preço**. Assumi a taxa de câmbio $US\$1=R\$2,35$ e pedi ao Stata que atribuísse a essa variável o valor em dólares \times taxa de câmbio¹. Depois disso, eliminei o preço em dólares, que não seria mais útil.

Depois, percebi que a última variável, **mpg**, representava o **consumo de combustível** de cada carro na unidade **milhas por galão**. No entanto, essa unidade só é utilizada nos EUA e no Brasil utilizamos **quilômetros por litro**.

Sei que uma milha é aproximadamente igual a 1,6 quilômetros e que um galão tem aproximadamente 3,8 litros. Assim, criei uma nova variável chamada **kml** e pedi ao Stata que atribuísse a ela o valor do consumo de combustível (milha por galão) do carro correspondente, multiplicasse esse valor por 1.6 (observe que usei ponto no lugar de vírgula, que é a configuração básica do Stata), já que cada $1\text{ mi} \approx 1.6\text{ km}$ e dividisse por 3.8, já que $1\text{ galão} \approx 3.8\text{ l}$. Depois disso, eliminei a variável **mpg**, que não seria mais útil e adicionei uma descrição à variável **kml**. Ou seja, converti milhas/galão para km/litro, que é a medida que utilizamos no Brasil.

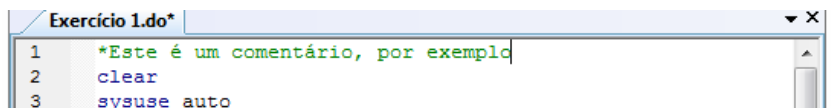
E agora? Ficou mais fácil entender? Talvez sim, mas ainda é um pouco difícil ter que ler a descrição e voltar para o código, não é mesmo? Por isso, o ideal é comentar seu código no próprio do-file. Para isso, é necessário aprender como fazer comentários no do-file.

5 COMENTÁRIOS E LINHAS EM BRANCO NO DO-FILE

Você pode incluir quantas linhas em branco quiser no seu do-file. Ou seja, mesmo se o código apresentado acima tivesse uma linha em branco entre cada linha de comando, o resultado seria o mesmo.

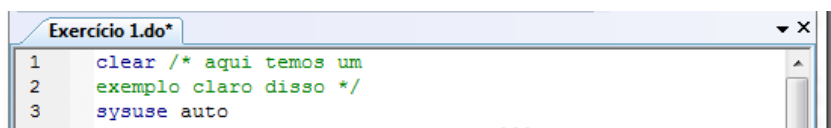
Já no caso de comentários, há quatro formas de inclui-los em um do-file. São as seguintes:

1. **Começar a linha com um asterisco (*):** O Stata vai ignorar o que está escrito nessas linhas, ou seja, não vai considera-las uma linha de comando, e todo o texto mudará de cor, passando a ficar verde;

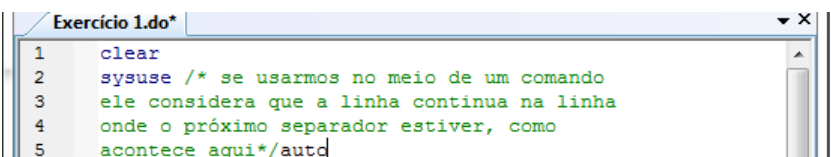


```
Exercício 1.do*
1  *Este é um comentário, por exemplo
2  clear
3  sysuse auto
```

2. **Colocar o comentário entre os delimitadores /* e */:** Assim, tudo que vier depois do primeiro delimitador (/*) e antes do último (*/), não importando se você usa eles em linhas diferentes, será ignorado pelo Stata (não será considerado um comando). Se você usar isso no meio de uma linha de comando, ele considera que a linha continuará na linha em que o último delimitador (*/) estiver;



```
Exercício 1.do*
1  clear /* aqui temos um
2  exemplo claro disso */
3  sysuse auto
```



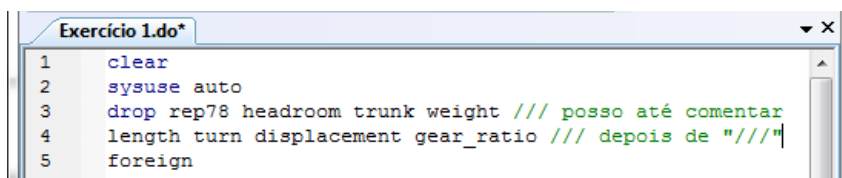
```
Exercício 1.do*
1  clear
2  sysuse /* se usarmos no meio de um comando
3  ele considera que a linha continua na linha
4  onde o próximo separador estiver, como
5  acontece aqui*/autd
```

¹ Do ponto de vista “mecânico” e da matemática, essa operação está correta. E economicamente? Essa nova variável faria sentido ou seria necessário fazer alguma operação ou especificação adicional? Por que?

3. **Colocar o comentário depois de duas barras, ou seja, //**: Tudo que vier depois de **//** no fim de uma linha será considerado um comentário (a menos que **//** seja parte de um endereço da web **http://...**);

```
9 label var preçodol "preço do carro em US$"
10 gen preço = preçodol * 2.35
11 label var preço "preço do carro em R$"
12 drop preçodol //usei "drop" para eliminar a variável
13 label var mpg "milhas por galão" //"label" = descrição
14 gen kml = mpg * 1.6 / 3.8 //"gen" = nova variável
15 drop mpg
```

4. **Colocar o comentário depois de três barras, ou seja, ///**: Tudo que vier depois de **///** no fim de uma linha será considerado um comentário. No entanto, quando se usa **///**, a próxima linha será considerada uma continuação da linha atual. Assim, essa é a melhor forma de “quebrar” linhas de comandos que sejam muito grandes.



```
1 clear
2 sysuse auto
3 drop rep78 headroom trunk weight /// posso até comentar
4 length turn displacement gear_ratio /// depois de '///'
5 foreign
```

Observe que as linhas 3, 4 e 5 são, na verdade, uma linha só de comando. Como ela é muito grande, é mais conveniente dividirmos ela em 3 partes usando **///**.

6 COMANDOS IMPORTANTES NO STATA

Temos abaixo uma lista de comandos (básicos) importantes no Stata. Ao lado de cada comando há um espaço para que você possa descrever cada um deles. Para isso, tente usar o comando e busque o que cada um faz e escreva nesses espaços, com suas próprias palavras. Você pode fazer isso de duas formas: Busque no Google o nome do comando com as palavras comando e Stata (ex: “comando help no stata” ou “o que faz o comando help no stata?”); ou no próprio Stata, digitando **help [comando]** (ex: “**help pwd**”) – nesse caso, uma janela se abrirá (em inglês) e haverá um texto explicando. Se você não conseguir entender, não se preocupe, revisaremos esses comandos durante as aulas presenciais. Abaixo estão alguns comandos²:

help	<u>l</u>abel
pwd	<u>g</u>enerate
clear	<u>s</u>ummarize
set more off³	log
cd	<u>d</u>escribe
<u>h</u>istogram	<u>t</u>abulate
drop	import
<u>r</u>ename	list
browse	use

² As partes sublinhadas dos comandos são as abreviações dos mesmo. Por exemplo, **hist** é o mesmo que **histogram**.

³ Busque o comando “**more**”, ou seja, digite “help more” em vez de “help set more off”

7 OPERAÇÕES MATEMÁTICAS NO STATA

Você deve ter percebido que no nosso modelo de do-file utilizado no item 4 foi gerada uma nova variável para preço em Real, que era o preço em dólar multiplicado pela taxa de câmbio (linha 10), e que para fazer isso utilizamos a equação abaixo

$$preço = price * 2.35$$

No Stata (e no R, no LaTeX e em vários outros softwares) utilizamos o asterisco para representar a multiplicação. Os operadores importantes no Stata são os seguintes

OPERAÇÃO	OPERADOR	EXEMPLO
Adição	+	$2 + 2 = 4$
Subtração	-	$12 - 3 = 9$
Multiplicação	*	$5 * 5 = 25$
Divisão	/	$48 / 4 = 12$
Elevar a uma potência	^	$x^5 = x^5$
Prefixo negativo	-	-5
Raiz quadrada	sqrt(.)	$sqrt(4) = \sqrt{4}$
Maior que	>	$8 > 5$
Maior ou igual	>=	$x >= 5$
Menor que	<	$3 < 5$
Menor ou igual	<=	$z <= 4$
Igual	==	$y == 12$
Não igual	!= ou ~=	$4 != 5$ ou $4 ~= 5$

Para testar se você entendeu, escreva dentro da caixa abaixo a fórmula equivalente a digitarmos a seguinte expressão no Stata

$$-(x+y^{(x-y)}) / (x*y)$$

Resposta:

Atenção! Se você usar o operador + entre dois *strings* (dois "textos"), o Stata vai unir esses textos. Por exemplo, a digitar no Stata expressão "**isto**"+"**aquilo**" resultará no *string* "**istoaquilo**".

Alguns exemplos:

COMANDO	SIGNIFICADO
. list if idade<=25	Listar todas as observações em que a variável idade é maior ou igual a 25
. list if dummy==0	Listar todas as observações em que a variável dummy é zero
. gen idade2 = idade^2	Gerar a variável idade2 , igual aos valores de idade elevados ao quadrado
. sum preço	Exibe estatísticas descritivas da variável preço

Na atividade a seguir, você aprenderá a adicionar comentários a um do-file.