

```
<<dd_version: 1>>
<<dd_include: header.txt >>
Dynamic Markdown documents in Stata 15: An initial look
=====
```

First I open the example data nlsw88 and run the describe the command.

The tags around the code `**<<dd_do>>**` tells Stata to execute this block of code and include its output in the document. `**<</dd_do>>**` indicates the end of a `<<do>>`.

The four tilde (~) symbols indicate that this block should appear as a block of code in the HTML document (i.e. they will be inset from the text).

Here I have run the commands twice. The first time I add the tag `**<<dd_ignore>>**` which tells Stata not to run this command but print all the code in the document. This just lets you see what the whole block of code looks like in the plain text file.

The second time I do not include the `**<<dd_ignore>>**` tag so you see the output and code as you would see it in Stata.

```
~~~~
<<dd_ignore>>
<<dd_do>>
sysuse nlsw88, clear
describe
<</dd_do>>
<</dd_ignore>>
```

```
<<dd_do>>
sysuse nlsw88, clear
describe
<</dd_do>>
~~~~
```

Now, I summarize the data.

```
~~~~
<<dd_ignore>>
<<dd_do>>
summ
<</dd_do>>
<</dd_ignore>>
```

```
<<dd_do>>
summ
<</dd_do>>
~~~~
```

We can use numbers from the Stata output in our text in the HTML document.
We

always want to avoid cutting and pasting numbers. We can use the `**dd_display**` tag to retrieve stored results from the `summ` command and use them in our text.

First we run the `summ` command for our variable of interest. We have run this above so we don't need this to appear in the HTML document. Therefore we can use the `**<<dd_do:quietly>>**` tag so Stata does the work but doesn't tell us about it.

```
~~~~~
<<dd_ignore>>
<<dd_do: quietly>>
summ age
<</dd_do>>
<</dd_ignore>>

<<dd_do: quietly>>
summ age
<</dd_do>>
~~~~~
```

Then we use the stored values for minimum and maximum age in the text.

```
~~~~~
<<dd_ignore>>
> The variable age has minimum value <<dd_display: %4.2f `r(min)'\>>
and has maximum value <<dd_display: %4.2f `r(max)'\>>.
<</dd_ignore>>

> The variable age has minimum value <<dd_display: %4.2f `r(min)'\>>
and has maximum value <<dd_display: %4.2f `r(max)'\>>.
~~~~~
```

Now lets see if this will work for something a little more complex. Here I have run a regression, the outcome variable is wage, I have run this quietly as we might not want to show all the output Stata gives in the standard regression output.

We store these estimates as "m1".

```
~~~~~
<<dd_ignore>>
<<dd_do: quietly>>
regress wage age married i.race, allbaselevels
est sto m1
<</dd_do>>
<</dd_ignore>>

<<dd_do: quietly>>
```

```
regress wage age married i.race, allbaselevels
est sto m1
<</dd_do>>
~~~~
```

No lets present the output of the regression in the HTML document.

```
~~~~
<<dd_ignore>>
<<dd_do>>
esttab m1, b se wide
<</dd_do>>
<</dd_ignore>>
```

```
<<dd_do>>
esttab m1, b se wide
<</dd_do>>
~~~~
```

We can also use the stored results to say a little bit more about this regression model.

```
~~~~
<<dd_ignore>>
> The sample size was <<dd_display: %4.2f `e(N)'\>>
and the model has an adjusted R2 value of <<dd_display: %4.2f `e(r2_a)'\>>.
<</dd_ignore>>
```

```
> The sample size was <<dd_display: %4.2f `e(N)'\>>
and the model has an adjusted R2 value of <<dd_display: %4.2f `e(r2_a)'\>>.
~~~~
```

Now lets do a graph.

Here is a scatterplot of age and wage.

```
~~~~
<<dd_ignore>>
<<dd_do:nooutput>>
scatter age wage, mcolor(blue%50)
<</dd_do>>
<</dd_ignore>>
```

```
<<dd_do:nooutput>>
scatter age wage, mcolor(blue%50)
<</dd_do>>
```

```
<<dd_ignore>>
<<dd_graph: sav("graph1.svg") alt("scatter age wage") replace
height(400)>>
<</dd_ignore>>
~~~~
```

Here is the graph:

```
<<dd_graph: sav("graph1.svg") alt("scatter age wage ") replace  
height(400)>>
```

```
end dyndoc blogexample.do
```