

R, Python, and Stata in parallel

Here, I run Stata parallel to R and Python using Statamarkdown.

```
library(Statamarkdown)
knitr::opts_chunk$set(collectcode = TRUE)
```

R

```
print("Hello, R world!")
```

```
[1] "Hello, R world!"
```

```
test <- "Am I still defined in R?"
```

Stata

```
dis "Hello, Stata world!"
global testg "Am I still defined in Stata?"
local testl "Am I still defined in Stata?"
```

```
Hello, Stata world!
```

Python

```
print("Hello, Python world!")
```

```
Hello, Python world!
```

```
test = "Am I still defined in Python?"
```

Test

R

```
test
```

```
[1] "Am I still defined in R?"
```

Stata

```
dis "Global: $testg Yes"  
dis "Local: `testl' No"
```

```
Global: Am I still defined in Stata? Yes
```

```
Local: No
```

Python

```
test
```

```
'Am I still defined in Python?'
```

As previously discussed, Stata local macros are not defined across codeblocks, even with the `collectcode=TRUE` option. But globals and other objects stored in memory (data, matrices, etc.) are not.

Limitations

A major limitation of tabset multilingual documents is the way in which they render as PDFs. One of the nice features of Quarto is the ability to seamlessly render html and pdf documents. In general, the pdf documents publish codeblocks very well, which html-like formatting. The only limitation I have seen is when longer lines of code (which would have a scrowler in html) are cut-off in the PDF.

Click on the linked PDF document (top right) to see how this page is rendered. You will see that code and corresponding output from each tabbed language are displayed in sequential order. As a result, adding a lot of multilingual code to a single document will invariably create a very long pdf.

If you want to avoid this, you will need to either avoid PDFs or render PDFs of each language in separate files and find a way to attach them via a link. On the next page I provide a temporary solution for this.