Running multiple instances of Stata using the `sessions` command.

Paul C. Lambert[1,2]

[1]Cancer Registry of Norway, Norwegian Institute of Public Health, Oslo, Norway
[2]Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden

Northern European Stata Conference, Oslo
2026-09-24







Background

- A typical thing we do when analysing cancer registry data is to loop over many cancer sites, with a separate analysis for each site.
- This is fine if the analysis is quick, but what if each analysis is computationally intensive?
- Each analysis can be considered completely independent - there is no need to communicate between the different analyses.
- This is know as an embarrassingly parallel workload.
- Another example is different scenarios in a simulation study.

Calling Stata from Stata

- An alternative is to start different Stata sessions using `winexec`.

```
. levelsof cancersites, local(sitelevels)
. foreach site in `sitelevels´ {
    winexec "C:/Program Files/Stata19/StataMP-64.exe" /b do analysis.do `site´
. }
```

- Potential problems
  - May run low on memory
  - May overload CPU resources,
  - No monitoring of progress.

Why another command

- There are other commands (`qsub`,`multishell`,`parallel`,`dobatch`), but none worked how we wanted them to.
- We want to monitor progress.
- We want to wait until sufficient system resources (CPU load / memory).
- We want to wait until completion of certain do files before progressing.
- We do not want editing/scanning of our do files.
- We want to detect errors.
- Should be easy to use.

The `sessions` command

- Recommend to use `sessions monitor` to start and monitor a set of Stata sessions.
- Also can access utility commands.

`sessions init`	Initialize sessions
`sessions start`	Start a new Stata session
`sessions list`	List completed, acive and queued Stata sessions
`sessions kill`	Kill a Stata session
`sessions resources`	list current resources (CPU load and free memory)
`sessions clear`	clear sessions

`sessions` uses java to communicate with operating system

- Java used to
  - Start Stata
  - Extract Process ID (PID) number
  - Monitor system resources
  - Monitor when Stata sessions have completed.
  - Kill sessions
  - Display current status of sessions.
- In addition
  - Monitors if any Stata sessions exits with error code.

`sessions` uses java to communicate with operating system

- Java used to
  - Start Stata
  - Extract Process ID (PID) number
  - Monitor system resources
  - Monitor when Stata sessions have completed.
  - Kill sessions
  - Display current status of sessions.
- In addition
  - Monitors if any Stata sessions exits with error code.
- Stata comes with the OpenJDK (Open Java Development Kit), so no installation needed.

Simple example

- I will analyses some old breast cancer survival data from England and Wales.
- There are 136,917 individuals in the dataset, in 9 regions.
- We will peform separate analyses within each region.
- The project will:
  - Do some data cleaning and preparation.
  - Compare non-parametric and parametric survival estimates for each region.
  - Combine the results of each analysis.
  - Produce table/graph of results

Simple example

- I will analyses some old breast cancer survival data from England and Wales.
- There are 136,917 individuals in the dataset, in 9 regions.
- We will peform separate analyses within each region.
- The project will:
  - Do some data cleaning and preparation.
  - Compare non-parametric and parametric survival estimates for each region.
  - Combine the results of each analysis.
  - Produce table/graph of results
- We do not want to start the analysis sessions until the data cleaning and preparation has completed.
- We do not want to start the table and plot sessions until the analysis sessions have completed

Main Stata Session: create list of sessions/dofile

SessionName	dofile	region	component
Setup	Data_setup.do	.	data setup
Setup	More_Data_setup.do	.	data setup
wait		.	
N & York	net_survival.do	1	analysis
Trent	net_survival.do	2	analysis
Ang & Ox	net_survival.do	3	analysis
N Thames	net_survival.do	4	analysis
S Thames	net_survival.do	5	analysis
S West	net_survival.do	6	analysis
W Mids	net_survival.do	7	analysis
N West	net_survival.do	8	analysis
Wales	net_survival.do	9	analysis
wait		.	
Combine	combine_results.do	.	postestimation
wait		.	
Results	graphs_comparison.do	.	postestimation
Results	tables_simple.do	.	postestimation

Comment on data

- Do files with the same session name will be sent to the same Stata session.
- using `_wait_` will pause starting a new session until all the sessions above are complete.
- Arguments can be passed to do files. Here `region` will be passed to `net_survival.do`
- Having a `component` variable is useful for running a subset of the Stata sessions.

Running the sessions command

```
. sessions init
--sessions is initialized--
Stata executable: C:/Program Files/Stata18/StataSE-64.exe
Available resources:
  Logical Processors: 14 | CPU Load:12% | Free memory: 15%
  You have 0 active sessions (excluding this one).
```

Running the sessions command

```
. sessions init
--sessions is initialized--
Stata executable: C:/Program Files/Stata18/StataSE-64.exe
Available resources:
  Logical Processors: 14 | CPU Load:12% | Free memory: 15%
  You have 0 active sessions (excluding this one).
```

```
. sessions monitor,   names(SessionName)          /// Session name
                      dofiles(dofile)             /// dofiles to run
                      arguments(region)           /// arguments
                      commondo("do/globals.do")   /// Common do files
                      logdelete                   /// delete existing logs
                      freememory(10)              /// free memory % needed
                      cpuload(70)                 /// start if <70% CPU load
                      maxsessions(4)              /// maxium number of sessions
                      update(3)                   //  update every 3 seconds
```

Initial list of Sessions

```
Update at 21 Sep 2026: 15:14:37
  Logical Processors: 14 | CPU Load: 7% | Free memory: 15%
  You have 1 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Setup	7440	Active	21Sep2026: 15:14:29	
2	N & York		Waiting		
3	Trent		Queued		
4	Ang & Ox		Queued		
5	N Thames		Queued		
6	S Thames		Queued		
7	S West		Queued		
8	W Mids		Queued		
9	N West		Queued		
10	Wales		Queued		
11	Combine		Queued(W)		
12	Results		Queued(W)		

Completed, Active and Queued

```
Update at 21 Sep 2026: 15:14:46
  Logical Processors: 14 | CPU Load:38% | Free memory: 12%
  You have 4 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Setup	7440	Completed	21Sep2026: 15:14:29	21Sep2026: 15:14:39
2	N & York	23848	Active	21Sep2026: 15:14:39	
3	Trent	30920	Active	21Sep2026: 15:14:40	
4	Ang & Ox	38540	Active	21Sep2026: 15:14:41	
5	N Thames	33344	Active	21Sep2026: 15:14:42	
6	S Thames		Queued		
7	S West		Queued		
8	W Mids		Queued		
9	N West		Queued		
10	Wales		Queued		
11	Combine		Queued(W)		
12	Results		Queued(W)		

Waiting for preceding sessions to finish

```
Update at 21 Sep 2026: 15:15:36
  Logical Processors: 14 | CPU Load:37% | Free memory: 11%
  You have 4 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Setup	7440	Completed	21Sep2026: 15:14:29	21Sep2026: 15:14:39
2	N & York	23848	Completed	21Sep2026: 15:14:39	21Sep2026: 15:15:02
3	Trent	30920	Completed	21Sep2026: 15:14:40	21Sep2026: 15:15:00
4	Ang & Ox	38540	Completed	21Sep2026: 15:14:41	21Sep2026: 15:15:04
5	N Thames	33344	Completed	21Sep2026: 15:14:42	21Sep2026: 15:15:15
6	S Thames	25416	Active	21Sep2026: 15:15:00	
7	S West	35096	Active	21Sep2026: 15:15:03	
8	W Mids	19520	Completed	21Sep2026: 15:15:04	21Sep2026: 15:15:36
9	N West	33856	Active	21Sep2026: 15:15:21	
10	Wales	16276	Active	21Sep2026: 15:15:36	
11	Combine		Queued(W)		
12	Results		Queued(W)		

All sessions completed

```
All sessions completed at 21 Sep 2026: 15:15:52


Update at 21 Sep 2026: 15:15:52
  Logical Processors: 14 | CPU Load:22% | Free memory: 14%
  You have 0 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Setup	7440	Completed	21Sep2026: 15:14:29	21Sep2026: 15:14:39
2	N & York	23848	Completed	21Sep2026: 15:14:39	21Sep2026: 15:15:02
3	Trent	30920	Completed	21Sep2026: 15:14:40	21Sep2026: 15:15:00
4	Ang & Ox	38540	Completed	21Sep2026: 15:14:41	21Sep2026: 15:15:04
5	N Thames	33344	Completed	21Sep2026: 15:14:42	21Sep2026: 15:15:15
6	S Thames	25416	Completed	21Sep2026: 15:15:00	21Sep2026: 15:15:39
7	S West	35096	Completed	21Sep2026: 15:15:03	21Sep2026: 15:15:40
8	W Mids	19520	Completed	21Sep2026: 15:15:04	21Sep2026: 15:15:36
9	N West	33856	Completed	21Sep2026: 15:15:21	21Sep2026: 15:15:48
10	Wales	16276	Completed	21Sep2026: 15:15:36	21Sep2026: 15:15:47
11	Combine	24932	Completed	21Sep2026: 15:15:48	21Sep2026: 15:15:49
12	Results	37272	Completed	21Sep2026: 15:15:49	21Sep2026: 15:15:52

Restricting to a subset of do files

- We may want to only run a subset of do files e.g. update Tables/Graphs.

```
. sessions monitor if component=="postestimation", /// posestimation results only
                      names(SessionName)           /// Session name
                      dofiles(dofile)              /// dofiles to run
                      arguments(region)            /// arguments
                      commondo("do/globals.do")    /// Common do files
                      logdelete                    /// delete existing logs
                      freememory(10)               /// free memory % needed
                      cpuload(70)                  /// start if <70% CPU load
                      maxsessions(4)               /// maxium number of sessions
                      update(3)                    //  update every 3 seconds
```

Only selected sessions run

```
Update at 21 Sep 2026: 15:27:46
  Logical Processors: 14 | CPU Load:14% | Free memory: 10%
  You have 1 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Combine	30512	Completed	21Sep2026: 15:27:31	21Sep2026: 15:27:34
2	Results	9416	Active	21Sep2026: 15:27:32	

mistake.do

- Detecting errors is important.
- I will now include the following do file.

```
1  // mistake.do
2  use results/relsurv_combined, clear
3
4  // The following line is a deliberate mistake
5  XXX is not a Stata command
```

Monitoring log files

```
. list SessionName dofile region component, noobs sepby(component) abbrev(12)
```

SessionName	dofile	region	component
Setup	Data_setup.do	.	data setup
Setup	More_Data_setup.do	.	data setup
wait		.	
N & York	net_survival.do	1	analysis
Trent	net_survival.do	2	analysis
wait		.	
Combine	combine_results.do	.	postestimation
wait		.	
Results	mistake.do	.	postestimation
Results	graphs_comparison.do	.	postestimation
Results	tables_simple.do	.	postestimation

Error reported in output

```
Update at 21 Sep 2026: 15:49:02
  Logical Processors: 14 | CPU Load:33% | Free memory: 20%
  You have 0 active sessions (excluding this one).
```

ID	SessionName	PID	status	starttime	stoptime
1	Setup	148	Completed	21Sep2026: 15:48:29	21Sep2026: 15:48:39
2	N & York	15964	Completed	21Sep2026: 15:48:39	21Sep2026: 15:49:00
3	Trent	14028	Completed	21Sep2026: 15:48:40	21Sep2026: 15:49:00
4	Combine	27004	Completed	21Sep2026: 15:49:00	21Sep2026: 15:49:01
5	Results	36576	ERROR	21Sep2026: 15:49:01	21Sep2026: 15:49:02

```
At least one session exited with an error code.
The log files where errors occurred are listed below.
   _sessionslog/session5_20260921T154901.log
```

The logfile is clickable, so can see where error is immediately.

Log file

```
. do "do/mistake.do"  "."

. // mistake.do
. use results/relsurv_combined, clear


.
. // The following line is a deliberate mistake
. XXX is not a Stata command
command XXX is unrecognized
r(199);

end of do-file
r(199);

end of do-file
```

Summary

- Useful for separate analyses (by cancer site, simulation scenarios etc)
- Some other useful other commands, but none worked in the exact way we wanted.
- Some thought from user needed about how to split up tasks.
- Clickable log files useful to detect errors.
- Currently only windows, but use of Java should help with implementation accross platforms.

Test version available

`https://pclambert.net/software/sessions.html`