

# SAS 9.4 on the Engaging HPC Cluster

Steven Finch, [sfinch@mit.edu](mailto:sfinch@mit.edu)

Research Computing Specialist, Sloan Technology Services

This tutorial will grow with time. My purpose is to give simple examples, using only basic statistics and rudimentary SLURM (job scheduling) commands. As questions arise, I will update and add materials as needed.

Let us start with batch job submissions. SSH log into “eosloan1.mit.edu”. Upload (using SecureFX, for example) the following data file

[furnace.csv](#)

to your Engaging directory and (optionally) read

[furnace\\_desc.txt](#)

for background. To the same directory, upload the SAS 9.4 program

[furnace.sas,](#)

a collection of various subroutines

[SAS\\_macros.sas](#)

and the shell script

[furnace\\_SAS.sh](#)

containing a number of commands starting with #SBATCH. We will explain these commands shortly; for now, our focus is on getting various statistical calculations to work. Type the command

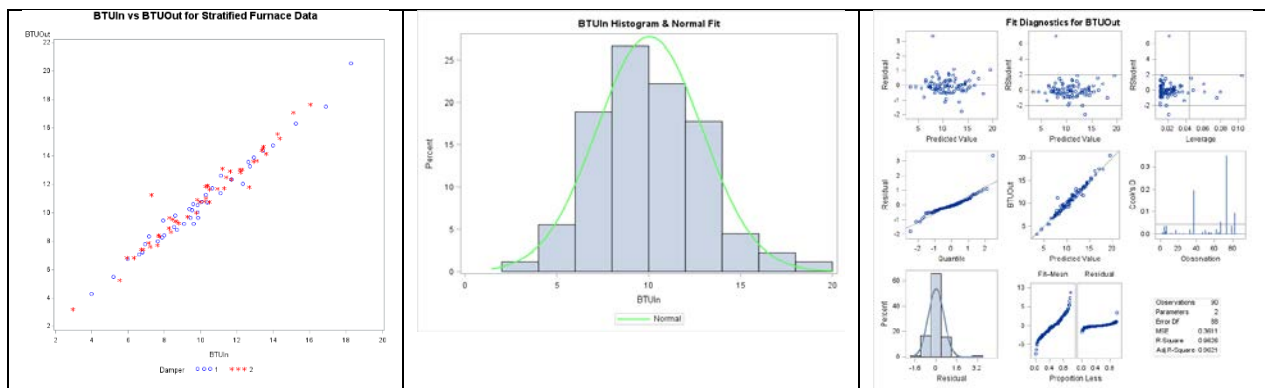
```
sbatch furnace_SAS.sh
```

in your remote terminal (SecureCRT or PuTTY, for example) and hit <Return>. You have now submitted a job to Engaging! Because this job is very short, seven new files should readily appear in your directory:

furnace\_SAS\_err.txt, furnace\_SAS\_out.txt, furnace.log, furnace.lst

scatterplot-BTU.pdf, histogram-BTU.pdf, regression-BTU.pdf

The first two of these should be empty since no errors ought to have occurred in the program execution. Download the remaining five files to your local computer. The LOG file is self-explanatory; the LST output file contains all numerical tables from the job; the PDF output files correspond to three plots (the latter consisting of nine subplots). If you wish to explore other statistical aspects of the furnace dataset, simply edit my SAS code, overwrite the original program and resubmit the job.



Let us turn now to understanding the shell script:

```
#!/bin/bash

#SBATCH --job-name=furnace_SAS
#SBATCH --output=furnace_SAS_out.txt
#SBATCH --error=furnace_SAS_err.txt
#SBATCH -p sched_any_quicktest
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=2000

module load engaging/sas/9.4

srun xvfb-run sas -noterminal furnace.sas -filelocks none -work
/nobackup1/username
```

Note that, while the standard output and standard error filenames are prescribed in #SBATCH lines, the graphical PDF filenames are given within the SAS code itself (at various locations). The argument

```
-work /nobackup1/username
```

within the “srun” line places the SAS temporary work directory on the very fast Lustre parallel file system for parallel input/output; please replace “username” by your MIT Kerberos username. The line

```
#SBATCH -p sched_any_quicktest
```

specifies the SLURM partition (or queue) under which the script will be run. The “sched\_any\_quicktest” partition is ideal for small, classroom examples like ours. It imposes a maximum runtime of 15 minutes on any job. Another partition, named “sched\_mit\_sloan”, imposes a maximum runtime of 2 weeks. Since many pre-existing jobs may already be running on this partition, you might be forced to wait a long time for your task to begin.

The lines

```
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=2000
```

require more explanation. For now, it suffices to say that we’ve requested 1 core on the cluster using 2 GB of RAM per core. Memory-intensive jobs will necessitate more cores. To be continued...

We finally discuss briefly how to open an interactive SAS session on Engaging. SecureCRT no longer suffices; you will need to use an X forwarding client such as MobaXterm (for Windows) or XQuartz (for Mac). With this in place, the SLURM commands

```
module load engaging/sas/9.4
srun --x11 --cpus-per-task=2 --mem=8000 --pty -p sched_mit_sloan sas
```

open a rudimentary SAS interface. Here we've requested 2 cores on the cluster using a total of 8 GB of RAM (hence 4 GB per core). This contrasts with our earlier batch SAS job, which did not use multiple cores. We recommend inserting the SAS command

```
ods pdf file='output.pdf';
```

near the beginning of your session, well before any tables or graphs are produced, and

```
ods pdf close;
```

at the conclusion of your session. Otherwise numerical and visual results will not become available (due to the default HTML presentation of such results in SAS: the inclusion of a web browser on Engaging is not feasible). Because of formidable limitations in interactive SAS via MobaXterm or XQuartz, we further recommend that you install SAS on your local computer, reserving the cluster for batch jobs only.

Please feel free to write to me ([sfinch@mit.edu](mailto:sfinch@mit.edu)) with questions and suggestions on how to improve this tutorial.

### **Addendum: Selecting text in SAS Unix**

Selecting portions of text interactively in SAS Unix is different than in SAS Windows! For Unix, you should: (i) click the pointer at the beginning of any section you wish to select; (ii) click Select from the Edit menu; and (iii) click the pointer at the end of the section to be selected. Selection mode stays in effect as long as any text is highlighted. You may need to end text selection mode before you can use other commands, or start a new selection. On the Edit menu click Deselect. All highlighting should then turn off.

### **Addendum: Replacing DOS line breaks**

If you receive a "Batch script contains DOS line breaks instead of expected UNIX line breaks" error message, you need to replace all DOS carriage returns by UNIX carriage returns. This is easily done on the UNIX command line within Engaging:

```
dos2unix furnace_SAS.sh
```

and eliminates the need to do further editing using VI or other UNIX text editor.