Transitioning from PBS to SLURM for fun and profit

Chun Lab Meeting 12/18

Using the Supercomputer

- A great resource to improve your life.
    - Not just for bioinformatics! Any processor intensive task you do on a computer can be done on the supercomputer.
- Previously, we used PBS to schedule jobs
- Now we’re using SLURM.
- Learn the new syntax!
    - Example using R

Things that are the same

- To use the supercomputer, you'll need to be connected to a UMN network
  - If off campus, use VPN to connect.
- Logging into MSI is the same: If on a mac or linux, use **terminal** and ssh. If on a PC, use **PuTTy**.

```
ssh login.msi.umn.edu -l yourx500
```

Things that are the same

- The supercomputer has a few computing clusters. The one I most frequently use is **Mesabi**, but **Mangi** is also an option. Other clusters have a graphic user interface, which is comforting, but those clusters generally have fewer resources available.

```
ssh mesabi
```

(you’ll need to enter your x500 password again)

Things that are Different

Making an Interactive Job

Old PBS way

```
qsub -I -lwalltime=1:00:00,nodes=1:ppn=8,mem=16gb
```

New SLURM way

```
srun -N 1 -n 8 --mem=16gb -t 1:00:00 -p interactive --pty bash
```

- The spaces are very important.
- **-N 1** requests one node
- **-n 8** requests 8 processors
- **--mem=16gb** requests 16gig of ram
- **-t 1:00:00:00** is your time limit, in HH:MM:SS
- **-p interactive** sends your job to the interactive queue
- **--pty bash** tells slurm you will be using a bash shell.

What to do next?

Example

- Right now most users are learning how to use SLURM, so securing resources is very fast!

- If you’ve been using MSI to run mothur, continue as normal. Everything is the same.

- For everyone else, maybe you want your R scripts to run faster. Let’s run R.

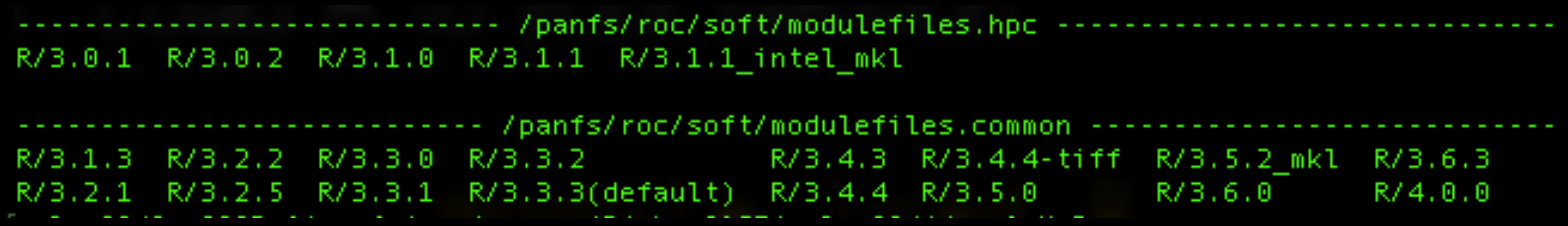
What to do next?

Example

1. Load R to your session.

```
module load R
```

This loads the system default version of R, which is probably the most recent. For best chances of success, load the same R you work with on your computer. Check which ones are available like this:



```
------------------------------ /panfs/roc/soft/modulefiles.hpc ------------------------------
R/3.0.1  R/3.0.2  R/3.1.0  R/3.1.1  R/3.1.1_intel_mkl

----------------------------- /panfs/roc/soft/modulefiles.common -----------------------------
R/3.1.3  R/3.2.2  R/3.3.0  R/3.3.2           R/3.4.3  R/3.4.4-tiff  R/3.5.2_mkl  R/3.6.3
R/3.2.1  R/3.2.5  R/3.3.1  R/3.3.3(default)  R/3.4.4  R/3.5.0       R/3.6.0      R/4.0.0
```

```
module load R/3.5.0
```

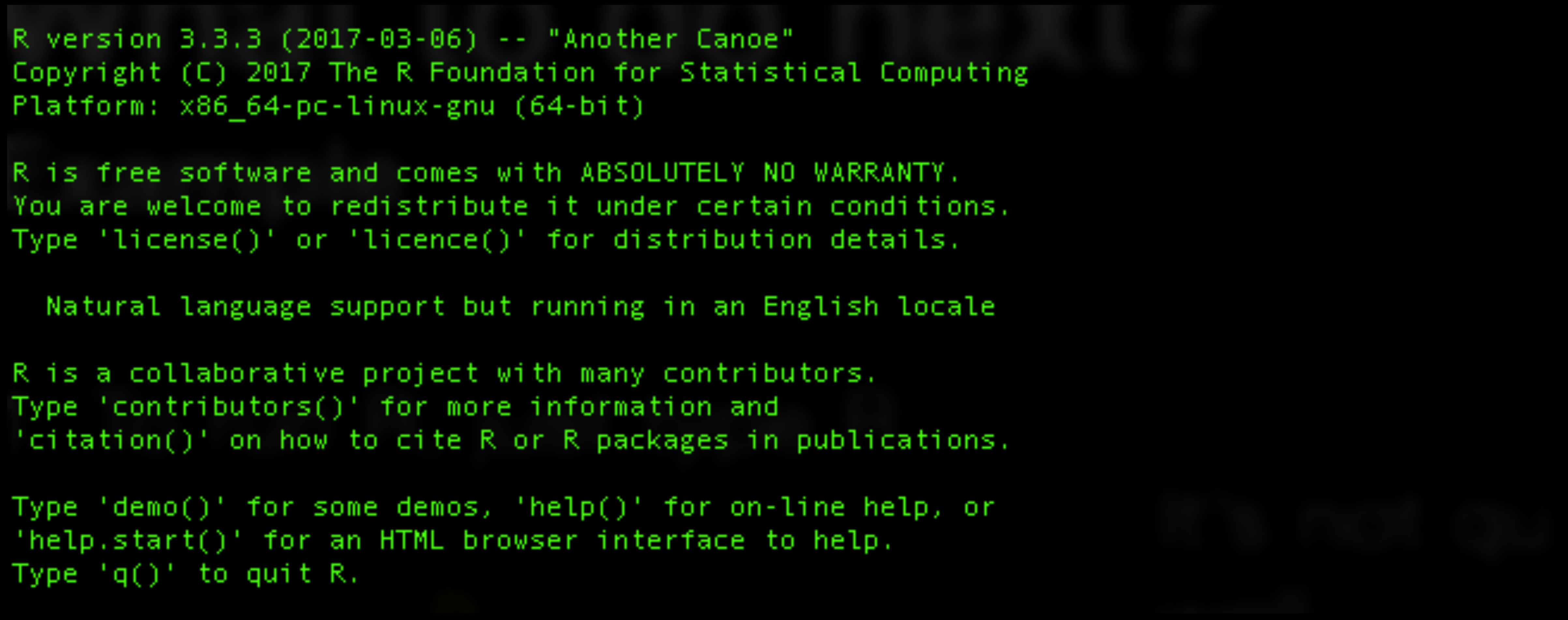
What to do next?

Example

2. To run R, just type R

It’s not quite as nice as RStudio, but works just as well.

```
R
```



```
R version 3.3.3 (2017-03-06) -- "Another Canoe"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

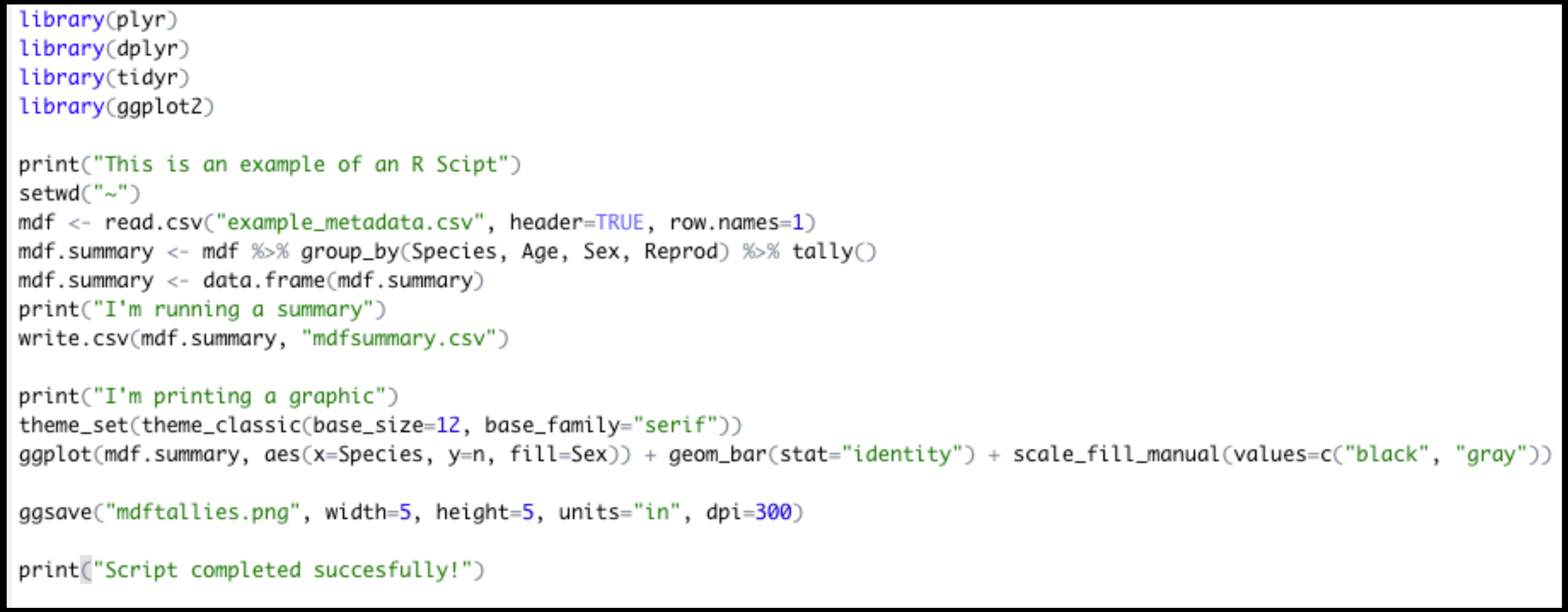
You can install packages just like normal, using install.packages(). Everything will be installed to your directory only.

What to do next?

Example

Maybe most usefully, you can also run an R script noninteractively.

Here's an example that I wrote on my laptop.



```
library(plyr)
library(dplyr)
library(tidyr)
library(ggplot2)

print("This is an example of an R Scipt")
setwd("~")
mdf <- read.csv("example_metadata.csv", header=TRUE, row.names=1)
mdf.summary <- mdf %>% group_by(Species, Age, Sex, Reprod) %>% tally()
mdf.summary <- data.frame(mdf.summary)
print("I'm running a summary")
write.csv(mdf.summary, "mdfsummary.csv")

print("I'm printing a graphic")
theme_set(theme_classic(base_size=12, base_family="serif"))
ggplot(mdf.summary, aes(x=Species, y=n, fill=Sex)) + geom_bar(stat="identity") + scale_fill_manual(values=c("black", "gray"))

ggsave("mdftallies.png", width=5, height=5, units="in", dpi=300)

print("Script completed succesfully!")
```

If you want to follow along in this section, download the R script and metadata here: https://github.com/mgaley-004/MiSeq-Analysis/tree/main/Help

Before running it, I need to make sure I have installed any libraries I use on MSI and then upload the script. I also need to upload "`example_metadata.csv`" since my script reads it in.

What to do next?

Example

To upload files, you **can** use a GUI like WinFCP or FileZilla, but I would really encourage you to try using sftp directly through the terminal.

In a new window: Connect using either ssh (mac/linux) or pUTTy.

You must first navigate to the folder where your files are, like this:

```
cd /volumes/dean/chunlab/meetings
```

Then connect like this:

```
sftp yourx500@login.msi.umn.edu
```

What to do next?

Example

Once connected, you can navigate around your MSI filesystem normally and upload files using the command `put`

```
put meetingExample121820.R

put example_metadata.csv
```

Keep this window open since we will return to it in just a moment.

What to do next?

Example

Now that everything is in place, you can return to your SLURM window and run the script.

```
Rscript meetingExample121820.R
```

That’s it! Some output will be printed to your screen, and two new files will be create

What to do next?

Example

To examine the output, use sftp to download the files to your computer.

In the sftp window, view the newest files:

```
ls -lt
```

Then download them by using the `get` command.

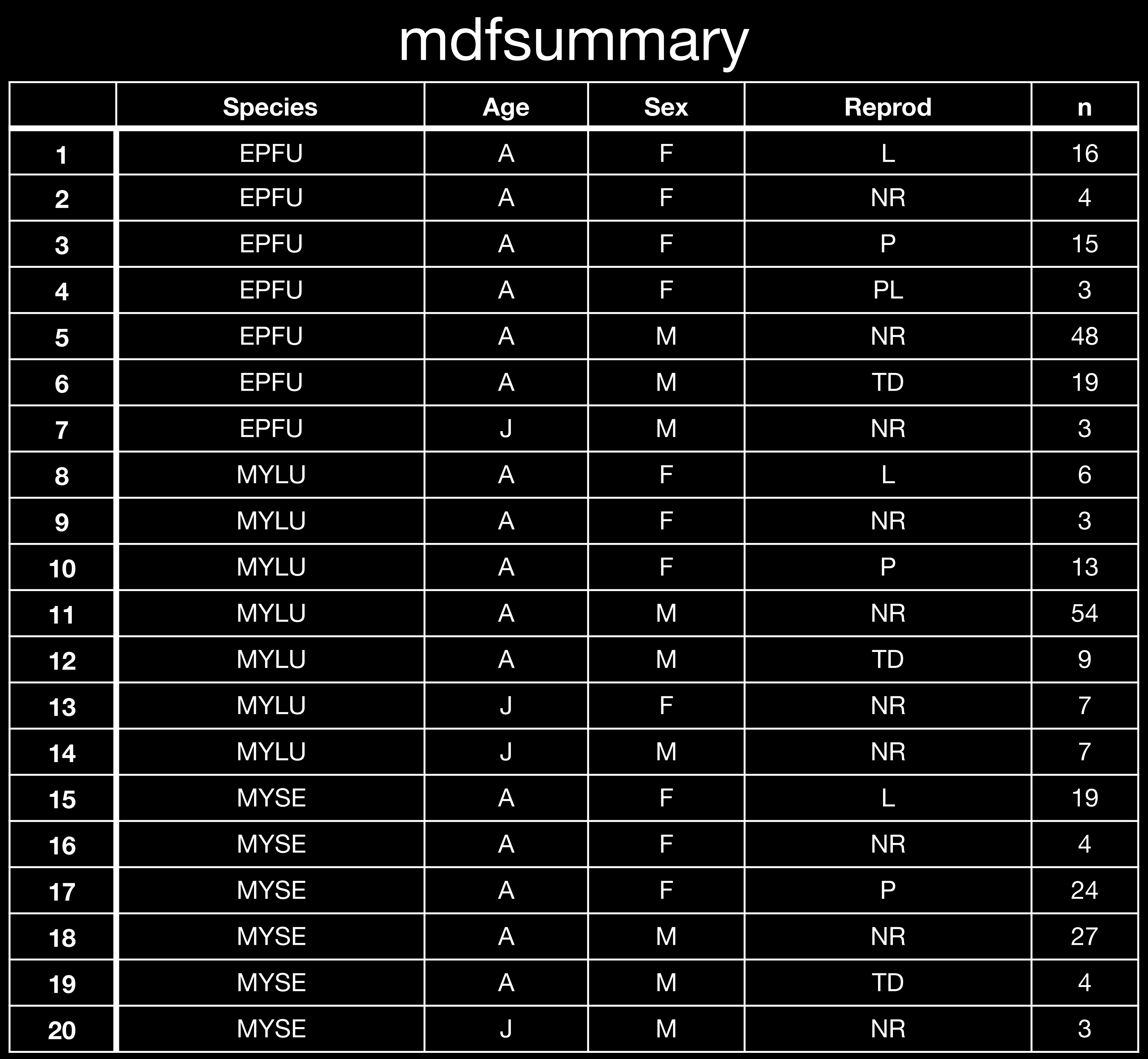
```
get mdfsummary.csv

get mdftallies.png
```

Close sftp by typing `quit`

What to do next?

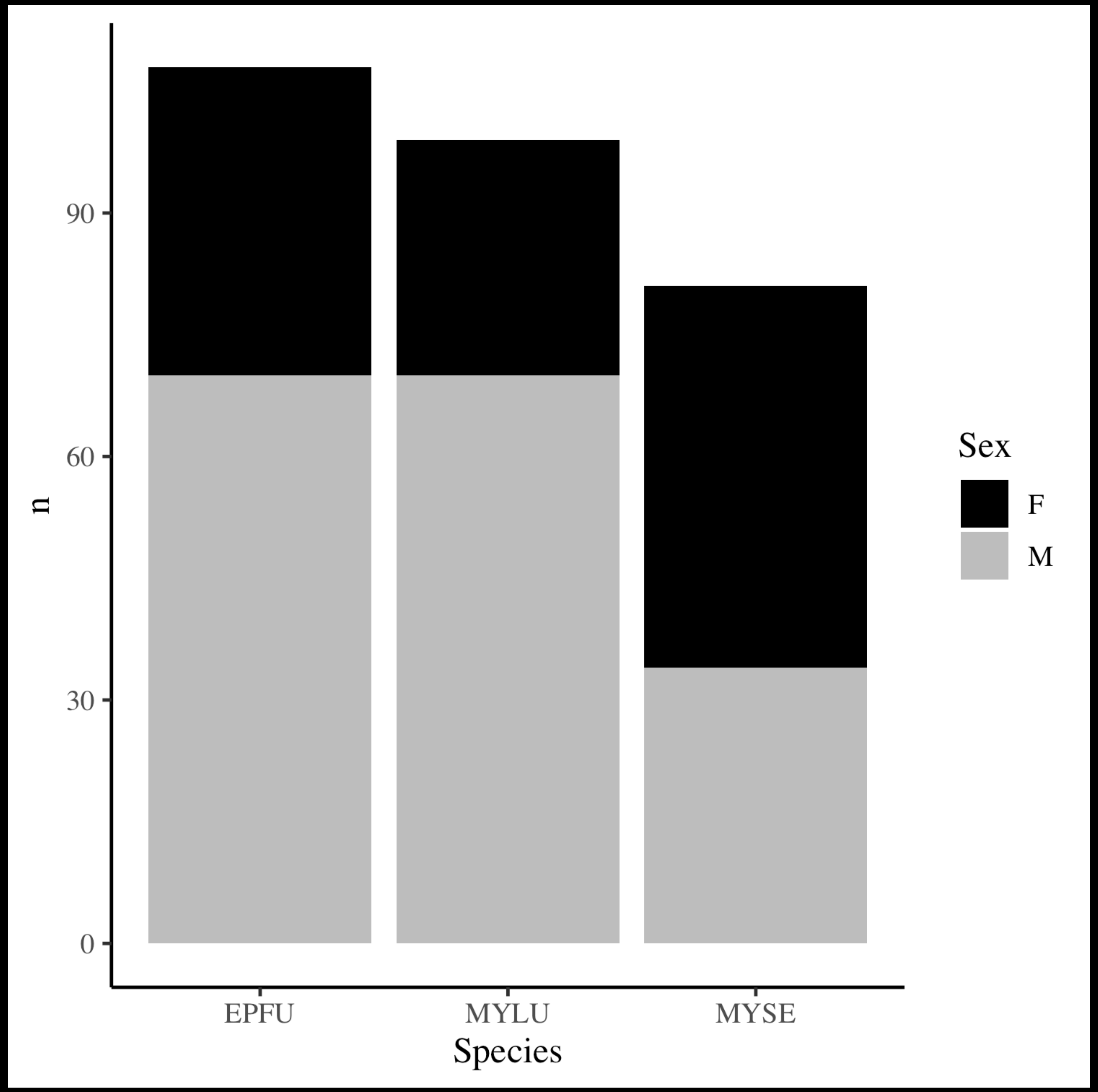
Example

If you open these files, you should see something like this:



mdfsummary

	Species	Age	Sex	Reprod	n
1	EPFU	A	F	L	16
2	EPFU	A	F	NR	4
3	EPFU	A	F	P	15
4	EPFU	A	F	PL	3
5	EPFU	A	M	NR	48
6	EPFU	A	M	TD	19
7	EPFU	J	M	NR	3
8	MYLU	A	F	L	6
9	MYLU	A	F	NR	3
10	MYLU	A	F	P	13
11	MYLU	A	M	NR	54
12	MYLU	A	M	TD	9
13	MYLU	J	F	NR	7
14	MYLU	J	M	NR	7
15	MYSE	A	F	L	19
16	MYSE	A	F	NR	4
17	MYSE	A	F	P	24
18	MYSE	A	M	NR	27
19	MYSE	A	M	TD	4
20	MYSE	J	M	NR	3



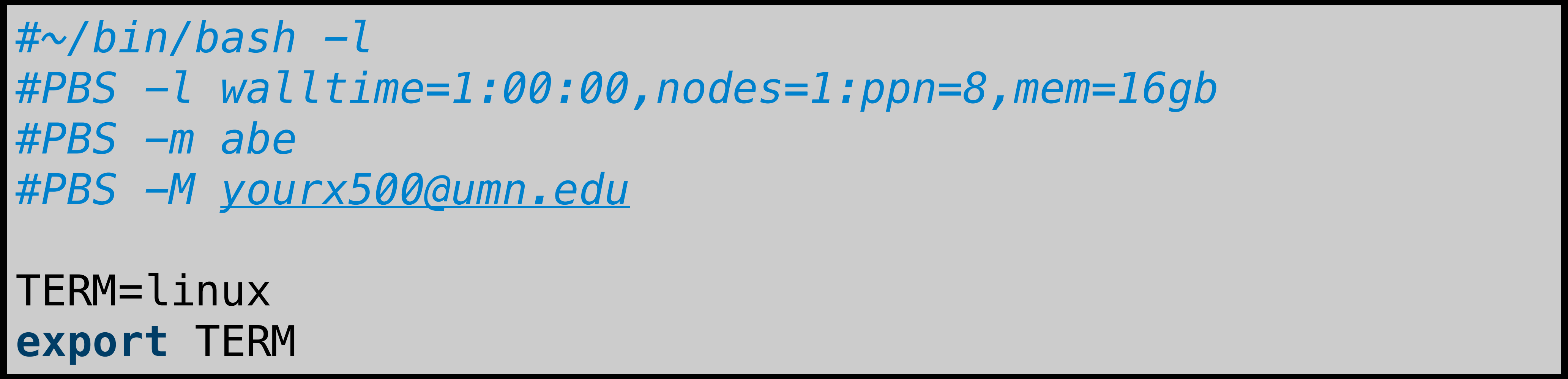
Things that are Different

Submitting a remote job

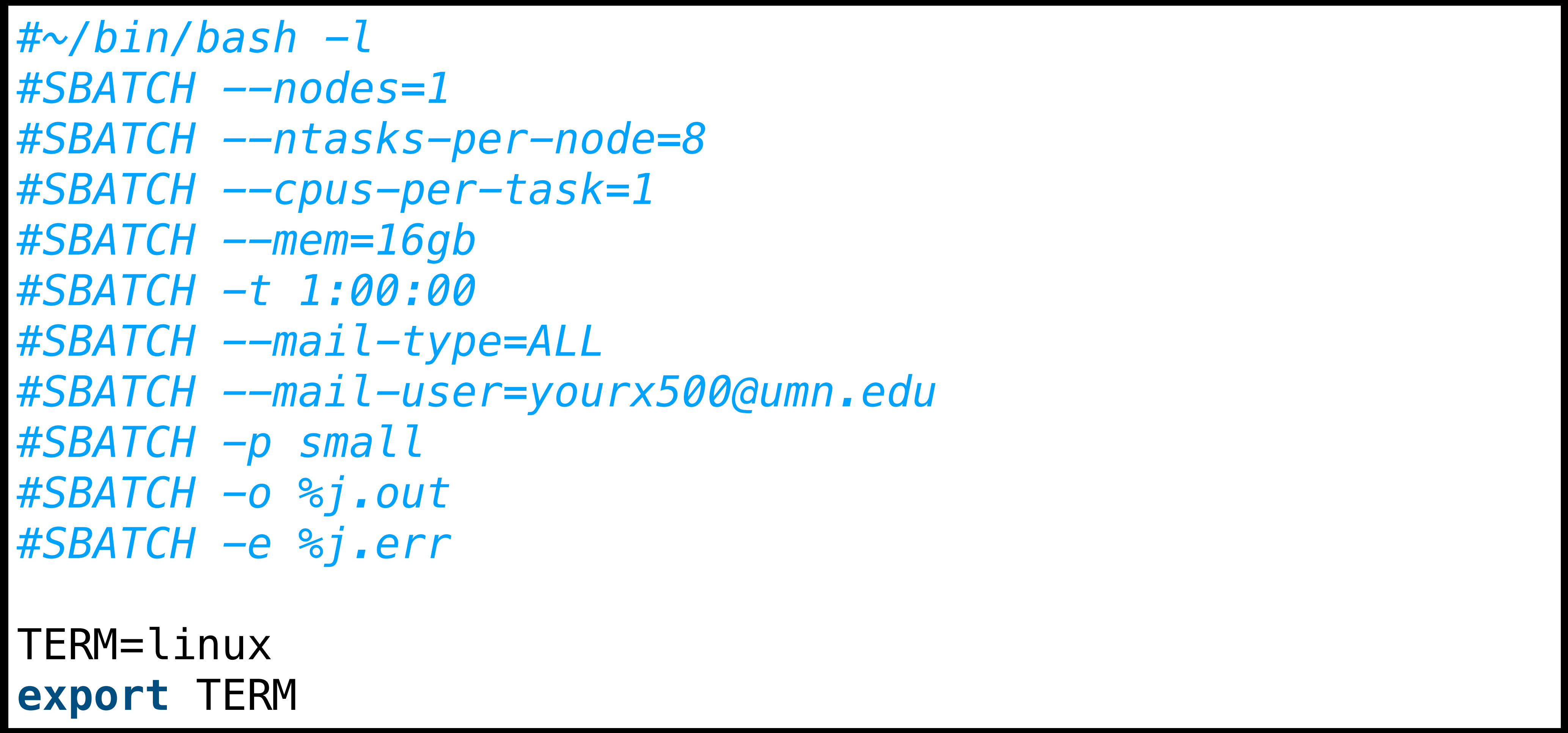
Old PBS way

New SLURM way

The header has changed significantly



```
#~/bin/bash -l
#PBS -l walltime=1:00:00,nodes=1:ppn=8,mem=16gb
#PBS -m abe
#PBS -M yourx500@umn.edu

TERM=linux
export TERM
```



```
#~/bin/bash -l
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH --cpus-per-task=1
#SBATCH --mem=16gb
#SBATCH -t 1:00:00
#SBATCH --mail-type=ALL
#SBATCH --mail-user=yourx500@umn.edu
#SBATCH -p small
#SBATCH -o %j.out
#SBATCH -e %j.err

TERM=linux
export TERM
```

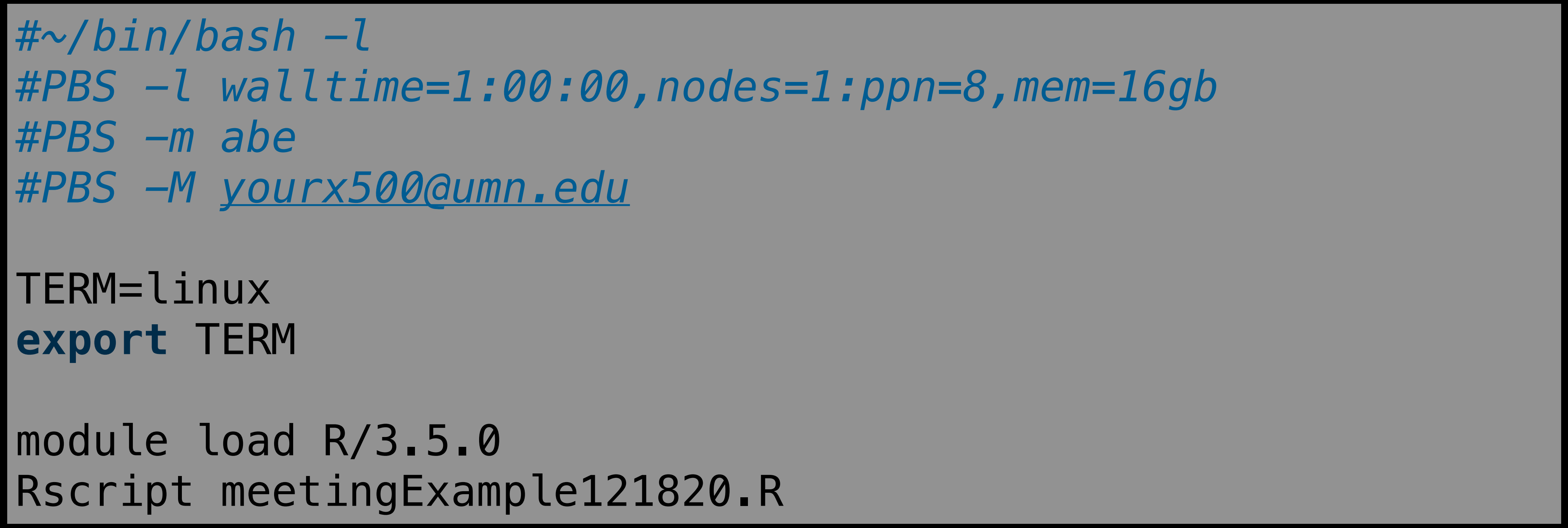
Things that are Different

Submitting a remote job

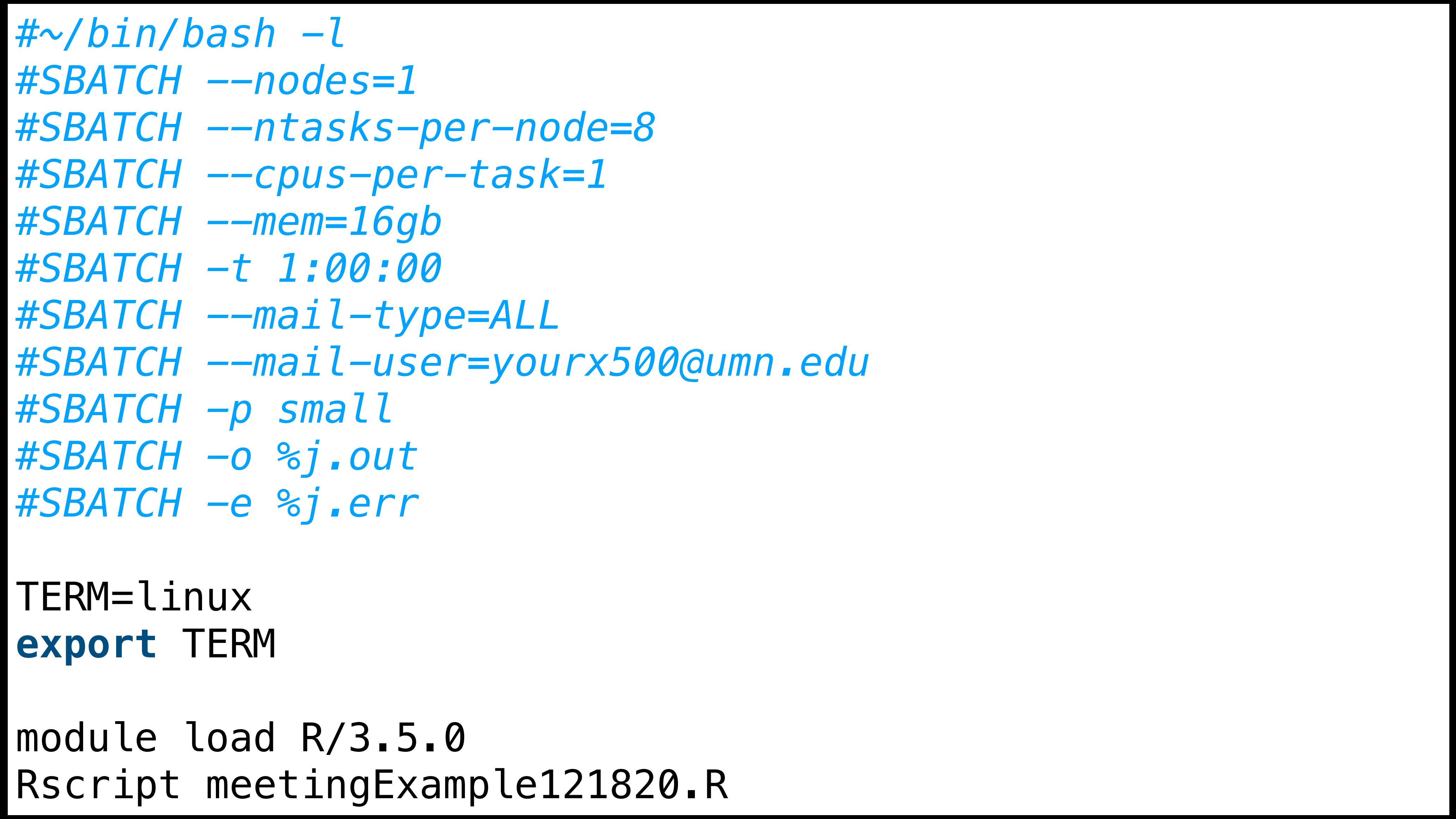
Old PBS way

New SLURM way

The rest is the same. To continue our R example:



```
#~/bin/bash -l
#PBS -l walltime=1:00:00,nodes=1:ppn=8,mem=16gb
#PBS -m abe
#PBS -M yourx500@umn.edu

TERM=linux
export TERM

module load R/3.5.0
Rscript meetingExample121820.R
```



```
#~/bin/bash -l
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=8
#SBATCH --cpus-per-task=1
#SBATCH --mem=16gb
#SBATCH -t 1:00:00
#SBATCH --mail-type=ALL
#SBATCH --mail-user=yourx500@umn.edu
#SBATCH -p small
#SBATCH -o %j.out
#SBATCH -e %j.err

TERM=linux
export TERM

module load R/3.5.0
Rscript meetingExample121820.R
```

Things that are Different

Submitting a remote job

Old PBS way

```
qsub -q small yourjobname
```

New SLURM way

```
sbatch yourjobname
```

Submitting is very similar, but the queue is specific in the job header rather than at submission time.

You will recieve an email when your job starts and when it concludes.

More resources

- MSI currently has a whole set of pages devoted to helping you transition your scripts to slurm. See here: https://www.msi.umn.edu/content/job-submission-and-scheduling-slurm
- Navigating your file system and the supercomputer file system is an important skill. Learn more about unix navigation here: http://www.ee.surrey.ac.uk/Teaching/Unix/
- Learn more about Windows file system navigation here: https://www.cs.princeton.edu/courses/archive/spr05/cos126/cmd-prompt.html
- It can be easy to get mixed up switching back and forth between your Windows file system and Unix. See this handy cheat sheet: https://www.lemoda.net/windows/windows2unix/windows2unix.html
- “I like computer science and bioinformatics and I want to learn more!,” you say? Start here: http://rosalind.info/problems/list-view/?location=python-village
- Download PuTTy: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- Help using sftp: https://linuxize.com/post/how-to-use-linux-sftp-command-to-transfer-files/