Statistics Computing Lab Manual

Mike Cammilleri

University of Wisconsin - Madison
lab@stat.wisc.edu

August 6, 2018

Abstract

This manual is an attempt to give a brief overview of the computing facilities and infrasturcture offered at the Department of Statistics at University of Wisconsin - Madison. It is not intended to be fully complete and will be updated at odd intervals as time becomes available. Information regarding day-to-day computing such as thin-client terminal skills, user names and passwords, printing, email, calendaring, disk quota and available software will be provided here. In addition this document will cover advanced topics in research computing such as the department high throughput computing (HPC) cluster, software, research computing methods, teaching tools and where to find help.

Contents

I. Introduction

Computing in Statistics consists of several major areas that will be covered in this document. Subject headings will cover these areas in order starting with day-to-day computing tasks and increase in complexity to research computing topics.

II. Department Productivity Computing

This section deals with the tasks relating to daily use of a computer. Word processing, email, logging in, printing, networks, personal laptops, and so on.

i. Accounts

There are two types of accounts: the UW Netid which is managed by central campus at `http://my.wisc.edu/` and department accounts managed by the Statistics department. These two accounts have separate passwords and give authorization for separate resources. The main differences are:

- NetID gives access to wireless networks and email at `http://wiscmail.wisc.edu`
- NetID authorizes access campus level service sites such as Box, My.wisc.edu, Moodle, Learn@UW, etc.
- Department account gives access to workstations, servers, and our website `http://www.stat.wisc.edu`

In order to utlize any of these accounts, you must first activate your campuse NetID upon university admission at `http://mynetid.wisc.edu`. Once this is completed, wait for the department administration office at Statistics to tell you to activate your Stats account and do this at `http://www.stat.wisc.edu/services/account-activation/`.

ii. Department Website

Use the department website to get information about faculty, students, staff, and IT services. An updated respository of department technical reports are maintained here as well. Most preliminary information about computing in Statistics can be found under the **Services** tab. Check here first before contacting the lab with basic questiongs regarding daily tasks such as priting, email, etc.

iii. Computer Lab

There is a small computer lab available to anyone with a Statistics login in room 1270. Six computers are available for use as well as a printer/scanner. The scanner will scan directly to email - follow the instructions under "black and white pdf" on the printer touch screen. This room is locked after hours.

iv. Email

You have two email addresses: <netID>@wisc.edu and <statID>@stat.wisc.edu. **Both email addresses have the same inbox located in your WiscMail email account which is also referred to as "Office 365."** Access your email via the web client at `http://wiscmail.wisc.edu/`. You can choose which email address is your preferred address to appear to your recipients by administering your account at `http://wiscmail.wisc.edu/admin`. You may set other options there as well such as vacation/away automatic replies, forwards, etc.

v. Printing

- All printers have names correlating to which room they are located. For example, pr1270 is a printer in room 1270
- **Graduate students receive a paper quota of 2000 pages per semester.** Visitors receive 500 pages, and faculty have unlimited quota
- The Computer Lab staff does not have authority over paper quotas and will not increase the limit for you. The paper quota amount is set and enforced by the Depart-

ment. If you run out of quota please see Denise Runyan in room 1220
- If you need to print multiples of the same document, e.g. handouts for class, print once and use the copy machine in room 1224. You will need a copy code, see Denise Runyan
- Large print jobs, like journals and book chapters should be printed using other services. **Please do not print hundreds of pages at a time.**
- Color printers available upon request

At times you may need to check your print quota. From a linux terminal window the command to run is

```
lpquota <username>
```

An example of the output looks like this:

```
[mikec@darwin03] (8)$ lpquota cdong
User                Quota       Printed
cdong               500             20
```

Print queues list what is currently in line for printing on any given printer. To check if your job or another person's job is waiting in the queue,

```
lpq -Ppr1270
```

where -P is the printer option and **there is no space between the option and printer name**, in this case pr1270. The output example is

```
[mikec@darwin03] (9)$ lpq -Ppr1270
pr1270 is ready
no entries
```

vi. Networks

The network is a shared file space where you may store data for backup, share files with peers and access software. The type of shared file system in use at Statistics and Computer Sciences is the Andrew File System, commonly referred to as "AFS." When logging into the department system you are in your home directory. All users have home directory space of 100 GB, located in an alphabetical hierarchy of folders. For example, user 'psmith' would be located at /u/p/s/psmith, where /u is the user directory space for all users, p is the first letter of the username, and s is the second letter of the username.

The important distinction to make is the difference between AFS storage space and local disk. Your data will be available across all computers in the department if it is located in AFS space, and it will have periodic backups that can be used for recovery. Local disk, such as your computer hard drive, laptop, mobile phone, etc., is not universally accessible or backed up on our system. Please be aware of where you are storing data.

If you exceed your quota of 100 GB, you will not be able to log into any computer in the department. Email lab@stat.wisc.edu or stop in room 1280 for help.

vi.1 AFS Tokens and Authentication Principles

The AFS filesystem relies on a kerberos ticketing system in order to determine correct authorization when reading and writing files across the network. When you log in via *ssh* or on a department thin client terminal computer, you will obtain a ticket which in turn grants you an AFS token used to authorize your account. You can check to see if you have a token by using the *tokens* command.

```
[mikec@lunchbox] (30)$ tokens

Tokens held by the Cache Manager:

User's (AFS ID 3691) rxkad tokens
    for cs.wisc.edu [Expires Aug
    29 09:37]
   --End of list--
```

You can see that the user *mikec* has an AFS ID of 3691, which is the token number. In many cases when a "permission denied" error occurs a user may not have tokens for a number of reasons. A good first troubleshooting step is to see if you have a token by running the *tokens* command. If you do not have a token issued to your user account the output will look similar to the following:

```
[mikec@lunchbox] (34)$ tokens

Tokens held by the Cache Manager:

    --End of list--
```

If you see this output you can try logging out and back into your session or run the *klog* command and provide the password for your Statistics user account. You **must** have valid tokens to do anything meaningful on the network.

vi.2 Wireless Access

All wireless is provided by campus and is not part of the Statistics network. This means you will connect to these wireless networks using your campus NetID and password, and should you need Statistics-provided services such as printing, you will have to use a VPN (http://vpn.wisc.edu/) client to authenticate. **The recommended wireless network is the one 'Eduroam' or 'UWNet.'**

vii. Software

Many types of software are provided by Statistics as well as by campus. To access some licensed software, go to `http://software.wisc.edu/` and log in with your UW NetID. As a student, faculty or staff member, you may be eligible for free download and licensing of software packages such as

- Adobe Acrobat Pro
- Maple
- Matlab
- Microsoft Office
- Microsoft Windows
- Norton Antivirus
- SAS
- SecureCRT
- SecureFX
- SPSS
- STATA

Much more specific software, customized for use in the Statistics Department, is available in AFS network file space. **Software that is updated and maintained is located in the /s directory.** Look in these locations for software, libraries, and compilers that you may need. If you do not see something you can make a request to lab@stat.wisc.edu for the software in question.

Some examples of software paths are

```
/s/python-3.5.2/bin/python3
/s/gcc-5.3.0/bin/gcc
/usr/bin/atril
```

In some cases you may want to have direct access to these programs without typing the full path name as shown in the example above. In this case you would need to update your 'PATH' as it is called in Linux type systems. PATH and other ENVIRONMENT VARIABLES can be set modifying your /.bashrc.local file. **See section viii.3**.

Also consider the DoIT Techstore located online at `http://it.wisc.edu/` and physically in the Computer Sciences building at 1210 W. Dayton St or at 333 East Campus Mall (corner of State St and Lake St). Software may be purchased there as well as hardware and peripherals.

viii. Linux

There is entirely too much to cover when discussing Linux. In this section only the most basic of information will be provided to get you started. Please consider Google your best friend for most Linux questions. Do not be afraid to ask your peers for help.

Most computers and servers in Statistics are running some version of Ubuntu. At the time of this writing, most are either Ubuntu 14.04 or 16.04.

viii.1 Terminal, xterm

Most work in Linux is done at the command line. You type your commands into the computer using the keyboard. You can open the window used to type commands by either selecting it from the Applications drop-down menu (MATE Terminal is recommended) or by using the hot key combination [Ctrl]+t.

When the terminal window first opens, it loads an environment that tells the window where things are located, how to interpret commands and which default options to use when executing programs and scripts - and much more. Each user has an environment defined by two files, .bashrc and .bashrc.local. **Any modifcations to your environment that are to remain permanent should go into .bashrc.local**. This file is located in your home directory. Note: the '.' that precedes the file name is what we call a 'dot file' and it is hidden from regular view.

Examples of useful entries to this file is explained later in this section.

viii.2 Quick, basic commands to know

The first commands you may want to know are the ones that help you move around the file system, changing directories, opening and closing files for editing, copying/renaming files, etc. Here's a quick list, each followed by a brief explanation

- cd - change directory (takes a path argument)
- clear - clear the current terminal winow screen
- locate - find a file
- ls - list files in current directory
- man - read the manual page for a given command
- mkdir - create a directory
- mv - move a file
- pwd - print working directory to screen
- rm - remove a file or directory
- rmdir - remove a directory
- touch - create a file without opening it

Commands related to the network file system

- fs listacl <path> - show the permissions for the network directory <path>
- fs seta <path> - set the permissions for the network directory <path>
- fs listquota - show the disk quota usage for current users' (given by the '~' character) home directory.

To know more about these commands and all their options, always read the manual page, often called the "man page" by simply typing

```
man <program name>
```

For example

```
man rm
```

would show the manual page for the 'rm' command (remove)

viii.3 bash shell and .bashrc.local

The shell in which you type commands is called the bourne again shell (bash). It's job is to interpret the commands you give it using aliases, environment variables, known paths, and other configurable parameters. The most basic example, and one of the most commonly used features is the PATH definition. When you type the name of a program, for example, 'ls', to list files in a directory, the shell needs to know where the 'ls' program actually is in the local file system on the hard drive. We can manually run the 'which' command to tell us where 'ls' is.

```
[mikec@ubuntultsp2] (2)$ which ls
/bin/ls
```

Here we see that 'ls' is in the directory /bin. We don't have to type '/bin/ls' because the bash shell will look in /bin because /bin is in the PATH definition in the file ~/.bashrc - therefor, we can simply type 'ls' and the program will run.

```
[mikec@ubuntultsp2] (13)$ ls
adm/  backups/  cache/  crash/
lib/  local/
```

The ~/.bashrc file contains a preset set of paths for the most common locations of programs. To add custom paths to other locations add them to ~/.bashrc.local, which is designed for users to append additional parameters and is included when bash reads ~/.bashrc.

```
PATH=$PATH:/s/slurm/bin # SLURM
    commands
```

The above example adds /s/slurm/bin to the PATH, so that any programs in that directory can be run without typing that full path. The # begins a comment.

NOTE: The order in which paths are listed in ~/.bashrc and ~/.bashrc.local is significant. If you have two paths for python listed in your PATH variable, for example:

```
PATH=$PATH:/s/python-3.5.2/bin
PATH=$PATH:/workspace/software/bin
```

when typing 'python' at the command prompt, the first python location will be the one that gets executed, in this case /s/python-3.5.2, and the python in /workspace/software/bin will be ignored.

You can also set an alias which will run a command based on a command you create.

```
alias rm="rm -i" # always ask
    before removing a file
```

You can do a lot more with the bash shell, use your Google skills to learn more.

viii.4 Personal Laptop/Computer Support Policy

Statistics computers fall into two categories: supported and unsupported machines. Supported Machines: Are installed with an operating system that is maintained by the Computer Sciences Laboratory (CSL). This means that it is campus licensed (if its Windows) and is updated on a regular basis with software patches, security updates, and virus protection. Statistics department supported machines include "thin-clients" (diskless workstations located on desks throughout the department and in the lab in room 1270), as well as desktops in the administration and faculty offices.

Unsupported machines are computers and devices that are personal, or otherwise not purchased on department or university funds. They may or may not have campus licensed software running. Unsupported machines are personal laptops, desktops, smart phones, iPads, etc. Unsupported can also mean the machine was purchased on department or university funds but does not have a supported operating system image from the Computer Sciences Laboratory. If you purchase a machine through grant funds but do not use a CSL supported operating system (i.e. you installed the operating system yourself from your own media, or campus media that is not configured by the CSL), then the Statistics department will not support the device.

Please be aware of whether you are using a supported or unsupported computer before submitting questions to lab@stat.wisc.edu.

viii.5 Purchasing Policy for Faculty

All faculty have the option to order a Computer Sciences Lab (CSL) and Statistics supported PC on their own grants. Although the use of privately owned laptops is encouraged, the use of any CSL/Statistics supported hardware is acceptable as an alternative to the widely popular thin client solution. The following restrictions apply:

- All Statistics supported machines will carry an operating system image that is created, maintained, and distributed by the Computer Sciences Laboratory (CSL). CentOS (RedHat) and Windows 10 are the currently available platforms.
- Only faculty may purchase a standalone Statistics supported PC for their direct use. Purchasing machines for students or others on behalf of faculty is not permitted.
- Upgrades to the hardware will be necessary to keep up with CSL operating system demands. The purchaser will be expected to cover all upgrade costs, usually once every three years. Components like RAM and hard disks are typically replaced as they wear out or become obsolete. Funds must be available for future maintenance.
- Any machine that does not carry a Statistics supported operating system image will not be supported by the Statistics Computing Lab (SCL) or the CSL. Any machine found not to be using Statistics supported hardware and software will not be put on the network. All machines must

be fully supported under the CSL/Statistics policy in order for the SCL to assist in maintaining.

If you are interested in purchasing a supported machine or have any questions, please send an email to lab@stat.wisc.edu.

viii.6 Remote computing (file access and printing)

WINDOWS

The two active wireless networks on campus are UWNet and Eduroam. UWNet is specific to UW-Madison campus only while Eduroam allows you to connect to hundreds of other campuses and research institutions using your UW-Madison campus NetID email and password. If authenticating to Eduroam use your entire campus email address as the username, e.g. bbadger@wisc.edu. If using UWNet just your NetID is needed.

To test the wireless connection, open a browser window (Firefox is the preferred browser, or use Internet Explorer). You should get a default web page with the title Welcome to UWNet with a left side-bar message Log in for Network Access.

To use the full network from a wireless connected laptop, you must log in to the network and be authenticated as a valid user. You can initially authenticate by entering your NetID and Password on the left side of the Welcome to UWNet screen. Once you get a Login Successful message, you will have access to the network. The preferred method for authentication is to install and use the WiscVPN client `http://vpn.wisc.edu/` First download and install Cisco Any Connect `http://kb.wisc.edu/helpdesk/page.php?id=39889` This authentication method provides greater security for communications. For additional details about the UWNet wireless setup and services, see the Wireless UWNet Info Page `https://kb.wisc.edu/helpdesk/page.php?id=3251`

PRINTING REMOTELY: To print on a Statistics printer in MSC,you must first setup a VPN connection to the Stat/CSL network. You can download the client for Mac or Windows. Launch the Cisco Anyconnect VPN Client software and copy and paste the central campus VPN connection address of dept-racssc.vpn.wisc.edu to connect.

The procedures for setting up local printer connections are basically the same as those described in the CSL Wireless and Laptop Network FAQ. `https://csl.cs.wisc.edu/services#Printing_From_The_Wireless_Lapto`

Queue names can be set to pr1270, pr1207, pr1224, pr1335 and prb248.

Detailed instructions follow however they are cumbersome. If you need assistance please stop into room 1280 but be advised that the help desk may not have time to assist with this and an appointment may be necessary.

WINDOWS

- Open Start Menu
- Select Devices and Printer
- Select Add a printer
- Select Add a network, wireless or Bluetooth printer
- Select The printer I want isn't listed
- Select Add a printer using TCP/IP address or hostname and click Next
- Enter the following parameters for the printer hostname/IP address:
- Device type: TCP/IP Devices
- Hostname or IP address: print-gw.cs.wisc.edu
- Port name: printer name example: pr1270
- Uncheck Query the printer and automatically select the driver to used
- Click Next. You may see a long delay on the Detecting TCP/IP port screen before the next screen is available for input. Please be patient; this delay is expected.
- Select Device Type: Custom then click Settings
- Enter the following Port Settings:
- Protocol: LPR
- LPR Settings Queue Name: enter queue name (example: pr1270)
- Check LPR Byte Counting Enabled
- Click OK

- Click Next Select the printer manufacturer and model. If you can't find the model you are interested in adding, you'll nee to download the driver, click Have Disk... and specify the driver location. The file you downloaded may be an exe file which will extract the needed drivers to a folder. This will be the folder you will wish to use when specifying driver location. See printer driver for more on finding printer drivers.
- Click Next
- Specify the name of the printer (for example pr1270, but your choice as to what you want to name it), then click Next
- On the Printer Sharing screen, select Do not share this printer and click Next. You may want to disable the Set as the default printer option (your choice)
- Click Finish

If you are an international student, or your PC or copy of Windows came from a foreign country, you may need to change the default paper size from A4 to Letter.

Go to Devices and Printers, if it is not already open. Right click on the printer you just added. Click Printing Preferences You will be presented with a window that gives you the option to select paper size. Select Letter (8.5" x 11") from the drop-down box if it is not already selected. (Please note that window you see may be different from the example picture depending on what printer you added.) Click OK. You are good to go.

You will be asked to select a manufacturer and model for your printer. You first need to check what specific type of printer you are printing to (pr1207, pr1270, pr1224, pr1335 and prb248). For statcolor, you will need the Xerox Phaser 6250DP PS driver. For If you can't find the driver for the Xerox Phaser 6250 you will need to go the the Xerox download site and select an appropriate driver for your OS. Choose the "Optional 32/64-bit Postscript Driver" for the greatest flexibility (on Windows systems). When installing, use the "Have Disk" option and point to the Xerox driver directory to load the Xerox Phaser 6250DP PS option into the printer list.

MAC

- Select System Preferences from the Apple pull down icon, then select Print and Fax from the Hardware line In the Print and Fax window, select the + sign to add a new printer, then choose the IP icon on the top row of the add printer window
- For Protocol select the Line Printer Daemon - LPD
- For a UWNet or Eduroam wireless connection, enter for Address
- For Queue, enter the queue identification for the printer you want to use, for example pr1270
- For Name, enter a useful name, like pr1270
- The printer Location information is optional, but can be set to something like 1270 MSC, for example
- Select Add to complete the configuration
- This will generally result in a two-sided (duplex) printing output, but there is some inconsistency in printer control available for some versions of browsers and OS X. Please check printer "layout" if you get something you don't expect. Normally, one-sided (simplex) printing is accomplished by adding a separate print queue for the simplex version of the printer. In this case the queue identification would be pr1270-simplex

The new printer should appear in the printing menu for OS/X applications. Select the new printer and test printing function.

There is a per-page charge to print on the Statistics color printer, so you need to have an account set up for this purpose. To set up an account, please contact Denise Runyan.

III. Teaching Tools

In this section we'll discuss the various resources available to you if you are teaching a class. In some cases, information found here such as calendaring, equipment checkout, and website hosting can be useful to everyone.

i. Reserving a classroom

This information can also be found on our website under the "Services" tab. `http://www.stat.wisc.edu/services/calendar`, **however, we'll include this information here as well.**

In Office 365 Web App, click the grid menu on the top left and select Calendar. Once in Calendar, right click on "Other Calendars" in the left-side menu. Select "Open Calendar" and a dialog box opens. In the "from directory" field, enter any of the following Statistics calendars.

ROOMS:

- stat-msc-1210-conf@stat.wisc.edu
- stat-msc-1217C-conf@stat.wisc.edu
- stat-msc-1475-conf@stat.wisc.edu
- stat-msc-1219-conf@stat.wisc.edu

The department also provides one laptop and several projectors for checkout. We use calendars to reserve them just like we do for rooms.

EQUIPMENT:

- stat-laptop-01@stat.wisc.edu
- stat-projector-01@stat.wisc.edu
- stat-projector-02@stat.wisc.edu
- stat-projector-03@stat.wisc.edu
- stat-projector-04@stat.wisc.edu

NOTE: Once you can view the calendar, you can add your reservations to them by creating a new "event". New events you create get added to your personal calendar. To make sure they also show up on the conference room or other resource calendar, add the calendar under the 'People' section of the event details. **Treat the resource you are making a reservation for as a person and it will show up on that resource calendar.** If you need to cancel your event/reservation, do so from your personal calendar and it will automatically remove from the room/resource calendar that was an attendee (See 'How to edit a resource reservation' in the next section).

If you wish for your event title to show up instead of your name, you can put the event title in the "Location" field. This is not intuitive and is a "known-issue" but is a workaround campus is providing at this time.

i.1 How to edit a resource reservation

If you create an event or meeting for a room, let's say room 1210 MSC, and it is incorrect - you cannot edit the event directly from the stat-msc-1210-conf room calendar.

When you create a room reservation on one of the room calendars Outlook will create a duplicate of the event on your own personal calendar (sometimes there is a delay before it shows up on your personal calendar - be patient). To make an edit to the room reservation on calendar stat-msc-1210-conf you would look for that same room reservation event/meeting on your own, personal calendar, and make edits from there. Those edits will automatically update the room reservation event/meeting on the stat-msc-1210-conf calendar.

i.2 Classroom AV Equipment

Room 331 lecture hall is our most robust classroom for technology. All controls are mounted on the podium on a convenient punch pad. Be patient when turning on the projector - it takes a minute. This room provides:

- HD 16:9 wide screen projector and screen
- HDMI, VGA and wireless video connections
- Document viewer
- Laptop audio, lavalier microphone, and wireless handheld amplification
- Lecture recording of video and audio signals straight to file - stored on media server

If anything is not working please ask for help from the lab.

Room 1210 and 133 have modernized projectors as well, but offer less for audio and lecture recording services. These rooms include:

- HD 16:9 wide screen projector and screen

- HDMI, VGA video connections
- Headset microphone (133 only)

You must provide your own video adapters for your personal laptops. They can be purchased at the DoIT Techstore in the Computer Sciences building. If you need assistance getting setup in any of these classrooms please ask someone in the lab at least 15 minutes ahead of your class period.

i.3 Recording lectures

As stated in the previous section, room 331 offers lecture recording. This is not live video of the person giving the lecture - it is a recording of the video signal to the projector along with the audio from the microphone.

To record a lecture, simply push the 'Record' button on the control panel on the podium. You will not see anything on the screen to indicate that recording is in progress, however, recording is happening behind the scenes. When you are finished with your lecture, contact lab@stat.wisc.edu to retrieve your lecture recording files. The lab can distribute them to you in different ways.

ii. Shiny Server for R

A Shiny Server is provided to host interactive R applications via the web. This service is free. It is hosted at `http://www.statlab.wisc.edu/shiny/`. To get your app hosted please email lab@stat.wisc.edu to set up a time to consult with lab staff. Depending on your app, certain R packages may need to be configured on the server. Once configuration is complete, we recommend hosting your app files on `https://github.com/` so that the server can regularly pull updates from your repository.

iii. Website Hosting

Each user in Statistics gets a personal web space. Your site is located at `http://www.stat.wisc.edu/~<username>/` where <username> is your Statistics login.

To begin your website, place files in /public/html. This directory is designed to take static HTML/php files.

If you want to host a website that requiers a database or some form of file writing to the server (form submission), you will need to email lab@stat.wisc.edu to have a different web space configured. The space in /public/html is for simple websites and is recommended for information regarding who you are, how to contact you, what courses you are teaching, a place to download homework, etc.

IV. Research Computing

The second major area of technology in the Statistics department is the research computing environment. Running time intensive computation while using a variety of statistical methods requires many different types of software and hardware that are usually kept separate from the day-to-day computing resources.

The Statistics Department has a host of hardware and software to aid in computation-intensive tasks. In conjunction with the Computer Sciences, the lab maintains their own data center and software repository tailored to the types of of tasks commonly requested by research in the field of Statistics.

In this section we'll cover statistical software, computation servers, high performance computing (HPC), high throughput computing (HTC), parallel processing, scripting, and most importantly, where to find help.

i. Research Software

There is a core set of software offered by the lab for research computing. It can often be confusing about the location of different softwares on the network since a lot depends on what your computation goals are. In general, most software can always be found in one of two places: either in /usr/bin which will be considered by default by most command line interpreters, or /s which is the network based location for custom and lab supported software versions.

Below is an example list of what is currently in /s on a Ubuntu 14.04 server:

```
[mikec@lunchbox] (23)$ ls -1
afstools@
afstools-1@
amd64_rhel6.r.151013
auks@
auks-0.4.0@
beagle-2.1.2@
bucky-1.4.4@
cmake-3.4.1@
cmake-3.6.2@
dhcp@
dhcp-4.3.0@
doxygen-1.8.12@
gcc-4.8.4@
gcc-4.9.4@
gcc-5.3.0@
gcc-5.4.0@
gcc-6.1@
gcc-6.2.0@
gtkwave@
gtkwave-3.3.76@
intelcompilers@
intelcompilers-2015@
intelcompilers-2017@
isl-0.16.1@
julia-0.5.0@
krb5@
krb5-1.0@
krb5-csl-0.2@
lab@
ldap-csl-0.2@
llvm-3.7.1@
lm-1.2@
maple@
matlab@
mentor-2016@
mrbayes-3.2.6@
mrbayes-3.2.6-1@
nagios-csl@
nagios-csl-0.3@
openblas-0.2.15@
openssl-csl-0.1@
perl@
perl-5.18.1@
perl-5.24.0@
perl-5.8.8@
phbook@
phbook-1@
postgresql@
postgresql-8.4@
postgresql-8.4.10@
python-2.7.12@
python-2.7.3@
python-3.5.2@
R@
rancid@
rr-project@
slurm@
slurm-16.05.0@
slurm-16.05.6@
spank@
spank-0.23@
sqlite-3.15.0@
std/
sushi@
sushi-0.3.19@
synopsys-2012_06_27@
synopsys-2016_09_23@
util@
vtune-xe-2016@
vtune-xe-2017@
w3m-0.4.2@
what@
what-0.1.31@
www@
```

As you can see various custom built software offered ranges from common applications such as Matlab and R, to specific versions of compilers like gcc-4.8.4, and more.

The software found in these two locations will cover nearly all research computing scenarios. However, there are times when a user may need something that is not currently available in which case they should email **lab@stat.wisc.edu**.

Users can also build and install their own software into their AFS home directory. There are many ways to this - the most common is using the *–prefix=* command during the configure stage.

```
{mikec@lunchbox} (24)$ ./
    configure --prefix=/u/m/i/
```

```
mikec/mysoftware
```

The above example tells the software you're configuring in the current directory to use the user home directory, in this case 'mikec', for the location of the installed program. User's are free to install software into their home directory but be aware that including library locations, specifying certain compilers, and linking other header files into the build can get complicated and is up to the user to decipher.

When in doubt, use the *which* command to see if a given program is within your command line path.

```
{mikec@lunchbox} (25)$ which gcc
/usr/bin/gcc
```

Special systems such as the High Performance Computing (HPC) cluster uses even more custom locations for software. Please refer to the section about the HPC cluster for more information on software specific to cluster computing.

ii. High Performance Computing Cluster

The bulk of all computation should be done on the Statistics HPC system. You may also hear it referred to as *"slurm"* as that is the name of the program used to submit jobs.

The primary reason for using the HPC cluster for computation is because it uses a scheduler system to prioritize each users job requirements without interactive user input. Jobs may be submitted to the system and when resources become available, the HPC system will reserve the resources for that particular job and queue any additional jobs that are awaiting more resources. This results in a balanced and 100 percent utilization of hardware nodes. It also allows for fair usage among groups of mutually exclusive research projects.

ii.1 Infrastructure

The cluster is currently made up of 13 compute nodes, two submit nodes, and two storage nodes. Each compute node consists of two Intel Xeon E5-2680 v3 @ 2.50GHz CPU's and 128 GB of RAM, and with hyperthreading there are total cluster resources of 624 cores and 1.7 terabytes of RAM available. High speed Infiniband interconnects are also available for true parallel processing procedures with speeds around 40gb/s or about 40 times the speed of "normal" gigabit networks.

Three 'partitions' are available to run your jobs depending on time required. To view available partitions to run your job on:

```
[mikec@lunchbox] (25)$ sinfo
PARTITION TIMELIMIT NODES STATE NODELIST
debug* up 2:00:00 1 idle marzano01
short  up 2-00:00:00 3 idle marzano[02-04]
long   up 5-00:00:0 3 idle marzano[05-13]
hipri  up 5-00:00:00 3 idle marzano[02-13]
```

NOTE: The *sinfo* command is located in the special software location of

```
/s/slurm/bin
```

We see three partitions in the above output - The first one listed is named 'debug' and this has a time limit two hours. This partition is default and should be used to test your job submissions or for very short run times. The second partition listed is named 'short' - it's time limit is two days. We also see that the 'long' partition has nine nodes (marzano[05-13]) with a limit of five days and the 'short' partition has a total of three (marzano0[2-4]) nodes. The third partition, 'hipri', is for special priority cases only. You can specify which partition to run your job in by using the -p option in your submit script.

ii.2 Getting Access to The HPC

Send mail to lab@stat.wisc.edu to request access to the HPC submit node named lunchbox. **From within the submit node, all your data must reside in the /workspace directory in order for the HPC cluster to read/write for your job.** The HPC cluster is NOT configured to run from within your users home directory in AFS. If you have custom R packages or software requests that you need, please email the lab and we will configure the cluster for your needs.

ii.3 SLURM

The job scheduler program used on the HPC system is called SLURM - or *Simple Linux Utility for Resource Management*. Read more about SLURM here `https://slurm.schedmd.com/` It is advised to read the manual pages in order to familiarize yourself with all the commands and options. Examples can be found on our website under the *Services -> HPC Cluster* tab, and some examples will be given here as well.

ii.4 Getting Started

There are few key concepts to know before getting started with the HPC.

- READ THE MANUAL - there is information in this document but for the most recent information always refer to the website at `https://www.stat.wisc.edu/services/hpc-cluster1/about`
- All jobs, data, and programs must be ran from the **/workspace/<username>** directory on *lunchbox.stat.wisc.edu*. This is because that location is visible by all nodes on the cluster.
- You must use software located in **/workspace/software/bin** or **/s**. Some software in /usr/bin will also work, but try to use software from these first two locations when possible.
- If using R, your path to R packages must be accesible by each node on the cluster. Therefore, all packages and special libraries needed by your job must be located in **/workspace/<username>/<someDirectory>** or request that the lab install packages for you. See the section on R libraries for more information on how to set your R library path.
- Always submit small/simple jobs to the cluster first before expanding to larger, longer running jobs.
- Data stored in */workspace* **is not backed up** and is periodically cleaned out at the beginning of semesters. Move all important data to your AFS home directory for backup or move to an external drive. There is 8 terabytes of storage in /workspace and it is shared among all users. If you have need for large data storage please email the lab.

ii.5 Requesting Access

Email lab@stat.wisc.edu and request that your account be given access to both submit nodes of the HPC which are named *lunchbox.stat.wisc.edu* and *jetstar.stat.wisc.edu*. Once a member of the lab staff adds your account to the system, it can take up to one hour before it will function fully.

Once you are given access you can ssh into a submit node to begin using Slurm on the HPC. If you are on Windows you will need to download a shell emulator such as *putty.exe* downloadable at `http://www.putty.org/`. On a Mac you can use the built-in app called *Terminal*. If you are on Linux or a department thin-client (which uses Linux) there's already an xterm or terminal application installed. On the department thin-clients in particular there is the *MATE Terminal* app that is recommended.

ii.6 Submitting Jobs

There is a lot of information regarding job submission to the cluster. There are many options and methods but the basic principal remains the same throughout.

Slurm commands are located in */s/slurm/bin*. The primary command to submit a batch job to the cluster is *sbatch*. This command is used to submit a shell script containing instructions for the cluster.

1. log into lunchbox or jetstar
2. cd /workspace
3. mkdir -p [your Statistics username]
4. cd [your Statistics username]
5. create your script file here - it can be called anything, for example *submit.sh*

An example *submit.sh* file with basic options:

```
#!/bin/bash
#SBATCH --mail-type=ALL
#SBATCH --mail-user=<user>@stat.
    wisc.edu
#SBATCH -o outputfile.out
#SBATCH -e error.out
#SBATCH -D /workspace/<username>
#SBATCH -J job_name
#SBATCH -t 72:00:00
#SBATCH -p long
#SBATCH --cpus-per-tas=4
#SBATCH --mem-per-cpu=1000M
module load R/R-3.5.0
R CMD BATCH --no-save myjob.R \
    myjob.out
```

Be sure to make your submit script executable:

```
chmod +x submit.sh
```

In the above example, the file *myjob.R* contains your R code. But before we run 'R CMD BATCH' we load the software using the 'module load' command. ALWAYS load software using the *module* command if possible as it will set all necessary paths and environment variables for use on the cluster. All code and data should reside in the same directory as your submit script file or otherwise you must specify the complete or relative path to the file.

TIP: To see a list of available software that can use the 'module' command, run 'module avail'. To unload a module, type 'module unload <module name>. To list which modules you currently have loaded type 'module list'.

You can now attempt to submit the job using the *sbatch* command.

```
[mikec@lunchbox] (5)$ cd /workspace/mikec/
[mikec@lunchbox] (6)$ sbatch submit.sh
Submitted batch job 793290
```

NOTE: A submitted job is given a job number, in this case, 793290. Use this number to keep track of your queued job.

ii.7 Job Status

To view currently running jobs, use the *scontrol* and *squeue* commands.

```
[mikec@lunchbox] (17)$ scontrol \
    show job 793290
JobId=793290 JobName=submit.sh
UserId=mikec(3691) GroupId=mikec
    (3691) MCS_label=N/A
Priority=110085 Nice=0 Account=
    mikec QOS=normal
JobState=RUNNING Reason=None
    Dependency=(null)
Requeue=1 Restarts=0 BatchFlag=1
    Reboot=0 ExitCode=0:0
RunTime=00:00:05 TimeLimit
    =4-00:00:00 TimeMin=N/A
SubmitTime=2016-12-06T12:47:28
    EligibleTime=2016-12-06T12
    :47:28
StartTime=2016-12-06T12:47:28
    EndTime=2016-12-10T12:47:28
    Deadline=N/A
PreemptTime=None SuspendTime=None
    SecsPreSuspend=0
Partition=short AllocNode:Sid=
    lunchbox:3213
ReqNodeList=(null) ExcNodeList=(
    null)
NodeList=marzano05
BatchHost=marzano05
NumNodes=1 NumCPUs=4 NumTasks=1
    CPUs/Task=1 ReqB:S:C:T=0:0:*:*
TRES=cpu=2,node=1
Socks/Node=* NtasksPerN:B:S:C
    =0:0:*:* CoreSpec=*
MinCPUsNode=1 MinMemoryNode=0
    MinTmpDiskNode=0
Features=(null) Gres=(null)
    Reservation=(null)
OverSubscribe=OK Contiguous=0
    Licenses=(null) Network=(null)
Command=/workspace/mikec/submit.sh
WorkDir=/workspace/mikec
StdErr=/workspace/mikec/slurm
    -793290.out
StdIn=/dev/null
```

```
StdOut=/workspace/mikec/slurm
    -793290.out
Power=

 [mikec@lunchbox] (18)$ squeue
JOBID PARTITION NAME USER ST TIME
    NODES NODELIST(REASON)
793290 short submit.s mikec R 0:06
     1 marzano05
```

The *sinfo* command shows current usage of all partitions on the cluster.

```
 [mikec@lunchbox] (9)$ sinfo
PARTITION TIMELIMIT NODES
STATE NODELIST
debug* up 2:00:0 1 idle marzano01
long up 5-00:00:0 4 mix marzano05
long up 5-00:00:0 2 idle marzano[06-13]
hipri up 4-00:00:00 2 idle marzano[02-15]
```

ii.8 Specifying Resources and Other Options

Other options are set using the '#SBATCH' directive. If a space is put after the '#' symbol, the line becomes a comment. The following options in the example submit script are explained below:

```
#SBATCH -o outputfile.out
#SBATCH -e error.out
#SBATCH -D working_Directory
#SBATCH -J job_name
#SBATCH -t 72:00:00
#SBATCH -p long
#SBATCH --cpus-per-task=4
#SBATCH --mem-per-cpu=1000M
```

SLURM defaults with slurm-.out for all output, but we can specify an output file with -o. We can tell it to send errors to a file with -e. It is useful to use -D to explicity set the working directory - this will eliminate having to use full paths in the actual job execution code below. -J simply gives the job a name but has no bearing on the jobs behavior. We can give maximum time our job will run with -t. This tells the scheduler that if our job will finish sometime before that, and if it's still running it can stop it. The -p option tells the scheduler where to run your job. In this example, our job will run longer than 2 days so we will specify the 'long' partition. The option -n specifies that we will need 4 cpu's (our application does something that will utilize 4 cpus). **Each node has a cpu limit of 48 unless your application is using MPI or another parallel platform.** Always use less cpu's if you want your job scheduled more quickly. The last option is required if you want your job to run as fast as possible. The default memory allocation for any submitted job is only 50 megabytes. This will almost always be too small. You must specify how much memory you think your job will need using the –mem-per-cpu option. This number must be provided in megabytes using the 'M' at the end. In this example we are allocating 1000 megabytes, or 1 gig per cpu.

IMPORTANT: If you set your job's intended requirements, especially -t (time), you will have a greater chance for getting your job queued and completed quickly. If you do not set parameters, your job will be set equal to others who have also not set these parameters, thus resulting in a "First In, First Out" linear scheduling which can result in long wait periods. Please set your job's time expectations.

Other directives for multi-core jobs and multi-node jobs are:

```
#!/bin/bash
#SBATCH -N 6
#SBATCH --ntasks=8
module load R/R-3.4.4-mkl
R CMD BATCH --no-save myjob.R
    myjob.out
```

-N tells the scheduler to use 6 compute nodes (individual machines) which is only used for MPI or similar parallel jobs. –ntasks=8 sets 8 cores on each node AND will run the application one time on each of those cpus. Do not confuse –ntasks with –cpus-per-task. Refer to the tutorials on the department website for further details. `https://www.stat.wisc.edu/services/`

hpc-cluster1/users-guide-tutorials

In most cases you'll only be concerned with –cpus-per-task unless you are invoking true parallel tasks with MPI or multithreading.

NOTE: The #SBATCH directives tell the scheduler what resources we require. It is up to your code to actually use the allotted resources. If you do not allocate the resources with #SBATCH for what your job requires, it will most likely fail or run slowly.

Please spend time reading the man page for *sbatch*. It is also available online at https://slurm.schedmd.com/sbatch.html

Also, our friends at UC Berkeley are using a similar system. Chris Paciorek has useful information regarding SLURM and MPI, among other things found at http://statistics.berkeley.edu/computing/servers/cluster-high#how-to-submit-single. Some of this information will translate to our system. Thanks to Chris Paciorek for assisting us on our SLURM implementation.

ii.9 Tutorials and Examples

More detailed examples of SLURM submit scripts can be found on the department website in the online Users's Guide. http://www.stat.wisc.edu/services/hpc-cluster1/users-guide-tutorials. A few example are provided here for convenience.

Run an interactive R job on a node using 2 gigs of memory:
Never run computation on the submit nodes. Instead, you can run programs on the nodes of the cluster and work interactively in order to test or debug your work. At the command line use the *srun* command to launch R on a cluster node interactively.

```
srun --pty --mem-per-cpu=2000M /
    workspace/software/bin/R
```

Submit a single R job to the default "debug" partition with default time and memory allocations:

```
#!/bin/bash
#SBATCH --mail-user=user@stat.wisc
    .edu
#SBATCH --mail-type=ALL
module load R/R-3.4.3
R CMD BATCH --no-save test.R
    output.Rout
```

The example above will submit to the "debug" partition since we did not specify #SBATCH -p and will have a hard time limit of two hours. The one CPU that is allocated will have only 50 megabytes of memory to work with since we did not specify *–mem-per-cpu*.

Submit a single R job and specify your own R library for packages:

```
#!/bin/bash
#SBATCH --mail-user=user@stat.wisc
    .edu
#SBATCH --mail-type=ALL
#SBATCH --mem-per-cpu=800M
module load R/R-3.4.4-mkl
export R_LIBS=/workspace/user/
    myRlibrary
R CMD BATCH --no-save test.R
    output.Rout
```

The above example sets the bash environment variable for a specific R library location using the export command. You can set any environment variables you wish in your submit scripts. Order matters, and sometimes the module load line will set various environment variables for you. Be careful as to how you set your environment variable order as to not overwrite something you are expecting to be already set. The job is set to run with 800 MB of memory. This example is an alternative to setting your R library path using .libPaths() in your R code as mentioned in the previous section of this manual.

Submit a single R job that is programmed to use 8 threads:

```
#!/bin/bash
#SBATCH --mail-user=u@stat.wisc.edu
#SBATCH --mail-type=ALL
#SBATCH -p short
```

```
#SBATCH -t 24:00:00 # 24 hours
#SBATCH --cpus-per-task=8
#SBATCH -n 1
#SBATCH --mem-per-cpu=1000M
module load R/R-3.4.3
R CMD BATCH --no-save test.R
```

The example above allocates 1 CPU for the task with -n 1, but assigns 8 total CPUs with –cpus-per-task=8 so that each thread can run on its own CPU. If we did not set –cpus-per-task=8 then all threads would attempt to run on 1 allocated CPU, resulting in an 800% usage of the one processor - which would be slow. Always remember to set –mem-per-cpu.

Launching an array of R jobs with unique inputs:

```
#!/bin/bash
#SBATCH --mail-user=u@stat.wisc.
    edu
#SBATCH --mail-type=ALL
#SBATCH -p short
#SBATCH -t 2-00:00:00
#SBATCH --array=1-250
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=900M
module load R/R-3.4.3-mkl
R CMD BATCH --no-save test.R
    myData.$SLURM_ARRAY_TASK_ID
```

Job arrays allow you to submit many iterations of the same job. It is used instead of typing sbatch submit.sh 100 times in the cases in which you would want to do that. Instead, a job array allows you to submit with sbatch one time but will in turn launch many jobs into the queue. **There are ways to specify unique input files for each job in the array using BASH scripting and environment variables.**

This simple case has test.R looking for a data file in the same directory with names like myData.1 myData.2 myData.3 myData.4 and so on up until the end of the array range, in this case 250. We can use the bash variable $SLURM_ARRAY_TASK_ID set by SLURM when using an arrays in order to increment through the file names. This requires you to name your files using sequential numbers. This is the simplest scenario involving multiple data files.

Remember to visit the online User's Guide for more complete examples.

iii. R Library Paths

R is the most common application used on the cluster. In many cases you will need to be concerned about what path R is using to look for your custom installed packages.

First, check to see what path R is currently using by starting R and running the *.libPaths()* function. This will output a list, in order, of the locations where R will look when loading a library. Different installations of R may default to different library paths for packages. You will want to first, load R using the 'modules' command so that the system sets its paths first as they will exist when you run your HPC job.

```
[user@lunchbox] (20)$ which R
/usr/bin/R
[user@lunchbox] (21)$ module load
    R/R-3.5.0
[user@lunchbox] (22)$ which R
/workspace/software/R-3.5.0/bin/R
[user@lunchbox] (23)$ R
 > .libPaths()
[1] "/workspace/software/R-3.5.0/
    lib/R/library"
```

The above example shows that after we load R-3.5.0 by using the 'module' command, it sets the default system R library path to be *"/workspace/software/R-3.5.0/lib/R/library"*. This will be the **first** path checked for libraries when using the *library()* or *require()* functions in R. Only system administrators can install packages to this location. You can request a package be installed here by emailing lab@stat.wisc.edu.

If you would rather have your personal R library location checked first, you can update this list of paths using the *.libPaths()* function. Inserting the following line at the beginning of your .R code file can achieve this. For example:

```
.libPaths(c("/workspace/user/R/
    library"))
```

Now we can run *.libPaths()* to see its output and our personal R library path should be listed **first**.

```
> .libPaths()
[1] "/workspace/user/R/library"
[2] "/workspace/software/R-3.5.0/
    lib/R/library"
```

By far the easiest way to make sure an R package is available for your job is by simply emailing lab@stat.wisc.edu and making the request. A member of the lab will make sure the package is working and available cluster-wide in the relevant default R library for whichever version of R you are using.

iii.1 A case for long running jobs and batch submission

There are many logical reasons for using a batch job scheduler to run most types of computation in Statistics. The most basic is that of logistics when using Linux and active displays. In the scenario where you would like to run a simulation but it takes longer than a full working day, **without** a batch submission to a scheduler you would start your application by double clicking and opening a graphical user interface, or perhaps start it from the command line. You begin your computation and the application starts running.

At this point you need to leave the office, the coffee shop, or your laptop on the kitchen table as you move on for the day. You must be sure that your computer remains on, without falling asleep, and stay connected to the session in order for your job to complete. This is not efficient as lots of things can interrupt your job such as

- Laptop battery dies
- Your operating system freezes
- Wifi connection shuts off
- Power goes out at home
- A remote server reboots

For these reasons alone it is not advised to start long running jobs from an interactive session on any computer. Instead, the batch job submission method offers a way for you to "hand off" your job to a service that will hold onto your request for resources until they are made available - and even if its 12 hours later it will then start your job automatically for you. It will also handle what to do if a computer is restarted - the scheduler will restart your job where it left off. For this reason alone it is always advised to submit your job to the HPC for scheduling in order to be sure your job completes successfully. If it does not it will also provide you with proper logging and error output so that you can troubleshoot and submit again.

iv. Center for High Throughput Computing

Located on campus in the Wisconsin Institute of Discovery (directly across the street from the Department of Statistics) is the Center for High Throughput Computing. As described on their website, the CHTC "supports a variety scalable computing resources and services for UW-affiliated researchers and their collaborators, including high-throughput computing (HTC) and, tightly-couple computations (e.g. message passing interface, or "MPI"), high-memory, and GPUs. CHTC compute systems and personnel are funded by the National Science Foundation, the National Institutes of Health, the Department of Energy (DOE), the National Science Foundation (NSF), the Morgridge Institute for Research, and various grants from the university.

Standard access to CHTC resources are provided to all UW-Madison researchers, free of charge. Even external collaborators with an on-campus sponsor may be given access to resources. We also offer hardware buy-in options for priority access to computing capacity on a case-by-case basis, though standard access is more than sufficient for the vast majority of CHTC users."

We recommend the use of the CHTC when

the Statistics HPC cluster can no longer meet your needs. Some of these situations include but are not limited to

- Data sets larger than several TB needed for single runs
- The need for hundreds or thousands (or even millions) of cores
- Additional support from CHTC staff would be helpful
- Access to other types of software not available in Statistics
- Faster file access

It is possible to submit directly to the CHTC from Statistics using the special submit node *stat-submit-1.stat.wisc.edu*. In order to use the CHTC you can request access at `http://chtc.cs.wisc.edu/form.shtml`. Much more information regarding the CHTC and their services can be found on their website.