

FH Cluster Guide

October 04, 2022

Contents

About this Course	7
Cluster 101	11
1 What is a Cluster?	11
2 Account Setup	13
2.1 Check your HutchNet ID	13
2.2 Connect to a PI Account	13
3 Terminal Setup	15
3.1 What is a terminal?	15
3.2 Windows Setup	15
3.3 Mac Setup	17
3.4 Linux Setup	18
4 Logging In	21
4.1 What is SSH?	21
4.2 Windows Login	21
4.3 Mac Login	23
4.4 Linux Login	24

5 Submit Your First Job	25
5.1 Download the Script	25
5.2 Confirm the Download	26
5.3 Inspect the Script	27
5.4 Submit the Script	27
5.5 Check the Output	27
6 File Upload and Download	29
6.1 Download Cyberduck	29
6.2 Create Connection	29
6.3 Download and Edit the Script	31
6.4 Upload the New Script	32
6.5 Run the New Script	33
7 Interactive Session	35
7.1 Starting the session	35
7.2 Running Interactive Commands	37
7.3 Using Pre-installed Software Modules	38
8 Getting Help	41
Send an Email	41
Visit the SciWiki	41
Find Community Support on Slack	42
Assessment	45
9 Cluster 101 Self Test	45
Appendix	49
10 Provide Feedback	49

CONTENTS	5
11 FAQ / Troubleshooting	51
11.1 FAQ	51
11.2 Troubleshooting	52
About the Authors	55
References	59

About this Course

Fred Hutch maintains a high performance computing cluster specifically to support research work that requires more intensive computing than your average desktop computer can handle.

Our goal for this course is to get you running on the Fred Hutch cluster **quickly and efficiently**. It is intended for everyone whether you're completely new to using a cluster, or who may have used a cluster at another institution but are new to Fred Hutch.

The Fred Hutch cluster is supported by a group in IT called Scientific Computing, and this course was developed by the Fred Hutch Data Science Lab in collaboration with them. Please see the author credits for more information.

We hope that the following modules will help you take advantage of the powerful resources the Fred Hutch has to offer!

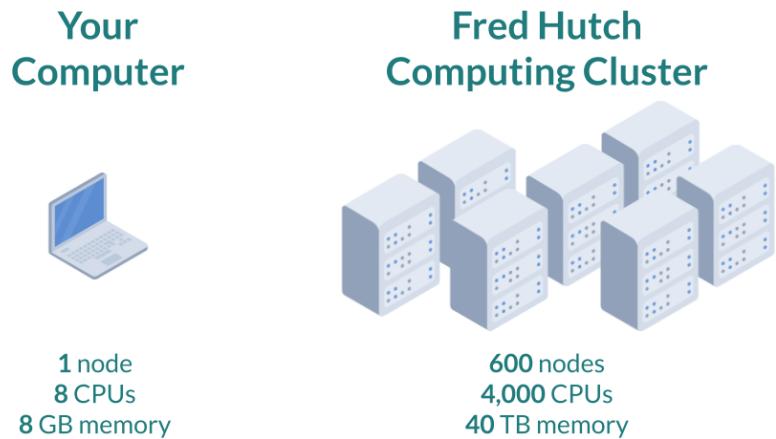
Cluster 101

Chapter 1

What is a Cluster?

A computing cluster is a set of many computers networked together. Because there are many computers working together, the network is able to handle computationally expensive tasks, like genome assemblies or advanced algorithms. Imagine you're building a house. It would take a long time by yourself! It's much better to have many builders working together.

Now that we have a team of workers, the next challenge is task management. A home construction team will need a manager to help delegate tasks. Similarly, the computing cluster uses management software to prioritize tasks, delegate workers (resources), and check on progress. The Fred Hutch cluster uses a common management and scheduling tool called Slurm.



How is the cluster different from a laptop or desktop? First, on your laptop you most likely interact with it using an operating system like Windows or MacOS.

The Fred Hutch cluster uses a Linux operating system. Second, because many people use the cluster for many tasks, there isn't a central screen and keyboard. You access the cluster remotely from your computer! We will talk more about how to connect to the cluster in a following chapter.

Computing cluster

A set of computers networked together to perform large tasks.

Chapter 2

Account Setup

You will need an account to log in to the cluster. This ensures that data stays protected and that computing resources are shared fairly.

2.1 Check your HutchNet ID

Your HutchNet ID is the standard login you receive when you start working at the Hutch or are an official affiliate. You can use it to login to most resources at Fred Hutch (Desktop Computer, Employee Self Service, VPN, Webmail) and our Scientific Computing systems.

For example:

- my email is `jsmith3@fredhutch.org`.
- my HutchNet ID is `jsmith3`.

If one of your collaborators requires access to the Fred Hutch network you can submit a non-employee action form. Non-employee is a generic administrative term for affiliates, students, contractors, etc.

2.2 Connect to a PI Account

Your HutchNet ID must be associated with a PI cluster “account”. The Scientific Computing Team (SciComp) tries to set users up with a connection to a PI account before they need it, but this is not automatic! To ensure that you have been set up to use the cluster, please fill out this Account Setup Form and we will ensure you are set up correctly!

Now, let's connect to the cluster using a terminal!

Chapter 3

Terminal Setup

The next step is getting familiar with your Terminal. This is your portal to the cluster.

3.1 What is a terminal?

The Terminal is a command line interface. In other words, the Terminal is a software application that allows you to issue commands directly to your laptop or desktop computer. The Terminal is very useful because it allows you to run commands that don't have a graphical user interface (GUI). It can also connect you to computer networks, such as the Fred Hutch cluster! The Terminal setup is different depending on your operating system. Jump to the Windows, MacOS, or Linux sections below.

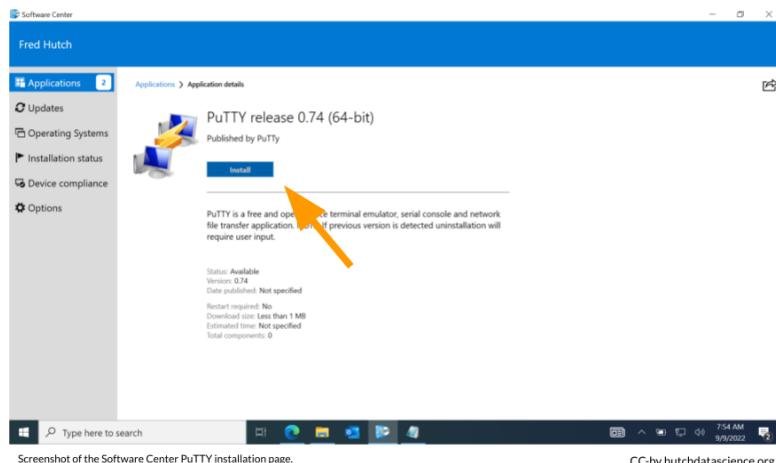
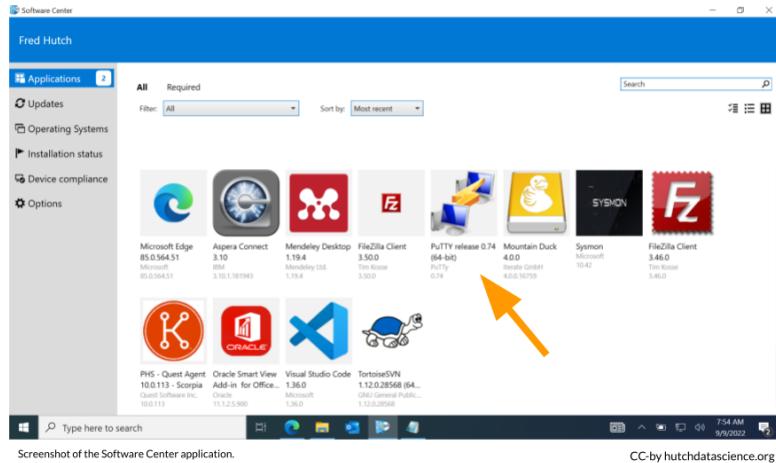
“Terminal” used to be synonymous with “computer”. With the creation of operating systems like Windows and MacOS, computers became much easier to use and exploded in popularity! Your colleagues are almost always referring to the command line application when they say “Terminal”.

3.2 Windows Setup

[Click to view steps](#)

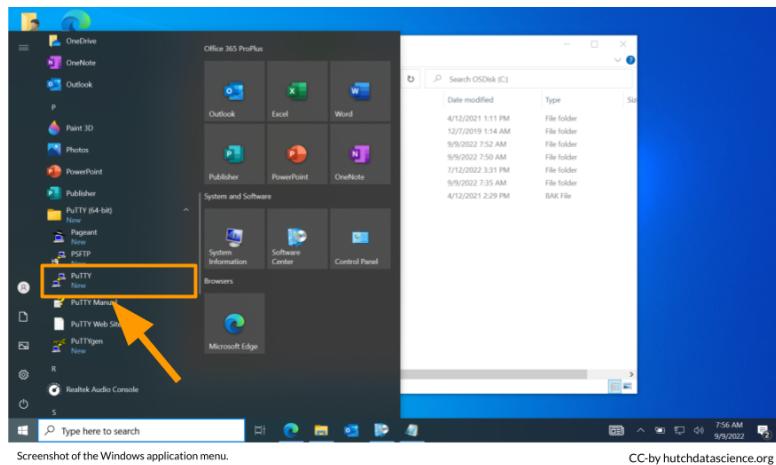
You will need to install a Terminal application called PuTTY to connect to the Fred Hutch Cluster.

1. You should then see PuTTY available in the Software Center. Click “Install” and go through the Setup Wizard.



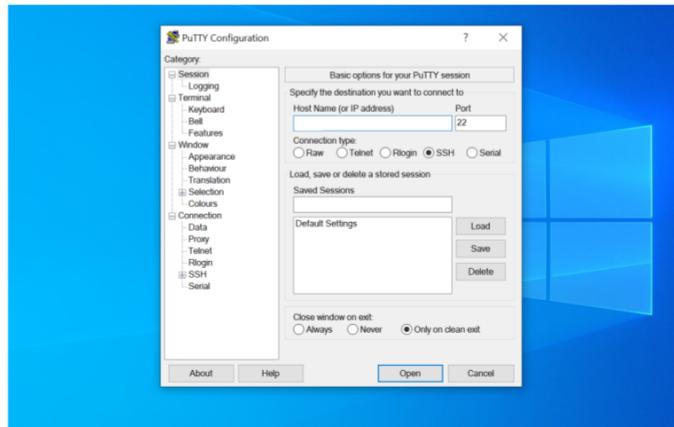
You can also install PuTTY manually if you don't see it in the Software Center.

- PuTTY should now be available in your applications. Click on PuTTY to open.



Screenshot of the Windows application menu.
CC-by hutchdatascience.org

3. You should now see the PuTTY Configuration menu.



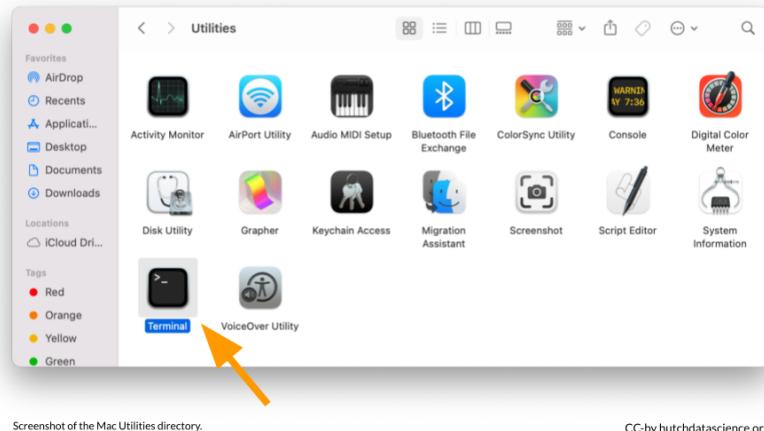
Screenshot of the PuTTY Configuration menu.
CC-by hutchdatascience.org

3.3 Mac Setup

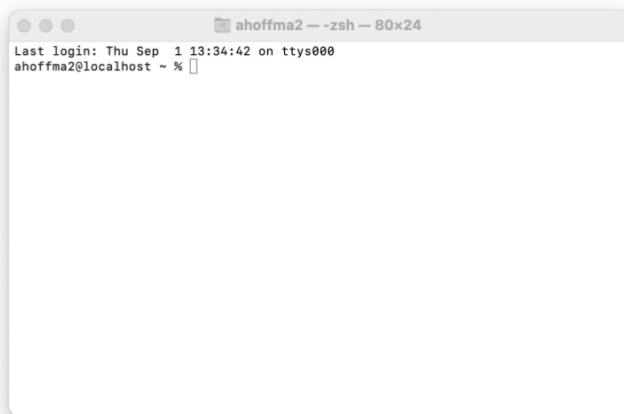
Click to view steps

Mac machines come with a Terminal installed.

1. Go to Finder > Applications > Utilities > Terminal and double-click.



2. Your Terminal should look like this:



3.4 Linux Setup

Click to view steps

The commonly used Linux distribution, Ubuntu, already comes with a Terminal installed.

1. Press **ctrl + alt + T**. Your open Terminal window should look like this:

[SCREENSHOT]

1. Update the Terminal and prepare it for connecting to the cluster by running:

```
sudo apt install openssh-client
```

Enter your password and enter Y when prompted.

Chapter 4

Logging In

Now that you have your Terminal application ready, you want to connect to the cluster. You will do this using a method called SSH, which stands for “Secure SHell”.

4.1 What is SSH?

SSH is a secure way to remotely connect to another computer or network of computers. In other words, SSH helps us protect your data and the data on the Fred Hutch cluster through authentication.

Hostname

The hostname is the name, or label, assigned to a computer in a network. We are connecting to hostname `rhino.fhcrc.org` or `rhino` for short.

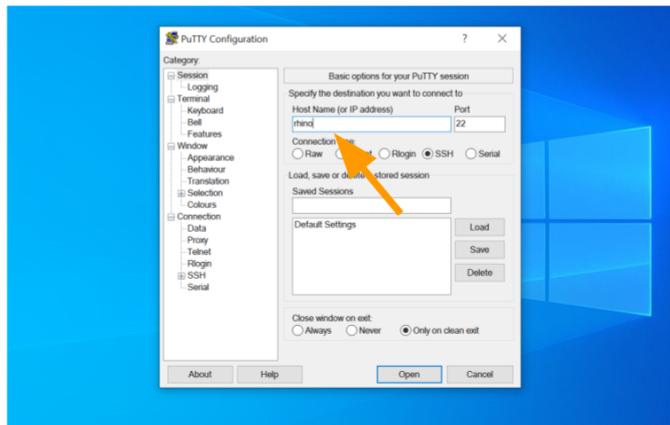
Before moving on, you will need to connect to the Fred Hutch wifi network, a networked ethernet jack, or the Fred Hutch VPN. This is the first layer of security.

The next set of steps are specific to your operating system.

4.2 Windows Login

Click to view steps

1. Go to the PuTTY Configuration menu. Under “Host Name” type `rhino` and click “Open”.



Screenshot of the PuTTY Configuration menu.

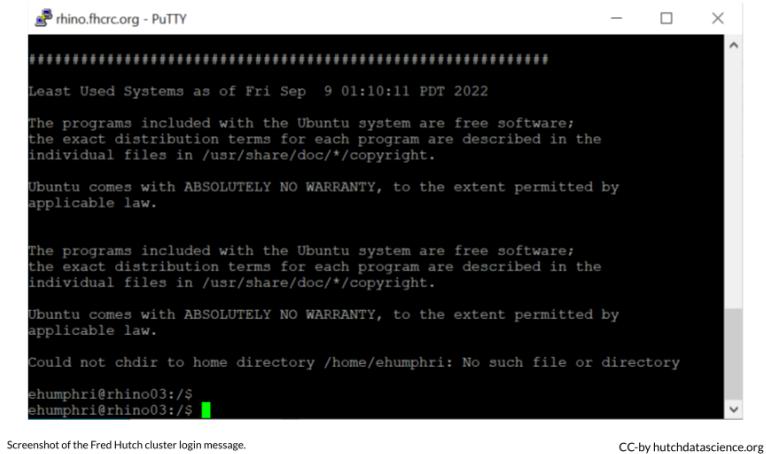
CC-by hutchdatascience.org

2. You will be prompted to login. Type in your HutchNetID (e.g., `jsmith3`).

Screenshot of the PuTTY login prompt for `zhino`.

CC-by hutchdatascience.org

3. Enter your password. No* or symbols will show up, so type it in carefully!
4. You are now logged in! There should be a login message, with your name at the bottom.



Screenshot of the Fred Hutch cluster login message.

CC-by hutchdatascience.org

Congratulations! You are now logged in to the Fred Hutch cluster!

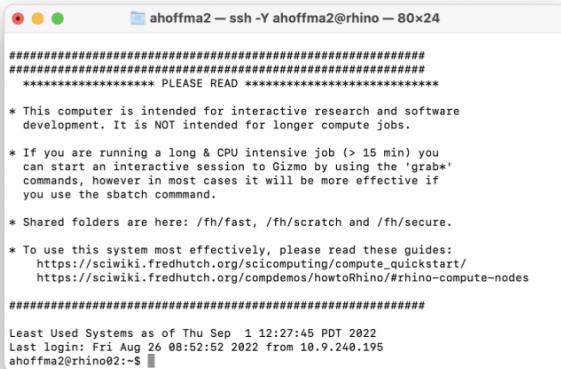
4.3 Mac Login

Click to view steps

1. Type the following commands, substituting in your HutchNet ID:

```
ssh -Y HutchID@rhino
```

2. You will see a message that looks like `The authenticity of host 'rhino (XXX.XXX.XX.XX)' can't be established.`. Type in `yes` and hit enter.
3. Enter your password. No* or symbols will show up, so type it in carefully!
4. You are now logged in! There should be a login message, with your name at the bottom.



```
ahoffma2 - ssh -Y ahoffma2@rhino - 80x24

#####
##### PLEASE READ #####
#####
* This computer is intended for interactive research and software
development. It is NOT intended for longer compute jobs.
* If you are running a long & CPU intensive job (> 15 min) you
can start an interactive session to Gizmo by using the 'grab*' command,
however in most cases it will be more effective if you use the sbatch command.
* Shared folders are here: /fh/fast, /fh/scratch and /fh/secure.
* To use this system most effectively, please read these guides:
  https://sciwiki.fredhutch.org/scicomputing/compute_quickstart/
  https://sciwiki.fredhutch.org/compdemos/howtoRhino/#rhino-compute-nodes
#####

Least Used Systems as of Thu Sep 1 12:27:45 PDT 2022
Last login: Fri Aug 26 08:52:52 2022 from 10.9.240.195
ahoffma2@rhino02:~$
```

Screenshot of the Fred Hutch cluster login message.

CC-by hutchdatascience.org

Congratulations! You are now logged in to the Fred Hutch cluster!

4.4 Linux Login

Click to view steps

Congratulations! You are now logged in to the Fred Hutch cluster!

Chapter 5

Submit Your First Job

The strength of a computing cluster is the ability to do many jobs in parallel or on computers with more computing power than you have on your local computer. The best way to use the cluster is to create short snippets of instructions (a script) that a computer can perform without human input. Your script tells the computers to execute the instructions as individual jobs.

Now that you've logged into rhino you will be able to send scripts to the networked computers in the cluster. The Fred Hutch cluster uses Slurm to organize and prioritize jobs. Based on the instructions in your script, Slurm will find a computer to run your job along with all the other requests from other users.

The part of the cluster where you log in is called `rhino`. The part of the cluster where jobs are run is called `gizmo`.

5.1 Download the Script

We can use the `wget` command to download a script from GitHub. This means we don't have to write the script from scratch. Copy and paste the following into the terminal, and hit return:

```
 wget https://raw.githubusercontent.com/FredHutch/slurm-examples/master/introduction/1-hello-world.sh
```

```
ahoffma2 — ssh -Y ahoffma2@rhino — 80x24
[ahoffma2@rhino03:~$ wget https://raw.githubusercontent.com/FredHutch/slurm-examples/master/introduction/1-hello-world/01.sh
--2022-09-21 10:01:03-- https://raw.githubusercontent.com/FredHutch/slurm-examples/master/introduction/1-hello-world/01.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.1
33, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33 [text/plain]
Saving to: '01.sh'

01.sh          100%[=====]      33  --.-KB/s   in 0s

2022-09-21 10:01:03 (908 KB/s) - '01.sh' saved [33/33]
ahoffma2@rhino03:~$
```

Screenshot of wget command output.

CC-by hutchdatascience.org

5.2 Confirm the Download

Let's confirm that we can see the file we just downloaded. We can use the `ls` (list files) command for this. Type `ls` and hit return. You should see the file `01.sh` in your home directory. The `.sh` ending means this is a script meant to run from the command line.

`ls`

```
ahoffma2 — ssh -Y ahoffma2@rhino — 80x24
[ahoffma2@rhino03:~$ ls
01.sh
ahoffma2@rhino03:~$
```

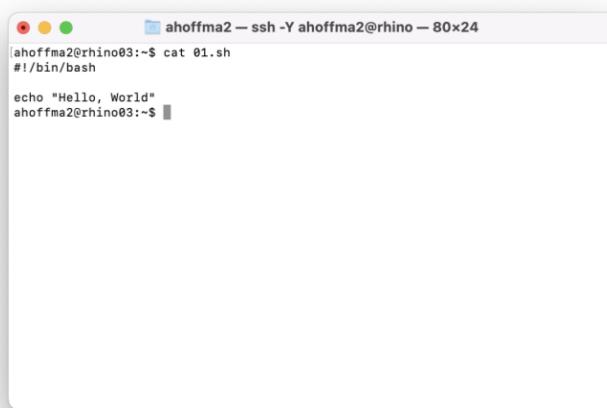
Screenshot of ls command output.

CC-by hutchdatascience.org

5.3 Inspect the Script

Let's next inspect the script. The `cat` command, followed by a file name, lists the entire contents of a specific file.

```
cat 01.sh
```



Screenshot of cat command output, the contents of the script.

CC-by hutchdatascience.org

- The first line of the script, `#!/bin/bash`, indicates that this is a command line or “bash” script.
- The second line is empty, and the third line, `echo "Hello, World"` means that the computer will “echo”, or print out, “Hello, World”.

5.4 Submit the Script

We use the `sbatch` command to submit a script and start running a job on the cluster. Copy the following and hit return. You should see a message like “Submitted batch job 12345678”. Your number will vary because this is a unique job identifier.

```
sbatch 01.sh
```

5.5 Check the Output

Type `ls` again. You should now see a log file like `slurm-12345678.out` listed alongside your script `01.sh`. Let's use `cat` to inspect the output in the log file. We should see our message has been printed!

```
cat slurm-<your-number-here>.out
```

A screenshot of a terminal window titled "ahoffma2 — ssh -Y ahoffma2@rhino — 80x24". The window shows the following command and its output:

```
[ahoffma2@rhino03:~$ ls
01.sh  slurm-65739499.out
ahoffma2@rhino03:~$ cat slurm-65739499.out
Hello, World
ahoffma2@rhino03:~$ ]
```

The terminal has a light gray background and black text. The title bar is blue.

Screenshot of cat command output, the contents of the log file.

CC-by hutchdatascience.org

ls

This command lists the files in the current directory.

cat *filename*

This command prints the contents of a specific file .

sbatch *filename.sh*

This command submits a job to the cluster with instructions specified in .sh

Chapter 6

File Upload and Download

Exchanging files with the cluster is very important. You can imagine scenarios where:

- You want to download log files or output files
- You want to upload a custom .sh script file that you wrote on your laptop
- You want to upload other files

In this course, upload and download of files is performed using Cyberduck. Cyberduck is a tool that lets us connect to the cluster securely, browse files, and transfer files securely.

6.1 Download Cyberduck

Download the latest version of Cyberduck here.

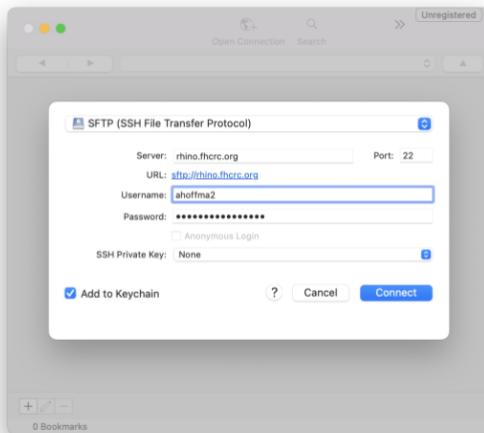
Note that the version of Cyberduck in the Software Center or Self Service might not be current, causing compatibility issues with some operating systems.

6.2 Create Connection

Launch Cyberduck and click on “Open Connection”.

- From the dropdown menu, select “SFTP (SSH File Transfer Protocol)”
- For Server, type “rhino.flcrc.org”
- Fill in your HutchNetID for Username and fill in your password

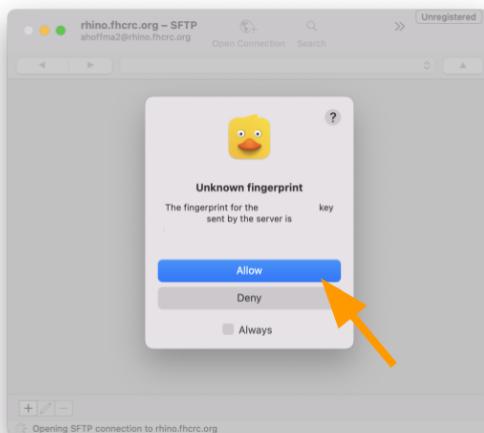
Click “Connect”



Screenshot of the Cyberduck "Open Connection" configuration.

CC-by hutchdatascience.org

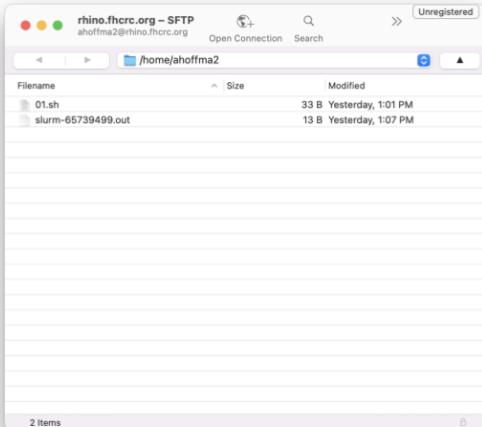
Click “Allow”. You can also check the box to indicate “Always”.



Screenshot of server fingerprint prompt.

CC-by hutchdatascience.org

You should see your script file “01.sh” and the log file.

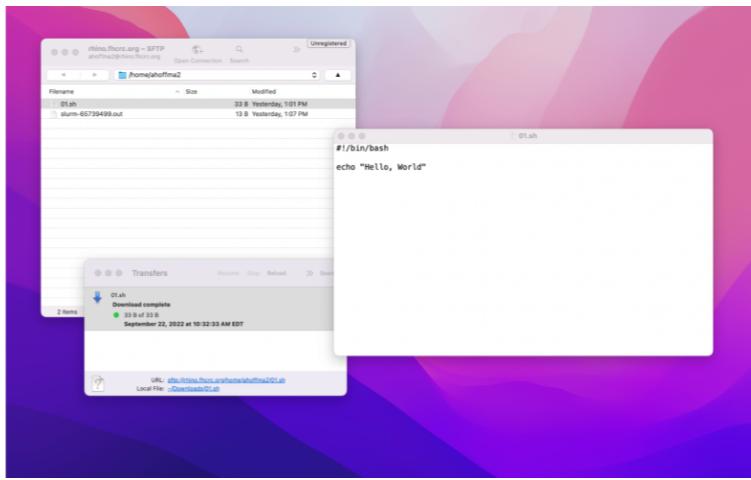


Screenshot of files on the cluster, viewed through Cyberduck.

CC-by hutchdatascience.org

6.3 Download and Edit the Script

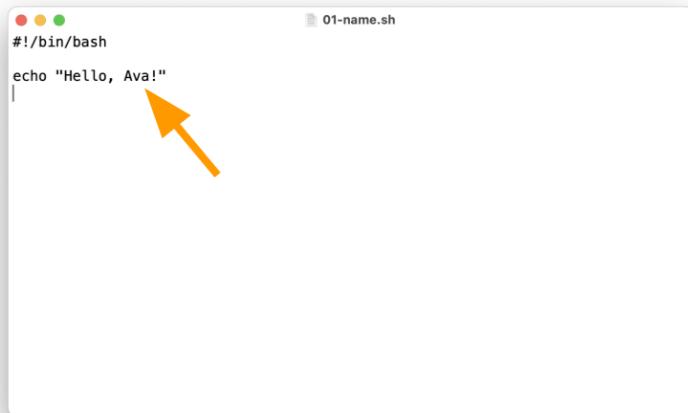
- Right click on “01.sh” and select “Download”
- You will see a “Transfers” prompt open, and the 01.sh file should now appear in your Downloads folder
- Open the 01.sh file



Screenshot of files, Transfers window, and the downloaded script.

CC-by hutchdatascience.org

Edit the message to include your name and save the file. Rename the file **01-name.sh**



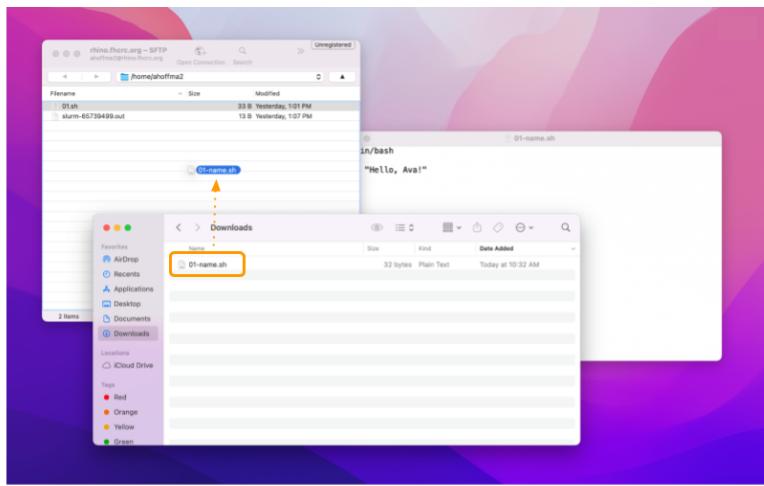
```
#!/bin/bash
echo "Hello, Ava!"
```

Screenshot of the script, edited with a name.

CC-by hutchdatascience.org

6.4 Upload the New Script

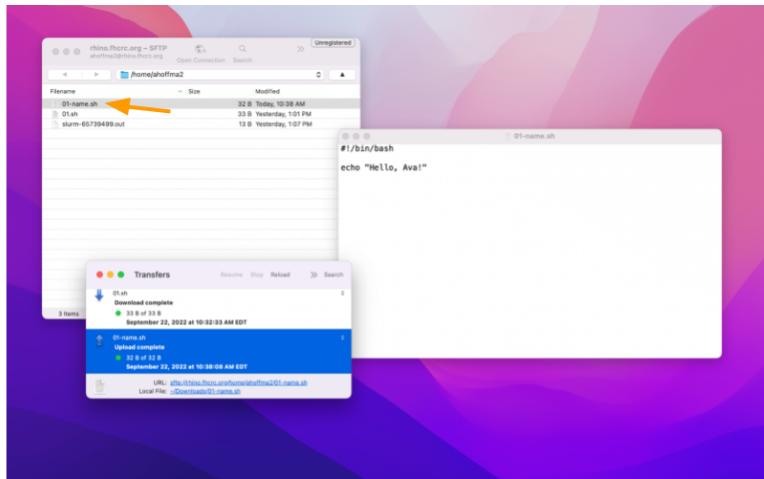
From your Downloads folder, simply drag the file to Cyberduck.



Screenshot of Downloads folder with edited script being dragged over to files via Cyberduck.

CC-by hutchdatascience.org

You should now see the new script among your cluster files.



Screenshot of edited script among cluster files visible on Cyberduck.

CC-by hutchdatascience.org

6.5 Run the New Script

Return to your Terminal. Submit a job with your new script by running the following. When you type `ls` you should see a new log file!

```
sbatch 01-name.sh
```



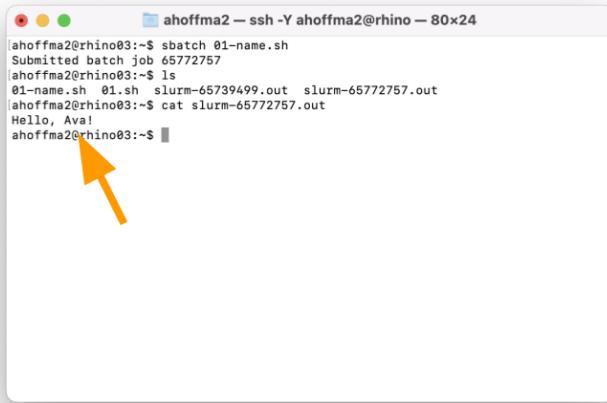
Screenshot showing new log file present on the cluster.

CC-by hutchdatascience.org

The job numbers included in log file names generally increase in number. The greater the number, the more recently the job was run.

Use the `cat` command to inspect the log. The message should show the new text that you added!

```
cat slurm-<your-number-here>.out
```



```
ahoffma2 - ssh -Y ahoffma2@rhino - 80x24
ahoffma2@rhino03:~$ sbatch 01-name.sh
Submitted batch job 65772757
ahoffma2@rhino03:~$ ls
01-name.sh 01.sh slurm-65739499.out slurm-65772757.out
ahoffma2@rhino03:~$ cat slurm-65772757.out
Hello, Ava!
ahoffma2@rhino03:~$
```

Screenshot of edited script message output using 'cat' command.

CC-by hutchdatascience.org

Chapter 7

Interactive Session

While using the cluster, you might need to build and test scripts interactively before running them. Luckily, you can work directly on the cluster by creating an interactive session. When you launch an interactive session, the cluster assigns you a portion of the networked computers called a “node”. This node (or part of one) is dedicated to you for a period of time rather than using the Slurm job submission system. Because an interactive session takes up resources directly on the cluster whether you’re actively using it or not, it’s best to use interactive sessions only when a task cannot be done by submitting a script.

7.1 Starting the session

When starting an interactive session, you’re going to need to think about what you are testing and what resources you might need on the node you are requesting to use. You can always start an interactive session using the default values if you aren’t sure what you need yet.

Start an interactive session on a node by running the command:

```
grabnode
```

You will be prompted with several questions about the type of resources on the node you want. We don’t need anything fancy, so we will set up the session to use minimal resources. You can enter the following:

- *How many CPUs/cores would you like to grab on the node?* **1**
- *How much memory (GB) would you like to grab?* **20**
- *Please enter the max number of days you would like to grab this node:* **1**
- *Do you need a GPU?* **N**

- When prompted, enter your password

The CPU, or Central Processing Unit, is the brain of the computer that performs and orchestrates computational tasks. Modern computers often perform multiple tasks at once, ranging from 4 tasks on a typical laptop to 48 tasks or more on higher end servers.

RAM, or Random Access Memory, is often simply referred to as memory. This short term memory holds the information that the CPU needs to perform calculations. One distinctive feature of memory is that it is short term. In other words, when the electricity is shut off, the data stored in memory disappears. To save the CPU's work, you usually save files to your computer. Running highly complicated analyses or algorithms can often require additional memory resources.

The GPU, or Graphics Processing Unit, is similar to the CPU. The GPU was originally designed to quickly render graphics (such as for video games), but today can be used to run complex artificial intelligence applications or computationally intensive jobs.

CPU



Memory

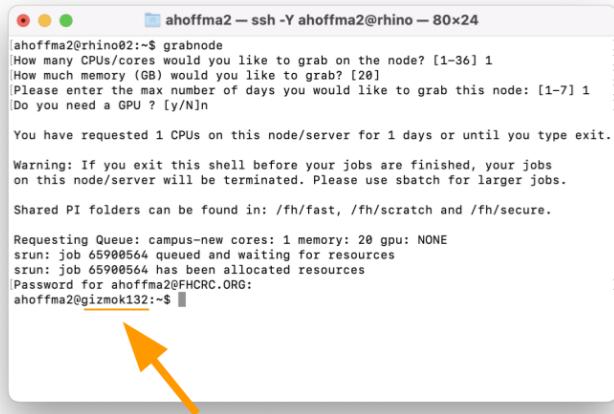


“The Brain”

“Short Term Storage”

CC-by hutchdatascience.org

You will see that you are now logged on to “gizmo” instead of “rhino”. Remember that the part of the cluster where you log in is called `rhino`. The part of the cluster where jobs are run is called `gizmo`.



```
ahoffma2@rhino02:~$ grabnode
How many CPUs/cores would you like to grab on the node? [1-36] 1
How much memory (GB) would you like to grab? [28]
Please enter the max number of days you would like to grab this node: [1-7] 1
Do you need a GPU ? [y/N]

You have requested 1 CPUs on this node/server for 1 days or until you type exit.

Warning: If you exit this shell before your jobs are finished, your jobs
on this node/server will be terminated. Please use sbatch for larger jobs.

Shared PI folders can be found in: /fh/fast, /fh/scratch and /fh/secure.

Requesting Queue: campus-new cores: 1 memory: 20 gpu: NONE
srun: job 65900564 queued and waiting for resources
srun: job 65900564 has been allocated resources
Password for ahoffma2@FHRCR.CORG:
ahoffma2@gizmok132:~$
```

Screenshot of configuration prompts for an interactive node.

CC-by hutchdatascience.org

7.2 Running Interactive Commands

You can start working on the node by running a similar command as we used in the job we submitted via script. Echo a message by running:

```
echo "Hello, again!"
```



```
ahoffma2@rhino02:~$ ssh -Y ahoffma2@rhino - 80x24
ahoffma2@gizmok132:~$ echo "Hello, again!"
Hello, again!
ahoffma2@gizmok132:~$
```

Screenshot of an echo message on an interactive node.

CC-by hutchdatascience.org

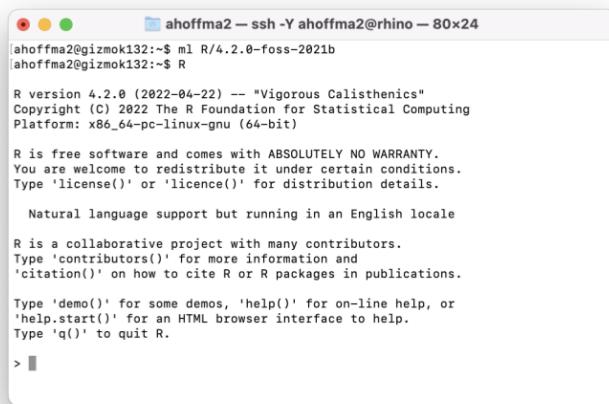
7.3 Using Pre-installed Software Modules

Let's get a bit more advanced. We can load a preconfigured software bundle called a module. This is very convenient because it means we don't need to install anything manually! In this example, we will load a module containing R version 4.2.0. You can learn more about what modules are available and how to request new ones for the Fred Hutch cluster here.

```
ml R/4.2.0-foss-2021b
```

Next, launch R:

```
R
```



The screenshot shows a terminal window titled "ahoffma2 — ssh -Y ahoffma2@rhino — 80x24". The window displays the R 4.2.0 welcome message:

```
ahoffma2@gizmok132:~$ ml R/4.2.0-foss-2021b
ahoffma2@gizmok132:~$ R

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 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.

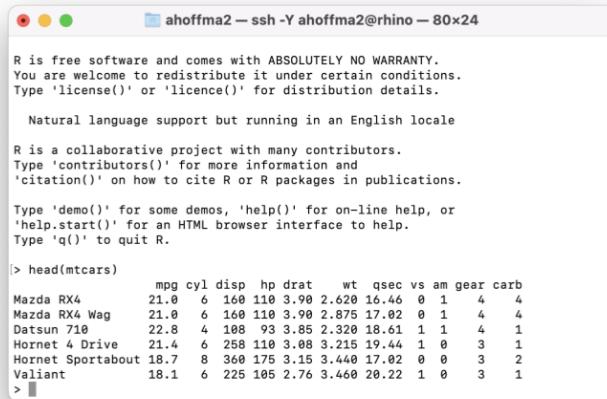
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> 
```

Screenshot of R session launched on an interactive node.

CC-by hutchdatascience.org

You can play around with R here. For example, you might run:

```
head(mtcars)
```



```
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.

> head(mtcars)
      mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4     21.0   6 160 110 3.98 2.620 16.46 0  1   4   4
Mazda RX4 Wag 21.0   6 160 110 3.98 2.875 17.02 0  1   4   4
Datsun 710    22.8   4 108  93 3.85 2.320 18.61 1  1   4   1
Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44 1  0   3   1
Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02 0  0   3   2
Valiant      18.1   6 225 105 2.76 3.460 20.22 1  0   3   1
```

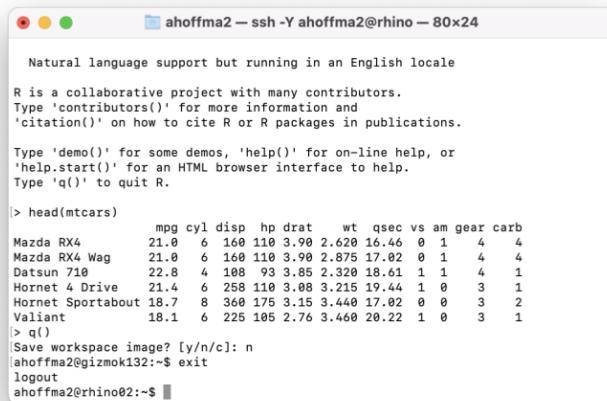
Screenshot of R commands on an interactive node.

CC-by hutchdatascience.org

Close the R session by typing:

`q()`

Close the interactive node by typing:

`exit`


```
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.

> head(mtcars)
      mpg cyl disp hp drat wt qsec vs am gear carb
Mazda RX4     21.0   6 160 110 3.98 2.620 16.46 0  1   4   4
Mazda RX4 Wag 21.0   6 160 110 3.98 2.875 17.02 0  1   4   4
Datsun 710    22.8   4 108  93 3.85 2.320 18.61 1  1   4   1
Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44 1  0   3   1
Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02 0  0   3   2
Valiant      18.1   6 225 105 2.76 3.460 20.22 1  0   3   1
> q()
Save workspace image? [y/n/c]: n
ahoffma2@gizmok132:~$ exit
logout
ahoffma2@rhino02:~$
```

Screenshot exiting the interactive node.

CC-by hutchdatascience.org

`grabnode`

This command starts an interactive session on the cluster.

Node One of the networked computers in the cluster.

CPU

A computer component that performs and orchestrates computational tasks.

Memory

A computer component that stores calculations and information in the short term.

Chapter 8

Getting Help

The Scientific Computing group in IT manages the cluster, provides support with software, advises on data storage, and holds office hours specifically to help users. Here are some ways you can get help for your work on the cluster.

Send an Email

The primary way you can request help for a problem is to send SciComp an email, so a ticket will be created in their tracking system. This allows the details of the problem you're having to be sent to them so they can better help you. Submitting a good email ticket helps the SciComp Team address your needs quickly and efficiently. We suggest you submit the following information:

1. A brief overview of what the problem is.
2. Some specifics about the problem, such as the full text (it's ok if it's long) of any error message or terminal command, or a screen shot of the interface you were using when you had the problem.
3. A description of what you wanted to have happen or what your overall goal is (in case perhaps there is another strategy that might work better).

Visit the SciWiki

The SciWiki Scientific Computing page is full of useful tips and guides. Remember when using the search that the login “nodes” to the Fred Hutch cluster are called rhino and the cluster “nodes” are called gizmo.

Find Community Support on Slack

Peer-to-peer support can be very valuable in learning and troubleshooting your work. The Fred Hutch Data Slack workspace is open to all with a fredhutch.org, uw.edu, seattlechildrens.org, or related institution email addresses (whi.org, scharp.org, etc). You can ask questions, find out about office hours, and discover other live support and training events that can help you learn more about how to leverage resources at Fred Hutch to advance your science.

Assessment

Chapter 9

Cluster 101 Self Test

Appendix

Chapter 10

Provide Feedback

Please submit an issue about this course at our GitHub repository. You can also click the edit button on the top of the page in question.

Chapter 11

FAQ / Troubleshooting

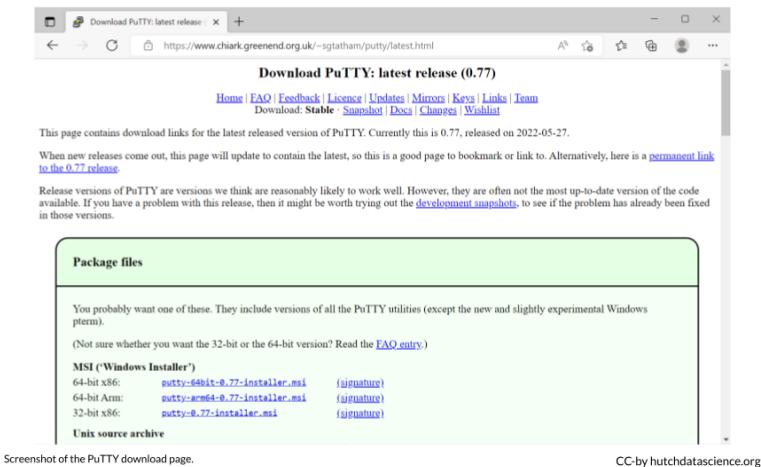
11.1 FAQ

Here are some questions you might have.

11.1.1 How can I manually install PuTTY?

Click to view steps

1. Click here to install the latest version of PuTTY. You will choose the 64-bit x86 installation with few exceptions.



2. Click through to install via the Setup Wizard.

11.2 Troubleshooting

Here are some issues you might encounter.

```
ssh: Could not resolve hostname rhino: nodename nor servname provided, or not known
```

[Click to view steps](#)

This error means that your computer is having trouble connecting to rhino. Ensure one of the following is true:

1. You are connected to the Fred Hutch wifi network on campus.
2. You are connected to the Fred Hutch VPN
3. You are plugged into an ethernet cable on campus that taps into the Fred Hutch network. Note that not all ethernet wall jacks have this capability, so try another jack if you are having trouble. Please email the IT helpdesk and include your office number and the number on the jack if you find a jack that isn't working.

```
ssh: connect to host rhino port 22: Undefined error: 0
```

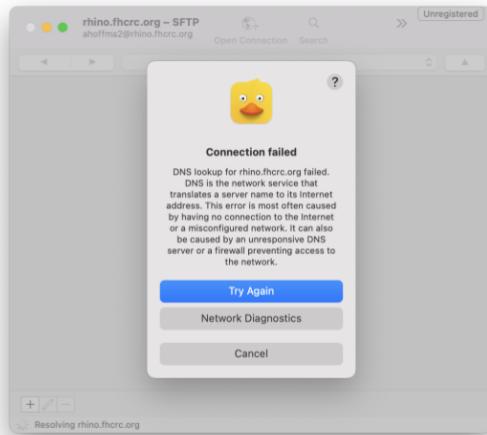
[Click to view steps](#)

This likely indicates a disruption to your internet connection and/or VPN. Ensure you are connected to the internet and connected to the Fred Hutch network on campus or the VPN.

Connection failed message in Cyberduck

[Click to view steps](#)

This likely indicates a disruption to your internet connection and/or VPN. Ensure you are connected to the internet and connected to the Fred Hutch network on campus or the VPN.



Screenshot of the Cyberduck "Connection failed" message.

CC-by hutchdatascience.org

Invalid account or account/partition when logging in

Errors similar to this typically indicate that the account hasn't been set up by SciComp. This is a quick fix if you use the form mentioned in the course.

About the Authors

These credits are based on our course contributors table guidelines.

Credits	Names
Pedagogy	
Lead Content Instructor(s)	FirstName LastName
Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved	Delivered the course in some way - video or audio
Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved	If any other authors besides lead instructor
Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved	Wrote less than a chapter
Content Editor(s)/Reviewer(s)	Checked your content
Content Director(s)	Helped guide the content direction
Content Consultants (include chapter name/link in parentheses or word "General") - make new line if more than one chapter involved	Gave high level advice on content
Acknowledgments	Gave small assistance to content but not to the level of consulting
Production	
Content Publisher(s)	Helped with publishing platform
Content Publishing Reviewer(s)	Reviewed overall content and aesthetics on publishing platform

Credits	Names
Technical	
Course Publishing Engineer(s)	Helped with the code for the technical aspects related to the specific course generation
Template Publishing Engineers	Candace Savonen, Carrie Wright
Publishing Maintenance Engineer	Candace Savonen
Technical Publishing Stylists	Carrie Wright, Candace Savonen
Package Developers (otrpal)	Candace Savonen, John Muschelli, Carrie Wright
Art and Design	
Illustrator(s)	Created graphics for the course
Figure Artist(s)	Created figures/plots for course
Videographer(s)	Filmed videos
Videography Editor(s)	Edited film
Audiographer(s)	Recorded audio
Audiography Editor(s)	Edited audio recordings
Funding	
Funder(s)	Institution/individual who funded course including grant number
Funding Staff	Staff members who help with funding

```
## - Session info -----
## setting value
## version R version 4.0.2 (2020-06-22)
## os      Ubuntu 20.04.3 LTS
## system x86_64, linux-gnu
## ui      X11
## language (EN)
## collate en_US.UTF-8
## ctype   en_US.UTF-8
## tz      Etc/UTC
## date    2022-10-04
##
## - Packages -----
## package     * version    date      lib source
## assertthat    0.2.1     2019-03-21 [1] RSPM (R 4.0.3)
## bookdown     0.24      2022-02-15 [1] Github (rstudio/bookdown@88bc4ea)
```

```
## callr      3.4.4    2020-09-07 [1] RSPM (R 4.0.2)
## cli        2.0.2    2020-02-28 [1] RSPM (R 4.0.0)
## crayon     1.3.4    2017-09-16 [1] RSPM (R 4.0.0)
## desc       1.2.0    2018-05-01 [1] RSPM (R 4.0.3)
## devtools   2.3.2    2020-09-18 [1] RSPM (R 4.0.3)
## digest     0.6.25   2020-02-23 [1] RSPM (R 4.0.0)
## ellipsis   0.3.1    2020-05-15 [1] RSPM (R 4.0.3)
## evaluate   0.14     2019-05-28 [1] RSPM (R 4.0.3)
## fansi      0.4.1    2020-01-08 [1] RSPM (R 4.0.0)
## fs         1.5.0    2020-07-31 [1] RSPM (R 4.0.3)
## glue       1.6.1    2022-01-22 [1] CRAN (R 4.0.2)
## htmltools  0.5.0    2020-06-16 [1] RSPM (R 4.0.1)
## knitr      1.33     2022-02-15 [1] Github (yihui/knitr@a1052d1)
## lifecycle  1.0.0    2021-02-15 [1] CRAN (R 4.0.2)
## magrittr   2.0.2    2022-01-26 [1] CRAN (R 4.0.2)
## memoise    1.1.0    2017-04-21 [1] RSPM (R 4.0.0)
## pkgbuild   1.1.0    2020-07-13 [1] RSPM (R 4.0.2)
## pkgload    1.1.0    2020-05-29 [1] RSPM (R 4.0.3)
## prettyunits 1.1.1    2020-01-24 [1] RSPM (R 4.0.3)
## processx   3.4.4    2020-09-03 [1] RSPM (R 4.0.2)
## ps          1.3.4    2020-08-11 [1] RSPM (R 4.0.2)
## purrr      0.3.4    2020-04-17 [1] RSPM (R 4.0.3)
## R6          2.4.1    2019-11-12 [1] RSPM (R 4.0.0)
## remotes    2.2.0    2020-07-21 [1] RSPM (R 4.0.3)
## rlang       0.4.10   2022-02-15 [1] Github (r-lib/rlang@f0c9be5)
## rmarkdown   2.10     2022-02-15 [1] Github (rstudio/rmarkdown@02d3c25)
## rprojroot   2.0.2    2020-11-15 [1] CRAN (R 4.0.2)
## sessioninfo 1.1.1    2018-11-05 [1] RSPM (R 4.0.3)
## stringi     1.5.3    2020-09-09 [1] RSPM (R 4.0.3)
## stringr     1.4.0    2019-02-10 [1] RSPM (R 4.0.3)
## testthat    3.0.1    2022-02-15 [1] Github (R-lib/testthat@e99155a)
## usethis    2.1.5.9000 2022-02-15 [1] Github (r-lib/usethis@57b109a)
## withr      2.3.0    2020-09-22 [1] RSPM (R 4.0.2)
## xfun        0.32     2022-08-10 [1] CRAN (R 4.0.2)
## yaml       2.2.1    2020-02-01 [1] RSPM (R 4.0.3)
##
## [1] /usr/local/lib/R/site-library
## [2] /usr/local/lib/R/library
```


References