

**3rd Annual Econometrics Game**

**Connecting to Research Computing Center Resources**

**- part 2 -**

**Information about accessing RCC resources**

Multiple members of the same team can connect to the RCC cluster and access computing resources simultaneously so long as the resources are available.

The typical workflow is:

1. Login to one of the two “login nodes” (e.g., via SSH);
2. Once you are in the login node, you can perform simple tasks (e.g., view and copy files);
3. Reserve a “compute node” on the cluster using the sinteractive command.

We assume that you are already familiar with parts 1. and 2. We will cover in this documentation sheet the only the part 3, the connection to a compute node and running jobs in this reserved node.

* Note that **‘XXXX’** represents the **code of the team** (**rccguestXXXX**).

# 

# **Accessing the dataset**

The dataset for the game is located at:

/project/economgame/data/



# **Connecting to the compute node**

To connect to the compute node reserved for your team, once you are in the login node (see part 1 of the documentation).

**Each team** has associated one compute node in the broadwl-lc partition. The compute node corresponding to this partition contains 28 CPUs and 64GB of memory. You can specify the time you want the job to be available.

We suggest you to run tmux or screen command prior connecting to the compute node. They allow using multiple shell windows from a single SSH session and keep it active even through network disruptions. More information about the basic commands: [tmux](http://man.openbsd.org/OpenBSD-current/man1/tmux.1) and [screen](https://www.gnu.org/software/screen/manual/screen.html#Commands).

sinteractive --account=economgame --partition=broadwl-lc --reservation=economgame-rccguestXXXX

you will have access to the default configuration: use of 1 CPU and 2GB of memory for 2 hours.

Here is an example on how to configure the number of CPUs, the memory, and the time:

sinteractive --account=economgame --partition=broadwl-lc --reservation=economgame-rccguestXXXX --time=15:00:00 --ntasks-per-node=2 --mem-per-cpu=4000M

This will give you access to 2 CPUs and 8GB of memory for 15 hours. Note that the reservation is only available until 12 am on Sunday, so if you set --time to a period beyond the deadline of the reservation, you will get an error.



**Running IPython notebook**

* Copy your IPython scripts in the team’s home directory (to access home directory: cd ~)
* Get an interactive session (See Connecting to the compute node section)
* Once you get the interactive session, run the /project/economgame/IPython/run\_ipython.sh script and open the displayed address in your browser (on your computer).
* Note that this script will load Python 2.7 by default (the script needs to be changed if you want to use other version of Python).
* Note: If you want to access IPython through Anaconda, please run run\_ipython.sh script, as:

./run\_ipython.sh Anaconda3

# **Installing an R package**

To install a **system-wide R package**, or a package that requires special compilation steps, please contact Teodora and ask for the required package to be installed. If you are using a specific version of R on Midway, you also need to indicate the version of R that you want this package to be installed for.

Note: The default version of R is 3.3.2 and has many packages already installed (over 200). Please use installed.packages() R function to list all the installed packages.

To install an **R** package locally in the team’s home directory, use the install.packages function in R. Click [here](https://rcc-uchicago.github.io/R-intro/demo_gwas_catalog.html) for more details. The default settings of R should work correctly for installing packages locally. However, if you would prefer installing to a non-default, the simplest approach is to set the R\_LIBS\_USER environment variable to the desired directory pathname in the ~/.Renviron file; for details, see [here](https://stat.ethz.ch/R-manual/R-devel/library/base/html/libPaths.html) or type “help(.libPaths)” in R.



**Installing a Python package**

To install a **system-wide Python package**, please contact Teodora and ask for the required package to be installed.

To install a **Python** package locally in the team’s home directory, please run the following commands:

* module load python/desired-version
* pip install --user missing-python-package
* export $PATH=$HOME/.local/bin:$PATH

The missing-python-package will be installed under your home directory (located at the /home/rccguestXXXX/.local folder). To use the installed package, you always need to load the python/desired-version module first. You could put the export command in the $HOME/.bashrc file to avoid running it every time after logging in.

**Graphical capabilities**

* Note: If you need to access the graphical user interface of a software module, one option is by using trusting X11 forwarding from the Terminal (e.g.,):

ssh -i ~/Downloads/TeamY/Linux\_MAC/id\_rsa.rccguestXXXX -Y rccguestXXXX@midway2.rcc.uchicago.edu

* [XQuartz](https://www.xquartz.org/) is required for Mac computers if users decide to use ssh -Y (trusted X11 forwarding)
* For Windows computer, you can use [MobaXterm](http://mobaxterm.mobatek.net/documentation.html). See **Use private key** of the documentationsection for how to use ssh-key to connect to MobaXterm.
* Note that the X Windows forwarding can be slow for more complex plots; an alternative is to generate an image file (e.g., PDF, PNG) and view the file locally on your laptop.