

# Introduction to Jupyter at HCC

Project Jupyter (<https://jupyter.org>) provides free, open-source software for individuals to write and run computer programs. The name “Jupyter” comes from the three main languages used with it - Julia, Python, and R. Note that this project was originally developed for Python. The corresponding organization that supports the project was founded in 2015.

Most individuals will use the Jupyter Notebook program of the project when writing and running of R code. While this program can be downloaded and installed by users on their own computer, the operating system must be Linux. For this reason, many individuals use Jupyter Notebook via a central Linux-based server through JupyterHub.

Jupyter Notebook provides a web-based browser interface so that a user does not really notice the underlying Linux aspects. This provides a VERY convenient way for using R and other programs, rather than using a command-line, terminal interface (see the HPC notes from my STAT 950 course for more on this alternative use).

The actual coding environment is somewhat different than what is experienced in the most often way RStudio is used. Jupyter Notebook (hereafter just referred to as “Jupyter”) uses a notebook interface that is similar to what is found in mathematical software, like Maple and Mathematica. In this setting, code/output is presented in the following form:

Code

Output

Code

Output

Code

Output

---

...

all within the program editor. Thus, it is like using the R Console window directly, but one can easily re-run previous parts of the code by simply moving the cursor back to that portion of the code. Most statistical software packages instead will keep the code/output in separate windows (although the code may be echoed in the output window).

While it is nice to see the output right after the code, problems can be encountered! In particular, code from earlier portions of the program could be changed and re-run, but later portions of the output would not change.

The corresponding Jupyter notebook for this set of notes is `Examples.ipynb`. Data files used with previous notes are `gpa.txt` and `gpa.csv`.

## *Holland Computing Center*

The easiest way for faculty/students to use Jupyter is through the Holland Computing Center. The HCC is the University of Nebraska system's home to supercomputers. The UNL home to the HCC is in the building under the south side of Memorial Stadium. Walk-in help is available there if you have problems with using their computers.

To use their computers, you need an account as part of a “group” (this account is different from your normal UNL account). Most students obtain an account by using their advisor's group, if their advisor has a group. If a student in our course does not have this type of group available, they can be part of my own “stattools” group (all users within the group will be deleted at the end of the semester). Students can go to <https://hcc.unl.edu/new-user-request> to start the registration process.

Once a student has an account, you can log into it through the

---

Crane or Rhino supercomputers (often referred to as “clusters” because multiple core processors are clustered together to form a supercomputer). For the purposes of using Jupyter, simply use your web browser to go to one of these two addresses:


- <https://crane.unl.edu>
- <https://rhino.unl.edu>

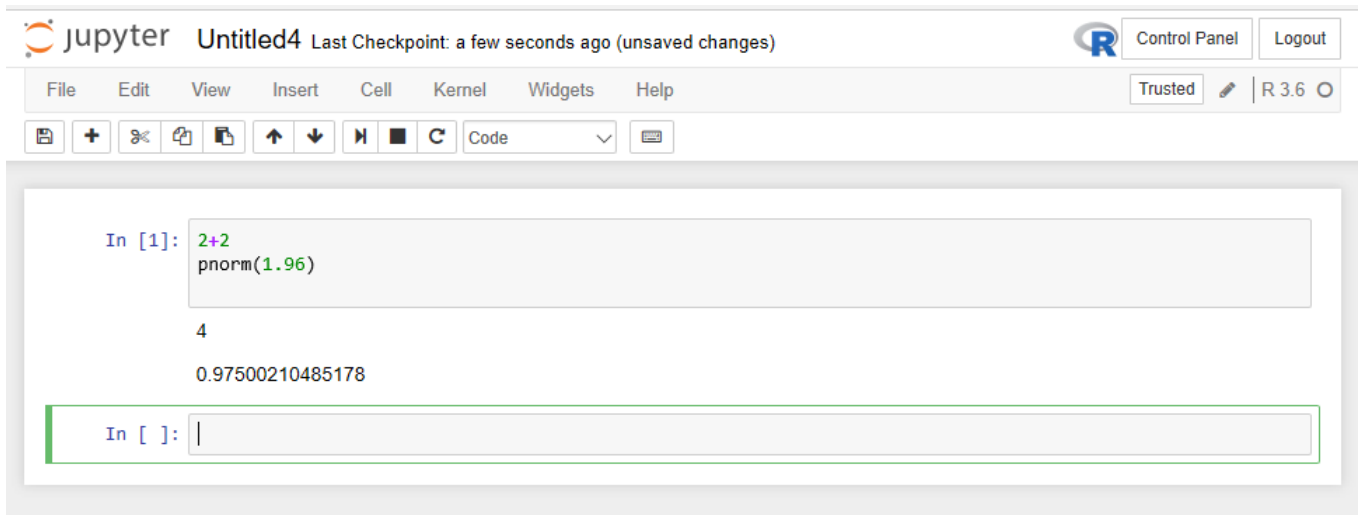
The HCC uses two-factor authentication so you will need to be prepared for the second form, where most people use the Duo app on their smartphones. If you need to use a different second method, please contact HCC directly.

### *Jupyter Notebook*

The HCC has a set of help documents available at [https://hcc.unl.edu/docs/guides/running\\_applications/jupyter](https://hcc.unl.edu/docs/guides/running_applications/jupyter) for using Jupyter. The purpose of this section is to provide a few notes about it.

- After you log in, you will be asked about starting a SLURM job. In the context of running Jupyter in our course, one needs to be concerned with how many core processors you will need. You will only need one core processor for our course (more would be needed for parallel processing). SLURM will be discussed later in these notes. For more detailed information on SLURM beyond these notes, please see my HPC notes for STAT 950.
- A file/folder interface to your account is provided.
- A Jupyter file has a .ipynb extension (Jupyter was originally part of iPython). These are simply plain text files, but they are not as easy to examine in a text editor as a SAS or R program is.

- Select NEW and the R (or SAS) to open a new Jupyter notebook for that language.
- Code goes into “cells” as denoted by the “In [ ]:” notation. To run a set of code, select CELL > RUN CELLS or select . Below is code/output for a simple set of code that has been run:



The screenshot shows the Jupyter Notebook interface. At the top, there is a header with the Jupyter logo, the text "jupyter Untitled4", and a status message "Last Checkpoint: a few seconds ago (unsaved changes)". On the right side of the header, there are buttons for "Control Panel" and "Logout". Below the header is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar, there is a "Trusted" indicator and the version "R 3.6". Below the menu bar is a toolbar with various icons for file operations and execution. The main area of the notebook contains a code cell. The cell is labeled "In [1]:" and contains the following R code:

```
2+2
pnorm(1.96)
```

The output of the code is displayed below the code:

```
4
0.97500210485178
```

Below the output, there is an empty code cell labeled "In [ ]:" with a cursor.

- Jupyter does NOT support dragging and dropping programs on your own computer into a cell. Rather, you can copy/paste the program into a cell. Unfortunately, the results obtained after running the code will be organized in an odd manner because the code does not echo:

```

In [2]: #####
# Section 1

2+2
pnorm(1.96)
(2-3)/6
2^2
sin(pi/2)
log(1)

save<-2+2
save

ls()
objects()

4
0.97500210485178
-0.166666666666667
4
1
0
4
'save'
'save'

```

- The notebook file associated with these set of notes provides a demonstration of where problems can occur when using notebooks.
- Other items
  - Download your R code via FILE > DOWNLOAD
  - Save your notebook with FILE > SAVE AND CHECKPOINT or by selecting the save icon
  - End your session with FILE > CLOSE AND HALT
  - The user interface for the file/folder and Jupyter can be slow
  - Programs are scheduled to be run at HCC via SLURM rather than run immediately (SLURM is a scheduling utility for Linux); thus, programs can take much longer to run depending on whether code can be executed immediately

## *Other notebook interfaces for R*

Select `FILE > NEW FILE > R NOTEBOOK` in RStudio. A new package will be installed with your first time using a notebook. RStudio's notebook format uses the same format as R Markdown. The difference is that code can be executed by selecting the run arrow for a chunk. Below is a small example (no associated file is made available):

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Examples.Rmd

Preview Insert Run

```

1 ---
2 title: "R Notebook"
3 output: html_notebook
4 ---
5
6 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code
7 within the notebook, the results appear beneath the code.
8 Try executing this chunk by clicking the *Run* button within the chunk or by placing
9 your cursor inside it and pressing *Ctrl+Shift+Enter*.
10 ```{r}
11 2+2
12 pnorm(1.96)
13 ```
14
15 ```{r}
16 #####
17 # Section 1
18
19 2+2
20 pnorm(1.96)
21 (2-3)/6
22 2^2
23 sin(pi/2)
24 log(1)
25
26 save<-2+2
27 save
28
29 ls()
30 objects()
31
32 ```

```

[1] 4  
[1] 0.9750021

[1] 4  
[1] 0.9750021  
[1] -0.1666667  
[1] 4  
[1] 1  
[1] 0  
[1] 4  
[1] "save"  
[1] "save"