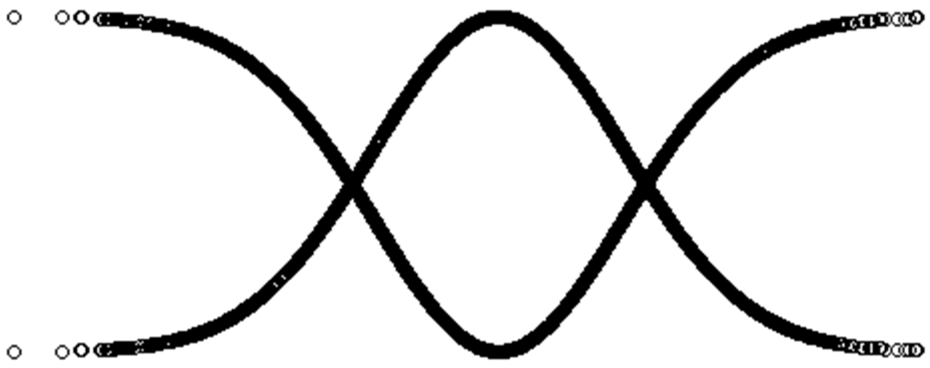
Using the R Software in a High Performance Computing Cluster

by

Carlos García
<cgarci8 AT tigers.lsu.edu>



Louisiana State University and A&M College
Baton Rouge, Louisiana

Last update: 05/22/2014 2:00 PM

Contents

Learning Objectives

- To setup R in the software environment.
- To create a basic .sh file for running R.
- To submit a job that runs a basic R script.
- To make an elementary inspection of the output from a job.
- To import data from a script.
- To install R packages.
- To show how to work with R packages effectively.

Learn more about supercomputers at LSU:
Philip http://www.hpc.lsu.edu/docs/guides.php?system=Philip
Mike http://www.hpc.lsu.edu/resources/hpc/system.php?system=SuperMike-II

In a nutshell

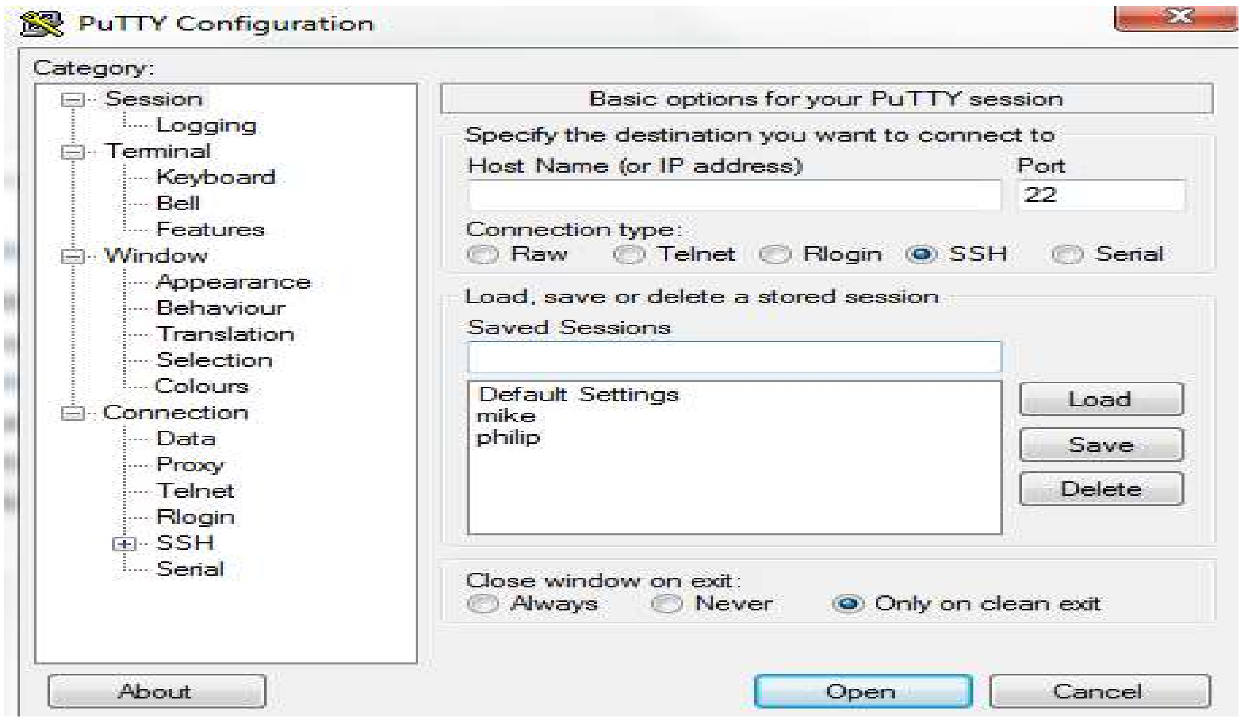
- Log into the hpc cluster:
  - Mike or Philip
- Modify the corresponding ~/.soft file (see details in section: Setting R in Philip).
- Change the present directory to your own work directory
  - $ cd /work/myusername/
- Create folder "rscripts" in your own work directory
  - $ mdkdir rscripts
- Change directory again
  - $ cd /work/myusername/rscripts/
- Upload two files (via terminal or WinSCP)
  - R script: testofr.R
  - Shell script: testmyr.sh
- Edit the shell script, change the path of the directory in the 7th line with your own username. The line is located immediately after the #PBS commands:
  cd /work/myusername/rscripts/
  - There are two ways for making this change:
    - Directly from the terminal: $ nano testmyr.sh
    - Or simply make the change in text editor before uploading the files
  - CAUTION: if you do not make this change, you will get this error message after job submission: `/var/spool/torque/mom_priv/jobs/226838.philip1.SC: line 7: cd: /work/myusername/rscripts/: No such file or directory`
- Submit job
  - $ qsub testmyr.sh
- Inspect the output files
- How to install and work with R packages? See section: Working with R packages.

Setting the R software

Before using any software in the cluster, the software environment has to be modified, by adding the particular software version that is going to be used. Furthermore, the compiler specific to the software installation has to be specified. This initiation is made by editing the .soft file which stores all of the software packages that are linked to the user's account.

Accessing the HPC cluster

Start by logging into your account with the terminal emulator putty (you have the option of saving the session settings for loading them in future connections):



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

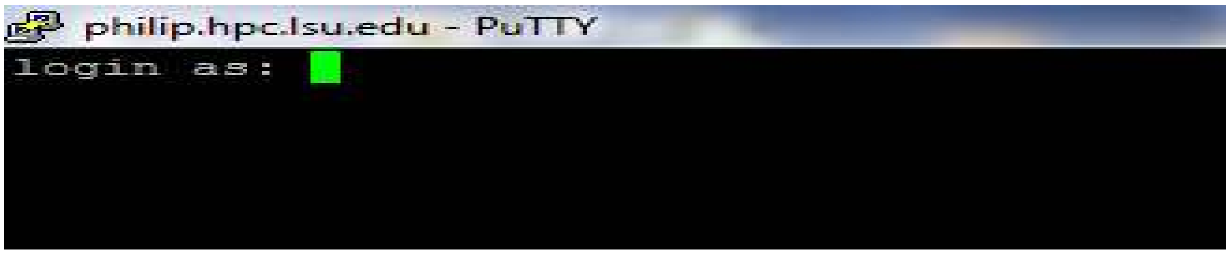
Load, save or delete a stored session

Saved Sessions

Default Settings
mike
philip

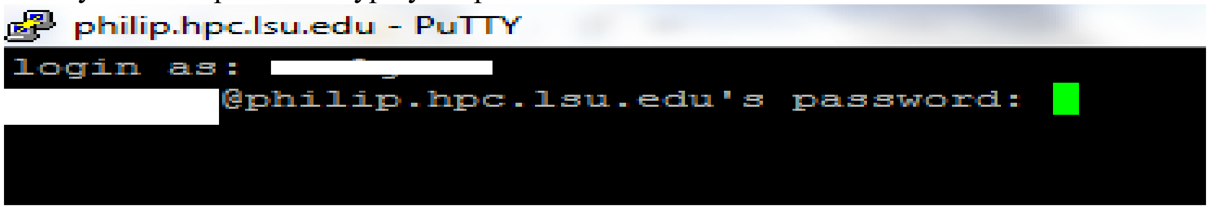
Close window on exit: Always Never [✓ Only on clean exit]

Hostname: philip.hpc.lsu.edu [if you use Mike, use this: mike.hpc.lsu.edu]
Connection type: SSH
Port: 22
Finally, click on OPEN. And the following window will appear, and now you can log in with your username (login as: myusername).



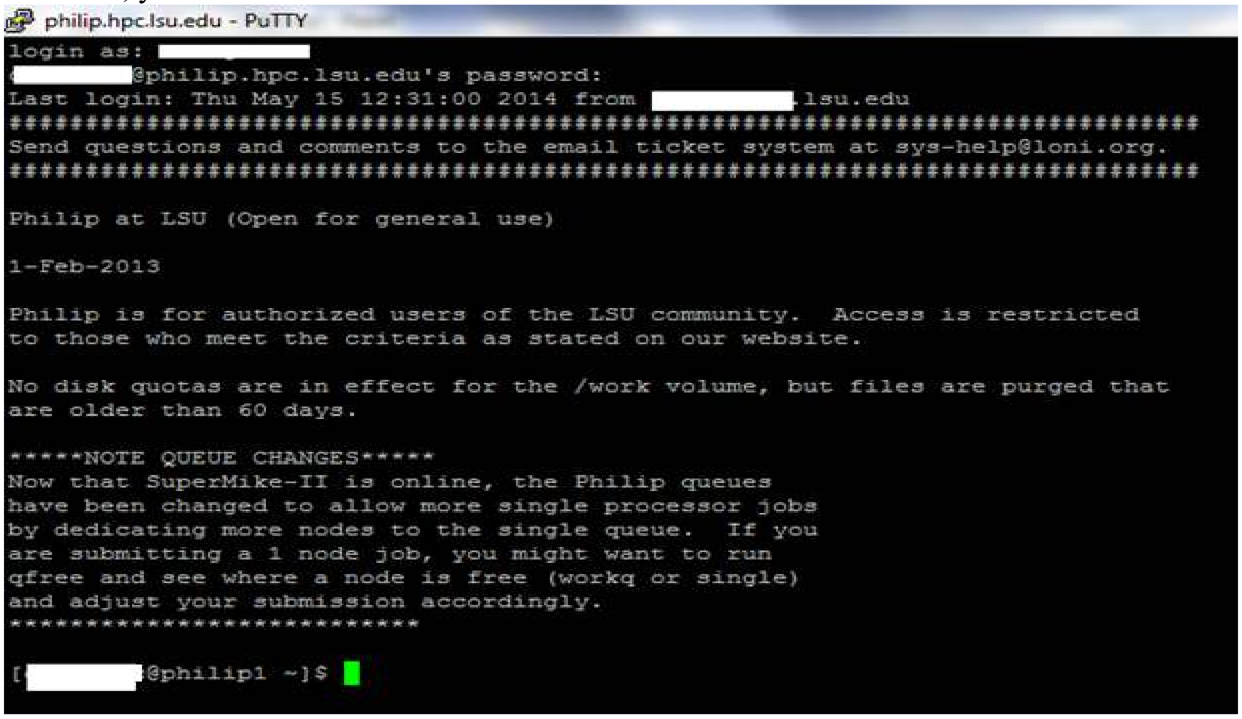
```
login as:
```

Then you are requested to type your password:



```
login as: [redacted]
[redacted]@philip.hpc.lsu.edu's password:
```

And then, you will see a window similar to this one:



```
login as: [redacted]
[redacted]@philip.hpc.lsu.edu's password:
Last login: Thu May 15 12:31:00 2014 from [redacted].lsu.edu
##############################################################################
Send questions and comments to the email ticket system at sys-help@loni.org.
##############################################################################

Philip at LSU (Open for general use)

1-Feb-2013

Philip is for authorized users of the LSU community.  Access is restricted
to those who meet the criteria as stated on our website.

No disk quotas are in effect for the /work volume, but files are purged that
are older than 60 days.

*****NOTE QUEUE CHANGES*****
Now that SuperMike-II is online, the Philip queues
have been changed to allow more single processor jobs
by dedicating more nodes to the single queue.  If you
are submitting a 1 node job, you might want to run
qfree and see where a node is free (workq or single)
and adjust your submission accordingly.
***************************

[[redacted]@philip1 ~]$
```

At this point, you are ready to work in the terminal. Now, you are ready to work!

Editing the ~/. soft file

Check if R is already installed in Philip by typing "$ softenv", and read the ordered list of programs that are already installed. If the R version that you required is not listed, then contact the helpdesk. If R is already installed, determine the software version (e.g. 3.0.0), then proceed to edit the user's environment by opening the ~/.soft file, type:
$ nano ~/.soft

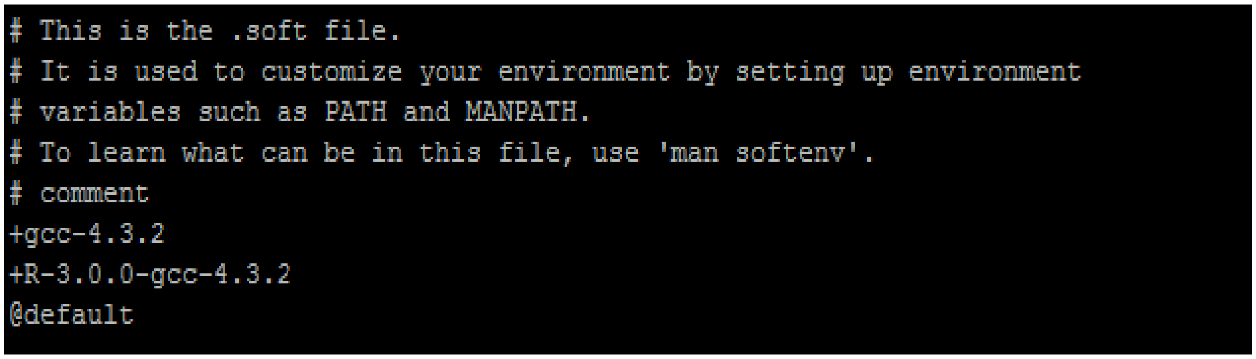
Add the following 2 lines in the NANO editor right before the @default line:
+gcc-4.3.2
+R-3.0.0-gcc-4.3.2

The last line of text is specific to the R software; the text is set by the administrator of the HPC systems. Make sure that the editor does not have any empty comments for not altering paths of the operating system, linux. The text "+gcc-4.3.2" makes reference to the compiler necessary for the R installation, notice the match. In Philip, an older version of R is also available +R-2.8.1-gcc-4.3.2 and on mike:).

The following table clarifies what to do in Philip and on Mike:

Supercomputer	Reference to compiler	R version	Add the following lines in the .soft file
Philip	Yes: +gcc-4.3.2	+R-3.0.0-gcc-4.3.2	+gcc-4.3.2 +R-3.0.0-gcc-4.3.2
Mike	It is not needed.	+R-2.15.1-gcc-4.4.6	+R-2.15.1-gcc-4.4.6

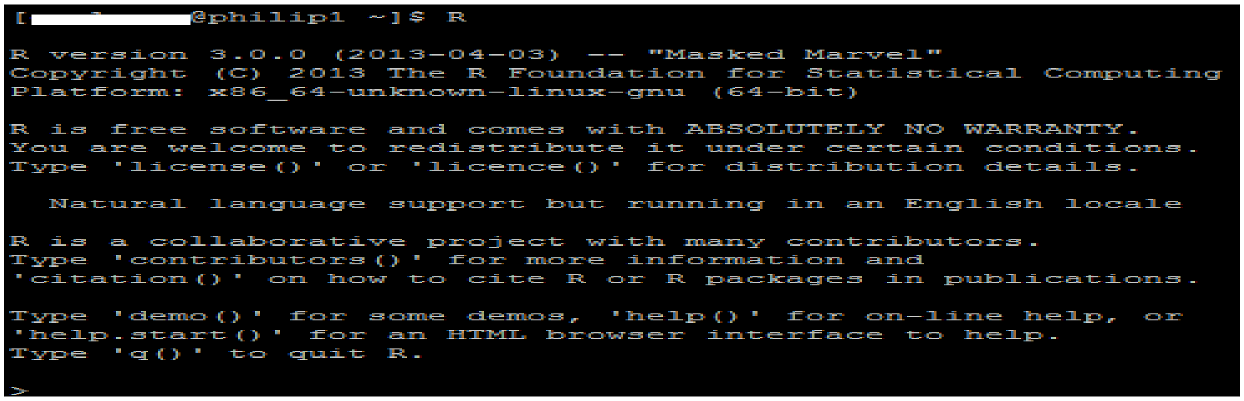
To finish, save the ~/.soft file with ctrl + x, a message will appear asking you “do you want to save changes?” Say yes. Then, say no to option for changing the name of the file; so, just hit ENTER. Finally, review the environment for displaying the software available in the system:
$ cat ~/.soft This would display the text saved into the ~/.soft file.



```
# This is the .soft file.
# It is used to customize your environment by setting up environment
# variables such as PATH and MANPATH.
# To learn what can be in this file, use 'man softenv'.
# comment
+gcc-4.3.2
+R-3.0.0-gcc-4.3.2
@default
```

You should see the version of R that is now available for use in the system. Now, you are ready to update the software environment by typing in the command line: $ resoft

Another check that can be made requires just typing the letter “R” in the command line, and you should not get an error. If you do, then log out, log in again, and type the letter R again.



```
[[redacted]@philip1 ~]$ R

R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-unknown-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.

>
```

Type q() to exit R. And you will be prompted if you want to save the workspace image, say no. Now, R is ready for use, and the R script “testofr.R” should not have a problem running.

Setting the working directory

In the cluster, you need to move to the work directory associated with your user's account, you can move into you work directory: $ cd /work/myusername/

And create a folder called rscripts: $ mkdir rscripts

Create or upload all of your shell and R scripts into this directory. I have used WinSCP for transferring files between the cluster and my computer. This new folder "rscripts" can be accessed with the "cd" command too ($$ cd /work/myusername/rscripts/). Instructions on how to create those files are explained next.

The shell script (testmyr.sh)

For submitting a job, a shell script is necessary. It makes reference to the R script and settings that instruct the cluster. The shell script has a .sh file extension. Create an .sh file or simply upload it (testmyr.sh). However, assuming that you are in the directory (/work/myusername/rscripts/), the shell script can be created from the command line too ($ touch testmyr.sh). And then, the .sh file can be edited with the nano editor ($ nano testmyr.sh). The following lines correspond to the basic content that goes into the shell script (testmyr.sh) that would control the execution of the R script:

```
#!/bin/bash
#PBS -q single
#PBS -l nodes=1:ppn=1
#PBS -l walltime=1:00:00
#PBS -o rtestout
#PBS -N testr
cd /work/myusername/rscripts/
R CMD BATCH "testofr.R"  > log_rtest.logfile
date
#And we're out'a here!
exit 0
```

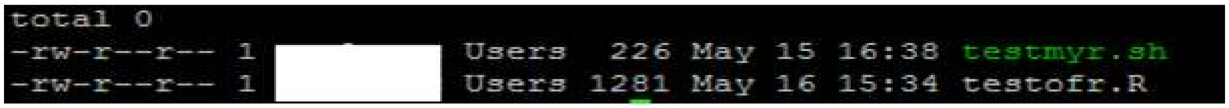
<u>Detailed explanation:</u>
Warning: in line 7: modify "myusername" with your own username.
q: type of queue. nodes: number of nodes requested. ppn: number of processors.
walltime: maximum number of hours for work
Job's output information: rtestout
Output from the screen in R: testofr.Rout [tesfofr corresponds to the name of the script]
Job name: testr
Directory assignment: cd /work/myusername/rscripts/ [change myusername with your own]
Script in R: testofr.R
Log file to be created: log_rtest.logfile

`NOTE: do not forget to change the 7th line with your own username.`

Submitting a job

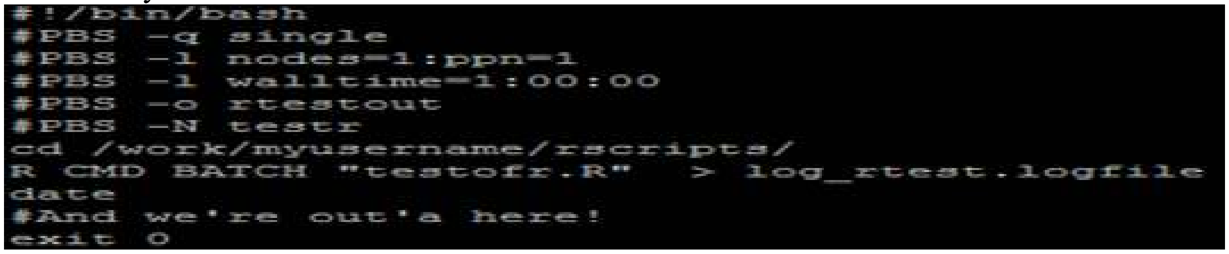
Before submitting a job, will need the .sh file and the R script in the same directory, in the specific folder that you created for this job (rscripts). In the command line, set the present work directory in the proper folder where your .sh and .R files are located, for example:
$ cd /work/myusername/rscripts/

You can corroborate the change with the pwd command. For checking if the necessary files are located in the present work directory (You shall see at least two files: testmyr.sh testofr.R.):
$ ls –l



```
total 0
-rw-r--r-- 1 [redacted] Users  226 May 15 16:38 testmyr.sh
-rw-r--r-- 1            Users 1281 May 16 15:34 testofr.R
```

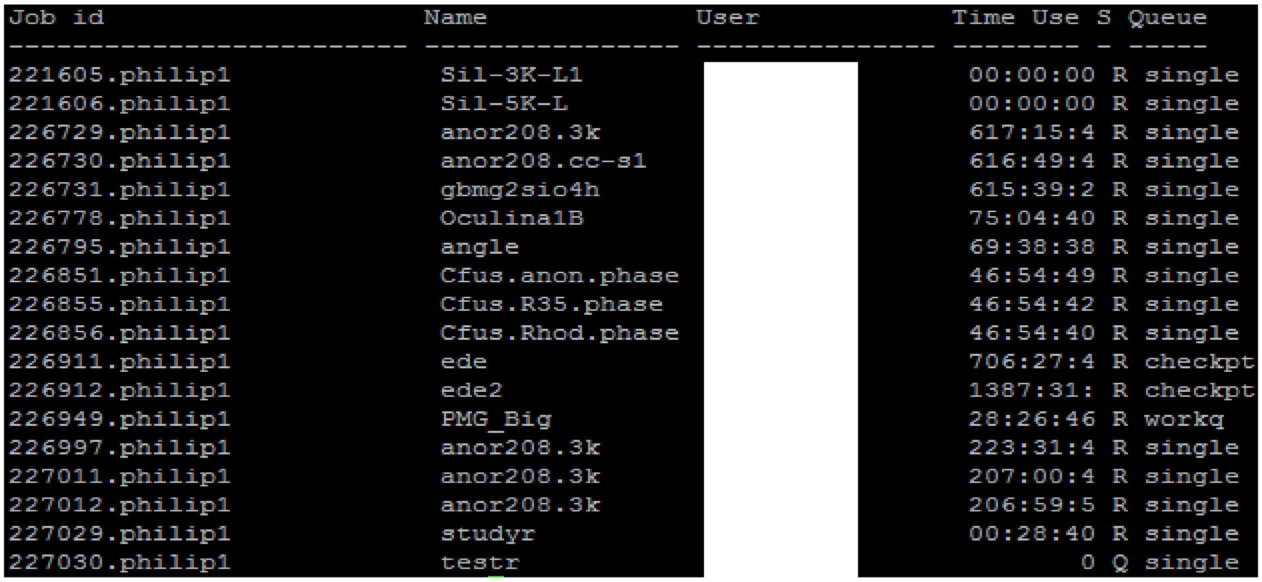
You may re-read your shell script to check for potential errors:
$ cat testmyr.sh



```
#!/bin/bash
#PBS -q single
#PBS -l nodes=1:ppn=1
#PBS -l walltime=1:00:00
#PBS -o rtestout
#PBS -N testr
cd /work/myusername/rscripts/
R CMD BATCH "testofr.R"  > log_rtest.logfile
date
#And we're out'a here!
exit 0
```

Finally, the job submission uses the “qsub” command as follows:
$ qsub testmyr.sh

Look at the status (S column, R > running Q > on queue) of your job by simply typing: $ qstat



```
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
221605.philip1             Sil-3K-L1       [redacted]       00:00:00 R single
221606.philip1             Sil-5K-L                         00:00:00 R single
226729.philip1             anor208.3k                       617:15:4 R single
226730.philip1             anor208.cc-s1                    616:49:4 R single
226731.philip1             gbmg2sio4h                       615:39:2 R single
226778.philip1             Oculina1B                        75:04:40 R single
226795.philip1             angle                            69:38:38 R single
226851.philip1             Cfus.anon.phase                  46:54:49 R single
226855.philip1             Cfus.R35.phase                   46:54:42 R single
226856.philip1             Cfus.Rhod.phase                  46:54:40 R single
226911.philip1             ede                              706:27:4 R checkpt
226912.philip1             ede2                             1387:31: R checkpt
226949.philip1             PMG_Big                          28:26:46 R workq
226997.philip1             anor208.3k                       223:31:4 R single
227011.philip1             anor208.3k                       207:00:4 R single
227012.philip1             anor208.3k                       206:59:5 R single
227029.philip1             studyr                           00:28:40 R single
227030.philip1             testr                                   0 Q single
```

Note: the study name and job id are displayed.

Output and Further input

After the job is performed, output files will be created in: /work/myusername/rscripts/

If an R script requires further input (e.g. a dataset), make sure that this is requested from the R script itself (make sure the reference to a directory is well set, ideally use the same directory).

Cautions

If for some reason you need to cancel a particular job, then you can use the "qdel" command, make reference to the job id that is obtained either when you submit the job or by the "qstat" command (e.g. 987513.philip1). For instance: $ qdel 987513.philip1

If the walltime, that is set on the shell script, has been exceeded during the execution of the script, the job will be killed. And the script would have produced only the output until the time of cancelation. So, do not panic! Check the log file for error messages and adjust the walltime accordingly. An output file named as the job name would have a message similar to this one:

```
=>> PBS: job killed: walltime 7219 exceeded limit 7200
```

If the R script is not located in the same folder as the shell script when you submit the job, you will get an error message in the testofr.Rout file that reads as follow:

```
Fatal error: cannot open file 'testofr.R': No such file or directory
```

If folders are not properly referenced to your username "myusername" in the work directory, you may obtain an error message:

```
/var/spool/torque/mom_priv/jobs/226838.philip1.SC: line 7: cd:
/work/myusername/rscripts/: No such file or directory
```

Main R script (testofr.R)

Before running the script in Philip, test it on your own computer. It there are errors, you can correct them, and make more efficient use of the available resources.

```
#Author: Carlos Garcia
#e mail: cgarci8 [A~T] tigers.lsu.edu
#Date of last update: 5/15/2014
#Time of last update: 10:55 AM
#Name of file: testofr.R
#Description: script in R for testing R in the philip HPC
#Actions of script:
#Calculates the means of y and z of simulated samples
#Provides descriptive statistics of y and z
#Exports data of means to a csv file (dm.csv)
#Creates plots
#Creates histograms
#Obtains percentile measures
#imports the produced csv file
```

```
#fits a simple linear regression

#START
simul <-10000
simul
sasi <- 50
sasi
iter <- rep(NA, simul)
y    <- rep(NA, simul)
z    <- rep(NA, simul)
dm   <- data.frame(iter,y,z)

set.seed(5)
for(i in 1:simul){
  x1 <- runif(sasi, 0, 1)
  x2 <- rnorm(sasi)
  a1 <- mean(x1)
  a2 <- mean(x2)
  dm$iter[i] <-i
  dm$y[i]    <-a1
  dm$z[i]    <-a2
}

summary(dm)
write.csv(dm, file="dm.csv")
quantile(sort(dm$y), c(.85, .90, .95,0.99,0.999))
quantile(sort(dm$z), c(.85, .90, .95,0.99,0.999))

#graphs
png('rplot.png')
plot(dm$y,dm$z)
dev.off()

png('Rplot01dmy.png')
hist(dm$y)
dev.off()

png('Rplot02dmz.png')
hist(dm$z)
dev.off()

jpeg('rplot.jpg')
plot(dm$y,dm$z)
dev.off()
```

```
mydm = read.csv("dm.csv")
fit <- lm(z ~ y, data=mydm)
summary(fit)
#END
```

Output of R script

The following files will be produced by the R script (testofr.R):

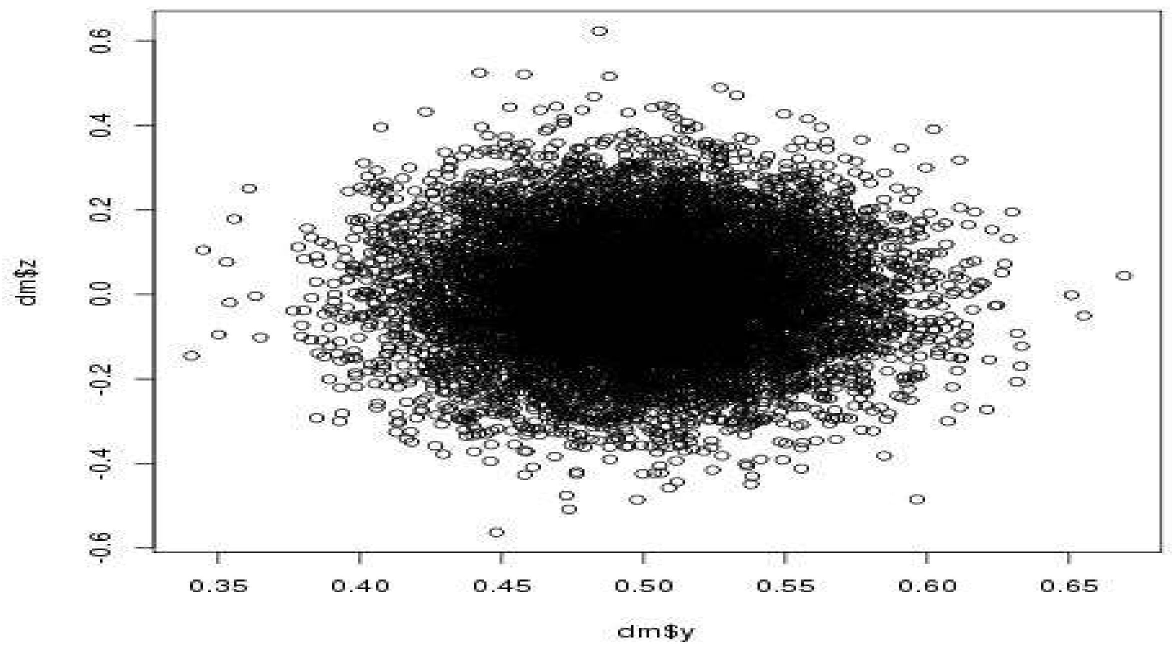
- Output from the screen in R: testofr.Rout
- Graphs:
    - rplot.jpg
    - rplot.png
    - Rplot01dmy.png
    - Rplot02dmz.png
- A csv file: dm.csv
- Log empty file: log_rtest.logfile
- Job's output information: rtestout

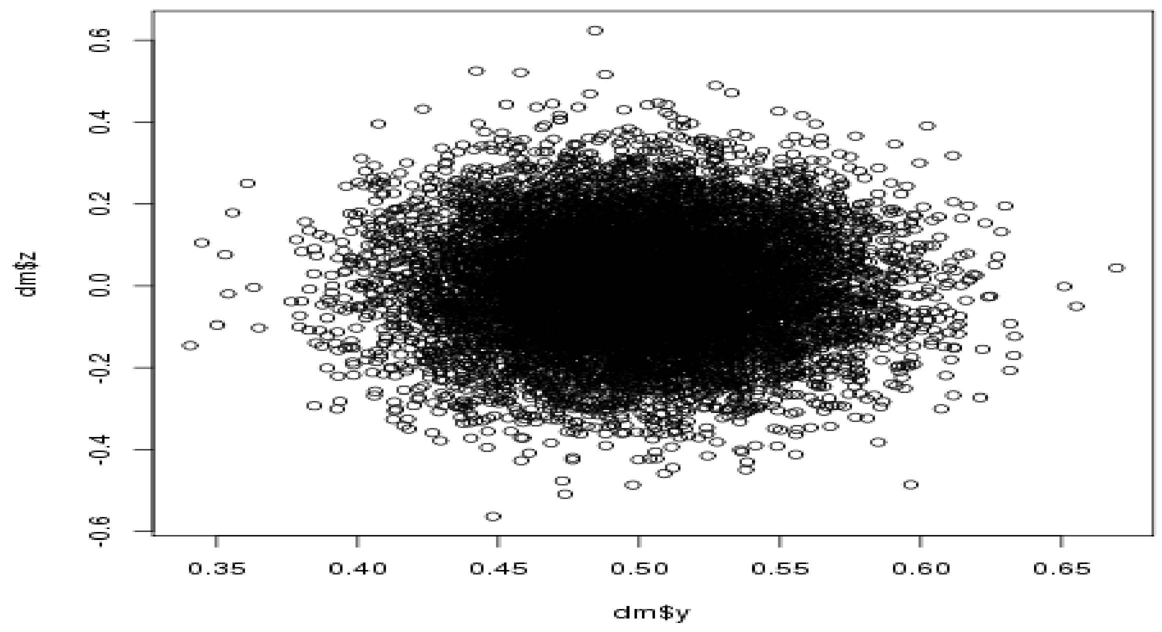
FILE: testofr.Rout

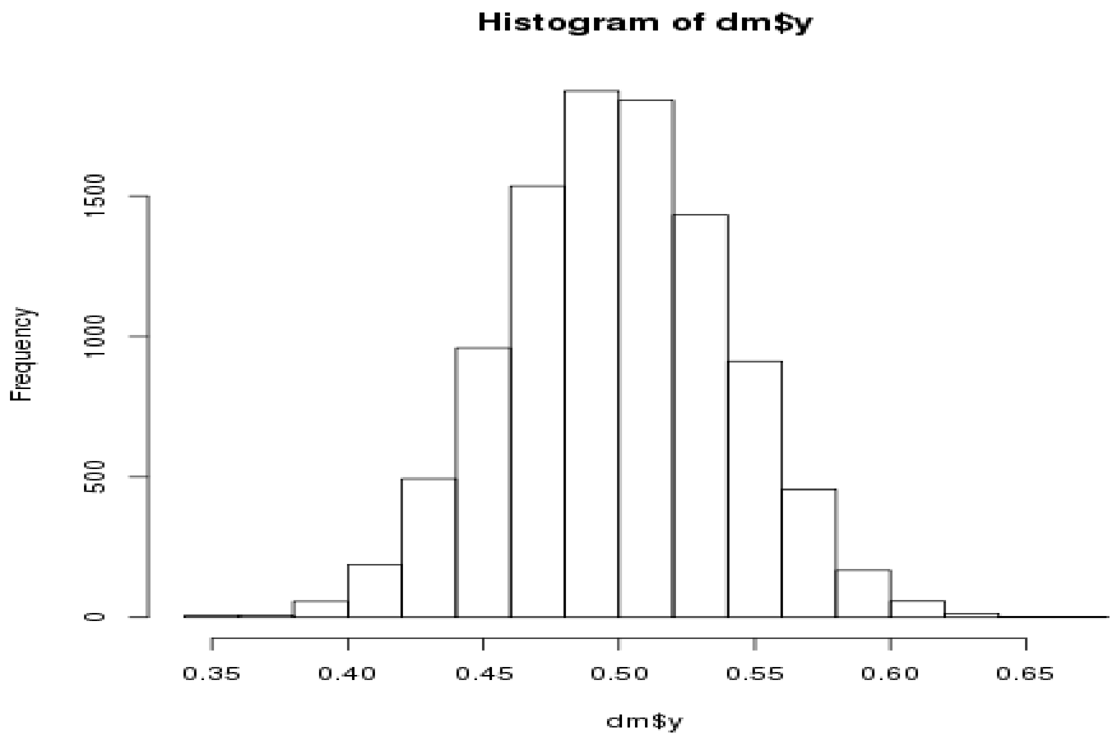
```
R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

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.

> #Author: Carlos Garcia
> #e mail: cgarci8 [A~T] tigers.lsu.edu
> #Date of last update: 5/15/2014
> #Time of last update: 10:55 AM
> #Name of file: testofr.R
> #Description: script in R for testing R in the philip HPC
> #Actions of script:
> #Calculates the means of y and z of simulated samples
> #Provides descriptive statistics of y and z
> #Exports data of means to a csv file (dm.csv)
> #Creates plots
> #Creates histograms
> #Obtains percentile measures
> #imports the produced csv file
> #fits a simple linear regression
> #START
> simul <-10000
> simul
[1] 10000
> sasi <- 50
> sasi
[1] 50
> iter <- rep(NA, simul)
> y    <- rep(NA, simul)
> z    <- rep(NA, simul)
> dm   <- data.frame(iter,y,z)
>
> set.seed(5)
> for(i in 1:simul){
+   x1 <- runif(sasi, 0, 1)
+   x2 <- rnorm(sasi)
+   a1 <- mean(x1)
+   a2 <- mean(x2)
+   dm$iter[i] <-i
+   dm$y[i]    <-a1
+   dm$z[i]    <-a2
+ }
>
> summary(dm)
      iter             y                  z
 Min.   :    1   Min.   :0.3406   Min.   :-0.563339
 1st Qu.: 2501   1st Qu.:0.4714   1st Qu.:-0.096695
 Median : 5000   Median :0.4987   Median :-0.003024
 Mean   : 5000   Mean   :0.4991   Mean   :-0.001426
 3rd Qu.: 7500   3rd Qu.:0.5266   3rd Qu.: 0.093427
 Max.   :10000   Max.   :0.6695   Max.   : 0.623114
> write.csv(dm, file="dm.csv")
> quantile(sort(dm$y), c(.85, .90, .95,0.99,0.999))
      85%       90%       95%       99%     99.9%
0.5421492 0.5524645 0.5665974 0.5955694 0.6262931
> quantile(sort(dm$z), c(.85, .90, .95,0.99,0.999))
      85%       90%       95%       99%     99.9%
0.1458127 0.1808530 0.2305243 0.3288482 0.4422989
>
> #graphs
> png('rplot.png')
> plot(dm$y,dm$z)
> dev.off()
null device
          1
>
> png('Rplot01dmy.png')
> hist(dm$y)
> dev.off()
```

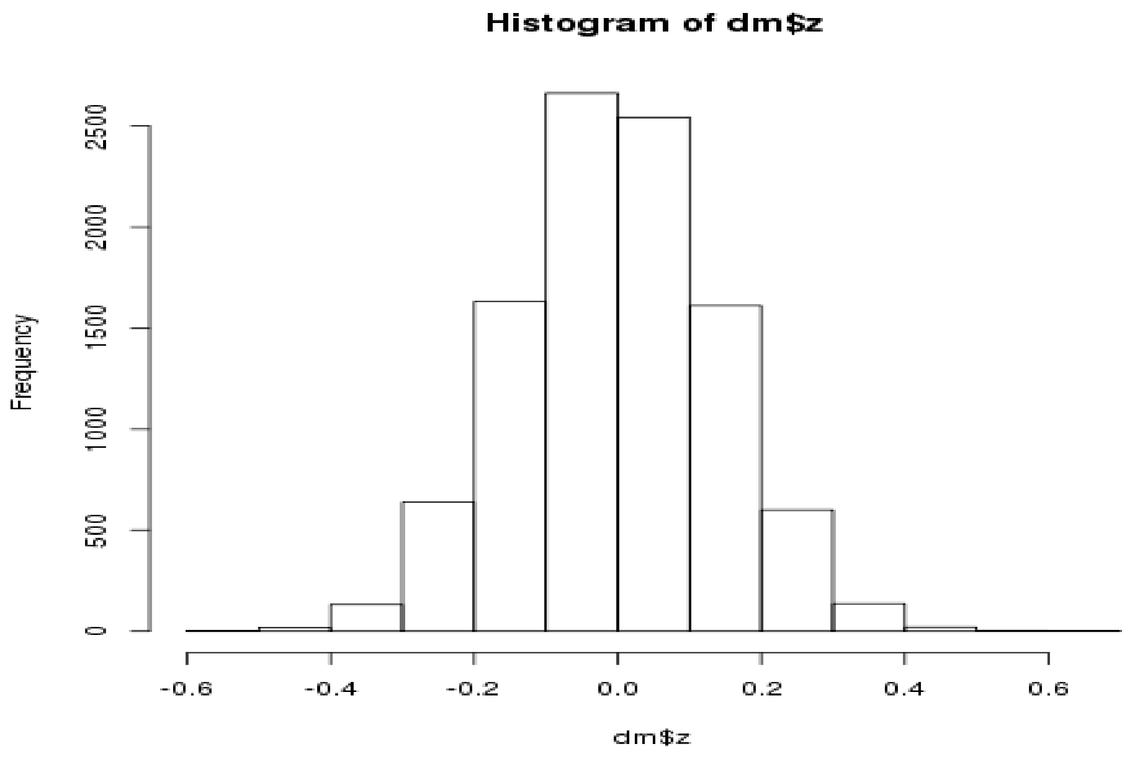
```
null device
          1
>
> png('Rplot02dmz.png')
> hist(dm$z)
> dev.off()
null device
          1
>
> jpeg('rplot.jpg')
> plot(dm$y,dm$z)
> dev.off()
null device
          1
>
> mydm = read.csv("dm.csv")
> fit <- lm(z ~ y, data=mydm)
> summary(fit)

Call:
lm(formula = z ~ y, data = mydm)

Residuals:
     Min       1Q   Median       3Q      Max
-0.56205 -0.09519 -0.00155  0.09476  0.62450

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0001093  0.0172000  -0.006    0.995
y           -0.0026380  0.0343452  -0.077    0.939

Residual standard error: 0.141 on 9998 degrees of freedom
Multiple R-squared:  5.901e-07,  Adjusted R-squared:  -9.943e-05
F-statistic: 0.0059 on 1 and 9998 DF,  p-value: 0.9388

> #END
>
> proc.time()
   user  system elapsed
 12.066   0.027  12.487
```

Graphs


dm$z
dm$y
0.35
0.40
0.45
0.50
0.55
0.60
0.65
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6


dm$z
dm$y
0.35
0.40
0.45
0.50
0.55
0.60
0.65
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6


Histogram of dm$y
Frequency
0
500
1000
1500
0.35
0.40
0.45
0.50
0.55
0.60
0.65
dm$y


Histogram of dm$z
Frequency
0
500
1000
1500
2000
2500
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
dm$z

Job's output information: rtestout

```
--------------------------------------
Running PBS prologue script
--------------------------------------
User and Job Data:
--------------------------------------
Job ID:    226800.philip1
Username:  yourusername
Group:     Users
Date:      15-May-2014 11:08
Node:      philip003 (22655)
--------------------------------------
PBS has allocated the following nodes:

philip003

A total of 1 processors on 1 nodes allocated
---------------------------------------------
Check nodes and clean them of stray processes
---------------------------------------------
Checking node philip003 11:08:53
-> User yourusername running job 226800.philip1:state=R:ncpus=1 (This job)
Done clearing all the allocated nodes
------------------------------------------------------
Concluding PBS prologue script - 15-May-2014 11:08:53
------------------------------------------------------
Thu May 15 11:09:07 CDT 2014
------------------------------------------------------
Running PBS epilogue script    - 15-May-2014 11:09:07
------------------------------------------------------
Checking node philip003 (MS)
-> User   running job 221574.philip1:state=R:ncpus=1
-> User   running job 226721.philip1:state=R:ncpus=1
-> User   running job 226723.philip1:state=R:ncpus=1
-> User   running job 226724.philip1:state=R:ncpus=1
-> User   running job 226725.philip1:state=R:ncpus=1
-> User   running job 226726.philip1:state=R:ncpus=1
-> User   running job 226795.philip1:state=R:ncpus=1
-> User yourusername  running job 226800.philip1:state=R:ncpus=1 (This job)
Checking node philip003 ok
-> User   running job 221574.philip1:state=R:ncpus=1
-> User   running job 226721.philip1:state=R:ncpus=1
-> User   running job 226723.philip1:state=R:ncpus=1
-> User   running job 226724.philip1:state=R:ncpus=1
-> User   running job 226725.philip1:state=R:ncpus=1
-> User   running job 226726.philip1:state=R:ncpus=1
-> User   running job 226795.philip1:state=R:ncpus=1
-> User yourusername  running job 226800.philip1:state=R:ncpus=1 (This job)
------------------------------------------------------
Concluding PBS epilogue script - 15-May-2014 11:09:09
------------------------------------------------------
Exit Status:     0
Job ID:          226800.philip1
Username:        yourusername
Group:           Users
Job Name:        testr
Session Id:      22654
Resource Limits: ncpus=1,neednodes=1:ppn=1,nodes=1:ppn=1,walltime=15:00:00
Resources Used:  cput=00:00:14,mem=0kb,vmem=0kb,walltime=00:00:16
Queue Used:      single
Account String:
Node:            philip003
Process id:      23542
------------------------------------------------------
```

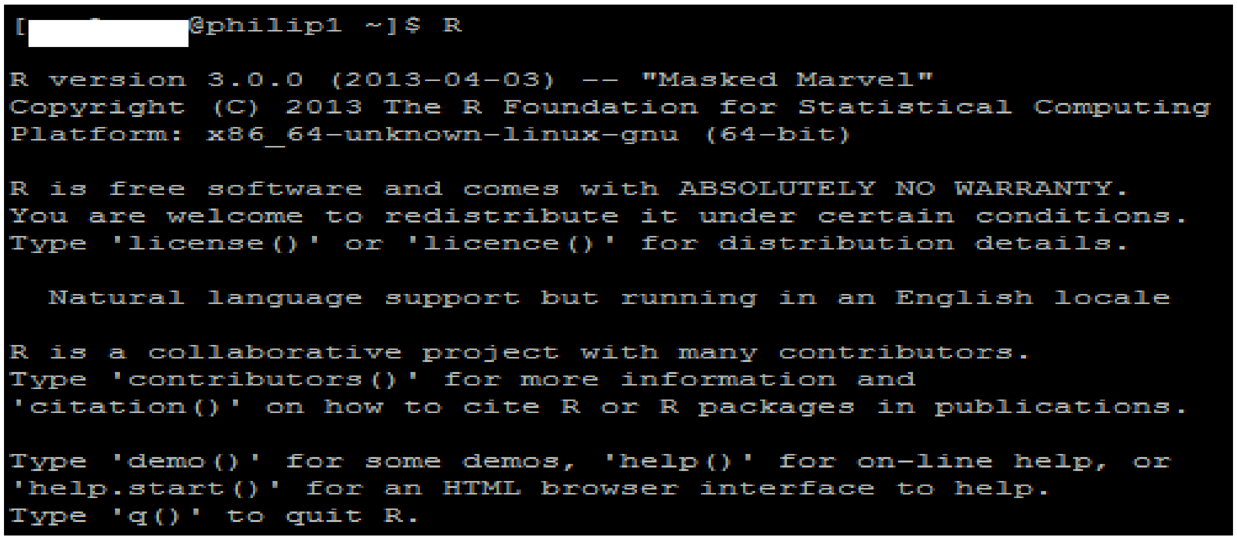
Files needed for replicating demonstration

R script: testofr.R
Shell script: testmyr.sh

Working with R packages

Installing R packages

In this demonstration, we would install a commonly used package ggplot2. On the home directory, start by running an R shell as follows: $ R

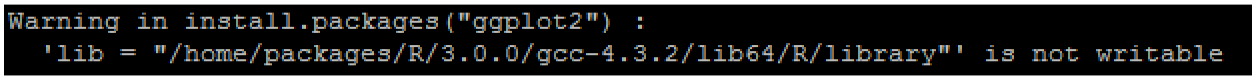


```
[[redacted]@philip1 ~]$ R

R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-unknown-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.
```

And then, type on the R shell the command for installing packages, the following line installs the ggplot2 packgage:

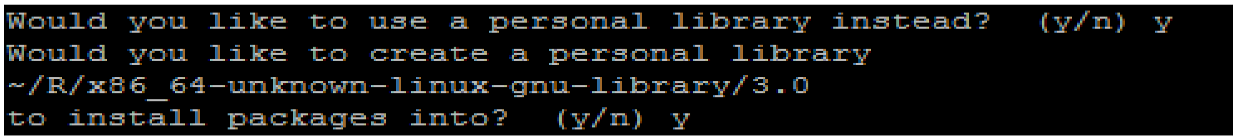
$ install.packages("ggplot2")

You may obtain a warning message



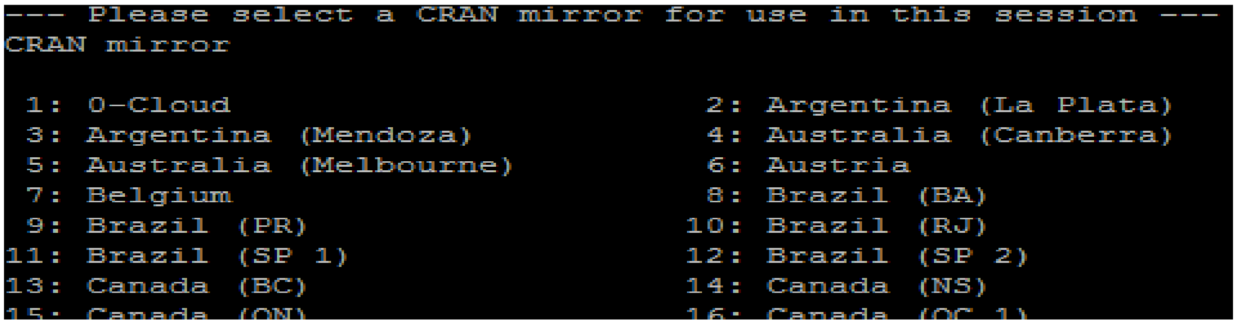
```
Warning in install.packages("ggplot2") :
  'lib = "/home/packages/R/3.0.0/gcc-4.3.2/lib64/R/library"' is not writable
```

And then you will be asked two questions, say yes with the letter “y” and hit enter:



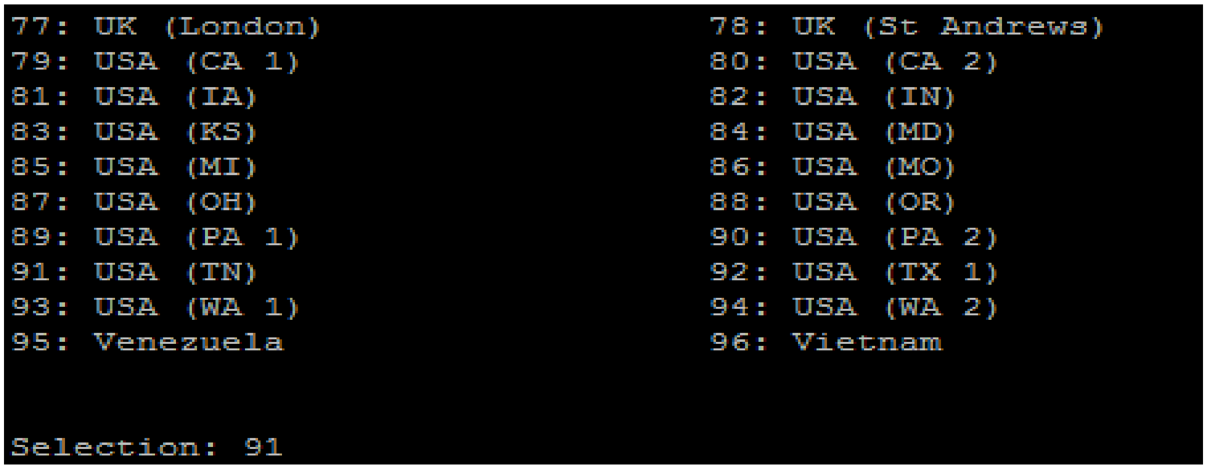
```
Would you like to use a personal library instead?  (y/n) y
Would you like to create a personal library
~/R/x86_64-unknown-linux-gnu-library/3.0
to install packages into?  (y/n) y
```

Then, you will be prompted to select a CRAN (Comprehensive R Archive Network for the R programming language) for downloading the package



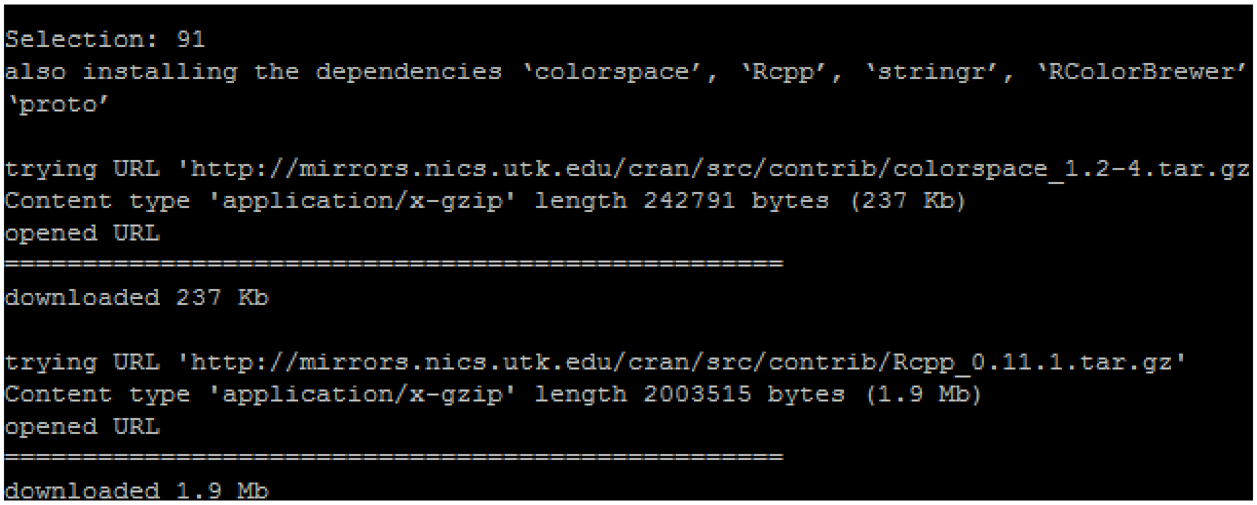
```
--- Please select a CRAN mirror for use in this session ---
CRAN mirror

 1: 0-Cloud                          2: Argentina (La Plata)
 3: Argentina (Mendoza)              4: Australia (Canberra)
 5: Australia (Melbourne)            6: Austria
 7: Belgium                          8: Brazil (BA)
 9: Brazil (PR)                     10: Brazil (RJ)
11: Brazil (SP 1)                   12: Brazil (SP 2)
13: Canada (BC)                     14: Canada (NS)
15: Canada (ON)                     16: Canada (QC 1)
```

I selected CRAN #91.



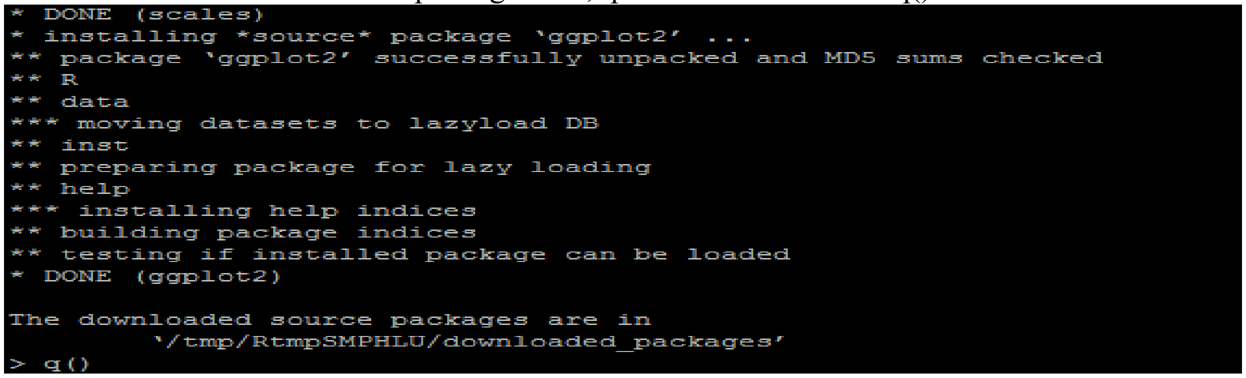
```
77: UK (London)                     78: UK (St Andrews)
79: USA (CA 1)                      80: USA (CA 2)
81: USA (IA)                        82: USA (IN)
83: USA (KS)                        84: USA (MD)
85: USA (MI)                        86: USA (MO)
87: USA (OH)                        88: USA (OR)
89: USA (PA 1)                      90: USA (PA 2)
91: USA (TN)                        92: USA (TX 1)
93: USA (WA 1)                      94: USA (WA 2)
95: Venezuela                       96: Vietnam


Selection: 91
```

And once you type the CRAN number and hit enter, the download and installation begins.



```
Selection: 91
also installing the dependencies ‘colorspace’, ‘Rcpp’, ‘stringr’, ‘RColorBrewer’
‘proto’

trying URL 'http://mirrors.nics.utk.edu/cran/src/contrib/colorspace_1.2-4.tar.gz
Content type 'application/x-gzip' length 242791 bytes (237 Kb)
opened URL
==================================================
downloaded 237 Kb

trying URL 'http://mirrors.nics.utk.edu/cran/src/contrib/Rcpp_0.11.1.tar.gz'
Content type 'application/x-gzip' length 2003515 bytes (1.9 Mb)
opened URL
==================================================
downloaded 1.9 Mb
```

And once the installation of the package ends, quit the R shell with $ q().



```
* DONE (scales)
* installing *source* package 'ggplot2' ...
** package 'ggplot2' successfully unpacked and MD5 sums checked
** R
** data
*** moving datasets to lazyload DB
** inst
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (ggplot2)

The downloaded source packages are in
        '/tmp/RtmpSMPHLU/downloaded_packages'
> q()
```

Using R packages

R script (testofrp.R)

The process of working with R packages in the HPC cluster is similar to the process that one implements when working on a personal computer. However, the R script, as usual has to contain references to the packages to be used while the system runs the script. Thus, the R script "testofr.R" is modified accordingly. The package ggplot2 is called with the following code line:
library(ggplot2)

Then, the qplot command is used in two ways.
First way:
qplot(y,z, data=mydm)

Second way:
jpeg("cuterplot.jpg")
qplot(y,z, data=mydm)
dev.off()

In all, the new R script "testofrp.R" contains the following lines of code at the end:

```
library(ggplot2)
qplot(y,z, data=mydm)
jpeg("cuterplot.jpg")
qplot(y,z, data=mydm)
dev.off()
```

Shell script (testmyrp.sh)

The shell script is modified, for making reference to a new R script "testofrp.R", a new job name “testrp” and a new log file log_rtestp.logfile. The shell script contains the following lines:

```
#!/bin/bash
#PBS -q single
#PBS -l nodes=1:ppn=1
#PBS -l walltime=1:00:00
#PBS -o rtestout
#PBS -N testrp
cd /work/myusername/rscripts/
R CMD BATCH "testofrp.R"  > log_rtestp.logfile
date
#And we're out'a here!
exit 0
```

Changes
Reference to the R script is now: testofrp.R
Reference to the log file is now: log_rtestp.logfile
Reference to job name is now: testrp

`NOTE: do not forget to change the 7th line with your username.`

Job submission

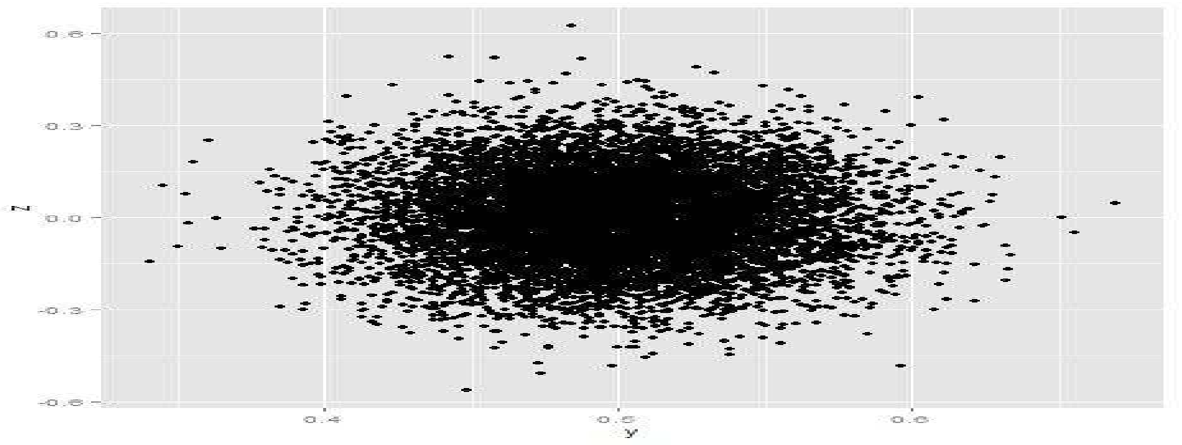
The job submission is performed as usual, with the qsub command.

$ qsub testmyrp.sh

Note: The files testmyrp.sh and mytestofrp.R should be located in the working directory. Access it with: $cd /work/myusername/rscripts/

Output

You should see the same output from the tested R script (testofr.R). And if the package was correctly installed, you should obtain a PDF file containing the new plot (Rplots.pdf). Moreover, the same graph in JPG format is produced too (cuterplot.jpg). It looks like the following graph:



Files needed for replicating demonstration

R script: testofrp.R
Shell script: testmyrp.sh

Useful resources

Linux commands
The following commands were used repeatedly throughout this demonstration:
cat, resoft, cd, mkdir, pwd, touch, qsub, qstat, qdel, ls, ls -l

R commands: R, q(), install.packages("packgagename"), library(libname), qplot

Please review some information about the mentioned commands before reproducing these demonstrations.

Useful websites

R Software
- R project http://www.r-project.org/
- R phylo http://www.r-phylo.org/wiki/HowTo/Taskview
- R software description from the LSU HPC: http://www.hpc.lsu.edu/docs/guides/software.php?software=R
- CRAN Task View: Phylogenetics, Especially Comparative Methods http://cran.r-project.org/web/views/Phylogenetics.html
- CRAN Task View: Statistical Genetics http://cran.r-project.org/web/views/Genetics.html
- CRAN Task View: High-Performance and Parallel Computing with R http://cran.r-project.org/web/views/HighPerformanceComputing.html

LSU HPC
- Main website: http://www.hpc.lsu.edu/
- Philip: http://www.hpc.lsu.edu/docs/guides.php?system=Philip
- Mike: http://www.hpc.lsu.edu/resources/hpc/system.php?system=SuperMike-II

Utilities
- Notepad++ http://notepad-plus-plus.org/
- Putty: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- Nano editor http://www.nano-editor.org/
- WinSCP: http://winscp.net/eng/index.php
- GCC, the GNU Compiler Collection: https://gcc.gnu.org/

Linux
- Cheat sheet http://linoxide.com/linux-command/linux-commands-cheat-sheet/
- Best Linux Cheat Sheets http://www.nixtutor.com/linux/all-the-best-linux-cheat-sheets/

Last piece of advice

If possible, every time that output is obtained from the HPC cluster, save it on different folders. In each folder, create a .txt file with detailed information about the contents and respective job (e.g. aboutfolder.txt). The updated version of this demonstration can be found at http://www.carlosignacio.com/ignatius/hpc.php. Feel free to send questions or suggestions <cgarci8 AT tigers . lsu . edu >