

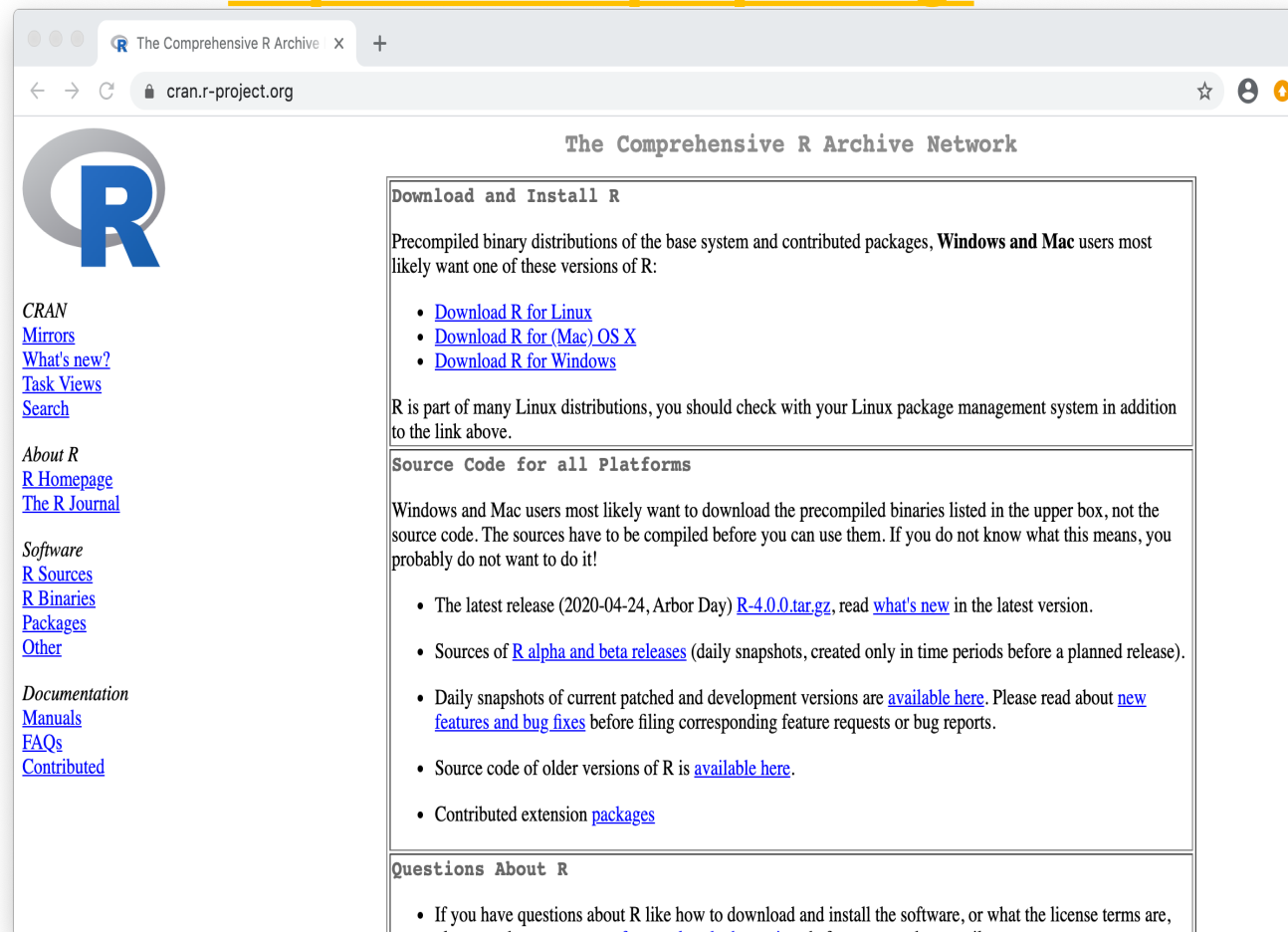
Data Analysis in R using High Performance Computing (HPC)

Tannistha Nandi, PhD (Bioinformatics)
Data Scientist, Research Computing Services
Bioinformatics National Team Lead, DRAC

Is R the right tool?



<https://cran.r-project.org/>



The screenshot shows the CRAN website in a web browser. The browser's address bar displays "cran.r-project.org". The page title is "The Comprehensive R Archive Network". The R logo is prominently displayed on the left. The main content area is titled "Download and Install R" and provides information about precompiled binary distributions for Windows and Mac users. It lists links for downloading R for Linux, Mac OS X, and Windows. Below this, it mentions that R is part of many Linux distributions and suggests checking with the Linux package management system. The "Source Code for all Platforms" section explains that Windows and Mac users should download precompiled binaries rather than source code, which needs to be compiled. It lists links for the latest release (R-4.0.0.tar.gz), alpha and beta releases, daily snapshots, and source code of older versions. The "Questions About R" section provides a link to frequently asked questions.

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-04-24, Arbor Day) [R-4.0.0.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Does your workflow need HPC?

Scenario 1:

same R code needs to be run multiple times (usually on different input data)

Scenario 2:

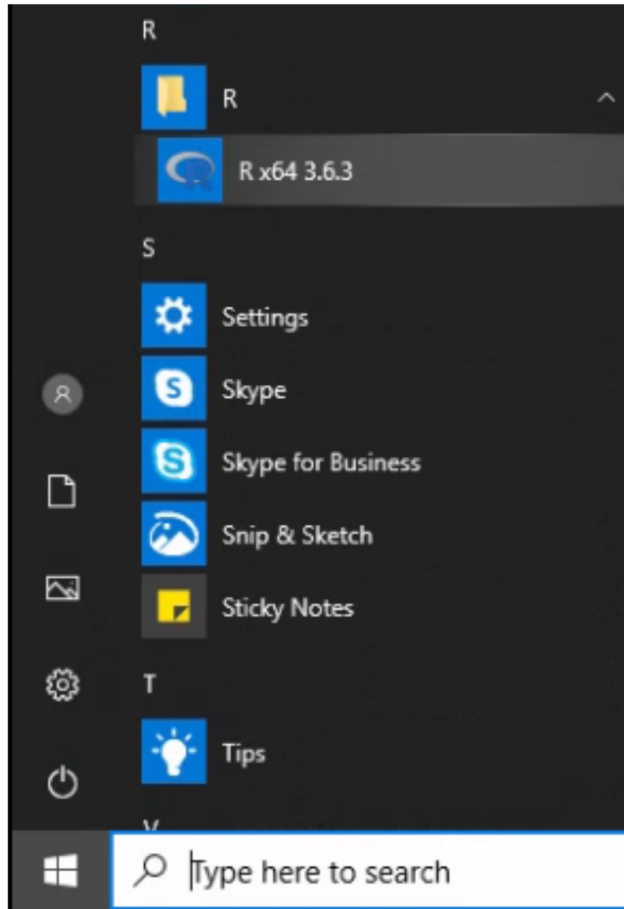
computations need much more memory than what is available on your computer

Scenario 3:

Workflow needs GPU accelerator (code can make use of GPU).

HPC supports Command Line Interface

Graphical User Interface



Command line Interface

```
[tannistha.nandi@arc ~]$
```

R in HPC environment.....1

```
[tannistha.nandi@arc ~]$  
[tannistha.nandi@arc ~]$ module avail R  
----- /global/software/Modules/4.6.0/modulefiles -----  
R/3.5.3  R/3.6.2  R/4.2.6  
[tannistha.nandi@arc ~]$ module load R/3.6.2  
Loading R/3.6.2  
  Loading requirement: lib/openblas/0.3.5-gnu lib/readline/6.3  
[tannistha.nandi@arc ~]$
```



R in HPC environment2

```
[tannistha.nandi@arc ~]$ R
```

```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night" Copyright (C) 2019
```

```
The R Foundation for Statistical Computing
```

```
Platform: x86_64-pc-linux-gnu (64-bit) R is free software and comes with  
ABSOLUTELY NO 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.
```

```
>
```

```
> 2 + 3
```

```
[1] 5
```

```
> sum(2, 3)
```

```
[1] 5
```

```
> sum(2, 3, 4)
```

```
[1] 9
```

```
> x=c(2, 3, 4)
```

```
> sum(x)
```

```
[1] 9
```

```
>
```



R in HPC environment3

```
> x=c(2,3,3,5,5,6,6,6,7)
> hist(x)
> jpeg(file="plot1.jpeg")
> hist(x, col="darkgreen")
> dev.off()
```

>?hist

hist

package:graphics

R Documentation

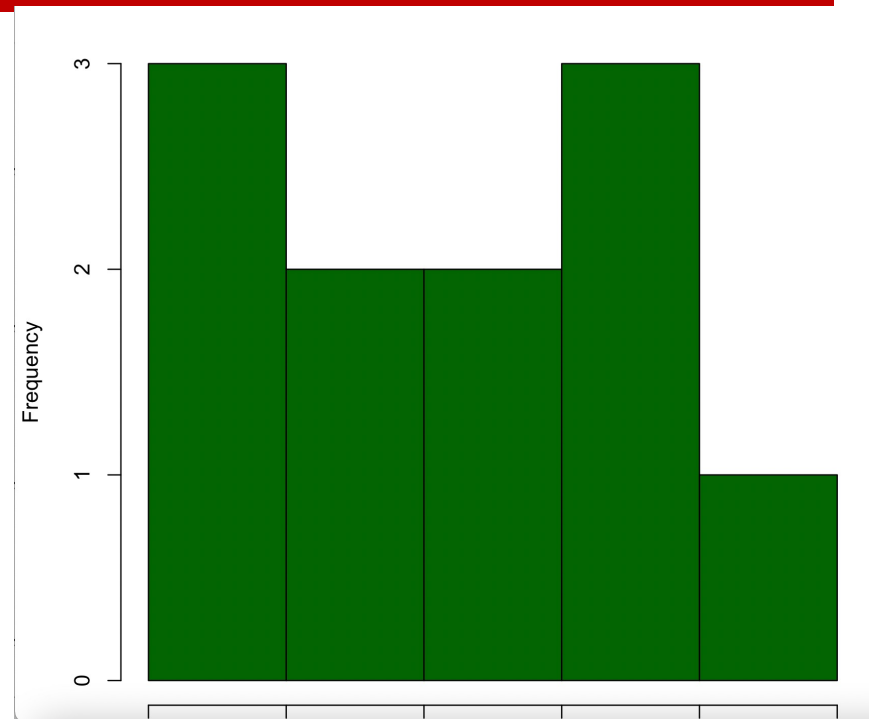
Histograms

Description:

The generic function 'hist' computes a histogram of the given data values. If 'plot = TRUE', the resulting object of class '"histogram"' is plotted by 'plot.histogram', before it is returned.

Usage:

```
hist(x, ...)
```





R in HPC environment4

```
> help(hist)
```

```
hiHistograms
```

```
Description:
```

```
The generic function 'hist' computes a histogram of the given data values. If 'plot = TRUE', the resulting object of class '"histogram"' is plotted by 'plot.histogram', before it is returned.
```

```
st
```

```
package:graphics
```

```
R Documentation
```

```
Usage:
```

```
hist(x, ...)
```

```
> example(hist)
```

```
hist> op <- par(mfrow = c(2, 2))
```

```
hist> hist(islands)
```

```
hist> utils::str(hist(islands, col = "gray", labels = TRUE))
```

```
List of 6
```

```
$ breaks : num [1:10] 0 2000 4000 6000 8000 10000 12000 14000 16000 18000
```

```
$ counts : int [1:9] 41 2 1 1 1 1 0 0 1
```

```
$ density : num [1:9] 4.27e-04 2.08e-05 1.04e-05 1.04e-05 1.04e-05 ...
```

```
$ mids : num [1:9] 1000 3000 5000 7000 9000 11000 13000 15000 17000
```

```
$ xname : chr "islands"
```

```
$ equidist: logi TRUE
```

```
- attr(*, "class")= chr "histogram"
```

```
> quit()
```

```
Save workspace image? [y/n/c]: n
```

```
[tannistha.nandi@arc ~]$
```



R in HPC environment6

```
[tannistha.nandi@arc ~]$ cat mycode.R
```

```
x=c(2,3,3,5,5,6,6,6,7)
```

```
jpeg(file="plot1.jpeg")
```

```
hist(x, col="darkgreen")
```

```
dev.off()
```

```
[tannistha.nandi@arc ~]$ Rscript mycode.R
```

```
[tannistha.nandi@arc ~]$ ls plot1.jpeg
```

```
plot1.jpeg
```

How to request for compute resources?

```
[tannistha.nandi@arc ~]$ salloc --mem=20G --time=01:00:00 --cpus-per-task=6
salloc: Granted job allocation 19501396
salloc: Waiting for resource configuration
salloc: Nodes fc29 are ready for job
[tannistha.nandi@fc29 ~]$
[tannistha.nandi@fc29 ~]$ export PATH=~/software/R-4.2.3/bin:$PATH
[tannistha.nandi@fc29 ~]$ R
```

R version 4.2.3 (2023-03-15) -- "**Shortstop Beagle**"

Copyright (C) 2023 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.

>

Speeding up R code

```
> Simulation <- function(n) {  
+ ntests <- 10000  
+ pop <- 1:365  
+ anydup <- function(i)  
+ any(duplicated(  
+ sample(pop, n, replace=TRUE)))  
+ sum(sapply(seq(ntests), anydup))/ntests  
+ }
```

#SEQUENTIAL

```
> getDoParWorkers()  
[1] 1  
> system.time(s_result <- sapply(1:100, Simulation))  
   user  system elapsed  
15.334    0.001   15.395
```

Speeding up R code2

Loop with foreach

```
foreach (n=1:100) %dopar% Simulation(n)
```

Parallel processing by registering a backend, some options are:

- | | |
|-----------------------|---|
| 1. registerDoSEQ | default with 'foreach' |
| 2. registerDoParallel | register 'DoParallel' to be used with 'foreach' |
| 3. registerDoMC | register 'doMC' to be used with 'foreach' |

Speeding up R code3

```
> library(doParallel)
> cluster1 <- makeCluster(2) #local cluster with 2 workers
> registerDoParallel(cluster1)
> getDoParWorkers()
[1] 2

> system.time(p1_result <-foreach(n=1:100) %dopar% Simulation(n))
   user  system elapsed 
0.044   0.009    8.859
```

Speeding up R code4

```
> library(doMC)
> registerDoMC(6)
> getDoParWorkers()
[1] 6

> system.time(p2_result <-foreach(n=1:100) %dopar% Simulation(n))
   user  system elapsed 
17.153   0.351   3.027
```

Efficient parallelisation & communication overhead

Parallelisation doesn't come for free:

There is a communication overhead in sending objects to and receiving objects from each parallel core.

R in HPC environment8

Province	Age	Salary	Product Purchased
Alberta	44	72000	No
Toronto	29	48000	Yes
Toronto	30	55000	No
Vancouver	38	61000	No
Vancouver	36	-	Yes
Toronto	35	58000	Yes
Toronto	-	52000	No
Alberta	48	79000	Yes
Alberta	50	83000	No
Vancouver	37	63000	Yes
Toronto	35	58000	Yes
Toronto	54	52000	No
Alberta	36	79000	Yes
Alberta	89	48000	No
Vancouver	57	76000	Yes



Import data in R

```
#import a tab delimited file (.txt)
```

```
data = read.table("Data.txt", sep="\t", header=T)
```

```
# import a comma delimited file (.csv)
```

```
data = read.csv("Data.csv", sep=",", header=T)
```

```
# Install and Load packages
```

```
#install.packages("readxl")
```

```
library("readxl")
```

```
#import an excel file(.xlsx)
```

```
data = read_excel("Data.xlsx", sheet =1) #specify the  
sheet by its index/ name
```

Getting ready for HPC

1. Command line editors like nano, vi , emacs

2. Knowledge of Linux

<http://linuxcommand.org/tlcl.php>

3. Job scheduler :SLURM

(Simple Linux Utility for Resource Management)

Reference resources

Linux

<http://linuxcommand.org/tlcl.php>

R

<https://cran.r-project.org/>

Job scheduler

<https://slurm.schedmd.com>



4/30/23

UNIVERSITY OF
CALGARY

20



Reach us if you need
help to get started

support@hpc.ucalgary.ca

Happy to take your questions!!