1. Login to HARDAC

> ssh ds394@hardac-login.genome.duke.edu

1. Login to an interactive node

> srun -p interactive --pty /bin/bash

1. Load R and its packages (<https://wiki.duke.edu/pages/viewpage.action?spaceKey=GI&title=Installing+custom+add-on+packages>)

> module load R

> module load R\_packages

1. To run R, type R + enter

> R

Now you will be in R, that looks like the old school GUI. This is the GUI that everyone made fun of me for using instead of RStudio. Comes in handy now, huh? The environment is the directory, in which you started R.

1. How do we run a basic script?

This assumes the user knows how to read and write files in R.

Exit R.

> quit()

1. Create a silly file in your terminal

> for i in {1..10}; do echo $i ; done > silly.txt

1. In your favorite text editor, create a file with the extension .R :

> vim silly.R

#!/usr/bin/env Rscript

# Read in the table

silly <- read.table(‘silly.txt’)

# do something to silly table

silly$V2 <- silly$V1+1

# write out new silly table

write.table(silly, ‘silly\_2.txt’,sep=”\t”,colnames=F,rownames=F,quotes=F)

1. To run your scripts in your terminal

> Rscript silly.R

1. What if we want to pass through arguments? <https://www.r-bloggers.com/passing-arguments-to-an-r-script-from-command-lines/>
   1. Create a list containing all the arguments you will pass through your Rscript command

args = commandArgs(trailingOnly=TRUE)

* 1. To create variables from your list

silly <- args[1]

output <- args[2]

* 1. Run the Rscript

> Rscript silly.R silly.txt silly\_2.txt

1. What if we have a s#!t ton of files?

Create a shell script to run in HARDAC

> vim silly.sh

#!/usr/bin/env bash

#BATCH --mail-type=END

#SBATCH --mail-user=devjanee.swain.lenz@duke.edu

##SBATCH --array=1-2

#SBATCH --mem=10G

S1=silly"$SLURM\_ARRAY\_TASK\_ID".txt

S2=silly"$SLURM\_ARRAY\_TASK\_ID".txt

module load R

Rscript silly.R $S1 $S2