

Graduate School Class Reminders

- ▶ Maintain six feet of distancing
- ▶ Please sit in the same chair each class time
- ▶ Observe entry/exit doors as marked
- ▶ Use hand sanitizer when you enter/exit the classroom
- ▶ Use a disinfectant wipe/spray to wipe down your learning space before and after class
- ▶ Media Services: 414 955-4357 option 2

Documentation on the web

- ▶ CRAN: <http://cran.r-project.org>
- ▶ R manuals: <https://cran.r-project.org/manuals.html>
- ▶ SAS: <http://support.sas.com/documentation>
- ▶ Step-by-Step Programming with Base SAS 9.4 (SbS):
<https://documentation.sas.com/api/docsets/basess/9.4/content/basess.pdf>
- ▶ SAS 9.4 Programmer's Guide: Essentials (PGE):
<https://documentation.sas.com/api/docsets/lepg/9.4/content/lepg.pdf>
- ▶ Wiki: <https://wiki.biostat.mcw.edu> (MCW/VPN)

R debugging tips: common pitfalls

- ▶ **subsetting a matrix with a variable length index:**
when resolving to one column (or row) returns a vector
and your code is expecting a matrix to be returned
counter-measures: when returning a subset of columns (rows),
encase within `cbind` (`rbind`) which creates a matrix if needed
- ▶ **unintentional backward for loops**

```
L=length(cuts) ## a list
xinfo=rbind(cuts[[1]]) ## creating a matrix
## NEXT LINE IS WRONG
## for(i in 2:L) xinfo=rbind(xinfo, cuts[[i]])
## IF LENGTH IS ONE: THE ABOVE LOOP RUNS BACKWARDS
## NEXT LINE IS RIGHT: DEFENSIVE PROGRAMMING
if(L>1) for(i in 2:L) xinfo=rbind(xinfo, cuts[[i]])
```
- ▶ logical subset of a vector with 0/1 rather than FALSE/TRUE
use which religiously to avoid this *error-less* hard to find bug
- ▶ typos: `==` in comparisons vs. `=` in assignments
- ▶ scalar `&&/||` vs. vector `&/|` (all/any might be of help)
- ▶ similarly, scalar `min/max` vs. vector `pmin/pmax`

Building and installing R packages: CRAN

- ▶ The Comprehensive R Archive Network (CRAN) has 16335 R add-on packages as of this writing there will be more by the time you read this
- ▶ On gouda, we have 14449 installed mainly from CRAN
- ▶ To install an R package from CRAN
The two most reliable, and likely complete, nearby mirrors are <http://lib.stat.cmu.edu/R/CRAN> at Carnegie-Mellon and <http://cran.wustl.edu> at Washington University in St.L.
N.B. [http](http://) NOT [https](https://)

```
> options(repos=c(CRAN="http://lib.stat.cmu.edu/R/CRAN"))  
> install.packages("remotes", dependencies=TRUE)
```

To install all CRAN packages (takes hours: we run this over-night)

```
> install.packages(available.packages()[ , 1])
```

Some of them will fail for missing system dependencies like device drivers, required software, etc.

Building and installing R packages: Bioconductor

- ▶ The Bioconductor Project produces R packages for bioinformatics: <http://bioconductor.org>
- ▶ Bioconductor versions are tied to specific R versions
> `tools:::.BioC_version_associated_with_R_version()`
returns '3.10' with R 3.6.2
- ▶ To install the package named `limma` (and R or Bioconductor package dependencies, if any)
> `source("http://bioconductor.org/biocLite.R")`
> `biocLite("limma")`
- ▶ To install all Bioconductor packages (takes a while):
> `biocLite(all_group())`

Building and installing R packages: command line

- ▶ You can build and install R packages from the command line
- ▶ This works with your own R packages or those of others
- ▶ If it is your own in the sub-directory PACKAGE, then build it:
 > R CMD build PACKAGE
- ▶ For others, download the archive of source files
 either a tar file ending in .tar.gz or .tgz
 or a ZIP file .zip
- ▶ Unpack it: > tar xzf TARFILE or > unzip ZIPFILE
 which should create the PACKAGE sub-directory
- ▶ You don't have to change directories
- ▶ Build the package: > R CMD build PACKAGE
- ▶ Sometimes the vignettes crash the build or take a long time
 > R CMD build --no-build-vignettes PACKAGE
- ▶ So now you have created PACKAGE.VERSION.tar.gz
- ▶ Install it: > R CMD INSTALL PACKAGE.VERSION.tar.gz
- ▶ And you can remove it later: > R CMD REMOVE PACKAGE

Building and installing R packages: remotes package

- ▶ You can build and install R packages from anywhere on the internet with the remotes package
- ▶ For example, former CRAN packages that have been Archived:
<https://cran.r-project.org/src/contrib/Archive>
- ▶ These can be installed with the `install_url` function
- ▶ Or R packages on <https://github.com>
- ▶ These can be installed with the `install_github` function
- ▶ However, R 3.6.2 or higher appears to be necessary
- ▶ And installing from the command line seems to be much faster
- ▶ Typically, there are also Orphaned packages, however, currently there are none (as of this writing)
<https://cran.r-project.org/src/contrib/Orphaned>

HW Hands-on: Building and installing R packages

- ▶ Let's install the BART3 R package from `https://github.com/rsparapa/bnptools`
 - ▶ with the `install_github` function
 - ▶ and with the command line using `git`
 - ▶ create your own sub-directory: `> mkdir ~/git; cd ~/git`
- ```
> git clone https://github.com/rsparapa/bnptools.git
> cd bnptools
```