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`
- SAS 9.3: `https://support.sas.com/en/documentation/documentation-for-SAS-93-and-earlier.html`
- 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 and GCC upgrade

- https://wiki.biostat.mcw.edu/Running_R_on_Clusters

```
#!/bin/bash
## ~/emacs-26.3 shell script
(module load gcc/9.2 emacs/26.3 R/3.6.2; emacs "$@")
```

Advanced SAS Macros

- We are going to look at some exemplars from the RASmacro library for inspiration
- It is located locally at `/usr/local/sasmacro`
- And it is on github (but it needs to be updated)
- These macros will give us ideas for how to use SAS macros, how to write them and what further can be done with them
- In particular, *hardening* of macros to quotes, embedded ampersands, etc.

Advanced SAS Macros: Building Blocks

Name	Description
`_abend.sas`	Abend the SAS program
`_count.sas`	Number of items in a list
`_dsexist.sas`	Does a data set exist?
`_exist.sas`	Does a file exist?
`_fn.sas`	Create footnotes with the SAS program name
`_level.sas`	Create a macro value from a data set var.
`_list.sas`	Create a list: more general than a var. list
`_nobs.sas`	Number of obs. in a data set
`_pdfjam.sas`	Combine ODS graphic files
`_require.sas`	Is a required argument present?
`_retain.sas`	RETAIN with BY-group processing
`_sort.sas`	Sort a data set if necessary
`_substr.sas`	Smart sub-string function
`_unwind.sas`	OS-specific statements: UNIX/Linux vs. Windows
`_vorder.sas`	Order the var. in a data set

Advanced SAS Macros: Higher-level

Name	Description
`_dropobs.sas`	Drop obs. of missing variables
`_dropvar.sas`	Drop var. that's always missing/constant
`_summary.sas`	Flexible tables with the statistics that you want: correctness vs. prettiness

HW part 1: SAS macro for hot-decking and the NTDB

- Write a SAS macro to perform hot-decking for the NTDB example: see the details in lecture 4, slide 4
- Hints: for random number generation from the Uniform distribution, use the `rand("unif")` and `call streaminit(SEED);` to set the seed. This seed is an argument to the macro, `seed=REQUIRED`.
- Other arguments are an input data set, `data=REQUIRED`, and an output data set, `out=REQUIRED`.
- Previous hints from the HW in lecture 13 still apply.
- Use arrays with the `array` statement
- Use the `set` statement with the `point` option to iterate through the data set BEWARE: with `point`, don't forget the `stop; run;` at the end of the DATASTEP
- With many functions you can use an `of` clause with a variable list, e.g., `nmiss(of VAR1-VARn)` instead of `nmiss(VAR1, ..., VARn)`
- Or with the STARTS-WITH colon operator `nmiss(of VAR:)`
- Use the `%_nobs()` macro to determine how many observations

HW part 2: stratified random sampling macro for the NTDB data set

- Write a SAS macro to perform stratified random sampling: see the details in lecture 4, slide 7
- Hints: use the `call _permute` subroutine to create permutations. Use `call streaminit(SEED);` to set the seed. This seed is an argument to the macro, `seed=REQUIRED`.
- Other arguments are an input data set, `data=REQUIRED`, an output data set, `out=REQUIRED`, and a variable list to be kept, `var=REQUIRED`.
- Previous hints from the HW in lecture 13 still apply.
- A variable list can be used in many functions with the `of` clause like `of VAR1-VARn`