

Statistical Computing

- Instructor:** Steve Kachman
Office: 354B Hardin Hall
Phone: 472-7302
Office Hours: Immediately after class
- Grading:**
- | | |
|-----------------|-----|
| Labs | 35% |
| Midterm Project | 25% |
| Presentation | 15% |
| Final Project | 25% |
- A final average of 90% will guarantee an A-, 80% a B-, 70% a C- and 60% a D-.
- Available at <http://my.unl.edu/>
- Labs:** There will be approximately one lab assignment a week.
- Project:** Each student will complete a midterm and final project.
- Presentation:** Each student will give a one hour presentation on their project. The presentations will be during the last week of classes.
- Final Exam:** The final is scheduled for **Wednesday, May 7 from 10:00 - Noon**. Final reports on final projects will be due at this time.
- Disabilities:** Students with disabilities are encouraged to contact the instructor for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Nebraska-Lincoln to provide flexible and individualized accommodation to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, students must be registered with the Services for Students with Disabilities (SSD) office, 132 Canfield Administration, 472-3787 voice or TTY.
- Web page:** <http://statistics.unl.edu/faculty/steve/cpp/2008/>

The purpose of this course is to introduce the use of object oriented programming concepts of C++ for developing statistical software. In part 1 of the course, students will learn object oriented programming concepts by creating new C++ classes that can be used for mixed linear model analyses. In part 2, students will learn how to use existing C++ classes to handle more complex models.

TENTATIVE COURSE OUTLINE

Principles for efficient construction of normal equations for main-effects models:

- One-way model
- Two-way model
- Models with covariates

A C++ class (Recoder) will be introduced to sequentially number levels of factors such as herd and animal. The Recoder class will be used to: 1) build normal equations without having to pre-process data files, and 2) include interactions in models.

Modify normal equations to obtain the mixed model equations. A C++ class (Pedigree) will be introduced to construct the inverse of the additive relationship matrix \mathbf{A} . The Pedigree class uses the Recoder, and so the animals in the pedigree can be identified using an alpha-numeric system.

Models with maternal effects and multi-trait models.

Develop a C++ class (MME) to build the mixed model equations from a description of the model. Initially, the model will be described in terms of a vector of “model-term” objects.

Introduce methods (functions) to construct vector of model-term objects from a model string such as:

“y1 = breed + directAdditive; y2 = breed + directAdditive;”.

Use of iterative solvers of mixed model equations

Use of C++ classes from Matvec for non-linear models

Use of C++ classes for marker assisted selection.