APSTA-GE-2352 Practicum in Applied Statistics: Statistical Computing

Klint Kanopka

Fall 2025-2026

1 Course Description

This course will introduce the student to statistical programming and simulation using R. Students will first understand variables, data structures, program flow (e.g., conditional execution, looping) and functional programming, then apply these skills to answer interesting statistical questions involving the comparison of groups. Most statistical analysis will be motivated via simulations, rather than mathematical theory. The course content (programming and data analysis) requires significant outside reading and programming.

1.1 Additional Notes

This course is designed to treat `R` *as a programming language*[3]. Programming skills are taught through the analysis, design, and implementation of algorithms, with special attention paid to modern statistical algorithms (e.g., gradient descent, k-Means clustering, Markov Chain Monte Carlo). Time will also be spent on optimization techniques (e.g., vectorization and parallelization) useful for statistical analysis. Everything done in this course will be implemented in base `R`. Notable exceptions include the use of the `ggplot2` library for visualization and libraries like `MASS` for sampling from distributions[5, 4]. Students must implement algorithms, write unit tests, and debug almost exclusively using the tools available in base `R`.

1.2 Prerequisites

This course assumes some experience with the `R` programming language and probability. You may find prior experience with computer science fundamentals and `ggplot2` to be helpful. No previous exposure to the design and analysis of algorithms is assumed.

2 Student Learning Outcomes

During the course of the semester:

1. Students will implement literate programming to produce coherent and reproducible code.
2. Students will verify code function through the implementation of unit tests.
3. Students will write more efficient code by applying optimization techniques (e.g., vectorization, parallelization).
4. Students will solve problems by implementing and modifying algorithms.
5. Students will answer statistical questions by implementing Monte Carlo simulations.

3 Course Logistics

3.1 Instructors

- **Klint Kanopka** - Instructor, Lecture
  - klint.kanopka@nyu.edu
  - Office Hours: Tuesdays, 2-3p
    Location: Kimball Hall, Rm 205W
  - Appointments also available
- **Ruiting Shen** - TA, Lab Section
  - rs84228@nyu.edu
  - Office Hours: Wednesdays, 9-10a
    Location: Kimball Hall, 3E Lounge

3.2 Meeting Times and Locations

Attendance at in-person lectures is required. Students are to attend their assigned lab section unless prior permission is received from the teaching team. Attendance is taken at lab.

- **Lecture:** Thursdays, 4.55-6.35p @ 60 5th Ave, Rm C10
- **Lab Section:** Wednesdays, 3.45-4.35p @ 60 5th Ave, Rm C03

3.3 Required and Recommended Reading

- Required: *Introduction to Algorithms, Fourth Edition* [2]
- Required: *R for Data Science, Second Edition*[6]
- Recommended: *The R Inferno*[1]

- Recommended: *Comparing Groups: Randomization and Bootstrap Methods Using R* [7]

Both texts are available online at no cost through the NYU library.

4 Grading

This is a three-credit class. In exceptional circumstances with instructor approval, it may be taken for one credit. Your graded output for this course consists of eight (8) equally weighted problem sets and your score on a comprehensive final exam. Each problem set is typically worth 100 points. Table 1 contains the proportion of the final grade coming from each assignment category. Table 2 contains the transformation from the weighted proportion of earned points to letter grades. When transforming to a letter grade category, weighted proportions of earned points are **not** rounded.

Category	p
Problem Sets	0.7
Final Exam	0.3

Table 1: Final grade weighting scheme

	G^-	G	G^+
A	$[.895, .945)$	$[.945, 1]$	—
B	$[.795, .825)$	$[.825, .865)$	$[.865, .895)$
C	$[.695, .725)$	$[.725, .765)$	$[.765, .795)$
D	$[.600, .640)$	$[.640, .670)$	$[.670, .695)$
F	—	$[0, .600)$	—

Table 2: Grading Scale

4.1 Problem Sets

The course contains eight equally weighted problem sets (PS0-PS7).

4.1.1 Submission Guidelines

All problem sets should be submitted on Brightspace using Quarto and adhere to the following conventions:

1. **Submit assignments by the deadline.** Assignment deadlines are typically **Thursdays before lecture begins** (i.e., 4.54p). Late assignments will receive a 10% penalty per day.

2. **Name your files correctly.** For each assignment, name your files using the convention: `LastName_FirstInitial_PS#.ext`

3. **Put your name and the assignment in text of the file.** These should occupy the author and title fields, respectively.

4. **Submit both a source file and a knitted file.** Source files have the extension `.qmd` or `.Rmd`. Files should be knitted to `.pdf`. As an example for the first homework, I would submit two files:

   - `Kanopka_K_PS0.qmd`
   - `Kanopka_K_PS0.pdf`

   Note that you will need a functional LaTeX installation on your machine in order to knit to `.pdf`.

5. **Your source file should run without modification.** If you load a file, please use relative paths. Do not load libraries unless explicitly requested in an assignment.

6. **Do not use `install.packages()` calls in your assignments.** Please install packages locally and then load them in your assignment source code.

4.2 Final Exam

This course contains a written final exam, administered after week 14 during the university's scheduled final exam week. This comprehensive exam will cover material from the entire course. Students will be required to write `R` code *by hand* and be provided with a list of function names and arguments.

4.3 Extra Credit

This course allows extra credit to be earned in any of the following three ways:[1]

1. The first person to report typographical errors or mistakes in any of my course materials will receive one (1) extra credit point per error (I expect there to be a lot of them). These should be reported in your course section's Slack channel.

2. "Valuable suggestions" will also result in one (1) extra credit point and acknowledgment in a footnote.

3. Reporting a coding error with a solution or a valuable suggestion that requires additional labor (code snippets, a figure, a dataset, etc.) will be worth more, up to a maximum of 2^8 points. This amount will be scaled based on the nature and duration of required labor.

[1] This policy is inspired by the Knuth Reward Check

5 Course Policies

5.1 Academic Integrity

I take academic integrity incredibly seriously. Please review NYU Steinhardt's Academic Policies and Procedures for more information on specific policies, the disciplinary process, and sanctions.

5.2 Collaboration

I **strongly** encourage students to form study groups. Students may discuss and work on problem sets in groups. Each student must (1) report at the top of each question what other students they consulted with, (2) write their code and solution independently, and (3) understand their work well enough in order to reconstruct it entirely on their own.

5.3 AI Tool Policy

All assignments should be your own original work, but students should feel comfortable consulting AI tools to support their work. The expectation is that students will write their own software implementations, interpret their results, and develop their own narrative responses, but may use AI tools to help with tasks like debugging and refining their grammar. If you have a question about a specific tool or use case, please reach out to me.

5.4 Students with Disabilities

Students with physical or learning disabilities are required to register with the Moses Center for Student Accessibility, 726 Broadway, 2nd Floor, (212-998-4980 and online at `http://www.nyu.edu/csd`). They must present a letter from the Center to the instructor at the start of the semester to be considered for appropriate accommodation.

5.5 Mental Health Statement

If you are experiencing undue personal and/or academic stress during the semester that may be interfering with your ability to perform academically, the NYU Wellness Exchange (212-443-9999) offers a range of services to assist and support you. I am available to speak with you about stresses related to your work in my course, and I can assist you in connecting with the Wellness Exchange. Additionally, if you anticipate any challenges with completing the assignments, readings, exams and other work required in this course, I encourage you to register with the Moses Center (212-998-4980) in advance so that you may be granted the proper academic accommodations.

5.6 Inclusion

NYU values an inclusive and equitable environment for all our students. I hope to foster a sense of community in this class and consider it a place where individuals of all backgrounds, beliefs, ethnicities, national origins, gender identities, sexual orientations, religious and political affiliations, and abilities will be treated with respect. I intend that all students' learning needs be addressed in and out of class and that the diversity students bring to this class be viewed as a resource and strength. Please feel free to speak with me if this standard is not being upheld.

6 Course Calendar

Table 3 contains topics, assignments, and deadlines. Note that this is subject to change.

Week	**Lab (Wednesday)**	**Lecture (Thursday)**	**Assignments**
1: 9.4	Topics: • No lab	Topics: • Data types • Environments • Data manipulation with `dplyr` • Visualizing data with `ggplot2`	Released: • PS0 • PS1
2: 9.11	Topics: • Working with Quarto documents • Practice with `dplyr` tools • Practice with `ggplot2` visualizations	Topics: • Vector and matrix Arithmetic • Logical statements • Functions • Object oriented programming • Unit Testing	Due: • PS0
3: 9.18	Topics: • Code style • Writing test functions • Recovering distributions	Topics: • Indexing • Control flow • Conditional statements • `for` loops • Sorting algorithms	Due: • PS1 Released: • PS2
4: 9.25	Topics: • Evaluating logical statements • Debugging and unit testing	Topics: • Distance metrics • `while` loops • The k-Means clustering algorithm • Advanced debugging	

5: 10.2	Topics: • Selection Sort • Function composition • Backwards design for code	Topics: • Randomization and sampling • Monte Carlo methods • Writing simulations • Buffon's Needle	Due: • PS2 Released: • PS3
6: 10.9	Topics: • Building simulation studies • Nested loops • Constructing plots	Topics: • Advanced matrix computation • Dimensionality reduction • Principal component analysis • Power iteration	
7: 10.16	Topics: • Monte Carlo simulations • `R` tricks for simulation studies	Topics: • Numerical optimization • Writing loss functions • Gradient descent • Using `optim()`	Due: • PS3 Released: • PS4
8: 10.23	Topics: • Interpreting and visualizing PCA	Topics: • Regularization • The Bootstrap	
9: 10.30	Topics: • Solving problems with `optim()`	Topics: • Randomized algorithms • Non-convex optimization • Markov Chain Monte Carlo	Due: • PS4 Released: • PS5

10: 11.6	Topics: • Regularization	Topics: • Matrix factorization • The Singular Value Decomposition (SVD) • Quantification of text • Similarity metrics for text • Latent semantic analysis	
11: 11.13	Topics: • Advanced debugging	Topics: • The `apply()` family • Linear Programming • Convex Programming	Due: • PS5 Released: • PS6
12: 11.20	Topics: • `apply()` family practice	Topics: • Computer architecture • Vectorization • Parallelization • Space and time complexity	
13: 12.4	Topics: • Parallelization practice	Topics: • Model comparison • Out of sample evaluation • Cross validation	Due: • PS6 Released: • PS7
14: 12.11	Topics: • Final exam review	Topics: • High-performance computing • Next steps in computing	Due: • PS7 (12.15 @11.59p)

Table 3: Course Calendar

References

[1] P. Burns. *The R inferno*. 2011. URL https://www.burns-stat.com/documents/books/the-r-inferno/.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2022.

[3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL https://www.R-project.org/.

[4] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL https://www.stats.ox.ac.uk/pub/MASS4/. ISBN 0-387-95457-0.

[5] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

[6] H. Wickham, M. Cetinkaya-Rundel, and G. Grolemund. *R for data science*, volume 2. O'Reilly Sebastopol, 2017. URL https://r4ds.hadley.nz/.

[7] A. S. Zieffler, J. R. Harring, and J. D. Long. *Comparing groups: Randomization and bootstrap methods using R*. John Wiley & Sons, 2011.