Data Processing, Analysis, and Visualization with R

Daniel Anderson, University of Oregon - College of Education

R is a freely available and open source computing environment for statistical programming and data visualization that has tremendous advantages for research and dissemination of empirical findings. As an open-source program, the code for all functions are available. This means you do not have to blindly trust that the software is doing what you intend. Rather, you can look beneath the hood a bit, if you choose, and see exactly how the analysis is being conducted by examining the underlying functions. Countless packages have been developed for R, which are all also freely available, helping make complex analyses within specific contexts more possible. Further, as we discuss at multiple points throughout the class, R lends itself to reproducible research – meaning that others can view all the procedures taken in any analysis and reproduce the results exactly. Reproducible research facilitates transparency and development of research communities, with interested users sharing analytic procedures and outcomes. Readers of reports also do not have to guess at any intermediary steps taken during the analysis, if the source code is available. Finally, R is immensely flexible. Dr. Simon Blomberg, an evolutionary biologist and R programmer, once replied to an R user who was asking "if" a specific process was possible with R by stating that "This is R. There is no if. Only how". Once users move beyond adopting functions written by others, and begin programming independently, there are no limits to what can be done. This flexibility in programming also leads to the capability of automating many procedures, leading to substantial gains in efficiency. For example, a suite of functions can be written to move a dataset from its raw to analytic form and then applied to a set of common datasets (e.g., divided by a categorical variable). Functions for analysis and visualization could also be embedded, with all datasets processed simultaneously. In sum, R facilitates transparency in research while increasing efficiency and flexibility. Programming with R represents a substantial shift from point-and-click interfaces, but also comes with considerable advantages. This shift may even change the way you think about data.

The overall purpose of this course is to provide a basic foundation in programming with R. The course has three main components, as the title suggests: data processing/munging/wrangling, basic analysis, and visualization. These three components are routine in applied data analysis work. The focus is on working with and programming in R, as opposed to the specifics of any given dataset/analysis. One should not expect to leave the course with a deep understanding of any specific analysis. Rather, students should expect to leave with a deep understanding of statistical programming in R. Students are expected to have a basic understanding of statistics prior to taking the course, but high-level statistics courses are not a prerequisite.

The course begins in Week 1 with a basic overview of the R syntax and data structures, as well as working with *R Markdown* to help facilitate reproducible research. In Week 2, we work through an applied example, using a process/plot/analyze/plot procedure that is emphasized throughout the course. We then dig deeper into data structures in Week 3 before moving to functions in Week 4 and loops in Week 5. In Week 6 we discuss string manipulations, before moving to advanced plotting features in Week 7. We discuss very basic multilevel models and visualizations in Week 8, and walk through a second (and more complicated) complete, applied example. In Week 9, we discuss the basics of writing R packages, before moving to batch processing and a final review in Week 10.

Throughout the course, we apply the topics of the week to help process "messy" data into workable formats, conduct basic analyses, and visualize the data. We use data visualization to explore both raw data and the results of fitted models. We work very basically with *git*, which is a version control software that can help when collaborating on code. This class is constructed with the philosophy that the only way to truly learn R and become proficient with it, is to dive in and practice, practice, practice. Class sessions are highly interactive. During class, you are asked to write code with the instructor concurrently, independently other times, and in small groups at other times. Weekly homeworks are assigned, which are intended to be brief but get you more practice. These can be completed independently or in small groups, and are scored on a completion basis. The course also includes a term project, which requires you to complete the three main components of the course described above with a "real" dataset. This can, again, be completed independently or in small groups. You are encouraged to find data that you are actually interested in using. If you do not have access to data, please contact the instructor as soon as possible.

Required Materials

Prior to the first class you should have the latest version of R downloaded and installed on your computer, available through the comprehensive R archive network (CRAN): https://cran.r-project.org. We work with various packages as well, but these can be installed as we work through the material. All readings are provided as PDFs or are freely available online. Please also download and install *git*: https://git-scm.com/downloads. You are expected to clone the entire course repository with all the materials (you should have received a tutorial describing all the steps fore cloning). If you have trouble with cloning the course repository prior to the first class, please contact the instructor to help you troubleshoot outside of normal class hours. You are encouraged you to create a *github* repository (https://github.com) where you can store all your work for this class and get additional practice with git.

Recommended Materials: Prior to the first class, it is highly recommended that you spend some time exploring different text editors and/or integrated development enviornments (IDEs) for R. The instructor of the course uses the text editor Sublime Text (http://www.sublimetext.com) which has some really nice features and is very aesthetically pleasing. By far the most popular IDE is RStudio, which is being actively developed and has numerous plugin-like features that make many tasks simpler. Most of this is just personal preference. The built in editor for Mac is also pretty nice, but on the Windows side is about as basic as can be. Getting a text editor/environment that you feel comfortable with can really ease the process.

Resources: One of the great things about R is that it is open-source. Like many open-source platforms, there is a tremendous community behind it. Online forums can be a great place to find answers to specific problems (e.g., http://stackoverflow.com/questions/tagged/r). There are also lots of sites that give basic tutorials (e.g., http://www.statmethods.net/interface/help.html). UCLA has also listed a set of resources that may be helpful (http://www.ats.ucla.edu/stat/r/). In this course, we primarily rely on the following resources:

- Chacon, S., and Straub, B. (2014). *Pro Git: Everything you need to know about git.* (2nd edition). Available online at https://git-scm.com/book/en/v2
- Gelman, A., and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models.* New York, NY: Cambridge University Press.
- Kabacoff, R, I. (2011). *R in Action: Data analysis and graphics with R.* Shelter Island, NY: Manning Publications.
- Peng, R. D. (2015). *R Programming for Data Science.* Victoria, BC: Lean Publishing.
- Sanchez, G. (2013). *Handling and Processing Strings in R.* Trowchez Editions: Berkely. (http://www.gastonsanchez.com/Handling_and_Processing_Strings_in_R.pdf)
- Wickham, H. (2015). *Advanced R.* Available online at http://adv-r.had.co.nz
- Wickham, H. (2010). *ggplot2.* New York, NY: Springer.

Course Website

All the code for this course, as well as lectures and course readings, are available in a github repository (https://github.com/DJAnderson07/CourseR). This is where the basic work with git comes in, as the course requires you clone the repository onto your local machine and update it periodically as new content is pushed into the repository.

Course Objectives

Upon completion of this course, the successful student will:

1. Understand the foundations of the R environment (i.e., OOP, data structures, functions, loops, etc.).
2. Understand and be able to work efficiently with different data types (i.e., string, integer, double, factor).
3. Be able to read various data into R from various sources (e.g., .csv, .sav, .txt, etc.) and prep the data for analysis.
4. Understand the basic structure of plotting with the base graphics and the *ggplot2* package.
5. Be able to conduct basic analyses (e.g., simple and multiple linear regression, basic varying intercept models, etc.) with R, and explore the fitted model through visualizations.
6. Understand the components of functions and be able to develop custom generic functions for a variety of purposes.
7. Understand and be able to apply loops with functions.

Course Schedule

Before the first class

- Decide on a text editor or IDE for R
- Clone course github repository (see git tutorials)

Week 1.1: Getting Started Introduction and overview of the R environment

- Objectives
  - Understand vectors, matrices, and subsetting each
- Topics
  - Objects in R
  - R packages
  - Getting help
  - Vectors and matrices
  - The grammar of syntax
  - Subsetting vectors and matrices
- Lab
  - Subsetting
- Readings
  - Syntax style: http://adv-r.had.co.nz/Style.html
  - Kabacoff: Chapter 1
  - Chacon: Chapters 1 & 2

Week 1.2: *R Markdown* and reading/writing data into/from R

- Topics
  - Introduction to *R Markdown*
  - Directory management
  - Reading data into R
  - Subsetting data frames
  - Writing data

- Lab
  - R Markdown, and reading data
- Readings
  - R Markdown
    - http://RMarkdown.rstudio.com
    - http://RMarkdown.rstudio.com/authoring_basics.html
    - http://RMarkdown.rstudio.com/authoring_rcodechunks.html
- **Homework 1 Assigned**
  - Creating an *R Markdown* document

Week 2.1: A complete, applied example

- Topics
  - Process/Plot/Analyze/Plot procedure
  - Analyze data
    - `cor()`
    - `lm()`
- Lab
  - Full example
- Readings
  - Gelman and Hill, Chapter 3

Week 2.2: Basic plotting

- Histograms and density plots
- Scatter plots
  - Controls:
    - titles
    - line width, color, and type
    - point size, color, and type
    - x and y axis labels
- Lab
  - Basic plots

- Readings
  - UCLA Tutorial: http://www.ats.ucla.edu/stat/r/pages/introduction.htm
- **Homework 1 Due**
- **Homework 2 Assigned**
  - A complete, applied example

Week 3.1: Data structures and data types

- Topics
  - Vectors, Matrices, and (briefly) Arrays
  - Data types: Logical, Integer, Double, Character
  - Coercion
  - Attributes: Names, Dimensions, Custom
- Lab
  - Coercions
- Readings
  - Kabacoff: Chapter 2

Week 3.2: Data structures, classes, and routine functions

- Topics
  - Matrix algebra versus element-wise algebra
- Lists
- routine functions
  - `rep()`, `seq()`, `:`
  - `table()`
  - `c()`, `cbind()`, `rbind()`
  - `ifelse()`
  - `rnorm()`, `rbinom()`, `dnorm()`, etc.
  - `length()`, `nrow()`, `ncol()`
  - `str()`, `head()`, `tail()`
  - `summary()`
  - `with()`, `attach()`
  - etc.
- Lab
  - Applying routine functions and working with lists
- Readings
  - Wickham: http://adv-r.had.co.nz/Data-structures.html
  - Wickham: http://adv-r.had.co.nz/S3.html
- **Homework 2 Due**
- **Homework 3 Assigned**
  - Data Structures

Week 4.1: Functions, part 1

- Topics
  - Overview of functions
  - Function components
  - Primitive functions
  - Basics on writing functions
  - Using functions with plotting

- Lab
  - Writing custom functions
- Readings
  - Peng: Functions

Week 4.2: Functions, part 2

- Topics
  - Classes and methods (object-oriented programming)
  - Storing and applying functions with `source()`
  - Functional programming
  - Scoping
  - Infix functions (and every operator as a function)
- Lab
  - Writing more advanced functions
- Readings
  - Wickham (http://adv-r.had.co.nz/Functions.html)
- **Homework 3 Due**
- **Homework 4 Assigned**
  - Functions

Week 5.1: Loops, part 1

- Topics
  - For loops (generally the most useful)
  - While loops
  - Repeat loops
- Lab
  - Loops and functions
- Readings
  - http://blog.datacamp.com/tutorial-on-loops-in-r/
  - Peng: Control Structures

Week 5.2: Loops, part 2

- Topics
  - Apply family of loops
- Lab
  - Batch processes with lists and loops
- Readings
  - Wickham: http://adv-r.had.co.nz/Functionals.html
- **Final Project Outline Due**
- **Homework 4 Due**
- **Homework 5 Assigned**
  - Loops

Week 6.1: String manipulations

- Topics
  - Regular exrpessions
  - Base string functions
    - `paste()`
    - `grep()`
    - `substr()`
    - `nchar()`
  - `stringr` package
    - Padding, Wrapping, and Trimming
    - Word extraction
    - Detecting patterns
    - Extracting patterns
      - all, first, or last matches
- Lab
  - Analyzing strings & creating new variables
- Readings
  - Sanchez: Chapters 3 and 5

Week 6.2: Data restructuring with strings, loops, and functions

- Topics
  - Merging
    - Merging as a tool for producing new variables
    - Create index for merging based on strings
  - Aggregation
  - Reshaping data
    - Long to wide and wide to long

- Lab
  - Data restrucuring
- **Homework 5 Due**
- **Homework 6 Assigned**
  - String manipulations and data restructuring

Week 7.1: Advanced plotting with base graphics

- Topics
  - Graphical Parameters: `par()` and `layout()`
    - `mfrow`, `mfcol`, `mar`, `oma`
  - Low-level functions
  - `points`, `lines`, `text`, `axis`, `polygon`, `curve`
  - Applying loops and functions to plots
  - Saving plots
- Lab
  - Overlaying plots
- Readings
  - http://rpubs.com/SusanEJohnston/7953
  - http://www.ling.upenn.edu/~joseff/rstudy/week4.html

Week 7.2: The `ggplot2` package

- Topics
  - `qplot()` (similar to base)
  - layering with `ggplot()`
  - faceting
- Lab
  - Cluster-level visualizations
- Readings
  - Wickham (ggplot book): Chapter 2 & 3 (especially 3)
- **Homework 6 Due**
- **Homework 7 Assigned**
  - Plotting with loops and functions

Week 8.1: A few basic multilevel models (and visualizations)

- Topics
  - lme4 and the `lmer()` function
  - Varying intercepts
  - Varying slopes
  - Varying intercepts and slopes
- Lab
  - Fitting, visualizing, and interpreting varying intercept models
- Readings
  - Gelman & Hill: Chapter 12

Week 8.2: A second complete example

- Topics
  - Process/Plot/Analyze/Plot procedure
  - Applying string manipulations
  - Applying functions to data manipulations
  - Applying loops to data manipulations
  - Plotting with functions and loops
  - Analysis - varying intercepts and slopes multilevel model
  - Plotting with `ggplot2`
- Lab
  - A complete, applied example with varying intercepts
- **Homework 7 Due**
- **Homework 8 Assigned**
  - A second complete applied example

Week 9.1: Writing R packages, part 1

- Topics
  - `devtools` and `roxygen2` packages
  - Package components
    - Code (R/)
    - Metadata (DESCRIPTION)
    - Object documentation (man/)
    - Namespaces (NAMESPACE)
    - Data (data/)
- Lab
  - Creating a basic package
- Readings
  - Parker: Writing R Packages (http://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/)

Week 9.2: Writing R packages, part 2

- Topics
  - Developing packages from base
  - Classes and methods (object-oriented programming)
  - tests/, src/, exec/, inst/
- Lab
  - Developing a package for estimating and visualizing means across groups
- Readings
  - Leisch, 2009: https://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf
- **Homework 8 Due**
- **Homework 9 Assigned**
  - Writing R packages

Week 10.1: Batch processing

- Topics
  - Reading all files of a specific type from a directory
  - Applying functions and loops to batch process the datasets
  - Applying loops to batch analyze datasets
  - Running R in batch mode
  - .bat files (and their potential use in packages)
- Lab
  - Batch processing
- Readings

Week 10.2: Overview and review

- Topics
  - Object-oriented programming
  - Data structures
  - String manipulations
  - Functions and loops
  - Writing R packages
  - Questions on final project
- **Homework 9 Due**
- Lab
  - R Overview

Week 11: No class, finals week

- **Final Project Due**

Homeworks

Homeworks are worth 10 points each (90 points total), and are graded on a completion basis. As long as you address each part of each homework, full credit is awarded (regardless of correctness). Below is a brief overview of each homework. More details are given at the time the homework is assigned. All homeworks are due one week after the day they are assigned.

1. Creating an *R Markdown* document
    - The purpose of this homework is to get you some basic familiarity with *R Markdown*. You are required to produce a document with some of the basic features of *R Markdown*.
2. A complete, applied, example
    - The purpose of this homework is to have you gain some very basic experience with the process/plot/analyze/plot procedure.
3. Data structures
    - This homework asks you to build upon Homework 2 with more of a focus on data structures and data types.
4. Functions
    - This homework focuses on writing custom functions to perform specific tasks.
5. Looping
    - This homework focuses on using various types of loops (specifically, for loops and the apply family of loops) with functions to various data structures.
6. String manipulations
    - This homework asks you to apply base functionss with regular expressions, as well as functions from the `stringr` package. You are also asked to make data transformations based on the previous string manipulations.
7. Plotting with loops and functions
    - This homeworks requires you apply functions and loops to generate plots with the base graphics. You should produce publication-level plots. You are also asked to produce exploratory plots with the `ggplot2` package.
8. A second complete, applied, example
    - The purpose of this homework is essentially the same as Homework 2, but the manipulations, plotting, and analysis requirements are more complicated. You are required to use functions and loops throughout the homework.
9. Writing R packages
    - This homework requires you to author a basic R package and provide function documentation.

Final Project (100 points)

For the final project in this class, you are required to complete the three major components of the class with a "real world" dataset. If you do not have access to data, please contact the instructor as soon as possible. More details are provided later in the term, but the project consists of the following steps: (a) processing the raw data to a format appropriate for analysis and exploratory plotting, (b) producing at least three exploratory plots of the analytic sample, (c) analyzing the data, and (d) plotting at least one aspect of the fitted model. You are also asked to interpret your model relative to the coefficients and plot(s) of the fitted model. Your plan must be approved by the instructor. The final project is due week 11 of the term (finals week), while a full description of your plan (i.e., project outline) is due week 5.

The final project must be produced in accordance with the principles of reproducible research. In other words, You are required to use *R Markdown* (or something similar, e.g., LaTeX). The final paper should not include any code in the report, but the code should be accessible and reproducible. You are encouraged to use *git* to make this all accesible and transparent. All of this should become more clear as the term progresses.

Grading

Points breakdown (300 possible)

- Participation: 60 points (20%)
  - 10 labs @ 5 points each
  - 10 points for general involvment (asking questions, coming prepared, etc.)
- Homeworks: 9 @ 10 points each = 90 points (30%)
- Final Project outline: 50 points (17%)
- Final Project: 100 points (33%)

Letter grade breakdown

- Below is a breakdown of letter grades, based on the number of points received.

Lower point range	Grade	Upper point range
$\geq$ 93% (279 points)	**A**	
$\geq$ 90% (270 points)	**A-**	$<$ 93% (279 points)
$\geq$ 87% (261 points)	**B+**	$<$ 90% (270 points)
$\geq$ 83% (249 points)	**B**	$<$ 87% (261 points)
$\geq$ 80% (240 points)	**B-**	$<$ 83% (249 points)
$\geq$ 77% (231 points)	**C+**	$<$ 80% (240 points)
$\geq$ 73% (219 points)	**C**	$<$ 77% (231 points)
$\geq$ 70% (210 points)	**C-**	$<$ 73% (219 points)
$<$ 70% (210 points)	**F**	

Course Policies

Attendence Policy

Attendance is required to succeed in this course and master the course material. If a student does miss class, it is the student's responsibility to get class notes, and handouts or other distributed materials. Contact the instructor in case of illness or emergencies that preclude completing assignments as scheduled or attending class sessions. Messages can be left on the instructor's voice mail or e-mail at any time of the day or night, prior to class. If no prior arrangements have been made before class time, the absence is unexcused.

Absence Policy

Students must contact the instructor in case of illness or emergencies that preclude attending class sessions or taking quizzes as scheduled. Messages can be left on the instructor's voice mail or e-mail at any time prior to class. If no prior arrangements have been made before class time, the absence is unexcused.

If you are unable to complete an assignment due to a personal and/or family emergency, you should contact your instructor or discussion leader as soon as possible. On a case-by-case basis, the instructor determines whether the emergency qualifies as an excused absence.

Academic Misconduct Policy

All students are subject to the regulations stipulated in the UO Student Conduct Code. See the following website: (http://uodos.uoregon.edu/StudentConductandCommunityStandards/AcademicMisconduct/tabid/248/Default.aspx). This code represents a compilation of important regulations, policies, and procedures pertaining to student life. It is intended to inform students of their rights and responsibilities during their association with this institution, and to provide general guidance for enforcing those regulations and policies essential to the educational and research missions of the University.

Conflict Resolution

Several options, both informal and formal, are available to resolve conflicts for students who believe they have been subjected to or have witnessed bias, unfairness, or other improper treatment. It is important to exhaust the administrative remedies available to you including discussing the conflict with the specific individual, contacting the Department Head, or within the College of Education, you can contact Joe Stevens, Associate Dean for Academic Affairs, at 346-2445 or stevensj@uoregon.edu or Surendra Subramani, Diversity Coordinator, at 346- 1472 or surendra@uoregon.edu.

Outside the College, you can contact:

- *UO Bias Response Team:* 346-1139 or http://bias.uoregon.edu/whatbrt.htm
- *Conflict Resolution Services:* 346-0617 or http://uodos.uoregon.edu/SupportandEducation/ConflictResolutionServices/tabid/134/Default.aspx
- *Affirmative Action and Equal Opportunity:* 346-3123 or http://aaeo.uoregon.edu/

Diversity

It is the policy of the University of Oregon to support and value diversity. To do so requires that we:

- Respect the dignity and essential worth of all individuals.
- Promote a culture of respect throughout the University community.
- Respect the privacy, property, and freedom of others.
- Reject bigotry, discrimination, violence, or intimidation of any kind.
- Practice personal and academic integrity and expect it from others.
- Promote the diversity of opinions, ideas and backgrounds, which is the lifeblood of the university.

Documented Disability

Appropriate accommodations are be provided for students with documented disabilities. If you have a documented disability and require accommodation, arrange to meet with the course instructor within the first two weeks of the term. The documentation of your disability must come in writing from the Disability Services in the Office of Academic Advising and Student Services. Disabilities may include (but are not limited to) neurological impairment, orthopedic impairment, traumatic brain injury, visual impairment, chronic medical conditions, emotional/psychological disabilities, hearing impairment, and learning disabilities. For more information on Disability Services, please see http://ds.uoregon.edu/.

Expected Classroom Behavior

Classroom expectations include:

- Participating in class activities.
- Respecting the diversity of cultures, opinions, viewpoints in the classroom.
- Listening to fellow students, professors, and lecturers with respect.
- Arriving on time, prepared for class.
- Attending for the duration of class.
- Not reading other materials, books, newspapers, or using laptops for other activities.
- Turn off cell phones and other electronic devices.
- Racist, homophobic, sexist, and other disrespectful comments are not tolerated.

Grievance

A student or group of students of the College of Education may appeal decisions or actions pertaining to admissions, programs, evaluation of performance and program retention and completion. Students who decide to file a grievance should follow the student grievance procedure, or alternative ways to file a grievance outlined in the Student Grievance Policy (http://education.uoregon.edu/feature.htm?id=399) or enter search: student grievance.

Inclement Weather

In the event the university operates on a curtailed schedule or closes, UO media relations notifise the Eugene-Springfield area radio and television stations as quickly as possible. In addition, a notice regarding the university's schedule is posted on the UO main home page (in the News section) at http://www.uoregon.edu. Additional information is available at http://hr.uoregon.edu/policy/weather.html.

If an individual class must be canceled due to inclement weather, illness, or other reason, a notice is posted via email. During periods of inclement weather, please check your email rather than contact department personnel. Due to unsafe travel conditions, departmental staff may be limited and unable to handle the volume of calls from you and others.