useR! 2020 Tutorials – Morning Session

First steps in spatial data handling and visualization

by S. Rochette, D. Scott and J. Nowosad

Spatial data analysis has long been one of R's strengths. R's spatial ecosystem allows for easy spatial data access, handling, visualization, and modelling. This tutorial will focus on getting started with the spatial data analysis in R by showing how to create maps and handle spatial data.The tutorial is designed for R users from a variety of fields who are interested in working with spatial data and creating maps with R. No knowledge of cartography is required. Prior knowledge of (non-spatial) graphic making and data handling with R is recommended.

Learning outcomes: By the end of the tutorial participants should:

- Be able to create static and interactive maps with popular packages such as `tmap`, `ggplot2`, and `leaflet`;
- Understand basic cartographic principles in map-making;
- Manipulate vector objects using the `sf` and `dplyr` packages;
- Understand and be able to apply map projections to maps.

Requirements: Attendees should already have experience with R and be able to:

- manipulate rectangular data with `dplyr`: `select`, `filter`, `mutate`, `group_by`, `summarize`;
- create graphs with `ggplot2`;
- work with Rmarkdown documents.

Attendees should install the necessary software and packages before the tutorial. This may be tricky for GNU Linux and MacOS. Please follow instructions here.

Predictive modeling with text using tidy data principles

by J. Silge and E. Hvitfeldt

Have you ever encountered text data and suspected there was useful insight latent within it but felt frustrated about how to find that insight?

Are you familiar with `dplyr` and `ggplot2`, and ready to learn how unstructured text data can be used for prediction within the tidyverse and tidymodels ecosystems?

Do you need a flexible framework for handling text data that allows you to engage in tasks from exploratory data analysis to supervised predictive modeling?

This tutorial is geared toward an R user with intermediate familiarity with R, RStudio, the basics of regression and classification modeling, and tidyverse packages such as `dplyr` and `ggplot2`. This person is comfortable with the main functions from `dplyr` and `ggplot2` and is now ready to learn how to analyze and model text using tidy data principles. This R user has some experience with statistical modeling (such as using `lm()` and `glm()`) for prediction and classification and wants to learn how to build models with text.

Learning outcomes: At the end of this tutorial, participants will understand how to:

- perform exploratory data analyses of text datasets, including summarization and data visualization;
- create flexible, appropriate features for modeling from raw text, using strategies such as tokenization and word embeddings;

- understand the pros and cons of such strategies and how knowledge of language can inform better modeling decisions;
- build supervised models (regression and classification) for text using tidy data principles;
- evaluate models to assess how they perform and which models are appropriate in specific circumstances.

Requirements: Participants will be expected to have laptops with a modern browser and internet access for the tutorial. The tutorial will use RStudio Cloud for hands-on exercises; attendees are encouraged to create an RStudio Cloud account ahead of time, using either GitHub, Google, or email.

This course will use direct instruction with examples from compelling, real-world datasets, live coding by the instructors illustrating each topic, and participant coding as well. Participants will be invited to synthesize the direct instruction material and make "next steps" on their own during independent and co-working time. For a similar (although not exactly the same) workshop, look here.

So, you want to learn Python? An introduction to Python for the R lover

by S. Ellis

R and Python are two of the most commonly used and fastest growing programming languages. Both are awesome, and the `reticulate` package in R makes moving between the two easier than ever. If you're an R lover who has wanted to learn Python, but who just hasn't had the time, this tutorial will give you that time! Examples used will be appropriate for a general R- and data-loving audience.

This tutorial would be ideal for someone familiar with R, RStudio, and R Markdown and who wants to be familiar with Python... but just hasn't gotten around to it yet. The tutorial will include R to Python translations while introducing and using the `reticulate` package.

Learning outcomes: At the end of this tutorial, participants will be able to:

- program using basic Python syntax (including variable assignment, conditionals, operators, and loops);
- read data into R using `reticulate` and pandas;
- import modules and run Python scripts from RStudio;
- carry out data wrangling tasks using pandas, generate data visualizations, and carry out basic analyses using `scikit-learn`.

Requirements: Participants should bring a laptop with Internet access and have an RStudio Cloud account. Expected level of audience's R background: The person who will get the most out of this tutorial will be a beginner-intermediate R user but a complete Python novice.

Participants should be comfortable with the following R concepts/packages:

- variable assignment;
- operators, conditionals, loops;
- data import into R;
- working with common data types (strings, factors, numerics);
- working with data frames/tibbles;
- packages: `ggplot2`, `dplyr`, `readr`.

Application of Gaussian graphical models to metabolomics

by D. Scholtens and R. Balasubramanian

This tutorial will identify a suite of tools in R for network analyses of high dimensional data. Examples will be utilize metabolomics data from ongoing large-scale health research studies, although techniques are transferrable across multiple domains. This workshop will link network visualizations with statistical analyses of high dimensional metabolomics data, and will emphasize detection of network subcomponents that link to health outcomes. Utility of these tools for 'story telling' in complex data settings will be illustrated.

Learning outcomes: At the end of this workshop, attendees will:

- gain exposure to the utility of network models in metabolomics and other genomic studies;
- apply R programming to analyze metabolomics and other omics data;
- visualize networks using R.

Requirements: This workshop would be most relevant for applied researchers involved in genomic studies in health research settings. Expected background of attendees includes:

- introductory level familiarity with R programming;
- introductory level familiarity with statistical concepts including correlation and association analyses;
- a basic understanding of metabolomics or other genomic data in which dependencies among assayed features are common;
- familiarity with methods for high dimensional data such as penalized regression and control of false discovery rates would be helpful for a deeper understanding of the models.

Attendees should have R installed on a laptop computer and suite of R packages (e.g. `igraph`, `glasso`) that will be communicated to attendees in advance of the workshop. Attendees should bring their own laptops with R >=3.5.1 and the following packages: `ggplot2` (>=3.0.0), `glasso` (>=1.10), `huge` (>=1.3.2), `iDINGO` (>=1.0.2), `igraph` (>=1.2.2). Data are not overly large so basic laptop functionality should suffice.

`Periscope` and `CanvasXpress` – Creating an enterprise-grade big-data visualization application in a day

by C. Brett

Shiny applications have gained increasing acceptance as a tool for exploring, publishing, manipulating and otherwise interacting with datasets. The `periscope` R package addresses delivery of performant, reliable, scalable, UI-consistent shiny application experiences across a large enterprise with developers in many different departments and roles. We pair this framework with the `canvasXpress` R package to showcase how to deliver a sophisticated and powerful big-data visualization solution. Workshop participants will produce an enterprise-grade shiny application visualizing a publicly available single-cell dataset (bioinformatics/systems-biology example).

Learning outcomes: At the end of this tutorial, participants will:

- understand the business and development problems facing the ever-expanding use of shiny applications in the enterprise;
- understand the limitations that can affect the usability of web-delivered interactive visualizations;
- identify the need to address these issues to accommodate the explosion of exploratory big-data visualization applications in shiny;
- gain hands-on experience using the periscope R package to jump-start the creation of a shiny application that will incorporate a large dataset visualization. The periscope framework will be demonstrated;
- become familiar with the capabilities of `canvasXpress` and will gain experience utilizing `canvasXpress` visualizations from both the user- and developer- perspectives to create and interact with sophisticated big-data visualizations.

Requirements: Participants will need to bring a computer with a current version of RStudio (1.1+) running on the operating system of their choice and R (3.5.3 or above (3.6 preferred)). A modern web browser and internet connection will be required to download the example data, install R packages, and access the `canvasXpress` documentation during the workshop. 8GB of RAM is recommended as a minimum requirement for computers running shiny applications locally.

The audience targeted are intermediate R users familiar with shiny applications and reactive programming. Experience creating data visualizations in R using any package is helpful, as is understanding the tidyverse ecosystem. All subject areas will find relevance in this workshop – the principles apply to any discipline where the ever-expanding size of data visualizations needs to be addressed in shiny applications. Intermediate R users should have experience creating data visualizations (in any package) plus a general understanding of shiny applications and reactive programming. The workshop attendees should have created at least a basic shiny application before attending this workshop.

Seamless R and C++ integration with `Rcpp`

by D. Eddelbuettel

R has become the *lingua franca* of statistical research and applications. At the same time, user demands on computing resources and performance have also increased driven by the ever-growing size of datasets, and may be coupled with increases in their complexity. `Rcpp` is used across different areas and fields. Its user base contains both advanced R developers with prior experience in compiled languages, as well as users for whom this is their first foray away from R. The tutorial aims to address both groups: gently introducing going to *compiled code* without fear thanks to the excellent toolchain supplied by R.

`Rcpp` allows users to extend R with compiled code. This offers two distinct approaches for performance gains. The first approach is replacing R code with similar C++ code. By offering similar types (vector, matrix, lists, ...) and functions (what we call *Rcpp sugar* offers numerous vectorised compiled functions with names and behaviour identical to what R users expect, a transition can be made with less effort than in other approaches. The second approach covers extensions via new libraries, either "wholecsale" or in small increments (as for example via snippets from Boost C++ via the `BH package`).

Learning outcomes:

- the *what*: show by example what `Rcpp` can do, and its key appeal in extending and accelerating work with R;
- the *how*: by using concrete (short) examples, users are lead to their first actual experiments with this approach;
- *next steps*: by covering a little bit of the fundamentals, enough of a foundation is provided to empower attendees to continue in self-study after the tutorial;
- *reachable goals*: by tying `Rcpp` extensions to particular topics of interest (often from the equally broad field of *machine learning*) students are motivated to aim for the next step and feel ready to undertake it themselves.

Requirements: All required packages, primarily `Rcpp` itself but also key extensions such as `RcppArmadillo` are easily installable from CRAN. This does of course depend somewhat on the OS, and each one of macOS, Windows or Linux do on occassion throw people off. But `Rcpp` is a first-class citizen on all platforms at CRAN and widely used on either OS. A good alternative is provided by RStudio Cloud which (while in 'beta') is widely accessible.

Create and share reproducible code with R Markdown and `workflowr`

by J. Blischak

This workshop introduces simple programming practices to help all R users, regardless of background, develop code that is more reproducible, transparent, and shareable—aspects that are increasingly important to a data scientist's productivity and impact. In hands-on activities, we will develop these practices using a new R package, `workflowr`: you will learn how to use `workflowr` to version control your code and data, and produce a fully functioning project website to share with colleagues. Additional topics covered include literate programming (R Markdown) and version control (Git, GitHub).

Learning outcomes: By the end of the tutorial, all participants will:

- appreciate the importance of using literate programming (R Markdown) and version control (Git) to develop and maintain reproducible research code;
- incorporate simple coding practices with `workflowr` to improve the reproducibility of their research code;
- use `workflowr` to produce and share a fully functioning project website from R Markdown source;
- identify "reproducibility bugs", and develop strategies to address them.

Requirements: Participants who will benefit most from our workshop have used R in their research or to analyze data. We have developed our tutorial for people who have relatively little background in computing. Participants will need a laptop that can connect to Wi-Fi Internet, with the following software installed:

- R (>= 3.2.5)
- RStudio (>= 1.1)
- Web browser

Participants will also need to have a GitHub account. Git is helpful, but is not required. The computation will be minimal, so any laptop should be fine. We have developed many online learning resources that the participants can use to build on the examples covered in-class.

Causal inference in R

by L. D'Agostino McGowan & M. Barrett

Making believable causal claims can be difficult, especially with the much repeated adage "correlation is not causation". This workshop will walk through some tools often used to practice safe causation, such as propensity scores and sensitivity analyses, allowing us to explicitly name our assumptions, and be able to use data to make causal claims. You'll be able to use the tools you already know–the tidyverse, regression models, and more–to answer the questions that are important to your work.

Learning outcomes: After this course, participants will be able to:

- understand the assumptions of causal inference;
- understand and create causal diagrams to describe the relationship(s) between variables;
- compute propensity scores and inverse probability weights using R;
- assess the pre-and post-propensity score model balance between exposure groups;
- understand how and when to apply causal inference techniques to their work.

Requirements: This tutorial will appeal to statisticians and data scientists interested in taking their inference to the next level, incorporating causal techniques. Audience members should be familiar with basic data manipulation and fitting regression models in R. Attendees will need access to R, RStudio, `tidyverse`, `ggdag`, and `tableone` R packages.

Reproducible computation at scale with `drake`: hands-on practice with a machine learning project

by W. Landau

The techniques in this tutorial enhance the maintainability, hygiene, speed, scale, and reproducibility of complicated projects with long runtimes. The `drake` R package resolves the dependency structure of data analysis pipelines, skips tasks that are already up to date, and cleanly organizes the output. Workflows with `drake` are efficient to maintain, and they provide tangible evidence that the results are synchronized with the underlying code and data. Thus, `drake` increases one's ability to trust the conclusions of research.

Learning outcomes:

- Function-oriented programming: participants will learn to express data analysis tasks as user-defined functions, experience the advantages of functions over imperative scripts, and understand the role of functions in drake-powered projects.
- Declarative workflows: students will declare drake targets to represent data analysis artifacts, identify the dependencies of those targets, and inspect the emergent dependency structures of entire workflows.
- Project maintenance: students will learn how drake responds to changing code and data, and they will understand the conditions that allow drake to skip steps and save time.
- Large plans: participants will practice declaring large workflows compactly using drake's domain-specific language.

Requirements: Data analysis projects can be large, ambitious, and complicated, with uncomfortably long runtimes and too many interconnected steps to neatly fit into a single R Markdown report. Participants of this tutorial are intermediate to advanced R users who expect to encounter large projects and who feel open to trying an alternative style of workflow management.

Each participant must bring a laptop with a web browser and the ability to connect to the internet. Students will spend most of their time working through cloud-hosted R notebooks and supporting Shiny applications.