Programación para el Análisis de Datos

Introducción a 

Daniel Fraiman

Maestría en Ciencia de Datos, Universidad de San Andrés

Plan

- Teórica-Práctica
- Vamos a practicar mediante ejercicios:
  - Estructuras de datos.
  - Estructuras de control: operaciones lógicas y loops.
  - Funciones, integración y optimización.
  - Estadística descriptiva
    - Exploración de datos: cargar, “limpiar”, analizar y guardar.
  - Simulación numérica.

Dinámica del curso

Se aprende haciendo ejercicios y enfrentándose a nuevos problemas.

Bibliografía

- La bibliografía de R es muy extensa. Algunos libros y páginas interesantes:
  - "R for data science". G.Grolemund, H. Wickham `https://r4ds.had.co.nz/`
  - "R Cookbook". JD Long.
  - `https://www.datanalytics.com/libro_r/`
  - `https://bookdown.org/jboscomendoza/r-principiantes4/`
  - https://stackoverflow.com/questions
  - *The R Journal* enteramente dedicada a artículos sobre el desarrollo y la aplicación de R. https://journal.r-project.org/.
  - cheatsheets (explicación resumida) de paquetes.

R es un lenguaje de programación interpretado de alto nivel con funciones orientadas a objetos. Permite, entre otras cosas, implementar técnicas estadísticas muy fácilmente en un entorno interactivo y gráfico.

R nace principalmente del leguaje *S* que era comercial.

¿Por qué aprender R?

- Flexibilidad: R tiene un montón de comandos y funciones específicas en estadística (y en otras áreas) que permite fácilmente implementar y evaluar técnicas nuevas.
- Popularidad: R tiene un gran cobertura y una gran disponibilidad de aplicaciones de vanguardia en infinidad de campos.
- Vanguardia: R tiene un montón de librerías que investigadores e usuarios disponibilizan. Los nuevos desarrollos teóricos rápidamente se convierten en librerías.
- Apoyo: enorme calidad del apoyo y soporte disponible. Existe una gran comunidad de R-users dedicados a mejorar y a contestar las preguntas. Dos congresos internacionales (useR -anual- y DSC -bianual-) y un journal *The R Journal* enteramente dedicada a artículos sobre el desarrollo y la aplicación de R.

¿Por qué aprender R?

- R es open source.
- nos permite modelar infinidad de situaciones reales.
- los nuevos desarrollos teóricos muy rápido alguien los convierte en paquetes.

¿Por qué aprender R?

SCIENTIFIC REPORTS

OPEN

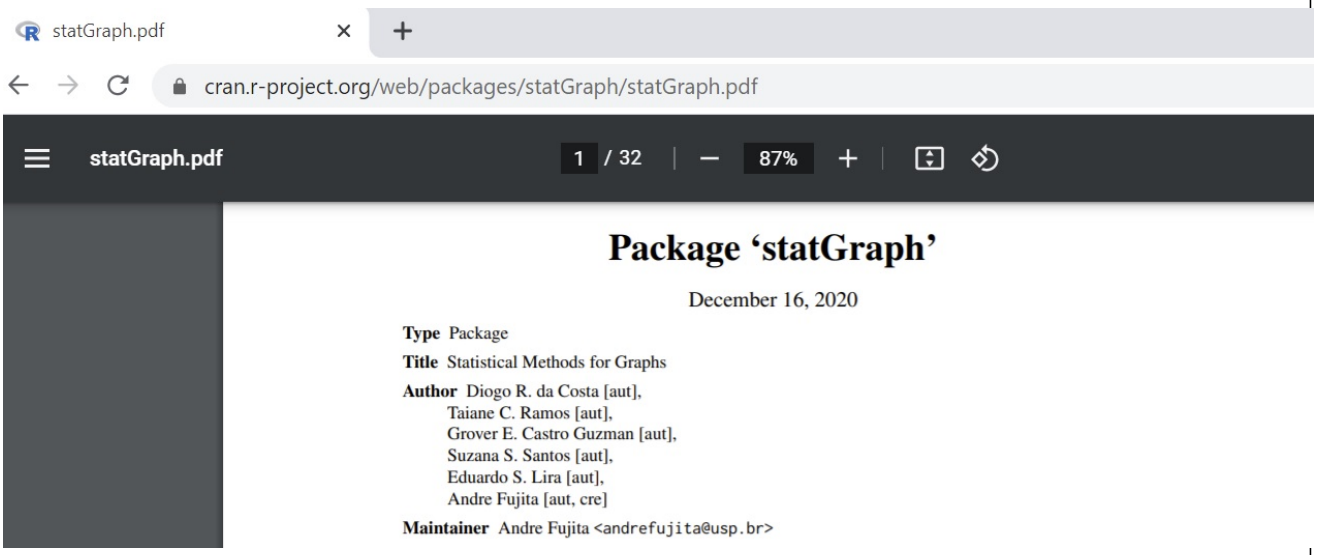
An ANOVA approach for statistical comparisons of brain networks

Daniel Fraiman[1,2] & Ricardo Fraiman[3,4]

Received: 13 November 2017
Accepted: 6 March 2018
Published online: 16 March 2018

The study of brain networks has developed extensively over the last couple of decades. By contrast, techniques for the statistical analysis of these networks are less developed. In this paper, we focus on the statistical comparison of brain networks in a nonparametric framework and discuss the associated detection and identification problems. We tested network differences between groups with an analysis of variance (ANOVA) test we developed specifically for networks. We also propose and analyse the behaviour of a new statistical procedure designed to identify different subnetworks. As an example, we show the application of this tool in resting-state fMRI data obtained from the Human Connectome Project. We identify, among other variables, that the amount of sleep the days before the scan is a relevant variable that must be controlled. Finally, we discuss the potential bias in neuroimaging findings that is generated by some behavioural and brain structure variables. Our method can also be applied to other kind of networks such as protein interaction networks, gene networks or social networks.

¿Por qué aprender R?



Package 'statGraph'

December 16, 2020

Type Package

Title Statistical Methods for Graphs

Author Diogo R. da Costa [aut],
Taiane C. Ramos [aut],
Grover E. Castro Guzman [aut],
Suzana S. Santos [aut],
Eduardo S. Lira [aut],
Andre Fujita [aut, cre]

Maintainer Andre Fujita <andrefujita@usp.br>

¿Por qué aprender R?

RoxygenNote 7.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2020-12-16 22:30:02 UTC

R topics documented:

¿Por qué aprender R?

2 statGraph-package

RoxygenNote 7.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2020-12-16 22:30:02 UTC

R topics documented:

Entornos R

- R básico. Corre un poco más rápido pero no es muy amigable.
- RStudio más amigable (completa comandos y funciones) y ayuda a ser más ordenado.
- Para correr en la nube:
  - RStudio http://rstudio.cloud
  - CO http://colab.to/r

Inicio y cierre de sesión

- Con la tecla # comentamos código.
- Con la tecla ↑ recuperamos las instrucciones utilizadas en la sesión.
- Para separar expresiones se emplea punto y coma (;).
- La combinación Ctrl-C interrumpe la edición o ejecución en curso.
- q() es el comando para salir de R. Permite grabar todo lo que estaba en memoria (no conviene!).

Algunas características de R

La mayoría de las operaciones y funciones en R están definidas con carácter vectorial. Pensar en forma vectorial agiliza mucho los procedimientos.

R almacena los resultados en objetos, para ser observados o analizados posteriormente, produciendo unas salidas mínimas.

Algunas características de R

Algunos comandos básicos

- Comando de asignación: <- (también se puede usar el =, pero no es prolijo).
- Los paréntesis () se emplean para los argumentos de las funciones y para agrupar expresiones algebráicas.
- Los corchetes [ ] o dobles corchetes [[ ]] para seleccionar partes de un objeto así como también el $.
- Las llaves { } para agrupar expresiones.
- Help: help(mean) o bien ? mean ; o algo más genérico help.search("data input"). example(mean). Se sale del help con q.
- Los nombres de los objetos tienen que comenzar con alguna letra mayúsculas o minúsculas. Luego puede agregarse números y puntos a la derecha (NO blancos).

Algunas características de R

Clases de objetos

- Los vectores son el tipo básico de objeto en R.
- Las matrices o los arrays son generalizaciones multidimensionales de los vectores.
- Los factores sirven para representar datos categóricos.
- Las listas son una forma generalizada de vector en las cuales los elementos no tienen por qué ser del mismo tipo y a menudo son a su vez vectores o listas.
- Las tablas de datos (data frames) son estructuras similares a una matriz, en que cada columna puede ser de un tipo distinto a las otras.
- Las funciones son también objetos de R que pueden almacenarse en el espacio de trabajo.

Algunas comandos y comentarios útiles

Algunos comandos

- Cuando aparecen dos veces dos puntos (::) esto nos indica que el comando de la derecha lo corremos con el paquete que se encuentra a la izquierda del :: (Ej: stats::rnorm(16))
- set.seed(42)
- getwd()
- *setwd*(" ∼ /*Documents*/*Daniel*/*maestria*")
- history(100) # muestra las últimas 100 líneas
- library(pracma); tic("el programa tarda") ; total <- sum(rnorm(1e7)); toc()
- outer(x,y,funcion)
  outer(month.abb, 1999:2003, FUN = "paste")

Algunos paquetes interesantes

Paquete lubridate: para acomodar y leer fechas

Paquete stringr: para analizar cadenas de caracteres

Paquete dplyr: manejo de bases

Paquete ggplot2: para hacer gráficos lindos

Paquete tidyverse: "metapaquete" con los anteriores y algunos más

Algunos paquetes interesantes

Paquete haven: para importar SPSS', 'Stata' and 'SAS'

Paquete readxl: para importar Excel

Paquete janitor: para ordenar Excel

Orden de temas

- Estructuras de datos: vectores, matrices, arrays, listas.
- Permutaciones, funciones, estructuras de control, plots básicos
- Integrales y búsqueda de máximos (mínimos). integrate, optimize.
- Elementos de estadística descriptiva, y dependencia entre tipos de variables.
- Gráficos
- Cargar, guardar, y "limpieza" de datos.
- Simulación numérica.

Muy importante

- Reproducibilidad.
- Prolijidad. ¿Dentro de un año podrás reproducir lo que hiciste?