

Lecture 1. Introduction and R Basics

R and Data Visualization

BIG2006, Hanyang University, Fall 2022

Course syllabus

- ▶ **Instructor:** Jaehong Jeong (jaehongjeong@hanyang.ac.kr)
- ▶ **Course schedule:** 09:00 - 12:00 Thursday
- ▶ **Prerequisites**

Only students who are double majoring **Big Data Science** Introduction to Probability and Statistics (BIG2001) or an equivalent course

Statistical Computing (MAT2021) is recommended for a better understanding of this course

A willingness to learn some advanced features in R independently is needed.

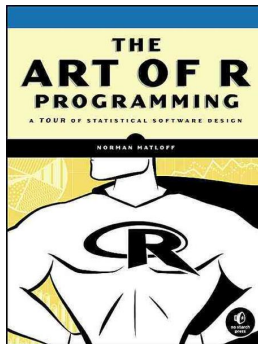
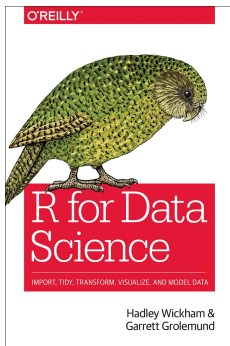
Course description

This is a course intended for students **majoring Big Data Science** who want to explore and visualize data sets using R. The goal is to learn the **basics of data science with R** and some of the most essential **R tools for data visualization**.

The main topics covered will include programming structures, communicating data, graphical features, data visualization, and explanatory data analysis. Additional topics, such as statistical inference and machine learning examples, will be discussed if time permits.

Textbook

- ▶ No textbook is required.
- ▶ R manuals by the R Development Core team (<https://cran.r-project.org/manuals.html>) are available.



Recommended textbook

- ▶ **R for Data Science** (<https://r4ds.had.co.nz/>). Wickham and Grolemund. O'REILLY
- ▶ **The Art of R Programming**. Matloff. No Starch Press
- ▶ The Book of R. Davies. No Starch Press
- ▶ Using R for Introductory Statistics (2nd edition). Verzani. CRC Press
- ▶ Displaying Time Series, Spatial, and Space-Time Data with R (2nd edition). Lamigueiro. CRC Press
- ▶ Data Visualisation with R. Rahlf. Springer

How this class will work

- ▶ No programming knowledge presumed.
- ▶ Some statistics knowledge presumed:
 - Probability and probability distribution
 - Hypothesis testing (t-test, confidence intervals)
 - Linear regression
- ▶ Class will be very cumulative.

Grading

- ▶ **Attendance (5%)**
- ▶ **Assignment (20%):** (Bi)weekly homework will be assigned.
Late submissions are not accepted after the deadline.
- ▶ **Midterm Exam (30%):** R programming
- ▶ **Final Exam (30%):** R and Data visualization
- ▶ **Team Project (15%)**

A project on real data visualization and analysis is arranged for each student or group throughout the semester. Students are encouraged to look for and analyze one data set of their interests. A final report is submitted for evaluation.

Grading

- ▶ **An F score is given if a student misses the class over 5 times or misses one of the exams.**
- ▶ **Students who achieved a total performance of less than 50% will get an F.**

Tentative weekly plan

- ▶ Week 1 - 4: Introduction to R
- ▶ Week 5: Communicating and transforming data
- ▶ Week 6: Math and stat functions, simulation
- ▶ Week 7: **Midterm Exam**
- ▶ Week 8 - 11: Data visualization
- ▶ Week 12 - 13: Exploratory data analysis, linear regression
- ▶ Week 14: Additional topics
- ▶ Week 15: **Final Exam**
- ▶ Week 16: **Team project**

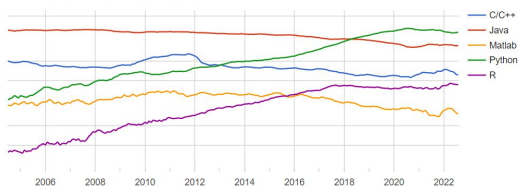
What is R?

- ▶ R was created by Rob Gentleman and Ross Ihaka in 1993; it is based on the S language developed at Bell Labs by John Chambers (Stanford Statistics).
- ▶ It is an **open-source language and environment** for statistical computing and graphics.

Who uses R?

Traditionally, academics and researchers. However, recently R has extended also to industry and enterprise market. Worldwide usage on log-scale:

PYPL Popularity of Programming Language



Worldwide, Aug 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.11 %	-2.6 %
2		Java	17.35 %	-0.9 %
3		JavaScript	9.48 %	+0.2 %
4		C#	7.08 %	+0.1 %
5		C/C++	6.19 %	-0.3 %
6		PHP	5.47 %	-0.8 %
7		R	4.35 %	+0.6 %
8	↑↑↑	TypeScript	2.79 %	+1.1 %
9	↑↑↑	Swift	2.09 %	+0.5 %
10	↓↓↓	Objective-C	2.03 %	+0.2 %

Source: <https://pypl.github.io/PYPL.html>

The PYPL Popularity of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

Why should you learn R?

Pros:

- ▶ Open source and cross-platform
- ▶ Created with statistics and data in mind; new ideas and methods in statistics usually appear in R first.
- ▶ Provides a wide range of high-quality packages for data analysis and visualization.
- ▶ Arguably, the most commonly used language by data scientists.

Cons:

- ▶ Performance/Scalability: low speed, poor memory management
- ▶ Some packages are low-quality and provide no support.
- ▶ A unconventional syntax and a few unusual features compared to other languages.

A few alternatives to R

- ▶ **Python**: fastest growing, general-purpose programming, with data science libraries
- ▶ **SAS**: used for statistical analysis; commercial and expensive, slower development
- ▶ **SQL**: designed for managing data held in a relational database management system
- ▶ **MATLAB**: proprietary, mostly for numerical computing, and matrix computations.

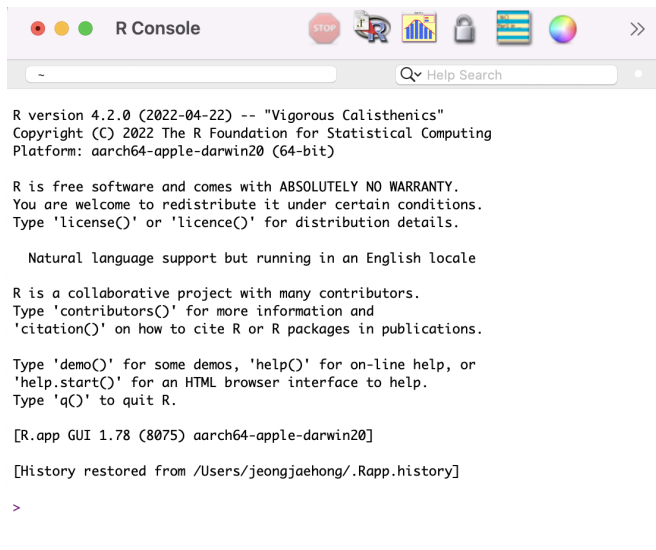
What makes R good?

- ▶ R is an **interpreted language**, i.e., programs do not need to be compiled into machine-language instructions.
- ▶ R is **object oriented**, i.e., it can be extended to include non-standard data structures (**object**). A generic function can act differently depending on what objects you pass to it.
- ▶ R supports **matrix arithmetics**.
- ▶ R packages can generate **publication-quality** plots, and **interactive graphics**.
- ▶ Many **user-created R package** contain implemetations of **cutting edge statistics methods**. Currently (August 2022), the CRAN package repository features 18,527 available packages (<https://cran.r-project.org/web/packages/>).

Installing R

- ▶ R is open sources and cross platform (Linux, Mac, Windows).
- ▶ To download it, go to the Comprehensive R Archive Network CRAN (<https://cran.r-project.org/>) website. Download the latest version for your OS and follow the instructions.
- ▶ Each year a new version of R is available, and 2-3 minor releases. You should update your software regularly.

R console



```
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.78 (8075) aarch64-apple-darwin20]

[History restored from /Users/jeongjaehong/.Rapp.history]

>
```

Running R code

Interpreter mode:

- ▶ open a terminal and launch R by calling "R" (or open an R console).
- ▶ type R commands interactively in the command line, pressing enter to execute.
- ▶ use `q()` to quit R

Scripting mode:

- ▶ write a text file containing all commands you want to run.
- ▶ save your script as an R script file, e.g., "myscript.R"
- ▶ execute your code from the terminal by calling "Rscript myscript.R"

R editors

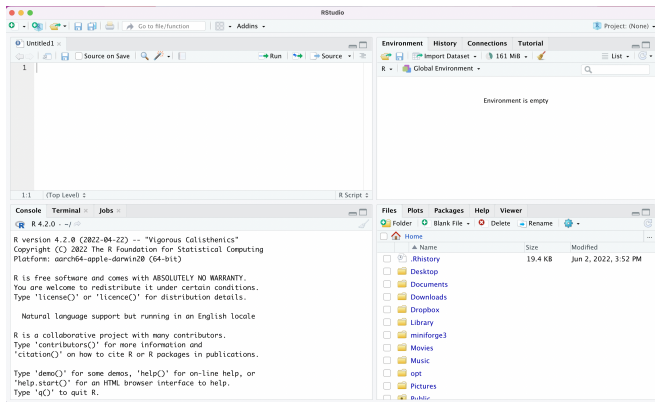
The most popular R editors are

- ▶ **RStudio**, an integrated development environment (IDE) for R.
- ▶ **Emacs**, a free, powerful, customizable editor for many languages.

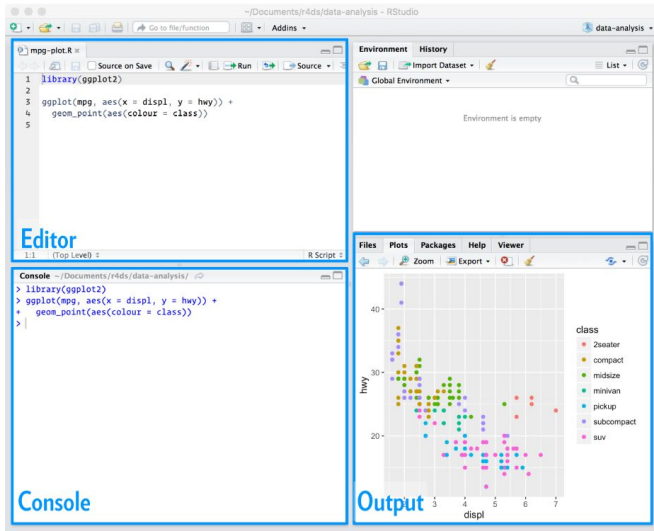
In this class, we will use RStudio, as it is more user-friendly.

Installing RStudio

RStudio is open-source and cross-platform (Linux, Mac, Windows). Download and install the latest version for your OS from the official website.



RStudio window



R document types

- ▶ **R script** a text file containing R commands stored together.
- ▶ **R Markdown** files can generate high quality reports containing notes, code, and code outputs. **Python and bash code** can also be executed.
- ▶ **R Notebook** is an R Markdown document with **chunks that can be executed independently and interactively**, with output visible immediately beneath the input.
- ▶ **R presentation** lets you author **slides** that make use of R code and LaTeX equations as **straightforward** as possible.
- ▶ **R Sweave** enables the embedding of **R code within LaTeX documents**.
- ▶ **Other documents**

R packages

- ▶ R packages are a **collection of R functions, compiled code, and sample data**.
- ▶ They are stored under a directory called **library** in the R environment.
- ▶ Some packages are **installed by default** during R installation and are always automatically loaded at the beginning of an R session.
- ▶ Additional packages by the user from
 - **CRAN**: the first and biggest R repository.
 - **Bioconductor**: Bioinformatics packages for the analysis of biological data.
 - **github**: packages under development

Installing R packages from different repositories

From CRAN

```
# install.packages("Package Name"), e.g.  
install.packages("glmnet")
```

From Bioconductor

```
# First, load Bioconductor script. You need to have an R version >=3.3.0.  
source("https://bioconductor.org/biocLite.R")  
# Then you can install packages with: biocLite("Package Name"), e.g.  
biocLite("limma")
```

From github

```
# You need to first install a package "devtools" from CRAN  
install.packages("devtools")  
# Load the "devtools" package  
library(devtools)  
# Then you can install a package from some user's repository, e.g.  
install_github("twitter/AnomalyDetection")  
# or using install_git("url"), e.g.  
install_git("https://github.com/twitter/AnomalyDetection")
```


Where are R packages stored?

```
# Get library locations containing R packages  
.libPaths()
```

```
## [1] "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
## [2] "C:/Program Files/R/R-4.2.1/library"
```

```
# Get the info on all the packages installed  
installed.packages()[1:5, 1:3]
```

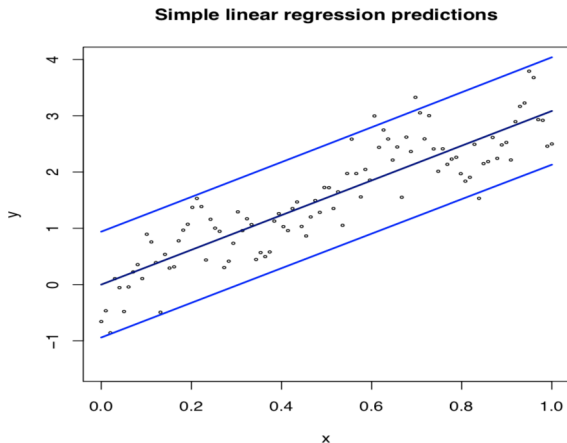
```
##           Package      LibPath  
## ape          "ape"        "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
## ash          "ash"        "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
## askpass      "askpass"     "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
## assertthat  "assertthat" "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
## backports   "backports"   "C:/Users/durlc/AppData/Local/R/win-library/4.2"  
##           Version  
## ape          "5.6-2"  
## ash          "1.0-15"  
## askpass      "1.1"  
## assertthat  "0.2.1"  
## backports   "1.4.1"
```

```
# Get all packages currently loaded in the R environment  
search()
```

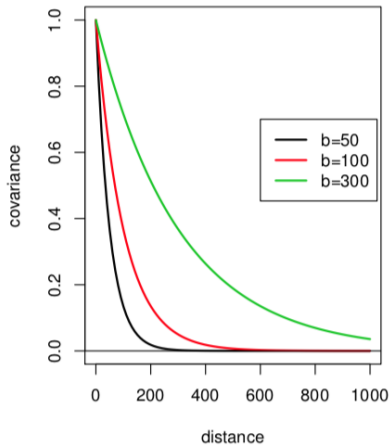
```
## [1] ".GlobalEnv"      "package:stats"    "package:graphics"  
## [4] "package:grDevices" "package:utils"    "package:datasets"  
## [7] "package:methods" "Autoloads"        "package:base"
```

R graphics

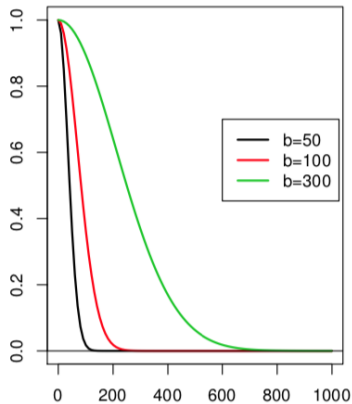
- ▶ Various plots available for EDA
- ▶ All kind of optional bells and whistles for plotting

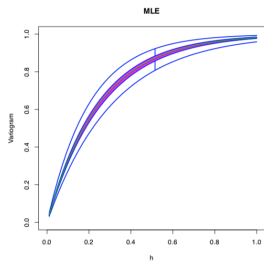
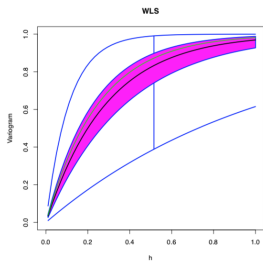
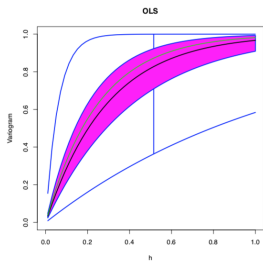


Exponential covariance function

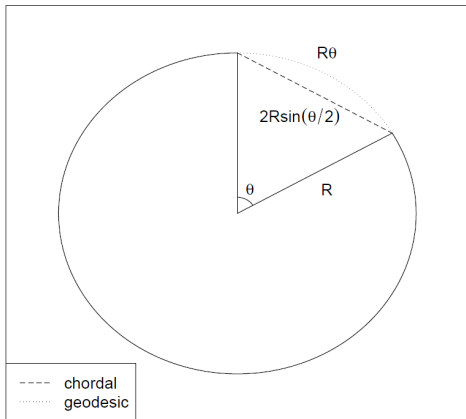


Gaussian covariance function

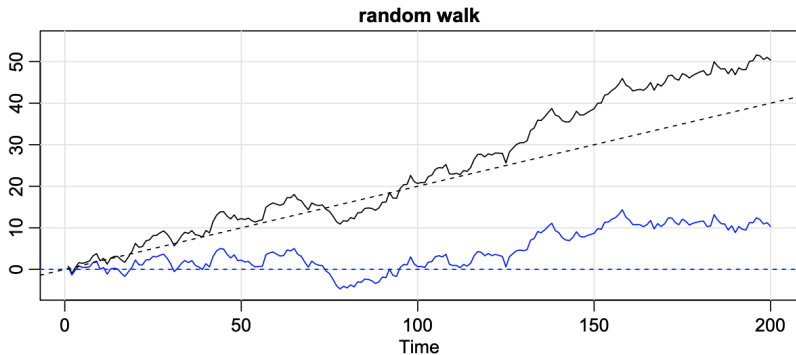




Chordal vs. Geodesic distance

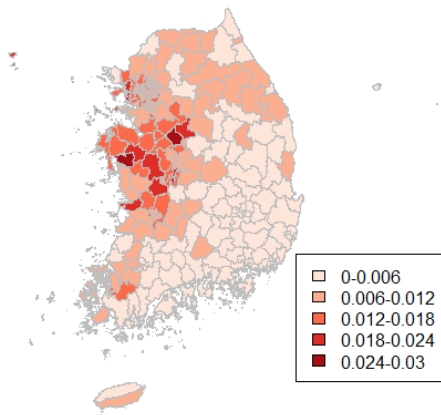


Credit: Jeong, Jun, and Genton (2017). Spherical process model for global spatial statistics. Statistical Science

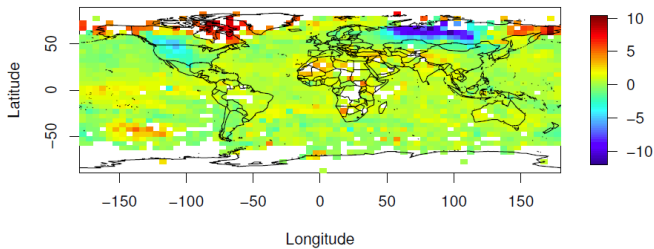


Credit: Shumway and Stoffer (2017). Time series analysis and its application. Springer

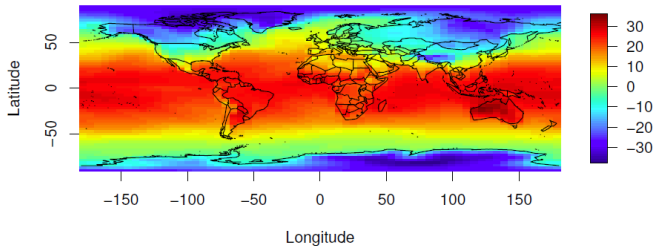
of cases per 1,000 in 2017

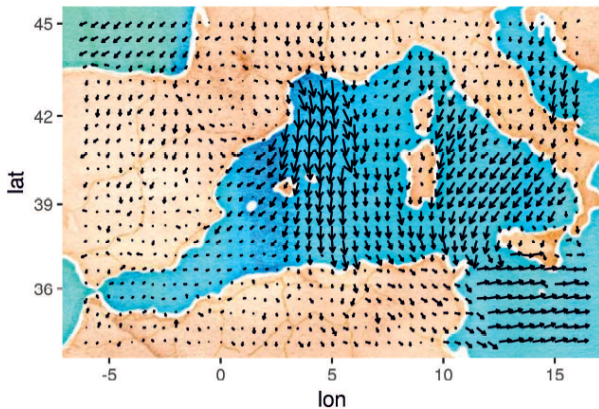


Surface temperature anomaly (Observational estimate)



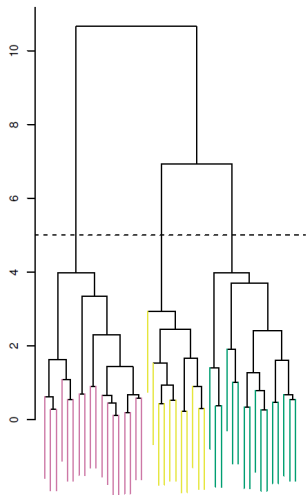
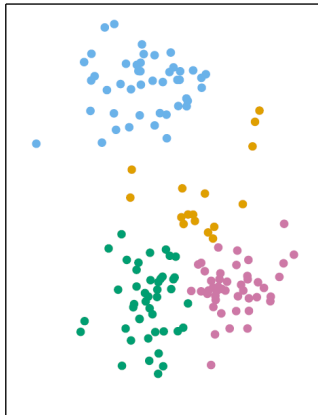
Surface temperature (Climate model output)





Credit: Wikle, Zammit-Mangion, and Cressie (2019).
Spatio-temporal statistics with R. Chapman and Hall/CRC

K=4



Reference

- ▶ Matloff, N. [The Art of R Programming: A Tour of Statistical Software Design](#). No Starch Press.
- ▶ Nguyen, L. H. [Introduction to R](#). CME/STATS 195, Stanford. Lecture 1.