

M11-intro

LearningSpoonsR

2018-12-22

시작

```
"Hello World"
```

```
## [1] "Hello World"
```

0. WIFI + R/Rstudio 설치
1. 강사 소개
2. Final Project로 보는 강의 목적 (M41)
3. 수강생 Project 소개

R로 시작하는 데이터 분석 및 시각화

- 부제: 마케팅/영업/기획/매출관리 직무가 무엇이든 데이터 분석은 필수
- 강의페이지 <https://github.com/LearningSpoonsR/LS-DS>
- FB그룹(가입신청) <https://www.facebook.com/groups/440130033101009/>
- 일정
 - 2018.12.23 ~ 2019.01.27 (5회)
 - **2018.12.30 휴강**
 - 매주 일요일 10:30 ~ 13:30
 - 2018.12.23, 2019.1.6, 2019.1.13, 2019.1.20, 2019.1.27
- 코스매니저: 김형중 (출결등 기타)
- 강사이메일: learningSpoonsR@gmail.com

데이터 사이언스 vs 프로그래밍

- 데이터 사이언스
 - 데이터를 과학적인 방법으로 접근하여
 - 이해, 분석, 의사결정의 도구로 삼는 것
- 프로그래밍
 - 컴퓨터가 특정한 목적을 가진 작업을 수행하게 하는 것
- 관계
 1. 프로그래밍은 데이터 사이언스의 도구로 사용됨
 2. 데이터 규모가 커질 수록, 이해와 분석의 깊이가 깊어질 수록 프로그래밍 의존도 상승

DS 프로젝트 과정과 요구 Skill Set

- DS?
 - Mathematics is to find patterns. - Dr. Feynman
 - DS is to find patterns **from data**.
- DS 프로젝트 과정
 1. 결론이 될 “검증하고 싶은 가설”을 정의
 2. “가설의 검증”을 프로그램의 “목표”로 설정
 3. “목표”를 이루기 위한 과정을 작업 단위로 분리하여 설계
 4. 프로그램 작성
 5. 가설 검증
 6. 결론 도출
 7. 결론의 공유
- Skill Set (현재 자신의 Skill을 별5개 만점으로 표시해보세요)
 1. 컴퓨터 기술 (SW/HW/Programming)
 2. 수학/통계학
 3. 커뮤니케이션
 4. 논리적/과학적 접근
 5. 영어

수업 목표 & 진행 방식

- 목표

1. DS를 위한 프로그래밍을 배우고 프로젝트의 과정을 체득합니다.
2. Final Project의 결과물을 낼 수 있을 정도로 R 프로그램을 다룰 수 있습니다.
3. 수업 Final Project에 준하는 결과물을 강의 마지막에 만들어 냅니다.
4. R 프로그래밍을 스스로 프로그래밍할 수 있게 됩니다.
5. 앞의 별표 갯수를 늘립니다!
6. Skill set을 일터로 가져갑니다.

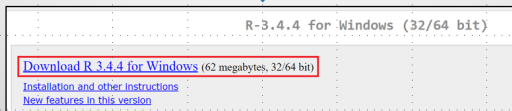
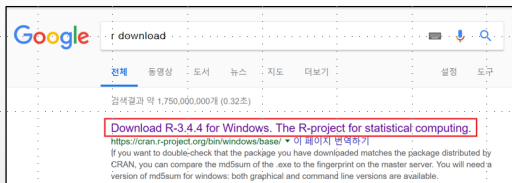
- 진행 방식

1. 데이터 분석 과정을 포함한
2. Mini-Project를 반복하며
3. 프로젝트 과정을 체득합니다.

R

- GOOGLE "R DOWNLOAD"

- LINK → BASE → DOWNLOAD → INSTALL → 아이콘 2개



! google 검색 생활화 합시다.
! Chrome 설치를 권장합니다.

RStudio (2)

Editor (메모장)
 Ctrl + Enter
 → 명령실행
 Ctrl + I
 → 전체화면

Console (실행창)
 Enter
 → 명령실행
 Ctrl + 2
 → 전체화면

현재 변수 및 환경

최근 명령 (Command)

HELP 창

R과 CPU의 통신 기록

탐색기 기능

그래프

The screenshot shows the RStudio interface with the following components highlighted:

- Editor (메모장):** The main area for writing R code. It shows a script with ggplot2 and flexdashboard code. Annotations indicate that Ctrl + Enter runs the current line and Ctrl + I switches to full-screen mode.
- Console (실행창):** The area for running R commands. It shows the output of the R setup function. Annotations indicate that Enter runs the current command and Ctrl + 2 switches to full-screen mode.
- Environment:** The panel showing the current environment. It is currently empty.
- History:** The panel showing the command history.
- Files:** The panel showing the file explorer. It lists files like Rhistory, data, ggplot.html, ggplot.R, ggplot.Rmd, rscconnect, script.docx, and 새 엑셀의 문서.txt.
- Plots:** The panel showing the plots.
- Packages:** The panel showing the installed packages.
- Help:** The panel showing the help documentation.
- Viewer:** The panel showing the viewer.

RStudio (3)

1. Tool -> Global Option

- **Appearance:** 폰트와 색상 등을 조정 (어두운 바탕에 밝은 글씨 vs 밝은 바탕에 어두운 글씨)
- **PANE Layout:** 모니터 크기와 작업 목적에 따라 조정 (좌우 vs 상하로 Editor와 Console창 조정)
- **CODE -> Saving -> Text Encoding in -> UTF-8**

2. 윈도우 사용자 계정과 R파일 폴더 이름은 영어로

- 제어판 -> 사용자 계정 -> 계정 이름 변경 -> 영어로 입력
- R파일, 불러올 데이터 파일 등은 가급적이면 영어로
- R Studio는 관리자 권한으로 실행해야 함 (default로 관리자 권한)

3. 한영 전환

- 한글로 전환: `Sys.Setlocale("LC_ALL", "ko_KR.UTF-8")`
- 영어로 전환: `Sys.Setlocale("LC_ALL", "en_KR.UTF-8")`

RStudio (4)

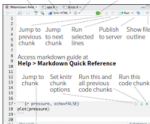
RStudio IDE : : CHEAT SHEET

Documents and Apps



Open Shiny, R Markdown, knitr, Sweave, LaTeX, R Markdown and more in Source Pane

Check spelling
Render output
Choose output format
Insert code location
Choose chunk



RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app



Debug Mode

Open with **debug()**, **browser()**, or a breakpoint. RStudio will open the debugger mode when it encounters a breakpoint while executing code.



Click next to line number to add/remove a breakpoint.



Highlighted line shows where execution has paused

Run commands in environment where execution has paused

Examine variables in executing environment

Select function in traceback to debug

Step through code one line at a time

Step into and out of functions to run

Resume execution mode

Quit debug

Write Code

Navigate Open New Save Find and replace Compile as notebook Run selected code

Curators of previous code

Source with or without Echo

Save file

Multiple cursors/column selection with **Alt** + mouse drag.

Code diagnostics that appear in the margin.

Hover over diagnostic symbols for details.

Syntax highlighting based on your file's extension

Tab completion to finish function names, file paths, arguments, and more.

Multi-language code snippets to quickly use common blocks of code.

Change file type

Working Directory

Maximize, minimize panes

Press **Ctrl** + **Q** to see command history

Drag pane boundaries

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

Change file type

R Support

Import data with wizard

History of past commands to run/redo

Display RShiny slideshows

File > New File > R Presentation

Save workspace

Delete all saved objects

Search inside environment

Choose environment to display from

list of parent environments

Display objects as list or grid

Displays saved objects by type with short description

View in data viewer

View function source code

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Change directory

Path to displayed directory

Upload file

Delete file

Rename file

Pro Features

Share Project with Collaborators

Active shared collaborators

Start New R Session in current project

Close R Session in project

Select R Version

PROJECT SYSTEM

File > New Project

RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

Name of current project

RStudio opens plots in a dedicated Plots pane

GUI Package manager lists every installed package

Install Update Create reproducible package

Package Library for your project

Click to load package with package with detach

Package version

Delete from installed library

RStudio opens documentation in a dedicated Help pane

Home page of helpful links

Search within help file

Search for help file

Viewer Pane displays HTML content, such as Shiny apps, R Markdown reports, and interactive visualizations

Stop Shiny Publish to shinyapps.io, Refresh app, RStudio connects

View 'data' opens spreadsheet-like view of data set

Filter rows by value or value range

Sort by values

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value

Search for value



RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at www.rstudio.com • RStudio IDE 0.99.832 • Updated: 2016-01

RStudio (5)

1 LAYOUT

Move focus to Source Editor
Move focus to Console
Move focus to Help
Show History
Show Files
Show Plots
Show Packages
Show Environment
Show Git/ SVN
Show Build

Windows/Linux **Mac**
Ctrl+1 Ctrl+1
Ctrl+2 Ctrl+2
Ctrl+3 Ctrl+3
Ctrl+4 Ctrl+4
Ctrl+5 Ctrl+5
Ctrl+6 Ctrl+6
Ctrl+7 Ctrl+7
Ctrl+8 Ctrl+8
Ctrl+9 Ctrl+9
Ctrl+0 Ctrl+0

2 RUN CODE

Search command history
Navigate command history
Move cursor to start of line
Move cursor to end of line
Change working directory

Windows/Linux **Mac**
Ctrl+H Ctrl+H
Home Home
Ctrl+Shift+H Ctrl+Shift+H
Esc Esc

Interrupt current command
Clear console

Windows/Linux **Mac**
Ctrl+Q Ctrl+Q
Ctrl+Shift+F10 Ctrl+Shift+F10

Restart R session

Run current line (selection)

Windows/Linux **Mac**
Ctrl+Enter Ctrl+Enter
Alt+Enter Alt+Enter

Run from current to end

Windows/Linux **Mac**
Ctrl+Shift+F10 Ctrl+Shift+F10

Run the current function

Windows/Linux **Mac**
Ctrl+Alt+F Ctrl+Alt+F

Source a file

Windows/Linux **Mac**
Ctrl+Alt+G Ctrl+Alt+G

Source the current file

Windows/Linux **Mac**
Ctrl+Shift+S Ctrl+Shift+S

Source with echo

Windows/Linux **Mac**
Ctrl+Shift+Enter Ctrl+Shift+Enter

3 NAVIGATE CODE

Goto File/Function

Windows/Linux **Mac**
Ctrl+L Ctrl+L
Alt+L Alt+L

Fold Selected

Windows/Linux **Mac**
Ctrl+M Ctrl+M

Unfold Selected

Windows/Linux **Mac**
Ctrl+M Ctrl+M

Fold All

Windows/Linux **Mac**
Ctrl+M Ctrl+M

Unfold All

Windows/Linux **Mac**
Ctrl+M Ctrl+M

Go to line

Windows/Linux **Mac**
Ctrl+G Ctrl+G

Jump to

Windows/Linux **Mac**
Ctrl+G Ctrl+G

Switch to tab

Windows/Linux **Mac**
Ctrl+Tab Ctrl+Tab

Previous tab

Windows/Linux **Mac**
Ctrl+F11 Ctrl+F11

Next tab

Windows/Linux **Mac**
Ctrl+F12 Ctrl+F12

First tab

Windows/Linux **Mac**
Ctrl+F11 Ctrl+F11

Last tab

Windows/Linux **Mac**
Ctrl+F12 Ctrl+F12

Navigate back

Windows/Linux **Mac**
Ctrl+F9 Ctrl+F9

Navigate forward

Windows/Linux **Mac**
Ctrl+F10 Ctrl+F10

Jump to Brace

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Select within Braces

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Use Selection for Find

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Find in Files

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Find Next

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Find Previous

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Jump to Word

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Jump to Start/End

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Toggle Outline

Windows/Linux **Mac**
Ctrl+F Ctrl+F

4 WRITE CODE

Attempt completion

Windows/Linux **Mac**
Ctrl+Space Ctrl+Space

Navigate candidates

Windows/Linux **Mac**
Ctrl+Space Ctrl+Space

Accept candidate

Windows/Linux **Mac**
Ctrl+Space Ctrl+Space

Dismiss candidates

Windows/Linux **Mac**
Ctrl+Space Ctrl+Space

Undo

Windows/Linux **Mac**
Ctrl+Z Ctrl+Z

Redo

Windows/Linux **Mac**
Ctrl+Y Ctrl+Y

Cut

Windows/Linux **Mac**
Ctrl+X Ctrl+X

Copy

Windows/Linux **Mac**
Ctrl+C Ctrl+C

Paste

Windows/Linux **Mac**
Ctrl+V Ctrl+V

Select All

Windows/Linux **Mac**
Ctrl+A Ctrl+A

Delete Line

Windows/Linux **Mac**
Ctrl+L Ctrl+L

Select Word

Windows/Linux **Mac**
Ctrl+D Ctrl+D

Select to Line Start

Windows/Linux **Mac**
Ctrl+Home Ctrl+Home

Select to Line End

Windows/Linux **Mac**
Ctrl+End Ctrl+End

Select Page Up/Down

Windows/Linux **Mac**
Ctrl+PageUp Ctrl+PageUp

Select to Start/End

Windows/Linux **Mac**
Ctrl+Home Ctrl+Home

Delete Word Left

Windows/Linux **Mac**
Ctrl+Backspace Ctrl+Backspace

Delete Word Right

Windows/Linux **Mac**
Ctrl+Delete Ctrl+Delete

Delete to Line End

Windows/Linux **Mac**
Ctrl+End Ctrl+End

Delete to Line Start

Windows/Linux **Mac**
Ctrl+Home Ctrl+Home

Indent

Windows/Linux **Mac**
Ctrl+Tab Ctrl+Tab

Outdent

Windows/Linux **Mac**
Ctrl+Tab Ctrl+Tab

Yank line up to cursor

Windows/Linux **Mac**
Ctrl+Y Ctrl+Y

Yank line after cursor

Windows/Linux **Mac**
Ctrl+Y Ctrl+Y

Insert yanked text

Windows/Linux **Mac**
Ctrl+Y Ctrl+Y

Alt+Enter

Windows/Linux **Mac**
Ctrl+Enter Ctrl+Enter

Insert %

Windows/Linux **Mac**
Ctrl+Enter Ctrl+Enter

Show help for function

Windows/Linux **Mac**
Ctrl+F1 Ctrl+F1

Show source code

Windows/Linux **Mac**
Ctrl+F2 Ctrl+F2

New document

Windows/Linux **Mac**
Ctrl+N Ctrl+N

New document (Chrome)

Windows/Linux **Mac**
Ctrl+N Ctrl+N

Open document

Windows/Linux **Mac**
Ctrl+O Ctrl+O

Save document

Windows/Linux **Mac**
Ctrl+S Ctrl+S

Close document

Windows/Linux **Mac**
Ctrl+W Ctrl+W

Close document (Chrome)

Windows/Linux **Mac**
Ctrl+W Ctrl+W

Extract function

Windows/Linux **Mac**
Ctrl+Alt+X Ctrl+Alt+X

Extract variable

Windows/Linux **Mac**
Ctrl+Alt+V Ctrl+Alt+V

Resident lines

Windows/Linux **Mac**
Ctrl+Shift+C Ctrl+Shift+C

Reformat Comment

Windows/Linux **Mac**
Ctrl+Shift+F Ctrl+Shift+F

Reformat Selection

Windows/Linux **Mac**
Ctrl+Shift+F Ctrl+Shift+F

Select within braces

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Show Diagnostics

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Transpose Letters

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Move Lines Up/Down

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Copy Lines Up/Down

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Add New Cursor Above

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Add New Cursor Below

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Move Active Cursor Up

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Move Active Cursor Down

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Find and Replace

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Use Selection for Find

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Replace and Find

Windows/Linux **Mac**
Ctrl+F Ctrl+F

Windows/Linux

Tab or Ctrl+Space

Windows/Linux **Mac**
Ctrl+Space Ctrl+Space

Enter, Tab, or ↵

Windows/Linux **Mac**
Ctrl+Z Ctrl+Z

Ctrl+Shift+Z

Ctrl+X

Ctrl+C

Ctrl+V

Ctrl+A

Ctrl+D

Shift+Arrow

Ctrl+Shift+↵

Alt+Shift+↵

Alt+Shift+↵

Shift+PageUp/Down

Shift+Alt+↵

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Ctrl+Backspace

Ctrl+Delete

Mac

Tab or Cmd+Space

Mac
Cmd+Space

Enter, Tab, or ↵

Mac
Cmd+Z

Cmd+Shift+Z

Cmd+X

Cheatsheet & English

- Cheatsheets?
 - 모든 문법과 명령어를 외우는 것은 불가능합니다.
 - 언제든지 참조하고 검색하는 것이 프로그래밍의 과정입니다.
 - Cheat Sheet은 “컨닝페이퍼”로서 사용법이 정리되어 있는 문서입니다.
 - 여러분은 수강후에 Cheat Sheet과 과거의 경험을 참조하고 검색하면서 프로그래밍을 하게 될 것입니다.
- English?
 - Cheat Sheet을 가득 채운 영어가 겁나시나요?
 - 웹 검색 결과와 프로그램 설명서 등은 대부분 영어로 되어있습니다.
 - 그러나 대부분 컴퓨터와 프로그래밍에 관련된 용어일 뿐입니다.
 - 자주 사용되는 용어는 약 1000개 정도일 뿐이므로 경험과 함께 금방 익숙해 집니다.
 - 프로그래밍에 관련된 설명은 어려운 문법을 사용하는 경우가 드물어 구글/네이버 번역기으로도 아주 잘 이해할 수 있습니다.

Excel vs R

Excel

- GUI (화면에 자동으로 보임)
- 분석 과정이 기록되지 않으며 결과 위주
- 매우 쉬움
- 데이터가 커지면 느려서 버벅됨
- 눈에 보이기에 직관적
- 실수에 관대함

R

- Console Interface (명령어로 접근)
- Reproducibility (재현가능성)
- 분석 과정이 기록되어 결과를 재생산할 수 있음
- 초반의 학습곡선이 있음
- 훨씬 빠르게 대용량 데이터 처리
- 추상적, 논리적인 사고가 요구되고 길러짐
- 결과물에 실수 없음이 어느정도 보장

R vs 다른 프로그래밍 언어

분석 (Scientific Computing)		개발 (Development)	
R	가장 배우기 쉽고 빠르게 성장	C, C++	가장 빠르며 다른 언어의 기반
MATLAB	수리 계산 (유료)	JAVA	웹 개발용으로 주로 쓰임
SAS	슈퍼 대용량 Data (매우 유료)		
Python: R보다 좀 더 어려움, 분석 및 개발 양쪽을 충분히 지원 R보다 조금 더 빠르며 R을 배우고 나면 더 쉽게 배울 수 있음			

• R의 장점

- 무료
- 가장 왕성한 open-source 확장 프로그램들
- Rstudio 회사가 존재
 - 확장 프로그램들을 통합하여 관리
 - 매뉴얼 제공, 교육
 - 더 쉽고 빠르고 편리하고 강력한 프로그램 개발
- 예를 들어, 어려운 문법의 딥러닝 라이브러리인 Tensorflow도 현재는 Keras를 거쳐 Rstudio에서 거의 모든 기능을 사용할 수 있음.

R은 왜 배워야 하나요?

1. 일 잘하는 사람

- 암산 -> 주판 -> 엑셀 -> 프로그래밍
- 계산기 -> 도스 -> 윈도우 -> 프로그래밍
- 반복되는 작업을 컴퓨터에게 맡겨서 처리
- 프로그래밍 언어 사용자 중 최고 연봉 (미국)

2. 빠르게 변하는 세상과 교육 수준

- 구구단 -> 19단 -> 코딩 교육 열풍
- 처리해야 할 데이터는 급격하게 증가
- 커리어의 발전을 위해 오퍼레이션의 반복이 아닌 구조를 파악하여 자동으로 해결할 수 있는 접근법을 익혀야 함

M1X: Introduction & Basics

- **M11 intro**
 - 강의 계획서, Data Science와 R 프로그램에 대해 소개합니다.
- **M12 base**
 - base
 - Data Type과 Data Structure에 대해서 논의합니다.
- **M13 Quiz 1**
 - `rmarkdown`
 - M12 review (.docx)

M2X: The First Project - Carseat Sales

- 사용하기 편한 간단한 패키지들을 학습합니다.
- Carseats Sales 데이터를 이용해서 `rmarkdown`으로 작성된 report를 만듭니다.
- **M21 flow**
 - `dplyr`, `ggplot2`, `rmarkdown`, ISLR
 - ISLR의 Carseats 매출데이터로 자료 처리, 시각화, 문서화의 전 과정을 다룹니다.
- **M22 CarseatsReport**
 - `rmarkdown` (.html, .docx, .pdf)
 - 실제로 분석 결과를 보고서로 작성합니다.

- **M23 More on ggplot2**
 - ggplot2
 - 시각화 기법에 대해서 좀 더 깊이있게 다룹니다.
- **M24 ggplot2 Gallery**
 - ggplot2
 - 사용성이 높은 ggplot2를 이용한 아름다운 시각화 예제를 살펴봅니다.
- **M25 .pdf**
 - rmarkdown, texlive
 - 한글로 된 pdf문서를 작성할 수 있는 템플릿입니다. M22에 pdf 조판을 위한 textlive 프로그램 설치 과정을 진행해야 합니다.
- **M26 beamer**
 - rmarkdown, texlive
 - 한글로 된 프리젠테이션 문서(수업 강의노트)를 작성할 수 있는 템플릿입니다. M22에 pdf 조판을 위한 textlive 프로그램 설치 과정을 진행해야 합니다.
- **M27 slidy**
 - rmarkdown
 - rmarkdown을 사용하면 손쉽게 .html로 된 프리젠테이션 문서를 만들 수 있습니다. 이에 해당하는 slidy 포맷의 템플릿입니다.

M3X: Textmining & Web Applications

- **M3X** 부터는 dependency가 없습니다.
- 즉, **M1X**와 **M2X**까지를 익숙하게 다룰 수 있다면 **M3X**이후에는 모듈 각각에 대해서 독립적으로 이해하고 사용할 수 있습니다.
- Textmining 기법(워드클라우드, 크롤링, 번역)에 대한 예제를 제공합니다.
- Flexdashboard와 Shiny Application으로 구현해 봅니다.
- **M31 I have a dream**
 - `tm`, `KONLP`, `wordcloud`, `function`
 - 마틴루터킹 목사의 “I have a dream”에 대해 Wordcloud 분석을 하고, 이를 다양한 텍스트 파일 포맷과 한/영 문서에 대해서도 일반화 시킵니다. |
- **M32 flexdashboard**
 - `flexdashboard`, `rmarkdown`
 - `rmarkdown`을 다룰수 있다면 손쉽게 이를 `.html`의 대시보드로 제작할 수 있습니다. 이는 반응형 대시보드(사용자의 마우스/키보드 동작에 반응)로 연장되는 `shiny` 패키지의 기본이 됩니다.
- **M33 shiny on fd**
 - `shiny`, `flexdashboard`, `rmarkdown`
 - `flexdashboard`와 `rmarkdown`을 다룰수 있다면 손쉽게 반응형 대시보드(사용자의 마우스/키보드 동작에 반응)또한 구현할 수 있습니다.
 - `shiny`는 빠른 시간에 구현할 수 있는, elegant하고 professional한 프리젠테이션 기법이 됩니다.

- **M34 Happy B-day, ggplot**
 - `rvest`, `googleLanguageR`
 - `ggplot2`는 최근 10년에 걸쳐 데이터 시각화의 혁신을 가져왔습니다. 이에대한 영문 article을 web crawling하여 google 번역기를 이용해서 자동번역하는 프로그램을 제작합니다. |
- **M35 R and Python are joining**
 - `rvest`, `googleLanguageR`
 - 위의 **M34**와 같은 프로그램입니다. 데이터 사이언스에 있어서 R와 Python의 장단점과 둘의 융합이라는 트렌드에 대한 article을 역시 web crawling으로 얻어와 google 번역기를 자동으로 실행시켜 한글 문서로 제작합니다.

M4X: Case Studies - Data Analysis

- M41 longevity
 - **shiny**
- M42 subway
 -
- M43 retail
 -
- M44 SNS
 -
- M45 sales log
 -

M5X: Other Advanced Applications

- M51 tidyr
 - `tidyr`, SQL
- M52 Time Series
 - `xts`, `dygraph`, `lubridate`
- M53 python
 - `feather`
- M54 KMA & Auto
 - `jsonlite`, `mailR`, `taskscheduleR`
- M55 Outro

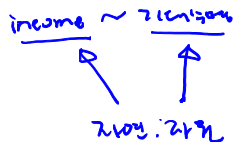
M6X: References

- Book
- Cheatsheets
- Dr. Hadley

M41-longevity (1/2)

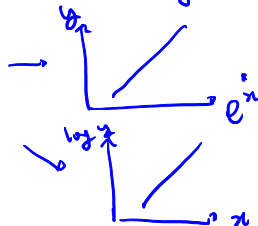
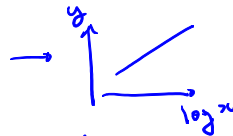
- 상관관계 (correlation) vs 인과관계 (causation)

- income \rightarrow 기대수명 : 인간 / 동물
- 부모수명 \rightarrow 자녀수명 : 인간 / 동물
- 쌍둥이 형 \rightarrow 쌍둥이 동생 : 인간 / 동물



✓ $Y = \beta_0 + \beta_1 X + \epsilon$

- $y = ax + b$ $r \rightarrow \Delta x$
- β_1 기울기
- β_0
- ϵ



M41-longevity (2/2)

수강생 Survey (1)

1. 이름, 이메일
2. 프로그래밍 경험이 있으신가요?
3. 수강 후에 어떤 능력을 배워가고 싶으신가요?

수강생 Survey (2)

1. 생각하고 있는 프로젝트에 대해서 적어주세요.

- 주제 & 목적:
- 데이터셋 설명:
- 결과물 포맷 (보고서/대시보드):

2. 팀워크 진행 선호 여부에 대해서 아래 중에서 선택하고 적어주세요.

- 2.1 해당 주제를 혼자 진행하기를 원함
- 2.2 해당 주제 crew를 원함 (최대 인원을 적어주세요)
- 2.3 다른 이의 팀에 join하기를 원함 (어떤 주제의 프로젝트에 join하기를 원하시는 지 적어주세요)

3. 팀워크 가이드라인

- 호칭은 XX님으로; 상호 존칭 및 매너 엄수; 실례가 될 수 있는 질문 삼가
- .Rmd파일 공유 및 소통
- The best way to learn is to teach!
- 소통, 질문, 대답하는 분위기를 조성해주세요.

M11-intro