

## M11 – Introduction

`learningSpoonsR@gmail.com`



I. 강의 소개

II. 설치 및 준비

III. Rstudio?

IV. 논 의

V. 수업 과정

## I. 강의 소개

## 강의 소개

### 강의 일정 및 소통

- ▶ 강의 자료 페이지 <https://github.com/LearningSpoonsR/LS-DS>
- ▶ facebook 그룹
- ▶ [learingSpoonsR@gmail.com](mailto:learingSpoonsR@gmail.com) (강사 이메일)
- ▶ 김형종 매니저

## 프로젝트 소개

수강생 프로젝트 소개 <http://bitly.kr/1G0Q>

### M41으로 보는 강의 목표

▶ 상관관계 (correlation) vs 인과관계 (causation)

1. income  $\rightarrow$  기대수명
2. 부모수명  $\rightarrow$  자녀수명
3. 쌍둥이 형  $\rightarrow$  쌍둥이 동생

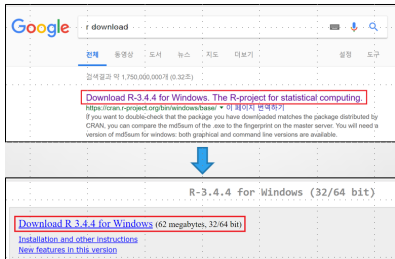
▶  $Y = \beta_0 + \beta_1 X + \epsilon$

1.  $y = ax + b$
2.  $\beta_1$
3.  $\beta_0$
4.  $\epsilon$

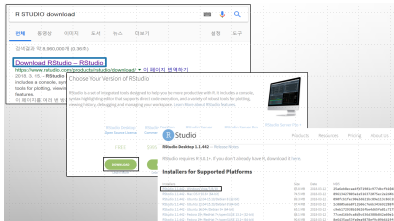
## II. 설치 및 준비

## 설치 및 준비 (1/5)

0. 아래 과정을 순서대로 진행해 주세요.
1. google에서 “R download”를 검색
2. OS에 맞는 최신 버전 설치



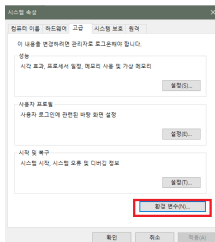
3. google에서 “R Studio download”를 검색
4. OS에 맞는 최신 버전 설치



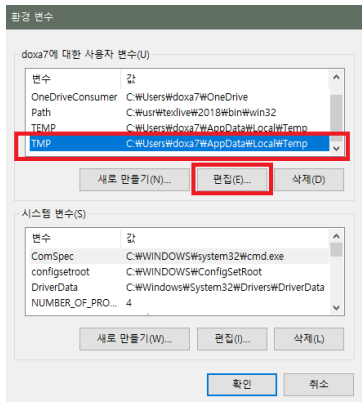
5. 윈도우 사용자 계정이 한글이라면 6번으로, 영어라면 14번으로 점프

## 설치 및 준비 (2/5)

6. 제어판 → 시스템 및 보안 클릭 → 시스템 클릭 → 고급 시스템 속성 클릭 → 시스템 속성 창이 뜸
7. (윈도우 버전이나 시스템 설정에 따라서 과정이 다를 수 있습니다. 중요한 것은 아래 화면을 띄워야 합니다.)
8. “환경변수” 클릭



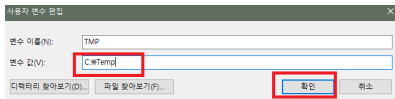
9. 사용자 변수 **TMP**의 값이 전부 다 영어로 되어있어야 합니다.
10. 해당 부분이 전부다 영어라면 14번으로 점프



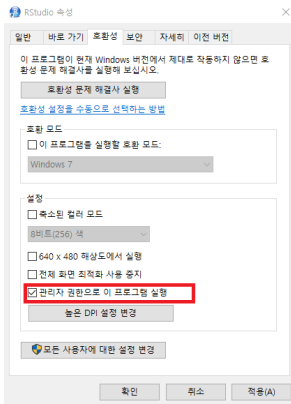


## 설치 및 준비 (3/5)

11. TMP의 값에 한글이 포함되어 있다면 탐색기에서 C:\Temp라는 디렉토리를 만듭니다.
12. 위의 그림의 환경변수 윈도우에서 편집을 누릅니다.
13. 아래 화면처럼 C:\Temp를 입력합니다.



14. RStudio 아이콘을 찾아서 우클릭 → 호환성 탭 → 관리자 권한으로 이 프로그램 실행을 체크



## 설치 및 준비 (4/5)

## 15. R Studio 프로그램을 실행합니다.

**현재 변수 및 환경**

**최근 명령 (Command)**

```

ggplot.Rmd
1
2 title: "Rich Country = Live Longer?"
3 runtime: shiny
4 output:
5   flexdashboard::flex_dashboard:
6     source: embed
7
8
9
10 [r setup, include=FALSE]
11 library(flexdashboard)
12 library(ggplot2)
13

```

**HELP 창**

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.

> |

**R과 CPU의 통신 기록**

**탐색기 기능**

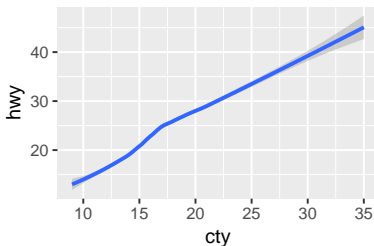
**그래프**

File	Size	Modified
Rhistory	23.5 KB	Mar 12, 20
data	870.5 KB	Feb 18, 20
ggplot.html	3.9 KB	Feb 18, 20
ggplot.R	2.6 KB	Apr 14, 20
ggplot.Rmd	2.6 KB	Apr 14, 20
rconnect	14.6 KB	Feb 20, 20
script.docx	204 B	Feb 19, 20
새 텍스트 문서.txt	204 B	Feb 19, 20

## 설치 및 준비 (5/5)

16. 왼쪽 혹은 왼쪽 아래에 프롬프트 표시 (>)가 있는 화면이 보입니다.
17. 여기에 아래 명령어를 차례로 입력합니다. 아래와 같은 그림이 보이나요?

```
install.packages("ggplot2")
library(ggplot2)
ggplot(mpg, aes(cty, hwy)) + geom_smooth()
```



18. Tool → Global Option → Appearance
  - ▶ Editor Font Size: 14
  - ▶ Editor Theme: "Tomorrow Midnight Blue"
19. 향후에 한글이 깨질때는 File → Reopen with Encoding → UTF-8을 시도해보세요. 만약에 안되면 맨 마지막에서 CP-949를 선택하세요. 그래도 안되면 질문주세요!
20. 설치 및 기본 설정 끝!

## III. Rstudio?

**현재 변수 및 환경**

**최근 명령 (Command)**

**HELP 창**

**R과 CPU의 통신 기록**

**탐색기 기능**

**그래프**

```

1 ggplot.Rmd
2 title: "Rich Country = Live Longer??"
3 runtime: shiny
4 output:
5   flexdashboard::flex_dashboard:
6     source: embed
7
8
9 {r setup, include=FALSE}
10 library(flexdashboard)
11 library(ggplot2)
12
13
  
```

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.

> |

File	Size	Modified
.Rhistory	23.5 KB	Mar 12, 20
data	870.5 KB	Feb 18, 20
ggplot.html	3.9 KB	Feb 18, 20
ggplot.R	2.6 KB	Apr 14, 20
ggplot.Rmd	14.6 KB	Feb 20, 20
rsconnect	204 B	Feb 19, 20
script.docx		
새 텍스트 문서.txt		

## 1. 좌상 패널 (Editor)

- ▶ 메모장과 같은 기능을 제공합니다. (편집 및 저장)
- ▶ **Ctrl + Enter**
  - ▶ 커서가 있는 라인
  - ▶ 선택된 블록이 있을 경우 그것을 실행합니다.
- ▶ **Ctrl + 1**
  - ▶ 다른 패널에서 Editor 패널로 이동합니다.
- ▶ **Ctrl + Shift + 1**
  - ▶ Editor 패널을 전체화면으로 표시합니다.

## 2. 좌하 패널 (Console)

- ▶ DOS창 처럼 R의 명령어를 실행할 수 있습니다.
- ▶ **Ctrl + 2**
  - ▶ 다른 패널에서 Console 패널로 이동합니다.
- ▶ **Ctrl + Shift + 2**
  - ▶ Console 패널을 전체화면으로 표시합니다.
- ▶ **Terminal** 탭은 R와 CPU의 통신 기록을 보여줍니다.

## 3. 우상 패널

- ▶ 현재 메모리 상태등을 보여줍니다.

## 4. 우하 패널

- ▶ 파일 탐색기, 출력된 그래프, HELP등의 기능을 제공합니다.

아래 코드를 콘솔(>가 있는 프롬프트 창)에 입력하세요.

```
"Hello"  
a <- 10  
a <- a + 5  
a
```

▶ 에러가 없이 수행되고 아래의 결과가 나오시나요?

```
## [1] 15
```

- ▶ 우상단 패널에서 “History” 탭을 누르면 지금까지 실행한 명령을 볼 수 있습니다.
- ▶ 이들을 카피해서 좌상단 Editor 패널에 입력하세요.
- ▶ **File** → **Save As**를 이용해서 **first.R**로 저장하세요.
- ▶ Rstudio를 종료하고 윈도우 탐색기에서 **first.R**을 찾아 더블클릭하세요.

## RStudio Cheatsheet (1)

## RStudio IDE : : CHEAT SHEET

## Documents and Apps



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

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

Jump to previous chunk Run selected chunk Publish selected chunk to server outline

Access markdown guide at Help > Markdown Quick Reference

Set knitr chunk options Run this chunk code chunk

RStudio recognizes that this name is app.R, server.R, ui.R, and global.R belonging to a shiny app

Run Choose Publish to Manage app location to shinyapps.io view app or server accounts

## Write Code

Navigate Open in new window Save Find and replace Compile as notebook Run selected code

Multiple cursors/column selection with Alt + mouse drag

Source with or without Echo

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 Pins > to use command history

Maximize, minimize panes Drag panes

## R Support

Import data History of past commands to File > New File > R Presentation

Save Delete all workspace Search inside environment

Choose environment to display from list of parent environments

Display objects as list or grid

View in data viewer View function source code

Displays saved objects by type with short description

Create Upload Delete Rename File folder Set as Working Directory Click on file or directory name to open.

Link to displayed directory

A file browser keyed to your working directory.

## Pro Features

Share Project Active shared collaboration Start new R Session in current project

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.

RStudio opens plots in a dedicated Plots pane

Navigate recent plots Export Delete Delete plot plot

GUI Package manager lists every installed package

Install Update Create reproducible package Packages library for your project

Click to load package with library(). Unlick to detach package with detach()

Viewer Pane displays HTML content, such as Shiny apps, Jupyter notebooks, and interactive visualizations

Stop Shiny app Publish to shinyapps.io Refresh 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 current environment

Select function in traceback to debug

Launch debugger mode from origin

Open traceback to examine the functions that the error called before the error occurred

Console

Error in get\_digits(x) : Show Traceback

Console

Next Continue Stop

Step through code one line at a time

Step into and out of functions to run

Resume execution mode

Quit debug

## Version Control with Git or SVN

Turn on at Tools > Project Options > Git/SVN

Stage Show file Commit Push/Pull staged files to remote View History

Added Deleted Modified Renamed Untracked

Open shell to type commands

## Package Writing

File > New Project > New Directory > R Package

Turn project into package. Enable roxygen documentation with Tools > Project Options > R Package

Roxygen guide at Help > Roxygen Quick Reference

View<data> opens spreadsheet like view of data set

Filter rows by value < or value range

Sort by value

Search for value



## RStudio Cheatsheet (2)

## 1 LAUNCH

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

## 2 RUN CODE

## Search command history

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

## Interrupt current command

## Clear console

## Restart R Session

## Run current line/selection

## Run current (retain cursor)

## Run from current to end

## Run the current function

## Source a file

## Source the current file

## Source with echo

## 3 NAVIGATE CODE

## Goto File/Function

## Fold Selected

## Unfold Selected

## Fold All

## Unfold All

## Go to line

## Jump to

## Switch to tab

## Previous tab

## Next tab

## First tab

## Last tab

## Navigate back

## Navigate forward

## Jump to Brace

## Select within Braces

## Use Selection for Find

## Find in Files

## Find Next

## Find Previous

## Jump to Word

## Jump to Start/End

## Toggle Outline

## Windows/Linux

Ctrl+1  
Ctrl+2  
Ctrl+3  
Ctrl+4  
Ctrl+5  
Ctrl+6  
Ctrl+7  
Ctrl+8  
Ctrl+9  
Ctrl+0

## Mac

Cmd+1  
Cmd+2  
Cmd+3  
Cmd+4  
Cmd+5  
Cmd+6  
Cmd+7  
Cmd+8  
Cmd+9  
Cmd+0

## Windows/Linux

Ctrl+.  
Alt+L  
Shift+Alt+L  
Alt+O  
Shift+Alt+O  
Shift+Alt+G  
Shift+Alt+J  
Ctrl+Shift+.  
Ctrl+F11  
Ctrl+F12  
Ctrl+Shift+F11  
Ctrl+Shift+F12  
Ctrl+F9  
Ctrl+F10  
Ctrl+P  
Ctrl+Shift+Alt+E  
Ctrl+F3  
Ctrl+Shift+F  
Win: F3, Linux: Ctrl+G  
W: Shift+F3, L:  
Ctrl+Shift+O

## Mac

Ctrl+1  
Ctrl+2  
Ctrl+3  
Ctrl+4  
Ctrl+5  
Ctrl+6  
Ctrl+7  
Ctrl+8  
Ctrl+9  
Ctrl+0  
Cmd+1  
Cmd+2  
Cmd+3  
Cmd+4  
Cmd+5  
Cmd+6  
Cmd+7  
Cmd+8  
Cmd+9  
Cmd+0  
Cmd+Shift+F10  
Cmd+Enter  
Alt+Enter  
Ctrl+Alt+E  
Ctrl+Alt+F  
Ctrl+Alt+G  
Ctrl+Shift+S  
Ctrl+Shift+Enter

## 4 WRITE CODE

## Attempt completion

## Navigate candidates

## Accept candidate

## Dismiss candidates

## Undo

## Redo

## Cut

## Copy

## Paste

## Select All

## Delete Line

## Select

## Select Word

## Select to Line Start

## Select to Line End

## Select Page Up/Down

## Select to Start/End

## Delete Word Left

## Delete Word Right

## Delete to Line End

## Delete to Line Start

## Outdent

## Yank line up to cursor

## Yank line after cursor

## Insert yanked text

## Insert %%

## Show help for function

## Show source code

## New document

## New document (Chrome)

## Open document

## Save document

## Close document

## Close document (Chrome)

## Close all documents

## Extract function

## Extract variable

## Reindent lines

## (Un)Comment lines

## Refactor Comment

## Reformat Selection

## Select within braces

## Show Diagnostics

## Transpose Letters

## Move Lines Up/Down

## Copy Lines Up/Down

## Add New Cursor Above

## Add New Cursor Below

## Move Active Cursor Up

## Move Active Cursor Down

## Find and Replace

## Use Selection for Find

## Replace and Find

## Windows/Linux

Tab or Ctrl+Space  
Enter, Tab, or  
Esc  
Cmd+Z  
Cmd+Shift+Z  
Ctrl+X  
Ctrl+C  
Ctrl+V  
Ctrl+A  
Cmd+D  
Shift+(Arrow)  
Ctrl+Shift+←  
Alt+Shift+←  
Shift+PageUp/Down  
Shift+Alt+↑/↓  
Ctrl+Backspace  
Ctrl+Shift+←  
Alt+Shift+←  
Shift+PageUp/Down  
Ctrl+Opt+Backspace  
Option+Delete  
Ctrl+K  
Option+Backspace  
Tab (at start of line)  
Shift+Tab  
Ctrl+J  
Ctrl+K  
Ctrl+Y  
Option+.  
Ctrl+Shift+M  
Cmd+Shift+M

## Mac

Tab or Cmd+Space  
Enter, Tab, or  
Esc  
Cmd+Z  
Cmd+Shift+Z  
Cmd+X  
Cmd+C  
Cmd+V  
Cmd+A  
Cmd+D  
Shift+(Arrow)  
Option+Shift+←  
Cmd+Shift+←  
Cmd+Shift+→  
Cmd+Opt+Backspace  
Option+Delete  
Ctrl+K  
Option+Backspace  
Tab (at start of line)  
Shift+Tab  
Ctrl+J  
Ctrl+K  
Ctrl+Y  
Option+.  
Ctrl+Shift+M  
Cmd+Shift+M

## WHY RSTUDIO SERVER PRO?

RSP extends the open source server with a commercial license, support, and more:

- open and run multiple R sessions at once
- tune your resources to improve performance
- edit the same project at the same time as others
- see what you and others are doing on your server
- switch easily from one version of R to a different version
- integrate with your authentication, authorization, and audit practices

Download a free 45 day evaluation at  
[www.rstudio.com/products/rstudio-server-pro/](http://www.rstudio.com/products/rstudio-server-pro/)

## 5 DEBUG CODE

## Toggle Breakpoint

## Execute Next Line

## Step Into Function

## Finish Function/Loop

## Continue

## Stop Debugging

## Ctrl+F9

## F10

## Shift+F4

## Shift+F6

## Shift+F5

## Shift+F8

## Ctrl+Alt+D

## Ctrl+Alt+M

## Ctrl+Option+M

## Ctrl+Shift+T

## Ctrl+Opt+F

## Ctrl+Shift+E

## Cmd+Shift+D

## Ctrl+Shift+B

## Cmd+Shift+B

## Ctrl+Shift+L

## Cmd+Shift+L

## Ctrl+Shift+T

## Cmd+Shift+T

## Ctrl+Shift+H

## Cmd+Shift+H

## Ctrl+Alt+I

## Cmd+Option+I

## Ctrl+Shift+R

## Cmd+Shift+R

## Ctrl+Alt+P

## Cmd+Shift+P

## Ctrl+Alt+B

## Cmd+Option+B

## Ctrl+Alt+T

## Cmd+Option+T

## Ctrl+Alt+C

## Cmd+Option+C

## Ctrl+Alt+N

## Cmd+Option+N

## Ctrl+F8

## Cmd+F8

## Ctrl+Alt+F11

## Cmd+Option+F11

## Ctrl+Alt+F12

## Cmd+Option+F12

## Alt+Shift+K

## Option+Shift+K



## Cheatsheet?

- ▶ 모든 문법과 명령어를 외우는 것은 불가능합니다.
  - ▶ Cheatsheet은 사용법이 정리되어 있는 문서입니다.
  - ▶ 경험을 바탕으로 Cheat Sheet과 웹 검색을 하면서 프로그래밍을 합니다.
  - ▶ “Know-how” 보다 “Know-where”가 중요하다는 얘기 들어보셨죠?
- ▶ English?
  - ▶ 검색 결과와 설명서 등은 대부분 영어로 되어있습니다.
  - ▶ Cheat Sheet을 가득 채운 영어가 겁나시나요?
  - ▶ 그러나 컴퓨터와 프로그래밍에 관련된 용어이며,
  - ▶ 약 1000개만 익히면 되므로 경험과 함께 금방 익숙해 집니다.
  - ▶ 프로그래밍에 관련된 설명은 구글/네이버 번역기으로도 쉽게 이해할 수 있습니다.

## IV. 논 의

## R vs Excel

### ▶ R에서는

1. 한 번 했던 작업을 쉽게 재현할 수 있습니다.
2. 처음에는 어렵고 오래걸릴 수 도 있습니다.
3. 익숙해지면 **빠르고 정확하게** 분석을 할 수 있습니다.
4. 데이터 구조와 특성에 대해서 **논리적으로** 사고하게 됩니다.

-	Excel	R
인터페이스	GUI	Console
결과 재현	반복해서 작업	정확한 재현 가능 (Reproducible)
난이도	쉬움	초반의 학습 곡선
속도, 용량	느리고 한정적	빠르고 대용량
실수	관대함	결과물의 실수 없음이 보장
장점	눈에 보여서 직관적	추상적, 논리적 사고가 요구됨

## R vs 다른 언어들

### ▶ R Studio의 Support

1. 안정적인 개발 환경을 제공합니다.
2. 확장 프로그램들을 통합하여 관리합니다.
3. 쉽고 직관적인 메뉴얼을 제공합니다.

### ▶ 다른 언어들과 비교

- ▶ MATLAB: R과 가장 유사하며 수리 계산에 강점. 유료.
- ▶ SAS: 초고용량 데이터 처리. 매우 유료.
- ▶ C, C++: 배우기 어렵지만 빠르고 다른 언어의 토대.
- ▶ JAVA: C와 유사하지만 주로 웹개발 용으로 쓰임.
- ▶ Python
  - ▶ 분석과 개발의 양쪽을 지원하여 인기가 높음
  - ▶ R에 비해서 난이도, 시각화, 문서화가 약점

## R은 왜 배워야 하나요?

### ▶ 시대의 흐름

- ▶ 데이터에 기반한 분석과 의사결정이 필수인 시대
- ▶ 반복작업은 최대한 컴퓨터에게 맡겨야 하는 시대
- ▶ 분석과 의사 결정을 하는 도구는 시대에 따라 진화
- ▶ 암산 → 주판 → 엑셀 → 프로그래밍
- ▶ 불과 20년 전에만 해도 엑셀 1급이면 전문가였습니다. 지금은 모두가 사용합니다.
- ▶ 스마트폰에서 사진을 찍고 카톡에 첨부해서 보내는 기능도 예전에는 생소했습니다.

### ▶ 지적인 발전

- ▶ 유소년 교육도 구구단 → 19단 → 프로그래밍으로 트렌드 변화
- ▶ 커리어 발전을 위해 다루는 데이터 구조를 파악하고 자동으로 처리해야 합니다.
- ▶ 비전공자가 처음으로 배우기에 가장 쉽습니다.

## V. 수업 과정

## 목표

1. R 프로그래밍을 배웁니다.
2. 데이터 분석 프로젝트의 흐름을 익힙니다.
3. 기수강생 결과물 수준의 결과물을 만들어 냅니다.
4. Take it to your workplace!

## 프로젝트 과정

1. 분석
  - ▶ Design (가설 설정 및 디자인)
  - ▶ Analysis (데이터 분석)
2. 공유
  - ▶ Visualization (시각화)
  - ▶ Documentation (문서화)



## 강의 순서

주제	Module	Note
0. 강의 소개	M11, M41	.
1. 기본 문법	M12, M13	.
2. 첫 번째 프로젝트	M21, M22, M23	분석, 시각화, 문서화
3. 문서화 templates	M24, M26, M28	pdf, docx
4. advanced 시각화	M24	.
5. advanced 문서화	M32	html 대시보드
6. 실무 프로젝트	M43, M46	.
7. 데이터 다루기	M51, M52	전처리, 시계열 데이터
8. advanced 문서화	M33	동적인 html

## Appendix: 기타 가능한 설정

### 1. Tool → Global Option →

- ▶ **Appearance:** 폰트와 색상 등을 조정
- ▶ **PANE Layout:** 모니터 크기와 작업 목적에 따라 조정
- ▶ **CODE -> Saving -> Text Encoding in -> UTF-8:** 한글이 깨지는 현상을 방지합니다.

### 2. 한영 전환

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





## Survey

1. 이름, 이메일
2. 전공
3. 프로그래밍 경험
4. 직무 소개와 수강후에 직무에 어떻게 도움이 되리라 기대하는지 말씀 부탁드립니다.





blank