2. R 프로그래밍 기초, 설치, 패키지 관리

R이란?

R은 데이터분석을 위한 Open Source를 말하며 하기와 같은 다양한 의미로 사용되고 있다.

오픈소스 데이터 분석 S/W

- ❑ 오픈소스 형태의 데이터 분석 소프트웨어이다.
- ❑ 데이터 사이언티스트를 위한 개발 플랫폼이다.

R언어 커뮤니티

- ❑ 오픈소스 소프트웨어 프로젝트이다.
  - Free, Open, Active
- ❑ 수천~수만명의 참여자가 있는 커뮤니티이다.
- ❑ 커뮤니티를 통하여 다양한 분야의 리소스와 지원을 받을 수 있다.

데이터 분석 환경

- ❑ 방대한 분야의 알고리즘 라이브러리
  - 데이터 접근/수집/불러오기 등
  - 데이터 전처리(manipulation)
  - 데이터 분석
  - 데이터 가시화

R언어의 특성

R은 데이터분석을 위한 Open Source로서 하기와 같은 다양한 특성을 갖고 있다.

❑ **오픈 소스이다.**

- 통계 분석용 S 언어의 오픈소스 버전이다.

❑ **배우기 쉬운 스크립트 언어이다.**

- 인터프리터 언어로 다른 언어에 비해 배우기 쉽다.
- 빅데이터를 대상으로 데이터 분석 작업에 편리하며, 다양한 가시화 함수를 제공한다.

❑ **글로벌 측면에서 사용자가 많아 필요한 자료를 구하기 쉽다.**

- 마케팅, 금융, 제조, 공공 등 다양한 분야의 사용자가 있어 조언을 구하기가 쉽다.

❑ **기능별 다양한 Package를 제공한다.**

- 동일한 기능을 수행하는 다양한 Package가 있어 본인에 맞는 것을 설치/사용해야 한다.

R프로그램 설치 및 환경 설정

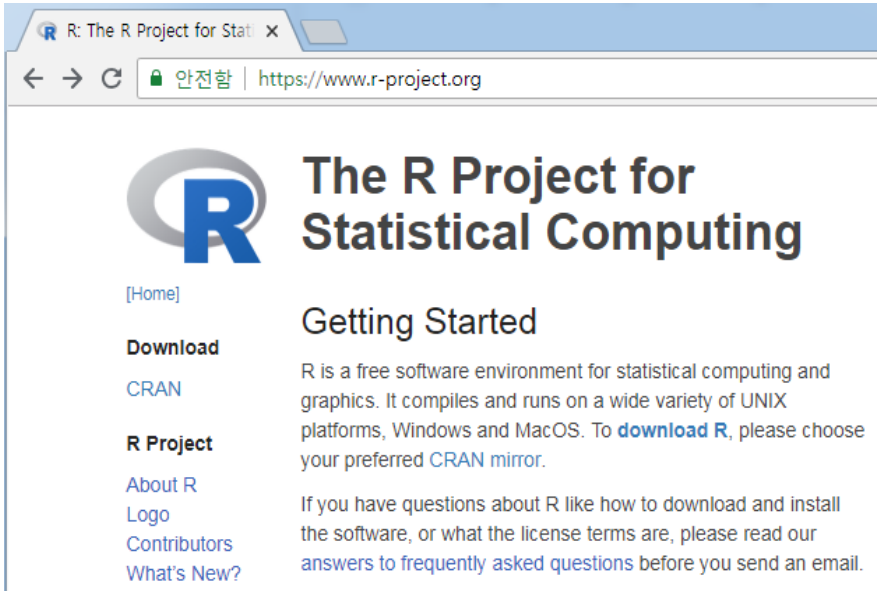
❑ R 프로그램 설치

- 미국 사이트

http://www.r-project.org/

- Korea Mirror 사이트

https://cran.seoul.go.kr/



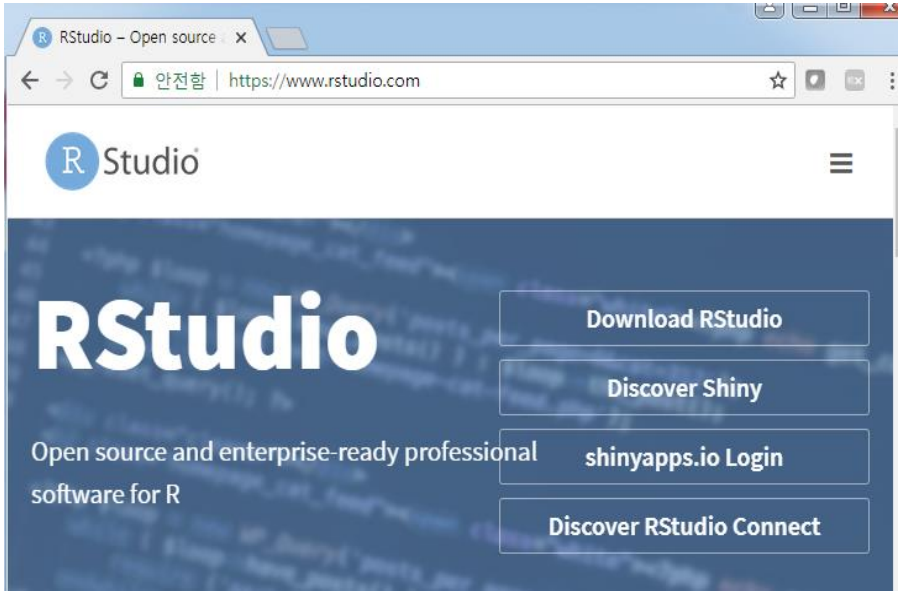
❑ R코드 Editor 설치

- RStudio

https://www.rstudio.com

- RAnalyticFlow

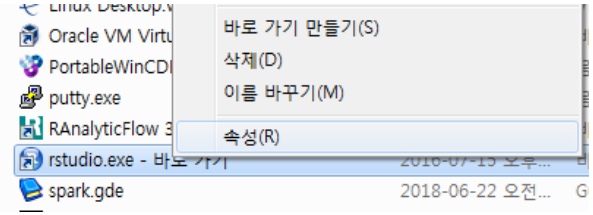
https://r.analyticflow.com/en/



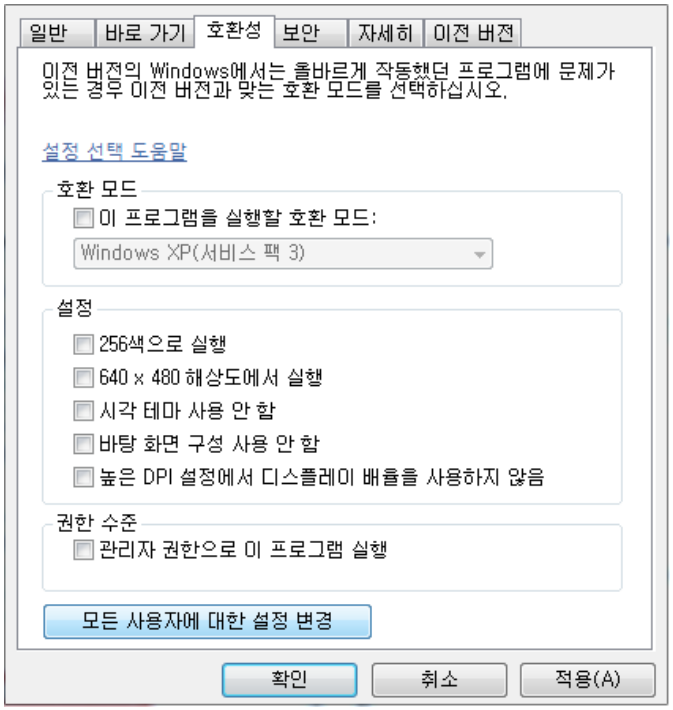
R프로그램 설치 및 환경 설정

Rstudio를 설치한 후에 하기와 같이 환경 설정을 실행한다(R이 패키지를 다운받아 압축풀 때 권한이 필요함).

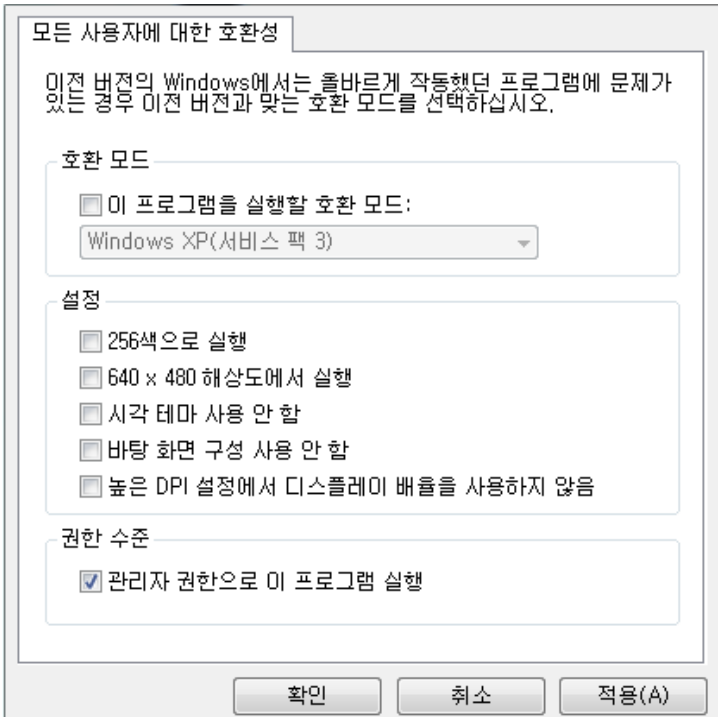
❑ **Rstudio 마우스 우클릭 → 속성 클릭**



❑ **호환성탭 → 모든 사용자에 대한 설정 변경 클릭**

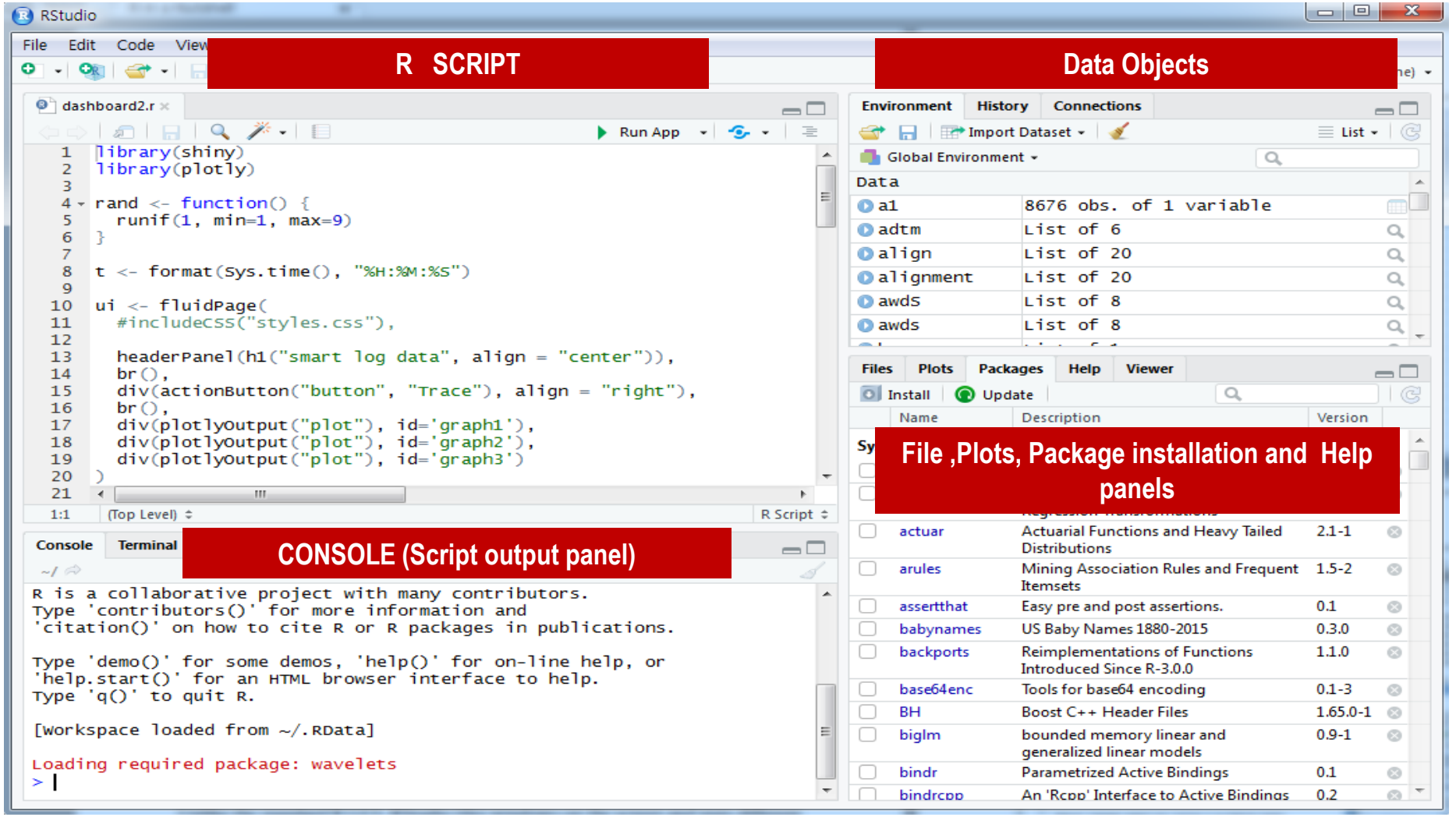


❑ **관리자 권한으로 이 프로그램 실행 체크 → 확인**



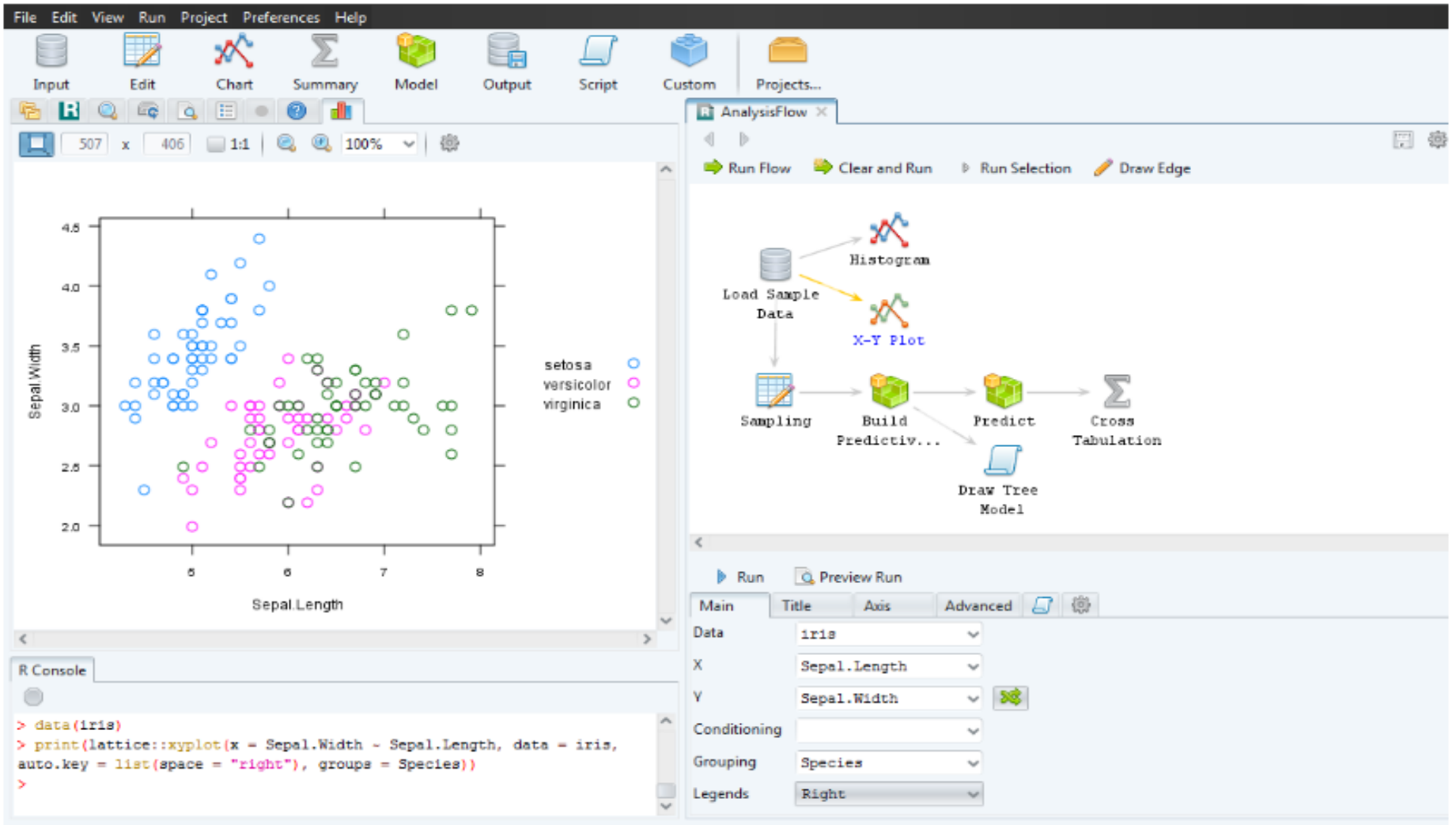
Rstudio 화면

Rstudio를 설치한 후에 실행한 화면은 하기와 같다.



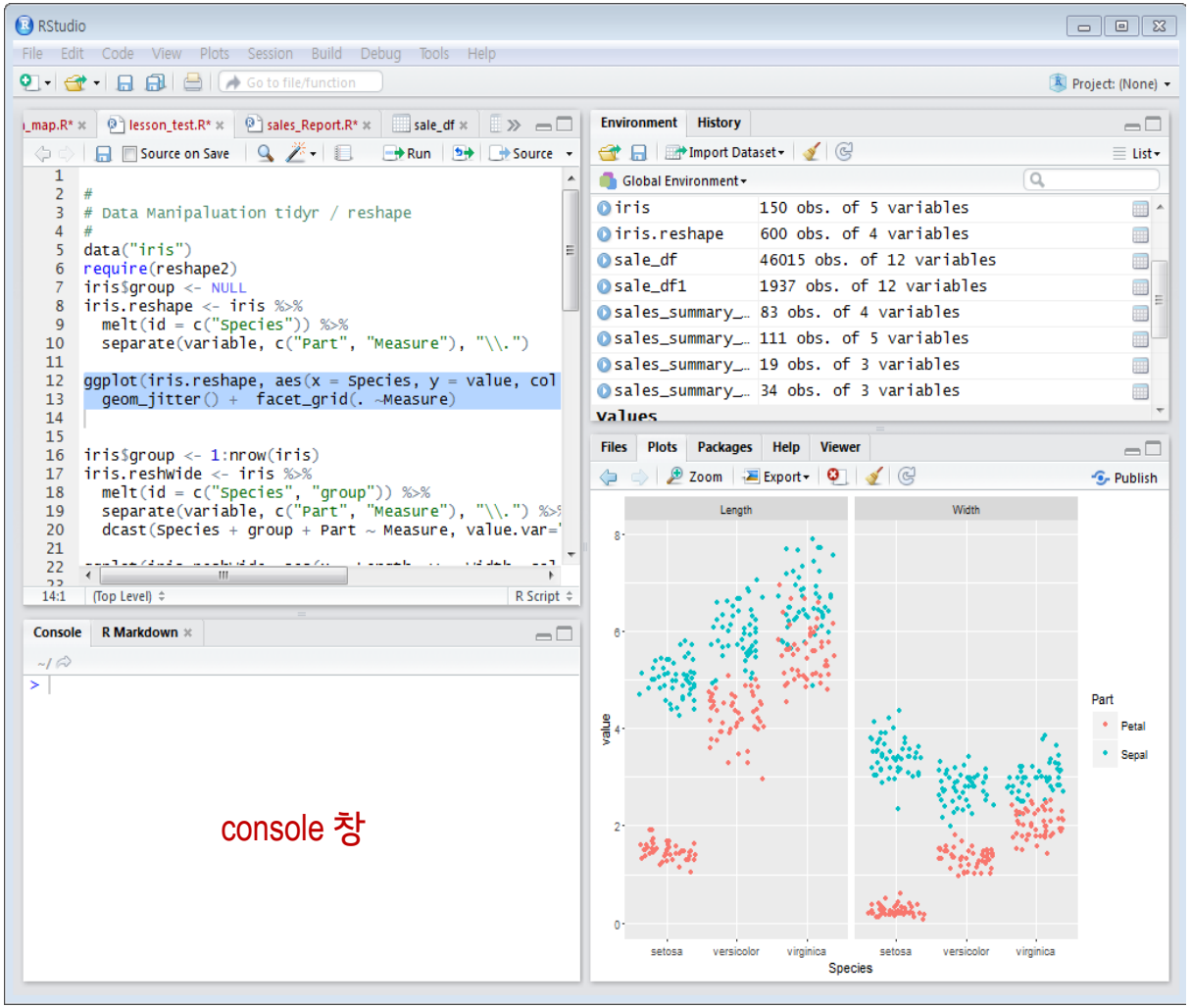
RAnalyticFlow화면

RAnalyticFlow를 설치한 후에 실행한 화면은 하기와 같다.



필요한 패키지 설치

Console 창에서 아래와 같은 명령을 입력하여 필요한 Package를 설치한다.




```
install.packages("dplyr")

install.packages("shiny")

install.packages("ggplot2")

install.packages("plotly")

install.packages("RODBC")

install.packages("lubridate")

install.packages("tidyr")

install.packages("xlsx")
```

R Upgrade하기

R을 사용하다 보면 신규 R버전으로 Upgrade를 해야 할 필요가 있는데 손쉽게 하기 위한 방법은 하기와 같다.

❑ 현재 설치된 버전 정보 확인하기

```
> version
platform       x86_64-w64-mingw32
arch           x86_64
os             mingw32
system         x86_64, mingw32
status
major          3
minor          4.1
year           2017
month          06
day            30
svn rev        72865
language       R
version.string R version 3.4.1 (2017-06-30)
nickname       Single Candle
```

❑ R을 최신버전으로 업데이트 하기

```
## 관련 패키지를 설치한다.
install.packages("installr")
## 설치한 패키지를 불러온다.
library(installr)
## R 업데이트 명령을 실행한다.
updateR()
## updateR()명령을 실행하면 몇가지 질문창이 나타나며
## 본인에 맞는 옵션을 선택하여 설치하면 된다.
## 메시지창에서 Rgui에서 update하는것을 권장하는데
## Rstudio에서도 문제없이 update할 수 있다.
```

R Upgrade하기

Upgrade 과정중에 기존 버전의 packages를 신규버전으로 복사할 것인지와 기존 버전 내용은 삭제할 것인지를 묻는 화면이 나타나며 본인에 맞게 선택한 후 Upgrade를 진행하면 된다.

❑ Upgrade 진행 화면

```
> updateR()
Installing the newest version of R,
 please wait for the installer file to be download and executed.
 Be sure to click 'next' as needed...
trying URL 'https://cran.rstudio.com/bin/windows/base/R-3.5.0-win.exe'
Content type 'application/x-msdos-program' length 83351952 bytes (79.5 MB)
downloaded 79.5 MB


The file was downloaded successfully into:
 C:\Users\ADMINI~1\AppData\Local\Temp\RtmpsFjEHv/R-3.5.0-win.exe

Running the installer now...

Installation status:  TRUE . Removing the file:
 C:\Users\ADMINI~1\AppData\Local\Temp\RtmpsFjEHv/R-3.5.0-win.exe
 (In the future, you may keep the file by setting keep_install_file=TRUE)
-----------------------
I am now copying  141  packages from: d:/PROGRA~2/R/R-3.4.1/library  ; into:  d:/PROGRA~2/R/R-3.5.0/library----------
------------
=====================
Done. We finished copying all the packages to the new location
Next: we will remove the packages from the old R installation ('FROM')
Done. The old packages were deleted.
```

기본 명령어

R과 Rstudio을 사용하는데 있어 알고 있어야 하는 필수 명령어는 하기와 같다.

Rstudio 단축키

❑ **Tab 키**

- 명령어 자동 완성 기능
- 명령어 일부를 작성후 Tab 키를 누르면 Rstudio가 함수나 파일이름을 제안함

❑ **Ctrl + 위쪽 화살표(↑) 키**

- 기존에 입력했던 모든 명령어 조회
- 원하는 명령을 선택후 Enter키를 누름

❑ **Ctrl + Enter 키**

- 편집기의 현재 코드 라인을 받아 콘솔로 보낸 후 실행됨
- 여러행의 코드를 선택하는 경우 선택한 모든 행의 코드가 실행됨

R 기본 명령어

❑ **작업디렉토리 설정**

- setwd("d:/lge/Rprojects")

❑ **패키지 관련**

- 설치 : install.packages("thepackagename")
- 설치된 패키지 확인 : installed.packages()
- 설치된 패키지 사용 : library("thepackagename")
- 설치된 패키지의 최신버전 다운로드 : update.packages()
- 설치된 패키지 삭제 : remove.packages("thepackagename")

❑ **도움말 관련**

- ?functionName
- help(functionName)
- example(functionName)
- help.search("your search term")

기본 명령어

R과 Rstudio을 사용하는데 있어 알고 있어야 하는 필수 명령어는 하기와 같다.

R 기본 명령어

❑ 객체 관련

- length(object) # 객체 구성요소의 개수
- str(object) # 객체의 구조
- class(object) # 객체의 유형
- names(object) # 객체의 이름
- c(object,object,...) # 객체를 벡터로 연결
- cbind(object, object, ...) # 객체를 열로 연결
- rbind(object, object, ...) # 객체를 행으로 연결
- object # 객체를 출력
- ls() # 현 객체의 리스트 출력
- rm(object) # 해당 객체 삭제 rm(list=ls()
- newobject <- edit(object) # 객체를 편집후 newobject에 저장
- fix(object) # 객체를 편집후 object에 저장
- summary(object) # 객체의 기술 통계값을 계산함

R 기본 명령어

❑ 수학 연산자

+, -, *, ^, %*%, %/%

❑ 관계 연산자

>, >=, <, <=, ==, !=

❑ 논리 연산자

!, &, |

❑ 할당 연산자

<-, =

❑ 리스트 인덱싱 연산자

$

❑ 시퀀스 연산자

:

기본 명령어

R과 Rstudio을 사용하는데 있어 알고 있어야 하는 필수 명령어는 하기와 같다.

기타

- save() : 외부 .Rdata파일로 현재 생성한 R객체를 저장함
- load() : 외부에 저장된 .Rdata파일에서 R객체를 읽어옮
- boxplot() : 박스플롯을 그려줌
- hist() : 히스토그램을 그려줌
- var() : 분산을 계산함
- sd() : 표준편차를 계산함
- table() : 일원배치표 입력한 항목의 빈도수를 계산함
- prob.table() : 일원배치표 입력한 항목의 빈도를 %로 계산함
- plot() : x, y좌표상의 점으로 데이터를 표시함
- head() : 데이터 객체의 첫번째 6라인만을 보여줌
- typeof() : 데이터의 형태를 확인함
- is.integer() : 데이터가 정수인지 확인함
- as.integer() : 데이터를 정수로 형변환함
- seq() : 일정하게 증가하는 데이터 생성

기타

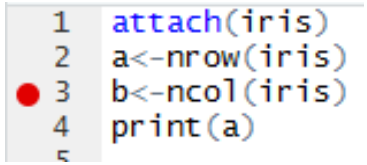
- c() : 기존 벡터를 결합하여 새로운 벡터를 생성함
- rep() : 벡터를 반복시켜 새로운 벡터를 생성함
- sample() : 랜덤한 숫자를 발생시킴
- sort() : 순서대로 정렬을 함
- which() : 조건에 맞는 벡터의 원소 인덱스를 알려줌

Debug

Rstudio를 이용하여 R코드를 디버깅하는 방법은 하기와 같다.

❑ Breakpoints를 설정

- 코드 실행중 결과를 모니터링하고 싶은 위치로 이동한다.
- 해당 위치에 중단점을 설정한다.
  (마우스로 클릭하거나 Shift+F9를 입력함)
- 중단점이 제대로 설정되었으면 해당 행에 빨간색 원이 표시된다.

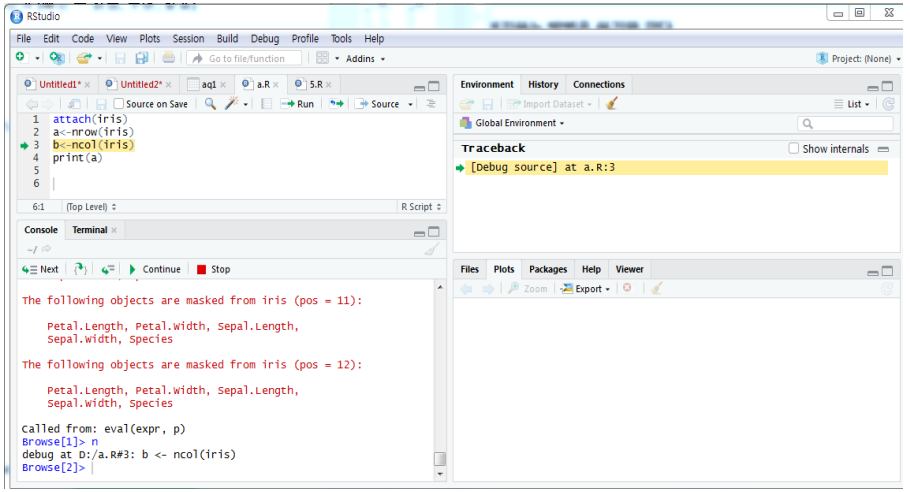


❑ Breakpoints위치까지 R코드를 실행

- 해당 코드를 파일로 저장한다.
- 툴바에 위치한 Source Button을 눌러 실행한다.

❑ Debug 모드를 실행

- 코드실행중 Rstudio가 중단점을 만나게 되면 디버그 모드가 된다.



❑ Debug 모드를 진행

- Next(F10) : 다음코드 실행
- Continue(Shift+F5) : 중단점 만난때까지 실행
- Stop(Shift+F8) : 코드 실행 중단, 디버그모드 종료

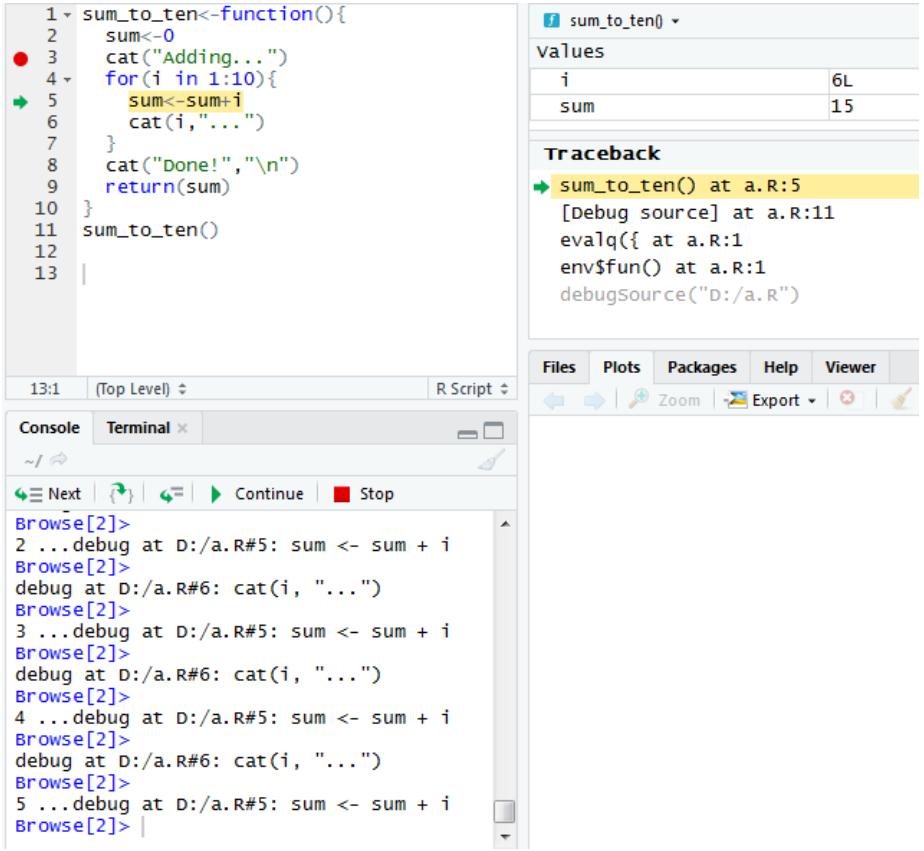
Debug

Rstudio에서 R코드를 Debug하는 예제는 하기와 같다.

코드

```
Source on Save
1  sum_to_ten<-function(){
2    sum<-0
3    cat("Adding...")
4    for(i in 1:10){
5      sum<-sum+i
6      cat(i,"...")
7    }
8    cat("Done!","\n")
9    return(sum)
10 }
11 sum_to_ten()
12
13
```

실행결과



Github 이용하기

Rstudio에서 개발한 코드를 Github에 올려놓고 이력관리를 할 수 있으며 여러 분석가와 공동작업을 할 수 있다.

❑ Github에 들어가서 계정을 생성한다.

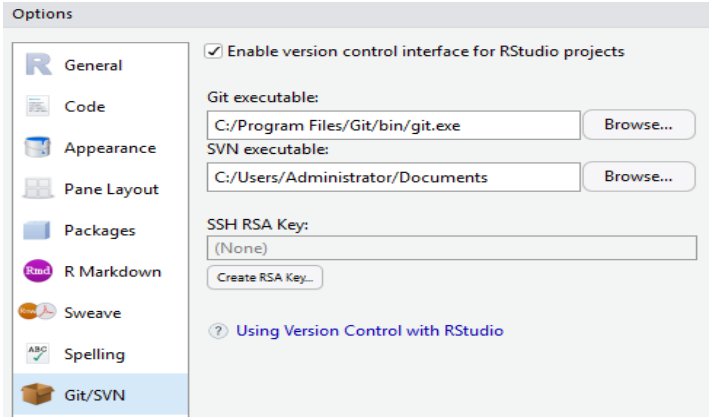
❑ OS에 적합한 Git을 설치한다.

❑ Git의 Bash버전을 실행하여 다음과 같이 입력한다.

```
sabum.jung@PTSMF10-NA101D5 MINGW64 /
$ git config --global user.name sabumjung

sabum.jung@PTSMF10-NA101D5 MINGW64 /
$ git config --global user.email skkie1991@gmail.com
```

❑ Rstudio의 메뉴에서 Tools-Global Options-Git/SVN으로 가서 Git을 설치한 폴더의 git실행파일을 찾아서 링크한다.



❑ Rstudio를 재시작한다.

❑ Github의 개인페이지로 가서 데이터를 업로드할 프로젝트를 만든다.

❑ Rstudio에서 File-New Project-Version Control-Git으로 가서 다음과 같이 만든다.

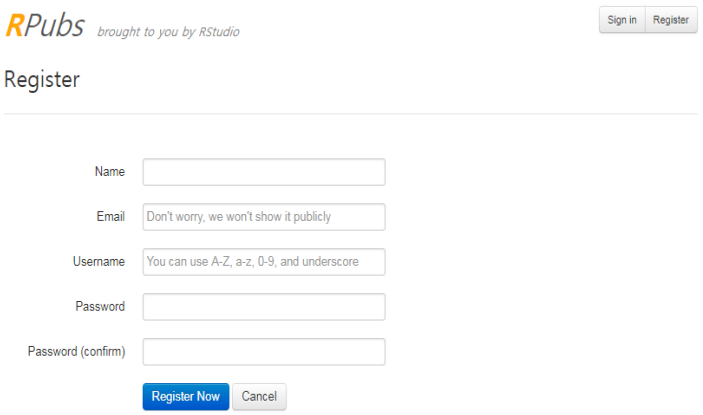
- URL은 이전 Github프로젝트 페이지의 http주소를 입력한다.

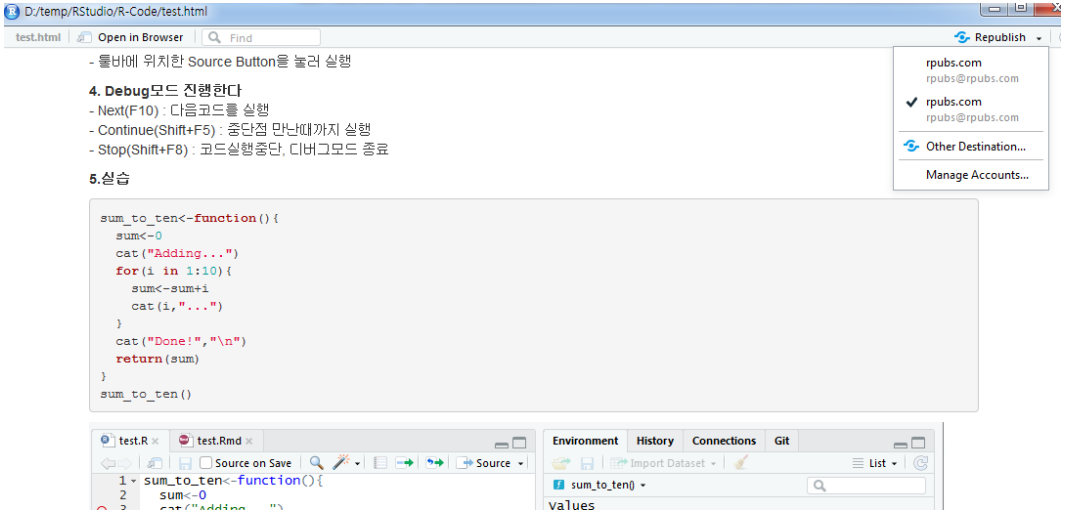
❑ 필요한 코드를 작성한 후 생성된 파일을 Github에 올린다.

Rpubs 이용하기

Rstudio에서 개발한 코드와 문서내용을 Rpubs에 publish하여 여러 사람과 공유할 수 있다.

- ❑ Rpubs사이트에 가서 회원가입한다.
- ❑ .Rmd파일을 작서한다. 작성법은 rmarkdown명령요약내용(cheat sheet)을 참고한다.
- ❑ R코드 편집기창의 knit아이콘을 클릭하여 문서(html)를 만든다.
- ❑ 생성된 문서를 확인하고 Republish버튼을 클릭하여 Rpubs에 Publish한다.

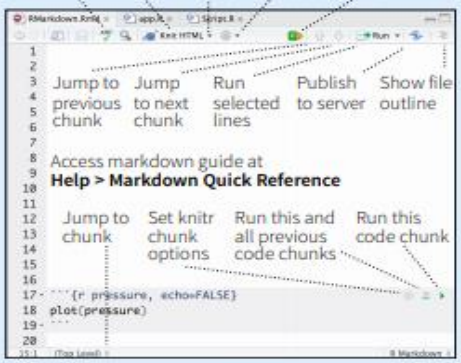




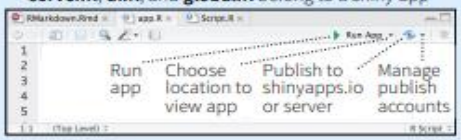
Rstudio Cheat sheet

Documents and Apps

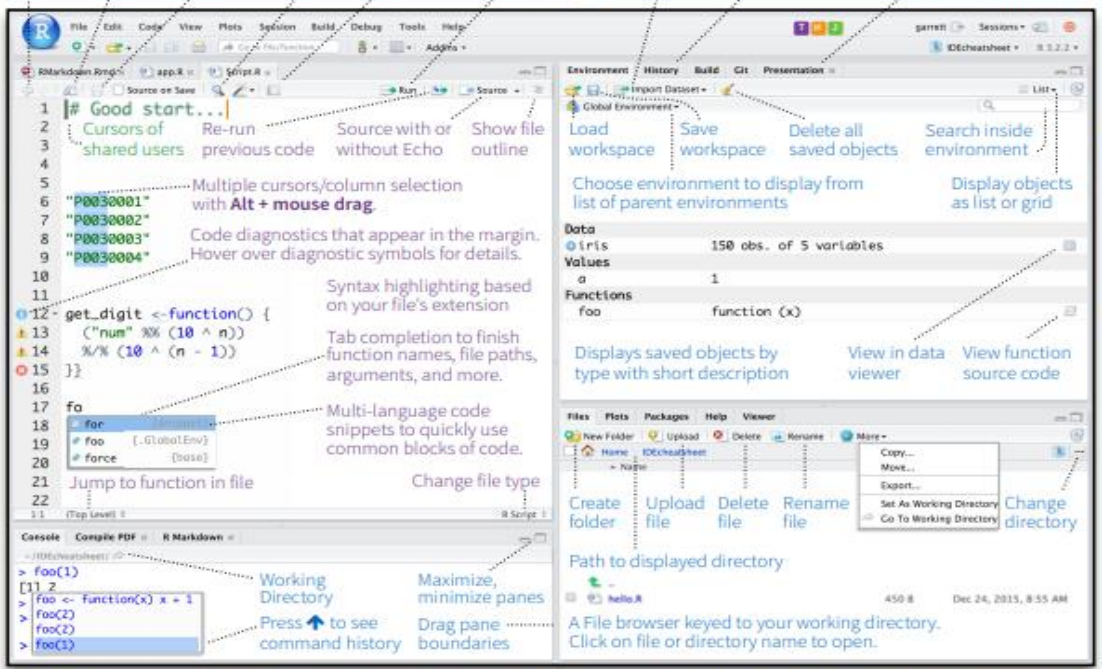
Open Shiny, R Markdown, knitr, Sweave, LaTeX, .Rd files and more in Source Pane



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

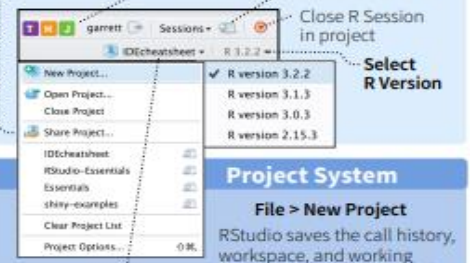


Write Code



R Support

RStudio Pro Features



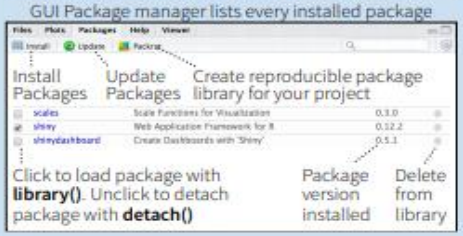
Project System

File > New Project

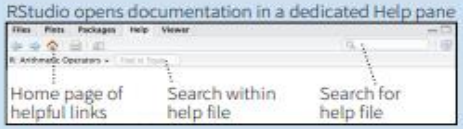
RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

RStudio opens plots in a dedicated Plots pane

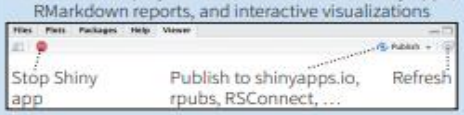
GUI Package manager lists every installed package



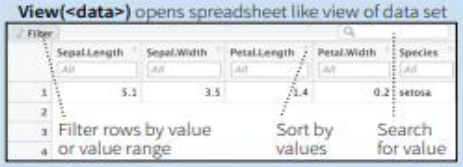
RStudio opens documentation in a dedicated Help pane



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



View(<data>) opens spreadsheet like view of data set

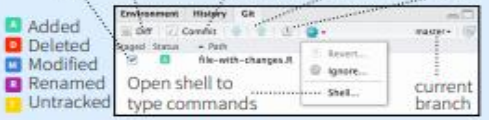


Debug Mode

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

Version Control with Git or SVN

Turn on at **Tools > Project Options > Git/SVN**



Package Writing

Rstudio Cheat sheet

1 LAYOUT	Windows/Linux	Mac
Move focus to Source Editor	Ctrl+1	Ctrl+1
Move focus to Console	Ctrl+2	Ctrl+2
Move focus to Help	Ctrl+3	Ctrl+3
Show History	Ctrl+4	Ctrl+4
Show Files	Ctrl+5	Ctrl+5
Show Plots	Ctrl+6	Ctrl+6
Show Packages	Ctrl+7	Ctrl+7
Show Environment	Ctrl+8	Ctrl+8
Show Git/SVN	Ctrl+9	Ctrl+9
Show Build	Ctrl+0	Ctrl+0

2 RUN CODE	Windows/Linux	Mac
Search command history	**Ctrl+↑**	**Cmd+↑**
Navigate command history	↑/↓	↑/↓
Move cursor to start of line	Home	Cmd+←
Move cursor to end of line	End	Cmd+→
Change working directory	Ctrl+Shift+H	Ctrl+Shift+H
Interrupt current command	**Esc**	**Esc**
Clear console	**Ctrl+L**	**Ctrl+L**
Quit Session (desktop only)	Ctrl+Q	Cmd+Q
Restart R Session	**Ctrl+Shift+F10**	**Cmd+Shift+F10**
Run current line/selection	**Ctrl+Enter**	**Cmd+Enter**
Run current (retain cursor)	Alt+Enter	Option+Enter
Run from current to end	Ctrl+Alt+E	Cmd+Option+E
Run the current function	Ctrl+Alt+F	Cmd+Option+F
Source a file	Ctrl+Shift+O	Cmd+Shift+O
Source the current file	**Ctrl+Shift+S**	**Cmd+Shift+S**
Source with echo	Ctrl+Shift+Enter	Cmd+Shift+Enter

3 NAVIGATE CODE	Windows /Linux	Mac
Goto File/Function	**Ctrl+.**	**Ctrl+.**
Fold Selected	Alt+L	Cmd+Option+L
Unfold Selected	Shift+Alt+L	Cmd+Shift+Option+L
Fold All	Alt+O	Cmd+Option+O
Unfold All	Shift+Alt+O	Cmd+Shift+Option+O
Go to line	Shift+Alt+G	Cmd+Shift+Option+G
Jump to	Shift+Alt+J	Cmd+Shift+Option+J
Switch to tab	Ctrl+Shift+.	Ctrl+Shift+.
Previous tab	Ctrl+F11	Ctrl+F11
Next tab	Ctrl+F12	Ctrl+F12
First tab	Ctrl+Shift+F11	Ctrl+Shift+F11
Last tab	Ctrl+Shift+F12	Ctrl+Shift+F12
Navigate back	Ctrl+F9	Cmd+F9
Navigate forward	Ctrl+F10	Cmd+F10
Jump to Brace	Ctrl+P	Ctrl+P
Select within Braces	Ctrl+Shift+Alt+E	Ctrl+Shift+Alt+E
Use Selection for Find	Ctrl+F3	Cmd+E
Find in Files	Ctrl+Shift+F	Cmd+Shift+F
Find Next	Win: F3, Linux: Ctrl+G	Cmd+G
Find Previous	W: Shift+F3, L: Ctrl+Shift	Cmd+Shift+G
Jump to Word	Ctrl+ ←/→	Option+ ←/→
Jump to Start/End	Ctrl+↑/↓	Cmd+↑/↓

4 WRITE CODE	Windows /Linux	Mac
Attempt completion	**Tab or Ctrl+Space**	**Tab or Cmd+Space**
Navigate candidates	↑/↓	↑/↓
Accept candidate	Enter, Tab, or →	Enter, Tab, or →
Dismiss candidates	Esc	Esc
Undo	Ctrl+Z	Cmd+Z
Redo	Ctrl+Shift+Z	Cmd+Shift+Z
Cut	Ctrl+X	Cmd+X
Copy	Ctrl+C	Cmd+C
Paste	Ctrl+V	Cmd+V
Select All	Ctrl+A	Cmd+A
Delete Line	Ctrl+D	Cmd+D
Select	Shift+[Arrow]	Shift+[Arrow]
Select Word	Ctrl+Shift+ ←/→	Option+Shift+ ←/→
Select to Line Start	Alt+Shift+←	Cmd+Shift+←
Select to Line End	Alt+Shift+→	Cmd+Shift+→
Select Page Up/Down	Shift+PageUp/Down	Shift+PageUp/Down
Select to Start/End	Shift+Alt+↑/↓	Cmd+Shift+↑/↓
Delete Word Left	Ctrl+Backspace	Ctrl+Opt+Backspace
Delete Word Right		Option+Delete
Delete to Line End		Ctrl+K
Delete to Line Start		Option+Backspace
Indent	Tab (at start of line)	Tab (at start of line)
Outdent	Shift+Tab	Shift+Tab
Yank line up to cursor	Ctrl+U	Ctrl+U
Yank line after cursor	Ctrl+K	Ctrl+K
Insert yanked text	Ctrl+Y	Ctrl+Y
Insert <-	**Alt+-**	**Option+-**
Insert %>%	**Ctrl+Shift+M**	**Cmd+Shift+M**
Show help for function	F1	F1
Show source code	F2	F2
New document	Ctrl+Shift+N	Cmd+Shift+N
New document (Chrome)	Ctrl+Alt+Shift+N	Cmd+Shift+Alt+N
Open document	Ctrl+O	Cmd+O
Save document	Ctrl+S	Cmd+S
Close document	Ctrl+W	Cmd+W
Close document (Chrome)	Ctrl+Alt+W	Cmd+Option+W
Close all documents	Ctrl+Shift+W	Cmd+Shift+W
Extract function	Ctrl+Alt+X	Cmd+Option+X
Extract variable	Ctrl+Alt+V	Cmd+Option+V
Reindent lines	Ctrl+I	Cmd+I
(Un)Comment lines	**Ctrl+Shift+C**	**Cmd+Shift+C**
Reflow Comment	Ctrl+Shift+/	Cmd+Shift+/
Reformat Selection	Ctrl+Shift+A	Cmd+Shift+A
Select within braces	Ctrl+Shift+E	Ctrl+Shift+E
Show Diagnostics	Ctrl+Shift+Alt+P	Cmd+Shift+Alt+P
Transpose Letters		Ctrl+T
Move Lines Up/Down	Alt+↑/↓	Option+↑/↓
Copy Lines Up/Down	Shift+Alt+↑/↓	Cmd+Option+↑/↓
Add New Cursor Above	Ctrl+Alt+Up	Ctrl+Alt+Up
Add New Cursor Below	Ctrl+Alt+Down	Ctrl+Alt+Down
Move Active Cursor Up	Ctrl+Alt+Shift+Up	Ctrl+Alt+Shift+Up
Move Active Cursor Down	Ctrl+Alt+Shift+Down	Ctrl+Alt+Shift+Down
Find and Replace	Ctrl+F	Cmd+F
Use Selection for Find	Ctrl+F3	Cmd+E
Replace and Find	Ctrl+Shift+J	Cmd+Shift+J

5 DEBUG CODE	Windows/Linux	Mac
Toggle Breakpoint	Shift+F9	Shift+F9
Execute Next Line	F10	F10
Step Into Function	Shift+F4	Shift+F4
Finish Function/Loop	Shift+F6	Shift+F6
Continue	Shift+F5	Shift+F5
Stop Debugging	Shift+F8	Shift+F8

6 VERSION CONTROL	Windows/Linux	Mac
Show diff	Ctrl+Alt+D	Ctrl+Option+D
Commit changes	Ctrl+Alt+M	Ctrl+Option+M
Scroll diff view	Ctrl+↑/↓	Ctrl+↑/↓
Stage/Unstage (Git)	Spacebar	Spacebar
Stage/Unstage and move to next	Enter	Enter

7 MAKE PACKAGES	Windows/Linux	Mac
Build and Reload	Ctrl+Shift+B	Cmd+Shift+B
Load All (devtools)	**Ctrl+Shift+L**	**Cmd+Shift+L**
Test Package (Desktop)	**Ctrl+Shift+T**	**Cmd+Shift+T**
Test Package (Web)	Ctrl+Alt+F7	Cmd+Alt+F7
Check Package	Ctrl+Shift+E	Cmd+Shift+E
Document Package	**Ctrl+Shift+D**	**Cmd+Shift+D**

8 DOCUMENTS AND APPS	Windows/Linux	Mac
Preview HTML (Markdown, etc.)	Ctrl+Shift+K	Cmd+Shift+K
Knit Document (knitr)	**Ctrl+Shift+K**	**Cmd+Shift+K**
Compile Notebook	Ctrl+Shift+K	Cmd+Shift+K
Compile PDF (TeX and Sweave)	Ctrl+Shift+K	Cmd+Shift+K
Insert chunk (Sweave and Knitr)	Ctrl+Alt+I	Cmd+Option+I
Insert code section	Ctrl+Shift+R	Cmd+Shift+R
Re-run previous region	Ctrl+Shift+P	Cmd+Shift+P
Run current document	Ctrl+Alt+R	Cmd+Option+R
Run from start to current line	**Ctrl+Alt+B**	**Cmd+Option+B**
Run the current code section	**Ctrl+Alt+T**	**Cmd+Option+T**
Run previous Sweave/Rmd code	Ctrl+Alt+P	Cmd+Option+P
Run the current chunk	Ctrl+Alt+C	Cmd+Option+C
Run the next chunk	Ctrl+Alt+N	Cmd+Option+N
Sync Editor & PDF Preview	Ctrl+F8	Cmd+F8

Previous plot	Ctrl+Alt+F11	Cmd+Option+F11
Next plot	Ctrl+Alt+F12	Cmd+Option+F12

Show Keyboard Shortcuts	Alt+Shift+K	Option+Shift+K

Rmarkdown Cheat sheet

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

rmarkdown 0.2.50 Updated: 8/14

RStudio

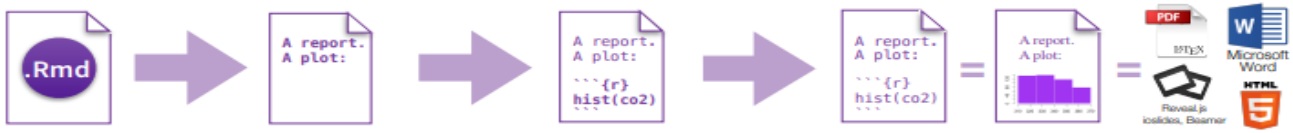
1. Workflow R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

i. Open - Open a file that uses the .Rmd extension.

ii. Write - Write content with the easy to use R Markdown syntax

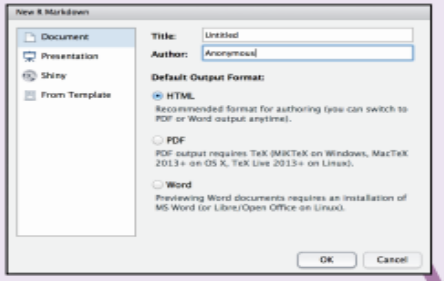
iii. Embed - Embed R code that creates output to include in the report

iv. Render - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.



2. Open File Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ▶ New File ▶ R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header  | Second Header
------------- | -------------
Table Cell    | Cell 2
Cell 3        | Cell 4
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image: 
horizontal rule (or slide break):

> block quote

- unordered list
- item 2
    - sub-item 1
    - sub-item 2

1. ordered list
2. item 2
    - sub-item 1
    - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

4. Choose Output Write a YAML header that explains what type of document to build from your R Markdown file.

YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (- - -)

```
---
title: "Untitled"
author: "Anonymous"
output: html_document
---

This is the start of my
report. The above is metadata
saved in a YAML header.
```

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

output: html_document	html file (web page)
output: pdf_document	pdf document
output: word_document	Microsoft Word .docx
output: beamer_presentation	beamer slideshow (pdf)
output: ioslides_presentation	ioslides slideshow (html)

Rmarkdown Cheat sheet

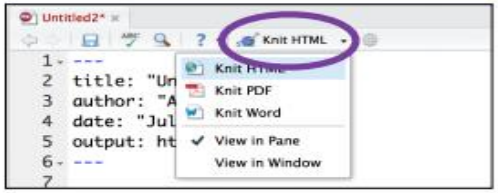
5. Embed Code
Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.

inline code
Surround code with back ticks and r.
R replaces inline code with its results.

```
Two plus two equals `r 2 + 2`.
```

Two plus two equals 4.

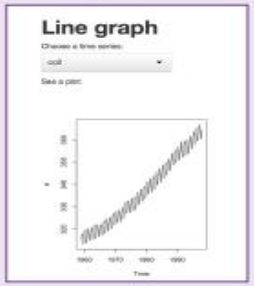
code chunks
Start a chunk with ```{r}.
End a chunk with ```

```
Here's some code
```{r}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

```
## [1] 150   5
```

display options
Use knitr options to style the output of a chunk.
Place options in brackets above the chunk.

```
Here's some code
```{r eval=FALSE}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

```
Here's some code
```{r echo=FALSE}
dim(iris)
```
```

Here's some code

```
## [1] 150   5
```

option	default	effect
eval	TRUE	Whether to evaluate the code and include its results
echo	TRUE	Whether to display code along with its results
warning	TRUE	Whether to display warnings
error	FALSE	Whether to display errors
message	TRUE	Whether to display messages
tidy	FALSE	Whether to reformat code in a tidy way when displaying it
results	"markup"	"markup", "asis", "hold", or "hide"
cache	FALSE	Whether to cache results for future renders
comment	"##"	Comment character to preface results with
fig.width	7	Width in inches for plots created in chunk
fig.height	7	Height in inches for plots created in chunk

For more details visit yihui.name/knitr/

6. Render
Use your .Rmd file as a blueprint to build a finished report.

Render your report in one of two ways

1. Run **rmarkdown::render("<file path>")**
2. Click the **knit HTML** button at the top of the RStudio scripts pane



When you render, R will

- execute each embedded code chunk and insert the results into your report
- build a new version of your report in the output file type
- open a preview of the output file in the viewer pane
- save the output file in your working directory

7. Interactive Docs
Turn your report into an interactive Shiny document in 3 steps

1 Add **runtime: shiny** to the YAML header

```
---
title: "Line graph"
output: html_document
runtime: shiny
---
```

2 In the code chunks, add Shiny **input** functions to embed widgets. Add Shiny **render** functions to embed reactive output

```
---
title: "Line graph"
output: html_document
runtime: shiny
---

Choose a time series:
```{r echo = FALSE}
selectInput("data", "",
  c("co2", "lh"))
```

See a plot:
```{r echo = FALSE}
renderPlot({
  d <- get(input$data)
  plot(d)
})
```
```

3 Render with **rmarkdown::run** or click **Run Document** in RStudio



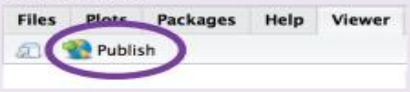
** Note: your report will be a Shiny app, which means you must choose an html output format, like **html_document** (for an interactive report) or **ioslides_presentation** (for an interactive slideshow).*

8. Publish
Share your report where users can visit it online

Rpubs.com
Share non-interactive documents on RStudio's free R Markdown publishing site
www.rpubs.com

ShinyApps.io
Host an interactive document on RStudio's server. Free and paid options
www.shinyapps.io

Click the "Publish" button in the RStudio preview window to publish to rpubs.com with one click.



9. Learn More

Documentation and examples - rmarkdown.rstudio.com
Further Articles - shiny.rstudio.com/articles
- blog.rstudio.com
- @rstudio