

R Tutorials

Akihito Kamata, Ph.D.
Southern Methodist University
akamata@smu.edu

March 7, 2020
University of Southern Mississippi

Materials

- All materials including this slide set are available at:
 - <https://github.com/kamataak/WorkshopUSM2020>
- Download all materials.
 - Click on “Clone or download”.
 - If you know how to use Git, clone it!
 - Otherwise, click on “Download ZIP”.
 - The entire folder will be downloaded to your download folder as a compressed ZIP file.
 - Move the ZIP file where you desire.
 - Unzip the folder.

Part I

Workflow on RStudio

Installing R

- Download R from R-Project website.
 - www.r-project.org
 - Follow links to go to the download page.
 - Go to “CRAN”
 - Then, choose one of the links, such as one under “0-Cloud”
 - R is available for Windows, Mac OS, and Linux platforms.

Download and Install R

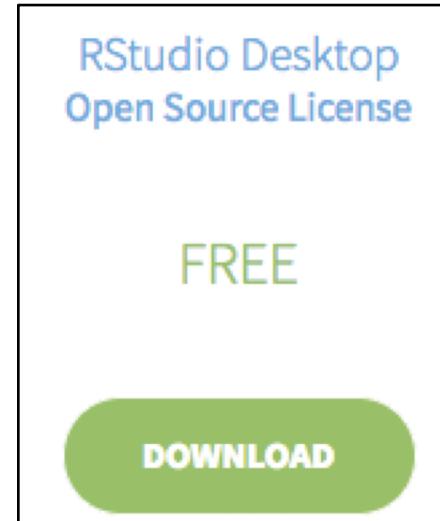
Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

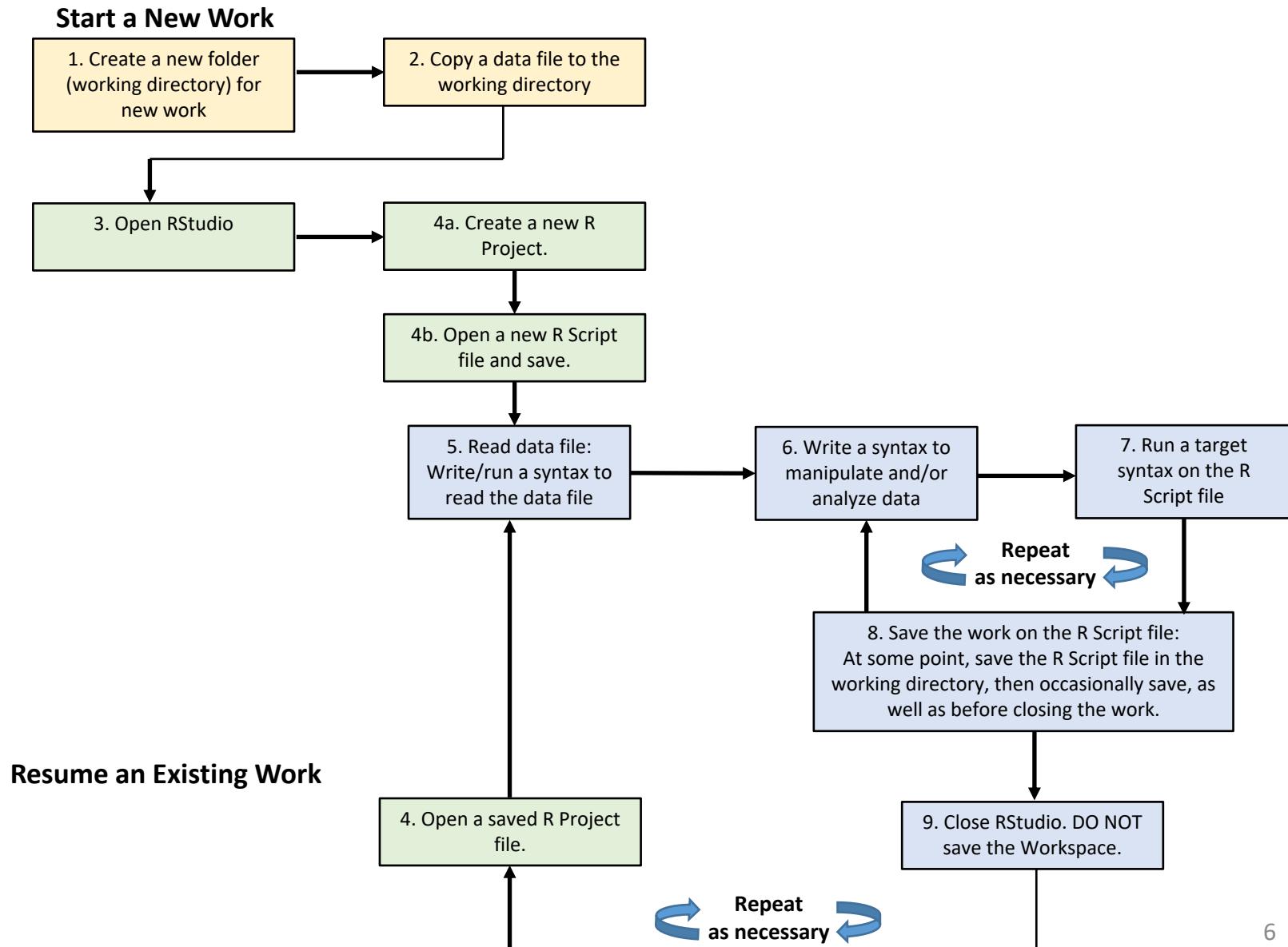
R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Installing R Studio

- This will make R much more user friendly.
- Download from R Studio website.
 - www.rstudio.com
 - Follow links to go to the download page.
 - Download the free version.



Recommended Work Management on RStudio

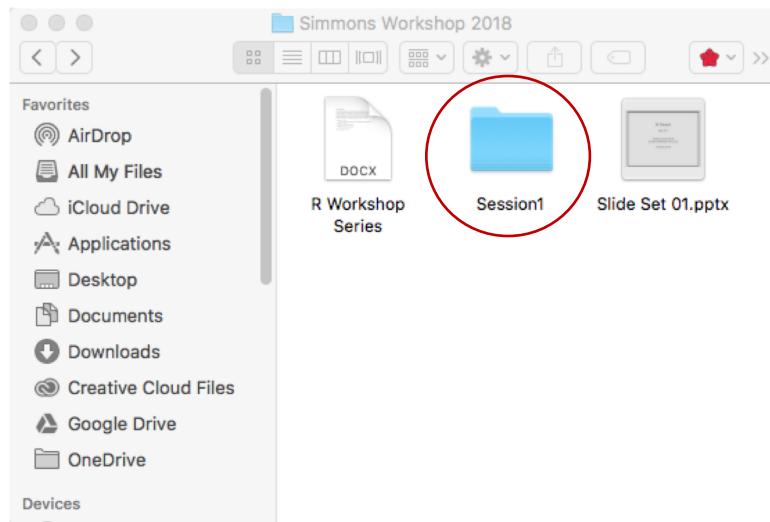


Start a New Work:

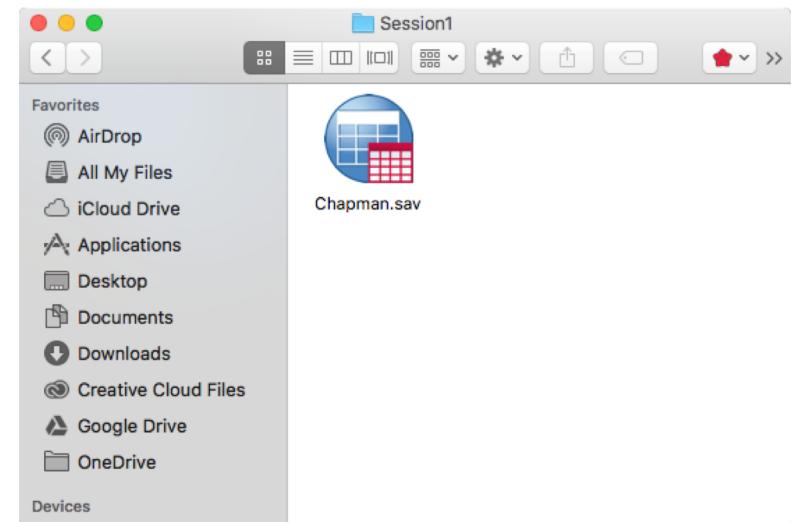
Steps 1 & 2: Preparation of Working Directory

- Create a new folder. This is a working directory.
- Copy the data file to analyze to the working directory.
 - For this demonstration, download an SPSS data from:
http://akikamata.s3.amazonaws.com/Web_Files/Chapman.sav

Create a new folder (working directory).



Download the data file to the working directory.



Step 3: Open R Studio

- Open RStudio, just like opening other programs on your computer.
- It will automatically link to R.

R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

[Workspace loaded from ~/.RData]

> |

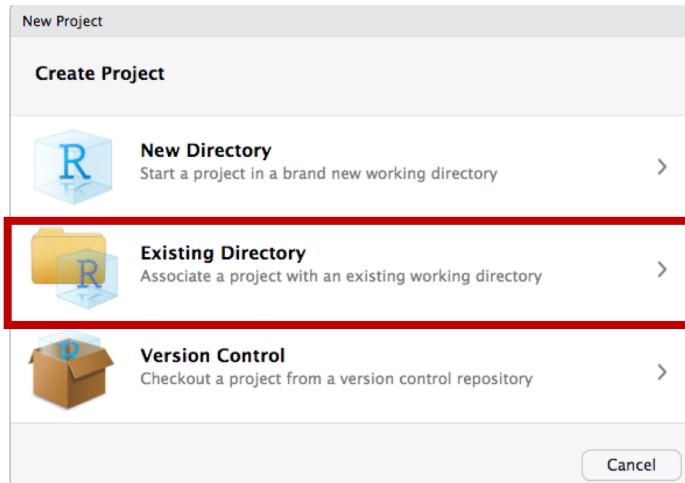
Environment History
Import Dataset List
Global Environment

Environment is empty

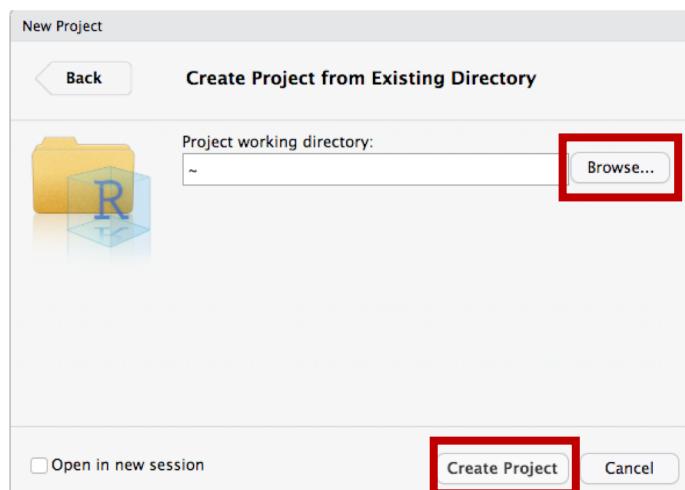
Files Plots Connections Packages Help Viewer
Zoom Export

Step 4a: Create a New R Project

- Click on “Create a project” button .



- Click on “Existing Directory”.



- Click on “Browse...”, and select the target working directory folder, where the data file is.
- Then, click on “Create Project”.

R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

> |

Session 1

Environment History

Import Dataset List

Global Environment

Environment is empty

Files Plots Connections Packages Help Viewer

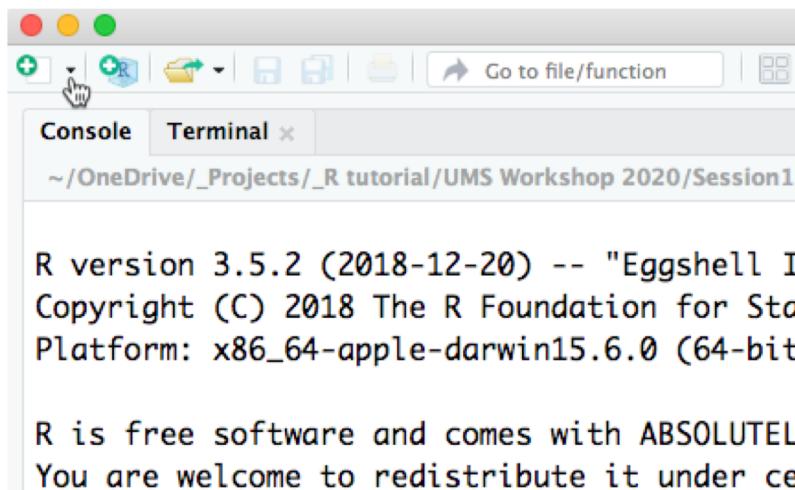
New Folder Delete Rename More

.. Chapman.sav 5.8 KB Oct 23, 2016

Session1.Rproj 205 B Jan 17, 2020,

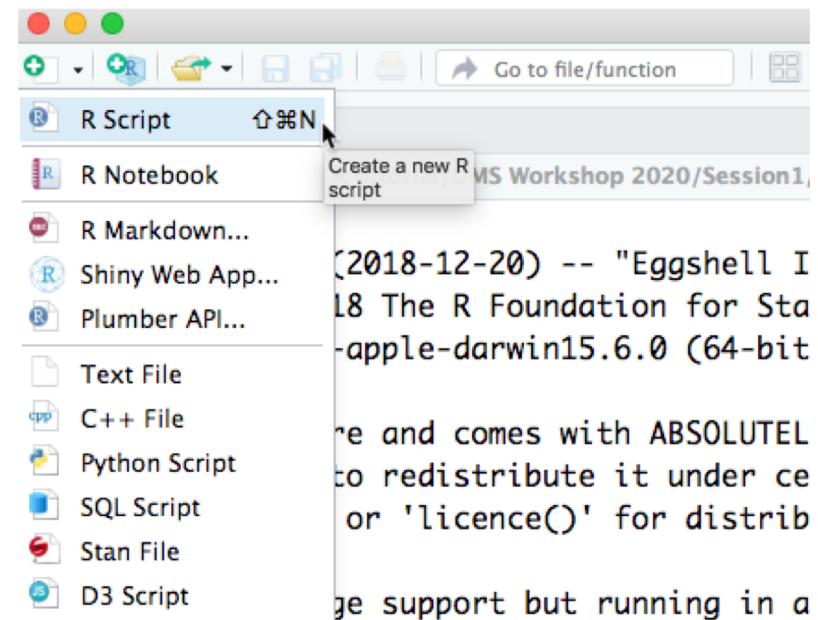
Steps 4b: Open a New R Script File and Save

- Open a new script file.
 - Choose an “R Script” option from the pull-down menu under  button.



R version 3.5.2 (2018-12-20) -- "Eggshell I
Copyright (C) 2018 The R Foundation for Sta
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTEL
You are welcome to redistribute it under ce



- This is R script file window.
- You will type R code here later.

Untitled1

Source on Save | Run | Source | Environment

1

1:1 (Top Level) R Script

Console Terminal

~/OneDrive/_Projects/_R tutorial/UMS Workshop 2020/Session1/

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.

> |

Environment History Import Dataset Global Environment

Environment is empty

Files Plots Connections Packages Help Viewer

New Folder Delete Rename More

drive > _Projects > _R tutorial > UMS Workshop 2020 > Session1

Name	Size	Modified
Chapman.sav	5.8 KB	Oct 23, 2016
Session1.Rproj	205 B	Jan 17, 2020,

Steps 4b: Open a New R Script File and Save

- Save the script file.
 - Click  icon.
 - The file will be saved in the target working directory by default.

Script01.R x

Source on Save | Run | Source | Environment

1 |

1:1 (Top Level) R Script

Console Terminal

~/OneDrive/_Projects/_R tutorial/UMS Workshop 2020/Session1/

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.

> |

Environment History

Import Dataset | Global Environment

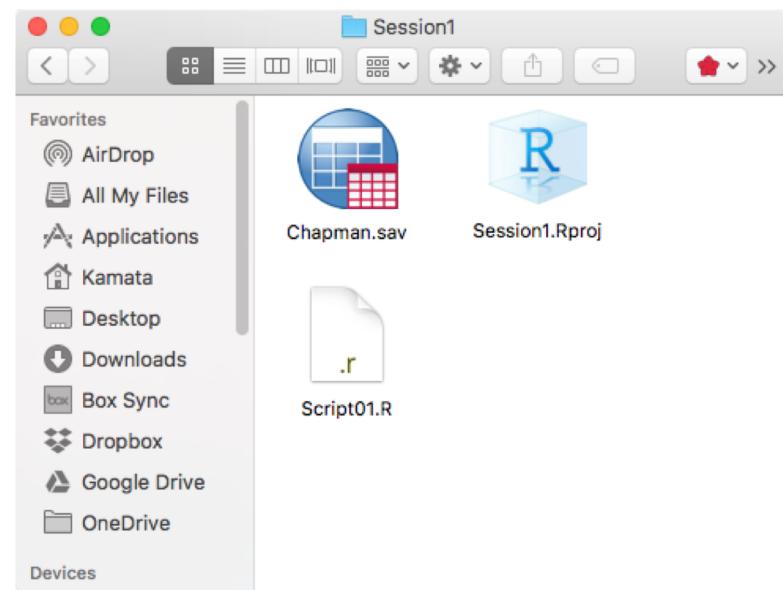
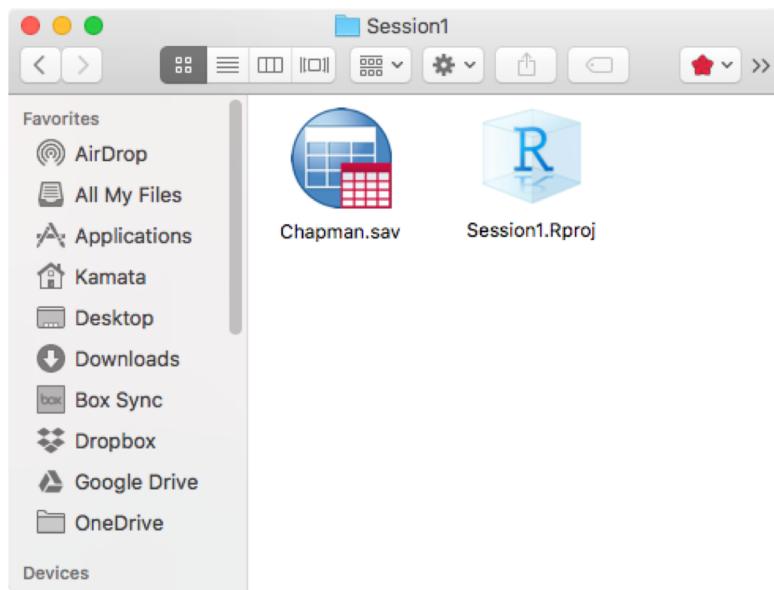
Environment is empty

Files Plots Connections Packages Help Viewer

New Folder Delete Rename More

drive > _Projects > _R tutorial > UMS Workshop 2020 > Session1

Name	Size	Modified
Chapman.sav	5.8 KB	Oct 23, 2016
Script01.R	0 B	Jan 17, 2020
Session1.Rproj	205 B	Jan 17, 2020



Step 5: Read Data into R (1)

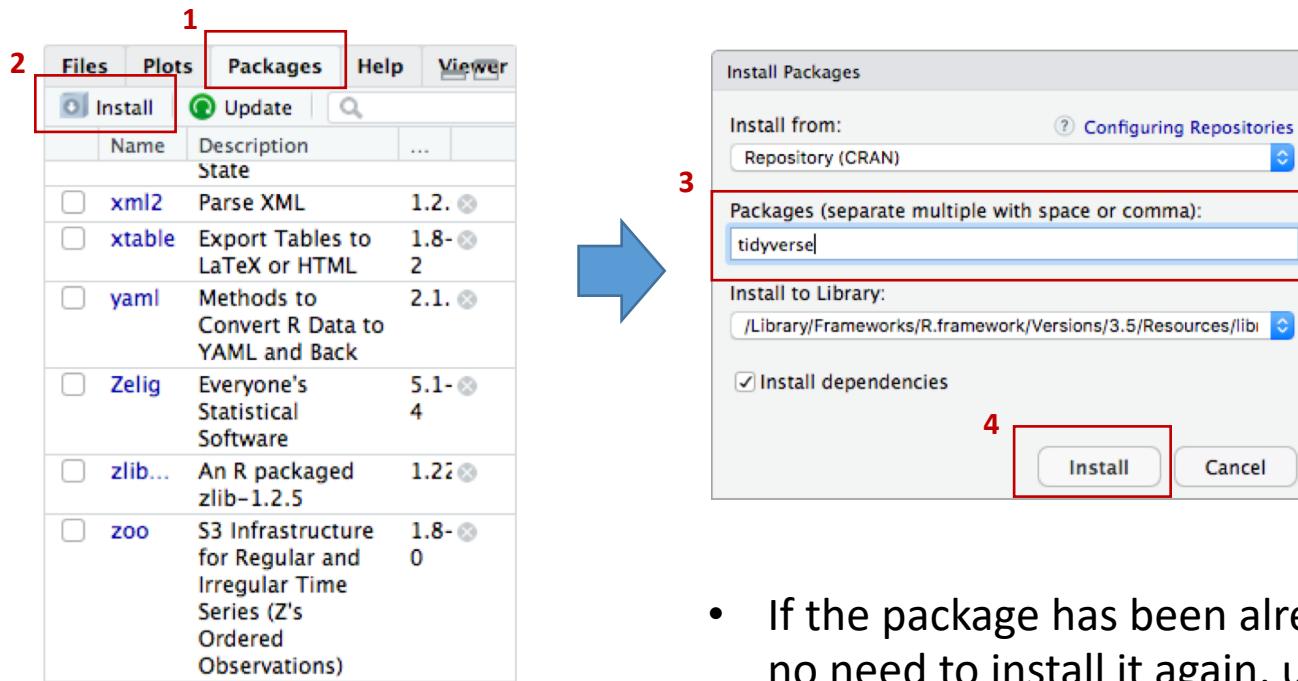
- R can read many different data file formats, including
 - SPSS, CSV, SAS, STATA, Excel, etc.
- To read SPSS data, use `read_spss` function in `haven` package.
 - Other proprietary data formats, including SAS and STATA data can be read by other functions in `haven` package.
- To read CSV data, use `read_csv` function in `readr` package.
 - Also, `read.csv` function in the base package can be used.
- To reading Excel file, use `use read_excel` function in `readxl` package, for example.

Aside: Packages for R

- R comes with only limited set of base packages.
- To do something realistic, we almost always need to use other packages.
 - For example, `haven` package is not part of the base packages, so we need to install and load it.
 - Let's install a package called `tidyverse`, which contains `haven` package and other useful packages.

Aside: Install a Package

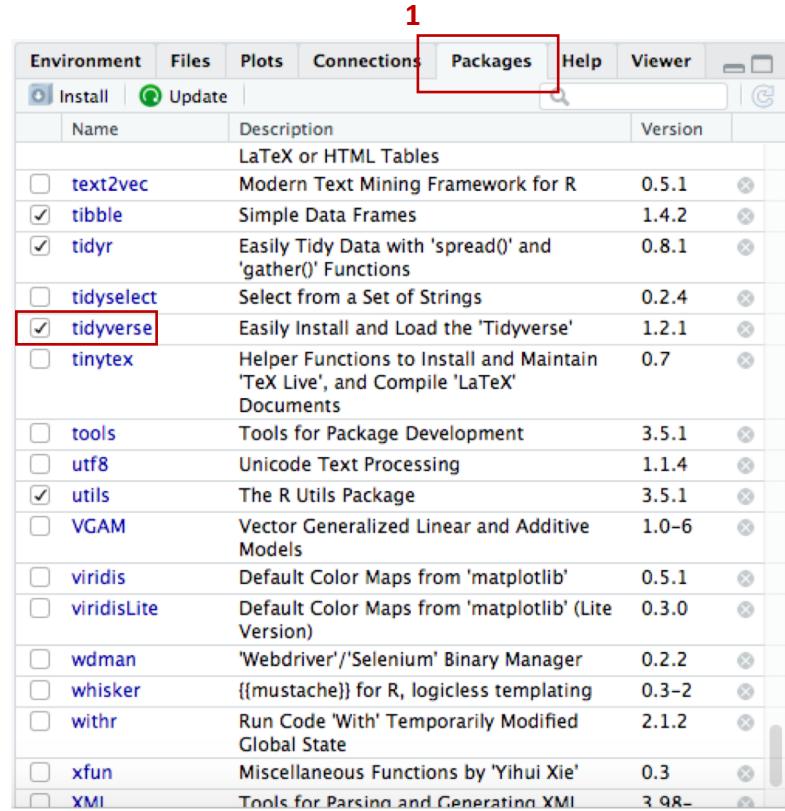
- Let's install the `tidyverse` package:
 - Run a command `install.packages("tidyverse")`
 - Alternatively, install it by point-and-click operations.
(Recommended)



- If the package has been already installed, no need to install it again, unless it needs to be updated to a newer version.

Aside: Load a Package

- For example, to load `tidyverse` package:
 - Run a command `library(tidyverse)` (Recommended)
 - Alternatively, we can load it by point-and-click operations.



- A package must be loaded every time you start a session, regardless starting new work or resuming existing work.

Step 5: Read SPSS Data File (2)

- Use `read_spss` function in `haven` package.



```
library(haven)
Chapman.dat <- read_spss("Chapman.sav")
```

- Running line 2 will create a data object on R, called `Chapman.dat`

Aside: How to Run R Syntax Lines



```
library(haven)
Chapman.dat <- read_spss("Chapman.sav")
```

- To run a syntax line by line,
 1. Place the cursor on the target line.
 2. Click the  button.
 - Alternatively, use keyboard shortcut **Ctrl+Enter** (Win) or **Command+Enter** (Mac).
- To run multiple lines sequentially,
 1. Highlight the target lines.
 2. Click the  button, or type **Ctrl+Enter** (Win) or **Command+Enter** (Mac).

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Shows the R script `Script01.R` containing the code:

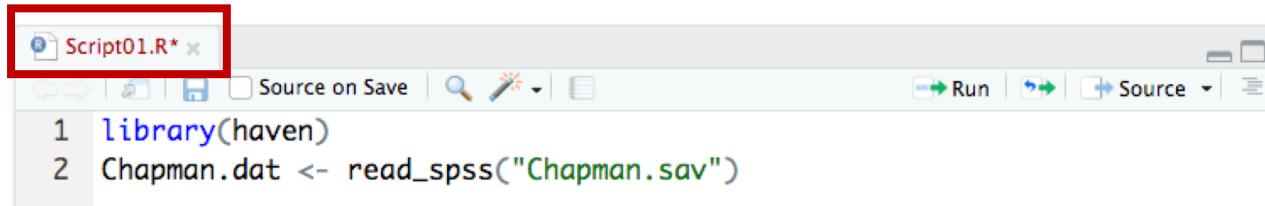
```
1 library(haven)
2 Chapman.dat <- read_spss("Chapman.sav")
3
```
- Environment View:** Shows the newly created object `Chapman.dat` in the Global Environment.
- Files View:** Shows the project structure and files:

Name	Size	Modified
..		
Chapman.sav	5.8 KB	Oct 23, 2016, 3:38 PM
Session1.Rproj	205 B	Jan 17, 2020, 11:11 AM
Script01.R	0 B	Jan 17, 2020, 11:42 AM
- Console View:** Shows the output of the R session, including the R language support message and the execution of the script code.

Annotations:

- A red callout box points from the text "Newly created object 'Chapman.dat' is shown in here" to the `Chapman.dat` entry in the Environment view.
- A red callout box points from the text "When a command line is executed, the line will show up in the console." to the first two lines of the Console output (`> library(haven)` and `> Chapman.dat <- read_spss("Chapman.sav")`).

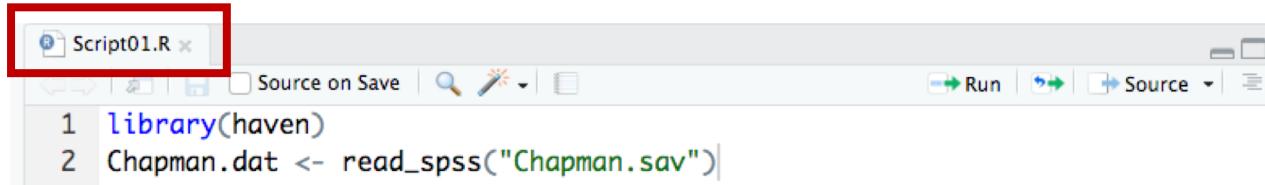
Aside: Saving R Script File



The screenshot shows the RStudio interface with a single script editor window. The title bar of the window is highlighted with a red box and displays the text "Script01.R*" with a small document icon. The main area of the window contains two lines of R code:

```
1 library(haven)
2 Chapman.dat <- read_spss("Chapman.sav")
```

- When the script file name is red with an asterisk, it means that the changes on the script file is unsaved.
- It is highly recommended to save changes periodically.
 - Click the  button to save.
 - Alternatively, use keyboard shortcut **Ctrl+S** (Win) or **⌘+S** (Mac).

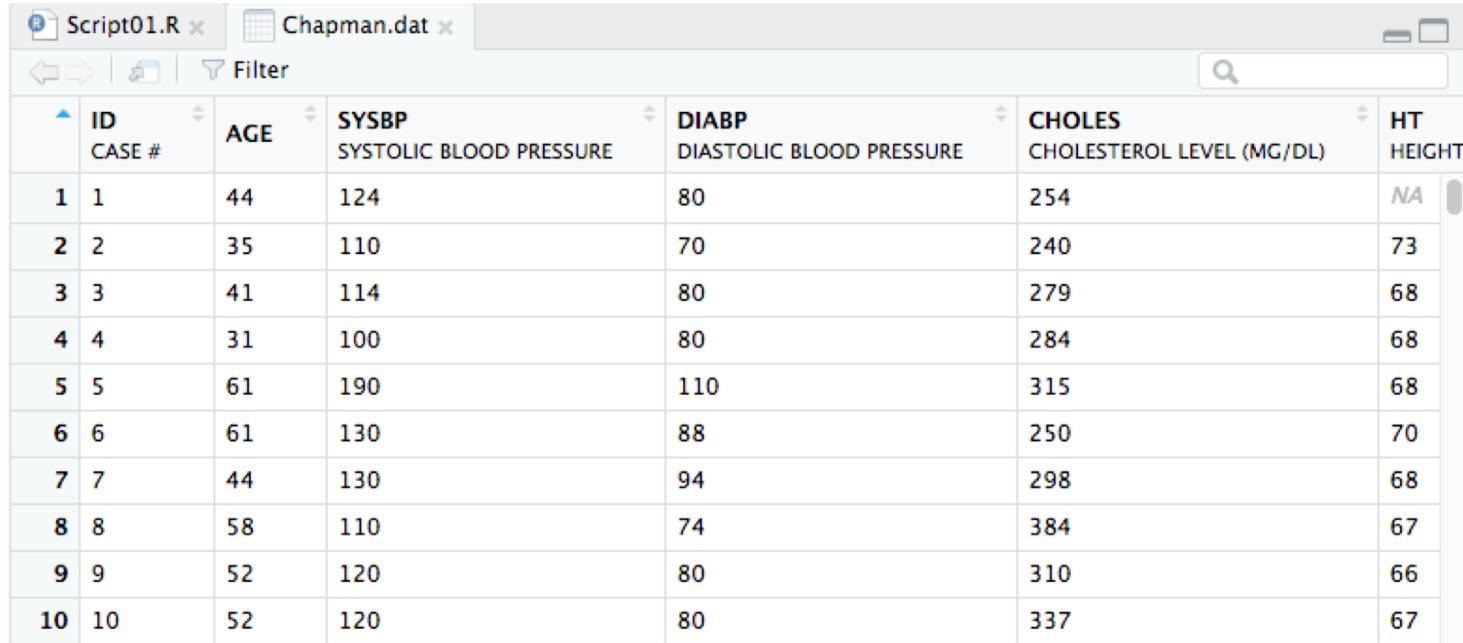


The screenshot shows the RStudio interface with a single script editor window. The title bar of the window is highlighted with a red box and displays the text "Script01.R" with a small document icon. The main area of the window contains the same two lines of R code as the previous screenshot:

```
1 library(haven)
2 Chapman.dat <- read_spss("Chapman.sav")
```

Step 5: Read SPSS Data File (3)

- To view the imported data in a spread sheet format, click right on the data object name, or  icon for the target data object.



The screenshot shows the RStudio interface with the Chapman.dat data frame loaded. The data frame has 10 rows and 6 columns. The columns are labeled: ID CASE #, AGE, SYSBP SYSTOLIC BLOOD PRESSURE, DIABP DIASTOLIC BLOOD PRESSURE, CHOLES CHOLESTEROL LEVEL (MG/DL), and HT HEIGHT. The data is as follows:

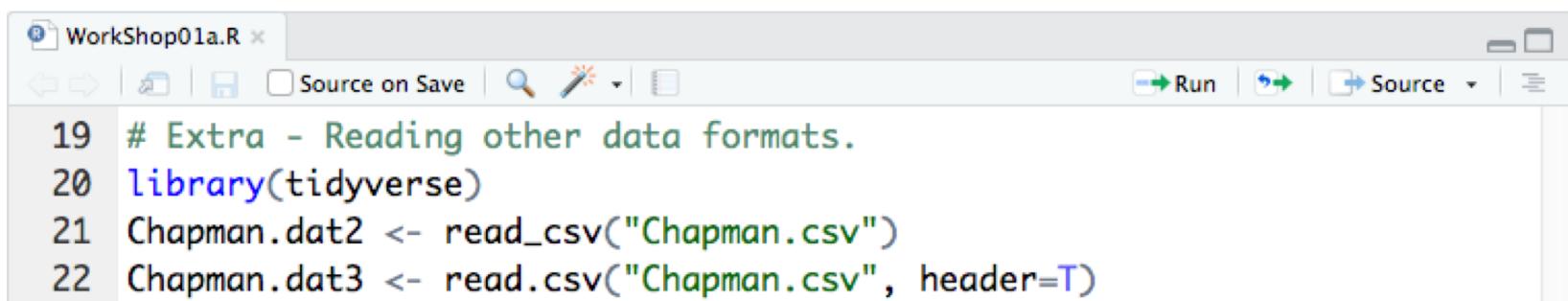
	ID CASE #	AGE	SYSBP SYSTOLIC BLOOD PRESSURE	DIABP DIASTOLIC BLOOD PRESSURE	CHOLES CHOLESTEROL LEVEL (MG/DL)	HT HEIGHT
1	1	44	124	80	254	NA
2	2	35	110	70	240	73
3	3	41	114	80	279	68
4	4	31	100	80	284	68
5	5	61	190	110	315	68
6	6	61	130	88	250	70
7	7	44	130	94	298	68
8	8	58	110	74	384	67
9	9	52	120	80	310	66
10	10	52	120	80	337	67

Aside: Variable Names, etc.

- Variable name has to start with a Roman letter.
 - No numeric, like 4 and 0.
 - No special character, like % and \$.
- R is case sensitive.
 - For example, a variable name `StuID` is not the same as a variable name `stuid`.
 - Similarly, object names, function names, and package names are all case sensitive.
 - For example, the function `read_spss` will not run by typing `read_SPSS`.

Aside: Read CSV data

- CSV data format is one of most convenient ways to manage data on R.
 - The first row of a CSV data file can be a list of variable names.
- Use `read_csv` function in the `readr` package, which will be loaded if the `tidyverse` package is loaded, or `read.csv` function in the base package just like this:

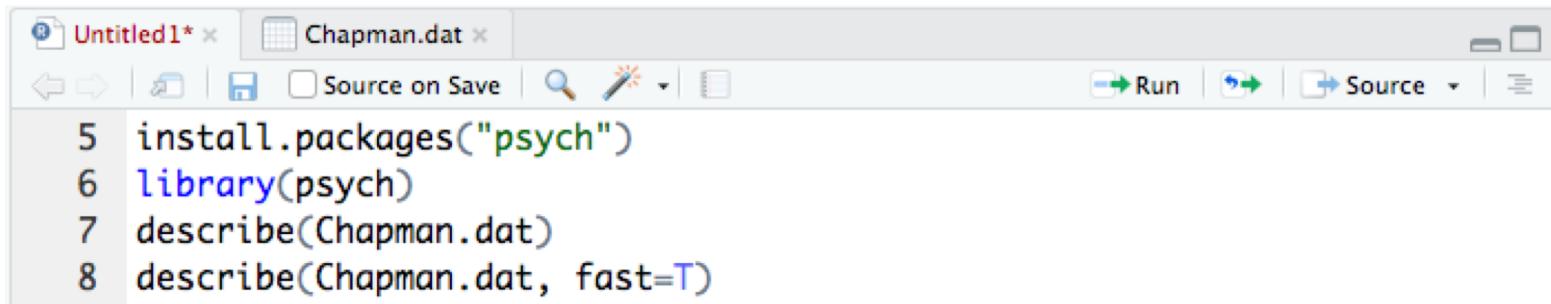
A screenshot of the RStudio interface showing an R script named "WorkShop01a.R". The code in the script is:

```
19 # Extra - Reading other data formats.
20 library(tidyverse)
21 Chapman.dat2 <- read_csv("Chapman.csv")
22 Chapman.dat3 <- read.csv("Chapman.csv", header=T)
```

The RStudio toolbar at the top includes icons for back, forward, file, search, and run. The status bar at the bottom right shows the page number 27.

Steps 6 & 7: Let's Do Something with the Data (1)

- Descriptive statistics for all variables:
 - Use `describe` function in `psych` package.
- Type the following lines and run each line.



The screenshot shows the RStudio interface with the script editor open. The code pane contains the following R code:

```
5 install.packages("psych")
6 library(psych)
7 describe(Chapman.dat)
8 describe(Chapman.dat, fast=T)
```

- Again, if the package has been already installed, no need to install it again.
- However, a package must be loaded every time you start a session, regardless starting new work or resuming existing work.

Aside: How to find a function & package

- For example, if I am interested in how to do a regression analysis, I would search by “regression in r” on Google.

The screenshot shows a Google search results page. The search bar at the top contains the query "regression in r". Below the search bar are navigation links for "All", "Videos", "Images", "Shopping", "News", and "More", along with "Settings" and "Tools" buttons. A message indicates "About 56,500,000 results (0.49 seconds)". The first result is a link to "Quick-R: Multiple Regression" from <https://www.statmethods.net/stats/regression.html>. The snippet describes multiple linear regression in R and mentions K-Fold cross-validation. The second result is a link to "Linear Regression With R - r-statistics.co" from r-statistics.co/Linear-Regression.html. The snippet discusses using the cars dataset in R for regression analysis. The third result is a link to "Simple Linear Regression | R-bloggers" from <https://www.r-bloggers.com/simple-linear-regression-2/>. The snippet describes simple linear regression with one response variable and one independent variable, mentioning the use of the data argument.

regression in r

All Videos Images Shopping News More Settings Tools

About 56,500,000 results (0.49 seconds)

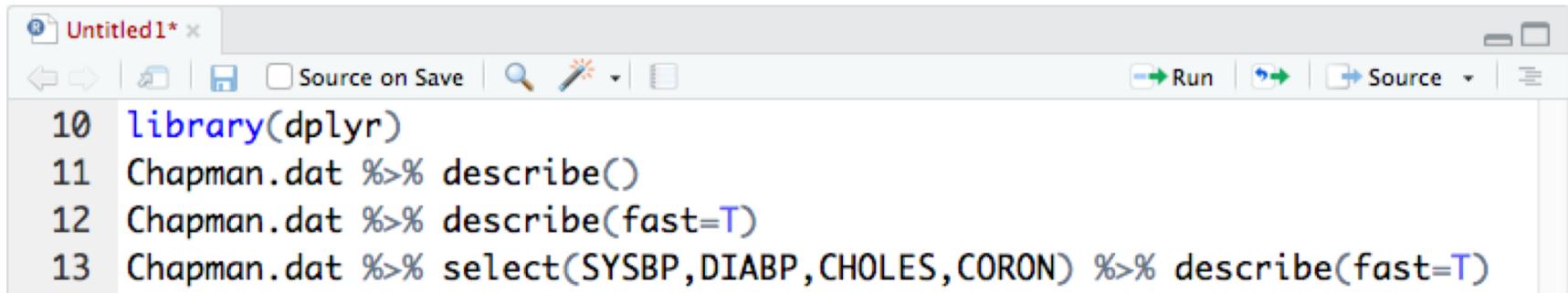
Quick-R: Multiple Regression
<https://www.statmethods.net/stats/regression.html> ▾
Multiple (Linear) Regression. R provides comprehensive support for multiple linear regression. ... You can do K-Fold cross-validation using the cv.lm() function in the DAAG package.

Linear Regression With R - r-statistics.co
r-statistics.co/Linear-Regression.html ▾
For this analysis, we will use the cars dataset that comes with R by default. cars is a standard built-in dataset, that makes it convenient to demonstrate linear regression in a simple and easy to understand fashion. You can access this dataset simply by typing in cars in your R console. You will find that it consists of 50 ...
[Logistic Regression](#) · [Advanced Regression Models](#) · [Statistical Tests](#) · [Assumption](#)

Simple Linear Regression | R-bloggers
<https://www.r-bloggers.com/simple-linear-regression-2/> ▾
Apr 23, 2010 - In this post we will consider the case of simple linear regression with one response variable and a single independent variable. ... The data argument is used to tell R where to look for the variables used in the formula.

Steps 6 & 7: Let's Do Something with the Data (2)

- Basic data manipulation - select only a subset of variables:
 - Use `select` function in `dplyr` package.
- Type the following lines and run each line.



The screenshot shows an RStudio interface with a script editor window titled "Untitled1*". The code in the editor is:

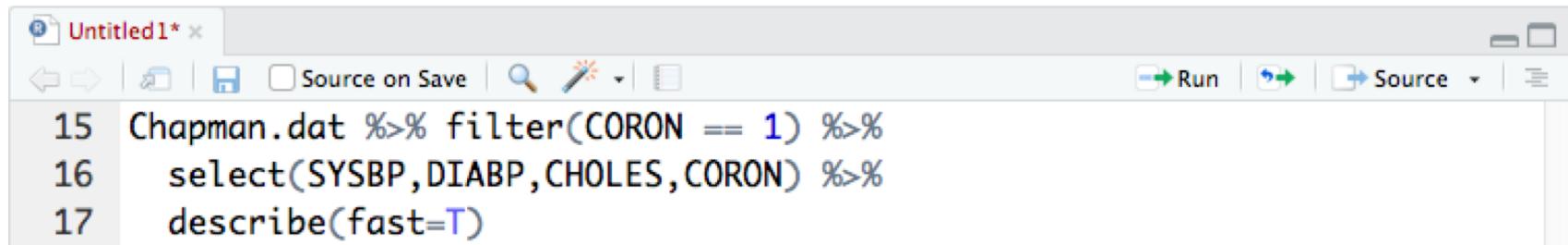
```
10 library(dplyr)
11 Chapman.dat %>% describe()
12 Chapman.dat %>% describe(fast=T)
13 Chapman.dat %>% select(SYSBP,DIABP,CHOLES,CORON) %>% describe(fast=T)
```

The code uses the pipe operator (`%>%`) to chain multiple operations: loading the `dplyr` package, describing the entire dataset, describing the dataset with fast output, and finally selecting specific columns and describing them with fast output.

- `%>%` is called a pipe operator, which is part of `dplyr` package.
- In short, the left-hand side of `%>%` will be used as the first entry of the function on the right-hand side of `%>%`.
- It is extremely useful for executing multiple operations sequentially, such as in data manipulations.

Steps 6 & 7: Let's Do Something with the Data (3)

- Select only a subset of cases and variables:
 - Use `filter` and `select` functions in `dplyr` package sequentially.
- Type the following lines and run.



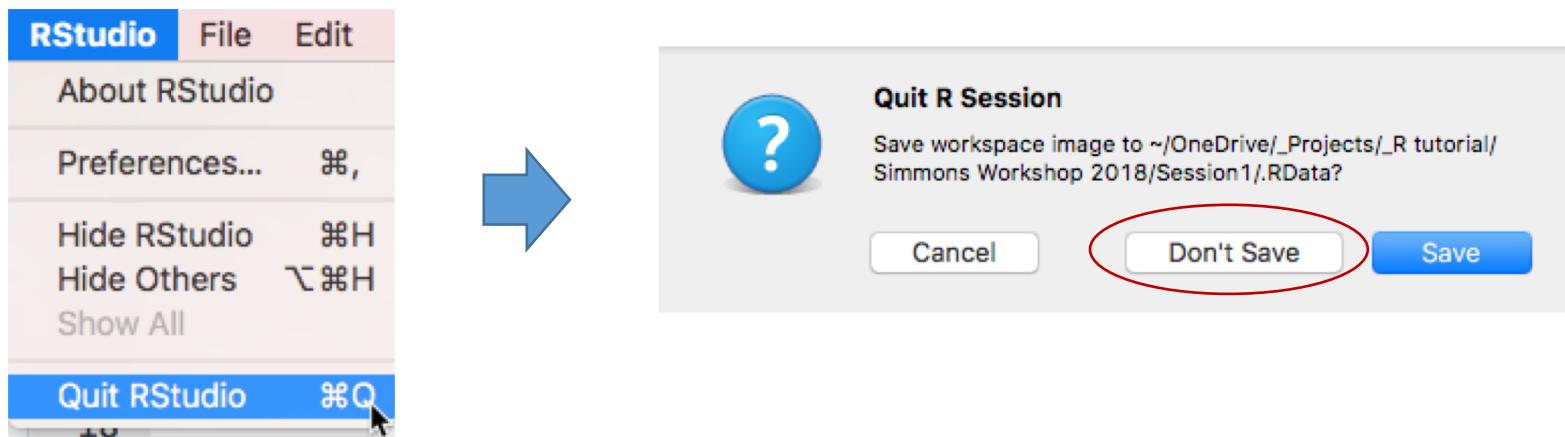
The screenshot shows the RStudio interface with an R script file named "Untitled1". The code in the script is:

```
15 Chapman.dat %>% filter(CORON == 1) %>%
16   select(SYSBP,DIABP,CHOLES,CORON) %>%
17   describe(fast=T)
```

The "Run" button in the toolbar is highlighted.

Step 9: Closing R Studio Session

- Be sure to save your syntax for the last time.
- It is highly recommended NOT to save the “workspace”, because you can easily replicate your work by running the saved syntax.



Resuming Existing Work

- Step 4: Open RStudio by double-clicking a saved R Project file (.Rproj file).



Session1.Rproj

- The target R script file will be automatically open, if it was open when you closed R Studio last time.
- Otherwise, go to “Files” tab of Rstudion, and click the target R script file name to open.
 - Alternatively, within RStudio, click button, and locate the target R script file and open.

```

1 library(haven)
2 Chapman.dat <- read_spss("Chapman.sav")
3
4 install.packages("psych")
5 library(psych)
6 describe(Chapman.dat)
7 describe(Chapman.dat, fast=T)
8
9 library(dplyr)
10 Chapman.dat %>% describe()
11 Chapman.dat %>% describe(fast=T)
12 Chapman.dat %>% select(SYSBP,DIABP,CHOLES,CORON) %>% describe(fast=T)
13
14 Chapman.dat %>% filter(CORON == 1) %>%
15   select(SYSBP,DIABP,CHOLES,CORON) %>%
16   describe(fast=T)
17

```

2:6 (Top Level) R Script

Console Terminal

~/OneDrive/_Projects/_R tutorial/UMS Workshop 2020/Session1/ ↵

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.

>

Environment History

Import Dataset

Global Environment

Environment is empty

Files Plots Connections Packages Help Viewer			
<input type="checkbox"/> New Folder	<input type="checkbox"/> Delete	<input type="checkbox"/> Rename	<input type="checkbox"/> More
<input type="checkbox"/> me	> OneDrive > _Projects > _R tutorial > UMS Workshop 2020 > Session1		
	▲ Name	Size	Modified
	..		
	.Rhistory	132 B	Jan 17, 2020, 12:39
	Chapman.sav	5.8 KB	Oct 23, 2016, 3:38 F
	Script01.R	541 B	Jan 17, 2020, 12:39
	Session1.Rproj	205 B	Jan 17, 2020, 12:39

Resuming Existing Work

- Step 5: Read data file
 - Run the syntax to read the data file.



```
library(haven)
Chapman.dat <- read_spss("Chapman.sav")
```

- Steps 6-9: The rest is the same as before!

Let's Do It

- Replicate what have been demonstrated.
 1. Create a new folder `FirstRprojedt`, and copy the data file `Chapman.sav`.
 2. Open RStudio, and create a new project for the newly created folder.
 3. Install `tidyverse` package, if you have not.
 4. Open a new script file, and write lines to (1) load `heaven` package, and (2) read the `Chapman.sav` data as `Chapman.dat` .
 5. Save the R script file by naming it `FirstRscript` .
 6. Close RStudio.
 7. Re-open RStudio by double-clicking `FirstRproject.Rprj` .

Part II

Basic Statistics on R

Basic Statistics

- We will learn:
 - Basic statistics
 - Basic graphics by `ggplot`
 - Basic data manipulations
- Let's look at:
 - 01_BasicStats.R

Part III

R Markdown

R Markdown

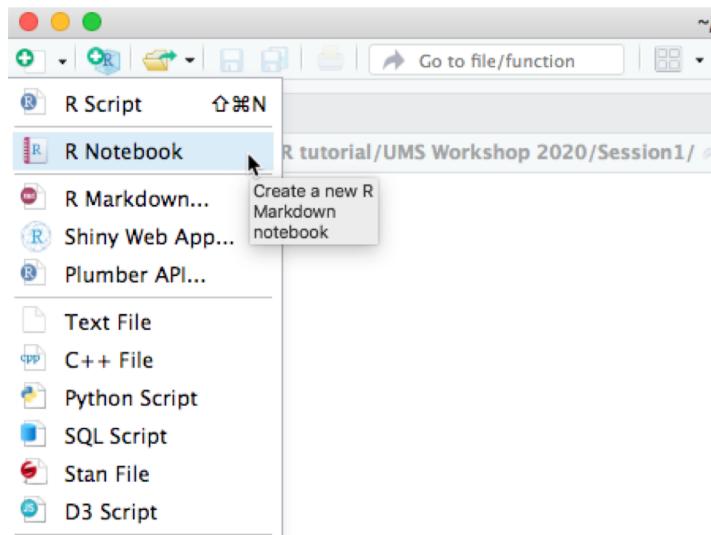
- R Markdown is a way to produce a document that can integrate what you do on R and text you want to type in.
 - R code
 - R output (e.g., table, graph)
 - Your comments about analysis
- It is an effective way to run, view, save, and share your analysis on R.
 - R Markdown can produce many kinds of document (e.g., presentation slides and APA style manuscript) in different file formats (HTML, PDF, and MSWord).
- The most basic and interactive form of R Markdown is "R Notebook", which produce an HTML file.

R Notebook

- Besides producing an external file, R Notebook provides a convenient platform to work on R codes.
 - R code and output will be presented on the same window.
 - Output remains on the R Notebook document unless codes are modified for the code chunk.
 - No need to re-run to review outputs from previous run or session.

Start R Notebook File

- First, `rmarkdown` package needs to be installed.
- Open an R Notebook file by choosing `R Notebook` command from `File` pull down menu .



- This will open a template R Notebook file that contains minimal but sufficient example to produce an HTML file.
- Once this file is saved, an HTML file will be produced in the workspace.

~/OneDrive/_Projects/_R tutorial/UMS Workshop 2020/S... Addins

Notebook01.Rmd x ABC Preview Insert Run

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you  
execute code within the notebook, the results appear beneath the code.  
7  
8 Try executing this chunk by clicking the *Run* button within the chunk or  
by placing your cursor inside it and pressing *Cmd+Shift+Enter*.  
9  
10 ``{r}  
11 plot(cars)  
12 ````  
13  
14 Add a new chunk by clicking the *Insert Chunk* button on the toolbar or  
by pressing *Cmd+Option+I*.  
15  
16 When you save the notebook, an HTML file containing the code and output  
will be saved alongside it (click the *Preview* button or press  
*Cmd+Shift+K* to preview the HTML file).  
17  
18 The preview shows you a rendered HTML copy of the contents of the editor.  
Consequently, unlike *Knit*, *Preview* does not run any R code chunks.  
Instead, the output of the chunk when it was last run in the editor is  
displayed.  
19  
20
```

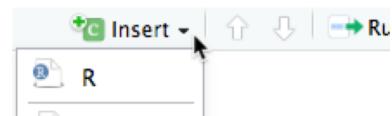
- Some more texts are added between code chunks.

- This is a “YAML header”, where some information about the document is provided, by being tagged by: --- at the beginning and at the end.

- Some texts are added here. These will appear in the HTML file to be produced.
- Some formatting appear here:
 - Hyperlink
 - Italicized text
- To change paragraph, provide an empty line in between two paragraphs.

- This is a “code chunk”, where R code is placed, by being tagged by: ``{r} at the beginning and `` at the end.

- To insert the code chunk tags:
 - Use “Insert R” pull-down menu.



- Use keyboard shortcut:
 - ⌘ + ⌥ + I (Mac).
 - cmd + option + I (Windows).
- Manually type.



Code ▾

R Notebook

This is an [R Markdown](#) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

[Hide](#)

```
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

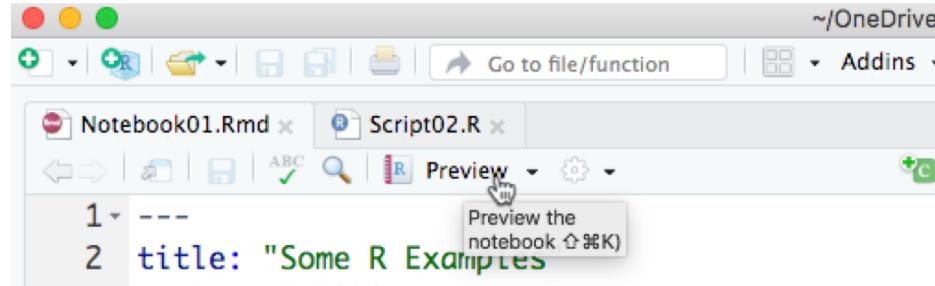
When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

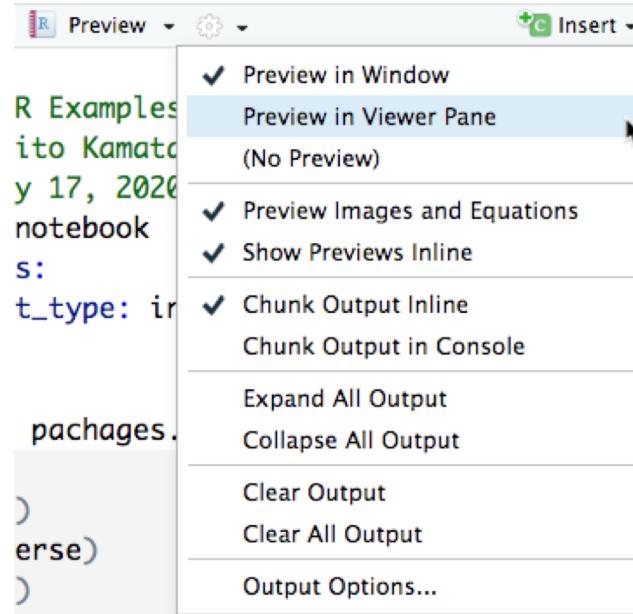
- Note that R code is in a gray box. This is a default to R Markdown.
- Texts outside of the code chunk is printed.
- However, the result of `plot(cars)` is not printed.
 - We need to actually run the code chunk.

Aside: Preview R Notebook File

- Another way to view the Notebook HTML file is by “Preview”.

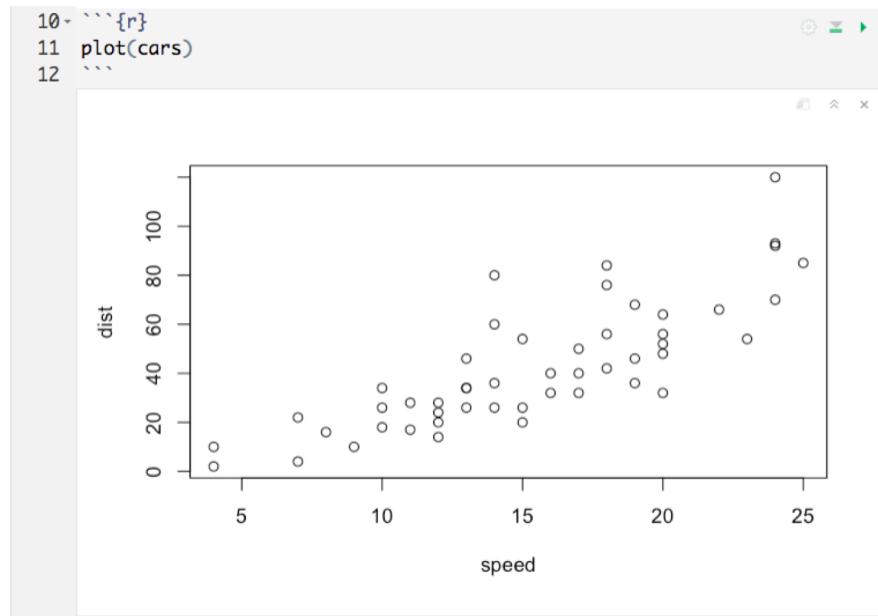


- By default, a separate preview window will open.
- Alternatively, a preview can be open within the main RStudio window.



Run Code Chunk in R Notebook File

- One way to run a code chunk is to click an arrow button for the target code chunk.
- This will create an output window right below the code chunk.



- Once the R Notebook file is saved, the HTML file will be updated.

R Notebook

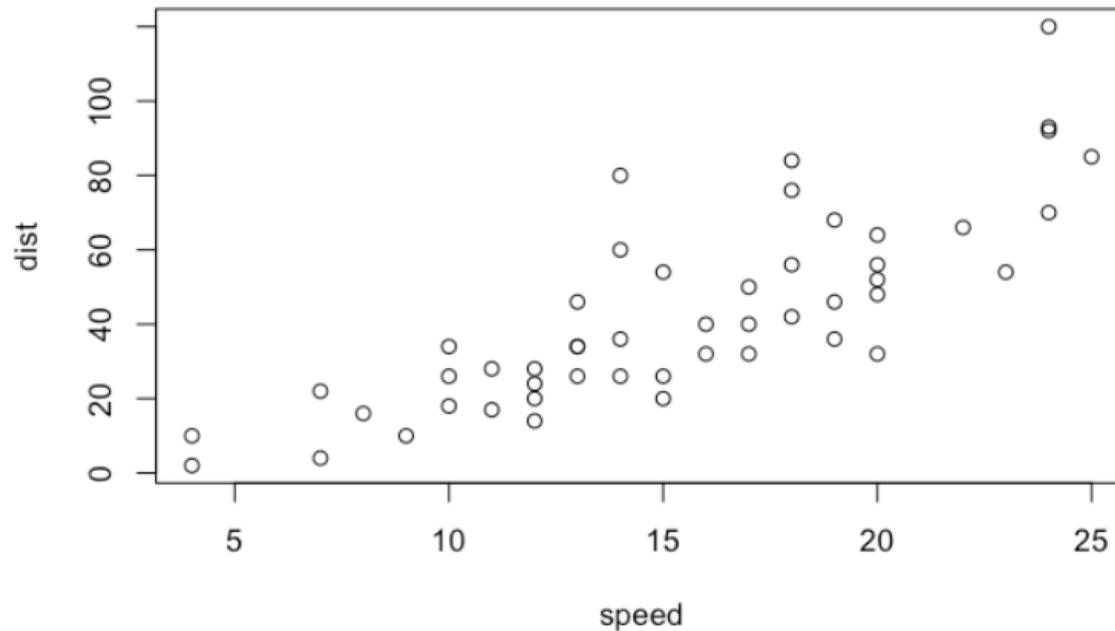
Code ▾

This is an [R Markdown](#) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

[Hide](#)

```
plot(cars)
```



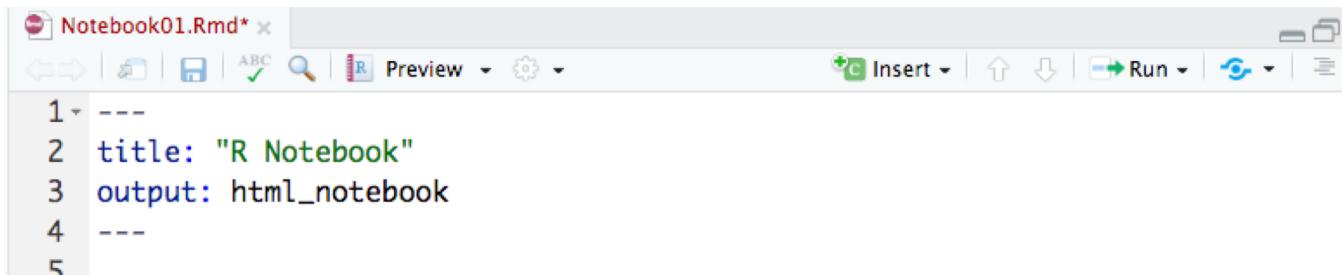
Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, Preview does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Edit R Notebook File

- We can use the template Notebook file to get what we need.
1. Delete all the contents, except the YAML header.

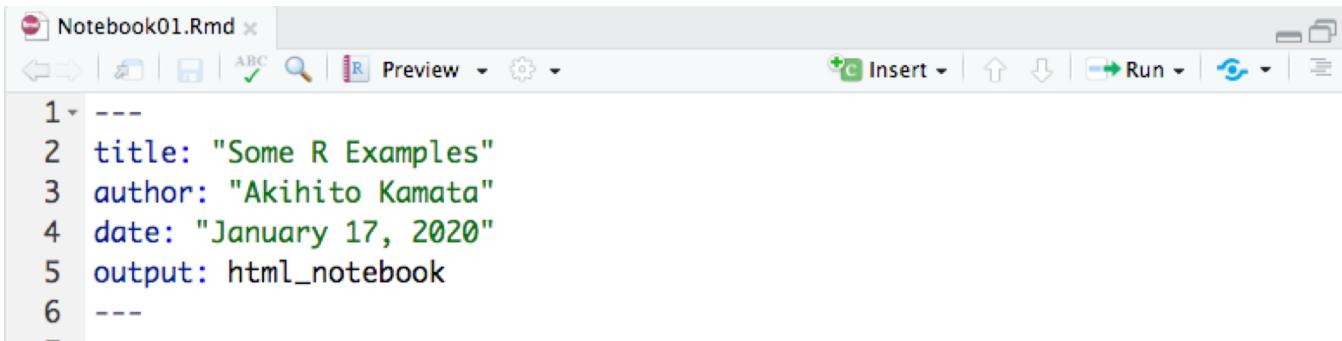


A screenshot of the RStudio interface showing a file named "Notebook01.Rmd". The code editor displays the following YAML header:

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5
```

The interface includes a toolbar with icons for file operations, preview, and run, along with a menu bar at the top.

2. Change the title. Perhaps add your name as an author and date, too.



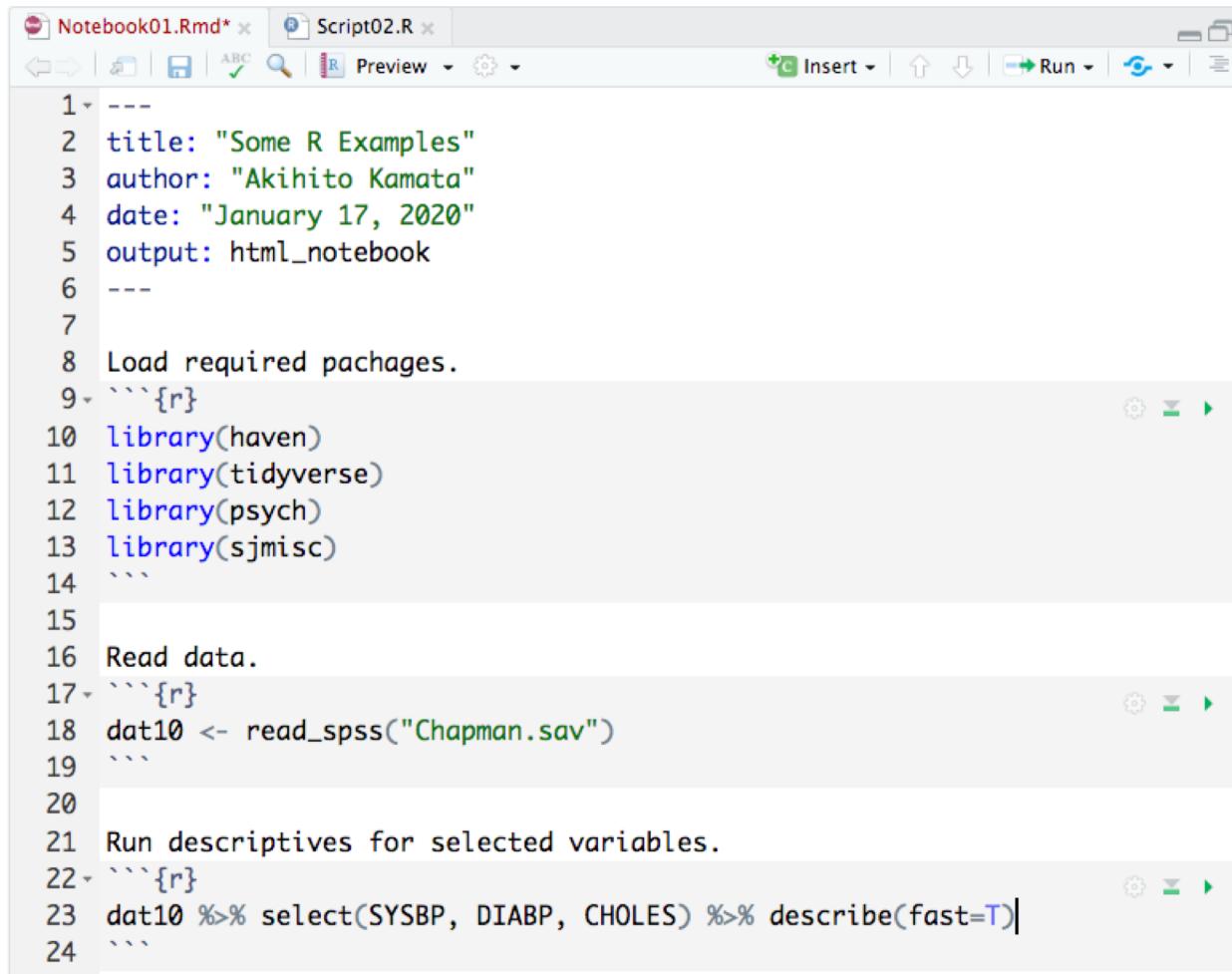
A screenshot of the RStudio interface showing the same file "Notebook01.Rmd". The code editor now displays the following YAML header with modifications:

```
1 ---  
2 title: "Some R Examples"  
3 author: "Akihito Kamata"  
4 date: "January 17, 2020"  
5 output: html_notebook  
6 ---  
7
```

The interface remains the same, showing the toolbar and menu bar.

Edit R Notebook File

3. Add some texts and code chunks.



The screenshot shows the RStudio interface with the 'Notebook01.Rmd' tab selected. The notebook contains the following R code:

```
1 ---  
2 title: "Some R Examples"  
3 author: "Akihito Kamata"  
4 date: "January 17, 2020"  
5 output: html_notebook  
---  
7  
8 Load required packages.  
9 ```{r}  
10 library(haven)  
11 library(tidyverse)  
12 library(psych)  
13 library(sjmisc)  
14 ```  
15  
16 Read data.  
17 ```{r}  
18 dat10 <- read_spss("Chapman.sav")  
19 ```  
20  
21 Run descriptives for selected variables.  
22 ```{r}  
23 dat10 %>% select(SYSBP, DIABP, CHOLESL) %>% describe(fast=T)  
24 ```
```

Edit R Notebook File

4. Run code chunks, and save the Markdown file.

The screenshot shows the RStudio interface with an R Notebook file open. The code editor pane contains the following R code:

```
1 ---  
2 title: "Some R Examples"  
3 author: "Akihito Kamata"  
4 date: "January 17, 2020"  
5 output: html_notebook  
6 ---  
7 |  
8 Load required packages.  
9 ````{r}  
10 library(haven)  
11 library(tidyverse)  
12 library(psych)  
13 library(sjmisc)  
14 ````  
15 |  
16 Read data.  
17 ````{r}  
18 dat10 <- read_spss("Chapman.sav")  
19 ````  
20 |  
21 Run descriptives for selected variables.  
22 ````{r}  
23 dat10 %>% select(SYSBP, DIABP, CHOLESL) %>% describe(fast=T)  
24 ````
```

Below the code editor is a table output from the `describe` function:

	vars	n	mean	sd	min	max	range	se
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SYSBP	1	200	121.64	16.70	90	190	100	1.18
DIABP	2	200	81.59	9.99	55	112	57	0.71
CHOLESL	3	200	285.11	65.04	135	520	385	4.60

Actually, viewing only R Notebook file with results will enhance our productivity.

There is an option to show output only to console. However, not so much recommended.

Edit R Notebook File

5. Open or refresh the HTML file.

The screenshot shows an R Notebook interface with the following content:

- Title:** Some R Examples
- Author:** Akihito Kamata
- Date:** January 17, 2020
- Description:** Load required packages.
- Code:**

```
library(haven)
library(tidyverse)
library(psych)
library(sjmisc)
```
- Description:** Read data.
- Code:**

```
dat10 <- read_spss("Chapman.sav")
```
- Description:** Run descriptives for selected variables.
- Code:**

```
dat10 %>% select(SYSBP, DIABP, CHOLESL) %>% describe(fast=T)
```

	vars	n	mean	sd	min	max	range	se
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SYSBP	1	200	121.64	16.70	90	190	100	1.18
DIABP	2	200	81.59	9.99	55	112	57	0.71
CHOLESL	3	200	285.11	65.04	135	520	385	4.60

3 rows

Preview in Viewer Pane

The screenshot shows the RStudio interface with two panes: the Editor pane on the left and the Viewer pane on the right.

Editor Pane:

- File menu: File, Open, Save, Print, Go to file/function, Addins.
- Script tabs: Notebook01.Rmd (active), Script02.R.
- Toolbar: Run, Insert, Preview, Run, Stop, Reload, Save, Close.
- Code content:

```
1 ---  
2 title: "Some R Examples"  
3 author: "Akihito Kamata"  
4 date: "January 17, 2020"  
5 output: html_notebook  
6 editor_options:  
7   chunk_output_type: inline  
8 ---  
9  
10 Load required packages.  
11 ````{r}  
12 library(haven)  
13 library(tidyverse)  
14 library(psych)  
15 library(sjmisc)  
16 ````  
17  
18 Read data.  
19 ````{r}  
20 dat10 <- read_spss("Chapman.sav")  
21 ````  
22  
23 Run descriptives for selected variables.  
24 ````{r}  
25 dat10 %>% select(SYSBP, DIABP, CHOLESL) %>% describe(fast=T)  
26 ````
```
- Output pane: Displays a table of descriptives for selected variables.

	vars	n	mean	sd	min	max	range	se
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SYSBP	1	200	121.64	16.70	90	190	100	1.18
DIABP	2	200	81.59	9.99	55	112	57	0.71
CHOLESL	3	200	285.11	65.04	135	520	385	4.60

3 rows

27

9:1 (Top Level) ▾ R Markdown ▾

Viewer Pane:

- Environment tab: Global Environment.
- Data tab: dat10 (200 obs. of 8 variables).
- Files tab: Files, Plots, Connections, Packages, Help, Viewer.
- Plots tab: Publish.
- Content:

Some R Examples

Akihito Kamata
January 17, 2020

Load required packages.

```
library(haven)
library(tidyverse)
library(psych)
library(sjmisc)
```

Read data.

```
dat10 <- read_spss("Chapman.sav")
```

Run descriptives for selected variables.

```
dat10 %>% select(SYSBP, DIABP, CHOLESL) %>% describe(fast=T)
```

	vars	n	mean	sd	min	max	range	se
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SYSBP	1	200	121.64	16.70	90	190	100	1.18
DIABP	2	200	81.59	9.99	55	112	57	0.71
CHOLESL	3	200	285.11	65.04	135	520	385	4.60

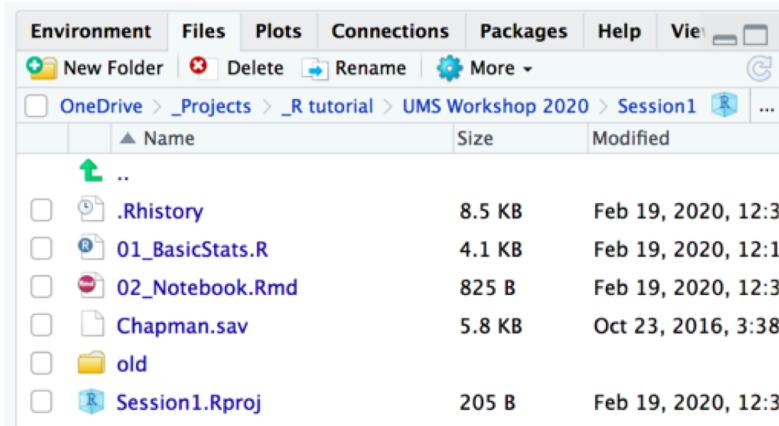
3 rows

Aside: Write Equations on R Notebook File

- We can insert symbols and equations on R Markdown files.
 - Use LATEX way of writing an equation by tagging a dollar sign (\$) at the beginning and the end of the equation.
(e.g.)
 - $\$a=150\$$ will become $a = 150$.
 - $\$\\alpha=150\$$ will become $\alpha = 150$.
 - $\$\\bar{X}=150\$$ will become $\bar{X} = 150$.
 - There are a number of online equation editor to generate LATEX equation code.
(e.g.)
 - <https://latex.codecogs.com/eqneditor/editor.php>
 - <http://www.sciweavers.org/free-online-latex-equation-editor>

Let's Do It

- Some examples are provided in `02_Notebook.Rmd` file.
- Open the file from the “Files” tab within RStudio.



- Try to edit & add some analyses and comments on the file.
- Click on “Preview” to see the changes.

Part III

Structural Equation Modeling on R

SEM on R

- We will use `lavaan` package.
 - Path analysis model for mediation analysis.
 - CFA and Measurement invariance tests.
- Let's look at:
 - `03_SEM.Rmd`

Part IV

Some Iterations on R

Iterations on R

- Let's look at:
 - 04_Iterations.Rmd