Introduction to R

Outline

- R introduction
- 執行R的環境
- R notebook
- R introduction
- Vector
- Matrix
- Factor

Why R?

Why R?

R有着非常多值得推荐的特性。

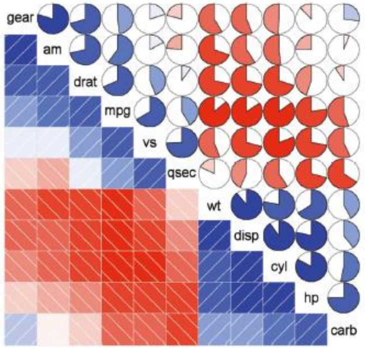
□ 多数商业统计软件价格不菲，投入成千上万美元都是可能的。而R是免费的！如果你是一位教师或一名学生，好处显而易见。

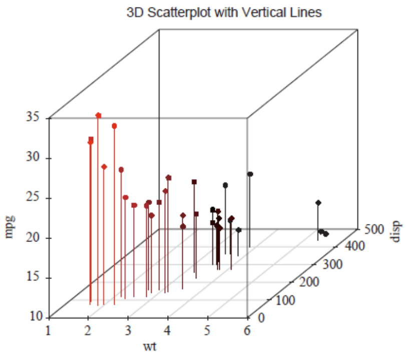
□ R是一个全面的统计研究平台，提供了各式各样的数据分析技术。几乎任何类型的数据分析工作皆可在R中完成。

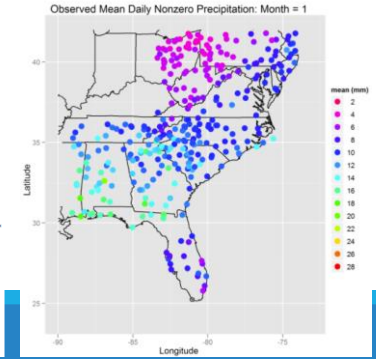
□ R拥有顶尖水准的制图功能。如果希望复杂数据可视化，那么R拥有最全面且最强大的一系列可用功能。

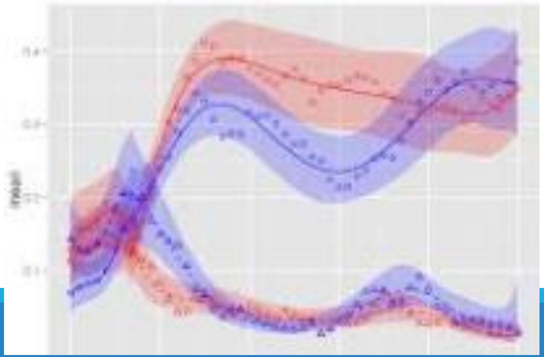
□ 从多个数据源获取并将数据转化为可用的形式，可能是一个富有挑战性的议题。R可以轻松地从各种类型的数据源导入数据，包括文本文件、数据库管理系统、统计软件，乃至专门的数据仓库。它同样可以将数据输出并写入到这些系统中。

□ R是一个无与伦比的平台，在其上可使用一种简单而直接的方式编写新的统计方法。它易于扩展，并为快速编程实现新方法提供了一套十分自然的语言。







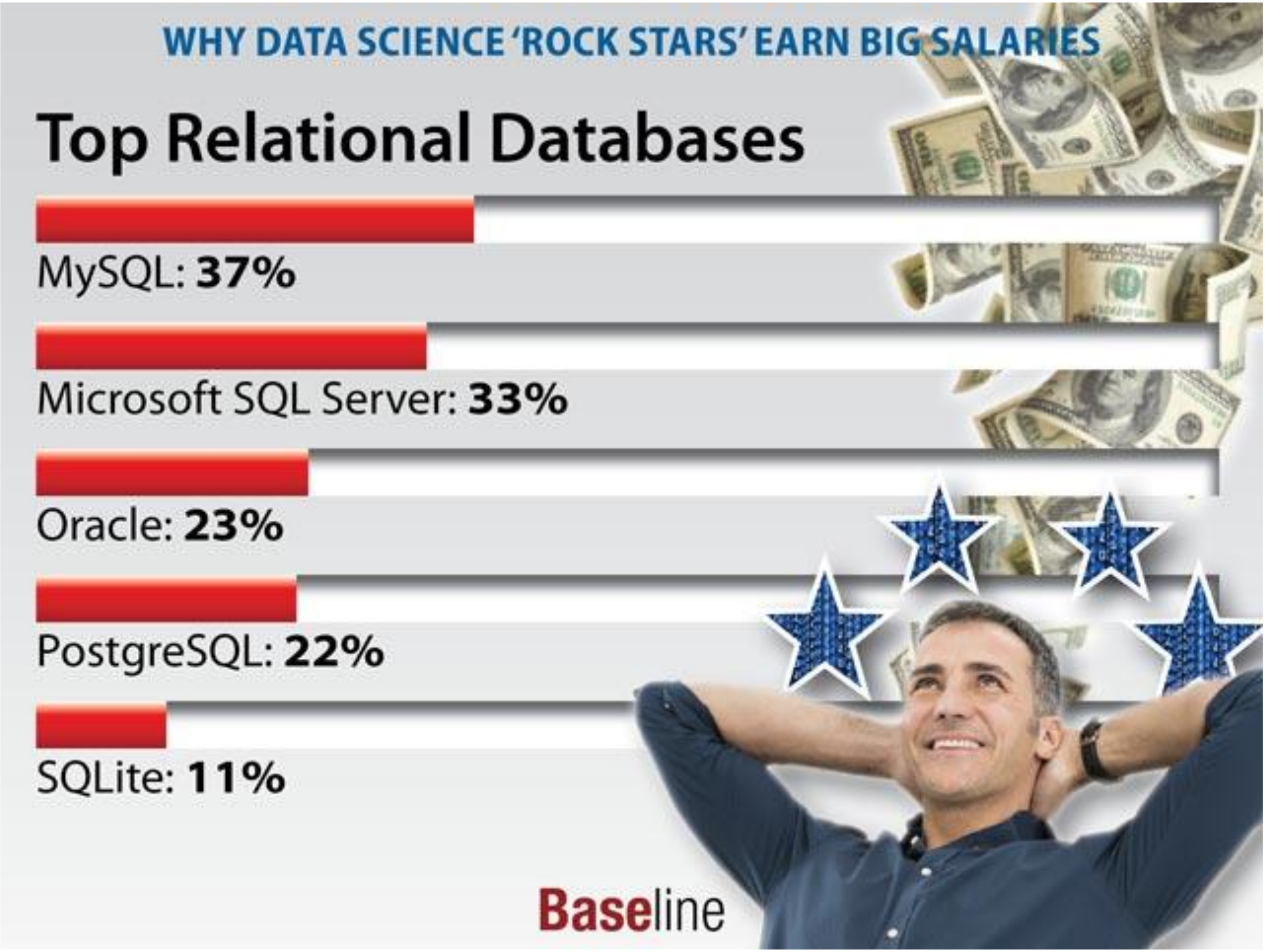


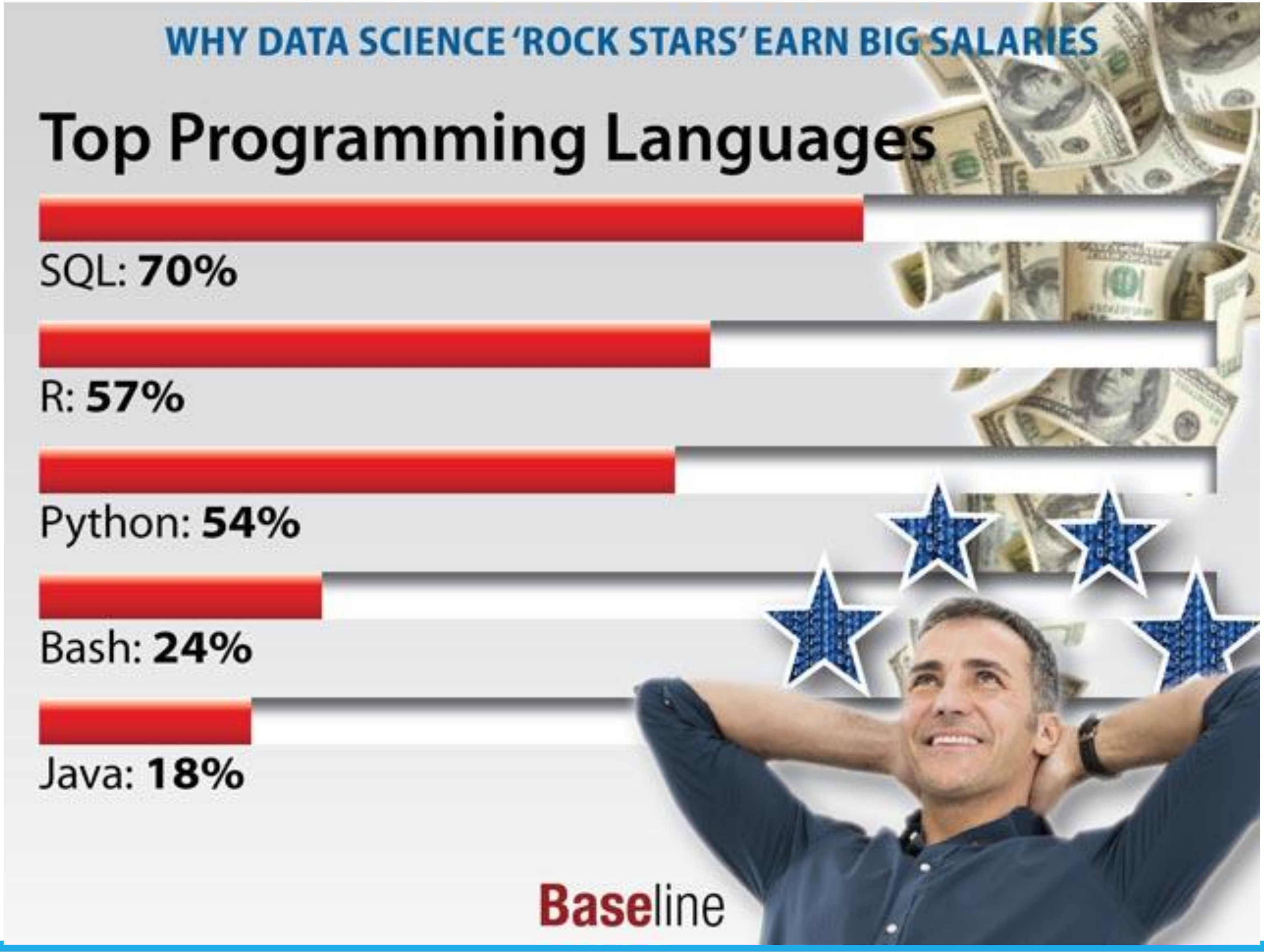
WHY DATA SCIENCE 'ROCK STARS' EARN BIG SALARIES

Paid in the USA

The median base salary of global survey respondents is $87,000 a year, but U.S. respondents make a median salary of $106,000.

Baseline

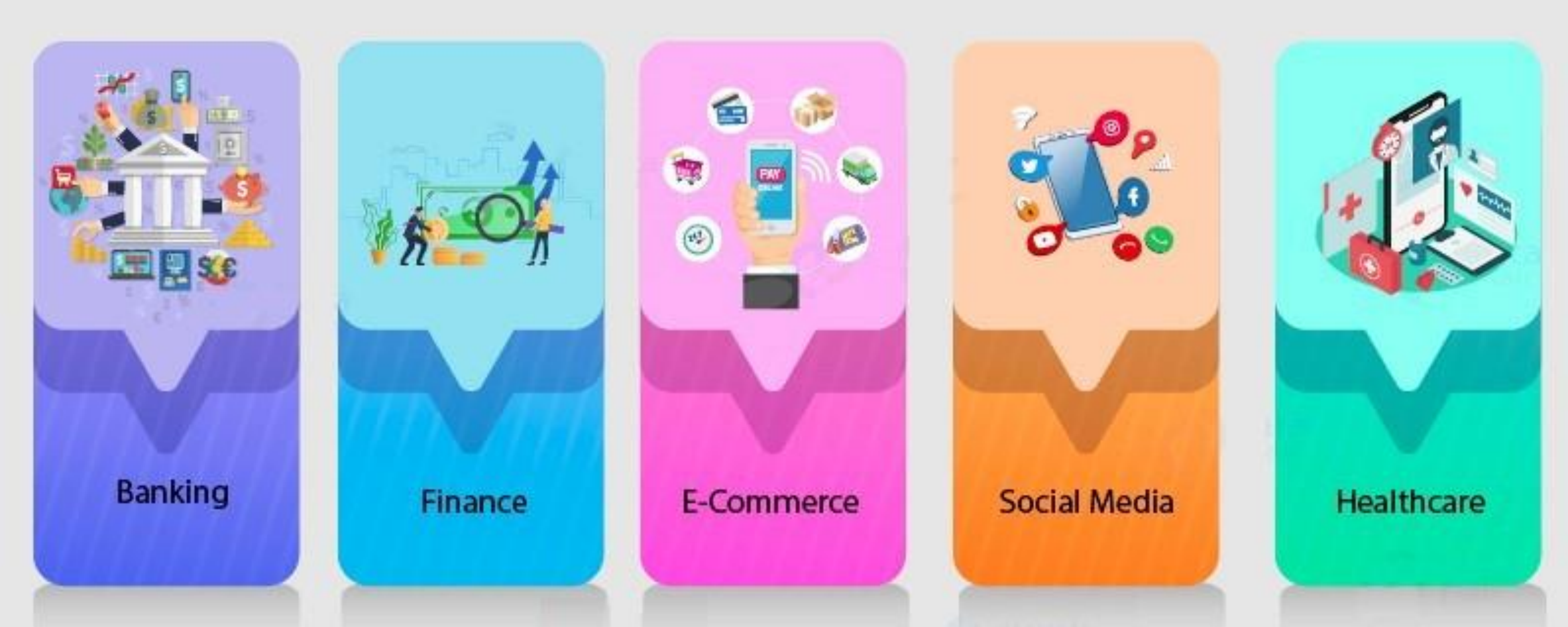

WHY DATA SCIENCE 'ROCK STARS' EARN BIG SALARIES
Top Relational Databases
MySQL: 37%
Microsoft SQL Server: 33%
Oracle: 23%
PostgreSQL: 22%
SQLite: 11%
Baseline


WHY DATA SCIENCE 'ROCK STARS' EARN BIG SALARIES
Top Programming Languages
SQL: 70%
R: 57%
Python: 54%
Bash: 24%
Java: 18%
Baseline

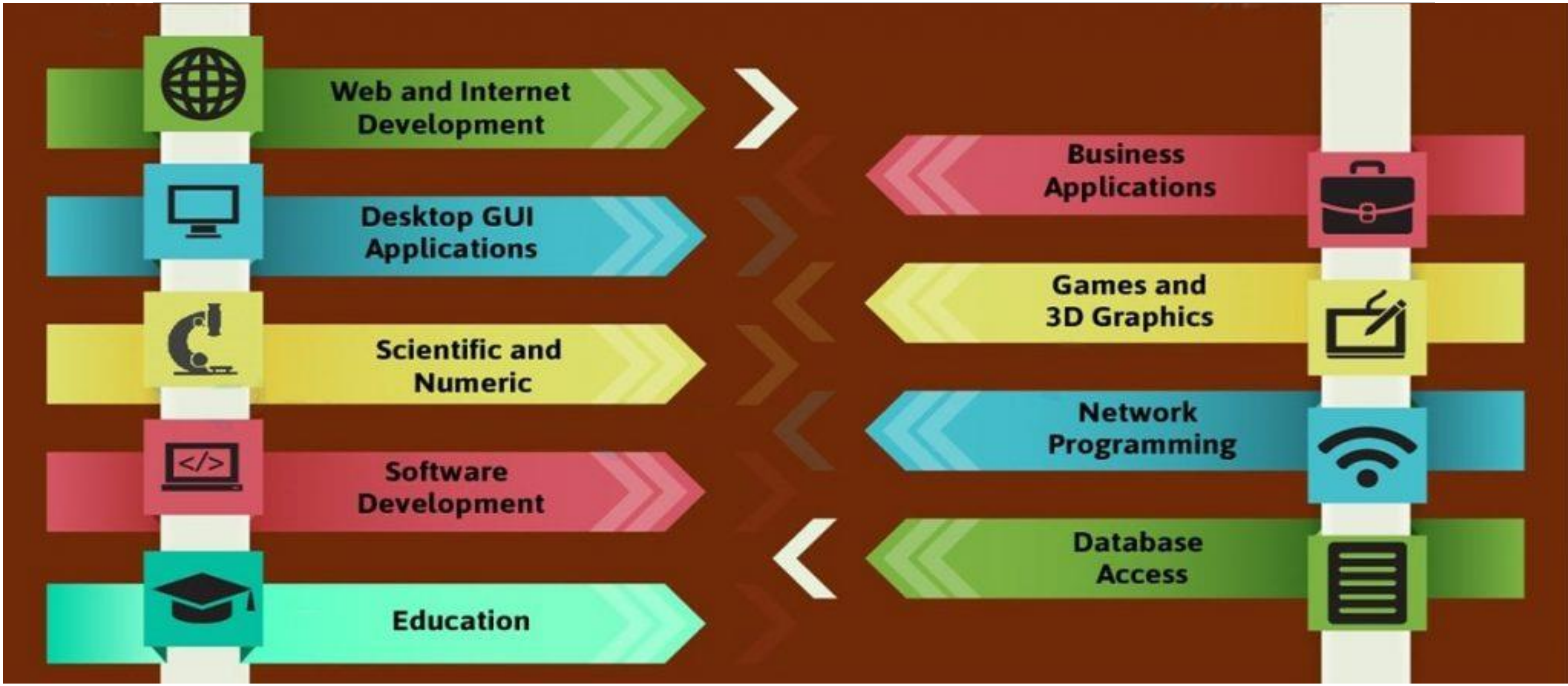
R vs Python For Data Science In 2019

https://codersera.com/blog/r-vs-python-for-data-science/

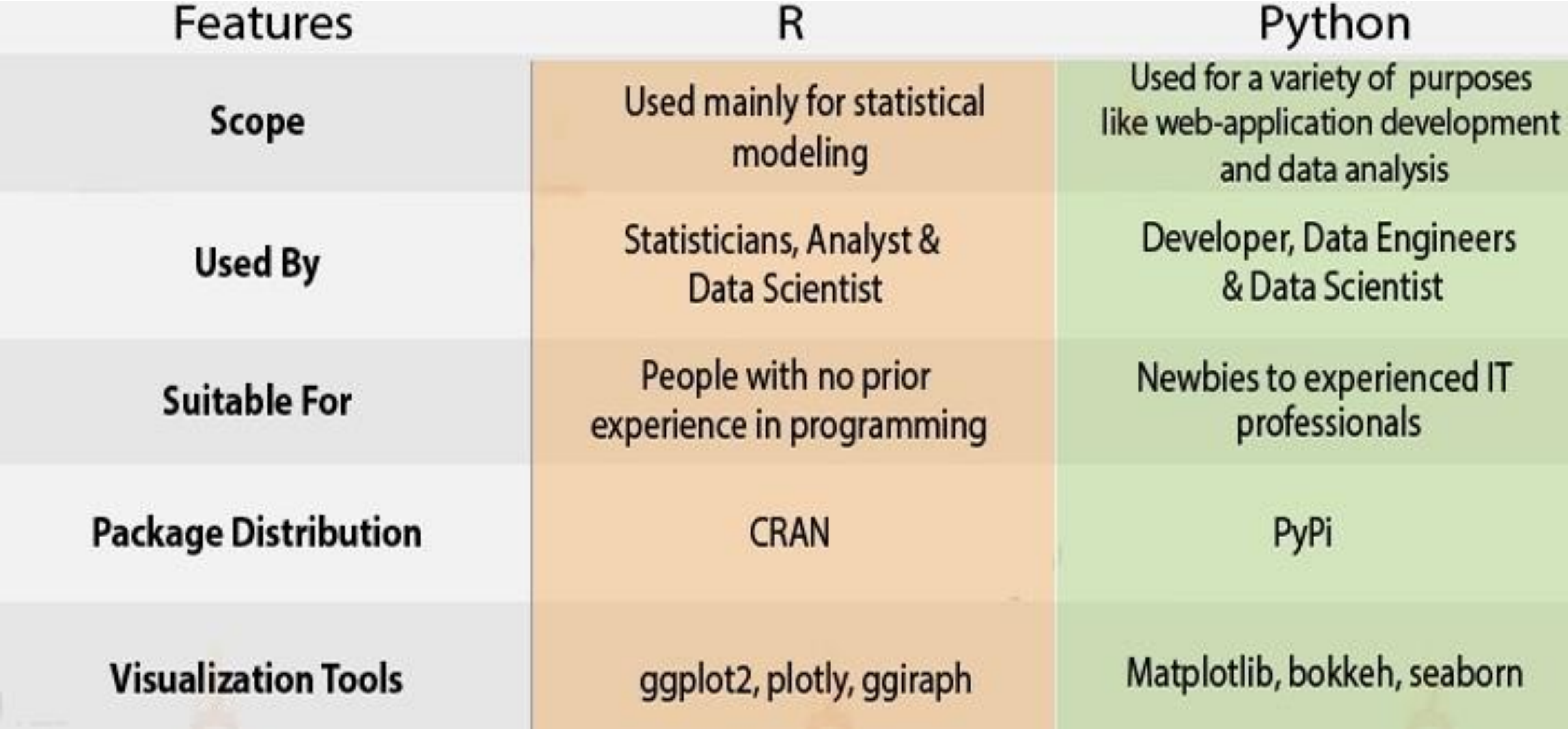
Applications of R



Applications of python



R vs Python for Data Science



Features	R	Python
Scope	Used mainly for statistical modeling	Used for a variety of purposes like web-application development and data analysis
Used By	Statisticians, Analyst & Data Scientist	Developer, Data Engineers & Data Scientist
Suitable For	People with no prior experience in programming	Newbies to experienced IT professionals
Package Distribution	CRAN	PyPi
Visualization Tools	ggplot2, plotly, ggiraph	Matplotlib, bokkeh, seaborn

Data Visualization

The Differences Between **PYTHON & R**

- A data analysis tool
- Has advanced libraries like Panda, Numpy & Scipy

- More of statistical analysis tool.
- Integrates well with other computer languages like C++, Java, and C

5 YEARS
10 YEARS
15 YEARS

Modeling Libraries in R

dplyr
forcats
ggplot2
haven
lubridate
purrr
stringr
readxl
TIBBLE
tidyr
tidyverse

Modeling libraries in Python

SciPy

NumPy

matplotlib

IP[y]: IPython
Interactive Computing

pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

scikit
learn

執行 R 的環境

R-studio on cloud9 (第一週安裝的環境)

線上工具:

http://www.r-fiddle.org

https://www.jdoodle.com/execute-r-online

R-studio on **R-studio 課程伺服器**

R-studio 課程伺服器

http://120.126.17.200:8787/

帳號: 學號 (小寫英文字母)

預設密碼: cgu學號

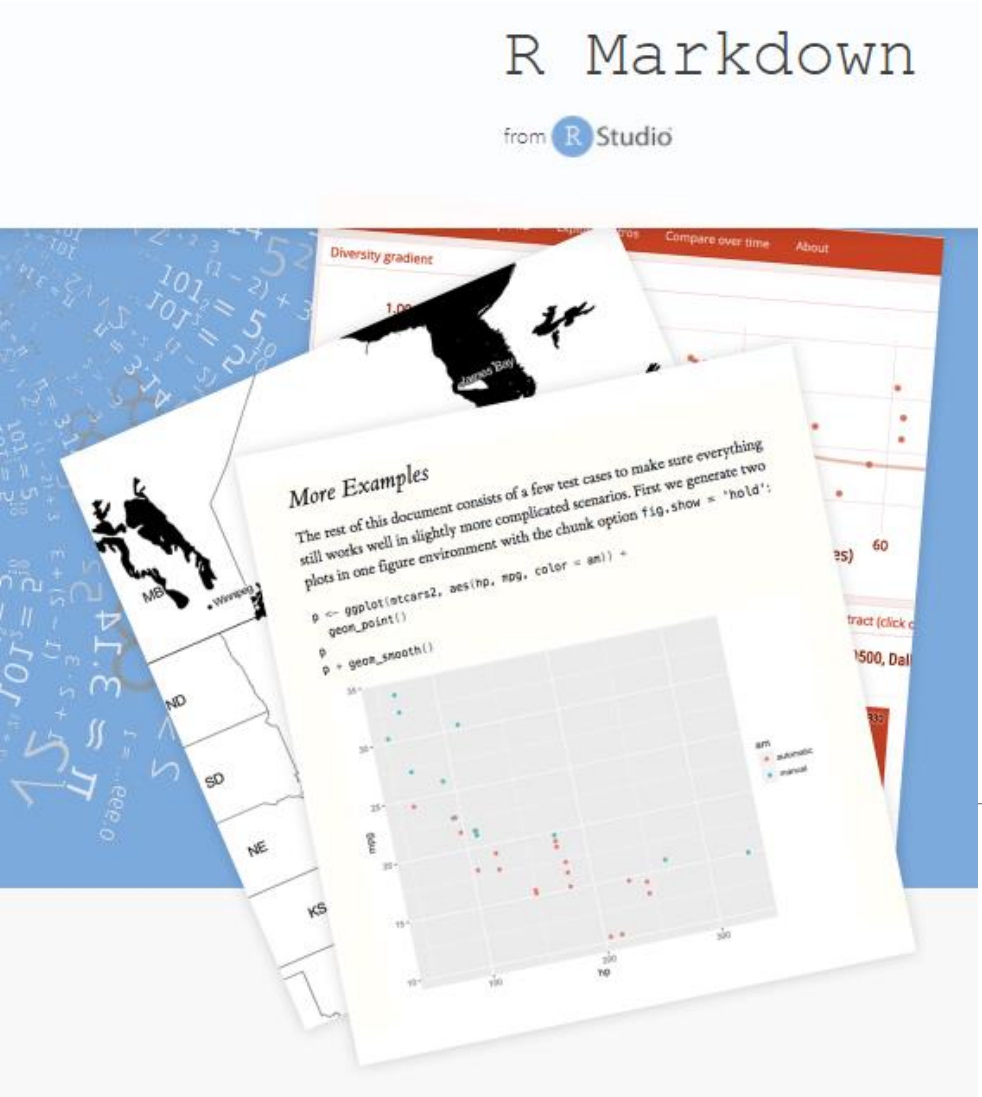
*請先用 Pietty 登入 Server, 修改自己的密碼，登入後輸入

指令: passwd

Pietty 下載: https://sites.google.com/view/pietty-project

R notebook

HTTP://120.126.17.201:8787/



R Markdown

from RStudio

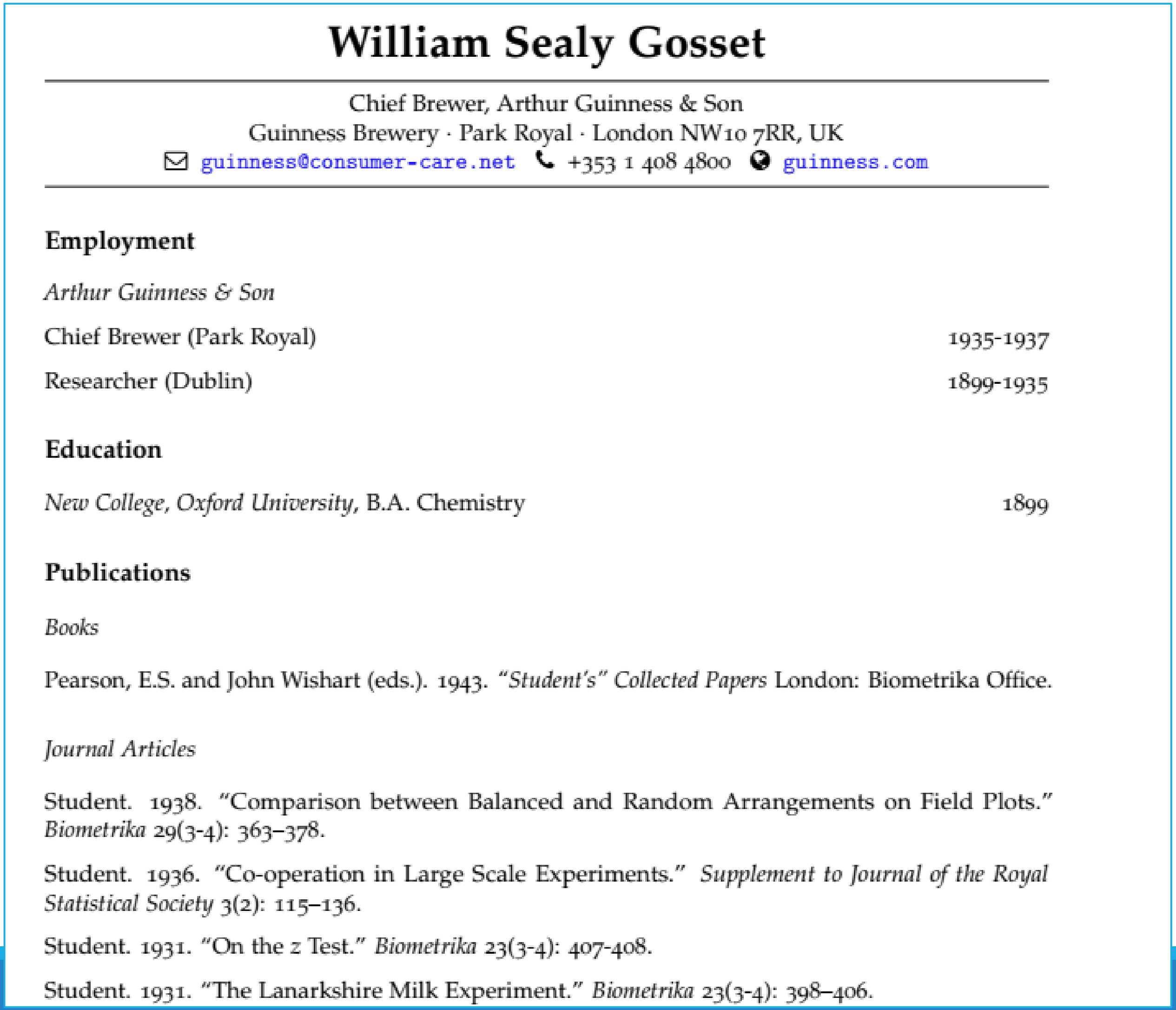
More Examples

The rest of this document consists of a few test cases to make sure everything still works well in slightly more complicated scenarios. First we generate two plots in one figure environment with the chunk option `fig.show = 'hold'`:

```
p <- ggplot(mtcars2, aes(hp, mpg, color = am)) +
  geom_point()
p
p + geom_smooth()
```

R Markdown documents are fully reproducible. Use a productive notebook interface to weave together narrative text and code to produce elegantly formatted output.

http://svmiller.com/blog/2016/03/svm-r-markdown-cv/



William Sealy Gosset

Chief Brewer, Arthur Guinness & Son
Guinness Brewery · Park Royal · London NW10 7RR, UK
✉ guinness@consumer-care.net ☎ +353 1 408 4800 🌐 guinness.com

Employment

Arthur Guinness & Son

Chief Brewer (Park Royal)	1935-1937
Researcher (Dublin)	1899-1935

Education

New College, Oxford University, B.A. Chemistry	1899

Publications

Books

Pearson, E.S. and John Wishart (eds.). 1943. *"Student's" Collected Papers* London: Biometrika Office.

Journal Articles

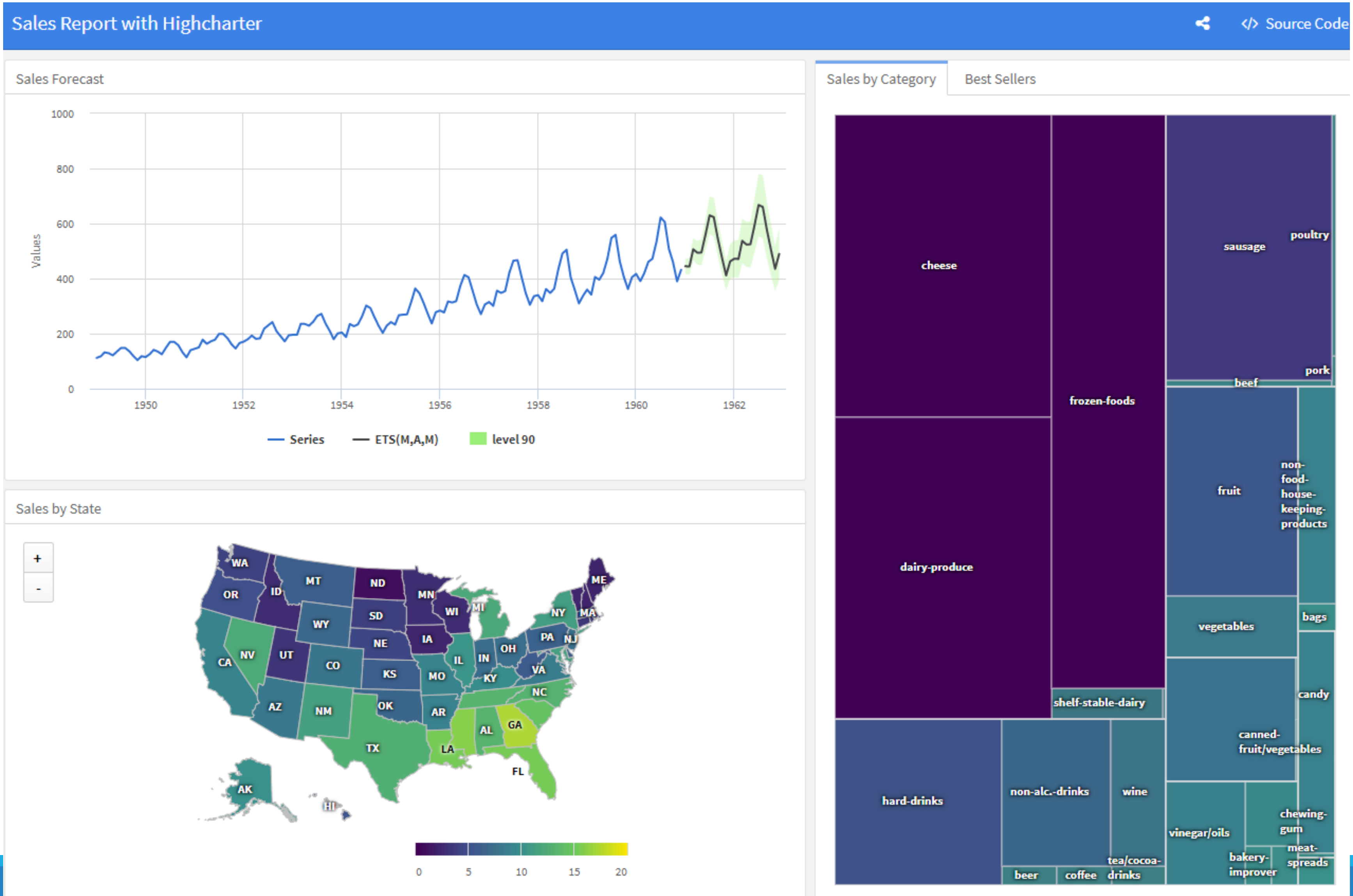
Student. 1938. "Comparison between Balanced and Random Arrangements on Field Plots." *Biometrika* 29(3-4): 363–378.

Student. 1936. "Co-operation in Large Scale Experiments." *Supplement to Journal of the Royal Statistical Society* 3(2): 115–136.

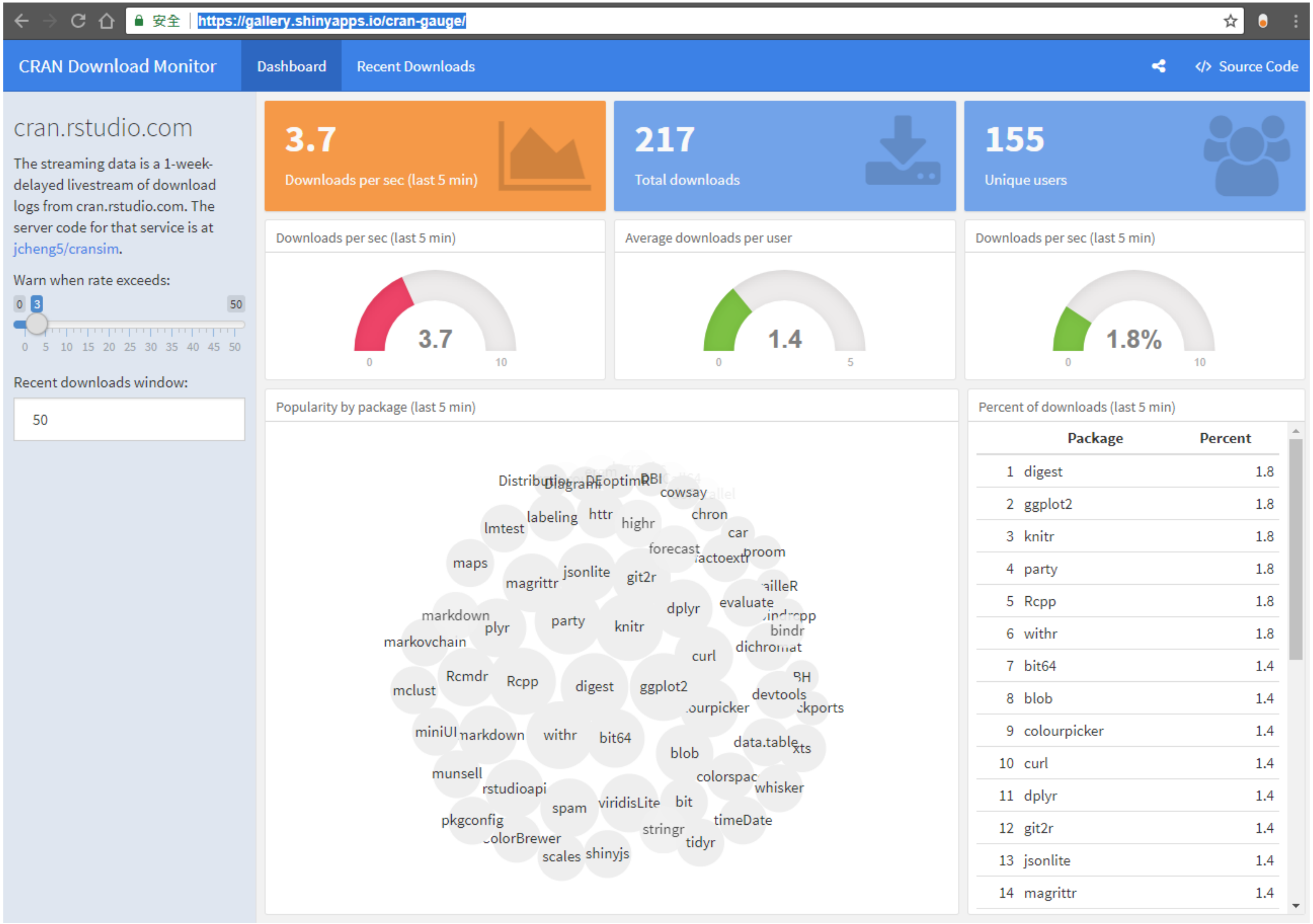
Student. 1931. "On the *z* Test." *Biometrika* 23(3-4): 407-408.

Student. 1931. "The Lanarkshire Milk Experiment." *Biometrika* 23(3-4): 398–406.

https://beta.rstudioconnect.com/jjallaire/htmlwidgets-highcharter/htmlwidgets-highcharter.html#sales-by-category



https://gallery.shinyapps.io/cran-gauge/



cran.rstudio.com

The streaming data is a 1-week-delayed livestream of download logs from cran.rstudio.com. The server code for that service is at jcheng5/cransim.

Warn when rate exceeds: 3

Recent downloads window: 50

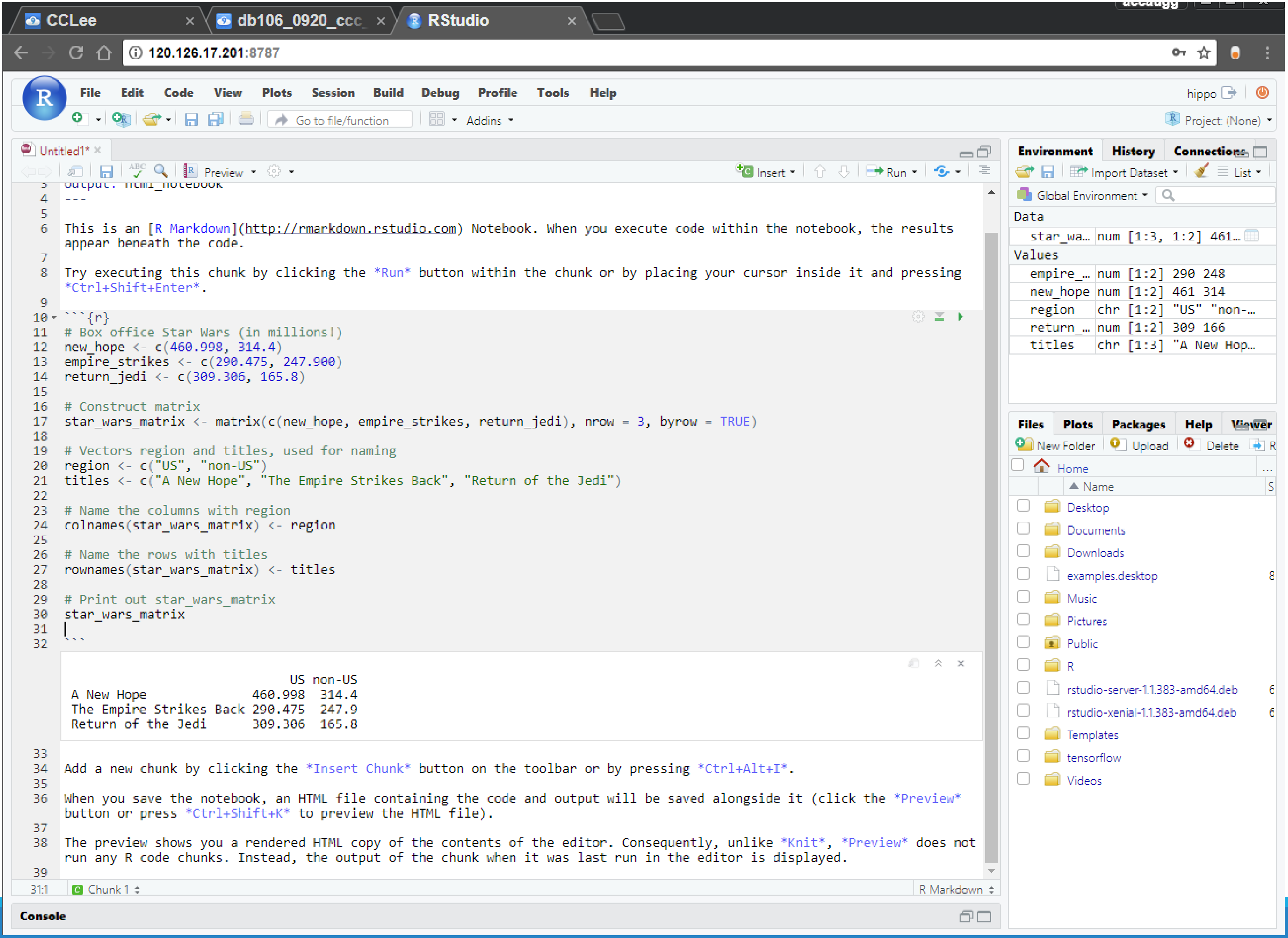
3.7 Downloads per sec (last 5 min)

217 Total downloads

155 Unique users

Percent of downloads (last 5 min)

	Package	Percent
1	digest	1.8
2	ggplot2	1.8
3	knitr	1.8
4	party	1.8
5	Rcpp	1.8
6	withr	1.8
7	bit64	1.4
8	blob	1.4
9	colourpicker	1.4
10	curl	1.4
11	dplyr	1.4
12	git2r	1.4
13	jsonlite	1.4
14	magrittr	1.4



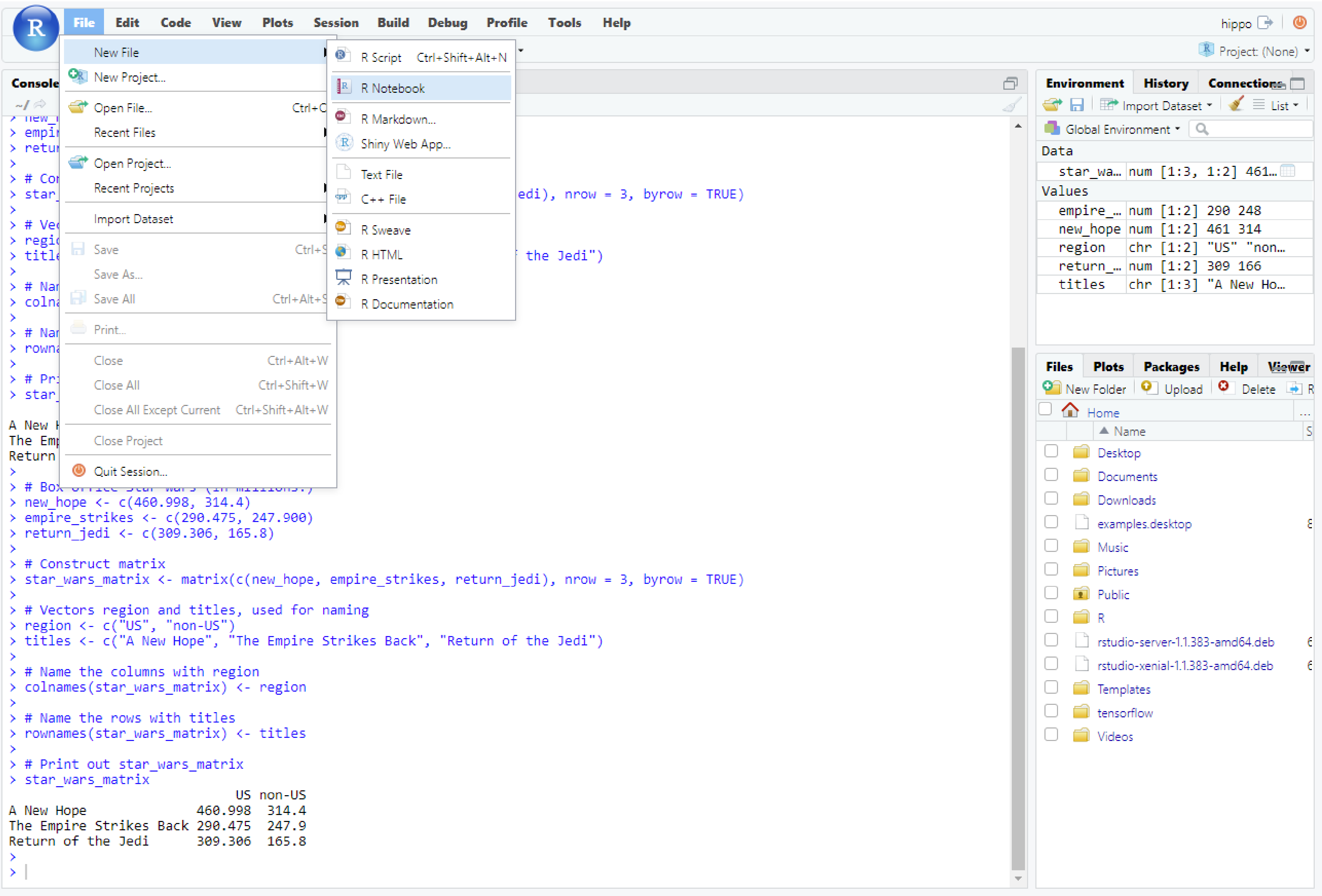
~~~
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) 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
*Ctrl+Shift+Enter*.

```{r}
# Box office Star Wars (in millions!)
new_hope <- c(460.998, 314.4)
empire_strikes <- c(290.475, 247.900)
return_jedi <- c(309.306, 165.8)

# Construct matrix
star_wars_matrix <- matrix(c(new_hope, empire_strikes, return_jedi), nrow = 3, byrow = TRUE)

# Vectors region and titles, used for naming
region <- c("US", "non-US")
titles <- c("A New Hope", "The Empire Strikes Back", "Return of the Jedi")

# Name the columns with region
colnames(star_wars_matrix) <- region

# Name the rows with titles
rownames(star_wars_matrix) <- titles

# Print out star_wars_matrix
star_wars_matrix

```
~~~

```
                             US non-US
A New Hope              460.998  314.4
The Empire Strikes Back 290.475  247.9
Return of the Jedi      309.306  165.8
```

```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+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 *Ctrl+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.

```

```
Data
  star_wa… num [1:3, 1:2] 461…
Values
  empire_… num [1:2] 290 248
  new_hope num [1:2] 461 314
  region   chr [1:2] "US" "non-…
  return_… num [1:2] 309 166
  titles   chr [1:3] "A New Hop…
```

- □ Desktop
- □ Documents
- □ Downloads
- □ examples.desktop
- □ Music
- □ Pictures
- □ Public
- □ R
- □ rstudio-server-1.1.383-amd64.deb
- □ rstudio-xenial-1.1.383-amd64.deb
- □ Templates
- □ tensorflow
- □ Videos



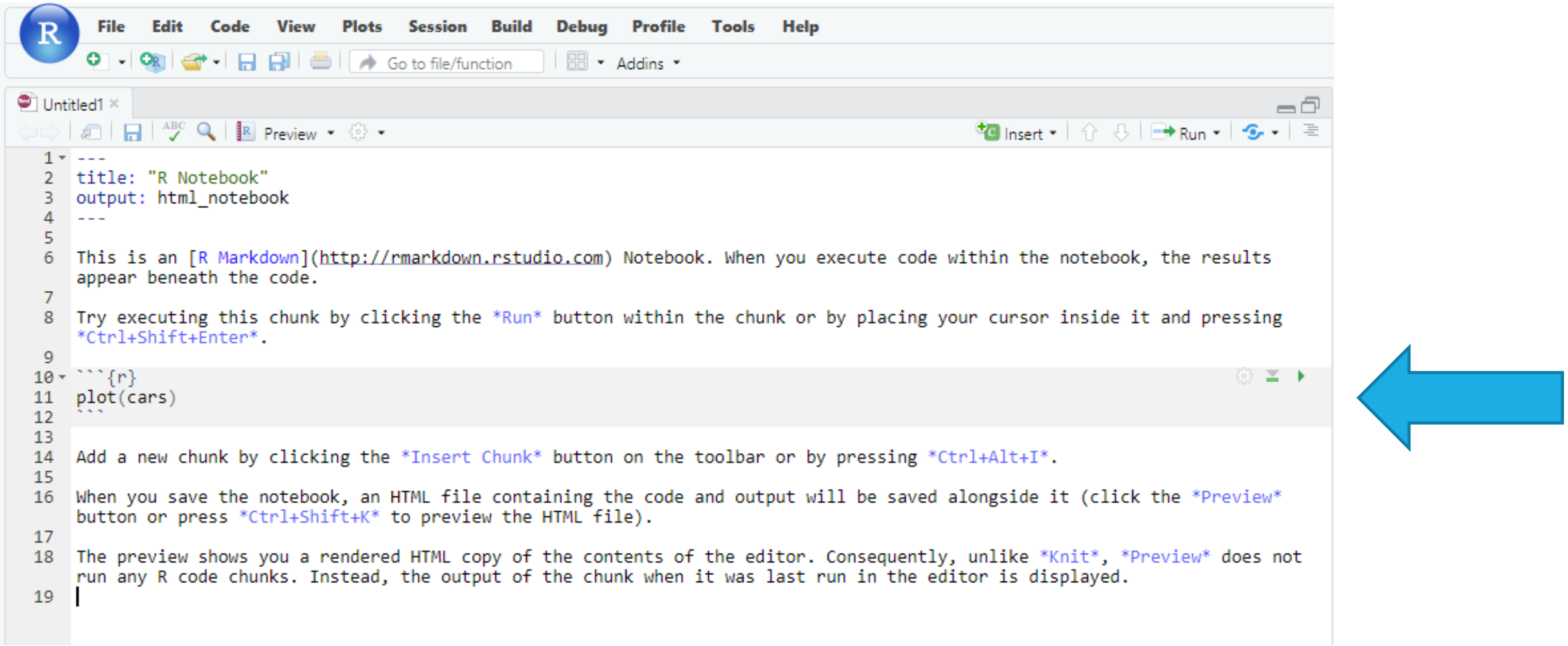
```
> new_
> empi
> retu
>
> # Co
> star_                                        edi), nrow = 3, byrow = TRUE)
>
> # Ve
> regi
> titl                                         f the Jedi")
>
> # Na
> colna
>
> # Na
> rowna
>
> # Pr
> star_

A New H
The Emp
Return
>
> # Box office Star Wars (in millions!)
> new_hope <- c(460.998, 314.4)
> empire_strikes <- c(290.475, 247.900)
> return_jedi <- c(309.306, 165.8)
>
> # Construct matrix
> star_wars_matrix <- matrix(c(new_hope, empire_strikes, return_jedi), nrow = 3, byrow = TRUE)
>
> # Vectors region and titles, used for naming
> region <- c("US", "non-US")
> titles <- c("A New Hope", "The Empire Strikes Back", "Return of the Jedi")
>
> # Name the columns with region
> colnames(star_wars_matrix) <- region
>
> # Name the rows with titles
> rownames(star_wars_matrix) <- titles
>
> # Print out star_wars_matrix
> star_wars_matrix
                             US non-US
A New Hope              460.998  314.4
The Empire Strikes Back 290.475  247.9
Return of the Jedi      309.306  165.8
>
> |
```

Data

star_wa...	num [1:3, 1:2] 461...

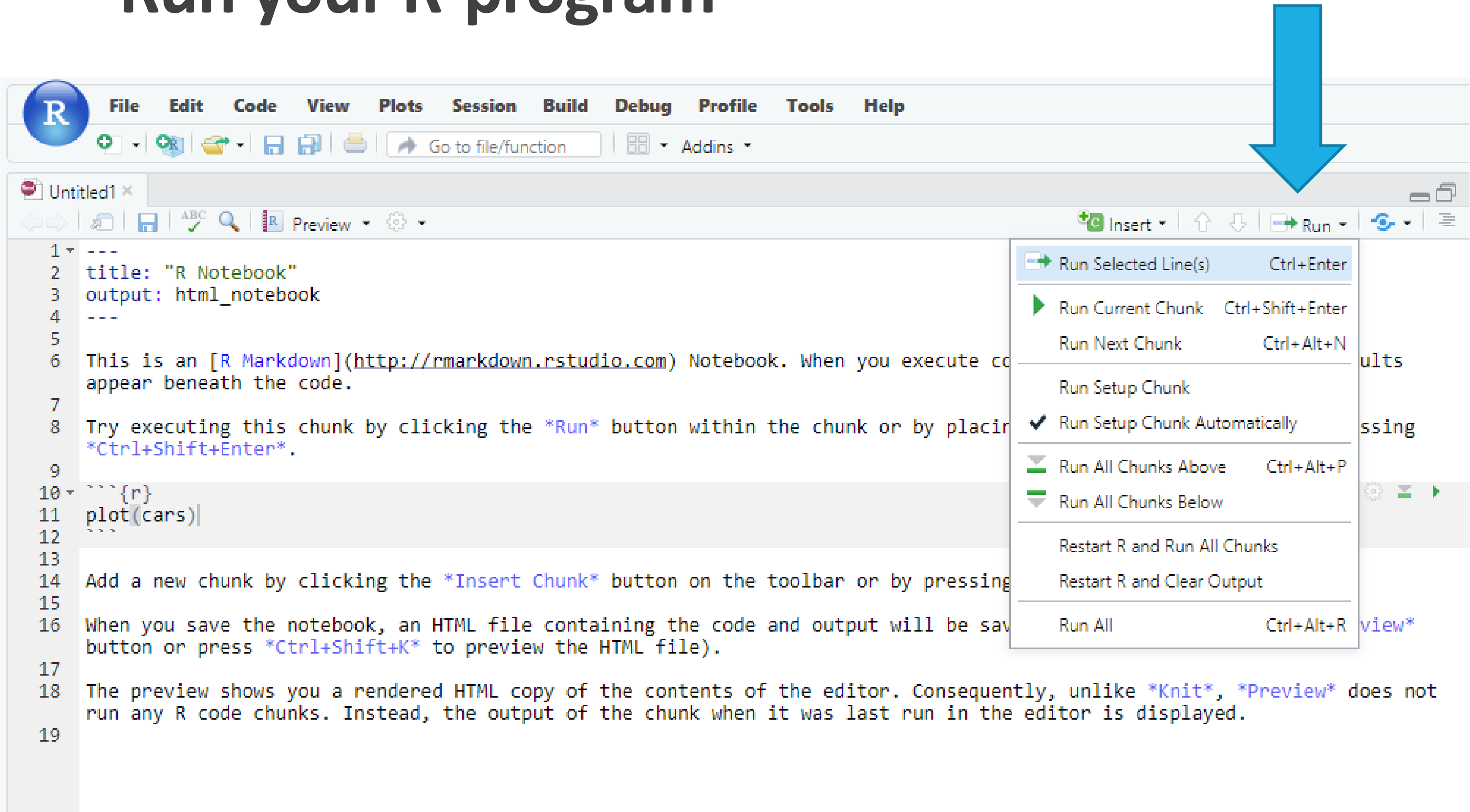
Values

empire_...	num [1:2] 290 248
new_hope	num [1:2] 461 314
region	chr [1:2] "US" "non...
return_...	num [1:2] 309 166
titles	chr [1:3] "A New Ho...



~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) 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 *Ctrl+Shift+Enter*.

```{r}
plot(cars)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+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 *Ctrl+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.
~~~

Run your R-program



File
Edit
Code
View
Plots
Session
Build
Debug
Profile
Tools
Help
Go to file/function
Addins
Untitled1
Preview
Insert
Run
Run Selected Line(s) Ctrl+Enter
Run Current Chunk Ctrl+Shift+Enter
Run Next Chunk Ctrl+Alt+N
Run Setup Chunk
✓ Run Setup Chunk Automatically
Run All Chunks Above Ctrl+Alt+P
Run All Chunks Below
Restart R and Run All Chunks
Restart R and Clear Output
Run All Ctrl+Alt+R

~~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute co ults
appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placir ssing
*Ctrl+Shift+Enter*.

```{r}
plot(cars)
```

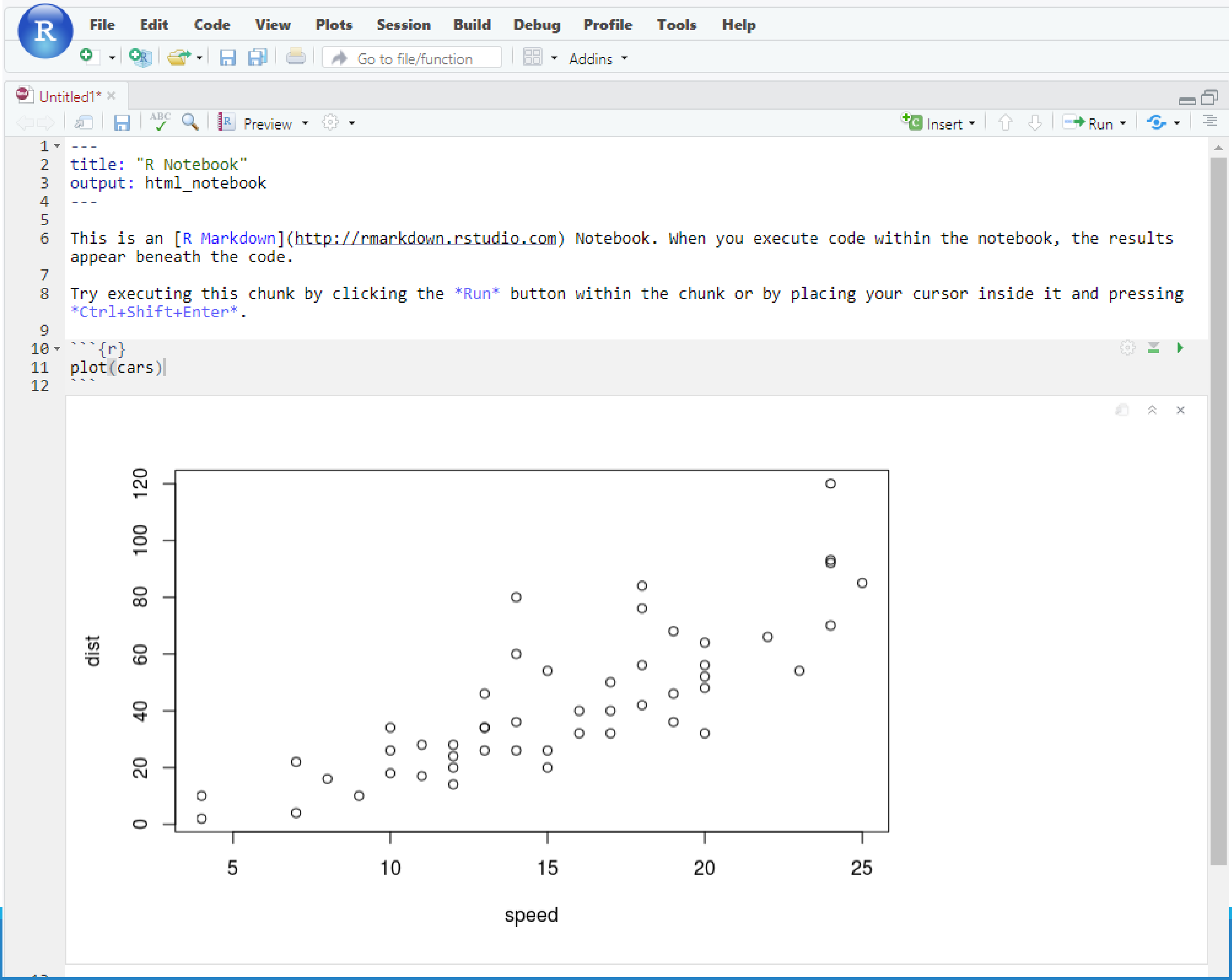

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing

When you save the notebook, an HTML file containing the code and output will be sav view*
button or press *Ctrl+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.

~~~~

結果直接顯示在block下方



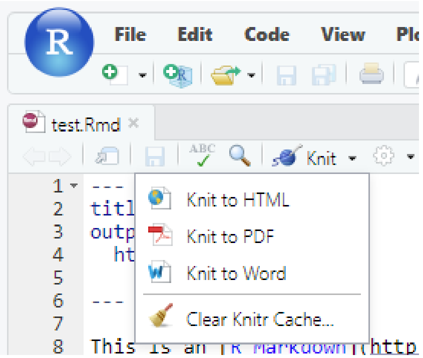
~~~
---
title: "R Notebook"
output: html_notebook
---

This is an [R Markdown](http://rmarkdown.rstudio.com) 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 *Ctrl+Shift+Enter*.

```{r}
plot(cars)
```
~~~

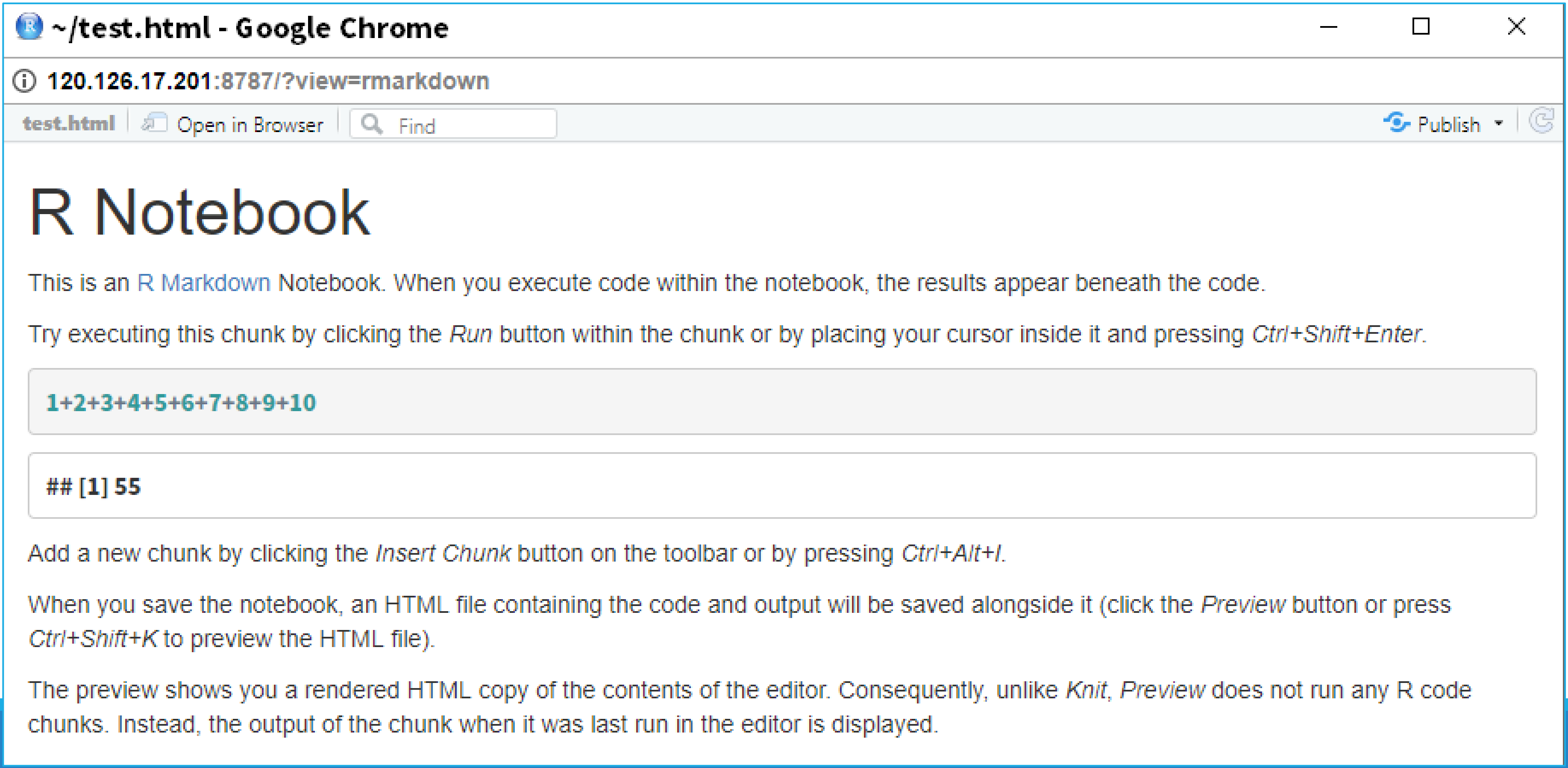
Try it!!

使用 R notebook

執行 1+2+3+4+5+6+7+8+9+10

輸出成 HTML文件




~/test.html - Google Chrome
120.126.17.201:8787/?view=rmarkdown
test.html
Open in Browser
Find
Publish

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 *Ctrl+Shift+Enter*.

```
1+2+3+4+5+6+7+8+9+10
```

```
## [1] 55
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+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 *Ctrl+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.

Markdown 語法

標題:

```
#測試標題1

##測試標題2

###測試標題3
```

強調文字:

```
*粗體字1*

_粗體字2_

**粗體字1**

__粗體字2__
```

超連結:

```
[Google]: http://google.com/
[Yahoo]: http://tw.yahoo.com/

Please [Google] it!
Go to [Yahoo] web site.
```

段落:

```
<p>text</p>
```

分隔線:

```
***
- - -
```

文字顏色:

```
<font color="red">紅色文字</font>
```

今日點名

R Notebook example

請用 R Notebook 製作一份CV，使用word格式輸出，word檔案開啟畫面截圖 + 程式碼 (一起貼在powerpoint)

內容須有:

- 姓名
- 學號
- e-mail (須以超連結表示)
- 本學期修課科目
- 專題、研究領域
- 程式語言經驗描述

格式需有主標題、分段落

Arithmetic with R

In its most basic form, R can be used as a simple calculator.

Consider the following arithmetic operators:

Addition: +

Subtraction: -

Multiplication: *

Division: /

Exponentiation: ^

Modulo: %%

```
# An addition
5 + 5

# A subtraction
5 - 5

# A multiplication
3 * 5

 # A division
(5 + 5) / 2

# Exponentiation
5^3

# Modulo
5 %% 2
```

Variable assignment

A basic concept in (statistical) programming is called a variable.

A variable allows you to store a value (e.g. 4) or an object (e.g. a function description) in R. You can then later use this variable's name to easily access the value or the object that is stored within this variable.

You can assign a value 4 to a variable my_var with the command

```
my_var <- 4
```

```
# Print out the value of the variable my_apples

my_var
```

Variable assignment (2)

```
# Assign a value to the variable my_apples
my_apples <- 5


# Fix the assignment of my_oranges
my_oranges <- "six"


# Create the variable my_fruit and print it out
my_fruit <- my_apples + my_oranges
my_fruit
```

Basic data types in R

R is case sensitive!

R works with numerous data types. Some of the most basic types to get started are:

Decimals values like 4.5 are called numerics.

Natural numbers like 4 are called integers. Integers are also numerics.

Boolean values (TRUE or FALSE) are called logical.

Text (or string) values are called characters.

使用 class(變數) 來顯示 data type

```
# Declare variables of different types:
my_numeric <- 42
my_character <- "universe"
my_logical <- FALSE

# Check class of my_numeric
class(my_numeric)

# Check class of my_character
class(my_character)

# Check class of my_logical
class(my_logical)
```

Vector

Vector

In R, you create a vector with the combine function c().

You place the vector elements separated by a comma between the parentheses.

```
numeric_vector <- c(1, 10, 49)
character_vector <- c("a", "b", "c")

# Complete the code for boolean_vector
boolean_vector <-
```

Naming a vector

```
poker_vector <- c(140, -50, 20, -120, 240)

poker_vector
```

```
# Assign days as names of poker_vector

names(poker_vector) <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")

poker_vector
```

Naming a vector (2)

```
# Poker winnings from Monday to Friday

poker_vector <- c(140, -50, 20, -120, 240)



# The variable days_vector

days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday",
"Friday")


# Assign the names of the day to poker_vector

names(poker_vector) <- days_vector
```

Vector 計算

```
A_vector <- c(1, 2, 3)
B_vector <- c(4, 5, 6)

# Take the sum of A_vector and B_vector
total_vector <- A_vector + B_vector

# Print out total_vector
total_vector
```

Vector 計算 (2)

```
A_vector <- c(1, 2, 3)

B_vector <- c(4, 5, 6)


# Take the sum of A_vector and B_vector

total_vector <- A_vector * B_vector


# Print out total_vector

total_vector
```

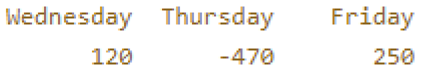
計算 Vector 的總和

```
poker_vector <- c(140, -50, 20, -120, 240)

total_poker <- sum(poker_vector)

total_poker
```

Vector selection

```
# Poker and roulette winnings from Monday to Friday:
poker_vector <- c(140, -50, 20, -120, 240)
roulette_vector <- c(-24, -50, 100, -350, 10)
days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
names(poker_vector) <- days_vector
names(roulette_vector) <- days_vector


# Define a new variable based on a selection
poker_wednesday <- poker_vector[3]
poker_wednesday    (PS. name 也會一併選進來)
```

1. 個別 輸出週三、四、五 的poker_vector 與 roulette_vector 的合



```
Wednesday  Thursday    Friday
      120      -470       250
```

2. 計算 週三、四、五 poker_vector 與 roulette_vector 相乘的結果

連續選 roulette_vector[2:5]

```
roulette_vector <- c(-24, -50, 100, -350, 10)

days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")

names(roulette_vector) <- days_vector


roulette_selection_vector <- roulette_vector[2:5]

roulette_selection_vector
```

利用 name 來選, 計算平均

```
poker_vector <- c(140, -50, 20, -120, 240)

days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")

names(poker_vector) <- days_vector

poker_start <- poker_vector[c("Monday", "Tuesday", "Wednesday")]


# Calculate the average of the elements in poker_start

mean(poker_start)
```

Selection by comparison

< for less than

> for greater than

<= for less than or equal to

>= for greater than or equal to

== for equal to each other

!= not equal to each other

```
# Poker and roulette winnings from Monday to Friday:
poker_vector <- c(140, -50, 20, -120, 240)
roulette_vector <- c(-24, -50, 100, -350, 10)
days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
names(poker_vector) <- days_vector
names(roulette_vector) <- days_vector

# Which days did you make money on poker?
selection_vector <- poker_vector > 0

# Print out selection_vector
selection_vector
```

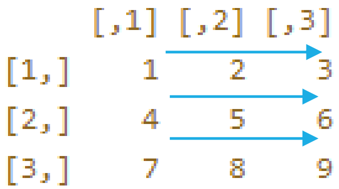
印出 True 的元素

```
# Poker and roulette winnings from Monday to Friday:
poker_vector <- c(140, -50, 20, -120, 240)
roulette_vector <- c(-24, -50, 100, -350, 10)
days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
names(poker_vector) <- days_vector
names(roulette_vector) <- days_vector

# Which days did you make money on poker?
selection_vector <- poker_vector > 0
selection_vector

# Select from poker_vector these days
poker_winning_days <- poker_vector[selection_vector]
poker_winning_days
```

Matrix

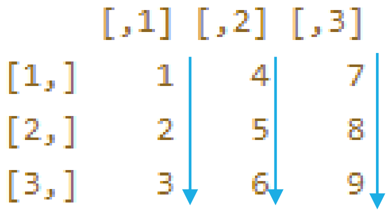
A matrix is a collection of elements of the same data type (numeric, character, or logical) arranged into a fixed number of rows and columns.

```
matrix(1:9, byrow = TRUE, nrow = 3)
```



```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

```
matrix(1:9, byrow = FALSE, nrow = 3)
```



```
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

Three rows

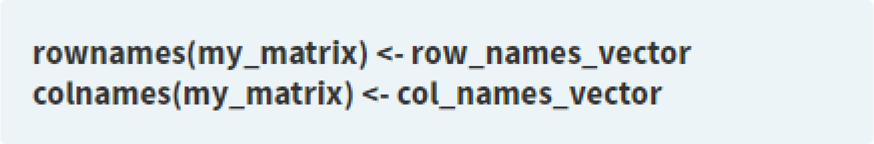
3. 如何做出下圖的結果?

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

Combine Vector to Matrix

```
# Box office Star Wars (in millions!)
new_hope <- c(460.998, 314.4)
empire_strikes <- c(290.475, 247.900)
return_jedi <- c(309.306, 165.8)

# Create box_office
box_office <- c(new_hope, empire_strikes, return_jedi)

# Construct star_wars_matrix
star_wars_matrix <- matrix(box_office, nrow = 3, byrow = TRUE)
star_wars_matrix
```

Naming a matrix



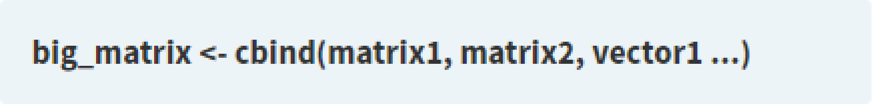
```
rownames(my_matrix) <- row_names_vector
colnames(my_matrix) <- col_names_vector
```

```
# Box office Star Wars (in millions!)
new_hope <- c(460.998, 314.4)
empire_strikes <- c(290.475, 247.900)
return_jedi <- c(309.306, 165.8)

# Construct matrix
star_wars_matrix <- matrix(c(new_hope, empire_strikes, return_jedi), nrow = 3, byrow =
TRUE)

# Vectors region and titles, used for naming
region <- c("US", "non-US")
titles <- c("A New Hope", "The Empire Strikes Back", "Return of the Jedi")

# Name the columns with region
colnames(star_wars_matrix) <- region

# Name the rows with titles
rownames(star_wars_matrix) <- titles

# Print out star_wars_matrix
star_wars_matrix
```

Matrix calculation

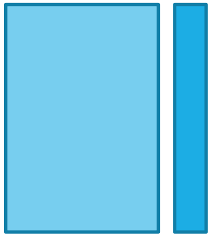
```
# Construct star_wars_matrix

box_office <- c(460.998, 314.4, 290.475, 247.900, 309.306, 165.8)

star_wars_matrix <- matrix(box_office, nrow = 3, byrow = TRUE,

dimnames = list(
        c("A New Hope", "The Empire Strikes Back", "Return of the Jedi"),
        c("US", "non-US")
        )

)


# Calculate worldwide box office figures

worldwide_vector <- rowSums(star_wars_matrix)

worldwide_vector
```

Adding a column ➔ Matrix



```
big_matrix <- cbind(matrix1, matrix2, vector1 ...)
```

```
# Construct star_wars_matrix

box_office <- c(460.998, 314.4, 290.475, 247.900, 309.306, 165.8)

star_wars_matrix <- matrix(box_office, nrow = 3, byrow = TRUE,

dimnames = list(c("A New Hope", "The Empire Strikes Back", "Return of the Jedi"), c("US", "non-US")))
```

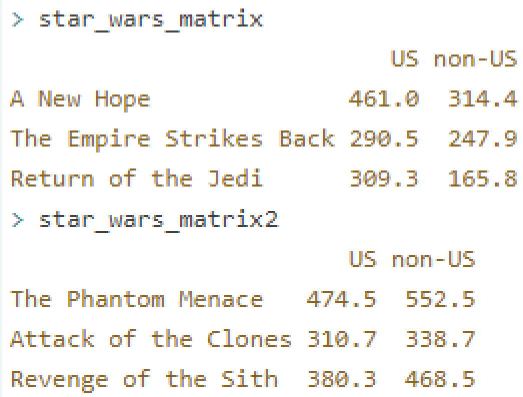


```
# The worldwide box office figures

worldwide_vector <- rowSums(star_wars_matrix)

worldwide_vector
```

```
# Bind the new variable worldwide_vector as a column to star_wars_matrix

all_wars_matrix <- cbind(star_wars_matrix, worldwide_vector)

all_wars_matrix
```

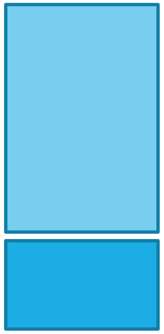
Adding a row ➔ Matrix

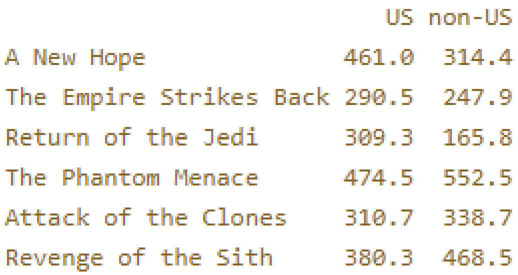
```
#

star_wars_matrix

star_wars_matrix2
```



```
> star_wars_matrix
                           US non-US
A New Hope              461.0  314.4
The Empire Strikes Back 290.5  247.9
Return of the Jedi      309.3  165.8
> star_wars_matrix2
                        US non-US
The Phantom Menace   474.5  552.5
Attack of the Clones 310.7  338.7
Revenge of the Sith  380.3  468.5
```

```
# Combine both Star Wars trilogies in one matrix

all_wars_matrix <- rbind(star_wars_matrix, star_wars_matrix2)

all_wars_matrix
```



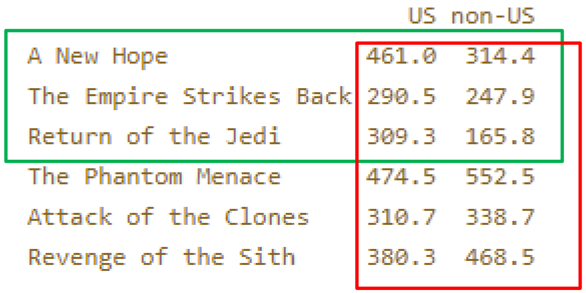


```
                           US non-US
A New Hope              461.0  314.4
The Empire Strikes Back 290.5  247.9
Return of the Jedi      309.3  165.8
The Phantom Menace      474.5  552.5
Attack of the Clones    310.7  338.7
Revenge of the Sith     380.3  468.5
```

Selection of matrix elements

all_wars_matrix[1,2] selects the element at the first row and second column.

all_wars_matrix[1:3,1:2] results in a matrix with the data on the rows 1, 2, 3 and columns 1, 2.



```
                           US non-US
A New Hope              461.0  314.4
The Empire Strikes Back 290.5  247.9
Return of the Jedi      309.3  165.8
The Phantom Menace      474.5  552.5
Attack of the Clones    310.7  338.7
Revenge of the Sith     380.3  468.5
```

Arithmetic with matrices

Similar to what you have learned with vectors,

the standard operators like +, -, /, *, etc.

work in an element-wise way on matrices in R.

```
# all_wars_matrix is available in your workspace
all_wars_matrix

# Estimate the visitors
visitors <- all_wars_matrix / 5

# Print the estimate to the console
visitors
```

4. 完成下列程式碼&輸出結果

# all_wars_matrix and ticket_prices_matrix are available in your workspace

all_wars_matrix

ticket_prices_matrix

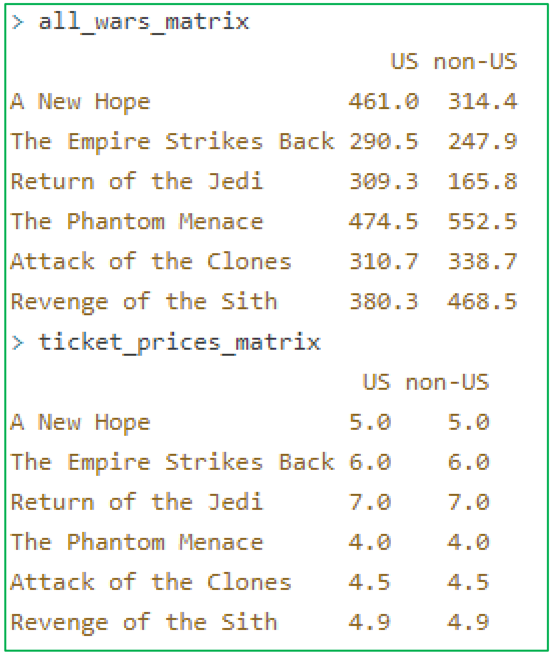
# Estimated number of visitors

visitors <-

# US visitors

us_visitors <-

# Average number of US visitors

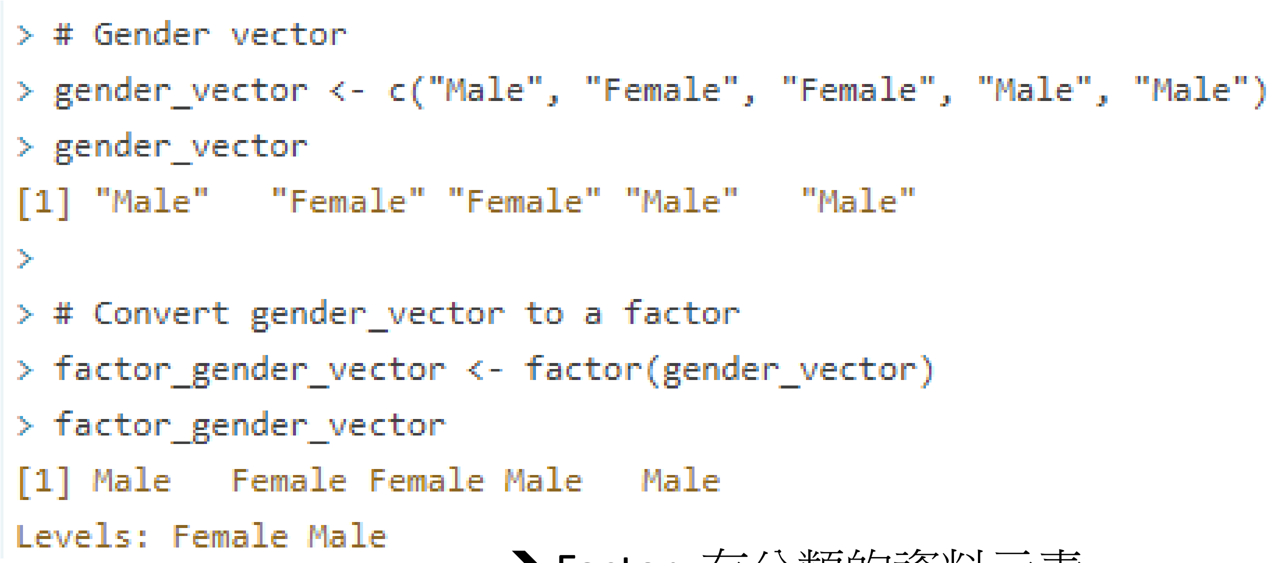


```
> all_wars_matrix
                           US non-US
A New Hope              461.0  314.4
The Empire Strikes Back 290.5  247.9
Return of the Jedi      309.3  165.8
The Phantom Menace      474.5  552.5
Attack of the Clones    310.7  338.7
Revenge of the Sith     380.3  468.5
> ticket_prices_matrix
                         US non-US
A New Hope              5.0    5.0
The Empire Strikes Back 6.0    6.0
Return of the Jedi      7.0    7.0
The Phantom Menace      4.0    4.0
Attack of the Clones    4.5    4.5
Revenge of the Sith     4.9    4.9
```

Factor

比較vector & factor 的不同

```
# Gender vector
gender_vector <- c("Male", "Female", "Female", "Male", "Male")
gender_vector

# Convert gender_vector to a factor
factor_gender_vector <- factor(gender_vector)
factor_gender_vector
```

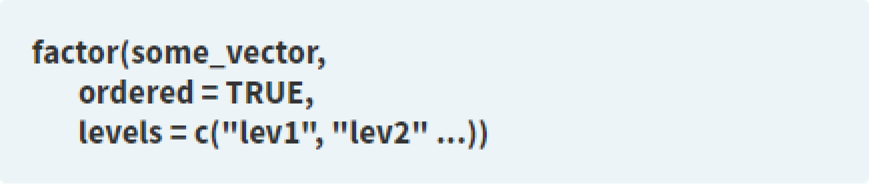


```
> # Gender vector
> gender_vector <- c("Male", "Female", "Female", "Male", "Male")
> gender_vector
[1] "Male"   "Female" "Female" "Male"   "Male"  
> 
> # Convert gender_vector to a factor
> factor_gender_vector <- factor(gender_vector)
> factor_gender_vector
[1] Male   Female Female Male   Male  
Levels: Female Male
```

➔ Factor: 有分類的資料元素

➔ categorical variables

可自訂分類的排序



```
factor(some_vector,
    ordered = TRUE,
    levels = c("lev1", "lev2" ...))
```

```
# Temperature

temperature_vector <- c("High", "Low", "High","Low", "Medium")

factor_temperature_vector <- factor(temperature_vector, order = TRUE, levels = c("Low", "Medium", "High"))

factor_temperature_vector


temperature_vector2 <- c("好熱", "冷", "太熱", "好冷")

factor_temperature_vector2 <- factor(temperature_vector2, order = TRUE, levels = c("好冷", "冷", "好熱", "太熱"))

factor_temperature_vector2
```

定義完大小之後可以拿來比較

(Code 續上)

```
element1 <- factor_temperature_vector2[2]

element2 <- factor_temperature_vector2[4]

element1 > element2
```

代號 → 完整字串

```
survey_vector <- c("M", "F", "F", "M", "M")
factor_survey_vector <- factor(survey_vector)
factor_survey_vector


# Specify the levels of factor_survey_vector
levels(factor_survey_vector) <- c("Female", "Male")
factor_survey_vector
```

Summarizing: vector vs factor

```
# Build factor_survey_vector with clean levels
survey_vector <- c("M", "F", "F", "M", "M")
factor_survey_vector <- factor(survey_vector)
levels(factor_survey_vector) <- c("Female", "Male")
factor_survey_vector

# Generate summary for survey_vector （這是vector)
summary(survey_vector)

# Generate summary for factor_survey_vector （這是factor)
summary(factor_survey_vector)
```

5.

有五位小朋友資料如下表，

姓名	生肖	國語成績	數學成績
小明	豬	80	48
小華	狗	75	38
瑪莉	龍	90	56
安妮	兔	85	37
彼得	牛	84	17

(1) 請用一個 Matrix (studentData)儲存上表

(2) 請依照十二生肖次序(鼠、牛、虎、兔、…、豬) 排序五位小朋友 (列出名子即可)

(3) 請計算每人的國語、數學平均成績、在 studentData 資加一平均成績欄位

(4) 數學大加分(開根號*10)，請輸出加分後的 studentDataNew (平均成績也要調整)

(5) 多了一位轉學生 珍妮佛(虎) 國語成績 80, 數學成績 84; 請數學加分後，加到
studentDataNew 內

(6) 請計算全班的國語、數學平均成績