

Data Science in R Cheatsheet

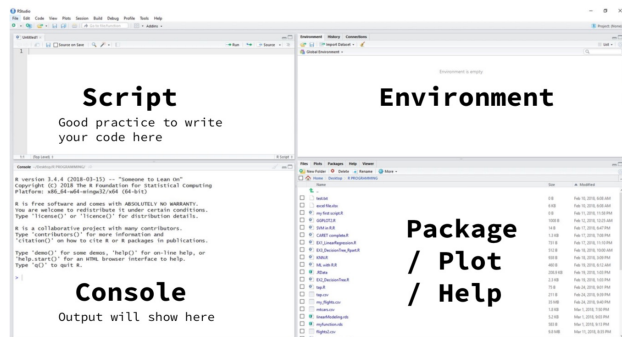
By: DataRockie & Data Science ชิลชิล

Version: 1.0

การใช้โปรแกรม RStudio



หน้าต่าง RStudio แบ่งเป็น 4 ส่วน

- 1) **Script** - พื้นที่สำหรับเขียนโค้ด
- 2) **Environment** - แสดงตัวแปรที่อยู่ในระบบ
- 3) **Console** - แสดงผลลัพธ์จากการรันโค้ด
- 4) **Files / Plots / Packages / Help** - ลิสต์ไฟล์ในโฟลเดอร์, พื้นที่แสดงผลสด, แพ็กเกจทั้งหมดในเครื่อง, อ่านคู่มือช่วยเหลือ



Working Directory

หากต้องการอ่านไฟล์ เช่น ไฟล์ข้อมูล จากในโฟลเดอร์ที่เราต้องการ ต้องเซตโฟลเดอร์นั้นเป็น Working Directory ก่อน โดย:

- 1) ที่แท็บ Files คลิกเข้าไปในโฟลเดอร์ที่ต้องการ หรือคลิกปุ่ม  ด้านบนขวาแล้วเลือกโฟลเดอร์
- 2) คลิก  More เลือก Set As Working Directory

วิธีการสั่งให้โค้ดทำงาน / รัน

โค้ดที่เราเขียนในหน้าต่าง Script สามารถสั่งรันโดย

การรันโค้ด 1 บรรทัด: คลิกบนบรรทัดที่ต้องการรัน (ไม่จำเป็นต้อง Highlight) แล้วกดปุ่ม Ctrl+Enter (Windows) หรือ Cmd+Enter (Mac)

การรันโค้ดทั้งไฟล์: กดปุ่ม Ctrl+Shift+Enter (Windows หรือ Cmd+Shift+Enter (Mac)

พื้นฐานตัวแปร

การคำนวณตัวเลข

```
x <- a + b # บวก
x <- a - b # ลบ
x <- a * b # คูณ
x <- a / b #หาร
x <- a ** b # a ยกกำลัง b
x <- a %% b # เศษจาก a หาร b
```

การสร้างตัวแปรแบบต่าง ๆ

```
x <- c(1,2,3) # สร้าง Vector
x <- list(1,2,3) # สร้าง List
x <- matrix(1:3, ncol=3)
# สร้าง Matrix
x <- data.frame(x = 1, y = 1:10)
# สร้าง DataFrame
```

เลือกคอลัมน์ที่ต้องการ จาก DataFrame

```
df[1:5] # เลือก 5 คอลัมน์แรก
df['col1'] # เลือกคอลัมน์ชื่อ col1
```

เลือกแถวที่ต้องการ จาก DataFrame

```
df[1:5, ] # เลือก 5 แถวแรก
df[ df['col1'] > 10, ]
# เลือกแถวที่ col1 มีค่ามากกว่า 10
```

ดึงค่าทั้งหมดจากคอลัมน์ col1 ออกมาเป็น Vector
df\$col1

การเขียนเงื่อนไข และลูป

เขียนเงื่อนไข if

```
if (เงื่อนไข) {
  # รันโค้ดส่วนนี้ ถ้าถูก
} else {
  # รันโค้ดส่วนนี้ ถ้าผิด
}
```

เขียนลูป while

```
while (เงื่อนไข) {
  # รันโค้ดส่วนนี้ จนกว่าเงื่อนไขจะผิด
}
```

การใช้ฟังก์ชัน

สร้างฟังก์ชัน

```
sum_two_numbers <- function(a,b) {
  return(a+b)
}
```

เรียกใช้ฟังก์ชัน

```
x <- sum_two_numbers(5, 10);
# x จะมีค่า 15
```

การใช้แพ็คเกจ

ติดตั้งแพ็คเกจ

```
install.packages("name")
```

เรียกใช้แพ็คเกจ

```
library(name)
```

ฟังก์ชันพื้นฐานของแพ็คเกจ Tidyverse และของ R

ดู 6 แถวแรกของข้อมูล

```
head(df)
```

ดู 6 แถวสุดท้ายของข้อมูล

```
tail(df)
```

ดูสรุปของชุดข้อมูลว่ามีกี่แถวกี่คอลัมน์ มีข้อมูลอะไร

```
glimpse(df)
```

ดูสรุปค่าทางสถิติของชุดข้อมูล

```
summary(df)
```

ดูสรุปค่าทั้งหมด และจำนวนแถวของแต่ละค่า ในคอลัมน์ที่เราต้องการ

```
table(df$col1)
```

คำนวณค่าเฉลี่ย

```
mean(df$col1)
```

คำนวณค่ามัธยฐาน

```
median(df$col1)
```

คำนวณค่าเบี่ยงเบนมาตรฐาน

```
sd(df$col1)
```

คำนวณผลรวม

```
sum(df$col1)
```

การ Clean ข้อมูล

พลอตจำนวนค่าที่หายไปในชุดข้อมูล (Package:

DataExplorer)

```
plot_missing(df)
```

ตรวจเช็คค่าที่หายไปทั้งหมด (Package: Tidyverse)

```
complete.cases(df)
```

ลบข้อมูลทุกแถวที่มีค่าที่หายไป (Package:

Tidyverse)

```
drop_na(df)
```

เซ็ต seed การสุ่ม เพื่อให้การสุ่มต่อจากบรรทัดนี้ทุกครั้งได้ผลเท่าเดิม (เปลี่ยน 123 เป็นเลขอื่นได้)

```
set.seed(123)
```

สุ่มข้อมูล Training Set **70%** และ Test Set 30%

```
index <- sample(1:nrow(df),  
0.7*nrow(df), replace=FALSE)  
train_df <- df[index, ]  
Test_df <- df[-index, ]
```

การแปรรูปข้อมูลด้วยแพ็คเกจ Dplyr

เลือกคอลัมน์ col1 และ col2

```
select(df, col1, col2)
```

เลือกคอลัมน์ที่ 1 ถึง 5

```
select(df, 1:5)
```

เลือกคอลัมน์ที่มีคำว่า text

```
select(df, contains("text"))
```

เลือกแถวที่ 1 ถึง 5

```
slice(df, 1:5)
```

เลือกแถวที่ค่า x และ y มากกว่า 10

```
filter(df, x > 10 & y > 10)
```

เลือกแถวที่ค่า x หรือ y มากกว่า 10

```
filter(df, x > 10 | y > 10)
```

เลือกแถวที่มีค่า x เป็น 1, 2, หรือ 3

```
filter(df, x %in% c(1,2,3))
```

เรียงข้อมูล โดยอิงค่าจากคอลัมน์ x น้อยไปมาก

```
arrange(df, x)
```

เรียงข้อมูล โดยอิงค่าจากคอลัมน์ x มากไปน้อย

```
arrange(df, desc(x))
```

สร้างคอลัมน์ชื่อ new ที่มีค่าเท่ากับ old ยกกำลัง 2

```
mutate(df, new = old * 2)
```

จับกลุ่มข้อมูลตามตัวแปร x แล้วสรุปข้อมูลออกมาเป็น 4 คอลัมน์: ค่าเฉลี่ยของ y, ค่า SD ของ y, ค่าสูงสุดของ y, ค่าต่ำสุดของ y

```
df %>%  
  group_by(x) %>%  
  summarise(avg_y = mean(y),  
            sd_y = sd(y),  
            max_y = max(y),  
            min_y = min(y))
```

Pipe operator (%>%) ช่วยเขียนโค้ดให้สั้นลง โดยส่งค่าจากคำสั่งด้านซ้ายของ pipe ไปเป็น Argument ของคำสั่งด้านขวาให้ทันที

```
df %>%  
  select(col1, col2, col3) %>%  
  filter(col1 > 20) %>%  
  arrange(col1) %>%  
  head(5)
```

การพลอตด้วยแพ็คเกจ ggplot2

พลอตชุดข้อมูล df แบบ Scatterplot ของตัวแปร x และ y

```
ggplot(data = df,  
mapping = aes(x, y)) + geom_point()
```

การสร้างโมเดล Linear Regression

สร้างโมเดลโดยใช้ทุกตัวแปร

```
model <- lm(y ~ . , data = train_df)
```

สร้างโมเดลโดยใช้ตัวแปร x1, x2, และ x3

```
model <- lm(y ~ x1 + x2 + x3,  
data = train_df)
```

ใช้โมเดลมาทำนายผลจากข้อมูลทดสอบ

```
predictions <- predict(model, test_df)
```

คำนวณค่า RMSE

```
sqrt(mean(  
(test_df$target - predictions)**2))
```

การสร้างโมเดล Neural Network (แพ็คเกจ nnet)

สร้างโมเดลโดยใช้ตัวแปร x1, x2, และ x3 และมี 4 unit ใน Hidden Layer

```
model <- nnet(y ~ x1 + x2 + x3,  
data = df, size = 4)
```

หากพบข้อมูลผิดพลาด หรือ อยากให้เพิ่มเติมส่วนไหน มาคุยกับเราได้ที่ <http://m.me/datarockie> และ <https://m.me/datasciencechill>