



tidyverse

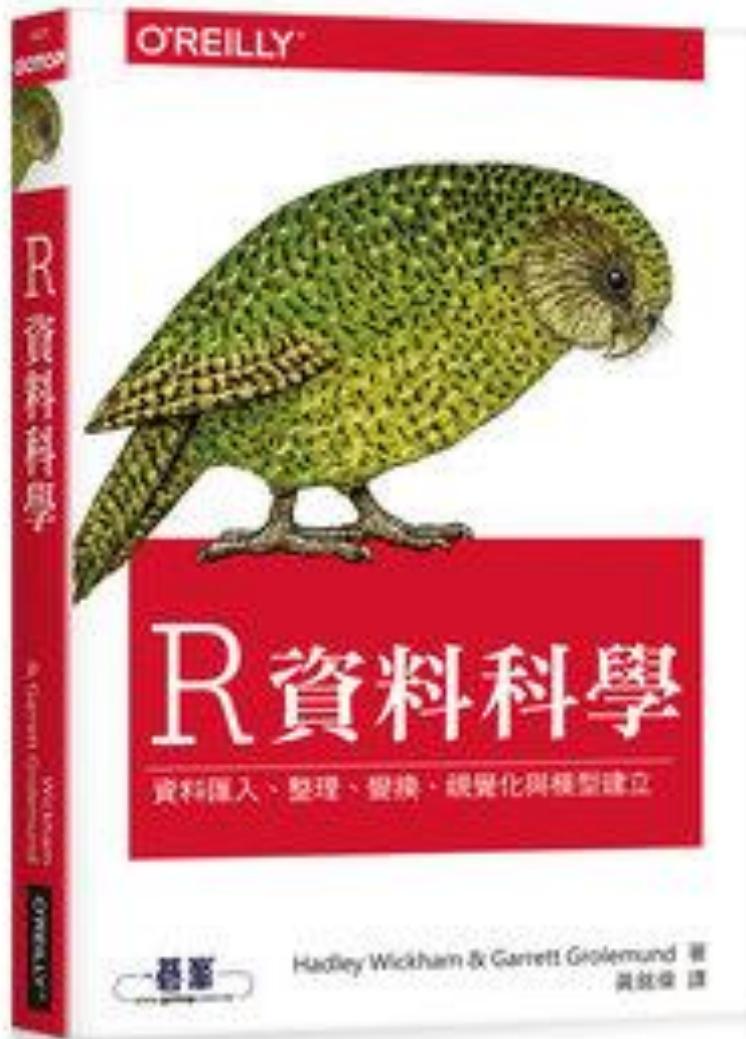
陳曾基

國立陽明交通大學醫學院醫務管理研究所
國立陽明交通大學醫學院急重症醫學研究所
國立陽明交通大學醫學院醫學系家庭醫學科
臺北榮民總醫院家庭醫學部
臺北榮民總醫院醫學研究部大數據中心

Topics

- R for data science
 - tidyverse
 - Function
 - Resources
-
- 本次上課請開始使用 RStudio 軟體操作 !!!
 - RStudio 裡執行 `view()` 但只顯示前 1000 筆
(row) 資料。當資料輸出量大時，請勿任意
執行 `view()`，否則會使 RStudio 近乎當掉。

Lecture Source :



SECTION I

R FOR DATA SCIENCE

Hadley Wickham

the man who revolutionized R



<https://commons.wikimedia.org/wiki/File:Hadley-wickham2016-02-04.jpg>

Institute of Mathematical Statistics

Fostering the development and dissemination of the theory and applications of statistics and probability

RENEW / JOIN IMS

JOURNALS & PUBLICATIONS

AWARDS & HONORS

MEETINGS

RESOURCES

LEADERSHIP

CONTACTS



COPSS Presidents' Award: Hadley Wickham

SEPTEMBER 2, 2019

Hadley Wickham wins the prestigious 2019 COPSS Presidents' Award



Hadley Wickham, Chief Scientist at RStudio, is the recipient of the 2019 COPSS Presidents' Award. This award is presented annually to a young member of one of the participating societies of COPSS in recognition of outstanding contributions to the profession of statistics. The award citation recognized Wickham "for influential work in statistical computing, visualization, graphics, and data analysis; for developing and implementing an impressively comprehensive computational infrastructure for data analysis through R software; for making statistical thinking and computing accessible to large audience; and for enhancing an appreciation for the important role of statistics among data scientists."

<https://imstat.org/2019/09/02/copss-presidents-award-hadley-wickham/>

Hadley Wickham

TEACHING

If you'd like to learn more about what I do, and how to use R effectively, I'd recommend starting with one of my books:

- [R for Data Science](#), with Garrett Grolemund, introduces the key tools for doing data science with R.
- [ggplot2: elegant graphics for data analysis](#) shows you how to use ggplot2 to create graphics that help you understand your data.
- [Advanced R](#) helps you master R as a programming language, teaching you what makes R tick.
- [R packages](#) teaches good software engineering practices for R, using packages for bundling, documenting, and testing your code.

I also teach in person workshops from time-to-time; see the [RStudio workshops page](#) for more details.

CODE

Most of my work is in the form of open source R code, which you can find on [my github](#). You can roughly divide my work into three categories: tools for data science, tools for data import, and software engineering tools.

DATA SCIENCE WITH THE TIDYVERSE

- [ggplot2](#) for visualising data.
- [dplyr](#) for manipulating data.
- [tidyverse](#) for tidying data.
- [stringr](#) for working with strings.
- [lubridate](#) for working with date/times.

DATA IMPORT

- [readr](#) for reading .csv and fwf files.
- [readxl](#) for reading .xls and .xlsx files.
- [haven](#) for SAS, SPSS, and Stata files.
- [httr](#) for talking to web APIs.
- [rvest](#) for scraping websites.
- [xml2](#) for importing XML files.

SOFTWARE ENGINEERING

- [devtools](#) for general package development.
- [roxygen2](#) for in-line documentation.
- [testthat](#) for unit testing
- [pkgdown](#) to create beautiful package websites

<https://r4ds.had.co.nz/>

The screenshot shows the homepage of the "R for Data Science" website. On the left, there's a sidebar with a navigation menu. The main content area has a title, authors' names, and a "Welcome" section with a detailed description of the book's purpose and contents. To the right of the main content is a graphic of the book cover.

Navigation Menu (Sidebar):

- Welcome
- 1 Introduction
- I Explore
- 2 Introduction
- 3 Data visualisation
- 4 Workflow: basics
- 5 Data transformation
- 6 Workflow: scripts
- 7 Exploratory Data Analysis
- 8 Workflow: projects
- II Wrangle
- 9 Introduction
- 10 Tibbles
- 11 Data import
- 12 Tidy data
- 13 Relational data
- 14 Strings
- 15 Factors
- 16 Dates and times
- III Program

Main Content Area:

R for Data Science

Garrett Grolemund
Hadley Wickham

Welcome

This is the website for "**R for Data Science**". This book will teach you how to do data science with R: You'll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you'll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You'll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You'll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualising, and exploring data.

Book Cover Graphic:

The book cover features a green parrot-like bird (a Kakapo) standing on a red background. The title "R for Data Science" is written in white on the red background, with the subtitle "VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA" below it. The authors' names, "Hadley Wickham & Garrett Grolemund", are at the bottom. The O'Reilly logo is at the top left of the book cover image.

Welcome

Acknowledgments

License

1 Introduction

I Explore

2 Introduction

3 Data visualisation

4 Workflow: basics

5 Data transformation

6 Workflow: scripts

7 Exploratory Data Analysis

8 Workflow: projects

II Wrangle

9 Introduction

10 Tibbles

11 Data import

12 Tidy data

13 Relational data

14 Strings

15 Factors

R for Data Science: Exercise Solutions

R for Data Science: Exercise Solutions

Jeffrey B. Arnold

February 27, 2019

Welcome

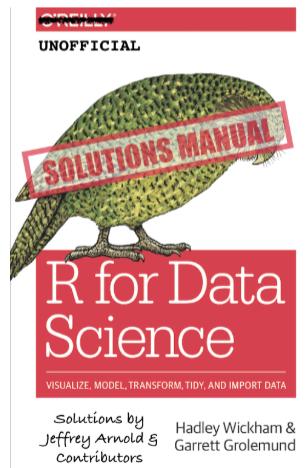
This book contains the **exercise solutions** for the book *R for Data Science*, by Hadley Wickham and Garret Grolemund (Wickham and Grolemund 2017).

R for Data Science itself is available online at r4ds.had.co.nz, and physical copy is published by O'Reilly Media and available from [amazon](#).

Acknowledgments

These solutions have benefited from many contributors. A special thanks to:

- Garrett Grolemund and Hadley Wickham for writing the truly fantastic *R for Data Science*, without whom these solutions would not exist—literally.



Advanced R

☰ ⌂ A 🔍 Advanced R

Welcome

License

Other books

Preface

1 Introduction

I Foundations

Introduction

2 Names and values

3 Vectors

4 Subsetting

5 Control flow

6 Functions

7 Environments

8 Conditions

II Functional programming

Introduction

9 Functionals

10 Function factories

Advanced R

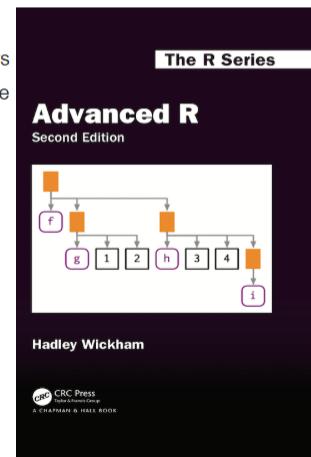
Hadley Wickham

Welcome

This is the website for 2nd edition of “**Advanced R**”, a book in Chapman & Hall’s R Series. The book is designed primarily for R users who want to improve their programming skills and understanding of the language. It should also be useful for programmers coming to R from other languages, as help you to understand why R works the way it does.

If you’re looking for the electronic version of the 1st edition, you can find it online at <http://adv-r.had.co.nz/>.

License



R packages

R packages

Preface

1 Introduction

- 1.1 Philosophy
- 1.2 In this book
- 1.3 Getting started
- 1.4 Acknowledgments
- 1.5 Conventions
- 1.6 Colophon

2 The Whole Game

- 2.1 Load devtools and friends
- 2.2 Toy package: foofactors
- 2.3 Peek at the finished product
- 2.4 `create_package()`
- 2.5 `use_git()`
- 2.6 Write the first function
- 2.7 `use_r()`
- 2.8 `load_all()`

R Packages

Hadley Wickham

Jenny Bryan

R packages

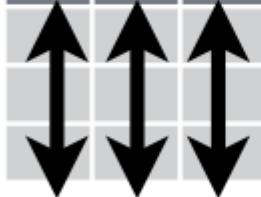
Packages are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data. In this book you'll learn how to turn your code into packages that others can easily download and use. Writing a package can seem overwhelming at first. So start with the basics and improve it over time. It doesn't matter if your first version isn't perfect as long as the next version is better.

This edition is a work in progress. If you're looking for the electronic version of the 1st edition, you can find it online at <http://r-pkgs.had.co.nz/>.

The image shows the front cover of the book 'R Packages' by Hadley Wickham and Jenny Bryan. The cover is white with a red band across the top that says 'OREILLY'. Below the band is a detailed black and white illustration of a bird, possibly a parrot or similar, in flight. At the bottom of the cover, there is a red rectangular area containing the title 'R Packages' in large white letters, followed by the subtitle 'ORGANIZE, TEST, DOCUMENT AND SHARE YOUR CODE' in smaller white letters.

A table is tidy if:

A	B	C



Each **variable** is in its own **column**

&

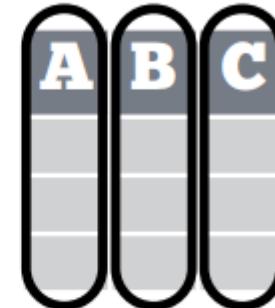
A	B	C



Each **observation**, or **case**, is in its own **row**

Tidy data:

A	B	C



Makes variables easy to access as vectors

<https://github.com/rstudio/cheatsheets/raw/master/data-import.pdf>

SECTION II

TIDYVERSE

Tidyverse

Packages Blog Learn Help Contribute

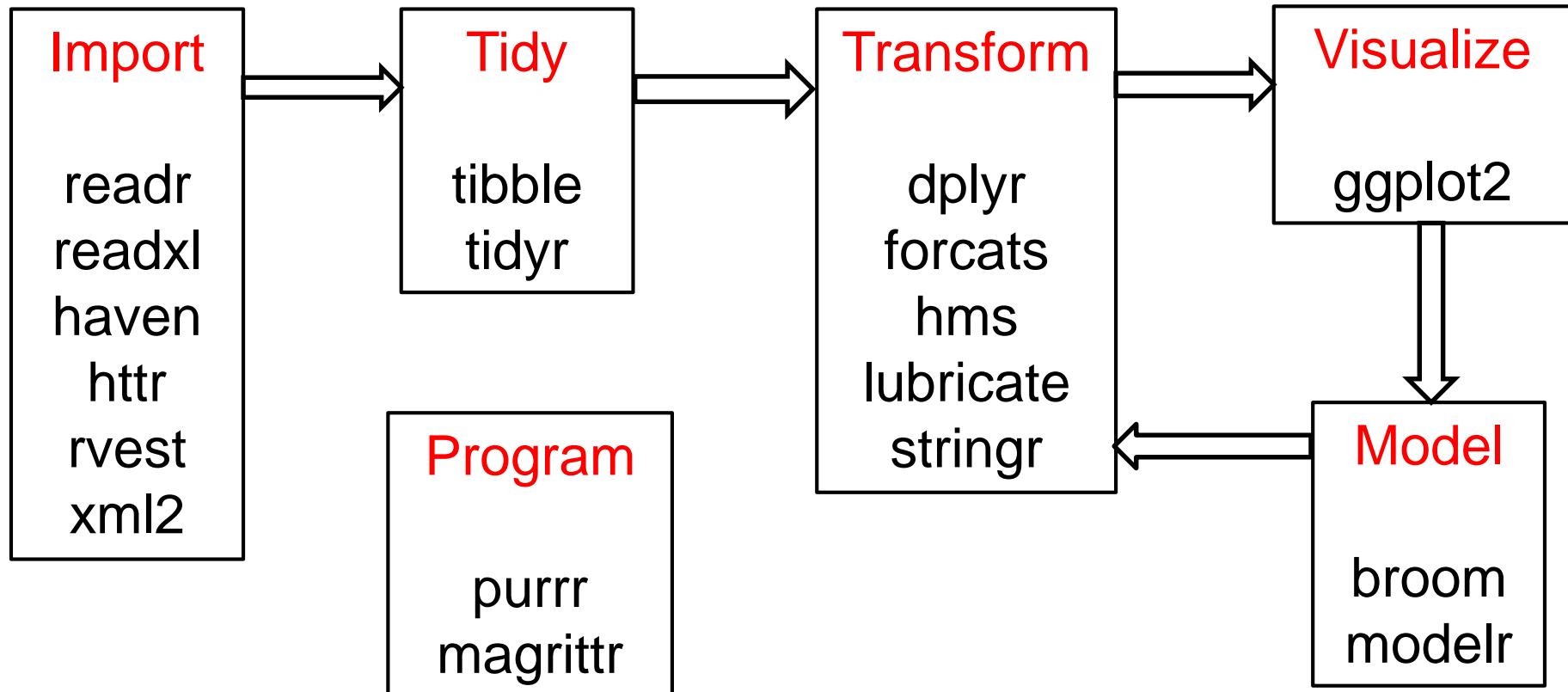
R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

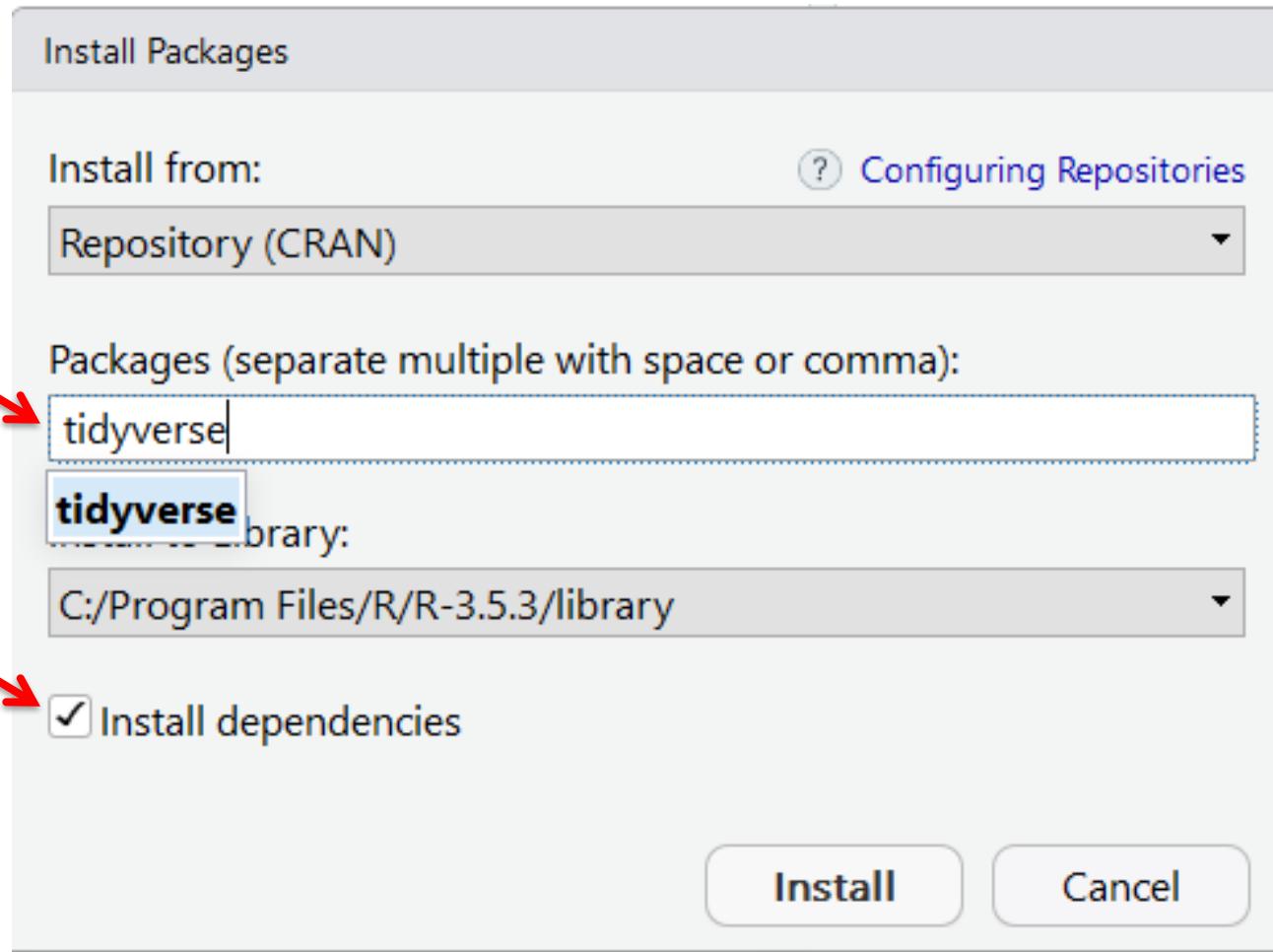
```
install.packages("tidyverse")
```

Flow of Data Processing



安裝 tidyverse 套件(工具包)

RStudio



也可 `install.packages("tidyverse")`

載入 tidyverse

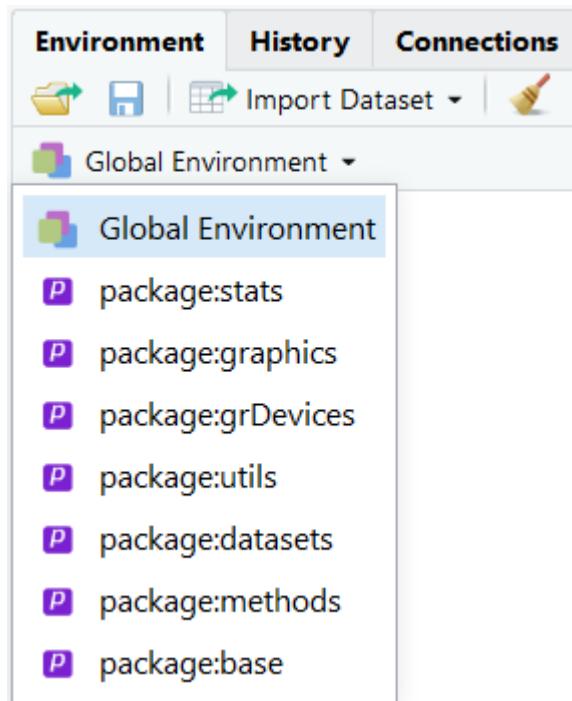
library(tidyverse)

- 會同時載入另外8個套件
- 顯示有2個函數會蓋過舊函數。如果還要使用舊函數，需加上舊套件名稱(即：舊套件::舊函數)

```
> library(tidyverse)
-- Attaching packages ----- tidyverse 1.3.0 --
✓ ggplot2 3.3.3     ✓ purrr   0.3.4
✓ tibble  3.1.0      ✓ dplyr   1.0.4
✓ tidyr   1.1.2      ✓ stringr 1.4.0
✓ readr   1.4.0      ✓ forcats 0.5.1
-- Conflicts ----- tidyverse_conflicts()
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

載入 tidyverse

- 載入前



- 載入後

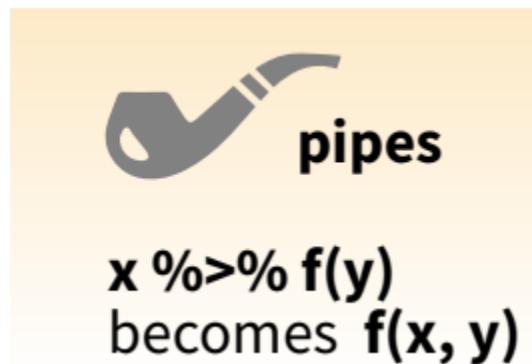
Global Environment
package:forcats
package:stringr
package:dplyr
package:purrr
package:readr
package:tidyverse
package:tibble
package:ggplot2
package:tidyverse
package:stats
package:graphics
package:grDevices
package:utils
package:datasets
package:methods
package:base

Most Important Features

- tibble : 簡化與優化傳統的data frame

```
> class(t)
[1] "spec_tb1_df"  "tb1_df"        "tb1"          "data.frame"
> mode(t)
[1] "list"
```

- pipe : %>%



Skepticism about tidyverse

从另一个视角看 R 语言的方言 Tidyverse

Norm Matloff

关键词：Base R; tidyverse

译者：李嵩；校对：任焱、黄湘云；编辑：任焱

从另一个视角看 R 语言的“方言”Tidyverse，以及 RStudio 对 Tidyverse 的提倡。

作者简介

作者 Norm Matloff 为 UC Davis 计算机科学教授（曾任 UCD 统计学教授）。中文翻译及投稿至 COS 经过作者同意。文中的“我”为作者视角，但译文中存在的任何不妥之处当然很可能是由译者引入的，还望读者不吝赐教。

<https://cosx.org/2020/10/alternative-view-tidyverse-r/>

<https://github.com/matloff/TidyverseSkeptic/blob/master/READMEFull.md>

SECTION II-A

DATA IMPORT



Overview

The goal of `readr` is to provide a fast and friendly way to read rectangular data (like `csv`, `tsv`, and `fwf`). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. If you are new to `readr`, the best place to start is the [data import chapter](#) in R for data science.

🔗 Installation

```
# The easiest way to get readr is to install the whole tidyverse:  
install.packages("tidyverse")  
  
# Alternatively, install just readr:  
install.packages("readr")  
  
# Or the the development version from GitHub:  
# install.packages("devtools")  
devtools::install_github("tidyverse/readr")
```

Links

Download from CRAN at
[https://cloud.r-project.org/
package=readr](https://cloud.r-project.org/package=readr)

Browse source code at
<https://github.com/tidyverse/readr>

Report a bug at
[https://github.com/tidyverse/readr/
issues](https://github.com/tidyverse/readr/issues)

Learn more at
[http://r4ds.had.co.nz/
data-import.html](http://r4ds.had.co.nz/data-import.html)

License

GPL (>= 2) | file [LICENSE](#)

Developers

[Hadley Wickham](#)

Author

[View GitHub profile](#)



🔗 Overview

The `readxl` package makes it easy to get data out of Excel and into R. Compared to many of the existing packages (e.g. `gdata`, `xlsx`, `xlsReadWrite`) `readxl` has no external dependencies, so it's easy to install and use on all operating systems. It is designed to work with *tabular* data.

`readxl` supports both the legacy `.xls` format and the modern xml-based `.xlsx` format. The `libxls` C library is used to support `.xls`, which abstracts away many of the complexities of the underlying binary format. To parse `.xlsx`, we use the [RapidXML](#) C++ library.

Installation

The easiest way to install the latest released version from CRAN is to install the whole tidyverse.

```
install.packages("tidyverse")
```

NOTE: you will still need to load `readxl` explicitly, because it is not a core tidyverse package loaded via
`library(tidyverse)`.

Links

Download from CRAN at
[https://cloud.r-project.org/
package=readxl](https://cloud.r-project.org/package=readxl)

Browse source code at
<https://github.com/tidyverse/readxl>

Report a bug at
[https://github.com/tidyverse/readxl/
issues](https://github.com/tidyverse/readxl/issues)

Learn more at
[http://r4ds.had.co.nz/
data-import.html](http://r4ds.had.co.nz/data-import.html)

License

[GPL-3](#)

Developers

[Hadley Wickham](#)
Author

[Jennifer Bryan](#)



part of the [tidyverse](#)
0.8.3

Tidy data Reference News



Overview

The goal of `tidyverse` is to help you create **tidy data**. Tidy data is data where:

1. Each variable is in a column.
2. Each observation is a row.
3. Each value is a cell.

Tidy data describes a standard way of storing data that is used wherever possible throughout the `tidyverse`. If you ensure that your data is tidy, you'll spend less time fighting with the tools and more time working on your analysis.

Installation

```
# The easiest way to get tidyverse is to install the whole tidyverse:  
install.packages("tidyverse")  
  
# Alternatively, install just tidyverse:  
install.packages("tidyverse")
```

Links

Download from CRAN at
[https://cloud.r-project.org/
package=tidyr](https://cloud.r-project.org/package=tidyr)

Browse source code at
<https://github.com/tidyverse/tidyr>

Report a bug at
[https://github.com/tidyverse/tidyr/
issues](https://github.com/tidyverse/tidyr/issues)

Learn more at
<http://r4ds.had.co.nz/tidy-data.html>

License

[Full license](#)

[MIT + file LICENSE](#)

Developers

[Hadley Wickham](#)

Author, maintainer

匯入示範資料

- 利用Import_tidyverse.r，匯入資料
- 將資料複製於D槽下，如果欲放置於其他位置，請同時更改Import_tidyverse.r
- 利用readr套件的**read_fwf**函數匯入
- 匯入速度相當快
- 不會將文字欄位轉換為因子(factor)
- # 資料與程式碼置於SampleData子目錄

注意 !!!

以**read_fwf**函數匯入的資料集，如果資料集某處並無資料(空白)，R 將其視為NA，而非長度為零的資料。與**read.csv**函數迥異

```
source("D:/SampleData/Import_tidyverse.r")  
# CHANGE HERE !!! 更動.r檔的位置
```

```

> columntypes <- "ccccccccccccccccccccccccnccnnncnnnc"
> cd <- read_fwf(
+   paste(directory, "CD2009.DAT", sep = ""),
+   fwf_widths(columnwidths, columnnames),
+   col_types = columntypes,
+   progress = TRUE
+ )
>
> cd
# A tibble: 598,574 x 37
# ... with 598,564 more rows, and 30 more variables:
#   CURE_ITEM_NO2 <chr>, CURE_ITEM_NO3 <chr>, CURE_ITEM_NO4 <chr>,
#   FUNC_TYPE <chr>, FUNC_DATE <chr>, TREAT_END_DATE <chr>,
#   ID_BIRTHDAY <chr>, ID <chr>, CARD_SEQ_NO <chr>, GAVE_KIND <chr>,
#   PART_NO <chr>, ACODE_ICD9_1 <chr>, ACODE_ICD9_2 <chr>,
#   ACODE_ICD9_3 <chr>, ICD_OP_CODE <chr>, DRUG_DAY <dbl>,
#   MED_TYPE <chr>, PRSN_ID <chr>, PHAR_ID <chr>, DRUG_AMT <dbl>,
#   TREAT_AMT <dbl>, TREAT_CODE <chr>, DIAG_AMT <dbl>, DSVC_NO <chr>,
#   DSVC_AMT <dbl>, BY_PASS_CODE <chr>, T_AMT <dbl>, PART_AMT <dbl>,
#   T_APPL_AMT <dbl>, ID_SEX <chr>
> |

```

tibble

注意 !!!
 以read_fwf函數
 匯入的資料集，
 如果資料集某處
 並無資料，R將
 其視為NA

Data	
cd	598574 obs. of 37 variables
\$ FEE_YM	: chr [1:598574] "200901" "200901" "200901" "200901" ...
\$ APPL_TYPE	: chr [1:598574] "1" "1" "1" "1" ...
\$ HOSP_ID	: chr [1:598574] "00000000000000000000000000000000174..."
\$ APPL_DATE	: chr [1:598574] "20090204" "20090212" "20090212" ...
\$ CASE_TYPE	: chr [1:598574] "01" "01" "01" "01" ...
\$ SEQ_NO	: num [1:598574] 3866 247 919 752 844 ...
\$ CURE_ITEM_N01	: chr [1:598574] NA NA NA NA ...
\$ CURE_ITEM_N02	: chr [1:598574] NA NA NA NA ...
\$ CURE_ITEM_N03	: chr [1:598574] NA NA NA NA ...
\$ CURE_ITEM_N04	: chr [1:598574] NA NA NA NA ...
\$ FUNC_TYPE	: chr [1:598574] "04" "11" "11" "11" ...
\$ FUNC_DATE	: chr [1:598574] "20090131" "20090105" "20090109" ...
\$ TREAT_END_DATE	: chr [1:598574] NA NA NA NA ...
\$ ID_BIRTHDAY	: chr [1:598574] "19720531" "19510203" "19510203" ...
\$ ID	: chr [1:598574] "0000000000000000000000000000035586..."
\$ CARD_SEQ_NO	: chr [1:598574] "0003" "0002" "0003" "0004" ...
\$ GAVE_KIND	: chr [1:598574] "4" "4" "4" "4" ...
\$ PART_NO	: chr [1:598574] "D10" "D10" "D10" "D10" ...
\$ ACODE_ICD9_1	: chr [1:598574] "4659" "3499" "3499" "5649" ...
\$ ACODE_ICD9_2	: chr [1:598574] NA NA NA NA ...
\$ ACODE_ICD9_3	: chr [1:598574] NA NA NA NA ...
\$ ICD_OP_CODE	: chr [1:598574] NA NA NA NA ...
\$ DRUG_DAY	: num [1:598574] 3 3 3 3 3 3 3 3 3 28 ...
\$ MED_TYPE	: chr [1:598574] "0" "0" "0" "0" ...
\$ PRSN_ID	: chr [1:598574] "0000000000000000000000000000040001..."
\$ PHAR_ID	: chr [1:598574] "0000000000000000000000000000040002..."
\$ DRUG_AMT	: num [1:598574] 75 75 75 75 75 75 75 75 14 14 516 ...
\$ TREAT_AMT	: num [1:598574] 0 0 0 0 0 0 950 360 0

讓空白還是空白

- 以read_fwf函數匯入的資料集，如果資料集某處並無資料(空白)，R將其視為NA(預設值)。想讓空白值匯入時維持空白，而非NA，需修改設定：

```
cd <- read_fwf(paste(directory, "CD2009.DAT", sep =  
""), fwf_widths(columnwidths, columnnames),  
col_types = columntypes, progress = TRUE, na = "NA")
```

```
# 預設值 : na = c("", "NA")
```

```
# na : Character vector of strings to interpret as missing  
values
```

重要觀念

匯入完畢後了解資料性質

```
mode(cd) # "list"  
class(cd) # "spec_tbl_df" "tbl_df" "tbl" "data.frame"
```

檔案匯入後同時具有 tibble 與 data frame 的特性

```
mode(cd$FEE_YM) # "character"
```

```
class(cd$FEE_YM) # "character"
```

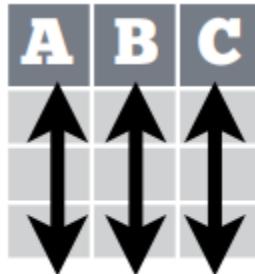
```
is.vector(cd$FEE_YM) # TRUE
```

```
is.factor(cd$FEE_YM) # FALSE
```

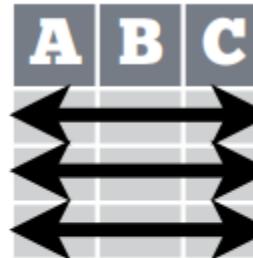
```
is.list(cd$FEE_YM) # FALSE
```

文字欄位沒被自動轉換為 factor

名詞翻譯



Each **variable** is in its own **column**



Each **observation**, or **case**, is in its own **row**

變數
欄位
直行

一筆資料
一筆觀察值
一筆紀錄(**record**)
橫列

SECTION II-B

TIBBLE



part of the [tidyverse](#)

2.1.1

[Intro](#) [Reference](#) [Articles ▾](#) [News](#)

Overview

A **tibble**, or `tbl_df`, is a modern reimagining of the `data.frame`, keeping what has proven to be effective, and throwing out what is not. Tibbles are `data.frames` that are lazy and surly: they do less (i.e. they don't change variable names or types, and don't do partial matching) and complain more (e.g. when a variable does not exist). This forces you to confront problems earlier, typically leading to cleaner, more expressive code. Tibbles also have an enhanced `print()` method which makes them easier to use with large datasets containing complex objects.

If you are new to tibbles, the best place to start is the [tibbles chapter](#) in *R for data science*.

Installation

```
# The easiest way to get tibble is to install the whole tidyverse:  
install.packages("tidyverse")  
  
# Alternatively, install just tibble:  
install.packages("tibble")
```

Links

Download from CRAN at
[https://cloud.r-project.org/
package=tibble](https://cloud.r-project.org/package=tibble)

Browse source code at
<https://github.com/tidyverse/tibble>

Report a bug at
[https://github.com/tidyverse/tibble/
issues](https://github.com/tidyverse/tibble/issues)

Learn more at
<http://r4ds.had.co.nz/tibbles.html>

License

[MIT + file LICENSE](#)

Developers

[Kirill Müller](#)
Author, maintainer
[Hadley Wickham](#)
Author

tibble

- 簡化與優化傳統的data frame
- 可用 `as_tibble(df)`，將既有的data frame轉成tibble
- 利用 `readr`套件匯入的資料集，自動轉成tibble資料型態，而且
 - 不會自動將字元向量轉成因子(factor)
 - 不會自動更改欄名稱(column name)
 - 不會使用列名稱(row name)
 - 匯入速度快很多
- 適用於data frame的函數，也可用於tibble (tibble 兼具data frame類別)

以 tibble 函數 建立 tibble

```
tibble(  
  x = 1:5,  
  y = 1,  
  z = x ^ 2 + y  
)
```

* tibble函數會自動循環延展個別向量的元素，讓tibble成為整齊的矩形(整齊的tidy)結構

```
> tibble(  
+   x = 1:5,  
+   y = 1,  
+   z = x ^ 2 + y  
+ )  
# A tibble: 5 × 3  
      x     y     z  
  <int> <dbl> <dbl>  
1     1     1     2  
2     2     1     5  
3     3     1    10  
4     4     1    17  
5     5     1    26  
> |
```

以 tribble 函數 建立 tibble

tribble(

~x, ~y, ~z,

#--|---|----

"a", 2, 3.6,

"b", 1, 8.5

)

```
> tribble(  
+   ~x, ~y, ~z,  
+   #--|---|----  
+   "a", 2, 3.6,  
+   "b", 1, 8.5  
+ )  
# A tibble: 2 × 3  
      x        y     z  
  <chr> <dbl> <dbl>  
1 a         2     3.6  
2 b         1     8.5  
> |
```

* transposed tibble

SECTION II-C

DATA MANIPULATION



Search...

Intro

Reference

Articles ▾

News ▾



Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation “by group”. You can learn more about them in `vignette("dplyr")`. As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in `vignette("two-table")`.

dplyr is designed to abstract over how the data is stored. That means as well as working with local data frames, you can also work with remote database tables, using exactly the same R code. Install the dbplyr package then read

```
vignette("databases", package = "dbplyr") .
```

<https://dplyr.tidyverse.org/articles/dplyr.html>

Links

Download from CRAN at
[https://cloud.r-project.org/
package=dplyr](https://cloud.r-project.org/package=dplyr)

Browse source code at
<https://github.com/tidyverse/dplyr>

Report a bug at
[https://github.com/tidyverse/dplyr/
issues](https://github.com/tidyverse/dplyr/issues)

Learn more at
<http://r4ds.had.co.nz/transform.html>

License

[Full license](#)

[MIT + file LICENSE](#)

Developers

[Hadley Wickham](#)

Author, maintainer



Overview

Strings are not glamorous, high-profile components of R, but they do play a big role in many data cleaning and preparation tasks. The stringr package provide a cohesive set of functions designed to make working with strings as easy as possible. If you're not familiar with strings, the best place to start is the [chapter on strings](#) in R for Data Science.

stringr is built on top of [stringi](#), which uses the [ICU](#) C library to provide fast, correct implementations of common string manipulations. stringr focusses on the most important and commonly used string manipulation functions whereas stringi provides a comprehensive set covering almost anything you can imagine. If you find that stringr is missing a function that you need, try looking in stringi. Both packages share similar conventions, so once you've mastered stringr, you should find stringi similarly easy to use.

Installation

```
# Install the released version from CRAN:  
install.packages("stringr")  
  
# Install the cutting edge development version from GitHub:
```

Links

Download from CRAN at
[https://cloud.r-project.org/
package=stringr](https://cloud.r-project.org/package=stringr)

Browse source code at
<https://github.com/tidyverse/stringr>

Report a bug at
[https://github.com/tidyverse/stringr/
issues](https://github.com/tidyverse/stringr/
issues)

Learn more at R4DS at
<http://r4ds.had.co.nz/strings.html>

License

[GPL-2](#) | [file LICENSE](#)

Developers

[Hadley Wickham](#)

Author, maintainer, copyright holder





part of the [tidyverse](#)
1.4.0.9000

Intro

RegEx

Reference

News ▾



Regular expressions

Source: vignettes/regular-expressions.Rmd

Regular expressions are a concise and flexible tool for describing patterns in strings. This vignette describes the key features of stringr's regular expressions, as implemented by `stringi`. It is not a tutorial, so if you're unfamiliar regular expressions, I'd recommend starting at <http://r4ds.had.co.nz/strings.html>. If you want to master the details, I'd recommend reading the classic *Mastering Regular Expressions* by Jeffrey E. F. Friedl.

Regular expressions are the default pattern engine in stringr. That means when you use a pattern matching function with a bare string, it's equivalent to wrapping it in a call to `regex()`:

```
# The regular call:  
str_extract(fruit, "nana")  
# Is shorthand for  
str_extract(fruit, regex("nana"))
```

Contents

- [Basic matches](#)
- [Escaping](#)
- [Special characters](#)
- [Matching multiple characters](#)
- [Alternation](#)
- [Grouping](#)
- [Anchors](#)
- [Repetition](#)
- [Look arounds](#)
- [Comments](#)



Overview

Date-time data can be frustrating to work with in R. R commands for date-times are generally unintuitive and change depending on the type of date-time object being used. Moreover, the methods we use with date-times must be robust to time zones, leap days, daylight savings times, and other time related quirks, and R lacks these capabilities in some situations. Lubridate makes it easier to do the things R does with date-times and possible to do the things R does not.

If you are new to lubridate, the best place to start is the [date and times chapter in R for data science](#).

Installation

```
# The easiest way to get lubridate is to install the whole tidyverse:  
install.packages("tidyverse")  
  
# Alternatively, install just lubridate:  
install.packages("lubridate")  
  
# Or the development version from GitHub:
```

Links

Download from CRAN at
[https://cloud.r-project.org/
package=lubridate](https://cloud.r-project.org/package=lubridate)

Browse source code at
[https://github.com/tidyverse/
lubridate](https://github.com/tidyverse/lubridate)

Report a bug at
[https://github.com/tidyverse/
lubridate/issues](https://github.com/tidyverse/lubridate/issues)

Learn more at
<http://r4ds.had.co.nz/dates-and-times.html>

License

GPL (>= 2)

Citation

[Citing lubridate](#)

重要觀念

重要觀念

重要觀念

dplyr 套件

- `select()`: 依名稱(column name(s))**挑出變數(欄位)**
- `filter()`: 依觀察值(value)**挑出資料(rows)**
- `arrange()`: 依條件(欄內的值)**排序資料(rows)**
- `mutate()`: 依既有變數**創造新的變數(欄位)**
- `summarize()`: 從很多值產生單一的**摘要資訊**
- `group_by()`: 與上述函數併用，**分組運算**

select()

```
select(cd, FEE_YM, FUNC_TYPE)
```

```
> select(cd, FEE_YM, FUNC_TYPE)
# A tibble: 598,574 x 2
  FEE_YM FUNC_TYPE
  <chr>  <chr>
1 200901  04
2 200901  11
3 200901  11
4 200901  11
5 200901  11
6 200901  11
7 200901  04
8 200901  06
9 200901  06
10 200901 10
# ... with 598,564 more rows
```

- 假設已先利用 Import_tidyverse.r 匯入資料
- 從資料集裡挑出 FEE_YM 與 FUNC_TYPE 欄位
- 第一個參數為資料集變數
- 欲挑出的欄位從第二個參數起排列

```
view(select(cd, FEE_YM, FUNC_TYPE))
```

```
select(cd, FEE_YM, FUNC_TYPE) %>% view()
```

```
cd %>% select(FEE_YM, FUNC_TYPE) %>% view()
```

	FEE_YM	FUNC_TYPE
1	200901	04
2	200901	11
3	200901	11
4	200901	11
5	200901	11
6	200901	11
7	200901	04
8	200901	06
9	200901	06
10	200901	10
11	200901	02
12	200901	01
13	200901	02
14	200901	06
15	200901	02

Showing 1 to 16 of 598,574 entries, 2 total columns

select()

`select(cd, SPECIALTY = FUNC_TYPE)`

- 可在挑選欄位時順便更改欄位名稱(只是螢幕/運算的名稱改變，原始資料的欄位名稱維持不變)
- 單純更改欄位名稱可用 `rename` 函數，其他欄位則維持不動

`rename(cd, SPECIALTY = FUNC_TYPE)`

```
> rename(cd, SPECIALTY = FUNC_TYPE)
# A tibble: 598,574 x 37
  FEE_LYM APPL_TYPE HOSP_ID APPL_DATE CASE_TYPE SEQ_NO CURE_ITEM_NO1
  <chr>   <chr>    <chr>   <chr>    <chr>   <dbl>  <chr>
1 200901  1        000000~ 20090204 01       3866 NA
2 200901  1        000000~ 20090212 01       247  NA
3 200901  1        000000~ 20090212 01       919  NA
4 200901  1        000000~ 20090212 01       752  NA
5 200901  1        000000~ 20090212 01       844  NA
6 200901  1        000000~ 20090212 01       852  NA
7 200901  1        000000~ 20090209 01       6542 NA
8 200901  1        000000~ 20090207 09       389  D0
9 200901  1        000000~ 20090207 09       369  NA
10 200901 1        000000~ 20090211 08      3196  33
# ... with 598,564 more rows, and 30 more variables:
#   CURE_ITEM_NO2 <chr>, CURE_ITEM_NO3 <chr>, CURE_ITEM_NO4 <chr>,
#   SPECIALTY <chr>, FUNC_DATE <chr>, TREAT_END_DATE <chr>,
#   ID_BIRTHDAY <chr>, ID <chr>, CARD_SEQ_NO <chr>, GAVE_KIND <chr>,
#   PART_NO <chr>, ACODE_ICD9_1 <chr>, ACODE_ICD9_2 <chr>,
#   ACODE_ICD9_3 <chr>, ICD_OP_CODE <chr>, DRUG_DAY <dbl>,
#   MED_TYPE <chr>, PRSN_ID <chr>, PHAR_ID <chr>, DRUG_AMT <dbl>,
#   TREAT_AMT <dbl>, TREAT_CODE <chr>, DIAG_AMT <dbl>, DSVC_NO <chr>,
#   DSVC_AMT <dbl>, BY_PASS_CODE <chr>, T_AMT <dbl>, PART_AMT <dbl>,
#   T_APPL_AMT <dbl>, ID_SEX <chr>
```

```
> select(cd, SPECIALTY = FUNC_TYPE)
# A tibble: 598,574 x 1
  SPECIALTY
  <chr>
1 04
2 11
3 11
4 11
5 11
6 11
7 04
8 06
9 06
10 10
# ... with 598,564 more rows
```

filter()

```
select(cd, FEE_YM, FUNC_TYPE) %>%  
  filter(FUNC_TYPE == "01")
```

```
> select(cd, FEE_YM, FUNC_TYPE) %>%  
+   filter(FUNC_TYPE == "01")  
# A tibble: 94,485 x 2  
  FEE_YM FUNC_TYPE  
  <chr>  <chr>  
1 200901 01  
2 200901 01  
3 200901 01  
4 200901 01  
5 200901 01  
6 200901 01  
7 200901 01  
8 200901 01  
9 200901 01  
10 200901 01  
# ... with 94,475 more rows
```

- %>% (pipe) : 從目前的結果資料 繼續往下操作
- %>% 不宜放在隔行的開頭，R會誤判只執行到前行尾
- 挑出科別(FUNC_TYPE)為家庭醫學科的資料
- 原本filter函數第一個參數應為資料集變數，因沿用上方的資料，故省略
- 可列多個篩選條件，以逗點分開
- 程式寫法次序也可對調 (先篩後挑)

```
filter(cd, FUNC_TYPE == "01") %>% select(FEE_YM, FUNC_TYPE)
```

arrange()

```
select(cd, FEE_YM, FUNC_TYPE) %>%
  filter(FUNC_TYPE == "01") %>%
  arrange(desc(FEE_YM))
```

- 依費用年月(FEE_YM)反向(desc)排序資料
- 預設為遞增排序
- 原本arrange函數第一個參數應為資料集變數，因沿用上方的資料，故省略
- 可列多個排序條件，在函數內以逗點分開

```
arrange(cd, desc(FEE_YM))
```

```
> select(cd, FEE_YM, FUNC_TYPE) %>%
+   filter(FUNC_TYPE == "01") %>%
+   arrange(desc(FEE_YM))
# A tibble: 94,485 x 2
  FEE_YM FUNC_TYPE
  <chr>  <chr>
1 200912 01
2 200912 01
3 200912 01
4 200912 01
5 200912 01
6 200912 01
7 200912 01
8 200912 01
9 200912 01
10 200912 01
# ... with 94,475 more rows
```

```
> arrange(cd, desc(FEE_YM))
# A tibble: 598,574 x 37
  FEE_YM APPL_TYPE HOSP_ID APPL_DATE CASE_TYPE SEQ_NO CURE_ITEM_NO1
  <chr>  <chr>    <chr>  <chr>      <chr>    <dbl>  <chr>
1 200912 1        000000~ 20100102 04        156 01
2 200912 1        000000~ 20100120 09        3359 NA
3 200912 1        000000~ 20100106 92        191 NA
```

mutate()

```
select(cd, FEE_YM, FUNC_TYPE) %>%
  filter(FUNC_TYPE == "01") %>%
  arrange(desc(FEE_YM)) %>%
  mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = ""))
```

```
> select(cd, FEE_YM, FUNC_TYPE) %>%
+   filter(FUNC_TYPE == "01") %>%
+   arrange(desc(FEE_YM)) %>%
+   mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = ""))
# A tibble: 94,485 x 3
  FEE_YM FUNC_TYPE YM_SPEC
  <chr>  <chr>    <chr>
1 200912 01        20091201
2 200912 01        20091201
3 200912 01        20091201
4 200912 01        20091201
5 200912 01        20091201
6 200912 01        20091201
7 200912 01        20091201
8 200912 01        20091201
9 200912 01        20091201
10 200912 01       20091201
# ... with 94,475 more rows
```

- 新增欄位：多由其他欄位經函數轉換而得
- 原本mutate函數第一個參數應為資料集變數，因沿用上方的資料，故省略
- 可新增多個欄位，在函數內以逗點分開
- 只是運算時增加欄位，原始的cd內容並無變化

```
mutate(cd, YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = ""))
```

summarize()

```
select(cd, FEE_YM, FUNC_TYPE) %>%
  filter(FUNC_TYPE == "01") %>%
  arrange(desc(FEE_YM)) %>%
  mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = "")) %>%
  summarize(monthcount = n_distinct(FEE_YM))
```

```
> select(cd, FEE_YM, FUNC_TYPE) %>%
+   filter(FUNC_TYPE == "01") %>%
+   arrange(desc(FEE_YM)) %>%
+   mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = ""))
+   summarize(monthcount = n_distinct(FEE_YM))
# A tibble: 1 × 1
  monthcount
  <int>
1       12
```

- 產生單一的摘要資訊：多係利用函數轉換而得
- 原本summarize函數第一個參數應為資料集變數，因沿用上方的資料，故省略
- 可同時計算多個摘要資訊，在函數內以逗點分開

```
summarize(cd, n_distinct(FEE_YM))
```

```
> summarize(cd, n_distinct(FEE_YM))
# A tibble: 1 × 1
  `n_distinct(FEE_YM)`
  <int>
1       12
```

summarize()

- 共有多少筆資料(row count)

```
summarize(cd, visitcount = n())
```

- 有多少個不同的年月份

```
summarize(cd, monthcount = n_distinct(FEE_YM))
```

- 每次就診平均醫療費用

```
summarize(cd, avgmoneypervisit = mean(T_AMT))
```

- 一次就診花費最多的費用

```
summarize(cd, max(T_AMT)) # 顯示欄位名稱 max(T_AMT)
```

- 有哪些不同的年月份

```
view(distinct(cd, FEE_YM)) # 等於unique(cd$FEE_YM)
```

NA - 1

CURE_ITEM_NO1
特定治療項目代號（一）

```
summarize(cd, n_distinct(CURE_ITEM_NO1)) # w/ NA
```

```
summarize(cd, n_distinct(CURE_ITEM_NO1, na.rm = TRUE)) # w/o NA
```

```
summarize(cd, sum(is.na(CURE_ITEM_NO1))) # count of NA
```

```
summarize(cd, sum(!is.na(CURE_ITEM_NO1))) # count of non-NA
```

```
distinct(cd, CURE_ITEM_NO1) # w/ NA
```

```
drop_na(cd, CURE_ITEM_NO1) %>%
```

```
distinct(CURE_ITEM_NO1) # w/o NA
```

```
filter(cd, !is.na(CURE_ITEM_NO1)) %>%
```

```
distinct(CURE_ITEM_NO1) # w/o NA
```

NA - 2

```
select(cd, CURE_ITEM_NO1) %>%
```

```
  drop_na() %>%
```

```
  summarize(n()) # w/o NA
```

```
select(cd, CURE_ITEM_NO1) %>%
```

```
  summarize(n()) # w/ NA
```

```
view(na.omit(cd$CURE_ITEM_NO1)) # w/o NA
```

```
length(na.omit(cd$CURE_ITEM_NO1)) # w/o NA
```

```
view(unique(cd$CURE_ITEM_NO1)) # w/ NA
```

```
view(unique(na.omit(cd$CURE_ITEM_NO1))) # w/o NA
```

```
view(table(cd$CURE_ITEM_NO1)) # w/o NA
```

```
view(table(cd$CURE_ITEM_NO1, useNA = "always")) # w/ NA
```

group_by()

```
select(cd, FEE_YM, FUNC_TYPE) %>%
  filter(FUNC_TYPE == "01") %>%
  arrange(desc(FEE_YM)) %>%
  mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = "")) %>%
  group_by(FEE_YM) %>%
  summarize(visitcountbymonth = n())
```

```
> select(cd, FEE_YM, FUNC_TYPE) %>%
+   filter(FUNC_TYPE == "01") %>%
+   arrange(desc(FEE_YM)) %>%
+   mutate(YM_SPEC = paste(FEE_YM, FUNC_TYPE, sep = "")) %>%
+   group_by(FEE_YM) %>%
+   summarize(visitcountbymonth = n())
# A tibble: 12 x 2
  FEE_YM visitcountbymonth
  <chr>          <int>
1 200901            8190
2 200902            7562
3 200903            8151
4 200904            8075
5 200905            7354
6 200906            6685
7 200907            6930
8 200908            6831
9 200909            7350
10 200910            9096
11 200911            8574
12 200912            9687
```

- 先分組，再進行後續運算，分組呈現結果
- 上例只是配合前面例子延伸，實際可簡化為：

```
filter(cd, FUNC_TYPE == "01") %>%
  group_by(FEE_YM) %>%
  summarize(n())
```

group_by()

group_by(cd, FEE_YM) %>%
summarize(visitcount = n())

```
> group_by(cd, FEE_YM) %>%  
+   summarize(visitcount = n())  
# A tibble: 12 x 2  
  FEE_YM visitcount  
  <chr>     <int>  
1 200901     47186  
2 200902     47593  
3 200903     51198  
4 200904     51205  
5 200905     48121  
6 200906     46594  
7 200907     48125  
8 200908     47162  
9 200909     49486  
10 200910    54604  
11 200911    52461  
12 200912    54839
```

- 先分組，再進行後續運算，分組呈現結果

group_by(cd, FEE_YM, ID_SEX) %>%
summarize(visitcount = n())

```
> group_by(cd, FEE_YM, ID_SEX) %>%  
+   summarize(visitcount = n())  
# A tibble: 24 x 3  
# Groups:   FEE_YM [12]  
  FEE_YM ID_SEX visitcount  
  <chr>  <chr>     <int>  
1 200901 F          25775  
2 200901 M          21411  
3 200902 F          26614  
4 200902 M          20979  
5 200903 F          28742  
6 200903 M          22456  
7 200904 F          28937  
8 200904 M          22268  
9 200905 F          27042  
10 200905 M         21079  
# ... with 14 more rows
```

- 可列多個分組依據，在函數內以逗點分開

group_by()

- 每人看診幾次(幾筆記錄)

```
group_by(cd, ID) %>% summarize(n())
```

- 每人每月看診幾次

```
group_by(cd, ID, FEE_YM) %>% summarize(n())
```

- 每人每季看診幾次

```
mutate(cd, Q = ifelse(FEE_YM %in% c("200901", "200902", "200903"), "Q1",
ifelse(FEE_YM %in% c("200904", "200905", "200906"), "Q2",
ifelse(FEE_YM %in% c("200907", "200908", "200909"), "Q3", "Q4")))) %>%
group_by(ID, Q) %>% summarize(n())
```

```
group_by(cd, ID, Q = ifelse(FEE_YM %in% c("200901", "200902", "200903"), "Q1",
ifelse(FEE_YM %in% c("200904", "200905", "200906"), "Q2",
ifelse(FEE_YM %in% c("200907", "200908", "200909"), "Q3", "Q4")))) %>%
summarize(VisitCount = n()) %>%
pivot_wider(names_from = Q, values_from = VisitCount)
```

group_by()

- 每個人看診幾次？依照看診次數排序

```
group_by(cd, ID) %>%  
  summarize(VisitCount = n()) %>%  
  arrange(desc(VisitCount))
```

- 只看主診斷碼的前三位數，各種診斷各有幾筆記錄？幾位病人？申報多少費用？請依費用多寡排序

```
group_by(cd, DX = substr(ACODE_ICD9_1, 1, 3)) %>%  
  summarize(VisitCount = n(), PatCount = n_distinct(ID),  
  TotalFee = sum(T_AMT)) %>%  
  arrange(desc(TotalFee)) %>%  
  view()
```

	DX	VisitCount	PatCount	TotalFee
1	585	1548	177	42535009
2	521	20947	10324	28636856
3	465	47431	15820	18916449
4	401	13148	2480	17860468
5	523	17671	11162	16269454
6	724	6808	3232	15859882
7	272	10298	1942	14470092
8	780	19008	6250	14028121
9	286	67	11	11992684
10	250	7799	1322	11945005
11	522	5473	3262	9849936
12	564	9972	3453	9772433
13	729	9911	3749	7399537
14	715	5567	1511	7375437
15	300	5738	1394	7293155

Showing 1 to 16 of 849 entries, 4 total columns

group_by()

- 各年齡層與性別，各有幾筆記錄？幾位病人？申報多少費用？

```
mutate(cd, AgeGroup =  
paste(as.integer(substr(ID_BIRTHDAY, 1, 4)) %% 10,  
'0s', sep = '')) %>%  
  
  group_by(AgeGroup, ID_SEX) %>%  
  
    summarize(VisitCount = n(), PatCount =  
n_distinct(ID), TotalFee = sum(T_AMT))
```

```
> mutate(cd, AgeGroup = paste(as.integer(substr(ID_BIRTHDAY, 1, 4)) %% 10,  
0, '0s', sep = '')) %>%  
+   group_by(AgeGroup, ID_SEX) %>%  
+   summarize(visitcount = n(), PatCount = n_distinct(ID), TotalFee = su  
m(T_AMT))  
# A tibble: 21 x 5  
# Groups:   AgeGroup [11]  
  AgeGroup ID_SEX VisitCount PatCount TotalFee  
  <chr>    <chr>     <int>     <int>      <dbl>  
1 1900s     F          23         2     14414  
2 1910s     F          1220        50    1453279  
3 1910s     M          1311        35    1661913  
4 1920s     F          12571       403   13692403  
5 1920s     M          16414       443   18914051  
6 1930s     F          30777       971   36269590  
7 1930s     M          26863       839   31209116  
8 1940s     F          38700      1381   46142980  
9 1940s     M          29971      1268   35322377  
10 1950s    F          53272      2553   51782555  
# ... with 11 more rows
```

%% : 回傳商數
% : 回傳餘數

group_by()

- 各年齡層與性別，各共申報多少費用？

```
mutate(cd, AgeGroup =  
paste(as.integer(substr(ID_BIRTHDAY, 1, 4)) %/% 10,  
'0s', sep = "")) %>%  
  group_by(AgeGroup, ID_SEX) %>%  
  summarize(TotalFee = sum(T_AMT)) %>%  
  pivot_wider(names_from = ID_SEX, values_from =  
TotalFee)
```

%/ % : 回傳商數
%% : 回傳餘數

# A tibble: 11 x 3	# Groups: AgeGroup [11]	M
AgeGroup	F	<dbl>
<chr>		
1 1900s	14414	NA
2 1910s	1453279	1661913
3 1920s	13692403	18914051
4 1930s	36269590	31209116
5 1940s	46142980	35322377
6 1950s	51782555	50998326
7 1960s	39141931	46099932
8 1970s	30921657	41494693
9 1980s	26017789	16358637
10 1990s	16448260	14969228
11 2000s	17483010	2283550

group_by()

- 在就診日為2009年的紀錄裡，每個人第一次看診日期？最後一次看診日期？相差多久？

```
filter(cd, substr FUNC_DATE, 1, 4) == '2009') %>%  
group_by(ID) %>%  
summarize( FirstDate = min(FUNC_DATE),  
           LastDate = max(FUNC_DATE),  
           as.Date(LastDate, '%Y%m%d') -  
           as.Date(FirstDate, '%Y%m%d')) )
```

```
> filter(cd, substr(FUNC_DATE, 1, 4) == '2009') %>%  
+   group_by(ID) %>%  
+   summarize(FirstDate = min(FUNC_DATE), LastDate = max(FUNC_DATE), as.  
Date(LastDate, '%Y%m%d') - as.Date(FirstDate, '%Y%m%d'))  
# A tibble: 36,081 x 4  
  ID    FirstDate      LastDate `as.Date(LastDate, "%Y%m%d") - a~  
  <chr>     <chr>       <chr>      <drtn>  
1 0000000000000000~ 20090211  20091107 269 days  
2 0000000000000000~ 20090125  20091226 335 days  
3 0000000000000000~ 20090115  20091219 338 days  
4 0000000000000000~ 20090218  20091113 268 days  
5 0000000000000000~ 20090523  20090523 0 days  
6 0000000000000000~ 20090130  20091209 313 days  
7 0000000000000000~ 20090110  20091211 335 days  
8 0000000000000000~ 20090206  20091109 276 days  
9 0000000000000000~ 20090223  20091223 303 days  
10 0000000000000000~ 20090731  20090803 3 days  
# ... with 36,071 more rows
```

group_by()

- 在門急診的用藥紀錄(oo)裡，有四種降血糖藥各開立幾次？各有幾次就診有開立？

```
filter(oo, DRUG_NO %in% c('B007152100', 'B022671100',  
'B020786100', 'B023206100')) %>%
```

```
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE,  
CASE_TYPE, SEQ_NO)) %>% # 前六個欄位用來連結cd檔
```

```
  group_by(DRUG_NO) %>%
```

```
  summarize(n(), n_distinct(FK))
```

B007152100 : GLUCOPHAGE

B022671100 : AMARYL

B020786100 : GLUCOBAY

B023206100 : ACTOS

```
> filter(oo, DRUG_NO %in% c('B007152100', 'B022671100', 'B020786100', 'B023206100')) %>%  
+   mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE,  
SEQ_NO)) %>% # 前六個欄位用來連結cd檔  
+   group_by(DRUG_NO) %>%  
+   summarize(n(), n_distinct(FK))  
# A tibble: 4 x 3  
  DRUG_NO     `n()` `n_distinct(FK)`  
  <chr>       <int>        <int>  
1 B007152100    671           671  
2 B020786100   1242          1242  
3 B022671100   2032          2027  
4 B023206100    539           539
```

縮減分組

```
count(cd, FUNC_TYPE) %>% view() # 47 entries
```

```
mutate(cd, SPECIALTY = fct_lump(FUNC_TYPE)) %>%  
  count(SPECIALTY) %>% view() # 45 entries (44 + Other)
```

```
mutate(cd, SPECIALTY = fct_lump(FUNC_TYPE, n = 8)) %>%  
  count(SPECIALTY) # 8 entries + Other
```

```
mutate(cd, SPECIALTY = fct_lump(FUNC_TYPE, n = 8)) %>%  
  count(SPECIALTY, sort = TRUE) # descending order by count
```

<https://dplyr.tidyverse.org/reference/tally.html>

https://forcats.tidyverse.org/reference/fct_lump.html

<https://r4ds.had.co.nz/factors.html#modifying-factor-order>

以grid格式顯示結果

```
z <- group_by(cd, FEE_YM) %>%
  summarize(visitcount = n())
```

- 將結果設為(儲存為)某變數，RStudio的Environment window即顯示該變數的摘要資訊，於其上點擊，Editor window即顯示以欄列方格呈現的結果，於其上點擊Show in new window按鈕，即能另開視窗呈現

`view(z)` #亦可以view函數指令顯示結果

```
group_by(cd, FEE_YM) %>%
  summarize(visitcount = n()) %>% view()
```

The screenshot shows the RStudio Environment window. It displays the following information:

	cd	oo	z
obs.	598574 obs. of 37 variables	2234235 obs. of 14 variables	12 obs. of 2 variables
FEE_YM	chr "200901" "200902" "200903" "200904"		
visitcount	int 47186 47593 51198 51205 48121 46		

The screenshot shows the RStudio Editor window displaying a grid of data. A red arrow points to the 'Filter' button at the top right of the grid header. The data grid contains the following rows:

	FEE_YM	visitcount
1	200901	47186
2	200902	47593
3	200903	51198
4	200904	51205
5	200905	48121
6	200906	46594
7	200907	48125
8	200908	47162
9	200909	49486
10	200910	54604
11	200911	52461
12	200912	54839

匯出資料

```
group_by(cd, FEE_YM) %>%  
  summarize(visitcount = n()) %>%  
  write_csv("D:/ZZZ.csv") # CHANGE HERE !!!
```

- 有多種設定，亦可轉換為多種格式
- 輸出檔案的字元編碼一般預設為 UTF-8

Exercise

- 門診就診檔(cd)內2009年1月1日就診紀錄中，依照病患生日排序後，取出前五筆紀錄

```
filter(cd, FUNC_DATE == '20090101') %>%
  arrange(ID_BIRTHDAY) %>%
  head(5) %>%
  view()
```

不能用 top_n 函數 (top_n係取每group的前幾筆)

Exercise

- 請計算門診就診檔(cd)中，主診斷為糖尿病(ICD-9-CM 250.x 或 ICD-10-CM E08-E13)的紀錄共有幾筆？有幾人？這些病人性別分布？

```
filter(cd, substr(ACODE_ICD9_1, 1, 3) == '250' | substr(ACODE_ICD9_1, 1, 3) == 'E08' | substr(ACODE_ICD9_1, 1, 3) == 'E09' | substr(ACODE_ICD9_1, 1, 3) == 'E10' | substr(ACODE_ICD9_1, 1, 3) == 'E11' | substr(ACODE_ICD9_1, 1, 3) == 'E12' | substr(ACODE_ICD9_1, 1, 3) == 'E13')
```

```
filter(cd, substr(ACODE_ICD9_1, 1, 3) == '250' | substr(ACODE_ICD9_1, 1, 3) == 'E08' | substr(ACODE_ICD9_1, 1, 3) == 'E09' | substr(ACODE_ICD9_1, 1, 3) == 'E10' | substr(ACODE_ICD9_1, 1, 3) == 'E11' | substr(ACODE_ICD9_1, 1, 3) == 'E12' | substr(ACODE_ICD9_1, 1, 3) == 'E13') %>% summarize(n_distinct(ID))
```

```
filter(cd, substr(ACODE_ICD9_1, 1, 3) == '250' | substr(ACODE_ICD9_1, 1, 3) == 'E08' | substr(ACODE_ICD9_1, 1, 3) == 'E09' | substr(ACODE_ICD9_1, 1, 3) == 'E10' | substr(ACODE_ICD9_1, 1, 3) == 'E11' | substr(ACODE_ICD9_1, 1, 3) == 'E12' | substr(ACODE_ICD9_1, 1, 3) == 'E13') %>%  
group_by(ID_SEX) %>%  
summarize(n_distinct(ID))
```

篩選條件的各種寫法

```
filter(cd, substr(ACODE_ICD9_1, 1, 3) == '250' | substr(ACODE_ICD9_1, 1, 3)
== 'E08' | substr(ACODE_ICD9_1, 1, 3) == 'E09' | substr(ACODE_ICD9_1, 1, 3)
== 'E10' | substr(ACODE_ICD9_1, 1, 3) == 'E11' | substr(ACODE_ICD9_1, 1, 3)
== 'E12' | substr(ACODE_ICD9_1, 1, 3) == 'E13')
```

```
filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08', 'E09', 'E10', 'E11',
'E12', 'E13'))
```

```
filter(cd, str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])"))
```

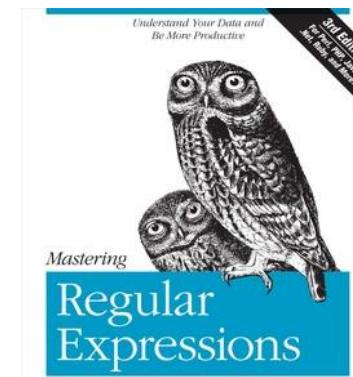
regex : regular expressions

^ : the start of the string

(|) : either ... or ...

[] : any one

[-] : any one from ... to ...



Exercise

- 請計算門診就診檔(cd)中，病人就診時的年齡

```
select(cd, ID_BIRTHDAY, FUNC_DATE) %>%
  mutate( Age = as.integer(as.Date(FUNC_DATE, '%Y%m%d') -
  as.Date(ID_BIRTHDAY, '%Y%m%d')) ) %/%
  365.25 )
```

FUNC_DATE與ID_BIRTHDAY為文字，需轉換為日期才能加減
如果ID_BIRTHDAY只提供年月，操作上加上日(01)：
paste(ID_BIRTHDAY, '01')
日期需明白轉換成數字(as.integer或as.numeric)後才能相除
%/%：整除，商數，去除餘數

一個函數套用在多個欄位

- 請計算門診就診檔(cd)中，每次就診的平均醫師診察費、藥費、檢查費、總醫療費用、部分負擔費用

```
cd %>%
```

```
  summarize( across(c(DIAG_AMT, DRUG_AMT, TREAT_AMT,  
T_AMT, PART_AMT), mean) )
```

```
# 比較一下
```

```
cd %>%
```

```
  summarize( mean(DIAG_AMT), mean(DRUG_AMT),  
mean(TREAT_AMT), mean(T_AMT), mean(PART_AMT) )
```

SECTION II-D

COMBINE TABLES

Relational Data

- `union()`
- `intersect()`
- `setdiff()`
- `inner_join() / full_join()`
- `left_join() / right_join()`
- `semi_join()`
- `anti_join()`

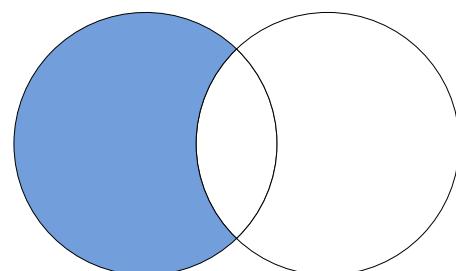
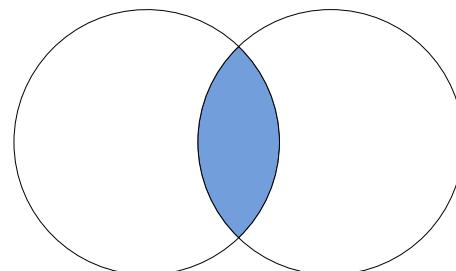
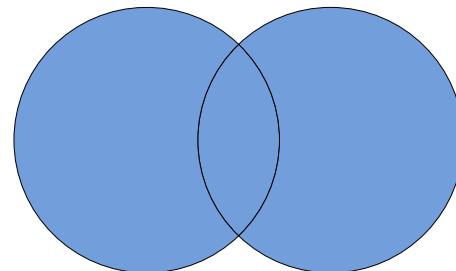
set operations

mutating joins

filtering joins

SET OPERATIONS

- **UNION** ($A \cup B$)
 - combining rows
- **INTERSECT** ($A \cap B$)
 - finding common rows
- **SETDIFF** ($A - B$)
 - finding different rows



Set Operations – 1

- 在門診就診檔(cd)裡，如果僅依主診斷碼來診斷三高(高血壓、高血糖、高血脂)：
 - 多少人至少有一高？
 - 多少人有三高(三種病)？
 - 多少人有糖尿病 而且 有高血壓？
 - 多少人有糖尿病 但是 沒有高血壓？

Set Operations – 1A-1

- 多少人至少有一高? (只看主診斷碼)

```
x <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '401' |  
substr(ACODE_ICD9_1, 1, 3) == 'I10')
```

```
y <- filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08',  
'E09', 'E10', 'E11', 'E12', 'E13'))
```

```
z <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '272' |  
substr(ACODE_ICD9_1, 1, 3) == 'E78')
```

```
union(x, y, z) %>% summarize(n_distinct(ID))
```

Set Operations – 1A-2

- 多少人至少有一高? (只看主診斷碼)

```
x <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '401' |  
substr(ACODE_ICD9_1, 1, 3) == 'I10') %>% select(ID)
```

```
y <- filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08',  
'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
z <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '272' |  
substr(ACODE_ICD9_1, 1, 3) == 'E78') %>% select(ID)
```

union 會
刪除重複值

count(**union**(x, y, z)) # x, y, z 只剩一個欄位 (較佳解法 以免搞混)

count(**union_all**(x, y, z)) # 意義：因三高來就診的總就診次數

71
union_all 會保留重複值

Set Operations – 1B

- 多少人有三高(一個人有三種病)? (只看主診斷碼)

```
x <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '401' |  
substr(ACODE_ICD9_1, 1, 3) == 'I10') %>% select(ID)
```

```
y <- filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08',  
'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
z <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '272' |  
substr(ACODE_ICD9_1, 1, 3) == 'E78') %>% select(ID)
```

intersect 會
刪除重複值

```
count(intersect(x, y, z)) # x, y, z 只剩一個欄位
```

理論上最好同時考慮主次診斷 !!!

Set Operations – 1C

- 多少人有糖尿病 而且 有高血壓? (只看主診斷碼)

```
x <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '401' |  
substr(ACODE_ICD9_1, 1, 3) == 'I10') %>% select(ID)
```

```
y <- filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08',  
'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
count(intersect(x, y))
```

理論上最好同時考慮主次診斷 !!!

Set Operations – 1D

- 多少人有糖尿病 但是 沒有高血壓? (只看主診斷碼)

```
x <- filter(cd, substr(ACODE_ICD9_1, 1, 3) == '401' |  
substr(ACODE_ICD9_1, 1, 3) == 'I10') %>% select(ID)
```

```
y <- filter(cd, substr(ACODE_ICD9_1, 1, 3) %in% c('250', 'E08',  
'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
count(setdiff(y, x))
```

rows that appear only in the first tibble, not in others

```
count(setdiff(x, y)) # 比較一下
```

理論上最好同時考慮主次診斷 !!!

Set Operations – 2A

- 在2009下半年方有糖尿病主診斷者(2009上半年無)？

```
x <- filter(cd, FEE_YM %in% c('200901', '200902', '200903',  
'200904', '200905', '200906'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
y <- filter(cd, FEE_YM %in% c('200907', '200908', '200909',  
'200910', '200911', '200912'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
setdiff(y, x)
```

```
# 找new cases，以計算incidence
```

Set Operations – 2B

- 在2009上半年有因糖尿病就診者，2009下半年亦有就診者？

```
x <- filter(cd, FEE_YM %in% c('200901', '200902', '200903',  
'200904', '200905', '200906'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
y <- filter(cd, FEE_YM %in% c('200907', '200908', '200909',  
'200910', '200911', '200912'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
intersect(x, y)
```

```
# 計算compliance
```

Set Operations – 2C

- 在2009上半年有因糖尿病就診者，2009下半年無因糖尿病就診者？

```
x <- filter(cd, FEE_YM %in% c('200901', '200902', '200903',  
'200904', '200905', '200906'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
y <- filter(cd, FEE_YM %in% c('200907', '200908', '200909',  
'200910', '200911', '200912'), substr(ACODE_ICD9_1, 1, 3) %in%  
c('250', 'E08', 'E09', 'E10', 'E11', 'E12', 'E13')) %>% select(ID)
```

```
setdiff(x, y)
```

```
# 計算compliance
```

Set Operations – 3

- 在cd檔裡，同一次就診時同時有開立血糖檢查[健保碼：09005C]與HBA1C檢查[健保碼：09006C]者，共有哪幾次？

```
x <- filter(oo, DRUG_NO %in% c('09005C')) %>%
  select(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)
```

```
y <- filter(oo, DRUG_NO %in% c('09006C')) %>%
  select(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)
```

```
z <- intersect(x, y)
```

intersect會
刪除重複值

⌚ oo	2234235 obs. of 14 variables
⌚ x	15886 obs. of 6 variables
⌚ y	4547 obs. of 6 variables
⌚ z	3862 obs. of 6 variables

JOIN





捷瑞有限公司

現金日記簿

101 年度

傳票 編號	收付日	總摘要	收付帳戶	支出總額	收入總額	科目 代號	科目名稱	明細摘要	支出明細	收入明細
6	1010001	101/01/02	公司設立	華南活存NT		1,000,000.00	306	股東出資	張大光	500,000.00
7							306	股東出資	陳中明	250,000.00
8							306	股東出資	李小亮	250,000.00
9										
10	1010002	101/01/03	簽約繳房租	華南活存NT	32,000.00		254	押金	房屋押金	20,000.00
11							203	租金支出	1月份租金	10,000.00
12							245	其他雜費	公證費	2,000.00
13										
14	1010003	101/01/03	買自用電腦一套	華南活存NT	33,000.00		251	電腦用品	桌上型電腦一套	25,000.00
15							251	電腦用品	OFFICE軟體一套	8,000.00
16										
17	1010004	101/01/03	提領零用金	華南活存NT	20,000.00		601	調撥轉出	提領零用金	20,000.00
18	1010004	101/01/03	提領零用金	零用現金NT		20,000.00	501	調撥轉入	提領零用金	20,000.00
19										
20	1010005	101/01/04	文具、粉刷、拜拜	零用現金NT	6,000.00		204	文具用品	文具一批	2,000.00
21							208	修繕費	粉刷辦公室	3,000.00
22							245	其他雜費	拜拜水果餅乾	1,000.00
23										
24	1010006	101/01/05	聯強-進10套電腦	華南活存NT	250,000.00		201	進貨	主機10台	200,000.00
25							201	進貨	螢幕10台	50,000.00

帳戶總覽 收支表 日記簿 帳戶明細 傳票 科目明細 說明



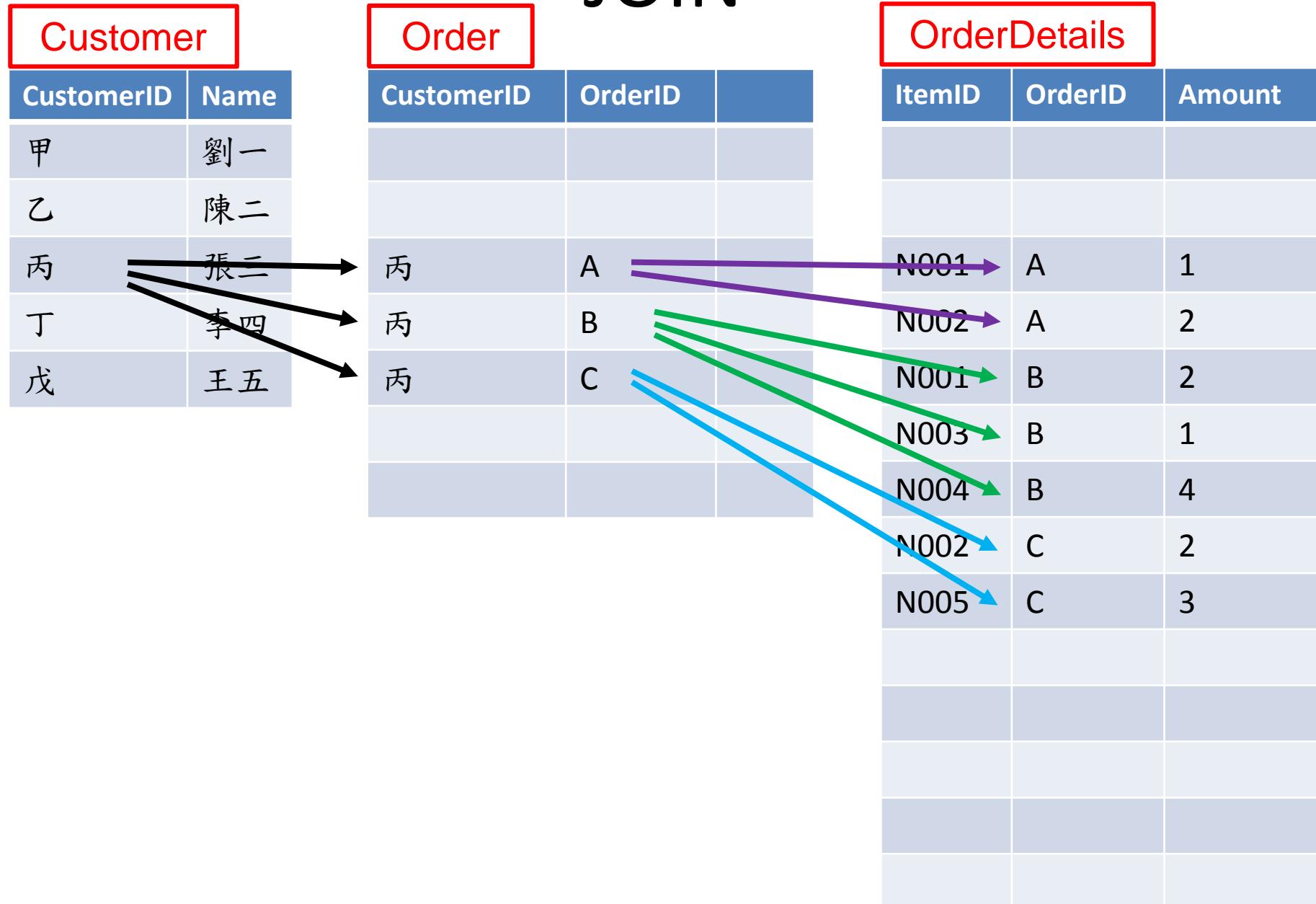
JOIN

Customer

CustomerID	Name	Tel.
甲	劉一	
乙	陳二	
丙	張二	
丁	李四	
戊	王五	

Order

JOIN



~ CD

心



全聯福利中心

電子發票證明聯

103年05-06月

HS-07431455

2014-05-07 21:21:00

隨機碼 4125

總計 350

賣方 28433582



店:025700 機:1 序:070407

TEL:02-27725870 延吉

IC卡:1999001711870402

退換貨請出示福利卡及發票含明細

全聯福利中心

全聯福利中心

全聯福利中心

全

~ 00

全聯福利中心

全聯福利中心

全聯福利中心

全聯福

[消費明細資料]

TEL: 02-27725870 延吉

店: 025700 機: 1 序: 070407

77 塑膠袋 10 公斤 1T

10 牛頭牌原味高湯

\$23 *2 46T

80 古早味梅子冰棒 39T

80 古早味梅子冰棒 39T

03 日本富士 #36 90

05 海藻蒟蒻麵 55T

97 大西瓜 89

合計: 359 紙電

現金: 350

紅利抵現: 9(-90點)

發票金額: 350

未稅: 163 稅額: 8

免稅: 179 本次: -81

卡餘額: 0 餘點: 226

IC卡: 1999001711870402

付現: 1000 找零: 650

退換貨請出示福利卡及發票



全聯福利中心

電子發票證明聯 103年05-06月

HS-07431455

2014-05-07 21:21:00

隨機碼 4125

總計 350

賣方 28433582



店:025700 機:1 序:070407

TEL:02-27725870 延吉

IC卡:1999001711870402

退換貨請出示福利卡及發票含明細

全聯
福利
中心

全聯
福利
中心

全聯
福利
中心

全聯
福利
中心

全
聯

* 明細資料可有多張

foreign
key

[消費明細資料]

TEL:02-27725870 延吉
店:025700 機:1 序:070407

77	塑膠袋 1.0 公斤	11
10	牛頭牌原味高湯	
\$23	*2	46T
80	古早味梅子冰棒	39T
80	古早味梅子冰棒	39T
03	日本富士 #36	90
05	海藻蒟蒻麵	55T
97	大西瓜	89
	合計:	359 紙電
	現金:	350
	紅利抵現:	9(-90點)
	發票金額:	350

未稅:	163	稅額:	8
免稅:	179	本次:	-81
卡餘額:	0	餘點:	226

IC卡:1999001711870402

付現: 1000 找零: 650

退換貨請出示福利卡及發票

全聯
福利
中心

全聯
福利
中心

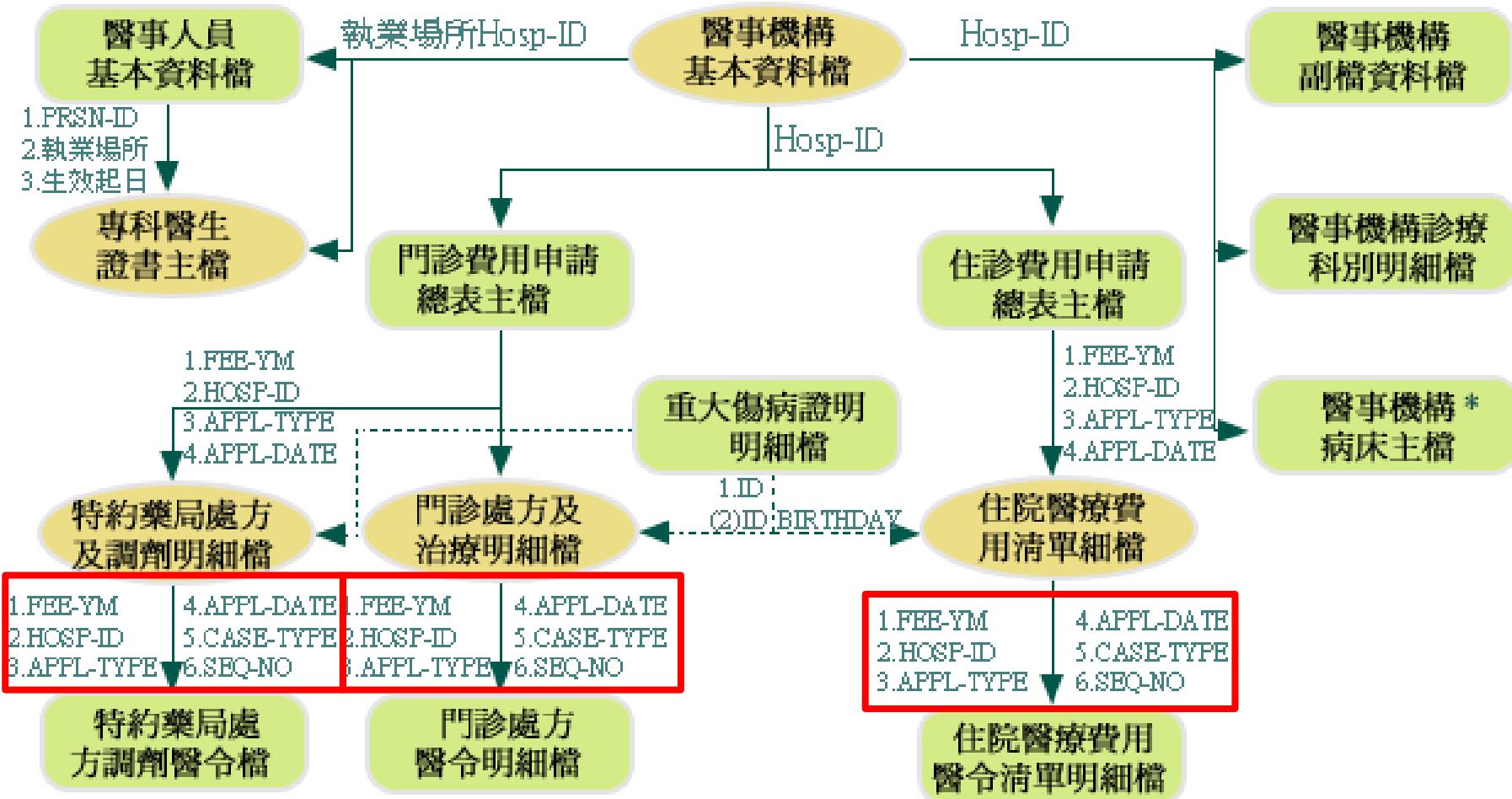
全聯
福利
中心

全聯
福

健保檔案關係

代碼	中文檔名	資料描述
CT	門診費用申請總表主檔	每家醫療院所一個月一筆記錄
CD	門診處方及治療明細檔	每次就診一筆記錄
OO	門診處方醫令明細檔	每一項開立藥品或檢驗項目一筆記錄
CT --(1-to-n)--> CD --(1-to-n)--> OO		
DT	住院費用申請總表主檔	每家醫療院所一個月一筆記錄
DD	住院醫療費用清單明細檔	每次住院(可分月申報)一筆記錄
DO	住院醫療費用醫令清單明細檔	每一項開立藥品或檢驗項目一筆記錄
DT --(1-to-n)--> DD --(1-to-n)--> DO		
GD	特約藥局處方及調劑明細檔	每張調劑處方箋一筆記錄
GO	特約藥局處方調劑醫令檔	每一項調劑藥品一筆記錄
CT --(1-to-n)--> GD --(1-to-n)--> GO		

各檔案間串檔變項說明



註:*須注意生效起訖日期

(2)可由ID+BIRTHDAY串檔

→ 各檔案間由所註明變項串檔可獲得對應資訊

→ 各檔案間可由所註明變項串檔,但未必獲得對應資料

Mutating Joins

Mutating joins combine variables from the two data.frames:

`inner_join()` return all rows from `x` where there are matching values in `y`, and all columns from `x` and `y`. If there are multiple matches between `x` and `y`, all combination of the matches are returned.

`left_join()` return all rows from `x`, and all columns from `x` and `y`. Rows in `x` with no match in `y` will have `NA` values in the new columns. If there are multiple matches between `x` and `y`, all combinations of the matches are returned.

`right_join()` return all rows from `y`, and all columns from `x` and `y`. Rows in `y` with no match in `x` will have `NA` values in the new columns. If there are multiple matches between `x` and `y`, all combinations of the matches are returned.

`full_join()` return all rows and all columns from both `x` and `y`. Where there are not matching values, returns `NA` for the one missing.

inner_join (1 : n)

Customer	
CustomerID	Name
甲	劉一
乙	陳二
丙	張三
丁	李四
戊	王五

Order		
CustomerID	OrderID	...
甲	A	
甲	A	
乙	B	
丙	A	
丙	B	
丙	C	
戊	B	
戊	D	
己	A	
己	B	

inner_join(Customer, Order,
by = "CustomerID")

CustomerID	Name	OrderID	...
甲	劉一	A	
甲	劉一	A	
乙	陳二	B	
丙	張三	A	
丙	張三	B	
丙	張三	C	
戊	王五	B	
戊	王五	D	

- 依據雙方對應欄位的相同值，擷取各種組合

inner_join (1 : 1)

```
DMdrug <- tribble(
```

假設藥名
沒有重複

```
  ~DRUG_ID, ~DRUG_NAME,  
  'B007152100', 'GLUCOPHAGE',  
  'B022671100', 'AMARYL',  
  'B020786100', 'GLUCOBAY',  
  'B023206100', 'ACTOS'
```

```
)
```

```
inner_join(oo, DMdrug, by = c("DRUG_NO" = "DRUG_ID")) %>%  
view()
```

```
inner_join(oo, DMdrug, by = c("DRUG_NO" = "DRUG_ID")) %>%  
select(DRUG_NO, DRUG_NAME) %>% view()
```

等號前方為第一個tibble
的連結欄位，後方為第
二個tibble的連結欄位。
如果兩者相同，可只寫
一個，例：
by = "DRUG_NO"

欄位前後
需加引號

注意！
已改名

inner_join (1 : 1)

```
DMdrug <- tribble(  
  ~DRUG_NO, ~DRUG_NAME,  
  'B007152100', 'GLUCOPHAGE',  
  'B022671100', 'AMARYL',  
  'B020786100', 'GLUCOBAY',  
  'B023206100', 'ACTOS'  
)
```

假設藥名
沒有重複

```
inner_join(oo, DMdrug, by = "DRUG_NO") %>%  
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%  
  group_by(DRUG_NAME) %>%  
  summarize(n(), n_distinct(FK))
```

與稍早的例子
互相比較一下

```
filter(oo, DRUG_NO %in% c('B007152100', 'B022671100', 'B020786100', 'B023206100')) %>%  
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%  
  group_by(DRUG_NO) %>%  
  summarize(n(), n_distinct(FK))
```

inner_join (1 : n)

1 entry

5 entries

```
inner_join(x, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",  
"APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>% view()
```

雖然cd檔只有一筆，連結oo檔後，cd檔部分顯示5次，而原oo檔前六個欄位則未出現

inner_join (1 : n)

```
x <- filter(cd, FUNC_TYPE == "01") # 限家醫科
```

cd	598574 obs. of 37 variables
oo	2234235 obs. of 14 variables
x	94485 obs. of 37 variables
y	307662 obs. of 45 variables
z	307662 obs. of 45 variables

```
y <- filter(cd, FUNC_TYPE == "01") %>%
```

```
  inner_join(oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE",
 "CASE_TYPE", "SEQ_NO")) # 家醫科的cd檔與oo檔inner_join
```

```
z <- inner_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE",
 "CASE_TYPE", "SEQ_NO")) %>%
```

```
  filter(FUNC_TYPE == "01") # 另一種寫法
```

```
filter(cd, FUNC_TYPE == "01") %>%
```

```
  inner_join(oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE",
 "CASE_TYPE", "SEQ_NO")) %>%
```

```
  filter(DRUG_NO %in% c('B007152100', 'B022671100', 'B020786100', 'B023206100'))
 %>%
```

```
  group_by(PRSN_ID) %>%
```

```
  summarize(n()) # 家醫科每一位醫師各曾開立幾筆糖尿病藥
```

left_join

Customer	
CustomerID	Name
甲	劉一
乙	陳二
丙	張三
丁	李四
戊	王五

Order			
CustomerID	OrderID	...	
甲	A		
甲	A		
乙	B		
丙	A		
丙	B		
丙	C		
戊	B		
戊	D		
己	A		
己	B		

left_join(Customer, Order,
by = "CustomerID")

CustomerID	Name	OrderID	...
甲	劉一	A	
甲	劉一	A	
乙	陳二	B	
丙	張三	A	
丙	張三	B	
丙	張三	C	
丁	李四	NA	←
戊	王五	B	
戊	王五	D	

- 依據雙方對應欄位的相同值，擷取各種組合，加上保留左方 tibble找不到對應者(其對應右方tibble的欄位值以NA替換)

left_join

```
x <- inner_join(cd, oo, by = c("FEE_YM", "APPL_TYPE",
  "HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO"))

y <- left_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
  "APPL_DATE", "CASE_TYPE", "SEQ_NO"))

z <- left_join(oo, cd, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
  "APPL_DATE", "CASE_TYPE", "SEQ_NO"))

inner_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
  "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>% summarize(n())

left_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
  "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>% summarize(n())

left_join(oo, cd, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
  "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>% summarize(n())
```

cd	598574 obs. of 37 variables
oo	2234235 obs. of 14 variables
x	2234235 obs. of 45 variables
y	2311630 obs. of 45 variables
z	2234235 obs. of 45 variables

left_join

```
mutate(cd, FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount = n_distinct(ID))
# 598574 598574 36109
```

```
mutate(oo, FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK))
# 2234235 521179 (oo檔裡無ID欄位)
```

```
inner_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>%
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount = n_distinct(ID))
# 2234235 521179 35735 (有些人僅看診未拿處方箋)
```

```
left_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>%
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount = n_distinct(ID))
# 2311630 598574 36109
```

```
full_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO")) %>%
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE, SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount = n_distinct(ID))
# 2311630 598574 36109
```

right_join

```
left_join(oo, cd, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE",
  "CASE_TYPE", "SEQ_NO")) %>%
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE,
    SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount =
    n_distinct(ID))
# 2234235 521179 35735
```

```
right_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE",
  "CASE_TYPE", "SEQ_NO")) %>%
  mutate(FK = paste(FEE_YM, APPL_TYPE, HOSP_ID, APPL_DATE, CASE_TYPE,
    SEQ_NO)) %>%
  summarize(recordcount = n(), visitcount = n_distinct(FK), IDcount =
    n_distinct(ID))
# 2234235 521179 35735
### right_join(x, y, by = ...) 等同 left(y, x, by = ...)
```

full_join

`Customer`

CustomerID	Name
甲	劉一
乙	陳二
乙	陳二
丙	張三
丁	李四
戊	王五

`Order`

CustomerID	OrderID	...
甲	A	
甲	A	
乙	B	
丙	A	
丙	B	
丙	C	
戊	B	
戊	D	
己	A	
己	B	

`full_join(Customer, Order, by = "CustomerID")`

CustomerID	Name	OrderID	...
甲	劉一	A	
甲	劉一	A	
乙	陳二	B	←
乙	陳二	B	←
丙	張三	A	
丙	張三	B	
丙	張三	C	
丁	李四	NA	←
戊	王五	B	
戊	王五	D	
己	NA	A	←
己	NA	B	←

- 依據雙方對應欄位的相同值，擷取各種組合，加上保留左方與右方tibble找不到對應者

Filtering Joins

Filtering joins keep cases from the left-hand data.frame:

`semi_join()` return all rows from `x` where there are matching values in `y`, keeping just columns from `x`.

A semi join differs from an inner join because an inner join will return one row of `x` for each matching row of `y`, where a semi join will never duplicate rows of `x`.

`anti_join()` return all rows from `x` where there are not matching values in `y`, keeping just columns from `x`.

- `semi_join(x, y, by = ...)` : 根據對應欄位(`by = ...`)，挑出x裡能與y對應的紀錄
- `anti_join(x, y, by = ...)` : 根據對應欄位(`by = ...`)，挑出x裡無法與y對應的紀錄

semi_join

Customer	
CustomerID	Name
甲	劉一
乙	陳二
丙	張三
丁	李四
戊	王五

Order		
CustomerID	OrderID	...
甲	A	
甲	A	
乙	B	
丙	A	
丙	B	
丙	C	
戊	B	
戊	D	
己	A	
己	B	

semi_join(Customer, Order,
by = "CustomerID")

CustomerID	Name
甲	劉一
乙	陳二
丙	張三
戊	王五

- 以第一個tibble (Customer)為主體，根據對應欄位(by = ...)，挑出第一個tibble裡能與第二個tibble (Order)對應的紀錄

semi_join

- 在門診檔裡，找出曾接受HBA1C檢查的就診(cd)記錄。

```
z <- filter(oo, DRUG_NO %in% c('09006C'))
```

```
# z : 有 HBA1C 的 oo 紀錄
```

```
semi_join(cd, z, by = c("FEE_YM", "APPL_TYPE",
"HOSP_ID", "APPL_DATE", "CASE_TYPE", "SEQ_NO"))
```

```
# 主體為 cd，挑出能與 oo 相 match 的 cd 紀錄
```

semi_join

- 在門診檔裡，就診時有開立HBA1C檢查，卻無糖尿病診斷碼，請列出其cd記錄。

```
z <- filter(oo, DRUG_NO %in% c('09006C')) # 4547
```

```
semi_join(cd, z, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE",  
"SEQ_NO")) # 4546 (其中有一次就診開立兩次HBA1C檢查)
```

```
semi_join(cd, z, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE",  
"SEQ_NO")) %>%  
  filter(str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])") | str_detect(ACODE_ICD9_2,  
"^(250|E0[89]|E1[0-3])" | str_detect(ACODE_ICD9_3, "^(250|E0[89]|E1[0-3])")) # 3671
```

```
semi_join(cd, z, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE",  
"SEQ_NO")) %>%  
  filter( !str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])" | is.na(ACODE_ICD9_1)),  
  !str_detect(ACODE_ICD9_2, "^(250|E0[89]|E1[0-3])" | is.na(ACODE_ICD9_2)),  
  (!str_detect(ACODE_ICD9_3, "^(250|E0[89]|E1[0-3])" | is.na(ACODE_ICD9_3)) )  
# 875 (= 4546 - 3671)
```

勿忘NA
108

anti_join

Customer

CustomerID	Name
甲	劉一
乙	陳二
丙	張三
丁	李四
戊	王五

Order

CustomerID	OrderID	...
甲	A	
甲	A	
乙	B	
丙	A	
丙	B	
丙	C	
戊	B	
戊	D	
己	A	
己	B	

anti_join(Customer, Order,
by = "CustomerID")

CustomerID	Name
丁	李四

- 以第一個tibble (Customer)為主體，根據對應欄位(by = ...)，挑出第一個tibble裡無法與第二個tibble (Order)對應的紀錄

anti_join

- 在門診檔裡，就診時有糖尿病診斷碼，卻無開立血糖檢查，請列出其cd記錄。

```
z <- filter(oo, DRUG_NO %in% c('09005C')) # 15886 項檢查
```

```
filter(cd, str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])") | str_detect(ACODE_ICD9_2,  
"^(250|E0[89]|E1[0-3])" | str_detect(ACODE_ICD9_3, "^(250|E0[89]|E1[0-3])")) # 24699 有診  
斷
```

```
filter(cd, str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])") | str_detect(ACODE_ICD9_2,  
"^(250|E0[89]|E1[0-3])" | str_detect(ACODE_ICD9_3, "^(250|E0[89]|E1[0-3])")) %>%  
semi_join(z, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE",  
"SEQ_NO")) # 7142 有診斷且有開立檢查
```

```
filter(cd, str_detect(ACODE_ICD9_1, "^(250|E0[89]|E1[0-3])") | str_detect(ACODE_ICD9_2,  
"^(250|E0[89]|E1[0-3])" | str_detect(ACODE_ICD9_3, "^(250|E0[89]|E1[0-3])")) %>%  
anti_join(z, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID", "APPL_DATE", "CASE_TYPE",  
"SEQ_NO"))
```

```
# 17557 (= 24699 - 7142) 有診斷但無開立檢查
```

Debugging

```
semi_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
"APPL_DATE", "CASE_TYPE", "SEQ_NO"))
```

521,179

```
anti_join(cd, oo, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
"APPL_DATE", "CASE_TYPE", "SEQ_NO"))
```

77,395 就診未開立處方箋

```
semi_join(oo, cd, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
"APPL_DATE", "CASE_TYPE", "SEQ_NO"))
```

2,234,235

```
anti_join(oo, cd, by = c("FEE_YM", "APPL_TYPE", "HOSP_ID",
"APPL_DATE", "CASE_TYPE", "SEQ_NO"))
```

0 開立處方項目卻無就診紀錄

Exercise

- 找出2009年耗用門診醫療費用最多的病人所有申報紀錄

```
x <- group_by(cd, ID) %>%
  summarize(TotalFee = sum(T_AMT)) %>%
  summarize(MaxFee = max(TotalFee)) %>% as.numeric()
```

依ID分組，分兩步找出最大值，將結果由tibble型態轉為數值型態

```
y <- group_by(cd, ID) %>%
  summarize(TotalFee = sum(T_AMT)) %>%
  filter(TotalFee == x)
```

找出個人總費用為最大值者(未必1人)

```
view(semi_join(cd, y, by = "ID"))
# 比對最大值的ID者，挑出cd裡的紀錄
```

SECTION III

FUNCTION

Recode in dplyr

```
cd %>%  
  mutate( GENDER = recode(ID_SEX, M = 1, F = 2, .default = 3) ) %>%  
  select(ID_SEX, GENDER) %>%  
  head(15) %>%  
  view()
```

case_when in dplyr

```
mutate(cd, SPECIALTY = case_when(  
  FUNC_TYPE == "00" ~ "不分科",  
  FUNC_TYPE == "01" ~ "家醫科",  
  ...  
  ...  
  ...  
  ...  
  TRUE ~ "未知"  
) %>%  
  select(FUNC_TYPE, SPECIALTY) %>%  
  head(15)
```

完整程式碼在
PPT備忘稿處

fct_recode inforcats

```
mutate(cd, SPECIALTYMAJOR = fct_recode(FUNC_TYPE,  
  "不分科" = "00",  
  "家醫科" = "01",  
  "家醫科" = "EA",  
  "老人醫學科" = "AK", ...  
  ...  
  ...  
  ...)) %>%  
count(FUNC_TYPE, SPECIALTYMAJOR) %>%  
view()
```

完整程式碼在
PPT備忘稿處

* 缺點：沒有others選項，當
是其他值時，直接列出原值

https://forcats.tidyverse.org/reference/fct_recode.html
<https://r4ds.had.co.nz/factors.html#modifying-factor-levels>

Function

Input

Output

- Scalar (element)
 - Vector
 - List
 - ...
- Scalar
 - Vector
 - List
 - ...

內建函數

```
x <- 1  
sum(x) # scalar -> scalar  
summary(x) # scalar -> scalarS  
sqrt(x) # scalar -> scalar
```

```
> x <- 1  
> sum(x)  
[1] 1  
> summary(x)  
   Min. 1st Qu. Median Mean 3rd Qu. Max.  
      1       1       1       1       1       1  
> sqrt(x)  
[1] 1
```

```
y <- 1:6  
sum(y) # vector -> scalar  
summary(y) # vector -> scalarS  
sqrt(y) # vector -> vector
```

```
> y <- 1:6  
> sum(y)  
[1] 21  
> summary(y)  
   Min. 1st Qu. Median Mean 3rd Qu. Max.  
     1.00    2.25    3.50    3.50    4.75    6.00  
> sqrt(y)  
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490
```

1 scalar -> 1 scalar

```
source("D:/SampleData/Specialty_Scalar.r")
```

請參閱函數的程式碼

```
# CHANGE HERE !!! 請更動放置Specialty_Scalar.r檔的位置
```

```
x <- "01"
```

```
specialty(x) # "家醫科" # R 4.0.4 版會產生 "\u5bb6\u91ab\u79d1"
```

```
view(specialty(x)) # "家醫科"
```

```
x <- c("01", "02")
```

```
specialty(x) # Error ... 必須是長度為 1 的向量
```

```
specialty(cd$FUNC_TYPE) # Error
```

R / RStudio 的資料顯示碰到中文時，常有亂碼的情形，不過只是顯示亂碼，內部與輸出仍正常運作。一般將結果匯出至檔案時，字元編碼預設為UTF-8。

vector -> vector

解決方法

```
y1 <- map(cd$FUNC_TYPE, specialty) # (tidyverse函數) 輸出 list  
y2 <- map_chr(cd$FUNC_TYPE, specialty) # 輸出 vector (character)  
y3 <- lapply(cd$FUNC_TYPE, specialty) # (r 內建函數) 輸出 list  
y4 <- sapply(cd$FUNC_TYPE, specialty) # (r 內建函數) 輸出 vector
```

```
y1 %>% head(10000) %>% as.data.frame() %>% write_delim("D:/y1.txt")  
y2 %>% head(10000) %>% as.data.frame() %>% write_delim("D:/y2.txt")  
y3 %>% head(10000) %>% as.data.frame() %>% write_delim("D:/y3.txt")  
y4 %>% head(10000) %>% as.data.frame() %>% write_delim("D:/y4.txt")  
# 為節省時間，只取前一萬筆資料  
# 需先轉換為 data frame 後才能輸出至檔案
```

vector -> vector

```
source("D:/SampleData/Specialty_Vector.r")
```

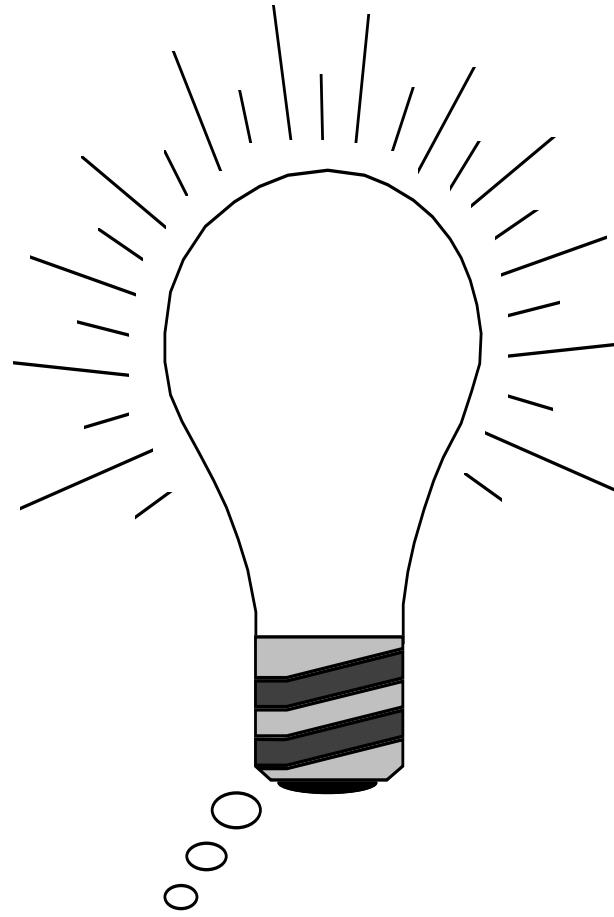
請參閱函數的程式碼

```
# CHANGE HERE !!! 更動.r檔的位置
```

```
x1 <- "01"  
view( specialty(x1) ) # "家醫科"  
x2 <- c("01", "02")  
view( specialty(x2) ) # "家醫科" "內科"  
specialty(cd$FUNC_TYPE) %>% head(15) %>% view()
```

```
mutate(cd, SPECIALTY = specialty(FUNC_TYPE), SPECIALTYMAJOR =  
specialty.major(FUNC_TYPE)) %>%  
select(FUNC_TYPE, SPECIALTY, SPECIALTYMAJOR) %>%  
head(100) %>% view()
```

Thanks for
Your Attention
!



SECTION IV

RESOURCES

<https://github.com/rstudio/expert>

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

rstudio / expert

Watch ▾ 32

Star 11

Fork 7

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Branch: master ▾

expert / pdfs /

Create new file

Upload files

Find file

History



garrettgman Added R Scripts and pdfs

Latest commit e027632 on 14 Sep 2015

..

01-Intro.pdf

Added R Scripts and pdfs

4 years ago

02-Manipulate.pdf

Added R Scripts and pdfs

4 years ago

03-Tidy.pdf

Added R Scripts and pdfs

4 years ago

04-Visualize.pdf

Added R Scripts and pdfs

4 years ago

05-Conclusion.pdf

Added R Scripts and pdfs

4 years ago

<https://rstudio.com/resources/cheatsheets/>

<https://github.com/rstudio/cheatsheets>



DOWNLOAD

SUPPORT

COMMUNITY



Products ▾

Resources ▾

Pricing

About ▾

Blogs ▾

RStudio Cheat Sheets

The cheat sheets below make it easy to use some of our favorite packages. From time to time, we will add new cheat sheets. If you'd like us to drop you an email when we do, click the button below.

[SUBSCRIBE TO CHEAT SHEET UPDATES](#)

CONTRIBUTED CHEAT

~~SHEE~~ETS TRANSLATIONS

HOW TO CONTRIBUTE

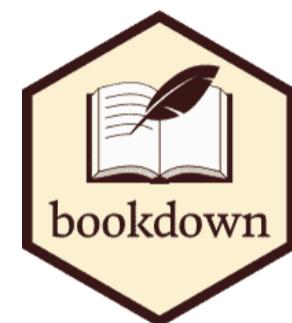


BOOKDOWN

Write HTML, PDF, ePUB, and Kindle books with R Markdown

The **bookdown** package is an [open-source R package](#) that facilitates writing books and long-form articles/reports with R Markdown. Features include:

- Generate printer-ready books and ebooks from R Markdown documents.
- A markup language easier to learn than LaTeX, and to write elements such as section headers, lists, quotes, figures, tables, and citations.
- Multiple choices of output formats: PDF, LaTeX, HTML, EPUB, and Word.
- Possibility of including dynamic graphics and interactive applications (HTML widgets and Shiny apps).
- Support a wide range of languages: R, C/C++, Python, Fortran, Julia, Shell scripts, and SQL, etc.
- LaTeX equations, theorems, and proofs work for all output formats.
- Can be published to GitHub, bookdown.org, and any web servers.
- Integrated with the RStudio IDE.
- One-click publishing to <https://bookdown.org>.



Below is a list of featured books. For a full list, please see the [archive](#) page. For the full documentation of the **bookdown** package, please see the free [online book](#) *bookdown: Authoring Books and Technical Documents with R Markdown*.

[Geocomputation with R](#)

[Statistical Inference via Data Science](#)



Below is a list of books written with **bookdown**, including those published to bookdown.org (books without substantial content are excluded) and a few hosted on external servers. The books are ordered roughly by date. An asterisk * after a date indicates the date is unknown, which often means a `date` field is missing in the YAML metadata of the source document `index.Rmd`. The list of books is automatically generated. For more information (including how to add or remove your books on this page), please see the [About](#) page.

Data Science avec R

by Fousseynou Bah

2019-02-27

Data Science avec R [...] En décidant d'écrire un livre sur la data science, j'ai longuement débattu dans ma propre tête, je me suis posé plusieurs questions dont une qui revenait constamment: "a-t-on vraiment besoin d'un autre livre sur la data science?" "N'en-t-on pas assez?" Avec le succès dont jouit la discipline, ce n'est certainement pas les ressources qui manquent, aussi bien en ligne que dans les librairies. Et surtout, je me demandais bien "qu'avais-je à dire qui n'avait pas été dit"? Et pourtant, quelques raisons m'ont poussé à reconsiderer ma position. La première est assez égoïte. ... *Read more →*

https://bookdown.org/fousseynoubah/dswr_book/

1

課程大綱 | ntpu-programming-for-data-sci...

by tpemartin

2019-02-27

資料科學程式設計（一） [...] 電子書網址：
<https://bookdown.org/tpemartin/ntpu-programming-for-data-science/>
電子書加個人註記：
<https://via.hypothes.is/https://bookdown.org/tpemartin/ntpu-programming-for-data-science/> gitter chatroom:
<https://gitter.im/ntpuecon/course-program-for-data-science107-2> This course is to build the foundation for being a data scientist—who masters both data analysis and data engineering. There are two programming languages that will be taught through the course: R and Javascript. R will serve as the data analysis backend, while ...
Read more →

2

前言

关于课程

课件中用到的宏包

RYouWithMe

致谢

作者简介

| 基础篇

1 数据科学与R语言

1.1 什么是数据科学

1.2 什么是R

1.2.1 R那些事

1.2.2 R是什么

1.2.3 R语言发展趋势

1.2.4 R路上的大神

数据科学中的 R 语言

王敏杰

2021-02-16

前言

你好，这里是四川师范大学研究生公选课《数据科学中的R语言》的课程内容。考虑到大家来自不同的学院，有着不同的学科背景，因此讲授的内容不会太深奥（要有信心喔）。

比如在课程中以下内容就不会出现

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

而出现更多的是

<https://static-bcrf.biochem.wisc.edu/courses/Tabular-data-analysis-with-R-and-Tidyverse/book/>

Tabular data analysis with R and Tidy...

Preamble

Learning goals

Software used during this tutorial

1 Introduction

1.1 Software installation

1.2 Installing R packages

1.3 Datasets: NHANES

1.4 Datasets: included in R

2 How R works

2.1 R is a software

2.2 R is a language

2.3 Working with R: objects and w...

3 Getting started

3.1 Launch RStudio

3.2 Organize with an RStudio proj...

3.3 Creating an R script



Tabular data analysis with R and Tidyverse:

Environmental Health



Environmental Health