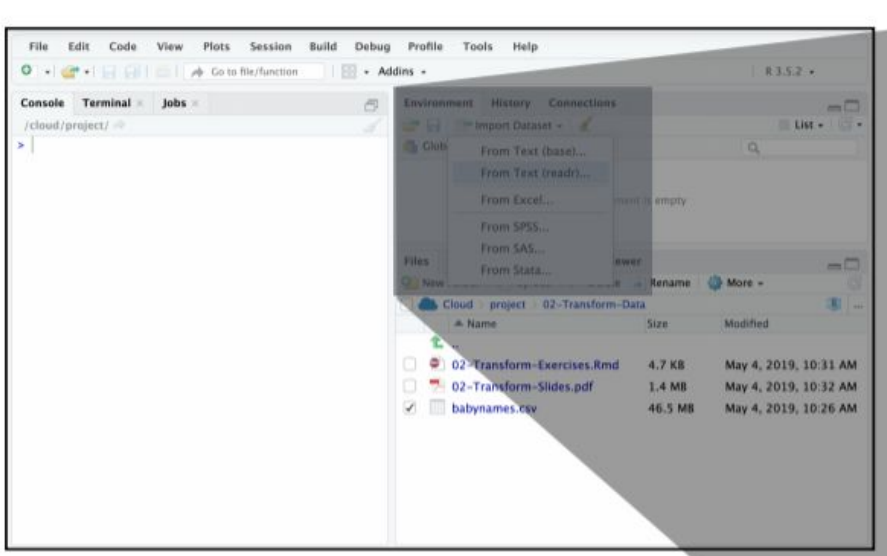
Lab 04

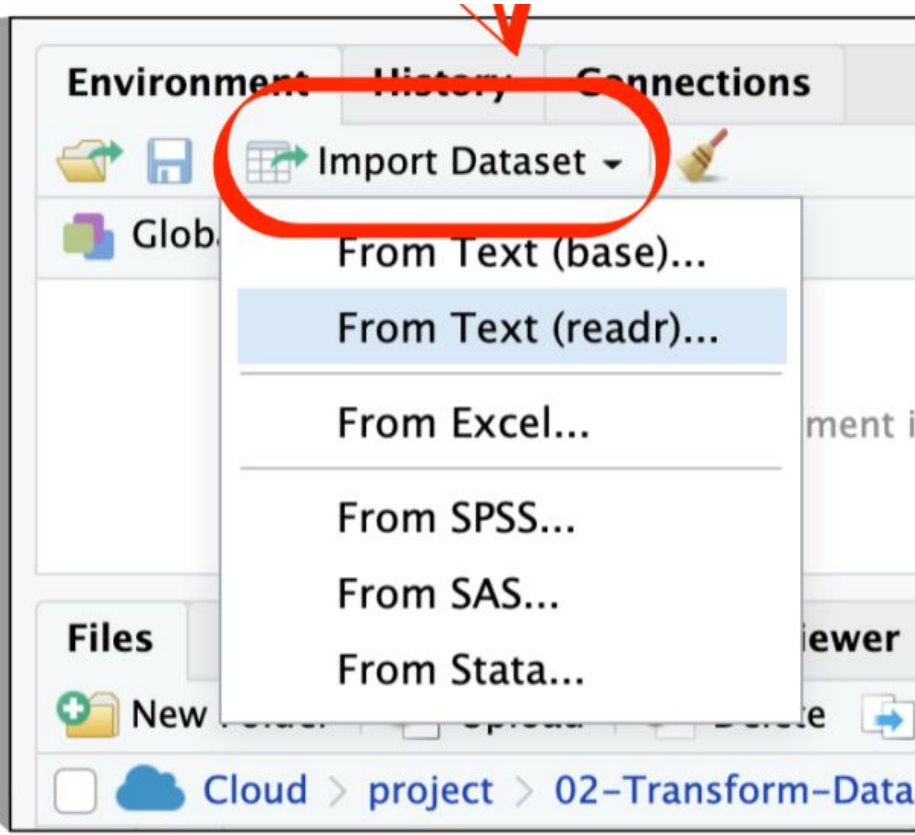
Data Transformation with dplyr

This presentation is largely based on Garrett Grolemund's *welcome-to-the-tidyverse*:
https://github.com/rstudio-education/welcome-to-the-tidyverse/tree/master/03-Transform

Import / Export Data

Import data





Environment Pane > Import Dataset > From Text (readr)

Export data: write_csv

```
write_csv(babynames, path = "babynames2.csv")
```

data frame

file path to save at

註:新版的 readr 是用 write_csv(babynames, file = "babynames2.csv")

dplyr

Why dplyr?

```
1  df <- tibble::as_tibble(iris)
2  df[(df$Species == "setosa") & (df$Sepal.Length < 5.8), "Species"]
```

Isolating data

`select()`	挑選**變項**
`filter()`	篩選**觀察值**
`arrange()`	排序**觀察值**

select()

```
select(babynames, name, prop )
```

data frame

babynames 中的
變項名稱

babynames

year	sex	name	n	prop
1880	M	John	9655	0.0815
1880	M	William	9532	0.0805
1880	M	James	5927	0.0501
1880	M	Charles	5348	0.0451
1880	M	Garrett	13	0.0001
1881	M	John	8769	0.081

name	prop
John	0.0815
William	0.0805
James	0.0501
Charles	0.0451
Garrett	0.0001
John	0.081

filter()

```
filter(babynames, name == "Garrett")
```

data frame

logical test

babynames

year	sex	name	n	prop
1880	M	John	9655	0.0815
1880	M	William	9532	0.0805
1880	M	James	5927	0.0501
1880	M	Charles	5348	0.0451
1880	M	Garrett	13	0.0001
1881	M	John	8769	0.081

year	sex	name	n	prop
1880	M	Garrett	13	0.0001
1881	M	Garrett	7	0.0001
…	…	Garrett	…	…

arrange()

```
arrange(babynames , n )
```

data frame

排序觀察值所據的
變項名稱

babynames

year	sex	name	n	prop
1880	M	John	9655	0.0815
1880	M	William	9532	0.0805
1880	M	James	5927	0.0501
1880	M	Charles	5348	0.0451
1880	M	Garrett	13	0.0001
1881	M	John	8769	0.081

year	sex	name	n	prop
1880	M	Garrett	13	0.0001
1880	M	Charles	5348	0.0451
1880	M	James	5927	0.0501
1881	M	John	8769	0.081
1880	M	William	9532	0.0805
1880	M	John	9655	0.0815

Your Turn

使用 `select()`, `filter()`, `arrange()` 找出...

1. 變項 **n** 的最小值為何？
2. 2017 年「最熱門 (最多)」的名字
3. 2017 年「最熱門 (最多)」的「女性」名字為何？

```
girls <- filter(babynames, year == 2017, sex == "F")

girls <- select(girls, name, n)

girls <- arrange(girls, desc(n))
```

The pipe operator %>%

```
filter(babynames, year == 2017)
```

```
babynames  %>%  filter(year == 2017)
```

傳入 filter() 的
第一個 argument

```
babynames  %>%  filter(_____, year == 2017)
```

```r
girls <- filter(babynames, year == 2017, sex == "F")

girls <- select(girls, name, n)

girls <- arrange(girls, desc(n))
```

```r
girls <- babynames %>%
   filter(year == 2017, sex == "F") %>%
   select(name, n) %>%
   arrange(desc(n))
```

Deriving information

`mutate()`	製造**變項**
`summarize()`	摘要**變項**
`group_by()`	分類**觀察值**

mutate()

```
babynames %>% mutate(percent = prop * 100)
```

新變項名稱

babynames

year	sex	name	n	prop
1880	M	John	9655	0.0815
1880	M	William	9532	0.0805
1880	M	James	5927	0.0501
1880	M	Charles	5348	0.0451
1880	M	Garrett	13	0.0001
1881	M	John	8769	0.081

year	sex	name	n	prop	percent
1880	M	John	9655	0.0815	8.15
1880	M	William	9532	0.0805	8.05
1880	M	James	5927	0.0501	5.01
1880	M	Charles	5348	0.0451	4.51
1880	M	Garrett	13	0.0001	0.01
1881	M	John	8769	0.081	8.1

summarize()

```
babynames %>%
  summarize(total = sum(n), max = max(n))
```

新變項名稱 新變項名稱

babynames

year	sex	name	n	prop
1880	M	John	9655	0.0815
1880	M	William	9532	0.0805
1880	M	James	5927	0.0501
1880	M	Charles	5348	0.0451
1880	M	Garrett	13	0.0001

total	max
127538	99680

複製貼上到 Console

```
pollution <- tibble::tribble(
       ~city,   ~size, ~amount,
  "New York", "large",      23,
  "New York", "small",      14,
    "London", "large",      22,
    "London", "small",      16,
   "Beijing", "large",      121,
   "Beijing", "small",      56
)
pollution
```

group_by()

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

```
pollution
```

group_by()

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14

London	large	22
London	small	16

Beijing	large	121
Beijing	small	56

```
pollution %>% group_by(city)
```

group_by()

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	mean	sum
New York	18.5	37
London	19.0	38
Beijing	88.5	177

```
pollution %>% group_by(city) %>%
  summarize(mean = mean(amount), sum = sum(amount))
```

group_by()

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14

city	particle size	amount ($\mu g/m^3$)
London	large	22
London	small	16

city	particle size	amount ($\mu g/m^3$)
Beijing	large	121
Beijing	small	56

city	mean	sum
New York	18.5	37
London	19.0	38
Beijing	88.5	177

```
pollution %>% group_by(city) %>%
  summarize(mean = mean(amount), sum = sum(amount))
```

Tidy Data Frame

A tidy data frame

	A	B	C	D
1	時間戳記	目前居住地	承上。您自何時定居於此	承上。您未來 5 年內是否
2	2018/6/12 下午 10:35:13	106; 臺北市; 大安區	1996 (民85)	是
3	2018/6/13 下午 7:04:51	204; 基隆市; 安樂區	2004 (民93)	是
4	2018/6/13 下午 7:05:44	103; 臺北市; 大同區	2013 (民102)	是
5	2018/6/13 下午 7:07:41	116; 臺北市; 文山區	2010 (民99)	是
6	2018/6/13 下午 7:07:46	105; 臺北市; 松山區	1997 (民86)	是
7	2018/6/13 下午 7:12:56	235; 新北市; 中和區	1986 (民75)	是
8	2018/6/13 下午 7:13:55	236; 新北市; 土城區	1980 (民69)	是
9	2018/6/13 下午 7:16:34	112; 臺北市; 北投區	2013 (民102)	是
10	2018/6/13 下午 7:19:20	237; 新北市; 三峽區	2000 (民89)	是
11	2018/6/13 下午 7:20:14	116; 臺北市; 文山區	1964 (民53)	是

一個變項

一筆資料
(觀察值)

A non-tidy data frame

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

3 筆資料
(觀察值)

變項 2:時間

變項 1:國家

變項 3:次數

Wide format
Tidy? ✗

Country	2011	2012	2013
FR	7000	6900	7000
DE	5800	6000	6200
US	15000	14000	13000

Long format
Tidy? ✓

Country	Year	n
FR	2011	7000
DE	2011	5800
US	2011	15000
FR	2012	6900
DE	2012	6000
US	2012	14000
FR	2013	7000
DE	2013	6200
US	2013	13000

tidyr

(r4ds.had.co.nz/tidy-data)