Tidying Data with tidyr

1: Multiple types of observational units are stored in the same table

~~2: Multiple variables are stored in one column~~

3: A single observational unit is stored in multiple tables

4: Variables are stored in both rows and columns

5: Every column contains a different variable

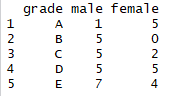
6: Column headers are values, not variable names

#### gather() 函数的使用

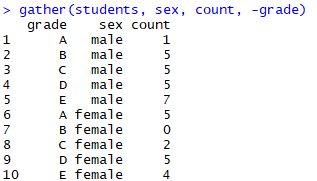
Gather columns into key-value pairs

gather(data, key = "key", value = "value", ..., na.rm = FALSE,convert = FALSE, factor\_ key = FALSE)

students



gather(students, sex, count, -grade) ##删除grade这个栏位，生成tidy的数据



> students2

grade male\_1 female\_1 male\_2 female\_2

1 A 3 4 3 4

2 B 6 4 3 5

3 C 7 4 3 8

4 D 4 0 8 1

5 E 1 1 2 7

这样的数据是不规范的，为了让他规范化，需要做数据的清洗和整理

> gather(students2, sex\_class, count, -grade)

grade sex\_class count

1 A male\_1 3

2 B male\_1 6

3 C male\_1 7

4 D male\_1 4

5 E male\_1 1

6 A female\_1 4

7 B female\_1 4

8 C female\_1 4

9 D female\_1 0

10 E female\_1 1

11 A male\_2 3

12 B male\_2 3

13 C male\_2 3

14 D male\_2 8

15 E male\_2 2

16 A female\_2 4

17 B female\_2 5

18 C female\_2 8

19 D female\_2 1

20 E female\_2 7

>res <- gather(students2, sex\_class, count, -grade)

#### separate()函数

Separate one column into multiple columns.

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE,

convert = FALSE, extra = "warn", fill = "warn", ...)

> separate(data = res, col = sex\_class, into = c("sex", "class") )

grade sex class count

1 A male 1 3

2 B male 1 6

3 C male 1 7

4 D male 1 4

5 E male 1 1

6 A female 1 4

7 B female 1 4

8 C female 1 4

9 D female 1 0

10 E female 1 1

11 A male 2 3

12 B male 2 3

13 C male 2 3

14 D male 2 8

15 E male 2 2

16 A female 2 4

17 B female 2 5

18 C female 2 8

19 D female 2 1

20 E female 2 7

#### Chain

students2 %>%

gather( sex\_class, count, -grade) %>%

separate( sex\_class, c("sex", "class")) %>%

print

> students3

name test class1 class2 class3 class4 class5

1 Sally midterm A <NA> B <NA> <NA>

2 Sally final C <NA> C <NA> <NA>

3 Jeff midterm <NA> D <NA> A <NA>

4 Jeff final <NA> E <NA> C <NA>

5 Roger midterm <NA> C <NA> <NA> B

6 Roger final <NA> A <NA> <NA> A

7 Karen midterm <NA> <NA> C A <NA>

8 Karen final <NA> <NA> C A <NA>

9 Brian midterm B <NA> <NA> <NA> A

10 Brian final B <NA> <NA> <NA> C

students3 %>%

gather( class, grade, class1:class5 ,na.rm = TRUE) %>%

print

#### spread函数

Spread a key-value pair across multiple columns.

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE,

sep = NULL)

students3 %>%

gather(class, grade, class1:class5, na.rm = TRUE) %>%

spread(test , grade) %>%

print

students3 %>%

gather(class, grade, class1:class5, na.rm = TRUE) %>%

spread(test, grade) %>%

mutate(class =parse\_number(class)) %>%

print

> passed <- mutate(passed, status = "passed")

> failed <- mutate(failed, status = "failed")

Call bind\_rows() with two arguments, passed and failed (in that order), to join the two tables.

>bind\_rows(passed, failed)

sat %>%

select(-contains("total")) %>%

gather(key = part\_sex, value = count, -score\_range) %>%

separate(part\_sex, into = c("part", "sex")) %>%

print

sat %>%

select(-contains("total")) %>%

gather(part\_sex, count, -score\_range) %>%

separate(part\_sex, c("part", "sex")) %>%

group\_by(part, sex) %>%

mutate(total = sum(count),

prop = count/ total

) %>% print