R tidyverse 마스터 클래스

8강 - tidyr 패키지 파헤치기 1탄

슬기로운통계생활

Issac Lee

학습목표

지저분한 데이터를 깨끗하게!

tidyr를 사용한 전처리 학습하기

Pivoting

- `pivot_longer()`
- `pivot_wider()`
- `separate()`
- `unite()`

Tidy한 데이터는 사랑입니다.♡

데이터를 tidy하게!

우리는 항상 **tidy**한 데이터만 다루는게 아님

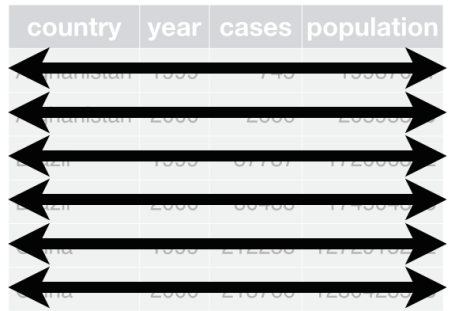
tidy data란 무엇인가?

- 가로 행이 하나의 표본을 나타내는 데이터
- 각 열은 하나의 변수를 나타내는 데이터

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

table1

variables



observations

- tidyverse는 tidy 한 데이터를 다룰때 최적화 되어있는 코드

Image from https://garrettgman.github.io

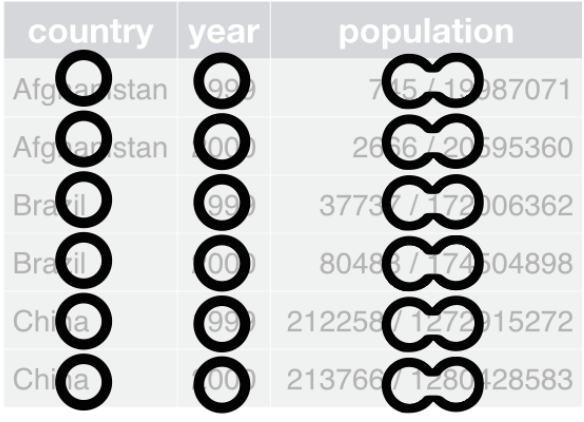
Messy data들의 예

- 한 열에 두개의 정보가 섞여있음

country	year	population
Afghanistan	1999	745 / 19987071
Afghanistan	2000	2666 / 20595360
Brazil	1999	37737 / 172006362
Brazil	2000	80488 / 174504898
China	1999	212258 / 1272915272
China	2000	213766 / 1280428583

table3

variables



values

Image from https://garrettgman.github.io/tidying/

Messy data들의 예

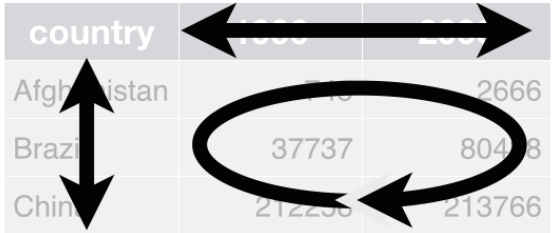
- 한 변수의 정보가 열로 이어져 있음

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

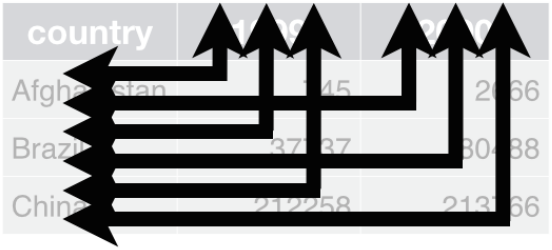
table4

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

table5



variables



observations

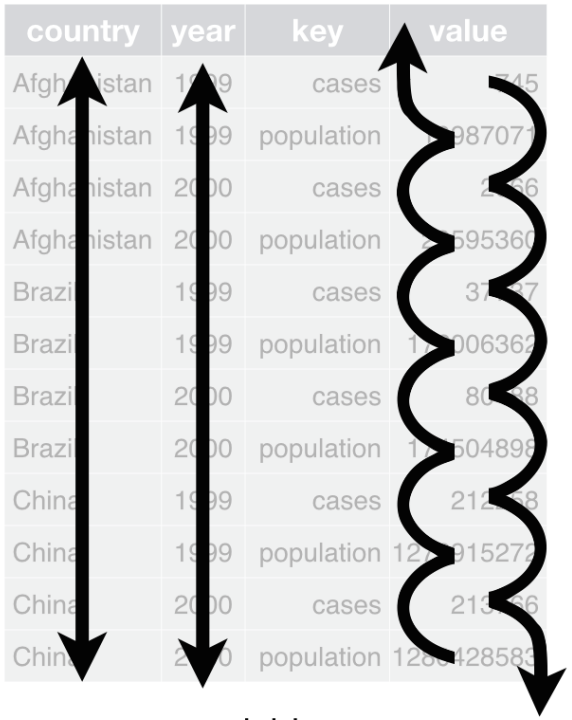
Image from https://garrettgman.github.io/tidying/

Messy data들의 예

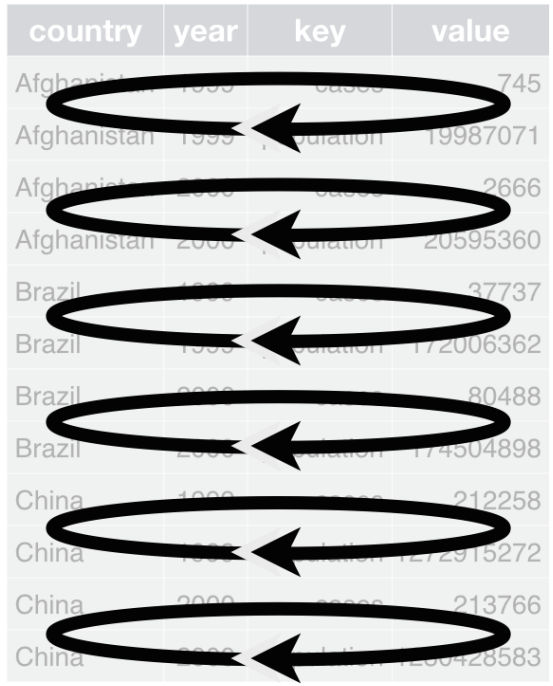
- 두 변수의 정보가 하나의 열에 섞여있음.

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2



variables



observations

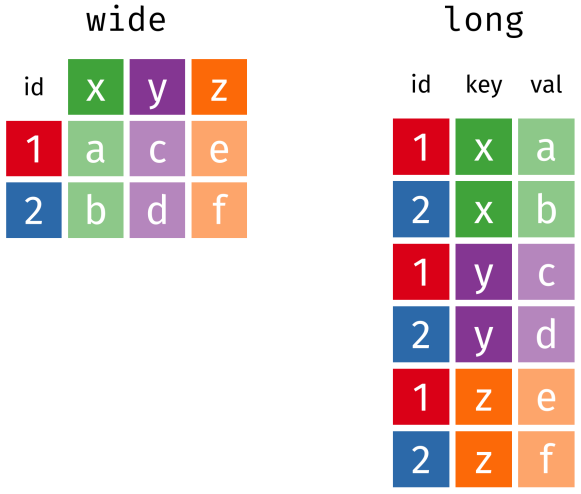
Image from https://garrettgman.github.io/tidying/

`pivot_longer()`

변수 정보를 잡아서 `pivot_longer()`

- 퍼져있는 정보 접어줌 (longer)

- 비교적 최신 함수 - `gather()` 함수를 계승



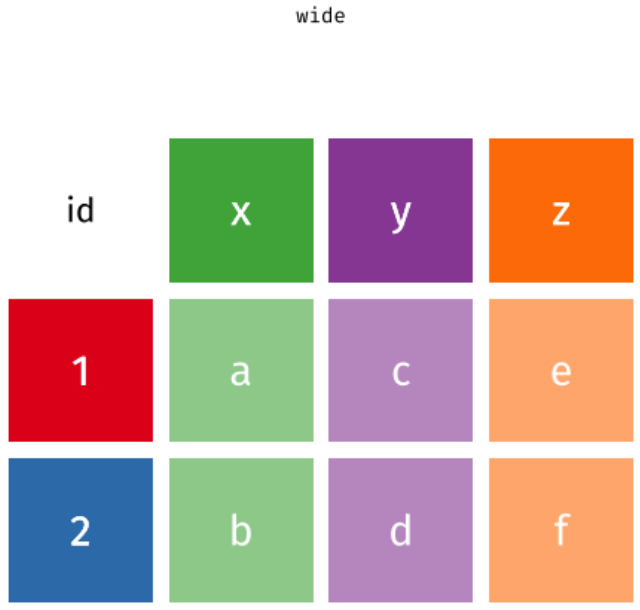


Image from https://education.rstudio.com/blog/2020/07/teaching-the-tidyverse-in-2020-part-3-data-wrangling-and-tidying/

데이터를 접어주는 - `pivot_longer()`

변수 정보를 잡아서

- 퍼져있는 정보 접어줌 (longer)

문법

```
dataframe_wide %>%
  pivot_longer(
    cols = !id,
    names_to = "key",
    values_to = "val"
  )
```

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

유용한 옵션들

```
sample_wider <- tibble::tribble(
     ~국가,   ~yr2019,   ~yr2020,   ~yr2021
    "한국",     13L,     NA,     56L,
    "미국",     22L,     52L,     NA,
    "일본",     44L,     63L,     82L
    )
sample_wider
```

```
## # A tibble: 3 x 4
##   국가  yr2019 yr2020 yr2021
##   <chr>  <int>  <int>  <int>
## 1 한국      13     NA     56
## 2 미국      22     52     NA
## 3 일본      44     63     82
```

names_prefix

- 변수명에 있는 정보만 빼내어 칼럼 만들기

```
sample_wider %>%
    pivot_longer(col = starts_with("yr"),
                 names_to = "year",
                 names_prefix = "yr",
                 values_to = "gdp")
```

```
## # A tibble: 9 x 3
##   국가  year    gdp
##   <chr> <chr> <int>
## 1 한국  2019     13
## 2 한국  2020     NA
## 3 한국  2021     56
```

유용한 옵션들

values_drop_na == TRUE

- 비어있는 정보 생략하기

```
sample_wider %>%
    pivot_longer(col = starts_with("yr"),
                 names_to = "year",
                 values_drop_na = TRUE,
                 values_to = "gdp")
```

```
## # A tibble: 7 x 3
##   국가  year     gdp
##   <chr> <chr>  <int>
## 1 한국  yr2019    13
## 2 한국  yr2021    56
## 3 미국  yr2019    22
```

names_transform

- 잡아온 정보의 타입지정

```
sample_wider %>%
    pivot_longer(col = starts_with("yr"),
        names_to = "year",
        names_prefix = "yr",
        names_transform = list(year = as.ir
        values_to = "gdp")
```

```
## # A tibble: 9 x 3
##   국가   year   gdp
##   <chr> <int> <int>
## 1 한국   2019    13
## 2 한국   2020    NA
```

패턴을 가진 칼럼이름

WHO 데이터

- Tuberculosis (결핵)

```
glimpse(who)
```

```
## Rows: 7,240
## Columns: 60
## $ country      <chr> "Afghanistan", "Afghanistan", "Afghanistan",
## $ iso2         <chr> "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF
## $ iso3         <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AF
## $ year         <int> 1980, 1981, 1982, 1983, 1984, 1985, 1986, 198
## $ new_sp_m014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_m65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
## $ new_sp_f014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N
```

diagnosis

- relapse
- negative pulmonary smear
- positive pulmonary smear
- extrapulmonary

gender

- male
- female

age

- 014: 0-14
- 3544: 35-44

```
names(who) %>% unique()
```

```
##  [1] "country"      "iso2"         "iso3"         "year"
##  [5] "new_sp_m014"  "new_sp_m1524" "new_sp_m2534" "new_sp_m3544"
##  [9] "new_sp_m4554" "new_sp_m5564" "new_sp_m65"   "new_sp_f014"
## [13] "new_sp_f1524" "new_sp_f2534" "new_sp_f3544" "new_sp_f4554"
## [17] "new_sp_f5564" "new_sp_f65"   "new_sn_m014"  "new_sn_m1524"
## [21] "new_sn_m2534" "new_sn_m3544" "new_sn_m4554" "new_sn_m5564"
## [25] "new_sn_m65"   "new_sn_f014"  "new_sn_f1524" "new_sn_f2534"
## [29] "new_sn_f3544" "new_sn_f4554" "new_sn_f5564" "new_sn_f65"
## [33] "new_ep_m014"  "new_ep_m1524" "new_ep_m2534" "new_ep_m3544"
## [37] "new_ep_m4554" "new_ep_m5564" "new_ep_m65"   "new_ep_f014"
## [41] "new_ep_f1524" "new_ep_f2534" "new_ep_f3544" "new_ep_f4554"
## [45] "new_ep_f5564" "new_ep_f65"   "newrel_m014"  "newrel_m1524"
## [49] "newrel_m2534" "newrel_m3544" "newrel_m4554" "newrel_m5564"
## [53] "newrel_m65"   "newrel_f014"  "newrel_f1524" "newrel_f2534"
## [57] "newrel_f3544" "newrel_f4554" "newrel_f5564" "newrel_f65"
```

정규표현식을 이용한 칼럼정보 잡아내기

- new 뒤에 _가 있을수도 있고 없을수도 있고 `?`
- 한글자 `.` 이상 `*` 매칭
- 밑줄 `_` 이후 한글자 `.`와 나머지 글자들 `.*`

```
who %>% pivot_longer(
  cols = new_sp_m014:newrel_f65,
  names_to = c("diagnosis", "gender", "age"),
  names_pattern = "new_?(.*)_(.)(.*)",
  values_to = "count"
)
```

```
## # A tibble: 405,440 x 8
##    country     iso2  iso3   year diagnosis gender age   count
##    <chr>       <chr> <chr> <int> <chr>     <chr>  <chr> <int>
##  1 Afghanistan AF    AFG    1980 sp        m      014      NA
##  2 Afghanistan AF    AFG    1980 sp        m      1524     NA
##  3 Afghanistan AF    AFG    1980 sp        m      2534     NA
##  4 Afghanistan AF    AFG    1980 sp        m      3544     NA
##  5 Afghanistan AF    AFG    1980 sp        m      4554     NA
##  6 Afghanistan AF    AFG    1980 sp        m      5564     NA
```

https://www.computerhope.com/unix/regex-quickref.htm

특정 구조의 칼럼 이름

한반에 (row)

- 2명 학생의
- 2개의 정보

```
school_db <- tibble::tribble(
    ~class, ~score_std1, ~score_std2, ~gender_std1, ~gender_std2,
        1L, 87L, 25L, "M", "F",
        2L, 45L, 36L, "F", "M",
        3L, 76L, 43L, "F", "F"
    )
glimpse(school_db)
```

```
## Rows: 3
## Columns: 5
## $ class       <int> 1, 2, 3
## $ score_std1  <int> 87, 45, 76
## $ score_std2  <int> 25, 36, 43
## $ gender_std1 <chr> "M", "F", "F"
## $ gender_std2 <chr> "F", "M", "F"
```

특정 구조의 칼럼 정보화

- `names_sep`을 구분점으로 나눔
  - `score`와 `gender` 정보는 `_` 기준 왼쪽에 위치
  - 왼쪽에 위치한 정보들은 그 이름 자체를 `values_to` 로 보내줘~!

```
school_db %>%
  pivot_longer(
    !class,
    names_to = c(".value", "student"),
    names_sep = "_",
    values_drop_na = TRUE
  )
```

```
## # A tibble: 6 x 4
##   class student score gender
##   <int> <chr>   <int> <chr>
## 1     1 std1       87 M
## 2     1 std2       25 F
## 3     2 std1       45 F
## 4     2 std2       36 M
## 5     3 std1       76 F
## 6     3 std2       43 F
```

데이터를 펼쳐주는 `pivot_wider()`

변수 정보를 잡아서

- 접혀있는 정보 펼쳐줌 (wider)

접미사가 다름

- names_from
- values_from

문법

```
dataframe_long %>%
  pivot_longer(
    cols = !id,
    names_from = "key",
    values_from = "val"
  )
```

wide

id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

데이터를 펼쳐주는 - `pivot_wider()`

펭귄 데이터

- 섬별 종별 contingency table

```
library(palmerpenguins)
sample_data <- penguins %>%
    group_by(species, island) %>%
    summarise(body_kg = mean(body_mass_g, na.rm = TRUE) / 1000,
              bill_mm = mean(bill_length_mm, na.rm = TRUE))
sample_data
```

```
## # A tibble: 5 x 4
## # Groups:   species [3]
##   species   island    body_kg bill_mm
##   <fct>     <fct>       <dbl>   <dbl>
## 1 Adelie    Biscoe       3.71    39.0
## 2 Adelie    Dream        3.69    38.5
## 3 Adelie    Torgersen    3.71    39.0
## 4 Chinstrap Dream        3.73    48.8
```

2차원 분할표 만들기

```
sample_data %>%
    pivot_wider(
        id_cols = species,
        names_from = island,
        values_from = body_kg
    )
```

```
## # A tibble: 3 x 4
## # Groups:   species [3]
##   species   Biscoe Dream Torgersen
##   <fct>      <dbl> <dbl>     <dbl>
## 1 Adelie      3.71  3.69      3.71
## 2 Chinstrap  NA     3.73     NA
## 3 Gentoo      5.08 NA        NA
```

```
sample_data %>%
    pivot_wider(
        id_cols = species,
        names_from = island,
        values_from = body_kg,
        values_fill = 0,
    )
```

```
## # A tibble: 3 x 4
## # Groups:   species [3]
##   species   Biscoe Dream Torgersen
##   <fct>      <dbl> <dbl>     <dbl>
## 1 Adelie      3.71  3.69      3.71
## 2 Chinstrap   0     3.73      0
## 3 Gentoo      5.08  0         0
```

여러 칼럼에서 값 정보 빼내기

```
sample_data %>%
    pivot_wider(
        id_cols = species,
        names_from = island, # Name 잡아오는 열은 하나
        values_from = c(body_kg, bill_mm), # Value를 잡아오는 칼럼이 여러개
        values_fill = 0
    )
```

```
## # A tibble: 3 x 7
## # Groups:   species [3]
##   species   body_kg_Biscoe body_kg_Dream body_kg_Torgersen bill_mm_Biscoe
##   <fct>              <dbl>         <dbl>              <dbl>          <dbl>
## 1 Adelie              3.71          3.69               3.71           39.0
## 2 Chinstrap           0             3.73               0               0
## 3 Gentoo              5.08          0                  0              47.5
## # ... with 2 more variables: bill_mm_Dream <dbl>,
```

칼럼 이름 조정하기

- 변수와 값 사이 연결고리

```
sample_data %>%
    pivot_wider(
        id_cols = species,
        names_from = island,
        values_from = c(body_kg,
                        bill_mm),
        values_fill = 0,
        names_sep = "*"
    )
```

```
## # A tibble: 3 x 7
## # Groups:   species [3]
##   species   `body_kg*Biscoe` `body_kg*Dr
##   <fct>                <dbl>          <
```

- 변수와 값 이름 Customization

```
sample_data %>%
    pivot_wider(
        id_cols = species,
        names_from = island,
        values_from = c(body_kg,
                        bill_mm),
        values_fill = 0,
        names_glue = "{island}섬의 {.value}"
    )
```

```
## # A tibble: 3 x 7
## # Groups:   species [3]
##   species   `Biscoe섬의 body_kg` `Dream섬
##   <fct>                   <dbl>
```

`values_fn` 옵션 이해하기

`summarise()`
단계 생략

```
penguins %>%
    drop_na() %>%
    select(species, island, body_mass_g) %>%
    pivot_wider(
        names_from = island,
        values_from = body_mass_g,
        values_fn = mean,
        values_fill = 0
    )
```

```
## # A tibble: 3 x 4
##   species   Torgersen Biscoe Dream
##   <fct>         <dbl>  <dbl> <dbl>
## 1 Adelie        3709.  3710. 3701.
## 2 Gentoo           0   5092.    0
```

잠깐 소개! glue 패키지

- `paste()`를 대체한 Rstudio 패키지


```
library(glue)
name <- "슬통이"
glue("안녕하세요!
      슬기로운 통계생활의
      {name}입니다.")
```

```
## 안녕하세요!
## 슬기로운 통계생활의
## 슬통이입니다.
```

- 구구단 만들기

```
a <- 1:9
x <- 9
glue('{x} X {a} = {x * a}')
```

```
## 9 X 1 = 9
## 9 X 2 = 18
## 9 X 3 = 27
## 9 X 4 = 36
## 9 X 5 = 45
## 9 X 6 = 54
## 9 X 7 = 63
## 9 X 8 = 72
## 9 X 9 = 81
```

한 셀에 담기 여러 정보들 분리하기

```
gugudan <- tibble(
    multiple = glue('{x} X {a}'),
    result = glue('{x * a}')
)
gugudan
```

```
## # A tibble: 9 x 2
##   multiple result
##   <glue>   <glue>
## 1 9 X 1    9
## 2 9 X 2    18
## 3 9 X 3    27
## 4 9 X 4    36
## 5 9 X 5    45
## 6 9 X 6    54
## 7 9 X 7    63
```

```
gugudan %>%
    separate(col = multiple,
             into = c("level", "multiplier"
             sep = " X ")
```

```
## # A tibble: 9 x 3
##   level multiplier result
##   <chr> <chr>      <glue>
## 1 9     1          9
## 2 9     2          18
## 3 9     3          27
## 4 9     4          36
## 5 9     5          45
## 6 9     6          54
## 7 9     7          63
## 8 9     8          72
```

`separate()` 함수 옵션 설정

- extra
- fill

```
df <- data.frame(x = c("x", "x y", "x y z"))
```

```
df
```

```
##       x
## 1     x
## 2   x y
## 3 x y z
```

extra

- `sep`으로 나눴을때 처리 방법
  - 기본세팅: 지우고 경고처리
  - "drop": 경고없이 지움
  - "merge": 남는 것들 붙임

fill

- 채우기 모자란 경우
  - 기본세팅: 빈칸을 오른쪽부터 채우고 경고
  - "left": 빈칸 왼쪽부터 채움

```
library(tidyverse)
df <- data.frame(x = c("x", "x y
df %>%
    separate(x, c("a", "b"),
             extra = "drop",
             fill = "left")
```

```
##      a b
## 1 <NA> x
## 2    x y
## 3    x y
```

한 열을 여러 줄 (rows)로 나눌 때

`separate_rows()`

- 여러 정보가 들어있는 열을 분리 후에 다른 줄로 만들어줌
  - `separate()` + `pivot_longer()` 느낌
  - `convert = TRUE`로 결과물을 숫자로 바꿔줌.

```
gugudan %>%
    separate_rows(multiple,
                  sep = " X ",
                  convert = TRUE)
```

```
## # A tibble: 18 x 2
##    multiple result
##       <int> <glue>
##  1        9 9
##  2        1 9
##  3        9 18
##  4        2 18
##  5        9 27
##  6        3 27
##  7        9 36
##  8        4 36
```

여러셀의 정보를 합칠땐 unite()

- `remove`
  합쳤으니 이전
  정보 지울까?
- `sep`
  합칠때 사이에
  뭘 끼워넣을까?
- `na.rm=TRUE`
  빈 셀 합칠때
  NA는 빈칸으로!

```
gugudan %>%
    unite("gugudan",
       multiple:result,
       remove = FALSE)
```

```
## # A tibble: 9 x 3
##   gugudan  multiple result
##   <chr>    <glue>   <glue>
## 1 9 X 1_9  9 X 1    9
## 2 9 X 2_18 9 X 2    18
## 3 9 X 3_27 9 X 3    27
## 4 9 X 4_36 9 X 4    36
## 5 9 X 5_45 9 X 5    45
## 6 9 X 6_54 9 X 6    54
## 7 9 X 7_63 9 X 7    63
## 8 9 X 8_72 9 X 8    72
```

```
gugudan %>%
    unite("gugudan",
       multiple:result,
       sep = "=")
```

```
## # A tibble: 9 x 1
##   gugudan
##   <chr>
## 1 9 X 1=9
## 2 9 X 2=18
## 3 9 X 3=27
## 4 9 X 4=36
## 5 9 X 5=45
## 6 9 X 6=54
## 7 9 X 7=63
## 8 9 X 8=72
```

다음시간

- 더 Deep dive 합니다..

NA 처리하는 방법

- `drop_na()`
- `fill()`
- `replace_na()`

Nested Data 개념 잡기

- 데이터 프레임 안에 데이터 프레임이 들어간다고??

같이 보면 좋은 책 추천

[1] R for Data Science

- 웹 상에 무료 공개된 책입니다.
- 위 교재의 한글 번역본 R 을 활용한 데이터과학도 있습니다.
- 도서 제목 클릭하셔서 구매하시면 저의 사리사욕을 충당하는데 도움이 됩니다.

참고자료

- tidyr pivot 설명