데이터 다듬기

문 건 웅

2017년 8월 26일

데이터 다듬기(Tidy Data)

"Happy families are all alike; every unhappy family is unhappy in its own way." — Leo Tolstoy

데이터 다듬기(Tidy Data)

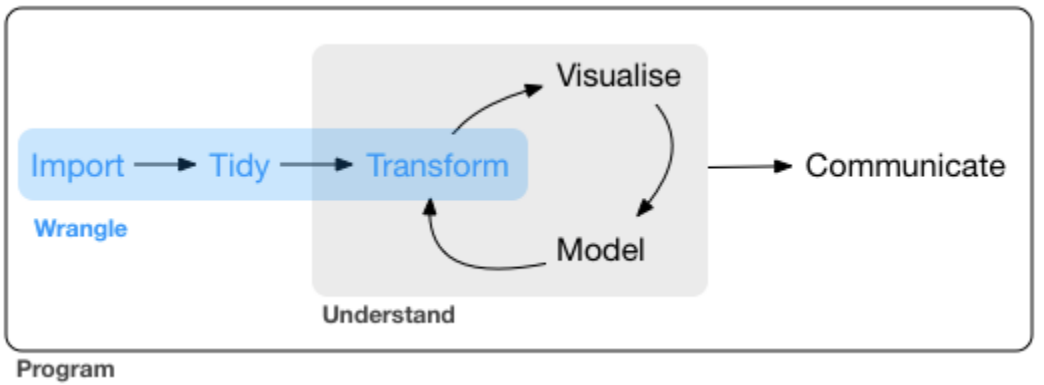
"Happy families are all alike; every unhappy family is unhappy in its own way." — Leo Tolstoy

"Tidy datasets are all alike, but every messy dataset is messy in its own way." — Hadley Wickham

데이터 다듬기(Tidy Data)

"Happy families are all alike; every unhappy family is unhappy in its own way." — Leo Tolstoy

"Tidy datasets are all alike, but every messy dataset is messy in its own way." — Hadley Wickham



Prerequisites

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

library(tidyverse)
```

```
table1
```

```
# A tibble: 6 x 4
      country  year  cases population
        <chr> <int>  <int>      <int>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3      Brazil  1999  37737  172006362
4      Brazil  2000  80488  174504898
5       China  1999 212258 1272915272
6       China  2000 213766 1280428583
```

```
table2
```

```
# A tibble: 12 x 4
       country  year       type      count
         <chr> <int>      <chr>      <int>
 1 Afghanistan  1999      cases        745
 2 Afghanistan  1999 population   19987071
 3 Afghanistan  2000      cases       2666
 4 Afghanistan  2000 population   20595360
 5      Brazil  1999      cases      37737
 6      Brazil  1999 population  172006362
 7      Brazil  2000      cases      80488
 8      Brazil  2000 population  174504898
 9       China  1999      cases     212258
10       China  1999 population 1272915272
11       China  2000      cases     213766
12       China  2000 population 1280428583
```

```
table3
```

```
# A tibble: 6 x 3
      country  year              rate
        <chr> <int>             <chr>
1 Afghanistan  1999      745/19987071
2 Afghanistan  2000     2666/20595360
3      Brazil  1999   37737/172006362
4      Brazil  2000   80488/174504898
5       China  1999 212258/1272915272
6       China  2000 213766/1280428583
```

```
table4a # cases
```

```
# A tibble: 3 x 3
      country `1999` `2000`
        <chr>  <int>  <int>
1 Afghanistan    745   2666
2      Brazil  37737  80488
3       China 212258 213766
```

```
table4b # population
```

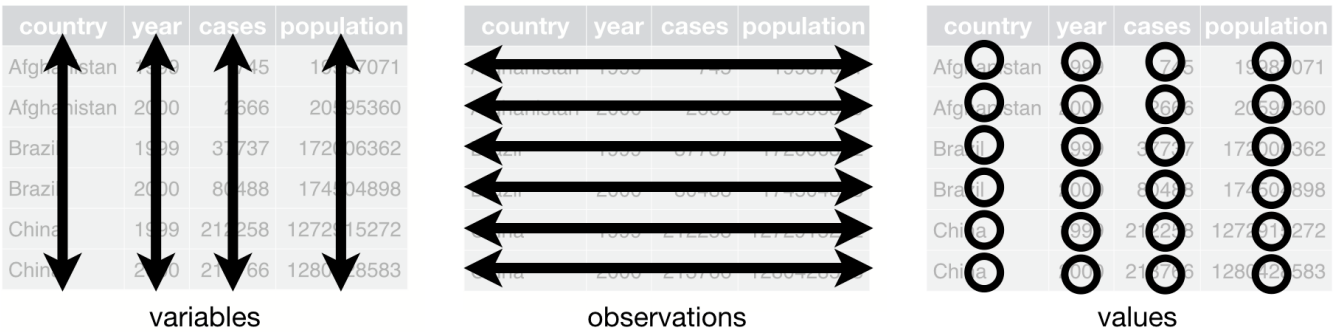
```
# A tibble: 3 x 3
      country     `1999`     `2000`
        <chr>      <int>      <int>
1 Afghanistan   19987071   20595360
2      Brazil  172006362  174504898
3       China 1272915272 1280428583
```

Tidy data 란? 세 가지 규칙(Three rules)

1. 각 변수는 고유한 열에 위치(Each variable must have its own column)
2. 각 관찰치는 고유한 행에 위치(Each observation must have its own row)
3. 각 수치는 고유한 cell에 위치(Each value must have its own cell)

Tidy data 란? 세 가지 규칙(Three rules)

1. 각 변수는 고유한 열에 위치(Each variable must have its own column)
2. 각 관찰치는 고유한 행에 위치(Each observation must have its own row)
3. 각 수치는 고유한 cell에 위치(Each value must have its own cell)



Tidy data의 잇점

1.일관성 있는 데이터의 구조

2.변수가 열에 위치하고 있기 때문에 R의 장점인 벡터화된 연산이 가능하다.

=> 데이터 분석에 유용한 구조로 dplyr, ggplot2등 tidyverse 패키지들은 모두 tidy data에서 작동한다.

Tidy data의 잇점

1.일관성 있는 데이터의 구조

2.변수가 열에 위치하고 있기 때문에 R의 장점인 벡터화된 연산이 가능하다.

=> 데이터 분석에 유용한 구조로 dplyr, ggplot2등 tidyverse 패키지들은 모두 tidy data에서 작동한다.

예 1) 인구 만 명당 유병률

```
table1 %>%
    mutate(rate = cases/population * 10000)
```

```
# A tibble: 6 x 5
      country  year  cases population     rate
        <chr> <int>  <int>      <int>    <dbl>
1 Afghanistan  1999    745   19987071 0.372741
2 Afghanistan  2000   2666   20595360 1.294466
3      Brazil  1999  37737  172006362 2.193930
4      Brazil  2000  80488  174504898 4.612363
5       China  1999 212258 1272915272 1.667495
6       China  2000 213766 1280428583 1.669488
```

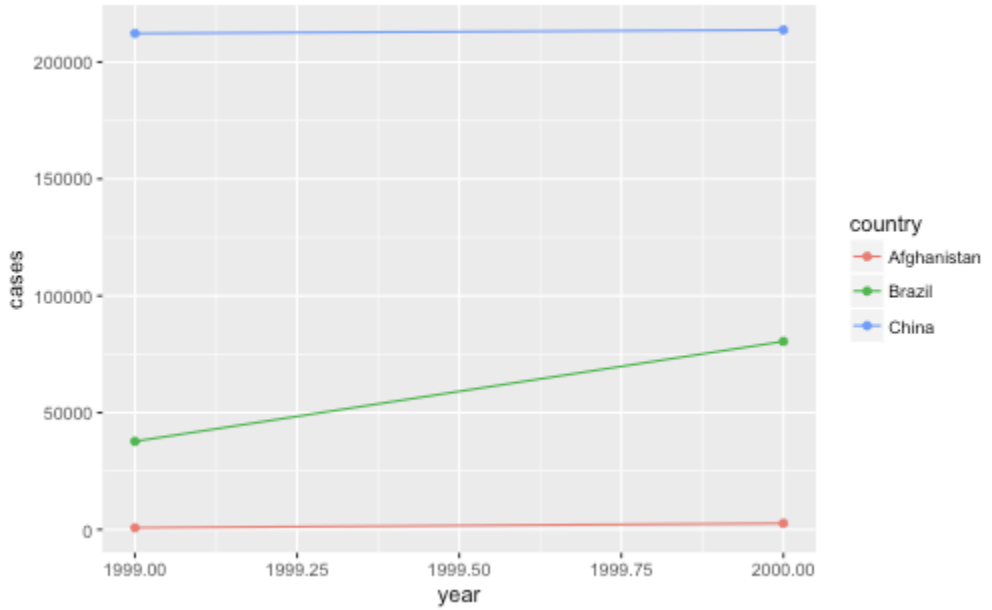
예 2) 년도별 환자수 총계

```
table1 %>%
    count(year,wt=cases)
```

```
# A tibble: 2 x 2
   year      n
  <int>  <int>
1  1999 250740
2  2000 296920
```

예 3) 년도별 환자수 시각화

```
ggplot(table1,aes(year,cases,colour=country)) +
    geom_point() +
    geom_line()
```



Spreading 과 Gathering

- 실제 접할 수 있는 대부분의 데이터는 tidy data가 아니다. 그 이유는 크게 두 가지인데 첫째, 대부분의 사람들은 tidy data의 개념이 없으며 둘째, 데이터는 종종 분석 이외에 다른 목적에 맞게 구조화되어 있기 때문이다. 어떤 데이터는 입력하기 쉬운 구조로 되어 있다.

- 깔끔한 데이타를 만들기 위한 첫번째 단계는 변수와 관측치를 구별하는 일이다. 두번째는 다음과 같은 흔한 문제를 해결하는 것이다.

1. 한 변수가 여러 열에 분산되어 있는 경우

2. 한 관측치가 여러행에 흩어져 있는 경우

=> spread()와 gather()로 해결할 수 있다.

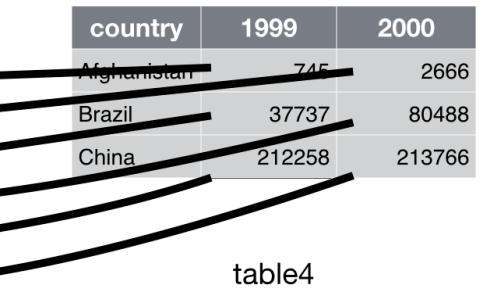
Gathering

```
table4a
```

```
# A tibble: 3 x 3
      country `1999` `2000`
        <chr>  <int>  <int>
1 Afghanistan    745   2666
2      Brazil  37737  80488
3       China 212258 213766
```

table4a의 1999와 2000은 변수의 이름이 아니고 year 변수의 값이고 각 행은 하나의 관측치가 아니라 두개의 관측치이다.

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



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

table4


```
table4a %>%
    gather(`1999`,`2000`,key="year",value="cases")
```

```
# A tibble: 6 x 3
      country  year  cases
        <chr> <chr>  <int>
1 Afghanistan  1999    745
2      Brazil  1999  37737
3       China  1999 212258
4 Afghanistan  2000   2666
5      Brazil  2000  80488
6       China  2000 213766
```

```
table4b
```

```
# A tibble: 3 x 3
      country     `1999`     `2000`
        <chr>      <int>      <int>
1 Afghanistan   19987071   20595360
2      Brazil  172006362  174504898
3       China 1272915272 1280428583
```

```
table4b %>%
  gather(`1999`, `2000`, key = "year", value = "population")
```

```
# A tibble: 6 x 3
      country  year population
        <chr> <chr>      <int>
1 Afghanistan  1999   19987071
2      Brazil  1999  172006362
3       China  1999 1272915272
4 Afghanistan  2000   20595360
5      Brazil  2000  174504898
6       China  2000 1280428583
```

```
tidy4a <- table4a %>%
  gather(`1999`, `2000`, key = "year", value = "cases")
tidy4b <- table4b %>%
  gather(`1999`, `2000`, key = "year", value = "population")
left_join(tidy4a, tidy4b)
```

```
# A tibble: 6 x 4
      country  year  cases population
        <chr> <chr>  <int>      <int>
1 Afghanistan  1999    745   19987071
2      Brazil  1999  37737  172006362
3       China  1999 212258 1272915272
4 Afghanistan  2000   2666   20595360
5      Brazil  2000  80488  174504898
6       China  2000 213766 1280428583
```

Spreading

```
table2
```

```
# A tibble: 12 x 4
       country  year       type      count
         <chr> <int>      <chr>      <int>
 1 Afghanistan  1999      cases        745
 2 Afghanistan  1999 population   19987071
 3 Afghanistan  2000      cases       2666
 4 Afghanistan  2000 population   20595360
 5      Brazil  1999      cases      37737
 6      Brazil  1999 population  172006362
 7      Brazil  2000      cases      80488
 8      Brazil  2000 population  174504898
 9       China  1999      cases     212258
10       China  1999 population 1272915272
11       China  2000      cases     213766
12       China  2000 population 1280428583
```

table2에는 하나의 관측치가 두개의 행에 나누어져 있다. 하나의 관측치는 한 나라, 한 해의 데이터인데 각 관측치가 두 행에 나뉘어져 있다. 이 경우 spread()함수로 데이터를 깔끔한 데이터로 만들 수 있다.

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

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

```
table2 %>% spread(key=type,value=count)
```

```
# A tibble: 6 x 4
      country  year  cases population
        <chr> <int>  <int>      <int>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3      Brazil  1999  37737  172006362
4      Brazil  2000  80488  174504898
5       China  1999 212258 1272915272
```

Exercise

다음 데이터를 깔끔하게 정리하라. spread 해야할까? gather해야할까?

```
preg <- tribble(
  ~pregnant, ~male, ~female,
  "yes",     NA,    10,
  "no",      20,    12
)
preg
```

```
# A tibble: 2 x 3
  pregnant  male female
     <chr> <dbl>  <dbl>
1      yes    NA     10
2       no    20     12
```

Answer

```
preg %>% gather(key="sex",value="n",male,female)
```

```
# A tibble: 4 x 3
  pregnant    sex     n
     <chr>  <chr> <dbl>
1      yes   male    NA
2       no   male    20
3      yes female    10
4       no female    12
```

Separating과 uniting

Separate

```
table3
```

```
# A tibble: 6 x 3
      country  year              rate
        <chr> <int>             <chr>
1 Afghanistan  1999      745/19987071
2 Afghanistan  2000     2666/20595360
3      Brazil  1999   37737/172006362
4      Brazil  2000   80488/174504898
5       China  1999 212258/1272915272
6       China  2000 213766/1280428583
```

table3에는 한 열(rate)에 두 개의 변수(cases와 population)가 포함되어 있다. 이 경우 separate()함수를 써서 분리할 수 있다.

country	year	rate
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

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

```
table3 %>%
    separate(rate, into = c("cases", "population"))
```

```
# A tibble: 6 x 4
      country  year  cases population
        <chr> <int>  <chr>      <chr>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3      Brazil  1999  37737  172006362
4      Brazil  2000  80488  174504898
5       China  1999 212258 1272915272
6       China  2000 213766 1280428583
```

separate()

```
table3 %>%
    separate(rate,into = c("cases", "population"),sep="/",convert=TRU
```

```
# A tibble: 6 x 4
      country  year  cases population
        <chr> <int>  <int>      <int>
1 Afghanistan  1999    745   19987071
2 Afghanistan  2000   2666   20595360
3      Brazil  1999  37737  172006362
4      Brazil  2000  80488  174504898
5       China  1999 212258 1272915272
6       China  2000 213766 1280428583
```

- 디폴트 값으로 sep는 알파벳 또는 숫자가 아닌 값으로 되어 있으며 sep인수로 지정할 수 있다.
- 분리된 열의 데이터 타입은 현재의 데이터 타입으로 바뀌나 convert=TRUE로 지정해주면 가장 알맞는 데이터 타입으로 바뀐다.

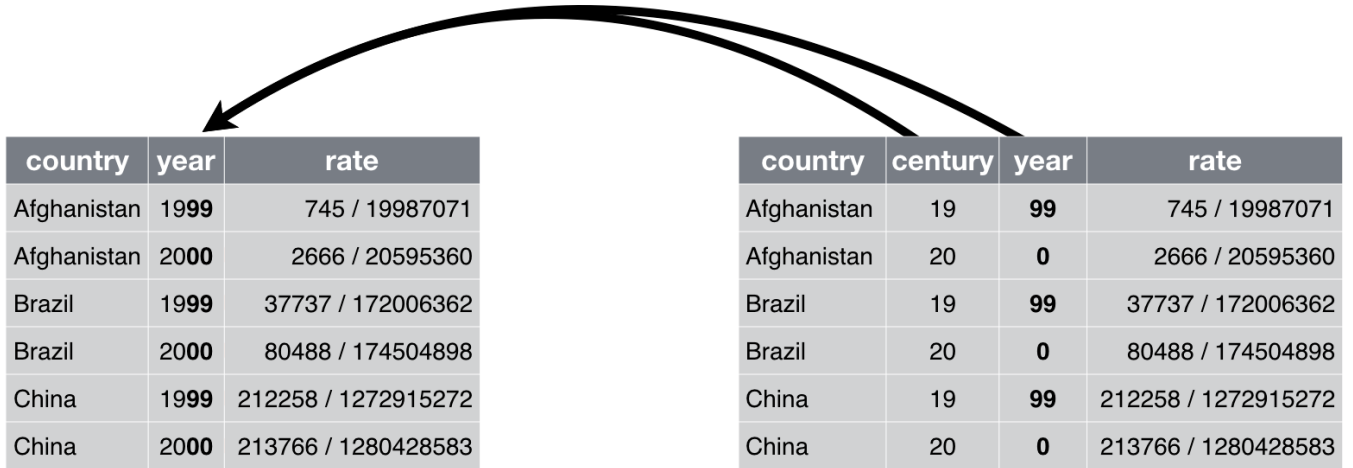
- sep 인수에 숫자를 지정할 경우 분리할 위치로 해석한다. 양수는 문자의 왼쪽부터 시작하고 음수는 문자의 오른쪽에서 시작한다. 예를들어 연도를 century와 year로 분리하려면 다음과 같이 한다.

```
table3 %>%
  separate(year, into = c("century", "year"), sep = 2)
```

```
# A tibble: 6 x 4
      country century  year              rate
        <chr>   <chr> <chr>             <chr>
1 Afghanistan      19    99      745/19987071
2 Afghanistan      20    00     2666/20595360
3      Brazil      19    99   37737/172006362
4      Brazil      20    00   80488/174504898
5       China      19    99 212258/1272915272
6       China      20    00 213766/1280428583
```

unite

unite()함수는 separate()함수의 반대이다.



country	year	rate
Afghanistan	19**99**	745 / 19987071
Afghanistan	20**00**	2666 / 20595360
Brazil	19**99**	37737 / 172006362
Brazil	20**00**	80488 / 174504898
China	19**99**	212258 / 1272915272
China	20**00**	213766 / 1280428583

country	century	year	rate
Afghanistan	19	**99**	745 / 19987071
Afghanistan	20	**0**	2666 / 20595360
Brazil	19	**99**	37737 / 172006362
Brazil	20	**0**	80488 / 174504898
China	19	**99**	212258 / 1272915272
China	20	**0**	213766 / 1280428583

table6

```
unite(data,col,...,sep="_",remove=TRUE)
```

```
table5
```

```
# A tibble: 6 x 4
      country century  year              rate
        <chr>   <chr> <chr>             <chr>
1 Afghanistan      19    99      745/19987071
2 Afghanistan      20    00     2666/20595360
3      Brazil      19    99   37737/172006362
4      Brazil      20    00   80488/174504898
5       China      19    99 212258/1272915272
6       China      20    00 213766/1280428583
```

```
table5 %>% unite(new, century, year)
```

```
# A tibble: 6 x 3
      country   new              rate
        <chr> <chr>             <chr>
1 Afghanistan 19_99      745/19987071
2 Afghanistan 20_00     2666/20595360
3      Brazil 19_99   37737/172006362
4      Brazil 20_00   80488/174504898
5       China 19_99 212258/1272915272
6       China 20_00 213766/1280428583
```

```
table5 %>%
  unite(new, century, year,sep="")
```

```
# A tibble: 6 x 3
      country   new              rate
        <chr> <chr>             <chr>
1 Afghanistan  1999      745/19987071
2 Afghanistan  2000     2666/20595360
3      Brazil  1999   37737/172006362
4      Brazil  2000   80488/174504898
5       China  1999 212258/1272915272
6       China  2000 213766/1280428583
```

Exercises

```
(tbl <- tibble(x = c("a,b,c", "d,e,f,g", "h,i,j")) )
```

```
# A tibble: 3 x 1
         x
     <chr>
1    a,b,c
2 d,e,f,g
3    h,i,j
```

Exercises

```
(tbl <- tibble(x = c("a,b,c", "d,e,f,g", "h,i,j")) )
```

```
# A tibble: 3 x 1
        x
    <chr>
1   a,b,c
2 d,e,f,g
3   h,i,j
```

```
?separate
```

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, **extra** = **"warn"**, **fill** = **"warn"**, ...)

```
tbl %>% separate(x,c("one","two","three"))
```

```
# A tibble: 3 x 3
    one   two three
  <chr> <chr> <chr>
1     a     b     c
2     d     e     f
3     h     i     j
```

```
tbl %>% separate(x,c("one","two","three"),extra="merge")
```

```
# A tibble: 3 x 3
    one   two three
  <chr> <chr> <chr>
1     a     b     c
2     d     e   f,g
3     h     i     j
```

```
(tbl2 <- tibble(x = c("a,b,c", "d,e", "f,g,i")))
```

```
# A tibble: 3 x 1
      x
  <chr>
1 a,b,c
2   d,e
3 f,g,i
```

```
tbl2 %>% separate(x, c("one", "two", "three"))
```

```
# A tibble: 3 x 3
    one   two three
  <chr> <chr> <chr>
1     a     b     c
2     d     e  <NA>
3     f     g     i
```

```
tbl2 %>% separate(x, c("one", "two", "three"),fill="left")
```

```
# A tibble: 3 x 3
    one   two three
  <chr> <chr> <chr>
1     a     b     c
2  <NA>     d     e
3     f     g     i
```

Missing Values

```
(stocks <- tibble(
  year   = c(2015, 2015, 2015, 2015, 2016, 2016, 2016),
  qtr    = c(   1,    2,    3,    4,    2,    3,    4),
  return = c(1.88, 0.59, 0.35,   NA, 0.92, 0.17, 2.66)
))
```

```
# A tibble: 7 x 3
   year   qtr return
  <dbl> <dbl>  <dbl>
1  2015     1   1.88
2  2015     2   0.59
3  2015     3   0.35
4  2015     4     NA
5  2016     2   0.92
6  2016     3   0.17
7  2016     4   2.66
```

Missing Values

```
(stocks <- tibble(
  year   = c(2015, 2015, 2015, 2015, 2016, 2016, 2016),
  qtr    = c(   1,    2,    3,    4,    2,    3,    4),
  return = c(1.88, 0.59, 0.35,   NA, 0.92, 0.17, 2.66)
))
```

```
# A tibble: 7 x 3
   year   qtr return
  <dbl> <dbl>  <dbl>
1  2015     1   1.88
2  2015     2   0.59
3  2015     3   0.35
4  2015     4     NA
5  2016     2   0.92
6  2016     3   0.17
7  2016     4   2.66
```

이 자료는 두 종류의 누락치가 있다.

- 명시적인 누락: 2015년도 4분기 return이 누락되어 있다.
- 암묵적인 누락: 2016년 1분기가 통째로 빠져있다.

```
stocks %>%
    spread(year,return)
```

```
# A tibble: 4 x 3
    qtr `2015` `2016`
  <dbl>  <dbl>  <dbl>
1     1   1.88     NA
2     2   0.59   0.92
3     3   0.35   0.17
4     4     NA   2.66
```

```
stocks %>%
    spread(year,return)
```

```
# A tibble: 4 x 3
    qtr `2015` `2016`
  <dbl>  <dbl>  <dbl>
1     1   1.88     NA
2     2   0.59   0.92
3     3   0.35   0.17
4     4     NA   2.66
```

```
stocks %>%
    spread(year,return) %>%
    gather(year,return,-qtr)
```

```
# A tibble: 8 x 3
    qtr  year return
  <dbl> <chr>  <dbl>
1     1  2015   1.88
2     2  2015   0.59
3     3  2015   0.35
4     4  2015     NA
5     1  2016     NA
6     2  2016   0.92
7     3  2016   0.17
8     4  2016   2.66
```

```
stocks %>%
    spread(year,return) %>%
    gather(year,return,-qtr,na.rm=TRUE)
```

```
# A tibble: 6 x 3
    qtr  year return
  <dbl> <chr>  <dbl>
1     1  2015   1.88
2     2  2015   0.59
3     3  2015   0.35
4     2  2016   0.92
5     3  2016   0.17
6     4  2016   2.66
```

```
stocks %>%
    complete(year,qtr)
```

```
# A tibble: 8 x 3
   year   qtr return
  <dbl> <dbl>  <dbl>
1  2015     1   1.88
2  2015     2   0.59
3  2015     3   0.35
4  2015     4     NA
5  2016     1     NA
6  2016     2   0.92
7  2016     3   0.17
8  2016     4   2.66
```

complete() 함수는 열이름들을 받아들여 모든 가능한 조합을 만들어 누락치가 있으면 명시적으로 NA를 표시해준다.

fill()

값의 중복을 피하기 위해 값이 바뀔 때만 기록한 자료가 있다.

```
treatment <- tribble(
  ~ person,           ~ treatment, ~response,
  "Derrick Whitmore", 1,           7,
  NA,                 2,           10,
  NA,                 3,           9,
  "Katherine Burke",  1,           4
)
treatment
```

```
# A tibble: 4 x 3
            person treatment response
             <chr>     <dbl>    <dbl>
1 Derrick Whitmore         1        7
2             <NA>         2       10
3             <NA>         3        9
4  Katherine Burke         1        4
```

```
treatment %>%
    fill(person)
```

```
# A tibble: 4 x 3
            person treatment response
             <chr>     <dbl>    <dbl>
1 Derrick Whitmore         1        7
2 Derrick Whitmore         2       10
3 Derrick Whitmore         3        9
4  Katherine Burke         1        4
```

```
treatment %>%
    fill(person)
```

```
# A tibble: 4 x 3
            person treatment response
             <chr>     <dbl>    <dbl>
1 Derrick Whitmore         1        7
2 Derrick Whitmore         2       10
3 Derrick Whitmore         3        9
4  Katherine Burke         1        4
```

```
treatment %>%
    fill(person,.direction="up")
```

```
# A tibble: 4 x 3
            person treatment response
             <chr>     <dbl>    <dbl>
1 Derrick Whitmore         1        7
2  Katherine Burke         2       10
3  Katherine Burke         3        9
4  Katherine Burke         1        4
```

Case Study

```
who
```

```
# A tibble: 7,240 x 60
       country  iso2  iso3  year new_sp_m014 new_sp_m1524 new_sp_m2534
         <chr> <chr> <chr> <int>       <int>        <int>        <int>
 1 Afghanistan    AF   AFG  1980          NA           NA           NA
 2 Afghanistan    AF   AFG  1981          NA           NA           NA
 3 Afghanistan    AF   AFG  1982          NA           NA           NA
 4 Afghanistan    AF   AFG  1983          NA           NA           NA
 5 Afghanistan    AF   AFG  1984          NA           NA           NA
 6 Afghanistan    AF   AFG  1985          NA           NA           NA
 7 Afghanistan    AF   AFG  1986          NA           NA           NA
 8 Afghanistan    AF   AFG  1987          NA           NA           NA
 9 Afghanistan    AF   AFG  1988          NA           NA           NA
10 Afghanistan    AF   AFG  1989          NA           NA           NA
# ... with 7,230 more rows, and 53 more variables: new_sp_m3544 <int>,
#   new_sp_m4554 <int>, new_sp_m5564 <int>, new_sp_m65 <int>,
#   new_sp_f014 <int>, new_sp_f1524 <int>, new_sp_f2534 <int>,
#   new_sp_f3544 <int>, new_sp_f4554 <int>, new_sp_f5564 <int>,
#   new_sp_f65 <int>, new_sn_m014 <int>, new_sn_m1524 <int>,
#   new_sn_m2534 <int>, new_sn_m3544 <int>, new_sn_m4554 <int>,
#   new_sn_m5564 <int>, new_sn_m65 <int>, new_sn_f014 <int>,
```

```
who1 <- who %>%
    gather(5:60,key="key",value="cases",na.rm=TRUE)
who1
```

```
# A tibble: 76,046 x 6
       country  iso2  iso3  year         key cases
 *       <chr> <chr> <chr> <int>       <chr> <int>
 1 Afghanistan    AF   AFG  1997 new_sp_m014     0
 2 Afghanistan    AF   AFG  1998 new_sp_m014    30
 3 Afghanistan    AF   AFG  1999 new_sp_m014     8
 4 Afghanistan    AF   AFG  2000 new_sp_m014    52
 5 Afghanistan    AF   AFG  2001 new_sp_m014   129
 6 Afghanistan    AF   AFG  2002 new_sp_m014    90
 7 Afghanistan    AF   AFG  2003 new_sp_m014   127
 8 Afghanistan    AF   AFG  2004 new_sp_m014   139
 9 Afghanistan    AF   AFG  2005 new_sp_m014   151
10 Afghanistan    AF   AFG  2006 new_sp_m014   193
# ... with 76,036 more rows
```

```
?who
unique(who1$key)
```

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

```
who2 <- who1 %>%
    mutate(key=stringr::str_replace(key,"newrel","new_rel"))
unique(who2$key)
```

```
 [1] "new_sp_m014"   "new_sp_m1524"  "new_sp_m2534"  "new_sp_m3544"
 [5] "new_sp_m4554"  "new_sp_m5564"  "new_sp_m65"    "new_sp_f014"
 [9] "new_sp_f1524"  "new_sp_f2534"  "new_sp_f3544"  "new_sp_f4554"
[13] "new_sp_f5564"  "new_sp_f65"    "new_sn_m014"   "new_sn_m1524"
[17] "new_sn_m2534"  "new_sn_m3544"  "new_sn_m4554"  "new_sn_m5564"
[21] "new_sn_m65"    "new_sn_f014"   "new_sn_f1524"  "new_sn_f2534"
[25] "new_sn_f3544"  "new_sn_f4554"  "new_sn_f5564"  "new_sn_f65"
[29] "new_ep_m014"   "new_ep_m1524"  "new_ep_m2534"  "new_ep_m3544"
[33] "new_ep_m4554"  "new_ep_m5564"  "new_ep_m65"    "new_ep_f014"
[37] "new_ep_f1524"  "new_ep_f2534"  "new_ep_f3544"  "new_ep_f4554"
[41] "new_ep_f5564"  "new_ep_f65"    "new_rel_m014"  "new_rel_m1524"
[45] "new_rel_m2534" "new_rel_m3544" "new_rel_m4554" "new_rel_m5564"
[49] "new_rel_m65"   "new_rel_f014"  "new_rel_f1524" "new_rel_f2534"
[53] "new_rel_f3544" "new_rel_f4554" "new_rel_f5564" "new_rel_f65"
```

```
who3 <- who2 %>%
    separate(key,c("new","type","sexage"))
who3
```

```
# A tibble: 76,046 x 8
       country  iso2  iso3  year   new  type sexage cases
 *       <chr> <chr> <chr> <int> <chr> <chr>  <chr> <int>
 1 Afghanistan    AF   AFG  1997   new    sp   m014     0
 2 Afghanistan    AF   AFG  1998   new    sp   m014    30
 3 Afghanistan    AF   AFG  1999   new    sp   m014     8
 4 Afghanistan    AF   AFG  2000   new    sp   m014    52
 5 Afghanistan    AF   AFG  2001   new    sp   m014   129
 6 Afghanistan    AF   AFG  2002   new    sp   m014    90
 7 Afghanistan    AF   AFG  2003   new    sp   m014   127
 8 Afghanistan    AF   AFG  2004   new    sp   m014   139
 9 Afghanistan    AF   AFG  2005   new    sp   m014   151
10 Afghanistan    AF   AFG  2006   new    sp   m014   193
# ... with 76,036 more rows
```

```
who3 %>% count(new)
```

```
# A tibble: 1 x 2
    new     n
  <chr> <int>
1   new 76046
```

```
who4 <- who3 %>%
    select(-new, -iso2, -iso3)
```

```r
who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)

who5
```

```
# A tibble: 76,046 x 6
       country  year  type   sex   age cases
 *       <chr> <int> <chr> <chr> <chr> <int>
 1 Afghanistan  1997    sp     m   014     0
 2 Afghanistan  1998    sp     m   014    30
 3 Afghanistan  1999    sp     m   014     8
 4 Afghanistan  2000    sp     m   014    52
 5 Afghanistan  2001    sp     m   014   129
 6 Afghanistan  2002    sp     m   014    90
 7 Afghanistan  2003    sp     m   014   127
 8 Afghanistan  2004    sp     m   014   139
 9 Afghanistan  2005    sp     m   014   151
10 Afghanistan  2006    sp     m   014   193
# ... with 76,036 more rows
```

```
who5 <-who %>%
  gather(code, value, new_sp_m014:newrel_f65, na.rm = TRUE) %>%
  mutate(code = stringr::str_replace(code, "newrel", "new_rel")) %>%
  separate(code, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1)
who5
```

```
# A tibble: 76,046 x 6
       country  year   var   sex   age value
 *       <chr> <int> <chr> <chr> <chr> <int>
 1 Afghanistan  1997    sp     m   014     0
 2 Afghanistan  1998    sp     m   014    30
 3 Afghanistan  1999    sp     m   014     8
 4 Afghanistan  2000    sp     m   014    52
 5 Afghanistan  2001    sp     m   014   129
 6 Afghanistan  2002    sp     m   014    90
 7 Afghanistan  2003    sp     m   014   127
 8 Afghanistan  2004    sp     m   014   139
 9 Afghanistan  2005    sp     m   014   151
10 Afghanistan  2006    sp     m   014   193
# ... with 76,036 more rows
```

Exercises

각 나이, 성별로 결핵 환자 전체 수를 구하라

```
who5 %>%
    group_by(sex,age) %>%
    summarise(total=sum(value)) %>%
    ggplot(aes(age,total,fill=sex)) + geom_col()
```