realation

박찬영

2024-08-26

tidyverse와 nycflights13 library를 사용합니다.

관계형 데이터

nycflights13에는 여러 데이터프레임이 존재한다.

```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1      517            515         2      830            819
##  2  2013     1     1      533            529         4      850            830
##  3  2013     1     1      542            540         2      923            850
##  4  2013     1     1      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # i 336,766 more rows
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
airlines #항공사 코드
```

```
## # A tibble: 16 x 2
##    carrier name
##    <chr>   <chr>
##  1 9E      Endeavor Air Inc.
##  2 AA      American Airlines Inc.
##  3 AS      Alaska Airlines Inc.
##  4 B6      JetBlue Airways
##  5 DL      Delta Air Lines Inc.
##  6 EV      ExpressJet Airlines Inc.
##  7 F9      Frontier Airlines Inc.
##  8 FL      AirTran Airways Corporation
##  9 HA      Hawaiian Airlines Inc.
## 10 MQ      Envoy Air
## 11 OO      SkyWest Airlines Inc.
## 12 UA      United Air Lines Inc.
## 13 US      US Airways Inc.
## 14 VX      Virgin America
## 15 WN      Southwest Airlines Co.
## 16 YV      Mesa Airlines Inc.
```

```
airports #공항 코드
```

```
## # A tibble: 1,458 x 8
##    faa   name                           lat    lon   alt    tz dst   tzone
##    <chr> <chr>                        <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 04G   Lansdowne Airport             41.1  -80.6  1044    -5 A     America/~
##  2 06A   Moton Field Municipal Airport  32.5  -85.7   264    -6 A     America/~
##  3 06C   Schaumburg Regional           42.0  -88.1   801    -6 A     America/~
##  4 06N   Randall Airport               41.4  -74.4   523    -5 A     America/~
##  5 09J   Jekyll Island Airport         31.1  -81.4    11    -5 A     America/~
##  6 0A9   Elizabethton Municipal Airport 36.4  -82.2  1593    -5 A     America/~
##  7 0G6   Williams County Airport       41.5  -84.5   730    -5 A     America/~
##  8 0G7   Finger Lakes Regional Airport  42.9  -76.8   492    -5 A     America/~
##  9 0P2   Shoestring Aviation Airfield  39.8  -76.6  1000    -5 U     America/~
## 10 0S9   Jefferson County Intl         48.1 -123.    108    -8 A     America/~
## # i 1,448 more rows
```

```
planes #여객기 코드
```

```
## # A tibble: 3,322 x 9
##    tailnum  year type              manufacturer model engines seats speed engine
##    <chr>   <int> <chr>             <chr>        <chr>   <int> <int> <int> <chr>
##  1 N10156   2004 Fixed wing multi~ EMBRAER      EMB-~       2    55    NA Turbo~
##  2 N102UW   1998 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  3 N103US   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  4 N104UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  5 N10575   2002 Fixed wing multi~ EMBRAER      EMB-~       2    55    NA Turbo~
##  6 N105UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  7 N107US   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  8 N108UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
##  9 N109UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## 10 N110UW   1999 Fixed wing multi~ AIRBUS INDU~ A320~       2   182    NA Turbo~
## # i 3,312 more rows
```

```
weather #공항 날씨
```

```
## # A tibble: 26,115 x 15
##    origin  year month   day  hour  temp  dewp humid wind_dir wind_speed
##    <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>
##  1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4
##  2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06
##  3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5
##  4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7
##  5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7
##  6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5
##  7 EWR     2013     1     1     7  39.0  28.0  64.4      240      15.0
##  8 EWR     2013     1     1     8  39.9  28.0  62.2      250      10.4
##  9 EWR     2013     1     1     9  39.9  28.0  62.2      260      15.0
## 10 EWR     2013     1     1    10  41    28.0  59.6      260      13.8
## # i 26,105 more rows
## # i 5 more variables: wind_gust <dbl>, precip <dbl>, pressure <dbl>,
## #   visib <dbl>, time_hour <dttm>
```

키

flights 는 여러가지 데이터들과 엮여있고 코드를 통해 식별된다. 두 데이터프레임을 연결하는 변수를 키라고 한다. 자신의 데이터를 고유하게 식별하는걸 기본키 라고한다. planes$tailnum은 기본키이다

다른 데이터를 고유하게 식별하면 외래키이다. flights$tailnum은 planes를 고유하게 식별하므로 외래키이다.

```
planes %>%
    count(tailnum) %>%
    filter(n>1) #tailnum 종류별로 셌을 때 2개 이상 세지지 않으므로 기본키
```

```
## # A tibble: 0 x 2
## # i 2 variables: tailnum <chr>, n <int>
```

```
flights %>%
    count(year, month, day, tailnum) %>%
    filter(n>1) #기본키 아님!
```

```
## # A tibble: 64,928 x 5
##     year month   day tailnum     n
##    <int> <int> <int> <chr>   <int>
##  1  2013     1     1 N0EGMQ      2
##  2  2013     1     1 N11189      2
##  3  2013     1     1 N11536      2
##  4  2013     1     1 N11544      3
##  5  2013     1     1 N11551      2
##  6  2013     1     1 N12540      2
##  7  2013     1     1 N12567      2
##  8  2013     1     1 N13123      2
##  9  2013     1     1 N13538      3
## 10  2013     1     1 N13566      3
## # i 64,918 more rows
```

```
#기본키를 만들고 싶기에 row_number를 이용해준다
#이렇게 만든 키를 대체키라고 한다
#대체 키를 만들면 데이터 변환 후 대조가 쉽다
```

조인

```
flights2 <- flights %>%
  select(year:day, hour, origin, dest, tailnum, carrier)
flights2
```

```
## # A tibble: 336,776 x 8
##     year month   day  hour origin dest  tailnum carrier
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>
##  1  2013     1     1     5 EWR    IAH   N14228  UA
##  2  2013     1     1     5 LGA    IAH   N24211  UA
##  3  2013     1     1     5 JFK    MIA   N619AA  AA
##  4  2013     1     1     5 JFK    BQN   N804JB  B6
##  5  2013     1     1     6 LGA    ATL   N668DN  DL
##  6  2013     1     1     5 EWR    ORD   N39463  UA
##  7  2013     1     1     6 EWR    FLL   N516JB  B6
##  8  2013     1     1     6 LGA    IAD   N829AS  EV
##  9  2013     1     1     6 JFK    MCO   N593JB  B6
## 10  2013     1     1     6 LGA    ORD   N3ALAA  AA
## # i 336,766 more rows
```

```
#쉬운 데이터를 하나 만들자

#여기에 airlines 데이터프레임을 추가하자
flights2 %>%
    select(-origin, -dest) %>%
    left_join(airlines, by="carrier")
```

```
## # A tibble: 336,776 x 7
##     year month   day  hour tailnum carrier name
##    <int> <int> <int> <dbl> <chr>   <chr>   <chr>
##  1  2013     1     1     5 N14228  UA      United Air Lines Inc.
##  2  2013     1     1     5 N24211  UA      United Air Lines Inc.
##  3  2013     1     1     5 N619AA  AA      American Airlines Inc.
##  4  2013     1     1     5 N804JB  B6      JetBlue Airways
##  5  2013     1     1     6 N668DN  DL      Delta Air Lines Inc.
##  6  2013     1     1     5 N39463  UA      United Air Lines Inc.
##  7  2013     1     1     6 N516JB  B6      JetBlue Airways
##  8  2013     1     1     6 N829AS  EV      ExpressJet Airlines Inc.
##  9  2013     1     1     6 N593JB  B6      JetBlue Airways
## 10  2013     1     1     6 N3ALAA  AA      American Airlines Inc.
## # i 336,766 more rows
```

```
#이러면 carrier에 대응하는 name열이 추가된다
#그래서 뮤테이팅 조인이다

flights2 %>%
  select(-origin, -dest) %>%
  mutate(name = airlines$name[match(carrier, airlines$carrier)])
```

```
## # A tibble: 336,776 x 7
##     year month   day  hour tailnum carrier name
##    <int> <int> <int> <dbl> <chr>   <chr>   <chr>
##  1  2013     1     1     5 N14228  UA      United Air Lines Inc.
##  2  2013     1     1     5 N24211  UA      United Air Lines Inc.
##  3  2013     1     1     5 N619AA  AA      American Airlines Inc.
##  4  2013     1     1     5 N804JB  B6      JetBlue Airways
##  5  2013     1     1     6 N668DN  DL      Delta Air Lines Inc.
##  6  2013     1     1     5 N39463  UA      United Air Lines Inc.
##  7  2013     1     1     6 N516JB  B6      JetBlue Airways
##  8  2013     1     1     6 N829AS  EV      ExpressJet Airlines Inc.
##  9  2013     1     1     6 N593JB  B6      JetBlue Airways
## 10  2013     1     1     6 N3ALAA  AA      American Airlines Inc.
## # i 336,766 more rows
```

```
#뮤테이트로 구현하기

#조인함수를 뜯어보자

x <- tribble(
  ~key, ~val_x,
     1, "x1",
     2, "x2",
     3, "x3"
)
y <- tribble(
  ~key, ~val_y,
     1, "y1",
     2, "y2",
     4, "y3"
)

inner_join(x, y , by="key")
```

```
## # A tibble: 2 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
```

```
#내부조인은 대응 안되는걸 없앤다
left_join(x,y,by="key")
```

```
## # A tibble: 3 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
```

```r
#좌측조인은 왼쪽데이터에서 대응 안되는걸 살린다
right_join(x,y,by="key")
```

```
## # A tibble: 3 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     4 <NA>  y3
```

```r
#우측 조인은 반대
full_join(x,y,by="key")
```

```
## # A tibble: 4 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     3 x3    <NA>
## 4     4 <NA>  y3
```

```r
#전체조인은 대응 안되는 걸 다 살린다

#기본적으로 다른 데이터프레임에서 가져오는 경우가 많아서
#좌측 조인을 많이 쓴다



x <- tribble(
  ~key, ~val_x,
     1, "x1",
     2, "x2",
     2, "x3",
     1, "x4"
)
y <- tribble(
  ~key, ~val_y,
     1, "y1",
     2, "y2"
)
```

```r
left_join(x,y,by="key")
```

```
## # A tibble: 4 x 3
##     key val_x val_y
##   <dbl> <chr> <chr>
## 1     1 x1    y1
## 2     2 x2    y2
## 3     2 x3    y2
## 4     1 x4    y1
```

```r
#키가 중복될경우 가능한 경우의 수를 다 보여준다
#이는 데카르트 곱이다

flights2 %>% left_join(weather)
```

```
## Joining with `by = join_by(year, month, day, hour, origin)`

## # A tibble: 336,776 x 18
##     year month   day  hour origin dest  tailnum carrier  temp  dewp humid
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>   <dbl> <dbl> <dbl>
##  1  2013     1     1     5 EWR    IAH   N14228  UA       39.0  28.0  64.4
##  2  2013     1     1     5 LGA    IAH   N24211  UA       39.9  25.0  54.8
##  3  2013     1     1     5 JFK    MIA   N619AA  AA       39.0  27.0  61.6
##  4  2013     1     1     5 JFK    BQN   N804JB  B6       39.0  27.0  61.6
##  5  2013     1     1     6 LGA    ATL   N668DN  DL       39.9  25.0  54.8
##  6  2013     1     1     5 EWR    ORD   N39463  UA       39.0  28.0  64.4
##  7  2013     1     1     6 EWR    FLL   N516JB  B6       37.9  28.0  67.2
##  8  2013     1     1     6 LGA    IAD   N829AS  EV       39.9  25.0  54.8
##  9  2013     1     1     6 JFK    MCO   N593JB  B6       37.9  27.0  64.3
## 10  2013     1     1     6 LGA    ORD   N3ALAA  AA       39.9  25.0  54.8
## # i 336,766 more rows
## # i 7 more variables: wind_dir <dbl>, wind_speed <dbl>, wind_gust <dbl>,
## #   precip <dbl>, pressure <dbl>, visib <dbl>, time_hour <dttm>
```

```r
#by를 안주면 알아서 판단함
#year month day hour origin으로 맞춰줌

flights2 %>% left_join(airports, c("dest"="faa"))
```

```
## # A tibble: 336,776 x 15
##     year month   day  hour origin dest  tailnum carrier name      lat   lon   alt
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>   <chr>   <dbl> <dbl> <dbl>
##  1  2013     1     1     5 EWR    IAH   N14228  UA      Georg~  30.0 -95.3    97
##  2  2013     1     1     5 LGA    IAH   N24211  UA      Georg~  30.0 -95.3    97
##  3  2013     1     1     5 JFK    MIA   N619AA  AA      Miami~  25.8 -80.3     8
##  4  2013     1     1     5 JFK    BQN   N804JB  B6      <NA>    NA    NA      NA
##  5  2013     1     1     6 LGA    ATL   N668DN  DL      Harts~  33.6 -84.4  1026
##  6  2013     1     1     5 EWR    ORD   N39463  UA      Chica~  42.0 -87.9   668
##  7  2013     1     1     6 EWR    FLL   N516JB  B6      Fort ~  26.1 -80.2     9
##  8  2013     1     1     6 LGA    IAD   N829AS  EV      Washi~  38.9 -77.5   313
##  9  2013     1     1     6 JFK    MCO   N593JB  B6      Orlan~  28.4 -81.3    96
## 10  2013     1     1     6 LGA    ORD   N3ALAA  AA      Chica~  42.0 -87.9   668
## # i 336,766 more rows
## # i 3 more variables: tz <dbl>, dst <chr>, tzone <chr>
```

```
#dest에 faa를 결합해서 만든다 (할당 연산자)

#필터링 조인을 해보자

top_dest = flights2 %>% count(dest, sort=TRUE) %>% head(10)

top_dest #상위 10개 목적지
```

```
## # A tibble: 10 x 2
##    dest      n
##    <chr> <int>
##  1 ORD   17283
##  2 ATL   17215
##  3 LAX   16174
##  4 BOS   15508
##  5 MCO   14082
##  6 CLT   14064
##  7 SFO   13331
##  8 FLL   12055
##  9 MIA   11728
## 10 DCA    9705
```

```
#이다음에 원하는거만 남길 수 있음
flights2 %>% filter(dest %in% top_dest$dest)
```

```
## # A tibble: 141,145 x 8
##      year month   day  hour origin dest  tailnum carrier
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>
##  1  2013     1     1     5 JFK    MIA   N619AA  AA
##  2  2013     1     1     6 LGA    ATL   N668DN  DL
##  3  2013     1     1     5 EWR    ORD   N39463  UA
##  4  2013     1     1     6 EWR    FLL   N516JB  B6
##  5  2013     1     1     6 JFK    MCO   N593JB  B6
##  6  2013     1     1     6 LGA    ORD   N3ALAA  AA
##  7  2013     1     1     6 JFK    LAX   N29129  UA
##  8  2013     1     1     6 EWR    SFO   N53441  UA
##  9  2013     1     1     5 JFK    BOS   N708JB  B6
## 10  2013     1     1     6 LGA    FLL   N595JB  B6
## # i 141,135 more rows
```

```
#상위 10개 놈들만 남길 수 있다


#이걸 간단히 하는게 semi_join이다


flights2 %>% semi_join(top_dest)
```

```
## Joining with `by = join_by(dest)`
```

```
## # A tibble: 141,145 x 8
##     year month   day  hour origin dest  tailnum carrier
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>
##  1  2013     1     1     5 JFK    MIA   N619AA  AA
##  2  2013     1     1     6 LGA    ATL   N668DN  DL
##  3  2013     1     1     5 EWR    ORD   N39463  UA
##  4  2013     1     1     6 EWR    FLL   N516JB  B6
##  5  2013     1     1     6 JFK    MCO   N593JB  B6
##  6  2013     1     1     6 LGA    ORD   N3ALAA  AA
##  7  2013     1     1     6 JFK    LAX   N29129  UA
##  8  2013     1     1     6 EWR    SFO   N53441  UA
##  9  2013     1     1     5 JFK    BOS   N708JB  B6
## 10  2013     1     1     6 LGA    FLL   N595JB  B6
## # i 141,135 more rows
```

```
#세미조인은 열을 추가하는 것이 아닌 겹치는 데이터를 보존한다.
#매칭 되기만 하면 남긴다
# 그 반대는 안티조인이다
flights2 %>% anti_join(top_dest)
```

```
## Joining with `by = join_by(dest)`
## # A tibble: 195,631 x 8
##     year month   day  hour origin dest  tailnum carrier
##    <int> <int> <int> <dbl> <chr>  <chr> <chr>   <chr>
##  1  2013     1     1     5 EWR    IAH   N14228  UA
##  2  2013     1     1     5 LGA    IAH   N24211  UA
##  3  2013     1     1     5 JFK    BQN   N804JB  B6
##  4  2013     1     1     6 LGA    IAD   N829AS  EV
##  5  2013     1     1     6 JFK    PBI   N793JB  B6
##  6  2013     1     1     6 JFK    TPA   N657JB  B6
##  7  2013     1     1     6 LGA    DFW   N3DUAA  AA
##  8  2013     1     1     6 EWR    LAS   N76515  UA
##  9  2013     1     1     6 EWR    PBI   N644JB  B6
## 10  2013     1     1     6 LGA    MSP   N971DL  DL
## # i 195,621 more rows
```

```
#상위 10개 도착지 빼고 남기기

#anti 조인은 조인이 안되는 놈들을 찾기 좋다

flights2 %>% anti_join(planes, by="tailnum") %>% count(tailnum, sort=TRUE)
```

```
## # A tibble: 722 x 2
##    tailnum     n
##    <chr>   <int>
##  1 <NA>     2512
##  2 N725MQ    575
##  3 N722MQ    513
##  4 N723MQ    507
##  5 N713MQ    483
##  6 N735MQ    396
##  7 N0EGMQ    371
##  8 N534MQ    364
##  9 N542MQ    363
## 10 N531MQ    349
## # i 712 more rows
```

```
#여기 남은 놈들은 planes에 등록되지 않은 비행기들이다
```

집합 연산

```
#데이터 프레임끼리 집합연산이 가능하다
df1 <- tribble(
  ~x, ~y,
   1,  1,
   2,  1
)
df2 <- tribble(
  ~x, ~y,
   1,  1,
   1,  2
)

intersect(df1, df2)
```

```
## # A tibble: 1 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     1
```

```
union(df1, df2)
```

```
## # A tibble: 3 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     1
## 2     2     1
## 3     1     2
```

```
setdiff(df1, df2)
```

```
## # A tibble: 1 x 2
##       x     y
##   <dbl> <dbl>
## 1     2     1
```

```
setdiff(df2, df1)
```

```
## # A tibble: 1 x 2
##       x     y
##   <dbl> <dbl>
## 1     1     2
```

```
#교집합, 합집합, 차집합 다 잘 된다
```