Rmarkdown & Tidyverse

Geonwoo Ban

2021 6 25

해당 문서는 책 R for data science(Hadley Wickham, Garrett Grolemund / O`REILLY)와 부산대학교 통계학과 통계적계산방법(박소영 교수님) 수업 내용을 참고하여 교육목적으로 만들어 졌으며, 무단 배포는 금지합니다.

Rmarkdown

cheat sheet (https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf)

- 보고서 제출이나 논문, 분석발표시에 매우 유용한 R의 기능 중 하나
- 코드와 결과를 포함하여 하나의 문서로 저장해줌
- 유사한 기능으로는 Python의 jupiter가 있음
- 기초적인 내용은 위 cheat sheet에 있고, 이걸 가능하면 프린트해서 들고다니는 것 추천
- 앞으로 모든 코딩이나 분석 결과는 Rmarkdown과 유사한 형식으로 저장하여 **Github**에 올릴 것
- Markdown은 기본적으로 word나 html과 같이 문구를 받는 입력칸과 코딩을 받는 입력칸으로 나뉘어져 있음

```
library(rmarkdown)
```

- 유용한 단축키
  - `Ctrl+Alt+i` : 코딩 입력칸 생성
  - `Ctrl+Shift+M` : tidyverse를 공부하며 많이 사용할 파이프 연산자 단축키 (%>%)
  - `Ctrl+Shift+K` : 출력 단축키
- 문구 입력칸 정보
  - `#` 으로 header나 글자 크기 조절
  - `**` 으로 *italics* 폰트
  - `****` 으로 **bold** 처리
  - link나 image 삽입 기능은 많이 사용하므로 cheat sheet 참고
  - `>` 으로 소주제나 보고서 작성 시 문단 나누기
  - 수식 또한 많이 사용하는데 사용하고자 하는 수식은 google에 물어보기
  - LaTex Symbols (https://strikers01.tistory.com/445)
  - 최종 결과물을 출력할 때는 `Knit` 버튼누르면 끝 (또는 단축키 `Ctrl+Shift+K`)
- 코딩 입력칸 정보(Chunk)
  - 코딩 입력칸에서는 출력물에 코드 및 결과를 어떻게 입력할지가 관건, 코딩에 있어서 관련된 기능은 모름

- `eval=T` : 코드를 실행시키지 않고 코드만 출력, 결과 출력 x (결과는 상관없이 코드를 보여주고 싶을 때)
- `echo=F` : 코드 출력 x, 결과만 출력 (코드는 상관없이 결과만 보여주고 싶을 때)
- `warning=F` : 코드를 실행하며 발생하는 warning은 출력 x (결과에 warning을 출력하고 싶지 않을 때)
- `error=F` : error 출력 x (결과에 error를 출력하고 싶지 않을 때)
- `message=F` : 결과를 제외한 다른 message들 출력 x (결과를 제외한 다른 message를 출력하고 싶지 않을 때)

Tidyverse

tidyverse (https://www.tidyverse.org/)

- Data를 가지고 목적에 맞추어 변형시키고 가공시키는 과정과 분석을 통하여 결론을 내는 과정을 일컫는 **data mining**을 함에 있어 매우 유용한 패키지
- 주로 Data를 목적에 맞추어 가공하는 data handling에 있어 유용

Data

```
#install.packages("nycflights13")
library(nycflights13)
library(tidyverse)

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
## # ... with 336,766 more rows, and 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>
```

- flights : 2013년에 관측된 NYC에서의 비행기 정보
- data를 보면 뭐부터?
    - head : 데이터가 실제 어떤 값들으로 이루어져있는지
    - dim : 변수의 개수와 관측치의 개수 파악
    - summary : 각 변수의 성질과 변환 계획잡을 때 필수

```
flights %>% head
```

```
## # A tibble: 6 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
## # ... with 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>
```

```
flights %>% dim
```

```
## [1] 336776     19
```

```
flights %>% summary
```

```
##       year          month             day           dep_time    sched_dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1   Min.   : 106
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907   1st Qu.: 906
##  Median :2013   Median : 7.000   Median :16.00   Median :1401   Median :1359
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349   Mean   :1344
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744   3rd Qu.:1729
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400   Max.   :2359
##                                                  NA's   :8255
##    dep_delay          arr_time    sched_arr_time   arr_delay
##  Min.   : -43.00   Min.   :   1   Min.   :   1   Min.   : -86.000
##  1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124   1st Qu.: -17.000
##  Median :  -2.00   Median :1535   Median :1556   Median :  -5.000
##  Mean   :  12.64   Mean   :1502   Mean   :1536   Mean   :   6.895
##  3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945   3rd Qu.:  14.000
##  Max.   :1301.00   Max.   :2400   Max.   :2359   Max.   :1272.000
##  NA's   :8255      NA's   :8713                  NA's   :9430
##    carrier              flight       tailnum             origin
##  Length:336776      Min.   :   1   Length:336776      Length:336776
##  Class :character   1st Qu.: 553   Class :character   Class :character
##  Mode  :character   Median :1496   Mode  :character   Mode  :character
##                     Mean   :1972
##                     3rd Qu.:3465
##                     Max.   :8500
##
##      dest              air_time        distance         hour
##  Length:336776      Min.   : 20.0   Min.   :  17   Min.   : 1.00
##  Class :character   1st Qu.: 82.0   1st Qu.: 502   1st Qu.: 9.00
##  Mode  :character   Median :129.0   Median : 872   Median :13.00
##                     Mean   :150.7   Mean   :1040   Mean   :13.18
##                     3rd Qu.:192.0   3rd Qu.:1389   3rd Qu.:17.00
##                     Max.   :695.0   Max.   :4983   Max.   :23.00
##                     NA's   :9430
##      minute        time_hour
##  Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :26.23   Mean   :2013-07-03 05:22:54
##  3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
##  Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

- 각자 summary를 통해서 변수 특성 파악해보기(변수 변환할 계획이 있다면 계획과 결과 예상하기)
    - ex.A변수와 B변수의 의미 중복으로 B변수 제외하기
- 데이터 다룰 때는 항상 자신이 데이터에 대해서 잘 알아야함
- 변수 변환할 때도 그냥 바꿔보는게 아니라 근거가 있어야하고 기대 효과도 생각해야함

type of variable

- `int` : integers
- `dbl` : doubles or real numbers
- `chr` : character vectors or strings
- `dttm` : date + time
- `lgl` : logical
- `fctr` : factors, categorical

basics function of data hadling in dplyr

- `filter()` : 원하는 조건을 만족하는 **관측치** 뽑기
- `arrange()` : 행을 재정렬
- `select()` : 사용하고자하는 **변수**만 선택
- `mutate()` : 원래 존재하던 변수들을 가지고 새로운 변수를 만듦
- `summarise()` : 원래 존재하던 변수들에 대한 기초통계 및 수학적 대표치를 계산하여 저장
- `group_by()` : 변수들에 대하여 그룹을 지정하여 추가적인 계산에 대한 기준을 제공

Filter rows with `filter()`

```
#filter(flights, month==1, day==1) %>% head(10)

flights %>% filter(month==1, day==1) %>% head(10)
```

```
## # A tibble: 10 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
## # ... with 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>
```

```
(dec25 <- flights %>% filter(month == 12, day == 25))
```

```
## # A tibble: 719 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    12    25      456            500        -4      649            651
##  2  2013    12    25      524            515         9      805            814
##  3  2013    12    25      542            540         2      832            850
##  4  2013    12    25      546            550        -4     1022           1027
##  5  2013    12    25      556            600        -4      730            745
##  6  2013    12    25      557            600        -3      743            752
##  7  2013    12    25      557            600        -3      818            831
##  8  2013    12    25      559            600        -1      855            856
##  9  2013    12    25      559            600        -1      849            855
## 10  2013    12    25      600            600         0      850            846
## # ... with 709 more rows, and 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>
```

```
flights %>% filter(month==11 | month==12) %>% head(5)
```

```
## # A tibble: 5 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    11     1        5           2359         6      352            345
## 2  2013    11     1       35           2250       105      123           2356
## 3  2013    11     1      455            500        -5      641            651
## 4  2013    11     1      539            545        -6      856            827
## 5  2013    11     1      542            545        -3      831            855
## # ... with 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>
```

```
flights %>% filter(month==11 & day==12) %>% head(5)
```

```
## # A tibble: 5 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    11    12      455            500        -5      639            651
## 2  2013    11    12      515            515         0      803            808
## 3  2013    11    12      540            545        -5      818            835
## 4  2013    11    12      540            545        -5      828            855
## 5  2013    11    12      547            600       -13      700            736
## # ... with 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>
```

```
flights %>% filter(!(month==11 | day==12)) %>% head(5)
```

```
## # A tibble: 5 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
## # ... with 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>
```

Your turn 1

1. 출발시간이 2시간 이상 지연된 비행기는?
2. 출발은 지연되지 않았는데 도착시간이 지연된 비행기는?
3. 출발시간이 1시간 이상 지연되었지만 도착시간은 지연된 시간보다 30분이상 일찍 도착한 비행기는?

- ex. 8시 출발 -> 10시 도착 비행기편이지만 출발시간이 지연되어 9시에 출발하여 원래 11시에 도착해야하지만 10시반에 도착한 경우

4. `is.na()` 를 사용하여 dep_time 변수에 대한 missing value 뽑은 다음 이 변수가 missing이면 다른 어떤 변수도 missing인지 확인 후 이러한 경우는 어떤 것을 의미하는지?

Arrange rows with `arrange()`

```
flights %>% arrange(dep_delay)
```

```
## # 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    12     7     2040           2123       -43       40           2352
##  2  2013     2     3     2022           2055       -33     2240           2338
##  3  2013    11    10     1408           1440       -32     1549           1559
##  4  2013     1    11     1900           1930       -30     2233           2243
##  5  2013     1    29     1703           1730       -27     1947           1957
##  6  2013     8     9      729            755       -26     1002            955
##  7  2013    10    23     1907           1932       -25     2143           2143
##  8  2013     3    30     2030           2055       -25     2213           2250
##  9  2013     3     2     1431           1455       -24     1601           1631
## 10  2013     5     5      934            958       -24     1225           1309
## # ... with 336,766 more rows, and 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>
```

```
flights %>% arrange(desc(dep_delay))
```

```
## # 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     9      641            900      1301     1242           1530
##  2  2013     6    15     1432           1935      1137     1607           2120
##  3  2013     1    10     1121           1635      1126     1239           1810
##  4  2013     9    20     1139           1845      1014     1457           2210
##  5  2013     7    22      845           1600      1005     1044           1815
##  6  2013     4    10     1100           1900       960     1342           2211
##  7  2013     3    17     2321            810       911      135           1020
##  8  2013     6    27      959           1900       899     1236           2226
##  9  2013     7    22     2257            759       898      121           1026
## 10  2013    12     5      756           1700       896     1058           2020
## # ... with 336,766 more rows, and 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>
```

```
flights %>% arrange(year, month, desc(day))
```

```
## # 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    31        1           2100       181      124           2225
##  2  2013     1    31        4           2359         5      455            444
##  3  2013     1    31        7           2359         8      453            437
##  4  2013     1    31       12           2250        82      132              7
##  5  2013     1    31       26           2154       152      328             50
##  6  2013     1    31       34           2159       155      135           2315
##  7  2013     1    31       37           2249       108      132           2357
##  8  2013     1    31       54           2250       124      152           2359
##  9  2013     1    31      453            500        -7      651            648
## 10  2013     1    31      522            525        -3      820            820
## # ... with 336,766 more rows, and 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>
```

```
# order by year, month, day
```

Your turn 2

1. 가장 출발시간이 지연된 비행기편으로 정렬 후, 가장 일찍 출발한 비행기는?
2. 비행기의 속력이 가장 빠른 비행기는?

Select columns with `select()`

```
flights %>% select(year, month, day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
flights %>% select(year:day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
flights %>% select(-(year:day))
```

```
## # A tibble: 336,776 x 16
##    dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier
##       <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>
##  1      517            515         2      830            819        11 UA
##  2      533            529         4      850            830        20 UA
##  3      542            540         2      923            850        33 AA
##  4      544            545        -1     1004           1022       -18 B6
##  5      554            600        -6      812            837       -25 DL
##  6      554            558        -4      740            728        12 UA
##  7      555            600        -5      913            854        19 B6
##  8      557            600        -3      709            723       -14 EV
##  9      557            600        -3      838            846        -8 B6
## 10      558            600        -2      753            745         8 AA
## # ... with 336,766 more rows, and 9 more variables: flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

- `select()` 의 변수명에 대한 추가적인 제약 옵션
    - `starts_with("abc")` : 변수명이 “abc”로 시작하는 변수들을 모두 선택
    - `ends_with("xyz")` : 변수명이 “xyz”로 끝나는 변수들을 모두 선택
    - `contains("abc")` : 변수명이 “abc”를 포함하는 변수들을 모두 선택
    - `matches("[A-z]+")` : 변수명이 “정규표현식(regular expression)”을 만족하는 변수들을 모두 선택

```
flights %>% select(starts_with("arr"))
```

```
## # A tibble: 336,776 x 2
##    arr_time arr_delay
##       <int>     <dbl>
##  1      830        11
##  2      850        20
##  3      923        33
##  4     1004       -18
##  5      812       -25
##  6      740        12
##  7      913        19
##  8      709       -14
##  9      838        -8
## 10      753         8
## # ... with 336,766 more rows
```

```
flights %>% select(ends_with("time"))
```

```
## # A tibble: 336,776 x 5
##    dep_time sched_dep_time arr_time sched_arr_time air_time
##       <int>          <int>    <int>          <int>    <dbl>
##  1      517            515      830            819      227
##  2      533            529      850            830      227
##  3      542            540      923            850      160
##  4      544            545     1004           1022      183
##  5      554            600      812            837      116
##  6      554            558      740            728      150
##  7      555            600      913            854      158
##  8      557            600      709            723       53
##  9      557            600      838            846      140
## 10      558            600      753            745      138
## # ... with 336,766 more rows
```

```
flights %>% select(contains("dep"))
```

```
## # A tibble: 336,776 x 3
##    dep_time sched_dep_time dep_delay
##       <int>          <int>     <dbl>
##  1      517            515         2
##  2      533            529         4
##  3      542            540         2
##  4      544            545        -1
##  5      554            600        -6
##  6      554            558        -4
##  7      555            600        -5
##  8      557            600        -3
##  9      557            600        -3
## 10      558            600        -2
## # ... with 336,766 more rows
```

Your turn 3

1. dep_time, dep_delay, arr_time, arr_delay 이 네 변수를 한번에 select 함수로 뽑으려면 가장 간단한 방법 스스로 생각하고 해보기

Add new variables with `mutate()`

```
flights %>% select(year:day, ends_with("delay"), distance, air_time) -> flights2

flights2 %>%
  mutate(gain = dep_delay-arr_delay,
         speed = distance/air_time*60)
```

```
## # A tibble: 336,776 x 9
##     year month   day dep_delay arr_delay distance air_time  gain speed
##    <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
##  1  2013     1     1         2        11     1400      227    -9  370.
##  2  2013     1     1         4        20     1416      227   -16  374.
##  3  2013     1     1         2        33     1089      160   -31  408.
##  4  2013     1     1        -1       -18     1576      183    17  517.
##  5  2013     1     1        -6       -25      762      116    19  394.
##  6  2013     1     1        -4        12      719      150   -16  288.
##  7  2013     1     1        -5        19     1065      158   -24  404.
##  8  2013     1     1        -3       -14      229       53    11  259.
##  9  2013     1     1        -3        -8      944      140     5  405.
## 10  2013     1     1        -2         8      733      138   -10  319.
## # ... with 336,766 more rows
```

```
flights2 %>%
  transmute(gain = dep_delay-arr_delay,
         speed = distance/air_time*60)
```

```
## # A tibble: 336,776 x 2
##     gain speed
##    <dbl> <dbl>
##  1    -9  370.
##  2   -16  374.
##  3   -31  408.
##  4    17  517.
##  5    19  394.
##  6   -16  288.
##  7   -24  404.
##  8    11  259.
##  9     5  405.
## 10   -10  319.
## # ... with 336,766 more rows
```

Your turn 4

1. dep_time과 sched_dep_time 저장형태를 잘 확인 후, 이를 각각 시 와 분을 구분하여 새로운 변수를 만들기.
2. air_time과 arr_time - dep_time 간의 비교했을 때, 자신의 견해 쓰기
3. dep_time, sched_dep_time 그리고 dep_delay 변수를 비교했을 때, 자신의 견해 쓰기

Grouped summaries with `summarise()`

```
flights %>% summarise(delay=mean(dep_delay, na.rm=T))
```

```
## # A tibble: 1 x 1
##   delay
##   <dbl>
## 1  12.6
```

```
flights %>%
  group_by(year, month, day) %>%
  summarise(delay=mean(dep_delay, na.rm=T))
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the `.groups` argum
ent.
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [12]
##     year month   day delay
##    <int> <int> <int> <dbl>
##  1  2013     1     1 11.5
##  2  2013     1     2 13.9
##  3  2013     1     3 11.0
##  4  2013     1     4  8.95
##  5  2013     1     5  5.73
##  6  2013     1     6  7.15
##  7  2013     1     7  5.42
##  8  2013     1     8  2.55
##  9  2013     1     9  2.28
## 10  2013     1    10  2.84
## # ... with 355 more rows
```

```
flights %>%
  group_by(year, month, day) %>%
  summarise(delay=mean(dep_delay, na.rm=T), n = n())
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the `.groups` argum
ent.
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [12]
##     year month   day delay     n
##    <int> <int> <int> <dbl> <int>
##  1  2013     1     1 11.5    842
##  2  2013     1     2 13.9    943
##  3  2013     1     3 11.0    914
##  4  2013     1     4  8.95   915
##  5  2013     1     5  5.73   720
##  6  2013     1     6  7.15   832
##  7  2013     1     7  5.42   933
##  8  2013     1     8  2.55   899
##  9  2013     1     9  2.28   902
## 10  2013     1    10  2.84   932
## # ... with 355 more rows
```

Your turn 5

1. 항공사 별 평균 출발지연시간과 출발지연시간의 표준편차를 계산 후, 평균이 가장 큰 항공사와 표준편차가 가장 큰 항공사 파악하기
2. 월과 일 별 평균 출발지연시간을 계산 후, 가장 지연이 길었던 날짜를 찾으시오
3. (2)번에서의 결과를 가지고 해당 날짜에 대하여 가장 작은 출발지연시간 표준편차를 가지는 항공사는?