lec1028

박효선 1585063

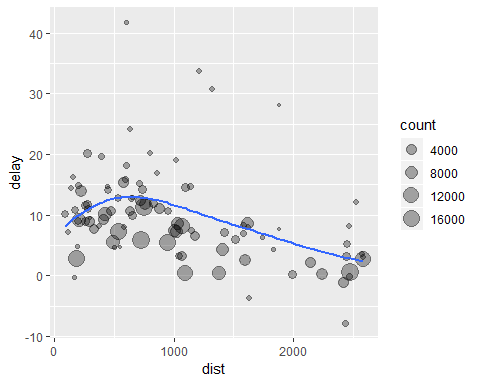
2018년 12월 3일

###############  
# 10 월 28 일 #  
###############  
# 물어볼 것 : count(wt) 누적합인가?  
  
# Combining multiple operations with the pipe  
# • 각 지역에서 distance 와 average delay 의 관계 살펴보기.  
  
by\_dest <- group\_by(flights, dest) # 도착지별로 group  
delay <- summarise(by\_dest, # 도착지별로  
 count = n(), # 비행건수  
 dist = mean(distance, na.rm = TRUE), # 뉴욕-목적지 거리, NA처리  
 delay = mean(arr\_delay, na.rm = TRUE)) # 도착지연 평균  
delay <- filter(delay, count > 20, dest != "HNL") # 비행건수 20 이상이고 호놀롤루 가아닌 것 만 뽑아내기

## Warning: package 'bindrcpp' was built under R version 3.5.1

ggplot(data = delay, aes(x = dist, y = delay)) + # 거리- 연착시간 산점도  
 geom\_point(aes(size = count), alpha = 1/3) +  
 geom\_smooth(se = FALSE) # 파이프를 통해 ggplot 동시에 쓸 수 이으

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



# Missing values   
# na.rm = TRUE 옵션은 통계량 계산 전에 NA 를 모두 제거한 후 계산  
flights %>%  
 group\_by(year, month, day) %>%  
 summarise(mean = mean(dep\_delay)) # na.rm = 옵션을 안주면 NA 값으로 계산된다.

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day mean  
## <int> <int> <int> <dbl>  
## 1 2013 1 1 NA  
## 2 2013 1 2 NA  
## 3 2013 1 3 NA  
## 4 2013 1 4 NA  
## 5 2013 1 5 NA  
## 6 2013 1 6 NA  
## 7 2013 1 7 NA  
## 8 2013 1 8 NA  
## 9 2013 1 9 NA  
## 10 2013 1 10 NA  
## # ... with 355 more rows

flights %>%  
 group\_by(year, month, day) %>%  
 summarise(mean = mean(dep\_delay, na.rm = TRUE))

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day mean  
## <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

# 아래와 같이 계산해도 같은 결과  
# dep\_delay, arr\_delay 가 missin기g value 가 아닌 데이터 추출  
not\_cancelled <- flights %>% # 아니면 아예 필터링 해버리기  
 filter(!is.na(dep\_delay), !is.na(arr\_delay)) # 지연정보가 NA인 행 제외  
not\_cancelled %>%   
 group\_by(year, month, day) %>%   
 summarise(mean = mean(dep\_delay))

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day mean  
## <int> <int> <int> <dbl>  
## 1 2013 1 1 11.4   
## 2 2013 1 2 13.7   
## 3 2013 1 3 10.9   
## 4 2013 1 4 8.97  
## 5 2013 1 5 5.73  
## 6 2013 1 6 7.15  
## 7 2013 1 7 5.42  
## 8 2013 1 8 2.56  
## 9 2013 1 9 2.30  
## 10 2013 1 10 2.84  
## # ... with 355 more rows

# Counts   
# n() : 총 자료의 갯수  
# sum(!is.na(x)) : missing 이 아닌 자료의 갯수  
# (TRUE/ FALSE 로 나오니까. TRUE 값만 합쳐진다)  
  
x <- c(1,2,3,NA,5)  
is.na(x); sum(is.na(x))

## [1] FALSE FALSE FALSE TRUE FALSE

## [1] 1

!is.na(x); sum(!is.na(x)) # NA 값이 아닌 자료의 개수는 4 개 이다

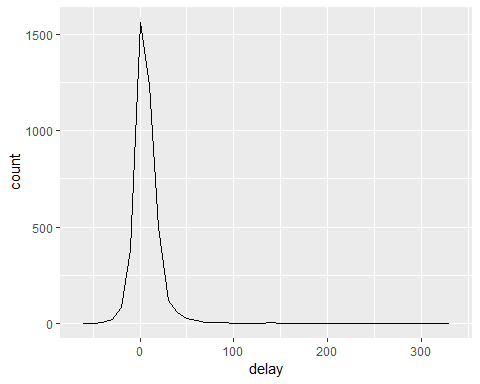
## [1] TRUE TRUE TRUE FALSE TRUE

## [1] 4

# delay 분포 살펴보기  
# 비행기별(tailnum)로 도착딜레이의 평균을 구하기  
delays <- not\_cancelled %>%   
 group\_by(tailnum) %>% # 비행기별  
 summarise(delay = mean(arr\_delay)) # 도착지연 평균  
delays

## # A tibble: 4,037 x 2  
## tailnum delay  
## <chr> <dbl>  
## 1 D942DN 31.5   
## 2 N0EGMQ 9.98   
## 3 N10156 12.7   
## 4 N102UW 2.94   
## 5 N103US -6.93   
## 6 N104UW 1.80   
## 7 N10575 20.7   
## 8 N105UW -0.267  
## 9 N107US -5.73   
## 10 N108UW -1.25   
## # ... with 4,027 more rows

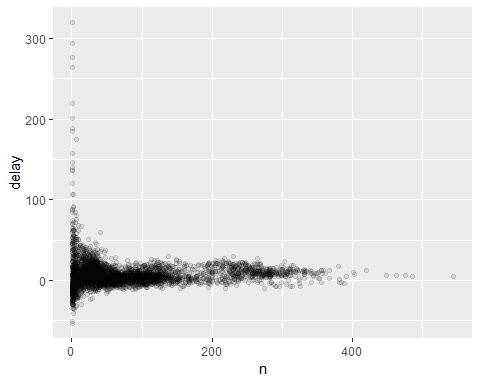
# 300분 이상의 평균 delay 인 비행기가 있음.  
ggplot(delays, aes(delay)) + geom\_freqpoly(binwidth = 10)



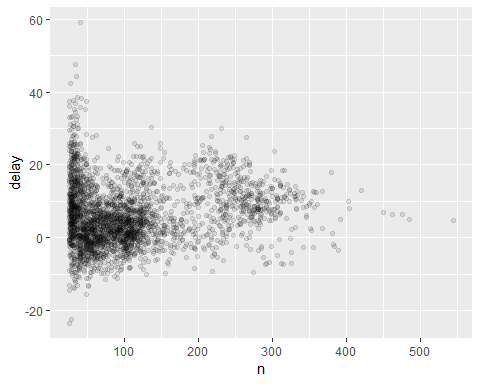
# 좀 더 자세히 살펴보기 위해 아래와 같이 delay 와 비행건수 간의 산점도를 그림  
# 한번 늦었는데 많이 늦은 경우도 있으니까... delay 건수 적은데 많이 늦은 케이스 찾아보기  
delays <- not\_cancelled %>%   
 group\_by(tailnum) %>% # 비행기 별로  
 summarise(delay = mean(arr\_delay, na.rm = TRUE), # 도착 연착의 평균  
 n = n()) # 운항 건수  
delays

## # A tibble: 4,037 x 3  
## tailnum delay n  
## <chr> <dbl> <int>  
## 1 D942DN 31.5 4  
## 2 N0EGMQ 9.98 352  
## 3 N10156 12.7 145  
## 4 N102UW 2.94 48  
## 5 N103US -6.93 46  
## 6 N104UW 1.80 46  
## 7 N10575 20.7 269  
## 8 N105UW -0.267 45  
## 9 N107US -5.73 41  
## 10 N108UW -1.25 60  
## # ... with 4,027 more rows

#평균 300 이상이긴 하지만, 비행건수가 매우 작음을 알 수 있다.  
# 0 근처에 너무 많네..  
ggplot(delays, aes(n, delay)) + geom\_point(alpha = 0.1)



# 비행건수 25 이상만을 그림  
# 많아봐야 60 분 연착된다. # 간단한 ggplot   
delays %>% filter(n > 25) %>%   
 ggplot(aes(n, delay)) +   
 geom\_point(alpha = 0.1)



# Useful summary functions  
# mean(), median()  
# delay 음수인 애들은 지연이 안된애들임. 실제로 지연된 애들만 고려하자  
not\_cancelled %>% filter(arr\_delay > 0) %>% # 실제로 지연된 것만  
 select(year:day, arr\_delay)

## # A tibble: 133,004 x 4  
## year month day arr\_delay  
## <int> <int> <int> <dbl>  
## 1 2013 1 1 11  
## 2 2013 1 1 20  
## 3 2013 1 1 33  
## 4 2013 1 1 12  
## 5 2013 1 1 19  
## 6 2013 1 1 8  
## 7 2013 1 1 7  
## 8 2013 1 1 31  
## 9 2013 1 1 12  
## 10 2013 1 1 16  
## # ... with 132,994 more rows

not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별  
 summarise(avg\_delay1 = mean(arr\_delay), # arr\_delay 의 평균  
 avg\_delay2 = mean(arr\_delay[arr\_delay > 0])) # 연착한 것만 가지고 arr\_delay 평균

## # A tibble: 365 x 5  
## # Groups: year, month [?]  
## year month day avg\_delay1 avg\_delay2  
## <int> <int> <int> <dbl> <dbl>  
## 1 2013 1 1 12.7 32.5  
## 2 2013 1 2 12.7 32.0  
## 3 2013 1 3 5.73 27.7  
## 4 2013 1 4 -1.93 28.3  
## 5 2013 1 5 -1.53 22.6  
## 6 2013 1 6 4.24 24.4  
## 7 2013 1 7 -4.95 27.8  
## 8 2013 1 8 -3.23 20.8  
## 9 2013 1 9 -0.264 25.6  
## 10 2013 1 10 -5.90 27.3  
## # ... with 355 more rows

not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별  
 summarise(avg\_delay1 = mean(arr\_delay), # 모든 운항기록의 arr\_delay 의 평균  
 n\_delay1 = n(), # 모든 운항건수  
 avg\_delay2 = mean(arr\_delay[arr\_delay > 0]), # 지연도착한 비행 arr\_delay 의 평균 (중요)  
 n\_delay2 = sum(arr\_delay > 0)) # 지연도착한 비행 건수 -> arr\_delay > 0 건수 (중요)

## # A tibble: 365 x 7  
## # Groups: year, month [?]  
## year month day avg\_delay1 n\_delay1 avg\_delay2 n\_delay2  
## <int> <int> <int> <dbl> <int> <dbl> <int>  
## 1 2013 1 1 12.7 831 32.5 461  
## 2 2013 1 2 12.7 928 32.0 535  
## 3 2013 1 3 5.73 900 27.7 460  
## 4 2013 1 4 -1.93 908 28.3 297  
## 5 2013 1 5 -1.53 717 22.6 238  
## 6 2013 1 6 4.24 829 24.4 381  
## 7 2013 1 7 -4.95 930 27.8 243  
## 8 2013 1 8 -3.23 892 20.8 275  
## 9 2013 1 9 -0.264 893 25.6 287  
## 10 2013 1 10 -5.90 929 27.3 220  
## # ... with 355 more rows

# sd(), IQR(), mad()  
# 목적지 별로 거리의 표준편차 구하기. 내림차순으로 정렬  
not\_cancelled %>% group\_by(dest) %>% # 목적지 별로  
 summarise(distance\_sd = sd(distance)) %>% # 거리의 표준편차  
 arrange(desc(distance\_sd)) # 내림차순 정렬

## # A tibble: 104 x 2  
## dest distance\_sd  
## <chr> <dbl>  
## 1 EGE 10.5   
## 2 SAN 10.4   
## 3 SFO 10.2   
## 4 HNL 10.0   
## 5 SEA 9.98  
## 6 LAS 9.91  
## 7 PDX 9.87  
## 8 PHX 9.86  
## 9 LAX 9.66  
## 10 IND 9.46  
## # ... with 94 more rows

# sd=0 아닌 애들은 출발하는 공항이 여러개인 목적지  
not\_cancelled %>% group\_by(dest) %>% # 목적지별로  
 summarise(distance\_sd = sd(distance)) %>% # 거리의 표준편차  
 arrange(distance\_sd) # 오름차순 정렬

## # A tibble: 104 x 2  
## dest distance\_sd  
## <chr> <dbl>  
## 1 ABQ 0  
## 2 ACK 0  
## 3 ALB 0  
## 4 ANC 0  
## 5 BDL 0  
## 6 BGR 0  
## 7 BUR 0  
## 8 BZN 0  
## 9 CAK 0  
## 10 CHO 0  
## # ... with 94 more rows

# 목적지 별로 거리의 표준편차와 거리의 평균을 구하기  
# 거리평균 기준으로 내림차순 정렬  
not\_cancelled %>% group\_by(dest) %>% # 목적지별로  
 summarise(distance\_sd = sd(distance), # 거리의 표준편차  
 distance\_mean = mean(distance)) %>% # 거리의 평균  
 arrange(desc(distance\_mean)) # 내림차순 정렬(큰 순서대로)

## # A tibble: 104 x 3  
## dest distance\_sd distance\_mean  
## <chr> <dbl> <dbl>  
## 1 HNL 10.0 4973.  
## 2 ANC 0 3370   
## 3 SFO 10.2 2578.  
## 4 OAK 0 2576   
## 5 SJC 0 2569   
## 6 SMF 0 2521   
## 7 LAX 9.66 2469.  
## 8 BUR 0 2465   
## 9 LGB 0 2465   
## 10 PDX 9.87 2446.  
## # ... with 94 more rows

not\_cancelled %>% group\_by(dest) %>%   
 summarise(distance\_sd = sd(distance)) %>%   
 arrange(distance\_sd)

## # A tibble: 104 x 2  
## dest distance\_sd  
## <chr> <dbl>  
## 1 ABQ 0  
## 2 ACK 0  
## 3 ALB 0  
## 4 ANC 0  
## 5 BDL 0  
## 6 BGR 0  
## 7 BUR 0  
## 8 BZN 0  
## 9 CAK 0  
## 10 CHO 0  
## # ... with 94 more rows

not\_cancelled %>% group\_by(dest) %>%   
 summarise(distance\_sd = sd(distance),  
 distance\_mean = mean(distance)) %>%   
 arrange(desc(distance\_mean))

## # A tibble: 104 x 3  
## dest distance\_sd distance\_mean  
## <chr> <dbl> <dbl>  
## 1 HNL 10.0 4973.  
## 2 ANC 0 3370   
## 3 SFO 10.2 2578.  
## 4 OAK 0 2576   
## 5 SJC 0 2569   
## 6 SMF 0 2521   
## 7 LAX 9.66 2469.  
## 8 BUR 0 2465   
## 9 LGB 0 2465   
## 10 PDX 9.87 2446.  
## # ... with 94 more rows

# min(), quantile(x, 25), max()  
# 일별로 가장 먼저, 가장 늦게 출발하는 시간은?  
not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로  
 summarise(first = min(dep\_time), # 출발시간의 최솟값 (하루중 첫 비행기)  
 last = max(dep\_time)) # 도착시간의 최솟값 (하루중 마지막 비행기)

## # A tibble: 365 x 5  
## # Groups: year, month [?]  
## year month day first last  
## <int> <int> <int> <dbl> <dbl>  
## 1 2013 1 1 517 2356  
## 2 2013 1 2 42 2354  
## 3 2013 1 3 32 2349  
## 4 2013 1 4 25 2358  
## 5 2013 1 5 14 2357  
## 6 2013 1 6 16 2355  
## 7 2013 1 7 49 2359  
## 8 2013 1 8 454 2351  
## 9 2013 1 9 2 2252  
## 10 2013 1 10 3 2320  
## # ... with 355 more rows

# first(), nth(x, 2), last() 정렬이 되어 있어야 한다.   
not\_cancelled %>% group\_by(year, month, day) %>%   
 summarise(first\_dep = first(dep\_time), # first() : 첫번째 값  
 last\_dep = last(dep\_time)) # last() : 자료의 마지막 값

## # A tibble: 365 x 5  
## # Groups: year, month [?]  
## year month day first\_dep last\_dep  
## <int> <int> <int> <int> <int>  
## 1 2013 1 1 517 2356  
## 2 2013 1 2 42 2354  
## 3 2013 1 3 32 2349  
## 4 2013 1 4 25 2358  
## 5 2013 1 5 14 2357  
## 6 2013 1 6 16 2355  
## 7 2013 1 7 49 2359  
## 8 2013 1 8 454 2351  
## 9 2013 1 9 2 2252  
## 10 2013 1 10 3 2320  
## # ... with 355 more rows

not\_cancelled %>% group\_by(year, month, day) %>%   
 arrange(desc(dep\_time)) %>% ### 오..정렬!!!!!!! 그래서 첫비행과 마지막비행 달라짐  
 summarise(first\_dep = first(dep\_time), # first() : 첫번째 값, 각 날짜별로 첫 비행시간  
 last\_dep = last(dep\_time)) # last() : 자료의 마지막 값

## # A tibble: 365 x 5  
## # Groups: year, month [?]  
## year month day first\_dep last\_dep  
## <int> <int> <int> <int> <int>  
## 1 2013 1 1 2356 517  
## 2 2013 1 2 2354 42  
## 3 2013 1 3 2349 32  
## 4 2013 1 4 2358 25  
## 5 2013 1 5 2357 14  
## 6 2013 1 6 2355 16  
## 7 2013 1 7 2359 49  
## 8 2013 1 8 2351 454  
## 9 2013 1 9 2252 2  
## 10 2013 1 10 2320 3  
## # ... with 355 more rows

# 출발시간이 늦은 것부터 순위를 매긴다.  
not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로  
 mutate(r1 = min\_rank(desc(dep\_time)), # 첫 비행기   
 r2 = min\_rank(dep\_time)) %>%   
 select(year, month, day, dep\_time, r1, r2)

## # A tibble: 327,346 x 6  
## # Groups: year, month, day [365]  
## year month day dep\_time r1 r2  
## <int> <int> <int> <int> <int> <int>  
## 1 2013 1 1 517 831 1  
## 2 2013 1 1 533 830 2  
## 3 2013 1 1 542 829 3  
## 4 2013 1 1 544 828 4  
## 5 2013 1 1 554 826 5  
## 6 2013 1 1 554 826 5  
## 7 2013 1 1 555 825 7  
## 8 2013 1 1 557 823 8  
## 9 2013 1 1 557 823 8  
## 10 2013 1 1 558 818 10  
## # ... with 327,336 more rows

##### 위와 비교?? 중요!!!!  
# 하루중 첫번째 출발과 마지막 출발한 비행을 뽑은 것이다!  
not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로  
 mutate(r = min\_rank(desc(dep\_time))) %>% # 순위매겨서 첫 비행기와 마지막 비행기 뽑아내기  
 filter(r %in% range(r)) %>% # range() 첫값과 끝값 반환 1월1일 1~831   
 select(year:day, dep\_time, r)

## # A tibble: 770 x 5  
## # Groups: year, month, day [365]  
## year month day dep\_time r  
## <int> <int> <int> <int> <int>  
## 1 2013 1 1 517 831  
## 2 2013 1 1 2356 1  
## 3 2013 1 2 42 928  
## 4 2013 1 2 2354 1  
## 5 2013 1 3 32 900  
## 6 2013 1 3 2349 1  
## 7 2013 1 4 25 908  
## 8 2013 1 4 2358 1  
## 9 2013 1 4 2358 1  
## 10 2013 1 5 14 717  
## # ... with 760 more rows

# n() 자료의 갯수  
# 목적지 별로 얼마나 갔는지  
not\_cancelled %>% group\_by(dest) %>% # 목적지 별로   
 summarise(n = n()) # 운항건수

## # A tibble: 104 x 2  
## dest n  
## <chr> <int>  
## 1 ABQ 254  
## 2 ACK 264  
## 3 ALB 418  
## 4 ANC 8  
## 5 ATL 16837  
## 6 AUS 2411  
## 7 AVL 261  
## 8 BDL 412  
## 9 BGR 358  
## 10 BHM 269  
## # ... with 94 more rows

# 인덱싱으로 filter 해보기  
flights[flights$dest == "ANC",] # flights %>% filter(dest == "ANC")

## # A tibble: 8 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 7 6 1629 1615 14 1954  
## 2 2013 7 13 1618 1615 3 1955  
## 3 2013 7 20 1618 1615 3 2003  
## 4 2013 7 27 1617 1615 2 1906  
## 5 2013 8 3 1615 1615 0 2003  
## 6 2013 8 10 1613 1615 -2 1922  
## 7 2013 8 17 1740 1625 75 2042  
## 8 2013 8 24 1633 1625 8 1959  
## # ... with 12 more variables: sched\_arr\_time <int>, 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>

not\_cancelled[not\_cancelled$dest == "ANC", ] # not\_cancelled %>% filter(dest == "ANC")

## # A tibble: 8 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 7 6 1629 1615 14 1954  
## 2 2013 7 13 1618 1615 3 1955  
## 3 2013 7 20 1618 1615 3 2003  
## 4 2013 7 27 1617 1615 2 1906  
## 5 2013 8 3 1615 1615 0 2003  
## 6 2013 8 10 1613 1615 -2 1922  
## 7 2013 8 17 1740 1625 75 2042  
## 8 2013 8 24 1633 1625 8 1959  
## # ... with 12 more variables: sched\_arr\_time <int>, 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>

# 목적지별로 운항하는 항공사 보기  
flights[flights$dest == "ANC", ]$carrier # ANC로 간 항공사를 보면 UA 뿐임

## [1] "UA" "UA" "UA" "UA" "UA" "UA" "UA" "UA"

table(not\_cancelled[not\_cancelled$dest == "AVL",]$carrier) # AVL로 간 항공사는 9E, EV

##   
## 9E EV   
## 10 251

table(not\_cancelled[not\_cancelled$dest == "ATL", ]$carrier)

##   
## 9E DL EV FL MQ UA WN   
## 56 10452 1656 2278 2235 102 58

# 가장 다양한 항공사에서 노선을 제공하고 있는 목적지는?  
not\_cancelled %>% group\_by(dest) %>% # 목적지별로  
 summarise(carriers = n\_distinct(carrier), # 운항하는 항공사 갯수  
 n = n()) # 운항건수

## # A tibble: 104 x 3  
## dest carriers n  
## <chr> <int> <int>  
## 1 ABQ 1 254  
## 2 ACK 1 264  
## 3 ALB 1 418  
## 4 ANC 1 8  
## 5 ATL 7 16837  
## 6 AUS 6 2411  
## 7 AVL 2 261  
## 8 BDL 2 412  
## 9 BGR 1 358  
## 10 BHM 1 269  
## # ... with 94 more rows

not\_cancelled %>% group\_by(dest) %>% #목적지별로   
 summarise(carriers = n\_distinct(carrier),  
 n = n()) %>%   
 arrange(desc(carriers)) # 항공사가 제일 많은 순서대로 정렬

## # A tibble: 104 x 3  
## dest carriers n  
## <chr> <int> <int>  
## 1 ATL 7 16837  
## 2 BOS 7 15022  
## 3 CLT 7 13674  
## 4 ORD 7 16566  
## 5 TPA 7 7390  
## 6 AUS 6 2411  
## 7 DCA 6 9111  
## 8 DTW 6 9031  
## 9 IAD 6 5383  
## 10 MSP 6 6929  
## # ... with 94 more rows

# 가장 많은 비행기(운항이)가 제공되는 목적지는?   
not\_cancelled %>% count(dest) %>% # 목적지별로  
 arrange(desc(n)) # 운항건수가 제일 많은 순서대로 정렬

## # A tibble: 104 x 2  
## dest n  
## <chr> <int>  
## 1 ATL 16837  
## 2 ORD 16566  
## 3 LAX 16026  
## 4 BOS 15022  
## 5 MCO 13967  
## 6 CLT 13674  
## 7 SFO 13173  
## 8 FLL 11897  
## 9 MIA 11593  
## 10 DCA 9111  
## # ... with 94 more rows

as.data.frame(table(not\_cancelled$dest))

## Var1 Freq  
## 1 ABQ 254  
## 2 ACK 264  
## 3 ALB 418  
## 4 ANC 8  
## 5 ATL 16837  
## 6 AUS 2411  
## 7 AVL 261  
## 8 BDL 412  
## 9 BGR 358  
## 10 BHM 269  
## 11 BNA 6084  
## 12 BOS 15022  
## 13 BQN 888  
## 14 BTV 2510  
## 15 BUF 4570  
## 16 BUR 370  
## 17 BWI 1687  
## 18 BZN 35  
## 19 CAE 106  
## 20 CAK 842  
## 21 CHO 46  
## 22 CHS 2759  
## 23 CLE 4394  
## 24 CLT 13674  
## 25 CMH 3326  
## 26 CRW 134  
## 27 CVG 3725  
## 28 DAY 1399  
## 29 DCA 9111  
## 30 DEN 7169  
## 31 DFW 8388  
## 32 DSM 523  
## 33 DTW 9031  
## 34 EGE 207  
## 35 EYW 17  
## 36 FLL 11897  
## 37 GRR 728  
## 38 GSO 1492  
## 39 GSP 790  
## 40 HDN 14  
## 41 HNL 701  
## 42 HOU 2083  
## 43 IAD 5383  
## 44 IAH 7085  
## 45 ILM 107  
## 46 IND 1981  
## 47 JAC 21  
## 48 JAX 2623  
## 49 LAS 5952  
## 50 LAX 16026  
## 51 LEX 1  
## 52 LGB 661  
## 53 MCI 1885  
## 54 MCO 13967  
## 55 MDW 4025  
## 56 MEM 1686  
## 57 MHT 932  
## 58 MIA 11593  
## 59 MKE 2709  
## 60 MSN 556  
## 61 MSP 6929  
## 62 MSY 3715  
## 63 MTJ 14  
## 64 MVY 210  
## 65 MYR 58  
## 66 OAK 309  
## 67 OKC 315  
## 68 OMA 817  
## 69 ORD 16566  
## 70 ORF 1434  
## 71 PBI 6487  
## 72 PDX 1342  
## 73 PHL 1541  
## 74 PHX 4606  
## 75 PIT 2746  
## 76 PSE 358  
## 77 PSP 18  
## 78 PVD 358  
## 79 PWM 2288  
## 80 RDU 7770  
## 81 RIC 2346  
## 82 ROC 2358  
## 83 RSW 3502  
## 84 SAN 2709  
## 85 SAT 659  
## 86 SAV 749  
## 87 SBN 10  
## 88 SDF 1104  
## 89 SEA 3885  
## 90 SFO 13173  
## 91 SJC 328  
## 92 SJU 5773  
## 93 SLC 2451  
## 94 SMF 282  
## 95 SNA 812  
## 96 SRQ 1201  
## 97 STL 4142  
## 98 STT 518  
## 99 SYR 1707  
## 100 TPA 7390  
## 101 TUL 294  
## 102 TVC 95  
## 103 TYS 578  
## 104 XNA 992

# weighted count 도 가능 : 가중치가 적용된 빈도?  
# 비행기별 누적 운항 거리. the total miles flown by each tail number.  
not\_cancelled %>% count(tailnum) # 비행기별 운항건수

## # A tibble: 4,037 x 2  
## tailnum n  
## <chr> <int>  
## 1 D942DN 4  
## 2 N0EGMQ 352  
## 3 N10156 145  
## 4 N102UW 48  
## 5 N103US 46  
## 6 N104UW 46  
## 7 N10575 269  
## 8 N105UW 45  
## 9 N107US 41  
## 10 N108UW 60  
## # ... with 4,027 more rows

not\_cancelled %>% count(tailnum, wt = distance) # 비행기별 누적 운항 거리

## # A tibble: 4,037 x 2  
## tailnum n  
## <chr> <dbl>  
## 1 D942DN 3418  
## 2 N0EGMQ 239143  
## 3 N10156 109664  
## 4 N102UW 25722  
## 5 N103US 24619  
## 6 N104UW 24616  
## 7 N10575 139903  
## 8 N105UW 23618  
## 9 N107US 21677  
## 10 N108UW 32070  
## # ... with 4,027 more rows

not\_cancelled %>% count(tailnum, wt = distance) %>% arrange(desc(n)) # 내림차순

## # A tibble: 4,037 x 2  
## tailnum n  
## <chr> <dbl>  
## 1 N328AA 929090  
## 2 N338AA 921172  
## 3 N335AA 902271  
## 4 N327AA 900482  
## 5 N323AA 839468  
## 6 N319AA 837924  
## 7 N336AA 833136  
## 8 N329AA 825826  
## 9 N324AA 786159  
## 10 N339AA 783648  
## # ... with 4,027 more rows

not\_cancelled %>% count(tailnum, wt = distance) %>% arrange(n) # 오름차순

## # A tibble: 4,037 x 2  
## tailnum n  
## <chr> <dbl>  
## 1 N505SW 185  
## 2 N746SK 229  
## 3 N881AS 292  
## 4 N824AS 296  
## 5 N701SK 419  
## 6 N702SK 419  
## 7 N705SK 419  
## 8 N710SK 419  
## 9 N726SK 419  
## 10 N740SK 419  
## # ... with 4,027 more rows

flights %>% group\_by(tailnum) %>%   
 summarise(sum(distance)) %>% arrange(desc(`sum(distance)`))

## # A tibble: 4,044 x 2  
## tailnum `sum(distance)`  
## <chr> <dbl>  
## 1 <NA> 1784167  
## 2 N328AA 939101  
## 3 N338AA 931183  
## 4 N327AA 915665  
## 5 N335AA 909696  
## 6 N323AA 844529  
## 7 N319AA 840510  
## 8 N336AA 838086  
## 9 N329AA 830776  
## 10 N324AA 794895  
## # ... with 4,034 more rows

# dplyr::count  
  
# sum(x > 10) : 횟수 sum 안에 조건문 - 합계 (TRUE개수) x가 10보다 큰 것의 갯수  
# mean(y == 0) : 비율. mean 안에 조건문 - 비율 (TRUE/(TRUE+FALSE개수)) x가 0 인 것의 비율  
# 5 시 이전에 출발하는 비행기는 몇 대?   
not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로  
 summarise(n\_early = sum(dep\_time < 500)) # 다섯시 이전에 출발하는 비행 건수

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day n\_early  
## <int> <int> <int> <int>  
## 1 2013 1 1 0  
## 2 2013 1 2 3  
## 3 2013 1 3 4  
## 4 2013 1 4 3  
## 5 2013 1 5 3  
## 6 2013 1 6 2  
## 7 2013 1 7 2  
## 8 2013 1 8 1  
## 9 2013 1 9 3  
## 10 2013 1 10 3  
## # ... with 355 more rows

# 1시간 이상 지연된 비행기 비율은?  
# not\_cancelled %>% filter(arr\_delay > 60)  
not\_cancelled %>% filter(arr\_delay >= 60) # 1시간 이상 지연 도착한 운항만 뽑기

## # A tibble: 28,317 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 1 1 811 630 101 1047  
## 2 2013 1 1 848 1835 853 1001  
## 3 2013 1 1 957 733 144 1056  
## 4 2013 1 1 1114 900 134 1447  
## 5 2013 1 1 1120 944 96 1331  
## 6 2013 1 1 1255 1200 55 1451  
## 7 2013 1 1 1301 1150 71 1518  
## 8 2013 1 1 1337 1220 77 1649  
## 9 2013 1 1 1342 1320 22 1617  
## 10 2013 1 1 1356 1259 57 1538  
## # ... with 28,307 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # 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>

not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로   
 summarise(hour\_perc = mean(arr\_delay > 60)) # 1시간 이상 지연된 운항건수 비율

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day hour\_perc  
## <int> <int> <int> <dbl>  
## 1 2013 1 1 0.0722  
## 2 2013 1 2 0.0851  
## 3 2013 1 3 0.0567  
## 4 2013 1 4 0.0396  
## 5 2013 1 5 0.0349  
## 6 2013 1 6 0.0470  
## 7 2013 1 7 0.0333  
## 8 2013 1 8 0.0213  
## 9 2013 1 9 0.0202  
## 10 2013 1 10 0.0183  
## # ... with 355 more rows

not\_cancelled %>% group\_by(year, month, day) %>% # 날짜별로  
 summarise(hour\_perc = sum(arr\_delay > 60)) # 1시간 이상 지연된 운항건수

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day hour\_perc  
## <int> <int> <int> <int>  
## 1 2013 1 1 60  
## 2 2013 1 2 79  
## 3 2013 1 3 51  
## 4 2013 1 4 36  
## 5 2013 1 5 25  
## 6 2013 1 6 39  
## 7 2013 1 7 31  
## 8 2013 1 8 19  
## 9 2013 1 9 18  
## 10 2013 1 10 17  
## # ... with 355 more rows

x <- 1:10  
mean(x>6); sum(x>6)

## [1] 0.4

## [1] 4

# 여러 변수를 이용한 그룹 지정  
# 날짜 별로 그룹  
# year, month, day [365] 365개의 그룹이 있다.   
(daily <- group\_by(flights, year, month, day)) # 각 년, 월, 일 별로 group by로 지정되어 있다.

## # A tibble: 336,776 x 19  
## # Groups: year, month, day [365]  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 1 1 517 515 2 830  
## 2 2013 1 1 533 529 4 850  
## 3 2013 1 1 542 540 2 923  
## 4 2013 1 1 544 545 -1 1004  
## 5 2013 1 1 554 600 -6 812  
## 6 2013 1 1 554 558 -4 740  
## 7 2013 1 1 555 600 -5 913  
## 8 2013 1 1 557 600 -3 709  
## 9 2013 1 1 557 600 -3 838  
## 10 2013 1 1 558 600 -2 753  
## # ... with 336,766 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # 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>

# 모든 obs가 다른 그룹이 됐음.. day라는 그룹의 의미가 없어진다 [?]  
# year, month [?]  
(per\_day <- summarise(daily, # 각 년, 월, 일별로  
 flights = n())) # 운항건수를 flights 변수에 저장

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day flights  
## <int> <int> <int> <int>  
## 1 2013 1 1 842  
## 2 2013 1 2 943  
## 3 2013 1 3 914  
## 4 2013 1 4 915  
## 5 2013 1 5 720  
## 6 2013 1 6 832  
## 7 2013 1 7 933  
## 8 2013 1 8 899  
## 9 2013 1 9 902  
## 10 2013 1 10 932  
## # ... with 355 more rows

# year [?]  
(per\_month <- summarise(per\_day, # 각 연, 달별로   
 flights1 = sum(flights))) #월별 운항건수 합계

## # A tibble: 12 x 3  
## # Groups: year [?]  
## year month flights1  
## <int> <int> <int>  
## 1 2013 1 27004  
## 2 2013 2 24951  
## 3 2013 3 28834  
## 4 2013 4 28330  
## 5 2013 5 28796  
## 6 2013 6 28243  
## 7 2013 7 29425  
## 8 2013 8 29327  
## 9 2013 9 27574  
## 10 2013 10 28889  
## 11 2013 11 27268  
## 12 2013 12 28135

# year [?]  
(per\_year <- flights %>% # 각 연도별로   
 group\_by(year, month) %>%   
 summarise(flights2 = n()))

## # A tibble: 12 x 3  
## # Groups: year [?]  
## year month flights2  
## <int> <int> <int>  
## 1 2013 1 27004  
## 2 2013 2 24951  
## 3 2013 3 28834  
## 4 2013 4 28330  
## 5 2013 5 28796  
## 6 2013 6 28243  
## 7 2013 7 29425  
## 8 2013 8 29327  
## 9 2013 9 27574  
## 10 2013 10 28889  
## 11 2013 11 27268  
## 12 2013 12 28135

(per\_year <- summarise(per\_month,   
 flights2 = sum(flights1)))

## # A tibble: 1 x 2  
## year flights2  
## <int> <int>  
## 1 2013 336776

# ungroup() 을 이용하여 그룹 해제   
class(daily) # 이미 grouped\_dataframe 이다

## [1] "grouped\_df" "tbl\_df" "tbl" "data.frame"

class(flights)

## [1] "tbl\_df" "tbl" "data.frame"

class(daily %>% ungroup())

## [1] "tbl\_df" "tbl" "data.frame"

daily %>% summarise(n = n()) # 각 일별로 n을 찾는다

## # A tibble: 365 x 4  
## # Groups: year, month [?]  
## year month day n  
## <int> <int> <int> <int>  
## 1 2013 1 1 842  
## 2 2013 1 2 943  
## 3 2013 1 3 914  
## 4 2013 1 4 915  
## 5 2013 1 5 720  
## 6 2013 1 6 832  
## 7 2013 1 7 933  
## 8 2013 1 8 899  
## 9 2013 1 9 902  
## 10 2013 1 10 932  
## # ... with 355 more rows

daily %>% ungroup() %>% # no longer grouped by date. 더이상 그룹이 지정되지 않아서 전체 sum을 찾는다  
 summarise(n = n()) # all flights

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 336776

# 각 그룹에서 변수 생성  
# 그룹에서 최하위 찾기  
flights %>%   
 group\_by(year, month, day) %>%   
 mutate(delay\_rank = rank(desc(arr\_delay))) %>% # 지연시간이 제일 큰 운항부터 순위를 매긴다  
 select(year:day, arr\_delay, delay\_rank)

## # A tibble: 336,776 x 5  
## # Groups: year, month, day [365]  
## year month day arr\_delay delay\_rank  
## <int> <int> <int> <dbl> <dbl>  
## 1 2013 1 1 11 296.  
## 2 2013 1 1 20 202   
## 3 2013 1 1 33 124.  
## 4 2013 1 1 -18 746.  
## 5 2013 1 1 -25 792   
## 6 2013 1 1 12 279   
## 7 2013 1 1 19 211   
## 8 2013 1 1 -14 704.  
## 9 2013 1 1 -8 616.  
## 10 2013 1 1 8 340.  
## # ... with 336,766 more rows

flights %>% group\_by(year, month, day) %>%   
 filter(rank(desc(arr\_delay)) <= 5) %>% # 순위가 5 이하인 운항. 지연이 많이된 top 5  
 select(year:day, arr\_delay)

## # A tibble: 1,805 x 4  
## # Groups: year, month, day [365]  
## year month day arr\_delay  
## <int> <int> <int> <dbl>  
## 1 2013 1 1 851  
## 2 2013 1 1 338  
## 3 2013 1 1 263  
## 4 2013 1 1 250  
## 5 2013 1 1 456  
## 6 2013 1 2 288  
## 7 2013 1 2 323  
## 8 2013 1 2 368  
## 9 2013 1 2 359  
## 10 2013 1 2 252  
## # ... with 1,795 more rows