your tunr

KD BAE

2021 7 1

library(nycflights13)  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.1.0 v dplyr 1.0.5  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

1-1

b <- flights %>% filter(dep\_delay >= 120) %>% head(5)  
b$carrier

## [1] "MQ" "UA" "UA" "B6" "EV"

1-2

c <- flights %>% filter(dep\_delay <=0 | arr\_delay >0) %>% head(5)  
c$carrier

## [1] "UA" "UA" "AA" "B6" "DL"

1-3

d <- flights %>% filter(dep\_delay >= 60 | arr\_delay <= -30) %>% head(5)  
  
d$carrier

## [1] "DL" "UA" "DL" "MQ" "DL"

1-4

sum(is.na(flights$sched\_dep\_time))

## [1] 0

sum(is.na(flights$sched\_arr\_time))

## [1] 0

sum(is.na(flights$dep\_time))

## [1] 8255

sum(is.na(flights$dep\_delay))

## [1] 8255

sum(is.na(flights$arr\_time))

## [1] 8713

sum(is.na(flights$arr\_delay))

## [1] 9430

출발하지 않은걸로보임 dep\_time은 출발시간인데 출발하지않았은 NA로 된것으로 판단 NA결측치들을 제거하는것이 올바른 판단인거 같음. 출발한시간이없으니 지연된 출발시간이없다고 생각. 그래서 결측값갯수를 확인했는데 같다. 물론 결측값갯수가 같다고 같은변수가 NA인지는 모르겠지만 추측할 수는 있다. 그래서 출발을 안했으니 출발딜레이도 없을을것이다. 도착도해봤지만 도착딜레이한것이랑 도착한것의 결측값이 다르다.도착한 결측값이 더 많은것으로보아 아마 출발하고 추락하여 폭파된 것이나 실종되거나 버뮤다삼각지대로 사라진건가? ㄹㅇㅋㅋ 예정된 출발시간과 예정된 도착시간은 결측값이없다.

2번

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>

a <-flights %>% arrange((dep\_delay))  
a[6,]$carrier

## [1] "MQ"

flights$speed <- flights$distance/flights$air\_time  
max(flights$distance/flights$air\_time,na.rm=TRUE)

## [1] 11.72308

library(tidyverse)  
MOC <- flights %>% arrange(desc(speed))  
  
MOC$carrier[1]

## [1] "DL"

속력을 구해서 데이터에 넣은후 데이터를 정렬하고 속력이 가장 큰값의 carrier을 구했다.

3번

flights %>% select(starts\_with("arr")|starts\_with("dep"))

## # A tibble: 336,776 x 4  
## arr\_time arr\_delay dep\_time dep\_delay  
## <int> <dbl> <int> <dbl>  
## 1 830 11 517 2  
## 2 850 20 533 4  
## 3 923 33 542 2  
## 4 1004 -18 544 -1  
## 5 812 -25 554 -6  
## 6 740 12 554 -4  
## 7 913 19 555 -5  
## 8 709 -14 557 -3  
## 9 838 -8 557 -3  
## 10 753 8 558 -2  
## # ... with 336,766 more rows

4번

flights %>% select(ends\_with("dep\_time")) -> flights2  
flights2 %>% mutate(hour = dep\_time %/% 60,  
 minute = dep\_time %% 60)

## # A tibble: 336,776 x 4  
## dep\_time sched\_dep\_time hour minute  
## <int> <int> <dbl> <dbl>  
## 1 517 515 8 37  
## 2 533 529 8 53  
## 3 542 540 9 2  
## 4 544 545 9 4  
## 5 554 600 9 14  
## 6 554 558 9 14  
## 7 555 600 9 15  
## 8 557 600 9 17  
## 9 557 600 9 17  
## 10 558 600 9 18  
## # ... with 336,766 more rows

a <- flights %>% transmute(air\_time - (arr\_time-dep\_time))   
head(a)

## # A tibble: 6 x 1  
## `air\_time - (arr\_time - dep\_time)`  
## <dbl>  
## 1 -86  
## 2 -90  
## 3 -221  
## 4 -277  
## 5 -142  
## 6 -36

length(which(a>0))

## [1] 56975

length(which(a<0))

## [1] 269153

보통 비행기는 땅에서 달리다가 날라가기 때문에 air\_time이 더 작은것이많고 만약 air\_time이 더크다면 아마 착륙이어려워서 공중에서 있어야할때 air\_time이 더 클것이다 그래서 대부분 air\_time이 더 작고 드문경우에 착륙이 어려우므로 air\_time이 더 크다

flights$delay <- flights %>% transmute(delay = dep\_time - sched\_dep\_time )  
length(which(flights$delay == flights$dep\_delay))

## [1] 228744

length(which(flights$delay != flights$dep\_delay))

## [1] 99777

flights %>% select(delay, dep\_delay)

## # A tibble: 336,776 x 2  
## delay$delay dep\_delay  
## <int> <dbl>  
## 1 2 2  
## 2 4 4  
## 3 2 2  
## 4 -1 -1  
## 5 -46 -6  
## 6 -4 -4  
## 7 -45 -5  
## 8 -43 -3  
## 9 -43 -3  
## 10 -42 -2  
## # ... with 336,766 more rows

흠 이거는 왜 다른지 모르겠네 잘못 작성한건가 근데 같은게 더 많긴해 같아야 정상인데 출발시간을 잘못 체크하신걸수도.

5번

a <- flights %>% group\_by(carrier) %>% summarise(mean\_1 = mean(dep\_delay,na.rm=TRUE))  
b <- flights %>% group\_by(carrier) %>% summarise(var\_1 = var(dep\_delay, na.rm=TRUE))  
a %>% arrange(desc(mean\_1))

## # A tibble: 16 x 2  
## carrier mean\_1  
## <chr> <dbl>  
## 1 F9 20.2   
## 2 EV 20.0   
## 3 YV 19.0   
## 4 FL 18.7   
## 5 WN 17.7   
## 6 9E 16.7   
## 7 B6 13.0   
## 8 VX 12.9   
## 9 OO 12.6   
## 10 UA 12.1   
## 11 MQ 10.6   
## 12 DL 9.26  
## 13 AA 8.59  
## 14 AS 5.80  
## 15 HA 4.90  
## 16 US 3.78

b %>% arrange(desc(var\_1))

## # A tibble: 16 x 2  
## carrier var\_1  
## <chr> <dbl>  
## 1 HA 5492.  
## 2 F9 3406.  
## 3 FL 2773.  
## 4 YV 2418.  
## 5 EV 2167.  
## 6 9E 2107.  
## 7 VX 2008.  
## 8 WN 1879.  
## 9 OO 1855.  
## 10 DL 1579.  
## 11 MQ 1535.  
## 12 B6 1483.  
## 13 AA 1395.  
## 14 UA 1276.  
## 15 AS 984.  
## 16 US 787.

#F9가 가장평균이크고 HA가 분산이 가장크다.

c <- flights %>% group\_by(month,day) %>% summarise(mean\_2=mean(dep\_delay, na.rm=T), n = n())

## `summarise()` has grouped output by 'month'. You can override using the `.groups` argument.

c %>% arrange(desc(mean\_2))

## # A tibble: 365 x 4  
## # Groups: month [12]  
## month day mean\_2 n  
## <int> <int> <dbl> <int>  
## 1 3 8 83.5 979  
## 2 7 1 56.2 966  
## 3 9 2 53.0 929  
## 4 7 10 52.9 1004  
## 5 12 5 52.3 969  
## 6 5 23 51.1 988  
## 7 9 12 50.0 992  
## 8 6 28 48.8 994  
## 9 6 24 47.2 994  
## 10 7 22 46.7 1000  
## # ... with 355 more rows

3월8일이 가장 크다.

d <- flights %>% group\_by(month=3,day=8,carrier) %>% summarise(var\_2 = var(dep\_delay,na.rm = TRUE))

## `summarise()` has grouped output by 'month', 'day'. You can override using the `.groups` argument.

d %>% arrange((var\_2))

## # A tibble: 16 x 4  
## # Groups: month, day [1]  
## month day carrier var\_2  
## <dbl> <dbl> <chr> <dbl>  
## 1 3 8 US 787.  
## 2 3 8 AS 984.  
## 3 3 8 UA 1276.  
## 4 3 8 AA 1395.  
## 5 3 8 B6 1483.  
## 6 3 8 MQ 1535.  
## 7 3 8 DL 1579.  
## 8 3 8 OO 1855.  
## 9 3 8 WN 1879.  
## 10 3 8 VX 2008.  
## 11 3 8 9E 2107.  
## 12 3 8 EV 2167.  
## 13 3 8 YV 2418.  
## 14 3 8 FL 2773.  
## 15 3 8 F9 3406.  
## 16 3 8 HA 5492.

US가 가장작다.