Soujanya\_Dash.ExercisesUpTo5.6.7

Soujanya Dash

03/06/2021

## Prerequisites

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.5  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

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

library(nycflights13)

## Warning: package 'nycflights13' was built under R version 4.0.5

### Q.1 Brainstorm different ways to assess the typical delay characteristics of a group of flights. Consider the following scenarios:

### A flight is 15 minutes early 50% of the time, and 15 minutes late 50% of the time.

not\_cancelled <- flights %>%  
 filter(!is.na(air\_time))  
not\_cancelled %>%  
 group\_by(tailnum) %>%  
 summarise(  
 count = n(),  
 p\_15\_early\_arr = mean(arr\_delay < -15),  
 p\_15\_dep\_arr = mean(dep\_delay < -15)  
 ) %>%  
 filter(p\_15\_early\_arr > 0.5 | p\_15\_dep\_arr > 0.5) %>%  
 filter(count > 30) %>%  
 arrange(desc(p\_15\_early\_arr), desc(p\_15\_dep\_arr))

## # A tibble: 43 x 4  
## tailnum count p\_15\_early\_arr p\_15\_dep\_arr  
## <chr> <int> <dbl> <dbl>  
## 1 N852VA 77 0.571 0.0130  
## 2 N548AA 49 0.571 0   
## 3 N579AA 49 0.571 0   
## 4 N3KTAA 39 0.564 0   
## 5 N425AA 41 0.561 0   
## 6 N540AA 34 0.559 0   
## 7 N855VA 74 0.554 0   
## 8 N403AS 51 0.549 0   
## 9 N595AA 51 0.549 0   
## 10 N514AA 42 0.548 0   
## # ... with 33 more rows

#both arrival and departure to be taken into account.

### A flight is always 10 minutes late.

not\_cancelled <- flights %>%  
 filter(!is.na(air\_time))  
#as all flights should have air time  
not\_cancelled %>%  
 group\_by(tailnum) %>%  
 summarise(  
 count = n(),  
 exact\_10 = mean(arr\_delay == 10)  
 ) %>%   
 filter(count > 10) %>%  
 arrange(desc(exact\_10))

## # A tibble: 3,397 x 3  
## tailnum count exact\_10  
## <chr> <int> <dbl>  
## 1 N8324A 13 0.154   
## 2 N780NC 14 0.143   
## 3 N265WN 25 0.12   
## 4 N625MQ 18 0.111   
## 5 N940WN 18 0.111   
## 6 N787SA 19 0.105   
## 7 N954AT 29 0.103   
## 8 N260WN 31 0.0968  
## 9 N994AT 31 0.0968  
## 10 N400WN 21 0.0952  
## # ... with 3,387 more rows

### A flight is 30 minutes early 50% of the time, and 30 minutes late 50% of the time.

not\_cancelled %>%  
 group\_by(tailnum) %>%  
 mutate(  
 count = n(),  
 arr\_30\_early = mean(arr\_delay < -30),  
 dep\_30\_early = mean(dep\_delay < -30),  
 arr\_30\_late = mean(arr\_delay > 30),  
 dep\_30\_late = mean(dep\_delay > 30)  
 ) %>%  
 arrange(desc(arr\_30\_early), desc(dep\_30\_early), arr\_30\_late, dep\_30\_late) %>%  
 select(dest)

## Adding missing grouping variables: `tailnum`

## # A tibble: 327,346 x 2  
## # Groups: tailnum [4,037]  
## tailnum dest   
## <chr> <chr>  
## 1 N7AYAA MIA   
## 2 N315AS SEA   
## 3 N560AS SEA   
## 4 N594AS SEA   
## 5 N537AS SEA   
## 6 N594AS SEA   
## 7 N538AS SEA   
## 8 N594AS SEA   
## 9 N538AS SEA   
## 10 N537AS SEA   
## # ... with 327,336 more rows

### Q.2 Come up with another approach that will give you the same output as not\_cancelled %>% count(dest) and not\_cancelled %>% count(tailnum, wt = distance) (without using count()).

Ans2. In the previous exercise, we understood that missing data would mean the plane has been cancelled. So, the approach could be by using is.na()

not\_cancelled <- filter(flights, !is.na(dep\_delay), !is.na(arr\_delay) )

Count() counts the number of instances of the label in a certain attribute or variable so maybe we can use group\_by()&summarise().

not\_cancelled %>% group\_by(dest) %>% summarise(num = length(dest))

## # A tibble: 104 x 2  
## dest num  
## <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

Similarly,for the other one,

not\_cancelled %>% group\_by(tailnum) %>% summarise(num = sum(distance))

## # A tibble: 4,037 x 2  
## tailnum num  
## <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

Here we used sum instead as we want to have the total distance.

### Q.3 Our definition of cancelled flights (is.na(dep\_delay) | is.na(arr\_delay) ) is slightly suboptimal. Why? Which is the most important column?

If there’s NA in the dep\_delay, it may suggest that the flight took off and may or maynot reach the destination (given it may be redirected or crashes). on the other hand, NA in arr\_delay would mean that the flight didn’t take off to begin with. In any case, we may also consider taking the total delay case instead (if total delay is NA that means the flight must be cancelled).

### Q.4 Look at the number of cancelled flights per day. Is there a pattern? Is the proportion of cancelled flights related to the average delay?

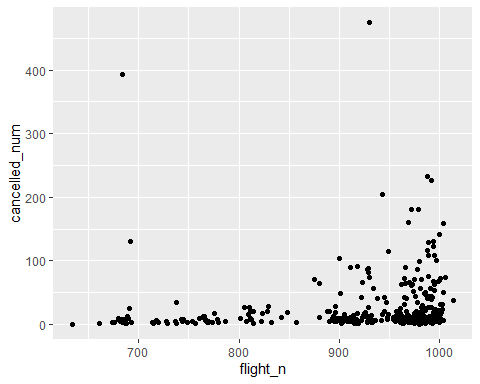
Ans4.

cancels\_day <-   
 flights %>% mutate(cancelled = (is.na(arr\_delay) | is.na(dep\_delay))) %>% group\_by(year, month, day) %>%  
 summarise(  
 cancelled\_num = sum(cancelled),  
 flight\_n = n()  
)

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

# Since we want to know the number of observations (cancelled flights per day)   
view(cancels\_day)

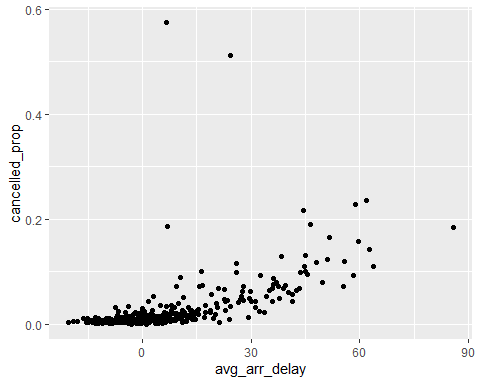
ggplot(cancels\_day) +  
 geom\_point(aes(flight\_n, cancelled\_num))

 Next is the relation between average delay and proportion of cancelled flights (mean).

cancels\_and\_delays <-   
 flights %>%  
 mutate(cancelled = (is.na(arr\_delay) | is.na(dep\_delay))) %>%  
 group\_by(year, month, day) %>%  
 summarise(  
 cancelled\_prop = mean(cancelled),  
 avg\_dep\_delay = mean(dep\_delay, na.rm = TRUE),  
 avg\_arr\_delay = mean(arr\_delay, na.rm = TRUE)  
 ) %>%  
 ungroup()

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

ggplot(cancels\_and\_delays) +  
 geom\_point(aes(x = avg\_arr\_delay, y = cancelled\_prop))

 There is an increasing relationship.

### Q.5 Which carrier has the worst delays?

Ans5. For finding the carrier with worst delays:

flights$delay <- flights$arr\_delay+flights$dep\_delay  
flights %>%  
 group\_by(carrier) %>%  
 summarise(delay = mean(delay, na.rm = TRUE)) %>%  
 arrange(desc(delay))

## # A tibble: 16 x 2  
## carrier delay  
## <chr> <dbl>  
## 1 F9 42.1   
## 2 FL 38.7   
## 3 EV 35.6   
## 4 YV 34.5   
## 5 WN 27.3   
## 6 OO 24.5   
## 7 9E 23.8   
## 8 B6 22.4   
## 9 MQ 21.2   
## 10 UA 15.6   
## 11 VX 14.5   
## 12 DL 10.9   
## 13 AA 8.93  
## 14 US 5.87  
## 15 HA -2.01  
## 16 AS -4.10

F9 flight has the worst delay according to the table.

filter(flights, carrier == "F9")

## # A tibble: 685 x 20  
## 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 1 833 835 -2 1134 1102  
## 2 2013 1 1 1716 1730 -14 1947 1953  
## 3 2013 1 2 827 835 -8 1120 1102  
## 4 2013 1 2 1728 1730 -2 1952 1953  
## 5 2013 1 3 835 835 0 1102 1102  
## 6 2013 1 3 1933 1730 123 2131 1953  
## 7 2013 1 4 834 835 -1 1059 1102  
## 8 2013 1 4 1831 1730 61 2029 1953  
## 9 2013 1 5 835 835 0 1057 1102  
## 10 2013 1 5 1726 1730 -4 1948 1953  
## # ... with 675 more rows, and 12 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>,  
## # delay <dbl>

### Q.6 What does the sort argument to count() do. When might you use it?

The sort argument to count() sorts in descending order of n. If sort is TRUE, it will show the largest groups at the top. As in the previous questions, we can get the most common group like F9 flights being delayed the most. The following is to find the most common carrier (UA).

flights %>%  
 count(carrier, sort = TRUE)

## # A tibble: 16 x 2  
## carrier n  
## <chr> <int>  
## 1 UA 58665  
## 2 B6 54635  
## 3 EV 54173  
## 4 DL 48110  
## 5 AA 32729  
## 6 MQ 26397  
## 7 US 20536  
## 8 9E 18460  
## 9 WN 12275  
## 10 VX 5162  
## 11 FL 3260  
## 12 AS 714  
## 13 F9 685  
## 14 YV 601  
## 15 HA 342  
## 16 OO 32

flights %>%  
 count(carrier)

## # A tibble: 16 x 2  
## carrier n  
## <chr> <int>  
## 1 9E 18460  
## 2 AA 32729  
## 3 AS 714  
## 4 B6 54635  
## 5 DL 48110  
## 6 EV 54173  
## 7 F9 685  
## 8 FL 3260  
## 9 HA 342  
## 10 MQ 26397  
## 11 OO 32  
## 12 UA 58665  
## 13 US 20536  
## 14 VX 5162  
## 15 WN 12275  
## 16 YV 601