Lab 4

Suixin Jiang

9/22/2019

- Use the `flights` data frame from the nycflights13 package.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.4.0
## Warning: package 'ggplot2' was built under R version 3.5.3
## Warning: package 'tibble' was built under R version 3.5.3
## Warning: package 'tidyr' was built under R version 3.5.3
## Warning: package 'purrr' was built under R version 3.5.3
## Warning: package 'dplyr' was built under R version 3.5.3
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 3.5.3
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 3.5.3
```

```
data("flights")
```

1. Which plane ('tailnum') has the worst departure delay record?

```
# As the result shows, N844MH has worst departure delay record with the average 297 minutes.
flights %>%
  group_by(tailnum) %>%
  summarize(tailnum_mean_dep = mean(dep_delay, na.rm = T)) %>%
  arrange(desc(tailnum_mean_dep))
```

```
## # A tibble: 4,044 x 2
##    tailnum tailnum_mean_dep
##    <chr>              <dbl>
##  1 N844MH               297
##  2 N922EV               274
##  3 N587NW               272
##  4 N911DA               268
##  5 N851NW               233
##  6 N654UA               227
##  7 N928DN               203
##  8 N7715E               186
##  9 N665MQ               177
## 10 N136DL               165
## # ... with 4,034 more rows
```

2. What time of day should you fly if you want to avoid delays as much as possible?

```r
# Relatively speaking, people who take a morning flight between 5am and 6am
# may avoid delays much than other time.
flights %>%
  group_by(hour) %>%
  summarize(less_mean_dep = mean(dep_delay, na.rm = T)) %>%
  arrange(less_mean_dep)
```

```
## # A tibble: 20 x 2
##     hour less_mean_dep
##    <dbl>         <dbl>
##  1     5         0.688
##  2     6         1.64
##  3     7         1.91
##  4     8         4.13
##  5     9         4.58
##  6    10         6.50
##  7    11         7.19
##  8    12         8.61
##  9    13        11.4
## 10    14        13.8
## 11    23        14.0
## 12    15        16.9
## 13    16        18.8
## 14    22        18.8
## 15    17        21.1
## 16    18        21.1
## 17    21        24.2
## 18    20        24.3
## 19    19        24.8
## 20     1       NaN
```

3. For each destination, compute the total minutes of arrival delay. For each flight, compute the proportion of the arrival delay for its destination.

```r
# Total minutes of arrival delay for each destination shown below as a list.
flights %>%
  group_by(dest) %>%
  summarize(sum_arr_delay = sum(arr_delay, na.rm = T)) %>%
  as.list(sum_arr_delay)
```

```
## $dest
##   [1] "ABQ" "ACK" "ALB" "ANC" "ATL" "AUS" "AVL" "BDL" "BGR" "BHM" "BNA"
##  [12] "BOS" "BQN" "BTV" "BUF" "BUR" "BWI" "BZN" "CAE" "CAK" "CHO" "CHS"
##  [23] "CLE" "CLT" "CMH" "CRW" "CVG" "DAY" "DCA" "DEN" "DFW" "DSM" "DTW"
##  [34] "EGE" "EYW" "FLL" "GRR" "GSO" "GSP" "HDN" "HNL" "HOU" "IAD" "IAH"
##  [45] "ILM" "IND" "JAC" "JAX" "LAS" "LAX" "LEX" "LGA" "LGB" "MCI" "MCO"
##  [56] "MDW" "MEM" "MHT" "MIA" "MKE" "MSN" "MSP" "MSY" "MTJ" "MVY" "MYR"
##  [67] "OAK" "OKC" "OMA" "ORD" "ORF" "PBI" "PDX" "PHL" "PHX" "PIT" "PSE"
##  [78] "PSP" "PVD" "PWM" "RDU" "RIC" "ROC" "RSW" "SAN" "SAT" "SAV" "SBN"
##  [89] "SDF" "SEA" "SFO" "SJC" "SJU" "SLC" "SMF" "SNA" "SRQ" "STL" "STT"
## [100] "SYR" "TPA" "TUL" "TVC" "TYS" "XNA"
##
## $sum_arr_delay
##   [1]   1113   1281   6018    -20 190260  14514   2089   2904   2874   4540
##  [11]  71867  43780   7322  22467  40883   3025  18096    266   4427  16586
##  [21]    437  29226  40344 100645  35260   1966  57233  17740  82609  61700
##  [31]   2702   9940  49038   1305    108  96153  13242  21056  12589     30
##  [41]   -957  14948  74631  30046    496  19692    590  31069   1534   8768
##  [51]    -22      0    -41  27359  76185  49766  17948  13782   3467  38379
##  [61]  11229  50375  24111     25    -60    267    951   9645  12009  97352
##  [71]  15701  55548   6900  15606   9659  21092   2818   -229   5812  26679
##  [81]  78107  47181  27260  11340   8504   4577  11332     65  13987  -4270
##  [91]  35210   1131  14551    432   3415  -6389   3702  45887  -1987  15199
## [101]  54749   9896   1232  13912   7406
```

```r
# Only positive arrival delay values were used to calculate the proportion of
# arrival delay for each flight.
flights %>%
  filter(arr_delay > 0) %>%
  group_by(dest) %>%
  mutate(total_arr_delay = sum(arr_delay, na.rm = T),
         prop_arr_delay = arr_delay / total_arr_delay) %>%
  select(dest,flight,tailnum,arr_delay,total_arr_delay,prop_arr_delay) %>%
  arrange(desc(prop_arr_delay))
```

```
## # A tibble: 133,004 x 6
## # Groups:   dest [103]
##    dest  flight tailnum arr_delay total_arr_delay prop_arr_delay
##    <chr>  <int> <chr>       <dbl>           <dbl>          <dbl>
##  1 ANC      887 N528UA         39              62          0.629
##  2 MTJ      385 N806UA        101             170          0.594
##  3 PSP       55 N839VA         17              36          0.472
##  4 SBN     5383 N398CA         53             125          0.424
##  5 SBN     5383 N761ND         50             125          0.4
##  6 HDN      441 N817UA         43             119          0.361
##  7 BZN      568 N436UA        154             491          0.314
##  8 JAC     1506 N16701        175             619          0.283
##  9 HDN      355 N474UA         32             119          0.269
## 10 CHO     5325 N611QX        228             947          0.241
## # ... with 132,994 more rows
```

4. Delays are typically temporally correlated: even once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using `lag()`, explore how the departure delay of a flight is related to the delay of the immediately preceding flight.

```r
flights1 <- arrange(flights, origin, year, month, day, hour, minute)
flights1$next_dep_delay <- lag(flights1$dep_delay)
flights2 <- group_by(flights1, origin)
cor_results <- summarize(flights2, corr = cor(dep_delay, next_dep_delay,
                                             use = 'pairwise.complete.obs'))
print(cor_results)
```

```
## # A tibble: 3 x 2
##   origin  corr
##   <chr>  <dbl>
## 1 EWR    0.254
## 2 JFK    0.238
## 3 LGA    0.282
```

5. Look at each destination. Can you find flights that are suspiciously fast? (i.e. flights that represent a potential data entry error). Compute the air time of a flight relative to the shortest flight to that destination. Which flights were most delayed in the air?

```r
# The result shows TWO suspicious flights for each destination.
flights %>%
  group_by(dest) %>%
  select(dest,flight,tailnum,sched_dep_time,sched_arr_time,air_time) %>%
  slice(1:2) %>%
  arrange(air_time)
```

```
## # A tibble: 208 x 6
## # Groups:   dest [105]
##    dest  flight tailnum sched_dep_time sched_arr_time air_time
##    <chr>  <int> <chr>            <int>          <int>    <dbl>
##  1 BDL     4276 N13903            2200           2253       24
##  2 BDL     4106 N19554            1322           1416       25
##  3 PVD     4404 N15912            2110           2212       28
##  4 PVD     4404 N17108            2110           2212       29
##  5 PHL     1467 N959UW             915           1033       32
##  6 ALB     4112 N13538            1317           1423       33
##  7 PHL     4088 N8968E            1610           1729       35
##  8 ALB     3260 N19554            1621           1724       36
##  9 MVY     1338 N368JB            1350           1453       36
## 10 MHT     4434 N13566            1355           1459       37
## # ... with 198 more rows
```

```r
# The result shows the most in-air delayed flight for each destination.
flights %>%
  group_by(dest) %>%
  mutate(time_waste = air_time - min(air_time, na.rm = T)) %>%
  select(dest,flight,tailnum,air_time,time_waste) %>%
  top_n(1, air_time) %>%
  arrange(desc(air_time))
```

```
## Warning in min(air_time, na.rm = T): min      ;  Inf
```

```
## # A tibble: 112 x 5
## # Groups:   dest [104]
##    dest  flight tailnum air_time time_waste
##    <chr>  <int> <chr>      <dbl>      <dbl>
##  1 HNL       15 N77066       695        133
```

```
## 2 SFO 841 N703TW 490 195
## 3 LAX 426 N178DN 440 165
## 4 ANC 887 N572UA 434 46
## 5 SAN 89 N794JB 413 134
## 6 SNA 1075 N16709 405 131
## 7 BUR 359 N624JB 403 110
## 8 LAS 587 N852UA 399 143
## 9 SJC 669 N632JB 396 91
## 10 SEA 1100 N17245 394 119
## # ... with 102 more rows
```

6. Find all destinations that are flown by at least two carriers. (hint: use `n_distinct()`)

```
flights %>%
  group_by(dest) %>%
  summarize(n = n_distinct(carrier, na.rm = T)) %>%
  filter(n >= 2) %>%
  arrange(desc(n))
```

```
## # A tibble: 76 x 2
##    dest      n
##    <chr> <int>
##  1 ATL       7
##  2 BOS       7
##  3 CLT       7
##  4 ORD       7
##  5 TPA       7
##  6 AUS       6
##  7 DCA       6
##  8 DTW       6
##  9 IAD       6
## 10 MSP       6
## # ... with 66 more rows
```