CPSC-375 Homework 4

Kenn Son, Hamid Suda, Vivian Truong

2/23/2022

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

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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>
```

1. List data only for flights that departed on February 12, 2013.

```
flights %>% filter(year=="2013", month=="2", day=="12")
```

```
## # A tibble: 893 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     2    12       17           2245        92      122           2356
##  2  2013     2    12      506            500         6      703            648
##  3  2013     2    12      520            525        -5      837            820
##  4  2013     2    12      524            530        -6      922            831
##  5  2013     2    12      535            540        -5      950           1016
##  6  2013     2    12      539            540        -1      828            850
##  7  2013     2    12      551            600        -9      645            708
##  8  2013     2    12      552            600        -8      925            910
##  9  2013     2    12      553            600        -7      652            703
## 10  2013     2    12      555            600        -5      903            911
## # ... with 883 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>
```

2. List data only for flights that were delayed (both arrival and departure) by more than 2 hours.

```
flights %>% filter(dep_delay > 200) %>% filter(arr_delay > 200)
```

```
## # A tibble: 2,376 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      848           1835       853     1001           1950
##  2  2013     1     1     1815           1325       290     2120           1542
##  3  2013     1     1     1842           1422       260     1958           1535
##  4  2013     1     1     2006           1630       216     2230           1848
##  5  2013     1     1     2115           1700       255     2330           1920
##  6  2013     1     1     2205           1720       285       46           2040
##  7  2013     1     1     2343           1724       379      314           1938
##  8  2013     1     2     1244            900       224     1431           1104
##  9  2013     1     2     1332            904       268     1616           1128
## 10  2013     1     2     1412            838       334     1710           1147
## # ... with 2,366 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>
```

3. List data only for flights that were delayed (either arrival or departure) by more than 2 hours.

```
flights %>% filter(dep_delay > 200|arr_delay > 200)
```

```
## # A tibble: 3,275 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      848           1835       853     1001           1950
##  2  2013     1     1     1815           1325       290     2120           1542
##  3  2013     1     1     1842           1422       260     1958           1535
##  4  2013     1     1     2006           1630       216     2230           1848
##  5  2013     1     1     2115           1700       255     2330           1920
##  6  2013     1     1     2205           1720       285       46           2040
##  7  2013     1     1     2343           1724       379      314           1938
##  8  2013     1     2     1244            900       224     1431           1104
##  9  2013     1     2     1332            904       268     1616           1128
## 10  2013     1     2     1412            838       334     1710           1147
## # ... with 3,265 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>
```

4. List data only for flights that were operated by United, American, or Delta.

```
flights %>% filter(carrier == "UA" | carrier == "AA" | carrier == "DL")
```

```
## # A tibble: 139,504 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      554            600        -6      812            837
##  5  2013     1     1      554            558        -4      740            728
##  6  2013     1     1      558            600        -2      753            745
##  7  2013     1     1      558            600        -2      924            917
##  8  2013     1     1      558            600        -2      923            937
##  9  2013     1     1      559            600        -1      941            910
## 10  2013     1     1      559            600        -1      854            902
## # ... with 139,494 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>
```

5. Sort data in order of fastest flights (air_time).

```
flights %>% arrange(air_time)
```

```
## # 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    16     1355           1315        40     1442           1411
##  2  2013     4    13      537            527        10      622            628
##  3  2013    12     6      922            851        31     1021            954
##  4  2013     2     3     2153           2129        24     2247           2224
##  5  2013     2     5     1303           1315       -12     1342           1411
##  6  2013     2    12     2123           2130        -7     2211           2225
##  7  2013     3     2     1450           1500       -10     1547           1608
##  8  2013     3     8     2026           1935        51     2131           2056
##  9  2013     3    18     1456           1329        87     1533           1426
## 10  2013     3    19     2226           2145        41     2305           2246
## # ... 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>
```

6. Sort data in order of longest duration flights (air_time).

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

```
## # 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     3    17     1337           1335         2     1937           1836
##  2  2013     2     6      853            900        -7     1542           1540
##  3  2013     3    15     1001           1000         1     1551           1530
##  4  2013     3    17     1006           1000         6     1607           1530
##  5  2013     3    16     1001           1000         1     1544           1530
##  6  2013     2     5      900            900         0     1555           1540
##  7  2013    11    12      936            930         6     1630           1530
##  8  2013     3    14      958           1000        -2     1542           1530
##  9  2013    11    20     1006           1000         6     1639           1555
## 10  2013     3    15     1342           1335         7     1924           1836
## # ... 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>
```

7. Show only the origin and destination of flights sorted by longest flights.

```
flights %>% arrange(desc(air_time)) %>% select(origin, dest)
```

```
## # A tibble: 336,776 x 2
##    origin dest
##    <chr>  <chr>
##  1 EWR    HNL
##  2 JFK    HNL
##  3 JFK    HNL
##  4 JFK    HNL
##  5 JFK    HNL
##  6 JFK    HNL
##  7 EWR    HNL
##  8 JFK    HNL
##  9 JFK    HNL
## 10 EWR    HNL
## # ... with 336,766 more rows
```

8. Add a new variable that indicates the total delay (both departure and arrival delay).

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay)
```

```
## # A tibble: 336,776 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      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 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>,
## #   total_delay <dbl>
```

9. Show only the origin and destination of flights sorted by descending order of total delay.

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay) %>%
  arrange(desc(total_delay)) %>% select(origin,dest)
```

```
## # A tibble: 336,776 x 2
##    origin dest 
##    <chr>  <chr>
##  1 JFK    HNL  
##  2 JFK    CMH  
##  3 EWR    ORD  
##  4 JFK    SFO  
##  5 JFK    CVG  
##  6 JFK    TPA  
##  7 LGA    MSP  
##  8 LGA    ATL  
##  9 EWR    MIA  
## 10 EWR    ORD  
## # ... with 336,766 more rows
```

10. Show only the origin and destination of 10 most delayed flights [Hint: there are multiple ways of solving this. Some additional functions that you will find useful are head(), slice(), min_rank().]

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay) %>%
  arrange(desc(total_delay)) %>% top_n(10, total_delay)
```

```
## # A tibble: 10 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     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     7    22     2257            759       898      121           1026
##  9  2013    12     5      756           1700       896     1058           2020
## 10  2013     5     3     1133           2055       878     1250           2215
## # ... with 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>, total_delay <dbl>
```

11. Show the average total delay for all flights

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay) %>%
  summarise(mean(total_delay, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   `mean(total_delay, na.rm = TRUE)`
##                               <dbl>
## 1                              19.5
```

12. Show the average total delay for every departure city.

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay) %>%
  group_by(origin) %>% summarise(mean(total_delay, na.rm = TRUE))
```

```
## # A tibble: 3 x 2
##   origin `mean(total_delay, na.rm = TRUE)`
##   <chr>                              <dbl>
## 1 EWR                                 24.1
## 2 JFK                                 17.6
## 3 LGA                                 16.1
```

13. Show the average total delay for every departure-arrival city pair.

```
flights %>% mutate(total_delay = flights$dep_delay + flights$arr_delay) %>%
  group_by(origin, dest) %>% summarise(mean(total_delay, na.rm = TRUE))
```

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

```
## # A tibble: 224 x 3
## # Groups:   origin [3]
##    origin dest  `mean(total_delay, na.rm = TRUE)`
##    <chr>  <chr>                             <dbl>
##  1 EWR    ALB                                37.8
##  2 EWR    ANC                                10.4
##  3 EWR    ATL                                28.6
##  4 EWR    AUS                                11
##  5 EWR    AVL                                17.4
##  6 EWR    BDL                                24.8
##  7 EWR    BNA                                30.3
##  8 EWR    BOS                                17.3
##  9 EWR    BQN                                34.5
## 10 EWR    BTV                                30.0
## # ... with 214 more rows
```