Homework 4

MTH 3270

Tobias Boggess

2/23/2022

Problems 1-3 from Worksheet

Question 1: Write commands that do the following.

a) Arrived more than two hours late but didn't leave late. Report your R command(s).

Code:

```
library(nycflights13)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
filter(.data = flights, dep_delay <= 0 & arr_delay > 120)
```

```
## # A tibble: 29 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    27     1419           1420        -1     1754           1550
##  2  2013    10     7     1350           1350         0     1736           1526
##  3  2013    10     7     1357           1359        -2     1858           1654
##  4  2013    10    16      657            700        -3     1258           1056
##  5  2013    11     1      658            700        -2     1329           1015
##  6  2013     3    18     1844           1847        -3       39           2219
##  7  2013     4    17     1635           1640        -5     2049           1845
##  8  2013     4    18      558            600        -2     1149            850
##  9  2013     4    18      655            700        -5     1213            950
## 10  2013     5    22     1827           1830        -3     2217           2010
## # ... with 19 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>
```

b) Were delayed by at least an hour, but made up over 30 minutes during flight. Report your R command(s).
Code:

```
filter(.data = flights, dep_delay > 60 & arr_delay < 30)
```

```
## # A tibble: 181 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     3     1850           1745        65     2148           2120
##  2  2013     1     3     1950           1845        65     2228           2227
##  3  2013     1     6     1019            900        79     1558           1530
##  4  2013     1     7     1543           1430        73     1758           1735
##  5  2013     1    12     1706           1600        66     1949           1927
##  6  2013     1    12     1953           1845        68     2154           2137
##  7  2013     1    19     1456           1355        61     1636           1615
##  8  2013     1    21     1531           1430        61     1843           1815
##  9  2013     1    21     1648           1545        63     1939           1910
## 10  2013    10    10     1938           1835        63     2158           2148
## # ... with 171 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>
```

Question 2: Do the following.

a) Find the fastest flights (i.e. the ones that spent the least time in the air). Report your R command(s).
Code:

```
arrange(.data = flights, 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>
```

b) Find the longest flights (i.e. the ones that spent the most time in the air). Report your R command(s).
Code:

```
head(arrange(.data = flights, desc(air_time)), 6)
```

```
## # A tibble: 6 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
## # ... with 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>
```

Question 3: Do the following.

a) Use filter() (from the "dplyr" package) to extract a subset of the rows of the nels88 data. Report your R command(s).
Code:

```
my.file <- file.choose()
nels.data <-
  read.csv(my.file,
           header = TRUE,
           sep = " ",
           stringsAsFactors =  FALSE)
head(filter(.data = nels.data, heldback == "YES"), 6)
```

```
##       id sch_id heldback         schtype               race    ses female
## 1 175544   1755      YES CATHOLIC SCHOOL           HISPANIC  0.089      0
## 2 175551   1755      YES CATHOLIC SCHOOL BLACK NOT HISPANIC  0.205      1
## 3 175558   1755      YES CATHOLIC SCHOOL WHITE NOT HISPANIC -0.050      0
## 4 175560   1755      YES CATHOLIC SCHOOL AMER IND/AK NATIVE  0.210      1
## 5 180650   1806      YES CATHOLIC SCHOOL               <NA> -0.923      0
## 6 180675   1806      YES CATHOLIC SCHOOL WHITE NOT HISPANIC -0.349      1
##   minority asian hispanic black white native catholic private bymath f1math
## 1        1     0        1     0     0      0        1       0  45.10     NA
## 2        1     0        0     1     0      0        1       0  40.45  43.27
## 3        0     0        0     0     1      0        1       0  38.73     NA
## 4        1     0        0     0     0      1        1       0  33.66  38.61
## 5       NA    NA       NA    NA    NA     NA        1       0  34.63     NA
## 6        0     0        0     0     1      0        1       0     NA  58.96
##   f2math         f2dropout
## 1     NA              <NA>
## 2     NA ALTRNATIVE STDNT
## 3     NA              <NA>
## 4  42.48 DID NOT DROP OUT
## 5     NA              <NA>
## 6  63.64 DID NOT DROP OUT
```

b) Use summarize() (from “dplyr”) to compute a summary statistic for each of at least three variables in the nels88 data. Report your R command(s).
Code:

```
summarize(.data = nels.data,
          mean(bymath, na.rm = TRUE),
          sum(female),
          min(ses, na.rm = TRUE))
```

```
##   mean(bymath, na.rm = TRUE) sum(female) min(ses, na.rm = TRUE)
## 1                   44.53073        3136                  -2.97
```

c) Use mutate() or transmute() (from “dplyr”) to compute at least one new variable from existing variables in the nels88 data. Report your R command(s).
Code:

```
mean.func <- function(x, y, z) {
  if (!is.na(x) & !is.na(y) & !is.na(z)){
    mean.result <- (x + y + z) / 3
    mean.result
  } else if (!is.na(x) & !is.na(y) & is.na(z)) {
    mean.result <- (x + y) / 2
    mean.result
  } else if (!is.na(x) & is.na(y) & is.na(z)) {
    mean.result <- x
    mean.result
  } else {
    mean.result <- NA
    mean.result
  }
}
head(mutate(
  .data = nels.data,
  math_avg = mean.func(bymath, f1math, f2math),
  .after = f2math
),
6)
```

```
##       id sch_id heldback         schtype                 race    ses female
## 1 175507   1755       NO CATHOLIC SCHOOL   WHITE NOT HISPANIC  0.912      1
## 2 175517   1755       NO CATHOLIC SCHOOL   BLACK NOT HISPANIC  0.761      1
## 3 175521   1755       NO CATHOLIC SCHOOL ASIAN/PACIFIC ISLNDR  0.786      1
## 4 175528   1755       NO CATHOLIC SCHOOL   WHITE NOT HISPANIC -0.019      1
## 5 175544   1755      YES CATHOLIC SCHOOL             HISPANIC  0.089      0
## 6 175550   1755       NO CATHOLIC SCHOOL   WHITE NOT HISPANIC -0.014      1
##   minority asian hispanic black white native catholic private bymath f1math
## 1        0     0        0     0     1      0        1       0  37.40     NA
## 2        1     0        0     1     0      0        1       0  46.73     NA
## 3        1     1        0     0     0      0        1       0  36.34     NA
## 4        0     0        0     0     1      0        1       0  49.16     NA
## 5        1     0        1     0     0      0        1       0  45.10     NA
## 6        0     0        0     0     1      0        1       0  38.44     NA
##   f2math math_avg f2dropout
## 1     NA    37.40      <NA>
## 2     NA    46.73      <NA>
## 3     NA    36.34      <NA>
## 4     NA    49.16      <NA>
## 5     NA    45.10      <NA>
## 6     NA    38.44      <NA>
```

Book Problems

Question 6: For each task, say which verb it is:

a) Find the average of one of the variables
The function for finding the average of one of the variables is mean().

b) Add a new column that is the ratio between two variables.
The best function to add a new column that is the ratio between two variables could be either mutate() with a parameter that calculates the proportion.

c) Sort the cases in descending order of a variable
The function in R to sort the cases based on the descending order of a variable would be desc().

d) Create a new data table that includes only those cases that meet a criterion.
A way to create a new data table that includes on the cases wanted would include filter().

e) From a data table with three categorical variables A, B, and C, and a quantitative variable X, produce a data frame that has the same cases but only the variables A and X.
The function in R that would keep only the variables A and x would be transmute().

Question 9: What month had the highest proportion of cancelled flights? What month had the lowest? Interpret any seasonal patterns.

Code:

```
flights %>% group_by(month) %>% summarize(Cancellations = sum(is.na(dep_time))) %>% arrange(desc(Cancel
```

```
## # A tibble: 12 x 2
##    month Cancellations
##    <int>         <int>
##  1     2          1261
##  2    12          1025
##  3     6          1009
##  4     7           940
##  5     3           861
##  6     4           668
##  7     5           563
##  8     1           521
##  9     8           486
## 10     9           452
## 11    10           236
## 12    11           233
```

The month that seems to have the most cancellations is February and the month with the least number of cancellations is November. The least amount of cancellations occur during the holidays like Thanksgiving. More Cancellations tend to occur during Winter and Summer than the Spring or Fall.

Question 14: What plane (specified by the tailnum variable) traveled the

most times from New York City airports in 2013? Plot the number of trips per month over the year.
Code:

```
library(ggplot2)
flightsByTailNum <-
  flights %>% group_by(tailnum) %>% summarise(num_flights = n())
arrange(.data = flightsByTailNum, desc(num_flights))
```

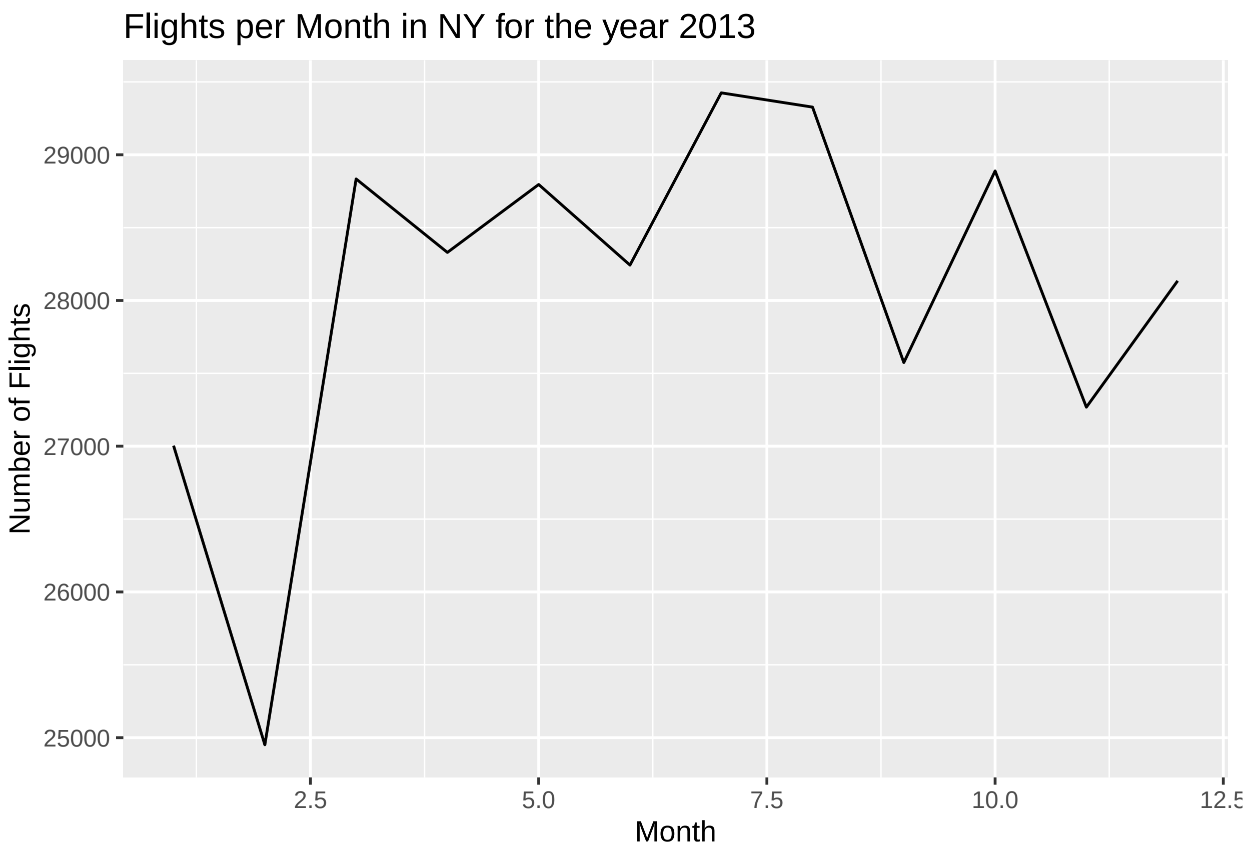
```
## # A tibble: 4,044 x 2
##    tailnum num_flights
##    <chr>         <int>
##  1 <NA>           2512
##  2 N725MQ          575
##  3 N722MQ          513
##  4 N723MQ          507
##  5 N711MQ          486
##  6 N713MQ          483
##  7 N258JB          427
##  8 N298JB          407
##  9 N353JB          404
## 10 N351JB          402
## # ... with 4,034 more rows
```

```
# tail_num_flights_by_month <- flights %>% group_by(month, tailnum) %>% summarise(numFlightsByMonth = s
# arrange(.data = tail_num_flights_by_month, month)
#
# monthlyFlights <- tail_num_flights_by_month %>% group_by(month) %>% summarise(numFlights = n())
# monthlyFlights

flightsByMonth <- transmute(.data = flights, year, month, tailnum)
flightsByMonth <-
  flightsByMonth %>% group_by(month) %>% summarize(sumOfFlights = n())
flightsByMonth
```

```
## # A tibble: 12 x 2
##    month sumOfFlights
##    <int>        <int>
##  1     1        27004
##  2     2        24951
##  3     3        28834
##  4     4        28330
##  5     5        28796
##  6     6        28243
##  7     7        29425
##  8     8        29327
##  9     9        27574
## 10    10        28889
## 11    11        27268
## 12    12        28135
```

```
ggplot(data = flightsByMonth, mapping = aes(x = month, y = sumOfFlights)) +
  geom_line() +
  ggtitle("Flights per Month in NY for the year 2013") +
  xlab("Month") + ylab("Number of Flights")
```



The plane with the tail number that flown the most out of the New York City airports is N725MQ.