Data Analysis

Tyler Kephart

2024-08-09

Question 1. Using the nycflights13 dataset, find all flights that departed in July, August, or September using the helper function between().

```
# filtering to flights that did depart in the months of July, August, and September
flights %>%
    filter(!is.na(dep_time), between(month, 7, 9))
```

```
## # A tibble: 84,448 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     7     1        1           2029       212      236           2359
##  2  2013     7     1        2           2359         3      344            344
##  3  2013     7     1       29           2245       104      151              1
##  4  2013     7     1       43           2130       193      322             14
##  5  2013     7     1       44           2150       174      300            100
##  6  2013     7     1       46           2051       235      304           2358
##  7  2013     7     1       48           2001       287      308           2305
##  8  2013     7     1       58           2155       183      335             43
##  9  2013     7     1      100           2146       194      327             30
## 10  2013     7     1      100           2245       135      337            135
## # i 84,438 more rows
## # i 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. Using the nycflights13 dataset sort flights to find the 10 flights that flew the furthest. Put them in order of fastest to slowest.

```
flights %>%
    # bringing distance and air_time forward to see them
    select(distance, air_time, everything()) %>%
    # filtering to max because there are more than 300 flights at max distance
    filter(distance == max(distance)) %>%
    # arranging to air_time, fastest to slowest, showing only top 10
    arrange(air_time) %>%
    head(10)
```

```
## # A tibble: 10 x 19
##    distance air_time  year month   day dep_time sched_dep_time dep_delay
##       <dbl>    <dbl> <int> <int> <int>    <int>          <int>     <dbl>
##  1     4983      580  2013     5     7      959           1000        -1
##  2     4983      580  2013     6     6     1044           1000        44
##  3     4983      580  2013     9    29      957           1000        -3
##  4     4983      581  2013     6     7      952           1000        -8
##  5     4983      582  2013     6     8      951           1000        -9
##  6     4983      582  2013     9     6      955           1000        -5
##  7     4983      584  2013     2    26     1000            900        60
##  8     4983      584  2013     5     6      956           1000        -4
##  9     4983      584  2013     9    28      955           1000        -5
## 10     4983      585  2013     7     3      957           1000        -3
## # i 11 more variables: arr_time <int>, sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Question 3. Using the nycflights13 dataset, calculate a new variable called "hr_delay" and arrange the flights dataset in order of the arrival delays in hours (longest delays at the top). Put the new variable you created just before the departure time.Hint: use the experimental argument .before.

```
flights %>%
    # create hr_delay by dividing arr_delay by 60 since it's in minutes
    mutate(hr_delay = arr_delay/60) %>%
    # sort longest delays to shortest
    arrange(desc(hr_delay)) %>%
    # move the hr_delay column in front of dep_time
    relocate(hr_delay, .before = dep_time)
```

```
## # A tibble: 336,776 x 20
##     year month   day hr_delay dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <dbl>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9     21.2      641            900      1301     1242
##  2  2013     6    15     18.8     1432           1935      1137     1607
##  3  2013     1    10     18.5     1121           1635      1126     1239
##  4  2013     9    20     16.8     1139           1845      1014     1457
##  5  2013     7    22     16.5      845           1600      1005     1044
##  6  2013     4    10     15.5     1100           1900       960     1342
##  7  2013     3    17     15.2     2321            810       911      135
##  8  2013     7    22     14.9     2257            759       898      121
##  9  2013    12     5     14.6      756           1700       896     1058
## 10  2013     5     3     14.6     1133           2055       878     1250
## # i 336,766 more rows
## # i 12 more variables: sched_arr_time <int>, 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 4. Using the nycflights13 dataset, find the most popular destinations (those with more than 2000 flights) and show the destination, the date info, the carrier. Then show just the number of flights for each popular destination.

```r
# get count of most popular destinations
pop_dest <- flights %>%
                count(dest) %>%
                filter(n > 2000)
# filter to most popular destinations with date and carrier
flights %>%
    filter(dest %in% pop_dest$dest) %>%
    distinct(dest, year, month, day, carrier)
```

```
## # A tibble: 59,206 x 5
##    dest   year month   day carrier
##    <chr> <int> <int> <int> <chr>
##  1 IAH    2013     1     1 UA
##  2 MIA    2013     1     1 AA
##  3 ATL    2013     1     1 DL
##  4 ORD    2013     1     1 UA
##  5 FLL    2013     1     1 B6
##  6 IAD    2013     1     1 EV
##  7 MCO    2013     1     1 B6
##  8 ORD    2013     1     1 AA
##  9 PBI    2013     1     1 B6
## 10 TPA    2013     1     1 B6
## # i 59,196 more rows
```

```r
# number of flights for each popular destination
pop_dest %>%
    arrange(desc(n))
```

```
## # A tibble: 46 x 2
##    dest      n
##    <chr> <int>
##  1 ORD   17283
##  2 ATL   17215
##  3 LAX   16174
##  4 BOS   15508
##  5 MCO   14082
##  6 CLT   14064
##  7 SFO   13331
##  8 FLL   12055
##  9 MIA   11728
## 10 DCA    9705
## # i 36 more rows
```

Question 5. Using the nycflights13 dataset, find the flight information (flight number, origin, destination, carrier, number of flights in the year, and percent late) for the flight numbers with the highest percentage of arrival delays. Only include the flight numbers that have over 100 flights in the year.

```
# get count of flight numbers
flight_nums <- flights %>%
                    group_by(flight, origin, dest, carrier) %>%
                    summarise(num_flights = n())
```

```
## ‘summarise()‘ has grouped output by ’flight’, ’origin’, ’dest’. You can
## override using the ‘.groups‘ argument.
```

```
# get count of arrival delays for flight numbers
flight_late <- flights %>%
                    group_by(flight, origin, dest, carrier) %>%
                    filter(arr_delay > 0) %>%
                    summarise(num_late = n())
```

```
## ‘summarise()‘ has grouped output by ’flight’, ’origin’, ’dest’. You can
## override using the ‘.groups‘ argument.
```

```
# join them, filter 100+ flights, get percent late, sort highest delay
full_join(flight_nums, flight_late) %>%
    filter(num_flights > 100) %>%
    transmute(flight, origin, dest, carrier, num_flights,
              perc_late = num_late / num_flights) %>%
    arrange(desc(perc_late))
```

```
## Joining with ‘by = join_by(flight, origin, dest, carrier)‘
```

```
## # A tibble: 1,114 x 6
## # Groups:   flight, origin, dest [1,113]
##    flight origin dest  carrier num_flights perc_late
##     <int> <chr>  <chr> <chr>         <int>     <dbl>
##  1    425 JFK    TPA   B6              101     0.802
##  2    985 LGA    TPA   B6              170     0.776
##  3   3075 JFK    CVG   MQ              162     0.710
##  4    527 EWR    MCO   B6              311     0.688
##  5   1103 JFK    SJU   B6              137     0.686
##  6   1201 JFK    FLL   B6              139     0.683
##  7   3616 LGA    MSP   MQ              127     0.677
##  8   4224 EWR    MKE   EV              257     0.677
##  9    381 LGA    FLL   B6              170     0.676
## 10   3433 JFK    DCA   MQ              111     0.676
## # i 1,104 more rows
```