Assignment 3: Flights of New York

Jordan HutchersonqView

2024-02-02

Exercise 1

- The dataset has 336,766 rows and 16 columns.
- A single row represents the data for a single flight, including details like time, date, flight number, destination and airports.
- The difference between them is when one scheduled arrive and the arrival time. Difference would give you the delay or early arrival.
- To identify individual airplane would be using the tailnumber

Exercise 2

```
flights %>%
  select(year, month)
```

```
## # A tibble: 336,776 x 2
##     year month
##    <int> <int>
##  1  2013     1
##  2  2013     1
##  3  2013     1
##  4  2013     1
##  5  2013     1
##  6  2013     1
##  7  2013     1
##  8  2013     1
##  9  2013     1
## 10  2013     1
## # i 336,766 more rows
```

- The result of the output is reducing the specified columns and using only two rows instead of 336,776 rows.

Exercise 3

```
flights %>%
  select(year:day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # i 336,766 more rows
```

- The colon is used to tell R to select all columns starting from the first one mentioned to the last one, including those two columns.

Exercise 4

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

```
## # 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     2     3     2153           2129        24     2247           2224
##  4  2013     2    12     2123           2130        -7     2211           2225
##  5  2013     3     8     2026           1935        51     2131           2056
##  6  2013    12     6      922            851        31     1021            954
##  7  2013     2     5     1303           1315       -12     1342           1411
##  8  2013     3    18     1456           1329        87     1533           1426
##  9  2013     3    19     2226           2145        41     2305           2246
## 10  2013     5     8       16           2159       137       53           2304
## # i 336,766 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>
```

- Yes both columns were sorted. Air time has appeared first because it is the smallest value. If you reverse it will sort the data by distance and then sort within each distance by air time.

Exercise 5

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

```
## # 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     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     6    27      959           1900       899     1236           2226
##  9  2013     7    22     2257            759       898      121           1026
## 10  2013    12     5      756           1700       896     1058           2020
## # i 336,766 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>
```

Exercise 6

```
flights %>%
  mutate(
    average_speed = distance / (air_time / 60)
  )
```

```
## # 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
## # i 336,766 more rows
## # i 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>, average_speed <dbl>
```

- The new column shows up at the end and the name is average_speed. The code controlling the name is average_speed within the mutate function.

Exercise 7

```
flights %>%
  mutate(
    dep_time_hour = dep_time %/% 100,
    dep_time_minute = dep_time %% 100,
    dep_time_minutes_midnight = (dep_time_hour * 60) + dep_time_minute
  )
```

```
## # A tibble: 336,776 x 22
##     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
## # i 336,766 more rows
## # i 14 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>, dep_time_hour <dbl>,
## #   dep_time_minute <dbl>, dep_time_minutes_midnight <dbl>
```

Exercise 8

```
flights %>%
  filter(
    arr_delay < 0 & carrier == "AA"
  )
```

```
## # A tibble: 20,769 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      606            610        -4      858            910
##  2  2013     1     1      628            630        -2     1137           1140
##  3  2013     1     1      656            659        -3      949            959
##  4  2013     1     1      659            700        -1     1008           1015
##  5  2013     1     1      712            715        -3     1023           1035
##  6  2013     1     1      739            745        -6      918            930
##  7  2013     1     1      753            755        -2     1056           1110
##  8  2013     1     1      803            810        -7      903            925
##  9  2013     1     1      840            845        -5     1311           1350
## 10  2013     1     1      940            945        -5     1119           1130
## # i 20,759 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>
```

Exercise 9

```
flights %>%
  group_by(carrier) %>%
  summarize(
    average_arr_delay = mean(arr_delay, na.rm = TRUE)
  )
```

```
## # A tibble: 16 x 2
##    carrier average_arr_delay
##    <chr>               <dbl>
##  1 9E                  7.38
##  2 AA                  0.364
##  3 AS                 -9.93
##  4 B6                  9.46
##  5 DL                  1.64
##  6 EV                 15.8
##  7 F9                 21.9
##  8 FL                 20.1
##  9 HA                 -6.92
## 10 MQ                 10.8
## 11 OO                 11.9
## 12 UA                  3.56
## 13 US                  2.13
## 14 VX                  1.76
## 15 WN                  9.65
## 16 YV                 15.6
```

- Carrier f9 had the longest delay while carrier AS had the shortest delay

```
flights %>%
  group_by(carrier) %>%
  summarize(
    average_arr_delay = mean(arr_delay, na.rm = TRUE),
    average_dep_delay = mean(arr_delay, na.rm = TRUE)

  )
```

```
## # A tibble: 16 x 3
##    carrier average_arr_delay average_dep_delay
##    <chr>               <dbl>             <dbl>
##  1 9E                  7.38              7.38
##  2 AA                  0.364             0.364
##  3 AS                 -9.93             -9.93
##  4 B6                  9.46              9.46
##  5 DL                  1.64              1.64
##  6 EV                 15.8              15.8
##  7 F9                 21.9              21.9
##  8 FL                 20.1              20.1
##  9 HA                 -6.92             -6.92
## 10 MQ                 10.8              10.8
## 11 OO                 11.9              11.9
## 12 UA                  3.56              3.56
## 13 US                  2.13              2.13
## 14 VX                  1.76              1.76
## 15 WN                  9.65              9.65
## 16 YV                 15.6              15.6
```

Exercise 10

```
late_flights_to_miami <- flights %>%
  filter(dest == "MIA", arr_delay > 0) %>%
  select(arr_delay, carrier)
```

Exercise 11

```
monthly_delays <- flights %>%
  group_by(month, carrier) %>%
  summarize(
    arrival_delay = mean(arr_delay, na.rm = TRUE),
    .groups = "drop"
```

```
  ) %>%
  spread(carrier, arrival_delay) %>%
  select(-`9E`)
```

```
pivoted_monthly_delays <- monthly_delays %>%
  pivot_longer(cols = -month, names_to = "airline_code", values_to = "delay")

qplot(
  x = month,
  y = delay,
  color = airline_code,
  geom = "line",
  data = pivoted_monthly_delays
)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```