Assignment 3: Flights of New York

JI WON MOK

2022-02-12

Exercise 1

i. How many rows and columns does this dataset have?

: 336776 rows and 21 columns

ii. What does a single row in this dataset represent?

: On-time data for flights that departed NYC

iii. What is the difference between the information contained in the arr_time and sched_arr_time columns? (Take a look at the column descriptions)

:arr_time is actual arrival times while sched_arr_time is scheduled arrival time.

iv. Airplanes are reused across many different flights. Which column(s) would be helpful to use in identifying individual airplanes?

: carrier

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
## # ... with 336,766 more 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
## # ... with 336,766 more rows
```

What does the colon : do?

colon ':' seems to work simliar to 'to' meaning year to day which represents year, month, and day.

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
## # ... 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>
```

i. Based on the output, answer the following questions. Does it look like both the air_time and distance columns were sorted?

: No, they are not excluded from the columns.

ii. Which column was sorted first? What happens if you reverse the order you specify the columns in arrange()?

: 'year' column is sorted first. reversing order does not have any impact on the actual order in the columns as you can see below.

```
flights %>%
  arrange(distance, 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     7    27       NA            106        NA       NA            245
##  2  2013     2     3     2153           2129        24     2247           2224
##  3  2013     2    12     2123           2130        -7     2211           2225
##  4  2013     1     6     2125           2129        -4     2224           2224
##  5  2013     1    23     2128           2129        -1     2221           2224
##  6  2013     2    10     2127           2129        -2     2209           2224
##  7  2013     2     1     2128           2129        -1     2216           2224
##  8  2013     3    30     1942           1950        -8     2026           2044
##  9  2013     1     7     2124           2129        -5     2212           2224
## 10  2013     1    14     2128           2129        -1     2215           2224
## # ... 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>
```

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
## # ... 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>
```

what flight experienced the longest departure delay?

-> HA

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
## # ... 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>,
## #   average_speed <dbl>
```

i.Where does the new column you just computed show up in the dataset and what is the name of this new column? -> right to the end, average_speed

ii. What part of the code is controlling the name of the new column? -> mutate()

Exercise 7

```
flights %>%
  mutate (
    dep_time_hour = dep_time %%100,
    dep_time_minute = dep_time %% 100,
    dep_time_minutes_midnight = dep_time_hour + 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
## # ... with 336,766 more rows, and 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>
```

modular arithmetic : %/% : integer division %% : remainder

Exercise 8

```
flights %>%
  filter(
    arr_delay < 0
  )
```

```
## # A tibble: 188,933 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      544            545        -1     1004           1022
##  2  2013     1     1      554            600        -6      812            837
##  3  2013     1     1      557            600        -3      709            723
##  4  2013     1     1      557            600        -3      838            846
##  5  2013     1     1      558            600        -2      849            851
##  6  2013     1     1      558            600        -2      853            856
##  7  2013     1     1      558            600        -2      923            937
##  8  2013     1     1      559            559         0      702            706
##  9  2013     1     1      559            600        -1      854            902
## 10  2013     1     1      600            600         0      851            858
## # ... with 188,923 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>
```

```
flights %>%
  filter(
    carrier == "AA"
  )
```

```
## # A tibble: 32,729 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      542            540         2      923            850
##  2  2013     1     1      558            600        -2      753            745
##  3  2013     1     1      559            600        -1      941            910
##  4  2013     1     1      606            610        -4      858            910
##  5  2013     1     1      623            610        13      920            915
##  6  2013     1     1      628            630        -2     1137           1140
##  7  2013     1     1      629            630        -1      824            810
##  8  2013     1     1      635            635         0     1028            940
##  9  2013     1     1      656            700        -4      854            850
## 10  2013     1     1      656            659        -3      949            959
## # ... with 32,719 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>
```

Tables show all the flights operated by American Airlines (airline code: AA) that arrived early

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
```

i. **Which airline carrier had the longest arrival delays on average? Which airline carrier had the**

    FL/AS

ii. Copy the previous code chunk and add a line of code within the summarize function to also calculate the average departure delay (i.e. the output of the summarize function should display the average departure and arrival delays for all carriers).

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

```
## # A tibble: 16 x 3
##    carrier average_arr_delay average_dep_delay
##    <chr>               <dbl>             <dbl>
##  1 9E                  7.38              16.7 
##  2 AA                  0.364              8.59
##  3 AS                 -9.93               5.80
##  4 B6                  9.46              13.0 
##  5 DL                  1.64               9.26
##  6 EV                 15.8               20.0 
##  7 F9                 21.9               20.2 
##  8 FL                 20.1               18.7 
##  9 HA                 -6.92               4.90
## 10 MQ                 10.8               10.6 
## 11 OO                 11.9               12.6 
## 12 UA                  3.56              12.1 
## 13 US                  2.13               3.78
## 14 VX                  1.76              12.9 
## 15 WN                  9.65              17.7 
## 16 YV                 15.6               19.0
```

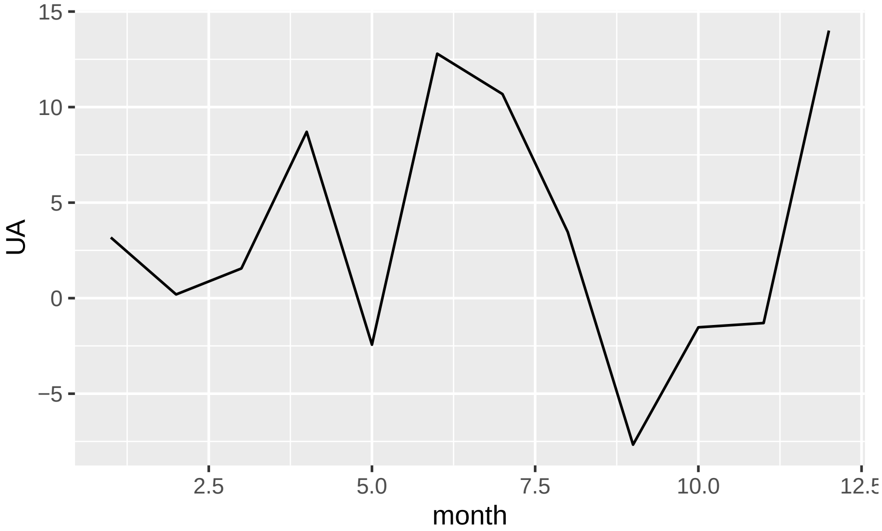
Exercise 10

```
flights_to_miami <- flights %>%
  filter(dest == "MIA")
late_flights_to_miami <- flights %>%
  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')
```

```
qplot( x= month, y = UA, geom="line", data=monthly_delays)
```



If you want line graph, you should include geom="line" argument. To make it easier, *a tidy format* -> pivot_longer function.

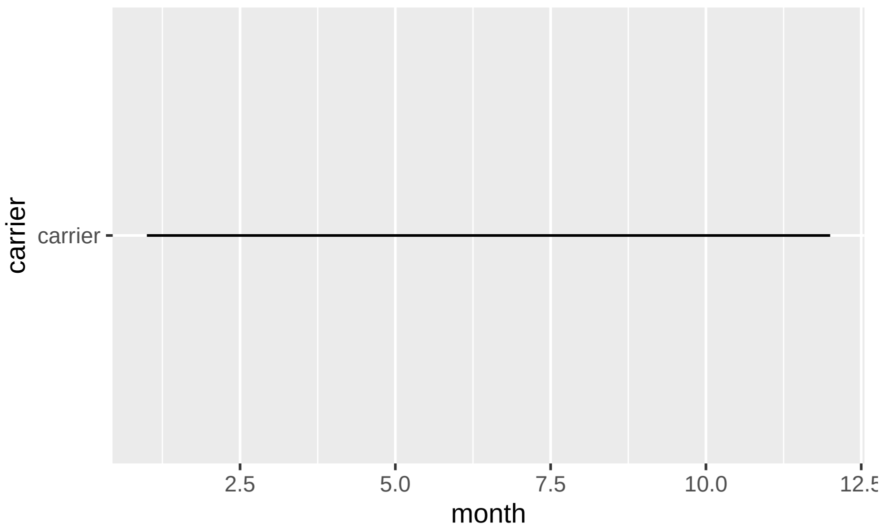
i. pivot_longer all 15 airline columns in the monthly_delays dataframe into two columns -> 3columns and 180rows

```
monthly_delays %>%
  pivot_longer(
    -month,
    names_to     = c("Arlines"),
    values_to    = "delays",
  )
```

```
## # A tibble: 180 x 3
##    month Arlines  delays
##    <int> <chr>     <dbl>
##  1     1 AA        0.982
##  2     1 AS        8.97
##  3     1 B6        4.72
##  4     1 DL       -4.40
##  5     1 EV       25.2
##  6     1 F9       21.8
##  7     1 FL        3.32
##  8     1 HA       27.5
##  9     1 MQ        7.88
## 10     1 OO      107
## # ... with 170 more rows
```

```
pivoted_monthly_delays <- monthly_delays %>%
  pivot_longer(-month, names_to = 'Arlines', values_to = 'delays')

qplot(x = month,
  y = 'carrier',
  color = ,
  geom ="line",
  data = pivoted_monthly_delays
  )
```


carrier
carrier
month