Week 5 DS Recitation: Exploratory Data Analysis

SIBDS 2024 @ Columbia

27 June, 2024

Getting Started

1. Create a new R project and named it `week5_DS_recitation`
2. Put the `week5_DS_recitation_EDA.Rmd` file into the same folder of the R project you created.

EDA with `nycflights13`

We are going to use the `nycflights13` package again. Please load the `flights` data from the `nycflights13` package using:

```
if (!requireNamespace("nycflights13", quietly = TRUE)) {
  install.packages("nycflights13")
}

library(nycflights13)
data("flights")
```

Tasks:

1. Which plane (`tailnum`) has the worst on-time record (remove planes that flew `<` 20 flights)? To find out, you may create an indicator to determine the on-time record with the code `mutate(on_time = !is.na(arr_time) & (arr_delay <= 0))`, or you can consider the average number of minutes delayed with `mean(arr_delay)`.

```
# Your answer starts here
flights %>%
  filter(!is.na(tailnum), !is.na(arr_time), !is.na(arr_delay))
```

```
## # A tibble: 327,346 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
## # i 327,336 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>
```

```
# using mean arr_delay
flights %>%
  filter(!is.na(tailnum), !is.na(arr_time), !is.na(arr_delay))
```

```
## # A tibble: 327,346 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
## # i 327,336 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>
```

2. What time of day (`hour`) should you fly if you want to avoid delays as much as possible?

```
# Your answer starts here
```

3. Delays are typically temporally correlated: even once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using `lag()`, explore how the delay of a flight is related to the delay of the immediately preceding flight, please use a plot to display the relationship. .

```
# Your answer starts here
lagged_delays <- flights %>%
  arrange(origin, month, day, dep_time) %>%
  group_by(origin)
```