FLIGHT DELAY ANALYSIS

by Markus

2022-09-30

INTRODUCTION

This project is intended to demonstrate the skills acquired from the Google Data Analytics Certificate Course hosted on COURSERA. The data set was retrieved from KAGGLE. Originally, the data set comes form the U.S. DEPARTMENT OF TRANSPORTATION'S (DOT) BUREAU OF TRANSPORTATION STATISTICS (BTS).

A description for the original column labels can be looked up by clicking the following LINK.

The attempt to analyze the data set in a Spreadsheet (Excel) failed due to its high volume. I personally decided to use R over SQL because R is more functional and also allows me to visualize the data.

GENERAL ANALYSIS

DATA PREPARATION

1 LOADING THE REQUIRED PACKAGES FOR THE ANALYSIS

If the packages are not installed yet, use the install.packages() function first!

Note that the library plyr has to be loaded prior to dplyr to prevent any issues

```
LIBRARY(TIDYVERSE)
LIBRARY(JANITOR)
DETACH("PACKAGE:PLYR") # DETACHING BOTH LIBRARIES ...
DETACH("PACKAGE:DPLYR")
LIBRARY(PLYR) # ... AND LOADING THEM AGAIN TO MAKE SURE
LIBRARY(DPLYR) # THEY ARE LOADED IN THE RIGHT ORDER
LIBRARY(READR)
LIBRARY(LUBRIDATE)
LIBRARY(GGCORRPLOT)
LIBRARY(RCOLORBREWER)
LIBRARY(SQLDF)
LIBRARY(SCALES)
LIBRARY(GGPUBR)
LIBRARY(GGCORRPLOT)
```

---

2 OPENING THE DATA SET

```
LOCAL_PATH <- ".../FLIGHT_DELAY.CSV"
FLIGHTS_DF <- READ_CSV(LOCAL_PATH)

## # A TIBBLE: 6 × 29
##   DAYOFW…¹ DATE  DEPTIME ARRTIME CRSAR…² UNIQU…³ AIRLINE FLIGH…⁴ TAILNUM ACTUA…
⁵
##      <DBL> <CHR>   <DBL>   <DBL>   <DBL> <CHR>   <CHR>     <DBL> <CHR>     <DBL
```

```
>
## 1          4 03-0…    1829    1959    1925 WN      SOUTHW…    3920 N464WN       9
0
## 2          4 03-0…    1937    2037    1940 WN      SOUTHW…     509 N763SW      24
0
## 3          4 03-0…    1644    1845    1725 WN      SOUTHW…    1333 N334SW      12
1
## 4          4 03-0…    1452    1640    1625 WN      SOUTHW…     675 N286WN      22
8
## 5          4 03-0…    1323    1526    1510 WN      SOUTHW…       4 N674AA      12
3
## 6          4 03-0…    1416    1512    1435 WN      SOUTHW…      54 N643SW       5
6
## # … WITH 19 MORE VARIABLES: CRSELAPSEDTIME <DBL>, AIRTIME <DBL>,
## #   ARRDELAY <DBL>, DEPDELAY <DBL>, ORIGIN <CHR>, ORG_AIRPORT <CHR>,
## #   DEST <CHR>, DEST_AIRPORT <CHR>, DISTANCE <DBL>, TAXIIN <DBL>,
## #   TAXIOUT <DBL>, CANCELLED <DBL>, CANCELLATIONCODE <CHR>, DIVERTED <DBL>,
## #   CARRIERDELAY <DBL>, WEATHERDELAY <DBL>, NASDELAY <DBL>,
## #   SECURITYDELAY <DBL>, LATEAIRCRAFTDELAY <DBL>, AND ABBREVIATED VARIABLE
## #   NAMES ¹DAYOFWEEK, ²CRSARRTIME, ³UNIQUECARRIER, ⁴FLIGHTNUM, …
```

---

3 FOR THE SAKE OF VISUAL APPEAL, I RENAMED THE COLUMN NAMES AND CONVERTED THEM ALL TO LOWERCASE

```
NAMES(FLIGHTS_DF) <- TOLOWER(NAMES(FLIGHTS_DF %>%
                        DPLYR::RENAME(WEEKDAY = DAYOFWEEK,
                                      DEP_TIME = DEPTIME,
                                      ARR_TIME = ARRTIME,
                                      SCHEDULED_ARR_TIME = CRSARRTIME,
                                      UNIQ_CARRIER_CODE = UNIQUECARRIER,
                                      FLIGHT_NUM = FLIGHTNUM,
                                      TAIL_NUM = TAILNUM,
                                      ACTUAL_FLIGHT_TIME_MIN = ACTUALELAPSEDTIME,
                                      ESTIMATE_FLIGHT_TIME_MIN = CRSELAPSEDTIME,
                                      AIR_TIME_MIN = AIRTIME,
                                      ARR_DELAY = ARRDELAY,
                                      DEP_DELAY = DEPDELAY,
                                      DEP_AIRPORT_CODE = ORIGIN,
                                      DEP_AIRPORT = ORG_AIRPORT,
                                      DEST_AIRPORT_CODE = DEST,
                                      DEST_AIRPORT = DEST_AIRPORT,
                                      DISTANCE_MILES = DISTANCE,
                                      LANDING_TO_GATE_MIN = TAXIIN,
                                      GATE_TO_TAKEOFF_MIN =TAXIOUT,
                                      CANCELLATION_CAUSE_CODE = CANCELLATIONCODE,
                                      CARRIER_DELAY = CARRIERDELAY,
                                      WEATHER_DELAY = WEATHERDELAY,
                                      NAS_DELAY = NASDELAY,
                                      SECURITY_DELAY = SECURITYDELAY,
                                      LATE_AIRCRAFT_DELAY = LATEAIRCRAFTDELAY)))
```

```
##  [1] "WEEKDAY"                  "DATE"
##  [3] "DEP_TIME"                 "ARR_TIME"
##  [5] "SCHEDULED_ARR_TIME"       "UNIQ_CARRIER_CODE"
##  [7] "AIRLINE"                  "FLIGHT_NUM"
##  [9] "TAIL_NUM"                 "ACTUAL_FLIGHT_TIME_MIN"
## [11] "ESTIMATE_FLIGHT_TIME_MIN" "AIR_TIME_MIN"
```

```
## [13] "ARR_DELAY"                   "DEP_DELAY"
## [15] "DEP_AIRPORT_CODE"            "DEP_AIRPORT"
## [17] "DEST_AIRPORT_CODE"           "DEST_AIRPORT"
## [19] "DISTANCE_MILES"              "LANDING_TO_GATE_MIN"
## [21] "GATE_TO_TAKEOFF_MIN"         "CANCELLED"
## [23] "CANCELLATION_CAUSE_CODE"     "DIVERTED"
## [25] "CARRIER_DELAY"               "WEATHER_DELAY"
## [27] "NAS_DELAY"                   "SECURITY_DELAY"
## [29] "LATE_AIRCRAFT_DELAY"
```

---

4 NEXT, WE REMOVE COLUMNS THAT DON'T GIVE US ANY INFORMATION (DUE TO A LACK OF DATA)

```
VECTOR <- C()
FOR (I IN NAMES(FLIGHTS_DF)) {
  IF (IS_DOUBLE(FLIGHTS_DF[[I]][2]) == TRUE) {
    IF (SUM(FLIGHTS_DF[I]) == 0 ) {
      VECTOR <- APPEND(VECTOR, I)
    }
  }
}

## THE VECTOR CONTAINS COLUMNS:
## -CANCELLED
## -DIVERTED
```

When we investigate the columns "cancelled" and "diverted", they only contain 0!

Let's get rid of the two unnecessary columns (2 methods)

```
#METHOD 1: SELECTING ALL EXCEPT FROM ELEMENTS OF VECTOR
FLIGHTS_DF <- SELECT(FLIGHTS_DF, -ALL_OF(VECTOR))

#METHOD 2: DROPPING USELESS COLUMNS
FLIGHTS_DF <- FLIGHTS_DF[!(NAMES(FLIGHTS_DF) %IN% VECTOR)]
```

---

5 CREATING A NEW COLUMN THAT CONTAINS VALUES OF THE TOTAL DELAY FOR EACH SPECIFIC FLIGHT

```
FLIGHTS_DF <- MUTATE(FLIGHTS_DF,
                     TOTAL_DELAY = (CARRIER_DELAY + WEATHER_DELAY + NAS_DELAY +
                                    SECURITY_DELAY + LATE_AIRCRAFT_DELAY))
```

---

6 CREATING A NEW COLUMN THAT CONTAINS THE MONTH EACH INDIVIDUAL FLIGHT TOOK PLACE

```
LIBRARY(LUBRIDATE)

FLIGHTS_DF <- FLIGHTS_DF %>% MUTATE(MONTH = MONTH(DMY(DATE)))

##  [1] "WEEKDAY"                     "DATE"
##  [3] "DEP_TIME"                    "ARR_TIME"
##  [5] "SCHEDULED_ARR_TIME"          "UNIQ_CARRIER_CODE"
##  [7] "AIRLINE"                     "FLIGHT_NUM"
##  [9] "TAIL_NUM"                    "ACTUAL_FLIGHT_TIME_MIN"
## [11] "ESTIMATE_FLIGHT_TIME_MIN"    "AIR_TIME_MIN"
## [13] "ARR_DELAY"                   "DEP_DELAY"
## [15] "DEP_AIRPORT_CODE"            "DEP_AIRPORT"
## [17] "DEST_AIRPORT_CODE"           "DEST_AIRPORT"
```

```
## [19] "DISTANCE_MILES"          "LANDING_TO_GATE_MIN"
## [21] "GATE_TO_TAKEOFF_MIN"     "CANCELLATION_CAUSE_CODE"
## [23] "CARRIER_DELAY"           "WEATHER_DELAY"
## [25] "NAS_DELAY"               "SECURITY_DELAY"
## [27] "LATE_AIRCRAFT_DELAY"     "TOTAL_DELAY"
## [29] "MONTH"
```

---

7 DETERMINING, IN WHICH DELAY CATEGORY EACH FLIGHT FALLS

I classified the delay according to the Federal Aviation Administration (FAA) that considers an actual arrival less than 15 min after the scheduled arrival as not delayed, an arrival between 15 and 45 min after the scheduled arrival as "medium delay" and beyond 45 min as "large delay". Source: WIKIPEDIA

```
FLIGHTS_DF <- FLIGHTS_DF %>%
  MUTATE(DEGREE_DELAY =
           IFELSE(TOTAL_DELAY <= 15, "NO DELAY",
                  IFELSE(TOTAL_DELAY >= 45, "LARGE DELAY", "MEDIUM DELAY")))
```

Having learned Python as a first programming language, I love to write loops, functions and conditional statements. In this case, it was a tedious mistake to apply Python practices to R:

Technically, this can be done with a for-loop and conditional statements too; however, the computing time is awfully long with bigger data frames (30-40 min) since functions in R usually do not directly modify the data frame, but instead making copies. For every single iteration, R therefore makes a copy of the entire data frame! Fortunately, I found help on STACK OVERFLOW.

```
VEC <- C()
FOR (T IN FLIGHTS_DF$TOTAL_DELAY) {
  IF (T <= 15) {
    VEC <- APPEND(VEC, "NO DELAY")
  }
  IF (T >= 45) {
    VEC <- APPEND(VEC, "LARGE DELAY")
  }
  ELSE {
    VEC <- APPEND(VEC, "MEDIUM DELAY")
  }
}

# CREATING A NEW COLUMN FROM THE VECTOR CONTAINING
# THE CATEGORIZATION OF EACH FLIGHT
FLIGHTS_DF["DELAY_DEGREE"] <- VEC
```

---

8 THIS STEP IS MAINLY FOR THE SAKE OF PRACTICING DATA MANIPULATION

(This case does not apply to the US since it is an EU law):

creating a new column which states whether the passenger are potentially subject to compensation according to EU261 law. Passengers are eligible to claim up to 600€ as soon as the flight is delayed for 3 hours, and receive a full refund, if delayed for 5 hours or longer.

```
FLIGHTS_DF <- FLIGHTS_DF %>%
  MUTATE(COMPENSATION =
           IFELSE(TOTAL_DELAY < 180, "NO COMPENSATION",
                  IFELSE(TOTAL_DELAY >= 300, "FULL REFUND", "UP TO 600€")))
```

As with the previous step, this code using the for-loop is highly inefficient. I still left it because it is technically correct viewing it from a logical perspective :)

```
VECT <- C()
FOR (C IN FLIGHTS_DF$TOTAL_DELAY){
  IF (C < 180){
    VECT <- APPEND(VECT, "NO COMPENSATION")
  }
  IF (C >= 300){
    VECT <- APPEND(VECT, "FULL REFUND")
  }
  ELSE {
    VECT <- APPEND(VECT, "UP TO 600€")
  }
}
FLIGHTS_DF["COMPENSATION"] <- VECT
```

Let's have a look at the structure of our final data frame:

```
GLIMPSE(FLIGHTS_DF)

## ROWS: 484,551
## COLUMNS: 31
## $ WEEKDAY                   <DBL> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, …
## $ DATE                      <CHR> "03-01-2019", "03-01-2019", "03-01-2019", "03…
## $ DEP_TIME                  <DBL> 1829, 1937, 1644, 1452, 1323, 1416, 1657, 142…
## $ ARR_TIME                  <DBL> 1959, 2037, 1845, 1640, 1526, 1512, 1754, 165…
## $ SCHEDULED_ARR_TIME        <DBL> 1925, 1940, 1725, 1625, 1510, 1435, 1735, 161…
## $ UNIQ_CARRIER_CODE         <CHR> "WN", "WN", "WN", "WN", "WN", "WN", "WN", "WN…
## $ AIRLINE                   <CHR> "SOUTHWEST AIRLINES CO.", "SOUTHWEST AIRLINES…
## $ FLIGHT_NUM                <DBL> 3920, 509, 1333, 675, 4, 54, 623, 188, 362, 4…
## $ TAIL_NUM                  <CHR> "N464WN", "N763SW", "N334SW", "N286WN", "N674…
## $ ACTUAL_FLIGHT_TIME_MIN    <DBL> 90, 240, 121, 228, 123, 56, 57, 155, 147, 135…
## $ ESTIMATE_FLIGHT_TIME_MIN <DBL> 90, 250, 135, 240, 135, 70, 70, 195, 165, 145…
## $ AIR_TIME_MIN              <DBL> 77, 230, 107, 213, 110, 49, 47, 143, 134, 118…
## $ ARR_DELAY                 <DBL> 34, 57, 80, 15, 16, 37, 19, 47, 64, 72, 29, 2…
## $ DEP_DELAY                 <DBL> 34, 67, 94, 27, 28, 51, 32, 87, 82, 82, 56, 1…
## $ DEP_AIRPORT_CODE          <CHR> "IND", "IND", "IND", "IND", "IND", "ISP", "IS…
## $ DEP_AIRPORT               <CHR> "INDIANAPOLIS INTERNATIONAL AIRPORT", "INDIAN…
## $ DEST_AIRPORT_CODE         <CHR> "BWI", "LAS", "MCO", "PHX", "TPA", "BWI", "BW…
## $ DEST_AIRPORT              <CHR> "BALTIMORE-WASHINGTON INTERNATIONAL AIRPORT",…
## $ DISTANCE_MILES            <DBL> 515, 1591, 828, 1489, 838, 220, 220, 1093, 97…
## $ LANDING_TO_GATE_MIN       <DBL> 3, 3, 6, 7, 4, 2, 5, 6, 6, 6, 5, 7, 3, 3, 8, …
## $ GATE_TO_TAKEOFF_MIN       <DBL> 10, 7, 8, 8, 9, 5, 5, 6, 7, 11, 5, 8, 7, 7, 7…
## $ CANCELLATION_CAUSE_CODE   <CHR> "N", "N", "N", "N", "N", "N", "N", "N", "N", …
## $ CARRIER_DELAY             <DBL> 2, 10, 8, 3, 0, 12, 7, 40, 5, 3, 0, 0, 282, 2…
## $ WEATHER_DELAY             <DBL> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ NAS_DELAY                 <DBL> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, …
## $ SECURITY_DELAY            <DBL> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ LATE_AIRCRAFT_DELAY       <DBL> 32, 47, 72, 12, 16, 25, 12, 7, 59, 69, 29, 15…
## $ TOTAL_DELAY               <DBL> 34, 57, 80, 15, 16, 37, 19, 47, 64, 72, 29, 2…
## $ MONTH                     <DBL> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ DEGREE_DELAY              <CHR> "MEDIUM DELAY", "LARGE DELAY", "LARGE DELAY",…
## $ COMPENSATION              <CHR> "NO COMPENSATION", "NO COMPENSATION", "NO COM…
```

DATA EXPLORATION

```
FLIGHTS_DF %>%
    DPLYR::GROUP_BY(AIRLINE) %>%
    DROP_NA() %>%
    SUMMARIZE(ACCUMULATED_DELAY = SUM(TOTAL_DELAY)) %>%
    ARRANGE(-ACCUMULATED_DELAY)
```

```
## # A TIBBLE: 12 × 2
##    AIRLINE                        ACCUMULATED_DELAY
##    <CHR>                                      <DBL>
##  1 SOUTHWEST AIRLINES CO.                   6075370
##  2 AMERICAN AIRLINES INC.                   4801746
##  3 UNITED AIR LINES INC.                    3963975
##  4 AMERICAN EAGLE AIRLINES INC.             3772945
##  5 SKYWEST AIRLINES INC.                    3284415
##  6 US AIRWAYS INC.                          1856212
##  7 ATLANTIC SOUTHEAST AIRLINES              1812756
##  8 DELTA AIR LINES INC.                     1791817
##  9 JETBLUE AIRWAYS                          1119565
## 10 ALASKA AIRLINES INC.                      575576
## 11 FRONTIER AIRLINES INC.                    378393
## 12 HAWAIIAN AIRLINES INC.                     80148
```

So far, so good. But simply concluding that Southwest Airline Co. is the least reliable Airline would be *false* since Southwest operates the most flights in the given time period.

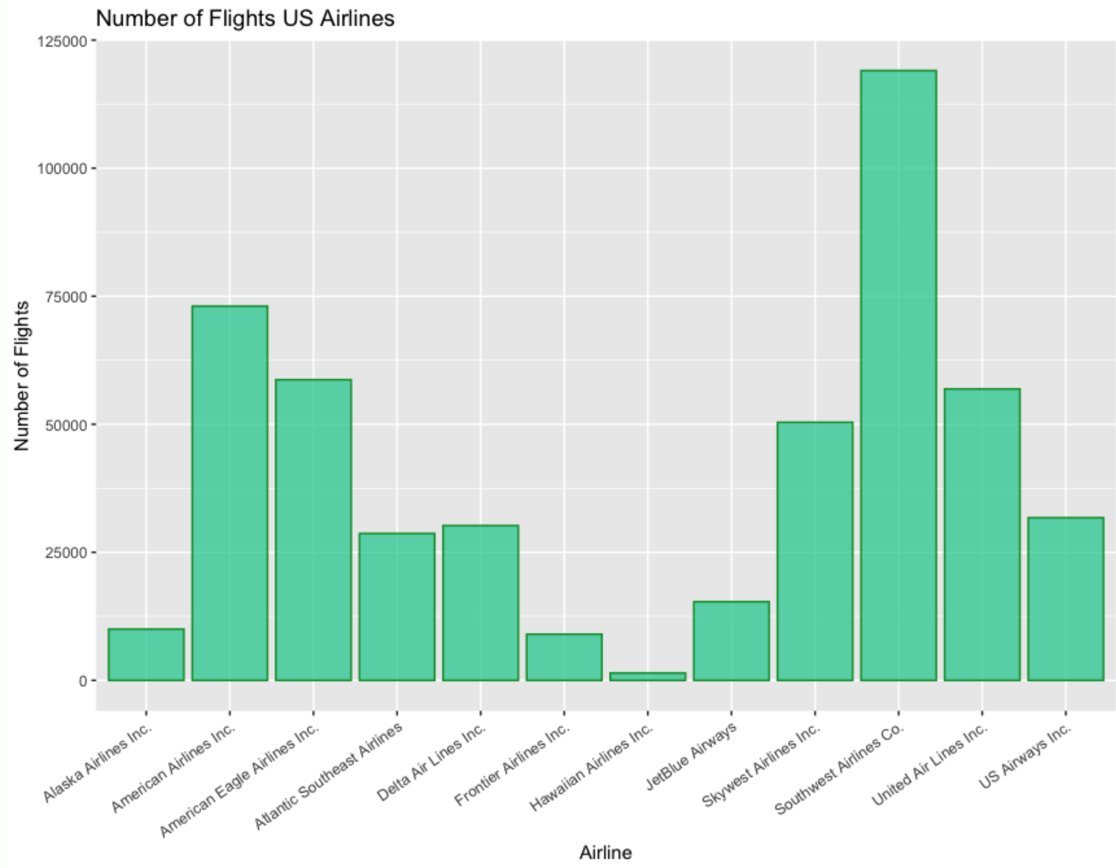
To demonstrate this, let's compute, and then display the number of flights of each individual airline.

```
AS.DATA.FRAME(TABLE(FLIGHTS_DF$AIRLINE)) %>% ARRANGE(-FREQ)
```

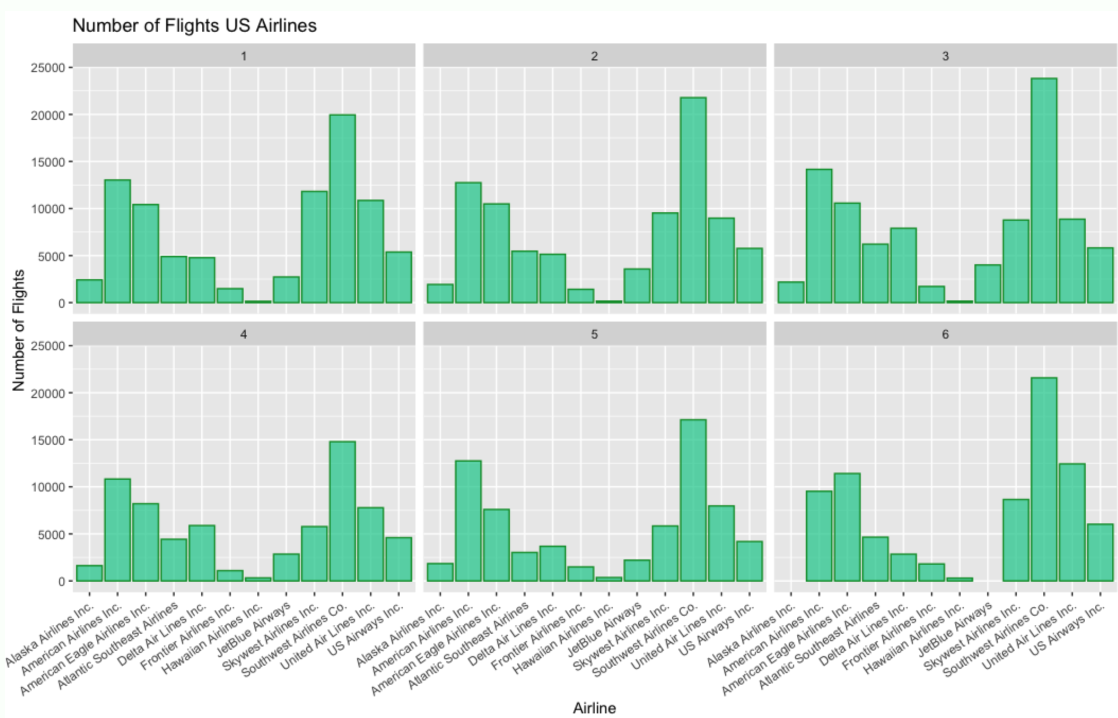
```
##                            VAR1   FREQ
## 1        SOUTHWEST AIRLINES CO. 119048
## 2        AMERICAN AIRLINES INC.  73053
## 3  AMERICAN EAGLE AIRLINES INC.  58698
## 4         UNITED AIR LINES INC.  56896
## 5         SKYWEST AIRLINES INC.  50384
## 6               US AIRWAYS INC.  31755
## 7          DELTA AIR LINES INC.  30220
## 8   ATLANTIC SOUTHEAST AIRLINES  28678
## 9               JETBLUE AIRWAYS  15364
## 10         ALASKA AIRLINES INC.  10000
## 11        FRONTIER AIRLINES INC.   9015
## 12        HAWAIIAN AIRLINES INC.   1440
```

ggplot2 is an awesome and handy package for data visualization

```
GGPLOT(FLIGHTS_DF) +
  GEOM_BAR(AES(X = AIRLINE), FILL = "#00CC99", COLOR = "#009933", ALPHA = 0.7) +
  THEME(AXIS.TEXT.X = ELEMENT_TEXT(ANGLE = 35, HJUST = 1)) +
  LABS(TITLE = "NUMBER OF FLIGHTS US AIRLINES",
        X = "AIRLINE", Y = "NUMBER OF FLIGHTS")
```




```
GGPLOT(FLIGHTS_DF) +
  GEOM_BAR(AES(X = AIRLINE), FILL = "#00CC99", COLOR = "#009933", ALPHA = 0.7) +
  THEME(AXIS.TEXT.X = ELEMENT_TEXT(ANGLE = 35, HJUST = 1)) +
  LABS(TITLE = "NUMBER OF FLIGHTS US AIRLINES",
       X = "AIRLINE", Y = "NUMBER OF FLIGHTS") +
  FACET_WRAP(~MONTH)
```



A better measure would be the average (or mean) delay for each airline.

```
FLIGHTS_DF %>%
  GROUP_BY(AIRLINE) %>%
  DROP_NA() %>%
```

```
  SUMMARIZE(DELAY = MEAN(TOTAL_DELAY)) %>%
  ARRANGE(-DELAY)
```

```
## # A TIBBLE: 12 × 2
##    AIRLINE                      DELAY
##    <CHR>                        <DBL>
##  1 JETBLUE AIRWAYS               72.9
##  2 UNITED AIR LINES INC.         69.7
##  3 AMERICAN AIRLINES INC.        65.7
##  4 SKYWEST AIRLINES INC.         65.2
##  5 AMERICAN EAGLE AIRLINES INC.  64.3
##  6 ATLANTIC SOUTHEAST AIRLINES   63.2
##  7 DELTA AIR LINES INC.          59.3
##  8 US AIRWAYS INC.               58.5
##  9 ALASKA AIRLINES INC.          57.6
## 10 HAWAIIAN AIRLINES INC.        55.7
## 11 SOUTHWEST AIRLINES CO.        51.0
## 12 FRONTIER AIRLINES INC.        42.0
```

---

10 NEXT, LET'S EXPLORE, WHAT IS THE BIGGEST DRIVER FOR DELAY?

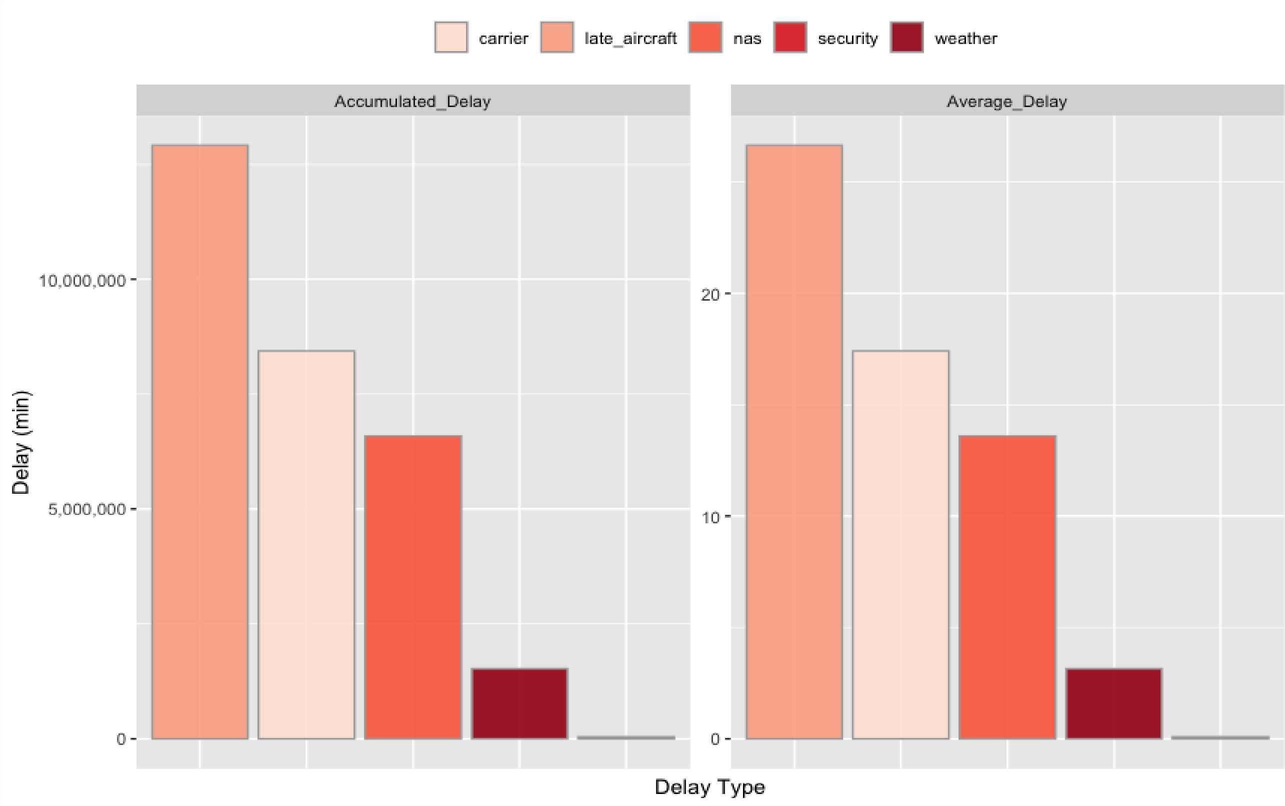
```
FLIGHTS_DF %>%  SUMMARIZE(TOTAL_CARRIER = SUM(CARRIER_DELAY),
                          TOTAL_WEATHER = SUM(WEATHER_DELAY),
                          TOTAL_NAS = SUM(NAS_DELAY),
                          TOTAL_SECURITY = SUM(SECURITY_DELAY),
                          TOTAL_LATE_AIRCRAFT = SUM(LATE_AIRCRAFT_DELAY)) %>%
  PIVOT_LONGER(COLS=1:5, NAMES_TO = 'DELAY_TYPE', VALUES_TO = 'ACCUMULATED_DELAY')
 %>%
  ARRANGE(-ACCUMULATED_DELAY)
```

```
## # A TIBBLE: 5 × 2
##   DELAY_TYPE          ACCUMULATED_DELAY
##   <CHR>                           <DBL>
## 1 TOTAL_LATE_AIRCRAFT          12915022
## 2 TOTAL_CARRIER                 8440607
## 3 TOTAL_NAS                     6589613
## 4 TOTAL_WEATHER                 1527927
## 5 TOTAL_SECURITY                  39749
```

Are we still getting the same ranting if we compare the accumulated delay of each delay type to the average delay?

```
DF1 <- FLIGHTS_DF %>%  SUMMARIZE(CARRIER = SUM(CARRIER_DELAY),
                          WEATHER = SUM(WEATHER_DELAY),
                          NAS = SUM(NAS_DELAY),
                          SECURITY = SUM(SECURITY_DELAY),
                          LATE_AIRCRAFT = SUM(LATE_AIRCRAFT_DELAY)) %>%
  PIVOT_LONGER(COLS=1:5, NAMES_TO = 'DELAY_TYPE', VALUES_TO = 'ACCUMULATED_DELAY')
 %>%
  ARRANGE(-ACCUMULATED_DELAY)

DF2 <- FLIGHTS_DF %>% SUMMARIZE(CARRIER = MEAN(CARRIER_DELAY),
                      WEATHER = MEAN(WEATHER_DELAY),
                      NAS = MEAN(NAS_DELAY),
                      SECURITY = MEAN(SECURITY_DELAY),
                      LATE_AIRCRAFT = MEAN(LATE_AIRCRAFT_DELAY)) %>%
  PIVOT_LONGER(COLS=1:5, NAMES_TO = 'DELAY_TYPE', VALUES_TO = 'AVERAGE_DELAY') %>%

  ARRANGE(-AVERAGE_DELAY)
```

```
#INNER JOIN OF BOTH DATA FRAMES BY THE PRIMARY KEY 'DELAY_TYPE'
MERGE(DF1, DF2) %>% ARRANGE(-AVERAGE_DELAY)

##      DELAY_TYPE ACCUMULATED_DELAY AVERAGE_DELAY
## 1 LATE_AIRCRAFT          12915022   26.65358652
## 2       CARRIER           8440607   17.41943985
## 3           NAS           6589613   13.59942091
## 4       WEATHER           1527927    3.15328417
## 5      SECURITY             39749    0.08203264

MERGE(DF1, DF2) %>%
  ARRANGE(-AVERAGE_DELAY) %>%
  PIVOT_LONGER(COLS = C("ACCUMULATED_DELAY", "AVERAGE_DELAY"),
               NAMES_TO ="METHOD", VALUES_TO = "VALUE") %>%
  GGPLOT() +
  GEOM_BAR(AES(X = REORDER(DELAY_TYPE, -VALUE), Y = VALUE, FILL = DELAY_TYPE),
           COLOR = "DARK GREY", ALPHA = 0.9, STAT="IDENTITY", POSITION = "DODGE")
+
  FACET_WRAP(~METHOD, SCALE = "FREE") +
  SCALE_Y_CONTINUOUS(LABELS = FORMAT_FORMAT(BIG.MARK = ",", SCIENTIFIC = FALSE)) +
  LABS(X = "DELAY TYPE", Y = "DELAY (MIN)", FILL = "") +
  THEME(LEGEND.POSITION="TOP", AXIS.TEXT.X = ELEMENT_BLANK(), AXIS.TICKS.X = ELEME
NT_BLANK()) +
  SCALE_FILL_BREWER(PALETTE = 14)
```



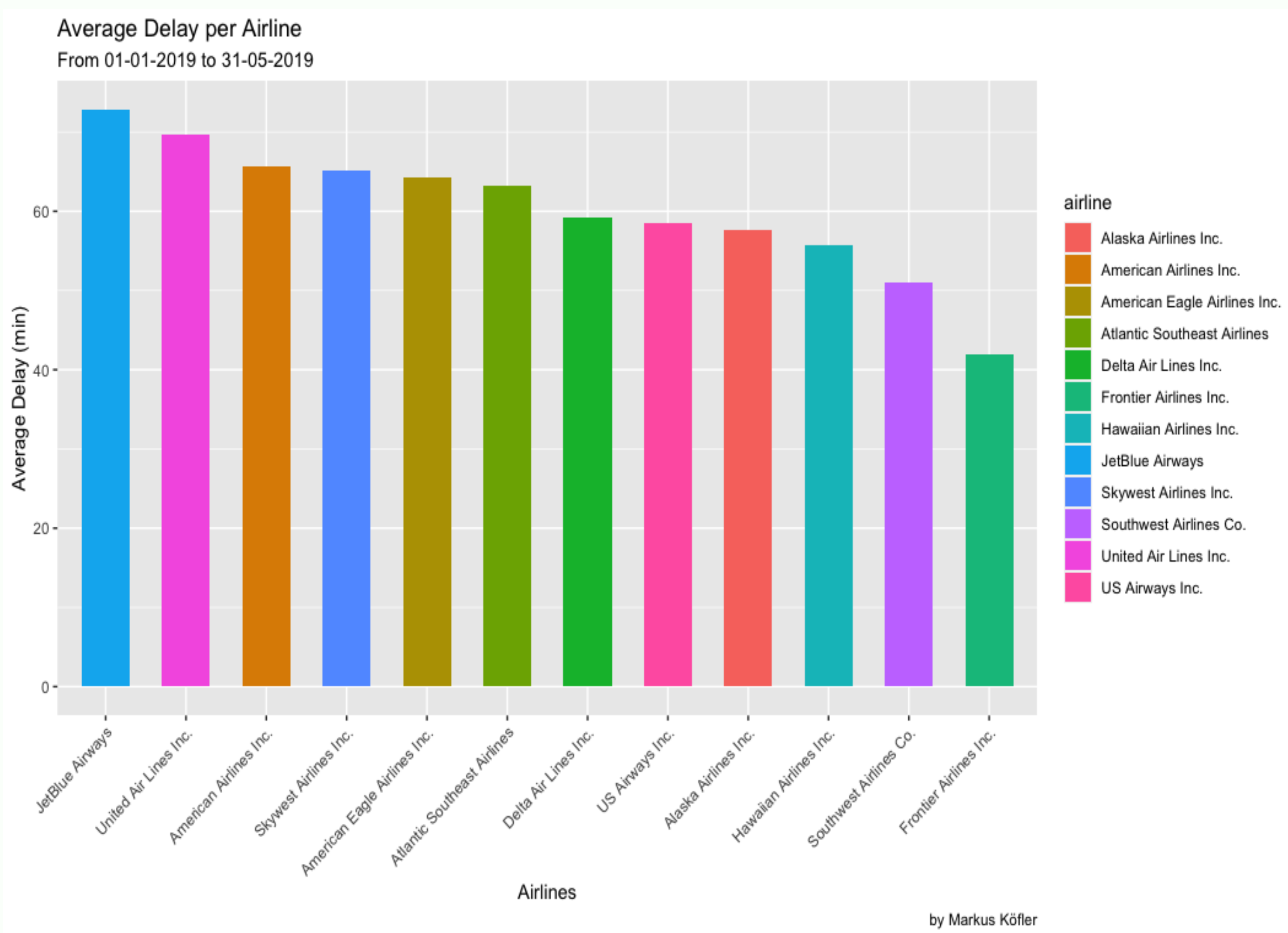
Let's come back to the average flight delay - How big are the differences in the average flight delay if we compare the 12 airlines to each other?

```
AVG <- FLIGHTS_DF %>%
  GROUP_BY(AIRLINE) %>%
  DROP_NA() %>%
  SUMMARIZE(DELAY = MEAN(TOTAL_DELAY)) %>%
  ARRANGE(-DELAY)

AVG
```

```
## # A TIBBLE: 12 × 2
##    AIRLINE                          DELAY
##    <CHR>                            <DBL>
##  1 JETBLUE AIRWAYS                   72.9
##  2 UNITED AIR LINES INC.             69.7
##  3 AMERICAN AIRLINES INC.            65.7
##  4 SKYWEST AIRLINES INC.             65.2
##  5 AMERICAN EAGLE AIRLINES INC.      64.3
##  6 ATLANTIC SOUTHEAST AIRLINES       63.2
##  7 DELTA AIR LINES INC.              59.3
##  8 US AIRWAYS INC.                   58.5
##  9 ALASKA AIRLINES INC.              57.6
## 10 HAWAIIAN AIRLINES INC.            55.7
## 11 SOUTHWEST AIRLINES CO.            51.0
## 12 FRONTIER AIRLINES INC.            42.0
```

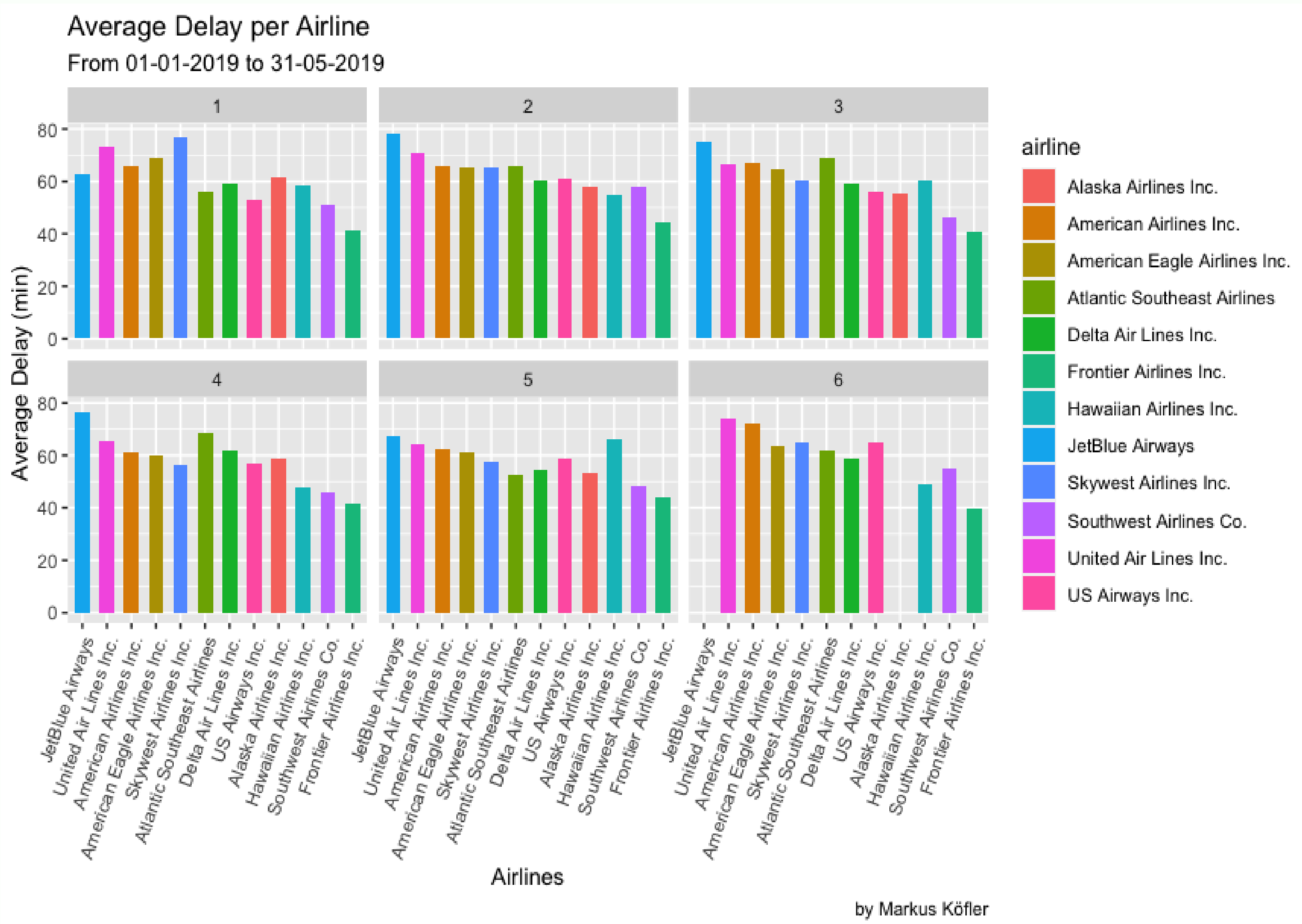
Let’s visualize the code by using another graph!

```
STARTDATE <-  MIN(FLIGHTS_DF$DATE)
ENDDATE <-  MAX(FLIGHTS_DF$DATE)

GGPLOT(DATA=AVG) +
  GEOM_BAR(AES(X = STATS::REORDER(AIRLINE, -DELAY), Y = DELAY, FILL = AIRLINE),
           STAT = "IDENTITY", WIDTH = 0.6) +
  LABS(TITLE = "AVERAGE DELAY PER AIRLINE", SUBTITLE = PASTE("FROM", STARTDATE, "T
O", ENDDATE),
       CAPTION = "BY MARKUS KÖFLER", X = "AIRLINES", Y = "AVERAGE DELAY (MIN)") +
  THEME(AXIS.TEXT.X = ELEMENT_BLANK()) +
  THEME(AXIS.TEXT.X = ELEMENT_TEXT(ANGLE = 45, HJUST = 1))
```



Or displaying the average delay of Airlines for each month - maybe we can get even better insights from the data?!

```
AG <- FLIGHTS_DF %>%
  GROUP_BY(AIRLINE, MONTH) %>%
  DROP_NA() %>%
  SUMMARIZE(DELAY = MEAN(TOTAL_DELAY))

## `SUMMARISE()` HAS GROUPED OUTPUT BY 'AIRLINE'. YOU CAN OVERRIDE USING THE
## `.GROUPS` ARGUMENT.

GGPLOT(DATA=AG) +
  GEOM_BAR(AES(X = REORDER(AIRLINE, -DELAY), Y = DELAY, FILL = AIRLINE),
           STAT = "IDENTITY", WIDTH = 0.6) +
  LABS(TITLE = "AVERAGE DELAY PER AIRLINE", SUBTITLE = PASTE("FROM", STARTDATE, "T
O", ENDDATE),
       CAPTION = "BY MARKUS KÖFLER", X = "AIRLINES", Y = "AVERAGE DELAY (MIN)") +
  THEME(AXIS.TEXT.X = ELEMENT_BLANK()) +
  THEME(AXIS.TEXT.X = ELEMENT_TEXT(ANGLE = 70, HJUST = 1)) +
  FACET_WRAP(~MONTH)
```



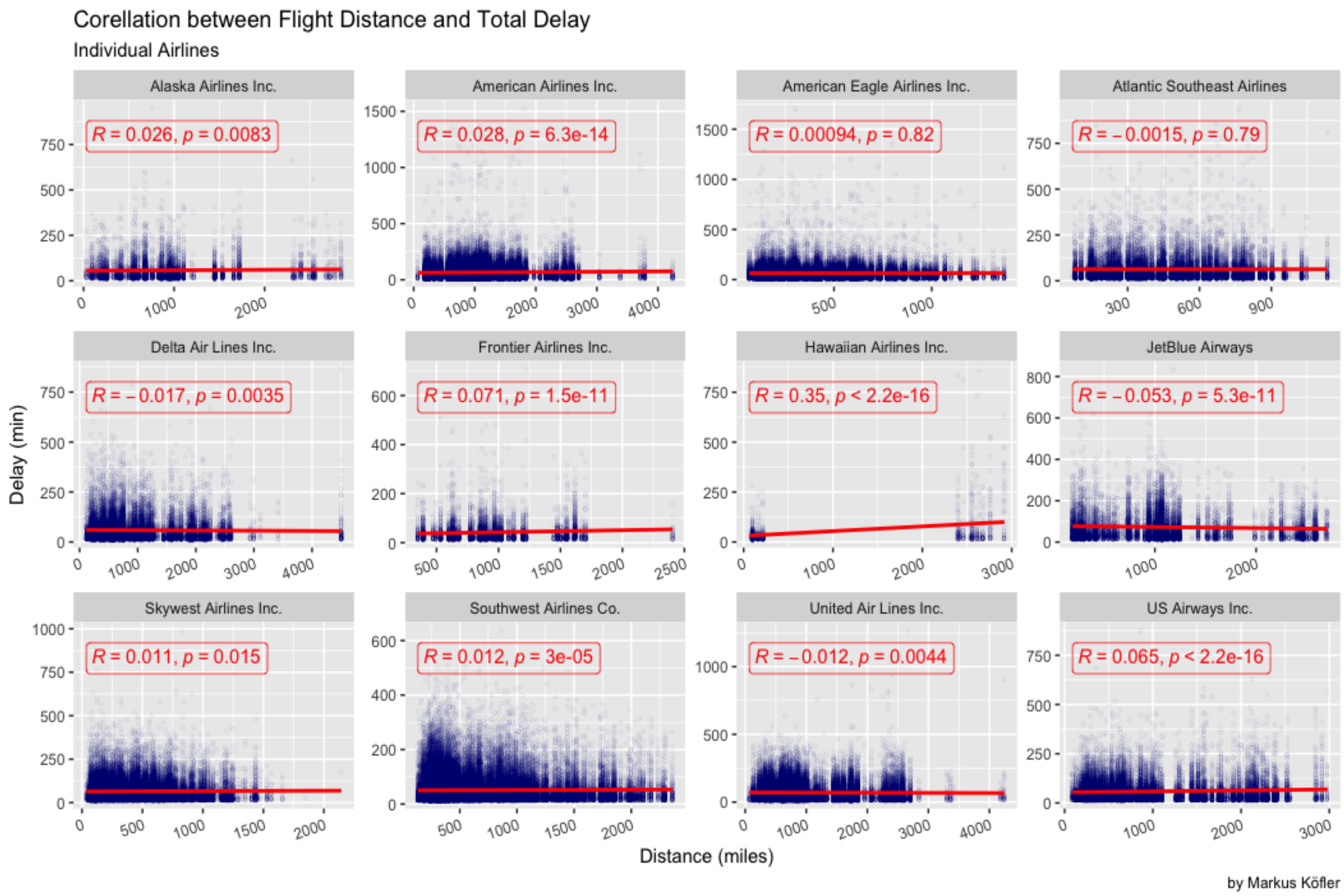
We can see that Alaska Airlines average delay for June is 0 min. Can Alaska Airlines really boast that none of their flights was delayed in June or are there just no recorded flights?

```
NROW(FILTER(FLIGHTS_DF, AIRLINE=="ALASKA AIRLINES INC." & MONTH==6))

## [1] 0
```

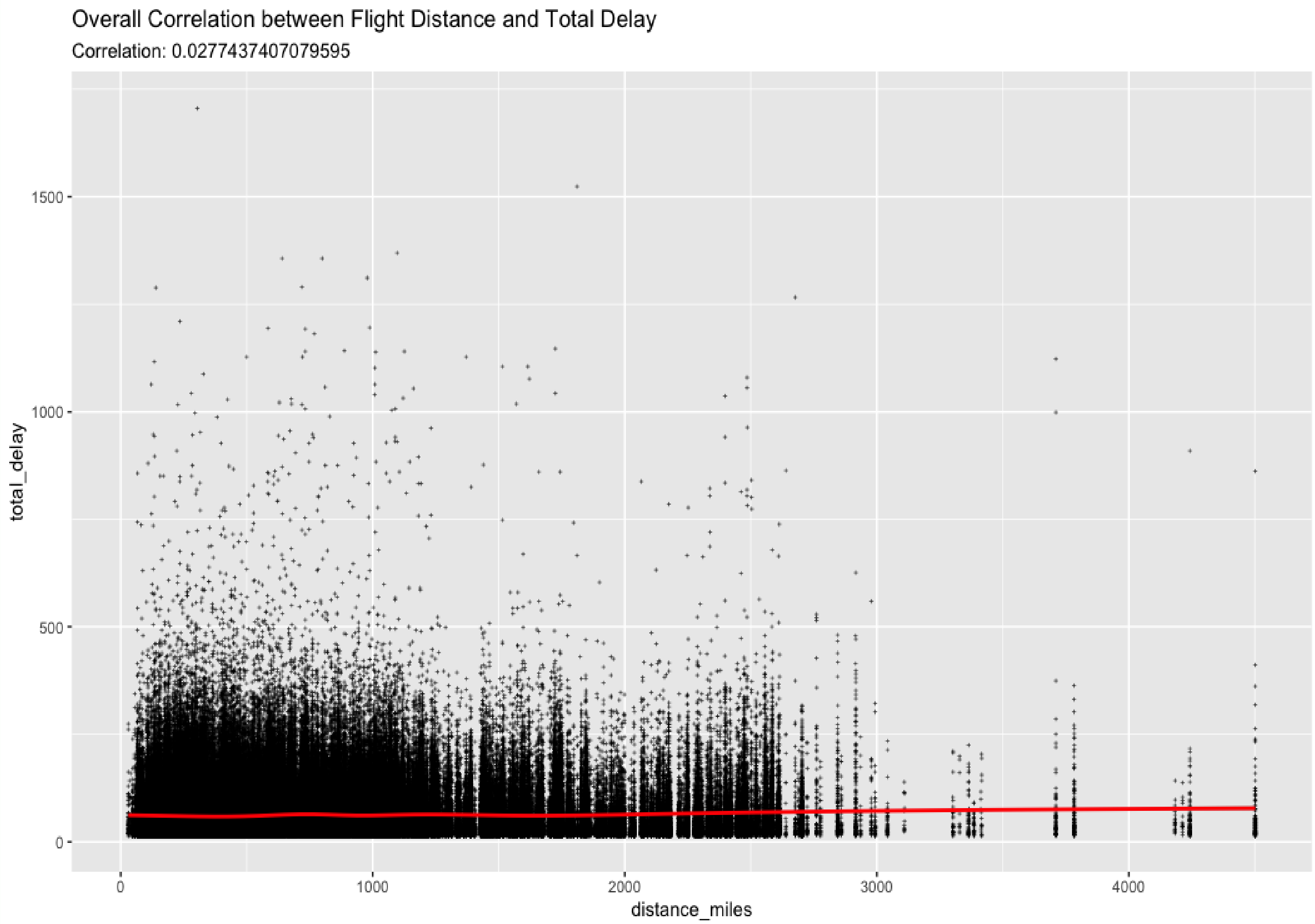
As the output suggests, the returned tibble contains 0 rows, meaning that there is no data on Alaska Airline flights in June. Further research needs to be done with regards to why this is the case.

11 THE RELATIONSHIP BETWEEN THE TOTAL DELAY AND THE FLIGHT DISTANCE - CAN PASSENGERS EXPECT A LONGER DELAY FOR LONGER TRAVELS? WE CAN ALSO ADD THE CORRELATION COEFFICIENTS WITH P-VALUES TO THE SCATTER PLOT

```
GGPLOT(FLIGHTS_DF) +
  GEOM_JITTER(AES(DISTANCE_MILES, TOTAL_DELAY), ALPHA = 0.1, SHAPE = "O", COLOR =
"NAVY") +
  GEOM_SMOOTH(AES(DISTANCE_MILES, TOTAL_DELAY), COLOR = "RED", METHOD = "LM") +
  FACET_WRAP(~AIRLINE, SCALE = "FREE", SHRINK = FALSE) + #ADJUSTED X- AND Y-AXIS
  STAT_COR(AES(DISTANCE_MILES, TOTAL_DELAY),
           COLOR = "RED", GEOM = "LABEL", FILL = "TRANSPARENT") +
  LABS(TITLE = "CORELLATION BETWEEN FLIGHT DISTANCE AND TOTAL DELAY",
       SUBTITLE = "INDIVIDUAL AIRLINES",
       CAPTION = "BY MARKUS KÖFLER", X = "DISTANCE (MILES)", Y = "DELAY (MIN)") +
  THEME(AXIS.TEXT.X = ELEMENT_TEXT(ANGLE = 20, HJUST = 1))

## `GEOM_SMOOTH()` USING FORMULA 'Y ~ X'
```




```
#RELATIONSHIP BETWEEN DELAY AND FLIGHT DURATION/ DISTANCE (DO LONGER TRIPS MEAN A
LONGER EXPECTED DELAY?)

GGPLOT(FLIGHTS_DF) +
  GEOM_JITTER(AES(DISTANCE_MILES, TOTAL_DELAY), SHAPE = "+", ALPHA = 0.9) +
  GEOM_SMOOTH(AES(DISTANCE_MILES, TOTAL_DELAY), COLOR = "RED") +
  LABS(TITLE = "OVERALL CORRELATION BETWEEN FLIGHT DISTANCE AND TOTAL DELAY",
       SUBTITLE = PASTE("CORRELATION:",
                        TOSTRING(COR(FLIGHTS_DF$DISTANCE_MILES, FLIGHTS_DF$TOTAL_D
ELAY)),
                        SEP = " "))

## `GEOM_SMOOTH()` USING METHOD = 'GAM' AND FORMULA 'Y ~ S(X, BS = "CS")'
```



---

12 NOW LETS FIND OUT WHAT ARE THE MOST POPULAR ARRIVAL AND DEPARTURE AIRPORTS

```
DEP_AIRPORT_DF <- DPLYR::RENAME(AS.DATA.FRAME(TABLE(FLIGHTS_DF$DEP_AIRPORT)) %>%
  ARRANGE(-FREQ), DEP_AIRPORT = VAR1, DEPARTURES = FREQ)

DEST_AIRPORT_DF <- DPLYR::RENAME(AS.DATA.FRAME(TABLE(FLIGHTS_DF$DEST_AIRPORT)) %>%

  ARRANGE(-FREQ), DEST_AIRPORT = VAR1, ARRIVALS = FREQ)

DEP_DEST_AIRPORTS <- CBIND(DEP_AIRPORT_DF, DEST_AIRPORT_DF)

HEAD(DEP_DEST_AIRPORTS, N = 10)

##                                          DEP_AIRPORT DEPARTURES
## 1              CHICAGO O'HARE INTERNATIONAL AIRPORT      46945
## 2            DALLAS/FORT WORTH INTERNATIONAL AIRPORT     33027
## 3  HARTSFIELD-JACKSON ATLANTA INTERNATIONAL AIRPORT      28834
## 4                     DENVER INTERNATIONAL AIRPORT      23543
## 5                LOS ANGELES INTERNATIONAL AIRPORT      17194
## 6                   MCCARRAN INTERNATIONAL AIRPORT      15529
## 7              SAN FRANCISCO INTERNATIONAL AIRPORT      14825
## 8            PHOENIX SKY HARBOR INTERNATIONAL AIRPORT    13873
## 9              CHICAGO MIDWAY INTERNATIONAL AIRPORT       9318
## 10                   ORLANDO INTERNATIONAL AIRPORT        9043
##                                         DEST_AIRPORT ARRIVALS
## 1              CHICAGO O'HARE INTERNATIONAL AIRPORT    40622
## 2            DALLAS/FORT WORTH INTERNATIONAL AIRPORT   24543
## 3  HARTSFIELD-JACKSON ATLANTA INTERNATIONAL AIRPORT    23557
```

```
## 4                      DENVER INTERNATIONAL AIRPORT    19250
## 5                 LOS ANGELES INTERNATIONAL AIRPORT    18350
## 6               SAN FRANCISCO INTERNATIONAL AIRPORT    15721
## 7                    MCCARRAN INTERNATIONAL AIRPORT    14930
## 8          PHOENIX SKY HARBOR INTERNATIONAL AIRPORT    12517
## 9           LAGUARDIA AIRPORT (MARINE AIR TERMINAL)    10692
## 10             SALT LAKE CITY INTERNATIONAL AIRPORT     9104
```

The created data frame tells us, what are the airports with the most (domestic) traffic. A tendency, that airports with the most departures also rank high when it comes to arrivals, is given. Let's investigate the correlation between the departure rank and the arrival rank:

```
LEN_OF_DF <- LENGTH(DEP_DEST_AIRPORTS$DEP_AIRPORT)

# ASSIGNING INTEGERS FROM 1 TO 260
RANK <- C(1:LEN_OF_DF)

# ADDING RANKING TO EACH INDIVIDUAL DATA FRAME
DEP_RANK_DF <- MUTATE(DPLYR::RENAME(DEP_AIRPORT_DF, AIRPORT = DEP_AIRPORT), RANK_D
EP = RANK)
DEST_RANK_DF <- MUTATE(DPLYR::RENAME(DEST_AIRPORT_DF, AIRPORT = DEST_AIRPORT), RAN
K_DEST = RANK)

#LIBRARY(PLYR)
# JOINING THE DATA FRAMES BASED ON A COMMON KEY WHICH IS THE COLUMN "AIRPORT"
DEP_DEST_RANK <- ARRANGE(PLYR::JOIN(DEP_RANK_DF,
                                    DEST_RANK_DF, TYPE = "FULL",
                                    BY = "AIRPORT"),
                         + RANK_DEP)

TOP_N(DEP_DEST_RANK, -10)

## SELECTING BY RANK_DEST

##                                          AIRPORT DEPARTURES RANK_DEP
## 1               CHICAGO O'HARE INTERNATIONAL AIRPORT      46945        1
## 2            DALLAS/FORT WORTH INTERNATIONAL AIRPORT      33027        2
## 3  HARTSFIELD-JACKSON ATLANTA INTERNATIONAL AIRPORT      28834        3
## 4                       DENVER INTERNATIONAL AIRPORT      23543        4
## 5                  LOS ANGELES INTERNATIONAL AIRPORT      17194        5
## 6                     MCCARRAN INTERNATIONAL AIRPORT      15529        6
## 7                SAN FRANCISCO INTERNATIONAL AIRPORT      14825        7
## 8           PHOENIX SKY HARBOR INTERNATIONAL AIRPORT      13873        8
## 9               SALT LAKE CITY INTERNATIONAL AIRPORT       8860       11
## 10           LAGUARDIA AIRPORT (MARINE AIR TERMINAL)       8719       12
##    ARRIVALS RANK_DEST
## 1     40622         1
## 2     24543         2
## 3     23557         3
## 4     19250         4
## 5     18350         5
## 6     14930         7
## 7     15721         6
## 8     12517         8
## 9      9104        10
## 10    10692         9
```

Now that we have the ranking for departures and arrivals, we can compute the correlation. I used the 3 common CORRELATION METHODS:

- Pearson => linear relationship between two variables
- Kendall => monotonic relationship (likelihood of two variables to move in one direction, but not necessarily in a constant manner)
- Spearman => monotonic relationship (similar to Kendall method, but not as popular)

```
# COMPUTING THE CORRELATION
# FUNCTION WHICH ITERATES THROUGH A VECTOR CONTAINING
# THE 3 CORRELATION METHODS USED IN DATA SCIENCE
COR_METHODS <- C("PEARSON", "KENDALL", "SPEARMAN")

FOR (COR_METHOD IN COR_METHODS) {
     PRINT(PASTE(COR_METHOD, SEP = ": ",
                COR(DEP_DEST_RANK$RANK_DEP, DEP_DEST_RANK$RANK_DEST, METHOD = COR_
METHOD)
                )
           )
}

## [1] "PEARSON: 0.99265760645071"
## [1] "KENDALL: 0.933887733887734"
## [1] "SPEARMAN: 0.99265760645071"
```

Here is a much more sophisticated syntax. I did this to make my code more reproducible. Next time I want to compute the statistical correlation with all 3 methods, I simply call the function and pass in the arguments for the parameters var1 and var2.

```
COR_CALCULATOR <- FUNCTION (METHOD_VECTOR = C("PEARSON", "KENDALL", "SPEARMAN")
                            , VAR1, VAR2) {
  RESULT <- C()
  FOR (COR_METHOD IN METHOD_VECTOR) {
    RESULT <- APPEND(RESULT, PASTE(COR_METHOD, SEP = ": ",
                COR(DEP_DEST_RANK$RANK_DEP, DEP_DEST_RANK$RANK_DEST, METHOD = COR_
METHOD)))
    }
  RETURN(RESULT)
}


VARIABLE_1 <- DEP_DEST_RANK$RANK_DEP
VARIABLE_2 <- DEP_DEST_RANK$RANK_DEST

COR_CALCULATOR(VAR1 = VARIABLE_1, VAR2 = VARIABLE_2)

## [1] "PEARSON: 0.99265760645071"  "KENDALL: 0.933887733887734"
## [3] "SPEARMAN: 0.99265760645071"
```

---

13 WHAT ARE THE MOST FREQUENT ROUTES FLOWN IN THE US FROM JANUARY TO JUNE 2019?

To answer this question, I combined the columns dep_airport and dest_airport to build a column which contains both departure airport as well as destination airport. This allows us to get unique flight routes.

```
FLIGHTS_DF["DEP_DEST_AIRPORTS"] <- PASTE("FROM:", FLIGHTS_DF$DEP_AIRPORT,
                                         "TO:", FLIGHTS_DF$DEST_AIRPORT,
                                         SEP = " ")

FLIGHTS_DF$DEP_DEST_AIRPORTS[1:5]
```

```
## [1] "FROM: INDIANAPOLIS INTERNATIONAL AIRPORT TO: BALTIMORE-WASHINGTON INTERNAT
IONAL AIRPORT"
## [2] "FROM: INDIANAPOLIS INTERNATIONAL AIRPORT TO: MCCARRAN INTERNATIONAL AIRPOR
T"
## [3] "FROM: INDIANAPOLIS INTERNATIONAL AIRPORT TO: ORLANDO INTERNATIONAL AIRPORT
"
## [4] "FROM: INDIANAPOLIS INTERNATIONAL AIRPORT TO: PHOENIX SKY HARBOR INTERNATIO
NAL AIRPORT"
## [5] "FROM: INDIANAPOLIS INTERNATIONAL AIRPORT TO: TAMPA INTERNATIONAL AIRPORT"
```

The next step is counting what unique flight route occurs the most in the newly created column. Finally, we can arrange the data frame in descending order.

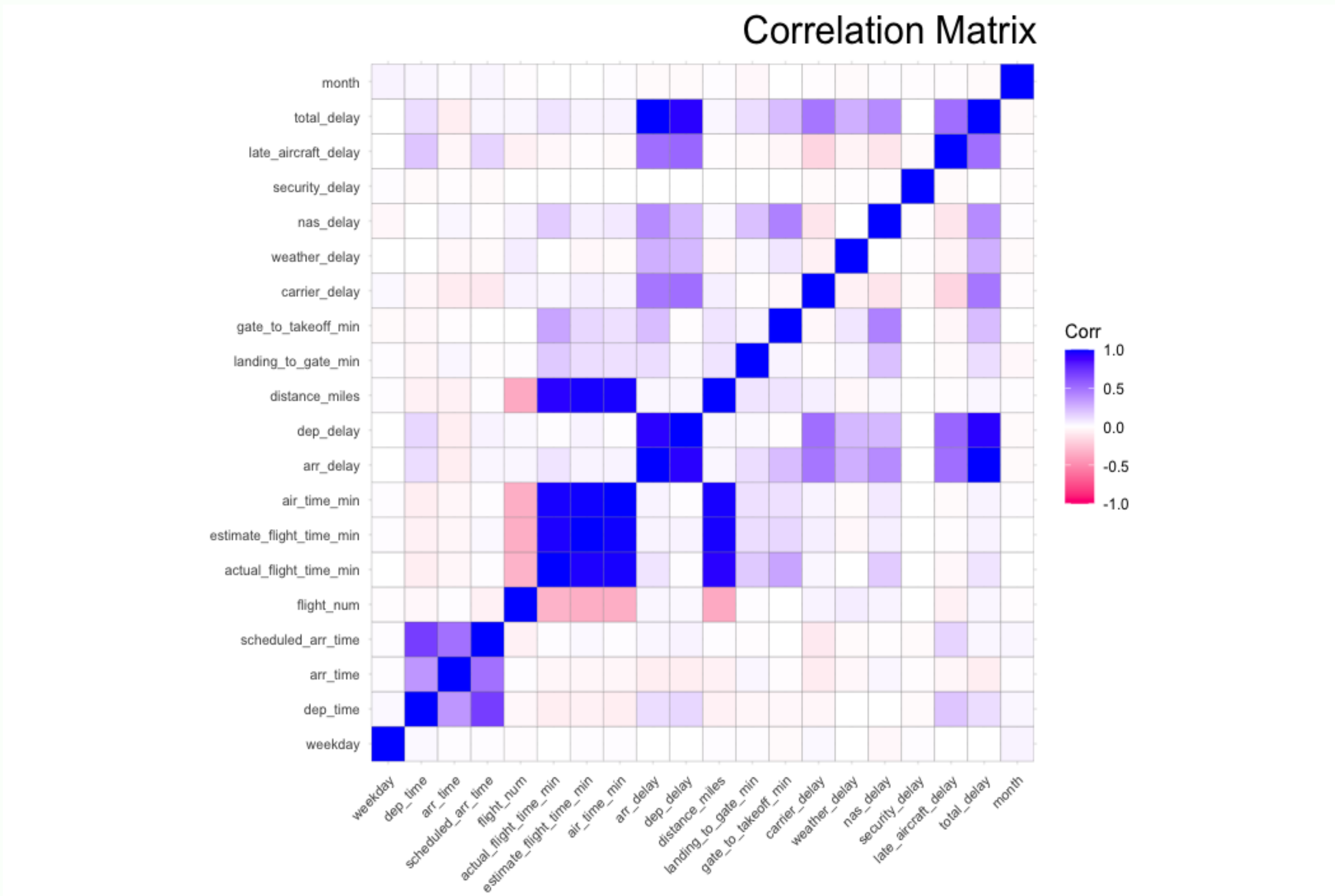
```
ROUTES_DF <- AS.DATA.FRAME(TABLE(FLIGHTS_DF["DEP_DEST_AIRPORTS"])) %>% ARRANGE(-FR
EQ)

# DISPLAY THE TOP 10 MOSTH FREQUENT TRAVEL ROUTES
TOP_N(ROUTES_DF, 10)
```

```
## SELECTING BY FREQ
```

```
##                                                                          DEP_DES
T_AIRPORTS
## 1  FROM: CHICAGO O'HARE INTERNATIONAL AIRPORT TO: LAGUARDIA AIRPORT (MARINE AIR
 TERMINAL)
## 2  FROM: LAGUARDIA AIRPORT (MARINE AIR TERMINAL) TO: CHICAGO O'HARE INTERNATION
AL AIRPORT
## 3        FROM: LOS ANGELES INTERNATIONAL AIRPORT TO: SAN FRANCISCO INTERNATION
AL AIRPORT
## 4        FROM: SAN FRANCISCO INTERNATIONAL AIRPORT TO: LOS ANGELES INTERNATION
AL AIRPORT
## 5             FROM: MCCARRAN INTERNATIONAL AIRPORT TO: LOS ANGELES INTERNATION
AL AIRPORT
## 6                                   FROM: WILLIAM P. HOBBY AIRPORT TO: DALLAS
LOVE FIELD
## 7                                   FROM: DALLAS LOVE FIELD TO: WILLIAM P. HOB
BY AIRPORT
## 8        FROM: CHICAGO O'HARE INTERNATIONAL AIRPORT TO: LOS ANGELES INTERNATION
AL AIRPORT
## 9      FROM: PHOENIX SKY HARBOR INTERNATIONAL AIRPORT TO: MCCARRAN INTERNATION
AL AIRPORT
## 10 FROM: DALLAS/FORT WORTH INTERNATIONAL AIRPORT TO: CHICAGO O'HARE INTERNATION
AL AIRPORT
##    FREQ
## 1  1920
## 2  1615
## 3  1603
## 4  1457
## 5  1305
## 6  1276
## 7  1200
## 8  1154
## 9  1152
## 10 1125
```

14 AT THE END OF THE GENERAL ANALYSIS I ALWAYS LIKE TO ADD ACORRELATION MATRIX. THE INTENTION IS TO HIGHLIGHT POSSIBLE RELATIONSHIPS AND TRENDS BETWEEN VARIABLES THAT HAVE NOT BEEN DISCOVERED YET.

```
#FILTERING FOR COLUMNS THAT ARE NUMERIC ONLY

FLIGHTS_NUMERIC <- SELECT_IF(FLIGHTS_DF, IS.NUMERIC)

# COMPUTING CORRELATION MATRIX
COR_MATRIX <- ROUND(COR(FLIGHTS_NUMERIC),3)

# VISUALIZING AND REORDERING CORRELATION MATRIX
GGCORRPLOT(COR_MATRIX, HC.ORDER =FALSE, TL.CEX = 8,
         OUTLINE.COLOR ="#808080", METHOD = "SQUARE", COLORS = C("#FF007F", "WHI
TE", "#0000FF")) +
  LABS(TITLE= "CORRELATION MATRIX") +
  THEME(PLOT.TITLE = ELEMENT_TEXT(SIZE = 22, HJUST = 1))
```



Based on den matrix, there is nothing outstanding to report.

Strongly positively related are:

- flight distance (distance_miles) with the air time (air_time_min), the estimated flight time (estimate_flight_time_min) and the actual flight time (actual_flight_time_min)
- departure delay with the arrival delay
- the total delay (total_delay) with the departure delay (dep_delay) and the arrival delay (arr_delay)

Optionally, we can compute the correlation matrix in numbers with p-values with the following code:

```
CORRP.MAT <- COR_PMAT(FLIGHTS_NUMERIC)
CORRP.MAT
```

THE BUSINESS TASK

1) A business consultancy company is sending their consultants to their customers within the US area (domestic flights).

2) The consultancy company is located in Chicago (IL)

3) Senior consultant Andrew needs to fly to a client located in Los Angeles. He passes his appointment to the HR team, which takes over responsibility for managing client meetings and travels for employees. HR manager Thomas asks for an analysis, what would be the best option to go from Chicago to Dallas.

We start preparing the data frame first - we create a column with the flight routes. This time, we only use Airport codes which consist of 3 uppercase letters to make the the script more readible:

```
FLIGHTS_DF <- MUTATE(FLIGHTS_DF,
                     ROUTE = PASTE(FLIGHTS_DF$DEP_AIRPORT_CODE,
                                   FLIGHTS_DF$DEST_AIRPORT_CODE,
                                   SEP = "-"))

FLIGHTS_DF$ROUTE[1:5]

## [1] "IND-BWI" "IND-LAS" "IND-MCO" "IND-PHX" "IND-TPA"
```

For finding the routes with the shortest average delay that can be expected (based on the data), I used SQL statements by using the library **sqldf**. It allows us to query the data frame in SQL-syntax style by passing in the SQL statement as a string.

SQL QUERY

```
SQLDF("
       SELECT
          ROUTE,
          AIRLINE,
          AVG(ACTUAL_FLIGHT_TIME_MIN) AS AVERAGE_TRAVEL_TIME,
          AVG(TOTAL_DELAY) AS AVERAGE_DELAY

       FROM
          FLIGHTS_DF

      WHERE
        ROUTE = 'ORD-LAX' OR ROUTE = 'MDW-LAX'

      GROUP BY
          AIRLINE

      ORDER BY
          AVERAGE_DELAY ASC
      ")

##      ROUTE                 AIRLINE AVERAGE_TRAVEL_TIME AVERAGE_DELAY
## 1 MDW-LAX SOUTHWEST AIRLINES CO.            271.8029      49.75627
## 2 ORD-LAX  UNITED AIR LINES INC.            273.1996      66.11586
## 3 ORD-LAX AMERICAN AIRLINES INC.            271.8010      69.16695
```

According to the results, the best option would be to book a flight from Chicago Midway (MDW) to LA International (LAX) in terms of expected reliability. The differences in average travel time is too insignificant and can be neglected.

Next, a consultant, who has been negotiating with a client in Dallas (TX) needs to directly visit a nother customer in New York. There are three target airports in NY to choose from at the time. There is also the option to either leave from Dallas Fort-Worth or Dallas Love Fields. What is the best constellation of airports to choose from?

```
SQLDF("
      SELECT
         AIRLINE,
         ROUTE,
         AVG(ACTUAL_FLIGHT_TIME_MIN) AS AVERAGE_TRAVEL_TIME,
         AVG(TOTAL_DELAY) AS AVERAGE_DELAY

      FROM
        FLIGHTS_DF

     WHERE
         ROUTE = 'DFW-JFK' OR
         ROUTE = 'DFW-LGA' OR
         ROUTE = 'DFW-EWR' OR
         ROUTE = 'DAL-JFK' OR
         ROUTE = 'DAL-LGA' OR
         ROUTE = 'DAL-EWR'

     GROUP BY
         ROUTE

     ORDER BY
         AVERAGE_TRAVEL_TIME ASC
       ")
```

```
##                  AIRLINE   ROUTE AVERAGE_TRAVEL_TIME AVERAGE_DELAY
## 1 AMERICAN AIRLINES INC. DFW-EWR            214.3131      70.10942
## 2 AMERICAN AIRLINES INC. DFW-LGA            214.4140      66.26858
## 3 AMERICAN AIRLINES INC. DFW-JFK            229.4828      62.87931
```

The results suggest that DFW has better connection to one of the popular NYC airports (since there are no other flights recorded from Dallas Love Fields). We assume that DFW has better flight schedules to NYC. When it comes to choosing an airport in NYC, we have to make a trade-off whether to accept a slightly higher average travel delay to have an overall shorter expected travel time.

Just to be certain - we check if there are really no flights from DAL to any NYC airport in our data set.

```
SUM((FLIGHTS_DF$DEP_AIRPORT_CODE == "DAL" & FLIGHTS_DF$DEST_AIRPORT_CODE == "JFK")
 |
    (FLIGHTS_DF$DEP_AIRPORT_CODE == "DAL" & FLIGHTS_DF$DEST_AIRPORT_CODE == "LGA")
 |
    (FLIGHTS_DF$DEP_AIRPORT_CODE == "DAL" & FLIGHTS_DF$DEST_AIRPORT_CODE == "EWR")
)
```

```
## [1] 0
```

Indeed, we cannot find any flights from Dallas Love Fields to a NYC airport.