STA 518 Data Transformations & Manipulation with dplyr Activity

Anto

2023-10-02

Let’s load some packages to start

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(knitr)  
library(skimr)  
library(scales)

##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

library(dplyr)  
library(lubridate)  
library(ggplot2)

Then, let’s import the flights data into R.

# Importing flights data  
miFlights <- read\_csv("C:/Users/antol/Downloads/miFlights2019-2021/miFlights2019-2021.csv")

## Rows: 463818 Columns: 37  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (9): carrier, tailnum, origin, dest, carrier\_name, plane\_type, plane\_m...  
## dbl (27): year, month, day, dep\_time, sched\_dep\_time, dep\_delay, arr\_time, ...  
## dttm (1): time\_hour  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Let’s use glimpse() and skim() to get familiar with the data.

glimpse(miFlights)

## Rows: 463,818  
## Columns: 37  
## $ year <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 201…  
## $ month <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ day <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ dep\_time <dbl> 55, 455, 506, 531, 534, 550, 555, 555, 555, 600, 60…  
## $ sched\_dep\_time <dbl> 2115, 500, 511, 535, 545, 600, 600, 555, 600, 600, …  
## $ dep\_delay <dbl> 220, -5, -5, -4, -11, -10, -5, 0, -5, 0, -2, 3, 4, …  
## $ arr\_time <dbl> 426, 830, 710, 647, 750, 712, 822, 709, 755, 559, 9…  
## $ sched\_arr\_time <dbl> 2323, 834, 730, 710, 742, 748, 834, 715, 817, 615, …  
## $ arr\_delay <dbl> 303, -4, -20, -23, 8, -36, -12, -6, -22, -16, 3, -7…  
## $ carrier <chr> "OH", "YX", "AA", "WN", "B6", "YX", "OO", "WN", "DL…  
## $ flight <dbl> 1019, 954, 185, 203, 310, 790, 803, 295, 348, 218, …  
## $ tailnum <chr> "N567NN", "N433YX", "N853NN", "N227WN", "N203JB", "…  
## $ origin <chr> "DTW", "GRR", "DTW", "DTW", "DTW", "DTW", "FNT", "D…  
## $ dest <chr> "CLT", "MIA", "DFW", "BWI", "BOS", "EWR", "ATL", "D…  
## $ air\_time <dbl> 88, 187, 162, 64, 71, 61, 125, 174, 105, 45, 179, 1…  
## $ distance <dbl> 500, 1214, 986, 409, 632, 488, 645, 1123, 640, 228,…  
## $ hour <dbl> 21, 5, 5, 5, 5, 6, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6,…  
## $ minute <dbl> 15, 0, 11, 35, 45, 0, 0, 55, 0, 0, 3, 0, 0, 5, 0, 1…  
## $ time\_hour <dttm> 2019-01-01 21:00:00, 2019-01-01 05:00:00, 2019-01-…  
## $ temp <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
## $ dewp <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
## $ humid <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
## $ wind\_dir <dbl> 10, 320, 290, 290, 290, 290, 330, 290, 290, 290, 29…  
## $ wind\_speed <dbl> 8.05546, 9.20624, 19.56326, 19.56326, 19.56326, 13.…  
## $ wind\_gust <dbl> 9.270062, 10.594357, 22.513008, 22.513008, 22.51300…  
## $ precip <dbl> NA, 1e-04, NA, NA, NA, NA, 1e-04, NA, NA, NA, NA, N…  
## $ pressure <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…  
## $ visib <dbl> 1.25, 9.00, 7.00, 7.00, 7.00, 4.00, 9.00, 7.00, 10.…  
## $ carrier\_name <chr> "PSA Airlines Inc.", "Republic Airline", "American …  
## $ plane\_year <dbl> 2015, 2014, 2010, 2005, 2006, 2016, 2006, 1999, 200…  
## $ plane\_type <chr> "Fixed wing multi engine", "Fixed wing multi engine…  
## $ plane\_manufacturer <chr> "BOMBARDIER INC", "EMBRAER S A", "BOEING", "BOEING"…  
## $ plane\_model <chr> "CL-600-2D24", "ERJ 170-200 LR", "737-823", "737-7H…  
## $ plane\_engines <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ plane\_seats <dbl> 95, 88, 162, 140, 20, 88, 95, 149, 100, 140, 162, 1…  
## $ plane\_speed <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ plane\_engine <chr> "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turbo-fan",…

skim(miFlights)

Data summary

|  |  |
| --- | --- |
| Name | miFlights |
| Number of rows | 463818 |
| Number of columns | 37 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 9 |
| numeric | 27 |
| POSIXct | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| carrier | 0 | 1.00 | 2 | 2 | 0 | 16 | 0 |
| tailnum | 2189 | 1.00 | 3 | 6 | 0 | 5250 | 0 |
| origin | 0 | 1.00 | 3 | 3 | 0 | 4 | 0 |
| dest | 0 | 1.00 | 3 | 3 | 0 | 130 | 0 |
| carrier\_name | 0 | 1.00 | 9 | 34 | 0 | 16 | 0 |
| plane\_type | 11140 | 0.98 | 23 | 23 | 0 | 1 | 0 |
| plane\_manufacturer | 11140 | 0.98 | 6 | 29 | 0 | 16 | 0 |
| plane\_model | 11140 | 0.98 | 5 | 15 | 0 | 93 | 0 |
| plane\_engine | 11140 | 0.98 | 9 | 9 | 0 | 2 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| year | 0 | 1.00 | 2019.92 | 0.85 | 2019.00 | 2019.00 | 2020.00 | 2021.00 | 2021.00 | ▇▁▆▁▆ |
| month | 0 | 1.00 | 6.53 | 3.48 | 1.00 | 3.00 | 7.00 | 10.00 | 12.00 | ▇▅▅▅▇ |
| day | 0 | 1.00 | 15.74 | 8.76 | 1.00 | 8.00 | 16.00 | 23.00 | 31.00 | ▇▇▇▇▆ |
| dep\_time | 9060 | 0.98 | 1372.26 | 490.52 | 1.00 | 950.00 | 1355.00 | 1754.00 | 2400.00 | ▁▇▇▇▆ |
| sched\_dep\_time | 0 | 1.00 | 1368.40 | 481.75 | 49.00 | 948.00 | 1355.00 | 1750.00 | 2336.00 | ▁▇▇▇▆ |
| dep\_delay | 9063 | 0.98 | 7.11 | 44.97 | -54.00 | -5.00 | -3.00 | 0.00 | 2672.00 | ▇▁▁▁▁ |
| arr\_time | 9324 | 0.98 | 1481.01 | 506.74 | 1.00 | 1053.00 | 1502.00 | 1905.00 | 2400.00 | ▁▅▇▇▆ |
| sched\_arr\_time | 0 | 1.00 | 1496.79 | 495.16 | 1.00 | 1103.00 | 1510.00 | 1910.00 | 2359.00 | ▁▃▇▇▇ |
| arr\_delay | 10239 | 0.98 | 0.16 | 47.21 | -85.00 | -17.00 | -9.00 | 2.00 | 2649.00 | ▇▁▁▁▁ |
| flight | 0 | 1.00 | 413.37 | 269.57 | 1.00 | 189.00 | 387.00 | 600.00 | 1322.00 | ▇▇▆▂▁ |
| air\_time | 10239 | 0.98 | 94.59 | 63.12 | 15.00 | 50.00 | 74.00 | 130.00 | 581.00 | ▇▂▁▁▁ |
| distance | 0 | 1.00 | 641.00 | 488.23 | 74.00 | 296.00 | 500.00 | 957.00 | 4475.00 | ▇▂▁▁▁ |
| hour | 0 | 1.00 | 13.41 | 4.79 | 0.00 | 9.00 | 13.00 | 17.00 | 23.00 | ▁▇▇▇▆ |
| minute | 0 | 1.00 | 27.49 | 17.94 | 0.00 | 11.00 | 27.00 | 44.00 | 59.00 | ▇▆▇▆▆ |
| temp | 441760 | 0.05 | 42.16 | 15.40 | -4.00 | 32.00 | 37.90 | 48.90 | 90.00 | ▁▆▇▂▁ |
| dewp | 441762 | 0.05 | 31.91 | 13.42 | -9.00 | 23.00 | 28.90 | 39.90 | 75.90 | ▁▆▇▃▁ |
| humid | 441773 | 0.05 | 68.87 | 15.10 | 25.87 | 57.93 | 71.82 | 80.66 | 100.00 | ▁▃▅▇▃ |
| wind\_dir | 9205 | 0.98 | 181.02 | 109.46 | 0.00 | 80.00 | 200.00 | 270.00 | 360.00 | ▇▃▆▇▆ |
| wind\_speed | 4367 | 0.99 | 8.59 | 5.64 | 0.00 | 4.60 | 8.06 | 11.51 | 42.58 | ▇▆▁▁▁ |
| wind\_gust | 4367 | 0.99 | 9.88 | 6.50 | 0.00 | 5.30 | 9.27 | 13.24 | 49.00 | ▇▆▁▁▁ |
| precip | 430846 | 0.07 | 0.01 | 0.02 | 0.00 | 0.00 | 0.00 | 0.01 | 0.44 | ▇▁▁▁▁ |
| pressure | 447131 | 0.04 | 1018.83 | 7.60 | 990.40 | 1014.10 | 1019.00 | 1023.40 | 1038.50 | ▁▂▇▇▂ |
| visib | 1934 | 1.00 | 8.18 | 2.56 | 0.06 | 7.00 | 10.00 | 10.00 | 10.00 | ▁▁▁▂▇ |
| plane\_year | 21647 | 0.95 | 2008.12 | 7.15 | 1987.00 | 2003.00 | 2007.00 | 2015.00 | 2021.00 | ▁▂▇▃▅ |
| plane\_engines | 11140 | 0.98 | 2.00 | 0.02 | 2.00 | 2.00 | 2.00 | 2.00 | 3.00 | ▇▁▁▁▁ |
| plane\_seats | 11140 | 0.98 | 127.86 | 66.68 | 20.00 | 80.00 | 95.00 | 182.00 | 451.00 | ▇▃▂▁▁ |
| plane\_speed | 11140 | 0.98 | 0.01 | 1.72 | 0.00 | 0.00 | 0.00 | 0.00 | 438.00 | ▇▁▁▁▁ |

**Variable type: POSIXct**

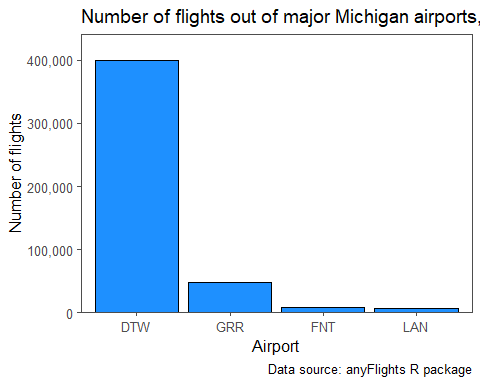
| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| time\_hour | 0 | 1 | 2019-01-01 05:00:00 | 2021-12-31 22:00:00 | 2020-03-26 06:00:00 | 19059 |

## Bar chart

*Create a bar chart showing how many flights departed out of each airport (origin) using the count() and geom\_col() functions. Also sort the bars by descending height using the fct\_reorder() function.*

Let’s make a bar chart.

# Making table of counts  
originCounts <- miFlights %>%  
 dplyr::count(origin)  
  
# Making a bar chart  
originCounts %>%  
 ggplot(aes(x = fct\_reorder(origin, n, .desc = TRUE), y = n)) +  
 geom\_col(fill = "dodgerblue",  
 color = "black") +  
 scale\_y\_continuous(labels = scales::label\_comma(),  
 expand = expansion(mult = c(0, 0.1))) +  
 labs(title = "Number of flights out of major Michigan airports, 2019-2021",  
 x = "Airport",  
 y = "Number of flights" ,  
 caption = "Data source: anyFlights R package") +  
 ggthemes::theme\_few()



## The filter() function

*Selecting all flights on January 1st in the data set, create a new object called janFlights.*

# Subsetting to January first flights  
janFlights <- miFlights %>%  
 dplyr::filter(month == 1, day == 1)

*Suppose we want to create a data set called dec25 that contains flight data from December 25th. What code would we need using the filter() function to create dec25?*

# Subsetting to January first flights  
dec25 <- miFlights %>%  
 dplyr::filter(month == 12, day == 25)

*Find all flights that departed in November or December, creating an object called novDec.*

# Subsetting to flights in November or December  
novDec <- miFlights %>%  
 dplyr::filter(month == 11 | month == 12)

*Find all flights that departed in November or December using the %in% operator, creating an object called novDec.*

# Subsetting to flights in November or December  
novDec <- miFlights %>%  
 dplyr::filter(month %in% c(11:12))

*Select all flights except those in the months of November and December using !.*

# Subsetting to flights in months except in November or December  
notNovDec <- miFlights %>%  
 dplyr::filter(!(month %in% 11:12))

*Create a new object called miFlightsComplete where all departure times are non-missing, and miFlightsMiss where all departure times are missing*

# Only keeping rows with missing values  
miFlightsMiss <- miFlights %>%  
 dplyr::filter(is.na(dep\_time))  
  
# Excluding missing values  
miFlightsComplete <- miFlights %>%  
 dplyr::filter(!is.na(dep\_time))

## The arrange() function

*Sort miFlights by the day of the flight (smallest to largest), and print the first 4 columns and 5 rows of the resulting data set using the slice\_head() function.*

# Printing rows of sorted data  
miFlights %>%  
 arrange(day) %>%  
 dplyr::select(1:4) %>%  
 slice\_head(n = 5)

## # A tibble: 5 × 4  
## year month day dep\_time  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 1 1 55  
## 2 2019 1 1 455  
## 3 2019 1 1 506  
## 4 2019 1 1 531  
## 5 2019 1 1 534

*Sort miFlights by the day of the flight (largest to smallest), and print the first 4 columns and 5 rows of the resulting data set using the slice\_head() function.*

# Printing rows of sorted data  
miFlights %>%  
 arrange(desc(day)) %>%  
 dplyr::select(1:4) %>%  
 slice\_head(n = 5)

## # A tibble: 5 × 4  
## year month day dep\_time  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 1 31 59  
## 2 2019 1 31 535  
## 3 2019 1 31 540  
## 4 2019 1 31 548  
## 5 2019 1 31 549

*Sort miFlights by the year, month, and day of the flight.*

# Printing rows of sorted data  
miFlights <- miFlights %>%  
 arrange(year, month, day)

1. Sort miFlights to find the 3 most delayed flights (arr\_delay), and the 3 that left the earliest relative to their scheduled departure (dep\_delay).

# To find the 3 most delayed flights (arr\_delay)  
most\_delayed <- miFlights %>%  
 arrange(desc(arr\_delay)) %>%  
 slice\_head(n = 3)  
  
# To find the 3 flights that left the earliest (dep\_delay)  
earliest\_departures <- miFlights %>%  
 arrange(dep\_delay) %>%  
 slice\_head(n = 3)

1. Sort miFlights to find the 3 fastest (highest speed) flights.

# Calculate speed for each flight (in miles per hour)  
miFlights <- miFlights %>%  
 mutate(speed = distance / (air\_time / 60)) # air\_time is in minutes, so dividing by 60 converts it to hours  
  
# Sort the data frame by speed to find the 3 fastest flights  
fastest\_flights <- miFlights %>%  
 arrange(desc(speed)) %>%  
 slice\_head(n = 3)

1. For flights coming out of GRR, find the 3 flights that traveled the farthest (distance) and that arrived the earliest in the morning (arr\_time) simultaneously.

# Filter flights coming out of GRR  
flights\_from\_GRR <- miFlights %>%  
 filter(origin == "GRR")  
  
# Find the 3 flights that traveled the farthest (distance) and arrived the earliest in the morning (arr\_time)  
farthest\_and\_earliest <- flights\_from\_GRR %>%  
 arrange(desc(distance), arr\_time) %>%  
 slice\_head(n = 3)

## The select() function

*Drop the year and month columns from miFlights creating a new data set called miDropped.*

# Dropping year and month columns  
miDropped <- miFlights %>%  
 dplyr::select(-year, -month)  
  
miDropped <- miFlights %>%  
 dplyr::select(-c(year:month))

selectExample <- dplyr::select(miFlights, time\_hour, air\_time, everything(), -day)  
  
selectExample <- miFlights %>%  
 dplyr::select(time\_hour, air\_time, everything(), -day)

1. Create a subset of the miFlights data set called timeFlights that only contains variables that end with the word “time”.

# Create a subset of miFlights with variables ending with "time"  
timeFlights <- miFlights %>%  
 select(ends\_with("time"))

1. Create a new data frame called departureInfo that only has variables that start with “dep”

# Create a subset of miFlights with variables starting with "dep"  
departureInfo <- miFlights %>%  
 select(starts\_with("dep"))

1. Create a new data frame call newFlights by rearranging the columns of the full miFlights data set so that flight number (flight), origin (origin), and destination (dest) are provided first, then all other columns except the tail number (tailnum).

# Rearranging columns in miFlights  
newFlights <- miFlights %>%  
 dplyr::select(flight, origin, dest, everything(), -tailnum)

## The mutate() function

flights\_sml <- miFlights %>% dplyr::select(ends\_with("delay"), distance, air\_time)  
  
flights\_sml <- flights\_sml %>% mutate(gain = dep\_delay - arr\_delay)

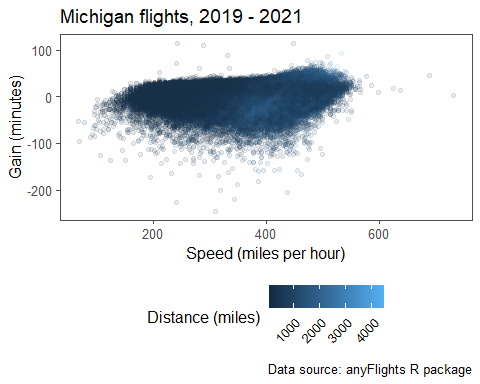
*Extending the code provided with a single call to mutate(), create a new variable, speed, that is equal to distance divided by air\_time, producing a new data set called flightSpeeds.*

flights\_sml <- miFlights %>%   
dplyr::select(ends\_with("delay"), distance, air\_time)  
  
flights\_sml <- flights\_sml %>%   
 mutate(gain = dep\_delay - arr\_delay,  
 speed = (distance / air\_time)\*60)

*Create a plot showing the relationship between the speed and time gain of each flight, adding appropriate axis and title labels.*

# Creating a scatter   
flights\_sml %>%  
 ggplot(aes(x = speed,  
 y = gain,  
 color = distance)) +  
 geom\_point(alpha = 0.1) +  
 labs(x = "Speed (miles per hour)",  
 y = "Gain (minutes)",  
 title = "Michigan flights, 2019 - 2021",  
 caption = "Data source: anyFlights R package",  
 color = "Distance (miles)") +  
 ggthemes::theme\_few() +  
 theme(legend.position = "bottom",  
 legend.text = element\_text(angle = 45, hjust = 1))

## Warning: Removed 10239 rows containing missing values (`geom\_point()`).



## The group\_by() and summarize() functions

## Calculating average departure delay by date  
miFlights %>%   
 group\_by(year, month, day) %>%   
summarize(delay = mean(dep\_delay, na.rm = TRUE)) %>%   
 ungroup() %>%  
 slice\_head(n = 5)

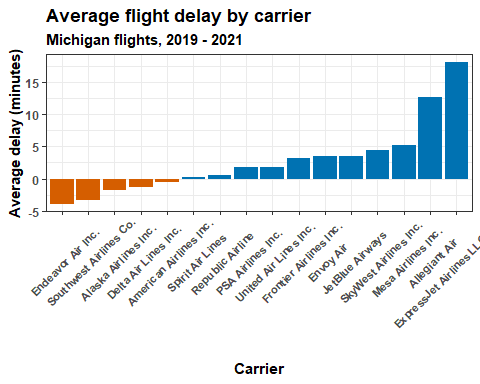
## `summarise()` has grouped output by 'year', 'month'. You can override using the  
## `.groups` argument.

## # A tibble: 5 × 4  
## year month day delay  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 1 1 8.58   
## 2 2019 1 2 12.5   
## 3 2019 1 3 0.721  
## 4 2019 1 4 -0.477  
## 5 2019 1 5 2.17

Reproducing the waterfall plot.

# Calculating average delay by carrier  
delaySummary <- miFlights %>%   
 group\_by(carrier\_name) %>%   
 summarize(Delay = mean(arr\_delay, na.rm = T))

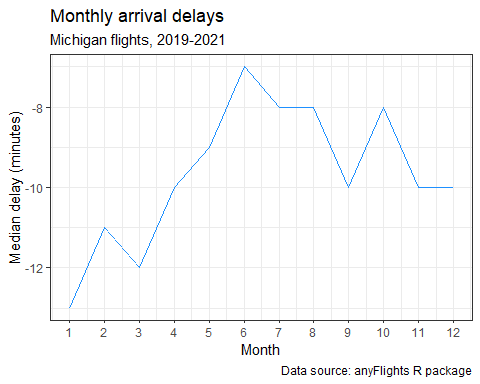
# Making a waterfall plot (bar plot)  
delaySummary %>%  
 ggplot(aes(x = fct\_reorder(carrier\_name, Delay),  
 y = Delay,  
 fill = Delay > 0)) +  
 geom\_col() +  
 labs(title = "Average flight delay by carrier",  
 subtitle = "Michigan flights, 2019 - 2021",  
 x = "Carrier" ,  
 y = "Average delay (minutes)") +  
 scale\_fill\_manual(values = c("#D55E00", "#0072B2")) +  
 theme\_bw() +  
 theme(legend.position = "none",  
 text = element\_text(face = "bold"),  
 axis.text.x = element\_text(angle = 45,  
 size = 8, vjust = 0.7))



# Create a data frame summarizing the median flight delay (arr\_delay) by month. Which month has the worst delays? In which month are flights most early / on-time?  
monthlyDelays <- miFlights %>% group\_by(month) %>%  
 summarize(delay = median(arr\_delay, na.rm = T))  
  
monthlyDelays %>% knitr::kable()

| month | delay |
| --- | --- |
| 1 | -13 |
| 2 | -11 |
| 3 | -12 |
| 4 | -10 |
| 5 | -9 |
| 6 | -7 |
| 7 | -8 |
| 8 | -8 |
| 9 | -10 |
| 10 | -8 |
| 11 | -10 |
| 12 | -10 |

# Creating a line chart   
monthlyDelays %>% ggplot(aes(x = month,  
 y = delay)) +  
 geom\_line(color = "dodgerblue") +  
 labs(title = "Monthly arrival delays",  
 x = "Month",  
 y = "Median delay (minutes)" ,  
 subtitle = "Michigan flights, 2019-2021",  
 caption = "Data source: anyFlights R package") +  
 scale\_x\_continuous(breaks = 1:12) +  
 theme\_bw()



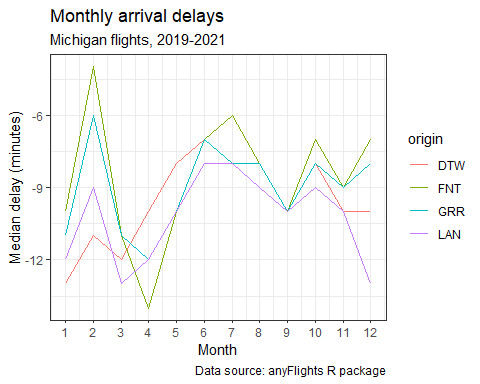
# Create a data frame summarizing the median flight delay (arr\_delay) by month. Which month has the worst delays? In which month are flights most early / on-time?  
monthlyAirportDelays <- miFlights %>% group\_by(month, origin) %>%  
 summarize(delay = median(arr\_delay, na.rm = T))

## `summarise()` has grouped output by 'month'. You can override using the  
## `.groups` argument.

monthlyAirportDelays %>% knitr::kable()

| month | origin | delay |
| --- | --- | --- |
| 1 | DTW | -13 |
| 1 | FNT | -10 |
| 1 | GRR | -11 |
| 1 | LAN | -12 |
| 2 | DTW | -11 |
| 2 | FNT | -4 |
| 2 | GRR | -6 |
| 2 | LAN | -9 |
| 3 | DTW | -12 |
| 3 | FNT | -11 |
| 3 | GRR | -11 |
| 3 | LAN | -13 |
| 4 | DTW | -10 |
| 4 | FNT | -14 |
| 4 | GRR | -12 |
| 4 | LAN | -12 |
| 5 | DTW | -8 |
| 5 | FNT | -10 |
| 5 | GRR | -10 |
| 5 | LAN | -10 |
| 6 | DTW | -7 |
| 6 | FNT | -7 |
| 6 | GRR | -7 |
| 6 | LAN | -8 |
| 7 | DTW | -8 |
| 7 | FNT | -6 |
| 7 | GRR | -8 |
| 7 | LAN | -8 |
| 8 | DTW | -8 |
| 8 | FNT | -8 |
| 8 | GRR | -8 |
| 8 | LAN | -9 |
| 9 | DTW | -10 |
| 9 | FNT | -10 |
| 9 | GRR | -10 |
| 9 | LAN | -10 |
| 10 | DTW | -8 |
| 10 | FNT | -7 |
| 10 | GRR | -8 |
| 10 | LAN | -9 |
| 11 | DTW | -10 |
| 11 | FNT | -9 |
| 11 | GRR | -9 |
| 11 | LAN | -10 |
| 12 | DTW | -10 |
| 12 | FNT | -7 |
| 12 | GRR | -8 |
| 12 | LAN | -13 |

# Creating a line chart   
monthlyAirportDelays %>% ggplot(aes(x = month,  
 y = delay, color = origin)) +  
 geom\_line() +  
 labs(title = "Monthly arrival delays",  
 x = "Month",  
 y = "Median delay (minutes)",  
 subtitle = "Michigan flights, 2019-2021",  
 caption = "Data source: anyFlights R package") +  
 scale\_x\_continuous(breaks = 1:12) +  
 theme\_bw()



# Create a line chart showing the average daily flight delay across time for each of the major airports

# Assuming the columns are named year, month, and day  
miFlights <- miFlights %>%  
 mutate(date = ymd(paste(year, month, day, sep = "-")))  
  
# Now calculating average daily flight delay and create the line chart  
averageDailyDelays <- miFlights %>%  
 group\_by(date, origin) %>%  
 summarize(avg\_delay = mean(arr\_delay, na.rm = TRUE))

## `summarise()` has grouped output by 'date'. You can override using the  
## `.groups` argument.

# Create a line chart  
ggplot(averageDailyDelays, aes(x = date, y = avg\_delay, color = origin)) +  
 geom\_line() +  
 labs(title = "Average Daily Flight Delay Across Major Airports",  
 x = "Date",  
 y = "Average Delay (minutes)",  
 color = "Airport") +  
 theme\_minimal()

