Data Transformation & Data Visualization Homework

Rujipas Mew

2024-08-05

Instruction

Homework Data Transformation

Write 5 codes to query data from **nycflights23** dataset with R Markdown.

Homework Data Visualization

Write 5 codes to create graphs from **nycflights23** dataset using **ggplots** package with R Markdown.

Download library

```
## Load library
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(nycflights23)
```

Inspect data

```
## Inspect nycflights23 data
ls("package:nycflights23")
```

```
## [1] "airlines" "airports" "flights"  "planes"   "weather"
```

```r
## Overview of dataset "flights"
str(flights)
```

```
## tibble [435,352 x 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:435352] 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ month         : int [1:435352] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:435352] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:435352] 1 18 31 33 36 503 520 524 537 547 ...
##  $ sched_dep_time: int [1:435352] 2038 2300 2344 2140 2048 500 510 530 520 545 ...
##  $ dep_delay     : num [1:435352] 203 78 47 173 228 3 10 -6 17 2 ...
##  $ arr_time      : int [1:435352] 328 228 500 238 223 808 948 645 926 845 ...
##  $ sched_arr_time: int [1:435352] 3 135 426 2352 2252 815 949 710 818 852 ...
##  $ arr_delay     : num [1:435352] 205 53 34 166 211 -7 -1 -25 68 -7 ...
##  $ carrier       : chr [1:435352] "UA" "DL" "B6" "B6" ...
##  $ flight        : int [1:435352] 628 393 371 1053 219 499 996 981 206 225 ...
##  $ tailnum       : chr [1:435352] "N25201" "N830DN" "N807JB" "N265JB" ...
##  $ origin        : chr [1:435352] "EWR" "JFK" "JFK" "JFK" ...
##  $ dest          : chr [1:435352] "SMF" "ATL" "BQN" "CHS" ...
##  $ air_time      : num [1:435352] 367 108 190 108 80 154 192 119 258 157 ...
##  $ distance      : num [1:435352] 2500 760 1576 636 488 ...
##  $ hour          : num [1:435352] 20 23 23 21 20 5 5 5 5 5 ...
##  $ minute        : num [1:435352] 38 0 44 40 48 0 10 30 20 45 ...
##  $ time_hour     : POSIXct[1:435352], format: "2023-01-01 20:00:00" "2023-01-01 23:00:00" ...
```

```r
## Overview of dataset "airlines"
str(airlines)
```

```
## tibble [14 x 2] (S3: tbl_df/tbl/data.frame)
##  $ carrier: chr [1:14] "9E" "AA" "AS" "B6" ...
##  $ name   : chr [1:14] "Endeavor Air Inc." "American Airlines Inc." "Alaska Airlines Inc." "JetBlue
```

```r
## Overview of dataset "airports"
str(airports)
```

```
## tibble [1,251 x 8] (S3: tbl_df/tbl/data.frame)
##  $ faa  : chr [1:1251] "AAF" "AAP" "ABE" "ABI" ...
##  $ name : chr [1:1251] "Apalachicola Regional Airport" "Andrau Airpark" "Lehigh Valley International
##  $ lat  : num [1:1251] 29.7 29.7 40.7 32.4 67.1 ...
##  $ lon  : num [1:1251] -85 -95.6 -75.4 -99.7 -157.9 ...
##  $ alt  : num [1:1251] 20 79 393 1791 334 ...
##  $ tz   : num [1:1251] -5 -6 -5 -6 -9 -7 -6 -5 -5 -6 ...
##  $ dst  : chr [1:1251] "A" "A" "A" "A" ...
##  $ tzone: chr [1:1251] "America/New_York" "America/Chicago" "America/New_York" "America/Chicago" ...
```

```r
## Overview of dataset "planes"
str(planes)
```

```
## tibble [4,840 x 9] (S3: tbl_df/tbl/data.frame)
##  $ tailnum     : chr [1:4840] "N101DQ" "N101DU" "N101HQ" "N101NN" ...
##  $ year        : int [1:4840] 2020 2018 2007 2013 2020 NA 2007 2013 1998 NA ...
##  $ type        : chr [1:4840] "Fixed wing multi engine" "Fixed wing multi engine" "Fixed wing multi
##  $ manufacturer: chr [1:4840] "AIRBUS" "C SERIES AIRCRAFT LTD PTNRSP" "EMBRAER-EMPRESA BRASILEIRA DE
##  $ model       : chr [1:4840] "A321-211" "BD-500-1A10" "ERJ 170-200 LR" "A321-231" ...
##  $ engines     : int [1:4840] 2 2 2 2 2 2 2 2 2 2 ...
##  $ seats       : int [1:4840] 199 133 80 379 199 133 80 379 182 133 ...
##  $ speed       : int [1:4840] 0 0 0 0 0 0 0 0 0 0 ...
##  $ engine      : chr [1:4840] "Turbo-fan" "Turbo-fan" "Turbo-fan" "Turbo-fan" ...
```

```
## Overview of dataset "weather"
str(weather)
```

```
## tibble [26,204 x 15] (S3: tbl_df/tbl/data.frame)
##  $ origin    : chr [1:26204] "JFK" "JFK" "JFK" "JFK" ...
##  $ year      : int [1:26204] 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ month     : int [1:26204] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day       : int [1:26204] 1 1 1 1 1 1 1 1 1 1 ...
##  $ hour      : int [1:26204] 0 1 2 3 4 5 6 7 8 9 ...
##  $ temp      : num [1:26204] NA NA NA NA NA NA NA NA NA NA ...
##  $ dewp      : num [1:26204] NA NA NA NA NA NA NA NA NA NA ...
##  $ humid     : num [1:26204] NA NA NA NA NA NA NA NA NA NA ...
##  $ wind_dir  : num [1:26204] 0 190 190 250 170 0 250 230 260 250 ...
##  $ wind_speed: num [1:26204] 0 4.6 5.75 5.75 8.06 ...
##  $ wind_gust : num [1:26204] 0 5.3 6.62 6.62 9.27 ...
##  $ precip    : num [1:26204] NA NA NA 0.02 NA NA NA NA NA NA ...
##  $ pressure  : num [1:26204] NA NA NA NA NA NA NA NA NA NA ...
##  $ visib     : num [1:26204] 0.25 2.5 0.25 4 0.75 0.75 0.24 0.5 8 5 ...
##  $ time_hour : POSIXct[1:26204], format: "2023-01-01 15:00:00" "2023-01-01 16:00:00" ...
```

Analyze the dataset

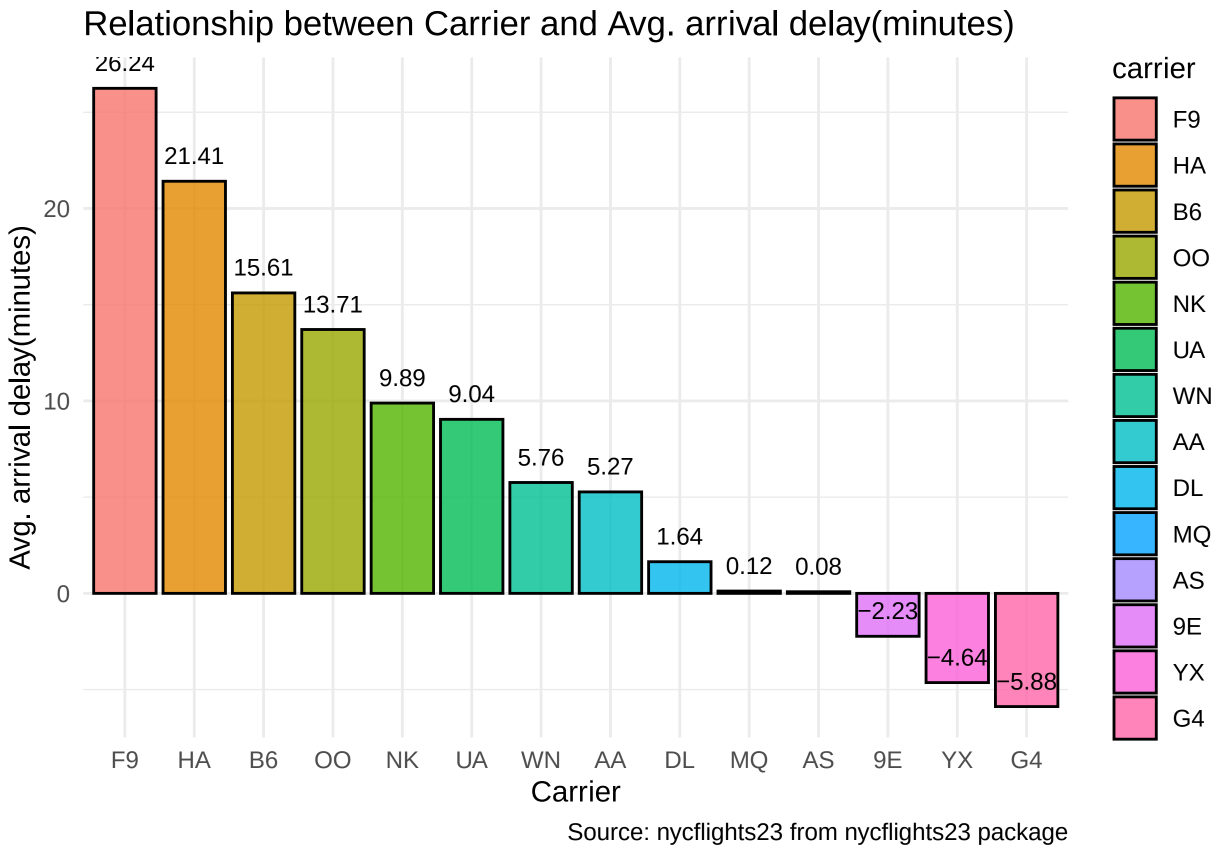
1. Finding the average, min, and max arrival delay by each carrier

```
best_carrier <- flights %>%
  group_by(carrier) %>%
  summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE),
            min_arr_delay = min(arr_delay, na.rm = TRUE),
            max_arr_delay = max(arr_delay, na.rm = TRUE)) %>%
  arrange(-mean_arr_delay)

print(best_carrier)
```

```
## # A tibble: 14 x 4
##    carrier mean_arr_delay min_arr_delay max_arr_delay
##    <chr>            <dbl>         <dbl>         <dbl>
##  1 F9               26.2            -66          1241
##  2 HA               21.4            -60          1086
##  3 B6               15.6            -92          1010
##  4 OO               13.7            -59          1409
##  5 NK                9.89           -74           878
##  6 UA                9.04           -80          1489
##  7 WN                5.76           -59           537
##  8 AA              5.27         -92       1812
##  9 DL              1.64         -97       1233
## 10 MQ              0.119        -46        161
## 11 AS              0.0844       -88       1012
## 12 9E             -2.23         -67       1271
## 13 YX             -4.64         -72       1162
## 14 G4             -5.88         -54       1382
```

Plot graph

```
best_carrier %>%
  mutate(carrier = factor(carrier, levels = carrier)) %>%
  ggplot(mapping = aes(x = carrier, y = mean_arr_delay,fill = carrier)) +
  geom_col(alpha = 0.8, color = "black") +
  theme_minimal() +
  labs(title  = "Relationship between Carrier and Avg. arrival delay(minutes)",
       x = "Carrier",
       y = "Avg. arrival delay(minutes)",
       caption = "Source: nycflights23 from nycflights23 package") +
  geom_text(aes(x = carrier, y = mean_arr_delay, label = round(mean_arr_delay,2)),
            vjust = -1,
            color = "black",
            size = 3)
```


Relationship between Carrier and Avg. arrival delay(minutes)

Source: nycflights23 from nycflights23 package

Observations

1. Carriers with the most negative values have the best on-time performance, while those with the highest positive values have the worst.

2. Carriers can be grouped into three clusters based on their average delays:

   - On-time: 9E, YX, G4
   - Low Delay: NK, UA, WN, AA, DL, MQ, AS
   - High Delay: F9, HA, B6, OO

3. Carriers like 9E, YX, G4 have the lowest average delays, with G4 arriving earlier than scheduled at an average of 5.88 minutes.

4. Carriers like F9, HA, B6, OO have the highest average delays, with F9 experiencing the longest delays at 26.24 minutes.

2. Finding the number of flights departed in each airports by month

```
flights_count_bymonth <- flights %>%
  left_join(airports, by = c("origin" = "faa")) %>%
  mutate(month = factor(month, levels = 1:12, labels = month.name)) %>%
  group_by(origin, name, month) %>%
  summarise(n = n()) %>%
  arrange(month, origin)
```

```
## ‘summarise()‘ has grouped output by ’origin’, ’name’. You can override using
## the ‘.groups‘ argument.
```

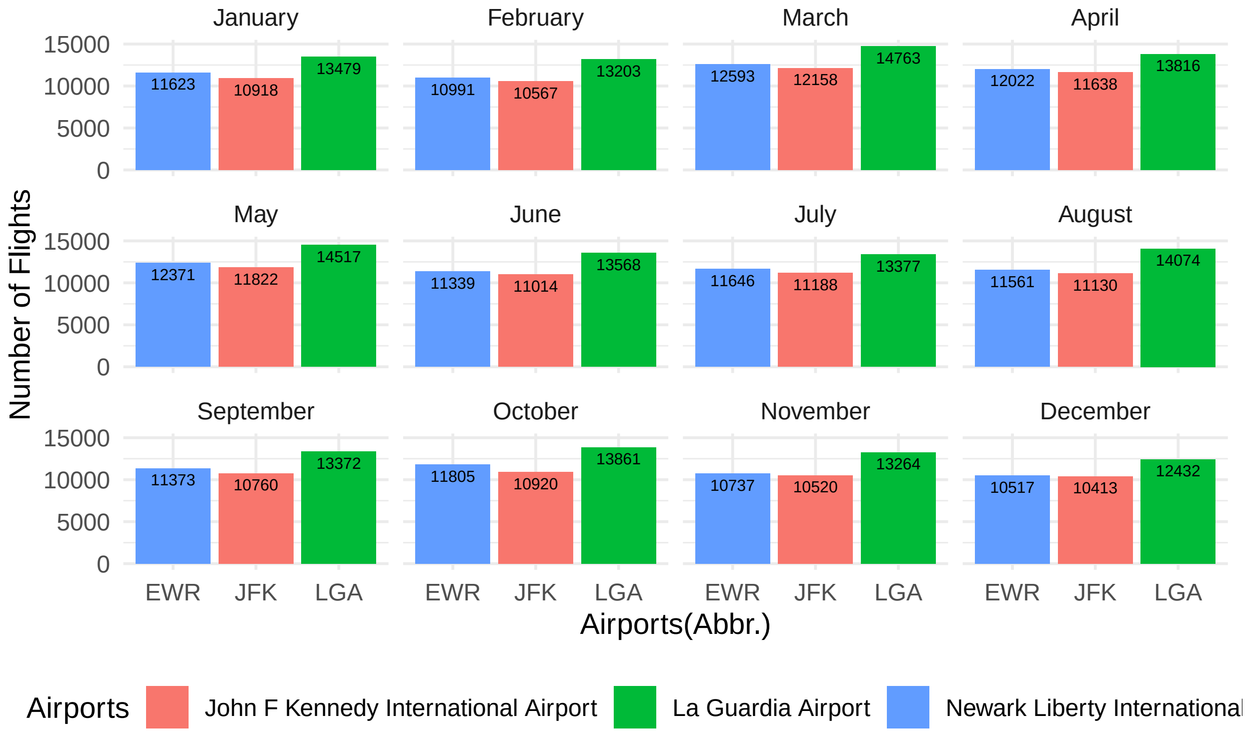
```
## can use 'count(origin, name, month)' instead of 'group_by()' %>% 'summarise()'

print(flights_count_bymonth)
```

```
## # A tibble: 36 x 4
## # Groups:   origin, name [3]
##    origin name                                month        n
##    <chr>  <chr>                               <fct>    <int>
##  1 EWR    Newark Liberty International Airport January  11623
##  2 JFK    John F Kennedy International Airport January  10918
##  3 LGA    La Guardia Airport                  January  13479
##  4 EWR    Newark Liberty International Airport February 10991
##  5 JFK    John F Kennedy International Airport February 10567
##  6 LGA    La Guardia Airport                  February 13203
##  7 EWR    Newark Liberty International Airport March    12593
##  8 JFK    John F Kennedy International Airport March    12158
##  9 LGA    La Guardia Airport                  March    14763
## 10 EWR    Newark Liberty International Airport April    12022
## # i 26 more rows
```

Plot graph

```
flights_count_bymonth %>%
  ggplot(aes(origin, n, fill = name)) +
  geom_col() +
  theme_minimal() +
  facet_wrap(~month) +
  labs(title = "The Total Flights Departed in each Airports by Month",
       x = "Airports(Abbr.)",
       y = "Number of Flights",
       fill = "Airports",
       caption = "Source: nycflights23 from nycflights23 package") +
  geom_text(aes(origin, n, label = n), vjust = 1.5, color = "black", size = 2) +
  theme(legend.position="bottom")
```

The Total Flights Departed in each Airports by Month



Source: nycflights23 from nycflights23 package

Observations

1. There seems to be a general trend of higher flight departures during the spring-summer months(Mar-Aug) compared to the autumn-winter months(Sept-Feb).
2. LGA consistently has the highest number of departures throughout the year.
3. Every airports have their highest number of departures in March.

3. Finding top 10 biggest plane

```
biggest_plane <- planes %>%
  group_by(manufacturer, model, type, engine) %>%
  summarize(max_seats = max(seats)) %>%
  arrange(desc(max_seats)) %>%
  head(10)
```

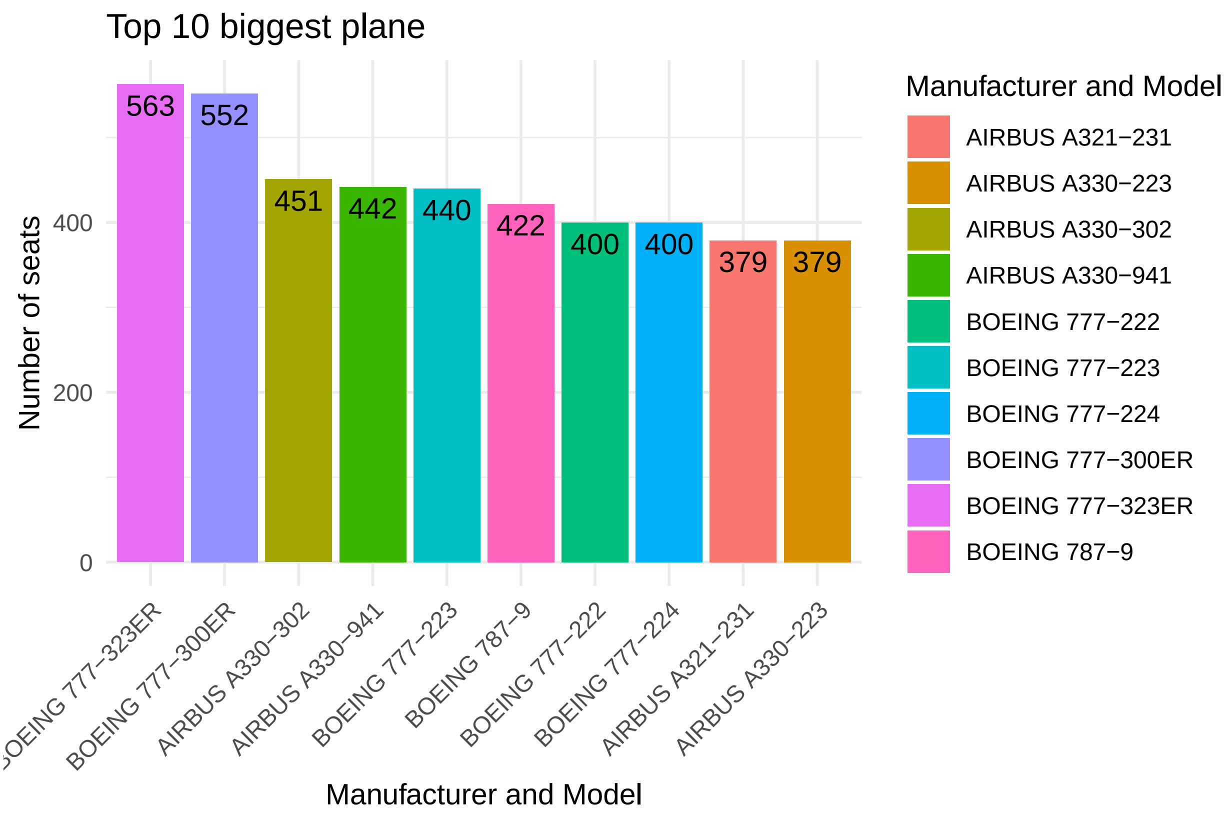
```
## `summarise()` has grouped output by 'manufacturer', 'model', 'type'. You can
## override using the `.groups` argument.
```

```
print(biggest_plane)
```

```
## # A tibble: 10 x 5
## # Groups:   manufacturer, model, type [10]
##    manufacturer model     type                     engine    max_seats
##    <chr>        <chr>     <chr>                    <chr>         <int>
##  1 BOEING       777-323ER Fixed wing multi engine Turbo-fan       563
##  2 BOEING       777-300ER Fixed wing multi engine Turbo-fan       552
##  3 AIRBUS       A330-302  Fixed wing multi engine Turbo-fan       451
##  4 AIRBUS       A330-941  Fixed wing multi engine Turbo-fan       442
##  5 BOEING       777-223   Fixed wing multi engine Turbo-fan       440
##  6 BOEING       787-9     Fixed wing multi engine Turbo-fan       422
##  7 BOEING       777-222   Fixed wing multi engine Turbo-fan       400
##  8 BOEING       777-224   Fixed wing multi engine Turbo-fan       400
##  9 AIRBUS       A321-231  Fixed wing multi engine Turbo-fan       379
## 10 AIRBUS       A330-223  Fixed wing multi engine Turbo-fan       379
```

Plot graph

```
biggest_plane %>%
  mutate(manufacturer_model = paste(manufacturer, model)) %>%
  ggplot(aes(x = reorder(manufacturer_model,max_seats, decreasing = TRUE), y = max_seats, fill = manufac
  geom_col() +
  theme_minimal() +
  labs(title = "Top 10 biggest plane",
       x = "Manufacturer and Model",
       y = "Number of seats",
       fill = "Manufacturer and Model",
       caption = "Source: nycflights23 from nycflights23 package") +
  geom_text(aes(manufacturer_model, max_seats, label = max_seats), vjust = 1.5, color = "black") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Source: nycflights23 from nycflights23 package

Observations

1. Boeing 777-323ER has the highest number of seats, with 563 seats.
2. Boeing 777-300ER comes in second with 552 seats
3. Boeing models occupy the majority of the top 10 list, with 6 out of 10 largest aircraft models.

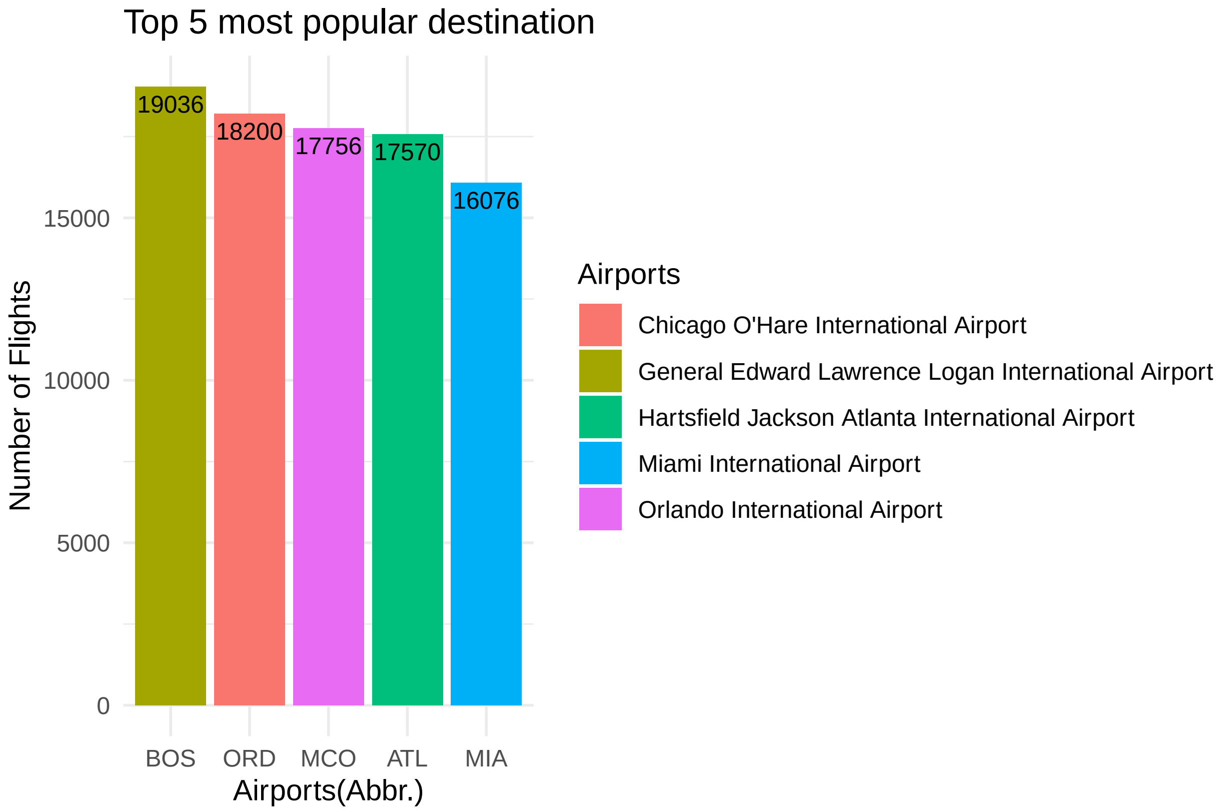
4. Finding top 5 most popular destination

```
pop_dest <- flights %>%
  left_join(airports, by = c("dest" = "faa")) %>%
  count(dest, name) %>%
  arrange(desc(n)) %>%
  head(5)

print(pop_dest)
```

```
## # A tibble: 5 x 3
##   dest  name                                            n
##   <chr> <chr>                                       <int>
## 1 BOS   General Edward Lawrence Logan International Airport 19036
## 2 ORD   Chicago O’Hare International Airport        18200
## 3 MCO   Orlando International Airport               17756
## 4 ATL   Hartsfield Jackson Atlanta International Airport    17570
## 5 MIA   Miami International Airport                 16076
```

Plot graph

```r
pop_dest %>%
  mutate(dest = factor(dest, levels = dest)) %>%
  ggplot(aes(dest, n, fill = name)) +
  geom_col() +
  theme_minimal() +
  labs(title = "Top 5 most popular destination",
       x = "Airports(Abbr.)",
       y = "Number of Flights",
       fill = "Airports",
       caption = "Source: nycflights23 from nycflights23 package") +
  geom_text(aes(dest, n, label = n), vjust = 1.5, color = "black", size= 3)
```



Source: nycflights23 from nycflights23 package

Observations

1. BOS (General Edward Lawrence Logan International Airport) is the most popular destination with the highest number of flights.
2. MIA (Miami International Airport) has the lowest number of flights among the top 5.
3. The ranking of airports from highest to lowest number of flights is: BOS, ORD, MCO, ATL, MIA

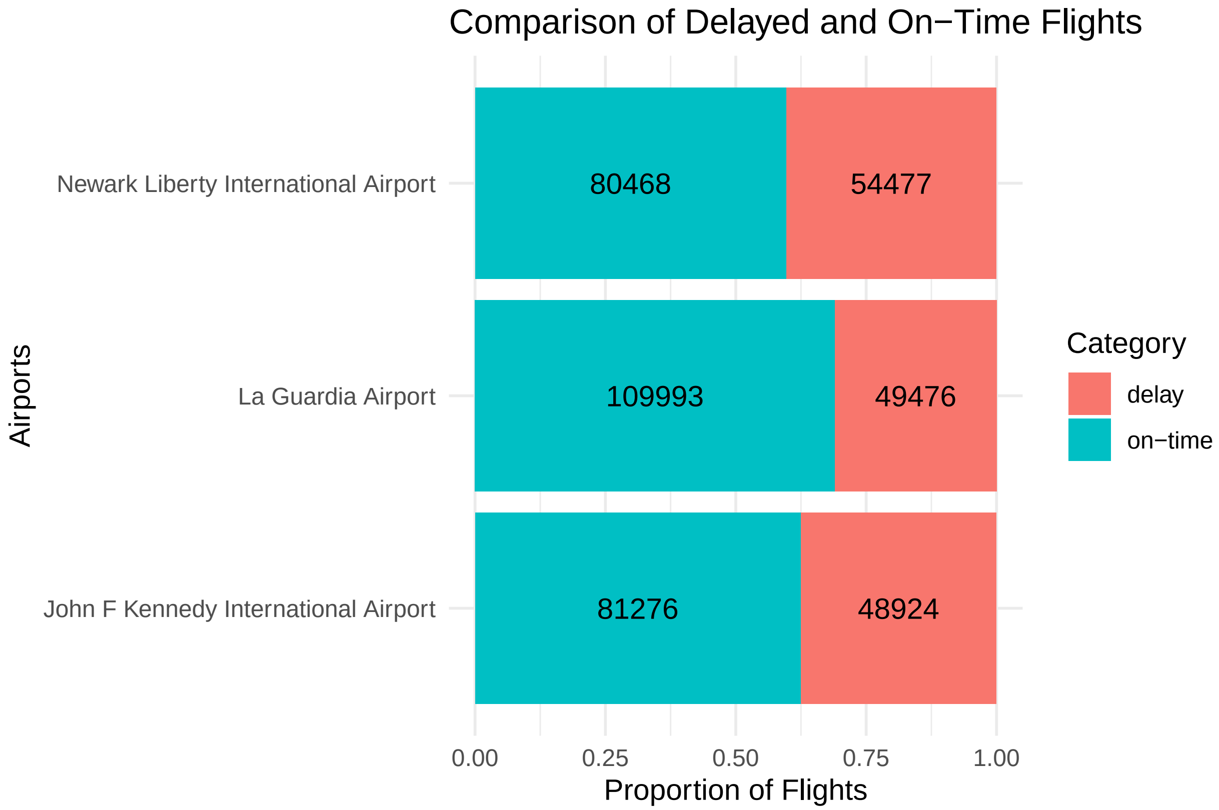
5. Finding Delayed and On-Time Departed Flights Across Airports

```r
flights_perf <- flights %>%
  filter(!is.na(flights$dep_delay)) %>% ## remove null in dep_delay
  left_join(airports, by = c("origin" = "faa")) %>%
  mutate(flg = if_else(dep_delay <= 0, "on-time", "delay")) %>%
  count(origin, name, flg)

print(flights_perf)
```

```
## # A tibble: 6 x 4
##   origin name                                 flg          n
##   <chr>  <chr>                                <chr>    <int>
## 1 EWR    Newark Liberty International Airport delay    54477
## 2 EWR    Newark Liberty International Airport on-time  80468
## 3 JFK    John F Kennedy International Airport delay    48924
## 4 JFK    John F Kennedy International Airport on-time  81276
## 5 LGA    La Guardia Airport                   delay    49476
## 6 LGA    La Guardia Airport                   on-time 109993
```

Plot Graph

```r
flights_perf %>%
  ggplot(aes(name, n, fill = flg)) +
  geom_col(position = "fill") +
  theme_minimal() +
  labs(title = "Comparison of Delayed and On-Time Flights",
       x = "Airports",
       y = "Proportion of Flights",
       fill = "Category",
       caption = "Source: nycflights23 from nycflights23 package") +
  geom_text(aes(name, n, label = n), position = position_fill(0.5), color = "black", size= 4) +
  coord_flip()
```



Source: nycflights23 from nycflights23 package

Observations

1. La Guardia Airport despite having the highest total number of flights, does not have the highest number of delayed flights.
2. Newark Liberty International Airport has the highest proportion of delayed flights.
3. John F Kennedy International Airport, while having fewer total flights than Newark Liberty International Airport, has a higher proportion of on-time departures.