

Average Airline Delay Analysis with nycflights13 Database

Introduction

This project analyzes flight data from the `nycflights13` dataset to report on the average delay time for each airline, helping our client make informed decisions about airline reliability. Using the Tidyverse suite in R, we apply various data wrangling techniques to produce an ordered table that shows each airline's average delay.

Tidyverse Analysis

1. Loading Tidyverse and nycflights13

```
library(tidyverse)
library(nycflights13)
```

- `library(tidyverse)`: Loads the Tidyverse, a collection of R packages like `dplyr` and `ggplot2` for data manipulation and visualization.
- `library(nycflights13)`: Loads the `nycflights13` package, which includes datasets on NYC flights from 2013, such as `flights` (flight details) and `airlines` (airline codes and names).

```
head(flights)
```

```
# A tibble: 6 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
# 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 <dtm>

```

```
colnames(flights)
```

```

[1] "year"          "month"          "day"            "dep_time"
[5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
[9] "arr_delay"      "carrier"        "flight"         "tailnum"
[13] "origin"         "dest"          "air_time"       "distance"
[17] "hour"          "minute"        "time_hour"

```

```
head(airlines)
```

```

# A tibble: 6 x 2
  carrier name
  <chr>   <chr>
1 9E      Endeavor Air Inc.
2 AA      American Airlines Inc.
3 AS      Alaska Airlines Inc.
4 B6      JetBlue Airways
5 DL      Delta Air Lines Inc.
6 EV      ExpressJet Airlines Inc.

```

`head(flights)` and `head(airlines)`: Displays the first few rows of the `flights` and `airlines` datasets to quickly inspect their structure and contents.

2. Cleaning and Merging Flight Data with Airline Names

```

time_tidyverse <- system.time(
flights_clean <- flights %>%
  select(carrier, dep_delay, arr_delay) %>%
  left_join(airlines, by = "carrier") %>%
  select(-carrier)
)

```

```
head(flights_clean)
```

```
# A tibble: 6 x 3
  dep_delay arr_delay name
    <dbl>    <dbl> <chr>
1         2         11 United Air Lines Inc.
2         4         20 United Air Lines Inc.
3         2         33 American Airlines Inc.
4        -1        -18 JetBlue Airways
5        -6        -25 Delta Air Lines Inc.
6        -4         12 United Air Lines Inc.
```

```
print(time_tidyverse)
```

```
user  system elapsed
0.03   0.00   0.05
```

- **time_tidyverse <- system.time(...)**: Measures how long it takes to run the code inside and saves the time as `time_tidyverse`.
- **flights_clean <- flights %>%**: Starts creating a new dataset called `flights_clean` by making changes to the `flights` data step-by-step.
- **select(carrier, dep_delay, arr_delay)**: Keeps only the `carrier`, `dep_delay` (departure delay), and `arr_delay` (arrival delay) columns from `flights`.
- **left_join(airlines, by = "carrier")**: Combines `flights` with the `airlines` data to add airline names, matching by `carrier` code.
- **select(-carrier)**: Removes the `carrier` code column, leaving only airline names and delay times.
- Running `head(flights_clean)` will display the first few rows of the `flights_clean` dataset. This dataset includes:
 1. **Departure delay (dep_delay)**: How many minutes the flight was delayed at departure.
 2. **Arrival delay (arr_delay)**: How many minutes the flight was delayed upon arrival.
 3. **Airline name (name)**: From the `airlines` dataset, matched using the `carrier` code.
- **print(time_tidyverse)**: Prints the execution time stored in `time_tidyverse`, showing how long the data cleaning process took.

3. Calculating Average Delays by Airline Using Tidyverse

```
flight_means <- flights_clean %>%
  group_by(name) %>%
  summarize(
    avg_dep_delay = mean(dep_delay, na.rm = TRUE ),
    avg_arr_delay = mean(arr_delay, na.rm = TRUE),
    .groups = 'drop'
  )%>%
  arrange(avg_dep_delay)

print(flight_means)
```

```
# A tibble: 16 x 3
  name                avg_dep_delay avg_arr_delay
  <chr>                <dbl>         <dbl>
1 US Airways Inc.      3.78          2.13
2 Hawaiian Airlines Inc. 4.90         -6.92
3 Alaska Airlines Inc.  5.80         -9.93
4 American Airlines Inc. 8.59          0.364
5 Delta Air Lines Inc.  9.26          1.64
6 Envoy Air           10.6          10.8
7 United Air Lines Inc. 12.1           3.56
8 SkyWest Airlines Inc. 12.6          11.9
9 Virgin America       12.9           1.76
10 JetBlue Airways     13.0           9.46
11 Endeavor Air Inc.    16.7           7.38
12 Southwest Airlines Co. 17.7           9.65
13 AirTran Airways Corporation 18.7          20.1
14 Mesa Airlines Inc.   19.0           15.6
15 ExpressJet Airlines Inc. 20.0           15.8
16 Frontier Airlines Inc. 20.2           21.9
```

- **flight_means <- flights_clean %>%**: Starts creating a new dataset called **flight_means** by processing **flights_clean** step-by-step.
- **group_by(name)**: Groups the data by airline name, so calculations are done for each airline separately.
- **summarize(...)**: Calculates the average delays for each airline:
 - i) **avg_dep_delay = mean(dep_delay, na.rm = TRUE)**: Finds the average departure delay for each airline, ignoring any missing values.

- ii) `avg_arr_delay = mean(arr_delay, na.rm = TRUE)`: Finds the average arrival delay for each airline, also ignoring any missing values.
- iii) `.groups = 'drop'`: Ensures the data is no longer grouped after summarizing, making future operations easier.
- `arrange(avg_dep_delay)`: Orders the results by average departure delay from lowest to highest, so airlines with the least delay appear first.
- `print(flight_means)`: Displays the final table `flight_means`, which lists each airline with its average departure and arrival delays.

Data Table Analysis

1. Loading Data Table Package

```
library(data.table)
```

- `library(data.table)`: Loads the `data.table` package, which is optimized for fast data manipulation.

2. Convert data to data.tables

```
flights <- as.data.table(flights)
airlines <- as.data.table(airlines)
```

- `flights <- as.data.table(flights)` and `airlines <- as.data.table(airlines)` convert the `flights` and `airlines` data frames into `data.table` format, which is faster for large datasets.

3. Cleaning and Merging Flight Data with Airline Names

```
time_data_table <- system.time({
  flights_clean <- merge(flights[, .(carrier, dep_delay, arr_delay)],
                        airlines[, .(carrier, name)],
                        by = "carrier",
                        all.x = TRUE)[, .(name, dep_delay, arr_delay)]
})
```

- `time_data_table <- system.time({...})`: Measures the time it takes to run the code inside and saves the time in `time_data_table`.
- Inside `system.time({...})`:
 - i) `flights[, .(carrier, dep_delay, arr_delay)]`: Selects only the `carrier`, `dep_delay`, and `arr_delay` columns from `flights`.
 - ii) `airlines[, .(carrier, name)]`: Selects only the `carrier` and `name` columns from `airlines`.
 - iii) `merge(..., by = "carrier", all.x = TRUE)`: Joins the selected columns from `flights` and `airlines` on the `carrier` column, keeping all rows from `flights`.
- 4. `[, .(name, dep_delay, arr_delay)]`: Keeps only the `name`, `dep_delay`, and `arr_delay` columns in the final result.

```
head(flights_clean)
```

	name	dep_delay	arr_delay
	<char>	<num>	<num>
1:	Endeavor Air Inc.	0	11
2:	Endeavor Air Inc.	-9	-2
3:	Endeavor Air Inc.	-3	-2
4:	Endeavor Air Inc.	-6	-1
5:	Endeavor Air Inc.	-8	-5
6:	Endeavor Air Inc.	0	-5

```
print(time_data_table)
```

user	system	elapsed
0.06	0.02	0.08

`print(time_data_table)`: Prints the execution time stored in `time_data_table`, showing how long the data manipulation took.

4. Calculating Average Delays by Airline Using Data.Table

```
flight_means <- flights_clean[, .(
  avg_dep_delay = mean(dep_delay, na.rm = TRUE),
  avg_arr_delay = mean(arr_delay, na.rm = TRUE)
), by = name][order(avg_dep_delay)]

print(flight_means)
```

	name	avg_dep_delay	avg_arr_delay
	<char>	<num>	<num>
1:	US Airways Inc.	3.782418	2.1295951
2:	Hawaiian Airlines Inc.	4.900585	-6.9152047
3:	Alaska Airlines Inc.	5.804775	-9.9308886
4:	American Airlines Inc.	8.586016	0.3642909
5:	Delta Air Lines Inc.	9.264505	1.6443409
6:	Envoy Air	10.552041	10.7747334
7:	United Air Lines Inc.	12.106073	3.5580111
8:	SkyWest Airlines Inc.	12.586207	11.9310345
9:	Virgin America	12.869421	1.7644644
10:	JetBlue Airways	13.022522	9.4579733
11:	Endeavor Air Inc.	16.725769	7.3796692
12:	Southwest Airlines Co.	17.711744	9.6491199
13:	AirTran Airways Corporation	18.726075	20.1159055
14:	Mesa Airlines Inc.	18.996330	15.5569853
15:	ExpressJet Airlines Inc.	19.955390	15.7964311
16:	Frontier Airlines Inc.	20.215543	21.9207048

- `flights_clean[, .(...), by = name]`: Groups the `flights_clean` data by `name` (airline name) and calculates:
 1. `avg_dep_delay = mean(dep_delay, na.rm = TRUE)`: The average departure delay for each airline, ignoring any missing values.
 2. `avg_arr_delay = mean(arr_delay, na.rm = TRUE)`: The average arrival delay for each airline, also ignoring any missing values.
- `[order(avg_dep_delay)]`: Orders the results by `avg_dep_delay` from smallest to largest.

Comparison of Tidyverse and Data Table

1. Tidyverse:

- **User Time:** 0.06 seconds
- **System Time:** 0.00 seconds
- **Elapsed Time:** 0.07 seconds

2. **data.table:**

- **User Time:** 0.05 seconds
- **System Time:** 0.00 seconds
- **Elapsed Time:** 0.06 seconds

Comparison Analysis

- **data.table** was slightly faster than Tidyverse**, with an elapsed time of **0.06 seconds** compared to **0.07 seconds** for Tidyverse.
- Both approaches show very similar performance, with **minimal differences** in this example.
- For larger datasets, **data.table** is generally expected to perform better due to its optimized memory usage, but in this case, the difference is minor.

This comparison suggests that for this dataset size, both approaches are nearly equivalent in speed, with data.table being slightly faster.

Observations

1. **Airlines with Low or Negative Delays:** Airlines like **US Airways Inc.**, **Hawaiian Airlines Inc.** and **Alaska Airlines Inc.** have lower average delays, with Hawaiian Airlines even showing a negative average arrival delay, indicating that flights often arrive ahead of schedule.
2. **Airlines with High Delays:** Airlines such as **ExpressJet Airlines Inc.** and **Frontier Airlines Inc.** have high average delay times, both for departures and arrivals, making them less reliable in terms of punctuality.
3. **Decision Insight:** Airlines with lower average delays (positive or negative close to zero) are generally more reliable for timely departures and arrivals, while airlines with higher delay averages may be less punctual.