Flight Data Analysis

Jose Michel Sammut

2024-07-29

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Use Ctrl + Alt + I (Windows/Linux) to insert a new code chunk in your RMarkdown document.

```r
# Load Necessary Libraries
suppressPackageStartupMessages({
library(arrow)
library(data.table)
library(dplyr)
library(lubridate)
})
setwd("D:/University of London/Programming for Data Science ST2195/ST2195_coursework_2023-24/Part 2")
```

Question 2

The Data Expo 2009: Airline On-Time Dataset provides detailed flight arrival and departure information for commercial flights within the USA from October 1987 to April 2008. With nearly 120 million records, this dataset includes variables such as departure and arrival delays, flight cancellations, and diversion indicators, offering a comprehensive view of on-time performance and operational disruptions across nearly two decades. For this analysis, we have selected a subset of the data covering the years 1998 to 2007 to focus on a decade of flight performance.

Parquet is a custom binary format designed specifically for the needs of big data. ***Source this script once to convert CSV Files to Parquet Format.***

```r
#source("convert_to_parquet.R")
```

(a) What are the best times and days of the week to minimise delays each year?

```r
# Function to compute the best times and days to minimize delays for a given year
compute_best_times <- function(data, year) {
  data %>%
    mutate(
      dep_hour = hour(hms::hms(DepTime)),
      dep_day_of_week = wday(ymd(paste(Year, Month, DayofMonth, sep = "-")), label = TRUE)
    ) %>%
    group_by(dep_day_of_week, dep_hour) %>%
    summarise(
      mean_dep_delay = mean(DepDelay, na.rm = TRUE),
      mean_arr_delay = mean(ArrDelay, na.rm = TRUE),
      .groups = 'drop'
    ) %>%
    summarise(
      year = year,
      best_dep_day = dep_day_of_week[which.min(mean_dep_delay)],
      best_dep_hour = dep_hour[which.min(mean_dep_delay)],
      best_arr_day = dep_day_of_week[which.min(mean_arr_delay)],
      best_arr_hour = dep_hour[which.min(mean_arr_delay)]
    )
}

# Directory containing the Parquet files
parquet_dir <- "dataverse_files/parquet_files"

# Initialize an empty list to store summaries
yearly_summaries <- list()

# Get a list of Parquet files
parquet_files <- list.files(parquet_dir, pattern = "\\.parquet$", full.names = TRUE)

# Process each year separately
for (file in parquet_files) {
  # Extract the year from the file name
  year <- as.integer(gsub(".*_(\\d{4}).*", "\\1", basename(file)))

  # Read the dataset for the year
  dataset <- open_dataset(file)

  # Filter data for the year and select relevant columns
  data <- dataset %>%
    select(Year, Month, DayofMonth, DayOfWeek, DepTime, ArrTime, DepDelay, ArrDelay) %>%
    collect()

  # Compute best times for the year
  summary <- compute_best_times(data, year)

  # Store the summary in the list
  yearly_summaries[[as.character(year)]] <- summary
}

# Combine all yearly summaries into a single data frame
final_summary <- bind_rows(yearly_summaries)

print(final_summary)
```

```
## # A tibble: 10 x 5
##     year best_dep_day best_dep_hour best_arr_day best_arr_hour
##    <int> <ord>                <int> <ord>                <int>
##  1  1998 Sat                      0 Sat                      0
##  2  1999 Tue                      0 Sat                      0
##  3  2000 Tue                      0 Sat                      0
##  4  2001 Tue                      0 Tue                      0
##  5  2002 Sat                      0 Sat                      0
##  6  2003 Sat                      0 Sat                      0
##  7  2004 Sat                      0 Sat                      0
##  8  2005 Sat                      0 Sat                      0
##  9  2006 Tue                      0 Sat                      0
## 10  2007 Sat                      0 Sat                      0
```