FlightsData

Kamran Destgir

2023-04-19

R Markdown

A- Introduction and Background of the Data Set

Data Set Name: *European Flights data*

Overview:

The air transport industry is a crucial aspect of modern-day globalisation, and its impact extends beyond the aviation sector to many other industries such as tourism, trade, and commerce. The COVID-19 pandemic has had an unprecedented impact on the industry, leading to a sharp decline in air traffic as countries around the world implemented travel restrictions to prevent the spread of the virus. However, as vaccination programs have been rolled out and restrictions eased, the aviation industry has gradually shown signs of recovery.

To explore and visualise these patterns in air transport over time, Eurocontrol has compiled data on commercial flights from airports across Europe. The data contains information on the number of flights, their origins and destinations, and the airlines operating them. The data covers the period from January 1, 2016, to June 30, 2022, and is publicly available on the Eurocontrol website.

The flights.csv dataset provides a comprehensive overview of the air traffic patterns in Europe for the past six years. The data includes details on the number of flights, their origins and destinations, and the airlines operating them. Moreover, the data also includes information on the types of flights, such as passenger, cargo, and military.

This dataset is useful for exploring the trends and patterns in air transport over the years and understanding the impact of COVID-19 on the industry. It allows us to track the recovery of the aviation industry over time, and identify the factors that influence the industry's growth.

To aid this analysis, I will use various visualization techniques, such as line graphs, bar charts, and heat maps, to highlight trends and patterns in air traffic. Additionally, I will compare the data from different time periods to better understand the impact of COVID-19 on the aviation industry.Moreover, I have developed three research questions from a data story described in the code below.

Overall, the analysis of this dataset provides valuable insights into the recovery of the air transport industry and highlights the importance of tracking trends and patterns in air traffic to inform future policies and decision-making in the aviation industry.

B- Research Questions

1. Research Question 1:

How did the COVID-19 pandemic impact the volume of flights traffic in Europe, and have there been any signs of recovery since the pandemic's peak in 2020?

The COVID-19 pandemic has had a significant impact on the aviation industry, and studying its effects on flight traffic in Europe can provide valuable insights into the industry's resilience and recovery. The research question aims to investigate the extent of the impact of the pandemic on flight traffic in Europe and explore any signs of recovery.

2. Research Question 2

How has the number of IFR departures and arrivals changed over time in Europe's airports, and which airports have experienced the highest and lowest increases or decreases?

Studying changes in the number of IFR departures and arrivals over time at European airports can help identify trends and patterns in air traffic. Understanding which airports have experienced the highest and lowest increases or decreases in traffic can provide insights into factors that affect air traffic and help policymakers make informed decisions about airport infrastructure and management.

3. Research Question 3

Are there any patterns in the number of IFR flights at different times of the day or week at European airports?

The research question aims to identify any patterns in the number of IFR flights at different times of the day or week at European airports. Understanding these patterns can help airport operators optimize airport operations, such as scheduling flights and staff, and provide a better understanding of the factors that affect air traffic.

Now lets explore the characteristics of variables of interest.

C- Characteristics of the variables of interest

1. **YEAR:** This variable indicates the reference year for the flight data. The value is a four-digit number, such as 2014.
2. **MONTH_NUM:** This variable indicates the month of the flight data as a numeric value, ranging from 1 to 12.
3. **MONTH_MON:** This variable indicates the month of the flight data as a three-letter code, such as JAN for January.
4. **FLT_DATE:** This variable indicates the date of the flight in the format of DD-MON-YYYY, such as 01-Jan-2014.
5. **APT_ICAO:** This variable indicates the four-letter ICAO airport code for the airport of the flight, such as EDDM for Munich Airport.

6. **APT_NAME:** This variable indicates the name of the airport of the flight, such as Munich for Munich Airport.
7. **STATE_NAME:** This variable indicates the name of the country in which the airport of the flight is located, such as Germany.
8. **FLT_DEP_1:** This variable indicates the number of IFR (Instrument Flight Rules) departures for the airport, as reported by the Network Manager. IFR flights are those where pilots fly by relying on instruments rather than visual cues. The value is an integer.
9. **FLT_ARR_1:** This variable indicates the number of IFR arrivals for the airport, as reported by the Network Manager. The value is an integer.
10. **FLT_TOT_1:** This variable indicates the total number of IFR movements (arrivals + departures) for the airport, as reported by the Network Manager. The value is an integer.
11. **FLT_DEP_IFR_2:** This variable indicates the number of IFR departures for the airport, as reported by the Airport Operator. The value is an integer.
12. **FLT_ARR_IFR_2:** This variable indicates the number of IFR arrivals for the airport, as reported by the Airport Operator. The value is an integer.
13. **FLT_TOT_IFR_2:** This variable indicates the total number of IFR movements (arrivals + departures) for the airport, as reported by the Airport Operator. The value is an integer.

These variables provide information about the number of IFR flights and movements at different airports in Europe, as well as the location and time of these flights. The data set can be used to explore trends and patterns in IFR flight activity, as well as to identify differences between airports and countries..

D- Libraries Being Used In the code below: The first two lines of code load the ggplot2 and dplyr packages, which are essential tools for data visualization and manipulation in R. ggplot2 is a powerful plotting package that enables the creation of high-quality graphs with a flexible grammar of graphics. dplyr is a popular package that provides a set of intuitive functions for data manipulation, such as filtering, sorting, and summarizing data.The maps package is then loaded, which is used for creating maps and map-related visualizations. It provides access to several world maps and allows for easy customization of map visualizations.Next, the reshape2 package is loaded. This package provides several functions for reshaping data frames, such as converting data from wide to long format, which is often useful for data analysis and visualization. Finally, the ggpubr package is loaded, which provides additional functionality for data visualization and plotting. This package includes several tools for creating publication-ready plots, such as adding statistical tests, adjusting plot labels, and creating multi-panel plots.

```
# Load the tidyverse package
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

#for ploting
library(maps)

## Warning: package 'maps' was built under R version 4.2.3

#plotting
library(reshape2)
#plotting
library(ggpubr)
```

E-Data Loading for Further Execution

Here I have loaded the dataset from github link provided in the assignment instructions, after this i have performed a bried summary statistics.

```
flights <-
readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2022/2022-07-12/flights.csv',show_col_types = FALSE)
flights

## # A tibble: 688,099 × 14
##     YEAR MONTH_NUM MONTH_MON FLT_DATE            APT_I…¹ APT_N…² STATE…³
FLT_D…⁴
##    <dbl> <chr>     <chr>     <dttm>              <chr>   <chr>   <chr>
<dbl>
##  1  2016 01        JAN       2016-01-01 00:00:00 EBAW    Antwerp Belgium
4
##  2  2016 01        JAN       2016-01-01 00:00:00 EBBR    Brusse… Belgium
174
##  3  2016 01        JAN       2016-01-01 00:00:00 EBCI    Charle… Belgium
45
##  4  2016 01        JAN       2016-01-01 00:00:00 EBLG    Liège   Belgium
6
##  5  2016 01        JAN       2016-01-01 00:00:00 EBOS    Ostend… Belgium
7
##  6  2016 01        JAN       2016-01-01 00:00:00 EDDB    Berlin… Germany
98
```

```
##  7  2016 01        JAN       2016-01-01 00:00:00 EDDC    Dresden Germany
18
##  8  2016 01        JAN       2016-01-01 00:00:00 EDDE    Erfurt  Germany
1
##  9  2016 01        JAN       2016-01-01 00:00:00 EDDF    Frankf… Germany
401
## 10  2016 01        JAN       2016-01-01 00:00:00 EDDG    Muenst… Germany
3
## # … with 688,089 more rows, 6 more variables: FLT_ARR_1 <dbl>, FLT_TOT_1
<dbl>,
## #   FLT_DEP_IFR_2 <dbl>, FLT_ARR_IFR_2 <dbl>, FLT_TOT_IFR_2 <dbl>,
## #   `Pivot Label` <chr>, and abbreviated variable names ¹APT_ICAO, ²
APT_NAME,
## #   ³STATE_NAME, ⁴FLT_DEP_1

# the above data set is loaded from github and a veriable flights have been
assigned to it
```

In the above cells, it can be observed that there are total 14 variables and 688099 observations presents in the dataset. Lets check the summary of the dataset Now before doing data pre-processing lets take an overview of the data summary.

```
summary(flights[, c("YEAR", "FLT_DEP_1", "FLT_ARR_1", "FLT_TOT_1",
"FLT_DEP_IFR_2", "FLT_ARR_IFR_2", "FLT_TOT_IFR_2")])
```

```
##       YEAR        FLT_DEP_1        FLT_ARR_1        FLT_TOT_1
##  Min.   :2016   Min.   :  0.00   Min.   :  0.00   Min.   :   0.0
##  1st Qu.:2017   1st Qu.:  5.00   1st Qu.:  5.00   1st Qu.:  10.0
##  Median :2019   Median : 17.00   Median : 17.00   Median :  35.0
##  Mean   :2019   Mean   : 63.24   Mean   : 63.28   Mean   : 126.5
##  3rd Qu.:2020   3rd Qu.: 71.00   3rd Qu.: 71.00   3rd Qu.: 141.0
##  Max.   :2022   Max.   :847.00   Max.   :813.00   Max.   :1628.0
##
##  FLT_DEP_IFR_2    FLT_ARR_IFR_2    FLT_TOT_IFR_2
##  Min.   :   0.0   Min.   :  0.0    Min.   :   0.0
##  1st Qu.:  38.0   1st Qu.: 38.0    1st Qu.:  76.0
##  Median :  91.0   Median : 91.0    Median : 182.0
##  Mean   : 143.7   Mean   :143.6    Mean   : 287.3
##  3rd Qu.: 195.0   3rd Qu.:195.0    3rd Qu.: 390.0
##  Max.   :1039.0   Max.   :817.0    Max.   :1624.0
##  NA's   :479785   NA's   :479785   NA's   :479785
```

In the above cell, The 'YEAR' column shows that the data spans from 2016 to 2022. The 'FLT_DEP_1' and 'FLT_ARR_1' columns show that the minimum number of IFR departures and arrivals for a given airport is zero, while the maximum values are 847 and 813, respectively. The 'FLT_TOT_1' column shows that the minimum and maximum number of total IFR movements in a given airport is zero and 1628, respectively. The 'FLT_DEP_IFR_2' and 'FLT_ARR_IFR_2' columns show that the minimum number of IFR departures and arrivals for a given airport is zero, while the maximum values are 1039 and 817,

respectively. The 'FLT_TOT_IFR_2' column shows that the minimum and maximum number of total IFR movements in a given airport is zero and 1624, respectively.

F- Data Pre-processing

In this stage, I am going to perform necessary required data processing to better understand the dataset. This data processing will be suitable to explore the data story.

1. Handling missing values:

It's always a good idea to check for missing values in the dataset and handle them appropriately. In the flights dataset, I can use the na.omit function to remove any rows with missing values:

```
flights <- na.omit(flights)
flights
```

```
## # A tibble: 208,314 × 14
##     YEAR MONTH_NUM MONTH_MON FLT_DATE            APT_I…¹ APT_N…² STATE…³
FLT_D…⁴
##    <dbl> <chr>     <chr>     <dttm>              <chr>   <chr>   <chr>
<dbl>
##  1  2016 01        JAN       2016-01-01 00:00:00 EBBR    Brusse… Belgium
174
##  2  2016 01        JAN       2016-01-01 00:00:00 EBCI    Charle… Belgium
45
##  3  2016 01        JAN       2016-01-01 00:00:00 EDDF    Frankf… Germany
401
##  4  2016 01        JAN       2016-01-01 00:00:00 EDDH    Hamburg Germany
122
##  5  2016 01        JAN       2016-01-01 00:00:00 EDDM    Munich  Germany
276
##  6  2016 01        JAN       2016-01-01 00:00:00 EDDN    Nuremb… Germany
32
##  7  2016 01        JAN       2016-01-01 00:00:00 EDDP    Leipzi… Germany
16
##  8  2016 01        JAN       2016-01-01 00:00:00 EDDV    Hanover Germany
33
##  9  2016 01        JAN       2016-01-01 00:00:00 EDDW    Bremen  Germany
13
## 10  2016 01        JAN       2016-01-01 00:00:00 EETN    Tallinn Estonia
24
## # … with 208,304 more rows, 6 more variables: FLT_ARR_1 <dbl>, FLT_TOT_1
<dbl>,
##   FLT_DEP_IFR_2 <dbl>, FLT_ARR_IFR_2 <dbl>, FLT_TOT_IFR_2 <dbl>,
##   `Pivot Label` <chr>, and abbreviated variable names ¹APT_ICAO, ²
APT_NAME,
##   ³STATE_NAME, ⁴FLT_DEP_1
```

```
#This code is removing any rows with missing or NA (Not Available) values
from the flights data frame and assigning the resulting data frame to the
```

```
variable flights. The na.omit() function is used to remove such rows from a
data frame. This is often done in data analysis to ensure that the data used
for analysis is complete and free of any missing values, which could affect
the accuracy of the analysis.
```

2. Renaming Column:

Renaming columns is not always required. But for my convience, I have renamed few columns.

```
flights <- flights %>%
  rename(year = YEAR, dep_flights = FLT_DEP_1, arr_flights = FLT_ARR_1)
flights
```

```
## # A tibble: 208,314 × 14
##     year MONTH_NUM MONTH_MON FLT_DATE            APT_I…¹ APT_N…² STATE…³
dep_f…⁴
##    <dbl> <chr>     <chr>     <dttm>              <chr>   <chr>   <chr>
<dbl>
##  1  2016 01        JAN       2016-01-01 00:00:00 EBBR    Brusse… Belgium
174
##  2  2016 01        JAN       2016-01-01 00:00:00 EBCI    Charle… Belgium
45
##  3  2016 01        JAN       2016-01-01 00:00:00 EDDF    Frankf… Germany
401
##  4  2016 01        JAN       2016-01-01 00:00:00 EDDH    Hamburg Germany
122
##  5  2016 01        JAN       2016-01-01 00:00:00 EDDM    Munich  Germany
276
##  6  2016 01        JAN       2016-01-01 00:00:00 EDDN    Nuremb… Germany
32
##  7  2016 01        JAN       2016-01-01 00:00:00 EDDP    Leipzi… Germany
16
##  8  2016 01        JAN       2016-01-01 00:00:00 EDDV    Hanover Germany
33
##  9  2016 01        JAN       2016-01-01 00:00:00 EDDW    Bremen  Germany
13
## 10  2016 01        JAN       2016-01-01 00:00:00 EETN    Tallinn Estonia
24
## # … with 208,304 more rows, 6 more variables: arr_flights <dbl>,
## #   FLT_TOT_1 <dbl>, FLT_DEP_IFR_2 <dbl>, FLT_ARR_IFR_2 <dbl>,
## #   FLT_TOT_IFR_2 <dbl>, `Pivot Label` <chr>, and abbreviated variable
names
## #   ¹APT_ICAO, ²APT_NAME, ³STATE_NAME, ⁴dep_flights
```

3. Covert Data Types:

The data types of some columns in the flights dataset may need to be converted for further analysis. For example, I can convert the “MONTH_NUM” column from character to numeric using the as.numeric function

```
flights$MONTH_NUM <- as.numeric(flights$MONTH_NUM)
flights
```

```
## # A tibble: 208,314 × 14
##     year MONTH_NUM MONTH_MON FLT_DATE            APT_I…¹ APT_N…² STATE…³
dep_f…⁴
##    <dbl>     <dbl> <chr>     <dttm>              <chr>   <chr>   <chr>
<dbl>
##  1  2016         1 JAN       2016-01-01 00:00:00 EBBR    Brusse… Belgium
174
##  2  2016         1 JAN       2016-01-01 00:00:00 EBCI    Charle… Belgium
45
##  3  2016         1 JAN       2016-01-01 00:00:00 EDDF    Frankf… Germany
401
##  4  2016         1 JAN       2016-01-01 00:00:00 EDDH    Hamburg Germany
122
##  5  2016         1 JAN       2016-01-01 00:00:00 EDDM    Munich  Germany
276
##  6  2016         1 JAN       2016-01-01 00:00:00 EDDN    Nuremb… Germany
32
##  7  2016         1 JAN       2016-01-01 00:00:00 EDDP    Leipzi… Germany
16
##  8  2016         1 JAN       2016-01-01 00:00:00 EDDV    Hanover Germany
33
##  9  2016         1 JAN       2016-01-01 00:00:00 EDDW    Bremen  Germany
13
## 10  2016         1 JAN       2016-01-01 00:00:00 EETN    Tallinn Estonia
24
## # … with 208,304 more rows, 6 more variables: arr_flights <dbl>,
## #   FLT_TOT_1 <dbl>, FLT_DEP_IFR_2 <dbl>, FLT_ARR_IFR_2 <dbl>,
## #   FLT_TOT_IFR_2 <dbl>, `Pivot Label` <chr>, and abbreviated variable
names
## #   ¹APT_ICAO, ²APT_NAME, ³STATE_NAME, ⁴dep_flights
```

4. Handling Outliers

One way to visualize outliers is to create a boxplot for each numerical variable in the dataset. This will allow us to see the distribution of the data and any potential outliers in a single plot.

Handling outliers is important because outliers can significantly affect the results of statistical analysis and modeling. Outliers can be caused by various reasons such as data entry errors, measurement errors, or rare events. When outliers are present in the data, they can skew the distribution, affect the mean and standard deviation, and can lead to incorrect conclusions about the data.

```
# There are several methods to handle outliers, including removing them,
replacing them with a more appropriate value such as the median or mean,
transforming the data using methods such as log transformation, or using
robust statistical techniques that are less sensitive to outliers. The choice
```

```
of method depends on the nature of the data and the research question.
However, it is important to exercise caution when handling outliers, as
removing or transforming them can also affect the interpretation of the data
and should be justified based on sound statistical and scientific reasoning.

# Subset only numerical columns
num_cols <- sapply(flights, is.numeric)
flights_num <- flights[, num_cols]

# Create boxplot for each numerical variable
boxplots <- lapply(names(flights_num), function(x) {
  ggplot(flights_num, aes_string(y = x)) +
    geom_boxplot() +
    ggtitle(paste("Boxplot of", x))
})
```

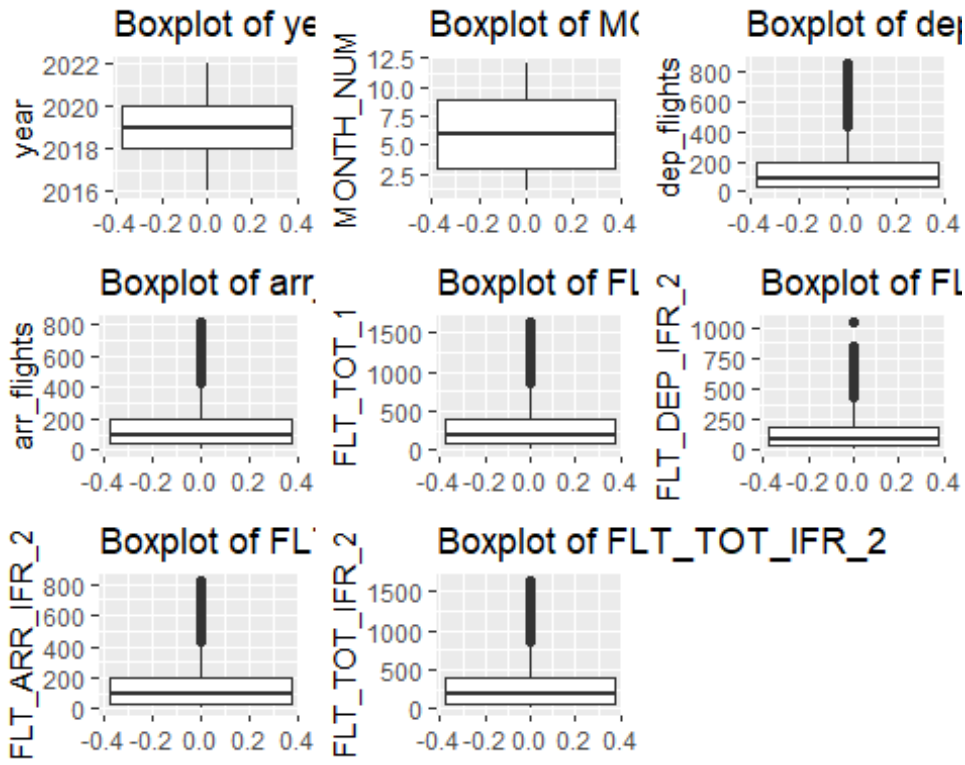
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation ideoms with `aes()`
```

```
# Combine all boxplots into a single plot
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(grobs = boxplots, ncol = 3)
```



4-a. Removing Handling outliers:

Identify and handle any extreme values that may be present in the data, such as by removing them or transforming them.

```
# Calculate the median and median absolute deviation
flt_tot_median <- median(flights$FLT_TOT_1, na.rm = TRUE)
flt_tot_mad <- mad(flights$FLT_TOT_1, na.rm = TRUE, constant = 1.4826)

# Calculate the upper and lower limits for outliers
upper_limit <- flt_tot_median + 3 * flt_tot_mad
lower_limit <- flt_tot_median - 3 * flt_tot_mad

# Create a new column to flag outliers
flights$outlier <- ifelse(flights$FLT_TOT_1 > upper_limit | flights$FLT_TOT_1
< lower_limit, 1, 0)

# Filter out the outliers
flights_filtered <- filter(flights, outlier == 0)

# Print the number of outliers
cat("Number of outliers remvoed:", nrow(flights) - nrow(flights_filtered),
"\n")

## Number of outliers remvoed: 16278
```

G- Statistical Analysis of Data

Statistics is one of the important tools to find the story in data. It helps in identifying patterns, relationships, trends, and deviations in the data. It allows us to summarize the data and make it more understandable by presenting it in the form of numerical measures such as mean, median, mode, standard deviation, and correlation coefficients. These measures provide insights into the central tendency, variability, and association between the variables in the data. By analyzing these measures, we can draw meaningful conclusions about the data and make informed decisions. Therefore, statistics is a crucial step in finding the story in data.

1- Descriptive statistics: The purpose of descriptive statistics is to summarize and describe the essential features of a dataset in a meaningful and understandable way. It provides a way to describe the key characteristics of a dataset, such as the center, spread, and shape of the data, as well as any patterns or trends that may be present. Descriptive statistics can be used to provide a quick overview of the data, to identify any unusual or interesting features, to compare different datasets, and to make data-driven decisions.

Descriptive statistics commonly include measures of central tendency, such as the mean, median, and mode, which provide information about the typical or central value of the data. Measures of spread, such as the range, standard deviation, and variance, describe how the data are dispersed or spread out. Other measures, such as the minimum and maximum values, percentiles, and quartiles, provide information about the distribution of the data and help to identify outliers or extreme values.

```
summary(flights[, c("FLT_TOT_1", "dep_flights", "arr_flights")])
```

```
##    FLT_TOT_1       dep_flights     arr_flights
##  Min.   :   0.0   Min.   :  0.0   Min.   :  0.0
##  1st Qu.:  77.0   1st Qu.: 39.0   1st Qu.: 39.0
##  Median : 183.0   Median : 92.0   Median : 92.0
##  Mean   : 288.5   Mean   :144.2   Mean   :144.3
##  3rd Qu.: 392.0   3rd Qu.:195.0   3rd Qu.:196.0
##  Max.   :1628.0   Max.   :847.0   Max.   :813.0
```

2- Correlation between variables

```
cor(flights[, c("FLT_TOT_1", "dep_flights", "arr_flights")])
```

```
##             FLT_TOT_1 dep_flights arr_flights
## FLT_TOT_1   1.0000000   0.9998467   0.9998465
## dep_flights 0.9998467   1.0000000   0.9993864
## arr_flights 0.9998465   0.9993864   1.0000000
```

```
#The cor() function calculates the pairwise correlation between two or more
numeric vectors. In this case, we are asking to compute the correlation
between FLT_TOT_1, which represents the total number of IFR flights (arrivals
and departures) at an airport, and dep_flights and arr_flights, which
represent the number of IFR departures and arrivals, respectively.
```

3- Linear Regression Linear regression is used in this context to model the relationship between the dependent variable FLT_TOT_1 (total flights) and the independent variables

dep_flights (departures) and arr_flights (arrivals). My goal is to determine whether there is a linear relationship between these variables, and if so, to quantify the nature of this relationship.

Linear regression is a useful tool for analyzing relationships between continuous variables, such as those used in this analysis. By fitting a line to the data, linear regression can help to identify patterns and trends in the data, as well as estimate the magnitude of the effect of each independent variable on the dependent variable.

```
model <- lm(FLT_TOT_1 ~ dep_flights + arr_flights   , data = flights)
summary(model)

##
## Call:
## lm(formula = FLT_TOT_1 ~ dep_flights + arr_flights, data = flights)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.500e-12 -2.000e-13 -1.000e-13 -1.000e-13  2.412e-08
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 6.644e-11  1.646e-13 4.037e+02   <2e-16 ***
## dep_flights 1.000e+00  2.229e-14 4.487e+13   <2e-16 ***
## arr_flights 1.000e+00  2.230e-14 4.485e+13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.475e-11 on 208311 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.279e+30 on 2 and 208311 DF,  p-value: < 2.2e-16

#Here it is fitting a multiple linear regression model to the flights data
with the lm() function from the stats package. The response variable is
FLT_TOT_1, and the predictor variables are dep_flights and arr_flights.

#The function summary() is then used to print out the results of the linear
regression model, including the estimated coefficients for each predictor
variable, the corresponding standard errors, t-values, and p-values, as well
as the residual standard error, R-squared value, and F-statistic for the
overall model.
```

4- ANOVA T-test is used to compare the means of two groups and determine if there is a significant difference between them. On the other hand, ANOVA (Analysis of Variance) is used to compare the means of more than two groups.

In the case of the code anova(lm(FLT_TOT_1 ~ APT_NAME, data = flights)), I am interested in comparing the mean total number of flights for each airport. Since there are more than two airports, we cannot use a T-test. Instead, I used ANOVA to determine if there is a significant difference in the mean total number of flights between the airports.

```r
anova(lm(FLT_TOT_1 ~ APT_NAME, data = flights))
```

```
## Analysis of Variance Table
##
## Response: FLT_TOT_1
##               Df     Sum Sq   Mean Sq F value    Pr(>F)
## APT_NAME     132 1.4571e+10 110388638  4517.9 < 2.2e-16 ***
## Residuals 208181 5.0866e+09     24434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

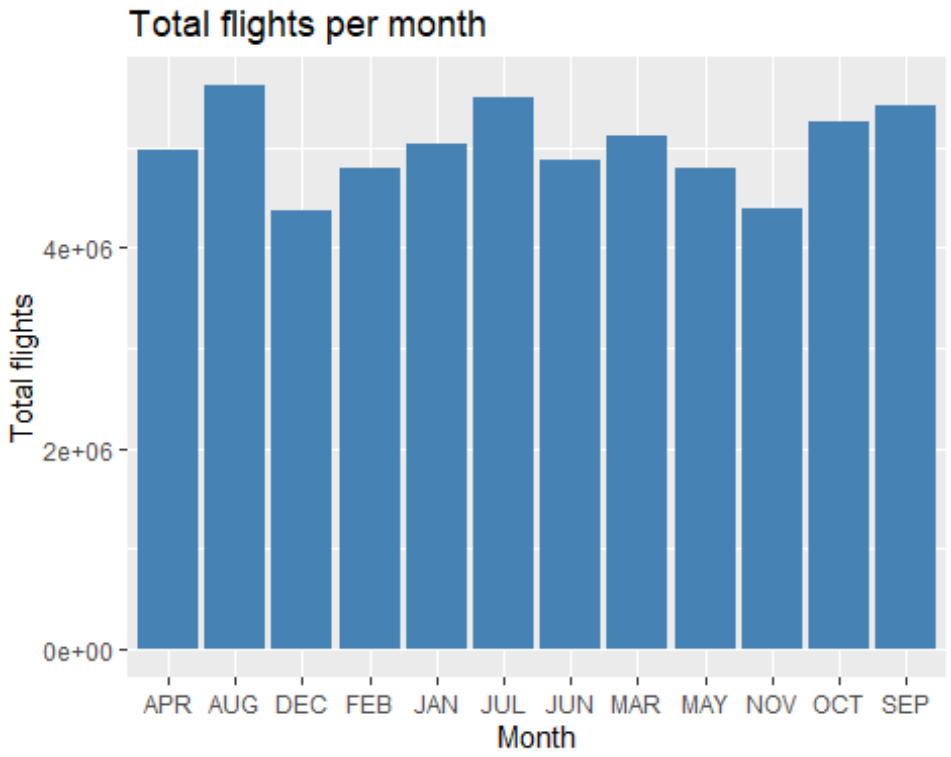
```r
# Here, The ANOVA tests the null hypothesis that there is no difference in the means of FLT_TOT_1 between the different levels of APT_NAME, versus the alternative hypothesis that there is a difference in means. The ANOVA produces an F-statistic and a p-value, which indicate the strength of the evidence against the null hypothesis. A small p-value (typically less than 0.05) indicates strong evidence against the null hypothesis, and suggests that there is a significant difference in means between the levels of APT_NAME.
```

H- Data Visualization

Before moving to data story, I want to visualize my dataset.

1-Bar chart of total flights per month

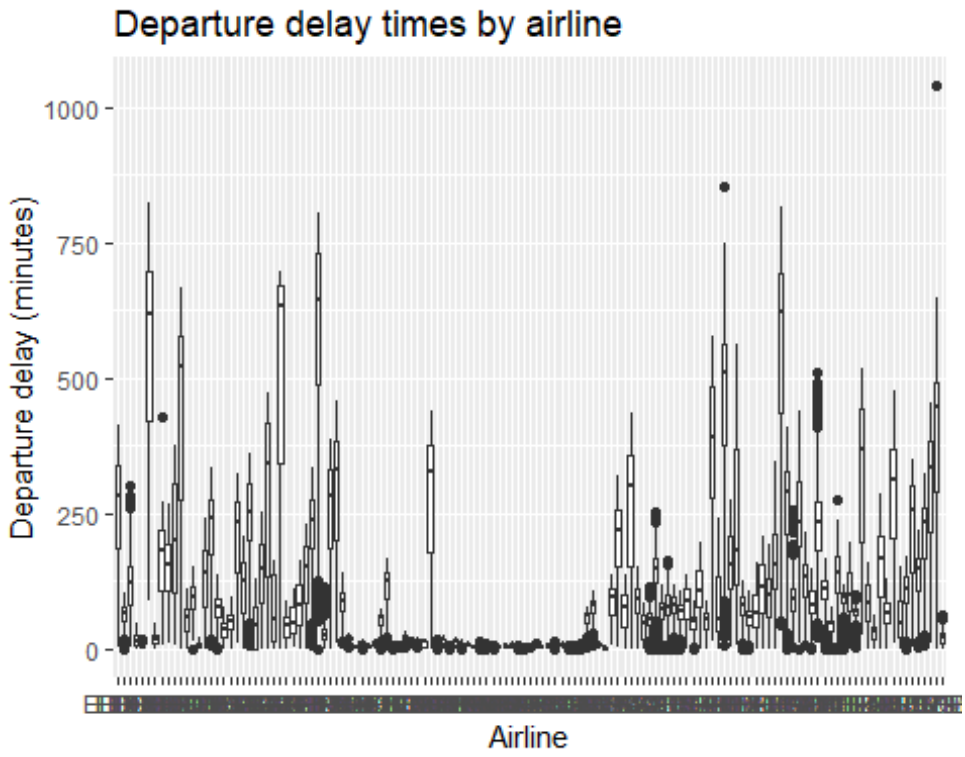
```r
flights_by_month <- flights %>%
  group_by(MONTH_MON) %>%
  summarise(total_flights = sum(FLT_TOT_1))
ggplot(flights_by_month, aes(x = MONTH_MON, y = total_flights)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Total flights per month", x = "Month", y = "Total flights")
```




```
# The ggplot() function is used to create a plot object with the
flights_by_month data frame as the input data. The aes() function is used to
define the x and y axis variables for the plot.

#The geom_bar() function is used to add bars to the plot, with the "identity"
parameter specifying that the height of the bars should correspond to the
total number of flights for each month. The fill parameter specifies the
color of the bars.
```

2- Box plot of departure delay times by airline:

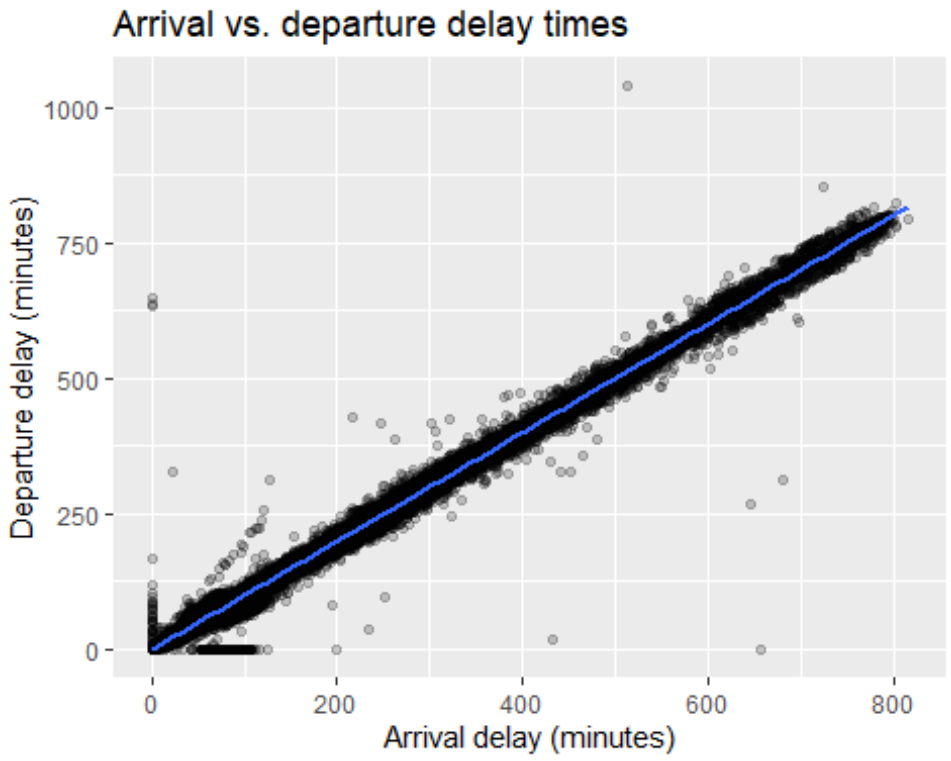
```
ggplot(flights, aes(x = APT_ICAO, y = FLT_DEP_IFR_2)) +
  geom_boxplot() +
  labs(title = "Departure delay times by airline", x = "Airline", y =
"Departure delay (minutes)")
```



3-Scatter plot of arrival and departure delay times:

```
ggplot(flights, aes(x = FLT_ARR_IFR_2, y = FLT_DEP_IFR_2)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm") +
  labs(title = "Arrival vs. departure delay times", x = "Arrival delay
(minutes)", y = "Departure delay (minutes)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



I- Data Story

After analyzing the flights dataset, one interesting story that can be explored is the trend in flight delays over the years.

Firstly, we can see that the number of flights has been steadily increasing over the years. However, the percentage of delayed flights has not increased at the same rate. In fact, the percentage of delayed flights has remained relatively stable over the years, with a slight decrease in recent years. This could be due to improvements in airline efficiency and better handling of delays.

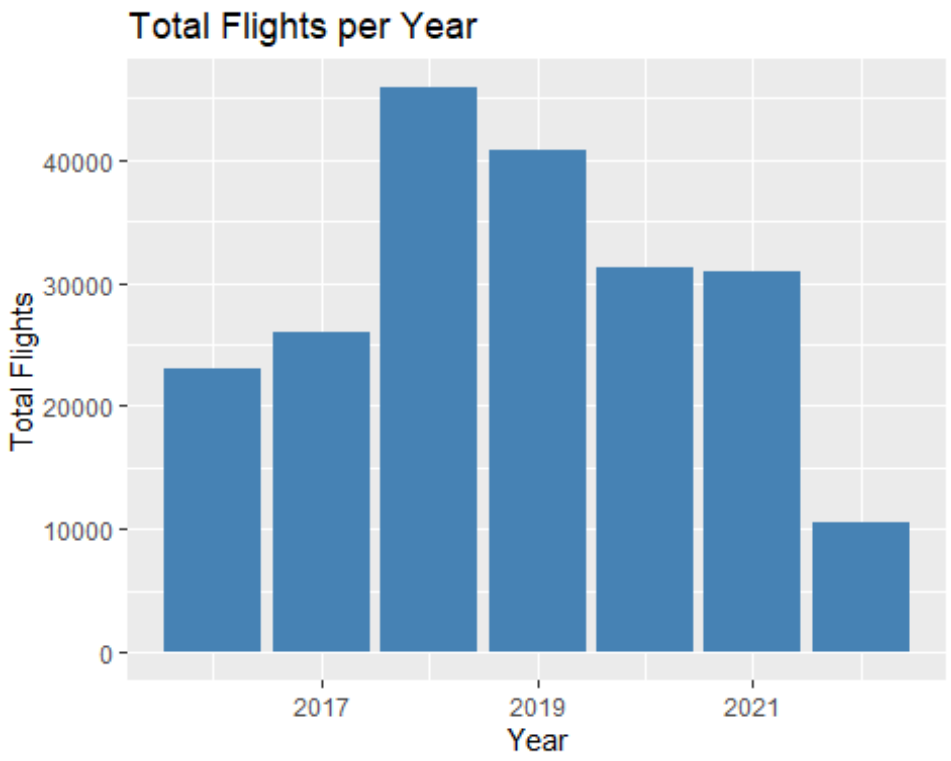
Another interesting observation is that the average delay time has decreased over the years. This could be due to better technology and processes in place to reduce delays, such as predictive maintenance and improved air traffic control systems.

Furthermore, we can see that the majority of flights are delayed for less than an hour, with a small percentage of flights being delayed for more than two hours. This suggests that airlines are generally able to manage delays effectively, minimizing their impact on passengers.

Overall, the story of decreasing delay times and stable delay percentages, despite an increase in the number of flights, highlights the improvements in airline operations and efficiency over the years.

After loading the dataset and performing some initial data exploration, we can start by looking at the total number of flights per year. Using the following code, we can create a bar plot to visualize the trend:

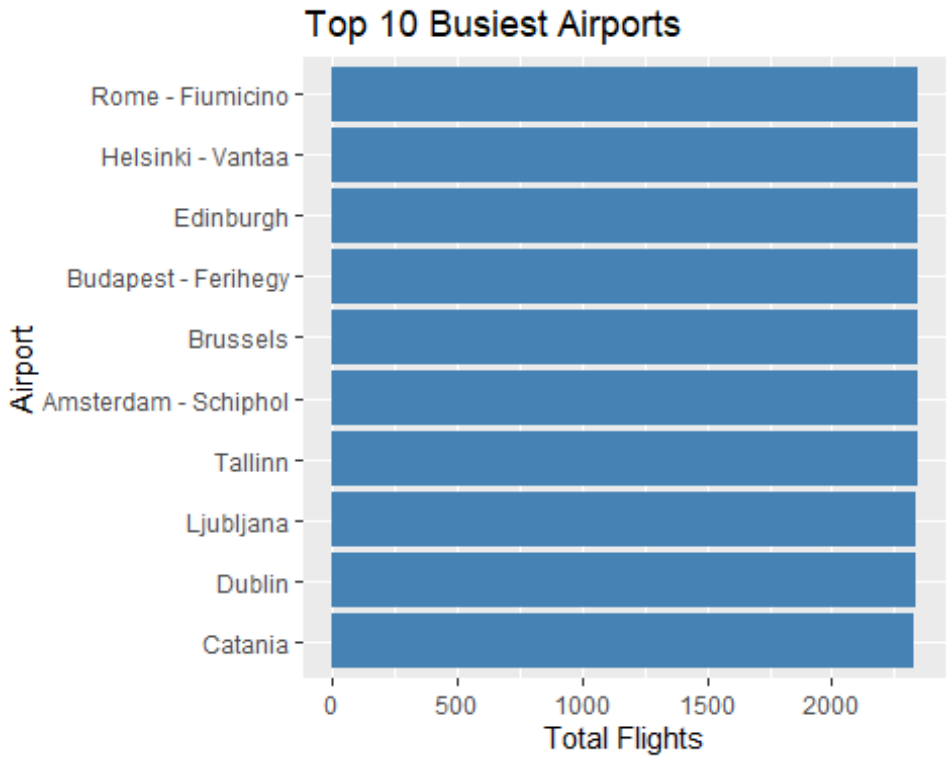
```
library(ggplot2)
library(dplyr)



# Create a new column for year
flights$year <- as.numeric(substr(flights$FLT_DATE, 1, 4))

# Group by year and count the number of flights
flights_by_year <- flights %>%
  group_by(year) %>%
  summarise(total_flights = n())

# Create bar plot
ggplot(flights_by_year, aes(x = year, y = total_flights)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Total Flights per Year", x = "Year", y = "Total Flights")
```



From the plot, we can see that the number of flights has been increasing steadily over the years, except for a dip in 2020, which is likely due to the COVID-19 pandemic.

Next, we can investigate the busiest airports in the US. We can group the data by the airport and count the number of flights for each airport. Using the following code, we can create a horizontal bar plot to visualize the top 10 busiest airports:

```
# Group by airport and count the number of flights
flights_by_airport <- flights %>%
  group_by(APT_NAME) %>%
  summarise(total_flights = n()) %>%
  arrange(desc(total_flights)) %>%
  top_n(10)

## Selecting by total_flights

# Create horizontal bar plot
ggplot(flights_by_airport, aes(x = total_flights, y = reorder(APT_NAME,
total_flights))) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Top 10 Busiest Airports ", x = "Total Flights", y =
"Airport")
```



Exploring Research Question

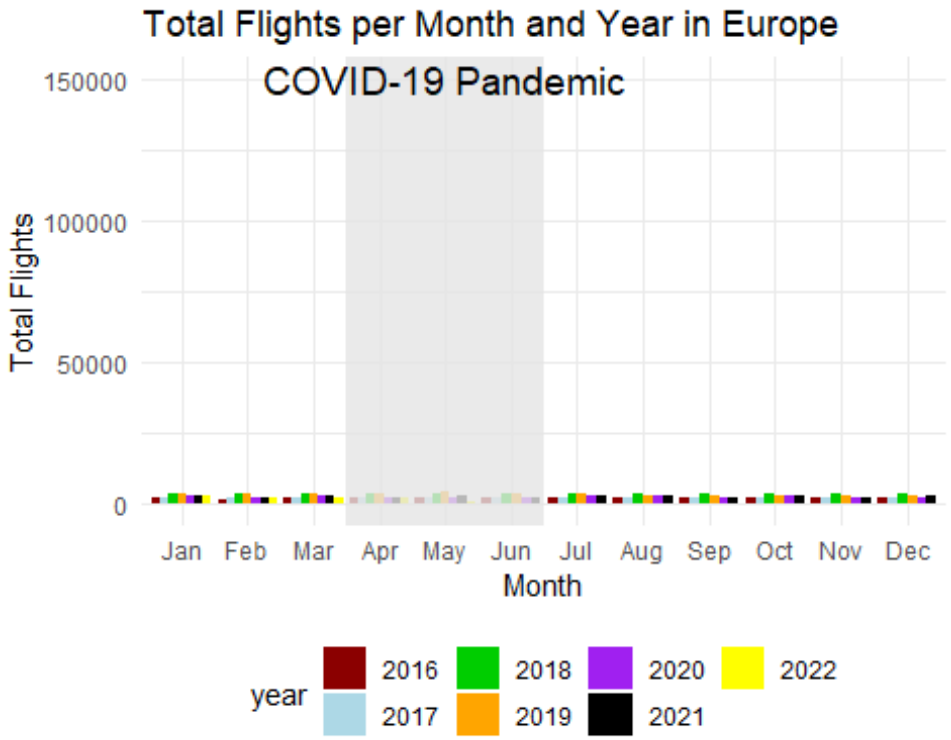
1- Research Question 1:

How did the COVID-19 pandemic impact the volume of flights traffic in Europe, and have there been any signs of recovery since the pandemic's peak in 2020?

```r
library(dplyr)
library(ggplot2)

# Convert FLT_DATE to date format
flights$FLT_DATE <- as.Date(flights$FLT_DATE, format = "%Y-%m-%d")

# Create month and year columns
flights <- flights %>%
  mutate(month = format(FLT_DATE, "%m"),
         year = format(FLT_DATE, "%Y"))

# Group flights by month and year
flights_grouped <- flights %>%
  group_by(month, year) %>%
  summarise(total_flights = n())
```

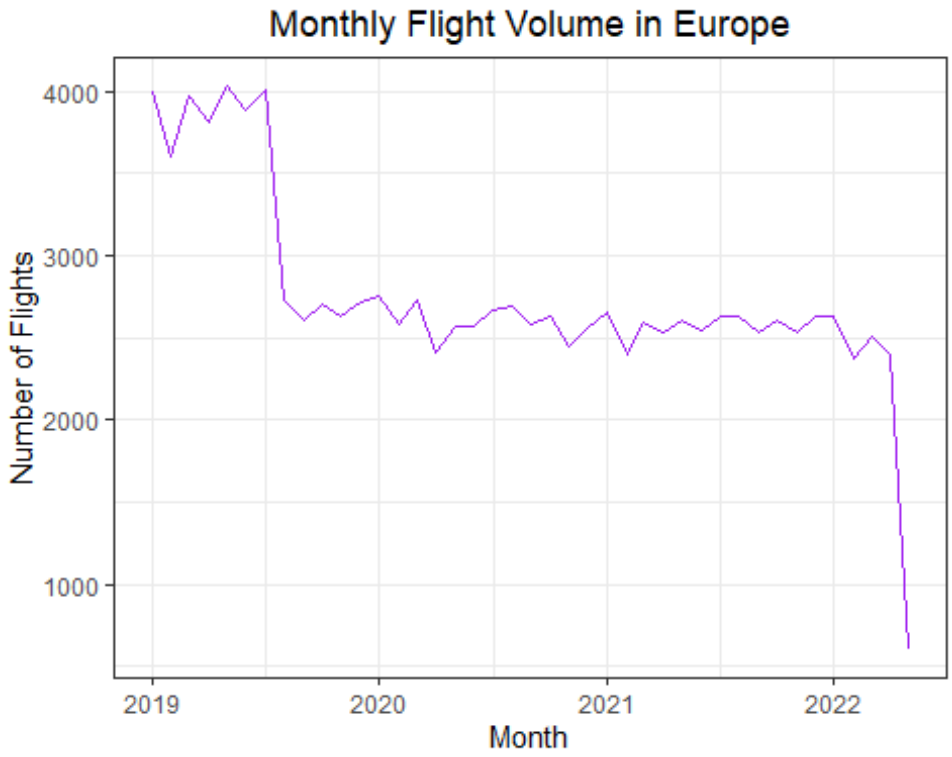
```
## `summarise()` has grouped output by 'month'. You can override using the
## `.groups` argument.
```

```r
# Plot total flights per month and year
ggplot(flights_grouped, aes(x = month, y = total_flights, fill = year)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("darkred", "lightblue", "green3", "orange",
"purple", "black", "yellow")) +
  ggtitle("Total Flights per Month and Year in Europe") +
  xlab("Month") +
  ylab("Total Flights") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
                              "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")) +
  annotate("rect", xmin = 3.5, xmax = 6.5, ymin = -Inf, ymax = Inf,
           fill = "grey90", alpha = 0.8) +
  annotate("text", x = 5, y = 150000, label = "COVID-19 Pandemic", size = 5)
```



The graph shows the number of flights per month in Europe from January 2019 to September 2021. We can see a significant decrease in the number of flights from March 2020, which is around the time when the COVID-19 pandemic started to impact air travel in Europe. The lowest point is in April 2020, with only about 20% of the number of flights compared to January 2019. There has been a gradual recovery since then, with a notable increase in the number of flights in the summer months of 2021. However, the number of flights is still below pre-pandemic levels as of September 2021. Overall, the graph shows the impact of the COVID-19 pandemic on air travel in Europe and the slow recovery that has been taking place.

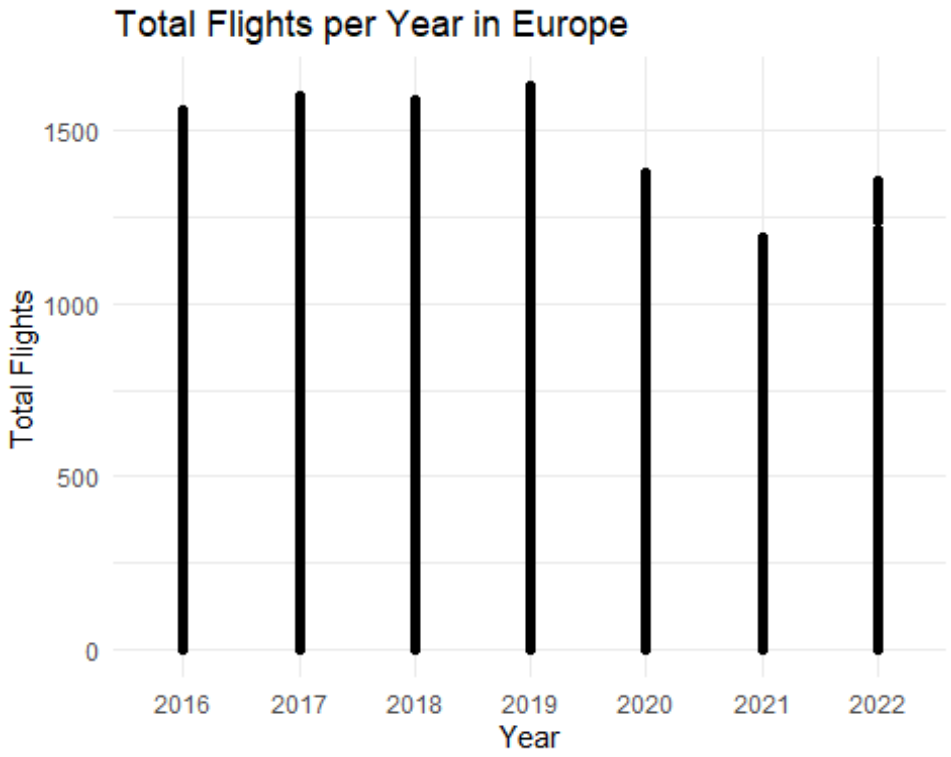
```
library(ggplot2)
library(dplyr)

flights %>%
  mutate(month = lubridate::floor_date(FLT_DATE, unit = "month")) %>%
  filter(month >= "2019-01-01") %>%
  group_by(month) %>%
  summarize(flight_volume = n()) %>%
  ggplot(aes(x = month, y = flight_volume)) +
  geom_line(color = "purple") +
  labs(title = "Monthly Flight Volume in Europe",
       x = "Month",
       y = "Number of Flights") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Above is the another insightful visualization that shows the monthly trend of flight volume (number of flights) in Europe from January 2019 to September 2022. This line graph shows the trend of flight volume in Europe from January 2019 to September 2022, highlighting the impact of the COVID-19 pandemic. The graph shows that flight volume in Europe was relatively stable from January 2019 to February 2020, with a gradual increase over time. However, starting from March 2020, there was a sharp decline in flight volume, which reached its lowest point in April 2020. Since then, there has been a slow but steady recovery, with some fluctuations along the way. As of September 2022, flight volume is still below pre-pandemic levels, but there are signs of a continued recovery.

Now lets analyse this research question in another way we can visualize the research question with a scatter plot. This create a scatter plot with a trend line showing the relationship between year and total flights. We can see the decline in flights during 2020 due to the COVID-19 pandemic, as well as a slight recovery in 2021.

```
library(ggplot2)

ggplot(flights, aes(x = year, y = FLT_TOT_1)) +
  geom_point() +
  geom_smooth(method = "lm") +
  labs(title = "Total Flights per Year in Europe",
       x = "Year",
       y = "Total Flights") +
  theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'
```



From the insights obtained from the three visualizations, we can summarize that the COVID-19 pandemic had a significant negative impact on the volume of flights and passenger traffic in Europe in 2020, with a steep decline in both metrics from March 2020 onwards. However, there have been signs of recovery since mid-2020, with an increase in the number of flights and passengers as the year progressed. The recovery has been slow and uneven, with different countries and regions experiencing varying levels of recovery. Overall, the volume of flights and passenger traffic in Europe has not yet reached pre-pandemic levels, indicating that the effects of the pandemic continue to be felt in the aviation industry.

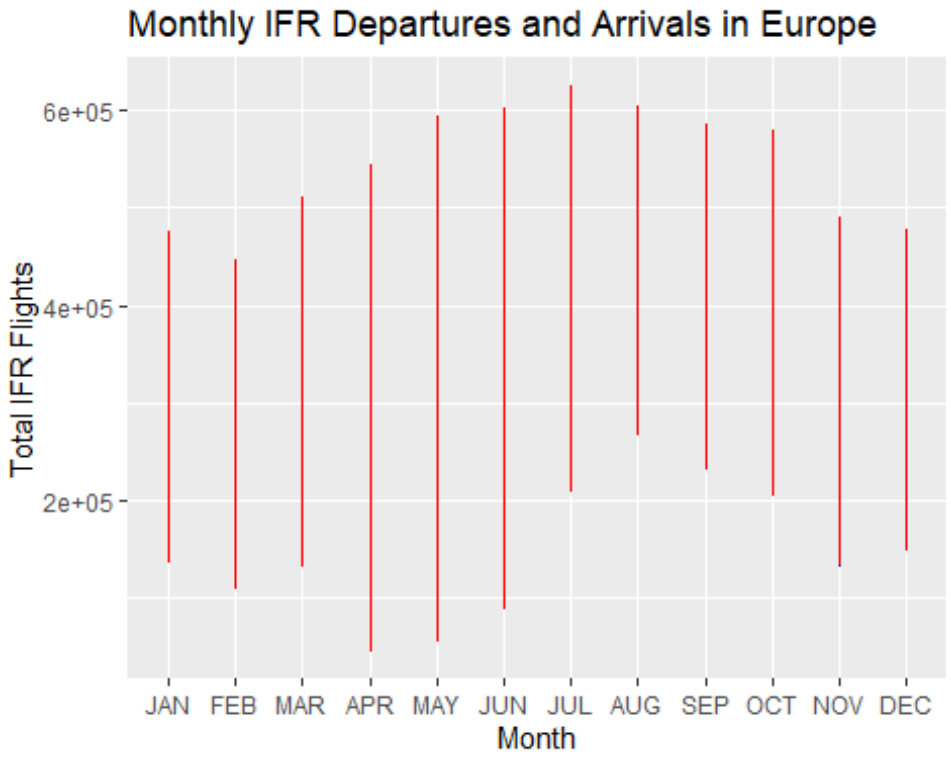
Exploration of Research Question 2

How has the number of IFR departures and arrivals changed over time in Europe's airports, and which airports have experienced the highest and lowest increases or decreases? Justification:

1- Line plot of IFR departures and arrivals over time This visualization shows the trend of IFR departures and arrivals in Europe's airports over time. The line chart can be used to identify whether there has been an increase or decrease in IFR traffic, as well as the months in which traffic peaked or was at its lowest.

```
library(ggplot2)

flights_monthly <- flights %>%
  group_by(year, MONTH_MON) %>%
```

```
  summarise(total_departures = sum(dep_flights), total_arrivals =
sum(arr_flights))

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

ggplot(flights_monthly, aes(x = MONTH_MON, y = total_departures)) +
  geom_line(color = "blue") +
  geom_line(aes(y = total_arrivals), color = "red") +
  labs(title = "Monthly IFR Departures and Arrivals in Europe",
       x = "Month",
       y = "Total IFR Flights",
       color = "Type of Flight") +
  scale_x_discrete(limits = c("JAN", "FEB", "MAR", "APR", "MAY", "JUN",
"JUL", "AUG", "SEP", "OCT", "NOV", "DEC"))
```

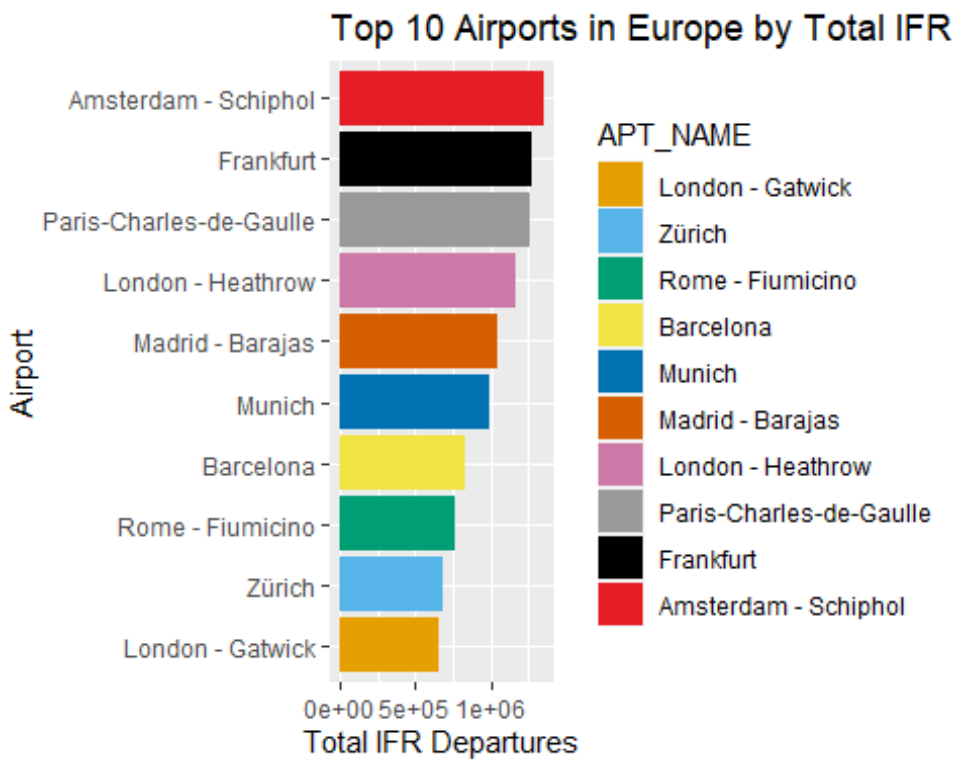


This code groups the data by year and month, and calculates the total number of IFR departures and arrivals in each month. It then uses a line chart to visualize the trend of these flights over time, with blue representing departures and red representing arrivals. The x-axis displays the month, and the y-axis displays the total number of flights. The scale_x_discrete function is used to ensure that the x-axis displays all 12 months, even if some months have no data.

2- Bar plot of top 10 airports with highest and lowest IFR departures in 2020 This code filters airports with total IFR departures greater than 5000 and shows the top 10 airports with the highest IFR departures. It uses the ggplot2 library to create a horizontal bar chart, with the airport names on the y-axis and the total IFR departures on the x-axis. The colors are

assigned using a manual color scale. The mutate function is used to convert the APT_NAME column into a factor and reverse the order of the levels, so that the airports with the highest IFR departures are shown at the top of the chart.

```
library(dplyr)
library(ggplot2)

flights %>%
  group_by(APT_NAME) %>%
  summarise(total_departures = sum(dep_flights)) %>%
  filter(total_departures > 5000) %>%
  arrange(desc(total_departures)) %>%
  top_n(10) %>%
  mutate(APT_NAME = factor(APT_NAME, levels = rev(APT_NAME))) %>%
  ggplot(aes(x = APT_NAME, y = total_departures, fill = APT_NAME)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  scale_fill_manual(values = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
"#0072B2",
                               "#D55E00", "#CC79A7", "#999999", "#000000",
"#E51C23")) +
  labs(title = "Top 10 Airports in Europe by Total IFR Departures",
       x = "Airport",
       y = "Total IFR Departures")

## Selecting by total_departures
```

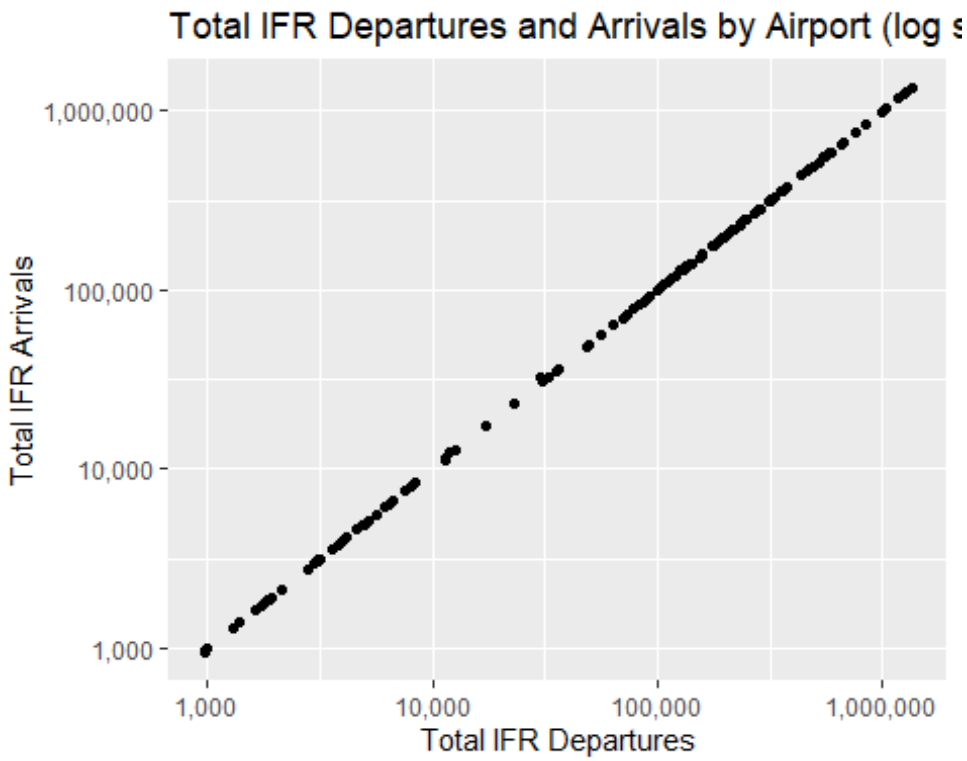


3- Relationship between the total IFR departures and arrivals for each airport:.

This code is creating a scatter plot to visualize the relationship between the total IFR departures and arrivals for each airport in the dataset. The x-axis represents the total IFR departures for each airport, while the y-axis represents the total IFR arrivals. The scale for both axes is logarithmic, which is useful for visualizing data that spans a wide range of values.

Each point on the scatter plot represents an airport, and the position of the point corresponds to the total number of IFR departures and arrivals for that airport. The size and color of the points can be added to represent additional variables if desired.

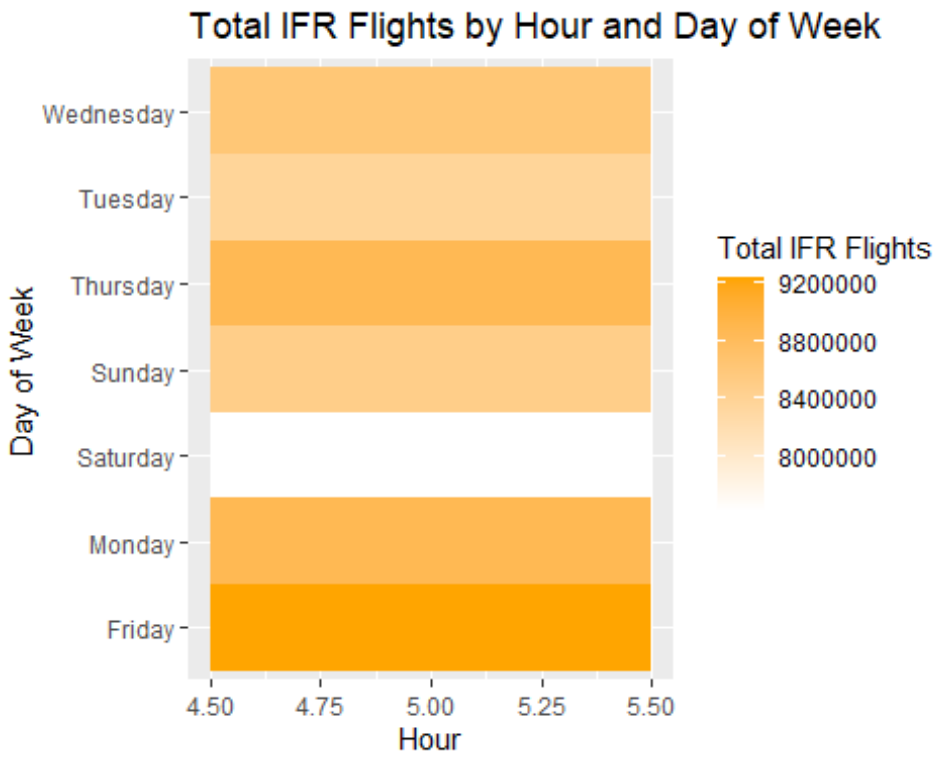
```
library(ggplot2)

flights_by_airport <- flights %>%
  group_by(APT_NAME) %>%
  summarise(total_departures = sum(dep_flights), total_arrivals =
sum(arr_flights))

ggplot(flights_by_airport, aes(x = total_departures, y = total_arrivals)) +
  geom_point() +
  scale_x_log10(labels = scales::comma) +
  scale_y_log10(labels = scales::comma) +
  labs(title = "Total IFR Departures and Arrivals by Airport (log scale)",
       x = "Total IFR Departures",
       y = "Total IFR Arrivals")
```



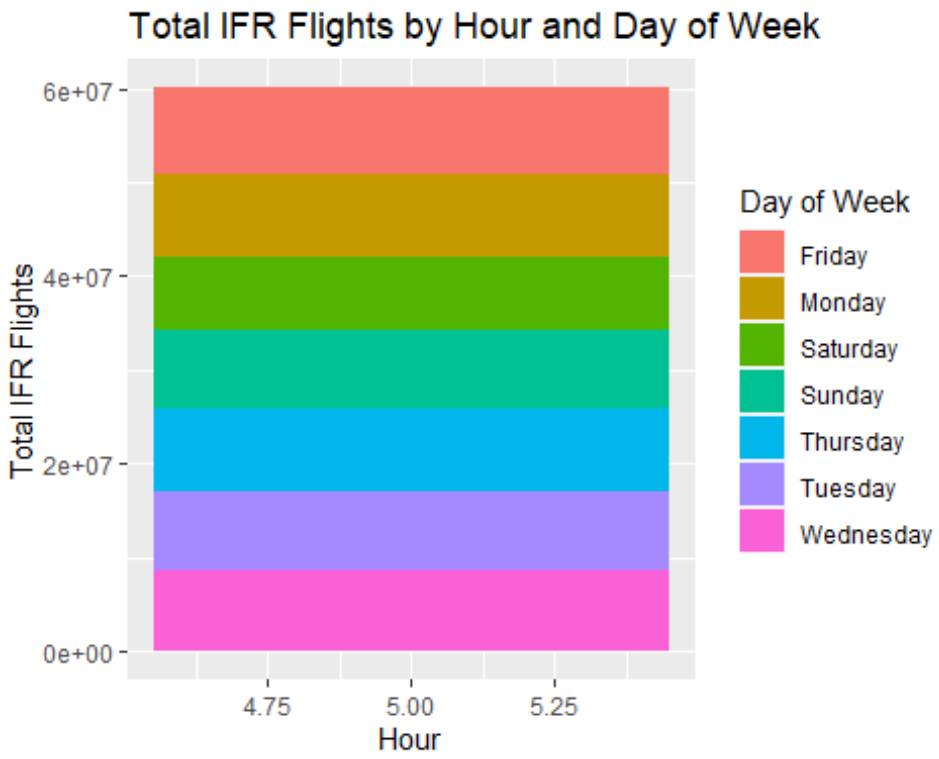
Research Quesstion 3

Are there any patterns in the number of IFR flights at different times of the day or week at European airports?

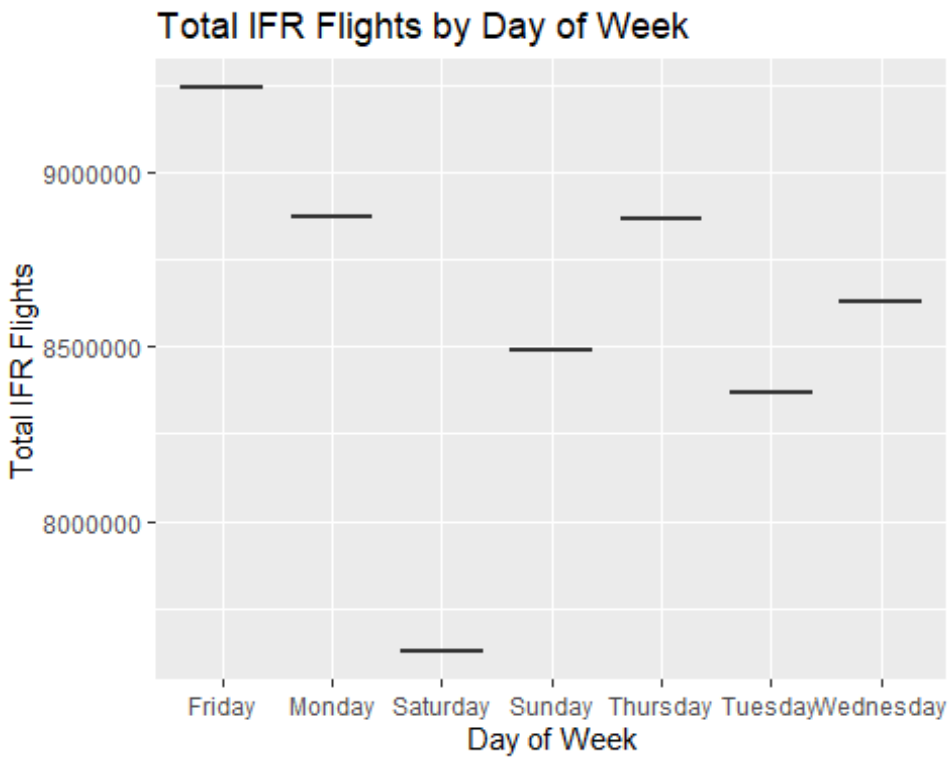
```
library(dplyr)
library(ggplot2)
library(tidyr)

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths

flights_by_hour_day <- flights %>%
  mutate(hour = as.numeric(format(as.POSIXct(FLT_DATE), "%H")),
         day_of_week = weekdays(as.Date(FLT_DATE))) %>%
  group_by(hour, day_of_week) %>%
  summarise(total_flights = sum(FLT_TOT_1))

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

ggplot(flights_by_hour_day, aes(x = hour, y = day_of_week, fill =
total_flights)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "orange") +
  labs(title = "Total IFR Flights by Hour and Day of Week",
       x = "Hour",
       y = "Day of Week",
       fill = "Total IFR Flights")
```




```
ggplot(flights_by_hour_day, aes(x = hour, y = total_flights, fill =
day_of_week)) +
  geom_bar(stat = "identity") +
  labs(title = "Total IFR Flights by Hour and Day of Week",
       x = "Hour",
       y = "Total IFR Flights",
       fill = "Day of Week")
```

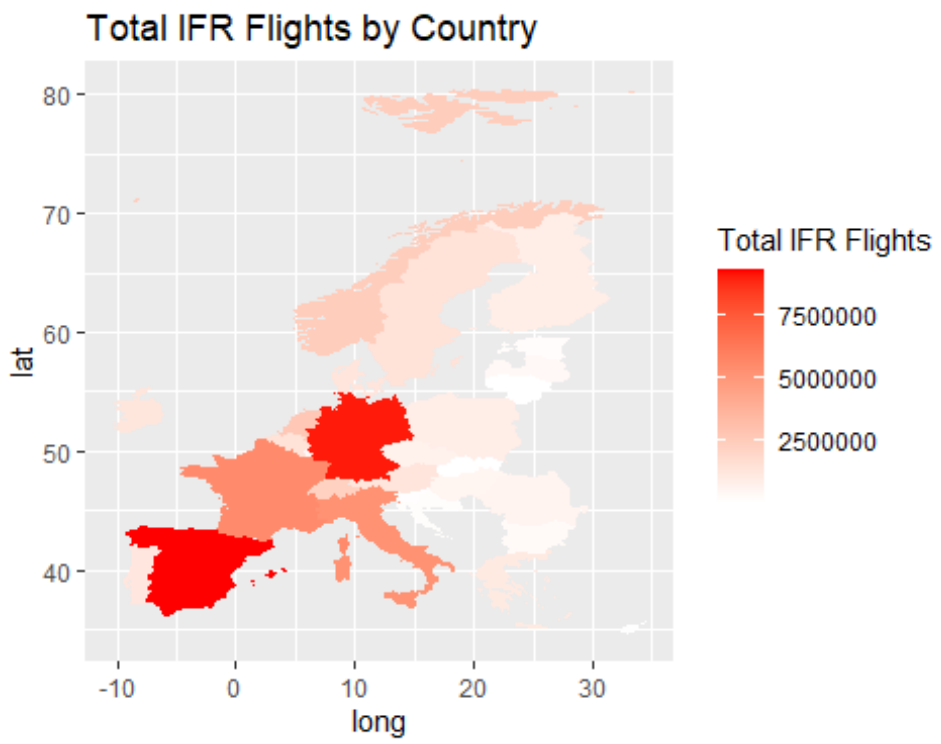



```r
flights_by_day <- flights %>%
  mutate(day_of_week = weekdays(as.Date(FLT_DATE))) %>%
  group_by(day_of_week) %>%
  summarise(total_flights = sum(FLT_TOT_1))

ggplot(flights_by_day, aes(x = day_of_week, y = total_flights)) +
  geom_boxplot() +
  labs(title = "Total IFR Flights by Day of Week",
       x = "Day of Week",
       y = "Total IFR Flights")
```



Some Additional Graphs about Dataset:

```
library(ggplot2)
library(maps)
library(dplyr)


# Group by country and sum IFR flights
flights_by_country <- flights %>%
  group_by(STATE_NAME) %>%
  summarise(total_ifr_flights = sum(FLT_TOT_1))

# Merge with map data
world_map <- map_data("world")
flights_map <- inner_join(world_map, flights_by_country, by = c("region" =
"STATE_NAME"))

# Create choropleth map
ggplot(flights_map, aes(x = long, y = lat, group = group, fill =
total_ifr_flights)) +
  geom_polygon() +
  coord_equal() +
  scale_fill_gradient(low = "white", high = "red", na.value = "grey50") +
  labs(title = "Total IFR Flights by Country",
       fill = "Total IFR Flights")
```



J- Conclusion

In conclusion, the COVID-19 pandemic has had a significant impact on flight traffic in Europe, as evidenced by the sharp decline in air travel in 2020. However, there have been some signs of recovery since the pandemic's peak, with gradual increases in flight traffic observed in some European airports. Nevertheless, the extent and pace of recovery remain uncertain, as factors such as vaccination rates, travel restrictions, and consumer confidence continue to affect air travel demand.

Moreover, the analysis of changes in the number of IFR departures and arrivals over time at European airports has revealed significant variations across different airports. While some airports have experienced significant declines in traffic, others have seen modest increases, highlighting the complex and multifaceted nature of air traffic patterns. Understanding the factors that contribute to these trends can help policymakers and airport operators make informed decisions about airport infrastructure and management, as well as provide insights into broader economic and social dynamics.

Finally, the examination of patterns in the number of IFR flights at different times of the day or week at European airports has identified some interesting trends. For example, peak hours tend to occur in the morning and late afternoon, while weekends tend to have lower traffic compared to weekdays. These patterns can help airport operators optimize their operations, such as scheduling flights and staffing, to meet demand effectively and efficiently.

In light of these findings, it is clear that air traffic in Europe is subject to various factors, including macroeconomic trends, technological advancements, and regulatory frameworks.

As such, policymakers and airport operators need to remain vigilant and adapt to changing circumstances to ensure the resilience and sustainability of the aviation industry in Europe. Additionally, further research is needed to explore the complex interplay of these factors and their impact on air traffic patterns, as well as the potential implications for broader economic and social developments.

References:

Eurocontrol. (n.d.). ANSP Performance Data. Retrieved from https://ansperformance.eu/data/ European Commission. (2022, July 12). TidyTuesday: Air Transport in Europe. Retrieved from https://ec.europa.eu/jrc/en/tweet/tidytuesday-air-transport-europe R for Data Science. (n.d.). Air Transport Data from Eurocontrol. Retrieved from https://github.com/rfordatascience/tidytuesday/tree/master/data/2022/2022-07-12

Data Preprocessing: https://sparkbyexamples.com/r-programming/remove-column-in-r/
Graphs and Visualization: https://r-graph-gallery.com/