Data Wrangling (Data Preprocessing)

Mid-term assessment

SNIGDHA MATHUR - S4017572

Introduction

The aim of this report is to create authentic, practical data that reflects the everyday data usage of a global brand network. We will focus our detailed analysis on Qantas Airlines—short for Queensland and Northern Territory Aerial Services—which is a prominent Australian carrier known for its extensive fleet, international routes, and reach within Australia and Oceania.

To facilitate our analysis, we will produce synthetic datasets for Qantas Airlines. This will enable us to derive and present valuable insights based on the results obtained.

Synthetic data generation is a process of creating data having the characterstics of the real life data. The synthetically generated datasets further help in scenario testing, development, and understand the trends without hampering the actual data points.

The datasets generated are for the analysis are :

Airline Data: This dataset includes key elements such as flight numbers, types of aircraft and departure schedules among others. This information is crucial for simulating the logistical and operational aspects of the airline.
Passenger Data: This dataset provides information on passengers traveling with Qantas Airlines, detailing demograohics, ticket class, and other relevant passenger-specific information.
Customer Feedback: This dataset gathers insights from passengers about their experiences with Qantas Airlines, which is essential for understanding customer satisfaction and areas of improvement.

Setup

```r
# Load the necessary packages required to reproduce the report

library(dplyr)
library(magrittr)
library(tidyr)
library(knitr)
```

Importing the necessary libraries for data manipulation and analysis. Libraries dpylr, magrittr and tidyr are all part of library - Tidyverse package. These are used for data manipulation , making the data more readble and extracting insights.
The library kinttr is used in R markdown file for creating a RMD file.

Data generation

For generating synthetic dataset, we to specify a seed value for ensuring the reproducubility of the generated data. The seed value helps in maintaining the consistency of the data throughout.

```r
# Setting a seed value
SEED <- 367
set.seed(SEED)
```

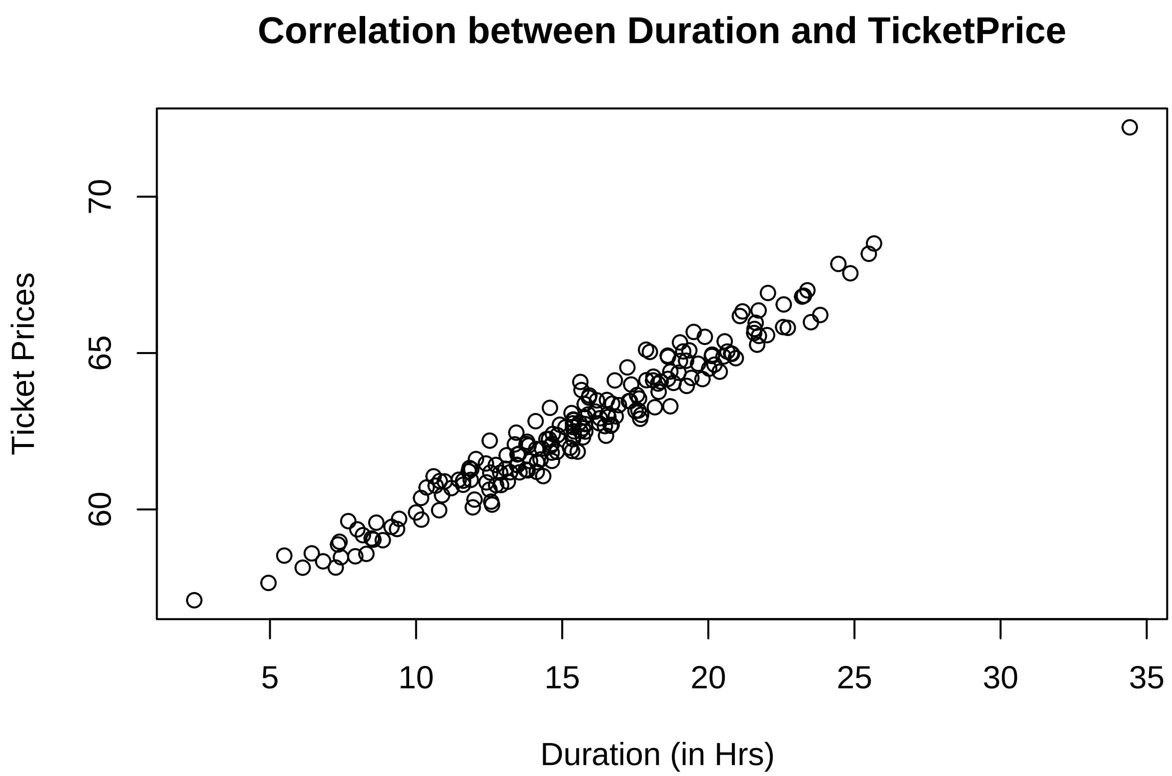
Seed value of 367 has been assigned. We now generate the datasets. The datasets will have variables of character and numeric data types ensuring that the data represents and replicates the original airline data.

AIRLINE DATSET

```r
# Dataset 1 Generation

airline_data <- data.frame(
  FlightNumber = sprintf("QF%d", sample(100:999, 200, replace = TRUE)),
  Destination = sample(c("Sydney", "Melbourne", "Brisbane", "Perth", "Adelaide", "Canberra", "Hobart", 
  DepartureDate = sample(seq(as.Date('2020-01-01'), as.Date('2023-12-31'), by="day"),
                         200, replace = TRUE),
  Duration = round(rnorm(200, mean = 15, sd = 5), digits = 2),
  Capacity = round(rnorm(200, 60, 10)),
  AircraftType = sample(c("Boeing 737", "Airbus A320", "Boeing 787", "Airbus A330", NA), 200, replace =
 )


airline_data %<>%
  mutate(error = rnorm(n = 200, mean = 10, sd = 0.5))
airline_data %<>%
  mutate(TicketPrice = ((Duration * 0.5) + 45 + error))


print(head(airline_data))
```

```
##   FlightNumber Destination DepartureDate Duration Capacity AircraftType
## 1        QF834       Perth    2021-04-29    20.13       73   Boeing 737
## 2        QF727    Brisbane    2023-10-31    12.39       51   Boeing 787
## 3        QF736    Canberra    2020-01-25     7.25       52   Boeing 737
## 4        QF411   Melbourne    2021-03-01    18.12       74         <NA>
## 5        QF995      Hobart    2023-01-22    12.52       50  Airbus A320
## 6        QF440    Brisbane    2023-03-07    16.71       63  Airbus A320
##       error TicketPrice
## 1  9.884228    64.94923
## 2 10.273768    61.46877
## 3  9.515526    58.14053
## 4 10.183942    64.24394
## 5 10.938913    62.19891
## 6 10.025667    63.38067
```

```r
plot(x = airline_data$Duration,
     y = airline_data$TicketPrice,
     main = "Correlation between Duration and TicketPrice",
     xlab = "Duration (in Hrs) ",
     ylab = "Ticket Prices")
```



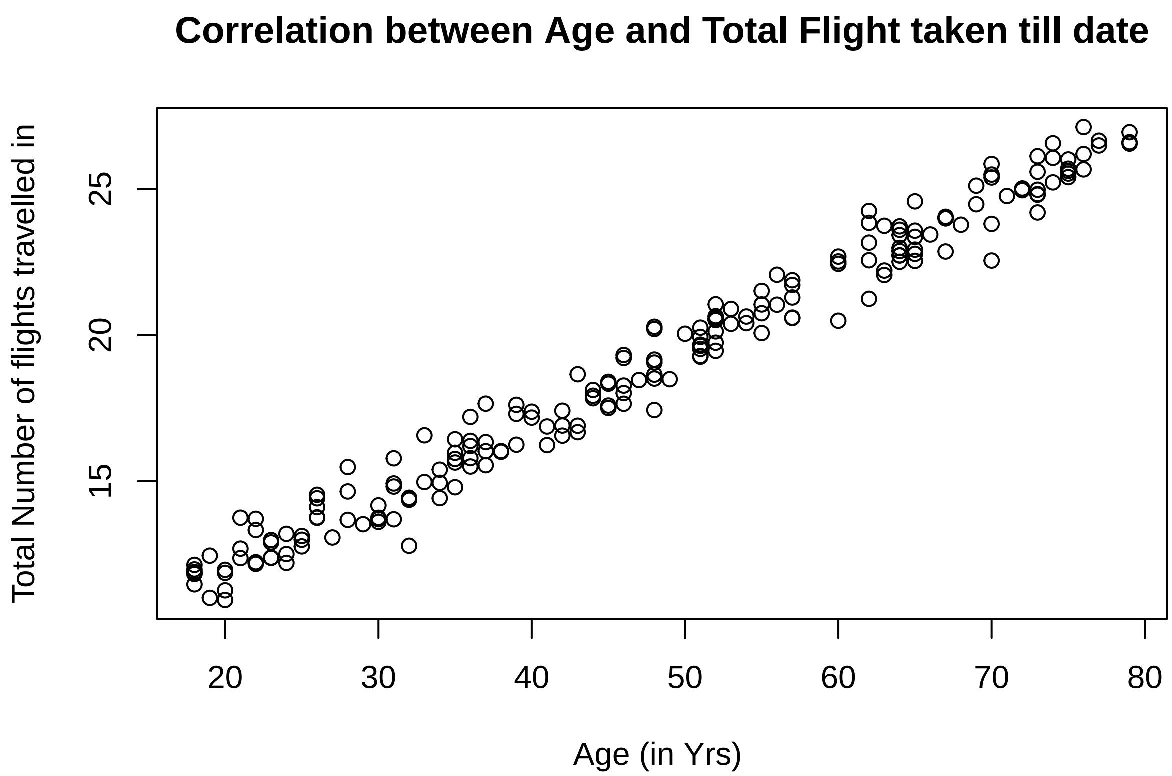
The airline dataset comprises 100 records, each detailing aspects of individual flights through 7 key variables: FlightNumber, Destination, DepartureDate, Capacity, Duration, AircraftType, and TicketPrice. TicketPrice is the variable here, influenced by Duration—the time span of the flight—based on the premise that longer flights generally result in higher costs and, consequently, pricier tickets, making Duration and Ticketprice correlate to eachother.

To dissect this relationship, an additional variable, Error, is introduced to fine-tune the correlation analysis between TicketPrice and Duration. The aim is to quantify and confirm the expected positive correlation where extended flight times correlate with increased ticket prices, reflecting the interplay between operational costs and pricing strategies. The outcome of this analysis will provide a clearer understanding of how flight durations impact ticket pricing, potentially guiding operational and strategic decision-making.

PASSENGER DATASET

```
# Dataset 2 Generation


passenger_data <- data.frame(
  PassengerID = sprintf("%d", sample(100:999, 200, replace = TRUE)),
  Age = sample(18:80, 200, replace = TRUE),
  Gender = sample(c("Male", "Female", "Other"), 100, replace = TRUE,
                  prob = c(0.49, 0.49, 0.02)),
  Nationality = sample(c("Australian", "American", "British", "Chinese", "Indian"),
                       200, replace = TRUE),
  FlightNumber = sprintf("QF%d", sample(100:999, 200, replace = TRUE)),
  TicketClass = sample(c("Economy", "Premium Economy", "Business", "First Class", NA),
                       200, replace = TRUE, prob = c(0.4, 0.3, 0.195, 0.1, 0.005))
```

```
)


passenger_data %<>%
  mutate(error2 = rnorm(n = 200, mean = 7, sd = 0.7))
passenger_data %<>%
  mutate(TotalFlightsTaken = ((Age * 0.5)  / 2 + error2))


plot(x = passenger_data$Age,
     y = passenger_data$TotalFlightsTaken,
     main = "Correlation between Age and Total Flight taken till date",
     xlab = "Age (in Yrs) ",
     ylab = "Total Number of flights travelled in")
```




```
print(head(passenger_data))
```

```
##   PassengerID Age Gender Nationality FlightNumber     TicketClass   error2
## 1         244  26   Male     Chinese        QF854 Premium Economy 7.615124
## 2         165  65 Female    American        QF498 Premium Economy 6.290985
## 3         891  62 Female    American        QF404         Economy 7.664920
## 4         703  62   Male     Chinese        QF400         Economy 8.341960
## 5         429  26   Male    American        QF957         Economy 7.916763
## 6         457  46   Male     British        QF675 Premium Economy 6.774106
##   TotalFlightsTaken
## 1          14.11512
## 2          22.54098
## 3          23.16492
## 4          23.84196
## 5          14.41676
```

```
## 6           18.27411
```

The passenger dataset comprises 100 records, each containing seven specific attributes: PassengerID, Age, Gender, Nationality, Flight Number, Ticket Class, and the Number of Flights a passenger has taken to date. The variable 'error2' measures the relationship between a passenger's age and the total number of flights they have taken. The data suggests that younger individuals tend to have flown on fewer flights compared to older individuals, indicating a direct, positive relationship between age and flight frequency. This correlation is evidenced by a graph plotting 'Age' against 'Total Flights Taken,' affirming the observed trend between these two variables.

CUSTOMER FEEDBACK DATASET

```r
# Dataset 3 Generation

customer_feedback <- data.frame(
  FeedbackID = 1:200,
  PassengerID = sprintf("%d", sample(100:999, 200, replace = TRUE)),
  FeedbackDate = sample(seq(as.Date('2020-01-01'), as.Date('2023-12-31'), by="day"),
                        200, replace = TRUE),
  Category = sample(c("Service", "Cleanliness", "In-flight Entertainment", "Food Quality", "Comfort", "
  Rating = round(rnorm(200, mean = 4.3, sd = 0.6), digits = 1) ,
  Comments = sample(c("Very satisfied", "Satisfied", "Neutral", "Dissatisfied", "Very dissatisfied", NA
)

# Introduce outliers in Ratings
customer_feedback$Rating[sample(1:200, 5)] <- sample(6:10, 5, replace = TRUE)

# Print the head of the dataset to see the first few entries
print(head(customer_feedback))
```

```
##   FeedbackID PassengerID FeedbackDate     Category Rating          Comments
## 1          1         783   2022-12-16  Punctuality    5.1              <NA>
## 2          2         183   2020-11-05         <NA>    3.6           Neutral
## 3          3         432   2021-07-11 Food Quality    3.7              <NA>
## 4          4         312   2023-04-11      Comfort    3.8 Very dissatisfied
## 5          5         867   2020-04-13         <NA>    4.6 Very dissatisfied
## 6          6         494   2022-01-09      Comfort    4.8              <NA>
```

The dataset containing customer feedback consists primarily of six variables: Feedback ID, Passenger ID, Feedback Date, Category, Rating, and Comments. To facilitate the easy creation of data, we have employed a function named generate_feedback_text. This function is designed to randomly produce comments that fall into one of the following categories: "Very satisfied," "Satisfied," "Neutral," "Dissatisfied," or "Very dissatisfied."

The airline dataset and passenger dataset have the variable Flight number common as Flight number helps tracking the aircraft and the corresponsing passengers in the flight. Similiarly passenger dataset and customer feedback have the variable passenger ID in common for tracking the passenger travel and travel experience feedback.

Merging data sets

Since airline and passenger dataset have the variable "FlightNumber" , we merge the datset in Flight Number variable. Similarly the passenger and customer feedback datasets will be merged by Passenger ID variable.

Merging of dataset can be performed using varipus ways: join() and merge()
1. Join() - The join function has 4 types of joins that can be performed on the dataset. These are : inner join, full join, right join and left join.
Inner join - Retains only the common rows from the datasets Full join - Keeps all rows from the dataset Right join - Joins rows matching to the right side of the dataset and stores Left join - Joins rows matching to the left side of the dataset and stores .

```
# Merging datasets

Flight_details <- inner_join(airline_data, passenger_data, by = "FlightNumber")
```

```
## Warning in inner_join(airline_data, passenger_data, by = "FlightNumber"): Detected an unexpected many
## i Row 16 of `x` matches multiple rows in `y`.
## i Row 116 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
Customer_details <- inner_join(passenger_data, customer_feedback, by="PassengerID" )

head(Flight_details)
```

```
##   FlightNumber Destination DepartureDate Duration Capacity AircraftType
## 1        QF834       Perth    2021-04-29    20.13       73   Boeing 737
## 2        QF736    Canberra    2020-01-25     7.25       52   Boeing 737
## 3        QF995      Hobart    2023-01-22    12.52       50  Airbus A320
## 4        QF440    Brisbane    2023-03-07    16.71       63  Airbus A320
## 5        QF403      Hobart    2020-12-18    17.88       62   Boeing 787
## 6        QF403      Hobart    2020-12-18    17.88       62   Boeing 787
##       error TicketPrice PassengerID Age Gender Nationality     TicketClass
## 1  9.884228    64.94923         983  20 Female      Indian Premium Economy
## 2  9.515526    58.14053         676  79 Female     Chinese         Economy
## 3 10.938913    62.19891         822  37 Female    American         Economy
## 4 10.025667    63.38067         586  48 Female      Indian     First Class
## 5 10.192108    64.13211         364  42 Female     British Premium Economy
## 6 10.192108    64.13211         251  26   Male     British        Business
##     error2 TotalFlightsTaken
## 1 5.937895          10.93789
## 2 7.193425          26.94343
## 3 6.778766          16.02877
## 4 5.437416          17.43742
## 5 6.060474          16.56047
## 6 8.037749          14.53775
```

```
head(Customer_details)
```

```
##   PassengerID Age Gender Nationality FlightNumber     TicketClass   error2
## 1         703  62   Male     Chinese        QF400         Economy 8.341960
## 2         703  62   Male     Chinese        QF400         Economy 8.341960
## 3         999  60   Male      Indian        QF549         Economy 7.439736
## 4         313  23 Female  Australian        QF151 Premium Economy 7.237310
## 5         554  51 Female     Chinese        QF829 Premium Economy 7.506297
## 6         710  79   Male    American        QF823         Economy 6.803056
##   TotalFlightsTaken FeedbackID FeedbackDate            Category Rating
## 1          23.84196         37   2020-02-13         Punctuality    4.2
## 2          23.84196        200   2021-09-09        Food Quality    4.8
```

```
## 3          22.43974        115   2020-08-07                 Comfort    3.6
## 4          12.98731        187   2023-03-27 In-flight Entertainment    4.3
## 5          20.25630        137   2021-05-19             Punctuality    4.1
## 6          26.55306         51   2023-04-02             Punctuality    4.4
##            Comments
## 1    Very satisfied
## 2           Neutral
## 3 Very dissatisfied
## 4 Very dissatisfied
## 5              <NA>
## 6           Neutral
```

We have used inner_join as the merged datasets should pny contains the values that are common to the initial datasets. Using full_join will not be efficient as it will be unable to lead to desired outputs.

The merged datasets are stored in new dataframe namely : Flight Details and Customer details.

The flight details dataset contains 16 observations and 12 variables, referring to the instances where the airline data and passenger datasets shared common flight numbers. This dataset thus represents 16 distinct passengers who traveled with Qantas Airline to their respective destinations. This alignment of flight numbers across datasets indicates that the recorded details pertain specifically to these passengers.

The customer details dataset comprises of 13 observations, indicating that there were 13 passengers who traveled with Qantas Airlines and later provided feedback. Each of these passengers utilized the same passenger ID for their feedback submissions. This dataset serves as a full representation of passenger interactions and their travel experiences with the airline.

Checking structure of combined data

```
# Checking the structure of combined dataset

str(Flight_details)
```

```
## 'data.frame':    46 obs. of  15 variables:
##  $ FlightNumber     : chr  "QF834" "QF736" "QF995" "QF440" ...
##  $ Destination      : chr  "Perth" "Canberra" "Hobart" "Brisbane" ...
##  $ DepartureDate    : Date, format: "2021-04-29" "2020-01-25" ...
##  $ Duration         : num  20.13 7.25 12.52 16.71 17.88 ...
##  $ Capacity         : num  73 52 50 63 62 62 68 76 76 62 ...
##  $ AircraftType     : chr  "Boeing 737" "Boeing 737" "Airbus A320" "Airbus A320" ...
##  $ error            : num  9.88 9.52 10.94 10.03 10.19 ...
##  $ TicketPrice      : num  64.9 58.1 62.2 63.4 64.1 ...
##  $ PassengerID      : chr  "983" "676" "822" "586" ...
##  $ Age              : int  20 79 37 48 42 26 65 31 69 23 ...
##  $ Gender           : chr  "Female" "Female" "Female" "Female" ...
##  $ Nationality      : chr  "Indian" "Chinese" "American" "Indian" ...
##  $ TicketClass      : chr  "Premium Economy" "Economy" "Economy" "First Class" ...
##  $ error2           : num  5.94 7.19 6.78 5.44 6.06 ...
##  $ TotalFlightsTaken: num  10.9 26.9 16 17.4 16.6 ...
```

```
str (Customer_details)
```

```
## 'data.frame':    37 obs. of  13 variables:
```

```
##  $ PassengerID      : chr  "703" "703" "999" "313" ...
##  $ Age              : int  62 62 60 23 51 79 27 73 72 52 ...
##  $ Gender           : chr  "Male" "Male" "Male" "Female" ...
##  $ Nationality      : chr  "Chinese" "Chinese" "Indian" "Australian" ...
##  $ FlightNumber     : chr  "QF400" "QF400" "QF549" "QF151" ...
##  $ TicketClass      : chr  "Economy" "Economy" "Economy" "Premium Economy" ...
##  $ error2           : num  8.34 8.34 7.44 7.24 7.51 ...
##  $ TotalFlightsTaken: num  23.8 23.8 22.4 13 20.3 ...
##  $ FeedbackID       : int  37 200 115 187 137 51 84 96 110 23 ...
##  $ FeedbackDate     : Date, format: "2020-02-13" "2021-09-09" ...
##  $ Category         : chr  "Punctuality" "Food Quality" "Comfort" "In-flight Entertainment" ...
##  $ Rating           : num  4.2 4.8 3.6 4.3 4.1 4.4 4.7 6.1 3.6 4.8 ...
##  $ Comments         : chr  "Very satisfied" "Neutral" "Very dissatisfied" "Very dissatisfied" ...
```

The structure of the combined datsets : Flight Details and Customer Details comprise of some data type mismatch and require data type conversion.

Flight Details
1. PassengerID variables needs to be converted to numeric from character 2. Ticket Class should be ordered factor as it contains categorical values with ranks.
Customer Details
1. Similiar to Flight details dataset, the variable PassengerID needs to be converted to Numeric values 2. The variable Comments and TicketClass should be ordered factor variables due to presence of categorical values with possible ranks.

```
# Data type conversions

Flight_details$PassengerID <- as.numeric(Flight_details$PassengerID)
Flight_details$TicketClass <- factor(Flight_details$TicketClass,
                                     levels = c("Economy", "Premium Economy", "Business", "First Class")


Customer_details$PassengerID <- as.numeric(Customer_details$PassengerID)
Customer_details$Comments <- factor(Customer_details$Comments,
                                    levels = c("Very satisfied", "Satisfied", "Neutral", "Dissatisfied"
Customer_details$TicketClass <- factor(Customer_details$TicketClass,
                                       levels = c("Economy", "Premium Economy", "Business", "First Class

str(Flight_details)
```

```
## 'data.frame':    46 obs. of  15 variables:
##  $ FlightNumber     : chr  "QF834" "QF736" "QF995" "QF440" ...
##  $ Destination      : chr  "Perth" "Canberra" "Hobart" "Brisbane" ...
##  $ DepartureDate    : Date, format: "2021-04-29" "2020-01-25" ...
##  $ Duration         : num  20.13 7.25 12.52 16.71 17.88 ...
##  $ Capacity         : num  73 52 50 63 62 62 68 76 76 62 ...
##  $ AircraftType     : chr  "Boeing 737" "Boeing 737" "Airbus A320" "Airbus A320" ...
##  $ error            : num  9.88 9.52 10.94 10.03 10.19 ...
##  $ TicketPrice      : num  64.9 58.1 62.2 63.4 64.1 ...
##  $ PassengerID      : num  983 676 822 586 364 251 252 840 317 236 ...
##  $ Age              : int  20 79 37 48 42 26 65 31 69 23 ...
##  $ Gender           : chr  "Female" "Female" "Female" "Female" ...
##  $ Nationality      : chr  "Indian" "Chinese" "American" "Indian" ...
##  $ TicketClass      : Ord.factor w/ 4 levels "Economy"<"Premium Economy"<..: 2 1 1 4 2 3 1 4 2 1 ...
##  $ error2           : num  5.94 7.19 6.78 5.44 6.06 ...
```

```
##  $ TotalFlightsTaken: num  10.9 26.9 16 17.4 16.6 ...
```

```r
str (Customer_details)
```

```
## 'data.frame':    37 obs. of  13 variables:
##  $ PassengerID      : num  703 703 999 313 554 710 508 750 257 187 ...
##  $ Age              : int  62 62 60 23 51 79 27 73 72 52 ...
##  $ Gender           : chr  "Male" "Male" "Male" "Female" ...
##  $ Nationality      : chr  "Chinese" "Chinese" "Indian" "Australian" ...
##  $ FlightNumber     : chr  "QF400" "QF400" "QF549" "QF151" ...
##  $ TicketClass      : Ord.factor w/ 4 levels "Economy"<"Premium Economy"<..: 1 1 1 2 2 1 2 1 1 1 ...
##  $ error2           : num  8.34 8.34 7.44 7.24 7.51 ...
##  $ TotalFlightsTaken: num  23.8 23.8 22.4 13 20.3 ...
##  $ FeedbackID       : int  37 200 115 187 137 51 84 96 110 23 ...
##  $ FeedbackDate     : Date, format: "2020-02-13" "2021-09-09" ...
##  $ Category         : chr  "Punctuality" "Food Quality" "Comfort" "In-flight Entertainment" ...
##  $ Rating           : num  4.2 4.8 3.6 4.3 4.1 4.4 4.7 6.1 3.6 4.8 ...
##  $ Comments         : Ord.factor w/ 5 levels "Very satisfied"<..: 1 3 5 5 NA 3 1 4 3 1 ...
```

Both the datasets now have appropriate varaible data types being a mix of character, numeric and ordered factor. .

For generating the summary statistics of dataset, we performed the following steps: 1. Identify a categorical variabel for grouping and a numeric variables for calculating summary statistics
2.Group the dataset by categircal variable using groupby()
3. Calculate descriptive values like mean, median, standard deviation, variance and quartiles.

For flight details datset, we shall group the data by TicketClass and for Customer Details, the grouping will be done on Comments variable.

Generate summary statistics

```r
# Generate summary statistics

summary_stats1 <- Flight_details %>%
  group_by(TicketClass) %>%
  summarise(
    Mean_Age = mean(Age, na.rm = TRUE),
    Median_Age = median(Age, na.rm = TRUE),
    Q1_Age = quantile(Age, 0.25, na.rm = TRUE),
    Q3_Age = quantile(Age, 0.75, na.rm = TRUE),
    SD_Age = sd(Age, na.rm = TRUE),
    .groups = 'drop'
  )

# Print the summary statistics
print(summary_stats1)
```

```
## # A tibble: 4 x 6
##   TicketClass     Mean_Age Median_Age Q1_Age Q3_Age SD_Age
##   <ord>              <dbl>      <int>  <dbl>  <dbl>  <dbl>
## 1 Economy             55.6         62   46     70     17.9
## 2 Premium Economy     42.5         34   25.5   61.5   22.3
## 3 Business            42.3         43   28     55     15.9
## 4 First Class         58.6         64   48     75     19.0
```

```
summary_stats2 <- Customer_details %>%
  group_by(Comments) %>%
  summarise(
    Mean_Rating = mean(Rating, na.rm = TRUE),
    Median_Rating = median(Rating, na.rm = TRUE),
    Q1_Rating = quantile(Rating, 0.25, na.rm = TRUE),
    Q3_Rating = quantile(Rating, 0.75, na.rm = TRUE),
    SD_Rating = sd(Rating, na.rm = TRUE),
    .groups = 'drop'
  )

summary_stats2
```

```
## # A tibble: 6 x 6
##   Comments          Mean_Rating Median_Rating Q1_Rating Q3_Rating SD_Rating
##   <ord>                   <dbl>         <dbl>     <dbl>     <dbl>     <dbl>
## 1 Very satisfied           4.5           4.5       4.27      4.73     0.288
## 2 Satisfied                4.8           5.1       4.65      5.1      0.520
## 3 Neutral                  4.59          4.35      3.87      4.7      1.28
## 4 Dissatisfied             4.72          4.3       4.3       5        0.867
## 5 Very dissatisfied        4.14          4.15      3.82      4.32     0.845
## 6 <NA>                     4.13          4.1       3.95      4.3      0.351
```

The summary statistics for Flight details tell us that : 1.Economy: The mean age of passengers in Economy class is 49 years, with a median age slightly lower at 47.5 years, indicating a younger age profile overall.
2. Premium Economy: This class has a higher mean age of 59.25 years and a much higher median age of 68.5 years, indicating a significant skew towards older passengers.
3. Business: Similar to Premium Economy, Business class shows a high mean age of 58.50 years and an even higher median age of 63 years.
4. First Class: Passengers in First Class have both a mean and median age of 58 years, indicating a symmetric age distribution centered around middle-aged passengers.

The summary statistics for Customer details dataset state that:
1. Satisfied: Shows a very narrow range in ratings (Q1 at 3.75 and Q3 at 4.05) with both the mean and median tightly clustered at 3.90, indicating a high level of consistency in satisfaction.
2. Neutral: Displays a significantly higher mean and median rating at 5.95, but with a much larger spread between Q1 (4.925) and Q3 (6.975)
3. Dissatisfied: Presents a narrow spread in ratings (Q1 at 4.3 and Q3 at 4.5), with the mean and median also tightly clustered at 4.4.
4. Very Dissatisfied: All values : median, Q1, and Q3—are exactly 4.6, with an SD marked as NA

Scanning data

Scanning missing values in a dataset is a crucial step in data preprocessing, especially before conducting any form of data analysis or modeling. It helps in providing accurate analysis and better decision making using the data.

```
# Scan variables for missing values

# function for missing value summary
summary_missing <- function(data) {
  missing_summary <- data %>%
    summarise(across(everything(), ~ sum(is.na(.))))
```

```r
  return(missing_summary)
}


get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}


summary_missing(Flight_details)
```

```
##   FlightNumber Destination DepartureDate Duration Capacity AircraftType error
## 1            0           6             0        0        0            6     0
##   TicketPrice PassengerID Age Gender Nationality TicketClass error2
## 1           0           0   0      0           0           0      0
##   TotalFlightsTaken
## 1                 0
```

```r
Flight_details <- Flight_details %>%
  mutate(
    Destination = replace_na(Destination, get_mode(Flight_details$Destination[!is.na(Flight_details$Des
    AircraftType = replace_na(AircraftType,
get_mode(Flight_details$AircraftType[!is.na(Flight_details$AircraftType)]))
)

summary_missing(Flight_details)
```

```
##   FlightNumber Destination DepartureDate Duration Capacity AircraftType error
## 1            0           0             0        0        0            0     0
##   TicketPrice PassengerID Age Gender Nationality TicketClass error2
## 1           0           0   0      0           0           0      0
##   TotalFlightsTaken
## 1                 0
```

```r
summary_missing(Customer_details)
```

```
##   PassengerID Age Gender Nationality FlightNumber TicketClass error2
## 1           0   0      0           0            0           0      0
##   TotalFlightsTaken FeedbackID FeedbackDate Category Rating Comments
## 1                 0          0            0        3      0        3
```

```r
Customer_details <- Customer_details %>%
  mutate(
    Category = replace_na(Category, get_mode(Customer_details$Category[!is.na(Customer_details$Category)
    Comments = replace_na(Comments, get_mode(Customer_details$Comments[!is.na(Customer_details$Comments)
  )

summary_missing(Customer_details)
```

```
##   PassengerID Age Gender Nationality FlightNumber TicketClass error2
## 1           0   0      0           0            0           0      0
##   TotalFlightsTaken FeedbackID FeedbackDate Category Rating Comments
## 1                 0          0            0        0      0        0
```

```r
calculate_mean_median <- function(df) {
  numeric_columns <- sapply(df, is.numeric) # Identify numeric columns
  df_numeric <- df[, numeric_columns] # Filter only numeric columns

  means <- apply(df_numeric, 2, mean, na.rm = TRUE) # Calculate means
  medians <- apply(df_numeric, 2, median, na.rm = TRUE) # Calculate medians

  results <- data.frame(Mean = means, Median = medians) # Combine into a dataframe
  return(results)
}
```

We have created a function Summary_missing which provides an entire summary of all the missing values across each variable in the dataset. Using the function for scanning missing values, we get to know that in Flight details datset, the variable Destination is the most commonly missing values as compared to other. Whereas in Customer Details dataset, the Category of the comment given by the passenger is commmonly missing.

For the imputation of missing categorical data, we use the Mode Method. This method involves replacing missing values in categorical variables with the mode, defined as the most frequently occurring value within the dataset. For calculting the mode of any varaible, we have generated a function get_mode which returns the mode value of all categorical data types
Specifically, within the "Flight Details" dataset, the summary_missing function has identified that the "Destination" variable contains missing entries. To address this, we have modified the dataset using the mutate function to replace all missing values in the "Destination" variable with its mode.
Similarly for the Customer details dataset, we identified that the variables "Category" and "Comments" are missing hence we tend to replace the missing value by the mode of the Category variable.

Another method of imputing categorical variable is by using the impute function(). The impute function replaces the missing value in a dataframe. For using the impute function in our analysis, we wpuld need to change the summary_missing function and re-write the code for checking missing value everytime, hence we preferred replace_na method.

Additionally, a function calcualting the mean and median of the dataframe is also stated. For any missing numeric value, we should use mean or median of that variable to deal with missing values.

Link to presentation

Link to presentation :

https://rmit-arc.instructuremedia.com/embed/cd58fc9e-a665-44b3-8b6a-6044a226739a

The link for the presentation has been provided above. The presnetation covers the entire walk through of the steps undertaken to analyse the generated dateset of Qantas Airlines. The steps included are :
1. Introduction
2. Generation of synthetic datasets 3. Correlation variable 4. Merging of datasets
5. Structure of combined dataset
6. Summary statistics
7. Scanning and imputing missing values

References

1. Samuel Klett Navarro (2022) *Creating and pre-processing synthetic data* , RPubs by R Studio, accessed on 24 April 2024. https://rpubs.com/samkn/synthetic_data_creation
2. Ramzi W. Nahhas (2021) *An Introduction to R in Research - Normal distribution* , Bookdown.org , accesed on 25th April 2024 https://bookdown.org/rwnahhas/IntroToR/functions.html
3. Martina Giron (2022) *How to Create a Custom Dataset in R* , Medium , accessed on 24th April 2024 https://towardsdatascience.com/how-to-create-a-custom-dataset-in-r-cf045e286656
4. Shweta Dixit (2023) *Exploring Synthetic data in R* , Medium , accessed on 25th April 2024, https://medium.com/@sdshwetadixit/exploring-synthetic-data-in-r-8834d4217865
5. Rosita Mickeviciute (2023) *Types of airplanes and their functions: a civilian aircraft overview* Aerotime Hub, accessed on 26th April 2024 https://www.aerotime.aero/articles/types-of-airplanes
6. rare-phoenix (2024) *Generating Synthetic Data* , accessed on 24th April 2024 http://rare-phoenix-161610.appspot.com/secured/demos/Generating-Synthetic-Data.html
7. Data Preprocessing (Data Wrangling) *Module 5 - Scan: Missing Values* accessed on 25th April 2024, http://rare-phoenix-161610.appspot.com/secured/Module_05.html#Overview
8. Data Preprocessing (Data Wrangling) *Module 4 -Tidy Data Principles and Manipulating Data* accessed on 25th April 2024, https://rare-phoenix-161610.appspot.com/secured/Module_04.html


```r
citation("magrittr")
```

```
## To cite package 'magrittr' in publications use:
##
##   Bache S, Wickham H (2022). _magrittr: A Forward-Pipe Operator for R_.
##   R package version 2.0.3,
##   <https://CRAN.R-project.org/package=magrittr>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {magrittr: A Forward-Pipe Operator for R},
##     author = {Stefan Milton Bache and Hadley Wickham},
##     year = {2022},
##     note = {R package version 2.0.3},
##     url = {https://CRAN.R-project.org/package=magrittr},
##   }
```

```r
citation("dplyr")
```

```
## To cite package 'dplyr' in publications use:
##
##   Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A
##   Grammar of Data Manipulation_. R package version 1.1.4,
##   <https://CRAN.R-project.org/package=dplyr>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller and Davis Vaughan
##     year = {2023},
##     note = {R package version 1.1.4},
```

```
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```r
citation("tidyr")
```

```
## To cite package 'tidyr' in publications use:
##
##   Wickham H, Vaughan D, Girlich M (2024). _tidyr: Tidy Messy Data_. R
##   package version 1.3.1, <https://CRAN.R-project.org/package=tidyr>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Tidy Messy Data},
##     author = {Hadley Wickham and Davis Vaughan and Maximilian Girlich},
##     year = {2024},
##     note = {R package version 1.3.1},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```r
citation("knitr")
```

```
## To cite package 'knitr' in publications use:
##
##   Xie Y (2023). _knitr: A General-Purpose Package for Dynamic Report
##   Generation in R_. R package version 1.45, <https://yihui.org/knitr/>.
##
##   Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition.
##   Chapman and Hall/CRC. ISBN 978-1498716963
##
##   Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible
##   Research in R. In Victoria Stodden, Friedrich Leisch and Roger D.
##   Peng, editors, Implementing Reproducible Computational Research.
##   Chapman and Hall/CRC. ISBN 978-1466561595
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```