TOURISM AUSTRALIA
TEAM FIVE FINAL DELIVERABLE

Group Members
Aashaya Bharambe (z5360013)
Azalea Qin (z5325013)
Carson Yeung (z5356262)
Leo Hui (z5334709)
Johan Choi (z5418345)

TOURISM
AUSTRALIA

TABLE OF CONTENTS

1. Introduction

Tourism Australia (TA) is a government agency with the key objective of attracting international business and sightseers to Australia. TA seeks to enhance marketing and provide timely insights for the travel industry by better understanding seat supply, travel demand, and relevant metrics like load factors and yield. TA's current processes are challenging and time-intensive, hindering effective decision-making. In addressing the problem statement outlined in *Appendix A,* this report aims to catalyse efficient decision-making through a comprehensive analysis of inbound airline flight schedules using Cirium and the Bureau of Infrastructure and Transport Research Economics (BITRE) data to create graphic visualisations and provide insights to guide marketing efforts. To streamline the process, R Programming Language (R Code) is utilised to allow TA to replicate the visualisations easily.

2. Methodology

Cirium's dataset contained detailed information, including total passenger statistics and transfer percentages. Still, it was flawed due to missing timeframes and many zero or null values in critical columns, potentially leading to inaccurate predictions.

In contrast, BITRE's data was more up-to-date, extending to mid-2023, and had more complete information. However, it lacked specific details such as the origin city, only showing the originating country. This limitation would provide a general overview of air traffic volume but would not allow the identification of popular routes.

3. Analysis

3.1. Seasonality Analysis

Figure 1 demonstrates the total number of passengers entering various Australian cities between April 2022 and April 2023. This suggests Sydney has the highest number of passengers, specifically 6,025,768 over the year.

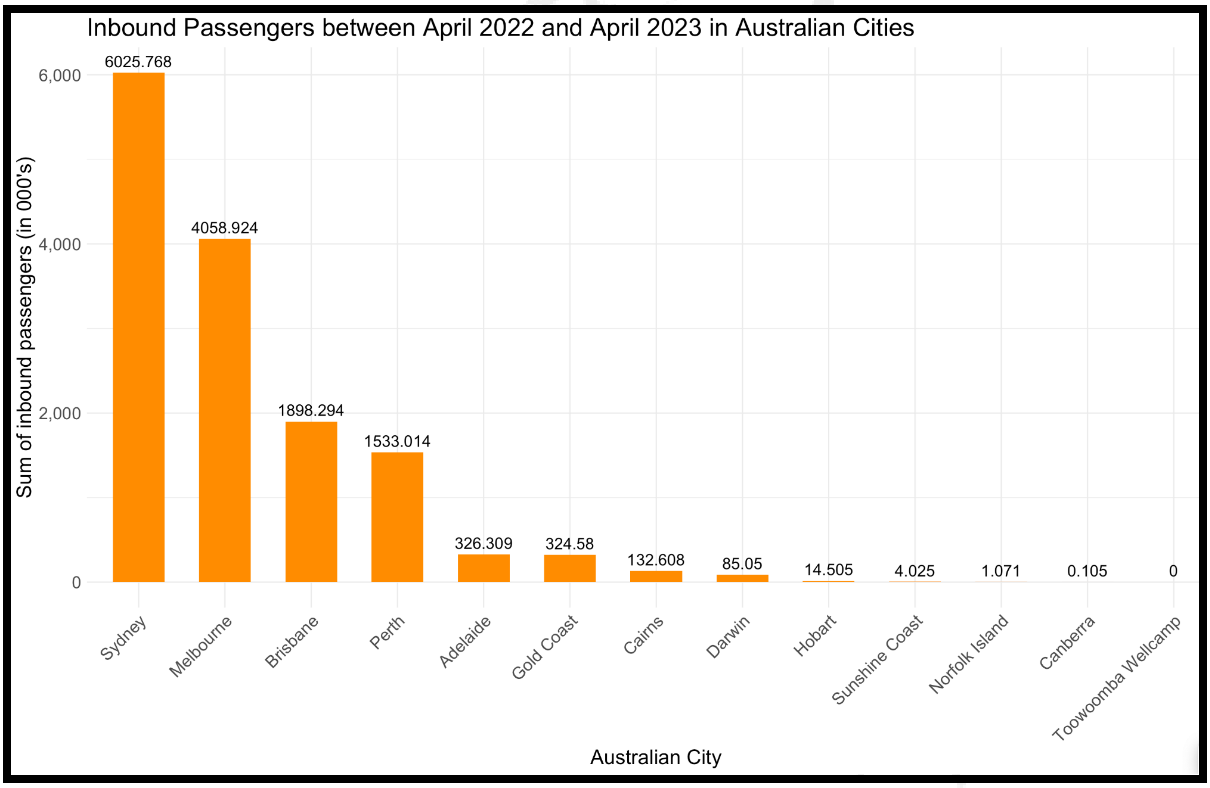

Figure 1: Analysis of Inbound Passengers from April 2022 to April 2023. Data sourced from BITRE.

Building upon the findings of Figure 1, which highlighted Sydney as a primary destination in Australia, *Figure 2* was created to conduct a comprehensive analysis regarding the monthly passenger volumes entering Sydney between 2015 and 2022 to examine potential seasonality trends.

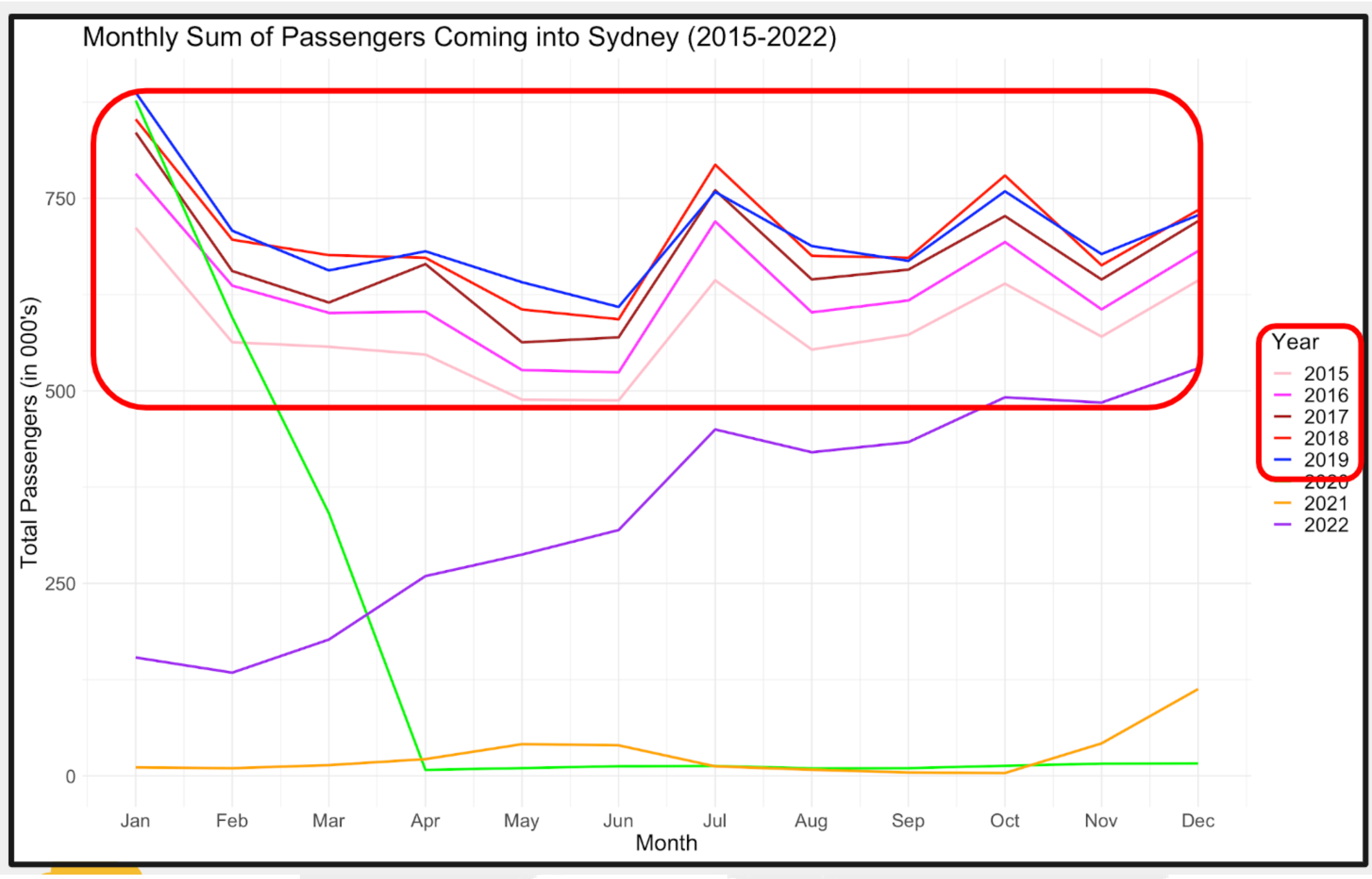

Figure 2: Monthly Total International Passengers Coming into Sydney (2015-2022). (BITRE)

The graph above reveals distinct seasonality from 2015-2019 indicated by the red box, with significant peaks during Australian summer (December to February), and a smaller peak from June to August, likely associated with northern hemisphere summer holidays. Lockdowns and Australian border closures impede seasonality during COVID-19 (2021 and 2021), however, in 2022, Sydney Airport experienced a gradual rise in passenger entrants, indicating recovered travel patterns.

To further confirm the seasonality trends, *Table 1* compares Load Factors between January 2023 and June 2023. Airline Load Factors (LF) are efficiency metrics crucial for airline profitability and identifying factors that may affect passenger loads (Szabo et al., 2018).

The results indicate that Load Factors for all airlines dropped during June, compared to a peak in January, which mirrors the peaks and troughs in *Figure 2.*

Airline	January 2023 Load Factor (%)	June 2023 Load Factor (%)
Qantas (QF)	89.83	77.59
Singapore Airline (SQ)	97.63	86.66
Air India (AI)	96.21	89.62
Delta Airlines (DL)	90.03	74.92
Japan Airlines (JL)	95.81	54.09
All Nippon Airways (NH)	89.80	35.59
Korean Air (KE)	98.51	74.93

Table 1: Comparison of Load Factors between January 2023 and June 2023. (BITRE)

The information presented in *Table 1* and *Figure 2* essentially acts as a guide regarding campaign release dates. Our recommendation for TA is to introduce campaigns during March to effectively increase tourism between April and June. One way is to present the pleasant weather in Queensland around autumn and promote the Great Barrier Reef, or another campaign solution is to create discounted ski packages, which allow ski lovers to experience the Australian Alps for a good bargain.

3.2. Top 5 Load Factor Analysis

During the analysis process, we created *Figure A*, in *Appendix C*, which informed us that Asia had the largest amount of inbound passengers, based on which we chose to focus on Asia. *Figure 3* identifies the leading five airlines with the highest LFs entering Australia. According to BITRE, Singapore - Sydney, Singapore - Melbourne and Singapore - Perth are three of Australia's top five city pairs in 2023, therefore, it is no shock that Scoot and Singapore Airlines have the highest LF(%).

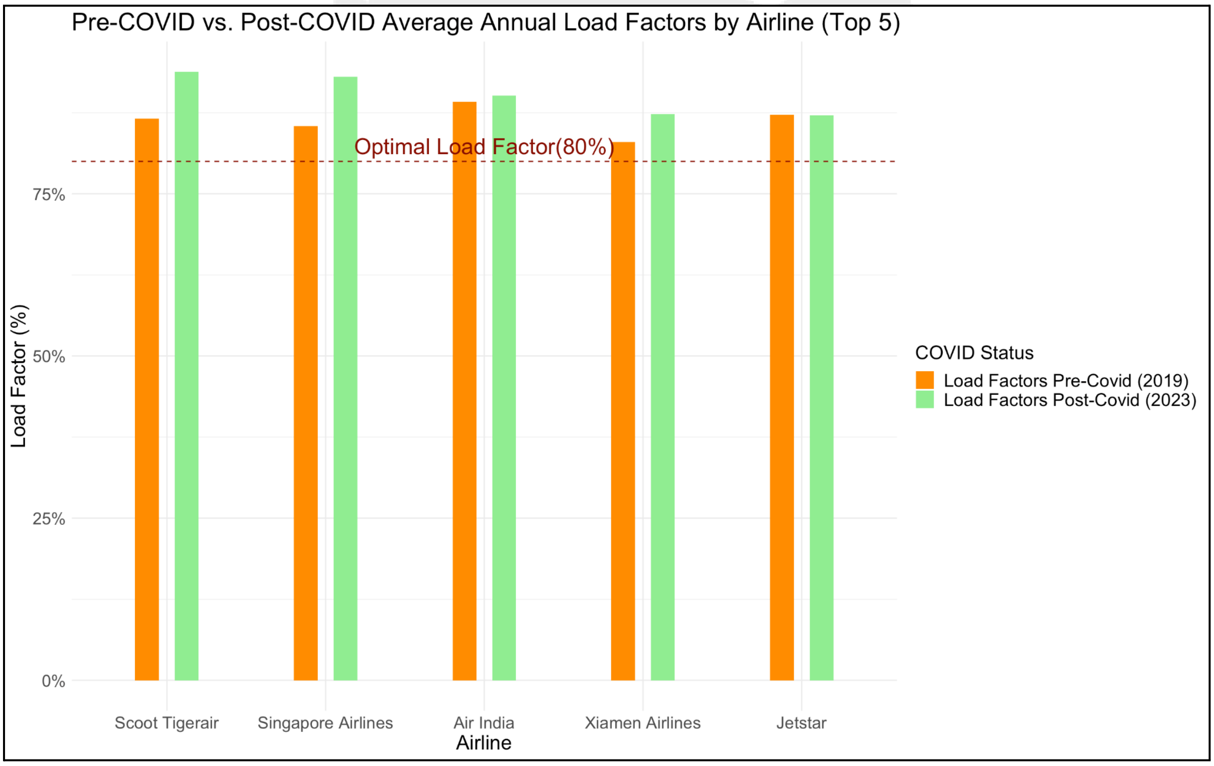

Figure 3: Pre-Covid vs Post-Covid Average Annual Load Factors by Airline (Top 5). Source: BITRE.

3.2.1. Exploring Singapore.

Tables 2.1, 2.2 and *2.3* demonstrate Yield for airlines operating on city-pairs from Singapore. Notably, full-service carriers (FSC) such as SQ and QF hold similar yield values suggesting demand and price are effectively managed for these competitive routes. However, since Scoot and Jetstar are low-cost carriers, their fares are lower, which leads to lower yields (Miyoshi & Rubio Molina-Prados, 2022). Based on this, it is recommended that TA focus its resources on other airlines and less competitive routes.

SINGAPORE TO SYDNEY	
Airline	**Yield (c/Km)**
Qantas Airways (QF)	10.54
British Airways (BA)	10.39
Singapore Airlines (SQ)	10.13
Scoot (TR)	5.24

Table 2.1: Passenger Yield of Singapore-Sydney route in 2023 (Cirium)

SINGAPORE TO MELBOURNE	
Airline	**Yield (c/Km)**
Emirates (EK)	11.19
Singapore Airlines (SQ)	10.17
Qantas Airways (QF)	8.85
Jetstar (JQ)	4.41
Scoot (TR)	3.17

Table 2.2: Passenger Yield of Singapore-Melbourne route in 2023 (Cirium)

SINGAPORE TO PERTH	
Airline	**Yield (c/Km)**
Qantas Airways (QF)	11.55
Singapore Airlines (SQ)	9.91
Scoot (TR)	4.91

Table 2.3: Passenger Yield of Singapore-Perth route in 2023 (Cirium)

3.2.2. Partnerships with India

Figure 3 demonstrates high load factors for Air India with an average LF of 90.11%, which indicates profitability; with extensive capacity, airlines can compete on these routes. According to Ganotra (2023), currently, only Air India and Qantas, operate on direct India-Australia routes stated in *Extract 1, Appendix C.*

In 2022, the Minister of Foreign Affairs stated that they are expecting the number of Indian tourists to increase to 70 million in 2035 from 23 million in 2017, which would create an injection of $6.1 (or an aspiring $9.1) billion into the Australian economy (Vats, 2023). Furthermore, Ghosh (2023) emphasised this trend by noting that Australia welcomed 383,000 visitors from India in the last financial year ending June 2023, which brought $1.3 billion into the Australian economy, which suggests that LF% are high due to lower seat supply.

Tourism Australia should capitalise on this growth by introducing direct flights coming from India, similar to AI's new Mumbai to Melbourne route starting 15th December 2023:

(1) Delhi - Brisbane

(2) Delhi - Gold Coast

(3) Delhi - Perth

(4) Mumbai - Sydney

There are constraint issues, however the re-establishment of Air India, Qantas's growth and Western Sydney Airport, this is a potential opportunity to bridge the gap between Australia and India and create the *"Taj to Opera"* campaign and utilise promotions to capture the cost-sensitive passengers stated by Department of Foreign Affairs. Adding these routes to the network is also mutually beneficial for the airlines.

3.3. Bottom Five Load Factors Analysis

Figure 4 illustrates five airlines with the lowest LF%. After close observation, 4 out of 5 airlines are LCCs and used to have an average LF of 75% before COVID-19.

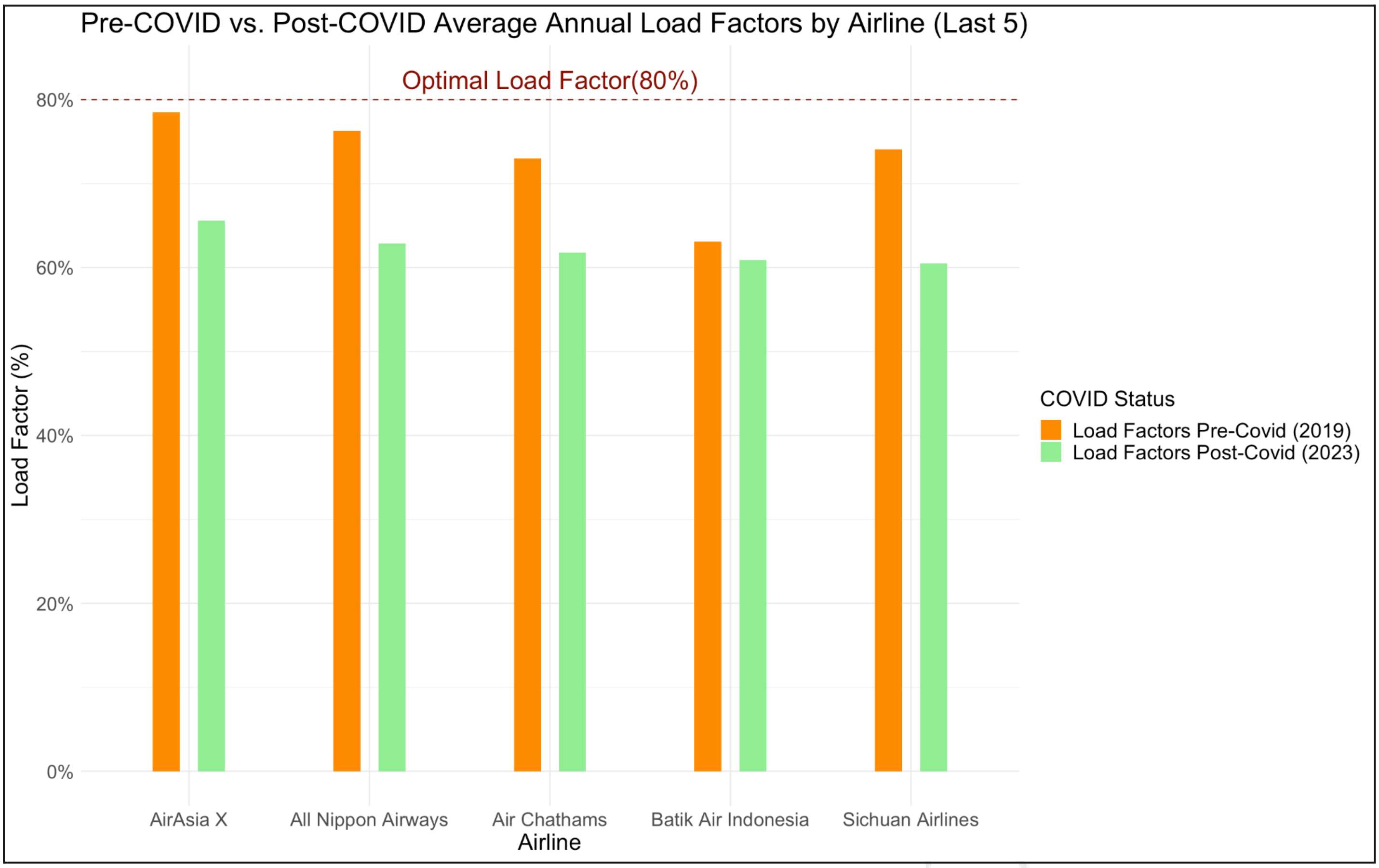

Figure 4: Pre-Covid vs Post-Covid Average Annual Load Factors by Airline (Last 5). Source: BITRE.

3.3.1. Collaborating with ANA

All Nippon Airways (ANA) presents an opportunity for TA to capitalise on. According to BITRE, between 2017 and 2023, ANA increased its seat supply by 6,774 seats, which could be a reason for its low LF%. Statistica (2023) states that Japan bought in 162,000 visitors from Japan. Currently, they operate two flights into Australia:

(1) Tokyo(Haneda) - Sydney (HND-SYD)

(2) Tokyo(Narita) - Perth (NRT-PER)

Table 3 demonstrates the average airfare during 2023 for HND - SYD. This table indicates that ANA offers the lowest prices; however, it still holds a lower LF% shown in *Figure 5*. This suggests that the capacity should be spread across Australia rather than concentrating in Sydney and Perth.

Given that ANA is Skytrax's third-best airline in the world, TA should collaborate and create promotions with ANA to encourage more routes from Tokyo to Melbourne and Brisbane (ANA, 2023).

Airline	Average Airfare 2023
Qantas Airways (QF)	AUD$ 830.02
All Nippon Airways (NH)	AUD$ 696.61
Japan Airlines (JL)	AUD$ 906.57

Table 3: Average Airfare 2023, for HND-SYD—data Source: Cirium

4. References (Harvard Style)

ANA (2023) *ANA AWARED 2023 TOP WINNER, ANA Awarded 2023 SKYTRAX Top Winner for Airport Services, Cleanliness and Airline Staff in Asia, Ranked 3rd in Airline of the Year | Press Release | ANA Group Corp.'s Information*. Available at: https://www.anahd.co.jp/group/en/pr/202306/20230620.html (Accessed: 09 November 2023).

Ganotra, H. (2021). List of Non-stop Flights Between Australia and India. *Myticketstoindia (AUS)*. Retrieved from: https://www.myticketstoindia.com.au/blog/non-stop-international-flights-between-australia-and-india/

Ghosh, A. (2023). Skift India Report: How Australia Is Gaining from India's Tourism Growth. *Skift*. Retrieved from: https://skift.com/2023/09/25/skift-india-report-how-australia-is-gaining-from-indias-tourism-growth/

Miyoshi, C. & Rubio Molina-Prados, J. (2022). 'Measuring the impact of long-haul low-cost carriers on lowering fares: A quasi-experimental design to assess the pre-COVID market', *Transport policy*, vol. 128, pp. 52–64. Retrieved from: https://doi.org/10.1016/j.tranpol.2022.09.007

Szabo, S., Makó, S., Tobisová, A., Hanák, P. & Pilát, M. (2018). 'Effect of the load factor on the ticket price', *Problemy Transportu*, vol. 13, no. 3, pp. 41–49. Retrieved from: https://doi.org/10.20858/tp.2018.13.3.4

Vats, N. (2023). 'An Indian Economic Strategy to 2035: Navigating from Potential to Delivery; Peter N Varghese AO', *Jindal Journal of International Affairs*, vol. 10, no. 2, pp. 84–86. Retrieved from: https://doi.org/10.54945/jjia.v10i2.179

Appendix A: Problem Statement

The following is the problem statement team 5 derived during the initial stages of the project.

Tourism Australia seeks to improve its marketing efforts and their ability to provide timely insights to the international and domestic travel industry to assist them in selling Australia as a destination. To achieve this, Tourism Australia requires a strong understanding of the seat supply and travel demand, which requires efficient and accurate methods of gathering performance indicators such as Available Seat kilometres (ASKs), Flight LFs, and Airline Yield. Currently, the data required for these KPIs is difficult to access and presents a time-consuming challenge, hindering Tourism Australia's decision-making process.

Appendix B: Methodology

Appendix C: Additional Graphs

Figure A: Bubble Map outlining the total number of incoming passengers from overseas. Datasource; Cirium

Extract One: Currently flights operating in Australia (Ganotra, 2023

(1) Delhi - Melbourne (Qantas)

(2) Bangalore - Sydney (Qantas)

(3) Delhi - Melbourne (Air India)

(4) Delhi - Sydney (Air India)

5. Appendix D - R CODE

```
## Read library
library(dplyr)
library(tidyr)
library(ggplot2)

## Import the dataset
bitre_data <- read.csv("Update Load Factor Data.csv", header = TRUE)
bitre_pax <- read.csv("number of intl pax by aus city.csv", header = TRUE)

traffic_asia <- read.csv("cirium_traffic_asia.csv", header = T)
traffic_europe <- read.csv("cirium_traffic_europe.csv", header = T)
traffic_northamerica <- read.csv("cirium_traffic_northamerica.csv", header = T)
traffic_oceania <- read.csv("cirium_traffic_oceania.csv", header = T)

## PREPARING THE DATASET
# eliminate the N/A column and rename the columns
bitre_data <- bitre_data[-c(1,2,3), ]
colnames(bitre_data) <- bitre_data[1, ]
bitre_data <- bitre_data[-1, -c(12,13,14,15,16,17,18,19,20,21,22,23)]

bitre_pax <- bitre_pax[ , -c(14:38)]

new_names <- c("Mkt_Al", "Orig", "Dest", "Op_Al", "Stop_1_Al", "Stop_1_Airport",
            "Stop_2_Al", "Stop_2_Airport", "Total_pax", "Pax_share", "Poo_Orig(%)",
            "Poo_Dest(%)", "Poo_Other(%)", "Fare", "Rev", "Yield", "Year", "Month",
            "Class", "Year_Month_Day")
colnames(traffic_asia) <- new_names
colnames(traffic_europe) <- new_names
colnames(traffic_northamerica) <- new_names
colnames(traffic_oceania) <- new_names

# eliminate the Cargo
bitre_data <- subset(bitre_data, !(bitre_data$`PAX IN` == ".."))
bitre_data <- subset(bitre_data, !(bitre_data$`PAX IN` == "(d)"))
bitre_data <- subset(bitre_data, !(bitre_data$`SEATS IN` == ".."))

bitre_pax <- subset(bitre_pax, !(bitre_pax$PaxIn == ".."))

# select the Destination in Australia for cirium_traffic data
Australian_airport <- c("SYD", "MEL", "BNE", "PER", "ADL", "CNS",
                  "OOL", "DRW", "CBR", "HBA")
traffic_asia_aus <- subset(traffic_asia, Dest %in% Australian_airport)
```

```r
traffic_europe_aus <- subset(traffic_europe, Dest %in% Australian_airport)
traffic_northamerica_aus <- subset(traffic_northamerica, Dest %in%
Australian_airport)
traffic_oceania_aus <- subset(traffic_oceania, Dest %in% Australian_airport)

# check and modify the type of cols into useful type
str(bitre_data$`PAX IN`)
str(bitre_data$`SEATS IN`)

Convert_number <- function(x) {
  return(as.numeric(gsub("[ ,]", "", x)))
} # modify method for bitre data

Converting_method <- function(x){
  return(as.numeric(x))
} # modify method for cirium _traffic data

for (col_index in 4:11) {
  bitre_data[ ,col_index] <- Convert_number(bitre_data[ ,col_index])
}

for (col_index in 5:13) {
  bitre_pax[ ,col_index] <- Convert_number(bitre_pax[ ,col_index])
}

for (i in 9:16){
  traffic_asia_aus[ , i] <- Converting_method(traffic_asia_aus[ , i])
}
for (i in 9:16){
  traffic_europe_aus[ , i] <- Converting_method(traffic_europe_aus[ , i])
}
for (i in 9:16){
  traffic_northamerica_aus[ , i] <- Converting_method(traffic_northamerica_aus[ , i])
}
for (i in 9:16){
  traffic_oceania_aus[ , i] <- Converting_method(traffic_oceania_aus[ , i])
}

# convert the column indicates time into date value
bitre_data$MONTH <- as.Date(paste0("01-", bitre_data$MONTH), format =
"%d-%b-%y")
bitre_pax$Month <- as.Date(paste0("01-", bitre_pax$Month), format = "%d-%b-%y")
```

```
traffic_asia_aus$Year_Month_Day <-
as.Date(as.character(traffic_asia_aus$Year_Month_Day), format="%Y-%m-%d")
traffic_europe_aus$Year_Month_Day <-
as.Date(as.character(traffic_europe_aus$Year_Month_Day), format="%Y-%m-%d")
traffic_northamerica_aus$Year_Month_Day <-
as.Date(as.character(traffic_northamerica_aus$Year_Month_Day),
format="%Y-%m-%d")
traffic_oceania_aus$Year_Month_Day <-
as.Date(as.character(traffic_oceania_aus$Year_Month_Day), format="%Y-%m-%d")

# delete the Year, Month column
traffic_asia_aus <- traffic_asia_aus[ ,-c(17,18)]
traffic_europe_aus <- traffic_europe_aus[ ,-c(17,18)]
traffic_northamerica_aus <- traffic_northamerica_aus[ ,-c(17,18)]
traffic_oceania_aus <- traffic_oceania_aus[ ,-c(17,18)]

# adding a new column to calculate load_factor
bitre_data$Load_factor <- round(bitre_data$`PAX IN` / bitre_data$`SEATS IN`, 4)

core_markets <- c("India", "Korea", "Qatar", "Canada", "Indonesia",
              "Japan", "Singapore", "USA", "Malaysia", "China",
              "United Arab Emirates", "New Zealand", "Hong Kong")

bitre_filtered_country <- subset(bitre_data, COUNTRY %in% core_markets)
```

Inbound Passenger Graph

```
# filter the data
bitre_pax_recent <- bitre_pax %>%
  filter(Month >= as.Date("2022-04-01") & Month <= as.Date("2023-04-01"))

# calculate the sum of the inbound passengers
sum_pax_in <- aggregate(PaxIn ~ AustralianPort, data = bitre_pax_recent, FUN =
sum)
sum_pax_in <- sum_pax_in %>%
  arrange(desc(PaxIn))

# convert the AustralianPort column to a factor with the desired order of levels
sum_pax_in$AustralianPort <- factor(sum_pax_in$AustralianPort,
unique(sum_pax_in$AustralianPort))

# fitting the data
ggplot(data = sum_pax_in, aes(x = AustralianPort, y = PaxIn / 1000)) +
```

```
  geom_bar(stat = "identity", fill = "darkorange", width = 0.6) +
  geom_text(aes(label = PaxIn/1000), vjust = -0.5, size = 5) +
  labs(x = "Australian City",
       y = "Sum of inbound passengers (in thousands)",
       title = "Inbound Passengers between April 2022 and April 2023 in Australian
Cities") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.text = element_text(size = 16),
        legend.title = element_text(size = 17),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 15),
        plot.title = element_text(size = 22))



## Seasonality analysis in Sydney
# prepare the data
syd_pax <- bitre_pax %>%
  filter(AustralianPort == "Sydney")
syd_pax_recent <- syd_pax %>%
  filter(Month >= as.Date("2015-01-01") &
           Month <= as.Date("2022-12-01"))

# calculate the monthly sum
syd_monthly_pax_sum <- syd_pax_recent %>%
  group_by(Month) %>%
  summarize(Total_PaxIn = sum(PaxIn))

# extract the Year and Month
syd_monthly_pax_sum$Year <- year(syd_monthly_pax_sum$Month)
syd_monthly_pax_sum$actual_month <- month(syd_monthly_pax_sum$Month)

# create the line chart
ggplot(syd_monthly_pax_sum, aes(x = actual_month, y = Total_PaxIn / 1000, color =
as.factor(Year))) +
  geom_line(linewidth = 1) +
  labs(title = "Monthly Sum of Passengers Coming into Sydney (2015-2022)",
       x = "Month",
       y = "Total Passengers (in thousands)",
       color = "Year") +
```

```
  scale_color_manual(values = c("2015" = "pink", "2016" = "magenta", "2017" =
"brown", "2018" = "red", "2019" = "blue", "2020" = "green", "2021" = "orange", "2022"
= "purple")) +
  scale_x_continuous(breaks = 1:12, labels = month.abb) +
  theme_minimal() +
  theme(legend.text = element_text(size = 16),
        legend.title = element_text(size = 17),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 15),
        plot.title = element_text(size = 22))
```

Load factor analysis by different airline

```
# find the sub-dataset of pre-covid and post-covid
bitre_filtered_pre_covid <- subset(bitre_filtered_country,
    MONTH >= as.Date("2019-01-01") & MONTH < as.Date("2020-01-01"))
bitre_filtered_post_covid <- subset(bitre_filtered_country,
                                    MONTH >= as.Date("2023-01-01"))

# calculate the annual mean of load factor
avg_load_factor_precovid <- aggregate(Load_factor ~ AIRLINE, data =
bitre_filtered_pre_covid, FUN = mean)
avg_load_factor_postcovid <- aggregate(Load_factor ~ AIRLINE, data =
bitre_filtered_post_covid, FUN = mean)

# find the joint airline
common_airline <- inner_join(avg_load_factor_precovid, avg_load_factor_postcovid,
                             by = "AIRLINE")
new_names <- c("Airline", "Load_factor_19", "Load_factor_23")
colnames(common_airline) <- new_names

# convert the original dataset to bar-chart suitable dataset
common_airline <- common_airline %>%
  arrange(desc(Load_factor_23))
common_airline <- common_airline %>%
  pivot_longer(cols = c(Load_factor_19, Load_factor_23),
               names_to = "Covid_Period",
               values_to = "Load_Factor")

# convert the Covid_Period column to a factor with the desired order of levels
```

```
common_airline$Covid_Period <- factor(common_airline$Covid_Period, levels =
c("Load_factor_pre_covid", "Load_factor_post_covid"))

# convert the Airline column to a factor with the desired order of levels
common_airline$Airline <- factor(common_airline$Airline, levels =
unique(common_airline$Airline))

# top 5 Load factor in post_Covid
common_top_5 <- common_airline[1:10, ]
ggplot(common_top_5, aes(x = Airline, y = Load_Factor, fill = Covid_Period)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.5), width = 0.3) +
  labs(
    x = "Airline",
    y = "Load Factor (%)",
    title = "Pre-COVID vs. Post-COVID Average Annual Load Factors by Airline (Top
5)",
    fill = "COVID Period"
  ) +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent_format(scale = 100)) +
  scale_fill_manual(values = c("Load_factor_19" = "darkorange", "Load_factor_23" =
"lightgreen"),
                    name = "COVID Status",
                    labels = c("Load Factors Pre-Covid (2019)", "Load Factors Post-Covid
(2023)"))  +
  geom_hline(yintercept = 0.8, linetype = "dashed", color = "darkred") +
  annotate("text", x = 3, y = 1.03 * 0.8, label = "Optimal Load Factor(80%)", color =
"darkred", size = 7) +
  theme(legend.text = element_text(size = 16),
        legend.title = element_text(size = 17),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 15),
        plot.title = element_text(size = 22))

# last 5 Load factor in post_Covid
common_last_5 <- common_airline[53:62, ]
ggplot(common_last_5, aes(x = Airline, y = Load_Factor, fill = Covid_Period)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.5), width = 0.3) +
  labs(
    x = "Airline",
    y = "Load Factor (%)",
    title = "Pre-COVID vs. Post-COVID Average Annual Load Factors by Airline (Last
5)",
    fill = "COVID Period"
```

```
  ) +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent_format(scale = 100)) +
  scale_fill_manual(values = c("Load_factor_19" = "darkorange", "Load_factor_23" =
"lightgreen"),
                    name = "COVID Status",
                    labels = c("Load Factors Pre-Covid (2019)", "Load Factors Post-Covid
(2023)"))  +
  geom_hline(yintercept = 0.8, linetype = "dashed", color = "darkred") +
  annotate("text", x = 3, y = 1.03 * 0.8, label = "Optimal Load Factor(80%)", color =
"darkred", size = 7) +
  theme(legend.text = element_text(size = 16),
        legend.title = element_text(size = 17),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 15),
        plot.title = element_text(size = 22))


## Yield analysis for cirium_traffic data
# subset the 2023 data
asia_2023 <- traffic_asia_aus %>%
  filter(Year_Month_Day >= as.Date("2023-01-01") & Year_Month_Day <
as.Date("2024-01-01"))
europe_2023 <- traffic_europe_aus %>%
  filter(Year_Month_Day >= as.Date("2023-01-01") & Year_Month_Day <
as.Date("2024-01-01"))
northamerica_2023 <- traffic_northamerica_aus %>%
  filter(Year_Month_Day >= as.Date("2023-01-01") & Year_Month_Day <
as.Date("2024-01-01"))
oceania_2023 <- traffic_oceania_aus %>%
  filter(Year_Month_Day >= as.Date("2023-01-01") & Year_Month_Day <
as.Date("2024-01-01"))

# define a function to look up different average yield for different routes
airline_yield <- function(data, Orig_name, Dest_name){
  temp <- data %>%
    filter(Orig == Orig_name & Dest == Dest_name)
  temp <- temp[temp$Stop_1_Al == "" | temp$Stop_1_Al == "NULL", ]
  output <- temp %>%
    group_by(Mkt_Al) %>%
    summarise(Annual_avg_yield = mean(Yield))
  return(output)
}
```

```
# explore the average yield for different airlines in each route
sin_syd <- airline_yield(asia_2023, "SIN", "SYD")
sin_mel <- airline_yield(asia_2023, "SIN", "MEL")
akl_syd <- airline_yield(oceania_2023, "AKL", "SYD")
sin_per <- airline_yield(asia_2023, "SIN","PER")
akl_mel <- airline_yield(oceania_2023, "AKL", "MEL")
dps_mel <- airline_yield(asia_2023, "DPS", "MEL")
sin_bne <- airline_yield(asia_2023, "SIN", "BNE")
lax_syd <- airline_yield(northamerica_2023, "LAX", "SYD")
dps_per <- airline_yield(asia_2023, "DPS", "PER")
lax_mel <- airline_yield(northamerica_2023, "LAX", "MEL")
icn_syd <- airline_yield(asia_2023, "ICN", "SYD")
sgn_syd <- airline_yield(asia_2023, "SGN", "SYD")
mnl_syd <- airline_yield(asia_2023, "MNL", "SYD")
blr_syd <- airline_yield(asia_2023, "BLR", "SYD")
del_syd <- airline_yield(asia_2023, "DEL", "SYD")
hnd_syd <- airline_yield(asia_2023, "HND", "SYD")
hnd_mel <- airline_yield(asia_2023, "HND", "MEL")

# List of data frames
data_frames <- list(
  sin_syd, sin_mel, akl_syd, sin_per, akl_mel,
  dps_mel, sin_bne, lax_syd, dps_per, lax_mel,
  icn_syd, sgn_syd, mnl_syd, del_syd
)

# define a function to filter out the zero yield (abnormal)
yield_filter <- function(data){
  output <- data %>%
    filter(Annual_avg_yield != 0)
  return(output)
}

# Loop through each data frame and apply the yield_filter function
for (i in seq_along(data_frames)) {
  data_frames[[i]] <- yield_filter(data_frames[[i]])
}

# Loop through each data frame and arrange by Annual_avg_yield
for (i in seq_along(data_frames)) {
  data_frames[[i]] <- data_frames[[i]] %>%
    arrange(desc(Annual_avg_yield))
}
```

```r
# Assign the modified data frames back to their respective variables
sin_syd <- data_frames[[1]]
sin_mel <- data_frames[[2]]
akl_syd <- data_frames[[3]]
sin_per <- data_frames[[4]]
akl_mel <- data_frames[[5]]
dps_mel <- data_frames[[6]]
sin_bne <- data_frames[[7]]
lax_syd <- data_frames[[8]]
dps_per <- data_frames[[9]]
lax_mel <- data_frames[[10]]
icn_syd <- data_frames[[11]]
sgn_syd <- data_frames[[12]]
mnl_syd <- data_frames[[13]]
del_syd <- data_frames[[14]]
```

Fare analysis

```r
# subset the hnd to syd data
hnd_syd_fare <- asia_2023 %>%
  filter(Orig == "HND" & Dest == "SYD")

# concentrate only on the direct flight
hnd_syd_fare <- hnd_syd[hnd_syd_fare$Stop_1_Al == "" , ]

# filter out the zero fare
hnd_syd_fare <- hnd_syd_fare %>%
  filter(Fare != 0)

# find the average fare in different airline
hnd_syd_fare <- hnd_syd_fare %>%
  group_by(Mkt_Al) %>%
  summarise(Avg_fare = mean(Fare))
```