REPORT - Analysis of Incheon International Airport Transfers

1. Introduction

[Data Introduction]

<Data Name>
We used data named 'Incheon International Airport's Transfer Passengers Information', from January 2023 to June 2023. There are two datasets of arrival and departure. The Arrival dataset means the transfer passenger's final destination from Incheon, and the Departure dataset means where did the transfer passenger leave before arriving at Incheon.

<Data Source>
We got them from Incheon International Airport Corporation's Open Data Platform.

<Data Size>
Both have about 61,000 rows and 8 columns, including country, airport, flight date, scheduled time, actual time, and the number of transfer passengers. Since only the last column is numeric data, we will use and process it as important feature.

We will import them as 'arrival' and 'departure' later in [Data Importing].

[Project Goal]
The Incheon International Airport is a world-class airport that is heavily used not only by Koreans but also by many people around the world. Through the data of departure and arrival transfers at Incheon International Airport, we aim to analyze the airport transfer of striking country and continent over time, and estimate and suggest the flight path related to them.

Our analysis will support travelers in understanding trends and establishing future travel plans, and from the airport's view, they can get an idea of they can get an idea of profitable/popular routes to develop and operate more intensively as a prominent transfer airport.

[Columns We Used]
As we mentioned above, our flow of analysis is i) analyzing the airport transfer of striking country/continent over time ii) estimating and suggesting the flight path related to them. Considering this, throughout the overall analysis, we focused on Arrival/Departure Country, Airport, Flight Date, Number of Transfer Passengers, and additionally, Scheduled Time in estimating the routes.

2. Data Import & Pre-processing

```
# installing packages and importing libraries

install.packages("tidyverse")
install.packages("dplyr")
install.packages("ggplot2")

library(tidyverse)
library(dplyr)
library(ggplot2)
```

[Data Importing]
Before starting, we installed and imported necessary packages like tidyverse, dplyr, ggplot2. And then we imported two datasets as dataframes named 'arrival' and 'departure'.

```
# importing dataset

arrival <- read_csv("환승승객현황 정보_2023년 상반기_도착.csv", locale = locale("ko", encoding = "euc-kr"), show_col_types = FALSE)

departure <- read_csv("환승승객현황 정보_2023년 상반기_출발.csv", locale = locale("ko", encoding = "euc-kr"), show_col_types = FALSE)

head(arrival)
head(departure)
```

도착국가 <chr>	도착공항 <chr>	공항코드 <chr>	운항일자 <date>	계획시간 <S3: hms>	실제시간 <S3: hms>	운항편명 <chr>	환승승객 <dbl>
베트남	두옹 당(푸꿕)	PQC	2023-01-01	00:30:00	01:01:00	VJ978	0
필리핀	마닐라	MNL	2023-01-01	04:10:00	03:36:00	7C2306	5
태국	방콕/수완나품	BKK	2023-01-01	04:20:00	04:05:00	KE658	15
베트남	나트랑	CXR	2023-01-01	04:30:00	04:24:00	VJ836	0
일본	도쿄/하네다	HND	2023-01-01	04:35:00	04:47:00	MM809	0
필리핀	마닐라	MNL	2023-01-01	04:45:00	03:59:00	KE624	99

6 rows

Figure 1: Arrival Dataset (see code chunk 2)

출발국가 <chr>	출발공항 <chr>	공항코드 <chr>	운항일자 <date>	계획시간 <S3: hms>	실제시간 <S3: hms>	운항편명 <chr>	환승승객 <dbl>
터키	이스탄불	IST	2023-01-01	00:15:00	00:19:00	TK091	0
카타르	도하	DOH	2023-01-01	00:25:00	00:40:00	QR859	2
에티오피아	볼레(아디스아바바)	ADD	2023-01-01	00:30:00	00:38:00	ET673	4
필리핀	마닐라	MNL	2023-01-01	00:40:00	00:57:00	5J187	0
네덜란드	암스테르담	AMS	2023-01-01	01:25:00	01:46:00	KL862	9
베트남	두옹 당(푸꿕)	PQC	2023-01-01	01:45:00	01:48:00	VJ975	0

6 rows

Figure 2: Departure Dataset (see code chunk 2)

[Data Pre-processing]

First, we checked missing data, and both datasets had no missing values. So there was no need for imputing.

```
# Preprocessing

# i) check missing data
sum(is.na(arrival))
sum(is.na(departure))
```

Second, we renamed Korean column names to English.

```
# ii) rename Korean column names into English
colnames(arrival) <- c("Arrival_Country", "Arrival_Airport", "A_Airport_Code", "A_Flight_Date", "A_Scheduled_Time", "A_Actual_Time", "A
_Flight_Name", "A_Transfer_Passengers")
colnames(departure) <- c("Departure_Country", "Departure_Airport", "D_Airport_Code", "D_Flight_Date", "D_Scheduled_Time", "D_Actual_Tim
```

```
e", "D_Flight_Name", "D_Transfer_Passengers")

head(arrival)
head(departure)
```

```
# change "푸\\u0085O" to original letter
arrival$Arrival_Airport <- gsub("\\u0085O", "꿕", arrival$Arrival_Airport)
departure$Departure_Airport <- gsub("\\u0085O", "꿕", departure$Departure_Airport)
```

Arrival_Country <chr>	Arrival_Airport <chr>	A_Airport_Code <chr>	A_Flight_Date <date>	A_Scheduled_Time <S3: hms>	A_Actual_Time <S3: hms>	A_Flight_Name <chr>
베트남	두옹 당(푸꿕)	PQC	2023-01-01	00:30:00	01:01:00	VJ978
필리핀	마닐라	MNL	2023-01-01	04:10:00	03:36:00	7C2306
태국	방콕/수완나품	BKK	2023-01-01	04:20:00	04:05:00	KE658
베트남	나트랑	CXR	2023-01-01	04:30:00	04:24:00	VJ836
일본	도쿄/하네다	HND	2023-01-01	04:35:00	04:47:00	MM809
필리핀	마닐라	MNL	2023-01-01	04:45:00	03:59:00	KE624

Figure 3: Arrival Dataset - Renamed Columns (see code chunk 4)

Departure_Country <chr>	Departure_Airport <chr>	D_Airport_Code <chr>	D_Flight_Date <date>	D_Scheduled_Time <S3: hms>	D_Actual_Time <S3: hms>	D_Flight_Name <chr>
터키	이스탄불	IST	2023-01-01	00:15:00	00:19:00	TK091
카타르	도하	DOH	2023-01-01	00:25:00	00:40:00	QR859
에티오피아	볼레(아디스아바바)	ADD	2023-01-01	00:30:00	00:38:00	ET673
필리핀	마닐라	MNL	2023-01-01	00:40:00	00:57:00	5J187
네덜란드	암스테르담	AMS	2023-01-01	01:25:00	01:46:00	KL862
베트남	두옹 당(푸꿕)	PQC	2023-01-01	01:45:00	01:48:00	VJ975

Figure 4: Departure Dataset - Renamed Columns (see code chunk 4)

Third, we created new columns of Continent and Day of Week, from existing columns Country and Flight date.

```
# iii) create new columns for future investigation

# first, continent column

# check unique data
unique(arrival$Arrival_Country)
unique(departure$Departure_Country)

# create dataframe to rename Korean country names to English
country_mapping <- data.frame(
  Korean = c("베트남", "필리핀", "태국", "일본", "홍콩", "미국", "싱가포르",
             "말레이시아", "라오스", "괌", "인도네시아", "캄보디아", "폴란드", "프랑스",
             "한국", "터키", "대만", "아랍에미리트", "독일", "중국", "몽골",
             "핀란드", "이탈리아", "헝가리", "미얀마", "스페인", "네덜란드", "에티오피아",
             "영국", "캐나다", "카타르", "하와이", "호주", "뉴질랜드", "사이판",
             "네팔", "인도", "우즈베키스탄", "스리랑카", "카자흐스탄", "오스트리아", "사우디아라비아",
             "방글라데시", "이스라엘", "브루나이", "이집트", "체코", "마카오", "스위스",
             "레바논", "노르웨이", "그리스", "팔라우", "키르기즈스탄", "나우루"),
  English = c("Vietnam", "Philippines", "Thailand", "Japan", "Hong Kong", "United States of America", "Singapore",
              "Malaysia", "Laos", "Guam", "Indonesia", "Cambodia", "Poland", "France",
              "South Korea", "Turkey", "Taiwan", "United Arab Emirates", "Germany", "China", "Mongolia",
              "Finland", "Italy", "Hungary", "Myanmar", "Spain", "Netherlands", "Ethiopia",
              "United Kingdom", "Canada", "Qatar", "Hawaii", "Australia", "New Zealand", "Saipan",
              "Nepal", "India", "Uzbekistan", "Sri Lanka", "Kazakhstan", "Austria", "Saudi Arabia",
              "Bangladesh", "Israel", "Brunei", "Egypt", "Czech Republic", "Macau", "Switzerland",
              "Lebanon", "Norway", "Greece", "Palau", "Kyrgyzstan", "Nauru")
)

# convert Departure_Country and Arrival_Country into English
departure <- departure %>%
  left_join(country_mapping, by = c("Departure_Country" = "Korean")) %>%
  mutate(Departure_Country = coalesce(English, Departure_Country)) %>%
  select(-English)

arrival <- arrival %>%
  left_join(country_mapping, by = c("Arrival_Country" = "Korean")) %>%
  mutate(Arrival_Country = coalesce(English, Arrival_Country)) %>%
  select(-English)

# define a function to map continent information
map_continent <- function(country) {
  if (country %in% c("Japan", "Taiwan", "Hong Kong", "Macau", "China", "Mongolia", "South Korea")) {
    return("East Asia")
  } else if (country %in% c("Vietnam", "Thailand", "Malaysia", "Singapore", "Indonesia", "Laos", "Cambodia", "Myanmar", "Philippines",
"Brunei", "Bhutan")) {
    return("SouthEast Asia")
  } else if (country %in% c("Uzbekistan", "Kyrgyzstan", "Turkmenistan", "Kazakhstan", "Azerbaijan")) {
    return("Central Asia")
  } else if (country %in% c("United Arab Emirates", "Turkey", "Qatar", "Saudi Arabia", "Israel", "Lebanon", "India", "Sri Lanka", "Paki
stan", "Bangladesh", "Nepal", "Maldives")) {
    return("West/South Asia")
  } else if (country %in% c("United States of America", "Canada")) {
    return("North America")
  } else if (country %in% c("Brazil", "Argentina", "Mexico", "Chile", "Colombia", "Peru")) {
    return("South America")
  } else if (country %in% c("Australia", "New Zealand", "Guam", "Palau", "Saipan", "Hawaii", "Nauru")) {
    return("Oceania")
  } else if (country %in% c("Ethiopia", "Nigeria", "Egypt", "Kenya", "South Africa", "Ghana", "Uganda", "Sudan", "Algeria", "Morocco",
```

```
"Tanzania", "Mozambique")) {
    return("Africa")
  } else if (country %in% c("United Kingdom", "Germany", "France", "Italy", "Spain", "Netherlands", "Belgium", "Sweden", "Norway", "Den
mark", "Finland", "Switzerland", "Austria", "Portugal", "Greece", "Poland", "Czech Republic", "Hungary", "Romania", "Croatia", "Ukrain
e", "Russia", "Belarus", "Ireland", "Scotland")) {
    return("Europe")
  } else {
    return("etc")
  }
}

# create "A_Continent" column
arrival$A_Continent <- sapply(arrival$Arrival_Country, map_continent)
# create "D_Continent" column
departure$D_Continent <- sapply(departure$Departure_Country, map_continent)

# check the work is done
nrow(arrival[arrival$A_Continent == "etc", ])
nrow(departure[departure$D_Continent == "etc", ])
```

For Continent, we checked unique countries, created a dataframe to rename unique Korean country names to English, and converted the country name into English with 'left_join()'. Then, we defined a function to map country and continent and applied it to both of Country columns.

```
# second, day of week column
arrival$A_Day_of_Week <- format(arrival$A_Flight_Date, "%a")
departure$D_Day_of_Week <- format(departure$D_Flight_Date, "%a")

# check new columns are added well
head(arrival)
head(departure)
```

Arrival_Country <chr>	Arrival_Airport <chr>	A_Airport_Code <chr>	A_Flight_Date <date>	A_Scheduled_Time <hms>	A_Actual_Time <hms>	A_Flight_Name <chr>	A_Transfer_Passengers <dbl>	A_Continent <chr>	A_Day_of_Week <chr>
Vietnam	두옹 닷(푸꿕)	PQC	2023-01-01	00:30:00	01:01:00	VJ978	0	SouthEast Asia	일
Philippines	매닐라	MNL	2023-01-01	04:10:00	03:36:00	7C2306	5	SouthEast Asia	일
Thailand	방콕/수완나품	BKK	2023-01-01	04:20:00	04:05:00	KE658	15	SouthEast Asia	일
Vietnam	나트랑	CXR	2023-01-01	04:30:00	04:24:00	VJ836	0	SouthEast Asia	일
Japan	도쿄/하네다	HND	2023-01-01	04:35:00	04:47:00	MM809	0	East Asia	일
Philippines	매닐라	MNL	2023-01-01	04:45:00	03:59:00	KE624	99	SouthEast Asia	일

Figure 5: Arrival Dataset - Created Columns (see code chunk 7)

Departure_Country <chr>	Departure_Airport <chr>	D_Airport_Code <chr>	D_Flight_Date <date>	D_Scheduled_Time <hms>	D_Actual_Time <hms>	D_Flight_Name <chr>	D_Transfer_Passengers <dbl>	D_Continent <chr>	D_Day_of_Week <chr>
Turkey	이스탄불	IST	2023-01-01	00:15:00	00:19:00	TK091	0	West/South Asia	일
Qatar	도하	DOH	2023-01-01	00:25:00	00:40:00	QR859	2	West/South Asia	일
Ethiopia	볼레(아디스아바바)	ADD	2023-01-01	00:30:00	00:38:00	ET673	4	Africa	일
Philippines	매닐라	MNL	2023-01-01	00:40:00	00:57:00	5J187	0	SouthEast Asia	일
Netherlands	암스테르담	AMS	2023-01-01	01:25:00	01:46:00	KL862	9	Europe	일
Vietnam	두옹 닷(푸꿕)	PQC	2023-01-01	01:45:00	01:48:00	VJ975	0	SouthEast Asia	일

Figure 6: Departure Dataset - Created Columns (see code chunk 7)

You can see the columns are well processed and created.

3. Basic Investigations & Insights

From now on, we will show our basic investigations and insights from them to get some ideas for our hypotheses. For basic statistics, we did visualizations on 3 kinds of dataset: arrival, departure, and total(arrival+departure merged), to see and compare each data's trends and patterns. However, the pattern of both datasets was almost identical to that of total dataset.

Therefore, we focused on presenting the total dataset's result at the presentation day, but we're going to prove the fact that they showed similar patterns now in this report.

[The Number of Transfer Passengers over Time]

```
# Number of Transfer Passengers by Date (Arrival)
a_daily_transfers <- arrival %>%
  group_by(A_Flight_Date) %>%
  summarise(Total_Transfers = sum(A_Transfer_Passengers))

ggplot(a_daily_transfers, aes(x = A_Flight_Date, y = Total_Transfers)) +
  geom_line(color = "#0073C2FF") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = "A: Number of Transfer Passengers by Date", x = "Date", y = "Number of Transfer Passengers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(labels = scales::comma)

# Number of Transfer Passengers by Date (Departure)
d_daily_transfers <- departure %>%
  group_by(D_Flight_Date) %>%
  summarise(Total_Transfers = sum(D_Transfer_Passengers))

ggplot(d_daily_transfers, aes(x = D_Flight_Date, y = Total_Transfers)) +
  geom_line(color = "#0073C2FF") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = "D: Number of Transfer Passengers by Date", x = "Date", y = "Number of Transfer Passengers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(labels = scales::comma)

# Number of Transfer Passengers by Date (Total)
# join arrival and departure transfer passengers by date

# i) rename date columns to merge by same column name
```

```
names(d_daily_transfers)[names(d_daily_transfers) == "D_Flight_Date"] <- "Flight_Date"
names(a_daily_transfers)[names(a_daily_transfers) == "A_Flight_Date"] <- "Flight_Date"

# ii) merge d_daily_transfers and a_daily_transfers and calculate total transfers
total_daily_transfers <- full_join(d_daily_transfers, a_daily_transfers, by = "Flight_Date") %>%
  mutate(Total_Transfers = Total_Transfers.x + Total_Transfers.y)

# iii) visualize total daily transfers
ggplot(total_daily_transfers, aes(x = Flight_Date, y = Total_Transfers)) +
  geom_line(color = "#0073C2FF") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = "Total Transfer Passengers by Date", x = "Date", y = "Total Transfer Passengers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(labels = scales::comma)
```

For total, we merged the arrival and departure with using 'full_join()', to calculate total transfer passengers by date.

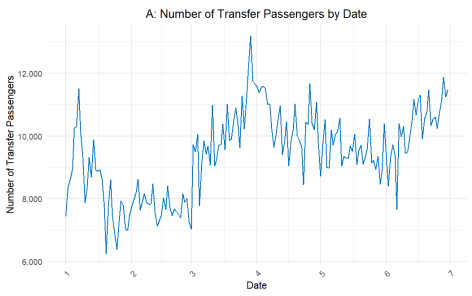

Figure 7: Arrival - Transfer Passengers by Date (see code chunk 8)

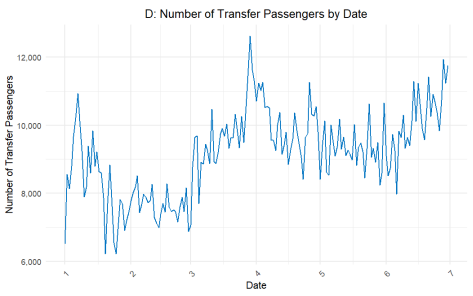

Figure 8: Departure - Transfer Passengers by Date (see code chunk 8)

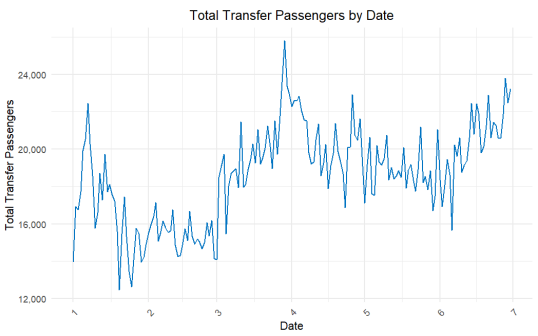

Figure 9: Total Transfer Passengers by Date (see code chunk 8)

<Finding>
From mid-January to February, it continuously decreased, then started to increase in March, reaching a peak at the end of March. It started to increase again in June. To sum up, there is a tendency to increase in the warming spring and in June when summer begins.

[Trend and Moving Average of Weekly Transfer Passengers]

However, we wanted to see a clear tendency whether it is increasing or decreasing over the entire period. So we drew a additional graph of weekly moving average using 'rollmean()'and linear regression model using 'lm()'.

```
# install and import a package for rollmean()
install.packages("zoo")
library(zoo)

# Trend and Moving Average of Weekly Transfer Passengers (Arrival)

# aggregate data by week to reduce density
arrival_weekly <- arrival %>%
  mutate(week = as.Date(cut(A_Flight_Date, breaks = "week"))) %>%
  group_by(week) %>%
  summarize(weekly_passengers = sum(A_Transfer_Passengers))

# calculate moving average
arrival_weekly$moving_avg <- rollmean(arrival_weekly$weekly_passengers, 4, fill = NA)

# fit linear regression model
model_a <- lm(weekly_passengers ~ week, data = arrival_weekly)
summary(model_a)

# plot the data with moving average and trend
ggplot(arrival_weekly, aes(x = week, y = weekly_passengers)) +
  geom_line() +
  geom_line(aes(y = moving_avg), color = "blue") +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "red") +
  labs(title = "A: Trend and Moving Average of Weekly Transfer Passengers", y = "Number of Passengers")

# Trend and Moving Average of Weekly Transfer Passengers (Departure)

# same steps as above
departure_weekly <- departure %>%
  mutate(week = as.Date(cut(D_Flight_Date, breaks = "week"))) %>%
```

```
  group_by(week) %>%
  summarize(weekly_passengers = sum(D_Transfer_Passengers))

departure_weekly$moving_avg <- rollmean(departure_weekly$weekly_passengers, 4, fill = NA)

model_d <- lm(weekly_passengers ~ week, data = departure_weekly)
summary(model_d)

ggplot(departure_weekly, aes(x = week, y = weekly_passengers)) +
  geom_line() +
  geom_line(aes(y = moving_avg), color = "blue") +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "red") +
  labs(title = "Trend and Moving Average of Weekly Transfer Passengers", y = "Number of Passengers")

# Trend and Moving Average of Weekly Transfer Passengers (Total)

# same steps as above
total_weekly_transfers <- total_daily_transfers %>%
  mutate(week = as.Date(cut(Flight_Date, breaks = "week"))) %>%
  group_by(week) %>%
  summarize(weekly_total_transfers = sum(Total_Transfers))

total_weekly_transfers$moving_avg <- zoo::rollmean(total_weekly_transfers$weekly_total_transfers, 4, fill = NA)

model_t <- lm(weekly_total_transfers ~ week, data = total_weekly_transfers)
summary(model_t)

ggplot(total_weekly_transfers, aes(x = week, y = weekly_total_transfers)) +
  geom_line() +
  geom_line(aes(y = moving_avg), color = "blue") +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "red") +
  labs(title = "Total Transfers: Trend and Moving Average of Weekly Transfer Passengers", y = "Number of Passengers")
```

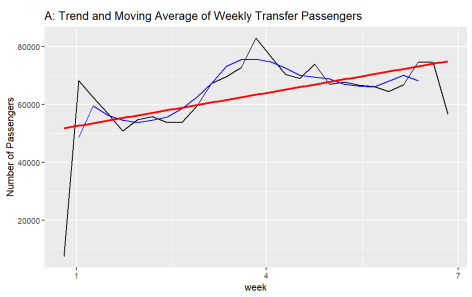

Figure 10: Arrival - Trend and Moving Average of Weekly Transfer Passengers (see code chunk 9)

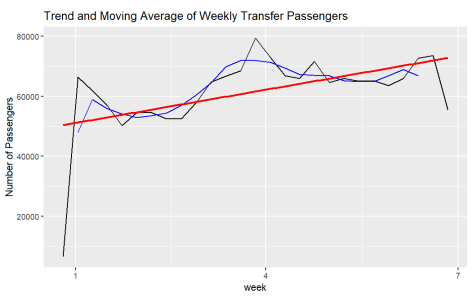

Figure 11: Departure - Trend and Moving Average of Weekly Transfer Passengers (see code chunk 9)

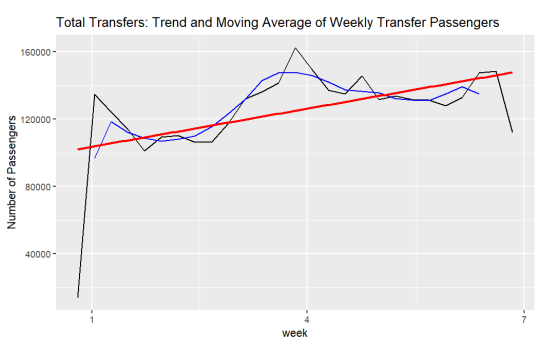

Figure 12: Total Trend and Moving Average of Weekly Transfer Passengers (see code chunk 9)

<Finding>
Each of the linear regression model's p-value was 0.00623, 0.00567, and 0.00591, which means the results are significant. We can say that as June comes, the number of transfer passengers tends to increase.

[The Number of Transfer Passengers by Country: World Map]

From now on, we are moving to countries and continents information.

First of all, to see overall picture of them, we drew world map with the number of transfer passengers by country.

```
# install and import packages needed for visualizing world map
install.packages("sf")
install.packages("rnaturalearth")
install.packages("rnaturalearthdata")
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)

# The Number of Transfer Passengers by Country (Arrival)

# summarize the number of transfer passengers by country (arrange in descending order for top 10 barplots below code chunk 13)
a_country_transfers <- arrival %>%
  group_by(Arrival_Country) %>%
  summarise(Total_Transfers = sum(A_Transfer_Passengers)) %>%
  arrange(desc(Total_Transfers))
```

```
# load world map data
world <- ne_countries(scale = "medium", returnclass = "sf")

# merge the transfer passenger data with world map data
world_a <- world %>%
  left_join(a_country_transfers, by = c("name" = "Arrival_Country"))

# set Total_Transfers to 0 if it is NA
world_a$Total_Transfers[is.na(world_a$Total_Transfers)] <- 0

# visualization
ggplot(data = world_a) +
  geom_sf(aes(fill = Total_Transfers)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "grey50", labels=scales::comma) +
  labs(title = "Transfer Passengers by Country (Arrival)",
       fill = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank())

# The Number of Transfer Passengers by Country (Departure)

# same steps as above
d_country_transfers <- departure %>%
  group_by(Departure_Country) %>%
  summarise(Total_Transfers = sum(D_Transfer_Passengers)) %>%
  arrange(desc(Total_Transfers))

world <- ne_countries(scale = "medium", returnclass = "sf")

world_d <- world %>%
  left_join(d_country_transfers, by = c("name" = "Departure_Country"))

world_d$Total_Transfers[is.na(world_d$Total_Transfers)] <- 0

ggplot(data = world_d) +
  geom_sf(aes(fill = Total_Transfers)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "grey50", labels=scales::comma) +
  labs(title = "Transfer Passengers by Country (Departure)",
       fill = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank())

# The Number of Transfer Passengers by Country (Total)

# join arrival+departure country transfer passenger
total_country_transfers <- full_join(a_country_transfers, d_country_transfers, by = c("Arrival_Country" = "Departure_Country")) %>%
  mutate(Total_Transfers = Total_Transfers.x + Total_Transfers.y) %>%
  select(Arrival_Country, Total_Transfers)

# from here, same steps as above
world <- ne_countries(scale = "medium", returnclass = "sf")

world_t <- world %>%
  left_join(total_country_transfers, by = c("name" = "Arrival_Country"))

world_t$Total_Transfers[is.na(world_t$Total_Transfers)] <- 0

ggplot(data = world_t) +
  geom_sf(aes(fill = Total_Transfers)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "grey50", labels=scales::comma) +
  labs(title = "Total Transfer Passengers by Country",
       fill = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank())
```

Using new packages like 'sf', 'rnaturalearth', and' rnaturalearthdata', we loaded world map data and merged them with the transfer passenger data. For total transfers, we also merged arrival and departure transfer passengers with 'full_join()'.

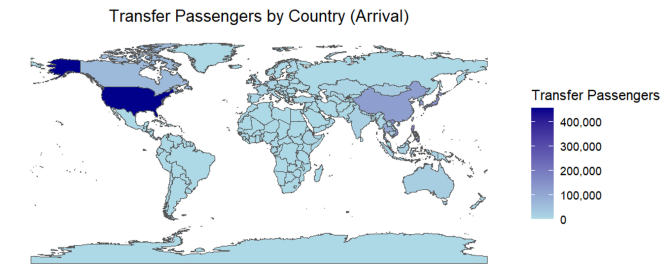

Figure 13: Transfer Passengers by Country (Arrival) (see code chunk 10)

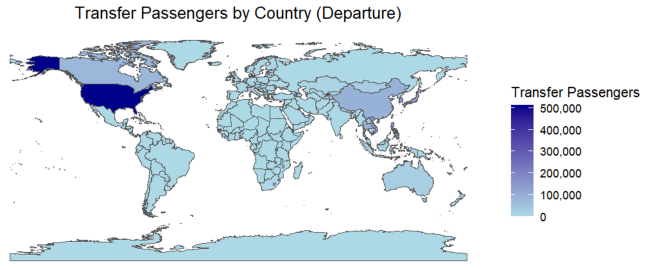

Figure 14: Transfer Passengers by Country (Departure) (see code chunk 10)

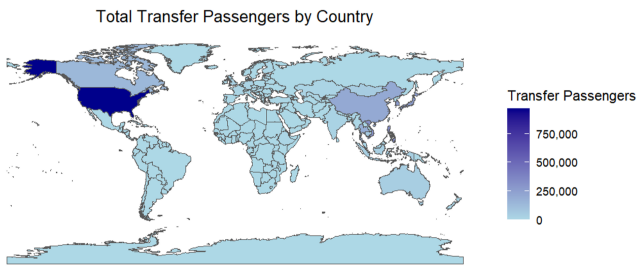

Figure 15: Total Transfer Passengers by Country (see code chunk 10)

As you can see, North America and Asia seem to be significant in all of our dataset. So, we drew more specific maps of Asia and America.

```
# filter the world map into Asia and America (Total)

# i) Asia
asia_bbox <- st_bbox(c(xmin = 60, ymin = -10, xmax = 150, ymax = 60), crs = st_crs(world_t))
asia_t <- st_crop(world_t, asia_bbox)

ggplot(data = asia_t) +
  geom_sf(aes(fill = Total_Transfers)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "grey50", labels=scales::comma) +
  labs(title = "Total Transfer Passengers by Country (Asia)",
       fill = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank())

# ii) America
america_bbox <- st_bbox(c(xmin = -170, ymin = -60, xmax = -30, ymax = 80), crs = st_crs(world_t))
america_t <- st_crop(world_t, america_bbox)

ggplot(data = america_t) +
  geom_sf(aes(fill = Total_Transfers)) +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "grey50", labels=scales::comma) +
  labs(title = "Total Transfer Passengers by Country (America)",
       fill = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid = element_blank())
```

We filtered Asia by defining a bounding box(asia_bbox) for Asia with coordinates ranging from 60°E to 150°E longitude and -10°S to 60°N latitude. And using st_crop, we cropped the world_t spatial data to this bounding box, resulting in asia_t. Similary, for America, we defined a bounding box(america_bbox) for America with coordinates ranging from -170°W to -30°W longitude and -60°S to 80°N latitude. The world_t spatial data was cropped to this bounding box, resulting in america_t.

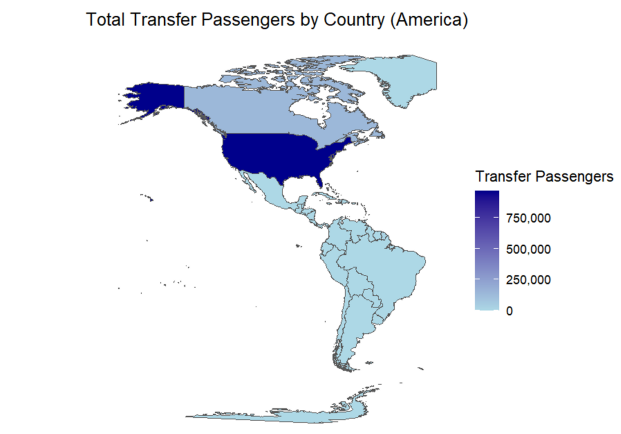

Figure 16: Total Transfer Passengers by Country (America) (see code chunk 11)

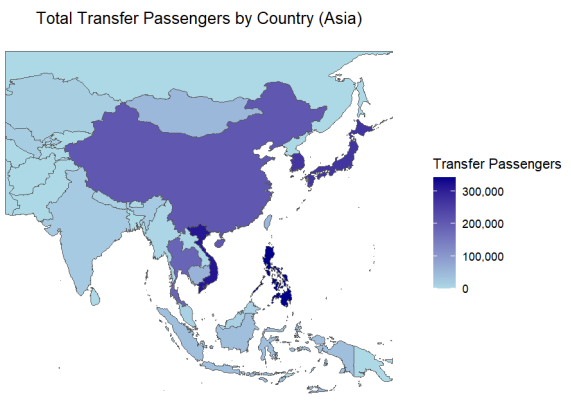

Figure 17: Total Transfer Passengers by Country (Asia) (see code chunk 11)

As a result, for America, the United States seemed to have the highest number of passengers. And for Asia, especially East and Southeast Asian countries seemed to show high numbers.

How will it be when we see them as barplots of both arrival and departure? We also drew barplots to see striking continents by the exact number of transfer passengers, and calculated the ratio of each continent.

[Arrival/Departure Continents and their Ratio by the number of transfer passengers]

```
# Transfer Passengers by Continent (arrival)
a_continent_transfers <- arrival %>%
  group_by(A_Continent) %>%
  summarise(Total_Transfers = sum(A_Transfer_Passengers)) %>%
  arrange(desc(Total_Transfers))

# print the exact number
a_continent_transfers

# transfer passenger ratio by continent
a_continent_transfers <- a_continent_transfers %>%
  mutate(Total_Ratio = (Total_Transfers / sum(a_continent_transfers$Total_Transfers)) * 100)

# visualization
ggplot(a_continent_transfers, aes(x = reorder(A_Continent, -Total_Transfers), y = Total_Transfers, fill = A_Continent)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(Total_Ratio, 1), "%")), vjust = -0.3, color = "black") +
  labs(title = "Arrival Continents by Transfer Passengers", x = "Continent", y = "Transfer Passengers") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))

# Transfer Passengers by Continent (departure)
d_continent_transfers <- departure %>%
  group_by(D_Continent) %>%
  summarise(Total_Transfers = sum(D_Transfer_Passengers)) %>%
  arrange(desc(Total_Transfers))

# print the exact number
d_continent_transfers

# transfer passenger ratio by continent
d_continent_transfers <- d_continent_transfers %>%
  mutate(Total_Ratio = (Total_Transfers / sum(d_continent_transfers$Total_Transfers)) * 100)

# visualization
ggplot(d_continent_transfers, aes(x = reorder(D_Continent, -Total_Transfers), y = Total_Transfers, fill = D_Continent)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(Total_Ratio, 1), "%")), vjust = -0.3, color = "black") +
  labs(title = "Departure Continents by Transfer Passengers", x = "Continent", y = "Transfer Passengers") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

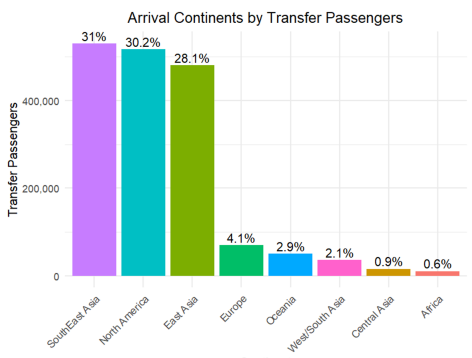

Figure 18: Arrival Continents by Transfer Passengers (see code chunk 12)

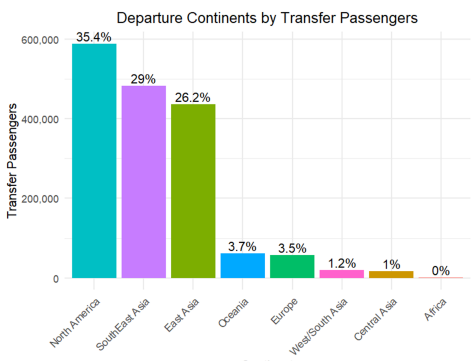

Figure 19: Departure Continents by Transfer Passengers (see code chunk 12)

Through the bar plot, we could exactly know which continent to focus on. There were three noticeable continents, which was quite similar to the result of the previous world maps.

In both arrival and departure, Southeast Asia, North America, East Asia accounted for about ninety percent. Other continents were very low. In arrival(a_continent_transfers), while the three continents each exceeded about 500,000 passengers, Europe had 70,000, Oceania had 50,000, and the rest ranged from 10,000 to 40,000.

Then, exactly which country was the most? To have more specific findings, we drew the top 10 countries by the transfer passengers with both arrival and departure.

[Top 10 Arrival/Departure Countries by the number of transfer passengers]

```
# transfer passenger ratio by country (arrival)
a_country_transfers <- a_country_transfers %>%
  mutate(Total_Ratio = (Total_Transfers / sum(a_country_transfers$Total_Transfers)) * 100) %>%
  top_n(10, Total_Transfers) # only top 10 countries

# visualization ('a_country_transfers' is already calculated above code chunk 10)
ggplot(a_country_transfers, aes(x = reorder(Arrival_Country, -Total_Transfers), y = Total_Transfers, fill = Arrival_Country)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(Total_Ratio, 1), "%")), vjust = -0.3, color = "black") +
  labs(title = "Top 10 Arrival Countries by Transfer Passengers", x = "Arrival Country", y = "Transfer Passengers") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
```

```
        plot.title = element_text(hjust = 0.5))

# transfer passenger ratio by country (departure)
d_country_transfers <- d_country_transfers %>%
  mutate(Total_Ratio = (Total_Transfers / sum(d_country_transfers$Total_Transfers)) * 100) %>%
  top_n(10, Total_Transfers) # only top 10 countries

# visualization ('d_country_transfers' is already calculated above code chunk 10)
ggplot(d_country_transfers, aes(x = reorder(Departure_Country, -Total_Transfers), y = Total_Transfers, fill = Departure_Country)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(Total_Ratio, 1), "%")), vjust = -0.3, color = "black") +
  labs(title = "Top 10 Departure Countries by Transfer Passengers", x = "Departure Country", y = "Total Transfer Passengers", fill = "D
eparture Country") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

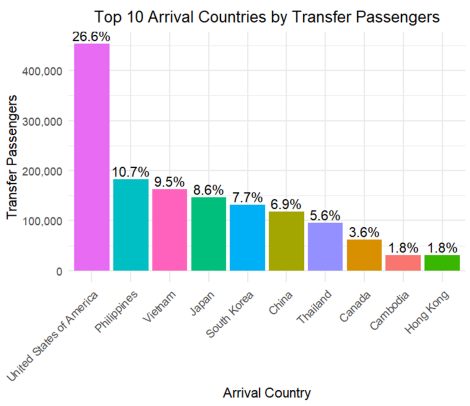

Figure 20: Top 10 Arrival Countries by Transfer Passengers (see code chunk 13)

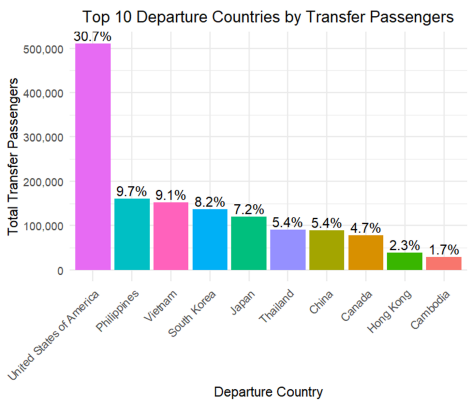

Figure 21: Top 10 Departure Countries by Transfer Passengers(see code chunk 13)

To see the barplots, the United States had an exclusively high number of transfer passengers. Southeast Asian countries like the Philippines and Vietnam followed the US, and East Asian countries like Japan and China followed them. We found that both plots have the same countries with a little difference in rank. Also, one interesting fact was that all the top 10 countries in both datasets were the countries in the top 3 continents.

[Continents of Top 10 Arrival/Departure Countries]

At this point, we drew donut charts to see the relative ratio of each continent.

```
# define continents of top 10 countries
continents <- list(
  'North America' = c('United States of America', 'Canada'),
  'East Asia' = c('China', 'Japan', 'South Korea', 'Hong Kong'),
  'SouthEast Asia' = c('Philippines', 'Vietnam', 'Thailand', 'Cambodia')
)

# aggregate percentages by continent (arrival)
continent_data_a <- a_country_transfers %>%
  mutate(Continent = case_when(
    Arrival_Country %in% continents[['North America']] ~ 'North America',
    Arrival_Country %in% continents[['East Asia']] ~ 'East Asia',
    Arrival_Country %in% continents[['SouthEast Asia']] ~ 'SouthEast Asia'
  )) %>%
  group_by(Continent) %>%
  summarise(Total_Percentage = sum(Total_Ratio))

# create donut chart
ggplot(continent_data_a, aes(x = 2, y = Total_Percentage, fill = Continent)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar(theta = "y") +
  xlim(0.5, 2.5) +
  theme_void() +
  theme(legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14)) +
  geom_text(aes(label = paste0(round(Total_Percentage, 1), "%")),
            position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("North America" = "#66c2a5", "East Asia" = "#fc8d62", "SouthEast Asia" = "#8da0cb")) +
  labs(title = "The Continents of Top 10 Countries (Arrival)")

# aggregate percentages by continent (arrival)
continent_data_d <- d_country_transfers %>%
  mutate(Continent = case_when(
    Departure_Country %in% continents[['North America']] ~ 'North America',
    Departure_Country %in% continents[['East Asia']] ~ 'East Asia',
    Departure_Country %in% continents[['SouthEast Asia']] ~ 'SouthEast Asia'
  )) %>%
  group_by(Continent) %>%
  summarise(Total_Percentage = sum(Total_Ratio))

# create the donut chart
```

```
ggplot(continent_data_d, aes(x = 2, y = Total_Percentage, fill = Continent)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar(theta = "y") +
  xlim(0.5, 2.5) +
  theme_void() +
  theme(legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5, size = 14)) +
  geom_text(aes(label = paste0(round(Total_Percentage, 1), "%")),
            position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("North America" = "#66c2a5", "East Asia" = "#fc8d62", "SouthEast Asia" = "#8da0cb")) +
  labs(title = "The Continents of Top 10 Countries (Departure)")
```

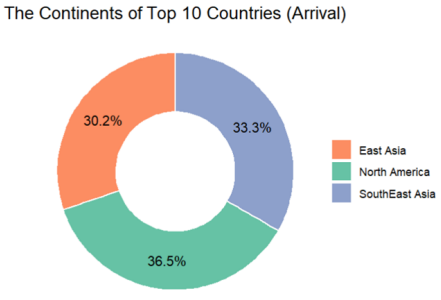

Figure 22: The Continents of Top 10 Countries (Arrival) (see code chunk 14)

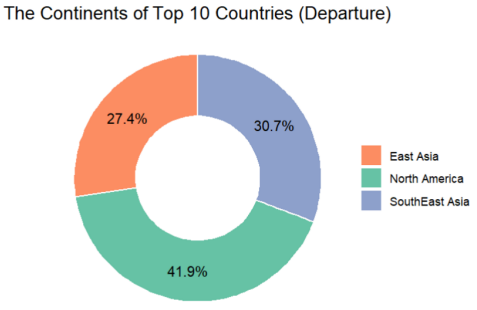

Figure 23: The Continents of Top 10 Countries (Departure) (see code chunk 14)

The 3 continents were ranked in order of North America, Southeast Asia, and East Asia. We were sure that it will be meaningful for us to concentrate on these continents during our analysis.

[The Number of Arrival Flights by Continent over Time]

Then, how about the number of flights to the continents over time? Now we are handling the flight frequencies arriving to the noticeable 3 continents. Since it is about frequency, flights with zero transfer passengers were excluded to prevent distorted results.

```
# extract flights with non-zero transfer passengers
non_zero_transfers <- arrival %>%
  filter(A_Transfer_Passengers != 0)

# calculate monthly arrival frequency by continent
monthly_continent_counts_nonzero <- non_zero_transfers %>%
  mutate(Month = format(A_Flight_Date, "%m")) %>%
  group_by(Month, A_Continent) %>%
  summarise(Count = n(), .groups = "drop")

# visualization
ggplot(monthly_continent_counts_nonzero, aes(x = Month, y = Count, fill = A_Continent)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Arrival Flights by Continent per Month (Excluding Flights with 0 Transfers)", x = "Month", y = "Flights", fill = "Conti
nent") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

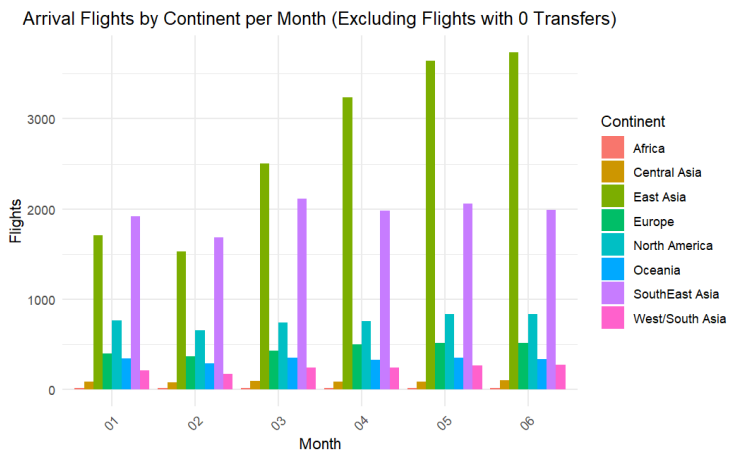

Figure 24: Arrival Flights by Continent per Month (see code chunk 15)

From January to June, the number of flights arriving in Southeast Asia and North America seems consistent. Rather, there was a notable increase in the arrival flights to East Asia. Then, which countries would have contributed to East Asia's increase?

[Arrival Flights to East Asia by Month]

```
# define a vector of east asian countries
east_asia_countries <- c("South Korea", "China", "Japan", "Hong Kong", "Mongolia", "Taiwan", "Macau")

# select only rows corresponding to east asian countries
east_asia_data <- non_zero_transfers %>%
  filter(Arrival_Country %in% east_asia_countries) %>%
  mutate(Month = format(A_Flight_Date, "%m"))

# calculate arrival frequency by month
monthly_east_asia_counts <- east_asia_data %>%
  group_by(Month, Arrival_Country) %>%
  summarise(Count = n(), .groups = "drop")
```

```
# visualization
ggplot(monthly_east_asia_counts, aes(x = Month, y = Count, fill = Arrival_Country)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Arrival Flights to East Asia by Month (Excluding Flights with 0 Transfers)", x = "Month", y = "Flights") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Figure 25: Arrival Flights to East Asia by Month (see code chunk 16)

We can see that most of the flights to East Asia arrived in Japan across the months, and arrivals to China increased rapidly over time.

4. Hypotheses and Verification

[Insights from previous visualizations]
So far, we could get some insights from previous visualizations.

First, the number of transfer passengers at Incheon Airport has increased over time.
Second, the most popular/frequent countries and continents were North America, South Asia, and East Asia.
Lastly, however, the number of flights showed a distinct change only in East Asia.

[Our Ideas for Hypotheses]
With this background, we could get an idea of our hypotheses.

First, The United States showed very high numbers of passengers. However, it is quite different from our common sense that the number of passengers will decrease as the distance between the country and the Incheon becomes further. Then, how will the number of transfer passengers change as the distance increases? Will it be less or more? (hypothesis 3)

Second, Southeast Asia was also remarkable. Considering the increase of transfer passengers as summer comes, we can assume that the number of transfer passengers going to Southeast Asia is expected to increase, potentially due to the summer resorts and closeness to Incheon. (hypothesis 1)

Third, East Asia was also ranked in the top continents. We can imagine that the number of transfer passengers to East Asia will be highest in winter due to the influence of the Chinese New Year. (hypothesis 2)

[Our Hypotheses 1-3]
To summarize, our hypotheses are as follows.

1. As summer comes, the number of transfer passengers arriving in Southeast Asia would have increased.
2. For the high number of transfer passengers in January, East Asia contributed significantly due to China's New Year holiday.
3. The further the distance from the transfer airport, the greater the number of transit passengers will be there.

We are going to verify them from now on.

[EDA Hypothesis 1]

1. As summer comes, the number of transfer passengers arriving in Southeast Asia would have increased.

```
# Install and import library
install.packages("gridExtra")
library(gridExtra)
library(scales)

# Define the seasons with start and end dates
seasons <- data.frame(
  season = c("Winter", "Spring", "Summer"),
  start = as.Date(c("2023-01-01", "2023-03-01", "2023-06-01")),
  end = as.Date(c("2023-02-28", "2023-05-31", "2023-07-31"))
)


# Function to classify the season based on date
classify_season <- function(date) {
  if (date >= as.Date("2023-01-01") & date <= as.Date("2023-02-28")) {
    return("Winter")
  } else if (date >= as.Date("2023-03-01") & date <= as.Date("2023-05-31")) {
    return("Spring")
  } else if (date >= as.Date("2023-06-01") & date <= as.Date("2023-07-31")) {
    return("Summer")
  } else {
    return(NA)
  }
}
```

```
# Add the season column to the arrival data frame
arrival <- arrival %>%
  mutate(season = sapply(A_Flight_Date, classify_season))
```

```
# Transfer Passengers Arriving in South Asia over time

# Define the `a_daily_passenger_count_se_asia` DataFrame for South Asia
a_daily_passenger_count_se_asia <- arrival %>%
  filter(A_Continent == "SouthEast Asia") %>%
  group_by(A_Flight_Date, season) %>%
  summarise(Total_Passengers = sum(A_Transfer_Passengers))

# Colors for the seasons
season_colors <- c("Winter" = "lightblue", "Spring" = "lightgreen", "Summer" = "lightyellow")

# Plot 1: South Asia
plot1 <- ggplot(a_daily_passenger_count_se_asia, aes(x = A_Flight_Date, y = Total_Passengers, color = season)) +
  geom_rect(data = seasons, aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf, fill = season), alpha = 0.1, inherit.aes = FALSE) +
  geom_line(aes(color = season)) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "darkblue", linetype = "dashed") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month", limits = as.Date(c("2023-01-01", "2023-06-30"))) +
  labs(title = "Number of Transfer Passengers, South Asia by Month",
       x = "Date", y = "Transfer Passengers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, face = "bold"),
        plot.title = element_text(hjust = 0.5, size = 10),
        axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        legend.title = element_text(size = 8)) +
  scale_fill_manual(values = season_colors) +
  scale_color_manual(values = c("Winter" = "blue", "Spring" = "green", "Summer" = "orange")) +
  guides(fill = guide_legend(title = "Season"), color = guide_legend(title = "Passenger Season"))

plot1

# Explanation:
# 1. A ggplot object 'plot1' is created for arrival passengers from South Asia.
# 2. Shaded regions indicate different seasons.
# 3. The plot shows the total number of transfer passengers from South Asia over time, with a regression line to indicate trends.
# 4. Custom themes and colors are applied to enhance the visual representation.
```

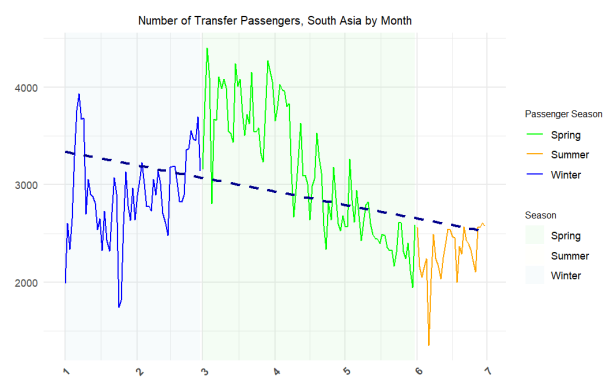

Figure 26: Number of Transfer Passengers to South Asia by Month (see code chunk 18)

When we looked at the previous pie(donut) chart, we noticed that South Asia has a larger portion than East Asia. This got us thinking about summer vacations in South Asia and how they might affect transfer passengers and flight numbers. We also considered the idea of short winter vacations, and that's when we decided to take a closer look. The graph shows the number of transfer passengers in South Asia for each month. At first glance, we can see a negative relationship between the date and the number of transferring passengers related to South Asia.

That means, there was a noticeable decrease in transfer passengers from winter to summer. We also noticed significant fluctuations in winter, with a peak in early March and April in spring and then a noticeable drop in early June, summer, possibly due to the beginning of the monsoon season. So, we realized our initial hypothesis should be off. Monsoon season might make people deter their travel plans due to heavy rainfall in South Asia and potential disruptions because of that.

[EDA Hypothesis 2]

2. For the high number of transfer passengers in January, East Asia contributed significantly due to China's New Year holiday.

```
# Transfer Passengers Arriving in East Asia over time

# Define the `a_daily_passenger_count_east_asia` DataFrame for East Asia
a_daily_passenger_count_east_asia <- arrival %>%
  filter(A_Continent == "East Asia") %>%
  group_by(A_Flight_Date, season) %>%
  summarise(Total_Passengers = sum(A_Transfer_Passengers))

# Plot 2: East Asia
plot2 <- ggplot(a_daily_passenger_count_east_asia, aes(x = A_Flight_Date, y = Total_Passengers, color = season)) +
  geom_rect(data = seasons, aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf, fill = season), alpha = 0.1, inherit.aes = FALSE) +
  geom_line(aes(color = season)) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "darkblue", linetype = "dashed") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month", limits = as.Date(c("2023-01-01", "2023-06-30"))) +
  labs(title = "Number of Transfer Passengers, East Asia by Month",
       x = "Date", y = "Transfer Passengers") +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, face = "bold"),
        plot.title = element_text(hjust = 0.5, size = 10),
        axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        legend.title = element_text(size = 8)) +
  scale_fill_manual(values = season_colors) +
  scale_color_manual(values = c("Winter" = "blue", "Spring" = "green", "Summer" = "orange")) +
  guides(fill = guide_legend(title = "Season"), color = guide_legend(title = "Passenger Season"))

plot2

# Explanation:
# 1. A ggplot object 'plot2' is created for arrival passengers from East Asia.
# 2. Shaded regions indicate different seasons.
# 3. The plot shows the total number of transfer passengers from East Asia over time, with a regression line to indicate trends.
# 4. Custom themes and colors are applied to enhance the visual representation.
```

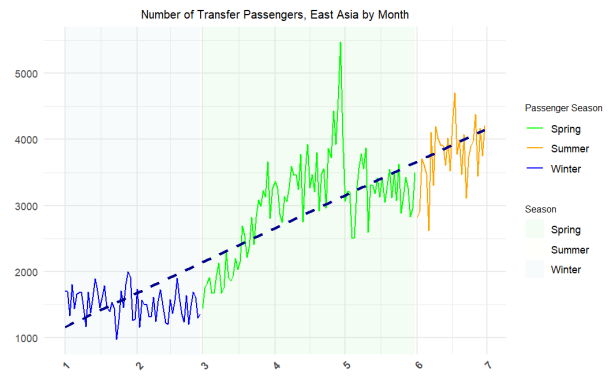

Figure 27: Number of Transfer Passengers to East Asia by Month (see code chunk 19)

Next, we looked at East Asia and saw a peak in mid-January on the total passengers' graph, which is shown above. We thought that a peak in mid-January on the total passenger graph might be due to China's New Year holiday. Especially considering that the Chinese New Year Holiday was in the middle of January in 2023, it takes us to verify where this peak exactly came from.

Looking at the graph first, we see a positive trend between the date and transfer passengers. That said, transfer passengers related to East Asia generally increased as the date went on and winter went to summer. So, what is the result of this?

As the trend shows, the hypothesis was completely rejected. We found a general increase as winter turned into summer, along with fluctuations and only a slight increase in mid-January in the winter season. A peak exists in late April of East Asia's Trending graphs, which is a little slower than South Asia's peak in Early March, and also the consistently increasing trends in Summer existed. However, we figured the trend of transfer passengers related to South Asia in the previous slide, and also on the above graph, might have contributed to the peak in January's trend on the overall total passenger trending graph.

[EDA Hypothesis 3]

3. The further the distance from the transfer airport, the greater the number of transit passengers will be there.

```
install.packages("airportr")

library(airportr)
library(tidyr)
```

<Using Function to Calculate Distance from Incheon Airport>

People usually think that airlines offer more transfer flights than direct flights considering the damage of a mismatch between supply and demand and the burden of distance. To see if this common belief is true, we hypothesized that 'the greater the distance from the airport, the greater the number of connections.'.
We try to find relationship between distance and the number of passengers. But the distance is not included in the dataset, it is necessary to calculate the distance from Incheon Airport to the departure and arrival airports one by one. This is not easy, so the distance is calculated one by one using the airport_distance function of the airportr package.

```
calculate_distance_to_icn <- function(airport_code) {
  tryCatch({
    distance <- airportr::airport_distance(airport_code, "ICN")
    return(distance)
  }, error = function(e) {
    return(NA)
  })
}

departure <- departure %>%
  rowwise() %>%
  mutate(
    distance_to_icn = calculate_distance_to_icn(D_Airport_Code))%>%
      ungroup()

arrival <- arrival %>%
  rowwise() %>%
  mutate(
    distance_to_icn = calculate_distance_to_icn(A_Airport_Code))%>%
  ungroup()
```

<Trend Graph between distance and transfer passengers in Departure all over the world>

The skyblue line means the trend line in the number of transit passengers by distance. In addition, the color of the dots was classified differently depending on the continent.

```
ggplot(departure,aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(D_Continent)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='skyblue')+
  labs(title="Transfer according to Distance")
```

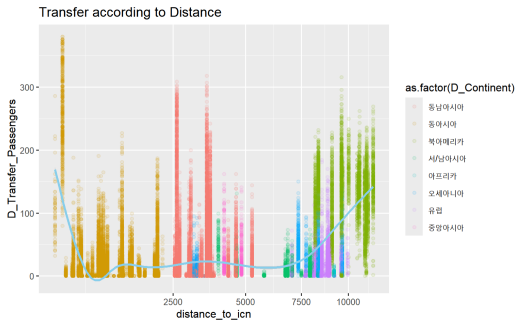

Figure 28: Trend Graph between distance and transfer passengers in Departure all over the world (see code chunk 22)

<Relation Graph between distance and transfer passengers in Departure all over the world>

The skyblue line means the linear relation in the number of transit passengers by distance.

```
ggplot(departure,aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(D_Continent)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method="lm",colour='skyblue')+
  labs(title="Transfer according to Distance")
```

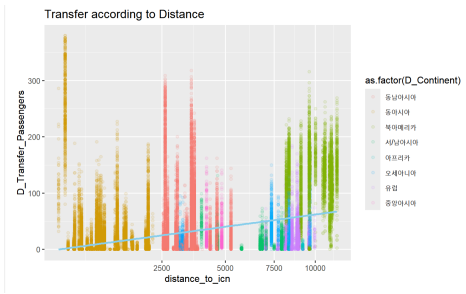

Figure 29: Relation Graph between distance and transfer passengers in Departure all over the world (see code chunk 23)

As the distance from Incheon Airport increases, the number of transit passengers increases by using linear regression.

```
library(DescTools)
```

<Summary for linear regression>

By the correlation test, relation between distance and the number of transfer is low.
(Only 15% represents the relation.)
If there seems to have relation, the relation seems to show positive.
P value is less than 0.05 therefore relationship is statistically significant (reject null hypothesis). Also PsueoR2 model is not a good fit to the data.

```
lm(distance_to_icn~D_Transfer_Passengers,data=departure)
PseudoR2(glm(distance_to_icn~D_Transfer_Passengers,data=departure))

summary(lm(distance_to_icn~D_Transfer_Passengers,data=departure))

cor.test(departure$distance_to_icn,departure$D_Transfer_Passengers)
```

```
Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = departure)

Coefficients:
          (Intercept)  D_Transfer_Passengers
               2773.06                 22.02

   McFadden
0.008908743

Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = departure)

Residuals:
     Min       1Q   Median       3Q      Max
-10801.7  -1912.5   -337.5    896.8   8673.6

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           2773.0557    12.1342   228.5   <2e-16 ***
D_Transfer_Passengers   22.0186     0.2093   105.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2651 on 60786 degrees of freedom
  (결측으로 인하여 826개의 관측치가 삭제되었습니다.)
Multiple R-squared:  0.154,     Adjusted R-squared:  0.154
F-statistic: 1.106e+04 on 1 and 60786 DF,  p-value: < 2.2e-16
```

Figure 30: Summary for linear regression1 (see code chunk 25)

```
        Pearson's product-moment correlation

data:  departure$distance_to_icn and departure$D_Transfer_Passengers
t = 105.18, df = 60786, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3856623 0.3991133
sample estimates:
      cor
0.3924088
```

Figure 31: Summary for linear regression2 (see code chunk 25)

<Relation between distance and the number of transit passengers in continent-by-continent departure>

We exclude the continent that have no relations.

```
ggplot(departure,aes(x=distance_to_icn,y=D_Transfer_Passengers))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth()+
  facet_wrap(~D_Continent)+
  labs(title="Transfer according to Distance",subtitle="Split by Continent")
```

The flow of the number of transfer passengers according to distance tends to be similar to that of East Asia, but the relationship of the number of transfer passengers according to distance tends to be similar to that of Europe and North America.

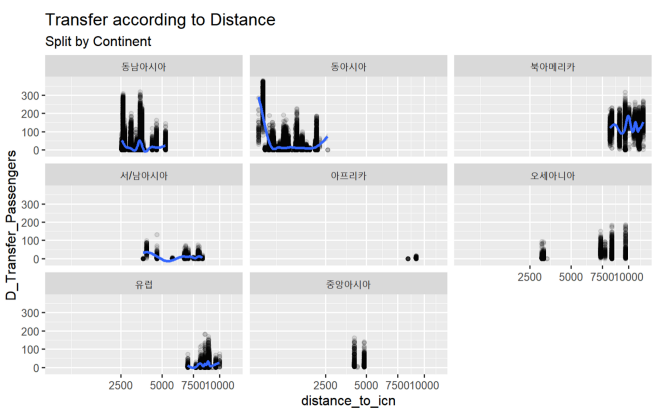

Figure 32: Relation between distance and the number of transit passengers in continent-by-continent departure (see code chunk 26)

<Trend between distance and the number of transit departure passengers in East Asia>

```
subset(departure,D_Continent=='East Asia')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='firebrick')+
  labs(title="Transfer according to Distance",subtitle="East Asia")
```

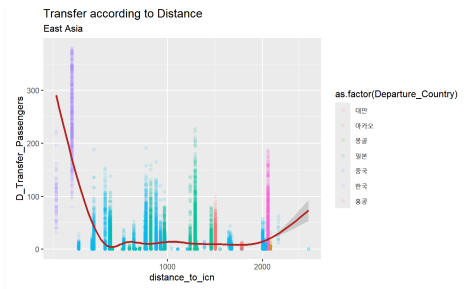

Figure 33: Trend between distance and the number of transit departure passengers in East Asia (see code chunk 27)

<Relation between distance and the number of transit departure passengers in East Asia>

```
subset(departure,D_Continent=='East Asia')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='firebrick')+
  labs(title="Transfer according to Distance",subtitle="East Asia")
```

It is similar to the flow of East Asia, but not similar to the relationship. This reveals that the linear regression of the distance and the number of transfer passengers does not fully represent the actual relationship.

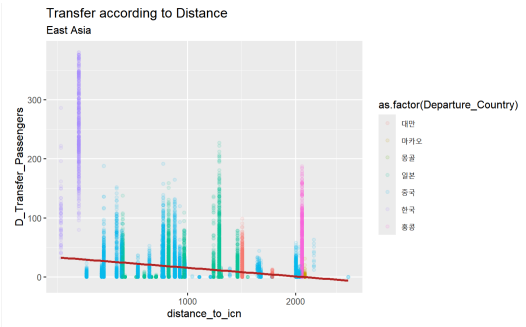

Figure 34: Relation between distance and the number of transit departure passengers in East Asia (see code chunk 28)

<Trend between distance and the number of transit departure passengers in North America>

```
subset(departure,D_Continent=='North America')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='Dark Slate Gray')+
  labs(title="Transfer according to Distance",subtitle="North America")
```

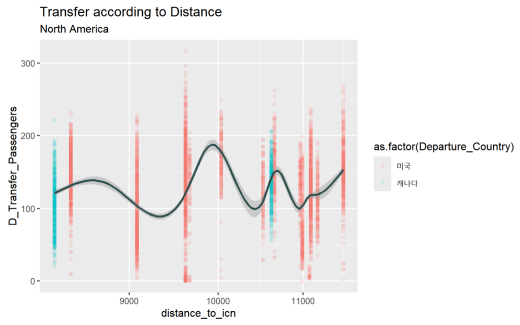

Figure 35: Trend between distance and the number of transit departure passengers in North America (see code chunk 29)

<Relation between distance and the number of transit departure passengers in North America>

```
subset(departure,D_Continent=='North America')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method = 'lm',colour='Dark Slate Gray')+
  labs(title="Transfer according to Distance",subtitle="North America")
```

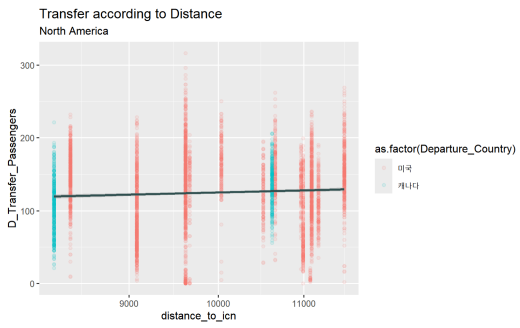

Figure 36: Relation between distance and the number of transit departure passengers in North America (see code chunk 30)

The flow between distances and transfer numbers around the world and that of North America are similar, also the relationship is similar. Through this, it can be seen that the flow of distances and transfer numbers around the world is similar to that of Asia and North America, which can be inferred that Asia and North America account for a significant proportion of the number of transfer numbers.

<Finding correlation between distance and the number of passengers in North America>

```
north_america_departure <- subset(departure, D_Continent == 'North America')
lm(distance_to_icn ~ D_Transfer_Passengers, data = north_america_departure)
summary(lm(distance_to_icn ~ D_Transfer_Passengers, data = north_america_departure))
```

```
Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = north_america_departure)

Coefficients:
          (Intercept)  D_Transfer_Passengers
             9793.968                 1.271


Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = north_america_departure)

Residuals:
    Min      1Q  Median      3Q     Max
-1872.5  -808.1  -166.7  1087.8  1694.2

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          9793.9677    40.5226 241.691  < 2e-16 ***
D_Transfer_Passengers    1.2708     0.2995   4.244 2.24e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1074 on 4713 degrees of freedom
Multiple R-squared:  0.003806,  Adjusted R-squared:  0.003595
F-statistic: 18.01 on 1 and 4713 DF,  p-value: 2.242e-05
```

Figure 37: Finding correlation between distance and the number of passengers in North America (see code chunk 31)

<Trend between distance and the number of transit departure passengers in Europe>

```
subset(departure,D_Continent=='Europe')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='Steel Blue')+
  labs(title="Transfer according to Distance",subtitle="Europe")
```

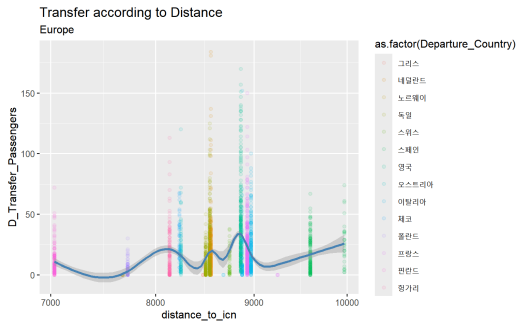

Figure 38: Trend between distance and the number of transit departure passengers in Europe (see code chunk 32)

<Relation between distance and the number of transit departure passengers in Europe>

```
subset(departure,D_Continent=='Europe')%>%
ggplot(aes(x=distance_to_icn,y=D_Transfer_Passengers,colour=as.factor(Departure_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='Steel Blue')+
  labs(title="Transfer according to Distance",subtitle="Europe")
```

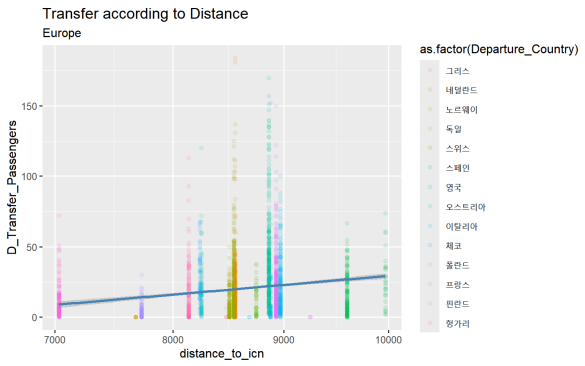

Figure 39: Relation between distance and the number of transit departure passengers in Europe (see code chunk 33)

<Finding correlation between distance and the number of passengers in Europe>

```
europe_departure <- subset(departure, D_Continent == 'Europe')
lm(distance_to_icn ~ D_Transfer_Passengers, data = europe_departure)
summary(lm(distance_to_icn ~ D_Transfer_Passengers, data = europe_departure))
```

```
Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = europe_departure)

Coefficients:
          (Intercept)  D_Transfer_Passengers
             8507.266                 4.475


Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = europe_departure)

Residuals:
     Min       1Q   Median       3Q      Max
-1795.12  -135.98    20.55   335.81  1452.11

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          8507.2664    13.9934  607.95   <2e-16 ***
D_Transfer_Passengers    4.4747     0.4671    9.58   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 556.1 on 2878 degrees of freedom
Multiple R-squared:  0.0309,    Adjusted R-squared:  0.03056
F-statistic: 91.77 on 1 and 2878 DF,  p-value: < 2.2e-16
```

Figure 40: Finding correlation between distance and the number of passengers in Europe (see code chunk 34)

```
install.packages("corrplot")
library(corrplot)
```

<Trend Graph between distance and transfer passengers in Arrival all over the world>

The skyblue line means the trend line in the number of transit passengers by distance. In addition, the color of the dots was classified differently depending on the continent.

```
ggplot(arrival,aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(A_Continent)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='skyblue')
```

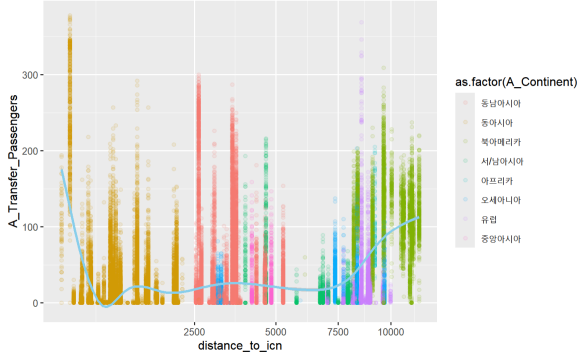

Figure 41: Trend Graph between distance and transfer passengers in Arrival all over the world (see code chunk 36)

<Relation Graph between distance and transfer passengers in Arrival all over the world>

The skyblue line means the linear relation in the number of transit passengers by distance.

```
ggplot(arrival,aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(A_Continent)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='skyblue')
```

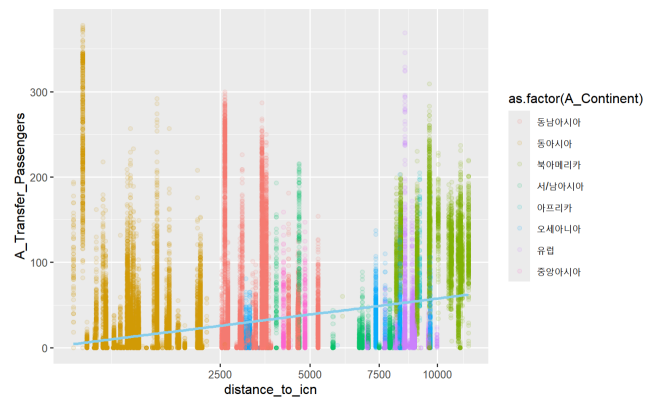

Figure 42: Relation Graph between distance and transfer passengers in Arrival all over the world (see code chunk 37)

<Summary for linear regression>

By the correlation test, relation between distance and the number of transfer is low.
(Only 33% represents the relation.)
If there seems to have relation, the relation seems to show positive.
P value is less than 0.05 therefore relationship is statistically significant (reject null hypothesis).

```
lm(distance_to_icn~A_Transfer_Passengers,data=arrival)
summary(lm(distance_to_icn~A_Transfer_Passengers,data=arrival))
glm(distance_to_icn~A_Transfer_Passengers,data=arrival)
PseudoR2(glm(distance_to_icn~A_Transfer_Passengers,data=arrival))

cor.test(arrival$distance_to_icn,arrival$A_Transfer_Passengers)
```

```
Call:
lm(formula = distance_to_icn ~ A_Transfer_Passengers, data = arrival)

Coefficients:
          (Intercept)  A_Transfer_Passengers
              2836.74                 18.93


Call:
lm(formula = distance_to_icn ~ A_Transfer_Passengers, data = arrival)

Residuals:
    Min      1Q  Median      3Q     Max
-9655.2 -1976.2  -385.7   833.1  8616.1

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           2836.7425    12.5107  226.75   <2e-16 ***
A_Transfer_Passengers   18.9334     0.2154   87.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2713 on 60847 degrees of freedom
  (결측으로 인하여 825개의 관측치가 삭제되었습니다.)
Multiple R-squared:  0.1126,    Adjusted R-squared:  0.1126
F-statistic:  7723 on 1 and 60847 DF,  p-value: < 2.2e-16
```

Figure 43: Summary for linear regression (see code chunk 38)

```
Call:  glm(formula = distance_to_icn ~ A_Transfer_Passengers, data = arrival)

Coefficients:
          (Intercept)  A_Transfer_Passengers
              2836.74                 18.93

Degrees of Freedom: 60848 Total (i.e. Null);  60847 Residual
  (결측으로 인하여 825개의 관측치가 삭제되었습니다.)
Null Deviance:      5.048e+11
Residual Deviance: 4.48e+11     AIC: 1135000
   McFadden
0.006366321

Call:
glm(formula = distance_to_icn ~ A_Transfer_Passengers, data = arrival)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           2836.7425    12.5107  226.75   <2e-16 ***
A_Transfer_Passengers   18.9334     0.2154   87.88   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 7362115)

    Null deviance: 5.0482e+11  on 60848  degrees of freedom
Residual deviance: 4.4796e+11  on 60847  degrees of freedom
  (결측으로 인하여 825개의 관측치가 삭제되었습니다.)
AIC: 1134822

Number of Fisher Scoring iterations: 2
```

Figure 44: Summary for linear regression 2 (see code chunk 38)

```
        Pearson's product-moment correlation

data:  arrival$distance_to_icn and arrival$A_Transfer_Passengers
t = 87.88, df = 60847, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3285311 0.3426325
sample estimates:
      cor
0.3356006
```

Figure 45: Summary for linear regression 3 (see code chunk)

<Relationship between distance and the number of transit passengers in continent-by-continent arrival>

We exclude the continent that have no relations.
The flow of the number of transfer passengers according to distance tends to be similar to that of East Asia, but the relationship of the number of transfer passengers according to distance tends to be similar to that of North America.

```
ggplot(arrival,aes(x=distance_to_icn,y=A_Transfer_Passengers))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth()+
```

```
  facet_wrap(~A_Continent)+
  labs(title="Transfer according to Distance",subtitle="Split by Continent")
```

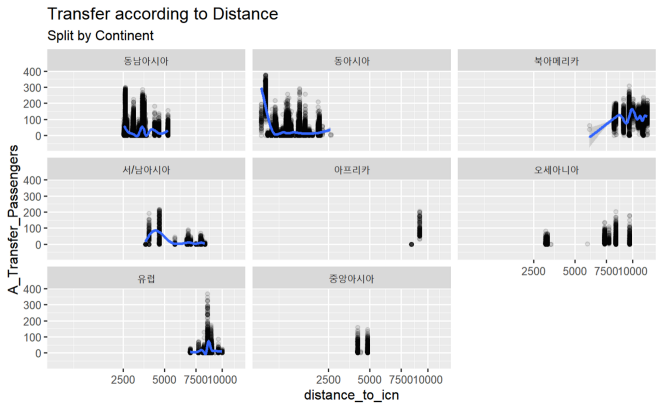

Figure 46: Relationship between distance and the number of transit passengers in continent-by-continent arrival (see code chunk 39)

<Trend between distance and the number of transit arrival passengers in East Asia>

```
subset(arrival,A_Continent=='동아시아')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='firebrick')+
  labs(title="Transfer according to Distance",subtitle="East Asia")
```

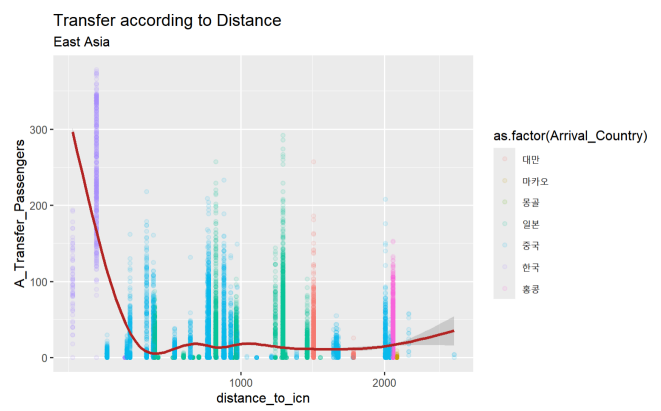

Figure 47: Trend between distance and the number of transit arrival passengers in East Asia (see code chunk 40)

<Relation between distance and the number of transit arrival passengers in East Asia>

```
subset(arrival,A_Continent=='동아시아')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='firebrick')+
  labs(title="Transfer according to Distance",subtitle="East Asia")
```

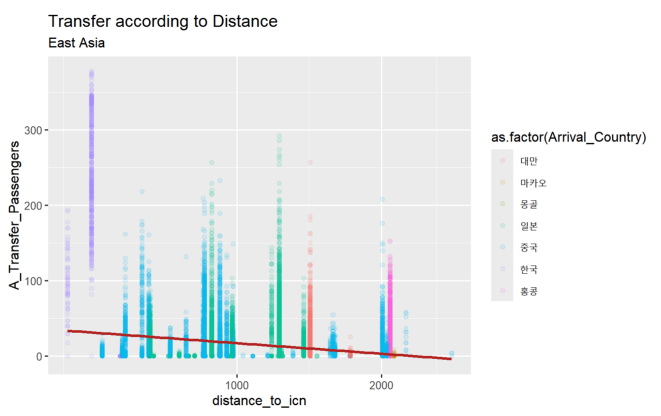

Figure 48: Relation between distance and the number of transit arrival passengers in East Asia (see code chunk 41)

<Trend between distance and the number of transit arrival passengers in Europe>

```
subset(arrival,A_Continent=='유럽')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='Steel Blue')+
  labs(title="Transfer according to Distance",subtitle="Europe")
```

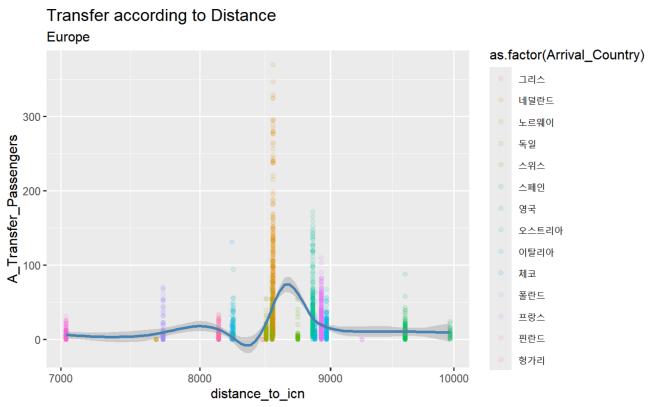

Figure 49: Trend between distance and the number of transit arrival passengers in Europe (see code chunk 42)

<Relation between distance and the number of transit arrival passengers in Europe>

```
subset(arrival,A_Continent=='유럽')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='Steel Blue')+
  labs(title="Transfer according to Distance",subtitle="Europe")
```

Rather, the relationship between distance and transfer numbers turned out to be proportional in Europe and North America, which are far from Incheon.

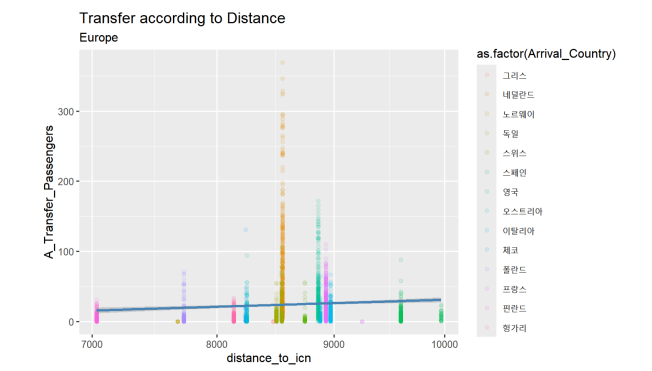

Figure 50: Relation between distance and the number of transit arrival passengers in Europe (see code chunk 43)

<Finding correlation between distance and the number of passengers in Europe>

```
europe_arrival <- subset(arrival, A_Continent == 'Europe')
lm(distance_to_icn ~ A_Transfer_Passengers, data = europe_arrival)
summary(lm(distance_to_icn ~ D_Transfer_Passengers, data = europe_departure))
```

The relationship in Europe is statistically significant because it is less than 0.05.
Only 3% of the data represents the relationship.

```
Call:
lm(formula = distance_to_icn ~ A_Transfer_Passengers, data = europe_arrival)

Coefficients:
          (Intercept)  A_Transfer_Passengers
             8572.297                 1.041


Call:
lm(formula = distance_to_icn ~ D_Transfer_Passengers, data = europe_departure)

Residuals:
     Min       1Q   Median       3Q      Max
-1795.12  -135.98    20.55   335.81  1452.11

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          8507.2664    13.9934  607.95   <2e-16 ***
D_Transfer_Passengers    4.4747     0.4671    9.58   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 556.1 on 2878 degrees of freedom
Multiple R-squared:  0.0309,    Adjusted R-squared:  0.03056
F-statistic: 91.77 on 1 and 2878 DF,  p-value: < 2.2e-16
```

Figure 51: Finding correlation between distance and the number of passengers in Europe (see code chunk 44)

<Trend between distance and the number of transit arrival passengers in North America>

```
subset(arrival,A_Continent=='North America')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(colour='Dark Slate Gray')+
  labs(title="Transfer according to Distance",subtitle="North America")
```

The flow between distances and transfer numbers around the world and that of North America are similar, also the relationship is similar. Through this, it can be seen that the flow of distances and transfer numbers around the world is similar to that of Asia and North America, which can be inferred that Asia and North America account for a significant proportion of the number of transfer numbers.

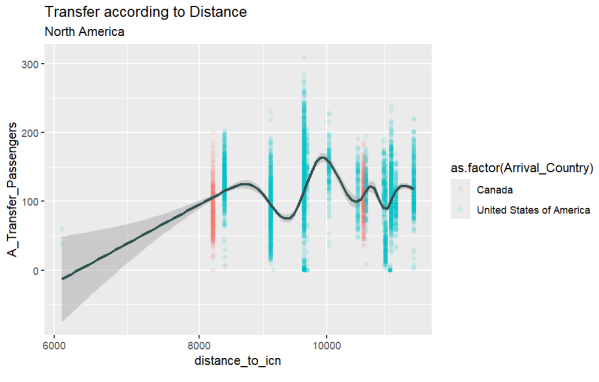

Figure 52: Trend between distance and the number of transit arrival passengers in North America (see code chunk 45)

<Relation between distance and the number of transit arrival passengers in North America>

```
subset(arrival,A_Continent=='North America')%>%
ggplot(aes(x=distance_to_icn,y=A_Transfer_Passengers,colour=as.factor(Arrival_Country)))+
  geom_point(alpha=0.1)+
  scale_x_continuous(trans="sqrt")+
  geom_smooth(method='lm',colour='Dark Slate Gray')+
  labs(title="Transfer according to Distance",subtitle="North America")
```

If we were to determine the relationship between the distance from the airport and the number of transfer passengers, it could be concluded that proportion is closer than inverse proportion, but it is actually irrelevant because the data do not represent the relationship. However, a certain pattern appeared in the relationship between distance and the number of transfer passengers, and that pattern was an important indicator of the transit influence in East Asia and North America.

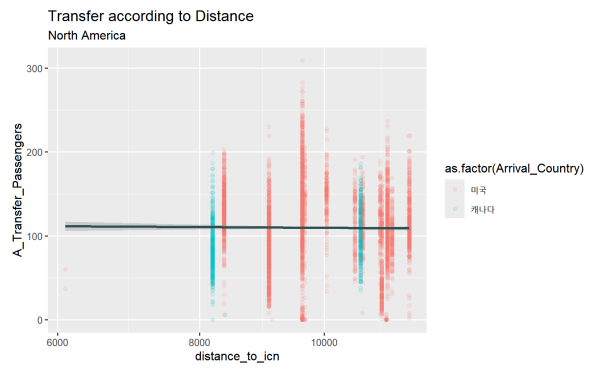

Figure 53: Relation between distance and the number of transit arrival passengers in North America (see code chunk 46)
* the US is red here

[Routes Proposal]

<Purpose>

Incheon International Airport has a status as a hub airport that transfers a lot. Incheon International Airport's transfer airliner supply was only skewed to Southeast Asia and the United States.
However, We believe that the diversity of transit passengers should be guaranteed at the hub airport. Additionally, data analysis showed that East Asia had a noticeable number of transit passengers and that Busan acted as a transit route, whether it was the destination or the departure point.
Although it is a popular place, airports that have not developed much as transfer routes were also seen in East Asia and Europe. Therefore, we wanted to propose airports that have not developed much as transfer routes as new transfer routes, and existing transfer routes as new direct routes.

<Method>

We believe that the time required for connecting flights is short if the route has many passengers. Therefore, we assumed a minimum of 0 minutes and a maximum of 90 minutes for connecting flights. Based on the above premise, we first selected up to 25 airlines with many connections. Then, we inner-joined them on the same day and grouped airlines with the same departure and arrival. Average passengers were summarized with the formula below and filtered only those transfer routes with more than 100 passengers and more than 100 combinations. Based on the above, we were able to suggest transit routes based on the combinations.

<Making a route Dataset with 30 minutes or less transfer time>

```
most_common_departure_airport <- departure %>%
  group_by(Departure_Airport) %>%
  summarise(count = sum(D_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Departure_Airport)

most_common_arrival_airport <- arrival %>%
  group_by(Arrival_Airport) %>%
  summarise(count = sum(A_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Arrival_Airport)

joined_data <- departure %>%
  filter(Departure_Airport %in% most_common_departure_airport) %>%
  inner_join(arrival %>% filter(Arrival_Airport %in% most_common_arrival_airport), by = c("D_Flight_Date" = "A_Flight_Date"))

filtered_data_all <- joined_data %>%
  filter(A_Scheduled_Time > D_Scheduled_Time &abs(difftime(A_Scheduled_Time, D_Scheduled_Time, units = "sec")) <= 1800)%>%filter(Depart
ure_Country != Arrival_Country)

airport_combinations <- filtered_data_all %>%
  group_by(Departure_Airport, Arrival_Airport) %>%
  summarise(
```

```
    count = n(),
    total_transfer_passengers = (sum(A_Transfer_Passengers, na.rm = TRUE) + sum(D_Transfer_Passengers, na.rm = TRUE)) / (2 * n())
  ) %>%
  arrange(desc(total_transfer_passengers)) %>%
  filter(count >= 100 & total_transfer_passengers >= 100)
```

<Making a route Dataset more than 30 minutes or less than an hour transfer time>

```
most_common_departure_airport <- departure %>%
  group_by(Departure_Airport) %>%
  summarise(count = sum(D_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Departure_Airport)

most_common_arrival_airport <- arrival %>%
  group_by(Arrival_Airport) %>%
  summarise(count = sum(A_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Arrival_Airport)

joined_data <- departure %>%
  filter(Departure_Airport %in% most_common_departure_airport) %>%
  inner_join(arrival %>% filter(Arrival_Airport %in% most_common_arrival_airport), by = c("D_Flight_Date" = "A_Flight_Date"))

filtered_data_all <- joined_data %>%
  filter(A_Scheduled_Time > D_Scheduled_Time &abs(difftime(A_Scheduled_Time, D_Scheduled_Time, units = "sec") > 1800 & difftime(A_Sched
uled_Time, D_Scheduled_Time, units = "sec")<=3600))%>%filter(Departure_Country != Arrival_Country)

airport_combinations2 <- filtered_data_all %>%
  group_by(Departure_Airport, Arrival_Airport) %>%
  summarise(
    count = n(),
    total_transfer_passengers = (sum(A_Transfer_Passengers, na.rm = TRUE) + sum(D_Transfer_Passengers, na.rm = TRUE)) / (2 * n())
  ) %>%
  arrange(desc(total_transfer_passengers)) %>%
  filter(count >= 100 & total_transfer_passengers >= 100)
```

<Making a route Dataset more than 60 minutes or less than 90 minutes transfer time>

```
most_common_departure_airport <- departure %>%
  group_by(Departure_Airport) %>%
  summarise(count = sum(D_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Departure_Airport)

most_common_arrival_airport <- arrival %>%
  group_by(Arrival_Airport) %>%
  summarise(count = sum(A_Transfer_Passengers)) %>%
  arrange(desc(count)) %>%
  slice(1:25) %>%
  pull(Arrival_Airport)

joined_data <- departure %>%
  filter(Departure_Airport %in% most_common_departure_airport) %>%
  inner_join(arrival %>% filter(Arrival_Airport %in% most_common_arrival_airport), by = c("D_Flight_Date" = "A_Flight_Date"))

filtered_data_all <- joined_data %>%
  filter(A_Scheduled_Time > D_Scheduled_Time &abs(difftime(A_Scheduled_Time, D_Scheduled_Time, units = "sec") > 3600 & difftime(A_Sched
uled_Time, D_Scheduled_Time, units = "sec")<=5400))%>%filter(Departure_Country != Arrival_Country)

airport_combinations3 <- filtered_data_all %>%
  group_by(Departure_Airport, Arrival_Airport) %>%
  summarise(
    count = n(),
    total_transfer_passengers = (sum(A_Transfer_Passengers, na.rm = TRUE) + sum(D_Transfer_Passengers, na.rm = TRUE)) / (2 * n())
  ) %>%
  arrange(desc(total_transfer_passengers)) %>%
  filter(count >= 100 & total_transfer_passengers >= 100)
```

<Making a visualizing map graph>

We made each airport's Korean name and latitude and longitude into a data frame one by one. After, latitude and longitude were entered into the previously created air route data frame and visualized on a map.

```
install.packages("geosphere")
install.packages("maps")
install.packages("ggmap")
library(geosphere)
library(maps)
library(ggmap)
```

Same color of line means countries are included in one route (it doesn't matter if it's a departure or arrival).
The more routes, the more colors can overlap, and the more passengers the thicker the lines.
For example, map for less than 30 minutes, Busan has more than 3 routes(Busan-Manila, Busan-Osaka, Busan-Ho-chi-min city)

<Map for 30 minutes or less transfer time>

You can see that Busan is very influential due to its short distance. Busan has a significant influence whether it is a departure or arrival airport. Also, there are a lot of East Asian routes such as Taipei, Shanghai, and Osaka. Through these routes, it is possible to propose to Busan Airport to accommodate direct flights from nearby Southeast Asian and East Asian countries to Busan Airport.

```
airport_locations <- data.frame(
  Airport = c("인천", "마닐라", "호찌민", "부산", "로스앤젤레스", "싱가포르", "타이베이", "오사카/ 간사이", "방콕/수완나품", "하노이", "도쿄/나리타", "시애
틀", "샌프란시스코", "상하이/푸동"),
  Latitude = c(37.4587, 14.5086, 10.8185, 35.1796, 33.9416, 1.3644, 25.0796, 34.4342, 13.6899, 21.2212, 35.7702, 47.4502, 37.6213, 31.1
433),
  Longitude = c(126.4420, 121.0198, 106.6519, 128.9408, -118.4085, 103.9915, 121.2340, 135.2440, 100.7501, 105.8079, 140.3843, -122.308
8, -122.3790, 121.8058)
)

# add Incheon Airport location
icn_location <- airport_locations %>% filter(Airport == "인천")

# merge airport location data with airport_combinations
airport_combinations_with_locations <- airport_combinations %>%
  left_join(airport_locations, by = c("Departure_Airport" = "Airport")) %>%
  rename(Departure_Latitude = Latitude, Departure_Longitude = Longitude) %>%
  left_join(airport_locations, by = c("Arrival_Airport" = "Airport")) %>%
  rename(Arrival_Latitude = Latitude, Arrival_Longitude = Longitude)

ggplot() +
  world_map +
  geom_segment(data = airport_combinations_with_locations,
               aes(x = Departure_Longitude, y = Departure_Latitude,
                   xend = icn_location$Longitude, yend = icn_location$Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
  geom_segment(data = airport_combinations_with_locations,
               aes(x = icn_location$Longitude, y = icn_location$Latitude,
                   xend = Arrival_Longitude, yend = Arrival_Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
  geom_point(data = airport_locations,
             aes(x = Longitude, y = Latitude, label = Airport),
             color = "blue", size = 3) +
  geom_text(data = airport_locations,
            aes(x = Longitude, y = Latitude, label = Airport),
            hjust = 1.5, vjust = 1.5, size=2) +
  scale_size_continuous(range = c(0.5, 2)) +
  labels(title = "Airport Routes Recommendation via Incheon",
       subtitle = "Based on Average Transfer Passengers",
       x = "Longitude", y = "Latitude") +
  theme_minimal() +
  scale_color_manual(values = colors)+
  coord_fixed(ratio =10000)+
  coord_map("ortho", orientation = c(icn_location$Latitude, icn_location$Longitude, 0))+
  coord_map(xlim = c(90, 180), ylim = c(-10, 60))
  theme(legend.position = "none")
```

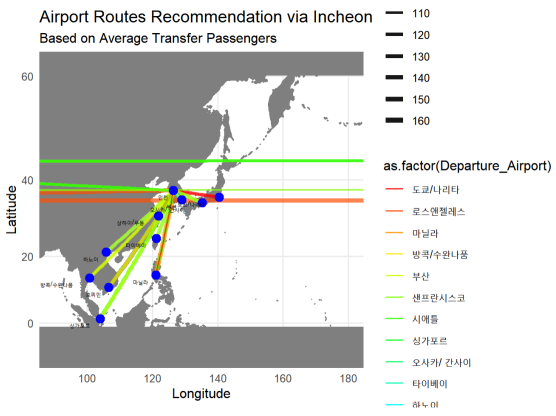

Figure 54: route map for 30 minutes or less transfer time (see code chunk 51)

<Map for more than 30 minutes and less than hour>

Again, Busan is highly influential, but we were also able to identify connecting routes between Western countries and East and Southeast Asian countries, such as Seattle, San Francisco to Manila, and Ho Chi Minh City to Los Angeles. Through the above analysis, West to Southeast and East Asia routes are worth proposing as direct or transfer routes.

```
airport_locations <- data.frame(
  Airport = c("인천", "마닐라", "호찌민", "부산", "로스앤젤레스", "싱가포르", "타이베이", "오사카/ 간사이", "방콕/수완나품", "하노이", "도쿄/나리타", "시애
틀", "샌프란시스코", "상하이/푸동","홍콩","애틀랜타","세부","신 울란바토르","토론토"),
  Latitude = c(37.4587, 14.5086, 10.8185, 35.1796, 33.9416, 1.3644, 25.0796, 34.4342, 13.6899, 21.2212, 35.7702, 47.4502, 37.6213, 31.1
433,22.3135,33.6362,10.3131,47.6514,43.6771),
```

```
  Longitude = c(126.4420, 121.0198, 106.6519, 128.9408, -118.4085, 103.9915, 121.2340, 135.2440, 100.7501, 105.8079, 140.3843, -122.308
8, -122.3790, 121.8058,113.9137,-84.4294,123.9764,106.8216,-79.6334)
)

# add Incheon Airport location
icn_location <- airport_locations %>% filter(Airport == "인천")

# merge airport location data with airport_combinations
airport_combinations_with_locations2 <- airport_combinations2 %>%
  left_join(airport_locations, by = c("Departure_Airport" = "Airport")) %>%
  rename(Departure_Latitude = Latitude, Departure_Longitude = Longitude) %>%
  left_join(airport_locations, by = c("Arrival_Airport" = "Airport")) %>%
  rename(Arrival_Latitude = Latitude, Arrival_Longitude = Longitude)

# set the world map
world_map <- borders("world", colour="gray50", fill="gray50")

# visualize the world map
ggplot() +
  world_map +
  geom_segment(data = airport_combinations_with_locations2,
               aes(x = Departure_Longitude, y = Departure_Latitude,
                   xend = icn_location$Longitude, yend = icn_location$Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
  geom_segment(data = airport_combinations_with_locations2,
               aes(x = icn_location$Longitude, y = icn_location$Latitude,
                   xend = Arrival_Longitude, yend = Arrival_Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
geom_point(data = airport_locations,
           aes(x = Longitude, y = Latitude),
           color = "blue", size = 3) +
geom_text(data = airport_locations,
          aes(x = Longitude, y = Latitude, label = Airport),
          hjust = 1.5, vjust = 1.5, size=2) +
scale_color_manual(values = colors)+
scale_size_continuous(range = c(0.5, 2)) +
labs(title = "Airport Routes Recommendation via Incheon",
     subtitle = "Based on Average Transfer Passengers",
     x = "Longitude", y = "Latitude") +
coord_map("ortho", orientation = c(icn_location$Latitude, icn_location$Longitude, 0)) +
theme_minimal() +
theme(legend.position = "none")
```

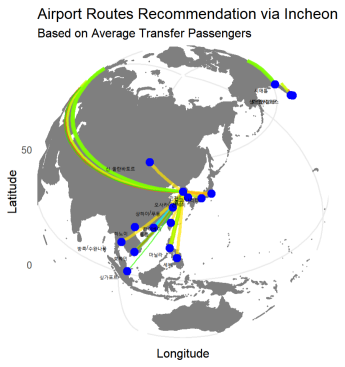

Figure 55: route map for more than 30 minutes and less than hour transfer time (see code chunk 52)

<Map for more than hour and less than 90 minutes>

The frequency of Western destinations has increased significantly compared to before. Once again, Western and Southeast Asian routes, such as Seattle and San Francisco to Manila and Ho Chi Minh City, dominated the list. Above 60 minutes, the airports are a bit more diverse. We can propose new routes with new airports such as New Ulaanbaatar in East Asia, Vancouver in the West, and Europe.

```
airport_locations <- data.frame(
  Airport = c("인천", "마닐라", "호찌민", "부산", "로스앤젤레스", "싱가포르", "타이베이", "오사카/간사이", "방콕/수완나품", "하노이", "도쿄/나리타", "시애
틀", "샌프란시스코", "상하이/푸동","홍콩","애틀란타","세부","신 울란바토르","토론토","밴쿠버"),
  Latitude = c(37.4587, 14.5086, 10.8185, 35.1796, 33.9416, 1.3644, 25.0796, 34.4342, 13.6899, 21.2212, 35.7702, 47.4502, 37.6213, 31.1
433,22.3135,33.6362,10.3131,47.6514,43.6771,49.1934),
  Longitude = c(126.4420, 121.0198, 106.6519, 128.9408, -118.4085, 103.9915, 121.2340, 135.2440, 100.7501, 105.8079, 140.3843, -122.308
8, -122.3790, 121.8058,113.9137,-84.4294,123.9764,106.8216,-79.6334,-123.1751)
)

# the same as above
icn_location <- airport_locations %>% filter(Airport == "인천")

airport_combinations_with_locations3 <- airport_combinations3 %>%
  left_join(airport_locations, by = c("Departure_Airport" = "Airport")) %>%
  rename(Departure_Latitude = Latitude, Departure_Longitude = Longitude) %>%
  left_join(airport_locations, by = c("Arrival_Airport" = "Airport")) %>%
  rename(Arrival_Latitude = Latitude, Arrival_Longitude = Longitude)

world_map <- borders("world", colour="gray50", fill="gray50")
```

```
ggplot() +
  world_map +
  geom_segment(data = airport_combinations_with_locations3,
               aes(x = Departure_Longitude, y = Departure_Latitude,
                   xend = icn_location$Longitude, yend = icn_location$Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
  geom_segment(data = airport_combinations_with_locations3,
               aes(x = icn_location$Longitude, y = icn_location$Latitude,
                   xend = Arrival_Longitude, yend = Arrival_Latitude,
                   size = total_transfer_passengers, color = as.factor(Departure_Airport)),
               alpha = 0.7) +
  geom_point(data = airport_locations,
             aes(x = Longitude, y = Latitude),
             color = "blue", size = 3) +
  geom_text(data = airport_locations,
            aes(x = Longitude, y = Latitude, label = Airport),
            hjust = 1.5, vjust = 1.5, size=2) +
  scale_color_manual(values = colors)+
  scale_size_continuous(range = c(0.5, 2)) +
  labs(title = "Airport Routes Recommendation via Incheon",
       subtitle = "Based on Average Transfer Passengers",
       x = "Longitude", y = "Latitude") +
  coord_map("ortho", orientation = c(icn_location$Latitude, icn_location$Longitude, 0)) +
  theme_minimal() +
  theme(legend.position = "none")
```

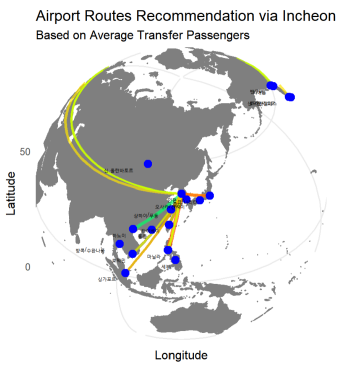

Figure 56: route map for more than an hour and less than 90 minutes transfer time (see code chunk 53)

5. Conclusions

To summarize, our conclusions are as follows.

Firstly, we discovered that as summer approached, the number of transfer passengers to South Asia didn't increase, likely due to the monsoon season. In the mid of January, While East Asia didn’t significantly contribute to a peak in total passengers, South Asia might have, and we plan to explore this further.

Secondly, we found that distance from the transfer airport doesn’t necessarily increase transfer passengers. Instead, regional factor like East Asia and Western countries play a significant role for this.

With these insights, we can propose new routes: direct flights to Busan from cities like Tokyo, Osaka, Shanghai, Taipei, and Hong Kong, and new connections such as Bangkok to Amsterdam and Vancouver to Ulaanbaatar. These routes will enhance connectivity and passenger experience, finally benefiting all of the passengers using the Incheon Airport and the airport itself, too.