

Explanatory Visualizations

2024-06-11

```
library(readr)
library(vioplplot)
```

```
## Loading required package: sm
## Package 'sm', version 2.2-6.0: type help(sm) for summary information
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v purrr      1.0.2
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.0      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
##
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(dplyr)
library(fmsb)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
```

```
library(tibble)
library(tidyr)
```

```
airline_passenger_satisfaction <- read_csv("airline_passenger_satisfaction.csv", show_col_types = F)
```

```

# Binning flight distance short, medium, long, very long-haul
airline_passenger_satisfaction <- airline_passenger_satisfaction %>%
  mutate(
    flightTypeDistance = case_when(
      `Flight Distance` < 500 ~ "Short-haul",
      `Flight Distance` <= 1500 ~ "Medium-haul",
      `Flight Distance` <= 3000 ~ "Long-haul",
      `Flight Distance` > 3000 ~ "Very long-haul",
    )
  )

# filtering by customer type
firstTimeFlyers <- airline_passenger_satisfaction %>%
  filter(`Customer Type` == "First-time")

returningFlyers <- airline_passenger_satisfaction %>%
  filter(`Customer Type` == "Returning")

#to set font
# par(family = "A", ps = 16)
# windowsFonts(A = windowsFont("Times New Roman"))

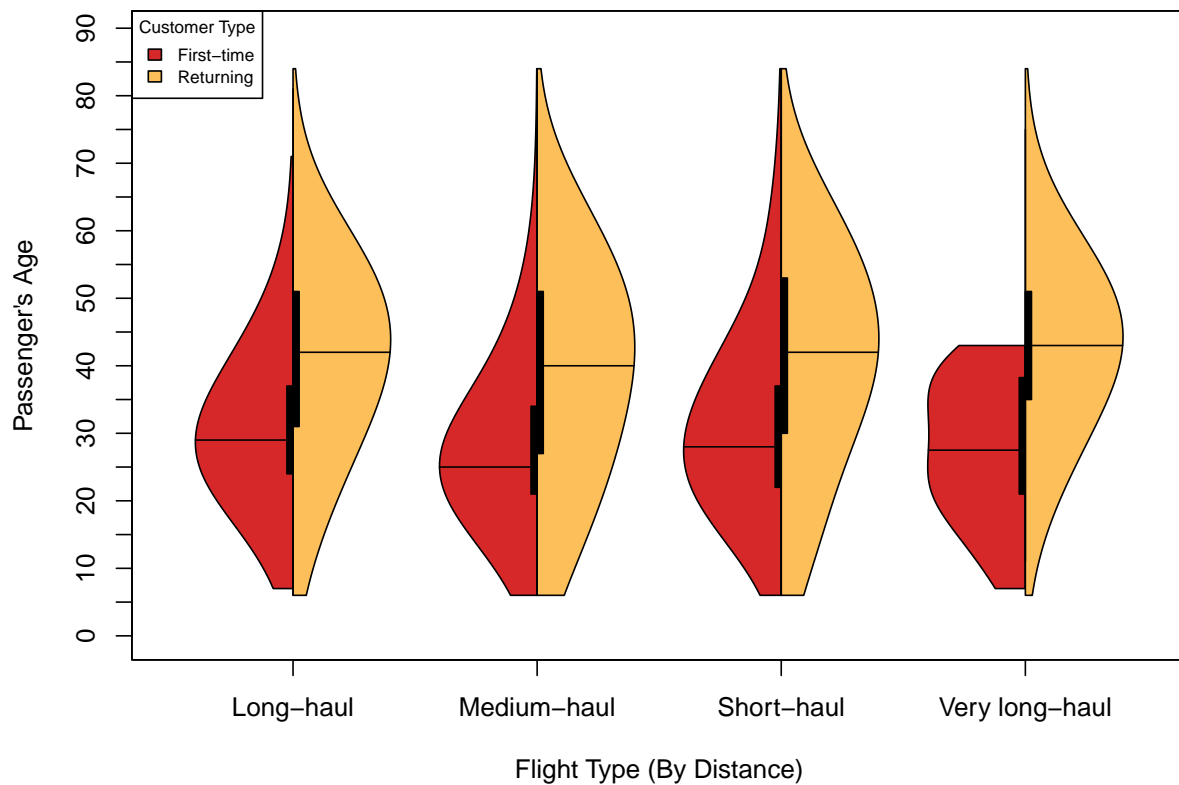
# Creating one half of the violin plot (first time flyers)
vioplot(Age~flightTypeDistance,
  data=firstTimeFlyers,
  col = "#d62828",
  ylim=c(1,90),
  yaxt = "n",
  plotCentre = "line",
  side = "left",
  ylab = "Passenger's Age",
  xlab = "Flight Type (By Distance)",
  las = 0)

#Adding a violin plot of display 2 to the previously created plot
vioplot(Age~flightTypeDistance,
  data=returningFlyers,
  col = "#FCBF59",
  plotCentre = "line",
  side = "right",
  add = T)

#tweaking non-data ink
axis(side = 2, at = seq(1,91,5), labels = seq(0,90,5))
legend("topleft", fill = c("#d62828", "#FCBF59"),
  legend = c("First-time", "Returning"), title = "Customer Type", cex = 0.7)
title(main = "Age Distribution of Travelers by Flight Distance")

```

Age Distribution of Travelers by Flight Distance



```
satisfaction_df <- read.csv("airline_passenger_satisfaction.csv")

# Function to calculate mode
get_mode <- function(v) {
  v <- v[!is.na(v)] # Remove NA values
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Ensure relevant columns are numeric
satisfaction_df <- satisfaction_df %>%
  mutate(across(c(Departure.and.Arrival.Time.Convenience, Ease.of.Online.Booking, Check.in.Service, Onl.
    On.board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness, Food.and.Drink,
    In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment, Baggage.Handling)
    as.numeric))

# Group Age
satisfaction_df <- satisfaction_df %>%
  mutate(Age_Group = case_when(
    Age >= 0 & Age <= 12 ~ "Children",
    Age >= 13 & Age <= 17 ~ "Teenagers",
    Age >= 18 & Age <= 24 ~ "Young Adults",
    Age >= 25 & Age <= 39 ~ "Adults",
    Age >= 40 & Age <= 59 ~ "Mid-Aged",
```

```

    Age >= 60 ~ "Seniors"
  ))

# Group ratings and calculate mode
satisfaction_modes <- satisfaction_df %>%
  group_by(Gender, Customer.Type, Age_Group, Class) %>%
  summarise(across(c(Departure.and.Arrival.Time.Convenience, Ease.of.Online.Booking, Check.in.Service, (
    On.board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness, Food.and.Drink,
    In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment, Baggage.Handling
  get_mode), .groups = 'drop')) %>%

mutate(
  `Booking Services` = Ease.of.Online.Booking,
  `Travel Time Convenience` = Departure.and.Arrival.Time.Convenience,
  `Pre-Boarding Services` = rowMeans(select(., Check.in.Service, Online.Boarding, Gate.Location), na.rm=T),
  `Flight Services` = rowMeans(select(., On.board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness, Food.and.Drink,
  `Essential Services` = rowMeans(select(., In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment, Baggage.Handling
  `Post Flight Services` = Baggage.Handling
)

# Data for radar chart
ratings_by_class_type <- satisfaction_modes %>%
  group_by(Class) %>%
  summarise(across(`Booking Services`:`Post Flight Services`, get_mode), .groups = 'drop')
ratings_by_class_type

## # A tibble: 3 x 7
##   Class      `Booking Services` Travel Time Convenien-1 Pre-Boarding Service-2
##   <chr>          <dbl>          <dbl>          <dbl>
## 1 Business          4              4              4
## 2 Economy           3              4              3
## 3 Economy Plus      2              4              3
## # i abbreviated names: 1: `Travel Time Convenience`, 2: `Pre-Boarding Services`
## # i 3 more variables: `Flight Services` <dbl>, `Essential Services` <dbl>,
## #   `Post Flight Services` <dbl>

# Data to have Business first, then First, and Economy
ratings_by_class_type <- ratings_by_class_type %>%
  arrange(factor(Class, levels = c("Business", "Economy Plus", "Economy")))
ratings_by_class_type

## # A tibble: 3 x 7
##   Class      `Booking Services` Travel Time Convenien-1 Pre-Boarding Service-2
##   <chr>          <dbl>          <dbl>          <dbl>
## 1 Business          4              4              4
## 2 Economy Plus      2              4              3
## 3 Economy           3              4              3
## # i abbreviated names: 1: `Travel Time Convenience`, 2: `Pre-Boarding Services`
## # i 3 more variables: `Flight Services` <dbl>, `Essential Services` <dbl>,
## #   `Post Flight Services` <dbl>

# Data for fmsb radarchart
ratings_data <- ratings_by_class_type %>%
  select(`Booking Services`, `Travel Time Convenience`, `Pre-Boarding Services`, `Flight Services`, `Essential Services`, `Post Flight Services`)
  rbind(rep(5, 6), rep(1, 6), .)

```

```

# Colors for each class
colors <- c("#003049", "#D62828", "#FCBF49")
fill_colors <- alpha(colors, 0.2)

# Save the plot as a PNG file
#png(filename = "Ratings_by_Class_new.png", width = 800, height = 800)

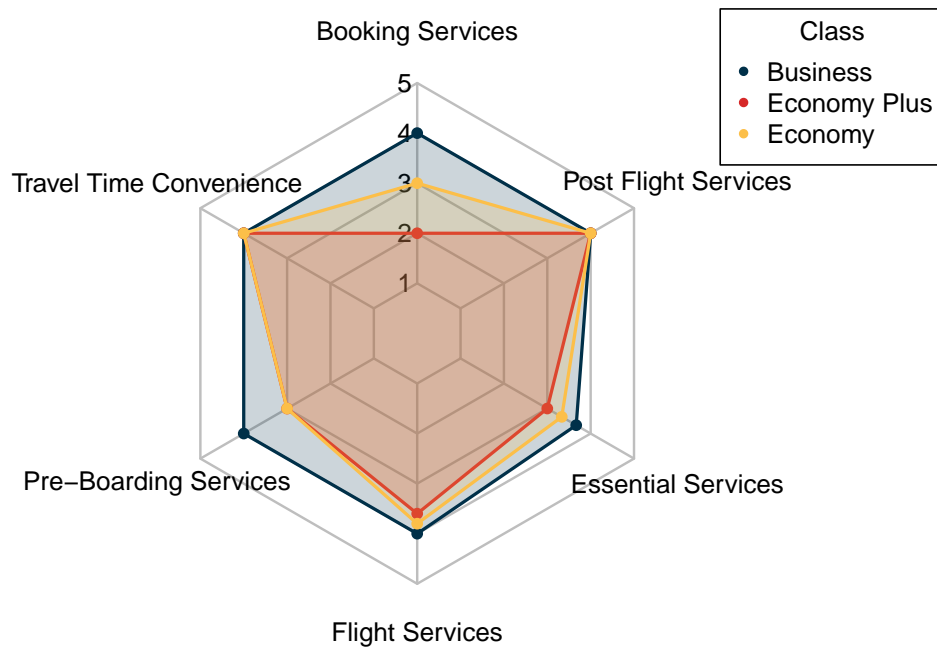
# Radar chart function
create_beautiful_radarchart <- function(data, color = "#00AFBB",
                                         vlabels = colnames(data), vlce = 0.7,
                                         caxislabels = NULL, title = NULL, ...){
  radarchart(
    data, axistype = 1,
    pcol = color, pfc = scales::alpha(color, 0.5), plwd = 10, plty = 1,
    cglcol = "grey", cglty = 1, cglwd = 0.8,
    axislabcol = "grey",
    vlce = vlce, vlabels = vlabels,
    caxislabels = caxislabels, title = title, ...
  )
}

# Radar chart with all classes
par(mfrow = c(1, 1))
radarchart(
  ratings_data, axistype = 1,
  pcol = colors, pfc = fill_colors, plwd = 2, plty = 1,
  cglcol = "grey", cglty = 1, cglwd = 1.5,
  axislabcol = "black",
  vlce = 1.0, vlabels = colnames(ratings_data),
  caxislabels = c(1, 2, 3, 4, 5)
)

title(main = "Service Ratings by Class", font.main = 2, family = "serif", cex.main = 1.6)
legend(x = "topright", legend = c("Business", "Economy Plus", "Economy"), col = colors, pch = 20, title

```

Service Ratings by Class



```
#dev.off()
```

```
satisfaction_df <- read.csv("airline_passenger_satisfaction.csv")
```

```
# Function to calculate mode
```

```
get_mode <- function(v) {
  v <- v[!is.na(v)] # Remove NA values
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
# Ensure relevant columns are numeric
```

```
satisfaction_df <- satisfaction_df %>%
  mutate(across(c(Departure.and.Arrival.Time.Convenience, Ease.of.Online.Booking, Check.in.Service, Onl.
    On.board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness, Food.and.Drink,
    In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment, Baggage.Handling)
    as.numeric))
```

```
# Group Age
```

```
satisfaction_df <- satisfaction_df %>%
  mutate(Age_Group = case_when(
    Age >= 0 & Age <= 12 ~ "Children",
    Age >= 13 & Age <= 17 ~ "Teenagers",
    Age >= 18 & Age <= 24 ~ "Young Adults",
```

```

    Age >= 25 & Age <= 39 ~ "Adults",
    Age >= 40 & Age <= 59 ~ "Mid-Aged",
    Age >= 60 ~ "Seniors"
  ))

# Filter dataset for only Economy and Economy Plus classes
satisfaction_df_filtered <- satisfaction_df %>%
  filter(Class %in% c("Economy", "Economy Plus"))

# Group ratings and calculate means for grouped services
satisfaction_modes_filtered <- satisfaction_df_filtered %>%
  group_by(Gender, Customer.Type, Age_Group, Class) %>%
  summarise(across(c(Departure.and.Arrival.Time.Convenience, Ease.of.Online.Booking, Check.in.Service, On.
    board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness, Food.and.Drink,
    In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment, Baggage.Handling),
    mean, na.rm = TRUE), .groups = 'drop') %>%

  mutate(
    `Booking Services` = Ease.of.Online.Booking,
    `Travel Time Convenience` = Departure.and.Arrival.Time.Convenience,
    `Pre-Boarding Services` = rowMeans(select(., Check.in.Service, Online.Boarding, Gate.Location), na.rm = TRUE),
    `Flight Services` = rowMeans(select(., On.board.Service, Seat.Comfort, Leg.Room.Service, Cleanliness), na.rm = TRUE),
    `Essential Services` = rowMeans(select(., In.flight.Service, In.flight.Wifi.Service, In.flight.Entertainment), na.rm = TRUE),
    `Post Flight Services` = Baggage.Handling
  )

## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(...)` .
## i In group 1: `Gender = "Female"`, `Customer.Type = "First-time"`, `Age_Group = "Adults"`, `Class = "Economy"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
## # Previously
## across(a:b, mean, na.rm = TRUE)
##
## # Now
## across(a:b, \(x) mean(x, na.rm = TRUE))

# Gather the data for plots
ratings_long <- satisfaction_modes_filtered %>%
  pivot_longer(cols = c(`Booking Services`, `Travel Time Convenience`, `Pre-Boarding Services`, `Flight Services`, `Essential Services`, `Post Flight Services`),
    names_to = "Service_Type", values_to = "Rating")
ratings_long$Age_Group <- factor(ratings_long$Age_Group,
  levels = c("Children", "Teenagers", "Young Adults", "Adults", "Mid-Aged", "Seniors"))

# Define custom colors for age groups
age_group_colors <- c("Children" = "#4F83CC", "Teenagers" = "#D62828", "Young Adults" = "#F77F00",
  "Adults" = "#FCBF49", "Mid-Aged" = "#9467BD", "Seniors" = "#2CA02C")

# Save the plot as a PNG file
#png(filename = "Ratings_by_Age_with_Class_Filter_new.png", width = 1050, height = 900)

# Create box plot

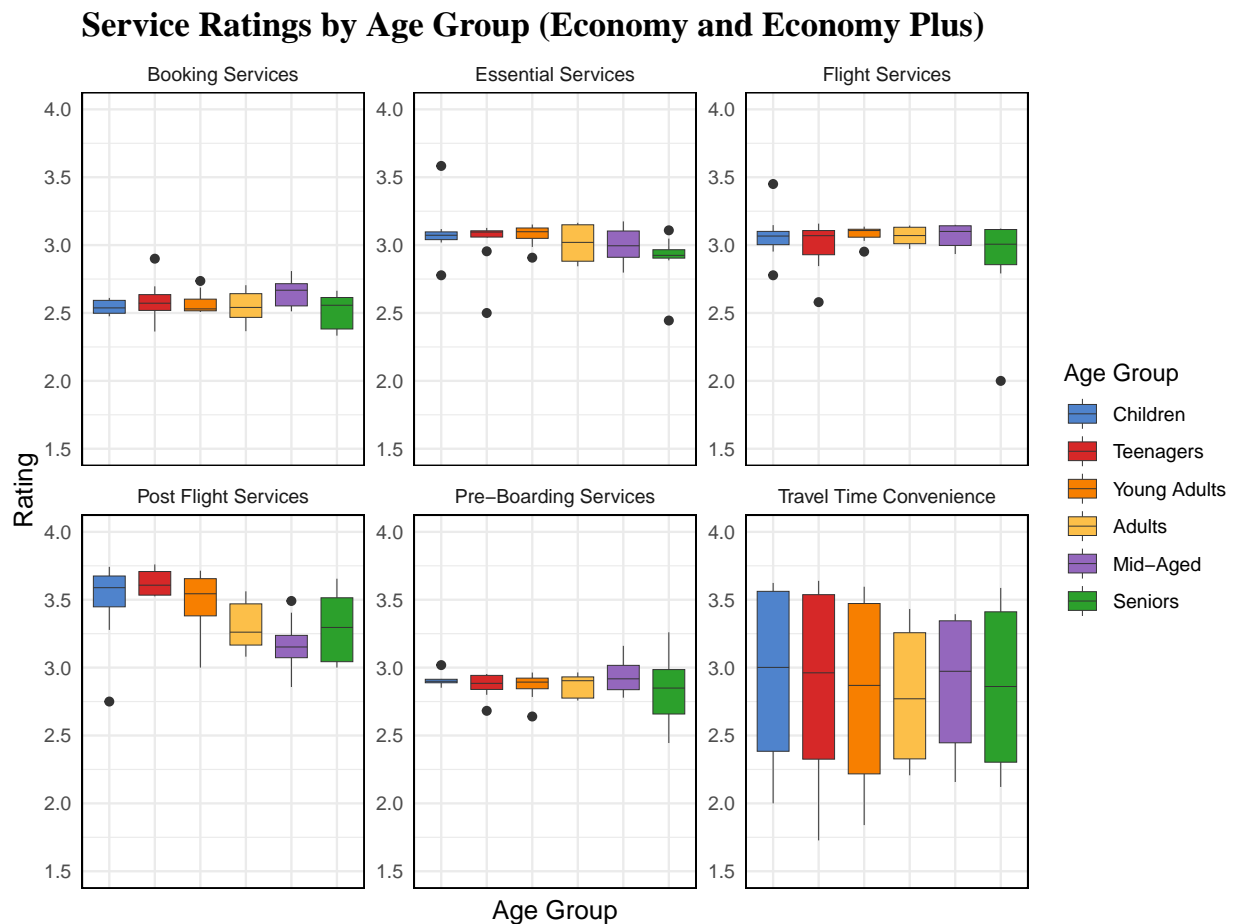
```

```

box_plot <- ggplot(ratings_long, aes(x = Age_Group, y = Rating, fill = Age_Group)) +
  geom_boxplot(width = 0.7, size = 0.2, # Adjust size for overall lines of the box plot
    fatten = 1) +
  facet_wrap(~ Service_Type, scales = "free_y") +
  theme_minimal() +
  scale_fill_manual("Age Group", values = age_group_colors) +
  theme(axis.text.x = element_text(angle = 0, hjust = 0.5, size = 7)) +
  labs(title = "Service Ratings by Age Group (Economy and Economy Plus)", x = "Age Group", y = "Rating")
  theme(
    plot.title = element_text(family = "serif", face = "bold", size = 16),
    axis.text.x = element_blank(),
    axis.title.x = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    panel.border = element_rect(color = "black", fill = NA)
  ) +
  scale_y_continuous(breaks = seq(1.5, 4, by = 0.5), limits = c(1.5, 4))

# Print the box plot
print(box_plot)

```



```
#dev.off()
```