Hotel Bookings Data Analysis

Mohamed Said

2024-09-16

```
setwd("C:\\Users\\mohamed said\\OneDrive\\Desktop\\Mohamed Said Mahmooud Emam_YAT200_project4")
hotel_bookings <- read.csv("hotel_bookings.csv")

# Verify column names
colnames(hotel_bookings)
```

```
##  [1] "hotel"                          "is_canceled"
##  [3] "lead_time"                      "arrival_date_year"
##  [5] "arrival_date_month"             "arrival_date_week_number"
##  [7] "arrival_date_day_of_month"      "stays_in_weekend_nights"
##  [9] "stays_in_week_nights"           "adults"
## [11] "children"                       "babies"
## [13] "meal"                           "country"
## [15] "market_segment"                 "distribution_channel"
## [17] "is_repeated_guest"              "previous_cancellations"
## [19] "previous_bookings_not_canceled" "reserved_room_type"
## [21] "assigned_room_type"             "booking_changes"
## [23] "deposit_type"                   "agent"
## [25] "company"                        "days_in_waiting_list"
## [27] "customer_type"                  "adr"
## [29] "required_car_parking_spaces"    "total_of_special_requests"
## [31] "reservation_status"             "reservation_status_date"
```

```
# Check for missing values in the 'market_segment' column
sum(is.na(hotel_bookings$market_segment))
```

```
## [1] 0
```

```
# Check for any unique or irrelevant values in the 'market_segment' column
unique(hotel_bookings$market_segment)
```

```
## [1] "Direct"        "Corporate"     "Online TA"     "Offline TA/TO"
## [5] "Complementary" "Groups"        "Undefined"     "Aviation"
```

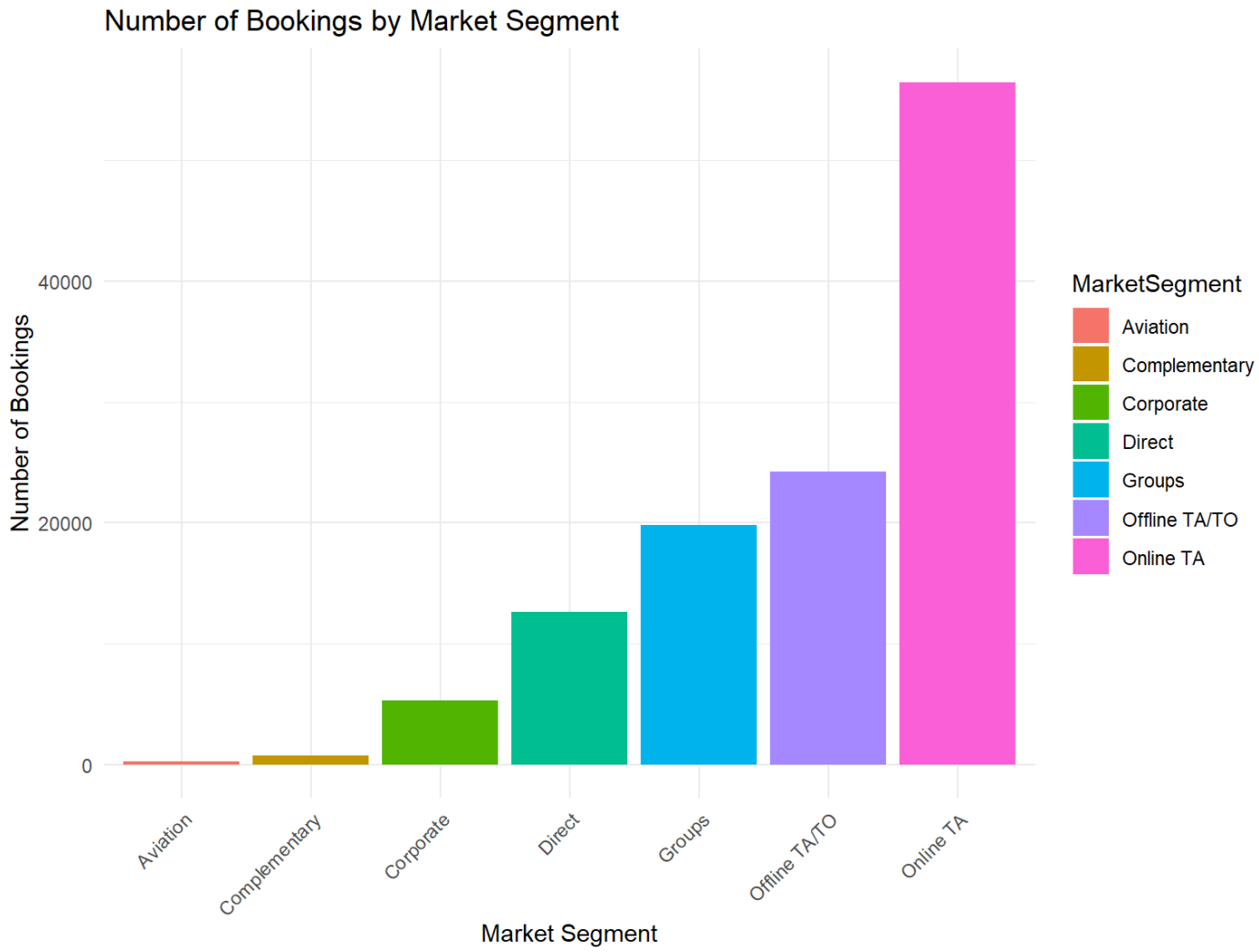
```
# Remove rows where 'market_segment' is "Undefined"
hotel_bookings_clean <- hotel_bookings[hotel_bookings$market_segment != "Undefined", ]
```

```
# Count the number of bookings per market segment
table(hotel_bookings_clean$market_segment)
```

```
##
##      Aviation Complementary     Corporate        Direct        Groups
##           237           743          5295         12606         19811
## Offline TA/TO     Online TA
##         24219         56477
```

```
# Create a data frame with counts for plotting
library(ggplot2)
market_segment_counts <- as.data.frame(table(hotel_bookings_clean$market_segment))
colnames(market_segment_counts) <- c("MarketSegment", "Count")
```

```
# Plot the bar chart
ggplot(market_segment_counts, aes(x = MarketSegment, y = Count, fill = MarketSegment)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Number of Bookings by Market Segment",
       x = "Market Segment",
       y = "Number of Bookings") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Load necessary libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
        library(ggplot2)


        # Convert reservation_status_date to Date type
        hotel_bookings_clean$reservation_status_date <- as.Date(hotel_bookings_clean$reservation_status_date,
format="%m/%d/%Y")

        # Check if the conversion was successful
        head(hotel_bookings_clean$reservation_status_date)
```

```
## [1] NA NA NA NA NA NA
```

```
        # Extract year and month
        hotel_bookings_clean <- hotel_bookings_clean %>%
          mutate(YearMonth = format(reservation_status_date, "%Y-%m"))

        # Aggregate the number of bookings per market segment by month
        booking_trends <- hotel_bookings_clean %>%
          group_by(YearMonth, market_segment) %>%
          summarise(BookingCount = n(), .groups = 'drop')

        # Convert YearMonth to a factor with the correct levels
        booking_trends$YearMonth <- factor(booking_trends$YearMonth, levels = unique(booking_trends$YearMonth))

        # Check the aggregated data
        print(head(booking_trends))
```

```
## # A tibble: 6 × 3
##   YearMonth market_segment BookingCount
##   <fct>     <chr>                 <int>
## 1 <NA>      Aviation                237
## 2 <NA>      Complementary           743
## 3 <NA>      Corporate              5295
## 4 <NA>      Direct                12606
## 5 <NA>      Groups                19811
## 6 <NA>      Offline TA/TO         24219
```

```
        print(summary(booking_trends))
```

```
##  YearMonth market_segment      BookingCount
##  NA's:7    Length:7           Min.   :  237
##            Class :character   1st Qu.: 3019
##            Mode  :character   Median :12606
##                               Mean   :17055
##                               3rd Qu.:22015
##                               Max.   :56477
```

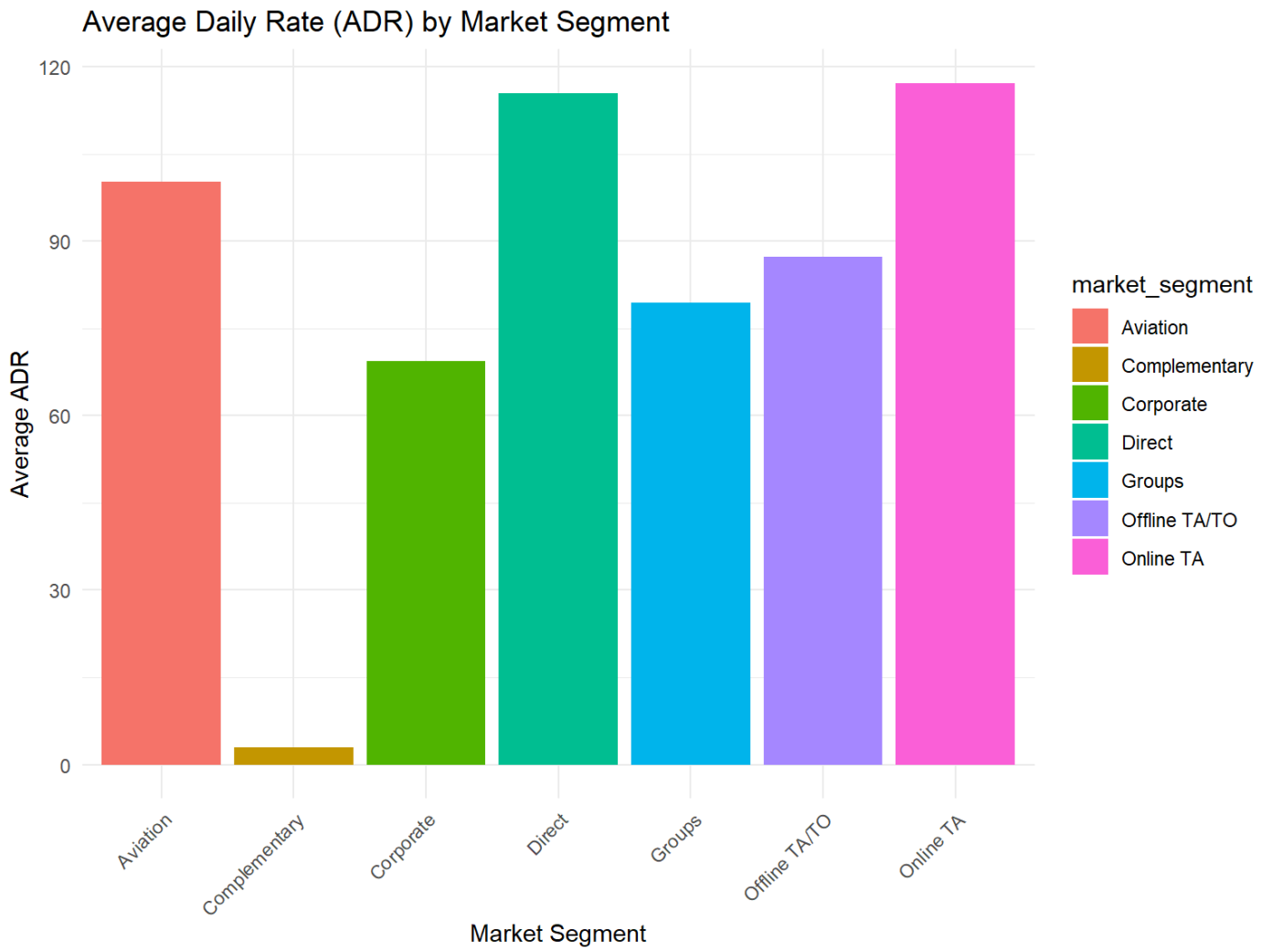
```
        print(table(booking_trends$YearMonth))
```

```
## < table of extent 0 >
```

```
        print(table(booking_trends$market_segment))
```

```
##
##      Aviation Complementary     Corporate        Direct        Groups
##             1             1             1             1             1
## Offline TA/TO     Online TA
##             1             1
```

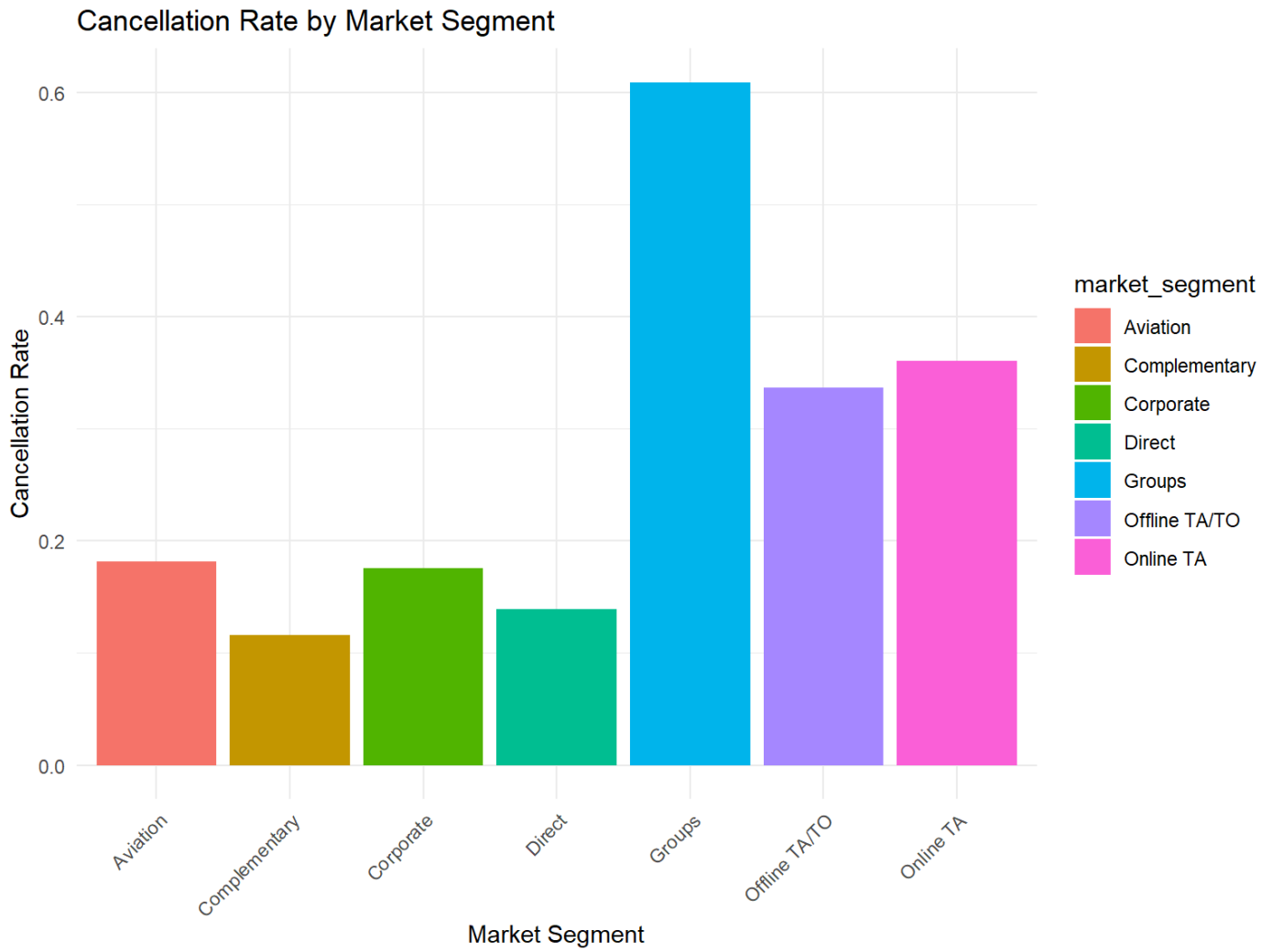
```
# Aggregate average ADR by market segment
adr_comparison <- hotel_bookings_clean %>%
  group_by(market_segment) %>%
  summarise(AverageADR = mean(adr, na.rm = TRUE))

# Plot the average ADR by market segment
ggplot(adr_comparison, aes(x = market_segment, y = AverageADR, fill = market_segment)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Daily Rate (ADR) by Market Segment",
       x = "Market Segment",
       y = "Average ADR") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Check unique values in the reservation_status column
unique(hotel_bookings_clean$reservation_status)
```

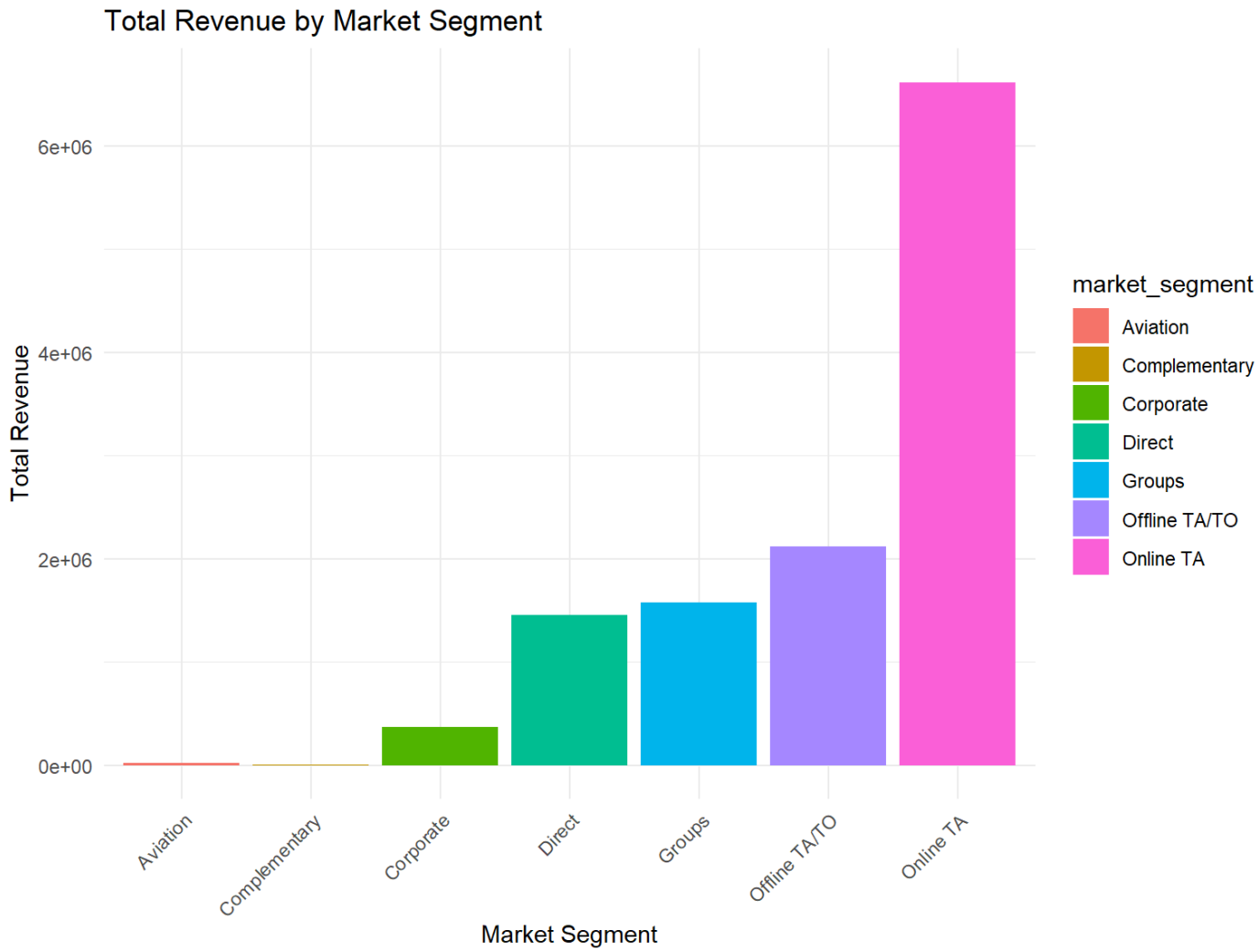
```
## [1] "Check-Out" "Canceled"  "No-Show"
```

```
library(tidyr)
# Create a summary table with cancellations and check-outs
cancellation_summary <- hotel_bookings_clean %>%
  group_by(market_segment, reservation_status) %>%
  summarise(BookingCount = n(), .groups = 'drop') %>%
  pivot_wider(names_from = reservation_status, values_from = BookingCount, values_fill = list(BookingCount
= 0))

# Calculate cancellation rate
cancellation_summary$CancellationRate <- with(cancellation_summary, Canceled / (`Check-Out` + Canceled))

# Plot cancellation rates
ggplot(cancellation_summary, aes(x = market_segment, y = CancellationRate, fill = market_segment)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Cancellation Rate by Market Segment",
       x = "Market Segment",
       y = "Cancellation Rate") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



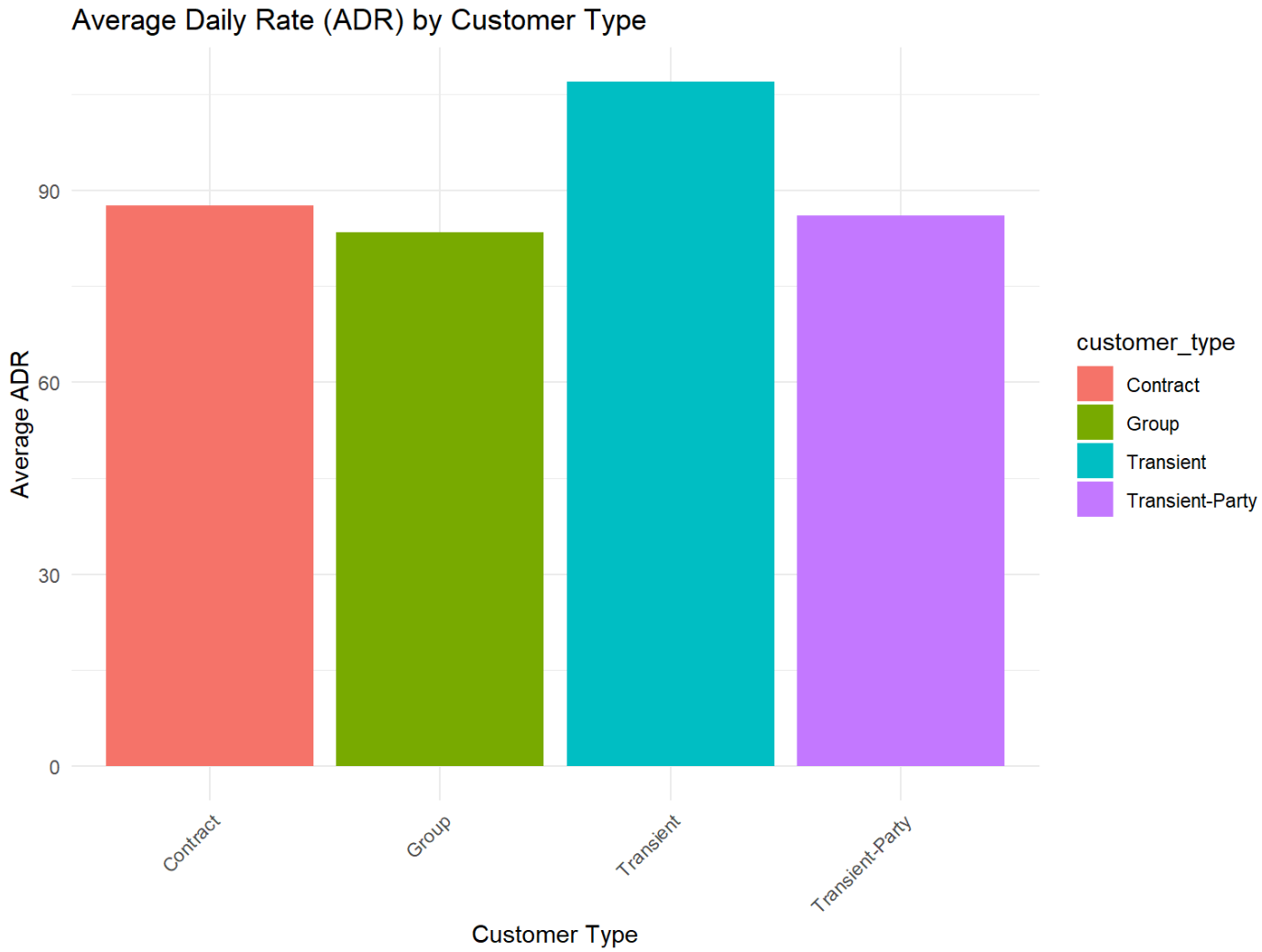

```
cancellation_summary$CancellationRate <- with(cancellation_summary, Canceled / (`Check-Out` + Canceled))
```

```r
# Calculate total revenue by market segment
revenue_by_segment <- hotel_bookings_clean %>%
  group_by(market_segment) %>%
  summarise(TotalRevenue = sum(adr, na.rm = TRUE))

# Plot total revenue by market segment
ggplot(revenue_by_segment, aes(x = market_segment, y = TotalRevenue, fill = market_segment)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Total Revenue by Market Segment",
       x = "Market Segment",
       y = "Total Revenue") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

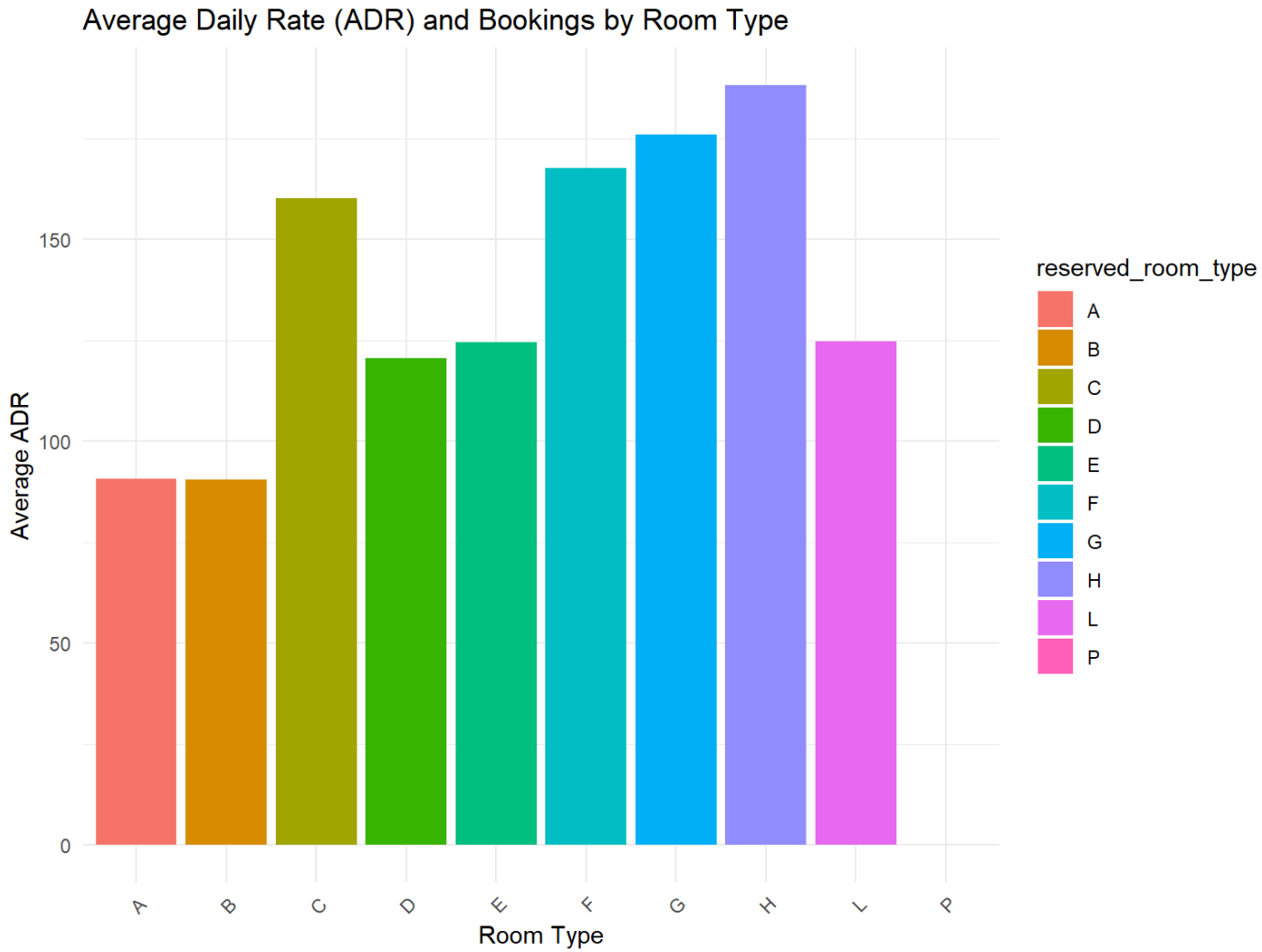



```r
# Aggregate ADR by customer type
adr_by_customer_type <- hotel_bookings_clean %>%
  group_by(customer_type) %>%
  summarise(AverageADR = mean(adr, na.rm = TRUE))

# Plot ADR by customer type
ggplot(adr_by_customer_type, aes(x = customer_type, y = AverageADR, fill = customer_type)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Daily Rate (ADR) by Customer Type",
```

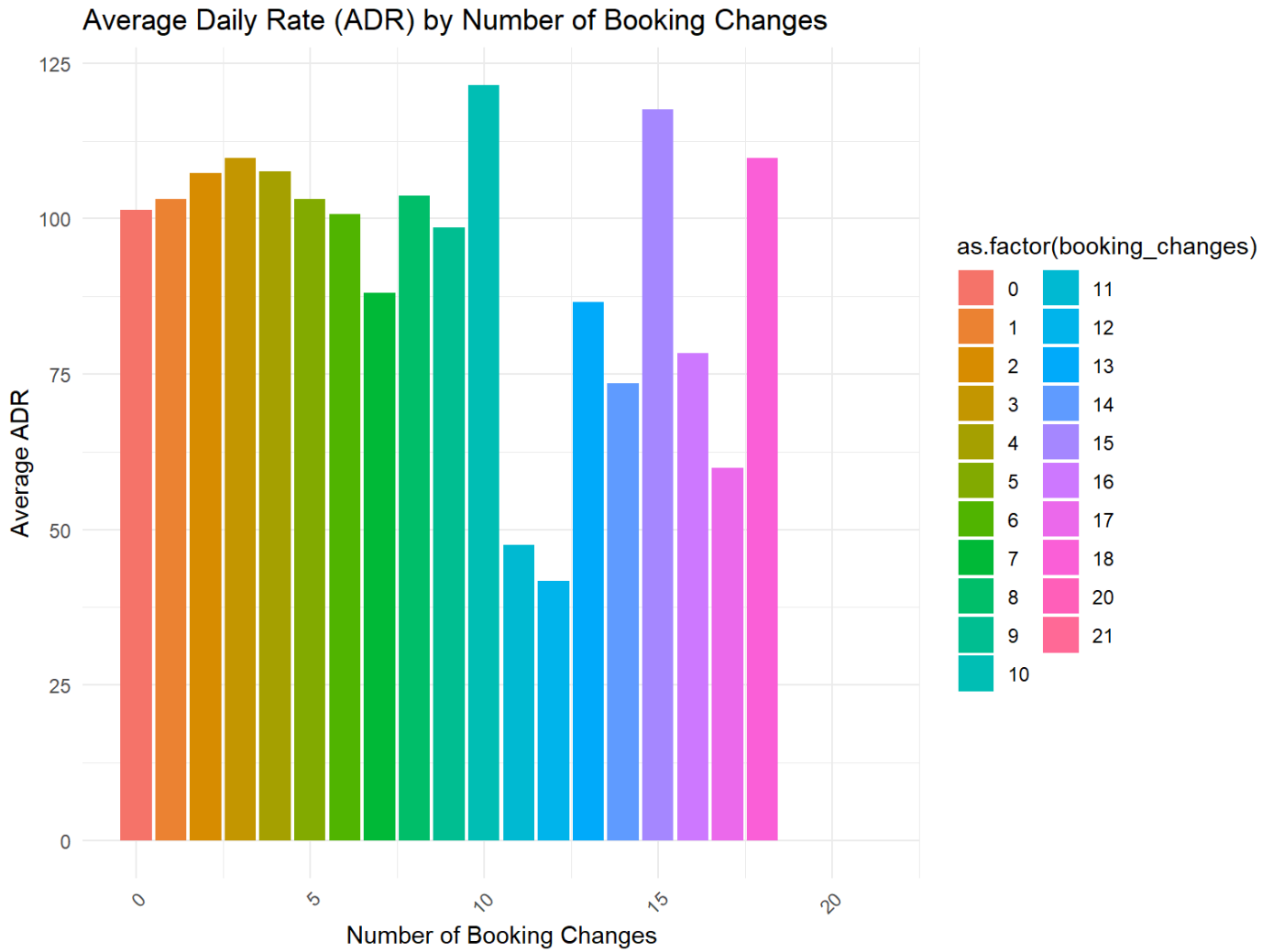
```
       x = "Customer Type",
       y = "Average ADR") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Aggregate ADR by room type
adr_by_room_type <- hotel_bookings_clean %>%
  group_by(reserved_room_type) %>%
  summarise(AverageADR = mean(adr, na.rm = TRUE),
            TotalBookings = n())

# Plot ADR and bookings by room type
ggplot(adr_by_room_type, aes(x = reserved_room_type, y = AverageADR, fill = reserved_room_type)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Daily Rate (ADR) and Bookings by Room Type",
       x = "Room Type",
       y = "Average ADR") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Aggregate ADR by the number of booking changes
adr_by_changes <- hotel_bookings_clean %>%
  group_by(booking_changes) %>%
  summarise(AverageADR = mean(adr, na.rm = TRUE))

# Plot ADR by booking changes
ggplot(adr_by_changes, aes(x = booking_changes, y = AverageADR, fill = as.factor(booking_changes))) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Daily Rate (ADR) by Number of Booking Changes",
       x = "Number of Booking Changes",
       y = "Average ADR") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Average Daily Rate (ADR) by Number of Booking Changes
Average ADR
125
100
75
50
25
0
0
5
10
15
20
Number of Booking Changes
as.factor(booking_changes)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
20
21