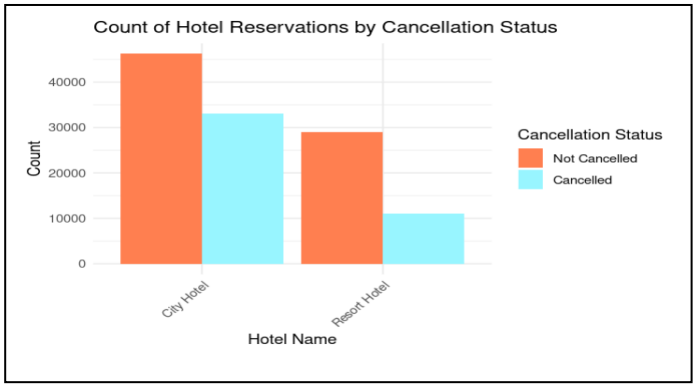
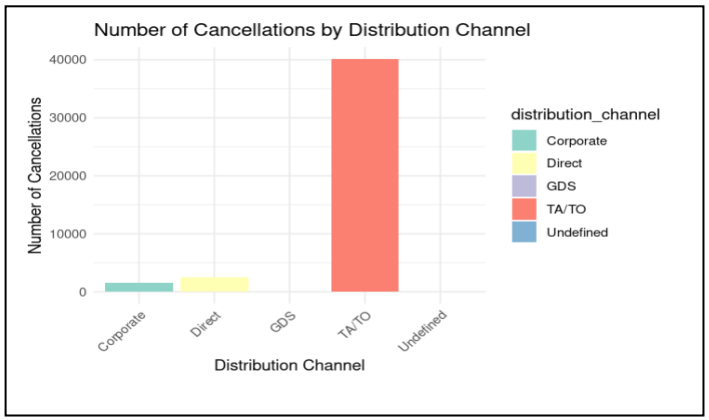
```
##install.packages(c("shiny", "ggplot2", "dplyr"))

library(shiny)
library(ggplot2)
library(dplyr)

# Define UI
ui <- fluidPage(
  tags$head(
    tags$style(HTML("
      .centered {
        text-align: center;
      }
    "))
  ),

  titlePanel("Hotel Cancellation Analytics Dashboard"),

  # Remove sidebarLayout, use a single layout instead
  fluidRow(
    # Add the help text on top
    column(4,
           helpText(
                    "This dashboard presents an analysis of hotel cancellations based on data from the Hotel Booking Demand
Datasets by Nuno Antonio, Ana Almeida, and Luis Nunes."),
           # Add some space between the help text and the plots
           HTML("<div style='height: 30px;'></div>")
    ),
    column(8, plotOutput("hotelCancellationsPlot"))
  ),

  fluidRow(
    column(4, plotOutput("distributionChannelPlot")),
    column(4, plotOutput("topAgentsPlot")),
    column(4, plotOutput("monthlyCancellationsPlot"))
  ),
  fluidRow(
    column(6, plotOutput("avgDaysWaitingPlot")),
    column(6, plotOutput("avgDailyRatePlot"))
  ),
  fluidRow(
    column(12, plotOutput("depositTypesPlot"))
  )
)

# Define server logic
server <- function(input, output) {
```

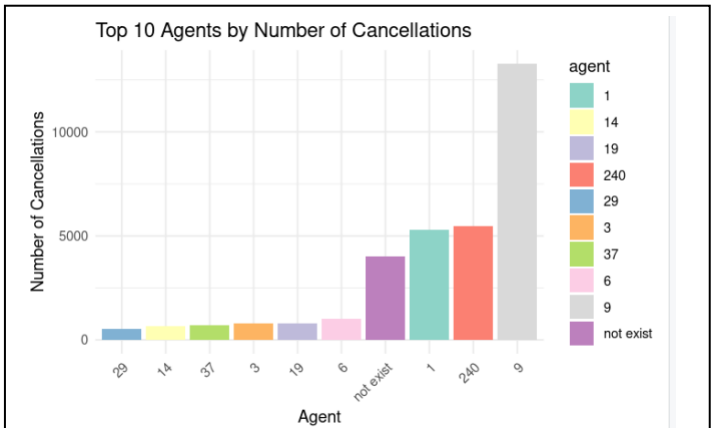
```
# Cancellations by Hotel
output$hotelCancellationsPlot <- renderPlot({
  hotel_counts <- hotel_bookings %>%
    group_by(hotel, is_canceled) %>%
    summarise(count = n(), .groups = 'drop')

  ggplot(data = hotel_counts, aes(x = hotel, y = count, fill = factor(is_canceled))) +
    geom_bar(stat = "identity", position = "dodge") +
    labs(title = "Count of Hotel Reservations by Cancellation Status",
         x = "Hotel Name", y = "Count", fill = "Cancellation Status") +
    scale_fill_manual(values = c("0" = "coral", "1" = "cadetblue1"),
                      labels = c("Not Cancelled", "Cancelled")) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
})
```



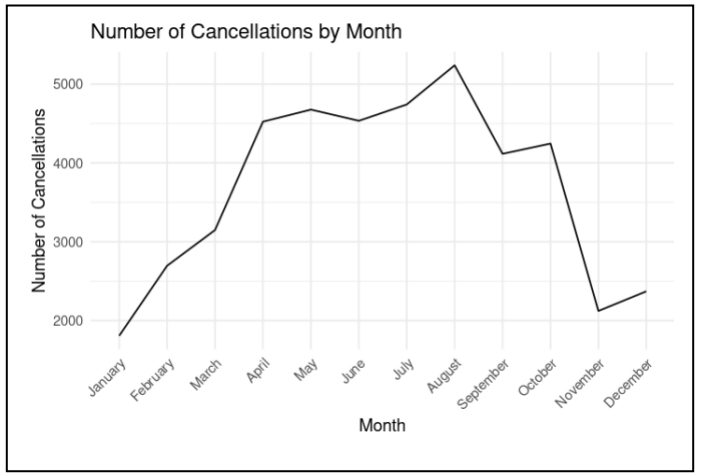

```
# Cancellations by Distribution Channel
output$distributionChannelPlot <- renderPlot({
  canc lled_counts <- hotel_bookings %>%
    filter(is_canceled == 1) %>%
    group_by(distribution_channel) %>%
    summarise(count = n(), .groups = 'drop')

  ggplot(data = cancelled_counts) +
    geom_bar(mapping = aes(x = distribution_channel, y = count, fill = distribution_channel),
             stat = "identity") +
    labs(title = "Number of Cancellations by Distribution Channel",
         x = "Distribution Channel", y = "Number of Cancellations") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    scale_fill_brewer(palette = "Set3")
})
```

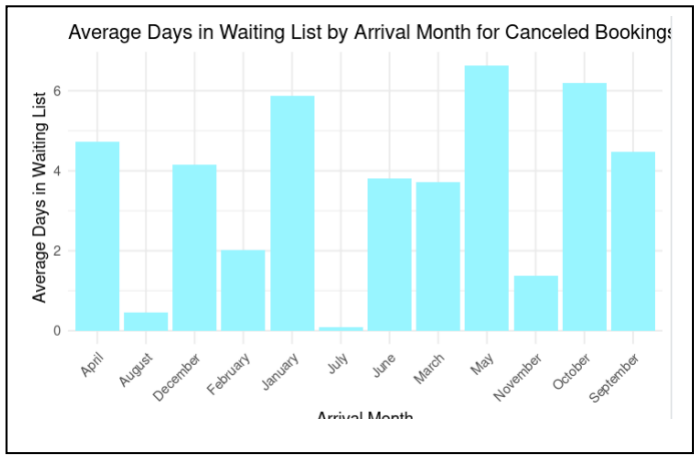



```
# Top 10 Agents for Cancellations
output$topAgentsPlot <- renderPlot({
  cancelled_counts <- hotel_bookings %>%
    filter(is_canceled == 1) %>%
    group_by(agent) %>%
    summarise(count = n(), .groups = 'drop') %>%
    top_n(10, count) %>%
    arrange(desc(count))

  ggplot(data = cancelled_counts) +
    geom_bar(mapping = aes(x = reorder(agent, count), y = count, fill = agent),
             stat = "identity") +
    labs(title = "Top 10 Agents by Number of Cancellations",
         x = "Agent", y = "Number of Cancellations") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
```



```
    scale_fill_brewer(palette = "Set3")
})

# Cancellations by Month
output$monthlyCancellationsPlot <- renderPlot({
  cancelled_counts <- hotel_bookings %>%
    filter(is_canceled == 1) %>%
    group_by(arrival_date_month) %>%
    summarise(count = n(), .groups = 'drop')

  cancelled_counts$arrival_date_month <- factor(cancelled_counts$arrival_date_month,
                                                levels = month.name)

  ggplot(data = cancelled_counts) +
    geom_line(mapping = aes(x = arrival_date_month, y = count, group = 1)) +
    labs(title = "Number of Cancellations by Month",
         x = "Month", y = "Number of Cancellations") +
    theme_minimal() +
    th me(axis.text.x = element_text(angle = 45, hjust = 1))
})

# Avg Days in Waiting List for Cancellations
output$avgDaysWaitingPlot <- renderPlot({
  cancelled_counts <- hotel_bookings %>%
    filter(is_canceled == 1) %>%
    group_by(arrival_date_month) %>%
    summarise(avg_days_in_waiting_list = mean(days_in_waiting_list, na.rm = TRUE), .groups = 'drop')

  cancelled_counts$arrival_date_month <- factor(cancelled_counts$arrival_date_month, levels = month.name)

  ggplot(cancelled_counts, aes(x = arrival_date_month, y = avg_days_in_waiting_list)) +
    geom_bar(stat = "identity", fill = "cadetblue1") +
    labs(title = "Average Days in Waiting List by Arrival Month for Canceled Bookings",
         x = "Arrival Month", y = "Average Days in Waiting List") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
})

# Avg Daily Rate for Cancellations
output$avgDailyRatePlot <- renderPlot({
  cancelled_counts <- hotel_bookings %>%
    filter(is_canceled == 1) %>%
    group_by(arrival_date_month) %>%
    summarise(avg_adr = mean(adr, na.rm = TRUE), .groups = 'drop')

  cancelled_counts$arrival_date_month <- factor(cancelled_counts$arrival_date_month, levels = month.name)

  ggplot(cancelled_counts, aes(x = arrival_date_month, y = avg_adr)) +
    geom_bar(stat = "identity", fill = "coral") +
```





```
      labs(title = "Average Daily Rate by Month for Canceled Bookings",
           x = "Arrival Month", y = "Average Daily Rate") +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
  })

  # Deposit Types Distribution
  output$depositTypesPlot <- renderPlot({
    deposit_counts <- hotel_bookings %>%
      filter(is_canceled == 1) %>%
      group_by(deposit_type) %>%
      summarise(count = n(), .groups = 'drop')

    ggplot(deposit_counts, aes(x = "", y = count, fill = deposit_type)) +
      geom_bar(stat = "identity", width = 1) +
      coord_polar(theta = "y") +
      labs(title = "Distribution of Deposit Types for Canceled Bookings",
           x = NULL, y = NULL) +
      th me_void() +
      theme(legend.title = element_blank())
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```