

Data Visualisation with R - Project

AfterWork Data Science: Data Visualisation with R Project

1. Defining the Question

a) Specifying the Data Analysis Question

Specify the research question that you'll be answering. i.e. Provide strategy recommendations that will lead to revenue growth.

b) Defining the Metric for Success

The solutions to the following questions will help us answer our research question:

- When is the best time of year to book a hotel room?
- When is the optimal length of stay in order to get the best daily rate?
- How will you know if a hotel was likely to receive a disproportionately high number of special requests?

c) Understanding the context

Provide some background information. . . .

d) Recording the Experimental Design

Describe the steps/approach that you will use to answer the given question.

e) Data Relevance

How relevant was the provided data?

2. Reading the Data

```
# Load the data below
# ---
# Dataset url =
# ---
# YOUR CODE GOES BELOW
#
library(readr)
hotel_bookings <- read_csv("hotel_bookings.csv")

## Rows: 119390 Columns: 32
## -- Column specification -----
## Delimiter: ","
## chr  (13): hotel, arrival_date_month, meal, country, market_segment, distrib...
## dbl  (18): is_canceled, lead_time, arrival_date_year, arrival_date_week_numb...
## date  (1): reservation_status_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```

library(readr)
hotel_bookings <- read_csv("hotel_bookings.csv")

## Rows: 119390 Columns: 32
## -- Column specification -----
## Delimiter: ","
## chr  (13): hotel, arrival_date_month, meal, country, market_segment, distrib...
## dbl  (18): is_canceled, lead_time, arrival_date_year, arrival_date_week_numb...
## date  (1): reservation_status_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
# Checking the first 5 rows of data
# ---
# YOUR CODE GOES BELOW
head(hotel_bookings, 5)

## # A tibble: 5 x 32
##   hotel          is_canceled lead_time arrival_date_year arrival_date_month
##   <chr>          <dbl>      <dbl>          <dbl> <chr>
## 1 Resort Hotel      0        342            2015 July
## 2 Resort Hotel      0        737            2015 July
## 3 Resort Hotel      0         7            2015 July
## 4 Resort Hotel      0        13            2015 July
## 5 Resort Hotel      0        14            2015 July
## # i 27 more variables: arrival_date_week_number <dbl>,
## #   arrival_date_day_of_month <dbl>, stays_in_weekend_nights <dbl>,
## #   stays_in_week_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,
## #   meal <chr>, country <chr>, market_segment <chr>,
## #   distribution_channel <chr>, is_repeated_guest <dbl>,
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
## #   reserved_room_type <chr>, assigned_room_type <chr>, ...
head(hotel_bookings,5)

## # A tibble: 5 x 32
##   hotel          is_canceled lead_time arrival_date_year arrival_date_month
##   <chr>          <dbl>      <dbl>          <dbl> <chr>
## 1 Resort Hotel      0        342            2015 July
## 2 Resort Hotel      0        737            2015 July
## 3 Resort Hotel      0         7            2015 July
## 4 Resort Hotel      0        13            2015 July
## 5 Resort Hotel      0        14            2015 July
## # i 27 more variables: arrival_date_week_number <dbl>,
## #   arrival_date_day_of_month <dbl>, stays_in_weekend_nights <dbl>,
## #   stays_in_week_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,
## #   meal <chr>, country <chr>, market_segment <chr>,
## #   distribution_channel <chr>, is_repeated_guest <dbl>,
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
## #   reserved_room_type <chr>, assigned_room_type <chr>, ...

#print(hotel_bookings)

```

```
# Checking the first 5 rows of data
```

```
# ---
```

```
# YOUR CODE GOES BELOW
```

```
#
```

```
head(hotel_bookings,5)
```

```
## # A tibble: 5 x 32
```

```
##   hotel      is_canceled lead_time arrival_date_year arrival_date_month
```

```
##   <chr>          <dbl>    <dbl>          <dbl> <chr>
```

```
## 1 Resort Hotel      0      342          2015 July
```

```
## 2 Resort Hotel      0      737          2015 July
```

```
## 3 Resort Hotel      0        7          2015 July
```

```
## 4 Resort Hotel      0       13          2015 July
```

```
## 5 Resort Hotel      0       14          2015 July
```

```
## # i 27 more variables: arrival_date_week_number <dbl>,
```

```
## #   arrival_date_day_of_month <dbl>, stays_in_weekend_nights <dbl>,
```

```
## #   stays_in_week_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,
```

```
## #   meal <chr>, country <chr>, market_segment <chr>,
```

```
## #   distribution_channel <chr>, is_repeated_guest <dbl>,
```

```
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
```

```
## #   reserved_room_type <chr>, assigned_room_type <chr>, ...
```

```
# Checking the last 5 rows of data
```

```
# ---
```

```
# YOUR CODE GOES BELOW
```

```
#
```

```
tail(hotel_bookings, 5)
```

```
## # A tibble: 5 x 32
```

```
##   hotel      is_canceled lead_time arrival_date_year arrival_date_month
```

```
##   <chr>          <dbl>    <dbl>          <dbl> <chr>
```

```
## 1 City Hotel      0       23          2017 August
```

```
## 2 City Hotel      0      102          2017 August
```

```
## 3 City Hotel      0       34          2017 August
```

```
## 4 City Hotel      0      109          2017 August
```

```
## 5 City Hotel      0      205          2017 August
```

```
## # i 27 more variables: arrival_date_week_number <dbl>,
```

```
## #   arrival_date_day_of_month <dbl>, stays_in_weekend_nights <dbl>,
```

```
## #   stays_in_week_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,
```

```
## #   meal <chr>, country <chr>, market_segment <chr>,
```

```
## #   distribution_channel <chr>, is_repeated_guest <dbl>,
```

```
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
```

```
## #   reserved_room_type <chr>, assigned_room_type <chr>, ...
```

```
#print(hotel_bookings)
```

```
# Sample 10 rows of data
```

```
# ---
```

```
# YOUR CODE GOES BELOW
```

```
#
```

```
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
set.seed(123) # Set seed for reproducibility
hotel_bookings %>% sample_n(10)

## # A tibble: 10 x 32
##   hotel      is_canceled lead_time arrival_date_year arrival_date_month
##   <chr>          <dbl>    <dbl>          <dbl> <chr>
## 1 City Hotel      1      158            2016 May
## 2 City Hotel      1       81            2016 October
## 3 Resort Hotel    0       79            2015 November
## 4 Resort Hotel    0       49            2016 November
## 5 City Hotel      0        9            2016 August
## 6 City Hotel      0        4            2016 December
## 7 City Hotel      1     104            2017 May
## 8 City Hotel      1     253            2017 January
## 9 City Hotel      0       72            2015 October
## 10 City Hotel     1       38            2017 March
## # i 27 more variables: arrival_date_week_number <dbl>,
## #   arrival_date_day_of_month <dbl>, stays_in_weekend_nights <dbl>,
## #   stays_in_week_nights <dbl>, adults <dbl>, children <dbl>, babies <dbl>,
## #   meal <chr>, country <chr>, market_segment <chr>,
## #   distribution_channel <chr>, is_repeated_guest <dbl>,
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>,
## #   reserved_room_type <chr>, assigned_room_type <chr>, ...
# Checking number of rows and columns
# ---
# YOUR CODE GOES BELOW
#

exists("hotel_bookings") # checking in the file exists

## [1] TRUE
print(paste("Printing nrow and ncol:", dim(hotel_bookings)))

## [1] "Printing nrow and ncol: 119390" "Printing nrow and ncol: 32"
print(paste('Printing nrow:', nrow(hotel_bookings)))

## [1] "Printing nrow: 119390"
print(paste('Printing ncol:', ncol(hotel_bookings)))

## [1] "Printing ncol: 32"
# Checking datatypes
# ---
# YOUR CODE GOES BELOW
#
str(hotel_bookings)

```

```

## spc_tbl_ [119,390 x 32] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ hotel : chr [1:119390] "Resort Hotel" "Resort Hotel" "Resort Hotel" "Reso
## $ is_canceled : num [1:119390] 0 0 0 0 0 0 0 0 1 1 ...
## $ lead_time : num [1:119390] 342 737 7 13 14 14 0 9 85 75 ...
## $ arrival_date_year : num [1:119390] 2015 2015 2015 2015 2015 ...
## $ arrival_date_month : chr [1:119390] "July" "July" "July" "July" ...
## $ arrival_date_week_number : num [1:119390] 27 27 27 27 27 27 27 27 27 27 ...
## $ arrival_date_day_of_month : num [1:119390] 1 1 1 1 1 1 1 1 1 1 ...
## $ stays_in_weekend_nights : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ stays_in_week_nights : num [1:119390] 0 0 1 1 2 2 2 2 3 3 ...
## $ adults : num [1:119390] 2 2 1 1 2 2 2 2 2 2 ...
## $ children : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ babies : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ meal : chr [1:119390] "BB" "BB" "BB" "BB" ...
## $ country : chr [1:119390] "PRT" "PRT" "GBR" "GBR" ...
## $ market_segment : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
## $ distribution_channel : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
## $ is_repeated_guest : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_cancellations : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_bookings_not_canceled : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ reserved_room_type : chr [1:119390] "C" "C" "A" "A" ...
## $ assigned_room_type : chr [1:119390] "C" "C" "C" "A" ...
## $ booking_changes : num [1:119390] 3 4 0 0 0 0 0 0 0 0 ...
## $ deposit_type : chr [1:119390] "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
## $ agent : chr [1:119390] "NULL" "NULL" "NULL" "304" ...
## $ company : chr [1:119390] "NULL" "NULL" "NULL" "NULL" ...
## $ days_in_waiting_list : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ customer_type : chr [1:119390] "Transient" "Transient" "Transient" "Transient" ..
## $ adr : num [1:119390] 0 0 75 75 98 ...
## $ required_car_parking_spaces : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
## $ total_of_special_requests : num [1:119390] 0 0 0 0 1 1 0 1 1 0 ...
## $ reservation_status : chr [1:119390] "Check-Out" "Check-Out" "Check-Out" "Check-Out" ..
## $ reservation_status_date : Date[1:119390], format: "2015-07-01" "2015-07-01" ...
## - attr(*, "spec")=
## .. cols(
## .. hotel = col_character(),
## .. is_canceled = col_double(),
## .. lead_time = col_double(),
## .. arrival_date_year = col_double(),
## .. arrival_date_month = col_character(),
## .. arrival_date_week_number = col_double(),
## .. arrival_date_day_of_month = col_double(),
## .. stays_in_weekend_nights = col_double(),
## .. stays_in_week_nights = col_double(),
## .. adults = col_double(),
## .. children = col_double(),
## .. babies = col_double(),
## .. meal = col_character(),
## .. country = col_character(),
## .. market_segment = col_character(),
## .. distribution_channel = col_character(),
## .. is_repeated_guest = col_double(),
## .. previous_cancellations = col_double(),
## .. previous_bookings_not_canceled = col_double(),

```

```
## .. reserved_room_type = col_character(),
## .. assigned_room_type = col_character(),
## .. booking_changes = col_double(),
## .. deposit_type = col_character(),
## .. agent = col_character(),
## .. company = col_character(),
## .. days_in_waiting_list = col_double(),
## .. customer_type = col_character(),
## .. adr = col_double(),
## .. required_car_parking_spaces = col_double(),
## .. total_of_special_requests = col_double(),
## .. reservation_status = col_character(),
## .. reservation_status_date = col_date(format = "")
## .. )
## - attr(*, "problems")=<externalptr>
```

```
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"
```

Record your general observations below:

Observation 1 Observation 2

The dataset contains **119,390 rows** and **32 columns** with a variety of information about hotel bookings, such as **cancellation status**, **lead time**, **arrival dates**, etc.

The dataset includes different data types: **character (chr)**, **numeric (dbl)**, and **date**.

Some columns have **missing values (Null Values)**, such as the **agent** and **company** columns. It's important to handle these missing values carefully, either by **imputation** (filling missing data with a calculated value) or **exclusion** (removing rows with missing values).

Additionally, some columns need to be **converted into the correct type** for proper analysis. For example, the **children**, **adults**, and **babies** columns should be converted into integers, as they currently might be in character or numeric formats.

To better answer questions related to **cancellation prediction** or **customer behavior analysis**, we may need additional relevant data points, such as **customer reviews**, **prices**, for each booking. These columns are missing from the dataset but could be vital for improving the analysis.

3. External Data Source Validation

The data is originally from the article Hotel Booking Demand Datasets, by Nuno Antonio, Ana Almeida, and Luis Nunes for Data in Brief, Volume 22, February 2019.

4. Data Preparation

Performing Data Cleaning

```
# Checking datatypes and missing entries of all the variables
# ---
# YOUR CODE GOES BELOW
#
#print(paste('printing the datatypes of the column hotel:',typeof(hotel_bookings$hotel)))
#print(paste('Printing column names:',colnames(hotel_bookings)))

#Checking datatypes
sapply(hotel_bookings, class)

##              hotel              is_canceled
##      "character"              "numeric"
##      lead_time      arrival_date_year
##      "numeric"              "numeric"
##      arrival_date_month      arrival_date_week_number
##      "character"              "numeric"
##      arrival_date_day_of_month      stays_in_weekend_nights
##      "numeric"              "numeric"
##      stays_in_week_nights      adults
##      "numeric"              "numeric"
##      children      babies
##      "numeric"              "numeric"
##      meal      country
##      "character"              "character"
##      market_segment      distribution_channel
##      "character"              "character"
##      is_repeated_guest      previous_cancellations
##      "numeric"              "numeric"
##      previous_bookings_not_canceled      reserved_room_type
##      "numeric"              "character"
##      assigned_room_type      booking_changes
##      "character"              "numeric"
##      deposit_type      agent
##      "character"              "character"
##      company      days_in_waiting_list
##      "character"              "numeric"
##      customer_type      adr
##      "character"              "numeric"
##      required_car_parking_spaces      total_of_special_requests
##      "numeric"              "numeric"
##      reservation_status      reservation_status_date
##      "character"              "Date"

#or
#str(hotel_bookings)
```

```
#missing entries of all the variables
colSums(is.na(hotel_bookings))
```

```
##           hotel           is_canceled
##           0           0
##           lead_time       arrival_date_year
##           0           0
##           arrival_date_month arrival_date_week_number
##           0           0
##           arrival_date_day_of_month stays_in_weekend_nights
##           0           0
##           stays_in_week_nights adults
##           0           0
##           children babies
##           4           0
##           meal country
##           0           0
##           market_segment distribution_channel
##           0           0
##           is_repeated_guest previous_cancellations
##           0           0
## previous_bookings_not_canceled reserved_room_type
##           0           0
##           assigned_room_type booking_changes
##           0           0
##           deposit_type agent
##           0           0
##           company days_in_waiting_list
##           0           0
##           customer_type adr
##           0           0
##           required_car_parking_spaces total_of_special_requests
##           0           0
##           reservation_status reservation_status_date
##           0           0
```

```
#or
sum(is.na(hotel_bookings))
```

```
## [1] 4
```

We observe the following from our dataset:

The childrens columns has the most missing entries with a total of 4(PS: we did not consider the NULL entries)

The data type has been discussed above

Observation 1 Observation 2

```
# Checking how many duplicate rows are there in the data
# ---
# YOUR CODE GOES BELOW
#
```



```

duplicated_rows <- (duplicated(hotel_bookings))

```

```

#print(duplicated_rows)
print(sum(duplicated_rows))

```

```
## [1] 31994
```

```

duplicate_rows <- hotel_bookings[duplicated(hotel_bookings), ]
#print(duplicate_rows)

```

We choose to keep the duplicates because we don't have a unique identifier to actually proof that we have duplicates.

```

# Checking if any of the columns are all null
# ---
# YOUR CODE GOES BELOW
#
#

```

```
colSums(is.na(hotel_bookings)) == nrow(hotel_bookings)
```

```

##             hotel             is_canceled
##             FALSE             FALSE
##             lead_time         arrival_date_year
##             FALSE             FALSE
##             arrival_date_month arrival_date_week_number
##             FALSE             FALSE
##             arrival_date_day_of_month stays_in_weekend_nights
##             FALSE             FALSE
##             stays_in_week_nights adults
##             FALSE             FALSE
##             children babies
##             FALSE             FALSE
##             meal country
##             FALSE             FALSE
##             market_segment distribution_channel
##             FALSE             FALSE
##             is_repeated_guest previous_cancellations
##             FALSE             FALSE
## previous_bookings_not_canceled reserved_room_type
##             FALSE             FALSE
##             assigned_room_type booking_changes
##             FALSE             FALSE
##             deposit_type agent
##             FALSE             FALSE
##             company days_in_waiting_list
##             FALSE             FALSE
##             customer_type adr
##             FALSE             FALSE
##             required_car_parking_spaces total_of_special_requests
##             FALSE             FALSE
##             reservation_status reservation_status_date
##             FALSE             FALSE

```

```
colnames(hotel_bookings)[colSums(is.na(hotel_bookings)) == nrow(hotel_bookings)]
```

```
## character(0)
```

We observe the following from our dataset:

Observation 1

The data contains 31,987 duplicated rows (excluding the first occurrence) and does not have any column that is completely empty.

```
# Checking if any of the rows are all null
```

```
# ---
```

```
# YOUR CODE GOES BELOW
```

```
#
```

```
rowSums(is.na(hotel_bookings)) == ncol(hotel_bookings)
```

```
##      [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##     [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## [99601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99637] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99697] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99709] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99733] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99757] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99769] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99781] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99793] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99805] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99817] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99829] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99841] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99853] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99865] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99877] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99889] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99901] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99913] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99925] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99937] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99949] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99961] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99973] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99985] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [99997] FALSE FALSE FALSE
## [ reached getOption("max.print") -- omitted 19391 entries ]
hotel_bookings[rowSums(is.na(hotel_bookings)) == ncol(hotel_bookings), ]
```

```
## # A tibble: 0 x 32
## # i 32 variables: hotel <chr>, is_canceled <dbl>, lead_time <dbl>,
## #   arrival_date_year <dbl>, arrival_date_month <chr>,
## #   arrival_date_week_number <dbl>, arrival_date_day_of_month <dbl>,
## #   stays_in_weekend_nights <dbl>, stays_in_week_nights <dbl>, adults <dbl>,
## #   children <dbl>, babies <dbl>, meal <chr>, country <chr>,
## #   market_segment <chr>, distribution_channel <chr>, is_repeated_guest <dbl>,
## #   previous_cancellations <dbl>, previous_bookings_not_canceled <dbl>, ...
```

We observe the following from our dataset:

Observation 1 Observation 2

The data contains 31,987 duplicated rows (excluding the first occurrence) and does not have any rows that is completely empty.

```
#library(dplyr)
```

```

#install.packages("reshape2")

# Checking the correlation of the features through the use of
# visualizations the correlation using heatmap
# ---
# YOUR CODE GOES BELOW
#

# Load necessary libraries
library(dplyr)
library(ggplot2)
library(reshape2)

# Select numeric columns from the dataset
numeric_data <- hotel_bookings %>%
  select(where(is.numeric))

# Compute the correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")

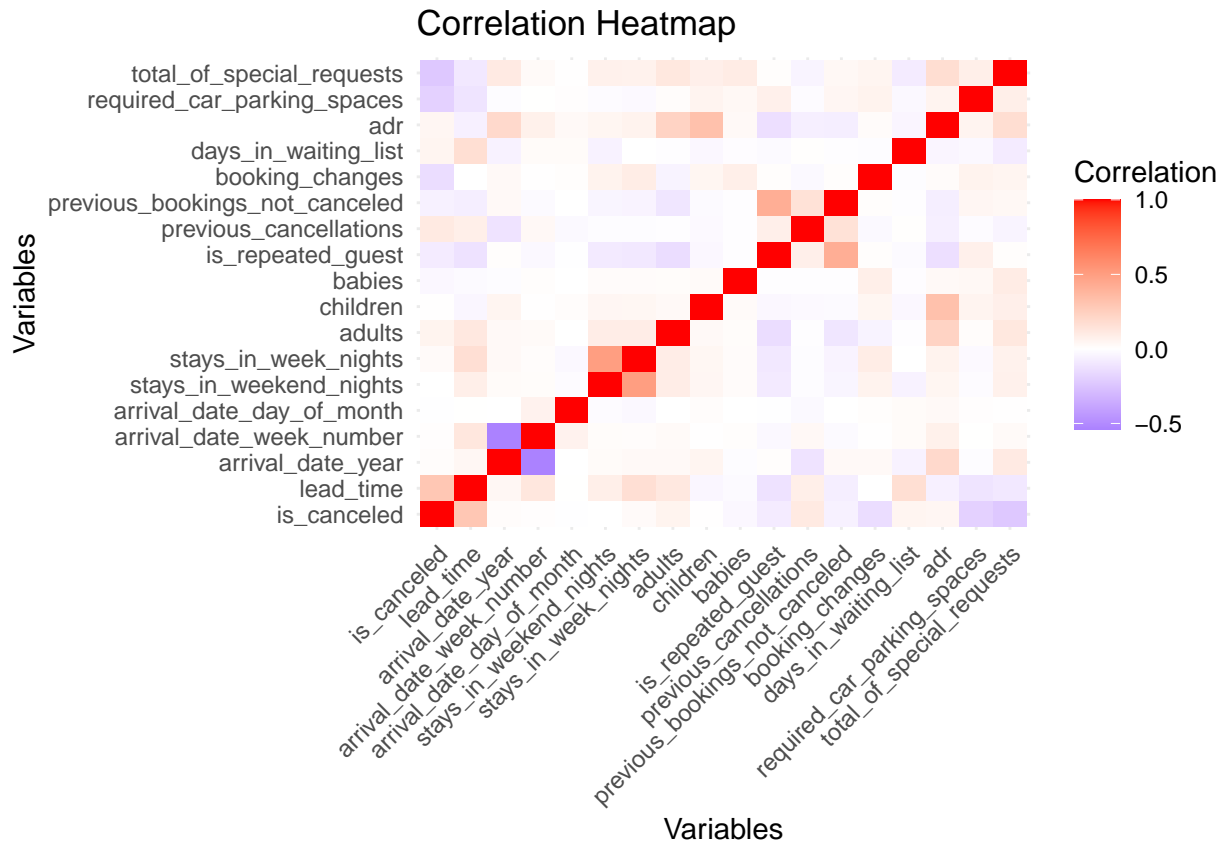
# Reshape the correlation matrix to long format for ggplot
cor_matrix_melted <- melt(cor_matrix)

# Create heatmap using ggplot2
p <- ggplot(cor_matrix_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
  theme_minimal() +
  labs(title = "Correlation Heatmap",
       x = "Variables",
       y = "Variables",
       fill = "Correlation") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Save the figure
#ggsave("correlation_heatmap.png", plot = p, width = 8, height = 6, dpi = 300)

# Show the plot
print(p)

```



We observe the following from our dataset:

Observation 1 Observation 2

Strong Positive Correlations:

We observe a strong positive correlation between variables such as `stays_in_week_nights` and `stays_in_weekend_nights`, indicating that longer stays during weekdays are often associated with longer weekend stays.

Additionally, `previous_bookings_not_canceled` and `is_repeated_guest` also appear positively correlated, suggesting that repeated guests tend to have fewer cancellations in their booking history.

The heatmap shows a positive correlation between `lead_time` and `is_canceled`. This implies that bookings with longer lead times (i.e., made far in advance) are more likely to be canceled. This is a logical finding because guests who book far in advance may face changing circumstances that lead to cancellations.

Weak and no Correlation:

Variables such as `arrival_date_day_of_month`, `arrival_date_week_number`, and `arrival_date_year` show weak correlations with most other variables, suggesting that arrival date information does not strongly influence other factors, such as cancellations or booking behavior.

There is a weak correlation between the lead time and `adr`

Potential Insights for Business Decisions:

`is_canceled` and `lead_time` (Positive Correlation): A positive correlation suggests that bookings made further in advance (longer lead times) are more likely to be canceled.

Business Insight: This is an important factor to consider. Customers who book far in advance may be more likely to find better deals elsewhere, change their plans, or simply forget about the booking.

Actionable Items: Consider implementing strategies to reduce cancellations for long lead times. This could involve sending reminder emails closer to the stay date, offering incentives for keeping the booking (e.g., a small discount on on-site services), and implementing stricter cancellation policies for very long lead times (but be transparent about this).

```
# Dropping company column because it has alot of missing values
# and we won't need to answer any of our questions
# ---
# YOUR CODE GOES BELOW
# I have to comment how I dropped the compagny column because it was given warning as it was not existi

"company" %in% colnames(hotel_bookings)

## [1] TRUE

#is.na(hotel_bookings$company)
#print(sum(is.na(hotel_bookings$company)))

#print(hotel_bookings$company)
print(ncol(hotel_bookings))

## [1] 32

#sum(hotel_bookings$company == "NULL")

#hotel_bookings$company <- NULL
print(ncol(hotel_bookings))

## [1] 32

"company" %in% colnames(hotel_bookings)

## [1] TRUE
```

From the data variable description we see that the Distribution Channel category that tells us about Booking distribution.

The term “TA” means “Travel Agents” The term “TO” means “Tour Operators” This allows us to fill the missing values in the agents column with TO

```
# We replace the mising values i.e. for TO
# ---
# YOUR GOES BELOW
#

#hotel_bookings$agent
unique1<-unique(hotel_bookings$agent)

#sum(is.na(hotel_bookings$agent))
#colnames(hotel_bookings)
#hotel_bookings$agent[is.na(hotel_bookings$agent)] <- 'TO'

hotel_bookings$agent[hotel_bookings$agent == "NULL"] <- "TO"
#print(hotel_bookings$agent)

# We drop rows where there is no adult, baby and child as
# these records won't help us.
# ---
```

```

# YOUR GOES BELOW
#
print(dim(hotel_bookings))

## [1] 119390      32
hotel_bookings <- hotel_bookings[!(hotel_bookings$adults == 0 & hotel_bookings$babies == 0 & hotel_bookings$children == 0)]

print(dim(hotel_bookings))

## [1] 119210      32
# We replace missing children values with rounded mean value
# ---
# Hint i.e. use round()
# ---
# YOUR GOES BELOW
#

mean_children <- mean(hotel_bookings$children, na.rm = TRUE)
print(mean_children)

## [1] 0.1040468
hotel_bookings$children[is.na(hotel_bookings$children)] <- round(mean_children)
print(hotel_bookings$children[hotel_bookings$children == mean_children])

## numeric(0)
#print(hotel_bookings$children)
print(typeof(hotel_bookings$children))

## [1] "double"
# Checking for missing values in the dataframe
# ---
# YOUR GOES BELOW
#
sum(is.na(hotel_bookings))

## [1] 0
# Converting the datatypes of the following columns from float to integer
# i.e. children, company, agent
# ---
# YOUR GOES BELOW
#

hotel_bookings$children <- as.integer(hotel_bookings$children)
#hotel_bookings$company <- as.integer(hotel_bookings$company) this column has been remove above
hotel_bookings$agent <- as.integer(hotel_bookings$agent)

## Warning: NAs introduced by coercion
str(hotel_bookings)

## tibble [119,210 x 32] (S3: tbl_df/tbl/data.frame)
## $ hotel                                : chr [1:119210] "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel"

```



```
## $ is_canceled           : num [1:119210] 0 0 0 0 0 0 0 0 1 1 ...
## $ lead_time             : num [1:119210] 342 737 7 13 14 14 0 9 85 75 ...
## $ arrival_date_year     : num [1:119210] 2015 2015 2015 2015 2015 ...
## $ arrival_date_month    : chr [1:119210] "July" "July" "July" "July" ...
## $ arrival_date_week_number : num [1:119210] 27 27 27 27 27 27 27 27 27 27 ...
## $ arrival_date_day_of_month : num [1:119210] 1 1 1 1 1 1 1 1 1 1 ...
## $ stays_in_weekend_nights : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ stays_in_week_nights   : num [1:119210] 0 0 1 1 2 2 2 2 3 3 ...
## $ adults                 : num [1:119210] 2 2 1 1 2 2 2 2 2 2 ...
## $ children               : int [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ babies                 : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ meal                   : chr [1:119210] "BB" "BB" "BB" "BB" ...
## $ country                : chr [1:119210] "PRT" "PRT" "GBR" "GBR" ...
## $ market_segment         : chr [1:119210] "Direct" "Direct" "Direct" "Corporate" ...
## $ distribution_channel    : chr [1:119210] "Direct" "Direct" "Direct" "Corporate" ...
## $ is_repeated_guest       : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_cancellations  : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_bookings_not_canceled : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ reserved_room_type      : chr [1:119210] "C" "C" "A" "A" ...
## $ assigned_room_type      : chr [1:119210] "C" "C" "C" "A" ...
## $ booking_changes         : num [1:119210] 3 4 0 0 0 0 0 0 0 0 ...
## $ deposit_type            : chr [1:119210] "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
## $ agent                   : int [1:119210] NA NA NA 304 240 240 NA 303 240 15 ...
## $ company                 : chr [1:119210] "NULL" "NULL" "NULL" "NULL" ...
## $ days_in_waiting_list    : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ customer_type           : chr [1:119210] "Transient" "Transient" "Transient" "Transient" ...
## $ adr                     : num [1:119210] 0 0 75 75 98 ...
## $ required_car_parking_spaces : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ total_of_special_requests : num [1:119210] 0 0 0 0 1 1 0 1 1 0 ...
## $ reservation_status      : chr [1:119210] "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
## $ reservation_status_date : Date[1:119210], format: "2015-07-01" "2015-07-01" ...
```

```
library(dplyr)
library(ggplot2)

# Summarize the count of canceled and non-canceled bookings
cancelled_count <- hotel_bookings %>%
  group_by(is_canceled) %>%
  summarise(count = n())

# Create a barplot to visualize the count of canceled and non-canceled bookings
p<-ggplot(cancelled_count, aes(x = factor(is_canceled), y = count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Bookings Canceled vs Not Canceled",
       x = "Booking Status (0 = Not Canceled, 1 = Canceled)",
       y = "Count of Bookings") +
  theme_minimal()

# Print the plot
print(p)
```



```
# Save the plot as an image (PNG format)
#ggsave("bookings_canceled_plot.png", plot = p, width = 6, height = 4, dpi = 300)
```

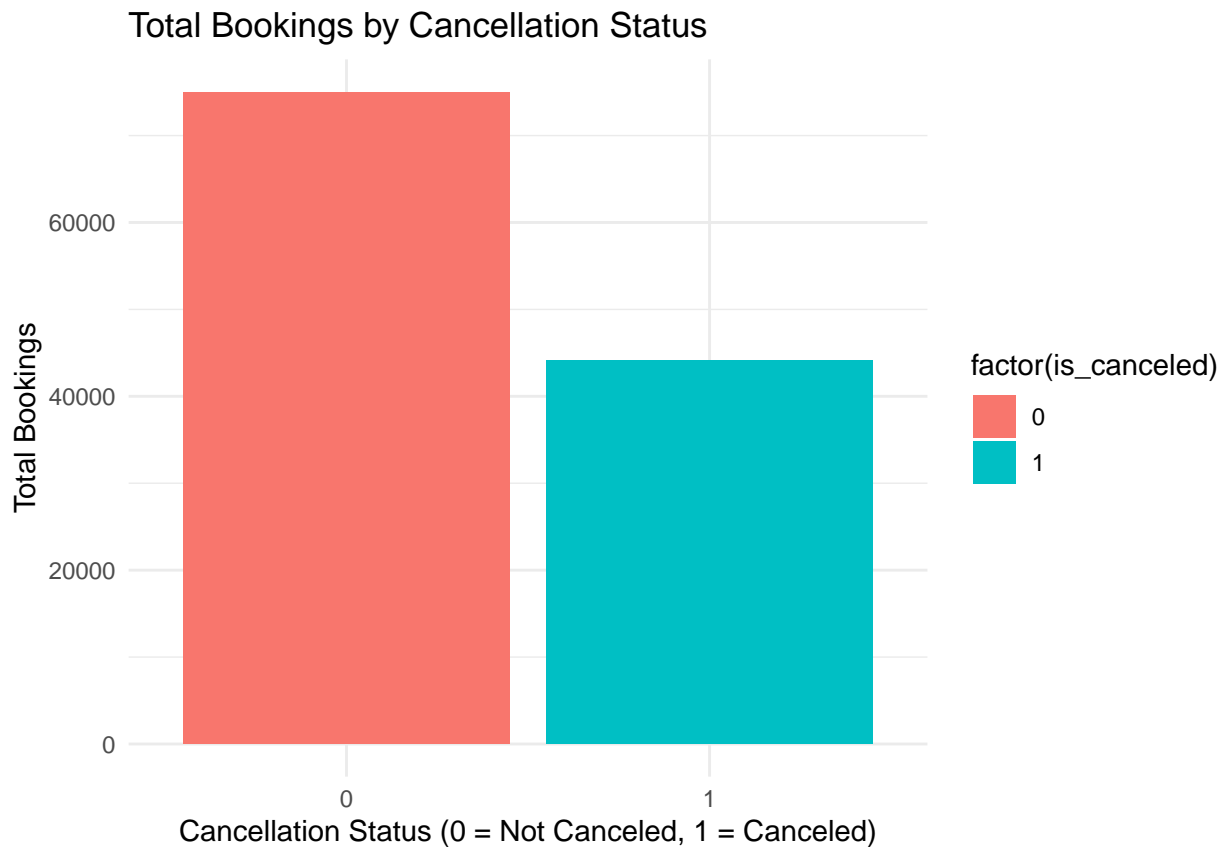
5. Solution Implementation

5.a) Questions

```
# 1. How many bookings were cancelled?
# ---
# Visualisation: Barplot
library(ggplot2)
library(dplyr)

# Summarize booking data by hotel and cancellation status
booking_counts <- hotel_bookings %>%
  group_by(hotel, is_canceled) %>%
  summarise(count = n(), .groups = "drop")

# Plot the data
p<-ggplot(booking_counts, aes(x = factor(is_canceled), y = count, fill = factor(is_canceled))) +
  geom_bar(stat = "identity") +
  labs(title = "Total Bookings by Cancellation Status",
       x = "Cancellation Status (0 = Not Canceled, 1 = Canceled)",
       y = "Total Bookings") +
  theme_minimal()
# Print the plot
print(p)
```



```
# Save the plot as an image (PNG format)
#ggsave("bookings_canceled_plot1.png", plot = p, width = 6, height = 4, dpi = 300)
```

```
#str(hotel_bookings)
#print(unique(hotel_bookings$is_canceled))
```

```
# 2. What was the booking ratio between resort hotel and city hotel?
# ---
# Barplot of booking ratio between resort hotel and city hotel
```

```
colnames(hotel_bookings)
```

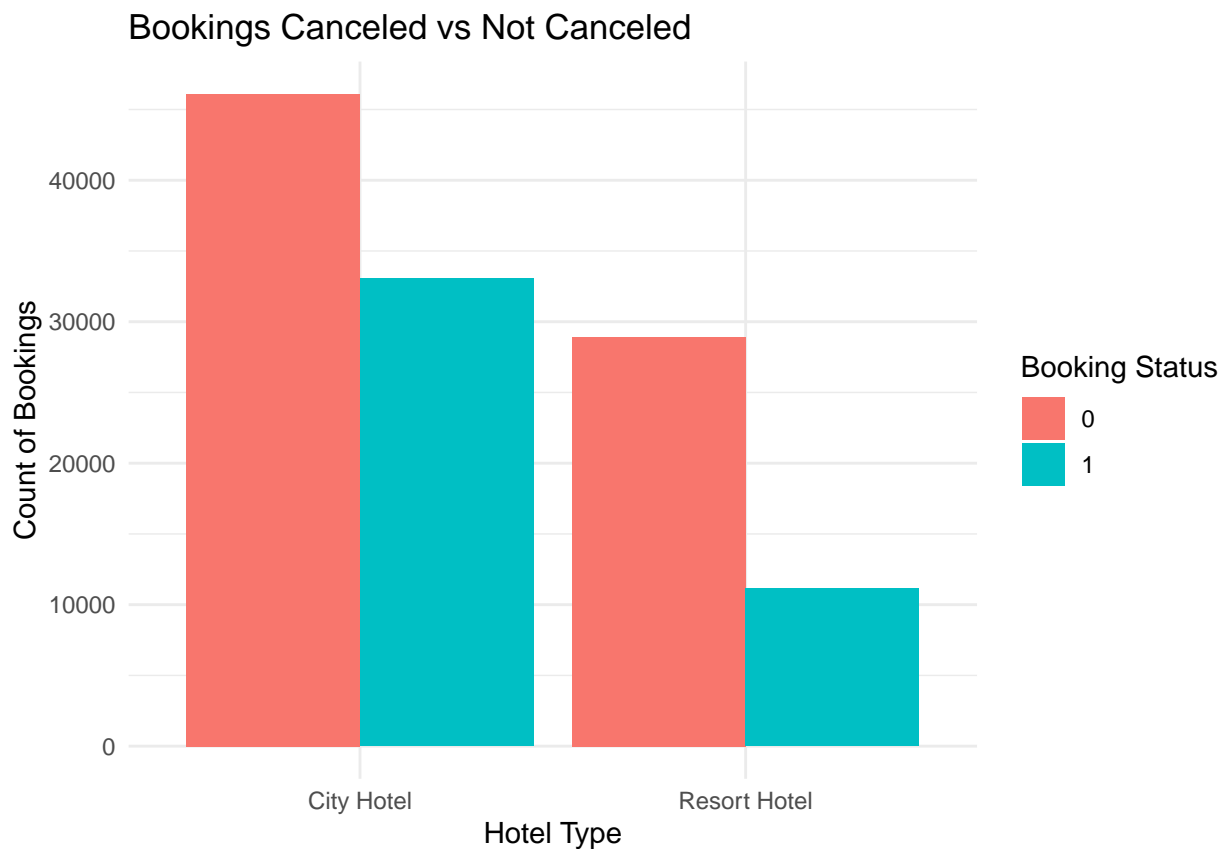
```
## [1] "hotel" "is_canceled"
## [3] "lead_time" "arrival_date_year"
## [5] "arrival_date_month" "arrival_date_week_number"
## [7] "arrival_date_day_of_month" "stays_in_weekend_nights"
## [9] "stays_in_week_nights" "adults"
## [11] "children" "babies"
## [13] "meal" "country"
## [15] "market_segment" "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"

booking_counts <- hotel_bookings %>%
  group_by(hotel, is_canceled) %>%
  summarise(count = n(), .groups = "drop")
print(booking_counts)

## # A tibble: 4 x 3
##   hotel          is_canceled count
##   <chr>          <dbl> <int>
## 1 City Hotel      0 46084
## 2 City Hotel      1 33079
## 3 Resort Hotel    0 28927
## 4 Resort Hotel    1 11120

p<-ggplot(booking_counts, aes(x = factor(hotel), y = count, fill = factor(is_canceled))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Bookings Canceled vs Not Canceled",
       x = "Hotel Type",
       y = "Count of Bookings",
       fill = "Booking Status") +
  theme_minimal()
# Print the plot
print(p)
```



```
# Save the plot as a PNG file
ggsave("bookings_canceled_vs_not_canceled.png", plot = p, width = 6, height = 4, dpi = 300)
```

```

# 3. What was the percentage of booking for each year?
# ---
#
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"

#str(hotel_bookings)
booking_data<-hotel_bookings %>%
  group_by(arrival_date_year) %>%
  summarise(total_bookings = n()) %>%
  mutate(percentage = total_bookings / sum(total_bookings) * 100)

print(booking_data)

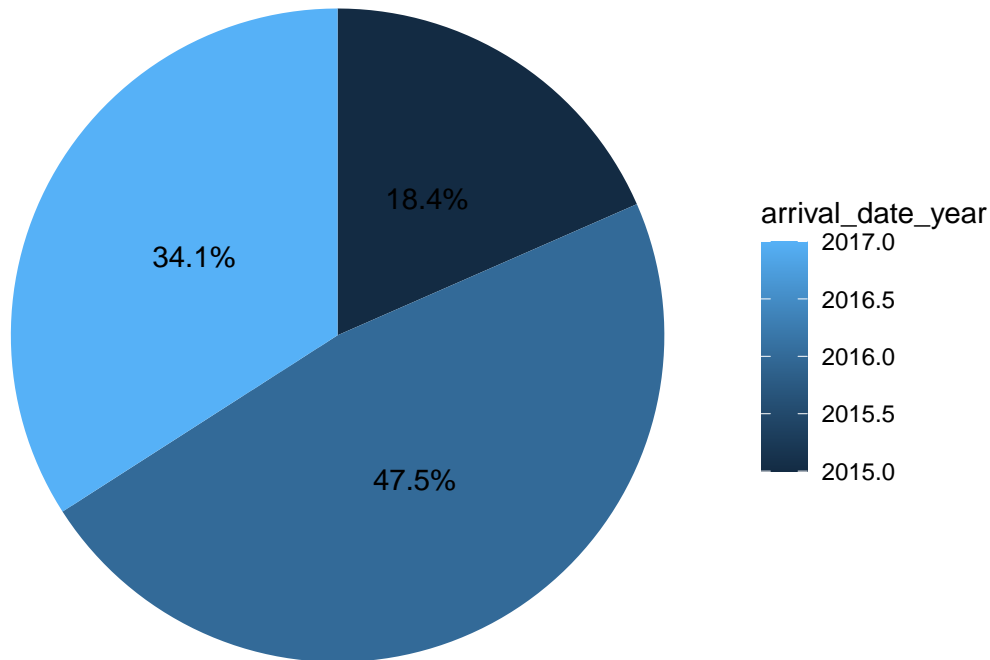
## # A tibble: 3 x 3
##   arrival_date_year total_bookings percentage
##           <dbl>           <int>         <dbl>
## 1             2015             21967          18.4
## 2             2016             56623          47.5
## 3             2017             40620          34.1

p<-ggplot(booking_data, aes(x = "", y = percentage, fill = arrival_date_year)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Booking Distribution by Arrival Year") +
  theme_void() +
  geom_text(aes(label = paste0(round(percentage, 1), "%")), position = position_stack(vjust = 0.5))

# Print the plot
print(p)

```

Booking Distribution by Arrival Year



```
# Save the plot as a PNG file
ggsave("percentage_bookings_per_year.png", plot = p, width = 6, height = 4, dpi = 300)
```

```
"total-bookings" %in% colnames(hotel_bookings)
```

```
## [1] FALSE
```

```
#print(hotel_bookings$arrival_date_year)
print(unique(hotel_bookings$arrival_date_year))
```

```
## [1] 2015 2016 2017
```

```
# 4. Which were the most busiest months for hotels?
```

```
# ---
```

```
#
```

```
str(hotel_bookings)
```

```
## tibble [119,210 x 32] (S3: tbl_df/tbl/data.frame)
```

```
## $ hotel                : chr [1:119210] "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
## $ is_canceled          : num [1:119210] 0 0 0 0 0 0 0 0 1 1 ...
## $ lead_time            : num [1:119210] 342 737 7 13 14 14 0 9 85 75 ...
## $ arrival_date_year    : num [1:119210] 2015 2015 2015 2015 2015 ...
## $ arrival_date_month   : chr [1:119210] "July" "July" "July" "July" ...
## $ arrival_date_week_number : num [1:119210] 27 27 27 27 27 27 27 27 27 27 ...
## $ arrival_date_day_of_month : num [1:119210] 1 1 1 1 1 1 1 1 1 1 ...
## $ stays_in_weekend_nights : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ stays_in_week_nights   : num [1:119210] 0 0 1 1 2 2 2 2 3 3 ...
## $ adults               : num [1:119210] 2 2 1 1 2 2 2 2 2 2 ...
## $ children             : int [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ babies               : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ meal                 : chr [1:119210] "BB" "BB" "BB" "BB" ...
```

```
## $ country : chr [1:119210] "PRT" "PRT" "GBR" "GBR" ...
## $ market_segment : chr [1:119210] "Direct" "Direct" "Direct" "Corporate" ...
## $ distribution_channel : chr [1:119210] "Direct" "Direct" "Direct" "Corporate" ...
## $ is_repeated_guest : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_cancellations : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ previous_bookings_not_canceled: num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ reserved_room_type : chr [1:119210] "C" "C" "A" "A" ...
## $ assigned_room_type : chr [1:119210] "C" "C" "C" "A" ...
## $ booking_changes : num [1:119210] 3 4 0 0 0 0 0 0 0 0 ...
## $ deposit_type : chr [1:119210] "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
## $ agent : int [1:119210] NA NA NA 304 240 240 NA 303 240 15 ...
## $ company : chr [1:119210] "NULL" "NULL" "NULL" "NULL" ...
## $ days_in_waiting_list : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ customer_type : chr [1:119210] "Transient" "Transient" "Transient" "Transient" ...
## $ adr : num [1:119210] 0 0 75 75 98 ...
## $ required_car_parking_spaces : num [1:119210] 0 0 0 0 0 0 0 0 0 0 ...
## $ total_of_special_requests : num [1:119210] 0 0 0 0 1 1 0 1 1 0 ...
## $ reservation_status : chr [1:119210] "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
## $ reservation_status_date : Date[1:119210], format: "2015-07-01" "2015-07-01" ...
```

```
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"
```

```
#str(hotel_bookings)
```

```
hotel_bookings %>%
  group_by(arrival_date_month) %>%
  summarise(total_bookings = n()) %>%
  arrange(desc(total_bookings)) # Sort by total bookings in descending order
```

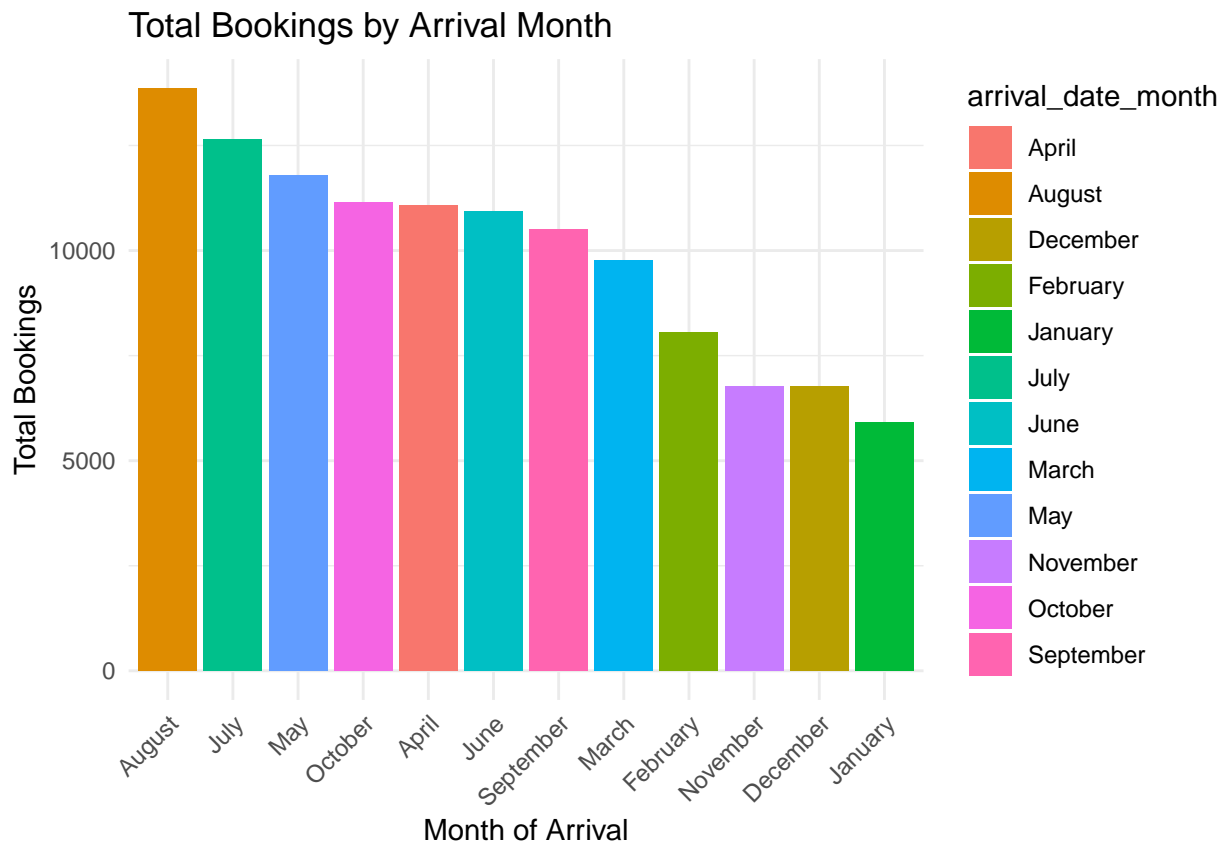
```
## # A tibble: 12 x 2
##   arrival_date_month total_bookings
##   <chr> <int>
## 1 August 13861
## 2 July 12644
## 3 May 11780
## 4 October 11147
## 5 April 11078
## 6 June 10929
```

```
## 7 September      10500
## 8 March          9768
## 9 February       8052
## 10 November      6771
## 11 December      6759
## 12 January       5921
```

```
monthly_bookings <- hotel_bookings %>%
  group_by(arrival_date_month) %>%
  summarise(total_bookings = n()) %>%
  arrange(desc(total_bookings)) # Sort by total bookings in descending order

# Plot the bar chart
p<-ggplot(monthly_bookings, aes(x = reorder(arrival_date_month, -total_bookings), y = total_bookings, fill = arrival_date_month)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Bookings by Arrival Month",
       x = "Month of Arrival",
       y = "Total Bookings") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability

# Print the plot
print(p)
```



```
# Save the plot as an image
ggsave("busiest_months_hotel_bookings.png", plot = p, width = 6, height = 4, dpi = 300)
```



```

# 5. From which top 3 countries did most guests come from?
# ---
# YOUR GOES BELOW
#
result <- hotel_bookings %>%
  group_by(country) %>%
  summarise(total_people = sum(adults + children + babies, na.rm = TRUE)) %>%
  arrange(desc(total_people))

# Print the result
print(result)

```

```

## # A tibble: 178 x 2
##   country total_people
##   <chr>      <dbl>
## 1 PRT          90036
## 2 GBR          24568
## 3 FRA          21579
## 4 ESP          18153
## 5 DEU          14198
## 6 ITA           7856
## 7 IRL           6909
## 8 BEL           4911
## 9 BRA           4867
## 10 USA          4318
## # i 168 more rows

```

```

'total_stay_nights' %in% colnames(hotel_bookings)

```

```

## [1] FALSE

```

```

# 6.a) How long do most people stay in hotels?}
# b) By city and resort? Separate the data by hotel
# ---
#

```

```

# Create a new column for total stay duration (nights)
hotel_bookings <- hotel_bookings %>%
  mutate(total_stay_nights = stays_in_week_nights + stays_in_weekend_nights)

# Summarize the most common stay durations
stay_distribution <- hotel_bookings %>%
  group_by(total_stay_nights) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

# Print the most common stay duration
print(stay_distribution)

```

```

## # A tibble: 42 x 2
##   total_stay_nights count
##   <dbl> <int>
## 1           2 27632
## 2           3 27064

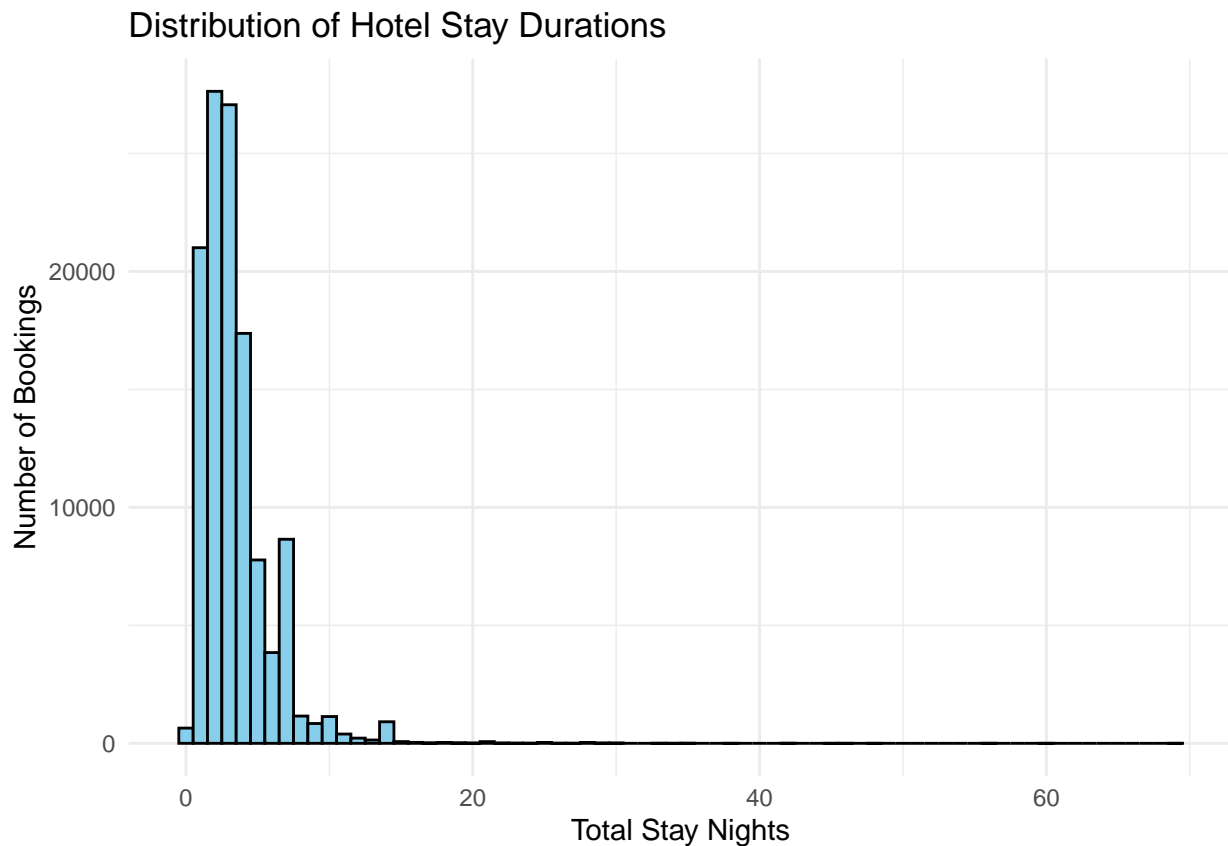
```

```
## 3          1 21005
## 4          4 17373
## 5          7 8648
## 6          5 7771
## 7          6 3846
## 8          8 1155
## 9         10 1135
## 10         14 913
## # i 32 more rows
```

```
# Plot a histogram of stay durations
p<-ggplot(hotel_bookings, aes(x = total_stay_nights)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Hotel Stay Durations",
       x = "Total Stay Nights",
       y = "Number of Bookings") +
  theme_minimal()

# Save the plot to a file
ggsave("hotel_stay_distribution_plot.png", plot = p, width = 8, height = 6)

# Optionally, print the plot to the R console as well
print(p)
```



```
###Question b
hotel_bookings %>%
  mutate(total_stay_nights = stays_in_week_nights + stays_in_weekend_nights) %>%
```

```

group_by(country, hotel) %>%
  summarise(avg_stay_nights = mean(total_stay_nights, na.rm = TRUE)) %>%
  arrange(desc(avg_stay_nights))

## `summarise()` has grouped output by 'country'. You can override using the
## `.groups` argument.

## # A tibble: 293 x 3
## # Groups:   country [178]
##   country hotel      avg_stay_nights
##   <chr>   <chr>          <dbl>
## 1 FRO     City Hotel         12
## 2 SEN     Resort Hotel        10
## 3 AZE     Resort Hotel         9
## 4 TGO     Resort Hotel         9
## 5 SEN     City Hotel          8.7
## 6 AGO     City Hotel          8.48
## 7 MKD     Resort Hotel         8
## 8 CPV     City Hotel          7.16
## 9 GNB     City Hotel          7.11
## 10 ARM    Resort Hotel         7
## # i 283 more rows

# 7. Which was the most booked accommodation type (Single, Couple, Family)?
# ---
#

#colnames(hotel_bookings)
#str(hotel_bookings)
accommodation_data<-hotel_bookings %>%
mutate(accommodation_type = case_when(
  adults == 1 & children == 0 & babies == 0 ~ "Single",
  adults == 2 & children == 0 & babies == 0 ~ "Couple",
  adults >= 2 | children > 0 | babies > 0 ~ "Family",
  TRUE ~ "Other"
)) %>%
group_by(accommodation_type) %>%
summarise(total_bookings = n()) %>%
arrange(desc(total_bookings))%>%
mutate(Percentage_total_bookings=(total_bookings/sum(total_bookings))*100)

print(accommodation_data)

## # A tibble: 3 x 3
##   accommodation_type total_bookings Percentage_total_bookings
##   <chr>                <int>          <dbl>
## 1 Couple                81560            68.4
## 2 Single                22577            18.9
## 3 Family                15073            12.6

# Create the pie chart with percentages
p<-ggplot(accommodation_data, aes(x = "", y = Percentage_total_bookings, fill = accommodation_type)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") + # Convert to pie chart
  geom_text(aes(label = paste0(round(Percentage_total_bookings, 1), "%")),

```

```

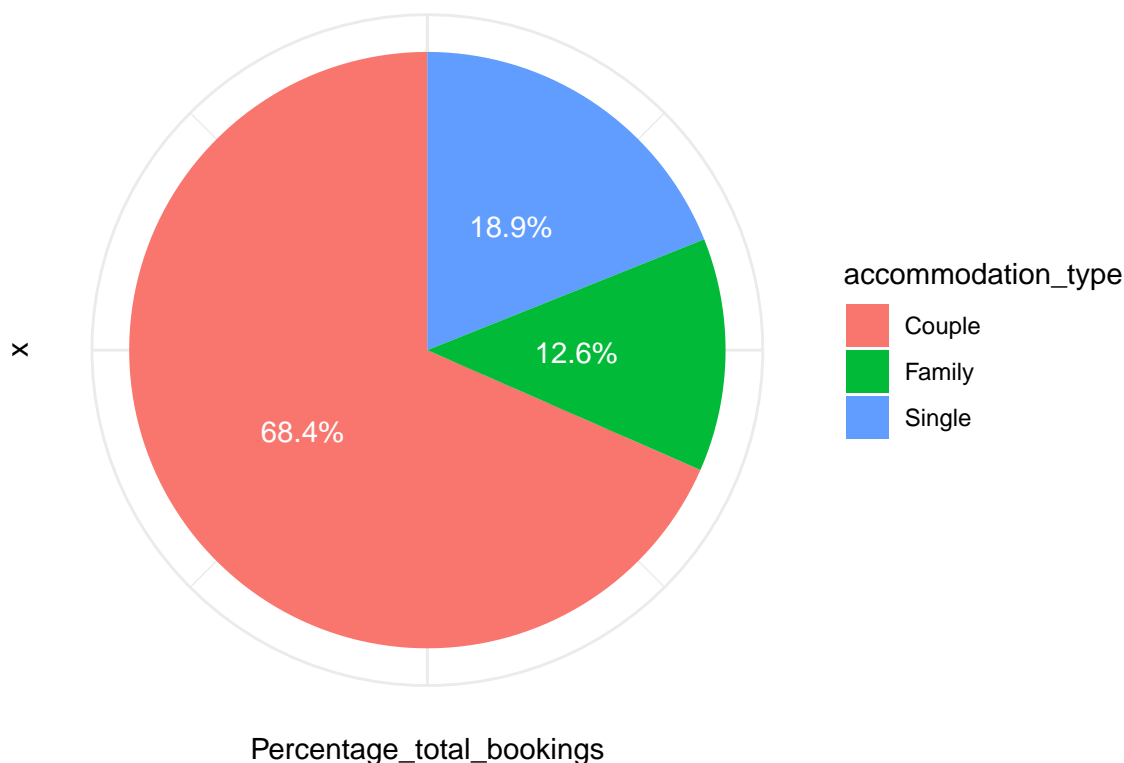
    position = position_stack(vjust = 0.5),
    color = "white") + # Add percentage labels
labs(title = "Distribution of Accommodation Types") +
theme_minimal() +
theme(axis.text.x = element_blank()) # Remove x-axis text for cleaner look

#ggsave("accommodation_type_pie_chart.png",
      #plot = p,
      #width = 8,
      #height = 6,
      #dpi = 300)

# Optionally, print the plot to the R console as well
print(p)

```

Distribution of Accommodation Types



5.b) Recommendations

From the above analysis, below are our recommendations:

Accommodation type and bookings: Couple bookings make up the majority, accounting for 68.42% of total bookings. Single bookings represent 18.94%, and family bookings account for 12.64%. This suggests that most bookings are for couples.

Hotel type and average stay: Resort hotels tend to have a slightly lower average stay (around 9 nights) compared to city hotels, which have a higher average stay (around 12 nights). Several countries (e.g., FRO, SEN, AZE) primarily have resort hotels with an average stay of 9 nights, while others (e.g., CPV, GNB) lean toward city hotels with average stays of 7 nights or lower.

Booking frequency by country: Portugal (PRT) has the highest number of bookings (90,036), followed by Great Britain (GBR) (24,568), and France (FRA) (21,579). USA and Brazil have lower booking counts (4,318 and 4,867, respectively), suggesting fewer bookings from these countries.

Bookings by month: August has the highest number of bookings (13,861), followed by July (12,644) and May (11,780). The months of March and November see significantly fewer bookings, with 9,768 and 6,771, respectively.

Cancellations: The data shows cancellations are more frequent for city hotels (with 33,079 cancellations compared to 46,084 non-cancellations) compared to resort hotels. This indicates that city hotels might experience a higher cancellation rate.

Bookings over the years: The number of bookings was highest in 2016 (56,623 bookings), followed by 2017 (40,620 bookings). 2015 had the lowest number of bookings (21,967), which could indicate an increase in bookings over time.

These trends suggest that couples prefer city hotels for longer stays, while families and singles lean more towards resort hotels. There are also clear seasonal peaks, with summer months (May-August) being particularly busy for bookings.

Based on the analysis of **cancellation rates** and **booking trends** by month, here are some recommendations for hotels:

Target marketing towards couples: Since couple bookings make up the majority of total bookings (68.42%), marketing campaigns could focus more on this group, offering tailored packages or promotions for couples. This could include offering longer stays at city hotels, which have higher average stays.

Focus on resort hotels for family and single travelers: Given that families and singles make up a smaller percentage of bookings, offering special family packages or single traveler discounts could help boost bookings in resort hotels, which tend to have slightly shorter stays. Emphasizing the leisure and relaxation aspects of resort hotels may attract these groups.

Seasonal promotions for peak months: With the highest number of bookings occurring during the summer months (May-August), it would be beneficial to launch special seasonal promotions or early-bird offers during these months to capitalize on the high demand.

Improve cancellation policies for city hotels: Since city hotels experience a higher rate of cancellations, offering more flexible cancellation policies or incentives for non-cancelled bookings (e.g., discounts on future stays) could help reduce cancellations and increase bookings.

Target specific countries with lower bookings: Countries like the USA and Brazil have fewer bookings. Special promotions or campaigns targeted toward these regions could help increase the number of bookings from these countries. This might include region-specific offers, localized marketing, or partnerships with travel agencies in those countries.

Encourage longer stays in resort hotels: Since resort hotels have slightly lower average stays, offering incentives for guests to stay longer, such as discounted rates for extended stays, could help increase the average stay duration and revenue for these properties.

6. Challenging your Solution

In this step, we review our solution and implement approaches that could potentially provide a better outcome. In our case, we could propose the following question that wasn't answered in our solution because it couldn't have greatly contributed to our recommendation.

```
# When should hotels provide special offers?
# ---
# YOUR GOES BELOW
#
```

```
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"
## [33] "total_stay_nights"
```

```
special_offer <- hotel_bookings %>%
  #mutate(total_stay_nights = stays_in_week_nights + stays_in_weekend_nights) %>%
  group_by(arrival_date_month) %>%
  summarise(
    total_bookings = n(),
    total_canceled = sum(is_canceled == 1),
    total_not_canceled = sum(is_canceled == 0),
    #total_stay_nights = sum(total_stay_nights, na.rm = TRUE)
  ) %>%
  arrange(desc(total_bookings))

print(special_offer)
```

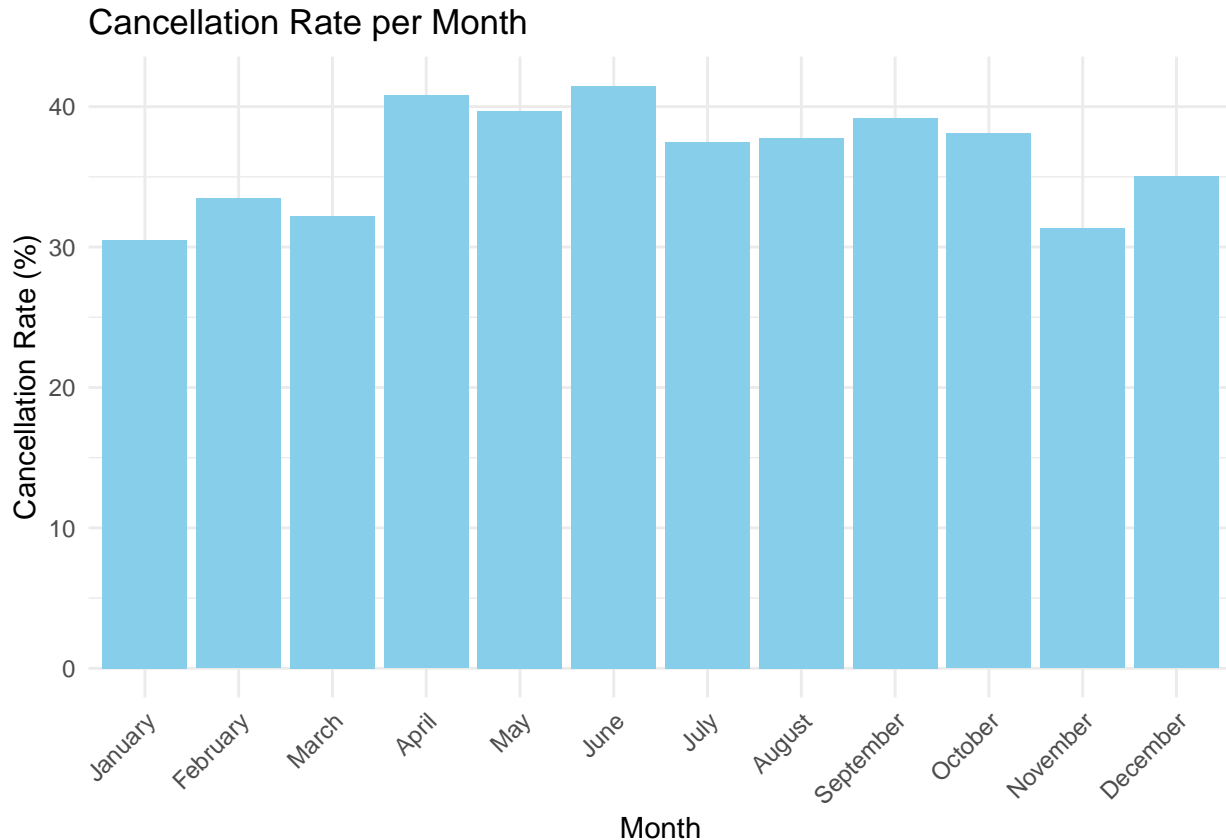
```
## # A tibble: 12 x 4
##   arrival_date_month total_bookings total_canceled total_not_canceled
##   <chr>                <int>          <int>          <int>
## 1 August                13861            5237            8624
## 2 July                  12644            4737            7907
## 3 May                   11780            4677            7103
## 4 October               11147            4246            6901
## 5 April                 11078            4518            6560
## 6 June                  10929            4534            6395
## 7 September             10500            4115            6385
## 8 March                  9768            3148            6620
## 9 February              8052            2693            5359
## 10 November              6771            2120            4651
## 11 December              6759            2368            4391
## 12 January               5921            1806            4115
```

```
# Calculate cancellation rate per month
offer <- special_offer %>%
  mutate(cancellation_rate = total_canceled / total_bookings * 100) %>%
  mutate(arrival_date_month = factor(arrival_date_month, levels = month.name)) # order months
```

```
print(offer)
```

```
## # A tibble: 12 x 5
##   arrival_date_month total_bookings total_canceled total_not_canceled
##   <fct>                <int>          <int>          <int>
## 1 August                13861            5237            8624
## 2 July                  12644            4737            7907
## 3 May                   11780            4677            7103
## 4 October               11147            4246            6901
## 5 April                 11078            4518            6560
## 6 June                  10929            4534            6395
## 7 September             10500            4115            6385
## 8 March                  9768            3148            6620
## 9 February              8052            2693            5359
## 10 November              6771            2120            4651
## 11 December              6759            2368            4391
## 12 January               5921            1806            4115
## # i 1 more variable: cancellation_rate <dbl>
```

```
# Plot the cancellation rate against the month
ggplot(offer, aes(x = arrival_date_month, y = cancellation_rate)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Cancellation Rate per Month",
       x = "Month",
       y = "Cancellation Rate (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for clarity
```



Our observations:

How does this observation tie to our solution?

Higher cancellation rates may signal suitable periods for special offers like in April and June.

Months with fewer bookings may indicate lower demand, and special offers can be used to attract more customers. For instance, as November and February have fewer bookings, hotels can offer discounts or promotions to boost demand.

7. Follow up questions

During this step, you rethink and propose other ways that you can improve your solution.

a). Did we have the right data? b). Do we need other data to answer our question? c). Did we have the right question?

```
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"
## [33] "total_stay_nights"
```

The provided data seems to cover essential aspects such as accommodation type, booking, duration in the hotels country etc. However, there could be gaps in data that might provide more insights, such as customer satisfaction metrics (reviews or feedback). These additional data points could help refine the recommendations, especially for targeting specific customer groups or improving the guest experience.

Do we need other data to answer our question?

Yes, additional data would enhance the solution. For instance:

Booking source or channel: Knowing where bookings originate from (e.g., direct website, third-party platforms) could help refine marketing strategies. **Customer feedback and satisfaction:** Data on guest satisfaction (ratings, reviews) would allow for better understanding of preferences and areas for improvement.

Booking trends over longer periods: Data from multiple years or months could help identify long-term trends rather than just focusing on short-term fluctuations.

Did we have the right question?

Generally, yes, but the data is insufficient to provide clear guidelines on how to optimize strategies for customer retention and loyalty. We need a deeper understanding of customer behavior and preferences to effectively develop strategies that foster customer loyalty and encourage repeat bookings. For example:

Why are bookings higher during certain months or in certain countries? Understanding the factors behind seasonality and country-specific preferences could lead to more targeted interventions but this requires also more data as we mentioned above mainly about the customers satisfaction and reviews what not even their salary, the payment method an so on.

What factors influence cancellations, and how can we mitigate them? A deeper exploration of what drives cancellations (e.g., price sensitivity, competition, customer dissatisfaction) could provide clearer strategies for reducing them.

How can we improve the guest experience to encourage repeat bookings? Focusing on customer retention (activities during stay...) and loyalty strategies could add a dimension to the recommendations