Hotel Booking Analysis

Anna Manina

2024-07-01

Introduction

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

The data was downloaded and cleaned by Thomas Mock and Antoine Bichat for #TidyTuesday during the week of February 11th, 2020 here.

To learn more about the dataset click here.

Scenario

In this scenario, I am a junior data analyst working for a hotel booking company. I have been asked to clean a .csv file that was created after querying a database to combine two different tables from different hotels. In order to learn more about this data, I am going to need to use functions to preview the data's structure, including its columns and rows. I will also need to use basic cleaning functions to prepare this data for analysis, and then create visualizations that highlight different aspects of the data to present to my stakeholder.

Step 1: Load packages

```r
install.packages("tidyverse")
install.packages("skimr")
install.packages("janitor")
install.packages("rmarkdown")
install.packages("readr")
```

```r
library(tidyverse)
library(skimr)
library(janitor)
library(rmarkdown)
library(readr)
```

Step 2: Import Data

```r
bookings_df <- readr::read_csv("C:/Users/denni/Documents/hotel_bookings.csv.csv")
```

Step 3: Get to Know Data

```
## 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.
```

```r
head(bookings_df)
```

```
## # A tibble: 6 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
## 6 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>, ...
```

```r
str(bookings_df)
```

```
## 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>
```

```r
dplyr::glimpse(bookings_df)
```

```
## Rows: 119,390
## Columns: 32
## $ hotel                          <chr> "Resort Hotel", "Resort Hotel", "Resort~
## $ is_canceled                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, ~
## $ lead_time                      <dbl> 342, 737, 7, 13, 14, 14, 0, 9, 85, 75, ~
## $ arrival_date_year              <dbl> 2015, 2015, 2015, 2015, 2015, 2015, 201~
## $ arrival_date_month             <chr> "July", "July", "July", "July", "July",~
## $ arrival_date_week_number       <dbl> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27,~
## $ arrival_date_day_of_month      <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ stays_in_weekend_nights        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ stays_in_week_nights           <dbl> 0, 0, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, ~
## $ adults                         <dbl> 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ children                       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ babies                         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ meal                           <chr> "BB", "BB", "BB", "BB", "BB", "BB", "BB~
## $ country                        <chr> "PRT", "PRT", "GBR", "GBR", "GBR", "GBR~
## $ market_segment                 <chr> "Direct", "Direct", "Direct", "Corporat~
## $ distribution_channel           <chr> "Direct", "Direct", "Direct", "Corporat~
## $ is_repeated_guest              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ previous_cancellations         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ previous_bookings_not_canceled <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ reserved_room_type             <chr> "C", "C", "A", "A", "A", "A", "C", "C",~
## $ assigned_room_type             <chr> "C", "C", "C", "A", "A", "A", "C", "C",~
## $ booking_changes                <dbl> 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ deposit_type                   <chr> "No Deposit", "No Deposit", "No Deposit~
## $ agent                          <chr> "NULL", "NULL", "NULL", "304", "240", "~
## $ company                        <chr> "NULL", "NULL", "NULL", "NULL", "NULL",~
## $ days_in_waiting_list           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ customer_type                  <chr> "Transient", "Transient", "Transient", ~
## $ adr                            <dbl> 0.00, 0.00, 75.00, 75.00, 98.00, 98.00,~
## $ required_car_parking_spaces    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ total_of_special_requests      <dbl> 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 3, ~
## $ reservation_status             <chr> "Check-Out", "Check-Out", "Check-Out", ~
## $ reservation_status_date        <date> 2015-07-01, 2015-07-01, 2015-07-02, 20~
```

```r
colnames(bookings_df)
```

```
##  [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"
```

```
skimr::skim_without_charts(bookings_df)
```

Table 1: Data summary

Name	bookings_df
Number of rows	119390
Number of columns	32
Column type frequency:	
character	13
Date	1
numeric	18
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
hotel	0	1	10	12	0	2	0
arrival_date_month	0	1	3	9	0	12	0
meal	0	1	2	9	0	5	0
country	0	1	2	4	0	178	0
market_segment	0	1	6	13	0	8	0
distribution_channel	0	1	3	9	0	5	0
reserved_room_type	0	1	1	1	0	10	0
assigned_room_type	0	1	1	1	0	12	0
deposit_type	0	1	10	10	0	3	0
agent	0	1	1	4	0	334	0
company	0	1	1	4	0	353	0
customer_type	0	1	5	15	0	4	0
reservation_status	0	1	7	9	0	3	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
reservation_status_date	0	1	2014-10-17	2017-09-14	2016-08-07	926

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
is_canceled	0	1	0.37	0.48	0.00	0.00	0.00	1	1
lead_time	0	1	104.01	106.86	0.00	18.00	69.00	160	737
arrival_date_year	0	1	2016.16	0.71	2015.00	2016.00	2016.00	2017	2017
arrival_date_week_number	0	1	27.17	13.61	1.00	16.00	28.00	38	53

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
arrival_date_day_of_month	0	1	15.80	8.78	1.00	8.00	16.00	23	31
stays_in_weekend_nights	0	1	0.93	1.00	0.00	0.00	1.00	2	19
stays_in_week_nights	0	1	2.50	1.91	0.00	1.00	2.00	3	50
adults	0	1	1.86	0.58	0.00	2.00	2.00	2	55
children	4	1	0.10	0.40	0.00	0.00	0.00	0	10
babies	0	1	0.01	0.10	0.00	0.00	0.00	0	10
is_repeated_guest	0	1	0.03	0.18	0.00	0.00	0.00	0	1
previous_cancellations	0	1	0.09	0.84	0.00	0.00	0.00	0	26
previous_bookings_not_canceled	0	1	0.14	1.50	0.00	0.00	0.00	0	72
booking_changes	0	1	0.22	0.65	0.00	0.00	0.00	0	21
days_in_waiting_list	0	1	2.32	17.59	0.00	0.00	0.00	0	391
adr	0	1	101.83	50.54	-6.38	69.29	94.58	126	5400
required_car_parking_spaces	0	1	0.06	0.25	0.00	0.00	0.00	0	8
total_of_special_requests	0	1	0.57	0.79	0.00	0.00	0.00	1	5

```
new_df <- dplyr::select(bookings_df, `adr`, adults)
```

```
dplyr::mutate(new_df, total = `adr` / adults)
```

```
## # A tibble: 119,390 x 3
##      adr adults total
##    <dbl>  <dbl> <dbl>
##  1     0       2   0
##  2     0       2   0
##  3   75        1  75
##  4   75        1  75
##  5   98        2  49
##  6   98        2  49
##  7  107        2  53.5
##  8  103        2  51.5
##  9   82        2  41
## 10  106.       2  52.8
## # i 119,380 more rows
```

```
library(dplyr)
```

Step 4: Clean Data

```
##
## 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(tidyr)
```

Based on my notes, I am primarily interested in the following variables: *hotel*, *is_canceled*, *lead_time*. I created a new data frame with just those columns, calling it ***trimmed_df***:

```
trimmed_df <- bookings_df %>%
  select(hotel, is_canceled, lead_time)
```

I renamed the variable *'hotel'* to be named *'hotel_type'* to be crystal clear on what the data is about:

```
trimmed_df %>%
  select(hotel, is_canceled, lead_time) %>%
  rename(hotel_type = hotel)
```

```
## # A tibble: 119,390 x 3
##    hotel_type   is_canceled lead_time
##    <chr>              <dbl>     <dbl>
##  1 Resort Hotel           0       342
##  2 Resort Hotel           0       737
##  3 Resort Hotel           0         7
##  4 Resort Hotel           0        13
##  5 Resort Hotel           0        14
##  6 Resort Hotel           0        14
##  7 Resort Hotel           0         0
##  8 Resort Hotel           0         9
##  9 Resort Hotel           1        85
## 10 Resort Hotel           1        75
## # i 119,380 more rows
```

The next task was to combine the arrival month and year into one column using the *unite()* function:

```
arrival_df <- bookings_df %>%
  select(arrival_date_year, arrival_date_month) %>%
  unite(arrival_month_year, c("arrival_date_month", "arrival_date_year"), sep = " ")
```

I also needed to create a new column that would sum up all the adults, children, and babies on a reservation for the total number of people. I used the ***mutate()*** function to make changes to my columns:

```
guests_df <- bookings_df %>%
  mutate(guests = adults + children + babies)
```

```
head(guests_df)
```

```
## # A tibble: 6 x 33
##   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
## 6 Resort Hotel           0        14               2015 July
## # i 28 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>, ...
```

For the summary statistics, I calculated the total number of canceled bookings and the average lead time for booking:

```
canceled_bookings_df <- bookings_df %>%
  summarize(number_canceled = sum(is_canceled),
            average_lead_time = mean(lead_time))
```

```
head(canceled_bookings_df)
```

```
## # A tibble: 1 x 2
##   number_canceled average_lead_time
##             <dbl>             <dbl>
## 1           44224              104.
```

Step5: Manipulate Data

The data needs to be arranged by most lead time to least lead time because the stakeholder wants to focus on bookings that were made far in advance:

```
arrange(bookings_df, lead_time)
```

```
## # A tibble: 119,390 x 32
##    hotel        is_canceled lead_time arrival_date_year arrival_date_month
##    <chr>              <dbl>     <dbl>             <dbl> <chr>
##  1 Resort Hotel           0         0              2015 July
##  2 Resort Hotel           0         0              2015 July
##  3 Resort Hotel           0         0              2015 July
##  4 Resort Hotel           0         0              2015 July
##  5 Resort Hotel           0         0              2015 July
##  6 Resort Hotel           0         0              2015 July
##  7 Resort Hotel           0         0              2015 July
##  8 Resort Hotel           0         0              2015 July
##  9 Resort Hotel           0         0              2015 July
## 10 Resort Hotel           0         0              2015 July
## # i 119,380 more rows
## # 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>, ...
```

```r
arrange(bookings_df, desc(lead_time))
```

```
## # A tibble: 119,390 x 32
##    hotel        is_canceled lead_time arrival_date_year arrival_date_month
##    <chr>              <dbl>     <dbl>             <dbl> <chr>
##  1 Resort Hotel           0       737              2015 July
##  2 Resort Hotel           0       709              2016 February
##  3 City Hotel             1       629              2017 March
##  4 City Hotel             1       629              2017 March
##  5 City Hotel             1       629              2017 March
##  6 City Hotel             1       629              2017 March
##  7 City Hotel             1       629              2017 March
##  8 City Hotel             1       629              2017 March
##  9 City Hotel             1       629              2017 March
## 10 City Hotel             1       629              2017 March
## # i 119,380 more rows
## # 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>, ...
```

The highest lead time for a hotel booking in this data set is 737 days.

I created a new data frame named *'bookings_df_v2'* that had those changes saved:

```r
bookings_df_v2 <-
  arrange(bookings_df, desc(lead_time))
```

```r
head(bookings_df_v2)
```

```
## # A tibble: 6 x 32
##   hotel        is_canceled lead_time arrival_date_year arrival_date_month
##   <chr>              <dbl>     <dbl>             <dbl> <chr>
## 1 Resort Hotel           0       737              2015 July
## 2 Resort Hotel           0       709              2016 February
## 3 City Hotel             1       629              2017 March
## 4 City Hotel             1       629              2017 March
## 5 City Hotel             1       629              2017 March
## 6 City Hotel             1       629              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>, ...
```

I can also find out the maximum and minimum lead times without sorting the whole data set using the ***`arrange()`*** function, and using the *max()* and *min()* functions instead:

```
max(bookings_df$lead_time)
```

```
## [1] 737
```

```
min(bookings_df$lead_time)
```

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

Now, I just want to know what the average lead time for booking is because I need to find out how early the stakeholder should run promotions for hotel rooms. I used the ***mean()*** function to answer that question:

```
mean(bookings_df$lead_time)
```

```
## [1] 104.0114
```

```
mean(bookings_df_v2$lead_time)
```

```
## [1] 104.0114
```

The average lead time is 104.0114 days.

Now, I want to know what the average lead time before booking is for just city hotels. My first step is creating a new data set that only contains data about city hotels. I did that using the ***filter()*** function, and named my new data frame *'bookings_df_city'*:

```
bookings_df_city <-
  filter(bookings_df, bookings_df$hotel=="City Hotel")
```

```
head(bookings_df_city)
```

```
## # A tibble: 6 x 32
##   hotel      is_canceled lead_time arrival_date_year arrival_date_month
##   <chr>            <dbl>     <dbl>             <dbl> <chr>
## 1 City Hotel           0         6              2015 July
## 2 City Hotel           1        88              2015 July
## 3 City Hotel           1        65              2015 July
## 4 City Hotel           1        92              2015 July
## 5 City Hotel           1       100              2015 July
## 6 City Hotel           1        79              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>, ...
```

```
mean(bookings_df_city$lead_time)
```

```
## [1] 109.7357
```

Now, I need to know some more information about city hotels, including the maximum and minimum lead time. I am also interested in how they are different from resort hotels. I used the *group_by()*, *summarize()* functions, and the *pipe operator* to make my code easier to follow. I stored the new data set in a data frame named *'hotel_summary'*:

```
hotel_summary <-
  bookings_df %>%
  group_by(hotel) %>%
  summarise(average_lead_time=mean(lead_time),
            min_lead_time=min(lead_time),
            max_lead_time=max(lead_time))
```

```
head(hotel_summary)
```

```
## # A tibble: 2 x 4
##   hotel        average_lead_time min_lead_time max_lead_time
##   <chr>                    <dbl>         <dbl>         <dbl>
## 1 City Hotel               110.              0           629
## 2 Resort Hotel              92.7             0           737
```

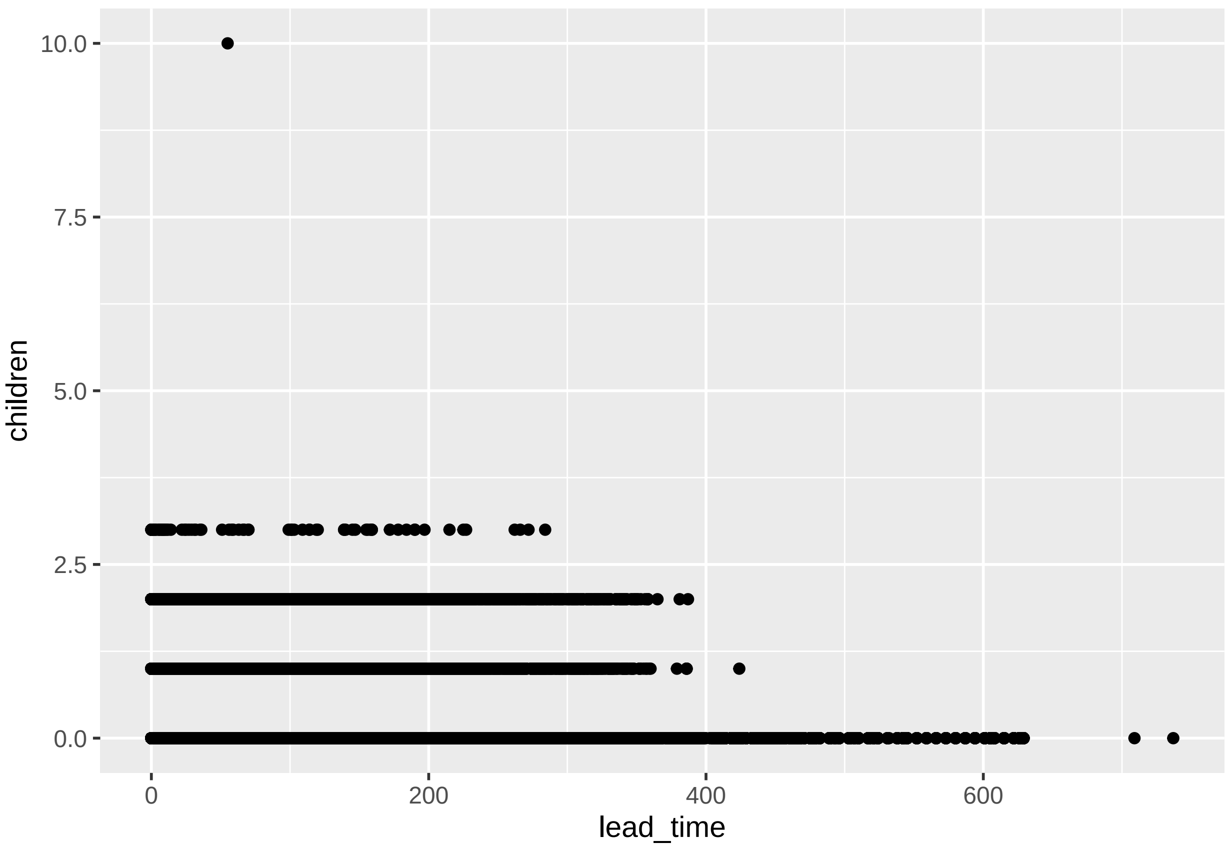
```
library(ggplot2)
```

Step 6:AestheticsandVisualization with ggplot2

I used *ggplot2* to determine if people with children book hotel rooms in advance. On the x-axis, the plot shows how far in advance a booking is made, with the bookings furthest to the right happening the most in advance. On the y-axis its hows how many children there are in a party:

```
ggplot(data = bookings_df) +
  geom_point(mapping = aes(x = lead_time, y = children))
```

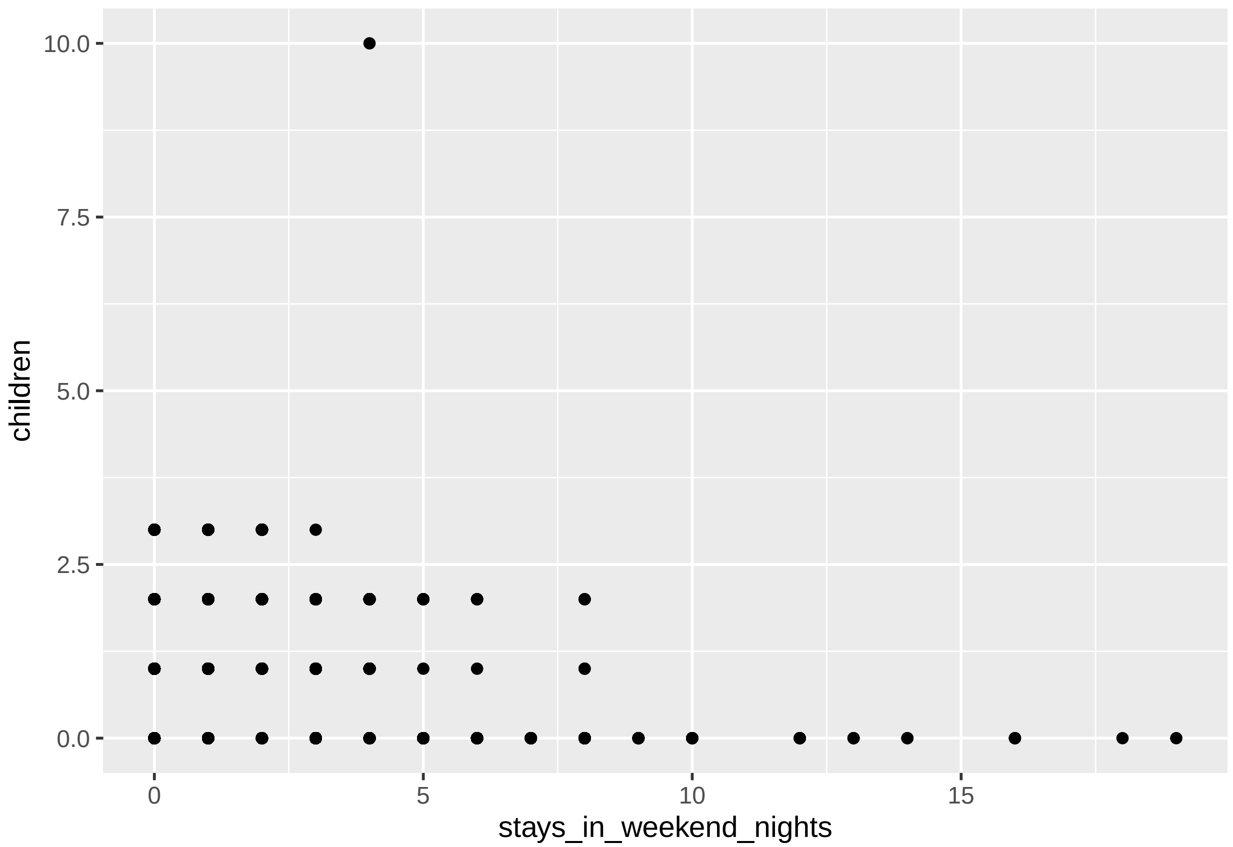
```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



In order to increase weekend bookings, the stakeholder needs to know what group of guests book the most weekend nights in order to target that group in a new marketing campaign:

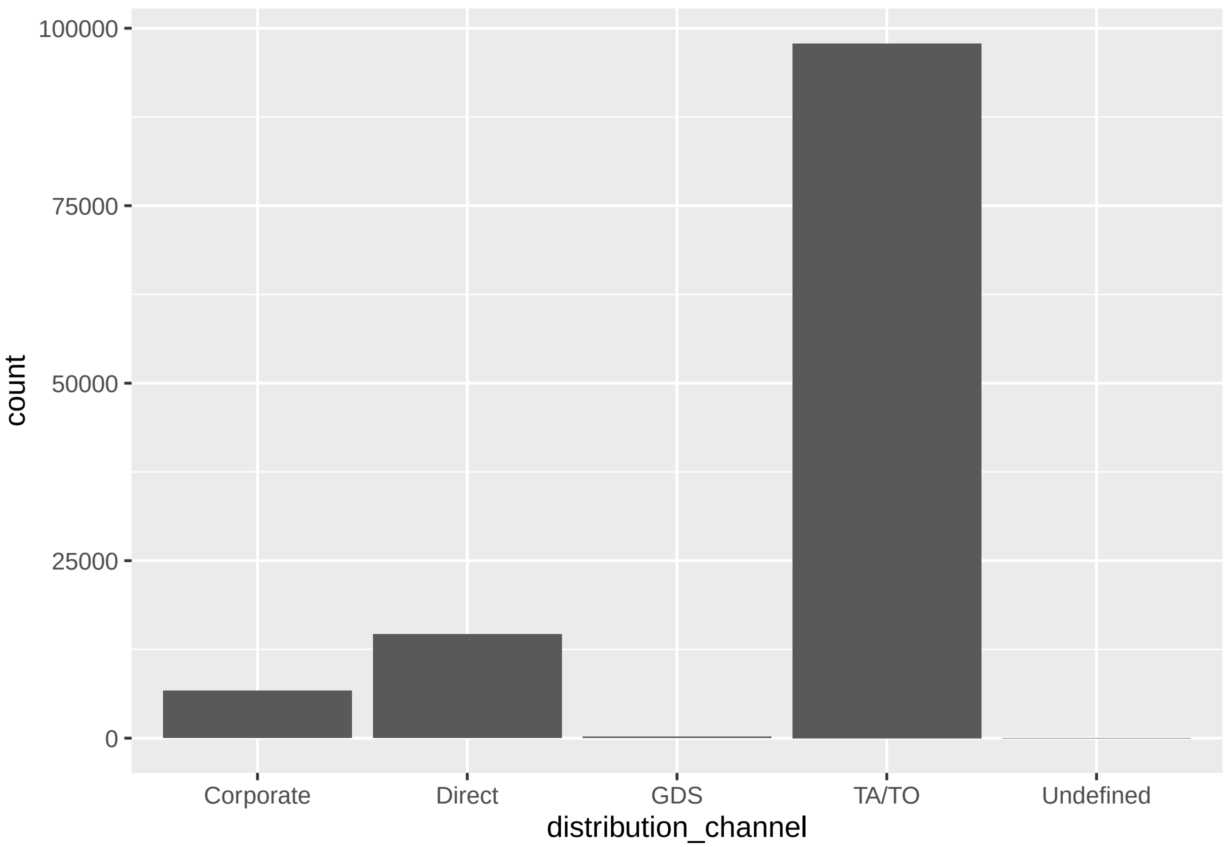
```r
ggplot(data = bookings_df) +
  geom_point(mapping = aes(x = stays_in_weekend_nights, y = children))
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



The stakeholder is also interested in developing promotions based on different booking distributions, but first they need to know how many of the transactions are occurring for each different distribution type.

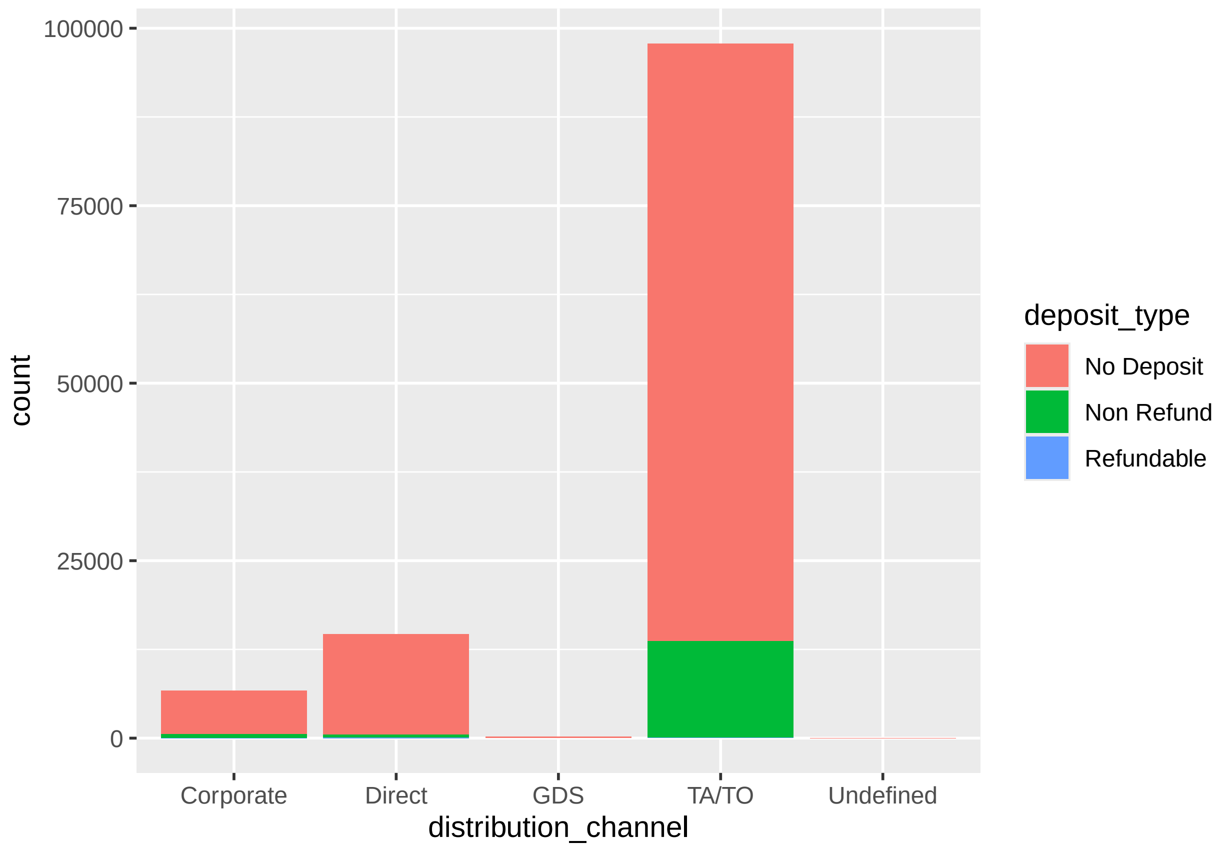
```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel))
```



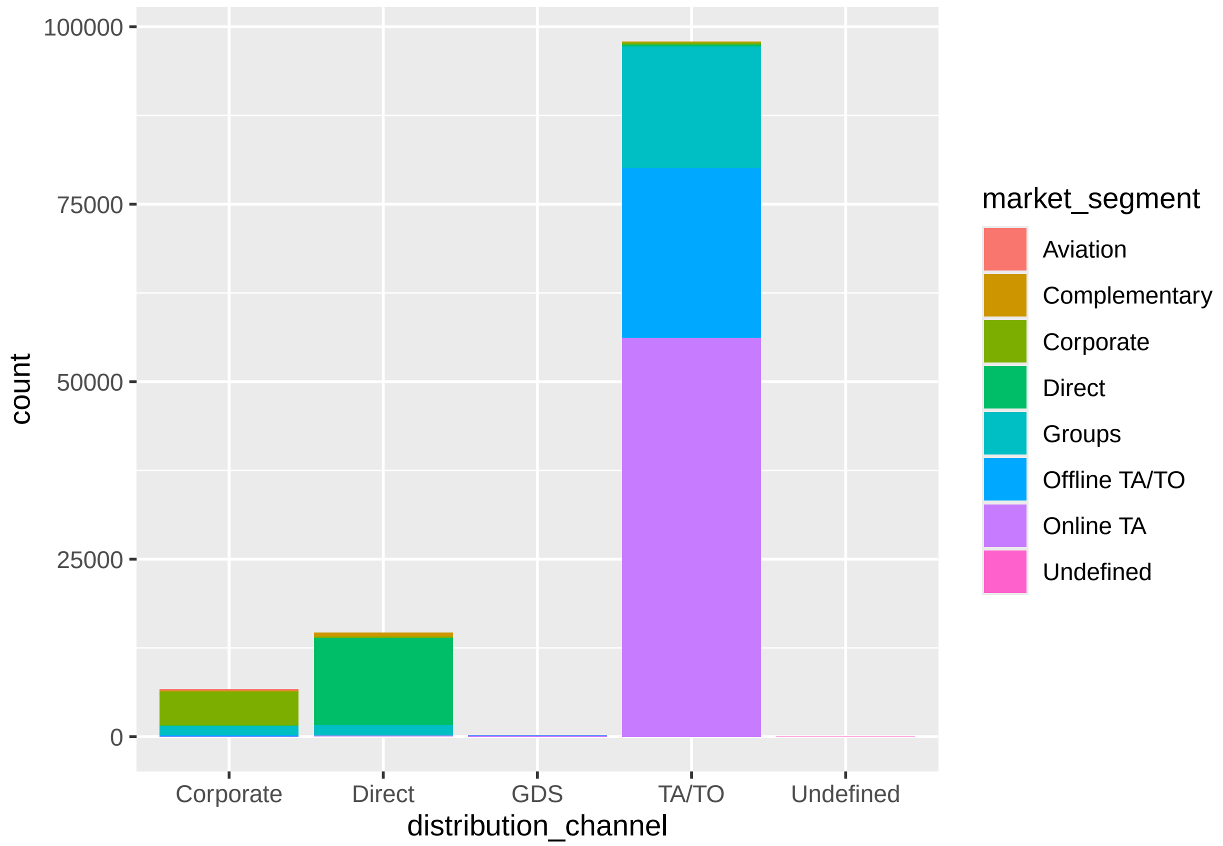
The TA/TO distribution type has the most number of bookings.

Now, I need to know if the number of bookings for each distribution type is different depending on whether or not there was a deposit or what market segment they represent:

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel, fill=deposit_type))
```



```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel, fill=market_segment))
```

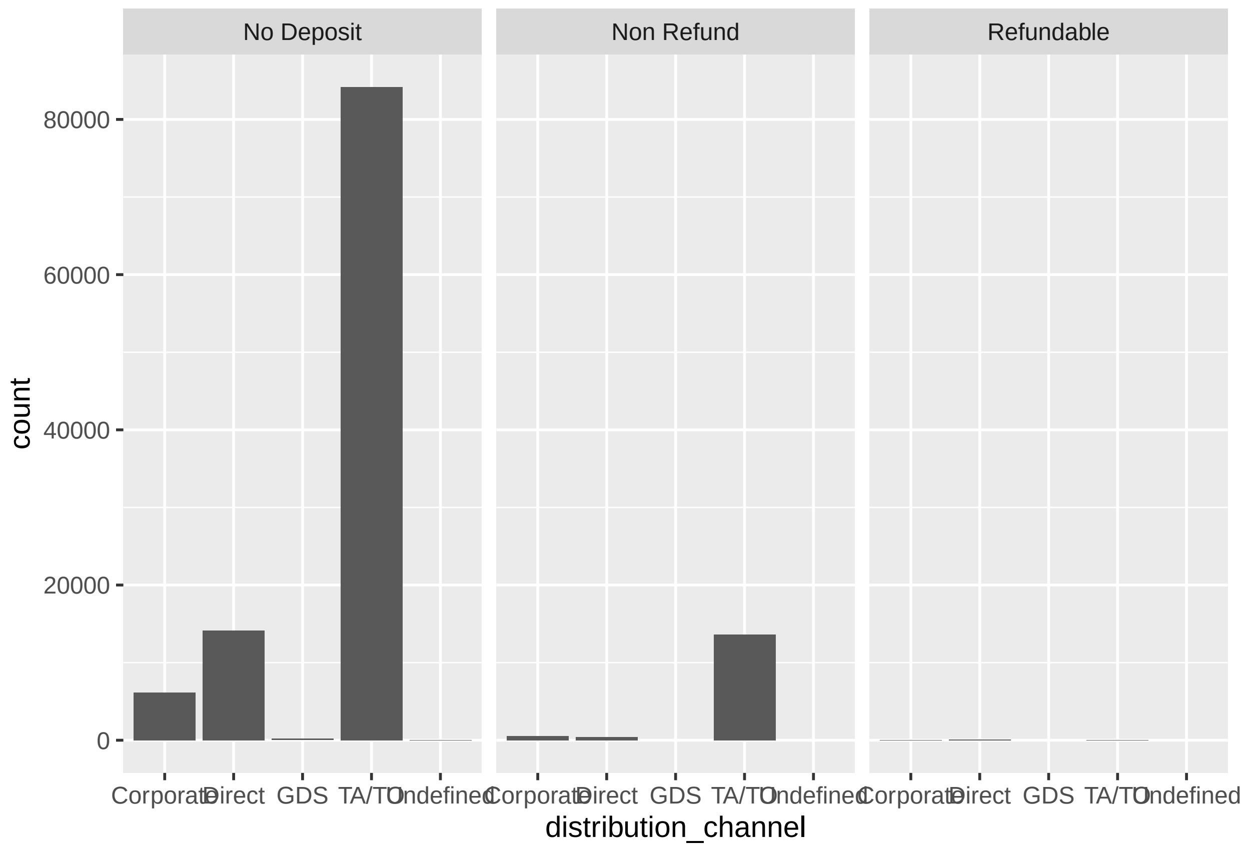


Step7:CreateFacets

The next task is to create separate charts for each deposit type and market segment to help the stakeholder understand the differences more clearly.

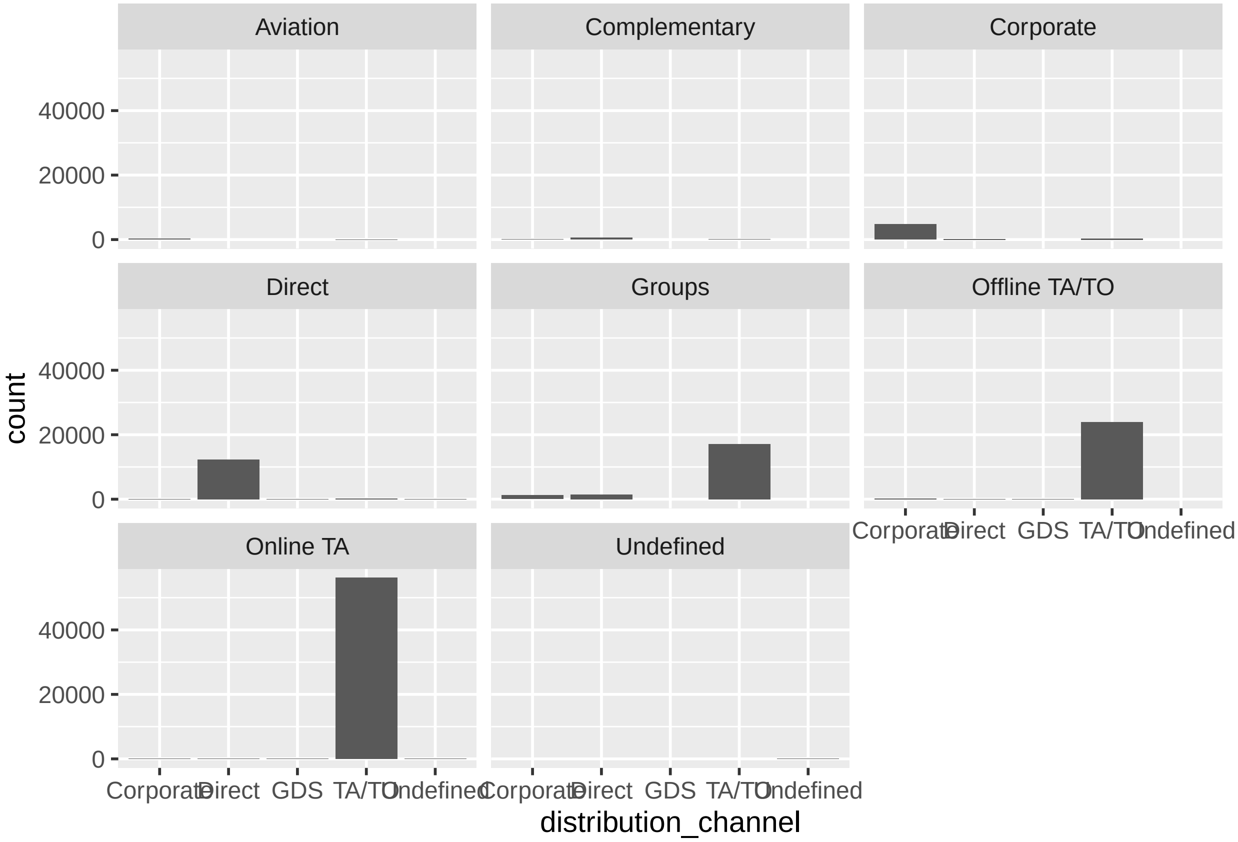
A different chart for each deposit type:

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~deposit_type)
```



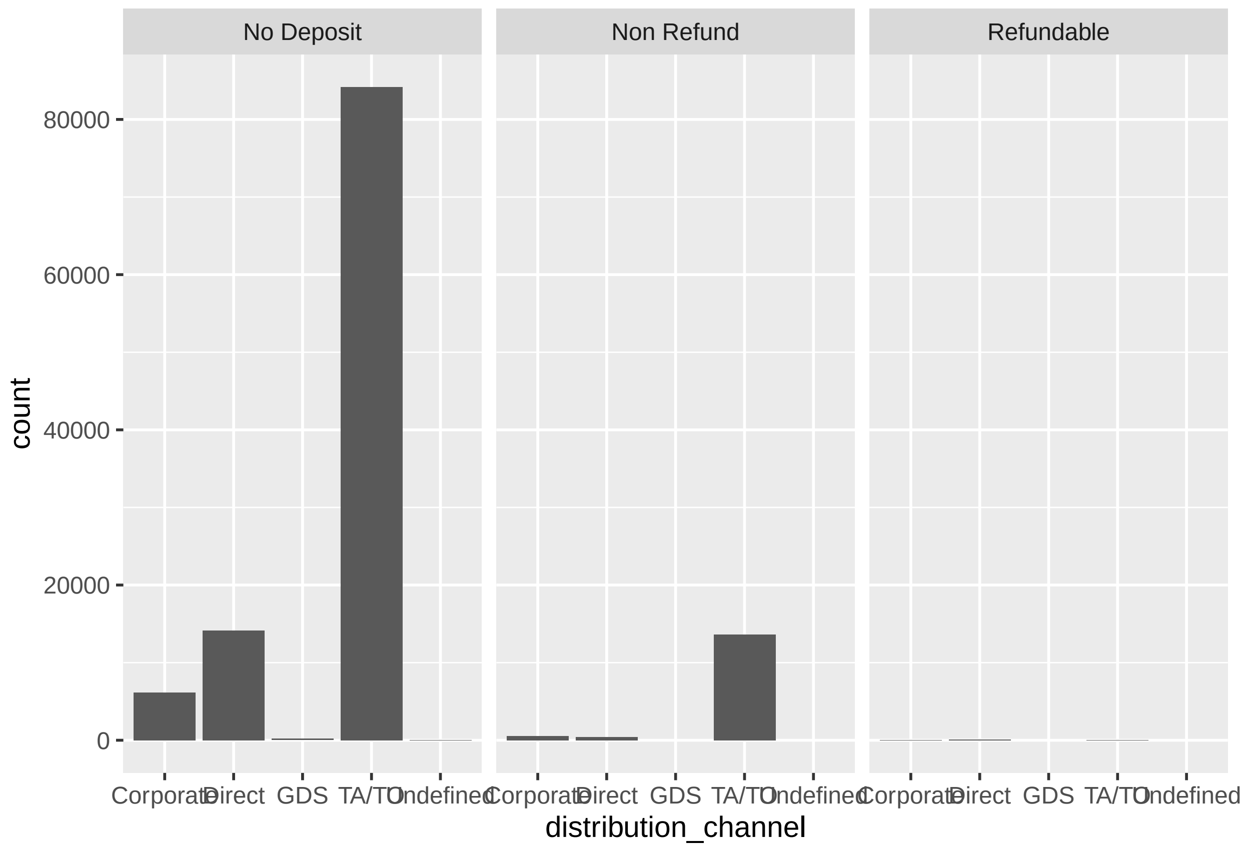
A different chart for each market segment:

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~market_segment)
```



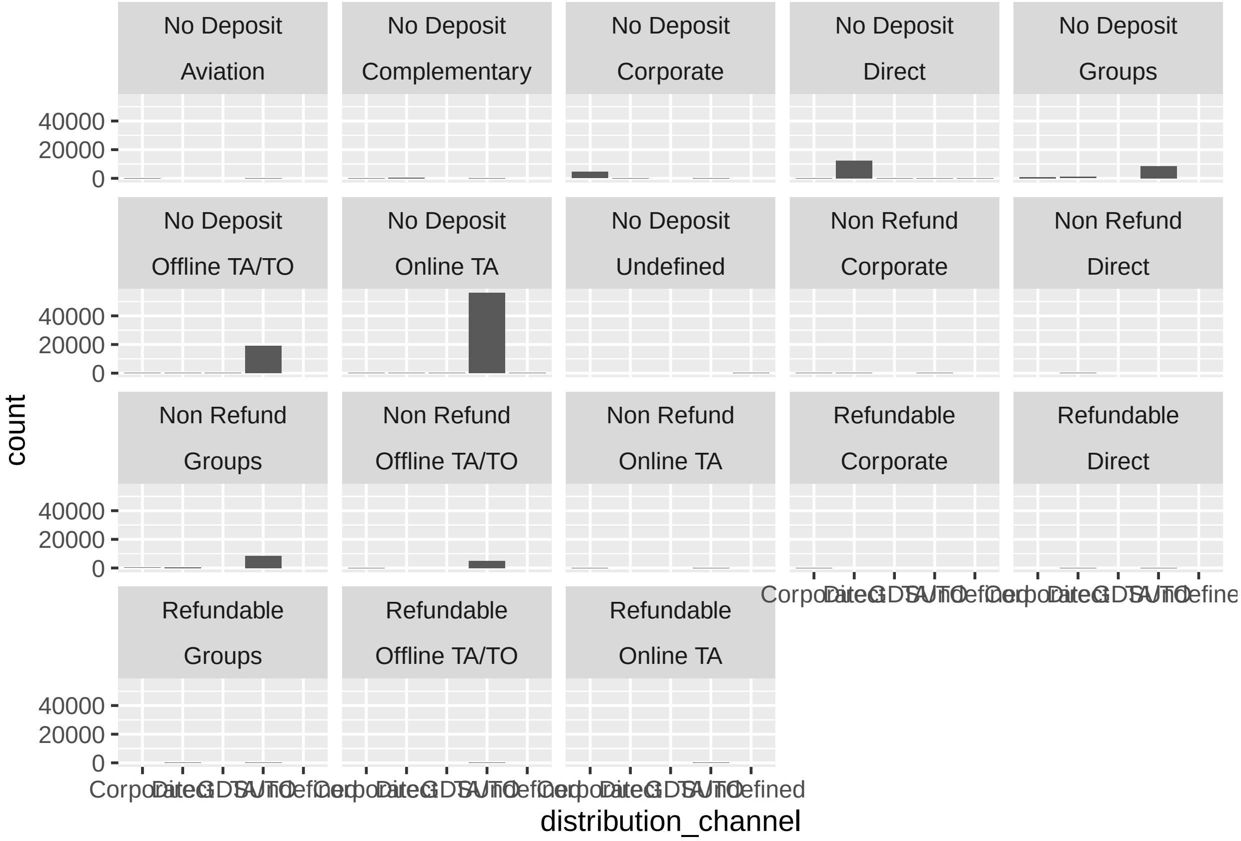
I used the *facet_grid* function to include plots even if they were empty.

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_grid(~deposit_type)
```



Finally, I put all of this in one chart to explore the differences by deposit type and market segment:

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = distribution_channel)) +
  facet_wrap(~deposit_type~market_segment)
```



Step 8: Apply Filters

After considering all the data, my stakeholder decides to send the promotion to families that make online bookings for city hotels. The online segment is the fastest growing segment, and families tend to spend more at city hotels than other types of guests. I need to create a plot that shows the relationship between lead time and guests traveling with children for online bookings at city hotels. This will give the stakeholder a better idea of the specific timing for the promotion:

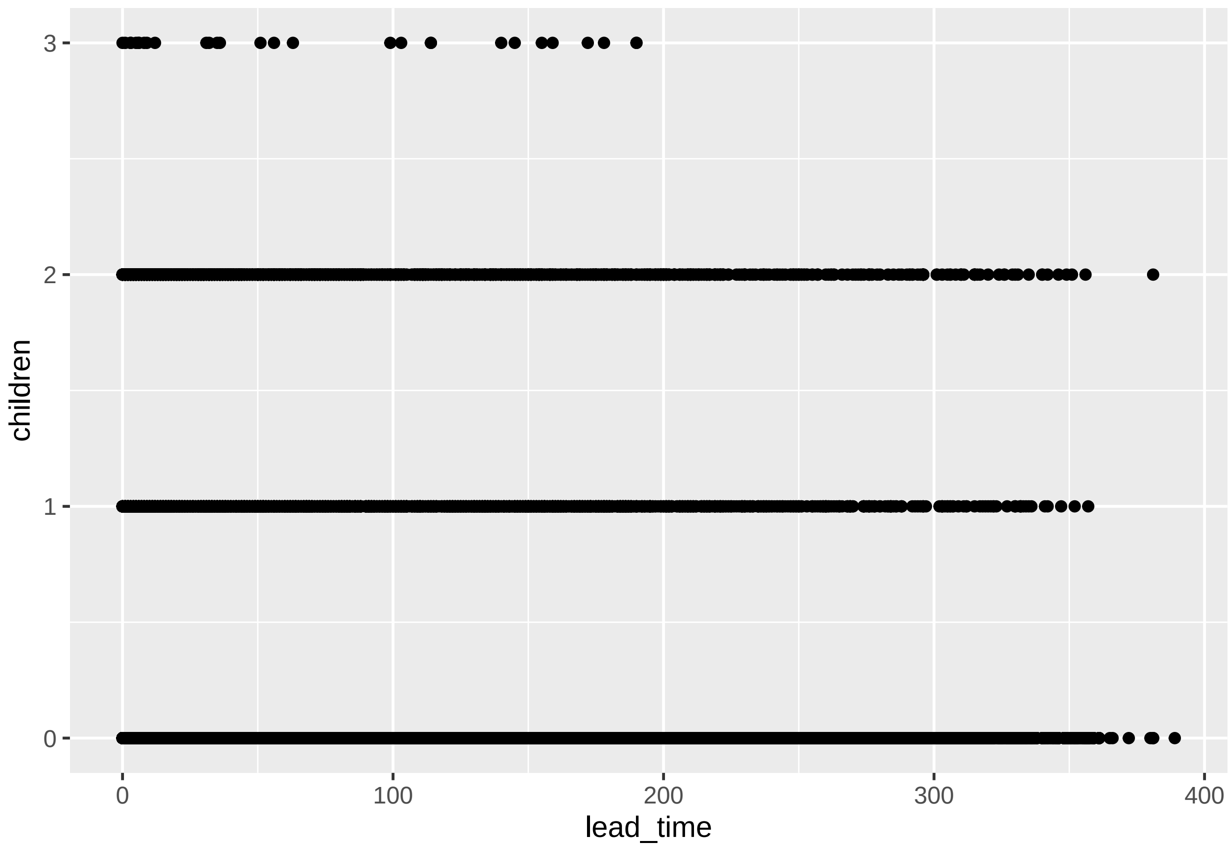
```
onlineta_city_hotels <- filter(bookings_df,
                               (hotel=="City Hotel" &
                                  bookings_df$market_segment=="Online TA"))
```

```
View(onlineta_city_hotels)
```

```
onlineta_city_hotels_v2 <- bookings_df %>%
  filter(hotel=="City Hotel") %>%
  filter(market_segment=="Online TA")
```

```
ggplot(data = onlineta_city_hotels) +
  geom_point(mapping = aes(x = lead_time, y = children))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_point()‘).
```

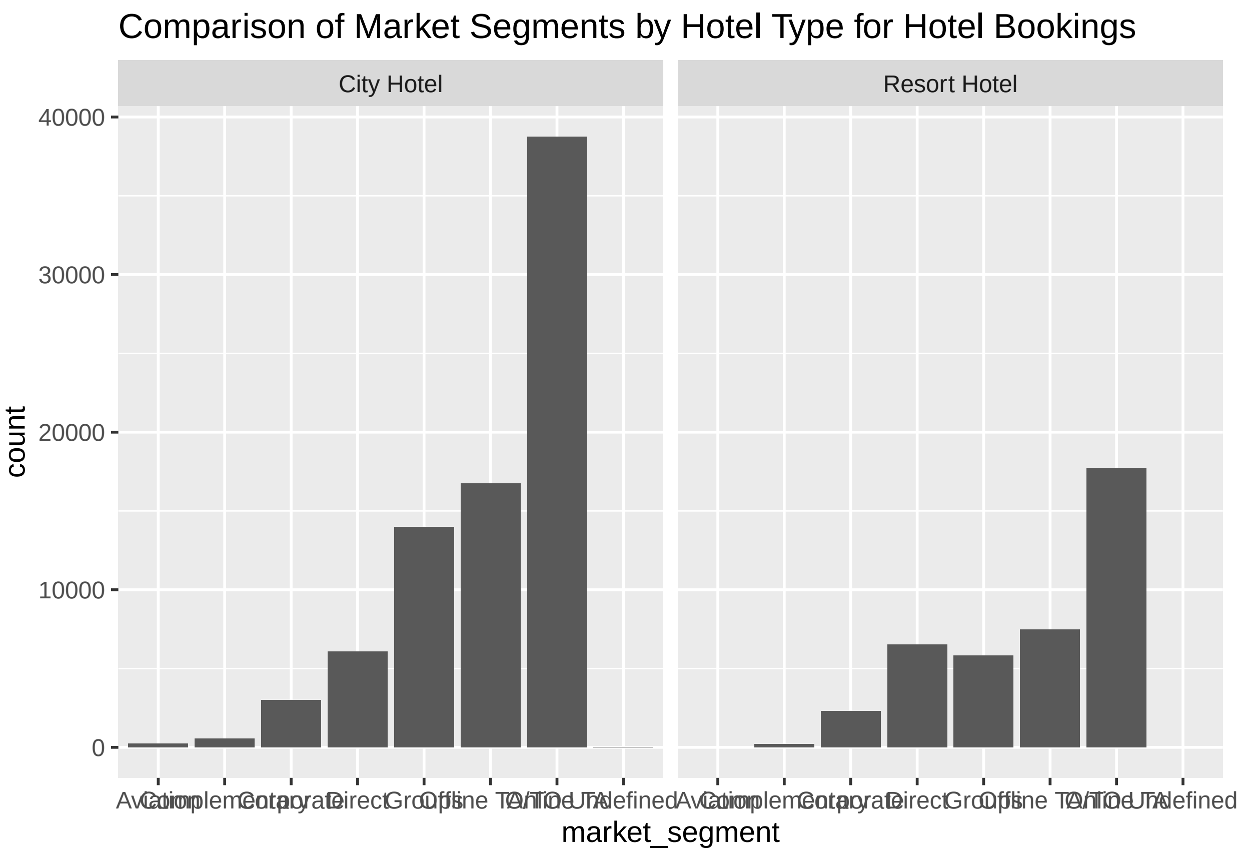


The plot reveals that bookings with children tend to have a shorter lead time, and bookings with 3 children have a significantly shorter lead time (<200 days). So, promotions targeting families can be made closer to the valid booking dates.

Step9:AddAnnotations

In these visualizations it is unclear where the data is from, what the main takeaway is, or even what the data is showing.To explain all of that, I leveraged annotations in*ggplot2*:

```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Comparison of Market Segments by Hotel Type for Hotel Bookings")
```



```r
min(bookings_df$arrival_date_year)
```

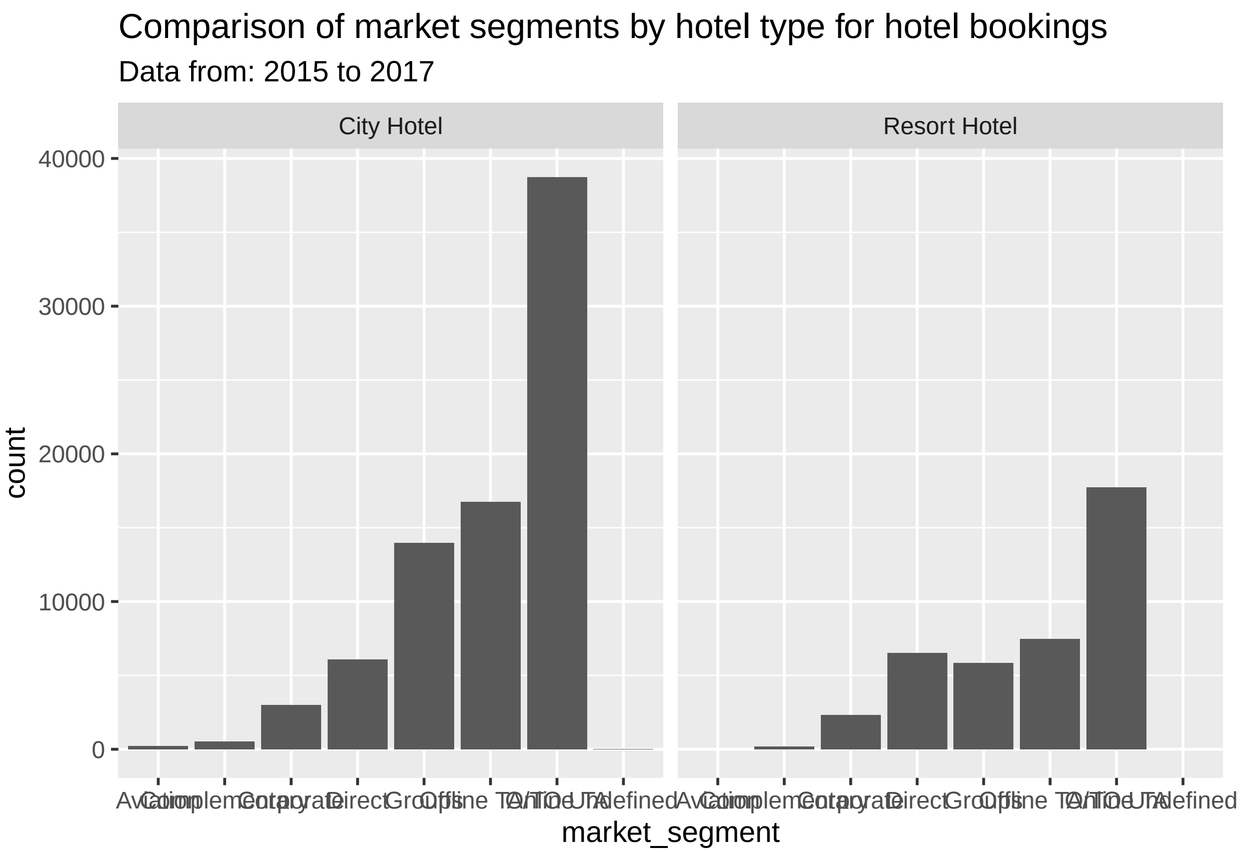
```
## [1] 2015
```

```r
max(bookings_df$arrival_date_year)
```

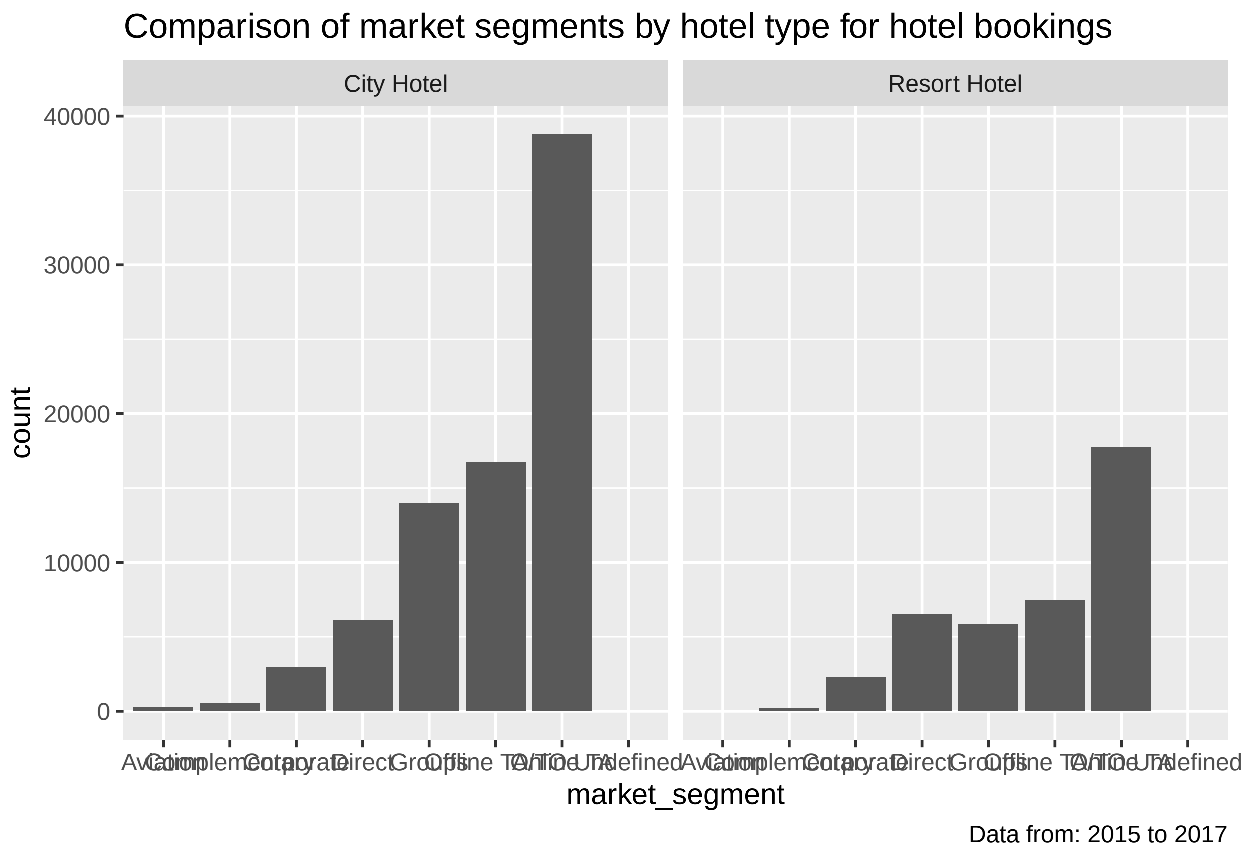
```
## [1] 2017
```

```r
mindate <- min(bookings_df$arrival_date_year)
maxdate <- max(bookings_df$arrival_date_year)
```

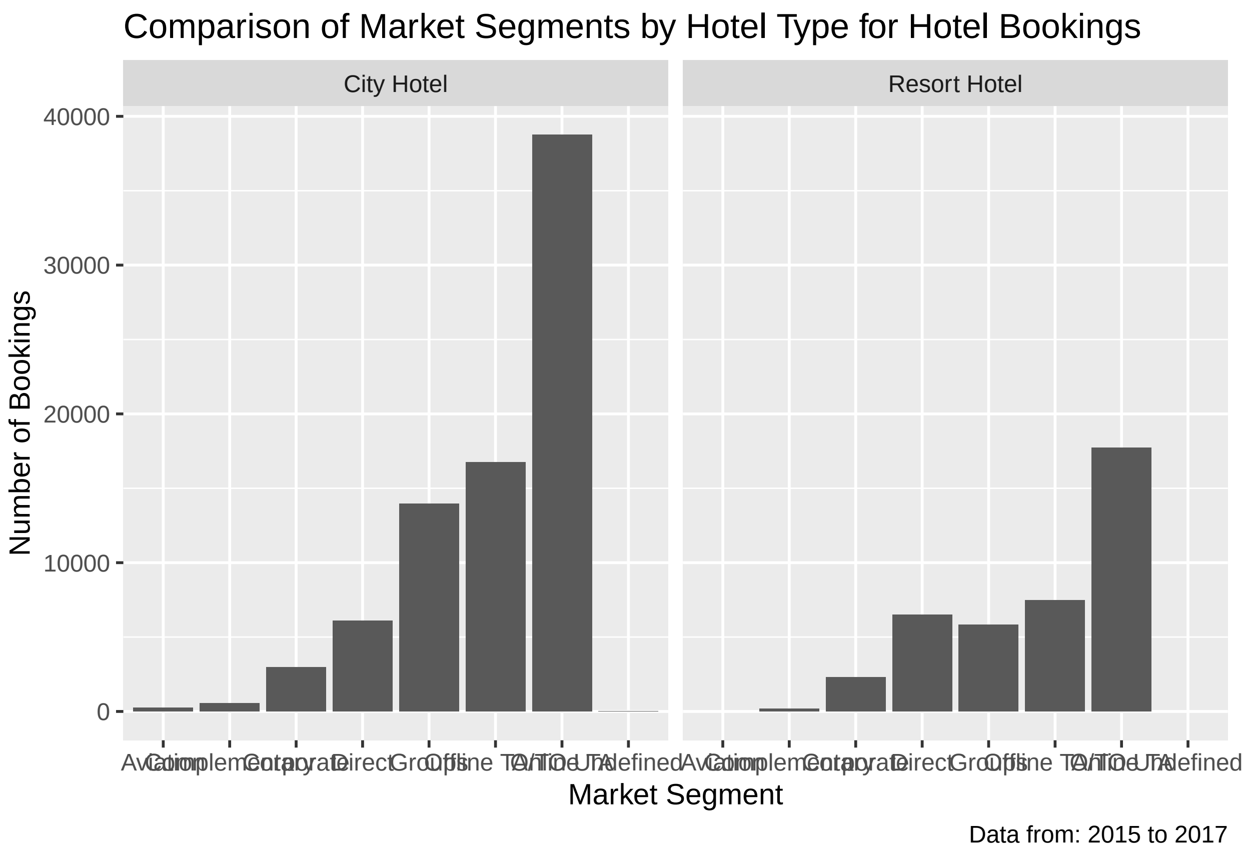
```r
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       subtitle=paste0("Data from: ", mindate, " to ", maxdate))
```



```
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Comparison of market segments by hotel type for hotel bookings",
       caption=paste0("Data from: ", mindate, " to ", maxdate))
```



```r
ggplot(data = bookings_df) +
  geom_bar(mapping = aes(x = market_segment)) +
  facet_wrap(~hotel) +
  labs(title="Comparison of Market Segments by Hotel Type for Hotel Bookings",
       caption=paste0("Data from: ", mindate, " to ", maxdate),
       x="Market Segment",
       y="Number of Bookings")
```



Step 10: Save the Plot

```
ggsave('hotel_booking_chart.png',
       width=16,
       height=8)
```