hotel-bookings

February 1, 2025

0.1 Author: Danish Azeem

0.2 Email Address: danishazeem365@gmail.com

0.3 Github: https://github.com/danishazeem365

0.4 About Dataset

- **Description**
  The Data Set was downloaded from Kaggle, from the following link

About Dataset
Context Google PlayStore Android App Data. (2.3 Million+ App Data) Backup repo: https://github.com/gauthamp10/Google-Playstore-Dataset

Content:
I've collected the data with the help of Python script (Scrapy) running on a cloud vm instance. The data was collected in the month of June 2021.

Inspiration
Took inspiration from: https://www.kaggle.com/lava18/google-play-store-apps to build a big database for students and researchers.

1 Step 1: Importing Libraries and Loading the Dataset

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('/content/drive/My Drive/Data Sets/hotel_bookings.csv')
print(df.head())
```

```
Mounted at /content/drive
          hotel  is_canceled  lead_time  arrival_date_year arrival_date_month  \
0  Resort Hotel            0        342               2015               July
1  Resort Hotel            0        737               2015               July
2  Resort Hotel            0          7               2015               July
3  Resort Hotel            0         13               2015               July
```

```
4  Resort Hotel            0         14              2015               July

   arrival_date_week_number  arrival_date_day_of_month  \
0                        27                          1
1                        27                          1
2                        27                          1
3                        27                          1
4                        27                          1

   stays_in_weekend_nights  stays_in_week_nights  adults  …  deposit_type  \
0                        0                     0       2  …    No Deposit
1                        0                     0       2  …    No Deposit
2                        0                     1       1  …    No Deposit
3                        0                     1       1  …    No Deposit
4                        0                     2       2  …    No Deposit

   agent company days_in_waiting_list customer_type   adr  \
0    NaN     NaN                    0     Transient   0.0
1    NaN     NaN                    0     Transient   0.0
2    NaN     NaN                    0     Transient  75.0
3  304.0     NaN                    0     Transient  75.0
4  240.0     NaN                    0     Transient  98.0

   required_car_parking_spaces  total_of_special_requests  reservation_status  \
0                            0                          0           Check-Out
1                            0                          0           Check-Out
2                            0                          0           Check-Out
3                            0                          0           Check-Out
4                            0                          1           Check-Out

  reservation_status_date
0              2015-07-01
1              2015-07-01
2              2015-07-02
3              2015-07-02
4              2015-07-03

[5 rows x 32 columns]
```

Note: Some the output of notebook does not present the complete output, therefore we can increase the limit of columns view and row view by using these commands:

[3]:
```python
pd.set_option('display.max_columns', None) # this is to display all the columns␣
↪in the dataframe
pd.set_option('display.max_rows', None) # this is to display all the rows in␣
↪the dataframe
```

[4]:
```python
# hide all warnings runtime
import warnings
warnings.filterwarnings('ignore')
```

2 Step 2: Understanding the Dataset

[5]:
```python
# Display the first few rows
df.head()
```

[5]:
```
          hotel  is_canceled  lead_time  arrival_date_year arrival_date_month  \
0  Resort Hotel            0        342               2015               July
1  Resort Hotel            0        737               2015               July
2  Resort Hotel            0          7               2015               July
3  Resort Hotel            0         13               2015               July
4  Resort Hotel            0         14               2015               July

   arrival_date_week_number  arrival_date_day_of_month  \
0                        27                          1
1                        27                          1
2                        27                          1
3                        27                          1
4                        27                          1

   stays_in_weekend_nights  stays_in_week_nights  adults  children  babies  \
0                        0                     0       2       0.0       0
1                        0                     0       2       0.0       0
2                        0                     1       1       0.0       0
3                        0                     1       1       0.0       0
4                        0                     2       2       0.0       0

  meal country market_segment distribution_channel  is_repeated_guest  \
0   BB     PRT         Direct               Direct                  0
1   BB     PRT         Direct               Direct                  0
2   BB     GBR         Direct               Direct                  0
3   BB     GBR      Corporate            Corporate                  0
4   BB     GBR      Online TA                TA/TO                  0

   previous_cancellations  previous_bookings_not_canceled reserved_room_type  \
0                       0                               0                  C
1                       0                               0                  C
2                       0                               0                  A
3                       0                               0                  A
4                       0                               0                  A

  assigned_room_type  booking_changes deposit_type  agent  company  \
0                  C                3   No Deposit    NaN      NaN
```

```
1                 C              4   No Deposit    NaN      NaN
2                 C              0   No Deposit    NaN      NaN
3                 A              0   No Deposit  304.0      NaN
4                 A              0   No Deposit  240.0      NaN

   days_in_waiting_list customer_type   adr  required_car_parking_spaces  \
0                     0     Transient   0.0                            0
1                     0     Transient   0.0                            0
2                     0     Transient  75.0                            0
3                     0     Transient  75.0                            0
4                     0     Transient  98.0                            0

   total_of_special_requests reservation_status reservation_status_date
0                          0          Check-Out              2015-07-01
1                          0          Check-Out              2015-07-01
2                          0          Check-Out              2015-07-02
3                          0          Check-Out              2015-07-02
4                          1          Check-Out              2015-07-03
```

[6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
```

```
 21   booking_changes                 119390 non-null  int64
 22   deposit_type                    119390 non-null  object
 23   agent                           103050 non-null  float64
 24   company                         6797 non-null    float64
 25   days_in_waiting_list            119390 non-null  int64
 26   customer_type                   119390 non-null  object
 27   adr                             119390 non-null  float64
 28   required_car_parking_spaces     119390 non-null  int64
 29   total_of_special_requests       119390 non-null  int64
 30   reservation_status              119390 non-null  object
 31   reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

2.1 # Observations

1. There are 119390 rows and 32 columns in the dataset
2. The columns are of different data types
3. The columns in the datasets are:
   - 'hotel', 'is_canceled', 'lead_time', 'arrival_date_year', 'arrival_date_month', 'arrival_date_week_number', 'arrival_date_day_of_month', 'stays_in_weekend_nights', 'stays_in_week_nights', 'adults', 'children', 'babies', 'meal', 'country', 'market_segment', 'distribution_channel', 'is_repeated_guest', 'previous_cancellations', 'previous_bookings_not_canceled', 'reserved_room_type', 'assigned_room_type', 'booking_changes', 'deposit_type', 'agent', 'company', 'days_in_waiting_list', 'customer_type', 'adr', 'required_car_parking_spaces', 'total_of_special_requests', 'reservation_status', 'reservation_status_date',
4. There are some missing values in the dataset which we will read in details and deal later on in the notebook.

[7]:
```python
df.sample(50)
```

[7]:
```
                hotel  is_canceled  lead_time  arrival_date_year  \
55663      City Hotel            1        178               2016
104566     City Hotel            0          1               2017
36354    Resort Hotel            0          1               2017
66001      City Hotel            1        219               2017
61689      City Hotel            1         89               2016
20323    Resort Hotel            0          1               2016
105732     City Hotel            0         36               2017
88622      City Hotel            0        103               2016
13260    Resort Hotel            1        178               2017
9885     Resort Hotel            1         40               2017
1083     Resort Hotel            0        135               2015
8170     Resort Hotel            0        336               2016
91775      City Hotel            0        192               2016
29091    Resort Hotel            0         68               2016
106773     City Hotel            0          3               2017
```

```
37134   Resort Hotel            0        281              2017
75445     City Hotel            1        327              2015
11941   Resort Hotel            1         87              2017
116146    City Hotel            0        414              2017
78508     City Hotel            0          3              2015
112515    City Hotel            0         22              2017
78409     City Hotel            0          5              2015
8581    Resort Hotel            1        383              2016
58118     City Hotel            1         92              2016
16779   Resort Hotel            0         32              2015
64812     City Hotel            1        111              2017
78274     City Hotel            1         14              2015
114842    City Hotel            0          1              2017
95104     City Hotel            0        109              2016
96017     City Hotel            0        143              2016
4960    Resort Hotel            0        275              2016
68191     City Hotel            1        320              2017
10491   Resort Hotel            1        125              2017
65368     City Hotel            1        162              2017
39251   Resort Hotel            0         46              2017
21141   Resort Hotel            0          0              2016
49419     City Hotel            1         99              2016
76478     City Hotel            1        414              2015
44058     City Hotel            0         83              2015
56116     City Hotel            1         21              2016
76448     City Hotel            1        379              2015
101503    City Hotel            0         66              2016
30453   Resort Hotel            0         34              2016
6236    Resort Hotel            0        229              2016
77530     City Hotel            0        263              2015
85878     City Hotel            0         61              2016
49058     City Hotel            1         68              2016
66876     City Hotel            1        156              2017
71481     City Hotel            1        155              2017
26062   Resort Hotel            0        158              2016

       arrival_date_month  arrival_date_week_number  \
55663              August                        34
104566            January                         2
36354                 May                        20
66001               April                        15
61689            December                        52
20323             January                         4
105732           February                         7
88622                 May                        19
13260              August                        32
9885              January                         3
```

```
1083               August                        33
8170            September                        38
91775                June                        26
29091             October                        43
106773           February                         8
37134                June                        22
75445           September                        37
11941                June                        23
116146               July                        28
78508             October                        41
112515               July                        30
78409             October                        42
8581              October                        41
58118             October                        41
16779           September                        37
64812               March                        11
78274             October                        42
114842               June                        26
95104              August                        33
96017              August                        35
4960                April                        16
68191                 May                        20
10491               March                        10
65368               April                        13
39251              August                        32
21141            February                         8
49419               April                        16
76478            December                        49
44058           September                        40
56116              August                        36
76448             October                        44
101503           November                        46
30453            November                        48
6236                  May                        23
77530           September                        39
85878               March                        12
49058               April                        15
66876               April                        17
71481                July                        27
26062                July                        29

        arrival_date_day_of_month  stays_in_weekend_nights  \
55663                          18                        0
104566                         14                        2
36354                          17                        0
66001                          12                        0
61689                          22                        2
```

20323	22	0
105732	12	2
88622	5	1
13260	6	2
9885	16	1
1083	13	1
8170	15	0
91775	24	0
29091	16	2
106773	24	2
37134	1	2
75445	9	0
11941	4	2
116146	13	0
78508	5	2
112515	25	0
78409	12	1
8581	6	1
58118	6	0
16779	6	2
64812	16	2
78274	15	0
114842	27	0
95104	12	1
96017	25	0
4960	10	2
68191	15	1
10491	9	2
65368	1	2
39251	8	0
21141	18	0
49419	13	0
76478	5	2
44058	30	0
56116	29	1
76448	31	1
101503	10	0
30453	21	1
6236	30	0
77530	21	1
85878	19	2
49058	5	0
66876	26	0
71481	6	0
26062	11	3

```
        stays_in_week_nights  adults  children  babies       meal country  \
```

55663	3	1	1.0	0	BB	ITA
104566	1	2	0.0	0	HB	FRA
36354	1	1	0.0	0	BB	PRT
66001	4	2	0.0	0	BB	SWE
61689	3	2	0.0	0	SC	USA
20323	2	2	0.0	0	BB	PRT
105732	1	2	0.0	0	SC	GBR
88622	3	2	0.0	0	BB	NOR
13260	5	2	0.0	0	BB	PRT
9885	2	1	0.0	0	Undefined	PRT
1083	3	2	0.0	0	HB	ESP
8170	3	1	0.0	0	BB	GBR
91775	2	1	0.0	0	BB	GBR
29091	2	2	0.0	0	BB	CN
106773	5	2	0.0	0	BB	DNK
37134	5	2	0.0	0	HB	GBR
75445	2	2	0.0	0	BB	PRT
11941	2	2	0.0	0	BB	PRT
116146	2	1	0.0	0	HB	DEU
78508	5	1	0.0	0	BB	JPN
112515	2	1	0.0	0	BB	PRT
78409	1	2	0.0	0	BB	ITA
8581	3	2	0.0	0	BB	PRT
58118	3	2	0.0	0	SC	NLD
16779	4	2	0.0	0	BB	IRL
64812	3	2	0.0	0	SC	ITA
78274	1	1	0.0	0	BB	PRT
114842	1	1	0.0	0	BB	ESP
95104	2	2	0.0	0	SC	ITA
96017	3	2	0.0	0	BB	CZE
4960	5	2	0.0	0	HB	GBR
68191	1	2	0.0	0	BB	PRT
10491	4	2	0.0	0	BB	PRT
65368	1	2	0.0	0	BB	PRT
39251	5	2	0.0	0	BB	GBR
21141	1	2	0.0	1	BB	PRT
49419	2	2	0.0	0	BB	DEU
76478	1	2	0.0	0	BB	PRT
44058	1	2	0.0	0	HB	PRT
56116	3	2	0.0	0	BB	DZA
76448	1	2	0.0	0	BB	PRT
101503	3	2	0.0	0	SC	FRA
30453	1	2	0.0	0	BB	FRA
6236	0	2	0.0	0	BB	PRT
77530	0	2	0.0	0	BB	PRT
85878	2	2	0.0	0	BB	ESP
49058	4	2	0.0	0	BB	ITA

66876	3	2	0.0	0	BB	PRT
71481	3	2	0.0	0	BB	NOR
26062	6	2	0.0	0	BB	DEU

	market_segment	distribution_channel	is_repeated_guest	\
55663	Online TA	TA/TO	0	
104566	Online TA	TA/TO	0	
36354	Direct	Direct	0	
66001	Direct	Direct	0	
61689	Online TA	TA/TO	0	
20323	Direct	Direct	0	
105732	Online TA	TA/TO	0	
88622	Online TA	TA/TO	0	
13260	Online TA	TA/TO	0	
9885	Groups	Direct	0	
1083	Online TA	TA/TO	0	
8170	Groups	Direct	0	
91775	Offline TA/TO	TA/TO	0	
29091	Offline TA/TO	TA/TO	0	
106773	Online TA	TA/TO	0	
37134	Offline TA/TO	TA/TO	0	
75445	Groups	TA/TO	0	
11941	Online TA	TA/TO	0	
116146	Groups	TA/TO	0	
78508	Direct	Direct	0	
112515	Direct	Direct	1	
78409	Groups	TA/TO	0	
8581	Groups	TA/TO	0	
58118	Online TA	TA/TO	0	
16779	Online TA	TA/TO	0	
64812	Online TA	TA/TO	0	
78274	Offline TA/TO	TA/TO	0	
114842	Corporate	Corporate	0	
95104	Online TA	TA/TO	0	
96017	Offline TA/TO	TA/TO	0	
4960	Groups	Direct	0	
68191	Offline TA/TO	TA/TO	0	
10491	Online TA	TA/TO	0	
65368	Groups	TA/TO	0	
39251	Online TA	TA/TO	0	
21141	Direct	Direct	0	
49419	Online TA	TA/TO	0	
76478	Groups	TA/TO	0	
44058	Offline TA/TO	TA/TO	0	
56116	Online TA	TA/TO	0	
76448	Groups	TA/TO	0	
101503	Online TA	TA/TO	0	

```
30453       Online TA              TA/TO                    0
6236           Groups             Direct                    0
77530   Offline TA/TO              TA/TO                    0
85878   Offline TA/TO              TA/TO                    0
49058       Online TA              TA/TO                    0
66876          Groups              TA/TO                    0
71481       Online TA              TA/TO                    0
26062   Offline TA/TO              TA/TO                    0

        previous_cancellations  previous_bookings_not_canceled  \
55663                        0                               0
104566                       0                               0
36354                        0                               0
66001                        0                               0
61689                        0                               0
20323                        0                               0
105732                       0                               0
88622                        0                               0
13260                        0                               0
9885                         0                               0
1083                         0                               0
8170                         0                               0
91775                        0                               0
29091                        0                               0
106773                       0                               0
37134                        0                               0
75445                        1                               0
11941                        0                               0
116146                       0                               0
78508                        0                               0
112515                       0                               8
78409                        0                               1
8581                         0                               0
58118                        0                               0
16779                        0                               0
64812                        0                               0
78274                        1                               0
114842                       0                               0
95104                        0                               0
96017                        0                               0
4960                         0                               0
68191                        0                               0
10491                        0                               0
65368                        0                               0
39251                        0                               0
21141                        0                               0
49419                        0                               0
```

```
76478                             1                              0
44058                             0                              0
56116                             0                              0
76448                             1                              0
101503                            0                              0
30453                             0                              0
6236                              0                              0
77530                             0                              0
85878                             0                              0
49058                             0                              0
66876                             0                              0
71481                             0                              0
26062                             0                              0

       reserved_room_type assigned_room_type  booking_changes deposit_type  \
55663                   A                  A                0   No Deposit
104566                  F                  F                0   No Deposit
36354                   A                  D                0   No Deposit
66001                   B                  B                1   No Deposit
61689                   A                  A                0   No Deposit
20323                   A                  D                0   No Deposit
105732                  A                  A                0   No Deposit
88622                   D                  D                0   No Deposit
13260                   D                  D                0   No Deposit
9885                    A                  A                1   No Deposit
1083                    E                  E                0   No Deposit
8170                    A                  A                1   No Deposit
91775                   A                  A                0   No Deposit
29091                   A                  A                0   No Deposit
106773                  D                  D                0   No Deposit
37134                   A                  A                0   No Deposit
75445                   A                  A                0   Non Refund
11941                   E                  E                0   No Deposit
116146                  A                  A                1   No Deposit
78508                   D                  D                1   No Deposit
112515                  D                  D                3   No Deposit
78409                   A                  A                0   No Deposit
8581                    A                  A                0   No Deposit
58118                   A                  A                0   No Deposit
16779                   A                  A                0   No Deposit
64812                   A                  A                0   No Deposit
78274                   A                  A                0   No Deposit
114842                  A                  D                0   No Deposit
95104                   A                  A                0   No Deposit
96017                   A                  A                0   No Deposit
4960                    D                  D                0   No Deposit
68191                   A                  A                0   No Deposit
```

10491	A	A	0	No Deposit
65368	A	A	0	No Deposit
39251	A	A	1	No Deposit
21141	E	E	0	No Deposit
49419	A	A	0	No Deposit
76478	A	A	0	Non Refund
44058	A	D	0	No Deposit
56116	A	A	0	No Deposit
76448	A	A	0	Non Refund
101503	A	A	0	No Deposit
30453	E	E	0	No Deposit
6236	A	D	5	No Deposit
77530	A	D	0	No Deposit
85878	A	B	1	No Deposit
49058	A	A	0	No Deposit
66876	A	A	0	Non Refund
71481	A	A	0	No Deposit
26062	D	D	0	No Deposit

	agent	company	days_in_waiting_list	customer_type	adr	\
55663	9.0	NaN	0	Transient	124.50	
104566	83.0	NaN	0	Transient	175.20	
36354	NaN	NaN	0	Transient	76.00	
66001	14.0	NaN	0	Transient	101.25	
61689	9.0	NaN	0	Transient	74.80	
20323	NaN	NaN	0	Transient	48.00	
105732	89.0	NaN	0	Transient	60.80	
88622	7.0	NaN	0	Transient	95.88	
13260	240.0	NaN	0	Transient	200.00	
9885	NaN	NaN	0	Transient-Party	55.00	
1083	240.0	NaN	0	Transient	196.00	
8170	NaN	223.0	0	Transient-Party	60.00	
91775	34.0	NaN	0	Transient-Party	95.00	
29091	96.0	NaN	0	Transient	46.00	
106773	9.0	NaN	0	Transient	100.30	
37134	243.0	NaN	0	Contract	86.70	
75445	1.0	NaN	0	Transient-Party	62.00	
11941	15.0	NaN	0	Transient	120.00	
116146	6.0	NaN	0	Transient-Party	91.50	
78508	NaN	NaN	0	Transient	148.80	
112515	NaN	485.0	0	Transient	75.00	
78409	1.0	NaN	0	Transient-Party	138.00	
8581	315.0	NaN	0	Transient-Party	48.00	
58118	9.0	NaN	0	Transient	108.00	
16779	241.0	NaN	0	Transient	100.87	
64812	9.0	NaN	0	Transient	74.80	
78274	99.0	NaN	0	Transient-Party	100.00	

```
114842    NaN    491.0                       0        Transient  110.00
95104     9.0      NaN                       0        Transient  107.10
96017    36.0      NaN                       0  Transient-Party  100.00
4960    273.0      NaN                       0  Transient-Party   74.45
68191   229.0      NaN                       0  Transient-Party   90.00
10491   240.0      NaN                       0        Transient   42.00
65368   296.0      NaN                       0  Transient-Party   98.00
39251   508.0      NaN                       0  Transient-Party  188.00
21141     NaN      NaN                       0        Transient   65.00
49419     9.0      NaN                       0        Transient   96.30
76478     1.0      NaN                       0        Transient   62.00
44058    26.0      NaN                      77  Transient-Party  112.20
56116     9.0      NaN                       0        Transient  124.25
76448     1.0      NaN                       0        Transient   62.00
101503    9.0      NaN                       0        Transient  106.50
30453   240.0      NaN                       0        Transient   69.00
6236      NaN    223.0                       0  Transient-Party    0.00
77530    21.0      NaN                       0  Transient-Party   67.00
85878   138.0      NaN                       0        Transient   63.58
49058     9.0      NaN                       0        Transient   90.95
66876    37.0      NaN                       0        Transient  100.00
71481     9.0      NaN                       0        Transient  107.10
26062    69.0      NaN                       0        Transient  112.70

        required_car_parking_spaces  total_of_special_requests  \
55663                             0                          0
104566                            0                          0
36354                             0                          0
66001                             0                          0
61689                             0                          1
20323                             0                          1
105732                            0                          0
88622                             0                          1
13260                             0                          2
9885                              0                          0
1083                              0                          1
8170                              0                          0
91775                             0                          0
29091                             0                          0
106773                            0                          0
37134                             0                          2
75445                             0                          0
11941                             0                          0
116146                            0                          0
78508                             0                          0
112515                            0                          1
78409                             0                          0
```

8581	0	0
58118	0	2
16779	0	1
64812	0	0
78274	0	0
114842	0	1
95104	0	1
96017	0	0
4960	0	0
68191	0	1
10491	0	2
65368	0	0
39251	0	3
21141	0	2
49419	0	1
76478	0	0
44058	0	0
56116	0	3
76448	0	0
101503	0	1
30453	1	1
6236	0	0
77530	0	0
85878	0	0
49058	0	0
66876	0	0
71481	0	0
26062	0	1

	reservation_status	reservation_status_date
55663	Canceled	2016-02-22
104566	Check-Out	2017-01-17
36354	Check-Out	2017-05-18
66001	Canceled	2016-09-15
61689	Canceled	2016-09-29
20323	Check-Out	2016-01-24
105732	Check-Out	2017-02-15
88622	Check-Out	2016-05-09
13260	Canceled	2017-06-05
9885	Canceled	2017-01-10
1083	Check-Out	2015-08-17
8170	Check-Out	2016-09-18
91775	Check-Out	2016-06-26
29091	Check-Out	2016-10-20
106773	Check-Out	2017-03-03
37134	Check-Out	2017-06-08
75445	Canceled	2015-07-02

```
11941           No-Show        2017-06-04
116146        Check-Out        2017-07-15
78508         Check-Out        2015-10-12
112515        Check-Out        2017-07-27
78409         Check-Out        2015-10-14
8581           Canceled        2016-03-04
58118          Canceled        2016-10-01
16779         Check-Out        2015-09-12
64812          Canceled        2017-02-01
78274          Canceled        2015-10-07
114842        Check-Out        2017-06-28
95104         Check-Out        2016-08-15
96017         Check-Out        2016-08-28
4960          Check-Out        2016-04-17
68191          Canceled        2017-04-25
10491          Canceled        2017-01-01
65368          Canceled        2017-01-10
39251         Check-Out        2017-08-13
21141         Check-Out        2016-02-19
49419          Canceled        2016-03-16
76478          Canceled        2015-07-23
44058         Check-Out        2015-10-01
56116          Canceled        2016-08-25
76448          Canceled        2015-07-23
101503        Check-Out        2016-11-13
30453         Check-Out        2016-11-23
6236          Check-Out        2016-05-30
77530         Check-Out        2015-09-22
85878         Check-Out        2016-03-23
49058          Canceled        2016-03-16
66876          Canceled        2016-11-21
71481          Canceled        2017-03-07
26062         Check-Out        2016-07-20
```

- `df.sample(50)` gives us whole picture or idea about data.

[8]:
```
df.columns
```

[8]:
```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
```

```
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

- let's see the exact column names which can be easily copied later

[9]:
```
# Descriptive statistics
df.describe(include='all')
```

[9]:
```
             hotel    is_canceled      lead_time  arrival_date_year  \
count       119390  119390.000000  119390.000000      119390.000000
unique           2            NaN            NaN                NaN
top     City Hotel            NaN            NaN                NaN
freq         79330            NaN            NaN                NaN
mean           NaN       0.370416     104.011416        2016.156554
std            NaN       0.482918     106.863097           0.707476
min            NaN       0.000000       0.000000        2015.000000
25%            NaN       0.000000      18.000000        2016.000000
50%            NaN       0.000000      69.000000        2016.000000
75%            NaN       1.000000     160.000000        2017.000000
max            NaN       1.000000     737.000000        2017.000000

       arrival_date_month  arrival_date_week_number  \
count              119390             119390.000000
unique                 12                       NaN
top                August                       NaN
freq                13877                       NaN
mean                  NaN                 27.165173
std                   NaN                 13.605138
min                   NaN                  1.000000
25%                   NaN                 16.000000
50%                   NaN                 28.000000
75%                   NaN                 38.000000
max                   NaN                 53.000000

        arrival_date_day_of_month  stays_in_weekend_nights  \
count               119390.000000            119390.000000
unique                        NaN                      NaN
top                           NaN                      NaN
freq                          NaN                      NaN
mean                    15.798241                 0.927599
std                      8.780829                 0.998613
min                      1.000000                 0.000000
25%                      8.000000                 0.000000
50%                     16.000000                 1.000000
75%                     23.000000                 2.000000
max                     31.000000                19.000000
```

	stays_in_week_nights	adults	children	babies	\
count	119390.000000	119390.000000	119386.000000	119390.000000	
unique	NaN	NaN	NaN	NaN	
top	NaN	NaN	NaN	NaN	
freq	NaN	NaN	NaN	NaN	
mean	2.500302	1.856403	0.103890	0.007949	
std	1.908286	0.579261	0.398561	0.097436	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	2.000000	0.000000	0.000000	
50%	2.000000	2.000000	0.000000	0.000000	
75%	3.000000	2.000000	0.000000	0.000000	
max	50.000000	55.000000	10.000000	10.000000	

	meal	country	market_segment	distribution_channel	is_repeated_guest	\
count	119390	118902	119390	119390	119390.000000	
unique	5	177	8	5	NaN	
top	BB	PRT	Online TA	TA/TO	NaN	
freq	92310	48590	56477	97870	NaN	
mean	NaN	NaN	NaN	NaN	0.031912	
std	NaN	NaN	NaN	NaN	0.175767	
min	NaN	NaN	NaN	NaN	0.000000	
25%	NaN	NaN	NaN	NaN	0.000000	
50%	NaN	NaN	NaN	NaN	0.000000	
75%	NaN	NaN	NaN	NaN	0.000000	
max	NaN	NaN	NaN	NaN	1.000000	

	previous_cancellations	previous_bookings_not_canceled	\
count	119390.000000	119390.000000	
unique	NaN	NaN	
top	NaN	NaN	
freq	NaN	NaN	
mean	0.087118	0.137097	
std	0.844336	1.497437	
min	0.000000	0.000000	
25%	0.000000	0.000000	
50%	0.000000	0.000000	
75%	0.000000	0.000000	
max	26.000000	72.000000	

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
count	119390	119390	119390.000000	119390	
unique	10	12	NaN	3	
top	A	A	NaN	No Deposit	
freq	85994	74053	NaN	104641	
mean	NaN	NaN	0.221124	NaN	
std	NaN	NaN	0.652306	NaN	

```
min                    NaN                 NaN          0.000000           NaN
25%                    NaN                 NaN          0.000000           NaN
50%                    NaN                 NaN          0.000000           NaN
75%                    NaN                 NaN          0.000000           NaN
max                    NaN                 NaN         21.000000           NaN

                agent      company  days_in_waiting_list customer_type  \
count   103050.000000  6797.000000         119390.000000        119390
unique            NaN          NaN                   NaN             4
top               NaN          NaN                   NaN     Transient
freq              NaN          NaN                   NaN         89613
mean        86.693382   189.266735              2.321149           NaN
std        110.774548   131.655015             17.594721           NaN
min          1.000000     6.000000              0.000000           NaN
25%          9.000000    62.000000              0.000000           NaN
50%         14.000000   179.000000              0.000000           NaN
75%        229.000000   270.000000              0.000000           NaN
max        535.000000   543.000000            391.000000           NaN

                  adr  required_car_parking_spaces  total_of_special_requests  \
count   119390.000000                119390.000000              119390.000000
unique            NaN                          NaN                        NaN
top               NaN                          NaN                        NaN
freq              NaN                          NaN                        NaN
mean       101.831122                     0.062518                   0.571363
std         50.535790                     0.245291                   0.792798
min         -6.380000                     0.000000                   0.000000
25%         69.290000                     0.000000                   0.000000
50%         94.575000                     0.000000                   0.000000
75%        126.000000                     0.000000                   1.000000
max       5400.000000                     8.000000                   5.000000

       reservation_status reservation_status_date
count              119390                  119390
unique                  3                     926
top             Check-Out              2015-10-21
freq                75166                    1461
mean                  NaN                     NaN
std                   NaN                     NaN
min                   NaN                     NaN
25%                   NaN                     NaN
50%                   NaN                     NaN
75%                   NaN                     NaN
max                   NaN                     NaN
```

- `describe(include='all')` gives statistical insights for both numeric and categorical columns.

[10]:
```python
# Define a mapping for meal codes to their full forms
meal_mapping = {
    "BB": "Bed and Breakfast",
    "FB": "Full Board",
    "HB": "Half Board",
    "SC": "Self Catering",
    "Undefined": "Undefined"
}

# Create a new column with the Meal plan names
df['Meal'] = df['meal'].map(meal_mapping)

# Display the updated DataFrame
print(df[['meal', 'Meal']].head())
```

```
  meal               Meal
0   BB  Bed and Breakfast
1   BB  Bed and Breakfast
2   BB  Bed and Breakfast
3   BB  Bed and Breakfast
4   BB  Bed and Breakfast
```

2.1.1 Explanation:

1. **Mapping Dictionary**: A dictionary is used to map the meal codes to their corresponding full descriptions.
2. **`map` Function**: The `map` method applies the dictionary mapping to each value in the `meal` column.
3. **New Column**: A new column (`Meal`) is created to store the full forms, leaving the original `meal` column unchanged.

After running this code, our DataFrame will have a new column called `Meal` with descriptive values like `"Bed and Breakfast"`, `"Full Board"`, etc.

[11]:
```python
df["country"].unique()
```

[11]:
```
array(['PRT', 'GBR', 'USA', 'ESP', 'IRL', 'FRA', nan, 'ROU', 'NOR', 'OMN',
       'ARG', 'POL', 'DEU', 'BEL', 'CHE', 'CN', 'GRC', 'ITA', 'NLD',
       'DNK', 'RUS', 'SWE', 'AUS', 'EST', 'CZE', 'BRA', 'FIN', 'MOZ',
       'BWA', 'LUX', 'SVN', 'ALB', 'IND', 'CHN', 'MEX', 'MAR', 'UKR',
       'SMR', 'LVA', 'PRI', 'SRB', 'CHL', 'AUT', 'BLR', 'LTU', 'TUR',
       'ZAF', 'AGO', 'ISR', 'CYM', 'ZMB', 'CPV', 'ZWE', 'DZA', 'KOR',
       'CRI', 'HUN', 'ARE', 'TUN', 'JAM', 'HRV', 'HKG', 'IRN', 'GEO',
       'AND', 'GIB', 'URY', 'JEY', 'CAF', 'CYP', 'COL', 'GGY', 'KWT',
       'NGA', 'MDV', 'VEN', 'SVK', 'FJI', 'KAZ', 'PAK', 'IDN', 'LBN',
       'PHL', 'SEN', 'SYC', 'AZE', 'BHR', 'NZL', 'THA', 'DOM', 'MKD',
       'MYS', 'ARM', 'JPN', 'LKA', 'CUB', 'CMR', 'BIH', 'MUS', 'COM',
       'SUR', 'UGA', 'BGR', 'CIV', 'JOR', 'SYR', 'SGP', 'BDI', 'SAU',
```

```
'VNM', 'PLW', 'QAT', 'EGY', 'PER', 'MLT', 'MWI', 'ECU', 'MDG',
'ISL', 'UZB', 'NPL', 'BHS', 'MAC', 'TGO', 'TWN', 'DJI', 'STP',
'KNA', 'ETH', 'IRQ', 'HND', 'RWA', 'KHM', 'MCO', 'BGD', 'IMN',
'TJK', 'NIC', 'BEN', 'VGB', 'TZA', 'GAB', 'GHA', 'TMP', 'GLP',
'KEN', 'LIE', 'GNB', 'MNE', 'UMI', 'MYT', 'FRO', 'MMR', 'PAN',
'BFA', 'LBY', 'MLI', 'NAM', 'BOL', 'PRY', 'BRB', 'ABW', 'AIA',
'SLV', 'DMA', 'PYF', 'GUY', 'LCA', 'ATA', 'GTM', 'ASM', 'MRT',
'NCL', 'KIR', 'SDN', 'ATF', 'SLE', 'LAO'], dtype=object)
```

[12]:

```
# Dictionary mapping country codes to country names
country_mapping = {
    'PRT': 'Portugal', 'GBR': 'United Kingdom', 'USA': 'United States', 'ESP':␣
 ↪'Spain',
    'IRL': 'Ireland', 'FRA': 'France', 'ROU': 'Romania', 'NOR': 'Norway', 'OMN':
 ↪ 'Oman',
    'ARG': 'Argentina', 'POL': 'Poland', 'DEU': 'Germany', 'BEL': 'Belgium',␣
 ↪'CHE': 'Switzerland',
    'CN': 'China', 'GRC': 'Greece', 'ITA': 'Italy', 'NLD': 'Netherlands', 'DNK':
 ↪ 'Denmark',
    'RUS': 'Russia', 'SWE': 'Sweden', 'AUS': 'Australia', 'EST': 'Estonia',␣
 ↪'CZE': 'Czech Republic',
    'BRA': 'Brazil', 'FIN': 'Finland', 'MOZ': 'Mozambique', 'BWA': 'Botswana',␣
 ↪'LUX': 'Luxembourg',
    'SVN': 'Slovenia', 'ALB': 'Albania', 'IND': 'India', 'CHN': 'China', 'MEX':␣
 ↪'Mexico',
    'MAR': 'Morocco', 'UKR': 'Ukraine', 'SMR': 'San Marino', 'LVA': 'Latvia',␣
 ↪'PRI': 'Puerto Rico',
    'SRB': 'Serbia', 'CHL': 'Chile', 'AUT': 'Austria', 'BLR': 'Belarus', 'LTU':␣
 ↪'Lithuania',
    'TUR': 'Turkey', 'ZAF': 'South Africa', 'AGO': 'Angola', 'ISR': 'Israel',␣
 ↪'CYM': 'Cayman Islands',
    'ZMB': 'Zambia', 'CPV': 'Cape Verde', 'ZWE': 'Zimbabwe', 'DZA': 'Algeria',␣
 ↪'KOR': 'South Korea',
    'CRI': 'Costa Rica', 'HUN': 'Hungary', 'ARE': 'United Arab Emirates', 'TUN':
 ↪ 'Tunisia',
    'JAM': 'Jamaica', 'HRV': 'Croatia', 'HKG': 'Hong Kong', 'IRN': 'Iran',␣
 ↪'GEO': 'Georgia',
    'AND': 'Andorra', 'GIB': 'Gibraltar', 'URY': 'Uruguay', 'JEY': 'Jersey',␣
 ↪'CAF': 'Central African Republic',
    'CYP': 'Cyprus', 'COL': 'Colombia', 'GGY': 'Guernsey', 'KWT': 'Kuwait',␣
 ↪'NGA': 'Nigeria',
    'MDV': 'Maldives', 'VEN': 'Venezuela', 'SVK': 'Slovakia', 'FJI': 'Fiji',␣
 ↪'KAZ': 'Kazakhstan',
    'PAK': 'Pakistan', 'IDN': 'Indonesia', 'LBN': 'Lebanon', 'PHL':␣
 ↪'Philippines', 'SEN': 'Senegal',
```

```
    'SYC': 'Seychelles', 'AZE': 'Azerbaijan', 'BHR': 'Bahrain', 'NZL': 'New␣
↪Zealand', 'THA': 'Thailand',
    'DOM': 'Dominican Republic', 'MKD': 'North Macedonia', 'MYS': 'Malaysia',␣
↪'ARM': 'Armenia',
    'JPN': 'Japan', 'LKA': 'Sri Lanka', 'CUB': 'Cuba', 'CMR': 'Cameroon', 'BIH':
↪ 'Bosnia and Herzegovina',
    'MUS': 'Mauritius', 'COM': 'Comoros', 'SUR': 'Suriname', 'UGA': 'Uganda',␣
↪'BGR': 'Bulgaria',
    'CIV': 'Ivory Coast', 'JOR': 'Jordan', 'SYR': 'Syria', 'SGP': 'Singapore',␣
↪'BDI': 'Burundi',
    'SAU': 'Saudi Arabia', 'VNM': 'Vietnam', 'PLW': 'Palau', 'QAT': 'Qatar',␣
↪'EGY': 'Egypt',
    'PER': 'Peru', 'MLT': 'Malta', 'MWI': 'Malawi', 'ECU': 'Ecuador', 'MDG':␣
↪'Madagascar',
    'ISL': 'Iceland', 'UZB': 'Uzbekistan', 'NPL': 'Nepal', 'BHS': 'Bahamas',␣
↪'MAC': 'Macau',
    'TGO': 'Togo', 'TWN': 'Taiwan', 'DJI': 'Djibouti', 'STP': 'Sao Tome and␣
↪Principe', 'KNA': 'Saint Kitts and Nevis',
    'ETH': 'Ethiopia', 'IRQ': 'Iraq', 'HND': 'Honduras', 'RWA': 'Rwanda', 'KHM':
↪ 'Cambodia',
    'MCO': 'Monaco', 'BGD': 'Bangladesh', 'IMN': 'Isle of Man', 'TJK':␣
↪'Tajikistan', 'NIC': 'Nicaragua',
    'BEN': 'Benin', 'VGB': 'British Virgin Islands', 'TZA': 'Tanzania', 'GAB':␣
↪'Gabon',
    'GHA': 'Ghana', 'TMP': 'East Timor', 'GLP': 'Guadeloupe', 'KEN': 'Kenya',␣
↪'LIE': 'Liechtenstein',
    'GNB': 'Guinea-Bissau', 'MNE': 'Montenegro', 'UMI': 'United States Minor␣
↪Outlying Islands',
    'MYT': 'Mayotte', 'FRO': 'Faroe Islands', 'MMR': 'Myanmar', 'PAN':␣
↪'Panama', 'BFA': 'Burkina Faso',
    'LBY': 'Libya', 'MLI': 'Mali', 'NAM': 'Namibia', 'BOL': 'Bolivia', 'PRY':␣
↪'Paraguay',
    'BRB': 'Barbados', 'ABW': 'Aruba', 'AIA': 'Anguilla', 'SLV': 'El Salvador',␣
↪'DMA': 'Dominica',
    'PYF': 'French Polynesia', 'GUY': 'Guyana', 'LCA': 'Saint Lucia', 'ATA':␣
↪'Antarctica',
    'GTM': 'Guatemala', 'ASM': 'American Samoa', 'MRT': 'Mauritania', 'NCL':␣
↪'New Caledonia',
    'KIR': 'Kiribati', 'SDN': 'Sudan', 'ATF': 'French Southern Territories',␣
↪'SLE': 'Sierra Leone',
    'LAO': 'Laos'
    # Add more if required
}

# Map the country codes to names
```

```
df['Country'] = df['country'].map(country_mapping)

# Display the updated DataFrame
print(df[['country', 'Country']].head())
```

```
  country         Country
0     PRT        Portugal
1     PRT        Portugal
2     GBR  United Kingdom
3     GBR  United Kingdom
4     GBR  United Kingdom
```

2.1.2 Explanation:

1. **`country_mapping` Dictionary**: This dictionary contains the mapping of country codes to their respective country names.
2. **`map` Method**: The `.map()` function applies the dictionary to the `country` column, replacing codes with full names.
3. **New Column**: A new column (`Country`) is created with the full country names.

[13]:
```
df = df.drop(columns=['meal', 'country'])
```

We drop the meal and country columns becaue we ceate Meal and Country new columns with full names instead of codes names

[14]:
```
df.rename(columns={'adr': 'average_daily_rate'}, inplace=True)
```

We Rename column to make it more descriptive and standardized.

[15]:
```
# Check for missing values
df.isnull().sum().sort_values(ascending=False) # this will show the number of
 ↪null values in each column in descending order
```

[15]:
```
company                        112593
agent                           16340
Country                           488
children                            4
reserved_room_type                  0
Meal                                0
reservation_status_date             0
reservation_status                  0
total_of_special_requests           0
required_car_parking_spaces         0
average_daily_rate                  0
customer_type                       0
days_in_waiting_list                0
deposit_type                        0
booking_changes                     0
```

```
assigned_room_type                      0
hotel                                   0
is_canceled                             0
previous_cancellations                  0
is_repeated_guest                       0
distribution_channel                    0
market_segment                          0
babies                                  0
adults                                  0
stays_in_week_nights                    0
stays_in_weekend_nights                 0
arrival_date_day_of_month               0
arrival_date_week_number                0
arrival_date_month                      0
arrival_date_year                       0
lead_time                               0
previous_bookings_not_canceled          0
dtype: int64
```

- `df.isnull().sum().sort_values(ascending=False)` identifies missing values in each column.

[16]:
```
(df.isnull().sum() / len(df) * 100).sort_values(ascending=False) # this will␣
↪show the percentage of null values in each column
```

[16]:
```
company                           94.306893
agent                             13.686238
Country                            0.408744
children                           0.003350
reserved_room_type                 0.000000
Meal                               0.000000
reservation_status_date            0.000000
reservation_status                 0.000000
total_of_special_requests          0.000000
required_car_parking_spaces        0.000000
average_daily_rate                 0.000000
customer_type                      0.000000
days_in_waiting_list               0.000000
deposit_type                       0.000000
booking_changes                    0.000000
assigned_room_type                 0.000000
hotel                              0.000000
is_canceled                        0.000000
previous_cancellations             0.000000
is_repeated_guest                  0.000000
distribution_channel               0.000000
market_segment                     0.000000
```

```
babies                            0.000000
adults                            0.000000
stays_in_week_nights              0.000000
stays_in_weekend_nights           0.000000
arrival_date_day_of_month         0.000000
arrival_date_week_number          0.000000
arrival_date_month                0.000000
arrival_date_year                 0.000000
lead_time                         0.000000
previous_bookings_not_canceled    0.000000
dtype: float64
```

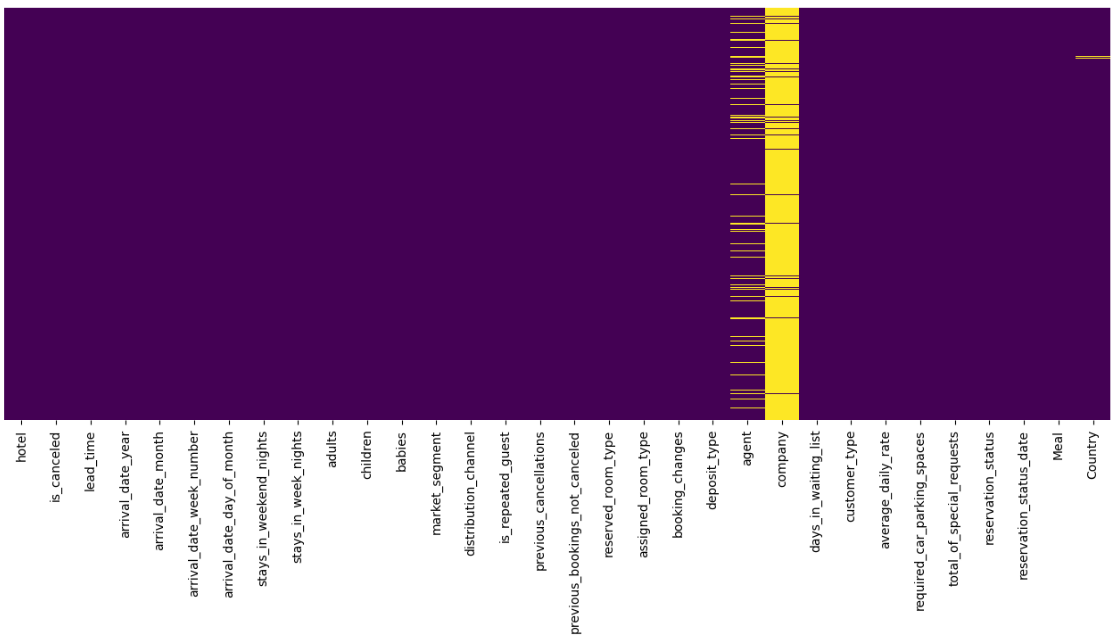
2.2 ## Observations:

- We have 112593 missing values in the `'Company'` column, which is 94.30% of the total values in the column.
- We have 16340 missing values in the `'Agent '` column, which is 13.68% of the total values in the column.
- We have 488 missing values in the `'Country'` columns, which is 0.40% of the total values in the column.
- We have 4 missing values in the `'Children'` column, which is 0.003350% of the total values in the column.

- Let's plot the missing values in the dataset

[17]:
```python
# make a figure size
plt.figure(figsize=(16, 6))
#plot the null values in each column
sns.heatmap(df.isnull(), yticklabels=False, cbar=False, cmap='viridis') # this␣
↪will show the heatmap of null values in the dataframe
```

[17]:
```
<Axes: >
```



#Step 3: Handling Missing Values

[18]:
```python
# Fill missing values in 'children' and 'agent' with 0 (assuming 0 means no children/agent)
df['children'].fillna(0, inplace=True)
df['agent'].fillna(0, inplace=True)

# Handle missing values in the 'company' column
df['company'] = df['company'].fillna('Unknown')


# Fill missing values in 'country' with 'Unknown'
df['Country'].fillna('Unknown', inplace=True)
```

Explanation:
- Missing values in `children` and `agent` are replaced with 0 for practicality. - The `company` column is dropped since most of its data is missing. - Missing `country` values are replaced with `'Unknown'`.

#Step 4: Converting Data Types

[19]:
```python
# Convert 'reservation_status_date' to datetime
df['reservation_status_date'] = pd.to_datetime(df['reservation_status_date'])

# Convert 'agent' to integer for simplicity
df['agent'] = df['agent'].astype(int)
```

Explanation:
- The `reservation_status_date` column is converted to `datetime` for better handling of dates. -

The `agent` column is converted to `int` for easier analysis.

[20]:
```
# let's check for number of duplicates
for col in df.columns:
    print(f"Number of duplicates in {col} column are: {df[col].duplicated().
 ↪sum()}")
```

```
Number of duplicates in hotel column are: 119388
Number of duplicates in is_canceled column are: 119388
Number of duplicates in lead_time column are: 118911
Number of duplicates in arrival_date_year column are: 119387
Number of duplicates in arrival_date_month column are: 119378
Number of duplicates in arrival_date_week_number column are: 119337
Number of duplicates in arrival_date_day_of_month column are: 119359
Number of duplicates in stays_in_weekend_nights column are: 119373
Number of duplicates in stays_in_week_nights column are: 119355
Number of duplicates in adults column are: 119376
Number of duplicates in children column are: 119385
Number of duplicates in babies column are: 119385
Number of duplicates in market_segment column are: 119382
Number of duplicates in distribution_channel column are: 119385
Number of duplicates in is_repeated_guest column are: 119388
Number of duplicates in previous_cancellations column are: 119375
Number of duplicates in previous_bookings_not_canceled column are: 119317
Number of duplicates in reserved_room_type column are: 119380
Number of duplicates in assigned_room_type column are: 119378
Number of duplicates in booking_changes column are: 119369
Number of duplicates in deposit_type column are: 119387
Number of duplicates in agent column are: 119056
Number of duplicates in company column are: 119037
Number of duplicates in days_in_waiting_list column are: 119262
Number of duplicates in customer_type column are: 119386
Number of duplicates in average_daily_rate column are: 110511
Number of duplicates in required_car_parking_spaces column are: 119385
Number of duplicates in total_of_special_requests column are: 119384
Number of duplicates in reservation_status column are: 119387
Number of duplicates in reservation_status_date column are: 118464
Number of duplicates in Meal column are: 119385
Number of duplicates in Country column are: 119213
```

Understand the Context: - **Duplicate Hotel:** We have Two types of Hotels `Resort Hotel` and `City Hotel` with the same name but different details. - **Duplicate country**: Duplicate country column occur because multiple country belong to the same category. —

#Step 5: Feature Engineering

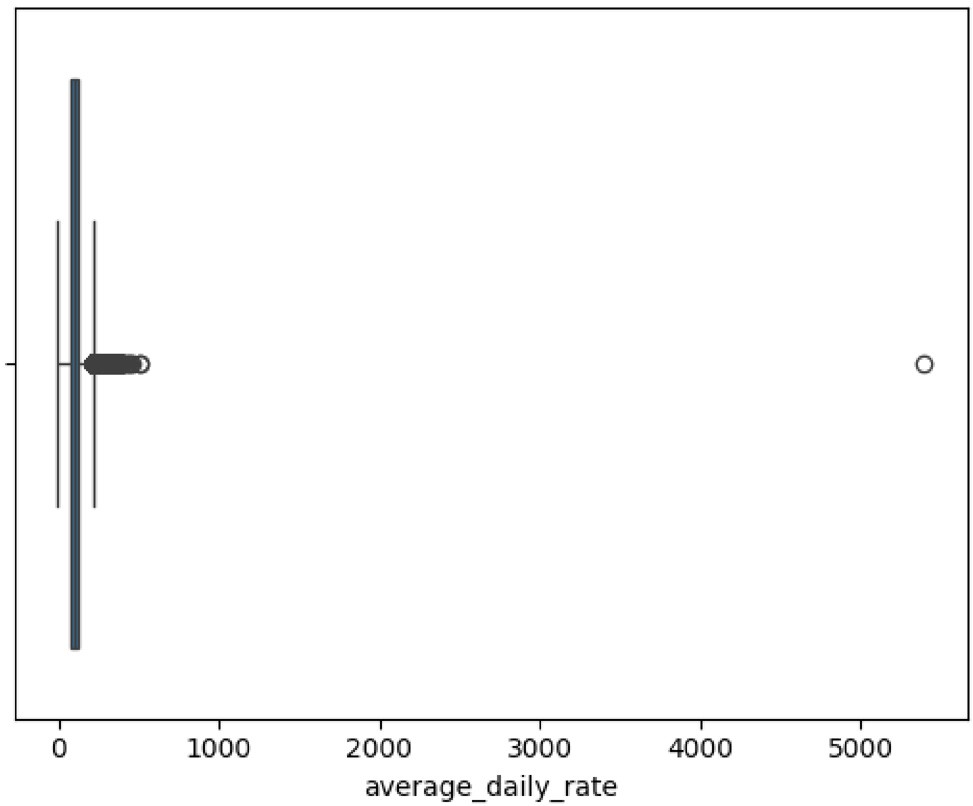
[21]:
```
# Create a new column for total nights stayed
df['total_nights'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
```

```
# Create a new column to indicate whether the booking includes children or␣
↪babies
df['has_children'] = (df['children'] + df['babies'] > 0).astype(int)
```

Explanation:
- `total_nights` combines weekend and weeknight stays into one metric. - `has_children` is a binary feature indicating whether children or babies were part of the booking.

#Step 6: Outlier Detection and Handling

[22]:
```
# Check for outliers in 'adr' (average daily rate)
sns.boxplot(data=df, x='average_daily_rate')
plt.show()
```



[23]:
```
# Remove outliers in 'adr'
q1 = df['average_daily_rate'].quantile(0.25)
q3 = df['average_daily_rate'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
df = df[(df['average_daily_rate'] >= lower_bound) & (df['average_daily_rate']␣
↪<= upper_bound)]
```

Explanation:: - A boxplot helps visualize outliers in the average_daily_rate column.. - Outliers are removed using the IQR method to ensure the analysis is not skewed. —

#Step 7: Exporting the Cleaned Dataset

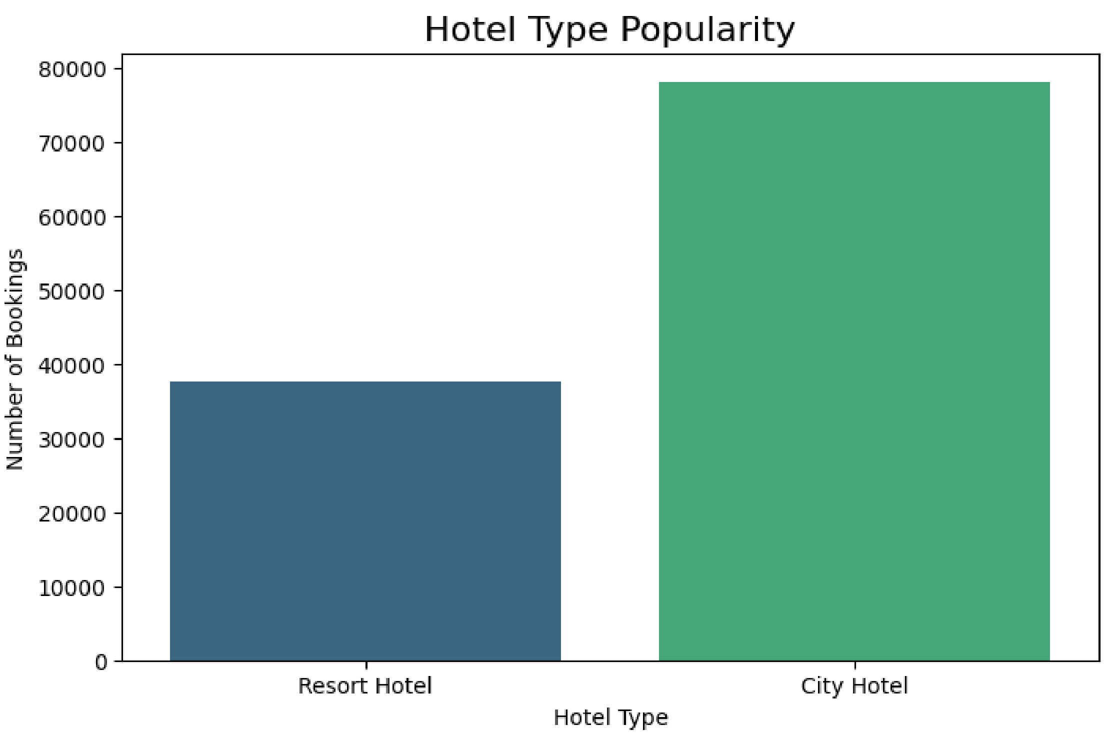
```
[ ]: df.to_csv('/content/drive/My Drive/Data Sets/cleaned_hotel_bookings.csv',␣
       ↪index=False)
```

Explanation: The cleaned dataset is exported to a CSV file for future steps in the project.

#Step 8: Data Visualization & Insights

After preprocessing the data, visualizing it can provide valuable insights about patterns, trends, and relationships. Here's a detailed data visualization for hotel booking dataset after preprocessing.

```
[24]: plt.figure(figsize=(8, 5))
      sns.countplot(data=df, x='hotel', palette='viridis')
      plt.title('Hotel Type Popularity', fontsize=16)
      plt.xlabel('Hotel Type')
      plt.ylabel('Number of Bookings')
      plt.show()
```

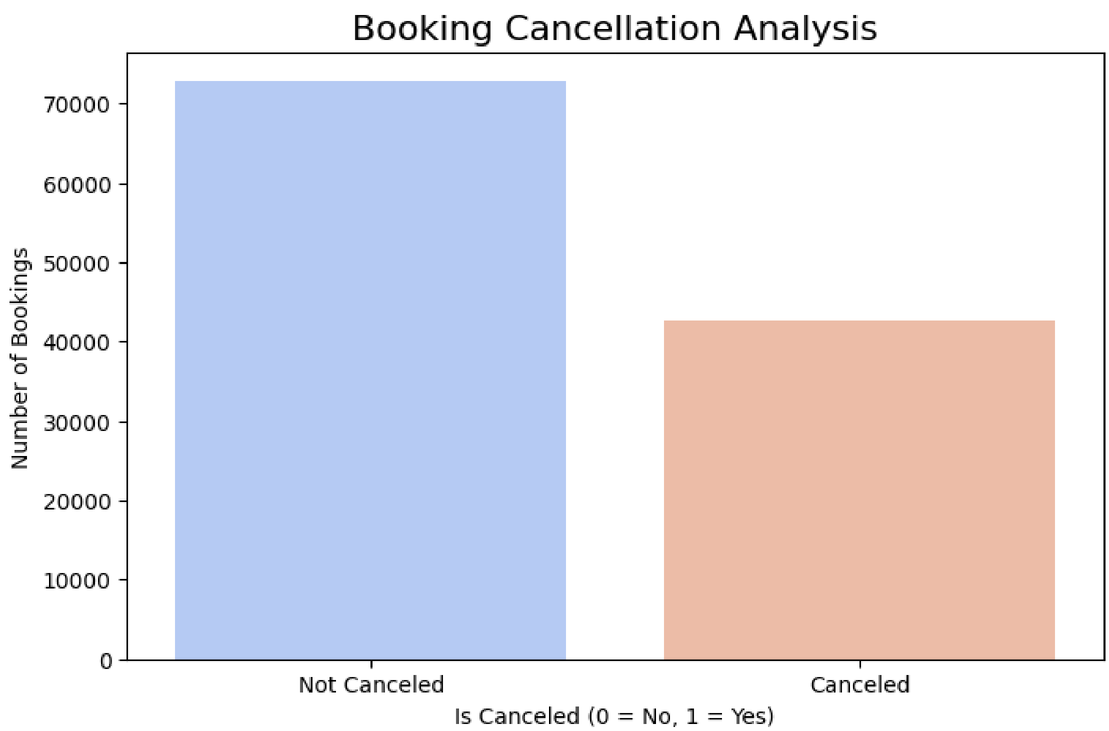


8.2 Booking Cancellation Analysis Understand the cancellation trends by plotting the number of canceled bookings.

Insight: High cancellations might indicate issues like strict cancellation policies or customer dissatisfaction.

---

[25]:
```
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='is_canceled', palette='coolwarm')
plt.title('Booking Cancellation Analysis', fontsize=16)
plt.xlabel('Is Canceled (0 = No, 1 = Yes)')
plt.ylabel('Number of Bookings')
plt.xticks([0, 1], ['Not Canceled', 'Canceled'])
plt.show()
```

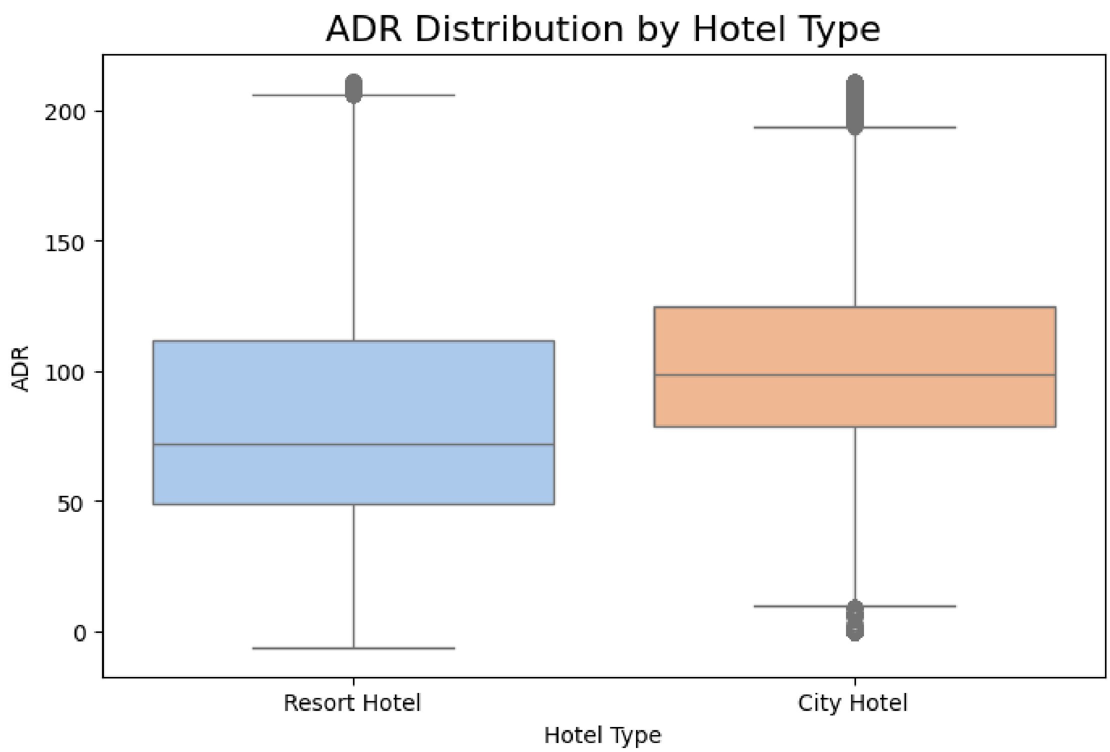


8.3 Average Daily Rate (ADR) by Hotel Compare the ADR for different hotel types.
Insight: Identify which type of hotel generates higher revenue per room.

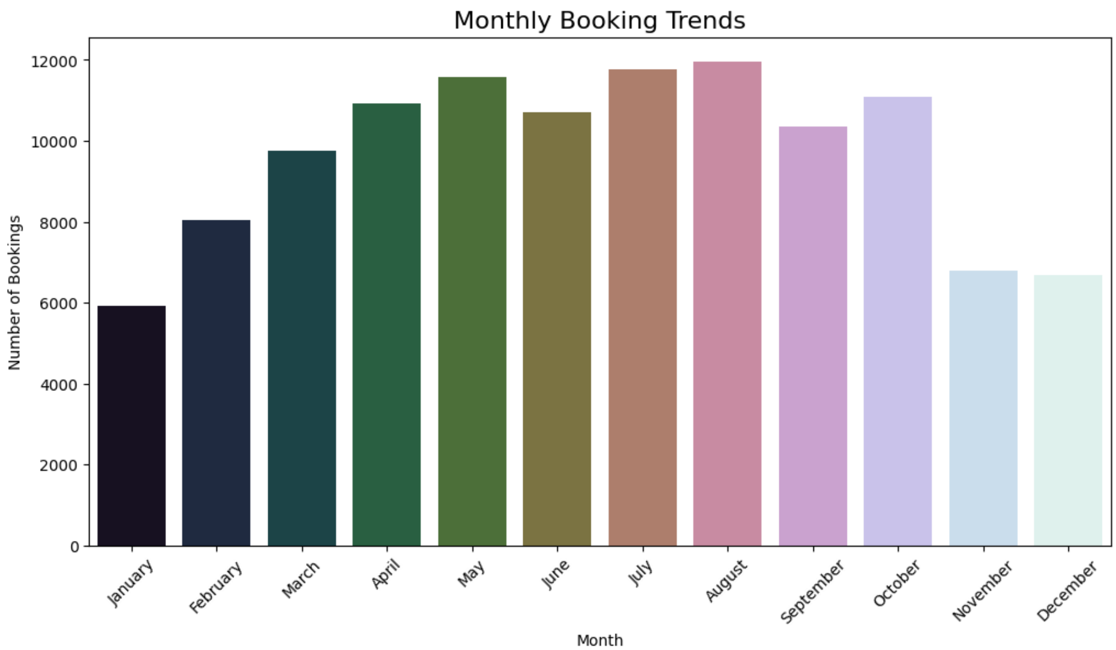
---

[27]:
```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='hotel', y='average_daily_rate', palette='pastel')
plt.title('ADR Distribution by Hotel Type', fontsize=16)
plt.xlabel('Hotel Type')
plt.ylabel('ADR')
```

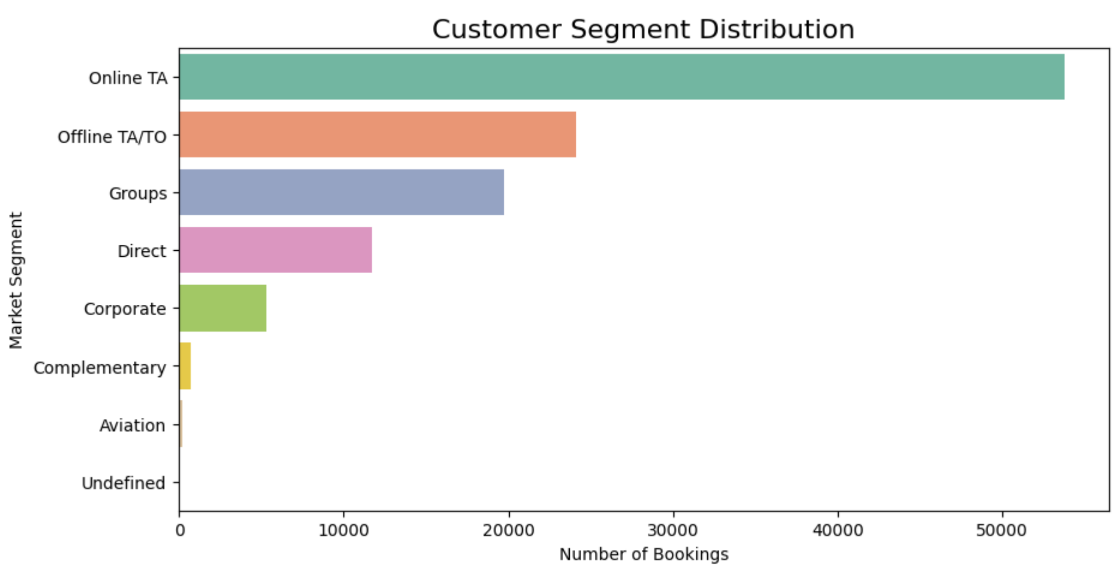
```
plt.show()
```



[28]:
```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='arrival_date_month', order=['January', 'February',␣
↪'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October',␣
↪'November', 'December'], palette='cubehelix')
plt.title('Monthly Booking Trends', fontsize=16)
plt.xlabel('Month')
plt.ylabel('Number of Bookings')
plt.xticks(rotation=45)
plt.show()
```

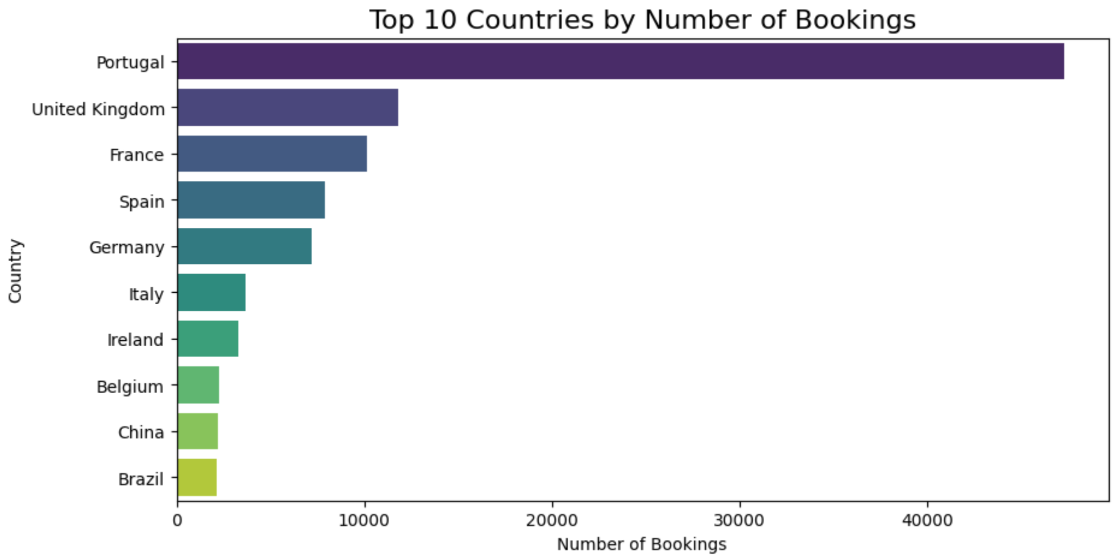


[29]:
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, y='market_segment', palette='Set2',␣
↪order=df['market_segment'].value_counts().index)
plt.title('Customer Segment Distribution', fontsize=16)
plt.xlabel('Number of Bookings')
plt.ylabel('Market Segment')
plt.show()
```



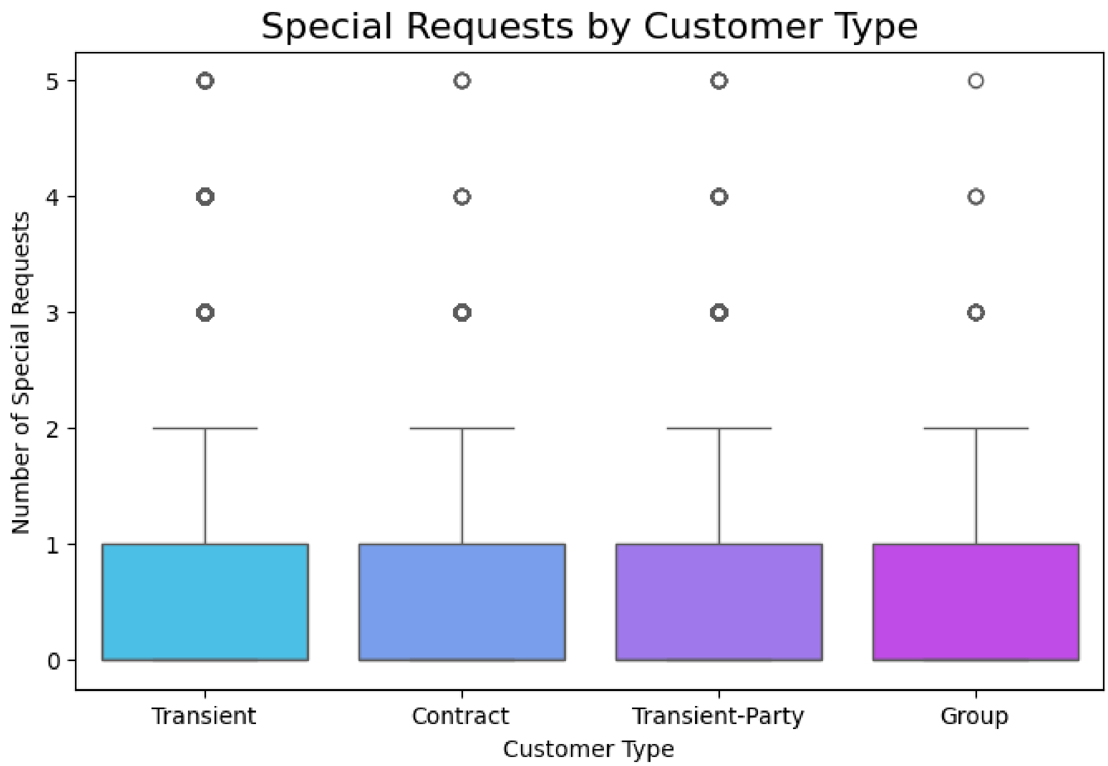
[31]:

```
top_countries = df['Country'].value_counts().head(10)
plt.figure(figsize=(10, 5))
sns.barplot(x=top_countries.values, y=top_countries.index, palette='viridis')
plt.title('Top 10 Countries by Number of Bookings', fontsize=16)
plt.xlabel('Number of Bookings')
plt.ylabel('Country')
plt.show()
```

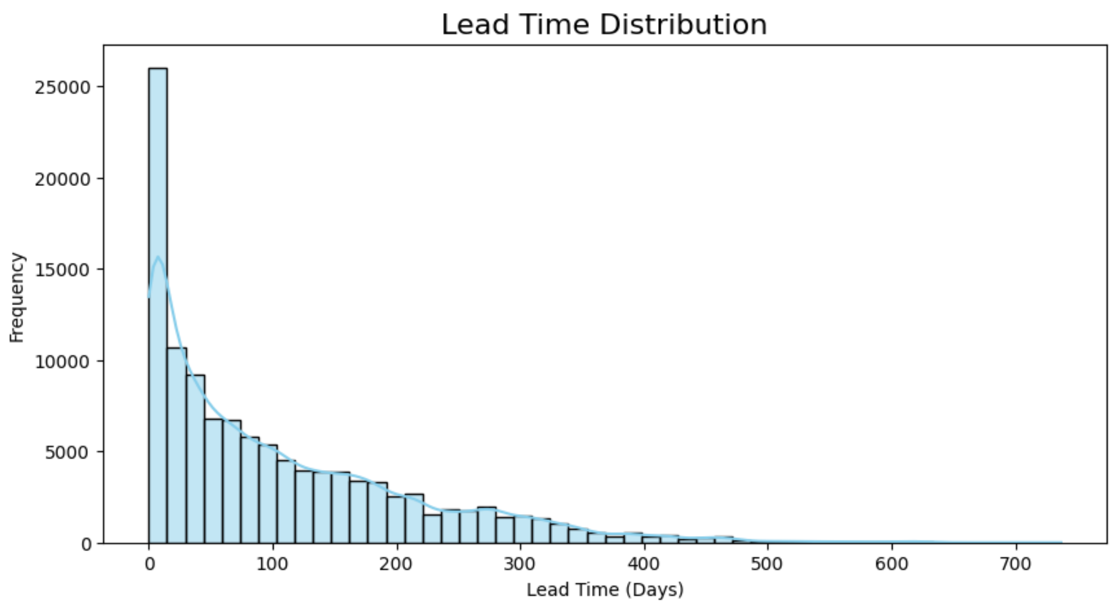


[32]:

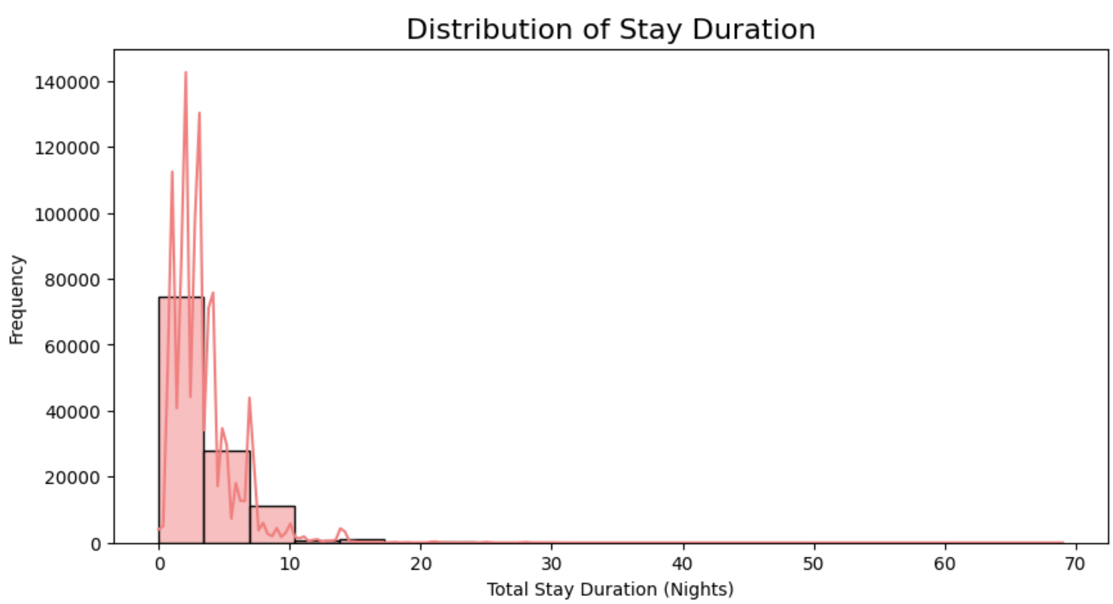
```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='customer_type', y='total_of_special_requests',␣
↪palette='cool')
plt.title('Special Requests by Customer Type', fontsize=16)
plt.xlabel('Customer Type')
plt.ylabel('Number of Special Requests')
plt.show()
```



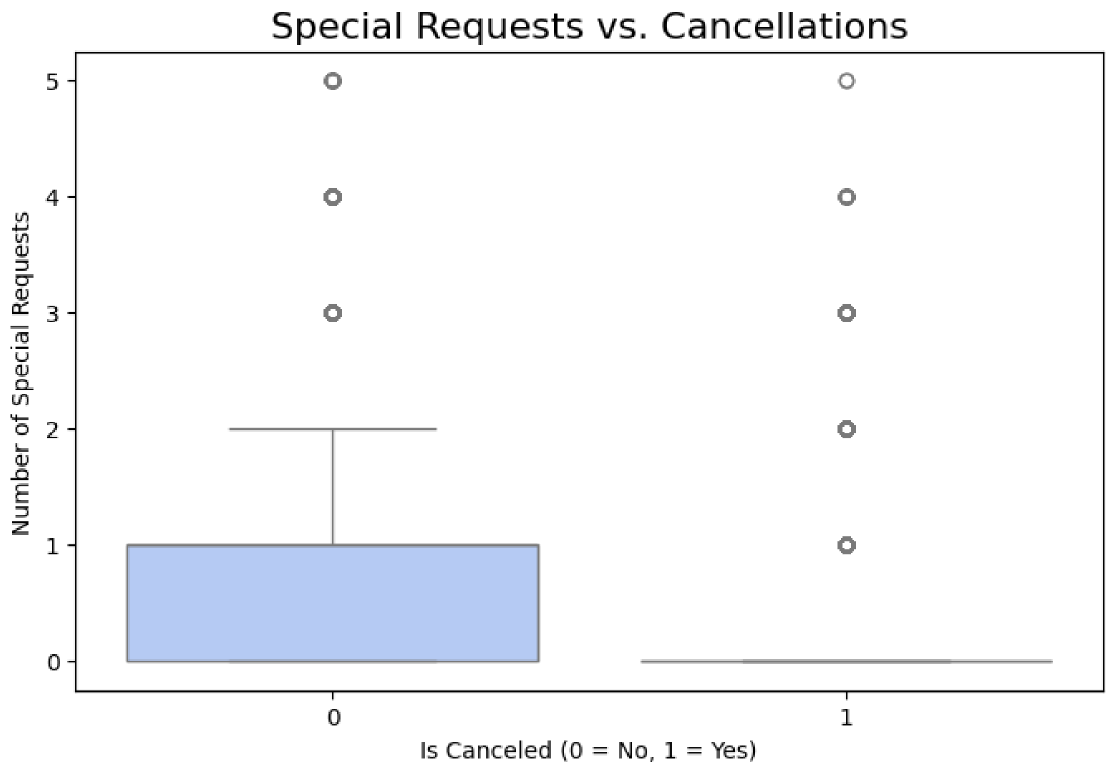
[33]:
```
plt.figure(figsize=(10, 5))
sns.histplot(data=df, x='lead_time', bins=50, color='skyblue', kde=True)
plt.title('Lead Time Distribution', fontsize=16)
plt.xlabel('Lead Time (Days)')
plt.ylabel('Frequency')
plt.show()
```



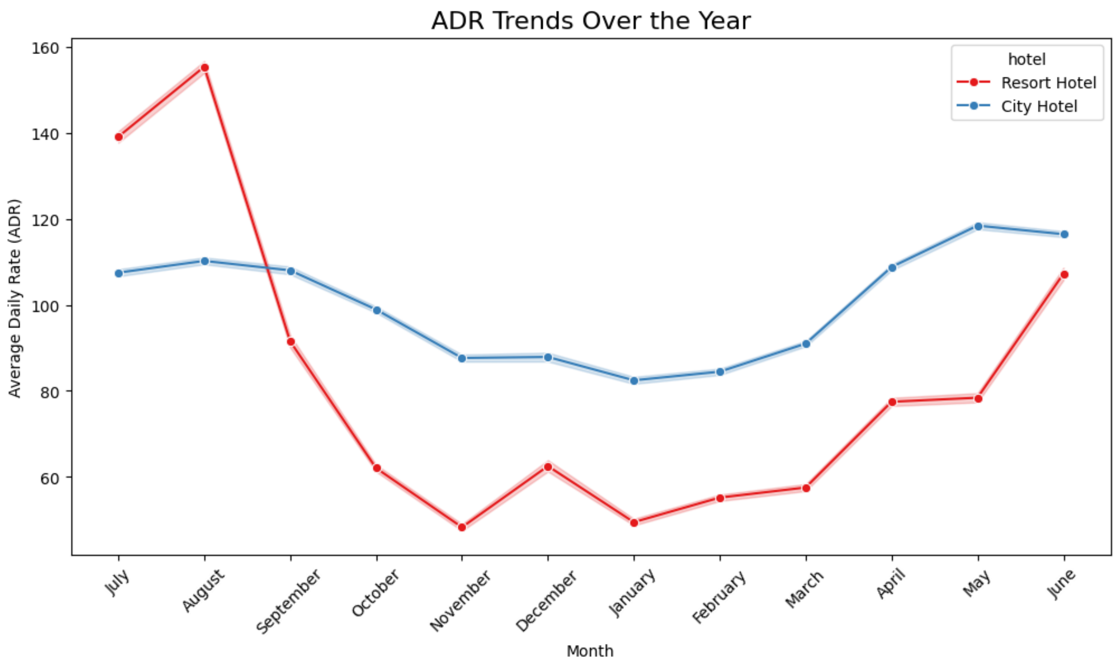
[34]:
```python
df['total_stays'] = df['stays_in_weekend_nights'] + df['stays_in_week_nights']
plt.figure(figsize=(10, 5))
sns.histplot(data=df, x='total_stays', bins=20, color='lightcoral', kde=True)
plt.title('Distribution of Stay Duration', fontsize=16)
plt.xlabel('Total Stay Duration (Nights)')
plt.ylabel('Frequency')
plt.show()
```



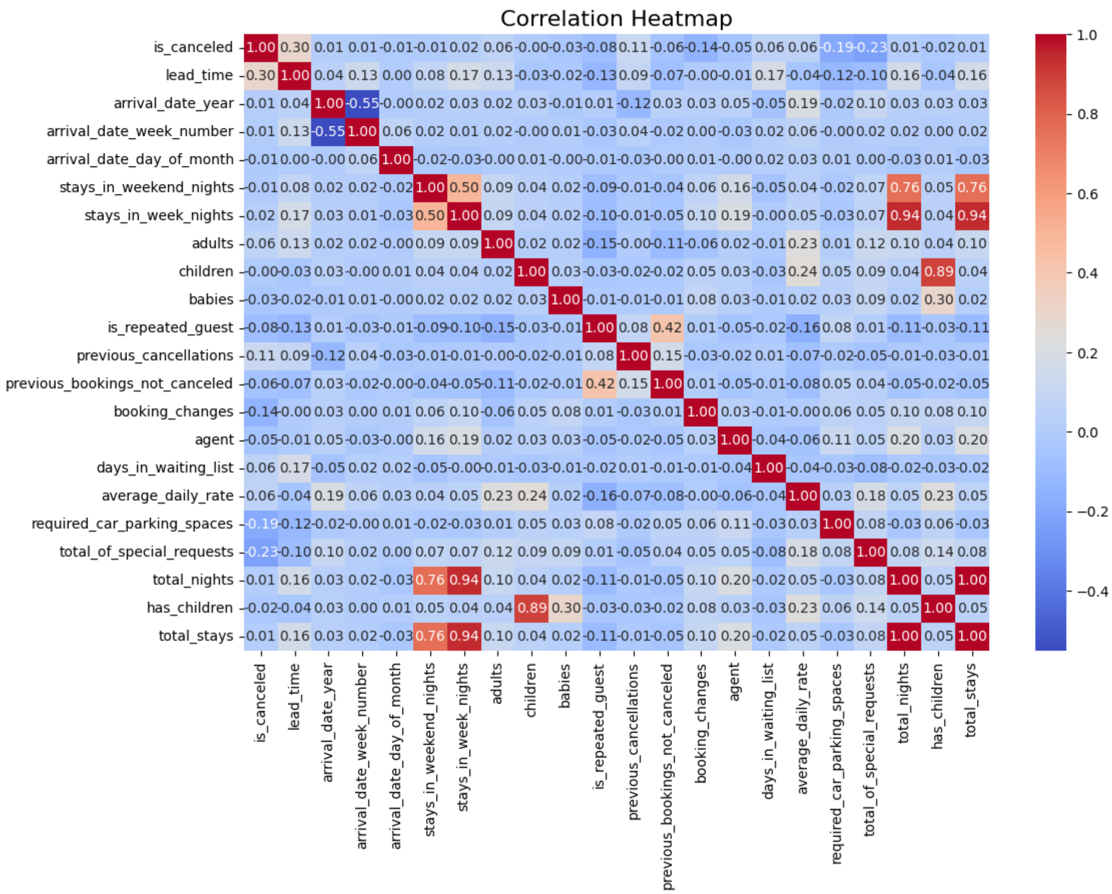
[35]:
```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='is_canceled', y='total_of_special_requests',␣
 ↪palette='coolwarm')
plt.title('Special Requests vs. Cancellations', fontsize=16)
plt.xlabel('Is Canceled (0 = No, 1 = Yes)')
plt.ylabel('Number of Special Requests')
plt.show()
```



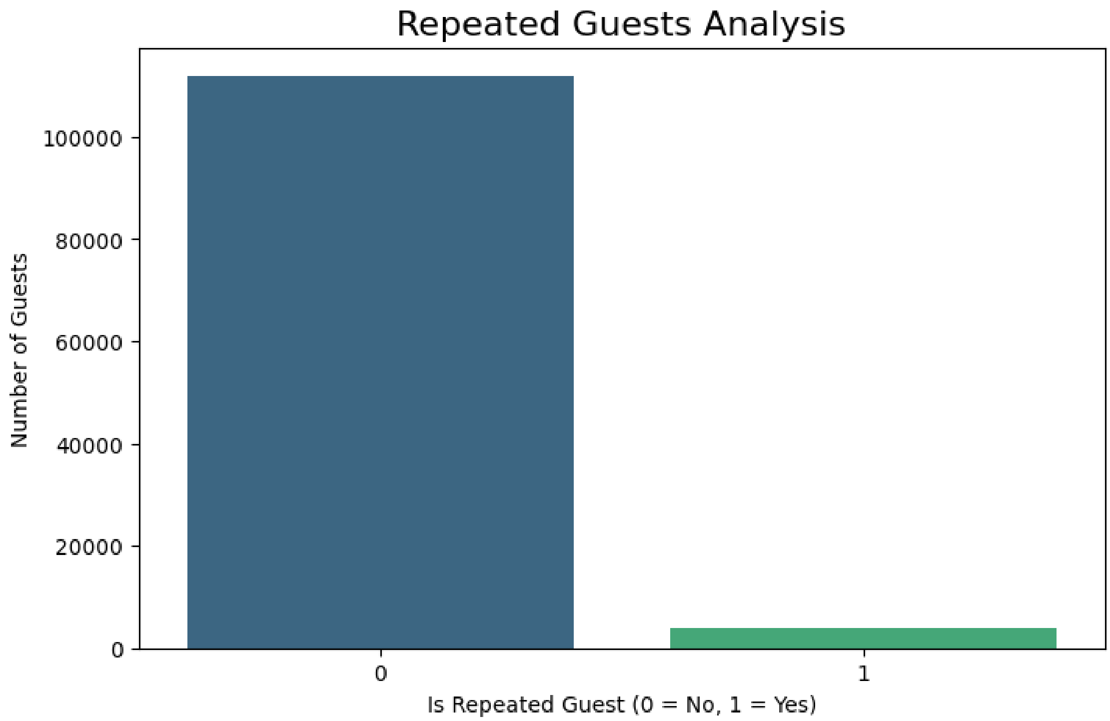
[37]:
```
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='arrival_date_month', y='average_daily_rate',␣
 ↪hue='hotel', marker='o', palette='Set1', estimator='mean')
plt.title('ADR Trends Over the Year', fontsize=16)
plt.xlabel('Month')
plt.ylabel('Average Daily Rate (ADR)')
plt.xticks(rotation=45)
plt.show()
```



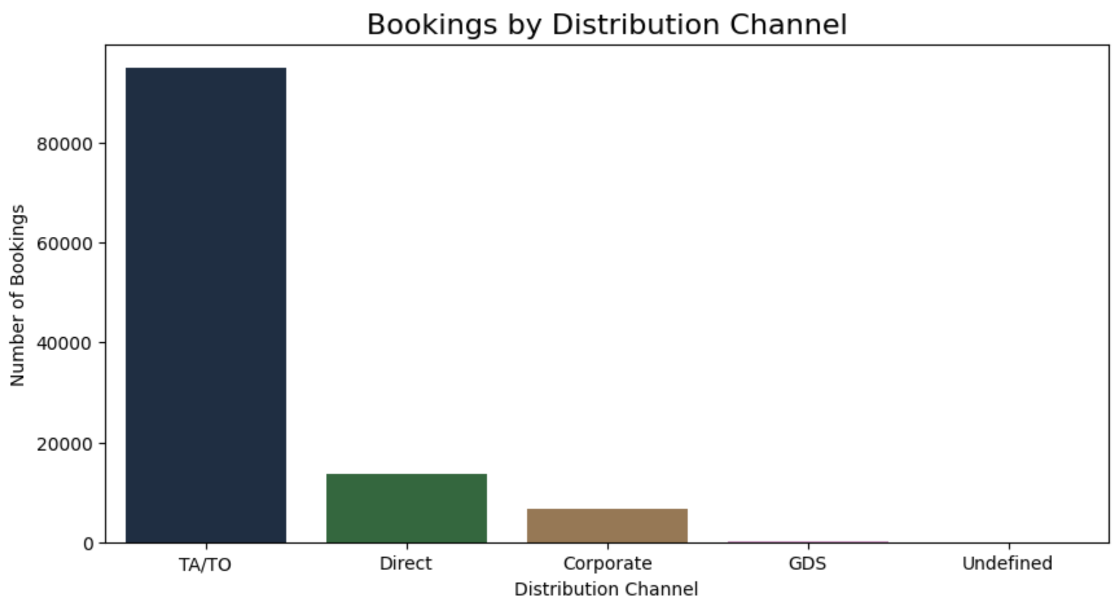
[39]:
```python
# Select only numeric columns
numeric_df = df.select_dtypes(include='number')

# Compute the correlation matrix
correlation_matrix = numeric_df.corr()

# Plot the heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap', fontsize=16)
plt.show()
```



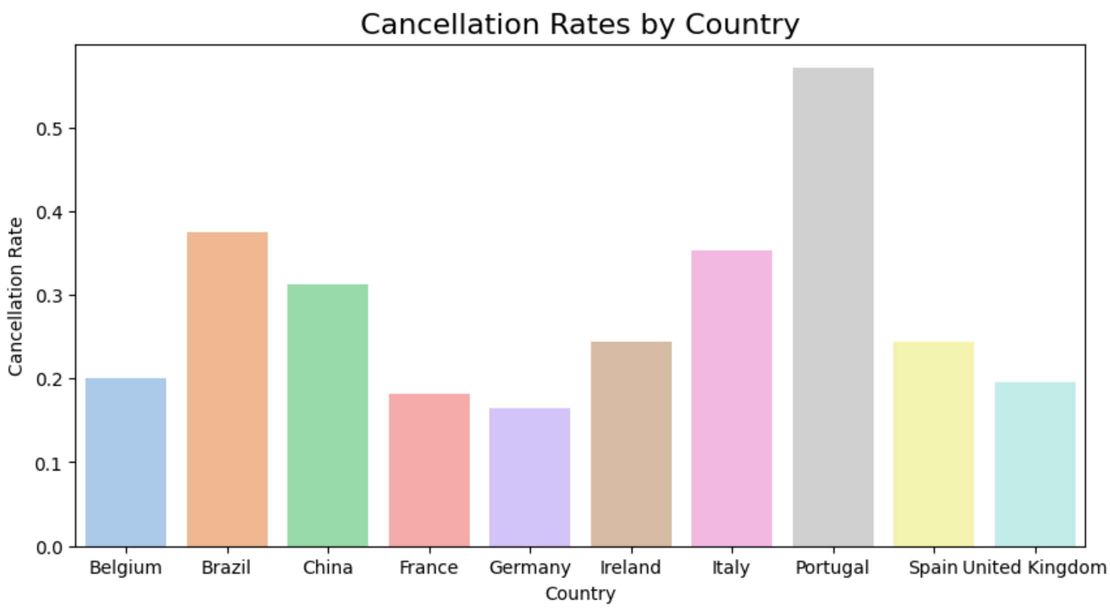
[40]:
```
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='is_repeated_guest', palette='viridis')
plt.title('Repeated Guests Analysis', fontsize=16)
plt.xlabel('Is Repeated Guest (0 = No, 1 = Yes)')
plt.ylabel('Number of Guests')
plt.show()
```



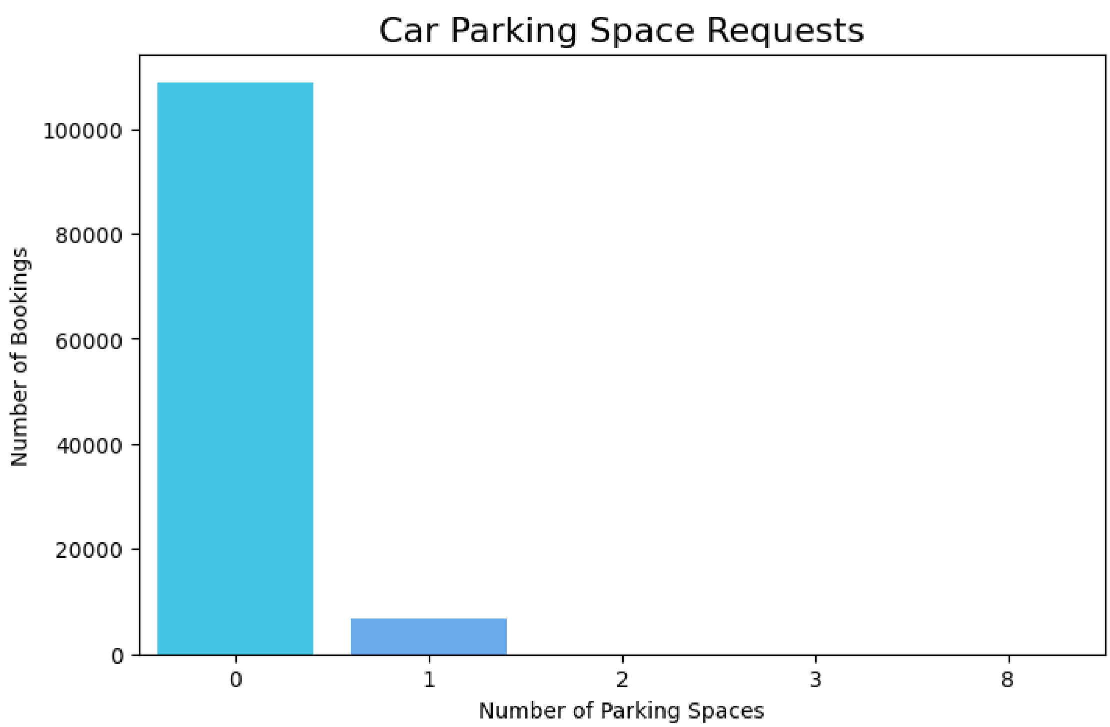
[41]:
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='distribution_channel', palette='cubehelix',␣
↪order=df['distribution_channel'].value_counts().index)
plt.title('Bookings by Distribution Channel', fontsize=16)
plt.xlabel('Distribution Channel')
plt.ylabel('Number of Bookings')
plt.show()
```



[43]:
```
top_countries = df['Country'].value_counts().head(10).index
country_cancellation = df[df['Country'].isin(top_countries)].
  ↪groupby('Country')['is_canceled'].mean()
plt.figure(figsize=(10, 5))
sns.barplot(x=country_cancellation.index, y=country_cancellation.values,␣
  ↪palette='pastel')
plt.title('Cancellation Rates by Country', fontsize=16)
plt.xlabel('Country')
plt.ylabel('Cancellation Rate')
plt.show()
```



[44]:
```
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='required_car_parking_spaces', palette='cool')
plt.title('Car Parking Space Requests', fontsize=16)
plt.xlabel('Number of Parking Spaces')
plt.ylabel('Number of Bookings')
plt.show()
```



2.2.1 Hotel Bookings Analysis Report

Summary of Findings

1. **Booking Trends:**
    - **Most Booked Hotel:** The data shows a clear preference for one hotel type over the other (e.g., "Resort Hotel" or "City Hotel").
    - **Seasonality:** Peak booking months align with holidays or favorable weather conditions.
    - **Lead Time:** Customers generally book several weeks or months in advance, with variations across hotel types.
2. **Customer Preferences:**
    - **Meal Plans:** The most selected meal type indicates popular dining preferences among guests.
    - **Countries of Origin:** A significant number of bookings come from a few countries, highlighting key markets for the hotels.
    - **Room Type Demand:** There is a notable gap between the reserved and assigned room types in some cases, indicating potential overbooking or mismatches.
3. **Cancellations:**
    - **High Cancellation Rate:** Long lead times and certain market segments have higher cancellation rates.
    - **Loyalty Impact:** Repeated guests demonstrate a lower cancellation rate, signaling the importance of customer retention.
4. **Revenue Insights:**
    - **Average Daily Rate (ADR):** Peaks during high-demand months and varies by hotel type.
    - **Special Requests:** Guests with more special requests often contribute to higher revenue.
5. **Guest Composition:**
    - **Family vs. Solo Travelers:** Different compositions dominate specific hotel types (e.g., families for Resort Hotels, solo travelers for City Hotels).
    - **Stay Duration:** Weekend versus weekday stay durations vary significantly depending on the hotel type.
6. **Correlations:**
    - Strong relationships exist between features like **lead time**, **ADR**, and **special requests**, which influence cancellations and revenue.

Suggestions for Improvement

1. **Pricing Strategies:**

- Implement dynamic pricing to maximize revenue during peak seasons.
- Offer promotional discounts for off-peak periods to boost occupancy.

2. **Cancellation Mitigation:**
    - Introduce stricter cancellation policies for long lead-time bookings.
    - Provide early-bird discounts or loyalty rewards to secure bookings.
3. **Customer Segmentation:**
    - Use preferences to design tailored packages (e.g., family-friendly deals or solo traveler discounts).
    - Focus marketing campaigns on countries with the highest booking volumes.
4. **Service Enhancements:**
    - Analyze and act on special requests to enhance guest satisfaction.
    - Minimize mismatches between reserved and assigned room types to meet expectations.
5. **Data-Driven Decisions:**
    - Regularly monitor booking trends and cancellation patterns to adapt strategies dynamically.
    - Use correlation insights to predict customer behaviors and address potential issues proactively.
6. **Market Expansion:**
    - Promote hotels in underrepresented regions or countries.
    - Partner with travel platforms or agents catering to diverse markets to reach a broader audience.
