In [3]:
```
## Load Packages
```

In [2]:
```
import os
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
```

Understanding Aspects of Data

adr : average daily rate

adults : number of adults

agent : ID of the travel agency that made the booking

arrival_date_day_of_month : Day of the month of the arrival date

arrival_date_month : Month of arrival date with 12 categories: “January” to “December”

arrival_date_week_number : Week number of the arrival date

arrival_date_year : Year of arrival date

assigned_room_type : Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons.

babies : Number of babies

booking_changes : Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation.

children : Number of children.

company : ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons.

country : Country of origin. Categories are represented in the ISO 3155–3:2013 format

customer_type : Type of booking, assuming one of four categories :

Contract - when the booking has an allotment or other type of contract associated to it;

Group – when the booking is associated to a group;

Transient – when the booking is not part of a group or contract, and is not associated to other transient booking;

Transient-party – when the booking is transient, but is associated to at least other transient booking

days_in_waiting_list : Number of days the booking was in the waiting list before it was confirmed to the customer

deposit_type : Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories:

No Deposit – no deposit was made.

Non Refund – a deposit was made in the value of the total stay cost.

Refundable – a deposit was made with a value under the total cost of stay.

distribution_channel : Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”.

is_canceled : Value indicating if the booking was canceled (1) or not (0).

is_repeated_quests : Value indicating if the booking name was from a repeated guest (1) or not (0).

lead_time : Number of days that elapsed between the entering date of the booking into the PMS and the arrival date.

market_segment : Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”.

meal : Type of meal booked. Categories are presented in standard hospitality meal packages:

Undefined/SC – no meal package BB – Bed & Breakfast HB – Half board (breakfast and one other meal – usually dinner) FB – Full board (breakfast, lunch and dinner)

previous_bookings_not_canceled : Number of previous bookings not cancelled by the customer prior to the current booking.

previous_cancellations : Number of previous bookings that were cancelled by the customer prior to the current booking.

required_car_parking_spaces : Number of car parking spaces required by the customer.

reservation_status : Reservation last status, assuming one of three categories:

Canceled – booking was canceled by the customer Check-Out – customer has checked in but already departed No-Show – customer did not check-in and did inform the hotel of the reason why

reservation_status_date : Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel.

reserved_room_type : Code of room type reserved. Code is presented instead of designation for anonymity reasons.

stays_in_weekend_nights : Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel.

stays_in_week_nights : Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel.

total_of_special_requests : Number of special requests made by the customer (e.g. twin bed or high floor).

Read and Understand the Data

In [3]:
```
hotel_data = pd.read_csv('/Users/doyin/documents/Data Projects/hotel_bookings.csv')
hotel_data.head(5)  # check first 5 rows of data
```

Out[3]:

	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	...	deposit_type	agent
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	No Deposit	NaN
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	No Deposit	NaN
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	No Deposit	NaN
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	No Deposit	304.0
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	No Deposit	240.0

5 rows × 32 columns

In [4]:
```
hotel_data.dtypes # understanding the format of the data
```

Out[4]:
```
hotel                              object
is_canceled                         int64
lead_time                           int64
arrival_date_year                   int64
arrival_date_month                 object
arrival_date_week_number            int64
arrival_date_day_of_month           int64
stays_in_weekend_nights             int64
stays_in_week_nights                int64
adults                              int64
children                          float64
babies                              int64
meal                               object
country                            object
market_segment                     object
distribution_channel               object
is_repeated_guest                   int64
previous_cancellations              int64
previous_bookings_not_canceled      int64
reserved_room_type                 object
assigned_room_type                 object
booking_changes                     int64
deposit_type                       object
agent                             float64
company                           float64
days_in_waiting_list                int64
customer_type                      object
adr                               float64
required_car_parking_spaces         int64
total_of_special_requests           int64
reservation_status                 object
reservation_status_date            object
dtype: object
```

DATA CLEANING

Deciding on Variables

In [10]:
```
# deleting agent and company column

del hotel_data["agent"]
del hotel_data["company"]
```

In [5]:
```
# looking further into unsure columns to decide if data is useful
hotel_data["market_segment"].describe()
```

Out[5]:
```
count        119390
unique            8
top       Online TA
freq          56477
Name: market_segment, dtype: object
```

In [6]:
```
hotel_data["market_segment"].unique()
```

Out[6]:
```
array(['Direct', 'Corporate', 'Online TA', 'Offline TA/TO',
       'Complementary', 'Groups', 'Undefined', 'Aviation'], dtype=object)
```

In [13]:
```
del hotel_data["market_segment"]
```

In [14]:
```
del hotel_data["distribution_channel"]
```

Checking for columns where the sum of null values is more than 70% of total rows

In [122…
```
for col in hotel_data.columns:
    if np.sum (hotel_data[col].isnull()) > (hotel_data.shape[0]*0.70):
        print(col)
```

company

In [7]:
```
# cancelled column variable can be changed

hotel_data["is_canceled"].dtype
```

Out[7]:
```
dtype('int64')
```

In [25]:
```
new_is_canceled = pd.Categorical(hotel_data["is_canceled"])
new_is_canceled = new_is_canceled.rename_categories(["Cancelled","Booked"])

new_is_canceled.describe()
```

Out[25]:

categories	counts	freqs
Cancelled	75011	0.629234
Booked	44199	0.370766

In [26]:
```
hotel_data["Booking Status"] = new_is_canceled
```

In [28]:
```
hotel_data.head(5)
```

Out[28]:

	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	...	agent	company
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	NaN	NaN
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	NaN	NaN
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	NaN	NaN
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	304.0	NaN
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	240.0	NaN

5 rows × 33 columns

In [20]:
```
# del hotel_data[is_canceled]
```

Combining Dates Columns

hotel_data["Arrival Date"] = pd.to_datetime(hotel_data["arrival_date_year"].astype(str) + '/' + hotel_data["arrival_date_month"].astype(str) + '/' + hotel_data["arrival_date_day_of_month"].astype(str))

In [29]:
```
hotel_data["Arrival Date"] = pd.to_datetime(hotel_data["arrival_date_year"].astype(str) + '/' + hotel_data["arrival_date_month"].astype(str) + '/' + hotel_data["arrival
```

In [30]:
```
hotel_data.head(5)
```

Out[30]:

	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	...	company
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	NaN
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	NaN
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	NaN
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	NaN
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	NaN

5 rows × 34 columns

In [31]:
```
# combining children and babies

hotel_data["children"].describe()
```

Out[31]:
```
count    119210.000000
mean          0.104047
std           0.398835
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          10.000000
Name: children, dtype: float64
```

In [32]:
```
np.sum(hotel_data.isnull())
```

Out[32]:
```
hotel                                  0
is_canceled                            0
lead_time                              0
arrival_date_year                      0
arrival_date_month                     0
arrival_date_week_number               0
arrival_date_day_of_month              0
stays_in_weekend_nights                0
stays_in_week_nights                   0
adults                                 0
children                               0
babies                                 0
meal                                   0
country                              478
market_segment                         0
distribution_channel                   0
is_repeated_guest                      0
previous_cancellations                 0
previous_bookings_not_canceled         0
reserved_room_type                     0
assigned_room_type                     0
booking_changes                        0
deposit_type                           0
agent                              16280
company                           112442
days_in_waiting_list                   0
customer_type                          0
adr                                    0
required_car_parking_spaces            0
total_of_special_requests              0
reservation_status                     0
reservation_status_date                0
Booking Status                         0
Arrival Date                           0
dtype: int64
```

In [33]:
```
new_children = np.where(hotel_data["children"].isnull(),
                        np.mean(hotel_data["children"]),
                        hotel_data["children"])

hotel_data["children"] = new_children

np.sum(hotel_data["children"].isnull())
```

Out[33]:
```
0
```

In [34]:
```
hotel_data["Children"]=hotel_data["children"].astype(int) + hotel_data["babies"].astype(int)
```

In [35]:
```
hotel_data.isnull().sum()
```

Out[35]:
```
hotel                                  0
is_canceled                            0
lead_time                              0
arrival_date_year                      0
arrival_date_month                     0
arrival_date_week_number               0
arrival_date_day_of_month              0
stays_in_weekend_nights                0
stays_in_week_nights                   0
adults                                 0
children                               0
babies                                 0
meal                                   0
country                              478
market_segment                         0
distribution_channel                   0
is_repeated_guest                      0
previous_cancellations                 0
previous_bookings_not_canceled         0
reserved_room_type                     0
assigned_room_type                     0
booking_changes                        0
deposit_type                           0
agent                              16280
company                           112442
days_in_waiting_list                   0
customer_type                          0
adr                                    0
required_car_parking_spaces            0
total_of_special_requests              0
reservation_status                     0
reservation_status_date                0
Booking Status                         0
Arrival Date                           0
Children                               0
dtype: int64
```

In [36]:
```
# hotel cancelled bookings
hotel_data['Booking Status'].value_counts()
```

Out[36]:
```
Cancelled    75011
Booked       44199
Name: Booking Status, dtype: int64
```

In [37]:
```
Cancelled = hotel_data[hotel_data["Booking Status"] == "Cancelled"]
Cancelled_per = (len(Cancelled)/hotel_data["Booking Status"].shape[0])*100
#here I have shown how to use shape[0] to get the total number of rows also use len() to get the total number of rows cancelled

Cancelled_per
```

Out[37]:
```
62.9234124653972
```

In [38]:
```
hotel_data["arrival_date_month"].value_counts()
```

Out[38]:
```
August       13861
July         12644
May          11780
October      11147
April        11078
June         10929
September    10500
March         9768
February      8052
November      6771
December      6759
January       5921
Name: arrival_date_month, dtype: int64
```

Errors

The columns for adults & children cannot be blank for the same row as this will mean no one is in the room. We can also assume that a booking cannot have children with having adults.

In [11]:
```
filter = (hotel_data.children == 0) & (hotel_data.adults == 0)
hotel_data[filter]
```

Out[11]:

	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	...	deposit_type	agent
2224	Resort Hotel	0	1	2015	October	41	6	0	3	0	...	No Deposit	NaN
2409	Resort Hotel	0	0	2015	October	42	12	0	0	0	...	No Deposit	NaN
3181	Resort Hotel	0	36	2015	November	47	20	1	2	0	...	No Deposit	38.0
3684	Resort Hotel	0	165	2015	December	53	30	1	4	0	...	No Deposit	308.0
3708	Resort Hotel	0	165	2015	December	53	30	2	4	0	...	No Deposit	308.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
115029	City Hotel	0	107	2017	June	26	27	0	3	0	...	No Deposit	7.0
115091	City Hotel	0	1	2017	June	26	30	0	1	0	...	No Deposit	NaN
116251	City Hotel	0	44	2017	July	28	15	1	1	0	...	No Deposit	425.0
116534	City Hotel	0	2	2017	July	28	15	2	5	0	...	No Deposit	9.0
117087	City Hotel	0	170	2017	July	30	27	0	2	0	...	No Deposit	52.0

180 rows × 32 columns

Data indicates there are 180 rows where children and adults are blank

In [20]:
```
## drop rows that have 0 for children and adults

error_guest = hotel_data[(hotel_data.children == 0) & (hotel_data.adults == 0)].index

hotel_data.drop(error_guest, inplace = True)

## reset index to new dataset
hotel_data.reset_index(drop = True, inplace=True)
hotel_data.head()
```

Out[20]:

	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	...	deposit_type	agent
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	No Deposit	NaN
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	No Deposit	NaN
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	No Deposit	NaN
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	No Deposit	304.0
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	No Deposit	240.0

5 rows × 32 columns

Checking rows were dropped - 119390 minus 180 = 119210

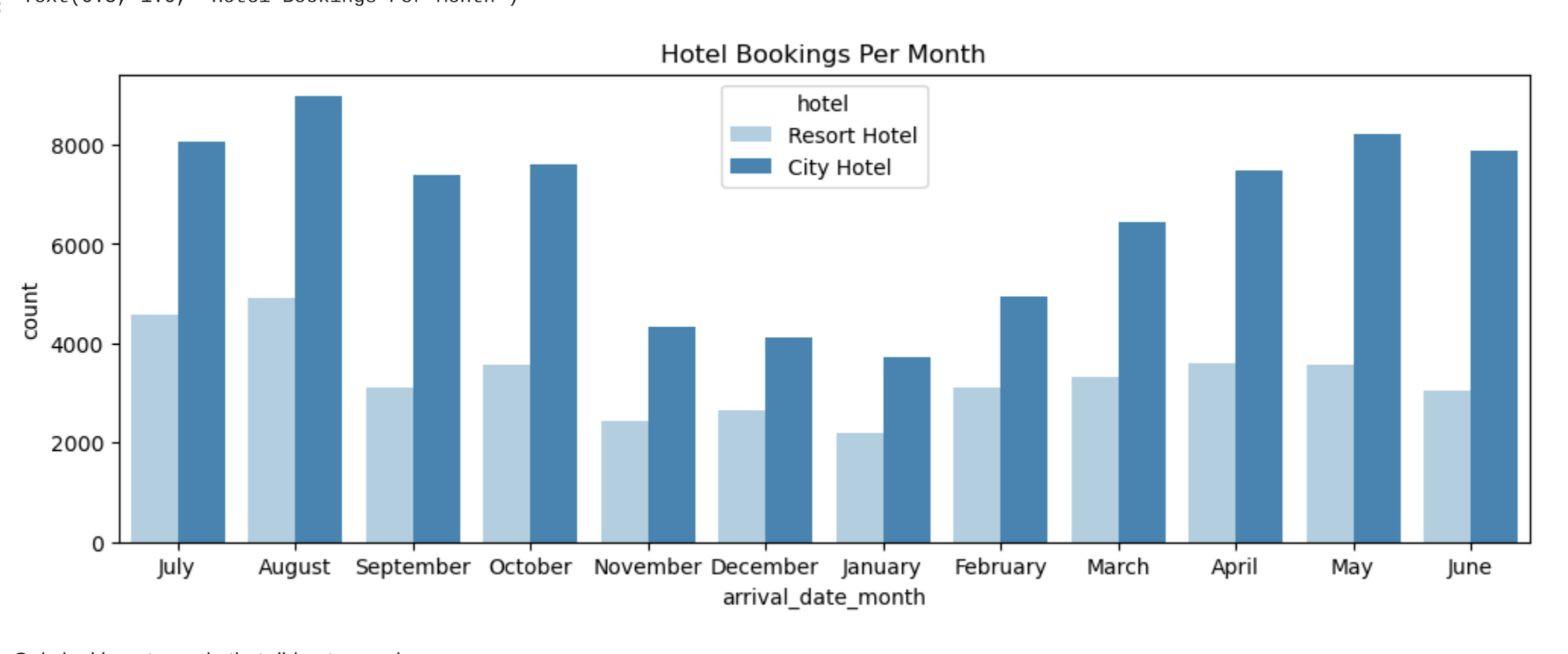
In [21]:
```
hotel_data.info()
```

```
&lt;class 'pandas.core.frame.DataFrame'&gt;
RangeIndex: 119210 entries, 0 to 119209
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119210 non-null  object
 1   is_canceled                     119210 non-null  int64
 2   lead_time                       119210 non-null  int64
 3   arrival_date_year               119210 non-null  int64
 4   arrival_date_month              119210 non-null  object
 5   arrival_date_week_number        119210 non-null  int64
 6   arrival_date_day_of_month       119210 non-null  int64
 7   stays_in_weekend_nights         119210 non-null  int64
 8   stays_in_week_nights            119210 non-null  int64
 9   adults                          119210 non-null  int64
 10  children                        119210 non-null  float64
 11  babies                          119210 non-null  int64
 12  meal                            119210 non-null  object
 13  country                         118732 non-null  object
 14  market_segment                  119210 non-null  object
 15  distribution_channel            119210 non-null  object
 16  is_repeated_guest               119210 non-null  int64
 17  previous_cancellations          119210 non-null  int64
 18  previous_bookings_not_canceled  119210 non-null  int64
 19  reserved_room_type              119210 non-null  object
 20  assigned_room_type              119210 non-null  object
 21  booking_changes                 119210 non-null  int64
 22  deposit_type                    119210 non-null  object
 23  agent                           102930 non-null  float64
 24  company                         6768 non-null    float64
 25  days_in_waiting_list            119210 non-null  int64
 26  customer_type                   119210 non-null  object
 27  adr                             119210 non-null  float64
 28  required_car_parking_spaces     119210 non-null  int64
 29  total_of_special_requests       119210 non-null  int64
 30  reservation_status              119210 non-null  object
 31  reservation_status_date         119210 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

Visualising the data

In [144…
```
plt.figure(figsize = (12,4))
sns.countplot(x = 'arrival_date_month', hue = 'hotel',data = hotel_data, palette = 'Blues')

plt.title("Hotel Bookings Per Month")
```

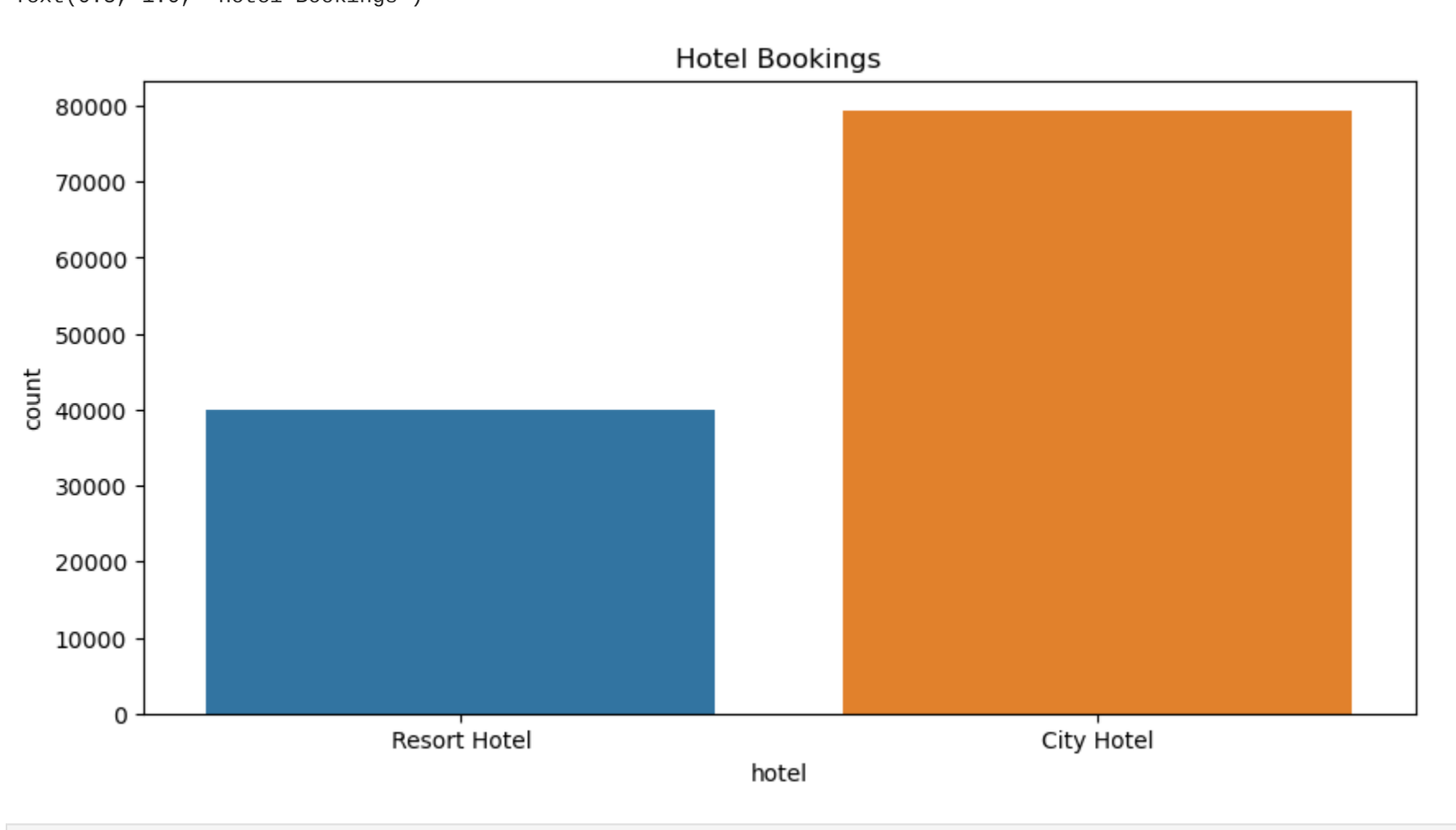
Out[144]:
```
Text(0.5, 1.0, 'Hotel Bookings Per Month')
```



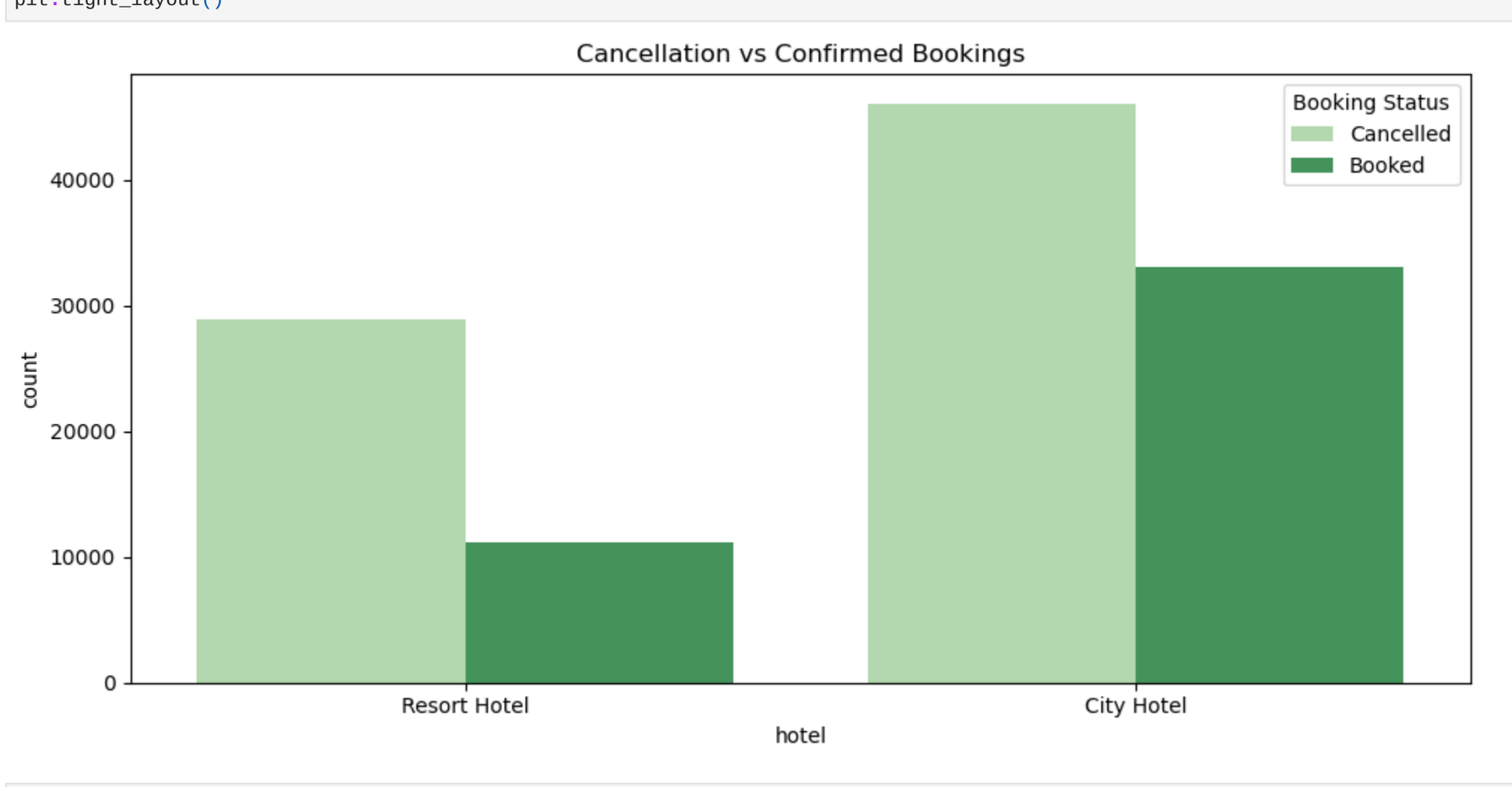
Only looking at people that did not cancel

In [32]:
```
plt.figure(figsize = (10,5))
sns.countplot(x = 'hotel', data = hotel_data)

plt.title("Hotel Bookings")
```

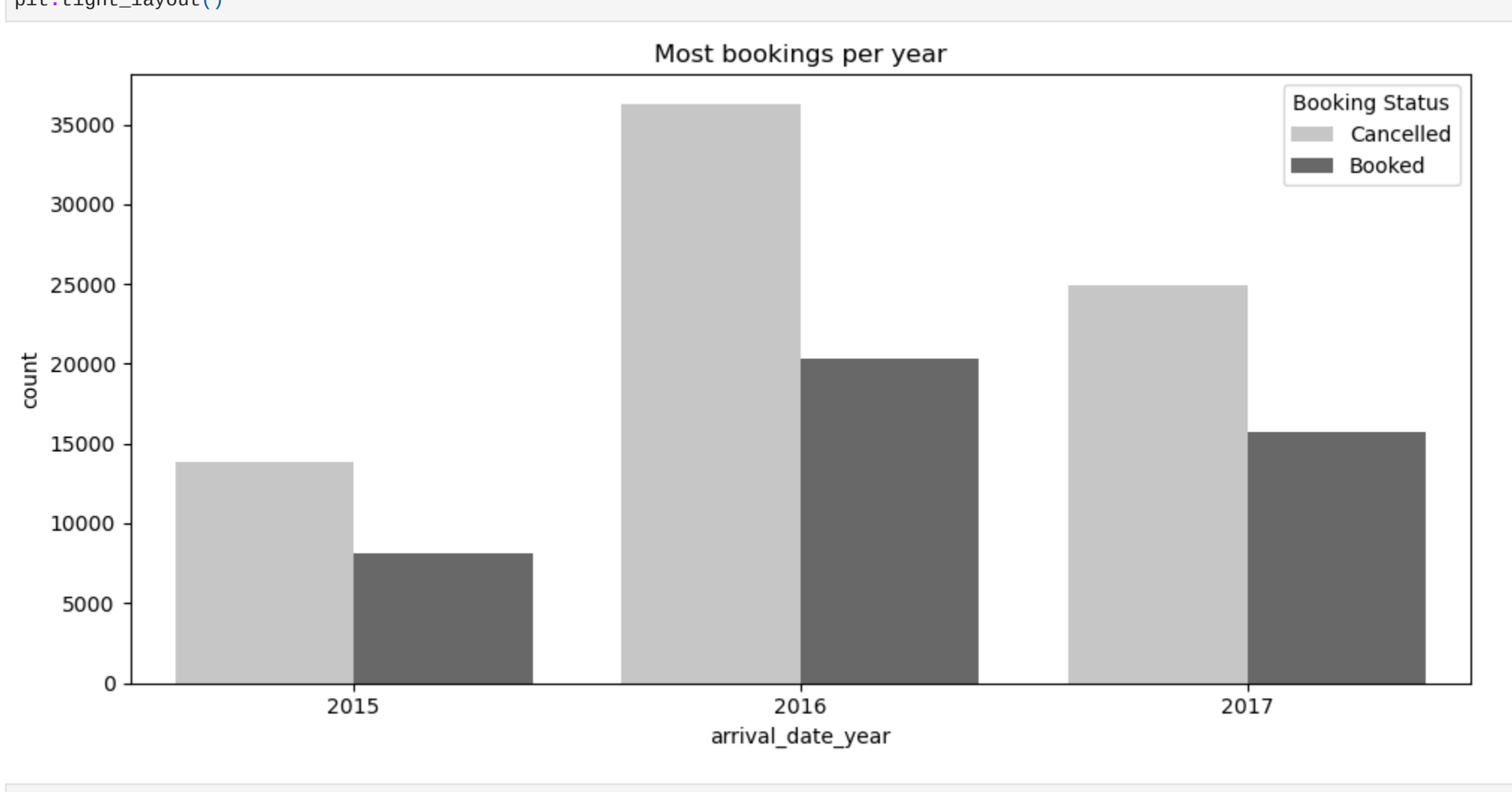
Out[32]:
```
Text(0.5, 1.0, 'Hotel Bookings')
```



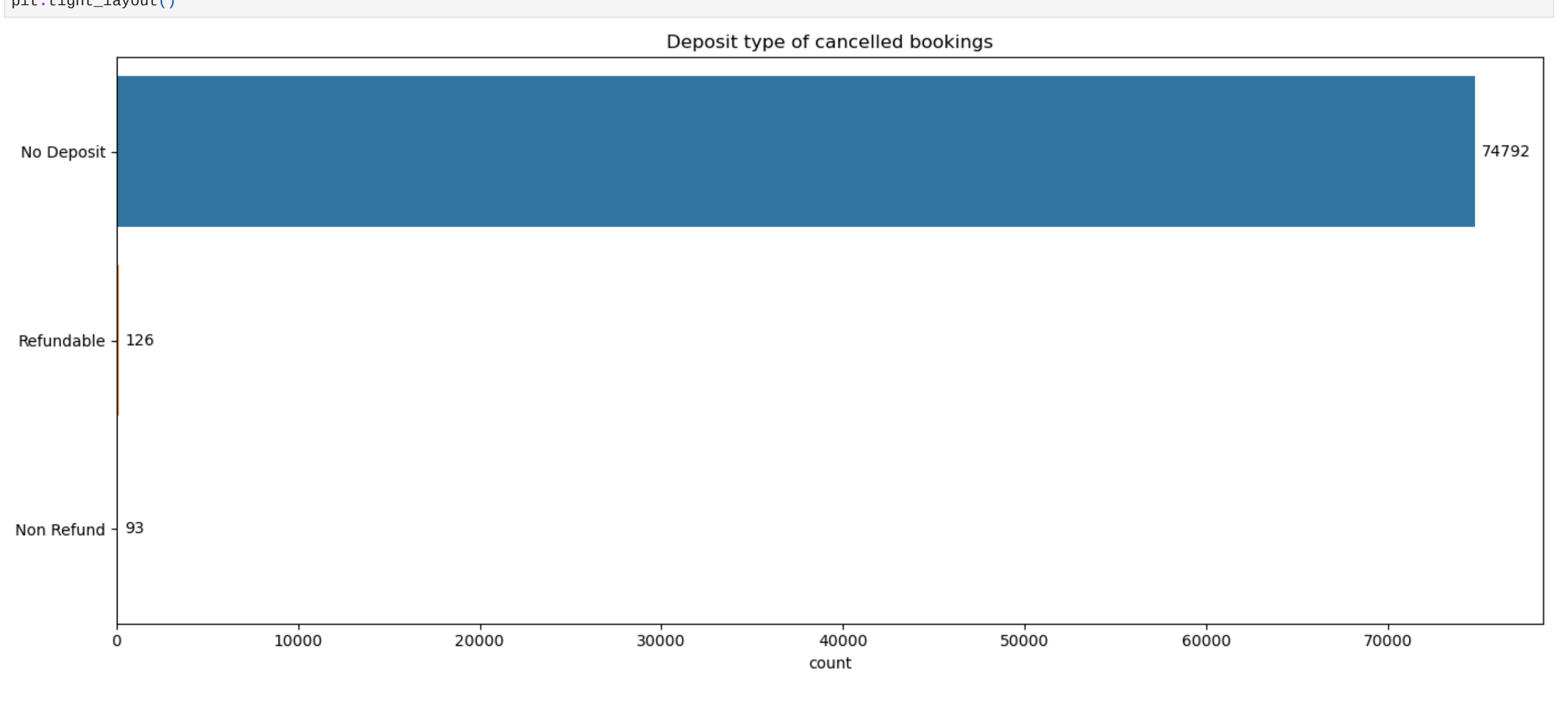
In [155…
```
plt.figure(figsize = (10,5))

sns.countplot(x = 'hotel', data = hotel_data, hue = 'Booking Status', palette = 'Greens')
plt.title("Cancellation vs Confirmed Bookings")

plt.tight_layout()
```



In [152…
```
plt.figure(figsize = (10,5))

sns.countplot(x = 'arrival_date_year', data = hotel_data, hue = 'Booking Status', palette = 'Greys')
plt.title("Most bookings per year")

plt.tight_layout()
```



In [151…
```
fig, ax = plt.subplots(figsize = (14, 6))

sns.countplot(ax = ax,
          y = 'deposit_type',
          data = Cancelled,
          orient = "h",
          ).set(ylabel = None)
ax.bar_label(ax.containers[0], padding = 4)
plt.title('Deposit type of cancelled bookings')

plt.tight_layout()
```



Only looking at people that did not cancel

In [154…
```
non_can = (hotel_data["Booking Status"]== "Booked")
hotel_data[non_can]
```

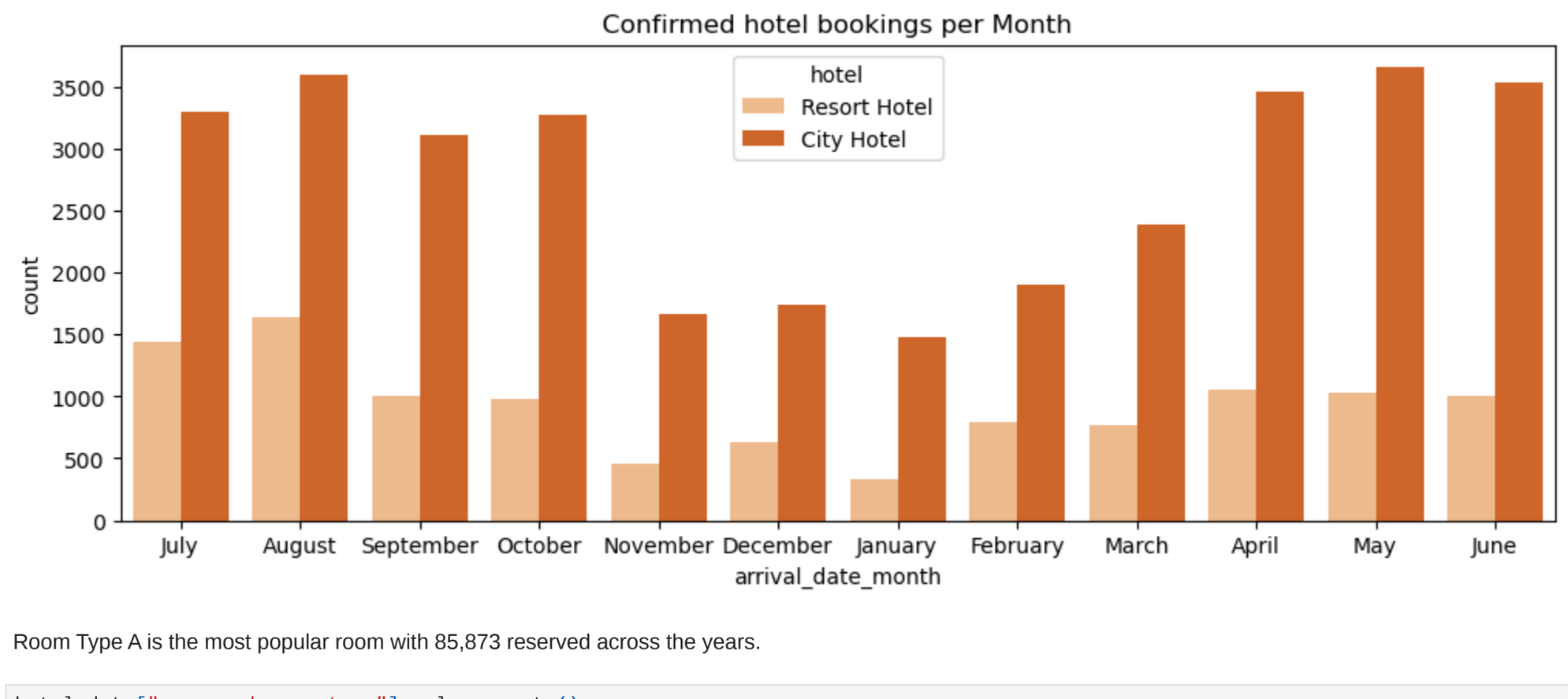
Out[154]:

	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	...	days_in_waiting_list
8	Resort Hotel	1	85	2015	July	27	1	0	3	2	...	0
9	Resort Hotel	1	75	2015	July	27	1	0	3	2	...	0
10	Resort Hotel	1	23	2015	July	27	1	0	4	2	...	0
27	Resort Hotel	1	60	2015	July	27	1	2	5	2	...	0
32	Resort Hotel	1	96	2015	July	27	1	2	8	2	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
108671	City Hotel	1	25	2017	May	18	6	2	1	1	...	0
111191	City Hotel	1	4	2017	June	23	5	1	0	1	...	0
111756	City Hotel	1	7	2017	May	22	31	0	1	1	...	0
111757	City Hotel	1	6	2017	July	29	17	1	0	1	...	0
117115	City Hotel	1	0	2017	August	31	2	0	2	1	...	0

44199 rows × 35 columns

In [150…
```
plt.figure(figsize = (12,4))
sns.countplot(x = 'arrival_date_month', hue = 'hotel',data = hotel_data[non_can], palette = 'Oranges')

plt.title("Confirmed hotel bookings per Month")
```

Out[150]:
```
Text(0.5, 1.0, 'Confirmed hotel bookings per Month')
```

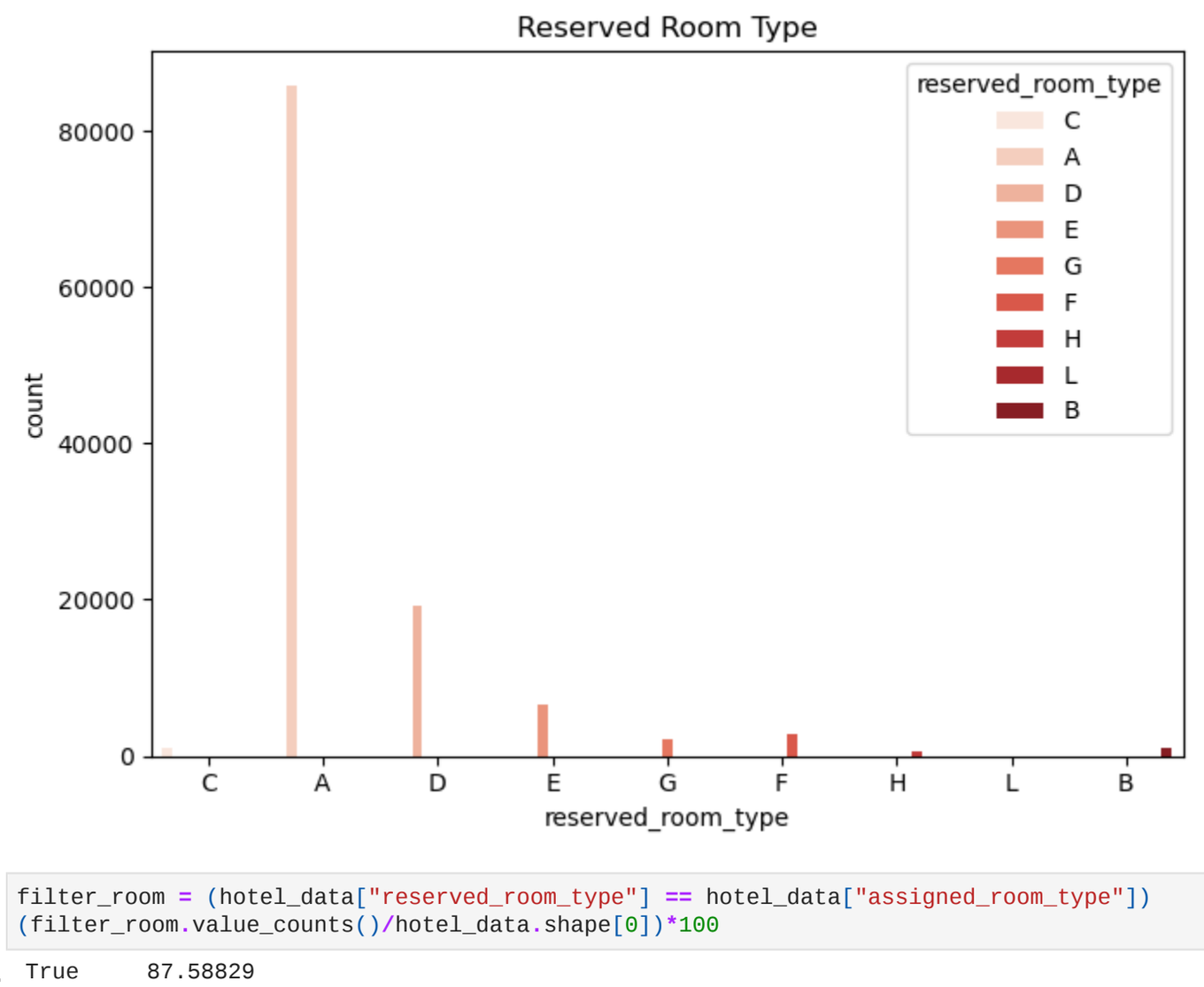


Room Type A is the most popular room with 85,873 reserved across the years.

In [89]:
```
hotel_data["reserved_room_type"].value_counts()
```

Out[89]:
```
A    85873
D    19179
E     6519
F     2894
G     2092
B     1115
C      931
H      601
L        6
Name: reserved_room_type, dtype: int64
```

In [142…
```
plt.figure(figsize = (7,5))

sns.countplot(x = 'reserved_room_type', data = hotel_data, hue = 'reserved_room_type', palette = 'Reds')
plt.title("Reserved Room Type")

plt.tight_layout()
```

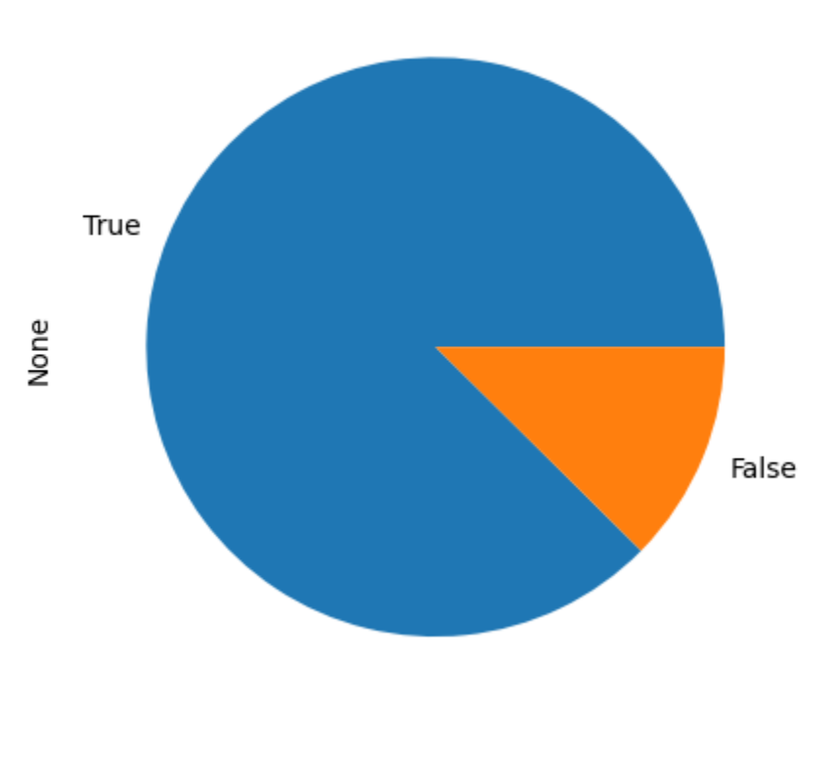


In [148…
```
filter_room = (hotel_data["reserved_room_type"] == hotel_data["assigned_room_type"])
(filter_room.value_counts()/hotel_data.shape[0])*100
```

Out[148]:
```
True     87.58829
False    12.41171
dtype: float64
```

In [133…
```
filter_room.value_counts().plot(kind='pie')
```

Out[133]:
```
&lt;AxesSubplot:ylabel='None'&gt;
```



Using a boolean data type to see if the room customers reserved was the same as what they were assigned. 88% of guests were assigned the room they reserved.

In [ ]: