1. Create a new Data Set as "df" and read the data set 'Hotel Reservations'.

In [2]:
```
import pandas as pd
df = pd.read_csv('Hotel Reservations.csv')
df.head(5)
```

Out[2]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_meal
0	INN00001	2	0	1	2	Meal I
1	INN00002	2	0	2	3	Not Se
2	INN00003	1	0	2	1	Meal I
3	INN00004	2	0	0	2	Meal I
4	INN00005	2	0	1	1	Not Se

2. Find out about the following charateristics of your data set...

- shape/ how many rows, how many colums
- columns'names
- data types of the columns entries
- null values
- decriptive statitics (mean, std, ...)

In [5]:
```
df.shape
```

Out[5]:
```
(36275, 19)
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Booking_ID', 'no_of_adults', 'no_of_children', 'no_of_weekend_nig
hts',
       'no_of_week_nights', 'type_of_meal_plan', 'required_car_parking_sp
ace',
       'room_type_reserved', 'lead_time', 'arrival_year', 'arrival_mont
h',
       'arrival_date', 'market_segment_type', 'repeated_guest',
       'no_of_previous_cancellations', 'no_of_previous_bookings_not_cance
led',
       'avg_price_per_room', 'no_of_special_requests', 'booking_status'],
      dtype='object')
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Booking_ID                            36275 non-null  object 
 1   no_of_adults                          36275 non-null  int64  
 2   no_of_children                        36275 non-null  int64  
 3   no_of_weekend_nights                  36275 non-null  int64  
 4   no_of_week_nights                     36275 non-null  int64  
 5   type_of_meal_plan                     36275 non-null  object 
 6   required_car_parking_space            36275 non-null  int64  
 7   room_type_reserved                    36275 non-null  object 
 8   lead_time                             36275 non-null  int64  
 9   arrival_year                          36275 non-null  int64  
 10  arrival_month                         36275 non-null  int64  
 11  arrival_date                          36275 non-null  int64  
 12  market_segment_type                   36275 non-null  object 
 13  repeated_guest                        36275 non-null  int64  
 14  no_of_previous_cancellations          36275 non-null  int64  
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64  
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64  
 18  booking_status                        36275 non-null  object 
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
Booking_ID                            0
no_of_adults                          0
no_of_children                        0
no_of_weekend_nights                  0
no_of_week_nights                     0
type_of_meal_plan                     0
required_car_parking_space            0
room_type_reserved                    0
lead_time                             0
arrival_year                          0
arrival_month                         0
arrival_date                          0
market_segment_type                   0
repeated_guest                        0
no_of_previous_cancellations          0
no_of_previous_bookings_not_canceled  0
avg_price_per_room                    0
no_of_special_requests                0
booking_status                        0
dtype: int64
```

In [9]:
```
df.describe()
```

Out[9]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	required_car_parking
count	36275.000000	36275.000000	36275.000000	36275.000000	36275.
mean	1.844962	0.105279	0.810724	2.204300	0.
std	0.518715	0.402648	0.870644	1.410905	0.
min	0.000000	0.000000	0.000000	0.000000	0.
25%	2.000000	0.000000	0.000000	1.000000	0.
50%	2.000000	0.000000	1.000000	2.000000	0.
75%	2.000000	0.000000	2.000000	3.000000	0.
max	4.000000	10.000000	7.000000	17.000000	1.

3. Show only data for stays of repeated guests with children

In [3]:
```
repeat_and_children = df.loc[(df['no_of_children'] > 0) & (df['repeated_gue
repeat_and_children.shape
```

Out[3]:
```
(13, 19)
```

4. Order the entries ascending by year and month

In [11]:
```
sorted_df = df.sort_values(['arrival_year', 'arrival_month'], ascending=[1,
sorted_df
```

Out[11]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_
153	INN00154	2	0	0	2	M
201	INN00202	2	0	0	2	M
331	INN00332	2	0	0	2	M
332	INN00333	2	0	2	1	M
560	INN00561	2	0	0	2	M
...	...	...	...	...	...	
36143	INN36144	2	0	0	3	No
36175	INN36176	2	0	0	3	M
36184	INN36185	2	1	2	0	M
36196	INN36197	2	0	0	3	M
36274	INN36275	2	0	1	2	M

36275 rows × 19 columns

5. What is the average price of each room type?

In [26]:
```
avg_price_per_roomtype = df.groupby('room_type_reserved')['avg_price_per_ro
avg_price_per_roomtype
```

Out[26]:
```
room_type_reserved
Room_Type 1     95.918532
Room_Type 2     87.848555
Room_Type 3     73.678571
Room_Type 4    125.287317
Room_Type 5    123.733623
Room_Type 6    182.212836
Room_Type 7    155.198291
Name: avg_price_per_room, dtype: float64
```

6. Create a new column as "total nights" which is the sum of weekend and week nights.

In [31]:
```
df['Total_Nights'] = df['no_of_weekend_nights'] + df['no_of_week_nights']
df
```

Out[31]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_
0	INN00001	2	0	1	2	M
1	INN00002	2	0	2	3	No
2	INN00003	1	0	2	1	M
3	INN00004	2	0	0	2	M
4	INN00005	2	0	1	1	No
...	...	...	...	...	...	...
36270	INN36271	3	0	2	6	M
36271	INN36272	2	0	1	3	M
36272	INN36273	2	0	2	6	M
36273	INN36274	2	0	0	3	No
36274	INN36275	2	0	1	2	M

36275 rows × 20 columns

7. Save your modified data set as a new csv file.

In [ ]:
```
df.to_csv('modified.csv', index=False)
```