```
import pandas as pd
```

```
hotel_data = pd.read_csv("/content/drive/MyDrive/hotel_bookings.csv")
```

```
hotel_data
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_da
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	
...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	
119386	City Hotel	0	102	2017	August	
119387	City Hotel	0	34	2017	August	
119388	City Hotel	0	109	2017	August	
119389	City Hotel	0	205	2017	August	

119390 rows × 32 columns

```
hotel_data.shape
```

```
(119390, 32)
```

```
hotel_data.columns #looking at the columns
```

```
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')
```

```
hotel_data.info() #datatypes of the columns is viewed here
```

```
<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
```

```
hotel_data.describe() #20columns in 32 are numerical
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival
count	119390.000000	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	27.165173	
std	0.482918	106.863097	0.707476	13.605138	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	
50%	0.000000	69.000000	2016.000000	28.000000	
75%	1.000000	160.000000	2017.000000	38.000000	
max	1.000000	737.000000	2017.000000	53.000000	

```
hotel_data.isna().sum()  #3{columns has missing values :  country,agent,company}
```

```
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                              4
babies                                0
meal                                  0
country                             488
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                             16340
company                          112593
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
dtype: int64
```

*company has some missing values and its a float *

```
hotel_data['company'].unique()
```

```
array([ nan, 110., 113., 270., 178., 240., 154., 144., 307., 268.,  59.,
       204., 312., 318.,  94., 174., 274., 195., 223., 317., 281., 118.,
        53., 286.,  12.,  47., 324., 342., 373., 371., 383.,  86.,  82.,
       218.,  88.,  31., 397., 392., 405., 331., 367.,  20.,  83., 416.,
        51., 395., 102.,  34.,  84., 360., 394., 457., 382., 461., 478.,
       386., 112., 486., 421.,   9., 308., 135., 224., 504., 269., 356.,
       498., 390., 513., 203., 263., 477., 521., 169., 515., 445., 337.,
       251., 428., 292., 388., 130., 250., 355., 254., 543., 531., 528.,
        62., 120.,  42.,  81., 116., 530., 103.,  39.,  16.,  92.,  61.,
       501., 165., 291., 290.,  43., 325., 192., 108., 200., 465., 287.,
       297., 490., 482., 207., 282., 437., 225., 329., 272.,  28.,  77.,
       338.,  72., 246., 319., 146., 159., 380., 323., 511., 407., 278.,
        80., 403., 399.,  14., 137., 343., 346., 347., 349., 289., 351.,
       353.,  54.,  99., 358., 361., 362., 366., 372., 365., 277., 109.,
       377., 379.,  22., 378., 330., 364., 401., 232., 255., 384., 167.,
       212., 514., 391., 400., 376., 402., 396., 302., 398.,   6., 370.,
       369., 409., 168., 104., 408., 413., 148.,  10., 333., 419., 415.,
       424., 425., 423., 422., 435., 439., 442., 448., 443., 454., 444.,
        52., 459., 458., 456., 460., 447., 470., 466., 484., 184., 485.,
        32., 487., 491., 494., 193., 516., 496., 499.,  29.,  78., 520.,
       507., 506., 512., 126.,  64., 242., 518., 523., 539., 534., 436.,
       525., 541.,  40., 455., 410.,  45.,  38.,  49.,  48.,  67.,  68.,
        65.,  91.,  37.,   8., 179., 209., 219., 221., 227., 153., 186.,
       253., 202., 216., 275., 233., 280., 309., 321.,  93., 316.,  85.,
       107., 350., 279., 334., 348., 150.,  73., 385., 418., 197., 450.,
       452., 115.,  46.,  76.,  96., 100., 105., 101., 122.,  11., 139.,
       142., 127., 143., 140., 149., 163., 160., 180., 238., 183., 222.,
       185., 217., 215., 213., 237., 230., 234.,  35., 245., 158., 258.,
       259., 260., 411., 257., 271.,  18., 106., 210., 273.,  71., 284.,
       301., 305., 293., 264., 311., 304., 313., 288., 320., 314., 332.,
       341., 352., 243., 368., 393., 132., 220., 412., 420., 426., 417.,
       429., 433., 446., 357., 479., 483., 489., 229., 481., 497., 451.,
       492.])
```

```
hotel_data['company'].dtype
```

```
dtype('float64')
```

agent column (float) : contains missing values :16340

```
hotel_data['agent'].dtype
```

```
dtype('float64')
```

```
hotel_data['agent'].isna().sum()
```

```
16340
```

```
hotel_data['agent'].unique()  #this also is containg float values
```

```
array([ nan, 304., 240., 303.,  15., 241.,   8., 250., 115.,   5., 175.,
       134., 156., 243., 242.,   3., 105.,  40., 147., 306., 184.,  96.,
         2., 127.,  95., 146.,   9., 177.,   6., 143., 244., 149., 167.,
       300., 171., 305.,  67., 196., 152., 142., 261., 104.,  36.,  26.,
        29., 258., 110.,  71., 181.,  88., 251., 275.,  69., 248., 208.,
       256., 314., 126., 281., 273., 253., 185., 330., 334., 328., 326.,
       321., 324., 313.,  38., 155.,  68., 335., 308., 332.,  94., 348.,
       310., 339., 375.,  66., 327., 387., 298.,  91., 245., 385., 257.,
       393., 168., 405., 249., 315.,  75., 128., 307.,  11., 436.,   1.,
       201., 183., 223., 368., 336., 291., 464., 411., 481.,  10., 154.,
       468., 410., 390., 440., 495., 492., 493., 434.,  57., 531., 420.,
       483., 526., 472., 429.,  16., 446.,  34.,  78., 139., 252., 270.,
        47., 114., 301., 193., 182., 135., 350., 195., 352., 355., 159.,
       363., 384., 360., 331., 367.,  64., 406., 163., 414., 333., 427.,
       431., 430., 426., 438., 433., 418., 441., 282., 432.,  72., 450.,
       180., 454., 455.,  59., 451., 254., 358., 469., 165., 467., 510.,
       337., 476., 502., 527., 479., 508., 535., 302., 497., 187.,  13.,
         7.,  27.,  14.,  22.,  17.,  28.,  42.,  20.,  19.,  45.,  37.,
        61.,  39.,  21.,  24.,  41.,  50.,  30.,  54.,  52.,  12.,  44.,
        31.,  83.,  32.,  63.,  60.,  55.,  56.,  89.,  87., 118.,  86.,
        85., 210., 214., 129., 179., 138., 174., 170., 153.,  93., 151.,
       119.,  35., 173.,  58.,  53., 133.,  79., 235., 192., 191., 236.,
       162., 215., 157., 287., 132., 234.,  98.,  77., 103., 107., 262.,
       220., 121., 205., 378.,  23., 296., 290., 229.,  33., 286., 276.,
       425., 484., 323., 403., 219., 394., 509., 111., 423.,   4.,  70.,
        82.,  81.,  74.,  92.,  99.,  90., 112., 117., 106., 148., 158.,
       144., 211., 213., 216., 232., 150., 267., 227., 247., 278., 280.,
       285., 289., 269., 295., 265., 288., 122., 294., 325., 341., 344.,
       346., 359., 283., 364., 370., 371.,  25., 141., 391., 397., 416.,
       404., 299., 197.,  73., 354., 444., 408., 461., 388., 453., 459.,
       474., 475., 480., 449.])
```

country column : has some 488 null values(object)

```
hotel_data['country'].unique()
```

```
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)
```

```
hotel_data['country'].isnull().sum()  #null values : 488
```

```
488
```

```
hotel_data['country'].dtype
```

```
dtype('O')
```

HOtels column (object datatype) no missing values in the column

```
hotel_data['hotel'].unique() #2 hotels (column 1)
```

```
array(['Resort Hotel', 'City Hotel'], dtype=object)
```

```
hotel_data['hotel'].dtype
```

```
dtype('O')
```

```
hotel_data[hotel_data['is_canceled']==0]
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_da
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	
...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	
119386	City Hotel	0	102	2017	August	
119387	City Hotel	0	34	2017	August	
119388	City Hotel	0	109	2017	August	
119389	City Hotel	0	205	2017	August	

75166 rows × 32 columns

Analyze the relationship between market_segment and country

```
hotel_data['market_segment'].unique()
```

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

```
hotel_data['market_segment'].value_counts()
```

```
market_segment
Online TA        56477
Offline TA/TO    24219
Groups           19811
Direct           12606
Corporate         5295
Complementary      743
Aviation           237
```

```
Undefined            2
Name: count, dtype: int64
```

```
hotel_data['country'].value_counts()  #highest country people booked from portugal
```

```
country
PRT    48590
GBR    12129
FRA    10415
ESP     8568
DEU     7287
       ...
DJI        1
BWA        1
HND        1
VGB        1
NAM        1
Name: count, Length: 177, dtype: int64
```

```
hotel_data[['country', 'market_segment']]
```

	country	market_segment
0	PRT	Direct
1	PRT	Direct
2	GBR	Direct
3	GBR	Corporate
4	GBR	Online TA
...	...	...
119385	BEL	Offline TA/TO
119386	FRA	Online TA
119387	DEU	Online TA
119388	GBR	Online TA
119389	DEU	Online TA

119390 rows × 2 columns

```
nan_rows = hotel_data.loc[hotel_data['country'].isna(), ['country', 'market_segment']]
print(nan_rows)
```

```
       country market_segment
30         NaN         Direct
4127       NaN  Offline TA/TO
7092       NaN      Corporate
```

```
7860       NaN         Direct
8779       NaN      Corporate
...        ...            ...
65908      NaN  Complementary
65909      NaN  Complementary
65910      NaN  Complementary
80830      NaN         Groups
101488     NaN         Direct

[488 rows x 2 columns]
```

*clearing the missing values with imputation of the highest mode country *

```
# Define a function to impute missing values with mode
def impute_country_mode(group):
    mode_country = group['country'].mode().iloc[0]  # Compute mode of 'country' column in th
    group['country'] = group['country'].fillna(mode_country)  # Fill NaN values with mode
    return group

# Group the data by 'market_segment' and apply the imputation function
grouped_data_market = hotel_data.groupby('market_segment')
hotel_data_imputed = grouped_data_market.apply(impute_country_mode)

# Check if there are any remaining NaN values after imputation
print(hotel_data_imputed['country'].isna().sum())
```

```
0
```

```
hotel_data_imputed
```

		hotel	is_canceled	lead_time	arrival_date_year	arrival_date_m
market_segment						
Aviation	**49013**	City Hotel	1	5	2016	
	49372	City Hotel	1	1	2016	
	49411	City Hotel	1	1	2016	
	50468	City Hotel	1	3	2016	
	50843	City Hotel	1	11	2016	
...	**...**	...	...	...	...	
Online TA	**119387**	City Hotel	0	34	2017	A
	119388	City Hotel	0	109	2017	A
	119389	City Hotel	0	205	2017	A
Undefined	**40600**	City Hotel	1	2	2015	A
	40679	City Hotel	1	1	2015	A

119390 rows × 32 columns

```
hotel_data = hotel_data_imputed
```

```
hotel_data.columns
```

```
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')
```

```
hotel_data
```

		hotel	is_canceled	lead_time	arrival_date_year	arrival_date_m
market_segment						
Aviation	**49013**	City Hotel	1	5	2016	
	49372	City Hotel	1	1	2016	
	49411	City Hotel	1	1	2016	
	50468	City Hotel	1	3	2016	
	50843	City Hotel	1	11	2016	
...	**...**	...	...	...	...	
Online TA	**119387**	City Hotel	0	34	2017	A
	119388	City Hotel	0	109	2017	A
	119389	City Hotel	0	205	2017	A
Undefined	**40600**	City Hotel	1	2	2015	A
	40679	City Hotel	1	1	2015	A

119390 rows × 32 columns

```
hotel_data.reset_index(drop=True, inplace=True)
```

```
hotel_data['country'].isnull().sum() #cleaned the country column
```

```
0
```

*clearing the missing values using placeholder in agent column *

```
hotel_data['agent']
```

```
0           NaN
1         153.0
2         153.0
3           NaN
4           NaN
          ...
119385      9.0
119386     89.0
119387      9.0
119388      NaN
119389      NaN
Name: agent, Length: 119390, dtype: float64
```

```
hotel_data['agent'] = hotel_data['agent'].fillna(0).astype(int)
```

```
hotel_data['agent']  #the agent column may represent the code or ID of the agency
```

```
0           0
1         153
2         153
3           0
4           0
          ...
119385      9
119386     89
119387      9
119388      0
119389      0
Name: agent, Length: 119390, dtype: int64
```

```
# Replace missing values in the 'agent' column with a placeholder value (-1)
hotel_data['agent'].fillna(-1, inplace=True)
```

```
hotel_data['agent'].isnull().sum()  #cleaned missing values in agent column
```

```
0
```

*cleaning company column *

```
hotel_data['company']
```

```
0         153.0
1           NaN
2           NaN
3         153.0
4         153.0
          ...
```

```
119385      NaN
119386      NaN
119387      NaN
119388      NaN
119389      NaN
Name: company, Length: 119390, dtype: float64
```

```
hotel_data['company'].fillna(-1, inplace=True)
```

```
hotel_data['company'].astype(int)
```

```
0         153
1          -1
2          -1
3         153
4         153
         ...
119385     -1
119386     -1
119387     -1
119388     -1
119389     -1
Name: company, Length: 119390, dtype: int64
```

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

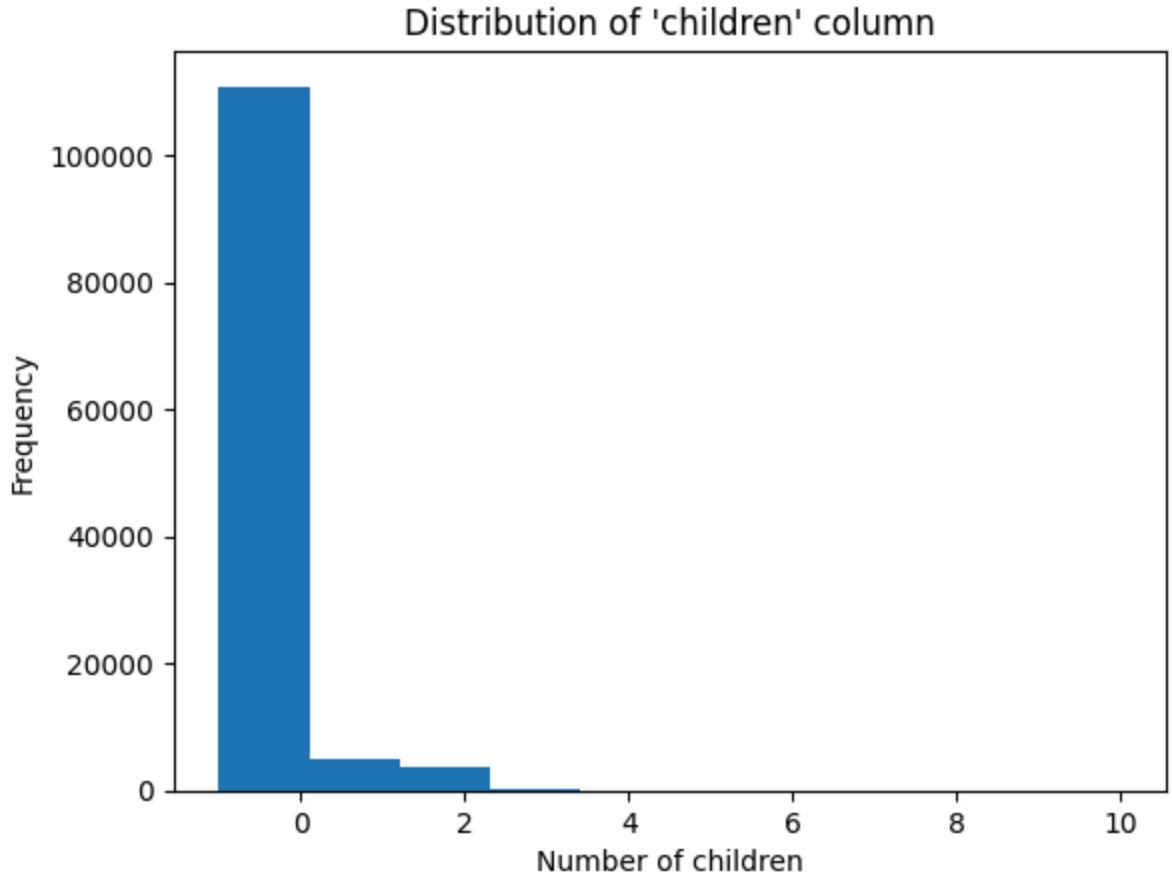
```
array([ 0.,  1.,  2.,  3., nan, 10.])
```

```
hotel_data["children"] = hotel_data["children"].fillna(-1)
```

seeing the skewness of the data

```
import matplotlib.pyplot as plt

plt.hist(hotel_data["children"], bins=10)
plt.title("Distribution of 'children' column")
plt.xlabel("Number of children")
plt.ylabel("Frequency")
plt.show()
```



Given that the 'children' column is heavily skewed towards lower values (0 and 1), and the majority of bookings have either 0 or 1 child, you might consider filling the missing values (NaN) with the mode of the column

```
mode_children = hotel_data['children'].mode()[0]
hotel_data['children'].fillna(mode_children,inplace=True)
```

```
hotel_data['children'].astype(int)
```

```
0         0
1         0
2         0
3         0
4         0
         ..
119385    0
119386    0
119387    0
119388   -1
119389   -1
Name: children, Length: 119390, dtype: int64
```

```
hotel_data['children'].isnull().sum()
```

```
0
```

```
hotel_data['arrival_date_year']#column is good
hotel_data['arrival_date_month']#did some changes to the column into numerical data (useful
```

```
0          April
1          April
2          April
3            May
4            May
           ...
119385    August
119386    August
119387    August
119388    August
119389    August
Name: arrival_date_month, Length: 119390, dtype: object
```

```
month_mapping = {
    'January': 1, 'February': 2, 'March': 3, 'April': 4,
    'May': 5, 'June': 6, 'July': 7, 'August': 8,
    'September': 9, 'October': 10, 'November': 11, 'December': 12
}

# Apply the mapping to the 'arrival_date_month' column
hotel_data['arrival_date_month'] = hotel_data['arrival_date_month'].map(month_mapping)
```

```
hotel_data['arrival_date_month'] #column is good
```

```
0         4
1         4
2         4
3         5
4         5
         ..
119385    8
119386    8
119387    8
119388    8
119389    8
Name: arrival_date_month, Length: 119390, dtype: int64
```

```
hotel_data['arrival_date_week_number'] #column is good
hotel_data['arrival_date_day_of_month'] #column is good
```

```
0          4
1         12
2         12
3          1
4          9
          ..
```

```
119385    31
119386    31
119387    29
119388     3
119389     5
Name: arrival_date_day_of_month, Length: 119390, dtype: int64
```

```
hotel_data['stays_in_weekend_nights'] #column is good
hotel_data['stays_in_week_nights'] #column is good
```

```
0          3
1          1
2          6
3          6
4         10
           ..
119385     5
119386     5
119387     7
119388     0
119389     2
Name: stays_in_week_nights, Length: 119390, dtype: int64
```

```
hotel_data['adults']
hotel_data['babies']
hotel_data['children']
hotel_data['meal'].unique()
```

```
array(['BB', 'FB', 'HB', 'SC', 'Undefined'], dtype=object)
```

The 'meal' column is represented as an object dtype because it contains string values. In pandas, the object dtype is used to store strings, but it's also used for columns that contain mixed data types or columns that pandas can't infer the data type for.

```
hotel_data['meal'].value_counts()
```

```
meal
BB           92310
HB           14463
SC           10650
Undefined     1169
FB             798
Name: count, dtype: int64
```

```
undefined_percentage = (1169 / len(hotel_data)) * 100
undefined_percentage
```

```
0.9791439819080325
```

```
hotel_data['distribution_channel'].unique()
```

```
array(['Corporate', 'TA/TO', 'Direct', 'Undefined', 'GDS'], dtype=object)
```

```
hotel_data['distribution_channel'].value_counts()
```

```
distribution_channel
TA/TO        97870
Direct       14645
Corporate     6677
GDS            193
Undefined        5
Name: count, dtype: int64
```

```
undefined_per_dis = (5/len(hotel_data))*100
undefined_per_dis
```

```
0.004187955440154116
```

```
hotel_data['is_repeated_guest'].unique() #column is good
```

```
array([0, 1])
```

```
hotel_data['previous_cancellations'].unique()#column is good
```

```
array([ 0,  1,  2,  3,  4,  5,  6, 13, 11, 25, 14, 24, 21, 26, 19])
```

```
hotel_data['previous_bookings_not_canceled'].unique() #column is good
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
       68, 69, 70, 71, 72])
```

```
hotel_data['reserved_room_type'].unique()
```

```
array(['A', 'D', 'E', 'H', 'C', 'F', 'G', 'B', 'P', 'L'], dtype=object)
```

```
hotel_data['reserved_room_type'].value_counts()
```

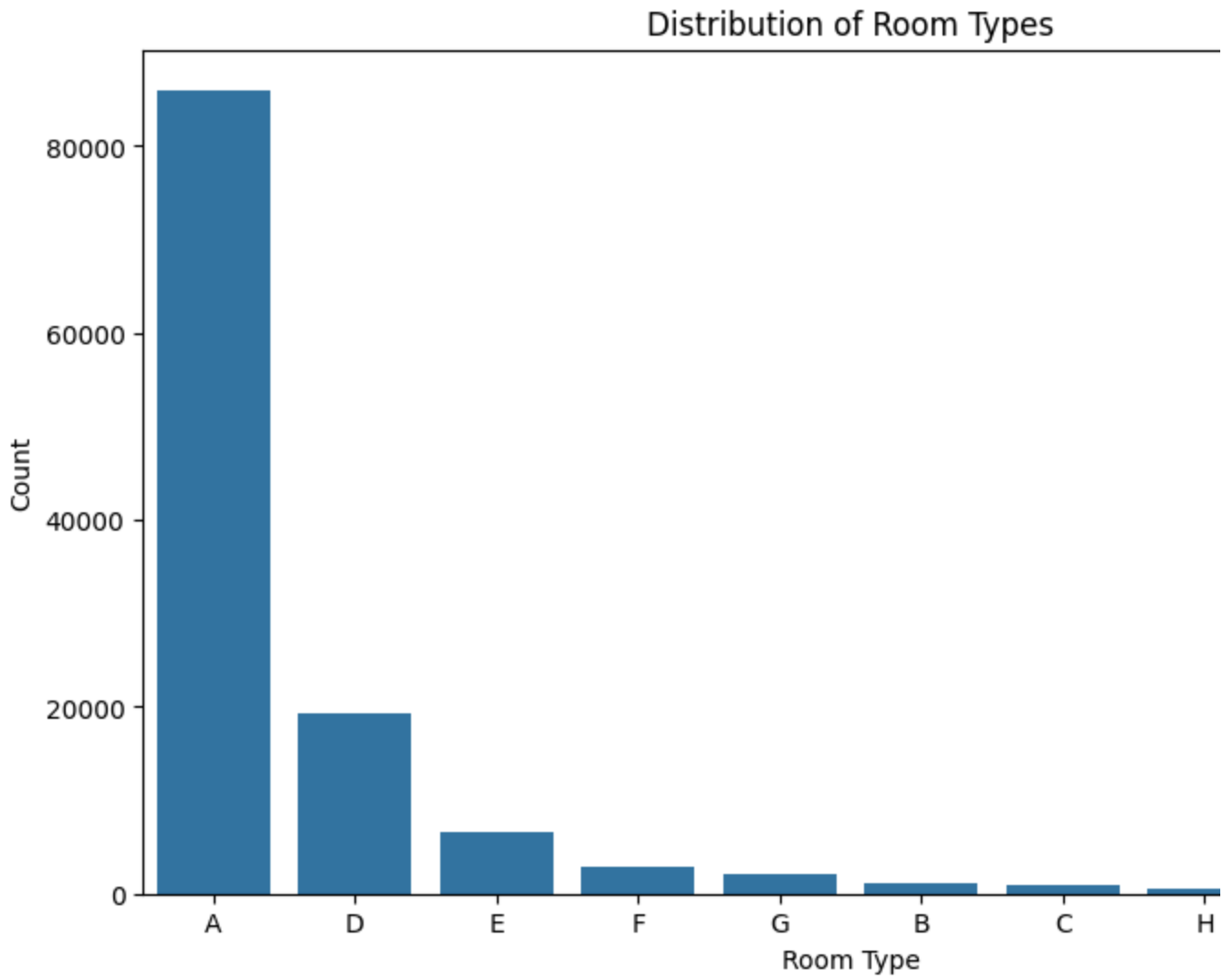
```
reserved_room_type
A    85994
D    19201
E     6535
F     2897
G     2094
B     1118
C      932
```

```
H      601
P       12
L        6
Name: count, dtype: int64
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming hotel_data is your DataFrame and 'reserved_room_type' is the column of interest
room_type_counts = hotel_data['reserved_room_type'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=room_type_counts.index, y=room_type_counts.values)
plt.title('Distribution of Room Types')
plt.xlabel('Room Type')
plt.ylabel('Count')
plt.show()
```



```
hotel_data['assigned_room_type'].unique()
```

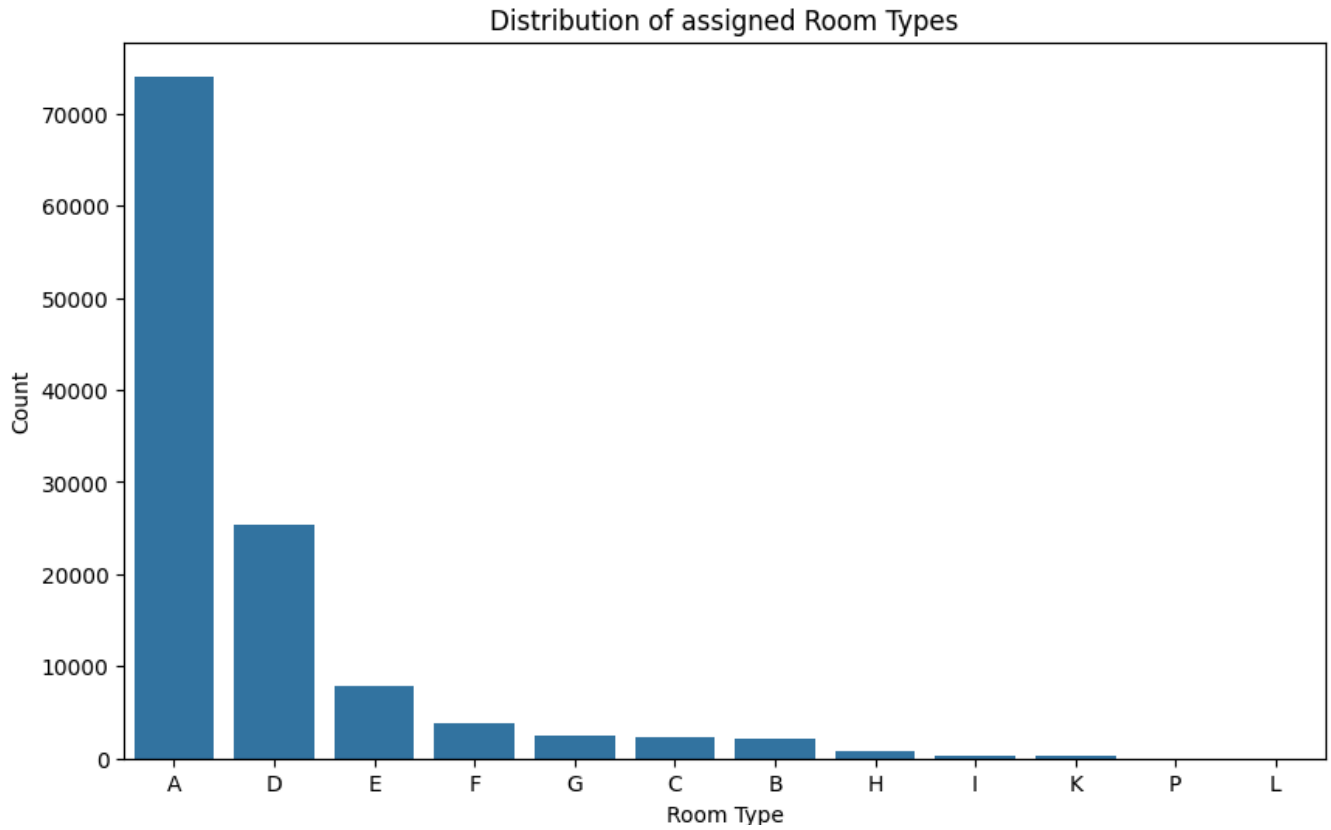
```
array(['A', 'D', 'B', 'G', 'K', 'E', 'H', 'F', 'C', 'I', 'P', 'L'],
      dtype=object)
```

```
hotel_data['assigned_room_type'].value_counts()
```

```
assigned_room_type
A    74053
D    25322
E     7806
F     3751
G     2553
C     2375
B     2163
H      712
I      363
K      279
P       12
L        1
Name: count, dtype: int64
```

```
assigned_room_type_counts = hotel_data['assigned_room_type'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=assigned_room_type_counts.index, y=assigned_room_type_counts.values)
plt.title('Distribution of assigned Room Types')
plt.xlabel('Room Type')
plt.ylabel('Count')
plt.show()
```




```
hotel_data['deposit_type'].unique()
```

```
array(['No Deposit', 'Non Refund', 'Refundable'], dtype=object)
```

```
hotel_data['days_in_waiting_list'].unique() #column is good
```

```
array([  0,   4,  35,  50,  20,  13,   8,  15,  16,  21,  59,  12,   1,
         9, 107, 122,   2,  93,   5,  11,   6,  10,  23,  47,  75, 101,
       150, 125,  14,  60,  34, 100,  22, 121,  61,  39,  43,  52, 142,
       116,  44,  97,  83, 113,  18, 185, 109,  37, 105, 154,  64,  99,
        53,  49,  58,  87,  57,  98,  31, 176, 236, 259,  48,  96,  41,
        32, 379,  38, 330, 174, 391,  76,  28,  17, 111,   7,  46,  84,
        30, 183,  56,  27, 117,  80,  71,  26,  73,  45,  40,  72,  25,
        81,  74,  65,  33,  77,  69,  91,  79,  85,  63,   3, 224, 187,
        55, 207, 215, 160, 120,  62,  24, 108, 147,  70, 178, 223, 162,
        68, 193, 165, 175,  54,  19,  42,  92, 167,  36,  89])
```

```
hotel_data['customer_type'].unique() #column is good
```

```
array(['Transient-Party', 'Transient', 'Group', 'Contract'], dtype=object)
```

```
hotel_data['adr'].dtype
```

```
dtype('float64')
```

```
hotel_data['adr'].nunique() #column good
```

```
8879
```

```
hotel_data.columns
```

```
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')
```

```
hotel_data['required_car_parking_spaces'].unique() #column good
```

```
array([0, 1, 2, 8, 3])
```

```
hotel_data['total_of_special_requests'].unique()  #column good
```

```
array([0, 2, 1, 3, 4, 5])
```

```
hotel_data['reservation_status'].unique()
```

```
array(['No-Show', 'Canceled', 'Check-Out'], dtype=object)
```

```
hotel_data['reservation_status_date'].unique()
```

```
       '2016-07-25', '2016-02-15', '2016-08-06', '2016-07-16',
       '2016-08-27', '2016-08-17', '2016-08-23', '2016-09-04',
       '2016-09-03', '2016-09-18', '2017-06-19', '2017-07-15',
       '2017-08-14', '2017-08-21', '2017-08-13', '2017-03-26',
       '2017-08-06', '2015-07-09', '2015-05-19', '2016-06-05',
       '2016-09-11', '2017-08-20', '2016-09-24', '2015-02-02',
       '2015-02-24', '2015-02-26', '2016-08-21', '2017-08-27',
       '2015-06-14', '2015-04-25', '2017-04-16', '2017-07-23',
       '2015-06-18', '2015-07-28', '2017-02-12', '2015-12-21',
       '2015-12-25', '2015-12-31', '2016-12-03', '2016-04-24',
       '2016-05-07', '2016-06-11', '2016-06-18', '2017-03-01',
       '2017-07-08', '2016-08-28', '2016-12-25', '2017-05-07',
       '2017-05-27', '2017-07-22', '2017-08-12', '2017-09-03',
       '2017-09-04', '2017-09-06', '2017-09-07', '2017-09-10',
       '2017-09-12', '2015-12-24', '2017-09-05', '2015-01-21',
       '2015-03-03', '2015-04-02', '2015-04-28', '2015-06-17',
       '2015-06-30', '2014-10-17', '2015-01-01', '2015-01-30',
       '2015-03-23', '2015-06-23', '2015-04-22', '2015-04-15',
       '2015-05-01', '2015-03-25', '2015-04-11', '2015-06-19',
       '2015-06-01', '2015-03-17', '2015-05-22', '2015-05-12',
       '2015-04-23', '2015-03-28', '2015-05-14', '2015-04-20',
       '2015-06-04', '2015-04-14', '2015-06-08', '2015-05-27',
       '2015-04-03', '2015-02-11', '2015-02-12', '2015-02-20',
       '2015-02-25', '2015-02-27', '2015-03-04', '2015-03-06',
       '2015-03-31', '2015-04-04', '2015-04-10', '2015-04-29',
       '2015-05-16', '2015-06-27', '2017-09-09', '2017-09-08',
       '2017-09-14', '2015-04-16', '2015-05-06', '2015-05-18',
       '2015-05-26', '2015-05-13', '2015-03-30', '2015-05-07',
       '2015-04-13', '2015-04-24', '2015-04-06', '2015-06-13',
       '2015-04-17', '2015-05-15', '2015-05-09', '2015-06-06',
       '2015-05-30', '2015-03-24', '2015-05-21', '2015-04-07',
       '2015-04-18', '2015-01-28', '2015-01-29', '2015-02-05',
       '2015-02-06', '2015-02-09', '2015-02-10', '2015-02-19',
       '2015-02-23', '2015-03-09', '2015-03-11', '2015-03-12',
       '2015-03-18', '2015-04-08', '2015-05-08', '2015-04-30',
       '2015-04-21', '2015-04-05', '2015-03-13', '2015-05-05',
       '2015-03-29', '2015-06-10', '2015-04-27', '2015-01-20',
       '2015-02-17', '2015-03-10'], dtype=object)
```

```
hotel_data['reservation_status_date'] = pd.to_datetime(hotel_data['reservation_status_date']

# Sorting the DataFrame by the reservation_status_date column
hotel_data = hotel_data.sort_values('reservation_status_date')

# Now the reservation_status_date column is in datetime format and sorted
```

```
hotel_data['reservation_status_date'].unique()
```

```
<DatetimeArray>
['2014-10-17 00:00:00', '2014-11-18 00:00:00', '2015-01-01 00:00:00',
 '2015-01-02 00:00:00', '2015-01-18 00:00:00', '2015-01-20 00:00:00',
 '2015-01-21 00:00:00', '2015-01-22 00:00:00', '2015-01-28 00:00:00',
 '2015-01-29 00:00:00',
 ...
```

```
       '2017-09-03 00:00:00', '2017-09-04 00:00:00', '2017-09-05 00:00:00',
       '2017-09-06 00:00:00', '2017-09-07 00:00:00', '2017-09-08 00:00:00',
       '2017-09-09 00:00:00', '2017-09-10 00:00:00', '2017-09-12 00:00:00',
       '2017-09-14 00:00:00']
      Length: 926, dtype: datetime64[ns]
```

```
hotel_data = hotel_data.reset_index(drop=True)
```

```
hotel_data.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  int64
 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                        119390 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         119390 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                           119390 non-null  int64
 24  company                         119390 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  datetime64[ns]
dtypes: datetime64[ns](1), float64(3), int64(18), object(10)
memory usage: 29.1+ MB
```

```
hotel_data['children'].astype(int)
```

```
0         0
1         0
2         0
3         0
4         0
         ..
119385    0
119386    0
119387    0
119388    0
119389    0
```

```
# To make the changes permanent, assign the converted column back to the original DataFrame
# Assuming you want to overwrite the existing "children" column
hotel_data['children'] = hotel_data['children'].astype(int)
```

```
hotel_data['company']=hotel_data['company'].astype(int)
```

```
hotel_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
```