In [2]:

```
import pandas as pd
```

In [3]:

```
# Import the pandas library as a pd. Load dataset bookings.csv with separator ;.
# Check the table size, variable types and then output the first 7 rows to look at the data.
bookings = pd.read_csv("C:/Users/stask/Analitics_Karpov/Module2/Project hotel bookings/bookings.csv", sep=';')
bookings.shape
```

Out[3]:

```
(119390, 21)
```

In [4]:

```
# 4 Lowercase the column names and replace spaces with underscores.
bookings.columns = bookings.columns.str.replace(' ', '_').str.lower()
bookings.columns
```

Out[4]:

```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_full_date',
       'arrival_date_year', 'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'stays_total_nights', 'adults', 'children',
       'babies', 'meal', 'country', 'reserved_room_type', 'assigned_room_type',
       'customer_type', 'reservation_status', 'reservation_status_date'],
      dtype='object')
```

In [5]:

```
# 5 Which countries have had the highest number of successful bookings? Please indicate top 7.
bookings.query('is_canceled == 0').country.value_counts().head(7)
```

Out[5]:

```
PRT    21071
GBR     9676
FRA     8481
ESP     6391
DEU     6069
IRL     2543
ITA     2433
Name: country, dtype: int64
```

In [6]:

```
# 6 How many nights on average do different types of hotels get booked?
bookings.groupby('hotel', as_index=False) \
        .agg({'stays_total_nights': 'mean'}) \
        .round(2) \
        .rename(columns = {'stays_total_nights': 'stays_avg'})
```

Out[6]:

	hotel	stays_avg
0	City Hotel	2.98
1	Resort Hotel	4.32

In [7]:

```
# 7 How many such observations are encountered in the dataset? (wrongly assigned room)
bookings.query('reserved_room_type != assigned_room_type').shape[0]
```

Out[7]:

```
14917
```

8 Analyse the dates of the scheduled arrival:

- For which month were most successful bookings in 2016? Has the most popular month changed in 2017?
- Group the data by year and check for which month City Hotel bookings were most frequently cancelled in each of the periods

In [8]:

```
bookings.loc[(bookings.is_canceled == 1) & (bookings.arrival_date_year == 2016)] \
        .arrival_date_month.value_counts()
```

Out[8]:

```
October      2514
June         2096
April        2061
September    2022
May          1915
August       1825
November     1636
July         1499
March        1477
December     1398
February     1337
January       557
Name: arrival_date_month, dtype: int64
```

In [9]:

```
bookings.loc[(bookings.is_canceled == 1) & (bookings.arrival_date_year == 2017)] \
        .arrival_date_month.value_counts()
```

Out[9]:

```
May         2762
April       2463
June        2439
July        1984
August      1816
March       1672
February    1359
January     1250
Name: arrival_date_month, dtype: int64
```

In [10]:

```
bookings.query('hotel == "City Hotel" and is_canceled == 1') \
        .groupby(['arrival_date_year', 'arrival_date_month'], as_index=False) \
        .hotel.value_counts()
```

Out[10]:

	arrival_date_year	arrival_date_month	hotel	count
0	2015	August	City Hotel	1232
1	2015	December	City Hotel	668
2	2015	July	City Hotel	939
3	2015	November	City Hotel	301
4	2015	October	City Hotel	1321
5	2015	September	City Hotel	1543
6	2016	April	City Hotel	1539
7	2016	August	City Hotel	1247
8	2016	December	City Hotel	1072
9	2016	February	City Hotel	930
10	2016	January	City Hotel	438
11	2016	July	City Hotel	1043
12	2016	June	City Hotel	1720
13	2016	March	City Hotel	1108
14	2016	May	City Hotel	1436
15	2016	November	City Hotel	1360
16	2016	October	City Hotel	1947
17	2016	September	City Hotel	1567
18	2017	April	City Hotel	1926
19	2017	August	City Hotel	1123
20	2017	February	City Hotel	971
21	2017	January	City Hotel	1044
22	2017	July	City Hotel	1324
23	2017	June	City Hotel	1808
24	2017	March	City Hotel	1278
25	2017	May	City Hotel	2217

In [13]:

```
# Look at the numerical characteristics of the three variables:
#     adults, children and babies.
# Which one has the highest average value?
bookings[['adults', 'children', 'babies']].mean()
```

Out[13]:

```
adults      1.856403
children    0.103890
babies      0.007949
dtype: float64
```

In [21]:

```
# Create a total_kids column by combining children and babies.
# On average, which type of hotel is more popular with customers with children?
bookings['total_kids'] = bookings.children + bookings.babies
bookings.groupby('hotel', as_index=False).agg({'total_kids':'mean'}).round(2)
```

Out[21]:

	hotel	total_kids
0	City Hotel	0.10
1	Resort Hotel	0.14

Create a variable has_kids that takes True if the client has at least one child (total_kids), otherwise it takes False. Calculate the churn rate as a percentage of the total number of customers. Indicate among which group the rate is higher.

In [29]:

```
bookings['has_kids'] = bookings.total_kids >= 1
```

In [42]:

```
churn_rate_general = round(bookings.query('is_canceled == 1').shape[0] / bookings.shape[0], 2)
churn_rate_has_kids = round(bookings.query('is_canceled == 1 and has_kids').shape[0]\
                            / bookings.query('has_kids').shape[0], 2)
churn_rate_has_not_kids = round(bookings.query('is_canceled == 1 and has_kids == False').shape[0]\
                            / bookings.query('has_kids == False').shape[0], 2)
print(churn_rate_general, churn_rate_has_kids, churn_rate_has_not_kids)
```

```
0.37 0.35 0.37
```