Hotel Booking Data Analysis

May 21, 2020

ENGR350

Introduction Data Science with Python Term - Project

Hotel Booking Dataset Analysis

Banu Yobaş

Fırat Tamur - Kutay Eroğlu

May 21, 2020

[1]:
```
# importing data libraries
import numpy as np
import pandas as pd

# statistics libraries
from scipy import stats

# importing visualization libraries
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[2]:
```
data = pd.read_csv('../Data/hotel_bookings.csv')
```

[3]:
```
data.shape
```

[3]:
```
(119390, 32)
```

[4]:
```
data.head()
```

[4]:
```
          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]
```

[5]:
```
data.describe()
```

[5]:
```
         is_canceled      lead_time  arrival_date_year  \
count  119390.000000  119390.000000      119390.000000
mean        0.370416     104.011416        2016.156554
std         0.482918     106.863097           0.707476
min         0.000000       0.000000        2015.000000
25%         0.000000      18.000000        2016.000000
50%         0.000000      69.000000        2016.000000
75%         1.000000     160.000000        2017.000000
max         1.000000     737.000000        2017.000000
```

```
       arrival_date_week_number  arrival_date_day_of_month  \
count             119390.000000              119390.000000
mean                  27.165173                  15.798241
std                   13.605138                   8.780829
min                    1.000000                   1.000000
25%                   16.000000                   8.000000
50%                   28.000000                  16.000000
75%                   38.000000                  23.000000
max                   53.000000                  31.000000

       stays_in_weekend_nights  stays_in_week_nights         adults  \
count            119390.000000         119390.000000  119390.000000
mean                  0.927599              2.500302       1.856403
std                   0.998613              1.908286       0.579261
min                   0.000000              0.000000       0.000000
25%                   0.000000              1.000000       2.000000
50%                   1.000000              2.000000       2.000000
75%                   2.000000              3.000000       2.000000
max                  19.000000             50.000000      55.000000

            children         babies  is_repeated_guest  \
count  119386.000000  119390.000000      119390.000000
mean        0.103890       0.007949           0.031912
std         0.398561       0.097436           0.175767
min         0.000000       0.000000           0.000000
25%         0.000000       0.000000           0.000000
50%         0.000000       0.000000           0.000000
75%         0.000000       0.000000           0.000000
max        10.000000      10.000000           1.000000

       previous_cancellations  previous_bookings_not_canceled  \
count           119390.000000                   119390.000000
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

       booking_changes          agent      company  days_in_waiting_list  \
count    119390.000000  103050.000000  6797.000000         119390.000000
mean          0.221124      86.693382   189.266735              2.321149
std           0.652306     110.774548   131.655015             17.594721
min           0.000000       1.000000     6.000000              0.000000
25%           0.000000       9.000000    62.000000              0.000000
```

```
50%           0.000000      14.000000   179.000000              0.000000
75%           0.000000     229.000000   270.000000              0.000000
max          21.000000     535.000000   543.000000            391.000000

                 adr  required_car_parking_spaces  total_of_special_requests
count  119390.000000                119390.000000              119390.000000
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
```

[6]:
```
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  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
```

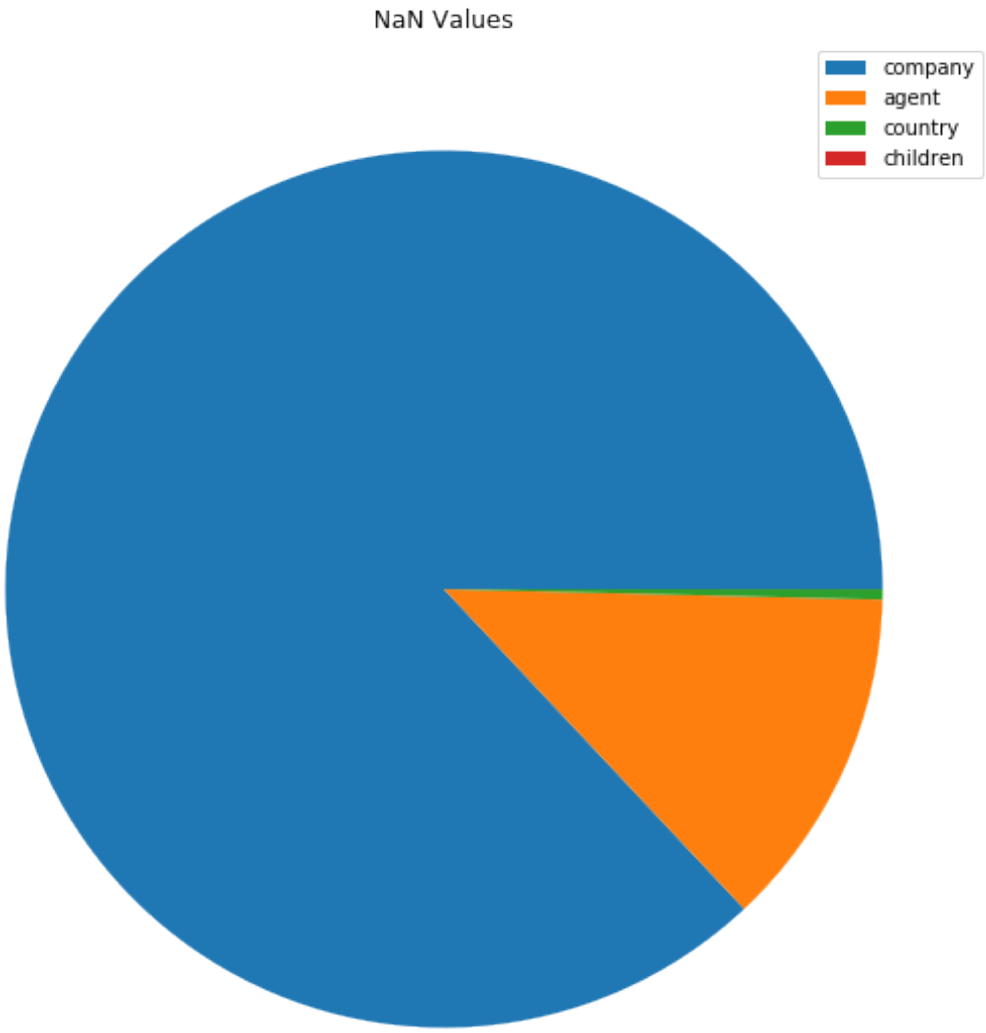
```
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

[7]:
```
# nan values

total = data.isnull().sum().sort_values(ascending=False)
percent = (data.isnull().sum() / data.isnull().count()).
 ↪sort_values(ascending=False)

nan_cols = pd.concat([total, percent], axis=1, keys=['Total', 'nan'])

nan_cols = nan_cols[nan_cols['nan'] > 0]

nan_cols
```

[7]:

	Total	nan
company	112593	0.943069
agent	16340	0.136862
country	488	0.004087
children	4	0.000034

[8]:
```
plt.figure(figsize=(10, 10))

plt.pie(nan_cols.reset_index()['Total'])

plt.title("NaN Values")
plt.legend(labels=nan_cols.reset_index()['index'])
plt.show()
```



Handling with Null Data

[9]:
```
# We will drop columns 'company' and 'agent' because, there are too many null␣
 ↪values for that columns

data.drop(['company', 'agent'], axis=1, inplace=True)
```

[10]:
```
# Reviewing rows with null country values
data[data['country'].isnull()]
```

[10]:

	hotel	is_canceled	lead_time	arrival_date_year	\
30	Resort Hotel	0	118	2015	

```
4127    Resort Hotel            1          0               2016
7092    Resort Hotel            1          8               2016
7860    Resort Hotel            1         39               2016
8779    Resort Hotel            1          0               2016
…              …          …         …                …
65908     City Hotel            1          0               2017
65909     City Hotel            1          0               2017
65910     City Hotel            1          0               2017
80830     City Hotel            0          4               2015
101488    City Hotel            0          1               2016

       arrival_date_month  arrival_date_week_number  \
30                   July                        27
4127             February                         8
7092                 July                        30
7860               August                        36
8779              October                        42
…                    …                       …
65908              April                        15
65909              April                        15
65910              April                        15
80830           November                        48
101488          November                        47

        arrival_date_day_of_month  stays_in_weekend_nights  \
30                              1                       4
4127                           15                       0
7092                           21                       0
7860                           30                       0
8779                           13                       0
…                             …                      …
65908                          10                       0
65909                          10                       0
65910                          10                       0
80830                          23                       1
101488                         13                       2

        stays_in_week_nights  adults  …  assigned_room_type  \
30                        10       1  …                   A
4127                       0       0  …                   P
7092                       1       1  …                   A
7860                       5       2  …                   A
8779                       1       1  …                   A
…                        …     …  …                 …
65908                      0       0  …                   P
65909                      0       0  …                   P
65910                      0       0  …                   P
```

```
80830                     2       1  …                   A
101488                    2       2  …                   A

        booking_changes deposit_type days_in_waiting_list    customer_type  \
30                    2   No Deposit                    0        Transient
4127                  0   No Deposit                    0        Transient
7092                  0   No Deposit                    0        Transient
7860                  0   No Deposit                    0        Transient
8779                  0   No Deposit                    0        Transient
…                   …          …                     …              …
65908                 0   No Deposit                    0        Transient
65909                 0   No Deposit                    0        Transient
65910                 0   No Deposit                    0        Transient
80830                 0   No Deposit                    0  Transient-Party
101488                0   No Deposit                    0            Group

          adr  required_car_parking_spaces  total_of_special_requests  \
30       62.0                            0                          2
4127      0.0                            0                          0
7092     73.0                            0                          2
7860    159.0                            0                          5
8779     50.0                            0                          0
…        …                             …                        …
65908     0.0                            0                          0
65909     0.0                            0                          0
65910     0.0                            0                          0
80830    70.0                            0                          0
101488  105.0                            0                          1

        reservation_status reservation_status_date
30               Check-Out              2015-07-15
4127              Canceled              2016-02-15
7092              Canceled              2016-07-20
7860              Canceled              2016-07-22
8779              Canceled              2016-10-13
…                    …                     …
65908             Canceled              2017-04-10
65909             Canceled              2017-04-10
65910             Canceled              2017-04-10
80830            Check-Out              2015-11-26
101488           Check-Out              2016-11-17

[488 rows x 30 columns]
```

[11]:
```
# Reviewing rows with null children values
data[data['children'].isnull()]
```

[11]:
```
            hotel  is_canceled  lead_time  arrival_date_year  \
40600  City Hotel            1          2               2015
40667  City Hotel            1          1               2015
40679  City Hotel            1          1               2015
41160  City Hotel            1          8               2015

      arrival_date_month  arrival_date_week_number  arrival_date_day_of_month  \
40600             August                        32                          3
40667             August                        32                          5
40679             August                        32                          5
41160             August                        33                         13

       stays_in_weekend_nights  stays_in_week_nights  adults  …  \
40600                        1                    0       2  …
40667                        0                    2       2  …
40679                        0                    2       3  …
41160                        2                    5       2  …

       assigned_room_type  booking_changes deposit_type days_in_waiting_list  \
40600                   B                0   No Deposit                    0
40667                   B                0   No Deposit                    0
40679                   B                0   No Deposit                    0
41160                   B                0   No Deposit                    0

         customer_type   adr  required_car_parking_spaces  \
40600  Transient-Party  12.0                            0
40667  Transient-Party  12.0                            0
40679  Transient-Party  18.0                            0
41160  Transient-Party  76.5                            0

       total_of_special_requests  reservation_status reservation_status_date
40600                          1            Canceled              2015-08-01
40667                          1            Canceled              2015-08-04
40679                          2            Canceled              2015-08-04
41160                          1            Canceled              2015-08-09

[4 rows x 30 columns]
```

[12]:
```
# There are 488 rows which have null country values and 4 rows which have null␣
 ↪children values
# These rows are not esential in terms of our purpose, therefore we will be␣
 ↪dropping them.

data.drop(data[data['country'].isnull()].index, axis=0, inplace=True)
data.drop(data[data['children'].isnull()].index, axis=0, inplace=True)
```

[13]:
```
total = data.isnull().sum().sort_values(ascending=False)
percent = (data.isnull().sum() / data.isnull().count()).
 ↪sort_values(ascending=False)

nan_cols = pd.concat([total, percent], axis=1, keys=['Total', 'nan'])

nan_cols = nan_cols[nan_cols['nan'] > 0]

nan_cols
```

[13]:
```
Empty DataFrame
Columns: [Total, nan]
Index: []
```

[14]:
```
def plot_canceling_prob(col_name: str, data: pd.DataFrame):
    """
    Displays canceling probabilities for categorical data.
    """

    plt.figure(figsize=(16, 8))

    x = data.groupby('is_canceled')[col_name].value_counts(sort=True,␣
 ↪normalize=True)[1].keys().values
    y = data.groupby('is_canceled')[col_name].value_counts(sort=True,␣
 ↪normalize=True)[1].values
    leg = data.groupby('is_canceled')[col_name].value_counts(normalize=True,␣
 ↪sort=True)[1].values


    g = sns.barplot(x, y)
    g.set(title=f'Canceled Booking Distribution on {col_name}')

    plt.legend(leg)
    plt.show(g)
```

[15]:
```
def count_cat_prob_plot(col_name: str, data: pd.DataFrame):

    g1 = sns.countplot(x=col_name, data=data)
    plt.title(f"Count Plot for {col_name}")
    plt.show(g1)

    g2 = sns.catplot(x=col_name, y='is_canceled', data=data, kind='bar',␣
 ↪aspect=3)
    plt.title(f"Canceling Probabilities for each {col_name}")
    plt.show(g2)

    plot_canceling_prob(col_name, data)
```

[16]:
```
# Looking at the overall data, after handling with null data

data.shape
```

[16]:
```
(118898, 30)
```

Data Analysis and Visualizations

Data Exploration for Canceling a Booking:

- Section **??**
- Section **??**
-
-
-
-
-
-
-
-

0.0.1 Analysis out of Canceling:

- Family size vs Country
- Hotel occupancy rate depending weekdays vs weeekend
-
-
-
-
-
-
-

0.0.2 All columns list:

- Section **??** Section **??**, Section **??** , Section **??**, Section **??**, Section **??**

[17]:
```
data.columns
```

[17]:
```
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',
       'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

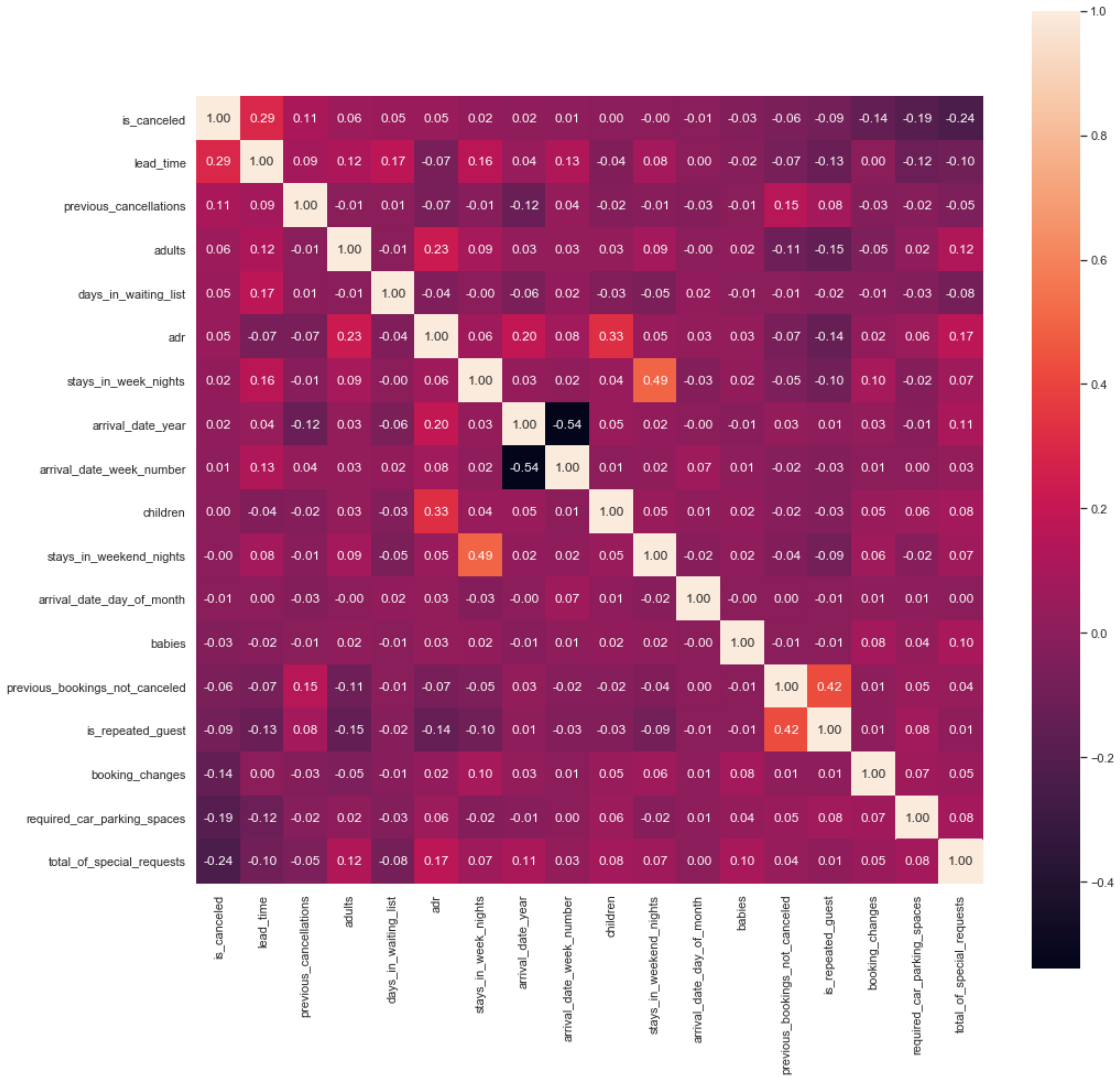
[18]:
```
# seaborn initial settings

sns.set(context='notebook', palette='Set1', style='whitegrid', rc={'figure.
↪figsize':(16, 8)})
```

[19]:
```
columns_to_remove = list()
columns_to_dummy = list()
```

[20]:
```
# keep analysis for each feature

analysis = {}

for col in data.columns:
    analysis[col] = []
```

0.0.3 0 - Correlation HeatMap: - Section ??

[21]:
```
data_corr = data.corr()

column = 'is_canceled'
corr_cols = data.shape[1]

cols = data_corr.nlargest(corr_cols, column)[column].index
coef = data_corr.nlargest(corr_cols, column)[cols].values

plt.figure(figsize=(16, 16))

g = sns.heatmap(coef, cbar=True, annot=True, square=True, fmt='.2f',
                yticklabels=cols.values, xticklabels=cols.values)
```



Most correlated columns with cancelation:

- lead_time
- previous_cancelations
- adults
- days_in_waiting_list
- adr
- stays_date_week_nights
- arrival_date_years

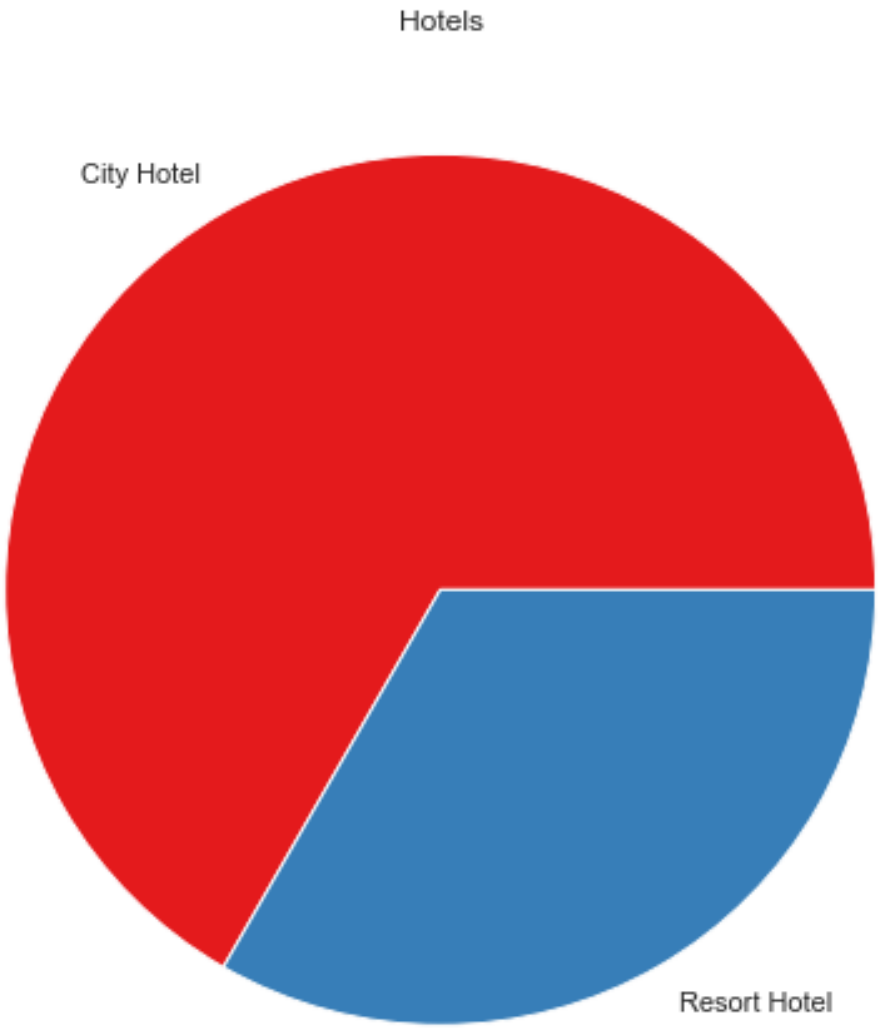
0.0.4 1 - hotel: - Section ??

- hotel name info.

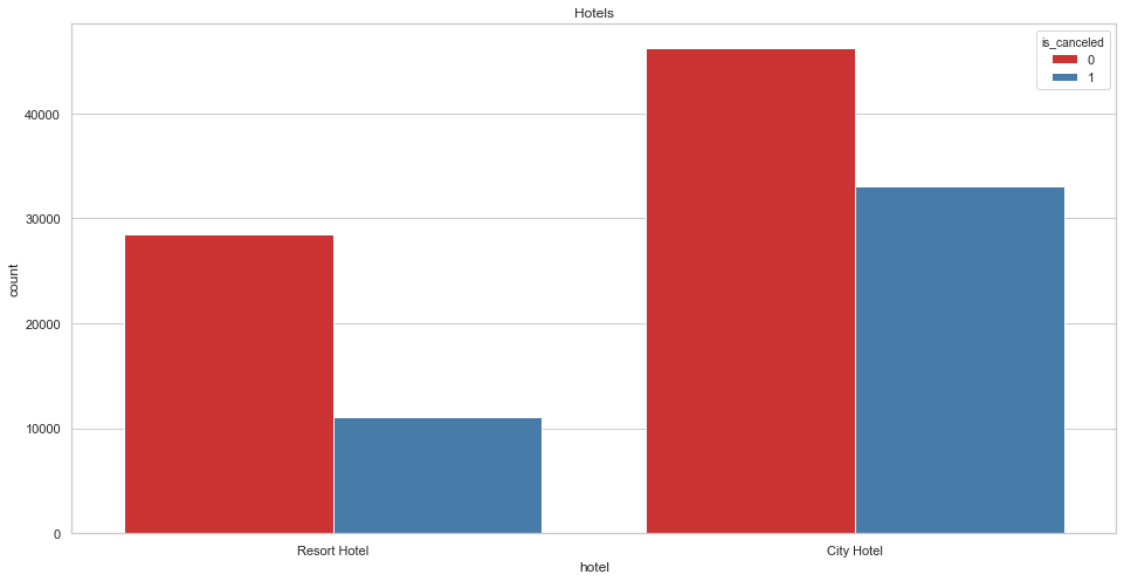
[22]:
```
data['hotel'].unique()
```

[22]:
```
array(['Resort Hotel', 'City Hotel'], dtype=object)
```

[23]:
```
plt.pie(data['hotel'].value_counts().values, labels=data['hotel'].
↪value_counts().keys())

plt.title("Hotels")
plt.show()
```



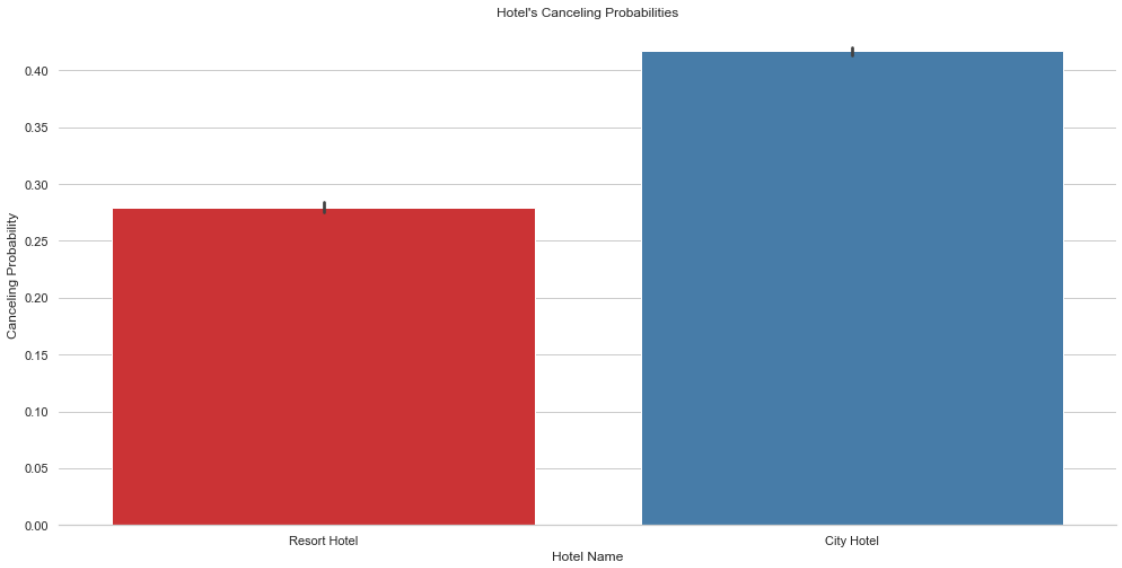
[24]:
```
g = sns.countplot(x='hotel', hue='is_canceled', data=data)

g.set_title("Hotels")

plt.show(g)
```



- City hotel's has more bookings than resort hotel. Also, cancellation rate of City Hotel is higher than Resort Hotel.

[25]:
```
analysis['hotel'].append('City hotel has more bookings and higher cancellation␣
↪rates.')
```

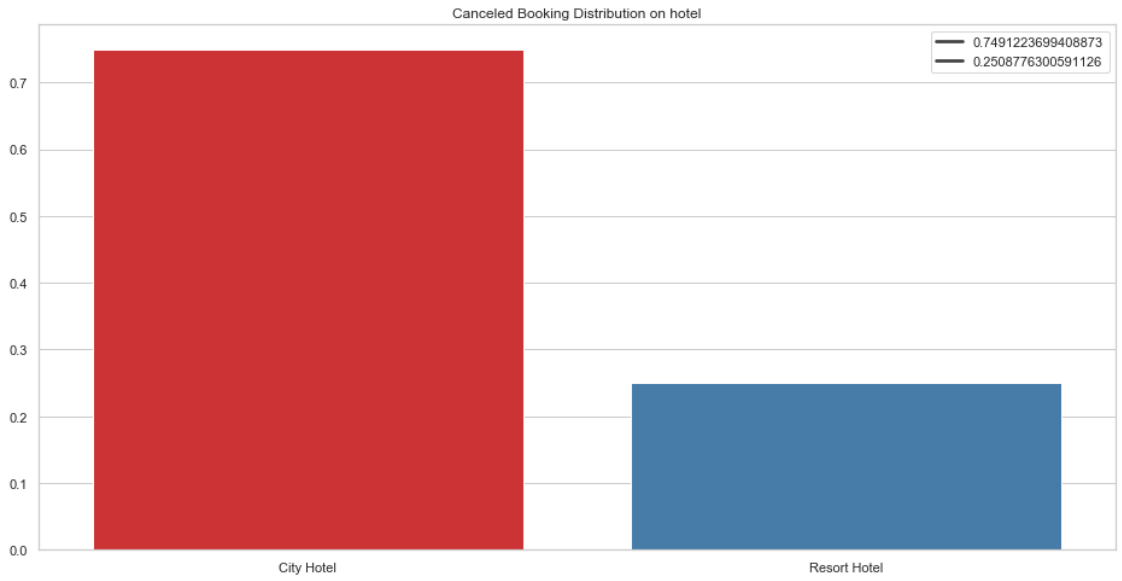
[26]:
```
g = sns.catplot(x='hotel', y='is_canceled', data=data, kind='bar', height=7,␣
↪aspect=2)

g.despine(left=True) # removes axis line. Here removes y axis line.

g.set(xlabel='Hotel Name', ylabel='Canceling Probability', title="Hotel's␣
↪Canceling Probabilities")

plt.show(g)
```



- Customers who booked to City hotel more likely to cancel their bookings

[27]:
```
analysis['hotel'].append('Customers who booked to City hotel more likely to␣
↪cancel their bookings.')
```

[28]:
```
plot_canceling_prob('hotel', data)
```



- In total, City hotel has more canceled bookings. This may be due to City hotel's higher number of bookings compared to Resort Hotel.

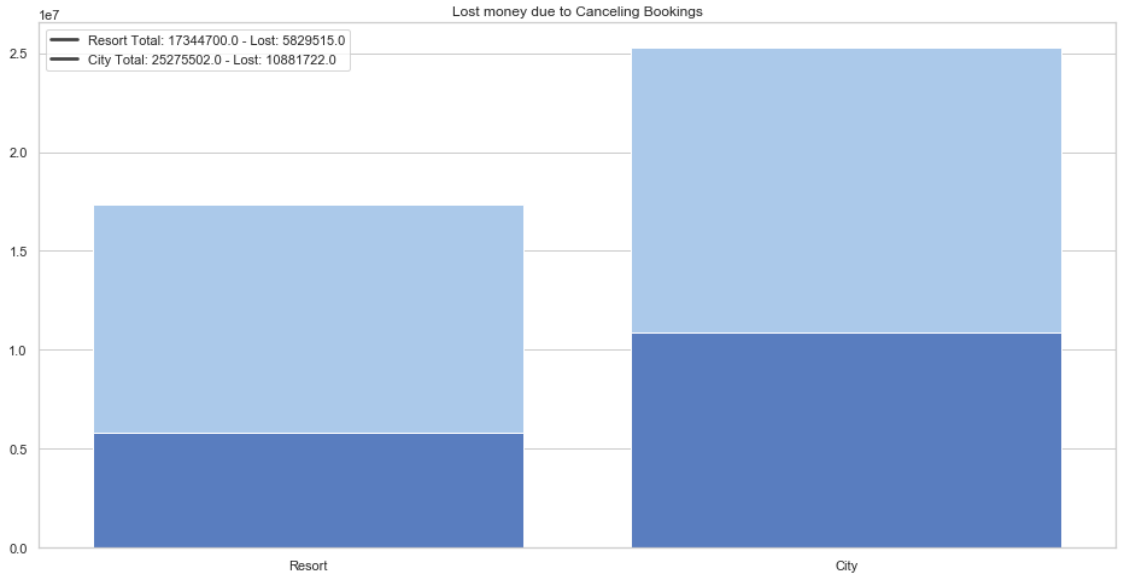
[29]:
```
analysis['hotel'].append("In total, City hotel has more canceled bookings. This␣
↪because City hotel's higher number of bookings compared to Resort Hotel.")
```

[30]:
```
#Displaying final indidual analysis on hotel

analysis['hotel']
```

[30]:
```
['City hotel has more bookings and higher cancellation rates.',
 'Customers who booked to City hotel more likely to cancel their bookings.',
 "In total, City hotel has more canceled bookings. This because City hotel's
higher number of bookings compared to Resort Hotel."]
```

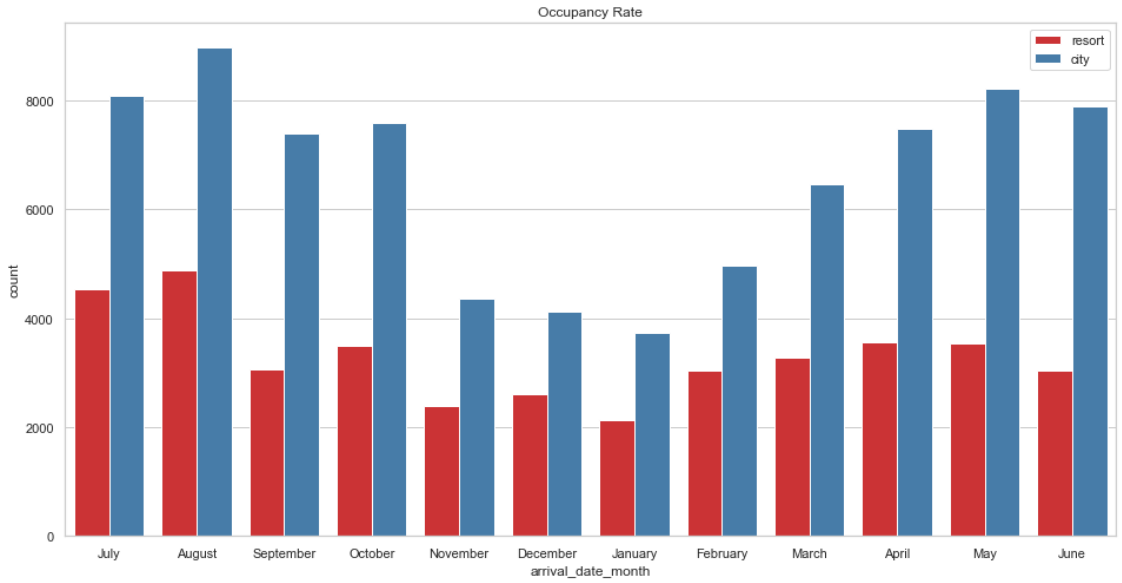
[31]:
```
# Total lost money due to canceling booking for each hotel


resort = data[data['hotel'] == 'Resort Hotel'].copy()
city = data[data['hotel'] == 'City Hotel'].copy()

resort['total_stays'] = resort['stays_in_week_nights'] +␣
↪resort['stays_in_weekend_nights']
city['total_stays'] = city['stays_in_week_nights'] +␣
↪city['stays_in_weekend_nights']

resort['customer_total_payment'] = resort['adr'].values * resort['total_stays'].
↪values
city['customer_total_payment'] = city['adr'] * city['total_stays']

resort_lost_revenue = resort[resort['is_canceled'] ==␣
↪1]['customer_total_payment'].sum()
city_lost_revenue = city[city['is_canceled'] == 1]['customer_total_payment'].
↪sum()

resort_total_revenue = resort['customer_total_payment'].sum()
city_total_revenue = city['customer_total_payment'].sum()

sns.set_color_codes("pastel")
g = sns.barplot(x=['Resort', 'City'], y=[resort_total_revenue,␣
↪city_total_revenue], color='b')
sns.set_color_codes("muted")
g = sns.barplot(x=['Resort', 'City'], y=[resort_lost_revenue,␣
↪city_lost_revenue], color='b')


plt.legend([f'Resort Total: {round(resort_total_revenue)} - Lost:␣
↪{round(resort_lost_revenue)}',
```

```
            f'City Total: {round(city_total_revenue)} - Lost:␣
 ↪{round(city_lost_revenue)}'])
plt.title('Lost money due to Canceling Bookings')
plt.show()
```

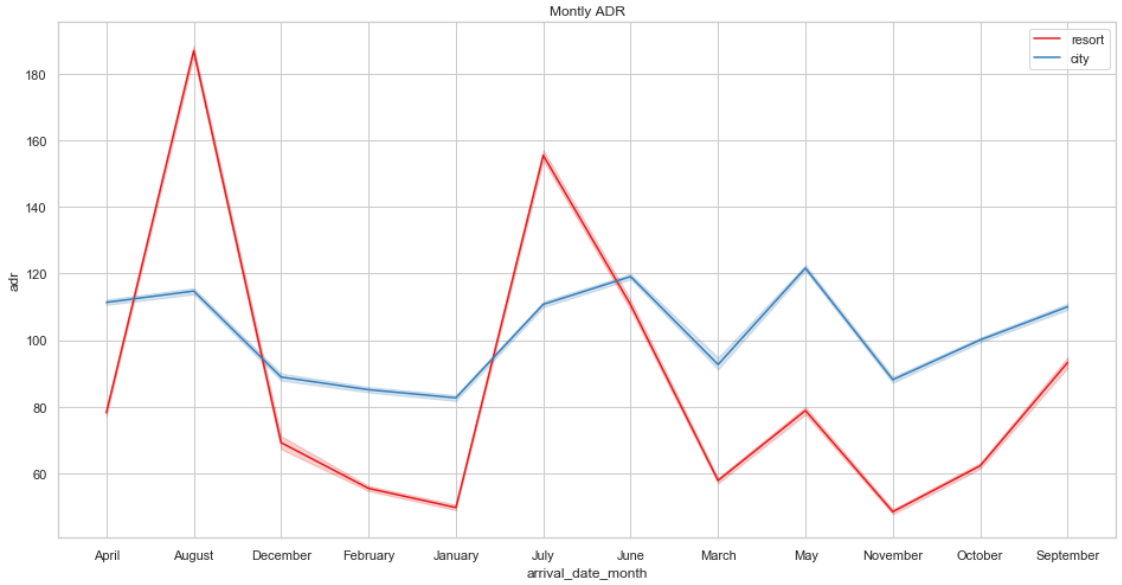


There is a huge lost for hotels due to canceling bookings.

[32]:
```
g = sns.countplot(x='arrival_date_month', data=data, hue='hotel')

plt.title("Occupancy Rate")
plt.legend(['resort', 'city'])
plt.show(g)
```



[33]:
```
data['total_stays'] = data['stays_in_week_nights'] +␣
 ↪data['stays_in_weekend_nights']
data['customer_total_payment'] = data['adr'] * data['total_stays']


months = ['', 'January', 'February', 'March', 'April', 'May', 'June', 'July',␣
 ↪'August', 'September',
              'October', 'November', 'December']

g = sns.lineplot(x='arrival_date_month', y='adr', data=data, hue='hotel',␣
 ↪color='r')

plt.title("Montly ADR")
plt.legend(['resort', 'city'])
plt.show(g)
```



[34]:
```
columns_to_dummy.append('hotel')
```

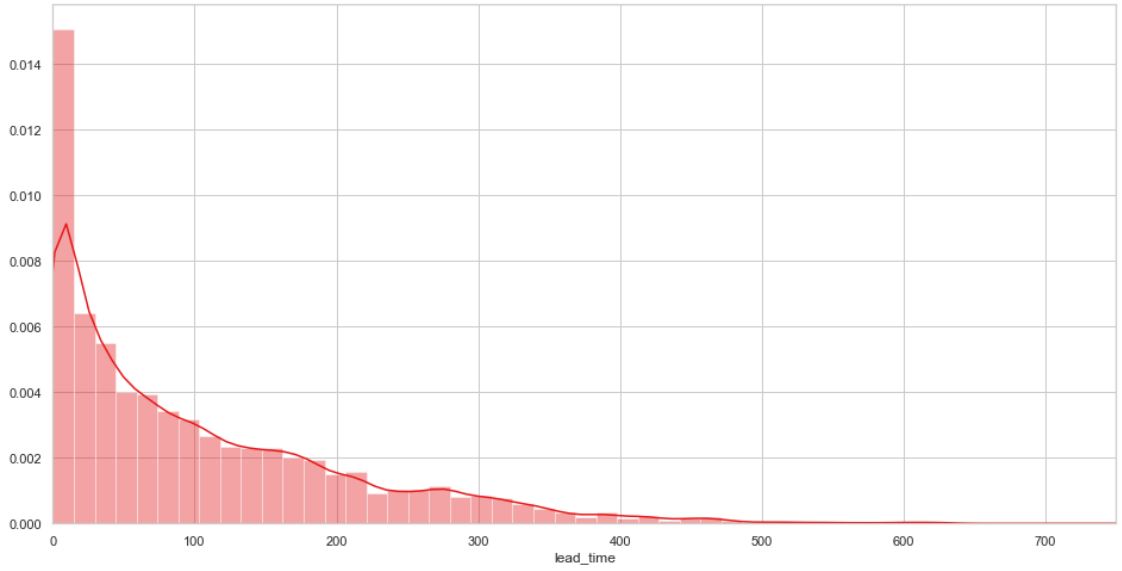
0.0.5 2 - lead_time: - Section ??

- Number of days that elapsed between the entering date of the booking into the PMS and the ar

[35]:
```
data['lead_time'].describe()
```

[35]:
```
count    118898.000000
mean        104.311435
std         106.903309
min           0.000000
25%          18.000000
50%          69.000000
75%         161.000000
max         737.000000
Name: lead_time, dtype: float64
```

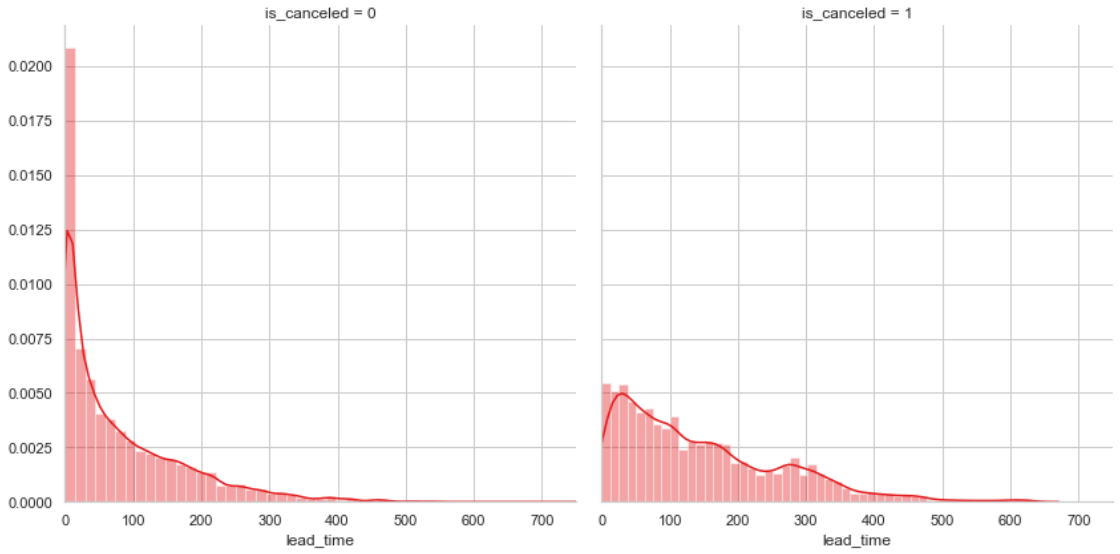
[36]:
```
g = sns.distplot(a=data['lead_time'], label='lead_time_distribution')

plt.xlim([0, 750])
plt.show(g)
```



- We see that there is a positive skewness in the lead time.
- Most of booking planned for a close time.

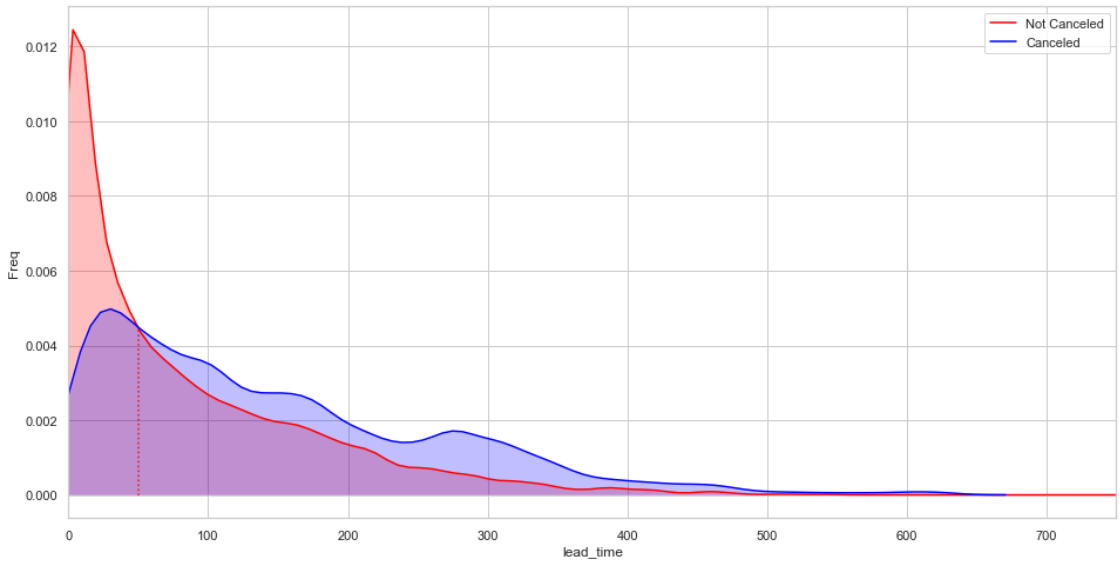
[37]:
```
analysis['lead_time'].append('We see that there is a positive skewness in the␣
↪lead time.')
```

[38]:
```
# lead_time vs canceled

g = sns.FacetGrid(data, col='is_canceled', height=6)
g = g.map(sns.distplot, 'lead_time')

plt.xlim([0, 750])
plt.show(g)
```



- Most of not canceled bookings have a short lead time comparing to canceled bookings.

[39]:
```
# lead_time dist vs is_canceled

g = sns.kdeplot(data['lead_time'][data['is_canceled'] == 0],
                color='Red', shade=True)

g = sns.kdeplot(data['lead_time'][data['is_canceled'] == 1],
                color='Blue', shade=True)

g.set_xlabel('lead_time')
g.set_ylabel('Freq')

g = g.legend(['Not Canceled', 'Canceled'])

plt.plot([50, 50], [0.00, 0.0045], ':')
plt.xlim([0, 750])
plt.show(g)
```



- Long term bookings more likely to be canceled.
- High lead time causes high canceling probability.

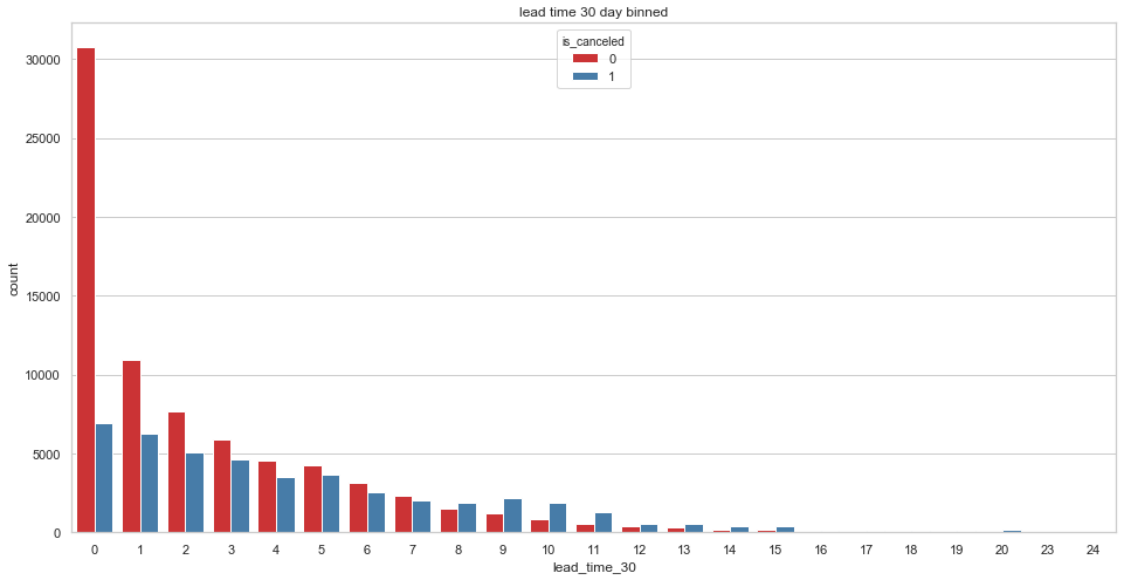
[40]:
```
analysis['lead_time'].append('High lead time causes high canceling probability.
 ↪')
```

[41]:
```
data['lead_time'].min(), data['lead_time'].mean(), data['lead_time'].max()
```

[41]:
```
(0, 104.31143501152248, 737)
```

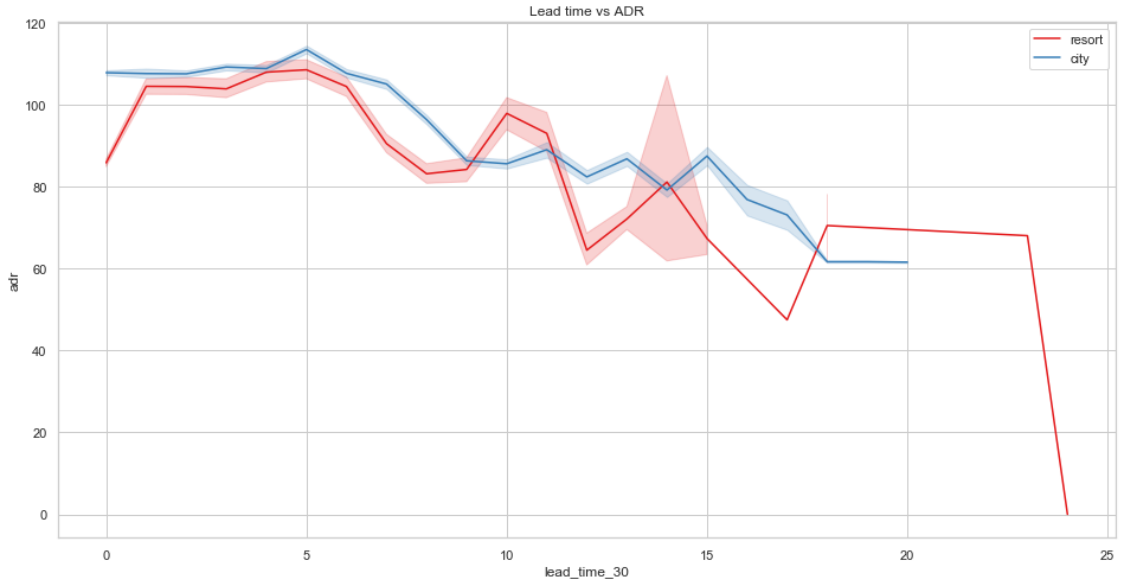
[42]:
```
# we can use binning method to convert lead time in day to months

data['lead_time_30'] = data['lead_time'] // 30
data['lead_time_60'] = data['lead_time'] // 60
data['lead_time_120'] = data['lead_time'] // 120
data['lead_time_360'] = data['lead_time'] // 360
```

[43]:
```
# respect to 30 days binning.
g = sns.countplot(x='lead_time_30', hue='is_canceled', data=data)

plt.title('lead time 30 day binned')
plt.show(g)
```



- Bookings are maded for 7 months later more likely to be canceled.

[44]:
```
# prices according to lead_time

# we will use monthly lead_time

g = sns.lineplot(x='lead_time_30', y='adr', data=data, hue='hotel',␣
 ↪markers=True, dashes=False)

plt.title("Lead time vs ADR")
plt.legend(['resort', 'city'])
plt.show(g)
```



- Increase in lead time decreases average daily room rate

[45]:
```
analysis['lead_time'].append('Bookings are maded for 7 months later more likely␣
↪to be canceled.')
```

[46]:
```
countries_lead_time = data.groupby('country')['lead_time'].sum().
↪reset_index(name = 'Total Lead Time')
```

[47]:
```
# Lead time averages by countries

import plotly.express as px

px.choropleth(countries_lead_time,
                    locations = "country",
                    color= "Total Lead Time",
                    hover_name= "Total Lead Time",
                    color_continuous_scale=px.colors.sequential.Oranges,
                    title="Lead Time by Countries")
```

[48]:
```
columns_to_remove.extend(['lead_time_60', 'lead_time_30', 'lead_time_120',␣
↪'lead_time_360'])
```

[49]:
```
#Showcasing final analysis on lead_time

analysis['lead_time']
```

[49]:
```
['We see that there is a positive skewness in the lead time.',
 'High lead time causes high canceling probability.',
```

```
'Bookings are maded for 7 months later more likely to be canceled.']
```

0.0.6 3 - arrival_date_year: - Section ??

[50]:
```
data['arrival_date_year'].describe()
```

[50]:
```
count    118898.000000
mean       2016.157656
std           0.707459
min        2015.000000
25%        2016.000000
50%        2016.000000
75%        2017.000000
max        2017.000000
Name: arrival_date_year, dtype: float64
```

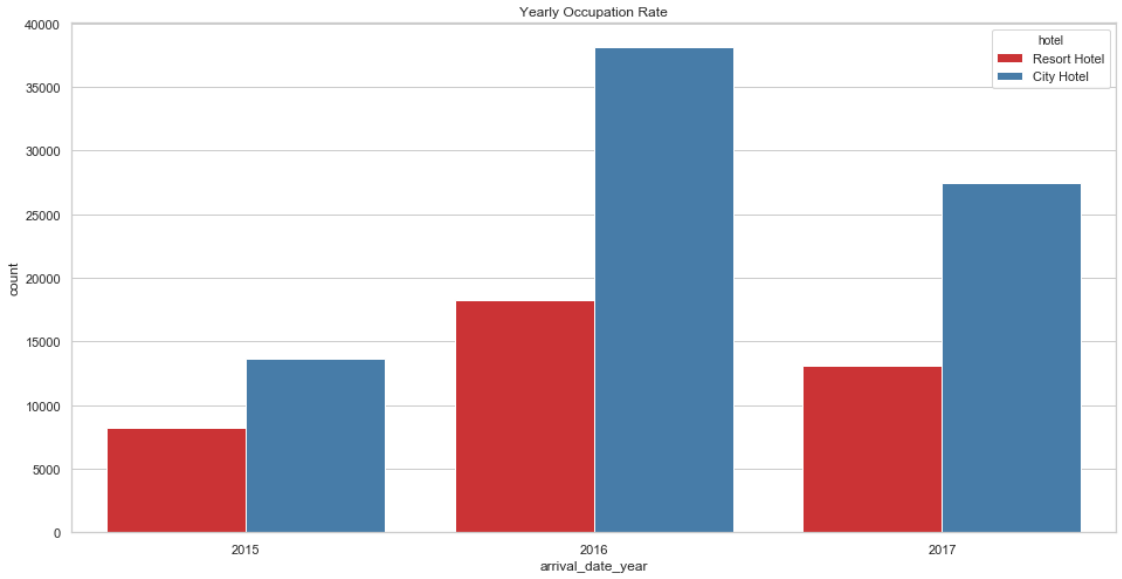
[51]:
```
data['arrival_date_year'].unique()
```

[51]:
```
array([2015, 2016, 2017])
```

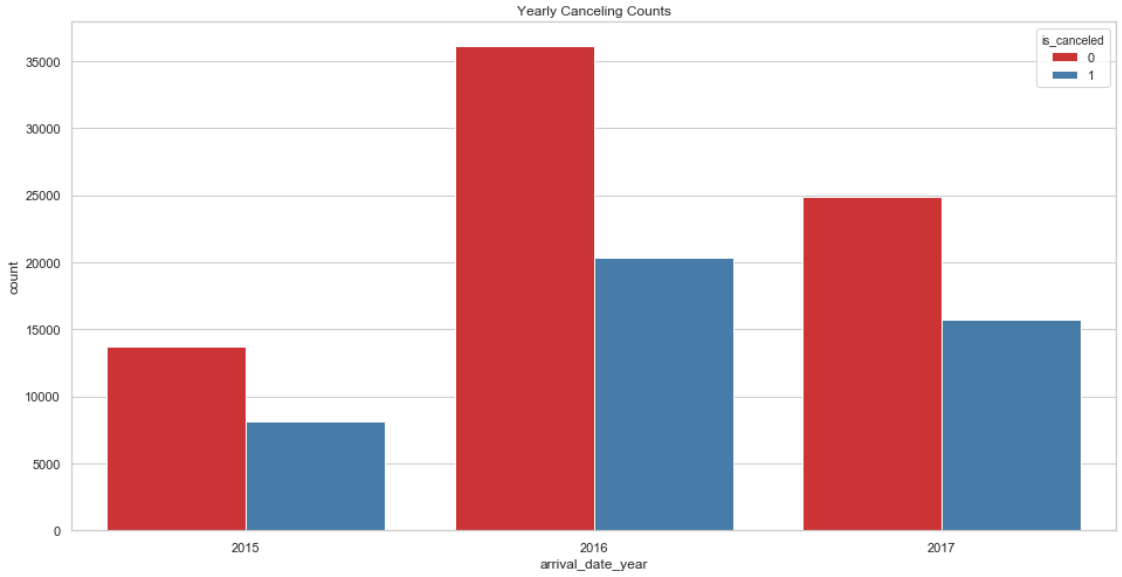
[52]:
```
data['arrival_date_year'].value_counts()
```

[52]:
```
2016    56435
2017    40604
2015    21859
Name: arrival_date_year, dtype: int64
```

[53]:
```
g1 = sns.countplot(x='arrival_date_year', hue='hotel', data=data)

plt.title('Yearly Occupation Rate')
plt.show(g1)
```



[54]:
```
g2 = sns.countplot(x='arrival_date_year', hue='is_canceled', data=data)

plt.title('Yearly Canceling Counts')
plt.show(g2)
```

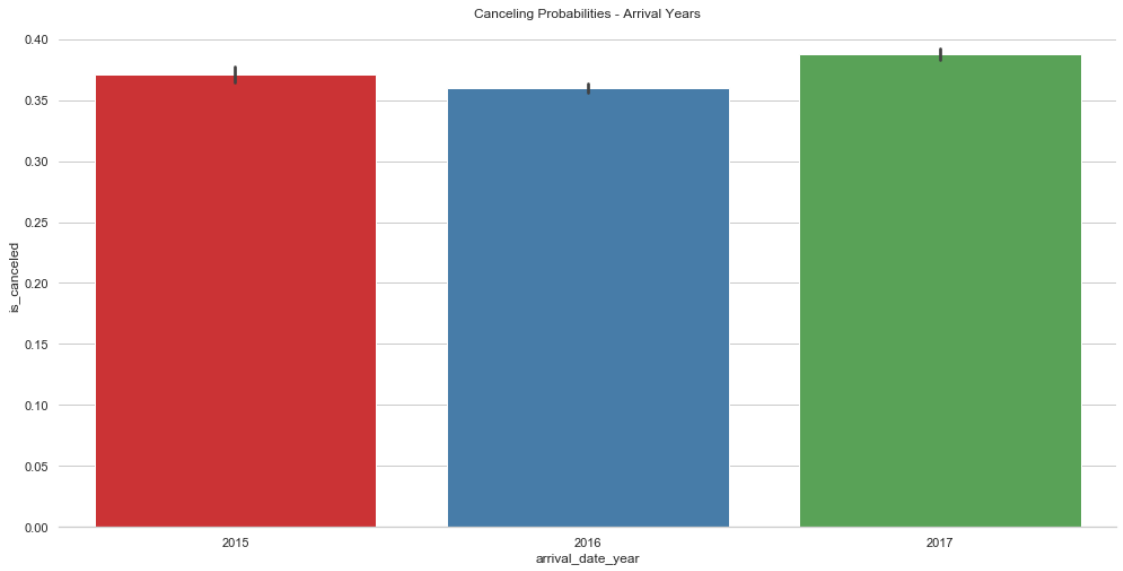


- The highest number of booking belongs to 2016 then 2017 and 2015

[55]:
```
analysis['arrival_date_year'].append('The highest number of booking belongs to␣
↪2016 then 2017 and 2015.')
```

[56]:
```
g = sns.catplot(x='arrival_date_year', y='is_canceled', data=data, kind='bar',␣
↪height=7, aspect=2)

g.despine(left=True)

g.set(title='Canceling Probabilities - Arrival Years')
```

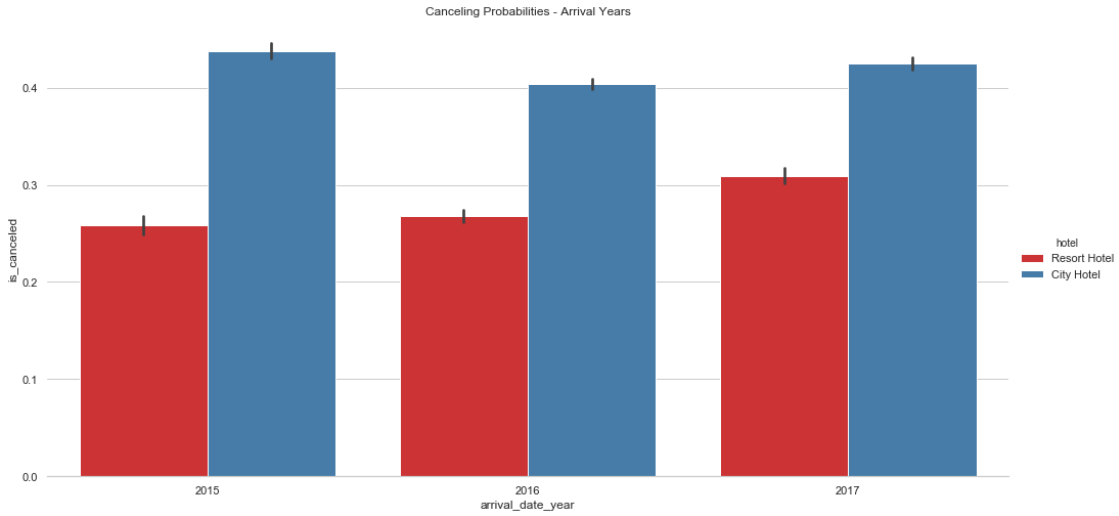
[56]:
```
<seaborn.axisgrid.FacetGrid at 0x1a1f0c1450>
```



- 2015 - 2016 - 2017 have similar canceling probabilities.

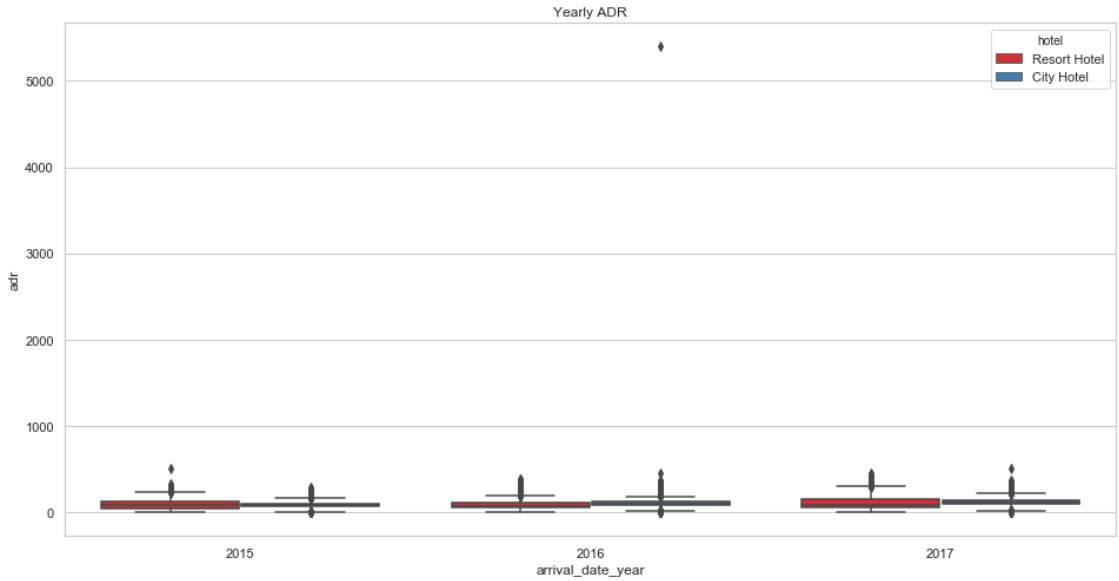
[57]:
```
analysis['arrival_date_year'].append('2015 - 2016 - 2017 have similar canceling␣
↪probabilities.')
```

[58]:
```
g = sns.catplot(x='arrival_date_year', y='is_canceled', hue='hotel', data=data,␣
↪kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Canceling Probabilities - Arrival Years')

plt.show(g)
```



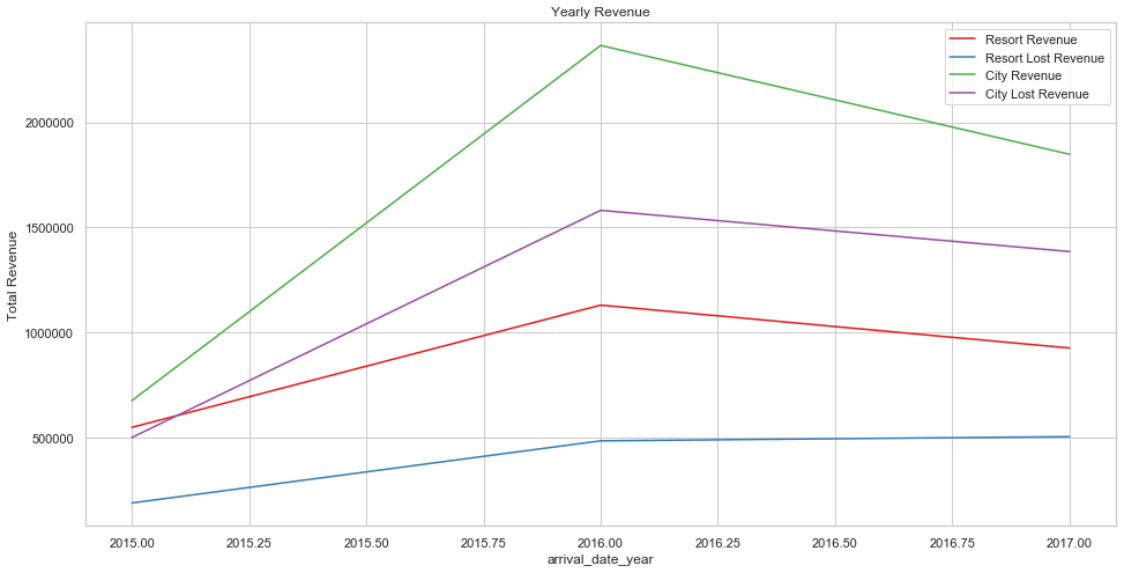
- In each year canceling probability is higher for City Hotel.

[59]:
```
g = sns.boxplot(x='arrival_date_year', y='adr', data=data, hue='hotel')

plt.title("Yearly ADR")
plt.show(g)
```



- Resort hotel made some discount on its room rate in 2016. City hotel increased it's rooms rate every year.

[60]:
```
## Revenue Total and lost by years

yearly_revenue_city = data[(data['hotel'] == 'City Hotel') &␣
 ↪(data['is_canceled'] == 0)].groupby('arrival_date_year')['adr'].sum().
 ↪reset_index(name = 'Total Revenue')
yearly_revenue_resort = data[(data['hotel'] == 'Resort Hotel') &␣
 ↪(data['is_canceled'] == 0)].groupby('arrival_date_year')['adr'].sum().
 ↪reset_index(name = 'Total Revenue')

yearly_revenue_city_cancel = data[(data['hotel'] == 'City Hotel') &␣
 ↪(data['is_canceled'] == 1)].groupby('arrival_date_year')['adr'].sum().
 ↪reset_index(name = 'Total Revenue')
yearly_revenue_resort_cancel = data[(data['hotel'] == 'Resort Hotel') &␣
 ↪(data['is_canceled'] == 1)].groupby('arrival_date_year')['adr'].sum().
 ↪reset_index(name = 'Total Revenue')



g = sns.lineplot(x='arrival_date_year', y='Total Revenue',␣
 ↪data=yearly_revenue_resort)
g = sns.lineplot(x='arrival_date_year', y='Total Revenue',␣
 ↪data=yearly_revenue_resort_cancel)

g = sns.lineplot(x='arrival_date_year', y='Total Revenue',␣
 ↪data=yearly_revenue_city)
g = sns.lineplot(x='arrival_date_year', y='Total Revenue',␣
 ↪data=yearly_revenue_city_cancel)



plt.title("Yearly Revenue")
plt.legend(['Resort Revenue', 'Resort Lost Revenue', 'City Revenue', 'City Lost␣
 ↪Revenue'])
plt.show(g)
```



- Yearly Revenue of the both hotel increased in 2016.

[61]:
```
analysis['arrival_date_year'].append('In each year canceling probability is␣
↪higher for Ciy Hotel.')
```

[62]:
```
columns_to_dummy.append('arrival_date_year')
```

[63]:
```
#Showcasing final analysis on arrival_date_year

analysis['arrival_date_year']
```

[63]:
```
['The highest number of booking belongs to 2016 then 2017 and 2015.',
 '2015 - 2016 - 2017 have similar canceling probabilities.',
 'In each year canceling probability is higher for Ciy Hotel.']
```

0.0.7 4 - arrival__date__week__number: - Section ??

[64]:
```
data['arrival_date_week_number'].describe()
```

[64]:
```
count    118898.000000
mean         27.166555
std          13.589971
min           1.000000
25%          16.000000
50%          28.000000
75%          38.000000
max          53.000000
Name: arrival_date_week_number, dtype: float64
```

[65]:
```
data['arrival_date_week_number'].value_counts()[:10]
```

[65]:
```
33    3571
30    3080
32    3039
34    3038
18    2909
21    2849
28    2845
17    2804
20    2783
29    2757
Name: arrival_date_week_number, dtype: int64
```

[66]:
```
# we have info for arrival month that's why we can use week info to detect␣
↪which week in a month they arrived.

data['arrival_date_weekth_in_month'] = data['arrival_date_week_number'] % 4
```

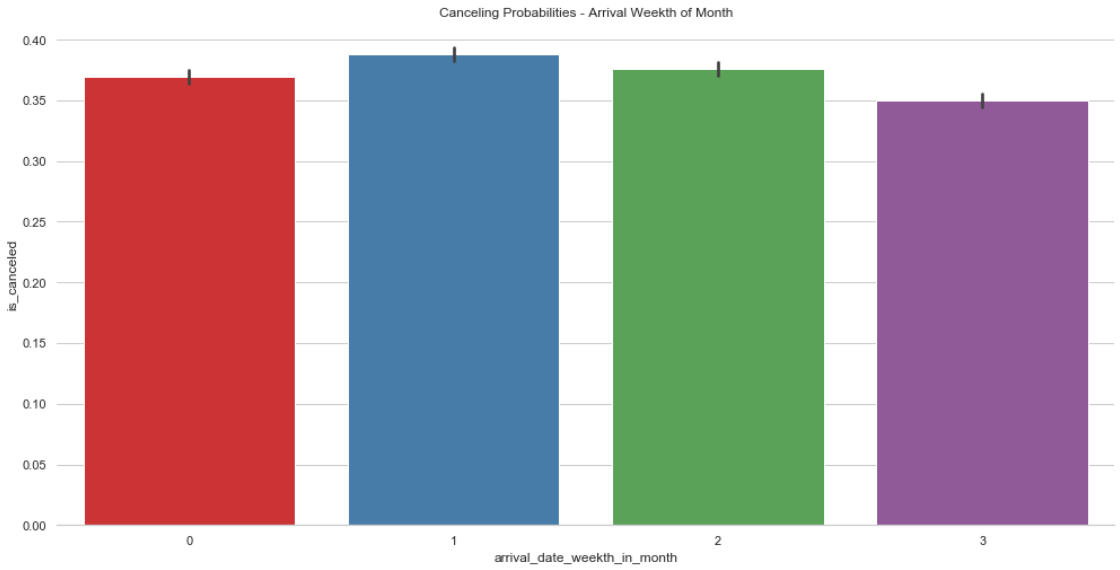
[67]:
```
data['arrival_date_weekth_in_month'].describe()
```

[67]:
```
count    118898.000000
mean          1.487569
std           1.102394
min           0.000000
25%           1.000000
50%           1.000000
75%           2.000000
max           3.000000
Name: arrival_date_weekth_in_month, dtype: float64
```

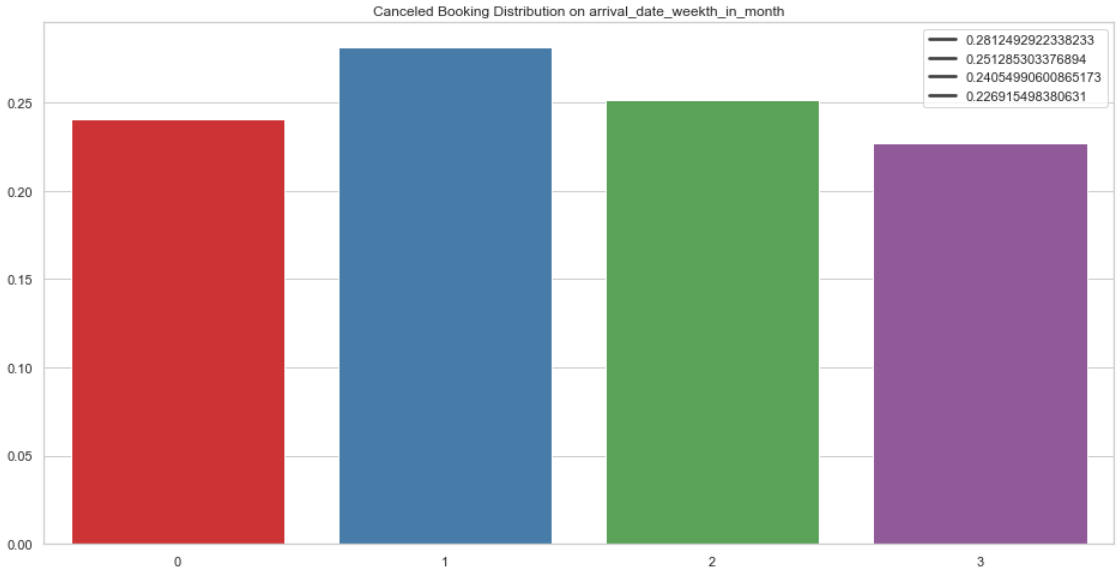
[68]:
```
data['arrival_date_weekth_in_month'].value_counts(sort=False)
```

[68]:
```
0    28772
1    32004
2    29501
3    28621
Name: arrival_date_weekth_in_month, dtype: int64
```

[69]:
```
g = sns.catplot(x='arrival_date_weekth_in_month', y='is_canceled', data=data,␣
↪kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Canceling Probabilities - Arrival Weekth of Month')

plt.show(g)
```



[70]:
```
plot_canceling_prob('arrival_date_weekth_in_month', data)
```

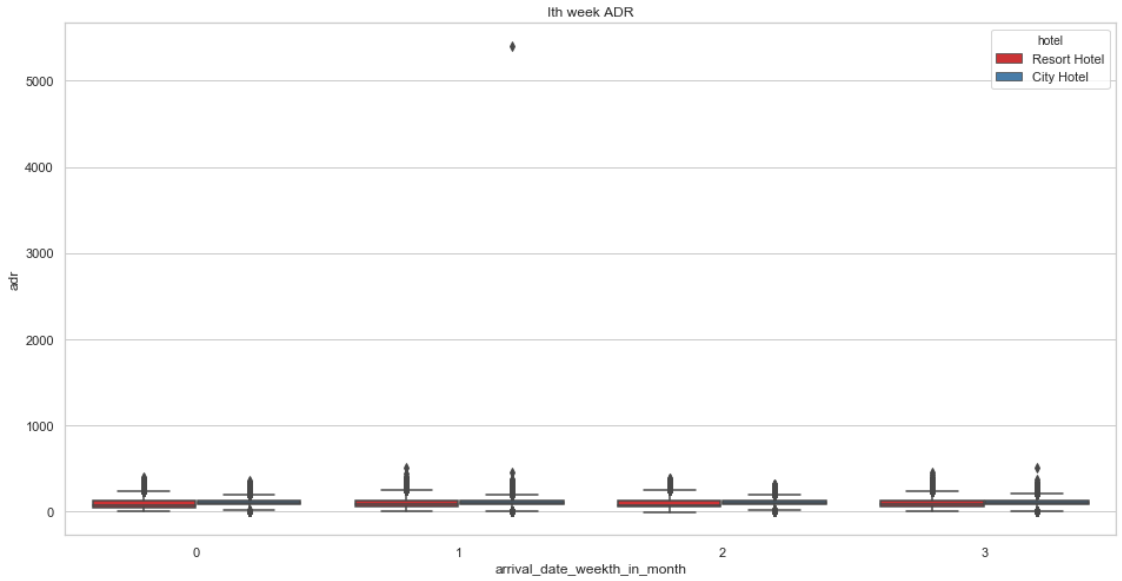


- Canceling probability is high for weeks 1st and 2nd.

[71]:
```
analysis['arrival_date_week_number'].append('Canceling probability is high for␣
↪weeks 1st and 2nd.')
```

[72]:
```
g = sns.boxplot(x='arrival_date_weekth_in_month', y='adr', data=data,␣
↪hue='hotel')
```

```
plt.title("Ith week ADR")
plt.show(g)
```



[73]:
```
columns_to_dummy.append('arrival_date_weekth_in_month')
```

[74]:
```
#Showcasing final analysis on arrival_date_week_number
analysis['arrival_date_week_number']
```

[74]:
```
['Canceling probability is high for weeks 1st and 2nd.']
```

0.0.8 5 - arrival_date_month: - Section ??

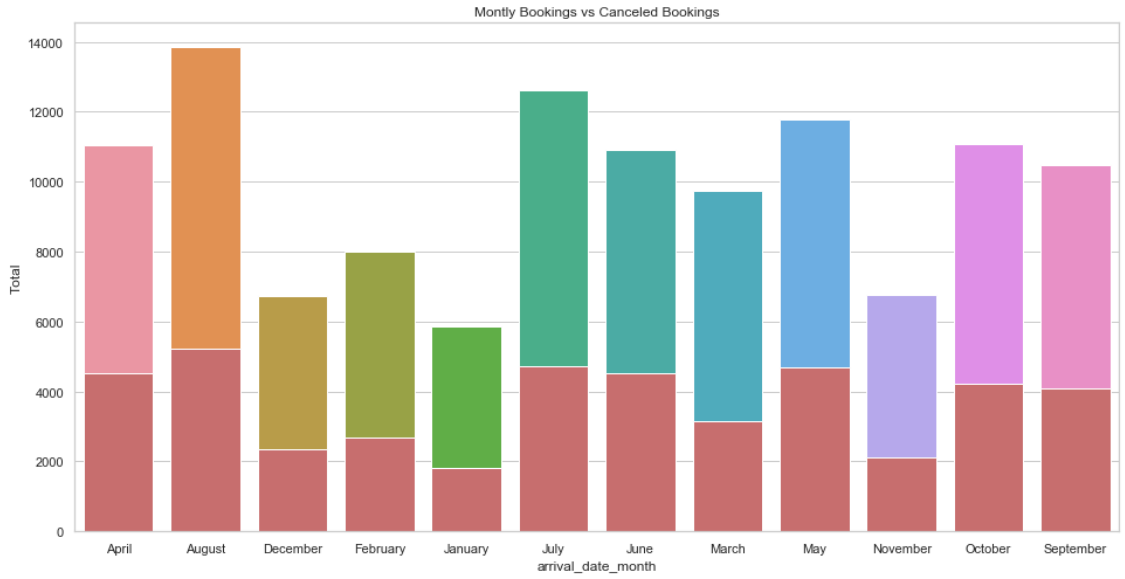
[75]:
```
data['arrival_date_month'].describe()
```

[75]:
```
count     118898
unique        12
top       August
freq       13852
Name: arrival_date_month, dtype: object
```

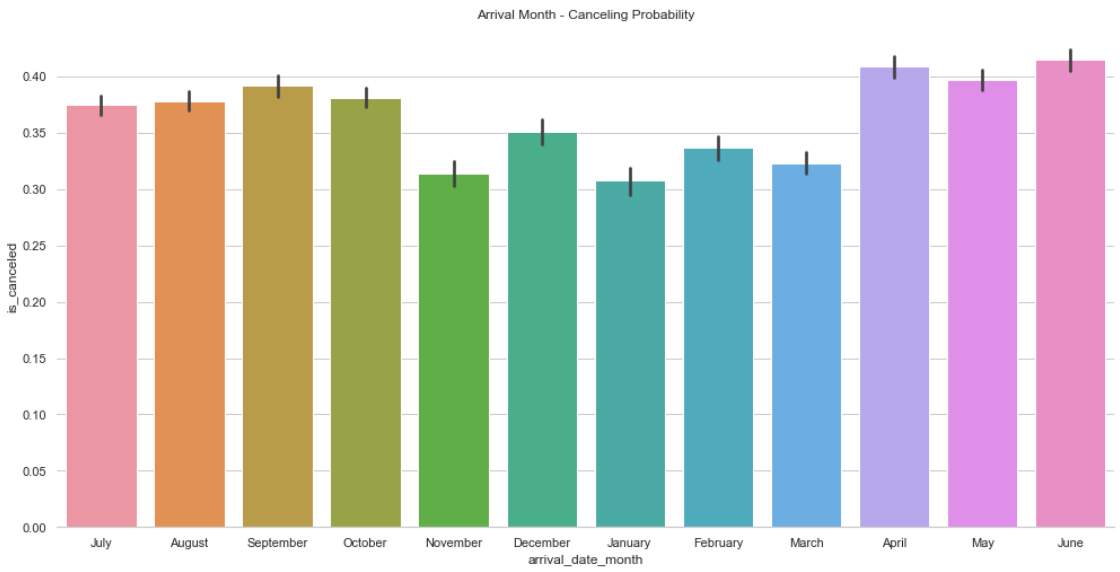
[76]:
```
data['arrival_date_month'].unique()
```

[76]:
```
array(['July', 'August', 'September', 'October', 'November', 'December',
       'January', 'February', 'March', 'April', 'May', 'June'],
      dtype=object)
```

[77]:
```
canceled_months = data[data['is_canceled'] == 1].groupby('arrival_date_month').
 ↪size().reset_index(name='Total')
total_months = data.groupby('arrival_date_month').size().
 ↪reset_index(name='Total')

g = sns.barplot(x='arrival_date_month', y='Total', data=total_months)
g = sns.barplot(x='arrival_date_month', y='Total', data=canceled_months,␣
 ↪color='r')

plt.title("Montly Bookings vs Canceled Bookings")
plt.show()
```



[78]:
```
g = sns.catplot(x='arrival_date_month', y='is_canceled', data=data, kind='bar',␣
 ↪height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Month - Canceling Probability')

plt.show(g)
```



- We have higher canceling probabilities for summer.

[79]:
```
analysis['arrival_date_month'].append('We have higher canceling probabilities␣
↪for summer.')
```

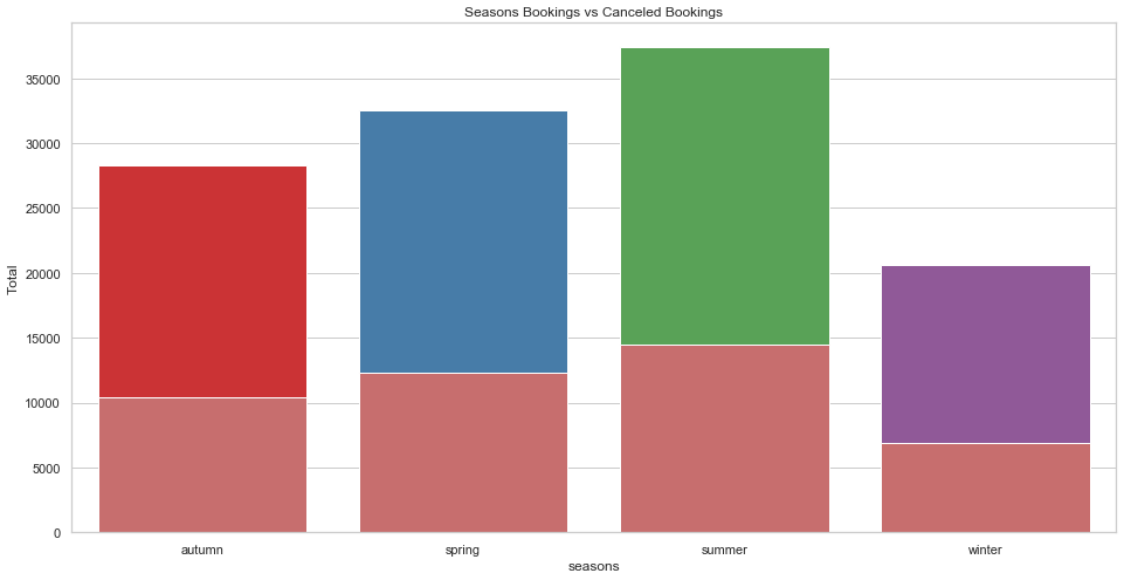
[80]:
```
# we may try to generate season data.

def month_to_season(month):

    if month in ['June', 'July', 'August']:
        return "summer"
    elif month in ['March', 'April', 'May']:
        return "spring"
    elif month in ['October', 'November', 'September']:
        return "autumn"
    else:
        return "winter"
```

[81]:
```
data['seasons'] = data['arrival_date_month'].apply(month_to_season)
```

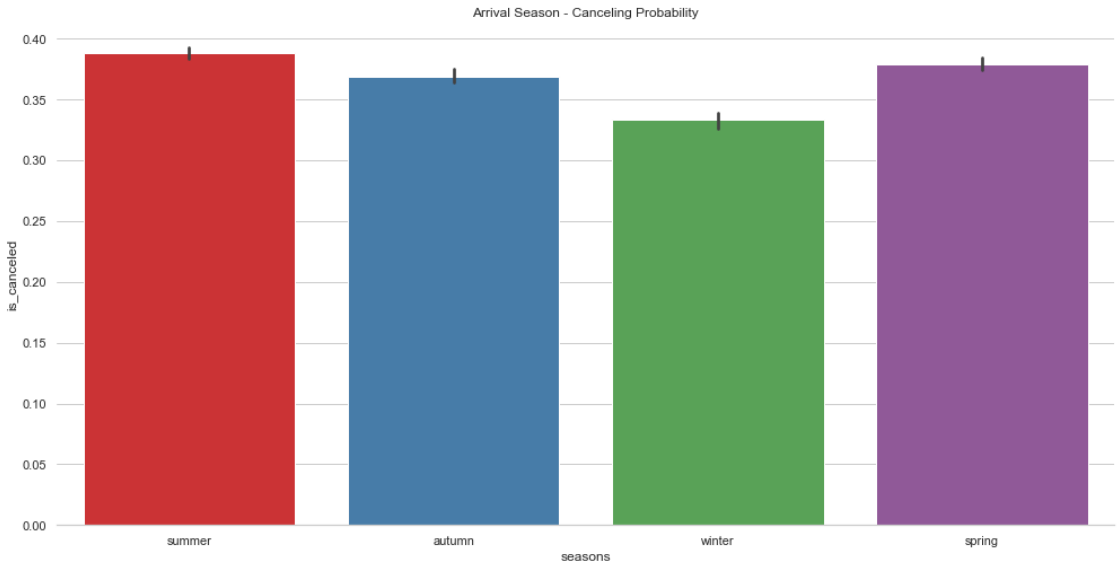
[82]:
```
data['seasons'].value_counts()
```

[82]:
```
summer    37407
spring    32563
autumn    28314
winter    20614
Name: seasons, dtype: int64
```

[83]:
```
canceled_seasons = data[data['is_canceled'] == 1].groupby('seasons').size().
↪reset_index(name='Total')
total_seasons = data.groupby('seasons').size().reset_index(name='Total')

g = sns.barplot(x='seasons', y='Total', data=total_seasons)
g = sns.barplot(x='seasons', y='Total', data=canceled_seasons, color='r')

plt.title("Seasons Bookings vs Canceled Bookings")
plt.show()
```



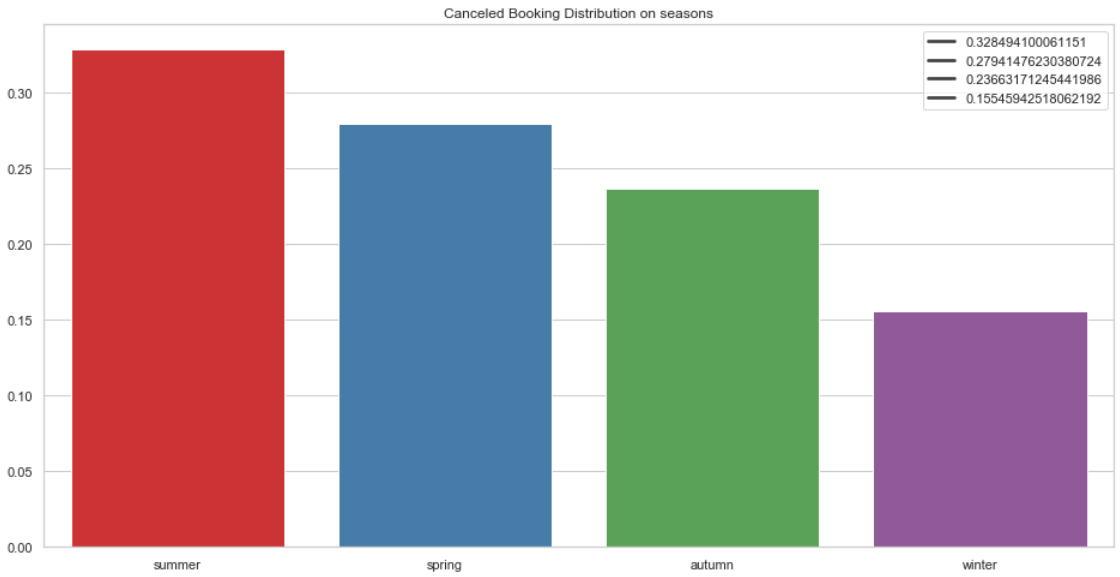
[84]:
```
g = sns.catplot(x='seasons', y='is_canceled', data=data, kind='bar', height=7,␣
↪aspect=2)

g.despine(left=True)

g.set(title='Arrival Season - Canceling Probability')

plt.show(g)
```



- Lowest Cancel Probability is for winter season.

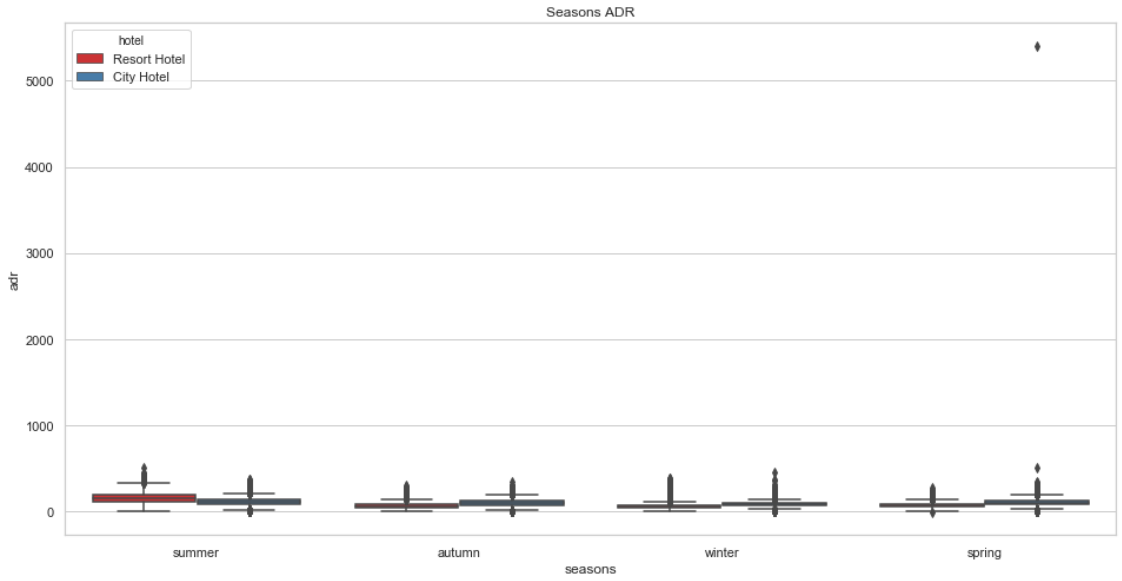
[85]:
```
analysis['arrival_date_month'].append('Lowest Cancel Probability is for winter␣
↪season.')
```

[86]:
```
plot_canceling_prob('seasons', data)
```

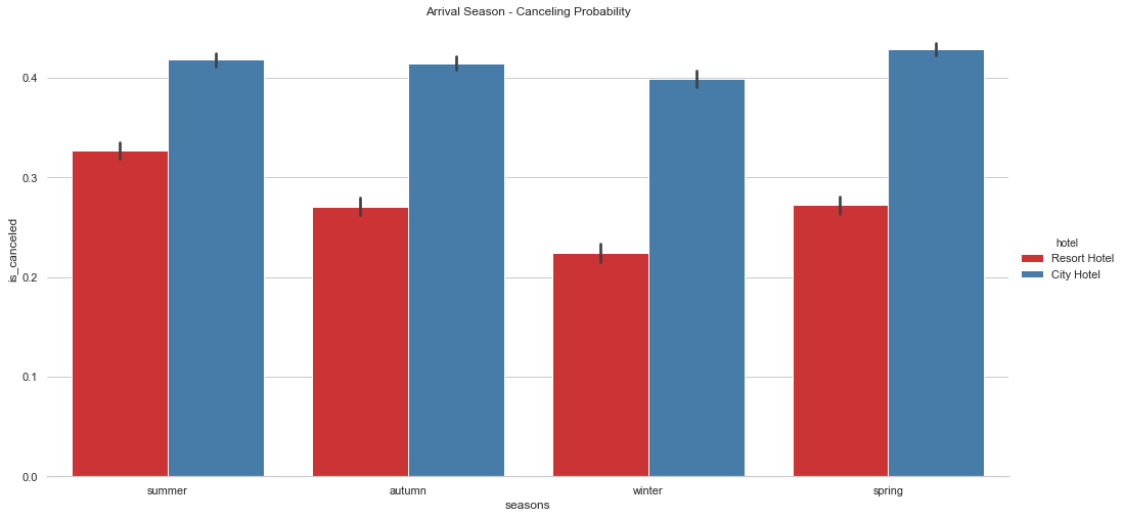


[87]:
```
g = sns.boxplot(x='seasons', y='adr', data=data, hue='hotel')
```

```
plt.title("Seasons ADR")
plt.show(g)
```



[88]:
```
g = sns.catplot(x='seasons', y='is_canceled', hue='hotel', data=data,␣
↪kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Season - Canceling Probability')

plt.show(g)
```



[89]:
```
columns_to_dummy.extend(['arrival_date_month'])
```

[90]:
```
#Showcasing final analysis on arrival_date_month
analysis['arrival_date_month']
```

[90]:
```
['We have higher canceling probabilities for summer.',
 'Lowest Cancel Probability is for winter season.']
```

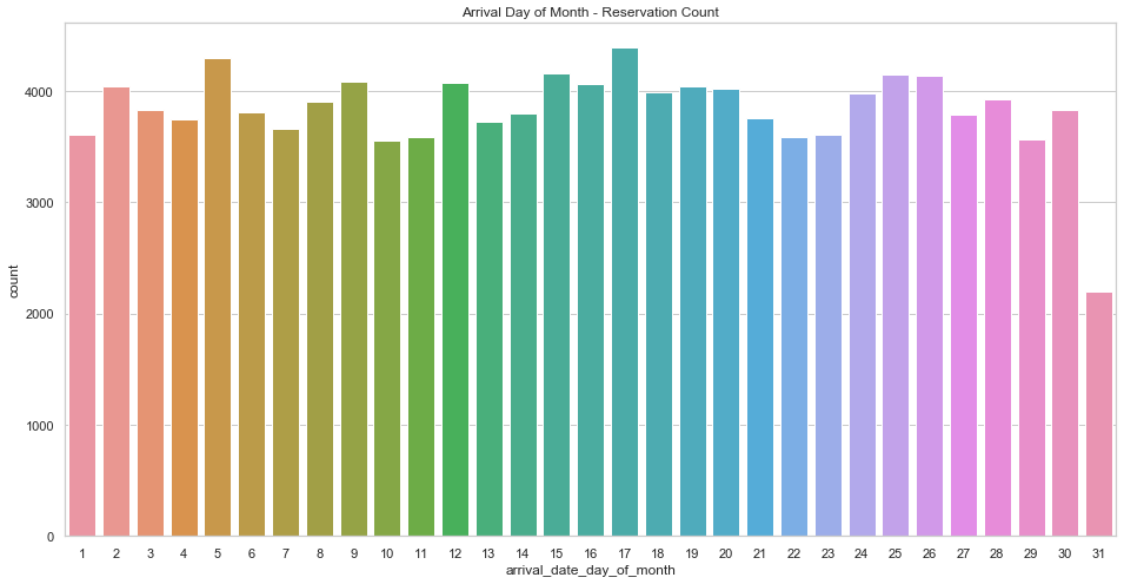
[91]:
```
columns_to_remove.append('seasons')
```

0.0.9 6 - arrival_date_day_of_month: - Section ??

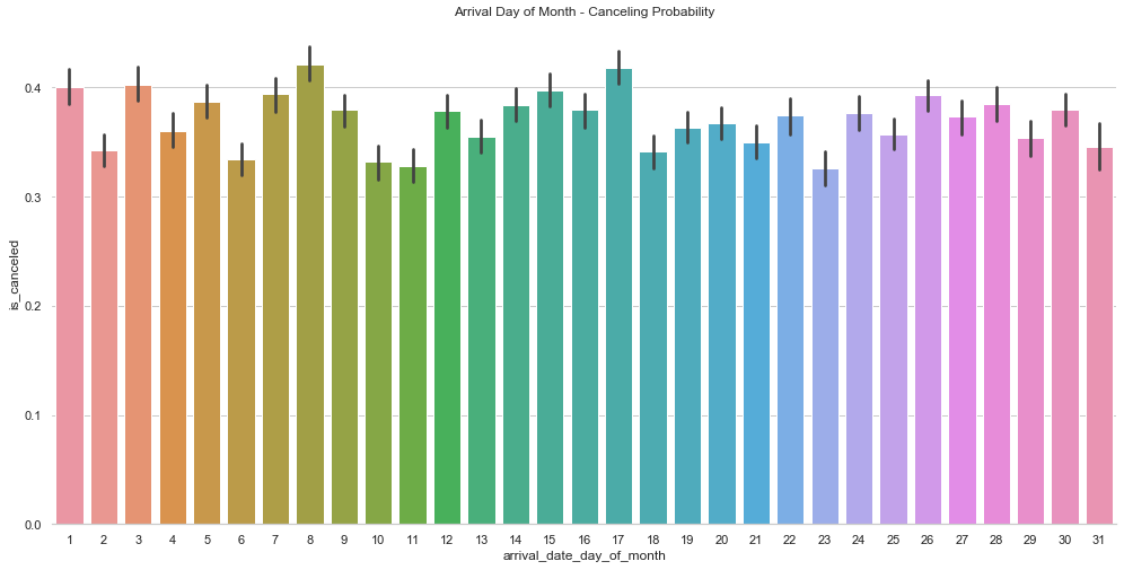
[92]:
```
data['arrival_date_day_of_month'].describe()
```

[92]:
```
count    118898.000000
mean         15.800880
std           8.780324
min           1.000000
25%           8.000000
50%          16.000000
75%          23.000000
max          31.000000
Name: arrival_date_day_of_month, dtype: float64
```

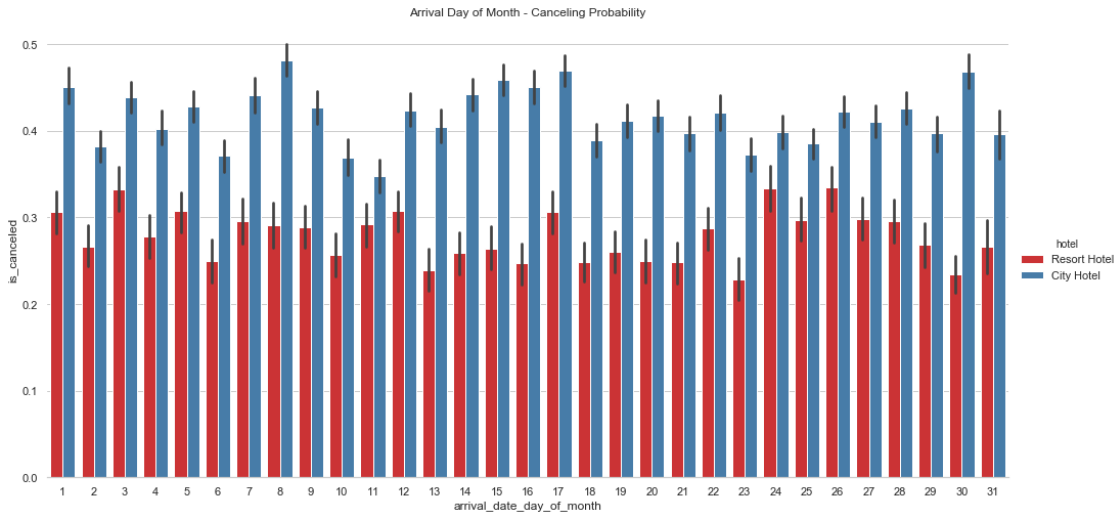
[93]:
```
g = sns.countplot(x='arrival_date_day_of_month', data=data)

g.set(title='Arrival Day of Month - Reservation Count')

plt.show(g)
```



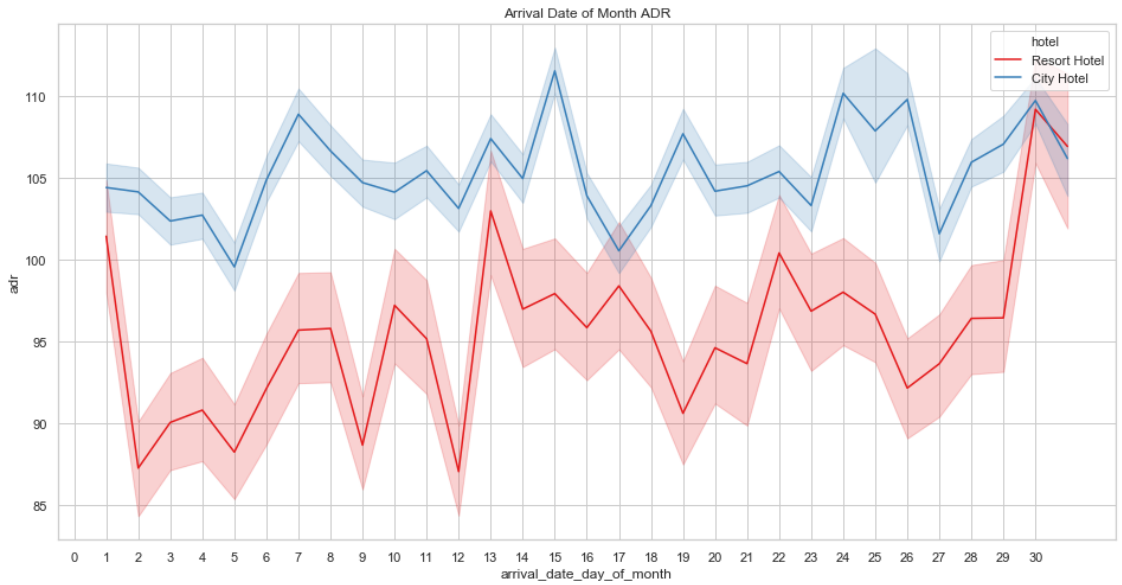
[94]:
```
g = sns.catplot(x='arrival_date_day_of_month', y='is_canceled', data=data,␣
↪kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Day of Month - Canceling Probability')

plt.show(g)
```



[95]:
```
g = sns.catplot(x='arrival_date_day_of_month', y='is_canceled', hue='hotel',␣
↪data=data, kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Day of Month - Canceling Probability')

plt.show(g)
```



[96]:
```
g = sns.lineplot(x='arrival_date_day_of_month', y='adr', data=data, hue='hotel')

plt.xticks(range(0, 31))
plt.title("Arrival Date of Month ADR")
plt.show(g)
```



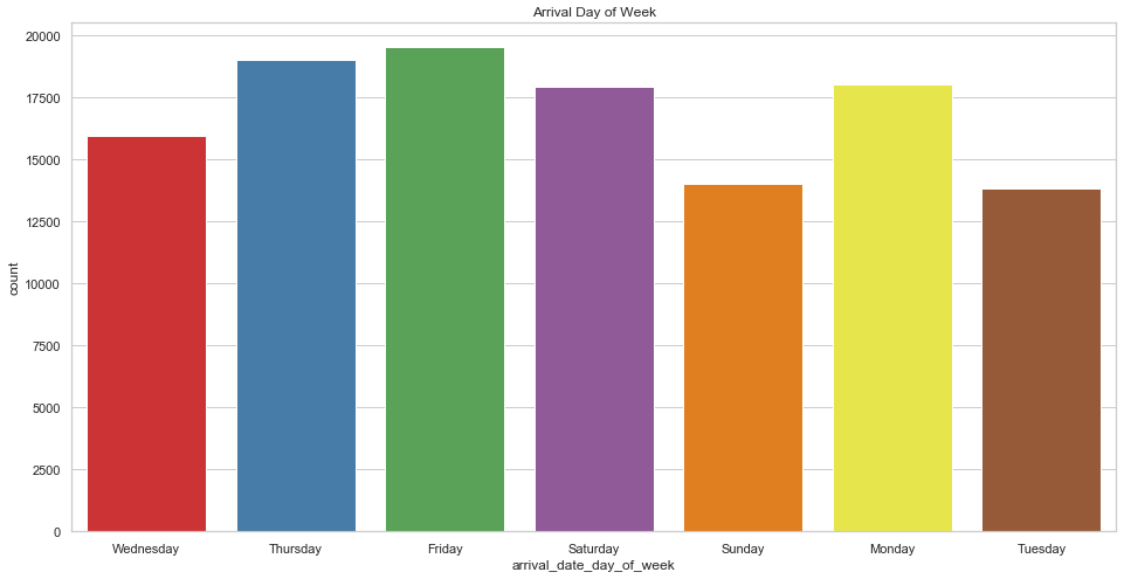
[97]:
```
# we will convert day of month to day of week

def date_to_day_of_week(row):

    import datetime

    months = ['', 'January', 'February', 'March', 'April', 'May', 'June',␣
 ↪'July', 'August', 'September',
              'October', 'November', 'December']

    year = row['arrival_date_year']
    month = months.index(row['arrival_date_month'])
    day = row['arrival_date_day_of_month']

    arrival_date = datetime.date(year, month, day)

    # row['arrival_date_day_of_week'] = arrival_date.strftime("%A")

    return arrival_date.strftime("%A")
```

[98]:
```
data['arrival_date_day_of_week'] = np.nan
```

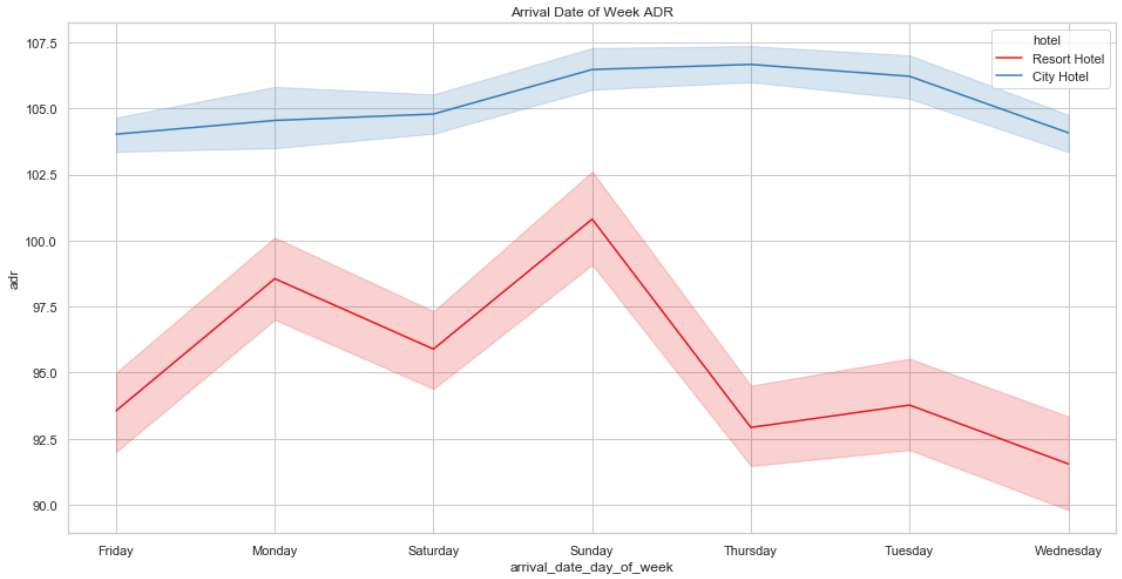
[99]:
```
data['arrival_date_day_of_week'] = data.reset_index().
 ↪apply(date_to_day_of_week, axis=1)
```

[100]:
```
g = sns.countplot(x='arrival_date_day_of_week', data=data)

g.set(title='Arrival Day of Week')
```

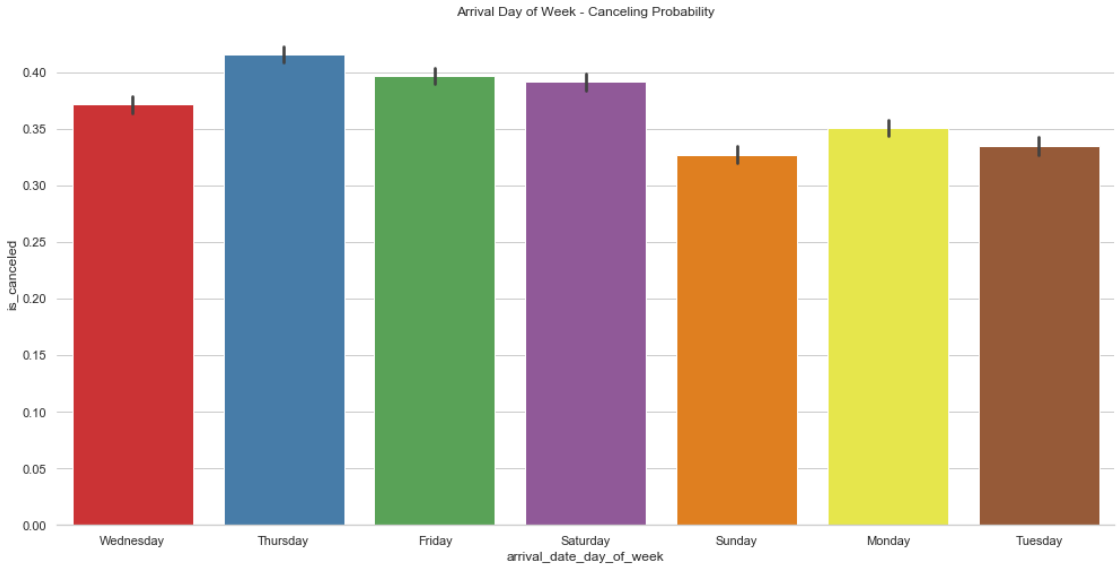
```
plt.show(g)
```



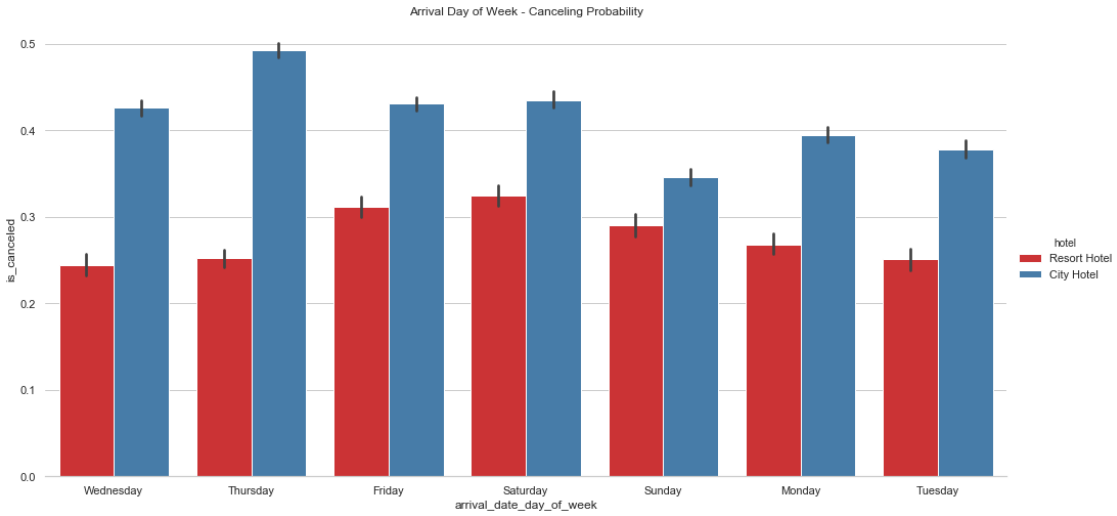
[101]:
```
g = sns.lineplot(x='arrival_date_day_of_week', y='adr', data=data, hue='hotel')

plt.title("Arrival Date of Week ADR")
plt.show(g)
```



[102]:
```
g = sns.catplot(x='arrival_date_day_of_week', y='is_canceled', data=data,␣
↪kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Day of Week - Canceling Probability')

plt.show(g)
```



[103]:
```
g = sns.catplot(x='arrival_date_day_of_week', y='is_canceled', hue='hotel',␣
↪data=data, kind='bar', height=7, aspect=2)

g.despine(left=True)

g.set(title='Arrival Day of Week - Canceling Probability')

plt.show(g)
```



[104]:
```
analysis['arrival_date_day_of_month'].append('Risky days for hotels differ from␣
↪each other.')
```

[105]:
```
# we will drop the column of day_of_month

columns_to_remove.append('arrival_date_day_of_month')
```

[106]:
```
columns_to_dummy.append('arrival_date_day_of_week')
```

[107]:
```
#Showcasing final analysis on 'arrival_date_day_of_month'
analysis['arrival_date_day_of_month']
```

[107]:
```
['Risky days for hotels differ from each other.']
```

0.0.10 7 - stays__in__weekend__nights: - Section ??

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

[108]:
```
data['stays_in_weekend_nights'].isna().sum()
```

[108]:
```
0
```

[109]:
```
data['stays_in_weekend_nights'].describe()
```

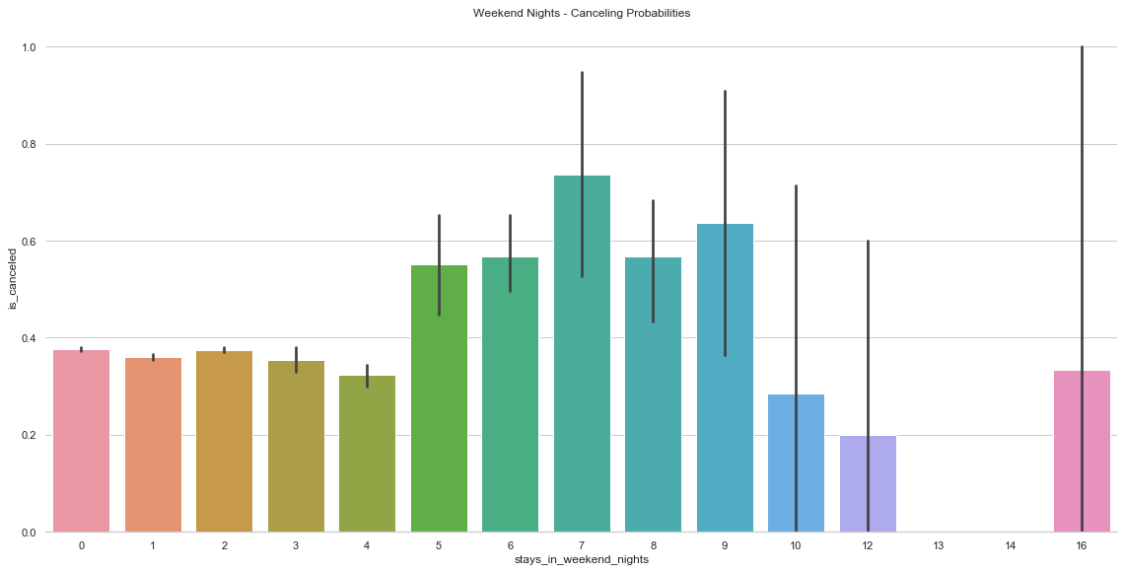
[109]:
```
count    118898.000000
mean          0.928897
std           0.996216
min           0.000000
25%           0.000000
```

```
50%            1.000000
75%            2.000000
max           16.000000
Name: stays_in_weekend_nights, dtype: float64
```

[110]:
```
sorted(data['stays_in_weekend_nights'].unique())
```

[110]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16]
```

[111]:
```
g = sns.catplot(x='stays_in_weekend_nights', y='is_canceled', data=data,␣
↪kind='bar', height=8, aspect=2)

g.despine(left=True)

g.set(title='Weekend Nights - Canceling Probabilities')

plt.show(g)
```



0.0.11 8 - stays__in__week__nights: - Section ??

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

[112]:
```
data['stays_in_week_nights'].isna().sum()
```

[112]:
```
0
```

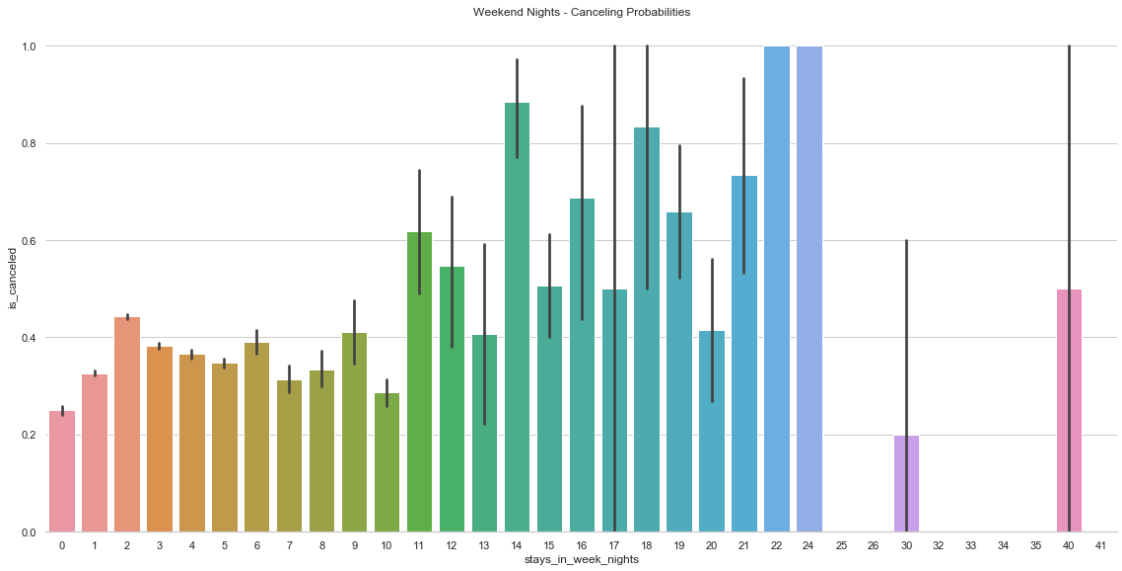
[113]:
```
data['stays_in_week_nights'].describe()
```

[113]:
```
count    118898.000000
mean          2.502145
std           1.900168
min           0.000000
25%           1.000000
50%           2.000000
75%           3.000000
max          41.000000
Name: stays_in_week_nights, dtype: float64
```

[114]:
```
data['stays_in_week_nights'].value_counts(normalize=True)
```

[114]:
```
2     0.282376
1     0.253082
3     0.186740
5     0.092945
4     0.080355
0     0.063861
6     0.012540
10    0.008663
7     0.008638
8     0.005501
9     0.001943
15    0.000715
11    0.000463
19    0.000370
12    0.000353
20    0.000345
14    0.000294
13    0.000227
16    0.000135
21    0.000126
22    0.000059
18    0.000050
25    0.000050
30    0.000042
17    0.000034
24    0.000025
40    0.000017
26    0.000008
32    0.000008
33    0.000008
34    0.000008
35    0.000008
41    0.000008
Name: stays_in_week_nights, dtype: float64
```

[115]:
```
g = sns.catplot(x='stays_in_week_nights', y='is_canceled', data=data,␣
↪kind='bar', height=8, aspect=2)

g.despine(left=True)

g.set(title='Weekend Nights - Canceling Probabilities')

plt.show(g)
```



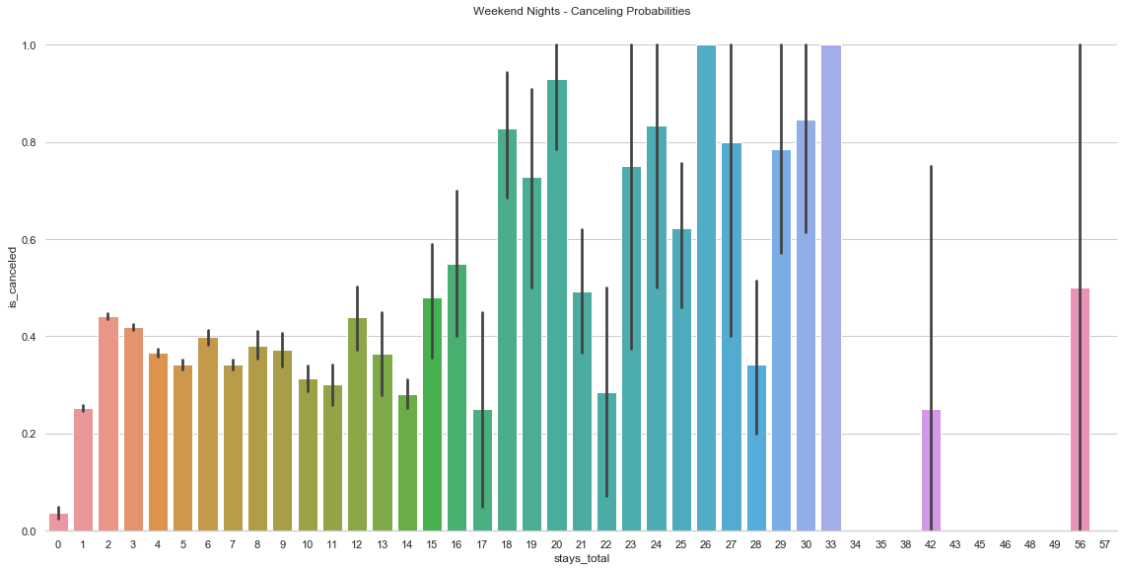
[116]:
```
# we may check the total stay time.

data['stays_total'] = data['stays_in_week_nights'] +␣
↪data['stays_in_weekend_nights']
```

[117]:
```
data['stays_total'].describe()
```

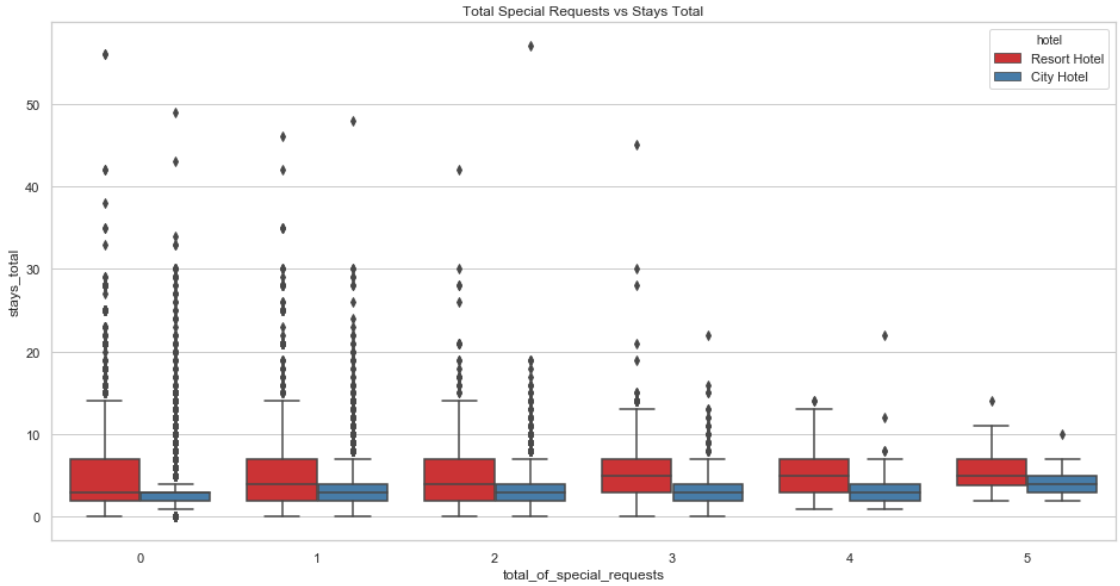
[117]:
```
count    118898.000000
mean          3.431042
std           2.544938
min           0.000000
25%           2.000000
50%           3.000000
75%           4.000000
max          57.000000
Name: stays_total, dtype: float64
```

[118]:
```
g = sns.catplot(x='stays_total', y='is_canceled', data=data, kind='bar',␣
↪height=8, aspect=2)
```

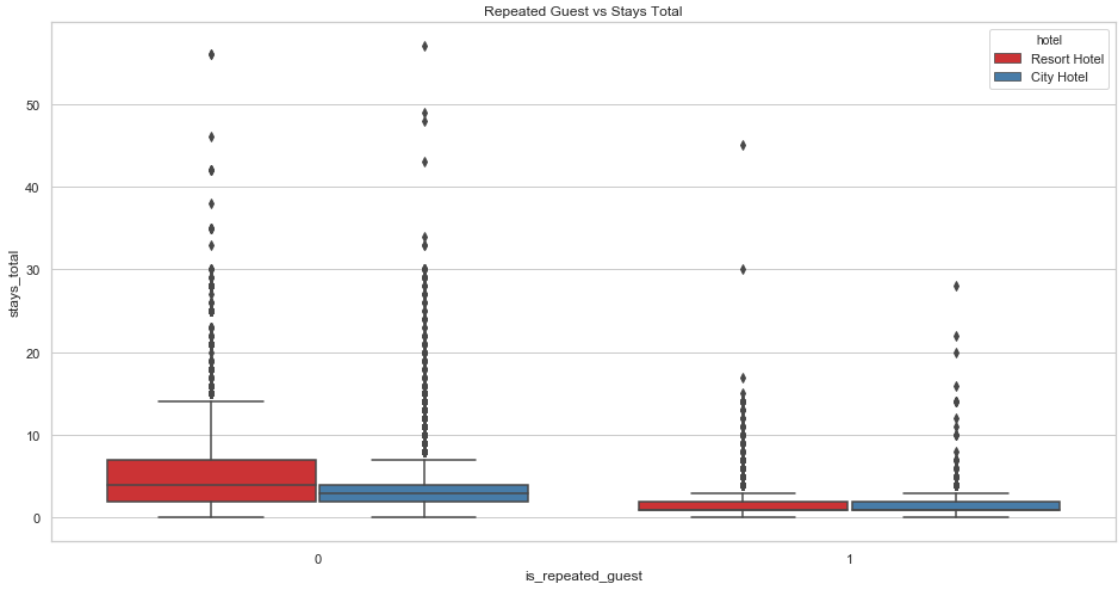
```
g.despine(left=True)

g.set(title='Weekend Nights - Canceling Probabilities')

plt.show(g)
```



[119]:
```
g = sns.boxplot(x='total_of_special_requests', y='stays_total', data=data,␣
↪hue='hotel')

plt.title("Total Special Requests vs Stays Total")
plt.show(g)
```



[120]:
```
g = sns.boxplot(x='is_repeated_guest', y='stays_total', data=data, hue='hotel')

plt.title("Repeated Guest vs Stays Total")
plt.show(g)
```



0.0.12 9 - adults: - Section ??

- Number of adults

[121]:
```
data['adults'].describe()
```

[121]:
```
count    118898.000000
mean          1.858391
std           0.578576
min           0.000000
25%           2.000000
50%           2.000000
75%           2.000000
max          55.000000
Name: adults, dtype: float64
```

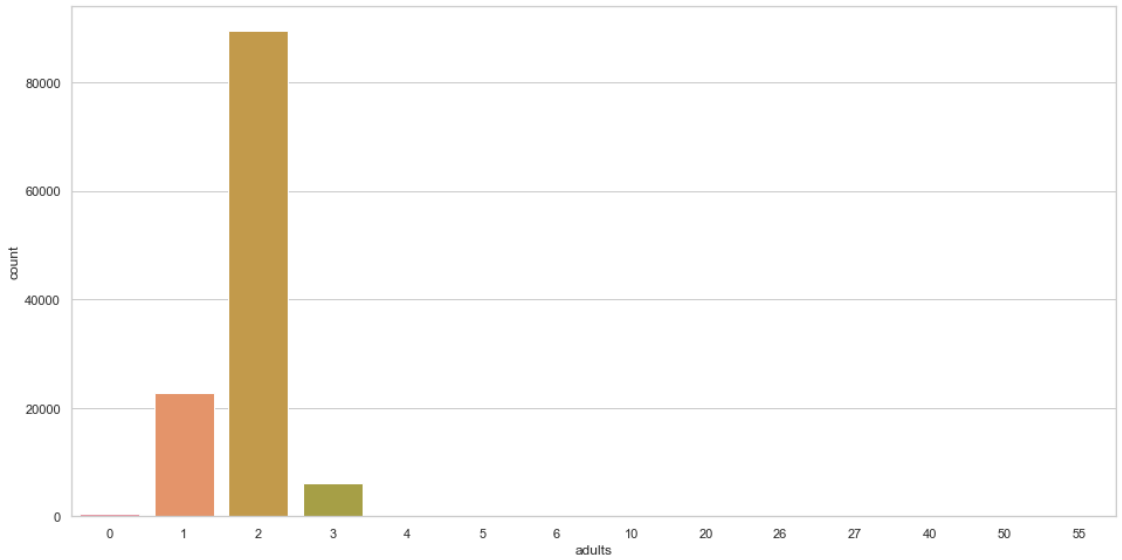
[122]:
```
data['adults'].unique()
```

[122]:
```
array([ 2,  1,  3,  4, 40, 26, 50, 27, 55,  0, 20,  6,  5, 10])
```

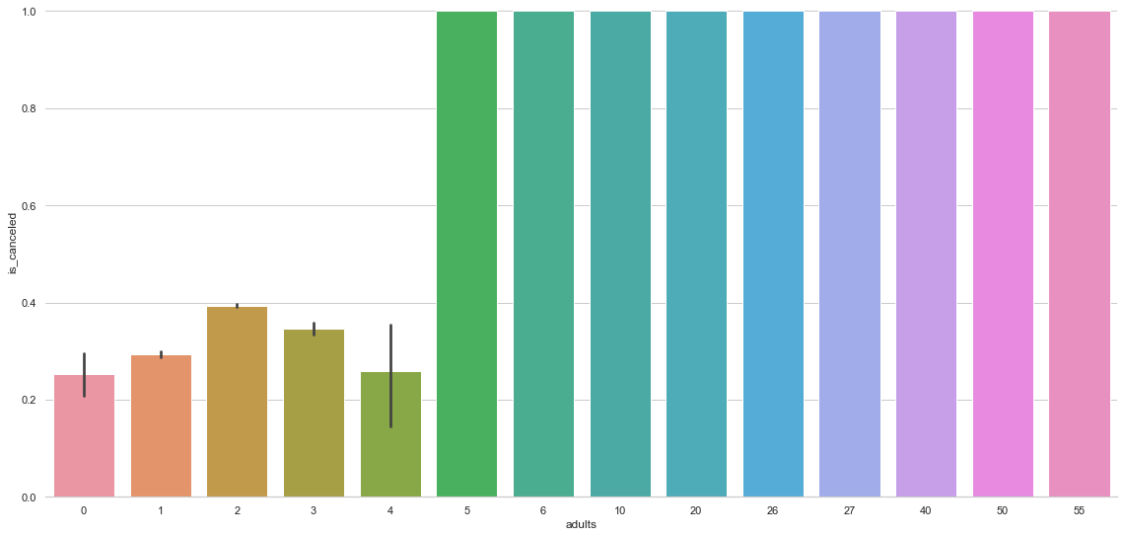
[123]:
```
data['adults'].value_counts()
```

[123]:
```
2     89495
1     22735
3      6197
0       393
4        62
26        5
27        2
20        2
5         2
55        1
50        1
40        1
10        1
6         1
Name: adults, dtype: int64
```

[124]:
```
g = sns.countplot(x='adults', data=data)

plt.show(g)
```



[125]:
```
g = sns.catplot(x='adults', y='is_canceled', data=data, kind='bar', height=8,␣
↪aspect=2)

g.despine(left=True)

plt.show(g)
```

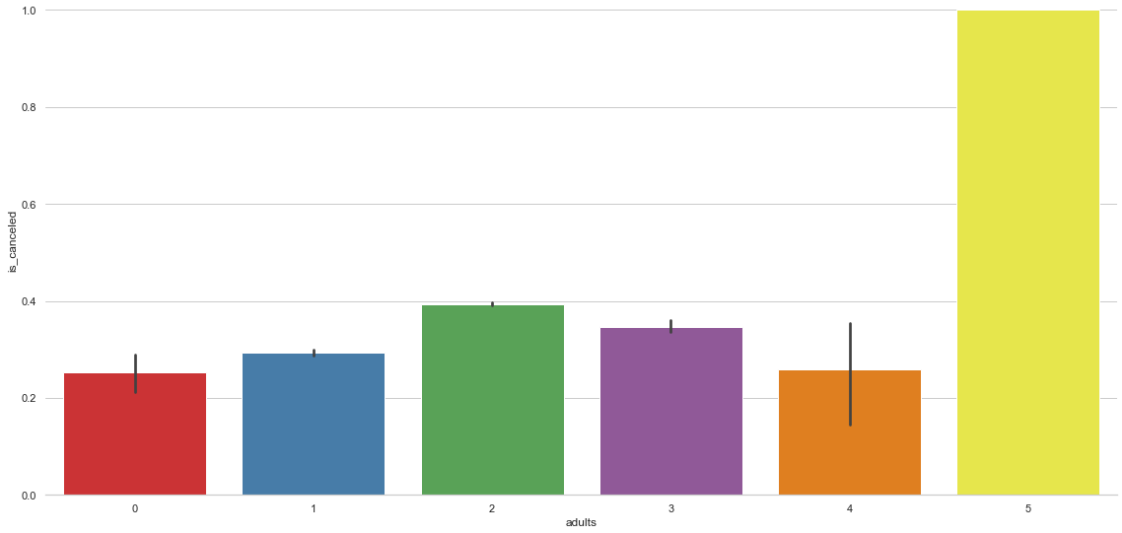


[126]:
```
# we can create a column for all adults count > 4

def adults_large(adults):
```

```
    if adults > 4:
        return 5
    else:
        return adults
```

[127]:
```
data['adults'] = data['adults'].apply(adults_large)
```

[128]:
```
g = sns.catplot(x='adults', y='is_canceled', data=data, kind='bar', height=8,␣
↪aspect=2)

g.despine(left=True)

plt.show(g)
```



0.0.13 10 - children: - Section ??

- Number of children

[129]:
```
data['children'].value_counts()
```

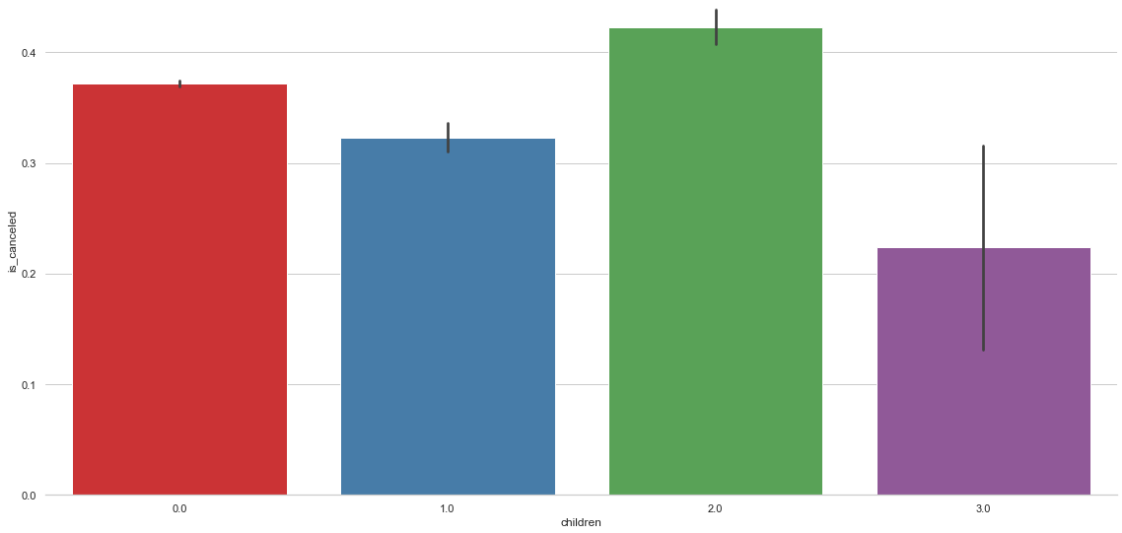
[129]:
```
0.0     110319
1.0       4852
2.0       3650
3.0         76
10.0         1
Name: children, dtype: int64
```

[130]:
```
# 10 children seems to be wrong or outliar value.

data[data['children'] > 9]
```

[130]:
```
            hotel  is_canceled  lead_time  arrival_date_year  \
328  Resort Hotel            1         55               2015

    arrival_date_month  arrival_date_week_number  arrival_date_day_of_month  \
328               July                        29                         12

     stays_in_weekend_nights  stays_in_week_nights  adults  …  total_stays  \
328                        4                   10       2  …           14

     customer_total_payment lead_time_30 lead_time_60 lead_time_120  \
328                 1864.24            1            0             0

    lead_time_360  arrival_date_weekth_in_month  seasons  \
328             0                             1   summer

     arrival_date_day_of_week stays_total
328                    Sunday          14

[1 rows x 40 columns]
```

[131]:
```
data.drop(data[data['children'] > 9].index, axis=0, inplace=True)
```

[132]:
```
g = sns.catplot(x='children', y='is_canceled', data=data, kind='bar', height=8,␣
↪aspect=2)

g.despine(left=True)

plt.show(g)
```



0.0.14 11 - babies: - Section ??

- Number of babies

[133]:
```
data['babies'].describe()
```

[133]:
```
count    118897.000000
mean          0.007948
std           0.097381
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          10.000000
Name: babies, dtype: float64
```

[134]:
```
data['babies'].unique()
```

[134]:
```
array([ 0,  1,  2, 10,  9])
```

[135]:
```
data['babies'].value_counts()
```

[135]:
```
0     117983
1        898
2         14
10         1
9          1
Name: babies, dtype: int64
```

- 9 and 10 babies are seems to be outliars

[136]:
```
data[data['babies'] == 9]
```

[136]:
```
            hotel  is_canceled  lead_time  arrival_date_year  \
78656  City Hotel            0         11               2015

      arrival_date_month  arrival_date_week_number  arrival_date_day_of_month  \
78656            October                        42                         11

       stays_in_weekend_nights  stays_in_week_nights  adults  …  \
78656                        2                     1       1  …

       total_stays  customer_total_payment lead_time_30 lead_time_60  \
78656            3                   285.0            0            0

      lead_time_120 lead_time_360  arrival_date_weekth_in_month  seasons  \
78656             0             0                             2   autumn

       arrival_date_day_of_week stays_total
78656                    Monday           3

[1 rows x 40 columns]
```

[137]:
```
data[data['babies'] == 10]
```

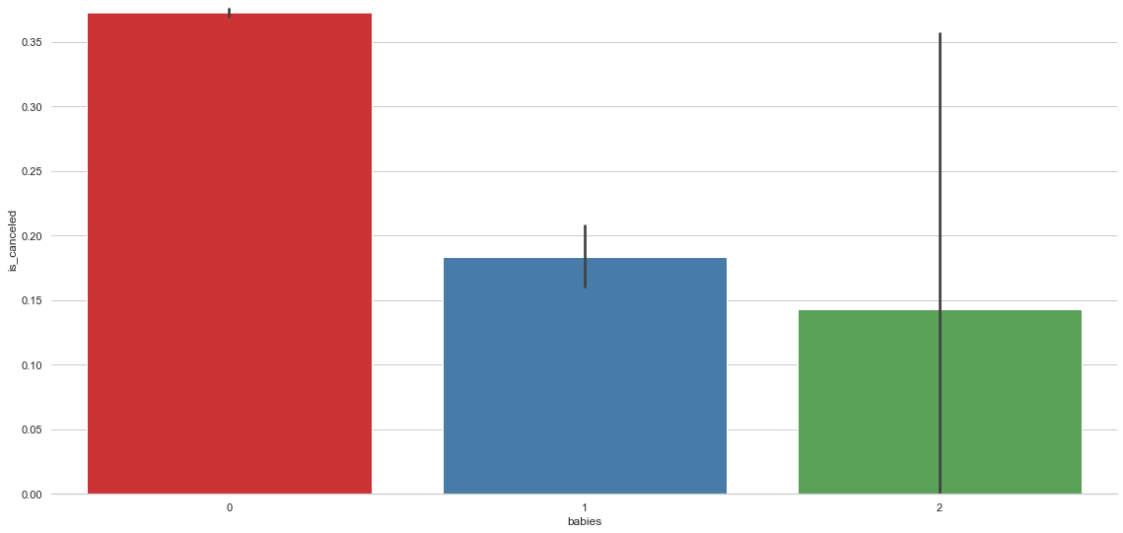
[137]:
```
            hotel  is_canceled  lead_time  arrival_date_year  \
46619  City Hotel            0         37               2016

      arrival_date_month  arrival_date_week_number  arrival_date_day_of_month  \
46619            January                         3                         12

       stays_in_weekend_nights  stays_in_week_nights  adults  …  \
46619                        0                     2       2  …

       total_stays  customer_total_payment lead_time_30 lead_time_60  \
46619            2                   168.9            1            0

      lead_time_120 lead_time_360  arrival_date_weekth_in_month  seasons  \
46619             0             0                             3   winter

       arrival_date_day_of_week stays_total
46619                    Sunday           2

[1 rows x 40 columns]
```

[138]:
```
data.groupby('babies')['is_canceled'].value_counts(normalize=True)
```

[138]:
```
babies  is_canceled
0       0              0.627192
        1              0.372808
1       0              0.816258
        1              0.183742
2       0              0.857143
        1              0.142857
9       0              1.000000
10      0              1.000000
Name: is_canceled, dtype: float64
```

[139]:
```
# we will remove the outliar baby counts

data.drop(data[data['babies'] > 8].index, axis=0, inplace=True)
```

[140]:
```
g = sns.catplot(x='babies', y='is_canceled', data=data, kind='bar', height=8,␣
↪aspect=2)

g.despine(left=True)

plt.show(g)
```



[141]:
```
data.drop(data[data['babies'] > 8].index, axis=0, inplace=True)
```

[142]:
```
data['babies'].value_counts() # we don't have outliars any more.
```

[142]:
```
0    117983
1       898
```

```
2        14
Name: babies, dtype: int64
```

0.0.15 12 - meal: - Section ??

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

BO, BL and ML

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)

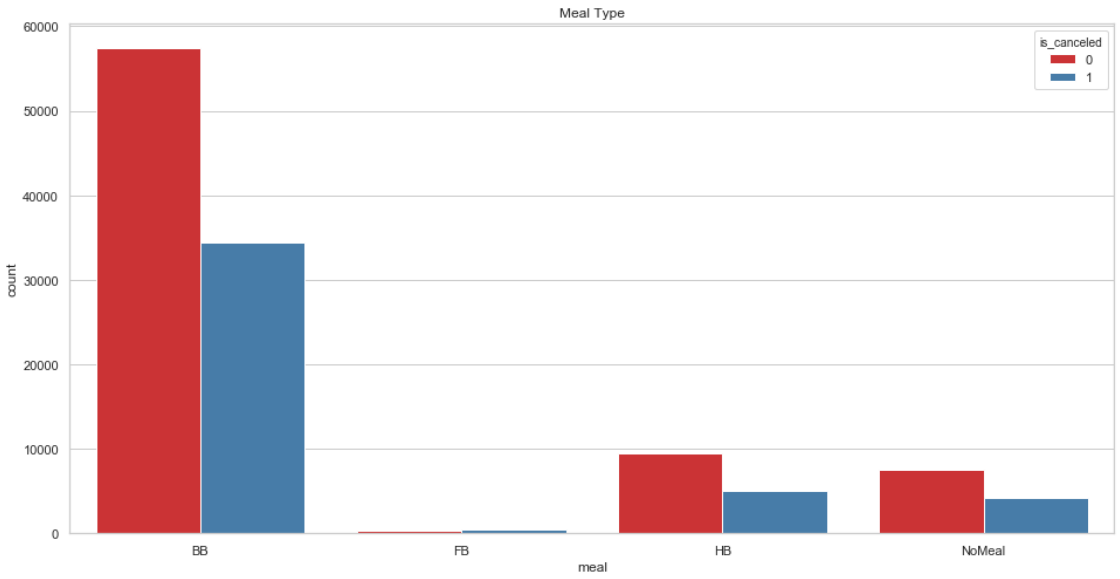
[143]:
```
data['meal'].unique()
```

[143]:
```
array(['BB', 'FB', 'HB', 'SC', 'Undefined'], dtype=object)
```

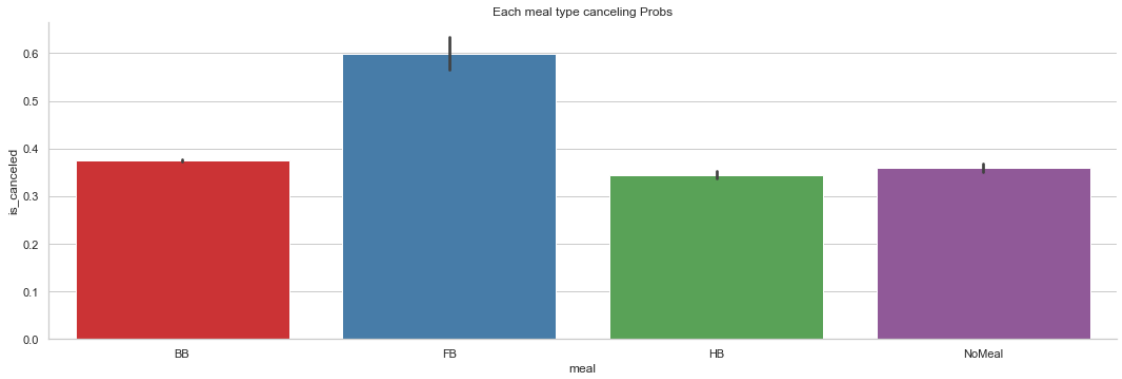
[144]:
```
# 'SC' and 'undefined' means same thing no meal
```

[145]:
```
data['meal'].replace(['SC', 'Undefined'], 'NoMeal', inplace=True)
```

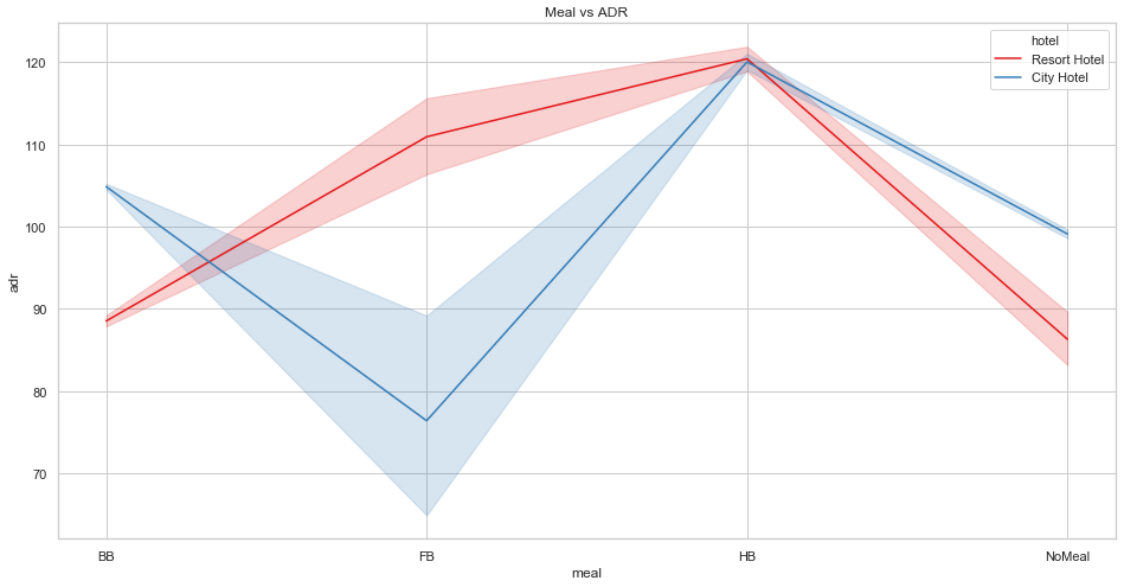
[146]:
```
g = sns.countplot(x='meal', hue='is_canceled', data=data)

g.set(title='Meal Type')

plt.show(g)
```



[147]:
```
g = sns.catplot(x='meal', y='is_canceled', data=data, kind='bar', aspect=3)

g.set(title='Each meal type canceling Probs')

plt.show(g)
```



[148]:
```
g = sns.lineplot(x='meal', y='adr', data=data, hue='hotel')

plt.title("Meal vs ADR")
plt.show(g)
```



- Full meal is the cheapest for city hotel and nomeal is the cheapest for the resort hotel.

[149]:
```
columns_to_dummy.append('meal')
```

0.0.16 13 - Country: - Section ??

[150]:
```
countries_bookings = data.groupby(['country']).size().reset_index(name =␣
↪'Total')
canceled_countries = data[data['is_canceled'] == 1].groupby(['country']).size().
↪reset_index(name = 'Canceled')
not_canceled_countries = data[data['is_canceled'] == 0].groupby(['country']).
↪size().reset_index(name = 'Not_Canceled')
```

[151]:
```
import pycountry

def country_code_to_name(country_code):

    if len(country_code) == 2:
        country = pycountry.countries.get(alpha_2=country_code)
    else:
        country = pycountry.countries.get(alpha_3=country_code)

    if not country:
        return 'Not Found'
    else:
        return country.name
```

[152]:
```
countries_bookings['country_name'] = countries_bookings['country'].
↪apply(country_code_to_name)
not_canceled_countries['country_name'] = not_canceled_countries['country'].
↪apply(country_code_to_name)
canceled_countries['country_name'] = canceled_countries['country'].
↪apply(country_code_to_name)
```

[153]:
```
import plotly.express as px

px.choropleth(countries_bookings,
                    locations = "country",
                    color= "Total",
                    hover_name= "country_name",
                    color_continuous_scale=px.colors.sequential.Oranges,
                    title="Booking Counts by Countries")
```

[154]:
```
px.choropleth(not_canceled_countries,
                    locations = "country",
                    color= "Not_Canceled",
                    hover_name= "country_name",
                    color_continuous_scale=px.colors.sequential.Oranges,
                    title="Not_Canceled Booking Counts by Countries")
```

[155]:
```
px.choropleth(canceled_countries,
                    locations = "country",
                    color= "Canceled",
                    hover_name= "country_name",
                    color_continuous_scale=px.colors.sequential.Oranges,
                    title="Canceled Booking Counts by Countries")
```

[156]:
```
columns_to_dummy.append('country')
```

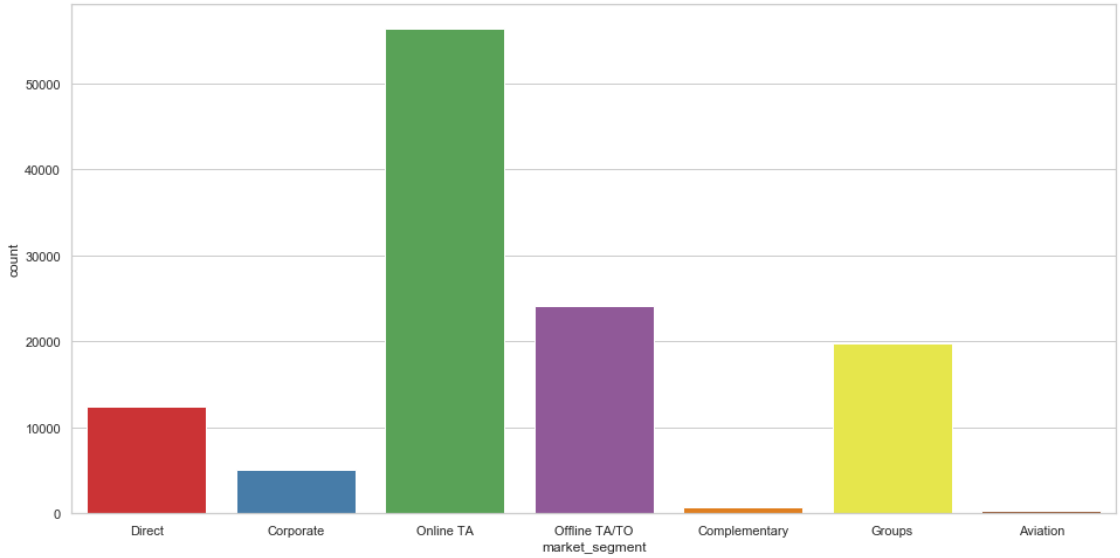
0.0.17 14 - market_segment: - Section ??

- Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"

[157]:
```
data['market_segment'].unique()
```

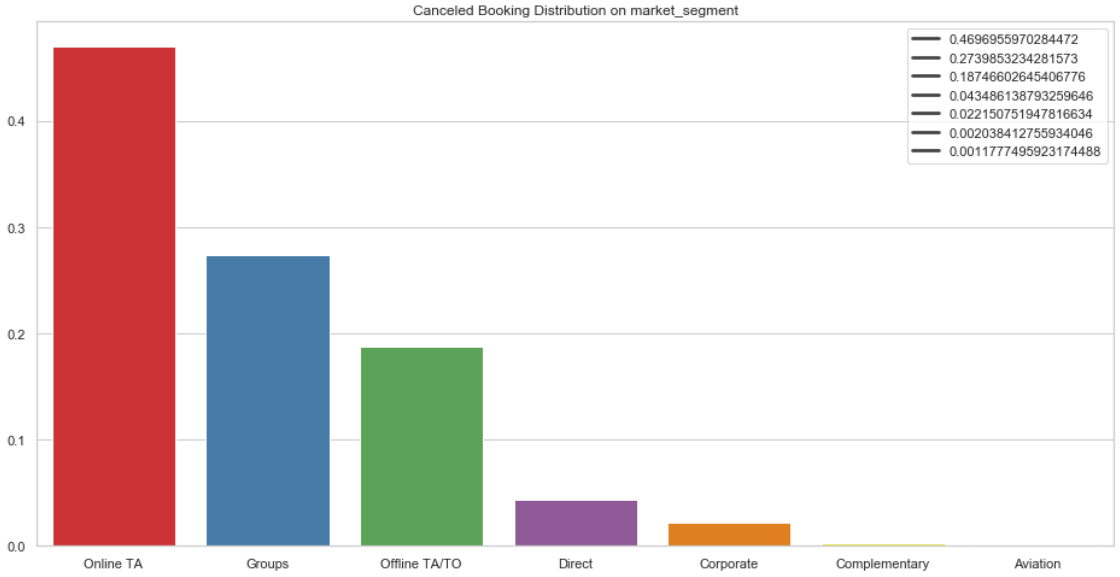
[157]:
```
array(['Direct', 'Corporate', 'Online TA', 'Offline TA/TO',
       'Complementary', 'Groups', 'Aviation'], dtype=object)
```

[158]:
```
g = sns.countplot(x='market_segment', data=data)

plt.show(g)
```



[159]:
```
g = sns.catplot(x='market_segment', y='is_canceled', data=data, kind='bar',␣
↪aspect=3)

plt.show(g)
```

[160]:
```
plot_canceling_prob('market_segment', data)
```



[161]:
```
columns_to_dummy.append('market_segment')
```

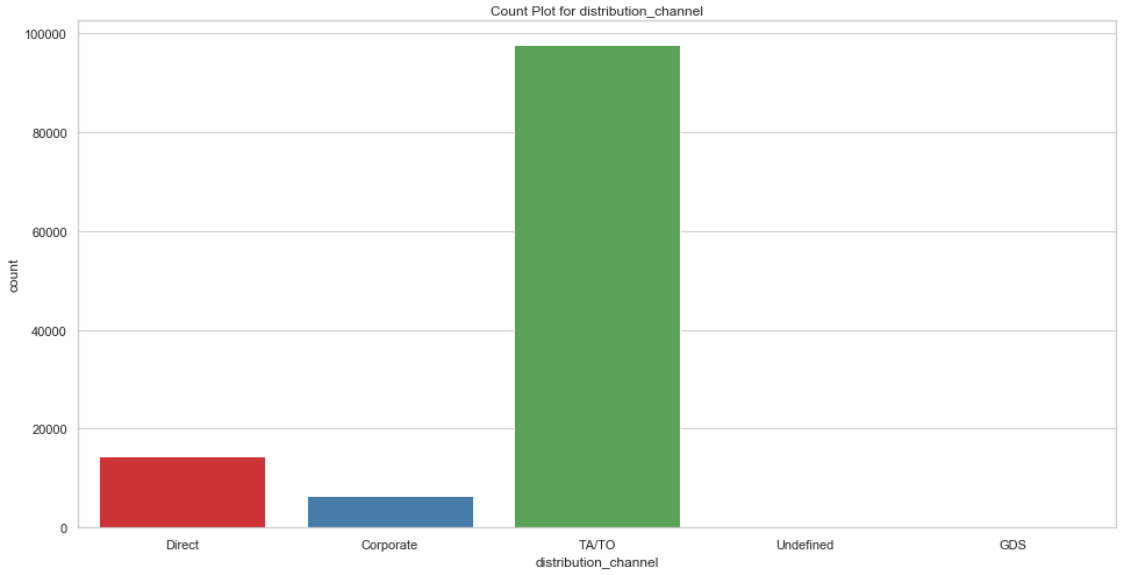
0.0.18 15 - distribution_channel: - Section ??

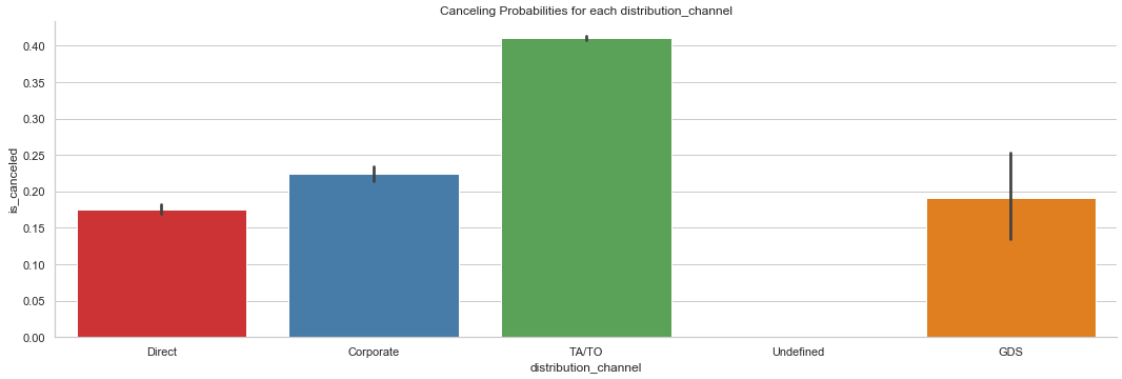
- Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators"

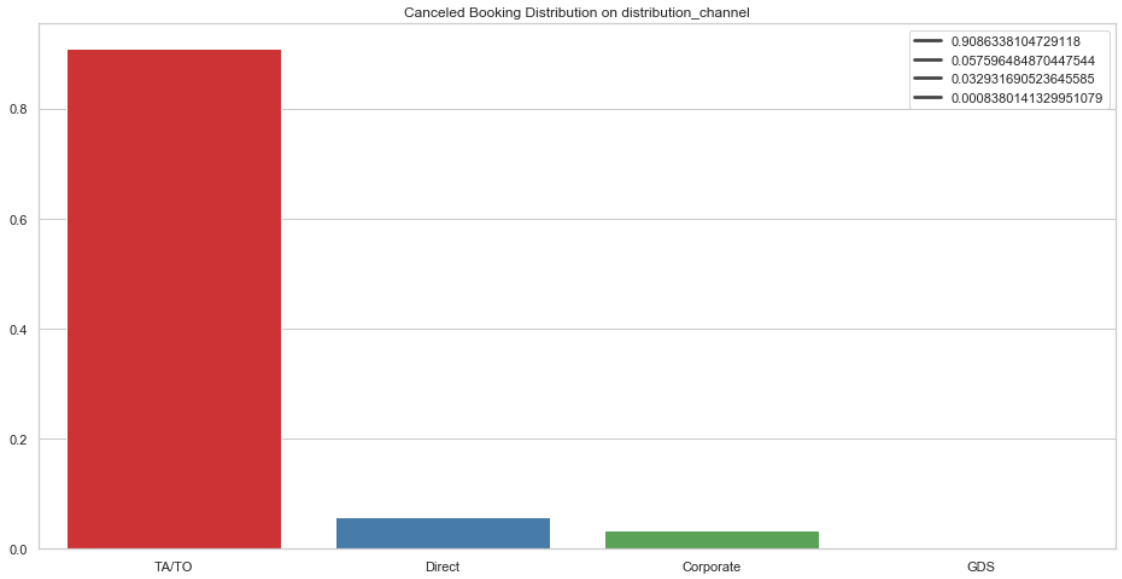
[162]:
```
data['distribution_channel'].unique()
```

[162]:
```
array(['Direct', 'Corporate', 'TA/TO', 'Undefined', 'GDS'], dtype=object)
```

[163]:
```
count_cat_prob_plot('distribution_channel', data)
```


Count Plot for distribution_channel
100000
80000
60000
40000
20000
0
count
Direct
Corporate
TA/TO
Undefined
GDS
distribution_channel


Canceling Probabilities for each distribution_channel
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
is_canceled
Direct
Corporate
TA/TO
Undefined
GDS
distribution_channel



[164]:
```
columns_to_dummy.append('distribution_channel')
```

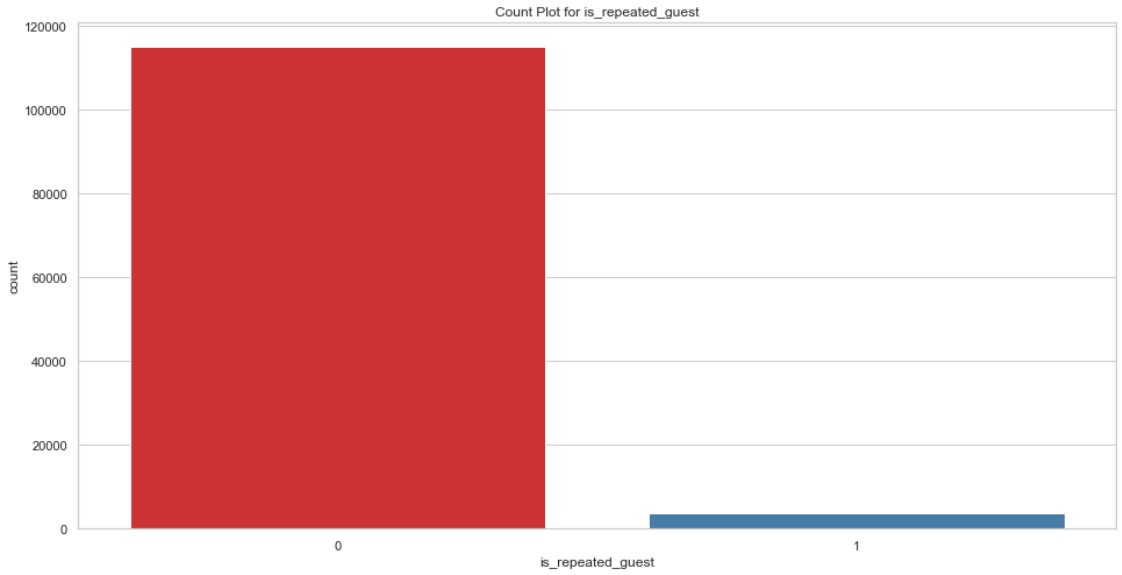
0.0.19 16 - is_repeated_guest: - Section ??

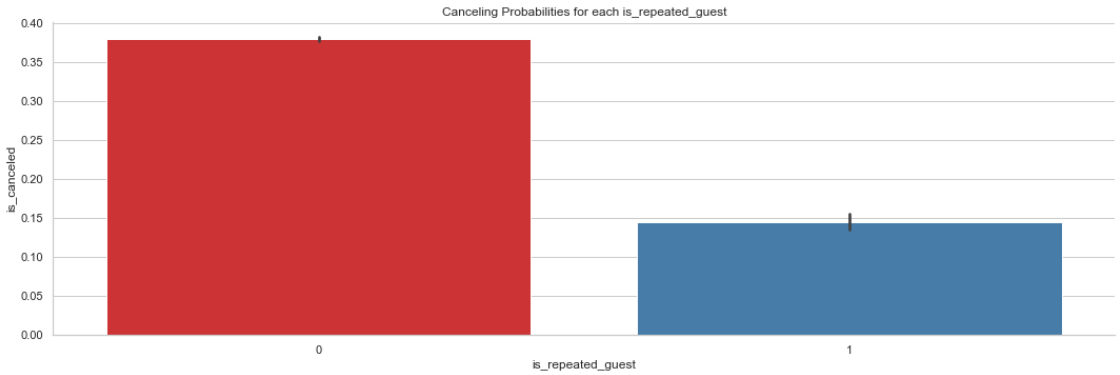
- Value indicating if the booking name was from a repeated guest (1) or not (0)

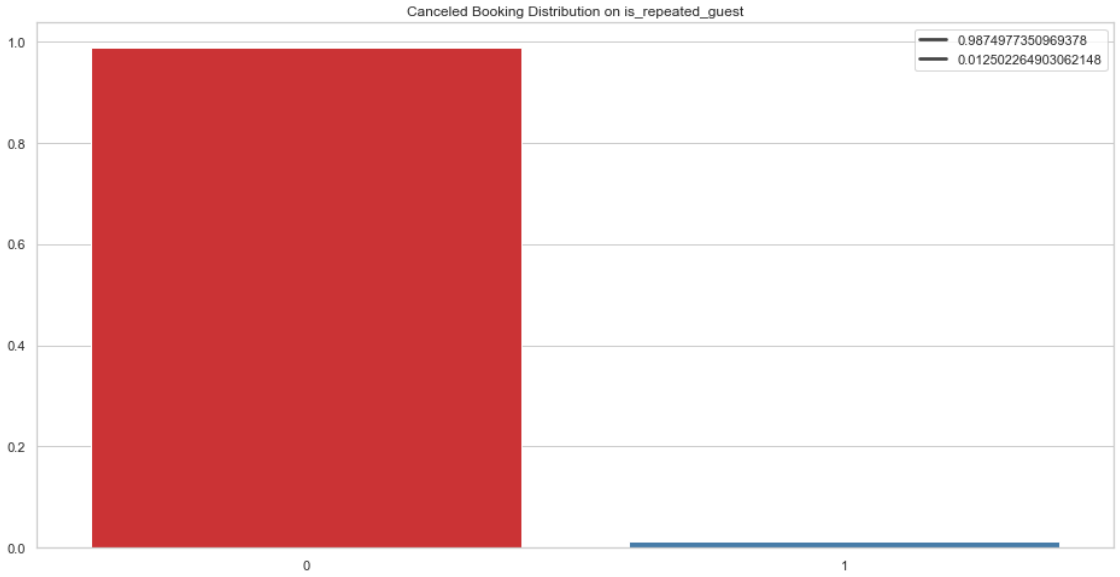
[165]:
```
data['is_repeated_guest'].unique()
```

[165]:
```
array([0, 1])
```

[166]:
```
count_cat_prob_plot('is_repeated_guest', data)
```

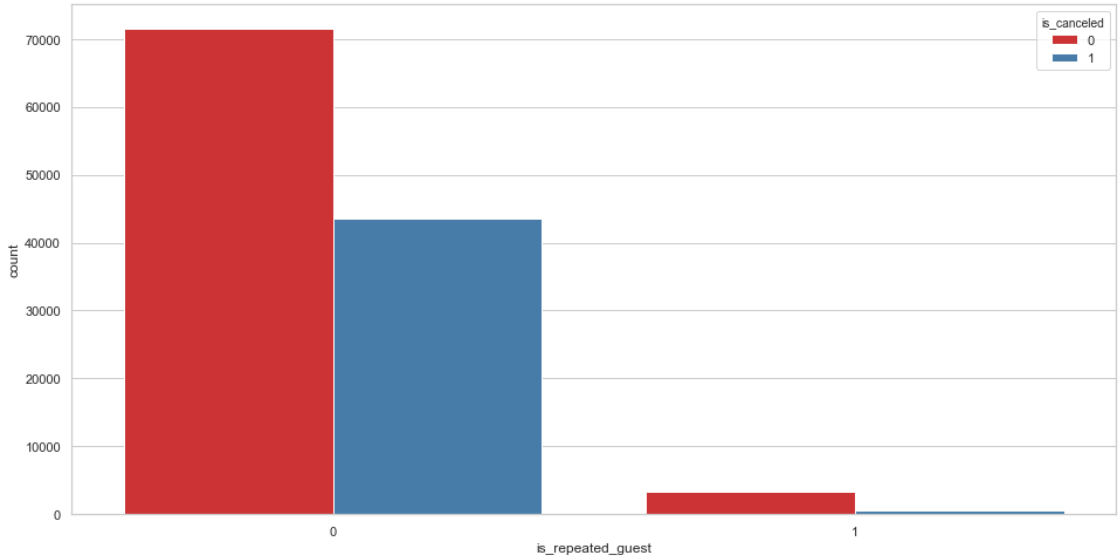






- Most of booking are from new customers.
- Repeated bookings have less canceling probability than new comers.

[167]:
```
g = sns.countplot(x='is_repeated_guest', data=data, hue='is_canceled')
```



[168]:
```
analysis['is_repeated_guest'].extend(['Most of booking are from new customers.
↪', 'Repeated bookings have less canceling probability than new comers.'])
```

[169]:
```
#Showcasing final analysis on is_repeated_guest
analysis['is_repeated_guest']
```

[169]:
```
['Most of booking are from new customers.',
 'Repeated bookings have less canceling probability than new comers.']
```

0.0.20 17 - previous_bookings_not_canceled and previous_cancellations: - Section ??

- Number of previous bookings not cancelled by the customer prior to the current booking

[170]:
```
data['previous_bookings_not_canceled'].unique()
```

[170]:
```
array([ 0,  1,  2,  3,  5,  4,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 21,
       24, 25, 27, 28, 30, 16, 17, 18, 19, 20, 22, 23, 26, 29, 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])
```

[171]:
```
data['previous_bookings_not_canceled'].describe()
```

[171]:
```
count    118895.000000
mean          0.131637
std           1.484691
min           0.000000
25%           0.000000
```

```
50%           0.000000
75%           0.000000
max          72.000000
Name: previous_bookings_not_canceled, dtype: float64
```

[172]:
```
def cancel_ratio(row):

    if not row['previous_bookings_not_canceled'] +␣
↪row['previous_cancellations'] == 0:
        return row['previous_cancellations'] /␣
↪(row['previous_bookings_not_canceled'] + row['previous_cancellations'])
    else:
        return 0
```

[173]:
```
data['customer_cancel_ratio'] = data.apply(cancel_ratio, axis=1)
```

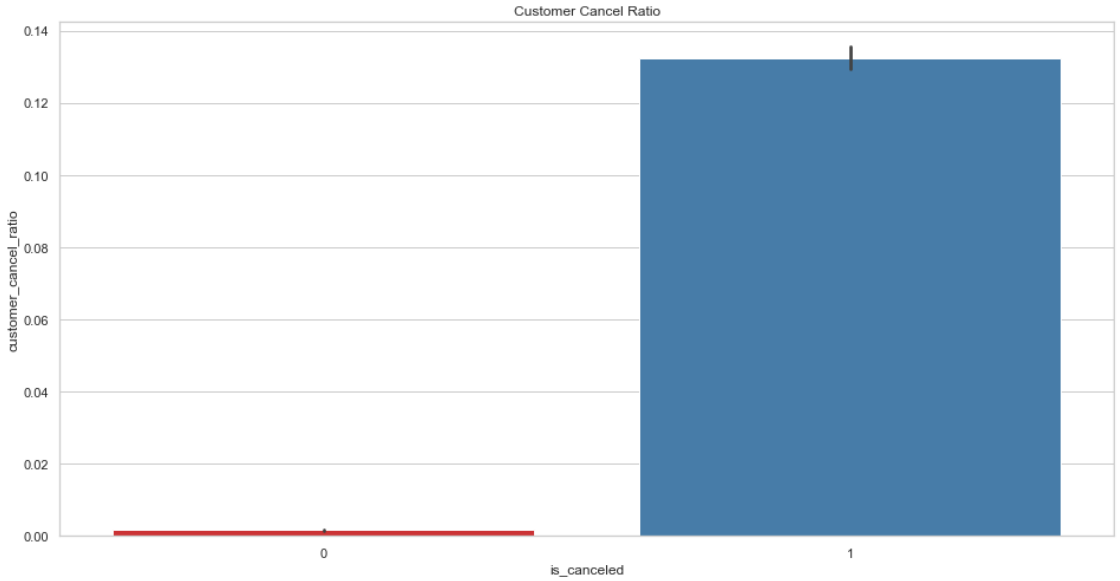
[174]:
```
data['customer_cancel_ratio'].describe()
```

[174]:
```
count    118895.000000
mean          0.050178
std           0.216564
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           1.000000
Name: customer_cancel_ratio, dtype: float64
```

[175]:
```
data.groupby('is_canceled')['customer_cancel_ratio'].mean()
```

[175]:
```
is_canceled
0    0.001649
1    0.132330
Name: customer_cancel_ratio, dtype: float64
```

[176]:
```
g = sns.barplot(x='is_canceled', y='customer_cancel_ratio', data=data)


plt.title("Customer Cancel Ratio")
plt.show(g)
```



[177]:
```
columns_to_remove.extend(['previous_bookings_not_canceled',␣
↪'previous_cancellations'])
```

0.0.21 18 - previous_cancellations: Section ??

- Number of previous bookings that were cancelled by the customer prior to the current booking

[178]:
```
data['previous_cancellations'].unique()
```

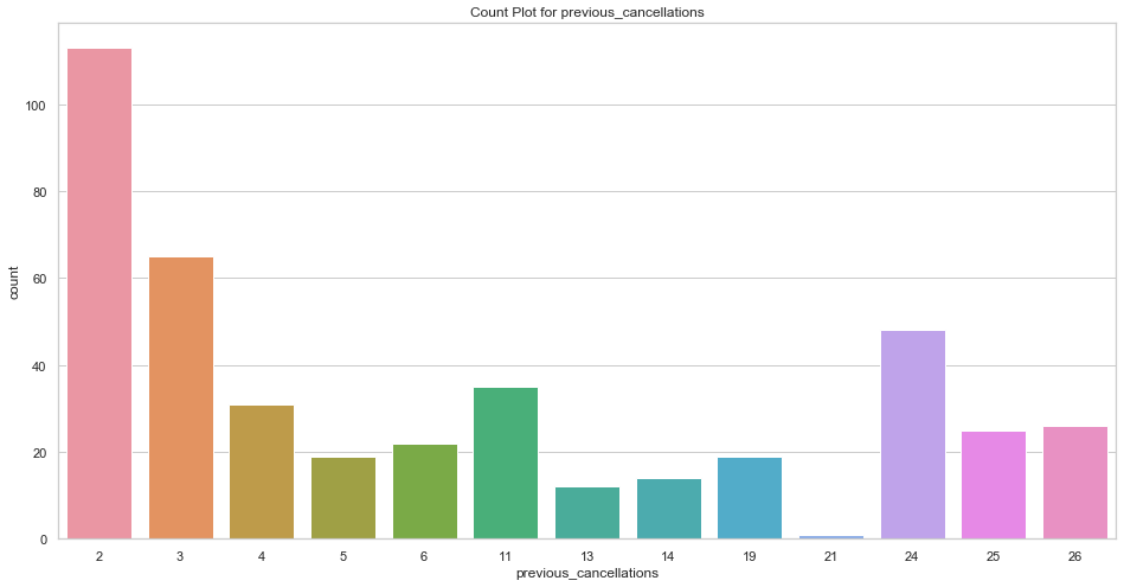
[178]:
```
array([ 0,  1,  2,  3, 26, 25, 14,  4, 24, 19,  5, 21,  6, 13, 11])
```

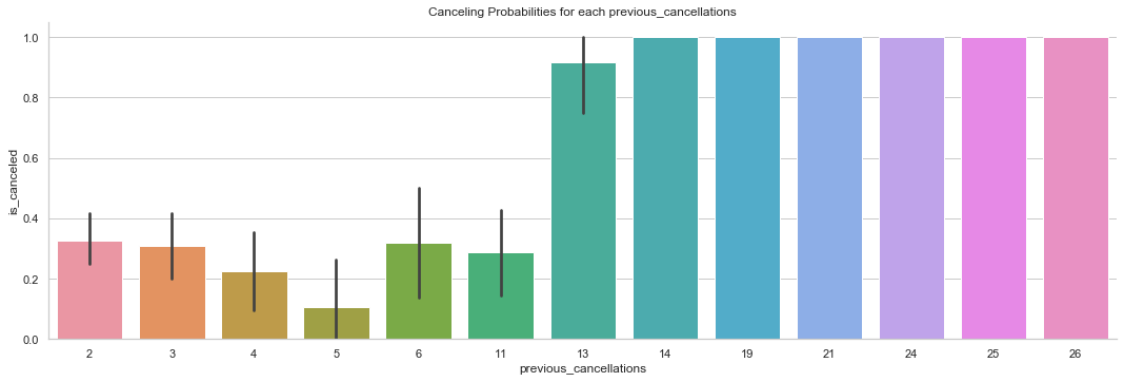
[179]:
```
data['previous_cancellations'].value_counts()
```

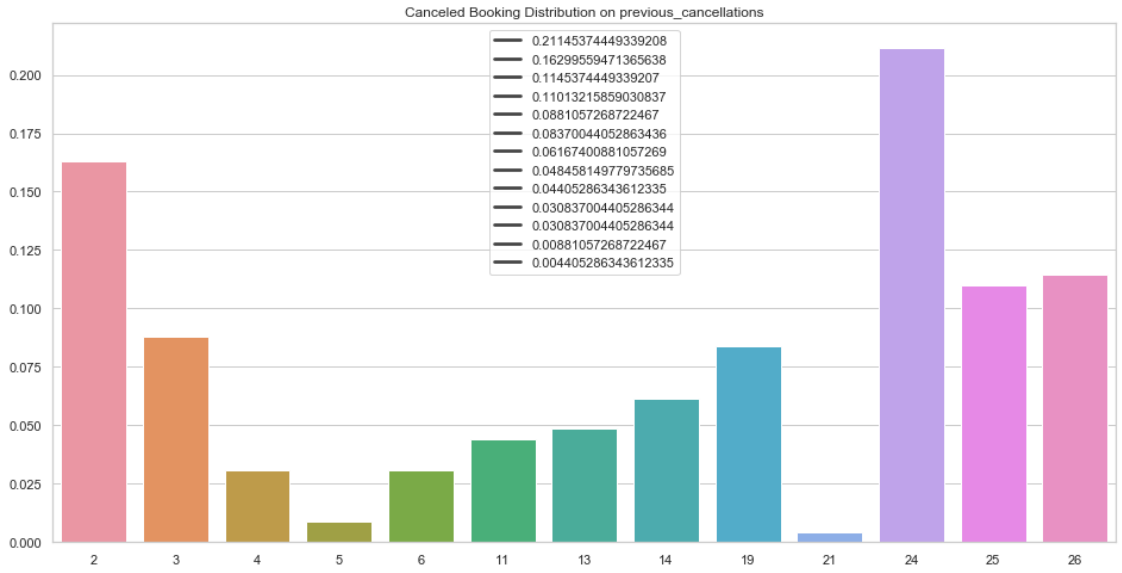
[179]:
```
0     112448
1       6017
2        113
3         65
24        48
11        35
4         31
26        26
25        25
6         22
19        19
5         19
14        14
13        12
21         1
```

```
Name: previous_cancellations, dtype: int64
```

[180]:
```
count_cat_prob_plot('previous_cancellations',␣
↪data[data['previous_cancellations'] > 1])
```







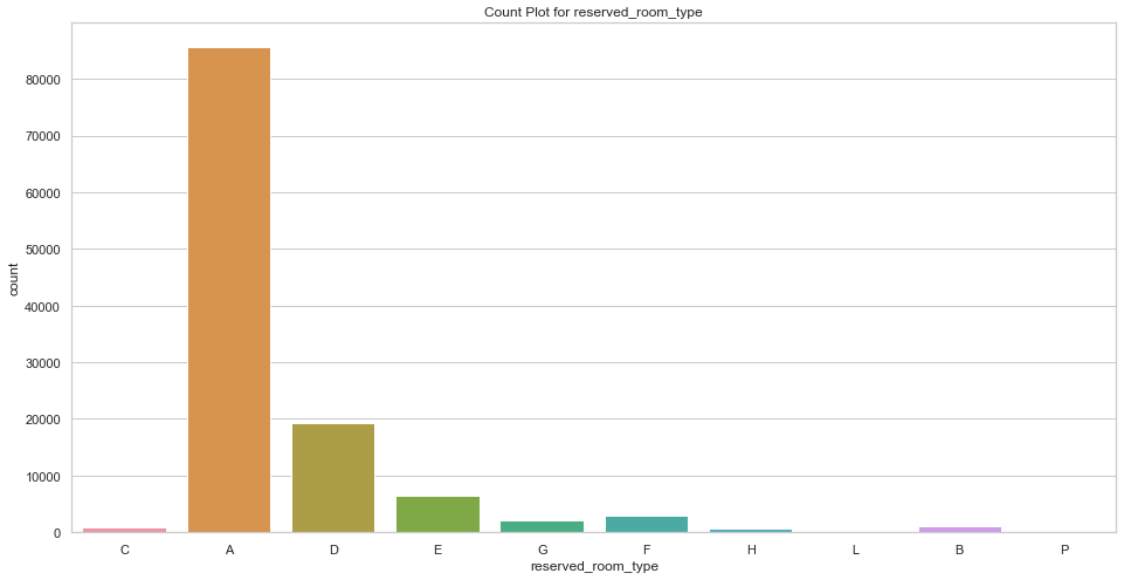
0.0.22 19 - reserved_room_type: Section ??

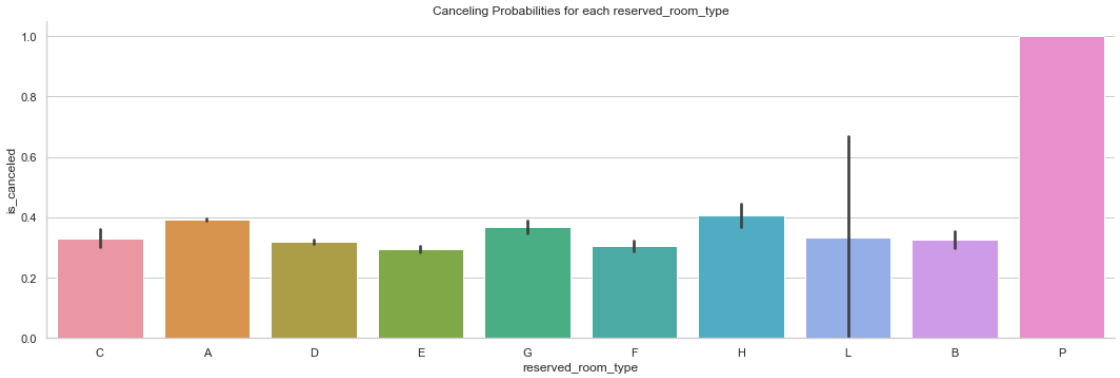
- Code of room type reserved. Code is presented instead of designation for anonymity reasons

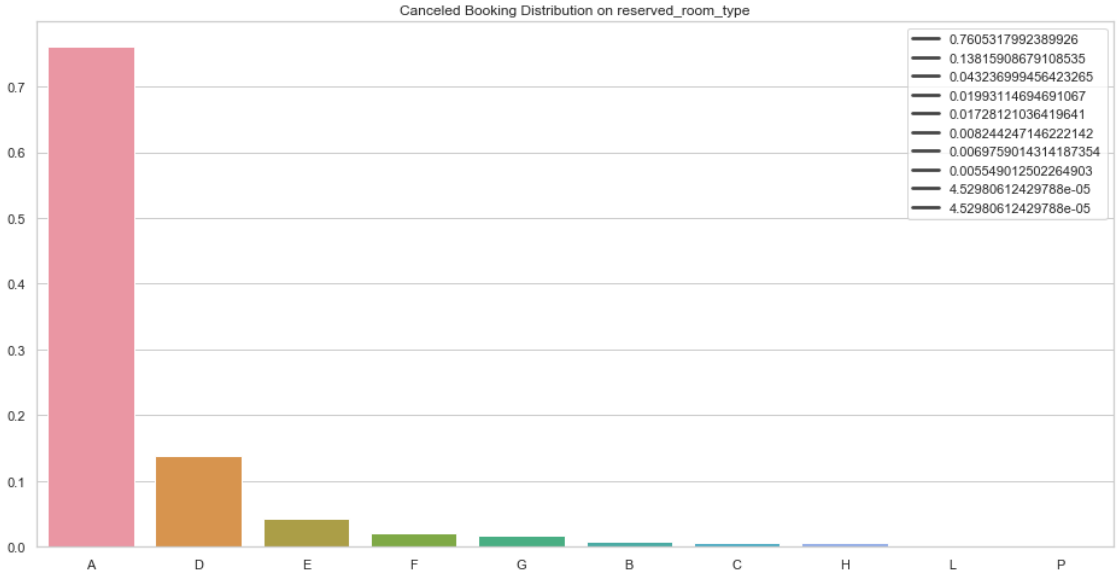
[181]:
```
data['reserved_room_type'].unique()
```

[181]:
```
array(['C', 'A', 'D', 'E', 'G', 'F', 'H', 'L', 'B', 'P'], dtype=object)
```

[182]:
```
count_cat_prob_plot('reserved_room_type', data)
```



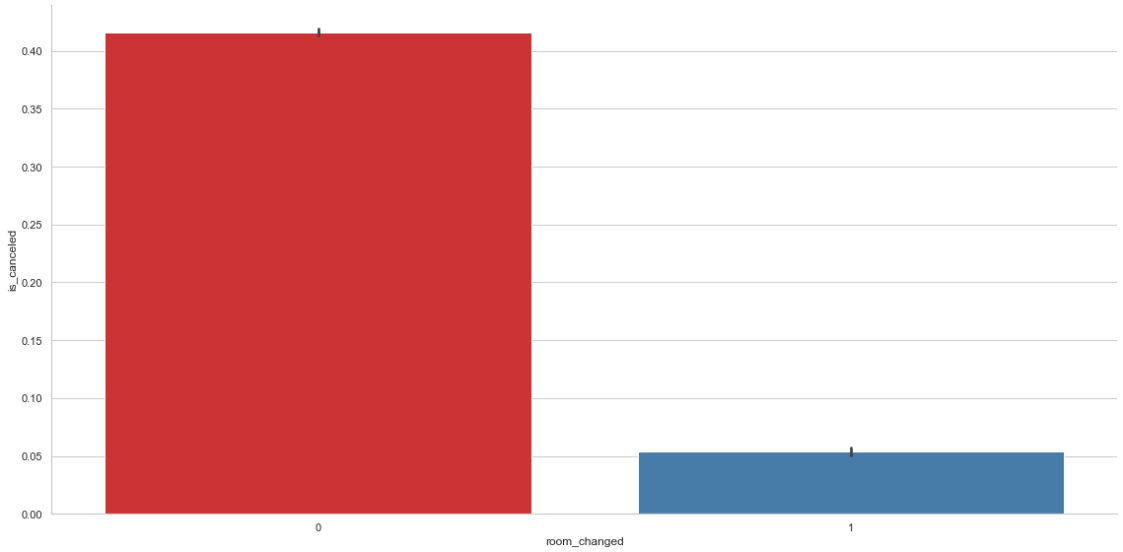




[183]:
```
def is_room_changed(row):

    if row['assigned_room_type'] == row['reserved_room_type']:
        return 0
    else:
        return 1
```

[184]:
```
data['room_changed'] = data.apply(is_room_changed, axis=1)
```

[185]:
```
g = sns.catplot(x='room_changed', y='is_canceled', data=data, kind='bar',␣
↪aspect=2, height=8)
```



[186]:
```
columns_to_dummy.append('reserved_room_type')
```

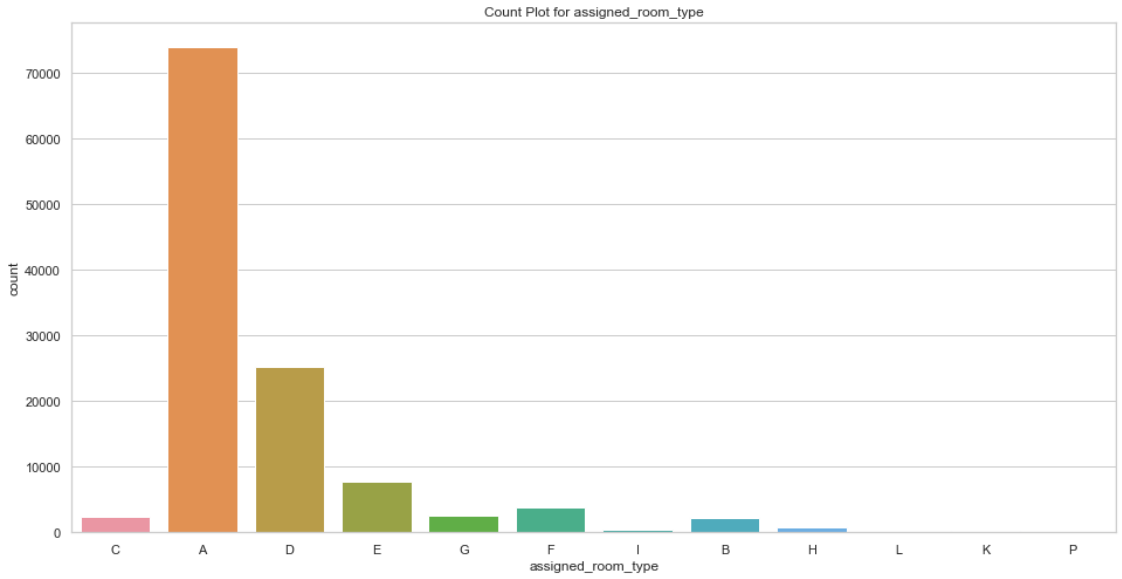
0.0.23 20 - assigned_room_type: Section ??

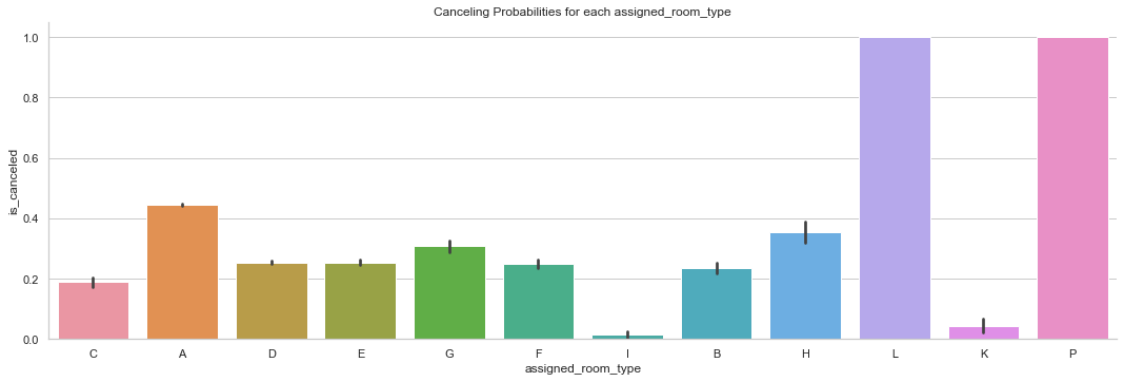
- Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g. overbooking) or by customer request. Code is presented instead of designation for anonymity reasons

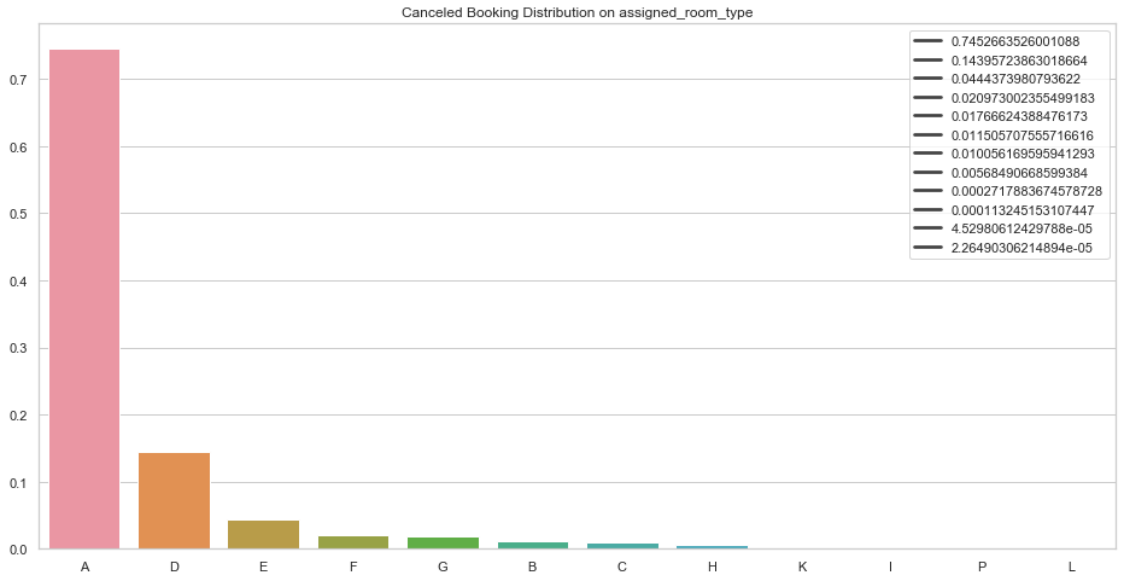
[187]:
```
data['assigned_room_type'].unique()
```

[187]:
```
array(['C', 'A', 'D', 'E', 'G', 'F', 'I', 'B', 'H', 'L', 'K', 'P'],
      dtype=object)
```

[188]:
```
count_cat_prob_plot('assigned_room_type', data)
```


Count Plot for assigned_room_type
count
70000
60000
50000
40000
30000
20000
10000
0
C
A
D
E
G
F
I
B
H
L
K
P
assigned_room_type


Canceling Probabilities for each assigned_room_type
is_canceled
1.0
0.8
0.6
0.4
0.2
0.0
C
A
D
E
G
F
I
B
H
L
K
P
assigned_room_type



[189]:
```
### assigned room and reserved_room looks like similar columns that's why we␣
↪will remove it.

columns_to_dummy.append('assigned_room_type')
```

0.0.24 21 - booking_changes: Section ??

- 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

[190]:
```
data['booking_changes'].describe()
```

[190]:
```
count    118895.000000
mean          0.221153
std           0.652765
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          21.000000
Name: booking_changes, dtype: float64
```

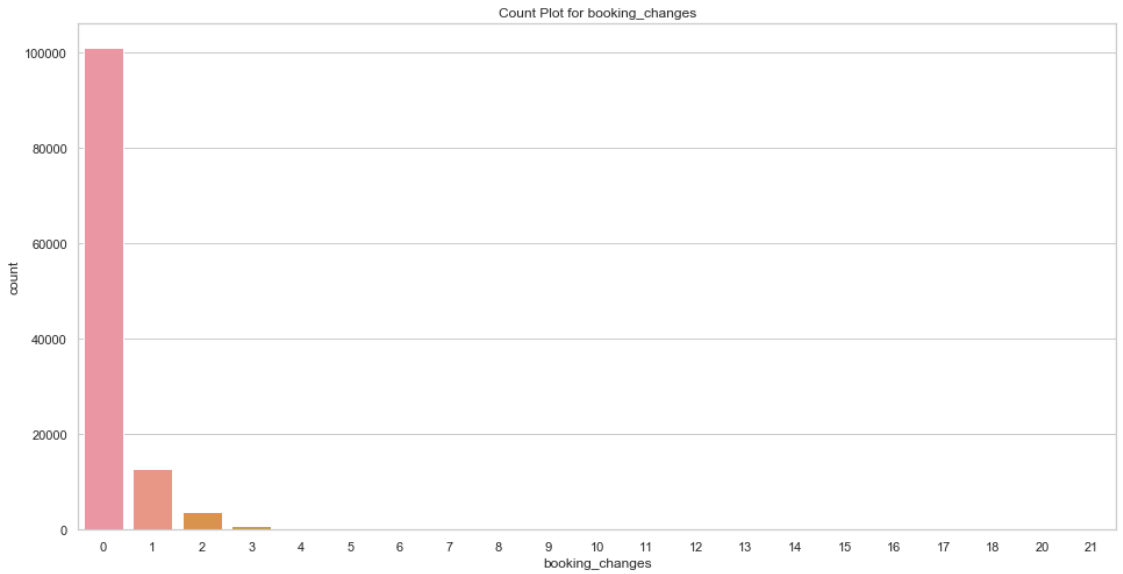
[191]:
```
data['booking_changes'].unique()
```

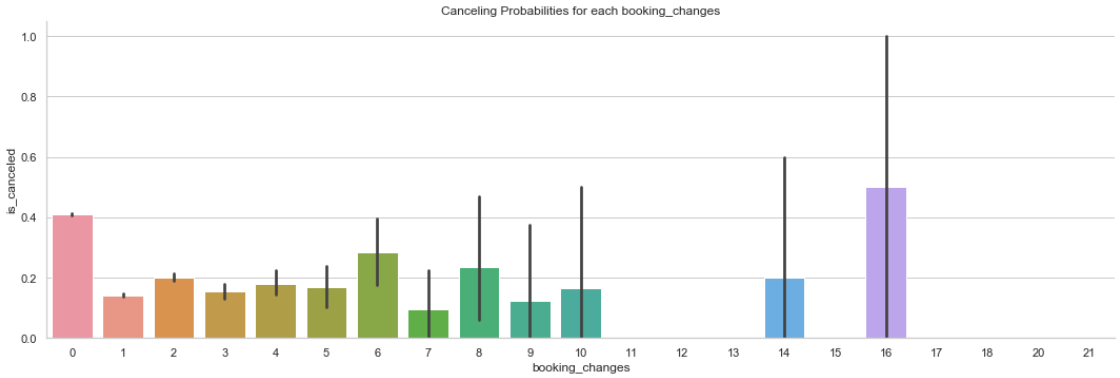
[191]:
```
array([ 3,  4,  0,  1,  2,  5, 17,  6,  8,  7, 10, 16,  9, 13, 12, 20, 14,
       15, 11, 21, 18])
```

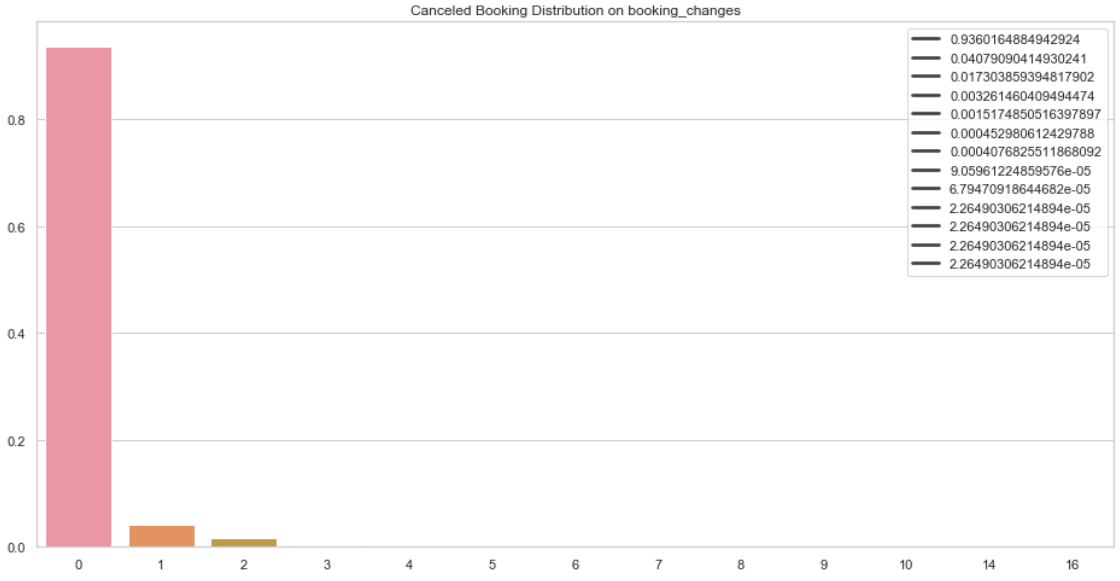
[192]:
```
data['booking_changes'].value_counts()
```

[192]:
```
0      100902
1       12637
2        3789
3         925
4         375
5         118
6          63
7          31
8          17
9           8
10          6
13          5
14          5
15          3
11          2
12          2
16          2
17          2
20          1
18          1
21          1
Name: booking_changes, dtype: int64
```

[193]:
```
count_cat_prob_plot('booking_changes', data)
```







0.0.25 22 - deposit_type: Section ??

- Indication on if the customer made a deposit to guarantee the booking. No Deposit – no deposit was made;

[194]:
```
data['deposit_type'].describe()
```

[194]:
```
count         118895
unique             3
top       No Deposit
freq          104160
Name: deposit_type, dtype: object
```

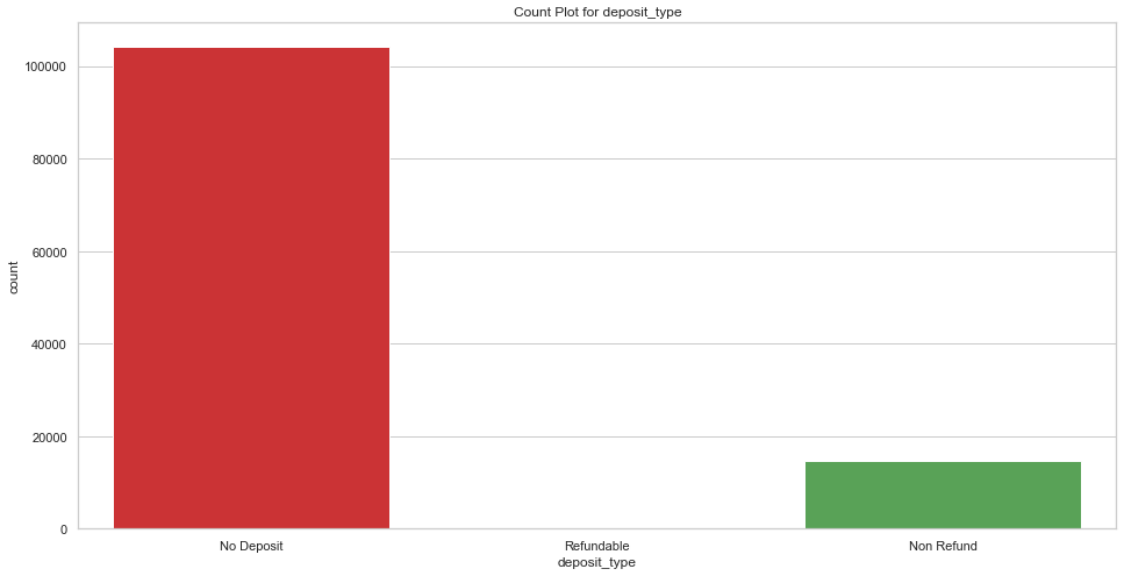
[195]:
```
data['deposit_type'].unique()
```

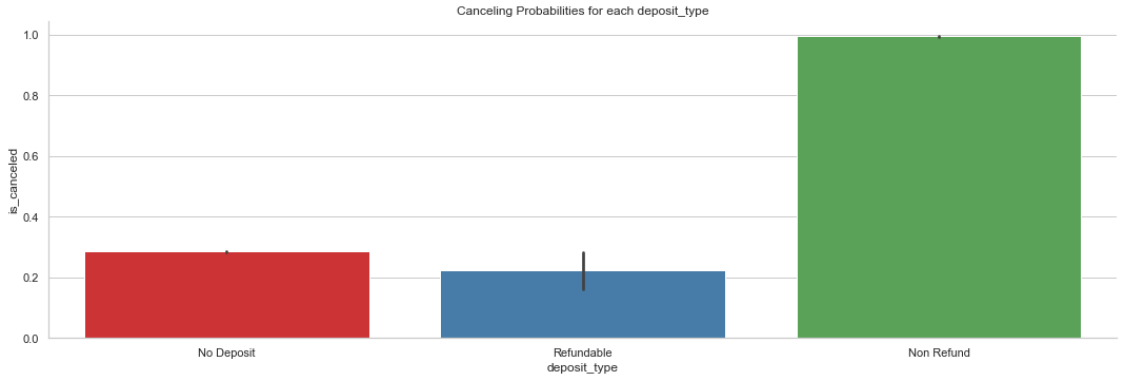
[195]:
```
array(['No Deposit', 'Refundable', 'Non Refund'], dtype=object)
```

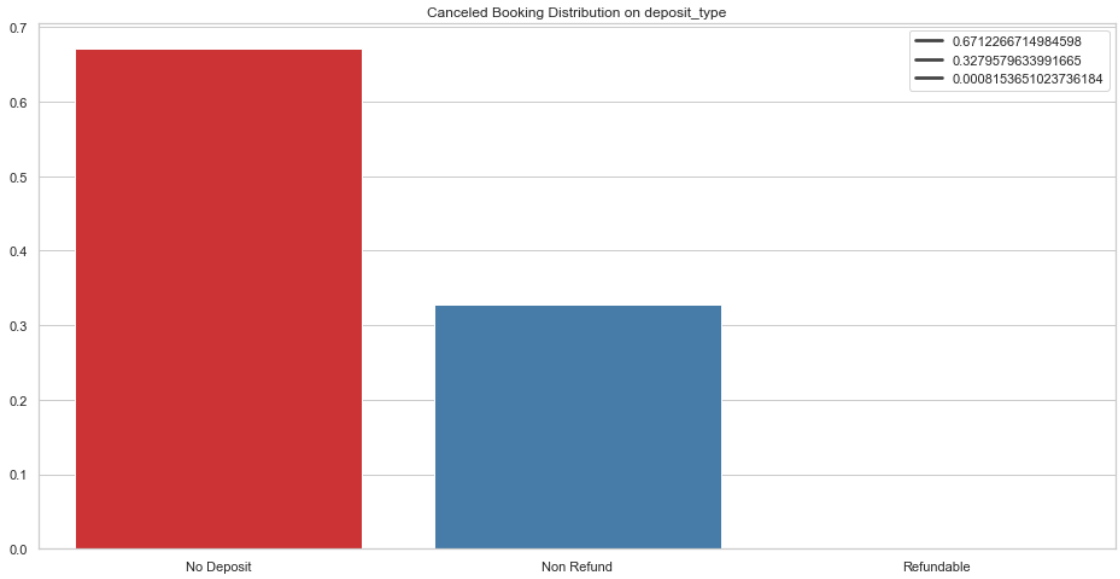
[196]:
```
data.groupby('deposit_type')['is_canceled'].value_counts(normalize=True)
```

[196]:
```
deposit_type  is_canceled
No Deposit    0              0.715476
              1              0.284524
Non Refund    1              0.993618
              0              0.006382
Refundable    0              0.777778
              1              0.222222
Name: is_canceled, dtype: float64
```

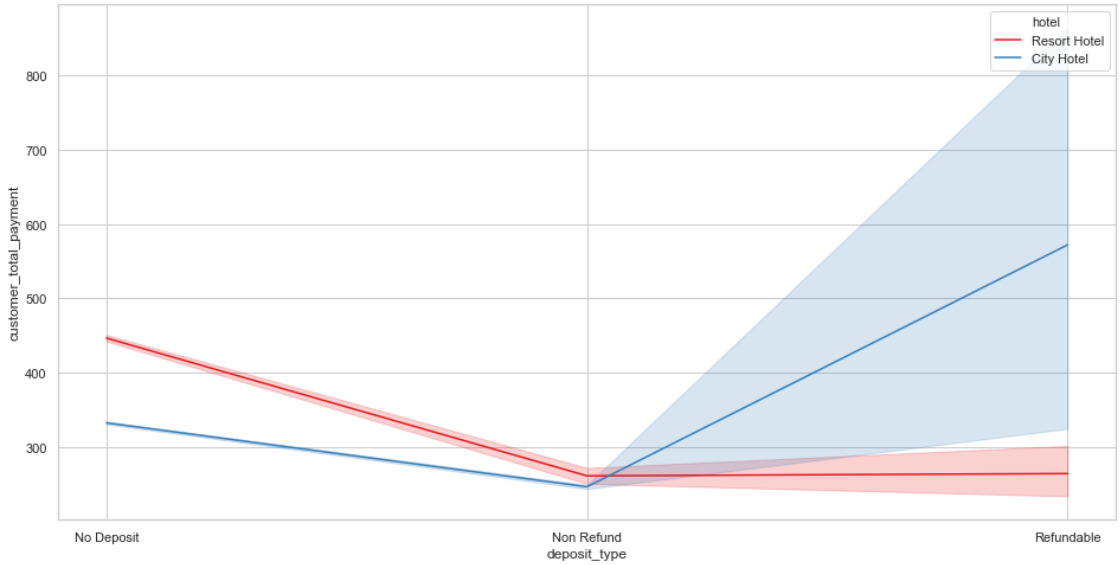
[197]:
```
count_cat_prob_plot('deposit_type', data)
```







[198]:
```
g = sns.lineplot(x='deposit_type', y='customer_total_payment', data=data,␣
↪hue='hotel')
```



- Total payment has least value for non-refund deposity type

[199]:
```
columns_to_dummy.append('deposit_type')
```

0.0.26 23 - days_in_waiting_list: Section ??

- Number of days the booking was in the waiting list before it was confirmed to the customer.

[200]:
```
data['days_in_waiting_list'].describe()
```

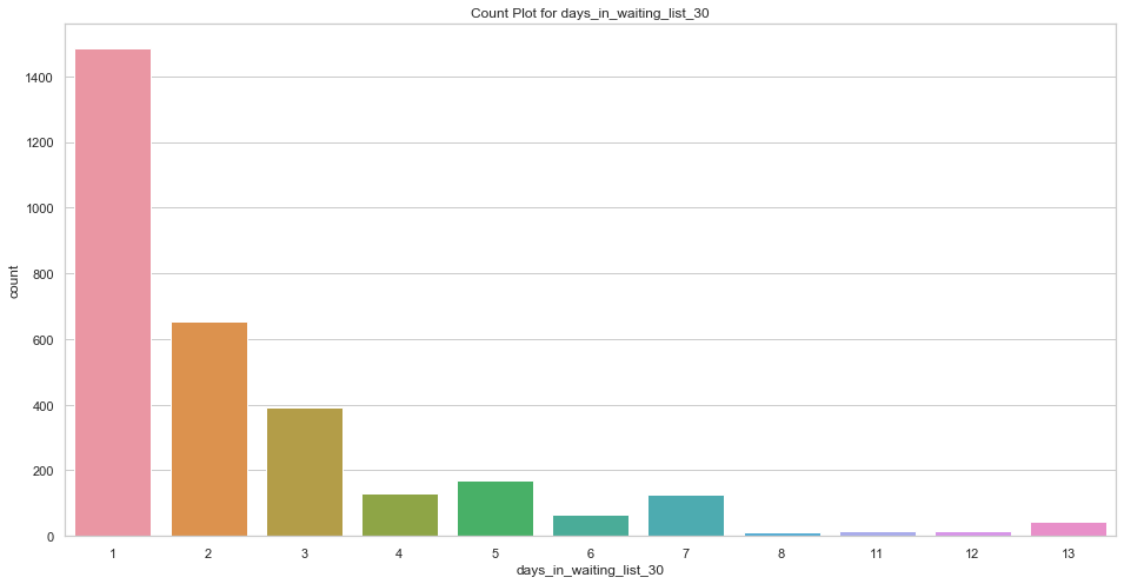
[200]:
```
count    118895.000000
mean          2.330813
std          17.630671
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max         391.000000
Name: days_in_waiting_list, dtype: float64
```

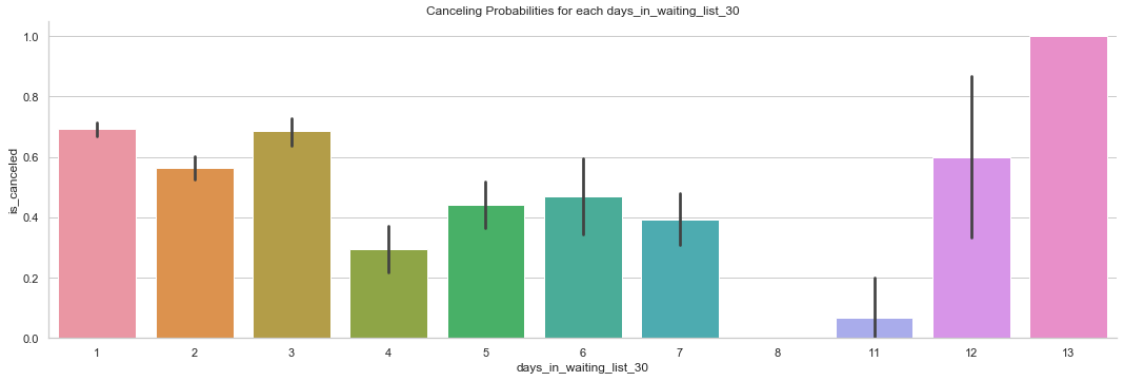
[201]:
```
data['days_in_waiting_list'].unique()
```

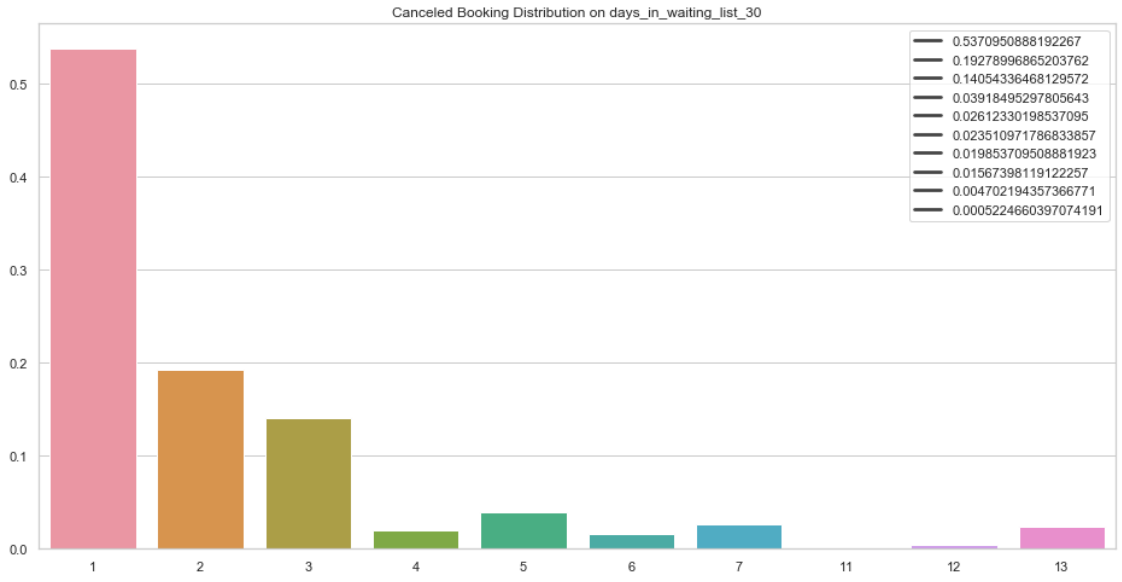
[201]:
```
array([  0,  50,  47,  65, 122,  75, 101, 150, 125,  14,  60,  34, 100,
        22, 121,  61,  39,   5,   1,   8, 107,  43,  52,   2,  11, 142,
       116,  13,  44,  97,  83,   4, 113,  18,  20, 185,  93, 109,   6,
        37, 105, 154,  64,  99,  38,  48,  33,  77,  21,  80,  59,  40,
        58,  89,  53,  49,  69,  87,  91,  57, 111,  79,  98,  85,  63,
        15,   3,  41, 224,  31,  56, 187, 176,  71,  55,  96, 236, 259,
       207, 215, 160, 120,  30,  32,  27,  62,  24, 108, 147, 379,  70,
        35, 178, 330, 223, 174, 162, 391,  68, 193,  10,  76,  16,  28,
         9, 165,  17,  25,  46,   7,  84, 175, 183,  23, 117,  12,  54,
        26,  73,  45,  19,  42,  72,  81,  92,  74, 167,  36])
```

[202]:
```
data['days_in_waiting_list_30'] = data['days_in_waiting_list'] // 30
```

[203]:
```
count_cat_prob_plot('days_in_waiting_list_30',␣
↪data[data['days_in_waiting_list_30'] > 0])
```







0.0.27 24 - customer_type: Section ??

- 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;

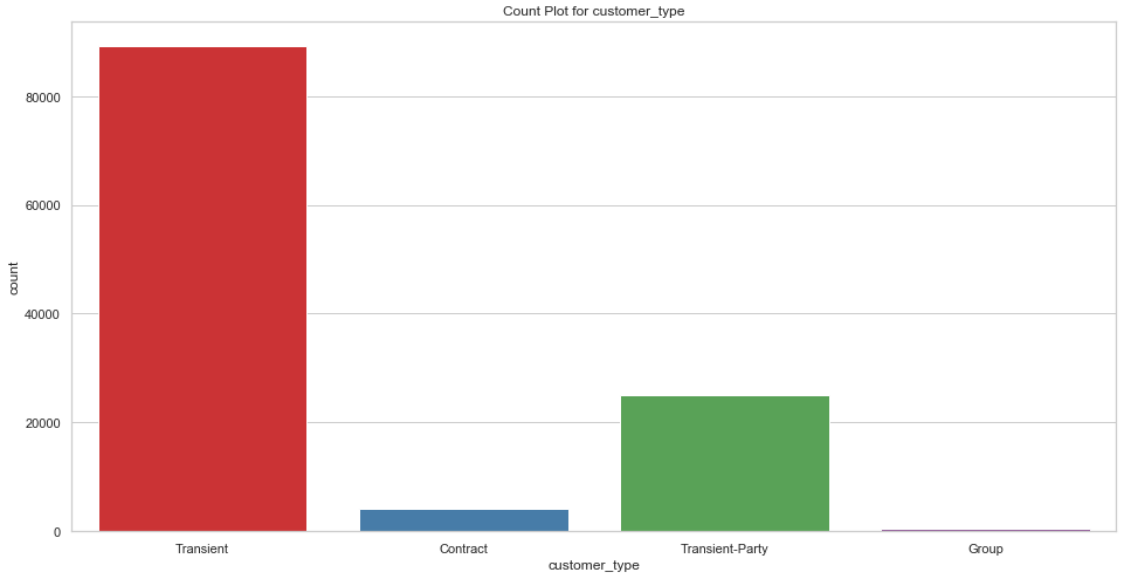
Transient – when the booking is not part of a group or contract, and is not associated to other transient booking;

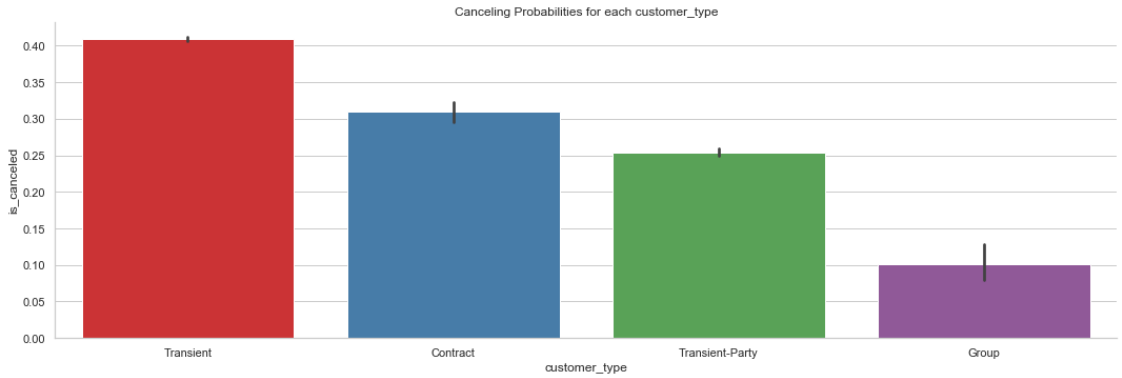
Transient-party – when the booking is transient, but is associated to at least other transient booking

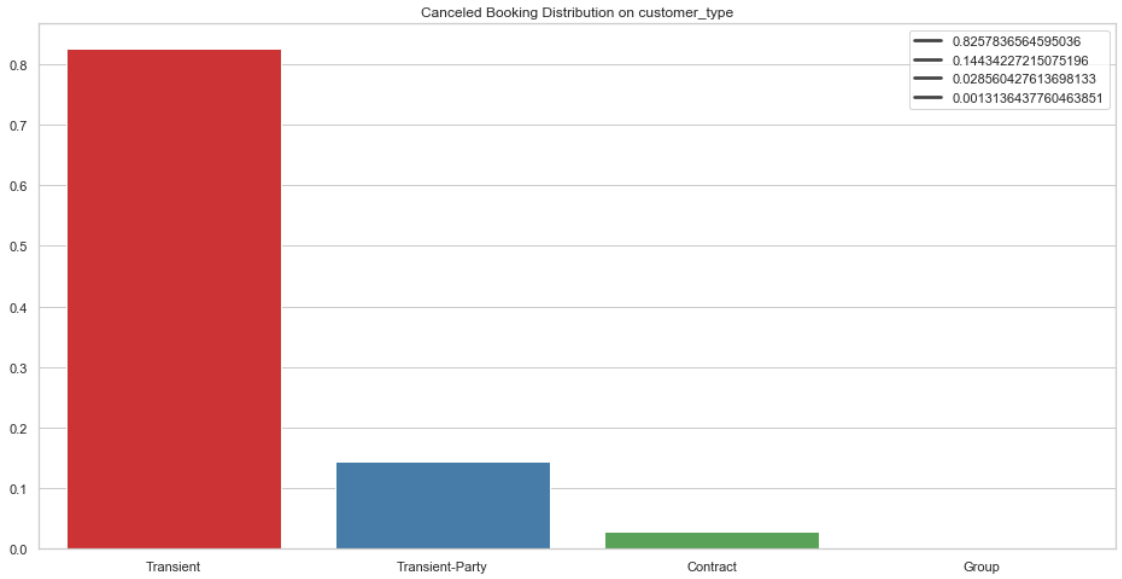
[204]:
```
data['customer_type'].unique()
```

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

[205]:
```
count_cat_prob_plot('customer_type', data)
```







[206]:
```
columns_to_dummy.append('customer_type')
```

0.0.28 25 - adr: Section ??

- Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
  A hotel's ADR, Average Daily Rate, is the measure of the average rate paid per room that's occupied at the property. Ultimately, it's a KPI that helps hoteliers identify their room rates from a day-to-day perspective. ADR is calculated to have an understanding of a hotel's profits and performance.

[207]:
```
data['adr'].describe()
```

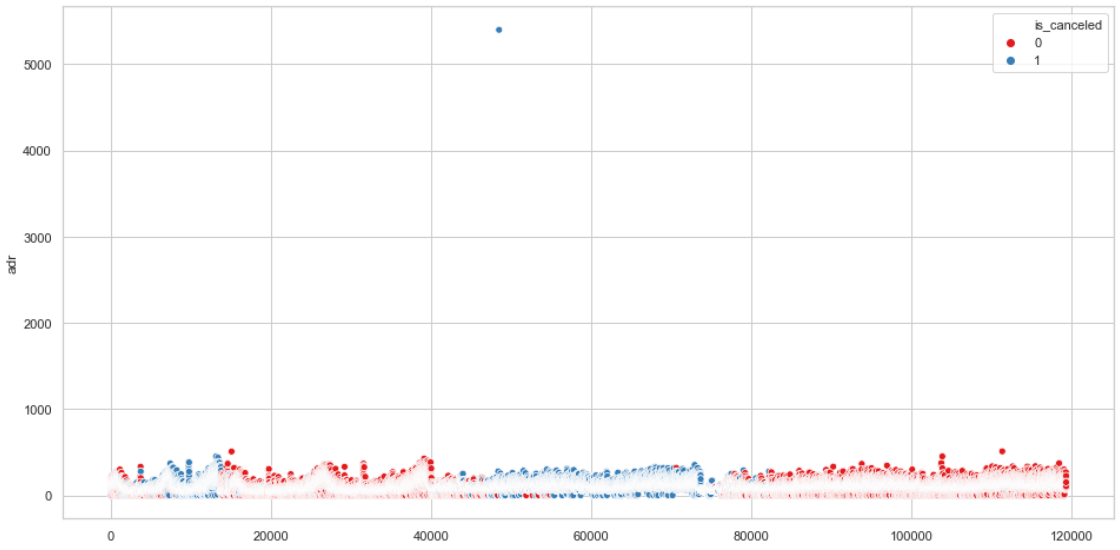
[207]:
```
count    118895.000000
mean        102.003187
std          50.486388
min          -6.380000
25%          70.000000
50%          95.000000
75%         126.000000
max        5400.000000
Name: adr, dtype: float64
```

[208]:
```
data[data['adr'] < 0]['adr']
```

[208]:
```
14969   -6.38
Name: adr, dtype: float64
```

[209]:
```
data.drop(data[data['adr'] < 0].index, axis=0, inplace=True)
```

[210]:
```
g = sns.scatterplot(x=data.index, y=data['adr'], hue=data['is_canceled'])

plt.show(g)
```



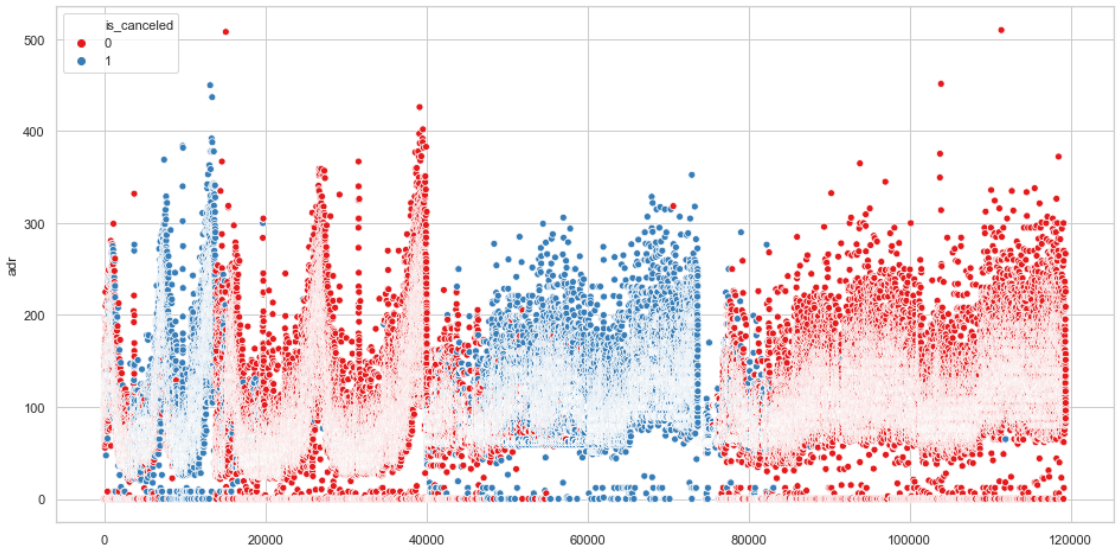
[211]:
```
# We have an outliar for adr column which greater than 5000
```

[212]:
```
data[data['adr'] > 1000]
```

[212]:
```
            hotel  is_canceled  lead_time  arrival_date_year  \
48515  City Hotel            1         35               2016

      arrival_date_month  arrival_date_week_number  arrival_date_day_of_month  \
48515              March                        13                         25

       stays_in_weekend_nights  stays_in_week_nights  adults  …  \
48515                        0                     1       2  …

       lead_time_60  lead_time_120 lead_time_360 arrival_date_weekth_in_month  \
48515             0              0             0                           1

      seasons arrival_date_day_of_week  stays_total  customer_cancel_ratio  \
48515  spring                  Monday            1                    0.0

       room_changed days_in_waiting_list_30
48515             0                       0

[1 rows x 43 columns]
```

[213]:
```
data.drop(data[data['adr'] > 1000].index, axis=0, inplace=True)
```

[214]:
```
g = sns.scatterplot(x=data.index, y=data['adr'], hue=data['is_canceled'])

plt.show(g)
```



- We have to be carefull when we train our model because room booking separated by index.

0.0.29 26 - required_car_parking_spaces: Section ??

- Number of car parking spaces required by the customer

[215]:
```
data['required_car_parking_spaces'].describe()
```

[215]:
```
count    118893.000000
mean          0.061888
std           0.244177
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max           8.000000
Name: required_car_parking_spaces, dtype: float64
```

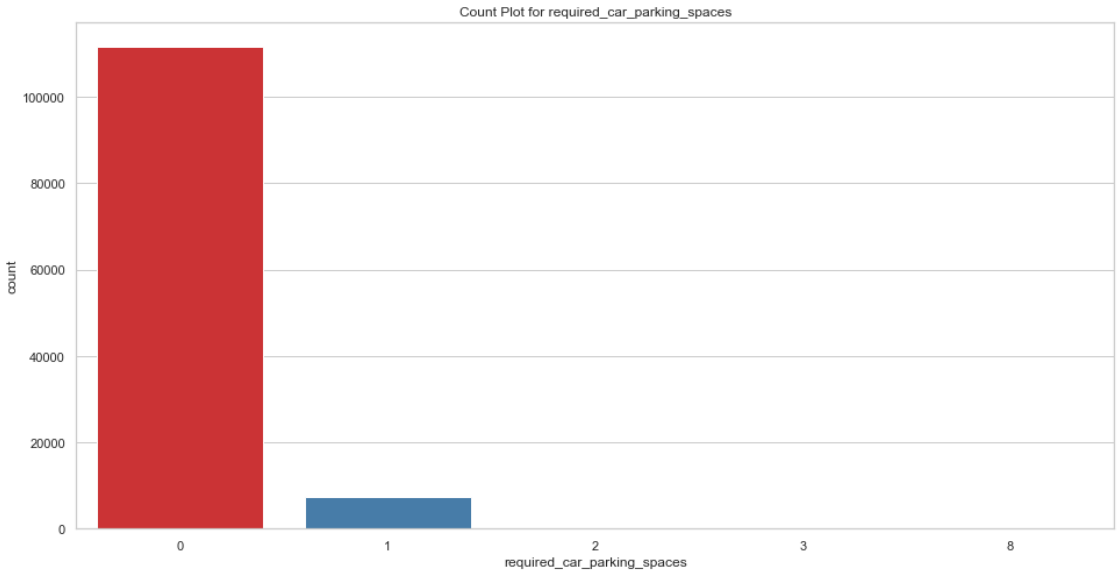
[216]:
```
data['required_car_parking_spaces'].unique()
```

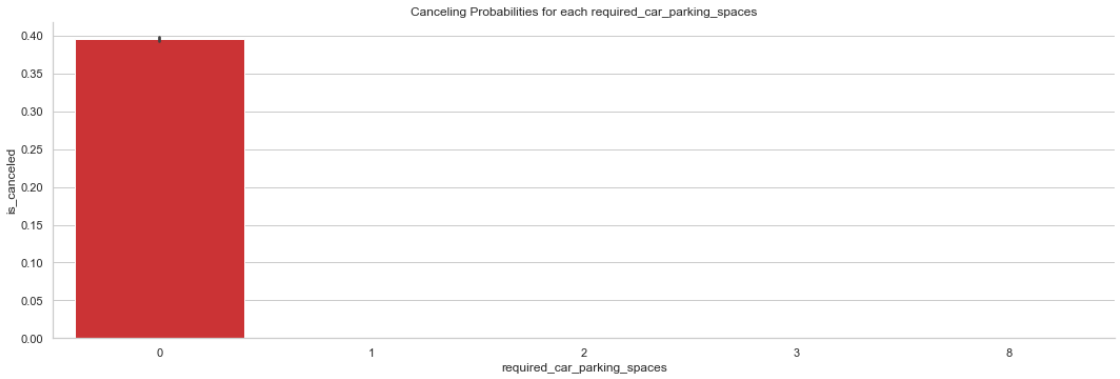
[216]:
```
array([0, 1, 2, 8, 3])
```

[217]:
```
data['required_car_parking_spaces'].value_counts()
```

[217]:
```
0    111583
1      7277
2        28
3         3
8         2
Name: required_car_parking_spaces, dtype: int64
```

[218]:
```
count_cat_prob_plot('required_car_parking_spaces', data)
```



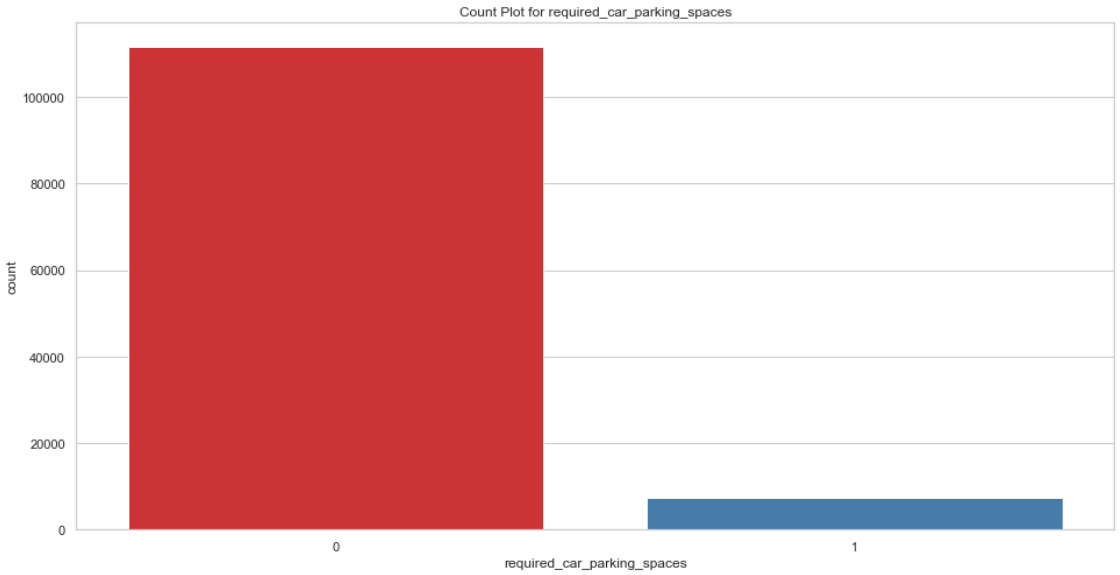


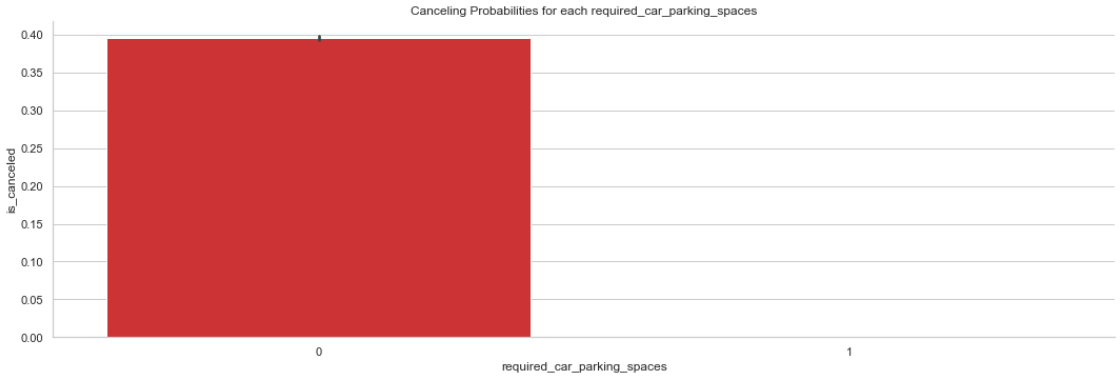
- Customers who required parking slots don't cancel their bookings.

[219]:
```
def car_parking_space(required_park):

    if required_park > 0:
        return 1
    else:
        return 0
```

[220]:
```
data['required_car_parking_spaces'] = data['required_car_parking_spaces'].
↪apply(car_parking_space)
```

[221]:
```
count_cat_prob_plot('required_car_parking_spaces', data)
```


Count Plot for required_car_parking_spaces
count
100000
80000
60000
40000
20000
0
0
1
required_car_parking_spaces


Canceling Probabilities for each required_car_parking_spaces
is_canceled
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
1
required_car_parking_spaces

[222]:
```
columns_to_remove.append('required_car_parking_spaces')
```

0.0.30 27 - total_of_special_requests: - Section ??

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

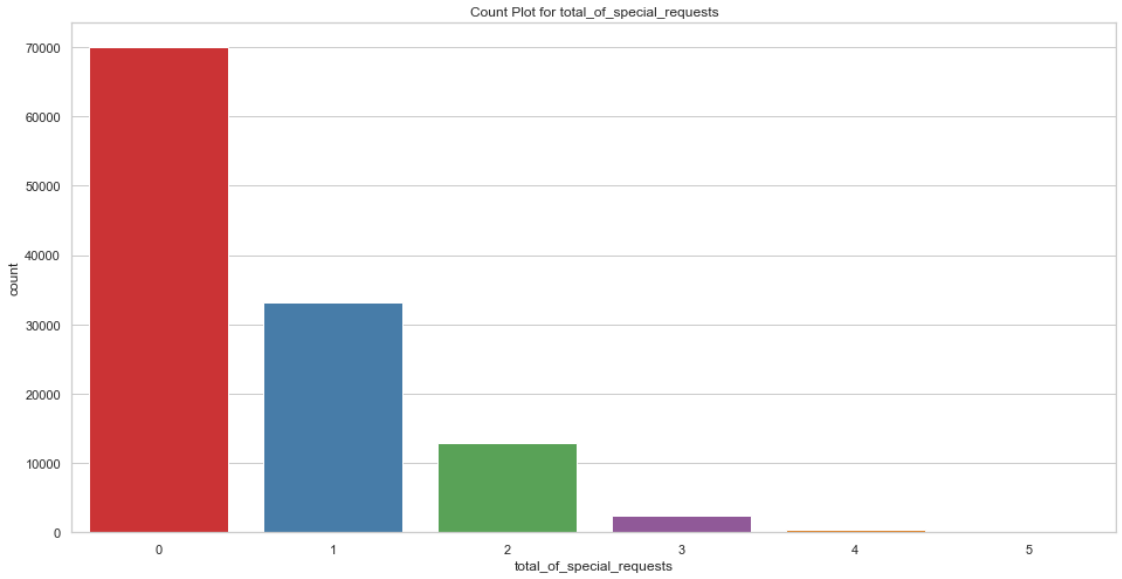
[223]:
```
data['total_of_special_requests'].unique()
```

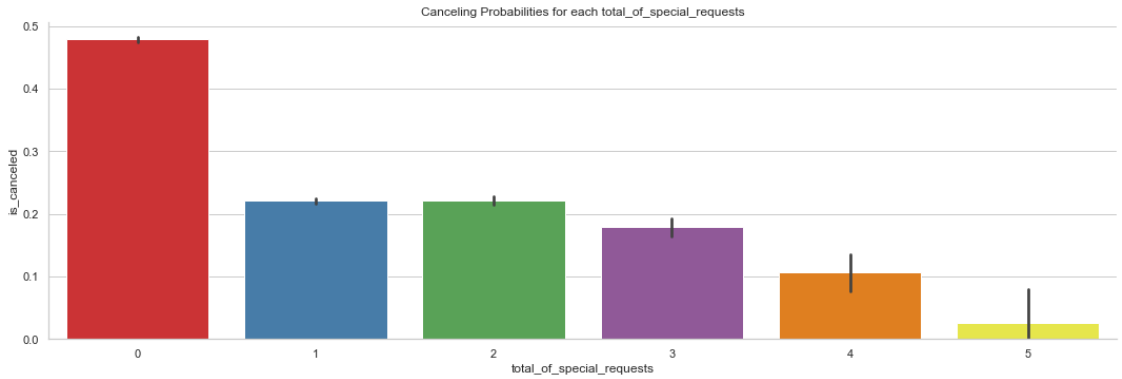
[223]:
```
array([0, 1, 3, 2, 4, 5])
```

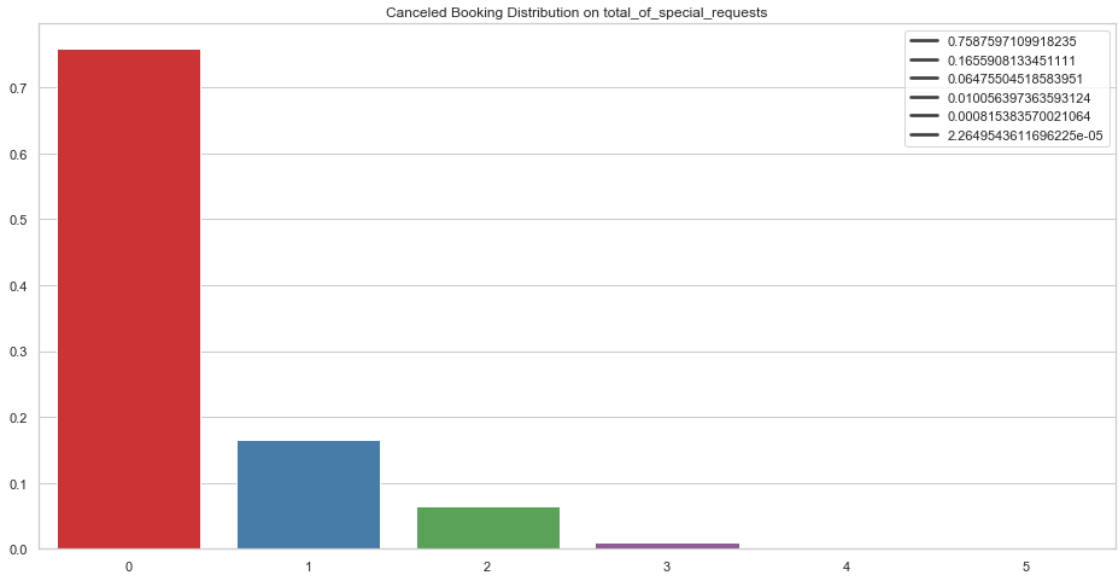
[224]:
```
data['total_of_special_requests'].value_counts()
```

[224]:
```
0    69988
1    33119
2    12922
3     2487
4      339
5       38
Name: total_of_special_requests, dtype: int64
```

[225]:
```
count_cat_prob_plot('total_of_special_requests', data)
```

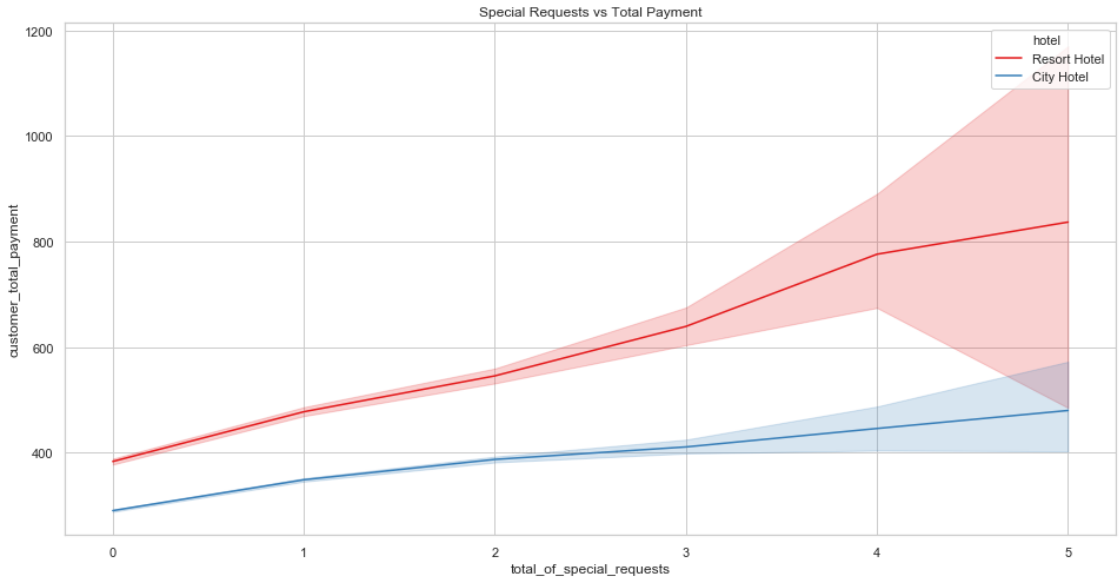

Count Plot for total_of_special_requests
70000
60000
50000
40000
30000
20000
10000
0
count
0
1
2
3
4
5
total_of_special_requests


Canceling Probabilities for each total_of_special_requests
0.5
0.4
0.3
0.2
0.1
0.0
is_canceled
0
1
2
3
4
5
total_of_special_requests



- Increasing special requests descreases canceling probabilities.

[226]:
```
g = sns.lineplot(x='total_of_special_requests', y='customer_total_payment',␣
↪data=data, hue='hotel')

plt.title('Special Requests vs Total Payment')
plt.show(g)
```



- Increase in special requests increases total payment

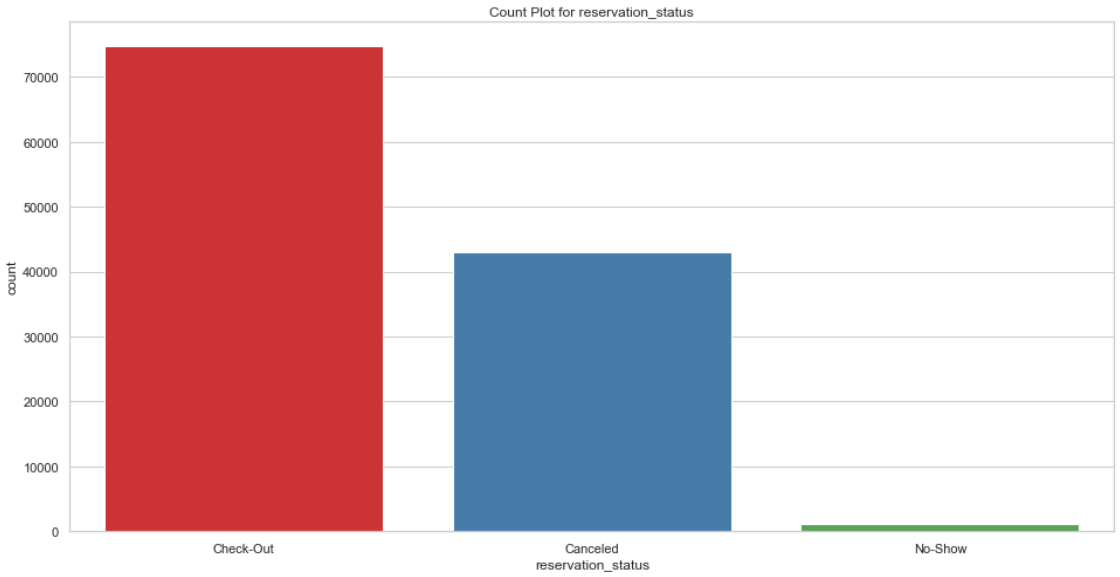
0.0.31 28 - reservation_status: - Section ??

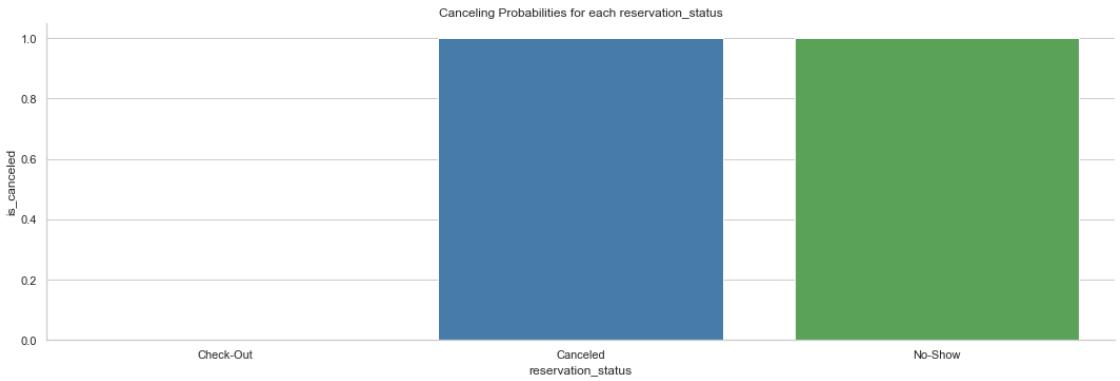
- Reservation last status, assuming one of three categories:
  Canceled – booking was canceled by the customer;
  Check-Out – customer has checked in but already departed;
  No-Show – customer did not check-in and did inform the hotel of the reason why

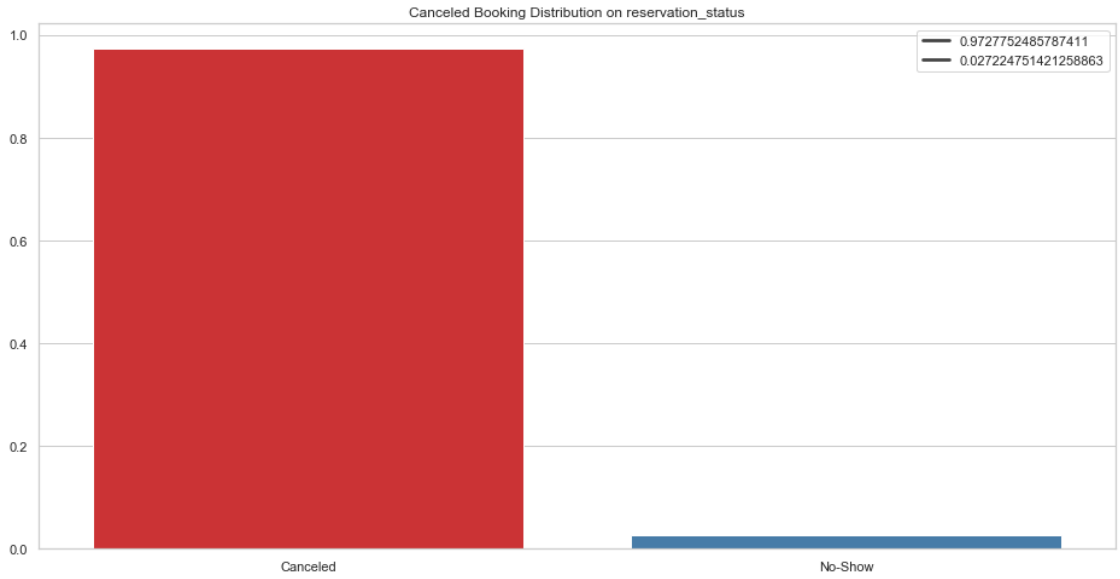
[227]:
```
data['reservation_status'].unique()
```

[227]:
```
array(['Check-Out', 'Canceled', 'No-Show'], dtype=object)
```

[228]:
```
count_cat_prob_plot('reservation_status', data)
```







[229]:
```
columns_to_remove.append('reservation_status')
```

[230]:
```
columns_to_remove.append('reservation_status_date')
```

Finalize Dataset and Save

[231]:
```
from sklearn.utils import shuffle

data = shuffle(data).reset_index(drop=True)
```

[232]:
```
data.shape
```

[232]:
```
(118893, 43)
```

[233]:
```
columns_to_remove
```

[233]:
```
['lead_time_60',
 'lead_time_30',
 'lead_time_120',
 'lead_time_360',
 'seasons',
 'arrival_date_day_of_month',
 'previous_bookings_not_canceled',
 'previous_cancellations',
 'required_car_parking_spaces',
 'reservation_status',
 'reservation_status_date']
```

[234]:
```
columns_to_dummy
```

[234]:
```
['hotel',
 'arrival_date_year',
 'arrival_date_weekth_in_month',
 'arrival_date_month',
 'arrival_date_day_of_week',
 'meal',
 'country',
 'market_segment',
 'distribution_channel',
 'reserved_room_type',
 'assigned_room_type',
 'deposit_type',
 'customer_type']
```

[235]:
```
cleaned_data = data.drop(columns=columns_to_remove, axis=1)
```

[236]:
```
cleaned_data = pd.get_dummies(cleaned_data, columns=columns_to_dummy)
```

[237]:
```
cleaned_data.shape # because of dummy columns we have too many columns.
```

[237]:
```
(118893, 269)
```

[238]:
```
cleaned_data.info() # we removed all object (str) data.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 118893 entries, 0 to 118892
Columns: 269 entries, is_canceled to customer_type_Transient-Party
dtypes: float64(4), int64(15), uint8(250)
memory usage: 45.6 MB
```

[239]:
```
cleaned_data.to_csv('../Data/hotel_bookings_cleaned.csv', index=False)
```

[ ]:
```

```