**Report: Hotel Bookings Dataset**

**Name : Lavina Parulekar**

**Roll Number: 250240325011**

**Problem Statement:**Uncovering Patterns in Hotel Booking Data for Operational Efficiency and Revenue Growth.

***1. Data Cleaning and Preprocessing***

→ imported pandas pandas library

import pandas as pd

→Loaded the dataset and checked number of rows and columns.

df.shape

* In the given dataset (119390, 32) 119390 rows and 32 columns are present.

→Previewed first 5 rows from the dataset.

df.head()

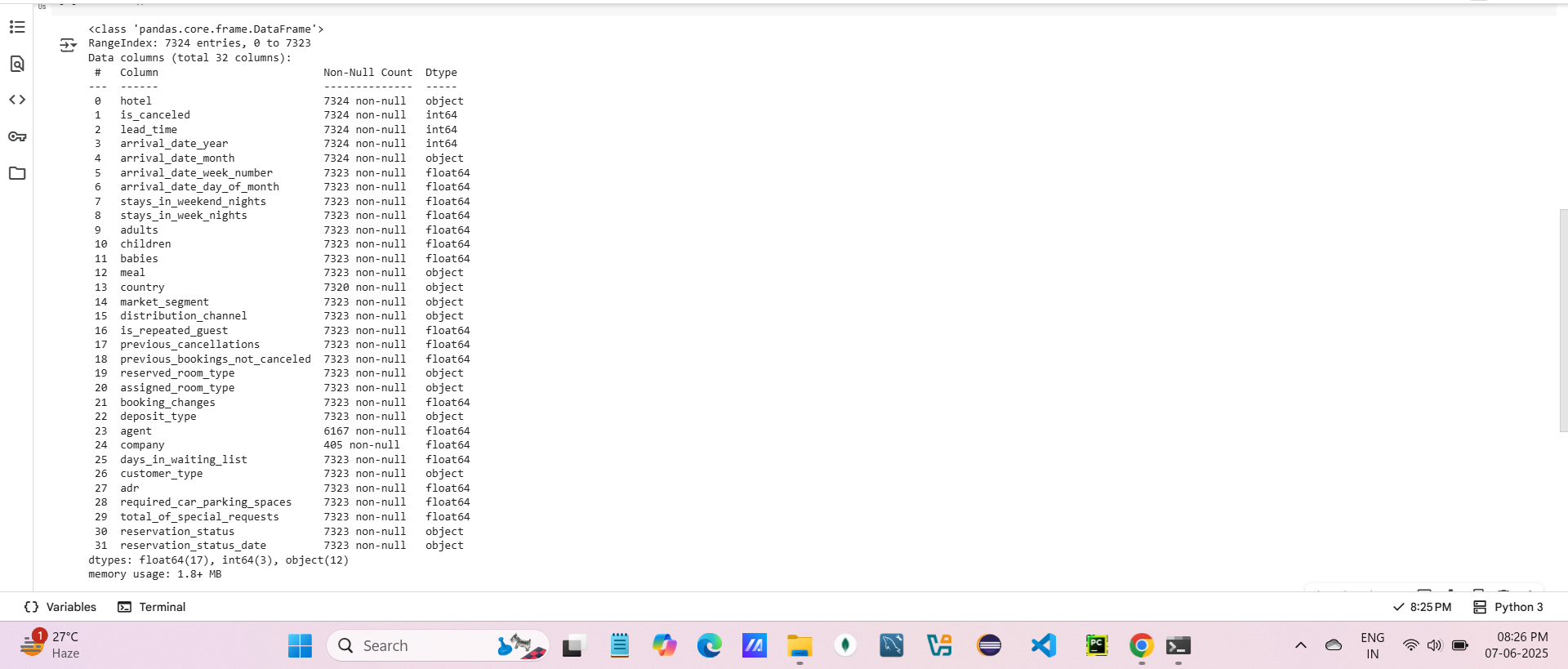


* Some columns was being truncated so by using following command we can able to see all the columns

pd.set\_option('display.max\_columns', None)

→To see datatypes of all the columns and no -null values following command used.

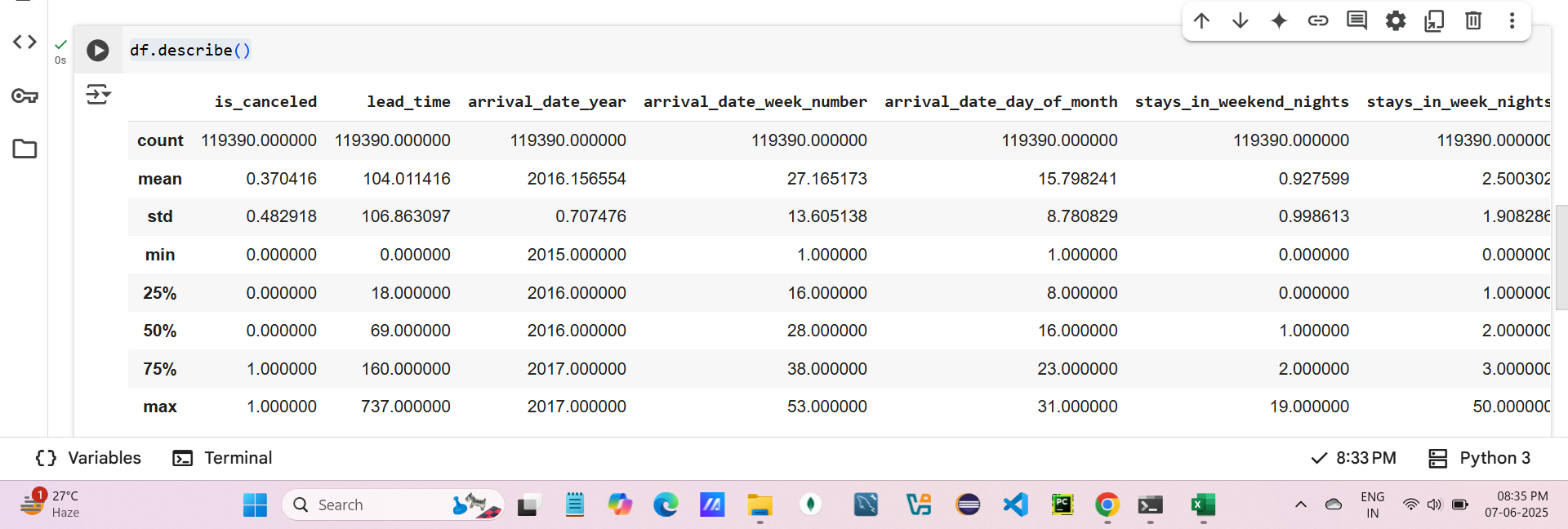
df.info()



**From the output of above command i could see country ,Company,agent contain null values.**

**→** To find the statistical summary of all numeric columns following Command is used.

df.describe()



Output of this includes measure of central tendency,distribution of the data and spread of the data.Following are the values included in the output.

Count - total number of rows

Mean - sum of all values in col / total numb of col

Std - It is distance of each datapoint from the center.

Variance - tells us how values in dataset differs from the mean.

Min - minimum value from that particular col

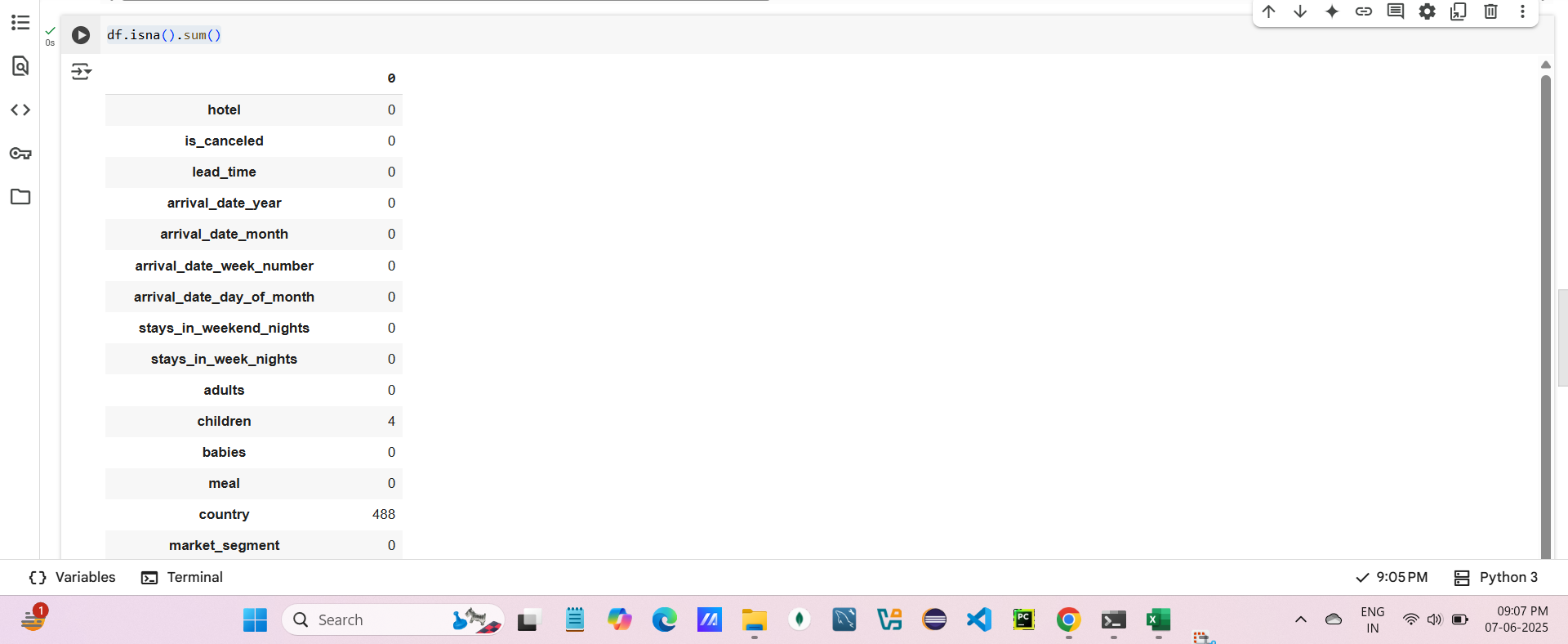
Max - maximum value from that col

25%,50%,75% - Tells us how many % values are below that 25%,50% and 75 %

Eg. 25 % of lead time is 18 means 25 % values from the dataset are less than or equal to 18.

→ Checked how many missing values are present in each column.

df.isna().sum()



Following are columns that contains null values:

1)Country - 488

2)agent - 16340

3)Company - 112593

***1(A):Handling missing values***

1. **Company , agent Columns**

**df = df.drop(['company','agent'],axis=1)**

Total values = (112593/119390)\*100

Approximately 90% values are missing in company col.

Hence it is removed from the dataset.

Also company col does have any significant impact on response variable.

Similarly

Total values = (16340/119390)\*100

Approximately 13% values are missing in company col.

Hence it is removed from the dataset.

Also agent col does have any significant impact on response variable.

**2)Country**

Total values = (400/119390)\*100

Only 0.4 % values are missing also this col has impact on response variable so we can substitute it with another values.

So country col contains categorical data hence we can substitute missing values with mode.

df.country.mode()

df.fillna('PRT',inplace=True)

Inplace = modify the original df directly

***1(B):Handling the duplicates.***

df.duplicated().sum()

There are 32039 duplicate values are present.

→Dropped duplicate values.

df.drop\_duplicates(inplace=True)

After dropping duplicates there (87351) rows are present



**Impact duplicates present in dataset:**

Example: If the same booking is logged twice:

In dataset ADR (Average Daily Rate) will be skewed.

It looks like the hotel earned more than it actually did.

***1(C):Creating derived column:***

df['total\_stay'] = df['stays\_in\_weekend\_nights'] + df['stays\_in\_week\_nights']

Here 2 separate cols are given for number days stated in hotel.

So combine it by using derived column ‘total\_stay’

cols= list(df.columns)

last = cols.pop()

cols.insert(3, last)

df = df[cols]

***1(D):Standardize date column.***

→ Created arrival\_date column.

df['arrival\_date'] = pd.to\_datetime(df['arrival\_date\_year'].astype(str) + '-' +

df['arrival\_date\_month'] + '-' +

df['arrival\_date\_day\_of\_month'].astype(str),

format='%Y-%B-%d')



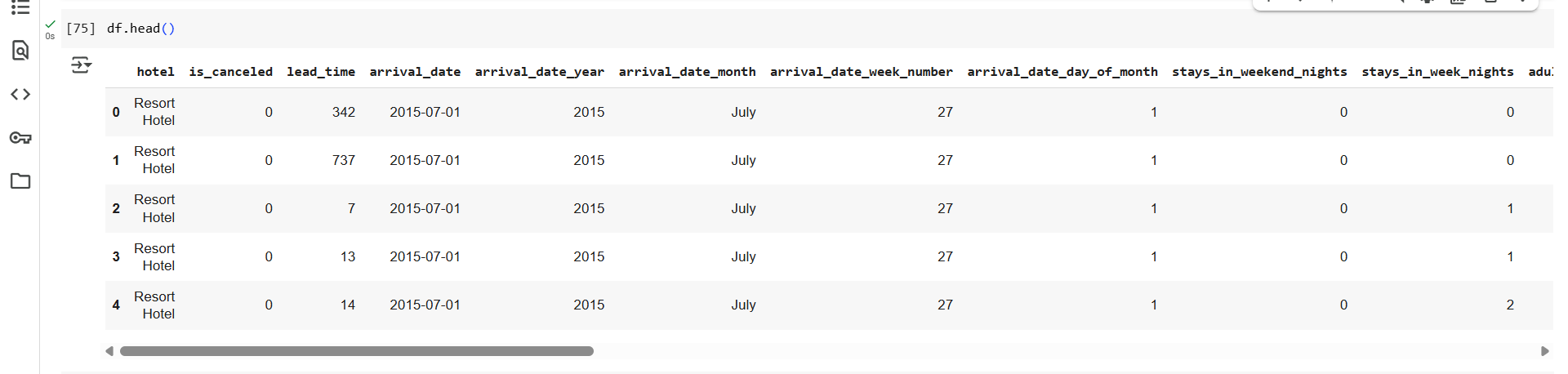
→shift that col to 3 index.

cols= list(df.columns)

last = cols.pop()

cols.insert(3, last)

df = df[cols]



***1(E):Handling the outliers.***

→Handling outliers is most imp.part of data cleaning.

→Outliers are those datapoints that has huge impact on mean.

→By using following IQR method find out outliers:

Q1 = df['adr'].quantile(0.25)

Q3 = df['adr'].quantile(0.75)

IQR = Q3 - Q1

lower = Q1 - 1.5 \* IQR

upper = Q3 + 1.5 \* IQR

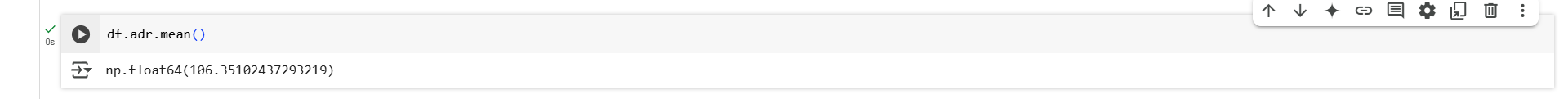
→Find out the indices where the outliers are present.

a = df[~((df['adr'] >= lower) & (df['adr'] <= upper))].index

a

→Mean of col adr before removing the outliers.

df.adr.mean()



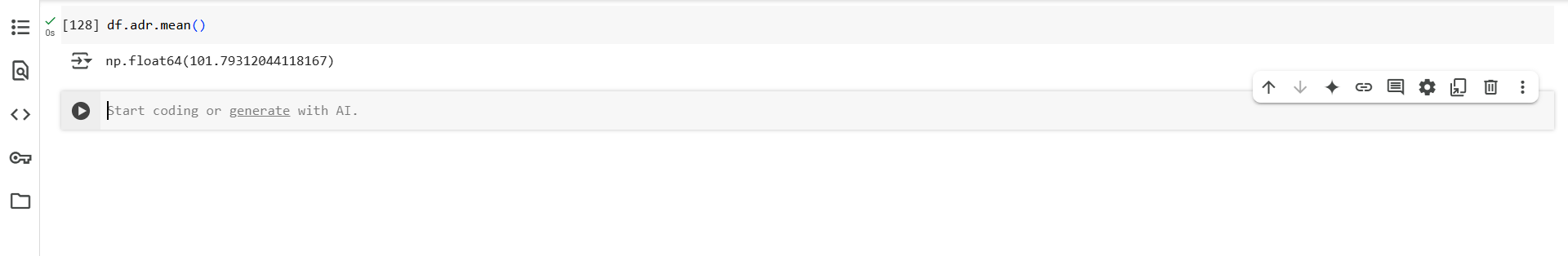
Here mean = 106.35102437293219

→Creating dataset after removing outliers.

df = df[(df['adr'] >= lower) & (df['adr'] <= upper)]



→Mean after removing outliers.



Here mean = 101.79312044118167

**Diff :** 106.35102437293219 - 101.79312044118167

=4.557903931750516

***1(F):Removing the invalid entries or booking.***

→In the given dataset there are some entries where adult count is 0.This is not the valid booking.

→finding out what are those rows.

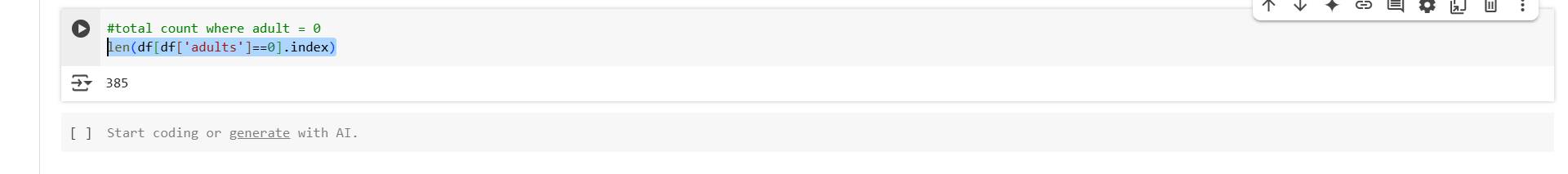
df[df['adults']==0].index

→o/p



→Total number of such rows.

len(df[df['adults']==0].index)



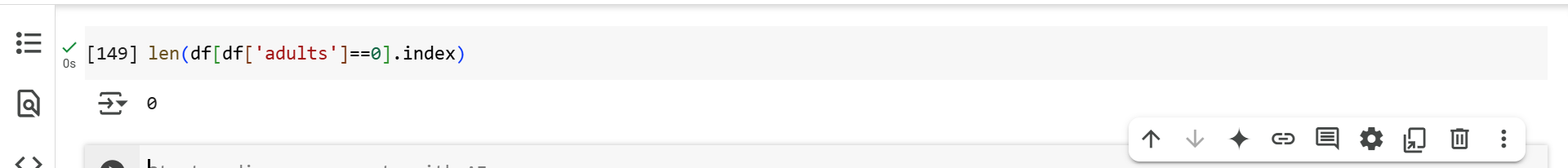
o/p: There are 385 rows present where adult count is 0.

→Removed those rows where adult count is 0.

df = df[df['adults']>0]



After removing total rows within the dataset is 84478.



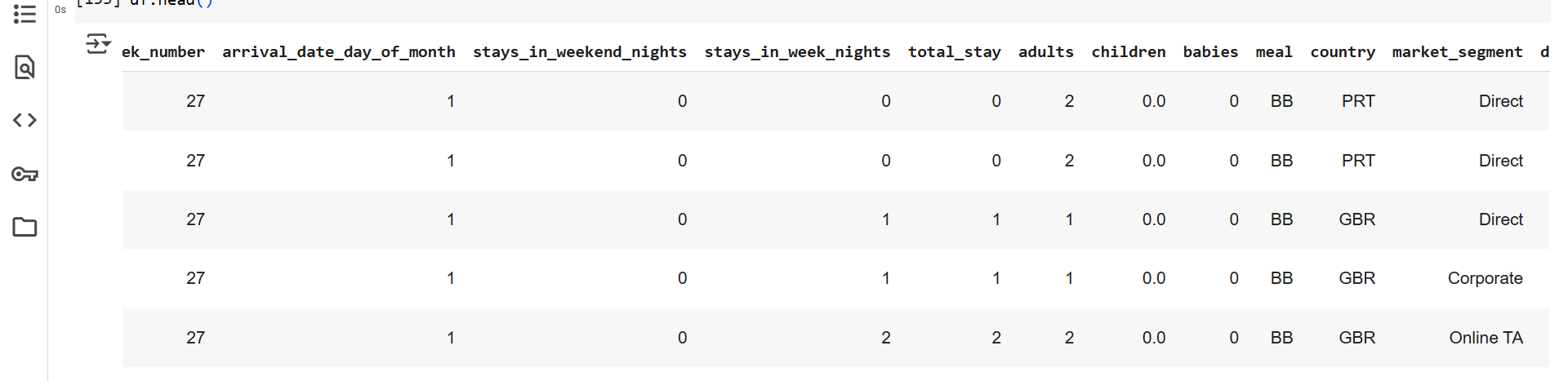
→Moved total\_stay col to index 10.

cols= list(df.columns)

last = cols.pop()

cols.insert(10, last)

df = df[cols]



***2. Exploratory Data Analysis.***

**1)Univariate Analysis**

→Purpose of univariate analysis is to analyze single variable,understanding it’s distribution ,central tendency spread.

1)bar-chart

→Graph is of total booking count for each hostels.



City Hotel receives more bookings than Resort Hotel.

* City Hotel gets more bookings, so the hotel company can spend more money on it — like improving rooms, hiring more staff, or even opening more city hotels in other places.

2)Histogram

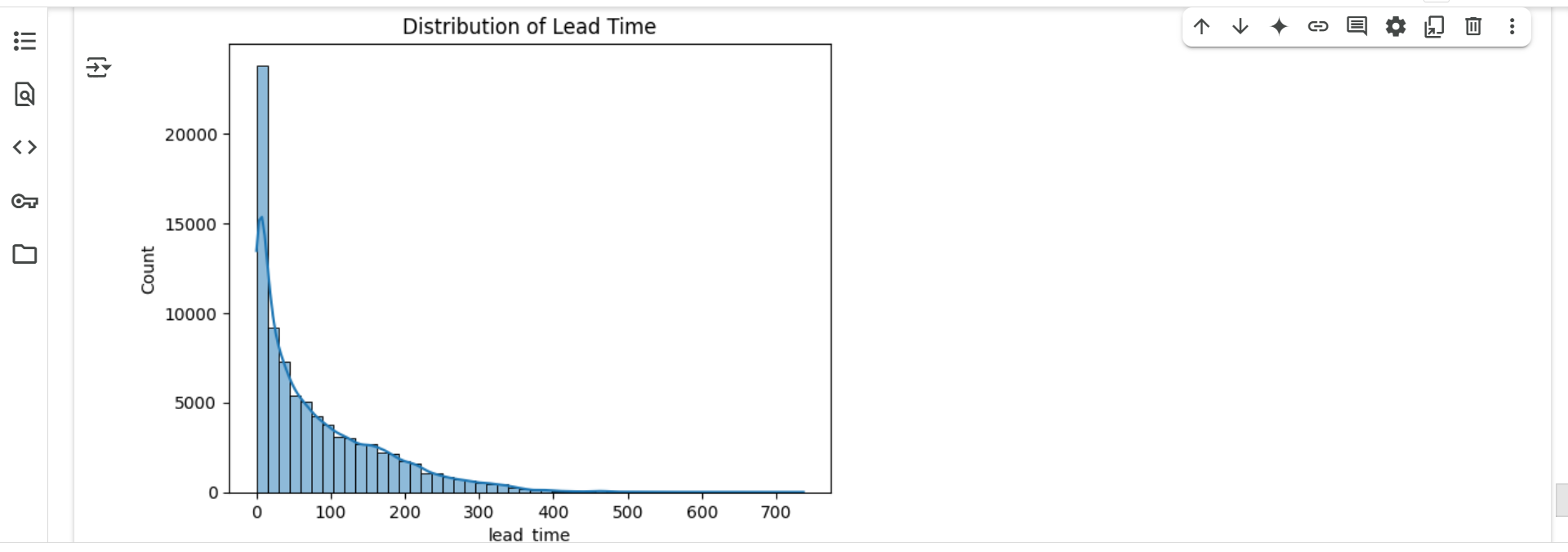
→To check the variation in lead time.

→Lead time indicates number of days between booking date and check-in date.

sns.histplot(data=df, x='lead\_time', bins=50,kde=True)

plt.title("Distribution of Lead Time")

plt.show()



→from the graph we can conclude that more number of people do last-minute bookings because peak is more sharper at left side and right skewed .

→A sharp peak near 0–10 indicates last-minute bookings.

* To increase the number of advanced bookings, a discount can be given to those who book at least 10 days before check-in.

3)Bar-Chart

→Between customer type and its counts.

df['customer\_type'].value\_counts()



→ from the above graph we can say that:

1)Most bookings are from Transient(Individual travelers or small groups who book short stays) customers (69,361), meaning individual travelers.

2)Transient-Party(A family or friends traveling together and making a joint booking) is the next biggest group (11,455), probably small groups or families traveling together.

3)Contract and Group customers(company booking rooms for employees regularly for work trips) are much fewer.

* For Contract and Group customers we can give offer like if particular company booking rooms for again time then can give 20-25% discount or rooms more than 4 then can any parking or any service free or with special discount.

4)Bar-chart:

→The bar graph displays the number of bookings for each **market segment** in your dataset.

plt.figure(figsize = (10, 3))

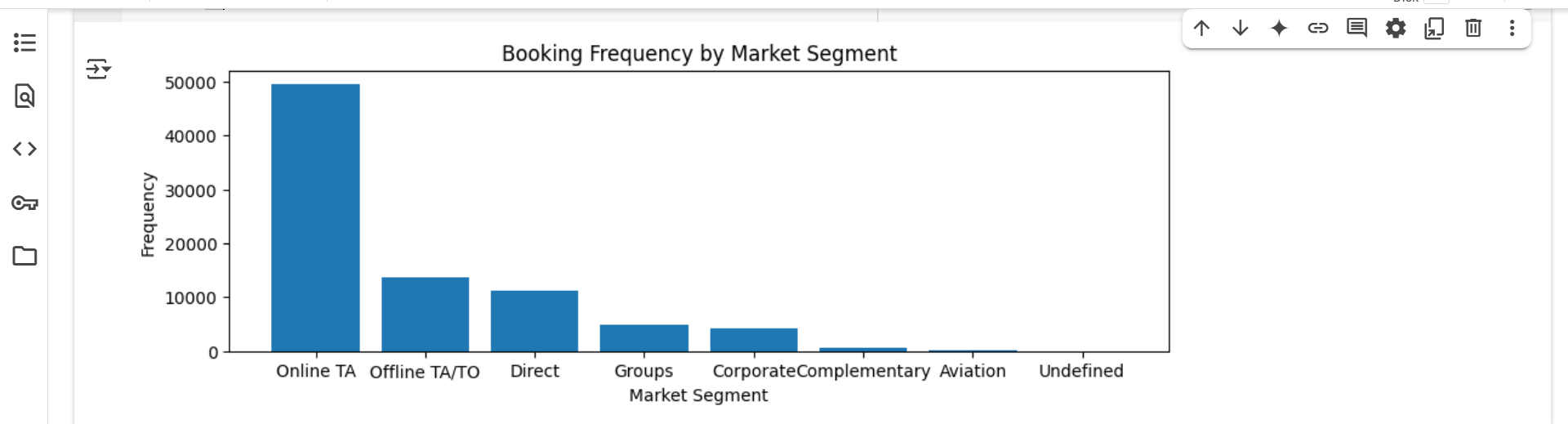
plt.xlabel('Market Segment')

plt.ylabel('Frequency')

plt.title('Booking Frequency by Market Segment')

plt.bar(df['market\_segment'].value\_counts().index,df['market\_segment'].value\_counts())

plt.show()



→from the above graph we can say that:

* Most bookings are made through third-party travel websites .
* A significant number of guests book directly through the hotel’s website or by phone.Indicates brand trust and opportunity to grow loyalty programs or offer direct booking discounts.
* Try to **encourage guests to book directly** through the hotel’s website or phone.
* This saves commission paid to OTAs and **increases profit**.

5)Bar chart:

→By plotting the bar chart between months and it’s frequency we can say that:

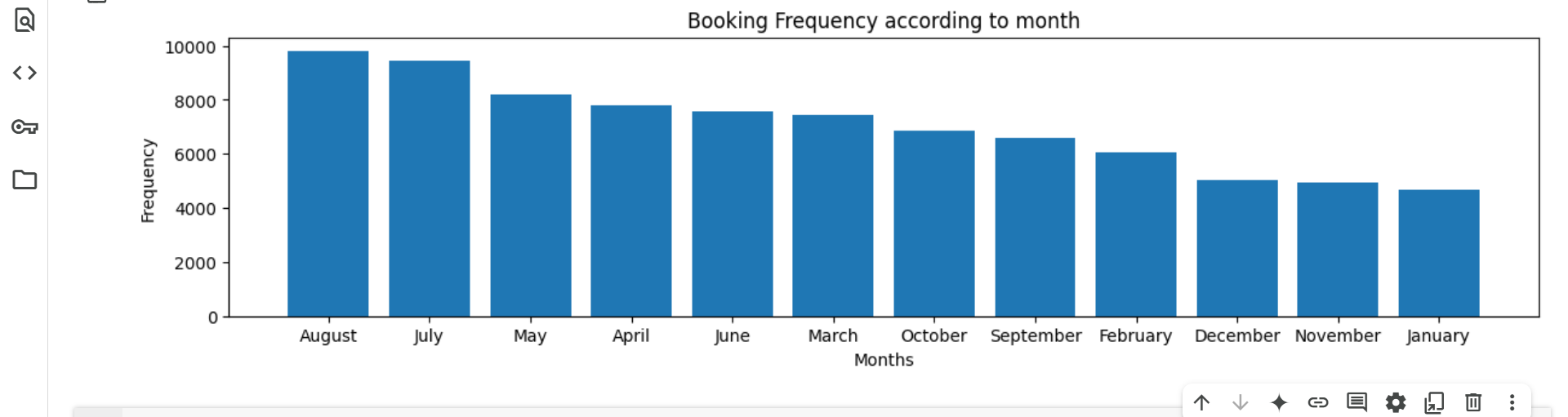
plt.figure(figsize = (14, 3))

plt.xlabel('Months')

plt.ylabel('Frequency')

plt.title('Booking Frequency according to moth')

plt.bar(df['arrival\_date\_month'].value\_counts().index,df['arrival\_date\_month'].value\_counts())



* August,july,may,april are most crowded months than Feb,dec,nov and January
* August,july,may,april are **high season**; prices can be higher, and more staff/resources may be needed.
* Feb,dec,nov and January are **off-season**; hotel may consider promotions or discounts to attract more guests.

6)Bar chart

→between type of meal and its frequency.

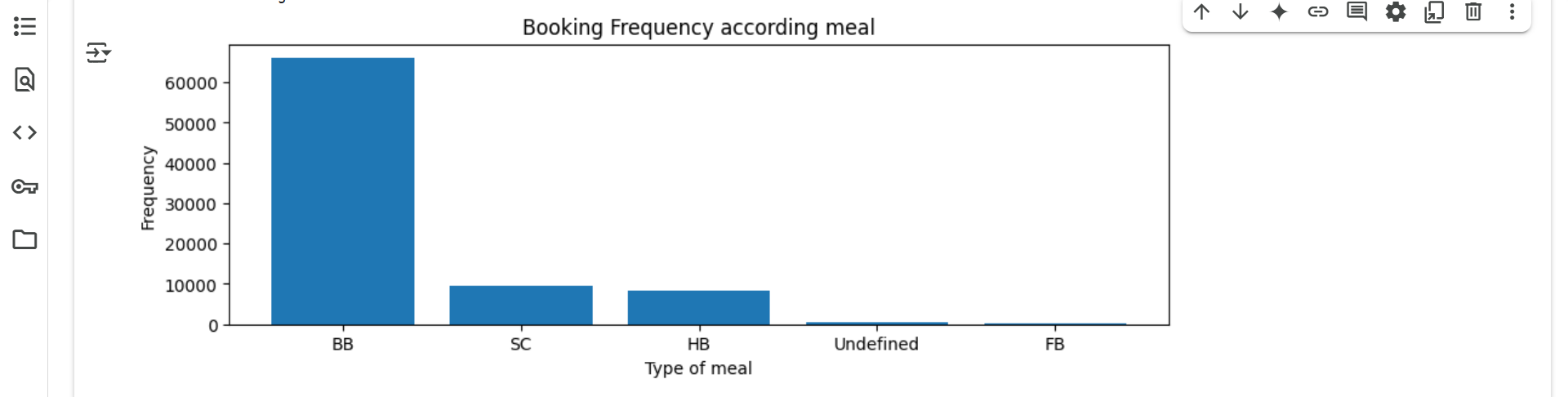
plt.figure(figsize = (10, 3))

plt.xlabel('Type of meal')

plt.ylabel('Frequency')

plt.title('Booking Frequency according meal')

plt.bar(df.meal.value\_counts().index,df.meal.value\_counts())



→Above graph shows:

* Highest frequency for BB(bread and breakfast) and lowest for FB(Full Board).
* By giving offers to FB hotel can increase revenue . The hotel should maintain a strong breakfast offering.

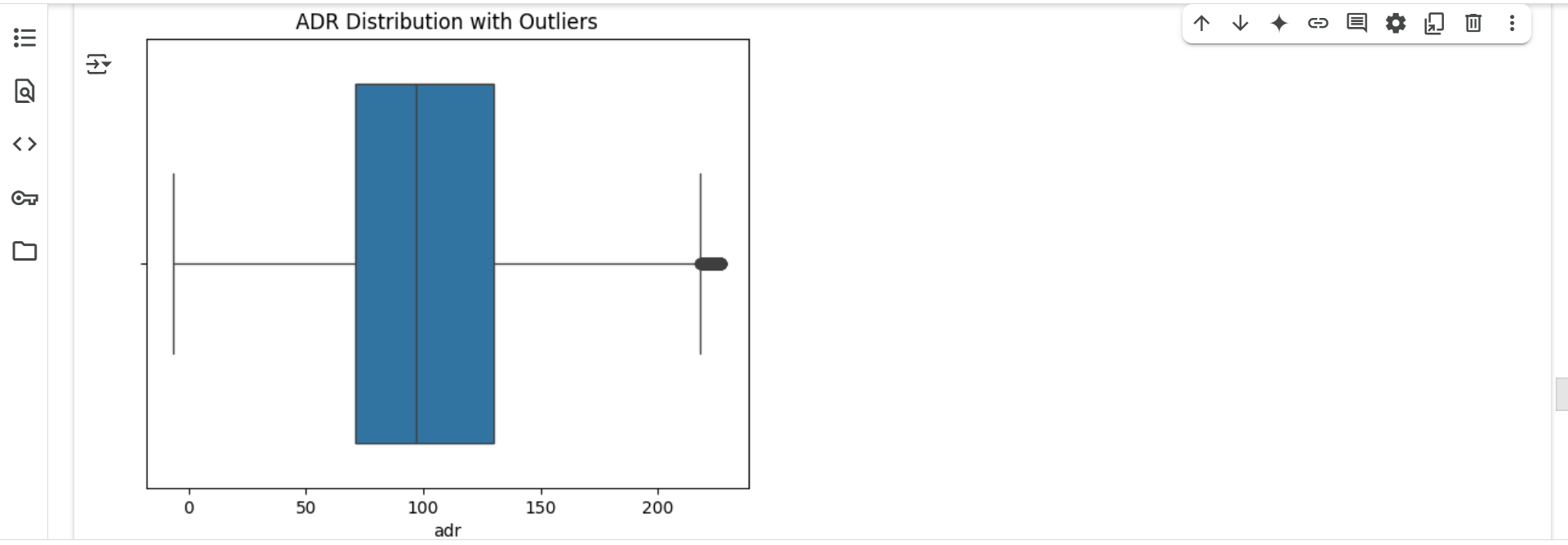
7)Boxplot:

→Plotted a boxplot of adr (Average Daily Rate), which shows how much people pay per night at the hotel.

sns.boxplot(x=df['adr'])

plt.title("ADR Distribution with Outliers")

plt.show()



→It helped me to see what is avg booking price and some following useful insights:

* Most people pay between ₹60 and ₹130 per night.
* The average booking price is around ₹100.
* Some people paid very high prices (above ₹200) — these outliers.

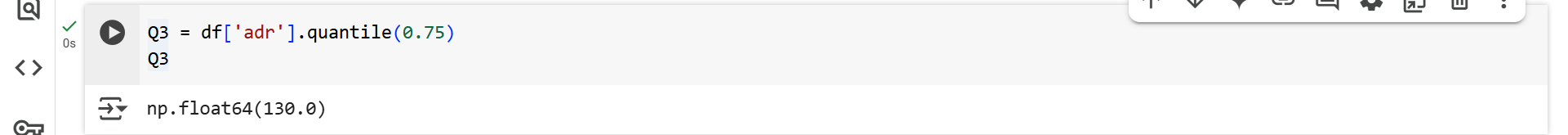
#Middle line inside the box:Median indicates avg paid is around ₹100

#length of box is IQR: It is Q3-Q1

Q3:75% is 130 means 75% people pay less than or equal to 130.

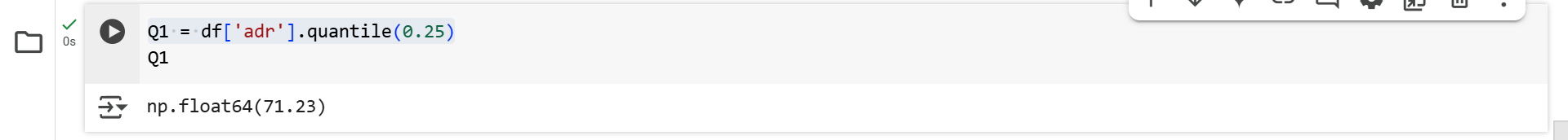
→

Q3 = df['adr'].quantile(0.75)

o/p : 

Q1:25% is 71.23 means 25% people pay less than or equal to 71.23.

Q1 = df['adr'].quantile(0.25)



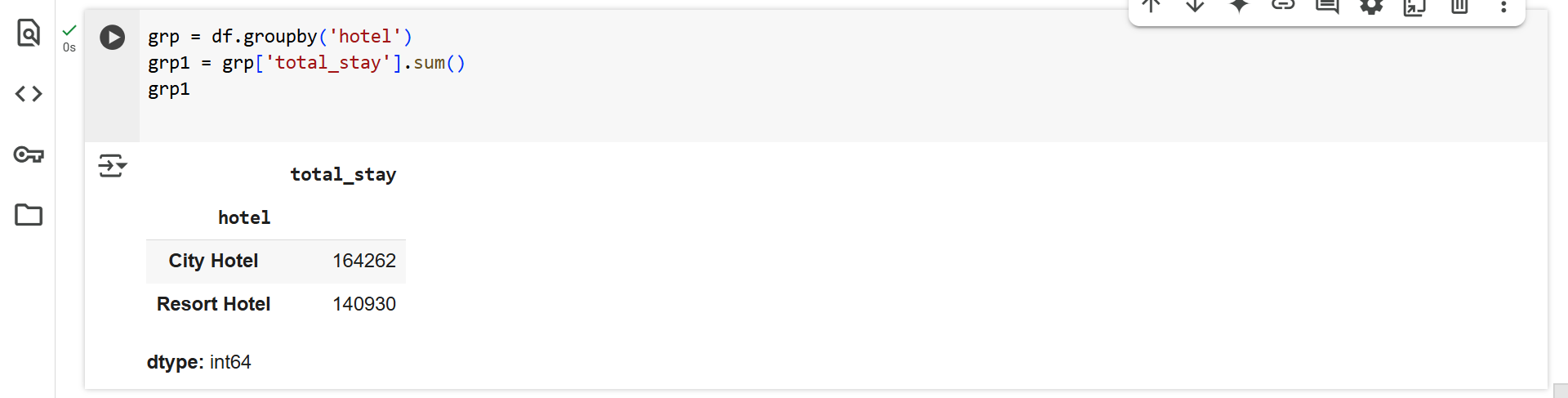
#LB:Whisker line extended towards left side.

#UB:Whisker line extended towards right side and on right side few dots are present that indicates very few people pay price more than ₹ 200.This dots are called as outliers.

**2)Bivariate Analysis**

1)Grouped bar chart:

→ between type of hotel and total\_stay.



plt.figure(figsize = (3, 3))

plt.xlabel('Type of Hotel')

plt.ylabel('Total Stay')

plt.title('Total Stay based on Hotel Type')

plt.bar(grp1.index,grp1)



→The City Hotel has a higher total stay count than the Resort Hotel.

→This suggests City Hotels are more popular or frequently booked.

2)Line plot

→ plotted line plot between monthwise adr between types of hotel.

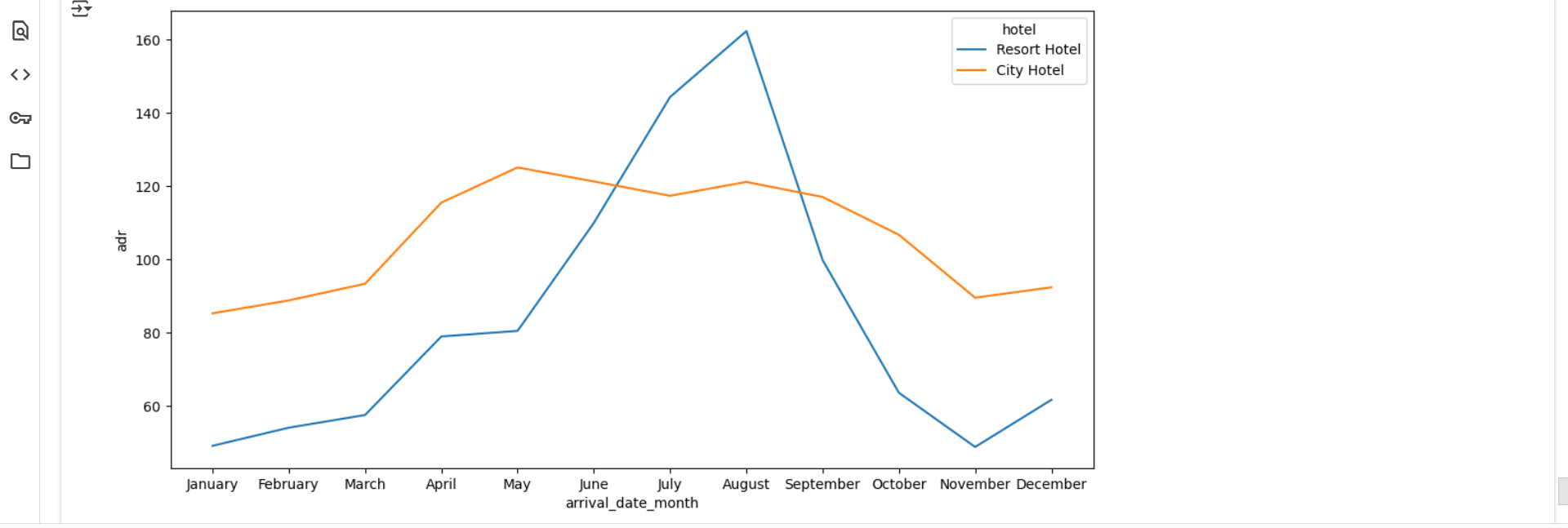
month\_order = ['January', 'February', 'March', 'April', 'May', 'June',

'July', 'August', 'September', 'October', 'November', 'December']

df['arrival\_date\_month'] = pd.Categorical(df['arrival\_date\_month'], categories=month\_order, ordered=True)

plt.figure(figsize=(12, 6))

sns.lineplot(x = df['arrival\_date\_month'], y = df.adr, hue = df.hotel, data = df, errorbar = None)



→ from the graph we can see resort hotel denoted by blue line has highest booking during aug and lowest during january and november

→ And City hotels denoted by orange line have highest booking april to may.

* City hotels are crowded during holidays like may so more staff will be required and other arrangements required.

3)Boxplot

→between two types of hotels and its adr.

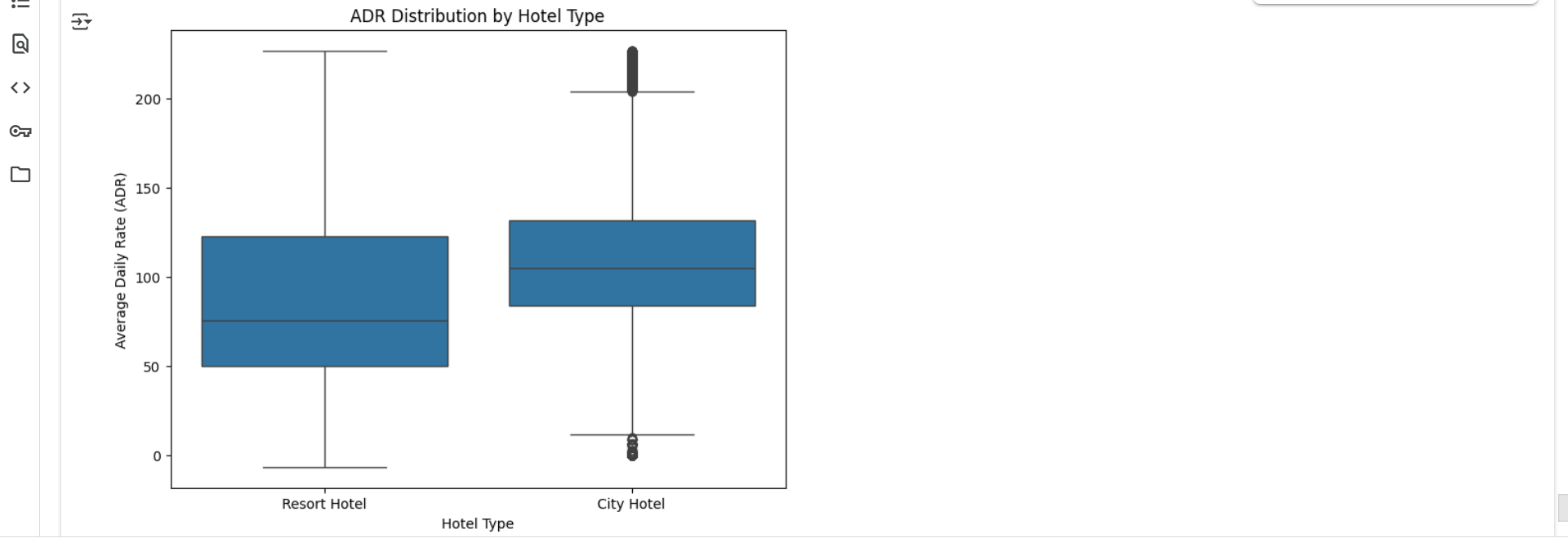
plt.figure(figsize=(8, 6))

sns.boxplot(x='hotel', y='adr', data=df)

plt.title('ADR Distribution by Hotel Type')

plt.xlabel('Hotel Type')

plt.ylabel('Average Daily Rate (ADR)')



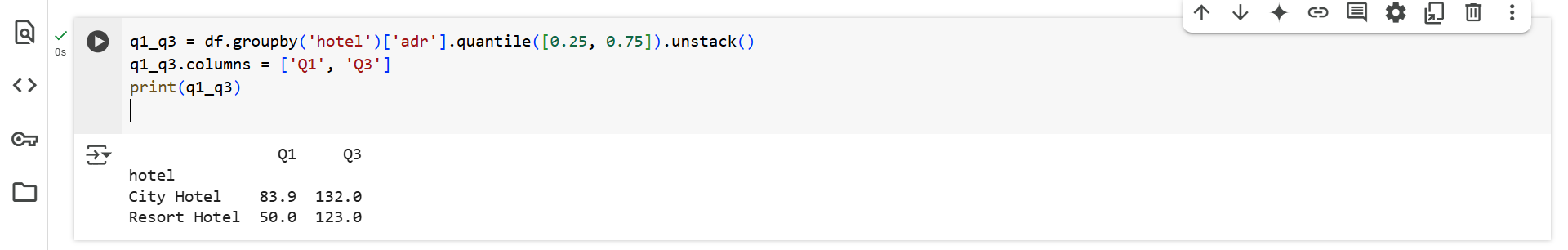
It helped me to see what is avg booking price(hotel typewise) and some following useful insights:

Resort hotel:

* Most people pay between ₹50 and ₹130 per night.
* The average booking price is around ₹80.

#Middle line inside the box:Median indicates avg paid is around ₹80

#length of box is IQR: It is Q3-Q1



Q3:75% is 123 means 75% people pay less than or equal to 123.

Q1:25% is 50 means 25% people pay less than or equal to 50.

City hotel:

* Most people pay between ₹50 and ₹130 per night.
* The average booking price is around ₹80.

#Middle line inside the box:Median indicates avg paid is around ₹80

#length of box is IQR: It is Q3-Q1

Q3:75% is 132 means 75% people pay less than or equal to 132.

Q1:25% is 83 means 25% people pay less than or equal to 83.

4)Boxplot

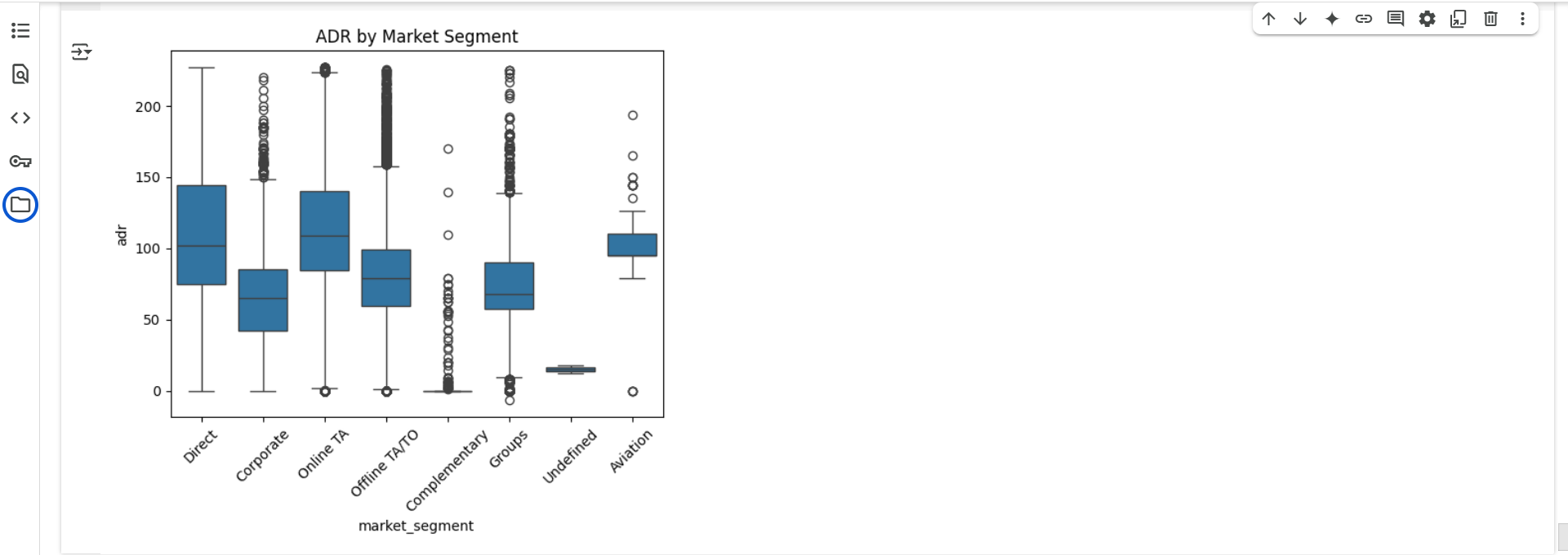
→between different market segments and adr.

sns.boxplot(data=df, x='market\_segment', y='adr')

plt.title("ADR by Market Segment")

plt.xticks(rotation=45)

plt.show()



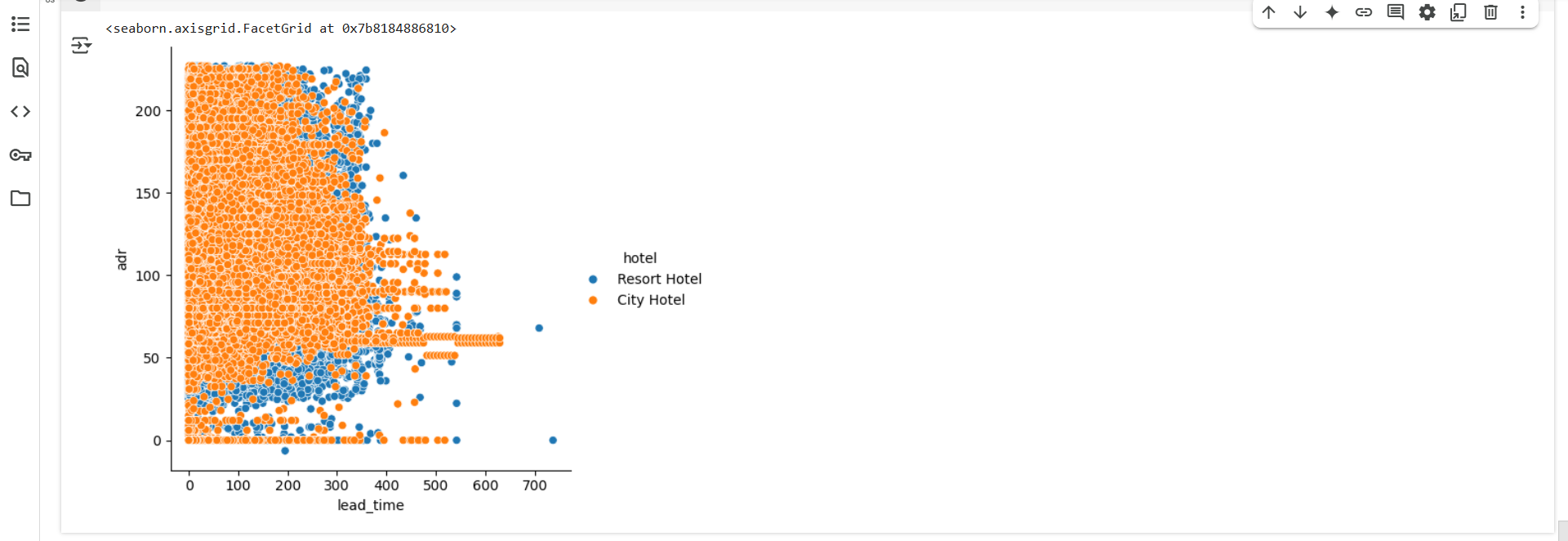
→from the above diagram i can conclude that:

* **Direct** and **Online TA (Travel Agent)** segments show **high median ADR** values, meaning they generally bring in higher-paying customers.
* **Complementary** and **Undefined** segments have the **lowest ADR**, which makes sense — complementary bookings are often free.
* **Groups**, **Online TA**, and **Offline TA/TO** have a **wide spread and many outliers**, suggesting inconsistent pricing. Possibly affected by bulk bookings or seasonal demand.
* **Undefined** shows **low spread and no outliers**, indicating consistent pricing
* **Corporate** segment also shows relatively **tight IQR**, indicating pricing control.

5)Scatter plot:

→ to check hotel typewise lead\_time and its effect on adr.

sns.relplot(x = df['lead\_time'], y = df['adr'], hue = df['hotel'])



→Firstly max booking is done in city hotels with last moment booking i.e small lead time.

→In city hotel,ADR spreed Covers a wide ADR range (from 0 to above 200), mostly **evenly distributed** resort Shows clusters at **lower ADR values.**

6)Grouped Bar chart

→I have taken top 10 countries with adr in both types of hotel.

# Step 1: get top 10 countries

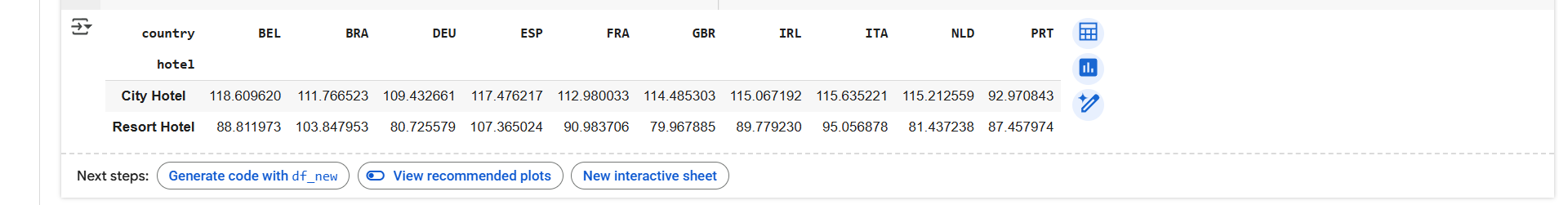
top\_10\_countries = df['country'].value\_counts().head(10).index

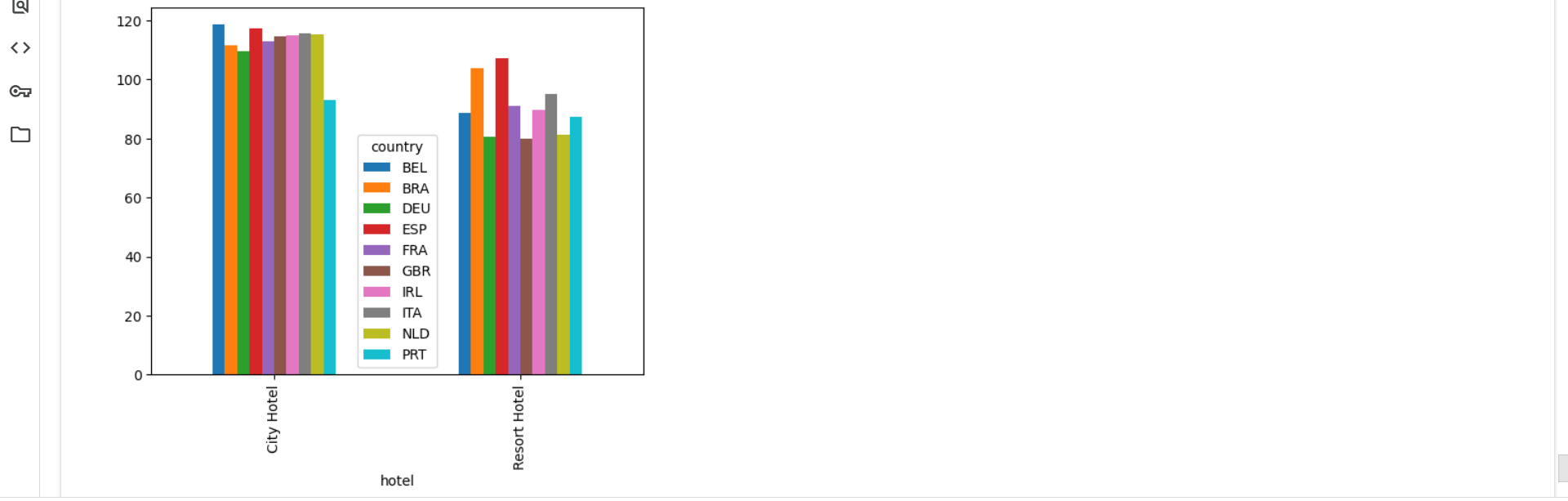
# Step 2: filter df for only these top countries

df\_top10 = df[df['country'].isin(top\_10\_countries)]

df\_new = df\_top10.pivot\_table(values='adr', index='hotel', columns='country', aggfunc='mean')

df\_new





→from the above graph:

* Booking from all top 10 countries is mostly done in city hotels.
* So revenue most generated in city hotel.
* ADR of BEL is about 110 in city and about 80 in resort.Here is huge diff.between ADR generated by countries in two diff.hotels.

**3)Multivariate Analysis**

1)Boxplot

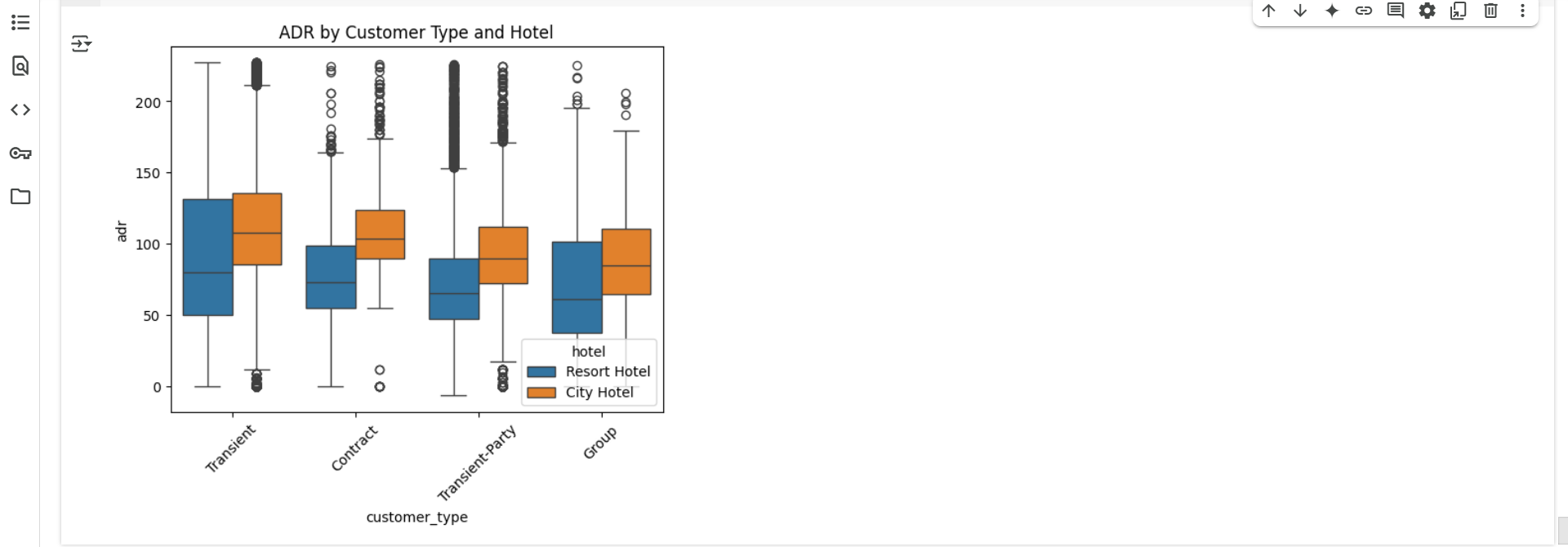
→Checking fluctuation in ADR by Customer type and type of hotel.

sns.boxplot(data=df, x='customer\_type', y='adr', hue='hotel')

plt.title("ADR by Customer Type and Hotel")

plt.xticks(rotation=45)

plt.show()



→insights:

* Here i can conclude that all type of customers pay more in city hotels than resort.
* There are more customers who pay large price in resort from transient party.

2)Boxplot

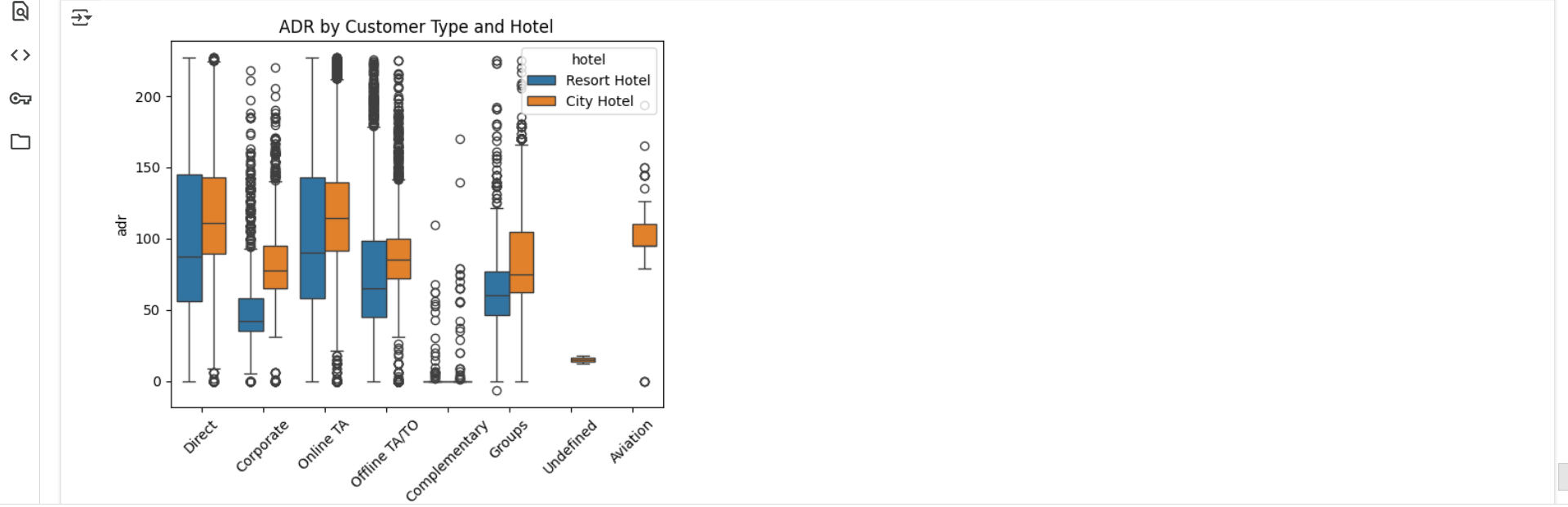
→Drawn boxplot to check what is effect of market segment used in both hotel types on ADR.

sns.boxplot(data=df, x='market\_segment', y='adr', hue='hotel')

plt.title("ADR by Customer Type and Hotel")

plt.xticks(rotation=45)

plt.show()



→Online TA and Direct has almost same and max avg ADR approx.120 in city hotels.

→More [numb.of](http://numb.of) customer prefer offline TA/TO in city hotels that generates large price ADR.

***3. Correlation Analysis.***

***4. Hypothesis Testing.***

1)H0: There is no difference in ADR between bookings made through Online TA and Direct channels.

For this we can use 2 sample t test:-

→ we have 2 independent samples Online TA and Direct channels having ADR that means continuous data.

→ by calculating means of both the sample if there diff. Is zero then we do not REJECT the HO else we REJECT the HO.

Step 1:

Ho: Samp2\_Direct.mean - Samp1\_OnlineTA.mean=0

Ha:Samp2\_Direct.mean - Samp1\_OnlineTA.mean!=0

Sample 1:Online TA

Samp1\_OnlineTA = df[(df['market\_segment'] == 'Online TA')]['adr']

Sample 2 : Direct

Samp2\_Direct = df[(df['market\_segment'] == 'Direct')]['adr']

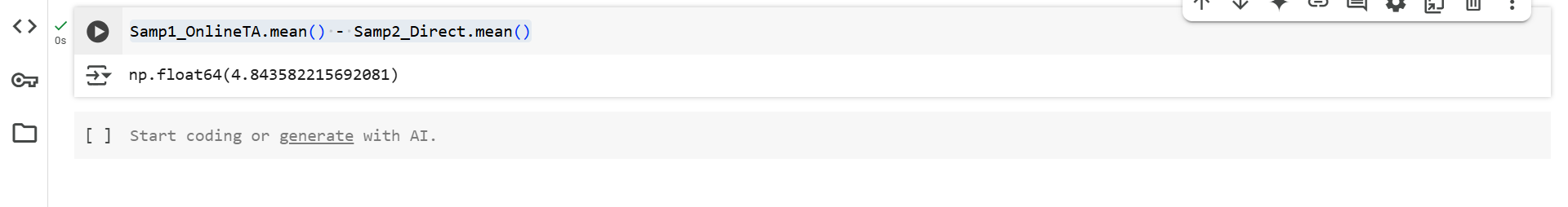
Step 2:Establishing LOS

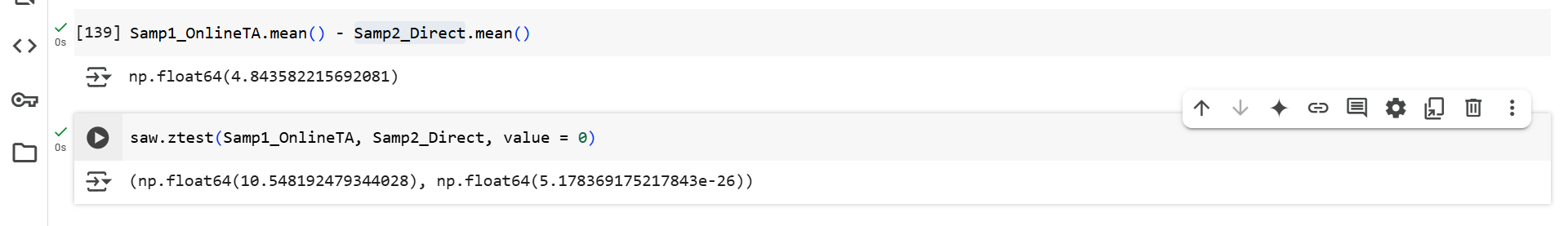
Los : 0.05

Step 3: Selecting test

→2 sample t test

Samp1\_OnlineTA.mean() - Samp2\_Direct.mean()





Here P\_value < 0.05

And Test Statistics is positive hence mean is greater than 0.

Hence we REJECT HO

**There is a difference in ADR between bookings made through Online TA and Direct channels.**

2)H0: Room upgrades are independent of lead time.

→For this we can use chi-square test for degree of association.

→Here we have to find is Room upgrades independent on lead time or not.

→Here lead time is continuous we have to convert it into categorical.

Step 1:

Ho :Room upgrades are independent of lead time.

Ha:Room upgrades are dependents on lead time.

Step 2:

LOS:0.05

Step 3:Selecting test.

chi-square test for degree of association

→ Converted lead time continuous to categorical.

df['lead\_time\_category'] = pd.cut(df['lead\_time'],

bins=[-1, 30, 90, df['lead\_time'].max()],

labels=['Short', 'Medium', 'Long'])

→Check if reserved room type != assigned

df['room\_upgrade'] = (df['reserved\_room\_type'] != df['assigned\_room\_type']).astype(int)

0 = Same

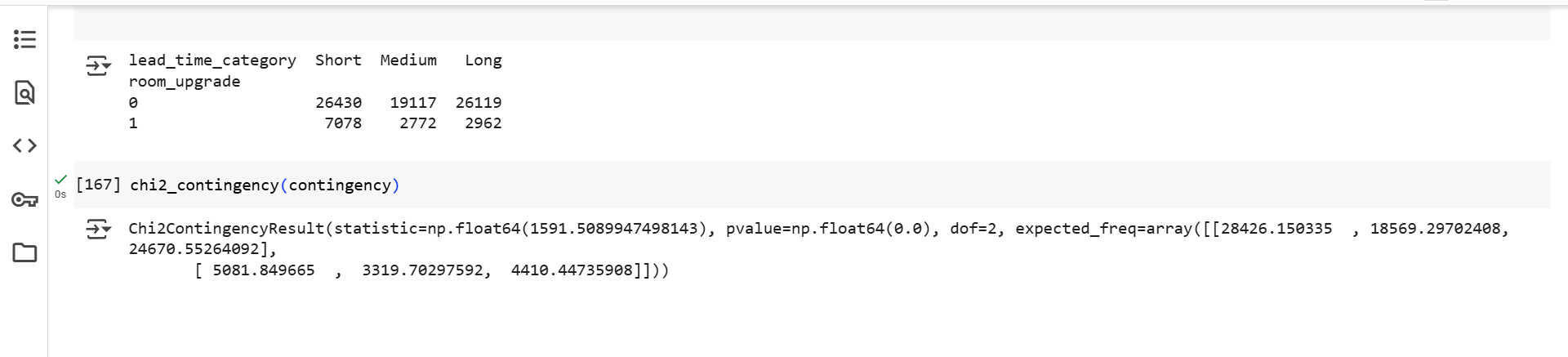
1 = different

→ created table having count of 0 and 1.

contingency = pd.crosstab(df['room\_upgrade'], df['lead\_time\_category'])

print(contingency)

→perform chi-square test.



From the test p\_value < 0.05

Hence REJECT HO.

Room upgrades are dependents on lead time.

Because last moment booking most of the times gets upgraded room.

3)H0: Average stay duration does not differ between customer types.

→ here we have 4 customer types .hence we can use one-way anova.

df1 = df[df['customer\_type'] == 'Transient']['total\_stay']

df2 = df[df['customer\_type'] == 'Contract']['total\_stay']

df3 = df[df['customer\_type'] == 'Transient-Party']['total\_stay']

df4 = df[df['customer\_type'] == 'Group']['total\_stay']

Step 1:

Ho :Average stay duration does not differ between customer types.

Ha:Average stay duration differs between customer types.

Step 2:

LOS : 0.05

Step 3:

One way Anova.

→ created one dataframe.

df1\_labeled = pd.DataFrame({'total\_stay': df[df['customer\_type'] == 'Transient']['total\_stay'],

'customer\_type': 'Transient'})

df2\_labeled = pd.DataFrame({'total\_stay': df[df['customer\_type'] == 'Contract']['total\_stay'],

'customer\_type': 'Contract'})

df3\_labeled = pd.DataFrame({'total\_stay': df[df['customer\_type'] == 'Transient-Party']['total\_stay'],

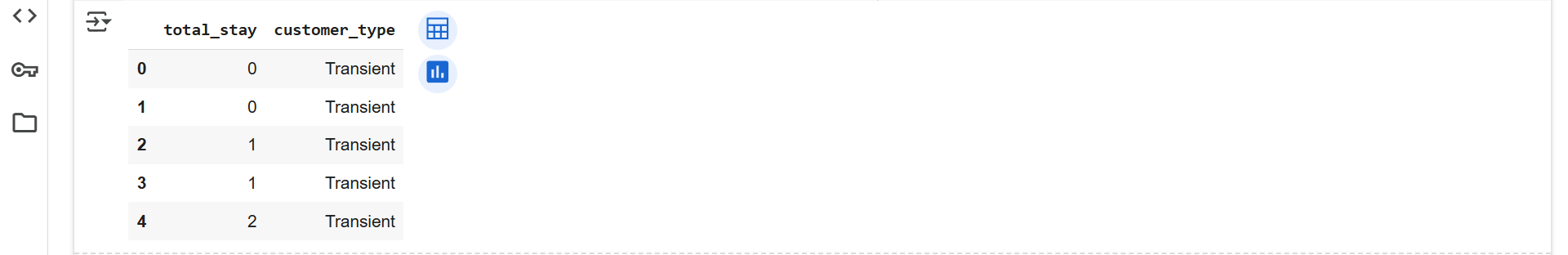
'customer\_type': 'Transient-Party'})

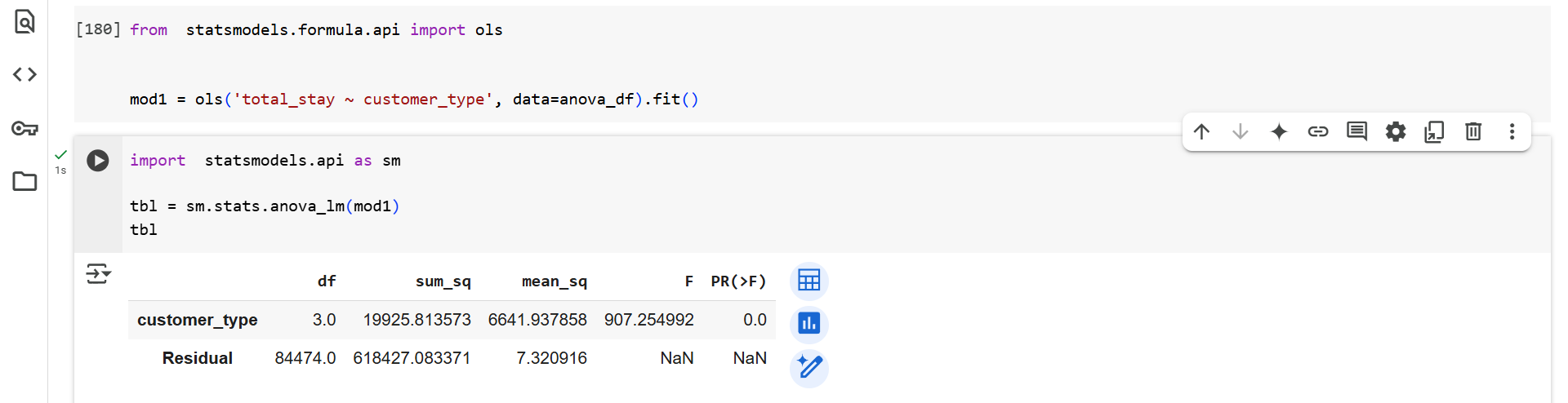
df4\_labeled = pd.DataFrame({'total\_stay': df[df['customer\_type'] == 'Group']['total\_stay'],

'customer\_type': 'Group'})

anova\_df = pd.concat([df1\_labeled, df2\_labeled, df3\_labeled, df4\_labeled], ignore\_index=True)

anova\_df.head()





Here p\_values < 0.05 hence we REJECT HO.

Hence Average stay duration differs between customer types.

***5. Key Business Questions***

1)What influences ADR the most?

→By using correlation matrix we can find out what influences ADR the most.

numerical\_features = df.select\_dtypes(include=['int64', 'float64'])

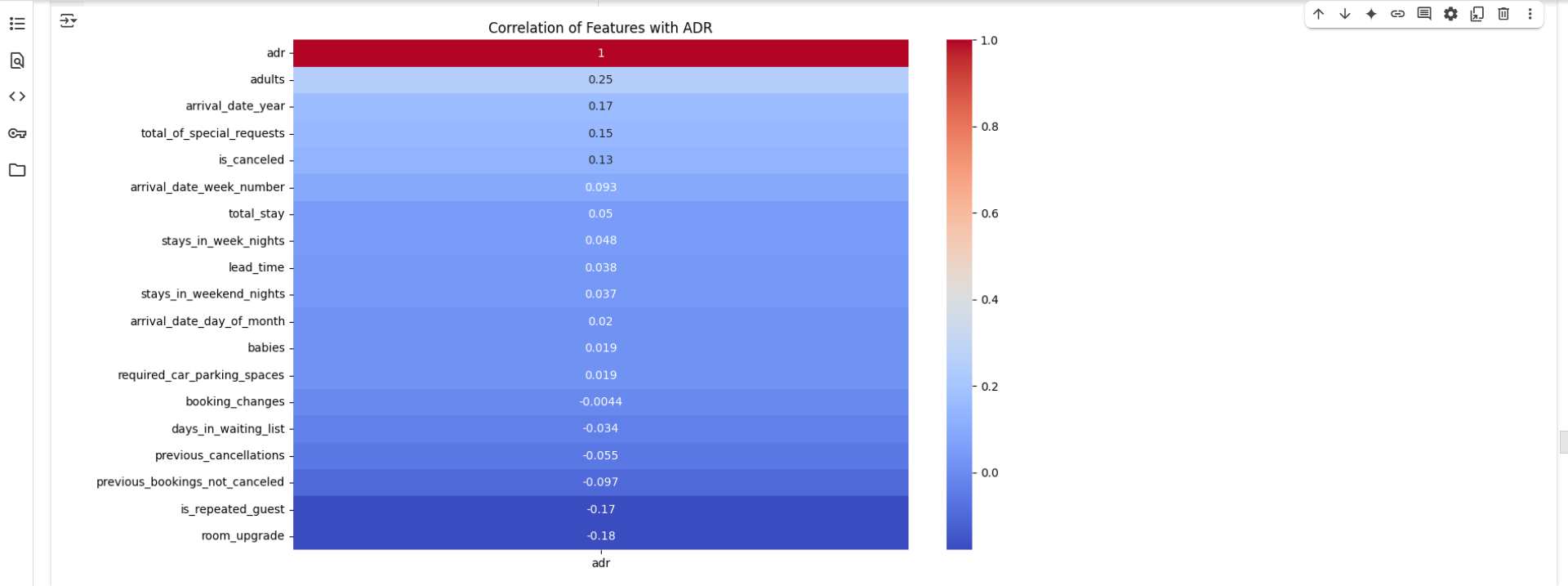
corr\_matrix = numerical\_features.corr()

plt.figure(figsize=(12, 8))

sns.heatmap(corr\_matrix[['adr']].sort\_values(by='adr', ascending=False), annot=True, cmap='coolwarm')

plt.title("Correlation of Features with ADR")

plt.show()



→From the above graph adr,adults,arrival\_date\_year ,total\_of\_special\_requests influences the ADR most.

2)Do guests who book earlier tend to request more changes?

→By plotting scatter plot we can find out Do guests who book earlier tend to request more changes or not.

plt.figure(figsize=(8, 5))

sns.scatterplot(data=df, x='lead\_time', y='booking\_changes', alpha=0.3)

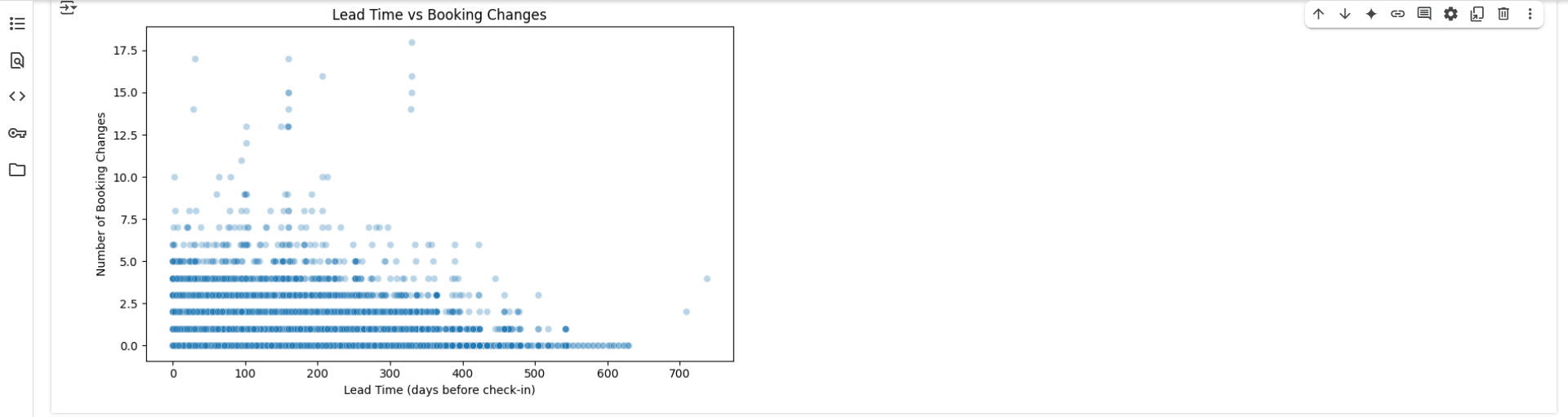
plt.title("Lead Time vs Booking Changes")

plt.xlabel("Lead Time (days before check-in)")

plt.ylabel("Number of Booking Changes")

plt.tight\_layout()

plt.show()



→More changes are done at the right side towards larger values of lead time.Hence it is proved that guests who book earlier tend to request more changes.

3)Are there pricing or booking differences across countries?

a)By country

df['total\_stay'] = df['stays\_in\_week\_nights'] + df['stays\_in\_weekend\_nights']

top\_countries = df['country'].value\_counts().head(10).index

df\_top = df[df['country'].isin(top\_countries)]

plt.figure(figsize=(10, 6))

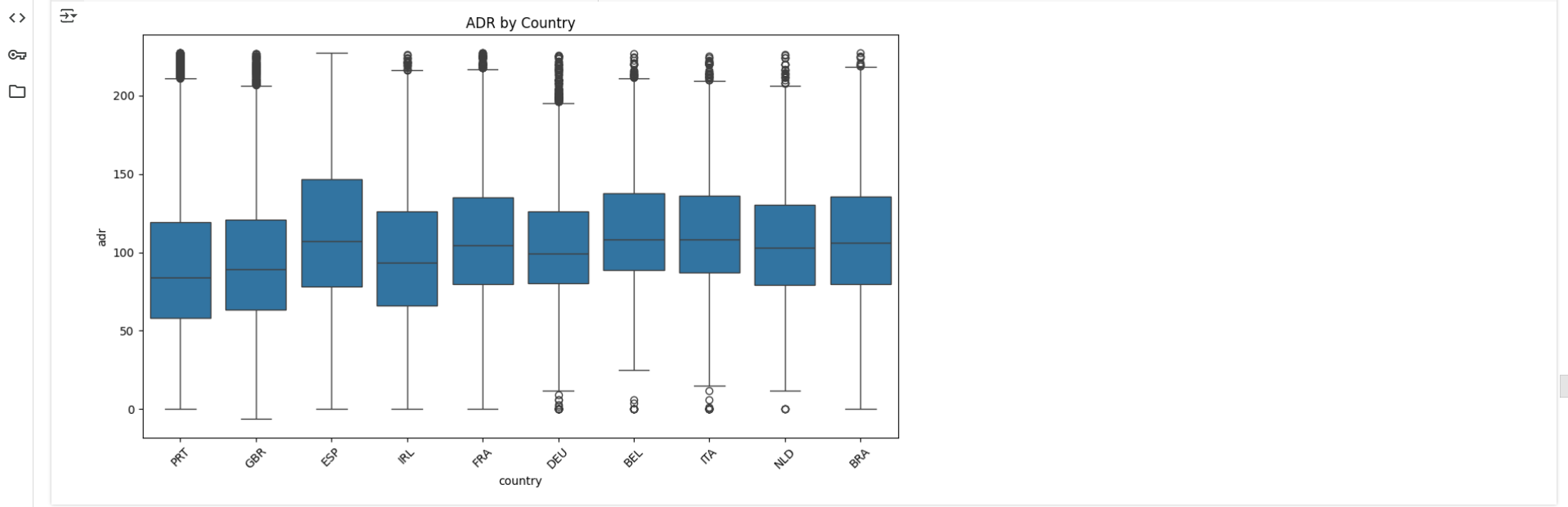
sns.boxplot(data=df\_top, x='country', y='adr')

plt.xticks(rotation=45)

plt.title("ADR by Country")

plt.tight\_layout()

plt.show()



→ESP country pays more price with comparison of others .

4) Is there a pattern in room upgrades or reassignment?

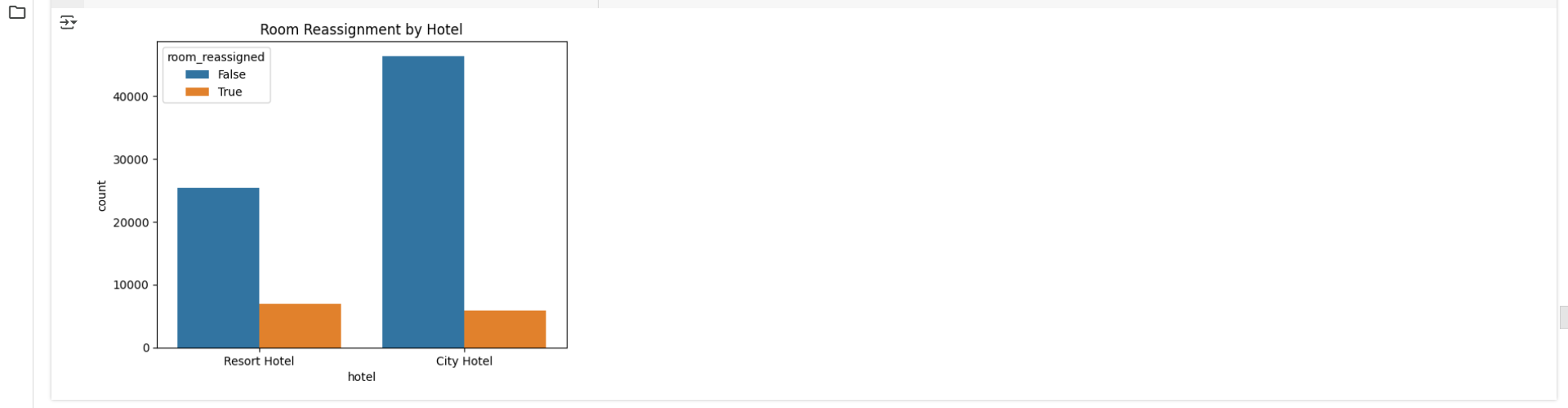
df['room\_reassigned'] = df['reserved\_room\_type'] != df['assigned\_room\_type']

df['room\_reassigned'].value\_counts(normalize=True) \* 100

sns.countplot(data=df, x='hotel', hue='room\_reassigned')

plt.title("Room Reassignment by Hotel")

plt.show()



5)Are reserved room types consistently matched with assigned room types?

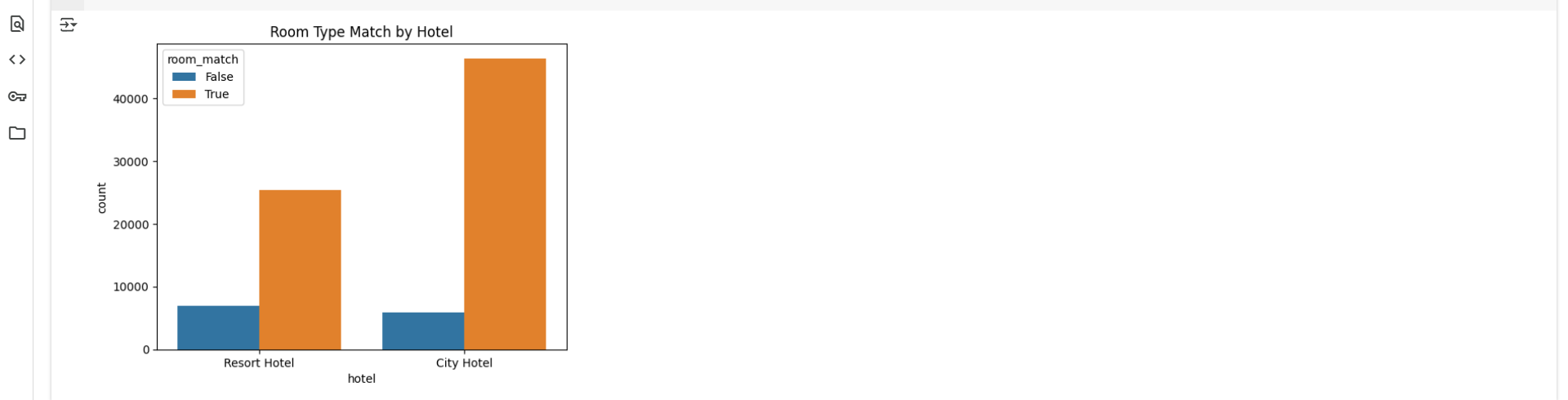
df['room\_match'] = df['reserved\_room\_type'] == df['assigned\_room\_type']

match\_rate = df['room\_match'].value\_counts(normalize=True) \* 100

sns.countplot(data=df, x='hotel', hue='room\_match')

plt.title("Room Type Match by Hotel")

plt.show()



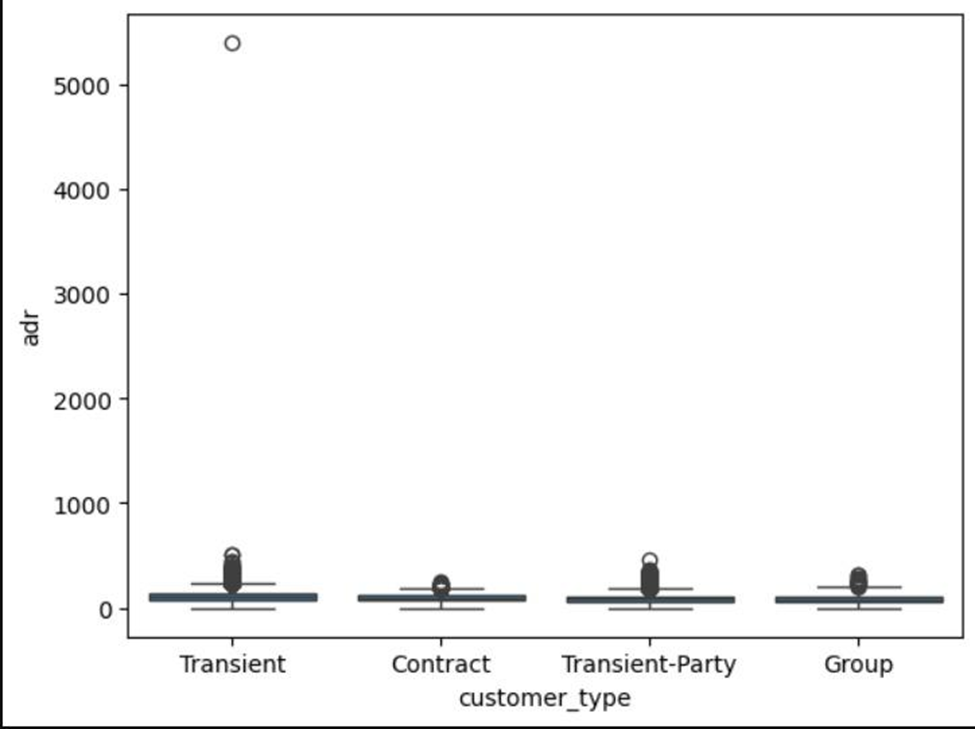
6)What are the most common guest demographics (e.g., group size, nationality)?

sns.boxplot(data=df, x='customer\_type', y='lead\_time')

plt.title("Lead Time by Customer Type")

plt.xticks(rotation=45)

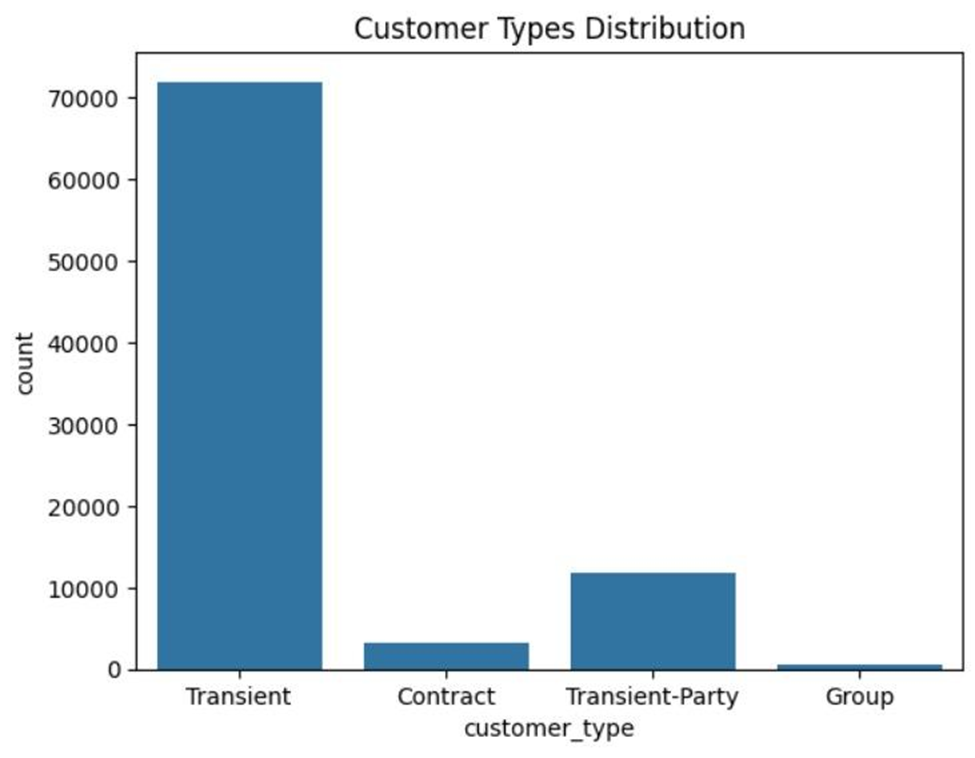
plt.show()

****

Almost all the groups were of the size of 0 to 5 people.The highest number of guests were from PRT followed by GBR and France and so on.

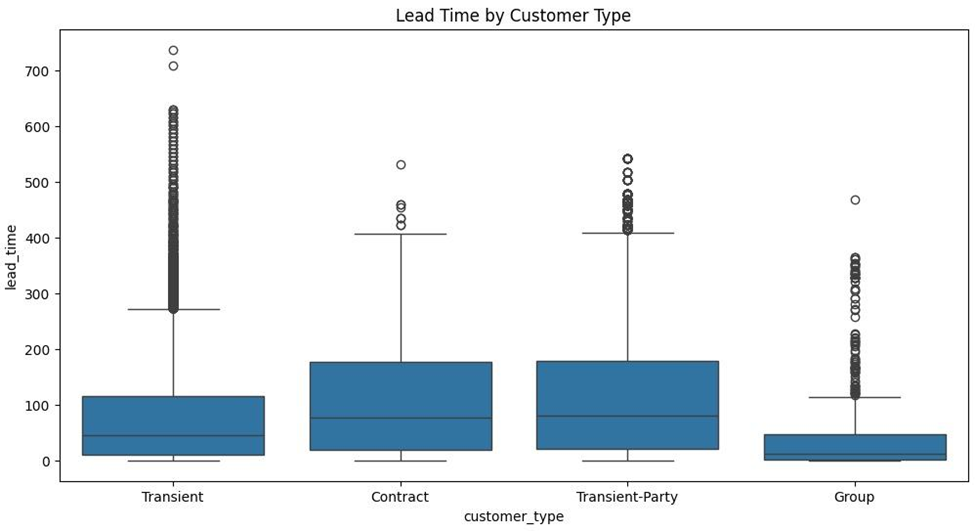
7)Are there patterns in guest types (e.g., transient vs. corporate) that influence booking behavior?

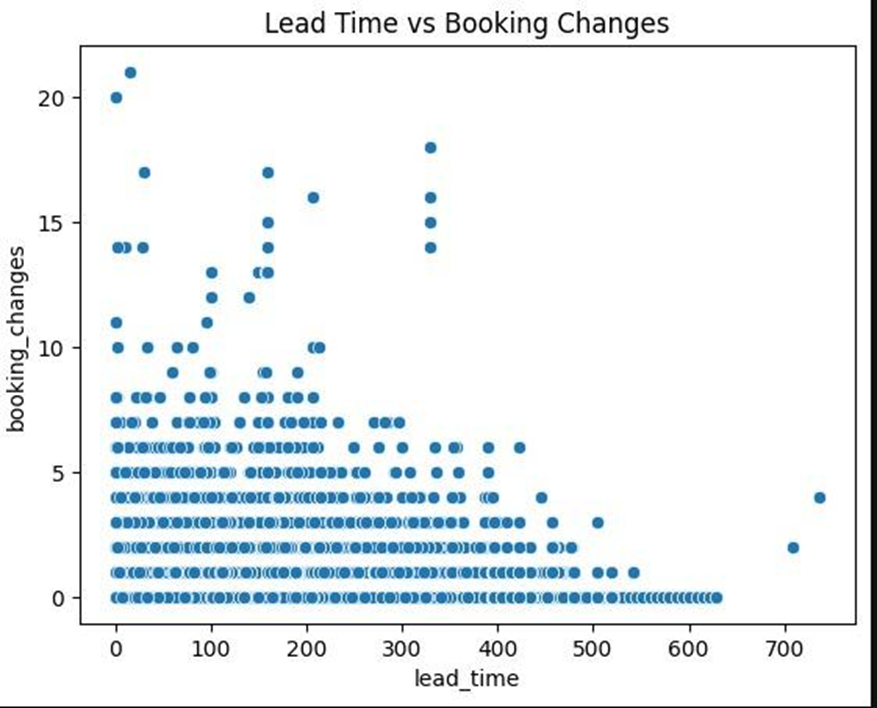
All customer types show outliers, but Transient guests have the widest ADR spread, suggesting variable pricing or booking flexibility.



8)How does booking lead time vary across customer types and countries?

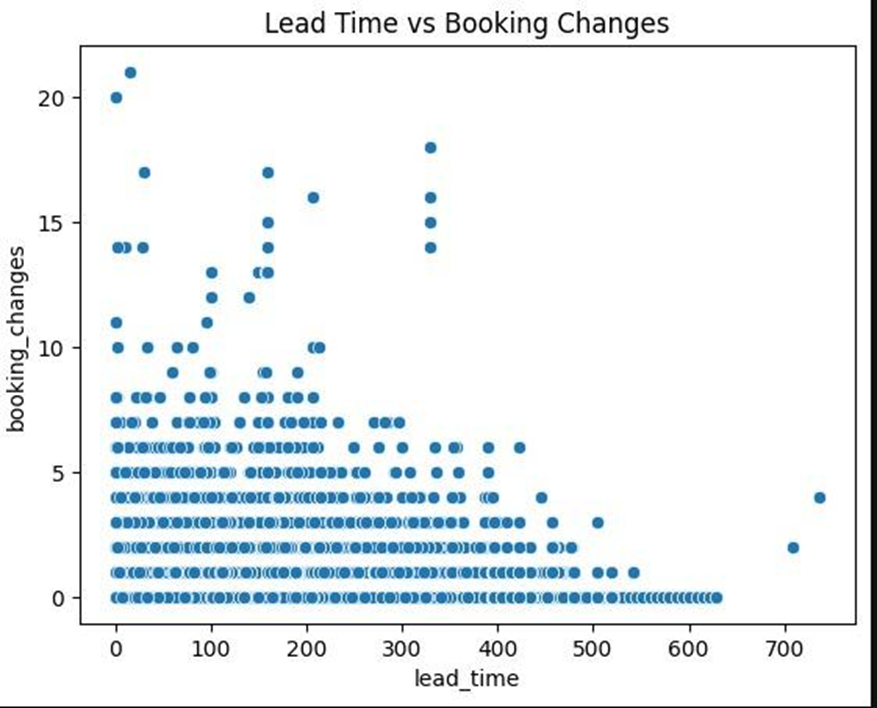
Longer booking lead times slightly lead to higher average daily rates.





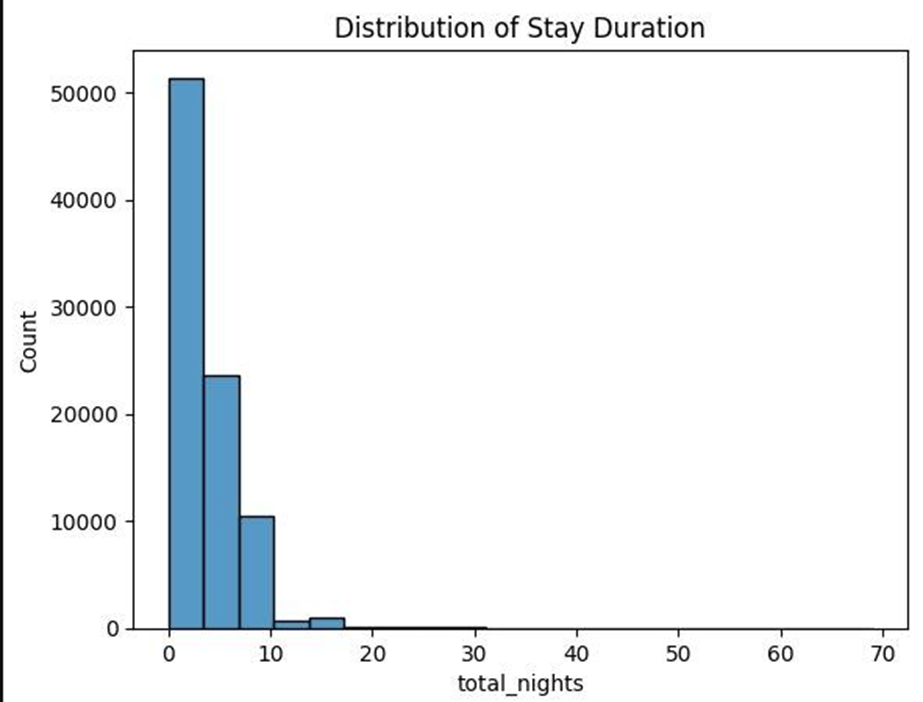
9) Are longer lead times associated with fewer booking changes or cancellations?

longer lead times results in fewer booking changes or cancellations



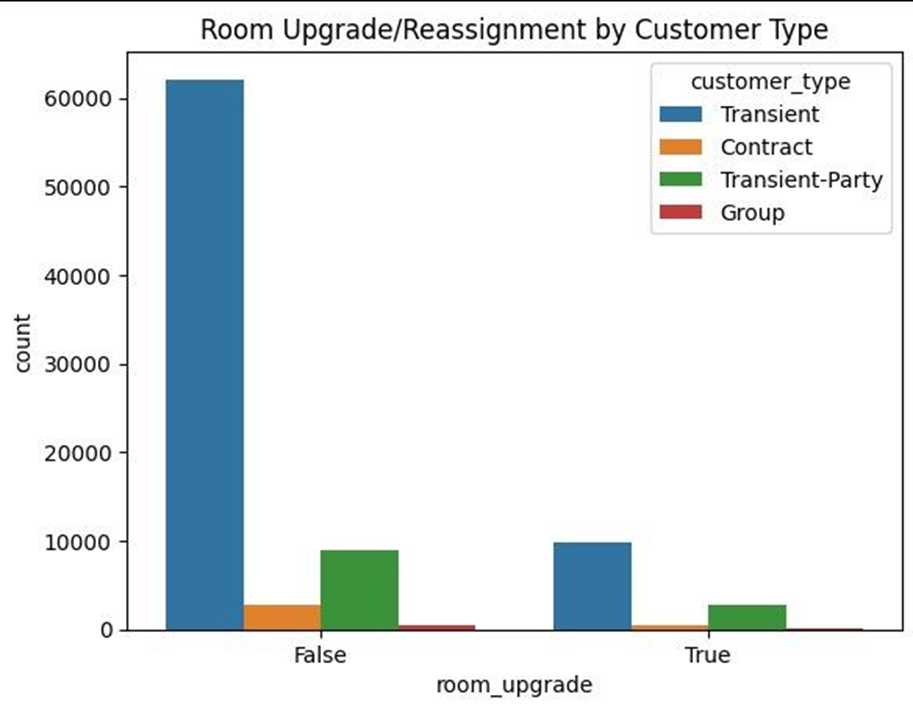
10)What is the typical duration of stay, and how does it vary by customer type or segment?

Ans : transient type of customer tend to stay at a longer duration as compared to group while the patter with contract and transient-party customers have a consistent stay duration.



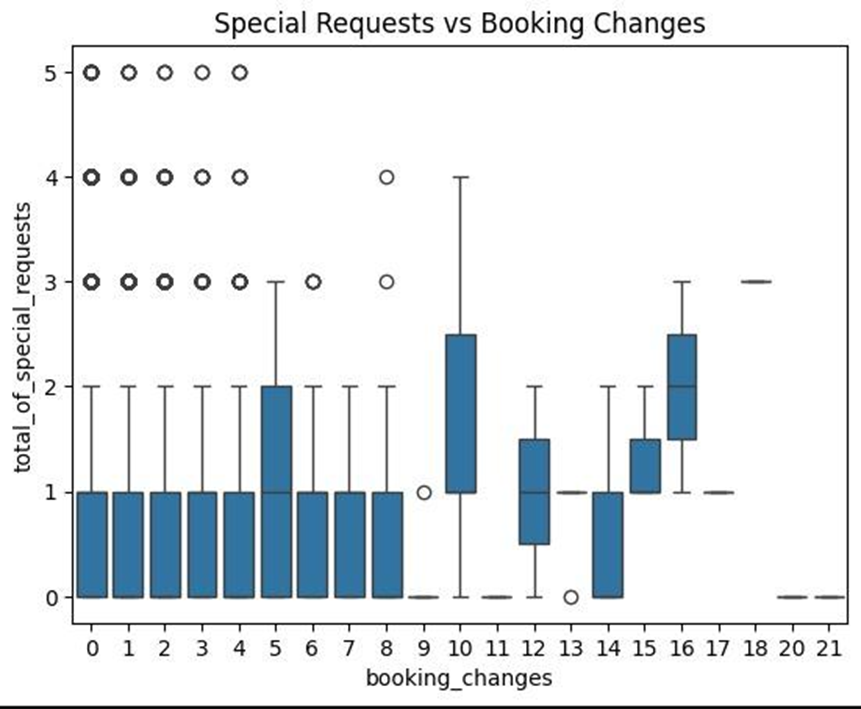
11)How often are guests upgraded or reassigned to a different room type?

Ans: Its very rare that customers are upgraded



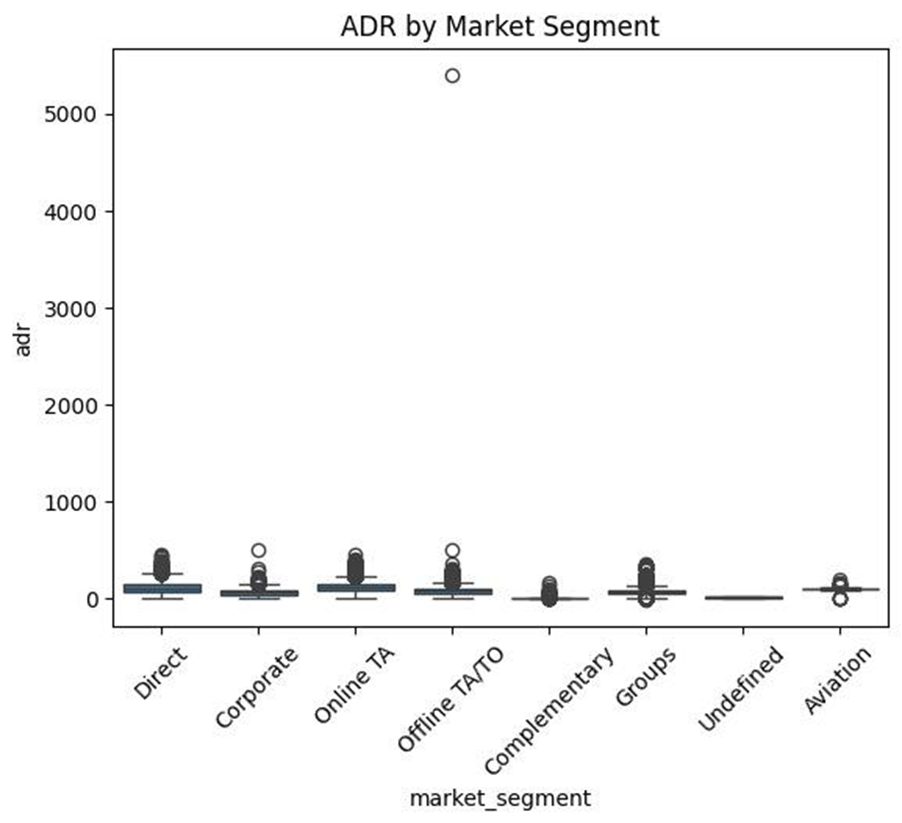
12)Are guests who make special requests more likely to experience booking changes or longer stays?

Ans: guests who make special requests more likely to experience booking changes. guests who make special requests more likely to stay for avg no of days



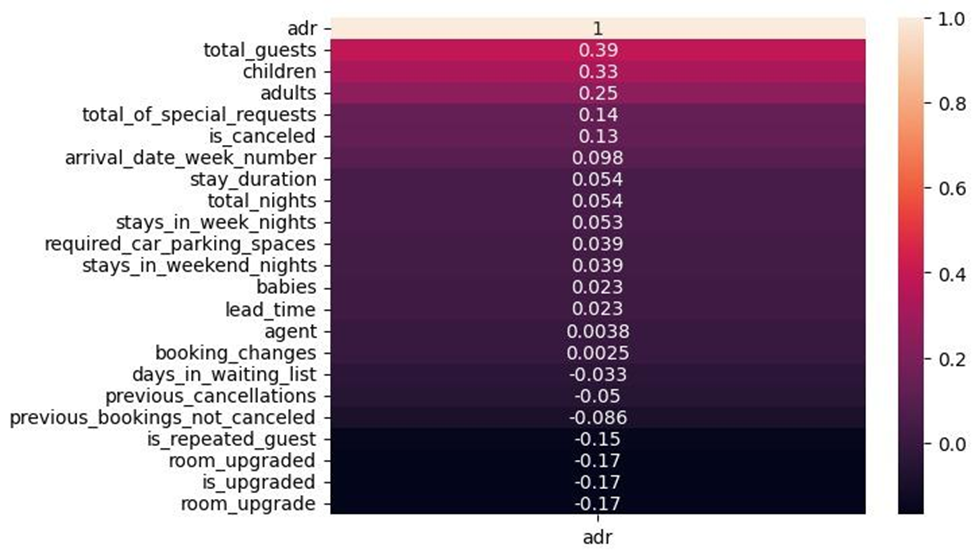
13)Do certain market segments or distribution channels show higher booking consistency or revenue?

Ans : higher booking consistency or revenue are showed by Online TA and Direct



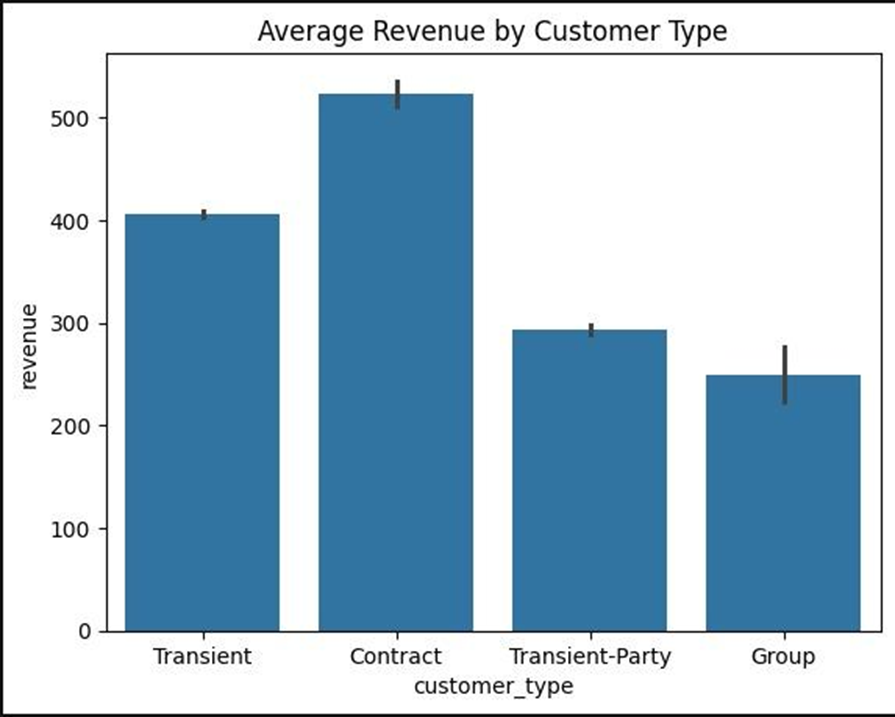
14)What factors are most strongly associated with higher ADR?

Ans : factors like revenue, total no of guest,group\_size are the reasons for strong ADR.



15)Are there customer types or segments consistently contributing to higher revenue?

Ans : Transient customers contribute to higher revenue



**Conclusion:**

The analysis of the hotel bookings dataset yields several important insights that can help guide strategic business decisions:

Data Quality and Integrity: After handling missing values and cleaning the dataset, we ensured the data is reliable for analysis. Columns like agent, children, and country were properly imputed, while irrelevant ones like company were removed to enhance clarity.

Customer Demographics: The total number of guests per booking (total) helps in understanding customer types—solo travelers, families, or groups. This can guide personalized marketing and service offerings.

Booking and Stay Patterns: The engineered stay column reveals the length of stays, crucial for identifying trends in short- vs long-term visits. This supports optimized pricing strategies and room availability planning.

Revenue Estimation: By calculating the revenue column, we get a proxy for the financial value of each booking. This is essential for forecasting income, identifying high-value customers, and optimizing advertising spend.

Seasonality: With the creation of the arrival\_month and unified date columns, we can analyze bookings across seasons and identify peak periods. This enables targeted promotions and staff allocation.