Group 9 - Hotel Market Research in Montreal

Your team works for an international Hotel Chains.
Imagine that your boss would like to know the condition in a specific city, because they want to invest in it. Therefore, your team should base on Airbnb's data and your domain knowledge to provide ideas or suggestions for the boss.

Data sources:
Inside Airbnb: Get the Data
Inside Airbnb: Montreal

Coding references:
New York City Airbnb Open Data | Kaggle
Data Exploration on NYC Airbnb | Kaggle
Airbnb Analysis, Visualization and Prediction | Kaggle
Boston Airbnb Open Data | Kaggle

Our Data Analysis Processes

1. Define the Problem

Clarify the reasons for entering the Montreal hotel market and the expected goals to be achieved.

Product / Service	International Hotel Chain
City Choosing	Montreal, Quebec, Canada
Introduction to Montreal	
Why Choose Montreal	Our groups have two French guys who want to visit Montreal in the future.
Our Goal	To evaluate whether to enter the Montreal market.
Key Performance Indicator	Estimated Revenue = Price * Future Booking days

Key Questions	
Location	Which neighborhood has the most potential for success?
Price	How much should we charge?
Competitiveness	Which factor affects the booking rate??
Strategy	How can we maximize our profit?

2. Data we used

a. Montreal Airbnb Data: listings.csv

Summary information and metrics for listings in Montreal (good for visualizations).

```
df = pd.read_csv('listings.csv')
```

b. Montreal Aribnb Data: calendar.csv

The calendar file records the price, availability and other details from the listing's calendar for each day of the next 365 days.

```
calendar = pd.read_csv('calendar.csv')
```

3. Data Cleaning

3-1. listings.csv

Data Cleaning Common data cleaning steps include:

1. **Duplicate data**: Delete duplicate information.

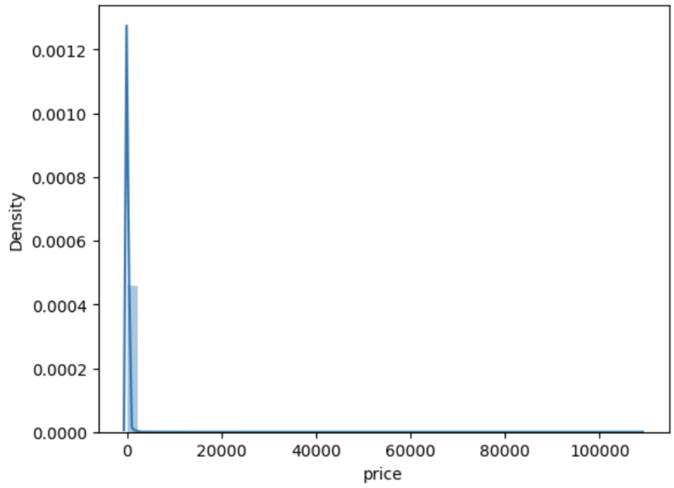
```
df.drop_duplicates(inplace=True)
```

2. **Irrelevant data**: Identify key fields for specific analysis and remove irrelevant data from the analysis.

```
df.drop(columns=['id','neighbourhood_group','last_review'], axis=1)
```

3. **Outliers**: Outliers significantly affect model performance, so you need to identify outliers and determine the appropriate action.

```
sns.distplot(df1['price'])
plt.show()
```



We found that the price distribution doesn't look like a normal distribution, which means the price distribution is affected by outlier data. Then we are going to see what the outlier data is:

name	host_id	host_name	neighbourhood	latitude	longitude	room_type	*price*
room in a shared apartment, the (room ma...	308782...	Sahil Rao	Le Plateau-Mont...	45.5131	-73.56996	Private room	108546
Large sunny room HEARTH of plateau	640693	Marco	Le Plateau-Mont...	45.51937	-73.56999	Private room	13294
Maison 4 étages + voiture	376326...	Anne-Marie	Mercier-Hochela...	45.61355	-73.53198	Entire home/apt	7200
Hotel Epik Montreal, Penthouse	320221...	Hotel Epik	Ville-Marie	45.50367	-73.55439	Private room	7000
DOWNTOWN MONTREAL 12 BEDROOMS ...	7250257	Stewie	Le Plateau-Mont...	45.51607	-73.58292	Entire home/apt	5000
Nice private apartment in the heart of NDG	729453...	Jason	Côte-des-Neige...	45.47145	-73.61348	Entire home/apt	4993
Room Griffintown 2019 / Rent CAD$6000 ...	384046...	Vania	Le Sud-Ouest	45.48952	-73.56633	Private room	4618
Waterfront Old Mtl Condo Floor	460375...	Darrell	Ville-Marie	45.5020...	-73.55400...	Entire home/apt	4574
Boutique Hotel in Montreal Old PORT	421148...	Hygie	Ville-Marie	45.51136	-73.55278	Entire home/apt	4000

Since we can't determine if these outliers are fake data, we decide to keep the data.

4. **Missing data**: Mark and delete or estimate missing data.

```
df1.isnull().sum()
```

name **2**

```
df1[df1['name'].isnull()]
```

	name	host_id	host_name	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_month
156	NaN	1646154	Dimitri	Le Sud-Ouest	45.45053	-73.60031	Entire home/apt	38	40	39	0.32
157	NaN	1646154	Dimitri	Le Sud-Ouest	45.45225	-73.60177	Entire home/apt	40	40	64	0.52

>>> Don't affect our analysis, do nothing.

```
license                                  3901
```

```
# replace nan with "0" in last review and reviews_per_month
df1['reviews_per_month'].fillna(0, inplace=True)
```

>>> replace nan with 0

5. **Structural errors**: Correct printing errors and other inconsistencies and make the data conform to the general pattern or agreement.

3-2. calendar.csv

1. **Duplicate data**: no duplicate data

```
calendar.duplicated().sum()
```

```
0
```

2. **Irrelevant data**: remove columns ‘minimum_nights’ and ‘maximum_nights’

```
calendar.drop(columns=['minimum_nights','maximum_nights'], axis=1,
inplace=True)
```

3. **Missing data**: no missing data

```
calendar.isnull().sum()
```

```
listing_id        0
date              0
available         0
price             0
adjusted_price    0
dtype: int64
```

4. **Structural errors**: before analyzing our data, we need to check the if there is any data type error

```
calendar.dtypes
```

```
listing_id         int64
date              object
available         object
price             object
adjusted_price    object
dtype: object
```

4-1. **date**: convert object to datetime64

```
calendar['date'] = pd.to_datetime(calendar['date'])
```

4-2. **available**: convert t/f to 1/0

```
def tf_to_10(x):
    if x == 't': return 1
    elif x == 'f': return 0
    else: return

calendar.available = calendar.available.apply(lambda x:
tf_to_10(x))
```

4-3. **price, adjusted_price**: remove '$' and ',' in price and adjusted_price, then convert to float

```
calendar[['price','adjusted_price']] =
calendar[['price','adjusted_price']].apply(lambda x:
x.str.replace('$','', regex=True).replace(',','',
regex=True).astype(float))
```

4. Data Analysis

Q1. Which neighborhood has the most potential for success?

a. Revenue Analysis by neighbourhood:

Revenue = Number of days unavailable * price

```
# get the total revenue of each unique listing
revenue_by_listing = calendar[calendar.available ==
0].groupby(['listing_id'])['price'].sum().to_frame().reset_index()
revenue_by_listing.sort_values(by='price', ascending=False, inplace=True)
revenue_by_listing.columns = ['listing_id','revenue']
```

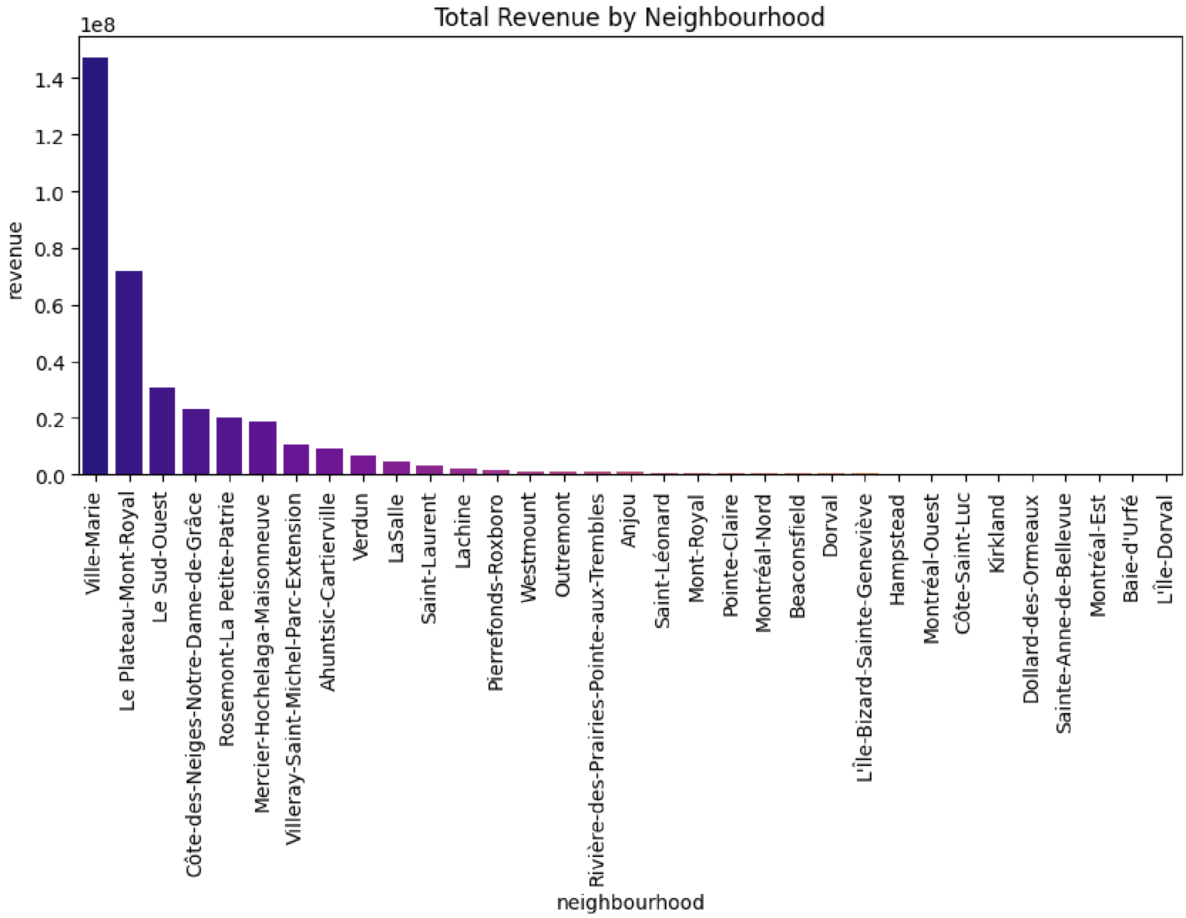
```
# concat the listing name to the dataframe
revenue_by_listing = pd.concat([revenue_by_listing,
df1[['name','neighbourhood','room_type']]], axis=1)
revenue_by_listing.head(5)
```

	listing_id	revenue	name	neighbourhood	room_type
3698	5.308865e+07	39619290.0	downtown mega loft room a	Ville-Marie	Private room
4349	6.152093e+17	3049443.0	private room, 5 minutes walking to metro	Le Sud-Ouest	Private room
2608	4.242870e+07	2555000.0	★spacious 2br ★business/relocation/techhub stay!	Le Plateau-Mont-Royal	Entire home/apt
4892	6.650916e+17	1769930.0	warming townhouse on the plateau mont-royal	Le Plateau-Mont-Royal	Entire home/apt
1520	2.640081e+07	1685570.0	"5 star reviews" spacious loft plateau on mt r...	Le Plateau-Mont-Royal	Entire home/apt

```python
# get the total revenue of each neighbourhood
revenue_by_neighbourhood =
revenue_by_listing.groupby(['neighbourhood'])['revenue'].sum().sort_values(ascen
ding=False).to_frame().reset_index()
revenue_by_neighbourhood.head()
```

	neighbourhood	revenue
0	Ville-Marie	147162421.0
1	Le Plateau-Mont-Royal	71860653.0
2	Le Sud-Ouest	30799970.0
3	Côte-des-Neiges-Notre-Dame-de-Grâce	23445759.0
4	Rosemont-La Petite-Patrie	20432811.0

```python
# plot the total revenue of next 365 days by neighbourhood.
plt.figure(figsize=(10,4))
sns.barplot(data=revenue_by_neighbourhood, x='neighbourhood', y='revenue',
palette='plasma')
plt.title('Total Revenue by Neighbourhood')
plt.xticks(rotation=90)
```

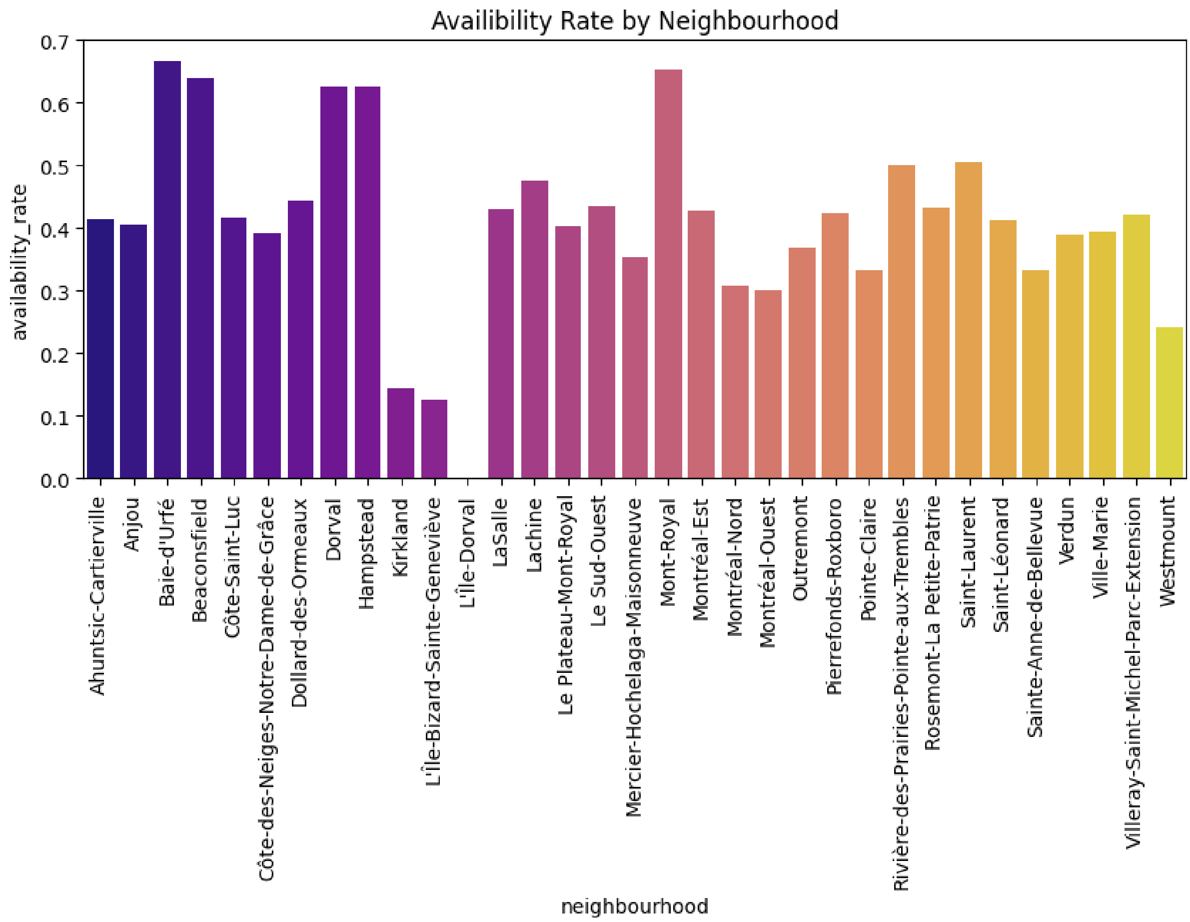


b. Availability Rate Analysis by neighbourhood:

```python
# concat the listing name, neighbourhood, room_type to the calendar dataframe
calendar = pd.concat([calendar, df1[['name','neighbourhood','room_type']]],
axis=1)
calendar.head(3)
```

	listing_id	date	available	price	adjusted_price	name	neighbourhood	room_type
0	29059	2023-04-15	0	99.0	89.0	lovely studio quartier latin	Ville-Marie	Entire home/apt
1	29059	2023-04-16	1	99.0	89.0	maison historique - quartier latin	Ville-Marie	Entire home/apt
2	29059	2023-04-17	0	99.0	89.0	chez patrac ! montreal - métro beaubien	Rosemont-La Petite-Patrie	Entire home/apt

```python
# plot the availability rate of next 365 days by neighbourhood
plt.figure(figsize=(10,4))
sns.barplot(data=availability_rate_by_neighbourhood, x='neighbourhood',
y='availability_rate', palette='plasma')
plt.title('Availibility Rate by Neighbourhood')
plt.xticks(rotation=90)
```



c. Find the neighbourhoods with the high revenue and the low availability rate:

```
listings_count_by_neighbourhood =
calendar.groupby(['neighbourhood'])['listing_id'].count().to_frame().reset_i
ndex()
listings_count_by_neighbourhood.columns = ['neighbourhood','listings_count']
neightbourhoods = pd.concat([revenue_by_neighbourhood,
availability_rate_by_neighbourhood['availability_rate']], axis=1)
neightbourhoods = pd.concat([neightbourhoods,
listings_count_by_neighbourhood['listings_count']], axis=1)
neightbourhoods.sort_values(by='revenue', ascending=False, inplace=True)
print(f'Total Listings: {neightbourhoods.listings_count.sum()}')
neightbourhoods.head(6)
```

	neighbourhood	revenue	availability_rate	listings_count
0	Ville-Marie	147162421	0.413613	191
1	Le Plateau-Mont-Royal	71860653	0.405405	37
2	Le Sud-Ouest	30799970	0.666667	3
3	Côte-des-Neiges-Notre-Dame-de-Grâce	23445759	0.640000	25
4	Rosemont-La Petite-Patrie	20432811	0.416667	12
5	Mercier-Hochelaga-Maisonneuve	18524379	0.391447	608

💡 Our hotel will open in “Ville-Marie” neighborhood.

Q2. How much should we charge?

a. Hotel Price Analysis Based on Ville-Marie

Focusing on the data in Ville-Marie, some room_type of listings are not "Hotel room" but the name contains "hotel", so we can assume that these listings are hotels.

Let’s create a dataframe called "vm_hotel" to store the listings in Ville-Marie with room_type "Hotel room" or name containing "hotel".

```
# convert the name column to lowercase
df1['name'] = df1['name'].apply(lambda x: str(x).lower())
df_hotel = df1[(df1['name'].str.contains('hotel')) | (df1['room_type'] ==
'Hotel room')]

# create new column 'revenue' = price * (365 - availability_365)
df_hotel['revenue'] = df_hotel['price'] * (365 -
df_hotel['availability_365'])
```

```
vm_hotel = df_hotel.loc[df_hotel['neighbourhood'] == 'Ville-Marie']
```

```
vm_hotel[['price','minimum_nights','number_of_reviews','availability_365']].
describe()
```

	price	minimum_nights	number_of_reviews	availability_365
count	171.000000	171.000000	171.000000	171.000000
mean	**278.198830**	1.964912	24.707602	228.883041
std	684.400327	4.586290	37.952089	118.686281
min	37.000000	1.000000	0.000000	0.000000
25%	91.000000	1.000000	3.000000	208.500000
50%	117.000000	1.000000	14.000000	283.000000
75%	238.000000	1.500000	30.000000	312.000000
max	7000.000000	32.000000	261.000000	359.000000

b. Plot the boxplot to better understand the distribution of the price

```
# remove the outliers
Q1 = vm_hotel['price'].quantile(0.25)
Q3 = vm_hotel['price'].quantile(0.75)
IQR = Q3 - Q1
boxplot_price = vm_hotel[(vm_hotel['price'] >= Q1 - 1.5 * IQR) &
(vm_hotel['price'] <= Q3 + 1.5 * IQR)]

# remove the listings with 0 availability_365 and 0 number_of_reviews
boxplot_price = boxplot_price[(boxplot_price['availability_365'] != 0) &
(boxplot_price['number_of_reviews'] != 0)]

# plot the price boxplot Ville-Marie hotels by seaborn
plt.figure(figsize=(10,4))
sns.boxplot(data=boxplot_price, x='price', palette='plasma', orient='h')
plt.title('Price Boxplot of Ville-Marie Hotels (removed outliers)')
```

💡Our price can be set based on the avg. price of our competitor $278. According to the boxplot, $278 is more expensive than 75% of the hotel listings in Ville-Marie, we need to provide high quality service.

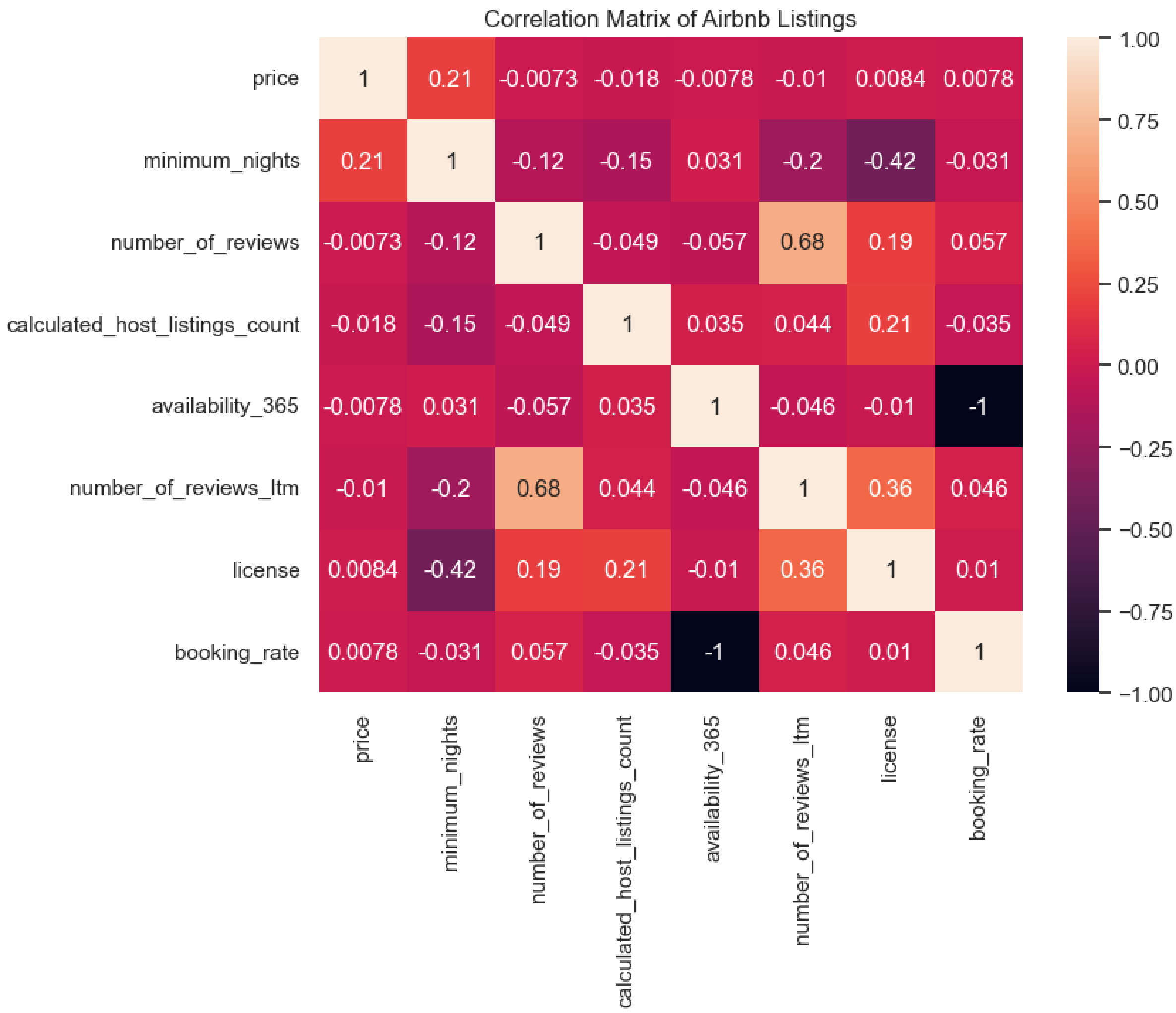
Q3. Which factor affects booking rate?

a. Correlation Analysis

```
df2 =
df1.drop(columns=['name','host_id','host_name','latitude','longitude','revie
ws_per_month'], axis=1, inplace=False)
# create new column 'booking_rate' = (365 - availability_365) / 365
df2['booking_rate'] = (365 - df2['availability_365']) / 365

# Correlation Matrix
corr = df2.corr(method='pearson')

# Correlation Matrix Heatmap by seaborn
plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True)
plt.title('Correlation Matrix of Airbnb Listings')
```



	price	minimum_nights	number_of_reviews	calculated_host_listings_count	availability_365	number_of_reviews_ltm	license	booking_rate
price	1	0.21	-0.0073	-0.018	-0.0078	-0.01	0.0084	0.0078
minimum_nights	0.21	1	-0.12	-0.15	0.031	-0.2	-0.42	-0.031
number_of_reviews	-0.0073	-0.12	1	-0.049	-0.057	0.68	0.19	0.057
calculated_host_listings_count	-0.018	-0.15	-0.049	1	0.035	0.044	0.21	-0.035
availability_365	-0.0078	0.031	-0.057	0.035	1	-0.046	-0.01	-1
number_of_reviews_ltm	-0.01	-0.2	0.68	0.044	-0.046	1	0.36	0.046
license	0.0084	-0.42	0.19	0.21	-0.01	0.36	1	0.01
booking_rate	0.0078	-0.031	0.057	-0.035	-1	0.046	0.01	1

💡As we can see, there is no strong correlation between 'booking rate' and the other variables. (booking rate = (365 - availability_365) / 365).

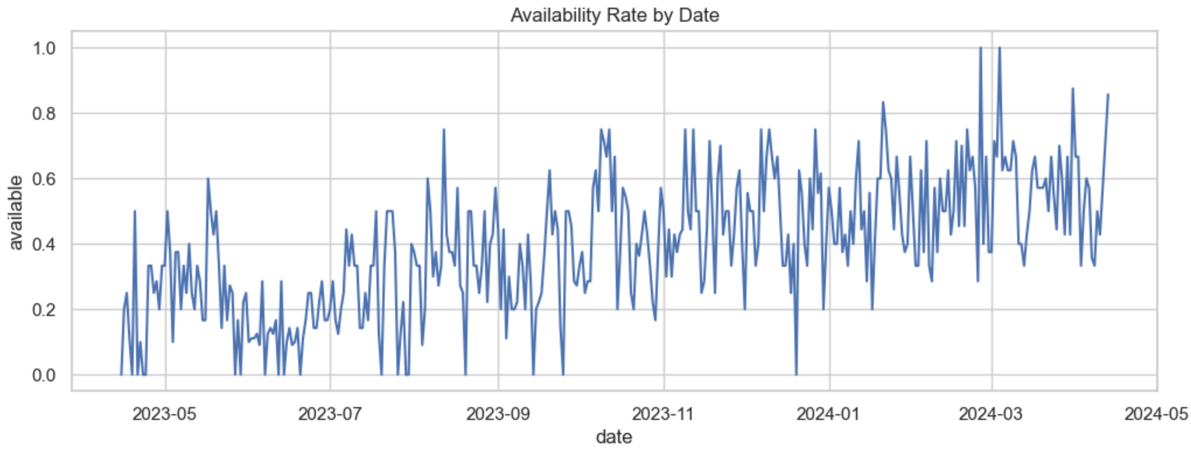
💡Other insight: License helps number of reviews in the last 12 months (r = 0.36). However, the number of reviews and booking rate are very weak corrections (r = 0.057). As a result, we can't say that increasing reviews improves bookings.

Q4. How can we maximize our profit?

a. Availability Rate Analysis in Ville-Marie (Time Series Analysis)

```
# availability rate = available listings / total listings
vm_availability_rate = calendar.loc[calendar['neighbourhood'] ==
'Ville-Marie'].groupby(['date'])['available'].mean().to_frame().reset_index(
)
```

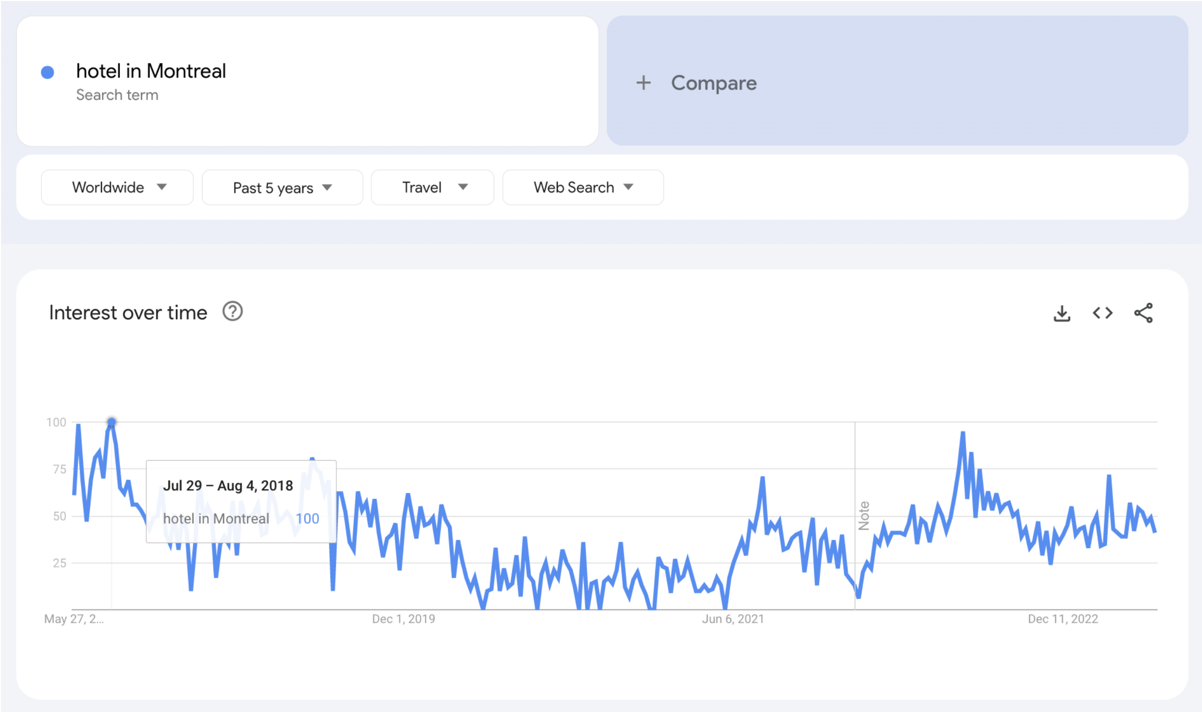
```
# plot the availability rate by date
plt.figure(figsize=(12,4))
sns.lineplot(data=vm_availability_rate, x='date', y='available',
palette='plasma', )
plt.title('Availability Rate by Date')
```



💡 Apparently, the closer of date, the lower Availability Rate is. (everybody knows hahaha)

💡 To know if there is any peak season for traveling to Montreal, we can demonstrate by Google Trends. Peak season is from June to August.

https://trends.google.com.tw/trends/explore?cat=67&date=today%205-y&q=hotel%20in%20Montreal&hl=en



Data Dictionary

listings.csv Summary information and metrics for listings in Montreal (good for visualizations)
• **id**, integer: Airbnb's unique identifier for the listing • **name**, text: Name of the listing • **host_id**, integer: Airbnb's unique identifier for the host/user • **host_name**, text: Name of the host. Usually just the first name(s). • **neighbourhood**, text • **neighbourhood_group**, text: The neighbourhood group as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles. • **latitude**, numeric : Uses the World Geodetic System (WGS84) projection for latitude and longitude. • **longitude**, numeric: Uses the World Geodetic System (WGS84) projection for latitude and longitude. • **room_type**, text: All homes are grouped into the following three room types: Private room, Shared room, Entire place, Private rooms, Shared rooms, Hotel • **price**, currency: daily price in local currency • **minimum_nights**, integer: minimum number of night stay for the listing (calendar rules may be different) • **number_of_reviews**, integer: The number of reviews the listing has • **last_review**, date: The date of the last/newest review

- **calculated_host_listings_count**, integer: The number of listings the host has in the current scrape, in the city/region geography. - **availability_365**, integer: The availability of the listing 365 days in the future as determined by the calendar. Note a listing may not be available because it has been booked by a guest or blocked by the host. - **number_of_reviews_ltm, integer**: The number of reviews - the listing has (in the last 12 months) - **license**, text: The license/permit/registration number

calendar.csv The calendar file records the price, availability and other details from the listing's calendar for each day of the next 365 days.
- **listing_id**, integer: Airbnb's unique identifier for the listing - **date**, datetime: The date in the listing's calendar - **available**, boolean: Whether the date is available for a booking - **price**, currency: The price listed for the day - **adjusted_price**, currency: The price after adjusting for discounts, cleaning fees, etc. - **minimum_nights**, integer: The minimum number of nights required to book the listing on this day - **maximum_nights**, integer: The maximum number of nights required to book the listing on this day

Appedix

Demographics	**Population**: 1,649,520 **Ethnicity**: European 60.3%, African 11.5%, Middle Eastern 9.3%, South Asian 4.6%, Latin American 4.5%), Southeast Asian 3.8%, East Asian, Indigenous 0.9% and Other/Multiracial 1.3% **Language**: French 47.0%, English 13%, Other 32.8% **Religion**: Christian 49.5%, No Religion 31%, Muslim 12.7%, Jewish 2.1% [References] Montreal - Wikipedia Profile table, Census Profile, 2021 Census of Population - Montréal, Ville (V) [Census subdivision], Quebec
Market Size (Annual Tourists)	2019: 11 million 2022: 8 million

<table>
<tr><td></td><td>

2023: 9.5 million (prediction)

[References]
Montreal - mtl.org

</td></tr>
<tr><td>Map</td><td>

</td></tr>
<tr><td>Consumer Preferences</td><td>

(Analyze the booking data to find the attributes of the hottest bnb) e.g. location, pricing, type, service)

[Reference]
How Airbnb Has Disrupted the Hotel Management Industry | Verdant

Customers aren't necessarily embracing the platform because they prefer it over hotels. Sometimes, **Airbnb is the only feasible option for many customers**. Airbnb's greatest success stories come from **cities with limited room availability during peak seasons**, such as New York, Los Angeles, and San Francisco.

>>> How can a hotel adjust its strategy?

The Welfare Effects of Peer Entry in the Accommodation Market: The Case of Airbnb | NBER

</td></tr>
<tr><td>Competitors (Hospitality Industry)</td><td>Other hotels, B&B, Inn, Hostel, Motel, Resort, Villa…</td></tr>
</table>