1. Import Libraries

In [1]:
```
# Import Libraries

import pandas as pd
import numpy as np
import os
```

2. Import Dataframe

In [2]:
```
path = r'C:\Users\admin\Desktop\Data Analyst Case Study'
```

In [3]:
```
path
```

Out[3]:
```
'C:\\Users\\admin\\Desktop\\Data Analyst Case Study'
```

In [4]:
```
# Read the excel file '202302_Task1_sessions.xlsx' from the specified path

df = pd.read_excel(os.path.join(path, 'Data', 'Original Data', '202303_Task1_Sessions.xlsx'), index_col = False)
```

In [5]:
```
import pandas as pd

# Read the excel file

df = pd.read_excel(r'C:\Users\admin\Desktop\Data Analyst Case Study\Data\Original Data\202303_Task1_Sessions.xlsx', 
```

3. Data Cleaning / Wrangling

In [6]:
```
df.head(30)
```

Out[6]:

	ymd,session_id,tracking_id,platform,is_app,is_repeater,traffic_type,country_name,agent_id,clickouts,bookings,session_duration,entry_page,total
0	20220626,202206262(
1	20220518,2022051821
2	20220508,20220508210
3	20220507,20220507060
4	20220523,202205232
5	20220618,20220618
6	20220609,2022060908(
7	20220523,20220523126
8	20220521,2022052109
9	20220506,2022050618
10	20220515,2022051523
11	20220512,2022051216(
12	20220518,2022051803(
13	20220624,2022062422
14	20220625,2022062506
15	20220610,20220610110
16	20220605,2022060518
17	20220531,2022053115
18	20220516,2022051613
19	20220602,2022060215
20	20220621,20220621233
21	20220619,20220619066
22	20220520,20220520060
23	20220617,202206171

	ymd,session_id,tracking_id,platform,is_app,is_repeater,traffic_type,country_name,agent_id,clickouts,bookings,session_duration,entry_page,total
24	20220615,2022061513(
25	20220602,202206021(
26	20220503,202205031(
27	20220601,2022060116
28	20220627,202206270
29	20220517,20220517209

In [7]:
```
print(df.columns)
```

```
Index(['ymd,session_id,tracking_id,platform,is_app,is_repeater,traffic_type,country_name,agent_id,clickouts,booking
s,session_duration,entry_page,total_ctp,arrival_day,departure_day'], dtype='object')
```

In [8]:
```
# Split the data into columns
df_split = df.iloc[:, 0].str.split(',', expand=True)

# Check the number of columns
print(f"Number of columns: {df_split.shape[1]}")

# Display the first few rows to inspect the structure
print(df_split.head())
```

```
Number of columns: 18
         0                    1           2   3  4  5  6              7   8   \
0  20220626  2022062620046057322  FA6JXA8TAJ  UK  0  1  2  United Kingdom  16
1  20220518  2022051821943006017  0X7RLU6KF7  BR  0  0  2         Brazil   2
2  20220508  2022050821020053928  0I59VWLQW0  UK  0  0  2  United Kingdom  20
3  20220507  2022050706015039122  JXNHOBQL50  CH  0  0  2     Switzerland  28
4  20220523  2022052320052048087  W24I0V5Z2L  IT  0  0  2           Italy  20

  9  10    11    12   13        14        15    16    17
0  0  0    29  2111    0        \N        \N  None  None
1  3  0  1485  2100   27  20220530  20220531  None  None
2  0  0   143  2100    0        \N        \N  None  None
3  0  0    69  2100    0        \N        \N  None  None
4  6  0   887  2100  100  20220609  20220610  None  None
```

In [9]:
```
# Assign proper column names
df_split.columns = [
    "ymd", "session_id", "tracking_id", "platform", "is_app", "is_repeater",
    "traffic_type", "country_name", "agent_id", "clickouts", "bookings",
    "session_duration", "entry_page", "total_ctp", "arrival_day", "departure_day",
    "extra_col_1", "extra_col_2"
]

# Display the cleaned DataFrame
print(df_split.head())
```

```
        ymd           session_id tracking_id platform is_app is_repeater  \
0  20220626  2022062620046057322  FA6JXA8TAJ       UK      0           1
1  20220518  2022051821943006017  0X7RLU6KF7       BR      0           0
2  20220508  2022050821020053928  0I59VWLQW0       UK      0           0
3  20220507  2022050706015039122  JXNHOBQL50       CH      0           0
4  20220523  2022052320052048087  W24I0V5Z2L       IT      0           0

  traffic_type    country_name agent_id clickouts bookings session_duration  \
0            2  United Kingdom       16         0        0               29
1            2          Brazil        2         3        0             1485
2            2  United Kingdom       20         0        0              143
3            2     Switzerland       28         0        0               69
4            2           Italy       20         6        0              887

  entry_page total_ctp arrival_day departure_day extra_col_1 extra_col_2
0       2111         0          \N            \N        None        None
1       2100        27    20220530      20220531        None        None
2       2100         0          \N            \N        None        None
3       2100         0          \N            \N        None        None
4       2100       100    20220609      20220610        None        None
```

In [10]:
```
df_split.head(30)
```

Out[10]:

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	booking
0	20220626	2022062620046057322	FA6JXA8TAJ	UK	0	1	2	United Kingdom	16	0	
1	20220518	2022051821943006017	0X7RLU6KF7	BR	0	0	2	Brazil	2	3	
2	20220508	2022050821020053928	0I59VWLQW0	UK	0	0	2	United Kingdom	20	0	
3	20220507	2022050706015039122	JXNHOBQL50	CH	0	0	2	Switzerland	28	0	
4	20220523	2022052320052048087	W24I0V5Z2L	IT	0	0	2	Italy	20	6	
5	20220618	2022061819050074027	III5DT3FFI	RU	0	0	2	Russia	18	1	
6	20220609	2022060908077048629	RK5EHUA3SV	UK	0	0	2	United Kingdom	16	0	
7	20220523	2022052312665009565	X7L34ZN7VH	TW	0	0	2	Taiwan	2	1	
8	20220521	2022052109090009274	4YZZV5US8E	FR	0	0	2	France	4	0	
9	20220506	2022050618041006990	BLSREVJXXU	HU	0	0	2	Hungary	20	0	
10	20220515	2022051523920004707	SHJTUTIZPN	AR	0	1	2	Argentina	20	0	
11	20220512	2022051216075016751	0OCXTTSXOL	RU	0	0	2	Russian Federation	20	2	
12	20220518	2022051803654008080	ITKH1ZED2G	MY	0	1	2	Malaysia	2	0	
13	20220624	2022062422973013513	2SZ5X9C4K1	AR	0	0	2	Argentina	20	3	
14	20220625	2022062506090040942	V3U98RLKQI	AA	0	1	2	Qatar	12	0	
15	20220610	2022061011089010517	Q4AR3V1LWC	UK	0	0	2	United Kingdom	2	2	
16	20220605	2022060518052078255	805I4VX05W	FR	0	0	2	France	20	1	
17	20220531	2022053115065003535	I1EVQZWSMI	SE	0	0	2	Sweden	18	2	
18	20220516	2022051613309007347	QPVI455BW5	US	0	1	2	United States	2	0	
19	20220602	2022060215959008735	ALI6H00KJN	CA	0	1	2	Canada	2	1	
20	20220621	2022062123325016453	4U2U783W76	US	0	0	2	United States	2	1	

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	booking
21	20220619	2022061906623011012	X5SR35FNWV	MY	0	0	2	Malaysia	20	1	
22	20220520	2022052006039048210	MQ92M6B3E4	AT	0	1	2	Austria	18	3	
23	20220617	2022061714092055412	8ERJHLVESY	AT	0	1	2	Austria	16	2	
24	20220615	2022061513008027766	UBRH7U3913	ZA	0	0	2	South Africa	2	2	
25	20220602	2022060219954008494	4U82769KL3	CL	0	0	2	Chile	16	0	
26	20220503	2022050319022007303	NPI3Z68XAP	TR	0	0	2	Malta	2	0	
27	20220601	2022060116933008189	09G5FVVIWL	BR	0	0	2	Brazil	20	2	
28	20220627	2022062707079035783	E906HYTFPB	ES	0	0	10	Spain	20	0	
29	20220517	2022051720968006777	OO4MCAZ466	CL	0	0	2	Chile	2	1	

In [11]:
```
# Drop the last two columns

df_split = df_split.iloc[:, :-2]
```

In [12]:
```
df_split.head(30)
```

Out[12]:

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	booking
0	20220626	2022062620046057322	FA6JXA8TAJ	UK	0	1	2	United Kingdom	16	0	
1	20220518	2022051821943006017	0X7RLU6KF7	BR	0	0	2	Brazil	2	3	
2	20220508	2022050821020053928	0I59VWLQW0	UK	0	0	2	United Kingdom	20	0	
3	20220507	2022050706015039122	JXNHOBQL50	CH	0	0	2	Switzerland	28	0	
4	20220523	2022052320052048087	W24I0V5Z2L	IT	0	0	2	Italy	20	6	
5	20220618	2022061819050074027	III5DT3FFI	RU	0	0	2	Russia	18	1	
6	20220609	2022060908077048629	RK5EHUA3SV	UK	0	0	2	United Kingdom	16	0	
7	20220523	2022052312665009565	X7L34ZN7VH	TW	0	0	2	Taiwan	2	1	
8	20220521	2022052109090009274	4YZZV5US8E	FR	0	0	2	France	4	0	
9	20220506	2022050618041006990	BLSREVJXXU	HU	0	0	2	Hungary	20	0	
10	20220515	2022051523920004707	SHJTUTIZPN	AR	0	1	2	Argentina	20	0	
11	20220512	2022051216075016751	0OCXTTSXOL	RU	0	0	2	Russian Federation	20	2	
12	20220518	2022051803654008080	ITKH1ZED2G	MY	0	1	2	Malaysia	2	0	
13	20220624	2022062422973013513	2SZ5X9C4K1	AR	0	0	2	Argentina	20	3	
14	20220625	2022062506090040942	V3U98RLKQI	AA	0	1	2	Qatar	12	0	
15	20220610	2022061011089010517	Q4AR3V1LWC	UK	0	0	2	United Kingdom	2	2	
16	20220605	2022060518052078255	805I4VX05W	FR	0	0	2	France	20	1	
17	20220531	2022053115065003535	I1EVQZWSMI	SE	0	0	2	Sweden	18	2	
18	20220516	2022051613309007347	QPVI455BW5	US	0	1	2	United States	2	0	
19	20220602	2022060215959008735	ALI6H00KJN	CA	0	1	2	Canada	2	1	
20	20220621	2022062123325016453	4U2U783W76	US	0	0	2	United States	2	1	

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	booking
21	20220619	2022061906623011012	X5SR35FNWV	MY	0	0	2	Malaysia	20	1	0
22	20220520	2022052006039048210	MQ92M6B3E4	AT	0	1	2	Austria	18	3	0
23	20220617	2022061714092055412	8ERJHLVESY	AT	0	1	2	Austria	16	2	0
24	20220615	2022061513008027766	UBRH7U3913	ZA	0	0	2	South Africa	2	2	0
25	20220602	2022060219954008494	4U82769KL3	CL	0	0	2	Chile	16	0	0
26	20220503	2022050319022007303	NPI3Z68XAP	TR	0	0	2	Malta	2	0	0
27	20220601	2022060116933008189	09G5FVVIWL	BR	0	0	2	Brazil	20	2	0
28	20220627	2022062707079035783	E906HYTFPB	ES	0	0	10	Spain	20	0	0
29	20220517	2022051720968006777	OO4MCAZ466	CL	0	0	2	Chile	2	1	0

In [13]:
```
# perform data info check
print(df_split.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900000 entries, 0 to 899999
Data columns (total 16 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   ymd               900000 non-null  object
 1   session_id        900000 non-null  object
 2   tracking_id       900000 non-null  object
 3   platform          900000 non-null  object
 4   is_app            900000 non-null  object
 5   is_repeater       900000 non-null  object
 6   traffic_type      900000 non-null  object
 7   country_name      900000 non-null  object
 8   agent_id          900000 non-null  object
 9   clickouts         900000 non-null  object
 10  bookings          900000 non-null  object
 11  session_duration  900000 non-null  object
 12  entry_page        900000 non-null  object
 13  total_ctp         900000 non-null  object
 14  arrival_day       900000 non-null  object
 15  departure_day     900000 non-null  object
dtypes: object(16)
memory usage: 109.9+ MB
None
```

In [14]:
```
df_split.dtypes
```

Out[14]:
```
ymd               object
session_id        object
tracking_id       object
platform          object
is_app            object
is_repeater       object
traffic_type      object
country_name      object
agent_id          object
clickouts         object
bookings          object
session_duration  object
entry_page        object
total_ctp         object
arrival_day       object
departure_day     object
dtype: object
```

In [15]:
```
df_split.describe()
```

Out[15]:

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	bool
count	900000	900000	900000	900000	900000	900000	900000	900000	900000	900000	90
unique	61	900000	892416	55	1	2	5	252	19	69	
top	20220501	2022062620046057322	L47NTB4H8A	US	0	0	6	United States	20	0	
freq	18332	1	9	117589	900000	537261	298502	106830	354645	547389	89

In [16]:
```
# Convert specific columns to numeric
numeric_columns = ["is_app", "is_repeater", "clickouts", "bookings", "session_duration"]
df_split[numeric_columns] = df_split[numeric_columns].apply(pd.to_numeric, errors='coerce')
```

In [17]:
```
# Verify the data types
print(df_split.dtypes)
```

```
ymd                 object
session_id          object
tracking_id         object
platform            object
is_app               int64
is_repeater          int64
traffic_type        object
country_name        object
agent_id            object
clickouts          float64
bookings             int64
session_duration     int64
entry_page          object
total_ctp           object
arrival_day         object
departure_day       object
dtype: object
```

In [18]:
```
df_split.describe()
```

Out[18]:

	is_app	is_repeater	clickouts	bookings	session_duration
count	900000.0	900000.000000	899993.000000	900000.000000	900000.000000
mean	0.0	0.403043	0.906624	0.012369	390.921253
std	0.0	0.490510	2.092830	0.147544	987.959027
min	0.0	0.000000	0.000000	0.000000	0.000000
25%	0.0	0.000000	0.000000	0.000000	12.000000
50%	0.0	0.000000	0.000000	0.000000	64.000000
75%	0.0	1.000000	1.000000	0.000000	285.000000
max	0.0	1.000000	86.000000	18.000000	83335.000000

4. Handle missing values

In [19]:
```
# count missing values in each column
print(df_split.isnull().sum())
```

```
ymd                0
session_id         0
tracking_id        0
platform           0
is_app             0
is_repeater        0
traffic_type       0
country_name       0
agent_id           0
clickouts          7
bookings           0
session_duration   0
entry_page         0
total_ctp          0
arrival_day        0
departure_day      0
dtype: int64
```

In [20]:
```
print(df_split[df_split == "\\N"].count())
```

```
ymd                     0
session_id              0
tracking_id             0
platform                0
is_app                  0
is_repeater             0
traffic_type            0
country_name          725
agent_id                0
clickouts               0
bookings                0
session_duration        0
entry_page              0
total_ctp               0
arrival_day        547389
departure_day      549817
dtype: int64
```

In [21]:
```
# replace \N with Na for easier handling
df_split.replace("\\N", pd.NA, inplace=True)
```

In [22]:
```
# fill numeric columns with the mean
df_split['clickouts']. fillna(df_split['clickouts'].mean(), inplace=True)
```

In [23]:
```
# fill catergorical column with unknown
df_split['country_name'].fillna("Unknown", inplace=True)
```

In [24]:
```
# Keep rows with at least 5 non-NA values
df_split.dropna(thresh=5, inplace=True)
```

In [25]:
```
# Remove Duplicates
df_split.drop_duplicates(subset=['session_id'], inplace=True)
```

5. Convert Data types

In [26]:
```
# convert the columns to datetime format using the specified format
df_split['ymd'] = pd.to_datetime(df_split['ymd'], format='%Y%m%d')
df_split['arrival_day'] = pd.to_datetime(df_split['arrival_day'], format='%Y%m%d', errors='coerce')
df_split['departure_day'] = pd.to_datetime(df_split['departure_day'], format='%Y%m%d', errors='coerce')
```

In [27]:
```
# Convert 'clickouts' column to numeric, coercing invalid values to NaN
df_split['clickouts'] = pd.to_numeric(df_split['clickouts'], errors='coerce')
df_split['bookings'] = pd.to_numeric(df_split['bookings'], errors='coerce')
```

6. Standardize and Normalize Data

In [29]:
```
# Standardize text in categorical columns
df_split['platform'] = df_split['platform'].str.lower().str.strip()
df_split['country_name'] = df_split['country_name'].str.title().str.strip()
```

In [30]:
```
# Remove Extra Spaces
df_split['entry_page'] = df_split['entry_page'].str.strip()
```

In [31]:
```
# Example for 'session_duration' column
Q1 = df_split['session_duration'].quantile(0.25)
Q3 = df_split['session_duration'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter rows within the bounds
df_split = df_split[(df_split['session_duration'] >= lower_bound) & (df_split['session_duration'] <= upper_bound)]
```

In [32]:
```
df_split = df_split[df_split['departure_day'] >= df_split['arrival_day']]
```

In [33]:
```
df_split.head(5)
```

Out[33]:

	ymd	session_id	tracking_id	platform	is_app	is_repeater	traffic_type	country_name	agent_id	clickouts	bookin
5	2022-06-18	2022061819050074027	III5DT3FFI	ru	0	0	2	Russia	18	1.0	
7	2022-05-23	2022052312665009565	X7L34ZN7VH	tw	0	0	2	Taiwan	2	1.0	
11	2022-05-12	2022051216075016751	0OCXTTSXOL	ru	0	0	2	Russian Federation	20	2.0	
15	2022-06-10	2022061011089010517	Q4AR3V1LWC	uk	0	0	2	United Kingdom	2	2.0	
16	2022-06-05	2022060518052078255	805I4VX05W	fr	0	0	2	France	20	1.0	

In [34]:
```
print(df_split.describe())  # Summary statistics
```

```
                                ymd    is_app    is_repeater      clickouts  \
count                        254177  254177.0  254177.000000  254177.000000
mean   2022-05-30 19:26:51.407798528       0.0       0.431648       1.581587
min              2022-05-01 00:00:00       0.0       0.000000       1.000000
25%              2022-05-15 00:00:00       0.0       0.000000       1.000000
50%              2022-05-31 00:00:00       0.0       0.000000       1.000000
75%              2022-06-15 00:00:00       0.0       1.000000       2.000000
max              2022-06-30 00:00:00       0.0       1.000000      41.000000
std                              NaN       0.0       0.495307       1.092401

            bookings  session_duration                   arrival_day  \
count  254177.000000     254177.000000                        254177
mean        0.016150        212.479268  2022-07-17 18:16:32.313230592
min         0.000000          0.000000            2021-04-02 00:00:00
25%         0.000000         71.000000            2022-06-09 00:00:00
50%         0.000000        158.000000            2022-07-01 00:00:00
75%         0.000000        315.000000            2022-08-04 00:00:00
max         8.000000        694.000000            2023-06-30 00:00:00
std         0.130259        175.382184                            NaN

                     departure_day
count                       254177
mean   2022-07-20 09:52:58.130200320
min              2021-04-03 00:00:00
25%              2022-06-11 00:00:00
50%              2022-07-03 00:00:00
75%              2022-08-08 00:00:00
max              2023-08-31 00:00:00
std                              NaN
```

7. Export data

In [ ]:
```
df_split.to_excel(os.path.join(path, 'Data','Prepared Data', 'cleaned_data.xlsx'))
```

In [ ]:
```
df_split.head(5)
```