EDA

September 24, 2024

[1]:
```
import pandas as pd
import numpy as np
import os

import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df = pd.read_csv("C:/Users/bendh/Desktop/data science/JN/customer_booking.csv",␣
↪encoding="ISO-8859-1")
df.head()
```

[2]:

	num_passengers	sales_channel	trip_type	purchase_lead	length_of_stay	flight_hour	flight_day	route	booking_origin	wants_extra_baggage	wants_preferred_seat	wants_in_flight_meals	flight_duration	booking_complete
0	2	Internet	RoundTrip	262	19	7	Sat	AKLDEL	New Zealand	1	0	0	5.52	0
1	1	Internet	RoundTrip	112	20	3	Sat	AKLDEL	New Zealand	0	0	0	5.52	0
2	2	Internet	RoundTrip	243	22	17	Wed	AKLDEL	India	1	1	0	5.52	0
3	1	Internet	RoundTrip	96	31	4	Sat	AKLDEL	New Zealand	0	0	1	5.52	0
4	2	Internet	RoundTrip	68	22	15	Wed	AKLDEL	India	1	0	1	5.52	0

[3]:
```
df.shape
```

[3]:
```
(50000, 14)
```

[4]:
```
df.describe()
```

[4]:
```
       num_passengers  purchase_lead  length_of_stay  flight_hour  \
count    50000.000000   50000.000000     50000.00000  50000.00000
mean         1.591240      84.940480        23.04456      9.06634
std          1.020165      90.451378        33.88767      5.41266
min          1.000000       0.000000         0.00000      0.00000
25%          1.000000      21.000000         5.00000      5.00000
50%          1.000000      51.000000        17.00000      9.00000
75%          2.000000     115.000000        28.00000     13.00000
max          9.000000     867.000000       778.00000     23.00000

       wants_extra_baggage  wants_preferred_seat  wants_in_flight_meals  \
count         50000.000000          50000.000000           50000.000000
mean              0.668780              0.296960               0.427140
std               0.470657              0.456923               0.494668
min               0.000000              0.000000               0.000000
25%               0.000000              0.000000               0.000000
50%               1.000000              0.000000               0.000000
75%               1.000000              1.000000               1.000000
max               1.000000              1.000000               1.000000

       flight_duration  booking_complete
count     50000.000000      50000.000000
mean          7.277561          0.149560
std           1.496863          0.356643
min           4.670000          0.000000
25%           5.620000          0.000000
50%           7.570000          0.000000
75%           8.830000          0.000000
max           9.500000          1.000000
```

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   num_passengers         50000 non-null  int64
 1   sales_channel          50000 non-null  object
 2   trip_type              50000 non-null  object
 3   purchase_lead          50000 non-null  int64
 4   length_of_stay         50000 non-null  int64
```

```
 5   flight_hour             50000 non-null  int64
 6   flight_day              50000 non-null  object
 7   route                   50000 non-null  object
 8   booking_origin          50000 non-null  object
 9   wants_extra_baggage     50000 non-null  int64
 10  wants_preferred_seat    50000 non-null  int64
 11  wants_in_flight_meals   50000 non-null  int64
 12  flight_duration         50000 non-null  float64
 13  booking_complete        50000 non-null  int64
dtypes: float64(1), int64(8), object(5)
memory usage: 5.3+ MB
```

0.1 Sales Channel

[6]:
```
#sales channel booking was made on (internet/phone call)
internet_sales_channel = df.sales_channel.value_counts().values[0] / df.
 ↪sales_channel.count() * 100
phone_sales_channel = df.sales_channel.value_counts().values[1] / df.
 ↪sales_channel.count() * 100
```

[7]:
```
print(f"Percentage of booking made through internet:{internet_sales_channel}%")
print(f"Percentage of booking made through phone calls:{phone_sales_channel}%")
```

```
Percentage of booking made through internet:88.764%
Percentage of booking made through phone calls:11.236%
```

0.2 Trip type

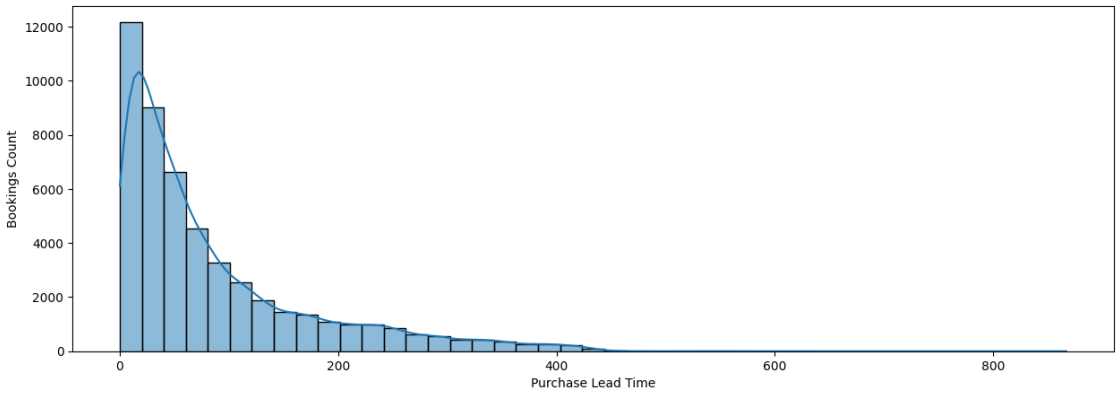
[8]:
```
#type of the trip (Round trip/One way/ Circle trip)
round_trip_type = df.trip_type.value_counts().values[0] / df.trip_type.count()␣
 ↪* 100
one_way_trip_type = df.trip_type.value_counts().values[1] / df.trip_type.
 ↪count() * 100
circle_trip_type = df.trip_type.value_counts().values[2] / df.trip_type.count()␣
 ↪* 100
```

[9]:
```
print(f"Percentage of booking a round trip type:{round_trip_type}%")
print(f"Percentage of booking a one way trip type:{one_way_trip_type}%")
print(f"Percentage of booking a circle trip type:{circle_trip_type}%")
```

```
Percentage of booking a round trip type:98.994%
Percentage of booking a one way trip type:0.774%
Percentage of booking a circle trip type:0.232%
```

0.3 Purchase Lead

[10]:
```
#number of days between travel date and booking date

# Set the figure size
plt.figure(figsize=(15,5))

# Create the histogram with KDE
sns.histplot(data=df, x="purchase_lead", binwidth=20, kde=True)

# Set the x-axis label with units (days)
plt.xlabel("Purchase Lead Time", fontsize=10)

# Set the y-axis label with a description (number of people)
plt.ylabel("Bookings Count", fontsize=10)

# Show the plot
plt.show()
```



[11]:
```
#removing the outliers (like the bookings that are made more than 2 year before␣
 ↪the flight (illogic))
(df.purchase_lead > 600 ).value_counts()
```

[11]:
```
purchase_lead
False    49992
True         8
Name: count, dtype: int64
```

[12]:
```
df[df.purchase_lead > 600]
```

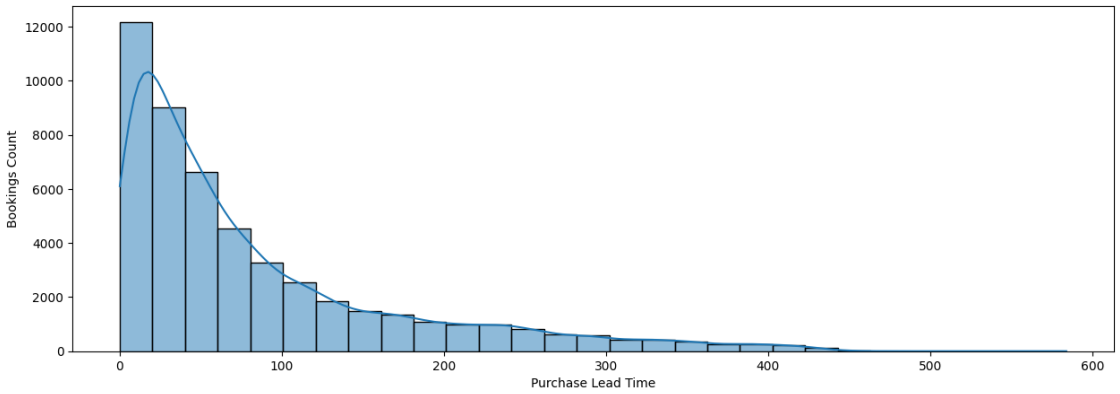
[12]:
```
       num_passengers sales_channel  trip_type  purchase_lead  length_of_stay  \
835                 3      Internet  RoundTrip            641              46
6148                1      Internet  RoundTrip            614              19
```

```
24119               1      Internet  RoundTrip           704              23
38356               2      Internet  RoundTrip           633               5
39417               1        Mobile  RoundTrip           625               5
42916               1        Mobile  RoundTrip           605               6
46716               2      Internet  RoundTrip           606               6
48259               3      Internet  RoundTrip           867               6

       flight_hour flight_day   route   booking_origin  wants_extra_baggage  \
835              6        Sun  AKLKUL         Malaysia                    1
6148            11        Wed  COKMEL        Australia                    0
24119            8        Tue  PNHSYD        Australia                    0
38356           10        Sat  HKTOOL        Australia                    0
39417           15        Fri  ICNRGN  Myanmar (Burma)                    0
42916           18        Thu  BLRMEL            India                    0
46716            6        Fri  HKTTPE    United States                    0
48259            7        Mon  KIXMLE            Japan                    0

       wants_preferred_seat  wants_in_flight_meals  flight_duration  \
835                       0                      1             8.83
6148                      0                      0             8.83
24119                     0                      0             8.58
38356                     0                      1             8.83
39417                     0                      0             6.62
42916                     0                      0             8.83
46716                     0                      1             4.67
48259                     0                      1             7.00

       booking_complete
835                   1
6148                  0
24119                 0
38356                 0
39417                 0
42916                 0
46716                 0
48259                 1
```

[13]:
```
#having only the data of bookings before 600 days
df = df[df.purchase_lead < 600]
```

[14]:
```
#number of days between travel date and booking date

# Set the figure size
plt.figure(figsize=(15,5))

# Create the histogram with KDE
sns.histplot(data=df, x="purchase_lead", binwidth=20, kde=True)
```

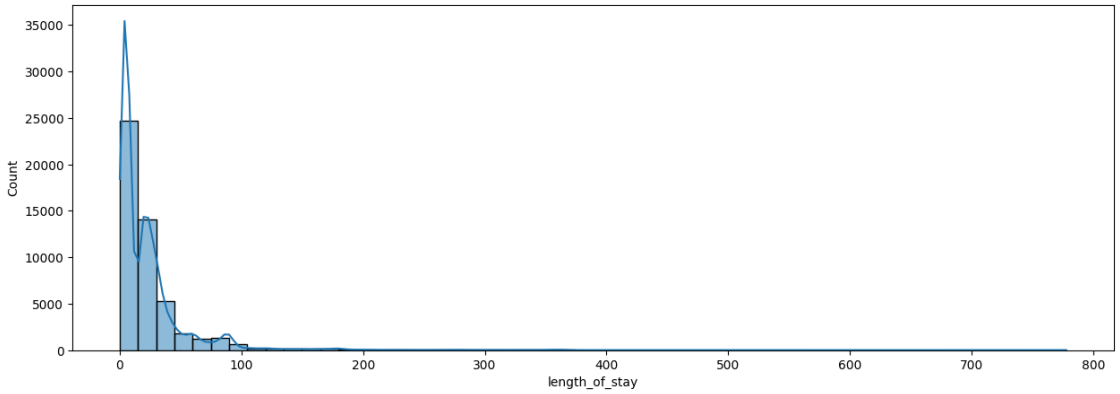
```
# Set the x-axis label with units (days)
plt.xlabel("Purchase Lead Time", fontsize=10)

# Set the y-axis label with a description (number of people)
plt.ylabel("Bookings Count", fontsize=10)

# Show the plot
plt.show()
```



0.4 Length of Stay

[15]:
```
plt.figure(figsize=(15,5))
sns.histplot(data=df, x="length_of_stay", binwidth=15,kde=True)
```

[15]: <Axes: xlabel='length_of_stay', ylabel='Count'>



[16]:
```
(df.length_of_stay> 200).value_counts()
```

[16]:
```
length_of_stay
False    49713
True       279
Name: count, dtype: int64
```

[17]:
```
df[df.length_of_stay> 500].booking_complete.value_counts()
```

[17]:
```
booking_complete
0    9
1    1
Name: count, dtype: int64
```

[18]:
```
#filtering the data to have only length of stay days less than 500 days
df = df[df.purchase_lead <500 ]
```

0.5 Flight Day

[19]:
```
mapping = {
    "Mon" : 1,
    "Tue" : 2,
    "Wed" : 3,
    "Thu" : 4,
    "Fri" : 5,
    "Sat" : 6,
    "Sun" : 7
}

df.flight_day = df.flight_day.map(mapping)
```

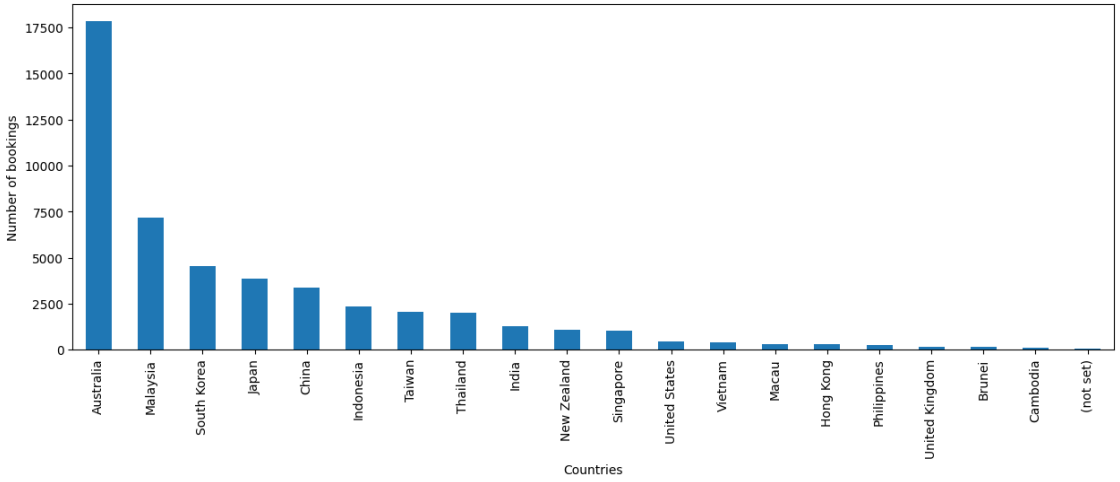
[20]:
```
df.flight_day.value_counts()
```

[20]:
```
flight_day
1    8100
3    7671
2    7670
4    7423
5    6759
7    6550
6    5809
Name: count, dtype: int64
```

0.6 Booking Origin

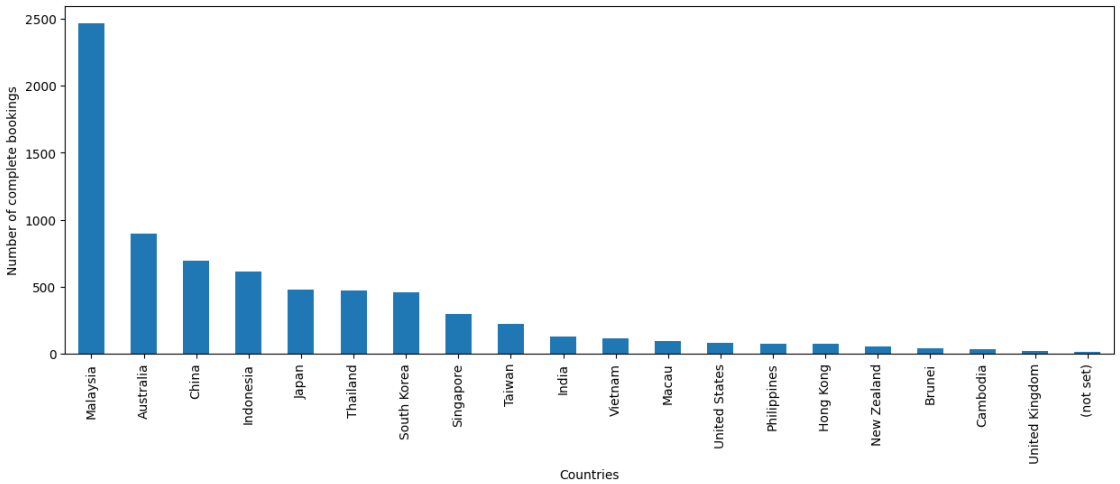
[21]:
```
plt.figure(figsize=(15,5))
ax = df.booking_origin.value_counts()[:20].plot(kind="bar")
ax.set_xlabel("Countries")
ax.set_ylabel("Number of bookings")
```

[21]: Text(0, 0.5, 'Number of bookings')



[22]:
```
plt.figure(figsize=(15,5))
ax = df[df.booking_complete ==1].booking_origin.value_counts()[:20].
  ↪plot(kind="bar")
ax.set_xlabel("Countries")
ax.set_ylabel("Number of complete bookings")
```

[22]: Text(0, 0.5, 'Number of complete bookings')



0.7 Booking Complete

[23]:
```
successful_booking_per = df.booking_complete.value_counts().values[0] / len(df)␣
↪* 100
```

[24]:
```
unsuccessful_booking_per = 100-successful_booking_per
```

[25]:
```
print(f"Out of 50000 booking entries only {round(unsuccessful_booking_per,2)} %␣
↪bookings were successfull or complete.")
```

```
Out of 50000 booking entries only 14.96 % bookings were successfull or complete.
```

0.7.1 Export dataset to CSV

[27]:
```
df.to_csv("C:/Users/bendh/Desktop/data science/JN//filtered_customer_booking.
↪csv")
```

[ ]:
```

```