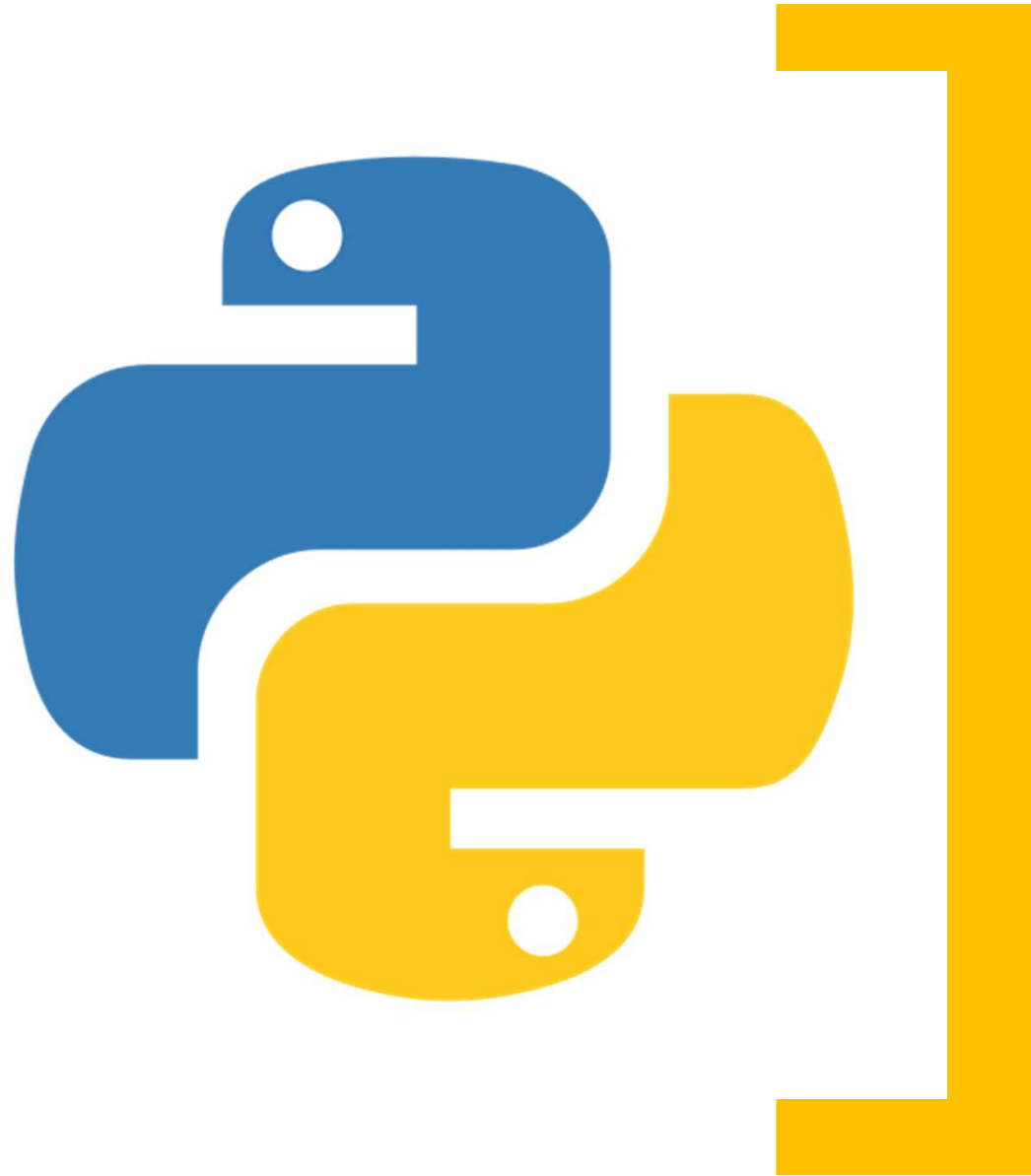




Python Project : Hotel Domain Analysis



1. Data Import and Data Exploration

```
df_bookings = pd.read_csv('datasets/fact_bookings.csv')
```

```
df_bookings.head()
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platfo
0	May012216558RT11	16558	27-04-22	1/5/2022	2/5/2022	-3.0	RT1	direct onl
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	oth
2	May012216558RT13	16558	28-04-22	1/5/2022	4/5/2022	2.0	RT1	logt
3	May012216558RT14	16558	28-04-22	1/5/2022	2/5/2022	-2.0	RT1	oth
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct onl

```
df_bookings.shape
```

(134590, 12)

```
df_bookings.room_category.unique()
```

Year	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	

2. Data Cleaning

```
df_bookings.describe()
```

	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
count	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
mean	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
std	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
min	16558.000000	-17.000000	1.000000	6.500000e+03	2600.000000
25%	17558.000000	1.000000	3.000000	9.900000e+03	7600.000000
50%	17564.000000	2.000000	4.000000	1.350000e+04	11700.000000
75%	18563.000000	2.000000	5.000000	1.800000e+04	15300.000000
max	19563.000000	6.000000	5.000000	2.856000e+07	45220.000000

(1) Clean invalid guests

```
df_bookings[df_bookings.no_guests<=0]
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category
0	16558000000	16558	2016-01-01	2016-01-01	2016-01-01	0	0

3. Data Transformation

Create occupancy percentage column

```
: df_agg_bookings.head(3)
```

```
:   property_id  check_in_date  room_category  successful_bookings  capacity
0         16559      1-May-22           RT1                25        30.0
1         19562      1-May-22           RT1                28        30.0
2         19563      1-May-22           RT1                23        30.0
```

```
: df_agg_bookings['occ_pct'] = df_agg_bookings.apply(lambda row: row['successful_bookings']/row['capacity'], axis=1)
```

```
: new_col = df_agg_bookings.apply(lambda row: row['successful_bookings']/row['capacity'], axis=1)
df_agg_bookings = df_agg_bookings.assign(occ_pct=new_col.values)
df_agg_bookings.head(3)
```

```
:   property_id  check_in_date  room_category  successful_bookings  capacity  occ_pct
0         16559      1-May-22           RT1                25        30.0  0.833333
1         19562      1-May-22           RT1                28        30.0  0.933333
2         19563      1-May-22           RT1                23        30.0  0.766667
```

4. Insights Generation

1. What is an average occupancy rate in each of the room categories?

```
df_agg_bookings.head(3)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct
0	16559	1-May-22	RT1	25	30.0	83.33
1	19562	1-May-22	RT1	28	30.0	93.33
2	19563	1-May-22	RT1	23	30.0	76.67

```
df_agg_bookings.groupby("room_category")["occ_pct"].mean()
```

```
room_category
RT1    58.224247
RT2    58.040278
RT3    58.028213
RT4    59.300461
Name: occ_pct, dtype: float64
```

```
df = pd.merge(df_agg_bookings, df_rooms, left_on="room_category", right_on="room_id")
df.head(4)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_id	room_class
--	-------------	---------------	---------------	---------------------	----------	---------	---------	------------

Output

```
df.groupby("room_class")["occ_pct"].mean()
```

room_class	
Elite	58.040278
Premium	58.028213
Presidential	59.300461
Standard	58.224247

Name: occ_pct, dtype: float64

2. Print average occupancy rate per city

Output

```
df_hotels.head(3)
```

	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi

```
df.groupby("city")["occ_pct"].mean()
```

```
city
Bangalore    56.594207
Delhi        61.606467
Hyderabad    58.144651
Mumbai       57.936305
Name: occ_pct, dtype: float64
```

```
df = pd.merge(df, df_hotels, on="property_id")
df.head(3)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name
0	16559	1-May-22	RT1	25	30.0	83.33	Standard	Atliq Exotica
1	16559	2-May-22	RT1	20	30.0	66.67	Standard	Atliq Exotica
2	16559	3-May-22	RT1	17	30.0	56.67	Standard	Atliq Exotica

3. When was the occupancy better? Weekday or Weekend?

```
|: df_date.head(3)
```

```
|:      date  mmm yy  week no  day_type
0  01-May-22   May 22    W 19  weekend
1  02-May-22   May 22    W 19  weekday
2  03-May-22   May 22    W 19  weekday
```

```
|: df = pd.merge(df, df_date, left_on="check_in_date", right_on="date")
df.head(3)
```

```
|:      property_id  check_in_date  room_category  successful_bookings  capacity  occ_pct  room_class  property_name
0          16559    10-May-22          RT1              18          30.0    60.00    Standard    Atliq Exotic
1          16559    10-May-22          RT2              25          41.0    60.98         Elite    Atliq Exotic
2          16559    10-May-22          RT3              20          32.0    62.50    Premium    Atliq Exotic
```

Output

```
df.groupby("day_type")["occ_pct"].mean().round(2)

day_type
weekday    50.90
weekend    72.39
Name: occ_pct, dtype: float64
```

4: In the month of June, what is the occupancy for different cities

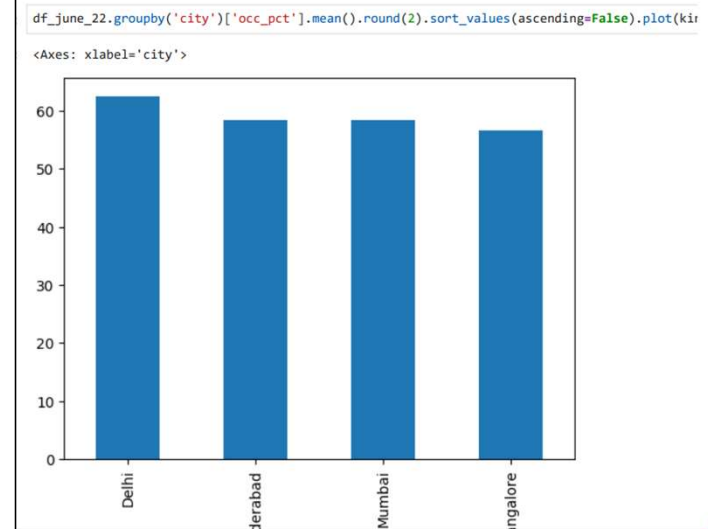
```
df_june_22 = df[df["mmm yy"]=="Jun 22"]
df_june_22.head(4)
```

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_n
2200	16559	10-Jun-22	RT1	20	30.0	66.67	Standard	Atliq Ex
2201	16559	10-Jun-22	RT2	26	41.0	63.41	Elite	Atliq Ex
2202	16559	10-Jun-22	RT3	20	32.0	62.50	Premium	Atliq Ex
2203	16559	10-Jun-22	RT4	11	18.0	61.11	Presidential	Atliq Ex

```
df_june_22.groupby('city')['occ_pct'].mean().round(2).sort_values(ascending=False)
```

```
city
Delhi      62.47
Hyderabad  58.46
Mumbai     58.38
Bangalore  56.58
Name: occ_pct, dtype: float64
```

Output



5: We got new data for the month of august. Append that to existing data

```
df_august = pd.read_csv("datasets/new_data_august.csv")
df_august.head(3)
```

```
property_id  property_name  category  city  room_category  room_class  check_in_date
0      16559    Atliq Exotica  Luxury  Mumbai           RT1      Standard    01-Aug-22
1      19562      Atliq Bay  Luxury  Bangalore           RT1      Standard    01-Aug-22
2      19563    Atliq Palace  Business  Bangalore           RT1      Standard    01-Aug-22
```

```
df_august.columns
```

```
Index(['property_id', 'property_name', 'category', 'city', 'room_category',
       'room_class', 'check_in_date', 'mmm yy', 'week no', 'day_type',
       'successful_bookings', 'capacity', 'occ_pct'],
      dtype='object')
```

```
df.columns
```

```
Index(['property_id', 'check_in_date', 'room_category', 'successful_bookings',
       'capacity', 'occ_pct', 'room_class', 'property_name', 'category',
       'city', 'date', 'mmm yy', 'week no', 'day_type'],
      dtype='object')
```

```
df_august.shape
```

```
(7, 13)
```

Output

	property_id	check_in_date	room_category	successful_bookings	capacity	occ_pct	room_class	property_name	category	city	date	mmm yy	week no	day_type
6494	16563	31-Jul-22	RT2	32	38.0	84.21	Elite	Atliq Palace	Business	Delhi	31-Jul-22	Jul 22	W 32	weeken
6495	16563	31-Jul-22	RT3	14	20.0	70.00	Premium	Atliq Palace	Business	Delhi	31-Jul-22	Jul 22	W 32	weeken
6496	16563	31-Jul-22	RT4	13	18.0	72.22	Presidential	Atliq Palace	Business	Delhi	31-Jul-22	Jul 22	W 32	weeken
6497	16559	01-Aug-22	RT1	30	30.0	NaN	Standard	Atliq Exotica	Luxury	Mumbai	Aug-22	Aug-22	W 32	weekeda
6498	19562	01-Aug-22	RT1	21	30.0	NaN	Standard	Atliq Bay	Luxury	Bangalore	Aug-22	Aug-22	W 32	weekeda
6499	19563	01-Aug-22	RT1	23	30.0	NaN	Standard	Atliq Palace	Business	Bangalore	Aug-22	Aug-22	W 32	weekeda
6500	19558	01-Aug-22	RT1	30	40.0	NaN	Standard	Atliq Grands	Luxury	Bangalore	Aug-22	Aug-22	W 32	weekeda
6501	19560	01-Aug-22	RT1	20	26.0	NaN	Standard	Atliq City	Business	Bangalore	Aug-22	Aug-22	W 32	weekeda
6502	17561	01-Aug-22	RT1	18	26.0	NaN	Standard	Atliq Blu	Luxury	Mumbai	Aug-22	Aug-22	W 32	weekeda
6503	17564	01-Aug-22	RT1	10	16.0	NaN	Standard	Atliq Seasons	Business	Mumbai	Aug-22	Aug-22	W 32	weekeda

6. Print revenue realized per city



```
df_bookings.head()
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0		
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others	4.0		
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others	NaN		
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0	RT1	logtrip	NaN		

◀

```
df_hotels.head(3)
```

	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi

```
df_bookings_all = pd.merge(df_bookings, df_hotels, on="property_id")
df_bookings_all.head(3)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	Mav012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	

Output

```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```

6. Print revenue realized per city

📄 ⬆ ⬇ ⬇ ⬇ ⬇

```
df_bookings.head()
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
1	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
4	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	
5	May012216558RT16	16558	1/5/2022	1/5/2022	3/5/2022	2.0	RT1	others			
6	May012216558RT17	16558	28-04-22	1/5/2022	6/5/2022	2.0	RT1	others			
7	May012216558RT18	16558	26-04-22	1/5/2022	3/5/2022	2.0	RT1	logtrip			

4

```
df_hotels.head(3)
```

	property_id	property_name	category	city
0	16558	Atliq Grands	Luxury	Delhi
1	16559	Atliq Exotica	Luxury	Mumbai
2	16560	Atliq City	Business	Delhi

```
df_bookings_all = pd.merge(df_bookings, df_hotels, on="property_id")
df_bookings_all.head(3)
```

	booking_id	property_id	booking_date	check_in_date	checkout_date	no_guests	room_category	booking_platform	ratings_given	booking_status	revenue_g
0	May012216558RT12	16558	30-04-22	1/5/2022	2/5/2022	2.0	RT1	others	NaN	Cancelled	
1	May012216558RT15	16558	27-04-22	1/5/2022	2/5/2022	4.0	RT1	direct online	5.0	Checked Out	

Output

```
city
Bangalore    420383550
Delhi        294404488
Hyderabad    325179310
Mumbai       668569251
Name: revenue_realized, dtype: int64
```