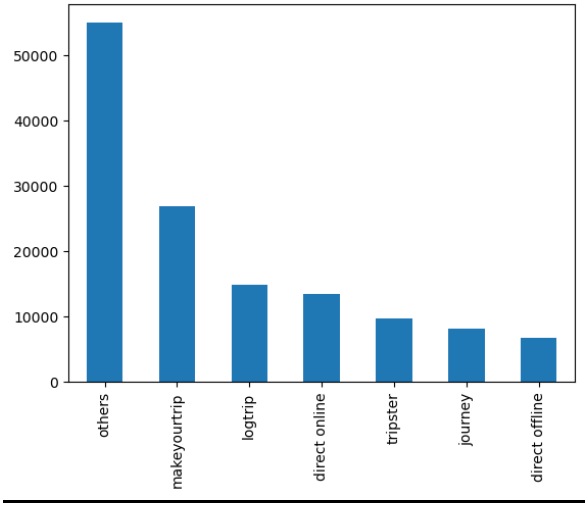
Hotel Data Analytics Using Pandas

This is a python project made about Hotel Data Analytics. This was done during my certification of Python from Codebasics.

Datasets used in the page include:

1. fact_bookings.csv
2. dim_date.csv
3. dim_hotels.csv
4. dim_rooms.csv
5. fact_aggregated_bookings.csv

- Comparing the number of bookings done using different platforms.



- Considering the statistical description of the **fact_bookings** data frame, it shows that the mean rating is 3.61 but minimum number of guests in the **no_guests** show -17 which indicates that Data cleaning is required for further analysis of the data.

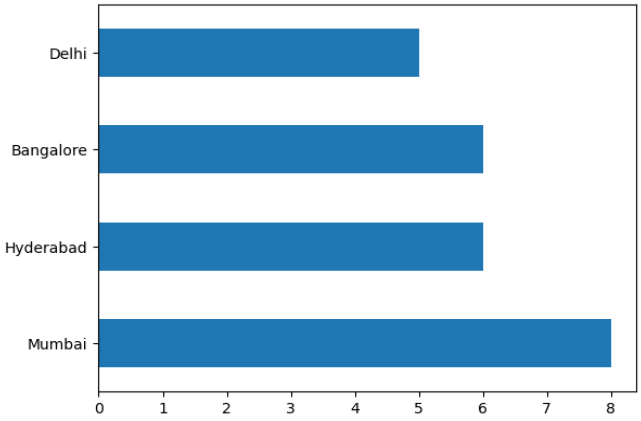
	property_id	no_guests	ratings_given	revenue_generated	revenue_realized
count	134590.000000	134587.000000	56683.000000	1.345900e+05	134590.000000
mean	18061.113493	2.036170	3.619004	1.537805e+04	12696.123256
std	1093.055847	1.034885	1.235009	9.303604e+04	6928.108124
min	16558.000000	-17.000000	1.000000	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

Exploratory Data Analysis

- Category of hotels across all the cities and number of hotels in each city.

```
Luxury      16
Business     9
Name: category, dtype: int64
```

```
Mumbai       8
Hyderabad    6
Bangalore    6
Delhi        5
Name: city, dtype: int64
```



- Total booking per property id:

```
property_id
16558    3153
16559    7338
16560    4693
16561    4418
16562    4820
16563    7211
17558    5053
17559    6142
17560    6013
17561    5183
17562    3424
17563    6337
17564    3982
18558    4475
18559    5256
18560    6638
18561    6458
18562    7333
18563    4737
19558    4400
19559    4729
19560    6079
19561    5736
19562    5812
19563    5413
Name: successful_bookings, dtype: int64
```

- Days in which Bookings are more than the capacity:

	property_id	check_in_date	room_category	successful_bookings	capacity
3	17558	1-May-22	RT1	30	19.0
12	16563	1-May-22	RT1	100	41.0
4136	19558	11-Jun-22	RT2	50	39.0
6209	19560	2-Jul-22	RT1	123	26.0
8522	19559	25-Jul-22	RT1	35	24.0
9194	18563	31-Jul-22	RT4	20	18.0

Data Cleaning

- The rows in which the number of guests were less than 0 were removed.
- As seen above in the statistical description of the **fact_bookings** dataset, thus removing the rows which have revenue realised more than 3 * Standard deviation away from the mean of the revenue realised.
- Exploring the **dim_rooms** Dataset the following conclusions were derived.

	room_id	room_class
0	RT1	Standard
1	RT2	Elite
2	RT3	Premium
3	RT4	Presidential

We will se just the Statistics of RT 4 rooms

We will se the **mean + 3 * standatd deviation** for just RT4 rooms and see is the upper limit a valid amount.

Here we are getting 50585.105 as the maximum price of 3 standard deviation for RT4 rooms

Now the rows having the **revenue_realized** more than 50585.105 will be considered as a outlier, but we can see that max when used describe is 45220.00 which is ok.

Thus no Data cleaning required for revenue_realized column.

Data Transformation

- Using the **successful_bookings** and **capacity** column of **fact_aggregated_bookings** to make a column by the name of Occupancy Percent, defined by the formula;

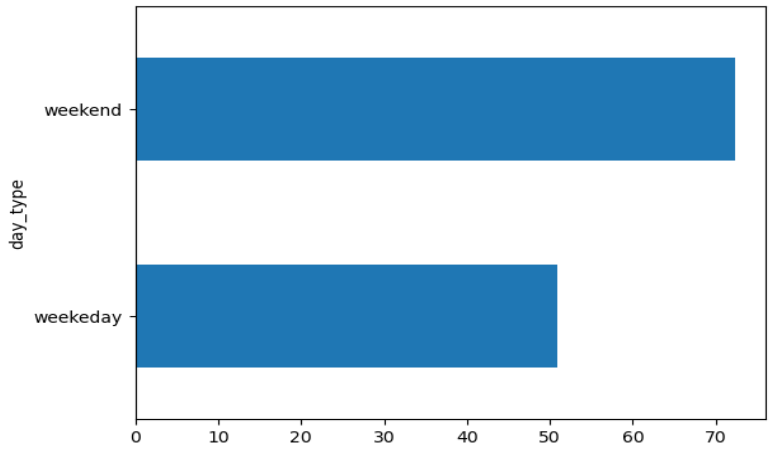
Occupancy percent = (successful bookings) * 100 / capacity

Insight Generation

- Average occupancy rate per room class.
    - Grouping by room category and taking the average of occupancy present rounded up to 2 positions.

```
room_class
Elite           58.01
Premium         58.03
Presidential    59.28
Standard        57.89
Name: Occupancy_percent, dtype: float64
```

- When are the occupancy higher, Weekends or Weekday?
    - Merging the **fact_aggregated_bookings** with **dim_date** to visualize it usinga bar plot.



- Maximum Occupancy rate of different city in June
    - Selecting the rows which have ‘June 22’ in their date and storing it in a new dataframe.
    - Grouping them by city and considering their Occupancy rate rounded upto 1 decimal place.

```
city
Delhi        62.5
Hyderabad    58.5
Mumbai       58.4
Bangalore    56.4
Name: Occupancy_percent, dtype: float64
```

- Average rating per city
    - Grouping by city and considering the average rating per city.

```
city
Bangalore    3.4
Delhi        3.8
Hyderabad    3.7
Mumbai       3.6
Name: ratings_given, dtype: float64
```

- Visualizing booking realized per booking platform using a pie chart.