This is a project to Analyse Top Indian Places to Visit.

Objective : To practice data clearning and sorting.

Dataset : https://www.kaggle.com/datasets/dhrubangtalukdar/top-indian-places-to-visit-indian-tourism

Tools : Python, Jupyter Notebook

Libraries : Numpy as np, Pandas as pd

In [200…
```
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
```

After we import the needed libraries, we read the CSV file. pd library is used because we want a dataframe, not a list or array. (r is used to ask python to ignore special characters in the path. The path is copied and paster. We are reading a csv file, hence, pd.read_csv(r"")

In [201…
```
dataframe= pd.read_csv(r"C:\Users\USER\Downloads\Top Indian Places to Visit.csv")
```

We check the basic properties, .shape, .head(), .tail() to make sure all data was loaded or not.

In [202…
```
dataframe.head()
```

Out[202…

	Unnamed: 0	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Go rev ra
0	0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	
1	1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	
2	2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	
3	3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	
4	4	Northern	Delhi	Delhi	Jantar Mantar	Observatory	1724	2.0	

In [203…
```
dataframe.tail()
```

Out[203…

	Unnamed: 0	Zone	State	City	Name	Type	Establishment Year	ne to i
320	320	Western	Gujarat	Gandhinagar	Akshardham	Temple	1992	
321	321	Central	Uttar Pradesh	Agra	Agra Fort	Fort	1565	
322	322	Central	Madhya Pradesh	Bhopal	Madhya Pradesh Tribal Museum	Museum	2013	
323	323	Northern	Rajasthan	Jaipur	City Palace	Palace	1727	
324	324	Northern	Rajasthan	Jaipur	Albert Hall Museum	Museum	1887	

Note: The .shape says about the number of rows and columns, including the header, but excluding the index numbers.

In [204…

```
dataframe.shape
```

Out[204…

```
(325, 16)
```

Next, we check the data types for each columns using .info()

In [205…

```
dataframe.info()
```

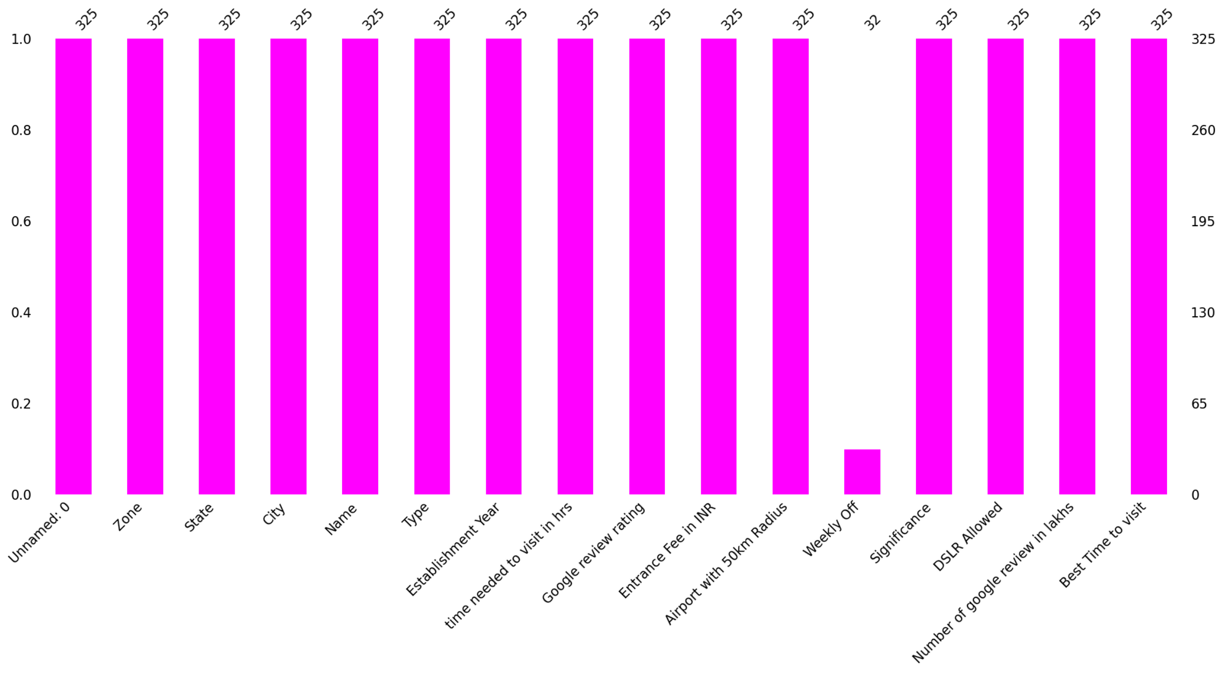
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 325 entries, 0 to 324
Data columns (total 16 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Unnamed: 0                       325 non-null    int64  
 1   Zone                             325 non-null    object 
 2   State                            325 non-null    object 
 3   City                             325 non-null    object 
 4   Name                             325 non-null    object 
 5   Type                             325 non-null    object 
 6   Establishment Year               325 non-null    object 
 7   time needed to visit in hrs      325 non-null    float64
 8   Google review rating             325 non-null    float64
 9   Entrance Fee in INR              325 non-null    int64  
 10  Airport with 50km Radius         325 non-null    object 
 11  Weekly Off                       32 non-null     object 
 12  Significance                     325 non-null    object 
 13  DSLR Allowed                     325 non-null    object 
 14  Number of google review in lakhs 325 non-null    float64
 15  Best Time to visit               325 non-null    object 
dtypes: float64(3), int64(2), object(11)
memory usage: 40.8+ KB
```

We see that some columns have null values. We import missingno library as msno to use this library to visualise the number of missing numbers (Nan) (null values) in each column. This helps us to decide if we can drop a column from analysis (if missing no is >90% of all values)

In [206…
```python
import missingno as msno
```

In [207…
```python
msno.bar(dataframe, color=(1,0,1))
```

Out[207…
```
<Axes: >
```



msno.matrix(dataframe, color=(0.0, 0, 1))

After Checking the Nan, we check for the quality of the data (duplicate, inconsistency). To do so, we perform 4 checks as below:

A. Determining Categories & Identifying Columns: If a value has limited number of unique values, they can be considered as categories. Here : Zone, State, City, Type, Airpot Within 50KM, significance, dslr allowed, best time to visit are good categories.

In [208…
```
dataframe.nunique()
```

Out[208…
```
Unnamed: 0                          325
Zone                                  6
State                                33
City                                214
Name                                321
Type                                 78
Establishment Year                  162
time needed to visit in hrs          11
Google review rating                 14
Entrance Fee in INR                  33
Airport with 50km Radius              2
Weekly Off                            5
Significance                         25
DSLR Allowed                          2
Number of google review in lakhs    108
Best Time to visit                    7
dtype: int64
```

B. Checking the unique values for each categories.

In [209…
```
sorted(dataframe['Zone'].unique())
```

Out[209…
```
['Central', 'Eastern', 'North Eastern', 'Northern', 'Southern', 'Western']
```

In [210…
```
sorted(dataframe['State'].unique())
```

Out[210…

```
['Andaman and Nicobar Islands',
 'Andhra Pradesh',
 'Arunachal Pradesh',
 'Assam',
 'Bihar',
 'Chhattisgarh',
 'Daman and Diu',
 'Delhi',
 'Goa',
 'Gujarat',
 'Haryana',
 'Himachal Pradesh',
 'Jammu and Kashmir',
 'Jharkhand',
 'Karnataka',
 'Kerala',
 'Ladakh',
 'Madhya Pradesh',
 'Maharashtra',
 'Maharastra',
 'Meghalaya',
 'Nagaland',
 'Odisha',
 'Puducherry',
 'Punjab',
 'Rajasthan',
 'Sikkim',
 'Tamil Nadu',
 'Telangana',
 'Tripura',
 'Uttar Pradesh',
 'Uttarakhand',
 'West Bengal']
```

In [211…

```
sorted(dataframe['City'].unique())
```

```
Out[211…  ['Agartala',
           'Agra',
           'Ahmedabad',
           'Ajanta',
           'Ajmer',
           'Alappuzha',
           'Alibaug',
           'Aligarh',
           'Allahabad',
           'Almora',
           'Amarkantak',
           'Amravati',
           'Amritsar',
           'Anantapur',
           'Anantnag',
           'Auli',
           'Aurangabad',
           'Auroville',
           'Ayodhya',
           'Badami',
           'Badrinath',
           'Balasore',
           'Bandhavgarh',
           'Bandipur',
           'Bangalore',
           'Baratang Island',
           'Barot',
           'Bastar',
           'Bekal',
           'Bengaluru',
           'Berhampur',
           'Bhimbetka',
           'Bhopal',
           'Bhubaneswar',
           'Bhuj',
           'Bijapur',
           'Bikaner',
           'Bir Billing',
           'Bodh Gaya',
           'Bolpur',
           'Chamba',
           'Chandigarh',
           'Chennai',
           'Cherrapunji',
           'Chidambaram',
           'Chikmagalur',
           'Chilika',
           'Chitrakoot',
           'Chittorgarh',
           'Chopta',
           'Coimbatore',
           'Cooch Behar',
           'Coorg',
           'Cuttack',
           'Dalhousie',
           'Darjeeling',
'Dehradun',
'Delhi',
'Deoghar',
'Digha',
'Diskit',
'Diu',
'Dras',
'Dumboor',
'Dwarka',
'Dzükou Valley',
'Fatehpur Sikri',
'Gandhinagar',
'Gangtok',
'Goa',
'Gokarna',
'Greater Noida',
'Guntur',
'Gurugram',
'Guwahati',
'Gwalior',
'Hajo',
'Halebidu',
'Hampi',
'Haridwar',
'Havelock Island',
'Hemis',
'Hooghly',
'Hyderabad',
'Indore',
'Jabalpur',
'Jaipur',
'Jaisalmer',
'Jalpaiguri',
'Jammu',
'Jhansi',
'Jim Corbett',
'Jodhpur',
'Joshimath',
'Junagadh',
'Kadapa',
'Kangra',
'Kanha',
'Kannur',
'Kanpur',
'Kanyakumari',
'Kargil',
'Kaziranga',
'Kedarnath',
'Kendujhar',
'Keonjhar',
'Kevadia',
'Khajuraho',
'Kinnaur',
'Kishtwar',
'Kochi',
'Kodaikanal',
'Kolhapur',
'Kolkata',
'Konark',
'Kovalam',
'Kozhikode',
'Kufri',
'Kullu',
'Kumarakom',
'Kurnool',
'Leh',
'Lonavala',
'Lucknow',
'Madurai',
'Mahabalipuram',
'Majuli',
'Manali',
'Manas',
'Mandi',
'Mandu',
'Mangalore',
'Manikaran',
'Matheran',
'Mathura',
'McLeod Ganj',
'Meerut',
'Mount Abu',
'Mumbai',
'Munnar',
'Murshidabad',
'Murudeshwar',
'Mussoorie',
'Mysore',
'Nagpur',
'Nainital',
'Namchi',
'Narkanda',
'Nashik',
'Neil Island',
'Nelliyampathy',
'New Delhi',
'Noida',
'Nubra Valley',
'Ooty',
'Orchha',
'Pachmarhi',
'Pahalgam',
'Palampur',
'Patna',
'Pelling',
'Porbandar',
'Port Blair',
'Puducherry',
'Pune',
'Puri',
'Purulia',
'Pushkar',
 'Puttaparthi',
 'Rajahmundry',
 'Rameswaram',
 'Ranchi',
 'Ranikhet',
 'Rann of Kutch',
 'Ratnagiri',
 'Ravangla',
 'Rishikesh',
 'Rourkela',
 'Sambalpur',
 'Sarnath',
 'Satara',
 'Sawai Madhopur',
 'Shimla',
 'Shirdi',
 'Shivamogga',
 'Shoja',
 'Siliguri',
 'Sivasagar',
 'Somnath',
 'Spiti Valley',
 'Srinagar',
 'Srisailam',
 'Sundarbans',
 'Tarkarli',
 'Tawang',
 'Thanjavur',
 'Thekkady',
 'Thiruvananthapuram',
 'Tirunelveli',
 'Udaipur',
 'Udhampur',
 'Ujjain',
 'Unakoti',
 'Uttarkashi',
 'Vadodara',
 'Varanasi',
 'Varkala',
 'Vijayawada',
 'Visakhapatnam',
 'Vizianagaram',
 'Vrindavan',
 'Wayanad',
 'Yercaud',
 'dalhousie']
```

In [212…

```
sorted(dataframe['Type'].unique())
```

```
Out[212…  ['Adventure Sport',
           'Amusement Park',
           'Aquarium',
           'Beach',
           'Bird Sanctuary',
           'Border Crossing',
           'Botanical Garden',
           'Bridge',
           'Cave',
           'Church',
           'Commercial Complex',
           'Confluence',
           'Cricket Ground',
           'Cultural',
           'Dam',
           'Entertainment',
           'Film Studio',
           'Fort',
           'Ghat',
           'Government Building',
           'Gravity Hill',
           'Gurudwara',
           'Hill',
           'Historical',
           'Island',
           'Lake',
           'Landmark',
           'Mall',
           'Market',
           'Mausoleum',
           'Memorial',
           'Monastery',
           'Monument',
           'Mosque',
           'Mountain Peak',
           'Museum',
           'National Park',
           'Natural Feature',
           'Observatory',
           'Orchard',
           'Palace',
           'Park',
           'Prehistoric Site',
           'Promenade',
           'Race Track',
           'Religious Complex',
           'Religious Shrine',
           'Religious Site',
           'River Island',
           'Rock Carvings',
           'Scenic Area',
           'Scenic Point',
           'Science',
           'Sculpture Garden',
           'Shrine',
           'Site',
           'Ski Resort',
           'Spiritual Center',
           'Stepwell',
           'Sunrise Point',
           'Suspension Bridge',
           'Tea Plantation',
           'Temple',
           'Temples',
           'Theme Park',
           'Tomb',
           'Tombs',
           'Township',
           'Trekking',
           'Urban Development Project',
           'Valley',
           'Viewpoint',
           'Village',
           'Vineyard',
           'War Memorial',
           'Waterfall',
           'Wildlife Sanctuary',
           'Zoo']
```

In [213…

```
sorted(dataframe['Airport with 50km Radius'].unique())
```

Out[213…

```
['No', 'Yes']
```

In [214…

```
sorted(dataframe['Significance'].unique())
```

Out[214…

```
['Adventure',
 'Agricultural',
 'Archaeological',
 'Architectural',
 'Artistic',
 'Botanical',
 'Cultural',
 'Educational',
 'Engineering Marvel',
 'Entertainment',
 'Environmental',
 'Food',
 'Historical',
 'Market',
 'Natural Wonder',
 'Nature',
 'Recreational',
 'Religious',
 'Scenic',
 'Scientific',
 'Shopping',
 'Spiritual',
 'Sports',
 'Trekking',
 'Wildlife']
```

In [215…
```
sorted(dataframe['DSLR Allowed'].unique())
```
Out[215…
```
['No', 'Yes']
```
In [216…
```
dataframe['Weekly Off'].unique()
```
Out[216…
```
array([nan, 'Monday', 'Sunday', 'Friday', 'Yes', 'Tuesday'], dtype=object)
```
In [217…
```
dataframe[dataframe['Weekly Off']=='Yes']
```
Out[217…

	Unnamed: 0	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Goog revie ratir
131	131	Western	Maharastra	Lonavala	Karla Caves	Cave	200	1.5	4

Here, we see that in Weekly Off, one entry is given as 'Yes' instead of the day. So we will replace the data with the correct one.

In [218…
```
dataframe.at[131,'Weekly Off']= np.nan
```
In [219…
```
dataframe['Weekly Off'].unique()
```
Out[219…
```
array([nan, 'Monday', 'Sunday', 'Friday', 'Tuesday'], dtype=object)
```
In [220…
```
sorted(dataframe['Best Time to visit'].unique())
```
Out[220…
```
['Afternoon', 'All', 'All ', 'Anytime', 'Evening', 'Morning', 'Night']
```
Here, in 'Best Time to visit', we see that the 'All' value is duplicate, caused due to a spacing error. So, we can ensure that the spacing errors are fixed using .str.strip() method from Pandas Library. So, we replace the existing column with the new column where the extra space is removed.

In [221…
```
dataframe['Best Time to visit'] = dataframe['Best Time to visit'].str.strip()
```
In [222…
```
sorted(dataframe['Best Time to visit'].unique())
```
Out[222…
```
['Afternoon', 'All', 'Anytime', 'Evening', 'Morning', 'Night']
```
Also, anytime and all are the same. So we combine these two categories. we will use .replace. So we replace the existing column with a new column where the two elements are the same. Instead of All i will add anytime

In [223…
```
dataframe['Best Time to Visit']= dataframe['Best Time to visit'].replace('All','Any
```

In [224…

```
sorted(dataframe['Best Time to Visit'].unique())
```

Out[224…

```
['Afternoon', 'Anytime', 'Evening', 'Morning', 'Night']
```

After Standardisation, we enter Analysis.