

Loading the data and packages

In [69]:

```
import pandas as pd
import numpy as np
#Loading the data
data= pd.read_csv('EnquiryTotal.csv')
```

List all column names

In [70]:

```
data.columns.tolist()
```

Out[70]:

```
['Enquiry.Date',
 'Enquiry.Time',
 'Enquiry.Day',
 'Allocated.Time',
 'Web.or.Phone',
 'Hotkey.',
 'Number.of.Conversation.RCD',
 'Number.of.Quote.Templates.Sent.to.client',
 'Holiday.Type',
 'Accommodation.type',
 'Departure.Airport',
 'DepartureDate',
 'Lead.Time..weeks.',
 'Destination.1',
 'Duration',
 'Adults',
 'Children',
 'Infants',
 'Transport.Type',
 'Answered.Questionnaire.',
 'Questionnaire.Notes.Completed',
 'Title',
 'Any.Enquiry.Comments.',
 'Booked.Status']
```

Rename column names

In [71]:

```
data.columns = ['Date',  
                'EnquiryTime',  
                'EnquiryDay',  
                'AllocatedTime',  
                'WeborPhone',  
                'Hotkey',  
                'NumberofConversationRCD',  
                'TemplatesSent',  
                'HolidayType',  
                'Accommodationtype',  
                'DepartureAirport',  
                'DepartureDate',  
                'Lead.Timeweeks.',  
                'Destination1',  
                'Duration',  
                'Adults',  
                'Children',  
                'Infants',  
                'TransportType',  
                'Answered.Questionnaire',  
                'NotesCompleted',  
                'Title',  
                'Any.Enquiry.Comments',  
                'BookedStatus']
```

View first 10 rows (Data.tail for last)

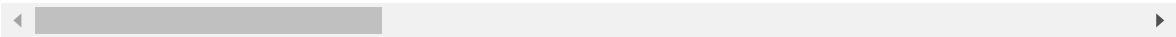
In [72]:

```
data.head(10)
```

Out[72]:

	Date	EnquiryTime	EnquiryDay	AllocatedTime	WeborPhone	Hotkey	NumberofConv
0	5/18/2018	9:00	Friday	3d 2h 46m	WEB	NaN	
1	2/18/2018	19:46	Sunday	1d 0h 20m	WEB	NaN	
2	1/5/2018	23:38	Friday	2d 22h 13m	WEB	NaN	
3	1/30/2019	20:21	Wednesday	0d 12h 56m	WEB	Hot Key	
4	1/26/2019	12:42	Saturday	0d 0h 3m	WEB	Hot Key	
5	1/22/2019	12:19	Tuesday	0d 0h 2m	WEB	Hot Key	
6	1/12/2019	20:13	Saturday	0d 16h 10m	WEB	Hot Key	
7	1/2/2019	17:09	Wednesday	0d 0h 2m	WEB	Hot Key	
8	12/27/2018	15:39	Thursday	0d 0h 2m	WEB	Hot Key	
9	12/26/2018	21:08	Wednesday	1d 21h 44m	WEB	Hot Key	

10 rows × 24 columns



Check data type

In [73]:

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178347 entries, 0 to 178346
Data columns (total 24 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                  178347 non-null  object
1   EnquiryTime                          178347 non-null  object
2   EnquiryDay                           178347 non-null  object
3   AllocatedTime                        165064 non-null  object
4   WeborPhone                           178347 non-null  object
5   Hotkey                               49251 non-null   object
6   NumberofConversationRCD              178347 non-null  int64
7   TemplatesSent                        178347 non-null  int64
8   HolidayType                          178347 non-null  object
9   Accommodationtype                    165740 non-null  object
10  DepartureAirport                      178199 non-null  object
11  DepartureDate                         178347 non-null  object
12  Lead.Timeweeks.                       178347 non-null  int64
13  Destination1                          178322 non-null  object
14  Duration                              178322 non-null  float64
15  Adults                                178347 non-null  int64
16  Children                              178347 non-null  int64
17  Infants                              178347 non-null  int64
18  TransportType                         172508 non-null  object
19  Answered.Questionnaire                178347 non-null  object
20  NotesCompleted                        178347 non-null  object
21  Title                                 178347 non-null  object
22  Any.Enquiry.Comments                  178347 non-null  object
23  BookedStatus                          178347 non-null  object
dtypes: float64(1), int64(6), object(17)
memory usage: 32.7+ MB
```

Change data type

In [74]:

data.Adults=data.Adults.astype(float)

Select all except 1 field

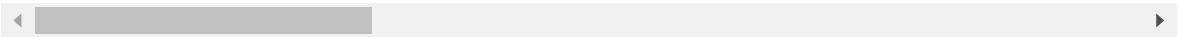
In [75]:

```
data.loc[:, data.columns != 'Infants']
```

Out[75]:

	Date	EnquiryTime	EnquiryDay	AllocatedTime	WeborPhone	Hotkey	Numberof
0	5/18/2018	9:00	Friday	3d 2h 46m	WEB	NaN	
1	2/18/2018	19:46	Sunday	1d 0h 20m	WEB	NaN	
2	1/5/2018	23:38	Friday	2d 22h 13m	WEB	NaN	
3	1/30/2019	20:21	Wednesday	0d 12h 56m	WEB	Hot Key	
4	1/26/2019	12:42	Saturday	0d 0h 3m	WEB	Hot Key	
...
178342	8/16/2017	16:12:04	Wednesday	0 m	PHONE	NaN	
178343	8/15/2017	12:11:38	Tuesday	2d 21h 9m	PHONE	NaN	
178344	8/15/2017	11:32:24	Tuesday	6d 6h 46m	PHONE	NaN	
178345	8/15/2017	11:21:27	Tuesday	6d 23h 13m	PHONE	NaN	
178346	8/15/2017	11:51:10	Tuesday	3d 23h 44m	WEB	NaN	

178347 rows × 23 columns



Drop multiple columns

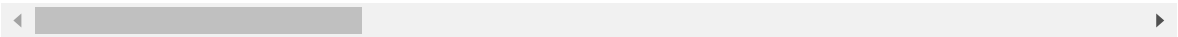
In [76]:

```
data.loc[:, ~data.columns.isin(['Infants', 'Children'])]
```

Out[76]:

	Date	EnquiryTime	EnquiryDay	AllocatedTime	WeborPhone	Hotkey	Numberof
0	5/18/2018	9:00	Friday	3d 2h 46m	WEB	NaN	
1	2/18/2018	19:46	Sunday	1d 0h 20m	WEB	NaN	
2	1/5/2018	23:38	Friday	2d 22h 13m	WEB	NaN	
3	1/30/2019	20:21	Wednesday	0d 12h 56m	WEB	Hot Key	
4	1/26/2019	12:42	Saturday	0d 0h 3m	WEB	Hot Key	
...
178342	8/16/2017	16:12:04	Wednesday	0 m	PHONE	NaN	
178343	8/15/2017	12:11:38	Tuesday	2d 21h 9m	PHONE	NaN	
178344	8/15/2017	11:32:24	Tuesday	6d 6h 46m	PHONE	NaN	
178345	8/15/2017	11:21:27	Tuesday	6d 23h 13m	PHONE	NaN	
178346	8/15/2017	11:51:10	Tuesday	3d 23h 44m	WEB	NaN	

178347 rows × 22 columns



select multiple columns

In [77]:

```
data.iloc[:,[0,1,4]]
```

Out[77]:

	Date	EnquiryTime	WeborPhone
0	5/18/2018	9:00	WEB
1	2/18/2018	19:46	WEB
2	1/5/2018	23:38	WEB
3	1/30/2019	20:21	WEB
4	1/26/2019	12:42	WEB
...
178342	8/16/2017	16:12:04	PHONE
178343	8/15/2017	12:11:38	PHONE
178344	8/15/2017	11:32:24	PHONE
178345	8/15/2017	11:21:27	PHONE
178346	8/15/2017	11:51:10	WEB

178347 rows × 3 columns

Select columns 0 to 2

In [78]:

```
data.iloc[:, 0:2]
```

Out[78]:

	Date	EnquiryTime
0	5/18/2018	9:00
1	2/18/2018	19:46
2	1/5/2018	23:38
3	1/30/2019	20:21
4	1/26/2019	12:42
...
178342	8/16/2017	16:12:04
178343	8/15/2017	12:11:38
178344	8/15/2017	11:32:24
178345	8/15/2017	11:21:27
178346	8/15/2017	11:51:10

178347 rows × 2 columns

selecting multiple required columns

In [79]:

```
data.iloc[:, np.r_[0:3,15:19,22,23]]
```

Out[79]:

	Date	EnquiryTime	EnquiryDay	Adults	Children	Infants	TransportType	Any.E
0	5/18/2018	9:00	Friday	2.0	0	0	Return transfers	
1	2/18/2018	19:46	Sunday	3.0	2	0	Return transfers	
2	1/5/2018	23:38	Friday	4.0	0	0	Return transfers	
3	1/30/2019	20:21	Wednesday	4.0	0	0	Return transfers	
4	1/26/2019	12:42	Saturday	3.0	3	0	Return transfers	
...
178342	8/16/2017	16:12:04	Wednesday	2.0	0	0	NaN	
178343	8/15/2017	12:11:38	Tuesday	2.0	0	0	NaN	
178344	8/15/2017	11:32:24	Tuesday	2.0	0	0	NaN	
178345	8/15/2017	11:21:27	Tuesday	2.0	0	0	NaN	
178346	8/15/2017	11:51:10	Tuesday	1.0	0	0	NaN	

178347 rows × 9 columns



Select columns which have enquiry in the name

In [80]:

```
data.filter(like='Enquiry')
```

Out[80]:

	EnquiryTime	EnquiryDay	Any.Enquiry.Comments
0	9:00	Friday	NO
1	19:46	Sunday	NO
2	23:38	Friday	NO
3	20:21	Wednesday	NO
4	12:42	Saturday	YES
...
178342	16:12:04	Wednesday	NO
178343	12:11:38	Tuesday	NO
178344	11:32:24	Tuesday	NO
178345	11:21:27	Tuesday	NO
178346	11:51:10	Tuesday	NO

178347 rows × 3 columns

Select columns based on multiple patterns

In [81]:

```
data.filter(regex='Type|Enquiry')
```

Out[81]:

	EnquiryTime	EnquiryDay	HolidayType	TransportType	Any.Enquiry.Comments
0	9:00	Friday	Cruise + Stay	Return transfers	NO
1	19:46	Sunday	Accommodation Only	Return transfers	NO
2	23:38	Friday	Package Holiday	Return transfers	NO
3	20:21	Wednesday	Multi Centre	Return transfers	NO
4	12:42	Saturday	Package Holiday	Return transfers	YES
...
178342	16:12:04	Wednesday	Cruise + Stay	NaN	NO
178343	12:11:38	Tuesday	Package Holiday	NaN	NO
178344	11:32:24	Tuesday	Package Holiday	NaN	NO
178345	11:21:27	Tuesday	Package Holiday	NaN	NO
178346	11:51:10	Tuesday	Accommodation Only	NaN	NO

178347 rows × 5 columns

Remove columns

In [82]:

```
data = data.drop(["EnquiryTime", "AllocatedTime"],axis=1)
```

Check data type

In [83]:

```
print(data.dtypes)
```

Date	object
EnquiryDay	object
WeborPhone	object
Hotkey	object
NumberofConversationRCD	int64
TemplatesSent	int64
HolidayType	object
Accommodationtype	object
DepartureAirport	object
DepartureDate	object
Lead.Timeweeks.	int64
Destination1	object
Duration	float64
Adults	float64
Children	int64
Infants	int64
TransportType	object
Answered.Questionnaire	object
NotesCompleted	object
Title	object
Any.Enquiry.Comments	object
BookedStatus	object
dtype:	object

Change date to date format and extract year and month

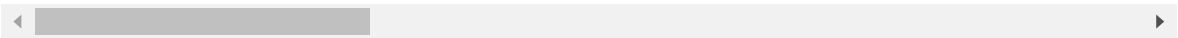
In [84]:

```
data['Date'] = pd.to_datetime(data['Date'])
data['year'], data['month'] = data['Date'].dt.year, data['Date'].dt.month
# Remove date
data = data.drop(['Date'],axis=1)
data
```

Out[84]:

	EnquiryDay	WeborPhone	Hotkey	NumberofConversationRCD	TemplatesSent	Hol
0	Friday	WEB	NaN	0	1	Crui
1	Sunday	WEB	NaN	0	1	Accom
2	Friday	WEB	NaN	0	1	
3	Wednesday	WEB	Hot Key	1	1	Mt
4	Saturday	WEB	Hot Key	3	3	
...	
178342	Wednesday	PHONE	NaN	6	3	Crui
178343	Tuesday	PHONE	NaN	0	0	
178344	Tuesday	PHONE	NaN	0	1	
178345	Tuesday	PHONE	NaN	0	1	
178346	Tuesday	WEB	NaN	0	0	Accom

178347 rows × 23 columns



Understand numerical variables

In [85]:

```
data.describe()
```

Out[85]:

	NumberOfConversationRCD	TemplatesSent	Lead.Timeweeks.	Duration	A
count	178347.000000	178347.000000	178347.000000	178322.000000	178347.00
mean	1.259489	1.341295	50.094445	13.106448	3.43
std	2.960769	1.147345	27.735302	3.630358	2.03
min	0.000000	0.000000	-44.000000	0.000000	1.00
25%	0.000000	1.000000	29.000000	11.000000	2.00
50%	0.000000	1.000000	48.000000	14.000000	3.00
75%	1.000000	1.000000	68.000000	14.000000	4.00
max	356.000000	29.000000	216.000000	63.000000	58.00

Understand categorical variables (use include='all' to show both numerical and categorical)

In [86]:

```
data.describe(include=[object])
```

Out[86]:

	EnquiryDay	WeborPhone	Hotkey	HolidayType	Accommodationtype	DepartureAirpor
count	178347	178347	49251	178347	165740	178199
unique	7	2	1	14	6	3
top	Sunday	WEB	Hot Key	Package Holiday	Hotel	Mancheste
freq	30865	154890	49251	134230	91055	5420

Filtering numerical data

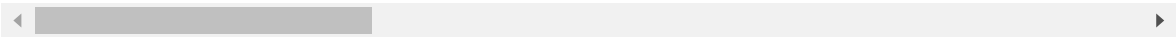
In [87]:

```
df_filtered=data.query('Infants < 50 & NumberofConversationRCD > 0')
df_filtered
```

Out[87]:

	EnquiryDay	WeborPhone	Hotkey	NumberofConversationRCD	TemplatesSent	Holida
3	Wednesday	WEB	Hot Key	1	1	Multi
4	Saturday	WEB	Hot Key	3	3	Pε t
5	Tuesday	WEB	Hot Key	3	2	Pε t
6	Saturday	WEB	Hot Key	3	2	Pε t
7	Wednesday	WEB	Hot Key	1	1	Multi
...	
178333	Monday	PHONE	NaN	3	2	Pε t
178335	Sunday	PHONE	NaN	1	2	Pε t
178338	Wednesday	PHONE	NaN	2	1	Pε t
178341	Wednesday	PHONE	NaN	10	1	Multi
178342	Wednesday	PHONE	NaN	6	3	C

71246 rows × 23 columns



Filtering categorical data

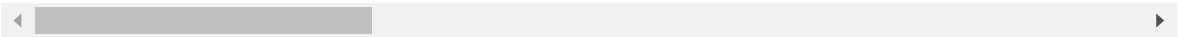
In [88]:

```
df_filtered=data.query('EnquiryDay == "Friday" and Hotkey == "Hot Key"')
df_filtered
```

Out[88]:

	EnquiryDay	WeborPhone	Hotkey	NumberofConversationRCD	TemplatesSent	Holida
12	Friday	WEB	Hot Key	6	5	Multi
19	Friday	WEB	Hot Key	1	1	Pε t
56	Friday	WEB	Hot Key	2	1	Pε t
59	Friday	WEB	Hot Key	1	1	C
103	Friday	WEB	Hot Key	0	1	Pε t
...	
166869	Friday	WEB	Hot Key	8	2	Fl
173174	Friday	WEB	Hot Key	2	2	Pε t
175261	Friday	WEB	Hot Key	0	1	Pε t
175266	Friday	WEB	Hot Key	1	2	Pε t
175271	Friday	WEB	Hot Key	2	1	Pε t

6181 rows × 23 columns



Replacing NAN values

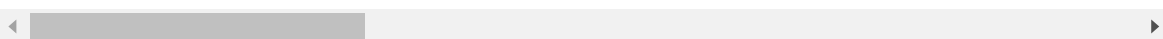
In [89]:

```
data["Hotkey"].fillna("No",inplace=True)
data
```

Out[89]:

	EnquiryDay	WeborPhone	Hotkey	NumberofConversationRCD	TemplatesSent	Hol
0	Friday	WEB	No	0	1	Crui
1	Sunday	WEB	No	0	1	Accom
2	Friday	WEB	No	0	1	
3	Wednesday	WEB	Hot Key	1	1	Mt
4	Saturday	WEB	Hot Key	3	3	
...	
178342	Wednesday	PHONE	No	6	3	Crui
178343	Tuesday	PHONE	No	0	0	
178344	Tuesday	PHONE	No	0	1	
178345	Tuesday	PHONE	No	0	1	
178346	Tuesday	WEB	No	0	0	Accom

178347 rows × 23 columns



checking frequency of variables

In [90]:

```
pd.value_counts(data["WeborPhone"])
```

Out[90]:

```
WEB      154890
PHONE    23457
Name: WeborPhone, dtype: int64
```

Replace specific values in a dataframe column

```
data.Hotkey = data.Hotkey.replace({"NaN":"No", "Hot Key":"Yes"}) data
```

Replace numerical variables with median or mode (replace 0 with the median)

In [91]:

```
data.Children=data.Children.replace(0,data.Children.median())
data
```

Out[91]:

	EnquiryDay	WeborPhone	Hotkey	NumberofConversationRCD	TemplatesSent	Hol
0	Friday	WEB	No	0	1	Crui
1	Sunday	WEB	No	0	1	Accom
2	Friday	WEB	No	0	1	
3	Wednesday	WEB	Hot Key	1	1	Mt
4	Saturday	WEB	Hot Key	3	3	
...	
178342	Wednesday	PHONE	No	6	3	Crui
178343	Tuesday	PHONE	No	0	0	
178344	Tuesday	PHONE	No	0	1	
178345	Tuesday	PHONE	No	0	1	
178346	Tuesday	WEB	No	0	0	Accom

178347 rows × 23 columns

Merge two datasets

In [92]:

```
# data frame 1
d1 = {'Customer_id':pd.Series([1,2,3,4,5,6]),
      'Product':pd.Series(['Oven','Oven','Oven','Television','Television','Television'])}
df1 = pd.DataFrame(d1)
df1
```

Out[92]:

	Customer_id	Product
0	1	Oven
1	2	Oven
2	3	Oven
3	4	Television
4	5	Television
5	6	Television

In [93]:

```
# data frame 2
d2 = {'Customer_id':pd.Series([2,4,6,7,8]),
      'State':pd.Series(['California','California','Texas','New York','Indiana'])}
df2 = pd.DataFrame(d2)
df2
```

Out[93]:

	Customer_id	State
0	2	California
1	4	California
2	6	Texas
3	7	New York
4	8	Indiana

In [94]:

```
#inner join in python pandas (inner,outer,left,right)

inner_join_df= pd.merge(df1, df2, on='Customer_id', how='inner')
inner_join_df
```

Out[94]:

	Customer_id	Product	State
0	2	Oven	California
1	4	Television	California
2	6	Television	Texas