Pandas libraries can be used for Data Cleansing, Data Analysis and Data Transformation. In this project, we are working on hotel-booking-data.

1. Read hotel-booking-data.txt file using pandas read_table method.
2. Use drop_na() function to drop the Nan values. Don't use inplace=True initially as we need the data present in one column.
3. Use of set_option('display.max.rows', size)
4. Find the number of bookings made by each company?
5. Use isna() fn in dataframe,'where' function in numpy, and 'bfill()' from df to clean the dataset.

In [1]:
```
import numpy as np
import pandas as pd
df=pd.read_table(r"C:\Users\vysal\OneDrive\Documents\Datasets for Pandas\hotel-book
df
```

Out[1]:

	Date	Company	Person Name	Room number
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0
3	Hotels	NaN	NaN	NaN
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0
...	...	...	...	...
190	Cleartrip	NaN	NaN	NaN
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0
194	Hotels	NaN	NaN	NaN

195 rows × 4 columns

In [2]:
```
# From the above df we can see that, sample data has few text in between as well as

df.dropna()

# By doing this it actually removed the text data that was present in those rows. L
#further analysis, in that case this is not a good solution  I have not used inplac
```

Out[2]:

	Date	Company	Person Name	Room number
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0
5	1-Jan-2022	Eadel	Broderic Handscombe	3015.0
...	...	...	...	...
186	1-Jan-2022	Tagpad	Stephani Lafee	1015.0
187	1-Jan-2022	Meevee	Victoria Lavery	7002.0
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0

134 rows × 4 columns

In [3]:
```
pd.set_option('display.max.rows', 134)
```

In [4]:
```
# No of bookings made by each company
df['Company'].value_counts()
```

Out[4]:

Company	
Leexo	3
Fivechat	3
Quinu	3
Jabbersphere	2
Twitterlist	2
Ntags	2
Realcube	2
Meevee	2
Tagtune	2
Realfire	2
Skinix	2
Brightdog	2
Aimbo	2
Jayo	2
Eadel	2
Rhynyx	2
Ainyx	2
Jatri	1
Tambee	1
Divavu	1
Buzzbean	1
Camido	1
Buzzster	1
Fiveclub	1
Yadel	1
Meetz	1
Yozio	1
Quire	1
Flashpoint	1
Vipe	1
Leenti	1
Edgeclub	1
Yabox	1
Dazzlesphere	1
Oloo	1
Miboo	1
Avamba	1
Vinder	1
Zoomdog	1
Innojam	1
Tagpad	1
Skiba	1
Twitterwire	1
Jaxworks	1
Vitz	1
Voomm	1
Youfeed	1
Voolia	1
Thoughtbridge	1
Centidel	1
Zooxo	1
Demivee	1
Rhynoodle	1
Devbug	1
Wordpedia	1

Bubbletube	1
Brightbean	1
Photobean	1
Yodo	1
Avamm	1
Divanoodle	1
Feedbug	1
Skinder	1
Oyope	1
Feedspan	1
Kwinu	1
Tagchat	1
Dynabox	1
Muxo	1
Centizu	1
Kimia	1
Dynazzy	1
Babblestorm	1
Aimbu	1
Mita	1
Skynoodle	1
Feedmix	1
Topicblab	1
Flipopia	1
Geba	1
Shuffletag	1
Skinte	1
Browsezoom	1
Tazz	1
Riffpedia	1
Innotype	1
Oozz	1
Rhyzio	1
Skalith	1
Buzzdog	1
Feedfish	1
Zava	1
Riffpath	1
Lajo	1
Fatz	1
Tagopia	1
Meembee	1
Rhycero	1
BlogXS	1
Dabshots	1
Eare	1
Devcast	1
Devshare	1
Topiczoom	1
Skibox	1
Mycat	1
Kazio	1
Dabjam	1
Youopia	1
Topiclounge	1
Shufflebeat	1

```
Skyvu            1
Youspan          1
Twitterworks     1
Name: count, dtype: int64
```

In [5]:
```
df
```

Out[5]:

	Date	Company	Person Name	Room number
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0
3	Hotels	NaN	NaN	NaN
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0
...	...	...	...	...
190	Cleartrip	NaN	NaN	NaN
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0
194	Hotels	NaN	NaN	NaN

195 rows × 4 columns

In [6]:
```
# Let's manipulate the dataframe by moving the text data (hotel, cleartrip etc) to
# Use isna() fn ( which gives a true or false value ) to check if one of those colu
#single column dataframe.
df['Room number']. isna()
```

Out[6]:
```
0      False
1      False
2      False
3       True
4      False
       ...
190     True
191    False
192    False
193    False
194     True
Name: Room number, Length: 195, dtype: bool
```

numpy module has a 'where' function, which acts like an if condition. Based on the condition (maskedvalue), we provide the true and false values. It is kind of an if condition but then imagine as if it goes inside a loop.

df['text_value']=np.where(maskedvalue,df['Date'],np.NaN) => If masked value is 'true' then get the value from df['Date'] column , else insert NaN value to the newly created text_value column. df['text_value'] => new column created df['Date'] => the first column in the df

In [7]:
```
maskedvalue= df['Room number']. isna()
df['text_value']= np.where(maskedvalue,df['Date'],np.NaN)
df
```

Out[7]:

	Date	Company	Person Name	Room number	text_value
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0	NaN
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0	NaN
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0	NaN
3	Hotels	NaN	NaN	NaN	Hotels
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0	NaN
...	...	...	...	...	...
190	Cleartrip	NaN	NaN	NaN	Cleartrip
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0	NaN
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0	NaN
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0	NaN
194	Hotels	NaN	NaN	NaN	Hotels

195 rows × 5 columns

In [8]:
```
# Now let's use the fill function in the dataframe to fill it up the values. bfill
# The code df['text_value'].bfill() performs a backward fill on the column text_val
#found below in the column.
#df['text_value'].fillna('bfill',inplace=True) # deprecated syntax
df['text_value'] =df['text_value'].bfill()
df
```

Out[8]:

	Date	Company	Person Name	Room number	text_value
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0	Hotels
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0	Hotels
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0	Hotels
3	Hotels	NaN	NaN	NaN	Hotels
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0	Booking
...	...	...	...	...	...
190	Cleartrip	NaN	NaN	NaN	Cleartrip
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0	Hotels
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0	Hotels
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0	Hotels
194	Hotels	NaN	NaN	NaN	Hotels

195 rows × 5 columns

In [9]:
```
# Drop the remaining NaN values in the other columns
df.dropna(inplace=True)
```

In [10]:
```
# Now we have a clean dataset to perform data anaylsis
df
```

Out[10]:

	Date	Company	Person Name	Room number	text_value
0	1-Jan-2022	Avamba	Anatole Ridehalgh	4008.0	Hotels
1	1-Jan-2022	Fatz	Aldrich McKevin	2002.0	Hotels
2	1-Jan-2022	Leexo	Stanley Hadrill	4012.0	Hotels
4	1-Jan-2022	Rhyzio	Lyndell Tice	1006.0	Booking
5	1-Jan-2022	Eadel	Broderic Handscombe	3015.0	Booking
6	1-Jan-2022	Oozz	Deina Harwin	2005.0	Booking
7	1-Jan-2022	Innotype	Benyamin Crocetti	2001.0	Booking
10	1-Jan-2022	Jayo	Tate Manntschke	3014.0	Booking
13	1-Jan-2022	Riffpedia	Elianore Vigar	3002.0	Cleartrip
14	1-Jan-2022	Tazz	Alonso Mundee	4006.0	Cleartrip
16	1-Jan-2022	Browsezoom	Ysabel Lordon	6003.0	Hotels
17	1-Jan-2022	Skinte	Raff Verecker	3012.0	Hotels
19	1-Jan-2022	Twitterlist	Andrea Humpatch	4005.0	Hotels
21	1-Jan-2022	Shuffletag	Cammy Curle	2016.0	Expedia
22	1-Jan-2022	Geba	Howey Oseman	6001.0	Expedia
25	1-Jan-2022	Flipopia	Prince Coppenhall	2002.0	Booking
26	1-Jan-2022	Topicblab	Noni Tarbett	1012.0	Booking
27	1-Jan-2022	Feedmix	Bondon Tuny	3007.0	Booking
28	1-Jan-2022	Skynoodle	Andros Cathcart	1008.0	Booking
30	1-Jan-2022	Mita	Aubert Racher	4004.0	Travel Agent 007
31	1-Jan-2022	Skinix	Curcio Lewis	3008.0	Travel Agent 007
32	1-Jan-2022	Aimbu	Weider Brookz	1001.0	Travel Agent 007
33	1-Jan-2022	Babblestorm	Alric Reeder	7001.0	Travel Agent 007
34	1-Jan-2022	Jayo	Melany Brimblecombe	3010.0	Travel Agent 007
35	1-Jan-2022	Quinu	Art Giannotti	3013.0	Travel Agent 007
37	1-Jan-2022	Dynazzy	Ericha MacBain	4002.0	Expedia
38	1-Jan-2022	Tagtune	Scarlett Berthel	2001.0	Expedia
40	1-Jan-2022	Kimia	Lottie Barnsdall	2014.0	Expedia
41	1-Jan-2022	Centizu	Faydra Hulland	5006.0	Expedia
42	1-Jan-2022	Muxo	Felice Kramer	4002.0	Expedia

	Date	Company	Person Name	Room number	text_value
43	1-Jan-2022	Dynabox	Deane Gemson	5005.0	Expedia
44	1-Jan-2022	Skalith	Willie Norree	3002.0	Expedia
45	1-Jan-2022	Tagchat	Almira Bartolomeotti	2004.0	Expedia
47	1-Jan-2022	Buzzdog	Sharona Ferreres	4012.0	Travel Agent 007
48	1-Jan-2022	Zava	Ilaire Gaynes	1013.0	Travel Agent 007
51	1-Jan-2022	Ntags	Corny Madill	3002.0	Booking
53	1-Jan-2022	Youspan	Yasmin Snelgar	3012.0	Travel Agent 007
54	1-Jan-2022	Skyvu	Sherlock Hyland	6001.0	Travel Agent 007
55	1-Jan-2022	Quinu	Kalindi Shaughnessy	6003.0	Travel Agent 007
56	1-Jan-2022	Twitterlist	Lauretta Stoke	4004.0	Travel Agent 007
58	1-Jan-2022	Brightdog	Lucie Jewiss	2009.0	Hotels
59	1-Jan-2022	Shufflebeat	Dorita Boulger	6005.0	Hotels
60	1-Jan-2022	Fivechat	Orsola Cowdery	4001.0	Hotels
61	1-Jan-2022	Topiclounge	Lilah Attryde	1005.0	Hotels
62	1-Jan-2022	Youopia	Alley Pyer	2002.0	Hotels
64	1-Jan-2022	Dabjam	Pearline Toolan	4010.0	Expedia
65	1-Jan-2022	Kazio	Alvan Hardwick	4005.0	Expedia
67	1-Jan-2022	Mycat	Gipsy Bellison	5006.0	Expedia
69	1-Jan-2022	Fivechat	Kinna Linsley	2001.0	Expedia
70	1-Jan-2022	Meevee	Rodrique Brockbank	2013.0	Expedia
71	1-Jan-2022	Skibox	Jonathan Ewebank	3007.0	Expedia
72	1-Jan-2022	Topiczoom	Ladonna Castagna	3012.0	Expedia
74	1-Jan-2022	Realfire	Jules Evill	4010.0	Hotels
75	1-Jan-2022	Devshare	Nehemiah Huke	2013.0	Hotels
76	1-Jan-2022	Devcast	Jacques Stiffkins	1013.0	Hotels
77	1-Jan-2022	Eare	Fields Dovinson	4010.0	Hotels
78	1-Jan-2022	Dabshots	Gertie Dominguez	1011.0	Hotels
80	1-Jan-2022	BlogXS	Dal McGaughay	1004.0	Booking
83	1-Jan-2022	Rhycero	Merill Kleinhandler	1014.0	Booking
84	1-Jan-2022	Meembee	Fredrick Hurtic	2008.0	Booking

	Date	Company	Person Name	Room number	text_value
85	1-Jan-2022	Ntags	Angeline Goodbairn	1008.0	Booking
86	1-Jan-2022	Tagopia	Steven Casacchia	2001.0	Booking
87	1-Jan-2022	Feedspan	Abel Do	3016.0	Booking
88	1-Jan-2022	Lajo	Margareta Crathern	6001.0	Booking
90	1-Jan-2022	Ainyx	Etty Boland	4007.0	Hotels
92	1-Jan-2022	Riffpath	Sara Emlin	2008.0	Expedia
93	1-Jan-2022	Rhynyx	Camellia O'Rudden	1004.0	Expedia
94	1-Jan-2022	Feedfish	Margarette Blanchard	2016.0	Expedia
95	1-Jan-2022	Kwinu	Talbot Martinho	4010.0	Expedia
97	1-Jan-2022	Skinder	Bobbie Derx	1012.0	Expedia
98	1-Jan-2022	Jabbersphere	Merry Lawler	1008.0	Expedia
99	1-Jan-2022	Tambee	Reggie Chaldecott	3006.0	Expedia
101	1-Jan-2022	Yadel	Adda Dive	5002.0	Expedia
102	1-Jan-2022	Flashpoint	Virgina Congram	1015.0	Expedia
104	1-Jan-2022	Vipe	Emmet Schmidt	4006.0	Booking
106	1-Jan-2022	Edgeclub	Jeffrey Stelfax	1007.0	Expedia
108	1-Jan-2022	Realfire	Oby Reddington	1011.0	Hotels
109	1-Jan-2022	Leexo	Leslie Crisp	5003.0	Hotels
111	1-Jan-2022	Jatri	Harmon Hurren	4008.0	Hotels
113	1-Jan-2022	Yozio	Chickie Altimas	4001.0	Hotels
117	1-Jan-2022	Oloo	Curtis Oxley	2005.0	Expedia
118	1-Jan-2022	Eadel	Dukey Hansod	7001.0	Expedia
119	1-Jan-2022	Dazzlesphere	Dunn Davydzenko	4011.0	Expedia
120	1-Jan-2022	Yabox	Matteo Lovelady	4005.0	Expedia
122	1-Jan-2022	Rhynyx	Junie Parrott	4007.0	Hotels
123	1-Jan-2022	Miboo	Pierce Meese	5007.0	Hotels
124	1-Jan-2022	Leenti	Marven Burbudge	3010.0	Hotels
125	1-Jan-2022	Quire	Leland Somers	3007.0	Hotels
126	1-Jan-2022	Meetz	Celia Concklin	1016.0	Hotels
127	1-Jan-2022	Realcube	Terrie Barme	2011.0	Hotels

	Date	Company	Person Name	Room number	text_value
131	1-Jan-2022	Fiveclub	Akim Chang	4004.0	Cleartrip
132	1-Jan-2022	Buzzster	Harwell Vance	2011.0	Cleartrip
134	1-Jan-2022	Camido	Steven Britnell	1013.0	Cleartrip
135	1-Jan-2022	Buzzbean	Moyra Ferrini	1008.0	Cleartrip
137	1-Jan-2022	Tagtune	Pammi Powley	3001.0	Expedia
138	1-Jan-2022	Divavu	Cozmo Rosenblum	1009.0	Expedia
139	1-Jan-2022	Vinder	Lurette Vaissiere	1004.0	Expedia
140	1-Jan-2022	Quinu	Brandtr Groven	1005.0	Expedia
141	1-Jan-2022	Aimbo	Percy Robertz	2009.0	Expedia
142	1-Jan-2022	Demivee	Walsh Yarranton	1004.0	Expedia
144	1-Jan-2022	Ainyx	Georg Jelley	2011.0	Expedia
145	1-Jan-2022	Oyope	Kim Caldecourt	1012.0	Expedia
146	1-Jan-2022	Feedbug	Dari Weetch	2004.0	Expedia
147	1-Jan-2022	Divanoodle	Reta Yearn	3015.0	Expedia
149	1-Jan-2022	Avamm	Kakalina Farryan	1009.0	Booking
150	1-Jan-2022	Yodo	Nanci Slyman	2009.0	Booking
151	1-Jan-2022	Photobean	Marty Jerome	6001.0	Booking
153	1-Jan-2022	Realcube	Violetta Hannond	2015.0	Hotels
154	1-Jan-2022	Brightbean	Hagan McRobert	1008.0	Hotels
157	1-Jan-2022	Bubbletube	Annabel Tuxell	4010.0	Cleartrip
158	1-Jan-2022	Wordpedia	Estella Jirick	2005.0	Cleartrip
160	1-Jan-2022	Leexo	Zebulen Gillfillan	5004.0	Expedia
163	1-Jan-2022	Devbug	Benji Quadrio	1016.0	Hotels
164	1-Jan-2022	Jabbersphere	Jarrett Salazar	3011.0	Hotels
165	1-Jan-2022	Rhynoodle	Delora Cescoti	3010.0	Hotels
166	1-Jan-2022	Zooxo	Licha Attyeo	7002.0	Hotels
167	1-Jan-2022	Zoomdog	Charlene Pickard	1009.0	Hotels
169	1-Jan-2022	Skinix	Julianne Ghelardi	5004.0	Cleartrip
170	1-Jan-2022	Centidel	Heda Burcombe	2008.0	Cleartrip
171	1-Jan-2022	Thoughtbridge	Vivianna Syvret	5005.0	Cleartrip

	Date	Company	Person Name	Room number	text_value
173	1-Jan-2022	Voolia	Sandor Sagg	2002.0	Booking
174	1-Jan-2022	Youfeed	Marguerite Bodell	3014.0	Booking
176	1-Jan-2022	Voomm	Rudyard Stallibrass	1014.0	Booking
179	1-Jan-2022	Vitz	Pepillo Arkcoll	4009.0	Hotels
180	1-Jan-2022	Aimbo	Ashil Conichie	1014.0	Hotels
182	1-Jan-2022	Jaxworks	Norry Satch	3015.0	Expedia
183	1-Jan-2022	Twitterwire	Kelwin Gouldthorpe	2009.0	Expedia
184	1-Jan-2022	Skiba	George Downgate	7001.0	Expedia
185	1-Jan-2022	Brightdog	Bertrando Redman	3014.0	Expedia
186	1-Jan-2022	Tagpad	Stephani Lafee	1015.0	Expedia
187	1-Jan-2022	Meevee	Victoria Lavery	7002.0	Expedia
191	1-Jan-2022	Fivechat	Corabella Saye	4008.0	Hotels
192	1-Jan-2022	Innojam	Leandra Potapczuk	5002.0	Hotels
193	1-Jan-2022	Twitterworks	Valentia Ledson	1010.0	Hotels