In [37]:
```
import pandas as pd
import numpy as np
```

In [38]:
```
df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN',
                   'londON_StockhOlm','Budapest_PaRis', 'Brussels_londOn'],
                   'FlightNumber': [10045, np.nan, 10065, np.nan, 10085],
                   'RecentDelays': [[23, 47], [], [24, 43, 87], [13], [67, 32]],
                   'Airline': ['KLM(!)', '<Air France> (12)', '(British Airways. )',
                   '12. Air France', '"Swiss Air"']})
```

In [39]:
```
df
```

Out[39]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	<Air France> (12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	NaN	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

1. Some values in the the FlightNumber column are missing. These numbers are meant to increase by 10 with each row so 10055 and 10075 need to be put in place. Fill in these missing numbers and make the column an integer column (instead of a float column).

In [40]:
```
df['FlightNumber'] = np.where(df['FlightNumber'].isnull(),(df['FlightNumber'].fillna(method='ffill')+10),df["FlightNumber"]).astype(int)
df
```

Out[40]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045	[23, 47]	KLM(!)
1	MAdrid_miLAN	10055	[]	<Air France> (12)
2	londON_StockhOlm	10065	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	10075	[13]	12. Air France
4	Brussels_londOn	10085	[67, 32]	"Swiss Air"

2. The From*To column would be better as two separate columns! Split each string on the underscore delimiter* to give a new temporary DataFrame with the correct values. Assign the correct column names to this temporary DataFrame.

In [41]:
```
temp_data=pd.DataFrame(df["From_To"].str.split("_",1).to_list(),columns=["From","To"])
temp_data
```

Out[41]:

	From	To
0	LoNDon	paris
1	MAdrid	miLAN
2	londON	StockhOlm
3	Budapest	PaRis
4	Brussels	londOn

3. Notice how the capitalisation of the city names is all mixed up in this temporary DataFrame. Standardise the strings so that only the first letter is uppercase (e.g. "londON" should become "London".)

In [42]:
```
temp_data["From"]=temp_data["From"].str.capitalize()
temp_data["To"]=temp_data["To"].str.capitalize()
temp_data
```

Out[42]:

	From	To
0	London	Paris
1	Madrid	Milan
2	London	Stockholm
3	Budapest	Paris
4	Brussels	London

4. Delete the From_To column from df and attach the temporary DataFrame from the previous questions.

In [43]:
```
df.drop(["From_To"],inplace=True, axis=1)
df
```

Out[43]:

	FlightNumber	RecentDelays	Airline
0	10045	[23, 47]	KLM(!)
1	10055	[]	<Air France> (12)
2	10065	[24, 43, 87]	(British Airways.)
3	10075	[13]	12. Air France
4	10085	[67, 32]	"Swiss Air"

In [44]:
```
New=pd.concat([temp_data,df],axis=1)
New
```

Out[44]:

	From	To	FlightNumber	RecentDelays	Airline
0	London	Paris	10045	[23, 47]	KLM(!)
1	Madrid	Milan	10055	[]	<Air France> (12)
2	London	Stockholm	10065	[24, 43, 87]	(British Airways.)
3	Budapest	Paris	10075	[13]	12. Air France
4	Brussels	London	10085	[67, 32]	"Swiss Air"

5. In the RecentDelays column, the values have been entered into the DataFrame as a list. We would like each first value in its own column, each second value in its own column, and so on. If there isn't an Nth value, the value should be NaN. Expand the Series of lists into a DataFrame named delays, rename the columns delay_1, delay_2, etc. and replace the unwanted RecentDelays column in df with delays.

In [32]:
```
delays=pd.DataFrame(df["RecentDelays"].to_list(),columns=['delay_1','delay_2','delay_3'])
delays
```

Out[32]:

	delay_1	delay_2	delay_3
0	23.0	47.0	NaN
1	NaN	NaN	NaN
2	24.0	43.0	87.0
3	13.0	NaN	NaN
4	67.0	32.0	NaN

In [33]:
```
New.drop(['RecentDelays'],inplace=True,axis=1)
New
```

Out[33]:

	From	To	FlightNumber	Airline
0	London	Paris	10045	KLM(!)
1	Madrid	Milan	10055	<Air France> (12)
2	London	Stockholm	10065	(British Airways.)
3	Budapest	Paris	10075	12. Air France
4	Brussels	London	10085	"Swiss Air"

In [34]:
```
New.insert(loc=3,column='delay_1',value=delays['delay_1'])
New.insert(loc=4,column='delay_2',value=delays['delay_2'])
New.insert(loc=5,column='delay_3',value=delays['delay_3'])
```

In [35]:
```
New
```

Out[35]:

	From	To	FlightNumber	delay_1	delay_2	delay_3	Airline
0	London	Paris	10045	23.0	47.0	NaN	KLM(!)
1	Madrid	Milan	10055	NaN	NaN	NaN	<Air France> (12)
2	London	Stockholm	10065	24.0	43.0	87.0	(British Airways.)
3	Budapest	Paris	10075	13.0	NaN	NaN	12. Air France
4	Brussels	London	10085	67.0	32.0	NaN	"Swiss Air"

In [ ]:
```
```