Data Science Masters :Assignment 11

It happens all the time: someone gives you data containing malformed strings, Python, lists and missing data. How do you tidy it up so you can get on with the analysis?
Take this monstrosity as the DataFrame to use in the following puzzles:
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"']})

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 [274]:

```python
# Solution

# importing libraries
import pandas as pd
import numpy as np

df = pd.DataFrame({
    'From_To': ['LoNDon_paris', 'MAdrid_miLAN', 'londON_StockhOlm','Budapest_PaRis', 'Bruss
    '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', '"Sw
    })

fill_values = df['FlightNumber'].ffill()
fill_values.loc[df['FlightNumber'].isnull()]=fill_values.astype(int)+10
df['FlightNumber'] = fill_values.astype(int)
df
```

Out[274]:

	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 [275]:

```
# Solution
fromList = [];
toList = [];
for val in df['From_To']:
    splitList = val.split('_')
    fromList.append(splitList[0])
    toList.append(splitList[1])
df_temp =  df
df_temp["From"] = fromList
df_temp["To"] = toList
df_temp = df_temp.drop(['From_To'], axis=1)
df_temp
```

Out[275]:

	FlightNumber	RecentDelays	Airline	From	To
0	10045	[23, 47]	KLM(!)	LoNDon	paris
1	10055	[]	<Air France> (12)	MAdrid	miLAN
2	10065	[24, 43, 87]	(British Airways.)	londON	StockhOlm
3	10075	[13]	12. Air France	Budapest	PaRis
4	10085	[67, 32]	"Swiss Air"	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 [276]:

```
# Solution
df_temp['From']=df_temp['From'].str.capitalize()
df_temp['To']=df_temp['To'].str.capitalize()
df_temp
```

Out[276]:

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

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

In [277]:

```python
# Solution
df_temp['From_To'] = df['From_To']
print("df_temp ->\n")
print(df_temp)
df = df.drop(['From_To'],axis=1)
print("\ndf->\n")
print(df)
```

df_temp ->

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

          From_To
0     LoNDon_paris
1     MAdrid_miLAN
2  londON_StockhOlm
3    Budapest_PaRis
4    Brussels_londOn
```

df->

```
   FlightNumber  RecentDelays             Airline      From         To
0         10045      [23, 47]              KLM(!)    LoNDon      paris
1         10055            []     <Air France> (12)   MAdrid      miLAN
2         10065  [24, 43, 87]  (British Airways. )    londON  StockhOlm
3         10075          [13]       12. Air France  Budapest      PaRis
4         10085      [67, 32]           "Swiss Air"  Brussels     londOn
```

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 [278]:

```python
# Solution

recentDelays = df['RecentDelays']
delayList=[]
for l in recentDelays:
    while(len(l)<=2):
        l.append("NaN")
    delayList.append(l)
delays = pd.DataFrame(delayList,columns=['delay_1','delay_2','delay_3'])
for col in delays.columns:
    df[col] = delays[col]
df = df.drop(['RecentDelays'],axis=1)
df
```

Out[278]:

	FlightNumber	Airline	From	To	delay_1	delay_2	delay_3
0	10045	KLM(!)	LoNDon	paris	23	47	NaN
1	10055	<Air France> (12)	MAdrid	miLAN	NaN	NaN	NaN
2	10065	(British Airways.)	londON	StockhOlm	24	43	87
3	10075	12. Air France	Budapest	PaRis	13	NaN	NaN
4	10085	"Swiss Air"	Brussels	londOn	67	32	NaN