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(!)', ' (12)', '(British Airways.)', '12. Air France', '"Swiss Air"']})

In [66]:
```
# Import of libraries

import pandas as pd
import numpy as np
```

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

In [68]:
```
 df.head(10)
```

Out[68]:

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

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 [69]:
```
initialFlightNumber = 100045

df["FlightNumber"] = df[["FlightNumber"]].apply(lambda value: initialFlightNumber
In [70]: df.head(10)
```

```
Out[70]:
                Airline  FlightNumber           From_To  RecentDelays
0               KLM(!)        100045       LoNDon_paris      [23, 47]
1     <Air France> (12)       100055        MAdrid_miLAN            []
2  (British Airways. )        100065   londON_StockhOlm  [24, 43, 87]
3        12. Air France       100075       Budapest_PaRis          [13]
4           "Swiss Air"       100085     Brussels_londOn      [67, 32]
```

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 [71]: df_from_to = pd.DataFrame()
         df_from_to = pd.DataFrame(df.From_To.str.split('_', expand=True).values, columns=
```

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 [72]: df_from_to["From"] = df_from_to.From.str.capitalize()
         df_from_to["To"] = df_from_to.To.str.capitalize()
```

```
In [73]: df_from_to
```

```
Out[73]:
       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 [74]:

```
df = df.drop("From_To", axis=1)
df_new = pd.concat([df_from_to, df], axis = 1)
df_new
```

Out[74]:

	From	To	Airline	FlightNumber	RecentDelays
0	London	Paris	KLM(!)	100045	[23, 47]
1	Madrid	Milan	<Air France> (12)	100055	[]
2	London	Stockholm	(British Airways.)	100065	[24, 43, 87]
3	Budapest	Paris	12. Air France	100075	[13]
4	Brussels	London	"Swiss Air"	100085	[67, 32]

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 owncolumn, 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.

In [75]:

```
df_RecentDelays = df_new['RecentDelays'].apply(pd.Series)

# Integrate temp columns back into original Dataframe (while naming column)
for col in df_RecentDelays:
    df_new["Delays_%d" % (col+1)] = df_RecentDelays[col]
```

In [76]:

```
#6 Replace the unwanted RecentDelays column in df with delays.
df_new = df_new.drop("RecentDelays", axis=1)
df_new
```

Out[76]:

	From	To	Airline	FlightNumber	Delays_1	Delays_2	Delays_3
0	London	Paris	KLM(!)	100045	23.0	47.0	NaN
1	Madrid	Milan	<Air France> (12)	100055	NaN	NaN	NaN
2	London	Stockholm	(British Airways.)	100065	24.0	43.0	87.0
3	Budapest	Paris	12. Air France	100075	13.0	NaN	NaN
4	Brussels	London	"Swiss Air"	100085	67.0	32.0	NaN

In [ ]: