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 [1]:

```
# Import the necessary libraries

import pandas as pd
import numpy as np
```

In [2]:

```
# Set the dataset
df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN', 'londON_StockhOlm', 'Budapest_PaRis', 'Brussel
s_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', '"Swis
s Air"']})
```

In [3]:

```
df.head(10)
```

Out[3]:

	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"

In [4]:

```
# 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.

initialFlightNumber = 100045

df["FlightNumber"] = df[["FlightNumber"]].apply(lambda value: initialFlightNumber + df.index *10)
```

In [5]:

```
df.head(10)
```

Out[5]:

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

In [6]:

```
# Fill in these missing numbers and make the column an integer column (instead of a float column).
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
From_To         5 non-null object
FlightNumber    5 non-null int64
RecentDelays    5 non-null object
Airline         5 non-null object
dtypes: int64(1), object(3)
memory usage: 240.0+ bytes
```

In [7]:

```
# 2. The From_To column would be better as two separate columns! Split each string on the underscore delimite
r _ to give a new temporary DataFrame with the correct values.
# Assign the correct column names to this temporary DataFrame.

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

In [8]:

```
#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".)

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

In [9]:

```
df_from_to
```

Out[9]:

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

In [10]:

```
#4. Delete the From_To column from df and attach the temporary DataFrame from the previous questions.
df = df.drop("From_To", axis=1)
df_new = pd.concat([df_from_to, df], axis = 1)
df_new
```

Out[10]:

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

In [11]:

```
# 5. In the RecentDelays column, the values have been entered into the DataFrame as a list. We would like eac
h 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.

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 [12]:

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

Out[12]:

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