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).
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.
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".)
4. Delete the From_To column from df and attach the temporary DataFrame from the previous questions.
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 [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
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"']
    }
)

#Displaying Orignal(Given) DataFrame
df
```

Out[2]:

	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 [3]:
```
#Converting FlightNumber column to integer and filling the missing column to 0
df.loc[:,'FlightNumber'] = df.loc[:,'FlightNumber'].fillna(0).astype(int)

#Iterating FlightNumber column and adding missing column in incremental order(+10 from previous column)
df.loc[:,'FlightNumber'] = [df.loc[i-1, 'FlightNumber'] +10 if df.loc[i, 'FlightNumber']==0 else df.loc[i, 'FlightNumber']
                      for i in range(0, len(df.FlightNumber))]

#Displaying DataFrame after 1st question updates
df
```

Out[3]:

	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 [4]:
```
#Creating a new temporary DataFrame
df_fromto = pd.DataFrame()

#Adding 'From_To' column from orignal DataFrame to new temporary DataFrame
df_fromto['From_To'] = df['From_To']
```

```
#Creating correct column and spliting values from 'From_To' column
df_fromto['From'] = df_fromto['From_To'].str.split('_').apply(lambda x: x[0])
df_fromto['To'] = df_fromto['From_To'].str.split('_').apply(lambda x: x[1])

#Deleting 'From_To' column from temporary DataFrame
df_fromto.drop(columns='From_To', inplace=True)

#Displaying DataFrame after 2nd question updates
df_fromto
```

Out[4]:

	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 [5]:
```
#Converting 'From' and 'To' column first letter in upper case and others in lower case
df_fromto.loc[:,:] = df_fromto.loc[:,:].applymap(lambda x: x.title())

#Displaying DataFrame after 3rd question updates
df_fromto
```

Out[5]:

	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 [6]:
```
#Droping 'From_To' column from DateFrame
df.drop(columns='From_To', inplace=True)

#Adding 'df_fromto' DataFrame in df DataFrame
df = pd.concat([df,df_fromto], axis=1)

#Displaying DataFrame after 4th question updates
df
```

Out[6]:

	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.

5.1 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.

5.2 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.

5.1 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

In [7]:
```
#Creating variable to store maximun length of lists in 'RecentDelay' column
max_value = df.RecentDelays.apply(len).max()

#For loop for adding NaN if there is no value in list
for item in df.RecentDelays:
    if len(item)!=max_value:
        diff = max_value - len(item)
        for i in range(diff):
            item.append(np.nan)

#Displaying DataFrame after 5.1 question updates
df
```

Out[7]:

	FlightNumber	RecentDelays	Airline	From	To
0	10045	[23, 47, nan]	KLM(!)	London	Paris
1	10055	[nan, nan, nan]	<Air France> (12)	Madrid	Milan

2	10065	[24, 43, 87]	(British Airways.)	London	Stockholm
3	10075	[13, nan, nan]	12. Air France	Budapest	Paris
4	10085	[67, 32, nan]	"Swiss Air"	Brussels	London

5.2 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 [8]:
```
#Creating a temporary DataFrame
df_delays = pd.DataFrame()

#For loop for adding Delay column in DataFrame
for i in range(max_value):
    df_delays['Delay_'+str(i+1)] = [df.RecentDelays[item][i] for item in range(df.RecentDelays.size)]

#Deleting 'RecentDelays' from df DataFrame
df.drop(columns='RecentDelays', inplace=True)

#Adding 'df_delay' DataFrame to 'df' DataFrame
df = pd.concat([df,df_delays], axis=1)

#Displaying DataFrame after 5.2 question updates
df
```

Out[8]:

	FlightNumber	Airline	From	To	Delay_1	Delay_2	Delay_3
0	10045	KLM(!)	London	Paris	23.0	47.0	NaN
1	10055	<Air France> (12)	Madrid	Milan	NaN	NaN	NaN
2	10065	(British Airways.)	London	Stockholm	24.0	43.0	87.0
3	10075	12. Air France	Budapest	Paris	13.0	NaN	NaN
4	10085	"Swiss Air"	Brussels	London	67.0	32.0	NaN

6. From Airline column correct the Airline name by removing the punctuations and numbers

In [9]:
```
#Removing punctuations and number from the 'Airline' column to clear and get the corrected 'Airline' column
df.loc[:, 'Airline'] = df.loc[:, 'Airline'].str.replace('[\W\d]',' ').str.strip()

#Displaying final DataFrame
df
```

Out[9]:

	FlightNumber	Airline	From	To	Delay_1	Delay_2	Delay_3
0	10045	KLM	London	Paris	23.0	47.0	NaN
1	10055	Air France	Madrid	Milan	NaN	NaN	NaN
2	10065	British Airways	London	Stockholm	24.0	43.0	87.0
3	10075	Air France	Budapest	Paris	13.0	NaN	NaN
4	10085	Swiss Air	Brussels	London	67.0	32.0	NaN