Problem Statement

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"']})

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 [1]:
```
import pandas as pd
import numpy as np

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"']})

df
```

Out[1]:

	Airline	FlightNumber	From_To	RecentDelays
0	KLM(!)	10045.0	LoNDon_paris	[23, 47]
1	(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 [2]:
```
i=0
for fn in df['FlightNumber']:
    if np.isnan(fn) :
        df['FlightNumber'][i] = (df['FlightNumber'][i-1]+df['FlightNumber'][i+1])
    print(df['FlightNumber'][i])
    i = i +1
```

```
10045.0
10055.0
10065.0
10075.0
10085.0

C:\Users\prashant_gupta1\AppData\Local\Continuum\anaconda3\lib\site-packages\ip
ykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stab
le/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy)
  after removing the cwd from sys.path.
```

In [3]:
```
df['FlightNumber']=df.FlightNumber.astype('int64')
df
```

Out[3]:

	Airline	FlightNumber	From_To	RecentDelays
0	KLM(!)	10045	LoNDon_paris	[23, 47]
1	(12)	10055	MAdrid_miLAN	[]
2	(British Airways.)	10065	londON_StockhOlm	[24, 43, 87]
3	12. Air France	10075	Budapest_PaRis	[13]
4	"Swiss Air"	10085	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 [4]:
```
df = df.join(df['From_To'].str.split('_', expand=True).add_prefix('From'))
df. rename(columns={'From0': 'From'}, inplace=True)
df. rename(columns={'From1': 'To'}, inplace=True)
df
```

Out[4]:

	Airline	FlightNumber	From_To	RecentDelays	From	To
0	KLM(!)	10045	LoNDon_paris	[23, 47]	LoNDon	paris
1	(12)	10055	MAdrid_miLAN	[]	MAdrid	miLAN
2	(British Airways.)	10065	londON_StockhOlm	[24, 43, 87]	londON	StockhOlm
3	12. Air France	10075	Budapest_PaRis	[13]	Budapest	PaRis
4	"Swiss Air"	10085	Brussels_londOn	[67, 32]	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]:
```
df['From'] =df.From.str.title()
df['To'] =df.To.str.title()
#df(df['From'], inplace=True).str.title
df
```

Out[5]:

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

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

previous questions.

In [6]:
```
df.drop('From_To',inplace=True, axis=1)
df
```

Out[6]:

	Airline	FlightNumber	RecentDelays	From	To
0	KLM(!)	10045	[23, 47]	London	Paris
1	(12)	10055	[]	Madrid	Milan
2	(British Airways.)	10065	[24, 43, 87]	London	Stockholm
3	12. Air France	10075	[13]	Budapest	Paris
4	"Swiss Air"	10085	[67, 32]	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 [7]:
```
# Converting it into series
delays = df['RecentDelays'].apply(pd.Series)
# Creating the format of delay as delay_1 etc
delays.columns = ['delay_{}'.format(n) for n in range(1, len(delays.columns)+1)]
# replacing the RecentDelays with delays
df = df.drop('RecentDelays', axis=1).join(delays)
df
```

Out[7]:

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