DATA CLEANING - Assignment

By Prakash Ghosh ¶

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 [295]:
```
import pandas as pd
import numpy as np
```

In [296]:
```
# Define the DataFrme
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"']})

print('View Original DataFrame:')
df
```

```
View Original DataFrame:
```

Out[296]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	(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 [297]:
```
# Fill Missising values FlightNumber (10 increment by previous value)

# Create a list with the FlightNumber values from Data frame and if it is nan 10 increment previous value
lst_FlightNumber=list()
for num in df['FlightNumber']:
    if np.isnan(num):
        num=prev_num+10
    lst_FlightNumber.append(num)

# Update the Datarame FlightNumber column by the list
df['FlightNumber'] = lst_FlightNumber
print('\nView DataFrame after Filling Missising values in FlightNumber:')
df
```

```
View DataFrame after Filling Missising values in FlightNumber:
```

Out[297]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	10095.0	[]	(12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	10095.0	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [298]:
```
# Make the column an integer column (instead of a float column)
df['FlightNumber'] = df['FlightNumber'].astype(int)
print('\nView DataFrame after converting FlightNumber into int:')
df
```

```
View DataFrame after converting FlightNumber into int:
```

Out[298]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045	[23, 47]	KLM(!)
1	MAdrid_miLAN	10095	[]	(12)
2	londON_StockhOlm	10065	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	10095	[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 [299]:
```
# From_To column name to be split into From and To Column
# Value From_To column to be splited by _

# Create new Column From taking the value before _ of the From_To column using List Comprehension
df['From'] = [x.split('_', 1)[0] for x in df['From_To'].values]

# Create new Column To taking the value after _ of the From_To column using List Comprehension
df['To'] = [x.split('_', 1)[1] for x in df['From_To'].values]

print('\nView DataFrame after spliting From_To column:')
df
```

```
View DataFrame after spliting From_To column:
```

Out[299]:

	From_To	FlightNumber	RecentDelays	Airline	From	To
0	LoNDon_paris	10045	[23, 47]	KLM(!)	LoNDon	paris
1	MAdrid_miLAN	10095	[]	(12)	MAdrid	miLAN
2	londON_StockhOlm	10065	[24, 43, 87]	(British Airways.)	londON	StockhOlm
3	Budapest_PaRis	10095	[13]	12. Air France	Budapest	PaRis
4	Brussels_londOn	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 [300]:
```
# Change the values in the From column value into Title Case using List Comprehension
df['From'] = [x.title() for x in df['From'].values]

# Change the values in the To column value into Title Case using List Comprehension
df['To'] = [x.title() for x in df['To'].values]

print('\nView DataFrame after changing the From and To Column value into Title Case:')
df
```

```
View DataFrame after changing the From and To Column value into Title Case:
```

Out[300]:

	From_To	FlightNumber	RecentDelays	Airline	From	To
0	LoNDon_paris	10045	[23, 47]	KLM(!)	London	Paris
1	MAdrid_miLAN	10095	[]	(12)	Madrid	Milan
2	londON_StockhOlm	10065	[24, 43, 87]	(British Airways.)	London	Stockholm
3	Budapest_PaRis	10095	[13]	12. Air France	Budapest	Paris
4	Brussels_londOn	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 [301]:
```
# Drop From_To column from DataFrame and attach to temporary DataFrame
df_temp=pd.DataFrame(df.drop(['From_To'], axis=1))

print('\nView Temporary DataFrame after deleting From_To column from Original DataFrame:')
df_temp
```

```
View Temporary DataFrame after deleting From_To column from Original DataFrame:
```

Out[301]:

	FlightNumber	RecentDelays	Airline	From	To
0	10045	[23, 47]	KLM(!)	London	Paris
1	10095	[]	(12)	Madrid	Milan
2	10065	[24, 43, 87]	(British Airways.)	London	Stockholm
3	10095	[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 [302]:
```
# expand df.RecentDelays into delays dataframe
delays = df['RecentDelays'].apply(pd.Series)
delays
```

Out[302]:

	0	1	2
0	23.0	47.0	NaN
1	NaN	NaN	NaN
2	24.0	43.0	87.0
3	13.0	NaN	NaN
4	67.0	32.0	NaN

In [303]:
```
# rename the columns as delay_1, delay_2, etc
delays = delays.rename(columns = lambda x : 'delay_' + str(x+1))

# view the delays dataframe
delays
```

Out[303]:

	delay_1	delay_2	delay_3
0	23.0	47.0	NaN
1	NaN	NaN	NaN
2	24.0	43.0	87.0
3	13.0	NaN	NaN
4	67.0	32.0	NaN

In [304]:
```
# Replace the unwanted RecentDelays column in df with delays
df=df.drop('RecentDelays', axis=1).join(delays)
df
```

Out[304]:

	From_To	FlightNumber	Airline	From	To	delay_1	delay_2	delay_3
0	LoNDon_paris	10045	KLM(!)	London	Paris	23.0	47.0	NaN
1	MAdrid_miLAN	10095	(12)	Madrid	Milan	NaN	NaN	NaN
2	londON_StockhOlm	10065	(British Airways.)	London	Stockholm	24.0	43.0	87.0
3	Budapest_PaRis	10095	12. Air France	Budapest	Paris	13.0	NaN	NaN
4	Brussels_londOn	10085	"Swiss Air"	Brussels	London	67.0	32.0	NaN