datascience_assignment_12.1

July 11, 2018

0.1 Create dataframe as provided for the assignment

0.2 Steps:

- Import numpy, pandas
- Create DataFrame df as given for the assignment

In [45]:
```
# Import numpy and pandas
import numpy as np
import pandas as pd
```

In [46]:
```
#Create DataFrame df as given for the assignment
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"']})
```

0.2.1 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).

0.3 Steps

- Use interploate method on column FlightNumber to fill up missing values for fields so that numbers increase by 10 with each tow so 10055 and 10075 are in place
- Use asType method with int32 so that FlightNumber column becomes integer column

In [47]:
```
# Use interpolate method so that numbers increase by 10 with each row for FlightNumbe
# Use asType with arguments 'int32' and copy=False so that FlightNumber column become
df['FlightNumber'] = df['FlightNumber'].interpolate().astype('int32', copy=False)
```

In [48]:
```
df
```

Out[48]:

	Airline	FlightNumber	From_To	RecentDelays
0	KLM(!)	10045	LoNDon_paris	[23, 47]
1	<Air France> (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]

0.4 2. The From_To column would be better as two separate columns! Split each string onthe underscore delimiter _ to give a new temporary DataFrame with the correct values. Assign the correct column names to this temporary DataFrame.

0.5 Steps:

- Create an empty DataFrame df1
- Split From_To column based on delimiter _ and assign to two new Columns From, To on df1

```
In [52]: # Create an temporary DataFrame df1
         df1 = pd.DataFrame()

         #Split From_To column based on delimiter _ and assign to two new Columns From, To
         df1['From'], df1['To'] = df['From_To'].str.split('_', 1).str
```

```
In [53]: # Print df1
         df1
```

```
Out[53]:        From         To
         0    LoNDon      paris
         1    MAdrid      miLAN
         2    londON  StockhOlm
         3  Budapest      PaRis
         4  Brussels     londOn
```

0.6 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".)

0.7 Steps:

- Use string title method to convert the columns 'From' and 'To'

```
In [54]: #Use string title method to convert the columns 'From' and 'To'
         df1['From'], df1['To'] = df1['From'].str.title(), df1['To'].str.title()
```

```
In [55]: # Print df1
         df1
```

```
Out[55]:        From         To
         0    London      Paris
         1    Madrid      Milan
         2    London  Stockholm
         3  Budapest      Paris
         4  Brussels     London
```

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

0.9 Steps

- Use drop method on df to delete column 'From_To' use inplace=True to make it permanent

- Use concat method to concatenate df and df1 and assign to df

```
In [56]: # Use drop method on df to delete column 'From_To'
         df.drop(columns = ['From_To'], inplace=True)

In [57]: #Use concat method to concatenate df and df1 and assign to df
         df = pd.concat([df, df1], axis = 1)

In [58]: # Print df
         df
```

Out[58]:

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

0.10 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.

0.11 Steps

- RecentDelays value to list and assign to new columns delay_1, delay_2, delay_3 added to df

```
In [59]: # RecentDelays value to list and assign to new columns delay_1, delay_2, delay_3 adde
         df[['delay_1','delay_2','delay_3' ]] = pd.DataFrame(df.RecentDelays.values.tolist(),

In [60]: # Print df
         df
```

Out[60]:

	Airline	FlightNumber	RecentDelays	From	To	\
0	KLM(!)	10045	[23, 47]	London	Paris	
1	<Air France> (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	

	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