Session 11: Assignment 1

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(!)', '<Air France> (12)', '(British Airways. )',

'12. Air France', '"Swiss Air"']})

Question 1)

Code:

import numpy as np

import pandas as pd

df = pd.DataFrame({'From\_To': ['LoNDon\_paris', 'MAdrid\_miLAN', 'londON\_StockhOlm',

'Budapest\_PaRis', 'Brussels\_londOn'],

'FlightNumber': [10045, np.nan, 10065, np.nan, 10085],

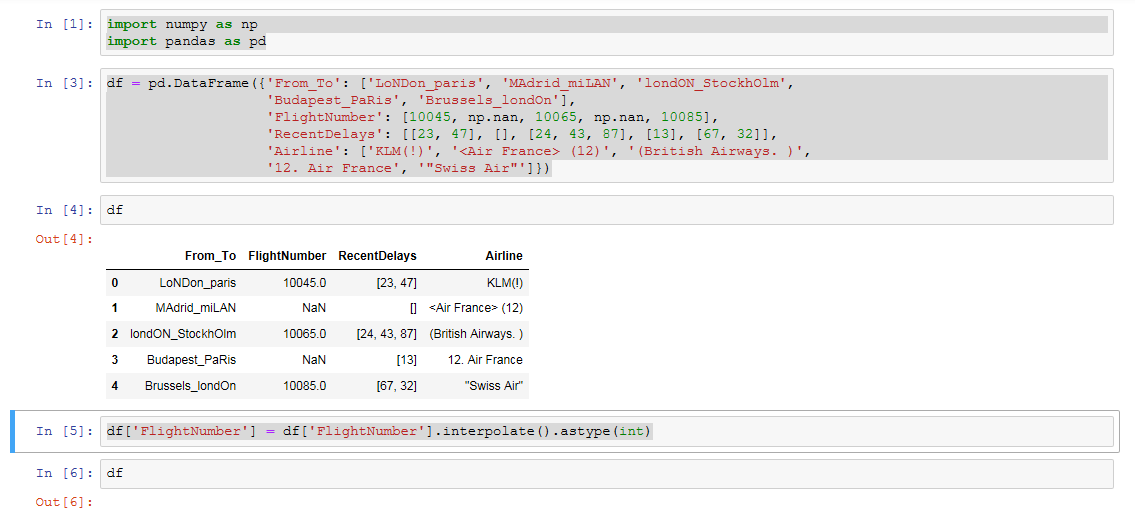
'RecentDelays': [[23, 47], [], [24, 43, 87], [13], [67, 32]],

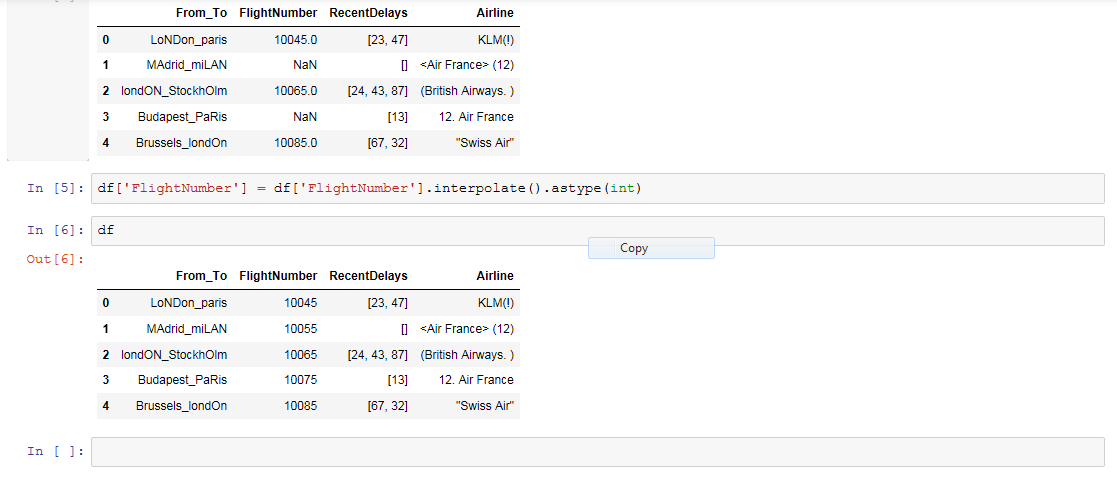
'Airline': ['KLM(!)', '<Air France> (12)', '(British Airways. )',

'12. Air France', '"Swiss Air"']})

df['FlightNumber'] = df['FlightNumber'].interpolate().astype(int)

Output:





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.

Code :

import numpy as np

import pandas as pd

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. )',

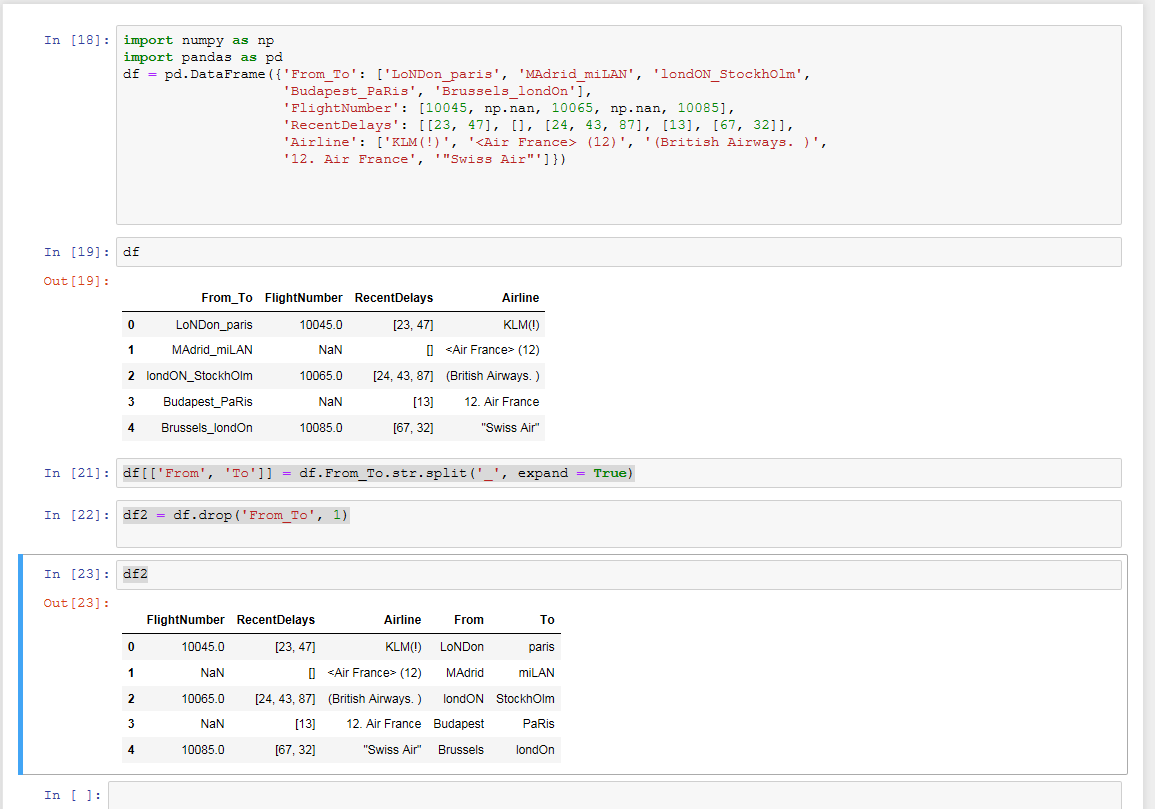
'12. Air France', '"Swiss Air"']})

df[['From', 'To']] = df.From\_To.str.split('\_', expand = True)

df2 = df.drop('From\_To', 1)

df2

Output:



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".)

Code:

import numpy as np

import pandas as pd

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"']})

df[['From', 'To']] = df.From\_To.str.split('\_', expand = True)

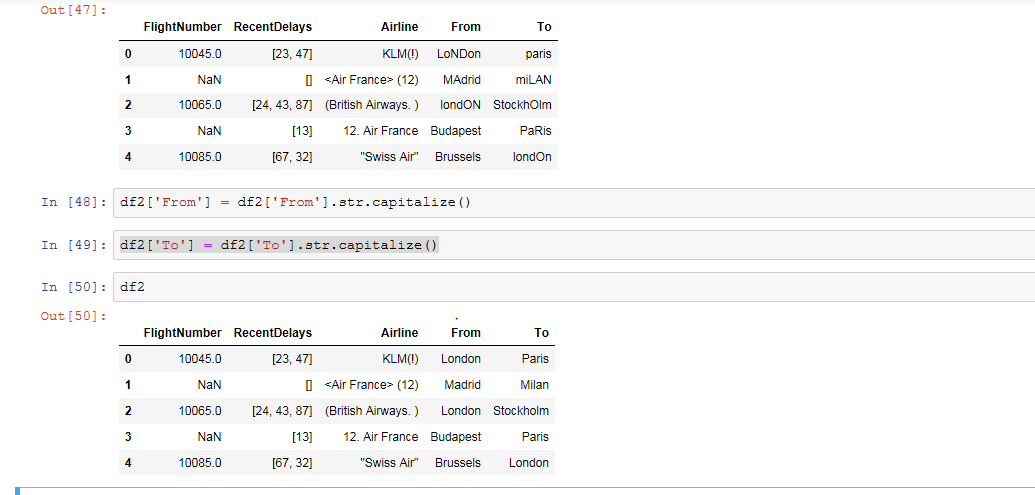
df2 = df.drop('From\_To', 1)

df2['Form'] = df2['From'].str.capitalize()

df2['To'] = df2['To'].str.capitalize()

df2

Output:



4) Delete the From\_To column from df and attach the temporary DataFrame from the

previous questions.

Code:

import numpy as np

import pandas as pd

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"']})

df[['From', 'To']] = df.From\_To.str.split('\_', expand = True)

df2 = df.drop('From\_To', 1)

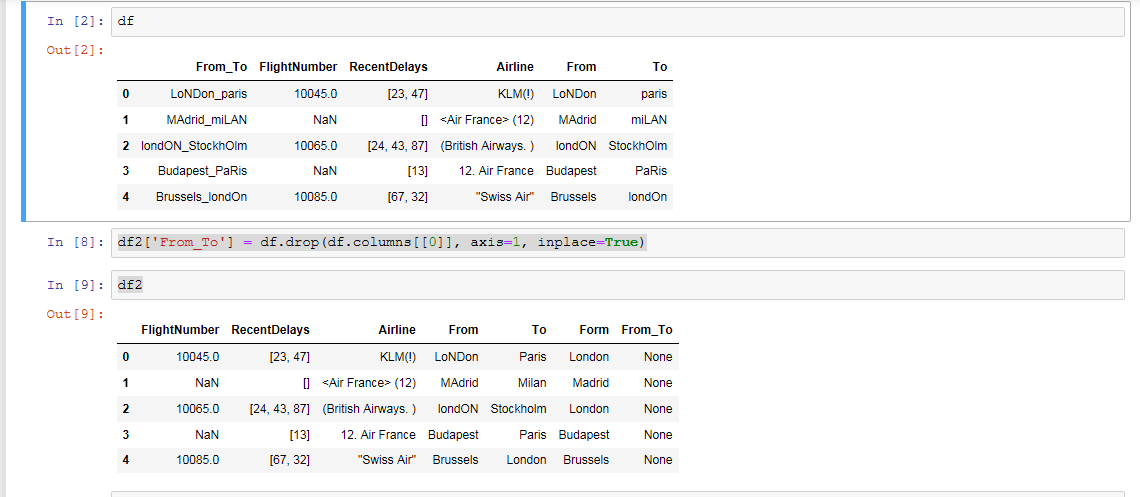
df2['Form'] = df2['From'].str.capitalize()

df2['To'] = df2['To'].str.capitalize()

df2['From\_To'] = df.drop(df.columns[[0]], axis=1, inplace=True)

df2

Output:



5) 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.