*Session 7: Assignment 1*

**Table of Contents**

1. Introduction
2. Problem Statement
3. Output

1. Introduction

This assignment will help you to consolidate the concepts learnt in the session.

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"']})

1. Some values in the Flight Number 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 capitalization of the city names is all mixed up in this temporary

DataFrame. Standardize 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 a 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.

**Note: The solution shared through Github should contain the source code used**

**and the screenshot of the output.**

***3. Solution:***

**Input**

***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(!)', '<Air France> (12)', '(British Airways. )', '12. Air France', '"Swiss Air"']})***

***#1. Fill in the missing numbers in the FlightNumber column and make the column an integer column.***

***df['FlightNumber'].loc[1] = (df['FlightNumber'].iloc[0] + 10)***

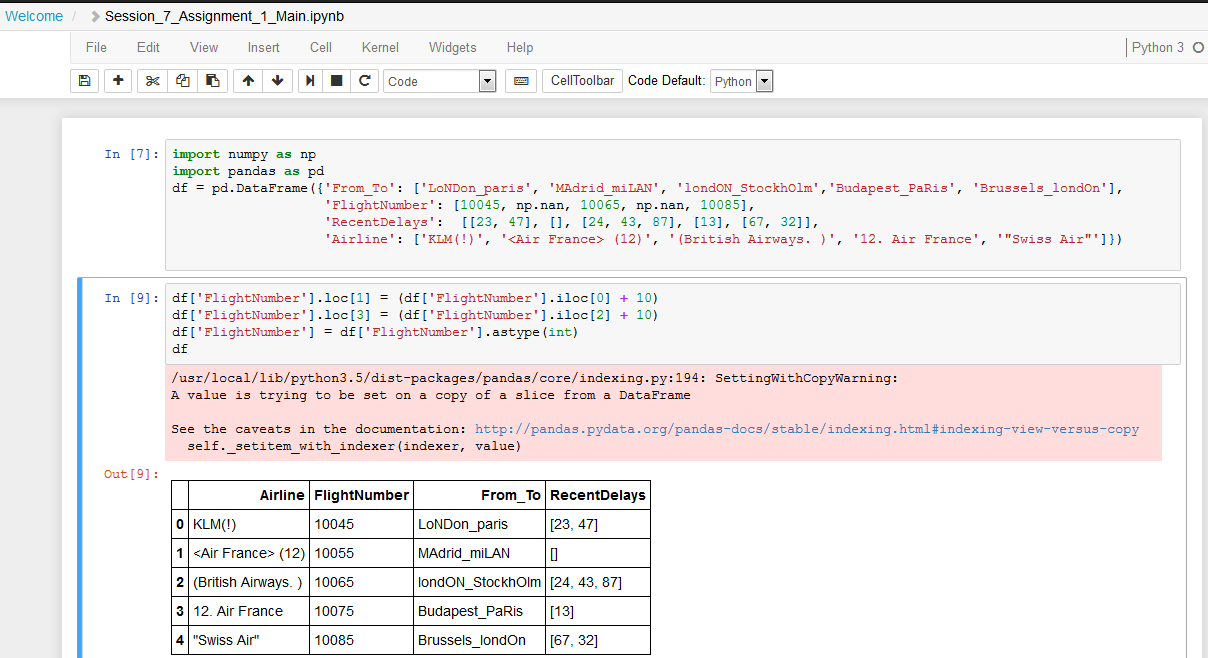
***df['FlightNumber'].loc[3] = (df['FlightNumber'].iloc[2] + 10)***

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

***df***

**Output 1:**

|  | **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] |

**Screen Shot 1:** 

***#2. Split the From\_To column with '\_' as the delimiter on a new temporary DataFrame. Assign the correct column names to this temporary DataFrame.***

***tDF = pd.DataFrame(df.From\_To)***

***tDF['From'] = tDF.From\_To.str.split('\_').str.get(0)***

***tDF['To'] = tDF.From\_To.str.split('\_').str.get(1)***

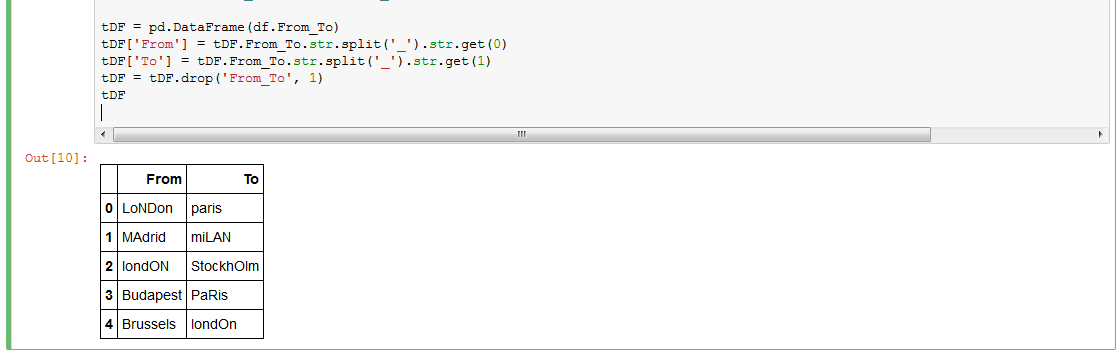
***tDF = tDF.drop('From\_To', 1)***

***tDF***

**Output 2:**

|  | **From** | **To** |
| --- | --- | --- |
| **0** | LoNDon | paris |
| **1** | MAdrid | miLAN |
| **2** | londON | StockhOlm |
| **3** | Budapest | PaRis |
| **4** | Brussels | londOn |

**Screen Shot2:**



***#3. Standardize the strings in the temporary DataFrame, so that only the first letter is uppercase.***

***tDF['From'] = tDF.From.str.title()***

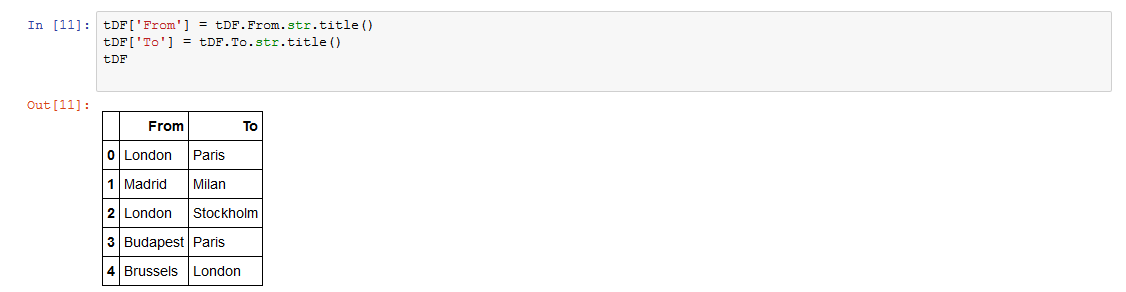
***tDF['To'] = tDF.To.str.title()***

***tDF***

**Output 3:**

|  | **From** | **To** |
| --- | --- | --- |
| **0** | London | Paris |
| **1** | Madrid | Milan |
| **2** | London | Stockholm |
| **3** | Budapest | Paris |
| **4** | Brussels | London |

**Screen Shot3:**



***#4. Delete the From\_To column from df, and attach the temporary DataFrame from the previous questions.***

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

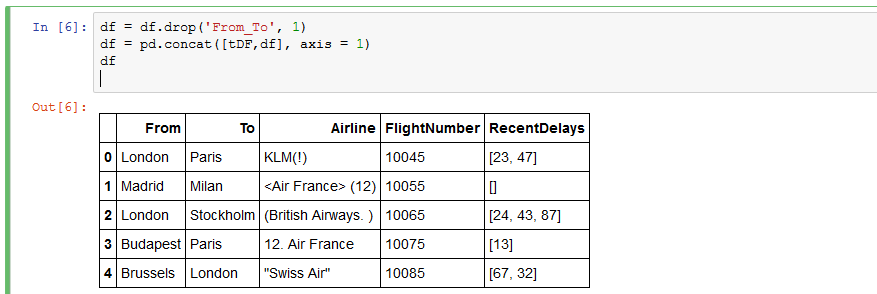
***df = pd.concat([tDF,df], axis = 1)***

***df***

**Output 4:**

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

**Screen Shot4:**



***#5. Keep each first value in its own column, each second value in its own column, and so on. If there isn't a 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.***

***#(i)***

***tDelay = pd.DataFrame(df.RecentDelays)***

***tDelay = pd.DataFrame(df['RecentDelays'].values.tolist())***

***tDelay.columns = ['Delay\_1', 'Delay\_2', 'Delay\_3']***

***df = df.drop('RecentDelays', 1)***

***df.insert(3, "Delay\_1", tDelay['Delay\_1'])***

***df.insert(4, "Delay\_2", tDelay['Delay\_2'])***

***df.insert(5, "Delay\_3", tDelay['Delay\_3'])***

***print(df)***

**Output 5(i):**

From To Airline Delay\_1 Delay\_2 Delay\_3 FlightNumber

0 London Paris KLM(!) 23.0 47.0 NaN 10045

1 Madrid Milan <Air France> (12) NaN NaN NaN 10055

2 London Stockholm (British Airways.) 24.0 43.0 87.0 10065

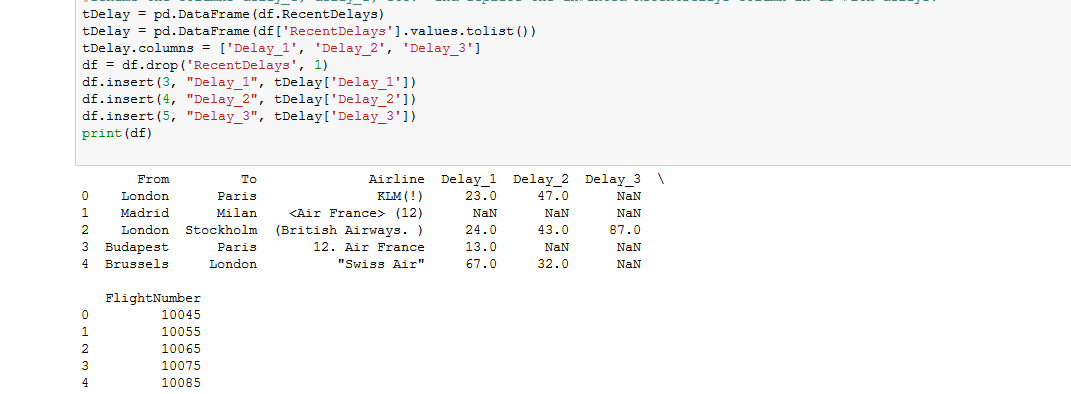
3 Budapest Paris 12. Air France 13.0 NaN NaN 10075

4 Brussels London "Swiss Air" 67.0 32.0 NaN 10085

**Output 5(i):**

|  | **From** | **To** | **Airline** | **Delay\_1** | **Delay\_2** | **Delay\_3** | **FlightNumber** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | London | Paris | KLM(!) | 23.0 | 47.0 | NaN | 10045 |
| **1** | Madrid | Milan | <Air France> (12) | NaN | NaN | NaN | 10055 |
| **2** | London | Stockholm | (British Airways. ) | 24.0 | 43.0 | 87.0 | 10065 |
| **3** | Budapest | Paris | 12. Air France | 13.0 | NaN | NaN | 10075 |
| **4** | Brussels | London | "Swiss Air" | 67.0 | 32.0 | NaN | 10085 |

**Screen Shot5:**



***df***

**Output 5(ii):**

