**Q: 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 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). 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 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".) 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 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.**

**answers:**

import pandas as pd

import numpy as np

# Initial DataFrame setup

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"']

})

print("Original DataFrame:")

print(df)

**1. Fill in missing FlightNumber values and convert the column to integers**

**Python Code**

# Fill missing FlightNumber values

for i in range(1, len(df)):

if pd.isna(df.loc[i, 'FlightNumber']):

df.loc[i, 'FlightNumber'] = df.loc[i - 1, 'FlightNumber'] + 10

# Convert FlightNumber to integers

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

print("\nDataFrame with FlightNumber filled and converted to integers:")

print(df)

**2. Split the From\_To column into two separate columns**

**Python Code**

# Split the From\_To column

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

temp\_df.columns = ['From', 'To']

print("\nTemporary DataFrame with 'From' and 'To' columns:")

print(temp\_df)

**3. Standardize the capitalization of city names**

**Python Code**

# Standardize capitalization

temp\_df['From'] = temp\_df['From'].str.capitalize()

temp\_df['To'] = temp\_df['To'].str.capitalize()

print("\nTemporary DataFrame with standardized city names:")

print(temp\_df)

**4. Delete the From\_To column from df and attach the new DataFrame**

**Python Code**

# Delete the From\_To column and attach the new DataFrame

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

df = pd.concat([df, temp\_df], axis=1)

print("\nDataFrame after deleting From\_To and attaching new columns:")

print(df)

**5. Expand RecentDelays into separate columns**

**Python Code**

# Expand RecentDelays into separate columns

delays = df['RecentDelays'].apply(pd.Series)

# Rename the columns

delays.columns = [f'delay\_{i+1}' for i in range(delays.shape[1])]

# Replace RecentDelays column with new delay columns

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

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

print("\nFinal DataFrame with expanded RecentDelays:")

print(df)