**Dataframe**

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"']})

**Problem statement**

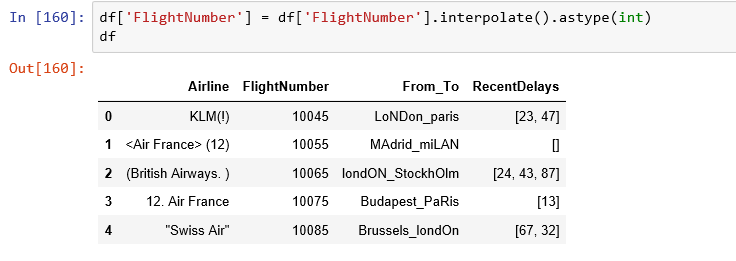
Some values in the the Flight Number column are missing. These numbers are meant to increase by 10 with each row so 10055 and 10075 need to be put in place. Fill in these missing numbers and make the column an integer column (instead of a float column).

**Code**

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

df

**Screenshot**



**Problem statement**

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**

df1=df

s = df1['From\_To'].apply(lambda x: x.split('\_'))

df1['from'] = s.apply(lambda x: x[0])

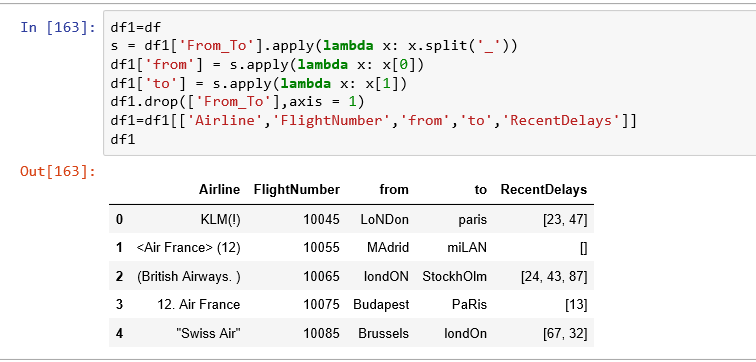
df1['to'] = s.apply(lambda x: x[1])

df1.drop(['From\_To'],axis = 1)

df1=df1[['Airline','FlightNumber','from','to','RecentDelays']]

df1

**Screenshot**



**Problem statement**

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**

df1=df1['to'].apply(lambda x: x.title())

df1=df1['from'].apply(lambda x: x.title())

**Screenshot**



**Problem statement**

Delete the From\_To column from df and attach the temporary DataFrame from the previous​ ​questions.

**Code**

to=df1['to'].apply(lambda x: x.title())

fro=df1['from'].apply(lambda x: x.title())

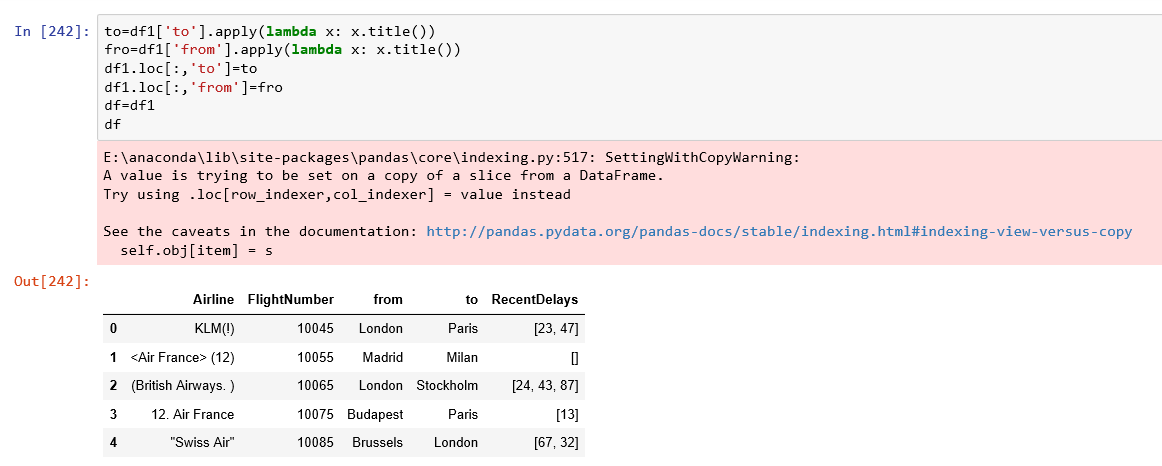
df1.loc[:,'to']=to

df1.loc[:,'from']=fro

df=df1

df

**Screenshot**



**Problem statement**

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

**Code**

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

delays.columns = ['delay\_{}'.format(n) for n in range(1, len(delays.columns)+1)]

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

df

**Screenshot**

