In [5]:
```
# It happens all the time: someone gives you data containing malformed string
s, Python,
# lists and missing data. How do you tidy it up so you can get on with the ana
lysis?
# Take this monstrosity as the DataFrame to use in the following puzzles:

# df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN', 'londON_Stock
hOlm','Budapest_PaRis', 'Brussels_londOn'],
#'FlightNumber': [10045, np.nan, 10065, np.nan, 10085],'RecentDelays': [[23, 4
7], [], [24, 43, 87], [13], [67, 32]],
#'Airline': ['KLM(!)', '<Air France> (12)', '(British Airways. )','12. Air Fra
nce', '"Swiss Air"']})
```

In [2]:
```
# 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 inthese missi
ng numbers and make the
# column an integer column (instead of a float column).

import numpy as np
import pandas as pd

df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN', 'londON_StockhO
lm',
'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
```

Out[2]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	<Air France> (12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	NaN	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [3]:
```
df['FlightNumber']
```

Out[3]:
```
0    10045.0
1        NaN
2    10065.0
3        NaN
4    10085.0
Name: FlightNumber, dtype: float64
```

In [4]:

```
#Setting up new index for the data frame. This index is used for the for loop
 iteration created in next step
newindex=np.arange(1,df.From_To.count()+1)
newindex
df.set_index(newindex, inplace=True)
df
```

Out[4]:

	From_To	FlightNumber	RecentDelays	Airline
1	LoNDon_paris	10045.0	[23, 47]	KLM(!)
2	MAdrid_miLAN	NaN	[]	<Air France> (12)
3	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
4	Budapest_PaRis	NaN	[13]	12. Air France
5	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [5]:

```
#using for loop for iteration along with isnull function to update the values
 for column FlightNumber
for i in np.arange(1,df.From_To.count()+1):
    if pd.isnull(df.FlightNumber.loc[i,]):
        df.loc[i,'FlightNumber'] = df.FlightNumber.loc[i-1,] + 10
df['FlightNumber']
df
```

Out[5]:

	From_To	FlightNumber	RecentDelays	Airline
1	LoNDon_paris	10045.0	[23, 47]	KLM(!)
2	MAdrid_miLAN	10055.0	[]	<Air France> (12)
3	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
4	Budapest_PaRis	10075.0	[13]	12. Air France
5	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [6]:

```
# Changing the data type for FlightNumber column to integer
df['FlightNumber'].astype(int)
```

Out[6]:

```
1    10045
2    10055
3    10065
4    10075
5    10085
Name: FlightNumber, dtype: int32
```

In [7]:
```
# 2. The From_To column would be better as two separate columns! Split each st
ring on
# the underscore delimiter _ to give a new temporary DataFrame with the correc
t values.
# Assign the correct column names to this temporary DataFrame.

df['From_To']
```

Out[7]:
```
1        LoNDon_paris
2        MAdrid_miLAN
3    londON_StockhOlm
4      Budapest_PaRis
5     Brussels_londOn
Name: From_To, dtype: object
```

In [8]:
```
#Creating a new temporary dataframe which is a copy of existing data frame df
temporarydf = df.copy()

#Spliting the column into two based on "_"
temporarydf[['From','To']] = temporarydf.From_To.str.split("_",expand=True)

#Printing new data frame
temporarydf
```

Out[8]:

	From_To	FlightNumber	RecentDelays	Airline	From	To
1	LoNDon_paris	10045.0	[23, 47]	KLM(!)	LoNDon	paris
2	MAdrid_miLAN	10055.0	[]	<Air France> (12)	MAdrid	miLAN
3	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)	londON	StockhOlm
4	Budapest_PaRis	10075.0	[13]	12. Air France	Budapest	PaRis
5	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"	Brussels	londOn

In [9]:

```
# 3. Notice how the capitalisation of the city names is all mixed up in this t
emporary
# DataFrame. Standardise the strings so that only the first letter is uppercas
e (e.g."londON" should become "London".)

#Converting the first letter of values in 'From 'column into uppercase
temporarydf.From = temporarydf.From.str.capitalize()

#Converting the first letter of values in 'To 'column into uppercase
temporarydf.To = temporarydf.To.str.capitalize()

#Converting the first letter of values in 'From_To 'column into uppercase
temporarydf.From_To = temporarydf.From_To.str.capitalize()

print(temporarydf)
```

```
             From_To  FlightNumber  RecentDelays             Airline  \
1      London_paris       10045.0      [23, 47]              KLM(!)
2      Madrid_milan       10055.0            []    <Air France> (12)
3  London_stockholm       10065.0  [24, 43, 87]  (British Airways. )
4    Budapest_paris       10075.0          [13]       12. Air France
5   Brussels_london       10085.0      [67, 32]         "Swiss Air"

       From         To
1    London      Paris
2    Madrid      Milan
3    London  Stockholm
4  Budapest      Paris
5  Brussels     London
```

In [10]:

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

#Printing the exisiting df
df
```

Out[10]:

	From_To	FlightNumber	RecentDelays	Airline
1	LoNDon_paris	10045.0	[23, 47]	KLM(!)
2	MAdrid_miLAN	10055.0	[]	<Air France> (12)
3	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
4	Budapest_PaRis	10075.0	[13]	12. Air France
5	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [23]:
```
#Printing the data frame after deleting the "From_To" column
df.drop('From_To',axis=1,inplace=True)
df
```

Out[23]:

	FlightNumber	RecentDelays	Airline
1	10045.0	[23, 47]	KLM(!)
2	10055.0	[]	<Air France> (12)
3	10065.0	[24, 43, 87]	(British Airways.)
4	10075.0	[13]	12. Air France
5	10085.0	[67, 32]	"Swiss Air"

In [22]:
```
# Adding the 'From_To' column from temporary database
df['From_To'] = temporarydf['From_To']
df
```

Out[22]:

	FlightNumber	RecentDelays	Airline	From_To
1	10045.0	[23, 47]	KLM(!)	London_paris
2	10055.0	[]	<Air France> (12)	Madrid_milan
3	10065.0	[24, 43, 87]	(British Airways.)	London_stockholm
4	10075.0	[13]	12. Air France	Budapest_paris
5	10085.0	[67, 32]	"Swiss Air"	Brussels_london

In [ ]:
```
# 5. In the RecentDelays column, the values have been entered into the DataFra
me 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 delay
s.
```

In [111]:

```
# 5. In the RecentDelays column, the values have been entered into the DataFra
me 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.

#Using the original dataframe provided for this problem.

df = pd.DataFrame({'From_To': ['LoNDon_paris', 'MAdrid_miLAN', 'londON_StockhO
lm',
'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
rows = []
_ = df.apply(lambda row:[rows.append([row['Airline'], row['FlightNumber'],nn,r
ow['From_To']])
                         for nn in row.RecentDelays], axis=1)
```

In [98]:

```
#Printing all values in recent delay column in seperate rows
rows
```

Out[98]:

```
[['KLM(!)', 10045.0, 23, 'LoNDon_paris'],
 ['KLM(!)', 10045.0, 47, 'LoNDon_paris'],
 ['(British Airways. )', 10065.0, 24, 'londON_StockhOlm'],
 ['(British Airways. )', 10065.0, 43, 'londON_StockhOlm'],
 ['(British Airways. )', 10065.0, 87, 'londON_StockhOlm'],
 ['12. Air France', nan, 13, 'Budapest_PaRis'],
 ['"Swiss Air"', 10085.0, 67, 'Brussels_londOn'],
 ['"Swiss Air"', 10085.0, 32, 'Brussels_londOn']]
```

In [99]:

```
#Converting the data into data frame
df_new = pd.DataFrame(rows, columns=df.columns)

#Printing existing dataframe (for comparison view)
df
```

Out[99]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	<Air France> (12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	NaN	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [101]:

```
# Printing the revised data frame as per the criteria defined in the problem.
df_new
```

Out[101]:

	From_To	FlightNumber	RecentDelays	Airline
0	KLM(!)	10045.0	23	LoNDon_paris
1	KLM(!)	10045.0	47	LoNDon_paris
2	(British Airways.)	10065.0	24	londON_StockhOlm
3	(British Airways.)	10065.0	43	londON_StockhOlm
4	(British Airways.)	10065.0	87	londON_StockhOlm
5	12. Air France	NaN	13	Budapest_PaRis
6	"Swiss Air"	10085.0	67	Brussels_londOn
7	"Swiss Air"	10085.0	32	Brussels_londOn

In [102]:

```
# 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 delay
s.

#Getting the recent delay values from the data frame
df3 = pd.DataFrame(df['RecentDelays'].values.tolist())
df3
```

Out[102]:

	0	1	2
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

In [103]:

```
length_cols = df3.shape[1]
length_cols
```

Out[103]:

```
3
```

In [104]:

```
df3.columns[0]
```

Out[104]:

```
0
```

In [105]:

```
#Creating a for loop iteration for renaming the columns

col_list = []
col_dict ={}
for i in range(length_cols):
    Key = df3.columns[i]
    #print(key,i)
    Value = "Delay" + str(i+1)
    col_dict[Key] = Value

col_dict
```

Out[105]:

```
{0: 'Delay1', 1: 'Delay2', 2: 'Delay3'}
```

In [106]:

```
# Renaming the columns

df3.rename(columns=col_dict,inplace=True)
df3
```

Out[106]:

	Delay1	Delay2	Delay3
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

In [112]:

```
#Printing the existing data frame for comparison
df
```

Out[112]:

	From_To	FlightNumber	RecentDelays	Airline
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)
1	MAdrid_miLAN	NaN	[]	<Air France> (12)
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)
3	Budapest_PaRis	NaN	[13]	12. Air France
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"

In [114]:

```
df[["Delay1","Delay2","Delay3"]] = df3[["Delay1","Delay2","Delay3"]]
```

In [115]:

```
#Adding the new columns to the data frame
df
```

Out[115]:

	From_To	FlightNumber	RecentDelays	Airline	Delay1	Delay2	Delay3
0	LoNDon_paris	10045.0	[23, 47]	KLM(!)	23.0	47.0	NaN
1	MAdrid_miLAN	NaN	[]	<Air France> (12)	NaN	NaN	NaN
2	londON_StockhOlm	10065.0	[24, 43, 87]	(British Airways.)	24.0	43.0	87.0
3	Budapest_PaRis	NaN	[13]	12. Air France	13.0	NaN	NaN
4	Brussels_londOn	10085.0	[67, 32]	"Swiss Air"	67.0	32.0	NaN

In [116]:

```
#Printing the revised dataframe by dropping the recent delays column as mentioned in the problem.

df.drop('RecentDelays',axis=1,inplace=True)
df
```

Out[116]:

	From_To	FlightNumber	Airline	Delay1	Delay2	Delay3
0	LoNDon_paris	10045.0	KLM(!)	23.0	47.0	NaN
1	MAdrid_miLAN	NaN	<Air France> (12)	NaN	NaN	NaN
2	londON_StockhOlm	10065.0	(British Airways.)	24.0	43.0	87.0
3	Budapest_PaRis	NaN	12. Air France	13.0	NaN	NaN
4	Brussels_londOn	10085.0	"Swiss Air"	67.0	32.0	NaN