Simple Data Imputation in Python

Introduction

Real world datasets have a lot of missing values. There can be many types of missing values:

1. Missing Completely at Random (MCAR)
2. Missing at Random (MAR
3. Not Missing at Random (NMAR)

An essential part of data wrangling involves handling these missing values. There are a few different approaches to do this:

1. Not do anything
   This approach will not work if the dataset has values crucial for predictive modelling which are missing, or if there are too many missing values
2. Imputation using statistical measures (mean/median/mode)
   This is a very common approach that will replace the missing values which the mean/median/mode of fields having values. Mean imputation is especially popular, mainly because it's a very easy method. But it has some glaring drawbacks: mean imputation alters the correlation between attributes (if the data IS NOT MCAR)
3. Imputing using values from predictive modelling, such as regression, KNN, deep learning
   Another way is to apply a machine learning algorithm on the non-missing values to predict the missing values. This gives a decent outcome, but there's a chance that it might cause data snooping.

Goal

Handling missing values is important for any statistical analyses and data visualization. The very popular sklearn package in Python comes with a ready-made imputation API which we will use.
The goal is to demonstrate how to use the SimpleImputer API.

Dataset

We are using the "World Bank International Arrivals" dataset, which has a lot of missing values characterized by <nan> placeholders.
The dataset mainly outlines the population, gdp, no.of hotels, and international passengers information such as passengers from USA, UK, Australia and overnight arrivals. All details of the dataset attributes/fields given below:

Variables	Description
country	Name of the country, indexed by an integer
year	Range from 0 to 12 representing a 13 year time frame
pop	Population of a country
areakm2	Area of a country in km^2
gdpnom	Gross domestic product (GDP) of the country, indicating the economic development
flights - WB	Number of flights
hotels	Number of hotels
hotrooms	Number of hotel rooms
receipt	Tourism receipts, which defined (by WTO) as expenditure of international inbound visitors including their payments to national carriers for international transport. They also include any other payments or payments afterwards made for goods and services received in the destination country.
ovnarriv	Overnight arrival
dayvisit	Number of visit days
arram	Number of arrivals from America
arreur	Number of arrivals from Europe
arraus	Number of arrivals from Australia

A quick glance at the dataset below shows too many missing values to ignore. Even replacing with a constant value such as 0 will make the dataset invalid for any practical uses.

country	year	pop	areakm2	gdpnom	flights - WB	hotels	hotrooms	receipt	ovnarriv	dayvisit	arram	arreur	arraus
1	0	1187000	2040	.	12162	.	.	732000000	656000	22000	.	.	.
1	1	1189800	2040	4684000000	12269	95	9024	820000000	660000	15000	.	.	.
1	2	1200200	2040	4911000000	12720	95	9623	829000000	682000	27000	.	.	.
1	3	1210400	2040	5825000000	12969	97	9647	960000000	702000	20000	.	.	.
1	4	1220500	2040	6594000000	14791	103	10640	1156000000	719000	20000	.	.	.
1	5	1243000	2040	6489000000	14743	99	10497	1189000000	761000	20000	.	.	.
1	6	1240800	2040	6732000000	13692	98	10666	1302000000	788000	19000	.	.	.
1	7	1250900	2040	7792000000	12090	97	10857	1663000000	907000	26000	10000	596000	10969
1	8	1274200	2040	9641000000	11742	102	11488	1823000000	930000	40000	14000	609000	8974
1	9	1284300	2040	8824000000	11144	102	11456	1390000000	871000	19000	13000	580000	8106
1	10	1283400	2040	9706000000	11750	112	12075	1585000000	935000	21000	14000	606000	9255
1	11	1288700	2040	1.1244E+10	12353	109	11925	1813000000	965000	18000	14000	610000	8822
1	12	1293500	2040	1.1466E+10	12258	.	.	.	.	.	.	.	.
2	0	.	451	615000000	18966	.	.	225000000	130000	.	.	.	.
2	1	81000	451	622000000	20249	.	.	221000000	130000	.	.	.	.
2	2	84000	451	698000000	22449	.	.	247000000	132000	.	.	.	.
2	3	83000	451	706000000	18647	.	.	258000000	122000	.	.	.	.
2	4	83000	451	857000000	19281	.	.	256000000	121000	.	.	.	.
2	5	83000	451	937000000	19581	.	.	269000000	129000	.	.	.	.
2	6	85000	451	1037000000	19526	.	.	323000000	141000	.	.	.	.
2	7	85000	451	1039000000	20727	103	2710	396000000	161000	10000	4	130	3316
2	8	87000	451	962000000	11896	107	2360	408000000	159000	14000	4	125	3070
2	9	87000	451	841000000	11238	115	2490	349000000	158000	20000	5	122	2852
2	10	90000	451	973000000	12989	124	2510	352000000	175000	16000	4	132	3446
2	11	87000	451	1060000000	14202	132	.	378000000	194000	.	5	144	3745
2	12	.	451	1031000000	11878	.	.	.	.	.	.	.	.
3	0	777000	440	788000000	62070			291000000	207000				
3	1	792000	440	778000000	72001			272000000	215000				
3	2	805000	440	807000000	63361			274000000	218000				
3	3	817000	440	849000000	67163			300000000	239000				
3	4	828000	440	905000000	25700			337000000	246000				
3	5	839000	440	1002000000	26985			309000000	245000				
3	6	849000	440	1141000000	26175			327000000	254000				
3	7	859000	440	1296000000	27746	4157	.	338000000	262000	673000	141000	115000	
3	8	869000	440	1355000000	28356	4673	.	334000000	266000	597000	152000	110000	

world_bank_international_arriva +

There are 27 countries in the dataset, each country is having information related to GDP, population size and international travelers' data for 13 years. As you can see, each country has entire attribute columns that are missing values for the 13-year timeframe. In some cases, only some years are missing values. In the screenshot above, country 1 is having missing values for "hotels" in years 0 and 12.

Country Key

Index	Country	Index	Country	Index	Country
1	Mauritius	10	Dominica	19	St Lucia
2	Seychelles	11	Kiribati	20	Cayman Islands
3	Antigua & Barbuda	12	Maldives	21	St Vincent & Grenadines
4	Grenada	13	Malta	22	Tonga
5	Bahrain	14	Marshall Islands	23	Tuvalu
6	Barbados	15	Fed Micronesia	24	Palau
7	Bermuda	16	Samoa	25	Singapore
8	Cape Verde	17	Sao Tome & Principe	26	Trinidad and Tobago
9	Comoros	18	St Kitts and Nevis	27	Solomon Islands

SimpleImputer API in Python

1. Import the dataset

```
In [4]: import pandas as pd
        import numpy as np

In [5]: df = pd.read_csv('/Users/jayasmitachakraborty/Downloads/world_bank_international_arrivals_islands.csv')
        df.replace('.', np.nan, inplace=True)
```

2. Import the SimpleImputer API

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
```

The SimpleImputer has various strategies for computing the missing values.

1. mean (default)
2. median
3. most frequent
4. constant

More information in documentation: https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html

3. Applying the SimpleImputer to the dataset

Note that there are 27 countries, and each country holds information for 13 years. So, we cannot use mean over each attribute columns. If we do that, we will use data of

country 1 to impute missing data in other countries. And this applies to each and every country.
The code below only computes mean for every individual country. Hence the while loop which will do the mean imputation for the missing values for every single country.

```
start = 0
stop = df.shape[0]
country = 0
while start<stop:
    country = country + 1
    df_ = df[start:start+13]

    imputer.fit(df_)
    df2 = pd.DataFrame(imputer.transform(df_))
    file_name = str(country) + ".csv"
    df2.to_csv(file_name)
    start = start+13
```

This creates 27 files for the 27 nations.

Problem

This sort of data imputation does not resolve the missing values issue completely.

1. We get 27 different files, which we need to combine manually
2. Note that we said that there are entire columns with missing values for certain countries. So, mean cannot be computed for such columns, and we are still left with missing values.

Conclusion

SimpleImputer with mean/median/mode/constant strategies of imputation is very basic. It is the starting point of data preprocessing and by no means a foolproof way to imputing missing values. But it gives us an idea of the data and helps us formulate the next steps in the data wrangling process. And in some cases, simple imputation may even populate the missing values correctly such that the final dataset can be used for predictive modelling.

Code in Github

For SimpleImputer in Python:
https://github.com/jayasmitachakraborty/SimpleImputer-in-Python

For the correct imputation of this dataset in R:
https://github.com/jayasmitachakraborty/Data-Visualization-Using-GGPLOT