### Data Management in Python –

### Handling Missing Values

Detecting, Excluding and Imputing NA's

### Contents

- 1. Introduction
- 2. Understanding Missing Data Mechanism
- 3. Replacing Missing Values with NA's while Importing
- 4. Detecting NA's
- 5. Excluding Missing Values from Analysis
- 6. Imputing Missing Values

### Introduction

- Missing values in data is a common phenomenon in real world problems and can create problems for simple and complicated analysis.
- You need to know the mechanism of missingness and how to treat them is a requirement to reduce the bias and to produce powerful models.
- Let's get familiar with the mechanisms of missingness and explore various options of how to deal with them.

### Missing Data Mechanism

#### Three types of missing data:

- Missing Completely at Random (MCAR):
   MCAR happens when missingness is totally unrelated to the variables in the dataset. For instance, if your equipment just flips out sometimes for no reason and doesn't record stuff, that will result in missing data that is MCAR.
- Missing at Random (MAR):
   MAR happens when the missingness is related to the information in your study.
   Other variables (but not the variable that is missing itself) in the dataset can be used to predict missingness. For instance, if men are more likely to tell you their weight than women, weight is MAR.
- Missing not at Random (MNAR):
   MNAR happens when missingness is related to missing data in your dataset.
   For instance, single people are less likely to report martial status than married people.

### Missing Data Mechanism

- MNAR is 'non ignorable' because we have to include some model for why the data are missing and what the likely values are as we deal with the missing data.
- MCAR and MAR are both considered 'ignorable' because we don't have to include any information about the missing data itself when we deal with the missing data.

Let's go ahead with testing and dealing missing data

# Data Snapshot

basicsalary data consist salary of each employee with it's Location & Grade. The data has 12 rows and 6 columns with 2 missing values.

	Variables							
	First_Na Alan Agath	Brown	GR1 DE		ation ELHI MBAI	ba 17990 12390	ms 16070 6630	
SUC	Columns	Description	Type Mea		Meas	urement	Possible values	
Observations	First_Name	First Name	character		-		-	
	Last_Name	Last Name	character		-		-	
	Grade	Grade	character		GR1, GR2		2	
	Location	Location	character		DELHI, MUMBAI		2	
	ba	Basic Allowance	numeric		Rs.		positive values	
	ms	Management Supplements	numeric		Rs.		positive values	

# Replacing Missing Values with NA while Importing the Data

A missing value is one whose value is unknown. Missing values in Python appears as NaN. NaN is not a string or a numeric value, but an indicator of missingness. Our data has two missing values, let's see what happens when we import this data in Python.

# Import Data and check how Python treats missing data while importing import pandas as pd salary data = pd.read csv("basic salary.csv") # Output First\_Name Last\_Name Grade Location ba ms Note **read csv()** replaces Alan GR1 17990 Brown DELHI 16070.0 blank fields with NaN Agatha Williams GR2 MUMBAI 12390 6630.0 Rajesh Kolte GR1 MUMBAI 19250 14960.0 Ameet Mishra GR2 DELHI 14780 9300.0 Neha MUMBAI 19235 15200.0 Rao NaN Chavan GR2 MUMBAI 13390 6700.0 Sagar 23280 13490.0 MUMBAI Aaron Jones GR1 Patil John MUMBAI 13500 10760.0 GR2 Joshi Sneha GR1 DELHI 20660 NaN Gauray Singh GR2 DELHI 13760 13220.0 10 Adela Thomas GR2 DELHI 13660 6840.0 11 GR2 MUMBAI 11960 7880.0 Anup Save

### Detecting NA's

# Check whether our data has missing values or not

salary data.isnull() First Name Last Name Grade Location ba isnull() returns logical matrix with False False False False False False the same dimensions as the data False frame. False True False True False 10

False False False

# Check total missing values

False False

False

11

- isnull() for dataframe returns a logical matrix with the same dimensions as the data frame, and with dimnames taken from the row and column names of the data frame.
- sum() returns the total no. of missing values in the data by column.

### Detecting NA's

# Check Number of missing data per column

```
salary data.info()
# Output
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12 entries, 0 to 11
                                       Using info() we can check how many
Data columns (total 6 columns):
                                       NaN's our data contains.
First Name
           12 non-null object
Last Name 12 non-null object
           11 non-null object
Grade
Location
           12 non-null object
            12 non-null int64
ba
             11 non-null float64
dtypes: float64(1), int64(1), object(4)
memory usage: 656.0+ bytes
```

### Excluding Missing Values from Analysis

- Missing data is problematic because most statistical procedures require a value for each variable. When the data is incomplete, you have to decide how to deal with it.
- When Python encounters missing value, it attempts to perform the requested procedure and returns a missing (NaN) value as a result. One way of dealing with

```
import numpy as np
from statistics import *
x = [10,30,12,np.nan, 9]
mean(x)

Nan
This output can be interpreted as: our vector contains missing value, so the requested statistic - the mean - is undefined for this data.
```

# We can calculate mean by dropping missing value like in the next example. # remove missing value

```
np.nanmean(x)

nanmean will remove all NaN's while performing the requested procedure.
```

### Excluding Missing Values from Analysis

Case wise deletion (complete case analysis) is the easiest way to deal with missing data. It simply removes all the cases with missing data anywhere in the data i.e. analysing only the cases with complete data.

#### # Case wise deletion

salary data.dropna()

# Output

ms	ba	Location	Grade	Last_Name	First_Name	
16070.0	17990	DELHI	GR1	Brown	Alan	0
6630.0	12390	MUMBAI	GR2	Williams	Agatha	1
14960.0	19250	MUMBAI	GR1	Kolte	Rajesh	2
9300.0	14780	DELHI	GR2	Mishra	Ameet	3
6700.0	13390	MUMBAI	GR2	Chavan	Sagar	5
13490.0	23280	MUMBAI	GR1	Jones	Aaron	6
10760.0	13500	MUMBAI	GR2	Patil	John	7
13220.0	13760	DELHI	GR2	Singh	Gaurav	9
6840.0	13660	DELHI	GR2	Thomas	Adela	10
7880.0	11960	MUMBAI	GR2	Save	Anup	11

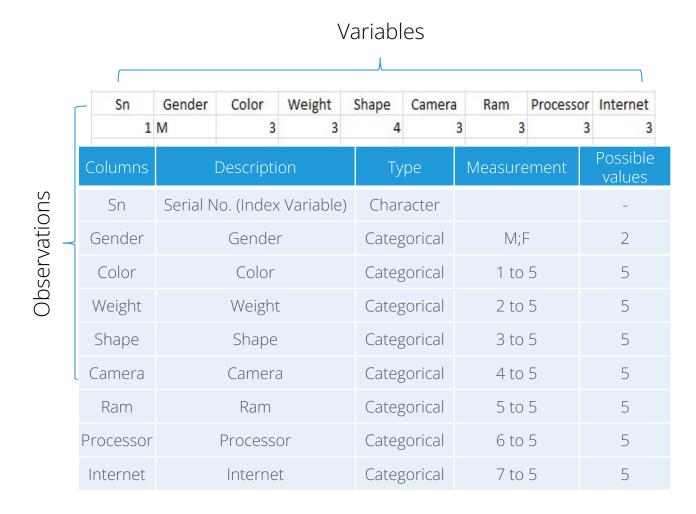
- .dropna() is used for case deletion.
- Here, .dropna()removes 2 rows thatcontains mising values



Complete case analysis is widely used method for handling missing data, and is a default method in many statistical packages. But it has limitations like it may introduce bias and some useful information will be omitted from analysis.

### Data Snapshot

Consumerpreference data consist information about 73 respondents & their preferences about 7 attributes on scale of 1-Least Important to 5-Most Important.



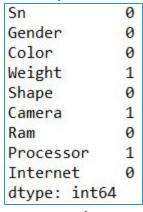
- If the amount of missing data is very small relative to the size of the data then case wise deletion may be the best strategy in order not to bias the analysis, however deleting available data points deprives the data of some amount of information.
- Basically, we need to decide how we're going to use our missing data, if at all, then either remove cases from our data or impute missing values before wiping out potentially useful data points from our data and proceed with our analysis.
- Single imputation can be done by replacing missing values with the mean /
  median / mode (or any other procedure) of the other values in the variable.
   In this tutorial we will primarily focus on performing single imputation .

Treating missing values in the variable 'Processor'

```
# Import the data
# Check number of missing values for each variable
```

```
consumer_pref = pd.read_csv("consumerpreference.csv")
consumer_pref.isnull().sum()
```

#### # Output



Our data has 3 missing values.

Here, we are calculating median of the values in variable 'Processor' using **median()** and replacing the **NaN** value with the median value. Now there are no missing values in variable

# Treating missing values in variable 'Processor'.
# Median imputation

consumer\_pref['Processor'].fillna((consumer\_pref['Processor'].median()

```
), inplace=True)
consumer_pref['Processor'].isnull().sum()
```

# Imputing missing value with forward fill

```
forward_fill = consumer_pref.fillna(method='ffill')
forward_fill.head(10)
```

# Output

	Sn	Gender	Color	Weight	Shape	Camera	Ram	Processor	Internet
0	1	M	3	3.0	4	3.0	3	3.0	3
1	2	M	3	4.0	3	3.0	4	4.0	4
2	3	M	1	2.0	2	5.0	4	4.0	4
3	4	M	1	1.0	2	5.0	5	5.0	5
4	5	F	4	3.0	4	5.0	5	5.0	5
5	6	F	4	3.0	4	4.0	3	3.0	3
6	7	F	4	4.0	4	1.0	2	2.0	2
7	8	F	5	4.0	5	1.0	1	1.0	1
8	9	M	1	1.0	1	1.0	4	3.0	4
9	10	M	5	1.0	4	3.0	2	3.0	2

- fillna() allows to fill the NaN values with the value specified or by predefined methods.
- method = 'ffill' allows to forward fill the NaN values
   i.e. it fill's the missing value with the previous value.

# Imputing missing value with backward fill

```
backward fill = consumer pref.fillna(method='bfill')
backward fill.head(10)
# Output
  Sn Gender Color Weight Shape
                          Camera
                                Ram
                                   Processor
                                            Internet
                 3.0
                                        3.0
                            3.0
                 4.0
                        3 3.0
                                        4.0
             1 2.0
                          5.0
                                        4.0
             1 1.0
                        2 5.0
                                        5.0
             4 3.0
                        4 5.0 5
                                        5.0
             4 3.0
                        4 4.0 3
                                        3.0
       F 4 4.0
F 5 4.0
                          1.0 2
                                        2.0
```

1.0 1

3.0

3.0

1

3.0

10

fillna() allows to fill the NaN values with the value specified or by predefined methods.

1.0

3.0

3.0

method = 'bfill' allows to backward fill the NaN values i.e. it fill's the missing value with the next value.

### Quick Recap

In this session, we learnt how to deal with different types of missing values. Here is the quick recap:

### Replacing missing values

- Blank fields in numeric & character column are replaced with NaN while importing data.
- fillna() is used to replace missing values with various methods.

### Recoding values to missing

- **isnull():** returns the logical matrix which indicates which elements are missing.
- **sum():** using this function we can calculate the count of NA's per column
- info(): check number of NA's per variable

# Excluding missing values from analysis

• dropna(): performs case wise deletion

### Imputing missing values

• Single imputation can be done by replacing missing values with the mean / median / mode (or any other procedure) of the other values in the variable