Code:-

```
import pandas as pd
import csv
import numpy as np
with open('/content/diabetes.csv') as obj:

    row_obj = csv.reader(obj)
    for row in row_obj:
        print(row)
```

Output:-

```
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']
['6', '148', '72', '35', '0', '33.6', '0.627', '50', '1']
['1', '85', '66', '29', '0', '26.6', '0.351', '31', '0']
['8', '183', '64', '0', '0', '23.3', '0.672', '32', '1']
['1', '89', '66', '23', '94', '28.1', '0.167', '21', '0']
['0', '137', '40', '35', '168', '43.1', '2.288', '33', '1']
['5', '116', '74', '0', '0', '25.6', '0.201', '30', '0']
['3', '78', '50', '32', '88', '31', '0.248', '26', '1']
['10', '115', '0', '0', '0', '35.3', '0.134', '29', '0']
['2', '197', '70', '45', '543', '30.5', '0.158', '53', '1']
['8', '125', '96', '0', '0', '0', '0.232', '54', '1']
['4', '110', '92', '0', '0', '37.6', '0.191', '30', '0']
['10', '168', '74', '0', '0', '38', '0.537', '34', '1']
['10', '139', '80', '0', '0', '27.1', '1.441', '57', '0']
['1', '189', '60', '23', '846', '30.1', '0.398', '59', '1']
['5', '166', '72', '19', '175', '25.8', '0.587', '51', '1']
['7', '100', '0', '0', '0', '30', '0.484', '32', '1']
['0', '118', '84', '47', '230', '45.8', '0.551', '31', '1']
['7', '107', '74', '0', '0', '29.6', '0.254', '31', '1']
['1', '103', '30', '38', '83', '43.3', '0.183', '33', '0']
['1', '115', '70', '30', '96', '34.6', '0.529', '32', '1']
['3', '126', '88', '41', '235', '39.3', '0.704', '27', '0']
['8', '99', '84', '0', '0', '35.4', '0.388', '50', '0']
['7', '196', '90', '0', '0', '39.8', '0.451', '41', '1']
['9', '119', '80', '35', '0', '29', '0.263', '29', '1']
['11', '143', '94', '33', '146', '36.6', '0.254', '51', '1']
['10', '125', '70', '26', '115', '31.1', '0.205', '41', '1']
['7', '147', '76', '0', '0', '39.4', '0.257', '43', '1']
['1', '97', '66', '15', '140', '23.2', '0.487', '22', '0']
['13', '145', '82', '19', '110', '22.2', '0.245', '57', '0']
['5', '117', '92', '0', '0', '34.1', '0.337', '38', '0']
['5', '109', '75', '26', '0', '36', '0.546', '60', '0']
['3', '158', '76', '36', '245', '31.6', '0.851', '28', '1']
['3', '88', '58', '11', '54', '24.8', '0.267', '22', '0']
['6', '92', '92', '0', '0', '19.9', '0.188', '28', '0']
['10', '122', '78', '31', '0', '27.6', '0.512', '45', '0']
['4', '103', '60', '33', '192', '24', '0.966', '33', '0']
['11', '138', '76', '0', '0', '33.2', '0.42', '35', '0']
['9', '102', '76', '37', '0', '32.9', '0.665', '46', '1']
['2', '90', '68', '42', '0', '38.2', '0.503', '27', '1']
['4', '111', '72', '47', '207', '37.1', '1.39', '56', '1']
['3', '180', '64', '25', '70', '34', '0.271', '26', '0']
['7', '133', '84', '0', '0', '40.2', '0.696', '37', '0']
['7', '106', '92', '18', '0', '22.7', '0.235', '48', '0']
['9', '171', '110', '24', '240', '45.4', '0.721', '54', '1']
['7', '159', '64', '0', '0', '27.4', '0.294', '40', '0']
['0', '180', '66', '39', '0', '42', '1.893', '25', '1']
```

Code:-

```
import csv
with open('/content/diabetes.csv') as file:
    reader = csv.DictReader(
    file, fieldnames=['DiabetesPedigreeFunction', 'Age', 'Glucose'])
    for row in reader:
        print(row)
```

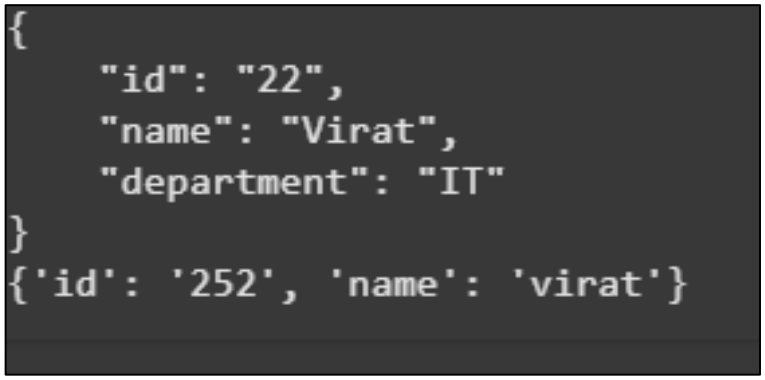
Output:-

```
{'DiabetesPedigreeFunction': 'Pregnancies', 'Age': 'Glucose', 'Glucose': 'BloodPressure', None: ['SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age',
{'DiabetesPedigreeFunction': '6', 'Age': '148', 'Glucose': '72', None: ['35', '0', '33.6', '0.627', '50', '1']}
{'DiabetesPedigreeFunction': '1', 'Age': '85', 'Glucose': '66', None: ['29', '0', '26.6', '0.351', '31', '0']}
{'DiabetesPedigreeFunction': '8', 'Age': '183', 'Glucose': '64', None: ['0', '0', '23.3', '0.672', '32', '1']}
{'DiabetesPedigreeFunction': '1', 'Age': '89', 'Glucose': '66', None: ['23', '94', '28.1', '0.167', '21', '0']}
{'DiabetesPedigreeFunction': '0', 'Age': '137', 'Glucose': '40', None: ['35', '168', '43.1', '2.288', '33', '1']}
{'DiabetesPedigreeFunction': '5', 'Age': '116', 'Glucose': '74', None: ['0', '0', '25.6', '0.201', '30', '0']}
{'DiabetesPedigreeFunction': '3', 'Age': '78', 'Glucose': '50', None: ['32', '88', '31', '0.248', '26', '1']}
{'DiabetesPedigreeFunction': '10', 'Age': '115', 'Glucose': '0', None: ['0', '0', '35.3', '0.134', '29', '0']}
{'DiabetesPedigreeFunction': '2', 'Age': '197', 'Glucose': '70', None: ['45', '543', '30.5', '0.158', '53', '1']}
{'DiabetesPedigreeFunction': '8', 'Age': '125', 'Glucose': '96', None: ['0', '0', '0', '0.232', '54', '1']}
{'DiabetesPedigreeFunction': '4', 'Age': '110', 'Glucose': '92', None: ['0', '0', '37.6', '0.191', '30', '0']}
{'DiabetesPedigreeFunction': '10', 'Age': '168', 'Glucose': '74', None: ['0', '0', '38', '0.537', '34', '1']}
{'DiabetesPedigreeFunction': '10', 'Age': '139', 'Glucose': '80', None: ['0', '0', '27.1', '1.441', '57', '0']}
{'DiabetesPedigreeFunction': '1', 'Age': '189', 'Glucose': '60', None: ['23', '846', '30.1', '0.398', '59', '1']}
{'DiabetesPedigreeFunction': '5', 'Age': '166', 'Glucose': '72', None: ['19', '175', '25.8', '0.587', '51', '1']}
{'DiabetesPedigreeFunction': '7', 'Age': '100', 'Glucose': '0', None: ['0', '0', '30', '0.484', '32', '1']}
{'DiabetesPedigreeFunction': '0', 'Age': '118', 'Glucose': '84', None: ['47', '230', '45.8', '0.551', '31', '1']}
{'DiabetesPedigreeFunction': '7', 'Age': '107', 'Glucose': '74', None: ['0', '0', '29.6', '0.254', '31', '1']}
{'DiabetesPedigreeFunction': '1', 'Age': '103', 'Glucose': '30', None: ['38', '83', '43.3', '0.183', '33', '0']}
{'DiabetesPedigreeFunction': '1', 'Age': '115', 'Glucose': '70', None: ['30', '96', '34.6', '0.529', '32', '1']}
{'DiabetesPedigreeFunction': '3', 'Age': '126', 'Glucose': '88', None: ['41', '235', '39.3', '0.704', '27', '0']}
{'DiabetesPedigreeFunction': '8', 'Age': '99', 'Glucose': '84', None: ['0', '0', '35.4', '0.388', '50', '0']}
{'DiabetesPedigreeFunction': '7', 'Age': '196', 'Glucose': '90', None: ['0', '0', '39.8', '0.451', '41', '1']}
{'DiabetesPedigreeFunction': '9', 'Age': '119', 'Glucose': '80', None: ['35', '0', '29', '0.263', '29', '1']}
{'DiabetesPedigreeFunction': '11', 'Age': '143', 'Glucose': '94', None: ['33', '146', '36.6', '0.254', '51', '1']}
{'DiabetesPedigreeFunction': '10', 'Age': '125', 'Glucose': '70', None: ['26', '115', '31.1', '0.205', '41', '1']}
{'DiabetesPedigreeFunction': '7', 'Age': '147', 'Glucose': '76', None: ['0', '0', '39.4', '0.257', '43', '1']}
{'DiabetesPedigreeFunction': '1', 'Age': '97', 'Glucose': '66', None: ['15', '140', '23.2', '0.487', '22', '0']}
{'DiabetesPedigreeFunction': '13', 'Age': '145', 'Glucose': '82', None: ['19', '110', '22.2', '0.245', '57', '0']}
{'DiabetesPedigreeFunction': '5', 'Age': '117', 'Glucose': '92', None: ['0', '0', '34.1', '0.337', '38', '0']}
{'DiabetesPedigreeFunction': '5', 'Age': '109', 'Glucose': '75', None: ['26', '0', '36', '0.546', '60', '0']}
{'DiabetesPedigreeFunction': '3', 'Age': '158', 'Glucose': '76', None: ['36', '245', '31.6', '0.851', '28', '1']}
{'DiabetesPedigreeFunction': '3', 'Age': '88', 'Glucose': '58', None: ['11', '54', '24.8', '0.267', '22', '0']}
{'DiabetesPedigreeFunction': '6', 'Age': '92', 'Glucose': '92', None: ['0', '0', '19.9', '0.188', '28', '0']}
{'DiabetesPedigreeFunction': '10', 'Age': '122', 'Glucose': '78', None: ['31', '0', '27.6', '0.512', '45', '0']}
{'DiabetesPedigreeFunction': '4', 'Age': '103', 'Glucose': '60', None: ['33', '192', '24', '0.966', '33', '0']}
{'DiabetesPedigreeFunction': '11', 'Age': '138', 'Glucose': '76', None: ['0', '0', '33.2', '0.42', '35', '0']}
{'DiabetesPedigreeFunction': '9', 'Age': '102', 'Glucose': '76', None: ['37', '0', '32.9', '0.665', '46', '1']}
{'DiabetesPedigreeFunction': '2', 'Age': '90', 'Glucose': '68', None: ['42', '0', '38.2', '0.503', '27', '1']}
{'DiabetesPedigreeFunction': '4', 'Age': '111', 'Glucose': '72', None: ['47', '207', '37.1', '1.39', '56', '1']}
```

Code:-

```
import json
dictionary ={"id":"22","name":"Virat","department":"IT"}
#dictionary
json_object = json.dumps(dictionary,indent=4)
print(json_object)
{
   "id":"252",
   "name":"virat"
}
```

Output:-



Code:-

```
import xml
data = pd.read_xml('/content/sample_data/sample1.xml')
data
```

Output:-

	name	age	email	title	author	year
0	John Doe	30.0	john.doe@example.com	None	None	NaN
1	Jane Smith	25.0	jane.smith@example.com	None	None	NaN
2	None	NaN	None	The Adventure Begins	Robert Johnson	2022.0

Code:-

```
import pandas as pd
import numpy as np
import openpyxl
file="/content/Groceries_dataset.xlsx"
df=pd.read_excel(file)
df
```

Output:-

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	2015-05-01 00:00:00	whole milk
2	2300	19-09-2015	pip fruit
3	1187	2015-12-12 00:00:00	other vegetables
4	3037	2015-01-02 00:00:00	whole milk
...	...	...	...
38760	4471	2014-08-10 00:00:00	sliced cheese
38761	2022	23-02-2014	candy
38762	1097	16-04-2014	cake bar
38763	1510	2014-03-12 00:00:00	fruit/vegetable juice
38764	1521	26-12-2014	cat food

38765 rows × 3 columns

a) Get the data types of the given excel data.

Code:-

```
import pandas as pd
import numpy as np
import openpyxl
file="/content/Groceries_dataset.xlsx"
df=pd.read_excel(file)
df
data_types = df.dtypes
print("Data Types:")
print(data_types)
```

Output:-

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	2015-05-01 00:00:00	whole milk
2	2300	19-09-2015	pip fruit
3	1187	2015-12-12 00:00:00	other vegetables
4	3037	2015-01-02 00:00:00	whole milk
...	...	...	...
38760	4471	2014-08-10 00:00:00	sliced cheese
38761	2022	23-02-2014	candy
38762	1097	16-04-2014	cake bar
38763	1510	2014-03-12 00:00:00	fruit/vegetable juice
38764	1521	26-12-2014	cat food

38765 rows × 3 columns

```
Data Types:
Member_number      int64
Date              object
itemDescription   object
dtype: object
```

b) Display the last ten rows.

Code:-

```
last_ten_rows = df.tail(10)
print("\nLast Ten Rows:")
Last_ten_rows
```

Output:-

Last Ten Rows:

	Member_number	Date	itemDescription
38755	4586	26-09-2014	bottled water
38756	1987	29-10-2014	fruit/vegetable juice
38757	4376	2014-07-12 00:00:00	rolls/buns
38758	2511	18-06-2014	long life bakery product
38759	3364	2014-06-05 00:00:00	oil
38760	4471	2014-08-10 00:00:00	sliced cheese
38761	2022	23-02-2014	candy
38762	1097	16-04-2014	cake bar
38763	1510	2014-03-12 00:00:00	fruit/vegetable juice
38764	1521	26-12-2014	cat food

c) Insert column in the sixth position of the said excel sheet and fill it with NaN values.

Code:-

```
column_name = 'New Column'
df.insert(3, column_name, np.nan)
df
```

Output:-

	Member_number	Date	itemDescription	New Column
0	1808	21-07-2015	tropical fruit	NaN
1	2552	2015-05-01 00:00:00	whole milk	NaN
2	2300	19-09-2015	pip fruit	NaN
3	1187	2015-12-12 00:00:00	other vegetables	NaN
4	3037	2015-01-02 00:00:00	whole milk	NaN
...	...	...	...	...
38760	4471	2014-08-10 00:00:00	sliced cheese	NaN
38761	2022	23-02-2014	candy	NaN
38762	1097	16-04-2014	cake bar	NaN
38763	1510	2014-03-12 00:00:00	fruit/vegetable juice	NaN
38764	1521	26-12-2014	cat food	NaN

38765 rows × 4 columns

Code:-

```
!pip install pdfminer.six
from pdfminer.high_level import extract_text
def parse_pdf(file_path):
  text = extract_text(file_path)
  return text
pdf_file = '/content/Sample-pdf.pdf'
parsed_text = parse_pdf(pdf_file)
print(parsed_text)
```

Output:-

```
Requirement already satisfied: pdfminer.six in /usr/local/lib/python3.11/dist-packages (20240706)
Requirement already satisfied: charset-normalizer>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from pdfminer.six) (3.4.1)
Requirement already satisfied: cryptography>=36.0.0 in /usr/local/lib/python3.11/dist-packages (from pdfminer.six) (43.0.3)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.11/dist-packages (from cryptography>=36.0.0->pdfminer.six) (1.17.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.12->cryptography>=36.0.0->pdfminer.six) (2.22)
Sample PDF Document

Robert Maron
Grzegorz Grudzi´nski

February 20, 1999

2

Contents

1 Template

.

1.1 How to compile a .tex file to a .pdf file . .
1.1.1 Tools
. . . . . .
1.1.2 How to use the tools
.

. . .
.
. .
1.2.1 The main document . .
1.2.2 Chapters
1.2.3

. . . . .
Spell-checking . .

1.2 How to write a document
```

Code:-

```
import pandas as pd
excel_file = "/content/file_example_XLS_50.xls"
data_frame=pd.read_excel(excel_file)
tables=data_frame.values.tolist()
for table in tables:
  print(table)
```

Output:-

```
[1, 'Dulce', 'Abril', 'Female', 'United States', 32, '15/10/2017', 1562]
[2, 'Mara', 'Hashimoto', 'Female', 'Great Britain', 25, '16/08/2016', 1582]
[3, 'Philip', 'Gent', 'Male', 'France', 36, '21/05/2015', 2587]
[4, 'Kathleen', 'Hanner', 'Female', 'United States', 25, '15/10/2017', 3549]
[5, 'Nereida', 'Magwood', 'Female', 'United States', 58, '16/08/2016', 2468]
[6, 'Gaston', 'Brumm', 'Male', 'United States', 24, '21/05/2015', 2554]
[7, 'Etta', 'Hurn', 'Female', 'Great Britain', 56, '15/10/2017', 3598]
[8, 'Earlean', 'Melgar', 'Female', 'United States', 27, '16/08/2016', 2456]
[9, 'Vincenza', 'Weiland', 'Female', 'United States', 40, '21/05/2015', 6548]
[10, 'Fallon', 'Winward', 'Female', 'Great Britain', 28, '16/08/2016', 5486]
[11, 'Arcelia', 'Bouska', 'Female', 'Great Britain', 39, '21/05/2015', 1258]
[12, 'Franklyn', 'Unknow', 'Male', 'France', 38, '15/10/2017', 2579]
[13, 'Sherron', 'Ascencio', 'Female', 'Great Britain', 32, '16/08/2016', 3256]
[14, 'Marcel', 'Zabriskie', 'Male', 'Great Britain', 26, '21/05/2015', 2587]
[15, 'Kina', 'Hazelton', 'Female', 'Great Britain', 31, '16/08/2016', 3259]
[16, 'Shavonne', 'Pia', 'Female', 'France', 24, '21/05/2015', 1546]
[17, 'Shavon', 'Benito', 'Female', 'France', 39, '15/10/2017', 3579]
[18, 'Lauralee', 'Perrine', 'Female', 'Great Britain', 28, '16/08/2016', 6597]
[19, 'Loreta', 'Curren', 'Female', 'France', 26, '21/05/2015', 9654]
[20, 'Teresa', 'Strawn', 'Female', 'France', 46, '21/05/2015', 3569]
[21, 'Belinda', 'Partain', 'Female', 'United States', 37, '15/10/2017', 2564]
[22, 'Holly', 'Eudy', 'Female', 'United States', 52, '16/08/2016', 8561]
[23, 'Many', 'Cuccia', 'Female', 'Great Britain', 46, '21/05/2015', 5489]
[24, 'Libbie', 'Dalby', 'Female', 'France', 42, '21/05/2015', 5489]
[25, 'Lester', 'Prothro', 'Male', 'France', 21, '15/10/2017', 6574]
[26, 'Marvel', 'Hail', 'Female', 'Great Britain', 28, '16/08/2016', 5555]
[27, 'Angelyn', 'Vong', 'Female', 'United States', 29, '21/05/2015', 6125]
[28, 'Francesca', 'Beaudreau', 'Female', 'France', 23, '15/10/2017', 5412]
[29, 'Garth', 'Gangi', 'Male', 'United States', 41, '16/08/2016', 3256]
[30, 'Carla', 'Trumbull', 'Female', 'Great Britain', 28, '21/05/2015', 3264]
[31, 'Veta', 'Muntz', 'Female', 'Great Britain', 37, '15/10/2017', 4569]
[32, 'Stasia', 'Becker', 'Female', 'Great Britain', 34, '16/08/2016', 7521]
[33, 'Jona', 'Grindle', 'Female', 'Great Britain', 26, '21/05/2015', 6458]
[34, 'Judie', 'Claywell', 'Female', 'France', 35, '16/08/2016', 7569]
[35, 'Dewitt', 'Borger', 'Male', 'United States', 36, '21/05/2015', 8514]
[36, 'Nena', 'Hacker', 'Female', 'United States', 29, '15/10/2017', 8563]
[37, 'Kelsie', 'Wachtel', 'Female', 'France', 27, '16/08/2016', 8642]
[38, 'Sau', 'Pfau', 'Female', 'United States', 25, '21/05/2015', 9536]
[39, 'Shanice', 'Mccrystal', 'Female', 'United States', 36, '21/05/2015', 2567]
[40, 'Chase', 'Karner', 'Male', 'United States', 37, '15/10/2017', 2154]
[41, 'Tommie', 'Underdahl', 'Male', 'United States', 26, '16/08/2016', 3265]
[42, 'Dorcas', 'Darity', 'Female', 'United States', 37, '21/05/2015', 8765]
[43, 'Angel', 'Sanor', 'Male', 'France', 24, '15/10/2017', 3259]
[44, 'Willodean', 'Harn', 'Female', 'United States', 39, '16/08/2016', 3567]
[45, 'Weston', 'Martina', 'Male', 'United States', 26, '21/05/2015', 6540]
[46, 'Roma', 'Lafollette', 'Female', 'United States', 34, '15/10/2017', 2654]
[47, 'Felisa', 'Cail', 'Female', 'United States', 28, '16/08/2016', 6525]
```

Code:-

```
import sqlite3
conn = sqlite3.connect('gajeara.db')
cursor = conn.cursor()
create_table_sql = '''
CREATE TABLE IF NOT EXISTS my_tables (
  id INTEGER PRIMARY KEY,
  name TEXT,
  age INTEGER
)
'''
cursor.execute(create_table_sql)
insert_data_sql = '''
INSERT INTO my_tables (name, age) VALUES
        ('Omkar', 25),
        ('Malay', 30),
        ('RajShekhar', 25),
        ('Manjunath', 21),
        ('Sandhya', 32),
        ('Aditya', 46)

'''
cursor.execute(insert_data_sql)
conn.commit()
select_data_sql = 'SELECT * FROM my_tables'
cursor.execute(select_data_sql)
result = cursor.fetchall()

for row in result:
  print(row)

conn.close()
```

Output:-

```
(1, 'Omkar', 25)
(2, 'Malay', 30)
(3, 'RajShekhar', 25)
(4, 'Manjunath', 21)
(5, 'Sandhya', 32)
(6, 'Aditya', 46)
```

a) Check duplicates and missing data.

Code:-

```
import re
import pandas as pd
excel_file = '/content/file_example_XLS_50.xls'
data_frame = pd.read_excel(excel_file)
duplicates = data_frame.duplicated()
if duplicates.any() :
  print("Duplicates found: ")
  print(data_frame[duplicates])
  data_frame = data_frame.drop_duplicates()
print("Duplicates removed.")
missing_data = data_frame.isnull()
if missing_data.any().any():
  print("Missing data found:")
  print(missing_data)
missing_data
```

Output:-

```
Duplicates removed.
```

	0	First Name	Last Name	Gender	Country	Age	Date	Id
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False	False

b) Eliminate Mismatches.

Code:-

```
data_frame['Country'] = data_frame['Country'].str.upper()
data_frame['Age'] = data_frame['Age'].replace({'MismatchedValue':
'CorrectValue'})
data_frame
```

Output:-

	0	First Name	Last Name	Gender	Country	Age	Date	Id
0	1	Dulce	Abril	Female	UNITED STATES	32	15/10/2017	1562
1	2	Mara	Hashimoto	Female	GREAT BRITAIN	25	16/08/2016	1582
2	3	Philip	Gent	Male	FRANCE	36	21/05/2015	2587
3	4	Kathleen	Hanner	Female	UNITED STATES	25	15/10/2017	3549
4	5	Nereida	Magwood	Female	UNITED STATES	58	16/08/2016	2468
5	6	Gaston	Brumm	Male	UNITED STATES	24	21/05/2015	2554
6	7	Etta	Hurn	Female	GREAT BRITAIN	56	15/10/2017	3598
7	8	Earlean	Melgar	Female	UNITED STATES	27	16/08/2016	2456
8	9	Vincenza	Weiland	Female	UNITED STATES	40	21/05/2015	6548
9	10	Fallon	Winward	Female	GREAT BRITAIN	28	16/08/2016	5486
10	11	Arcelia	Bouska	Female	GREAT BRITAIN	39	21/05/2015	1258
11	12	Franklyn	Unknow	Male	FRANCE	38	15/10/2017	2579
12	13	Sherron	Ascencio	Female	GREAT BRITAIN	32	16/08/2016	3256
13	14	Marcel	Zabriskie	Male	GREAT BRITAIN	26	21/05/2015	2587
14	15	Kina	Hazelton	Female	GREAT BRITAIN	31	16/08/2016	3259
15	16	Shavonne	Pia	Female	FRANCE	24	21/05/2015	1546
16	17	Shavon	Benito	Female	FRANCE	39	15/10/2017	3579
17	18	Lauralee	Perrine	Female	GREAT BRITAIN	28	16/08/2016	6597
18	19	Loreta	Curren	Female	FRANCE	26	21/05/2015	9654
19	20	Teresa	Strawn	Female	FRANCE	46	21/05/2015	3569
20	21	Belinda	Partain	Female	UNITED STATES	37	15/10/2017	2564
21	22	Holly	Eudy	Female	UNITED STATES	52	16/08/2016	8561
22	23	Many	Cuccia	Female	GREAT BRITAIN	46	21/05/2015	5489
23	24	Libbie	Dalby	Female	FRANCE	42	21/05/2015	5489
24	25	Lester	Prothro	Male	FRANCE	21	15/10/2017	6574
25	26	Marvel	Hail	Female	GREAT BRITAIN	28	16/08/2016	5555
26	27	Angelyn	Vong	Female	UNITED STATES	29	21/05/2015	6125

c) Cleans line breaks, spaces, and special characters.

Code:-

```
data_frame = data_frame.replace(r'\r|\n', '', regex=True)
data_frame = data_frame.applymap(lambda x: re.sub(r'\s+', '', str(x))
if pd.notnull(x) else x)
data_frame = data_frame.applymap(lambda x: re.sub(r'[^a-zA-Z0-9\s]', '', str(x))
if pd.notnull(x) else x)
print("Cleanned data :")
print(data_frame)
```

Output:-

```
Cleanned data :
     0 First Name   Last Name  Gender       Country Age      Date    Id
0    1       Dulce       Abril  Female  UNITEDSTATES  32  15102017  1562
1    2        Mara   Hashimoto  Female  GREATBRITAIN  25  16082016  1582
2    3      Philip        Gent    Male        FRANCE  36  21052015  2587
3    4    Kathleen      Hanner  Female  UNITEDSTATES  25  15102017  3549
4    5     Nereida     Magwood  Female  UNITEDSTATES  58  16082016  2468
5    6      Gaston       Brumm    Male  UNITEDSTATES  24  21052015  2554
6    7        Etta        Hurn  Female  GREATBRITAIN  56  15102017  3598
7    8     Earlean      Melgar  Female  UNITEDSTATES  27  16082016  2456
8    9    Vincenza     Weiland  Female  UNITEDSTATES  40  21052015  6548
9   10      Fallon     Winward  Female  GREATBRITAIN  28  16082016  5486
10  11     Arcelia      Bouska  Female  GREATBRITAIN  39  21052015  1258
11  12    Franklyn      Unknow    Male        FRANCE  38  15102017  2579
12  13     Sherron    Ascencio  Female  GREATBRITAIN  32  16082016  3256
13  14      Marcel   Zabriskie    Male  GREATBRITAIN  26  21052015  2587
14  15        Kina    Hazelton  Female  GREATBRITAIN  31  16082016  3259
15  16    Shavonne         Pia  Female        FRANCE  24  21052015  1546
16  17      Shavon      Benito  Female        FRANCE  39  15102017  3579
17  18    Lauralee     Perrine  Female  GREATBRITAIN  28  16082016  6597
18  19      Loreta      Curren  Female        FRANCE  26  21052015  9654
19  20      Teresa      Strawn  Female        FRANCE  46  21052015  3569
20  21     Belinda     Partain  Female  UNITEDSTATES  37  15102017  2564
21  22       Holly        Eudy  Female  UNITEDSTATES  52  16082016  8561
22  23        Many      Cuccia  Female  GREATBRITAIN  46  21052015  5489
23  24      Libbie       Dalby  Female        FRANCE  42  21052015  5489
24  25      Lester     Prothro    Male        FRANCE  21  15102017  6574
25  26      Marvel        Hail  Female  GREATBRITAIN  28  16082016  5555
26  27     Angelyn        Vong  Female  UNITEDSTATES  29  21052015  6125
27  28   Francesca    Beaudreau  Female        FRANCE  23  15102017  5412
28  29       Garth       Gangi    Male  UNITEDSTATES  41  16082016  3256
29  30       Carla     Trumbull  Female  GREATBRITAIN  28  21052015  3264
30  31        Veta       Muntz  Female  GREATBRITAIN  37  15102017  4569
31  32      Stasia      Becker  Female  GREATBRITAIN  34  16082016  7521
32  33        Jona      Grindle  Female  GREATBRITAIN  26  21052015  6458
33  34       Judie     Claywell  Female        FRANCE  35  16082016  7569
34  35      Dewitt       Borger    Male  UNITEDSTATES  36  21052015  8514
35  36        Nena      Hacker  Female  UNITEDSTATES  29  15102017  8563
36  37      Kelsie      Wachtel  Female        FRANCE  27  16082016  8642
37  38         Sau        Pfau  Female  UNITEDSTATES  25  21052015  9536
38  39     Shanice    Mccrystal  Female  UNITEDSTATES  36  21052015  2567
39  40       Chase       Karner    Male  UNITEDSTATES  37  15102017  2154
40  41      Tommie    Underdahl    Male  UNITEDSTATES  26  16082016  3265
41  42      Dorcas       Darity  Female  UNITEDSTATES  37  21052015  8765
42  43       Angel        Sanor    Male        FRANCE  24  15102017  3259
43  44   Willodean         Harn  Female  UNITEDSTATES  39  16082016  3567
44  45      Weston      Martina    Male  UNITEDSTATES  26  21052015  6540
45  46        Roma   Lafollette  Female  UNITEDSTATES  34  15102017  2654
46  47      Felisa         Cail  Female  UNITEDSTATES  28  16082016  6525
47  48    Demetria        Abbey  Female  UNITEDSTATES  32  21052015  3265
48  49      Jeromy         Danz    Male  UNITEDSTATES  39  15102017  3265
```

Code:-

```
!pip install agate
import agate
table = agate.Table.from_csv('/content/SOCR-HeightWeight.csv')
print(table)
print('Column Name:', table.column_names)
print("Number of Row:", len(table.rows))
total_Height = table.aggregate(agate.Sum('Height(Inches)'))
average_Height = table.aggregate(agate.Mean('Height(Inches)'))
total_Weight = table.aggregate(agate.Sum('Weight(Pounds)'))
average_Weight = table.aggregate(agate.Mean('Weight(Pounds)'))
print("Total Height:", total_Height)
print("Average Height:", average_Height)
print("Total Weight:", total_Weight)
print("Average Weight:", average_Weight)
total_Height = table.aggregate(agate.Sum('Height(Inches)'))
average_Height = table.aggregate(agate.Mean('Height(Inches)'))
total_Weight = table.aggregate(agate.Sum('Weight(Pounds)'))
average_Weight = table.aggregate(agate.Mean('Weight(Pounds)'))
print("Total Height:", total_Height)
print("Average Height:", average_Height)
print("Total Weight:", total_Weight)
print("Average Weight:", average_Weight)
```

Output:-

```
| column          | data_type |
| --------------- | --------- |
| Index           | Number    |
| Height(Inches)  | Number    |
| Weight(Pounds)  | Number    |
```

```
Column Name: ('\ufeffIndex', 'Height(Inches)', 'Weight(Pounds)')
Number of Row: 25000
```

```
Total Height: 1699827.83992
Average Height: 67.9931135968
Total Weight: 3176985.52902
Average Weight: 127.0794211608
```

```
Pearson Correlation Coefficient (Height vs Weight): 0.502858520602844
P-value:  0.0
```