CS23334-FUNDAMENTALS OF DATA SCIENCE

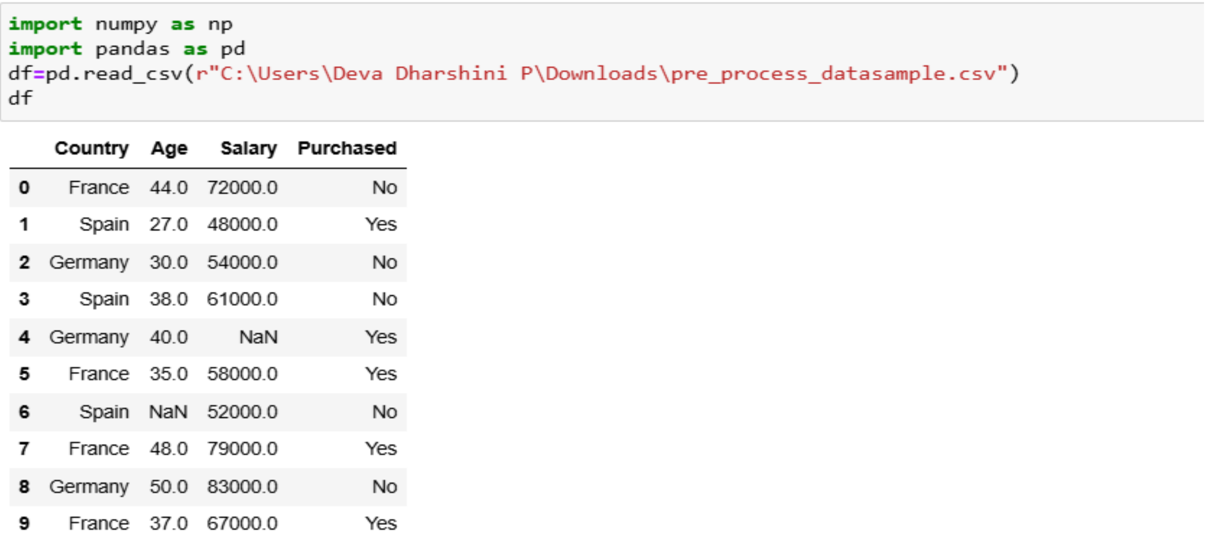
DEVA DHARSHINI P(240701107)

3.a) PANDAS LIBRARY-HANDLING MISSING VALUES

Aim:

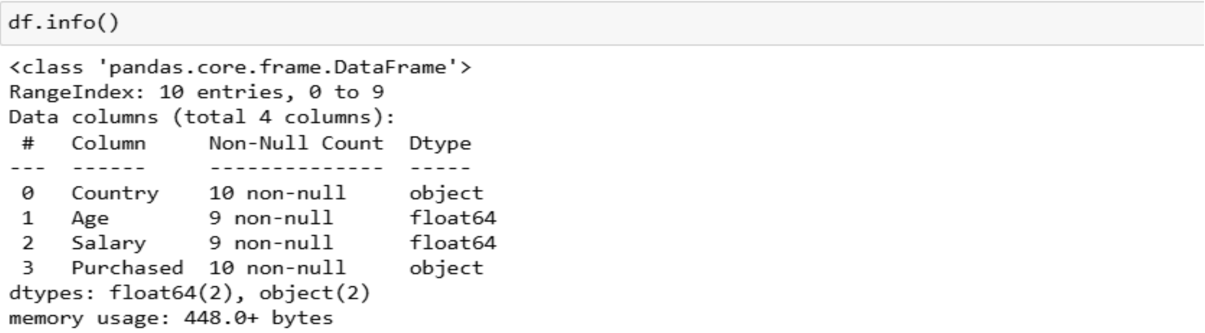
To understand and perform data preprocessing steps such as cleaning, handling missing values, encoding, and normalization to prepare raw data for analysis.

Code:



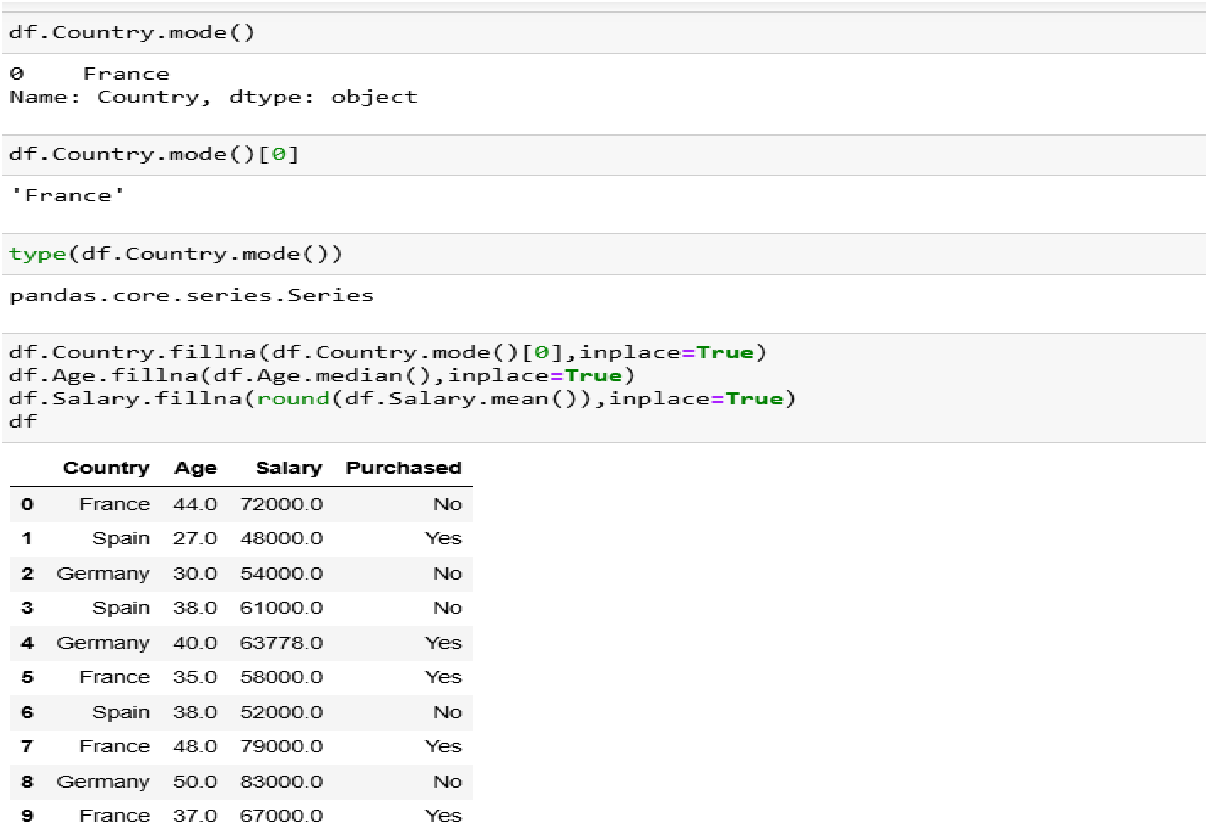
```
import numpy as np
import pandas as pd
df=pd.read_csv(r"C:\Users\Deva Dharshini P\Downloads\pre_process_datasample.csv")
df
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes
5	France	35.0	58000.0	Yes
6	Spain	NaN	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Country    10 non-null     object 
 1   Age        9 non-null      float64
 2   Salary     9 non-null      float64
 3   Purchased  10 non-null     object 
dtypes: float64(2), object(2)
memory usage: 448.0+ bytes
```



```
df.Country.mode()
```

```
0     France
Name: Country, dtype: object
```

```
df.Country.mode()[0]
```

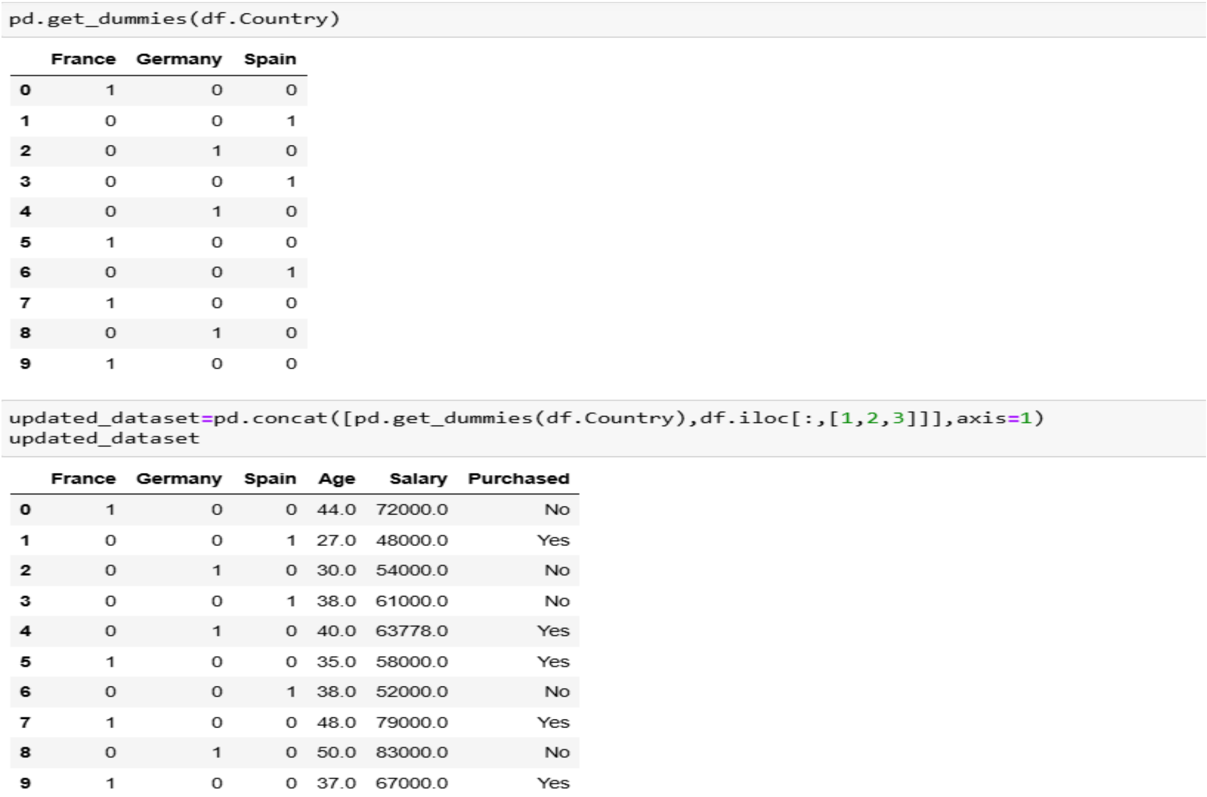
```
'France'
```

```
type(df.Country.mode())
```

```
pandas.core.series.Series
```

```
df.Country.fillna(df.Country.mode()[0],inplace=True)
df.Age.fillna(df.Age.median(),inplace=True)
df.Salary.fillna(round(df.Salary.mean()),inplace=True)
df
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	63778.0	Yes
5	France	35.0	58000.0	Yes
6	Spain	38.0	52000.0	No
7	France	48.0	79000.0	Yes
8	Germany	50.0	83000.0	No
9	France	37.0	67000.0	Yes

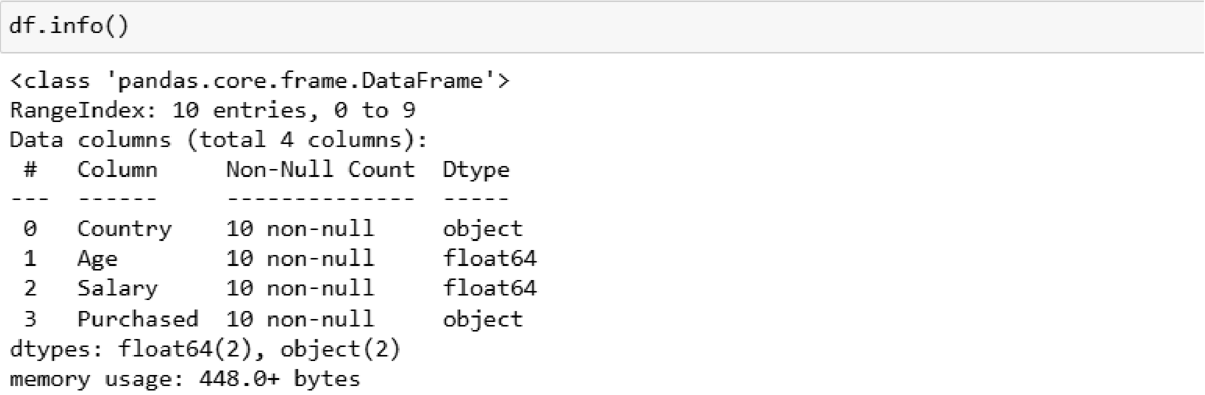


```
pd.get_dummies(df.Country)
```

	France	Germany	Spain
0	1	0	0
1	0	0	1
2	0	1	0
3	0	0	1
4	0	1	0
5	1	0	0
6	0	0	1
7	1	0	0
8	0	1	0
9	1	0	0

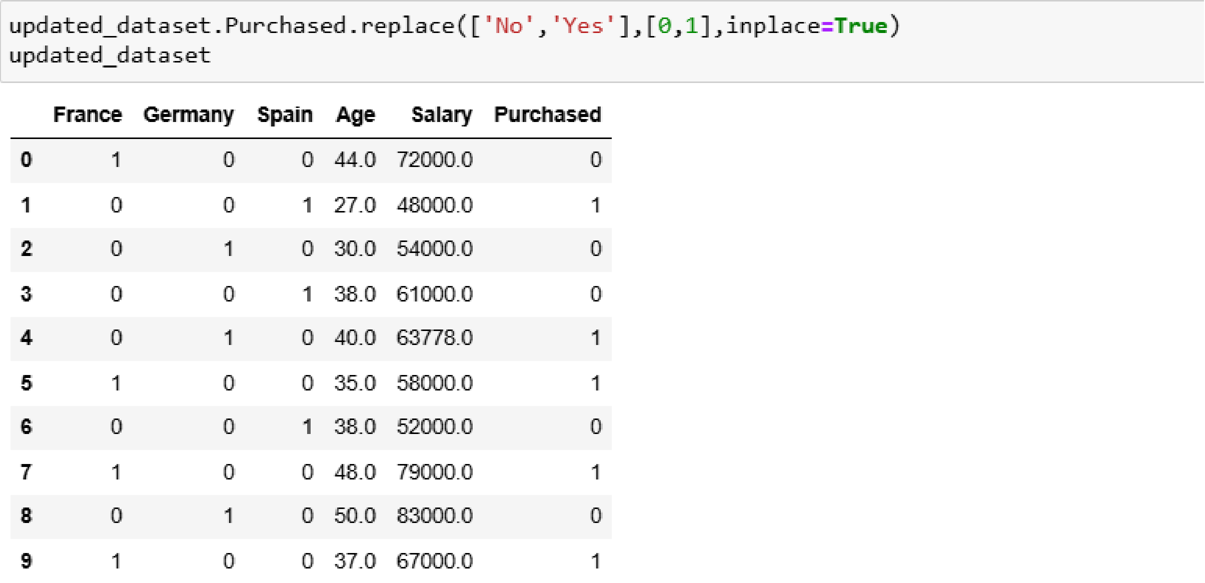
```
updated_dataset=pd.concat([pd.get_dummies(df.Country),df.iloc[:,[1,2,3]]],axis=1)
updated_dataset
```

	France	Germany	Spain	Age	Salary	Purchased
0	1	0	0	44.0	72000.0	No
1	0	0	1	27.0	48000.0	Yes
2	0	1	0	30.0	54000.0	No
3	0	0	1	38.0	61000.0	No
4	0	1	0	40.0	63778.0	Yes
5	1	0	0	35.0	58000.0	Yes
6	0	0	1	38.0	52000.0	No
7	1	0	0	48.0	79000.0	Yes
8	0	1	0	50.0	83000.0	No
9	1	0	0	37.0	67000.0	Yes



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Country    10 non-null     object
 1   Age        10 non-null     float64
 2   Salary     10 non-null     float64
 3   Purchased  10 non-null     object
dtypes: float64(2), object(2)
memory usage: 448.0+ bytes
```



```
updated_dataset.Purchased.replace(['No','Yes'],[0,1],inplace=True)
updated_dataset
```

	France	Germany	Spain	Age	Salary	Purchased
0	1	0	0	44.0	72000.0	0
1	0	0	1	27.0	48000.0	1
2	0	1	0	30.0	54000.0	0
3	0	0	1	38.0	61000.0	0
4	0	1	0	40.0	63778.0	1
5	1	0	0	35.0	58000.0	1
6	0	0	1	38.0	52000.0	0
7	1	0	0	48.0	79000.0	1
8	0	1	0	50.0	83000.0	0
9	1	0	0	37.0	67000.0	1

Result:
The raw data was successfully preprocessed by cleaning and transforming it into a suitable format for further data analysis and model building.

3.b) PANDAS LIBRARY-DATA PREPROCESSING

Aim:

To perform data preprocessing using the Pandas library for cleaning, handling missing values, and preparing data for analysis.

Code:

```
import numpy as np
import pandas as pd
df=pd.read_csv(r"C:\Users\Deva Dharshini P\Downloads\Hotel_Dataset.csv")
df
```

	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary	Age_Group.1
0	1	20-25	4	Ibis	veg	1300	2	40000	20-25
1	2	30-35	5	LemonTree	Non-Veg	2000	3	59000	30-35
2	3	25-30	6	RedFox	Veg	1322	2	30000	25-30
3	4	20-25	-1	LemonTree	Veg	1234	2	120000	20-25
4	5	35+	3	Ibis	Vegetarian	989	2	45000	35+
5	6	35+	3	Ibys	Non-Veg	1909	2	122220	35+
6	7	35+	4	RedFox	Vegetarian	1000	-1	21122	35+
7	8	20-25	7	LemonTree	Veg	2999	-10	345673	20-25
8	9	25-30	2	Ibis	Non-Veg	3456	3	-99999	25-30
9	9	25-30	2	Ibis	Non-Veg	3456	3	-99999	25-30
10	10	30-35	5	RedFox	non-Veg	-6755	4	87777	30-35

```
df.duplicated()
```

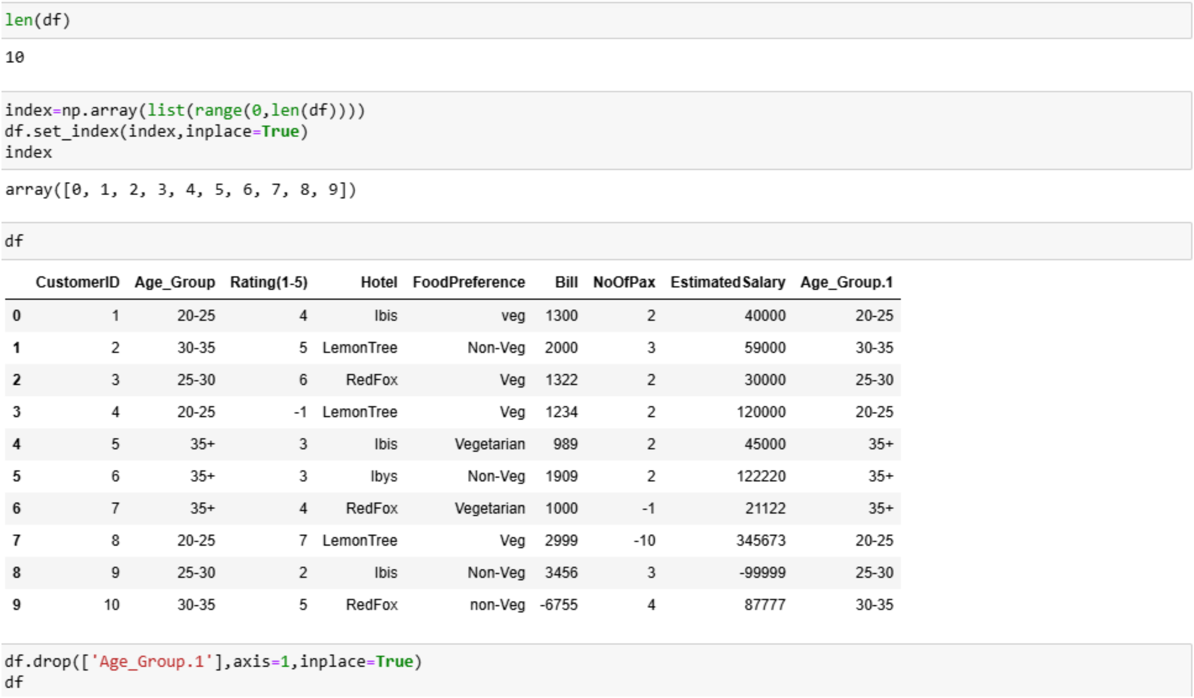
```
0     False
1     False
2     False
3     False
4     False
5     False
6     False
7     False
8     False
9      True
10    False
dtype: bool
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   CustomerID       11 non-null     int64 
 1   Age_Group        11 non-null     object
 2   Rating(1-5)      11 non-null     int64 
 3   Hotel            11 non-null     object
 4   FoodPreference   11 non-null     object
 5   Bill             11 non-null     int64 
 6   NoOfPax          11 non-null     int64 
 7   EstimatedSalary  11 non-null     int64 
 8   Age_Group.1      11 non-null     object
dtypes: int64(5), object(4)
memory usage: 920.0+ bytes
```

```
df.drop_duplicates(inplace=True)
df
```

	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary	Age_Group.1
0	1	20-25	4	Ibis	veg	1300	2	40000	20-25
1	2	30-35	5	LemonTree	Non-Veg	2000	3	59000	30-35
2	3	25-30	6	RedFox	Veg	1322	2	30000	25-30
3	4	20-25	-1	LemonTree	Veg	1234	2	120000	20-25
4	5	35+	3	Ibis	Vegetarian	989	2	45000	35+
5	6	35+	3	Ibys	Non-Veg	1909	2	122220	35+
6	7	35+	4	RedFox	Vegetarian	1000	-1	21122	35+
7	8	20-25	7	LemonTree	Veg	2999	-10	345673	20-25
8	9	25-30	2	Ibis	Non-Veg	3456	3	-99999	25-30
10	10	30-35	5	RedFox	non-Veg	-6755	4	87777	30-35



```
len(df)
```

```
10
```

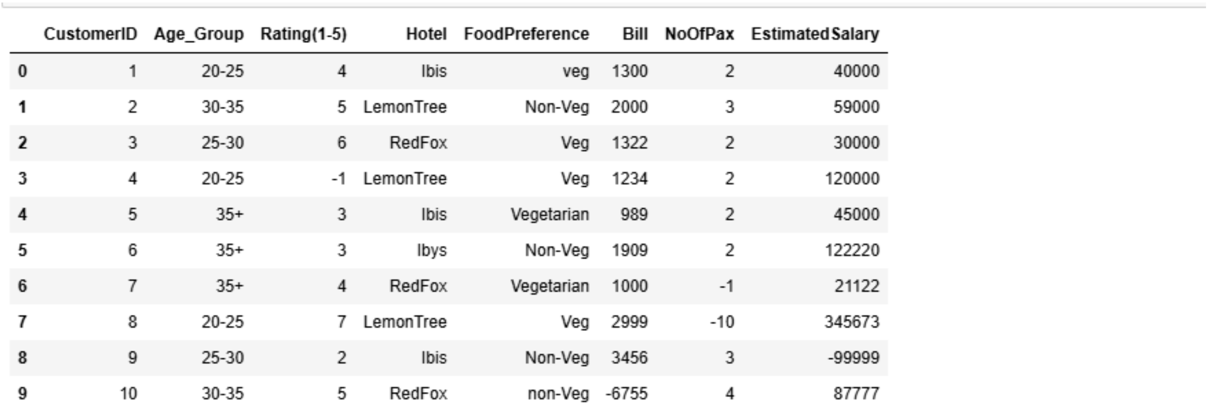
```
index=np.array(list(range(0,len(df))))
df.set_index(index,inplace=True)
index
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

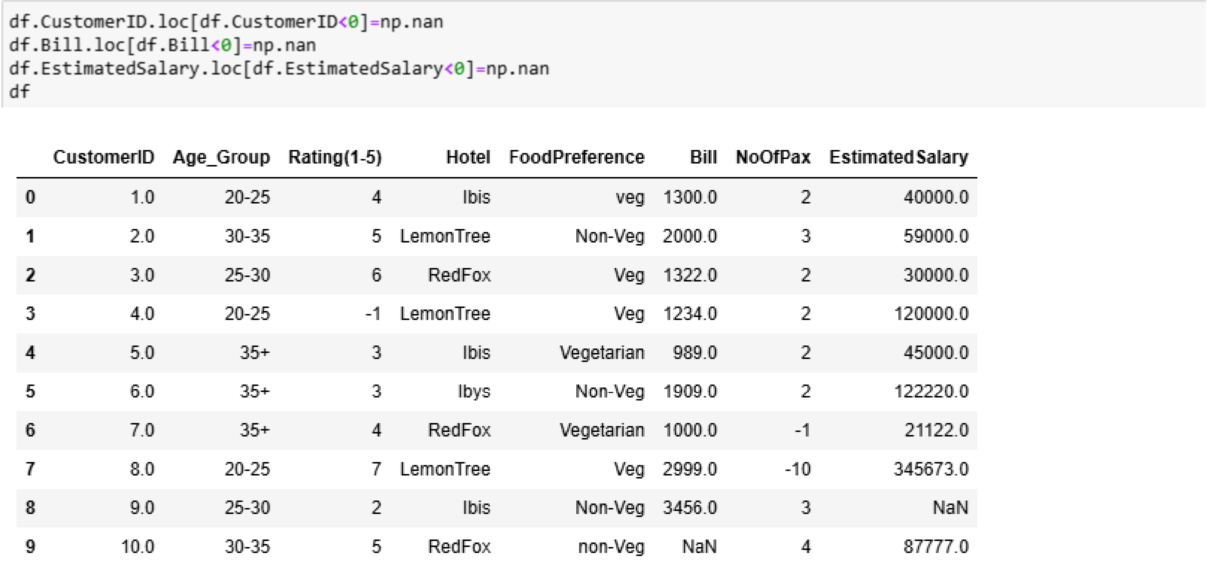
```
df
```

	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary	Age_Group.1
0	1	20-25	4	Ibis	veg	1300	2	40000	20-25
1	2	30-35	5	LemonTree	Non-Veg	2000	3	59000	30-35
2	3	25-30	6	RedFox	Veg	1322	2	30000	25-30
3	4	20-25	-1	LemonTree	Veg	1234	2	120000	20-25
4	5	35+	3	Ibis	Vegetarian	989	2	45000	35+
5	6	35+	3	Ibys	Non-Veg	1909	2	122220	35+
6	7	35+	4	RedFox	Vegetarian	1000	-1	21122	35+
7	8	20-25	7	LemonTree	Veg	2999	-10	345673	20-25
8	9	25-30	2	Ibis	Non-Veg	3456	3	-99999	25-30
9	10	30-35	5	RedFox	non-Veg	-6755	4	87777	30-35

```
df.drop(['Age_Group.1'],axis=1,inplace=True)
df
```

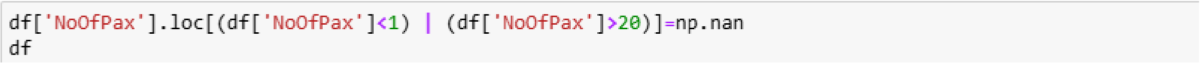


	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary
0	1	20-25	4	Ibis	veg	1300	2	40000
1	2	30-35	5	LemonTree	Non-Veg	2000	3	59000
2	3	25-30	6	RedFox	Veg	1322	2	30000
3	4	20-25	-1	LemonTree	Veg	1234	2	120000
4	5	35+	3	Ibis	Vegetarian	989	2	45000
5	6	35+	3	Ibys	Non-Veg	1909	2	122220
6	7	35+	4	RedFox	Vegetarian	1000	-1	21122
7	8	20-25	7	LemonTree	Veg	2999	-10	345673
8	9	25-30	2	Ibis	Non-Veg	3456	3	-99999
9	10	30-35	5	RedFox	non-Veg	-6755	4	87777

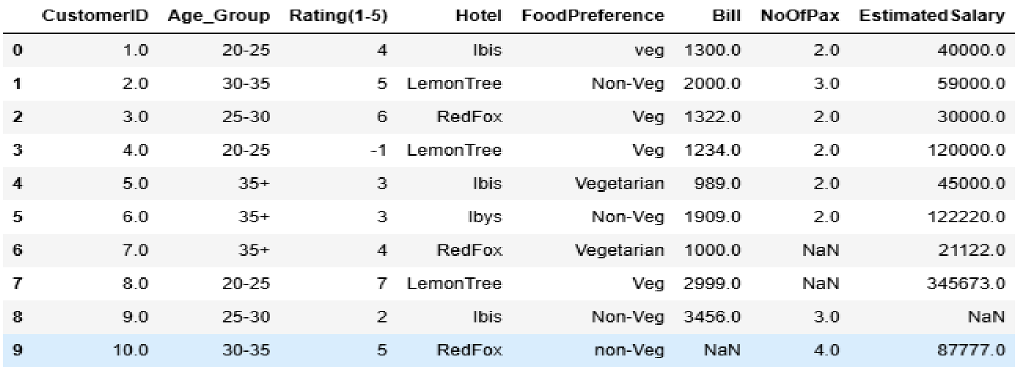


```
df.CustomerID.loc[df.CustomerID<0]=np.nan
df.Bill.loc[df.Bill<0]=np.nan
df.EstimatedSalary.loc[df.EstimatedSalary<0]=np.nan
df
```

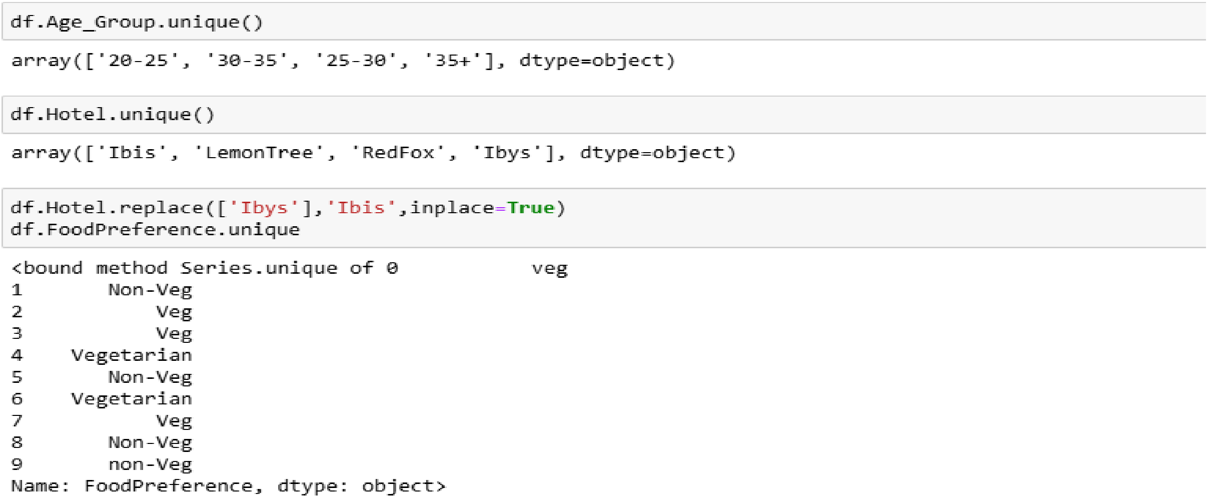
	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	EstimatedSalary
0	1.0	20-25	4	Ibis	veg	1300.0	2	40000.0
1	2.0	30-35	5	LemonTree	Non-Veg	2000.0	3	59000.0
2	3.0	25-30	6	RedFox	Veg	1322.0	2	30000.0
3	4.0	20-25	-1	LemonTree	Veg	1234.0	2	120000.0
4	5.0	35+	3	Ibis	Vegetarian	989.0	2	45000.0
5	6.0	35+	3	Ibys	Non-Veg	1909.0	2	122220.0
6	7.0	35+	4	RedFox	Vegetarian	1000.0	-1	21122.0
7	8.0	20-25	7	LemonTree	Veg	2999.0	-10	345673.0
8	9.0	25-30	2	Ibis	Non-Veg	3456.0	3	NaN
9	10.0	30-35	5	RedFox	non-Veg	NaN	4	87777.0



```
df['NoOfPax'].loc[(df['NoOfPax']<1) | (df['NoOfPax']>20)]=np.nan
df
```



	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	Estimated Salary
0	1.0	20-25	4	Ibis	veg	1300.0	2.0	40000.0
1	2.0	30-35	5	LemonTree	Non-Veg	2000.0	3.0	59000.0
2	3.0	25-30	6	RedFox	Veg	1322.0	2.0	30000.0
3	4.0	20-25	-1	LemonTree	Veg	1234.0	2.0	120000.0
4	5.0	35+	3	Ibis	Vegetarian	989.0	2.0	45000.0
5	6.0	35+	3	Ibys	Non-Veg	1909.0	2.0	122220.0
6	7.0	35+	4	RedFox	Vegetarian	1000.0	NaN	21122.0
7	8.0	20-25	7	LemonTree	Veg	2999.0	NaN	345673.0
8	9.0	25-30	2	Ibis	Non-Veg	3456.0	3.0	NaN
9	10.0	30-35	5	RedFox	non-Veg	NaN	4.0	87777.0



```
df.Age_Group.unique()
```

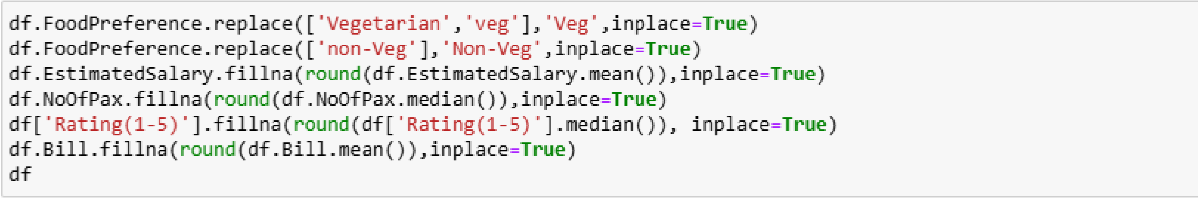
```
array(['20-25', '30-35', '25-30', '35+'], dtype=object)
```

```
df.Hotel.unique()
```

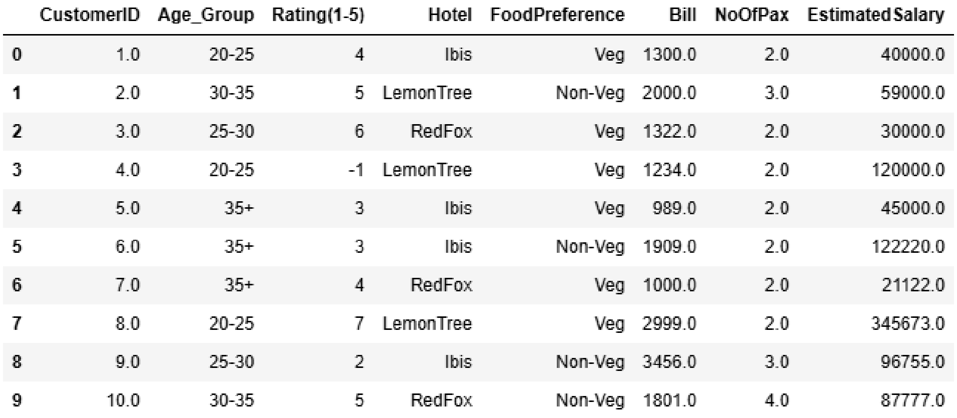
```
array(['Ibis', 'LemonTree', 'RedFox', 'Ibys'], dtype=object)
```

```
df.Hotel.replace(['Ibys'],'Ibis',inplace=True)
df.FoodPreference.unique
```

```
<bound method Series.unique of 0           veg
1        Non-Veg
2            Veg
3            Veg
4     Vegetarian
5        Non-Veg
6     Vegetarian
7            Veg
8        Non-Veg
9        non-Veg
Name: FoodPreference, dtype: object>
```



```
df.FoodPreference.replace(['Vegetarian','veg'],'Veg',inplace=True)
df.FoodPreference.replace(['non-Veg'],'Non-Veg',inplace=True)
df.EstimatedSalary.fillna(round(df.EstimatedSalary.mean()),inplace=True)
df.NoOfPax.fillna(round(df.NoOfPax.median()),inplace=True)
df['Rating(1-5)'].fillna(round(df['Rating(1-5)'].median()), inplace=True)
df.Bill.fillna(round(df.Bill.mean()),inplace=True)
df
```



	CustomerID	Age_Group	Rating(1-5)	Hotel	FoodPreference	Bill	NoOfPax	Estimated Salary
0	1.0	20-25	4	Ibis	Veg	1300.0	2.0	40000.0
1	2.0	30-35	5	LemonTree	Non-Veg	2000.0	3.0	59000.0
2	3.0	25-30	6	RedFox	Veg	1322.0	2.0	30000.0
3	4.0	20-25	-1	LemonTree	Veg	1234.0	2.0	120000.0
4	5.0	35+	3	Ibis	Veg	989.0	2.0	45000.0
5	6.0	35+	3	Ibis	Non-Veg	1909.0	2.0	122220.0
6	7.0	35+	4	RedFox	Veg	1000.0	2.0	21122.0
7	8.0	20-25	7	LemonTree	Veg	2999.0	2.0	345673.0
8	9.0	25-30	2	Ibis	Non-Veg	3456.0	3.0	96755.0
9	10.0	30-35	5	RedFox	Non-Veg	1801.0	4.0	87777.0

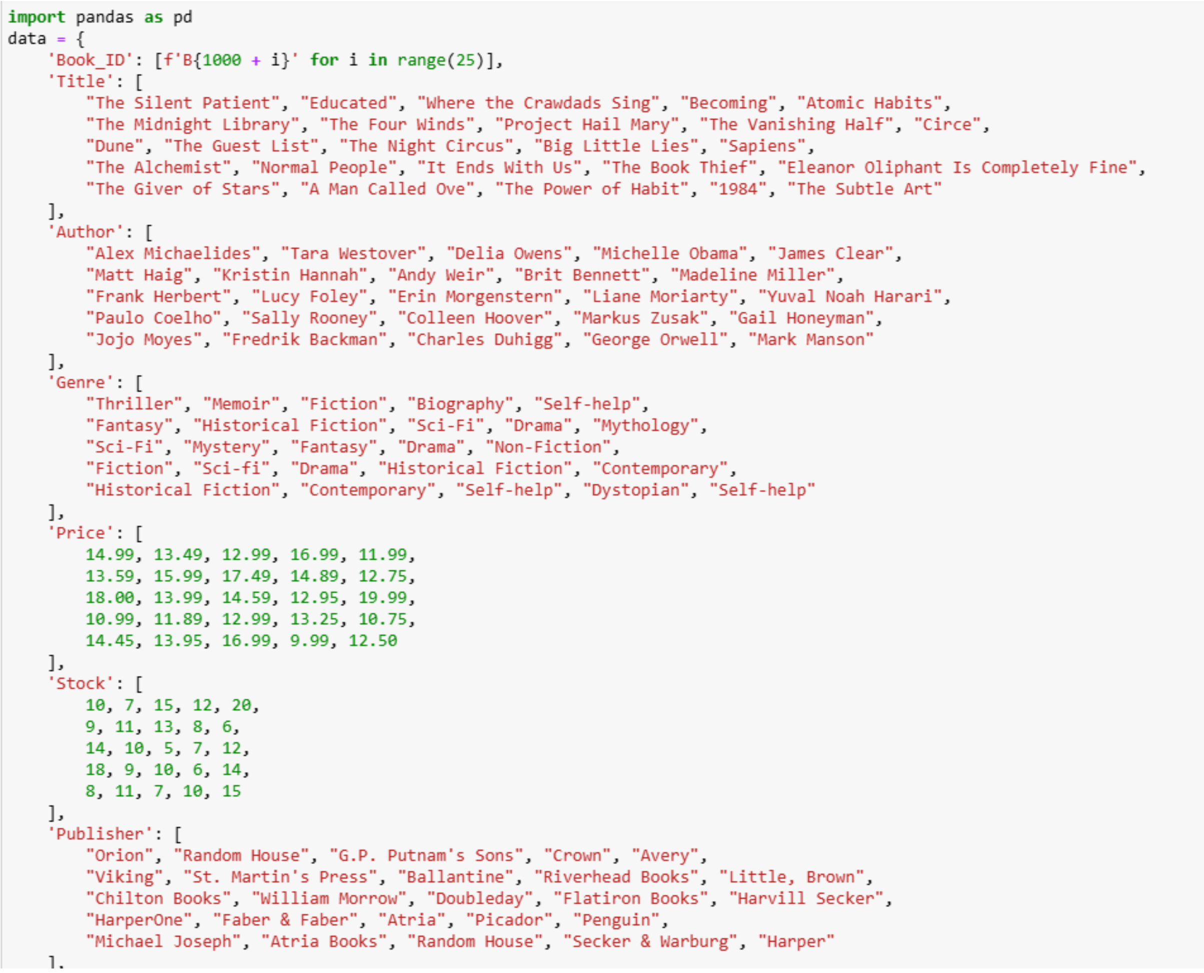
Result:
The dataset was successfully preprocessed using Pandas — missing values were handled, data was cleaned, and made ready for further analysis.

3.C) PANDAS LIBRARY – CREATE OWN DATASET IN CSV FORMAT

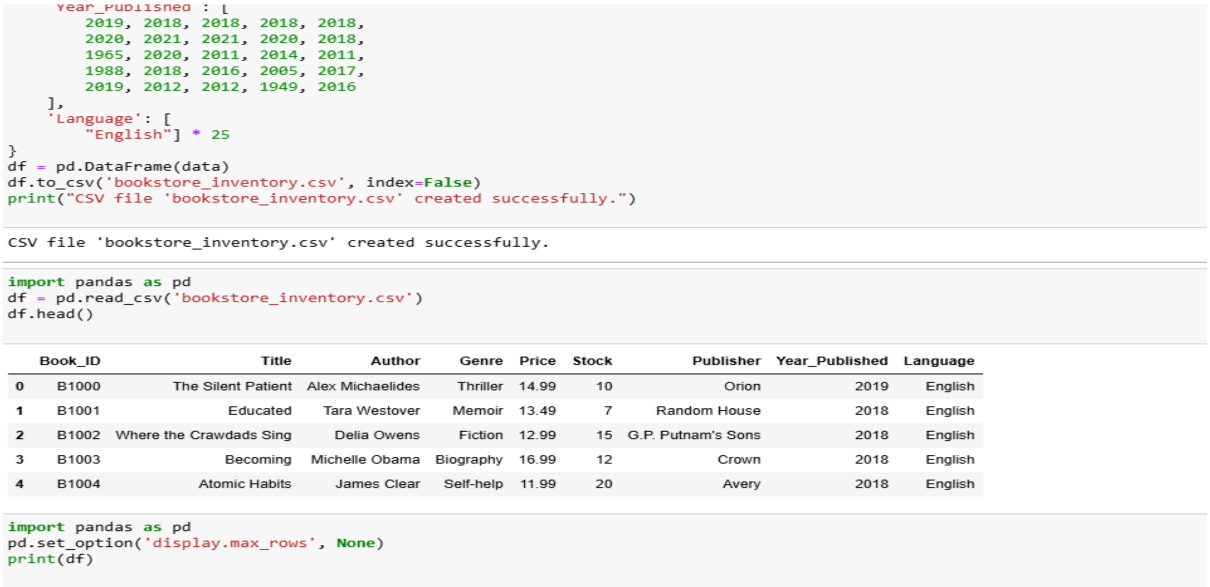
Aim:

To create our own dataset and store it in a CSV (Comma Separated Values) file using the Pandas library in Python.

Code:



```
import pandas as pd
data = {
    'Book_ID': [f'B{1000 + i}' for i in range(25)],
    'Title': [
        "The Silent Patient", "Educated", "Where the Crawdads Sing", "Becoming", "Atomic Habits",
        "The Midnight Library", "The Four Winds", "Project Hail Mary", "The Vanishing Half", "Circe",
        "Dune", "The Guest List", "The Night Circus", "Big Little Lies", "Sapiens",
        "The Alchemist", "Normal People", "It Ends With Us", "The Book Thief", "Eleanor Oliphant Is Completely Fine",
        "The Giver of Stars", "A Man Called Ove", "The Power of Habit", "1984", "The Subtle Art"
    ],
    'Author': [
        "Alex Michaelides", "Tara Westover", "Delia Owens", "Michelle Obama", "James Clear",
        "Matt Haig", "Kristin Hannah", "Andy Weir", "Brit Bennett", "Madeline Miller",
        "Frank Herbert", "Lucy Foley", "Erin Morgenstern", "Liane Moriarty", "Yuval Noah Harari",
        "Paulo Coelho", "Sally Rooney", "Colleen Hoover", "Markus Zusak", "Gail Honeyman",
        "Jojo Moyes", "Fredrik Backman", "Charles Duhigg", "George Orwell", "Mark Manson"
    ],
    'Genre': [
        "Thriller", "Memoir", "Fiction", "Biography", "Self-help",
        "Fantasy", "Historical Fiction", "Sci-Fi", "Drama", "Mythology",
        "Sci-Fi", "Mystery", "Fantasy", "Drama", "Non-Fiction",
        "Fiction", "Sci-fi", "Drama", "Historical Fiction", "Contemporary",
        "Historical Fiction", "Contemporary", "Self-help", "Dystopian", "Self-help"
    ],
    'Price': [
        14.99, 13.49, 12.99, 16.99, 11.99,
        13.59, 15.99, 17.49, 14.89, 12.75,
        18.00, 13.99, 14.59, 12.95, 19.99,
        10.99, 11.89, 12.99, 13.25, 10.75,
        14.45, 13.95, 16.99, 9.99, 12.50
    ],
    'Stock': [
        10, 7, 15, 12, 20,
        9, 11, 13, 8, 6,
        14, 10, 5, 7, 12,
        18, 9, 10, 6, 14,
        8, 11, 7, 10, 15
    ],
    'Publisher': [
        "Orion", "Random House", "G.P. Putnam's Sons", "Crown", "Avery",
        "Viking", "St. Martin's Press", "Ballantine", "Riverhead Books", "Little, Brown",
        "Chilton Books", "William Morrow", "Doubleday", "Flatiron Books", "Harvill Secker",
        "HarperOne", "Faber & Faber", "Atria", "Picador", "Penguin",
        "Michael Joseph", "Atria Books", "Random House", "Secker & Warburg", "Harper"
    ],
```



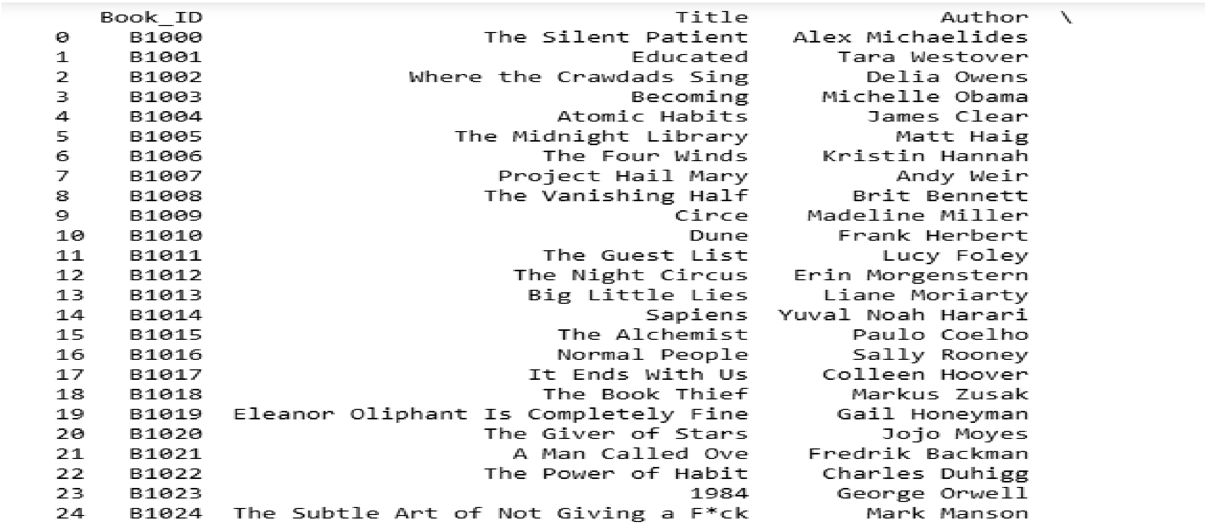
```
    'Year_Published': [
        2019, 2018, 2018, 2018, 2018,
        2020, 2021, 2021, 2020, 2018,
        1965, 2020, 2011, 2014, 2011,
        1988, 2018, 2016, 2005, 2017,
        2019, 2012, 2012, 1949, 2016
    ],
    'Language': [
        "English"] * 25
}
df = pd.DataFrame(data)
df.to_csv('bookstore_inventory.csv', index=False)
print("CSV file 'bookstore_inventory.csv' created successfully.")
```

```
CSV file 'bookstore_inventory.csv' created successfully.
```

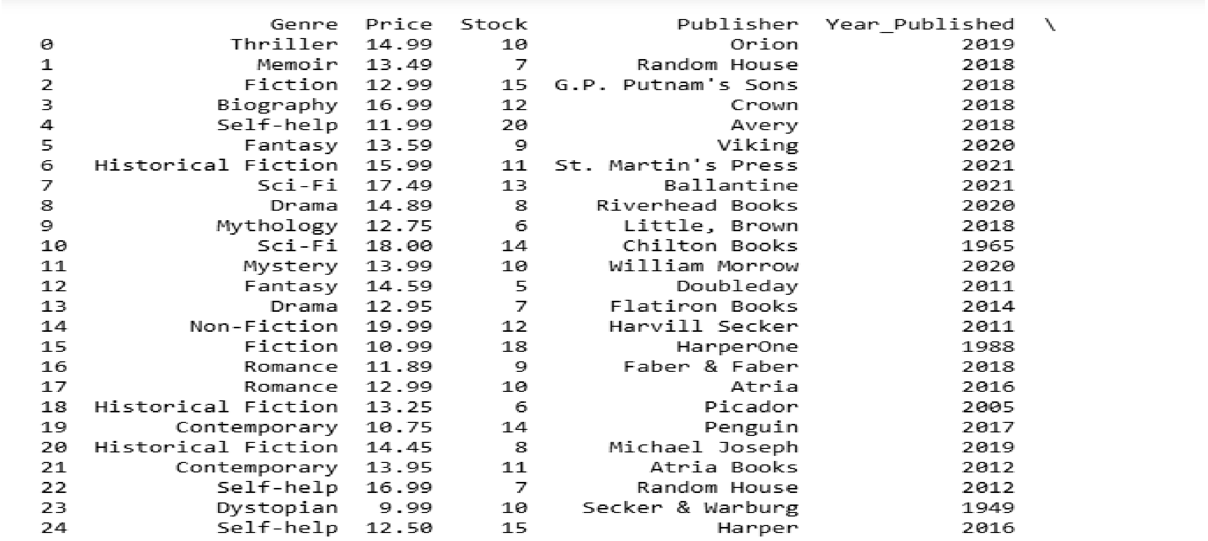
```
import pandas as pd
df = pd.read_csv('bookstore_inventory.csv')
df.head()
```

	Book_ID	Title	Author	Genre	Price	Stock	Publisher	Year_Published	Language
0	B1000	The Silent Patient	Alex Michaelides	Thriller	14.99	10	Orion	2019	English
1	B1001	Educated	Tara Westover	Memoir	13.49	7	Random House	2018	English
2	B1002	Where the Crawdads Sing	Delia Owens	Fiction	12.99	15	G.P. Putnam's Sons	2018	English
3	B1003	Becoming	Michelle Obama	Biography	16.99	12	Crown	2018	English
4	B1004	Atomic Habits	James Clear	Self-help	11.99	20	Avery	2018	English

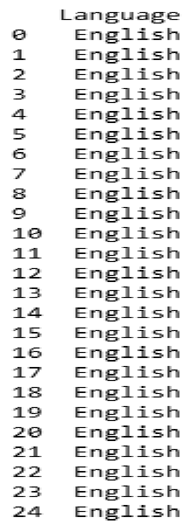
```
import pandas as pd
pd.set_option('display.max_rows', None)
print(df)
```



```
   Book_ID                                   Title             Author  \
0    B1000                      The Silent Patient   Alex Michaelides
1    B1001                                Educated     Tara Westover
2    B1002                  Where the Crawdads Sing       Delia Owens
3    B1003                                Becoming     Michelle Obama
4    B1004                           Atomic Habits       James Clear
5    B1005                     The Midnight Library         Matt Haig
6    B1006                          The Four Winds    Kristin Hannah
7    B1007                        Project Hail Mary         Andy Weir
8    B1008                      The Vanishing Half      Brit Bennett
9    B1009                                   Circe   Madeline Miller
10   B1010                                    Dune     Frank Herbert
11   B1011                          The Guest List        Lucy Foley
12   B1012                        The Night Circus  Erin Morgenstern
13   B1013                         Big Little Lies    Liane Moriarty
14   B1014                                 Sapiens Yuval Noah Harari
15   B1015                           The Alchemist      Paulo Coelho
16   B1016                           Normal People      Sally Rooney
17   B1017                          It Ends With Us     Colleen Hoover
18   B1018                           The Book Thief      Markus Zusak
19   B1019  Eleanor Oliphant Is Completely Fine     Gail Honeyman
20   B1020                       The Giver of Stars        Jojo Moyes
21   B1021                        A Man Called Ove   Fredrik Backman
22   B1022                      The Power of Habit    Charles Duhigg
23   B1023                                    1984     George Orwell
24   B1024  The Subtle Art of Not Giving a F*ck       Mark Manson
```



```
                Genre  Price  Stock           Publisher  Year_Published  \
0            Thriller  14.99     10               Orion            2019
1              Memoir  13.49      7        Random House            2018
2             Fiction  12.99     15  G.P. Putnam's Sons            2018
3           Biography  16.99     12               Crown            2018
4           Self-help  11.99     20               Avery            2018
5             Fantasy  13.59      9              Viking            2020
6  Historical Fiction  15.99     11   St. Martin's Press            2021
7              Sci-Fi  17.49     13          Ballantine            2021
8               Drama  14.89      8      Riverhead Books            2020
9           Mythology  12.75      6       Little, Brown            2018
10             Sci-Fi  18.00     14       Chilton Books            1965
11            Mystery  13.99     10      William Morrow            2020
12            Fantasy  14.59      5           Doubleday            2011
13              Drama  12.95      7      Flatiron Books            2014
14        Non-Fiction  19.99     12       Harvill Secker            2011
15            Fiction  10.99     18           HarperOne            1988
16            Romance  11.89      9        Faber & Faber            2018
17            Romance  12.99     10               Atria            2016
18 Historical Fiction  13.25      6             Picador            2005
19       Contemporary  10.75     14             Penguin            2017
20 Historical Fiction  14.45      8       Michael Joseph            2019
21       Contemporary  13.95     11         Atria Books            2012
22          Self-help  16.99      7        Random House            2012
23          Dystopian   9.99     10      Secker & Warburg            1949
24          Self-help  12.50     15              Harper            2016
```



```
   Language
0   English
1   English
2   English
3   English
4   English
5   English
6   English
7   English
8   English
9   English
10  English
11  English
12  English
13  English
14  English
15  English
16  English
17  English
18  English
19  English
20  English
21  English
22  English
23  English
24  English
```

Result:
A new dataset was successfully created and saved as a CSV file using Pandas.