Python Coding Challenge

Execute Data cleaning programs & Panda joins in Python

Execute Data cleaning 3 programs:

Import Data:

import pandas as pd

df = pd.read_csv("products.csv.txt")

Program 1: Handling Missing Values

print("Missing values before cleaning:\n", df.isnull().sum())

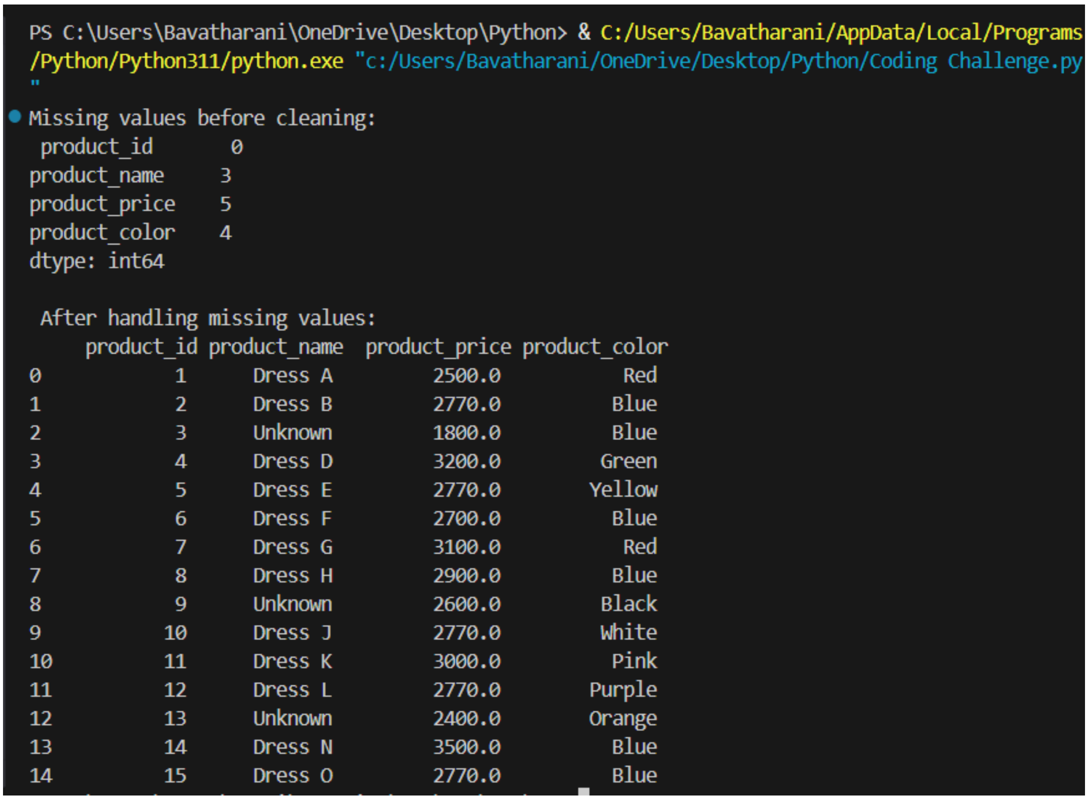
df['product_name'] = df['product_name'].fillna("Unknown")

df['product_price'] = df['product_price'].fillna(df['product_price'].mean())

df['product_color'] = df['product_color'].fillna(df['product_color'].mode()[0])

print("\n After handling missing values:\n", df)

Output:



```
PS C:\Users\Bavatharani\OneDrive\Desktop\Python> & C:/Users/Bavatharani/AppData/Local/Programs
/Python/Python311/python.exe "c:/Users/Bavatharani/OneDrive/Desktop/Python/Coding Challenge.py
"
Missing values before cleaning:
 product_id       0
product_name     3
product_price    5
product_color    4
dtype: int64

 After handling missing values:
     product_id product_name  product_price product_color
0            1      Dress A         2500.0           Red
1            2      Dress B         2770.0          Blue
2            3      Unknown         1800.0          Blue
3            4      Dress D         3200.0         Green
4            5      Dress E         2770.0        Yellow
5            6      Dress F         2700.0          Blue
6            7      Dress G         3100.0           Red
7            8      Dress H         2900.0          Blue
8            9      Unknown         2600.0         Black
9           10      Dress J         2770.0         White
10          11      Dress K         3000.0          Pink
11          12      Dress L         2770.0        Purple
12          13      Unknown         2400.0        Orange
13          14      Dress N         3500.0          Blue
14          15      Dress O         2770.0          Blue
```

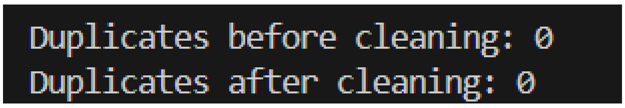
Program 2: Removing Duplicate Records

```
print("Duplicates before cleaning:", df.duplicated().sum())

df = df.drop_duplicates()

print("Duplicates after cleaning:", df.duplicated().sum())
```

Output:



```
Duplicates before cleaning: 0
Duplicates after cleaning: 0
```

Program 3: Changing Data Types

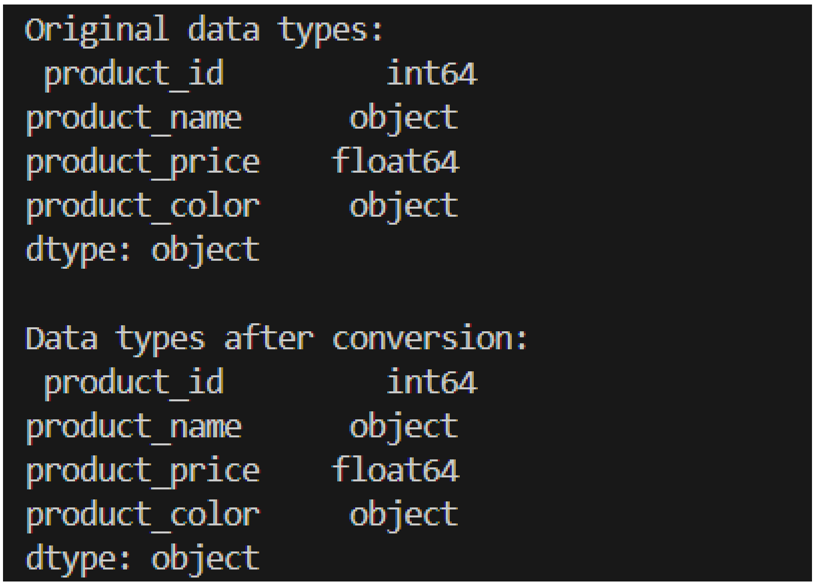
```
print("Original data types:\n", df.dtypes)

df['product_id'] = df['product_id'].astype(int)

df['product_price'] = pd.to_numeric(df['product_price'])

print("\nData types after conversion:\n", df.dtypes)
```

Output:



```
Original data types:
 product_id         int64
product_name      object
product_price    float64
product_color     object
dtype: object

Data types after conversion:
 product_id         int64
product_name      object
product_price    float64
product_color     object
dtype: object
```

Joins

import pandas as pd

# Step 1: Load both CSV files

products_df = pd.read_csv("products.csv.txt")   # Make sure file name is correct

stock_df = pd.read_csv("products_stock.csv.txt")   # Related stock file

products_df.rename(columns={'product_id': 'id'}, inplace=True)

stock_df.rename(columns={'product_id': 'id'}, inplace=True)

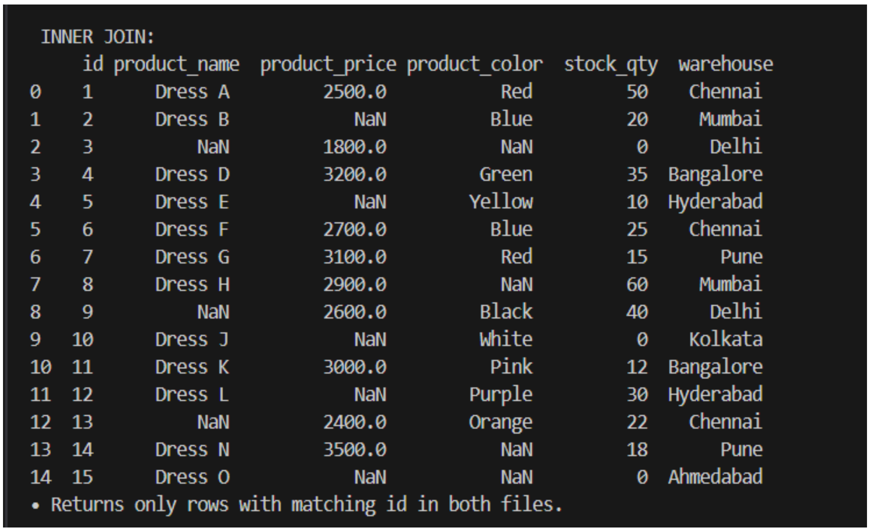
# Step 2: INNER JOIN

inner_join = pd.merge(products_df, stock_df, on='id', how='inner')

print("\n INNER JOIN:\n", inner_join)

print("• Returns only rows with matching id in both files.\n")

Output:



```
 INNER JOIN:
     id product_name  product_price product_color  stock_qty  warehouse
0    1      Dress A         2500.0           Red         50    Chennai
1    2      Dress B            NaN          Blue         20     Mumbai
2    3          NaN         1800.0           NaN          0      Delhi
3    4      Dress D         3200.0         Green         35  Bangalore
4    5      Dress E            NaN        Yellow         10  Hyderabad
5    6      Dress F         2700.0          Blue         25    Chennai
6    7      Dress G         3100.0           Red         15       Pune
7    8      Dress H         2900.0           NaN         60     Mumbai
8    9          NaN         2600.0         Black         40      Delhi
9   10      Dress J            NaN         White          0    Kolkata
10  11      Dress K         3000.0          Pink         12  Bangalore
11  12      Dress L            NaN        Purple         30  Hyderabad
12  13          NaN         2400.0        Orange         22    Chennai
13  14      Dress N         3500.0           NaN         18       Pune
14  15      Dress O            NaN           NaN          0  Ahmedabad
• Returns only rows with matching id in both files.
```

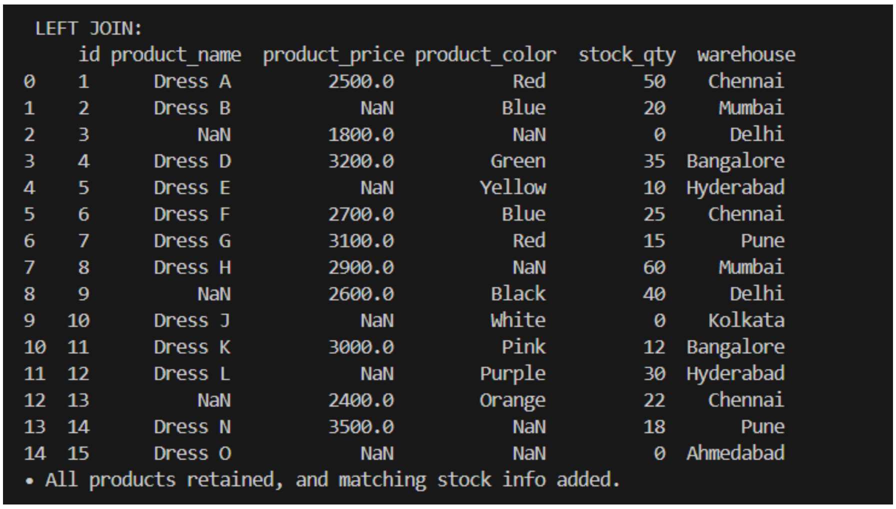
# Step 3: LEFT JOIN

left_join = pd.merge(products_df, stock_df, on='id', how='left')

print(" LEFT JOIN:\n", left_join)

print("• All products retained, and matching stock info added.\n")

Output:



```
 LEFT JOIN:
     id product_name  product_price product_color  stock_qty  warehouse
0    1      Dress A         2500.0           Red         50    Chennai
1    2      Dress B            NaN          Blue         20     Mumbai
2    3          NaN         1800.0           NaN          0      Delhi
3    4      Dress D         3200.0         Green         35  Bangalore
4    5      Dress E            NaN        Yellow         10  Hyderabad
5    6      Dress F         2700.0          Blue         25    Chennai
6    7      Dress G         3100.0           Red         15       Pune
7    8      Dress H         2900.0           NaN         60     Mumbai
8    9          NaN         2600.0         Black         40      Delhi
9   10      Dress J            NaN         White          0    Kolkata
10  11      Dress K         3000.0          Pink         12  Bangalore
11  12      Dress L            NaN        Purple         30  Hyderabad
12  13          NaN         2400.0        Orange         22    Chennai
13  14      Dress N         3500.0           NaN         18       Pune
14  15      Dress O            NaN           NaN          0  Ahmedabad
• All products retained, and matching stock info added.
```

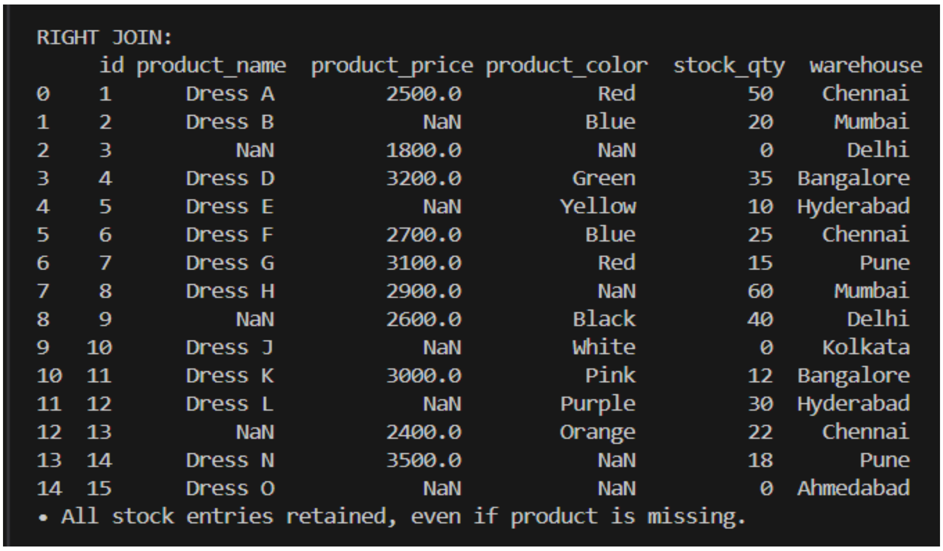
# Step 4: RIGHT JOIN

right_join = pd.merge(products_df, stock_df, on='id', how='right')

print("RIGHT JOIN:\n", right_join)

print("• All stock entries retained, even if product is missing.\n")

Output:



```
RIGHT JOIN:
     id product_name  product_price product_color  stock_qty  warehouse
0    1      Dress A         2500.0           Red         50    Chennai
1    2      Dress B            NaN          Blue         20     Mumbai
2    3          NaN         1800.0           NaN          0      Delhi
3    4      Dress D         3200.0         Green         35  Bangalore
4    5      Dress E            NaN        Yellow         10  Hyderabad
5    6      Dress F         2700.0          Blue         25    Chennai
6    7      Dress G         3100.0           Red         15       Pune
7    8      Dress H         2900.0           NaN         60     Mumbai
8    9          NaN         2600.0         Black         40      Delhi
9   10      Dress J            NaN         White          0    Kolkata
10  11      Dress K         3000.0          Pink         12  Bangalore
11  12      Dress L            NaN        Purple         30  Hyderabad
12  13          NaN         2400.0        Orange         22    Chennai
13  14      Dress N         3500.0           NaN         18       Pune
14  15      Dress O            NaN           NaN          0  Ahmedabad
• All stock entries retained, even if product is missing.
```

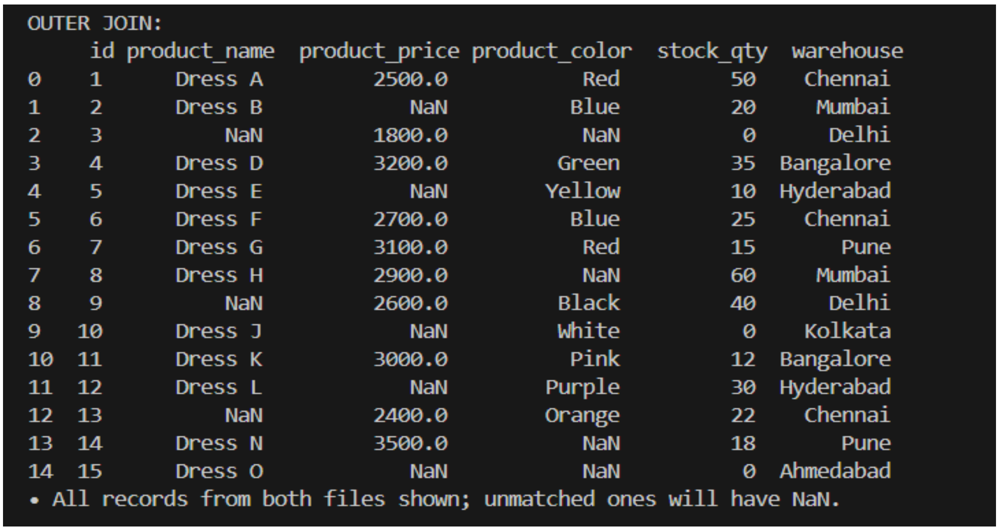
# Step 5: OUTER JOIN

outer_join = pd.merge(products_df, stock_df, on='id', how='outer')

print("OUTER JOIN:\n", outer_join)

print("• All records from both files shown; unmatched ones will have NaN.\n")

Output:



```
OUTER JOIN:
    id product_name  product_price product_color  stock_qty  warehouse
0    1      Dress A         2500.0           Red         50    Chennai
1    2      Dress B            NaN          Blue         20     Mumbai
2    3          NaN         1800.0           NaN          0      Delhi
3    4      Dress D         3200.0         Green         35  Bangalore
4    5      Dress E            NaN        Yellow         10  Hyderabad
5    6      Dress F         2700.0          Blue         25    Chennai
6    7      Dress G         3100.0           Red         15       Pune
7    8      Dress H         2900.0           NaN         60     Mumbai
8    9          NaN         2600.0         Black         40      Delhi
9   10      Dress J            NaN         White          0    Kolkata
10  11      Dress K         3000.0          Pink         12  Bangalore
11  12      Dress L            NaN        Purple         30  Hyderabad
12  13          NaN         2400.0        Orange         22    Chennai
13  14      Dress N         3500.0           NaN         18       Pune
14  15      Dress O            NaN           NaN          0  Ahmedabad
• All records from both files shown; unmatched ones will have NaN.
```