

# DATA CLEANING EXERCISE

Use Python to download a raw data file and clean it.

USJ  
INGENIERIA DE DATOS CURSO 2025-2026

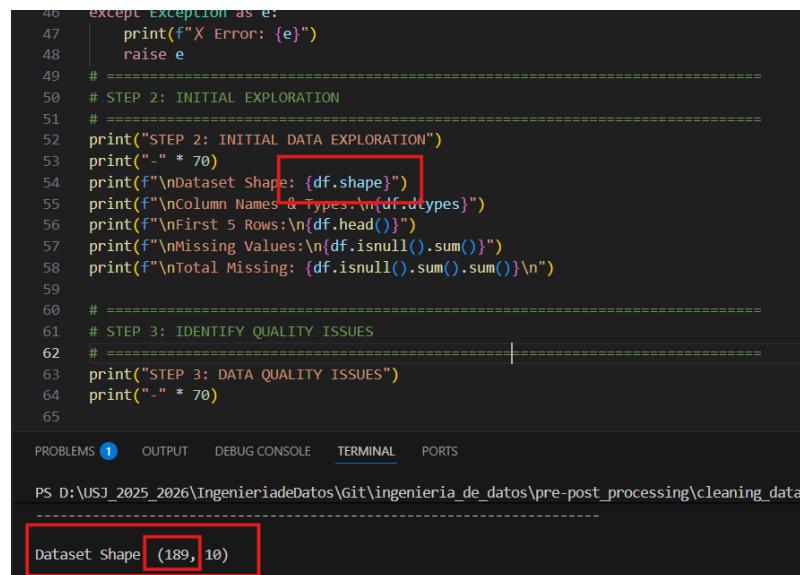
FRANCISCO JAVIER LIARTE NASARRE

## Statement of the problem

- Use python ‘requests’ package to get the file from here

[https://raw.githubusercontent.com/victorbrub/data-engineeringclass/refs/heads/main/pre-post\\_processing/exercise.csv](https://raw.githubusercontent.com/victorbrub/data-engineeringclass/refs/heads/main/pre-post_processing/exercise.csv)

- Check the file data to fast check. Read it with Pandas.
- How many rows do we have?



```
46 except Exception as e:
47     print("X Error: {e}")
48     raise e
49 # =====
50 # STEP 2: INITIAL EXPLORATION
51 # =====
52 print("STEP 2: INITIAL DATA EXPLORATION")
53 print("." * 70)
54 print(f"\nDataset Shape: {df.shape}")
55 print(f"\nColumn Names & Types: {df.dtypes}")
56 print(f"\nFirst 5 Rows: {df.head()}")
57 print(f"\nMissing Values: {df.isnull().sum()}")
58 print(f"\nTotal Missing: {df.isnull().sum().sum()}\n")
59
60 # =====
61 # STEP 3: IDENTIFY QUALITY ISSUES
62 # =====
63 print("STEP 3: DATA QUALITY ISSUES")
64 print("." * 70)
65
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\USJ\_2025\_2026\Ingenieria de Datos\git\ingenieria\_de\_datos\pre-post\_processing\cleaning\_data

Dataset Shape (189, 10)

We have 189 rows

- Is there any sensible information?

Yes, I mean, with the CustomerName, Phone and Age, you can easily find that person nowadays (if he does not use some type of alias or has other security options)

- What kind of problems can we have regarding the nature of this data?

Looking at the data we can see is from a shop or some warehouse. This data is very important, because is crucial to keep the orders of the clients. This data can be used to solve issues and to keep records of the orders of the clients and create analyses to improve the shop’s performance.

- Clean it.

In the step of Python Script.

- Define the rules we need to clean the data.

Dimension	Definition	Example	Impact
<b>Accuracy</b>	Data represents reality correctly	Age should be realistic (not 999 years)	Wrong decisions
<b>Completeness</b>	All required data is present	Every customer must have an email	Missing insights
<b>Consistency</b>	Data is uniform across systems	"USA" vs "United States"	Failed joins
<b>Validity</b>	Data conforms to required format	Email must have @ symbol	System errors
<b>Uniqueness</b>	No duplicate records exist	One OrderID per order	Inflated metrics
<b>Timeliness</b>	Data is current and available	Latest customer address	Outdated insights

Following this table, we will do the following for the columns:

OrderID:

Missing -> Drop It. Is the primary key so if there is an error is the easiest way to solve it  
 Duplicate -> Drop em.

CustomerName:

Missing -> Impute unknown

Validate the format -> Name Surname, if is only changing the letters, reformat it.

Email:

Missing -> Drop It

Validate the format -> [juan@email.com](mailto:juan@email.com) if is only changing the letters, reformat it.

Phone:

Missing -> Drop It

Validate the format -> 555-XXXX if incorrect drop it

Country:

Missing -> Impute unknown

Validate the format -> CAPITAL if is only changing the letters, reformat it.

OrderDate:

Missing -> Drop It

Validate the format -> min-> 1/1/2023 max -> 31/12/2023 if incorrect, drop it

Quantity:

Missing -> Drop It

Validate the format -> INT, if no drop it

Price:

Missing -> Drop It

Validate the format -> INT, if no drop it

CustomerAge:

Missing -> Impute media()

Validate the format -> between 18-99 if incorrect drop it

OrderStatus:

Missing -> Drop It

- Implement a Python Script that cleans the data and analyzes the clean process.

Check the python files in my github repository: [https://github.com/liliarte-1/ingenieria\\_de\\_datos#](https://github.com/liliarte-1/ingenieria_de_datos#)

- Output should be a new file with cleaned data.

In step 6, in the cleaned\_exercise.csv.

- Create a test on the raw data and on the cleaned data, for each one of the dimensions. The output should be a percentage: rows that meet the test requirement / total rows. Compare the results of each file.

The screenshot shows a Jupyter Notebook cell with the following code:

```
136 # =====
137 # STEP 6: SAVE CLEANED DATA
138 # =====
139 # =====
140 df.to_csv('cleaned_exercise.csv', index=False)
141 #The cleaned data is saved to 'cleaned_exercise.csv'
142 print("\nCleaned data saved to 'cleaned_exercise.csv'")
143
144 df_cleaned_rows = len(df)
145 print(f"\nRows before cleaning: {df_raw_rows}, Rows after cleaning: {df_cleaned_rows}")
146 print(f"Rows percent meeting the quality criteria: {df_cleaned_rows / df_raw_rows * 100:.2f}%")
147 # =====
148 # SUMMARY
149 # =====
150
151
152 print(f"Completed at: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")
```

Below the code, the terminal output shows:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\USJ_2025_2026\Ingenieria de Datos\Git\ingenieria_de_datos\pre-post_processing\cleaning_data_lab> py main.py
dtype: int64

Cleaned data saved to 'cleaned_exercise.csv'

Rows before cleaning: 189, Rows after cleaning: 29
Rows percent meeting the quality criteria 15.34%
Completed at: 2025-11-27 16:29:53
PS D:\USJ_2025_2026\Ingenieria de Datos\Git\ingenieria_de_datos\pre-post_processing\cleaning_data_lab>
```

A red box highlights the output text "Rows before cleaning: 189, Rows after cleaning: 29" and "Rows percent meeting the quality criteria 15.34%".

As we can see, this data retrieval needs to be improved.