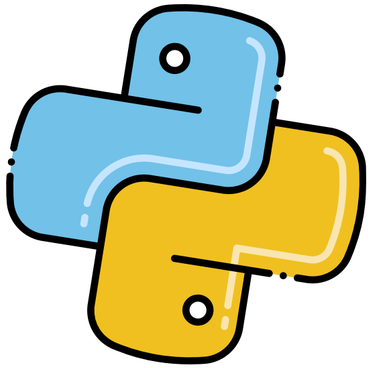
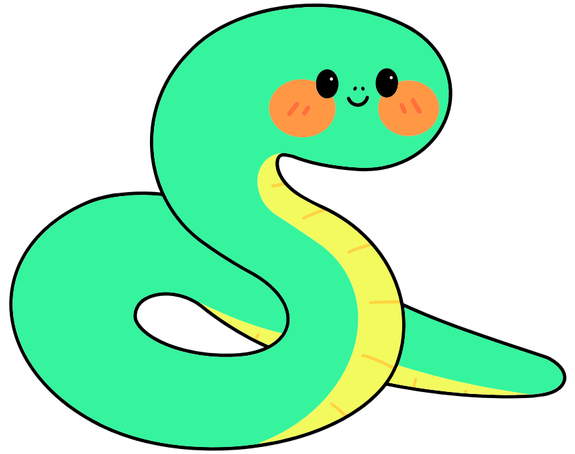
Data Cleaning using Python (Pandas)

This project focuses on cleaning and preparing a dataset using Python.



The goal is to

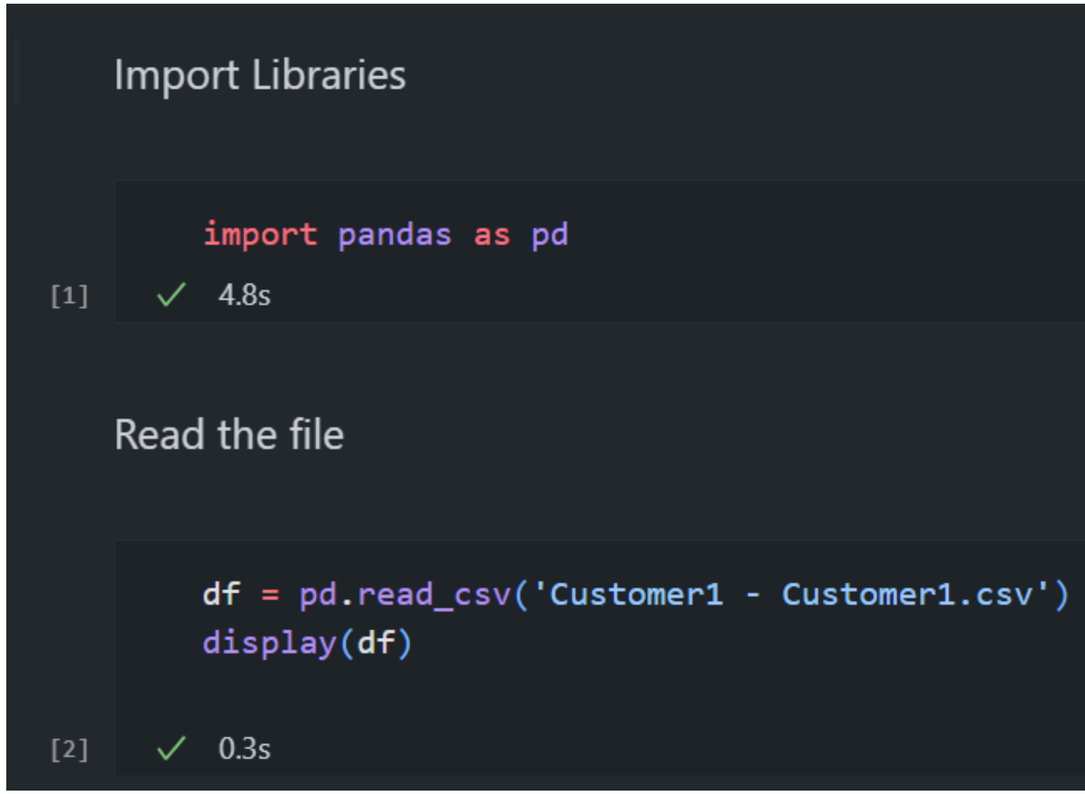
- handle missing values
- correct data types
- remove duplicates
- standardize categorical data



Importing Libraries & Reading the Data

- In this step, I started by importing the required library Pandas, which is essential for data manipulation and cleaning.

- Then, I loaded the dataset using the pd.read_csv() function to begin the analysis process.



Checking Missing Values

- Inspected all columns for missing values.
- Found null cells in Prefix and Gender columns.

Missing Values

```
# check how many missing values in the Columns
print(df.isnull().sum())
```

[3] ✓ 0.0s

```
CustomerKey        0
Prefix           130
FirstName          0
LastName           0
BirthDate          0
MaritalStatus      0
Gender           130
EmailAddress       0
AnnualIncome       0
TotalChildren      0
EducationLevel     0
Occupation         0
HomeOwner          0
dtype: int64
```

Handling Missing Values

Prefix Column:

- Instead of filling missing values, I merged the Prefix column with two other related columns to create a single combined field.
- This helped preserve useful information and reduce redundancy.

Gender Column

- Replaced all null cells with "Unknown" using fillna("Unknown").

```
# Merge Prefix, FirstName, and LastName
df['FullName'] = (
    df['Prefix'].fillna('') + ' ' +
    df['FirstName'].fillna('') + ' ' +
    df['LastName'].fillna('')
)
df['FullName'] = df['FullName'].str.strip()

# Handle missing values in Gender column
df['Gender'] = df['Gender'].fillna('Unknown')
print(df.isnull().sum())
```

✓ 0.2s

```
CustomerKey       0
Prefix            0
FirstName         0
LastName          0
BirthDate         0
MaritalStatus     0
Gender            0
```

Fixing Data Types

- *Reviewed data types using df.dtypes.*

```
[13]  print(df.dtypes)

...   CustomerKey          int64
      Prefix              object
      FirstName           object
      LastName            object
      BirthDate           object
      EmailAddress        object
      AnnualIncome        object
      TotalChildren        int64
      EducationLevel      object
      Occupation          object
      HomeOwner           object
      dtype: object
```

- Converted BirthDate from object to datetime.
- Converted AnnualIncome from object to float.

```
#AnnualIncom from object to float
df['AnnualIncome'] = df['AnnualIncome'].astype(str).str.replace('$','').str.replace(',','').astype(float)

#BirthDate from object to date
df['BirthDate'] = pd.to_datetime(df['BirthDate'])

print(df.dtypes)
display(df)
```

[30]

```
CustomerKey                int64
Prefix                    object
FirstName                 object
LastName                  object
BirthDate         datetime64[ns]
MaritalStatus             object
Gender                    object
EmailAddress              object
AnnualIncome             float64
```

Checking for Duplicates

- Used df.duplicated() to detect any duplicate rows.
- Confirmed that there were no duplicates in the dataset.

Check is there any Duplicates

```
[23]
duplicate_rows = df.duplicated().sum()
print({duplicate_rows})
```

```
{np.int64(0)}
```

Cleaning Categorical Columns

- Checked MaritalStatus, Occupation, and EducationLevel columns for inconsistencies.
- Stripped extra white spaces using str.strip() to ensure unique values.

Check the unique values and if there is any spaces before

```
print(df['EducationLevel'].unique())
print(df['Occupation'].unique())
print(df['MaritalStatus'].unique())
```

[11] ✓ 0.0s

```
['Bachelors' 'Partial College' 'High School' 'Partial High School'
 'Graduate Degree']
['Professional' 'Management' 'Skilled Manual' 'Clerical' 'Manual']
['M' 'S']
```

Final Result & Next Steps

The dataset is now fully cleaned, structured, and ready for further analysis or modeling.
All missing values were handled, data types were corrected, and categorical values were standardized.
This ensures accurate and reliable results for any upcoming data analysis or visualization tasks.

See You