TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
COLLEGE OF COMPUTER STUDIES
ITE 404 – Introduction to Data Science in Python
Prelim Assessment

Name: Jaspher F. Cadelina	**Date: 15/02/2025**
Section / Program: CS32s3	**Instructor: Ms. Hazel Patilano**
Laboratory Activity #4: Data Cleaning Using Pandas	

Objective:

The objective of this assessment is to develop students' proficiency in data cleaning using Python and Pandas. Students will learn to inspect datasets, handle duplicates, manage missing values, remove outliers, and standardize data formats. By completing these tasks, they will gain practical skills in preparing clean and structured datasets for analysis.

Instructions:

- Write Python code to perform the specified tasks using pandas.
- Use a dataset of your choice or create a sample dataset within your script.
- Ensure that your code is well-commented and properly structured.

Part 1: Data Inspection and Exploration

1. Load a dataset using pandas and display the first five rows.
2. Identify categorical and numerical columns in the dataset.
3. Count the number of unique values in categorical columns using .nunique().
4. Use value_counts() to display the frequency of unique values in a categorical column.

Part 2: Handling Duplicate Data

5. Check for duplicate rows using .duplicated() and print the total number of duplicate rows.
6. Remove duplicate rows using .drop_duplicates() and display the dataset shape before and after removing duplicates.

Part 3: Handling Missing Data

7. Identify missing values in the dataset and count them per column.
8. Drop rows with missing values and display the dataset shape before and after.
9. Impute missing values in numerical columns using the mean.

Part 4: Handling Outliers

10. Identify outliers in a numerical column using the IQR (Interquartile Range) method.

11. Remove rows containing outliers based on IQR.

Part 5: Final Data Cleaning Steps

12. Convert a categorical column to lowercase using .str.lower().
13. Use Python's built-in set() function to remove duplicate values from a categorical column.
14. Standardize column names (replace spaces with underscores and convert to lowercase).
15. Print the final cleaned dataset.

Submission Instructions:

- Complete the tasks above and write your answers in a Python file (e.g., data_cleaning_lastname.py).
- Include comments in your code explaining what each part does.
- After completing the code, run it and check the output to ensure everything works as expected.
- Capture a clear screenshot of your code and its corresponding output for each part of the activity, organize them logically (e.g., code first, followed by outputs), label them appropriately, and review to ensure everything is complete and readable before submission.
- Submit the PDF file for your task.

PART 1

In [4]:
```
import pandas as pd

# Specify the path to your CSV file
file_path = r'C:\Users\Jaspher\Documents\BRCA.csv'

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path)

df
```

Out[4]:

	Patient_ID	Age	Gender	Protein1	Protein2	Protein3	Protein4	Tumour_Stage	Histology	ER status	PR status	HER2 status	Surgery_type	Date_of_Surgery	Date_of_Last_Visit	Patient_Status
0	TCGA-D8-A1XD	36.0	FEMALE	0.080353	0.42638	0.54715	0.273680	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	15-Jan-17	19-Jun-17	Alive
1	TCGA-EW-A1OX	43.0	FEMALE	-0.420320	0.57807	0.61447	-0.031505	II	Mucinous Carcinoma	Positive	Positive	Negative	Lumpectomy	26-Apr-17	09-Nov-18	Dead
2	TCGA-A8-A079	69.0	FEMALE	0.213980	1.31140	-0.32747	-0.234260	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	08-Sep-17	09-Jun-18	Alive
3	TCGA-D8-A1XR	56.0	FEMALE	0.345090	-0.21147	-0.19304	0.124270	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	25-Jan-17	12-Jul-17	Alive
4	TCGA-BH-A0BF	56.0	FEMALE	0.221550	1.90680	0.52045	-0.311990	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	06-May-17	27-Jun-19	Dead
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
336	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
337	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
338	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
339	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
340	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

341 rows × 16 columns

In [5]:
```
# 2. Identify categorical and numerical columns
categorical_columns = df.select_dtypes(include=['object']).columns
numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns
```

In [6]:
```
categorical_columns
```

Out[6]:
```
Index(['Patient_ID', 'Gender', 'Tumour_Stage', 'Histology', 'ER status',
       'PR status', 'HER2 status', 'Surgery_type', 'Date_of_Surgery',
       'Date_of_Last_Visit', 'Patient_Status'],
      dtype='object')
```

In [7]:
```
numerical_columns
```

Out[7]:
```
Index(['Age', 'Protein1', 'Protein2', 'Protein3', 'Protein4'], dtype='object')
```

In [8]:
```
# 3. Count the number of unique values in categorical columns
print("\nNumber of unique values in categorical columns:")
for col in categorical_columns:
    print(f"{col}: {df[col].nunique()}")
```

```
Number of unique values in categorical columns:
Patient_ID: 334
Gender: 2
Tumour_Stage: 3
Histology: 3
ER status: 1
PR status: 1
HER2 status: 2
Surgery_type: 4
Date_of_Surgery: 181
Date_of_Last_Visit: 285
Patient_Status: 2
```

In [9]:
```
# 4. Display the frequency of unique values in a categorical column
print("\nFrequency of unique values in 'Tumour_Stage' column:")
print(df['Tumour_Stage'].value_counts())
```

```
Frequency of unique values in 'Tumour_Stage' column:
Tumour_Stage
II     189
III     81
I       64
Name: count, dtype: int64
```

In [10]:
```
# Part 2: Handling Duplicate Data
```

In [11]:
```
# 5. Check for duplicate rows and print the total number
duplicate_rows = df.duplicated().sum()
print(f"\nTotal number of duplicate rows: {duplicate_rows}")
```

```
Total number of duplicate rows: 6
```

In [12]:
```
# 6. Remove duplicate rows and display the dataset shape before and after
print("\nDataset shape before removing duplicates:", df.shape)
df = df.drop_duplicates()
print("Dataset shape after removing duplicates:", df.shape)
```

```
Dataset shape before removing duplicates: (341, 16)
Dataset shape after removing duplicates: (335, 16)
```

PART 3

```
Dataset shape after removing duplicates: (335, 16)
```

In [13]:
```
# Part 3: Handling Missing Data
```

In [14]:
```
# 7. Identify missing values and count them per column
```

In [14]:
```
# 7. Identify missing values and count them per column
print("\nMissing values per column:")
print(df.isnull().sum())
```

```
Missing values per column:
Patient_ID             1
Age                    1
Gender                 1
Protein1               1
Protein2               1
Protein3               1
Protein4               1
Tumour_Stage           1
Histology              1
ER status              1
PR status              1
HER2 status            1
Surgery_type           1
Date_of_Surgery        1
Date_of_Last_Visit    18
Patient_Status        14
dtype: int64
```

dtype: int64

In [15]:
```
# 8. Drop rows with missing values and display the dataset shape before and after
print("\nDataset shape before dropping rows with missing values:", df.shape)
df = df.dropna()
print("Dataset shape after dropping rows with missing values:", df.shape)
```

```
Dataset shape before dropping rows with missing values: (335, 16)
Dataset shape after dropping rows with missing values: (317, 16)
```

In [17]:
```
# 9. Impute missing values in numerical columns using the mean
# Re-introduce missing values for demonstration (if needed)
# df.loc[2:4, 'Age'] = np.nan
# df.loc[5, 'Protein1'] = np.nan

# Corrected imputation without chained assignments
df['Age'] = df['Age'].fillna(df['Age'].mean())
df['Protein1'] = df['Protein1'].fillna(df['Protein1'].mean())
df['Protein2'] = df['Protein2'].fillna(df['Protein2'].mean())
df['Protein3'] = df['Protein3'].fillna(df['Protein3'].mean())
df['Protein4'] = df['Protein4'].fillna(df['Protein4'].mean())

print("\nDataset after imputing missing values in numerical columns:")
print(df.head())
```

```
Dataset after imputing missing values in numerical columns:
     Patient_ID   Age  Gender  Protein1  Protein2  Protein3  Protein4  \
0  TCGA-D8-A1XD  36.0  FEMALE  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0  FEMALE -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0  FEMALE  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0  FEMALE  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0  FEMALE  0.221550   1.90680   0.52045 -0.311990

  Tumour_Stage                     Histology ER status PR status HER2 status  \
0          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
1           II             Mucinous Carcinoma  Positive  Positive    Negative
2          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
3           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
4           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative

                 Surgery_type Date_of_Surgery Date_of_Last_Visit  \
0  Modified Radical Mastectomy       15-Jan-17          19-Jun-17
1                  Lumpectomy       26-Apr-17          09-Nov-18
2                       Other       08-Sep-17          09-Jun-18
3  Modified Radical Mastectomy       25-Jan-17          12-Jul-17
4                       Other       06-May-17          27-Jun-19

  Patient_Status
0          Alive
1           Dead
2          Alive
3          Alive
4           Dead
```

PART 4

```
4           Dead
```

```
In [18]: # Part 4: Handling Outliers
```

```
In [19]: # 10. Identify outliers in a numerical column using the IQR method
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = df[(df['Age'] < lower_bound) | (df['Age'] > upper_bound)]
print("\nOutliers in 'Age' column:")
print(outliers)
```

```
Outliers in 'Age' column:
Empty DataFrame
Columns: [Patient_ID, Age, Gender, Protein1, Protein2, Protein3, Protein4, Tumour_Stage, Histology, ER status, PR status, HER2 status, Surgery_type, Date_of_Surgery, Date_of_Last_Visit, Patient_St
atus]
Index: []
```

```
In [20]: # 11. Remove rows containing outliers based on IQR
df = df[(df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)]
print("\nDataset shape after removing outliers:", df.shape)
```

```
Dataset shape after removing outliers: (317, 16)
```

```
In [21]: # Part 5: Final Data Cleaning Steps
```

PART 5

```
In [21]: # Part 5: Final Data Cleaning Steps
```

```python
# 12. Convert a categorical column to lowercase
df['Gender'] = df['Gender'].str.lower()
print("\nDataset after converting 'Gender' column to lowercase:")
print(df.head())
```

```
Dataset after converting 'Gender' column to lowercase:
     Patient_ID   Age  Gender  Protein1  Protein2  Protein3  Protein4  \
0  TCGA-D8-A1XD  36.0  female  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0  female -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0  female  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0  female  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0  female  0.221550   1.90680   0.52045 -0.311990

  Tumour_Stage                      Histology ER status PR status HER2 status  \
0          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
1           II             Mucinous Carcinoma  Positive  Positive    Negative
2          III  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
3           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative
4           II  Infiltrating Ductal Carcinoma  Positive  Positive    Negative

                 Surgery_type Date_of_Surgery Date_of_Last_Visit  \
0  Modified Radical Mastectomy       15-Jan-17          19-Jun-17
1                   Lumpectomy       26-Apr-17          09-Nov-18
2                        Other       08-Sep-17          09-Jun-18
3  Modified Radical Mastectomy       25-Jan-17          12-Jul-17
4                        Other       06-May-17          27-Jun-19

  Patient_Status
0          Alive
1           Dead
2          Alive
3          Alive
4           Dead
```

In [22]:
```
# 13. Remove duplicate values from a categorical column using set()
df['Histology'] = df['Histology'].apply(lambda x: ' '.join(set(x.split())))
print("\nDataset after removing duplicate values from 'Histology' column:")
print(df.head())
```

```
Dataset after removing duplicate values from 'Histology' column:
     Patient_ID   Age  Gender  Protein1  Protein2  Protein3  Protein4  \
0  TCGA-D8-A1XD  36.0  female  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0  female -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0  female  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0  female  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0  female  0.221550   1.90680   0.52045 -0.311990

  Tumour_Stage                     Histology ER status PR status HER2 status  \
0          III  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
1           II             Carcinoma Mucinous  Positive  Positive    Negative
2          III  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
3           II  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
4           II  Ductal Infiltrating Carcinoma  Positive  Positive    Negative

                 Surgery_type Date_of_Surgery Date_of_Last_Visit  \
0  Modified Radical Mastectomy       15-Jan-17          19-Jun-17
1                  Lumpectomy       26-Apr-17          09-Nov-18
2                       Other       08-Sep-17          09-Jun-18
3  Modified Radical Mastectomy       25-Jan-17          12-Jul-17
4                       Other       06-May-17          27-Jun-19

  Patient_Status
0          Alive
1           Dead
2          Alive
3          Alive
4           Dead
```

In [23]:

In [23]:
```
# 14. Standardize column names (replace spaces with underscores and convert to lowercase)
df.columns = df.columns.str.replace(' ', '_').str.lower()
print("\nDataset after standardizing column names:")
print(df.head())
```

```
Dataset after standardizing column names:
     patient_id   age  gender  protein1  protein2  protein3  protein4  \
0  TCGA-D8-A1XD  36.0  female  0.080353   0.42638   0.54715  0.273680
1  TCGA-EW-A1OX  43.0  female -0.420320   0.57807   0.61447 -0.031505
2  TCGA-A8-A079  69.0  female  0.213980   1.31140  -0.32747 -0.234260
3  TCGA-D8-A1XR  56.0  female  0.345090  -0.21147  -0.19304  0.124270
4  TCGA-BH-A0BF  56.0  female  0.221550   1.90680   0.52045 -0.311990

  tumour_stage                     histology er_status pr_status her2_status  \
0          III  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
1           II             Carcinoma Mucinous  Positive  Positive    Negative
2          III  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
3           II  Ductal Infiltrating Carcinoma  Positive  Positive    Negative
4           II  Ductal Infiltrating Carcinoma  Positive  Positive    Negative

                 surgery_type date_of_surgery date_of_last_visit  \
0  Modified Radical Mastectomy       15-Jan-17          19-Jun-17
1                  Lumpectomy       26-Apr-17          09-Nov-18
2                       Other       08-Sep-17          09-Jun-18
3  Modified Radical Mastectomy       25-Jan-17          12-Jul-17
4                       Other       06-May-17          27-Jun-19

  patient_status
0          Alive
1           Dead
2          Alive
3          Alive
4           Dead
```

```
In [24]: # 15. Print the final cleaned dataset
         print("\nFinal cleaned dataset:")
         print(df)
```

```
Final cleaned dataset:
      patient_id   age  gender  protein1  protein2  protein3  protein4  \
0    TCGA-D8-A1XD  36.0  female  0.080353   0.42638   0.54715  0.273680
1    TCGA-EW-A1OX  43.0  female -0.420320   0.57807   0.61447 -0.031505
2    TCGA-A8-A079  69.0  female  0.213980   1.31140  -0.32747 -0.234260
3    TCGA-D8-A1XR  56.0  female  0.345090  -0.21147  -0.19304  0.124270
4    TCGA-BH-A0BF  56.0  female  0.221550   1.90680   0.52045 -0.311990
..            ...   ...     ...       ...       ...       ...       ...
329  TCGA-AN-A04A  36.0  female  0.231800   0.61804  -0.55779 -0.517350
330  TCGA-A8-A085  44.0    male  0.732720   1.11170  -0.26952 -0.354920
331  TCGA-A1-A0SG  61.0  female -0.719470   2.54850  -0.15024  0.339680
332  TCGA-A2-A0EU  79.0  female  0.479400   2.05590  -0.53136 -0.188480
333  TCGA-B6-A40B  76.0  female -0.244270   0.92556  -0.41823 -0.067848

    tumour_stage                      histology er_status pr_status  \
0            III  Ductal Infiltrating Carcinoma  Positive  Positive
```

In [25]:
```
df
```

Out[25]:

	patient_id	age	gender	protein1	protein2	protein3	protein4	tumour_stage	histology	er_status	pr_status	her2_status	surgery_type	date_of_surgery	date_of_last_visit	patient_status
0	TCGA-D8-A1XD	36.0	female	0.080353	0.42638	0.54715	0.273680	III	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	15-Jan-17	19-Jun-17	Alive
1	TCGA-EW-A1OX	43.0	female	-0.420320	0.57807	0.61447	-0.031505	II	Carcinoma Mucinous	Positive	Positive	Negative	Lumpectomy	26-Apr-17	09-Nov-18	Dead
2	TCGA-A8-A079	69.0	female	0.213980	1.31140	-0.32747	-0.234260	III	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Other	08-Sep-17	09-Jun-18	Alive
3	TCGA-D8-A1XR	56.0	female	0.345090	-0.21147	-0.19304	0.124270	II	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	25-Jan-17	12-Jul-17	Alive
4	TCGA-BH-A0BF	56.0	female	0.221550	1.90680	0.52045	-0.311990	II	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Other	06-May-17	27-Jun-19	Dead
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
329	TCGA-AN-A04A	36.0	female	0.231800	0.61804	-0.55779	-0.517350	III	Ductal Infiltrating Carcinoma	Positive	Positive	Positive	Simple Mastectomy	11-Nov-19	09-Feb-20	Dead
330	TCGA-A8-A085	44.0	male	0.732720	1.11170	-0.26952	-0.354920	II	Infiltrating Lobular Carcinoma	Positive	Positive	Negative	Other	01-Nov-19	04-Mar-20	Dead
331	TCGA-A1-A0SG	61.0	female	-0.719470	2.54850	-0.15024	0.339680	II	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Lumpectomy	11-Nov-19	18-Jan-21	Dead
332	TCGA-A2-A0EU	79.0	female	0.479400	2.05590	-0.53136	-0.188480	I	Ductal Infiltrating Carcinoma	Positive	Positive	Positive	Lumpectomy	21-Nov-19	19-Feb-21	Dead
333	TCGA-B6-A40B	76.0	female	-0.244270	0.92556	-0.41823	-0.067848	I	Ductal Infiltrating Carcinoma	Positive	Positive	Negative	Lumpectomy	11-Nov-19	05-Jan-21	Dead

317 rows × 16 columns

Honor Pledge for Graded Assignments (Recommended):

“I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own.”