LAPORAN TUGAS PRAKTIKUM

“BIG DATA AND PREDICTIVE ANALYSIS”

“Data Visualization”

PERTEMUAN KE :

NAMA	Muhamad Dekhsa Afnan
NIM	23.61.0245
Dosen Pengampu	Ainul Yaqin
Nama Asisten Praktikum	
Kelas	23-BCI-01

Koor Asisten Praktikum

Tanggal :

NAMA :
Ttd :

1. Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Imports essential Python libraries for data manipulation (pandas, numpy) and visualization (matplotlib, seaborn).

2. Import Datasets

```
df = pd.read_csv('datasets/cancer patient data sets.csv')
df
```

	index	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	...	Fatigue	Weight Loss	Shortness of Breath	Wheezing	Swallowing Difficulty	Clubbing of Finger Nails
0	0	P1	33	1	2	4	5	4	3	2	...	3	4	2	2	3	1
1	1	P10	17	1	3	1	5	3	4	2	...	1	3	7	8	6	2
2	2	P100	35	1	4	5	6	5	5	4	...	8	7	9	2	1	4
3	3	P1000	37	1	7	7	7	7	6	7	...	4	2	3	1	4	5
4	4	P101	46	1	6	8	7	7	7	6	...	3	2	4	1	4	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	995	P995	44	1	6	7	7	7	7	6	...	5	3	2	7	8	2
996	996	P996	37	2	6	8	7	7	7	6	...	9	6	5	7	2	4
997	997	P997	25	2	4	5	6	5	5	4	...	8	7	9	2	1	4
998	998	P998	18	2	6	8	7	7	7	6	...	3	2	4	1	4	2
999	999	P999	47	1	6	5	6	5	5	4	...	8	7	9	2	1	4

1000 rows × 26 columns

Reads the lung cancer dataset from a CSV file into a pandas DataFrame called df and displays its contents.

3. Data Overview

```
# Data types and missing values
df.info()
df.isnull().sum()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 26 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   index                   1000 non-null   int64
 1   Patient Id              1000 non-null   object
 2   Age                     1000 non-null   int64
 3   Gender                  1000 non-null   int64
 4   Air Pollution           1000 non-null   int64
 5   Alcohol use             1000 non-null   int64
 6   Dust Allergy            1000 non-null   int64
 7   OccuPational Hazards    1000 non-null   int64
 8   Genetic Risk            1000 non-null   int64
 9   chronic Lung Disease    1000 non-null   int64
 10  Balanced Diet           1000 non-null   int64
 11  Obesity                 1000 non-null   int64
 12  Smoking                 1000 non-null   int64
 13  Passive Smoker          1000 non-null   int64
 14  Chest Pain              1000 non-null   int64
 15  Coughing of Blood       1000 non-null   int64
 16  Fatigue                 1000 non-null   int64
 17  Weight Loss             1000 non-null   int64
 18  Shortness of Breath     1000 non-null   int64
 19  Wheezing                1000 non-null   int64
...
 24  Snoring                 1000 non-null   int64
 25  Level                   1000 non-null   object
dtypes: int64(24), object(2)
memory usage: 203.3+ KB
```

df.info() shows the data types and non-null counts for each column.

df.isnull().sum() counts missing values in each column.

```
# Descriptive statistics for all columns
df.describe(include='all')
```

✓ 0.0s

	index	Patient Id	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	...	Fatigue	Weight Loss
count	1000.000000	1000	1000.000000	1000.000000	1000.0000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	...	1000.000000	1000.000000
unique	NaN	1000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
top	NaN	P999	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
freq	NaN	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
mean	499.500000	NaN	37.174000	1.402000	3.8400	4.563000	5.165000	4.840000	4.580000	4.380000	...	3.856000	3.855000
std	288.819436	NaN	12.005493	0.490547	2.0304	2.620477	1.980833	2.107805	2.126999	1.848518	...	2.244616	2.206546
min	0.000000	NaN	14.000000	1.000000	1.0000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000
25%	249.750000	NaN	27.750000	1.000000	2.0000	2.000000	4.000000	3.000000	2.000000	3.000000	...	2.000000	2.000000
50%	499.500000	NaN	36.000000	1.000000	3.0000	5.000000	6.000000	5.000000	5.000000	4.000000	...	3.000000	3.000000
75%	749.250000	NaN	45.000000	2.000000	6.0000	7.000000	7.000000	7.000000	7.000000	6.000000	...	5.000000	6.000000
max	999.000000	NaN	73.000000	2.000000	8.0000	8.000000	8.000000	8.000000	7.000000	7.000000	...	9.000000	8.000000

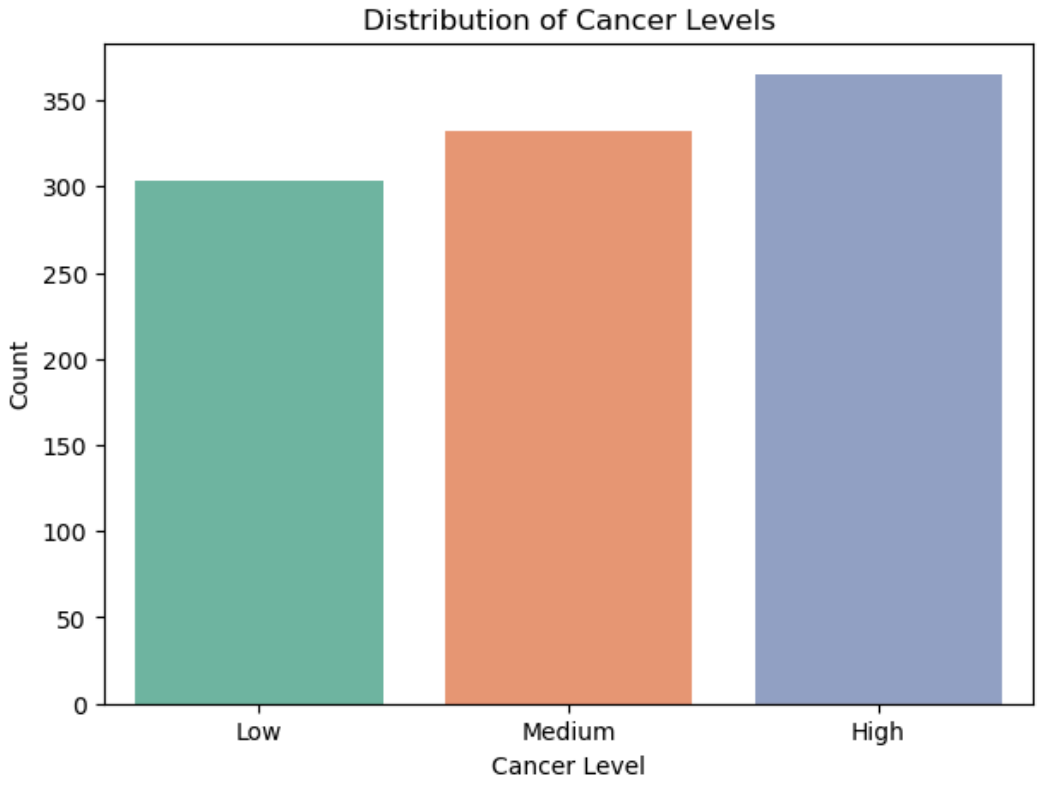
11 rows × 26 columns

Displays summary statistics (mean, std, min, max, etc.) for all columns, including categorical ones.

4. Visualization

```
plt.figure(figsize=(7,5))
sns.countplot(data=df, x='Level', palette='Set2')
plt.title('Distribution of Cancer Levels')
plt.xlabel('Cancer Level')
plt.ylabel('Count')
plt.show()
```

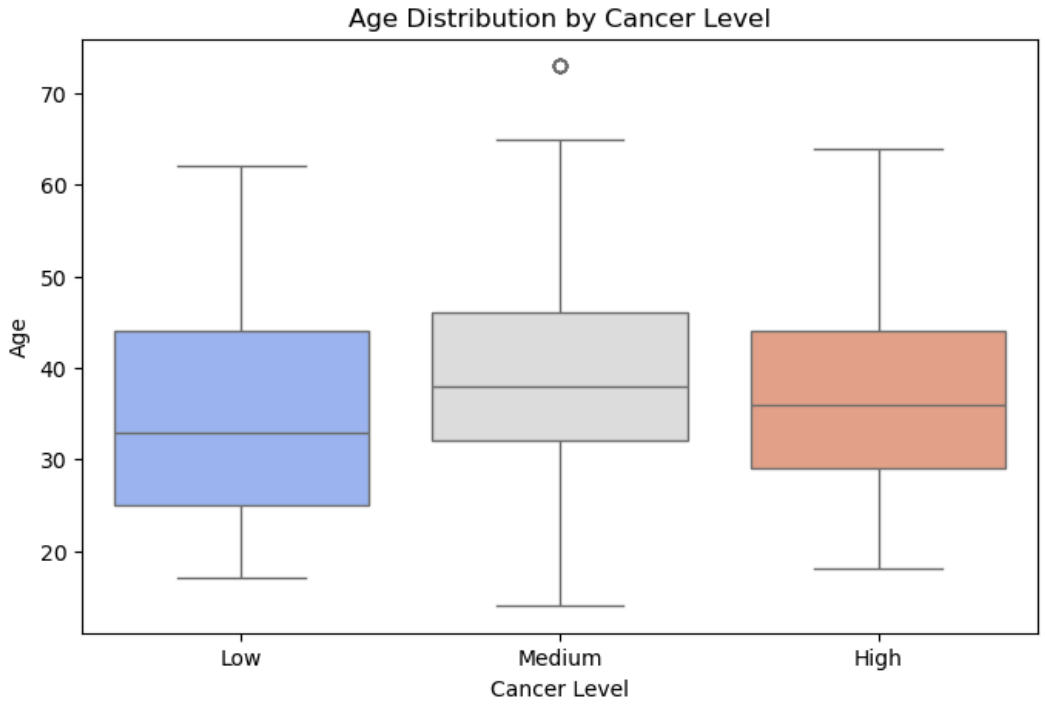
✓ 0.1s



Plots the distribution of cancer levels using a countplot for visualizing class balance.

```
plt.figure(figsize=(8,5))
sns.boxplot(data=df, x='Level', y='Age', palette='coolwarm')
plt.title('Age Distribution by Cancer Level')
plt.xlabel('Cancer Level')
plt.ylabel('Age')
plt.show()
```

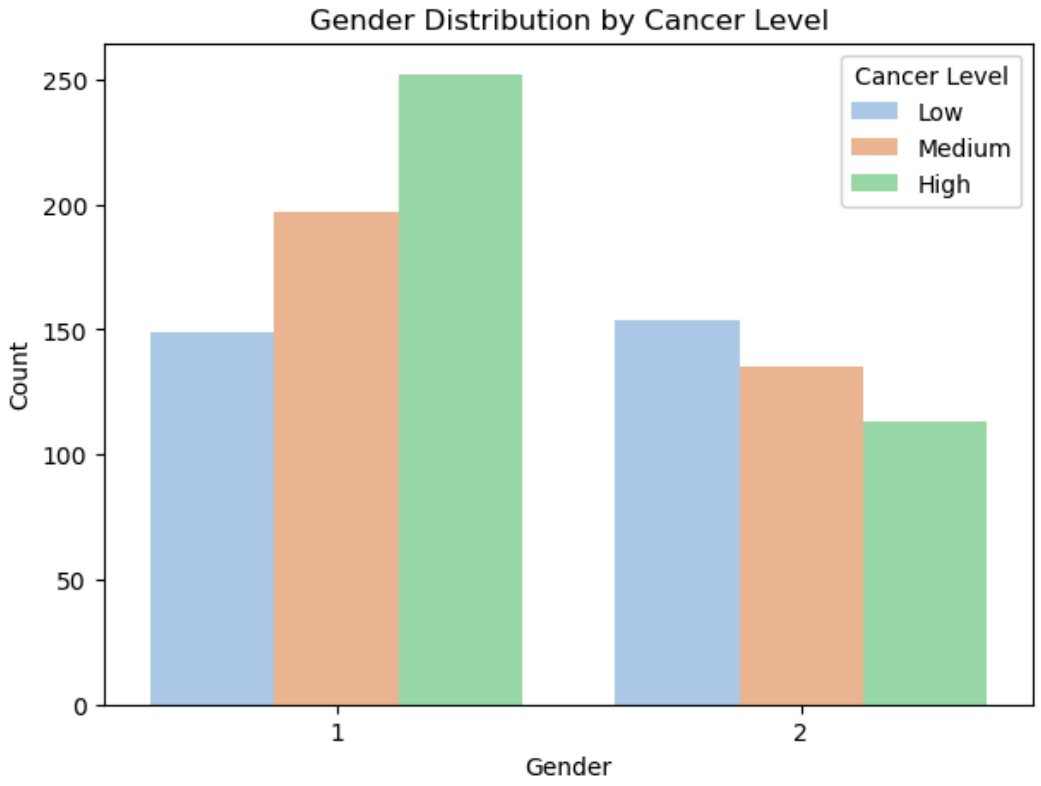
✓ 0.2s



Shows the age distribution for each cancer level using a boxplot.

```
plt.figure(figsize=(7,5))
sns.countplot(data=df, x='Gender', hue='Level', palette='pastel')
plt.title('Gender Distribution by Cancer Level')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Cancer Level')
plt.show()
```

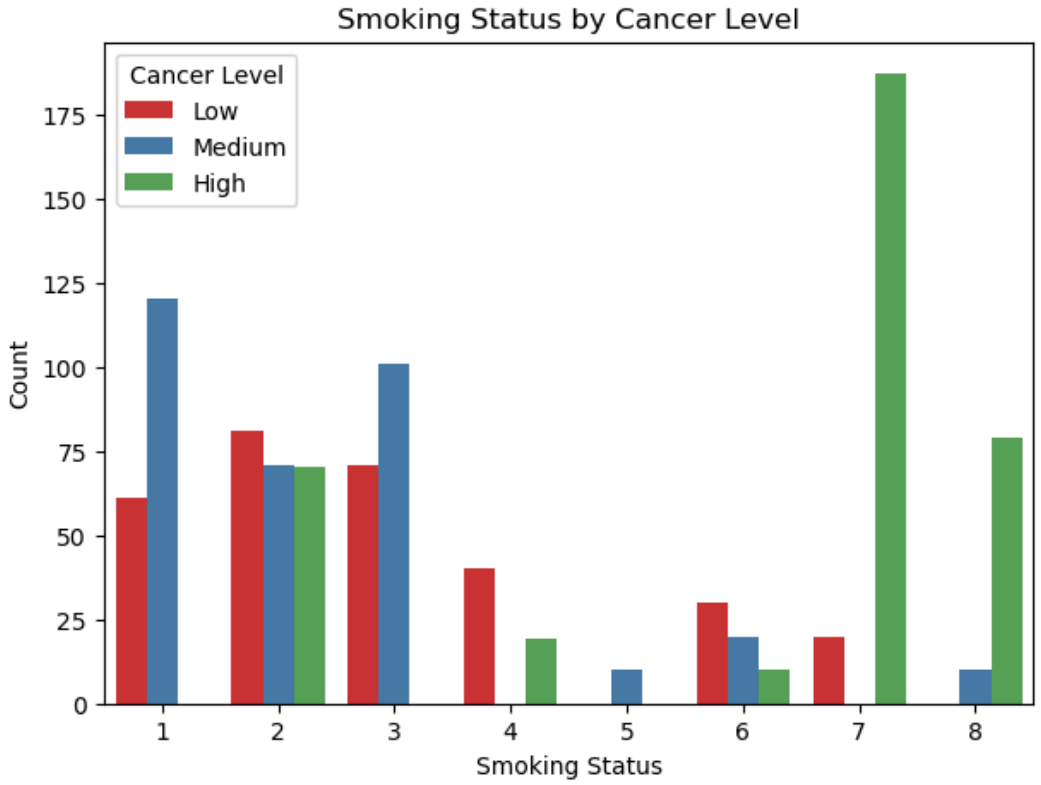
✓ 0.1s



Visualizes the distribution of genders across different cancer levels.

```python
if 'Smoking' in df.columns:
    plt.figure(figsize=(7,5))
    sns.countplot(data=df, x='Smoking', hue='Level', palette='Set1')
    plt.title('Smoking Status by Cancer Level')
    plt.xlabel('Smoking Status')
    plt.ylabel('Count')
    plt.legend(title='Cancer Level')
    plt.show()
else:
    print('Smoking column not found.')
```

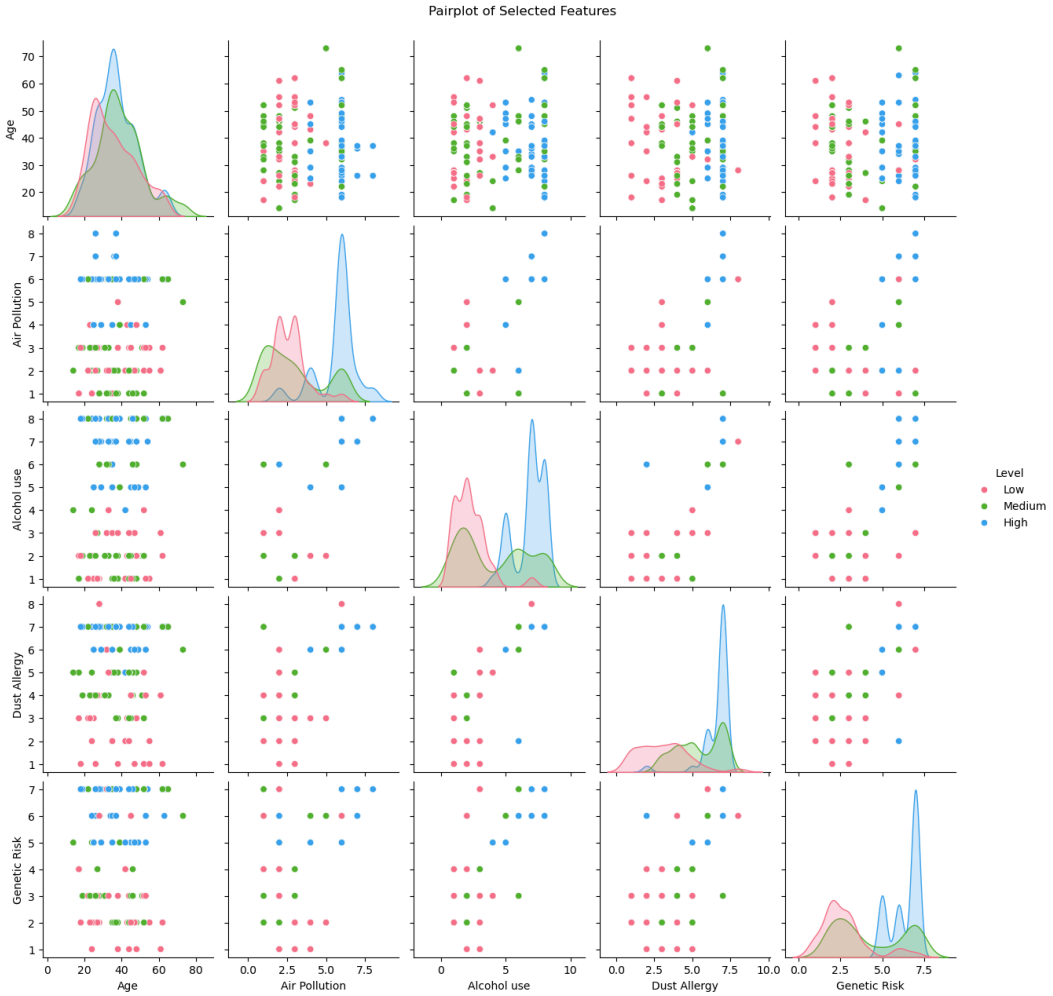
✓ 0.2s



Shows the relationship between smoking status and cancer level, if the column exists.

```python
selected_cols = ['Age', 'Air Pollution', 'Alcohol use', 'Dust Allergy', 'Genetic Risk', 'Level']
existing_cols = [col for col in selected_cols if col in df.columns]
if len(existing_cols) > 1:
    sns.pairplot(df[existing_cols], hue='Level', palette='husl')
    plt.suptitle('Pairplot of Selected Features', y=1.02)
    plt.show()
else:
    print('Not enough columns for pairplot.')
```

✓ 4.5s



Creates a pairplot for selected features to visualize pairwise relationships, colored by cancer level.