Programming for AI
Assignment#2

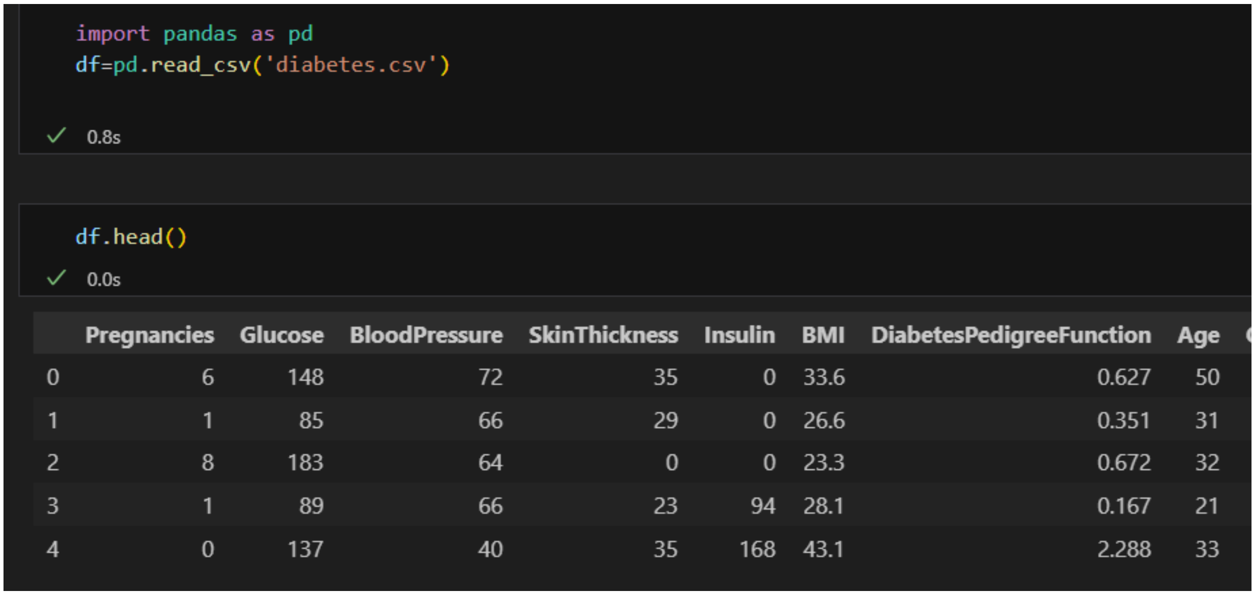
- **Aiman Arif**
- **22p-9262**

Data Analysis using Pandas

1. Data Collection:

Collected a dataset of diabetes from kaggle,it contains information about diabetic patients, including demographics, diagnosis, medications, and hospital outcomes.

2. Data Loading:



```python
import pandas as pd
df=pd.read_csv('diabetes.csv')
```

✓ 0.8s

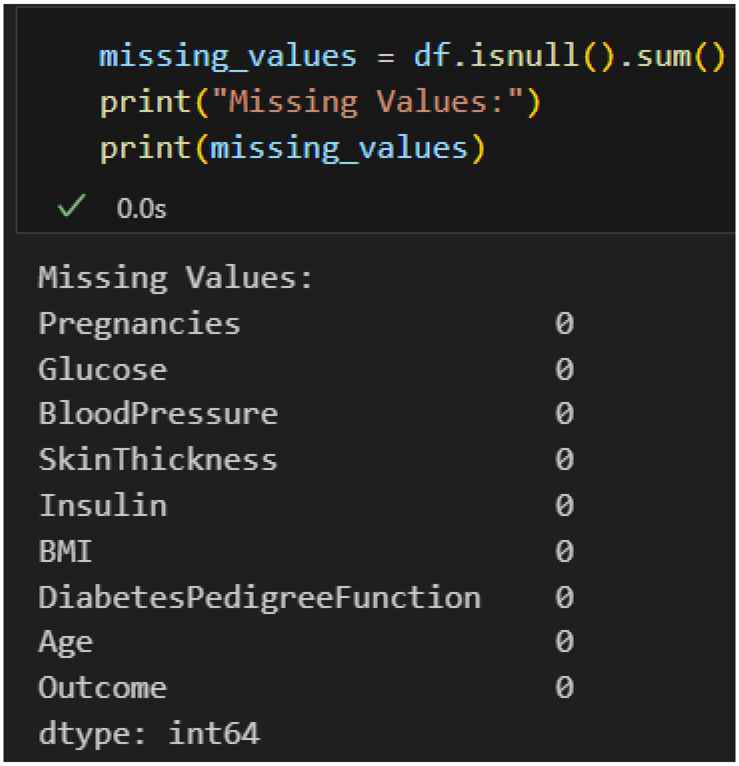
```python
df.head()
```

✓ 0.0s

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33

3. Data Cleaning:

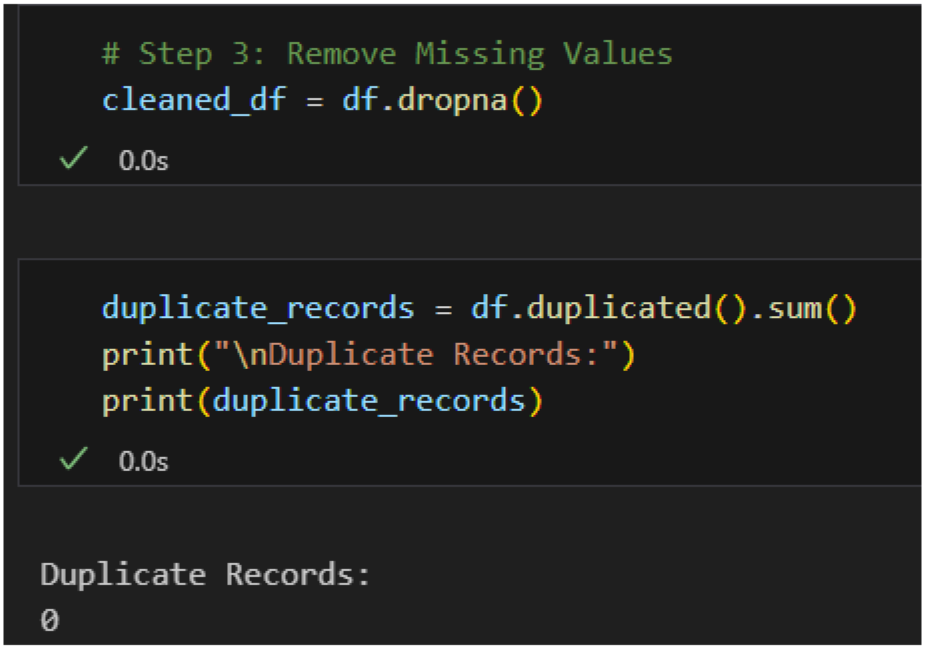
Remove missing values, duplicate records, and outliers from the loaded dataframe.



```
missing_values = df.isnull().sum()
print("Missing Values:")
print(missing_values)
```

✓ 0.0s

```
Missing Values:
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```



```
# Step 3: Remove Missing Values
cleaned_df = df.dropna()
```

✓ 0.0s

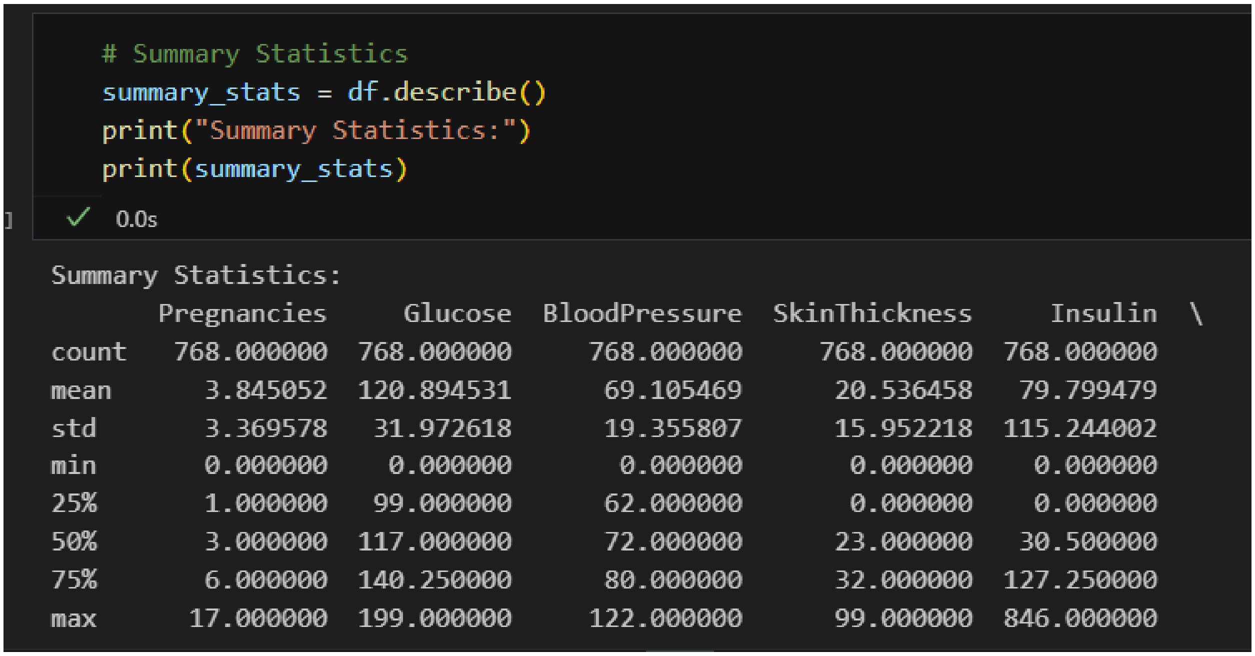
```
duplicate_records = df.duplicated().sum()
print("\nDuplicate Records:")
print(duplicate_records)
```

✓ 0.0s

```

Duplicate Records:
0
```

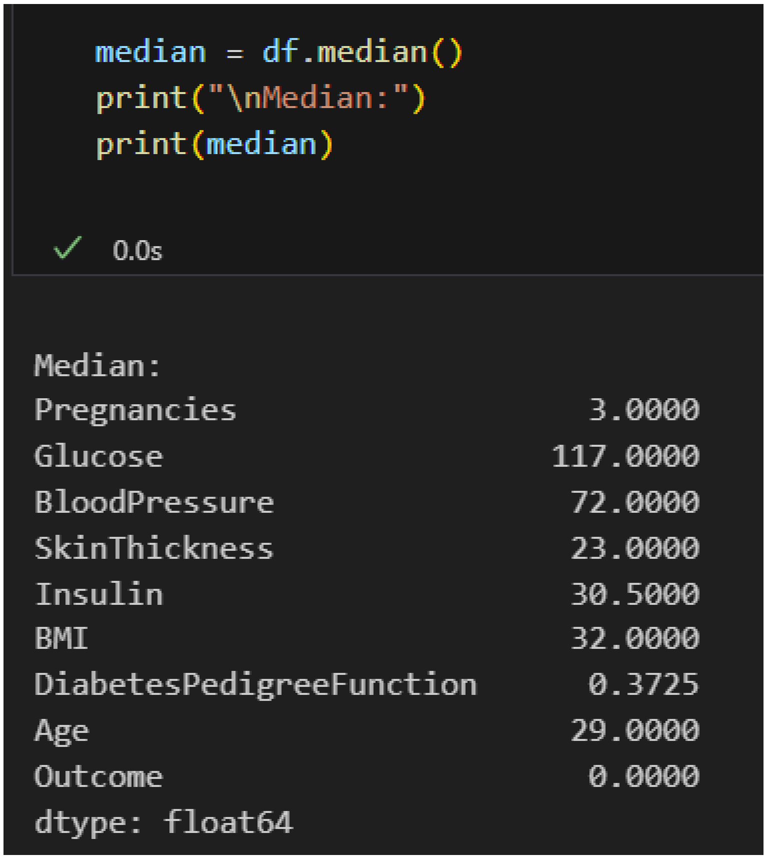
4. Statistical Analysis:



```
# Summary Statistics
summary_stats = df.describe()
print("Summary Statistics:")
print(summary_stats)
```

✓ 0.0s

```
Summary Statistics:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000   
mean      3.845052  120.894531      69.105469      20.536458   79.799479   
std       3.369578   31.972618      19.355807      15.952218  115.244002   
min       0.000000    0.000000       0.000000       0.000000    0.000000   
25%       1.000000   99.000000      62.000000       0.000000    0.000000   
50%       3.000000  117.000000      72.000000      23.000000   30.500000   
75%       6.000000  140.250000      80.000000      32.000000  127.250000   
max      17.000000  199.000000     122.000000      99.000000  846.000000   
```



```
median = df.median()
print("\nMedian:")
print(median)
```

✓ 0.0s

```
Median:
Pregnancies                   3.0000
Glucose                     117.0000
BloodPressure                72.0000
SkinThickness                23.0000
Insulin                      30.5000
BMI                          32.0000
DiabetesPedigreeFunction      0.3725
Age                          29.0000
Outcome                       0.0000
dtype: float64
```