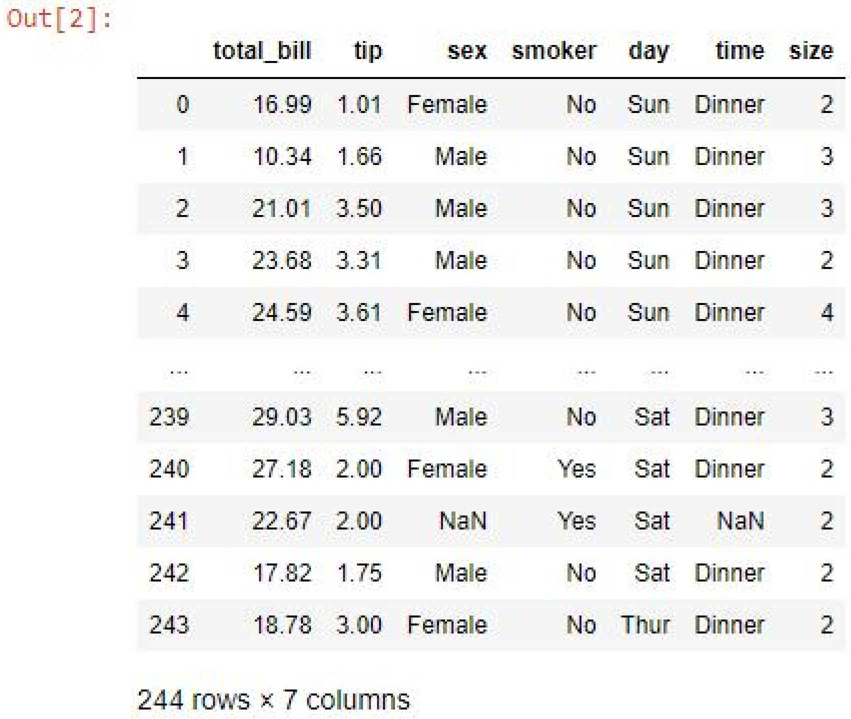
Python for Data Analysis

1. **Data Cleaning: Write a Python script that reads a CSV file using Pandas, drops rows with missing values, and outputs the cleaned data.**

```
#Import libraries
import pandas as pd

# Read the CSV file
data = pd.read_csv('tips_uncleaned.csv')
data
```

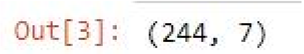


Out[2]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	NaN	Yes	Sat	NaN	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

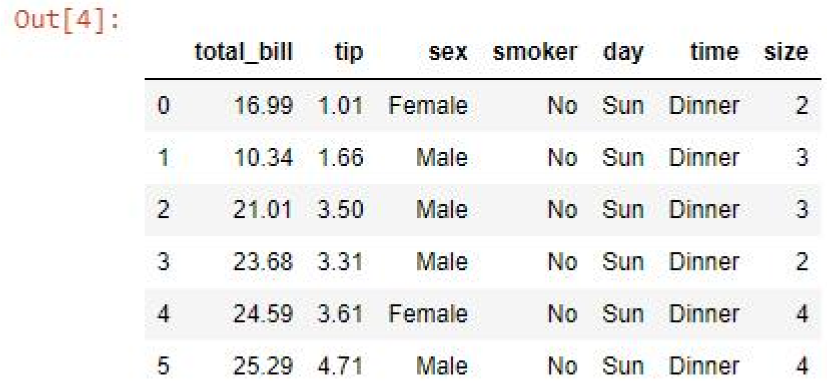
244 rows × 7 columns

```
data.shape
```



```
Out[3]: (244, 7)
```

```
data.head(6)
```

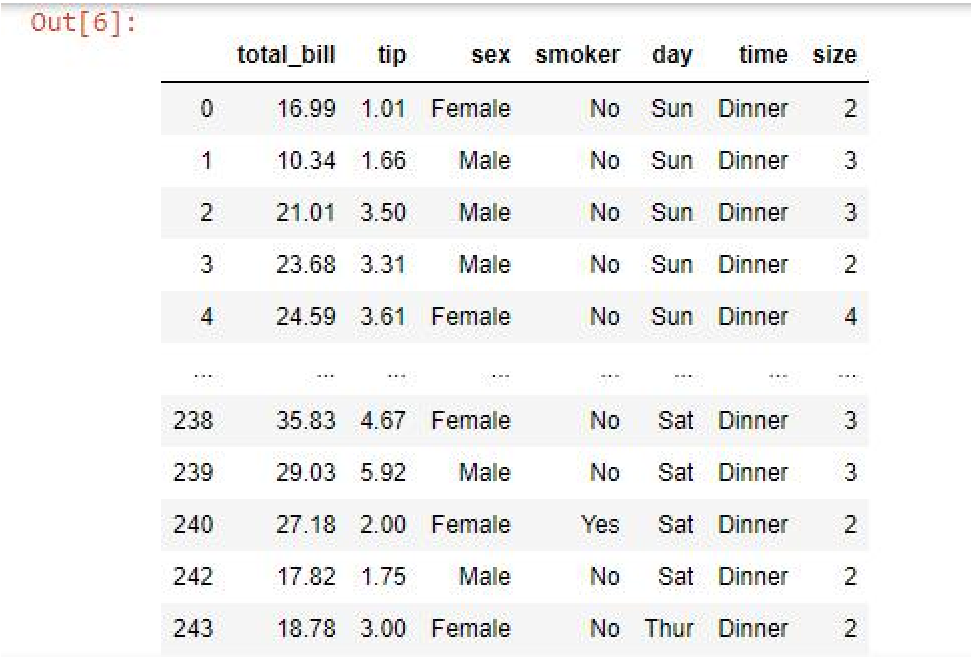


Out[4]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4

```
# Drop rows with missing values
```

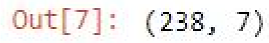
```
cleaned_data = data.dropna()
cleaned_data
```



Out[6]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
238	35.83	4.67	Female	No	Sat	Dinner	3
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

```
cleaned_data.shape
```



```
Out[7]: (238, 7)
```

```
# Save the cleaned data to a new CSV file
cleaned_data.to_csv('tips_cleaned.csv', index=False)
```

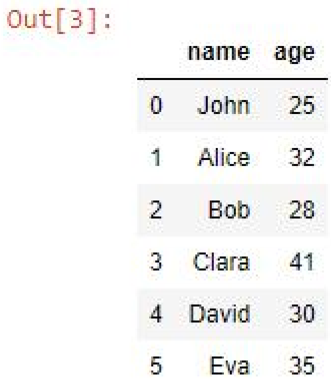
2. **Data Manipulation: Using Pandas, write a Python function that takes a DataFrame and returns the top 5 rows where a specific column (e.g., "age") has values greater than 30.**

```
#Import libraries
import pandas as pd

# Sample data for demonstration
data = {'name': ['John', 'Alice', 'Bob', 'Clara', 'David', 'Eva'],
        'age': [25, 32, 28, 41, 30, 35]}

# Create a DataFrame
df = pd.DataFrame(data)

# Display DataFrame
Df
```



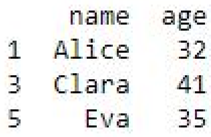
Out[3]:

	name	age
0	John	25
1	Alice	32
2	Bob	28
3	Clara	41
4	David	30
5	Eva	35

```
# Function to filter and return top 5 rows where a specific column has values
greater than a threshold
def filter_top_rows(df, column_name, threshold):
    # Filter rows where the values in the specified column are greater than the
threshold
    filtered_df = df[df[column_name] > threshold]
    # Return the top 5 rows
        return filtered_df.head(5)

# Filter top 5 rows where 'age' > 30
result = filter_top_rows(df, 'age', 30)

# Display the result
print(result)
```



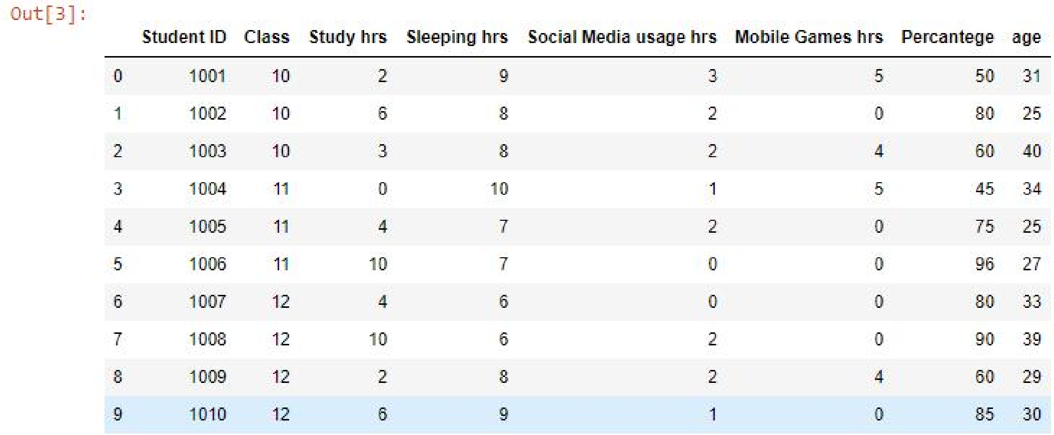
```
    name  age
1  Alice   32
3  Clara   41
5    Eva   35
```

3. Data Visualization: Create a bar chart using Matplotlib to visualize the distribution of user ages from a dataset.

```
#Import libraries
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset from CSV file
df = pd.read_csv('student_results.csv')

df
```



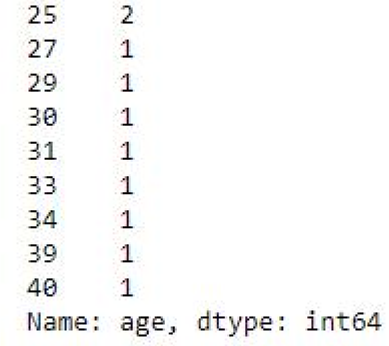
Out[3]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege	age
0	1001	10	2	9	3	5	50	31
1	1002	10	6	8	2	0	80	25
2	1003	10	3	8	2	4	60	40
3	1004	11	0	10	1	5	45	34
4	1005	11	4	7	2	0	75	25
5	1006	11	10	7	0	0	96	27
6	1007	12	4	6	0	0	80	33
7	1008	12	10	6	2	0	90	39
8	1009	12	2	8	2	4	60	29
9	1010	12	6	9	1	0	85	30

```
# Get the frequency distribution of ages
age_distribution = df['age'].value_counts()

# Sort the distribution by age values for a better visual representation
age_distribution = age_distribution.sort_index()

# Display the age distribution (for understanding)
print(age_distribution)
```

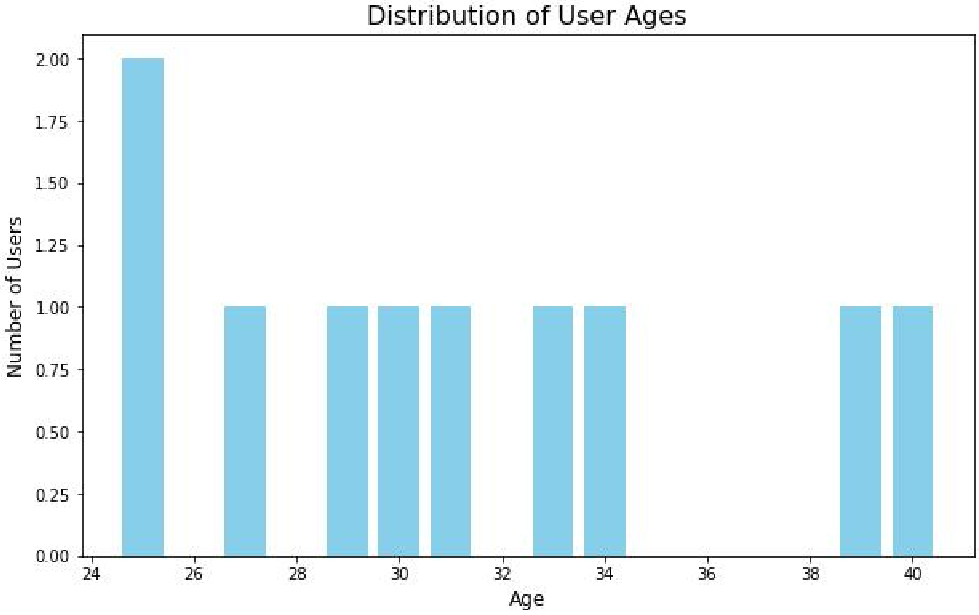


```
25    2
27    1
29    1
30    1
31    1
33    1
34    1
39    1
40    1
Name: age, dtype: int64
```

```
# Plot a bar chart
plt.figure(figsize=(10, 6))  # Set the size of the figure
plt.bar(age_distribution.index, age_distribution.values, color='skyblue')

# Add titles and labels
plt.title('Distribution of User Ages', fontsize=16)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Number of Users', fontsize=12)

# Display the bar chart
plt.show()
```



```
# Save the figure as a PNG file
plt.savefig('age_distribution.png')
```

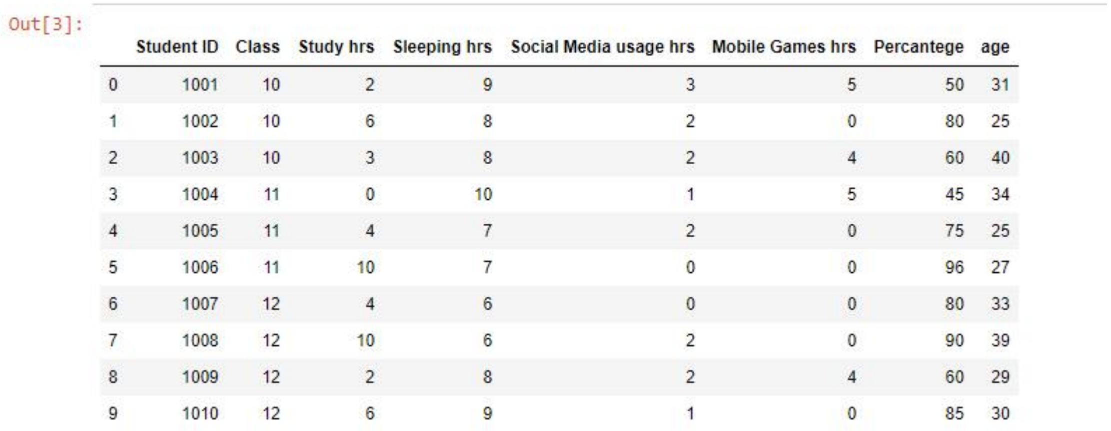
```
<Figure size 432x288 with 0 Axes>
```

4. Descriptive Statistics: Using NumPy and Pandas, write a script that calculates the mean, median, and standard deviation of a column (e.g., "age") in a dataset.

```
#Import libraries
import numpy as np
import pandas as pd

# Load dataset from CSV file
df = pd.read_csv('student_results.csv')

df
```

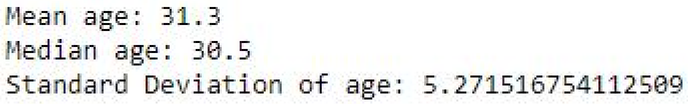


Out[3]:

	Student ID	Class	Study hrs	Sleeping hrs	Social Media usage hrs	Mobile Games hrs	Percantege	age
0	1001	10	2	9	3	5	50	31
1	1002	10	6	8	2	0	80	25
2	1003	10	3	8	2	4	60	40
3	1004	11	0	10	1	5	45	34
4	1005	11	4	7	2	0	75	25
5	1006	11	10	7	0	0	96	27
6	1007	12	4	6	0	0	80	33
7	1008	12	10	6	2	0	90	39
8	1009	12	2	8	2	4	60	29
9	1010	12	6	9	1	0	85	30

```
# Calculate mean, median, and standard deviation using Pandas
mean_age = df['age'].mean()          # Mean
median_age = df['age'].median()      # Median
std_dev_age = df['age'].std()        # Standard Deviation

# Display the results
print(f"Mean age: {mean_age}")
print(f"Median age: {median_age}")
print(f"Standard Deviation of age: {std_dev_age}")
```



```
Mean age: 31.3
Median age: 30.5
Standard Deviation of age: 5.271516754112509
```