

Import the necessary libraries to perform the processing and plotting.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
```

Load and isolate the Married column needed for the univariate analysis.

```
raw_data = pd.read_csv("./data/raw_data.csv")
marital_data = raw_data['Married']
```

Find the mode of the data and prepare it to replace missing data with the mode.

```
mode = marital_data.mode().values[0]
if mode == "Yes":
    mode = 1
elif mode == "No":
    mode = 0
```

Encode the Married column to save space and improve processing speeds.

```
le = LabelEncoder()
encoded_marital = le.fit_transform(marital_data)
```

Replace every instance of missing data (transformed to 2) with the mode of the data (in this case Yes or 0 after encoded).

```
encoded_marital = np.where(encoded_marital == 2, mode,
                           encoded_marital)
```

Now we can continue to count the number of each instance of Yes(401) or No(213) = 614.

```
a = np.array(encoded_marital)
b = np.unique(a, return_counts=True)
yes_totals = b[1][1]
no_totals = b[1][0]
marital_totals = yes_totals, no_totals
```

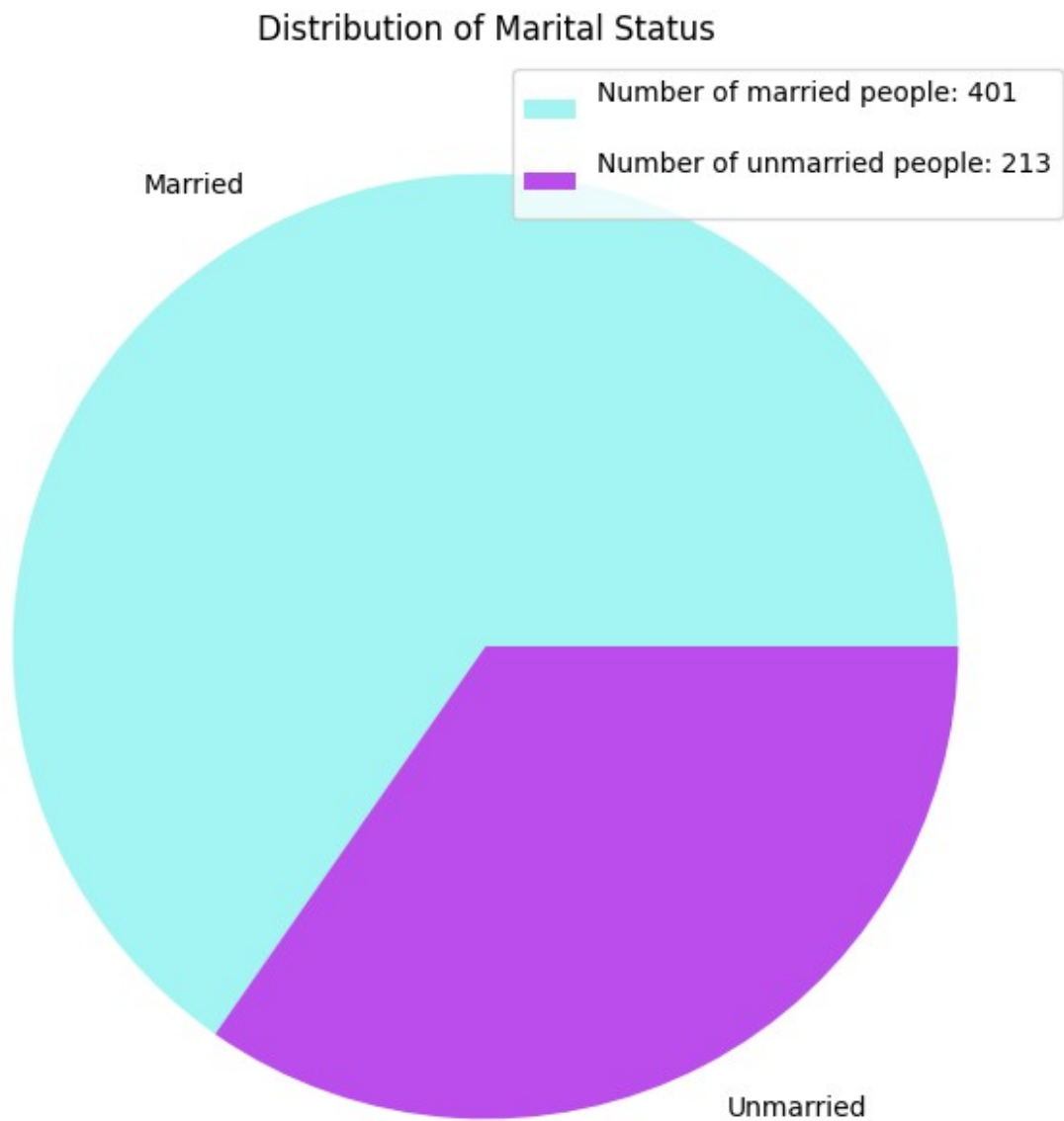
Plot the data using a bar graph.

```
plt.figure(figsize=(8, 8))
plt.pie(marital_totals, labels=['Married', 'Unmarried'],
        colors=['#A3F5F3', '#BB4DED'])
plt.title('Distribution of Marital Status')
```

```

legend_text = [
    f"Number of married people: {maritua_total[0]}\n",
    f"Number of unmarried people: {maritua_total[1]}\n",
]
plt.legend(labels=legend_text, loc='upper right', facecolor=None)
plt.show()

```



As we can see from the chart, the majority of people are married with a total of 401 of 614 people and 213 are unmarried.