Convert user_type from integer to category

In [ ]:
```
ride_sharing['user_type_cat'] = ride_sharing['user_type'].astype('category'
```

Strip duration of minutes

In [ ]:
```
ride_sharing['duration_trim'] = ride_sharing['duration'].str.strip('minutes
```

Set all values above 27 to 27

In [ ]:
```
ride_sharing.loc[ride_sharing['tire_sizes'] > 27, 'tire_sizes'] = 27
```

Find duplicates

In [ ]:
```
duplicates = ride_sharing.duplicated(subset='ride_id', keep=False)
```

Drop complete duplicates from ride_sharing

In [ ]:
```
ride_dup = ride_sharing.drop_duplicates()
```

Print number of missing values in banking

In [ ]:
```
print(banking.isna().sum())
```

Visualize missingness matrix

In [ ]:
```
msno.matrix(banking)
plt.show()
```

In [6]:
```
from fuzzywuzzy import process
wrong_spells=['tahaa','teha','tahaa']
matches = process.extract('taha', wrong_spells, limit=len(wrong_spells))
print(matches)
```

```
C:\Users\Rubab\AppData\Local\Programs\Python\Python310\lib\site-packages
\fuzzywuzzy\fuzz.py:11: UserWarning: Using slow pure-python SequenceMatch
er. Install python-Levenshtein to remove this warning
  warnings.warn('Using slow pure-python SequenceMatcher. Install python-L
evenshtein to remove this warning')

[('tahaa', 89), ('tahaa', 89), ('teha', 75)]
```

Drop rows with missing values

In [ ]:
```
df.dropna()
```

Fill missing values with a specific value

In [ ]:
```
df.fillna(value)
```

Interpolate missing values

In [ ]:
```
df.interpolate()
```

Remove duplicates based on all columns

In [ ]:
```
df.drop_duplicates()
```

Remove duplicates based on specific columns

In [ ]:
```
df.drop_duplicates(subset=['col1', 'col2'])
```

Apply a function to a column

In [ ]:
```
df['new_col'] = df['col'].apply(lambda x: x * 2)
```

Create a new column based on conditions

In [ ]:
```
df['category'] = np.where(df['col'] > threshold, 'High', 'Low')
```

Convert categorical column to numerical using label encoding

In [ ]:
```
df['category_encoded'] = df['category'].astype('category').cat.codes
```

Convert categorical column to one-hot encoded columns

In [ ]:
```
df = pd.get_dummies(df, columns=['category'], drop_first=True)
```