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

from google.colab import files
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving EcoActions Dataset.txt to EcoActions Dataset (1).txt
```

```
df = pd.read_csv(list(uploaded.keys())[0])

df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 50,\n  \"fields\": [\n
{\n      \"column\": \"user_id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 1,\n
\"max\": 50,\n        \"num_unique_values\": 50,\n        \"samples\":
[\n          14,\n          40,\n          31\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"action_type\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"Recycle plastic\",\n          \"Recycle paper\",\n          \"Pick
up litter\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"date\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 49,\n        \"samples\": [\n          \"2025-
02-05\",\n          \"2025-01-04\",\n          \"2025-01-02\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"age\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 3,\n        \"min\":
22,\n        \"max\": 35,\n        \"num_unique_values\": 14,\n
\"samples\": [\n          32,\n          23,\n          24\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Male\",\n          \"Female\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"location\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
8,\n        \"samples\": [\n          \"London\",\n
\"Seattle\"\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"frequency\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 3,\n   \"samples\":
[\n   \"Daily\",\n   \"Weekly\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"outcome\",\n   \"properties\":
{\n   \"dtype\": \"category\",\n   \"num_unique_values\":
2,\n   \"samples\": [\n   \"Short-term\",\n
\"Long-term\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"category\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   \"Energy-saving\",\n   \"Transportation\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"impact_score\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1,\n   \"min\": 6,\n   \"max\": 9,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   7,\n
6\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df"}

```
df.head(10)
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 50,\n \"fields\": [\n
{\n   \"column\": \"user_id\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 14,\n   \"min\": 1,\n
\"max\": 50,\n   \"num_unique_values\": 50,\n   \"samples\":
[\n   14,\n   40,\n   31\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"action_type\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 10,\n   \"samples\": [\n
\"Recycle plastic\",\n   \"Recycle paper\",\n   \"Pick
up litter\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"date\",\n   \"properties\": {\n   \"dtype\": \"object\",\n
\"num_unique_values\": 49,\n   \"samples\": [\n   \"2025-
02-05\",\n   \"2025-01-04\",\n   \"2025-01-02\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"age\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 3,\n   \"min\":
22,\n   \"max\": 35,\n   \"num_unique_values\": 14,\n
\"samples\": [\n   32,\n   23,\n   24\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"gender\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"Male\",\n   \"Female\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\

```
n    },\n    {\n      \"column\": \"location\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
8,\n        \"samples\": [\n          \"London\",\n
\"Seattle\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"frequency\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"Daily\",\n          \"Weekly\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"outcome\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Short-term\",\n
\"Long-term\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"Energy-saving\",\n          \"Transportation\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"impact_score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 6,\n        \"max\": 9,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          7,\n
6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
df.tail()
```


```
{"repr_error":"0","type":"dataframe"}
```


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   user_id       50 non-null     int64
 1   action_type   50 non-null     object
 2   date          50 non-null     object
 3   age           50 non-null     int64
 4   gender        50 non-null     object
 5   location      50 non-null     object
 6   frequency     50 non-null     object
 7   outcome       50 non-null     object
 8   category      50 non-null     object
 9   impact_score  50 non-null     int64
dtypes: int64(3), object(7)
memory usage: 4.0+ KB
```

```
df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"user_id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 17.716559962530223,\n
\"min\": 1.0,\n        \"max\": 50.0,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          50.0,\n          25.5,\n
37.75\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 13.04281304730728,\n        \"min\": 3.352976137641279,\n
\"max\": 50.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          28.68,\n          29.0,\n          50.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"impact_score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
15.567649939757962,\n        \"min\": 1.013379877198183,\n
\"max\": 50.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          50.0,\n          7.44,\n          8.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
<google.colab._quickchart_helpers.SectionTitle at 0x7f01b3aec2d0>
```

```
from matplotlib import pyplot as plt
_df_8['user_id'].plot(kind='hist', bins=20, title='user_id')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_9['age'].plot(kind='hist', bins=20, title='age')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_10['impact_score'].plot(kind='hist', bins=20,
title='impact_score')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f01b3ab0990>
```

```
from matplotlib import pyplot as plt
_df_11.plot(kind='scatter', x='user_id', y='age', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_12.plot(kind='scatter', x='age', y='impact_score', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
<google.colab._quickchart_helpers.SectionTitle at 0x7f01b3a7c1d0>
```

```
from matplotlib import pyplot as plt
_df_13['user_id'].plot(kind='line', figsize=(8, 4), title='user_id')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_14['age'].plot(kind='line', figsize=(8, 4), title='age')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
from matplotlib import pyplot as plt
_df_15['impact_score'].plot(kind='line', figsize=(8, 4),
title='impact_score')
plt.gca().spines[['top', 'right']].set_visible(False)
```

```
df.shape
```

```
(50, 10)
```

```
df.dtypes
```

```
user_id          int64
action_type     object
date            object
age              int64
gender          object
location        object
frequency       object
outcome         object
category        object
impact_score     int64
dtype: object
```

```
df.columns
```

```
Index(['user_id', 'action_type', 'date', 'age', 'gender', 'location',
       'frequency', 'outcome', 'category', 'impact_score'],
      dtype='object')
```

```
df.isnull().sum()
```

```
user_id         0
action_type     0
date            0
age             0
gender          0
location        0
frequency       0
outcome         0
category        0
impact_score    0
dtype: int64
```

```
df.fillna(df.mode().iloc[0], inplace=True)  # For categorical columns
```

```
df.dropna(inplace=True)  # Removes rows with any missing values

df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 50,\n  \"fields\": [\n
{\n      \"column\": \"user_id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 1,\n
\"max\": 50,\n        \"num_unique_values\": 50,\n        \"samples\":
[\n          14,\n          40,\n          31\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"action_type\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"Recycle plastic\",\n          \"Recycle paper\",\n          \"Pick
up litter\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"date\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 49,\n        \"samples\": [\n          \"2025-
02-05\",\n          \"2025-01-04\",\n          \"2025-01-02\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"age\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 3,\n        \"min\":
22,\n        \"max\": 35,\n        \"num_unique_values\": 14,\n
\"samples\": [\n          32,\n          23,\n          24\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Male\",\n          \"Female\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"location\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
8,\n        \"samples\": [\n          \"London\",\n
\"Seattle\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"frequency\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"Daily\",\n          \"Weekly\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"outcome\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Short-term\",\n
\"Long-term\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"Energy-saving\",\n          \"Transportation\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"impact_score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
1,\n        \"min\": 6,\n        \"max\": 9,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          7,\n
6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```


```
df['category'].value_counts()  # Counts how many times each category
appears
```

```
category
Recycling          17
Waste Reduction    15
Energy-saving       8
Diet                6
Transportation      4
Name: count, dtype: int64
```

```
df.groupby('category')['impact_score'].mean()  # Get the average
impact_score per category
df.groupby('category')['impact_score'].sum()  # Get the total
impact_score per category
```

```
category
Diet                42
Energy-saving       66
Recycling          131
Transportation      30
Waste Reduction    103
Name: impact_score, dtype: int64
```

```
df.groupby('category').size()  # Counts the number of rows per
category
```

```
category
Diet                6
Energy-saving       8
Recycling          17
Transportation      4
Waste Reduction    15
dtype: int64
```

```
df.rename(columns={'impact_score': 'Eco Impact'}, inplace=True)
```

```
df['date'] = pd.to_datetime(df['date'])
```

```
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 50,\n  \"fields\": [\n
{\n      \"column\": \"user_id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 14,\n        \"min\": 1,\n
\"max\": 50,\n        \"num_unique_values\": 50,\n        \"samples\":
```

```
[\n          14,\n          40,\n          31\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"action_type\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n
\"Recycle plastic\",\n          \"Recycle paper\",\n          \"Pick
up litter\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"date\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": \"2025-01-01 00:00:00\",\n        \"max\": \"2025-02-18
00:00:00\",\n        \"num_unique_values\": 49,\n        \"samples\":
[\n          \"2025-02-05 00:00:00\",\n          \"2025-01-04
00:00:00\",\n          \"2025-01-02 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 3,\n        \"min\": 22,\n
\"max\": 35,\n        \"num_unique_values\": 14,\n        \"samples\":
[\n          32,\n          23,\n          24\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"gender\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Male\",\n          \"Female\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"location\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 8,\n        \"samples\":
[\n          \"London\",\n          \"Seattle\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"frequency\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"Daily\",\n          \"Weekly\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"outcome\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Short-term\",\n
\"Long-term\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"Energy-saving\",\n          \"Transportation\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Eco Impact\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 6,\n        \"max\": 9,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          7,\n
6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
df['category'].nunique()
```

```
5
```

```
df['category'].value_counts().idxmax()
```

```
{"type":"string"}
```

```
df.groupby('category')['Eco Impact'].mean()
```

```
category
Diet                7.000000
Energy-saving       8.250000
Recycling           7.705882
Transportation      7.500000
Waste Reduction     6.866667
Name: Eco Impact, dtype: float64
```

```
df.nlargest(5, 'Eco Impact')
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"user_id\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 3,\n
\"max\": 30,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          10,\n          30,\n          11\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"action_type\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n          \"Plant a
tree\",\n          \"Save water\",\n          \"Compost kitchen
waste\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"date\",\n      \"properties\": {\n        \"dtype\": \"date\",\n
\"min\": \"2025-01-21 00:00:00\",\n        \"max\": \"2025-02-17
00:00:00\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"2025-02-10 00:00:00\",\n          \"2025-01-21
00:00:00\",\n          \"2025-02-09 00:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 27,\n
\"max\": 32,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          32,\n          30,\n          27\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"gender\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Male\",\n          \"Female\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"location\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"San Francisco\",\n          \"New York\"\n        ],\n
```

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"frequency\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n   \"samples\": [\n
\"Weekly\",\n   \"One-time\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"outcome\",\n   \"properties\":
{\n   \"dtype\": \"category\",\n   \"num_unique_values\":
1,\n   \"samples\": [\n   \"Long-term\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"category\",\n   \"properties\":
{\n   \"dtype\": \"string\",\n   \"num_unique_values\": 3,\n
\"samples\": [\n   \"Recycling\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Eco Impact\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0,\n   \"min\": 9,\n   \"max\": 9,\n
\"num_unique_values\": 1,\n   \"samples\": [\n   9\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}

```
df['year'] = df['date'].dt.year
df['year'].value_counts()
```

```
year
2025    50
Name: count, dtype: int64
```

```
(df['category'].value_counts(normalize=True) * 100).round(2)
```

```
category
Recycling           34.0
Waste Reduction     30.0
Energy-saving       16.0
Diet                12.0
Transportation       8.0
Name: proportion, dtype: float64
```

```
brown_beige_palette = ["#6B4226", "#A47551", "#8F9779", "#556B2F",
"#A3B18A", "#6A994E"]
```

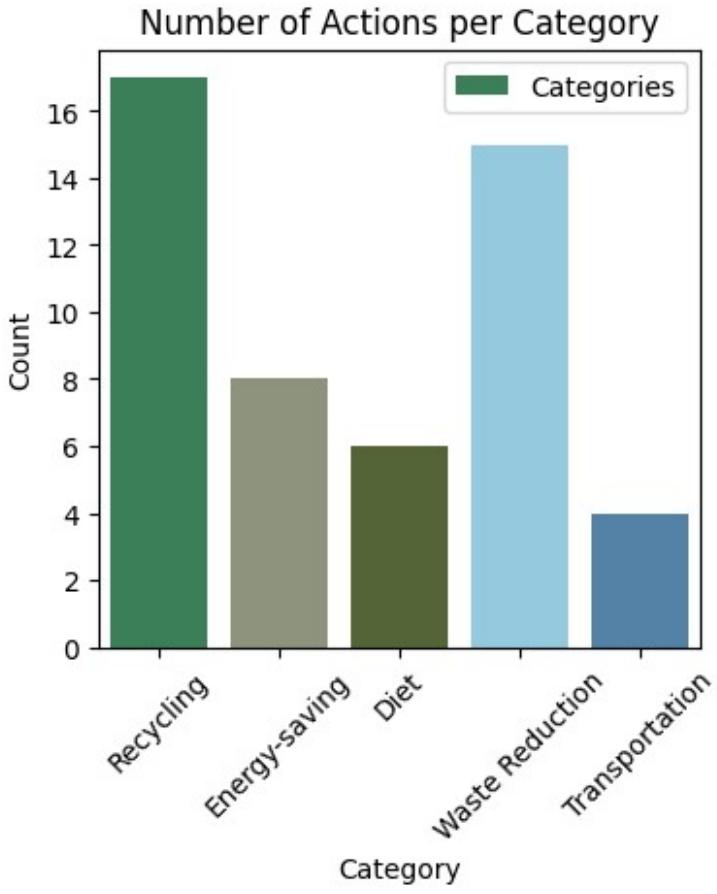
```
df.columns
```

```
Index(['user_id', 'action_type', 'date', 'age', 'gender', 'location',
       'frequency', 'outcome', 'category', 'Eco Impact', 'year'],
      dtype='object')
```

```
custom_palette = ["#2E8B57", "#8F9779", "#556B2F", "#87CEEB",
"#4682B4", "#FFD700", "#8B4513", "#C2B280"]
```

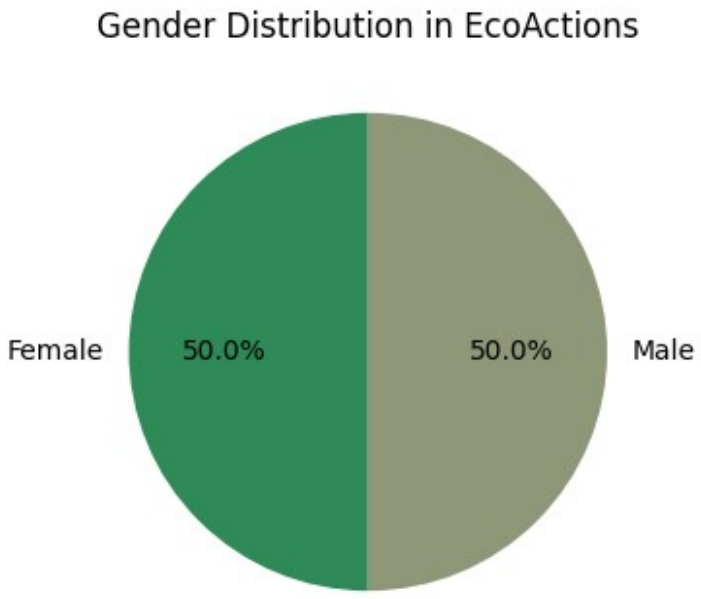
```
plt.figure(figsize=(4, 4))
sns.countplot(data=df, x='category', palette=custom_palette)
plt.xticks(rotation=45)
plt.title("Number of Actions per Category")
plt.xlabel("Category")
plt.ylabel("Count")
plt.legend(["Categories"], loc="upper right")
plt.show()
```

```
<ipython-input-108-644c45555482>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data=df, x='category', palette=custom_palette)
<ipython-input-108-644c45555482>:2: UserWarning: The palette list has
more values (8) than needed (5), which may not be intended.
  sns.countplot(data=df, x='category', palette=custom_palette)
```



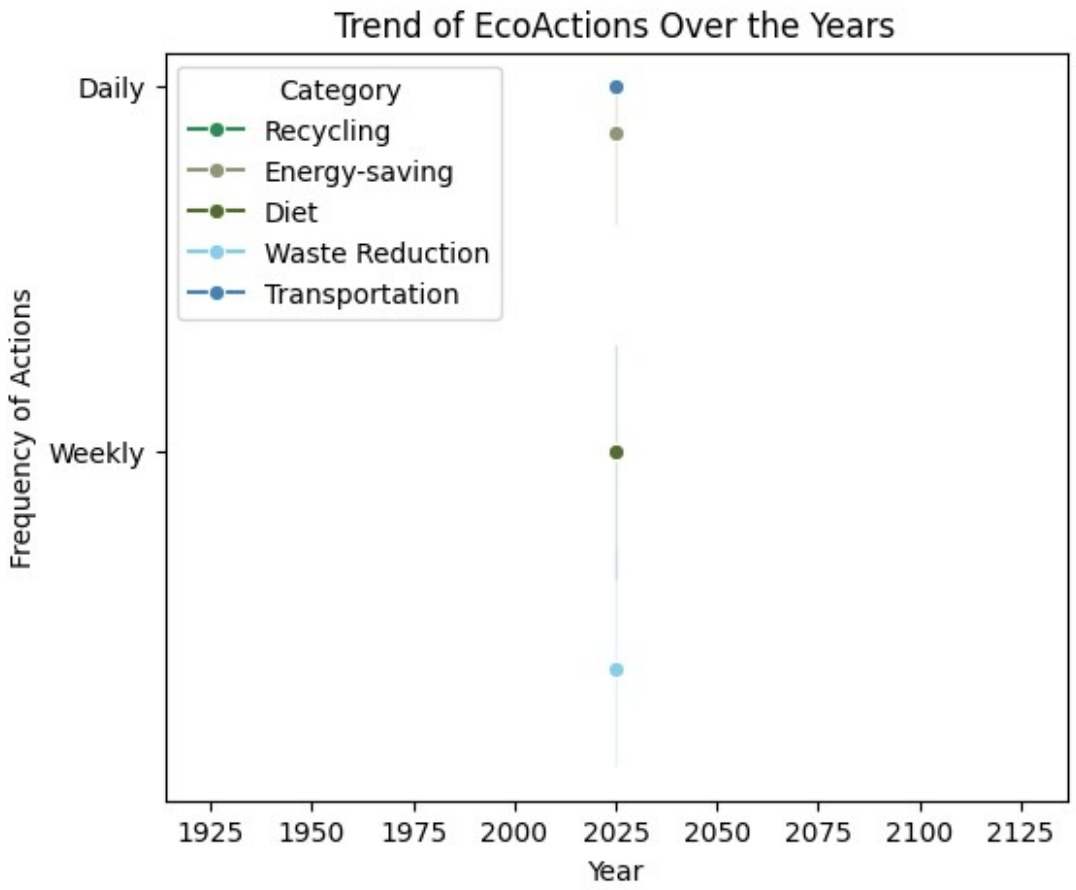

```
plt.figure(figsize=(4, 4))
df['gender'].value_counts().plot.pie(autopct='%1.1f%%',
```

```
colors=custom_palette, startangle=90)
plt.title("Gender Distribution in EcoActions")
plt.ylabel("")
plt.show()
```

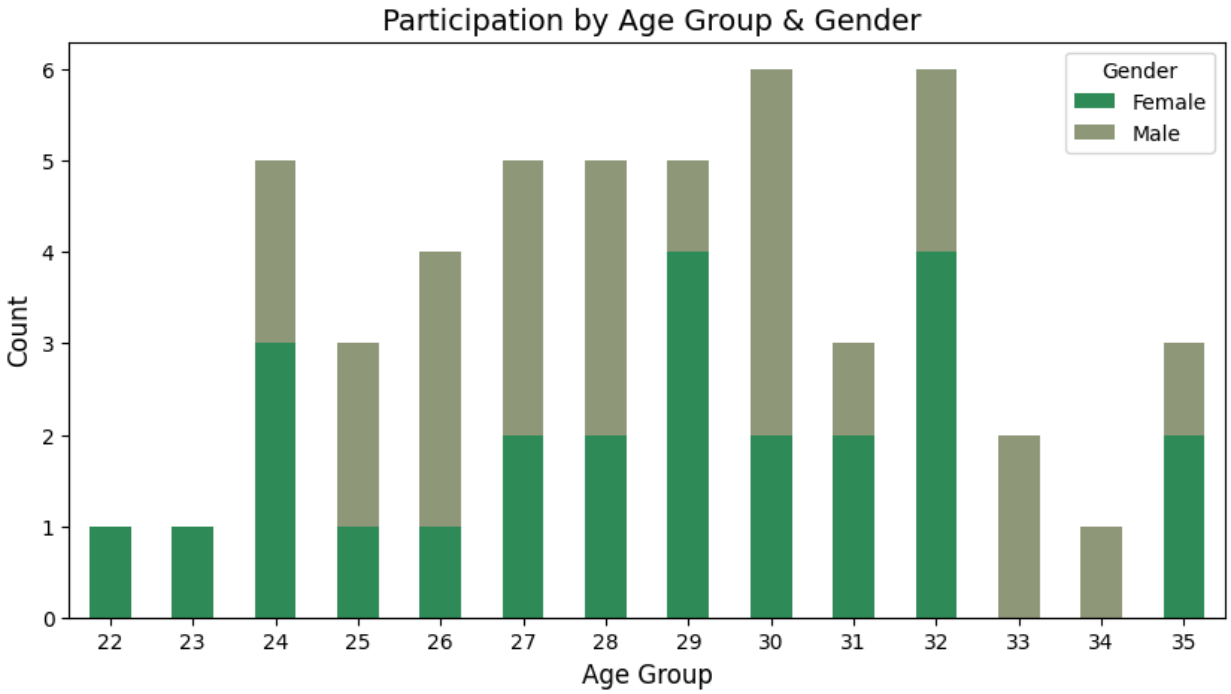



```
plt.figure(figsize=(6, 5))
sns.lineplot(data=df, x="year", y="frequency", hue="category",
marker="o", palette=custom_palette)
plt.title("Trend of EcoActions Over the Years")
plt.xlabel("Year")
plt.ylabel("Frequency of Actions")
plt.legend(title="Category", loc="upper left")
plt.show()
```

```
<ipython-input-110-aa3f43ca0941>:2: UserWarning: The palette list has
more values (8) than needed (5), which may not be intended.
  sns.lineplot(data=df, x="year", y="frequency", hue="category",
marker="o", palette=custom_palette)
```

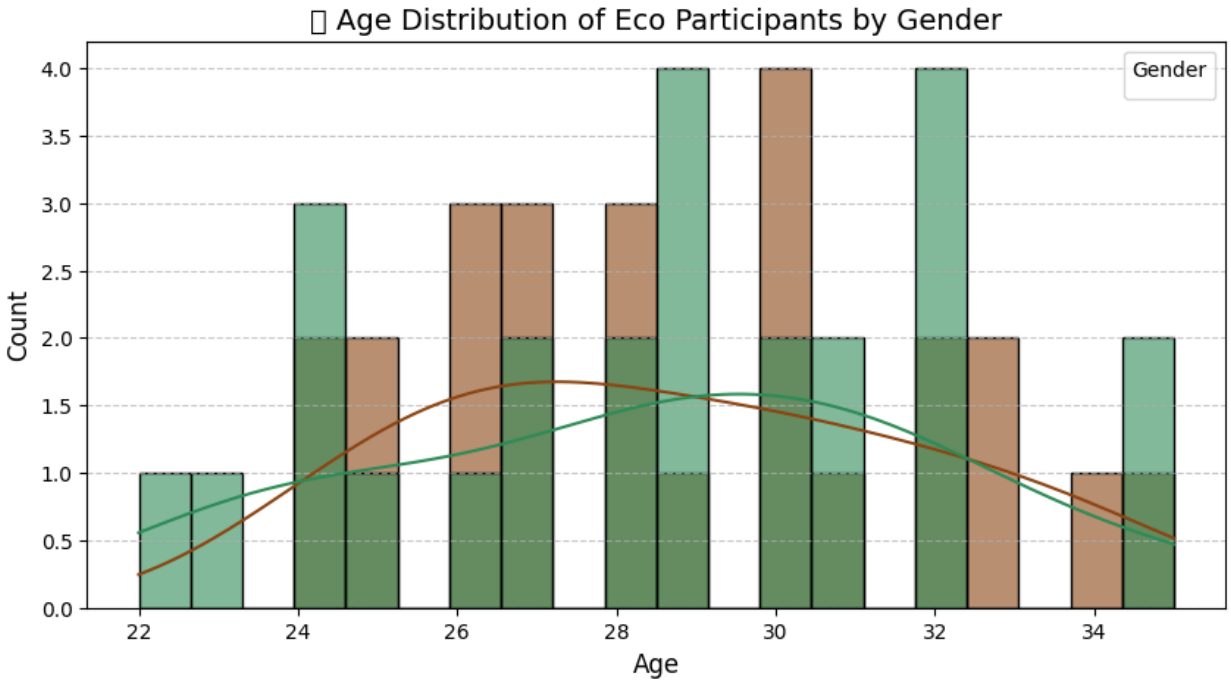



```
plt.figure(figsize=(10, 5))
age_gender_counts = df.groupby(["age", "gender"]).size().unstack()
age_gender_counts.plot(kind="bar", stacked=True, color=custom_palette,
figsize=(10, 5))
plt.title(" Participation by Age Group & Gender", fontsize=14)
plt.xlabel("Age Group", fontsize=12)
plt.ylabel("Count", fontsize=12)
plt.legend(title="Gender", loc="upper right")
plt.xticks(rotation=0)
plt.show()

<Figure size 1000x500 with 0 Axes>
```



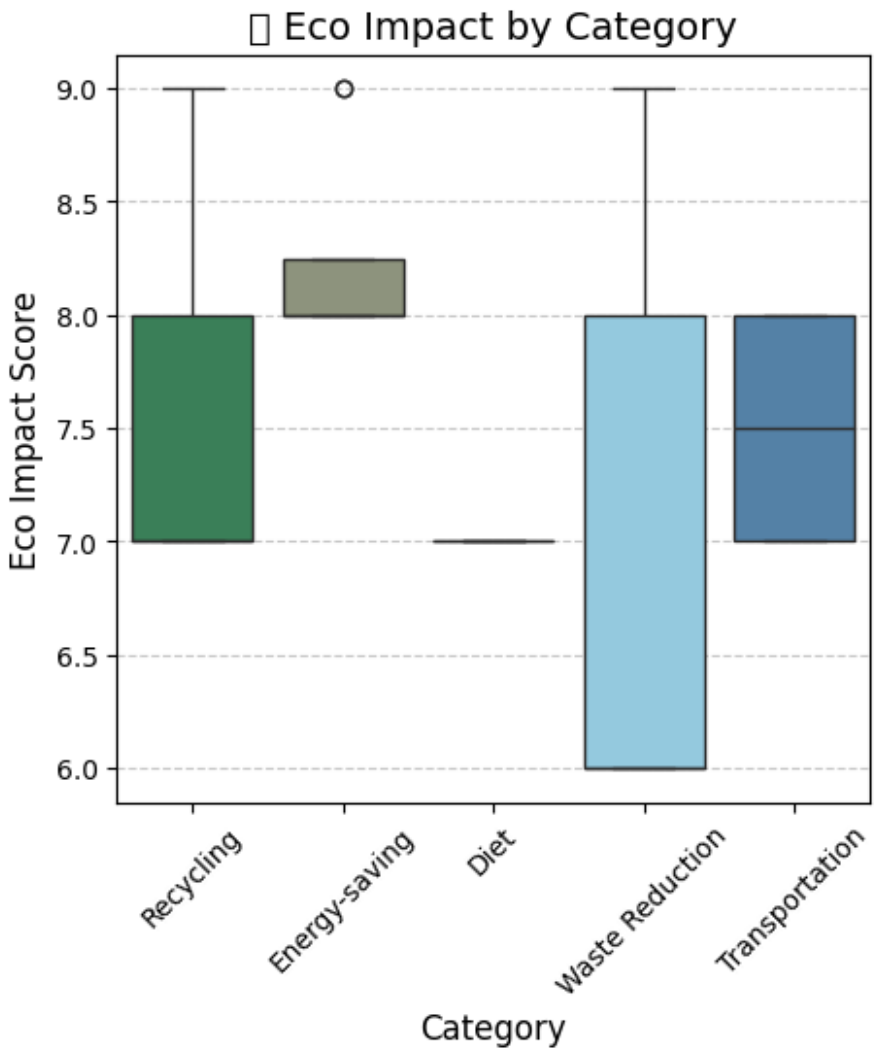

```
plt.figure(figsize=(10, 5))
sns.histplot(df, x="age", bins=20, kde=True, hue="gender",
palette=["#2E8B57", "#8B4513"], alpha=0.6)
plt.title("□ Age Distribution of Eco Participants by Gender",
fontsize=14)
plt.xlabel("Age", fontsize=12)
plt.ylabel("Count", fontsize=12)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.legend(title="Gender")
plt.show()
```

```
<ipython-input-116-0522ba3778c8>:7: UserWarning: No artists with
labels found to put in legend.  Note that artists whose label start
with an underscore are ignored when legend() is called with no
argument.
  plt.legend(title="Gender")
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151
: UserWarning: Glyph 128994 (\N{LARGE GREEN CIRCLE}) missing from
font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```




```
plt.figure(figsize=(5, 5))
sns.boxplot(data=df, x="category", y="Eco Impact",
palette=custom_palette)
plt.title(" Eco Impact by Category", fontsize=14)
plt.xlabel("Category", fontsize=12)
plt.ylabel("Eco Impact Score", fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```

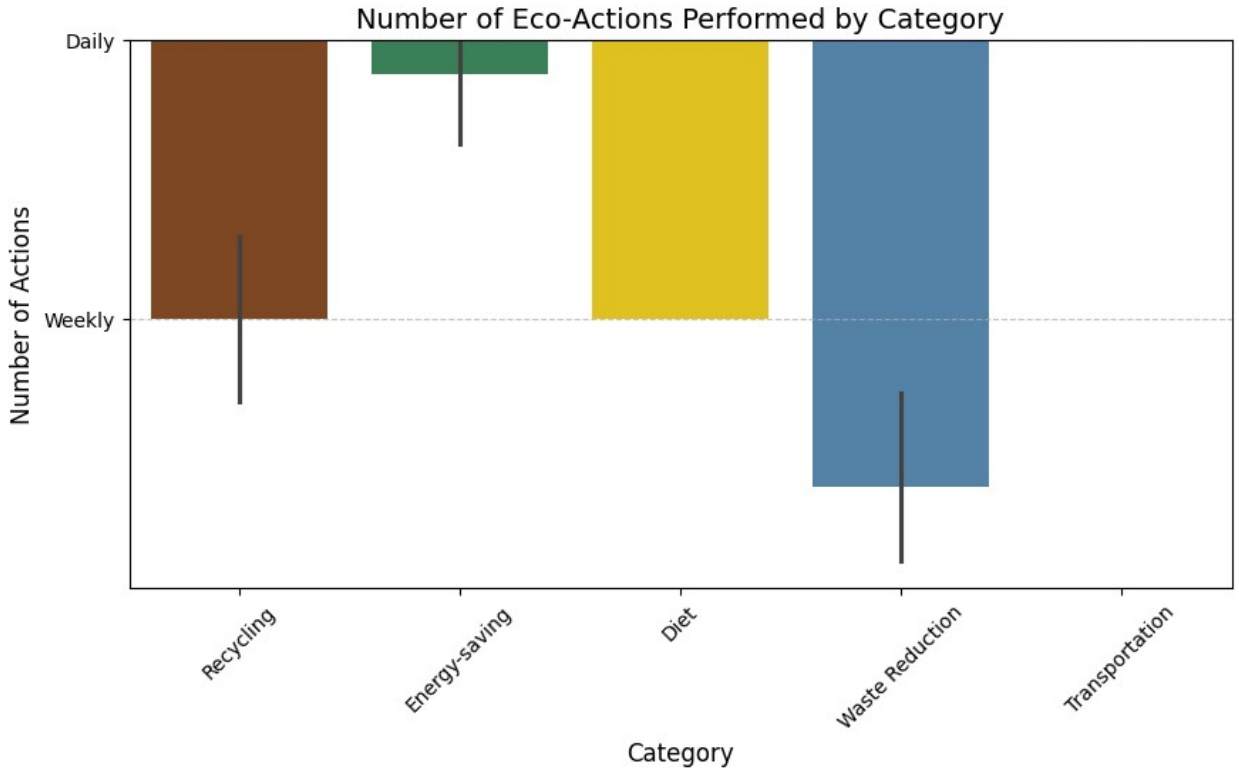
```
<ipython-input-118-0cc67a015bdf>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df, x="category", y="Eco Impact",
palette=custom_palette)
<ipython-input-118-0cc67a015bdf>:2: UserWarning: The palette list has
more values (8) than needed (5), which may not be intended.
  sns.boxplot(data=df, x="category", y="Eco Impact",
palette=custom_palette)
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151
: UserWarning: Glyph 127793 (\N{SEEDLING}) missing from font(s) DejaVu
Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```




```
plt.figure(figsize=(10, 5))
sns.barplot(data=df, x="category", y="frequency", palette=["#8B4513",
"#2E8B57", "#FFD700", "#4682B4"])
plt.title("Number of Eco-Actions Performed by Category", fontsize=14)
plt.xlabel("Category", fontsize=12)
plt.ylabel("Number of Actions", fontsize=12)
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```

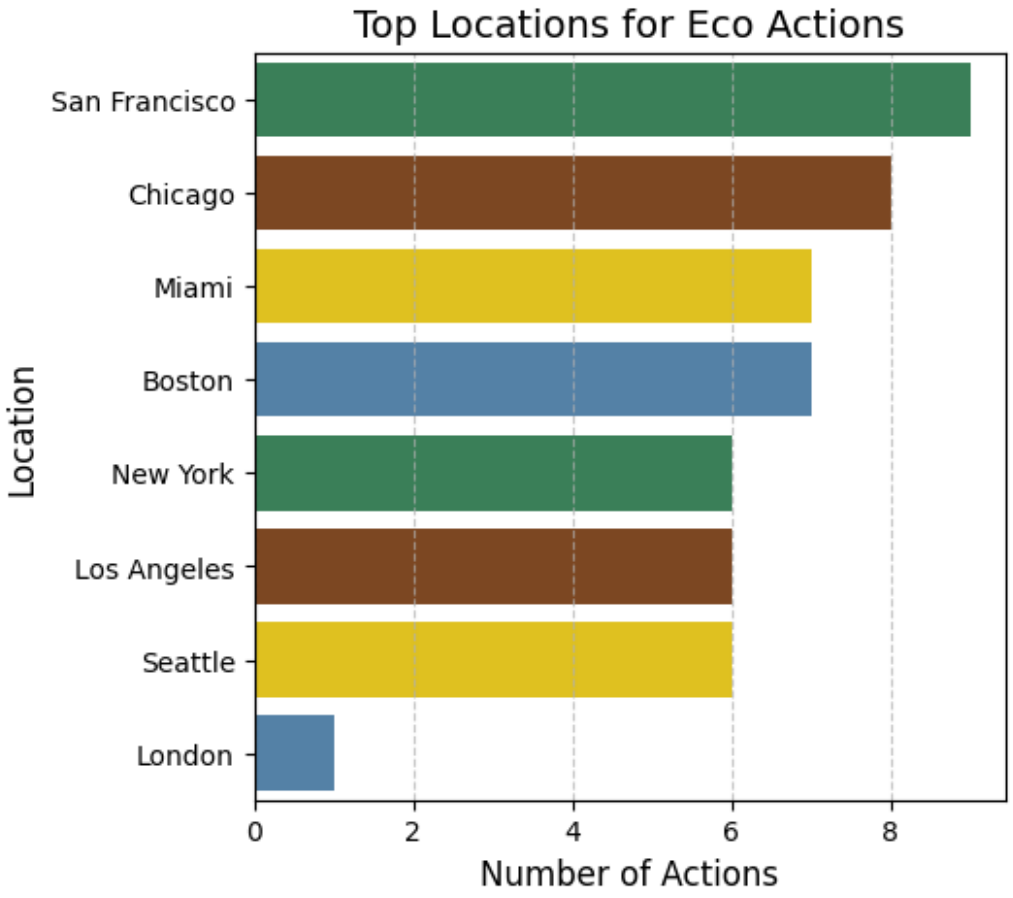
```
<ipython-input-121-906996a90a5e>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=df, x="category", y="frequency",
palette=["#8B4513", "#2E8B57", "#FFD700", "#4682B4"])
```

```
<ipython-input-121-906996a90a5e>:2: UserWarning:
The palette list has fewer values (4) than needed (5) and will cycle,
which may produce an uninterpretable plot.
  sns.barplot(data=df, x="category", y="frequency",
palette=["#8B4513", "#2E8B57", "#FFD700", "#4682B4"])
```




```
plt.figure(figsize=(5, 5))
sns.countplot(data=df, y="location",
order=df["location"].value_counts().index, palette=["#2E8B57",
"#8B4513", "#FFD700", "#4682B4"])
plt.title("Top Locations for Eco Actions", fontsize=14)
plt.xlabel("Number of Actions", fontsize=12)
plt.ylabel("Location", fontsize=12)
plt.grid(axis="x", linestyle="--", alpha=0.7)
plt.show()

<ipython-input-128-f06fcbc99c85>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(data=df, y="location",
order=df["location"].value_counts().index, palette=["#2E8B57",
"#8B4513", "#FFD700", "#4682B4"])
```

```
<ipython-input-128-f06fcbc99c85>:2: UserWarning:
The palette list has fewer values (4) than needed (8) and will cycle,
which may produce an uninterpretable plot.
  sns.countplot(data=df, y="location",
order=df["location"].value_counts().index, palette=["#2E8B57",
"#8B4513", "#FFD700", "#4682B4"])
```



##This project analyzed eco-actions data to understand patterns in participation, success rates, and environmental impac