CSE 578: Data Visualization Project

Sahil Yogesh Hadke | shadke1@asu.edu | 1229679960

Import Statements

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

Read input

```
# Define the headers based on your provided structure
headers = [
    'age', 'workclass', 'fnlwgt', 'education', 'education-num',
'marital-status',
    'occupation', 'relationship', 'race', 'sex', 'capital-gain',
'capital-loss',
    'hours-per-week', 'native-country', 'income'
]

df = pd.read_csv('./data/adult.data', header=None, names=headers)

df.head()
```

```
   age          workclass  fnlwgt   education  education-num  \
0   39          State-gov   77516   Bachelors             13
1   50   Self-emp-not-inc   83311   Bachelors             13
2   38            Private  215646     HS-grad              9
3   53            Private  234721        11th              7
4   28            Private  338409   Bachelors             13

        marital-status          occupation    relationship    race
sex  \
0        Never-married        Adm-clerical   Not-in-family   White
Male
1   Married-civ-spouse     Exec-managerial         Husband   White
Male
2             Divorced   Handlers-cleaners   Not-in-family   White
Male
3   Married-civ-spouse   Handlers-cleaners         Husband   Black
Male
4   Married-civ-spouse      Prof-specialty            Wife   Black
Female

   capital-gain  capital-loss  hours-per-week  native-country  income
```

```
0          2174             0              40   United-States   <=50K

1             0             0              13   United-States   <=50K

2             0             0              40   United-States   <=50K

3             0             0              40   United-States   <=50K

4             0             0              40            Cuba   <=50K


df['age'].count()

32561
```

Data Preprocessing

```
# drop columns which are not relevant
df = df.drop(['fnlwgt', 'education-num'], axis=1)

# Remove ? (NaN)
df.replace(' ?', np.nan, inplace=True)
df = df.dropna()

df['age'].count()

30162
```

Export to Excel for better understanding and exploring

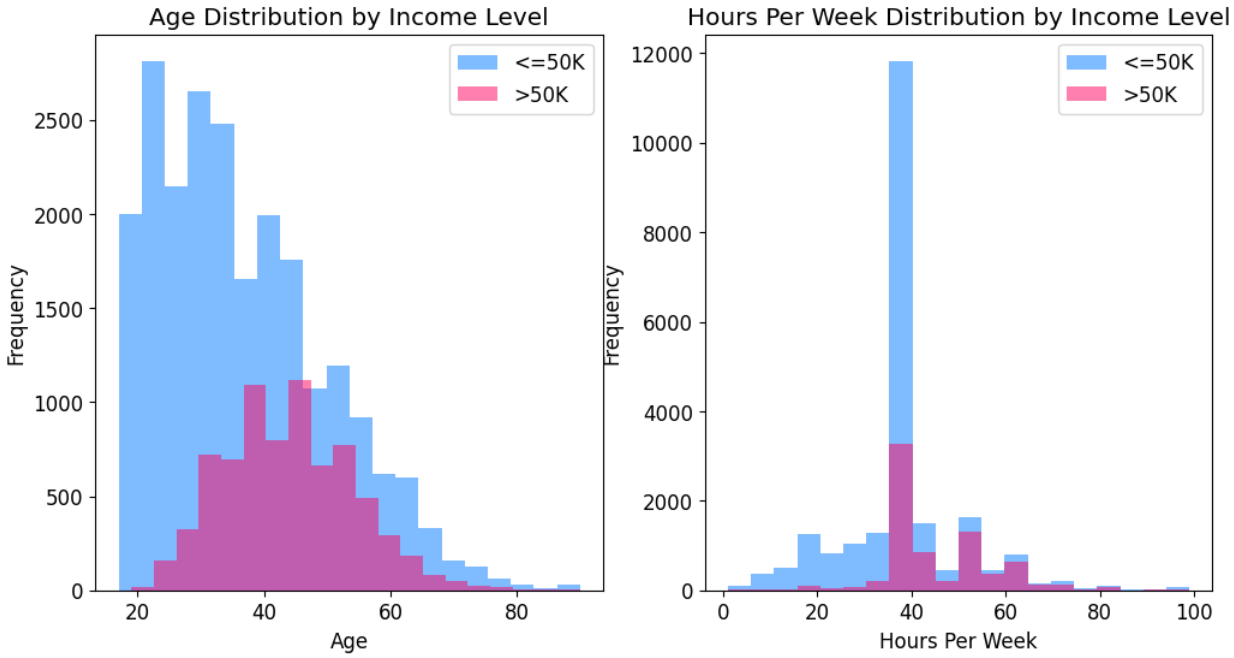
```
df.to_csv("./data/data.csv")
```

User Story 1 (Multivariate)

Age, Hours Per Week Vs. Income

```
# Age Income Analysis
income_low = df[df['income'] == ' <=50K']
income_high = df[df['income'] == ' >50K']

# Create a figure and axis
fig, ax = plt.subplots(1,2, figsize=(12, 6))

# Plot the histograms for both income levels
ax[0].hist(income_low['age'], bins=20, alpha=0.5, label='<=50K',
color='#0079FF')
ax[0].hist(income_high['age'], bins=20, alpha=0.5, label='>50K',
color='#FF0060')

# Add labels and title
ax[0].set_xlabel('Age')
```

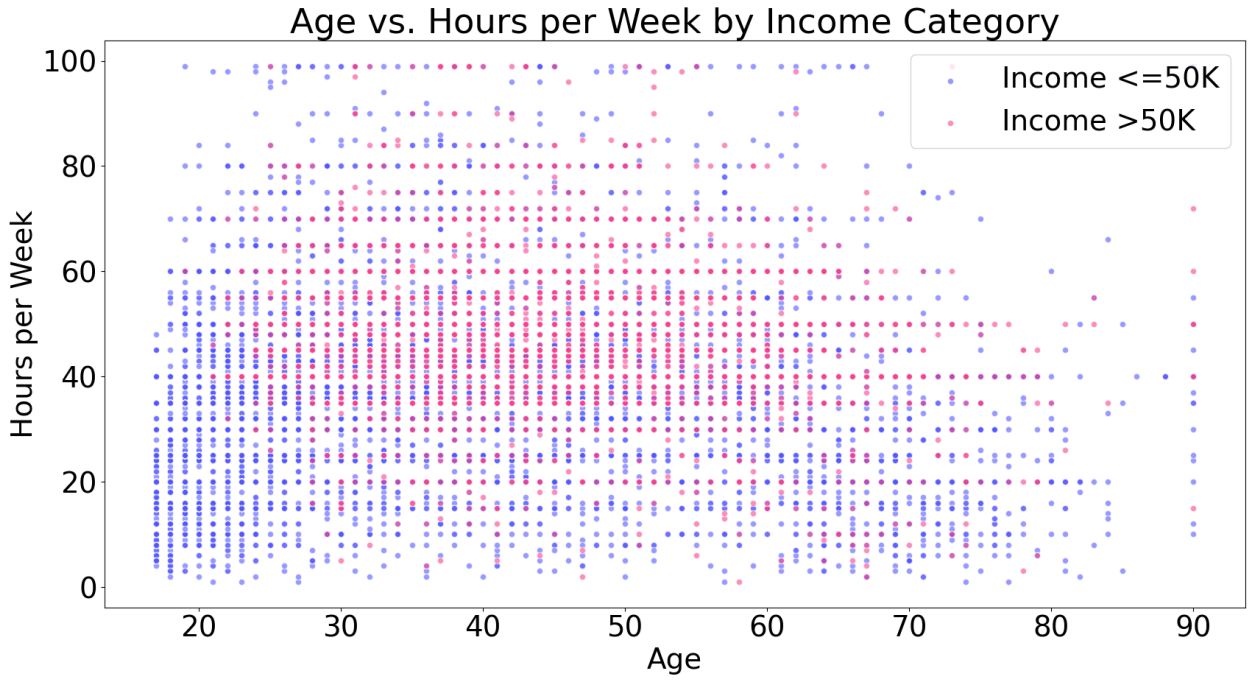
```
ax[0].set_ylabel('Frequency')
ax[0].set_title('Age Distribution by Income Level')
ax[0].legend()


# Plot the histograms for both income levels
ax[1].hist(income_low['hours-per-week'], bins=20, alpha=0.5,
label='<=50K', color='#0079FF')
ax[1].hist(income_high['hours-per-week'], bins=20, alpha=0.5,
label='>50K', color='#FF0060')

# Add labels and title
ax[1].set_xlabel('Hours Per Week')
ax[1].set_ylabel('Frequency')
ax[1].set_title('Hours Per Week Distribution by Income Level')
ax[1].legend()


plt.savefig('age-hours-per-week-histogram.png')
# Show the plot
plt.show()
```



Scatterplot

```
# Setting up the figure and axis
fig, ax = plt.subplots(figsize=(20, 10))

plt.rcParams['font.size'] = '28'
# Scatter plot for income <=50K
subset2 = df[df['income'] == ' <=50K']
```

```
sc2 = ax.scatter(subset2['age'], subset2['hours-per-week'],
label='Income <=50K', alpha=0.6, edgecolor='w', color='#5356FF')

# Scatter plot for income >50K
subset1 = df[df['income'] == ' >50K']
sc1 = ax.scatter(subset1['age'], subset1['hours-per-week'],
label='Income >50K', alpha=0.6, edgecolor='w', color='#F7418F')

ax.set_title('Age vs. Hours per Week by Income Category')
ax.set_xlabel('Age')
ax.set_ylabel('Hours per Week')
ax.legend()

# font = {'size'   : 24}

# plt.rc('font', **font)
plt.rcParams['font.weight'] = 'normal'


plt.savefig('age-hours-per-week-scatterplot.png')
# plt.show()
```

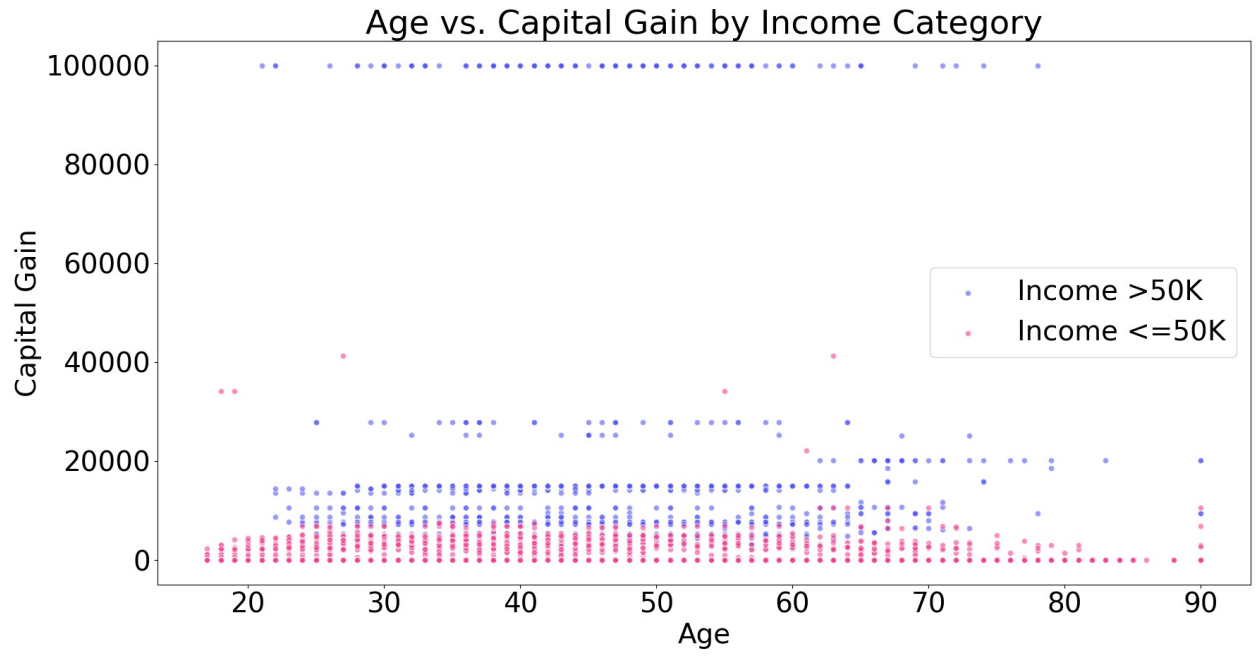


User Story 2 (Multivariate)

Age, Capital Gain Vs. Income

```
# Setting up the figure and axis
fig, ax = plt.subplots(figsize=(20, 10))
```

```
# Scatter plot for income >50K
subset1 = df[df['income'] == ' >50K']
sc1 = ax.scatter(subset1['age'], subset1['capital-gain'],
label='Income >50K', alpha=0.6, edgecolor='w', color='#5356FF')

# Scatter plot for income <=50K
subset2 = df[df['income'] == ' <=50K']
sc2 = ax.scatter(subset2['age'], subset2['capital-gain'],
label='Income <=50K', alpha=0.6, edgecolor='w', color='#F7418F')

ax.set_title('Age vs. Capital Gain by Income Category')
ax.set_xlabel('Age')
ax.set_ylabel('Capital Gain')
ax.legend()

plt.savefig('age-capital-gain-scatterplot.png')
plt.show()
```

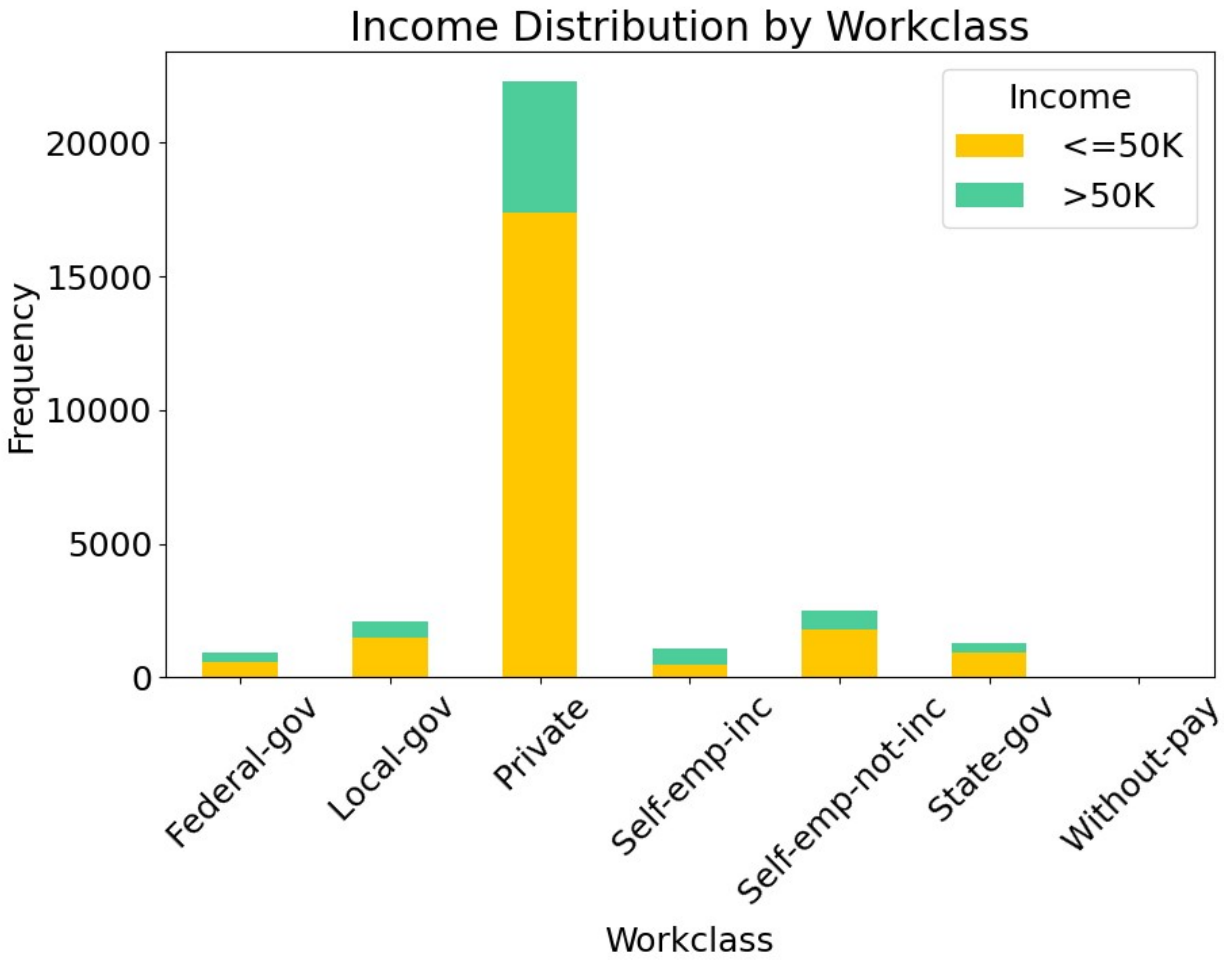


User Story 3(Univariate)

Workclass Vs. Income

```
# Create a cross-tabulation of workclass and income
workclass_income_ct = pd.crosstab(df['workclass'], df['income'])

# Plotting

colors = ['#FFC700', '#4CCD99']
workclass_income_ct.plot(kind='bar', figsize=(10, 6), stacked=True,
color=colors)
```

```
plt.title('Income Distribution by Workclass')
plt.xlabel('Workclass')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Income')

plt.rcParams['font.size'] = '12'

plt.savefig('workclass-income.png')
plt.show()
```

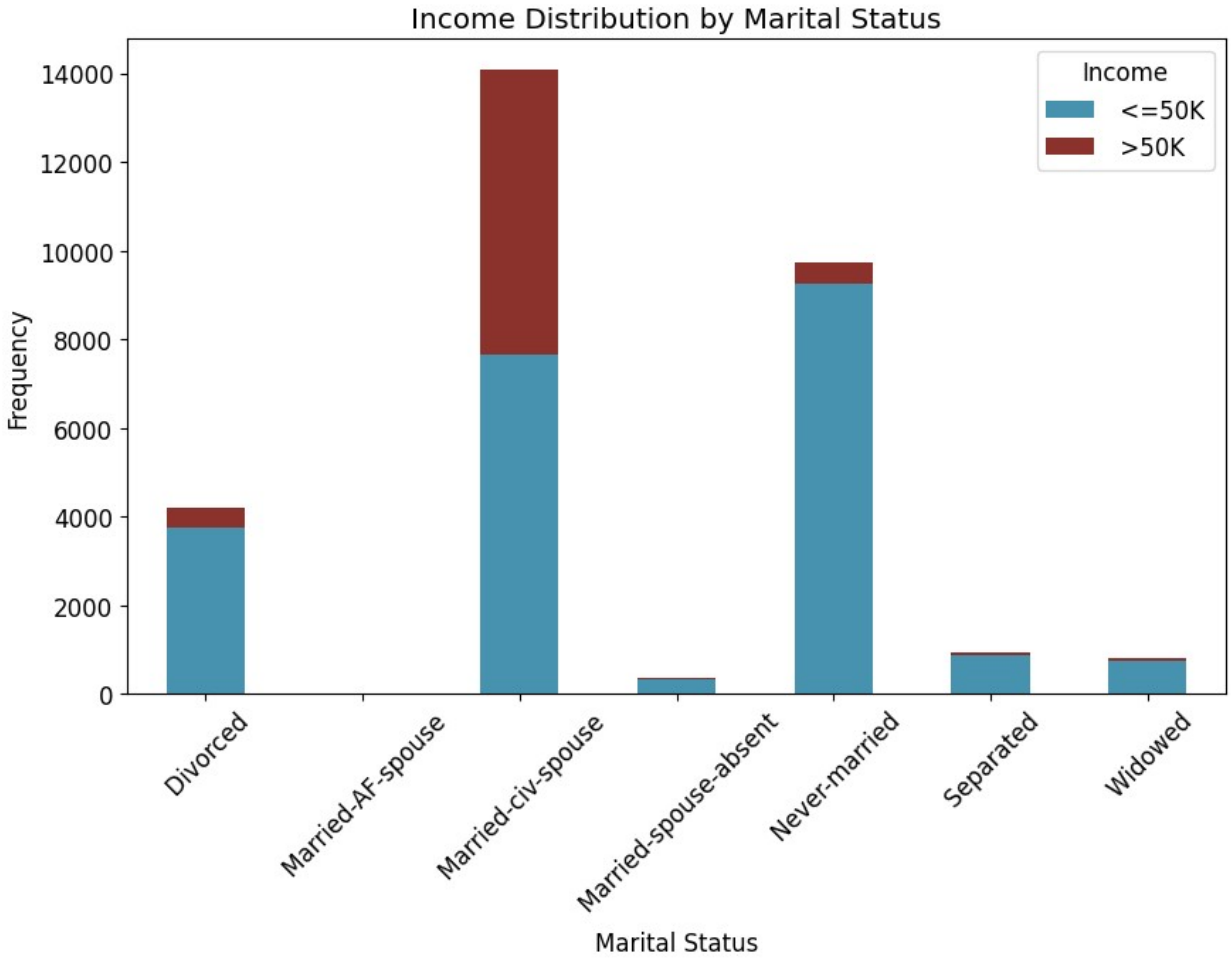


User Story 4 (Univariate)

Marital Status and Income

```
# Create a cross-tabulation of workclass and income
workclass_income_ct = pd.crosstab(df['marital-status'], df['income'])

# Plotting
```

```
colors = ['#4793AF', '#8B322C']
workclass_income_ct.plot(kind='bar', figsize=(10, 6), stacked=True,
color=colors)
plt.title('Income Distribution by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Income')

plt.savefig('marital-status-income.png')
plt.show()
```

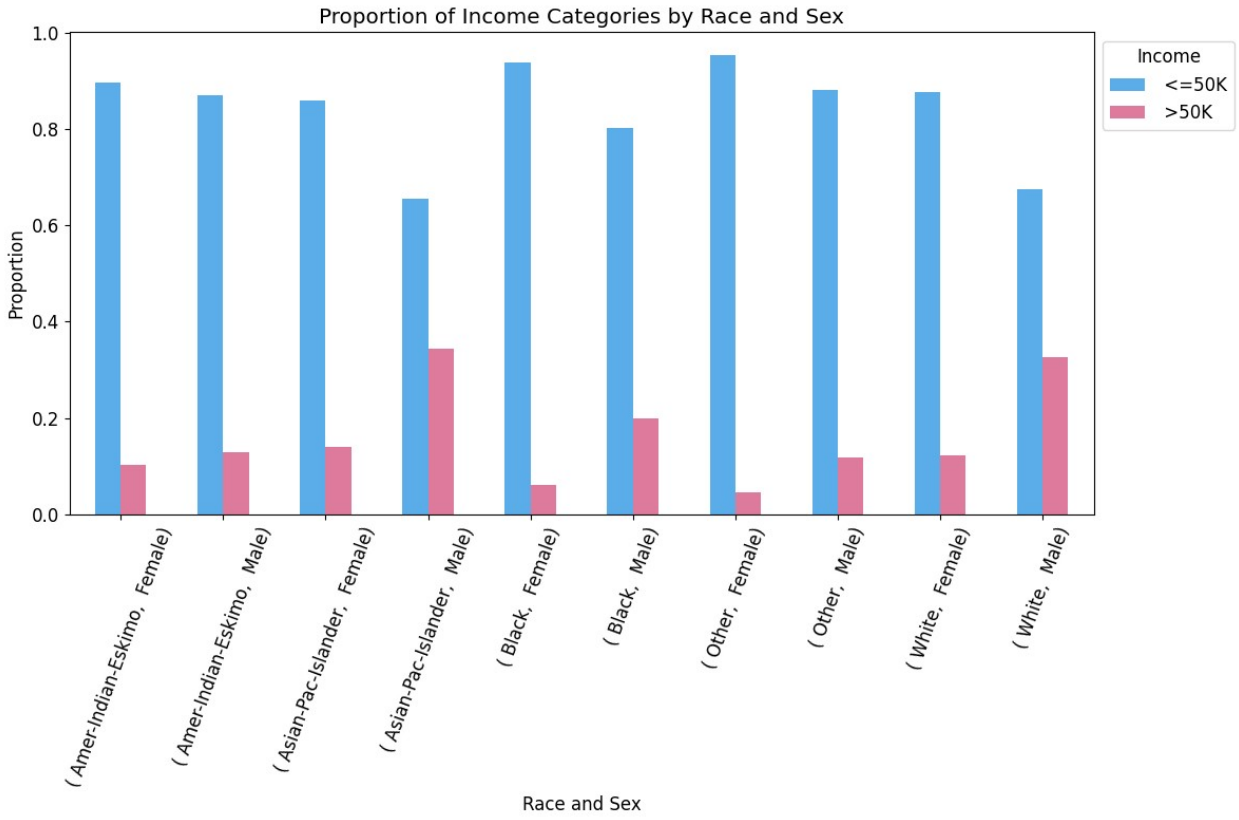


User Story 5 (Multivariate)

Race, Sex Vs. Income

```
race_sex_income_counts = df.groupby(['race', 'sex',
'income']).size().unstack().fillna(0)

# Normalizing to show proportions
```

```
race_sex_income_props =
race_sex_income_counts.div(race_sex_income_counts.sum(axis=1), axis=0)

# Plotting
fig, ax = plt.subplots(figsize=(12, 8))
colors = ['#5aade8', '#de7a9b']
race_sex_income_props.plot(kind='bar', stacked=False, ax=ax,
color=colors)
plt.title('Proportion of Income Categories by Race and Sex')
plt.xlabel('Race and Sex')
plt.ylabel('Proportion')
plt.xticks(rotation=70)
plt.legend(title='Income', loc='upper left', bbox_to_anchor=(1, 1))
plt.tight_layout()

plt.savefig('race-sex-income.png')
plt.show()
```

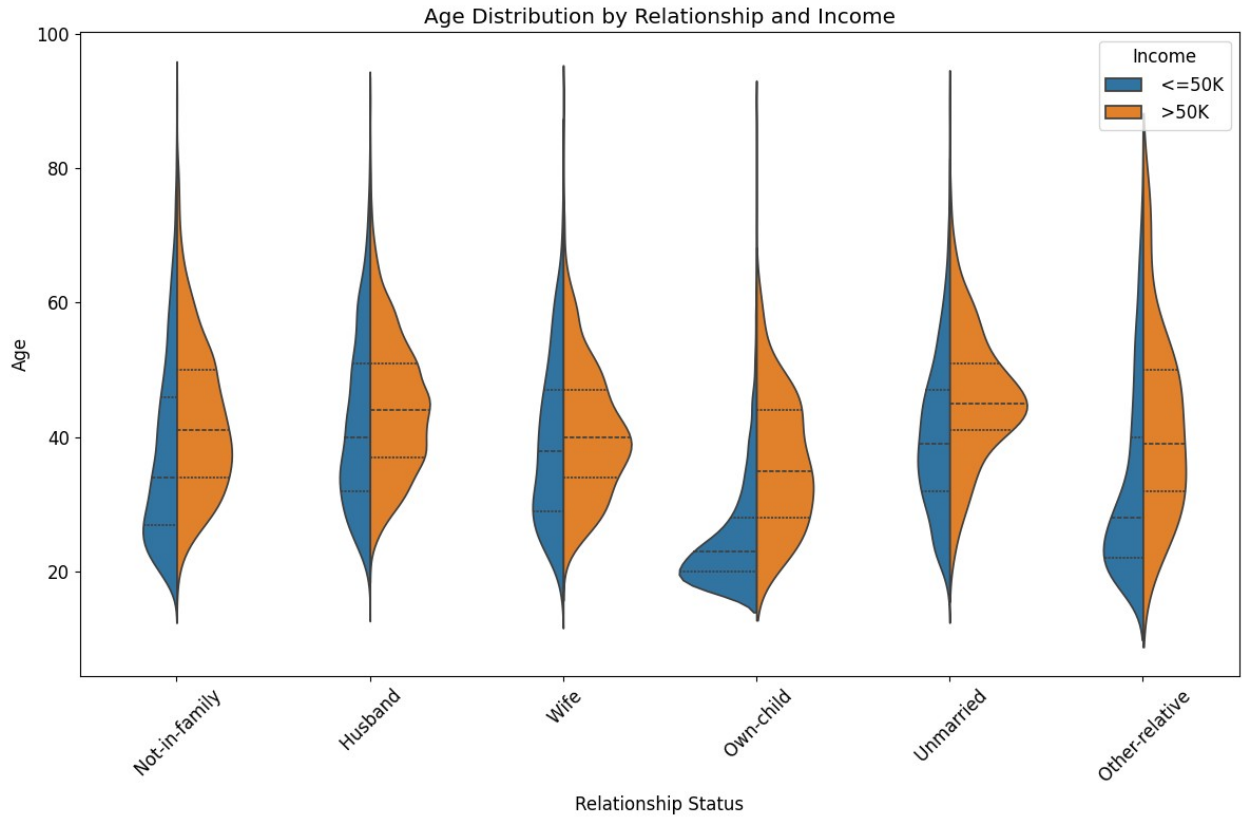


User Story 6 (Multivariate)

Age, Relationship Vs. Income

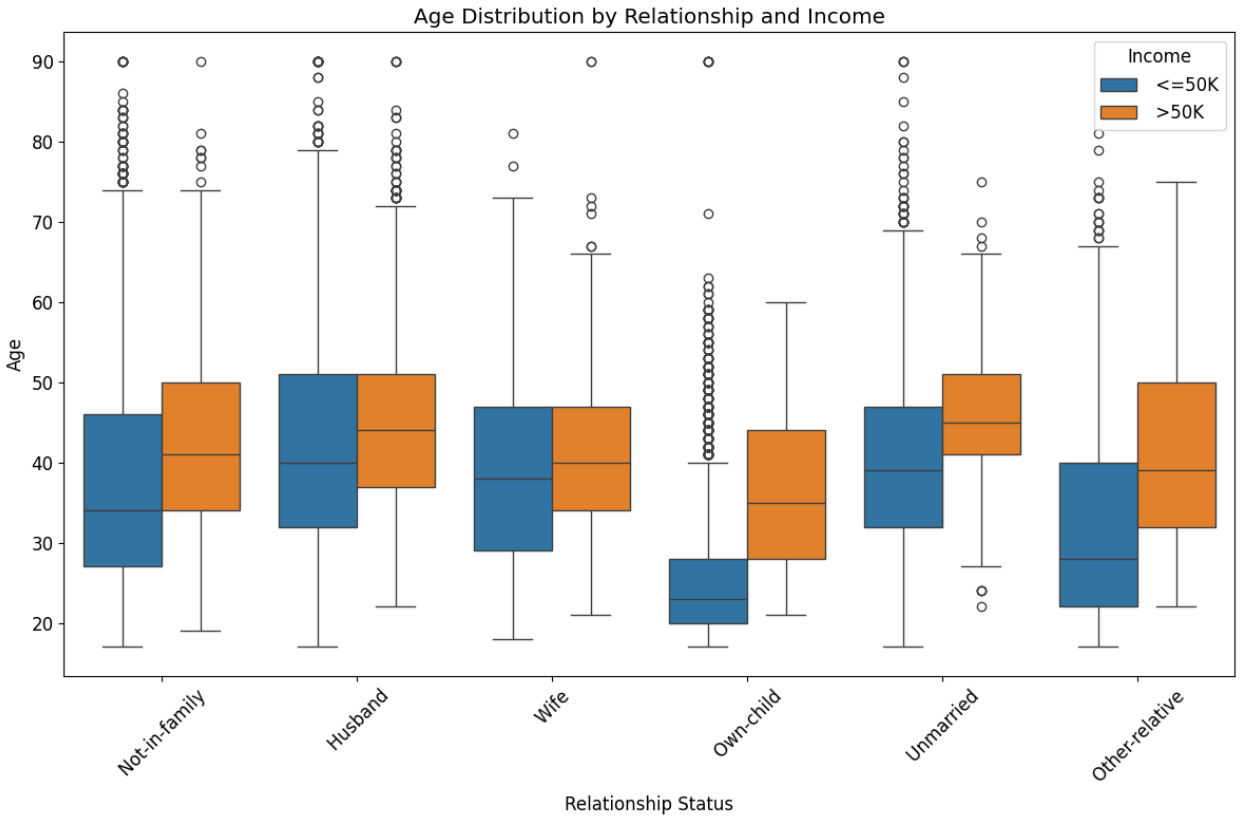
```
plt.figure(figsize=(12, 8))
sns.violinplot(x='relationship', y='age', hue='income', data=df,
split=True, inner='quartile')
```

```
plt.title('Age Distribution by Relationship and Income')
plt.xlabel('Relationship Status')
plt.ylabel('Age')
plt.xticks(rotation=45)
plt.legend(title='Income', loc='upper right')
plt.tight_layout()

plt.savefig('age-relationship-income-violin.png')
plt.show()
```



Extra

```
plt.figure(figsize=(12, 8))
sns.boxplot(x='relationship', y='age', hue='income', data=df)
plt.title('Age Distribution by Relationship and Income')
plt.xlabel('Relationship Status')
plt.ylabel('Age')
plt.xticks(rotation=45)
plt.legend(title='Income', loc='upper right')
plt.tight_layout()

plt.savefig('age-relationship-income.png')
plt.show()
```


Age Distribution by Relationship and Income
Income
<=50K
>50K
Age
90
80
70
60
50
40
30
20
Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
Relationship Status