CSE 578: Data Visualization Project

Sahil Yogesh Hadke | shadke1@asu.edu | 1229679960

Import Statements

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Read input

In [3]:
```
# Define the headers based on your provided structure
headers = [
    'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-
    'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital
    'hours-per-week', 'native-country', 'income'
]

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

In [4]:
```
df.head()
```

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationsl
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-fan
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husba
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-fan
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husba
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	W

In [5]:
```
df.count()
```

Out[5]:
```
age               32561
workclass         32561
fnlwgt            32561
education         32561
education-num     32561
marital-status    32561
occupation        32561
relationship      32561
race              32561
sex               32561
capital-gain      32561
capital-loss      32561
hours-per-week    32561
native-country    32561
income            32561
dtype: int64
```

Data Preprocessing

In [6]:
```
# drop columns which are not relevant
df = df.drop(['fnlwgt', 'education-num'], axis=1)
```

In [7]:
```
# Remove ? (NaN)
df.replace(' ?', np.nan, inplace=True)
df = df.dropna()
```

In [8]:
```
df.count()
```

Out[8]:
```
age               30162
workclass         30162
education         30162
marital-status    30162
occupation        30162
relationship      30162
race              30162
sex               30162
capital-gain      30162
capital-loss      30162
hours-per-week    30162
native-country    30162
income            30162
dtype: int64
```

Export to Excel for better understanding and exploring

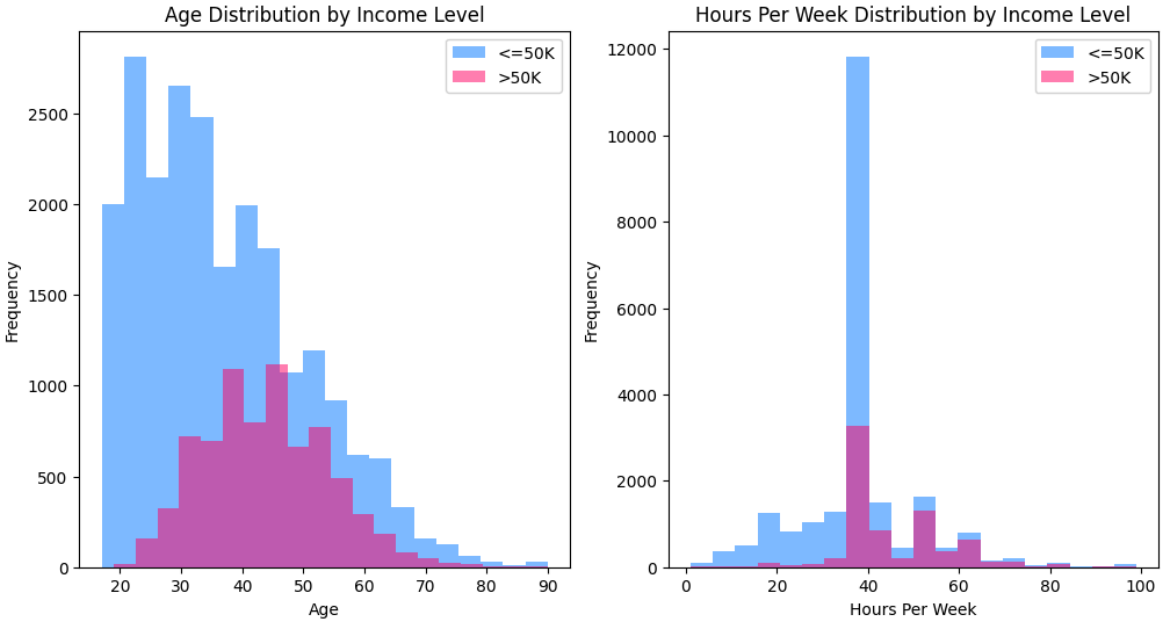
In [9]:
```
df.to_csv("./data/data.csv")
```

User Story 1 (Multivariate)

Age, Hours Per Week Vs. Income

In [28]:
```
# Age Income Analysis
income_low = df[df['income'] == ' <=50K']
income_high = df[df['income'] == ' >50K']

# Create a figure and axis
```

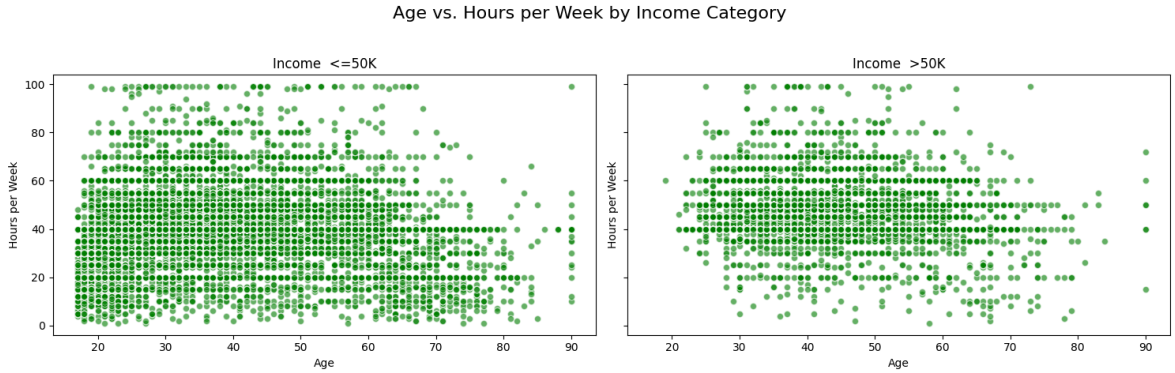
```
fig, ax = plt.subplots(1,2, figsize=(12, 6))

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

# 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
ax[1].hist(income_high['hours-per-week'], bins=20, alpha=0.5, label='>50K

# 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()


# Show the plot
plt.show()
```



Scatterplot

In [41]:
```
# Unique income categories
income_categories = df['income'].unique()

# Setting up the figure and axes
fig, axes = plt.subplots(nrows=1, ncols=len(income_categories), figsize=(

for ax, category in zip(axes, income_categories):
    # Filter data by income
    subset = df[df['income'] == category]

    # Scatter plot for age vs hours-per-week
```

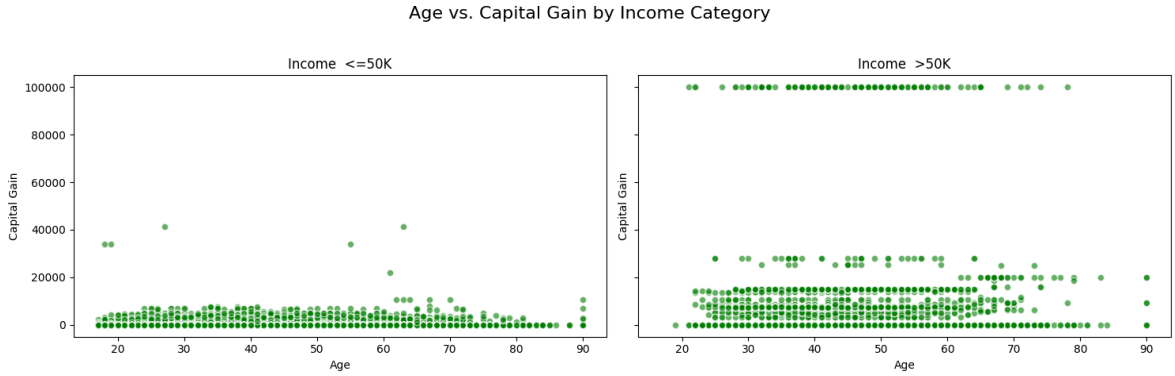
```
    sc = ax.scatter(subset['age'], subset['hours-per-week'], label=f'Inco

    ax.set_title(f'Income {category}')
    ax.set_xlabel('Age')
    ax.set_ylabel('Hours per Week')

# Add a general title
plt.suptitle('Age vs. Hours per Week by Income Category', fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust subplots to fit supti
plt.show()
```



User Story 2 (Multivariate)

Age, Capital Gain Vs. Income

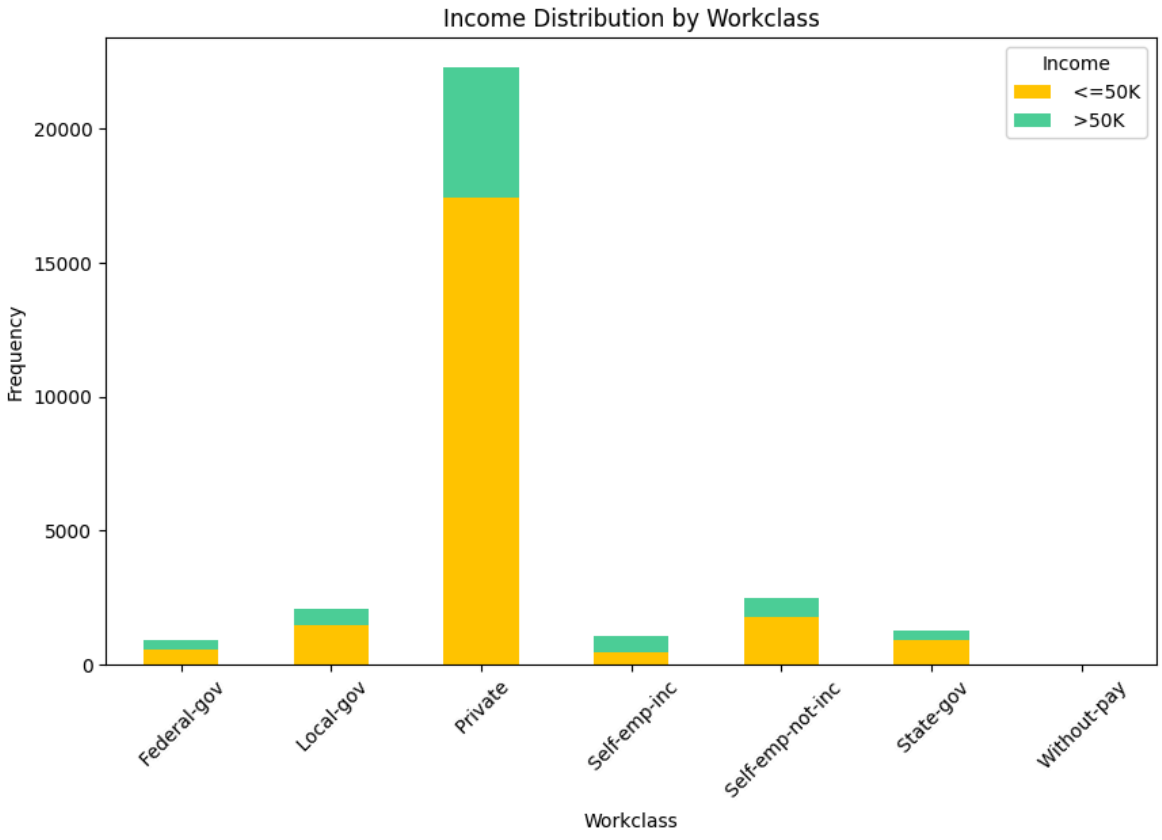
In [48]:
```
# Unique income categories
income_categories = df['income'].unique()

# Setting up the figure and axes
fig, axes = plt.subplots(nrows=1, ncols=len(income_categories), figsize=(

for ax, category in zip(axes, income_categories):
    # Filter data by income
    subset = df[df['income'] == category]

    # Scatter plot for age vs hours-per-week
    sc = ax.scatter(subset['age'], subset['capital-gain'], label=f'Income

    ax.set_title(f'Income {category}')
    ax.set_xlabel('Age')
    ax.set_ylabel('Capital Gain')

# Add a general title
plt.suptitle('Age vs. Capital Gain by Income Category', fontsize=16)
plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust subplots to fit supti
plt.show()
```



User Story 3(Univariate)

Workclass Vs. Income

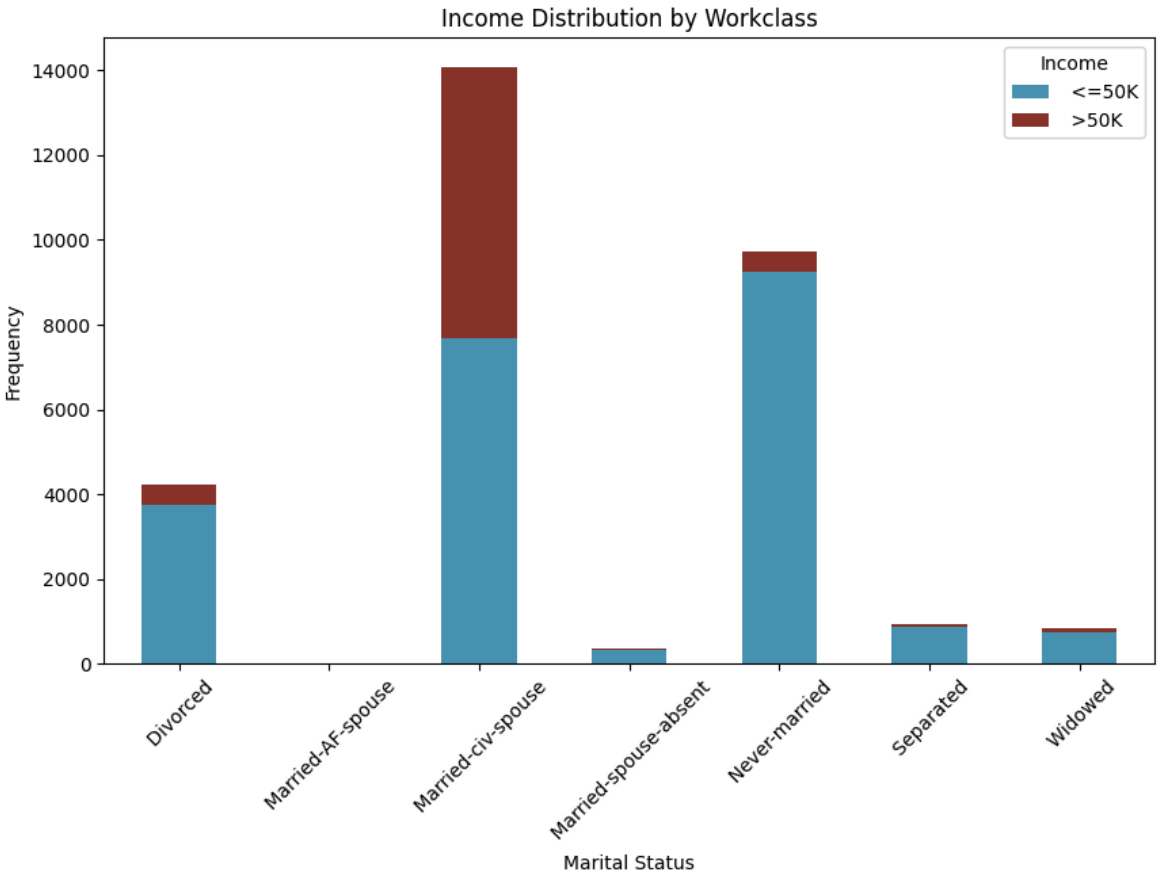
In [25]:
```
# Create a cross-tabulation of workclass and income
workclass_income_ct = pd.crosstab(df['workclass'], df['income'])

# Plotting

colors = ['#FFC700', '#4CCD99']
workclass_income_ct.plot(kind='bar', figsize=(10, 6), stacked=True, color
plt.title('Income Distribution by Workclass')
plt.xlabel('Workclass')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Income')
plt.show()
```



User Story 4 (Univariate)

Marital Status and Income

In [22]:
```
# 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
plt.title('Income Distribution by Workclass')
plt.xlabel('Marital Status')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.legend(title='Income')
plt.show()
```

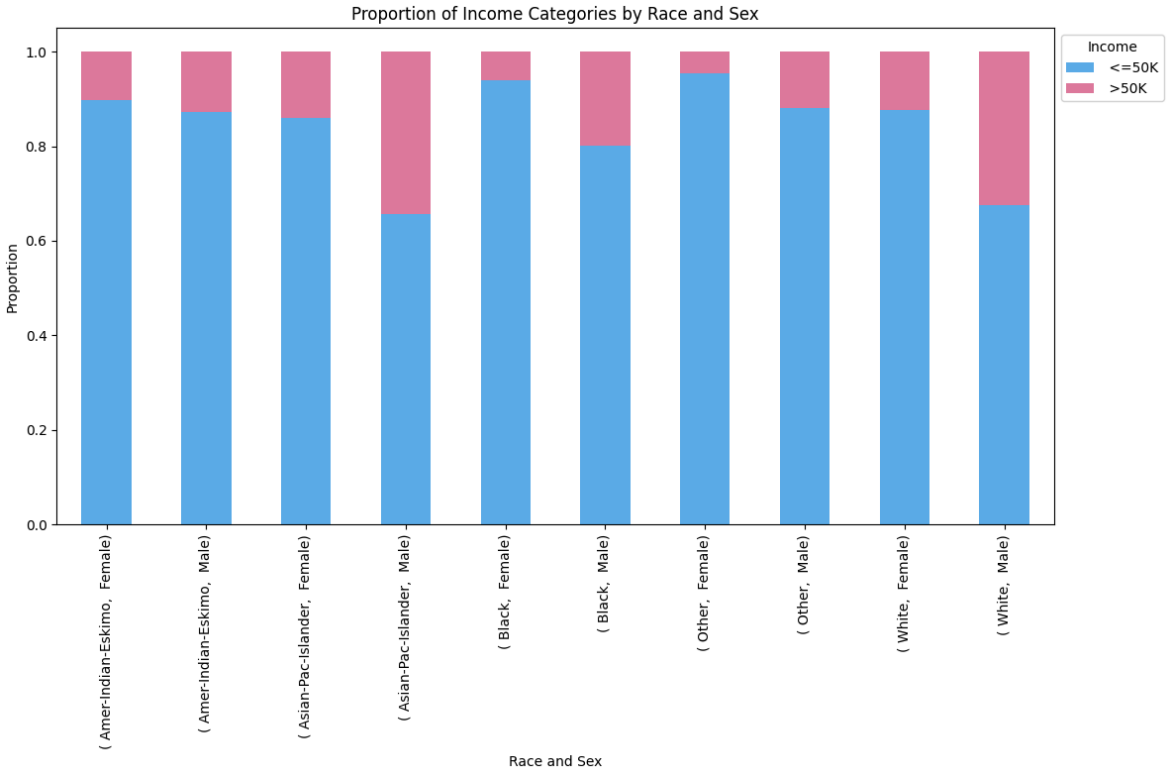


User Story 5 (Multivariate)

Race, Sex Vs. Income

In [16]:
```
race_sex_income_counts = df.groupby(['race', 'sex', 'income']).size().uns

# Normalizing to show proportions
race_sex_income_props = race_sex_income_counts.div(race_sex_income_counts

# Plotting
fig, ax = plt.subplots(figsize=(12, 8))
colors = ['#5aade8', '#de7a9b']
race_sex_income_props.plot(kind='bar', stacked=True, ax=ax, color=colors)
plt.title('Proportion of Income Categories by Race and Sex')
plt.xlabel('Race and Sex')
plt.ylabel('Proportion')
```

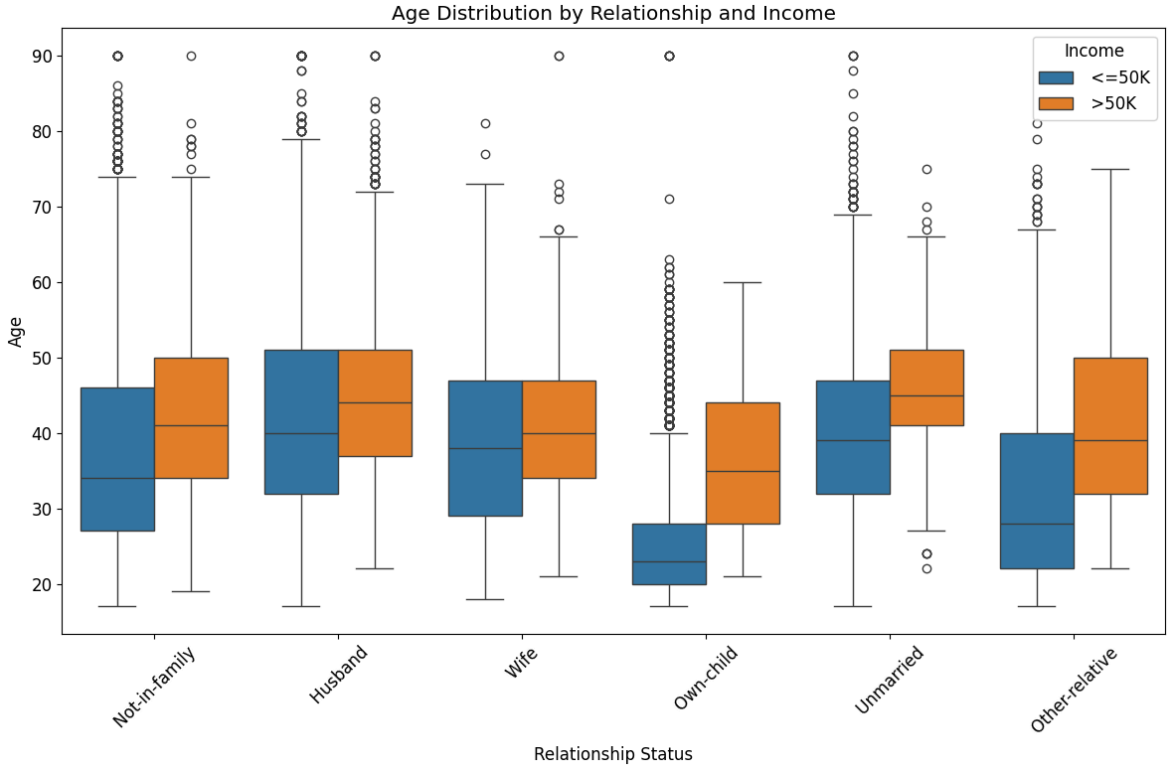
```
plt.xticks(rotation=90)
plt.legend(title='Income', loc='upper left', bbox_to_anchor=(1, 1))
plt.tight_layout()
plt.show()
```



User Story 6 (Multivariate)

Age, Relationship Vs. Income

In [93]:
```
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.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