Midterm Skills Exam: Data Wrangling and Analysis

Submitted by: Dela Cruz, Eugene D.G.

Submitted to: Engr. Roman Richard

Section: CPE22S3

Submitted on: 4/14/2024

Installing the ucimlrepo package and importing the dataset into the code

```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
census_income = fetch_ucirepo(id=20)

# data (as pandas dataframes)
X = census_income.data.features
y = census_income.data.targets

# metadata
print(census_income.metadata)

# variable information
print(census_income.variables)
```

```
{'uci_id': 20, 'name': 'Census Income', 'repository_url': 'https://archive.ics.uci.edu/dataset/20/census+income', 'data_url': 'https://archive.ics.uci.edu/static/public/20/data.csv', '
              name     role         type      demographic  \
0              age  Feature      Integer              Age
1        workclass  Feature  Categorical           Income
2           fnlwgt  Feature      Integer             None
3        education  Feature  Categorical  Education Level
4    education-num  Feature      Integer  Education Level
5   marital-status  Feature  Categorical            Other
6       occupation  Feature  Categorical            Other
7     relationship  Feature  Categorical            Other
8             race  Feature  Categorical             Race
9              sex  Feature       Binary              Sex
10    capital-gain  Feature      Integer             None
11    capital-loss  Feature      Integer             None
12  hours-per-week  Feature      Integer             None
13  native-country  Feature  Categorical            Other
14          income   Target       Binary           Income

                                      description units missing_values
0                                             N/A  None             no
1   Private, Self-emp-not-inc, Self-emp-inc, Feder...  None            yes
2                                            None  None             no
3    Bachelors, Some-college, 11th, HS-grad, Prof-...  None             no
4                                            None  None             no
5   Married-civ-spouse, Divorced, Never-married, S...  None             no
6   Tech-support, Craft-repair, Other-service, Sal...  None            yes
7   Wife, Own-child, Husband, Not-in-family, Other...  None             no
8   White, Asian-Pac-Islander, Amer-Indian-Eskimo,...  None             no
9                                  Female, Male.  None             no
10                                           None  None             no
11                                           None  None             no
12                                           None  None             no
13  United-States, Cambodia, England, Puerto-Rico,...  None            yes
14                                    >50K, <=50K.  None             no
```

```
X #display the data features
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capi
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	
...	...	...	...	...	...	...	...	...	...	...	...	

```
y #display the data targets
```

	income
0	<=50K
1	<=50K
2	<=50K
3	<=50K
4	<=50K
...	...
48837	<=50K.
48838	<=50K.
48839	<=50K.
48840	<=50K.
48841	>50K.

48842 rows × 1 columns

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

```
X.describe(include='all')
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex
count	48842.000000	47879	4.884200e+04	48842	48842.000000	48842	47876	48842	48842	48842
unique	NaN	9	NaN	16	NaN	7	15	6	5	2
top	NaN	Private	NaN	HS-grad	NaN	Married-civ-spouse	Prof-specialty	Husband	White	Male
freq	NaN	33906	NaN	15784	NaN	22379	6172	19716	41762	32650
mean	38.643585	NaN	1.896641e+05	NaN	10.078089	NaN	NaN	NaN	NaN	NaN
std	13.710510	NaN	1.056040e+05	NaN	2.570973	NaN	NaN	NaN	NaN	NaN
min	17.000000	NaN	1.228500e+04	NaN	1.000000	NaN	NaN	NaN	NaN	NaN
25%	28.000000	NaN	1.175505e+05	NaN	9.000000	NaN	NaN	NaN	NaN	NaN
50%	37.000000	NaN	1.781445e+05	NaN	10.000000	NaN	NaN	NaN	NaN	NaN

```
y.describe()
```

	income
count	48842
unique	4
top	<=50K
freq	24720

```
censusinc_df = pd.concat([X,y], axis=1)
censusinc_df
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capi
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	
...	...	...	...	...	...	...	...	...	...	...	...	...

Next steps: View recommended plots

```
# Removing the duplicate rows

censusinc_df.drop_duplicates(inplace=True)


# Check if there is still duplicate rows

duplicates = censusinc_df.duplicated().sum()
duplicates
```

```
0
```

Making numeric values of income in a dataframe

```
censusinc_df['numeric_income'] = censusinc_df['income'].map({'<=50K': 0, '<=50K.': 0, '>50K': 1, '>50K.': 1})
censusinc_df
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capi
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	
...	...	...	...	...	...	...	...	...	...	...	...	...

Next steps: View recommended plots

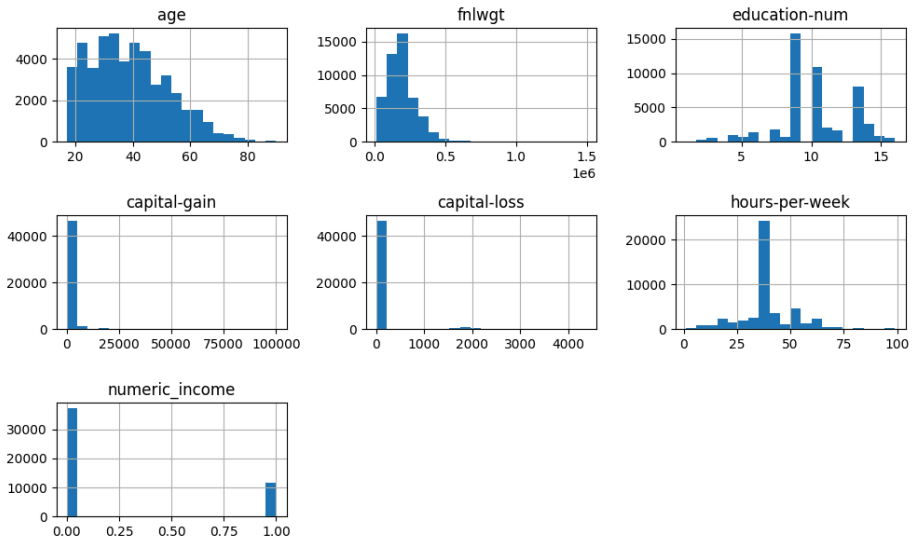
Removing typographical errors

```
censusinc_df.income.replace({'<=50K.': '<=50K','>50K.' : '>50K'},  inplace = True)
censusinc_df.income.unique()
```

```
array(['<=50K', '>50K'], dtype=object)
```

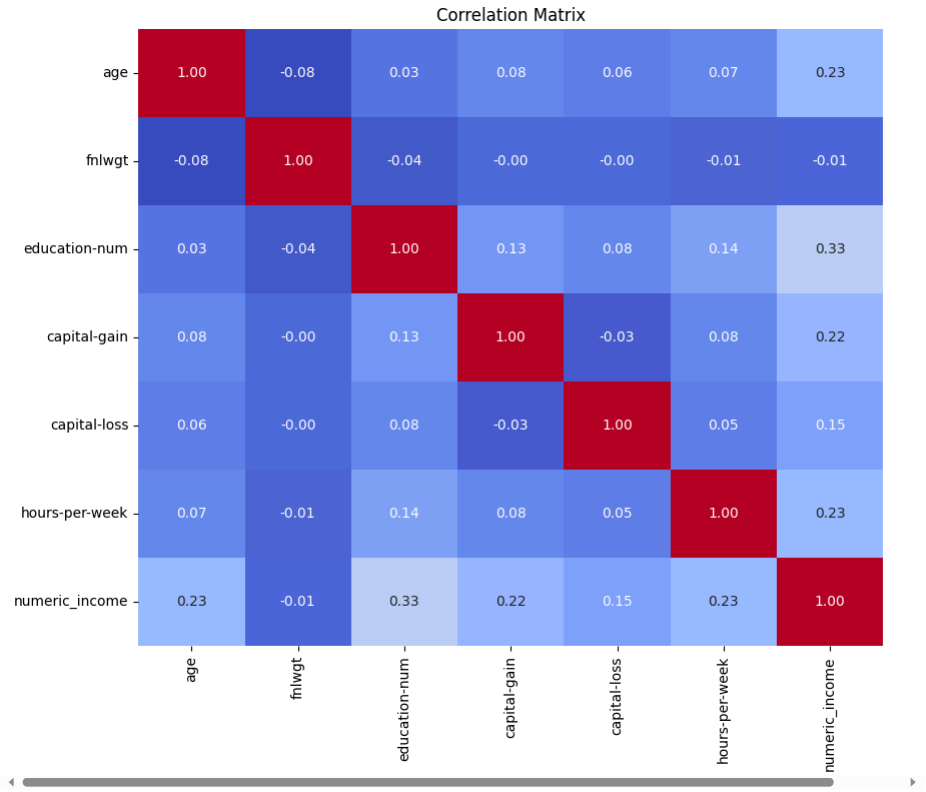
Histograph of each integer values of the data

```
censusinc_df.hist(bins=20, figsize=(10, 6))
plt.tight_layout()
plt.show()
```



Correlation matrix that includes the columns with numerical values only

```
correlation_matrix = censusinc_df.select_dtypes(include=np.number).corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

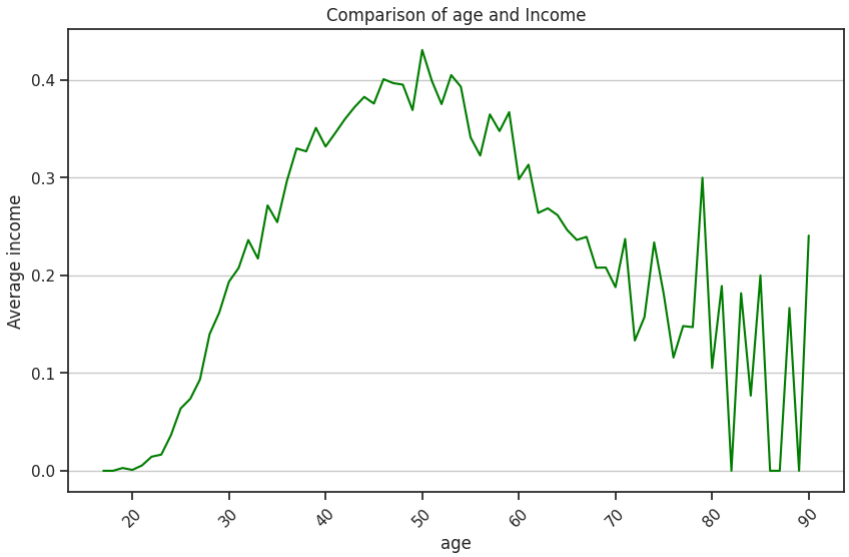


Correlation between age and income

The graph illustrates the correlation between age and income, revealing how income tends to vary with age, providing insights into potential trends or patterns in earnings over the lifespan.

Note: The 0 on income shows <=50K while the 1 shows the >50K

```
sns.set_theme(style="ticks")

plt.figure(figsize=(10, 6))
age_order = censusinc_df['age'].value_counts().index.sort_values()
age_counts = censusinc_df.groupby('age')['numeric_income'].mean().loc[age_order]
sns.lineplot(x=age_counts.index, y=age_counts.values, color='green')
plt.title('Comparison of age and Income')
plt.xlabel('age')
plt.ylabel('Average income')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```

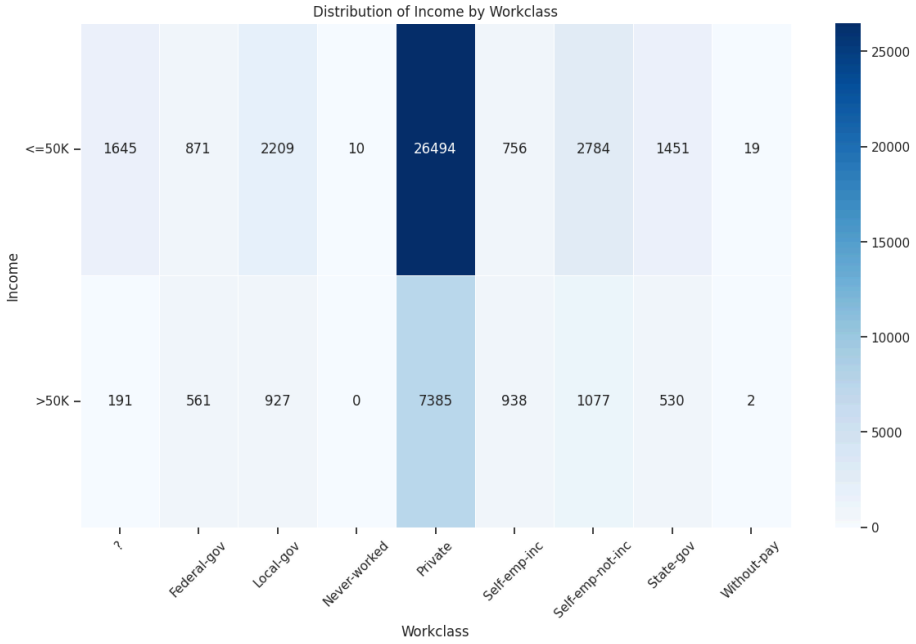


Correlation between workclass and income

The graph reveals insights into the relationship between individuals' employment status and their earnings. A positive correlation suggests that certain workclass categories may be associated with higher income levels, while a negative correlation may indicate lower income levels.

Note: The 0 on income shows <=50K while the 1 shows the >50K

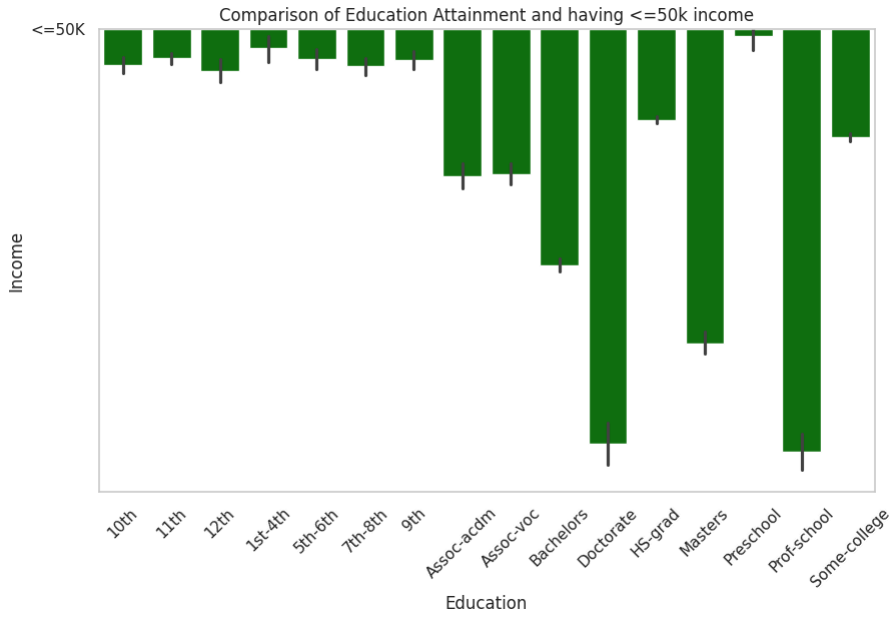
```
workclass_income_pivot = censusinc_df.pivot_table(index='income', columns='workclass', aggfunc='size', fill_value=0)

plt.figure(figsize=(14, 8))
sns.heatmap(workclass_income_pivot, cmap='Blues', annot=True, fmt='g', linewidths=.5)
plt.title('Distribution of Income by Workclass')
plt.xlabel('Workclass')
plt.ylabel('Income')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```



Correlation between education and income

The graph illustrates the correlation between education level and the likelihood of earning less than or equal to $50,000 annually. Each education level is represented on the $x-axis$, 50k income is depicted on the y-axis. The visual comparison $while the proportion of individuals earning <=$ highlights any trends or disparities in income across different education levels, offering insights into the relationship between educational attainment and income stability.

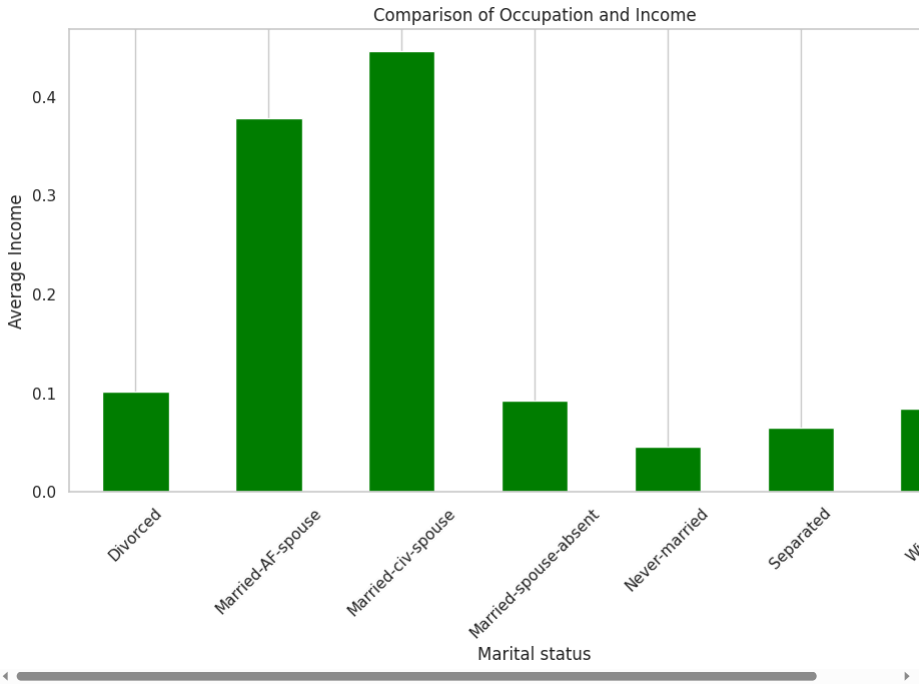
```
censusinc_df['education_numeric'] = censusinc_df['education'].map({
    '10th': 1, '11th': 2, '12th': 3, '1st-4th': 4,
    '5th-6th': 5, '7th-8th': 6, '9th': 7, 'Assoc-acdm': 8,
    'Masters': 9, 'Preschool': 10, 'Prof-school': 11, 'Some-college': 12
})

sns.set_theme(style="whitegrid")

plt.figure(figsize=(10, 6))
sns.barplot(x='education', y='income', data=censusinc_df, order=censusinc_df['education'].value_counts().index.sort_values(), color='green')
plt.title('Comparison of Education Attainment and having <=50k income')
plt.xlabel('Education')
plt.ylabel('Income')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



Correlation between marital-status and income

illustrates the correlation between marital status and income levels within the dataset. Each marital status category is represented along the x-axis, while the y-axis depicts the average income associated with each category. By visually comparing income across different marital statuses, the graph highlights any trends or disparities in income based on marital status. This provides valuable insights into how marital status may influence earning potential or socioeconomic status within the population studied.

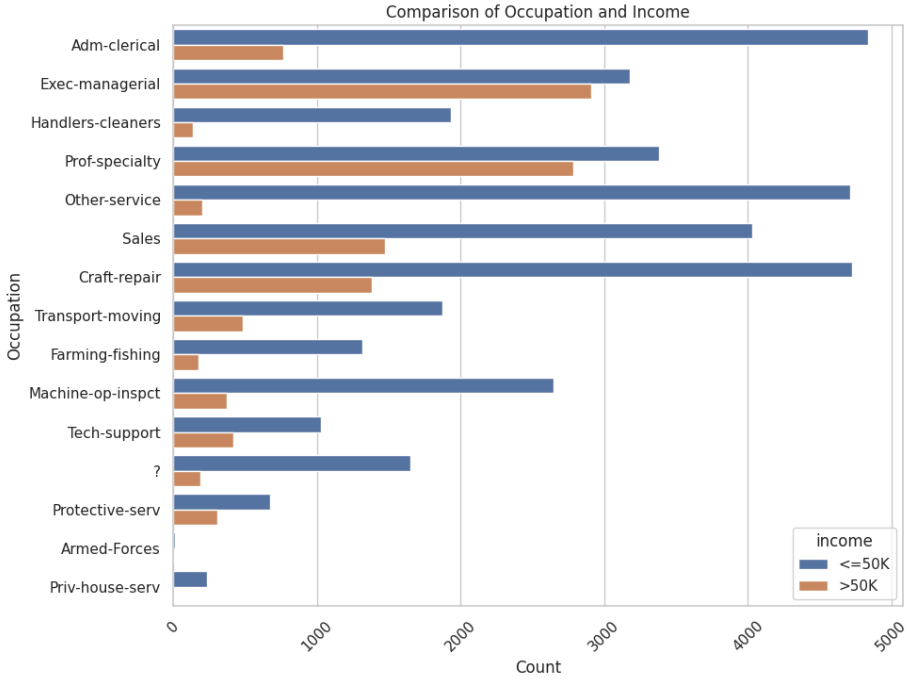
```
censusinc_df['marital_numeric'] = censusinc_df['marital-status'].map({'Married-civ-spouse': 0, 'Never-married': 1})

plt.figure(figsize=(12, 6))
marital_order = censusinc_df['marital-status'].value_counts().index.sort_values()
marital_counts = censusinc_df.groupby('marital-status')['numeric_income'].mean().loc[marital_order]
marital_counts.plot(kind='bar', color='green')
plt.title('Comparison of Occupation and Income')
plt.xlabel('Marital status')
plt.ylabel('Average Income')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



Correlation of Occupation and Income

The graph illustrates the correlation between occupation and income levels within the dataset. By visually comparing income proportions across different occupations, the graph highlights any trends or disparities in income based on occupation type. This provides valuable insights into how occupation may influence earning potential or socioeconomic status within the population studied.

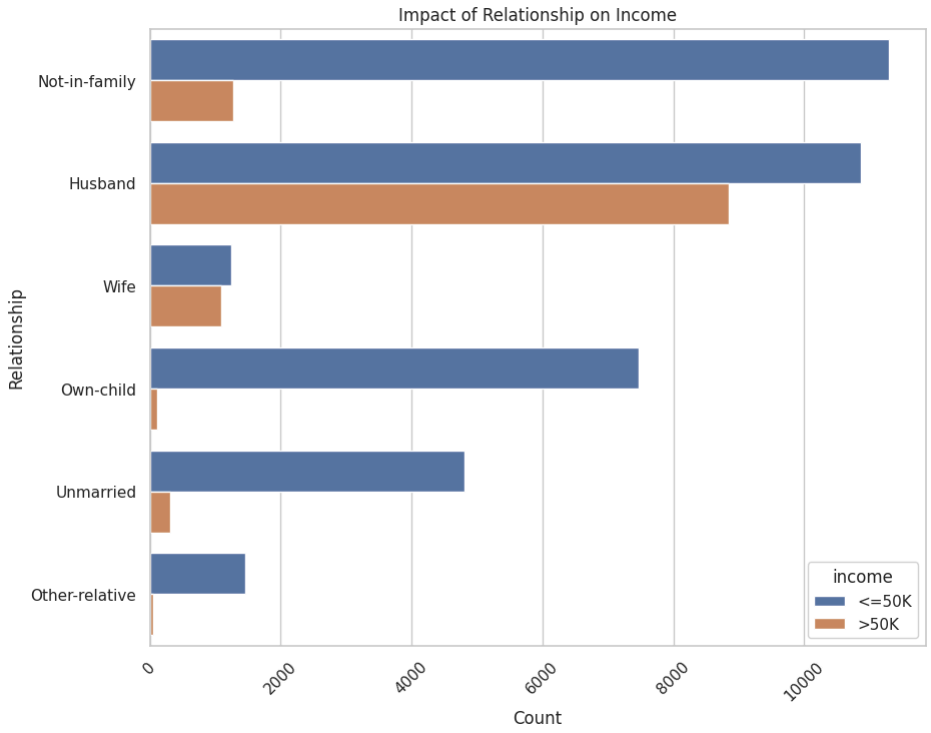
```
plt.figure(figsize=(10, 8))
sns.countplot(y='occupation', data=censusinc_df, hue='income')
plt.title('Comparison of Occupation and Income')
plt.xlabel('Count')
plt.ylabel('Occupation')
plt.xticks(rotation=45)
plt.show()
```



Correlation between relationship and income

The graph shows how being in different types of relationships relates to how much money people make. Each type of relationship is shown on the left side, and the average amount of money earned is shown on the bottom. By looking at this graph, we can see if there's a connection between the type of relationship someone has and how much they earn.

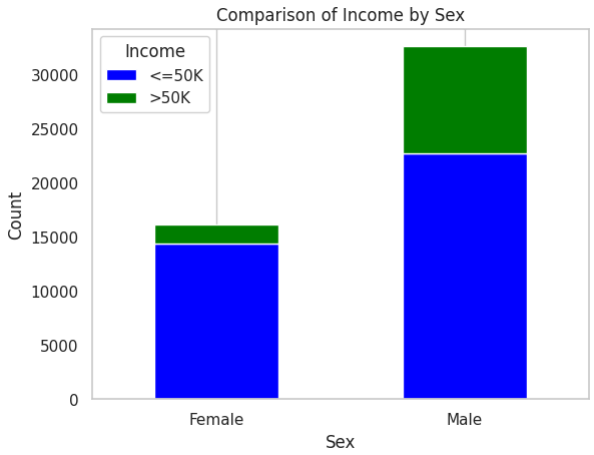
```
plt.figure(figsize=(10, 8))
sns.countplot(y='relationship', data=censusinc_df, hue='income')
plt.title('Impact of Relationship on Income')
plt.xlabel('Count')
plt.ylabel('Relationship')
plt.xticks(rotation=45)
plt.show()
```



Correlation of income by sex

illustrates the correlation between sex and income levels within the Census Income dataset. Each bar represents the count of individuals categorized by sex, with segments showing the proportion earning less than or equal to $50,000 (<= 50K) and those earning more than$ 50,000 (>50K).

```
income_counts = censusinc_df.groupby(['sex', 'income']).size().unstack()

income_counts.plot(kind='bar', stacked=True, color=['blue', 'green'])
plt.title('Comparison of Income by Sex')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.legend(title='Income', labels=['<=50K', '>50K'])
plt.grid(axis='y')
plt.show()
```

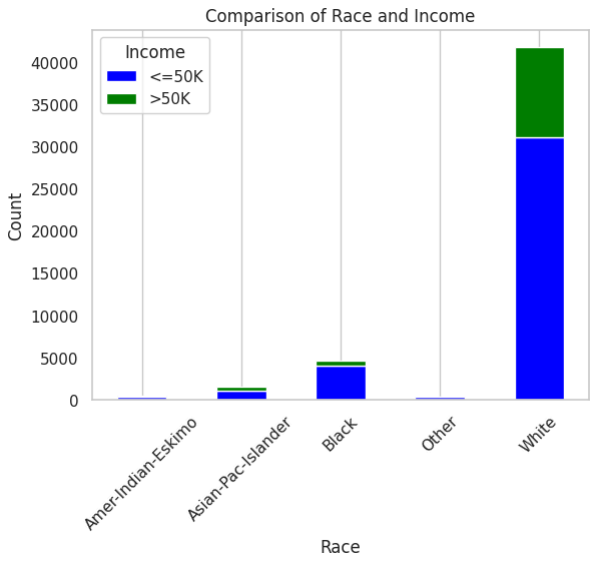


Correlation between race and income

The graph visualizes the impact of race on income levels within the dataset. Each bar represents the average income associated with a specific race category. By examining the heights of the bars, we can observe any disparities or patterns in income distribution across different racial groups.

```
plt.figure(figsize=(8, 6))
income_counts = censusinc_df.groupby(['race', 'income']).size().unstack()
income_counts.plot(kind='bar', stacked=True, color=['blue', 'green'])

plt.title('Comparison of Race and Income')
plt.xlabel('Race')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Income', labels=['<=50K', '>50K'])
plt.grid(axis='y')
plt.show()
```

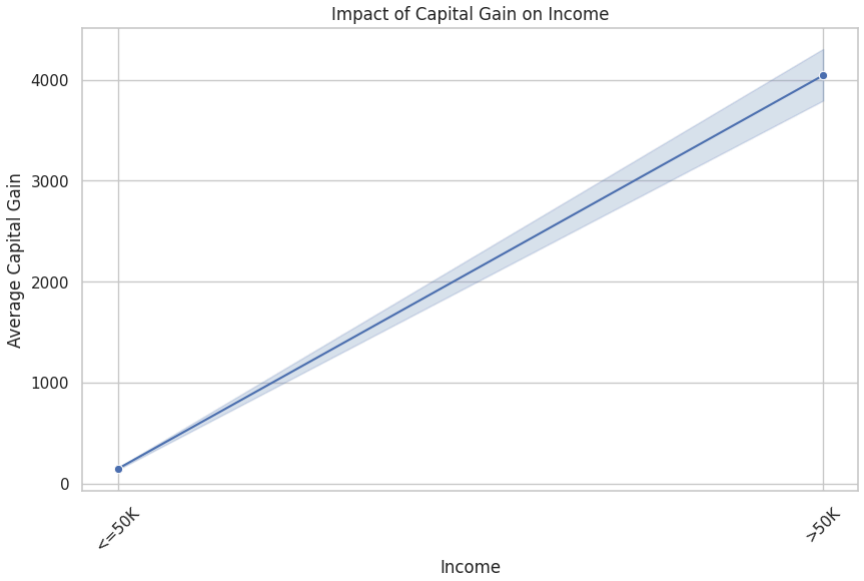
```
<Figure size 800x600 with 0 Axes>
```



Correlation between capital-gain and income

This graph looks at how capital gains relate to different income levels. It helps us see how money earned from investments affects how much money people have overall and how they build up their wealth. Understanding this connection can give us ideas about how to plan our finances and make smart investment choices.

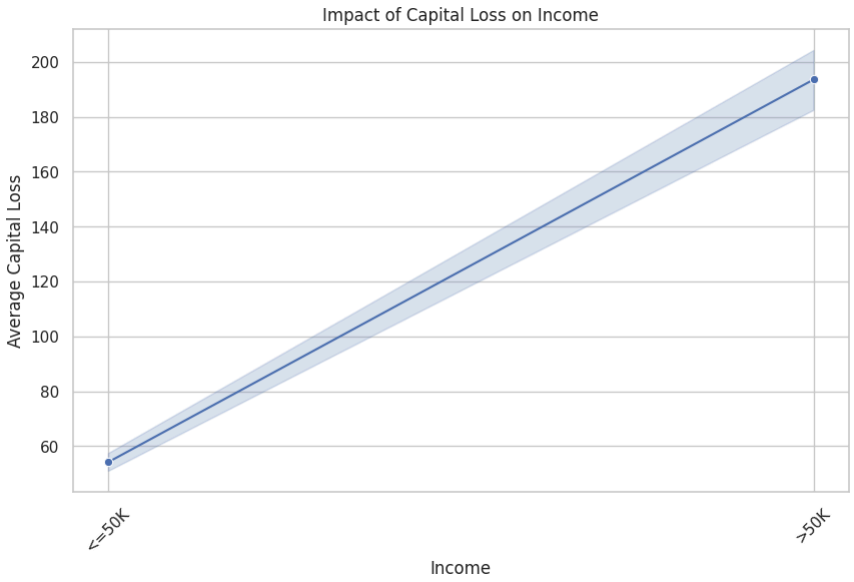
```
plt.figure(figsize=(10, 6))
sns.lineplot(x='income', y='capital-gain', data=censusinc_df, marker='o')
plt.title('Impact of Capital Gain on Income')
plt.xlabel('Income')
plt.ylabel('Average Capital Gain')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



Correlation between capital-loss and income

This graph looks at how losing money in investments relates to different income levels. It helps us see how losing money affects overall income and financial stability. Understanding this connection can help us manage risks better and make smarter investment decisions.

```
plt.figure(figsize=(10, 6))
sns.lineplot(x='income', y='capital-loss', data=censusinc_df, marker='o')
plt.title('Impact of Capital Loss on Income')
plt.xlabel('Income')
plt.ylabel('Average Capital Loss')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

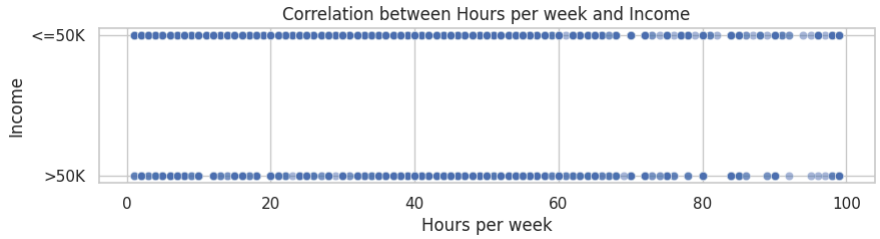


Correlation between hours-per-week and income

displays the relationship between hours worked per week and income levels within the dataset. Each point represents an individual, with their weekly work hours on the x-axis and their income on the y-axis. The plot allows us to visually examine whether there's a correlation between working longer hours and earning higher income.

```
sns.set_theme(style="whitegrid")
plt.figure(figsize=(10, 2))
sns.scatterplot(x=censusinc_df['hours-per-week'], y=censusinc_df['income'], alpha=0.5)
plt.title('Correlation between Hours per week and Income')
plt.xlabel('Hours per week')
```

```
plt.ylabel('Income')
plt.grid(True)
plt.show()
```



Correlation between native-country and income

```
country_counts = censusinc_df.groupby(['native-country', 'income']).size().unstack()

country_counts.plot(kind='bar', stacked=True, color=['blue', 'green'])
plt.title('Comparison of Income by country')
plt.xlabel('country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.legend(title='Income', labels=['<=50K', '>50K'])
plt.grid(axis='y')
plt.show()
```