```
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.
              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
```

```
print("First few rows of the dataset:")
print(X.head())
```

```
First few rows of the dataset:
   age  fnlwgt  education-num  capital-gain  capital-loss  hours-per-week  \
0   39   77516             13          2174             0              40
1   50   83311             13             0             0              13
2   38  215646              9             0             0              40
3   53  234721              7             0             0              40
4   28  338409             13             0             0              40

   workclass_Federal-gov  workclass_Local-gov  workclass_Private  \
0                  False                False              False
1                  False                False              False
2                  False                False               True
3                  False                False               True
4                  False                False               True

   workclass_Self-emp-inc  ...  native-country_Scotland  native-country_South  \
0                   False  ...                    False                 False
1                   False  ...                    False                 False
2                   False  ...                    False                 False
3                   False  ...                    False                 False
4                   False  ...                    False                 False

   native-country_Taiwan  native-country_Thailand  \
0                  False                    False
1                  False                    False
2                  False                    False
3                  False                    False
4                  False                    False

   native-country_Trinadad&Tobago  native-country_United-States  \
0                           False                         True
1                           False                         True
2                           False                         True
```

```
3                          False                         True
4                          False                        False

   native-country_Vietnam  native-country_Yugoslavia  income  outcome
0                   False                      False   <=50K    <=50K
1                   False                      False   <=50K    <=50K
2                   False                      False   <=50K    <=50K
3                   False                      False   <=50K    <=50K
4                   False                      False   <=50K    <=50K

[5 rows x 106 columns]
```

```
print("\nSummary of the dataset:")
print(X.info())  # Changed from df.info()
```

```
Summary of the dataset:
<class 'pandas.core.frame.DataFrame'>
Index: 45222 entries, 0 to 48841
Columns: 106 entries, age to outcome
dtypes: bool(98), int64(6), object(2)
memory usage: 7.3+ MB
None
```

```
print("\nDescriptive statistics:")
print(X.describe())  # Changed from df.describe()))
```

```
Descriptive statistics:
                age        fnlwgt  education-num  capital-gain  capital-loss  \
count  45222.000000  4.522200e+04   45222.000000  45222.000000  45222.000000
mean      38.547941  1.897347e+05      10.118460   1101.430344     88.595418
std       13.217870  1.056392e+05       2.552881   7506.430084    404.956092
min       17.000000  1.349200e+04       1.000000      0.000000      0.000000
25%       28.000000  1.173882e+05       9.000000      0.000000      0.000000
50%       37.000000  1.783160e+05      10.000000      0.000000      0.000000
75%       47.000000  2.379260e+05      13.000000      0.000000      0.000000
max       90.000000  1.490400e+06      16.000000  99999.000000   4356.000000

       hours-per-week
count    45222.000000
mean        40.938017
std         12.007508
min          1.000000
25%         40.000000
50%         40.000000
75%         45.000000
max         99.000000
```

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

```
# Fetch the dataset
from ucimlrepo import fetch_ucirepo
census_income = fetch_ucirepo(id=20)
X = census_income.data.features
y = census_income.data.targets
```

```
print(census_income.metadata)
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.
               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
```

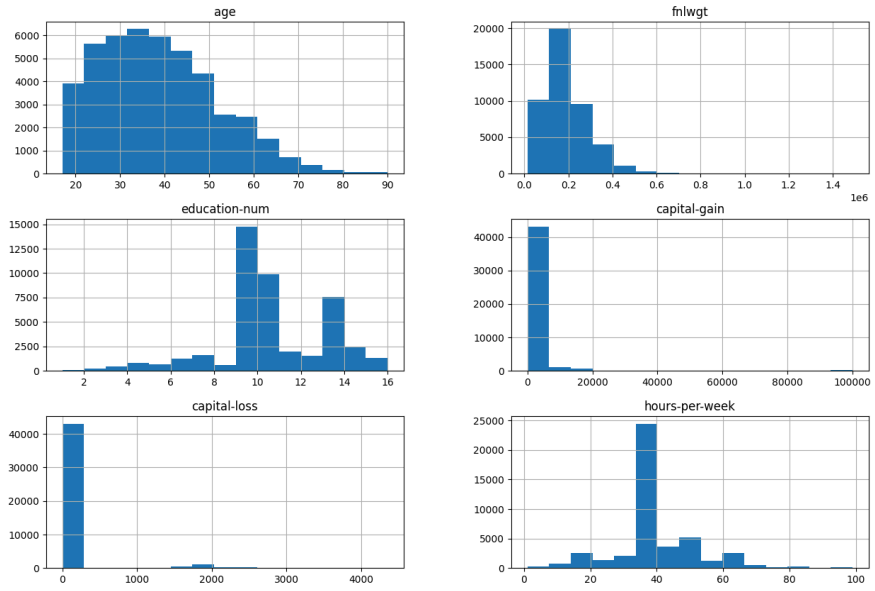
```
12                                          None  None            no
13  United-States, Cambodia, England, Puerto-Rico,...  None           yes
14                                  >50K, <=50K.  None            no
```

```
X.replace('?', np.nan, inplace=True)
X.dropna(inplace=True)  # or you could use an imputation strategy
```

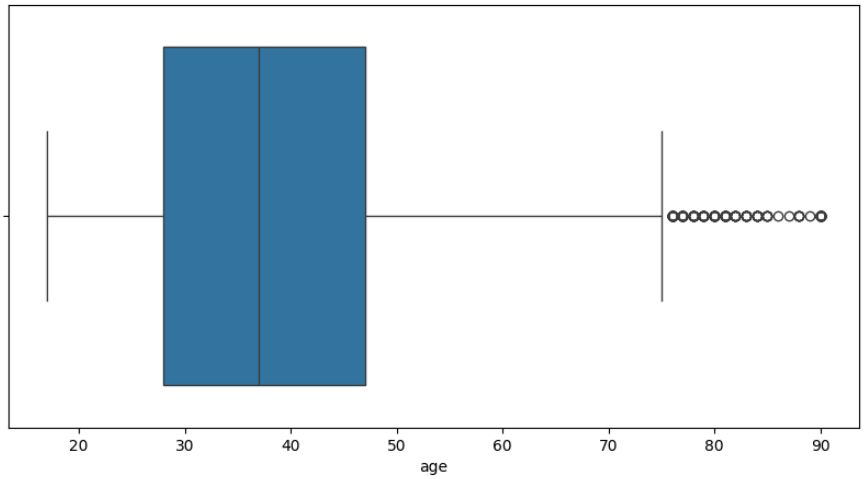
```
<ipython-input-40-2e10e0e9c1ff>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X.replace('?', np.nan, inplace=True)
<ipython-input-40-2e10e0e9c1ff>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X.dropna(inplace=True)  # or you could use an imputation strategy
```

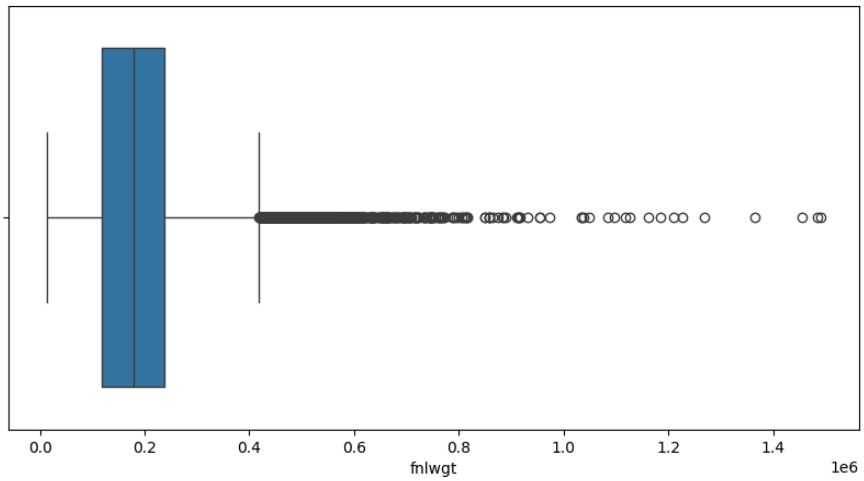
```
X = pd.get_dummies(X)
```

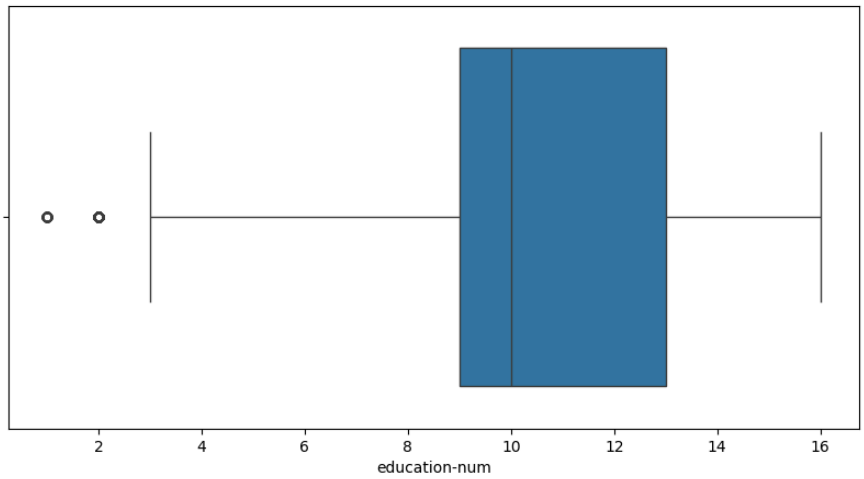
```
X.hist(bins=15, figsize=(15, 10))
plt.show()
```

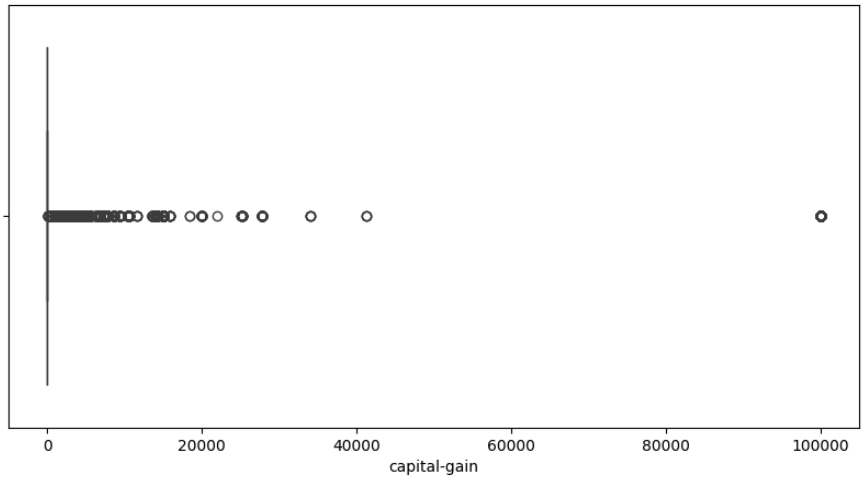


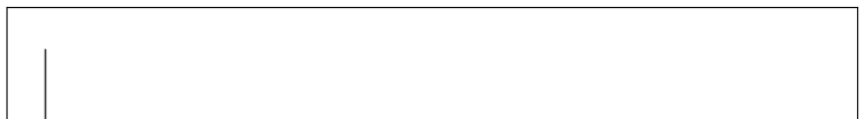

```
for column in X.select_dtypes(include=[np.number]).columns:
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=X[column])
    plt.show()
```

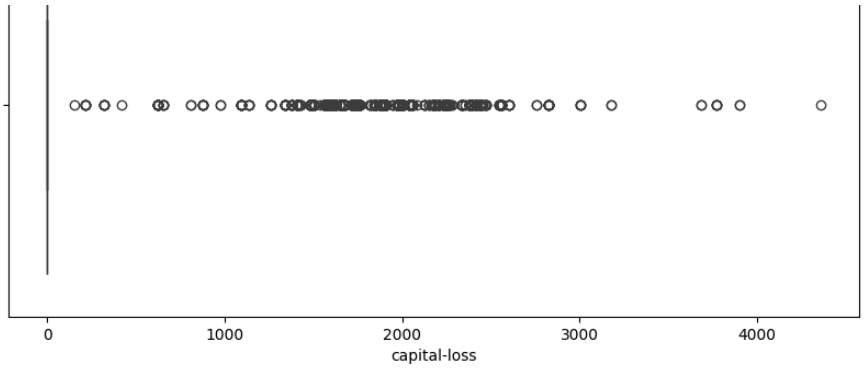

20
30
40
50
60
70
80
90
age

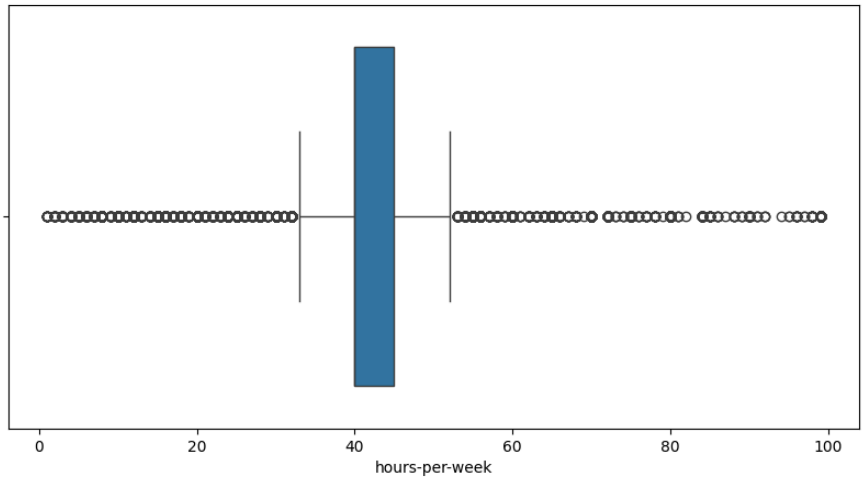

0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
fnlwgt
1e6


2
4
6
8
10
12
14
16
education-num

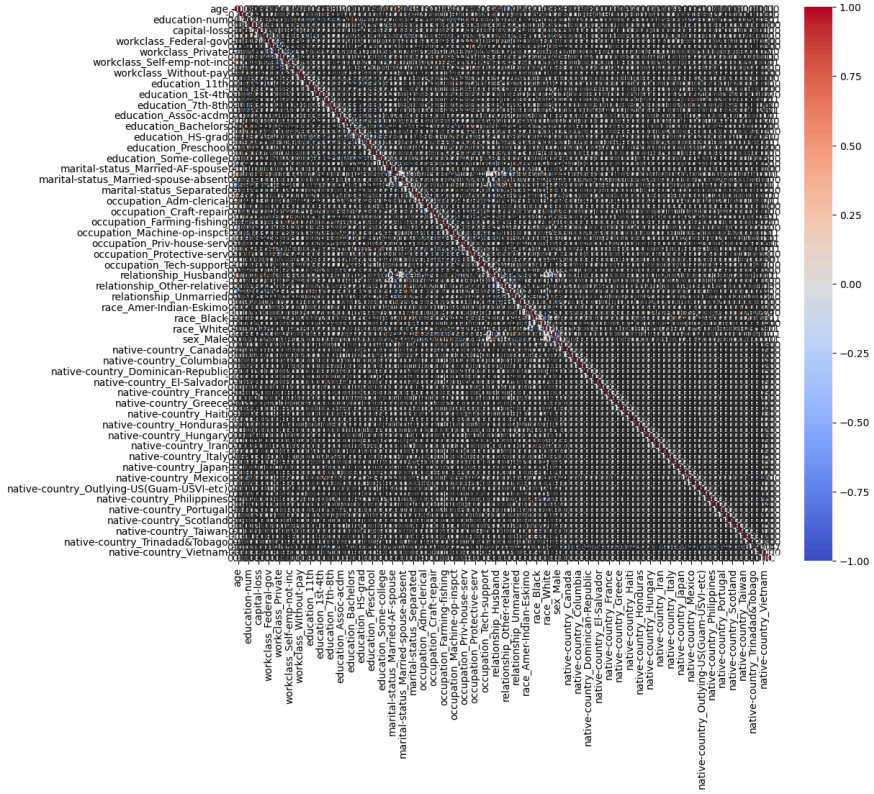

0
20000
40000
60000
80000
100000
capital-gain





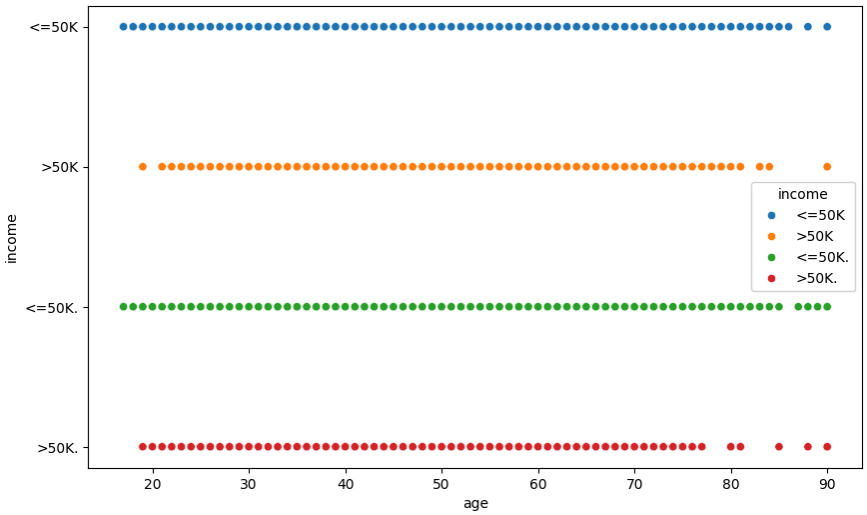

```
plt.figure(figsize=(12, 10))
sns.heatmap(X.corr(), annot=True, fmt=".2f", cmap='coolwarm')
plt.show()
```



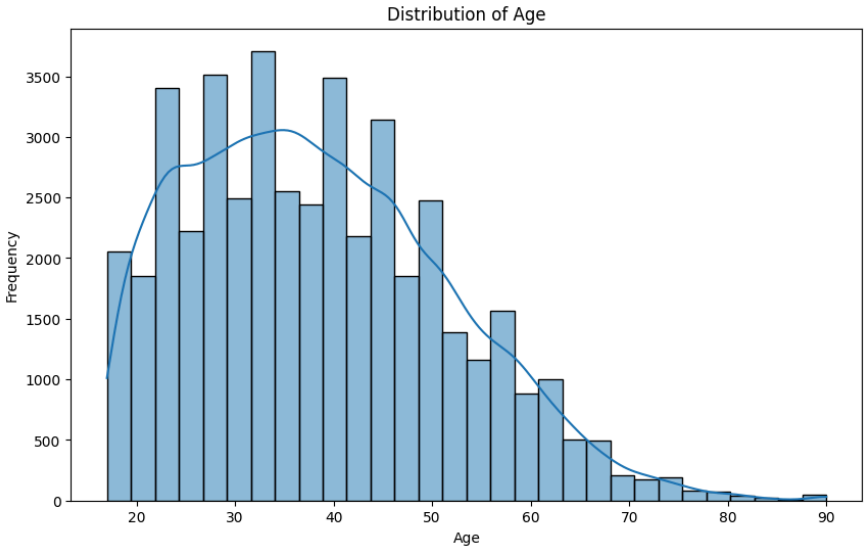

```
print(X.columns)
```

```
Index(['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss',
       'hours-per-week', 'workclass_Federal-gov', 'workclass_Local-gov',
       'workclass_Private', 'workclass_Self-emp-inc',
       ...
       'native-country_Portugal', 'native-country_Puerto-Rico',
       'native-country_Scotland', 'native-country_South',
       'native-country_Taiwan', 'native-country_Thailand',
       'native-country_Trinadad&Tobago', 'native-country_United-States',
       'native-country_Vietnam', 'native-country_Yugoslavia'],
      dtype='object', length=104)
```

```
# If 'y' is a series and should be used as the 'hue', you might need to add it back to the dataframe or adjust your code.
X['income'] = y  # Only do this if it makes sense for your analysis; otherwise, handle separately.
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age', y='income', data=X, hue='income')
plt.show()
```



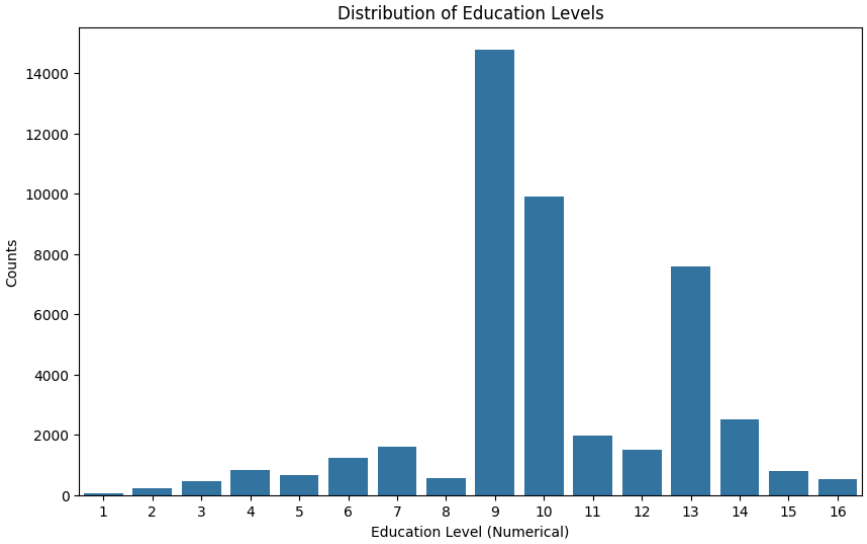

```
import matplotlib.pyplot as plt
import seaborn as sns

# Example: Plotting the distribution of 'age'
plt.figure(figsize=(10, 6))
sns.histplot(data=X, x='age', bins=30, kde=True)  # Assuming 'age' is a column in your DataFrame X
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



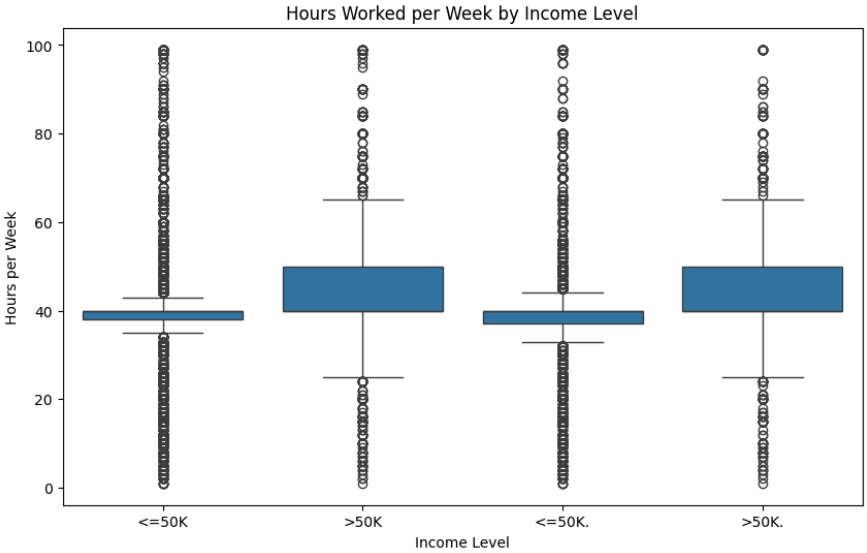

```
print(X.columns)
```

```
Index(['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss',
       'hours-per-week', 'workclass_Federal-gov', 'workclass_Local-gov',
       'workclass_Private', 'workclass_Self-emp-inc',
       ...
       'native-country_Scotland', 'native-country_South',
       'native-country_Taiwan', 'native-country_Thailand',
       'native-country_Trinadad&Tobago', 'native-country_United-States',
       'native-country_Vietnam', 'native-country_Yugoslavia', 'income',
       'outcome'],
      dtype='object', length=106)
```

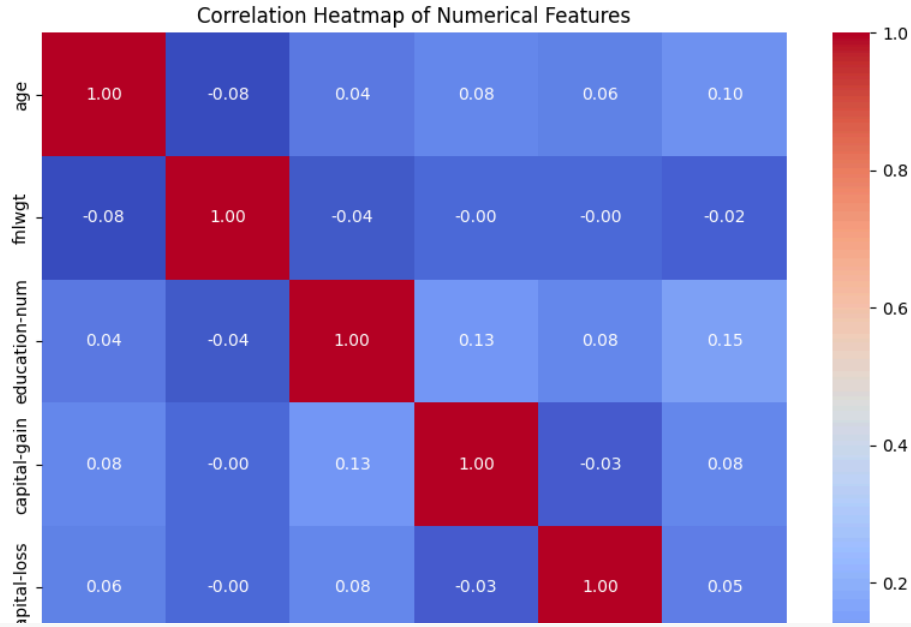
```
plt.figure(figsize=(10, 6))
sns.countplot(data=X, x='education-num')  # Using 'education-num' instead of 'education'
plt.title('Distribution of Education Levels')
plt.xlabel('Education Level (Numerical)')
plt.ylabel('Counts')
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.boxplot(x='income', y='hours-per-week', data=X)
plt.title('Hours Worked per Week by Income Level')
plt.xlabel('Income Level')
plt.ylabel('Hours per Week')
plt.show()
```




```
numeric_columns = X.select_dtypes(include=['number'])
correlation_matrix = numeric_columns.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title('Correlation Heatmap of Numerical Features')
plt.show()
```




```
subset = X[['age', 'hours-per-week', 'education-num', 'income']]  # Adjust column names as needed
sns.pairplot(subset, hue='income', diag_kind='kde', markers=["o", "s"])  # Ensure 'income' is available or adjust accordingly
plt.title('Pair Plot of Age, Hours per Week, and Education Level')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
The markers list has fewer values (2) than needed (4) and will cycle, which may produce an uninterpretable plot.
  func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
The markers list has fewer values (2) than needed (4) and will cycle, which may produce an uninterpretable plot.
  func(x=x, y=y, **kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:1615: UserWarning:
The markers list has fewer values (2) than needed (4) and will cycle, which may produce an uninterpretable plot
```