EDA_adult_income

March 23, 2023

1 EDA on Adult Census Income

Dataset link: https://archive.ics.uci.edu/ml/datasets/adult

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.express as px
import plotly.graph_objects as go

pd.pandas.set_option('display.max_columns', None)
```

[2]:
```
df = pd.read_csv('adult.data',names = ['age',␣
↪'workclass','fnlwgt','education','education-num','marital-status','occupation','relationshi
```

[3]:
```
df.head()
```

[3]:
```
   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
```

[4]:
```
#drop duplicates

df.drop_duplicates(keep='first',inplace=True)
```

[5]:
```
df.shape
```

[5]:
```
(32537, 15)
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 32537 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32537 non-null  int64
 1   workclass       32537 non-null  object
 2   fnlwgt          32537 non-null  int64
 3   education       32537 non-null  object
 4   education-num   32537 non-null  int64
 5   marital-status  32537 non-null  object
 6   occupation      32537 non-null  object
 7   relationship    32537 non-null  object
 8   race            32537 non-null  object
 9   sex             32537 non-null  object
 10  capital-gain    32537 non-null  int64
 11  capital-loss    32537 non-null  int64
 12  hours-per-week  32537 non-null  int64
 13  native-country  32537 non-null  object
 14  income          32537 non-null  object
dtypes: int64(6), object(9)
memory usage: 4.0+ MB
```

1.1 Obeservation:

1. There are total 32537 rows and 15 columns in the dataset
2. Categorical features = 9 and Numerical features = 6

[7]:
```
for i in df.columns:
    print(df[i].unique())
```

```
[39 50 38 53 28 37 49 52 31 42 30 23 32 40 34 25 43 54 35 59 56 19 20 45
 22 48 21 24 57 44 41 29 18 47 46 36 79 27 67 33 76 17 55 61 70 64 71 68
 66 51 58 26 60 90 75 65 77 62 63 80 72 74 69 73 81 78 88 82 83 84 85 86
 87]
[' State-gov' ' Self-emp-not-inc' ' Private' ' Federal-gov' ' Local-gov'
 ' ?' ' Self-emp-inc' ' Without-pay' ' Never-worked']
```

```
[ 77516  83311 215646 …  34066  84661 257302]
[' Bachelors' ' HS-grad' ' 11th' ' Masters' ' 9th' ' Some-college'
 ' Assoc-acdm' ' Assoc-voc' ' 7th-8th' ' Doctorate' ' Prof-school'
 ' 5th-6th' ' 10th' ' 1st-4th' ' Preschool' ' 12th']
[13  9  7 14  5 10 12 11  4 16 15  3  6  2  1  8]
[' Never-married' ' Married-civ-spouse' ' Divorced'
 ' Married-spouse-absent' ' Separated' ' Married-AF-spouse' ' Widowed']
[' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' ' ?'
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
[' Not-in-family' ' Husband' ' Wife' ' Own-child' ' Unmarried'
 ' Other-relative']
[' White' ' Black' ' Asian-Pac-Islander' ' Amer-Indian-Eskimo' ' Other']
[' Male' ' Female']
[ 2174     0 14084  5178  5013  2407 14344 15024  7688 34095  4064  4386
  7298  1409  3674  1055  3464  2050  2176   594 20051  6849  4101  1111
  8614  3411  2597 25236  4650  9386  2463  3103 10605  2964  3325  2580
  3471  4865 99999  6514  1471  2329  2105  2885 25124 10520  2202  2961
 27828  6767  2228  1506 13550  2635  5556  4787  3781  3137  3818  3942
   914   401  2829  2977  4934  2062  2354  5455 15020  1424  3273 22040
  4416  3908 10566   991  4931  1086  7430  6497   114  7896  2346  3418
  3432  2907  1151  2414  2290 15831 41310  4508  2538  3456  6418  1848
  3887  5721  9562  1455  2036  1831 11678  2936  2993  7443  6360  1797
  1173  4687  6723  2009  6097  2653  1639 18481  7978  2387  5060]
[   0 2042 1408 1902 1573 1887 1719 1762 1564 2179 1816 1980 1977 1876
 1340 2206 1741 1485 2339 2415 1380 1721 2051 2377 1669 2352 1672  653
 2392 1504 2001 1590 1651 1628 1848 1740 2002 1579 2258 1602  419 2547
 2174 2205 1726 2444 1138 2238  625  213 1539  880 1668 1092 1594 3004
 2231 1844  810 2824 2559 2057 1974  974 2149 1825 1735 1258 2129 2603
 2282  323 4356 2246 1617 1648 2489 3770 1755 3683 2267 2080 2457  155
 3900 2201 1944 2467 2163 2754 2472 1411]
[40 13 16 45 50 80 30 35 60 20 52 44 15 25 38 43 55 48 58 32 70  2 22 56
 41 28 36 24 46 42 12 65  1 10 34 75 98 33 54  8  6 64 19 18 72  5  9 47
 37 21 26 14  4 59  7 99 53 39 62 57 78 90 66 11 49 84  3 17 68 27 85 31
 51 77 63 23 87 88 73 89 97 94 29 96 67 82 86 91 81 76 92 61 74 95]
[' United-States' ' Cuba' ' Jamaica' ' India' ' ?' ' Mexico' ' South'
 ' Puerto-Rico' ' Honduras' ' England' ' Canada' ' Germany' ' Iran'
 ' Philippines' ' Italy' ' Poland' ' Columbia' ' Cambodia' ' Thailand'
 ' Ecuador' ' Laos' ' Taiwan' ' Haiti' ' Portugal' ' Dominican-Republic'
 ' El-Salvador' ' France' ' Guatemala' ' China' ' Japan' ' Yugoslavia'
 ' Peru' ' Outlying-US(Guam-USVI-etc)' ' Scotland' ' Trinadad&Tobago'
 ' Greece' ' Nicaragua' ' Vietnam' ' Hong' ' Ireland' ' Hungary'
 ' Holand-Netherlands']
[' <=50K' ' >50K']
```

[8]:
```
df.isnull().sum()
```

[8]:
```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

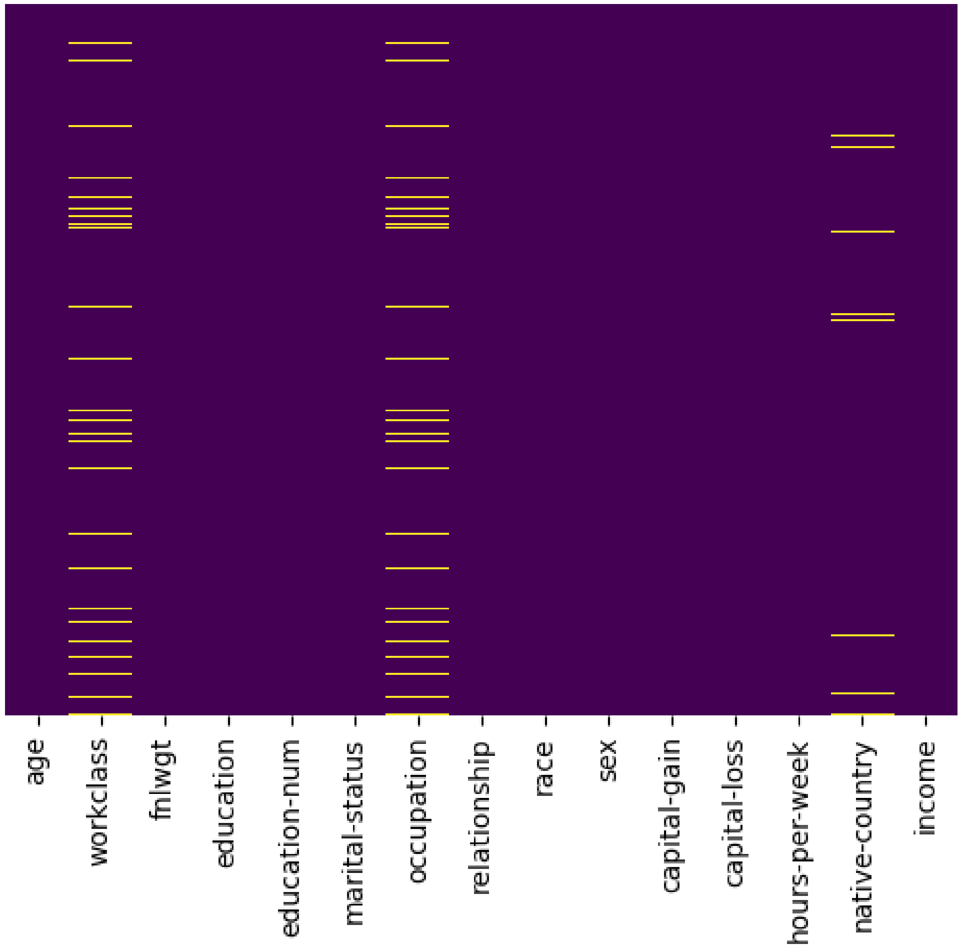
1.2 Observation:

1. '?' seems to be NaN values

[9]:
```
#Check Null values

df.replace(' ?',np.nan,inplace=True) #replacing '?' with NaN
```

[10]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

[10]:
```
<AxesSubplot: >
```



1.3 Observation:

workclass, occupation and native_country has missing values

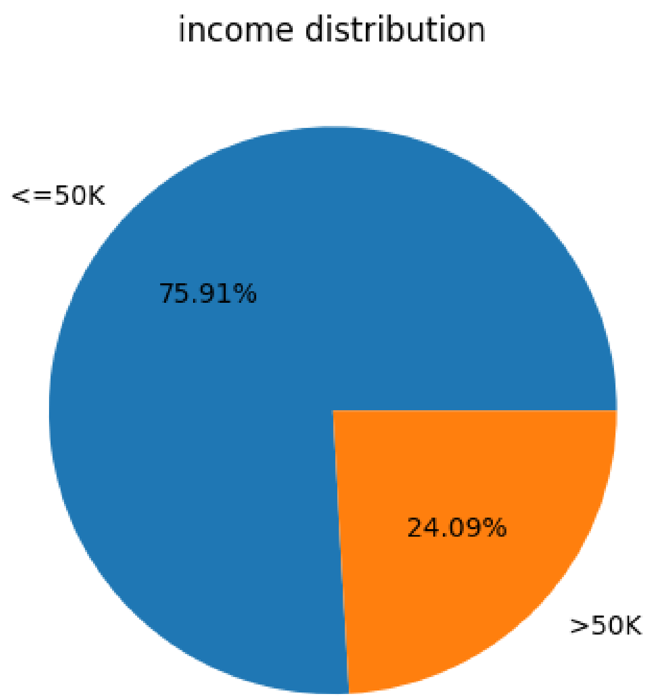
[11]:
```
df.head()
```

[11]:
```
   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
```

[87]:

```
## distribution of our terget varibale -> income
income = df['income'].value_counts()

plt.pie(income,labels=income.index,autopct="%1.2f%%")
plt.title("income distribution")
plt.show()
```

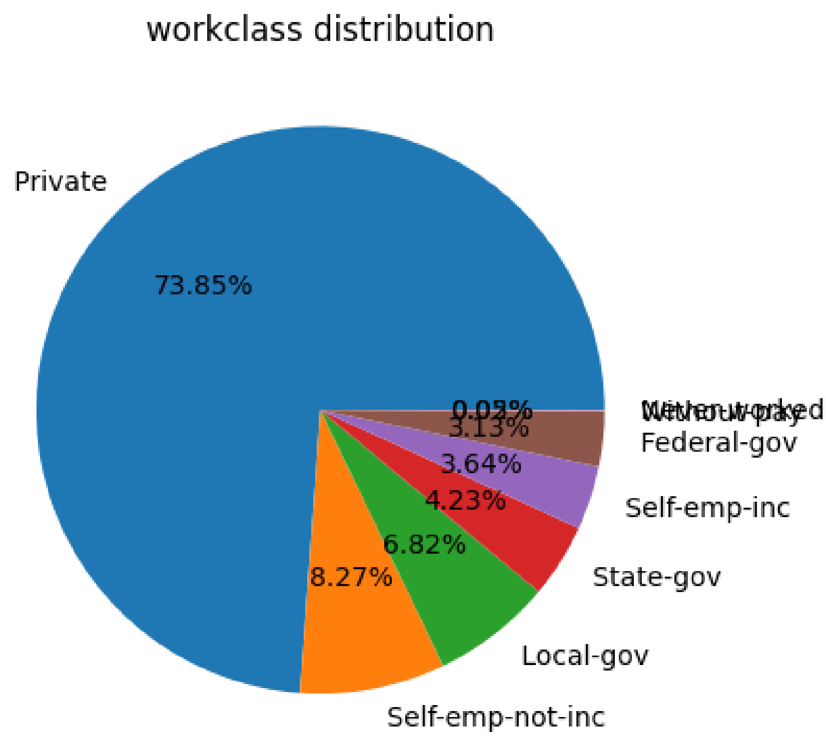


1.4 Observation:

People with <=50K income: 75.91%

People with >50K income: 24.09%

```
[88]: ## Distribution of workclass column
temp = df['workclass'].value_counts()

plt.pie(temp,labels=temp.index,autopct="%1.2f%%")
plt.title("workclass distribution")
plt.show()
```

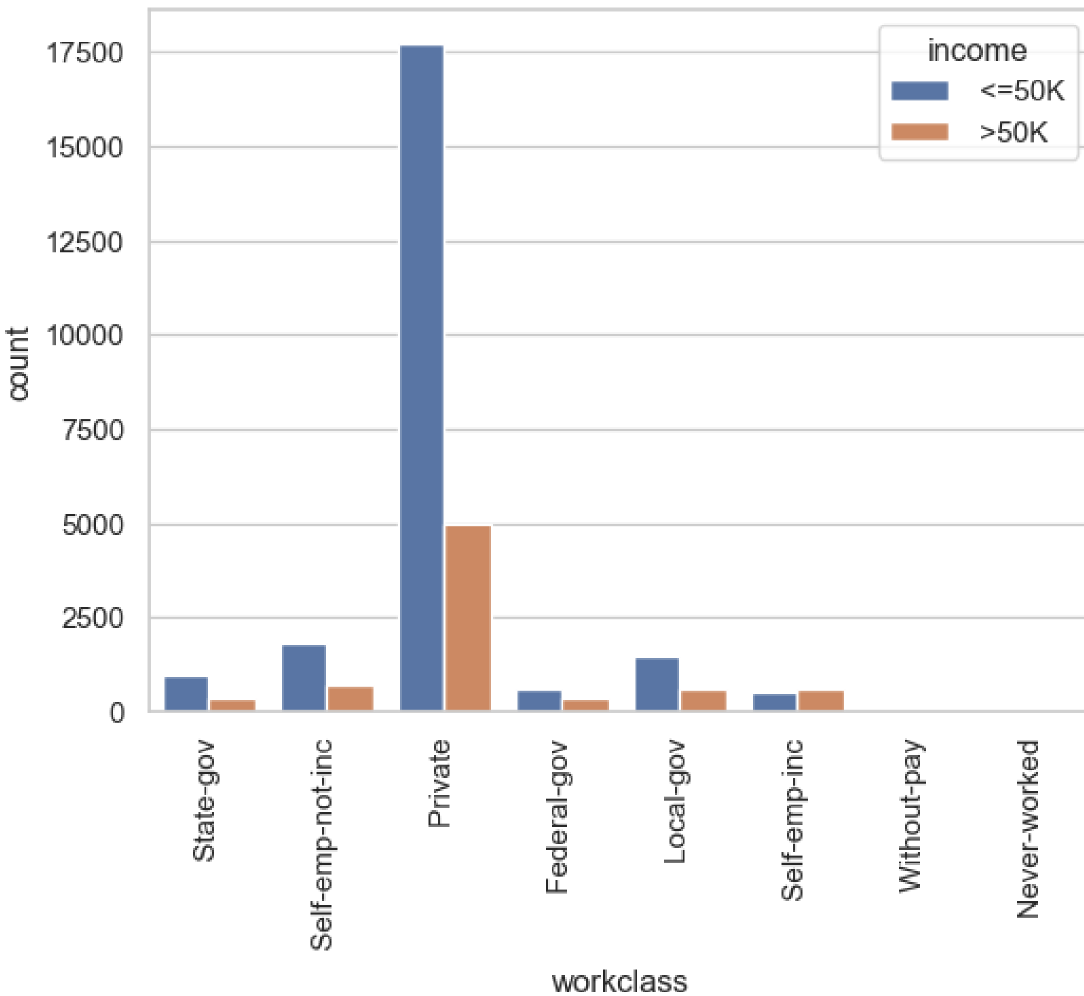



```
[14]: ## Relationship between workclass and income

workclass = df.groupby('workclass')['income']

workclass.value_counts()
```

```
[14]: workclass         income
       Federal-gov       <=50K       589
                         >50K        371
       Local-gov         <=50K      1476
                         >50K        617
       Never-worked      <=50K         7
       Private           <=50K     17712
                         >50K       4961
       Self-emp-inc      >50K        622
```

```
                    <=50K       494
 Self-emp-not-inc   <=50K      1816
                    >50K        724
 State-gov          <=50K       945
                    >50K        353
 Without-pay        <=50K        14
Name: income, dtype: int64
```

[104]:
```python
sns.countplot(df,x='workclass',hue='income')
plt.xticks(rotation=90)
plt.show()
```

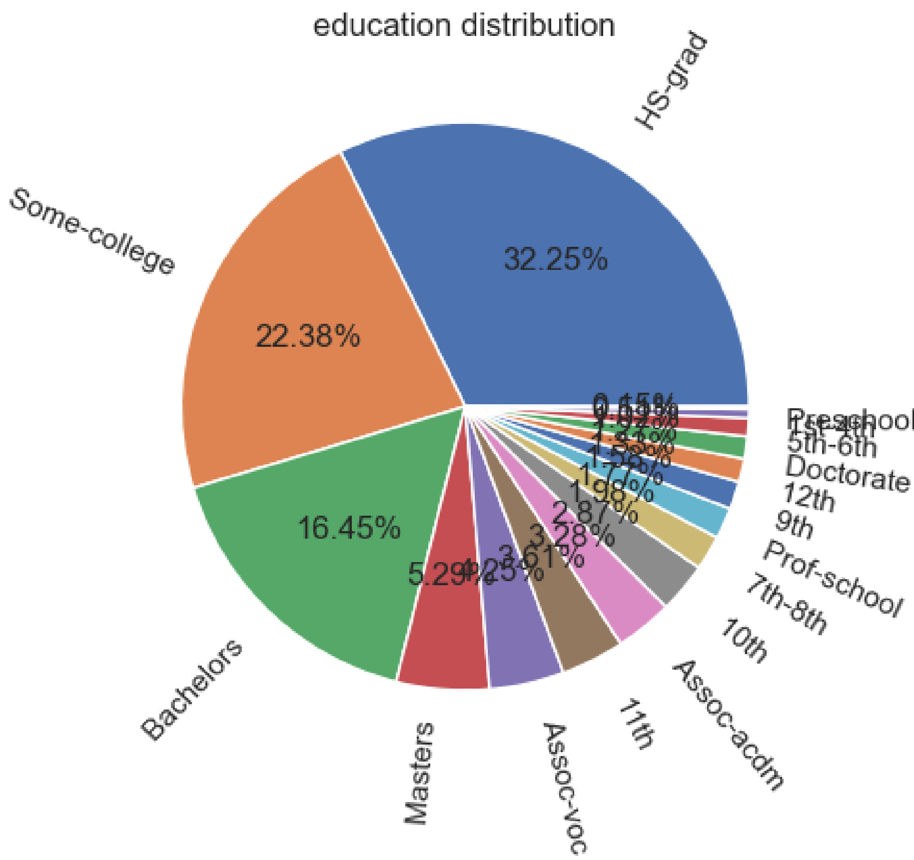


1.5 Observation:

1. Most people work in the Private sector
2. In every sector (except self-emp-inc), the number of people who earns <=50K are more than

the number of people who earns >50K

[106]:
```
## distribution of education feature

education = df['education'].value_counts()

plt.pie(education,labels=education.index,autopct="%1.2f%%",rotatelabels=True)
plt.title("education distribution")
plt.show()
```



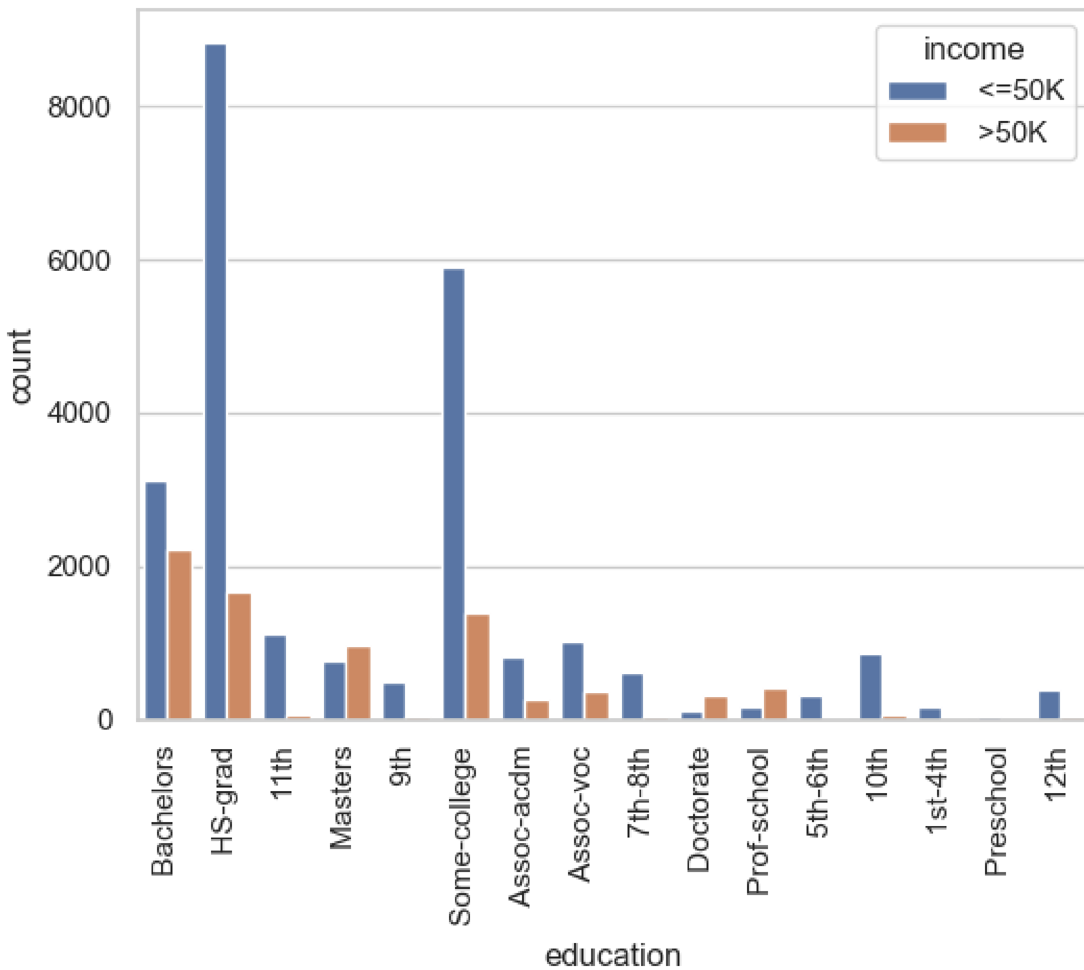
[17]:
```
## Relationship between education and income feature

edu_income = df.groupby('education')['income']
edu_income.value_counts()
```

[17]:
```
education      income
 10th           <=50K     871
                >50K       62
 11th           <=50K    1115
                >50K       60
```

```
 12th          <=50K     400
               >50K       33
 1st-4th       <=50K     160
               >50K        6
 5th-6th       <=50K     316
               >50K       16
 7th-8th       <=50K     605
               >50K       40
 9th           <=50K     487
               >50K       27
 Assoc-acdm    <=50K     802
               >50K      265
 Assoc-voc     <=50K    1021
               >50K      361
 Bachelors     <=50K    3132
               >50K     2221
 Doctorate     >50K      306
               <=50K     107
 HS-grad       <=50K    8820
               >50K     1674
 Masters       >50K      959
               <=50K     763
 Preschool     <=50K      50
 Prof-school   >50K      423
               <=50K     153
 Some-college  <=50K    5896
               >50K     1386
Name: income, dtype: int64
```

[107]:

```
sns.countplot(df,x='education',hue='income')
plt.xticks(rotation=90)
plt.show()
```

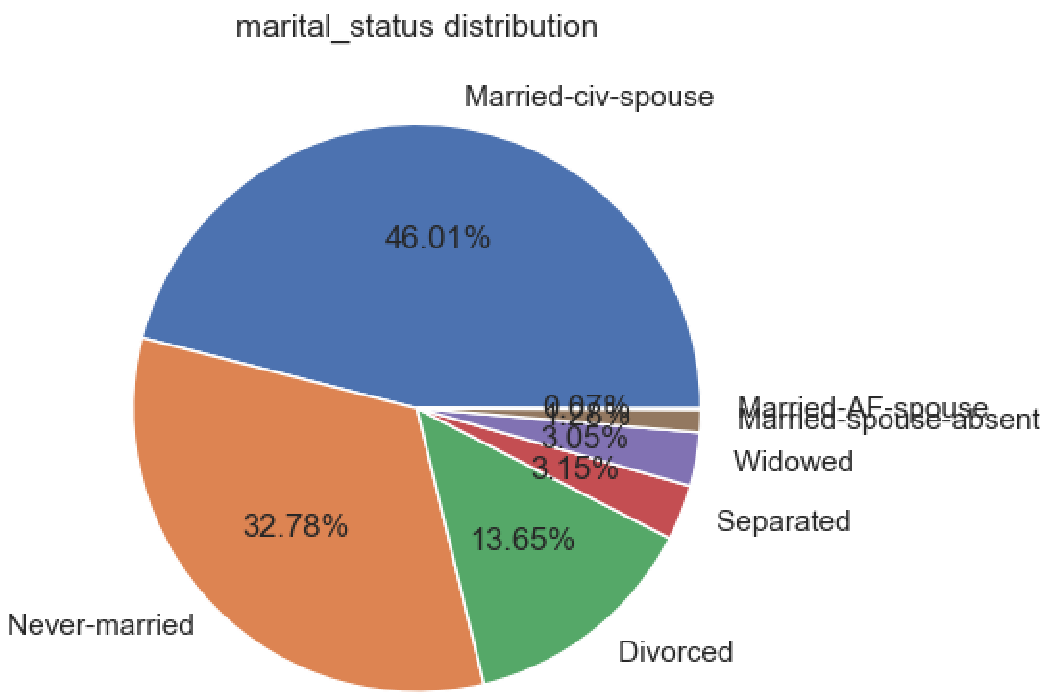


1.6 Observation:

1. In Bachelors, HS-grad, Masters, Doctorate, Prof-school there are more people who are earning money more than 50K.
2. In Master, doctorate and Prof-school - the number of people with income >50K is greater than the number of people with income <=50K

[109]:

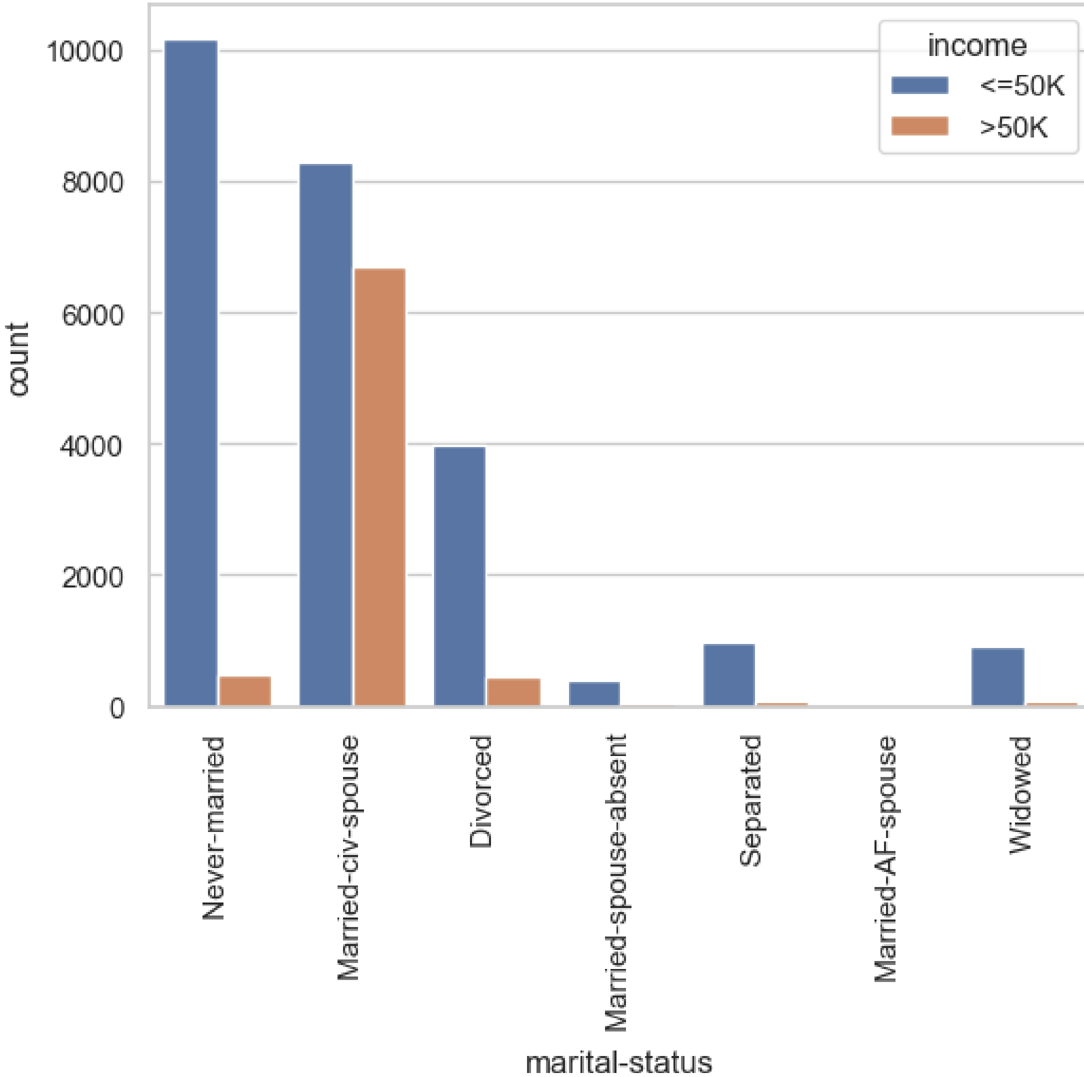
```
# distribution of Marital status feature

marital_status = df['marital-status'].value_counts()

plt.pie(marital_status,labels=marital_status.index,autopct="%1.2f%%")
plt.title("marital_status distribution")
plt.show()
```



[20]:
```
## Relationship between marital-status and income feature
marital_status_income = df.groupby('marital-status')['income']
marital_status_income.value_counts()
```

[20]:
```
marital-status          income
 Divorced                <=50K     3978
                         >50K       463
 Married-AF-spouse       <=50K       13
                         >50K        10
 Married-civ-spouse      <=50K     8280
                         >50K      6690
 Married-spouse-absent   <=50K      384
                         >50K        34
 Never-married           <=50K    10176
                         >50K       491
 Separated               <=50K      959
                         >50K        66
 Widowed                 <=50K      908
                         >50K        85
Name: income, dtype: int64
```

[110]:
```
sns.countplot(df,x='marital-status',hue='income')
plt.xticks(rotation=90)
plt.show()
```

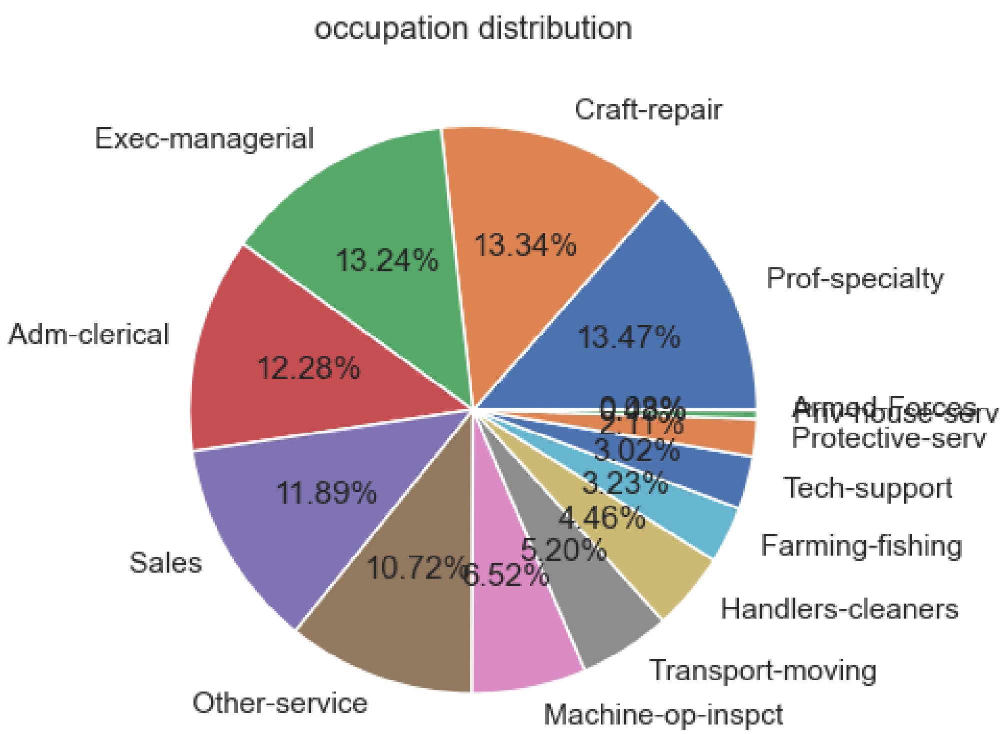


1.7 Observation:

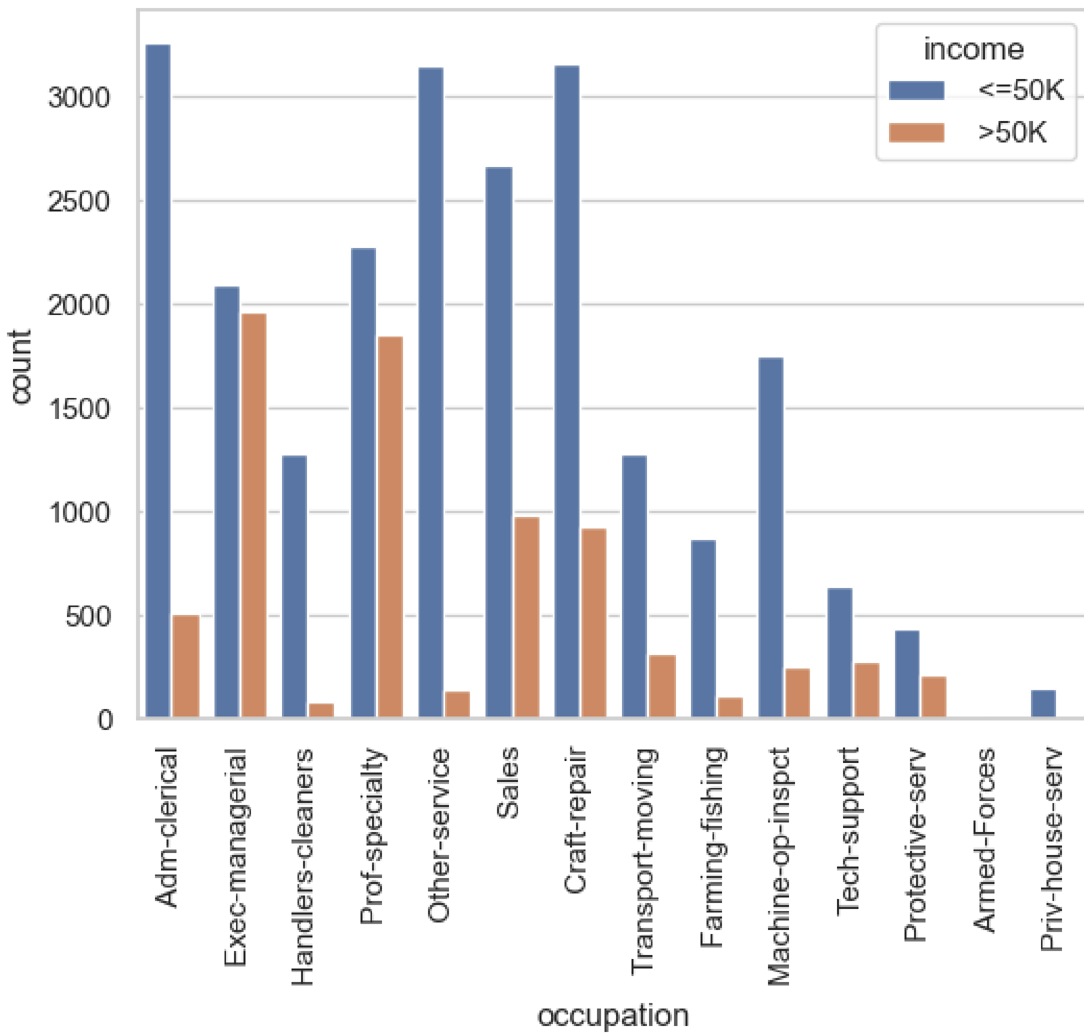
1. Most of the people earning >50K are Married-civ-spouse
2. Most of the people earning <=50K are Never-married
3. The differnece between two income groups in Never-married column is very high.

[111]:
```
## Distribution of Occupation feature

occupation = df['occupation'].value_counts()
```

```
plt.pie(occupation,labels=occupation.index,autopct="%1.2f%%")
plt.title("occupation distribution")
plt.show()
```



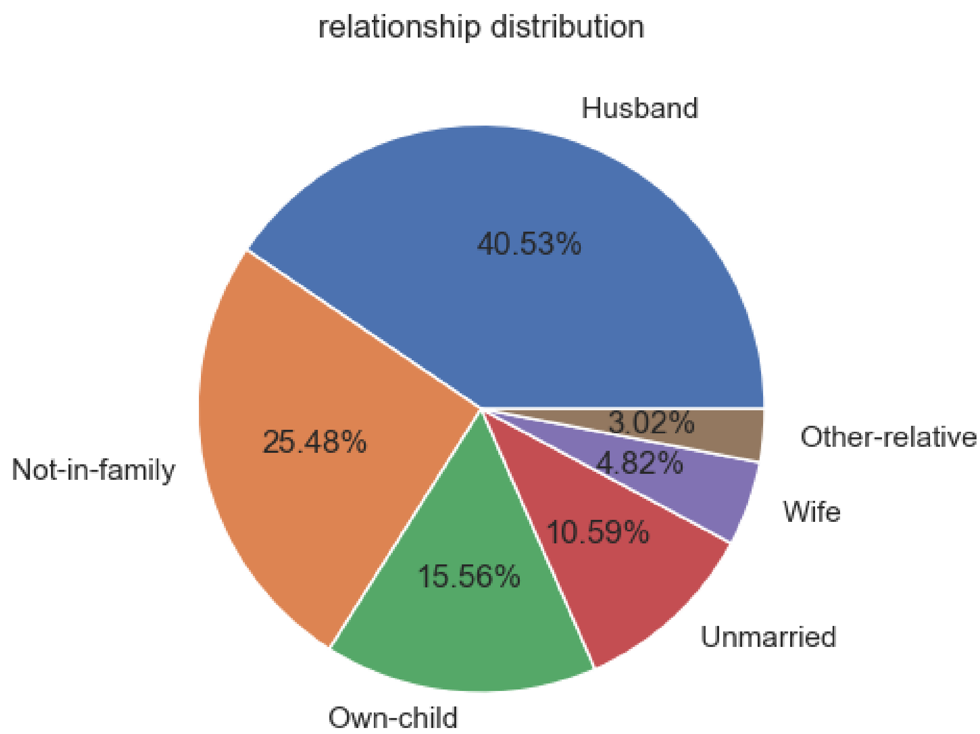
[112]:
```
# relationship between occupation and income feature
sns.countplot(df,x='occupation',hue='income')
plt.xticks(rotation=90)
plt.show()
```



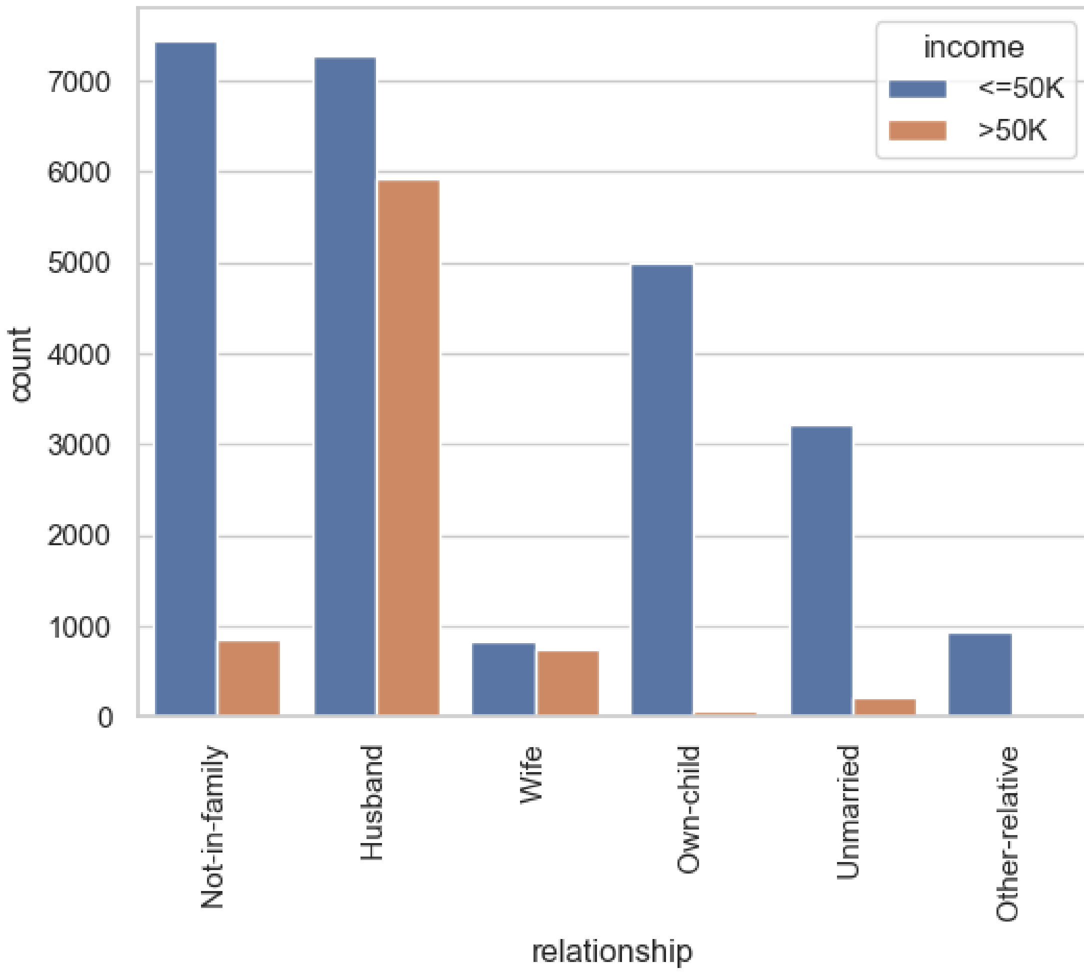
1.8 Observation:

1. More People are earning >50K in Exec-managerial and Prof-speciality than other groups.

[113]:
```
## Distribution of relationship feature

relationship = df['relationship'].value_counts()

plt.pie(relationship,labels=relationship.index,autopct="%1.2f%%")
plt.title("relationship distribution")
plt.show()
```



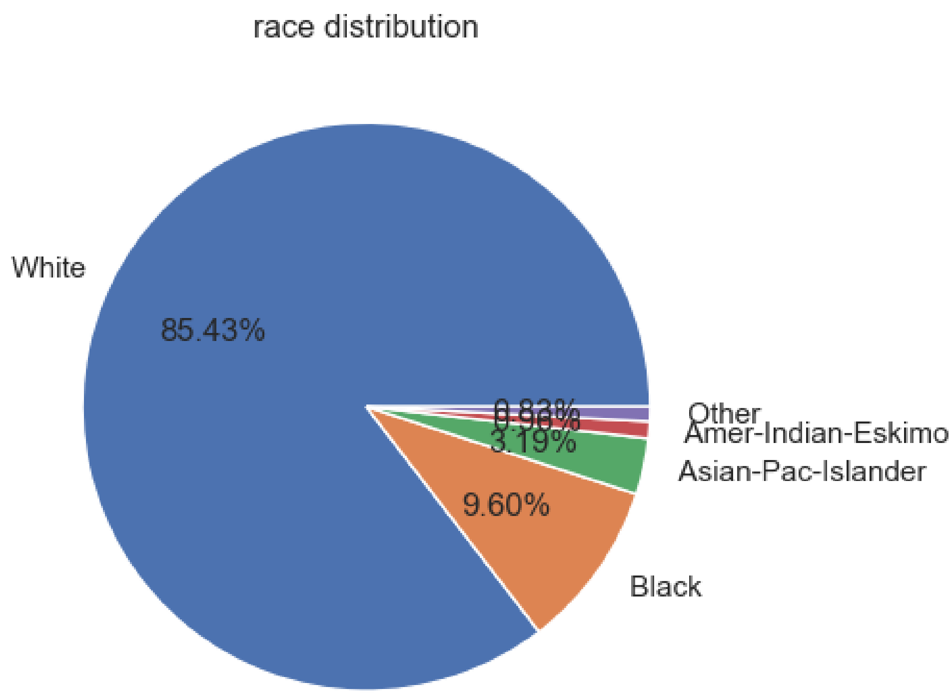
[114]:
```
# relationship between relationship and income feature
sns.countplot(df,x='relationship',hue='income')
plt.xticks(rotation=90)
plt.show()
```



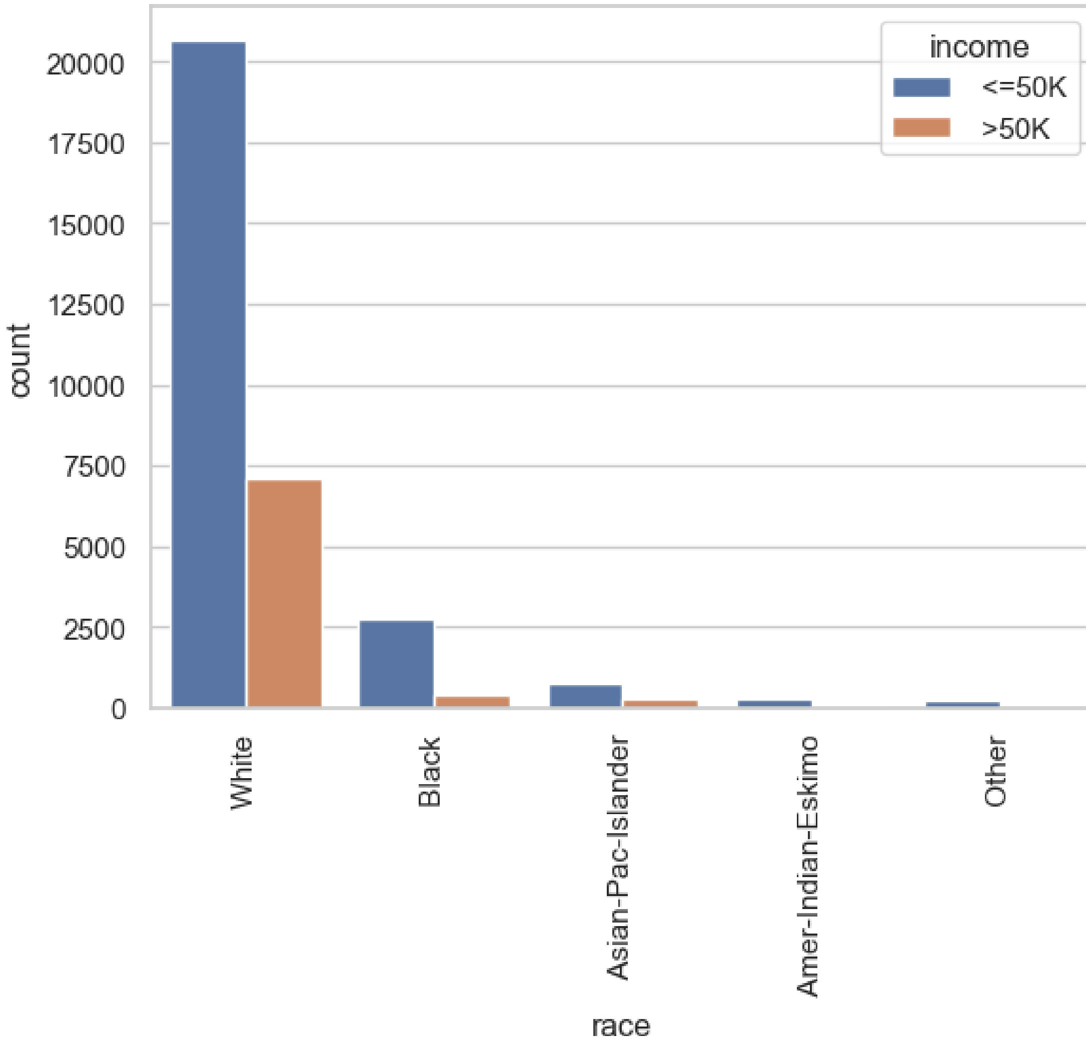
1.9 Observation:

1. In relationship column, 40.5% are husband.
2. Husbands are more likely to earn >50K than others.

[115]:
```python
## Distribution of race feature

race = df['race'].value_counts()

plt.pie(race,labels=race.index,autopct="%1.2f%%")
plt.title("race distribution")
plt.show()
```



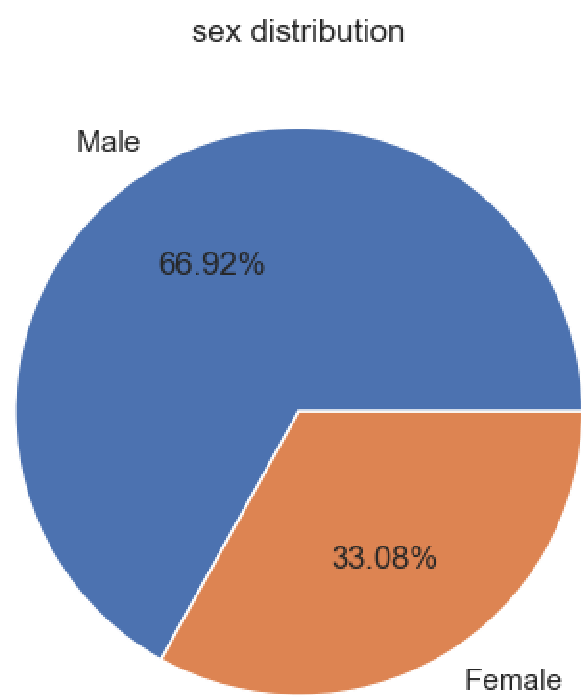
[116]:
```
# relationship between race and income feature
sns.countplot(df,x='race',hue='income')
plt.xticks(rotation=90)
plt.show()
```



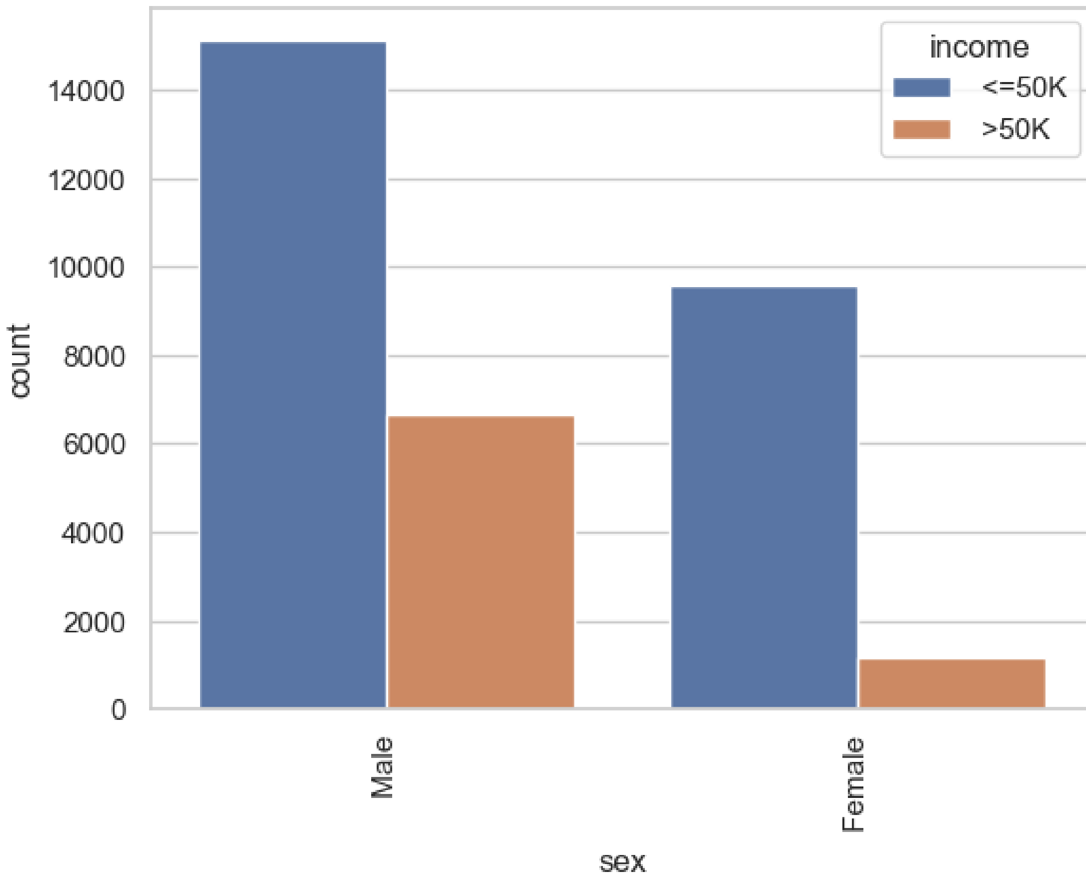
1.10 Observation:

1. in race column, maximum people are White.
2. White people are more likely to earn income of >50K.

[117]:
```
## Distribution of sex feature

sex = df['sex'].value_counts()

plt.pie(sex,labels=sex.index,autopct="%1.2f%%")
plt.title("sex distribution")
plt.show()
```



[118]:
```
# relationship between sex and income feature
sns.countplot(df,x='sex',hue='income')
plt.xticks(rotation=90)
plt.show()
```



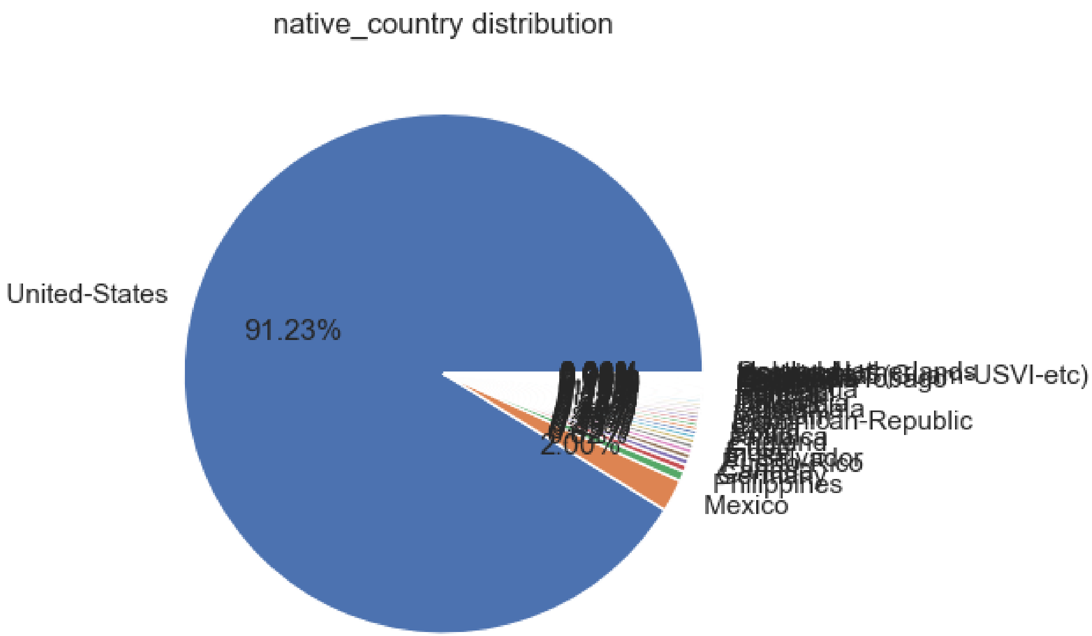
1.11 Observation:

1. More male(66.9%) than female(33.1%) in sex column
2. Males are more likely to earn >50K than females.

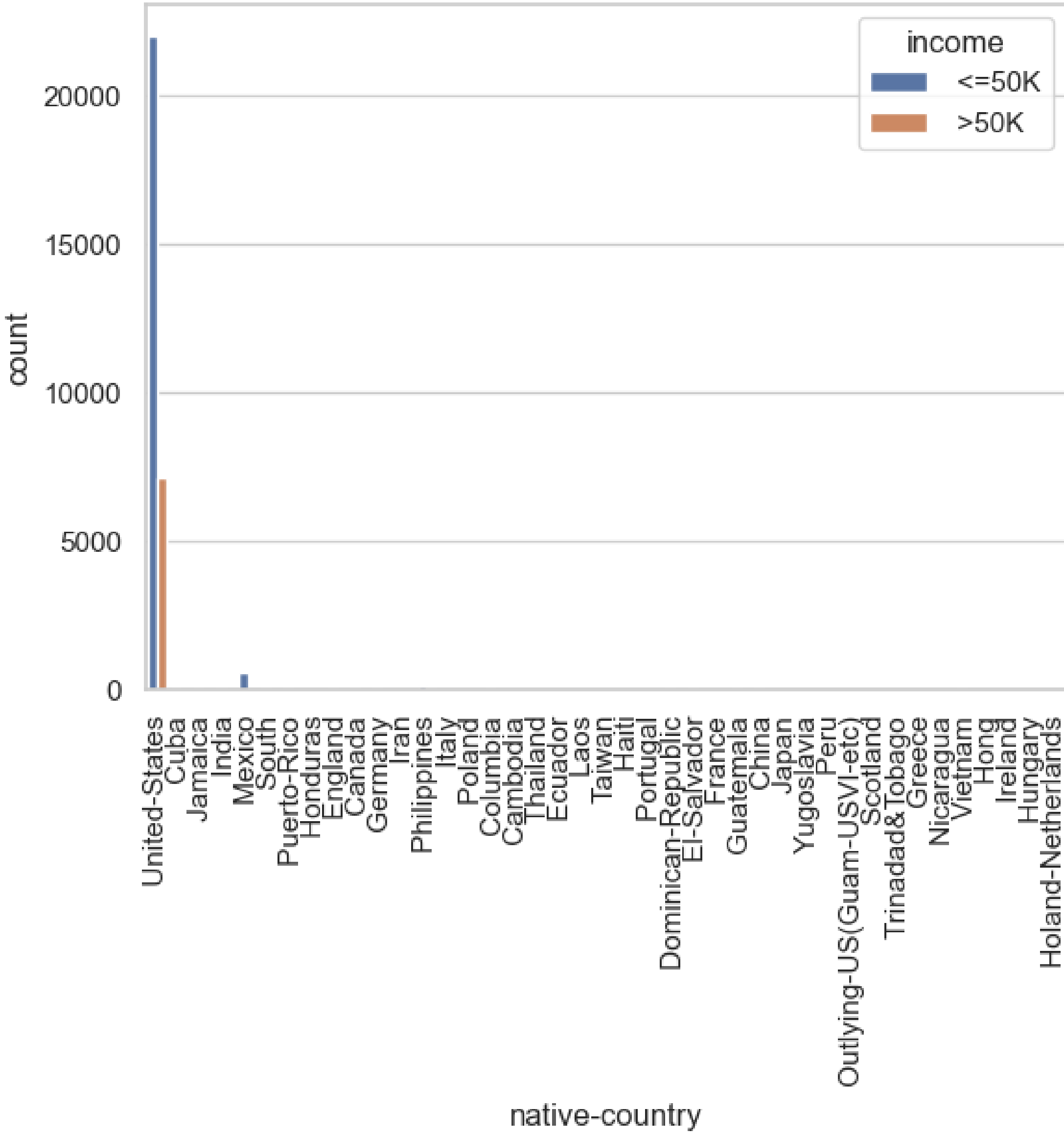
[31]:
```
## unique values in native-country feature

df['native-country'].nunique()
```

[31]: 41

[119]:
```
## Distribution of native-country feature

native_country = df['native-country'].value_counts()

plt.pie(native_country,labels=native_country.index,autopct="%1.2f%%")
plt.title("native_country distribution")
plt.show()
```



[120]:
```
# relationship between native-country and income feature
sns.countplot(df,x='native-country',hue='income')
plt.xticks(rotation=90)
plt.show()
```



1.12 Observation:

1. Total 41 unique countries are present.
2. Most datapoints(91.2%) are from united States.

1.13 Numerical features

[37]:
```
numerical_features = [feature for feature in df.columns if df[feature].dtype !=␣
↪'O']
```

[38]:
```
numerical_features
```

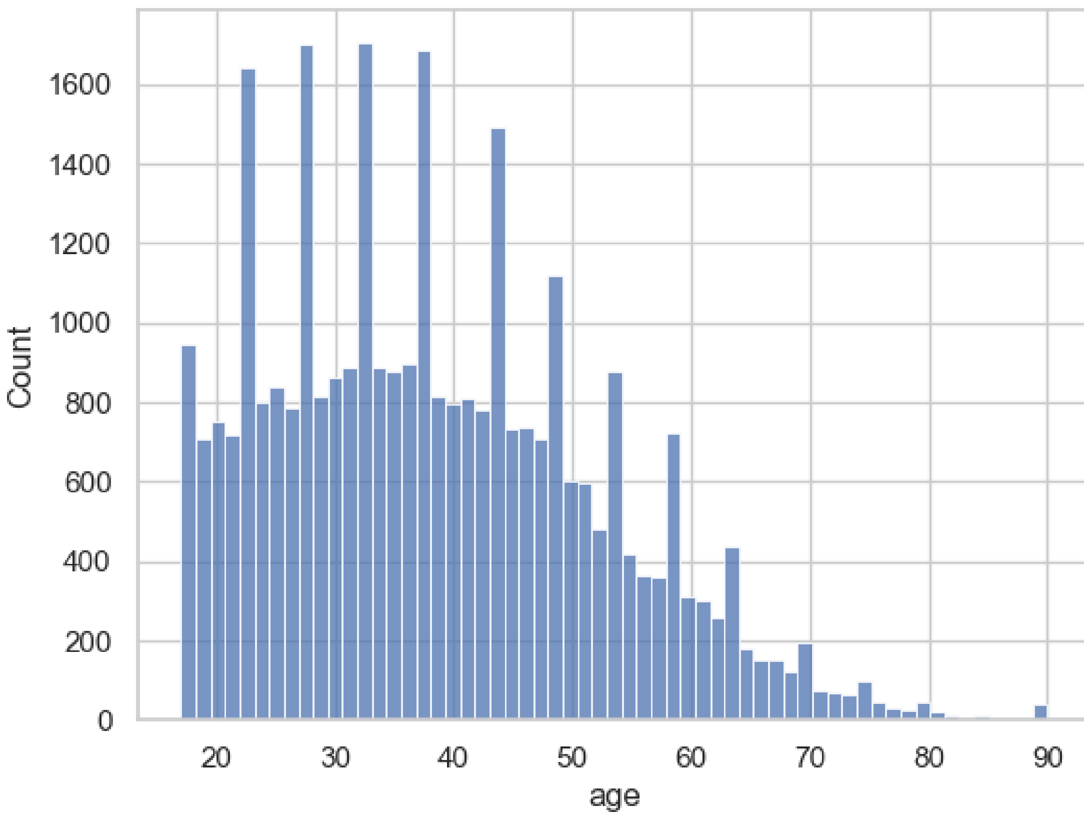
[38]:
```
['age',
 'fnlwgt',
 'education-num',
 'capital-gain',
 'capital-loss',
 'hours-per-week']
```

[39]:
```
df[numerical_features].head()
```

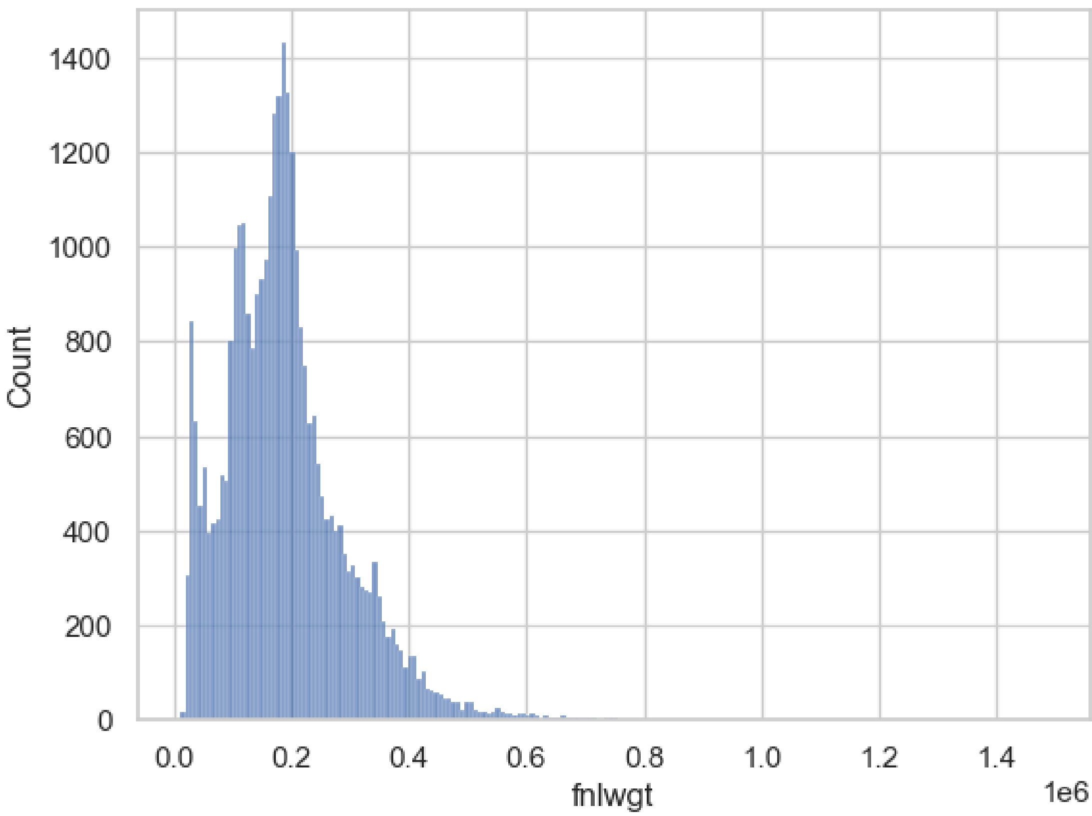
[39]:

	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

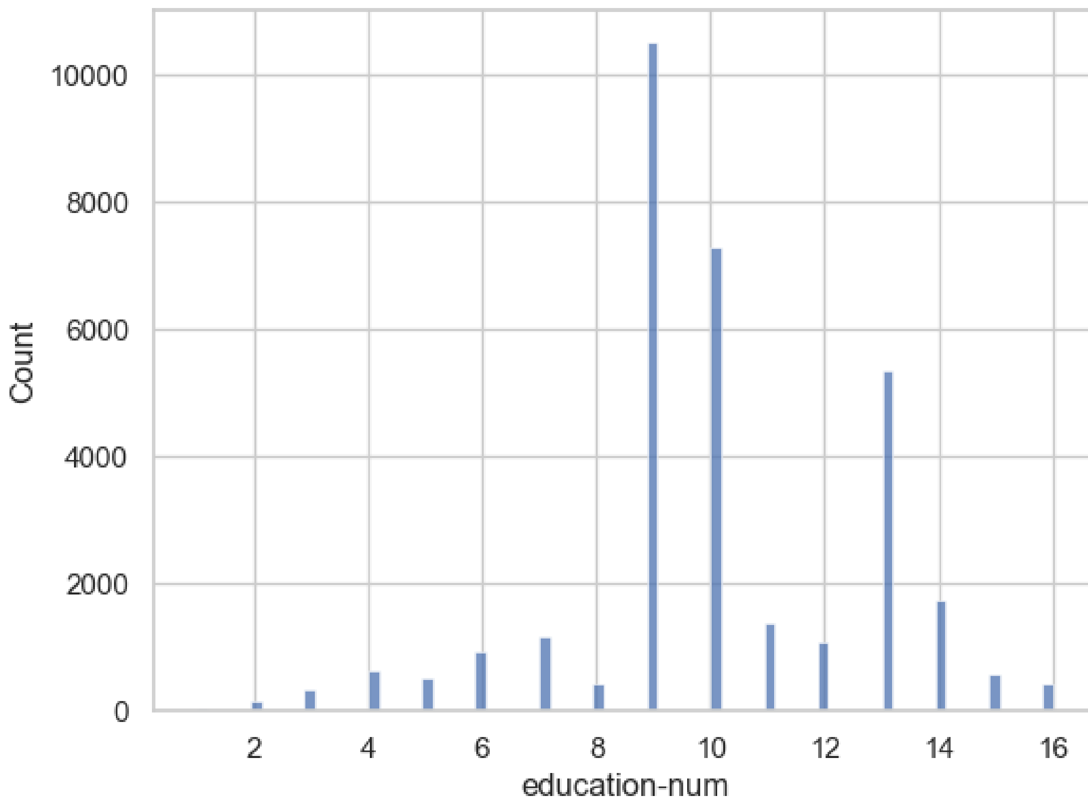
[123]:
```
## Distribution of numerical features
## Univariate analysis

for feature in numerical_features:
    sns.histplot(df,x=feature)
    plt.show()

plt.tight_layout()
```

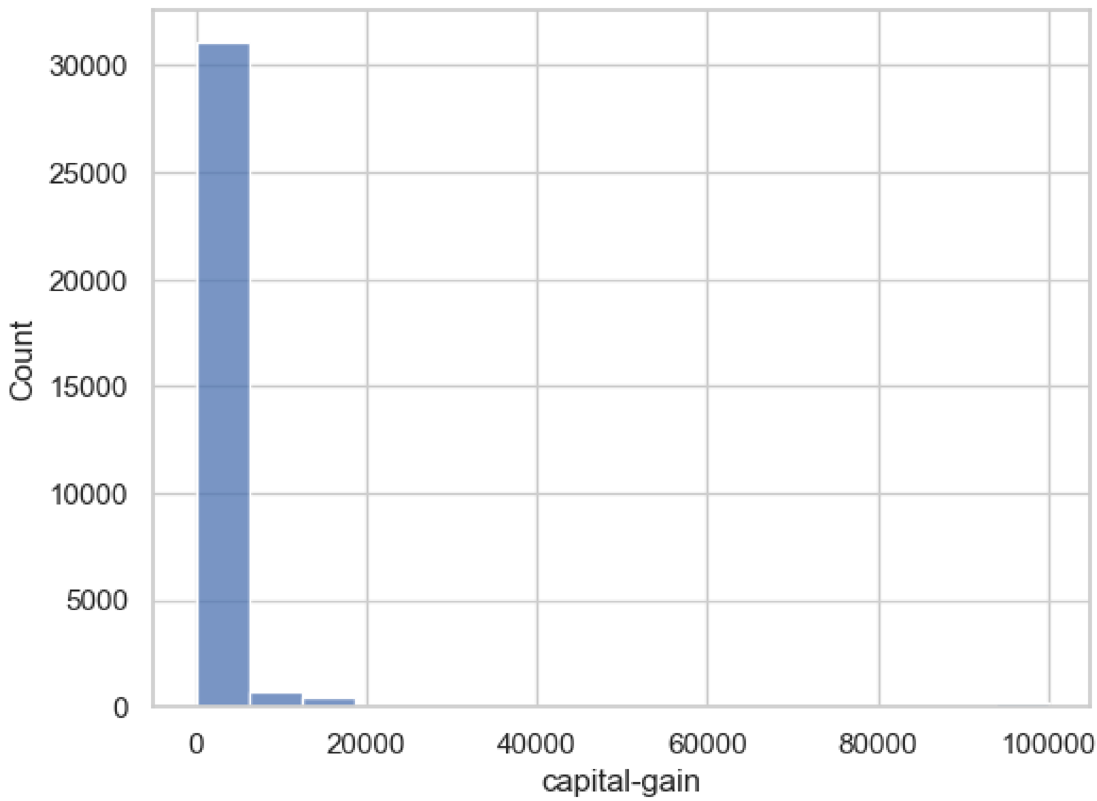


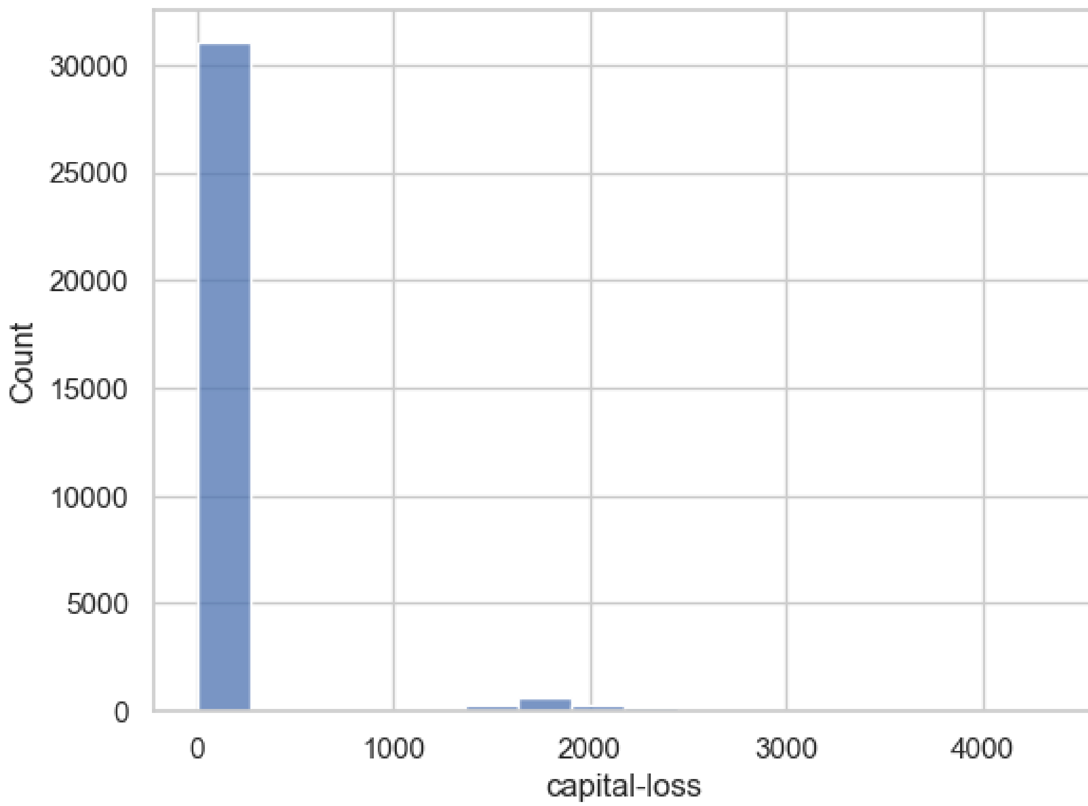
1600
1400
1200
1000
800
600
400
200
0
Count
20
30
40
50
60
70
80
90
age

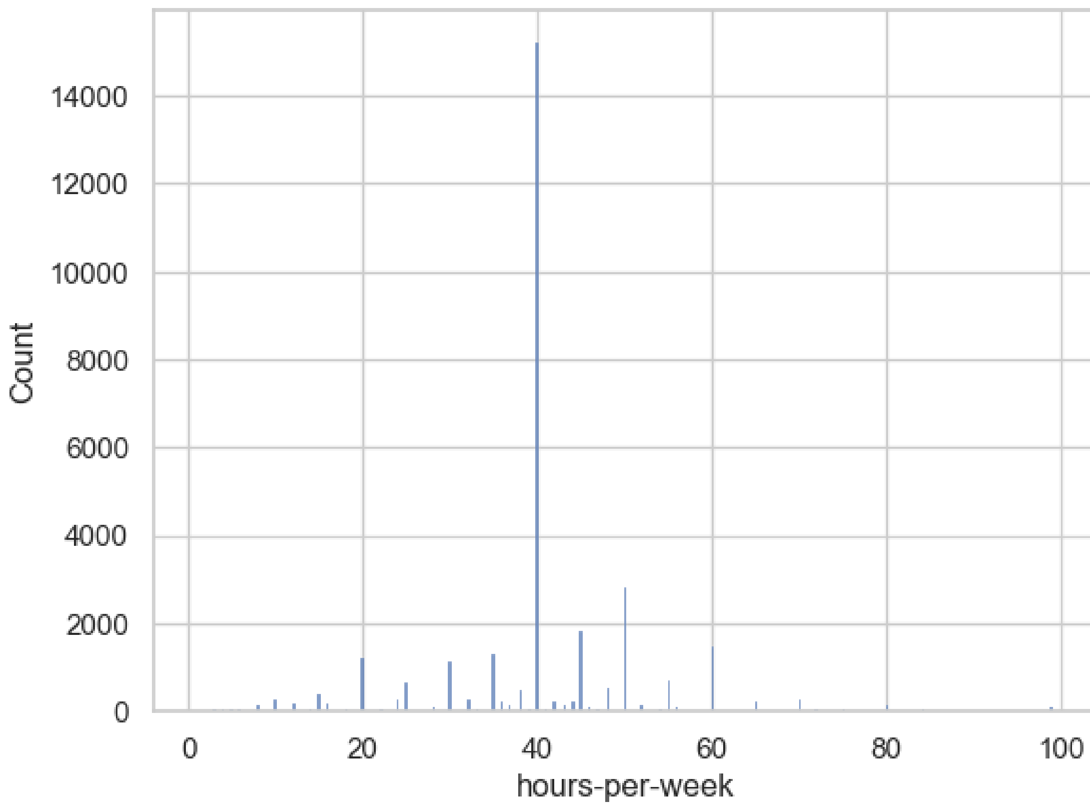


1400
1200
1000
800
600
400
200
0
Count
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
fnlwgt


Count
10000
8000
6000
4000
2000
0
2
4
6
8
10
12
14
16
education-num


30000
25000
20000
15000
10000
5000
0
Count
0
20000
40000
60000
80000
100000
capital-gain


Count
30000
25000
20000
15000
10000
5000
0
0
1000
2000
3000
4000
capital-loss

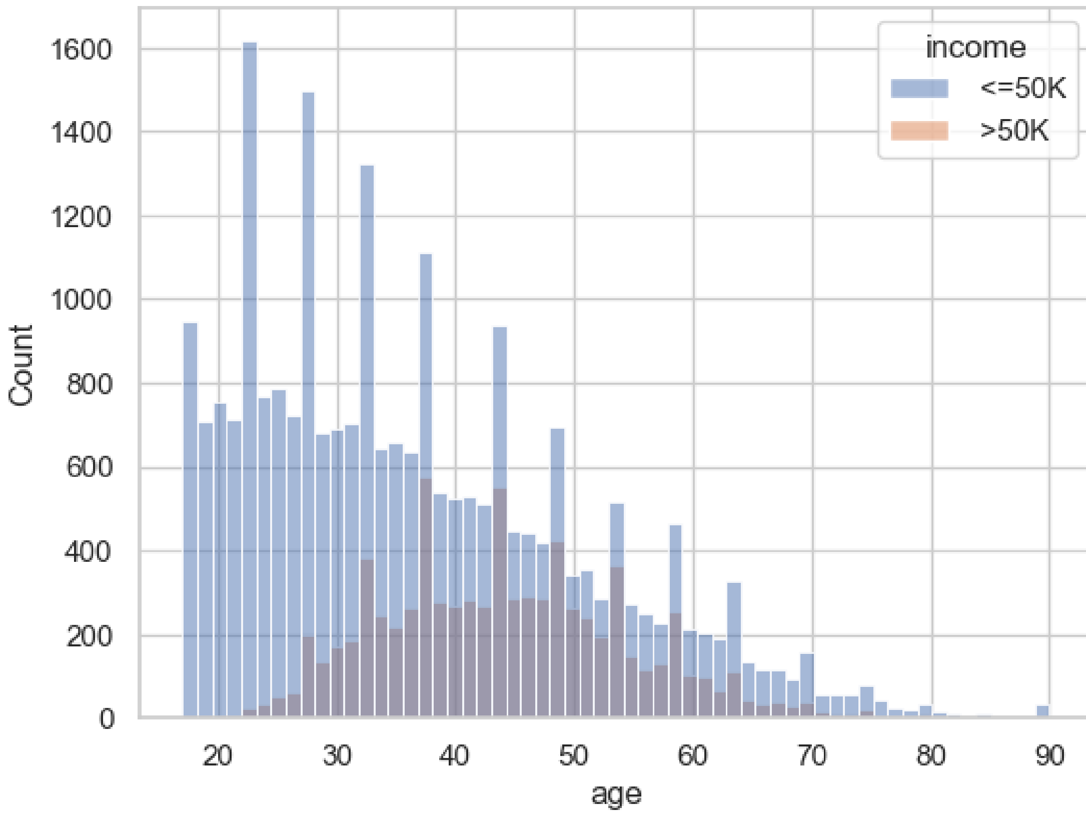


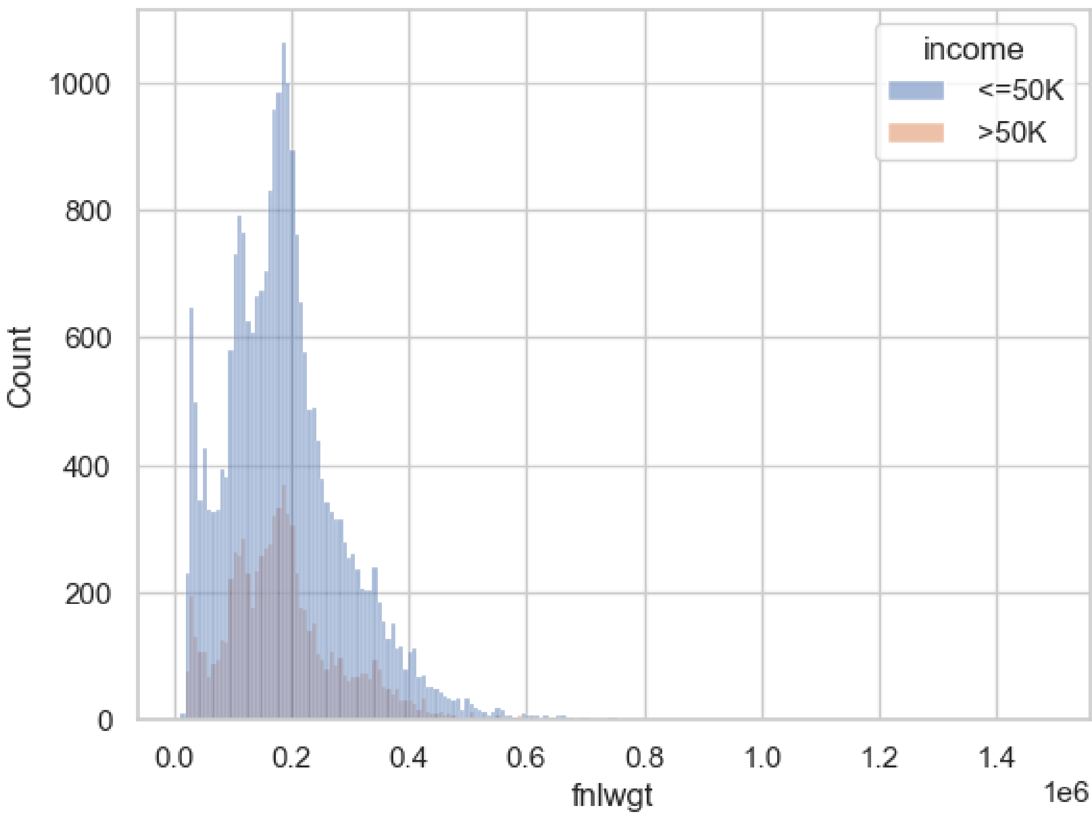

```
<Figure size 640x480 with 0 Axes>
```

1.14 Observation:

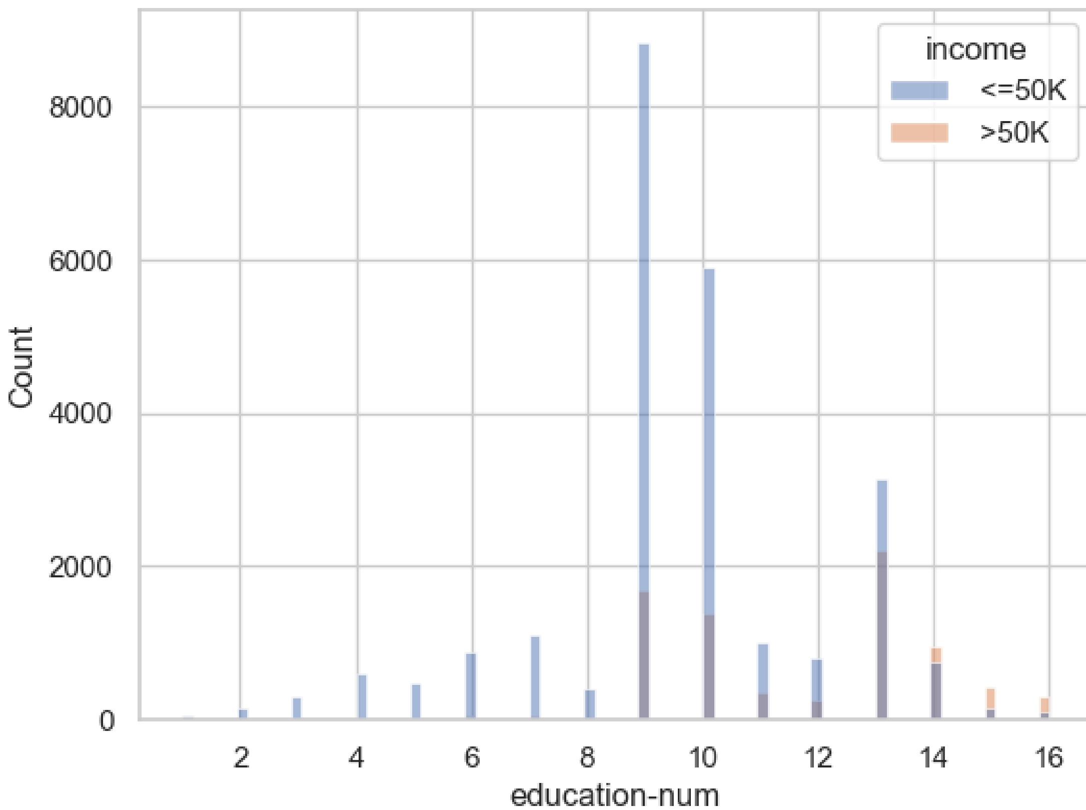
1. The age column is sightly right-skewed or postively skewed.
2. Capital gain and capital loss are mostly 0
3. In 'hours-per-week' column, most datapoints are concentrated on 40.

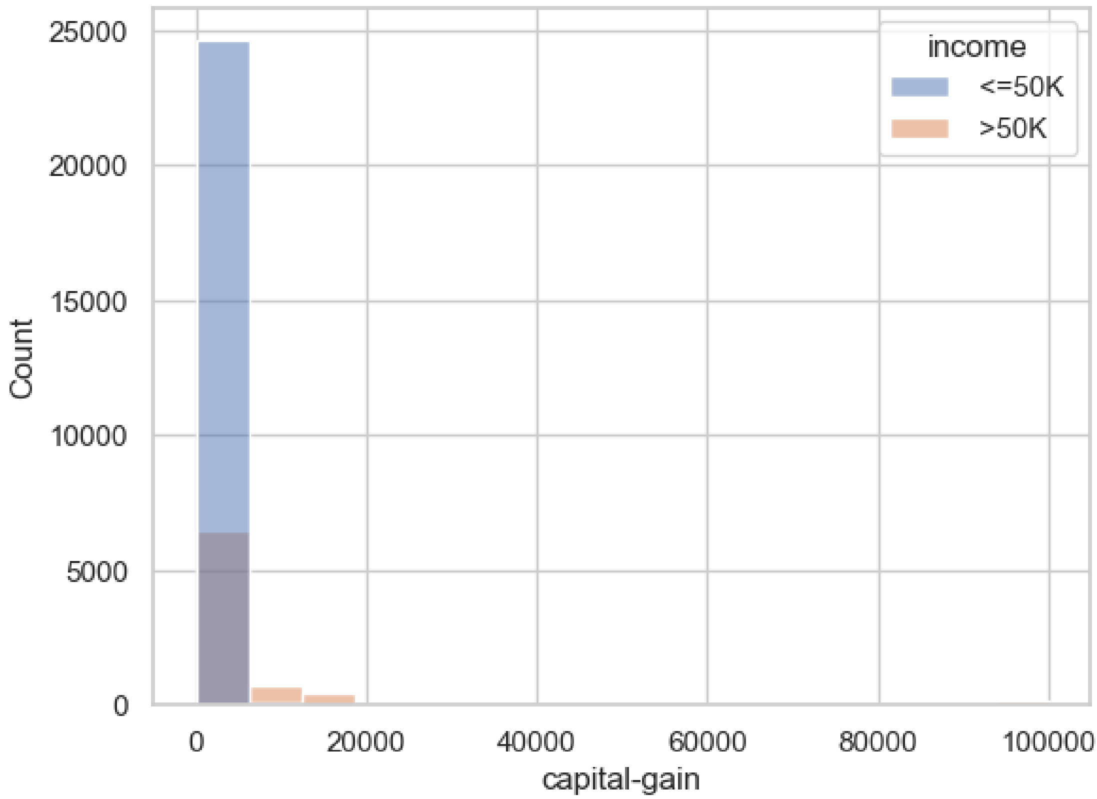
[124]:
```
for feature in numerical_features:
    sns.histplot(df,x=feature,hue='income')
    plt.show()

plt.tight_layout()
```

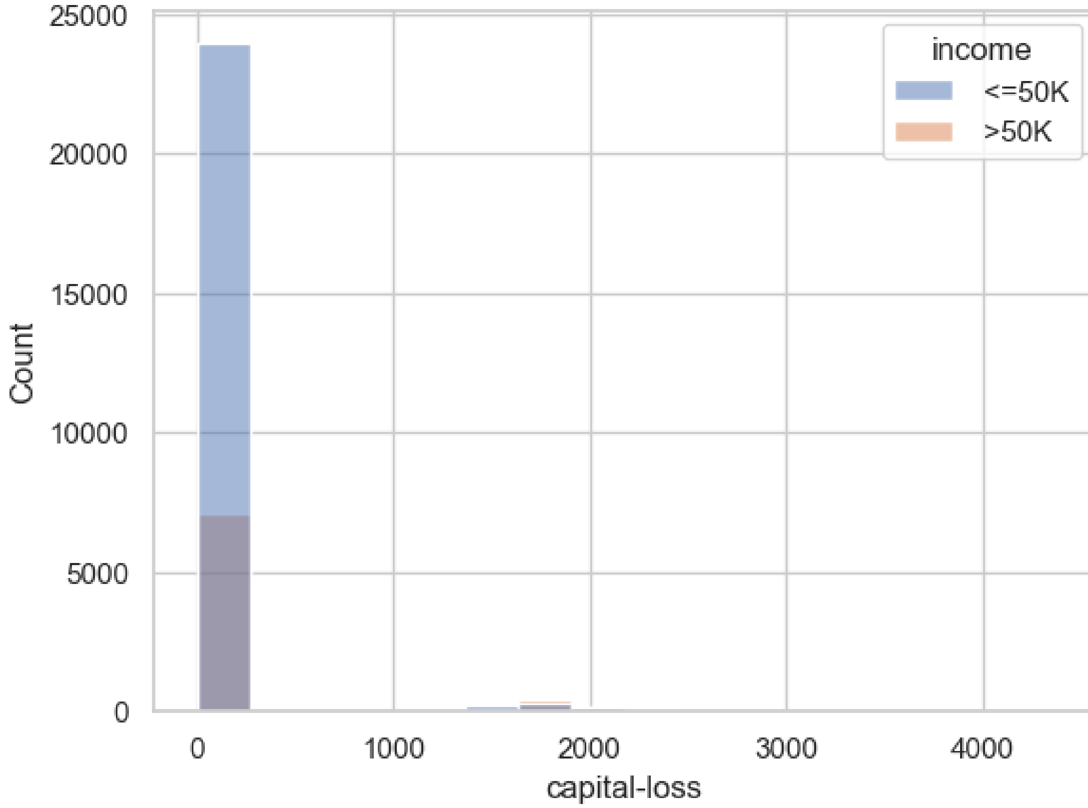

income
<=50K
>50K
Count
age
0
200
400
600
800
1000
1200
1400
1600
20
30
40
50
60
70
80
90

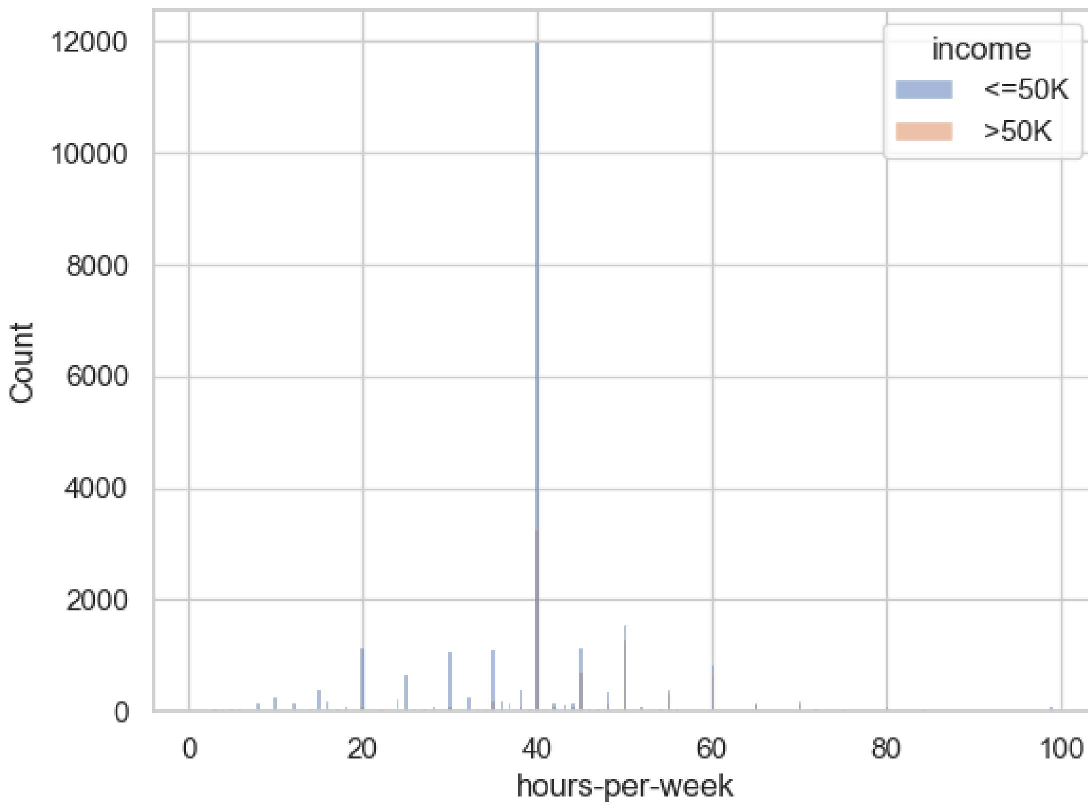


income
<=50K
>50K
Count
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
fnlwgt


income
<=50K
>50K
Count
8000
6000
4000
2000
0
2
4
6
8
10
12
14
16
education-num


25000
20000
15000
10000
5000
0
Count
0
20000
40000
60000
80000
100000
capital-gain
income
<=50K
>50K


income
<=50K
>50K
Count
25000
20000
15000
10000
5000
0
0
1000
2000
3000
4000
capital-loss




```
<Figure size 640x480 with 0 Axes>
```

1.15 Observation:

1. In 'age' column, most of the people earning >50K follows a different distribution than overall 'age' column distribution.
2. People with higher-education, are earning more income.
3. More the capital-gain, more income(>50K).

[68]:
```
s = pd.crosstab(df['hours-per-week'],df['income'],normalize="index")
```

[74]:
```
s
```

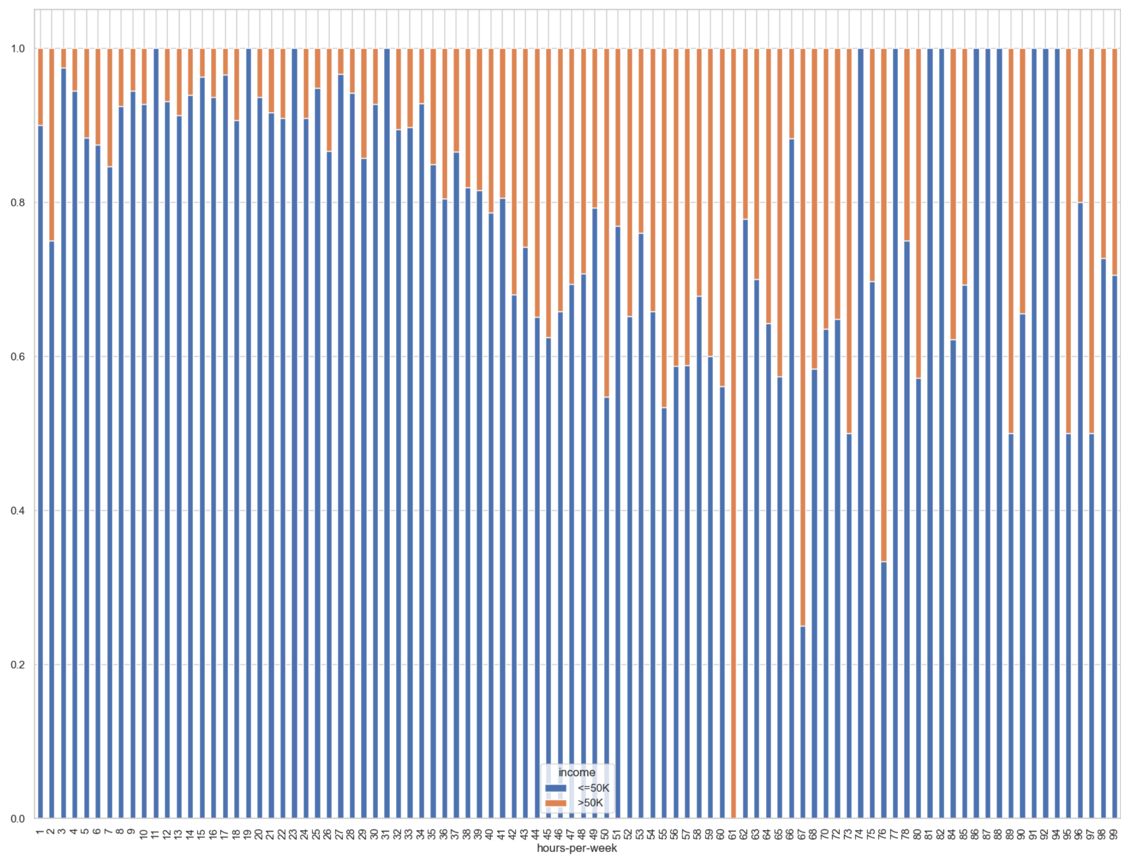
[74]:
```
income                 <=50K      >50K
hours-per-week
1                   0.900000  0.100000
2                   0.750000  0.250000
3                   0.974359  0.025641
4                   0.944444  0.055556
5                   0.883333  0.116667
…                        …         …
95                  0.500000  0.500000
```

```
96              0.800000  0.200000
97              0.500000  0.500000
98              0.727273  0.272727
99              0.705882  0.294118

[94 rows x 2 columns]
```

[71]:
```
fig= px.bar(s,color_discrete_sequence=['#c789f0','#f0927a'])
fig.show()
```

[134]:
```
s.plot(kind='bar',stacked=True, figsize=(20,15))
```

[134]:
```
<AxesSubplot: xlabel='hours-per-week'>
```



1.16 Observation:

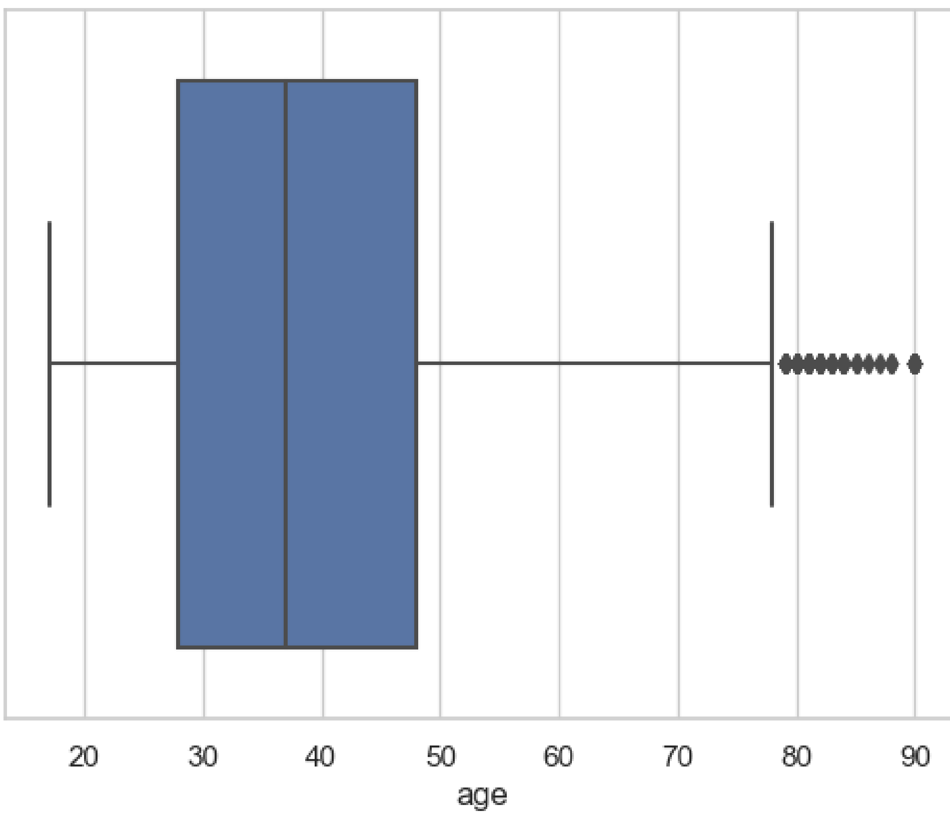
1. Longer working hours does not mean higher income.

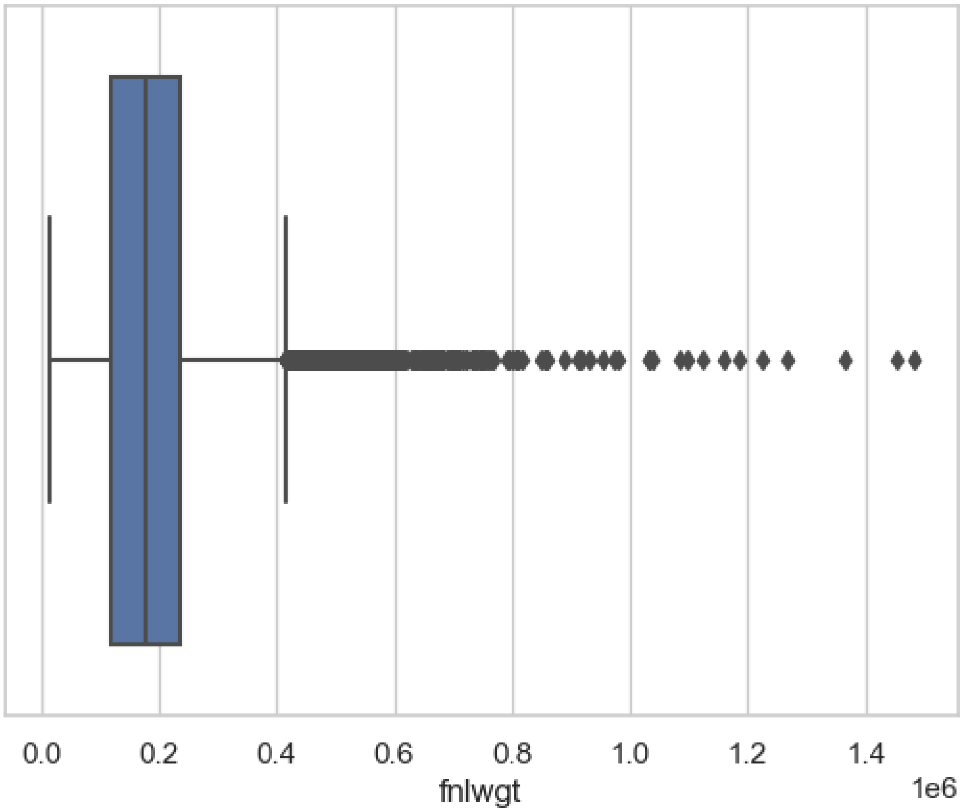
1.17 Outliers

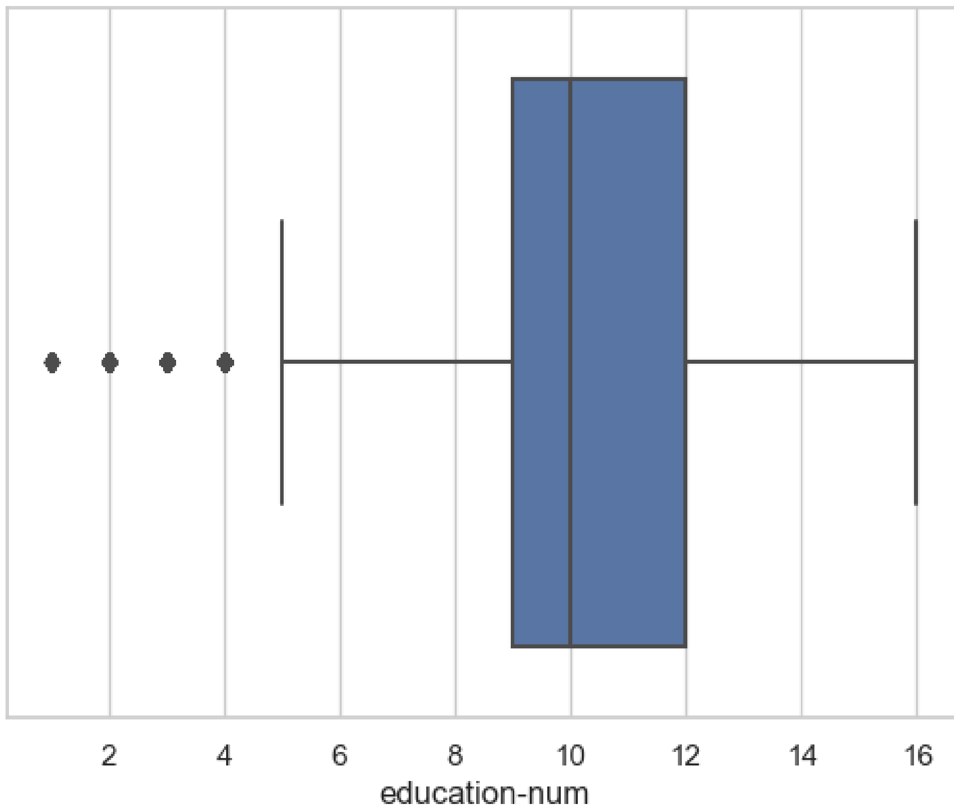
[135]:
```
numerical_features
```

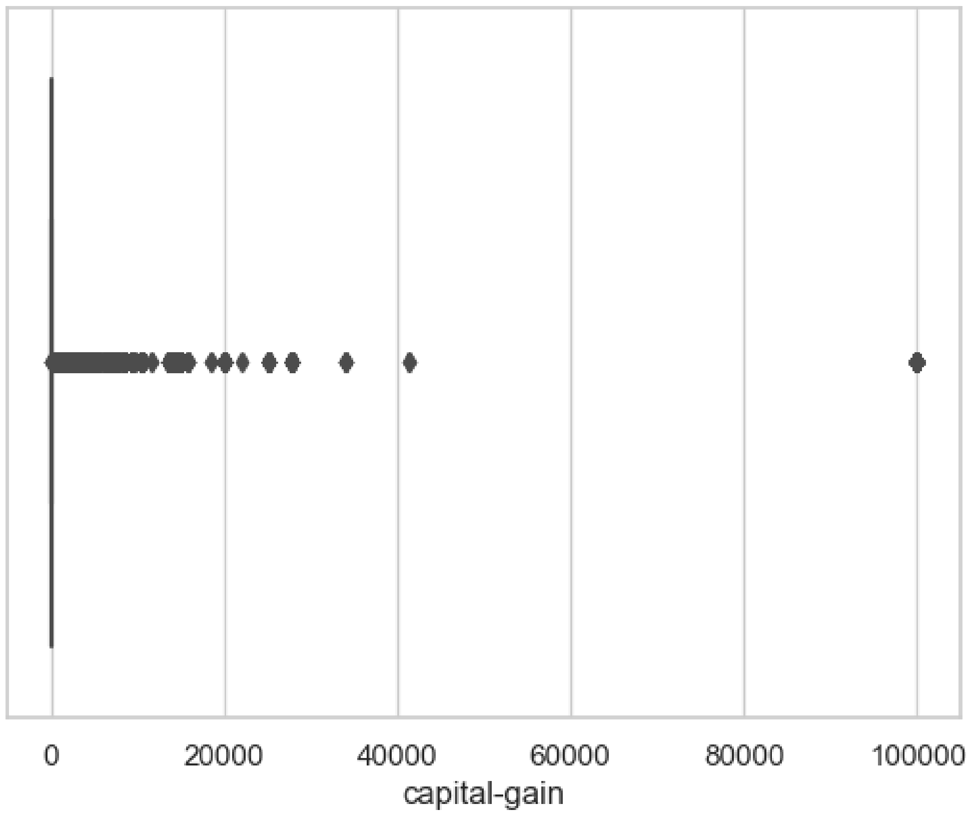
[135]:
```
['age',
 'fnlwgt',
 'education-num',
 'capital-gain',
 'capital-loss',
 'hours-per-week']
```

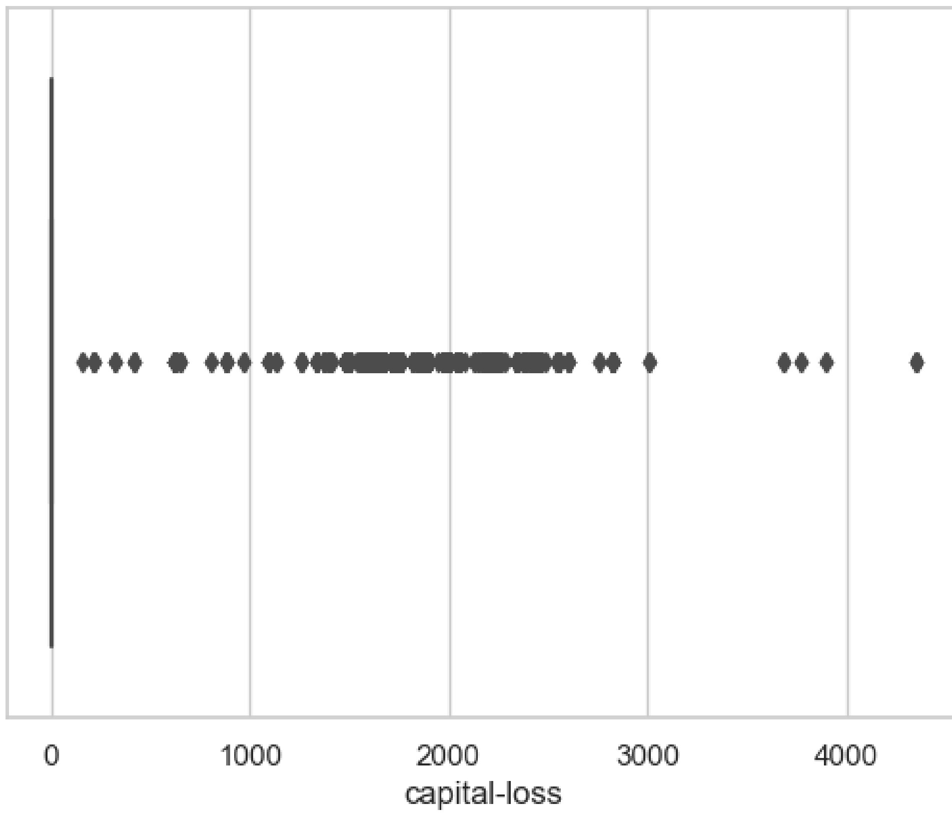
[139]:
```
for feature in numerical_features:
    sns.boxplot(x=df[feature])
    plt.show()
```

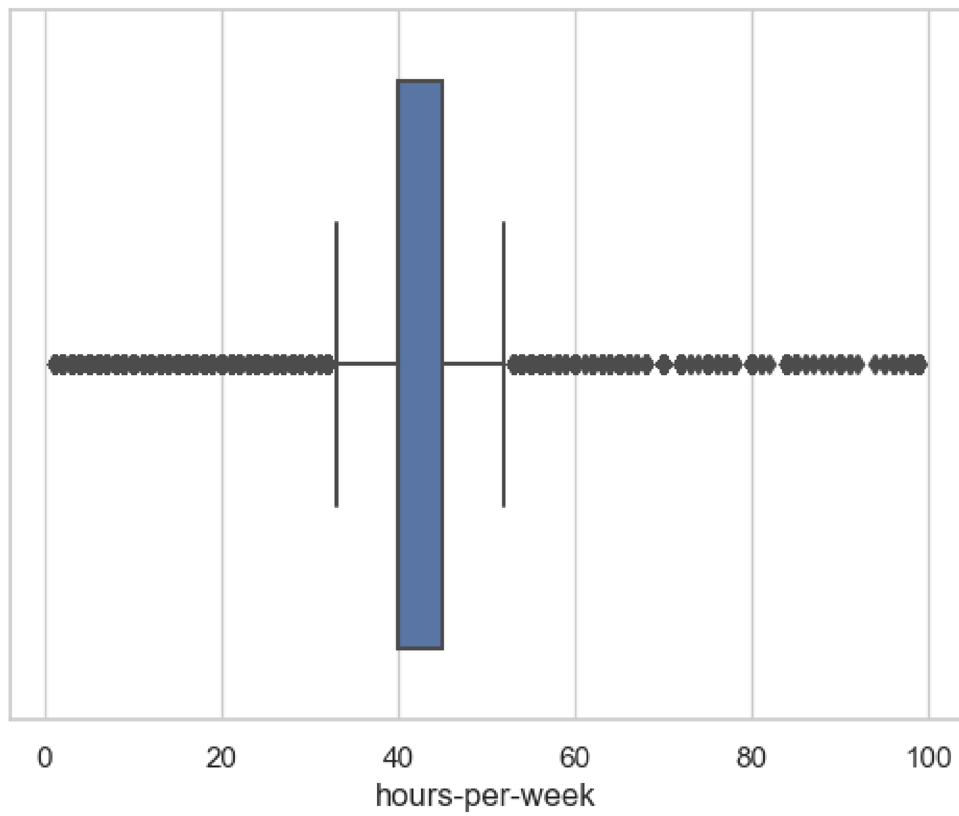



0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
fnlwgt
1e6


2
4
6
8
10
12
14
16
education-num


0
20000
40000
60000
80000
100000
capital-gain


0
1000
2000
3000
4000
capital-loss



1.18 Observation:

1. There are outliers in all numerical features

```
[ ]:
```