In [1]:

```
import numpy as np
import pandas
import seaborn
import matplotlib.pyplot as plt
from sklearn.feature_selection import mutual_info_classif
```

In [2]:

```
df = pandas.read_csv('datasets/adult.csv')
df.head()
```

Out[2]:

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

In [3]:

```
continuous_features = ['age','hours-per-week']
discrete_features = ['workclass', 'education', 'education-num', 'marital-status','occupation', 'relationship', 'race', 'sex','native-country', 'income']
```

Calculate the entropy for each feature

$$h(X) = -\sum_x P(x) log_2 P(x)$$

Graph the mutual information matrix

$$IM(X,Y) = \sum_x \sum_y P(x,y) log_2 \frac{P(x,y)}{P(x)P(y)}$$

In [4]:

```
def entropy(X):
    n = len(X)
    valuesX, countsX = np.unique(X,return_counts=True)
    probX = countsX/n
    return -np.sum( probX * np.log2(probX) ), -np.log2(1/len(valuesX))
```

In [5]:

```
def mutual_information(X,Y):
    n = len(X) # n = len(Y)
    valuesX, countsX = np.unique(X,return_counts=True)
    valuesY, countsY = np.unique(Y,return_counts=True)
    probX = countsX/n
    probY = countsY/n

    im = 0
    for i,x in enumerate(valuesX):
        for j,y in enumerate(valuesY):
            probXY = np.sum(np.logical_and( X==x, Y==y))/n
            if probXY>0:
                im += probXY * np.log2( probXY/ (probX[i]*probY[j]) )
    return im
```

In [6]:

```
for feature in discrete_features:
    print(feature, entropy(df[feature].values) )
```

```
workclass (1.647976927509927, 3.1699250014423126)
education (2.9313508978037115, 4.0)
education-num (2.9313508978037115, 4.0)
marital-status (1.8336493538835446, 2.807354922057604)
occupation (3.516903064343104, 3.9068905956085187)
relationship (2.1544237955049743, 2.584962500721156)
race (0.7987406510139586, 2.321928094887362)
sex (0.9157360598501509, 1.0)
native-country (0.9437954138017222, 5.392317422778761)
income (0.7963839552022132, 1.0)
```

In [7]:

```
IM = np.zeros((len(discrete_features)+1,len(discrete_features)+1))
names = np.array(['entropy'])
names = np.concatenate( (names,discrete_features))
for f1, feature1 in enumerate(discrete_features):
    print(f1,'/',len(discrete_features))
    IM[f1+1,0] = IM[0,f1+1] =  entropy(df[feature1].values)[0]
    for f2, feature2 in enumerate( discrete_features):
        IM[f1+1,f2+1] = mutual_information( df[feature1].values, df[feature2].values
)

dfIM = pandas.DataFrame(IM,columns=names,index=names)
dfIM
```

```
0 / 10
1 / 10
2 / 10
3 / 10
4 / 10
5 / 10
6 / 10
7 / 10
8 / 10
9 / 10
```

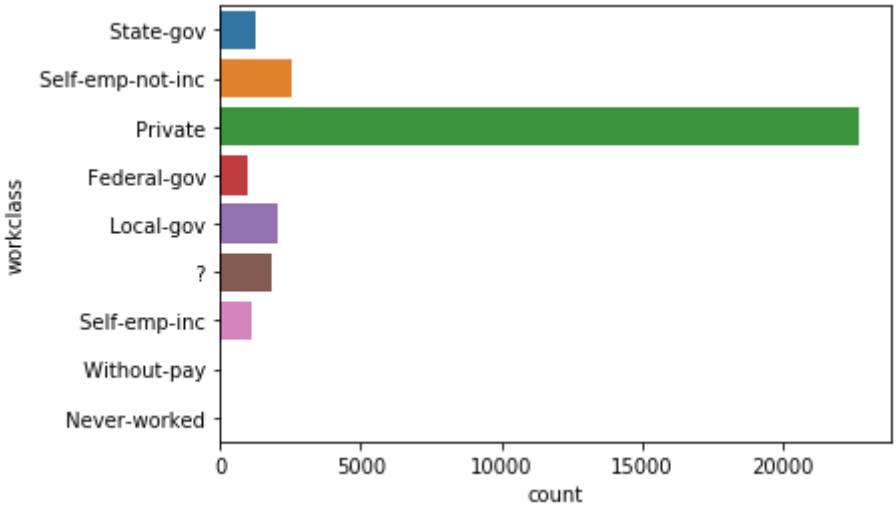
Out[7]:

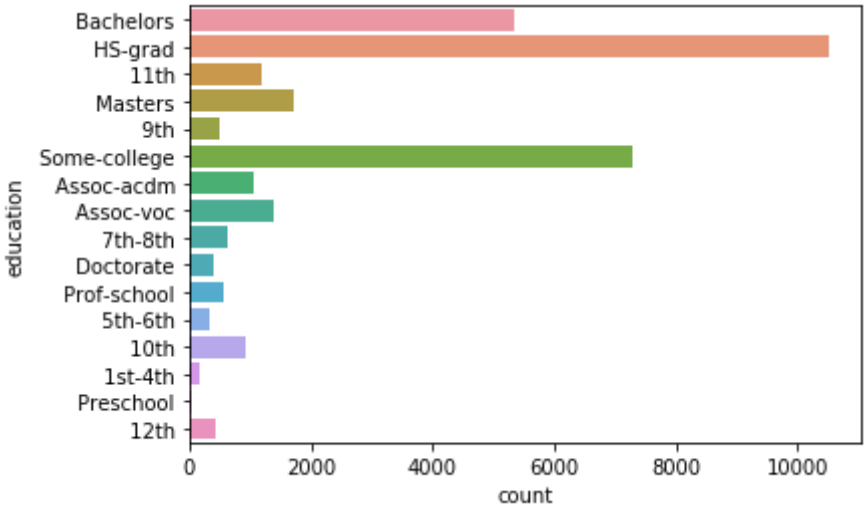
| | entropy | workclass | education | education-num | marital-status | occupation | relationship | |
|---|---|---|---|---|---|---|---|---|
| **entropy** | 0.000000 | 1.647977 | 2.931351 | 2.931351 | 1.833649 | 3.516903 | 2.154424 | 0 |
| **workclass** | 1.647977 | 1.647977 | 0.047141 | 0.047141 | 0.031334 | 0.471202 | 0.035284 | 0 |
| **education** | 2.931351 | 0.047141 | 2.931351 | 2.931351 | 0.034507 | 0.322677 | 0.054353 | 0 |
| **education-num** | 2.931351 | 0.047141 | 2.931351 | 2.931351 | 0.034507 | 0.322677 | 0.054353 | 0 |
| **marital-status** | 1.833649 | 0.031334 | 0.034507 | 0.034507 | 1.833649 | 0.077623 | 1.046677 | 0 |
| **occupation** | 3.516903 | 0.471202 | 0.322677 | 0.322677 | 0.077623 | 3.516903 | 0.121359 | 0 |
| **relationship** | 2.154424 | 0.035284 | 0.054353 | 0.054353 | 1.046677 | 0.121359 | 2.154424 | 0 |
| **race** | 0.798741 | 0.010107 | 0.014904 | 0.014904 | 0.018547 | 0.018315 | 0.024373 | 0 |
| **sex** | 0.915736 | 0.018833 | 0.006852 | 0.006852 | 0.163419 | 0.143273 | 0.394068 | 0 |
| **native-country** | 0.943795 | 0.013488 | 0.077934 | 0.077934 | 0.017175 | 0.042093 | 0.018312 | 0 |
| **income** | 0.796384 | 0.021572 | 0.093591 | 0.093591 | 0.156528 | 0.092922 | 0.165366 | 0 |

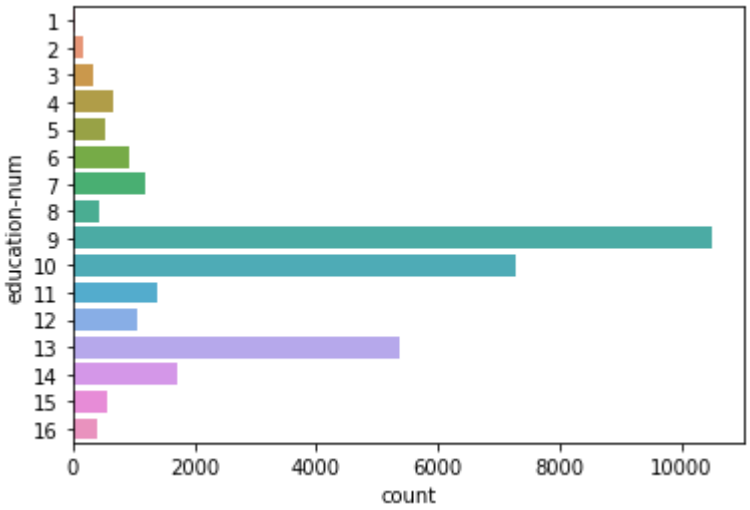
Graph a countplot or a rectangle plot for each feature

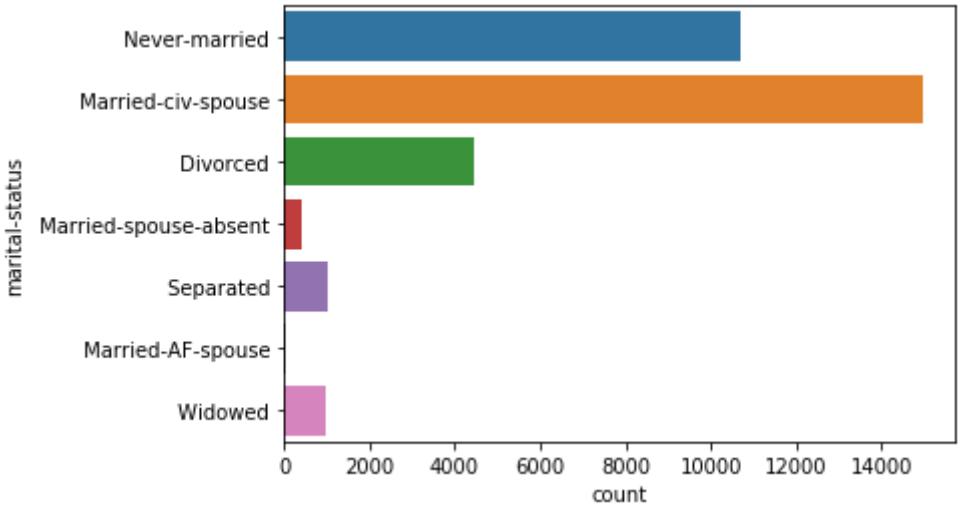
In [8]:

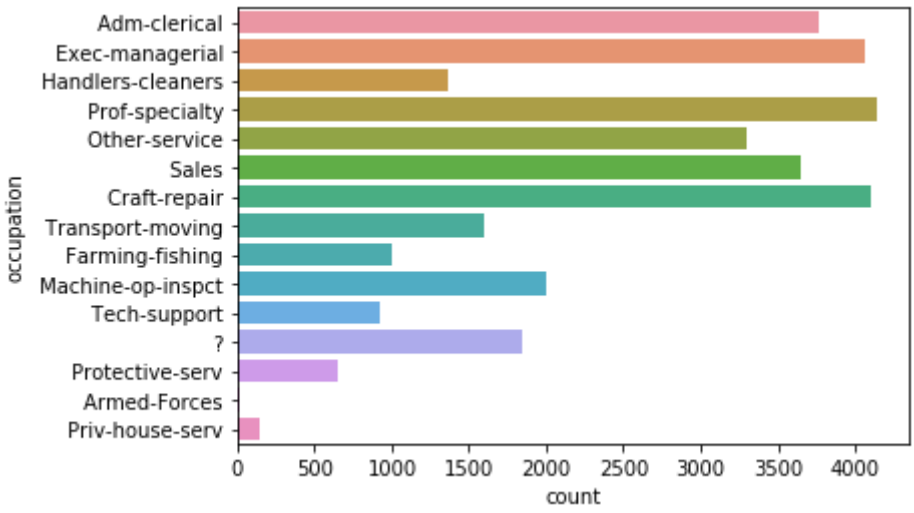
```
for feature in discrete_features:
    plt.figure()
    seaborn.countplot(data=df,y=feature)
    plt.show()
```

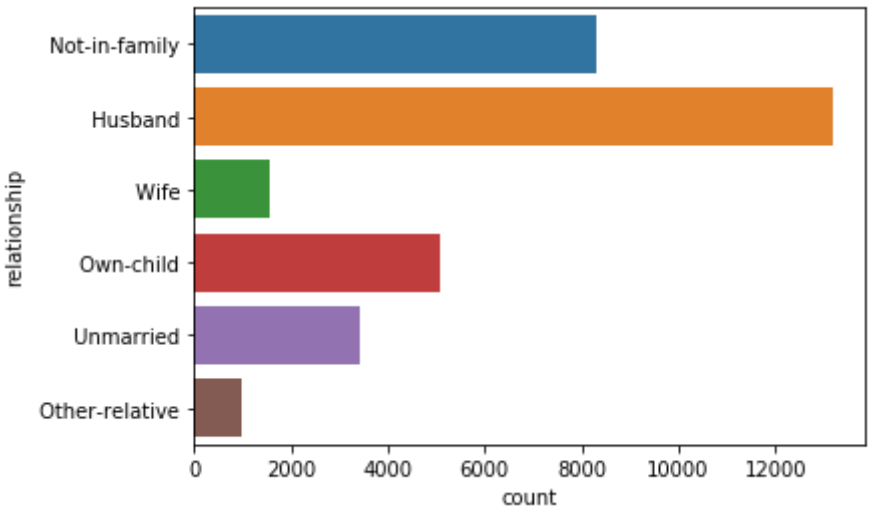

State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
?
Self-emp-inc
Without-pay
Never-worked
workclass
0
5000
10000
15000
20000
count

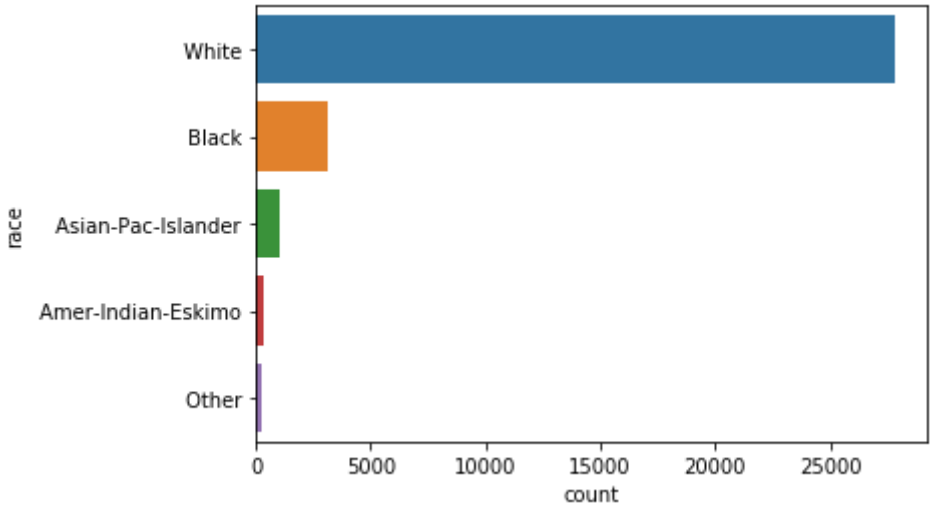

Bachelors
HS-grad
11th
Masters
9th
Some-college
Assoc-acdm
Assoc-voc
7th-8th
Doctorate
Prof-school
5th-6th
10th
1st-4th
Preschool
12th
education
0
2000
4000
6000
8000
10000
count

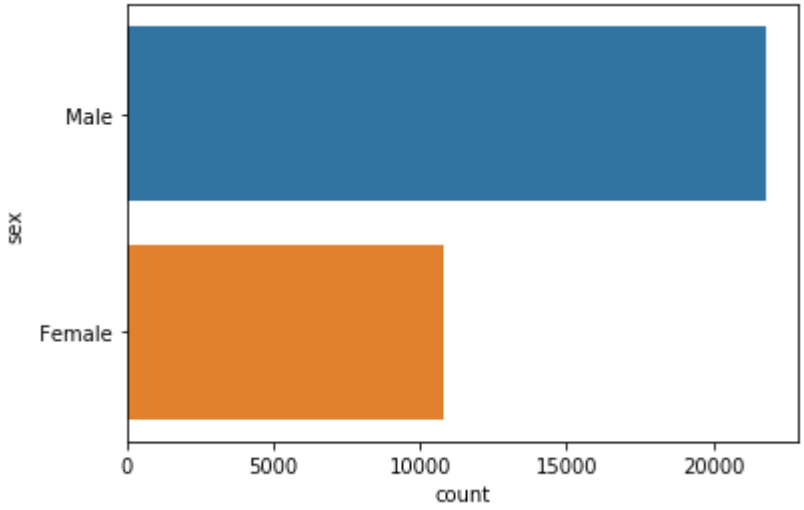

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
education-num
0
2000
4000
6000
8000
10000
count

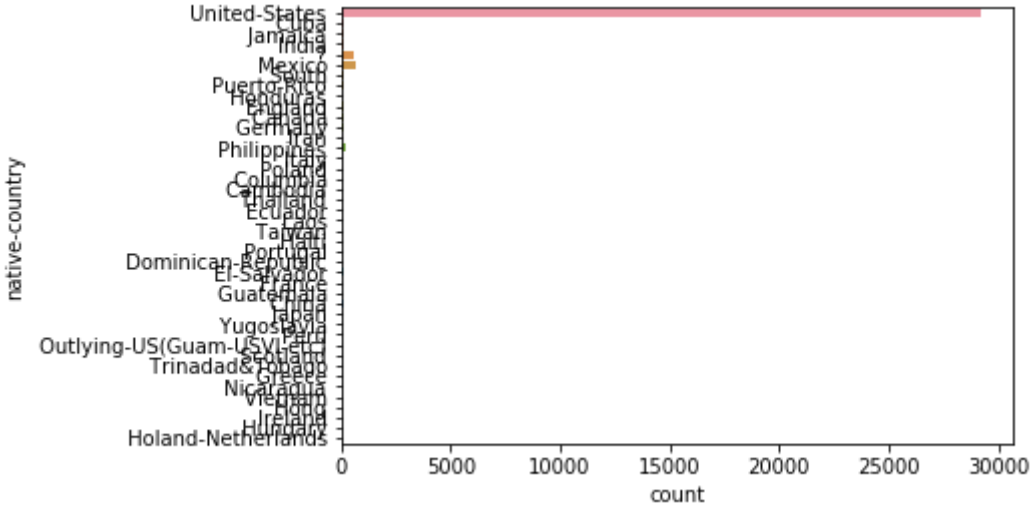

Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed
marital-status
0
2000
4000
6000
8000
10000
12000
14000
count

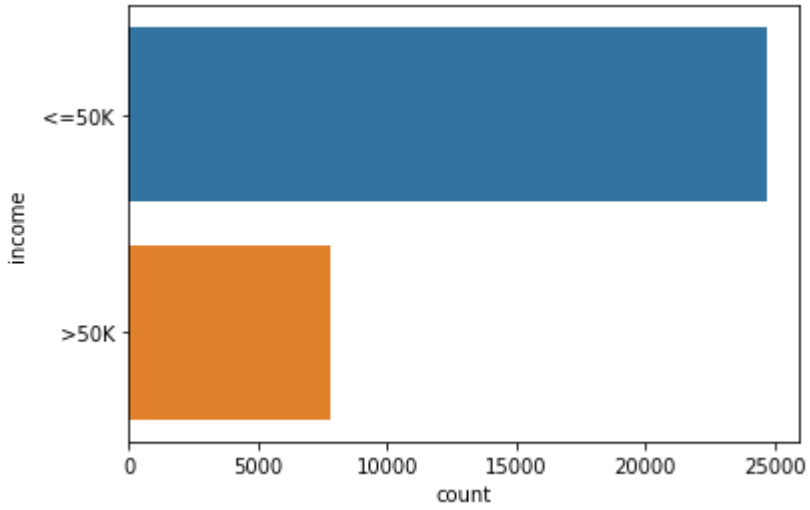

Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Craft-repair
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
?
Protective-serv
Armed-Forces
Priv-house-serv
occupation
0
500
1000
1500
2000
2500
3000
3500
4000
count


Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
relationship
0
2000
4000
6000
8000
10000
12000
count


White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other
race
0
5000
10000
15000
20000
25000
count


Male
Female
sex
0
5000
10000
15000
20000
count


United-States
Mexico
Philippines
Dominican-Republic
El-Salvador
Yugoslavia
Outlying-US(Guam-USVI-etc)
Trinadad&Tobago
Nicaragua
Holand-Netherlands
native-country
0
5000
10000
15000
20000
25000
30000
count




Generate a visualization to compare two features, for example, sex and education

In [9]:

```
np.unique(df['sex'].values)
```

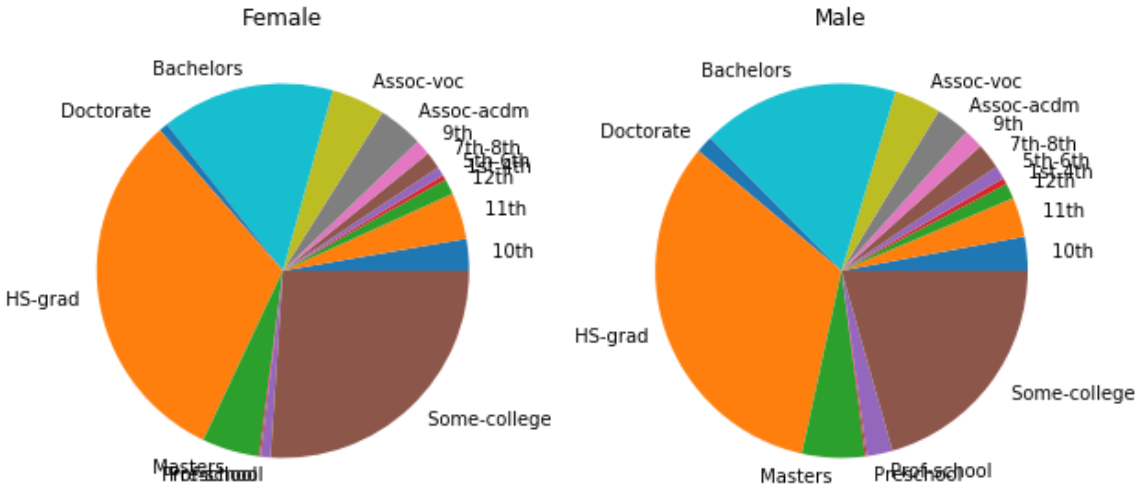
Out[9]:

```
array([' Female', ' Male'], dtype=object)
```

In [10]:

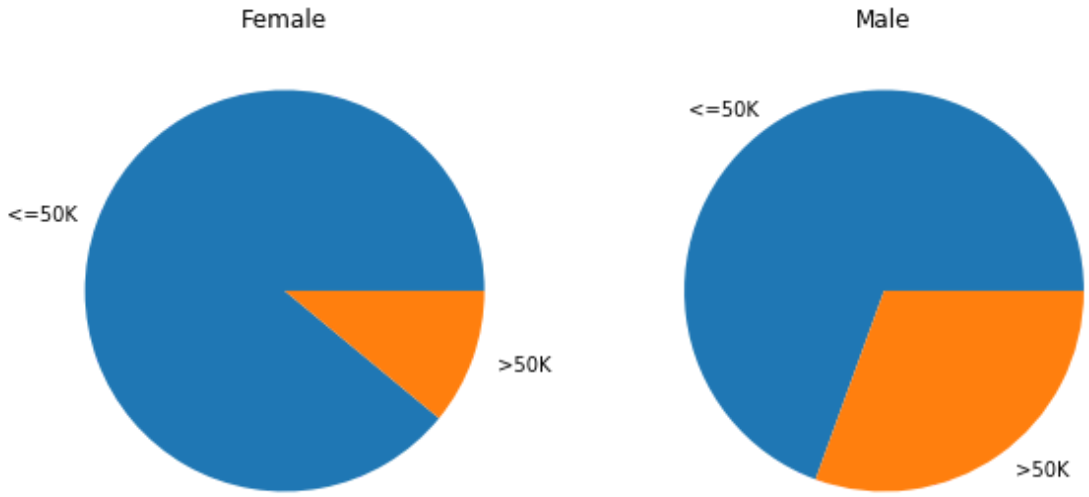
```
idx1 = df['sex'].values == ' Female'
idx2 = df['sex'].values == ' Male'


plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title('Female')
values,counts = np.unique( df['education'].values[idx1],return_counts=True )
plt.pie(counts,labels=values)
plt.subplot(1,2,2)
plt.title('Male')
values,counts = np.unique( df['education'].values[idx2],return_counts=True )
plt.pie(counts,labels=values)
plt.show()
```



In [11]:

```
idx1 = df['sex'].values == ' Female'
idx2 = df['sex'].values == ' Male'
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title('Female')
values,counts = np.unique( df['income'].values[idx1],return_counts=True )
plt.pie(counts,labels=values)
plt.subplot(1,2,2)
plt.title('Male')
values,counts = np.unique( df['income'].values[idx2],return_counts=True )
plt.pie(counts,labels=values)
plt.show()
```

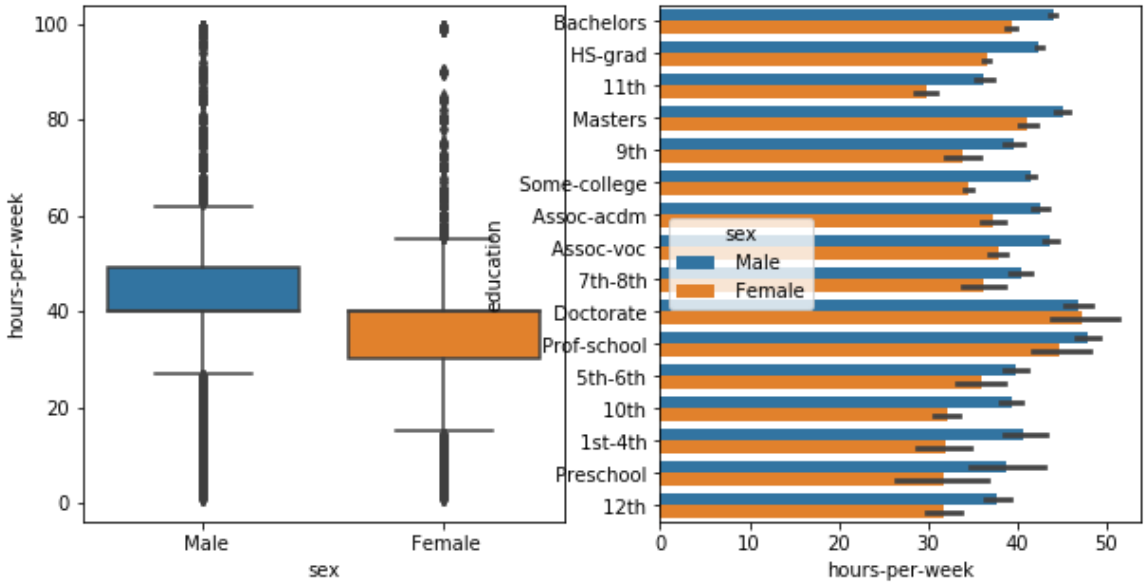



Write at least two insights from the data (justifying with metrics or graphs). Insights consist of some interesting facts that you discover from the Exploratory Data Analysis.

1. Based on data, female work fewer hours than male

In [12]:

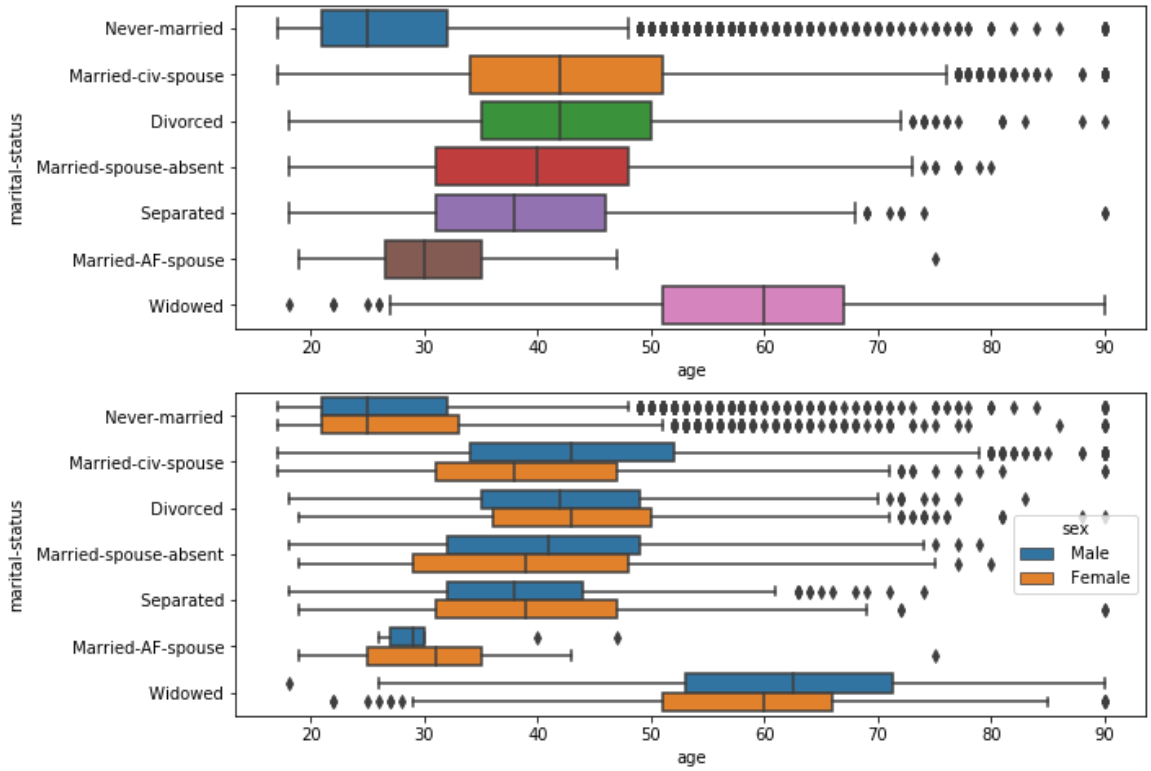
```
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
seaborn.boxplot(data=df, x='sex', y='hours-per-week')
plt.subplot(1,2,2)
seaborn.barplot(data=df, y='education', x='hours-per-week', hue='sex')
plt.show()
```




1. The never-married people are young and widowed people are old. It is not related to sex. Widowed people less than 27-year-old are considered atypical.

In [13]:

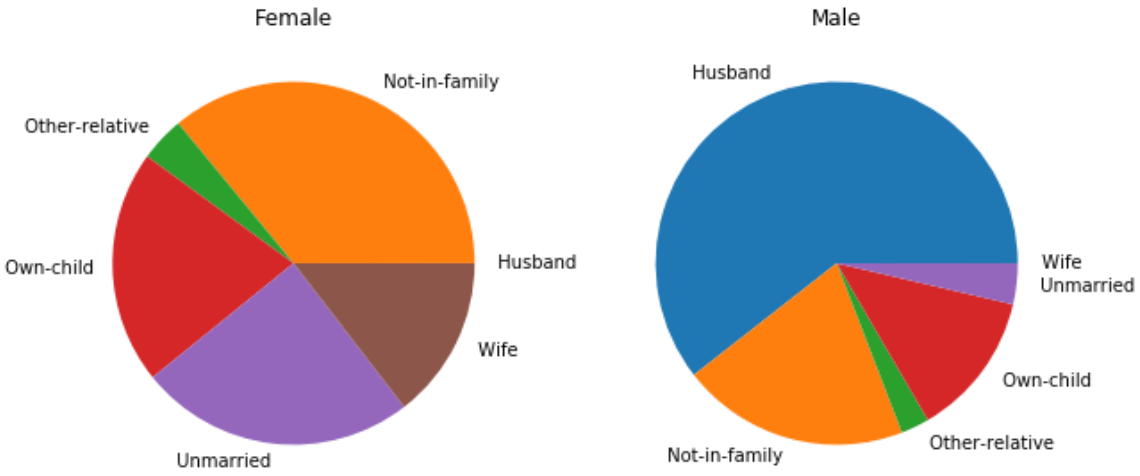
```
plt.figure(figsize=(10,8))
plt.subplot(2,1,1)
seaborn.boxplot(data=df, y='marital-status', x='age')
plt.subplot(2,1,2)
seaborn.boxplot(data=df, y='marital-status', x='age', hue='sex')
plt.show()
```




1. There are more unmarried women than men. There are more married men than women.

In [14]:

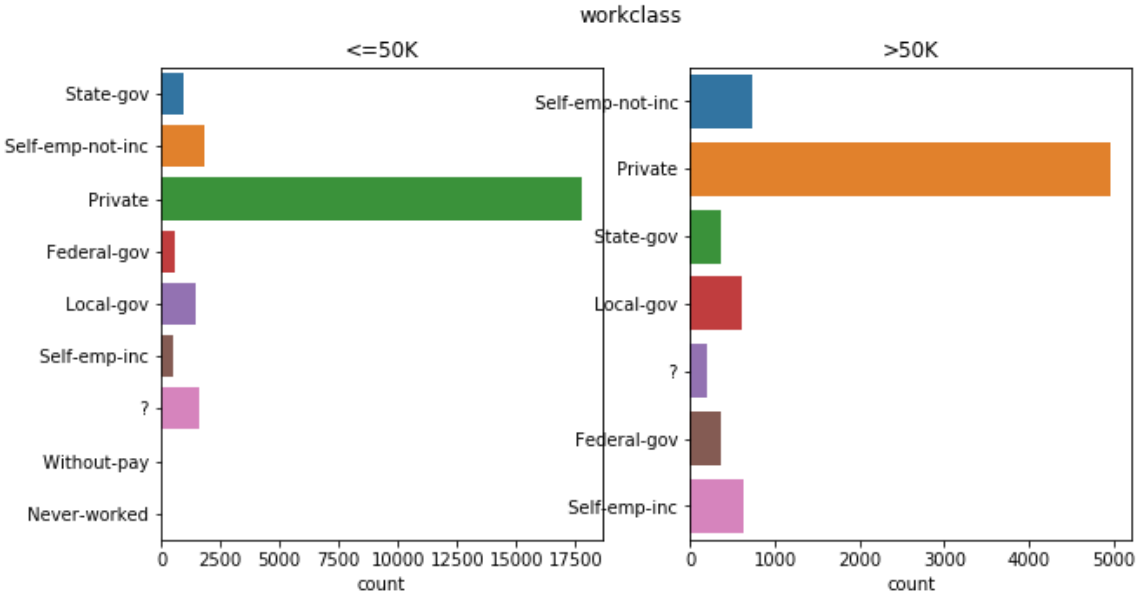
```
idx1 = df['sex'].values == ' Female'
idx2 = df['sex'].values == ' Male'
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title('Female')
values,counts = np.unique( df['relationship'].values[idx1],return_counts=True )
plt.pie(counts,labels=values)
plt.subplot(1,2,2)
plt.title('Male')
values,counts = np.unique( df['relationship'].values[idx2],return_counts=True )
plt.pie(counts,labels=values)
plt.show()
```

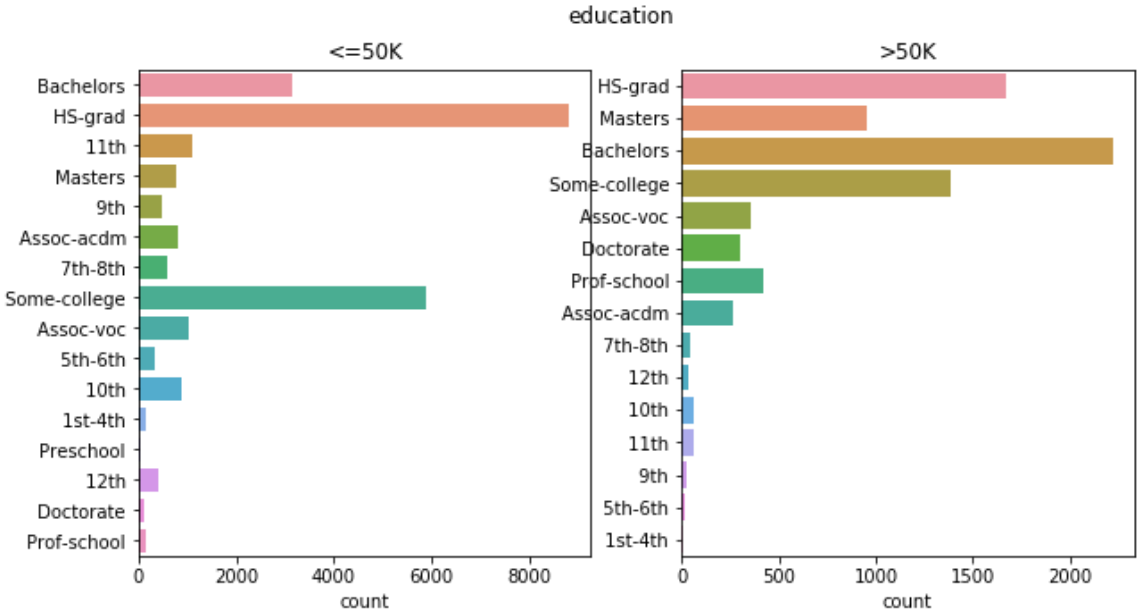


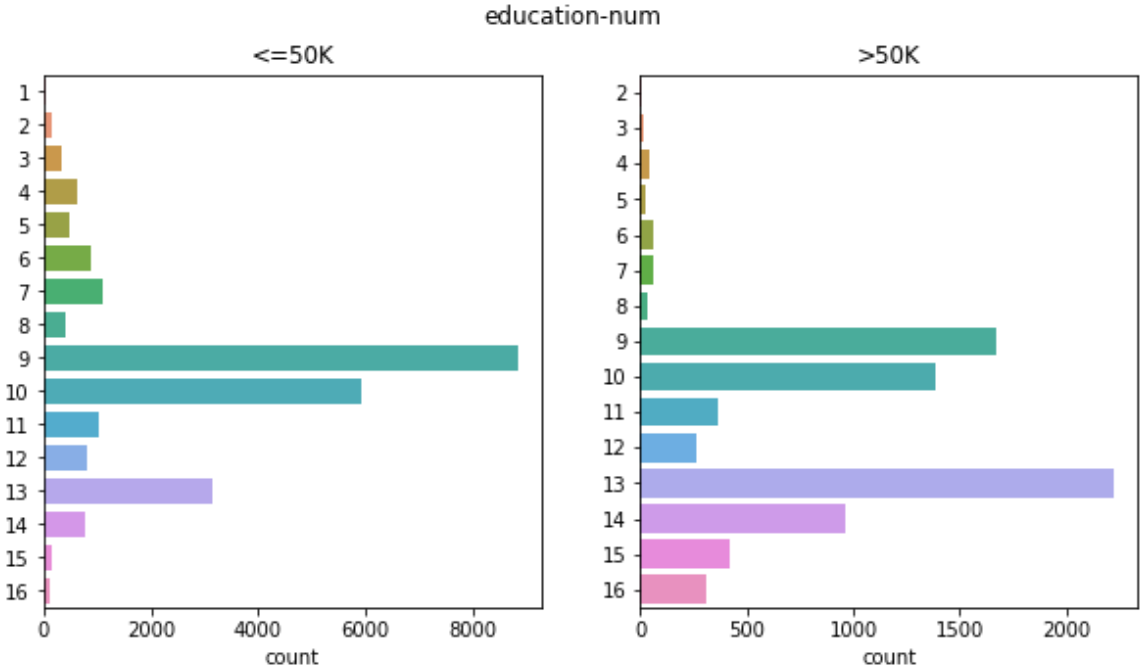

1. The income is related to your education, marital status, relationship, and sex

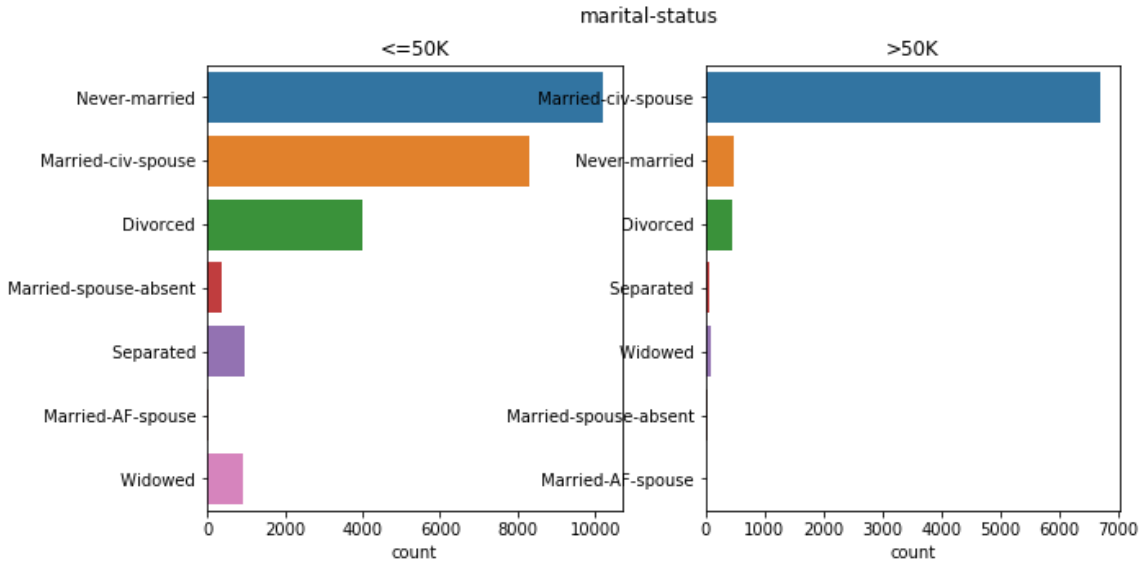
In [15]:

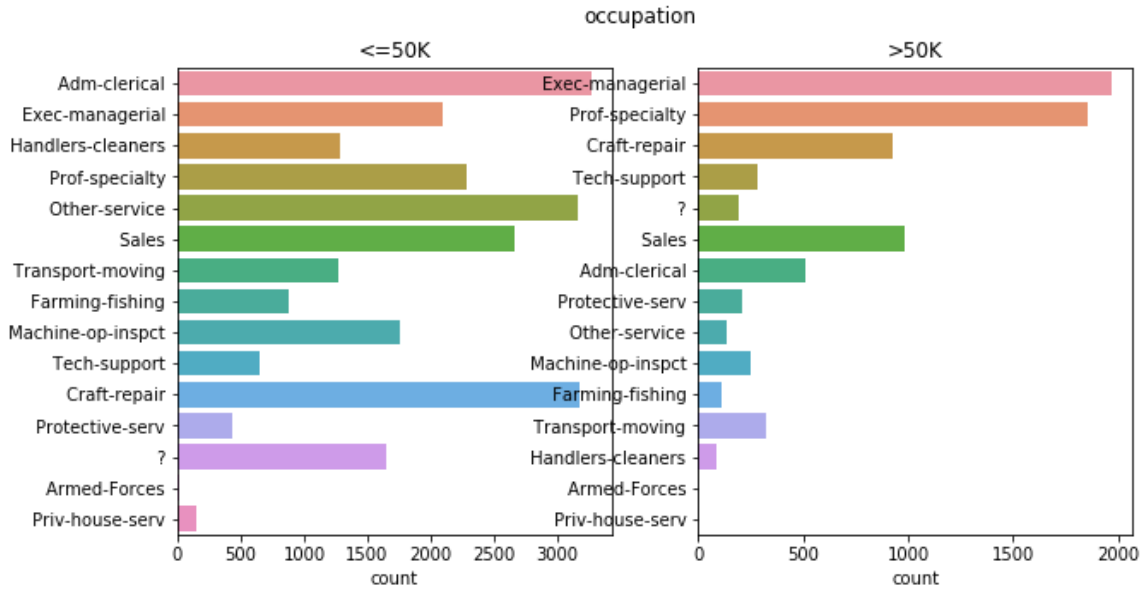
```
for feature in discrete_features:
    idx1 = df['income'].values == ' <=50K'
    idx2 = df['income'].values == ' >50K'
    plt.figure(figsize=(10,5))
    plt.suptitle(feature)
    plt.subplot(1,2,1)
    plt.title('<=50K')
    seaborn.countplot( y= df[feature].values[idx1] )
    plt.subplot(1,2,2)
    plt.title('>50K')
    seaborn.countplot( y= df[feature].values[idx2] )
    plt.show()
```

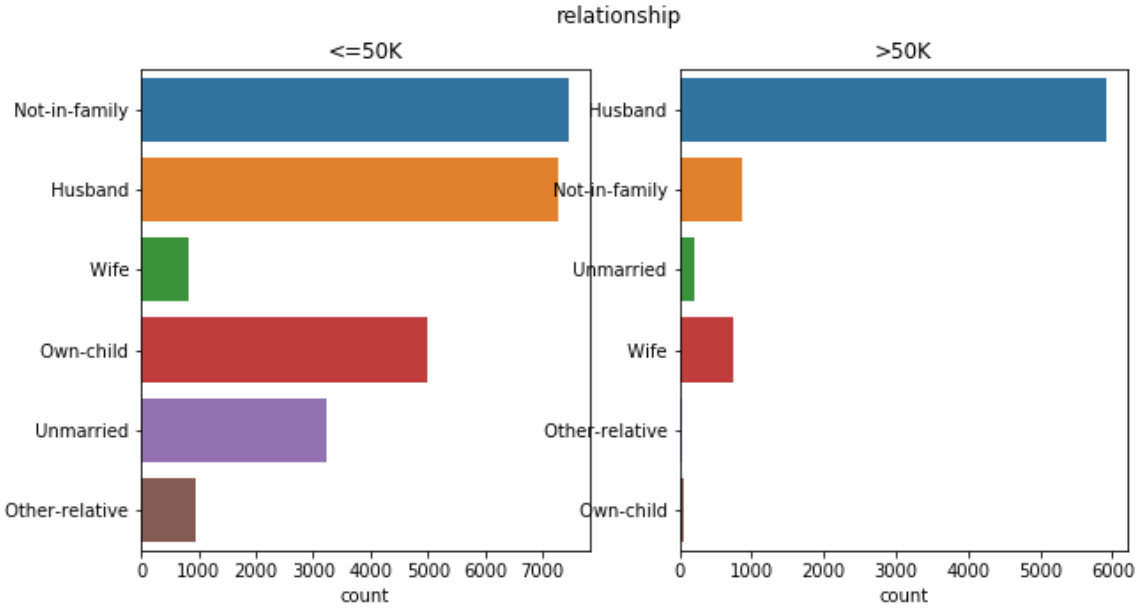

workclass
<=50K
State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
Self-emp-inc
?
Without-pay
Never-worked
0 2500 5000 7500 10000 12500 15000 17500
count
>50K
Self-emp-not-inc
Private
State-gov
Local-gov
?
Federal-gov
Self-emp-inc
0 1000 2000 3000 4000 5000
count

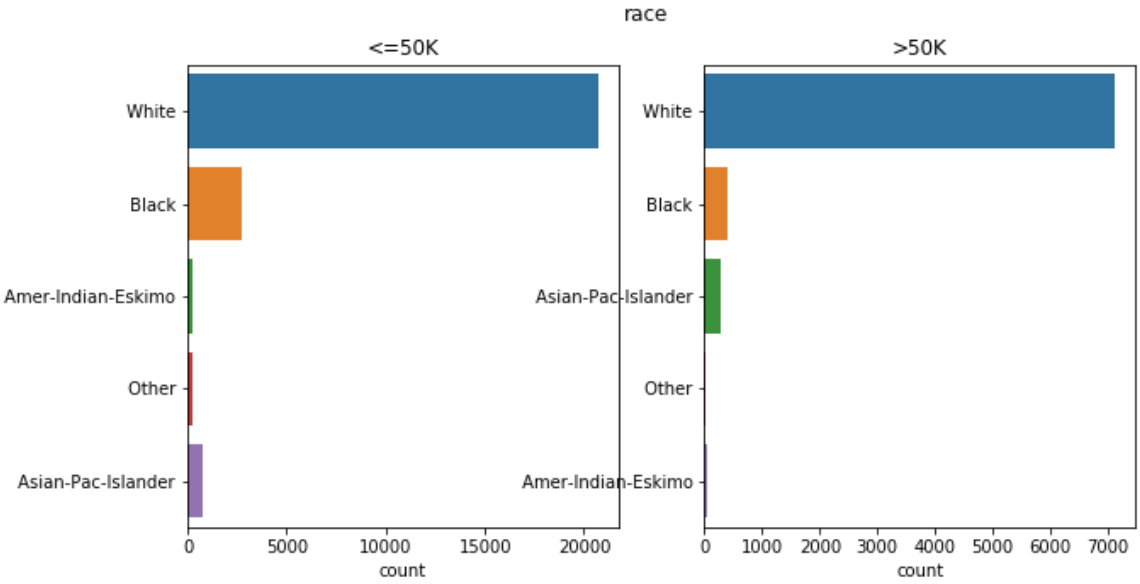

education
<=50K
Bachelors
HS-grad
11th
Masters
9th
Assoc-acdm
7th-8th
Some-college
Assoc-voc
5th-6th
10th
1st-4th
Preschool
12th
Doctorate
Prof-school
0 2000 4000 6000 8000
count
>50K
HS-grad
Masters
Bachelors
Some-college
Assoc-voc
Doctorate
Prof-school
Assoc-acdm
7th-8th
12th
10th
11th
9th
5th-6th
1st-4th
0 500 1000 1500 2000
count

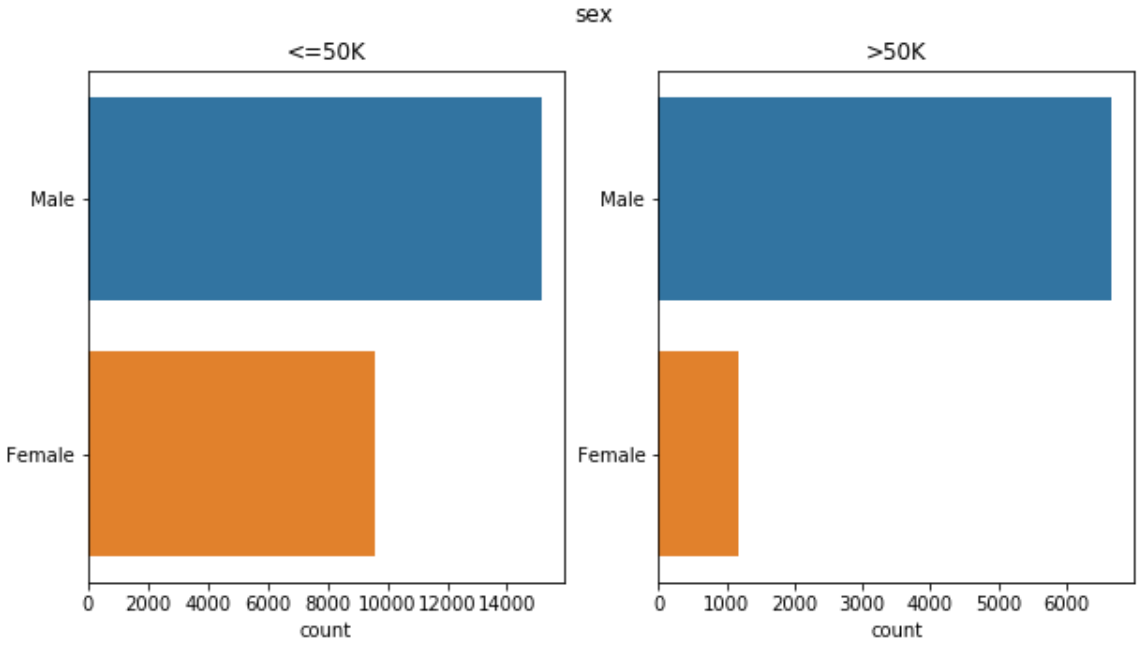

education-num
<=50K
>50K
count
count

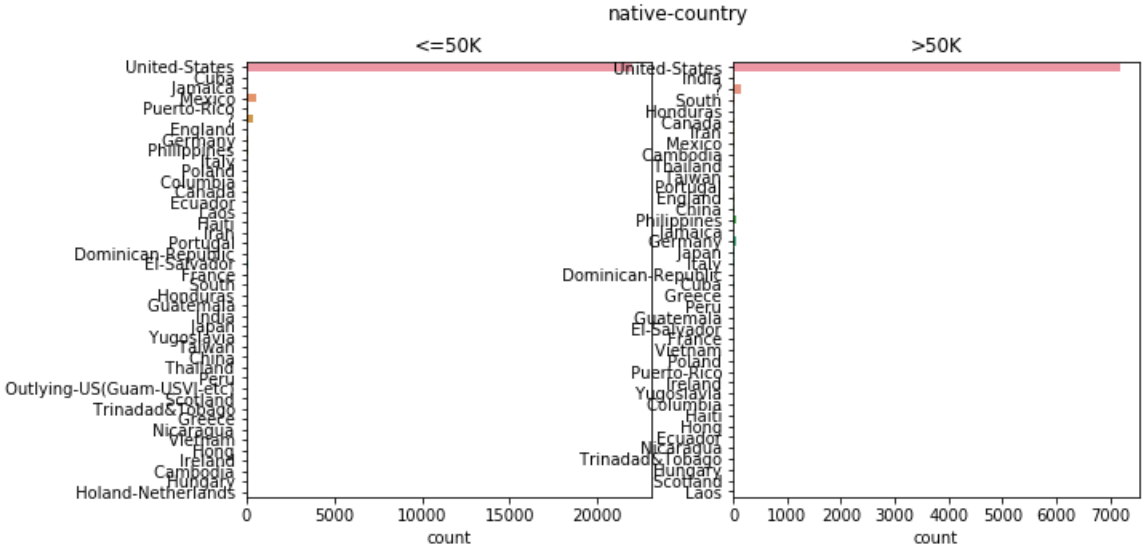

marital-status
<=50K
>50K
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed
count
count

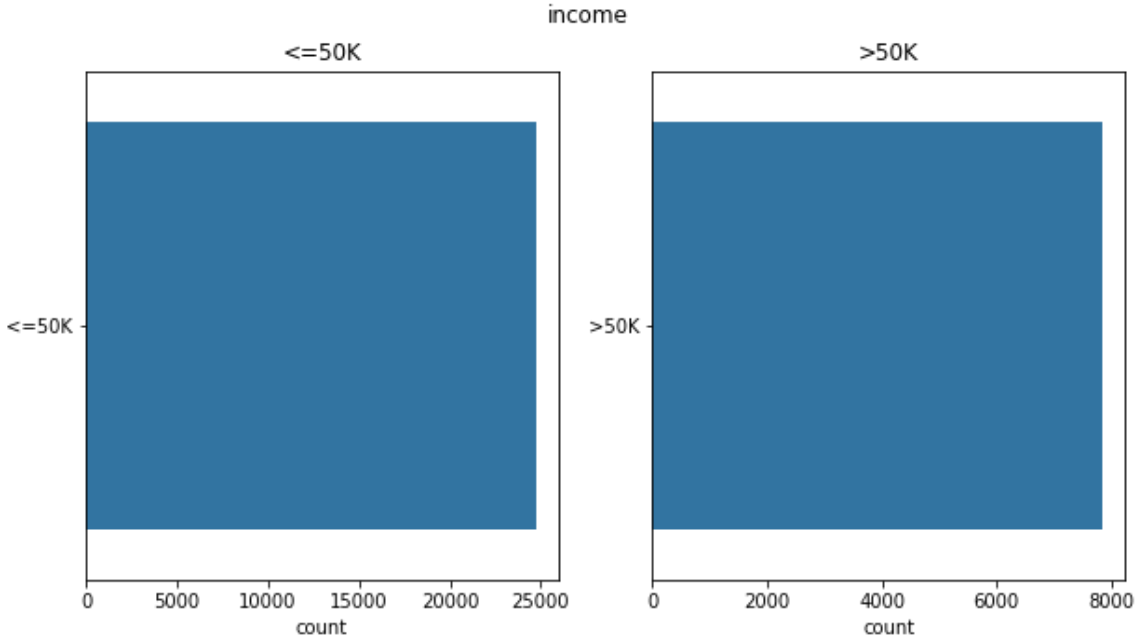

occupation
<=50K
>50K
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
Craft-repair
Protective-serv
?
Armed-Forces
Priv-house-serv
count
count


relationship
<=50K
Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
0 1000 2000 3000 4000 5000 6000 7000
count
>50K
Husband
Not-in-family
Unmarried
Wife
Other-relative
Own-child
0 1000 2000 3000 4000 5000 6000
count


race
<=50K
White
Black
Amer-Indian-Eskimo
Other
Asian-Pac-Islander
0 5000 10000 15000 20000
count
>50K
White
Black
Asian-Pac-Islander
Other
Amer-Indian-Eskimo
0 1000 2000 3000 4000 5000 6000 7000
count


sex
<=50K
Male
Female
0 2000 4000 6000 8000 10000 12000 14000
count
>50K
Male
Female
0 1000 2000 3000 4000 5000 6000
count


native-country
<=50K
>50K
United-States
Cuba
Jamaica
Mexico
Puerto-Rico
?
England
Germany
Philippines
Italy
Poland
Columbia
Canada
Ecuador
Laos
Haiti
Iran
Portugal
Dominican-Republic
El-Salvador
France
South
Honduras
Guatemala
India
Japan
Yugoslavia
Taiwan
China
Thailand
Peru
Outlying-US(Guam-USVI-etc)
Scotland
Trinadad&Tobago
Greece
Nicaragua
Vietnam
Hong
Ireland
Cambodia
Hungary
Holand-Netherlands
India
South
Honduras
Canada
Iran
Mexico
Cambodia
Thailand
Taiwan
Portugal
England
China
Philippines
Jamaica
Germany
Japan
Italy
Dominican-Republic
Cuba
Greece
Peru
Guatemala
El-Salvador
France
Vietnam
Poland
Puerto-Rico
Ireland
Yugoslavia
Columbia
Haiti
Hong
Ecuador
Nicaragua
Trinadad&Tobago
Hungary
Scotland
Laos
0
5000
10000
15000
20000
count
0
1000
2000
3000
4000
5000
6000
7000
count


income
<=50K
>50K
<=50K
>50K
0
5000
10000
15000
20000
25000
count
0
2000
4000
6000
8000
count