Which Feature indicates to higher Income level (>50k)

- Importing all the neccessary Libraries for the analysis

In [54]:

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

- Importing the CSV file and replacing hypen with underscore and assigning new names to the column

In [55]:

```
new_cols = ["age","workclass", "fnlwgt","education","education_num","marital_status","occu
df = pd.read_csv("C:/Users/ROSHAN D K/Desktop/Python Projects/censusData.csv", names = new
df.head()
```

Out[55]:

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

- Replacing the missing data from ? to np.nan so that we can access or delete all the empty records from the dataset

In [56]:

```
df.replace("?", np.nan, inplace = True) # replaced with null values
```

Exploring the dataset

We have 15 columns and 32561 rows/records

In [57]:

```
df.shape
```

Out[57]:

```
(32561, 15)
```

In [58]:

```
df.dtypes # There are more number of categorical variables as compared to numerical varial
```

Out[58]:

```
age                int64
workclass         object
fnlwgt             int64
education         object
education_num      int64
```

```
marital_status    object
occupation        object
relationship      object
race              object
sex               object
capital_gain       int64
capital_loss       int64
hours_per_week     int64
native_country    object
income            object
dtype: object
```

Assigning education number to each education layer for better analysis(A. (2021, July 4)).

In [59]:

```
education_level = pd.DataFrame(df.groupby(['education','education_num'])[['education']].co
education_level.columns = ['education','education_num']
education_level.sort_values(by='education_num')
```

Out[59]:

	education	education_num
13	Preschool	1
3	1st-4th	2
4	5th-6th	3
5	7th-8th	4
6	9th	5
0	10th	6
1	11th	7
2	12th	8
11	HS-grad	9
15	Some-college	10
8	Assoc-voc	11
7	Assoc-acdm	12
9	Bachelors	13
12	Masters	14
14	Prof-school	15
10	Doctorate	16

From the table below we can see that we have empty values in the 3 columns

In [60]:

```
df.isna().sum()
```

Out[60]:

```
age                  0
workclass         1836
fnlwgt               0
education            0
education_num        0
marital_status       0
occupation        1843
relationship         0
race                 0
sex                  0
capital_gain         0
```

```
capital_loss         0
hours_per_week       0
native_country     583
income               0
dtype: int64
```

- Now, we are going to check how many records have 3 important feature missing (I. (2017, July 24)).

In [61]:

```
filt = ((df.native_country.isna()) & (df.occupation.isna()) & (df.workclass.isna()))
filt.sum()
```

Out[61]:

```
27
```

Now this 27 records are complete useless for visualisation as it does not contain most of the important feature. we could remove all this record.

In [62]:

```
filt = ((df.workclass.isna()) & (df.occupation.isna()))
filt.sum()
```

Out[62]:

```
1836
```

Again here this two variable can be a important feature for my analysis which is missing so I need to remove this records too. Or Try to find the missing values.

In [63]:

```
filt = ((df.native_country.isna()) | (df.occupation.isna()) | (df.workclass.isna()))
filt.sum()
```

Out[63]:

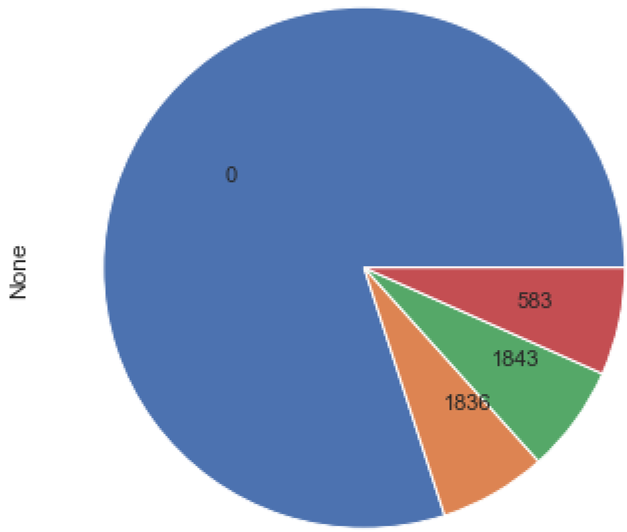
```
2399
```

Now, this number of records have either one or more number of missing values. All of these variables are categorical data type. So, I could impute the missing values with the most frequent value which is mode.

Distribution of Missing values in the dataset
It seems like we have majority of the data present and small percentage is missing

In [64]:

```
# pie chart for proportion of missing values and present values and some suggestion to dea
ol = df.isna().sum().value_counts(normalize = False)
ol.plot(kind='pie', figsize=(10,6),
        pctdistance=1.25, labeldistance=.6);
```



Now, we are going to segregate columns into categorical and numerical variable (P. (2020, March 4)).

In [65]:

```
categ = df[["workclass","education",'marital_status','occupation','relationship','race', '
numeric = df[['age','fnlwgt', 'education_num','capital_gain', 'capital_loss', 'hours_per_w
for i, col in enumerate(categ.columns):
  print(categ.columns[i].upper(), '\n', categ[str(col)].unique(), '\n')
```

```
WORKCLASS
 ['State-gov' 'Self-emp-not-inc' 'Private' 'Federal-gov' 'Local-gov' nan
 'Self-emp-inc' 'Without-pay' 'Never-worked']

EDUCATION
 ['Bachelors' 'HS-grad' '11th' 'Masters' '9th' 'Some-college' 'Assoc-acdm'
 'Assoc-voc' '7th-8th' 'Doctorate' 'Prof-school' '5th-6th' '10th'
 '1st-4th' 'Preschool' '12th']

MARITAL_STATUS
 ['Never-married' 'Married-civ-spouse' 'Divorced' 'Married-spouse-absent'
 'Separated' 'Married-AF-spouse' 'Widowed']

OCCUPATION
 ['Adm-clerical' 'Exec-managerial' 'Handlers-cleaners' 'Prof-specialty'
 'Other-service' 'Sales' 'Craft-repair' 'Transport-moving'
 'Farming-fishing' 'Machine-op-inspct' 'Tech-support' nan
 'Protective-serv' 'Armed-Forces' 'Priv-house-serv']

RELATIONSHIP
 ['Not-in-family' 'Husband' 'Wife' 'Own-child' 'Unmarried' 'Other-relative']

RACE
 ['White' 'Black' 'Asian-Pac-Islander' 'Amer-Indian-Eskimo' 'Other']

SEX
 ['Male' 'Female']

INCOME
 ['<=50K' '>50K']

NATIVE_COUNTRY
 ['United-States' 'Cuba' 'Jamaica' 'India' nan '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']
```

In [66]:
```
categ.nunique() # Number of subcategories in each column represented here
```

Out[66]:
```
workclass          8
education         16
marital_status     7
occupation        14
relationship       6
race               5
sex                2
income             2
native_country    41
dtype: int64
```

Exploring the missing values to get better understanding of what we are dealing with

NA - workclass, occupation, native country

In [67]:
```
country = df['native_country'].isna() # missing values
df.loc[country]
workc = df['workclass'].isna() # missing values
df.loc[workc]
occu = df['occupation'].isna() # missing values
df.loc[occu]
```

Out[67]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex
27	54	NaN	180211	Some-college	10	Married-civ-spouse	NaN	Husband	Asian-Pac-Islander	Male
61	32	NaN	293936	7th-8th	4	Married-spouse-absent	NaN	Not-in-family	White	Male
69	25	NaN	200681	Some-college	10	Never-married	NaN	Own-child	White	Male
77	67	NaN	212759	10th	6	Married-civ-spouse	NaN	Husband	White	Male
106	17	NaN	304873	10th	6	Never-married	NaN	Own-child	White	Female
...	...	...	...	...	...	...	...	...	...	...
32530	35	NaN	320084	Bachelors	13	Married-civ-spouse	NaN	Wife	White	Female
32531	30	NaN	33811	Bachelors	13	Never-married	NaN	Not-in-family	Asian-Pac-Islander	Female
32539	71	NaN	287372	Doctorate	16	Married-civ-spouse	NaN	Husband	White	Male
32541	41	NaN	202822	HS-grad	9	Separated	NaN	Not-in-family	Black	Female
32542	72	NaN	129912	HS-grad	9	Married-civ-spouse	NaN	Husband	White	Male

1843 rows × 15 columns

Data cleaning process

I am going to drop the column name capital gain, capital looses, education number and relationship because it doesnt align with my reserach

In [68]:

```
colsn = df[['relationship', 'capital_gain', 'capital_loss','education_num']]
df.drop(colsn, axis = 1, inplace = True)
df.head()
```

Out[68]:

	age	workclass	fnlwgt	education	marital_status	occupation	race	sex	hours_per_week	native_country	i
0	39	State-gov	77516	Bachelors	Never-married	Adm-clerical	White	Male	40	United-States	
1	50	Self-emp-not-inc	83311	Bachelors	Married-civ-spouse	Exec-managerial	White	Male	13	United-States	
2	38	Private	215646	HS-grad	Divorced	Handlers-cleaners	White	Male	40	United-States	
3	53	Private	234721	11th	Married-civ-spouse	Handlers-cleaners	Black	Male	40	United-States	
4	28	Private	338409	Bachelors	Married-civ-spouse	Prof-specialty	Black	Female	40	Cuba	

Feature Engineering

Now we are going to reduce the categories to visulaise only variables which is relevant to our research (A. (2023, February 13)).

For workclass :- government job :- State-gov, Federal-gov, Local-gov
self employed:- Self-emp-not-inc, Self-emp-inc,
private: / other: Without-pay, Never-worked

Education :- Masters and PhD - Masters, Doctorate, Prof-school
Undergrad :- Bachelors, HS-grad , Assoc-acdm, Assoc-voc, Some-college
Primary education :- Some-college, 5th-6th', ' 10th', ' 1st-4th',' Preschool', ' 12th' 11th, 9th
others

relationship :- Married : Married-civ-spouse, Married-AF-spouse
Single :- Never-married, Widowed , Married-spouse-absent
Divorced :- Divorced, Separated

Race :- white, Black, Asian
Other:' Amer-Indian-Eskimo', Other

I can also apply binning method on the numerical variable like age
age - 0-18 - (child)
19-60 - middle age
61+ - senior citizen

hours/week
below 40 hours - part time

40 hours - full time

40+ - overtime

In [69]:
```
df3 = pd.read_csv("C:/Users/ROSHAN D K/Desktop/Python Projects/censusData.csv", names = ne
df3['sex'].replace({'Male':0, 'Female':1}, inplace = True)
df3['marital_status'].replace({"Married-civ-spouse":"Married",'Never-married':'Single','Se
df3['education'].replace({'Preschool':'Primary education','1st-4th':'Primary education','5
        'Prof-school':'Masters and PhD','Assoc-acdm':'Undergrad','Assoc-voc':'Undergrad','
        'Bachelors':'Undergrad','Masters':'Masters and PhD','Doctorate':'Masters and PhD'}
df3['race'].replace({' Asian-Pac-Islander':'Other', 'Amer-Indian-Eskimo':'Other'}, inplace
df3.head(5)
```

Out[69]:

	age	workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	sex	capital_
0	39	State-gov	77516	Undergrad	13	Single	Adm-clerical	Not-in-family	White	0	2
1	50	Self-emp-not-inc	83311	Undergrad	13	Married	Exec-managerial	Husband	White	0	
2	38	Private	215646	Undergrad	9	Divorced	Handlers-cleaners	Not-in-family	White	0	
3	53	Private	234721	Primary education	7	Married	Handlers-cleaners	Husband	Black	0	
4	28	Private	338409	Undergrad	13	Married	Prof-specialty	Wife	Black	1	

In [70]:
```
df3.nunique()
```

Out[70]:
```
age                  73
workclass             9
fnlwgt            21648
education             3
education_num        16
marital_status        3
occupation           15
relationship          6
race                  4
sex                   2
capital_gain        119
capital_loss         92
hours_per_week       94
native_country       42
income                2
dtype: int64
```

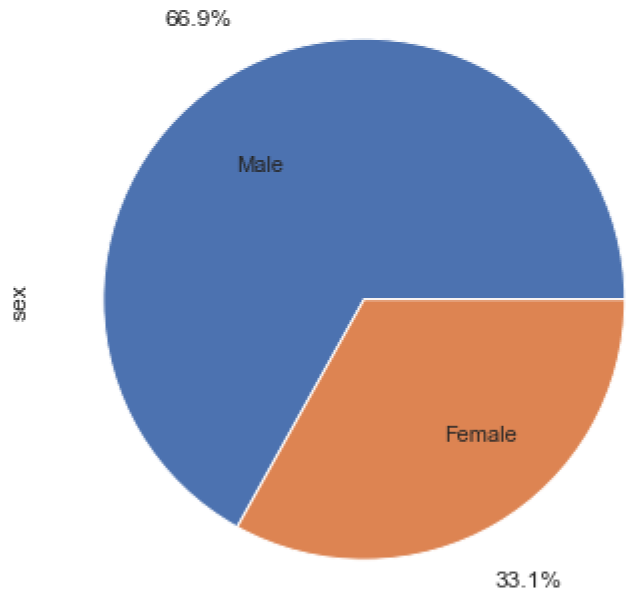
As we can see after applying feature engineering multiple subcategories have been combine together for better visualisation and for model building

Visualisation

In [71]:
```
import seaborn as sns
sns.set_style("whitegrid")
sns.set(rc={'figure.figsize':(10,6)})
```

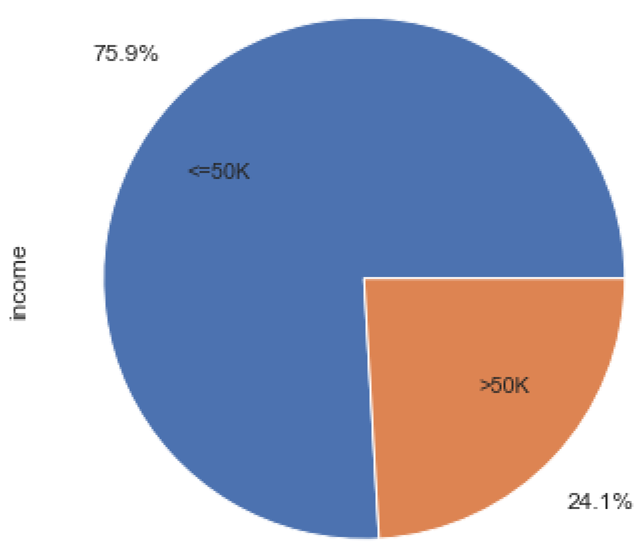
- Checking the Data Distribution of Independent and dependent variable (A. (2023, February 13)).

In [72]:
```
ex = df['sex'].value_counts()
ex.plot(kind='pie', figsize=(10,6), autopct='%1.1f%%',
       pctdistance=1.25, labeldistance=.6,subplots=True);
plt.ylabel('sex');
```



Observation - The result clearly shows that the male proportion in the dataset is double that of the female proportion, indicating that the dataset is not a proper representation of sex.
As a result, we must use the mode technique to fill in the missing values.

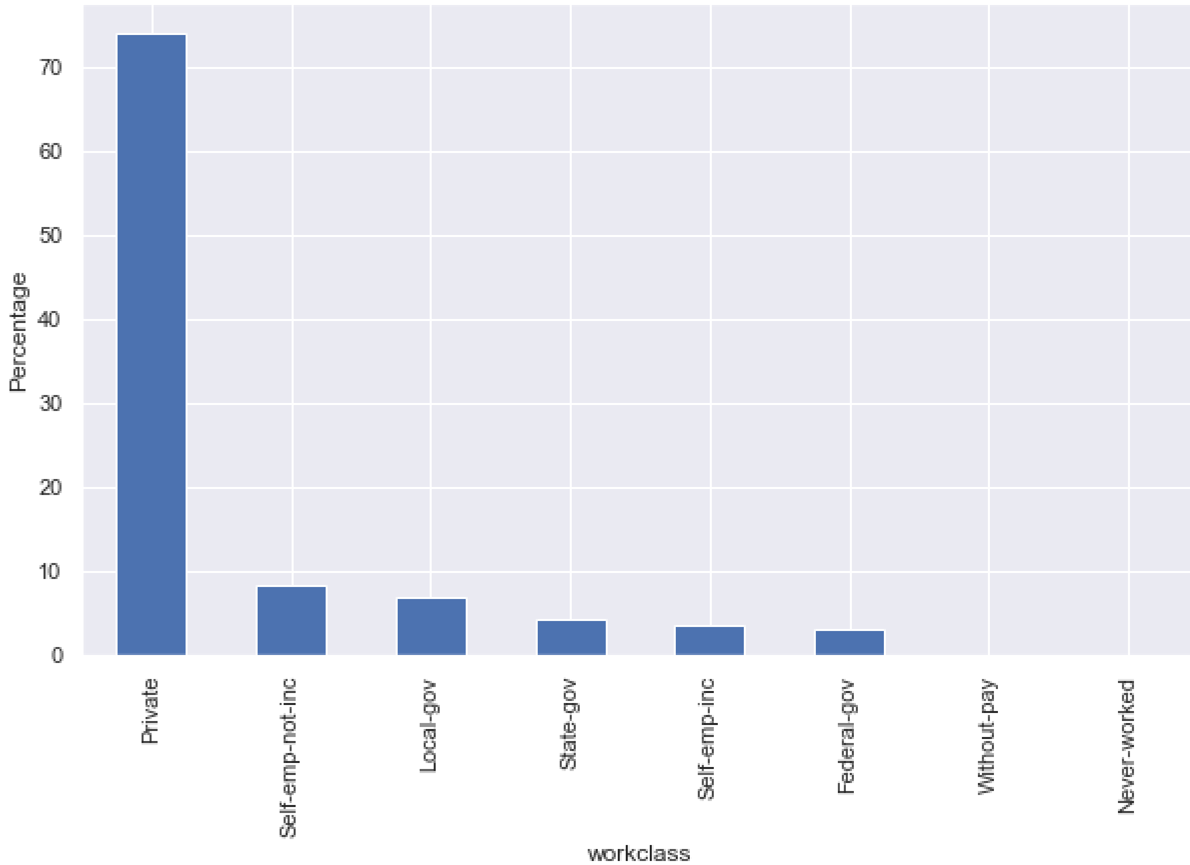
In [73]:
```
ola = df['income'].value_counts()
ax = ola.plot(kind='pie', figsize=(10,6), autopct='%1.1f%%',
       pctdistance=1.25, labeldistance=.6,subplots=True)
plt.ylabel('income');
```



Observation - In this graph it is visible, that there are more than 75% of the people who earns less than 50K while only 24% people earn above 50k.

In [74]:
```
stats = df['workclass'].value_counts(normalize = True)
```

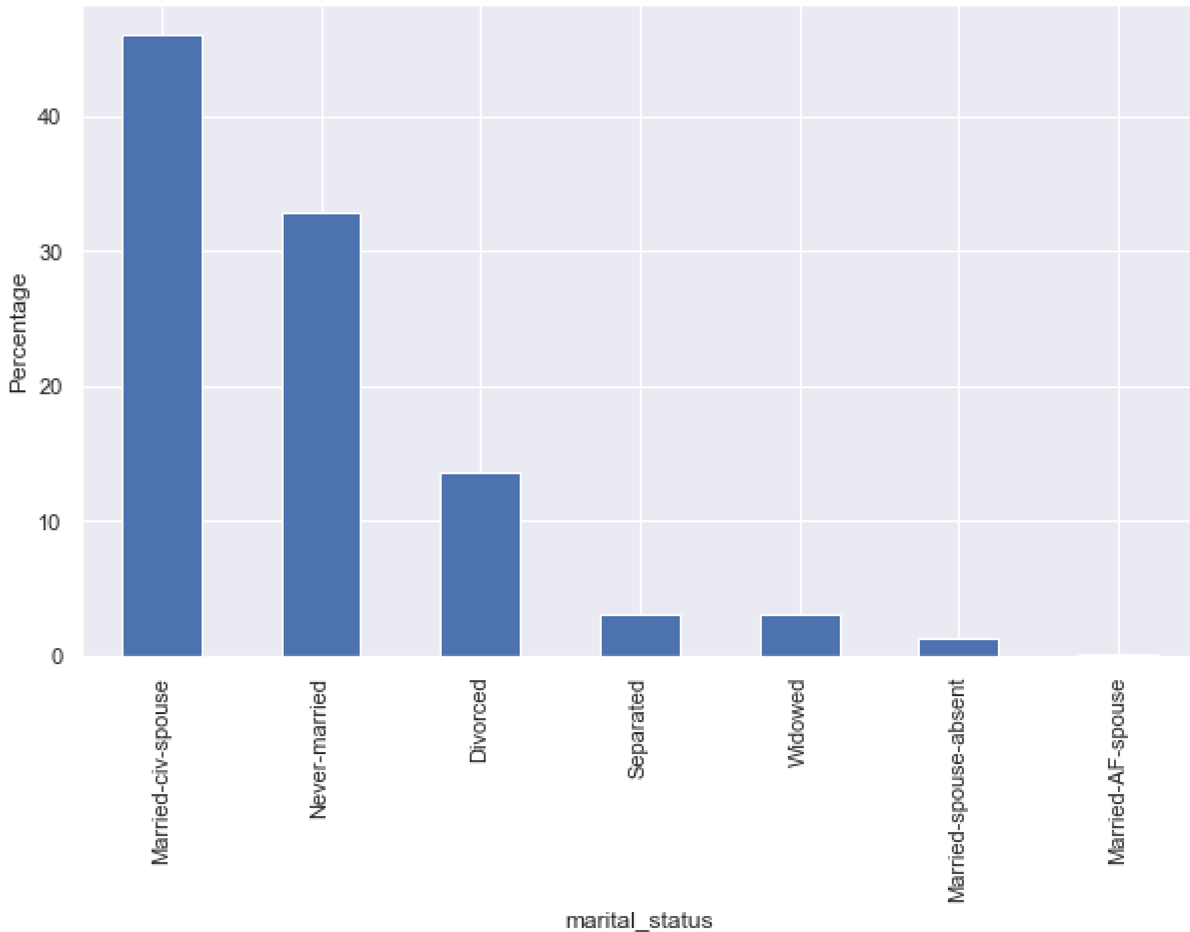
```
ax = stats.mul(100).plot(kind='bar')
plt.xlabel("workclass")
plt.ylabel("Percentage")
plt.title = ('Workclass Distrubtion');
```



Observation - Individuals working in private firms have dominated, while those working in government jobs are less compared to those working in private companies.

In [75]:

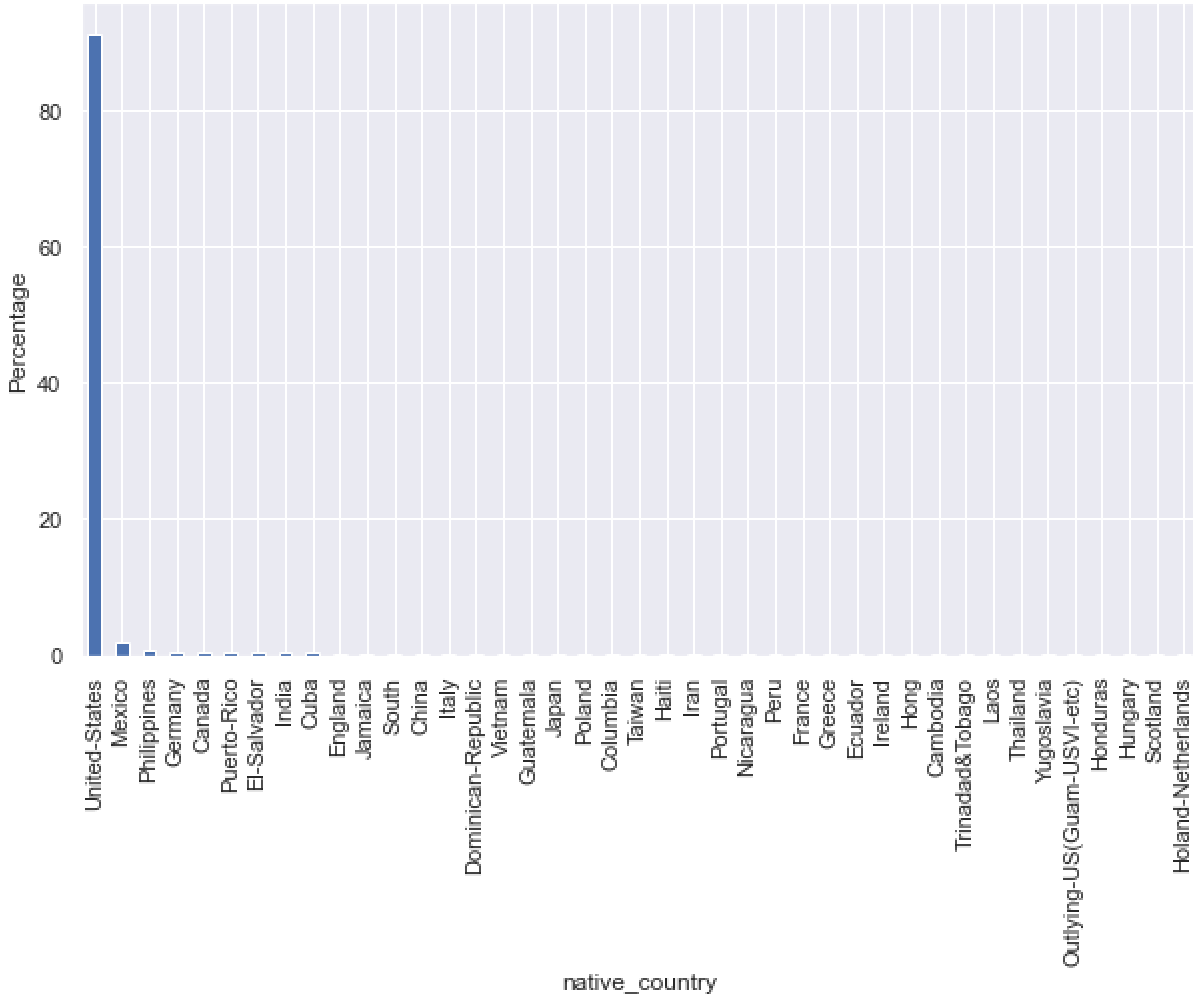
```
mari = df['marital_status'].value_counts(normalize = True)
mari.mul(100).plot(kind='bar')
plt.xlabel("marital_status")
plt.ylabel("Percentage")
plt.title = ('Marital_status Distrubtion');
```



Observation - People who 'never married', 'married with civil spouse' and 'Divorced' people constitute of majority of the population in the dataset.

In [76]:

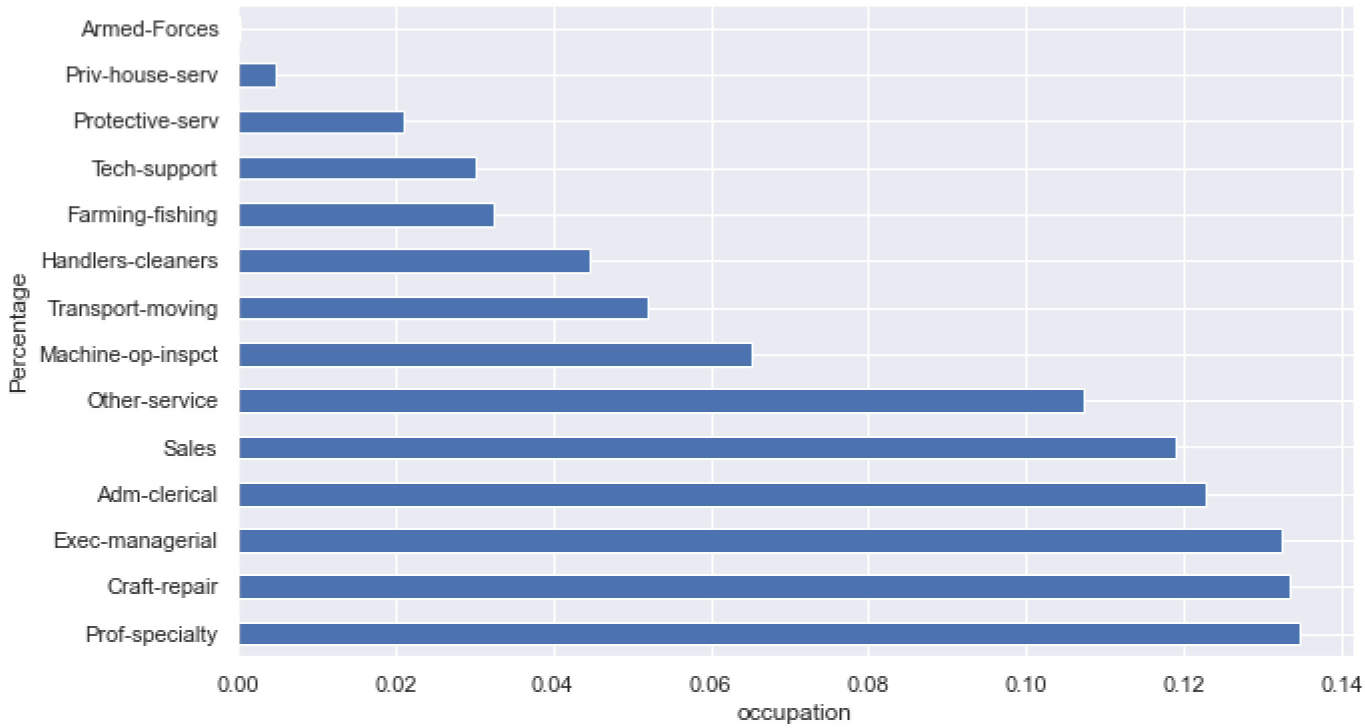
```
edu = df['native_country'].value_counts(normalize = True)
edu.mul(100).plot(kind='bar');
plt.xlabel("native_country")
plt.ylabel("Percentage")
plt.title = ('Country Distrubtion');
```



Observation - From the above chart we can see most of the people are american which make sense that majority of the people falls in white racial group.

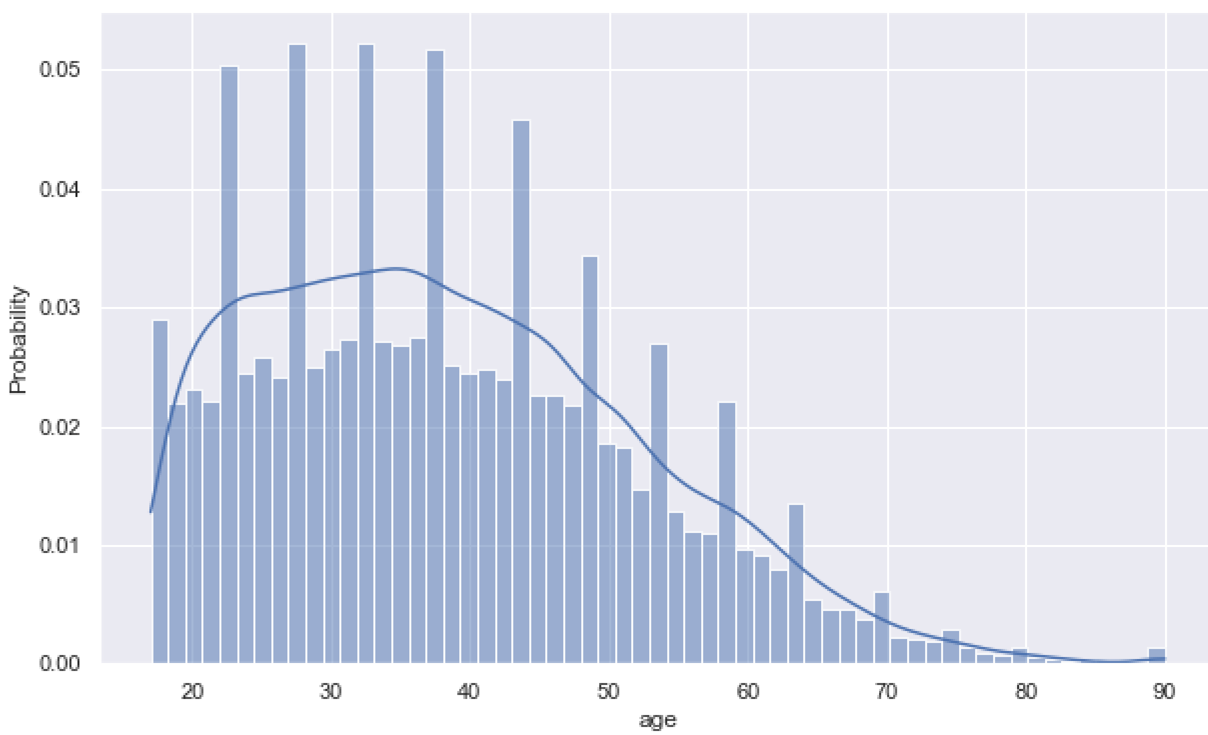
In [77]:

```
occua = df['occupation'].value_counts(normalize = True)
ax = occua.plot(kind='barh')
plt.xlabel("occupation")
plt.ylabel("Percentage")
plt.title = ('Occupation Distrubtion');
```



In [78]:

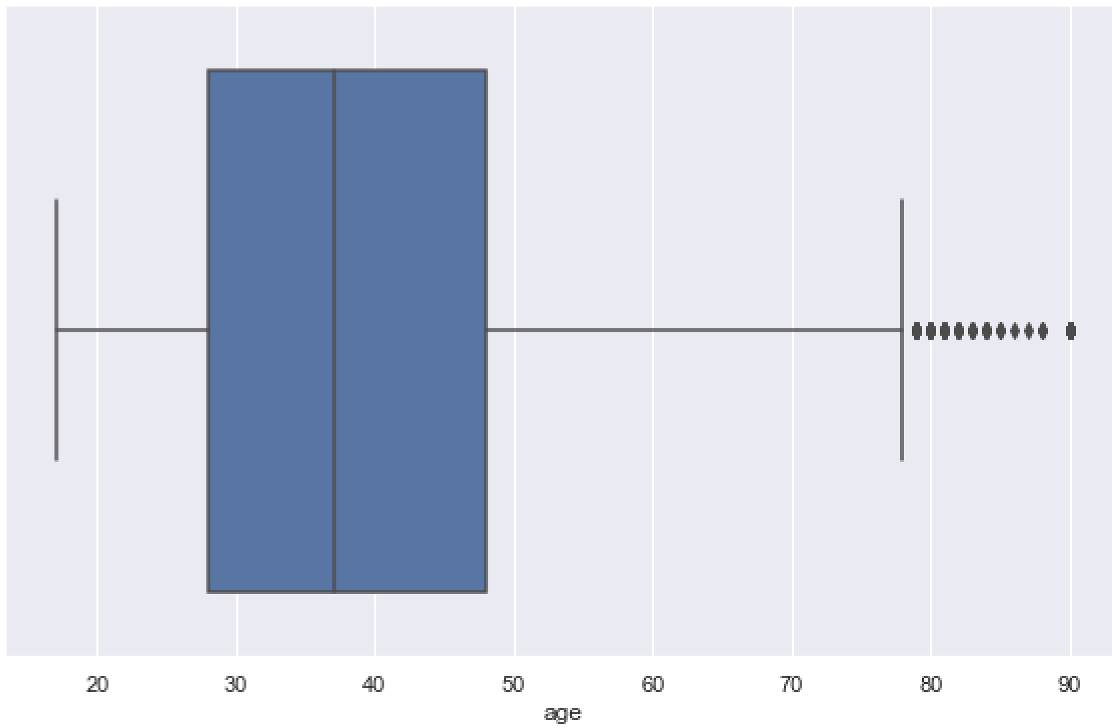
```
sns.histplot(df, x = df.age, stat = "probability", kde=True);
```



Observation - from the age distribution we can see that data is skewed to the right

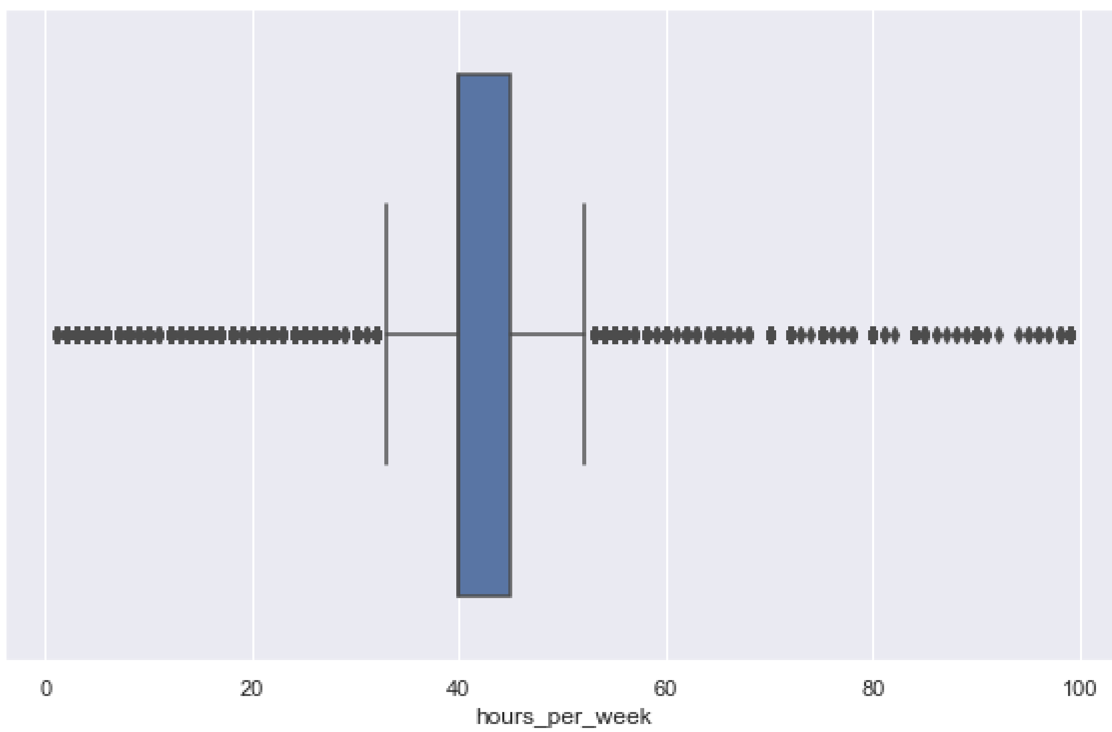
In [79]:

```
sns.boxplot(x=df.age);
```



Range of age is between 18 to 78 and median is 38.
Anything beyond 18 to 78 range are outliers so either we have to remove those values or Further investigation required

In [80]:
```
sns.boxplot(x=df.hours_per_week);
```

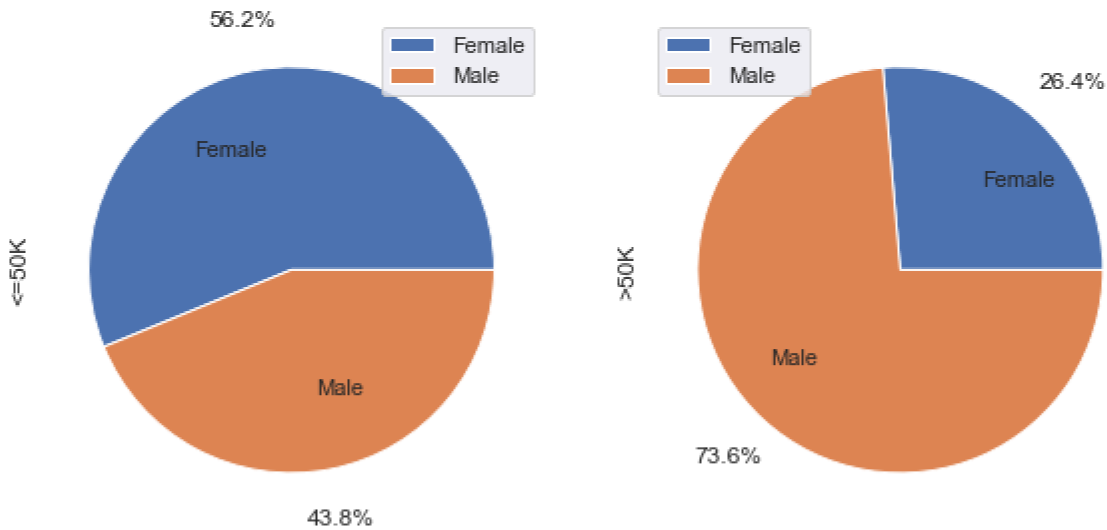


Hours_per_week range is 28 to 55
Anything beyond those range are outliers so either we have to remove those values or Further investigation required

Bivariate Distribution

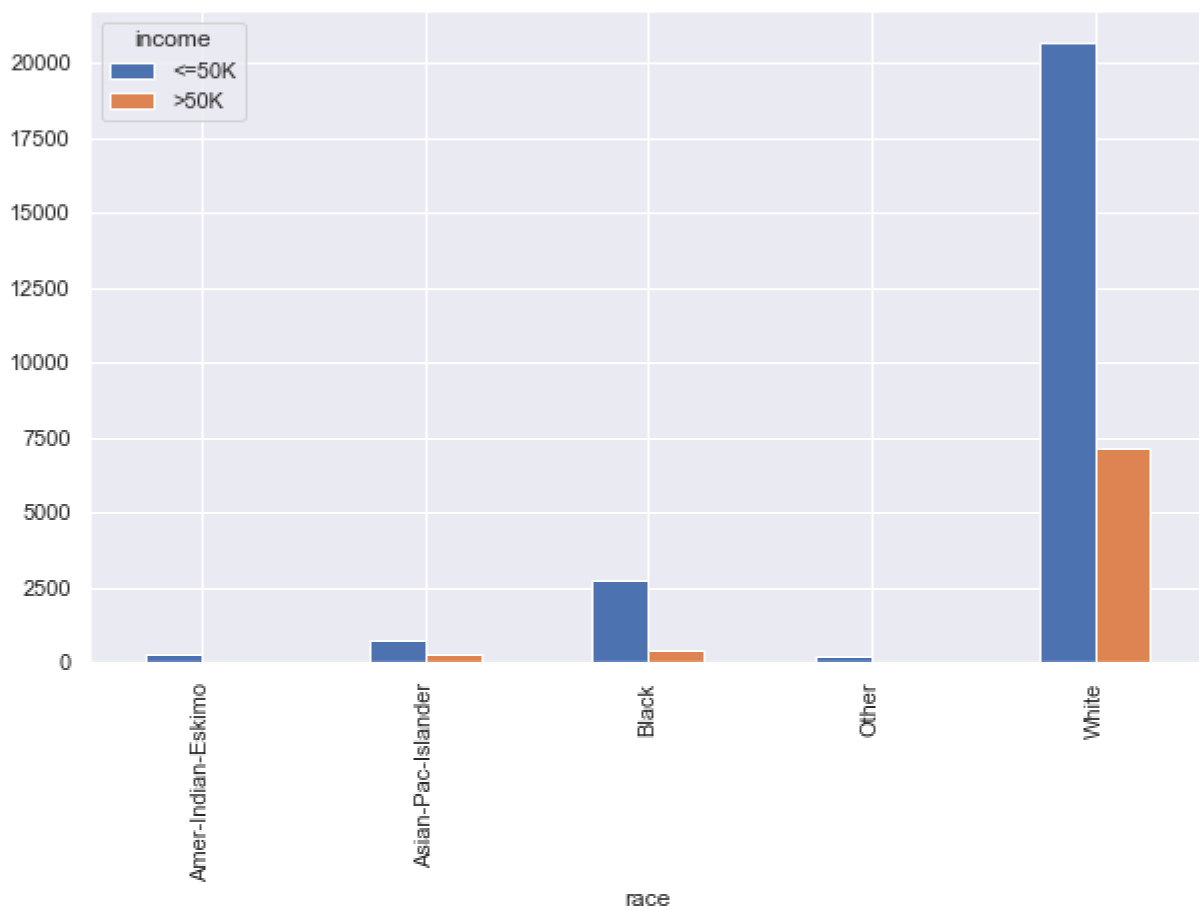
In [81]:

```
CrosstabResult=pd.crosstab(index=df.sex,columns=df.income, normalize='index')
CrosstabResult.mul(100).plot.pie(rot=0, subplots=True,autopct='%1.1f%%',pctdistance=1.25,
```

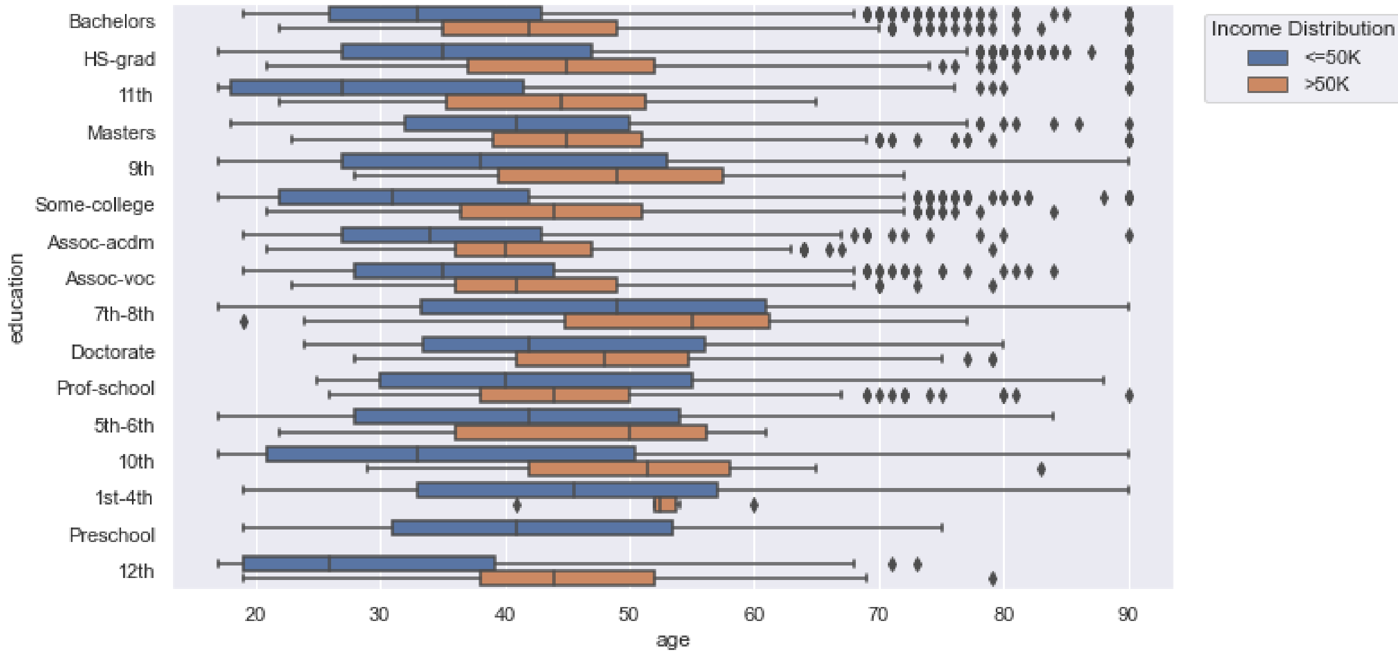


Observation - Male have dominated the chart of higher salary bracket with more than 73.6% population.

In [82]:
```
CrosstabResult=pd.crosstab(index=df.race,columns=df.income)
CrosstabResult.plot.bar(rot=90);
```



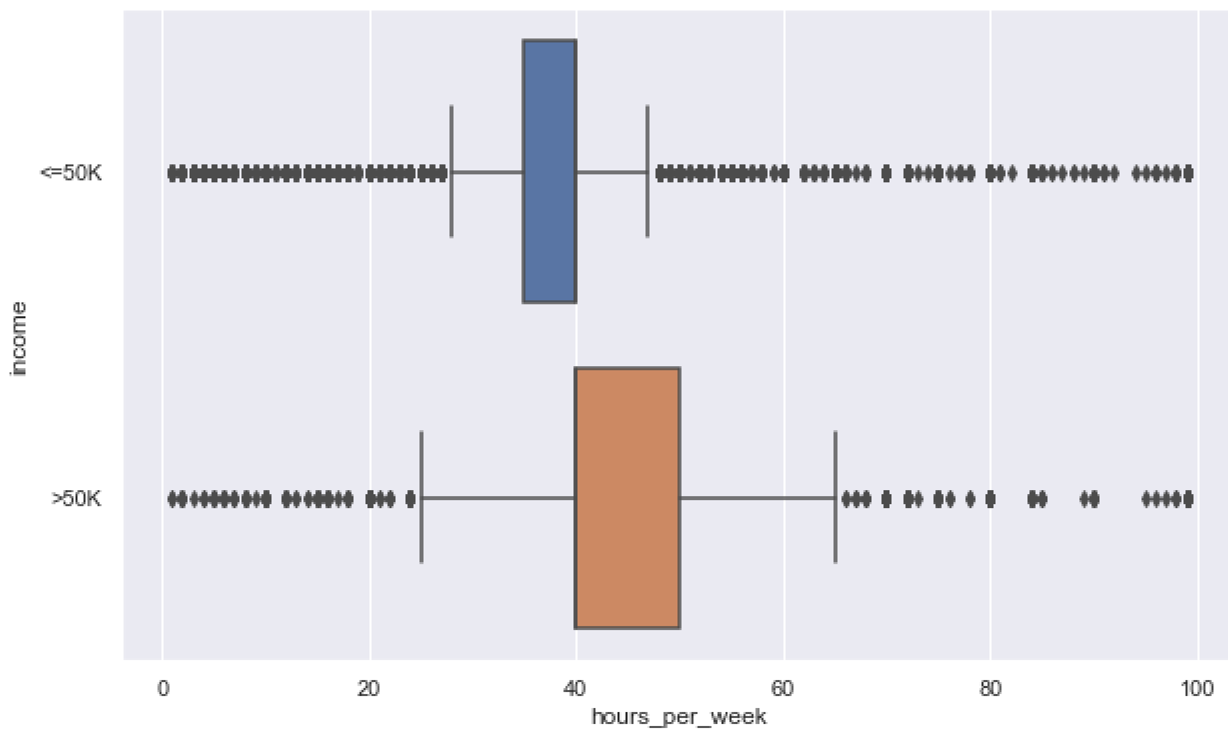
In [83]:
```
sns.boxplot(x=df.age,y=df.education, hue=df.income)
plt.legend(bbox_to_anchor=(1.02, 1),loc='upper left', title = 'Income Distribution');
```



Observation - from the graph it is clear that people who are less qualified have lower income as compared to their peers who are highly qualified.

In [84]:

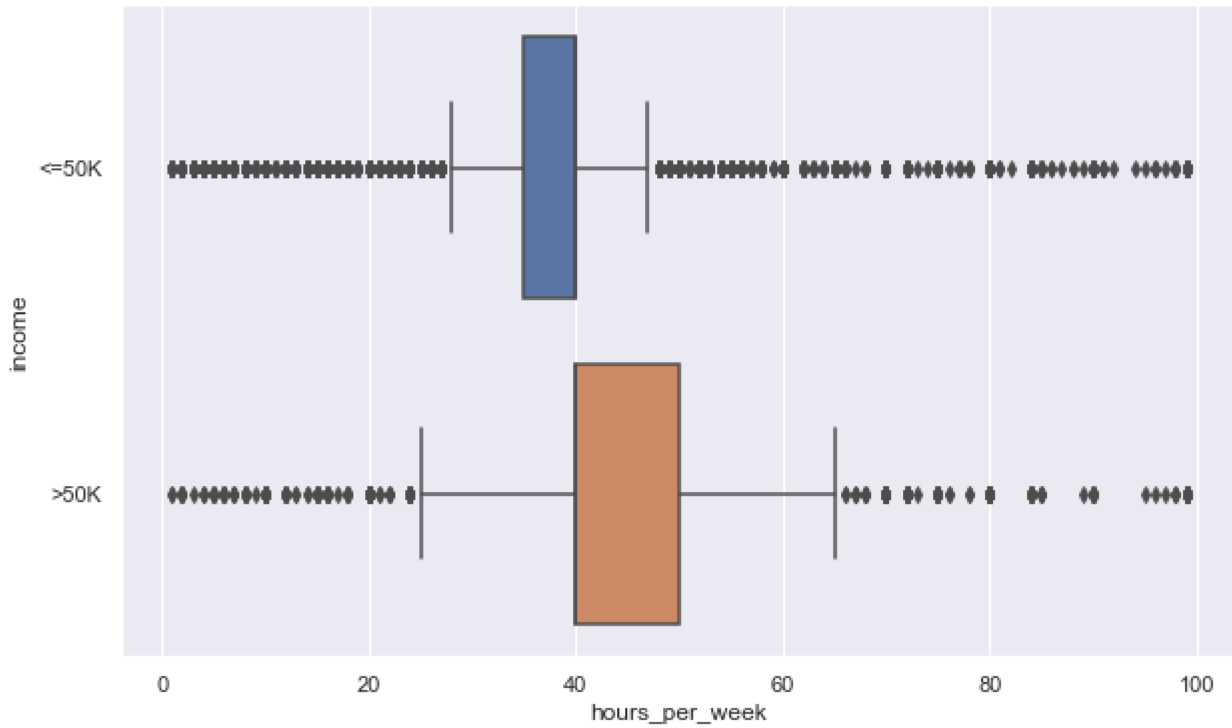
```
sns.boxplot(x= df.hours_per_week, y=df.income);
```



Observation - Its a clear trend that people who works more than 40hours/week have income higher than 50k and vice versa.

In [85]:

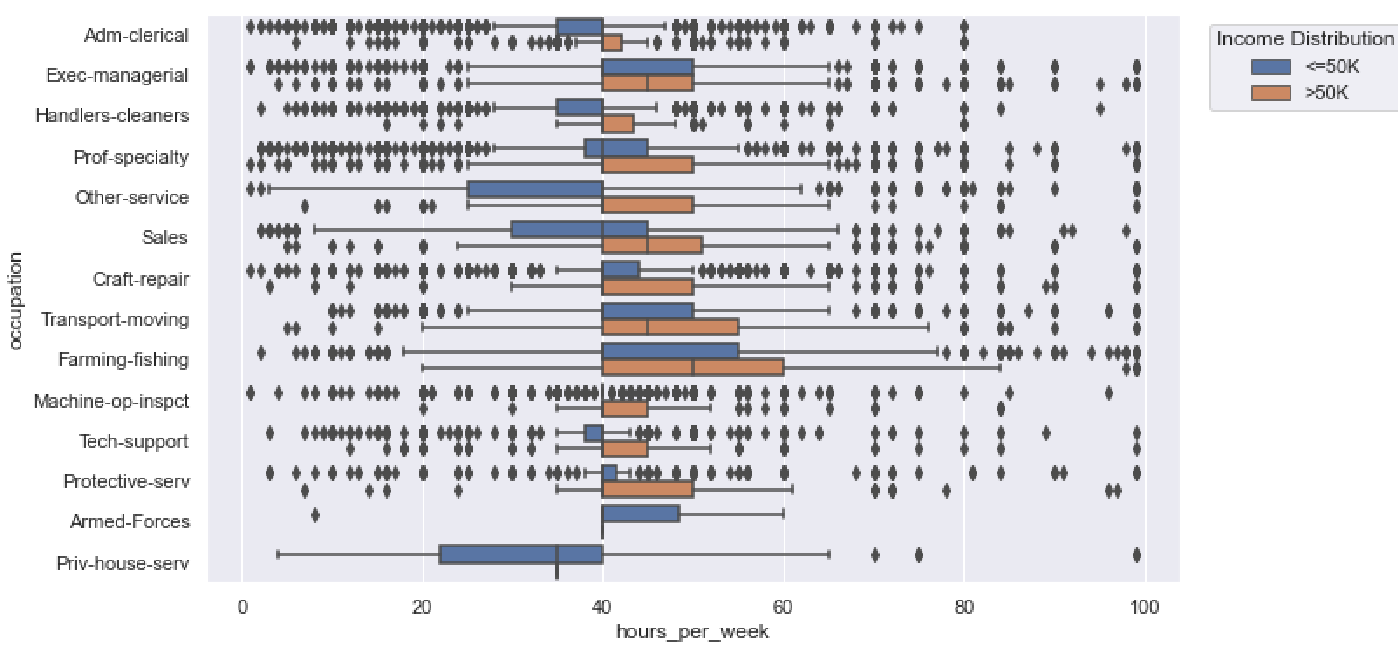
```
sns.boxplot(x= df.hours_per_week, y=df.income);
```



Observation - people with the median age of 33 earns less than 50k while people with the median age of 45 earns more than 50K. It Actually make sense more experience you are much higher the pay will be.

In [86]:

```
sns.boxplot(x=df.hours_per_week,y=df.occupation, hue=df.income)
plt.legend(bbox_to_anchor=(1.02, 1),loc='upper left', title = 'Income Distribution');
```



Observation - In all the occupation people have to work higher than 40hrs/week in order to earn higher income with some exceptions.

Conclusion - According to our data, if you want to make more than $50,000 annually, your age, hours worked each week, and qualifications all matter. Contrarily, the importance of the working class is little. In general, if you work hard enough, you will be given equal opportunities regardless of the type of job you do.

Data Quality:-

1. Missing values - In this data missing values are represented by ?. So, firstly we need to collect more data or One approach to handle missing data is to impute the missing values with appropriate values beacuse it is

categorical varaibles we can use mode method to fill the missing values(Baheti, P. (2023, February 2)).

2. Outliers - Using the boxplot we found the range of the numerical variables any values goes beyond that should either be trimmed or we need to use models which are not affected by extreme values or replacing extreme values with the next highest or lowest value in the distribution(Baheti, P. (2023, February 2)).
3. bias data - from the above analysis it is clear that dataset is biased in categories like sex, native_country and race. so to fix this bias is to collect more data in order to balance out biases or we can use data augmentation technique where we increase the diversity of the dataset by introducing new examples for under represented groups.
4. Reducing the sub-categoiries in the variable to reduce the complexity of the data and to get better visualisation.
5. Encoding categorical variables like '<=50k' & '>50k' to 0 & 1.
6. Data Completeness - if the dataset had more feature which could give better indication on income level such as number of years of experience, health condition or Postal code and etc.

Model which could best fit this dataset:
Our dataset has target variable which means we can apply supervised machine learning model.
As our independent variable is not linear to the independent variable we can not apply linear regression model.
But, beacuse our Target variable contains only two categories like '<=50k' & '>50k' which makes it suitable for logistic regression.
But before applying logistic regression, we need to first check for multicolinearity between the variables.
If there is any strong correlations between the variables our model will be greatly impacted.

Referencing: A. (2021, July 4). Adult Income Dataset | From Scratch. Kaggle.
https://www.kaggle.com/code/aditimulye/adult-income-dataset-from-scratch
P. (2020, March 4). EDA + Logistic Regression + PCA. Kaggle. https://www.kaggle.com/code/prashant111/eda-logistic-regression-pca
I. (2017, July 24). Income Prediction (84.369% Accuracy). Kaggle. https://www.kaggle.com/code/ipbyrne/income-prediction-84-369-accuracy
A. (2023, February 13). Eda📊|Feature_Engineering|📈Logistic_Regression. Kaggle.
https://www.kaggle.com/code/abhi011097/eda-feature-engineering-logistic-regression#1-|-Preprocessing-Steps
Baheti, P. (2023, February 2). A Simple Guide to Data Preprocessing in Machine Learning. V7.
https://www.v7labs.com/blog/data-preprocessing-guide