In [1]:

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

Import Data-Set

In [3]:

```
da=pd.read_csv(r"C:\Users\User\Downloads\adultmanisha.csv")
```

a.)Display Top 10 Rows of The Dataset

In [4]:

```
da.head(10)
```

Out[4]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	g
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	F
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White	
6	29	?	227026	HS-grad	9	Never-married	?	Unmarried	Black	
7	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	
8	24	Private	369667	Some-college	10	Never-married	Other-service	Unmarried	White	F
9	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White	

b.) Check Last 10 Rows of The Dataset

In [5]:

```
da.tail(10)
```

Out[5]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	
48832	32	Private	34066	10th	6	Married-civ-spouse	Handlers-cleaners	Husband	A In Es
48833	43	Private	84661	Assoc-voc	11	Married-civ-spouse	Sales	Husband	V
48834	32	Private	116138	Masters	14	Never-married	Tech-support	Not-in-family	A Isl
48835	53	Private	321865	Masters	14	Married-civ-spouse	Exec-managerial	Husband	V
48836	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	V
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	V
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	V
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	V
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	V
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	V

Find Shape,Size,Rows,Columns,Info about DataSet

In [8]:

```
 print(da.shape)
print(da.shape[0])
print(da.shape[1])
print(da.size)
print("\n")
print(da.info())
```

```
(48842, 15)
48842
15
732630


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   age              48842 non-null  int64 
 1   workclass        48842 non-null  object
 2   fnlwgt           48842 non-null  int64 
 3   education        48842 non-null  object
 4   educational-num  48842 non-null  int64 
 5   marital-status   48842 non-null  object
 6   occupation       48842 non-null  object
 7   relationship     48842 non-null  object
 8   race             48842 non-null  object
 9   gender           48842 non-null  object
 10  capital-gain     48842 non-null  int64 
 11  capital-loss     48842 non-null  int64 
 12  hours-per-week   48842 non-null  int64 
 13  native-country   48842 non-null  object
 14  income           48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
None
```

Fetch Random Sample From the Dataset (50%)

In [9]:

```
da.sample(frac=0.50)
```

Out[9]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	r
35133	47	Private	150768	Bachelors	13	Divorced	Handlers-cleaners	Not-in-family	W
30484	21	?	163911	Some-college	10	Never-married	?	Own-child	W
8512	38	Self-emp-inc	179579	Doctorate	16	Married-civ-spouse	Exec-managerial	Husband	W
21416	42	Private	219288	7th-8th	4	Widowed	Craft-repair	Unmarried	W
18625	44	Private	198282	Masters	14	Married-civ-spouse	Exec-managerial	Husband	W
...	...	...	...	...	...	...	...	...	
11060	41	Private	356934	Some-college	10	Married-civ-spouse	Tech-support	Husband	W
9132	37	Private	112264	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	W
18652	61	Local-gov	192060	Bachelors	13	Separated	Prof-specialty	Not-in-family	W
45745	52	Local-gov	236497	Bachelors	13	Married-civ-spouse	Tech-support	Husband	W
33990	39	Private	76417	Masters	14	Married-civ-spouse	Prof-specialty	Husband	W

24421 rows × 15 columns

Check Null Values In The Dataset

In [10]:

```
da.isna()
```

Out[10]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
0	False	False	False	False	False	False	False	False	Fal
1	False	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	Fal
3	False	False	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	Fal
...	...	...	...	...	...	...	...	...	
48837	False	False	False	False	False	False	False	False	Fal
48838	False	False	False	False	False	False	False	False	Fal
48839	False	False	False	False	False	False	False	False	Fal
48840	False	False	False	False	False	False	False	False	Fal
48841	False	False	False	False	False	False	False	False	Fal

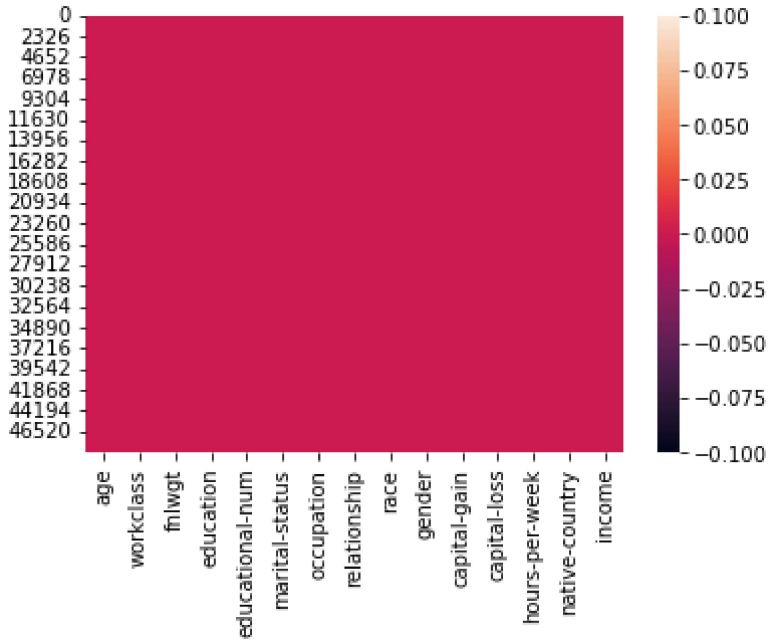
48842 rows × 15 columns

In [11]:

```
sns.heatmap(da.isnull())
```

Out[11]:

```
<AxesSubplot:>
```



It shows There is no null values there are ? instead of null values

Perform Data Cleaning [Replace '?' with NaN]

In [12]:

```
da.isin(["?"]).sum()
```

Out[12]:

```
age                  0
workclass         2799
fnlwgt               0
education            0
educational-num      0
marital-status       0
occupation        2809
relationship         0
race                 0
gender               0
capital-gain         0
capital-loss         0
hours-per-week       0
native-country     857
income               0
dtype: int64
```

In [13]:

```
da["workclass"]=da["workclass"].replace("?",np.nan)
da["occupation"]=da["occupation"].replace("?",np.nan)
da["native-country"]=da["native-country"].replace("?",np.nan)
```

In [14]:

```
da.isin(["?"]).sum()
```

Out[14]:

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

In [16]:

```
da.isin([np.nan]).sum()
```

Out[16]:

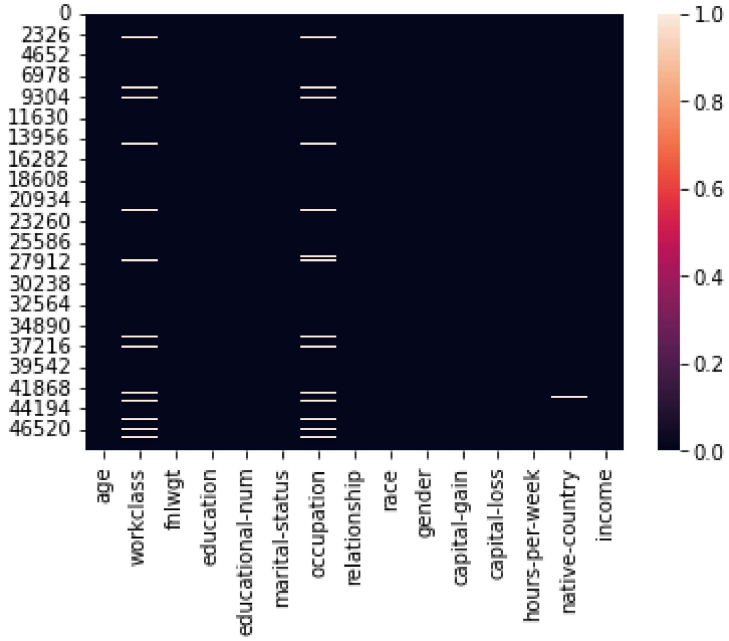
```
age                   0
workclass          2799
fnlwgt                0
education             0
educational-num       0
marital-status        0
occupation         2809
relationship          0
race                  0
gender                0
capital-gain          0
capital-loss          0
hours-per-week        0
native-country      857
income                0
dtype: int64
```

In [18]:

```
sns.heatmap(da.isnull())
```

Out[18]:

<AxesSubplot:>



Drop missing values

In [19]:

```
da.dropna(how="any",inplace=True)
```

In [20]:

```
da.shape
```

Out[20]:

```
(45222, 15)
```

Shape is decreased

Check for duplicate and drop them

In [22]:

```
du=da.duplicated().any()
du
```

Out[22]:

True

In [23]:

```
do=da.drop_duplicates()
```

In [24]:

```
do.shape
```

Out[24]:

(45175, 15)

Statistics

In [25]:

```
da.describe()
```

Out[25]:

	age	fnlwgt	educational-num	capital-gain	capital-loss	hours-per-week
count	45222.000000	4.522200e+04	45222.000000	45222.000000	45222.000000	45222.000000
mean	38.547941	1.897347e+05	10.118460	1101.430344	88.595418	40.938017
std	13.217870	1.056392e+05	2.552881	7506.430084	404.956092	12.007508
min	17.000000	1.349200e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.173882e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783160e+05	10.000000	0.000000	0.000000	40.000000
75%	47.000000	2.379260e+05	13.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

Drop the columns capital-gain,capital-loss

In [26]:

```
da.drop(["capital-gain",'capital-loss'],axis=1)
```

Out[26]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Bla
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Wh
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Wh
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Bla
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	Wh
...	...	...	...	...	...	...	...	...	
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	Wh
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Wh
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	Wh
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Wh
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Wh

45222 rows × 13 columns

Univariate Analysis

Taking one variable at a time

What is the distribution of Age column

In [27]:

```
da["age"].describe()
```

Out[27]:

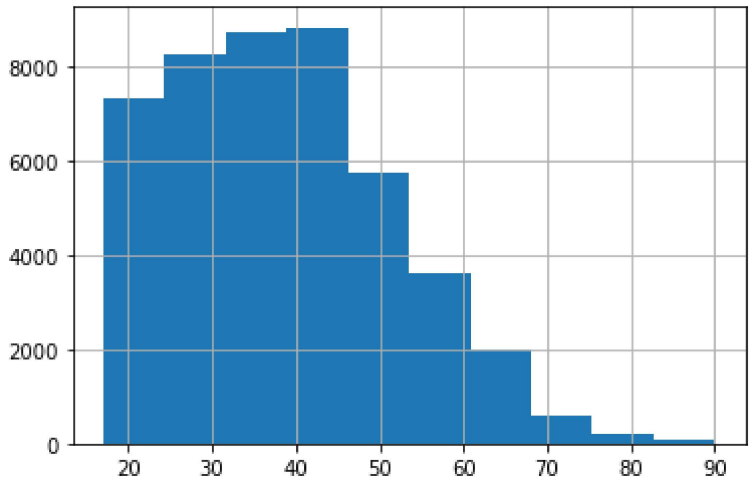
```
count    45222.000000
mean        38.547941
std         13.217870
min         17.000000
25%         28.000000
50%         37.000000
75%         47.000000
max         90.000000
Name: age, dtype: float64
```

In [37]:

```
da["age"].hist()
```

Out[37]:

```
<AxesSubplot:>
```



Find Total Number of Persons Having Age Between 17 To 48 (Inclusive) Using Between Method

In [32]:

```
t=da[da["age"].between(17,48)]
t["age"].sum()
```

Out[32]:

```
1150833
```

What is The Distribution of Workclass Column?

In [33]:

```
da["workclass"].describe()
```

Out[33]:

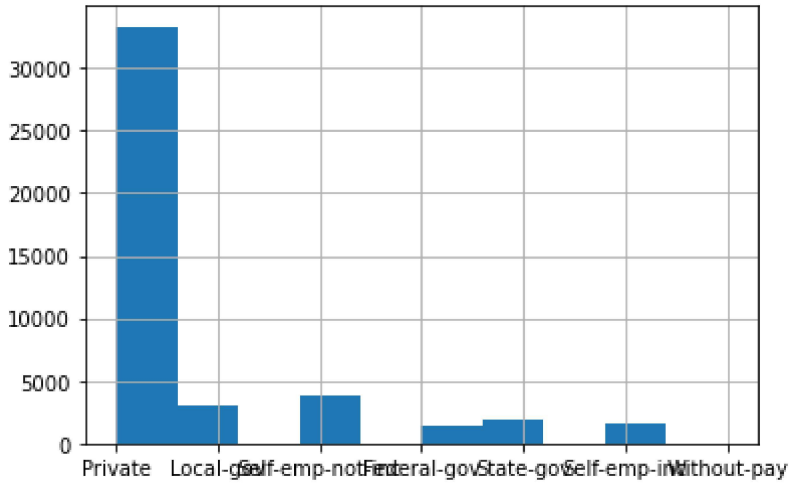
```
count       45222
unique          7
top       Private
freq        33307
Name: workclass, dtype: object
```

In [35]:

```
da["workclass"].hist()
#overlapping change the figure size
```

Out[35]:

```
<AxesSubplot:>
```

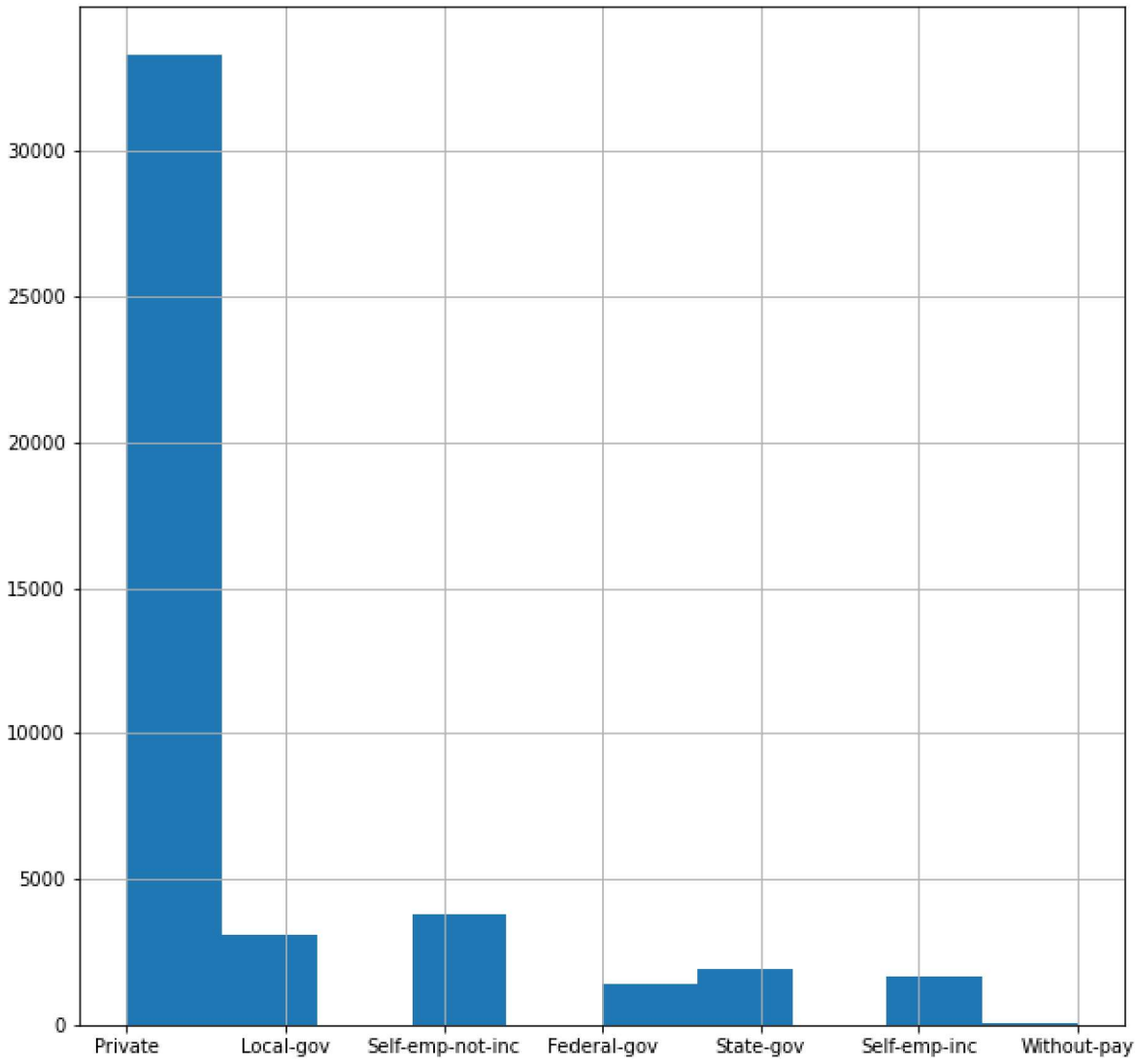


In [43]:

```
plt.figure(figsize=(10,10))
da["workclass"].hist()
```

Out[43]:

<AxesSubplot:>



How Many Persons Having Bachelors and Masters Degree?

In [45]:

```
f=da["education"]=="Bachelors"
g=da["education"]=="Masters"
y=da[f|g]
y["education"].count()
```

Out[45]:

```
10084
```

Bivariate Analysis

Relationship between two Variables

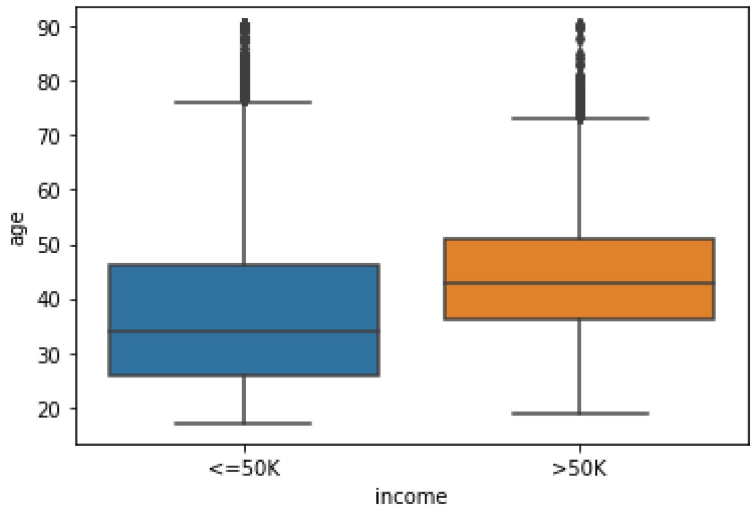
Replace Salary(<=50 k ,>=50k) Values With 0 and 1

In [46]:

```
sns.boxplot(x="income",y="age",data=da)
```

Out[46]:

```
<AxesSubplot:xlabel='income', ylabel='age'>
```



In [47]:

```
da["income"].unique()
```

Out[47]:

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

In [48]:

```
da["income"].value_counts()
```

Out[48]:

```
<=50K    34014
>50K     11208
Name: income, dtype: int64
```

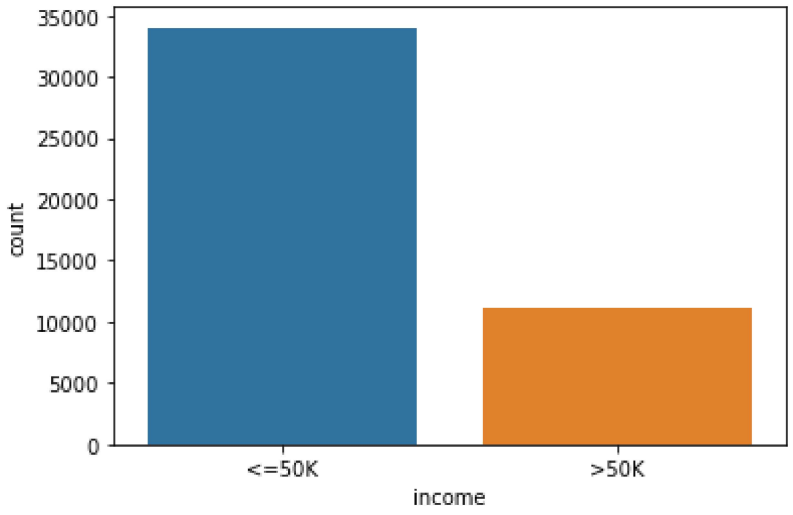
In [49]:

```
sns.countplot("income",data=da)
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[49]:

```
<AxesSubplot:xlabel='income', ylabel='count'>
```



In [50]:

```
da.replace(to_replace=['<=50K', '>50K'],value=[0,1],inplace=True)
```

In [51]:

```
da
```

Out[51]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	ra
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Bla
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	Wh
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	Wh
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Bla
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	Wh
...	...	...	...	...	...	...	...	...	
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	Wh
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	Wh
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	Wh
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Wh
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Wh

45222 rows × 15 columns

Which Workclass Getting The Highest Salary?

In [52]:

```
da.groupby("workclass")["income"].mean().sort_values(ascending=False)
```

Out[52]:

```
workclass
Self-emp-inc        0.554070
Federal-gov         0.390469
Local-gov           0.295161
Self-emp-not-inc    0.278978
State-gov           0.267215
Private             0.217702
Without-pay         0.095238
Name: income, dtype: float64
```

How Has Better Chance To Get Salary greater than 50K Male or Female?

In [53]:

```
da.groupby("gender")["income"].mean().sort_values(ascending=False)
```

Out[53]:

```
gender
Male      0.312477
Female    0.113576
Name: income, dtype: float64
```

Covert workclass Columns Datatype To Category Datatype

In [54]:

```
da.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45222 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   age              45222 non-null  int64 
 1   workclass        45222 non-null  object
 2   fnlwgt           45222 non-null  int64 
 3   education        45222 non-null  object
 4   educational-num  45222 non-null  int64 
 5   marital-status   45222 non-null  object
 6   occupation       45222 non-null  object
 7   relationship     45222 non-null  object
 8   race             45222 non-null  object
 9   gender           45222 non-null  object
 10  capital-gain     45222 non-null  int64 
 11  capital-loss     45222 non-null  int64 
 12  hours-per-week   45222 non-null  int64 
 13  native-country   45222 non-null  object
 14  income           45222 non-null  int64 
dtypes: int64(7), object(8)
memory usage: 6.5+ MB
```

In [56]:

```
da["workclass"]=da["workclass"].astype("category")
```

In [57]:

```
da.info()
```

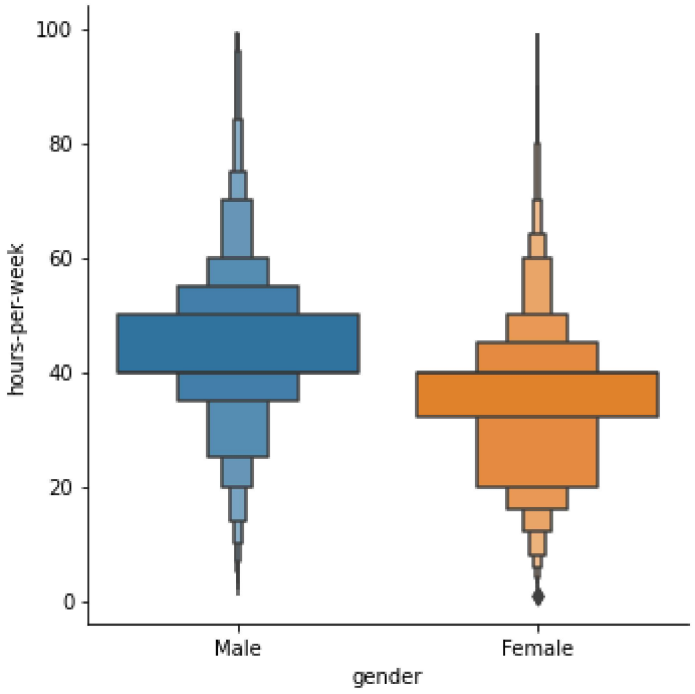
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 45222 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype   
---  ------           --------------  -----   
 0   age              45222 non-null  int64   
 1   workclass        45222 non-null  category
 2   fnlwgt           45222 non-null  int64   
 3   education        45222 non-null  object  
 4   educational-num  45222 non-null  int64   
 5   marital-status   45222 non-null  object  
 6   occupation       45222 non-null  object  
 7   relationship     45222 non-null  object  
 8   race             45222 non-null  object  
 9   gender           45222 non-null  object  
 10  capital-gain     45222 non-null  int64   
 11  capital-loss     45222 non-null  int64   
 12  hours-per-week   45222 non-null  int64   
 13  native-country   45222 non-null  object  
 14  income           45222 non-null  int64   
dtypes: category(1), int64(7), object(7)
memory usage: 6.2+ MB
```

In [59]:

```
sns.catplot(x="gender",y="hours-per-week",data=da,kind="boxen")
```

Out[59]:

```
<seaborn.axisgrid.FacetGrid at 0x1944b631e80>
```

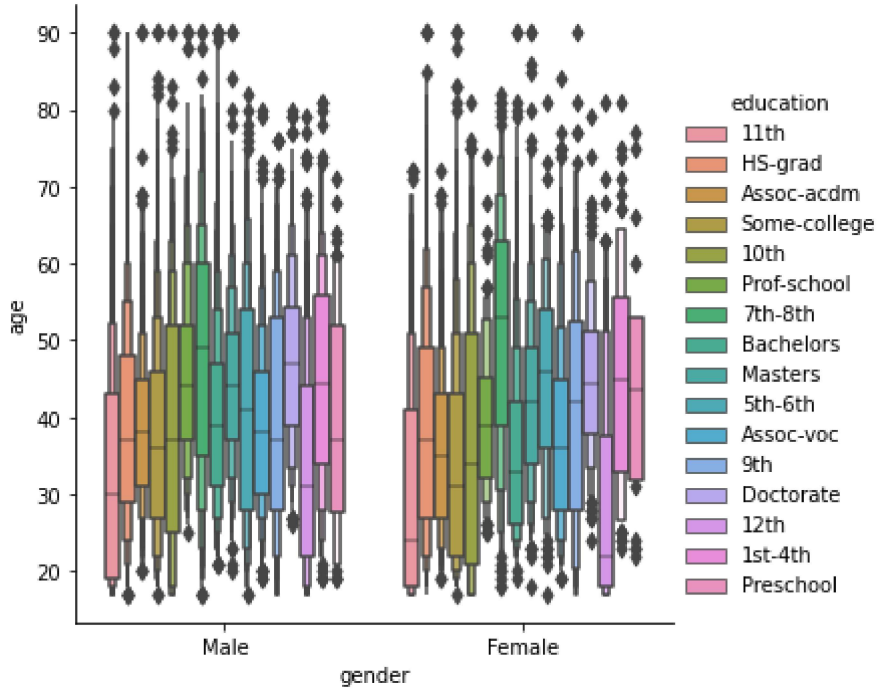


In [60]:

```
sns.catplot(x="gender",y="age",data=da,hue="education",kind="boxen")
```

Out[60]:

```
<seaborn.axisgrid.FacetGrid at 0x19444e8ebb0>
```

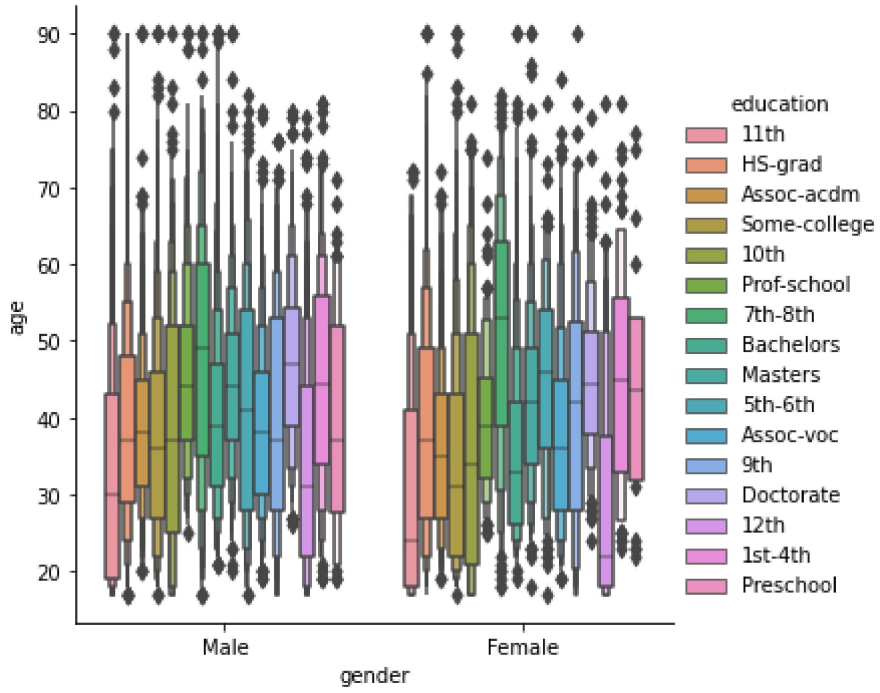


In [61]:

```
sns.catplot(x="gender",y="age",data=da,hue="education",kind="boxen")
```

Out[61]:

```
<seaborn.axisgrid.FacetGrid at 0x19444f0ffa0>
```

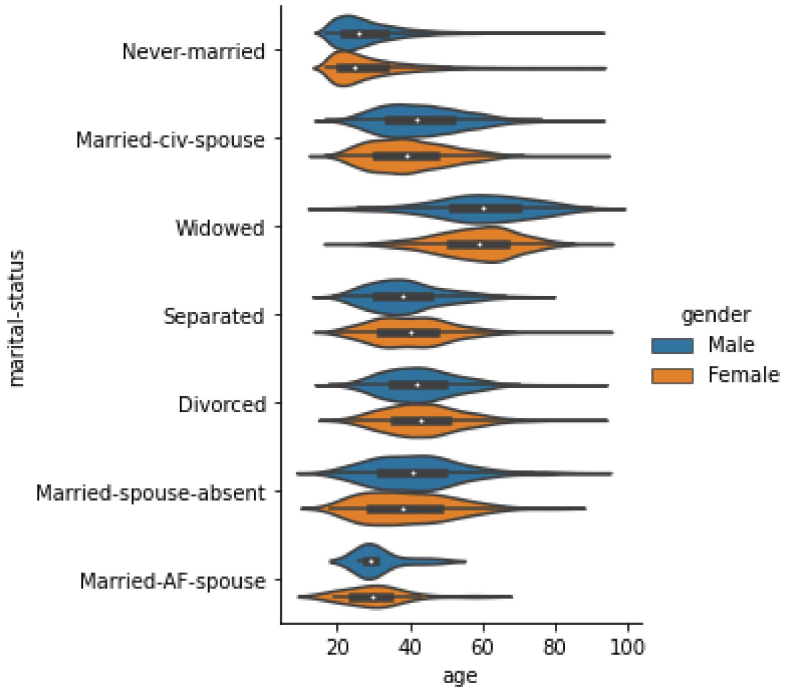


In [62]:

```
sns.catplot(x="age",y="marital-status",data=da,hue="gender",kind="violin")
```

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x19444ec78e0>
```



In [ ]: