INCOME_QUESTIONS/ANSWERED_BASED_EDA_PROJECT_VIVEK_CHA

In [1]:
```
# upload the necessary libraries to work with dataset

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# forward / is used for path.
data=pd.read_csv("C:/Users/VIVEK CHAUHAN/Desktop/eda-projects (1)/3-EDA Problem Statement (1)/3-income (1).csv")
data
```

Out[2]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	na co
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	U
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	U
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	U
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	U
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	U
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	U
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	U
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	U
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	U
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	U

48842 rows × 15 columns

We will learn to use all the below functions in this Income EDA Projects

How to fetch random samples from the Dataset ?

isin

between

unique

dropna

replace

duplicated

drop_duplicates

astype

apply

What is Univariate Analysis ?

What is Bivariate Analysis ?

Memory Optimization

In [3]:
```
# display top 10 rows of the dataset

data.head(10)
```

Out[3]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	30	United-States
6	29	?	227026	HS-grad	9	Never-married	?	Unmarried	Black	Male	0	0	40	United-States
7	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	3103	0	32	United-States
8	24	Private	369667	Some-college	10	Never-married	Other-service	Unmarried	White	Female	0	0	40	United-States
9	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	10	United-States

In [4]:
```
# display bottom 10 rows of the dataset

data.tail(10)
```

Out[4]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week
48832	32	Private	34066	10th	6	Married-civ-spouse	Handlers-cleaners	Husband	Amer-Indian-Eskimo	Male	0	0	40
48833	43	Private	84661	Assoc-voc	11	Married-civ-spouse	Sales	Husband	White	Male	0	0	45
48834	32	Private	116138	Masters	14	Never-married	Tech-support	Not-in-family	Asian-Pac-Islander	Male	0	0	11
48835	53	Private	321865	Masters	14	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40
48836	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	White	Male	0	0	40
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40

In [5]:
```
# find shape of our dataset(number of rows & columns)

data.shape
```

Out[5]:
```
(48842, 15)
```

In [6]:
```
# get the number of rows & columns in the dataset

print("Number of Rows", data.shape[0])
print("Number of Columns", data.shape[1])
```

```
Number of Rows 48842
Number of Columns 15
```

In [7]:
```
# Getting information about our dataset like total number of rows,
# columns, datatype of each column & memory requirement

data.info()
```

```
<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
```

In [8]:
```
# Fetch Random Samples from the dataset (50%)

data1 = data.sample(frac = 0.50,random_state = 100)
data1
```

Out[8]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	na cou
12393	37	Private	110331	Prof-school	15	Married-civ-spouse	Other-service	Wife	White	Female	0	0	60	Ur S
48701	23	Private	45834	Bachelors	13	Never-married	Exec-managerial	Not-in-family	White	Female	0	0	50	Ur S
17918	28	Private	89718	HS-grad	9	Never-married	Sales	Not-in-family	White	Female	2202	0	48	Ur S
11352	30	Private	351770	9th	5	Divorced	Other-service	Unmarried	White	Female	0	0	38	Ur S
36198	31	Private	164190	10th	6	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	40	Ur S
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48573	41	Private	318046	Some-college	10	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	48	Ur S
47252	41	Local-gov	33658	Some-college	10	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	45	Ur S
33142	69	Private	312653	Some-college	10	Married-civ-spouse	Sales	Husband	White	Male	0	0	25	Ur S
2965	21	?	334593	Some-college	10	Never-married	?	Not-in-family	White	Male	0	0	40	Ur S
32089	34	Private	186269	HS-grad	9	Divorced	Adm-clerical	Own-child	White	Male	0	0	40	Ur S

24421 rows × 15 columns

In [9]:
```
# check null values in the dataset

data.isnull()
```

Out[9]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	nat cou
0	False	False	False	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	False	False	False	False	False	False	False	False	False	False	False	False	False	F
48838	False	False	False	False	False	False	False	False	False	False	False	False	False	F
48839	False	False	False	False	False	False	False	False	False	False	False	False	False	F
48840	False	False	False	False	False	False	False	False	False	False	False	False	False	F
48841	False	False	False	False	False	False	False	False	False	False	False	False	False	F

48842 rows × 15 columns

In [10]:
```
# let's count the sum of our how many nulls are present in our dataset column wise

data.isnull().sum()
```

Out[10]:
```
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
```

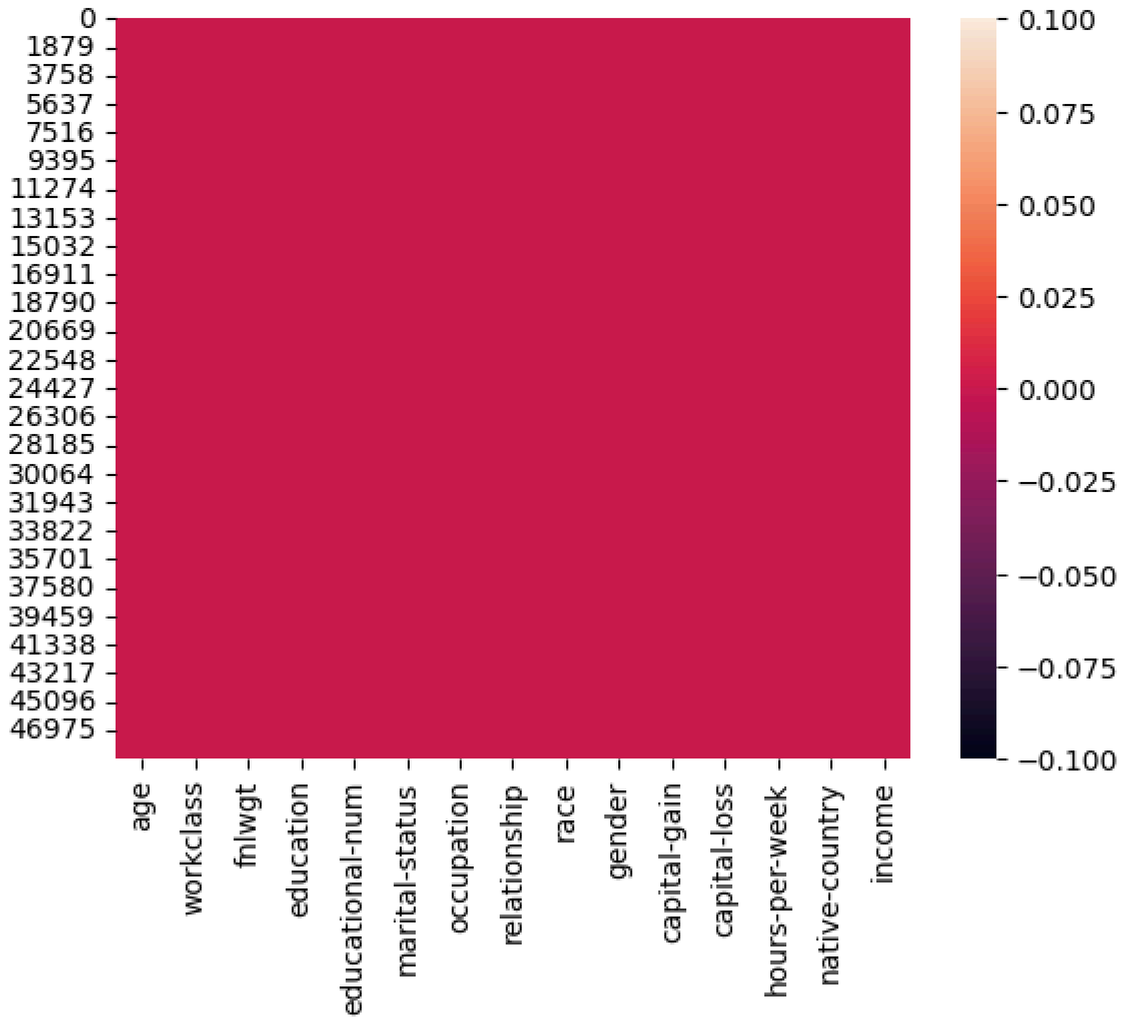
In [11]:
```
# let's count the sum of our how many nulls are present in our dataset row wise

data.isnull().sum(axis=0)
```

Out[11]:
```
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 [12]:
```
# we are using heatmap as it will show lighter colour if there is any missing values
```

```
sns.heatmap(data.isnull())
```

Out[12]: <Axes: >



In [13]:
```
# sum of ? in the dataset column wise

data.isin(["?"]).sum().sort_values(ascending = False)
```

Out[13]:
```
occupation         2809
workclass          2799
native-country      857
age                   0
fnlwgt                0
education             0
educational-num       0
marital-status        0
relationship          0
race                  0
gender                0
capital-gain          0
capital-loss          0
hours-per-week        0
income                0
dtype: int64
```

In [14]:
```
# perform data cleaning [replace "?" with Nan

data.replace("?",np.NaN,inplace=True)
data
```

Out[14]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	na co
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	U
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	U
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	U
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	U
4	18	NaN	103497	Some-college	10	Never-married	NaN	Own-child	White	Female	0	0	30	U
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	U
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	U
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	U
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	U
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	U

48842 rows × 15 columns

In [15]:
```
# drop all missing values

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

In [16]:
```
# check whether it has duplicate data or not

dup = data.duplicated().any()
dup
```

Out[16]: True

In [17]:
```
# first of all drop the duplicated data

data = data.drop_duplicates()
```

In [18]:
```
# print the shape of hte dataset after removing duplicates values

data.shape
```

Out[18]: (45175, 15)

In [19]:
```
# get overall statistics about the dataframe

data.describe()
```

Out[19]:

	age	fnlwgt	educational-num	capital-gain	capital-loss	hours-per-week
count	45175.000000	4.517500e+04	45175.000000	45175.000000	45175.000000	45175.000000
mean	38.556170	1.897388e+05	10.119314	1102.576270	88.687593	40.942512
std	13.215349	1.056524e+05	2.551740	7510.249876	405.156611	12.007730
min	17.000000	1.349200e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.173925e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783120e+05	10.000000	0.000000	0.000000	40.000000
75%	47.000000	2.379030e+05	13.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

In [20]:

```
# drop the columns education-num,capital gain, & capital loss

a = data.drop(['educational-num','capital-gain','capital-loss'],axis=1)
a
```

Out[20]:

	age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	gender	hours-per-week	native-country	income
0	25	Private	226802	11th	Never-married	Machine-op-inspct	Own-child	Black	Male	40	United-States	<=50K
1	38	Private	89814	HS-grad	Married-civ-spouse	Farming-fishing	Husband	White	Male	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	Married-civ-spouse	Protective-serv	Husband	White	Male	40	United-States	>50K
3	44	Private	160323	Some-college	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	40	United-States	>50K
5	34	Private	198693	10th	Never-married	Other-service	Not-in-family	White	Male	30	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...
48837	27	Private	257302	Assoc-acdm	Married-civ-spouse	Tech-support	Wife	White	Female	38	United-States	<=50K
48838	40	Private	154374	HS-grad	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	40	United-States	>50K
48839	58	Private	151910	HS-grad	Widowed	Adm-clerical	Unmarried	White	Female	40	United-States	<=50K
48840	22	Private	201490	HS-grad	Never-married	Adm-clerical	Own-child	White	Male	20	United-States	<=50K
48841	52	Self-emp-inc	287927	HS-grad	Married-civ-spouse	Exec-managerial	Wife	White	Female	40	United-States	>50K

45175 rows × 12 columns

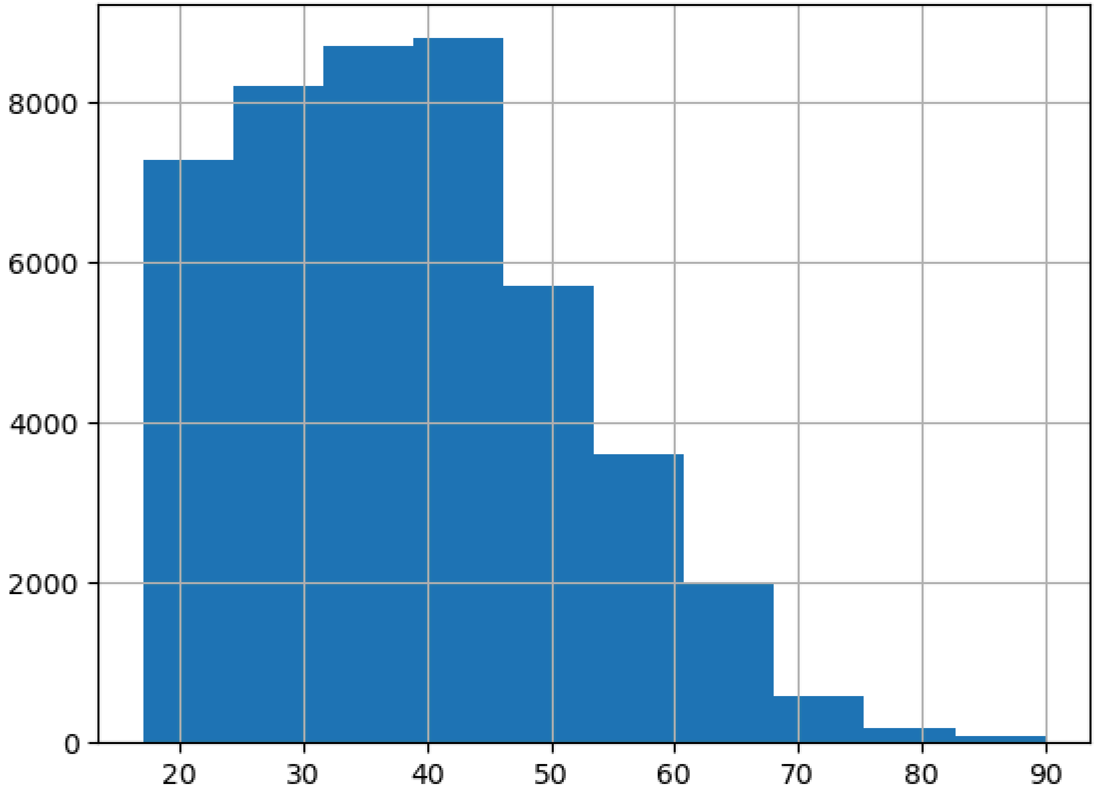
Univariate Analysis (means we will take one variable at a time and perform some analysis on it)

In [21]:
```
# what is the distribution of age column ?

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

Out[21]:
```
count    45175.000000
mean        38.556170
std         13.215349
min         17.000000
25%         28.000000
50%         37.000000
75%         47.000000
max         90.000000
Name: age, dtype: float64
```

In [22]:
```
# histogram of the age column

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

Out[22]:
```
<Axes: >
```



In [23]:
```
# other method to get the total number of persons have age between 17 to 48 (inclusive) using between method

sum((data["age"]>=17) & (data["age"]<=48))
```

Out[23]: 34858

In [24]:
```
# another method to find the # other method to get the total number of persons have age between 17 to 48 (inclusive) using bet

sum(data['age'].between(17,48))
```

Out[24]: 34858

In [25]:
```
# Find the total number of persons have age between 17 to 48 (inclusive) using between method

a = data["age"]>=17
```

```
b= data["age"]<=48

ans = data.where(a & b)
ans
```

Out[25]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	c
0	25.0	Private	226802.0	11th	7.0	Never-married	Machine-op-inspct	Own-child	Black	Male	0.0	0.0	40.0	
1	38.0	Private	89814.0	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	White	Male	0.0	0.0	50.0	
2	28.0	Local-gov	336951.0	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	White	Male	0.0	0.0	40.0	
3	44.0	Private	160323.0	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688.0	0.0	40.0	
5	34.0	Private	198693.0	10th	6.0	Never-married	Other-service	Not-in-family	White	Male	0.0	0.0	30.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	27.0	Private	257302.0	Assoc-acdm	12.0	Married-civ-spouse	Tech-support	Wife	White	Female	0.0	0.0	38.0	
48838	40.0	Private	154374.0	HS-grad	9.0	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0.0	0.0	40.0	
48839	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
48840	22.0	Private	201490.0	HS-grad	9.0	Never-married	Adm-clerical	Own-child	White	Male	0.0	0.0	20.0	
48841	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

45175 rows × 15 columns

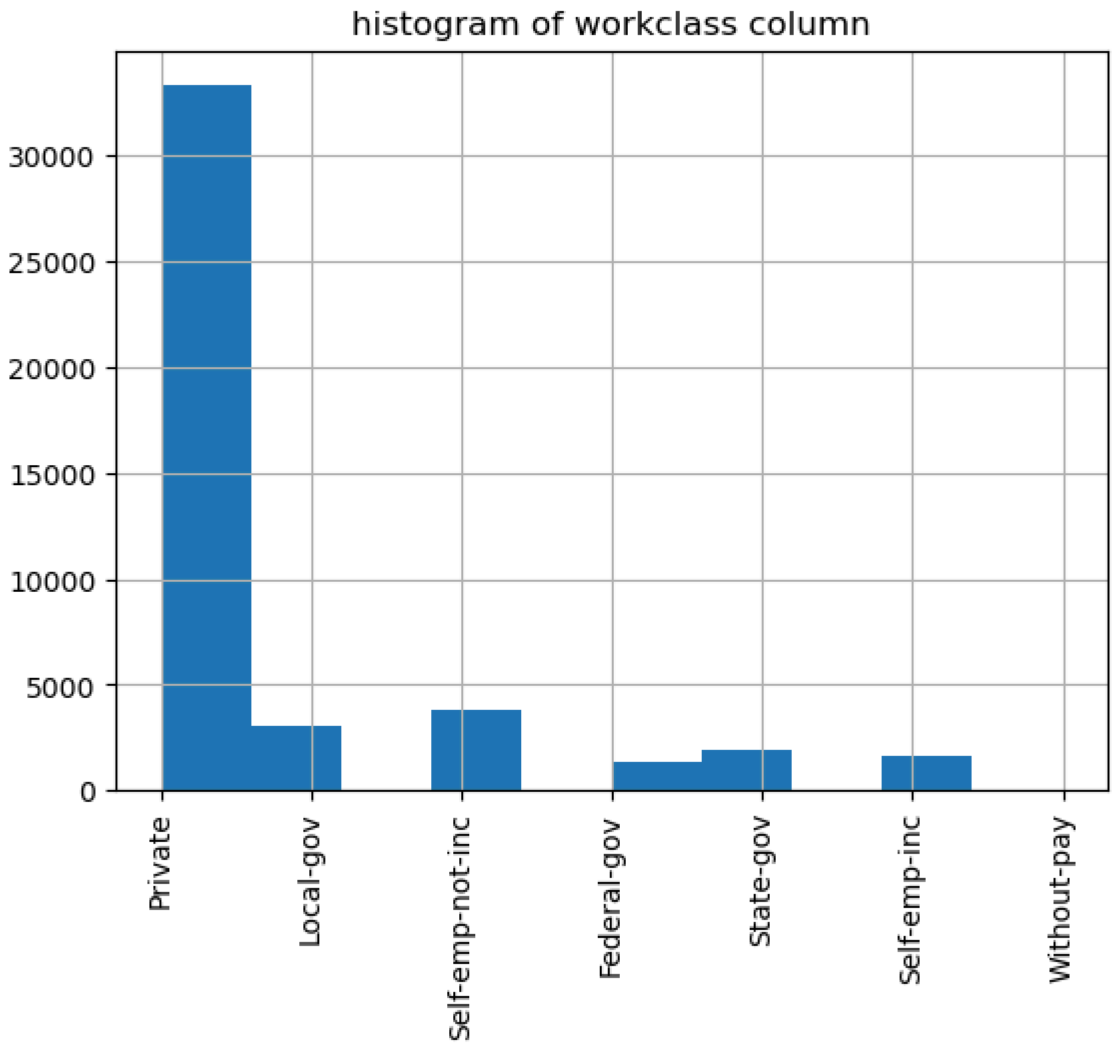
In [26]:
```
# other method to get sum of age between 17 to 48

sum(data['age'].between(17,48))
```

Out[26]:
```
34858
```

In [27]:
```
# what is the distribution of the workclass column

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

Out[27]:
```
count       45175
unique          7
top       Private
freq        33262
Name: workclass, dtype: object
```

In [28]:
```
# histogram of our workclass column

plt.title("histogram of workclass column")
data['workclass'].hist()
plt.xticks(rotation = 90)
plt.show()
```



In [29]:
```
# how many persons have bachelor's and marster's degree?

stat1 = data["education"]=="Bachelors"
ans1 = stat1.sum()
ans1
```

Out[29]: 7559

In [30]:
```
# how many persons have bachelor's and marster's degree?

stat1 = data["education"]=="Masters"
ans2 = stat1.sum()
ans2
```

Out[30]: 2513

In [31]:
```
# total persons that have bachelor's and marster's degree?

total_persons = ans1 + ans2
total_persons
```

Out[31]: 10072

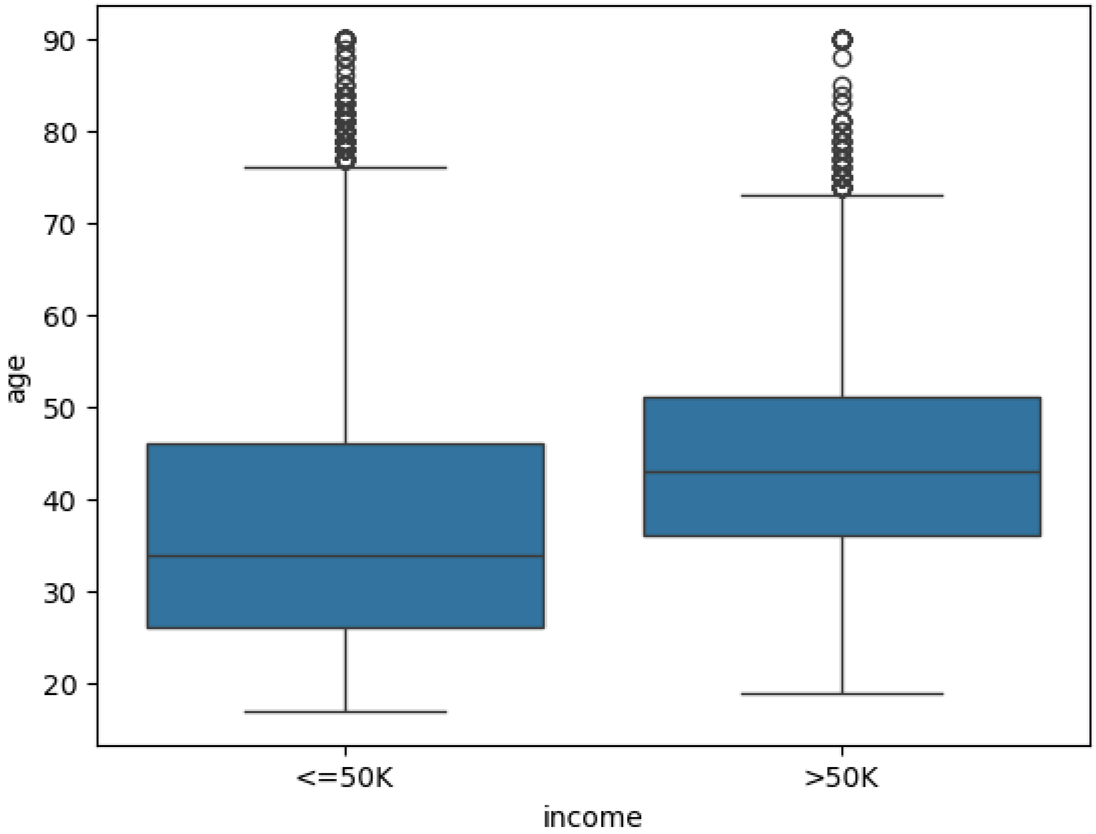
In [32]:
```
# other method to get how many persons have bachelor's and marster's degree?

sum(data["education"].isin(["Bachelors","Masters"]))
```

Out[32]: 10072

In [33]:
```
# Bivariate Analysis (we will use bivraiate to find relationship between 2 different variables)
```

In [34]:
```
# income vs age boxplot

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

Out[34]: <Axes: xlabel='income', ylabel='age'>



In [35]:
```
# so as we can see above most of the people are younger having salary about less than or equal to 50k &
# most of the people are aged having salaru above 50k
```

In [36]:
```
# Replace Salary Values['<=50k','>50k'] with 0 & 1

data.replace("<=50k",0,inplace=True)
data
```

Out[36]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	na co
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	U
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	U
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	U
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	U
5	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White	Male	0	0	30	U
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	U
48838	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	U
48839	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	U
48840	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	U
48841	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	U

45175 rows × 15 columns

In [37]:
```
# other method to replace the income column data when column has the object datatype
# we have to_replace function to replace more items in the dataset

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

In [38]:
```
# Which Workclass getting the highest salary ?

data["income"].max()
```

Out[38]:
```
1
```

In [39]:
```
# which is the unique salary?

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

Out[39]:
```
array([0, 1], dtype=int64)
```

In [40]:
```
# no of unique salary in the dataset?

data["income"].nunique()
```

Out[40]:
```
2
```

In [41]:
```
# let's count the unique salary of the dataset

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

Out[41]:
```
income
0    33973
1    11202
Name: count, dtype: int64
```

In [42]:
```
# who has better chances to get salary>50K Male or Female ?

a = data["income"]==">50k"
```

```
b = data["gender"]=="Male"
data.where(data[a & b])
```

Out[42]:

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	nati cour
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
48837	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
48838	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
48839	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
48840	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
48841	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

45175 rows × 15 columns

In [43]:

```
# who has better chances to get salary>50K Male or Female ?

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

Out[43]:

```
gender
Male      0.312609
Female    0.113692
Name: income, dtype: float64
```

as per above data we can say that male have better chanches to get higher salary

In [44]:
```
# which workclass getting the highest salary?

data.groupby('workclass')['income'].mean().sort_values(ascending = False)
```

Out[44]:
```
workclass
Self-emp-inc        0.554407
Federal-gov         0.390469
Local-gov           0.295161
Self-emp-not-inc    0.279051
State-gov           0.267215
Private             0.217816
Without-pay         0.095238
Name: income, dtype: float64
```

In [45]:
```
# Convert workclass columns datatype to category datatype

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

In [46]:
```
# to check column type is updated or not

data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 45175 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column           Non-Null Count  Dtype   
---  ------           --------------  -----   
 0   age              45175 non-null  int64   
 1   workclass        45175 non-null  category
 2   fnlwgt           45175 non-null  int64   
 3   education        45175 non-null  object  
 4   educational-num  45175 non-null  int64   
 5   marital-status   45175 non-null  object  
 6   occupation       45175 non-null  object  
 7   relationship     45175 non-null  object  
 8   race             45175 non-null  object  
 9   gender           45175 non-null  object  
 10  capital-gain     45175 non-null  int64   
 11  capital-loss     45175 non-null  int64   
 12  hours-per-week   45175 non-null  int64   
 13  native-country   45175 non-null  object  
 14  income           45175 non-null  int64   
dtypes: category(1), int64(7), object(7)
memory usage: 5.2+ MB
```

In [47]:

```
# print all the column names in the dataset

data.columns
```

Out[47]:

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'educational-num',
       'marital-status', 'occupation', 'relationship', 'race', 'gender',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'income'],
      dtype='object')
```