Exploratory analysis: Loading and exploring the dataset

In [1]:
```
#Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [57]:
```
#Loading dataset to a dataframe.
data = pd.read_csv('adult.csv')
```

In [3]:
```
data.head()
```

Out[3]:

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

In [4]:
```
data.head(2)
```

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	s
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Ma
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Ma

In [5]:
```
data.head(10)
```

Out[5]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	F
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	F
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	F
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	F
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	

In [6]:
```
data.tail(2)
```

Out[6]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	rac
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	Whit
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	Whit

Q1.This is my answer;

- A significant higher number of people work for private firms, and mostly work 40hours a week(full time).
- The number of adults in this dataset earning less than 50k is almost two-thirds of the entire data.
- There are fewer blacks than other race and they mostly work for private companies.
- There is a mix of important information in the dataset and we need to do more analysis to bring out the usefullness.

In [7]:
```
data.shape
```

Out[7]:
```
(32561, 15)
```

There are 32561 rows and 15 columns in this dataset.

Generating my unique dataset for this task

In [58]:
```
data = data.sample(n=30000, random_state = 65)
```

In [9]:
```
data.shape
```

Out[9]:
```
(30000, 15)
```

In [10]:
```
data.describe()
```

Out[10]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	30000.000000	3.000000e+04	30000.000000	30000.00000	30000.000000	30000.000000
mean	38.540633	1.893694e+05	10.081900	1070.96060	86.269400	40.451733
std	13.644660	1.056206e+05	2.566569	7368.20945	400.924814	12.377417
min	17.000000	1.228500e+04	1.000000	0.00000	0.000000	1.000000
25%	28.000000	1.177578e+05	9.000000	0.00000	0.000000	40.000000
50%	37.000000	1.779070e+05	10.000000	0.00000	0.000000	40.000000
75%	48.000000	2.364970e+05	12.000000	0.00000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.00000	4356.000000	99.000000

In [11]:
```
data['education-num'].value_counts()
```

Out[11]:
```
9     9692
10    6740
13    4926
14    1585
11    1267
7     1066
12     977
6      865
4      601
15     526
5      469
8      412
16     383
3      289
2      156
1       46
Name: education-num, dtype: int64
```

In [12]:
```
data['education'].value_counts()
```

Out[12]:
```
 HS-grad         9692
 Some-college    6740
 Bachelors       4926
 Masters         1585
 Assoc-voc       1267
 11th            1066
 Assoc-acdm       977
 10th             865
 7th-8th          601
 Prof-school      526
 9th              469
 12th             412
 Doctorate        383
 5th-6th          289
 1st-4th          156
 Preschool         46
Name: education, dtype: int64
```

In [13]:
```
data = data.drop(['fnlwgt'], axis=1)
```

In [14]:
```
data.shape
```

Out[14]:
```
(30000, 14)
```

In [15]:
```
data.describe(include='all')
```

Out[15]:

	age	workclass	education	education-num	marital-status	occupation	relationship
count	30000.000000	30000	30000	30000.000000	30000	30000	30000
unique	NaN	9	16	NaN	7	15	6
top	NaN	Private	HS-grad	NaN	Married-civ-spouse	Prof-specialty	Husband
freq	NaN	20901	9692	NaN	13802	3797	12145
mean	38.540633	NaN	NaN	10.081900	NaN	NaN	NaN
std	13.644660	NaN	NaN	2.566569	NaN	NaN	NaN
min	17.000000	NaN	NaN	1.000000	NaN	NaN	NaN
25%	28.000000	NaN	NaN	9.000000	NaN	NaN	NaN
50%	37.000000	NaN	NaN	10.000000	NaN	NaN	NaN
75%	48.000000	NaN	NaN	12.000000	NaN	NaN	NaN
max	90.000000	NaN	NaN	16.000000	NaN	NaN	NaN

In [16]:
```
data['education'].value_counts()
```

Out[16]:
```
HS-grad         9692
Some-college    6740
Bachelors       4926
Masters         1585
Assoc-voc       1267
11th            1066
Assoc-acdm       977
10th             865
7th-8th          601
Prof-school      526
9th              469
12th             412
Doctorate        383
5th-6th          289
1st-4th          156
Preschool         46
Name: education, dtype: int64
```

In [17]:
```
data['education'].nunique()
```

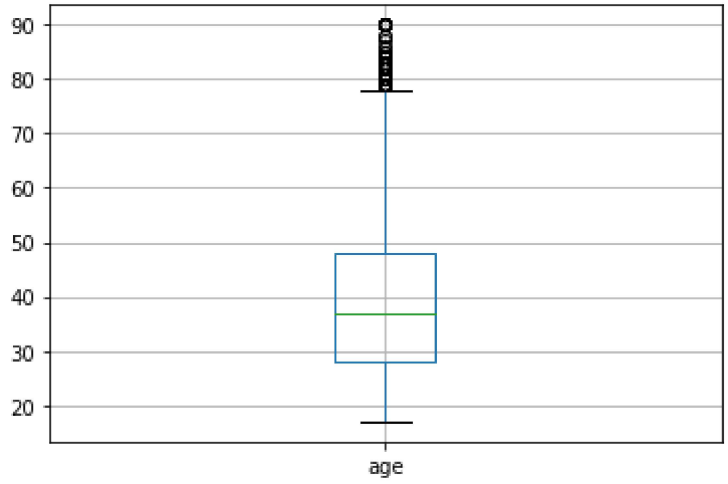
Out[17]: 16

In [18]:
```
data['age'].value_counts()
```

Out[18]:
```
36    831
31    819
34    812
23    812
35    812
     ...
83      3
85      3
88      2
87      1
86      1
Name: age, Length: 73, dtype: int64
```

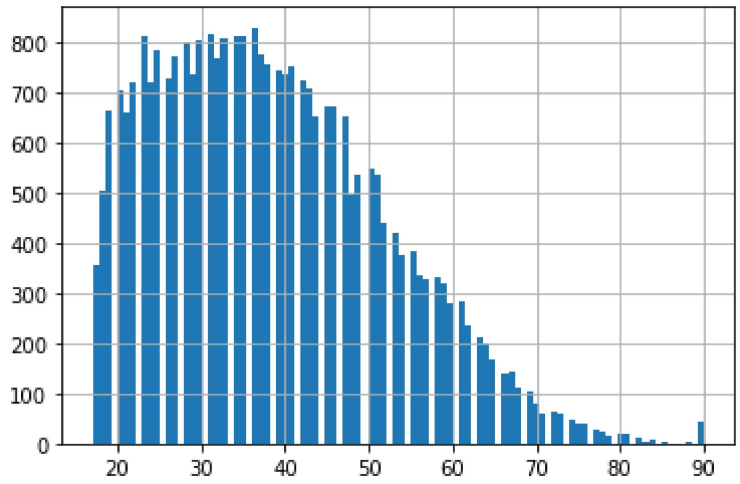
In [19]:
```
data.boxplot(column='age')
```

Out[19]: <AxesSubplot:>



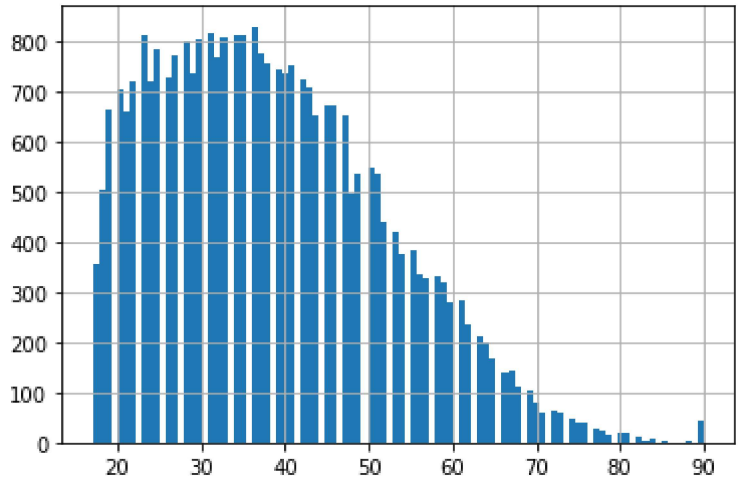
In [20]:
```
data['age'].hist(bins=100)
```

Out[20]: <AxesSubplot:>



In [21]:
```
data.age.hist(bins=100)
```

Out[21]: <AxesSubplot:>



In [22]:
```
data['sex'].value_counts()
```

Out[22]:
```
 Male      20105
 Female     9895
Name: sex, dtype: int64
```

In [23]:
```
data.columns
```

Out[23]:
```
Index(['age', 'workclass', 'education', 'education-num', 'marital-status',
       'occupation', 'relationship', 'race', 'sex', 'capital-gain',
       'capital-loss', 'hours-per-week', 'native-country', 'Class label'],
      dtype='object')
```

In [24]:
```
data['workclass'].value_counts()
```

Out[24]:
```
 Private             20901
 Self-emp-not-inc     2344
 Local-gov            1904
 ?                    1711
 State-gov            1207
 Self-emp-inc         1025
 Federal-gov           888
 Without-pay            13
 Never-worked            7
Name: workclass, dtype: int64
```

In [25]:
```
data['sex'].value_counts()
```

Out[25]:
```
 Male      20105
 Female     9895
Name: sex, dtype: int64
```

Q2;This is my answer;

- There are 20,105 males and 9,895 female in the dataset.

Applying groupby functions in order to summarise the data.

In [26]:
```
data['age'].groupby([data['sex']]).mean()
```

Out[26]:
```
sex
 Female    36.812734
 Male      39.391047
Name: age, dtype: float64
```

In [27]:
```
data['age'].groupby([data['sex'],data['education']]).mean()
```

Out[27]:
```
sex      education
 Female   10th            35.210332
          11th            30.463158
          12th            30.100000
          1st-4th         48.023256
          5th-6th         44.276316
          7th-8th         49.439189
          9th             41.527559
          Assoc-acdm      36.322835
          Assoc-voc       37.903433
          Bachelors       35.694181
          Doctorate       45.462500
          HS-grad         38.654522
          Masters         42.844130
          Preschool       43.000000
          Prof-school     40.456790
          Some-college    33.633797
 Male     10th            38.161616
          11th            33.508746
          12th            33.231618
          1st-4th         44.955752
          5th-6th         42.276995
          7th-8th         47.958057
          9th             40.450292
          Assoc-acdm      37.904362
          Assoc-voc       38.865169
          Bachelors       40.402842
          Doctorate       48.607261
          HS-grad         39.125857
          Masters         44.560953
          Preschool       43.032258
          Prof-school     45.301124
          Some-college    36.824266
Name: age, dtype: float64
```

In [28]:
```
#Q3. The average contribution to capital-gain of each sex and occupation cate
data['capital-gain'].groupby([data['sex'],data['occupation']]).mean()
```

Out[28]:
```
sex      occupation
 Female   ?                    304.350064
          Adm-clerical         517.586854
          Craft-repair         766.352657
          Exec-managerial     1025.592417
          Farming-fishing      691.385965
          Handlers-cleaners    131.283784
          Machine-op-inspct    169.126214
          Other-service        159.763253
          Priv-house-serv      334.178862
          Prof-specialty      1239.608508
          Protective-serv     1808.628571
          Sales                285.069264
          Tech-support         705.930380
          Transport-moving     501.292683
 Male     ?                    828.630181
          Adm-clerical         482.856261
          Armed-Forces           0.000000
          Craft-repair         649.968838
          Exec-managerial     2804.484352
          Farming-fishing      519.286866
          Handlers-cleaners    274.015206
          Machine-op-inspct    384.676162
          Other-service        237.854257
          Priv-house-serv       74.250000
          Prof-specialty      3472.546473
          Protective-serv      606.616412
          Sales               1819.189937
          Tech-support         702.014981
          Transport-moving     506.069885
Name: capital-gain, dtype: float64
```

In [29]:
```
#Q4 The average capital-gain by males and females accross different marital-s
data['capital-gain'].groupby([data['sex'],data['marital-status']]).mean()
```

Out[29]:
```
sex      marital-status
 Female   Divorced                441.961491
          Married-AF-spouse       189.500000
          Married-civ-spouse     1598.004543
          Married-spouse-absent   244.089947
          Never-married           323.870577
          Separated               365.792808
          Widowed                 481.621477
 Male     Divorced               1109.865598
          Married-AF-spouse       810.888889
          Married-civ-spouse     1770.704918
          Married-spouse-absent   978.663317
          Never-married           427.659865
          Separated               897.828804
          Widowed                 997.949045
Name: capital-gain, dtype: float64
```

In [30]:
```
data['race'].value_counts()
```

Out[30]:
```
 White                 25660
 Black                  2873
 Asian-Pac-Islander      935
 Amer-Indian-Eskimo      290
 Other                   242
Name: race, dtype: int64
```

In [31]:
```
data['age'].groupby([data['race']]).max()
```

Out[31]:
```
race
 Amer-Indian-Eskimo    82
 Asian-Pac-Islander    90
 Black                 90
 Other                 77
 White                 90
Name: age, dtype: int64
```

In [32]:
```
#Minimum age by sex:
data['age'].groupby([data['sex']]).min()
```

Out[32]:
```
sex
 Female    17
 Male      17
Name: age, dtype: int64
```

In [33]:
```
#maximum age by sex:
data['age'].groupby([data['sex']]).max()
```

Out[33]:
```
sex
 Female    90
 Male      90
Name: age, dtype: int64
```

Q5; This is my answer

- Yes, the minimum and maximum ages by sex are the same.The minimum ages for both male and female is 17 years old and the maximum ages for both male and female are 90 years old.

Data visualisation

In [34]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

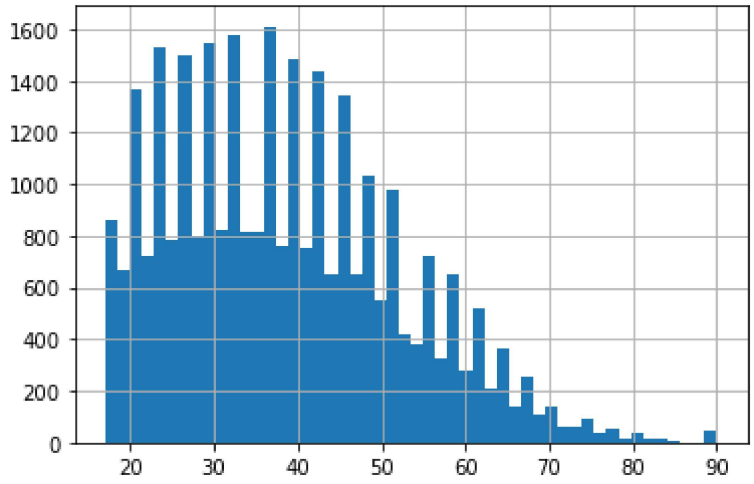
In [35]:
```
data.describe()
```

Out[35]:

	age	education-num	capital-gain	capital-loss	hours-per-week
count	30000.000000	30000.000000	30000.00000	30000.000000	30000.000000
mean	38.540633	10.081900	1070.96060	86.269400	40.451733
std	13.644660	2.566569	7368.20945	400.924814	12.377417
min	17.000000	1.000000	0.00000	0.000000	1.000000
25%	28.000000	9.000000	0.00000	0.000000	40.000000
50%	37.000000	10.000000	0.00000	0.000000	40.000000
75%	48.000000	12.000000	0.00000	0.000000	45.000000
max	90.000000	16.000000	99999.00000	4356.000000	99.000000

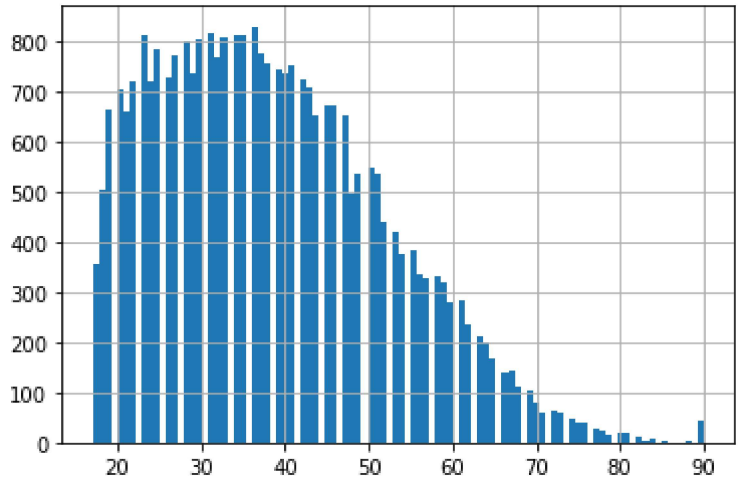
In [36]:
```
data['age'].hist(bins=50)
```

Out[36]: <AxesSubplot:>



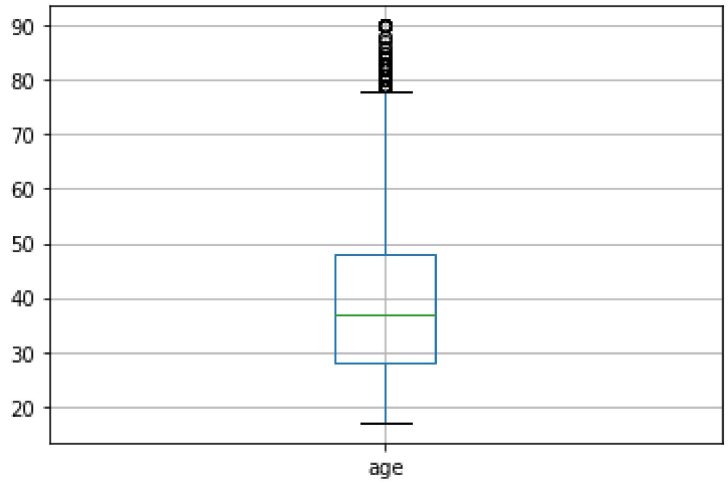
In [37]:
```
data['age'].hist(bins=100)
```

Out[37]: <AxesSubplot:>



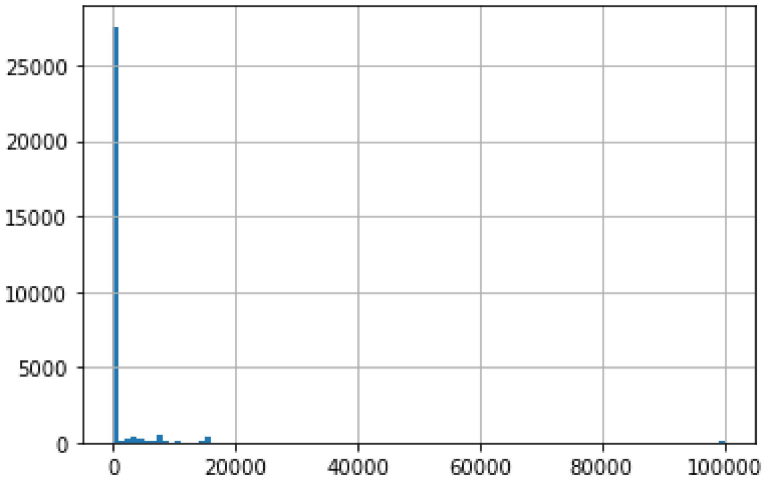
In [38]:
```
data.boxplot(column='age')
```

Out[38]: <AxesSubplot:>



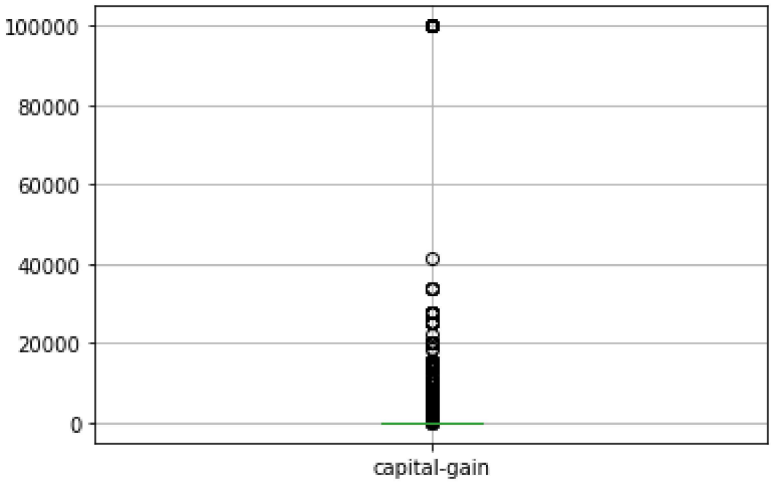
In [39]:
```
data['capital-gain'].hist(bins=100)
```

Out[39]: <AxesSubplot:>



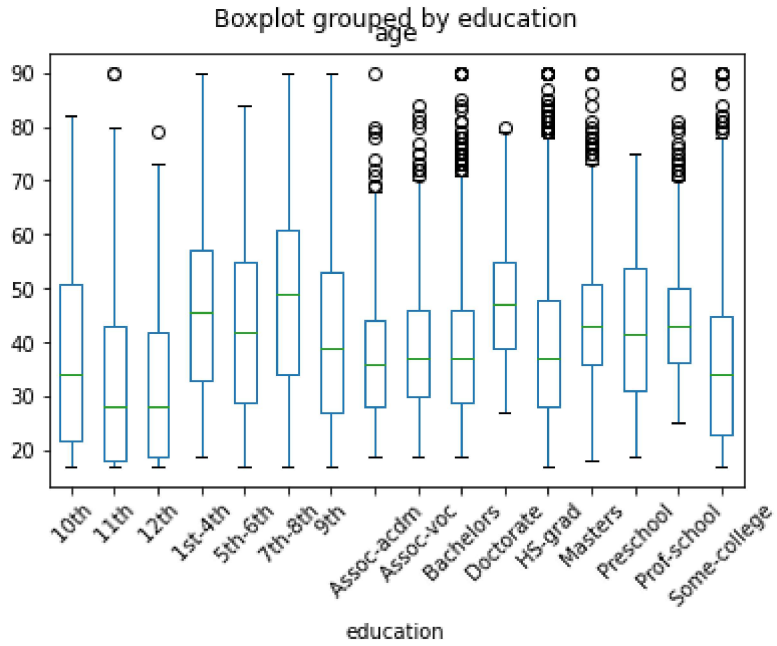
In [40]:
```
data.boxplot(column='capital-gain')
```

Out[40]: <AxesSubplot:>



In [41]:
```
data.boxplot(column='age', by = 'education', grid=False, rot = 45, fontsize =
```

Out[41]: <AxesSubplot:title={'center':'age'}, xlabel='education'>

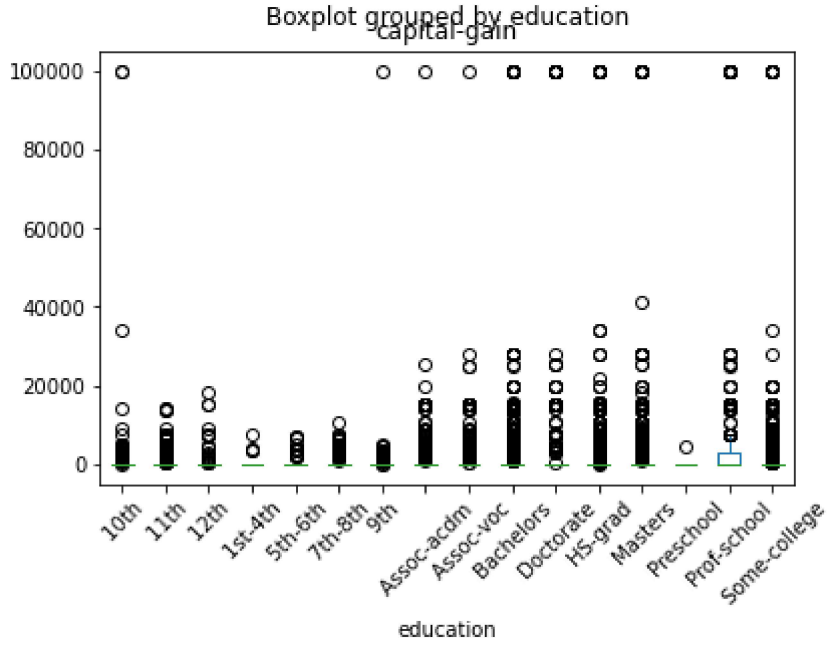


In [42]:
```
data['education'].value_counts()
```

Out[42]:
```
HS-grad         9692
Some-college    6740
Bachelors       4926
Masters         1585
Assoc-voc       1267
11th            1066
Assoc-acdm       977
10th             865
7th-8th          601
Prof-school      526
9th              469
12th             412
Doctorate        383
5th-6th          289
1st-4th          156
Preschool         46
Name: education, dtype: int64
```

In [43]:
```
data.boxplot(column='capital-gain', by = 'education', grid=False, rot = 45, f
```

Out[43]:
```
<AxesSubplot:title={'center':'capital-gain'}, xlabel='education'>
```



In [44]:
```
data['marital-status'].value_counts()
```

Out[44]:
```
Married-civ-spouse       13802
Never-married             9870
Divorced                  4063
Separated                  952
Widowed                    902
Married-spouse-absent      388
Married-AF-spouse           23
Name: marital-status, dtype: int64
```

Checking NULL values in the dataset

In [45]:
```
data.apply(lambda x: sum(x.isnull()), axis = 0)
```

Out[45]:
```
age              0
workclass        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
Class label      0
dtype: int64
```

Data transformation

Label encoding:

In [46]:
```
from sklearn.preprocessing import LabelEncoder
```

In [47]:
```
data.head()
```

Out[47]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	se
19591	41	Private	Bachelors	13	Never-married	Prof-specialty	Not-in-family	White	Femal
18262	31	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Mal
27345	50	Private	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Mal
32095	27	Private	Bachelors	13	Never-married	Exec-managerial	Not-in-family	White	Femal
26379	56	Private	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Mal

In [48]:
```
data.dtypes
```

Out[48]:
```
age               int64
workclass        object
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
Class label      object
dtype: object
```

In [49]:
```
columns = list(data.select_dtypes(exclude=['int64']))
```

In [50]:
```
columns
```

Out[50]:
```
['workclass',
 'education',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'Class label']
```

In [51]:
```
data['Class label'].value_counts()
```

Out[51]:
```
 <=50K    22796
 >50K      7204
Name: Class label, dtype: int64
```

Data transformation

In [52]:
```
le = LabelEncoder()
for i in columns:
    #print(i)
    data[i] = le.fit_transform(data[i])
data.dtypes
```

Out[52]:
```
age               int64
workclass         int32
education         int32
education-num     int64
marital-status    int32
occupation        int32
relationship      int32
race              int32
sex               int32
capital-gain      int64
capital-loss      int64
hours-per-week    int64
native-country    int32
Class label       int32
dtype: object
```

In [53]:
```
data.head()
```

Out[53]:

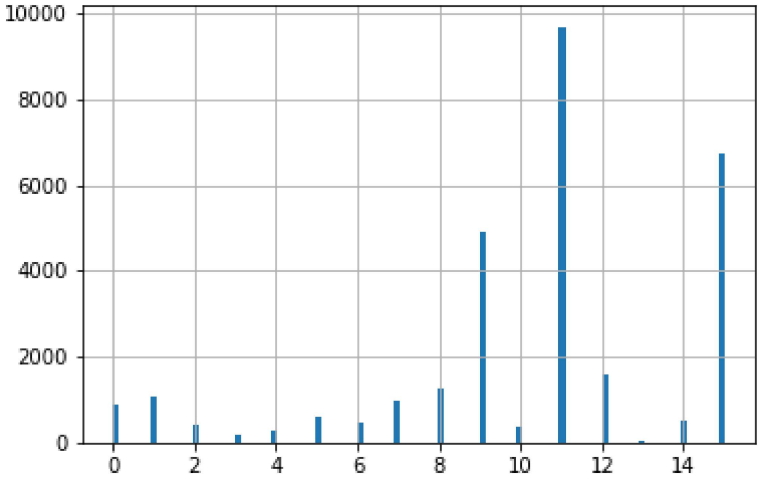
	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	c
19591	41	4	9	13	4	10	1	4	0	
18262	31	4	9	13	2	10	0	4	1	
27345	50	4	9	13	2	4	0	4	1	
32095	27	4	9	13	4	4	1	4	0	
26379	56	4	11	9	2	3	0	4	1	

In [54]:
```
data['workclass'].value_counts()
```

Out[54]:
```
4    20901
6     2344
2     1904
0     1711
7     1207
5     1025
1      888
8       13
3        7
Name: workclass, dtype: int64
```

In [55]:
```
data['education'].hist(bins=100)
```

Out[55]: <AxesSubplot:>



In [56]:
```
data.describe(include='all')
```

Out[56]:

	age	workclass	education	education-num	marital-status	occupation	r
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	3
mean	38.540633	3.868767	10.305533	10.081900	2.614767	6.564433	
std	13.644660	1.460005	3.869343	2.566569	1.503409	4.229670	
min	17.000000	0.000000	0.000000	1.000000	0.000000	0.000000	
25%	28.000000	4.000000	9.000000	9.000000	2.000000	3.000000	
50%	37.000000	4.000000	11.000000	10.000000	2.000000	7.000000	
75%	48.000000	4.000000	12.000000	12.000000	4.000000	10.000000	
max	90.000000	8.000000	15.000000	16.000000	6.000000	14.000000	

REPORT

#Q6 this is my answer; data.describe() 5 number summary of the dataset.

- It can be seen that there is a really large variance between the capial loss and capital gain which can possibly caused by the extreme values or data collection anomalies but in our dataset it is a case of extreme values and in our dataset and there are a lot of 0 values.
- The average age is 38.58 which means they might have various responsibilities that may affect their work like dependants. minimum and maximum ages are 17 and 90 respectively.

- The minimum time spent on education is 1 year while the maximum time is 16 years and the mean time is 10 years, which means they all have some sort of formal education.
- The hours spent per week were at 1hour at the minimum and 99 hours at the maximum with an average of 40 hours which can be from self employed individuals that may work more or less hours than staff.
- The average, minimum and maximum capital-gain are 1070, 0 and 99999 respectively which seems like an anomaly from data collection or storage errors, therefore, the data still needs pre-processing.

In [59]:
```
#Q7 Datatypes
data.dtypes
```

Out[59]:
```
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
Class label       object
dtype: object
```

Q7. This is my answer;

- Data types define how data is stored and manipulates,there are several datatypes we use in data-mining, such as floats representing decimal numbers, integers representing numbers, character(Char) representing a single alphanumeric character,string representing one or more alphanumeric character, boolean representing two option choice like true/false, and object is usually used a a refernve data type. In this dataset.
- We have both int32 and int64 from the data above.
- int32 and int64 both represent numbers but stored in different spaces.
- object datatype represents a generic object that can contain any type of data in this case categorical variables.

In [60]:
```
data['native-country'].value_counts()
```

Out[60]:
```
United-States                 26899
Mexico                          577
?                               539
Philippines                     182
Germany                         127
Canada                          114
Puerto-Rico                     108
El-Salvador                      95
India                            91
England                          86
Cuba                             85
South                            75
China                            74
Italy                            69
Jamaica                          67
Dominican-Republic               63
Guatemala                        61
Vietnam                          59
Japan                            57
```

Q8.- Highest migrants belongs to which country? This is my answer;

- 15(Holand-Netherlands) has the least native-country indigen, which means all the indigens have migrated, therefore the country with the most migration.

In [61]:
```
#Q9 occupation representing more males than females
data.groupby(['occupation','sex']).count()
```

Out[61]:

occupation	sex	age	workclass	fnlwgt	education	education-num	marital-status	relationship	rac
?	Female	777	777	777	777	777	777	777	77
	Male	941	941	941	941	941	941	941	94
Adm-clerical	Female	2343	2343	2343	2343	2343	2343	2343	234
	Male	1134	1134	1134	1134	1134	1134	1134	113
Armed-Forces	Male	8	8	8	8	8	8	8	
Craft-repair	Female	207	207	207	207	207	207	207	20
	Male	3562	3562	3562	3562	3562	3562	3562	356
Exec-managerial	Female	1055	1055	1055	1055	1055	1055	1055	105
	Male	2684	2684	2684	2684	2684	2684	2684	268
Farming-fishing	Female	57	57	57	57	57	57	57	5
	Male	868	868	868	868	868	868	868	86
Handlers-cleaners	Female	148	148	148	148	148	148	148	14
	Male	1118	1118	1118	1118	1118	1118	1118	111
Machine-op-inspct	Female	515	515	515	515	515	515	515	51
	Male	1334	1334	1334	1334	1334	1334	1334	133
Other-service	Female	1660	1660	1660	1660	1660	1660	1660	166
	Male	1386	1386	1386	1386	1386	1386	1386	138
Priv-house-serv	Female	123	123	123	123	123	123	123	12
	Male	8	8	8	8	8	8	8	
Prof-specialty	Female	1387	1387	1387	1387	1387	1387	1387	138
	Male	2410	2410	2410	2410	2410	2410	2410	241
Protective-serv	Female	70	70	70	70	70	70	70	7
	Male	524	524	524	524	524	524	524	52
Sales	Female	1155	1155	1155	1155	1155	1155	1155	115
	Male	2206	2206	2206	2206	2206	2206	2206	220
Tech-support	Female	316	316	316	316	316	316	316	31
	Male	534	534	534	534	534	534	534	53
Transport-moving	Female	82	82	82	82	82	82	82	8
	Male	1388	1388	1388	1388	1388	1388	1388	138

Q9. This is my answer;

- As seen from the table above, there are no women in the armed forces so we can conclude that from this dataset armed forces represents more males than females.

Q10. What is the difference between data.head() and data.tail()? -- This is my answer;

- data.head() outputs a specified number of rows from the top of the dataset. The head() returns the first 5 rows if a number is not specified.
- data.tail() outputs a specified number of rows from the end of the dataset. The tail() returns the last 5 rows if a number is not specified.

Introduction

Exploratory analysis refers to the process of analyzing and summarizing data to discover underlying patterns, trends, relationships, and insights. It involves using statistical and graphical methods to explore the data, identify any outliers or anomalies, and generate hypotheses that can be tested with further analysis. Data Processing and Experimentation

In this workshop, we worked with an "Adult Dataset" and pre-processed the dataset to make it suitable for analysis. A unique dataset was created from the original dataset to work with and a random state of 65, a description, and the analysis of individual features were performed to get a deeper understanding of the dataset. There were no missing values in this dataset because they had already been replaced with a question mark from the origin of the dataset and therefore our analysis would not be affected by this, although in the case of a missing value, we look at other values given for that particular column, if there is a significantly large population having a value it is best to give the missing value that large value(mode), in some cases where the values do not really have a significant difference the mean can be used to replace a missing value.

A group by function was also used to perform analysis by grouping data by columns to see the relationships between them to draw some insights and conclusions. We also did label encoding on the dataset and checked out all the data types present in the dataset. Some visualizations were done with boxplots and histograms switching the number of bins around to get a better understanding.

Conclusion

Exploratory analysis is a very important and crucial part of data mining because it will determine how well your data can perform when it is fed into a machine learning algorithm, overall, exploratory analysis is a crucial first step in any data analysis project. By gaining a deeper understanding of the data and identifying potential relationships and patterns, analysts can develop more effective models and make more informed decisions.

In [ ]: