SAMUEL ATEKHA OSAGHAE

STUDENT ID 2120094

In [59]:

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

In [60]:

```
data = pd.read_csv('adult.csv')
```

Q1. Use head(2), head(10), tail(2). Explain your observations, in no more than 2 to 3 lines.

ANSWER

In [61]:

```
data.head(2)
```

Out[61]:

	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	M
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	M

In [62]:

```
data.head(10)
```

Out[62]:

	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
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black
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
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White

In [63]:

```
data.tail(2)
```

Out[63]:

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

EXPLANATION; I observed that most of the adult in this dat from the selected view are all native of America 90 percent is a native of United state of American while other are from Cuba and Jamaica

Generating my Unique datasets

In [66]:

```
data = data.sample(n=30000, random_state = 94)
```

In [67]:

```
data.to_csv('samuelosaghae_2120094_ WEEK1.csv')
```

In [7]:

```
data.shape
```

Out[7]:

```
(30000, 15)
```

In [8]:

```
data.describe()
```

Out[8]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	30000.000000	3.000000e+04	30000.000000	30000.000000	30000.000000	30000.000000
mean	38.588733	1.898409e+05	10.077967	1074.551000	87.514767	40.433033
std	13.619539	1.051800e+05	2.579698	7394.107726	403.735249	12.315220
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.179040e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783765e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.377148e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

from the above the oldest adult is 90years while the youngst is 17years

In [9]:

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

Out[9]:

```
9     9667
10    6714
13    4920
14    1595
11    1268
7     1091
12     980
6      842
4      598
15     531
5      481
8      404
16     388
3      312
2      162
1       47
Name: education-num, dtype: int64
```

In [10]:

```
data['education'].value_counts()
```

Out[10]:

```
 HS-grad         9667
 Some-college    6714
 Bachelors       4920
 Masters         1595
 Assoc-voc       1268
 11th            1091
 Assoc-acdm       980
 10th             842
 7th-8th          598
 Prof-school      531
 9th              481
 12th             404
 Doctorate        388
 5th-6th          312
 1st-4th          162
 Preschool         47
Name: education, dtype: int64
```

In [11]:

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

The above cell will drop/remove 'fnlwgt' from data.

drop(): To drop a column from the dataframe, pass agruments - column name to be dropped and axis = 1. axis = 0 is to dropping row.

In [12]:

```
data.shape
```

Out[12]:

```
(30000, 14)
```

After removing the 'fnlwgt' column it is ovbserve that there are now 14 columns instead of 15.

In [13]:

```
data.describe(include='all')
```

Out[13]:

	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	20935	9667	NaN	13802	3815	12174
mean	38.588733	NaN	NaN	10.077967	NaN	NaN	NaN
std	13.619539	NaN	NaN	2.579698	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 [14]:

```
data['education'].value_counts()
```

Out[14]:

```
 HS-grad         9667
 Some-college    6714
 Bachelors       4920
 Masters         1595
 Assoc-voc       1268
 11th            1091
 Assoc-acdm       980
 10th             842
 7th-8th          598
 Prof-school      531
 9th              481
 12th             404
 Doctorate        388
 5th-6th          312
 1st-4th          162
 Preschool         47
Name: education, dtype: int64
```

value_counts() produces a frequency table, which shows occurance of each feature or attribute in a dataset.

In [15]:

```
data['education'].nunique()
```

Out[15]:

```
16
```

In [16]:

```
data['age'].value_counts()
```

Out[16]:

```
31    827
36    825
23    814
34    811
28    796
     ...
83      5
88      2
85      2
87      1
86      1
Name: age, Length: 73, dtype: int64
```

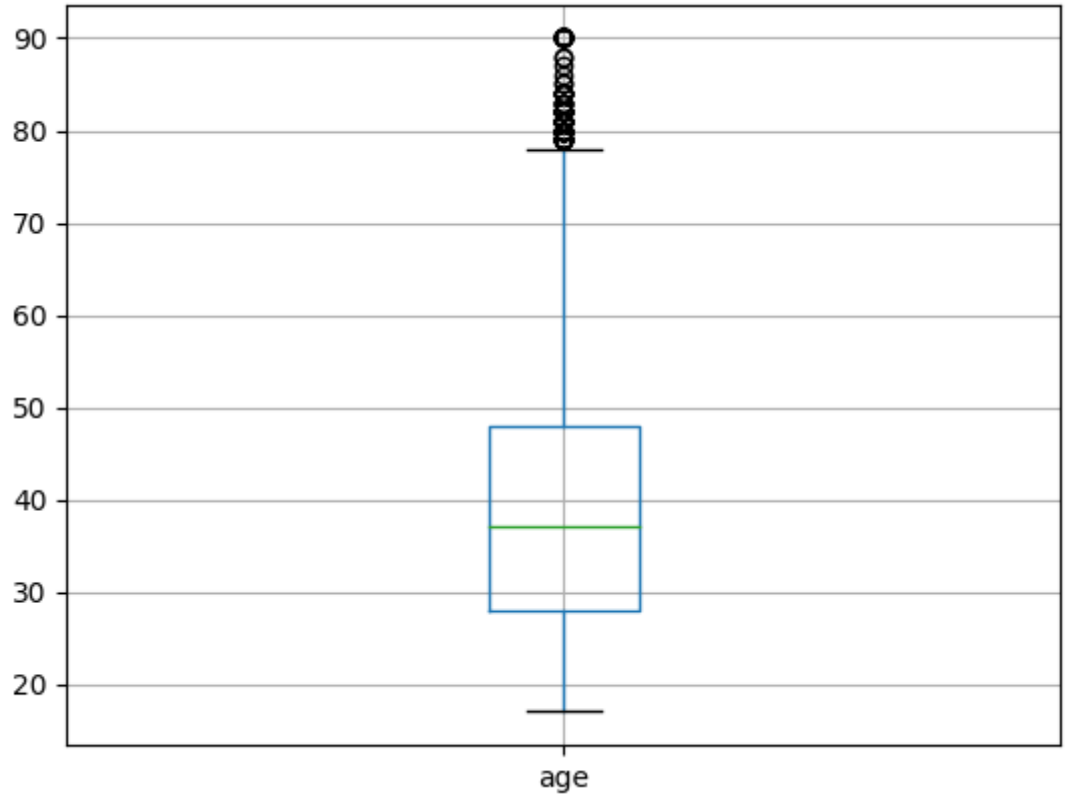
Age frequency table is too lengthy to be analysed. This is due to 'age' being continuous value and frequency of each value is displayed. Let's visualise 'age' through graphs instead to make observations.

In [17]:

```
data.boxplot(column='age')
```

Out[17]:

<AxesSubplot:>



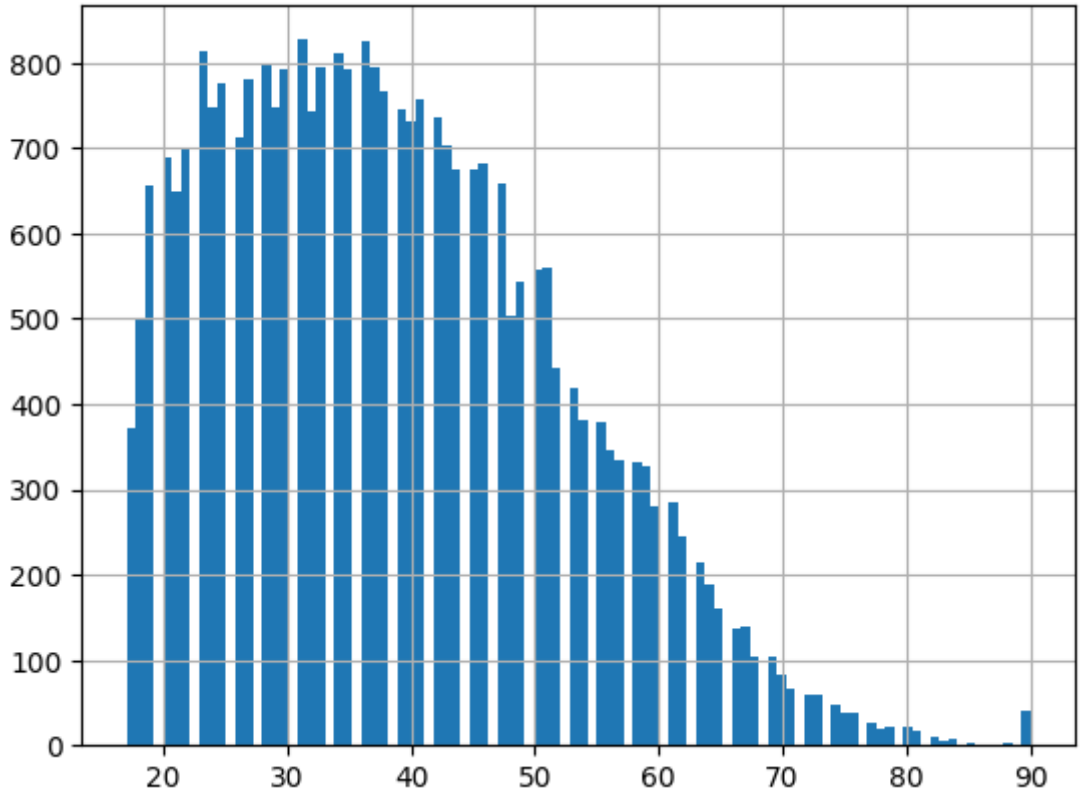
from the above Boxplot, it is seen that the plot gave us the maximun age value as about 79 while the minimum age value as lower than 20 and then shows that ages of adult above 80 as outliers which is quite different from the data described above

In [18]:

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

Out[18]:

<AxesSubplot:>

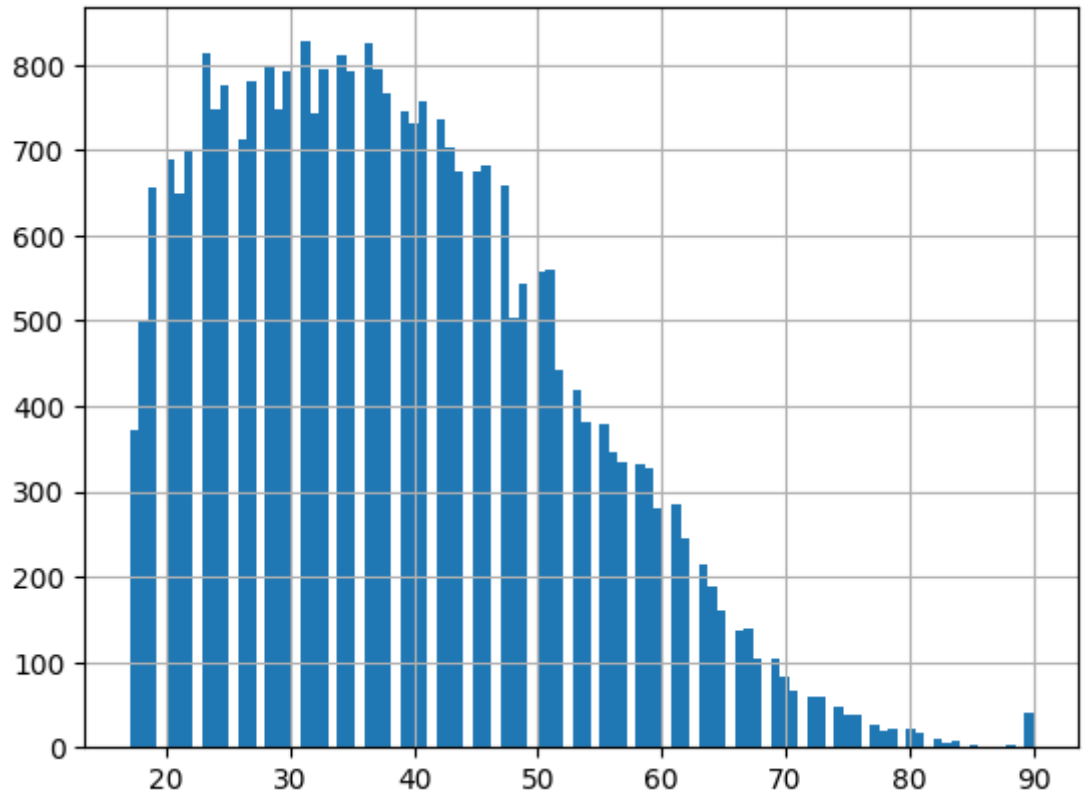


In [19]:

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

Out[19]:

```
<AxesSubplot:>
```



In [20]:

```
data['sex'].value_counts()
```

Out[20]:

```
 Male      20097
 Female     9903
Name: sex, dtype: int64
```

In [21]:

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

Out[21]:

```
 United-States                 26857
 Mexico                          598
 ?                               541
 Philippines                     181
 Germany                         123
 Canada                          109
 Puerto-Rico                     104
 El-Salvador                      99
 India                            94
 Cuba                             90
 England                          81
 Jamaica                          76
 South                            74
 China                            71
 Italy                            70
 Vietnam                          65
 Dominican-Republic               64
 Guatemala                        61
 Japan                            57
 Poland                           51
 Columbia                         51
 Taiwan                           47
 Iran                             42
 Haiti                            41
 Portugal                         36
 Nicaragua                        31
 Peru                             31
 France                           27
 Ecuador                          27
 Greece                           23
 Ireland                          20
 Cambodia                         19
 Trinadad&Tobago                  19
 Hong                             19
 Laos                             18
 Thailand                         16
 Yugoslavia                       15
 Outlying-US(Guam-USVI-etc)       14
 Honduras                         13
 Scotland                         12
 Hungary                          12
 Holand-Netherlands                1
Name: native-country, dtype: int64
```

In [22]:

```
data.columns
```

Out[22]:

```
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 [23]:

```
data['workclass'].value_counts()
```

Out[23]:

```
 Private             20935
 Self-emp-not-inc     2331
 Local-gov            1923
 ?                    1671
 State-gov            1198
 Self-emp-inc         1029
 Federal-gov           892
 Without-pay            14
 Never-worked            7
Name: workclass, dtype: int64
```

Q2. How many males and females exist in the dataset? In a new cell, use a correct command to answer the question and write your answer.

In [24]:

```
data['sex'].value_counts()
```

Out[24]:

```
 Male      20097
 Female     9903
Name: sex, dtype: int64
```

ANSWER: The number males and females exist in the dataset Male are 20097 while Female are 9903

Question: What is the average age of each gender in the given population?

In [25]:

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

Out[25]:

```
sex
 Female    36.819651
 Male      39.460467
Name: age, dtype: float64
```

ANSWER: the average of the male is 39.4604 while the female is 36.819651

Question. What is the average age of male and female across different educAtion categories?

ANSWER

In [26]:

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

Out[26]:

```
sex      education
 Female   10th            35.296296
          11th            30.269802
          12th            29.971014
          1st-4th         47.800000
          5th-6th         44.282051
          7th-8th         49.219178
          9th             42.180451
          Assoc-acdm      36.377892
          Assoc-voc       37.789474
          Bachelors       35.667566
          Doctorate       44.850000
          HS-grad         38.626087
          Masters         43.135729
          Preschool       40.133333
          Prof-school     40.058824
          Some-college    33.751845
 Male     10th            38.159091
          11th            33.563319
          12th            33.112782
          1st-4th         45.692308
          5th-6th         42.598291
          7th-8th         47.893805
          9th             40.913793
          Assoc-acdm      37.944162
          Assoc-voc       39.023399
          Bachelors       40.331685
          Doctorate       48.396104
          HS-grad         39.089150
          Masters         44.750457
          Preschool       42.906250
          Prof-school     45.607623
          Some-college    37.045663
Name: age, dtype: float64
```

Q3. What is the average contribution to capital-gain of each sex and occupation category?

ANSWER

The average contribution to capital-gain of each sex

In [27]:

```
data['capital-gain'].groupby([data['sex']]).mean()
```

Out[27]:

```
sex
 Female     546.062405
 Male      1334.969100
Name: capital-gain, dtype: float64
```

The average contribution to capital-gain of each occupation category.

In [28]:

```
data['capital-gain'].groupby([data['occupation']]).mean()
```

Out[28]:

```
occupation
 ?                    614.206198
 Adm-clerical         524.931214
 Armed-Forces           0.000000
 Craft-repair         666.146903
 Exec-managerial     2263.708746
 Farming-fishing      567.100223
 Handlers-cleaners    185.401569
 Machine-op-inspct    338.375599
 Other-service        161.597965
 Priv-house-serv      289.225564
 Prof-specialty      2670.550459
 Protective-serv      666.846154
 Sales               1365.213523
 Tech-support         678.293503
 Transport-moving     494.641781
Name: capital-gain, dtype: float64
```

Q4. Identify the average capital-gain by males and females accross different marital-status.

ANSWER

The average average capital-gain by males and females accross different marital-status is shown below

In [29]:

```
data['capital-gain'].groupby([data['sex'],data['marital-status']]).mean()
```

Out[29]:

```
sex      marital-status
 Female   Divorced                452.086601
          Married-AF-spouse       189.500000
          Married-civ-spouse     1527.466623
          Married-spouse-absent   383.697297
          Never-married           319.586332
          Separated               194.157986
          Widowed                 524.287433
 Male     Divorced               1110.794226
          Married-AF-spouse       810.888889
          Married-civ-spouse     1806.516153
          Married-spouse-absent   986.150000
          Never-married           411.749863
          Separated               592.784916
          Widowed                1019.164474
Name: capital-gain, dtype: float64
```

Question. What is the maximum age accross differnt races?

ANSWER

Let's first see what are the different races and then apply groupby.

In [30]:

```
data['race'].value_counts()
```

Out[30]:

```
 White                 25608
 Black                  2898
 Asian-Pac-Islander      970
 Amer-Indian-Eskimo      283
 Other                   241
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
```

ANSWER: The maximum age by race is Amer-Indian-Eskimo 82, Asian-Pac-Islander 90, Black 90, Other 77, White 90

Q5. Are minimum and maximum age by sex same?

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
```

ANSWER: The minimum and maximum age by sex same are the same

DATA VISUALISATION

In [34]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [35]:

```
data.describe()
```

Out[35]:

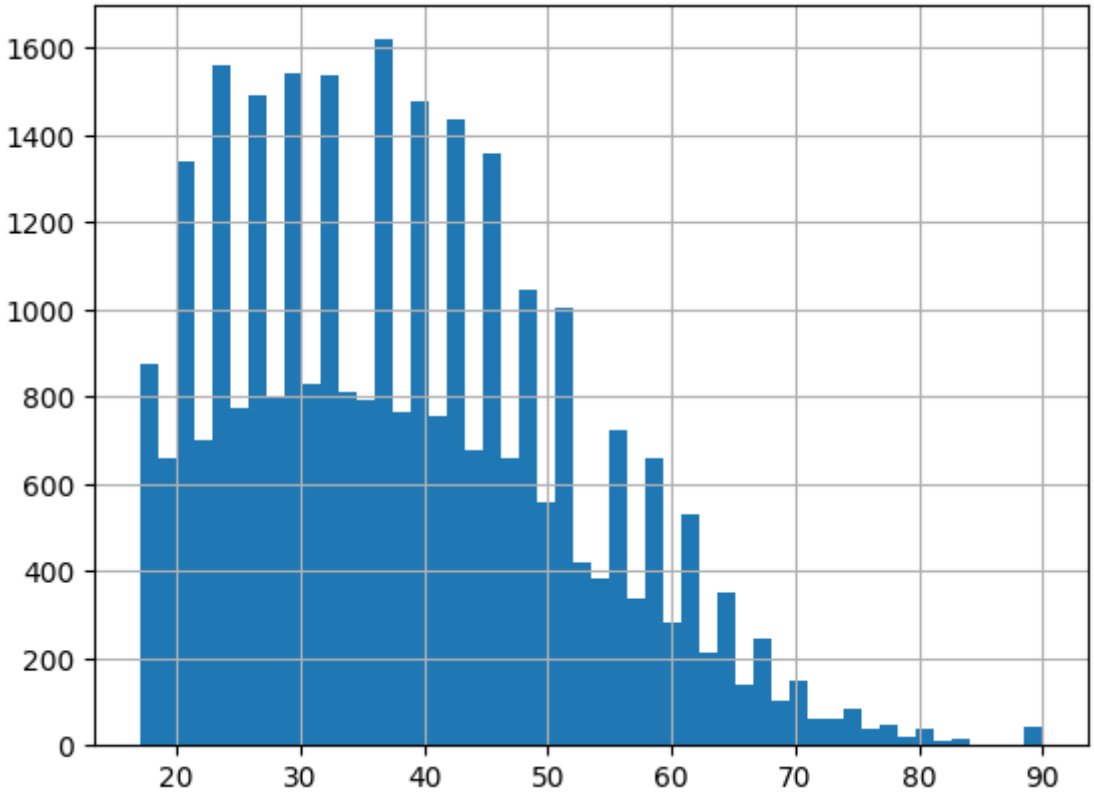
	age	education-num	capital-gain	capital-loss	hours-per-week
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	38.588733	10.077967	1074.551000	87.514767	40.433033
std	13.619539	2.579698	7394.107726	403.735249	12.315220
min	17.000000	1.000000	0.000000	0.000000	1.000000
25%	28.000000	9.000000	0.000000	0.000000	40.000000
50%	37.000000	10.000000	0.000000	0.000000	40.000000
75%	48.000000	12.000000	0.000000	0.000000	45.000000
max	90.000000	16.000000	99999.000000	4356.000000	99.000000

In [36]:

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

Out[36]:

<AxesSubplot:>

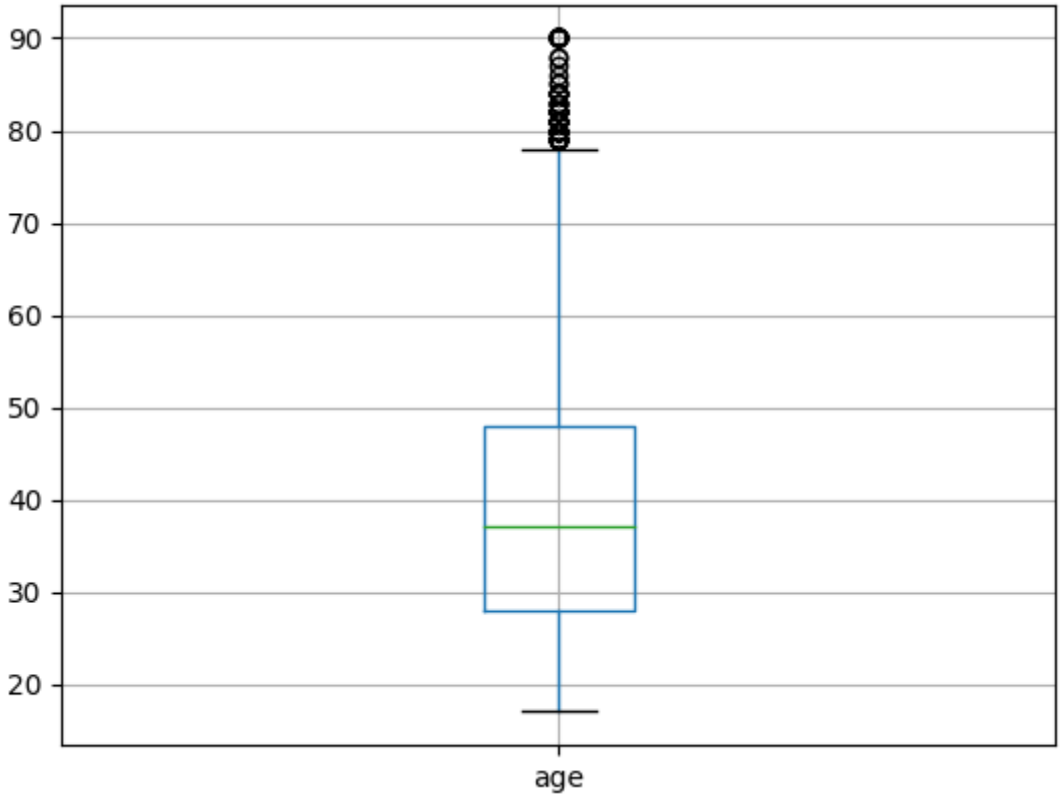


In [37]:

```
data.boxplot(column='age')
```

Out[37]:

<AxesSubplot:>

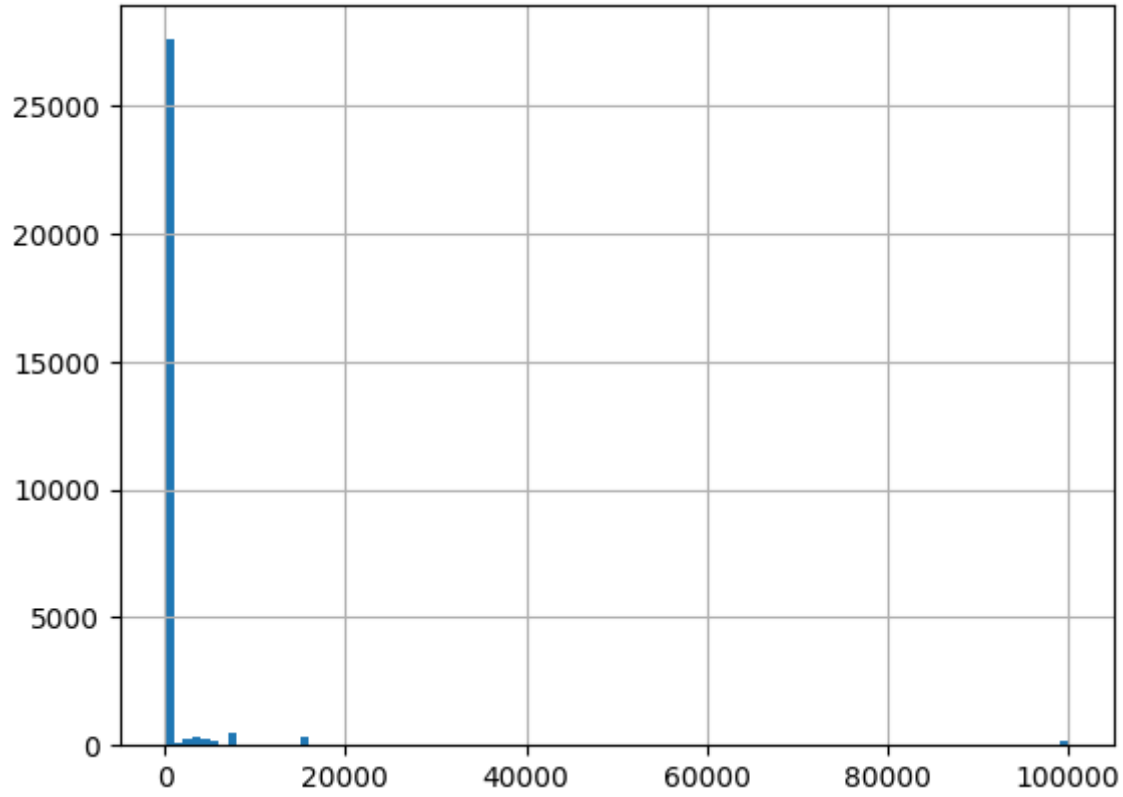


In [38]:

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

Out[38]:

<AxesSubplot:>

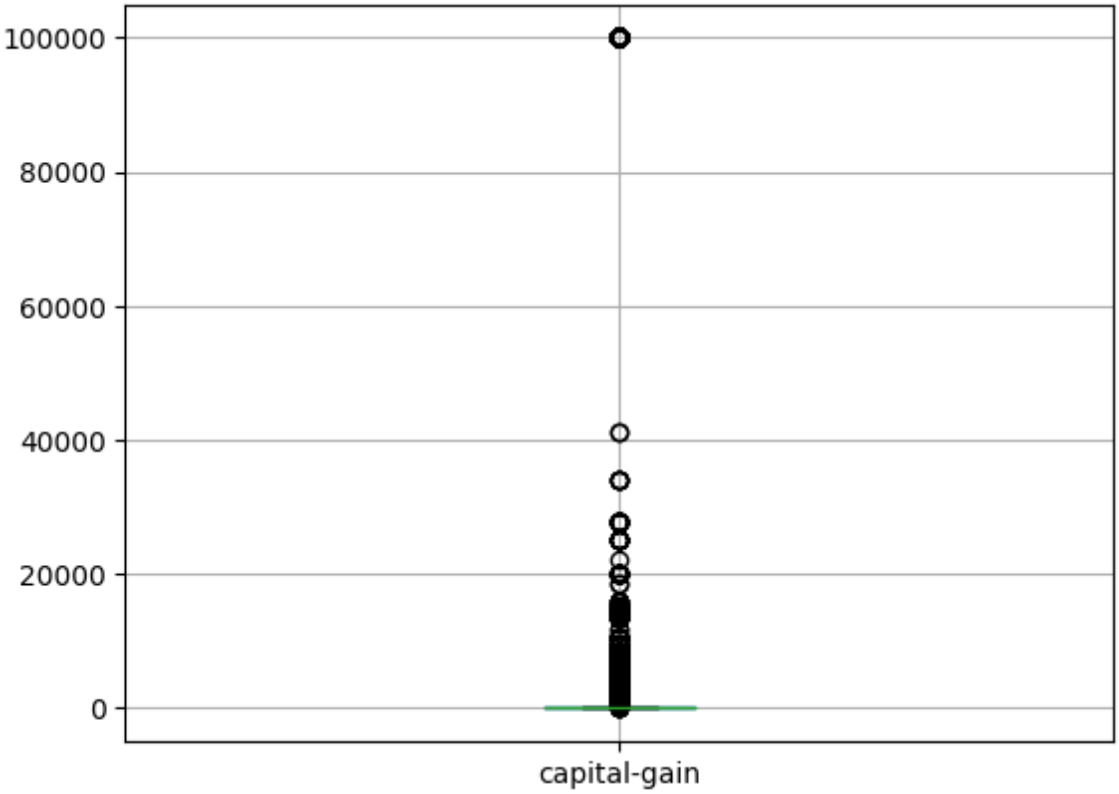


In [39]:

```
data.boxplot(column='capital-gain')
```

Out[39]:

<AxesSubplot:>

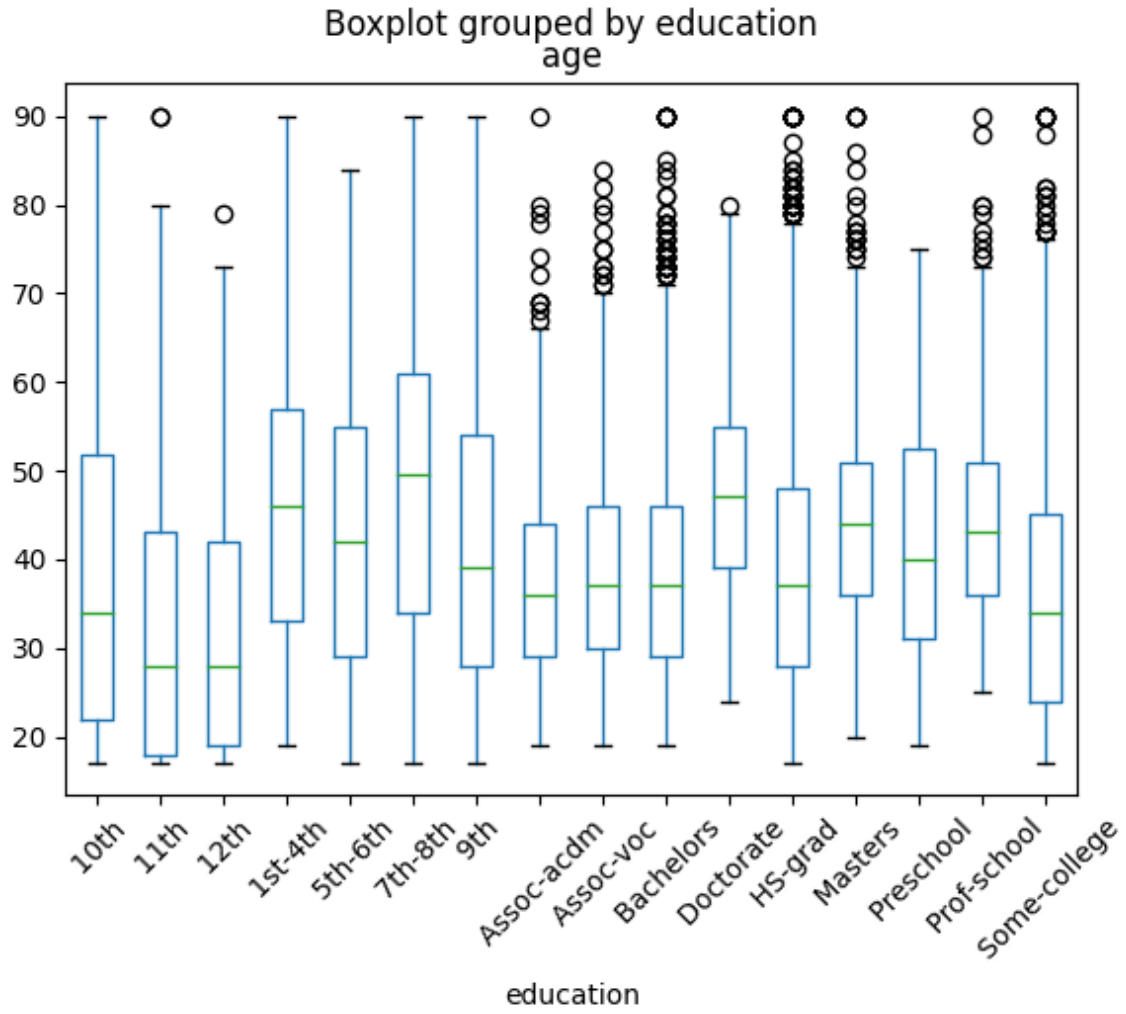


In [40]:

```
data.boxplot(column='age', by = 'education', grid=False, rot = 45, fontsize = 10)
```

Out[40]:

```
<AxesSubplot:title={'center':'age'}, xlabel='education'>
```



In [41]:

```
data['education'].value_counts()
```

Out[41]:

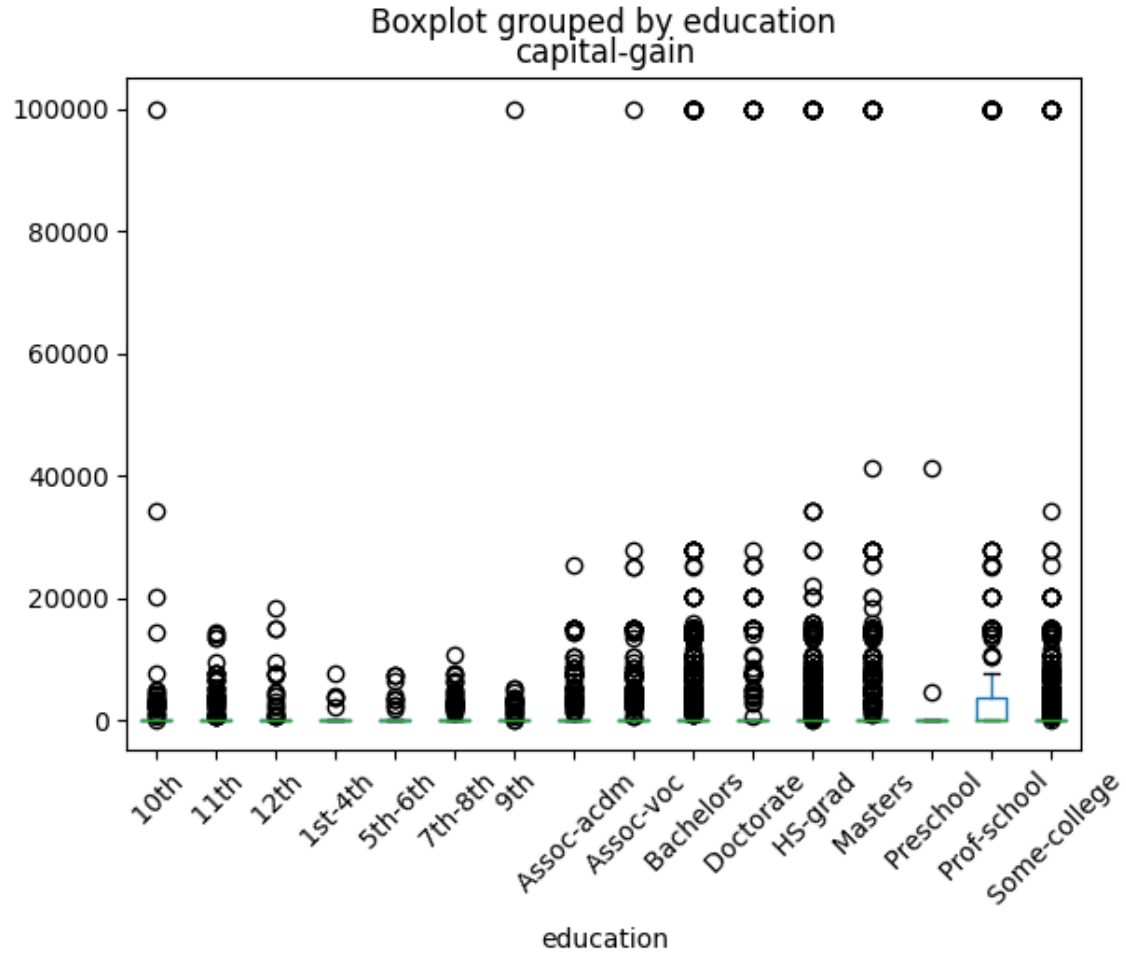
```
 HS-grad         9667
 Some-college    6714
 Bachelors       4920
 Masters         1595
 Assoc-voc       1268
 11th            1091
 Assoc-acdm       980
 10th             842
 7th-8th          598
 Prof-school      531
 9th              481
 12th             404
 Doctorate        388
 5th-6th          312
 1st-4th          162
 Preschool         47
Name: education, dtype: int64
```

In [42]:

```
data.boxplot(column='capital-gain', by = 'education', grid=False, rot = 45, fontsize = 10
```

Out[42]:

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



In [43]:

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

Out[43]:

```
 Married-civ-spouse       13802
 Never-married             9880
 Divorced                  4076
 Separated                  934
 Widowed                    900
 Married-spouse-absent      385
 Married-AF-spouse           23
Name: marital-status, dtype: int64
```

Checking NULL values in the dataset

In [44]:

```
data.apply(lambda x: sum(x.isnull()), axis = 0)
```

Out[44]:

```
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

In [45]:

```
from sklearn.preprocessing import LabelEncoder
```

In [46]:

```
data.head()
```

Out[46]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	
23176	18	?	HS-grad	9	Never-married	?	Own-child	White	Fen
22684	54	Private	Masters	14	Married-civ-spouse	Prof-specialty	Husband	White	M
14705	54	Private	10th	6	Separated	Sales	Unmarried	White	Fen
8186	21	Private	HS-grad	9	Separated	Machine-op-inspct	Other-relative	White	M
28725	52	State-gov	HS-grad	9	Separated	Other-service	Not-in-family	White	Fen

In [47]:

```
data.dtypes
```

Out[47]:

```
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 [48]:

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

In [49]:

```
columns
```

Out[49]:

```
['workclass',
 'education',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'class-label']
```

In [50]:

```
data['class-label'].value_counts()
```

Out[50]:

```
 <=50K    22763
 >50K      7237
Name: class-label, dtype: int64
```

In [51]:

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

Out[51]:

```
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 [52]:

```
data.head(10)
```

Out[52]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	c
23176	18	0	11	9	4	0	3	4	0	
22684	54	4	12	14	2	10	0	4	1	
14705	54	4	0	6	5	12	4	4	0	
8186	21	4	11	9	5	7	2	4	1	
28725	52	7	11	9	5	8	1	4	0	
14214	32	4	11	9	2	12	0	1	1	
3820	24	6	9	13	4	5	4	4	1	
2448	29	2	11	9	4	11	1	2	0	
23832	27	4	8	11	4	8	1	1	0	
24058	18	4	0	6	4	12	1	4	1	

In [53]:

```
data['workclass'].value_counts()
```

Out[53]:

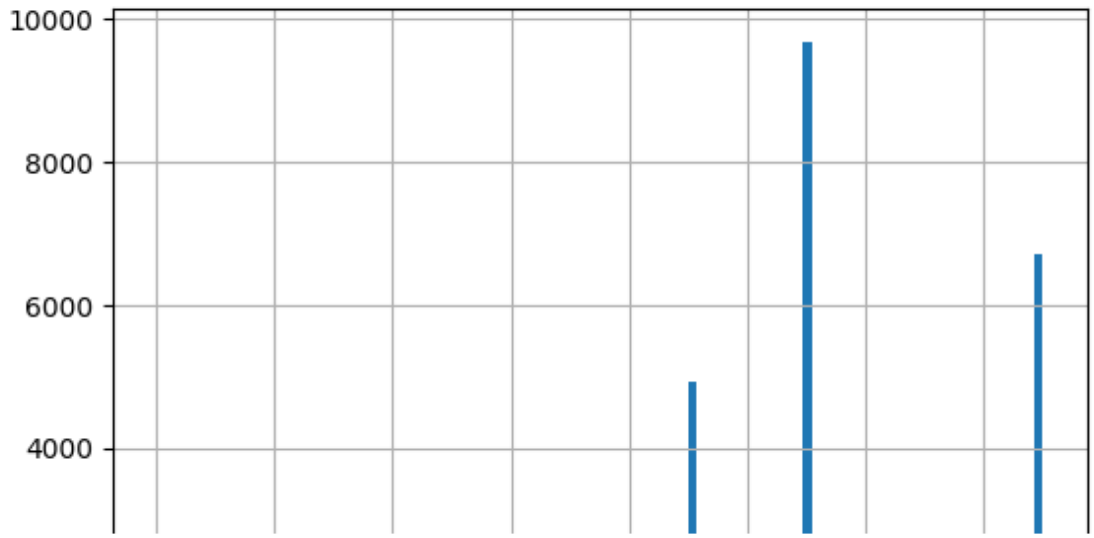
```
4    20935
6     2331
2     1923
0     1671
7     1198
5     1029
1      892
8       14
3        7
Name: workclass, dtype: int64
```

In [54]:

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

Out[54]:

```
<AxesSubplot:>
```



In [55]:

```
data.describe(include='all')
```

Out[55]:

	age	workclass	education	education-num	marital-status	occupation	
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.00000	30000.000000	3
mean	38.588733	3.870933	10.296833	10.077967	2.61240	6.581867	
std	13.619539	1.452863	3.869723	2.579698	1.50321	4.223248	
min	17.000000	0.000000	0.000000	1.000000	0.00000	0.000000	
25%	28.000000	4.000000	9.000000	9.000000	2.00000	3.000000	
50%	37.000000	4.000000	11.000000	10.000000	2.00000	7.000000	
75%	48.000000	4.000000	12.000000	12.000000	4.00000	10.000000	
max	90.000000	8.000000	15.000000	16.000000	6.00000	14.000000	

Q9. Which occupation represents more males than females?

ANSWER

In [56]:

```
data['occupation'].groupby([data['sex']]).value_counts()
```

Out[56]:

```
sex  occupation
0    1             2335
     8             1654
     10            1399
     12            1162
     4             1065
     0              760
     7              513
     13             324
     3              202
     6              153
     9              125
     14              82
     11              69
     5               60
1    3             3576
     4             2674
     10            2416
     12            2210
     8             1393
     14            1378
     7             1364
     1             1125
     6             1122
     0              918
     5              838
     13             538
     11             529
     2                8
     9                8
Name: occupation, dtype: int64
```

The difference betwen data head and data tail

In [57]:

```
data.head(10)
```

Out[57]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	c
23176	18	0	11	9	4	0	3	4	0	
22684	54	4	12	14	2	10	0	4	1	
14705	54	4	0	6	5	12	4	4	0	
8186	21	4	11	9	5	7	2	4	1	
28725	52	7	11	9	5	8	1	4	0	
14214	32	4	11	9	2	12	0	1	1	
3820	24	6	9	13	4	5	4	4	1	
2448	29	2	11	9	4	11	1	2	0	
23832	27	4	8	11	4	8	1	1	0	
24058	18	4	0	6	4	12	1	4	1	

In [58]:

```
data.tail(10)
```

Out[58]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	c
14170	49	4	15	10	0	8	1	4	0	
21060	41	2	15	10	0	4	1	4	0	
29530	47	4	11	9	2	6	5	2	0	
16199	30	4	11	9	4	5	3	4	1	
20193	49	4	10	16	2	10	0	3	1	
29256	50	6	11	9	2	12	0	2	1	
20266	25	0	9	13	4	0	3	4	0	
22236	38	6	7	12	2	12	0	4	1	
23713	30	2	9	13	2	11	0	4	1	
6663	18	0	15	10	4	0	3	4	1	

In [ ]:

```

```

In [ ]: