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

In [2]:
```
data = pd.read_csv('C:\\Users\\User\\Downloads\\adult.data.csv')
```

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

Out[3]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

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

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K

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

Out[5]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	0	16	Jamaica	<=50K
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	45	United-States	>50K
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	0	50	United-States	>50K
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	0	40	United-States	>50K

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

Out[6]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

I observed that when the value of head is not specified, it shows only the first 5 rows. We can specify the number of rows we want by putting the number in the bracket. We can also use head or tail to specify if we want to print the rows at the beginning or the end of the data, head for beginning, tail for end.

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

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

In [8]:
```
data = data.sample(n=30000, random_state = 10)
```

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	30000.000000	30000.000000	30000.000000
mean	38.586867	1.898092e+05	10.085067	1095.360267	87.023167	40.427933
std	13.643079	1.055980e+05	2.570016	7468.186357	402.059923	12.359357
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178702e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.784690e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.368790e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

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

Out[11]:
```
9     9673
10    6710
13    4938
14    1586
11    1295
7     1098
12     981
6      841
4      589
15     530
5      459
8      403
16     383
3      313
2      155
1       46
Name: education-num, dtype: int64
```

In [12]:
```
data = data.drop(['fnlwgt'], axis=1)
```

In [13]:
```
data.shape
```

Out[13]:
```
(30000, 14)
```

In [14]:
```
data.describe(include='all')
```

Out[14]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-w
count	30000.000000	30000	30000	30000.000000	30000	30000	30000	30000	30000	30000.000000	30000.000000	30000.000
unique	NaN	9	16	NaN	7	15	6	5	2	NaN	NaN	
top	NaN	Private	HS-grad	NaN	Married-civ-spouse	Prof-specialty	Husband	White	Male	NaN	NaN	
freq	NaN	20910	9673	NaN	13817	3801	12159	25627	20031	NaN	NaN	
mean	38.586867	NaN	NaN	10.085067	NaN	NaN	NaN	NaN	NaN	1095.360267	87.023167	40.427
std	13.643079	NaN	NaN	2.570016	NaN	NaN	NaN	NaN	NaN	7468.186357	402.059923	12.359
min	17.000000	NaN	NaN	1.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	1.000
25%	28.000000	NaN	NaN	9.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	40.000
50%	37.000000	NaN	NaN	10.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	40.000
75%	48.000000	NaN	NaN	12.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	45.000
max	90.000000	NaN	NaN	16.000000	NaN	NaN	NaN	NaN	NaN	99999.000000	4356.000000	99.000

In [15]:
```
data['education'].value_counts()
```

Out[15]:
```
 HS-grad         9673
 Some-college    6710
 Bachelors       4938
 Masters         1586
 Assoc-voc       1295
 11th            1098
 Assoc-acdm       981
 10th             841
 7th-8th          589
 Prof-school      530
 9th              459
 12th             403
 Doctorate        383
 5th-6th          313
 1st-4th          155
 Preschool         46
Name: education, dtype: int64
```

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

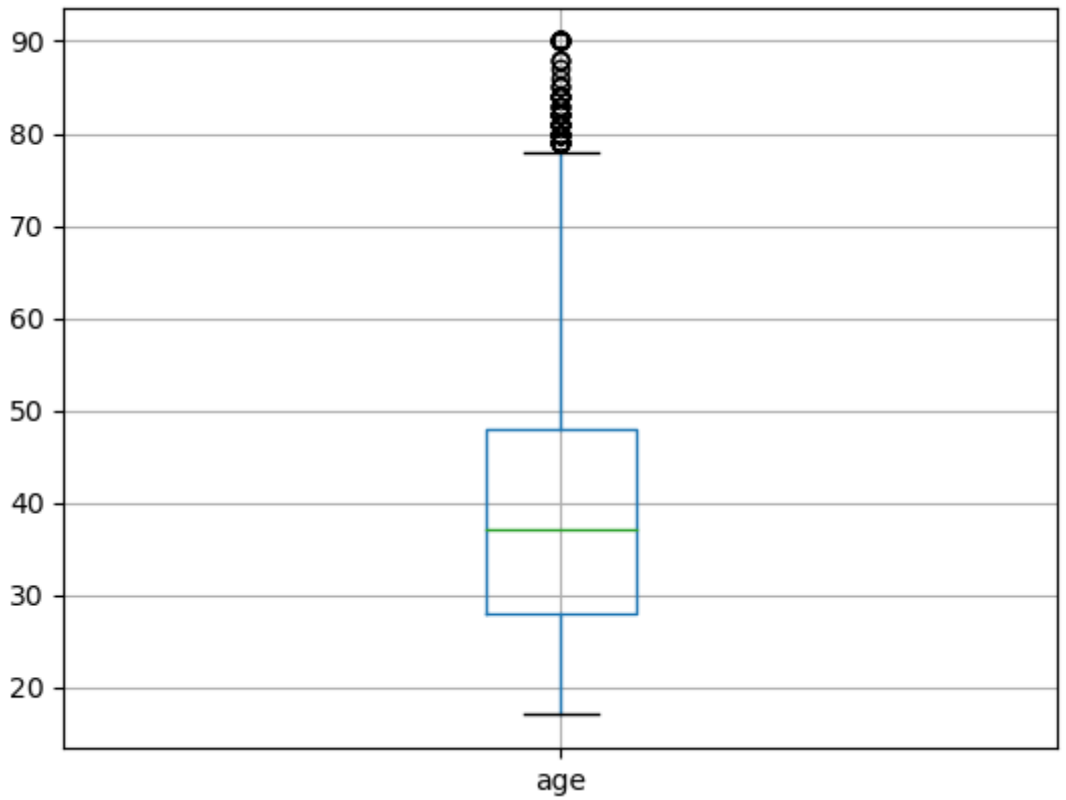
Out[16]:
```
16
```

In [17]:
```
data['age'].value_counts()
```

Out[17]:
```
23    837
36    823
34    815
31    813
28    808
     ...
83      5
88      3
85      3
87      1
86      1
Name: age, Length: 73, dtype: int64
```

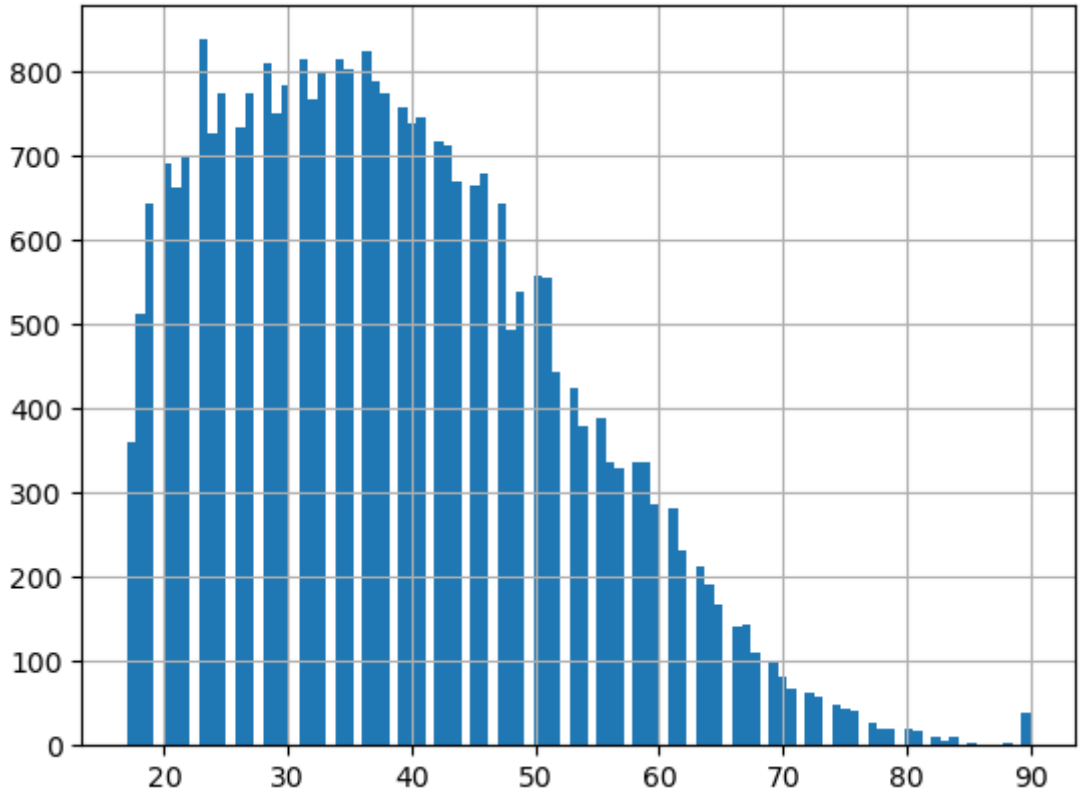
In [18]:
```
data.boxplot(column='age')
```

Out[18]: <AxesSubplot:>



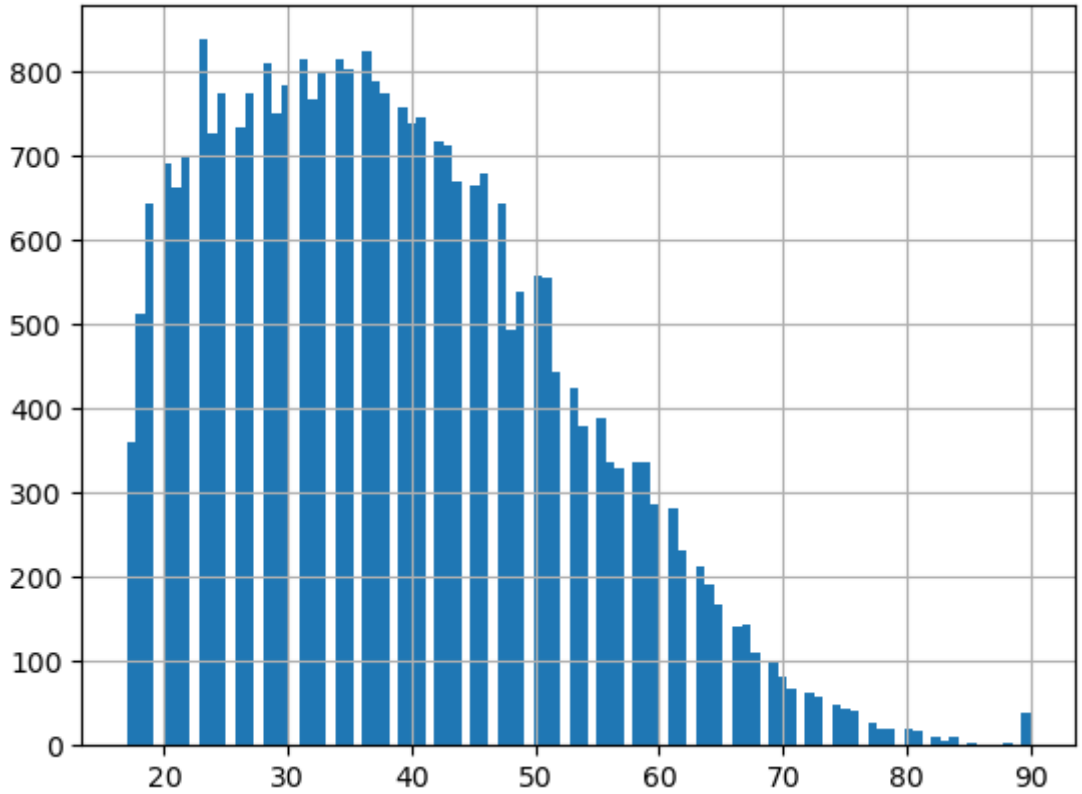
In [19]:
```
data['age'].hist(bins=100)
```

Out[19]: <AxesSubplot:>



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

Out[20]: <AxesSubplot:>



In [21]:
```
data['sex'].value_counts()
```

Out[21]:
```
 Male      20031
 Female     9969
Name: sex, 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             20910
 Self-emp-not-inc     2335
 Local-gov            1929
 ?                    1688
 State-gov            1203
 Self-emp-inc         1026
 Federal-gov           888
 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.
From the code in line 24, there are 20,031 Males and 9969 Females.

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

Out[24]:
```
 Male      20031
 Female     9969
Name: sex, dtype: int64
```

In [25]:
```
data['age'].groupby([data['sex']]).mean()
```

Out[25]:
```
sex
 Female    36.865684
 Male      39.443463
Name: age, dtype: float64
```

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

Out[26]:
```
sex      education
 Female   10th            35.690566
          11th            30.588089
          12th            29.941606
          1st-4th         47.931818
          5th-6th         43.857143
          7th-8th         49.438356
          9th             41.952756
          Assoc-acdm      36.507614
          Assoc-voc       37.818763
          Bachelors       35.486667
          Doctorate       45.400000
          HS-grad         38.721138
          Masters         43.331313
          Preschool       41.750000
          Prof-school     39.705882
          Some-college    33.809140
 Male     10th            38.430556
          11th            33.453237
          12th            33.406015
          1st-4th         45.819820
          5th-6th         42.584746
          7th-8th         48.259594
          9th             40.581325
          Assoc-acdm      37.993186
          Assoc-voc       39.050847
          Bachelors       40.336824
          Doctorate       48.732673
          HS-grad         39.068744
          Masters         44.494959
          Preschool       45.133333
          Prof-school     45.575281
          Some-college    36.996347
Name: age, dtype: float64
```

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

```
#Please see codes  in line 27 below for the answer.
```

In [27]:
```
data['capital-gain'].groupby([data['sex'],data['occupation']]).mean()
```

Out[27]:
```
sex      occupation
 Female   ?                    342.748387
          Adm-clerical         522.166171
          Craft-repair         777.595122
          Exec-managerial      988.086271
          Farming-fishing     1139.610169
          Handlers-cleaners    125.802632
          Machine-op-inspct    172.112205
          Other-service        105.622902
          Priv-house-serv      316.184615
          Prof-specialty      1336.941092
          Protective-serv     1889.611940
          Sales                282.102128
          Tech-support         678.891641
          Transport-moving     489.357143
 Male     ?                    716.485870
          Adm-clerical         469.710619
          Armed-Forces           0.000000
          Craft-repair         654.515135
          Exec-managerial     2764.238823
          Farming-fishing      591.269860
          Handlers-cleaners    278.696997
          Machine-op-inspct    409.199550
          Other-service        242.845428
          Priv-house-serv       74.250000
          Prof-specialty      3786.083437
          Protective-serv      617.631086
          Sales               1840.253772
          Tech-support         672.562384
          Transport-moving     505.820885
Name: capital-gain, dtype: float64
```

```
Q4. Identify the average capital-gain by males and females accross different marital-status.
#Please see codes in line 28 below for the answer.
```

In [28]:
```
data['capital-gain'].groupby([data['sex'],data['marital-status']]).mean()
```

Out[28]:
```
sex      marital-status
 Female   Divorced                  424.265746
          Married-AF-spouse         204.076923
          Married-civ-spouse       1598.179454
          Married-spouse-absent     392.278689
          Never-married             342.578864
          Separated                 359.550256
          Widowed                   505.130208
 Male     Divorced                 1208.248922
          Married-AF-spouse         810.888889
          Married-civ-spouse       1818.726851
          Married-spouse-absent     990.241206
          Never-married             407.028461
          Separated                 889.558659
          Widowed                  1006.671053
Name: capital-gain, dtype: float64
```

In [29]:
```
data['race'].value_counts()
```

Out[29]:
```
 White                 25627
 Black                  2881
 Asian-Pac-Islander      951
 Amer-Indian-Eskimo      287
 Other                   254
Name: race, dtype: int64
```

In [30]:
```
data['age'].groupby([data['race']]).max()
```

Out[30]:
```
race
 Amer-Indian-Eskimo    82
 Asian-Pac-Islander    90
 Black                 90
 Other                 77
 White                 90
Name: age, dtype: int64
```

```
Q5. Are minimum and maximum age by sex same?
#Minimum age for males is the same with females, while maximum age for females is the same with males
```

```
#See codes in line 31 and 32 below showing the answer.
```

In [31]:
```
#minimum age by sex
data['age'].groupby([data['sex']]).min()
```

Out[31]:
```
sex
 Female    17
 Male      17
Name: age, dtype: int64
```

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

Out[32]:
```
sex
 Female    90
 Male      90
Name: age, dtype: int64
```

```
Q8. Highest migrants belongs to which country?
The United States have the highest number of immigrants with 26885 as seen in the output of the codes run in line 33
```

In [33]:
```
data['native-country'].value_counts()
```

Out[33]:

United-States	26885
Mexico	589
?	533
Philippines	177
Germany	131
Canada	112
Puerto-Rico	105
El-Salvador	96
India	92
Cuba	85
England	85
Jamaica	75
China	72
South	72
Italy	69
Vietnam	64
Dominican-Republic	61
Guatemala	59
Japan	59
Columbia	55
Poland	53
Taiwan	48
Iran	39
Haiti	39
Portugal	34
Nicaragua	31
Peru	30
Greece	28
Ecuador	27
France	25
Ireland	22
Laos	18
Thailand	17
Cambodia	17
Hong	17
Yugoslavia	15
Trinadad&Tobago	15
Outlying-US(Guam-USVI-etc)	14
Honduras	13
Hungary	12
Scotland	9

```
 Holand-Netherlands                1
Name: native-country, dtype: int64
```

Q9. Which occupation represents more males than females?
From codes run in line #34, the Armed Forces; Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Prof-specialty, Protective-serv, Sales, Tech-support and Transport-moving have more males than females as shown in the result of the code below.
The Armed Forces have only Males, Craft-repair have 3568 males as against 205 females, Exec-managerial have 2684 Males, as against 1078 females, Farming-fishing have 859 Males as against 59 females; Handlers-cleaners 1099 Males as against 152 females , Machine-op-inspct have 1333 Males as against 508 females , Prof-specialty have 2409 Males as against 1392 females , Protective-ser have 534 Males as against 67 femalev , Sales have 2187 Males as against 1175 females , Tech-support have 537 Males as against 323 females; Transport-moving have 1379 Males as against 84 females

In [34]:
```
data['sex'].groupby([data['occupation']]).value_counts()
```

Out[34]:
```
occupation          sex
 ?                   Male       920
                     Female     775
 Adm-clerical        Female    2353
                     Male      1130
 Armed-Forces        Male         9
 Craft-repair        Male      3568
                     Female     205
 Exec-managerial     Male      2684
                     Female    1078
 Farming-fishing     Male       856
                     Female      59
 Handlers-cleaners   Male      1099
                     Female     152
 Machine-op-inspct   Male      1333
                     Female     508
 Other-service       Female    1668
                     Male      1378
 Priv-house-serv     Female     130
                     Male         8
 Prof-specialty      Male      2409
                     Female    1392
 Protective-serv     Male       534
                     Female      67
 Sales               Male      2187
                     Female    1175
 Tech-support        Male       537
                     Female     323
 Transport-moving    Male      1379
                     Female      84
Name: sex, dtype: int64
```

In [35]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

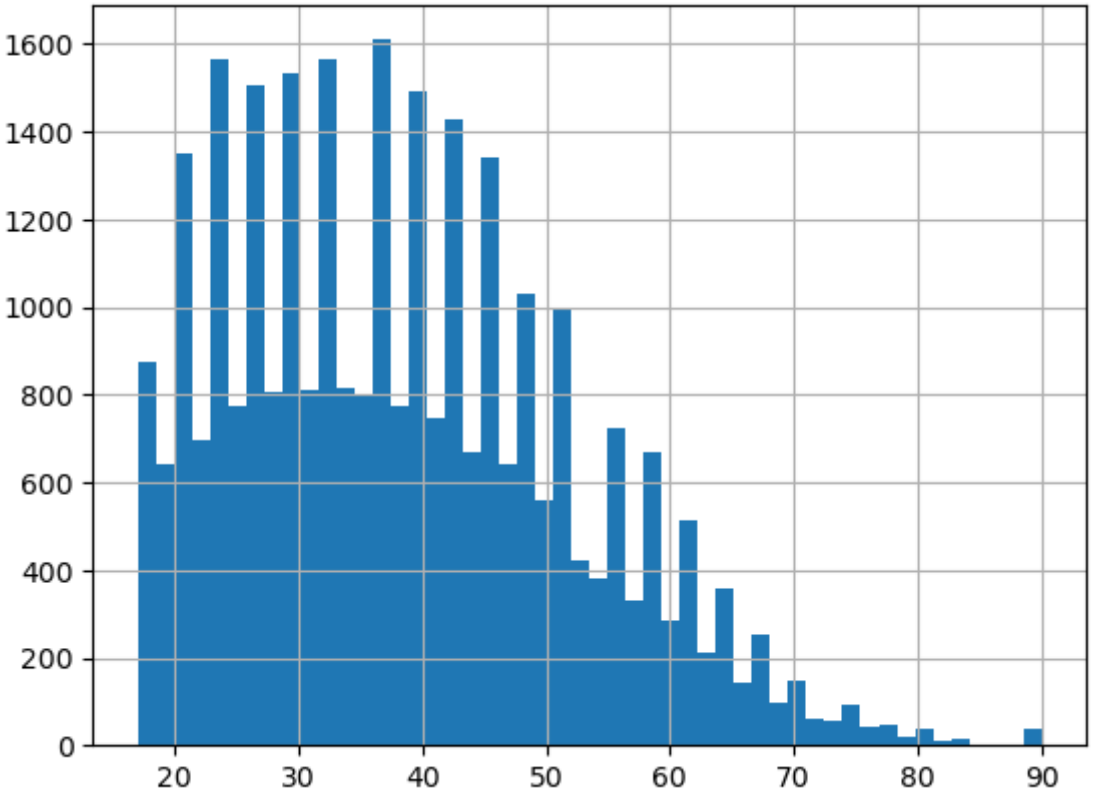
In [36]:
```
data.describe()
```

Out[36]:

	age	education-num	capital-gain	capital-loss	hours-per-week
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000
mean	38.586867	10.085067	1095.360267	87.023167	40.427933
std	13.643079	2.570016	7468.186357	402.059923	12.359357
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

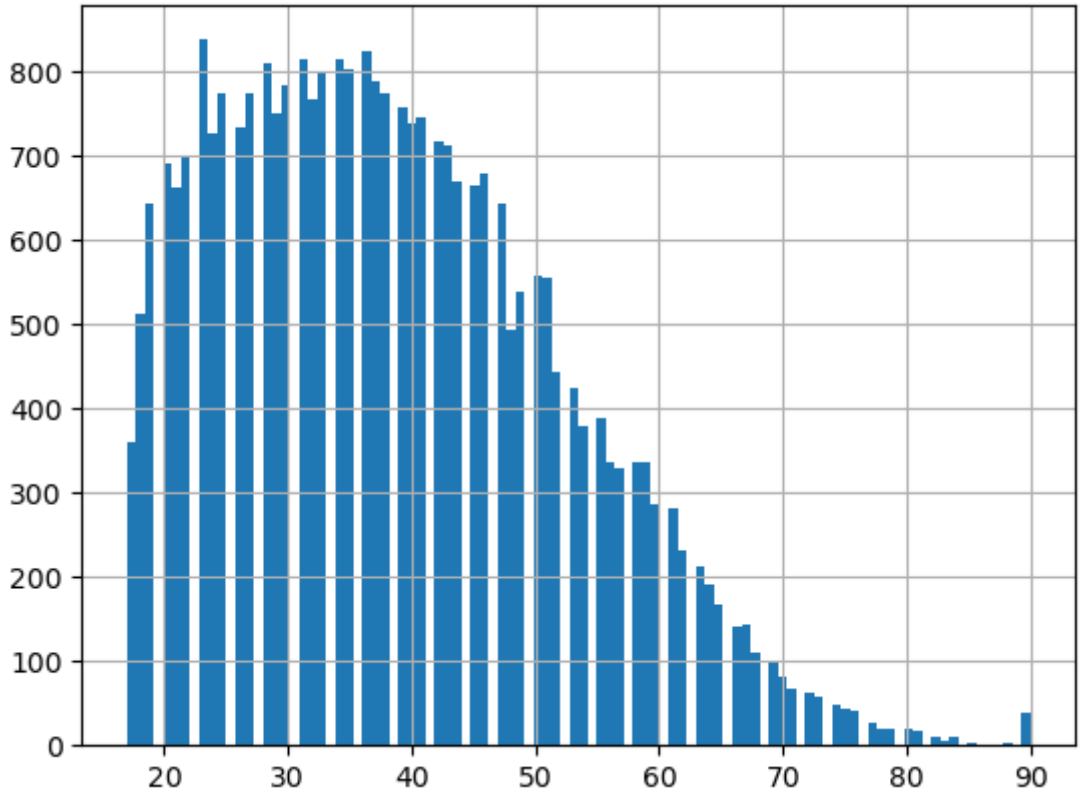
In [37]:
```
data['age'].hist(bins=50)
```

Out[37]: <AxesSubplot:>



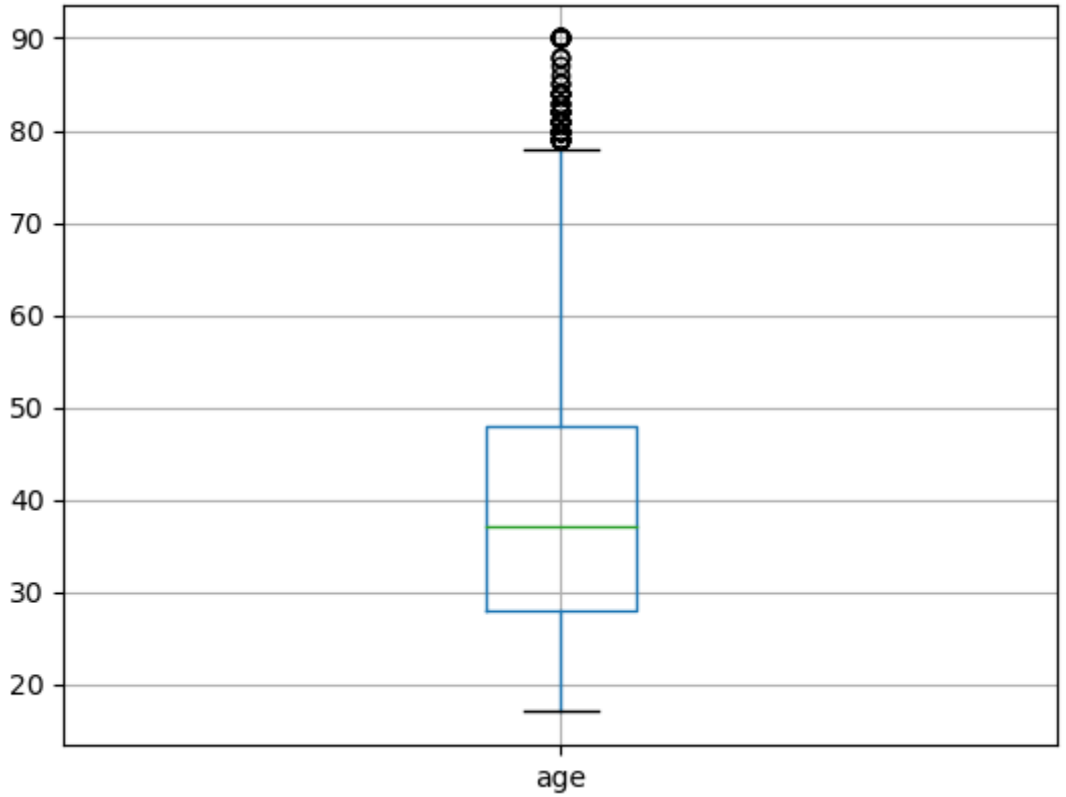
In [38]:
```
data['age'].hist(bins=100)
```

Out[38]: <AxesSubplot:>



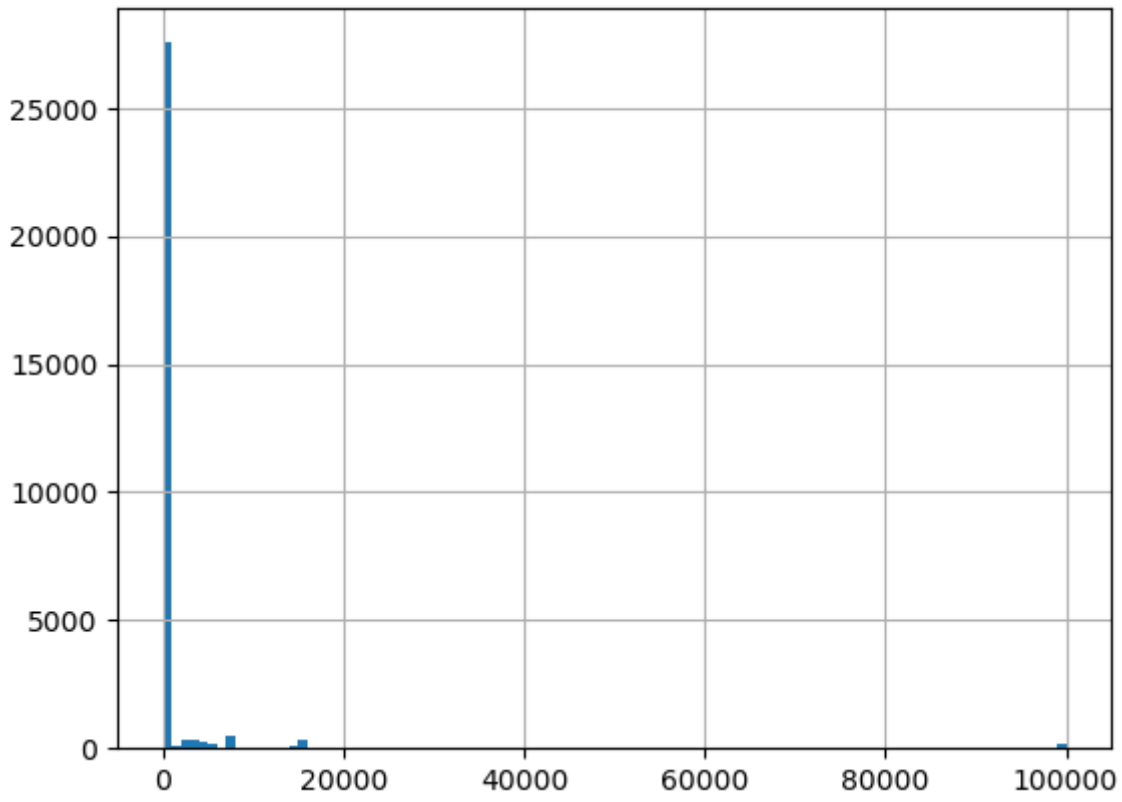
In [39]:
```
data.boxplot(column='age')
```

Out[39]: <AxesSubplot:>



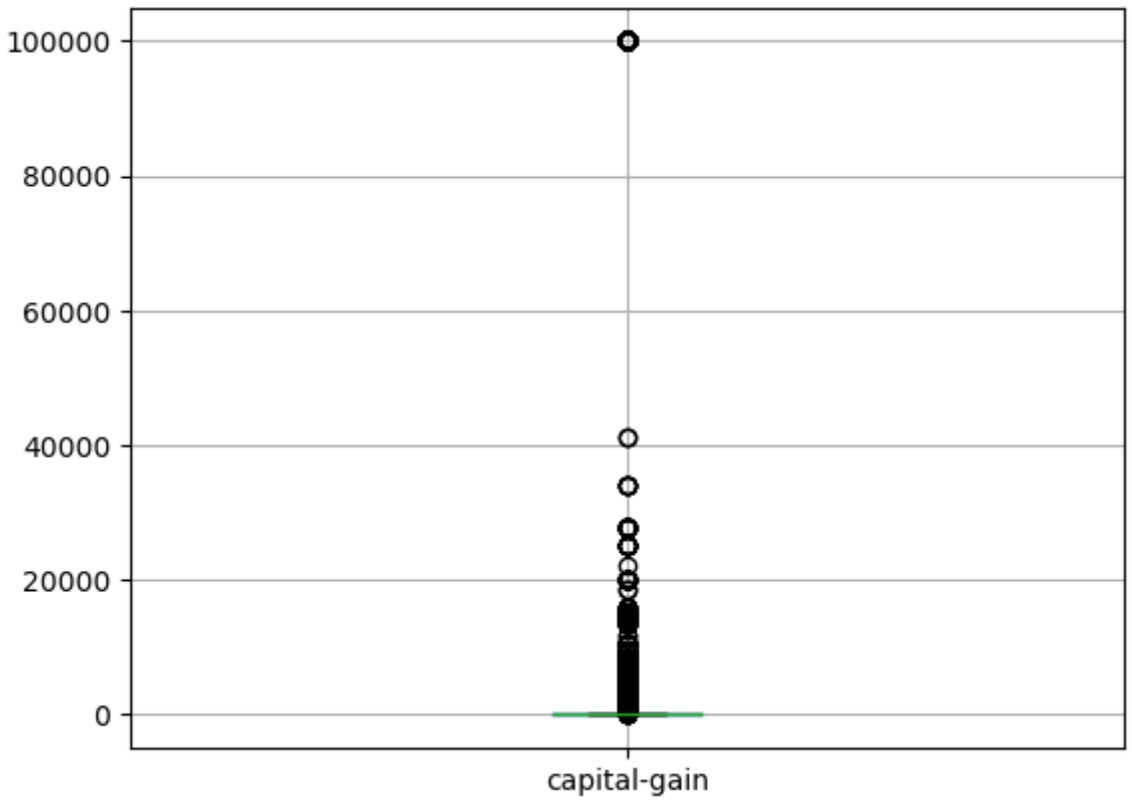
In [40]:
```
data['capital-gain'].hist(bins=100)
```

Out[40]: <AxesSubplot:>



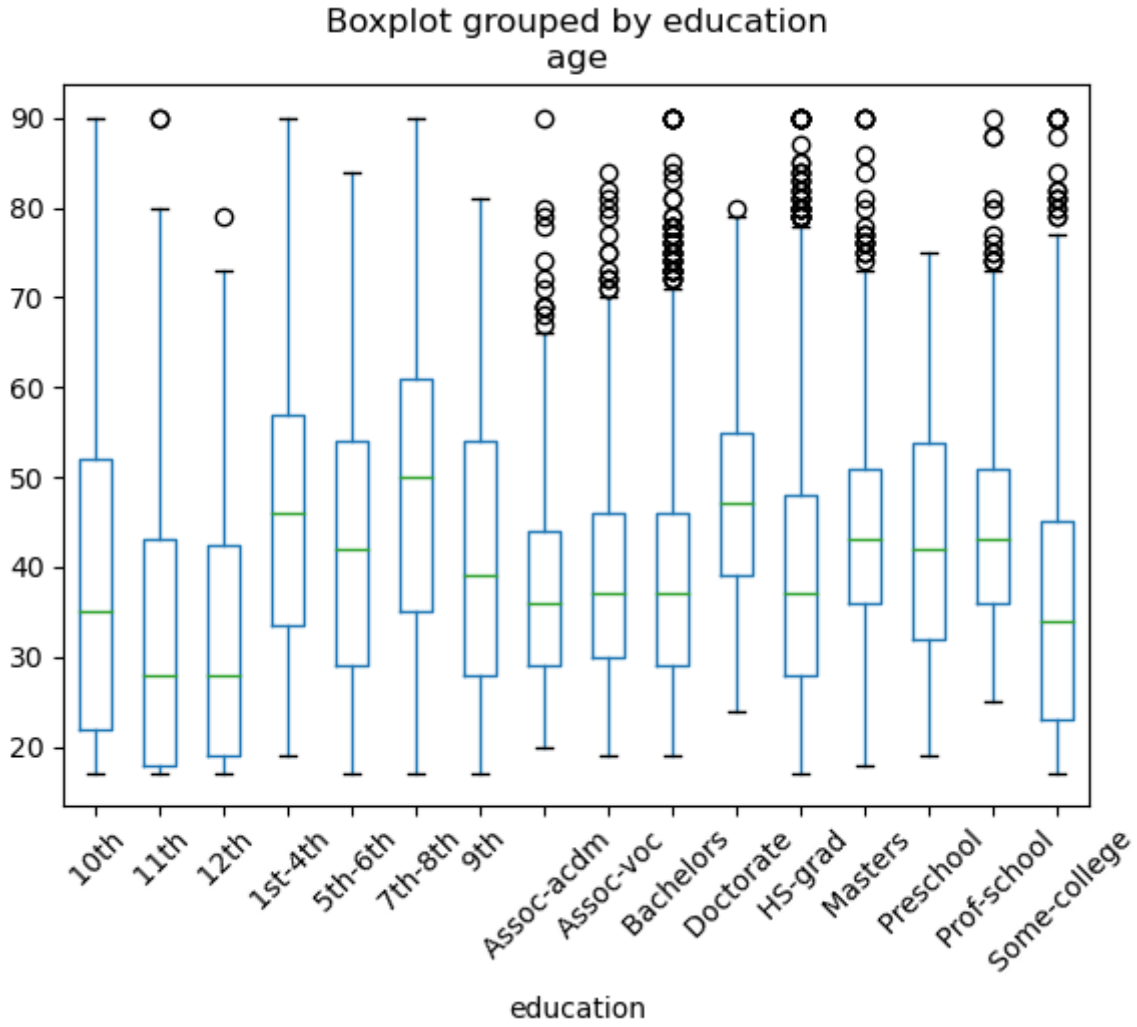
In [41]:
```
data.boxplot(column='capital-gain')
```

Out[41]: <AxesSubplot:>



In [42]:
```
data.boxplot(column='age', by = 'education', grid=False, rot = 45, fontsize = 10)
```

Out[42]:
```
<AxesSubplot:title={'center':'age'}, xlabel='education'>
```

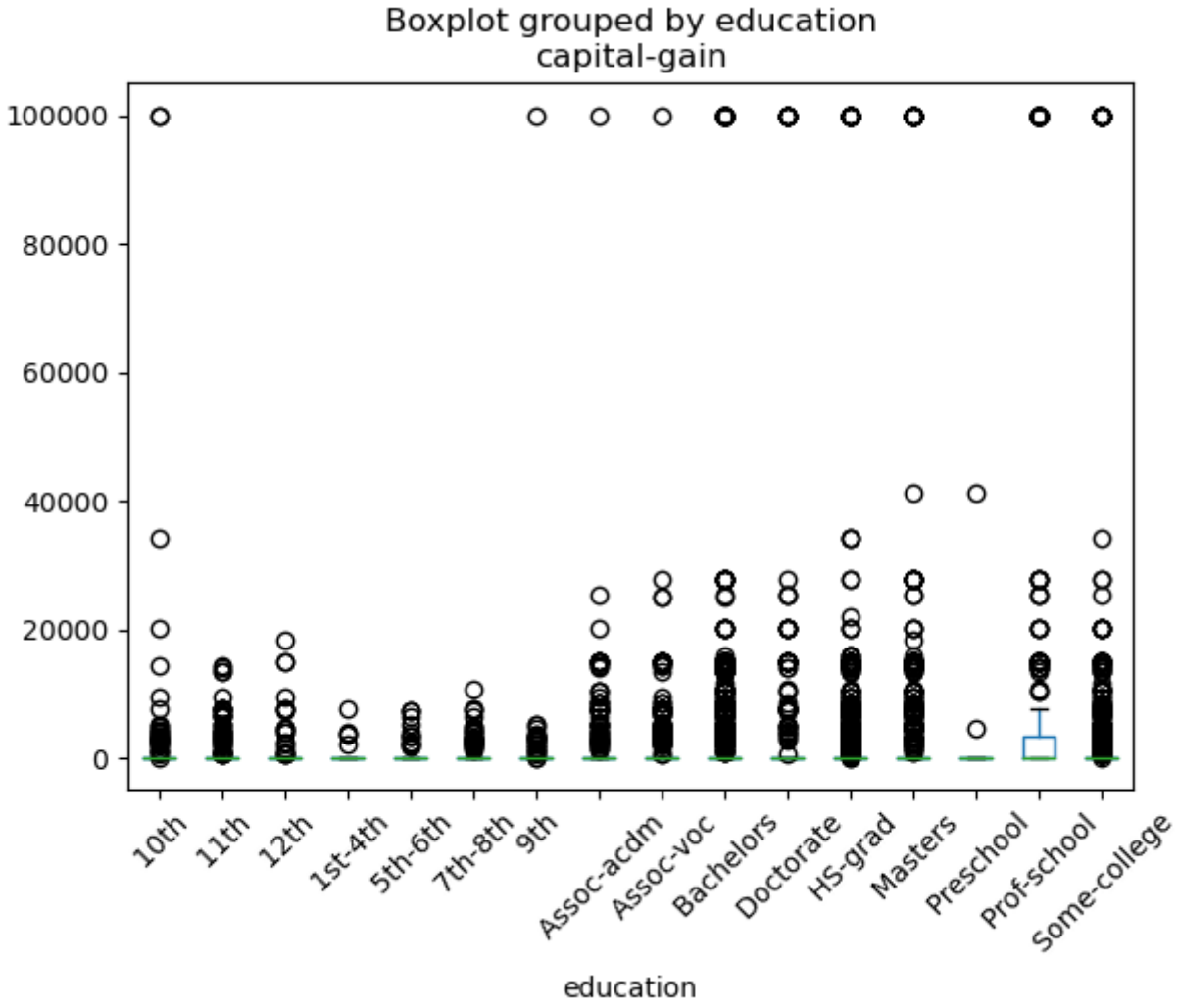


In [43]:
```
data['education'].value_counts()
```

Out[43]:
```
 HS-grad         9673
 Some-college    6710
 Bachelors       4938
 Masters         1586
 Assoc-voc       1295
 11th            1098
 Assoc-acdm       981
 10th             841
 7th-8th          589
 Prof-school      530
 9th              459
 12th             403
 Doctorate        383
 5th-6th          313
 1st-4th          155
 Preschool         46
Name: education, dtype: int64
```

In [44]:
```
data.boxplot(column='capital-gain', by = 'education', grid=False, rot = 45, fontsize = 10)
```

Out[44]:
```
<AxesSubplot:title={'center':'capital-gain'}, xlabel='education'>
```



In [45]:
```
data['marital-status'].value_counts()
```

Out[45]:
```
 Married-civ-spouse       13817
 Never-married             9830
 Divorced                  4084
 Separated                  945
 Widowed                    920
 Married-spouse-absent      382
 Married-AF-spouse           22
Name: marital-status, dtype: int64
```

In [46]:
```
data.apply(lambda x: sum(x.isnull()), axis = 0)
```

Out[46]:
```
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
```

In [47]:
```
from sklearn.preprocessing import LabelEncoder
```

In [48]:
```
data.head()
```

Out[48]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
11358	58	State-gov	HS-grad	9	Never-married	Adm-clerical	Not-in-family	White	Female	0	0	16	United-States	<=50K
10859	23	Local-gov	HS-grad	9	Never-married	Prof-specialty	Not-in-family	White	Female	0	0	40	United-States	<=50K
30948	41	Private	Assoc-voc	11	Married-civ-spouse	Sales	Husband	White	Male	4386	0	60	United-States	>50K
29811	58	Private	Some-college	10	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40	United-States	<=50K
18408	47	Self-emp-inc	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	99999	0	50	United-States	>50K

In [49]:
```
data.dtypes
```

Out[49]:
```
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 [50]:
```
columns = list(data.select_dtypes(exclude=['int64']))
```

In [51]:
```
columns
```

Out[51]:
```
['workclass',
 'education',
 'marital-status',
 'occupation',
 'relationship',
 'race',
 'sex',
 'native-country',
 'class-label']
```

In [52]:
```
data['class-label'].value_counts()
```

Out[52]:
```
 <=50K    22747
 >50K      7253
Name: class-label, dtype: int64
```

In [53]:
```
le = LabelEncoder()
for i in columns:
    #print(i)
    data[i] = le.fit_transform(data[i])
data.dtypes
```

Out[53]:
```
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 [54]:
```
data.head()
```

Out[54]:

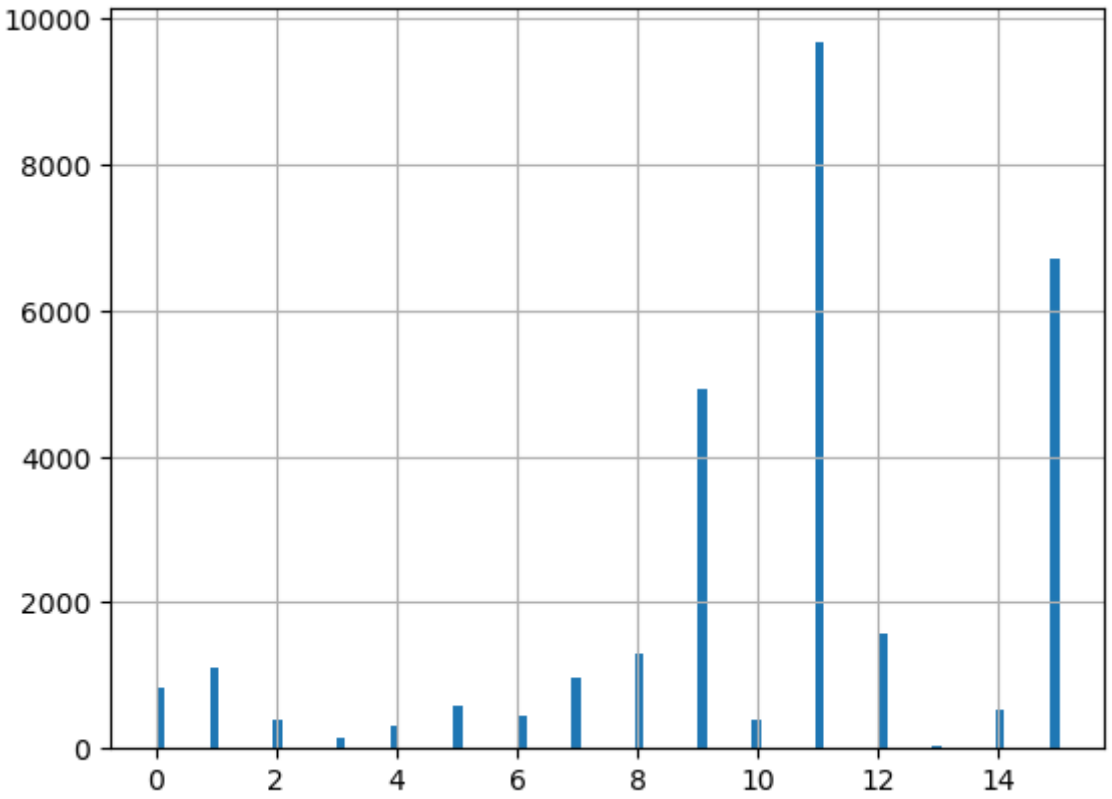
	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	class-label
11358	58	7	11	9	4	1	1	4	0	0	0	16	39	0
10859	23	2	11	9	4	10	1	4	0	0	0	40	39	0
30948	41	4	8	11	2	12	0	4	1	4386	0	60	39	1
29811	58	4	15	10	2	4	0	4	1	0	0	40	39	0
18408	47	5	14	15	2	10	0	4	1	99999	0	50	39	1

In [55]:
```
data['workclass'].value_counts()
```

Out[55]:
```
4    20910
6     2335
2     1929
0     1688
7     1203
5     1026
1      888
8       14
3        7
Name: workclass, dtype: int64
```

In [56]:
```
data['education'].hist(bins=100)
```

Out[56]: <AxesSubplot:>



In [57]:
```
data.describe(include='all')
```

Out[57]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-g
count	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000000	30000.000(
mean	38.586867	3.869333	10.297400	10.085067	2.612233	6.569100	1.448467	3.666100	0.667700	1095.360:
std	13.643079	1.456364	3.867521	2.570016	1.506014	4.228367	1.608674	0.848234	0.471046	7468.186:
min	17.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000(
25%	28.000000	4.000000	9.000000	9.000000	2.000000	3.000000	0.000000	4.000000	0.000000	0.000(
50%	37.000000	4.000000	11.000000	10.000000	2.000000	7.000000	1.000000	4.000000	1.000000	0.000(
75%	48.000000	4.000000	12.000000	12.000000	4.000000	10.000000	3.000000	4.000000	1.000000	0.000(
max	90.000000	8.000000	15.000000	16.000000	6.000000	14.000000	5.000000	4.000000	1.000000	99999.000(

Q6. Write a summary of the outcome of data.describe(). Data.describe() is used to generate the statistical decription of the data in the dataset. It tells us the number of values, the mean and the median, the minimum and maximum values, the standard deviation, 25th and 75th percentile. In this analysis, it helps us to know that the columns have 3000 rows, i.e. the number of values in each row is 3000, the mean and the median, standard deviation, 25th and 75th percentile of each attribute. For instance, considering the atribute 'age', the mean=38.586867 std=13.643079 minimum age =17 25% percentile =28 50% percentile (median) = 37 75% percentile =48 maximum age = 90.

Q7. What are the different data types (or attribut types) in data mining? Explain with the help of the examples from Adult dataset. HINT: Don't get confused with data types in Python or Pandas. Nominal data: This is categorical data that does not have any inherent order of ranking. In the adult data set, the attributes; 'occupation', 'workclass', 'marital-status', 'relationshp', ;race' are examples of nominal data. Ordinal data : This is categorical data that has an inherent order of ranking, but the differences between the values is no meaningful or cannot be determined. They can be arranged from the highest to the lowest, or from the least to the most, but it is not possible to measure the magnitude of the differences between these values. In the adult data set, we have 'education', as an example. Interval data : Refers to numerical data where the differences in values are meaningful but have no true zero point i.e. zero value does not necessarily mean the absence of the attribute being measured. In the adult data set, the columns 'education-num' , 'age' are examples of an interval data. Ratio: Refers to numerical data where the differences in values are meaningful but have a true zero point i.e. the value of zero means that their is complete absence of the attribute being measured. In the adlt data set, we have the following attibues as ratio type of data: capital-gain, capital-loss, hours-per-week/

Binary data refers to data that can take on only one or two values. For instance it can only be either a 'yes' or a 'no', or 'true' or 'false'. In the adult data set, the attribute 'sex' is binary data. sex is only either male or female. text: This is unstructured data in the form of a text. This can be in the form of reviews, comments etc. We do not have any text in the adult data set. Date/Time is also a type of data that tells us the date and the time. We do not have any column for this in the adult data set.

Q8. Highest migrants belongs to which country? The United States have the highest number of immigrants with 26885. #See line 33 for the code and the output generated for this answer.

Q9. Which occupation represents more males than females? The Armed Forces; Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Prof-specialty, Protective-serv, Sales, Tech-support and Transport-moving have more males than females as shown in the result of the code below. The Armed Forces have only Males, Craft-repair have 3568 males as against 205 females, Exec-managerial have 2684 Males, as against 1078 females, Farming-fishing have 859 Males as against 59 females; Handlers-cleaners 1099 Males as against 152 females , Machine-op-inspct have 1333 Males as against 508 females , Prof-specialty have 2409 Males as against 1392 females , Protective-ser have 534 Males as against 67 femalev , Sales have 2187 Males as against 1175 females , Tech-support have 537 Males as against 323 females; Transport-moving have 1379 Males as against 84 females. #line 34 shows the code and the output generated for this answer.

Q10. What is the difference between data.head() and data.tail()? data.head() is the code used to generate the first rows of a data , while data.tail() is used to generate the last rows of a data. By default, if a number is not specified in the bracket, the code returns only 5 rows.