Analyze the Pew.txt file

This time using separate function to read in the file.

In [18]:
```
import pandas as pd
%load_ext autoreload
%autoreload 2
from pew_data_analysis import *
pd.set_printoptions(max_columns=0)
```

In [19]:
```
%time pew = read_pew_txt('Pew.txt','data/Merge_Codebook.csv','data/US_FIPS_Codes.csv')
```

```
CPU times: user 21.17 s, sys: 3.20 s, total: 24.37 s
Wall time: 24.62 s
```

In [20]:
```
pew.describe()
```

Out[20]:

	id	rid	weight	year	date	age	fipsst	fipsco	density	regvoter
count	442262.000000	389963.000000	441892.000000	442262.000000	382454.000000	442262.000000	428020.000000	319442.000000	130895.000000	305512
mean	221131.500000	54188.247245	0.996082	2002.931262	408640.208807	48.735786	28.603262	82.707014	2.816555	1
std	255250.217899	153832.541757	0.500715	6.359258	2588051.467483	18.667399	15.723518	97.583575	1.393472	0
min	1.000000	1.000000	0.100000	1990.000000	0.000000	0.000000	1.000000	0.000000	1.000000	1
25%	110566.250000	1147.000000	0.649572	1998.000000	40397.000000	34.000000	13.000000	25.000000	2.000000	1
50%	221131.500000	4409.000000	0.881001	2004.000000	70212.000000	48.000000	29.000000	61.000000	3.000000	1
75%	331696.750000	100083.000000	1.204123	2008.000000	100301.000000	62.000000	42.000000	107.000000	4.000000	1
max	442262.000000	2020955.000000	5.824109	2013.000000	20010528.000000	99.000000	78.000000	840.000000	9.000000	1

In [57]:
```
pew.head(2)
```

Out[57]:

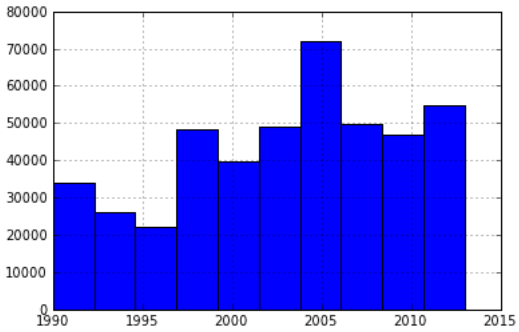
	id	rid	weight	year	date	survey	language	age	age2	sex	race	racethn	hisp	income	income2	educ	fipsst	state_name	fipsco	county_name
0	1	1	0.941828	1990	NaN	Jan90NII	English only	46	30-49	Female	White	NaN	NaN	NaN	Missing\not asked	College graduate	26	Michigan	NaN	NaN
91640	2	2	1.465066	1990	NaN	Jan90NII	English only	77	65+	Male	White	NaN	NaN	NaN	Missing\not asked	Less than high school	36	New York	NaN	NaN

Analyze each column

Dates

In [22]:
```
pew.year.hist()
```

Out[22]:
```
<matplotlib.axes.AxesSubplot at 0x19cb6410>
```



There are lots of surveys!

In [23]:
```
print len(pew.survey.value_counts())
print pew.survey.value_counts().head(2)
```

```
print pew.survey.value_counts().tail(2)
276
Decout00    5719
Typo99      3973
AprNII04     790
Janomni02    406
```

We could parse these *date* strings. They look like mddyy.

In [25]:
```
pew.date.value_counts()[:5]
```

Out[25]:
```
31901     1478
80101     1277
91299     1205
0         1107
101708     922
```

In [26]:
```
#pew['date_new'] = pew.date.apply(lambda d:
```

Demographics

In [27]:
```
pew.language.value_counts()
```

Out[27]:
```
English only           348561
English and Spanish     93701
```

These ages seem strange.

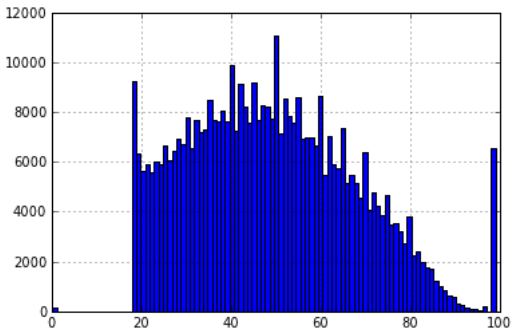
In [28]:
```
pew.age.value_counts()[:5]
```

Out[28]:
```
50    11054
40     9876
18     9210
45     9207
42     9137
```

There are spikes at regular intervals, e.g., 45 and 50, and at ages with special significance, e.g. 18. Perhaps some surveys were done on people with specific ages, or the people or the poll data collector was rounding down (or up).

In [29]:
```
pew.age.hist(bins=100)
```

Out[29]:
```
<matplotlib.axes.AxesSubplot at 0xac67970>
```



In [30]:
```
pew.age2.value_counts()
```

Out[30]:
```
30-49               159056
50-64               111068
65+                  88069
18-29                77350
DK\Refused            6563
Missing\not asked      156
```

In [31]:
```
pew.sex.value_counts()
```

Out[31]:
```
Female    230365
Male      211897
```

Females are *more* likely to answer DK\Refused.

In [32]:
```
pew[pew.age2 == 'DK\Refused'].sex.value_counts()
```

Out[32]:
```
Female    4293
Male      2270
```

In [33]:
```
pew.racethn.value_counts()
```

```
Out[33]: White non-Hisp    327168
         Black non-Hisp     39701
         Hispanic           28933
         Other              21586
         DK/Ref              5092
```

```
In [34]: pew.hisp.value_counts()
```

```
Out[34]: No            389830
         Yes            28933
         DK/Refused      3561
```

```
In [35]: pew.race.value_counts()
```

```
Out[35]: White          358296
         Black           43238
         Other/Mixed     22943
         Asian           10041
         DK/Refused       6457
```

Why are there two *income* columns?

```
In [36]: pew.income.value_counts()
```

```
Out[36]: $50,000 to $74,999    56115
         DK\Refused            51980
         $20,000 to $29,999    44149
         $30,000 to $39,999    42308
         $75,000 to $99,999    37979
         $40,000 to $49,999    36696
         $10,000 to $19,999    36216
         $100k to $149,999     32736
         less than $10,000     22826
         $150,000+             15229
```

```
In [37]: pew.income2.value_counts()
```

```
Out[37]: $75,000+              106621
         $30,000 to $49,999     79004
         Missing\not asked      77028
         less than $20,000      59042
         $50,000 to $74,999     56115
         $20,000 to $29,999     44149
         DK\Refused             20303
```

```
In [40]: pew.educ.value_counts()
```

```
Out[40]: High school graduate          127171
         Some College                  109343
         College graduate               91640
         Post-graduate                  53238
         High school, incomplete        28952
         Business, Technical, Trade     14465
         Less than high school           8540
         Post-graduate degree            5036
         DK/Refused                      2594
```

```
In [39]: pew.head(2)
```

Out[39]:

	id	rid	weight	year	date	survey	language	age	age2	sex	race	racethn	hisp	income	income2	educ	fipsst	state_name	fipsco	county_name
0	1	1	0.941828	1990	NaN	Jan90NII	English only	46	30-49	Female	White	NaN	NaN	NaN	Missing\not asked	College graduate	26	Michigan	NaN	NaN
91640	2	2	1.465066	1990	NaN	Jan90NII	English only	77	65+	Male	White	NaN	NaN	NaN	Missing\not asked	Less than high school	36	New York	NaN	NaN

Locations

States, then counties, with most and least responses.

```
In [47]: print pew.state_name.value_counts()[:3]
         print pew.state_name.value_counts()[-3:]
```

```
California    40684
Texas         27152
New York      24533
District of Columbia    850
Hawaii                  150
```

```
Alaska                  124
```

In [48]:
```
print pew.county_name.value_counts()[:3]
print pew.county_name.value_counts()[-3:]
```

```
Los Angeles    6718
Cook           4083
Jefferson      4060
Griggs            1
Issaquena         1
Mellette          1
```

Mostly suburbanites.

In [44]:
```
pew.usr.value_counts()
```

Out[44]:
```
Suburban     139778
Urban         82713
Rural         60954
```

According to the Merge Codebook document, the scale goes from lowest population densith (1) to greatest (5). Not sure what a *9* indicates; no data?

In [45]:
```
pew.density.value_counts()
```

Out[45]:
```
1    30252
2    28749
3    27409
4    24141
5    20241
9      103
```

Politics

In [49]:
```
pew.party.value_counts()
```

Out[49]:
```
Independent      140659
Democrat         140233
Republican       127981
No Preference     14636
DK                11067
Other              2312
```

In [50]:
```
pew.partyln.value_counts()
```

Out[50]:
```
Missing/not asked    262804
Lean Democrat         59109
Lean Republican       55178
Other/DK              51246
```

Fairly well balanced between those leaning Dem and those leaning Rep.

In [51]:
```
pew.partysum.value_counts()
```

Out[51]:
```
Dem/ln D                          199342
Rep/ln R                          183159
No leaning                         54387
Party or Partyln not available      5374
```

All are registered voters?

In [52]:
```
pew.regvoter.value_counts()
```

Out[52]:
```
1    305512
```

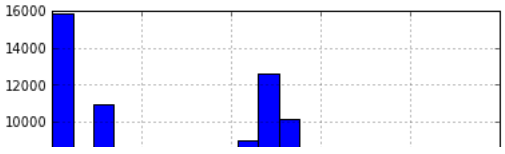
In [53]:
```
print len(pew.regvoter)
print len(pew[pew.regvoter.isnull()])
```

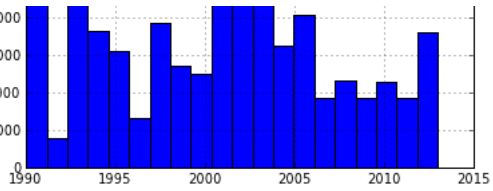
```
442262
136750
```

The number of responses with *regvoter* seems to be increasing over the years compared to the number with *regvoter* missing.

In [54]:
```
pew[pew.regvoter.isnull()].year.hist(bins=20)
```

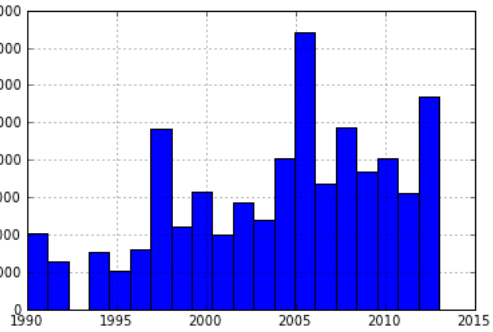
Out[54]:
```
<matplotlib.axes.AxesSubplot at 0x5483cd0>
```





In [56]:
```
pew[pew.regvoter.notnull()].year.hist(bins=20)
```

Out[56]: <matplotlib.axes.AxesSubplot at 0x54d5ad0>



Write to CSV

In [58]:
```
%time pew.to_csv('data/Pew_for_analysis.csv', index=False)
```

In [ ]:
```

```