Read the Pew.txt file as analyze some of the data

We have a Pew.txt file that was exported from the SPSS file from Paul.

See if we can parse it and visualize some of the data.

In [4]:
```
import pandas as pd
pd.set_printoptions(max_columns=0)
```

```
/Library/Frameworks/Python.framework/Versions/7.3/lib/python2.7/site-packages/pandas/core/format.py:1286: FutureWarning: set_printoptions
is deprecated, use set_option instead
  FutureWarning)
```

Read into dataframe.

In [2]:
```
pew = pd.read_csv('Pew.txt',delimiter=' ')
```

In [5]:
```
pew.head(2)
```

Out[5]:

	id	rid	weight	year	survey	date	language	age	age2	sex	race	hisp	racethn	educ	income	income2	fips	state	usr	density	party	partyln	partysum
0	1	1	0.941828	1990	Jan90NII	NaN	English only	46	30-49	Female	1	NaN	NaN	6	NaN	Missing\not asked	NaN	26		NaN	3	NaN	No leaning
1	2	2	1.465066	1990	Jan90NII	NaN	English only	77	65+	Male	1	NaN	NaN	1	NaN	Missing\not asked	NaN	36		NaN	1	NaN	Rep/ln R

Some basic summary statistics.

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

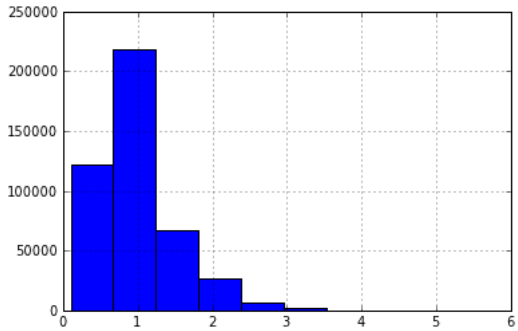
Out[10]:

	id	rid	weight	year	date	age	race	hisp	educ	fips
count	442262.000000	389963.000000	441892.000000	442262.000000	382454.000000	442262.000000	440983.000000	422327.000000	440980.000000	319442.000000
mean	221131.500000	54188.247245	0.996082	2002.931262	408640.208807	48.735786	1.416916	1.990538	4.623049	28640.999286
std	255250.217899	153832.541757	0.500715	6.359258	2588051.467483	18.667399	1.192242	0.694090	1.683050	15797.612652
min	1.000000	1.000000	0.100000	1990.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1000.000000
25%	110566.250000	1147.000000	0.649572	1998.000000	40397.000000	34.000000	1.000000	2.000000	3.000000	13151.000000
50%	221131.500000	4409.000000	0.881001	2004.000000	70212.000000	48.000000	1.000000	2.000000	5.000000	29071.000000
75%	331696.750000	100083.000000	1.204123	2008.000000	100301.000000	62.000000	1.000000	2.000000	6.000000	42011.000000
max	442262.000000	2020955.000000	5.824109	2013.000000	20010528.000000	99.000000	9.000000	9.000000	9.000000	78010.000000

Looking at the data

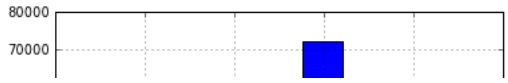
In [21]:
```
pew.weight.hist()
```

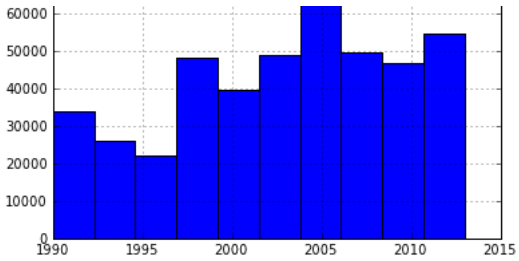
Out[21]: <matplotlib.axes.AxesSubplot at 0x108d450>



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

Out[22]: <matplotlib.axes.AxesSubplot at 0x4dc3730>





There are lots of surveys!

In [29]:
```
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 [13]:
```
pew.date.value_counts()[:5]
```

Out[13]:
```
31901     1478
80101     1277
91299     1205
0         1107
101708     922
```

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

Out[30]:
```
English only           348561
English and Spanish     93701
```

These ages seem strange.

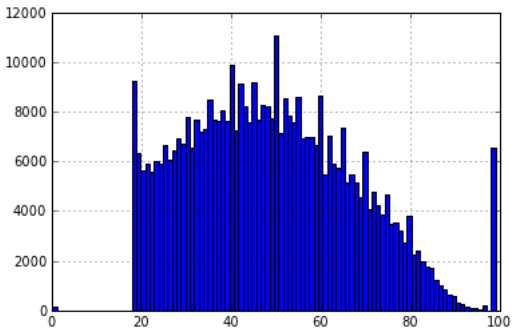
In [16]:
```
pew.age.value_counts()[:5]
```

Out[16]:
```
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 [18]:
```
pew.age.hist(bins=100)
```

Out[18]:
```
<matplotlib.axes.AxesSubplot at 0x4cf6570>
```



In [8]:
```
pew.racethn.value_counts()
```

Out[8]:
```
White non-Hisp    327168
Black non-Hisp     39701
Hispanic           28933
Other              21586
DK/Ref              5092
```

It would be good to have dictionary for these category numbers into a file that could be joined with this Pew.txt file so we know what the values, e.g., *2*, mean.

In [20]:
```
pew.hisp.value_counts()
```

Out[20]:
```
2    389830
1     28933
9      3561
8         2
0         1
```

Why are there two *income* columns?

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

Out[31]:
```
$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 [32]:
```
pew.income2.value_counts()
```

Out[32]:
```
$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 [35]:
```
print pew.fips.value_counts().head(2)
print pew.fips.value_counts().tail(2)
```

```
6037     6718
17031    4058
2068    1
2050    1
```

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

Out[36]:
```
     152070
S    139778
U     82713
R     60954
2      3327
1      1891
3      1529
```

In [37]:
```
pew.head()
```

Out[37]:

	id	rid	weight	year	survey	date	language	age	age2	sex	race	hisp	racethn	educ	income	income2	fips	state	usr	density	party	partyln	partysum
0	1	1	0.941828	1990	Jan90NII	NaN	English only	46	30-49	Female	1	NaN	NaN	6	NaN	Missing\not asked	NaN	26		NaN	3	NaN	No leaning
1	2	2	1.465066	1990	Jan90NII	NaN	English only	77	65+	Male	1	NaN	NaN	1	NaN	Missing\not asked	NaN	36		NaN	1	NaN	Rep/ln R
2	3	3	0.812558	1990	Jan90NII	NaN	English only	29	18-29	Male	2	NaN	NaN	6	NaN	Missing\not asked	NaN	24		NaN	2	NaN	Dem/ln D
3	4	4	1.126500	1990	Jan90NII	NaN	English only	86	65+	Male	1	NaN	NaN	2	NaN	Missing\not asked	NaN	17		NaN	9	NaN	No leaning
4	5	5	2.406894	1990	Jan90NII	NaN	English only	39	30-49	Male	1	NaN	NaN	5	NaN	Missing\not asked	NaN	34		NaN	3	NaN	No leaning

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

Out[39]:
```
3    140659
2    140233
```

In [41]:
```
pew.partyln.value_counts().head(2)
```

Out[41]:
```
0    262804
2     59109
```

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

In [42]:
```
pew.partysum.value_counts()
```

Out[42]:
```
Dem/ln D                199342
Rep/ln R                183159
No leaning               54387
```

```
Party or Partyln not available      5374
```

All are registered voters?

In [43]:
```
pew.regvoter.value_counts()
```

Out[43]:
```
1    305512
```