In [1]:

```
#I think this data set will be much more interesting than the plane crashes one (also more nicely formatted)
#This data set contains data for all of the cities and states in the union, and many of their population demographic features
#over the last several years

#There is no missing data
#The Data is a combination of strings (State Name and County Name), and intergers (everything else)
#http://www.census.gov/popest/data/counties/totals/2014/files/CO-EST2014-alldata.pdf (link to descriptions of column headers)
```

In [85]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import vincent #rapid mapping tool

%matplotlib inline
```

In [95]:

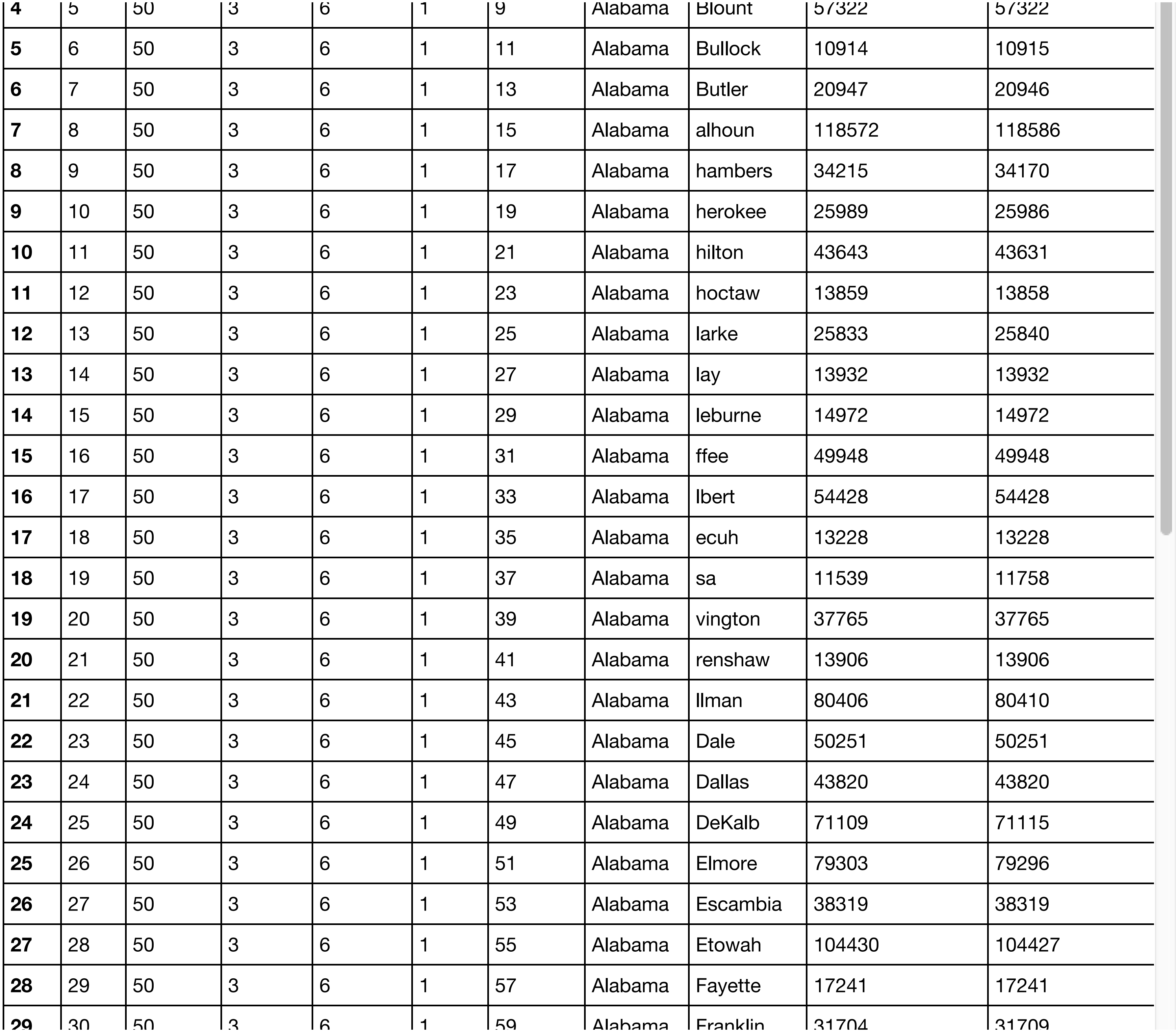
```
data = pd.read_table("population_data_2014.csv", sep = ',', parse_dates = True)
state_only = pd.DataFrame(columns = data.columns)
counties_only = pd.DataFrame(columns = data.columns)
for index, row in data.iterrows():
    if row.STNAME == row.CTYNAME and row.STNAME != "District of Columbia":
        #print "True!"
        state_only = state_only.append(row)
    elif row.STNAME != row.CTYNAME and row.STNAME != "District of Columbia":
        counties_only = counties_only.append(row)
```

In [138]:

```
counties_only
```

Out[138]:

	index	SUMLEV	REGION	DIVISION	STATE	COUNTY	STNAME	CTYNAME	CENSUS2010POP	ESTIMATESBASE
0	1	50	3	6	1	1	Alabama	Autauga	54571	54571
1	2	50	3	6	1	3	Alabama	Baldwin	182265	182265
2	3	50	3	6	1	5	Alabama	Barbour	27457	27457
3	4	50	3	6	1	7	Alabama	Bibb	22915	22919



4	5	50	3	6	1	9	Alabama	Blount	57322	57322
5	6	50	3	6	1	11	Alabama	Bullock	10914	10915
6	7	50	3	6	1	13	Alabama	Butler	20947	20946
7	8	50	3	6	1	15	Alabama	alhoun	118572	118586
8	9	50	3	6	1	17	Alabama	hambers	34215	34170
9	10	50	3	6	1	19	Alabama	herokee	25989	25986
10	11	50	3	6	1	21	Alabama	hilton	43643	43631
11	12	50	3	6	1	23	Alabama	hoctaw	13859	13858
12	13	50	3	6	1	25	Alabama	larke	25833	25840
13	14	50	3	6	1	27	Alabama	lay	13932	13932
14	15	50	3	6	1	29	Alabama	leburne	14972	14972
15	16	50	3	6	1	31	Alabama	ffee	49948	49948
16	17	50	3	6	1	33	Alabama	lbert	54428	54428
17	18	50	3	6	1	35	Alabama	ecuh	13228	13228
18	19	50	3	6	1	37	Alabama	sa	11539	11758
19	20	50	3	6	1	39	Alabama	vington	37765	37765
20	21	50	3	6	1	41	Alabama	renshaw	13906	13906
21	22	50	3	6	1	43	Alabama	llman	80406	80410
22	23	50	3	6	1	45	Alabama	Dale	50251	50251
23	24	50	3	6	1	47	Alabama	Dallas	43820	43820
24	25	50	3	6	1	49	Alabama	DeKalb	71109	71115
25	26	50	3	6	1	51	Alabama	Elmore	79303	79296
26	27	50	3	6	1	53	Alabama	Escambia	38319	38319
27	28	50	3	6	1	55	Alabama	Etowah	104430	104427
28	29	50	3	6	1	57	Alabama	Fayette	17241	17241
29	30	50	3	6	1	59	Alabama	Franklin	31704	31709

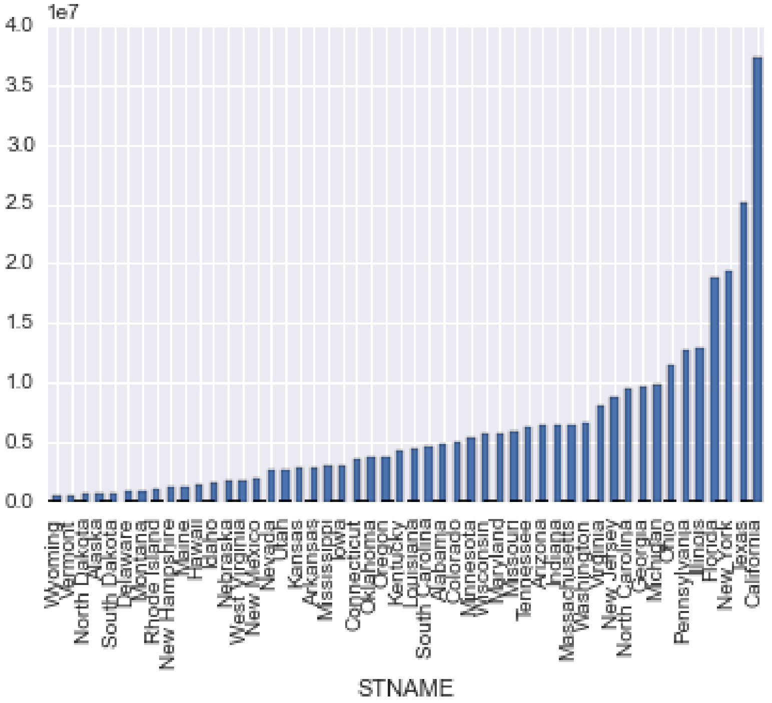
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Alabama	Franklin	[illegible]	[illegible]
...	...	...	...	...	...	...	...	...	...	...
3111	3162	50	2	3	55	129	Wisconsin	Washburn	15911	15911
3112	3163	50	2	3	55	131	Wisconsin	Washington	131887	131885

In [135]:

```
state_only = state_only.set_index(state_only.STNAME).sort(['CENSUS2010POP'])
state_only.CENSUS2010POP.plot(kind = 'bar')
#the most populus state is California, by quite a large margin
```

Out[135]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11016eb50>
```



In [96]:

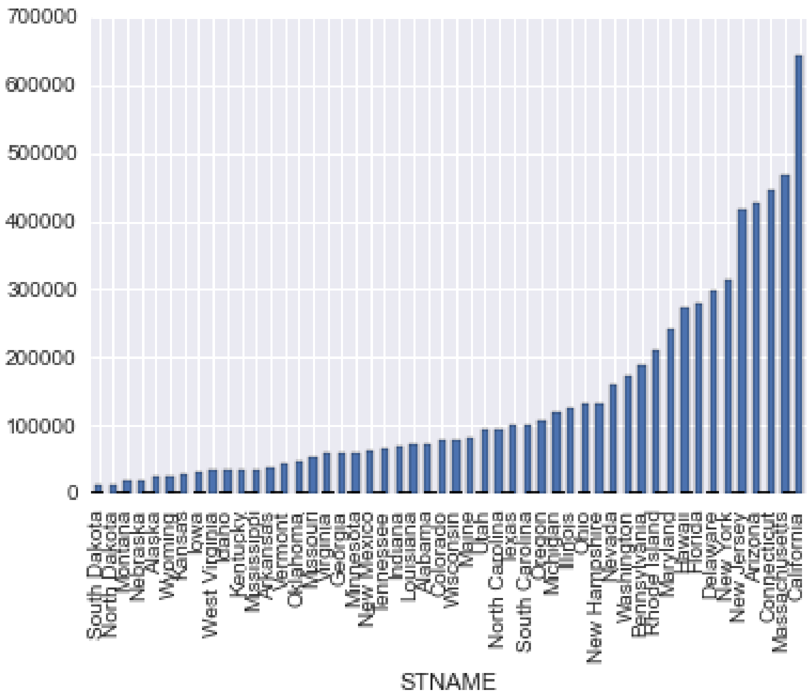
```
counties_only = counties_only.reset_index(drop=False)
counties_only.CTYNAME = counties_only.CTYNAME.map(lambda x: x.strip('County'))
```

In [137]:
```
counties_only.columns
```

Out[137]:
```
Index([u'index', u'SUMLEV', u'REGION', u'DIVISION', u'STATE', u'COUNTY', u'STNAME', u'CTYNAME', u'C
ENSUS2010POP', u'ESTIMATESBASE2010', u'POPESTIMATE2010', u'POPESTIMATE2011', u'POPESTIMATE2012',
u'POPESTIMATE2013', u'POPESTIMATE2014', u'NPOPCHG_2010', u'NPOPCHG_2011', u'NPOPCHG_2012', u'NPOPCH
G_2013', u'NPOPCHG_2014', u'BIRTHS2010', u'BIRTHS2011', u'BIRTHS2012', u'BIRTHS2013', u'BIRTHS201
4', u'DEATHS2010', u'DEATHS2011', u'DEATHS2012', u'DEATHS2013', u'DEATHS2014', u'NATURALINC2010',
u'NATURALINC2011', u'NATURALINC2012', u'NATURALINC2013', u'NATURALINC2014', u'INTERNATIONALMIG201
0', u'INTERNATIONALMIG2011', u'INTERNATIONALMIG2012', u'INTERNATIONALMIG2013', u'INTERNATIONALMIG20
14', u'DOMESTICMIG2010', u'DOMESTICMIG2011', u'DOMESTICMIG2012', u'DOMESTICMIG2013', u'DOMESTICMIG2
014', u'NETMIG2010', u'NETMIG2011', u'NETMIG2012', u'NETMIG2013', u'NETMIG2014', u'RESIDUAL2010',
u'RESIDUAL2011', u'RESIDUAL2012', u'RESIDUAL2013', u'RESIDUAL2014', u'GQESTIMATESBASE2010', u'GQEST
IMATES2010', u'GQESTIMATES2011', u'GQESTIMATES2012', u'GQESTIMATES2013', u'GQESTIMATES2014', u'RBIR
TH2011', u'RBIRTH2012', u'RBIRTH2013', u'RBIRTH2014', u'RDEATH2011', u'RDEATH2012', u'RDEATH2013',
u'RDEATH2014', u'RNATURALINC2011', u'RNATURALINC2012', u'RNATURALINC2013', u'RNATURALINC2014', u'RI
NTERNATIONALMIG2011', u'RINTERNATIONALMIG2012', u'RINTERNATIONALMIG2013', u'RINTERNATIONALMIG2014',
u'RDOMESTICMIG2011', u'RDOMESTICMIG2012', u'RDOMESTICMIG2013', u'RDOMESTICMIG2014', u'RNETMIG2011',
u'RNETMIG2012', u'RNETMIG2013', u'RNETMIG2014'], dtype='object')
```

In [136]:
```
counties_only.groupby(counties_only.STNAME).mean().sort(['CENSUS2010POP']).CENSUS2010POP.plot(kind
= 'bar')
#california also has, by far, the most populous county(ies) in the union, on average
```

```
Out[136]: <matplotlib.axes._subplots.AxesSubplot at 0x1169082d0>
```



In [101]:

In [ ]:

```
#the .pdf in the folder has all the more informative headers for these columns
#questions to ask:
    #where should you invest in real-estate based on people moving in?
    #Where are populations growing the fastest?
    #does immigration have a great affect on population increase?
    #which state has the most people moving around within the state?
    #In which state/county is the death rate outpacing the birth rate? Is the population declining
or are there enough
    #immigrants to support the population?
```

In [ ]: