In [1]:
```
# Dependencies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# import locale
# import requests
# import os
import csv
import scipy.stats as stats
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Collecting base data...
temp_df = pd.read_csv("Resources/election_all.csv")
pop_df = pd.read_csv("Resources/population2000-2016ALL.csv")
v2000 = pd.read_csv("Resources/FOR_JINHO_2000.csv")
v2004 = pd.read_csv("Resources/FOR_JINHO_2004.csv")
v20081 = pd.read_csv("Resources/FOR_JINHO_2008.csv")
v2012 = pd.read_csv("Resources/FOR_JINHO_2012.csv")
v2016 = pd.read_csv("Resources/FOR_JINHO_2016.csv")
year = '2000-2016'
```

In [3]:
```
overall = pd.merge(v2000,v2004, on='Location').merge(v20081, on='Location')
overall1 = overall.merge(v2012, on='Location').merge(v2016, on='Location')
overall1.columns = ['Location', 'total_votes_2000', '2000_republican_votes','2000_de
```

In [4]:
```
overall1['total_votes'] = overall1['total_votes_2000']+overall1['total_votes_2004']+
overall1['rep'] = overall1['2000_republican_votes']+overall1['2004_republican_votes'
overall1['dem'] = overall1['2000_democrat_votes']+overall1['2004_democrat_votes']+ov
overall1.head()
```

Out[4]:

	Location	total_votes_2000	2000_republican_votes	2000_democrat_votes	total_votes_2004	2004_republican_vo
0	Alabama, Autauga	17208	11993	4942	20081	15
1	Alabama, Baldwin	56480	40872	13997	69320	52
2	Alabama, Barbour	10395	5096	5188	10777	5
3	Alabama, Bibb	7101	4273	2710	7600	5
4	Alabama, Blount	17973	12667	4977	21504	17

In [5]:
```
overall2 = overall1[['Location','total_votes', 'rep', 'dem']]
overall2.columns = ['Location','total_votes','2008_democrat_votes','2008_republican_
overall2
v2008 = overall2
v2008.head()
```

Out[5]:

	Location	total_votes	2008_democrat_votes	2008_republican_votes
0	Alabama, Autauga	109835	80143.0	28092.0
1	Alabama, Baldwin	387766	294013.0	85864.0
2	Alabama, Barbour	54780	27865.0	26500.0
3	Alabama, Bibb	40584	28877.0	11174.0
4	Alabama, Blount	113338	94058.0	17563.0

In [6]:
```
# Creating master State, County details...
master_location_df = temp_df[['state','county','state_po']]
master_location_df = master_location_df.drop_duplicates()
master_location_df['Location'] = master_location_df.state+", "+master_location_df.cc
master_location_df.columns = ['State', 'County', 'State_po', 'Location']
master_location_df = master_location_df.dropna()
master_location_df.to_csv(r'Resources/master_state_info.csv', index=False, header=Tr
print(len(master_location_df))
master_location_df.head()
```

```
3150
```

Out[6]:

	State	County	State_po	Location
0	Alabama	Autauga	AL	Alabama, Autauga
4	Alabama	Baldwin	AL	Alabama, Baldwin
8	Alabama	Barbour	AL	Alabama, Barbour
12	Alabama	Bibb	AL	Alabama, Bibb
16	Alabama	Blount	AL	Alabama, Blount

In [7]:
```
# Collecting required fields...
# pop_df = pop_df[['Location','2000','2004','2008','2012','2016']]
pop_df = pop_df[['Location','2008']]
# print(len(pop_df))
# pop_df.head()
pop_final = pd.merge(pop_df, master_location_df, on='Location')
pop_final
# Creating summary data by state...
pop_state_summary = pop_final.groupby('State')
pop_state_summary = pop_state_summary.sum()

pop_county_summary = pop_final.groupby('Location')
pop_county_summary = pop_county_summary.sum()
pop_county_summary.head()
len(pop_county_summary)
```

Out[7]:
```
3001
```

In [8]:
```
# Creating summary for State...
v2008_final = pd.merge(v2008, master_location_df, on='Location')
v2008_final['2008_other_votes'] = v2008_final["total_votes"]-v2008_final["2008_democ
v2008_state_summary = v2008_final.groupby('State')
v2008_state_summary = v2008_state_summary.sum()
# v2008_state_summary.head()
v2008_state_summary['year'] = '2008'
year2 = v2008_state_summary['year'].iloc[0]
# print(v2008_state_summary['total_votes'].min()/1000, v2008_state_summary['total_vo
#
# merge population data to calcuate avg.
#
v2008_state_summary = v2008_state_summary.merge(pop_state_summary, on='State')
v2008_state_summary['avg_votes'] = round(v2008_state_summary['total_votes']*100/v200
v2008_state_summary['2008_democrat_votes'] = round(v2008_state_summary['2008_democra
v2008_state_summary['2008_republican_votes'] = round(v2008_state_summary['2008_repub
v2008_state_summary['2008_other_votes'] = round(v2008_state_summary['2008_other_vote
hv_st = v2008_state_summary['avg_votes'].max()
highst = v2008_state_summary.loc[v2008_state_summary['avg_votes'] == hv_st]
hv_st = highst.index.tolist()
lv_st = v2008_state_summary['avg_votes'].min()
lowst = v2008_state_summary.loc[v2008_state_summary['avg_votes'] == lv_st]
lv_st = lowst.index.tolist()
v2008_state_summary.head()
```

Out[8]:

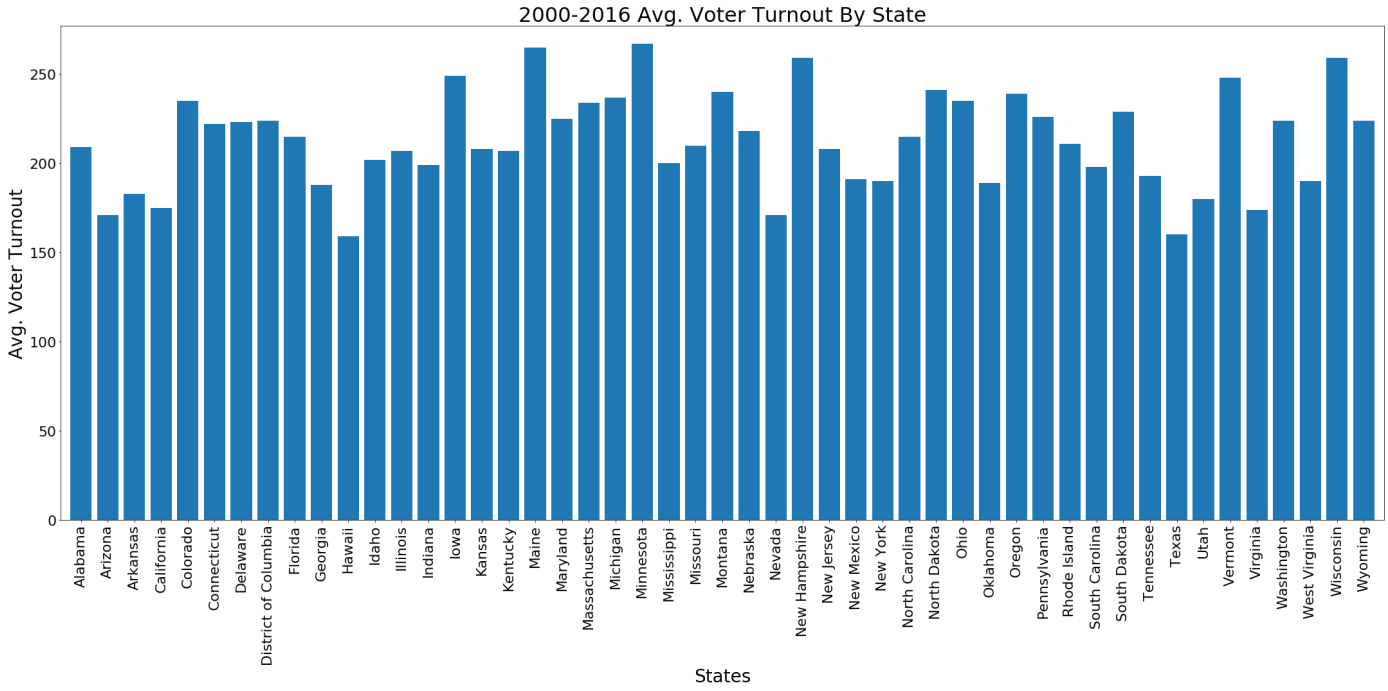
State	total_votes	2008_democrat_votes	2008_republican_votes	2008_other_votes	year	2008	avg_vote
Alabama	9849505	126.0	79.0	3.0	2008	4718206	209.
Arizona	10768183	89.0	76.0	6.0	2008	6280362	171.
Arkansas	5261456	105.0	73.0	5.0	2008	2874554	183.
California	64131576	67.0	102.0	6.0	2008	36604337	175.
Colorado	11498639	110.0	115.0	10.0	2008	4889730	235.

In [9]:
```
# Creating summary for Counties..
v2008_county_summary = v2008_final.groupby('Location', as_index=False)
v2008_county_summary = v2008_county_summary.sum()
# print(v2008_county_summary['total_votes'].min(), v2008_county_summary['total_votes
v2008_county_summary = v2008_county_summary.merge(pop_county_summary, on='Location')
v2008_county_summary['avg_votes'] = round(v2008_county_summary['total_votes']*100/v2
v2008_county_summary['2008_democrat_votes'] = round(v2008_county_summary['2008_democ
v2008_county_summary['2008_republican_votes'] = round(v2008_county_summary['2008_rep
v2008_county_summary['2008_other_votes'] = round(v2008_county_summary['2008_other_vo
hv_cy = v2008_county_summary['avg_votes'].max()
highcy = v2008_county_summary.loc[v2008_county_summary['avg_votes'] == hv_cy]
hv_cy = highcy.Location.tolist()
lv_cy = v2008_county_summary['avg_votes'].min()
lowcy = v2008_county_summary.loc[v2008_county_summary['avg_votes'] == lv_cy]
lv_cy = lowcy.Location.tolist()
v2008_county_summary.head()
```

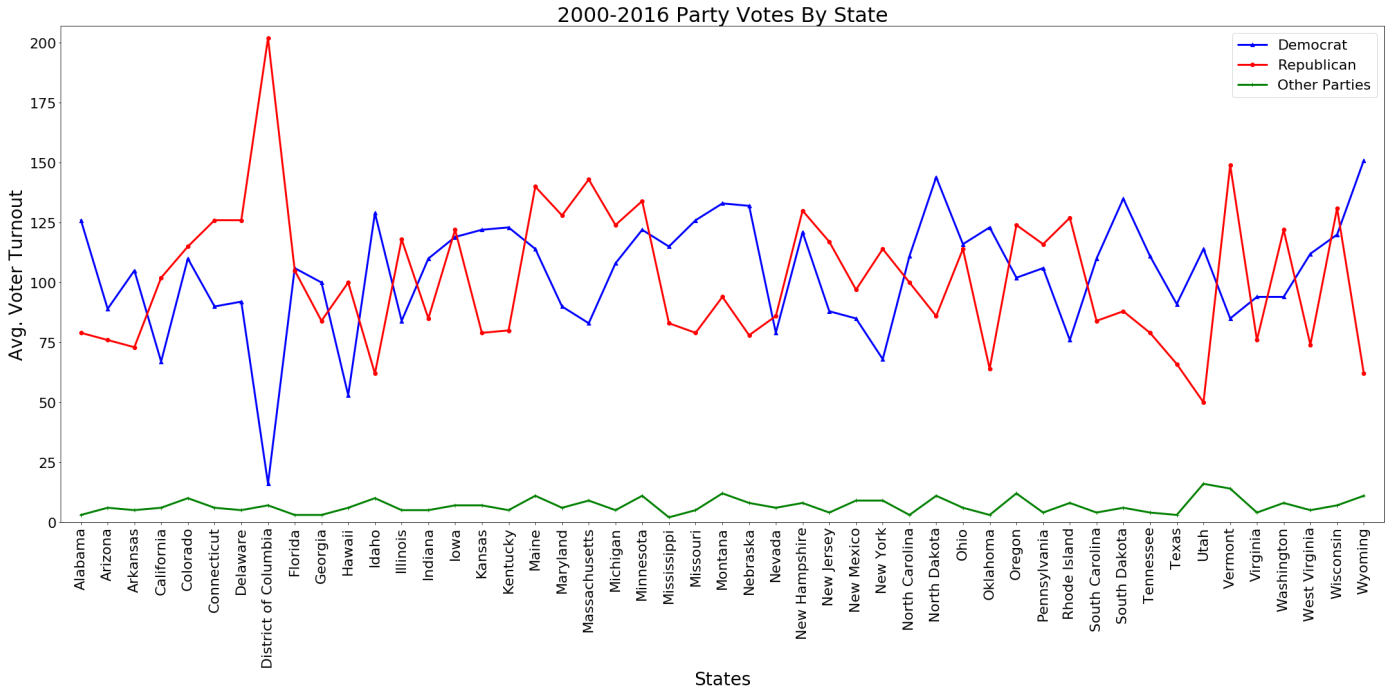
Out[9]:

	Location	total_votes	2008_democrat_votes	2008_republican_votes	2008_other_votes	2008	avg_votes
0	Alabama, Autauga	109835	150.0	53.0	3.0	53277	206.0
1	Alabama, Baldwin	387766	167.0	49.0	4.0	175827	221.0
2	Alabama, Barbour	54780	100.0	95.0	1.0	27808	197.0
3	Alabama, Bibb	40584	127.0	49.0	2.0	22705	179.0
4	Alabama, Blount	113338	165.0	31.0	3.0	57055	199.0

In [10]:
```
# Voter turnout by state...
#
title = year+' Avg. Voter Turnout By State'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
plt.figure(figsize=(30,15))
plt.rc('axes', titlesize=32)
plt.rc('axes', labelsize=28)
plt.rc('xtick', labelsize=22)
plt.rc('ytick', labelsize=22)
plt.bar(v2008_state_summary.index, v2008_state_summary.avg_votes)
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.xlabel('States')
plt.ylabel("Avg. Voter Turnout")
plt.title(title)
plt.xlim(-0.75, len(v2008_state_summary.index)-.25)
plt.ylim(0, v2008_state_summary.avg_votes.max()+10)
plt.tight_layout()
plt.savefig(pfile)
plt.show()
```

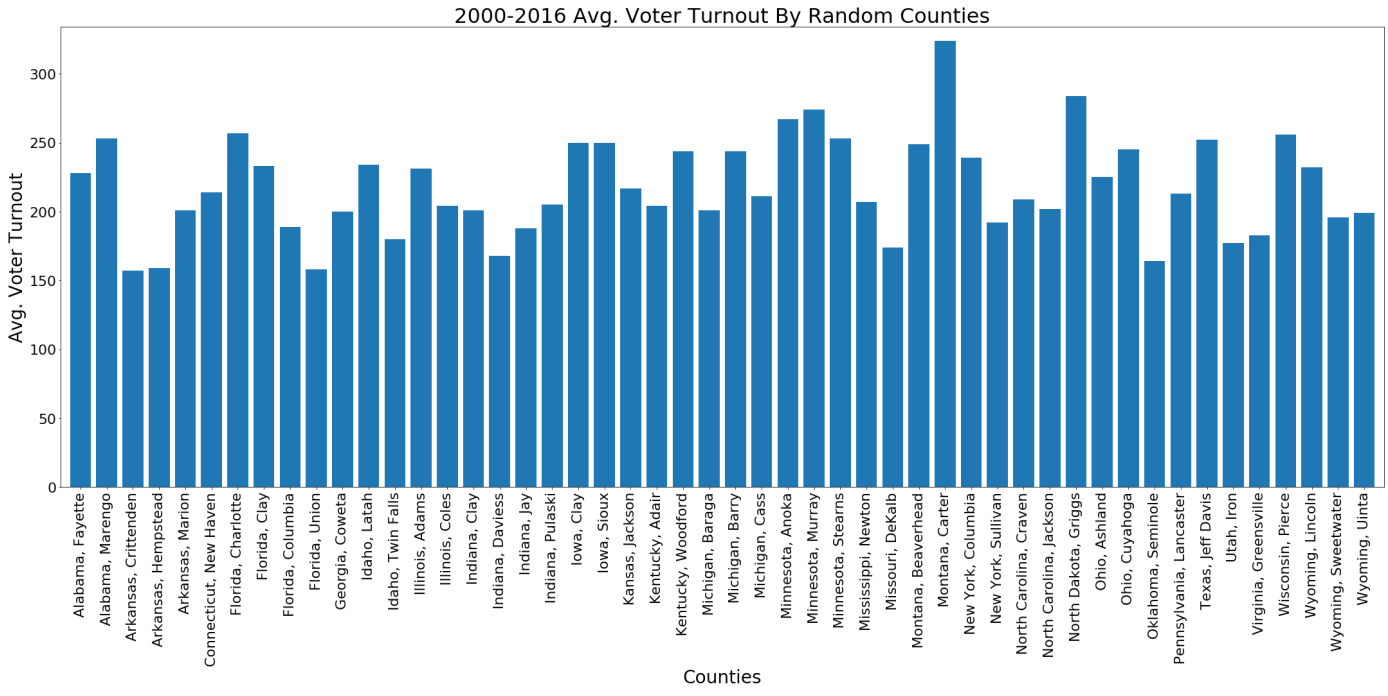


In [11]:
```
# Avg. Party votes by state...
#
plt.figure(figsize=(30,15))
plt.rc('axes', titlesize=32)
plt.rc('axes', labelsize=28)
plt.rc('xtick', labelsize=22)
plt.rc('ytick', labelsize=22)
plt.rc('legend', fontsize=22)
title = year+' Party Votes By State'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
maxy = v2008_state_summary['2008_democrat_votes'].max()
if maxy < v2008_state_summary['2008_republican_votes'].max():
    maxy = v2008_state_summary['2008_republican_votes'].max()
Democrat = plt.plot(v2008_state_summary.index, v2008_state_summary["2008_democrat_vo
Republican = plt.plot(v2008_state_summary.index, v2008_state_summary["2008_republica
Others = plt.plot(v2008_state_summary.index, v2008_state_summary["2008_other_votes"]
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.xlabel('States')
plt.ylabel("Avg. Voter Turnout")
plt.title(title)
plt.xlim(-0.75, len(v2008_state_summary.index)-.25)
plt.ylim(0, maxy+5)
plt.legend(loc="best")
plt.tight_layout()
plt.savefig(pfile)
plt.show()
```

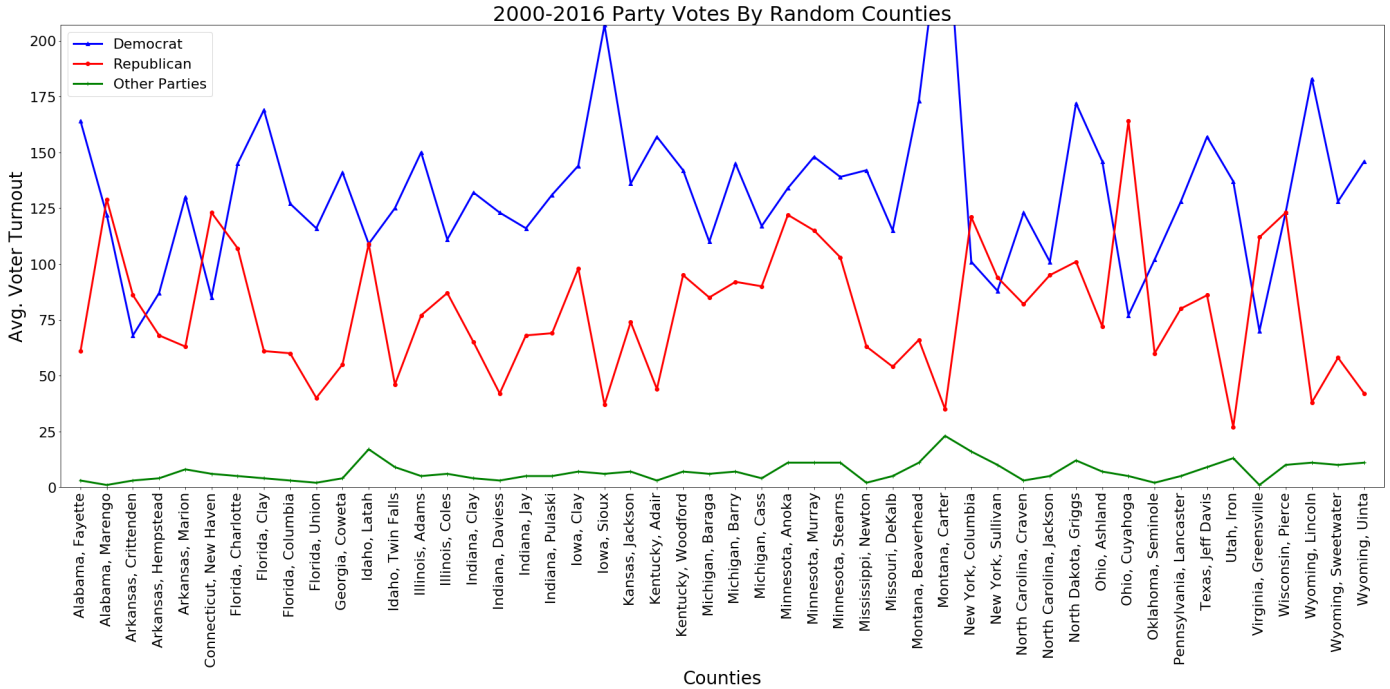


In [12]:
```
# Generating Random 50 counties...
random_sample_county = pd.DataFrame(v2008_county_summary.Location.sample(n=50, randc
random_sample_county.head()
v2008_filtered_county_summary = pd.merge(v2008_county_summary,random_sample_county,
# v2008_filtered_county_summary.head()
```

In [13]:
```
# Avg. Voter turnout by random counties...
#
title = year+' Avg. Voter Turnout By Random Counties'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
plt.figure(figsize=(30,15))
plt.rc('axes', titlesize=32)
plt.rc('axes', labelsize=28)
plt.rc('xtick', labelsize=22)
plt.rc('ytick', labelsize=22)
plt.bar(v2008_filtered_county_summary.Location, v2008_filtered_county_summary.avg_vc
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.xlabel('Counties')
plt.ylabel("Avg. Voter Turnout")
plt.title(title)
plt.xlim(-0.75, len(v2008_filtered_county_summary.Location)-.25)
plt.ylim(0, v2008_filtered_county_summary.avg_votes.max()+10)
plt.tight_layout()
plt.savefig(pfile)
plt.show()
```



In [14]:
```
# Party votes by random counties...
#
plt.figure(figsize=(30,15))
title = year+' Party Votes By Random Counties'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
plt.rc('axes', titlesize=32)
plt.rc('axes', labelsize=28)
plt.rc('xtick', labelsize=22)
plt.rc('ytick', labelsize=22)
plt.rc('legend', fontsize=22)
maxy = v2008_state_summary['2008_democrat_votes'].max()
if maxy < v2008_state_summary['2008_republican_votes'].max():
    maxy = v2008_state_summary['2008_republican_votes'].max()
Democrat = plt.plot(v2008_filtered_county_summary.Location, v2008_filtered_county_su
Republican = plt.plot(v2008_filtered_county_summary.Location, v2008_filtered_county_
Others = plt.plot(v2008_filtered_county_summary.Location, v2008_filtered_county_summ
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.xlabel('Counties')
plt.ylabel("Avg. Voter Turnout")
plt.title(title)
plt.xlim(-0.75, len(v2008_filtered_county_summary.Location)-.25)
plt.ylim(0, maxy+5)
plt.legend(loc="best")
plt.tight_layout()
plt.savefig(pfile)
plt.show()
```



In [15]:
```
# Highest/Lowest voter turnout...
#

print(year + " Highest Voter Turnout for the State  : " + ','.join(hv_st))
print(year + " Lowest Voter Turnout for the State   : " + ','.join(lv_st))
print("")
print(year + " Highest Voter Turnout for the County : " + ','.join(hv_cy))
print(year + " Lowest Voter Turnout for the County  : " + ','.join(lv_cy))
```

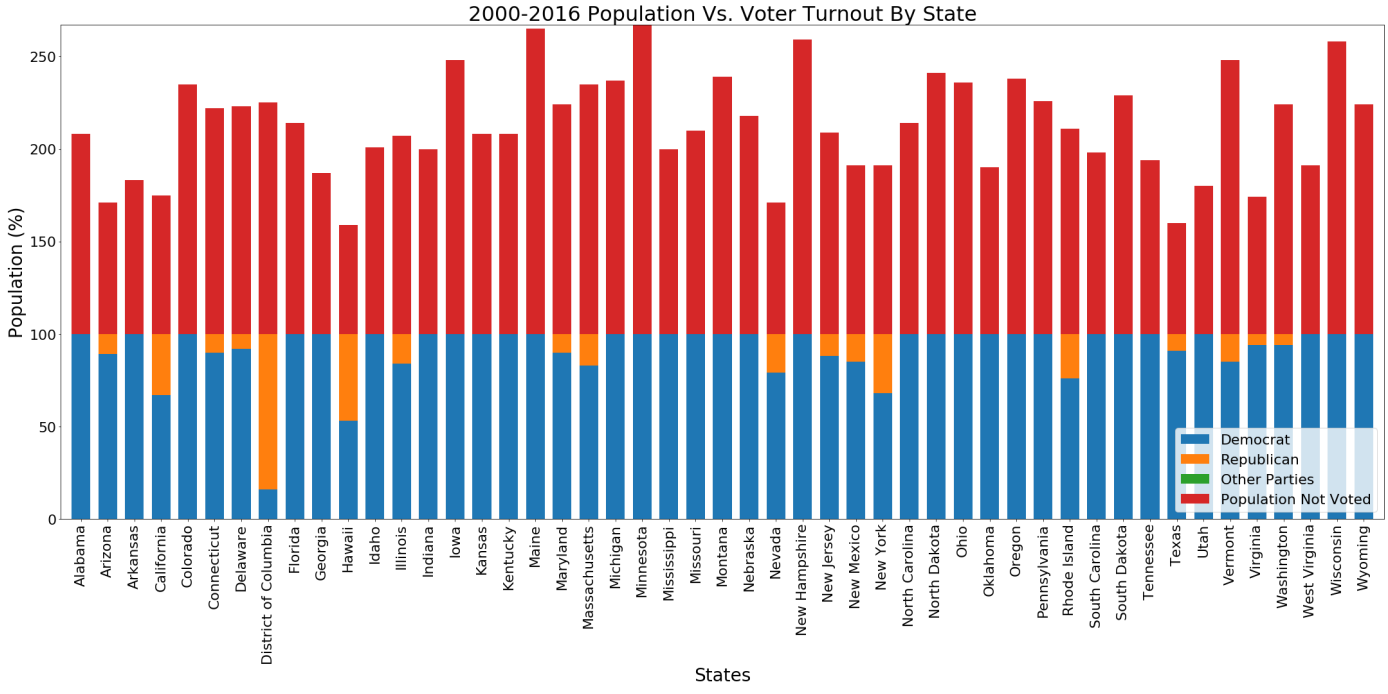
```
2000-2016 Highest Voter Turnout for the State  : Minnesota
2000-2016 Lowest Voter Turnout for the State   : Hawaii

2000-2016 Highest Voter Turnout for the County : Texas, Loving
2000-2016 Lowest Voter Turnout for the County  : Georgia, Chattahoochee
```

In [16]:
```
population_votes = v2008_state_summary[['2008_democrat_votes','2008_republican_votes
population_votes.columns = ['2008_democrat_votes', '2008_republican_votes','2008_oth
population_votes['pop_notvoted'] = (100-(population_votes['2008_democrat_votes']+pop
population_votes.head()
```

Out[16]:

State	2008_democrat_votes	2008_republican_votes	2008_other_votes	pop_notvoted
Alabama	126.0	79.0	3.0	-108.0
Arizona	89.0	76.0	6.0	-71.0
Arkansas	105.0	73.0	5.0	-83.0
California	67.0	102.0	6.0	-75.0
Colorado	110.0	115.0	10.0	-135.0

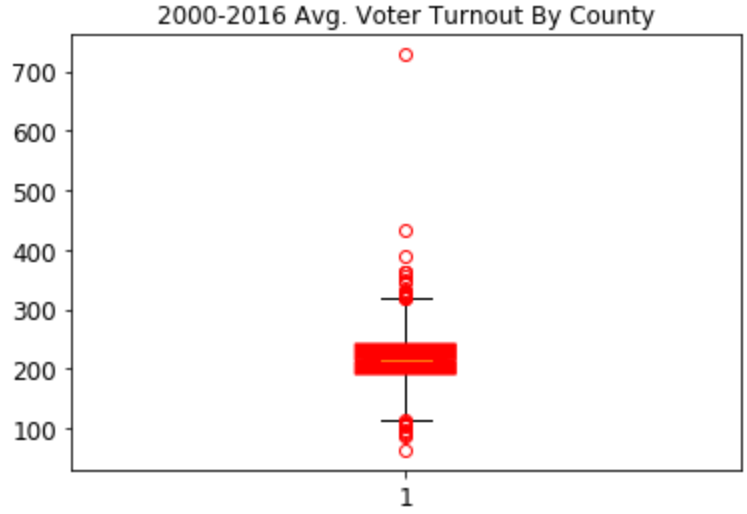
In [17]:
```
# Poplulation vs.Voter turnout by state for the year 2008...
#
title = year+' Population Vs. Voter Turnout By State'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
plt.figure(figsize=(30,15))
bottom = 0
plt.rc('axes', titlesize=32)
plt.rc('axes', labelsize=28)
plt.rc('xtick', labelsize=22)
plt.rc('ytick', labelsize=22)
label = ['Democrat','Republican','Other Parties','Population Not Voted']
#
for i in population_votes.columns:
    plt.bar(population_votes.index, population_votes[i],  width=0.7, bottom=bottom)
    bottom += population_votes[i]
#
plt.xticks(rotation=90)
plt.yticks(rotation=360)
plt.legend(label)
plt.xlabel('States')
plt.ylabel("Population (%)")
plt.title(title)
plt.xlim(-0.75, len(population_votes.index)-.25)
plt.tight_layout()
plt.savefig(pfile)
plt.show()
```



In [18]:
```
# Outliers
#
quartiles = v2008_county_summary['avg_votes'].quantile([.25,.5,.75])
lowerq = quartiles[0.25]
upperq = quartiles[0.75]
iqr = upperq-lowerq
lower_bound = lowerq - (1.5*iqr)
upper_bound = upperq + (1.5*iqr)

print(f"The lower quartile of county outliers  is :{lowerq}")
print(f"The upper quartile of county outliers  is :{upperq}")
print(f"The interquartile range of county outliers  is :{iqr}")
print(f"The median of county volume  is :{quartiles[0.5]}")
print(f"Values below {round(lower_bound,6)} could be outliers. ")
print(f"Values above {round(upper_bound,6)} could be outliers. ")
print(" ")
```

```
The lower quartile of county outliers  is :191.0
The upper quartile of county outliers  is :242.0
The interquartile range of county outliers  is :51.0
The median of county volume  is :214.0
Values below 114.5 could be outliers.
Values above 318.5 could be outliers.
```

In [19]:
```
title = year+' Avg. Voter Turnout By County'
pfile = title.replace(" ",'_')+'.png'
pfile = f"Resources/Images/{pfile}"
boxplot_data = v2008_county_summary['avg_votes']
fig = plt.figure()
plt.rc('axes', titlesize=12)
plt.rc('axes', labelsize=12)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)
plt.rc('legend', fontsize=12)
c = "red"
plt.boxplot(boxplot_data,  notch=True, patch_artist=True,
            boxprops=dict(facecolor=c, color=c), flierprops=dict(color=c, markeredge

plt.title(title)
plt.savefig(pfile)
plt.show()
```



In [20]:
```
pop_county = v2008_county_summary['avg_votes']
sample_county = pd.DataFrame(v2008_county_summary.avg_votes.sample(n=1000, random_st

stats.ttest_1samp(sample_county, pop_county.mean())
```

Out[20]:
```
Ttest_1sampResult(statistic=array([1.45453249]), pvalue=array([0.14611284]))
```

In [ ]:
```

```

In [ ]:
```

```