Analyzing Unemployment Data King County, WA

By: Gurpal Singh
Date: October 7,2019

Description:

This project uses two datasets, one containing employment data by county and one containing population by state to analyze unemployment trends and distributions. The datasets were obtained from government websites so they can be deemed reputable (supposedly ;)). See the links below if you would like to obtain the dataset for yourself.

- Employment Data: United States Department of Agriculture Economic Research Service (https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/)
- Population Data: United States Census Bureau (https://www.census.gov/data/tables/time-series/demo/popest/2010s-national-total.html)

Importing Libraries:

- pandas for dataframes to make data manipulation easy
- matplotlib for data visualization
- numpy for numerical operations on arrays

In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

Reading the datafile '*Unemployment_Data.xls*'

The excel file is read and saved into data frame '*df*'. Next we look at the shape and contents of the dataframe. Unnecessary data is dropped from the dataframe.

In [2]:
```
df = pd.read_excel(r'Unemployment_Data.xls')
df.head(10)
```

Out[2]:

	Unemployment and median household income for the U.S., States, and counties, 2007-18	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnam
0	NaN	NaN	NaN	NaN	
1	Sources: Unemployment - Bureau of Labor Statis...	NaN	NaN	NaN	
2	Median Household Income - Census Bureau - SAIP...	NaN	NaN	NaN	
3	For definitions of rural classifications, see ...	NaN	NaN	NaN	
4	This table was prepared by USDA, Economic Rese...	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	
6	FIPS	State	Area_name	Rural_urban_continuum_code_2013	Urban_influence_code_
7	0	US	United States	NaN	
8	1000	AL	Alabama	NaN	
9	1001	AL	Autauga County, AL	2	

10 rows × 56 columns

In [3]:
```
# Drop Junk Rows
df.drop(df.index[[0,1,2,3,4,5]],inplace = True)
df.reset_index(inplace=True)
```

Assigning Lablels to the features of the data

Notice the row with index zero contains label names we can use. So let's utilize them.

In [4]:
```
df.columns = df.iloc[0]
df.reset_index(inplace=True)
df.head(5)
```

Out[4]:

	index	6	FIPS	State	Area_name	Rural_urban_continuum_code_2013	Urban_influence_code_
0	0	6	FIPS	State	Area_name	Rural_urban_continuum_code_2013	Urban_influence_code_
1	1	7	0	US	United States	NaN	
2	2	8	1000	AL	Alabama	NaN	
3	3	9	1001	AL	Autauga County, AL	2	
4	4	10	1003	AL	Baldwin County, AL	3	

5 rows × 58 columns

Let's drop row 0 because it contains column names and we have already extracted those to our dataframe

In [5]:
```
df.drop([0],inplace=True)
df.reset_index(inplace=True)
df.head(5)
```

Out[5]:

	level_0	index	6	FIPS	State	Area_name	Rural_urban_continuum_code_2013	Urban_influen
0	1	1	7	0	US	United States	NaN	
1	2	2	8	1000	AL	Alabama	NaN	
2	3	3	9	1001	AL	Autauga County, AL	2	
3	4	4	10	1003	AL	Baldwin County, AL	3	
4	5	5	11	1005	AL	Barbour County, AL	6	

5 rows × 59 columns

Let's also delete columns for Rural_urban_continuum_code_2013 and Urban_influence_code_2013 since these seem useless.

In [6]:
```
df.drop([6,"FIPS","index","Rural_urban_continuum_code_2013","Urban_influence_code_2013","Metro_2013"], axis = 1,inplace=True)
df.reset_index(inplace=True)
df.head(5)
```

Out[6]:

	index	level_0	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_200
0	0	1	US	United States	152191093	145156134	703495
1	1	2	AL	Alabama	2175612	2089127	8648
2	2	3	AL	Autauga County, AL	24383	23577	80
3	3	4	AL	Baldwin County, AL	82659	80099	256
4	4	5	AL	Barbour County, AL	10334	9684	65

5 rows × 54 columns

Now our Data is clean and we can proceed!

First let's see how Unemployment has changed over the years in the United States as a whole. Since there other columns by year, we will save unemployment data to a data frame named '*Unemployed*.'
We will also plot population to see how it has changed during those years.
Note: Population Data only has data for years 2010 - 2018 and code for population is commented so we can analyze trend in Unemployment (scale)

In [7]:
```
Unemployed = df[['Unemployed_2007','Unemployed_2008','Unemployed_2009','Unemployed_2010','Unemployed_2011','Unemployed_2012',
         'Unemployed_2013','Unemployed_2014','Unemployed_2015','Unemployed_2016','Unemployed_2017','Unemployed_2018']]
Unemployed.head()
```

Out[7]:

	Unemployed_2007	Unemployed_2008	Unemployed_2009	Unemployed_2010	Unemployed_2011
0	7034959	8900745	14230757	14862528	13840507
1	86485	123012	238252	231483	212257
2	806	1267	2402	2282	2159
3	2560	3851	8048	8339	7627
4	650	894	1431	1262	1137

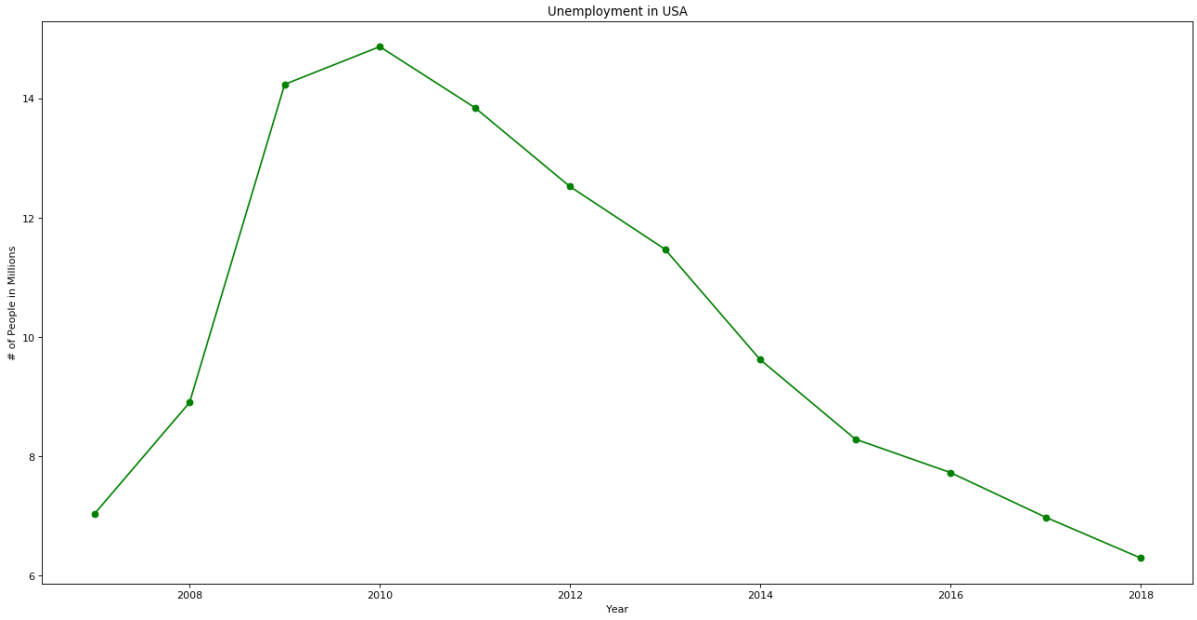
Loading the Population Data

In [8]:
```
pop = pd.read_excel(r'Population_USA.xlsx')
pop.head(10)
```

Out[8]:

	table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnan
0	Table 1. Annual Estimates of the Resident Popu...	NaN	NaN	NaN	NaN	NaN	NaN	
1	Geographic Area	2010-04-01 00:00:00	NaN	Population Estimate (as of July 1)	NaN	NaN	NaN	
2	NaN	Census	Estimates Base	2010	2011.0	2012.0	2013.0	2
3	United States	308745538	308758105	309326085	311580009.0	313874218.0	316057727.0	31838
4	Northeast	55317240	55318430	55380645	55600532.0	55776729.0	55907823.0	5601
5	Midwest	66927001	66929743	66974749	67152631.0	67336937.0	67564135.0	6775
6	South	114555744	114563045	114867066	116039399.0	117271075.0	118393244.0	11965
7	West	71945553	71946887	72103625	72787447.0	73489477.0	74192525.0	7496
8	.Alabama	4779736	4780138	4785448	4798834.0	4815564.0	4830460.0	484
9	.Alaska	710231	710249	713906	722038.0	730399.0	737045.0	73

In [9]:
```
# Making a List for years and extracting data from dataframe and scaling down
 by 1 million
year = list(range(2007, 2019))
Unemployment_US = list(Unemployed.iloc[0] / 1000000)

# Creating a line graph
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.plot(year, Unemployment_US, color='green', marker='o', linestyle ='solid')

# Add Title and Axis Labels
plt.title("Unemployment in USA")
plt.xlabel("Year")
plt.ylabel("# of People in Millions")

plt.show()
```



Insight:

It is clear that unemployment hit a peak in 2010. After doing some research, this is the direct effect of the recession that hit the united states in 2007-2008 due to the housing crisis. We can see from the plot that unemployment sky-rocketed from the years with 2008 to 2009 having the steepest slope. It also appears the decrease in unemployment is leveling out but only future data will confirm this.

Now let's examine state data and see how it is distributed within the USA.

- Notice, the dataset we have for unemployment has data for all counties. We will need to extract data only for states. First we will need to get some state names to use as search criteria. Using the link below, an csv can be downloaded with State Names.

In [10]:
```
# Building a list with just state names from State_Data.csv
import csv

State_Names = []
csv.register_dialect('myDialect',
                     delimiter = ',',
                     quoting=csv.QUOTE_ALL,
                     skipinitialspace=True)

with open('State_Data.csv', 'r') as f:
    reader = csv.reader(f, dialect='myDialect')
    for row in reader:
        State_Names.append(row[0])

# We need to remove the first item from the list (label 'State')
State_Names.pop(0)

len(State_Names) # Note: there are 51 elements because data includes District
 of Columbia
```

Out[10]:
```
51
```

Extracting State Data from the Unemployment Dataset '*df*'.

In [11]:
```
# Extract State rows to new DataFrame called States
states = df[df['Area_name'].isin(State_Names)]

# Drop Junk Columns
states.drop(columns=['index','level_0'],inplace=True)

# Reset the index of the dataframe
states.reset_index(inplace=True)

# Print to check
states
```

```
C:\Users\Gurpal\Anaconda3\lib\site-packages\pandas\core\frame.py:3940: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy
  errors=errors)
```

Out[11]:

	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_2007	Ur
0	1	AL	Alabama	2175612	2089127	86485	
1	69	AK	Alaska	350785	328579	22206	
2	102	AZ	Arizona	3034016	2917117	116899	
3	118	AR	Arkansas	1369284	1296572	72712	
4	194	CA	California	17893080	16931590	961490	
5	253	Co	Colorado	2664677	2565218	99459	
6	318	CT	Connecticut	1856209	1773159	83050	
7	327	DE	Delaware	443573	428312	15261	
8	331	DC	District of Columbia	322237	304426	17811	
9	332	DC	District of Columbia	322237	304426	17811	
10	333	FL	Florida	9157124	8789770	367354	
11	401	GA	Georgia	4815818	4597640	218178	
12	561	HI	Hawaii	638395	620535	17860	
13	566	ID	Idaho	754438	731235	23203	
14	611	IL	Illinois	6665601	6334010	331591	
15	714	IN	Indiana	3207687	3061042	146645	
16	807	IA	Iowa	1660677	1599332	61345	
17	907	KS	Kansas	1483458	1420449	63009	
18	1013	KY	Kentucky	2032082	1922220	109862	
19	1134	LA	Louisiana	2030434	1944038	86396	
20	1199	ME	Maine	700468	667781	32687	
21	1216	MD	Maryland	2970094	2867348	102746	
22	1241	MA	Massachusetts	3426009	3268096	157913	
23	1256	MI	Michigan	5011120	4658939	352181	
24	1340	MN	Minnesota	2906390	2773704	132686	
25	1428	MS	Mississippi	1303514	1224059	79455	
26	1511	MO	Missouri	3034579	2879647	154932	
27	1627	MT	Montana	502070	484189	17881	
28	1684	NE	Nebraska	978763	949494	29269	
29	1778	NV	Nevada	1330396	1270572	59824	
30	1796	NH	New Hampshire	737942	712008	25934	
31	1807	NJ	New Jersey	4441797	4251815	189982	
32	1829	NM	New Mexico	934027	898998	35029	
33	1863	NY	New York	9522056	9088207	433849	

	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_2007	Ur
34	1926	NC	North Carolina	4512856	4300304	212552	
35	2027	ND	North Dakota	367234	355766	11468	
36	2081	OH	Ohio	5990292	5657718	332574	
37	2170	OK	Oklahoma	1726259	1655490	70769	
38	2248	OR	Oregon	1921766	1822772	98994	
39	2285	PA	Pennsylvania	6342997	6064063	278934	
40	2353	RI	Rhode Island	573173	543401	29772	
41	2359	SC	South Carolina	2125891	2005686	120205	
42	2406	SD	South Dakota	442499	430011	12488	
43	2473	TN	Tennessee	3063669	2920352	143317	
44	2569	TX	Texas	11431631	10941413	490218	
45	2824	UT	Utah	1359129	1324060	35069	
46	2854	VT	Vermont	353739	339547	14192	
47	2869	VA	Virginia	4036835	3914087	122748	
48	3003	WA	Washington	3403163	3243308	159855	
49	3043	WV	West Virginia	811160	773990	37170	
50	3099	WI	Wisconsin	3087828	2936452	151376	
51	3172	WY	Wyoming	286560	278486	8074	

52 rows × 53 columns

In [12]:
```
# Note there is a duplicate for District of Columbia so we will drop by index
states.drop([8], inplace = True)
states.reset_index(inplace=True)
states
```

Out[12]:

	level_0	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_
0	0	1	AL	Alabama	2175612	2089127	8
1	1	69	AK	Alaska	350785	328579	2
2	2	102	AZ	Arizona	3034016	2917117	11
3	3	118	AR	Arkansas	1369284	1296572	7
4	4	194	CA	California	17893080	16931590	96
5	5	253	Co	Colorado	2664677	2565218	9
6	6	318	CT	Connecticut	1856209	1773159	8
7	7	327	DE	Delaware	443573	428312	1
8	9	332	DC	District of Columbia	322237	304426	
9	10	333	FL	Florida	9157124	8789770	36
10	11	401	GA	Georgia	4815818	4597640	21
11	12	561	HI	Hawaii	638395	620535	1
12	13	566	ID	Idaho	754438	731235	2
13	14	611	IL	Illinois	6665601	6334010	33
14	15	714	IN	Indiana	3207687	3061042	14
15	16	807	IA	Iowa	1660677	1599332	6
16	17	907	KS	Kansas	1483458	1420449	6
17	18	1013	KY	Kentucky	2032082	1922220	10
18	19	1134	LA	Louisiana	2030434	1944038	8
19	20	1199	ME	Maine	700468	667781	3
20	21	1216	MD	Maryland	2970094	2867348	10
21	22	1241	MA	Massachusetts	3426009	3268096	15
22	23	1256	MI	Michigan	5011120	4658939	35
23	24	1340	MN	Minnesota	2906390	2773704	13
24	25	1428	MS	Mississippi	1303514	1224059	7
25	26	1511	MO	Missouri	3034579	2879647	15
26	27	1627	MT	Montana	502070	484189	1
27	28	1684	NE	Nebraska	978763	949494	2
28	29	1778	NV	Nevada	1330396	1270572	5
29	30	1796	NH	New Hampshire	737942	712008	2
30	31	1807	NJ	New Jersey	4441797	4251815	18
31	32	1829	NM	New Mexico	934027	898998	3
32	33	1863	NY	New York	9522056	9088207	43
33	34	1926	NC	North Carolina	4512856	4300304	21

	level_0	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_
34	35	2027	ND	North Dakota	367234	355766	1
35	36	2081	OH	Ohio	5990292	5657718	33
36	37	2170	OK	Oklahoma	1726259	1655490	7
37	38	2248	OR	Oregon	1921766	1822772	9
38	39	2285	PA	Pennsylvania	6342997	6064063	27
39	40	2353	RI	Rhode Island	573173	543401	2
40	41	2359	SC	South Carolina	2125891	2005686	12
41	42	2406	SD	South Dakota	442499	430011	1
42	43	2473	TN	Tennessee	3063669	2920352	14
43	44	2569	TX	Texas	11431631	10941413	49
44	45	2824	UT	Utah	1359129	1324060	3
45	46	2854	VT	Vermont	353739	339547	1
46	47	2869	VA	Virginia	4036835	3914087	12
47	48	3003	WA	Washington	3403163	3243308	15
48	49	3043	WV	West Virginia	811160	773990	3
49	50	3099	WI	Wisconsin	3087828	2936452	15
50	51	3172	WY	Wyoming	286560	278486	

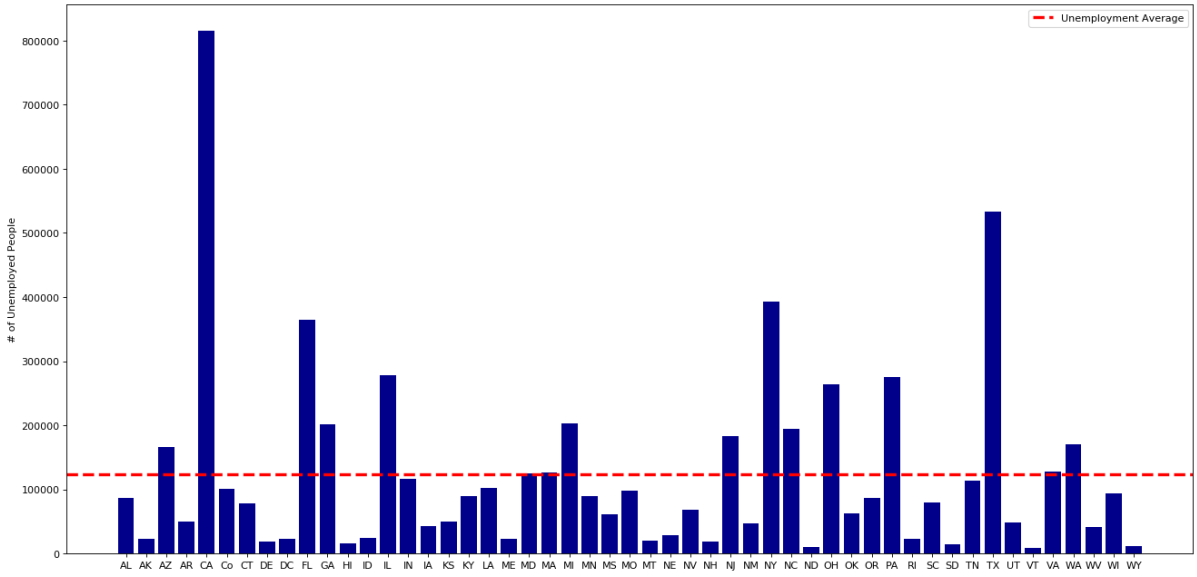
51 rows × 54 columns

Now the State Data is Clean and we can proceed to the plots

We will look at Unemployment data for the year of 2018 as it is the most recent year for which data exists. A histogram will be a good choice to look the distribution among states and it will be plotted in descending order.

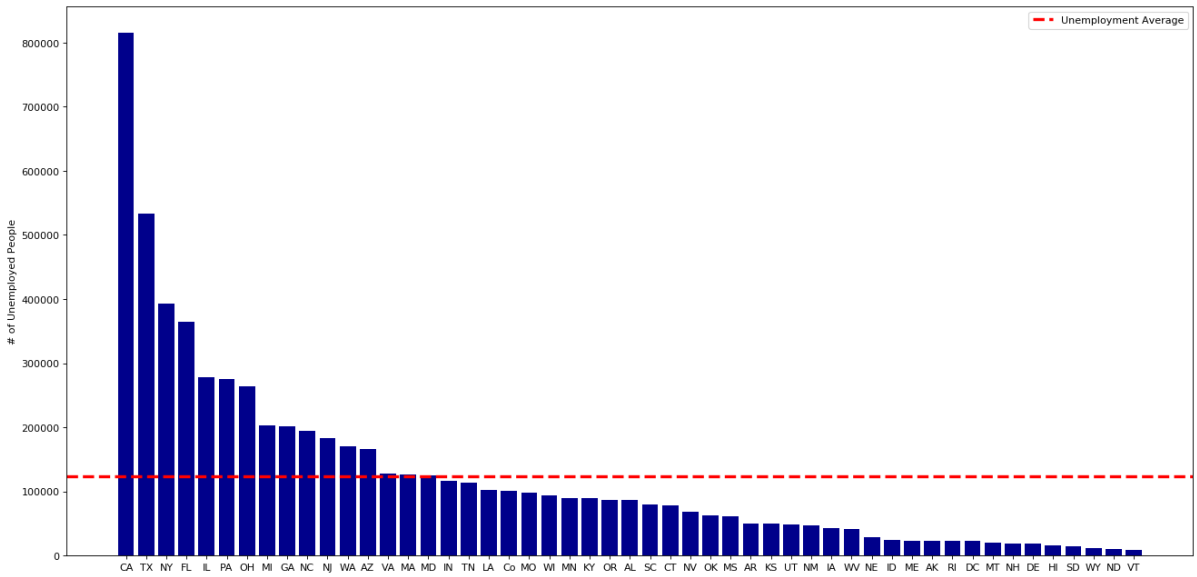
In [13]:
```
# Saving 2018 Unemployment State data to a list
Unemployment_2018 = states["Unemployed_2018"].tolist()

# We will use state abbreviations for plot for presentation
State_abr = states["State"].tolist()

# Check for match before plotting
print(len(Unemployment_2018))
print(len(State_Names))

# Let's also compute average for reference and plot it
avg_unemploy = np.mean(np.asarray(Unemployment_2018))
print("Average Unemployment 2018 = " + str(avg_unemploy))

# Plotting bar graph and horizontal line for average
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.axhline(y=avg_unemploy, linestyle = '--', color='red', linewidth = 3)
plt.legend(["Unemployment Average"])
plt.bar(State_abr, Unemployment_2018,color='darkblue')
plt.ylabel("# of Unemployed People")
plt.show()
```

```
51
51
Average Unemployment 2018 = 123456.7843137255
```



Lets make this easier to read and put in descending order.

In [14]:
```
# Sort states by Unemployment to make our graph look cleaner

# Zipping to keep pairs and casting to list, sort, and unzip
Unemp_zip = list(zip(Unemployment_2018,State_abr))
Unemp_zip.sort(reverse = True)
Unemp_unzip = list(zip(*Unemp_zip))

# Plotting
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.axhline(y=avg_unemploy, linestyle = '--', color='red', linewidth = 3)
plt.legend(["Unemployment Average"])
plt.bar(Unemp_unzip[1][:],Unemp_unzip[0][:],color='darkblue')
plt.ylabel("# of Unemployed People")
plt.show()
```



Insight:

The plot above doesn't provide much information other than numbers for Unemployment and which states are leading. For a more indicative measure let's form a new measure Unemployment/StatePopulation. This will be better basis for comparison as it regularizes the data.

Population Data

We loaded the population data earlier in a dataframe named '*pop*'. Now it is time to clean it.

In [15]:
```
pop.head(5)
```

Out[15]:

	table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnan
0	Table 1. Annual Estimates of the Resident Popu...	NaN	NaN	NaN	NaN	NaN	NaN	
1	Geographic Area	2010-04-01 00:00:00	NaN	Population Estimate (as of July 1)	NaN	NaN	NaN	
2	NaN	Census	Estimates Base	2010	2011.0	2012.0	2013.0	2
3	United States	308745538	308758105	309326085	311580009.0	313874218.0	316057727.0	31838
4	Northeast	55317240	55318430	55380645	55600532.0	55776729.0	55907823.0	5601

In [16]:
```
# Drop irrelevanat rows
pop.drop(pop.index[[0,1,2,3,4,5,6,7]], inplace = True)
pop.reset_index(inplace = True)
```

In [17]:
```
pop.head(5)
```

Out[17]:

	index	table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unn
0	8	.Alabama	4779736	4780138	4785448	4798834.0	4815564.0	4830460.0	484
1	9	.Alaska	710231	710249	713906	722038.0	730399.0	737045.0	73
2	10	.Arizona	6392017	6392288	6407774	6473497.0	6556629.0	6634999.0	673
3	11	.Arkansas	2915918	2916028	2921978	2940407.0	2952109.0	2959549.0	296
4	12	.California	37253956	37254523	37320903	37641823.0	37960782.0	38280824.0	3862

In [18]:
```
# Dropping the unnessary 'Index Column'
pop.drop(columns="index", inplace = True)
pop.head(5)
```

Out[18]:

	table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	.Alabama	4779736	4780138	4785448	4798834.0	4815564.0	4830460.0	4842481.0
1	.Alaska	710231	710249	713906	722038.0	730399.0	737045.0	736307.0
2	.Arizona	6392017	6392288	6407774	6473497.0	6556629.0	6634999.0	6733840.0
3	.Arkansas	2915918	2916028	2921978	2940407.0	2952109.0	2959549.0	2967726.0
4	.California	37253956	37254523	37320903	37641823.0	37960782.0	38280824.0	38625139.0

In [19]:
```
print(pop.shape)
pop.tail(5)
```

```
(58, 12)
```

Out[19]:

	table with row headers in column A and column headers in rows 3 through 4. (leading dots indicate sub-parts)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	U
53	Note: The estimates are based on the 2010 Cens...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
54	Suggested Citation:	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
55	Table 1. Annual Estimates of the Resident Popu...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
56	Source: U.S. Census Bureau, Population Division	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
57	Release Date: December 2018	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [20]:
```
# Dropping uncessary columns on tail end and giving our label names
pop.drop(pop.index[[51,52,53,54,55,56,57]], inplace = True)
pop.columns = ['State','Census','Estimate Base','2010','2011','2012','2013','2014','2015','2016','2017','2018']
pop.head(5)
```

Out[20]:

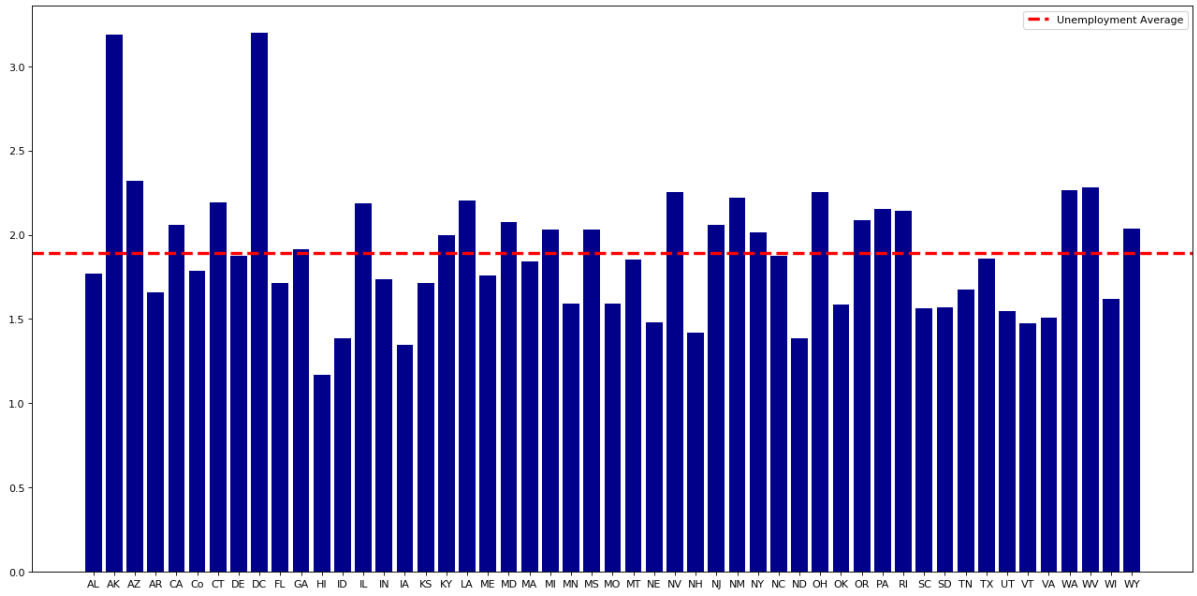
	State	Census	Estimate Base	2010	2011	2012	2013	2014	
0	.Alabama	4779736	4780138	4785448	4798834.0	4815564.0	4830460.0	4842481.0	4
1	.Alaska	710231	710249	713906	722038.0	730399.0	737045.0	736307.0	
2	.Arizona	6392017	6392288	6407774	6473497.0	6556629.0	6634999.0	6733840.0	6
3	.Arkansas	2915918	2916028	2921978	2940407.0	2952109.0	2959549.0	2967726.0	2
4	.California	37253956	37254523	37320903	37641823.0	37960782.0	38280824.0	38625139.0	38

The Population dataframe is now clean and ready to use.

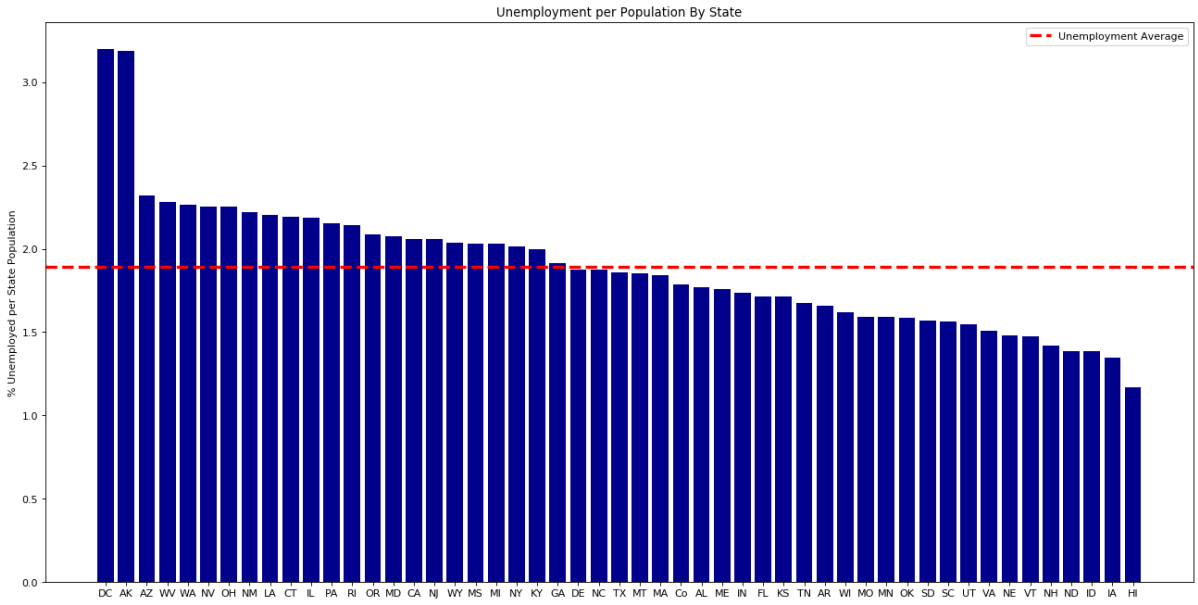
Plotting the Unemployent Distribution by State

In [21]:
```
states.loc[:,'Unemployment2018_over_Pop'] = (states['Unemployed_2018'] / pop['2018'] * 100)
avg_2018_Unemployment2018_over_Pop = states.loc[:,"Unemployment2018_over_Pop"].mean()

# No creating another plot for Unemployment per Population
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.axhline(y=avg_2018_Unemployment2018_over_Pop, linestyle = '--', color='red', linewidth = 3)
plt.legend(["Unemployment Average"])
plt.bar(State_abr, states.loc[:,'Unemployment2018_over_Pop'],color='darkblue')
plt.show()
```

```
C:\Users\Gurpal\Anaconda3\lib\site-packages\pandas\core\indexing.py:362: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  self.obj[key] = _infer_fill_value(value)
C:\Users\Gurpal\Anaconda3\lib\site-packages\pandas\core\indexing.py:543: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy
  self.obj[item] = s
```



In [22]:
```
# Saving Unemployment/Population Column to List
Unemp_per_pop = list(states['Unemployment2018_over_Pop'])

# Zipping the list to maintain pairs
ziplist = zip(Unemp_per_pop, State_abr)

# Cast to list object
ziplist = list(ziplist)

# Using built in sort to put in descending order
ziplist.sort(reverse = True)

# Unzip & convert to list object
unziplist = zip(*ziplist)
unziplist = list(unziplist)

# Now creating another plot for Unemployment per Population
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.bar(unziplist[1][:],unziplist[0][:], color='darkblue')
plt.title("Unemployment per Population By State")
plt.ylabel("% Unemployed per State Population")
plt.axhline(y=avg_2018_Unemployment2018_over_Pop, linestyle = '--', color='red', linewidth = 3)
plt.legend(["Unemployment Average"])
plt.show()
```



In [23]:
```
# Extract the data for Washington, California, and Idaho
WA_data = states[states['Area_name']=='Washington']
#CA_data = states[states['Area_name']=='California']
#ID_data = states[states['Area_name']=='Idaho']
#Unemployment_US = list(Unemployed.iloc[0] / 100000)

# Extract Unemployment Data for each of the states
WA_Unemployed = WA_data[['Unemployed_2007','Unemployed_2008','Unemployed_2009'
,'Unemployed_2010','Unemployed_2011','Unemployed_2012',
         'Unemployed_2013','Unemployed_2014','Unemployed_2015','Unemployed_201
6','Unemployed_2017','Unemployed_2018']]

#CA_Unemployed = CA_data[['Unemployed_2007','Unemployed_2008','Unemployed_200
9','Unemployed_2010','Unemployed_2011','Unemployed_2012',
#         'Unemployed_2013','Unemployed_2014','Unemployed_2015','Unemployed_20
16','Unemployed_2017','Unemployed_2018']]

#ID_Unemployed = ID_data[['Unemployed_2007','Unemployed_2008','Unemployed_200
9','Unemployed_2010','Unemployed_2011','Unemployed_2012',
#         'Unemployed_2013','Unemployed_2014','Unemployed_2015','Unemployed_20
16','Unemployed_2017','Unemployed_2018']]


# Scale the data by 100000
WA_Unemployed = list(WA_Unemployed.iloc[0]/100000)
#CA_Unemployed = list(CA_Unemployed.iloc[0]/100000)
#ID_Unemployed = list(ID_Unemployed.iloc[0]/100000)

# Need to scale up US data 10 (earlier it was scaled down by 100000)
```

In [24]:
```
# Plotting WA data and US Data for comparison
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.plot(year, WA_Unemployed, color='green', marker='o', linestyle ='solid')
#plt.plot(year, CA_Unemployed, color='red', marker='o', linestyle='solid')
#plt.plot(year, ID_Unemployed, color='purple', marker='o', linestyle='solid')
# plt.plot(year, Unemployment_US, color='blue', marker='o', linestyle ='soli
d')

# Add Title and Axis Labels
plt.title("Unemployment in Washington State")
plt.xlabel("Year")
plt.ylabel("# of People in Hundred Thousands")
#plt.legend(["Washington","California","Idaho","USA"])
plt.show()
```

From the plot we can see Washington State follows a similar trend to the country-wide unemployment. To view Unemployment from the other states simply uncomment the code. They are left out to emphasize the trend in Washington's data.

Washington State

Let's get into the details of Washington State. First we will look at the county wide distribution and then the trends in data with King County specifically.

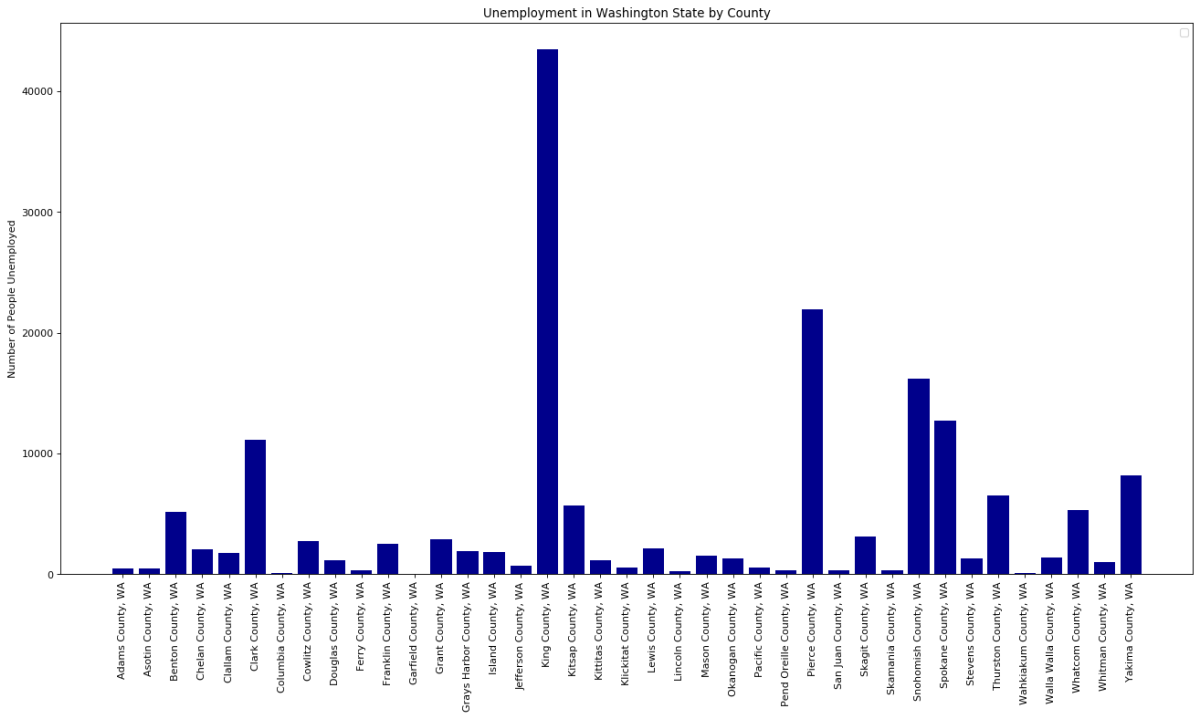
In [25]:
```
# Let's dig into Washington's data
WA_County_Data = df[df['State'].str.match("WA")]
WA_County_Data.drop(columns=["index","level_0"], inplace = True)
WA_County_Data.reset_index(inplace=True)

# We only want counties so remove first row
WA_County_Data.drop(WA_County_Data.index[0],inplace=True)
WA_County_Data.reset_index()
WA_County_Data.head(5)
```

Out[25]:

	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_2007	Unem
1	3004	WA	Adams County, WA	8021	7495	526	
2	3005	WA	Asotin County, WA	10264	9733	531	
3	3006	WA	Benton County, WA	86073	81368	4705	
4	3007	WA	Chelan County, WA	40622	38390	2232	
5	3008	WA	Clallam County, WA	30007	28041	1966	

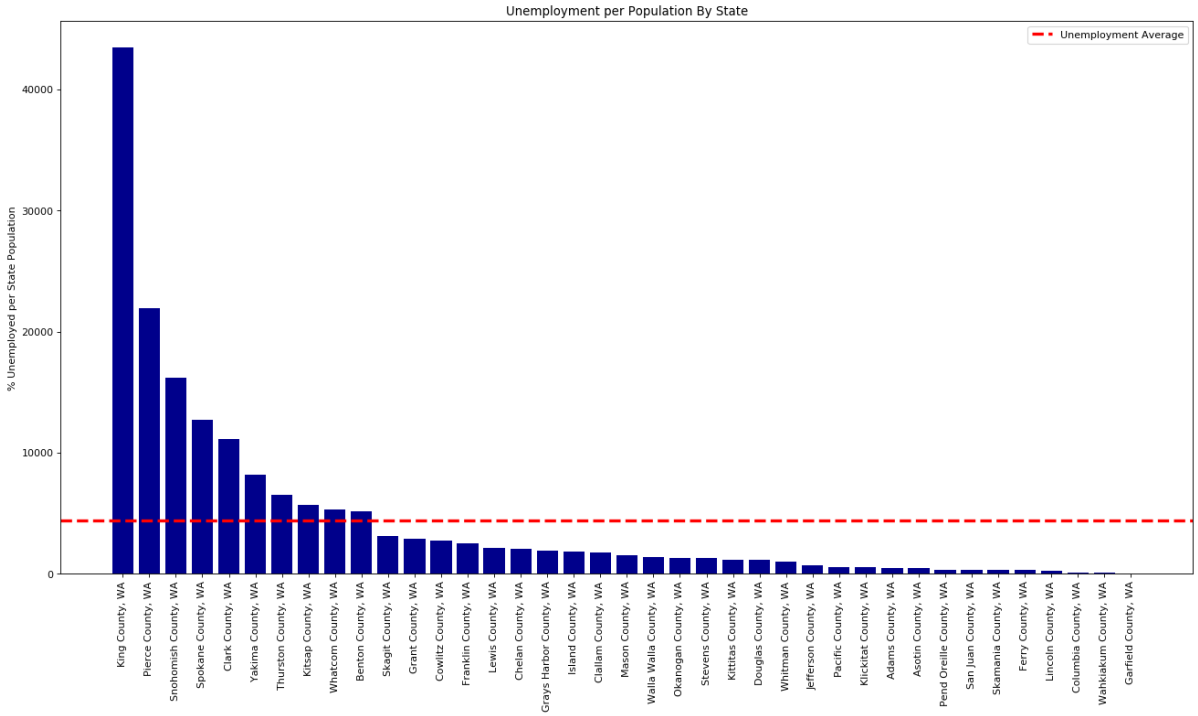
5 rows × 53 columns

In [26]:
```
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.legend(["Unemployment Average"])
plt.bar(WA_County_Data.loc[:,'Area_name'], WA_County_Data.loc[:,'Unemployed_2018'],color='darkblue')
plt.xticks(rotation='vertical')
plt.title('Unemployment in Washington State by County')
plt.ylabel('Number of People Unemployed')
plt.show()
```



Sorting the data for ease of view

In [27]:
```
# Calculating the County Average
Unemp_County_Avg = WA_County_Data.loc[:,"Unemployed_2018"].mean()


# Need to sort the chart to put in descending order
County_Names = WA_County_Data['Area_name'].tolist()
WA_Unemployed_2018 = WA_County_Data['Unemployed_2018'].tolist()

# Zipping the list to maintain pairs
ziplist = zip(WA_Unemployed_2018, County_Names)

# Cast to list object
ziplist = list(ziplist)

# Using built in sort to put in descending order
ziplist.sort(reverse = True)

# Unzip & convert to list object
unziplist = zip(*ziplist)
unziplist = list(unziplist)

# Now creating another plot for Unemployment per Population
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.bar(unziplist[1][:],unziplist[0][:], color='darkblue')
plt.title("Unemployment per Population By State")
plt.ylabel("% Unemployed per State Population")
plt.xticks(rotation='vertical')
plt.axhline(y=Unemp_County_Avg, linestyle = '--', color='red', linewidth = 3)
plt.legend(["Unemployment Average"])
plt.show()
```



King County

Now let's look at data for King County specifcally as this is where I will reside.

In [28]:
```
# Lets organize King County Data into 2 Separate Data frames for easy of plott
ing and calculation
King_data = WA_County_Data[WA_County_Data['Area_name'].str.match('King')]
King_data
```

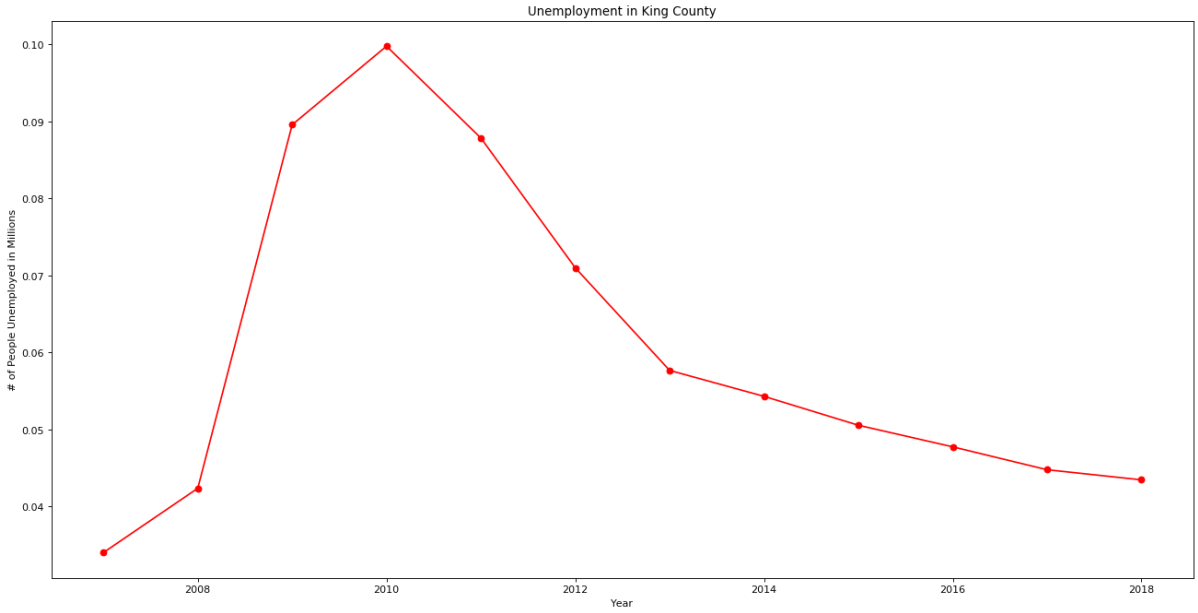
Out[28]:

	index	State	Area_name	Civilian_labor_force_2007	Employed_2007	Unemployed_2007	Uner
17	3020	WA	King County, WA	1067240	1033216	34024	

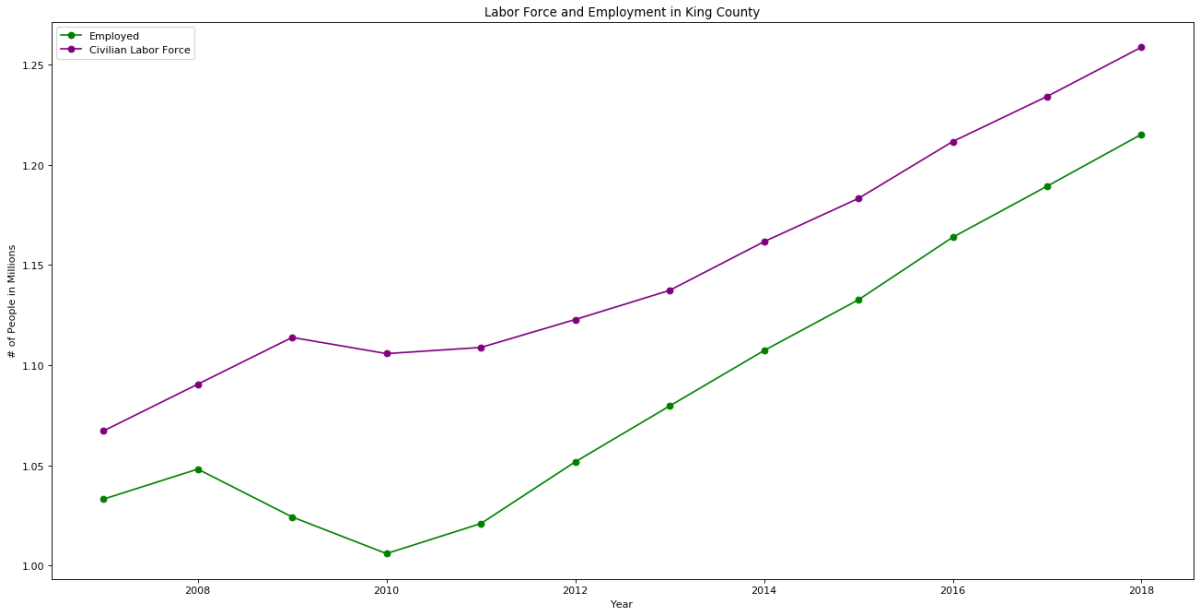
1 rows × 53 columns

In [29]:
```
# Extracting the Data

King_Unemployed = King_data[['Unemployed_2007','Unemployed_2008','Unemployed_2
009','Unemployed_2010','Unemployed_2011','Unemployed_2012',
        'Unemployed_2013','Unemployed_2014','Unemployed_2015','Unemployed_201
6','Unemployed_2017','Unemployed_2018']]

King_employed = King_data[['Employed_2007','Employed_2008','Employed_2009','Em
ployed_2010','Employed_2011','Employed_2012',
        'Employed_2013','Employed_2014','Employed_2015','Employed_2016','Empl
oyed_2017','Employed_2018']]

King_Civ = King_data[['Civilian_labor_force_2007','Civilian_labor_force_2008',
'Civilian_labor_force_2009','Civilian_labor_force_2010','Civilian_labor_force_
2011','Civilian_labor_force_2012',
        'Civilian_labor_force_2013','Civilian_labor_force_2014','Civilian_lab
or_force_2015','Civilian_labor_force_2016','Civilian_labor_force_2017','Civili
an_labor_force_2018']]
```

In [30]:
```
# Plotting Unemployment in King County
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.plot(year, King_Unemployed.iloc[0]/1000000, color='red', marker='o', lines
tyle ='solid')
plt.title("Unemployment in King County")
plt.xlabel("Year")
plt.ylabel("# of People Unemployed in Millions")
plt.show()
```



In [31]:
```
# Plotting Employment and the Civilian Labor Force in King County
plt.figure(num=None, figsize=(20,10), dpi=80)
plt.plot(year, King_employed.iloc[0]/1000000, color='green', marker='o', lines
tyle ='solid')
plt.plot(year, King_Civ.iloc[0]/1000000, color='purple', marker='o', linestyle
='solid')
plt.title("Labor Force and Employment in King County")
plt.xlabel("Year")
plt.ylabel("# of People in Millions")
plt.legend(["Employed","Civilian Labor Force"])
plt.show()
```



Insight:

From the above charts we can see that the unemployment trend almost exactly followed the United States when looking at Washington state and even more specifically the area of King County. When the Employment and Civilian Labor Force data was plotted, it is clear that they have been rising since 2008 with a small dip to reflect the recession.

So what does this mean?

This data shows that employment in King County is growing and at pretty constant slope. This is a good thing since I am looking for employment in the area. Another thing to consider is the civilian labor force is also growing at the same rate. Since the civilian labor force is greater than the number employed, it is clear there is some competition, and from the trend there always will be. It will be interesting to rerun this model in the future with updated data to see if this gap widens or narrows due the emergence of new tech companies and the massive recent migrations to the Seattle area.

In [ ]: