In [1]:

```
#Paul Galvez
#DSC 680 T301
#Term Project Weeks 4-8
#Applied Data Science
```

In [2]:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
import pandas as pd
from cycler import cycler
```

In [3]:

```
df = pd.read_csv('Children_Participating.csv')
df
```

Out[3]:

	State Agency or Indian Tribal Organization	10/1/2012 0:00	11/1/2012 0:00	12/1/2012 0:00	1/1/2013 0:00	2/1/2013 0:00	3/1/2013 0:00	4/1/2013 0:00	5/1/2013 0:00	6/1/2013 0:00	7/1/2013 0:00	8/1/2013 0:00
0	Connecticut	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0
1	Maine	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0
2	Massachusetts	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0
3	New Hampshire	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0
4	New York	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	Washington	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0
88	Northern Marianas	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0
89	Inter-Tribal Council, AZ	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0
90	Navajo Nation, AZ	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0
91	Inter-Tribal Council, NV	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0

92 rows × 14 columns

In [4]:

```
#making sure the data is loaded correctly

df.head()
```

Out[4]:

	State Agency or Indian Tribal Organization	10/1/2012 0:00	11/1/2012 0:00	12/1/2012 0:00	1/1/2013 0:00	2/1/2013 0:00	3/1/2013 0:00	4/1/2013 0:00	5/1/2013 0:00	6/1/2013 0:00	7/1/2013 0:00	8/1/2013 0:00
0	Connecticut	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0
1	Maine	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0
2	Massachusetts	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0
3	New Hampshire	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0
4	New York	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0

In [5]:

```
df.tail()
```

Out[5]:

	State Agency or Indian Tribal Organization	10/1/2012 0:00	11/1/2012 0:00	12/1/2012 0:00	1/1/2013 0:00	2/1/2013 0:00	3/1/2013 0:00	4/1/2013 0:00	5/1/2013 0:00	6/1/2013 0:00	7/1/2013 0:00	8/1/2013 0:00
87	Washington	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0
88	Northern Marianas	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0
89	Inter-Tribal Council, AZ	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0
90	Navajo Nation, AZ	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0
91	Inter-Tribal Council, NV	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0

In [6]:
```
df.describe()
```

Out[6]:

	10/1/2012 0:00	11/1/2012 0:00	12/1/2012 0:00	1/1/2013 0:00	2/1/2013 0:00	3/1/2013 0:00	4/1/2013 0:00	5/1/2013 0:00
count	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000	91.000000
mean	55430.010989	54525.890110	53189.439560	54022.021978	53020.736264	52778.967033	52867.879121	53149.241758
std	114820.563346	113209.879582	110149.065962	112514.930137	110602.462521	110025.403322	110810.426967	111130.547274
min	36.000000	30.000000	36.000000	37.000000	37.000000	40.000000	44.000000	39.000000
25%	585.000000	555.500000	519.000000	555.500000	536.000000	518.500000	548.000000	562.500000
50%	12152.000000	11829.000000	11630.000000	11707.000000	11421.000000	11027.000000	10922.000000	10706.000000
75%	68030.500000	66648.000000	65335.000000	65928.000000	64995.500000	64704.500000	64545.500000	65575.500000
max	854884.000000	842587.000000	815317.000000	840108.000000	823240.000000	820593.000000	833146.000000	835846.000000

In [7]:
```
df.columns
```

Out[7]:
```
Index(['State Agency or Indian Tribal Organization', '10/1/2012 0:00',
       '11/1/2012 0:00', '12/1/2012 0:00', '1/1/2013 0:00', '2/1/2013 0:00',
       '3/1/2013 0:00', '4/1/2013 0:00', '5/1/2013 0:00', '6/1/2013 0:00',
       '7/1/2013 0:00', '8/1/2013 0:00', '9/1/2013 0:00',
       'Average Participation'],
      dtype='object')
```

In [8]:
```
df.shape
```

Out[8]:
```
(92, 14)
```

In [10]:

```
#renaming the columns for readbility. Getting rid of the 0:00 from behind the dates because it was diffic
#what the data was and renaming also was made the data cleaner.

df.rename(columns={'10/1/2012 0:00':'10/1/2012', '11/1/2012 0:00':'11/1/2012', '12/1/2012 0:00':'12/1/201
                   '1/1/2013 0:00':'1/1/2013', '2/1/2013 0:00':'2/1/2013', '3/1/2013 0:00':'3/1/2013',
                   '4/1/2013 0:00':'4/1/2013', '5/1/2013 0:00':'5/1/2013', '6/1/2013 0:00':'6/1/2013',
                   '7/1/2013 0:00':'7/1/2013', '8/1/2013 0:00':'8/1/2013', '9/1/2013 0:00':'9/1/2013'}, in
df
```

Out[10]:

	State Agency or Indian Tribal Organization	10/1/2012	11/1/2012	12/1/2012	1/1/2013	2/1/2013	3/1/2013	4/1/2013	5/1/2013	6/1/2013	7/1/2013	8/1/2013
0	Connecticut	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0
1	Maine	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0
2	Massachusetts	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0
3	New Hampshire	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0
4	New York	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	Washington	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0
88	Northern Marianas	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0
89	Inter-Tribal Council, AZ	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0
90	Navajo Nation, AZ	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0
91	Inter-Tribal Council, NV	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0

92 rows × 14 columns

In [11]:

```
#creating library for visuals and comparing the top states in terms of enrollment in WIC for kids.

my_lib = {'Connecticut':29069, 'Maine':13718, 'Massachusetts': 65049, 'New Hampshire': 8078, 'New York':2
my_lib
```

Out[11]:

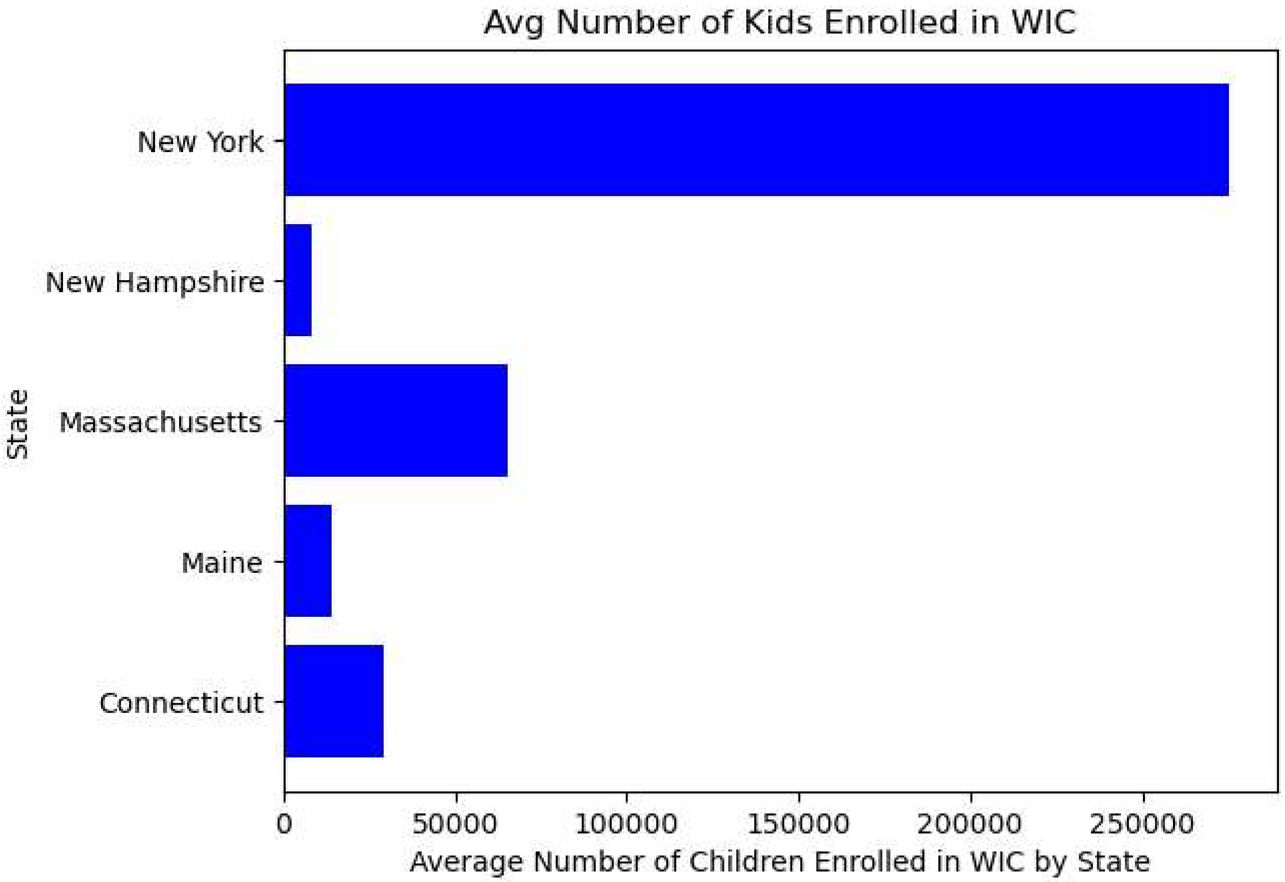
```
{'Connecticut': 29069,
 'Maine': 13718,
 'Massachusetts': 65049,
 'New Hampshire': 8078,
 'New York': 275498}
```

In [13]:

```
#bar chart below shows the comparisons per avg number of kids enrolled in WIC.

fig, ax = plt.subplots()
data = my_lib
state_name = list(data.values())
avg_prt = list(data.keys())

ax.barh(avg_prt, state_name, color='blue')
plt.xlabel('Average Number of Children Enrolled in WIC by State')
plt.ylabel('State')
plt.title('Avg Number of Kids Enrolled in WIC')
plt.show
```

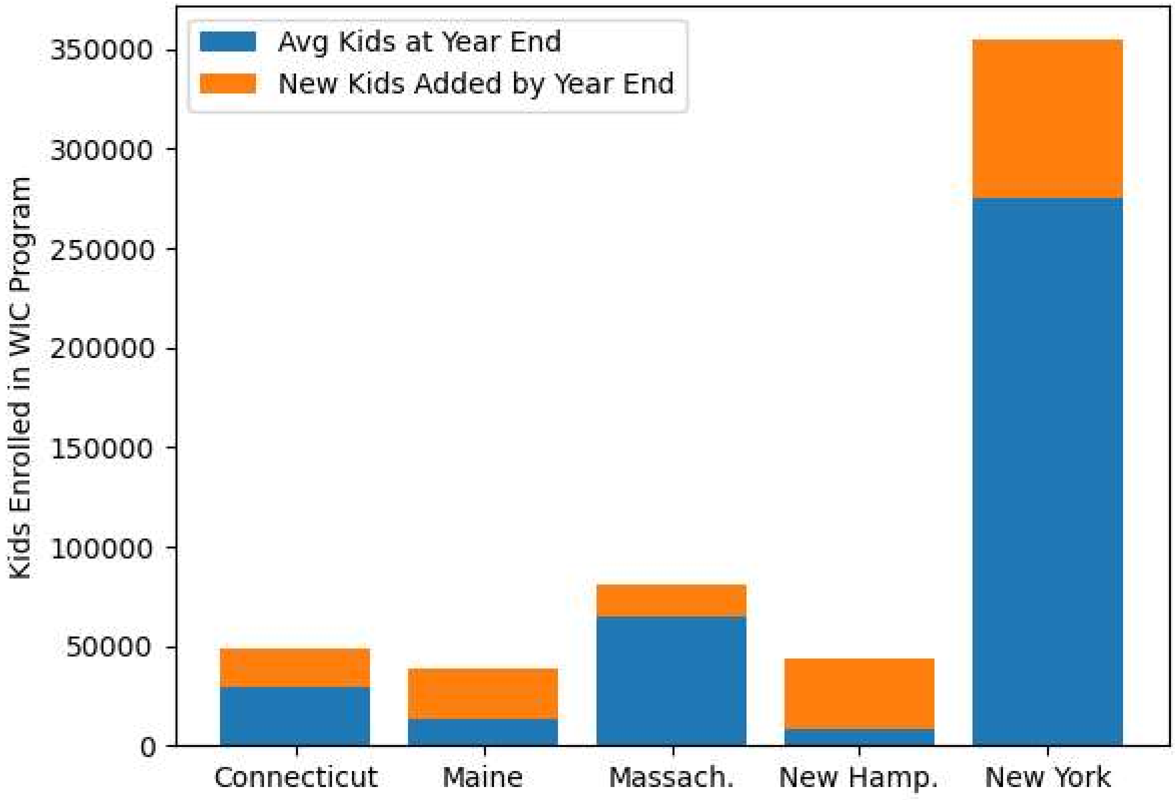
Out[13]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [14]:

```
#the stacked bar chart below shows the new kids added to the program versus the avg. We know if the new k
#programm exceeds the avg, additional resources are going to be needed in those states.

state = ['Connecticut', 'Maine', 'Massach.', 'New Hamp.', 'New York']
kids_avg = [29069, 13718, 65049, 8078, 275498]
new_kids = [19985, 24370, 15961, 35490, 78854]

fig, ax = plt.subplots()

ax.bar(state, kids_avg, label='Avg Kids at Year End')
ax.bar(state, new_kids, bottom = kids_avg, label='New Kids Added by Year End')

ax.legend()
ax.set_ylabel('Kids Enrolled in WIC Program')

plt.show()
```

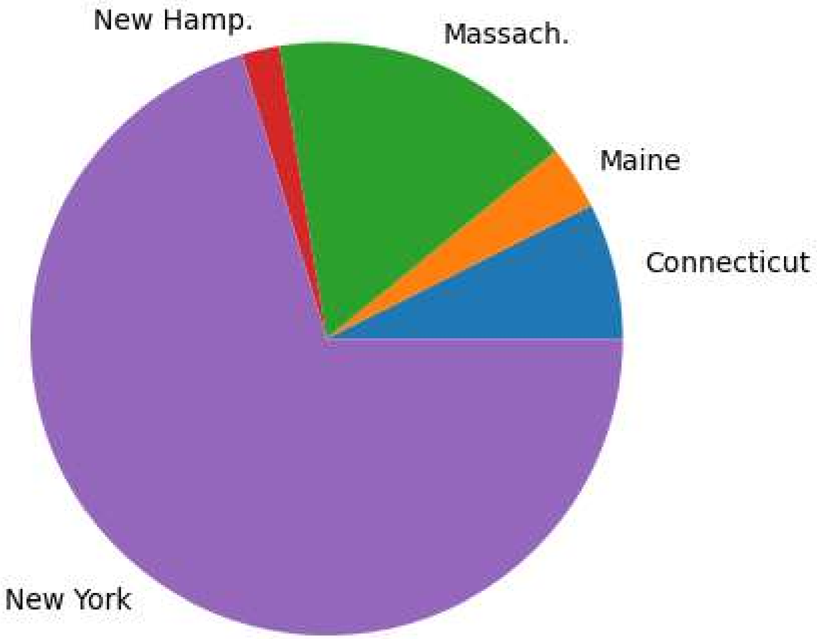


In [15]:

```
y=np.array([29069, 13718, 65049, 8078, 275498])
my_labels=['Connecticut', 'Maine', 'Massach.', 'New Hamp.', 'New York']

plt.pie(y, labels=my_labels)
plt.show
```

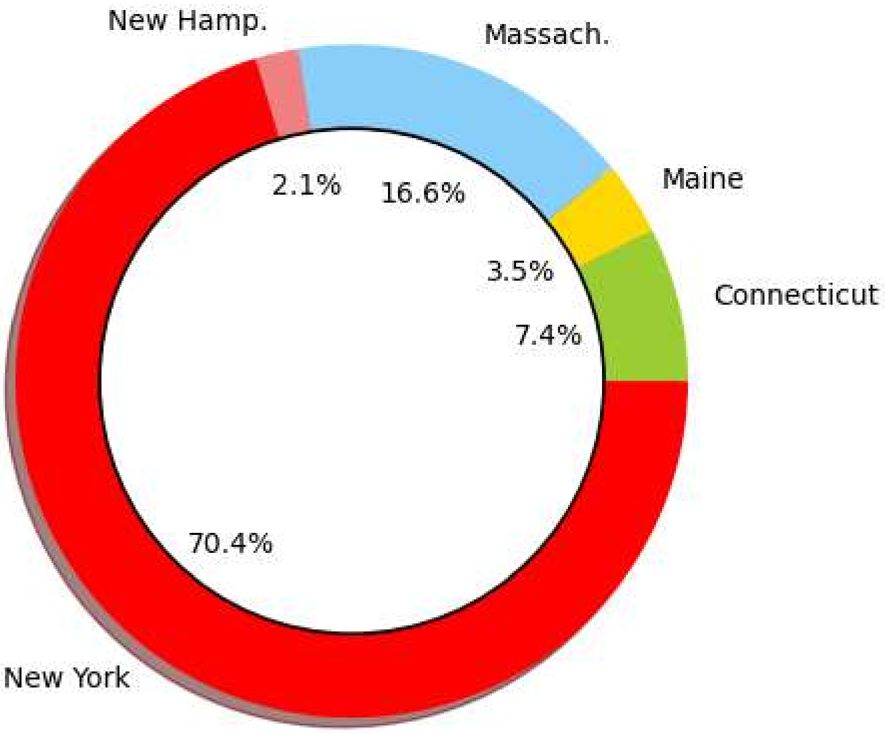
Out[15]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



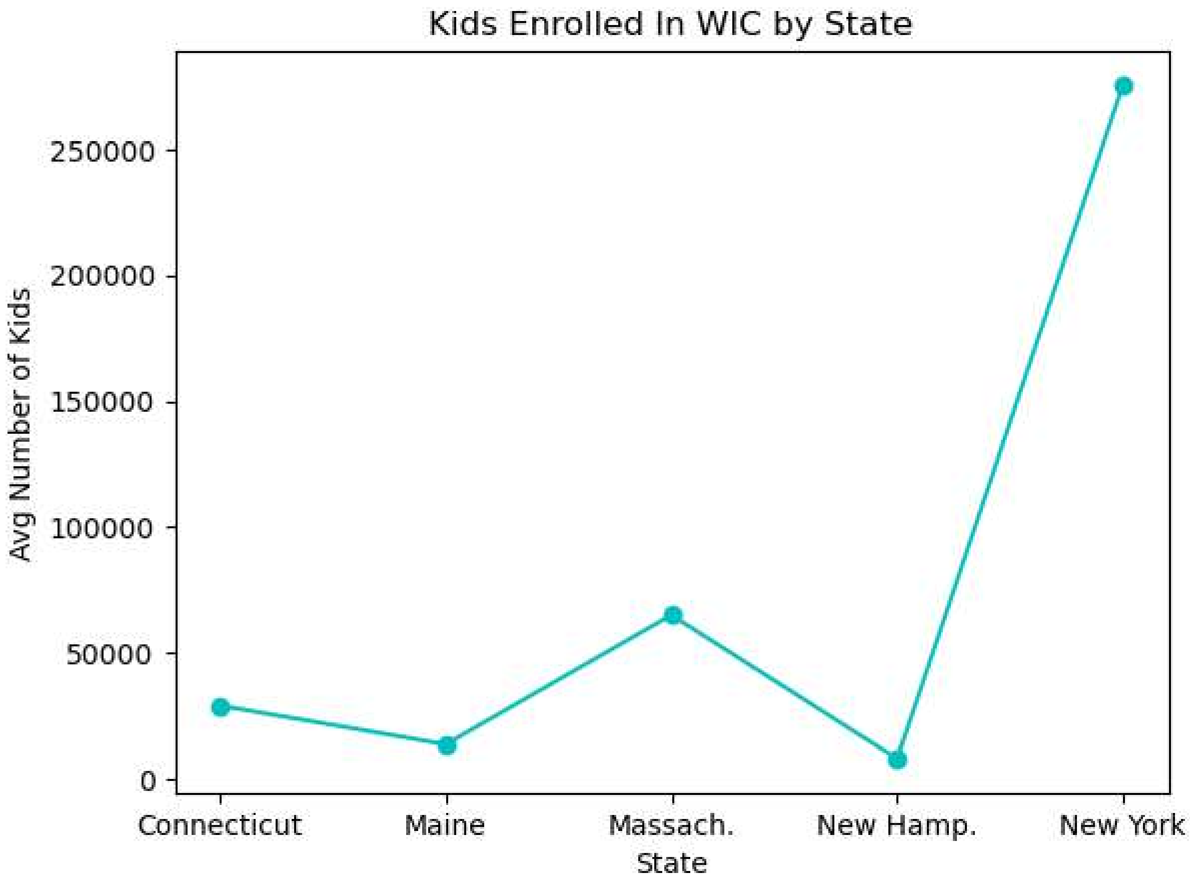
In [16]:

```
labels = 'Connecticut', 'Maine', 'Massach.', 'New Hamp.', 'New York'
sizes = [29069, 13718, 65049, 8078, 275498]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral', 'red']
explode = (0, 0, 0, 0, 0)

plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True)

centre_circle = plt.Circle((0,0),0.75,color='black', fc='white',linewidth=1.25)
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.axis('equal')
plt.show()
```



In [17]:

```
plt.plot(state, kids_avg, color ='c', marker = 'o')
plt.title('Kids Enrolled In WIC by State')
plt.xlabel('State')
plt.ylabel('Avg Number of Kids')
plt.show()
```



In [18]:

```
#dropped NA values in the dataset.

df.dropna()
```

Out[18]:

	State Agency or Indian Tribal Organization	10/1/2012	11/1/2012	12/1/2012	1/1/2013	2/1/2013	3/1/2013	4/1/2013	5/1/2013	6/1/2013	7/1/2013	8/1/2013
0	Connecticut	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0
1	Maine	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0
2	Massachusetts	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0
3	New Hampshire	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0
4	New York	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	Washington	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0
88	Northern Marianas	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0
89	Inter-Tribal Council, AZ	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0
90	Navajo Nation, AZ	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0
91	Inter-Tribal Council, NV	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0

91 rows × 14 columns

In [19]:

```
#fill in na values with 0

df.fillna(0)
df
```

Out[19]:

	State Agency or Indian Tribal Organization	10/1/2012	11/1/2012	12/1/2012	1/1/2013	2/1/2013	3/1/2013	4/1/2013	5/1/2013	6/1/2013	7/1/2013	8/1/2013
0	Connecticut	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0
1	Maine	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0
2	Massachusetts	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0
3	New Hampshire	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0
4	New York	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	Washington	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0
88	Northern Marianas	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0
89	Inter-Tribal Council, AZ	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0
90	Navajo Nation, AZ	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0
91	Inter-Tribal Council, NV	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0

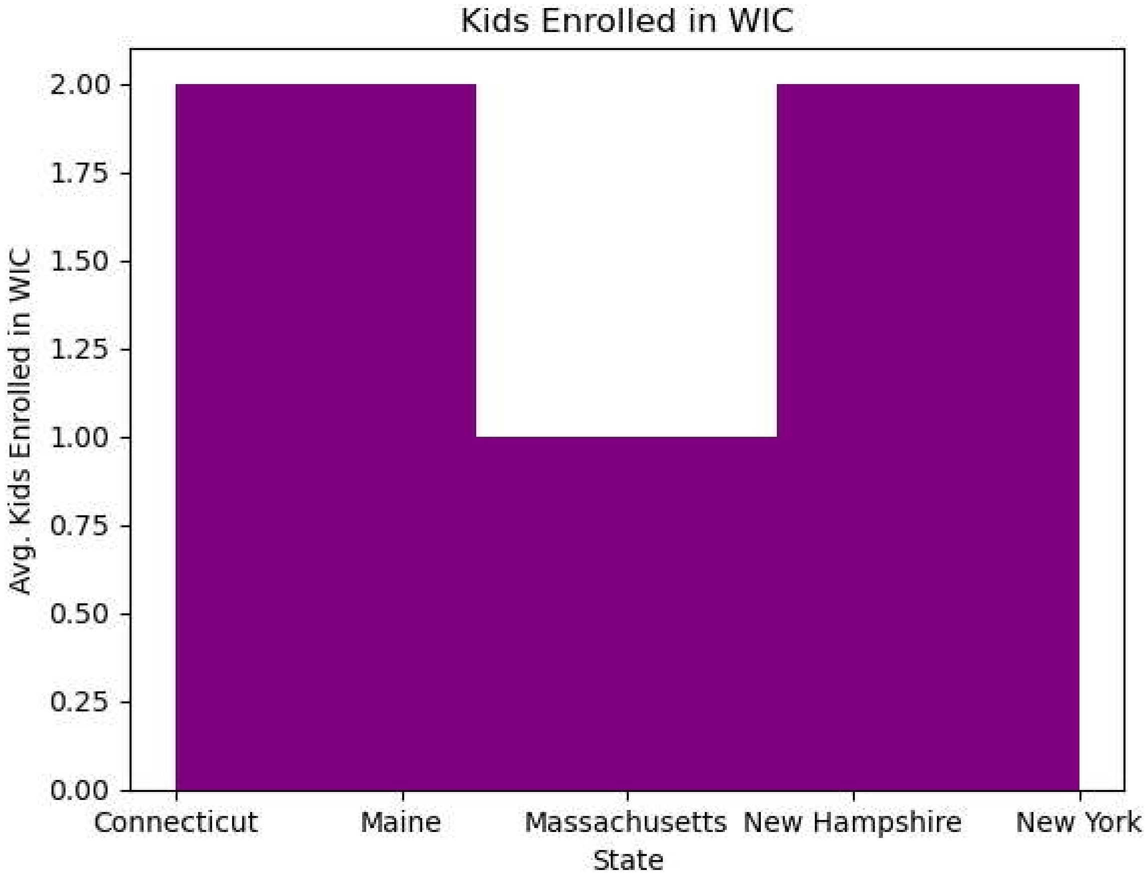
92 rows × 14 columns

In [20]:

```
plt.hist(my_lib, 3, color='purple')
plt.xlabel('State')
plt.ylabel('Avg. Kids Enrolled in WIC')
plt.title('Kids Enrolled in WIC')
plt.show
```

Out[20]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [21]:

```
df.columns
```

Out[21]:

```
Index(['State Agency or Indian Tribal Organization', '10/1/2012', '11/1/2012',
       '12/1/2012', '1/1/2013', '2/1/2013', '3/1/2013', '4/1/2013', '5/1/2013',
       '6/1/2013', '7/1/2013', '8/1/2013', '9/1/2013',
       'Average Participation'],
      dtype='object')
```

In [22]:

```
from scipy.stats import zscore
```

In [23]:

```
df = df.drop(['State Agency or Indian Tribal Organization', 'Average Participation'], axis=1)
df
```

Out[23]:

	10/1/2012	11/1/2012	12/1/2012	1/1/2013	2/1/2013	3/1/2013	4/1/2013	5/1/2013	6/1/2013	7/1/2013	8/1/2013	9/1/2013
0	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0	29569.0
1	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0	13293.0
2	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0	67373.0
3	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0	7773.0
4	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0	273708.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0	108719.0
88	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0	2580.0
89	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0	5517.0
90	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0	5745.0
91	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0	836.0

92 rows × 12 columns

In [24]:

```
df.dtypes
```

Out[24]:

```
10/1/2012    float64
11/1/2012    float64
12/1/2012    float64
1/1/2013     float64
2/1/2013     float64
3/1/2013     float64
4/1/2013     float64
5/1/2013     float64
6/1/2013     float64
7/1/2013     float64
8/1/2013     float64
9/1/2013     float64
dtype: object
```

In [25]:

```
df = df.dropna()
df
```

Out[25]:

	10/1/2012	11/1/2012	12/1/2012	1/1/2013	2/1/2013	3/1/2013	4/1/2013	5/1/2013	6/1/2013	7/1/2013	8/1/2013	9/1/2013
0	29985.0	29349.0	28559.0	29552.0	27948.0	27988.0	28623.0	29471.0	29046.0	29337.0	29405.0	29569.0
1	14370.0	13733.0	13705.0	13941.0	13857.0	13763.0	13707.0	13790.0	13649.0	13406.0	13410.0	13293.0
2	65961.0	64813.0	63693.0	65032.0	63698.0	63879.0	64072.0	64882.0	63788.0	66368.0	67032.0	67373.0
3	8490.0	8527.0	8128.0	8280.0	8007.0	8004.0	8078.0	8069.0	7941.0	7825.0	7817.0	7773.0
4	278854.0	275401.0	270033.0	274112.0	274773.0	275079.0	277498.0	278179.0	277716.0	276189.0	274443.0	273708.0
...	...	...	...	...	...	...	...	...	...	...	...	...
87	113842.0	112055.0	111888.0	110893.0	108994.0	108938.0	108722.0	110231.0	108794.0	109263.0	109630.0	108719.0
88	2808.0	2753.0	2719.0	2744.0	2688.0	2765.0	2676.0	2724.0	2720.0	2676.0	2683.0	2580.0
89	6026.0	5983.0	5595.0	5748.0	5267.0	5293.0	5368.0	5436.0	5590.0	5740.0	5740.0	5517.0
90	6380.0	6144.0	5961.0	6187.0	5726.0	5968.0	5945.0	5914.0	5766.0	5794.0	5952.0	5745.0
91	807.0	783.0	756.0	763.0	745.0	750.0	755.0	760.0	759.0	743.0	794.0	836.0

91 rows × 12 columns

In [26]:

```
X = df.drop(['6/1/2013'], axis=1)
y = df['6/1/2013']
```

In [27]:

```
#Loading the required libraries

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder,OneHotEncoder
```

In [28]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

In [29]:

```
model_linear_regression = LinearRegression()
model_linear_regression.fit(X_train, y_train)
y_pred = model_linear_regression.predict(X_test)
```

In [30]:

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
print(f'The RMSE value is: {rmse}')
print(f'The R2 value is: {r2}')
```

```
The RMSE value is: 1670.5932634509413
The R2 value is: 0.9998806245092784
```

In [31]:

```
model_linear_regression.coef_
```

Out[31]:

```
array([ 0.17810287, -0.34217395,  0.05590812, -0.00697537,  0.19766888,
       -0.04448484, -0.25076899,  0.79487417,  0.52783041, -0.12624859,
        0.01641258])
```

In [32]:

```
model_linear_regression.intercept_
```

Out[32]:

```
2.5691787285904866
```

In [33]:

```
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn import utils
```

In [34]:

```
lab = preprocessing.LabelEncoder()
y_transformed = lab.fit_transform(y)
```

In [35]:

```
print(y_transformed)
```

```
[52 47 65 40 87 46 41  0  1  2 44 39 72 76 81 78 73 30 51 70 85 84 66 58
 82 62 71 21 15 83 74 80 69 79 64 60 68 53 59 88 12  6 10 19  9  5 16 32
 27 29 22 18 26 25 14 28 61 54 56 67 43 49 38 42 55 37  5 11  4  7 20  8
 17 23 13  3 86 45 34 75 89 33 48 50 57 63 77 31 35 36 24]
```

In [36]:

```
X = df.drop(['7/1/2013'], axis=1)
y = df['7/1/2013']
```

In [37]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

In [38]:

```
model_linear_regression = LinearRegression()
model_linear_regression.fit(X_train, y_train)
y_pred = model_linear_regression.predict(X_test)
```

In [39]:

```
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
print(f'The RMSE value is: {rmse}')
print(f'The R2 value is: {r2}')
```

```
The RMSE value is: 1346.6019178344613
The R2 value is: 0.9999239075163163
```

In [40]:

```
model_linear_regression.coef_
```

Out[40]:

```
array([-0.23649657,  0.38238559,  0.03601638, -0.20711683,  0.00148828,
       -0.11973234,  0.03480782, -0.07494023,  0.7446604 ,  0.46534945,
       -0.02588198])
```

In [41]:

```
model_linear_regression.intercept_
```

Out[41]:

```
14.187039704644121
```