In [1]:
```
#load libraries
import pandas as pd
import seaborn as sns

import numpy as np
import matplotlib.pyplot as plt

#import squarify
#import matplotlib.ticker as plticker # for plot ticks
```

In [2]:
```
#import data as dataframe

nba = pd.read_csv('ppg2008.csv', index_col=0)
costco = pd.read_csv('costcos-geocoded.csv') #tab seperator
```

In [3]:
```
nba.head()
```

Out[3]:

	G	MIN	PTS	FGM	FGA	FGP	FTM	FTA	FTP	3PM	3PA	3PP	ORB	DRB	TRB
Name															
Dwyane Wade	79	38.6	30.2	10.8	22.0	0.491	7.5	9.8	0.765	1.1	3.5	0.317	1.1	3.9	5.0
LeBron James	81	37.7	28.4	9.7	19.9	0.489	7.3	9.4	0.780	1.6	4.7	0.344	1.3	6.3	7.6
Kobe Bryant	82	36.2	26.8	9.8	20.9	0.467	5.9	6.9	0.856	1.4	4.1	0.351	1.1	4.1	5.2
Dirk Nowitzki	81	37.7	25.9	9.6	20.0	0.479	6.0	6.7	0.890	0.8	2.1	0.359	1.1	7.3	8.4
Danny Granger	67	36.2	25.8	8.5	19.1	0.447	6.0	6.9	0.878	2.7	6.7	0.404	0.7	4.4	5.1

In [4]:

```
nba.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, Dwyane Wade  to Nate Robinson
Data columns (total 20 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   G       50 non-null     int64  
 1   MIN     50 non-null     float64
 2   PTS     50 non-null     float64
 3   FGM     50 non-null     float64
 4   FGA     50 non-null     float64
 5   FGP     50 non-null     float64
 6   FTM     50 non-null     float64
 7   FTA     50 non-null     float64
 8   FTP     50 non-null     float64
 9   3PM     50 non-null     float64
 10  3PA     50 non-null     float64
 11  3PP     50 non-null     float64
 12  ORB     50 non-null     float64
 13  DRB     50 non-null     float64
 14  TRB     50 non-null     float64
 15  AST     50 non-null     float64
 16  STL     50 non-null     float64
 17  BLK     50 non-null     float64
 18  TO      50 non-null     float64
 19  PF      50 non-null     float64
dtypes: float64(19), int64(1)
memory usage: 8.2+ KB
```

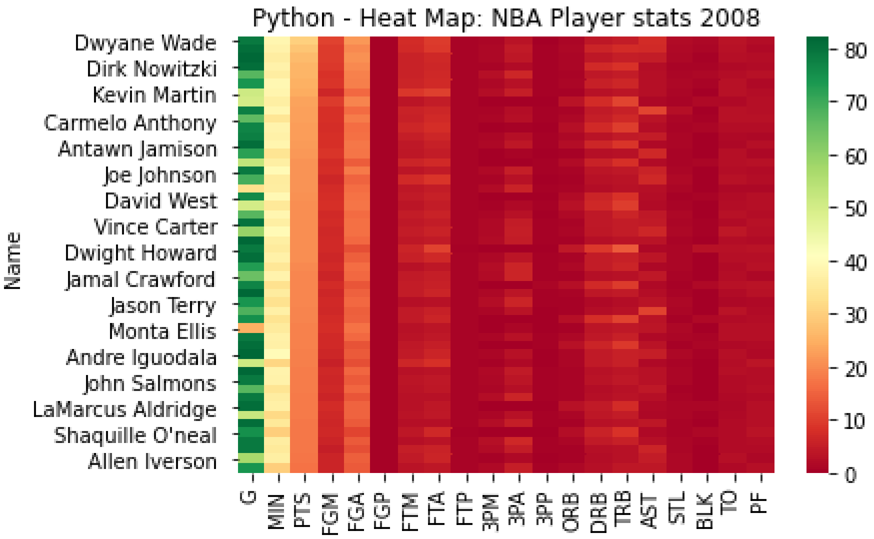
Heatmap

In [5]:

```
# Default heatmap
sns.heatmap(nba,cmap ='RdYlGn')
plt.title("Python - Heat Map: NBA Player stats 2008")
```

Out[5]:

```
Text(0.5, 1.0, 'Python - Heat Map: NBA Player stats 2008')
```



Spatial Chart

In [6]:
```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
import numpy as np
```

In [7]:
```
costco
```

Out[7]:

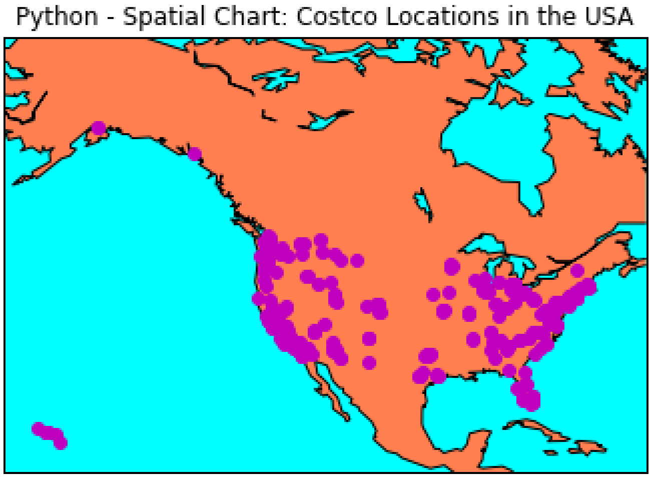
	Address	City	State	Zip Code	Latitude	Longitude
0	1205 N. Memorial Parkway	Huntsville	Alabama	35801-5930	34.743095	-86.600955
1	3650 Galleria Circle	Hoover	Alabama	35244-2346	33.377649	-86.812420
2	8251 Eastchase Parkway	Montgomery	Alabama	36117	32.363889	-86.150884
3	5225 Commercial Boulevard	Juneau	Alaska	99801-7210	58.359200	-134.483000
4	330 West Dimond Blvd	Anchorage	Alaska	99515-1950	61.143266	-149.884217
...	...	...	...	...	...	...
412	19610 SE 1st St	Vancouver	Washington	98607	45.621299	-122.459135
413	10990 Harbor Hill Dr	Gig Harbor	Washington	98335	47.357748	-122.603888
414	27520 Covington Way SE	Covington	Washington	98042	47.354838	-122.121185
415	2150 Deming Way	Middleton	Wisconsin	53562-5507	43.100195	-89.522751
416	950 Port Washington Rd	Grafton	Wisconsin	53024-9201	43.324691	-87.921615

417 rows × 6 columns

In [8]:
```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
```

In [9]:

```
#https://basemaptutorial.readthedocs.io/en/latest/plotting_data.html

m = Basemap(projection='mill',
              lat_0=39, lon_0=-100,
           llcrnrlon=-165,llcrnrlat=15,urcrnrlon=-62,urcrnrlat=70,)

#good god this is a PITA
#AK 71, -164
#FL 22, -78
#ME 46, -63
#HI 18, -155

m.drawmapboundary(fill_color='aqua')
m.fillcontinents(color='coral',lake_color='aqua')
m.drawcoastlines()

lons = costco['Longitude']
lats = costco['Latitude']

x, y = m(lons, lats)

m.scatter(x, y, marker='o',color='m',zorder=5)
plt.title("Python - Spatial Chart: Costco Locations in the USA")
plt.show()
```



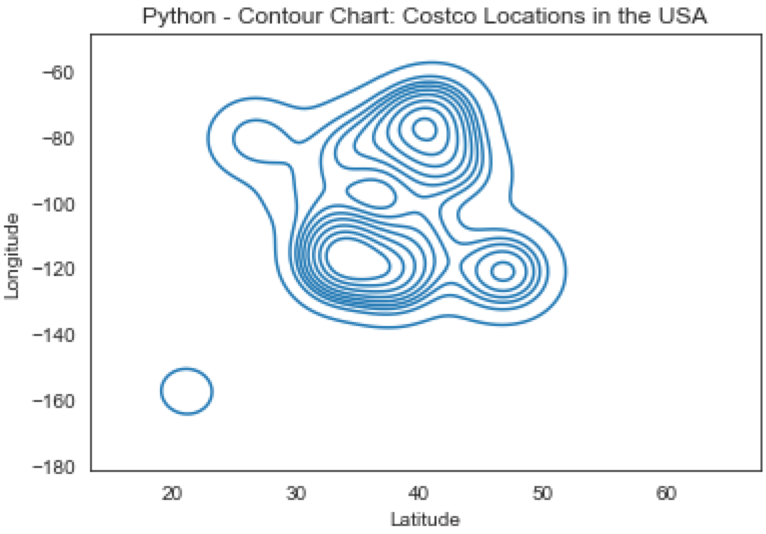
Contour Plot

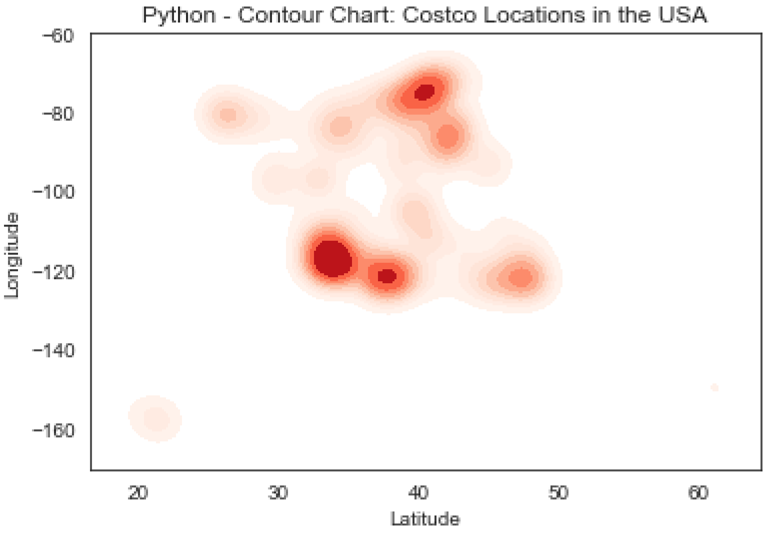
In [10]:

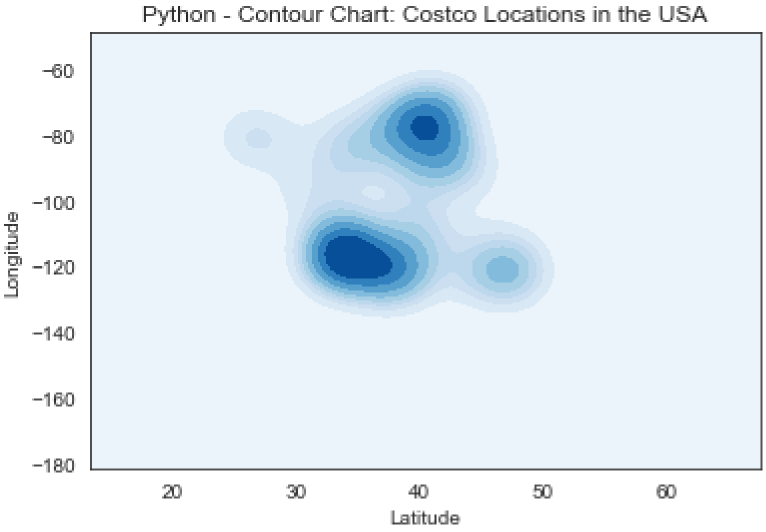
```
costco.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 417 entries, 0 to 416
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Address    417 non-null    object 
 1   City       417 non-null    object 
 2   State      417 non-null    object 
 3   Zip Code   417 non-null    object 
 4   Latitude   417 non-null    float64
 5   Longitude  417 non-null    float64
dtypes: float64(2), object(4)
memory usage: 19.7+ KB
```

In [11]:
```
# set seaborn style
sns.set_style("white")

# Basic 2D density plot
sns.kdeplot(x=costco.Latitude, y=costco.Longitude)
plt.title("Python - Contour Chart: Costco Locations in the USA")
plt.show()

# Custom the color, add shade and bandwidth
sns.kdeplot(x=costco.Latitude, y=costco.Longitude, cmap="Reds", shade=True, bw_ad
plt.title("Python - Contour Chart: Costco Locations in the USA")
plt.show()

# Add thresh parameter
sns.kdeplot(x=costco.Latitude, y=costco.Longitude, cmap="Blues", shade=True, thre
plt.title("Python - Contour Chart: Costco Locations in the USA")
plt.show()
```






Python - Contour Chart: Costco Locations in the USA
Longitude
−60
−80
−100
−120
−140
−160
−180
20
30
40
50
60
Latitude