In [58]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [59]:
```
df= pd.read_csv("ppg2008.csv")
df
```

Out[59]:

	Name	G	MIN	PTS	FGM	FGA	FGP	FTM	FTA	FTP	...	3PA	3PP	ORB	DR
0	Dwyane Wade	79	38.6	30.2	10.8	22.0	0.491	7.5	9.8	0.765	...	3.5	0.317	1.1	3.
1	LeBron James	81	37.7	28.4	9.7	19.9	0.489	7.3	9.4	0.780	...	4.7	0.344	1.3	6.
2	Kobe Bryant	82	36.2	26.8	9.8	20.9	0.467	5.9	6.9	0.856	...	4.1	0.351	1.1	4.
3	Dirk Nowitzki	81	37.7	25.9	9.6	20.0	0.479	6.0	6.7	0.890	...	2.1	0.359	1.1	7.
4	Danny Granger	67	36.2	25.8	8.5	19.1	0.447	6.0	6.9	0.878	...	6.7	0.404	0.7	4.
5	Kevin Durant	74	39.0	25.3	8.9	18.8	0.476	6.1	7.1	0.863	...	3.1	0.422	1.0	5.
6	Kevin Martin	51	38.2	24.6	6.7	15.9	0.420	9.0	10.3	0.867	...	5.4	0.415	0.6	3.
7	Al Jefferson	50	36.6	23.1	9.7	19.5	0.497	3.7	5.0	0.738	...	0.1	0.000	3.4	7.
8	Chris Paul	78	38.5	22.8	8.1	16.1	0.503	5.8	6.7	0.868	...	2.3	0.364	0.9	4.
9	Carmelo Anthony	66	34.5	22.8	8.1	18.3	0.443	5.6	7.1	0.793	...	2.6	0.371	1.6	5.
10	Chris Bosh	77	38.1	22.7	8.0	16.4	0.487	6.5	8.0	0.817	...	0.6	0.245	2.8	7.
11	Brandon Roy	78	37.2	22.6	8.1	16.9	0.480	5.3	6.5	0.824	...	2.8	0.377	1.3	3.
12	Antawn Jamison	81	38.2	22.2	8.3	17.8	0.468	4.2	5.6	0.754	...	3.9	0.351	2.4	6.
13	Tony Parker	72	34.1	22.0	8.9	17.5	0.506	3.9	5.0	0.782	...	0.9	0.292	0.4	2.
14	Amare Stoudemire	53	36.8	21.4	7.6	14.1	0.539	6.1	7.3	0.835	...	0.1	0.429	2.2	5.
15	Joe Johnson	79	39.5	21.4	7.8	18.0	0.437	3.8	4.6	0.826	...	5.2	0.360	0.8	3.
16	Devin	69	36.1	21.3	6.6	15.1	0.438	7.2	8.8	0.820	...	3.2	0.291	0.4	2.

	Harris														
17	Michael Redd	33	36.4	21.2	7.5	16.6	0.455	4.0	4.9	0.814	...	5.8	0.366	0.7	2.
18	David West	76	39.3	21.0	8.0	17.0	0.472	4.8	5.5	0.884	...	0.3	0.240	2.1	6.
19	Zachary Randolph	50	35.1	20.8	8.3	17.5	0.475	3.6	4.9	0.734	...	1.9	0.330	3.1	6.
20	Caron Butler	67	38.6	20.8	7.3	16.2	0.453	5.1	6.0	0.858	...	3.1	0.310	1.8	4.
21	Vince Carter	80	36.8	20.8	7.4	16.8	0.437	4.2	5.1	0.817	...	4.9	0.385	0.9	4.
22	Stephen Jackson	59	39.7	20.7	7.0	16.9	0.414	5.0	6.0	0.826	...	5.2	0.338	1.2	3.
23	Ben Gordon	82	36.6	20.7	7.3	16.0	0.455	4.0	4.7	0.864	...	5.1	0.410	0.6	2.
24	Dwight Howard	79	35.7	20.6	7.1	12.4	0.572	6.4	10.7	0.594	...	0.0	0.000	4.3	9.
25	Paul Pierce	81	37.4	20.5	6.7	14.6	0.457	5.7	6.8	0.830	...	3.8	0.391	0.7	5.
26	Al Harrington	73	34.9	20.1	7.3	16.6	0.439	3.2	4.0	0.793	...	6.4	0.364	1.4	4.
27	Jamal Crawford	65	38.1	19.7	6.4	15.7	0.410	4.6	5.3	0.872	...	6.1	0.360	0.4	2.
28	Yao Ming	77	33.6	19.7	7.4	13.4	0.548	4.9	5.7	0.866	...	0.0	1.000	2.6	7.
29	Richard Jefferson	82	35.9	19.6	6.5	14.9	0.439	5.1	6.3	0.805	...	3.6	0.397	0.7	3.
30	Jason Terry	74	33.6	19.6	7.3	15.8	0.463	2.7	3.0	0.880	...	6.2	0.366	0.5	1.
31	Deron Williams	68	36.9	19.4	6.8	14.5	0.471	4.8	5.6	0.849	...	3.3	0.310	0.4	2.
32	Tim Duncan	75	33.7	19.3	7.4	14.8	0.504	4.5	6.4	0.692	...	0.0	0.000	2.7	8.
33	Monta Ellis	25	35.6	19.0	7.8	17.2	0.451	3.1	3.8	0.830	...	1.0	0.308	0.6	3.
34	Rudy Gay	79	37.3	18.9	7.2	16.0	0.453	3.3	4.4	0.767	...	3.1	0.351	1.4	4.
35	Pau Gasol	81	37.1	18.9	7.3	12.9	0.567	4.2	5.4	0.781	...	0.0	0.500	3.2	6.
36	Andre Iguodala	82	39.8	18.8	6.6	14.0	0.473	4.6	6.4	0.724	...	3.2	0.307	1.1	4.
37	Corey Maggette	51	31.1	18.6	5.7	12.4	0.461	6.7	8.1	0.824	...	1.9	0.253	1.0	4.
38	O.J. Mayo	82	38.0	18.5	6.9	15.6	0.438	3.0	3.4	0.879	...	4.6	0.384	0.7	3.
39	John Salmons	79	37.5	18.3	6.5	13.8	0.472	3.6	4.4	0.830	...	3.8	0.417	0.7	3.
	Richard														

40	Hamilton	67	34.0	18.3	7.0	15.6	0.447	3.3	3.9	0.848	...	2.8	0.368	0.7	2.
41	Ray Allen	79	36.3	18.2	6.3	13.2	0.480	3.0	3.2	0.952	...	6.2	0.409	0.8	2.
42	LaMarcus Aldridge	81	37.1	18.1	7.4	15.3	0.484	3.2	4.1	0.781	...	0.3	0.250	2.9	4.
43	Josh Howard	52	31.9	18.0	6.8	15.1	0.451	3.3	4.2	0.782	...	3.2	0.345	1.1	3.
44	Maurice Williams	81	35.0	17.8	6.5	13.9	0.467	2.6	2.8	0.912	...	5.2	0.436	0.6	2.
45	Shaquille O'neal	75	30.1	17.8	6.8	11.2	0.609	4.1	6.9	0.595	...	0.0	0.000	2.5	5.
46	Rashard Lewis	79	36.2	17.7	6.1	13.8	0.439	2.8	3.4	0.836	...	7.0	0.397	1.2	4.
47	Chauncey Billups	79	35.3	17.7	5.2	12.4	0.418	5.3	5.8	0.913	...	5.0	0.408	0.4	2.
48	Allen Iverson	57	36.7	17.5	6.1	14.6	0.417	4.8	6.1	0.781	...	1.7	0.283	0.5	2.
49	Nate Robinson	74	29.9	17.2	6.1	13.9	0.437	3.4	4.0	0.841	...	5.2	0.325	1.3	2.

50 rows × 21 columns

In [60]:
```
assist=df['AST']
turn=df['TO']
pnt=df['PTS']
df1=pd.concat([assist,turn,pnt], axis=1)
df1
```

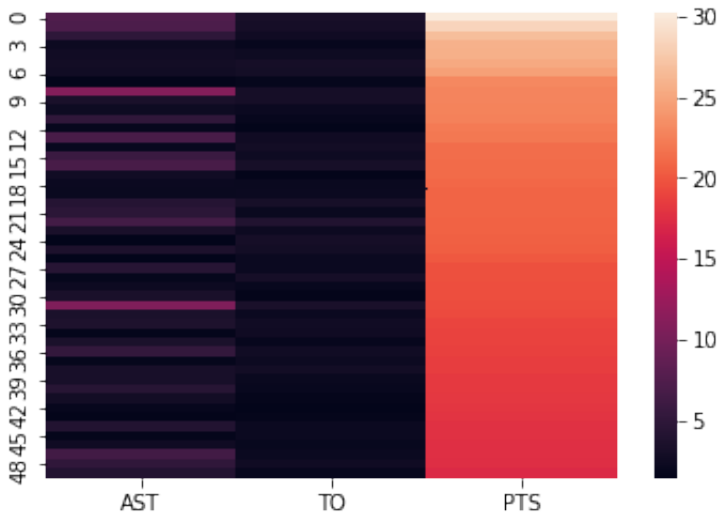
Out[60]:

	AST	TO	PTS
0	7.5	3.4	30.2
1	7.2	3.0	28.4
2	4.9	2.6	26.8
3	2.4	1.9	25.9
4	2.7	2.5	25.8
5	2.8	3.0	25.3
6	2.7	2.9	24.6
7	1.6	1.8	23.1
8	11.0	3.0	22.8
9	3.4	3.0	22.8

10	2.5	2.3	22.7
11	5.1	1.9	22.6
12	1.9	1.5	22.2
13	6.9	2.6	22.0
14	2.0	2.8	21.4
15	5.8	2.5	21.4
16	6.9	3.1	21.3
17	2.7	1.6	21.2
18	2.3	2.1	21.0
19	2.1	2.3	20.8
20	4.3	3.1	20.8
21	4.7	2.1	20.8
22	6.5	3.9	20.7
23	3.4	2.4	20.7
24	1.4	3.0	20.6
25	3.6	2.8	20.5
26	1.4	2.2	20.1
27	4.4	2.3	19.7
28	1.8	3.0	19.7
29	2.4	2.0	19.6
30	3.4	1.6	19.6
31	10.7	3.4	19.4
32	3.5	2.2	19.3
33	3.7	2.7	19.0
34	1.7	2.6	18.9
35	3.5	1.9	18.9
36	5.3	2.7	18.8
37	1.8	2.4	18.6
38	3.2	2.8	18.5
39	3.2	2.1	18.3
40	4.4	2.0	18.3
41	2.8	1.7	18.2

42	1.9	1.5	18.1
43	1.6	1.7	18.0
44	4.1	2.2	17.8
45	1.7	2.2	17.8
46	2.6	2.0	17.7
47	6.4	2.2	17.7
48	5.0	2.6	17.5
49	4.1	1.9	17.2

In [61]:
```
ax = sns.heatmap(df1)
plt.show()
```



In [62]:
```
df2=pd.read_csv('costcos-geocoded.csv')
df2
```

Out[62]:

	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

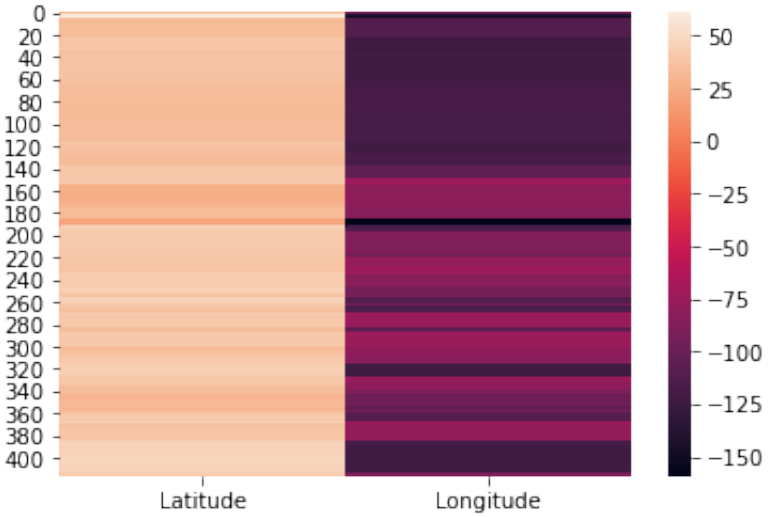
In [63]:
```
zipc=df2['Zip Code']
lat=df2['Latitude']
long=df2['Longitude']
df3=pd.concat([lat,long], axis=1)
df3
```

Out[63]:

	Latitude	Longitude
0	34.743095	-86.600955
1	33.377649	-86.812420
2	32.363889	-86.150884
3	58.359200	-134.483000
4	61.143266	-149.884217
...	...	...
412	45.621299	-122.459135
413	47.357748	-122.603888
414	47.354838	-122.121185
415	43.100195	-89.522751
416	43.324691	-87.921615

417 rows × 2 columns

In [64]:
```
ax = sns.heatmap(df3)
plt.show()
```



In [65]:
```
import geoplot as gplt
import geopandas as gpd
import geoplot.crs as gcrs
import imageio
import pathlib
import mapclassify as mc
```

In [66]:
```
%matplotlib inline
```

In [67]:
```
states = gpd.read_file('costcos-geocoded.csv')
```

In [69]:
```
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```

In [70]:
```
cities = gpd.read_file(gpd.datasets.get_path('naturalearth_cities'))
```

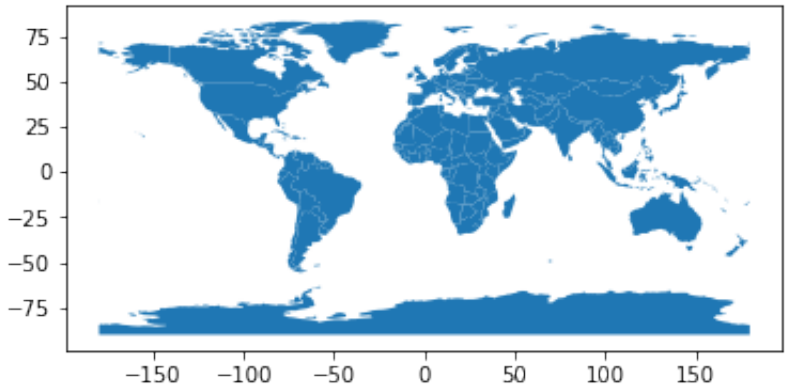
In [71]:
```
world.head()
```

Out[71]:

	pop_est	continent	name	iso_a3	gdp_md_est	geometry
0	920938	Oceania	Fiji	FJI	8374.0	MULTIPOLYGON (((180.00000 -16.06713, 180.00000...
1	53950935	Africa	Tanzania	TZA	150600.0	POLYGON ((33.90371 -0.95000, 34.07262 -1.05982...
2	603253	Africa	W. Sahara	ESH	906.5	POLYGON ((-8.66559 27.65643, -8.66512 27.58948...
3	35623680	North America	Canada	CAN	1674000.0	MULTIPOLYGON (((-122.84000 49.00000, -122.9742...
4	326625791	North America	United States of America	USA	18560000.0	MULTIPOLYGON (((-122.84000 49.00000, -120.0000...

In [72]:
```
world.plot()
```

Out[72]: <AxesSubplot:>



In [78]:
```
usa = gpd.read_file("cb_2018_us_state_20m/cb_2018_us_state_20m.shp")
usa.head()
```

Out[78]:

	STATEFP	STATENS	AFFGEOID	GEOID	STUSPS	NAME	LSAD	ALAND	
0	24	01714934	0400000US24	24	MD	Maryland	00	25151100280	69
1	19	01779785	0400000US19	19	IA	Iowa	00	144661267977	1(
2	10	01779781	0400000US10	10	DE	Delaware	00	5045925646	13
3	39	01085497	0400000US39	39	OH	Ohio	00	105828882568	102
4	42	01779798	0400000US42	42	PA	Pennsylvania	00	115884442321	33

In [80]:
```
state_pop = pd.read_csv("nst-est2018-alldata.csv")
state_pop.head()
```

Out[80]:

	SUMLEV	REGION	DIVISION	STATE	NAME	CENSUS2010POP	ESTIMATESBASE2010	P
0	10	0	0	0	United States	308745538	308758105	
1	20	1	0	0	Northeast Region	55317240	55318430	
2	20	2	0	0	Midwest Region	66927001	66929743	
3	20	3	0	0	South Region	114555744	114563045	
4	20	4	0	0	West Region	71945553	71946887	

5 rows × 136 columns

In [81]:
```
pop_states = usa.merge(state_pop, left_on="NAME", right_on="NAME")
pop_states.head()
```

Out[81]:

	STATEFP	STATENS	AFFGEOID	GEOID	STUSPS	NAME	LSAD	ALAND	
0	24	01714934	0400000US24	24	MD	Maryland	00	25151100280	69
1	19	01779785	0400000US19	19	IA	Iowa	00	144661267977	10
2	10	01779781	0400000US10	10	DE	Delaware	00	5045925646	13
3	39	01085497	0400000US39	39	OH	Ohio	00	105828882568	102
4	42	01779798	0400000US42	42	PA	Pennsylvania	00	115884442321	33

5 rows × 145 columns

In [82]:
```
path = gplt.datasets.get_path("contiguous_usa")
contiguous_usa = gpd.read_file(path)
```

In [83]:
```
gplt.polyplot(contiguous_usa)
```

Out[83]: <AxesSubplot:>

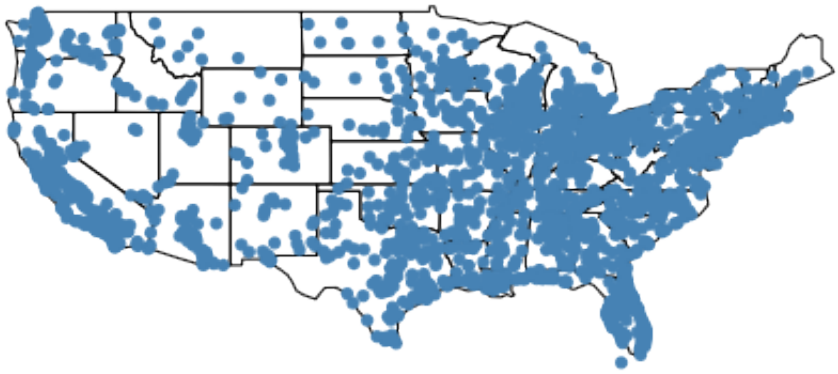
In [84]:
```
path = gplt.datasets.get_path("usa_cities")
usa_cities = gpd.read_file(path)
```

In [85]:
```
continental_usa_cities = usa_cities.query('STATE not in ["HI", "AK", "
gplt.pointplot(continental_usa_cities)
```

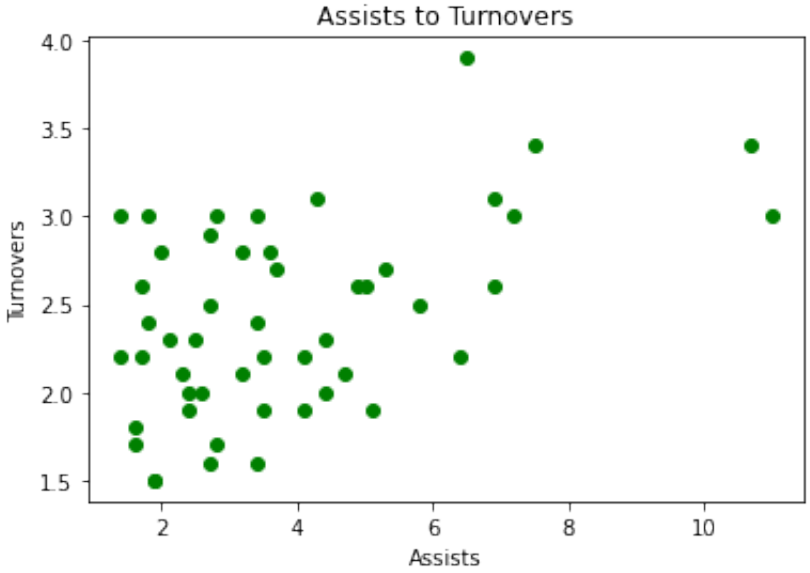
Out[85]: <AxesSubplot:>

In [86]:
```
ax = gplt.polyplot(contiguous_usa)
gplt.pointplot(continental_usa_cities, ax=ax)
```

Out[86]: <AxesSubplot:>



In [91]:
```
X=assist
Y=turn
fig, ax = plt.subplots()
ax.scatter(X, Y, color="green")
ax.set_title('Assists to Turnovers')
ax.set_xlabel('Assists')
ax.set_ylabel('Turnovers')
plt.show()
```

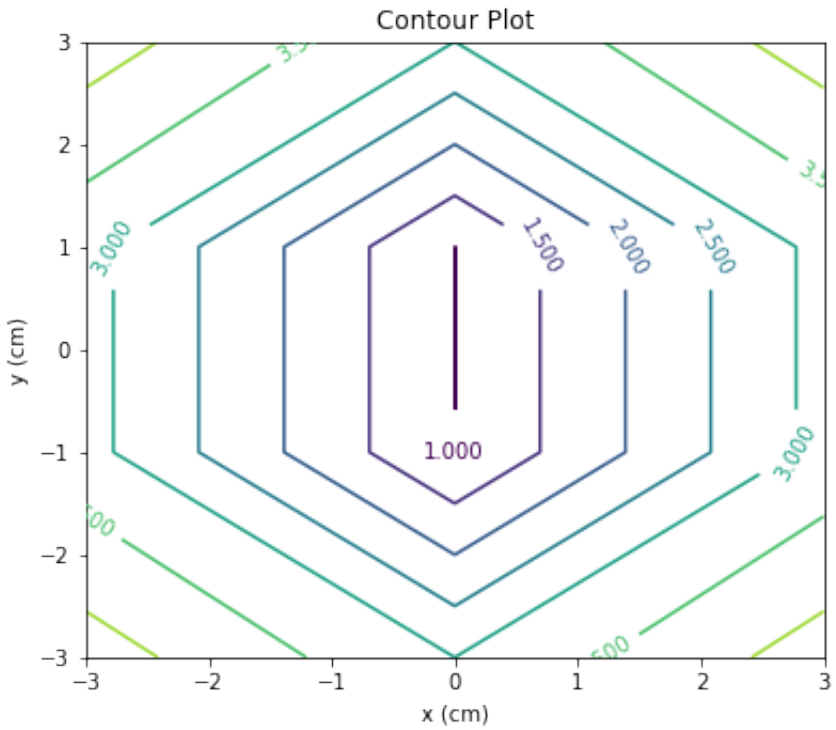


In [94]:
```
xlist = np.linspace(-3.0, 3.0, 3)
ylist = np.linspace(-3.0, 3.0, 4)
X, Y = np.meshgrid(xlist, ylist)
Z = np.sqrt(X**2 + Y**2)
print(Z)
```

```
[[4.24264069 3.         4.24264069]
 [3.16227766 1.         3.16227766]
 [3.16227766 1.         3.16227766]
 [4.24264069 3.         4.24264069]]
```

In [95]:
```
fig = plt.figure(figsize=(6,5))
left, bottom, width, height = 0.1, 0.1, 0.8, 0.8
ax = fig.add_axes([left, bottom, width, height])


Z = np.sqrt(X**2 + Y**2)
cp = ax.contour(X, Y, Z)
ax.clabel(cp, inline=True,
          fontsize=10)
ax.set_title('Contour Plot')
ax.set_xlabel('x (cm)')
ax.set_ylabel('y (cm)')
plt.show()
```



In [ ]: