Visulization

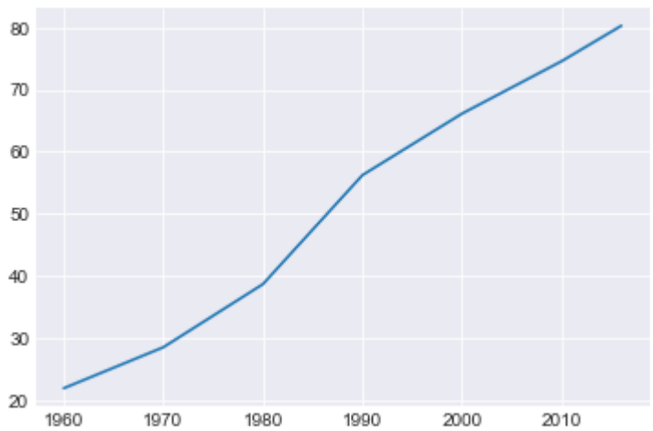
In [47]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [48]:
```
years = [1960, 1970, 1980, 1990, 2000, 2010, 2016]
iran_pop = [21.91, 28.51, 38.67, 56.23, 66.13, 74.57, 80.28]
```

Matplotlib

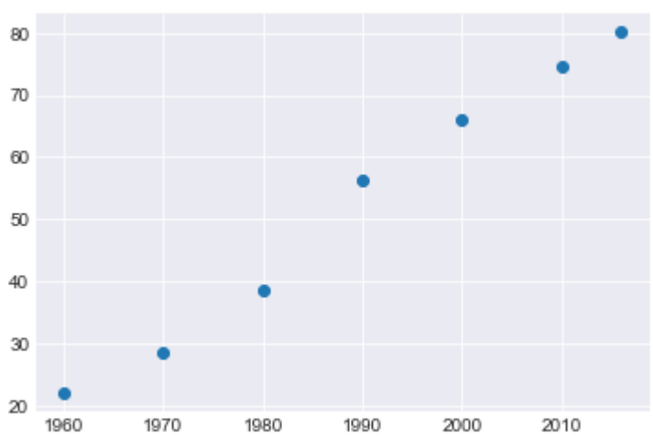
Line Plot

In [49]:
```
plt.plot(years, iran_pop)
plt.show()
```



Scatter Plot

In [50]:
```
plt.scatter(years, iran_pop)
plt.show()
```



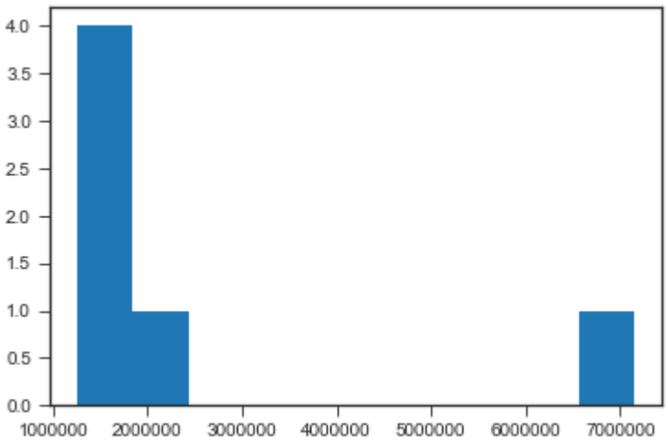
Histogram Plot

Histogram Help to Get Idea About Distribution

In [51]:
```
city_name = ['Tehran', 'Mashhad', 'Isfahan', 'Karaj', 'Tabriz', 'Shiraz']
city_pop = [7153309, 2307177, 1547164, 1448075, 1424641, 1249942]
```

Histogram Plot

In [70]:
```
plt.hist(city_pop)
plt.show()
```

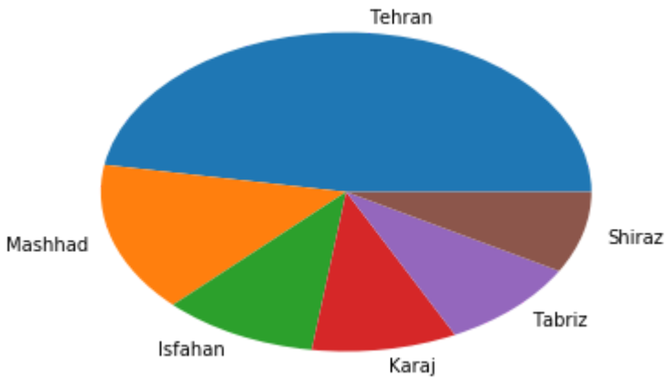


> The "square root rule" is a commonly-used rule of thumb for choosing number of bins

The bins parameter tells you the number of bins that your data will be divided into. set bins=1 to understand it

Pie Chart

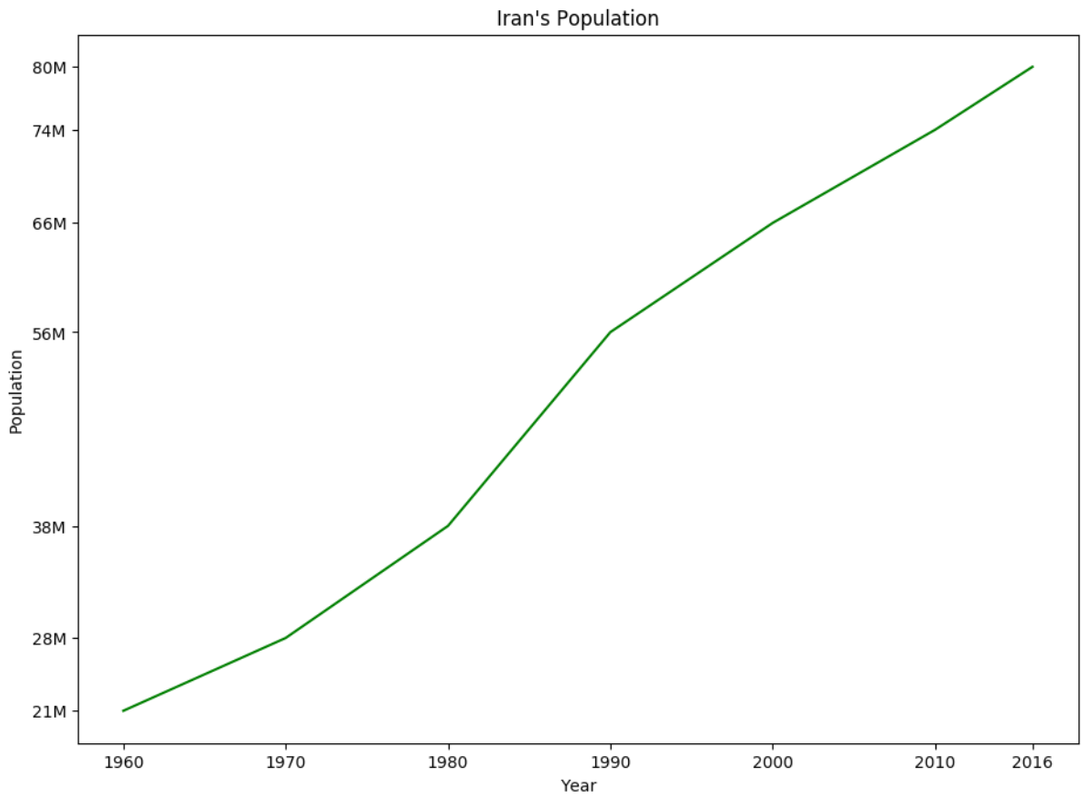
In [31]:
```
plt.pie(city_pop,labels=['Tehran', 'Mashhad', 'Isfahan', 'Karaj', 'Tabriz', 'Shiraz'] )
plt.show()
```



Customization plots

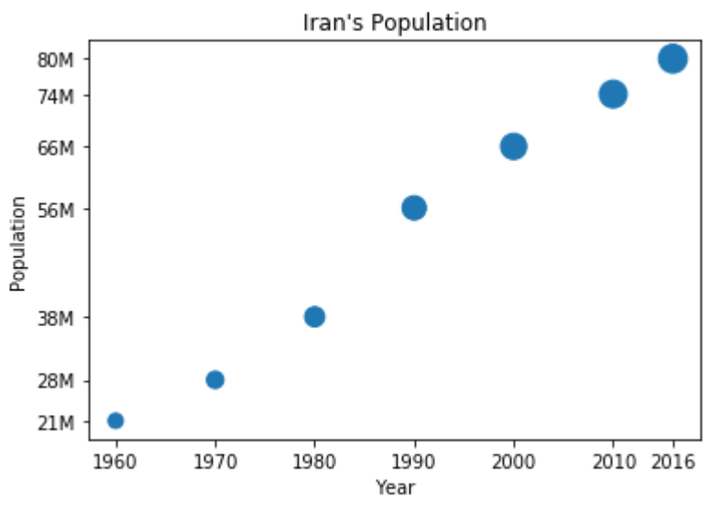
Customize Line Plot

In [67]:
```
plt.figure(figsize=(11, 8), dpi=100)
plt.plot(years, iran_pop, color='green')
plt.xlabel('Year')
plt.ylabel('Population')

plt.title('Iran\'s Population')
plt.yticks([21.91, 28.51, 38.67, 56.23, 66.13, 74.57, 80.28],['21M','28M','38M','56M',
'66M','74M','80M'])
plt.xticks([1960, 1970, 1980, 1990, 2000, 2010, 2016] )
plt.show()
```

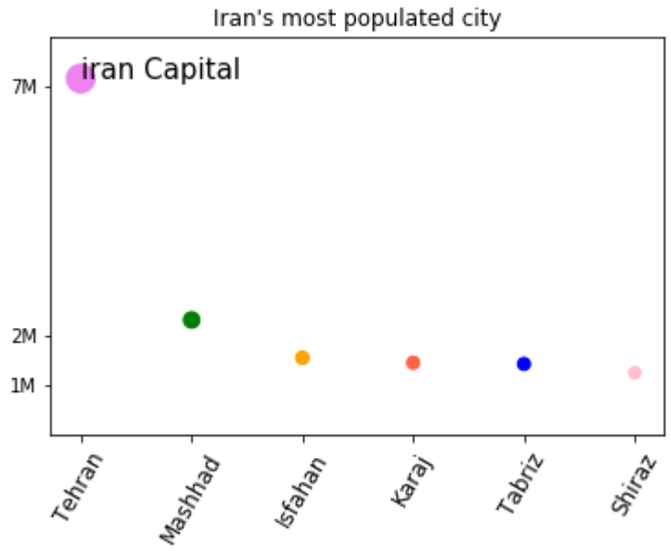


Customize Scatter plot

In [38]:
```
popsize = np.array(iran_pop)*3
plt.scatter(years, iran_pop, s=popsize)

plt.xlabel('Year')
plt.ylabel('Population')

plt.title('Iran\'s Population')
plt.yticks([21.91, 28.51, 38.67, 56.23, 66.13, 74.57, 80.28],['21M','28M','38M','56M',
'66M','74M','80M'])
plt.xticks([1960, 1970, 1980, 1990, 2000, 2010, 2016] )

plt.show()
```

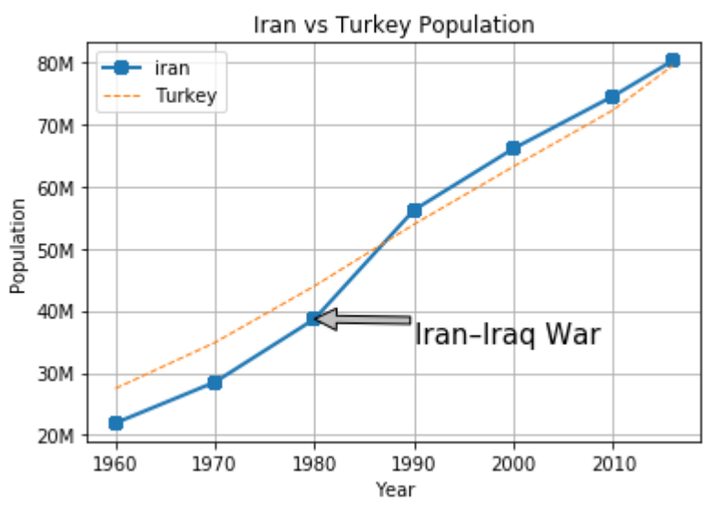


In [39]:
```
pop_size = np.array([7153309, 2307177, 1547164, 1448075, 1424641, 1249942])/30000
Colors = ['violet', 'green', 'orange', 'tomato', 'blue', 'pink']

plt.scatter(np.arange(6),city_pop, s=pop_size,c=Colors, alpha=1)
plt.xticks([0, 1, 2, 3, 4, 5],['Tehran', 'Mashhad', 'Isfahan', 'Karaj', 'Tabriz', 'Shir
az'],rotation=60,fontsize='large')
plt.yticks([1000000, 2000000, 7000000],['1M','2M','7M'])
plt.title('Iran\'s most populated city')

plt.text(0,7153309,'iran Capital',fontsize=15)
plt.ylim([0,8000000])

plt.show()
```



Multiple Plot

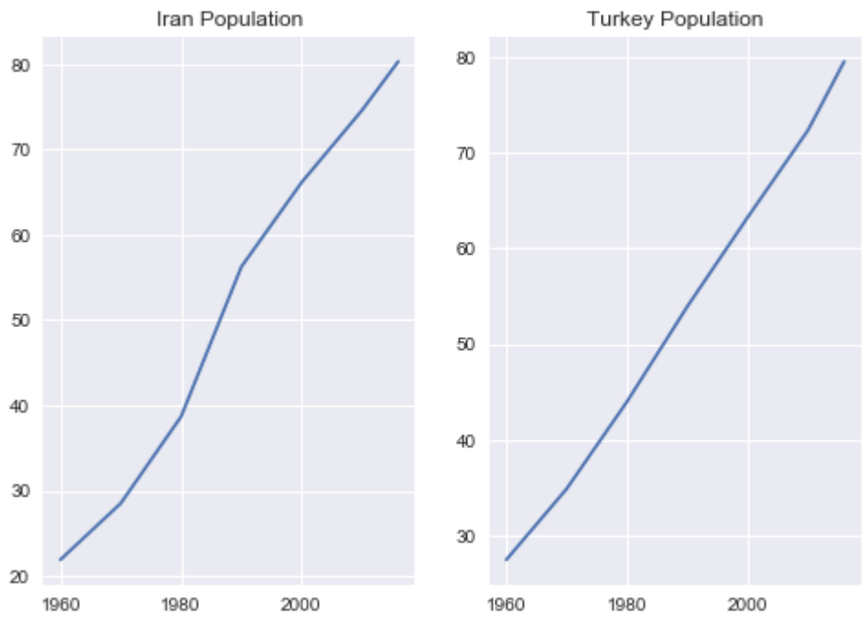
In [40]:
```
plt.plot(years, iran_pop,ls='-',lw=2,marker='+',mew=8)
aplt.plot(years,turkey_pop,ls='--',lw=1)
plt.title('Iran vs Turkey Population')
plt.xlabel('Year')
plt.ylabel('Population')
plt.yticks([20,30,40,50,60,70,80],['20M', '30M', '40M', '50M' ,'60M', '70M', '80M',])
plt.legend(['iran','Turkey'],loc='best')
#plt.text(1980,40,'Iran-Iraq War',fontsize=15)
plt.grid()

plt.annotate('Iran-Iraq War',xy=(1980,38.67) ,xytext=(1990,34.88) ,arrowprops=dict(face
color='silver',width=4),fontsize=15)
plt.show()
```



Subplot

subplot.png

In [94]:
```
plt.subplot(1,2,1)
plt.plot(years, iran_pop)
plt.title('Iran Population')

plt.subplot(1,2,2)
plt.plot(years,turkey_pop)
plt.title('Turkey Population')

plt.show()
```



Saving a plot

In [95]:
```
plt.savefig('myplot.jpg')
%pwd
```

Out[95]:
```
'C:\\Users\\rad'
```

Seaborn

what is seaborn

based top of matplotlib

work with Pandas Data Structures

Install Seaborn

In [75]:
```
! pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\rad\anaconda3\lib\site-packages
```

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [4]:
```
import pandas as pd
smartphones = pd.read_csv('c://smartphones.csv')
smartphones
```

Out[4]:

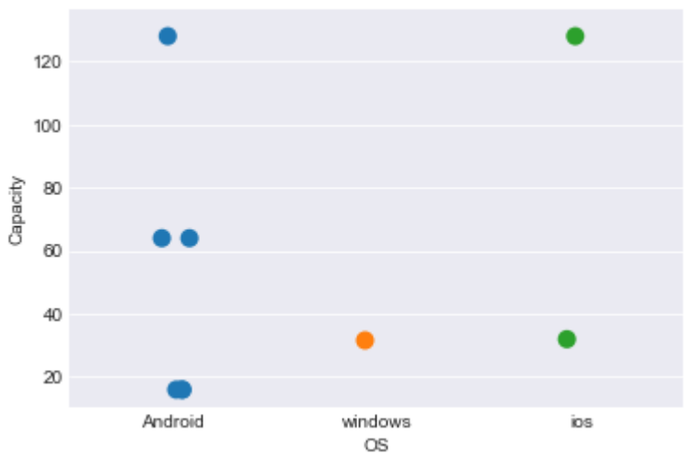
	Name	OS	Capacity	Ram	Weight	Company	inch
0	Galaxy S8	Android	64	4	149.0	Samsung	5.8
1	Lumia 950	windows	32	3	150.0	Microsoft	5.2
2	Xpreia L1	Android	16	2	180.0	Sony	5.5
3	iphone 7	ios	128	2	138.0	Apple	4.7
4	U Ultra	Android	64	4	170.0	HTC	5.7
5	Galaxy S5	Android	16	2	145.0	Samsung	5.1
6	iphone 5s	ios	32	1	112.0	Apple	4.0
7	Moto G5	Android	16	3	144.5	Motorola	5.0
8	Pixel	Android	128	4	143.0	Google	5.0

Strip Plot

> plots the distribution of variables for each category as individual datapoints

In [14]:
```
print(smartphones)
sb.stripplot(x='OS',y='Capacity',data=smartphones,size=10, jitter=True)
#sb.swarmplot(x='OS', y='Capacity', data=smartphones)

plt.show()
```

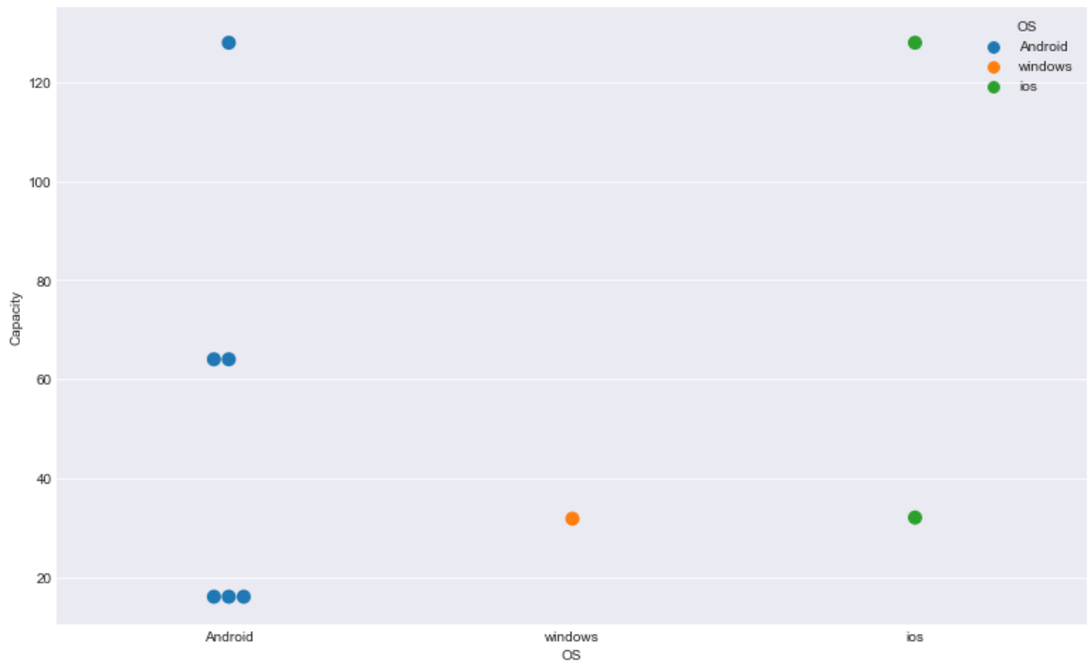
```
       Name       OS  Capacity  Ram  Weight    Company  inch
0  Galaxy S8  Android        64    4   149.0    Samsung   5.8
1  Lumia 950  windows        32    3   150.0  Microsoft   5.2
2  Xpreia L1  Android        16    2   180.0       Sony   5.5
3   iphone 7      ios       128    2   138.0      Apple   4.7
4   U Ultra   Android        64    4   170.0        HTC   5.7
5  Galaxy S5  Android        16    2   145.0    Samsung   5.1
6  iphone 5s      ios        32    1   112.0      Apple   4.0
7    Moto G5  Android        16    3   144.5   Motorola   5.0
8      Pixel  Android       128    4   143.0     Google   5.0
```



Swarm Plot

> This function is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don’t overlap

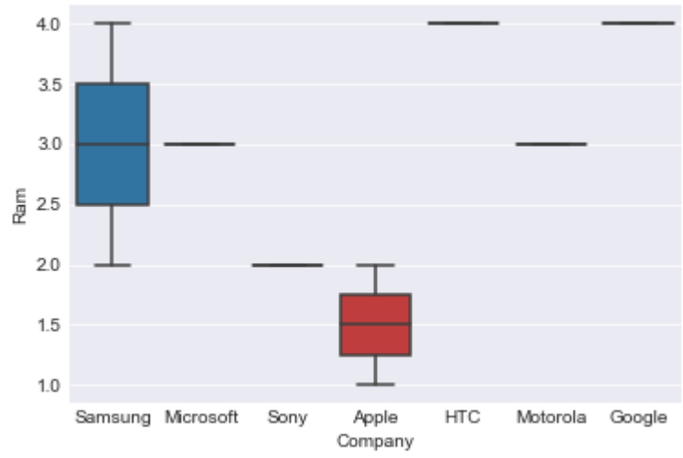
In [45]:
```
plt.figure(figsize=(13,8))
sb.swarmplot(x='OS', y='Capacity', data=smartphones, hue='OS', size=10)
plt.show()
```



Box Plot

boxplot.png

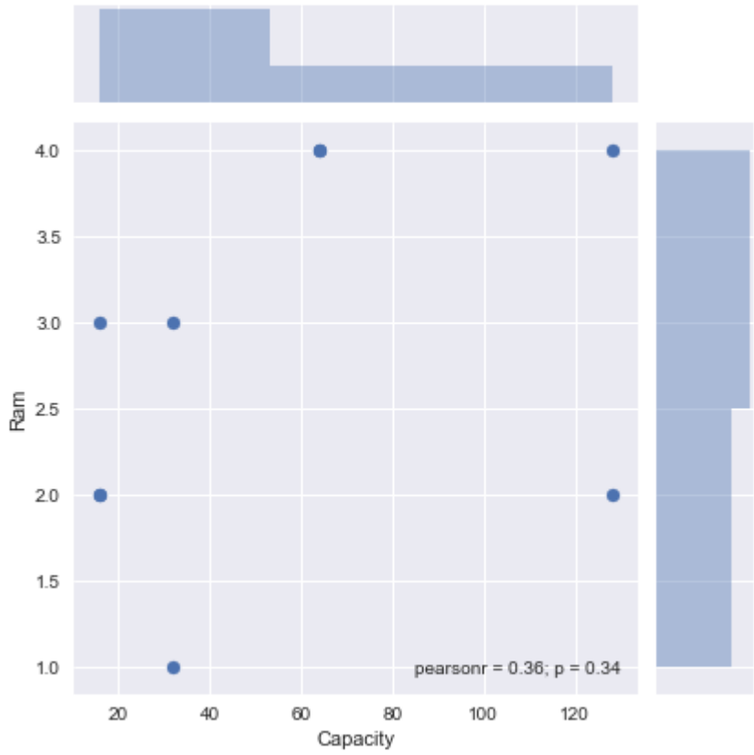
In [28]:
```
sb.boxplot(x='Company', y='Ram', data=smartphones)
plt.show()
```



Joint Plot

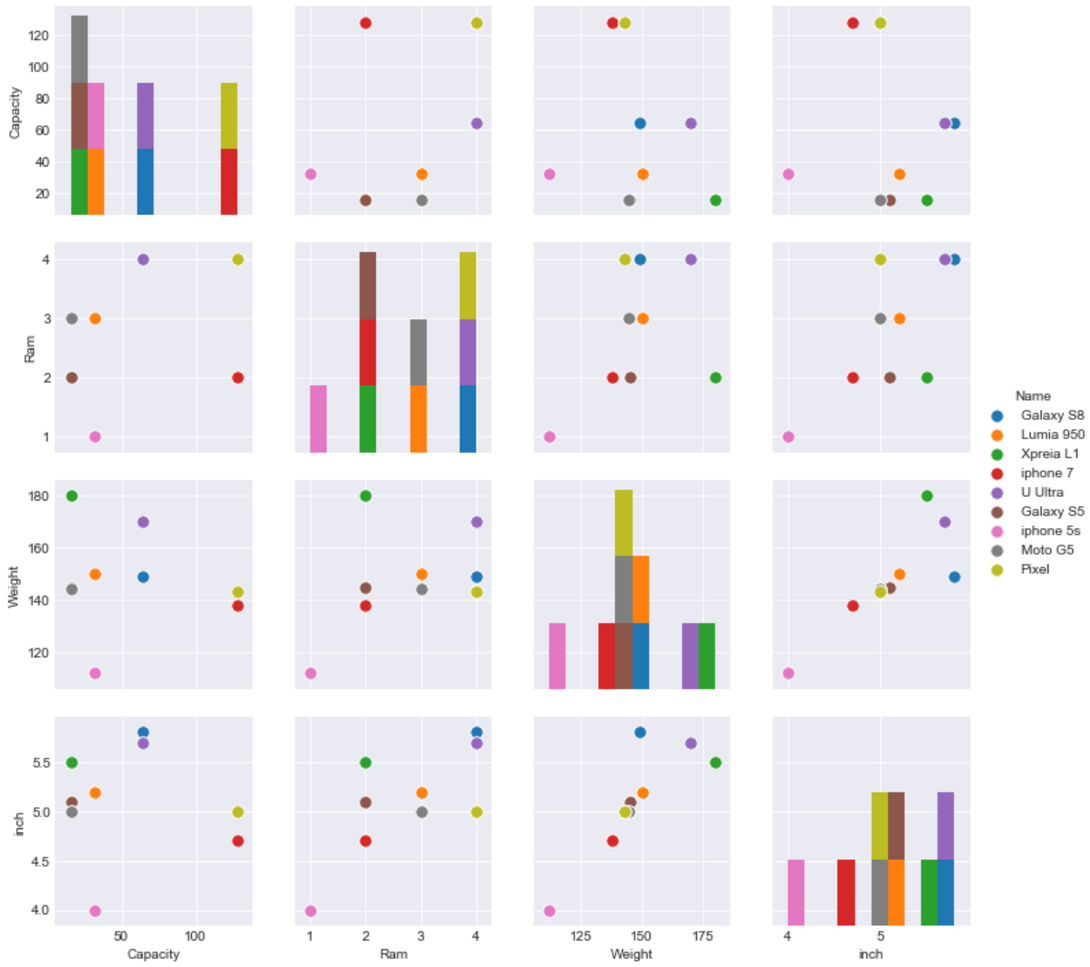
visualize a bivariate distribution

In [175]:
```
sb.jointplot(x='Capacity', y='Ram', data=smartphones,kind='scatter')
plt.show()
```



Pair Plot

In [13]:
```
import seaborn as sb
sb.set_style('darkgrid')
smartphones = pd.read_csv('c://smartphones.csv')
sb.pairplot(smartphones, hue='Name', plot_kws={'s':90})
plt.show()
```



http://seaborn.pydata.org/tutorial.html# (http://seaborn.pydata.org/tutorial.html#)