In [3]:
```python
#Innovation In Education By Oybek, Python Project leader


print("innovation in education across USA, China, and Germany: world need ideas and innovation to deal progessively w

#literacy rates across USA, China, and Germany
print(' the following data frames and graphs below show how the literacy rates vary between USA,China and Germany')
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


dataframe = pd.read_csv('cross-country-literacy-rates two.csv')

dataframe

dataframe.sort_values('rates_of_Literacy', ascending=False)

import re

dataframe.loc[dataframe['COUNTRY_Code'].str.contains('United States USA', flags=re.I, regex=True)]
```

```
innovation in education across USA, China, and Germany: world need ideas and innovation to deal progessively with the
major problems the humanity face.Countries should create opportunities and conditions to cultivate such creative and
talented people who contribute to this important work. let us look at the human resources'quality of those three coun
tries
 the following data frames and graphs below show how the literacy rates vary between USA,China and Germany
```

Out[3]:

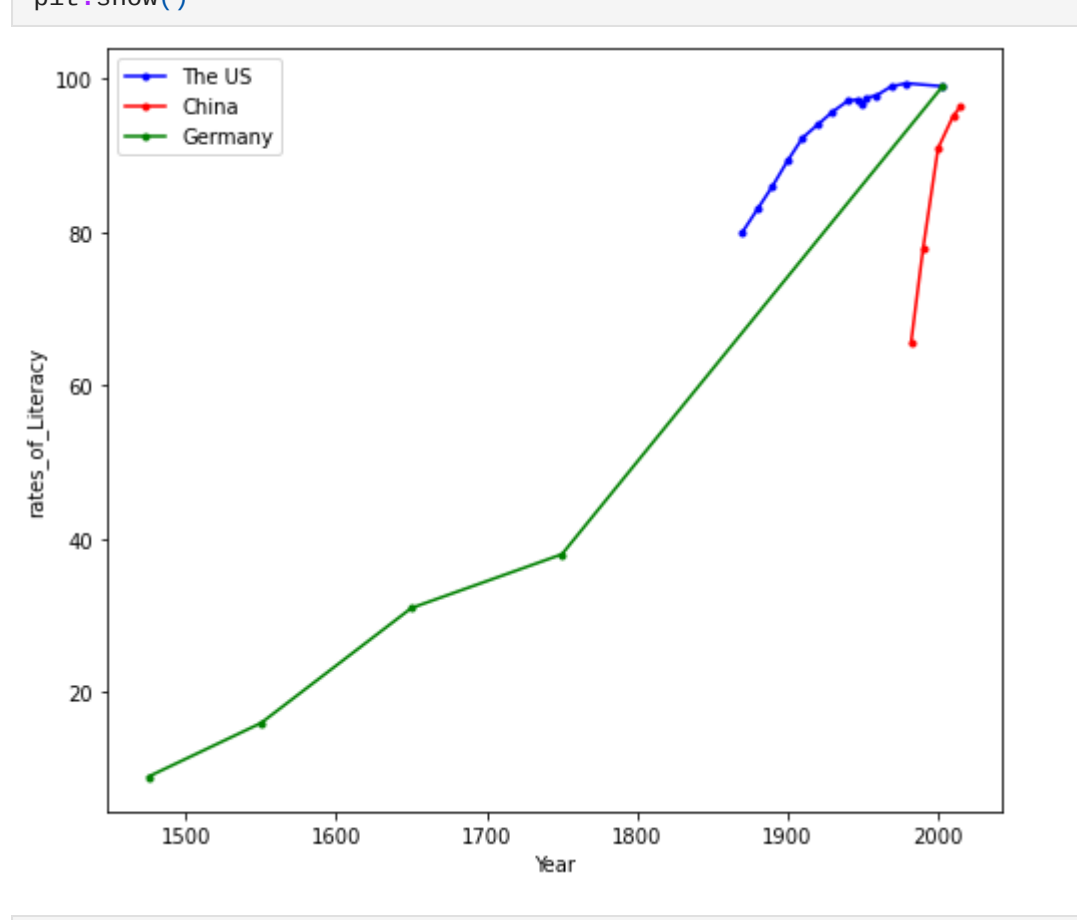
	COUNTRY_Code	Year	rates_of_Literacy
10	United States USA	1870	80.000000
11	United States USA	1880	83.000000
12	United States USA	1890	86.000000
13	United States USA	1900	89.300003
14	United States USA	1910	92.300003
15	United States USA	1920	94.000000
16	United States USA	1930	95.699997
17	United States USA	1940	97.099998
18	United States USA	1947	97.300003
19	United States USA	1950	96.800003
20	United States USA	1952	97.500000
21	United States USA	1959	97.800003
22	United States USA	1969	99.000000
23	United States USA	1979	99.400002
24	United States USA	2003	99.000000

In [4]:
```python
# let us Compare Literacy rates across the US, China and Germany
US = dataframe[ dataframe.COUNTRY_Code == 'United States USA' ]
China = dataframe[ dataframe.COUNTRY_Code == 'China CHN' ]
Germany = dataframe[ dataframe.COUNTRY_Code == 'Germany DEU' ]
Germany
```

Out[4]:

	COUNTRY_Code	Year	rates_of_Literacy
0	Germany DEU	1475	9.0
1	Germany DEU	1550	16.0
2	Germany DEU	1650	31.0
3	Germany DEU	1750	38.0
4	Germany DEU	2003	99.0

In [5]:
```python
plt.figure(figsize=(8,7))
plt.plot(US.Year, US.rates_of_Literacy,'b.-')
plt.plot(China.Year, China.rates_of_Literacy,'r.-')
plt.plot(Germany.Year, Germany.rates_of_Literacy,'g.-')
plt.legend(['The US', 'China', 'Germany'])
plt.xlabel('Year')
plt.ylabel('rates_of_Literacy')

plt.savefig('rates_of_Literacy_figure.png', dpi= 300)
plt.show()
```



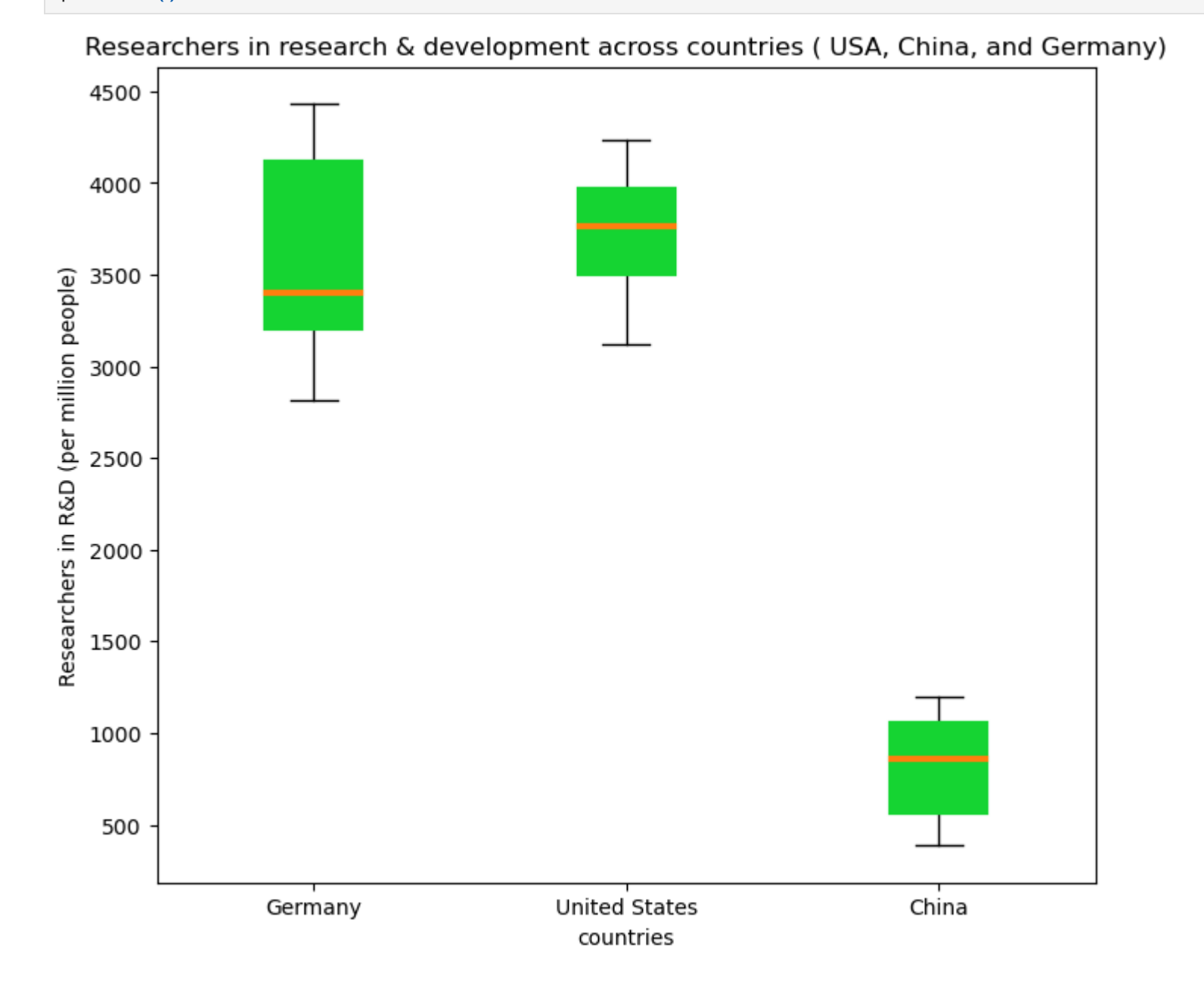
In [6]:
```python
#as we can see from these graphs, US was leader for gaining higher literacy rates within the past decades
```

In [7]:
```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
researchers = pd.read_csv('researchers-in-rd-per-million-peopleII.csv')

researchers

researchers.sort_values('Researchers_in_R_and_D_(per_million_people)', ascending=False)
import re

researchers.loc[researchers['country'].str.contains('United States', flags=re.I, regex=True)]
```

Out[7]:

	country	Code	Year	Researchers_in_R_and_D_(per_million_people)
40	United States	USA	1996	3122.57281
41	United States	USA	1997	3224.23913
42	United States	USA	1998	3388.20888
43	United States	USA	1999	3445.30961
44	United States	USA	2000	3475.69505
45	United States	USA	2001	3545.83215
46	United States	USA	2002	3630.61832
47	United States	USA	2003	3870.56966
48	United States	USA	2004	3765.10261
49	United States	USA	2005	3718.19505
50	United States	USA	2006	3781.80788
51	United States	USA	2007	3757.86688
52	United States	USA	2008	3911.53450
53	United States	USA	2009	4073.17633
54	United States	USA	2010	3868.56644
55	United States	USA	2011	4011.32862
56	United States	USA	2012	4015.88708
57	United States	USA	2013	4117.67409
58	United States	USA	2014	4231.98928

In [8]:
```python
plt.style.use('default')

plt.figure(figsize=(8,7))
Germany = researchers.loc[researchers.country =='Germany']['Researchers_in_R_and_D_(per_million_people)']
United_States = researchers.loc[researchers.country == 'United States']['Researchers_in_R_and_D_(per_million_people)'
China = researchers.loc[researchers.country == 'China']['Researchers_in_R_and_D_(per_million_people)']

labels = ['Germany','United States','China']
boxes = plt.boxplot([Germany, United_States,China],labels= labels,patch_artist=True, medianprops={'linewidth':3})
for box in boxes['boxes']:
    box.set(color='#15d432',linewidth =3)
plt.title('Researchers in research & development across countries ( USA, China, and Germany)')
plt.ylabel('Researchers in R&D (per million people)')
plt.xlabel('countries')

plt.savefig('Researchers in research & development across countries ( USA, China, and Germany).png', dpi= 300)
plt.show()
```



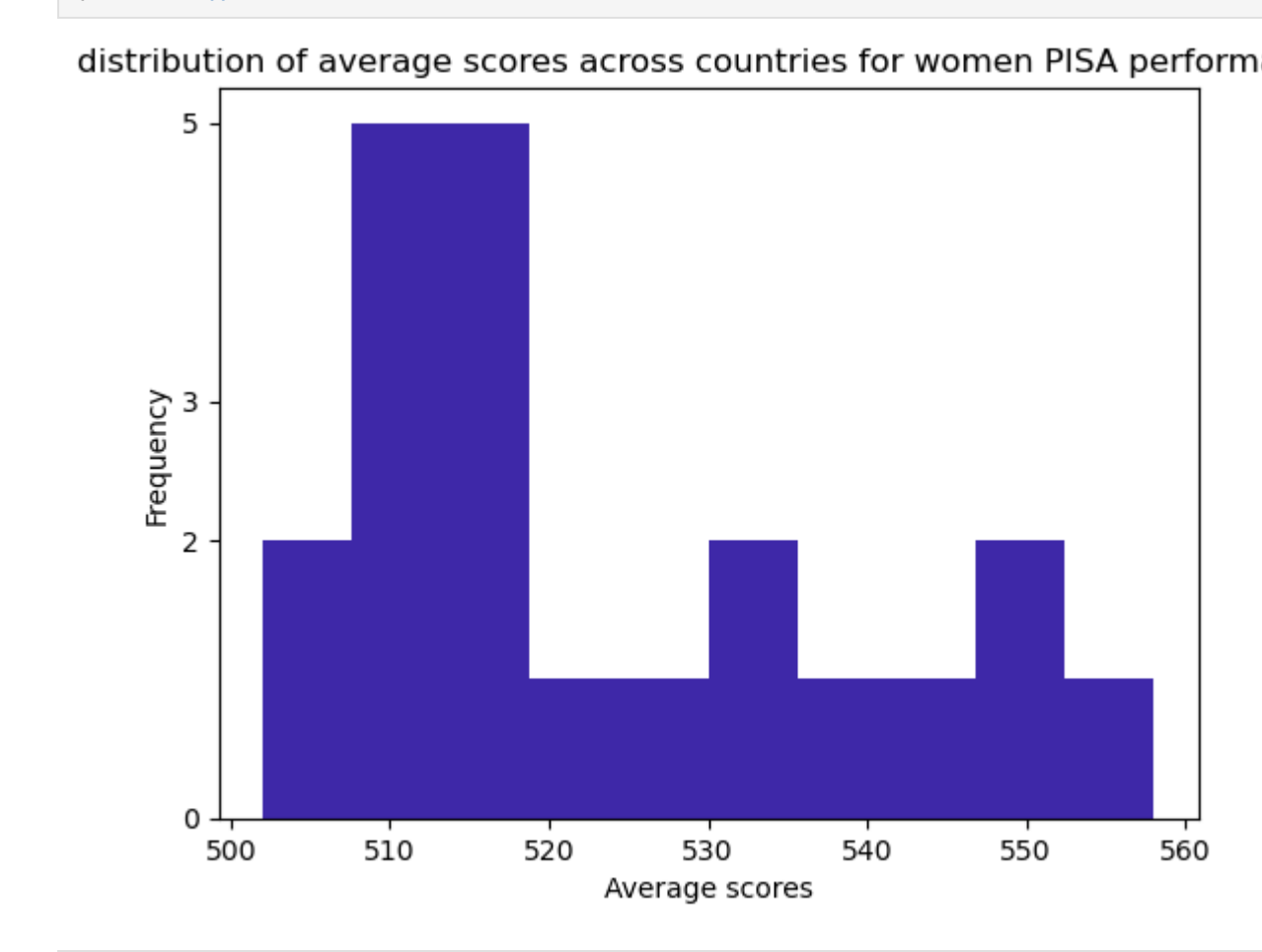
In [9]:
```python
#The OECD Programme for International Student Assessment (PISA) examines what students know in reading, mathematics a
#It provides the most comprehensive and rigorous international assessment of student learning outcomes to date.
#Results from PISA indicate the quality and equity of learning outcomes attained around the world, and allow educator

#NOTE: The Reading, Mathematics and Science scale ranges from 0 to 1000. Some apparent differences between estimates
#SOURCE: Organization for Economic Cooperation and Development (OECD), Program for International Student Assessment (

print("women vs men PISA reading performances across countries")

PISA = pd.read_csv('Averages for age 15 years PISA reading scale.csv')

PISA
```

```
women vs men PISA reading performances across countries
```

Out[9]:

	Year	Country	average	Standard_Error
0	2018	Germany	512	(3.2)
1	2018	United States	517	(3.6)
2	2018	Hong Kong (China)	542	(2.8)
3	2015	Germany	520	(3.1)
4	2015	United States	507	(3.9)
5	2015	Hong Kong (China)	541	(3.6)
6	2012	Germany	530	(3.1)
7	2012	United States	513	(3.8)
8	2012	Hong Kong (China)	558	(3.3)
9	2009	Germany	518	(2.9)
10	2009	United States	513	(3.8)
11	2009	Hong Kong (China)	550	(2.8)
12	2006	Germany	517	(4.4)
13	2006	United States	515	(3.8)
14	2006	Hong Kong (China)	551	(3.0)
15	2003	Germany	513	(3.9)
16	2003	United States	511	(3.5)
17	2003	Hong Kong (China)	525	(3.5)
18	2000	Germany	502	(3.9)
19	2000	United States	518	(6.2)
20	2000	Hong Kong (China)	533	(3.6)

In [10]:
```python
plt.hist(PISA.average,color='#3e28a8')
plt.yticks([0,2,3,5])
plt.xlabel('Average scores')
plt.ylabel('Frequency')
plt.title('distribution of average scores across countries for women PISA performance')
plt.savefig('distribution of average scores across countries for women PISA performance.png', dpi= 300)
plt.show()
```

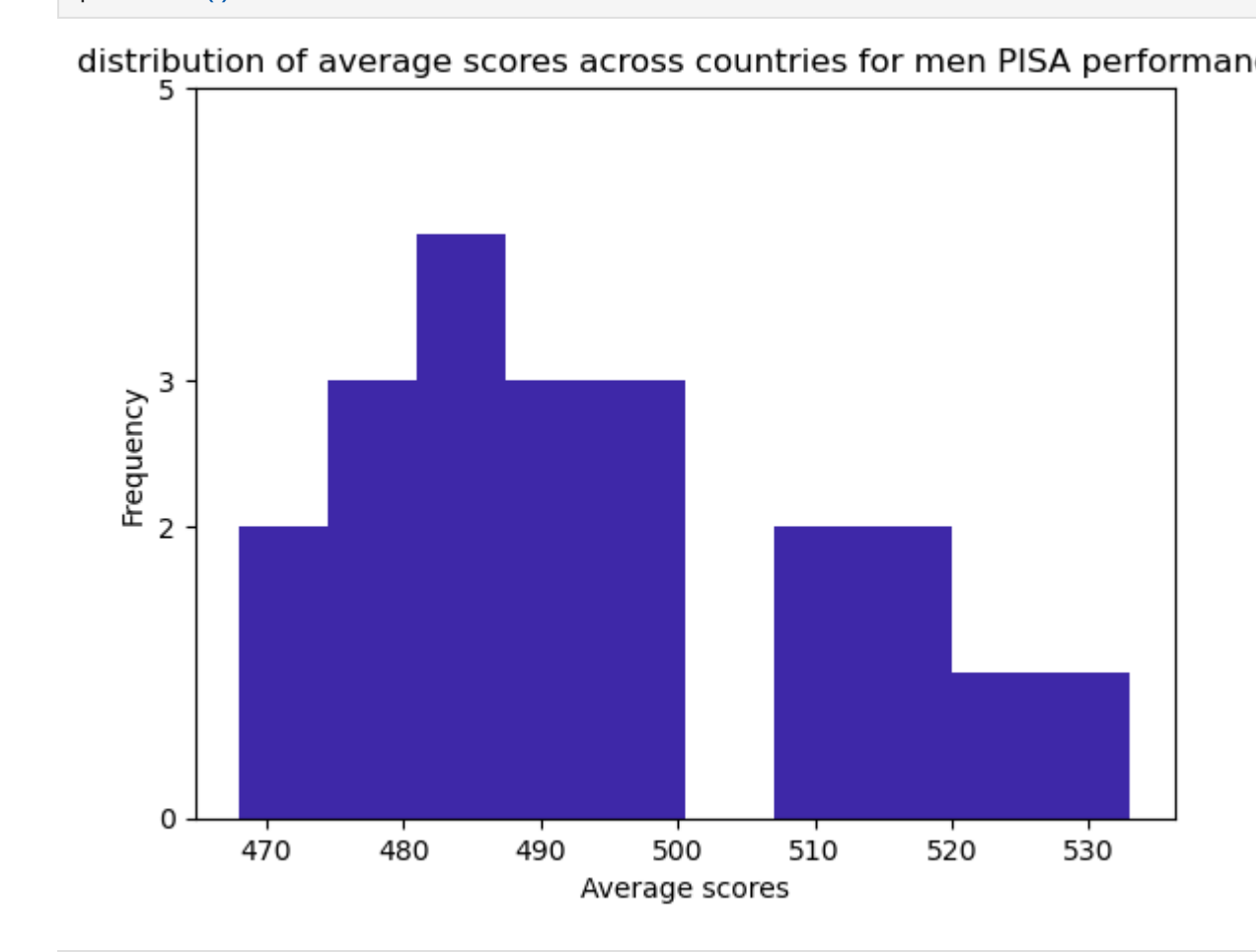


In [11]:
```python
PISAmen = pd.read_csv('Averages for age 15 years PISA reading scale ( male).csv')
PISAmen

PISAmen.sort_values(('average'), ascending=False)
```

Out[11]:

	Year	Country	average	Standard Error
8	2012	Hong Kong (China)	533	(3.8)
14	2006	Hong Kong (China)	520	(3.5)
20	2000	Hong Kong (China)	518	(4.8)
11	2009	Hong Kong (China)	518	(3.3)
5	2015	Hong Kong (China)	513	(3.4)
2	2018	Hong Kong (China)	507	(3.5)
3	2015	Germany	499	(3.7)
17	2003	Hong Kong (China)	494	(5.3)
1	2018	United States	494	(4.2)
19	2000	United States	490	(8.4)
13	2006	United States	490	(4.1)
10	2009	United States	488	(4.2)
4	2015	United States	487	(3.7)
6	2012	Germany	486	(2.9)
0	2018	Germany	486	(3.4)
7	2012	United States	482	(4.1)
16	2003	United States	479	(3.7)
9	2009	Germany	478	(3.6)
12	2006	Germany	475	(5.3)
15	2003	Germany	471	(4.2)
18	2000	Germany	468	(3.2)

In [12]:
```python
plt.hist(PISAmen.average, color='#3e28a8')
plt.yticks([0,2,3,5])
plt.xlabel('Average scores')
plt.ylabel('Frequency')
plt.title('distribution of average scores across countries for men PISA performance')
plt.savefig('distribution of average scores across countries for men PISA performance.png', dpi= 300)
plt.show()
```



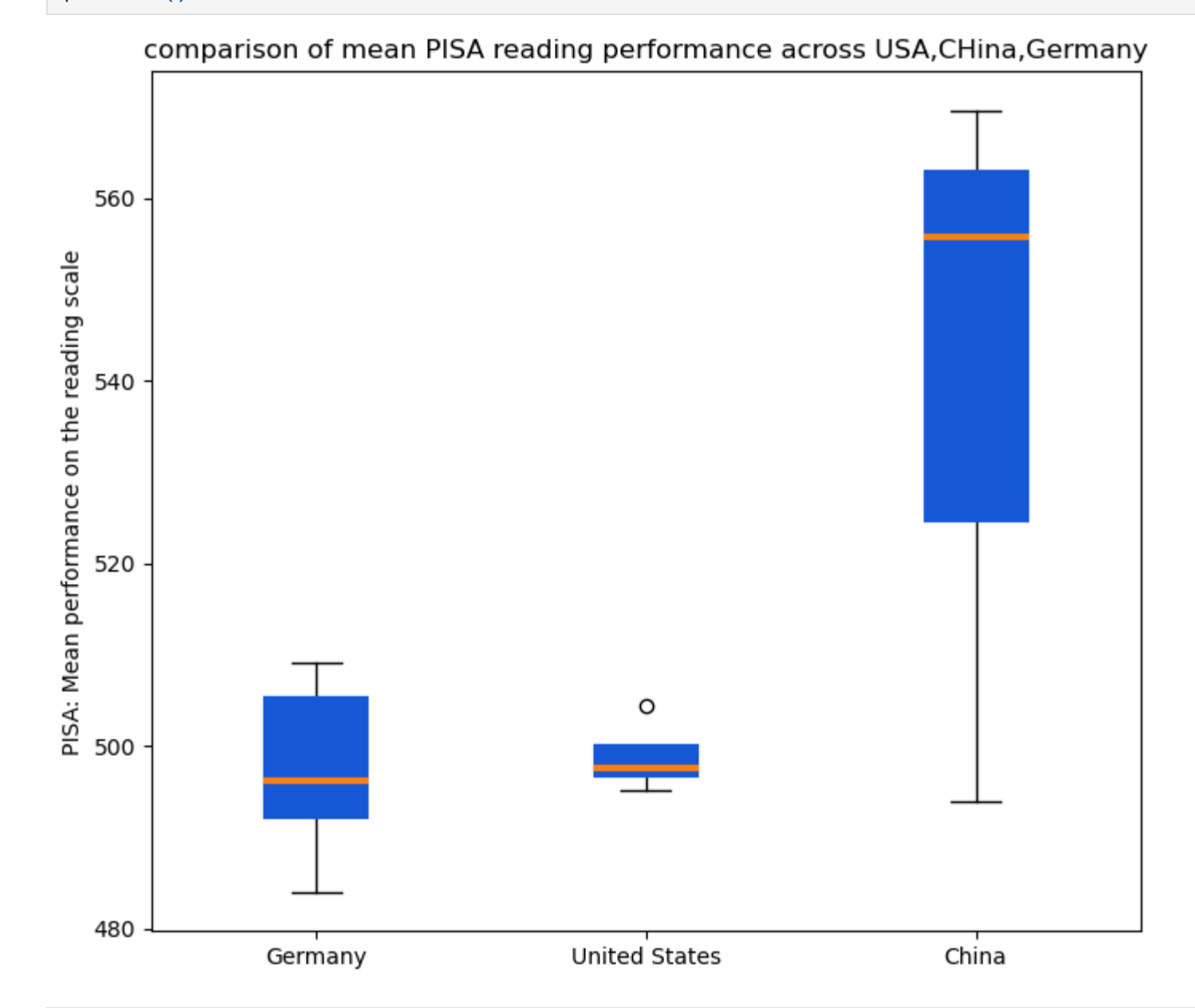
In [13]:
```python
# mean PISA performance in The science, math and reading across countries
print('mean reading performance across countries')
mean_PISA_reading = pd.read_csv('PISA mean reading performance.csv')
mean_PISA_reading

import re

mean_PISA_reading.loc[mean_PISA_reading['country'].str.contains('United States', flags=re.I, regex=True)]
```

```
mean reading performance across countries
```

Out[13]:

	country	Year	PISA: Mean performance on the reading scale
6	United States	2000	504.419691
7	United States	2003	495.182412
8	United States	2009	499.826821
9	United States	2012	497.581718
10	United States	2015	496.935100

In [14]:
```python
plt.style.use('default')

plt.figure(figsize=(8,7))
Germany = mean_PISA_reading.loc[mean_PISA_reading.country =='Germany']['PISA: Mean performance on the reading scale']
United_States = mean_PISA_reading.loc[mean_PISA_reading.country == 'United States']['PISA: Mean performance on the re
China = mean_PISA_reading.loc[mean_PISA_reading.country == 'China']['PISA: Mean performance on the reading scale']

labels = ['Germany','United States','China']
boxes = plt.boxplot([Germany, United_States,China],labels= labels,patch_artist=True, medianprops={'linewidth':3})
for box in boxes['boxes']:
    box.set(color='#1557d4',linewidth =3)
plt.title('comparison of mean PISA reading performance across USA,CHina,Germany')
plt.ylabel('PISA: Mean performance on the reading scale')
plt.savefig('comparison of mean PISA reading performance across USA,CHina,Germany.png', dpi= 300)

plt.show()
```



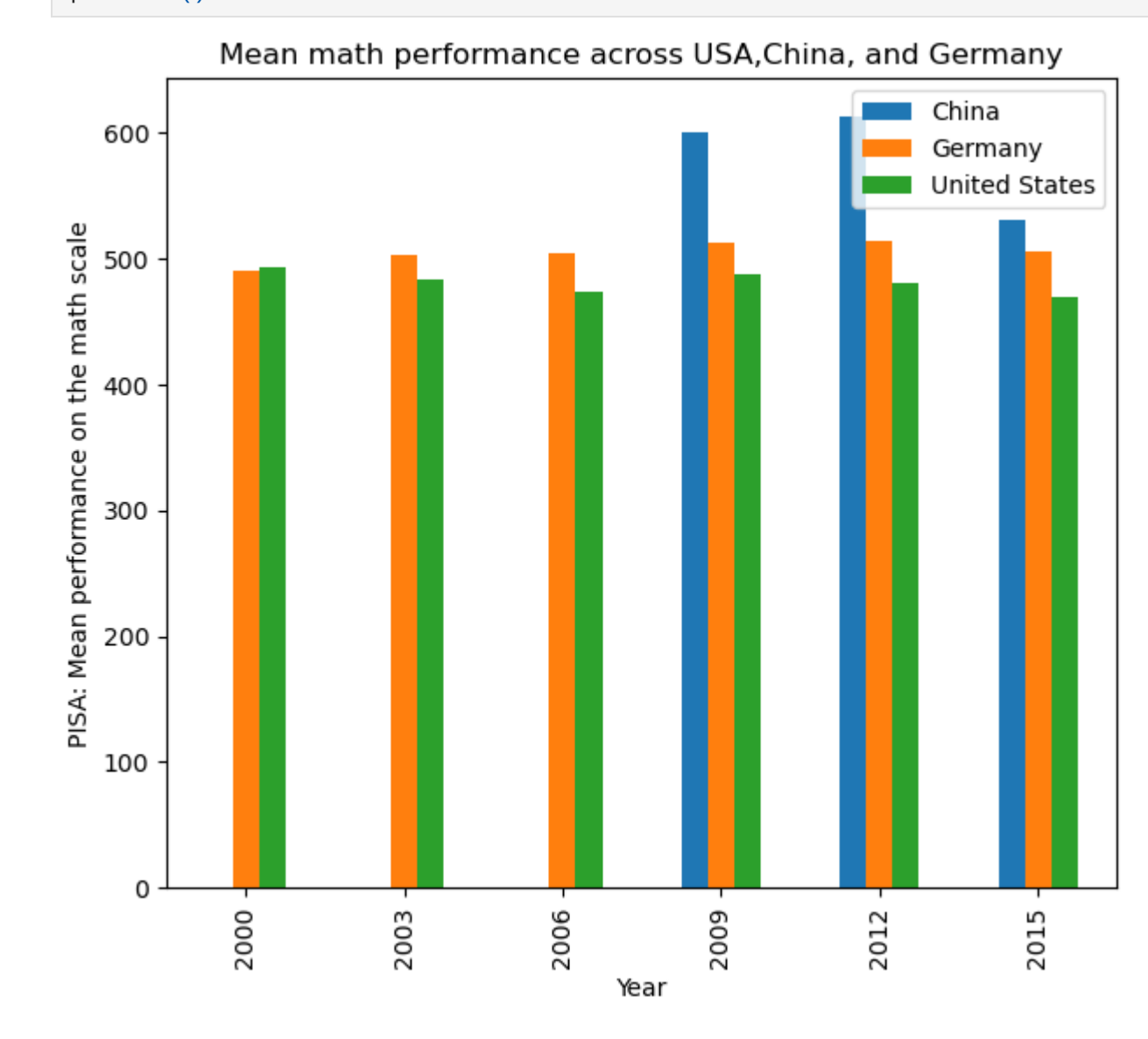
In [15]:
```python
print('mean math performance across countries')
mean_PISA_math = pd.read_csv('PISA mean math performance.csv')
mean_PISA_math
```

```
mean math performance across countries
```

Out[15]:

	country	Year	PISA: Mean performance on the mathematics scale
0	United States	2000	493.000000
1	United States	2003	482.885038
2	United States	2006	474.352158
3	United States	2009	487.396520
4	United States	2012	481.366786
5	United States	2015	469.628500
6	China	2009	600.076253
7	China	2012	612.675536
8	China	2015	531.296100
9	Germany	2000	490.000000
10	Germany	2003	502.985532
11	Germany	2006	503.790859
12	Germany	2009	512.777643
13	Germany	2012	513.525056
14	Germany	2015	505.971300

In [16]:
```python
mean_PISA_math.pivot(index='Year', columns ='country', values = 'PISA: Mean performance on the mathematics scale').pl
plt.xlabel('Year')
plt.ylabel('PISA: Mean performance on the math scale')
plt.title('Mean math performance across USA,China, and Germany')
plt.legend(loc='upper right')

plt.savefig('Mean math performance across USA,China, and Germany.png', dpi= 300)
plt.show()
```



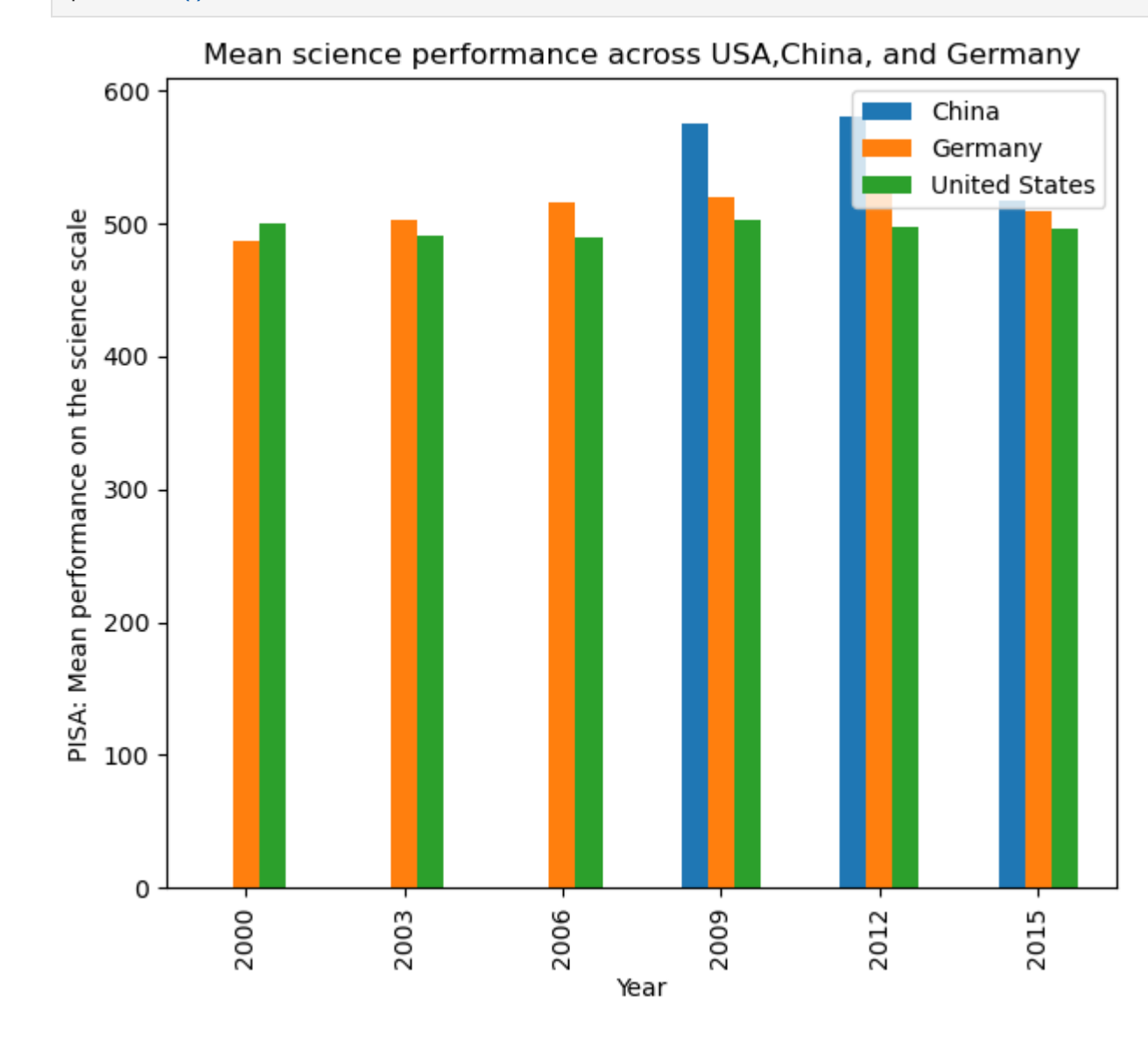
In [17]:
```python
print('mean science performance across countries')
mean_PISA_science = pd.read_csv('PISA mean science performance.csv')
mean_PISA_science
```

```
mean science performance across countries
```

Out[17]:

	country	Year	PISA: Mean performance on the science scale
0	United States	2000	499.460152
1	United States	2003	491.263463
2	United States	2006	488.906837
3	United States	2009	502.002265
4	United States	2012	497.409811
5	United States	2015	496.242400
6	China	2009	574.617355
7	China	2012	580.117831
8	China	2015	517.779300
9	Germany	2000	487.105836
10	Germany	2003	502.336476
11	Germany	2006	515.649130
12	Germany	2009	520.405349
13	Germany	2012	524.120799
14	Germany	2015	509.140600

In [18]:
```python
mean_PISA_science.pivot(index='Year', columns ='country', values = 'PISA: Mean performance on the science scale').plo
plt.xlabel('Year')
plt.ylabel('PISA: Mean performance on the science scale')
plt.title('Mean science performance across USA,China, and Germany')
plt.legend(loc='upper right')

plt.savefig('Mean science performance across USA,China, and Germany.png', dpi= 300)
plt.show()
```

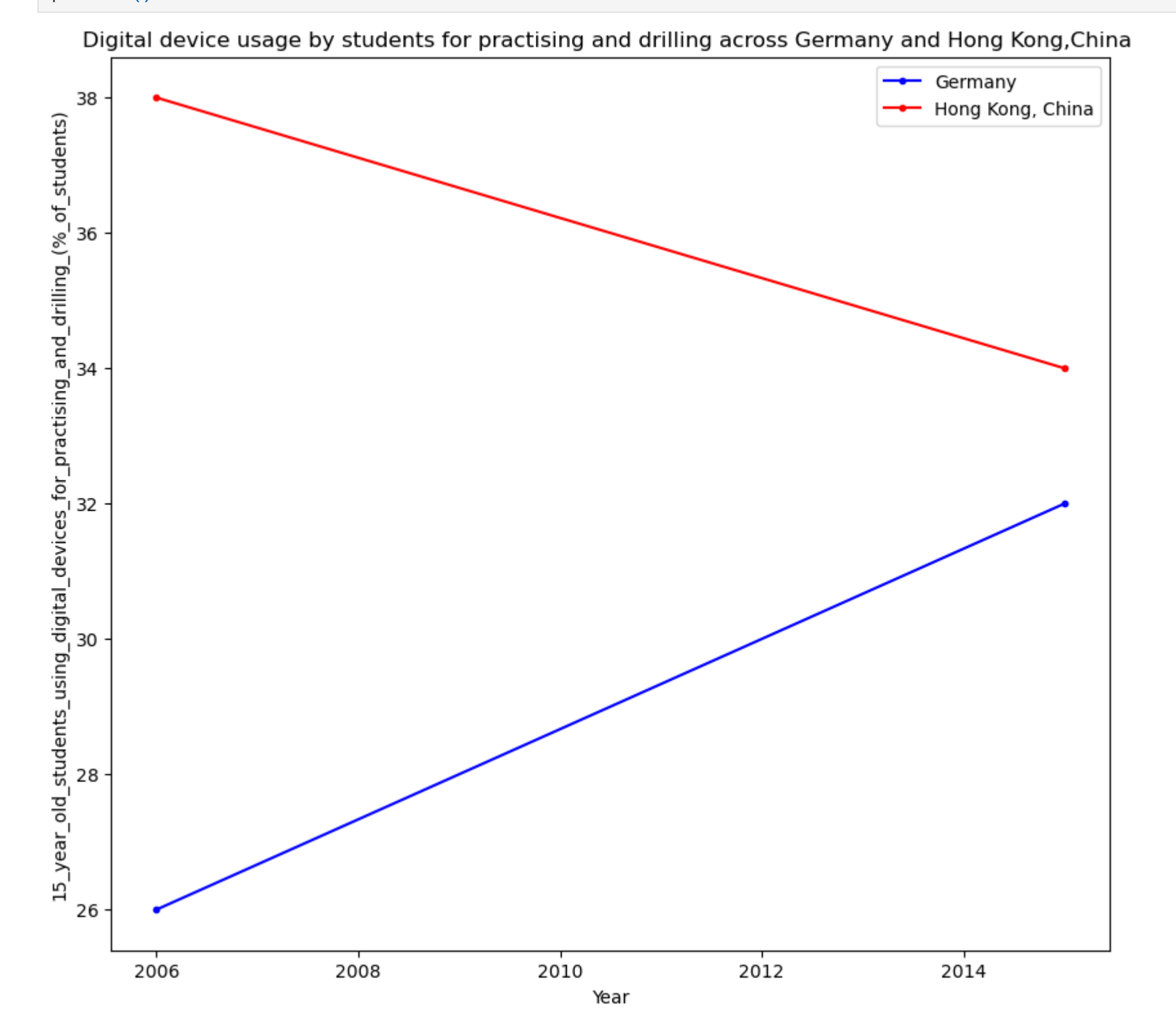


In [19]:
```python
Digital_device_usage = pd.read_csv('using digital devices.csv')
Digital_device_usage
```

Out[19]:

	Country	Year	15_year_old_students_using_digital_devices_for_practising_and_drilling_(%_of_students)
0	Germany	2006	26
1	Germany	2015	32
2	Hong Kong China	2006	38
3	Hong Kong China	2015	34

In [20]:
```python
Germany = Digital_device_usage[ Digital_device_usage.Country == 'Germany' ]
Hong_Kong_China = Digital_device_usage[ Digital_device_usage.Country == 'Hong Kong China' ]

Germany_students= Digital_device_usage.loc[Digital_device_usage.Country =='Germany']['15_year_old_students_using_digi
Hong_Kong_China_students= Digital_device_usage.loc[Digital_device_usage.Country =='Hong Kong China']['15_year_old_stu
plt.figure(figsize=(10,9))
plt.plot(Germany.Year,Germany_students,'b.-')
plt.plot(Hong_Kong_China.Year,Hong_Kong_China_students,'r.-')

plt.title('Digital device usage by students for practising and drilling across Germany and Hong Kong,China ')
plt.legend(['Germany', 'Hong Kong, China', ])
plt.xlabel('Year')
plt.ylabel('15_year_old_students_using_digital_devices_for_practising_and_drilling_(%_of_students)')
plt.savefig('Digital device usage by students for practising and drilling across Germany and Hong Kong,China.png', dp
plt.show()
```



In [ ]:

In [ ]:

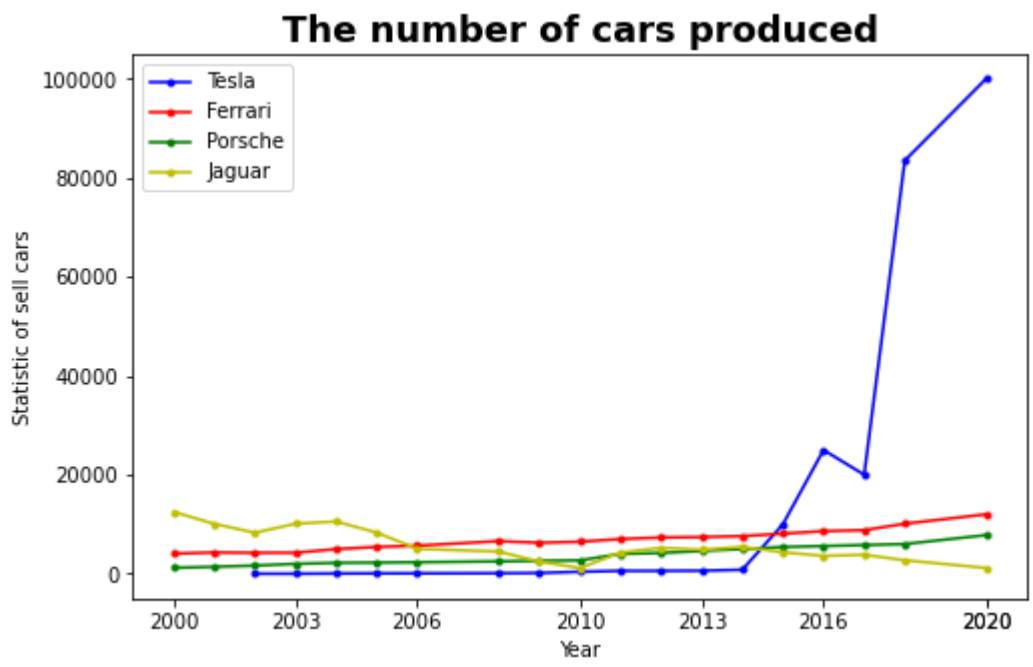
In [ ]:

In [1]:
```
# this part done by Jahongir.U

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

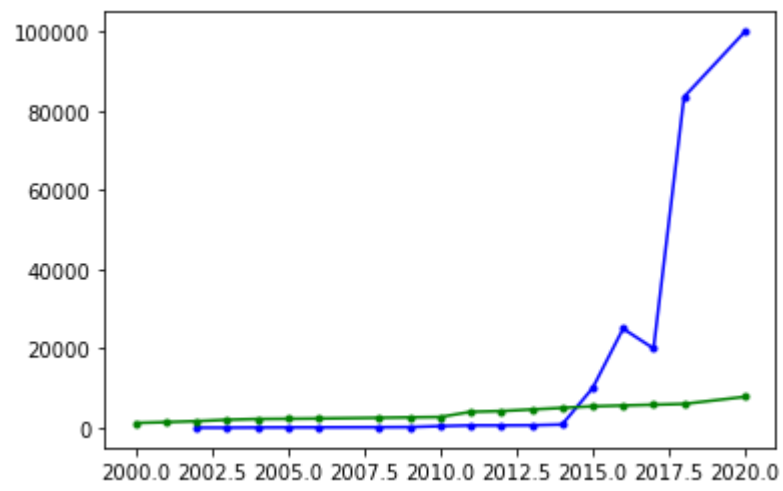
In [10]:
```
car = pd.read_csv(r'D:\Фотографии\statisti_of_car1.csv')

print(car)
```

```
    Year     Tesla  Ferrari  Porsche  Jaguar
0   2000       NaN     4070     1200   12492
1   2001       NaN     4289     1400   10082
2   2002       1.0     4236     1650    8286
3   2003       2.0     4238     2000   10102
4   2004      45.0     4975     2200   10552
5   2005      70.0     5409     2250    8304
6   2006      80.0     5671     2320    5027
7   2008     110.0     6587     2500    4474
8   2009     150.0     6250     2610    2452
9   2010     400.0     6461     2700    1161
10  2011     580.0     7001     4000    4278
11  2012     580.0     7318     4200    5235
12  2013     610.0     7415     4600    4852
13  2014     800.0     7600     5000    5434
14  2015   10000.0     8100     5400    4329
15  2016   25000.0     8600     5600    3611
16  2017   20000.0     8800     5800    3834
17  2018   83500.0    10100     6000    2721
18  2020  100000.0    12000     7800    1167
```

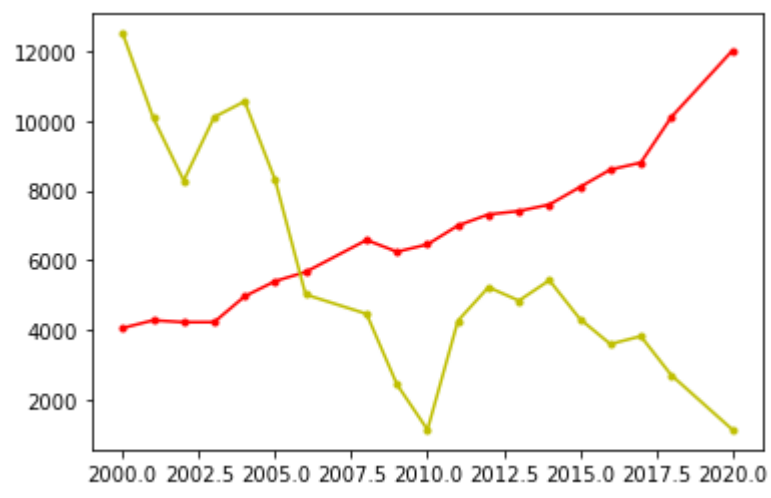
In [3]:
```
car = pd.read_csv(r'D:\Фотографии\statisti_of_car1.csv')

plt.figure(figsize=(8,5))

plt.title('The number of cars produced', fontdict={'fontweight':'bold', 'fontsize': 18})

plt.plot(car.Year, car.Tesla, 'b.-', label='Tesla')
plt.plot(car.Year, car.Ferrari, 'r.-', label = 'Ferrari')
plt.plot(car.Year, car['Porsche'],'g.-', label = 'Porsche')
plt.plot(car.Year, car.Jaguar,'y.-', label = 'Jaguar' )

plt.xticks(car.Year[::3].tolist()+[2020])

plt.xlabel('Year')
plt.ylabel('Statistic of sell cars')

plt.legend()

plt.savefig('statisti_of_car_figure.png', dpi=300)

plt.show()
```



In [4]:
```
car = pd.read_csv(r'D:\Фотографии\statisti_of_car1.csv')

plt.plot(car.Year, car.Tesla, 'b.-', label='Tesla')
plt.plot(car.Year, car['Porsche'],'g.-', label = 'Porsche')

plt.show()
```



In [5]:
```
car = pd.read_csv(r'D:\Фотографии\statisti_of_car1.csv')

plt.plot(car.Year, car.Ferrari, 'r.-', label='Tesla')
plt.plot(car.Year, car['Jaguar'],'y.-', label = 'Porsche')

plt.show()
```



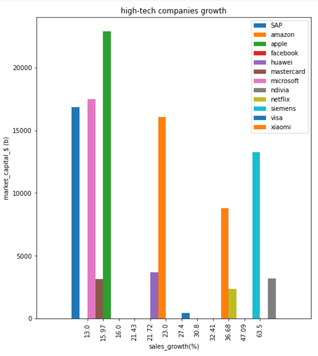
In [ ]:

In [4]:
```
# this part done by Xo'jamurodova Guljahon

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


df = pd.read_csv('high-tech companies.csv.')
df.sort_values('market_capital_dollar(b)', ascending=False)
df
```

Out[4]:

	company	sales_growth(%)	market_capital_dollar(b)
0	apple	21.43	22878
1	ndivia	63.50	3216
2	amazon	30.80	16087
3	microsoft	13.00	17495
4	facebook	47.09	7358
5	visa	21.72	426
6	mastercard	15.97	3153
7	netflix	32.41	2352
8	siemens	36.68	13265
9	SAP	16.00	16852
10	huawei	23.00	3698
11	xiaomi	27.40	8793

In [5]:
```
df.pivot(index='sales_growth(%)', columns ='company', values = 'market_capital_dollar(b)').plot(kind='bar', figsize=
plt.xlabel('sales_growth(%)')
plt.ylabel('market_capital_$ (b)')
plt.title('high-tech companies growth')
plt.legend(loc='upper right')

plt.savefig('Mean math performance across USA,China, and Germany.png', dpi= 300)
plt.show()
```

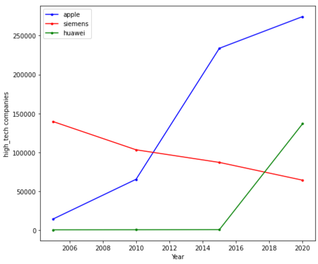


In [6]:
```
race = pd.read_csv('race.csv.')


race['total']= race['apple'] + race['siemens'] + race['huawei']
race
```

Out[6]:

	year	apple	siemens	huawei	total
0	2005	13931	139573	203	153707
1	2010	65225	103125	276	168626
2	2015	233715	86906	386	321007
3	2020	274515	64030	136717	475262

In [7]:
```
plt.figure(figsize=(8,7))
plt.plot(race.year, race.apple,'b.-')
plt.plot(race.year, race.siemens,'r.-')
plt.plot(race.year, race.huawei,'g.-')
plt.legend(['apple', 'siemens', 'huawei'])
plt.xlabel('Year')
plt.ylabel('high_tech companies')

plt.savefig('high_tech companies.png', dpi= 300)
plt.show()
```



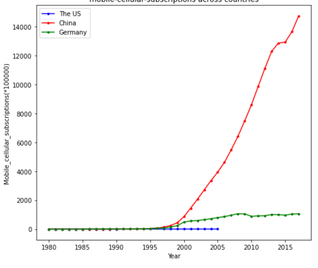
In [8]:
```
mobile_cellular_subscriptions = pd.read_csv('mobile-cellular-subscriptions.xls.csv')
mobile_cellular_subscriptions
```

Out[8]:

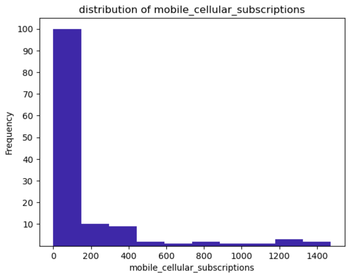
	Entity	Year	Mobile_cellular_subscriptions
0	China	1980	0
1	China	1981	0
2	China	1982	0
3	China	1983	0
4	China	1984	0
...	...	...	...
126	United States Virgin Islands	2001	41000
127	United States Virgin Islands	2002	45150
128	United States Virgin Islands	2003	49300
129	United States Virgin Islands	2004	64200
130	United States Virgin Islands	2005	80300

131 rows × 3 columns

In [18]:
```
US = mobile_cellular_subscriptions[ mobile_cellular_subscriptions.Entity == 'United States Virgin Islands' ]
China = mobile_cellular_subscriptions[ mobile_cellular_subscriptions.Entity == 'China' ]
Germany = mobile_cellular_subscriptions[ mobile_cellular_subscriptions.Entity == 'Germany' ]

plt.figure(figsize=(8,7))
plt.plot(US.Year, US.Mobile_cellular_subscriptions/100000,'b.-')
plt.plot(China.Year, China.Mobile_cellular_subscriptions/100000,'r.-')
plt.plot(Germany.Year, Germany.Mobile_cellular_subscriptions/100000,'g.-')
plt.legend(['The US', 'China', 'Germany'])
plt.xlabel('Year')
plt.ylabel('Mobile_cellular_subscriptions(*100000)')
plt.title('mobile-cellular-subscriptions across countries')
plt.show()
```



In [19]:
```
plt.style.use('default')
plt.hist(mobile_cellular_subscriptions.Mobile_cellular_subscriptions/1000000,color='#3e28a8')
plt.yticks([10,20,30,40,50,60,70,80,90,100])
plt.xlabel('mobile_cellular_subscriptions')
plt.ylabel('Frequency')
plt.title('distribution of mobile_cellular_subscriptions')
plt.savefig('distribution of mobile_cellular_subscriptions.png', dpi= 300)
plt.show()
```



In [ ]:

Team project GitHub link

Python-team-project---innovation/complete python project done by team 12.pdf at main · oybek1995-creator/Python-team-project---innovation (github.com)