first I used (select* from global_data;)for all three to extract the data

Here I imported pandas and matplotlib

In [1]:

```
import pandas as p
import matplotlib.pyplot as plt
%matplotlib inline
```

I read my city_data.csv file and printed it...

In [2]:

```
a= p.read_csv('city_data.csv' )
```

In [3]:

```
print(a)
```

```
             city                country
0          Abidjan          Côte D'Ivoire
1       Abu Dhabi  United Arab Emirates
2           Abuja                Nigeria
3           Accra                  Ghana
4           Adana                 Turkey
..            ...                    ...
337       Xuzhou                  China
338  Yamoussoukro          Côte D'Ivoire
339       Yerevan                Armenia
340        Zagreb                Croatia
341       Zapopan                 Mexico

[342 rows x 2 columns]
```

I read my city_list.csv file and printed it...

In [4]:

```
b=p.read_csv('city_list.csv' )
```

In [5]:

```
print(b)
```

```
       year     city        country  avg_temp
0      1849  Abidjan  Côte D'Ivoire     25.58
1      1850  Abidjan  Côte D'Ivoire     25.52
2      1851  Abidjan  Côte D'Ivoire     25.67
3      1852  Abidjan  Côte D'Ivoire       NaN
4      1853  Abidjan  Côte D'Ivoire       NaN
...     ...      ...            ...       ...
70787  2009  Zapopan         Mexico     21.76
70788  2010  Zapopan         Mexico     20.90
70789  2011  Zapopan         Mexico     21.55
70790  2012  Zapopan         Mexico     21.52
70791  2013  Zapopan         Mexico     22.19

[70792 rows x 4 columns]
```

I read my global_data.csv file and printed it...

In [6]:

```
c=p.read_csv('global_data.csv' )
```

In [7]:

```
print(c)
```

```
     year  avg_temp
0    1750      8.72
1    1751      7.98
2    1752      5.78
3    1753      8.39
4    1754      8.47
..    ...       ...
261  2011      9.52
262  2012      9.51
263  2013      9.61
264  2014      9.57
265  2015      9.83

[266 rows x 2 columns]
```

I am searching for fimiliar countries

In [8]:

```
a[a.country.isin(['Switzerland'])]
```

Out[8]:

	city	country
43	Bern	Switzerland

In [9]:

```
a[a.country.isin(['India'])]
```

Out[9]:

	city	country
6	Agra	India
7	Ahmadabad	India
12	Allahabad	India
14	Amritsar	India
30	Bangalore	India
44	Bhopal	India
85	Delhi	India
117	Haora	India
125	Hyderabad	India
129	Indore	India
135	Jaipur	India
145	Kanpur	India
181	Ludhiana	India
215	Nagpur	India
222	New Delhi	India
238	Patna	India
255	Pune	India
260	Rajkot	India
262	Ranchi	India
298	Surat	India
319	Vadodara	India
322	Varanasi	India

I decided to take New Dehli

In [10]:

```
a[a.city.isin(['New Delhi'])]
```

Out[10]:

	city	country
222	New Delhi	India

collected all Data from New Dehli the city_list.csv that I assined to b

In [11]:

```
new_dehli=b[b.city.isin(['New Delhi'])]
print(new_dehli)
```

```
       year       city country  avg_temp
45694  1796  New Delhi   India     25.03
45695  1797  New Delhi   India     26.71
45696  1798  New Delhi   India     24.29
45697  1799  New Delhi   India     25.28
45698  1800  New Delhi   India     25.21
...     ...        ...     ...       ...
45907  2009  New Delhi   India     26.55
45908  2010  New Delhi   India     26.52
45909  2011  New Delhi   India     25.63
45910  2012  New Delhi   India     25.89
45911  2013  New Delhi   India     26.71

[218 rows x 4 columns]
```

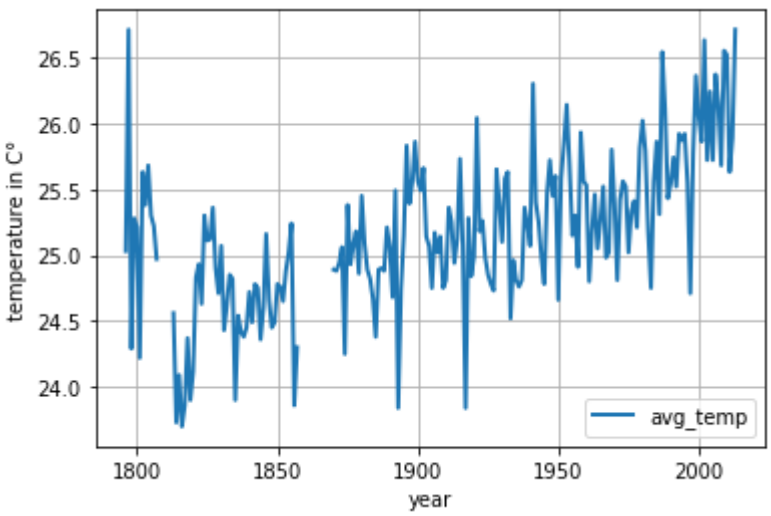
Here I ploted the average Temperature in New Dehli and you can see that the blue line is sometimes skipping over and that is for the Data that I did not get(NaN)

In [12]:

```
first=new_dehli.plot('year','avg_temp',grid=True,LineWidth=2)

plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[12]:

```
Text(0.5, 0, 'year')
```



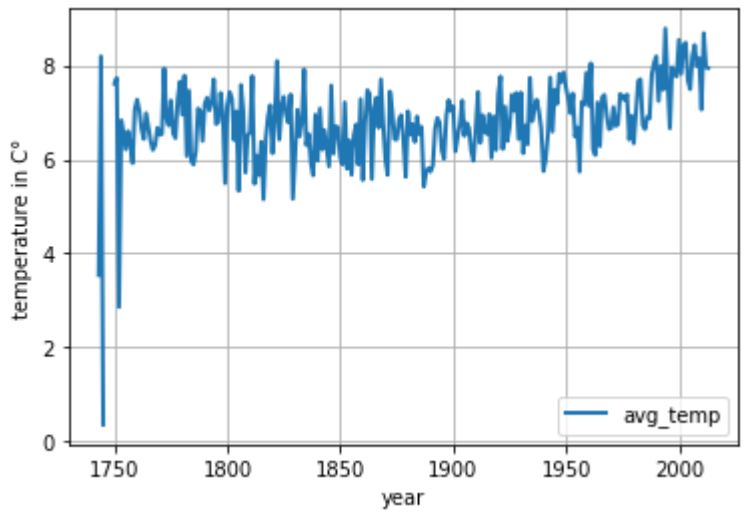
This is the plot for Bern,Zurich

In [13]:

```
s=b[b.city.isin(['Bern'])]

s.plot('year','avg_temp',grid=True,LineWidth=2)
plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[13]:

```
Text(0.5, 0, 'year')
```



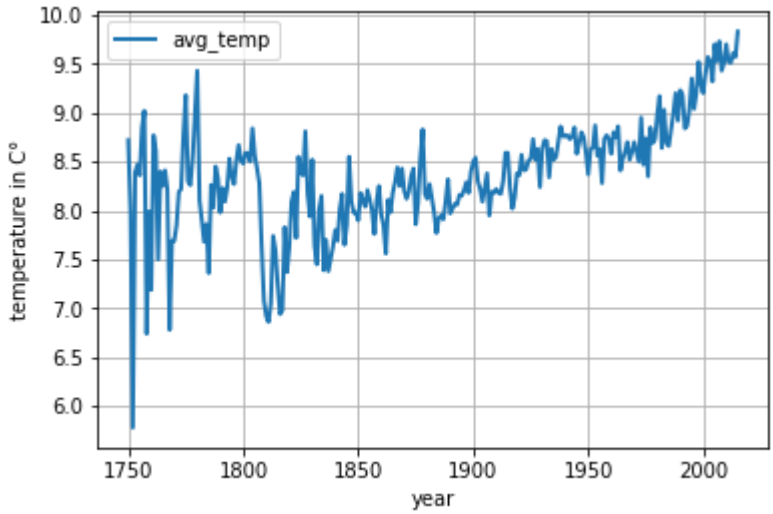
This is the average Temerature on the hole planet

In [14]:

```
second=c.plot('year','avg_temp',grid=True,LineWidth=2)

plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[14]:

```
Text(0.5, 0, 'year')
```



If you take the temperature range from o to 28 degrees c° the change does not seem visible anymore since the range is biiger but the numbers are the same and this is how you can fool somebody with satistics and at the same time prove a point.This oly proves the power of satistics.I did this to be able to compare the temperatures.

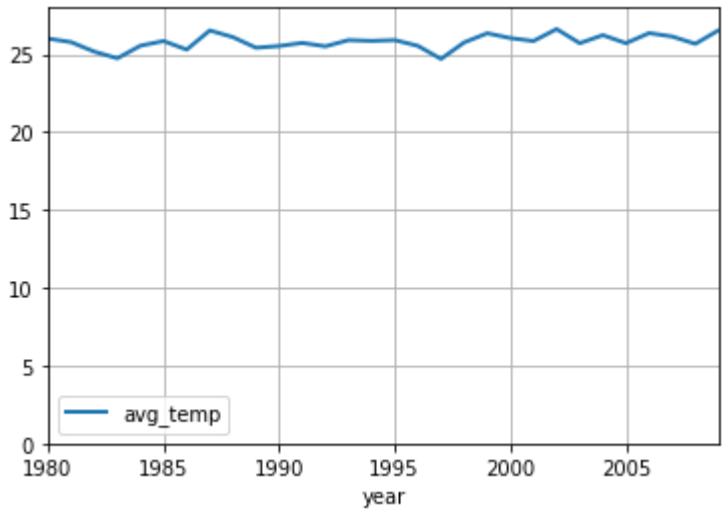
In [15]:

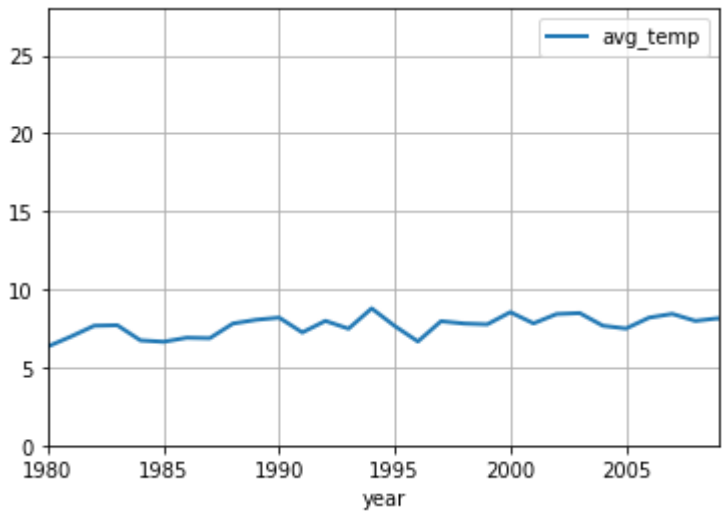
```
print('1)India,New Dehli:')
first=new_dehli.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
print('2)Switzerland,Bern:')
s.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
print('3)total average:')
second=c.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
```

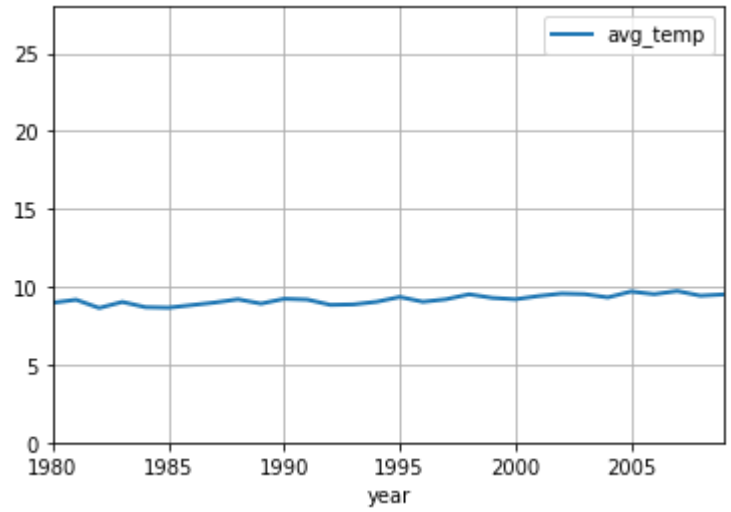
```
1)India,New Dehli:
2)Switzerland,Bern:
3)total average:
```

Out[15]:

```
[1980, 2009, 0, 28]
```







Here I created the moving avarge.You ca put in how accurate you want it to be and than press enter.

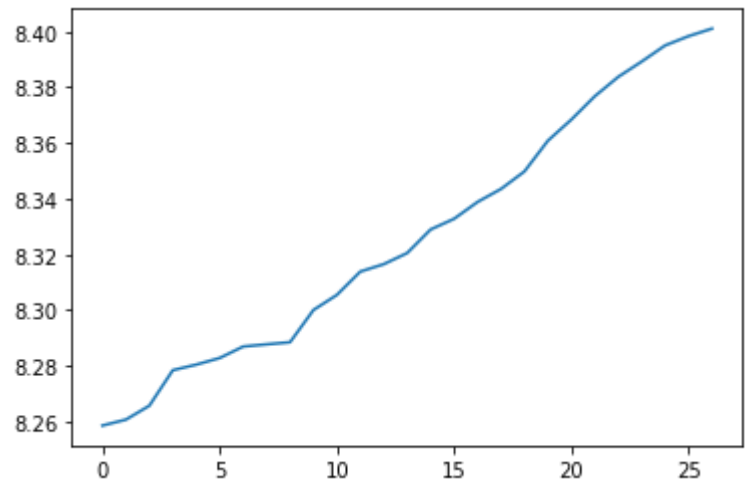
In [18]:

```
numbers = c.avg_temp
window_size = int(input("how exact do whant your moving avarage plot to be?(265 year

i = 0
moving_averages = []
while i < len(numbers) - window_size + 1:
    this_window = numbers[i : i + window_size]


    window_average = sum(this_window) / window_size
    moving_averages.append(window_average)
    i += 1

plt.plot(moving_averages)
#plt.axis([1980, 2009,0,10])
```

```
how exact do whant your moving avarage plot to be?(265 years)240
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x11af7ab10>]
```



-We can clearly see that the earth is getting warmer. -It is visible that New Dehli is warmer than the averge Temperature -Bern,Swizerland is colder than the average temperature -and that is was very warm durinf the

industrial times and that it went down from thaere and than it came back higher than ever

Thank you for your time --Simi

In [ ]:

```

```

In [ ]:

```

```