In [29]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import randn
```

In [30]:
```
p = []
for y in range(2008, 2011):
    frame = pd.read_csv('data/x%d.txt' % y, names=['name', 'sex', 'births'])
    frame['year'] = y
    p.append(frame)
```

In [31]:
```
dfnames = pd.concat(p, ignore_index=True)
dfnames
```

Out[31]:

	name	sex	births	year
0	Emma	F	18765	2008
1	Isabella	F	18564	2008
2	Emily	F	17397	2008
3	Olivia	F	17030	2008
4	Ava	F	17007	2008
...	...	...	...	...
103441	Zymaire	M	5	2010
103442	Zyonne	M	5	2010
103443	Zyquarius	M	5	2010
103444	Zyran	M	5	2010
103445	Zzyzx	M	5	2010

103446 rows × 4 columns

In [32]:
```
tb = dfnames.pivot_table('births', index='year', columns='sex', aggfunc=sum)
tb     # tb : total_births
```

Out[32]:

sex	F	M
year		
2008	1883645	2032310
2009	1827643	1973359
2010	1759010	1898382

In [33]:
```
def add_prop(g):
    g['prop'] = g.births / g.births.sum()
    return g
```

In [34]:
```
names = dfnames.groupby(['year', 'sex']).apply(add_prop)
names
```

Out[34]:

	name	sex	births	year	prop
0	Emma	F	18765	2008	0.009962
1	Isabella	F	18564	2008	0.009855
2	Emily	F	17397	2008	0.009236
3	Olivia	F	17030	2008	0.009041
4	Ava	F	17007	2008	0.009029
...	...	...	...	...	...
103441	Zymaire	M	5	2010	0.000003
103442	Zyonne	M	5	2010	0.000003
103443	Zyquarius	M	5	2010	0.000003
103444	Zyran	M	5	2010	0.000003
103445	Zzyzx	M	5	2010	0.000003

103446 rows × 5 columns

In [35]:
```
names.groupby(['year', 'sex']).prop.max()
```

Out[35]:
```
year  sex
2008  F      0.009962
      M      0.011075
2009  F      0.012159
      M      0.010660
2010  F      0.012923
      M      0.011523
Name: prop, dtype: float64
```

In [36]:
```
def top20(g):
    return g.sort_values(by='births', ascending=False)[:20]
```

In [37]:
```
g = names.groupby(['year', 'sex'])
t = g.apply(top20)
t.reset_index(inplace=True, drop=True)
t
```

Out[37]:

	name	sex	births	year	prop
0	Emma	F	18765	2008	0.009962
1	Isabella	F	18564	2008	0.009855
2	Emily	F	17397	2008	0.009236
3	Olivia	F	17030	2008	0.009041
4	Ava	F	17007	2008	0.009029
...	...	...	...	...	...
115	Matthew	M	13954	2010	0.007350
116	Logan	M	13943	2010	0.007345
117	Elijah	M	13735	2010	0.007235
118	James	M	13714	2010	0.007224
119	Joseph	M	13657	2010	0.007194

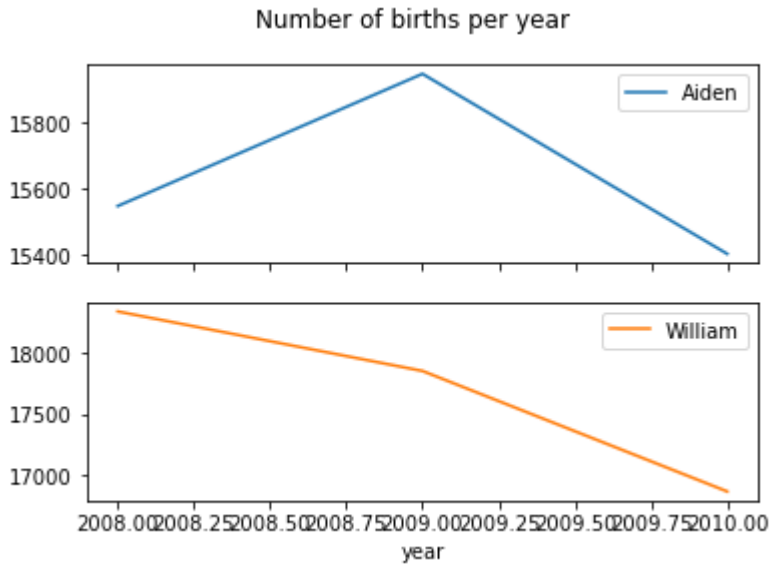
120 rows × 5 columns

In [38]:
```
total_births = t.pivot_table('births', index='year', columns='name', aggfunc=
```

In [39]:
```
total_births.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3 entries, 2008 to 2010
Data columns (total 45 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Abigail      3 non-null      float64
 1   Addison      3 non-null      float64
 2   Aiden        3 non-null      float64
 3   Alexander    3 non-null      float64
 4   Alexis       3 non-null      float64
 5   Alyssa       3 non-null      float64
 6   Andrew       3 non-null      float64
 7   Anthony      3 non-null      float64
 8   Ashley       2 non-null      float64
 9   Ava          3 non-null      float64
 10  Benjamin     1 non-null      float64
 11  Chloe        3 non-null      float64
 12  Christopher  3 non-null      float64
 13  Daniel       3 non-null      float64
 14  David        3 non-null      float64
 15  Elijah       1 non-null      float64
 16  Elizabeth    3 non-null      float64
 17  Ella         3 non-null      float64
 18  Emily        3 non-null      float64
 19  Emma         3 non-null      float64
 20  Ethan        3 non-null      float64
 21  Grace        3 non-null      float64
 22  Hailey       1 non-null      float64
 23  Hannah       1 non-null      float64
 24  Isabella     3 non-null      float64
 25  Jacob        3 non-null      float64
 26  James        3 non-null      float64
 27  Jayden       3 non-null      float64
 28  John         1 non-null      float64
 29  Joseph       3 non-null      float64
 30  Joshua       3 non-null      float64
 31  Lily         2 non-null      float64
 32  Logan        3 non-null      float64
 33  Madison      3 non-null      float64
 34  Mason        1 non-null      float64
 35  Matthew      3 non-null      float64
 36  Mia          3 non-null      float64
 37  Michael      3 non-null      float64
 38  Natalie      3 non-null      float64
 39  Noah         3 non-null      float64
 40  Olivia       3 non-null      float64
 41  Ryan         2 non-null      float64
 42  Samantha     3 non-null      float64
 43  Sophia       3 non-null      float64
 44  William      3 non-null      float64
dtypes: float64(45)
memory usage: 1.1 KB
```

In [40]:
```
s = total_births[['Aiden','William']]
s.plot(subplots=True, title="Number of births per year");
```



Measuring the increase in naming diversity

In [41]:
```
boys = t[t.sex == 'M']
girls = t[t.sex == 'F']
```

In [42]:
```
boys
```

Out[42]:

	name	sex	births	year	prop
20	Jacob	M	22507	2008	0.011075
21	Michael	M	20524	2008	0.010099
22	Ethan	M	20174	2008	0.009927
23	Joshua	M	19133	2008	0.009414
24	Daniel	M	18935	2008	0.009317
25	Alexander	M	18639	2008	0.009171
26	Anthony	M	18347	2008	0.009028
27	William	M	18337	2008	0.009023
28	Christopher	M	17876	2008	0.008796
29	Matthew	M	17526	2008	0.008624
30	Jayden	M	17088	2008	0.008408
31	Andrew	M	16711	2008	0.008223
32	Joseph	M	16483	2008	0.008110
33	David	M	16246	2008	0.007994
34	Noah	M	15748	2008	0.007749
35	Aiden	M	15547	2008	0.007650
36	James	M	15108	2008	0.007434
37	Ryan	M	14650	2008	0.007209
38	Logan	M	13818	2008	0.006799
39	John	M	13273	2008	0.006531
60	Jacob	M	21036	2009	0.010660
61	Ethan	M	19783	2009	0.010025
62	Michael	M	18822	2009	0.009538
63	Alexander	M	18175	2009	0.009210
64	William	M	17852	2009	0.009047
65	Joshua	M	17549	2009	0.008893
66	Daniel	M	17456	2009	0.008846
67	Jayden	M	17193	2009	0.008713
68	Noah	M	17176	2009	0.008704
69	Christopher	M	16264	2009	0.008242
70	Anthony	M	16237	2009	0.008228
71	Aiden	M	15945	2009	0.008080
72	Matthew	M	15895	2009	0.008055
73	David	M	15370	2009	0.007789

	name	sex	births	year	prop
74	Joseph	M	14819	2009	0.007510
75	Andrew	M	14778	2009	0.007489
76	Logan	M	14415	2009	0.007305
77	James	M	14121	2009	0.007156
78	Ryan	M	13071	2009	0.006624
79	Benjamin	M	13055	2009	0.006616
100	Jacob	M	21875	2010	0.011523
101	Ethan	M	17866	2010	0.009411
102	Michael	M	17133	2010	0.009025
103	Jayden	M	17030	2010	0.008971
104	William	M	16870	2010	0.008887
105	Alexander	M	16634	2010	0.008762
106	Noah	M	16281	2010	0.008576
107	Daniel	M	15679	2010	0.008259
108	Aiden	M	15403	2010	0.008114
109	Anthony	M	15364	2010	0.008093
110	Joshua	M	15238	2010	0.008027
111	Mason	M	14728	2010	0.007758
112	Christopher	M	14135	2010	0.007446
113	Andrew	M	14093	2010	0.007424
114	David	M	14042	2010	0.007397
115	Matthew	M	13954	2010	0.007350
116	Logan	M	13943	2010	0.007345
117	Elijah	M	13735	2010	0.007235
118	James	M	13714	2010	0.007224
119	Joseph	M	13657	2010	0.007194

In [43]:
```
dfboys = boys[boys.year == 2010]
dfboys
```

Out[43]:

	name	sex	births	year	prop
100	Jacob	M	21875	2010	0.011523
101	Ethan	M	17866	2010	0.009411
102	Michael	M	17133	2010	0.009025
103	Jayden	M	17030	2010	0.008971
104	William	M	16870	2010	0.008887
105	Alexander	M	16634	2010	0.008762
106	Noah	M	16281	2010	0.008576
107	Daniel	M	15679	2010	0.008259
108	Aiden	M	15403	2010	0.008114
109	Anthony	M	15364	2010	0.008093
110	Joshua	M	15238	2010	0.008027
111	Mason	M	14728	2010	0.007758
112	Christopher	M	14135	2010	0.007446
113	Andrew	M	14093	2010	0.007424
114	David	M	14042	2010	0.007397
115	Matthew	M	13954	2010	0.007350
116	Logan	M	13943	2010	0.007345
117	Elijah	M	13735	2010	0.007235
118	James	M	13714	2010	0.007224
119	Joseph	M	13657	2010	0.007194

In [44]:
```
pc = dfboys.sort_values(by='prop', ascending=False).prop.cumsum() # pc: prop_
pc[:10]
```

Out[44]:
```
100    0.011523
101    0.020934
102    0.029959
103    0.038930
104    0.047817
105    0.056579
106    0.065155
107    0.073414
108    0.081528
109    0.089621
Name: prop, dtype: float64
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی پیشرفته با پایتون
امین گلزاری اسکوئی
۱۴۰۰-۱۴۰۱

Codes and Projects (click here) (https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Advanced-2021) slides and videos (click here) (https://drive.google.com/drive/folders/1Dx3v7fD1QBWL-MNP2hd7iIxaRbeALkkA)