

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

1. Read the csv file as DataFrame.

```
df=pd.read_csv("nc-est2015-agesex-res.csv")
```

```
df.isnull().sum()
```

```
SEX      0
AGE      0
CENSUS2010POP  0
ESTIMATESBASE2010  0
POPESTIMATE2010  0
POPESTIMATE2011  0
POPESTIMATE2012  0
POPESTIMATE2013  0
POPESTIMATE2014  0
POPESTIMATE2015  0
dtype: int64
```

```
df
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010 \
0	0	0	3944153	3944160	3951330
1	0	1	3978070	3978090	3957888
2	0	2	4096929	4096939	4090862
3	0	3	4119040	4119051	4111920
4	0	4	4063170	4063186	4077551
...
301	2	97	53582	53605	54118
302	2	98	36641	36675	37532
303	2	99	26193	26214	26074
304	2	100	44202	44246	45058
305	2	999	156964212	156969328	157258820

	POPESTIMATE2011	POPESTIMATE2012	POPESTIMATE2013
POPESTIMATE2014 \			
0	3963087	3926540	3931141
3949775			
1	3966551	3977939	3942872
3949776			
2	3971565	3980095	3992720
3959664			
3	4102470	3983157	3992734
4007079			
4	4122294	4112849	3994449
4005716			

```

..          ...          ...          ...          ..
.
301          57159          59533          61255
62779
302          40116          42857          44359
46208
303          27030          29320          31112
32517
304          47556          50661          53902
58008
305          158427085          159581546          160720625
161952064

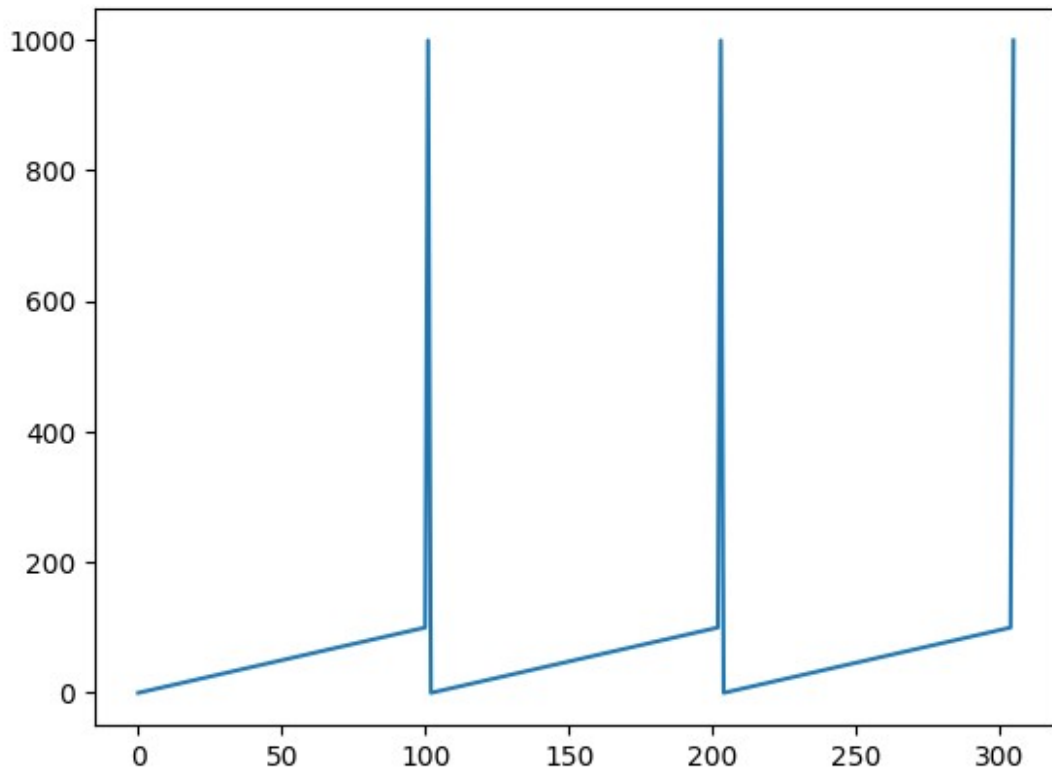
      POPESTIMATE2015
0          3978038
1          3968564
2          3966583
3          3974061
4          4020035
..          ...
301          69285
302          47272
303          34064
304          61886
305          163189523

[306 rows x 10 columns]

plt.plot(df["AGE"])

[<matplotlib.lines.Line2D at 0x1dd2b67ced0>]

```



```
df["AGE"].value_counts().tail(50)
```

AGE	
50	3
13	3
23	3
22	3
21	3
20	3
19	3
18	3
17	3
16	3
15	3
14	3
12	3
49	3
11	3
10	3
9	3
8	3
7	3
6	3
5	3
4	3

```

3      3
2      3
24     3
25     3
26     3
27     3
48     3
47     3
46     3
45     3
44     3
43     3
42     3
41     3
40     3
39     3
38     3
37     3
36     3
35     3
34     3
33     3
32     3
31     3
30     3
29     3
28     3
999    3
Name: count, dtype: int64

```

1. Find size

```

df.shape

(306, 10)

```

1. Describe the data.

```

df.describe()

```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010 \
count	306.000000	306.000000	3.060000e+02	3.060000e+02
mean	1.000000	59.303922	4.035889e+06	4.036054e+06
std	0.817834	98.060950	2.140881e+07	2.140968e+07
min	0.000000	0.000000	6.073000e+03	6.075000e+03
25%	0.000000	25.000000	9.987065e+05	9.987415e+05
50%	1.000000	50.500000	2.045286e+06	2.045342e+06
75%	2.000000	76.000000	2.304370e+06	2.304525e+06
max	2.000000	999.000000	3.087455e+08	3.087581e+08

	POPESTIMATE2010	POPESTIMATE2011	POPESTIMATE2012
count	3.060000e+02	3.060000e+02	3.060000e+02
mean	4.043750e+06	4.074756e+06	4.105917e+06
std	2.145038e+07	2.161430e+07	2.177900e+07
min	6.104000e+03	6.606000e+03	7.459000e+03
25%	1.010409e+06	1.045244e+06	1.073482e+06
50%	2.047912e+06	2.051284e+06	2.054609e+06
75%	2.304142e+06	2.321829e+06	2.335085e+06
max	3.093469e+08	3.117189e+08	3.141026e+08

	POPESTIMATE2014	POPESTIMATE2015
count	3.060000e+02	3.060000e+02
mean	4.168724e+06	4.201553e+06
std	2.211105e+07	2.228462e+07
min	8.951000e+03	9.577000e+03
25%	1.156036e+06	1.162476e+06
50%	2.060284e+06	2.062285e+06
75%	2.387392e+06	2.415332e+06
max	3.189074e+08	3.214188e+08

1. Re-label column years as "2014" and "2010"

```
df.rename(columns={"POPESTIMATE2014": "2014", "POPESTIMATE2010": "2010"},
inplace=True)
```

```
df
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	2010
POPESTIMATE2011 \					
0	0	0	3944153	3944160	3951330
1	0	1	3978070	3978090	3957888
2	0	2	4096929	4096939	4090862
3	0	3	4119040	4119051	4111920
4	0	4	4063170	4063186	4077551
...

301	2	97	53582	53605	54118
57159					
302	2	98	36641	36675	37532
40116					
303	2	99	26193	26214	26074
27030					
304	2	100	44202	44246	45058
47556					
305	2	999	156964212	156969328	157258820
158427085					

	POPESTIMATE2012	POPESTIMATE2013	2014	POPESTIMATE2015
0	3926540	3931141	3949775	3978038
1	3977939	3942872	3949776	3968564
2	3980095	3992720	3959664	3966583
3	3983157	3992734	4007079	3974061
4	4112849	3994449	4005716	4020035
...
301	59533	61255	62779	69285
302	42857	44359	46208	47272
303	29320	31112	32517	34064
304	50661	53902	58008	61886
305	159581546	160720625	161952064	163189523

[306 rows x 10 columns]

1. Display first 10 rows.

```
df.head(10)
```

SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	2010
POPESTIMATE2011 \				
0	0	0	3944153	3944160
3963087				3951330
1	0	1	3978070	3978090
3966551				3957888
2	0	2	4096929	4096939
3971565				4090862
3	0	3	4119040	4119051
4102470				4111920
4	0	4	4063170	4063186
4122294				4077551
5	0	5	4056858	4056872
4087709				4064653
6	0	6	4066381	4066412
4074993				4073013
7	0	7	4030579	4030594
4083225				4043046
8	0	8	4046486	4046497
4053203				4025604

9	0	9	4148353	4148369	4125415
4035710					
	POPESTIMATE2012	POPESTIMATE2013	2014	POPESTIMATE2015	
0	3926540	3931141	3949775	3978038	
1	3977939	3942872	3949776	3968564	
2	3980095	3992720	3959664	3966583	
3	3983157	3992734	4007079	3974061	
4	4112849	3994449	4005716	4020035	
5	4132242	4123626	4006900	4018158	
6	4097605	4142916	4135930	4019207	
7	4084913	4108349	4155326	4148360	
8	4093177	4095711	4120903	4167887	
9	4063152	4104072	4108349	4133564	

1. Select 15th and 22nd rows.

```
df.iloc[[14,21],:]
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	2010
POPESTIMATE2011 \					
14	0	14	4165982	4166059	4145614
4131501					
21	0	21	4354294	4354603	4387956
4596166					
	POPESTIMATE2012	POPESTIMATE2013	2014	POPESTIMATE2015	
14	4135460	4148966	4233839	4185386	
21	4632892	4580130	4492373	4456790	

1. Add columns change, percentage change.

```
df["change"]=df["POPESTIMATE2012"].diff()
df["Percentage_Change"]=df["POPESTIMATE2012"].pct_change()*100
df
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	2010
POPESTIMATE2011 \					
0	0	0	3944153	3944160	3951330
3963087					
1	0	1	3978070	3978090	3957888
3966551					
2	0	2	4096929	4096939	4090862
3971565					
3	0	3	4119040	4119051	4111920
4102470					
4	0	4	4063170	4063186	4077551
4122294					
..

```

...
301 2 97 53582 53605 54118
57159
302 2 98 36641 36675 37532
40116
303 2 99 26193 26214 26074
27030
304 2 100 44202 44246 45058
47556
305 2 999 156964212 156969328 157258820
158427085

POPESTIMATE2012 POPESTIMATE2013 2014 POPESTIMATE2015 \
0 3926540 3931141 3949775 3978038
1 3977939 3942872 3949776 3968564
2 3980095 3992720 3959664 3966583
3 3983157 3992734 4007079 3974061
4 4112849 3994449 4005716 4020035
.. ..
301 59533 61255 62779 69285
302 42857 44359 46208 47272
303 29320 31112 32517 34064
304 50661 53902 58008 61886
305 159581546 160720625 161952064 163189523

change Percentage_Change
0 NaN NaN
1 51399.0 1.309015
2 2156.0 0.054199
3 3062.0 0.076933
4 129692.0 3.256010
.. ..
301 -20437.0 -25.555833
302 -16676.0 -28.011355
303 -13537.0 -31.586439
304 21341.0 72.786494
305 159530885.0 314898.807761

[306 rows x 12 columns]

```

This will create two new columns in the DataFrame: `change`, which is the difference between the close and open prices, and `percent_change`, which is the percentage change in the close price relative to the previous row. Note that the first row of the `percent_change` column will be `NaN`, since there is no previous row to compare it to.

1. Analyse the population changes (67-71) – show “graying of US” aspect

```

df=pd.read_csv("nc-est2015-agesex-res.csv")
df=df[66:71]

```



```
for c in df.columns:
    df[c+"change"]=df[c].diff()
df
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010	\
66	0	66	2639141	2639209	2621335	
67	0	67	2649365	2649426	2693707	
68	0	68	2323672	2323736	2359816	
69	0	69	2142324	2142381	2167830	
70	0	70	2043121	2043178	2062577	

	POPESTIMATE2011	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014	\
66	2646364	2644042	3531521	3347060	
67	2586302	2612002	2609390	3485241	
68	2654106	2548669	2574501	2572359	
69	2321925	2612071	2507744	2534295	
70	2129707	2282127	2566440	2465438	

	POPESTIMATE2015	SEXchange	AGEchange	CENSUS2010POPchange	\
66	3344134	NaN	NaN	NaN	
67	3304187	0.0	1.0	10224.0	
68	3436357	0.0	1.0	-325693.0	
69	2532747	0.0	1.0	-181348.0	
70	2492490	0.0	1.0	-99203.0	

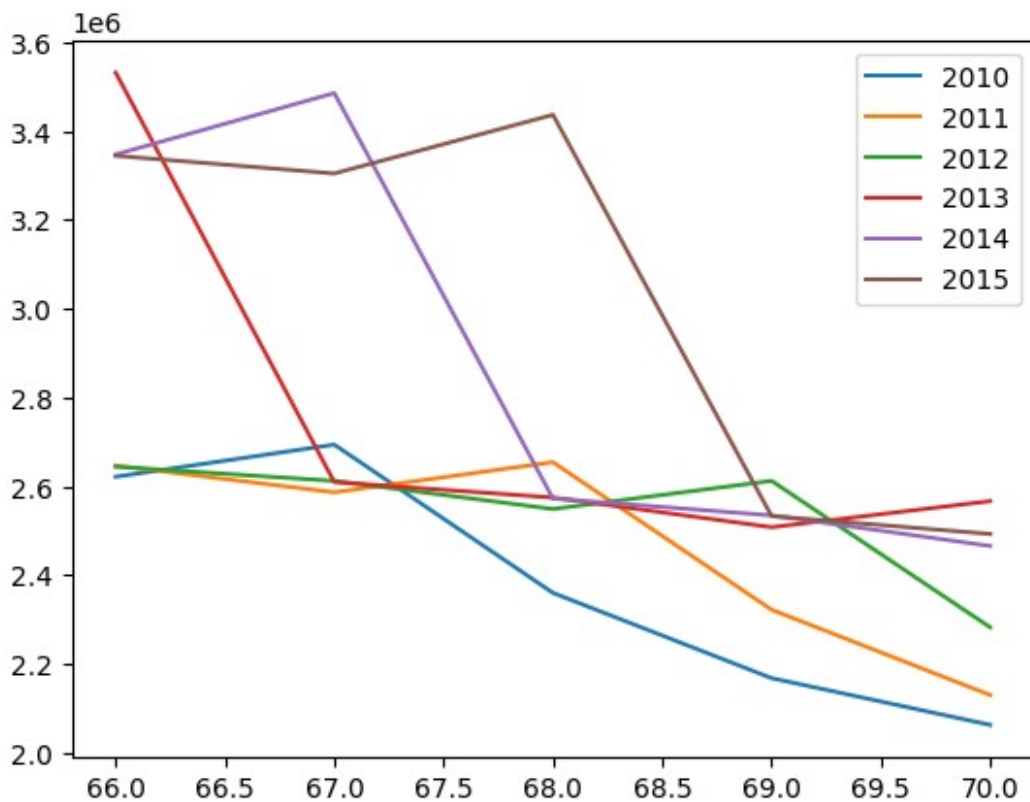
	ESTIMATESBASE2010change	POPESTIMATE2010change	
POPESTIMATE2011change \			
66	NaN	NaN	
NaN			
67	10217.0	72372.0	-
60062.0			
68	-325690.0	-333891.0	
67804.0			
69	-181355.0	-191986.0	-
332181.0			
70	-99203.0	-105253.0	-
192218.0			

	POPESTIMATE2012change	POPESTIMATE2013change	
POPESTIMATE2014change \			
66	NaN	NaN	
NaN			
67	-32040.0	-922131.0	
138181.0			

68	-63333.0	-34889.0	-
912882.0			
69	63402.0	-66757.0	-
38064.0			
70	-329944.0	58696.0	-
68857.0			

POPESTIMATE2015change	
66	NaN
67	-39947.0
68	132170.0
69	-903610.0
70	-40257.0

```
plt.plot(df["AGE"],df["POPESTIMATE2010"],label="2010")
plt.plot(df["AGE"],df["POPESTIMATE2011"],label="2011")
plt.plot(df["AGE"],df["POPESTIMATE2012"],label="2012")
plt.plot(df["AGE"],df["POPESTIMATE2013"],label="2013")
plt.plot(df["AGE"],df["POPESTIMATE2014"],label="2014")
plt.plot(df["AGE"],df["POPESTIMATE2015"],label="2015")
plt.legend()
plt.show()
```



1. Analyse gender ratio of boys and girls among infants (age=0)

```
df=pd.read_csv("nc-est2015-agesex-res.csv")
d = df[df["AGE"]==0]
d=d[0:2]
d
```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010	\
0	0	0	3944153	3944160	3951330	
102	1	0	2014276	2014278	2018420	

	POPESTIMATE2011	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014	\
0	3963087	3926540	3931141	3949775	
102	2028427	2007717	2009528	2020326	

	POPESTIMATE2015
0	3978038
102	2035134

1. Find years where the ratio of females to males ranged from 2 to 4.

```
d=d.iloc[0:,4:]
d
```

	POPESTIMATE2010	POPESTIMATE2011	POPESTIMATE2012	POPESTIMATE2013	\
0	3951330	3963087	3926540	3931141	
102	2018420	2028427	2007717	2009528	

	POPESTIMATE2014	POPESTIMATE2015
0	3949775	3978038
102	2020326	2035134

```
d.iloc[0]=d.iloc[0]/d.iloc[1]
```

```
d=d.iloc[0,:]
```

```
d
```

POPESTIMATE2010	1.957635
POPESTIMATE2011	1.953774
POPESTIMATE2012	1.955724
POPESTIMATE2013	1.956251
POPESTIMATE2014	1.955019
POPESTIMATE2015	1.954681

Name: 0, dtype: float64

```
f=d.index
for i in range(0,6):
```

```

    if d[i]>1:
        print(f[i][11:])
2010
2011
2012
2013
2014
2015

```

1. Plot the gender ratio versus age.

```

df=pd.read_csv("nc-est2015-agesex-res.csv")
df

```

	SEX	AGE	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010 \
0	0	0	3944153	3944160	3951330
1	0	1	3978070	3978090	3957888
2	0	2	4096929	4096939	4090862
3	0	3	4119040	4119051	4111920
4	0	4	4063170	4063186	4077551
...
301	2	97	53582	53605	54118
302	2	98	36641	36675	37532
303	2	99	26193	26214	26074
304	2	100	44202	44246	45058
305	2	999	156964212	156969328	157258820
	POPESTIMATE2011		POPESTIMATE2012	POPESTIMATE2013	
POPESTIMATE2014 \					
0	3963087		3926540	3931141	
3949775					
1	3966551		3977939	3942872	
3949776					
2	3971565		3980095	3992720	
3959664					
3	4102470		3983157	3992734	
4007079					
4	4122294		4112849	3994449	
4005716					
...	
.					
301	57159		59533	61255	
62779					
302	40116		42857	44359	
46208					
303	27030		29320	31112	
32517					
304	47556		50661	53902	
58008					

```
305      158427085      159581546      160720625
161952064
```

```
      POPESTIMATE2015
0      3978038
1      3968564
2      3966583
3      3974061
4      4020035
..      ...
301      69285
302      47272
303      34064
304      61886
305      163189523
```

```
[306 rows x 10 columns]
```

```
df=df[df["AGE"]<999]
df=df[df["SEX"]<2]
```

```
age=df["AGE"]
age
```

```
0      0
1      1
2      2
3      3
4      4
..      ...
198     96
199     97
200     98
201     99
202    100
```

```
Name: AGE, Length: 202, dtype: int64
```

```
d=df.groupby(["AGE","SEX"]).sum()
```

```
d
```

```
      CENSUS2010POP  ESTIMATESBASE2010  POPESTIMATE2010
POPESTIMATE2011 \
AGE SEX
0  0      3944153      3944160      3951330
3963087
  1      2014276      2014278      2018420
2028427
1  0      3978070      3978090      3957888
3966551
```

1	2030853	2030861	2020332
2025522			
2 0	4096929	4096939	4090862
3971565			
...
...			
98 1	9259	9263	9505
10554			
99 0	32266	32289	32178
33636			
1	6073	6075	6104
6606			
100 0	53364	53412	54410
57702			
1	9162	9166	9352
10146			

	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014
POPESTIMATE2015			
AGE SEX			

0 0	3926540	3931141	3949775
3978038			
1	2007717	2009528	2020326
2035134			
1 0	3977939	3942872	3949776
3968564			
1	2035460	2015470	2018401
2029295			
2 0	3980095	3992720	3959664
3966583			
...
...			
98 1	11652	12656	13518
14719			
99 0	36779	39271	41468
43641			
1	7459	8159	8951
9577			
100 0	61821	66189	71626
76974			
1	11160	12287	13618
15088			

[202 rows x 8 columns]

```
for i in range(0,len(age),2):
    for j in range(2,8):
        d.iloc[(i,j)]=d.iloc[(i,j)]/d.iloc[(i+1,j)]
d
```

CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010
POPESTIMATE2011 \		
AGE SEX		

0 0	3944153	3944160	1.957635e+00
1.953774e+00			
1	2014276	2014278	2.018420e+06
2.028427e+06			
1 0	3978070	3978090	1.959029e+00
1.958286e+00			
1	2030853	2030861	2.020332e+06
2.025522e+06			
2 0	4096929	4096939	1.958583e+00
1.959358e+00			
...
...			
98 1	9259	9263	9.505000e+03
1.055400e+04			
99 0	32266	32289	5.271625e+00
5.091735e+00			
1	6073	6075	6.104000e+03
6.606000e+03			
100 0	53364	53412	5.818007e+00
5.687167e+00			
1	9162	9166	9.352000e+03
1.014600e+04			

POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014
POPESTIMATE2015		
AGE SEX		

0 0	1.955724e+00	1.956251e+00	1.955019e+00
1.954681e+00			
1	2.007717e+06	2.009528e+06	2.020326e+06
2.035134e+06			
1 0	1.954319e+00	1.956304e+00	1.956884e+00
1.955637e+00			
1	2.035460e+06	2.015470e+06	2.018401e+06
2.029295e+06			
2 0	1.958617e+00	1.954677e+00	1.956672e+00
1.957256e+00			
...
...			
98 1	1.165200e+04	1.265600e+04	1.351800e+04
1.471900e+04			
99 0	4.930822e+00	4.813212e+00	4.632778e+00
4.556855e+00			
1	7.459000e+03	8.159000e+03	8.951000e+03
9.577000e+03			
100 0	5.539516e+00	5.386913e+00	5.259656e+00

```
5.101670e+00
1 1.116000e+04 1.228700e+04 1.361800e+04
1.508800e+04
```

```
[202 rows x 8 columns]
```

```
a=d.index
v=[]
for i in a:
    v.append(i[1])
v=np.array(v)
d["SEX"]=v
```

```
d
```

```
          CENSUS2010POP  ESTIMATESBASE2010  POPESTIMATE2010
POPESTIMATE2011 \
AGE SEX
```

```
0  0 3944153 3944160 1.957635e+00
1.953774e+00
1  0 2014276 2014278 2.018420e+06
2.028427e+06
1  0 3978070 3978090 1.959029e+00
1.958286e+00
1  0 2030853 2030861 2.020332e+06
2.025522e+06
2  0 4096929 4096939 1.958583e+00
1.959358e+00
... ..
... ..
98 1 9259 9263 9.505000e+03
1.055400e+04
99 0 32266 32289 5.271625e+00
5.091735e+00
1  0 6073 6075 6.104000e+03
6.606000e+03
100 0 53364 53412 5.818007e+00
5.687167e+00
1  0 9162 9166 9.352000e+03
1.014600e+04
```

```
          POPESTIMATE2012  POPESTIMATE2013  POPESTIMATE2014
POPESTIMATE2015 \
AGE SEX
```

```
0  0 1.955724e+00 1.956251e+00 1.955019e+00
1.954681e+00
1  0 2.007717e+06 2.009528e+06 2.020326e+06
2.035134e+06
```


1	0	1.954319e+00	1.956304e+00	1.956884e+00
		1.955637e+00		
	1	2.035460e+06	2.015470e+06	2.018401e+06
		2.029295e+06		
2	0	1.958617e+00	1.954677e+00	1.956672e+00
		1.957256e+00		
...	
...				
98	1	1.165200e+04	1.265600e+04	1.351800e+04
		1.471900e+04		
99	0	4.930822e+00	4.813212e+00	4.632778e+00
		4.556855e+00		
	1	7.459000e+03	8.159000e+03	8.951000e+03
		9.577000e+03		
100	0	5.539516e+00	5.386913e+00	5.259656e+00
		5.101670e+00		
	1	1.116000e+04	1.228700e+04	1.361800e+04
		1.508800e+04		

SEX

AGE	SEX	
0	0	0
	1	1
1	0	0
	1	1
2	0	0
...		...
98	1	1
99	0	0
	1	1
100	0	0
	1	1

[202 rows x 9 columns]

```
d=d[d["SEX"]<1]
d
```

		CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010
POPESTIMATE2011	\			
AGE	SEX			
0	0	3944153	3944160	1.957635
		1.953774		
1	0	3978070	3978090	1.959029
		1.958286		
2	0	4096929	4096939	1.958583
		1.959358		
3	0	4119040	4119051	1.956872
		1.958878		

4	0	4063170	4063186	1.956305
1.957104				
...				
...				
96	0	95223	95288	4.411997
4.254485				
97	0	68138	68168	4.662809
4.546944				
98	0	45900	45938	4.948659
4.801023				
99	0	32266	32289	5.271625
5.091735				
100	0	53364	53412	5.818007
5.687167				

	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014
POPESTIMATE2015 \			
AGE SEX			

0	0	1.955724	1.956251	1.955019
1.954681				
1	0	1.954319	1.956304	1.956884
1.955637				
2	0	1.958617	1.954677	1.956672
1.957256				
3	0	1.959705	1.958974	1.955058
1.957045				
4	0	1.959144	1.959963	1.959248
1.955342				
...				
...				
96	0	4.142240	3.966077	3.879979
3.771275				
97	0	4.378142	4.276019	4.091039
4.000390				
98	0	4.678081	4.504978	4.418257
4.211631				
99	0	4.930822	4.813212	4.632778
4.556855				
100	0	5.539516	5.386913	5.259656
5.101670				

	SEX	SEX
AGE	SEX	
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...		

```

96  0      0
97  0      0
98  0      0
99  0      0
100 0      0

```

```
[101 rows x 9 columns]
```

```

a=np.unique(age)
d["AGEE"]=a
d

```

C:\Users\vigne\AppData\Local\Temp\ipykernel_19252\237546701.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
d["AGEE"]=a

	CENSUS2010POP	ESTIMATESBASE2010	POPESTIMATE2010
POPESTIMATE2011 \			
AGE SEX			
0 0	3944153	3944160	1.957635
1.953774			
1 0	3978070	3978090	1.959029
1.958286			
2 0	4096929	4096939	1.958583
1.959358			
3 0	4119040	4119051	1.956872
1.958878			
4 0	4063170	4063186	1.956305
1.957104			
...
...			
96 0	95223	95288	4.411997
4.254485			
97 0	68138	68168	4.662809
4.546944			
98 0	45900	45938	4.948659
4.801023			
99 0	32266	32289	5.271625
5.091735			
100 0	53364	53412	5.818007
5.687167			

POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014
-----------------	-----------------	-----------------

```
POPESTIMATE2015 \
AGE SEX
```

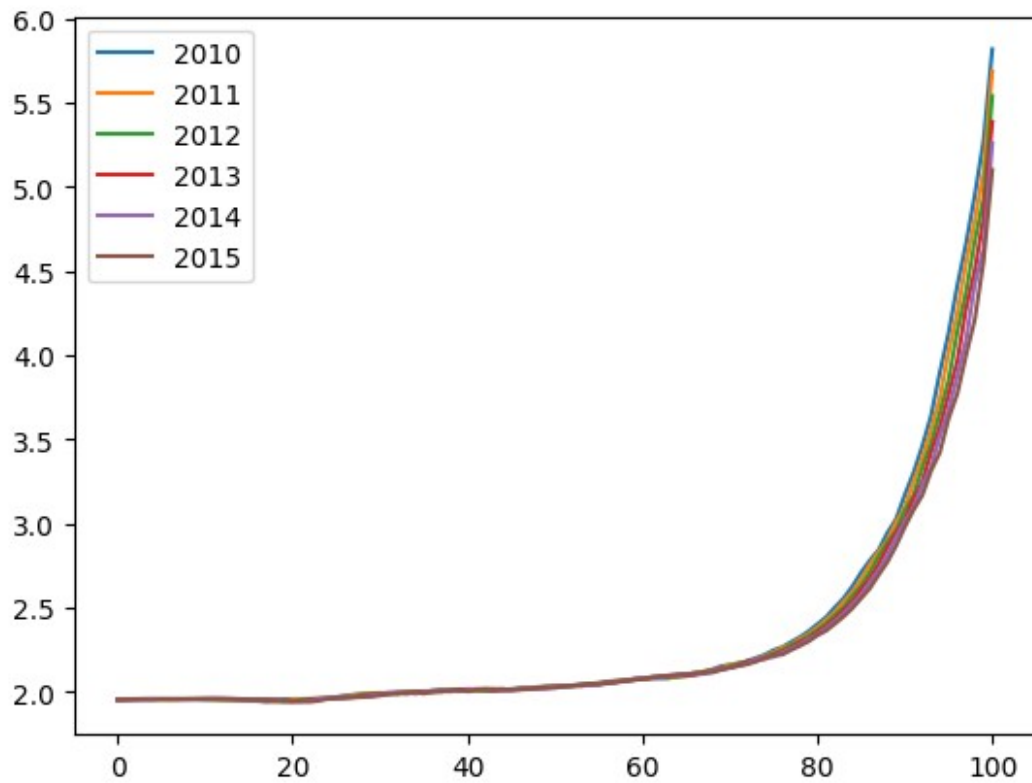
0	0	1.955724	1.956251	1.955019
1	0	1.954319	1.956304	1.956884
2	0	1.958617	1.954677	1.956672
3	0	1.959705	1.958974	1.955058
4	0	1.959144	1.959963	1.959248
...	
96	0	4.142240	3.966077	3.879979
97	0	4.378142	4.276019	4.091039
98	0	4.678081	4.504978	4.418257
99	0	4.930822	4.813212	4.632778
100	0	5.539516	5.386913	5.259656

AGE	SEX	SEXX	AGEE
0	0	0	0
1	0	0	1
2	0	0	2
3	0	0	3
4	0	0	4
...	
96	0	0	96
97	0	0	97
98	0	0	98
99	0	0	99
100	0	0	100

```
[101 rows x 10 columns]
```

```
plt.plot(d["AGEE"],d["POPESTIMATE2010"],label="2010")
plt.plot(d["AGEE"],d["POPESTIMATE2011"],label="2011")
plt.plot(d["AGEE"],d["POPESTIMATE2012"],label="2012")
plt.plot(d["AGEE"],d["POPESTIMATE2013"],label="2013")
plt.plot(d["AGEE"],d["POPESTIMATE2014"],label="2014")
plt.plot(d["AGEE"],d["POPESTIMATE2015"],label="2015")
plt.legend()
```

<matplotlib.legend.Legend at 0x1dd2b6d6150>



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