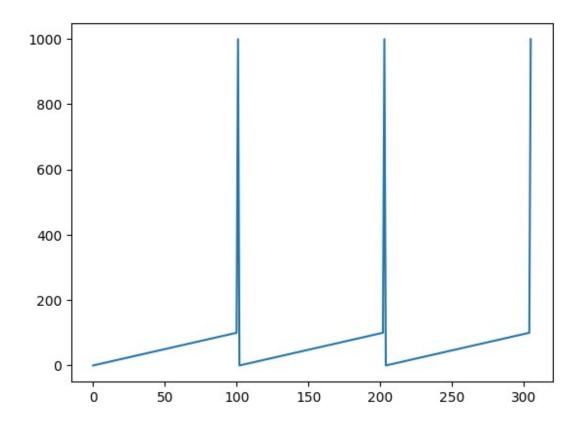
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1. Read the csy file as DataFrame.

```
df=pd.read csv("nc-est2015-agesex-res.csv")
df.isnull().sum()
SEX
                      0
AGE
                      0
CENSUS2010P0P
                      0
                      0
ESTIMATESBASE2010
POPESTIMATE2010
                      0
POPESTIMATE2011
                      0
                      0
POPESTIMATE2012
                      0
POPESTIMATE2013
POPESTIMATE2014
                      0
POPESTIMATE2015
                      0
dtype: int64
df
          AGE
              CENSUS2010P0P
                               ESTIMATESBASE2010
                                                   POPESTIMATE2010
     SEX
0
            0
                      3944153
                                          3944160
                                                            3951330
       0
1
            1
       0
                      3978070
                                          3978090
                                                            3957888
2
       0
            2
                      4096929
                                          4096939
                                                            4090862
3
       0
            3
                      4119040
                                          4119051
                                                            4111920
4
       0
            4
                      4063170
                                          4063186
                                                            4077551
       2
301
           97
                        53582
                                            53605
                                                              54118
302
       2
           98
                        36641
                                            36675
                                                              37532
       2
303
           99
                        26193
                                            26214
                                                              26074
       2
304
          100
                        44202
                                            44246
                                                              45058
       2
305
          999
                    156964212
                                        156969328
                                                          157258820
     POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013
POPESTIMATE2014 \
             3963087
                               3926540
                                                 3931141
3949775
             3966551
                               3977939
                                                 3942872
3949776
             3971565
                               3980095
                                                 3992720
3959664
                                                 3992734
             4102470
                               3983157
3
4007079
             4122294
                               4112849
                                                 3994449
4005716
```

```
301
                57159
                                  59533
                                                   61255
62779
                                 42857
                40116
                                                   44359
302
46208
                                 29320
303
                27030
                                                   31112
32517
                47556
                                 50661
304
                                                   53902
58008
305
           158427085
                             159581546
                                               160720625
161952064
     POPESTIMATE2015
0
             3978038
1
             3968564
2
             3966583
3
             3974061
4
             4020035
301
                69285
                47272
302
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
         13
23
22
21
20
19
18
17
16
15
14
12
49
11
10
9
8
7
6
5
4
```

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

1. Find size

df.shape (306, 10)

1. Describe the data.

df.describe()

```
CENSUS2010P0P
                                                ESTIMATESBASE2010
              SEX
                           AGE
       306.000000
                    306.000000
                                 3.060000e+02
                                                     3.060000e+02
count
         1.000000
                     59.303922
                                 4.035889e+06
                                                     4.036054e+06
mean
std
         0.817834
                     98.060950
                                 2.140881e+07
                                                     2.140968e+07
         0.000000
                     0.000000
                                 6.073000e+03
                                                     6.075000e+03
min
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
```

PC	PESTIMATE2010	POPESTIMATE2011	POPESTIMATE2012
POPESTIMA	ATE2013 \		
count		3.060000e+02	3.060000e+02
3.060000e			
	4.043750e+06	4.074756e+06	4.105917e+06
4.136306e			
	2.145038e+07	2.161430e+07	2.177900e+07
2.193965e	-	C C0C00002	7 450000 02
	6.104000e+03	6.606000e+03	7.459000e+03
8.159000e	1.010409e+06	1.045244e+06	1.073482e+06
1.121864e		1.0432446+00	1.0/34020+00
	2.047912e+06	2.051284e+06	2.054609e+06
2.050533e		210312046100	210340030100
	2.304142e+06	2.321829e+06	2.335085e+06
2.3494506	e+06		
max	3.093469e+08	3.117189e+08	3.141026e+08
3.1642746	e+08		
	PESTIMATE2014		
count	3.060000e+02		
		4.201553e+06	
		2.228462e+07 9.577000e+03	
25%	1.156036e+06		
50%		2.062285e+06	
75%		2.415332e+06	
max	3.189074e+08		

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

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

inp	lace=T	rue)			
df					
				ESTIMATESBASE2010	2010
POF	PESTIMA	TE201	•		
0	0	0	3944153	3944160	3951330
396	3087				
1	0	1	3978070	3978090	3957888
396	66551				
2	0	2	4096929	4096939	4090862
397	71565				
3	0	3	4119040	4119051	4111920
416	2470				
4	0	4	4063170	4063186	4077551
412	22294				

301	2	97	535	82	53605	54118	
57159							
302	2	98	366	641	36675	37532	
40116 303	2	99	261	0.3	26214	26074	
27030		99	201	.93	20214	20074	
304	2	100	442	.02	44246	45058	
47556							
305	2	999	1569642	212	156969328	157258820	
158427	085						
D	ODE	STIMATE20	112 DADE	STIMATE2013	2014	POPESTIMATE	2015
0	OFL	39265		3931141	3949775		78038
1		39779	-	3942872			8564
2		39800	95	3992720	3959664	396	6583
1 2 3 4		39831	_	3992734	4007079		74061
4		41128	349	3994449	4005716	402	20035
201		F.O.F.		61255	62770		
301 302		595 428		61255 44359			59285 17272
303		293		31112			34064
304		506		53902			51886
305		1595815		160720625		16318	
		10 7	-				
[306 r	OWS	x 10 col	Lumns]				

1. Display first 10 rows.

df	. head	(10)			
			CENSUS2010POP	ESTIMATESBASE2010	2010
	0 53087		3944153	3944160	3951330
	0 66551		3978070	3978090	3957888
	0 71565	2	4096929	4096939	4090862
	0 92470		4119040	4119051	4111920
	0 22294	4	4063170	4063186	4077551
	0 37709	5	4056858	4056872	4064653
6	0 74993		4066381	4066412	4073013
7	0 33225	7	4030579	4030594	4043046
8	0 53203		4046486	4046497	4025604

9	0	9	4148353	4148369	4125415
40	35710				
	DODECT	IMATE2012	POPESTIMATE2013	2014	POPESTIMATE2015
0	FUFLST	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 8		4084913 4093177	4108349 4095711	4155326 4120903	4148360 4167887
9		4063152	4104072	4108349	4133564

1. Select 15th and 22nd rows.

df.	iloc[[14,2	1],:]					
	SEX	AGE		2010P0P	ESTIMATE	SBASE2010	2010	
POP	ESTIM	ATE20	11 \					
14	0	14		4165982		4166059	4145614	
413	1501							
21	0	21		4354294		4354603	4387956	
459	6166							
	P0PE	STIMA [®]	TE2012	POPESTI	MATE2013	2014	POPESTIM/	ATE2015
14		4	135460		4148966	4233839	4	4185386
21		4	632892		4580130	4492373	4	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
                     3944153
                                        3944160
                                                   3951330
3963087
                     3978070
                                        3978090
                                                   3957888
3966551
            2
                     4096929
                                        4096939
                                                   4090862
3971565
                     4119040
                                        4119051
                                                   4111920
4102470
                                        4063186
                                                   4077551
                     4063170
4122294
```

201	2	0.7		E2E02		E260E	E 4110		
301 57159	2	97		53582		53605	54118		
302	2	98		36641		36675	37532		
40116	_			0.01.00					
303 27030	2	99		26193		26214	26074		
304	2	100		44202		44246	45058		
47556	_			• _			.5555		
305	2	999	156	5964212		156969328	157258820		
158427	/085								
0 1 2 3 4 301 302 303 304 305	POPES	42 29	540 939 095 157 849 533 857 320 661	394 399 399	E2013 31141 42872 92720 92734 94449 61255 44359 31112 53902 20625	2014 3949775 3949776 3959664 4007079 4005716 62779 46208 32517 58008 161952064		ATE2015 8978038 8968564 8966583 8974061 4020035 69285 47272 34064 61886 8189523	
0 1 2 3 4 301 302 303 304	1; -; -;	change NaN 51399.0 2156.0 3062.0 29692.0 20437.0 16676.0 13537.0 21341.0 30885.0		centage_Cha	nge NaN 015 199 933 010 833 355 439				
[306 r	rows	x 12 co	lumn	;1					
[306 rows x 12 columns]									

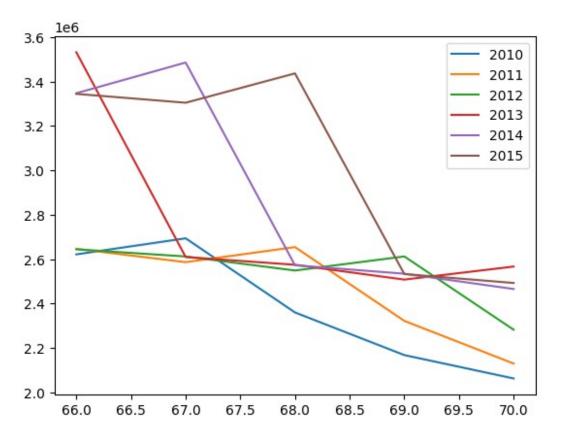
This will create two new columns in the DataFrame: change, which is the difference between the close and open prices, and percent_change, which is the percentage change in the close price relative to the previous row. Note that the first row of the percent_change column will be NaN, since there is no 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 df			olumns: nge"]=0	lf[c].dif	f()						
66 67 68 69 70	SEX 0 0 0 0	AGE 66 67 68 69 70	CENSUS	2010P0P 2639141 2649365 2323672 2142324 2043121	ES	TIMATE	263 264 232 214	2010 9209 9426 3736 2381 3178	POPES [*]	TIMATE2010 262133 269370 2359810 2167830 206257	5 7 6 9
	POPES	STIMA	TE2011	P0PESTI	MAT	E2012	P0PE	STIM	ATE2013	P0PESTII	MATE2014
\ 66		20	646364		26	44042			3531521		3347060
67		2.	586302		26	12002			2609390		3485241
68		20	654106		25	48669		:	2574501		2572359
69		2:	321925		26	12071			2507744		2534295
70		2	129707		22	82127		:	2566440		2465438
66 67 68 69 70	POPES	3: 3: 3: 2:	TE2015 344134 304187 436357 532747 492490	6 6	ge laN 0.0 0.0	AGEch	ange NaN 1.0 1.0 1.0	CEN		POPchange NaN 10224.0 -325693.0 -181348.0 -99203.0	\
DOD			BASE201 11chang	Ochange	P0	PESTIM	ATE20	10ch	ange		
66	E31111/	AIEZU.	TICHANG	e \ NaN					NaN		
NaN 67				10217.0				723	72.0		-
68	62.0		-3	25690.0			-	3338	91.0		
69	04.0		- 1	.81355.0			-	1919	86.0		-
70	181.0 218.0		-	99203.0			-	1052	53.0		-
DOD			TE2012c	-	OPE	STIMAT	E2013	chan	ge		
66		41EZU.	14chang	e \ NaN				N	aN		
NaN 67 138	181.0		-32	040.0			-92	2131	. 0		

```
68
                  -63333.0
                                          -34889.0
912882.0
69
                   63402.0
                                          -66757.0
38064.0
70
                 -329944.0
                                          58696.0
68857.0
    POPESTIMATE2015change
66
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]
     SEX AGE CENSUS2010POP ESTIMATESBASE2010
                                                 POPESTIMATE2010 \
0
      0
            0
                     3944153
                                        3944160
                                                         3951330
102
    1
            0
                     2014276
                                        2014278
                                                         2018420
     POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013
POPESTIMATE2014 \
             3963087
                                               3931141
                              3926540
3949775
                              2007717
102
             2028427
                                               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:]
     POPESTIMATE2010 POPESTIMATE2011 POPESTIMATE2012
POPESTIMATE2013 \
             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,:]
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.

I. Plot the gender ratio versus age.								
df=pd.rea	ad_csv(("nc-est20	915 - ag	esex-res.	csv")			
SEX 0 0 1 0 2 0 3 0 4 0	AGE 0 1 2 3 4	39 ² 409 41	44153 78070 96929 19040 63170	ESTIMATE	3944160 3978090 4096939 4119051 4063186	POPES1	3951330 3957888 4090862 4111920 4077551	\
301 2 302 2 303 2 304 2 305 2	97 98 99 100 999		53582 36641 26193 44202 64212		53605 36675 26214 44246 156969328		54118 37532 26074 45058 157258820	
POPE POPESTIMA 0 3949775 1 3949776 2 3959664 3 4007079 4 4005716	39 39 39 41		OPESTI	MATE2012 3926540 3977939 3980095 3983157 4112849	3	TE2013 931141 942872 992720 992734 994449		
301 62779 302 46208 303 32517 304 58008		57159 40116 27030 47556		59533 42857 29320 50661		61255 44359 31112 53902		

```
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]</pre>
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
               2014276
                                   2014278
                                                     2018420
2028427
               3978070
                                   3978090
                                                     3957888
    0
3966551
```

1	2030853	2030861	2020332
2025522 2 0	4096929	4096939	4090862
3971565	4030323	4030333	4030002
98 1	9259	9263	9505
10554	22266	22200	22170
99 0 33636	32266	32289	32178
1	6073	6075	6104
6606 100 0	E2264	E2/12	E4410
57702	53364	53412	54410
1	9162	9166	9352
10146			
		POPESTIMATE2013	POPESTIMATE2014
POPESTIMATE AGE SEX	2015		
0 0	3926540	3931141	3949775
3978038 1	2007717	2009528	2020326
2035134	2007717	2009320	2020320
1 0	3977939	3942872	3949776
3968564 1	2035460	2015470	2018401
2029295			
2 0 3966583	3980095	3992720	3959664
00 1	11650	12656	12510
98 1 14719	11652	12656	13518
99 0	36779	39271	41468
43641 1	7459	8159	8951
9577			
100 0 76974	61821	66189	71626
1	11160	12287	13618
15088			
[202 rows x	8 columns]		
for i in ra	nge(0,len(age)	2):	
for j i	n range(2,8):		
		loc[(i,j)]/d.iloc	[(i+ <mark>1</mark> ,j)]
u			

CENSU POPESTIMATE201 AGE SEX		ESTIMATESBASE2010	POPESTIMATE2010
0 0 1.953774e+00	3944153	3944160	1.957635e+00
1 2.028427e+06	2014276	2014278	2.018420e+06
1 0 1.958286e+00	3978070	3978090	1.959029e+00
1 2.025522e+06	2030853	2030861	2.020332e+06
2 0 1.959358e+00	4096929	4096939	1.958583e+00
98 1 1.055400e+04	9259	9263	9.505000e+03
99 0 5.091735e+00	32266	32289	5.271625e+00
1 6.606000e+03	6073	6075	6.104000e+03
100 0 5.687167e+00	53364	53412	5.818007e+00
1 1.014600e+04	9162	9166	9.352000e+03
POPES POPESTIMATE201 AGE SEX	STIMATE2012 15	POPESTIMATE2013	POPESTIMATE2014
0 0 1. 1.954681e+00	.955724e+00	1.956251e+00	1.955019e+00
	.007717e+06	2.009528e+06	2.020326e+06
	.954319e+00	1.956304e+00	1.956884e+00
	.035460e+06	2.015470e+06	2.018401e+06
2 0 1.	.958617e+00	1.954677e+00	1.956672e+00
1.957256e+00 			
	. 165200e+04	1.265600e+04	1.351800e+04
	.930822e+00	4.813212e+00	4.632778e+00
	.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
           1.116000e+04 1.228700e+04 1.361800e+04
   1
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["SEXX"]=v
d
        CENSUS2010POP ESTIMATESBASE2010 POPESTIMATE2010
POPESTIMATE2011 \
AGE SEX
0 0
              3944153
                                3944160
                                           1.957635e+00
1.953774e+00
              2014276
                                2014278
                                           2.018420e+06
   1
2.028427e+06
              3978070
                                3978090
                                           1.959029e+00
1.958286e+00
              2030853
                                           2.020332e+06
   1
                                2030861
2.025522e+06
                                4096939
                                           1.958583e+00
2 0
              4096929
1.959358e+00
98 1
                 9259
                                           9.505000e+03
                                   9263
1.055400e+04
99 0
                32266
                                  32289
                                           5.271625e+00
5.091735e+00
   1
                 6073
                                   6075
                                           6.104000e+03
6.606000e+03
                                           5.818007e+00
100 \ 0
                53364
                                  53412
5.687167e+00
                                   9166
                                           9.352000e+03
   1
                 9162
1.014600e+04
        POPESTIMATE2012 POPESTIMATE2013
                                         POPESTIMATE2014
POPESTIMATE2015 \
AGE SEX
0 0 1.955724e+00
                           1.956251e+00
                                           1.955019e+00
1.954681e+00
                           2.009528e+06
                                           2.020326e+06
           2.007717e+06
2.035134e+06
```

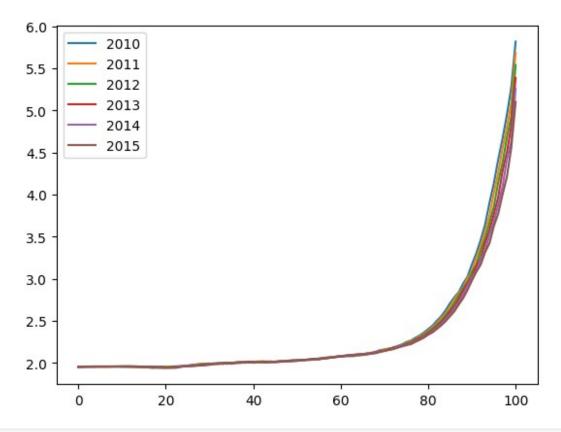
```
0
                              1.956304e+00
                                                1.956884e+00
            1.954319e+00
1.955637e+00
            2.035460e+06
                              2.015470e+06
                                                2.018401e+06
    1
2.029295e+06
            1.958617e+00
                              1.954677e+00
                                                1.956672e+00
1.957256e+00
. . .
98 1
                              1.265600e+04
                                                1.351800e+04
            1.165200e+04
1.471900e+04
99 0
            4.930822e+00
                              4.813212e+00
                                                4.632778e+00
4.556855e+00
            7.459000e+03
                              8.159000e+03
                                                8.951000e+03
    1
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
         SEXX
AGE SEX
    0
            0
    1
            1
1
    1
            1
2
    0
            0
98
   1
            1
99
   0
            0
            1
    1
100 0
            0
            1
[202 rows \times 9 columns]
d=d[d["SEXX"]<1]</pre>
         CENSUS2010POP ESTIMATESBASE2010 POPESTIMATE2010
POPESTIMATE2011 \
AGE SEX
0 0
               3944153
                                    3944160
                                                    1.957635
1.953774
               3978070
                                    3978090
                                                     1.959029
    0
1.958286
               4096929
                                    4096939
                                                    1.958583
1.959358
               4119040
                                    4119051
                                                    1.956872
   0
1.958878
```

4 0	4063170	4063186	1.956305
1.957104			
	05222	05200	
96 0 4.254485	95223	95288	4.411997
97 0	68138	68168	4.662809
4.546944 98 0	45900	45938	4.948659
4.801023			
99 0 5.091735	32266	32289	5.271625
100 0	53364	53412	5.818007
5.687167			
DODECTIM	POPESTIMATE2012	POPESTIMATE2013	POPESTIMATE2014
POPESTIMA AGE SEX	HIEZUID /		
0 0	1.955724	1.956251	1.955019
1.954681			
1 0 1.955637	1.954319	1.956304	1.956884
2 0	1.958617	1.954677	1.956672
1.957256 3 0	1.959705	1.958974	1.955058
1.957045			
4 0 1.955342	1.959144	1.959963	1.959248
96 0	4.142240	3.966077	3.879979
3.771275			
97 0 4.000390	4.378142	4.276019	4.091039
98 0	4.678081	4.504978	4.418257
4.211631 99 0	4.930822	4.813212	4.632778
4.556855			
100 0 5.101670	5.539516	5.386913	5.259656
	CEVV		
AGE SEX	SEXX		
$egin{array}{ccc} 0 & 0 \\ 1 & 0 \end{array}$	0 0		
2 0	0		
3 0 4 0	0 0		

```
96 0
            0
97 0
            0
98 0
            0
99 0
            0
            0
100 0
[101 rows x 9 columns]
a=np.unique(age)
d["AGEE"]=a
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
               3944153
                                  3944160
                                                   1.957635
1.953774
                                  3978090
                                                   1.959029
               3978070
1.958286
               4096929
                                  4096939
                                                   1.958583
   0
1.959358
               4119040
                                  4119051
                                                   1.956872
1.958878
               4063170
                                  4063186
                                                   1.956305
1.957104
96 0
                 95223
                                                   4.411997
                                    95288
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
                1.954319
                                  1.956304
                                                    1.956884
1.955637
    0
                 1.958617
                                  1.954677
                                                    1.956672
1.957256
                1.959705
                                  1.958974
3
                                                    1.955058
1.957045
                 1.959144
                                  1.959963
                                                    1.959248
4 0
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
         SEXX AGEE
AGE SEX
            0
                   0
0
    0
    0
            0
                   1
1
2
                   2
    0
            0
                   3
3
    0
            0
                   4
4
            0
    0
96
   0
            0
                 96
97
            0
                 97
   0
98
            0
                 98
   0
99
    0
            0
                 99
            0
100 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
```