lec06and07

September 7, 2021

[1]:
```
from datascience import *
import numpy as np

%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
```

0.1 Lecture 6 and 7

0.2 Census

[2]:
```
full = Table.read_table('nc-est2014-agesex-res.csv')
full
```

[2]:
```
SEX  | AGE  | CENSUS2010POP | ESTIMATESBASE2010 | POPESTIMATE2010 |
POPESTIMATE2011 | POPESTIMATE2012 | POPESTIMATE2013 | POPESTIMATE2014
0    | 0    | 3944153       | 3944160           | 3951330         | 3963071
| 3926665         | 3945610         | 3948350
0    | 1    | 3978070       | 3978090           | 3957888         | 3966510
| 3978006         | 3943077         | 3962123
0    | 2    | 4096929       | 4096939           | 4090862         | 3971573
| 3979952         | 3992690         | 3957772
0    | 3    | 4119040       | 4119051           | 4111920         | 4102501
| 3983049         | 3992425         | 4005190
0    | 4    | 4063170       | 4063186           | 4077552         | 4122303
| 4112638         | 3994047         | 4003448
0    | 5    | 4056858       | 4056872           | 4064653         | 4087713
| 4132210         | 4123408         | 4004858
0    | 6    | 4066381       | 4066412           | 4073013         | 4074979
| 4097780         | 4143094         | 4134352
0    | 7    | 4030579       | 4030594           | 4043047         | 4083240
| 4084964         | 4108615         | 4154000
0    | 8    | 4046486       | 4046497           | 4025604         | 4053206
| 4093213         | 4095827         | 4119524
0    | 9    | 4148353       | 4148369           | 4125415         | 4035769
| 4063193         | 4104133         | 4106832
… (296 rows omitted)
```

[3]:
```
# Keep only the columns we care about
partial = full.select('SEX', 'AGE', 'POPESTIMATE2010', 'POPESTIMATE2014')
partial
```

[3]:
```
SEX  | AGE  | POPESTIMATE2010 | POPESTIMATE2014
0    | 0    | 3951330         | 3948350
0    | 1    | 3957888         | 3962123
0    | 2    | 4090862         | 3957772
0    | 3    | 4111920         | 4005190
0    | 4    | 4077552         | 4003448
0    | 5    | 4064653         | 4004858
0    | 6    | 4073013         | 4134352
0    | 7    | 4043047         | 4154000
0    | 8    | 4025604         | 4119524
0    | 9    | 4125415         | 4106832
… (296 rows omitted)
```

[4]:
```
# Make things easier to read
simple = partial.relabeled(2, '2010').relabeled(3, '2014')
simple
```

[4]:
```
SEX  | AGE  | 2010    | 2014
0    | 0    | 3951330 | 3948350
0    | 1    | 3957888 | 3962123
0    | 2    | 4090862 | 3957772
0    | 3    | 4111920 | 4005190
0    | 4    | 4077552 | 4003448
0    | 5    | 4064653 | 4004858
0    | 6    | 4073013 | 4134352
0    | 7    | 4043047 | 4154000
0    | 8    | 4025604 | 4119524
0    | 9    | 4125415 | 4106832
… (296 rows omitted)
```

[5]:
```
# Sort by age
simple.sort('AGE')
```

[5]:
```
SEX  | AGE  | 2010    | 2014
0    | 0    | 3951330 | 3948350
1    | 0    | 2018420 | 2017857
2    | 0    | 1932910 | 1930493
0    | 1    | 3957888 | 3962123
1    | 1    | 2020332 | 2023253
2    | 1    | 1937556 | 1938870
0    | 2    | 4090862 | 3957772
1    | 2    | 2088685 | 2022502
2    | 2    | 2002177 | 1935270
```

```
0    | 3    | 4111920 | 4005190
… (296 rows omitted)
```

[6]:
```
# Sort by age (another way)
simple.sort('AGE', descending=True)
```

[6]:
```
SEX  | AGE  | 2010      | 2014
0    | 999  | 309347057 | 318857056
1    | 999  | 152089484 | 156936487
2    | 999  | 157257573 | 161920569
0    | 100  | 54409     | 72197
1    | 100  | 9351      | 13729
2    | 100  | 45058     | 58468
0    | 99   | 32178     | 41828
1    | 99   | 6104      | 9037
2    | 99   | 26074     | 32791
0    | 98   | 47037     | 60185
… (296 rows omitted)
```

0.3 Line Plots

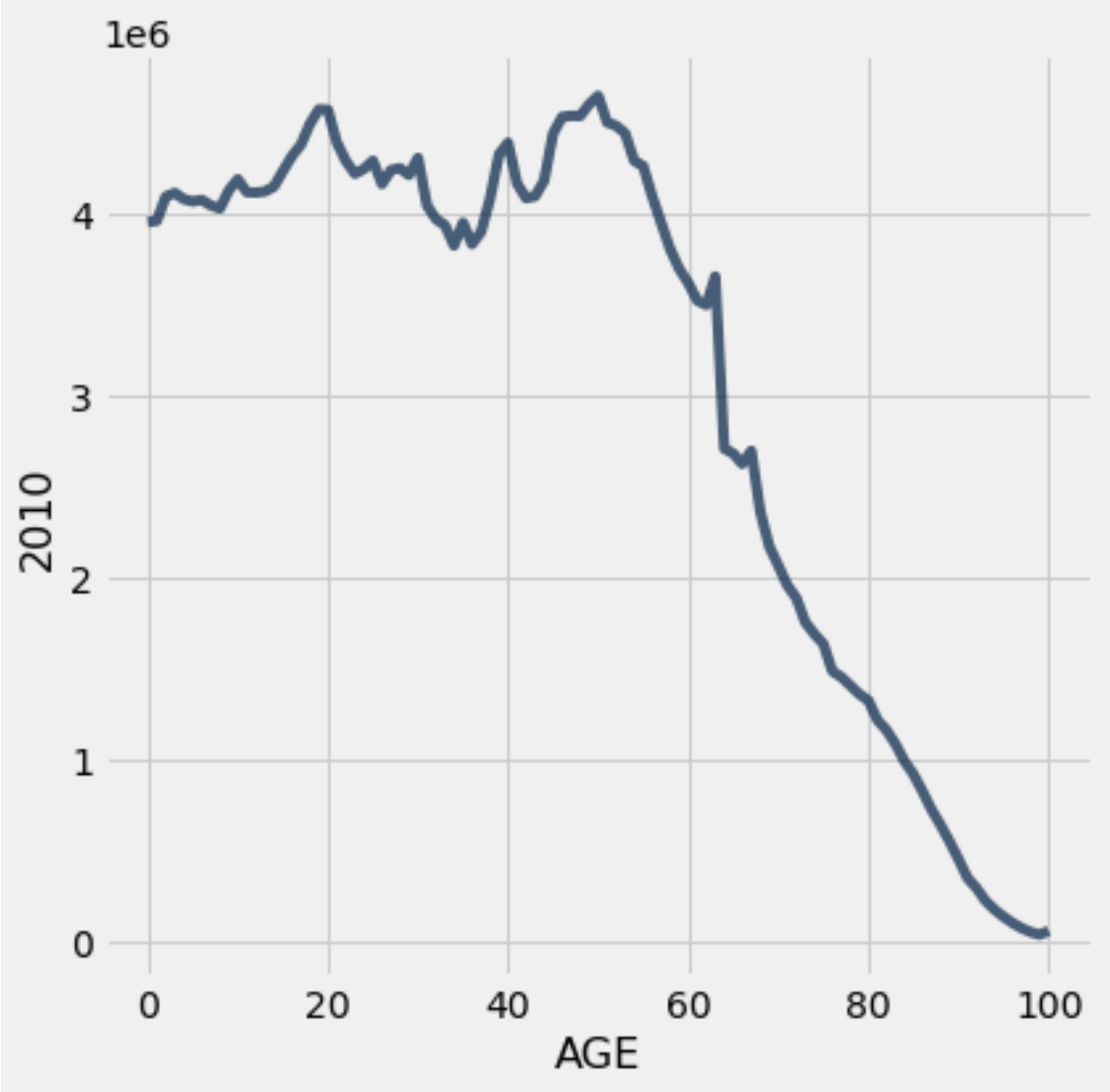
[7]:
```
# Remove the age totals
no_999 = simple.where('AGE', are.below(999))
```

[8]:
```
# Remove male and female (keep only combined)
everyone = no_999.where('SEX', 0).drop('SEX')
```

[9]:
```
everyone
```

[9]:
```
AGE  | 2010    | 2014
0    | 3951330 | 3948350
1    | 3957888 | 3962123
2    | 4090862 | 3957772
3    | 4111920 | 4005190
4    | 4077552 | 4003448
5    | 4064653 | 4004858
6    | 4073013 | 4134352
7    | 4043047 | 4154000
8    | 4025604 | 4119524
9    | 4125415 | 4106832
… (91 rows omitted)
```

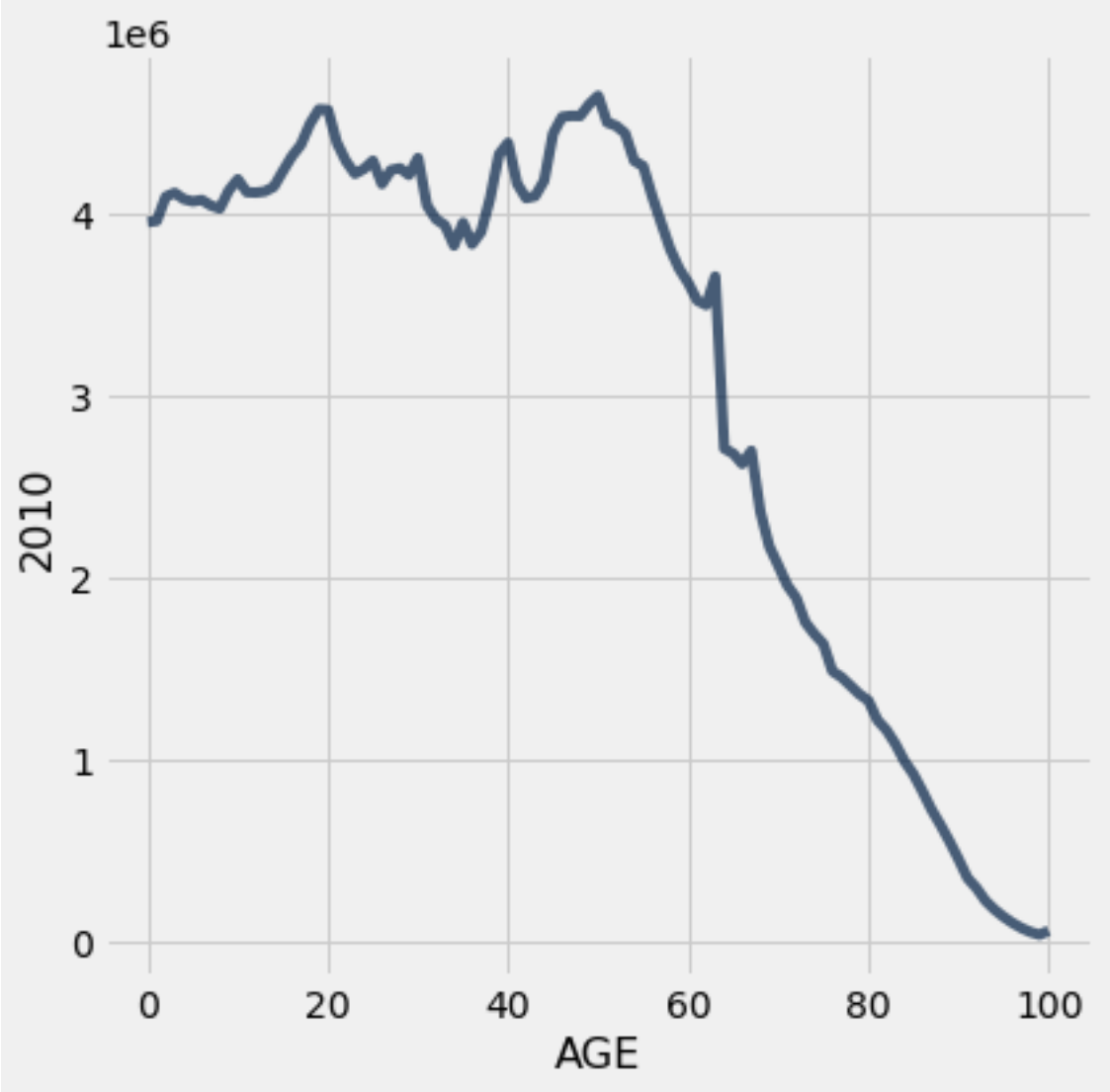
[10]:
```
everyone.plot('AGE', '2010')
```



[11]:
```
# ^^ That plot should be labeled! Here are 3 ways to label it:
```

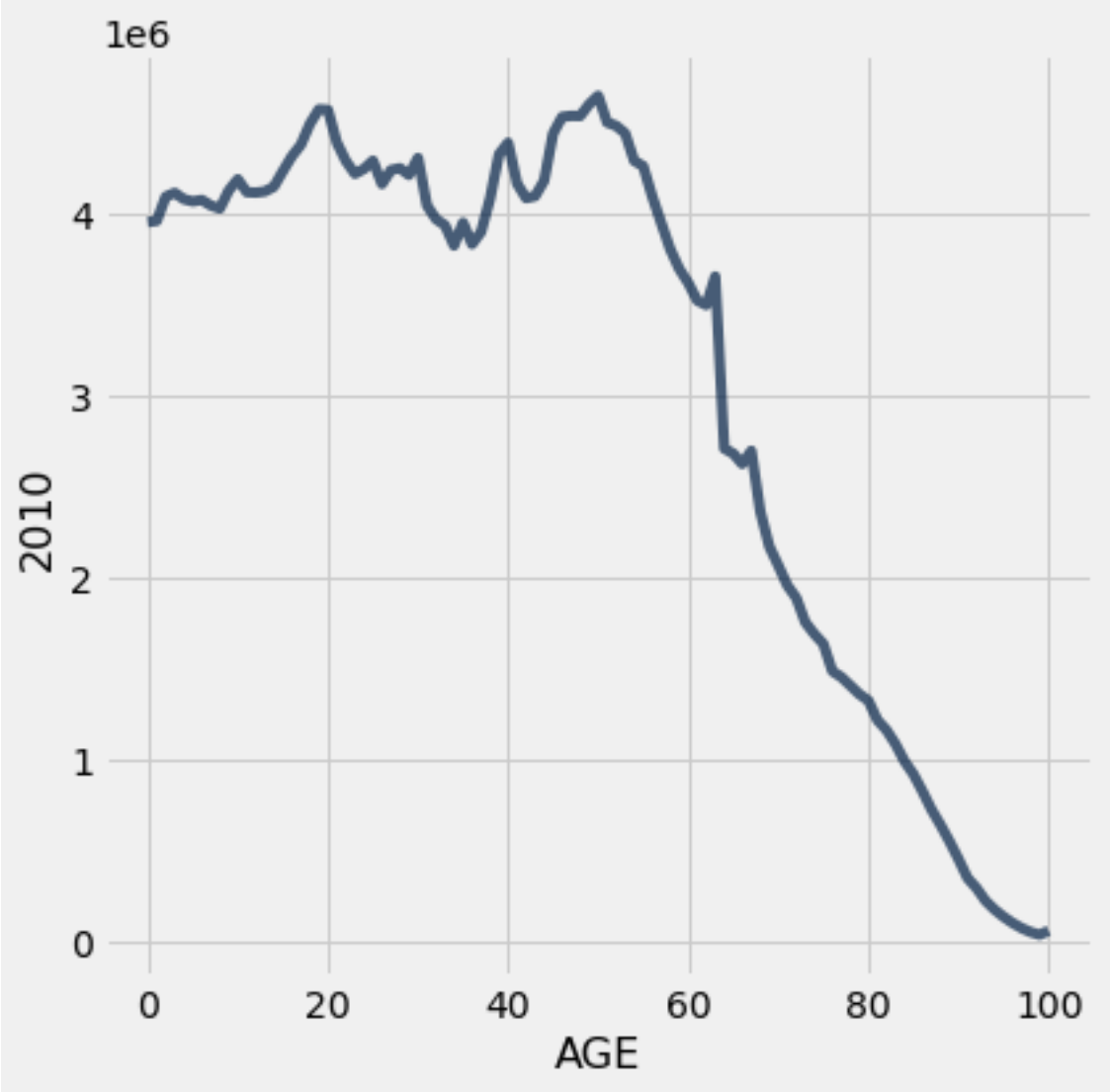
[12]:
```
# US Population  <--- Just add a comment

everyone.plot('AGE', '2010')
```

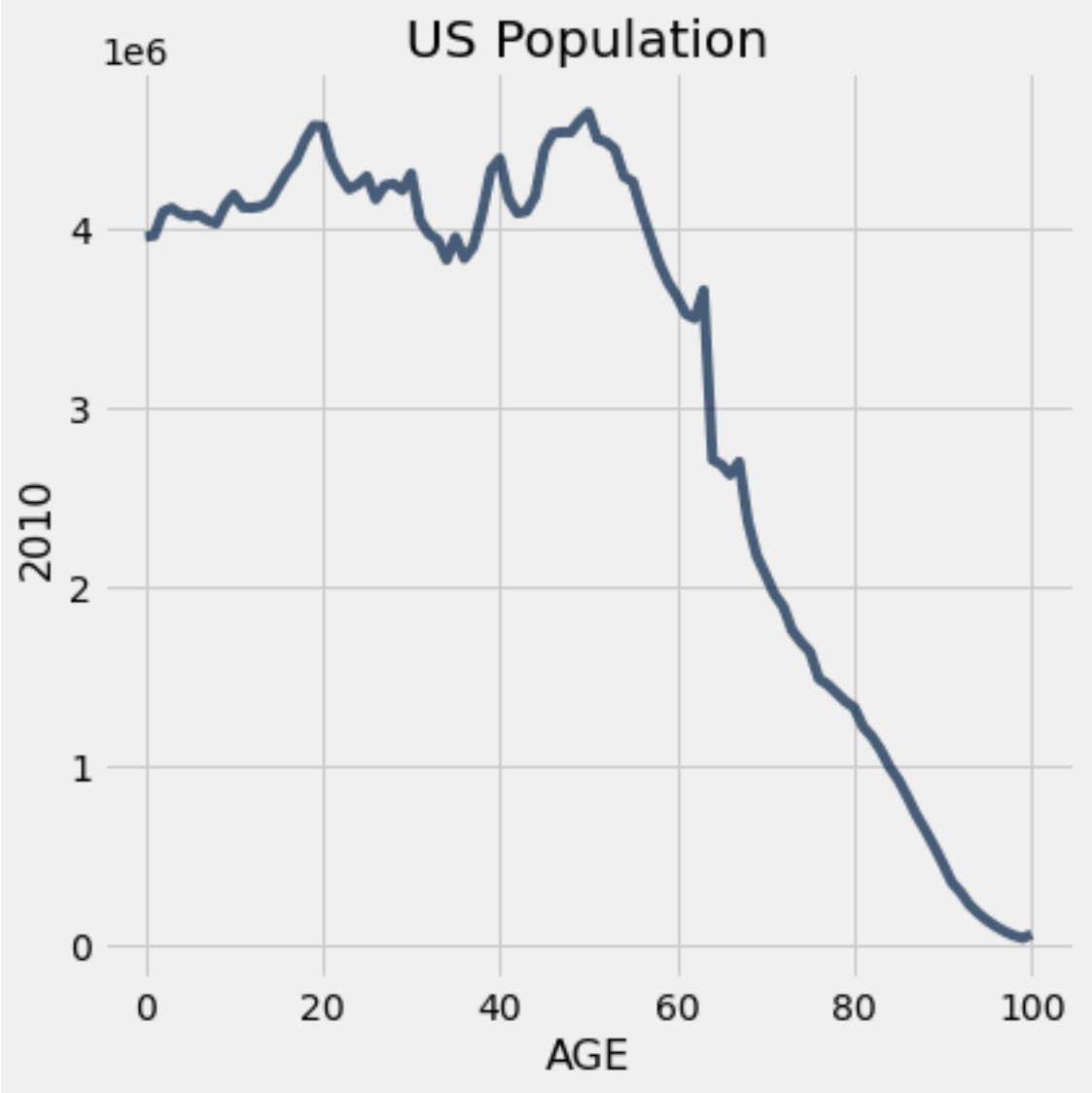


[13]:
```
everyone.plot('AGE', '2010')
print('US Population')  # <--- Print out what it is
```

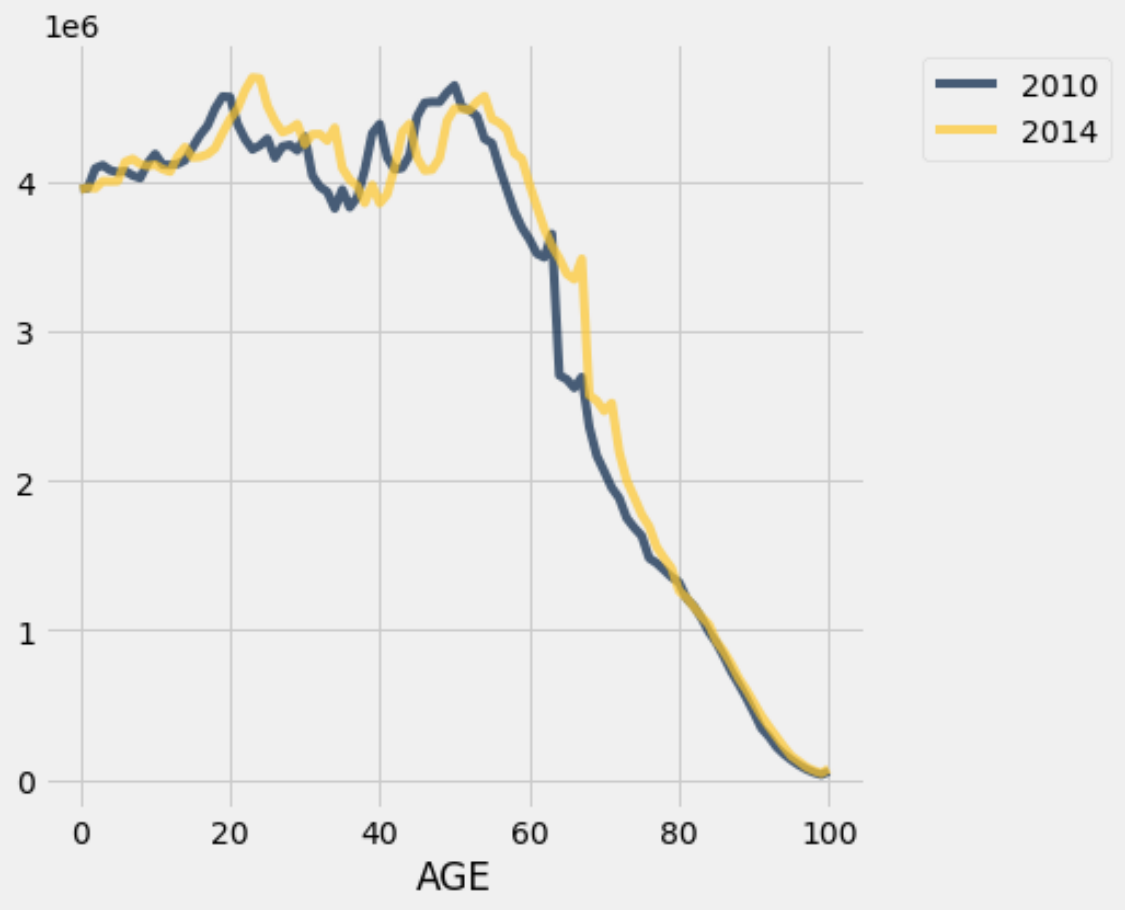
```
US Population
```



[14]:
```
everyone.plot('AGE', '2010')
plots.title('US Population');    # <--- OPTIONAL; not needed for Data 8
```



[15]:
```
# Age distribution for two different years
everyone.plot('AGE')
```



0.4 Males and Females in 2014

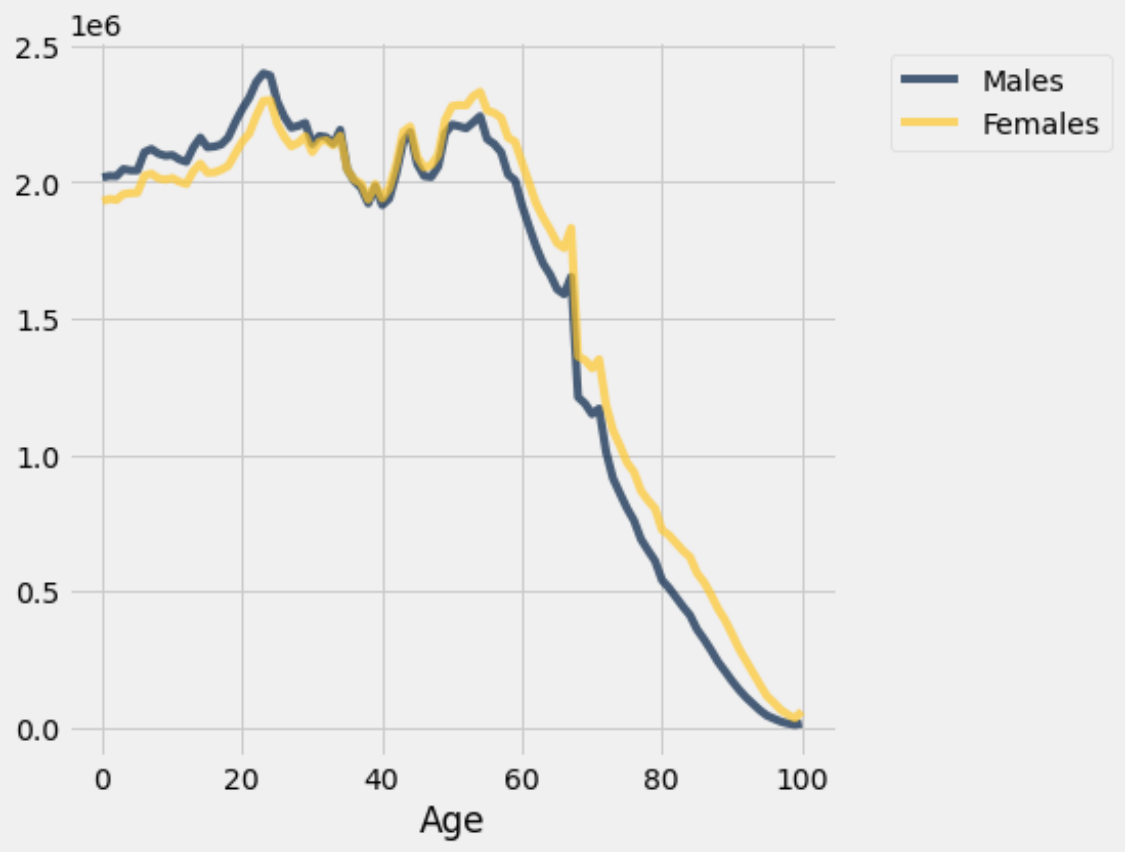
[16]:
```
# Let's compare male and female counts per age
males = no_999.where('SEX', 1).drop('SEX')
females = no_999.where('SEX', 2).drop('SEX')
```

[17]:
```
pop_2014 = Table().with_columns(
    'Age', males.column('AGE'),
    'Males', males.column('2014'),
    'Females', females.column('2014')
)
pop_2014
```

[17]:
```
Age  | Males   | Females
0    | 2017857 | 1930493
1    | 2023253 | 1938870
2    | 2022502 | 1935270
3    | 2048618 | 1956572
```

```
4    | 2043498 | 1959950
5    | 2043467 | 1961391
6    | 2110328 | 2024024
7    | 2122240 | 2031760
8    | 2105122 | 2014402
9    | 2097272 | 2009560
… (91 rows omitted)
```

[18]:
```
pop_2014.plot('Age')
```



[19]:
```
# Calculate the percent female for each age
total = pop_2014.column('Males') + pop_2014.column('Females')
pct_female = pop_2014.column('Females') / total * 100
pct_female
```

[19]:
```
array([48.89366444, 48.93512897, 48.89796583, 48.85091594, 48.95654945,
       48.9752945 , 48.95625723, 48.91092922, 48.89890191, 48.93212091,
       48.9734048 , 48.99969062, 48.99606517, 48.95633512, 48.86619665,
       48.88170995, 48.86445062, 48.88986227, 48.76308397, 48.668799  ,
       48.63932932, 48.5330796 , 48.66269094, 48.92327135, 49.03933504,
```

```
       49.08996242, 49.16509171, 49.21162965, 49.28169646, 49.44899983,
       49.65375766, 49.75983547, 49.86565039, 49.93824999, 49.74770193,
       49.99251351, 50.05521355, 50.20280862, 50.18189092, 50.10049432,
       50.31587643, 50.47966604, 50.40624483, 50.42907187, 50.23118137,
       50.32445422, 50.37830234, 50.53327291, 50.51106084, 50.55818402,
       50.75941276, 50.86725098, 50.93664868, 51.06392595, 50.97417608,
       51.18857886, 51.29709649, 51.45934869, 51.62031101, 51.70400468,
       51.97408419, 52.08985538, 52.15439053, 52.36152155, 52.36785492,
       52.49779211, 52.53185996, 52.56760719, 52.90700545, 53.15391012,
       53.40547378, 53.59234336, 53.92943506, 54.39955529, 54.64201112,
       54.72838462, 55.24864161, 55.70223967, 56.17908816, 56.81601714,
       57.27838288, 57.96988345, 58.65447794, 59.42565786, 60.30973204,
       61.15934696, 62.26066267, 63.23984507, 64.42329756, 65.70984639,
       66.7833361 , 67.65867346, 69.00900342, 69.97202558, 71.58537422,
       72.74509305, 74.22514523, 75.54191399, 77.32159176, 78.39485512,
       80.9839744 ])
```

[20]:
```
# Round it to 3 so that it's easier to read
pct_female = np.round(pct_female, 3)
pct_female
```

[20]:
```
array([48.894, 48.935, 48.898, 48.851, 48.957, 48.975, 48.956, 48.911,
       48.899, 48.932, 48.973, 49.   , 48.996, 48.956, 48.866, 48.882,
       48.864, 48.89 , 48.763, 48.669, 48.639, 48.533, 48.663, 48.923,
       49.039, 49.09 , 49.165, 49.212, 49.282, 49.449, 49.654, 49.76 ,
       49.866, 49.938, 49.748, 49.993, 50.055, 50.203, 50.182, 50.1  ,
       50.316, 50.48 , 50.406, 50.429, 50.231, 50.324, 50.378, 50.533,
       50.511, 50.558, 50.759, 50.867, 50.937, 51.064, 50.974, 51.189,
       51.297, 51.459, 51.62 , 51.704, 51.974, 52.09 , 52.154, 52.362,
       52.368, 52.498, 52.532, 52.568, 52.907, 53.154, 53.405, 53.592,
       53.929, 54.4  , 54.642, 54.728, 55.249, 55.702, 56.179, 56.816,
       57.278, 57.97 , 58.654, 59.426, 60.31 , 61.159, 62.261, 63.24 ,
       64.423, 65.71 , 66.783, 67.659, 69.009, 69.972, 71.585, 72.745,
       74.225, 75.542, 77.322, 78.395, 80.984])
```

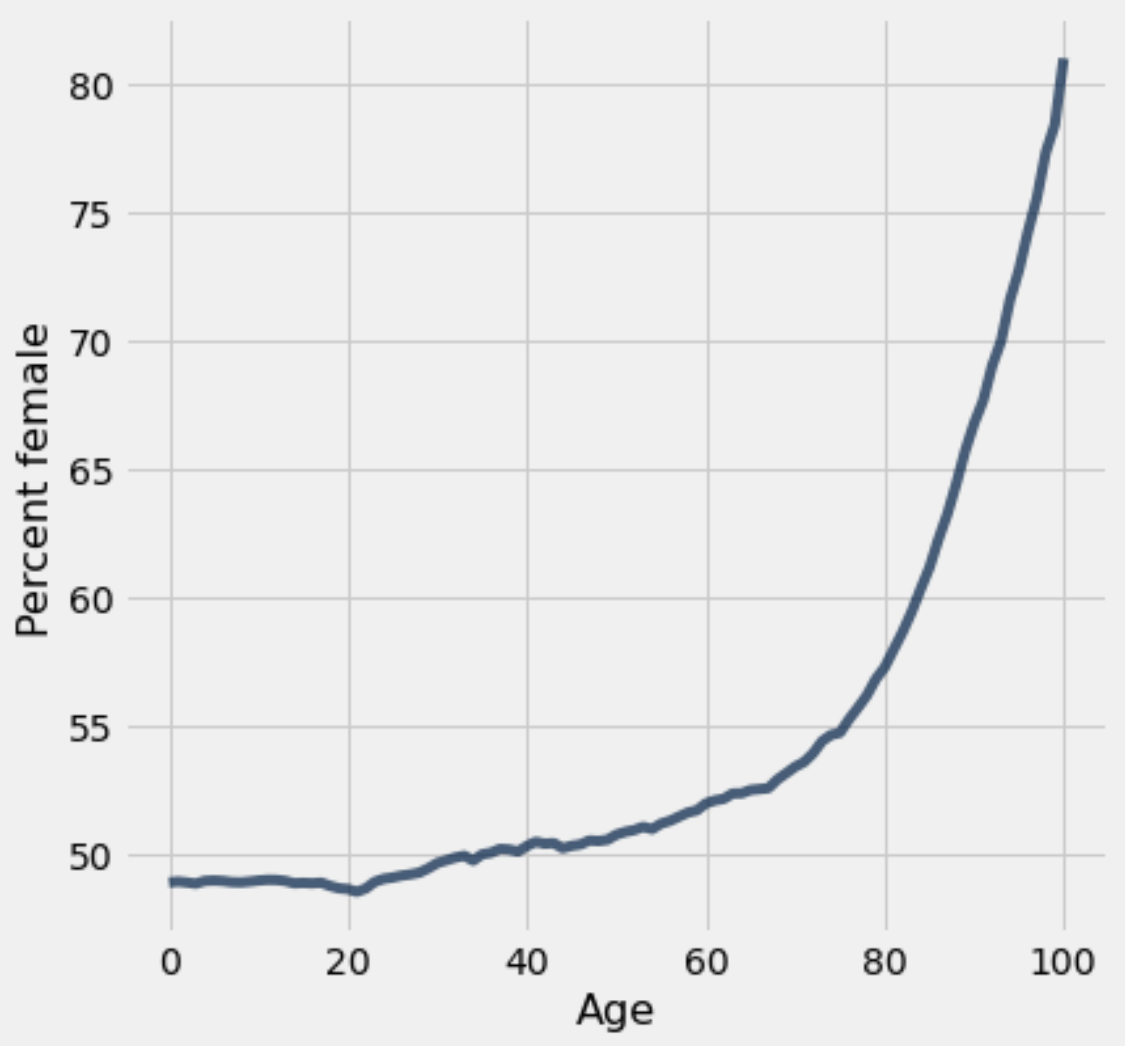
[21]:
```
# Add female percent to our table
pop_2014 = pop_2014.with_column('Percent female', pct_female)
pop_2014
```

[21]:

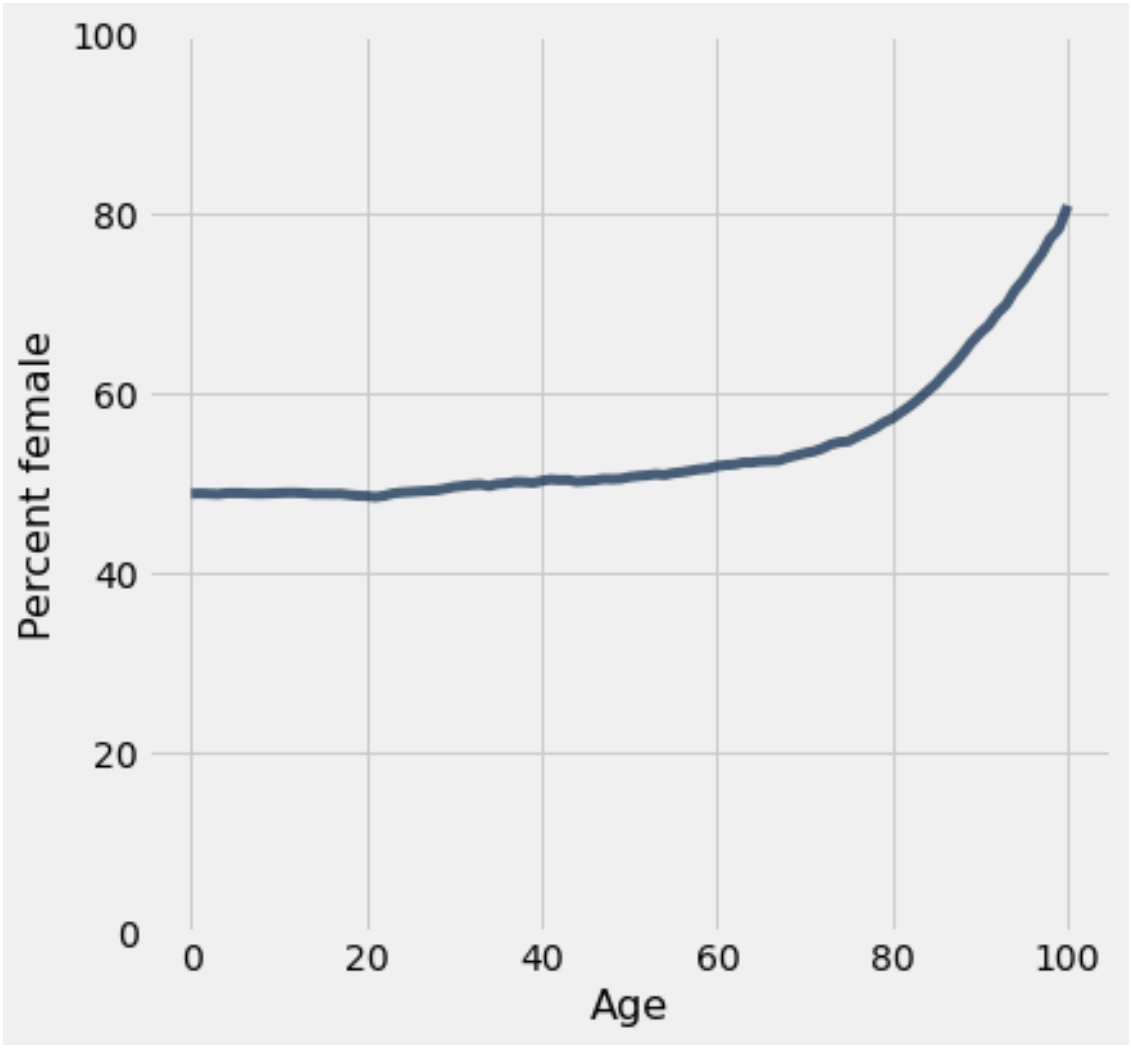
Age	Males	Females	Percent female
0	2017857	1930493	48.894
1	2023253	1938870	48.935
2	2022502	1935270	48.898
3	2048618	1956572	48.851
4	2043498	1959950	48.957
5	2043467	1961391	48.975
6	2110328	2024024	48.956

```
7    | 2122240 | 2031760 | 48.911
8    | 2105122 | 2014402 | 48.899
9    | 2097272 | 2009560 | 48.932
… (91 rows omitted)
```

[22]:
```
pop_2014.plot('Age', 'Percent female')
```



[23]:
```
# ^^ Look at the y-axis! Trend is not as dramatic as you might think
pop_2014.plot('Age', 'Percent female')
plots.ylim(0, 100);  # Optional for Data 8
```



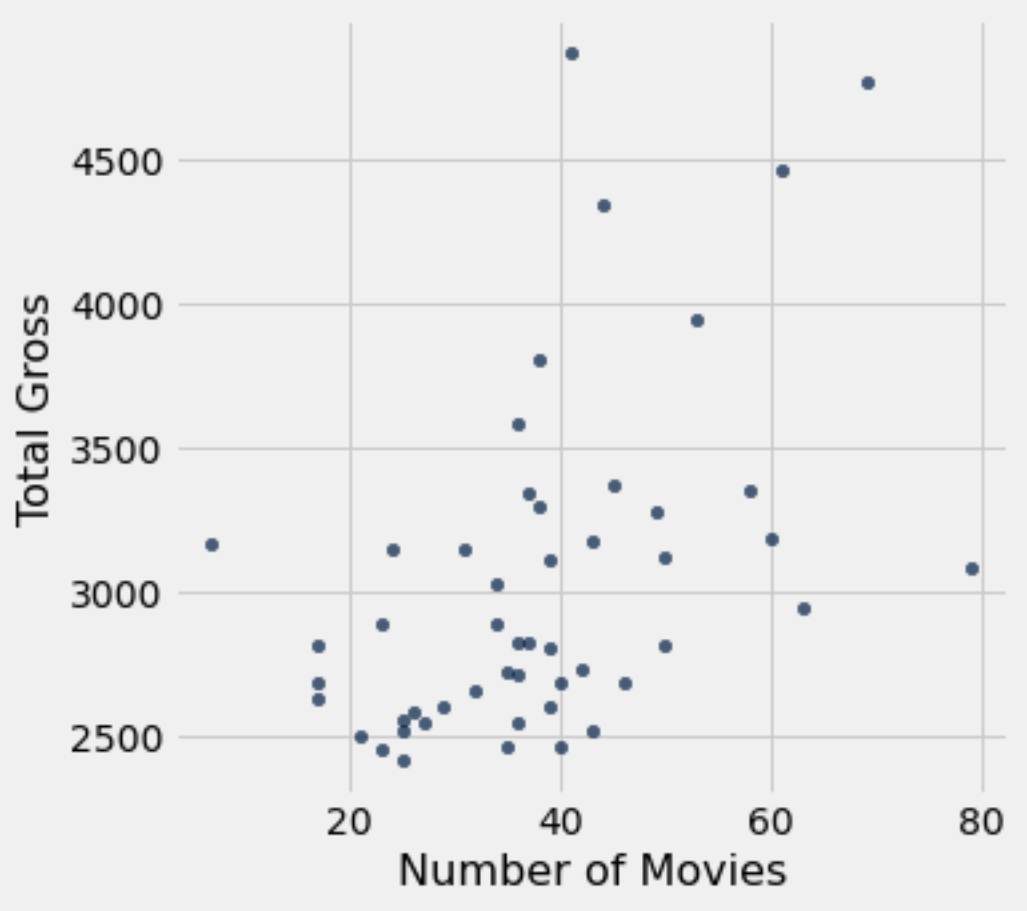
0.5 Scatter Plots

[24]:
```
# Actors and their highest grossing movies
actors = Table.read_table('actors.csv')
actors
```

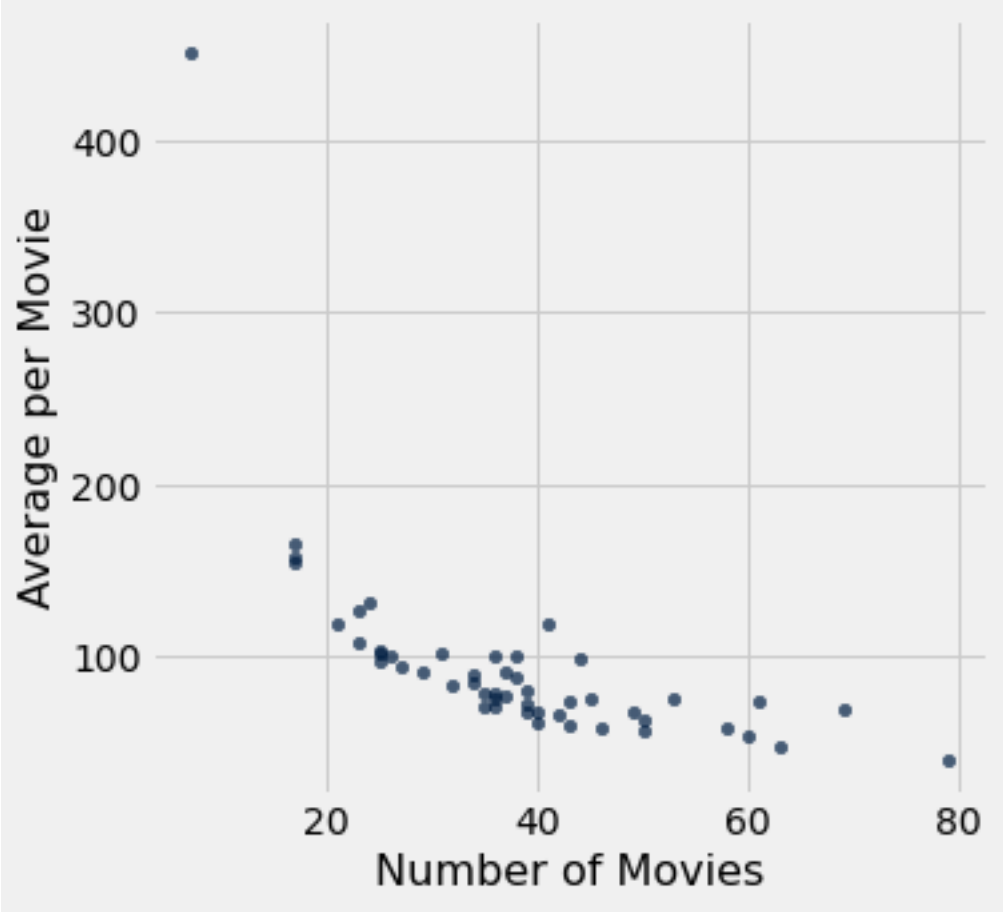
[24]:
```
Actor              | Total Gross | Number of Movies | Average per Movie | #1
Movie                    | Gross
Harrison Ford      | 4871.7      | 41               | 118.8             | Star
Wars: The Force Awakens | 936.7
Samuel L. Jackson  | 4772.8      | 69               | 69.2              | The
Avengers                 | 623.4
Morgan Freeman     | 4468.3      | 61               | 73.3              | The
Dark Knight              | 534.9
Tom Hanks          | 4340.8      | 44               | 98.7              | Toy
```

```
Story 3                  | 415
Robert Downey, Jr. | 3947.3      | 53              | 74.5              | The
Avengers                 | 623.4
Eddie Murphy       | 3810.4      | 38              | 100.3             | Shrek
2                        | 441.2
Tom Cruise         | 3587.2      | 36              | 99.6              | War of
the Worlds               | 234.3
Johnny Depp        | 3368.6      | 45              | 74.9              | Dead
Man's Chest              | 423.3
Michael Caine      | 3351.5      | 58              | 57.8              | The
Dark Knight              | 534.9
Scarlett Johansson | 3341.2      | 37              | 90.3              | The
Avengers                 | 623.4
… (40 rows omitted)
```

[25]:
```
actors.scatter('Number of Movies', 'Total Gross')
```



[26]:
```
actors.scatter('Number of Movies', 'Average per Movie')
```



[27]:
```
actors.where('Average per Movie', are.above(400))
```

[27]:
```
Actor           | Total Gross | Number of Movies | Average per Movie | #1 Movie
| Gross
Anthony Daniels | 3162.9      | 7                | 451.8             | Star
Wars: The Force Awakens | 936.7
```

0.6 Bar Charts

[28]:
```
# Highest grossing movies as of 2017
top_movies = Table.read_table('top_movies_2017.csv')
top_movies
```

[28]:
```
Title                          | Studio          | Gross     | Gross
(Adjusted) | Year
Gone with the Wind             | MGM             | 198676459 | 1796176700
| 1939
Star Wars                      | Fox             | 460998007 | 1583483200
| 1977
```

```
The Sound of Music               | Fox             | 158671368 | 1266072700
| 1965
E.T.: The Extra-Terrestrial      | Universal       | 435110554 | 1261085000
| 1982
Titanic                          | Paramount       | 658672302 | 1204368000
| 1997
The Ten Commandments             | Paramount       | 65500000  | 1164590000
| 1956
Jaws                             | Universal       | 260000000 | 1138620700
| 1975
Doctor Zhivago                   | MGM             | 111721910 | 1103564200
| 1965
The Exorcist                     | Warner Brothers | 232906145 | 983226600
| 1973
Snow White and the Seven Dwarves | Disney          | 184925486 | 969010000
| 1937
… (190 rows omitted)
```

[29]:
```
top10_adjusted = top_movies.take(np.arange(10))
top10_adjusted
```

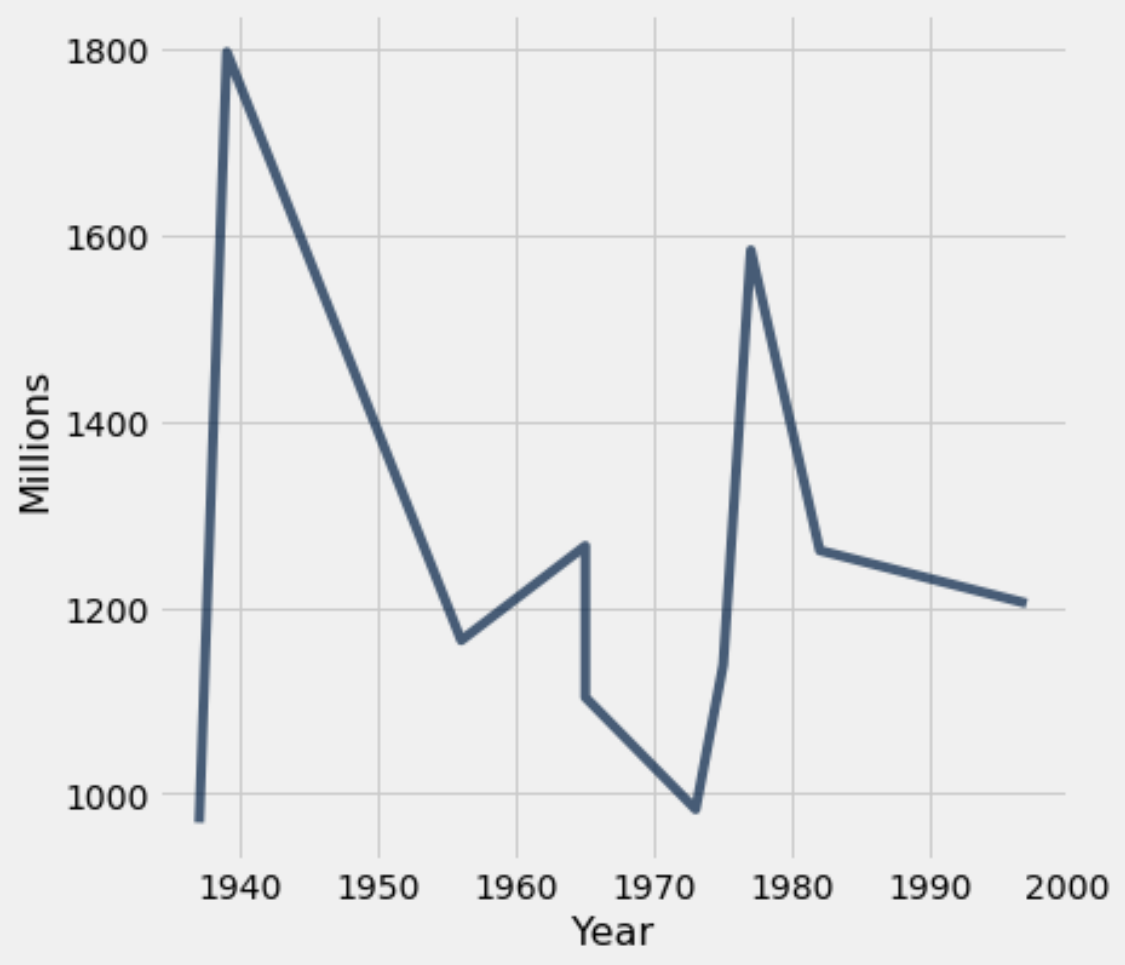
[29]:
```
Title                            | Studio          | Gross     | Gross
(Adjusted) | Year
Gone with the Wind               | MGM             | 198676459 | 1796176700
| 1939
Star Wars                        | Fox             | 460998007 | 1583483200
| 1977
The Sound of Music               | Fox             | 158671368 | 1266072700
| 1965
E.T.: The Extra-Terrestrial      | Universal       | 435110554 | 1261085000
| 1982
Titanic                          | Paramount       | 658672302 | 1204368000
| 1997
The Ten Commandments             | Paramount       | 65500000  | 1164590000
| 1956
Jaws                             | Universal       | 260000000 | 1138620700
| 1975
Doctor Zhivago                   | MGM             | 111721910 | 1103564200
| 1965
The Exorcist                     | Warner Brothers | 232906145 | 983226600
| 1973
Snow White and the Seven Dwarves | Disney          | 184925486 | 969010000
| 1937
```

[30]:
```
# Convert to millions of dollars for readability
millions = np.round(top10_adjusted.column('Gross (Adjusted)') / 1000000, 3)
top10_adjusted = top10_adjusted.with_column('Millions', millions)
```

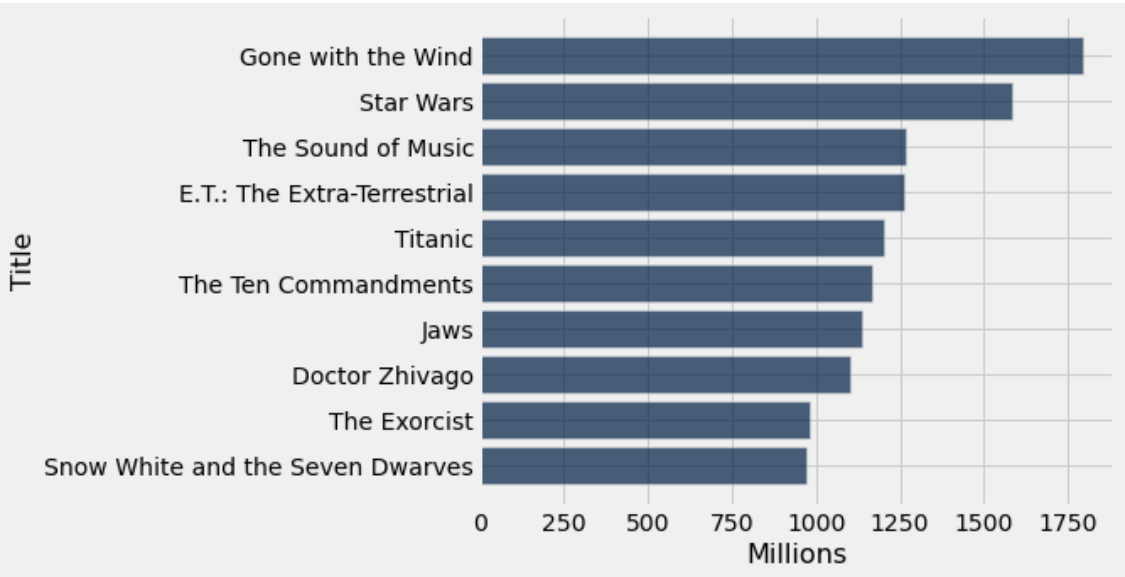
```
top10_adjusted
```

[30]:
```
Title                           | Studio          | Gross     | Gross
(Adjusted) | Year | Millions
Gone with the Wind              | MGM             | 198676459 | 1796176700
| 1939 | 1796.18
Star Wars                       | Fox             | 460998007 | 1583483200
| 1977 | 1583.48
The Sound of Music              | Fox             | 158671368 | 1266072700
| 1965 | 1266.07
E.T.: The Extra-Terrestrial     | Universal       | 435110554 | 1261085000
| 1982 | 1261.09
Titanic                         | Paramount       | 658672302 | 1204368000
| 1997 | 1204.37
The Ten Commandments            | Paramount       | 65500000  | 1164590000
| 1956 | 1164.59
Jaws                            | Universal       | 260000000 | 1138620700
| 1975 | 1138.62
Doctor Zhivago                  | MGM             | 111721910 | 1103564200
| 1965 | 1103.56
The Exorcist                    | Warner Brothers | 232906145 | 983226600
| 1973 | 983.227
Snow White and the Seven Dwarves | Disney          | 184925486 | 969010000
| 1937 | 969.01
```

[31]:
```
# A line plot doesn't make sense here: don't do this!
top10_adjusted.plot('Year', 'Millions')
```



[32]:
```
top10_adjusted.barh('Title', 'Millions')
```



[ ]:

[ ]: