Calibration notes for Endogenous Labor OG Model

Richard W. Evans and Jason DeBacker

February 18, 2020

There are two methods for calibrating the $\chi_{n,s}$ parameters of the OG model with endogenous labor. The first is to use the S initial-period labor supply Euler equations. The second is to use GMM to match average labor supply moments from the model steady-state to average labor supply moments from the data. The first method is orders of magnitude more simple, more intuitive, and more tractable. However, it is hard to get good data in the right format for the first method.

1 Method 1: Initial-period labor supply Euler equations

In the model from Chapter 7 of the textbook, the general form of the labor supply Euler equation is the following.

$$w_t \left(c_{s,t}\right)^{-\sigma} = \chi_s^n \left(\frac{b}{\tilde{l}}\right) \left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon-1} \left[1 - \left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}} \quad \forall s, t \tag{1}$$

We have already estimated the elliptic utility parameter values of b and υ, and we have calibrated the value for σ from other studies.

All the consumption $c_{s,t}$ and wage w_t values in the set of equations represented by (1) are given in consumption units (consumption is the numeraire good). So w_t represents the units of consumption that a worker is paid for each unit of labor supplied. And the implicit price of one unit of consumption is one consumption unit. This can be seen from the budget constraint.

$$c_{s,t} + b_{s+1,t+1} = (1 + r_t)b_{s,t} + w_t n_{s,t} \tag{2}$$

For this reason, $c_{s,t}$ also represents the total individual consumption expenditure of age-s household at in period t.

One may identify the χ_s^n via a method of moments estimation that uses Equation 1 as the set of moment conditions and data on consumption, wages, and labor supply. However, the estimates of χ_s^n will depend upon the units of measurement for wages and consumption. Thus we could only identify the χ_s^n from the model up to a scale. This is clearly seen if we rearrange Equation 1 to isolate χ_s^n on the left hand side:

$$\chi_s^n = \frac{w_t \left(c_{s,t}\right)^{-\sigma}}{\left(\frac{b}{\tilde{l}}\right)\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t \tag{3}$$

1.1 Scaling

We have assumed that $\tilde{l}=1$. Regardless of the value, the appearance of $\frac{n_{s,t}}{\tilde{l}}$ denominator on the right-hand-side of (3) is a unit-free percent of total time endowment. However, both consumption $c_{s,t}$ and wages w_t on the right-hand-side of (3) are in model units (consumption units).

To overcome this identification issue, consider a scaling that relates model units to data units. Call this parameter $factor_t$ and define it as:

$$factor_t = \frac{\bar{y}_t^{data}}{\bar{y}_t^{model}} \quad \forall t \implies \bar{y}_t^{data} = \bar{y}_t^{model} \times factor_t \tag{4}$$

In other words, $factor$ scales model units to data units for those variables measured in consumption units in the model and nominal amounts in the data. Thus we also have the relations

$$\begin{aligned} w_t^{model} \times factor_t &= w_t^{data} \\ c_{s,t}^{model} \times factor_t &= c_{s,t}^{data} \end{aligned} \tag{5}$$

With this, we can return to Equation 3. Let's write two versions of this equation. One that identifies the χ_s^n in the model and one that identifies it's data counterpart. To be clear, let χ_s^n be the model version and $\hat{\chi}_s^n$ represent the χ_s^n to be identified from the data. Thus we have:

$$\chi_s^n = \frac{w_t^{model} \left(c_{s,t}^{model}\right)^{-\sigma}}{\left(\frac{b}{\tilde{l}}\right)\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t \tag{6}$$

and

$$\hat{\chi}_s^n = \frac{w_t^{data} \left(c_{s,t}^{data}\right)^{-\sigma}}{\left(\frac{b}{\tilde{l}}\right)\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\tilde{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t \quad \text{where} \quad \hat{\chi}_s^n \equiv factor_t^{1-\sigma}\chi_s^n \tag{7}$$

Note that for brevity, we do not have *data* or *model* superscripts on the labor supply terms. This is because, as noted above, labor supply is always divided by labor endowment and so the ratio is in percentages both in the data and the model.

Now let's do some algebra with Equation 6:

$$\chi_s^n = \frac{w_t^{model}\left(c_{s,t}^{model}\right)^{-\sigma}}{\left(\frac{b}{\bar{l}}\right)\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t$$

$$\implies factor_t^{1-\sigma}\chi_s^n = \frac{factor_t^{1-\sigma}w_t^{model}\left(c_{s,t}^{model}\right)^{-\sigma}}{\left(\frac{b}{\bar{l}}\right)\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t$$

$$\implies factor_t^{1-\sigma}\chi_s^n = \frac{\left(factor_t w_t^{model}\right)\left(factor_t c_{s,t}^{model}\right)^{-\sigma}}{\left(\frac{b}{\bar{l}}\right)\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}} \quad \forall s,t \tag{8}$$

$$\implies factor_t^{1-\sigma}\chi_s^n = \underbrace{\frac{w_t^{data}\left(c_{s,t}^{data}\right)^{-\sigma}}{\left(\frac{b}{\bar{l}}\right)\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon-1}\left[1-\left(\frac{n_{s,t}}{\bar{l}}\right)^{\upsilon}\right]^{\frac{1-\upsilon}{\upsilon}}}}_{=\hat{\chi}_s^n} \quad \forall s,t$$

$$\implies \hat{\chi}_s^n \equiv factor_t^{1-\sigma}\chi_s^n \quad \forall s,t$$

Thus, by estimating $\hat{\chi}_s^n$ using the data on wages, consumption, and labor supply, one finds the model parameters up to a scale. That scale is function of the model scale parameter, $factor_t$.

1.2 Changes to Model Solution Algorithm

In theory, one wants to use the factor from model period t, where model period t corresponds to they year of your data (e.g., if the data are from 2017 and your initial period in the time path of your model is 2017, then you'd want determine the factor as $factor_0 = \frac{\bar{y}_{2017}^{data}}{\bar{y}_0^{model}}$). Because it depends on mean model income, the factor is endogenous and depends upon χ_s^n. Therefore, there is the need for some fixed point algorithm: guess a $factor$, use that to determine χ_s^n, solve the model and see if mean income in the data divided by mean income in the model returns the factor you guess, if not, update and do again. To compute the time path at each step in this fixed point algorithm would be very expensive. We therefore make a simplifying assumption. In particular, we assume that the factor is determined as the ratio of income in data from year t and from the model's steady state. That is,

$$factor_t = factor = \frac{\bar{y}_{2017}^{data}}{\bar{y}_{SS}^{model}} \ \forall t \tag{9}$$

While not a perfect mapping, this means that at each iteration of the fixed point algorithm that solves for the model $factor$ only the steady state needs to be computed.

Also note that the model income represents real, stationarized income. So growth and inflation are not affecting this measurement, which helps this approximation be more accurate.

The modification to the algorithm to solve the steady state in Chapter 7 need only be modified to include the guess of $factor$ as one of our outer-loop variables along with $\bar{r}$ in the steady-state computational approach. Given the guess for $\bar{r}$ and $factor$ one can use the relation in Equation 8 to transform the $\hat{\chi}_s^n$ into the model-scaled parameter. With these χ_s^n values, we can solve for the steady-state household decisions $\{\bar{n}_s\}_{s=1}^S$ and $\{\bar{b}_s\}_{s=2}^S$. From these decisions, we can compute the corresponding steady-state interest rate $\bar{r}$ and average household income in the model $\bar{y}$. We update $\bar{r}$ and $factor$ until the interest rate and factor implied by steady state equilibrium equal the guesses of $\bar{r}$ and $factor$ at that iteration. Once the steady state is solved and $factor$ is determined, then this same factor is applied over the time path, so no adjustment is needed for that solution method.

1.2.1 A Note About Initial the Initial Guess for *factor*

Because the difference between model units and real world units might be multiple orders of magnitude, it is helpful to get the initial guess for $factor$ near its true value. If one has trouble finding an initial guess that will not break the model, a good strategy is to compute the steady-state of the model assuming that $\chi_s^n = 1$ for all s. This means that only $\bar{r}$ is in the outer loop. Use the resulting $\bar{y}$ to derive an initial guess for $factor$ according to (4). This should get your initial guess in the neighborhood of the final value.

1.3 Data Issues

In this section, we discuss the detail behind data issues in the calibration.

1.3.1 Consumption data

The main equation (8) requires data on consumption expenditures by age $c_{s,t}$. For the United States, we use the Consumer Expenditures Survey (CEX) published by the U.S. Bureau of Labor Statistics. These data represent annual consumption expenditures by households in the survey. The survey is then used for many analyses and summaries of U.S. consumption patterns using characteristics of the underlying survey respondents and their corresponding population weights.

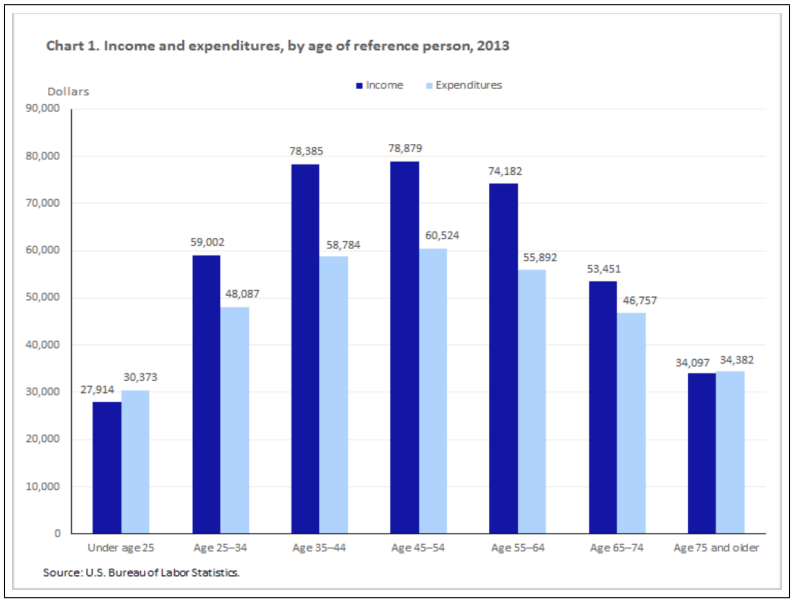
In Issue #5 of the LaborCalibrate repository, we detail two methods for calculating average consumption expenditure by arbitrary age bins.

1. The CEX has summary tables of consumption by broad age categories for each year that are precomputed (PDF and Excel). These summary data look like the light blue bars in Figure 1. One thing we could do to get consumption expenditure by arbitrary age—which is different and/or finer than the course age bins in the summary data—is to fit a curve to the summary data such that the curve has a similar shape and the average consumption expenditure across

the ages corresponding to the summary data equals the value of the summary data.

2. The more accurate thing we could do is to use the CEX survey microdata (PUMD) itself to calculate average consumption expenditure for each age group. There is a good paper here by Fernández-Villaverde and Krueger (2007). This paper calculates exactly the lifecycle consumption profiles by age that we are interested in using the CEX microdata. For our calibration, we would probably want to average data from two or three of the most recent surveys in order to get rid of any noise that comes with the fine granularity of one-year age bins.

Figure 1: CEX consumption by age summary



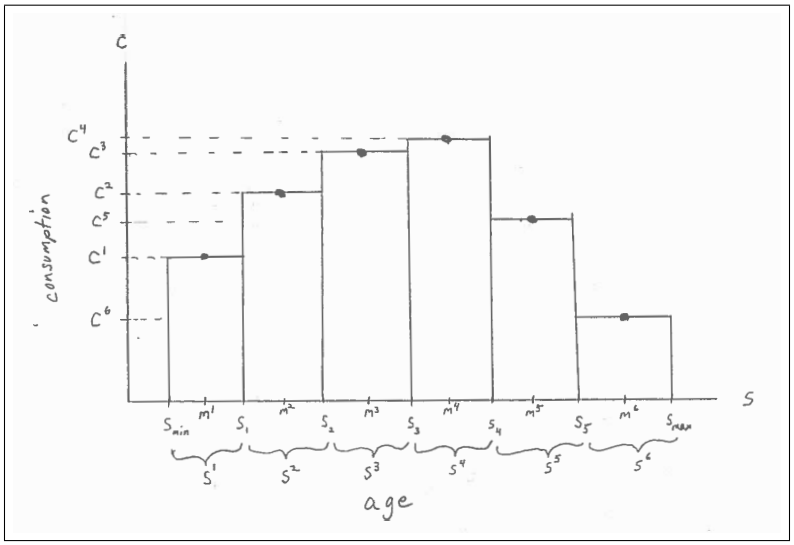
1.3.2 CEX interpolation

This section details a method for following approach (1) in the previous section for using CEX summary data by age to estimate a general function for consumption by age. This approach is nice because it bypasses dealing with the individual response-level data in the CEX survey microdata (PUMD). However, it requires more technical numerical care to estimate the function properly.

The guiding principle and assumption for this approach is that a continuous function $c(s)$ of consumption expenditure as a function of a continuous age variable s exists that has the same approximate shape as the CEX summary data. Figure 2 shows a generalization of the histogram of consumption expenditure in Figure 1. If the original data have I histogram bars, then the age bin cutoffis are given by the $I+1$ age values with subscripts $\{s_i\}_{i=0}^{I}$. An ordered age index for each of the bars as well as the average consumption expenditure for each age bin are given by I respective

age and consumption values with a superscript $\{s^i, c^i\}_{i=1}^I$. We have also included I midpoints of each of the age bins $\{m^i\}_{i=1}^I$ as a convenient reference age within each bin. With this notation, we can say that each age bin s^i is characterized by all ages between s_{i-1} and s_i, and c^i represents that average consumption of a household with a reference person with age between s_{i-1} and s_i.

Figure 2: Consumption expenditure by age summary generalization



As was mentioned at the beginning of this section, the average consumption values $\{c^i\}_{i=1}^I$ in Figure 2 are generated by an underlying continuous function of age $c(s)$. If one knows the continuous function $c(s)$ and the population density by age $f(s)$ such that $\int_s f(s)ds = 1$, one can solve for any arbitrary average consumption of an age bin s^i bounded by cutoffs s_{i-1} and s_i using the following expression,

$$c^i = \int_{s_{i-1}}^{s_i} \frac{f(s)c(s)}{F(s_i) - F(s_{i-1})} ds \tag{10}$$

where $F(s_i)$ is the cumulative distribution function (CDF) of the continuous density function $f(s_i)$.

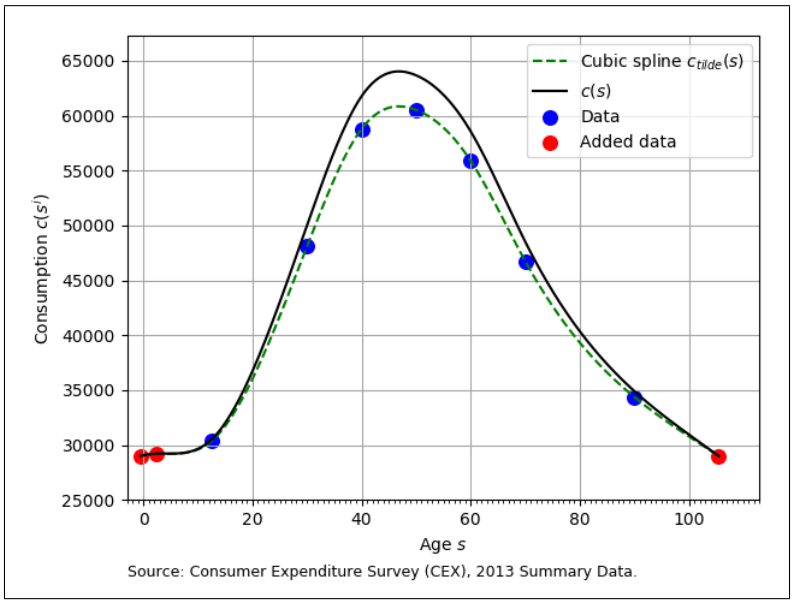
$$F(s_i) \equiv \int_{-\infty}^{s_i} f(s)ds \tag{11}$$

The first step toward approximating the continuous function $c(s)$ is to use the assumption that it will have roughly the same shape as the course histogram of age bins and average consumptions $\{s^i, c^i\}_{i=1}^I$. Figure 3 is a scatter plot of the points $\{m^i, c^i\}_{i=1}^6$ corresponding to the light blue histogram bars in Figure 1. The code for generating Figure 3 is given below.

```
# Import libraries
```

```
import numpy as np
import scipy.interpolate as si
import matplotlib.pyplot as plt
from matplotlib.ticker import MultipleLocator

# Read in (create) the data
avg_cons = np.array([29000, 29200, 30373, 48087, 58784,
                  60524, 55892, 46757, 34382, 29000])
age_cuts = np.array([-1, 0, 5, 25, 35, 45, 55, 65, 75, 105,
                  106])
age_midp = np.array([-0.5, 2.5, 12.5, 30, 40, 50, 60, 70, 90,
                  105.5])

# Generate interpolation function for consumption
# expenditures and use to get interpolated values
cons_func = si.interp1d(age_midp, avg_cons, kind='cubic')
age_fine = np.linspace(-0.5, 105.5, 1000)
cons_interp = cons_func(age_fine)

# Solve for factor_c and c(s) function
factor_c = 1.1 # This skipped spot is for factor_c calculation
c_s_pts = (factor_c * (cons_interp - 29000)) + 29000

# Create the scatter plot with interpolated curves
fig, ax = plt.subplots()
plt.scatter(age_midp[2:-1], avg_cons[2:-1], s=70, c='blue',
           marker='o', label='Data')
plt.scatter([age_midp[0], age_midp[1], age_midp[-1]],
           [avg_cons[0], avg_cons[1], avg_cons[-1]], s=70,
           c='red', marker='o', label='Added data')
plt.plot(age_fine, cons_interp, c='green', linestyle='--',
         label='Cubic spline $c_{tilde}(s)$')
plt.plot(age_fine, c_s_pts, c='black', linestyle='-',
         label='$c(s)$')
minorLocator = MultipleLocator(1)
ax.xaxis.set_minor_locator(minorLocator)
plt.grid(b=True, which='major', color='0.65', linestyle='-')
plt.title('Average consumption expenditure by age b +
          '$c(s^i)$', fontsize=15)
plt.xlabel(r'Age $s$')
plt.ylabel(r'Consumption $c(s^i)$')
plt.xlim((-3, 113))
plt.ylim((25000, 1.05 * c_s_pts.max()))
plt.legend(loc='upper right')
plt.text(-3, 18000, 'Source: Consumer Expenditure ' +
         'Survey (CEX), 2013 Summary Data.', fontsize=9)
plt.tight_layout(rect=(0, 0.03, 1, 1))
```

Figure 3: Histogram and midpoint coordinates of consumption summary data with interpolated function



Note in the `avg_cons`, `age_cuts`, and `age_midp` vectors and in the two left-most point and one right-most point in Figure 3, that I added three consumption expenditure values of \$29,000, \$29,200, and \$29,000, respectively. This ensures that my interpolating function will fit the data smoothly.

The cubic-spline interpolated function in Figure 3 (green dashed line) passes through each of the scatter plot points and has continuously differentiable first and second derivatives over the support. The the cubic spline be represented by $\tilde{c}(s)$. The problem with the function $\tilde{c}(s)$ is that its average consumption values generated by plugging it in for $c(s)$ in Equation 10 weighted by the population density $f(s)$ will not, in general, equal the actual average consumption values (scatter plot points) from the data.

The final step for estimating $c(s)$ is to adjust the interpolated function $\tilde{c}(s)$ by a constant $factor_c$ that sets the average consumption expenditure implied by the model equal to the average consumption expenditure in the data. We define $c(s)$ as being proportional to $\tilde{c}(s)$ by the constant $factor_c$.

$$c(s) \equiv \Big(factor_c\big[\tilde{c}(s) - 29,000\big]\Big) + 29,000 \tag{12}$$

The adding and subtracting of 29,000 in Equation (12) makes the end points in the scatter plot always stay constant. Combining (12) with (10) gives the equation that

characterizes the constant $factor_c$.

$$\text{Avg. CEX in data} = \int_{s_{min}}^{s_{max}} \frac{f(s)\left[\left(factor_c\left[\tilde{c}(s) - 29,000\right]\right) + 29,000\right]}{F(s_{max}) - F(s_{min})} ds \tag{13}$$

Once $factor_c$ is estimated from (13), that value with the continuous interpolated function $\tilde{c}(s)$ can be plugged in to (12) to get the estimate for the fundamental continuous consumption expenditure function that can be used to create any average consumption expenditure moments $c(s^i)$.

The dark solid line in Figure 3 shows the final $c(s)$ function.

2 Method 2: GMM estimation matching labor supply moments

The second approach is to estimate a polynomial function of a few parameters (5 or 6) that generates values for $\chi_{n,s}$ such that the steady-state labor supply by age $\bar{n}_s$ matches the labor supply moments from the data (see Exercises 1 and 2).

2.1 Chebyshev polynomials

The Stone-Weierstrass theorem states that any continuous function $f(x)$ defined on a closed interval $[a, b]$ can be uniformly approximated to arbitrary precision by a polynomial function.[1] In practice, some polynomials are easier to estimate than others. For example, let $P_4(x)$ be a general 4th-degree polynomial in x.

$$P_4(x) \equiv \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 \tag{14}$$

Imagine trying to estimate the coefficients of the 4th degree polynomial $P_4(x)$ to data y by least squares.

$$y \equiv \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \varepsilon \tag{15}$$

Separately identfying the parameters β_2, β_3, and β_4 become progressively harder because the coefficients x^2, x_3, and x_4 are increasingly positively correlated.

Orthogonal polynomials are a class of polynomials for which each of the coefficients is orthogonal to all of the other coefficients, respectively, despite each coefficient having a different order and the total orthogonal polynomial having order n representing the highest order coefficient.

3 Exercises (all method 2 for now)

Exercise 1. Use the `hrs_by_age.py` module in the `CPS_hrs_age` GitHub repository to create a vector of average hours by age as a percent of total possible hours. Choose

[1] See https://en.wikipedia.org/wiki/Stone%E2%80%93Weierstrass_theorem.

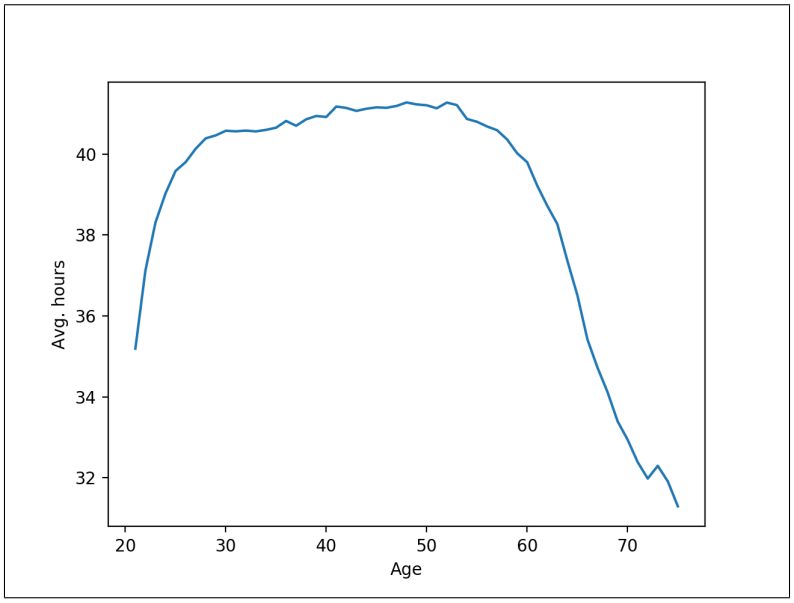
the following argument settings. Your plot of the `hrs_by_age_21to75` numpy array should look like Figure 4.

- Use input data from January 2015 to the most recent month's data available on the NBER site (October 2019).
- Have the module go scrape the data from the NBER site: `web=True`
- Create a Bokeh plot of the average of average annual hours by age: `graph=True, graph_type='bokeh'`
- Get data for all one-year duration ages starting with 21 and ending with 75: `age_bins=np.arange(21, 76)`

1. Fit a smooth curve to the `hrs_by_age` data for ages 21 to 75. Try using a Chebychev orthogonal polynomial of order 4, 5, or 6 (see numpy.polynomial.chebyshev.Chebyshev.fit). Or you could try to fit a fourth or fifth order standard polynomial. For ages 75 to 100, fit the following three-parameter negative exponential that satisfies the following three conditions.

$$g(age) = e^{b(age)^2+c(age)+d} \tag{16}$$
$$\text{s.t.} \quad T_d(age = 75) = g(age = 75) \tag{17}$$
$$\text{and} \quad T_d'(age = 75) = g'(age = 75) \tag{18}$$
$$\text{and} \quad g'(age = 100) = -0.05 \tag{19}$$

Figure 4: U.S. average annual hours by age, 21 to 75: Jan. 2015 to Oct. 2019



Exercise 2. Define a function named `calibrate_chi_n()` that has the following form.

```
def calibrate_chi_n(hrs_data):
    ...
    return chi_n_vec, poly_coefs
```

This function should return a vector `chi_n_vec` that has shape $(S,)$, where S is the number of ages and each value represents the household utility parameter $\chi_{n,s}$ that scales the disutility of labor supply in the household optimization problem. The function should also return polynomial coefficients `poly_coefs` that represent the estimated polynomial to generate the S calibrated values of $\chi_{n,s}$. The function `calibrate_chi_n()` estimates a polynomial to generate $\chi_{n,s}$ such that the steady-state labor supply $\bar{n}_s$ is as close as possible to the average labor supplies estimated in Exercise 1.

References

Fernández-Villaverde, Jesús and Dirk Krueger, "Consumption over the Life Cycle: Facts from Consumer Expenditure Survey Data," *Review of Economics and Statistics*, August 2007, *89* (3), 552–565.