Workshop 4: Random numbers & plotting

FIE463: Numerical Methods in Macroeconomics and Finance using Python

Richard Foltyn

Norwegian School of Economics (NHH)

February 6, 2025

See GitHub repository for notebooks and data:

https://github.com/richardfoltyn/FIE463-V25

1 Computing the mean for increasing sample sizes

In this exercise, you are asked to investigate how the mean of a simulated sample depends on the sample size.

Consider a set of random draws with samples sizes of 5, 10, 50, 100, 500, 1000, 5000, 10000, and 100000.

1. For each sample size, draw a normally-distributed sample with mean $\mu = 1$ and standard deviation $\sigma = 1$. Use a seed of 678 and make sure to reset the seed for each sample size.
2. For each sample size, compute the sample mean using `np.mean()`. Print a list of sample sizes and sample means.
3. Plot the means you computed against the sample size on the x-axis. You should use a log-scale for the x-axis and a marker symbol `'o'` to make the graph easier to read. Add a dashed horizontal to indicate the true mean, and label both axes.

   Hint: You can set the axis scale using `xscale()`.

   Hint: You can add a horizontal line with `axhline()`.

2 Static portfolio choice problem

Consider a portfolio choice problem where an investor chooses the fraction α to invest in a risky asset in order to maximize expected utility,

$$\max_{\alpha \in [0,1]} \mathbb{E}_t \left[u(W_{t+1}) \right]$$

Assume that the investor consumes all of next-period's wealth W_{t+1} which is given by

$$W_{t+1} = R_{t+1} \alpha W_t + R_f (1 - \alpha) W_t$$

where W_t is the initial investable wealth in period t, R_{t+1} is the gross return on the risky investment and R_f is the risk-free return on the fraction of the portfolio which is invested a risk-free asset (e.g., a bank deposit). The utility function $u(\bullet)$ has a constant relative risk aversion (CRRA) form and is given by

$$u(W) = \begin{cases} \frac{W^{1-\gamma}}{1-\gamma} & \text{if } \gamma \neq 1 \\ \log(W) & \text{if } \gamma = 1 \end{cases}$$

where γ is a parameter governing the investor's risk aversion.

For simplicity, let the gross risk-free return be $R_f = 1$. Finally, assume that the risky return can take on two realizations, high and low, with equal probability,

$$R_{t+1} = \begin{cases} 1 + \mu + \epsilon & \text{with probability } \frac{1}{2} \\ 1 + \mu - \epsilon & \text{with probability } \frac{1}{2} \end{cases}$$

where $\mu > 0$ is the risk premium and $\epsilon > 0$ parametrizes the volatility of risky returns.

In this exercise, you are asked to compute the optimal risky share and investigate how it depends on initial wealth.

1. Write a Python function that takes as arguments the risky share α, the initial wealth W_t, and the parameters μ, ϵ and γ, and returns the expected utility associated with the given values. Your function signature should look like this:

   ```
   def expected_util(alpha, W, mu, epsilon, gamma):
       # Compute the associated expected utility
       # eu = ...
       return eu
   ```

   Make sure that your function works correctly for both $\gamma = 1$ and $\gamma \neq 1$. Moreover, the function should allow for the arguments α and W to be passed as both scalar values as well as NumPy arrays!

2. Assume that the problem is parametrized with $W = 1$, $\gamma = 2$, $\mu = 0.04$, and $\epsilon = 0.2$.

   - Using the function you wrote, evaluate the expected utility on a grid of risky shares α with 1001 points which are uniformly spaced on the interval $[0, 1]$.
   - Plot this expected utility against the risky share. Label both axes and add a legend to your plot.

3. In the previous question, you plotted the expected against *all* possible risky share choices. Using grid search, find the optimal risky share and the associated expected utility.

   Augment the plot you created earlier and add a vertical dashed line to indicate the optimal risky share.

   Hint: You can use `np.argmax()` for this task.

   Hint: Use `axvline()` to add a vertical line.

4. Now consider a set of 100 initial wealth levels W_t which are uniformly spaces over the interval $[1, 10]$.

   - Write a loop that computes the optimal risky share for each of these wealth levels, using the same values for the remaining parameters as above.
   - Plot the optimal risky share by wealth against the grid of initial wealth levels. Set the y-axis limits to $(0, 1)$ to better visualize the result and explain your findings.