Inventory Management

Individual Assignment: Programming Track

2021–2022

Updated on April 18, 2022

The aim of this assignment is to assess your modelling skills—using Python. These skills will be of importance in the course. The assignment focuses on applications which involve writing formulas and recursions as well as making graphs. It covers only a few basic applications centered around inventory management. We will have many other applications in the course, yet all of these will be based on the building blocks that you exercise in this assignment.

We strongly encourage you to follow the suggestions below, not just to get a good grade but also to make the best out of the assignment.

1. COOPERATE: We suggest you work together and compare your results with your fellow students. Even though this is an individual assignment, the aim is that you learn as much as possible.

2. BE CRITICAL: Test your formulas, parameters, results, and graphs to see whether they are intuitive and reasonable.

3. READ: The exercises are not highly technical. They can be answered by making good use of examples which you can easily find on the internet.

If you need support on completing the assignment, you are welcome to the computer practical that will be held in the first week of the course. After finalizing the assignment, you should submit your answers on Nestor as a Jupyter Notebook (ipynb). The deadline can be found in the course manual.

Economic order quantity

Step 1. Make a section in your notebook and name it as "EOQ". All exercises in this part of this assignment will be done in this section.

Economic Order Quantity (EOQ) model is used to determine the optimal order quantity minimizing the cost per unit time in an inventory system. This inventory system is defined by the following parameters: demand per unit time D, unit procurement cost per item c, fixed ordering cost per order K, and holding cost per item per unit time h.

The optimal order quantity Q^* minimizing the total cost per unit time can be computed by means of the well-known EOQ formula

$$Q^* = \sqrt{\frac{2KD}{h}}.$$

Step 2. In the following, we will make use of the following parameter values—where the unit time is a year.

Variable	Definition	Value
D	Demand per year	60
c	Unit cost per item	2.5
K	Fixed cost per order	5
h	Holding cost per year per item	1.5

Insert a code cell. To facilitate access to parameter values, assign all the values in the table to explicit variables (e.g. `D = 60`).

Step 3. Implement the EOQ formula with the provided parameters and find the optimal order quantity Q^*. Print its value and verify that it is indeed correct.

The cost of an EOQ model depends on the choice of the order quantity Q. Notice that we already know that our best choice is Q^*. Now we will investigate the cost for different values of Q. We can write the cost per unit time as a function of Q as

$$C(Q) = \frac{hQ}{2} + \frac{KD}{Q} + cD.$$

Observe that this function is comprised of three components. That is, $\frac{hQ}{2}$ is the average holding cost per unit time, $\frac{KD}{Q}$ is the average ordering cost per unit time, and cD stands for the procurement cost per unit time. We shall abbreviate these components as $C_1(Q)$, $C_2(Q)$, $C_3(Q)$, such that $C(Q) = C_1(Q) + C_2(Q) + C_3(Q)$.

Step 4. Insert a code cell. Define each of the functions $C_1(Q)$, $C_2(Q)$, $C_3(Q)$, and $C(Q)$. These should take Q as a parameter and return the associated cost value.

Step 5. Compute the values of each component of the EOQ cost function for order quantities from 5 to 50 with steps of 5 and present your findings in a dataframe.[1] The contents of your dataframe should match the following table.

[1] Dataframe is a pandas data structure often used in data science and scientific computing. It contains two-dimensional data along with corresponding labels.

Q	$C_1(Q)$	$C_2(Q)$	$C_3(Q)$	$C(Q)$
5				
10				
⋮				
50				

Hint: It could be easier to collect the values of each column in a list and construct the dataframe from these lists.

Step 6. Look at the costs you obtained. Are they in line with your expectations? Is $C(Q)$ actually has its minimum at Q^* you computed before? Provide a short answer in a text cell.

Step 7. Insert a code cell. Make a line graph of all four functions in your table, where Q is on the x-axis and cost is on the y-axis. Add axis titles and legends in your graph.

Step 8. Examine how different functions behave in your graph. For instance, are they increasing or decreasing in Q? Provide a short answer in a text cell.

The following exercises are meant to analyze the inventory dynamics over time in an EOQ model. It is important to note at this point that the EOQ model assumes demand is "continuous in time". In other words, demand arrives every unit of time (day, hour, minute, second) no matter how small the unit of time is. This is obviously not very practical. To facilitate our analysis of inventory levels, we will discretize our unit time of a year in monthly intervals. To that end, we will first scale our parameters to a monthly basis.

Step 9. Insert a code cell. Introduce new demand and cost parameters that are scaled on a monthly basis rather than yearly. In doing so, refer to the parameters you introduced in Step 2. Implement the EOQ formula with the provided parameters and find the optimal order quantity Q^*. Print its value and verify that it is indeed correct.

Hint: Keep in mind that whether a particular parameter should or should not be scaled depends on its definition.

Step 10. Examine whether the optimal order quantity Q^* is the same after scaling parameters. Do you think that the result makes sense? Provide a short answer in a text cell.

We will now analyze inventory dynamics by computing order quantities and inventory levels over a 12-month planning horizon. To that end, we use the following variables: q_n is the order quantity at the beginning of month n, D_n is the demand in month n, and I_n is the inventory level at the end of month n. For all these variables, the month index n runs from 1 to 12. Nevertheless, for the inventory level variable we also use I_0 to express the initial inventory level at the start of the year. In the following, we will assume that $I_0 = 0$.

Step 11. Insert a code cell. Introduce three lists named `q`, `D`, and `I`. These should all contain 13 elements, initialized as zeros. Observe that the number of elements in the lists are one more than the number of months in the planning horizon. This is mainly for convenience as Python lists are indexed starting from zero. It enables us to use the same index for the time period and the list element. For instance, we can now write order quantities as $q_n =$ `q[n]`.

Step 12. Fill in $D_1, \ldots, D_{12}$ by referring to the monthly demand you computed earlier. Notice that the demand in each period is supposed to be the same, because the EOQ model assumes constant demand.

In our analysis, we will assume the following sequence of events in each month n. First, we observe the inventory level I_{n-1} at the end of the previous month. Second, we decide whether or not to place an order. We place an order if the inventory level I_{n-1} is not sufficient to cover the demand D_n in the current month. The order quantity q_n should be equal to Q^* if we decide to place an order, and it should be 0 otherwise. Third, demand realizes and satisfied from stock. Finally, the remaining inventory is carried to the next month.

Step 13. First, devise a formula to compute q_n provided the values of I_{n-1} and D_n. Second, devise a recursive formula that computes I_n provided the values of I_{n-1}, q_n, and D_n. The term "recursive" comes from the fact that we use I_{n-1} to compute I_n. Finally, implement these formulas in a loop to compute $q_1, \ldots, q_{12}$ and $I_1, \ldots, I_{12}$. Note that I_0 is already known.

Present your findings in a dataframe. The contents of your dataframe should match the following table.

n	q	D	I
0	0	0	0
1			
2			
⋮			
12			

Check your results and verify that an order is indeed placed whenever the inventory level at the end of the previous month is not sufficient to cover the actual demand.

Hint: The ordering decision can be determined by using an `if` statement. If the associated statement is true then order quantity should be Q^*, otherwise it should be 0.

Step 14. Examine whether it is possible to experience "stockouts" following our policy. Provide a short explanation in a text cell.

Hint: To better understand the system dynamics, you can try different values for I_0.

Step 15. Insert a code cell. Make a bar graph of the inventory level over time, where month index n is on the x-axis and inventory level I_n is on the y-axis. Add axis titles and legends in your graph.

Step 16. Examine how the inventory level behaves over time in your graph. Is it in line with your expectations? Provide a short answer in a text cell.

We will now incorporate costs in our analysis. To that end, we will compute the total holding, ordering, and procurement costs over the 12-month planning horizon.

Step 17. Insert a code cell. Devise the following formulas to compute the total holding, ordering,

and procurement costs, as well as the sum of all of them.

$$C_1 = h \sum_{n=1}^{12} I_n$$
$$C_2 = K \sum_{n=1}^{12} \mathbb{1}\{q_n > 0\}$$
$$C_3 = c \sum_{n=1}^{12} q_n$$
$$C = C_1 + C_2 + C_3$$

Here $\mathbb{1}\{\cdot\}$ stands for the indicator function. It takes the value of 1 if the statement in the curly brackets is true, and 0 otherwise. We are using this function to count the number of orders placed.

Implement these formulas while referring to your monthly data and print the values you obtained.

Hint: The indicator function can be implemented by means of an `if` statement.

Step 18. Examine the total holding, ordering, and procurement costs you have just computed and compare them against the costs you computed in Step 5 for the same order quantity. Are they the same? If not, why and what is the extent of the difference? Provide a short answer in a text cell.

Random variables and distributions

Step 19. Make a section in your notebook and name it as "Distributions". All exercises in this this part of this assignment will be done in this section.

In the context of inventory management, we often use random variables to model (probabilistic) demands. In what follows, we will focus on the distribution functions of random variables. We begin by introducing some notation. Let us consider an arbitrary random variable. We follow the general mathematical convention and denote this random variable by an uppercase letter X and any of its (possible) values by a lowercase letter x. We can define X by means of its probability distribution function[2] $f(\cdot)$ and cumulative distribution function $F(\cdot)$.

The probability distribution function indicates the probability that the random variable takes up a particular value x. That is

$$f(x) = \mathbb{P}\{X = x\}$$

where $\mathbb{P}\{\cdot\}$ stands for the probability operator which returns the probability of the event in the curly brackets.

The cumulative distribution function indicates the probability that the random variable takes up any value that is smaller than or equal to a particular value x. That is

$$F(x) = \mathbb{P}\{X \leq x\}.$$

[2]In case of discrete random variables, the proper naming convention is 'probability mass function' instead of 'probability distribution function'. Here, we use the same term for the sake of simplicity.

In the following exercises, we work on the aforementioned concepts by means of some discrete random variables with finite domain. To express such random variables, we introduce a list named `domain`, and two dictionaries named `f` and `F`. These will be the domain and the corresponding values of the probability distribution function and the cumulative distribution function of our random variable. For instance, consider a random variable X is such that $\mathbb{P}\{X = 3\} = 0.50$, $\mathbb{P}\{X = 11\} = 0.25$, and $\mathbb{P}\{X = 100\} = 0.25$. Then we have `domain = [3, 11, 100]`, `f = [3: 0.50, 11: 0.25, 100: 0.25]`, and `F = [3: 0.50, 11: 0.75, 100: 1.00]`.

Step 20. Let X be a random variable indicating the result of a dice roll. Insert a code cell. Introduce `domain`. Then compute `f` and `F`. Present your findings in a dataframe. The contents of your dataframe should match the following table.

x	$f(x)$	$F(x)$
$\vdots$		

Hint: It could be easier to compute $F(x)$ recursively. Besides, we must obviously have $F(x) = 1$ for the largest x. Make sure that this is indeed the case.

Step 21. Insert a code cell. Make a bar graph of $f(x)$ and $F(x)$, where x is on the x-axis and probabilities on the y-axis. Add axis titles and legends in your graph.

Step 22. Examine how $f(x)$ and $F(x)$ behave in your graph. For instance, are they increasing or decreasing in x? Why? Provide a short answer in a text cell.

Step 23. Repeat what you did in Steps 20–22. But this time assume that your random variable indicates the result of a rather peculiar dice roll. This dice has 21 sides which are numbered as $0, 1, \ldots, 20$.

Present your results as you did earlier except make a line graph instead of a bar graph for better visibility.

Step 24. Repeat what you did in Steps 20–22. But this time assume that your random variable follows a Poisson distribution with mean 10. Note that the domain of Poisson distribution is not bounded. That is, the values x can be arbitrarily large. In your computations, only consider values $0, 1, \ldots, 20$.

Present your results as you did earlier except make a line graph instead of a bar graph for better visibility.

Hint: The distribution functions of Poisson distribution can be accessed via the `poisson` function from the package `scipy.stats`.

Step 25. Examine the graphs you obtained in Step 23 and Step 24. Do you observe any differences? Why? Provide a short answer in a text cell.

Computing expectations

Step 26. Make a section in your notebook and name it as "Expectations". All exercises in this this part of this assignment will be done in this section.

The concept of computing expected values of functions involving random variables is important in inventory management. The aim of the following exercises is to practice doing such computations.

We first explain the term "expected value" in the context of discrete random variables. Let us consider a random variable X. Suppose X can take values $x = 1, 2, 3$ with probabilities $f(1) = 1/4$, $f(2) = 2/4$, and $f(3) = 1/4$. Then, the expected value of X can be computed as

$$\mathbb{E}\{X\} = \frac{1}{4} \times 1 + \frac{2}{4} \times 2 + \frac{1}{4} \times 3 = \frac{1+4+3}{4} = 2$$

Here $\mathbb{E}\{\cdot\}$ is the expectation operator. It returns the expectation of the expression (random variable or function) inside the curly brackets.

Now, consider the function $g(x) = x^2$. We can easily compute this function for any given x. For instance, $g(1) = 1$ and $g(5) = 25$. But what if x is the value of a random variable? Say, X. Then, we can compute the expected value of $g(X)$ as

$$\begin{aligned}\mathbb{E}\{g(X)\} &= \frac{1}{4} \times g(1) + \frac{2}{4} \times g(2) + \frac{1}{4} \times g(3) \\ &= \frac{1}{4} \times 1 + \frac{2}{4} \times 4 + \frac{1}{4} \times 9 = \frac{1+8+9}{4} = 4.5\end{aligned}$$

More formally, we can compute the expectation of any function $g(X)$ by means of the following formula

$$\mathbb{E}\{g(X)\} = \sum_x f(x)g(x)$$

where $\sum_x$ is the summation over all possible values of x that random variable X can take.[3]

Step 27. Consider the random variable characterized by the domain and the corresponding values of the probability distribution function provided in the following table. Insert a code cell. Introduce `domain` and `f`.

x	$f(x)$
0	0.02
1	0.11
2	0.14
3	0.13
4	0.04
5	0.04
6	0.03
7	0.11
8	0.07
9	0.17
10	0.14

Step 28. Consider the function "$g(x) = x$". Devise a formula to compute the expected value $\mathbb{E}\{g(X)\}$ of this function. Implement this formula and print the value you obtained.

Hint: It could be easier to use a generator expression when computing the expected value.

[3] This description assumes a discrete random variable. In case of continuous random variables, the summation should be replaced with an integral over the domain of the random variable.

Step 29. Repeat Step 28 with the function "$g(x) = 2x$".

Step 30. Repeat Step 28 with the function "$g(x) = x - 5$".

Step 31. Repeat Step 28 with the function "$g(x) = 2x + 5$".

Step 32. Repeat Step 28 with the function "$g(x) = x^2$".

Step 33. Repeat Step 28 with the function "$g(x) = \max\{x - 5, 0\}$".

Step 34. Repeat Step 28 with the function "$g(x) = \max\{5 - x, 0\}$".

Step 35. Insert a code cell. Devise a formula to compute the expected value $\mathbb{E}\{X\}$ of X.

Step 36. Devise formulas to compute $g(\mathbb{E}\{X\})$ for all of the functions $g(x)$ you considered in Step 28–34. In doing so, refer to the value of $\mathbb{E}\{X\}$ you computed in Step 35. Implement these formulas and print the values you obtained.

Hint: Mind the difference between $g(\mathbb{E}\{X\})$ and $\mathbb{E}\{g(X)\}$.

Step 37. Examine the values of $g(\mathbb{E}\{X\})$ and $\mathbb{E}\{g(X)\}$ for all different functions you considered. For which functions the values are the same? What are the structural similarities between these functions? Provide a short answer in a text cell.

Generating random numbers

Step 38. Make a section in your notebook and name it as "Random". All exercises in this this part of this assignment will be done in this section.

It is essential to be able to generate random numbers to simulate inventory systems. In the following exercises, we will generate random numbers from given probability distributions. To that end, we will use the Python package `scipy.stats` which provides access to many well-known distributions. For illustrative purposes, we will generate random numbers from discrete uniform distribution and Poisson distribution. These can be accessed via the `randint` and `poisson` functions of `scipy.stats`.

Step 39. Insert a code cell. Generate a list of 1000 random numbers from (1) a discrete uniform distribution within the interval $[0, 20]$ and (2) a Poisson distribution with mean 10. Name these lists as `rvs_randint` and `rvs_poisson`, respectively.

Next, we will compute the relative frequencies and the cumulative relative frequencies of certain values in each of the the lists of random numbers we generated. The former is the percentage of generated random numbers that are equal to the value x. The latter is the percentage of generated random numbers that are smaller than or equal to the value x. We will refer to these as $f^{\text{Sim}}(x)$ and $F^{\text{Sim}}(x)$, respectively.

Step 40. Insert separate code cells for Randint and Poisson random numbers. Devise formulas to compute the associated relative and cumulative relative frequencies for x values within $[0, 20]$. Present your results in a dataframe. The contents of your dataframes should match the following table.

Hint: It could be easier to use the built-in function `count` or the `count_nonzero` function of the `numpy` package in your computations.

x	$f^{\text{Sim}}(x)$	$F^{\text{Sim}}(x)$
0		
1		
⋮		
20		

Step 41. Insert separate code cells for Randint and Poisson random numbers. Make a line graph of the associated $f^{\text{Sim}}(x)$ and $F^{\text{Sim}}(x)$, where x is on the x-axis and relative frequencies on the y-axis. Add axis titles and legends in your graph.

Step 42. There is an analogy between Randint random numbers and the random variable in Step 23 as well as between Poisson random numbers and the random variables in Step 24. Compare the graphs of relative and cumulative relative frequencies with the graphs of probability distribution and cumulative distribution functions in Step 23 and Step 24. Do you observe any similarities and/or differences? Why? Provide a short answer in a text cell.

Simulation

Step 43. Make a section in your notebook and name it as "Simulation". All exercises in this this part of this assignment will be done in this section.

In the following exercises, we will simulate an inventory system under a pre-specified inventory policy over 10000 periods. To that end, we will use tools that we have already worked out in the previous parts of the assignment.

Step 44. Insert a code cell. Introduce three lists named `q`, `D`, and `I`. These should all contain 10001 elements, initialized as zeros.

Step 45. To simulate demands, generate 10000 random numbers from a Poisson distribution with mean 10. Fill in these random numbers into `D`, excluding its first element.

In our simulation, we will assume that the inventory system is controlled by an inventory policy which is often referred to as the (s, Q) policy. Following this policy, if the inventory level at the end of the previous period is (strictly) below a certain threshold s then a replenishment order of size Q is placed in the current period. We will assume that demand that cannot be satisfied from stock will be backordered. For instance, assume that the inventory level at the end of the previous period is 4 units and we decided not to order in the current period. Consequently, if the actual demand happens to be 7 units then we satisfy 4 units from stock and backorder 3 units of demand. This means that the inventory level at the end of the current period will be -3 units. Then we say that there is a backlog of 3 units. However, if the actual demand happens to be 2 units then we satisfy all demand from stock. This means that the inventory level at the end of the current period will be 2. Then we say that there is an on-hand inventory of 2 units. Observe that the inventory level can be positive or negative while the on-hand inventory and backlog can only take non-negative values. Besides, we can either have on-hand inventory or backlog in any given period.

Step 46. In our simulation, we will use the policy parameters $s = 5$ and $Q = 15$. To access these parameters, assign their values to explicit variables.

Step 47. First, devise a formula to compute q_n provided the values of I_{n-1} and D_n. Second, devise a recursive formula that computes I_n provided the values of I_{n-1}, q_n, and D_n. Finally, implement these formulas in a loop to compute $q_1, \ldots, q_{12}$ and $I_1, \ldots, I_{12}$.

Present your findings in a dataframe. The contents of your dataframe should match the following table.

n	q	D	I
0	0	0	0
1			
2			
⋮			
10000			

We have simulated the (s, Q) policy. The next step is to compute the key performance indicators of the inventory system. We will consider three performance indicators, namely the average on-hand inventory per period, the average backlog per period, and the fraction of periods where an order is placed.

Step 48. Devise formulas to compute the performance indicators of the system and print the values you obtained.

Step 49. Examine how changing the values of s and Q affects performance indicators. Provide a short explanation in a text cell.