Exercises - Section 4: Lecture 12 - Multiple Subscripts - Solutions

Just like the tradition of using letters at the start of the alphabet for data and using letters at the end of the alphabet for variables, the tradition for subscripts is to start with i, then j, k, et cetera. However, there are exceptions, especially when the letter chosen might have meaning; for example, "t" is a common subscript to refer to time. Different modelers will use different notation, so don't worry if yours is different from someone else's.

Suppose we have the following notation for a rental car fleet planning model:

DESCRIPTION	MATH	GUROBIPY
Car types: i=1 (small), i=2 (midsize), i=3 (luxury), i=4 (SUV), i=5 (minivan)	i	`i`
Rental locations (100 locations, so j=1,2,…,100)	j	`j`
Forecasted annual rental demand for car type i at location j (forecasted data)	D_{ij}	`D = {(i, j): d for i in range(1, 6) for j in range(1, 101)}`
Cost per car for car type i (known data; same cost regardless of location)	C_i	`C = {i: c for i in range(1, 6)}`
Number of cars of type i to purchase for location j (variable)	x_{ij}	`x= model.addVars([(i, j) for i in range(1, 6) for j in range(1, 101)], name="x")`

Fill in the table below, using the notation above to describe each of the following quantities mathematically and in Python.

QUESTION	MATH	GUROBIPY
1. Forecasted rental demand for luxury cars at location 75	$D_{3,75}$	`D[3,75]`
2. Forecasted rental demand for SUVs at location 31	$D_{4,31}$	`D[4,31]`
3. Cost per SUV	C_4	`C[4]`
4. Number of small cars to purchase at location 12	$x_{1,12}$	`x[1, 12]`

5. Constraint: Need to purchase at least 20 small cars at location 12	$x_{1,12} \geq 20$	`model.addConstr(x[1, 12] >= 20)`
6. Constraint: Can't purchase more than 30 small cars at location 12	$x_{1,12} \leq 30$	`model.addConstr(x[1, 12] <= 30)`

7. Forecasted demand for midsize cars and minivans at location 31	$D_{2,31} + D_{5,31}$
	or
	$\sum_{i \in \{2,5\}} D_{i,31}$
		`D[2, 31] + D[5, 31]`
		or
		`sum(D[i, 31] for i in [2, 5])`

8. Forecasted demand for midsize cars and minivans at locations 31-40	$D_{2,31} + D_{5,31} + D_{2,32} + D_{5,32} + D_{2,33} + D_{5,33} + D_{2,34} + D_{5,34} + D_{2,35} + D_{5,35} + D_{2,36} + D_{5,36} + D_{2,37} + D_{5,37} + D_{2,38} + D_{5,38} + D_{2,39} + D_{5,39} + D_{2,40} + D_{5,40}$
	or
	$\sum_{i \in \{2,5\}} \sum_{j=31}^{40} D_{ij}$
	Notice how much easier the summation notation is. From here down, we'll only use summation notation when there are 3 or more terms.
		`D[2, 31] + D[5, 31] +`
		`D[2, 32] + D[5, 32] +`
		`D[2, 33] + D[5, 33] +`
		`D[2, 34] + D[5, 34] +`
		`D[2, 35] + D[5, 35] +`
		`D[2, 36] + D[5, 36] +`
		`D[2, 37] + D[5, 37] +`
		`D[2, 38] + D[5, 38] +`
		`D[2, 39] + D[5, 39] +`
		`D[2, 40] + D[5, 40] +`
		or
		`sum(D[i, j] for j in range(31, 41) for i in [2, 5])`
		Notice how much easier the summation notation is. From here down, we'll only use summation notation when there are 3 or more terms.

9. Forecasted rental demand for cars (not SUVs or minivans) at locations 1-50	$\sum_{i=1}^{3} \sum_{j=1}^{50} D_{ij}$	`sum(D[i, j] for j in range(1, 51) for i in range(1, 4))`

10. Total cars to purchase above midsize at location 4	$\sum_{i=3}^{5} x_{i,4}$	`gp.quicksum(x[i, 4] for i in range(3, 6))`
		or
		`x.sum(range(3, 6), 4)`

11. Total luxury cars to purchase across all locations	$\sum_{j=1}^{100} x_{3,j}$	`gp.quicksum(x[3, j] for j in range(1, 101))`
		or
		`x.sum(3, range(1, 101))`

12. Total vehicles to purchase (across all locations)	$\sum_{i=1}^{5} \sum_{j=1}^{100} x_{ij}$	`gp.quicksum(x[i, j] for i in range(1, 6) for j in range(1, 101))`

		or `x.sum("*", "*")` *NOTE:* `"*"` *means to include all values.*
13. Cost of midsize cars purchased at location 1	$C_2 x_{2,1}$	`C[2] * x[2, 1]`
14. Cost of SUVs purchased at location 96	$C_4 x_{4,96}$	`C[4] * x[4, 96]`

15. Cost of midsize cars and minivans purchased at location 96	$C_2 x_{2,96} + C_5 x_{5,96}$
	or
	$\sum_{i \in \{2,5\}} C_i x_{i,96}$
		`C[2] * x[2, 96] + C[5] * x[5, 96]`
		or
		`gp.quicksum(C[i] * x[i, 96] for i in [2, 5])`

16. Cost of SUVs and minivans purchased at locations 90-100	$\sum_{i=4}^{5} \sum_{j=90}^{100} C_i x_{ij}$
	or
	$\sum_{i=4}^{5} \left(C_i \sum_{j=90}^{100} x_{ij} \right)$
		`gp.quicksum(C[i] * x[i, j] for i in [4, 5] for j in range(90, 101))`
		or
		`gp.quicksum(C[i] * gp.quicksum(x[i,j] for j in range(90,101) for i in [4,5])`

17. Cost of all cars (not SUVs or minivans) purchased across all locations	$\sum_{i=1}^{3} \sum_{j=1}^{100} C_i x_{ij}$
	or
	$\sum_{i=1}^{3} \left(C_i \sum_{j=1}^{100} x_{ij} \right)$
		`gp.quicksum(C[i] * x[i, j] for i in range(1, 4) for j in range(1, 101))`
		or
		`gp.quicksum(C[i] * gp.quicksum(x[i,j] for j in range(1,101) for i in range(1, 4))`

18. Cost of all vehicles purchased across all locations	$\sum_{i=1}^{5} \sum_{j=1}^{100} C_i x_{ij}$
	or
	$\sum_{i=1}^{5} \left(C_i \sum_{j=1}^{100} x_{ij} \right)$
		`gp.quicksum(C[i] * x[i, j] for i in range(1, 5) for j in range(1, 101))`
		or
		`gp.quicksum(C[i] * gp.quicksum(x[i,j] for j in range(1,101) for i in range(1, 5))`

19. Average purchase cost of all cars (not SUVs or minivans) at location 20	$\frac{\sum_{i=1}^{3} C_i x_{i,20}}{\sum_{i=1}^{3} x_{i,20}}$	`y = model.addVar(name="y")` `y * x.sum(range(1, 4), 20) == gp.quicksum(C[i] * x[i, 20] for i in range(1, 4))`

		NOTE: gurobipy is not able to handle having variables in the denominator of a fraction. So, directly translating the fraction (numerator divided by denominator) won't work. *Instead, it is necessary to create a new variable y for the fraction, and set y times the denominator equal to the numerator.* *Compare this to the answer to #21 below, where the denominator includes only data, with no variables. In that case, the straightforward answer is fine.*

20. Average purchase cost of all vehicles at all locations	$\frac{\sum_{i=1}^{5}\sum_{j=1}^{100} C_i x_{ij}}{\sum_{i=1}^{5}\sum_{j=1}^{100} x_{ij}}$	`y = model.addVar(name=”y”)`
		`y * x.sum(“*”, “*”) == gp.quicksum(C[i] * x[i, j] for i in range(1, 6) for j in range(1, 101))`
		
		Why is the new variable y needed? See note to #19.

21. Forecasted purchase cost per rental of SUVs at location 17	$\frac{C_4 x_{4,17}}{D_{4,17}}$	`(C[4] / D[4, 17]) * x[4, 17]` *Why is a new variable like in #19 and #20 not needed here? See note to #19.*

22. Forecasted purchase cost per rental of minivans across locations 1-10	$\frac{\sum_{j=1}^{10} C_5 x_{5,j}}{\sum_{j=1}^{10} D_{5,j}}$
	
	or
	
	$\frac{C_5 \sum_{j=1}^{10} x_{5,j}}{\sum_{j=1}^{10} D_{5,j}}$
		`gp.quicksum(C[5] * x[5, j] for j in range(1, 11)) / sum(D[5, j] for j in range(1, 11))`
		
		or
		
		`C[5] * x.sum(5, range(1, 11)) / sum(D[5, j] for j in range(1, 11))`
		
		Why is a new variable like in #19 and #20 not needed here? See note to #19.

23. Forecasted purchase cost per rental over all vehicles at location 17	$\frac{\sum_{i=1}^{5} C_i x_{i,17}}{\sum_{i=1}^{5} D_{i,17}}$	`gp.quicksum(C[i] * x[i, 17] for i in range(1, 6)) / sum(D[i, 17] for i in range(1, 6))`

		Why is a new variable like in #19 and #20 not needed here? See note to #19.
24. Forecasted purchase cost per rental over all vehicles at all locations	$\frac{\sum_{i=1}^{5}\sum_{j=1}^{100} C_i x_{ij}}{\sum_{i=1}^{5}\sum_{j=1}^{100} D_{ij}}$ *or* $\frac{\sum_{i=1}^{5}\left(C_i \sum_{j=1}^{100} x_{ij}\right)}{\sum_{i=1}^{5}\sum_{j=1}^{100} D_{ij}}$	`gp.quicksum(C[i] * x[i, j] for i in range(1, 6) for j in range(1, 101)) / sum(D[i, j] for i in range(1, 6) for j in range(1, 101))` *Why is a new variable like in #19 and #20 not needed here? See note to #19.*

NOTES: