Exercises - Section 4: Lecture 11 - Notation Subscripts Summations – Part 2 - Solutions

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

ENGLISH	MATH	GUROBIPY
Car types: $i=1$ (small), $i=2$	i	i
(midsize), $i=3$ (luxury), $i=4$		
(SUV), i=5 (minivan)		
Forecasted rental demand for	D_i	D = {i: d for i in range(1, 6)}
car type <i>i</i> (forecasted data)		
Cost per car for car type i	C_i	C = {i: c for i in range(1, 6)}
(known data)		
Number of cars of type i to	x_i	x = model.addVars(range(1, 6),
purchase (variable)		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	D_3	D[3]
luxury cars		
2. Forecasted rental demand for	D_4	D[4]
SUVs		
3. Cost per SUV	C_4	C[4]
4. Number of small cars to	x_1	x[1]
purchase		
5. Forecasted demand for	$D_2 + D_5 \text{ or } \sum_{i \in \{2,5\}} D_i$	D[2] + D[5]
midsize cars and minivans		or
		sum(D[i] for i in [2, 5])
6. Forecasted rental demand for	$D_1 + D_2 + D_3 \text{ or } \sum_{i=1}^3 D_i$	D[1] + D[2] + D[3]
cars (not SUVs or minivans)		or
		<pre>sum(D[i] for i in range(1,</pre>
		4)
7. Total cars to purchase above	$x_3 + x_4 + x_5$ or $\sum_{i=3}^5 x_i$	x[3] + x[4] + x[5]
midsize		or
		x.sum([3, 4, 5])
8. Total vehicles to purchase	$x_1 + x_2 + x_3 + x_4 + x_5$	x[1] + x[2] + x[3] + x[4] +
	or	x[5]
	5	or
	$\sum x_i$	x.sum("*")
	<u>i=1</u>	

Introduction to Optimization Through the Lens of Data Science Course Exercises

		NOTE: Summing over "*" means to	
		sum over all defined subscripts	
9. Cost of midsize cars purchased	C_2x_2	C[2] * x[2]	
10. Cost of SUVs purchased	C_4x_4	C[4] * x[4]	
11. Cost of midsize cars and	$C_2x_2 + C_5x_5$	C[2] * x[2] + C[5] * x[5]	
minivans purchased	$\sum_{i\in\{2,5\}} C_i x_i$	<pre>or gp.quicksum(C[i] * x[i] for i in [2, 5])</pre>	
12. Cost of SUVs and minivans	$C_4x_4 + C_5x_5$	i in [2, 5]) C[4] * x[4] + C[5] * x[5]	
	or	or v prod (C [2 5])	
	$\sum_{i=1}^{5} C_i x_i$	x.prod(C, [2, 5])	
	$\sum_{i=1}^{n} c_i x_i$	gp.quicksum(C[i] * x[i] for	
	l=4		
13. Cost of all cars (not SUVs or	$C_1x_1 + C_2x_2 + C_3x_3$	i in [4, 5]) C[1] * x[1] + C[2] * x[2] +	
minivans) purchased	or	C[3] * x[3]	
, p	3	or	
	$\sum_{i=1}^{} C_i x_i$	x.prod(C, [1, 2, 3])	
14. Cost of all vehicles purchased	$C_1x_1 + C_2x_2 + C_3x_3 +$	C[1] * x[1] + C[2] * x[2] +	
	$C_4x_4 + C_5x_5$	C[3] * x[3] + C[4] * x[4] +	
	or	C[5] * x[5]	
	$\sum_{i=1}^{5}$	or	
	$\sum_{i=1}^{\infty} C_i x_i$	x.prod(C)	
15. Forecasted purchase cost per	$C_4 x_4$	C[4] * x[4] / D[4]	
rental of SUVs	D_4		
16. Forecasted purchase cost per	$C_5 x_5$	C[5] * x[5] / D[5]	
rental of minivans	D_5		
17. Average purchase cost over	$C_4x_4 + C_5x_5$	y = model.addVar(name="y")	
all SUVs and minivans	$x_4 + x_5$	y * x.sum([4, 5]) ==	
	or	<pre>gp.quicksum(C[i] * x[i] for</pre>	
	$\frac{\sum_{i=4}^{5} C_i x_i}{\sum_{i=4}^{5} x_i}$	i in [4, 5])	
	$\sum_{i=4}^{5} x_i$	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 #19	
		below, where the denominator includes	

Introduction to Optimization Through the Lens of Data Science Course Exercises

		only data, with no variables. In that case, the straightforward answer is fine.
18. Average purchase cost of all cars (not SUVs or minivans)	$\frac{C_{1}x_{1} + C_{2}x_{2} + C_{3}x_{3}}{x_{1} + x_{2} + x_{3}}$ or $\frac{\sum_{i=1}^{3} C_{i}x_{i}}{\sum_{i=1}^{3} x_{i}}$	<pre>y = model.addVar(name="y") y * x.sum(range(1, 4)) == gp.quicksum(C[i] * x[i] for i in [1, 2, 3]) Why is the new variable y needed? See note to #17.</pre>
19. Forecasted purchase cost per rental over all cars (not SUVs or minivans)	$\frac{C_{1}x_{1} + C_{2}x_{2} + C_{3}x_{3}}{D_{1} + D_{2} + D_{3}}$ or $\frac{\sum_{i=1}^{3} C_{i}x_{i}}{\sum_{i=1}^{3} D_{i}}$	<pre>x.prod(C, [1, 2, 3]) / sum(D[i] for i in [1, 2, 3]) or gp.quicksum(C[i] * x[i] for i in [1, 2, 3]) / sum(D[i] for i in [1, 2, 3]) Why is a new variable like in #17 and #18 not needed here? See note to #17.</pre>
20. Forecasted purchase cost per rental over all vehicles	$\frac{C_{1}x_{1}+C_{2}x_{2}+C_{3}x_{3}+C_{4}x_{4}+C_{5}x_{5}}{D_{1}+D_{2}+D_{3}+D_{4}+D_{5}}$ or $\frac{\sum_{i=1}^{5}C_{i}x_{i}}{\sum_{i=1}^{5}D_{i}}$	<pre>x.prod(C) / sum(D.values()) or gp.quicksum(C[i] * x[i] for i in range(1, 6)) / sum(D[i] for i in range(1, 6)) Why is a new variable like in #17 and #18 not needed here? See note to #17.</pre>

NOTES:		
NOTES.		

This is an open education course, and we encourage faculty and students to use the exercises and materials included to help meet educational and instructional goals. Gurobi Optimization, LLC ("Gurobi") is the owner of the content (all right, title, and interest) of this course ("Content"). Content is not designed or intended for commercial use or for the sale of courses. Gurobi makes this Content available at no cost for educational and instructional use via faculty and student download and use of Content in courses offered at your institution/organization; provided that such use does not scale or otherwise result in the use of Content for monetary gain including in any other online course. Content is offered as-is and all risk of use resides with you. Gurobi makes no warranties or guarantees of a particular outcome and accepts no liability, direct, consequential, or otherwise, arising from the access to or use of the course and Content contained in it.