

```
In [1]: from pulp import LpMinimize, LpProblem, LpStatus, lpSum, LpVariable
import pandas as pd
import numpy as np
```

```
In [2]: ## Question A
```

```
In [3]: time = ['7-9', '9-11', '11-13', '13-15', '15-17', '17-19', '19-21']
data = {'total calls': [35*2, 90*2, 80*2, 105*2, 95*2, 60*2, 35*2]}
df = pd.DataFrame(data, index = time)
df
```

Out[3]:

total calls	
7-9	70
9-11	180
11-13	160
13-15	210
15-17	190
17-19	120
19-21	70

```
In [4]: numOperator = {'English':[0.75], 'Spanish':[0.25]}
type(numOperator)
```

Out[4]: dict

```
In [5]: df['numberCallperHour'] = [35, 90, 80, 105, 95, 60, 35]
df['English'] = np.ceil((df['total calls']/10) * numOperator['English'])
df['Spanish'] = np.ceil((df['total calls']/10) * numOperator['Spanish'])
df
```

Out[5]:

	total calls	numberCallperHour	English	Spanish
7-9	70	35	6.0	2.0
9-11	180	90	14.0	5.0
11-13	160	80	12.0	4.0
13-15	210	105	16.0	6.0
15-17	190	95	15.0	5.0
17-19	120	60	9.0	3.0
19-21	70	35	6.0	2.0

```
In [6]: ## Question B
```

```
In [7]: time = ['7-9', '9-11', '11-13', '13-15', '15-17', '17-19', '19-21']
```

```
In [8]: # possibility
#1:English-speaking operator who works from 7 to 3 and does paperwork at the star
#2:Spanish-speaking operator who works from 7 to 3 and does paperwork at the star
#3:English-speaking operator who works from 7 to 3 and does paperwork at the end
#4:Spanish-speaking operator who works from 7 to 3 and does paperwork at the end
#5:English-speaking operator who works from 11 to 7 and does paperwork at the sta
#6:Spanish-speaking operator who works from 11 to 7 and does paperwork at the sta
#7:English-speaking operator who works from 11 to 7 and does paperwork at the end
#8:Spanish-speaking operator who works from 11 to 7 and does paperwork at the end
#9:English-speaking operator who works part-time from 3 to 7
#10:Spanish-speaking operator who works part-time from 3 to 7
#11:English-speaking operator who works part-time from 5 to 9
#12:Spanish-speaking operator who works part-time from 5 to 9

assignment = {
    1: [0,5,5,5,0,0,0],
    2: [0,5,5,5,0,0,0],
    3: [5,5,5,0,0,0,0],
    4: [5,5,5,0,0,0,0],
    5: [0,0,0,5,5,5,0],
    6: [0,0,0,5,5,5,0],
    7: [0,0,5,5,5,0,0],
    8: [0,0,5,5,5,0,0],
    9: [0,0,0,0,5,5,0],
    10: [0,0,0,0,5,5,0],
    11: [0,0,0,0,0,5,5],
    12: [0,0,0,0,0,5,5]
}

table = pd.DataFrame(assignment, index = time)
table
```

Out[8]:

	1	2	3	4	5	6	7	8	9	10	11	12
7-9	0	0	5	5	0	0	0	0	0	0	0	0
9-11	5	5	5	5	0	0	0	0	0	0	0	0
11-13	5	5	5	5	0	0	5	5	0	0	0	0
13-15	5	5	0	0	5	5	5	5	0	0	0	0
15-17	0	0	0	0	5	5	5	5	5	5	0	0
17-19	0	0	0	0	5	5	0	0	5	5	5	5
19-21	0	0	0	0	0	0	0	0	0	0	5	5

```
In [9]: job = np.arange(1,13)
```

```
In [10]: model1 = LpProblem("Callcenter", LpMinimize)
```

```
In [11]: numOperator = LpVariable.dicts('nums_', ((t,j) for t in time for j in job), lowBo
```

```
In [12]: costs= {
    1: 75,
    2: 75,
    3: 75,
    4: 75,
    5: 75,
    6: 75,
    7: 75,
    8: 75,
    9: 45,
    10: 45,
    11: 45,
    12: 45
}

costs
```

Out[12]: {1: 75,
2: 75,
3: 75,
4: 75,
5: 75,
6: 75,
7: 75,
8: 75,
9: 45,
10: 45,
11: 45,
12: 45}

```
In [13]: model1 += lpSum(numOperator['7-9',x] * costs[x] for x in job)
model1
```

Out[13]: Callcenter:
MINIMIZE
75*nums__('7_9',_1) + 45*nums__('7_9',_10) + 45*nums__('7_9',_11) + 45*nums__('7_9',_12) + 75*nums__('7_9',_2) + 75*nums__('7_9',_3) + 75*nums__('7_9',_4) + 75*nums__('7_9',_5) + 75*nums__('7_9',_6) + 75*nums__('7_9',_7) + 75*nums__('7_9',_8) + 45*nums__('7_9',_9) + 0
VARIABLES
0 <= nums__('7_9',_1) Integer
0 <= nums__('7_9',_10) Integer
0 <= nums__('7_9',_11) Integer
0 <= nums__('7_9',_12) Integer
0 <= nums__('7_9',_2) Integer
0 <= nums__('7_9',_3) Integer
0 <= nums__('7_9',_4) Integer
0 <= nums__('7_9',_5) Integer
0 <= nums__('7_9',_6) Integer
0 <= nums__('7_9',_7) Integer
0 <= nums__('7_9',_8) Integer
0 <= nums__('7_9',_9) Integer

```
In [15]: for t in time:
          model1 += (lpSum(numOperator[t,x] * table.loc[t,x] for x in [2,4,6,8,10,12]))
model1
```

```
Out[15]: Callcenter:
MINIMIZE
75*nums__('7_9',_1) + 45*nums__('7_9',_10) + 45*nums__('7_9',_11) + 45*nums__
('7_9',_12) + 75*nums__('7_9',_2) + 75*nums__('7_9',_3) + 75*nums__('7_9',_4) +
75*nums__('7_9',_5) + 75*nums__('7_9',_6) + 75*nums__('7_9',_7) + 75*nums__('7_
9',_8) + 45*nums__('7_9',_9) + 0
SUBJECT TO
('7_9',_Spanish'): 5 nums__('7_9',_4) >= 8.75

('9_11',_Spanish'): 5 nums__('9_11',_2) + 5 nums__('9_11',_4) >= 22.5

('11_13',_Spanish'): 5 nums__('11_13',_2) + 5 nums__('11_13',_4)
+ 5 nums__('11_13',_8) >= 20

('13_15',_Spanish'): 5 nums__('13_15',_2) + 5 nums__('13_15',_6)
+ 5 nums__('13_15',_8) >= 26.25

('15_17',_Spanish'): 5 nums__('15_17',_10) + 5 nums__('15_17',_6)
+ 5 nums__('15_17',_8) >= 23.75

('17_19',_Spanish'): 5 nums__('17_19',_10) + 5 nums__('17_19',_12)
+ 5 nums__('17_19',_6) >= 15

('19_21',_Spanish'): 5 nums__('19_21',_12) >= 8.75

VARIABLES
0 <= nums__('11_13',_2) Integer
0 <= nums__('11_13',_4) Integer
0 <= nums__('11_13',_8) Integer
0 <= nums__('13_15',_2) Integer
0 <= nums__('13_15',_6) Integer
0 <= nums__('13_15',_8) Integer
0 <= nums__('15_17',_10) Integer
0 <= nums__('15_17',_6) Integer
0 <= nums__('15_17',_8) Integer
0 <= nums__('17_19',_10) Integer
0 <= nums__('17_19',_12) Integer
0 <= nums__('17_19',_6) Integer
0 <= nums__('19_21',_12) Integer
0 <= nums__('7_9',_1) Integer
0 <= nums__('7_9',_10) Integer
0 <= nums__('7_9',_11) Integer
0 <= nums__('7_9',_12) Integer
0 <= nums__('7_9',_2) Integer
0 <= nums__('7_9',_3) Integer
0 <= nums__('7_9',_4) Integer
0 <= nums__('7_9',_5) Integer
0 <= nums__('7_9',_6) Integer
0 <= nums__('7_9',_7) Integer
0 <= nums__('7_9',_8) Integer
0 <= nums__('7_9',_9) Integer
0 <= nums__('9_11',_2) Integer
0 <= nums__('9_11',_4) Integer
```

```
In [16]: for t in time:
        model1 += (lpSum(numOperator[t,x] * table.loc[t,x] for x in [1,3,5,7,9,11]))
model1
```

```
Out[16]: Callcenter:
MINIMIZE
75*nums__('7_9',_1) + 45*nums__('7_9',_10) + 45*nums__('7_9',_11) + 45*nums__
('7_9',_12) + 75*nums__('7_9',_2) + 75*nums__('7_9',_3) + 75*nums__('7_9',_4) +
75*nums__('7_9',_5) + 75*nums__('7_9',_6) + 75*nums__('7_9',_7) + 75*nums__('7_
9',_8) + 45*nums__('7_9',_9) + 0
SUBJECT TO
('7_9',_Spanish'): 5 nums__('7_9',_4) >= 8.75

('9_11',_Spanish'): 5 nums__('9_11',_2) + 5 nums__('9_11',_4) >= 22.5

('11_13',_Spanish'): 5 nums__('11_13',_2) + 5 nums__('11_13',_4)
+ 5 nums__('11_13',_8) >= 20

('13_15',_Spanish'): 5 nums__('13_15',_2) + 5 nums__('13_15',_6)
+ 5 nums__('13_15',_8) >= 26.25

('15_17',_Spanish'): 5 nums__('15_17',_10) + 5 nums__('15_17',_6)
+ 5 nums__('15_17',_8) >= 23.75

('17_19',_Spanish'): 5 nums__('17_19',_10) + 5 nums__('17_19',_12)
+ 5 nums__('17_19',_6) >= 15

('19_21',_Spanish'): 5 nums__('19_21',_12) >= 8.75

('7_9',_English'): 5 nums__('7_9',_3) >= 26.25

('9_11',_English'): 5 nums__('9_11',_1) + 5 nums__('9_11',_3) >= 67.5

('11_13',_English'): 5 nums__('11_13',_1) + 5 nums__('11_13',_3)
+ 5 nums__('11_13',_7) >= 60

('13_15',_English'): 5 nums__('13_15',_1) + 5 nums__('13_15',_5)
+ 5 nums__('13_15',_7) >= 78.75

('15_17',_English'): 5 nums__('15_17',_5) + 5 nums__('15_17',_7)
+ 5 nums__('15_17',_9) >= 71.25

('17_19',_English'): 5 nums__('17_19',_11) + 5 nums__('17_19',_5)
+ 5 nums__('17_19',_9) >= 45

('19_21',_English'): 5 nums__('19_21',_11) >= 26.25

VARIABLES
0 <= nums__('11_13',_1) Integer
0 <= nums__('11_13',_2) Integer
0 <= nums__('11_13',_3) Integer
0 <= nums__('11_13',_4) Integer
0 <= nums__('11_13',_7) Integer
0 <= nums__('11_13',_8) Integer
0 <= nums__('13_15',_1) Integer
0 <= nums__('13_15',_2) Integer
0 <= nums__('13_15',_5) Integer
0 <= nums__('13_15',_6) Integer
0 <= nums__('13_15',_7) Integer
0 <= nums__('13_15',_8) Integer
0 <= nums__('15_17',_10) Integer
0 <= nums__('15_17',_5) Integer
0 <= nums__('15_17',_6) Integer
0 <= nums__('15_17',_7) Integer
0 <= nums__('15_17',_8) Integer
0 <= nums__('15_17',_9) Integer
0 <= nums__('17_19',_10) Integer
0 <= nums__('17_19',_11) Integer
0 <= nums__('17_19',_12) Integer
0 <= nums__('17_19',_5) Integer
0 <= nums__('17_19',_6) Integer
0 <= nums__('17_19',_9) Integer
```

```
0 <= nums__('19_21',_11) Integer
0 <= nums__('19_21',_12) Integer
0 <= nums__('7_9',_1) Integer
0 <= nums__('7_9',_10) Integer
0 <= nums__('7_9',_11) Integer
0 <= nums__('7_9',_12) Integer
0 <= nums__('7_9',_2) Integer
0 <= nums__('7_9',_3) Integer
0 <= nums__('7_9',_4) Integer
0 <= nums__('7_9',_5) Integer
0 <= nums__('7_9',_6) Integer
0 <= nums__('7_9',_7) Integer
0 <= nums__('7_9',_8) Integer
0 <= nums__('7_9',_9) Integer
0 <= nums__('9_11',_1) Integer
0 <= nums__('9_11',_2) Integer
0 <= nums__('9_11',_3) Integer
0 <= nums__('9_11',_4) Integer
```

```
In [17]: for x in job:
         for t in time:
             model1 += (7*numOperator[t,x] == lpSum(numOperator[t,x] for t in time),(>
model1
```

```
Out[17]: Callcenter:
MINIMIZE
75*nums__('7_9',_1) + 45*nums__('7_9',_10) + 45*nums__('7_9',_11) + 45*nums__('7_9',_12) + 75*nums__('7_9',_2) + 75*nums__('7_9',_3) + 75*nums__('7_9',_4) + 75*nums__('7_9',_5) + 75*nums__('7_9',_6) + 75*nums__('7_9',_7) + 75*nums__('7_9',_8) + 45*nums__('7_9',_9) + 0
SUBJECT TO
('7_9',_Spanish'): 5 nums__('7_9',_4) >= 8.75

('9_11',_Spanish'): 5 nums__('9_11',_2) + 5 nums__('9_11',_4) >= 22.5

('11_13',_Spanish'): 5 nums__('11_13',_2) + 5 nums__('11_13',_4) + 5 nums__('11_13',_8) >= 20

('13_15',_Spanish'): 5 nums__('13_15',_2) + 5 nums__('13_15',_6) + 5 nums__('13_15',_8) >= 26.25

('15_17',_Spanish'): 5 nums__('15_17',_10) + 5 nums__('15_17',_6) + 5 nums__('15_17',_8) >= 23.75
```

```
In [18]: model1.solve()
```

Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/ahedgehog/opt/anaconda3/lib/python3.9/site-packages/pulp/apis/./solverdir/cbc/osx/64/cbc /var/folders/v0/m66y75t50n7f4m1fbz2k_1840000gn/T/73fab82e20774fc9a8fbd21f0c6528b8-pulp.mps timeMode elapsed branch printingOptions all solution /var/folders/v0/m66y75t50n7f4m1fbz2k_1840000gn/T/73fab82e20774fc9a8fbd21f0c6528b8-pulp.sol (default strategy 1)

At line 2 NAME MODEL
At line 3 ROWS
At line 103 COLUMNS
At line 904 RHS
At line 1003 BOUNDS
At line 1088 ENDDATA
Problem MODEL has 98 rows, 84 columns and 620 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 2820 - 0.00 seconds
Cgl0003I 0 fixed, 84 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0004I processed model has 94 rows, 84 columns (84 integer (0 of which binary)) and 616 elements
Cutoff increment increased from 1e-05 to 14.9999
Cbc0012I Integer solution of 3015 found by DiveCoefficient after 0 iterations and 0 nodes (0.02 seconds)
Cbc0038I Full problem 94 rows 84 columns, reduced to 40 rows 32 columns
Cbc0031I 6 added rows had average density of 2.666667
Cbc0013I At root node, 42 cuts changed objective from 2895 to 3015 in 1 passes
Cbc0014I Cut generator 0 (Probing) - 0 row cuts average 0.0 elements, 23 column cuts (23 active) in 0.000 seconds - new frequency is 1
Cbc0014I Cut generator 1 (Gomory) - 4 row cuts average 3.5 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is 1
Cbc0014I Cut generator 2 (Knapsack) - 0 row cuts average 0.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 3 (Clique) - 0 row cuts average 0.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 4 (MixedIntegerRounding2) - 4 row cuts average 3.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is 1
Cbc0014I Cut generator 5 (FlowCover) - 0 row cuts average 0.0 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0014I Cut generator 6 (TwoMirCuts) - 34 row cuts average 3.4 elements, 0 column cuts (0 active) in 0.000 seconds - new frequency is -100
Cbc0001I Search completed - best objective 3015, took 0 iterations and 0 nodes (0.02 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 2895 to 3015
Probing was tried 1 times and created 23 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Gomory was tried 1 times and created 4 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Knapsack was tried 1 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Clique was tried 1 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 1 times and created 4 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 1 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 1 times and created 34 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 1 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:	3015.00000000
Enumerated nodes:	0
Total iterations:	0
Time (CPU seconds):	0.01
Time (Wallclock seconds):	0.02

Option for printingOptions changed from normal to all
Total time (CPU seconds): 0.01 (Wallclock seconds): 0.03

Out[18]: 1

```
In [19]: model1.objective.value()
```

Out[19]: 3015.0


```
In [20]: for v in model1.variables(): print(f"{v.name}: {v.varValue}")
```

```
nums__('11_13',_1): 8.0
nums__('11_13',_10): 2.0
nums__('11_13',_11): 6.0
nums__('11_13',_12): 2.0
nums__('11_13',_2): 3.0
nums__('11_13',_3): 6.0
nums__('11_13',_4): 2.0
nums__('11_13',_5): 0.0
nums__('11_13',_6): 0.0
nums__('11_13',_7): 8.0
nums__('11_13',_8): 3.0
nums__('11_13',_9): 7.0
nums__('13_15',_1): 8.0
nums__('13_15',_10): 2.0
nums__('13_15',_11): 6.0
nums__('13_15',_12): 2.0
nums__('13_15',_2): 3.0
nums__('13_15',_3): 6.0
nums__('13_15',_4): 2.0
nums__('13_15',_5): 0.0
nums__('13_15',_6): 0.0
nums__('13_15',_7): 8.0
nums__('13_15',_8): 3.0
nums__('13_15',_9): 7.0
nums__('15_17',_1): 8.0
nums__('15_17',_10): 2.0
nums__('15_17',_11): 6.0
nums__('15_17',_12): 2.0
nums__('15_17',_2): 3.0
nums__('15_17',_3): 6.0
nums__('15_17',_4): 2.0
nums__('15_17',_5): 0.0
nums__('15_17',_6): 0.0
nums__('15_17',_7): 8.0
nums__('15_17',_8): 3.0
nums__('15_17',_9): 7.0
nums__('17_19',_1): 8.0
nums__('17_19',_10): 2.0
nums__('17_19',_11): 6.0
nums__('17_19',_12): 2.0
nums__('17_19',_2): 3.0
nums__('17_19',_3): 6.0
nums__('17_19',_4): 2.0
nums__('17_19',_5): 0.0
nums__('17_19',_6): 0.0
nums__('17_19',_7): 8.0
nums__('17_19',_8): 3.0
nums__('17_19',_9): 7.0
nums__('19_21',_1): 8.0
nums__('19_21',_10): 2.0
nums__('19_21',_11): 6.0
nums__('19_21',_12): 2.0
nums__('19_21',_2): 3.0
nums__('19_21',_3): 6.0
nums__('19_21',_4): 2.0
nums__('19_21',_5): 0.0
nums__('19_21',_6): 0.0
nums__('19_21',_7): 8.0
nums__('19_21',_8): 3.0
nums__('19_21',_9): 7.0
nums__('7_9',_1): 8.0
nums__('7_9',_10): 2.0
nums__('7_9',_11): 6.0
nums__('7_9',_12): 2.0
nums__('7_9',_2): 3.0
nums__('7_9',_3): 6.0
nums__('7_9',_4): 2.0
nums__('7_9',_5): 0.0
nums__('7_9',_6): 0.0
nums__('7_9',_7): 8.0
nums__('7_9',_8): 3.0
```

```
nums__('7_9',_9): 7.0
nums__('9_11',_1): 8.0
nums__('9_11',_10): 2.0
nums__('9_11',_11): 6.0
nums__('9_11',_12): 2.0
nums__('9_11',_2): 3.0
nums__('9_11',_3): 6.0
nums__('9_11',_4): 2.0
nums__('9_11',_5): 0.0
nums__('9_11',_6): 0.0
nums__('9_11',_7): 8.0
nums__('9_11',_8): 3.0
nums__('9_11',_9): 7.0
```

```
In [33]: # the min cost will be 3015, which
# English-speaking operator who works from 7 to 3 and does paperwork at the start
# Spanish-speaking operator who works from 7 to 3 and does paperwork at the start
# English-speaking operator who works from 7 to 3 and does paperwork at the end c
# Spanish-speaking operator who works from 7 to 3 and does paperwork at the end c
# English-speaking operator who works from 11 to 7 and does paperwork at the star
# Spanish-speaking operator who works from 11 to 7 and does paperwork at the star
# English-speaking operator who works from 11 to 7 and does paperwork at the end
# Spanish-speaking operator who works from 11 to 7 and does paperwork at the end
# English-speaking operator who works part-time from 3 to 7 will be 7
# Spanish-speaking operator who works part-time from 3 to 7 will be 2
# English-speaking operator who works part-time from 5 to 9 will be 6
# Spanish-speaking operator who works part-time from 5 to 9 will be 2
```

```
In [ ]: ## Question C
```

```
In [21]: model2 = LpProblem("Callcenter2", LpMinimize)
```

```
In [ ]: # possibility
#1:English-speaking operator who works from 7 to 3 and does paperwork at the star
#2:Spanish-speaking operator who works from 7 to 3 and does paperwork at the star
#3:English-speaking operator who works from 7 to 3 and does paperwork at the end
#4:Spanish-speaking operator who works from 7 to 3 and does paperwork at the end
#5:English-speaking operator who works from 11 to 7 and does paperwork at the sto
#6:Spanish-speaking operator who works from 11 to 7 and does paperwork at the sto
#7:English-speaking operator who works from 11 to 7 and does paperwork at the end
#8:Spanish-speaking operator who works from 11 to 7 and does paperwork at the end
#9:English-speaking operator who works part-time from 3 to 7
#10:Spanish-speaking operator who works part-time from 3 to 7
#11:English-speaking operator who works part-time from 5 to 9
#12:Spanish-speaking operator who works part-time from 5 to 9
```

```
In [22]: time = ["7-9","9-11","11-13","13-15","15-17","17-19","19-21"]
job = ['fulltime7-3-1','fulltime7-3-4','fulltime11-7-1','fulltime11-7-4','parttin
```

```
In [23]: choice = {
    'fulltime7-3-1': [0,5,5,5,0,0,0],
    'fulltime7-3-4': [5,5,5,0,0,0,0],
    'fulltime11-7-1': [0,0,0,5,5,5,0],
    'fulltime11-7-4': [0,0,5,5,5,0,0],
    'parttime3-7': [0,0,0,0,5,5,0],
    'parttime5-9': [0,0,0,0,0,5,5]
}

table2 = pd.DataFrame(choice, index = time)
table2
```

Out[23]:

	fulltime7-3-1	fulltime7-3-4	fulltime11-7-1	fulltime11-7-4	parttime3-7	parttime5-9
7-9	0	5	0	0	0	0
9-11	5	5	0	0	0	0
11-13	5	5	0	5	0	0
13-15	5	0	5	5	0	0
15-17	0	0	5	5	5	0
17-19	0	0	5	0	5	5
19-21	0	0	0	0	0	5

```
In [24]: assignment2 = LpVariable.dicts('nums_', ((t,j) for t in time for j in job), lowBc
assignment2
```

```
Out[24]: {('7-9', 'fulltime7-3-1'): nums__('7_9',_fulltime7_3_1'),
('7-9', 'fulltime7-3-4'): nums__('7_9',_fulltime7_3_4'),
('7-9', 'fulltime11-7-1'): nums__('7_9',_fulltime11_7_1'),
('7-9', 'fulltime11-7-4'): nums__('7_9',_fulltime11_7_4'),
('7-9', 'parttime3-7'): nums__('7_9',_parttime3_7'),
('7-9', 'parttime5-9'): nums__('7_9',_parttime5_9'),
('9-11', 'fulltime7-3-1'): nums__('9_11',_fulltime7_3_1'),
('9-11', 'fulltime7-3-4'): nums__('9_11',_fulltime7_3_4'),
('9-11', 'fulltime11-7-1'): nums__('9_11',_fulltime11_7_1'),
('9-11', 'fulltime11-7-4'): nums__('9_11',_fulltime11_7_4'),
('9-11', 'parttime3-7'): nums__('9_11',_parttime3_7'),
('9-11', 'parttime5-9'): nums__('9_11',_parttime5_9'),
('11-13', 'fulltime7-3-1'): nums__('11_13',_fulltime7_3_1'),
('11-13', 'fulltime7-3-4'): nums__('11_13',_fulltime7_3_4'),
('11-13', 'fulltime11-7-1'): nums__('11_13',_fulltime11_7_1'),
('11-13', 'fulltime11-7-4'): nums__('11_13',_fulltime11_7_4'),
('11-13', 'parttime3-7'): nums__('11_13',_parttime3_7'),
('11-13', 'parttime5-9'): nums__('11_13',_parttime5_9'),
('13-15', 'fulltime7-3-1'): nums__('13_15',_fulltime7_3_1'),
('13-15', 'fulltime7-3-4'): nums__('13_15',_fulltime7_3_4'),
('13-15', 'fulltime11-7-1'): nums__('13_15',_fulltime11_7_1'),
('13-15', 'fulltime11-7-4'): nums__('13_15',_fulltime11_7_4'),
('13-15', 'parttime3-7'): nums__('13_15',_parttime3_7'),
('13-15', 'parttime5-9'): nums__('13_15',_parttime5_9'),
('15-17', 'fulltime7-3-1'): nums__('15_17',_fulltime7_3_1'),
('15-17', 'fulltime7-3-4'): nums__('15_17',_fulltime7_3_4'),
('15-17', 'fulltime11-7-1'): nums__('15_17',_fulltime11_7_1'),
('15-17', 'fulltime11-7-4'): nums__('15_17',_fulltime11_7_4'),
('15-17', 'parttime3-7'): nums__('15_17',_parttime3_7'),
('15-17', 'parttime5-9'): nums__('15_17',_parttime5_9'),
('17-19', 'fulltime7-3-1'): nums__('17_19',_fulltime7_3_1'),
('17-19', 'fulltime7-3-4'): nums__('17_19',_fulltime7_3_4'),
('17-19', 'fulltime11-7-1'): nums__('17_19',_fulltime11_7_1'),
('17-19', 'fulltime11-7-4'): nums__('17_19',_fulltime11_7_4'),
('17-19', 'parttime3-7'): nums__('17_19',_parttime3_7'),
('17-19', 'parttime5-9'): nums__('17_19',_parttime5_9'),
('19-21', 'fulltime7-3-1'): nums__('19_21',_fulltime7_3_1'),
('19-21', 'fulltime7-3-4'): nums__('19_21',_fulltime7_3_4'),
('19-21', 'fulltime11-7-1'): nums__('19_21',_fulltime11_7_1'),
('19-21', 'fulltime11-7-4'): nums__('19_21',_fulltime11_7_4'),
('19-21', 'parttime3-7'): nums__('19_21',_parttime3_7'),
('19-21', 'parttime5-9'): nums__('19_21',_parttime5_9')}
```

```
In [25]: cost2 = {
    'fulltime7-3-1': 75,
    'fulltime7-3-4': 75,
    'fulltime11-7-1': 75,
    'fulltime11-7-4': 75,
    'parttime3-7': 45,
    'parttime5-9': 45
}

cost2
```

Out[25]: {'fulltime7-3-1': 75,
'fulltime7-3-4': 75,
'fulltime11-7-1': 75,
'fulltime11-7-4': 75,
'parttime3-7': 45,
'parttime5-9': 45}

```
In [27]: model2 += lpSum(assignment2['7-9',x] * cost2[x] for x in job)
model2
```

Out[27]: Callcenter2:
MINIMIZE
75*nums__('7_9',_fulltime11_7_1') + 75*nums__('7_9',_fulltime11_7_4') + 75*nu
ms__('7_9',_fulltime7_3_1') + 75*nums__('7_9',_fulltime7_3_4') + 45*nums__('7
_9',_parttime3_7') + 45*nums__('7_9',_parttime5_9') + 0
VARIABLES
0 <= nums__('7_9',_fulltime11_7_1') Integer
0 <= nums__('7_9',_fulltime11_7_4') Integer
0 <= nums__('7_9',_fulltime7_3_1') Integer
0 <= nums__('7_9',_fulltime7_3_4') Integer
0 <= nums__('7_9',_parttime3_7') Integer
0 <= nums__('7_9',_parttime5_9') Integer

```
In [28]: for t in time:
          model2 += (lpSum(assignment2[t,x] * table2.loc[t,x] for x in job) >= df.1
model2
```

```
Out[28]: Callcenter2:
MINIMIZE
75*nums__('7_9','fulltime11_7_1') + 75*nums__('7_9','fulltime11_7_4') + 75*nu
ms__('7_9','fulltime7_3_1') + 75*nums__('7_9','fulltime7_3_4') + 45*nums__('7
_9','parttime3_7') + 45*nums__('7_9','parttime5_9') + 0
SUBJECT TO
('7_9',_12): 5 nums__('7_9','fulltime7_3_4') >= 35

('9_11',_12): 5 nums__('9_11','fulltime7_3_1')
+ 5 nums__('9_11','fulltime7_3_4') >= 90

('11_13',_12): 5 nums__('11_13','fulltime11_7_4')
+ 5 nums__('11_13','fulltime7_3_1') + 5 nums__('11_13','fulltime7_3_4')
>= 80

('13_15',_12): 5 nums__('13_15','fulltime11_7_1')
+ 5 nums__('13_15','fulltime11_7_4') + 5 nums__('13_15','fulltime7_3_1')
>= 105

('15_17',_12): 5 nums__('15_17','fulltime11_7_1')
+ 5 nums__('15_17','fulltime11_7_4') + 5 nums__('15_17','parttime3_7')
>= 95

('17_19',_12): 5 nums__('17_19','fulltime11_7_1')
+ 5 nums__('17_19','parttime3_7') + 5 nums__('17_19','parttime5_9') >= 60

('19_21',_12): 5 nums__('19_21','parttime5_9') >= 35

VARIABLES
0 <= nums__('11_13','fulltime11_7_4') Integer
0 <= nums__('11_13','fulltime7_3_1') Integer
0 <= nums__('11_13','fulltime7_3_4') Integer
0 <= nums__('13_15','fulltime11_7_1') Integer
0 <= nums__('13_15','fulltime11_7_4') Integer
0 <= nums__('13_15','fulltime7_3_1') Integer
0 <= nums__('15_17','fulltime11_7_1') Integer
0 <= nums__('15_17','fulltime11_7_4') Integer
0 <= nums__('15_17','parttime3_7') Integer
0 <= nums__('17_19','fulltime11_7_1') Integer
0 <= nums__('17_19','parttime3_7') Integer
0 <= nums__('17_19','parttime5_9') Integer
0 <= nums__('19_21','parttime5_9') Integer
0 <= nums__('7_9','fulltime11_7_1') Integer
0 <= nums__('7_9','fulltime11_7_4') Integer
0 <= nums__('7_9','fulltime7_3_1') Integer
0 <= nums__('7_9','fulltime7_3_4') Integer
0 <= nums__('7_9','parttime3_7') Integer
0 <= nums__('7_9','parttime5_9') Integer
0 <= nums__('9_11','fulltime7_3_1') Integer
0 <= nums__('9_11','fulltime7_3_4') Integer
```

```
In [29]: for x in job:
        for t in time:
            model2 += (7*assignment2[t,x] == lpSum(assignment2[t,x] for t in time),(>

model2
```

```
Out[29]: Callcenter2:
MINIMIZE
75*nums__('7_9','fulltime11_7_1') + 75*nums__('7_9','fulltime11_7_4') + 75
*nums__('7_9','fulltime7_3_1') + 75*nums__('7_9','fulltime7_3_4') + 45*num
s__('7_9','parttime3_7') + 45*nums__('7_9','parttime5_9') + 0
SUBJECT TO
('7_9',_12): 5 nums__('7_9','fulltime7_3_4') >= 35

('9_11',_12): 5 nums__('9_11','fulltime7_3_1')
+ 5 nums__('9_11','fulltime7_3_4') >= 90

('11_13',_12): 5 nums__('11_13','fulltime11_7_4')
+ 5 nums__('11_13','fulltime7_3_1') + 5 nums__('11_13','fulltime7_3_4')
>= 80

('13_15',_12): 5 nums__('13_15','fulltime11_7_1')
+ 5 nums__('13_15','fulltime11_7_4') + 5 nums__('13_15','fulltime7_3_1')
>= 105

('15_17',_12): 5 nums__('15_17','fulltime11_7_1') + 5 nums__('15_17','fulltime11_7_4') + 5 nums__('15_17','fulltime7_3_1') + 5 nums__('15_17','fulltime7_3_4') + 5 nums__('15_17','parttime3_7') + 5 nums__('15_17','parttime5_9') >= 110
```

```
In [30]: model2.solve()
```

Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/ahedgehog/opt/anaconda3/lib/python3.9/site-packages/pulp/apis/./solverdir/cbc/osx/64/cbc /var/folders/v0/m66y75t50n7f4m1fbz2k_1840000gn/T/9ef9699e807348cc903163a6cb534e4a-pulp.mps timeMode elapsed branch printingOptions all solution /var/folders/v0/m66y75t50n7f4m1fbz2k_1840000gn/T/9ef9699e807348cc903163a6cb534e4a-pulp.sol (default strategy 1)
At line 2 NAME MODEL
At line 3 ROWS
At line 54 COLUMNS
At line 455 RHS
At line 505 BOUNDS
At line 548 ENDDATA
Problem MODEL has 49 rows, 42 columns and 310 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 2820 - 0.00 seconds
Cgl0003I 0 fixed, 42 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0004I processed model has 47 rows, 42 columns (42 integer (0 of which binary)) and 308 elements
Cutoff increment increased from 1e-05 to 14.9999
Cbc0012I Integer solution of 2820 found by DiveCoefficient after 0 iterations and 0 nodes (0.01 seconds)
Cbc0001I Search completed - best objective 2820, took 0 iterations and 0 nodes (0.01 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 2820 to 2820
Probing was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Gomory was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Knapsack was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
Clique was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
FlowCover was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cuts (0.000 seconds)

Result - Optimal solution found

Objective value:	2820.00000000
Enumerated nodes:	0
Total iterations:	0
Time (CPU seconds):	0.01
Time (Wallclock seconds):	0.02

Option for printingOptions changed from normal to all
Total time (CPU seconds): 0.01 (Wallclock seconds): 0.02

Out[30]: 1

```
In [31]: model2.objective.value()
```

Out[31]: 2820.0

```
In [32]: for v in model2.variables(): print(f"{v.name}: {v.varValue}")
```

```
nums__('11_13',_ 'fulltime11_7_1'): 10.0
nums__('11_13',_ 'fulltime11_7_4'): 0.0
nums__('11_13',_ 'fulltime7_3_1'): 11.0
nums__('11_13',_ 'fulltime7_3_4'): 7.0
nums__('11_13',_ 'parttime3_7'): 9.0
nums__('11_13',_ 'parttime5_9'): 7.0
nums__('13_15',_ 'fulltime11_7_1'): 10.0
nums__('13_15',_ 'fulltime11_7_4'): 0.0
nums__('13_15',_ 'fulltime7_3_1'): 11.0
nums__('13_15',_ 'fulltime7_3_4'): 7.0
nums__('13_15',_ 'parttime3_7'): 9.0
nums__('13_15',_ 'parttime5_9'): 7.0
nums__('15_17',_ 'fulltime11_7_1'): 10.0
nums__('15_17',_ 'fulltime11_7_4'): 0.0
nums__('15_17',_ 'fulltime7_3_1'): 11.0
nums__('15_17',_ 'fulltime7_3_4'): 7.0
nums__('15_17',_ 'parttime3_7'): 9.0
nums__('15_17',_ 'parttime5_9'): 7.0
nums__('17_19',_ 'fulltime11_7_1'): 10.0
nums__('17_19',_ 'fulltime11_7_4'): 0.0
nums__('17_19',_ 'fulltime7_3_1'): 11.0
nums__('17_19',_ 'fulltime7_3_4'): 7.0
nums__('17_19',_ 'parttime3_7'): 9.0
nums__('17_19',_ 'parttime5_9'): 7.0
nums__('19_21',_ 'fulltime11_7_1'): 10.0
nums__('19_21',_ 'fulltime11_7_4'): 0.0
nums__('19_21',_ 'fulltime7_3_1'): 11.0
nums__('19_21',_ 'fulltime7_3_4'): 7.0
nums__('19_21',_ 'parttime3_7'): 9.0
nums__('19_21',_ 'parttime5_9'): 7.0
nums__('7_9',_ 'fulltime11_7_1'): 10.0
nums__('7_9',_ 'fulltime11_7_4'): 0.0
nums__('7_9',_ 'fulltime7_3_1'): 11.0
nums__('7_9',_ 'fulltime7_3_4'): 7.0
nums__('7_9',_ 'parttime3_7'): 9.0
nums__('7_9',_ 'parttime5_9'): 7.0
nums__('9_11',_ 'fulltime11_7_1'): 10.0
nums__('9_11',_ 'fulltime11_7_4'): 0.0
nums__('9_11',_ 'fulltime7_3_1'): 11.0
nums__('9_11',_ 'fulltime7_3_4'): 7.0
nums__('9_11',_ 'parttime3_7'): 9.0
nums__('9_11',_ 'parttime5_9'): 7.0
```

```
In [ ]: # so the new cost is 2820
# Full time worker from 11 to 7 who does paperwork at the start of the shift will
# Full time worker from 11 to 7 who does paperwork at the end of the shift will b
# Full time worker from 7 to 3 who does paperwork at the start of the shift will
# Full time worker from 7 to 3 who does paperwork at the end of the shift will be
# Part time worker from 3 to 7 will be 9
# Part time worker from 5 to 9 will be 7
```