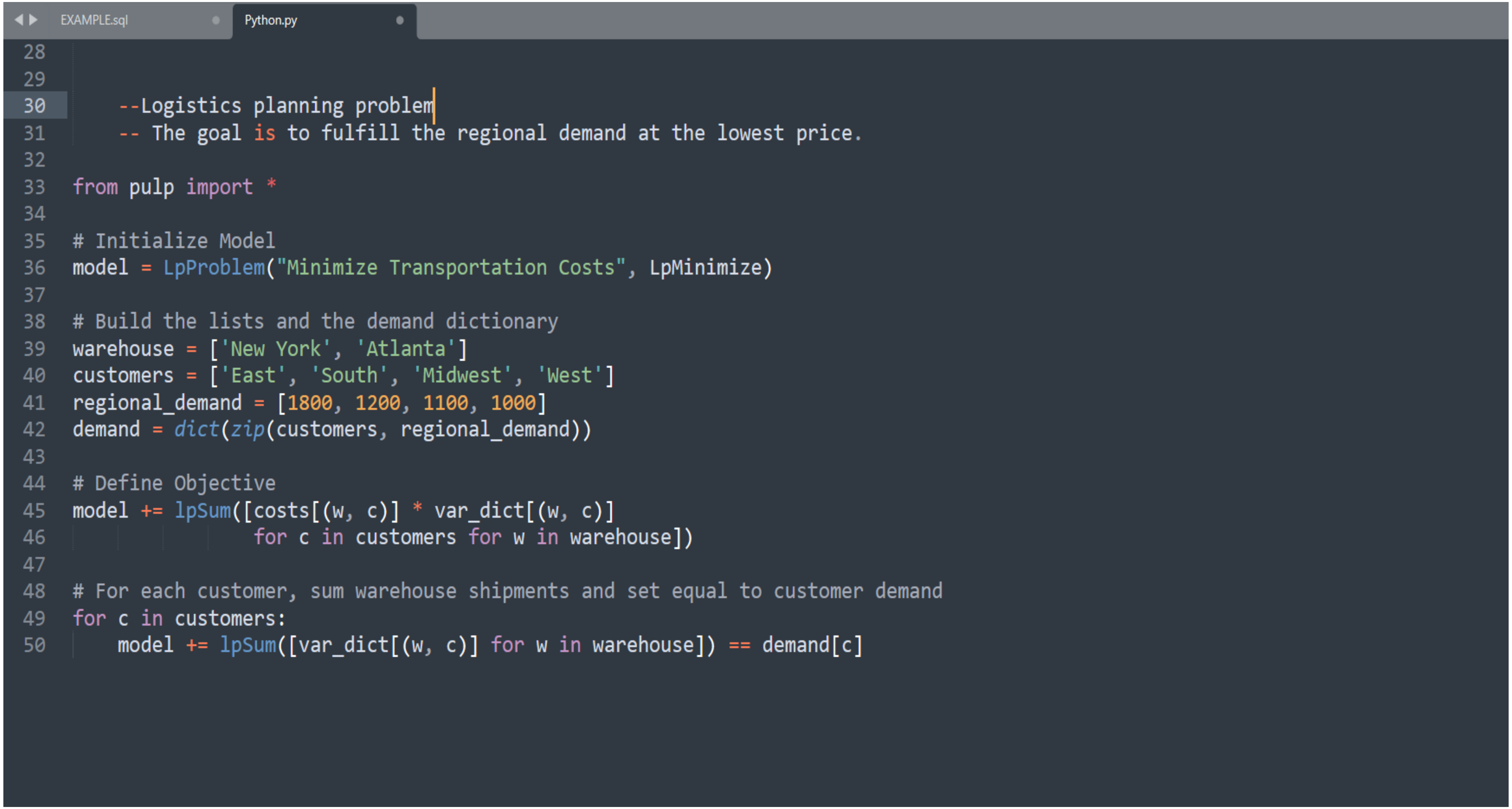
Supply Chain Analytics in Python

- 1- **Logistics planning problem**
- 2- **Scheduling workers problem**

- There are two warehouse locations (New York, and Atlanta), and four regional customer locations (East, South, Midwest, West). The expected demand next month for East it is 1,800, for South it is 1,200, for the Midwest it is 1,100, and for West it is 1000. The cost for shipping each of the warehouse locations to the regional customer's is listed in the table. The goal is to fulfill the regional demand at the lowest price.

Customer	New York	Atlanta
East	$211	$232
South	$232	$212
Midwest	$240	$230
West	$300	$280

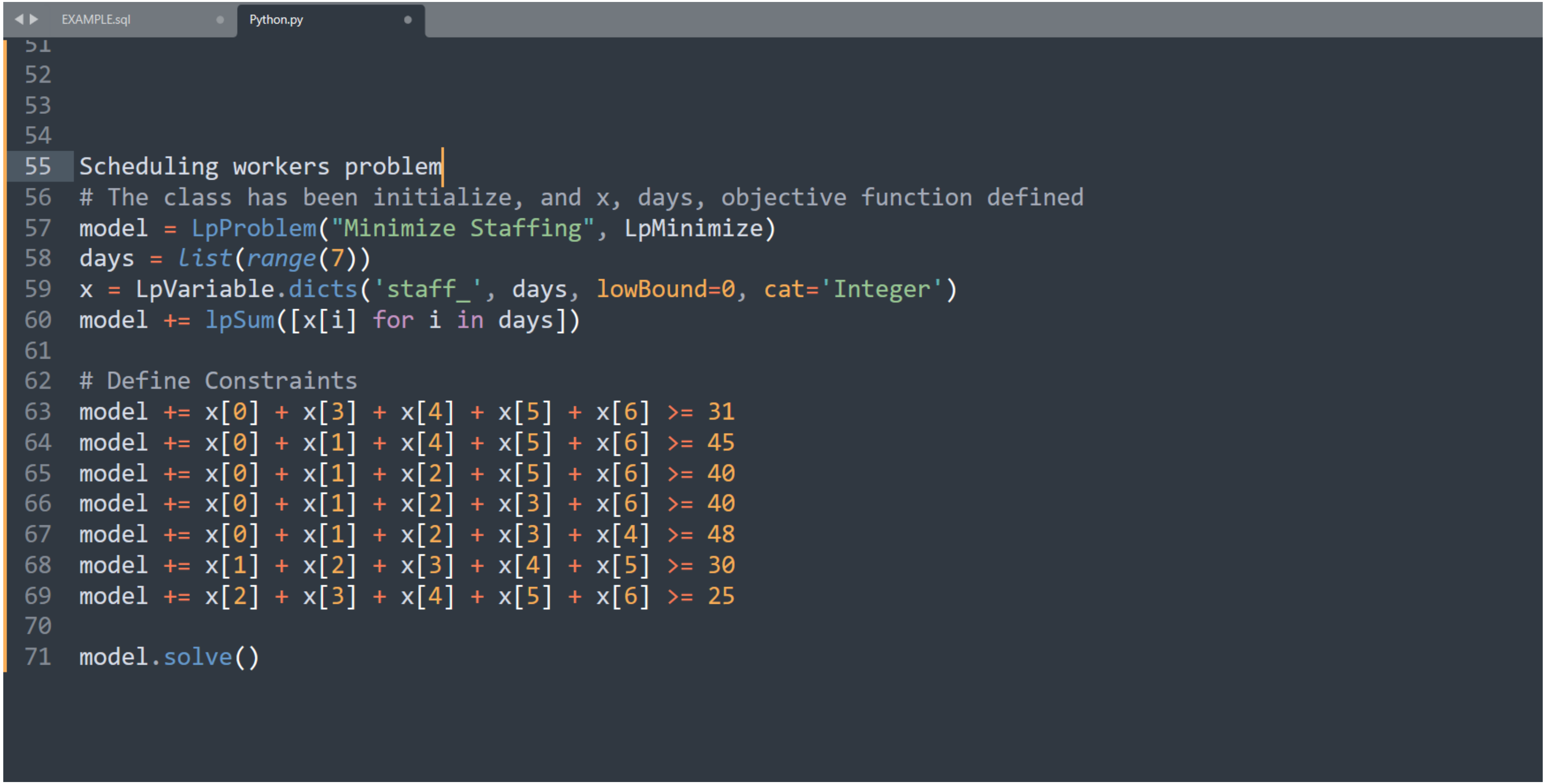


```


    --Logistics planning problem
    -- The goal is to fulfill the regional demand at the lowest price.

from pulp import *

# Initialize Model
model = LpProblem("Minimize Transportation Costs", LpMinimize)

# Build the lists and the demand dictionary
warehouse = ['New York', 'Atlanta']
customers = ['East', 'South', 'Midwest', 'West']
regional_demand = [1800, 1200, 1100, 1000]
demand = dict(zip(customers, regional_demand))

# Define Objective
model += lpSum([costs[(w, c)] * var_dict[(w, c)]
                for c in customers for w in warehouse])

# For each customer, sum warehouse shipments and set equal to customer demand
for c in customers:
    model += lpSum([var_dict[(w, c)] for w in warehouse]) == demand[c]
```

Scheduling workers problem

We are looking to hire workers to work in a warehouse. Each worker is expected to work 5 consecutive days and then have two days off. The chart has the estimated number of workers we will need each day. The goal is to hire the minimum number of workers to handle the workload for each day.

Day of Week	Employees Needed
0 = Monday	31
1 = Tuesday	45
2 = Wednesday	40
3 = Thursday	40
4 = Friday	48
5 = Saturday	30
6 = Sunday	25



```
Scheduling workers problem
# The class has been initialize, and x, days, objective function defined
model = LpProblem("Minimize Staffing", LpMinimize)
days = list(range(7))
x = LpVariable.dicts('staff_', days, lowBound=0, cat='Integer')
model += lpSum([x[i] for i in days])

# Define Constraints
model += x[0] + x[3] + x[4] + x[5] + x[6] >= 31
model += x[0] + x[1] + x[4] + x[5] + x[6] >= 45
model += x[0] + x[1] + x[2] + x[5] + x[6] >= 40
model += x[0] + x[1] + x[2] + x[3] + x[6] >= 40
model += x[0] + x[1] + x[2] + x[3] + x[4] >= 48
model += x[1] + x[2] + x[3] + x[4] + x[5] >= 30
model += x[2] + x[3] + x[4] + x[5] + x[6] >= 25

model.solve()
```