

```
In [2]: from gamspy import (Container, Variable, Equation, Model, Set, Parameter, Sum)
import numpy as np
```

```
In [6]: b = Container()

#SETS
t = Set(b, 't', records=['Pilsner', 'Doublebock', 'Stout'])
r = Set(b, 'r', records=['Hops', 'Malt', 'Barley'])
o = Parameter(b, 'o', [t], records=np.array([60, 40, 80]))
u = Parameter(b, 'u', [r], records=np.array([12, 9, 16]))
p = Parameter(b, 'p', [t, r], records=np.array([[2, 3, 2], [2, 2, 1], [2, 1, 1]]))

#VARIABLES
x = Variable(b, "x", "positive", domain=[t], description="beer")

#EQUATIONS
resource_con = Equation(b, 'resource_con', domain=[r], description="resource")
resource_con[r]=Sum(t, p[t, r]*x[t]) <= u[r]

ff = Model(b,
            name='ff',
            equations=b.getEquations(),
            problem=Problem.LP,
            sense=Sense.MAX,
            objective=Sum(t, o[t]*x[t]))
```

```
In [7]: ff.solve(options = Options(equation_listing_limit=100))
print("Objective Function Value: ", round(ff.objective_value, 4), "\n")
print("x: ", x.toList())
print("status: ", ff.status)
print("solver status: ", ff.solve_status)
print(ff.getEquationListing())
```

Objective Function Value: 440.0

```
x: [('Pilsner', 0.0), ('Doublebock', 1.0000000000000002), ('Stout', 5.0)]
status: ModelStatus.OptimalGlobal
solver status: SolveStatus.NormalCompletion
resource_con(Hops).. 2*x(Pilsner) + 2*x(Doublebock) + 2*x(Stout) =L= 12 ;
(LHS = 0)
resource_con(Malt).. 3*x(Pilsner) + 2*x(Doublebock) + x(Stout) =L= 9 ; (LHS
= 0)
resource_con(Barley).. 2*x(Pilsner) + x(Doublebock) + 3*x(Stout) =L= 16 ;
(LHS = 0)
ff_objective.. 60*x(Pilsner) + 40*x(Doublebock) + 80*x(Stout) - ff_objectiv
e_variable =E= 0 ; (LHS = 0)
```