```
!pip install -q amplpy
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 42.4 MB/s eta 0:00:00
```

```
from amplpy import AMPL, tools

ampl = tools.ampl_notebook(

    modules=["coin"],

    license_uuid="a1e9c56f-fff9-481c-a7cc-9e56e8a980b4")
```

```
Licensed to AMPL Community Edition License for <howardsarahv@gmail.com>.
```

```
#Question 1: Total supply chain costs and amount of demand that can be satisfied optimall
%%writefile abc.mod
set S = {"KC", "SA"}; #supply nodes
set D = {"T", "SH", "MC", "M", "L", "CA", "AT"}; #demand nodes

param C{S, D}; #cost matrix
param Supply{S}; #Supply capacity of the supply nodes
param Demand{D}; #demand at nodes in set D

#Variables
var x{S,D} >=0; # transportation quantity from nodes set S to demand set D

minimize Z: sum{i in S, j in D} C[i,j]*x[i,j]; #total Transportation cost

#Constraints
s.t. supply{i in S}: sum{j in D} x[i,j]= Supply[i]; # supply constraints
s.t. demand{j in D}: sum{i in S} x[i,j]<= Demand[j]; # demand constraints
```

```
Overwriting abc.mod
```

```
#Question 1: Total supply chain costs and amount of demand that can be satisfied optimall

%%writefile abc.dat

param C: T SH MC M L CA AT=
KC 1.79 2.13 1.76 2.34 1.86 1.90 1.82
SA 2.13 2.03 1.58 1.80 2.14 1.26 1.76;

param Supply:=
KC 60000
SA 60000;

param Demand:=
T 5000
```

```
SH 50000
MC 4000
M 6000
L 40000
CA 10000
AT 60000;
```

```
Writing abc.dat
```

```
#Question 1: Total supply chain costs and amount of demand that can be satisfied optimall
%%ampl_eval
reset;
# Model File
model abc.mod;
data abc.dat;
#Calling Optimization Engine and Optimizing
option solver cbc;
solve;
#Display Results
display Z, x;
```

```
cbc 2.10.10: optimal solution; objective 210570
0 simplex iterations
Z = 210570

x :=
KC AT   15000
KC CA       0
KC L    40000
KC M        0
KC MC       0
KC SH       0
KC T     5000
SA AT   45000
SA CA   10000
SA L        0
SA M     1000
SA MC    4000
SA SH       0
SA T        0
;
```

```
#Q2 Capacity Expansion Kansas City or Santiago
%%writefile expansion.mod
set S = {"KC", "SA"}; #supply nodes
set D = {"T", "SH", "MC", "M", "L", "CA", "AT"}; #demand nodes

#Variables
var x{S,D} >=0 integer; # transportation quantity from nodes set S to demand set D
```

```
var y{S} binary; #fixed cost of expanding

#Parameters
param C{S, D}; #cost matrix for transportation
param Supply{S}; #Supply capacity of the supply nodes
param Demand{D}; #demand at nodes in set D
param FC{S}; #one time fixed cost at one plant or another in set S


minimize Z: sum{i in S} y[i]*FC[i]+ sum {i in S, j in D} C[i,j]*x[i,j]; #total Transportat

#Constraints
s.t. supply{i in S}: sum{j in D} x[i,j]= 60000+50000*y[i]; # supply constraints
s.t. demand{j in D}: sum{i in S} x[i,j]<= Demand[j]; # demand constraints
s.t. KCorSA: y["KC"] + y["SA"]>=1; #expanding one plant or the other
```

```
Overwriting expansion.mod
```

```
%%writefile expansion.dat

param C: T SH MC M L CA AT=
KC 1.79 2.13 1.76 2.34 1.86 1.90 1.82
SA 2.13 2.03 1.58 1.80 2.14 1.26 1.76;

param Supply:=
KC 60000
SA 60000;

param Demand:=
T 5000
SH 50000
MC 4000
M 6000
L 40000
CA 10000
AT 60000;

param FC:=
KC 2590000
SA 2061000;
```

```
Overwriting expansion.dat
```

```
#Q2 Capacity Expansion Kansas City
%%ampl_eval
reset;
# Model File
model expansion.mod;
data expansion.dat;
```

```
#Calling Optimization Engine and Optimizing
option solver cbc;
solve;
#Display Results
display Z, x, y;
```

```
cbc 2.10.10: optimal solution; objective 2371920
0 simplex iterations
Z = 2371920

x :=
KC AT   15000
KC CA       0
KC L    40000
KC M        0
KC MC       0
KC SH       0
KC T     5000
SA AT   45000
SA CA   10000
SA L        0
SA M     6000
SA MC    4000
SA SH   45000
SA T        0
;

y [*] :=
KC  0
SA  1
;
```

```
#Q3: Where should we put the new plant? What is the associated cost?
%%writefile newplant.mod
set S= {"AUC", "BIR", "FRA", "MUM", "SIN"}; #supply nodes
set D= {"SH", "M"}; #demand nodes

#Variables
var x{S,D} >=0 integer; # transportation quantity from nodes set S to demand set D
var y{S} binary; #fixed cost of expanding

#Parameters
param C{S, D}; #cost matrix for transportation
param Supply{S}; #Supply capacity of the supply nodes
param Demand{D}; #demand at nodes in set D
param FC{S}; #one time fixed cost at one plant or another in set S

minimize Z: sum{i in S} y[i]*FC[i]+ sum {i in S, j in D} C[i,j]*x[i,j]; #total shipping +

#Constraints
s.t. oneormore: y["AUC"]+y["BIR"]+y["FRA"]+y["MUM"]+y["SIN"]>=1; #adding one or more plan
s.t. supply{i in S}: sum{j in D} x[i,j]<= Supply[i]*y[i]; # supply constraints
```

```
s.t. demand{j in D}: sum{i in S} x[i,j]>= Demand[j]; # demand constraints
```

```
Overwriting newplant.mod
```

```
#Q3
%%writefile newplant.dat

param C: SH M:=
AUC 1.18 0.91
BIR 1.6 1.52
FRA 1.65 1.73
MUM 1.21 1.38
SIN 1.44 1.43;

param Supply:=
AUC 15000
BIR 15000
FRA 20000
MUM 25000
SIN 20000;

param Demand:=
SH 50000
M 6000;

param FC:=
AUC 917000
BIR 962000
FRA 1093000
MUM 959000
SIN 1058000;
```

```
Overwriting newplant.dat
```

```
%%ampl_eval
reset;
# Model File
model newplant.mod;
data newplant.dat;
#Calling Optimization Engine and Optimizing
option solver cbc;
solve;
#Display Results
display Z, x, y;
```

```
cbc 2.10.10: optimal solution; objective 3003370
7 simplex iterations
7 barrier iterations
```

```
Z = 3003370

x :=
AUC M     6000
AUC SH    9000
BIR M        0
BIR SH       0
FRA M        0
FRA SH       0
MUM M        0
MUM SH   25000
SIN M        0
SIN SH   16000
;

y [*] :=
AUC  1
BIR  0
FRA  0
MUM  1
SIN  1
;
```

```
#Q4: Endurance Round Shanghai 20/70/90
%%writefile newshanghai.mod
set S= {"AUC", "BIR", "FRA", "MUM", "SIN"}; #supply nodes
set D= {"SH", "M"}; #demand nodes

#Variables
var x{S,D} >=0 integer; # transportation quantity from nodes set S to demand set D
var y{S} binary; #fixed cost of expanding

#Parameters
param C{S, D}; #cost matrix for transportation
param Supply{S}; #Supply capacity of the supply nodes
param Demand{D}; #demand at nodes in set D
param FC{S}; #one time fixed cost at one plant or another in set S

minimize Z: sum{i in S} y[i]*FC[i]+ sum {i in S, j in D} C[i,j]*x[i,j]; #total shipping + f

#Constraints
s.t. oneormore: y["AUC"]+y["BIR"]+y["FRA"]+y["MUM"]+y["SIN"]>=1; #adding one or more plants
s.t. supply{i in S}: sum{j in D} x[i,j]<= Supply[i]*y[i]; # supply constraints
s.t. demand{j in D}: sum{i in S} x[i,j]>= Demand[j]; # demand constraints
```

```
Overwriting newshanghai.mod
```

```
#Q4 Endurance ROund
%%writefile newshanghai.dat
```

```
param C: SH M:=
AUC 1.18 0.91
BIR 1.6 1.52
FRA 1.65 1.73
MUM 1.21 1.38
SIN 1.44 1.43;

param Supply:=
AUC 15000
BIR 15000
FRA 20000
MUM 25000
SIN 20000;

param Demand:=
SH 90000
M 6000;

param FC:=
AUC 917000
BIR 962000
FRA 1093000
MUM 959000
SIN 1058000;
```

```
Overwriting newshanghai.dat
```

```
%%ampl_eval
reset;
# Model File
model newshanghai.mod;
data newshanghai.dat;
#Calling Optimization Engine and Optimizing
option solver cbc;
solve;
#Display Results
display Z, x, y;
```

```
cbc 2.10.10: infeasible problem
0 simplex iterations
absmipgap=1.79769e+308, relmipgap=1
Z = 5123130

x :=
AUC M     5000
AUC SH   10000
BIR M        0
BIR SH   15000
FRA M     1000
FRA SH   20000
```

```
MUM M        0
MUM SH   25000
SIN M        0
SIN SH   20000
;

y [*] :=
AUC  1
BIR  1
FRA  1
MUM  1
SIN  1
;
```

```
#Q4 Capacity Expansion Shanghai20/70/90 Calculator
%%writefile shanghai20cap.mod
set S = {"KC", "SA"}; #supply nodes
set D = {"T", "SH", "MC", "M", "L", "CA", "AT"}; #demand nodes

#Variables
var x{S,D} >=0 integer; # transportation quantity from nodes set S to demand set D
var y{S} binary; #fixed cost of expanding

#Parameters
param C{S, D}; #cost matrix for transportation
param Supply{S}; #Supply capacity of the supply nodes
param Demand{D}; #demand at nodes in set D
param FC{S}; #one time fixed cost at one plant or another in set S


minimize Z: sum{i in S} y[i]*FC[i]+ sum {i in S, j in D} C[i,j]*x[i,j]; #total Transporta

#Constraints
s.t. supply{i in S}: sum{j in D} x[i,j]<= 60000+50000*y[i]; # supply constraints
s.t. demand{j in D}: sum{i in S} x[i,j]>= Demand[j]; # demand constraints
s.t. KCorSA: y["KC"] + y["SA"]>=1; #expanding one plant or the other
```

```
Overwriting shanghai20cap.mod
```

```
%%writefile shanghai20cap.dat

param C: T SH MC M L CA AT=
KC 1.79 2.13 1.76 2.34 1.86 1.90 1.82
SA 2.13 2.03 1.58 1.80 2.14 1.26 1.76;

param Supply:=
KC 60000
SA 60000;
```

```
param Demand:=
T 5000
SH 90000
MC 4000
M 6000
L 40000
CA 10000
AT 60000;

param FC:=
KC 2590000
SA 2061000;
```

```
Overwriting shanghai20cap.dat
```

```
#Q4 Shanghai 20
%%ampl_eval
reset;
# Model File
model shanghai20cap.mod;
data shanghai20cap.dat;
#Calling Optimization Engine and Optimizing
option solver cbc;
solve;
#Display Results
display Z, x, y;
```

```
cbc 2.10.10: optimal solution; objective 5055970
1 simplex iterations
1 barrier iterations
Z = 5055970

x :=
KC AT   60000
KC CA       0
KC L    40000
KC M        0
KC MC       0
KC SH       0
KC T     5000
SA AT       0
SA CA   10000
SA L        0
SA M     6000
SA MC    4000
SA SH   90000
SA T        0
;

y [*] :=
KC  1
SA  1
;
```