Question 2

Superior Roasters Coffee Company mixes specialty blends of coffee to sell to Big Cup Coffee, a small chain of coffee shops. The beans used in their specialty coffee are listed in the table below:

Bean Type	Cost / Pound ($)	Available amount (lbs)
Columbian	1.00	550
Brazilian	0.85	450
Sumatran	1.55	650

Superior Roaster's products are:

- Robust Joe must consist of 60-75% Sumatran beans and at least 10% Columbian beans. Each pound of Robust Joe can be sold to Big Cup for $4.25.
- Light Joe must consist of 50-60% Brazilian beans and no more than 20% Sumatran beans. Each pound of Light Joe can be sold to Big Cup for $3.95.

Formulate an LP to maximize the total profit (revenue - cost).

In [ ]:
```
# symbol vectors for types of beans and the two blends
types = [:Columbian, :Brazilian, :Sumatran];
blends = [:Robust, :Light];
```

In [ ]:
```
using JuMP
using HiGHS

# create model with appropriate optimizer
coffeeblend = Model(HiGHS.Optimizer);
```

In [ ]:
```
# cost per pound of each kind of bean (in $)
coffee_price = Dict(:Columbian => 1.00, :Brazilian => 0.85, :Sumatran => 1.55

# amount of each kind of bean available (in lbs)
bean_availability = Dict(:Columbian => 550, :Brazilian => 450, :Sumatran => 6

# money made from each type of blend sold (in $)
purchase_price = Dict(:Robust => 4.25, :Light => 3.95);
```

In [ ]:
```
# declaring variable that is indexed over types and blends
@variable(coffeeblend, x[i in types, j in blends] >= 0)
```

Out[ ]:
```
2-dimensional DenseAxisArray{VariableRef,2,...} with index sets:
    Dimension 1, [:Columbian, :Brazilian, :Sumatran]
    Dimension 2, [:Robust, :Light]
And data, a 3×2 Matrix{VariableRef}:
 x[Columbian,Robust]  x[Columbian,Light]
 x[Brazilian,Robust]  x[Brazilian,Light]
 x[Sumatran,Robust]   x[Sumatran,Light]
```

In [ ]:
```
# constraints for Robust Joe
@constraint(coffeeblend, RSLowerBound, x[:Sumatran, :Robust] >= 0.6*sum(x[i,:
@constraint(coffeeblend, RSUpperBound, x[:Sumatran, :Robust] <= 0.75*sum(x[i,
@constraint(coffeeblend, RCBound, x[:Columbian, :Robust] >= 0.1*sum(x[i,:Robu

# constraints for Light Joe
@constraint(coffeeblend, LBLowerBound, x[:Brazilian, :Light] >= 0.5*sum(x[i,:
@constraint(coffeeblend, LBUpperBound, x[:Brazilian, :Light] <= 0.6*sum(x[i,:
@constraint(coffeeblend, LSBound, x[:Sumatran, :Light] <= 0.2*sum(x[i,:Light]

# constraints for bean amount
@constraint(coffeeblend, SumBound, sum(x[:Sumatran, i] for i in blends) <= 65
@constraint(coffeeblend, SumBound2, sum(x[:Columbian, i] for i in blends) <=
@constraint(coffeeblend, SumBound3, sum(x[:Brazilian, i] for i in blends) <=
```

Out[ ]: SumBound3 : $x_{Brazilian,Robust} + x_{Brazilian,Light} \leq 450.0$

In [ ]:
```
# defining the objective function for the model

cost = 1.00*sum(x[:Columbian, i] for i in blends) + 0.85*sum(x[:Brazilian, i]
revenue = 4.25*sum(x[i, :Robust] for i in types) + 3.95*sum(x[i, :Light] for

@objective(coffeeblend, Max, revenue - cost);
```

In [ ]:
```
# solving the model
optimize!(coffeeblend);

# outputs detailed information about the solution process
@show solution_summary(coffeeblend);
```

```
Presolving model
9 rows, 6 cols, 24 nonzeros
9 rows, 6 cols, 24 nonzeros
Presolve : Reductions: rows 9(-0); columns 6(-0); elements 24(-0)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0    -4.4549962997e+01 Ph1: 8(10.175); Du: 6(44.55) 0s
          8    -4.9025000000e+03 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 8
Objective value     :  4.9025000000e+03
HiGHS run time      :          0.00
solution_summary(coffeeblend) = * Solver : HiGHS

* Status
  Termination status : OPTIMAL
  Primal status      : FEASIBLE_POINT
  Dual status        : FEASIBLE_POINT
  Message from the solver:
  "kHighsModelStatusOptimal"

* Candidate solution
  Objective value      : 4.90250e+03
  Objective bound      : 4.90250e+03
  Relative gap         : Inf
  Dual objective value : 4.90250e+03

* Work counters
  Solve time (sec)   : 3.87457e-03
  Simplex iterations : 8
  Barrier iterations : 0
  Node count         : -1
```