```
In [2]: using DataFrames
```

```
In [49]: quinta = readtable("SelectingHotels.csv")
```

Out[49]:

	Hotel	Location	Price	Price_normalized_	Square_Root_of_Median_Income_normaliz
1	1	Eureka, California	2925000	-0.3	-0.81
2	2	Fresno, California	10000000	1.7	-0.41
3	3	Fresno, California	3750000	-0.07	-0.41
4	4	Fresno, California	3500000	-0.14	-0.41
5	5	Fresno, California	325000	-1.04	-0.41
6	6	Long Beach, California	8950000	1.4	0.66
7	7	Los Angeles, California	1950000	-0.58	0.17
8	8	Los Angeles, California	1750000	-0.63	0.17
9	9	Los Angeles, California	4900000	0.26	0.17
10	10	South Lake Tahoe, California	1650000	-0.66	-0.79
11	11	South Lake Tahoe, California	1125000	-0.81	-0.79
12	12	South Lake Tahoe, California	2500000	-0.42	-0.79
		South			

13	13	Lake Tahoe, California	1975000	-0.57	-0.79
14	14	South Lake Tahoe, California	3750000	-0.07	-0.79
15	15	South Lake Tahoe, California	1475000	-0.71	-0.79
16	16	South Lake Tahoe, California	750000	-0.92	-0.79

In [50]:
```
39.05 - 5.41*quinta[1,7] + 5.86*quinta[1,4] - 3.09*quinta[1,5] + 1.75*quinta[1,6]
```

Out[50]:
```
44.25989999999999
```

In [51]:
```
profit = zeros(16)
for i = 1:16
    profit[i] = 39.05 - 5.41*quinta[i,7] + 5.86*quinta[i,4] - 3.09*quinta[i,5] + 1.75*quinta[i,6]
end
```

In [52]:
```
profit
```

Out[52]:
```
16-element Array{Float64,1}:
 44.2599
 53.3641
 42.9919
 42.5817
 37.3077
 49.0842
 23.7433
 23.4503
 28.6657
 38.9173
 38.0383
 40.3237
 39.4447
 42.3747
 38.6243
 37.3937
```

In [53]:
```
quinta[2,3]
```

Out[53]:
```
10000000
```

In [54]:
```
using JuMP, Gurobi
```

In [55]:
```
m = Model(solver=GurobiSolver())
```

Out[55]:

$$\min \quad 0$$

Subject to

In [56]:
```
n = 1:16
```

Out[56]:
```
1:16
```

In [57]:
```
@variable(m,x[n],Bin)
```

Out[57]:

$$x_i \qquad \{0,1\} \qquad i \qquad \{1,2,\dots,15,16\}$$

In [58]:
```
@constraint(m,budget,sum((x[i]*quinta[i,3]) for i in n) <= 10000000)
```

Out[58]:

$$2.925e6x_1 + 1.0e7x_2 + 3.75e6x_3 + 3.5e6x_4 + 325000x_5 + 8.95e6x_6 + 1.95e6x_7 + 1.75e$$
$$+ 2.5e6x_{12} + 1.975e6x_{13} + 3.75e6x_{14} + 1.475e6x_{15} + 7500$$

In [59]:
```
@objective(m,Max,sum(x[i]*profit[i] for i in n))
```

Out[59]:

$$44.25989999999999x_1 + 53.36409999999999x_2 + 42.99189999999999x_3 + 42.5816999$$
$$+ 23.743299999999994x_7 + 23.450299999999995x_8 + 28.665699999999998x_9 + 38.9$$
$$+ 42.3747x_{14} + 38.6243x_{15} + 37.3937x$$

In [60]:
```
solve(m)
```

Out[60]:
```
:Optimal

Optimize a model with 1 rows, 16 columns and 16 nonzeros
Variable types: 0 continuous, 16 integer (16 binary)
Coefficient statistics:
  Matrix range     [3e+05, 1e+07]
  Objective range  [2e+01, 5e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+07, 1e+07]
Found heuristic solution: objective 185.696
Presolve removed 0 rows and 2 columns
Presolve time: 0.03s
Presolved: 1 rows, 14 columns, 14 nonzeros
Variable types: 0 continuous, 14 integer (14 binary)

Root relaxation: objective 2.730760e+02, 1 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Ti
me

     0     0  273.07602    0    1  185.69650  273.07602  47.1%     -    0s
H    0     0                     270.0497000  273.07602  1.12%     -    0s

Explored 0 nodes (1 simplex iterations) in 0.16 seconds
Thread count was 4 (of 4 available processors)
```

```
Solution count 2: 270.05 185.696
Pool objective bound 270.05

Optimal solution found (tolerance 1.00e-04)
Best objective 2.700497000000e+02, best bound 2.700497000000e+02, gap 0.00
00%
```

In [61]:

```
getobjectivevalue(m)
```

Out[61]:

```
270.04970000000003
```

In [62]:

```
getvalue(x)
```

Out[62]:

```
x: 1 dimensions:
[ 1] = 0.0
[ 2] = 0.0
[ 3] = -0.0
[ 4] = -0.0
[ 5] = 1.0
[ 6] = 0.0
[ 7] = -0.0
[ 8] = -0.0
[ 9] = -0.0
[10] = 1.0
[11] = 1.0
[12] = 1.0
[13] = 1.0
[14] = -0.0
[15] = 1.0
[16] = 1.0
```

In [63]:

```
@constraint(m,x[2]+x[3]+x[4]+x[5]<=2)
```

Out[63]: $x_2 + x_3 + x_4 + x_5 \leq 2$

In [64]:

```
@constraint(m,x[6]+x[7]+x[8]+x[9]<=2)
```

Out[64]: $x_6 + x_7 + x_8 + x_9 \leq 2$

In [65]:

```
@constraint(m,x[10]+x[11]+x[12]+x[13]+x[14]+x[15]+x[16]<=2)
```

Out[65]: $x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} \leq 2$

In [66]:

```
solve(m)
```

Out[66]:

```
:Optimal

Optimize a model with 4 rows, 16 columns and 31 nonzeros
Variable types: 0 continuous, 16 integer (16 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+07]
  Objective range  [2e+01, 5e+01]
  Bounds range     [1e+00, 1e+00]
  RHS range        [2e+00, 1e+07]
Found heuristic solution: objective 185.696
Presolve removed 0 rows and 2 columns
Presolve time: 0.00s
Presolved: 4 rows, 14 columns, 27 nonzeros

MIP start did not produce a new incumbent solution
MIP start violates constraint R3 by 4.000000000

Variable types: 0 continuous, 14 integer (14 binary)

Root relaxation: objective 2.184885e+02, 5 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Ti
me

     0     0  218.48848    0    1  185.69650  218.48848  17.7%     -    0s
H    0     0                       204.1932000  218.48848  7.00%     -    0s
     0     0  212.15514    0    3  204.19320  212.15514  3.90%     -    0s
     0     0  206.17723    0    2  204.19320  206.17723  0.97%     -    0s
H    0     0                       205.7168000  206.17723  0.22%     -    0s
     0     0  206.16306    0    2  205.71680  206.16306  0.22%     -    0s

Cutting planes:
  Gomory: 1
  Cover: 1
  StrongCG: 1

Explored 0 nodes (15 simplex iterations) in 0.03 seconds
Thread count was 4 (of 4 available processors)

Solution count 3: 205.717 204.193 185.696
Pool objective bound 205.717

Optimal solution found (tolerance 1.00e-04)
Best objective 2.057168000000e+02, best bound 2.057168000000e+02, gap 0.00
00%
```

In [67]:

```
getobjectivevalue(m)
```

Out[67]:

```
205.71679999999998
```

In [68]:

```
getvalue(x)
```

Out[68]:

```
x: 1 dimensions:
[ 1] = 1.0
[ 2] = 0.0
[ 3] = -0.0
[ 4] = 0.0
[ 5] = 1.0
[ 6] = 0.0
[ 7] = 1.0
[ 8] = 1.0
[ 9] = -0.0
[10] = 1.0
[11] = 1.0
[12] = -0.0
[13] = 0.0
[14] = -0.0
```

```
[15] = 0.0
[16] = 0.0
```

```
In [ ]:
```