The first part of the notebook is from Friday unchanged. Scroll down about half way to where it says **Sensitivity Analysis** for what we did the following Monday.

Example 3.5. A large construction firm is currently excavating at three sites. Meanwhile, they are also building at four additional sites, where they require fill dirt. The excavations at sites 1, 2, and 3 produce 150, 400, and 325 cubic yards of dirt per day. The building sites A, B, C, and D require 175, 125, 225, and 450 cubic yards of dirt per day. Additional fill dirt can also be obtained from site 4 at a cost of 5 dollars per cubic yard. The cost of shipping fill dirt is about 20 dollars per mile for one truckload, and a truck carries 10 cubic yards of dirt. Table 3.3 gives the distance between sites in miles. Find the transportation plan that minimizes the cost to the company.

Table 3.3

site	A	B	C	D
1	5	2	6	10
2	4	5	7	5
3	7	6	4	4
4	9	10	6	2

In [1]:
```
D=[5 2 6 10; 4 5 7 5; 7 6 4 4; 9 10 6 2]
```

Out[1]:
```
4×4 Matrix{Int64}:
 5   2  6  10
 4   5  7   5
 7   6  4   4
 9  10  6   2
```

In [2]:
```
C=2*D
C[4,:].+=5
C
```

Out[2]:
```
4×4 Matrix{Int64}:
 10   4  12  20
  8  10  14  10
 14  12   8   8
 23  25  17   9
```

In [3]:
```
using JuMP, HiGHS
```

In [4]:
```
model=Model(HiGHS.Optimizer)
```

Out[4]:
```
A JuMP Model
├ solver: HiGHS
├ objective_sense: FEASIBILITY_SENSE
├ num_variables: 0
├ num_constraints: 0
└ Names registered in the model: none
```

In [5]:
```julia
@variable(model,x[1:4,1:4].>=0)
```

Out[5]:
```
4×4 Matrix{VariableRef}:
 x[1,1]  x[1,2]  x[1,3]  x[1,4]
 x[2,1]  x[2,2]  x[2,3]  x[2,4]
 x[3,1]  x[3,2]  x[3,3]  x[3,4]
 x[4,1]  x[4,2]  x[4,3]  x[4,4]
```

In [6]:
```julia
cost(x)=sum(C.*x)
cost(x)
```

Out[6]: $10x_{1,1} + 8x_{2,1} + 14x_{3,1} + 23x_{4,1} + 4x_{1,2} + 10x_{2,2} + 12x_{3,2} + 25x_{4,2} + 12x_{1,3} + 14x_{2,3} + 8x_{3,3} + 17x_{4,}$

In [7]:
```julia
# Fix it so it doesn't scroll off the edge
println(cost(x))
```

```
10 x[1,1] + 8 x[2,1] + 14 x[3,1] + 23 x[4,1] + 4 x[1,2] + 10 x[2,2] + 12 x
[3,2] + 25 x[4,2] + 12 x[1,3] + 14 x[2,3] + 8 x[3,3] + 17 x[4,3] + 20 x[1,4]
+ 10 x[2,4] + 8 x[3,4] + 9 x[4,4]
```

In [8]:
```julia
@objective(model,Min,cost(x))
```

Out[8]: $10x_{1,1} + 8x_{2,1} + 14x_{3,1} + 23x_{4,1} + 4x_{1,2} + 10x_{2,2} + 12x_{3,2} + 25x_{4,2} + 12x_{1,3} + 14x_{2,3} + 8x_{3,3} + 17x_{4,}$

In [9]:
```julia
# Amount of dirt received by site j
r=[sum(x[:,j]) for j=1:4]
```

Out[9]:
```
4-element Vector{AffExpr}:
 x[1,1] + x[2,1] + x[3,1] + x[4,1]
 x[1,2] + x[2,2] + x[3,2] + x[4,2]
 x[1,3] + x[2,3] + x[3,3] + x[4,3]
 x[1,4] + x[2,4] + x[3,4] + x[4,4]
```

In [10]:
```julia
c1=@constraint(model,r.>=[175,125,225,450])
```

Out[10]:
```
4-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{Math
OptInterface.ScalarAffineFunction{Float64}, MathOptInterface.GreaterThan{Fl
oat64}}, ScalarShape}}:
 x[1,1] + x[2,1] + x[3,1] + x[4,1] ≥ 175
 x[1,2] + x[2,2] + x[3,2] + x[4,2] ≥ 125
 x[1,3] + x[2,3] + x[3,3] + x[4,3] ≥ 225
 x[1,4] + x[2,4] + x[3,4] + x[4,4] ≥ 450
```

In [11]:
```julia
# Amount of dirt produced by site i
s=[sum(x[i,:]) for i=1:4]
```

Out[11]:
```
4-element Vector{AffExpr}:
 x[1,1] + x[1,2] + x[1,3] + x[1,4]
 x[2,1] + x[2,2] + x[2,3] + x[2,4]
 x[3,1] + x[3,2] + x[3,3] + x[3,4]
 x[4,1] + x[4,2] + x[4,3] + x[4,4]
```

In [12]:
```
c2=@constraint(model,s[1:3].<=[150,400,325])
```

Out[12]:
```
3-element Vector{ConstraintRef{Model, MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64}, MathOptInterface.LessThan{Float64}}, ScalarShape}}:
 x[1,1] + x[1,2] + x[1,3] + x[1,4] ≤ 150
 x[2,1] + x[2,2] + x[2,3] + x[2,4] ≤ 400
 x[3,1] + x[3,2] + x[3,3] + x[3,4] ≤ 325
```

In [13]:
```
print(model)
```

$$\begin{aligned}
\min \quad & 10x_{1,1} + 8x_{2,1} + 14x_{3,1} + 23x_{4,1} + 4x_{1,2} + 10x_{2,2} + 12x_{3,2} + 25x_{4,2} + 12x_{1,3} + 14x_{2,3} + 8 \\
\text{Subject to} \quad & x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \geq 175 \\
& x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} \geq 125 \\
& x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \geq 225 \\
& x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} \geq 450 \\
& x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} \leq 150 \\
& x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} \leq 400 \\
& x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} \leq 325 \\
& x_{1,1} \geq 0 \\
& x_{2,1} \geq 0 \\
& x_{3,1} \geq 0 \\
& x_{4,1} \geq 0 \\
& x_{1,2} \geq 0 \\
& x_{2,2} \geq 0 \\
& x_{3,2} \geq 0 \\
& x_{4,2} \geq 0 \\
& x_{1,3} \geq 0 \\
& x_{2,3} \geq 0 \\
& x_{3,3} \geq 0 \\
& x_{4,3} \geq 0 \\
& x_{1,4} \geq 0 \\
& x_{2,4} \geq 0 \\
& x_{3,4} \geq 0 \\
& x_{4,4} \geq 0
\end{aligned}$$

In [14]:
```
# Fix the scrolling left to right
print(sprint(print,model))
```

```
Min 10 x[1,1] + 8 x[2,1] + 14 x[3,1] + 23 x[4,1] + 4 x[1,2] + 10 x[2,2] + 12
x[3,2] + 25 x[4,2] + 12 x[1,3] + 14 x[2,3] + 8 x[3,3] + 17 x[4,3] + 20 x[1,
4] + 10 x[2,4] + 8 x[3,4] + 9 x[4,4]
Subject to
 x[1,1] + x[2,1] + x[3,1] + x[4,1] ≥ 175
 x[1,2] + x[2,2] + x[3,2] + x[4,2] ≥ 125
 x[1,3] + x[2,3] + x[3,3] + x[4,3] ≥ 225
 x[1,4] + x[2,4] + x[3,4] + x[4,4] ≥ 450
 x[1,1] + x[1,2] + x[1,3] + x[1,4] ≤ 150
 x[2,1] + x[2,2] + x[2,3] + x[2,4] ≤ 400
 x[3,1] + x[3,2] + x[3,3] + x[3,4] ≤ 325
 x[1,1] ≥ 0
 x[2,1] ≥ 0
 x[3,1] ≥ 0
 x[4,1] ≥ 0
 x[1,2] ≥ 0
 x[2,2] ≥ 0
 x[3,2] ≥ 0
 x[4,2] ≥ 0
 x[1,3] ≥ 0
 x[2,3] ≥ 0
 x[3,3] ≥ 0
 x[4,3] ≥ 0
 x[1,4] ≥ 0
 x[2,4] ≥ 0
 x[3,4] ≥ 0
 x[4,4] ≥ 0
```

In [15]:

```
optimize!(model)
```

```
Running HiGHS 1.13.1 (git hash: 1d267d97c): Copyright (c) 2026 under Apache
2.0 license terms
Using BLAS: blastrampoline
LP has 7 rows; 16 cols; 28 nonzeros
Coefficient ranges:
  Matrix  [1e+00, 1e+00]
  Cost    [4e+00, 2e+01]
  Bound   [0e+00, 0e+00]
  RHS     [1e+02, 4e+02]
Presolving model
7 rows, 16 cols, 28 nonzeros  0s
7 rows, 14 cols, 24 nonzeros  0s
Presolve reductions: rows 7(-0); columns 14(-2); nonzeros 24(-4)
Solving the presolved LP
Using dual simplex solver
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 4(975) 0.0s
          5     7.6500000000e+03 Pr: 0(0) 0.0s

Performed postsolve
Solving the original LP from the solution after postsolve

Model status        : Optimal
Simplex   iterations: 5
Objective value     :  7.6500000000e+03
P-D objective error :  0.0000000000e+00
HiGHS run time      :          0.00
```

In [16]:

```
objective_value(model)
```

Out[16]:

```
7650.0
```

Sensitivity Analysis. We check how sensitive the optimum is with respect to the constraints.

The following has has been simplified from what we did interactively to make it easier to follow when reading the notebook.

In [17]:

```
# All these zero mean the optimum lies on all constraints related to how
# dirt is needed at sites A, B, C and D.  No more dirt than necessary was
# supplied to any of the receiving sites.
value(r)-[175,125,225,450]
```

Out[17]:

```
4-element Vector{Float64}:
 0.0
 0.0
 0.0
 0.0
```

In [18]:

```
# The first two are positive means that not all the dirt available at
# sites 1 and 2 was used.  Thus, the amount of dirt available at the
# first two sites did not affect the optimum.
[150,400,325]-value(s[1:3])
```

Out[18]:
```
3-element Vector{Float64}:
  25.0
 225.0
   0.0
```

As shown below, the shadow prices for sites 1 and 2 are zero, since there was excess dirt left at those sites. The shadow price for site 3 is −1 dollar, meaning that the total cost would increase by −1 dollar if this constraint were increased by 1 cubic yard of dirt.

In [19]:
```
# The dual of the constraint is called the shadow price.
display(c2[1])
dual(c2[1])
```

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} \leq 150$$

Out[19]:
```
0.0
```

In [20]:
```
display(c2[2])
dual(c2[2])
```

$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} \leq 400$$

Out[20]:
```
0.0
```

In [21]:
```
# Means that increasing the third element of the second
# constraint group increases the price by -1.  That is,
# decreases it by 1.
display(c2[3])
dual(c2[3])
```

$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} \leq 325$$

Out[21]:
```
-1.0
```

In [22]:
```
# Simultaneously find the duals of all the c1 constraints
dual.(c1)
```

Out[22]:
```
4-element Vector{Float64}:
 8.0
 4.0
 9.0
 9.0
```

As shown above, the duals of each of c1 constraints are non-zero, so the optimum lies on the boundary of those contraints. For example, if the contraint given by c1[3] is is increased by 1, then the cost increases by 9.

Illustrate the shadow prices by changing the constraints and observing the change in price. Note that if we change the constraint by too much, then other contraints may become important at which the shadow price might change.

In [23]:
```
# Change the constraint with shadow price -1
# by deleting it and adding a constraint that is 1 greater.
delete(model,c2[3])
c2[3]=@constraint(model,s[3]<=326)
```

Out[23]:

$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} \leq 326$$

In [24]:
```
optimize!(model)
objective_value(model)
```

```
LP has 7 rows; 16 cols; 28 nonzeros
Coefficient ranges:
  Matrix  [1e+00, 1e+00]
  Cost    [4e+00, 2e+01]
  Bound   [0e+00, 0e+00]
  RHS     [1e+02, 4e+02]
Solving LP with useful basis so presolve not used
Using dual simplex solver
  Iteration        Objective     Infeasibilities num(sum)
          0    -1.0000001711e+00 Ph1: 2(2); Du: 1(1) 0.0s
          2     7.6490000000e+03 Pr: 0(0) 0.0s

Model status        : Optimal
Simplex   iterations: 2
Objective value     :  7.6490000000e+03
P-D objective error :  0.0000000000e+00
HiGHS run time      :          0.00
```

Out[24]: 7649.0

Note that 7649 is one less that 7650. This is consistent with the shadow price since increasing by -1 is the same as decreasing by 1.

In [25]:
```
# Change the constraint with shadow price 9
# by deleting it and adding a constraint that is 1 greater.
delete(model,c1[3])
c1[3]=@constraint(model,r[3]>=226)
```

Out[25]:

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \geq 226$$

In [26]:
```
optimize!(model)
objective_value(model)
```

```
LP has 7 rows; 16 cols; 28 nonzeros
Coefficient ranges:
  Matrix  [1e+00, 1e+00]
  Cost    [4e+00, 2e+01]
  Bound   [0e+00, 0e+00]
  RHS     [1e+02, 4e+02]
Solving LP with useful basis so presolve not used
Using dual simplex solver
  Iteration        Objective     Infeasibilities num(sum)
          0    -1.9000008583e+01 Ph1: 4(8); Du: 3(19) 0.0s
          2     7.6580000000e+03 Pr: 0(0) 0.0s

Model status        : Optimal
Simplex   iterations: 2
Objective value     :  7.6580000000e+03
P-D objective error :  0.0000000000e+00
HiGHS run time      :          0.00
```

Out[26]:
```
7658.0
```

Note that 7658 is nine more than 7649. This is consistent with the shadow price being 9.

In both of the above cases changing the contraint by a small amount did not affect which constraints were active so the shadow price exactly predicted by how much the objective function changed.

In [ ]:
```

```