Homework 2
Team 7

Ali Abdullah, Orkhan Khankishiyev, & Tahsin Tabassum

DSA-5113-001 - 2025

1 Grapes Problem

1.1 a. How does Grimes determine an upper bound for grapes that can be used for the raisins product?

Following this mathematical reasoning, we can reach the same number:

$$\because \text{ Average Quality } A = 9, \quad \text{Average Quality } B = 5, \quad \text{Quality for Raisins} \geq 8$$

$$\because \text{ Total Grapes } G = 8,400,000$$

$$\therefore \text{ Amount of } A = 0.25G = 2,100,000, \quad B = 0.75G = 6,300,000$$

$$\therefore \frac{9A + 5B}{A + B} \geq 8$$

$$\therefore A \geq 3B$$

$$\because A \leq 2,100,000$$

$$\therefore 3B \leq 2,100,000$$

$$\therefore B \leq 700,000$$

$$\therefore \text{ Maximum amount of Raisins } R = A + B \leq 2,100,000 + 700,000 = 2,800,000$$

$$\therefore \mathbf{R_{\text{max}} = 2,800,000}$$

1.2 b. How does Bollman compute the fruit costs in Table 3??

Regardless of validity Bollman tries to incroperate the quality measure into the cost using equations 1 through 5

Let z = cost per pound of A grapes (in cents)

Let y = cost per pound of B grapes (in cents)

$$(2,100,000 \times z) + (6,300,000 \times y) = (8,400,000 \times 28)$$

$$\frac{z}{y} = \frac{9}{5}$$

$$\therefore z = 42 \text{ cents per pound}, \quad y = \frac{231}{3} \text{ cents per pound}$$

Therefore, we can find the fruit costs in the way Bollman found them:

As we established in (a), raisins have to at least contain 3 parts A and 1 part B. (meaning 0.75 of it is A, and 0.25 of it is B.

Similarly, the average quality of juice has to be 6.

$$\frac{9A + 5B}{A + B} \geq 6$$

$$9A + 5B \geq 6(A + B)$$

$$9A + 5B \geq 6A + 6B$$

$$9A - 6A \geq 6B - 5B$$

$$3A \geq B$$

So we know now that juice needs at least 3 parts of B, and 1 part of A.

Jelly can be made completely from B.

Using this fact, and the table of amount of grapes needed, we can find the fruit costs for each product:

$$\text{fruit cost of raisins} = (0.75 \cdot z + 0.25 \cdot y) \cdot 6.75 = 2.52$$

$$\text{fruit cost of juice} = (0.25 \cdot z + 0.75 \cdot y) \cdot 15.5 = 4.34$$

$$\text{fruit cost of jelly} = (y \cdot 19) = 4.43$$

1.3 c. Ignoring for the moment the chance to buy additional A-grade grapes. Formulate the production question as a LP problem, solve with AMPL and answer the following questions:

First, The way I built the model takes care of two important things in both both of Ms. Bollman and Thomas analyses. They both fall for the sunk-cost fallacy, subtracting the fruit cost from potential profits. Ms. Bollman does that with extra steps, that, while mathematical, do not make any sense for our objective.

In making a decision about maximizing profits, the costs of the past should not matter, because we cannot change them. We should only account for variable costs that we have to take, and restrictions on quality, demand, and supply that we have to follow.

Following this reasoning, I end up in my data file using the variable costs without the "fruit cost" which both Thomas and Bollman agree on.

Mathematical Formulation

$$\max Z = \sum_{i \in \text{PRODS}} \text{profit}_i \cdot X_i$$

where:

- X_i is the amount of product i produced.
- profit_i is the profit per unit of product i.

Subject to:

1. Demand Constraint

$$X_i \leq \text{demand}_i, \quad \forall i \in \text{PRODS}$$

2. Quality Constraint

$$\sum_{j \in \text{INGRIDS}} Y_{i,j} \cdot \text{quality}_j \geq \text{min\_quality}_i \cdot \sum_{j \in \text{INGRIDS}} Y_{i,j}, \quad \forall i \in \text{PRODS}$$

3. Ingredient Allocation Constraint

$$\sum_{j \in \text{INGRIDS}} Y_{i,j} = \text{usage}_i \cdot X_i, \quad \forall i \in \text{PRODS}$$

4. Supply Constraint

$$\sum_{i \in \text{PRODS}} Y_{i,j} \leq \text{supply}_j, \quad \forall j \in \text{INGRIDS}$$

5. Non-Negativity Constraint

$$X_i \geq 0, \quad Y_{i,j} \geq 0, \quad \forall i \in \text{PRODS}, \quad \forall j \in \text{INGRIDS}$$

And the Model in AMPL is as follows:

```
#AMPL model for "Question 2" for HomeWork 1

#Note: Deep seek was used by ali to debug and help with defining issues.

reset;
#OPTIONS ------------------------------------
option solver cplex;

#SETS ---------------------------------------


set PRODS;   # Products: Raisins, Juice, Jelly
set INGRIDS; # Ingredients: A, B (types of grapes)

#PARAMETERS ---------------------------------

param usage {PRODS};            # Usage of grapes in each product from tabel
param demand {PRODS};           # Demand limit for each product
param supply {INGRIDS};         # Available supply for A and B grapes
param profit {PRODS};           # Profit per unit of each product
param quality {INGRIDS};        # Quality points for A and B grapes
param min_quality {PRODS};      # Minimum required quality per product

#DECISION VARIABLES --------------------------
var X {i in PRODS} >= 0;  # Amount of each product to produce
var Y {i in PRODS, j in INGRIDS} >= 0; # Amount of each ingredient used in each product

#OBJECTIVE -----------------------------------

maximize Total_Profit:
    sum {i in PRODS} profit[i] * X[i];

#CONSTRAINTS ---------------------------------
```

```

# 1. Demand Constraint
subject to Demand_Limit {i in PRODS}:
    X[i] <= demand[i];

# 2. Quality Constraint
subject to Quality_Requirement {i in PRODS}:
    sum {j in INGRIDS} Y[i,j] * quality[j] >= min_quality[i] * sum {j in INGRIDS} Y[i,j];

# 3. Ingredient Allocation Constraint
subject to Ingredient_Balance {i in PRODS}:
    sum {j in INGRIDS} Y[i,j] = usage[i] * X[i];

# 4. Supply Limits for A and B grapes
subject to Supply_Limit {j in INGRIDS}:
    sum {i in PRODS} Y[i,j] <= supply[j];

#DATA ------------------------------------------
data HW_2_ali.dat;


#COMMANDS --------------------------------------
solve;
display X, Y, Total_Profit;
```

with this data file:

```
data;

set PRODS := Raisins Juice Jelly;
set INGRIDS := A B;

# I ignored both of Ms. Bollman and Thomas. They fall for the sunk-cost fallacy.
# We already bought the grapes, so we should not include their "cost" as a limitationn when deciding how to maximize profits.
# Maximizing profits is a question of things we can change, and restrictions we have to follow.
# The restriciton of "quality" is just there to produce different types of products.
# It is not going to add costs to sell some type or another.
# This is why I used Ms. Bollman's variable cost (without fruit cost).
# Note that  Ms. Bollman's cost is equal to Thomas's if we ignore fruit costs.

param profit :=
    Raisins 3.32
    Juice 6.38
    Jelly 7.75;

param demand :=
    Raisins Infinity
    Juice 205000
    Jelly 290000;

param supply :=
    A 2100000
    B 6300000;

param quality :=
```

```
    A 9
    B 5;

param min_quality :=
    Raisins 8
    Juice 6
    Jelly 5;

param usage :=
    Raisins 6.75
    Juice 15.5
    Jelly 19;
```

The output is as follows:

```
{\rtf1\ansi\ansicpg1252\cocoartf2821
\cocoatextscaling0\cocoaplatform0{\fonttbl\f0\fnil\fcharset0 Menlo-Regular;}
{\colortbl;\red255\green255\blue255;}
{\*\expandedcolortbl;;}
\margl1440\margr1440\vieww11520\viewh8400\viewkind0
\deftab720
\pard\pardeftab720\partightenfactor0

\f0\fs24 \cf0 ampl: model '/Users/ali_guraiffi/Library/CloudStorage/OneDrive-UniversityofOklahoma/Masters/Spring 2025/DSA-5113-001/AMPL/HW_2_ali.mod';\
CPLEX 22.1.1:               CPLEX 22.1.1: optimal solution; objective 3658119.594\
3 simplex iterations\
X [*] :=\
  Jelly  290000\
  Juice    8709.677419354839\
Raisins  408148.1481481481\
;\
\
Y :=\
Jelly   A         0\
Jelly   B   5510000\
Juice   A     33750\
Juice   B    101250\
Raisins A   2066250\
Raisins B    688750\
;\
\
Total_Profit = 3658119.593787336}
```

First, we make sure we approximate to make the solutions whole:

- **Jelly:** $290,000$ (already a whole number)
- **Juice:** $8,710$ (rounded up from 8709.6 to maintain feasibility)
- **Raisins:** $408,145$ (rounded down by 3 because approx one juice needs 3 times the grapes to make one raisins)

i. Production Quantities

Based on the optimized solution, the amount of each product to be produced is:

Jelly: $290,000$ units
Juice: $8,710$ units
Raisins: $408,145$ units

ii. Maximum Profit Contribution

The maximum profit contribution is approximated by the model to be 3,658,119.59. However, because we changed things a bit to make them whole, a closer approximation is 3,658,111.20

iii. Leftover Grapes

Because the AMPL model did not restrict to integer, it found that the remaining is 0. However, according to my approximations.

However, with my approximation, We can do the math applying what percent quality requires, and I find that 15 Grade A are left, and 2 Grade B are left.

iv. Average Quality

As we see in the output, the average quality is exactly at the boundry.

$$\textbf{Raisins:} \quad \frac{(2,066,250 \times 9) + (688,750 \times 5)}{2,755,000} = 8.00$$

$$\textbf{Juice:} \quad \frac{(33,750 \times 9) + (101,250 \times 5)}{135,000} = 6.00$$

$$\textbf{Jelly:} \quad \frac{(5,510,000 \times 5)}{5,510,000} = 5.00$$

Q1.d.

We got the following result after adding 300,000 lbs. of grade A grape and ran the simulation with sensitivity.

Code:

```
reset;
#OPTIONS ------------------------------------
option solver cplex;
option cplex_options 'sensitivity';


#SETS ---------------------------------------


set PRODS;   # Products: Raisins, Juice, Jelly
set INGRIDS; # Ingredients: A, B (types of grapes)

#PARAMETERS ---------------------------------

param usage {PRODS};           # Usage of grapes in each product from tabel
param demand {PRODS};          # Demand limit for each product
param supply {INGRIDS};        # Available supply for A and B grapes
param profit {PRODS};          # Profit per unit of each product
param quality {INGRIDS};       # Quality points for A and B grapes
param min_quality {PRODS};     # Minimum required quality per product

#DECISION VARIABLES ---------------------------
var X {i in PRODS} >= 0;  # Amount of each product to produce
var Y {i in PRODS, j in INGRIDS} >= 0; # Amount of each ingredient used in each
product

#OBJECTIVE -------------------------------------

maximize Total_Profit:
    sum {i in PRODS} profit[i] * X[i]-0.5*300000;

#CONSTRAINTS -----------------------------------

# 1. Demand Constraint
subject to Demand_Limit {i in PRODS}:
    X[i] <= demand[i];

# 2. Quality Constraint
subject to Quality_Requirement {i in PRODS}:
    sum {j in INGRIDS} Y[i,j] * quality[j] >= min_quality[i] * sum {j in INGRIDS}
Y[i,j];

# 3. Ingredient Allocation Constraint
subject to Ingredient_Balance {i in PRODS}:
    sum {j in INGRIDS} Y[i,j] = usage[i] * X[i];

# 4. Supply Limits for A and B grapes
subject to Supply_Limit {j in INGRIDS}:
```

```
    sum {i in PRODS} Y[i,j] <= supply[j];

#DATA ----------------------------------------
data 1d.dat;


#COMMANDS -------------------------------------
solve;
display X, Y, Total_Profit;
display Supply_Limit.dual;
```

Data file:

```
data;

set PRODS := Raisins Juice Jelly;
set INGRIDS := A B;

# I ignored both of Ms. Bollman and Thomas. They fall for the sunk-cost fallacy.
# We already bought the grapes, so we should not include their "cost" as a
limitationn when deciding how to maximize profits.
# Maximizing profits is a question of things we can change, and restrictions we have
to follow.
# The restriciton of "quality" is just there to produce different types of products.
# It is not going to add costs to sell some type or another.
# This is why I used Ms. Bollman's variable cost (without fruit cost).
# Note that  Ms. Bollman's cost is equal to Thomas's if we ignore fruit costs.

param profit :=
    Raisins 3.32
    Juice 6.38
    Jelly 7.75;

param demand :=
    Raisins Infinity
    Juice 205000
    Jelly 290000;

param supply :=
    A 2400000
    B 6300000;

param quality :=
    A 9
    B 5;

param min_quality :=
    Raisins 8
    Juice 6
    Jelly 5;

param usage :=
    Raisins 6.75
    Juice 15.5
    Jelly 19;
```

Output:

```
CPLEX 20.1.0.0: sensitivity
CPLEX 20.1.0.0: optimal solution; objective 3667346.979
4 dual simplex iterations (2 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
X [*] :=
  Jelly  289474
  Juice       0
Raisins  474074
;

Y :=
Jelly   A         0
Jelly   B   5500000
Juice   A         0
Juice   B         0
Raisins A   2400000
Raisins B     8e+05
;

Total_Profit = 3667350

Supply_Limit.dual [*] :=
A  0.519838
B  0.407895
;
```

I. Adding extra 300,000 lbs of A-Grade grapes at 0.5 cents per pound resulted in additional net profit of 3667350-3658120=$9230. It is not a big difference overall but theoretically speaking, the company should buy the additional A-grade grapes. In this case, they should only produce Jelly from B grade, and Raisins from A and B grade grapes.

II. Sensitivity analysis showed that the company can pay maximum 52 cents for additional A-grade grape and maximum 40.8 cents for additional B-grade grape.

Q.2.e

I.

When using Thomas's net profit numbers (Net Profit $0.61, -$0.02, -$0.14 for raisin, juice and jelly accordingly), we can say without running the simulation that producing juice and jelly is not profitable.

Simulation result showed following:

```
CPLEX 20.1.0.0: optimal solution; objective 253037.037
2 dual simplex iterations (2 in phase I)

suffix up OUT;
```

```
suffix down OUT;
suffix current OUT;
X [*] :=
  Jelly       0
  Juice       0
Raisins  414815
;

Y :=
Jelly   A         0
Jelly   B         0
Juice   A         0
Juice   B         0
Raisins A   2100000
Raisins B     7e+05
;

Total_Profit = 253037

Supply_Limit.dual [*] :=
A  0.120494
B  0
;
```

This confirms our conclusion. However, the total profit of $253037 is significantly less than what we got in part C ($3658120). According to sensitivity analysis, the company can pay 12.1 cents for grade A grape maximum and not buy any additional grade B grape.

II.

Then we used the Bollman's marginal profit numbers (Marginal profit $0.80, $2.04, $3.32 for raisin, juice and jelly accordingly), and the simulation gave following results. It suggests that the company produce only jelly (290000) and juice (186452) and no raisins. The total profit would be $1343161.29 and compared to part C, this number is still significantly less. According to simulation, the company can pay 13 cents maximum for grade A and grade B grapes.

```
CPLEX 20.1.0.0: sensitivity
CPLEX 20.1.0.0: optimal solution; objective 1343161.29
3 dual simplex iterations (2 in phase I)

suffix up OUT;
suffix down OUT;
suffix current OUT;
X [*] :=
  Jelly  290000
  Juice  186452
Raisins       0
;

Y :=
Jelly   A         0
Jelly   B   5510000
Juice   A   2100000
Juice   B    790000
```

```
Raisins A         0
Raisins B         0
;

Total_Profit = 1343160

Supply_Limit.dual [*] :=
A  0.131613
B  0.131613
;
```

III.

Thomas's calculations are incorrect because they include fixed overhead costs, which should not influence product-level decisions. Fixed costs remain unchanged regardless of the production mix, so they should not impact profitability comparisons. Additionally, both Thomas and Bollman consider fruit costs in their analysis. However, since the company has already purchased the grapes at a fixed price of 28 cents per pound, this cost is sunk and should not be included when evaluating profit per product. The correct approach is to focus only on variable costs that change with production, which is what our figures reflect. By subtracting only the relevant variable costs from the selling price, our method gives a more accurate measure of profitability, making it the correct basis for decision-making.

Problem 2

A)

Formulation of the production question as a LP problem:

Sets:

P = Set of all projects {A, B, C, D, E}

Parameters:

Y = Number of years to consider the projects for

S = Maximum amount to be invested

r_{ij} = Return rate each year $j \in Y$ from a project $i \in P$

a_{ij} = Availability of a project $i \in P$ each year $j \in Y$

l_i = Limit on the total investment into a project $i \in P$

b = Banking rate of return for money not reinvested in a year

Decision Variables:

x_{ij} = Amount invested in each project $i \in P$ each year $j \in Y$

c_{ij} = Cumulative amount invested in a project $i \in P$ up to a given year $j \in Y$

y_j = Amount of cash at the beginning of each year $j \in Y$

z_j = Amount of cash at the end of each year $j \in Y$

Objective Function:

maximize Total Return $z_j - S$

Constraints:

$$y_0 = S$$

$$\sum_{i \in P} x_{i,0} = S$$

$$\sum_{i \in P} x_{ij} \leq y_j\,, \qquad \forall j \in Y$$

$$y_j = \left(z_{j-1} * (1+b)\right) + \sum_{i \in P} c_{i,j-1} * r_{ij}\,, \qquad \forall j \in Y$$

$$z_j = y_j - \sum_{i \in P} x_{ij}\,, \qquad \forall j \in Y$$

$$c_{ij} = \sum_{j \in Y} x_{ij}\,, \qquad \forall i \in P$$

$$x_{ij} = 0\,, \qquad \forall i \in P, j \in Y, a_{ij} = 0$$

$$c_{ij} \leq l_i, \qquad \forall i \in P, j \in Y$$
$$x_{ij}, c_{ij} \geq 0, \qquad \forall\, i \in P, j \in Q$$
$$y_j, z_j \geq 0, \qquad \forall\, j \in Q$$

<u>Solving the model using AMPL:</u>

Code:

```
reset;
option solver cplex;

set P;                                          # Set of all projects

param Y;                                        # Number of years to consider the
projects for
param S >= 0;                           # Maximum amount to be invested
param r {0 .. Y, P};                    # Return rate in a given year from a project
param a {0 .. Y, P};                    # Availability of a project in a given year
param l {P};                            # Limit on the total investment in to a
project
param b;                                        # Banking rate of return for money not
re-invested in given year

var I {0 .. Y, P} >= 0;                 # Amount invested in each project in a given
year
var C {0 .. Y, P};                      # Cumulative amount invested in a project up
to a given year
var B {0 .. Y} >= 0;                    # Amount of cash at the beginning of each
year
var E {0 .. Y} >= 0;                    # Amount of cash at the end of each year

maximize Total_Return: (E[Y]-S);                                        #
Amount of Money Made Beyond the Initial Investment

subject to initial_investment: B[0] = S;                                # The
begining balance is the initial investment amount
subject to initial_spend: sum {p in P} I[0, p] = S;                     # All of the
intitial investment must be invested in year 0

subject to cash_flow {y in 0 .. Y}:
	sum {p in P} I[y, p] <= B[y];
	# Cannot spend more in a year than we had at the beginning of the year

subject to begin_balance {y in 1 .. Y}:                                 #
Beginning balance in each year after 0 is last years ending balance
	B[y] = (E[y-1]*(1+b)) + (sum {p in P} C[y-1, p]*r[y,p]);   # with the bank
rate applied plus the returns on the existing investments

subject to end_balance {y in 0 .. Y}:                                   # End
balance each year is the beginning balance minus the
	E[y] = B[y] - (sum {p in P} I[y, p]);                           #
amount invested that year

subject to cumulative_investment {y in 0 .. Y, p in P}:         # The amount
invested in a project up to that point is the sum of all the
```

```
      C[y, p] = sum {x in 0 .. y} I[x, p];                                  #
years past. However, each project can only be invested in in 1 year

subject to availability {y in 0 .. Y, p in P}:                              # If a
project is not available that year, no money can be invested in it
    if a[y, p] = 0 then
       I[y, p] = 0;

subject to project_limits {y in 0 .. Y, p in P}:                      # Limits on
the amount invested in each project
      C[y,p] <= l[p];

data 2a.dat
solve;

for {y in 0 .. Y, p in P} {
      # Clean way to print results
      if I[y, p] >0 then printf "Year %d Project %s: = $%d \n", y, p, I[y, p];
      }
printf "Total Return: $%d\n", Total_Return;

display cumulative_investment.dual;
```

Data file:

```
data;

set P := A B C D E;

param Y:= 3;
param S := 1000000;
param b:= 0.06;

param r: A B C D E:=
0      0            0            0            0            0
1      0.3    0            1.1    0            0
2      1            0.3    0            0            0
3      0            1            0            1.75   1.40;

param l :=
A 500000
B 500000
C Infinity
D Infinity
E 750000;

param a: A B C D E :=
0 1 0 1 1 0
1 0 1 0 0 0
2 0 0 0 0 1
3 0 0 0 1 1;
```

```
Solution: CPLEX 22.1.1: CPLEX 22.1.1: optimal solution; objective 797600
2 simplex iterations
Year 0 Project A: = $500000
Year 0 Project D: = $500000
Year 2 Project E: = $659000
Total Return: $797600 # Amount of Money Made Beyond the Initial Investment
```

B)

Dual variable values obtained from the model:

```
cumulative_investment.dual :=
0 A 0.4452
0 B 0
0 C 1.6324
0 D 0
0 E 0
1 A 1.4
1 B 0.42
1 C 0
1 D 0
1 E 0
2 A -0.0952
2 B 1
2 C 0
2 D 1.75
2 E 1.4
3 A 0
3 B 0
3 C 0
3 D 0
3 E 0
```

The shadow prices indicate the marginal value of additional investment in each project at different times. High values, such as $1.6324 for Project C in Year 0 and $1.75 for Project D in Year 2, suggest that investing more in these projects would significantly improve returns. A zero shadow price, like for Project B in Year 0, implies no additional benefit from further investment. Titan can use these values to adjust hurdle rates dynamically instead of relying on a fixed 10%, ensuring capital is allocated to the most profitable opportunities based on real-time investment potential.

C)

Change in portfolio without re-running the model:

If Projects D and E were initially selected due to their high payouts, a reduction in these payouts would likely diminish their appeal relative to other projects. Since Project E offers the highest payouts, it can be inferred that it was significantly more attractive than other projects, influencing the selection of other investments. A decrease in Project E's payout would likely enhance the relative appeal of other projects. Similarly, if Project D's payout drops to \$1.70, the overall return at the end would also decrease.

Change in portfolio after re-running the model:

Cumulative investment sensitivity was run because it is related to the shadow price. The code does not change while datafile changed slightly.

Data file for D change to \$1.7:

```
data;

set P := A B C D E;

param Y:= 3;
param S := 1000000;
param b:= 0.06;

param r: A B C D E:=
0      0            0            0            0            0
1      0.3           0           1.1         0            0
2      1            0.3         0            0            0
3      0            1            0            1.7        1.4;

param l :=
A 500000
B 500000
C Infinity
D Infinity
E 750000;

param a: A B C D E :=
0 1 0 1 1 0
1 0 1 0 0 0
2 0 0 0 0 1
3 0 0 0 1 1;
```

Solution:

```
CPLEX 20.1.0.0: optimal solution; objective 772600
2 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $500000
Year 2 Project E: = $659000
Total Return: $772600
cumulative_investment.dual :=
0 A    0.4452
0 B    0
0 C    1.6324
0 D    0
0 E    0
1 A    1.4
```

```
1 B    0.42
1 C    0
1 D    0
1 E    0
2 A   -0.1452
2 B    1
2 C    0
2 D    1.7
2 E    1.4
3 A    0
3 B    0
3 C    0
3 D    0
3 E    0
```

Data file for E change to $1.34 and E stays at $1.75:

```
data;

set P := A B C D E;

param Y:= 3;
param S := 1000000;
param b:= 0.06;

param r: A B C D E:=
0      0            0            0            0            0
1      0.3          0            1.1          0            0
2      1            0.3          0            0            0
3      0            1            0            1.75         1.34;

param l :=
A 500000
B 500000
C Infinity
D Infinity
E 750000;

param a: A B C D E :=
0 1 0 1 1 0
1 0 1 0 0 0
2 0 0 0 0 1
3 0 0 0 1 1;
```

Solution:

```
CPLEX 20.1.0.0: optimal solution; objective 758060
2 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $500000
Year 2 Project E: = $659000
Total Return: $758060
cumulative_investment.dual :=
0 A    0.42612
0 B    0
```

```
0 C    1.56244
0 D    0
0 E    0
1 A    1.34
1 B    0.402
1 C    0
1 D    0
1 E    0
2 A   -0.01612
2 B    1
2 C    0
2 D    1.75
2 E    1.34
3 A    0
3 B    0
3 C    0
3 D    0
3 E    0
```

After the changes to Projects D and E, the total return decreased, aligning with the conclusions drawn in the section without re-running the model. It is important to note that investments in Projects A, D, and E remain unchanged, while the total cost has decreased. Additionally, the reduced price of Project D does not significantly impact the total return compared to the changes in Project E's price. In other words, Project E has a greater influence on the payouts. This observation is consistent with the inferences made in the non-running model section. In conclusion, running sensitivity analysis is unnecessary for broad insights. However, if detailed information on the payouts is required, sensitivity analysis would provide more precise results.

D)

To answer this question, we added F and G to the portfolio. The code is the same as the previous ones. The data file is as follows:

```
data;

set P := A B C D E F G;

param Y:= 3;
param S := 1000000;
param b:= 0.06;

param r: A B C D E F G:=
0      0.0    0.0    0.0    0.0    0.0    0.0    0.0
1      0.3    0.0    1.1    0.0    0.0    0.8    1.1
2      1.0    0.3    0.0    0.0    0.0    0.45   0.0
3      0.0    1.0    0.0    1.75   1.4    0.0    0.15;

param l :=
```

```
A 500000
B 500000
C Infinity
D Infinity
E 750000
F Infinity
G Infinity;

param a: A B C D E F G:=
0 1 0 1 1 0 1 1
1 0 1 0 0 0 0 0
2 0 0 0 0 1 0 0
3 0 0 0 1 1 1 1;
```

When F is available the return is as follows:

```
CPLEX 20.1.0.0: optimal solution; objective 802311.2481
2 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $429892
Year 0 Project F: = $70107
Year 2 Project E: = $750000
Total Return: $802311
```

When comparing it to the base case return of $797600, there is only additional $4711 profit in return of investing $70107 in Project F. The return is less than 10%, so it is not recommended to spend time and resources investing in F.

When G is available the return is as follows:

```
CPLEX 20.1.0.0: optimal solution; objective 800128.6449
2 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $421955
Year 0 Project G: = $78044
Year 2 Project E: = $750000
Total Return: $800128
```

When comparing it to the base case return of $797600, there is only additional $2528 profit in return of investing $78044 in Project G. The return is less than 10%, so it is not recommended to spend time and resources investing in G.

E)

The code stayed the same and project J made unavailable in the datafile.

```
data;

set P := A B C D E F G;

param Y:= 3;
param S := 1000000;
```

```
param b:= 0.06;

param r: A B C D E F G:=
0      0.0    0.0    0.0    0.0    0.0    0.0    0.0
1      0.3    0.0    1.1    0.0    0.0    0.8    1.1
2      1.0    0.3    0.0    0.0    0.0    0.45   0.0
3      0.0    1.0    0.0    1.75   1.34   0.0    0.15;

param l :=
A 500000
B 500000
C Infinity
D Infinity
E 750000
F Infinity
G Infinity;

param a: A B C D E F G:=
0 1 0 1 1 0 1 0
1 0 1 0 0 0 0 0
2 0 0 0 0 1 0 0
3 0 0 0 1 1 1 0;
```

When Project E only pays $1.34, results of the simulation are:

```
CPLEX 20.1.0.0: optimal solution; objective 758060
2 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $500000
Year 2 Project E: = $659000
Total Return: $758060
```

We see that we get optimal profits only investing in projects A, D and E, with total return of $758060. This is less than our base case. Even though project F is available, simulations says not to invest in project B, C and F. Compared to Q3.c, the portfolio did not change considerably. So, this change does not make any sense.

When Project E retains original return and Project D pays $1.7 per dollar invested, results of the simulation are:

```
CPLEX 20.1.0.0: optimal solution; objective 782608.6957
3 dual simplex iterations (1 in phase I)
Year 0 Project A: = $500000
Year 0 Project D: = $202898
Year 0 Project F: = $297101
Year 1 Project B: = $387681
Year 2 Project E: = $750000
Total Return: $782608
```

Again, the total profit is less than our original case but this time, the simulation suggests to invest in Projects A, B, D, E and F. Compared to Q3.c, the portfolio changed considerably and profits increased. So, this change makes sense.