Christopher Allred
A02233404

MAE 3210 - Spring 2020 - Homework 4

Homework 4 is due online through Canvas in PDF format by 11:59PM on Monday March 16.

You are required to submit code for all functions and/or subroutines built to solve these problems, which is designed to be easy to read and understand, in your chosen programming language, and which you have written yourself. The text from your code should both be copied into a single PDF file submitted on canvas. Your submitted PDF must also include responses to any assigned questions, which for problems requiring programming should be based on output from your code. For example, if you are asked to find a numerical answer to a problem, the number itself should be included in your submission.

NOTE: For this homework you are welcome to solve problem 1 by hand, without using programming or submitting code. However, you are required to solve problems 3-4 with programming, and a copy of your code must be submitted. Use of MS excel (or equivalent software) is acceptable and encouraged for problem 2; please include a copy of your final spreadsheet within your submitted PDF.

1. Consider the optimization problem:

$$\text{Maximize } f(x, y) = -3x + y$$

subject to the constraints x

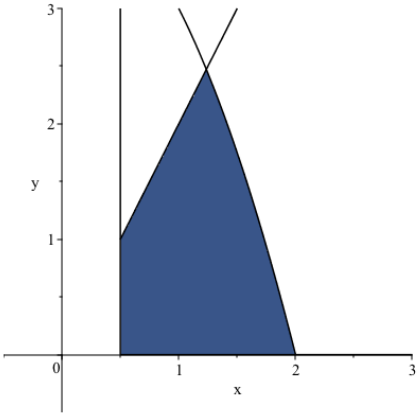
$$x^2 + y \leq 4,$$
$$-2x + y \leq 0,$$
$$x \geq 0.5,$$
$$y \geq 0.$$

(a) Plot the feasible solution space in the x − y plane.

```
with(plots):
cnsts := [x^2 + y ≤ 4,
-2 x + y ≤ 0,
x ≥ 0.5,
y ≥ 0.]
```

$$cnsts := \left[x^2 + y \le 4,\ -2\,x + y \le 0,\ 0.5 \le x,\ 0. \le y\right] \quad \textbf{(1)}$$

```
feasibleRegion := inequal(cnsts, x = -0.5 ..3, y = -0.5 ..3)
```

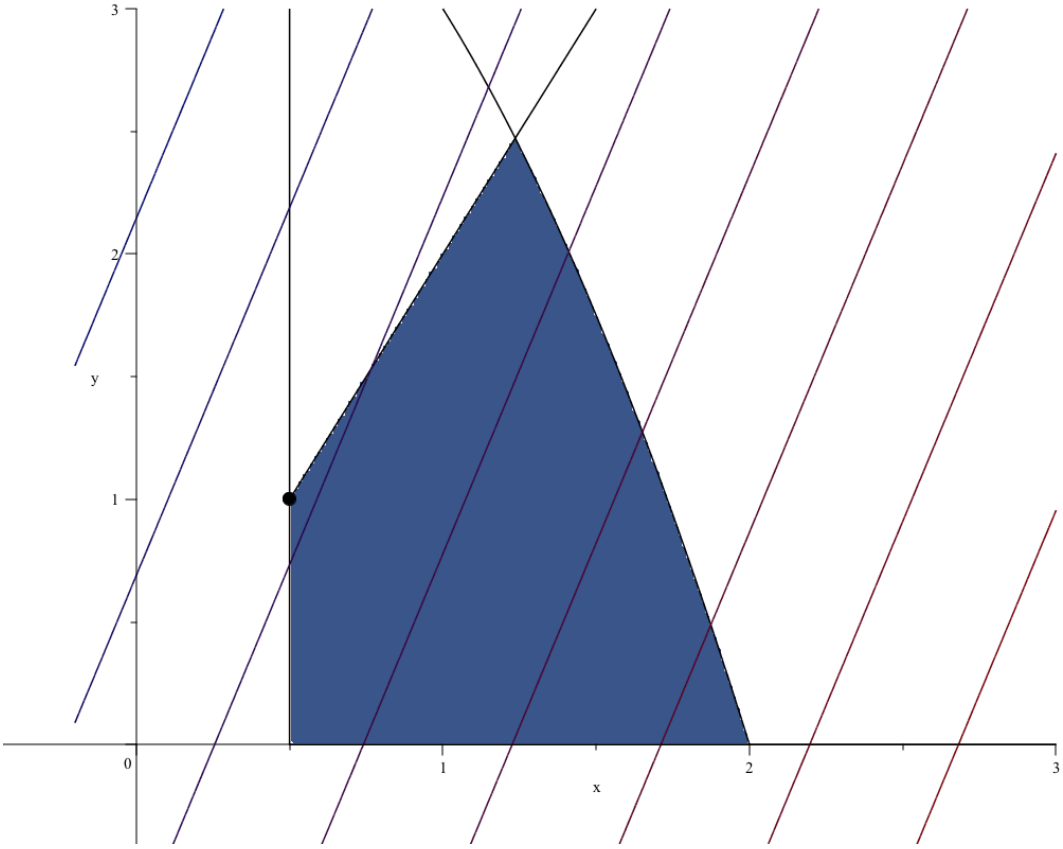


(b) Solve the optimization problem by using the graphical method.

```
obj := -3 x + y
#optimal pointdisplay(feasibleRegion,contours,optimalPoint)
```

$$obj := -3\,x + y \quad \textbf{(2)}$$

```
contours := contourplot(obj, x = -0.2 ..3, y = -0.5 ..3, ):
optimalPoint := pointplot({[0.5, 1]}, symbolsize = 10, symbol
    = solidcircle, ):
display(feasibleRegion, contours, optimalPoint)
```

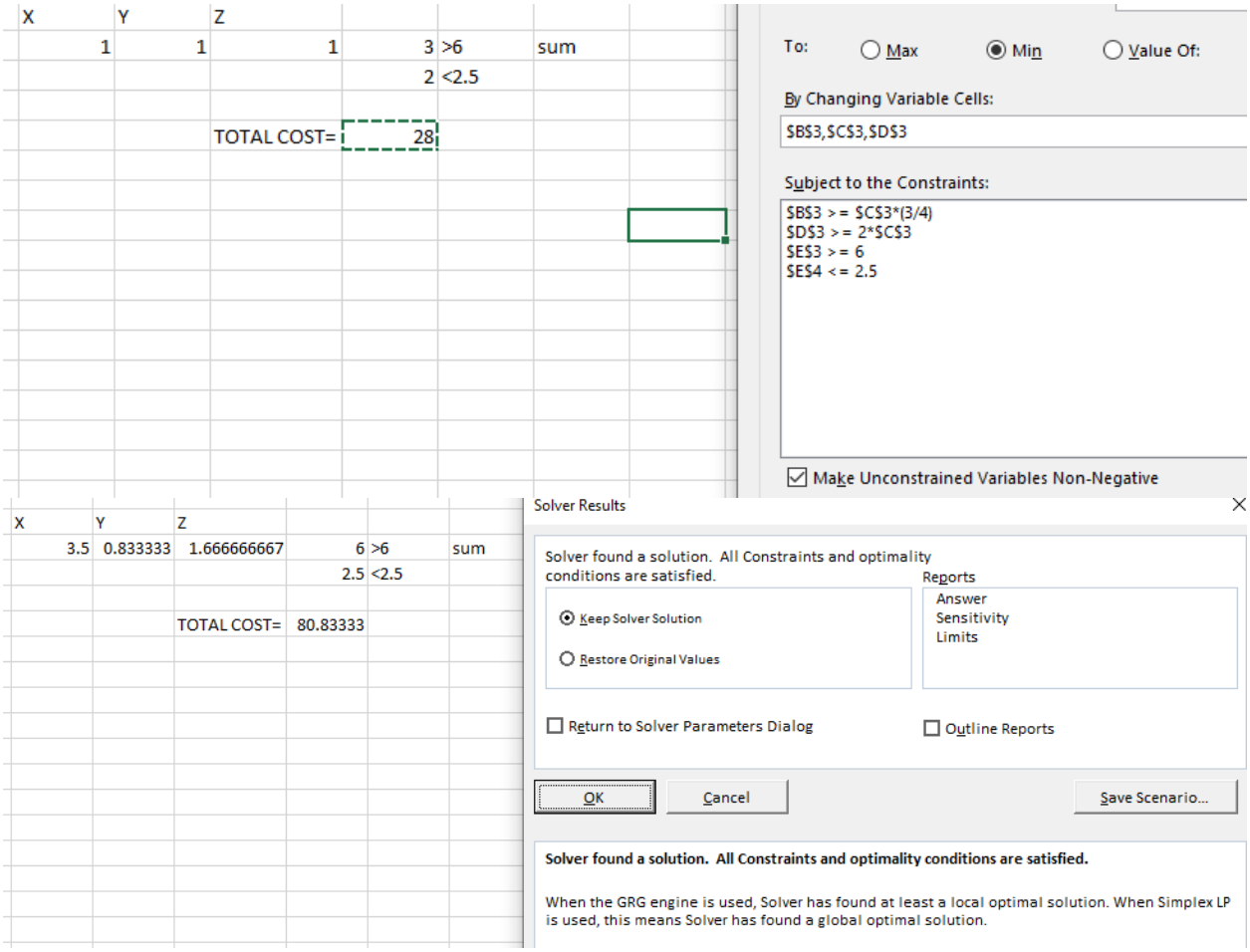


The objective function is maximized by the **point (0.5, 1)**

$$f(x, y) = -3x + y$$

$$f(0.5, 1) = -3(0.5) + (1) \quad = \quad \mathbf{-0.5}$$

2. An aerospace company is developing a new fuel additive for commercial airliners. The additive is composed of three ingredients: X, Y, and Z. For peak performance, the total amount of additive must be at least 6 mL/L of fuel. For safety reasons, the sum of the highly flammable Y and Z ingredients must not exceed 2.5 mL/L. In addition, for the additive to work, the amount of Z must **be greater than or equal to twice the amount of Y** , and the amount of X must be greater than or equal **to three quarters of the amount of Y** . If the cost per mL for the ingredients X, Y and Z is 20 cents, 3 cents, and 5 cents, respectively, use MS excel to determine the minimum cost of the additive mixture for each liter of fuel.

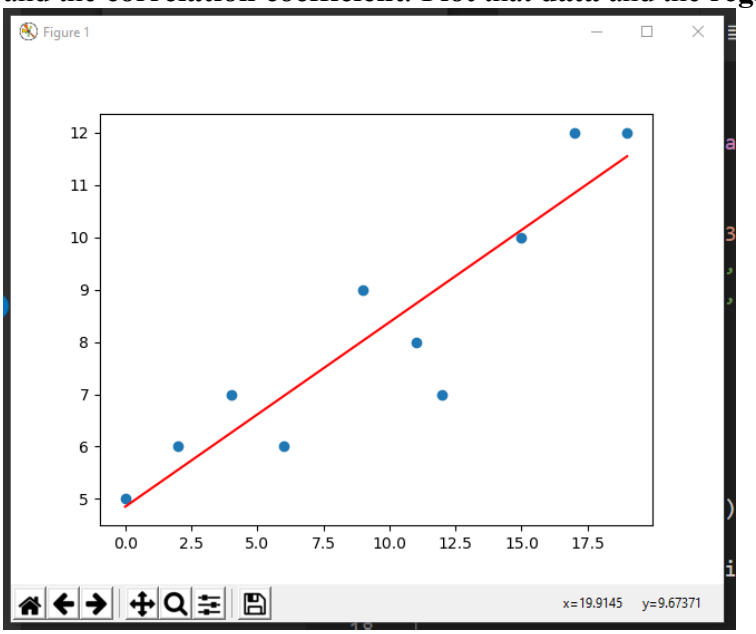


The minimum values are 3.5, 0.8333, and 1.6666 for X, Y, and Z respectively
The **total Cost minimized is 80.8333 cents**

3. Use least squares regression to fit a straight line to the data

x 0 2 4 6 9 11 12 15 17 19
y 5 6 7 6 9 8 7 10 12 12

Along with the **slope** and **intercept**, compute the **standard error** of the estimate and the **correlation coefficient**. **Plot** that **data** and the **regression line**.



Slope	**0.35246995994659547**
Intercept	**4.851535380507343**
standard error	**1.0650096999000411**
correlation coefficient	**0.9147672848789046**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL   1: Python Debu

bash-3.2$  env DEBUGPY_LAUNCHER_PORT=63800 /Library/Frameworks/Python.framework/Versions/3.7
/bin/python3 /Users/zenif/.vscode/extensions/ms-python.python-2020.3.69010/pythonFiles/lib/p
ython/debugpy/wheels/debugpy/launcher /Users/zenif/Documents/GitHub/NumMethods/HW/HW4/main.p
y
Starting Prob 3
Slope:  0.35246995994659547 Y Intersept: 4.851535380507343
Standard Error:  1.0650096999000411
Correlation Coefficient r:  0.9147672848789046
(sy/sx)*r:  0.35246995994659547  == m: 0.35246995994659547
True
```

Main.py

```
def prob3(x,y):
    print("Starting Prob 3")
    # x = [ 0, 2, 4, 6, 9, 11, 12, 15, 17, 19]
    # y = [ 5, 6, 7, 6, 9, 8 , 7 , 10, 12, 12]

    x_sum = np.sum(x)
    y_sum = np.sum(y)
    num = 0
    den = 0
    n = len(x)
    for i in range(len(x)):

        num +=n*( x[i]*y[i])
        den += n*x[i]**2

    num =  num - x_sum*y_sum
    den = den - (x_sum)**2

    m = num / den
    c =( y_sum - m * x_sum)/n

    print ("Slope: ",m,"Y Intersept:", c)
    e=0#[None]*len(x)

    sx = np.std(x)
    sy = np.std(y)
    x_mean = np.mean(x)
    y_mean = np.mean(y)
    r = 0
    for i in range(len(x)):
        # Standard Error Calulation
        ycal=(x[i]*m+c)
        e += (y[i]-ycal)**2

        # Colication Calulation
        zx = (x[i]-x_mean)/sx
        zy = (y[i]-y_mean)/sy
        r += zx*zy


    e =(e/(n-2))**0.5# square Root
    print("Standard Error: ",e)

    r = r/(n)
    print("Correlation Coefficient r: ",r)

    print ("(sy/sx)*r: ",(sy/sx)*r," == m:", m)
    if (sy/sx)*r == m:
        print(True)
    else:
        print(False)

    plt.scatter(x, y)
    y_values = [x[0]*m+c, x[len(x)-1]*m+c]
    x_values = [x[0], x[len(x)-1]]

    plt.plot(x_values, y_values,color='red')
    plt.show()
```

4. The following data are provided
x 1 2 3 4 5
y 2.2 2.8 3.6 4.5 5.5
Perform least squares regression to fit these data to the following model
y = a0 + a1x +a2/x
.
Note that this problem was solved in class, but here you are asked to reproduce the result on your own.

main.py

```
def prob4(x,y,numofConst):
    # x = [1  , 2  , 3  , 4  , 5  ]
    # y = [2.2, 2.8, 3.6, 4.5, 5.5]
    z = [[None]*numofConst ]*len(x)


    for i in range(len(x)):
        z[i]= [1, i+1,1/(i+1)]

    y = np.matrix(y).transpose()
    print(y)

    z = np.matrix(z)
    print(np.matrix(z))

    zt = z.transpose()
    vMat = zt*z
    print (vMat)

    ansMat = np.linalg.inv(vMat)*zt*y
    print(ansMat)

    return ansMat
```

Coefficients for the least squares regression to fit:
[0.37449664] = a1
[0.98644295] = a2
[0.84563758] = a3
y = 0.37449664 + 0.986442958x + 0.84563758/x

```
\Documents\GitHub\NumMethods\HW\HW4\main.py'
[[2.2]
 [2.8]
 [3.6]
 [4.5]
 [5.5]]
[[1.         1.         1.        ]
 [1.         2.         0.5       ]
 [1.         3.         0.33333333]
 [1.         4.         0.25      ]
 [1.         5.         0.2       ]]
[[ 5.          15.           2.28333333]
 [15.          55.           5.        ]
 [ 2.28333333  5.           1.46361111]]
[[0.37449664]
 [0.98644295]
 [0.84563758]]
PS D:\Christopher Allred\Documents\GitHub\NumMethods\HW\HW4>
```

thon: Current File (HW4) Ln 71, Col 1 (435 selected) Spaces: 4 UTF-8