LYDIA YU PS2

PROBLEM 1

a) The lowest value that the scroll bars will adjust to for the sum of the residuals is 77.947. The optimal linear estimator is $y = .67x + 20.61$ with the sum of the residuals = 77.894.

b) The lowest value that the scroll bars will adjust to is 849.14. The optimal linear estimator is $y = 0.46x + 33.79$ with the sum of the residuals = 849.14.

PROBLEM 2

a) decision variables: amount of radiation dose to healthy tissue voxels V_1, V_3, V_4, and V_9

objective function: minimize $z = V_1 + V_3 + V_4 + V_5 + V_9$

constraints:

$V_1 = 1.0(b_1 + b_4)$ $\quad V_3 = 2.0(b_1 + b_6)$ $\quad V_4 = 1.0(b_2 + b_4)$ $\quad V_9 = 2.5(b_3 + b_6)$

$V_2, V_6, V_7, V_8 \geq 7 \rightarrow 2(b_1 + b_5),\ 2.5(b_2 + b_6),\ 1.5(b_3 + b_4),\ 1.5(b_3 + b_5) \geq 7$

$V_5 \leq 5 \rightarrow 2(b_2 + b_5) \leq 5$

b) optimal solution = 31.017 with V_2 receiving 4 doses, V_6 receiving 4, and V_8 receiving 3. This does make sense intuitively because you are mostly hitting the cancerous voxels and hitting the other voxels less.

c) max dose: 4 optimal solution: 31.7667

3 opt. soln: 32.517

2 opt. soln: 33.267

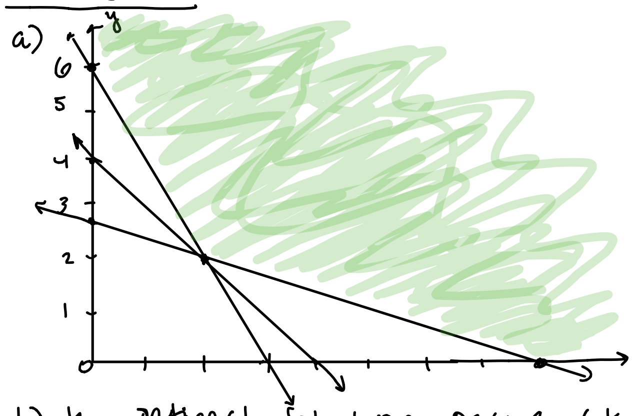
1 opt. soln: 34.017

0 opt. soln: 34.767

The trade-off for hitting less of the spinal cord means that you will hit more of the surrounding healthy tissue, since the optimal solution increases as you decrease the max. dosage to the spinal cord.

PROBLEM 3

a)



extreme pts: (2,2), (8,0), (0,6)

b) the optimal solution occurs at (2,2) and is equal to 4

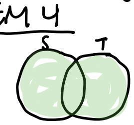
c) $x+y \geq 4$ is redundant

d) there is only one optimal solution because the objective function value at (8,0) and (0,6) is greater

e) Since the region is not bounded on all sides, it is not possible to find the optimal solution for maximizing. You could keep increasing x and y to ∞.

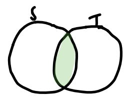
PROBLEM 4

a) i.



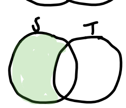
not convex

ii



convex

iii



not convex

b) i. True
ii. True
iii False

c) i. True
ii. True
iii. True
iv. False

PROBLEM 5

a) the optimal solution is 2.0 with $x_1 = 0$ and $x_2 = 2$ when the constraint = 6. When the constraint = 7, the optimal solution is 2.333 with $x_1 = 0$ and $x_2 = 2.333$.

```
#PROBLEM 5A

using JuMP, Gurobi
model = Model(with_optimizer(Gurobi.Optimizer))

#variables
@variable(model, x1>=0)
@variable(model, x2>=0)

@objective(model, Max, 2x1+x2)

#constraints
@constraint(model, 12x1+3x2<=7)
@constraint(model, -3x1+x2<=7)
@constraint(model, x2<=10)

model
```

```
Academic license - for non-commercial use only
```

$$
\begin{aligned}
\max \quad & 2x1 + x2\\
\text{Subject to} \quad & x1 \geq 0.0\\
& x2 \geq 0.0\\
& 12x1 + 3x2 \leq 7.0\\
& -3x1 + x2 \leq 7.0\\
& x2 \leq 10.0
\end{aligned}
$$

```
optimize!(model)

println("Objective Value: ", objective_value(model))
println("x1: ", value(x1))
println("x2: ", value(x2))
```

```
Optimize a model with 3 rows, 2 columns and 5 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+01]
  Objective range  [1e+00, 2e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [7e+00, 1e+01]
Presolve removed 3 rows and 2 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.3333333e+00   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.00 seconds
Optimal objective  2.333333333e+00
Objective Value: 2.3333333333333335
x1: 0.0
x2: 2.3333333333333335
```

b) The optimal solution is 0.3007 with $x1 = 2.4$, $x2 = 0$, $x3 = 8.1$, $x4 = 0$, $x5 = 0.5$, and $y = 1$. With the given changes, the optimal solution is 0.2887 with $x1 = 1.9$, $x2 = 0$, $x3 = 8.75$, $x4 = 0$, $x5 = 0.35$, and $y = 1$.

In [15]:

```julia
#PROBLEM 5B

using JuMP, Gurobi, CSV
model = Model(with_optimizer(Gurobi.Optimizer))

data = CSV.read("ps2_Portfolio1.csv", header = true)
rating = data[:,4]
maturity = data[:,5]
afterTax = data[:,8]

AvgBankQuality = 1.3
AvgYearsToMaturity = 4.8


@variable(model, x[1:5]>=0)
@variable(model, 0<=y<=1)

println(float(rating[1]))
@constraint(model, (sum(x[1:5])-y)<=10)
@constraint(model, x[2]+x[3]+x[4]>=4)
@constraint(model, sum((rating[i])*x[i] for i in 1:5)<=AvgBankQuality*sum(x[1:5]))
@constraint(model, sum(maturity[i]*x[i] for i in 1:5)<=AvgYearsToMaturity*sum(x[1:5]))

@objective(model, Max, (sum(afterTax[i]*x[i] for i in 1:5)-0.0275*y))

optimize!(model)

println("Objective Value: ", objective_value(model))
for i in 1:5
    println("x", i, ": ", value(x[i]))
end
println("y: ", value(y))
```

```
Academic license - for non-commercial use only
2.0
Optimize a model with 4 rows, 6 columns and 19 nonzeros
Coefficient statistics:
  Matrix range     [3e-01, 1e+01]
  Objective range  [2e-02, 4e-02]
  Bounds range     [1e+00, 1e+00]
  RHS range        [4e+00, 1e+01]
Presolve time: 0.00s
Presolved: 4 rows, 6 columns, 19 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.6200000e+29   7.250000e+30   1.620000e-01      0s
       3    2.8870000e-01   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.00 seconds
Optimal objective  2.887000000e-01
Objective Value: 0.2887
x1: 1.8999999999999995
x2: 0.0
x3: 8.750000000000002
x4: 0.0
x5: 0.3500000000000003
y: 1.0
```

PROBLEM 6

a) decision variables: x_0, y_0 = the xy coordinates of the coordinates of the warehouse.

Objective function: Min. $z = \sum_{i=1}^{5} \text{shipment}_i \sqrt{(x_0-x_i)^2+(y_0-y_i)^2}$

shipment_i = # shipments/year for customer i

x_i, y_i = x y coordinates of customer i

b) optimal location: (3.068, 7.076)

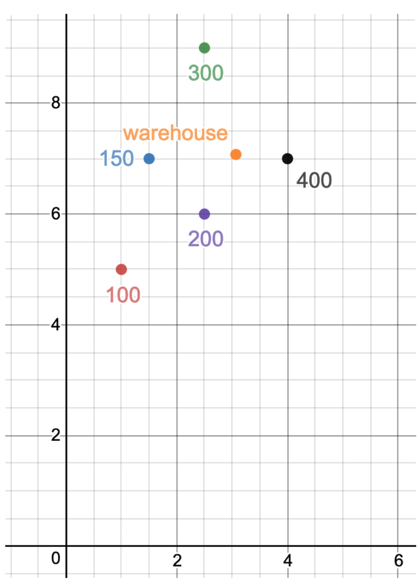
Minimized total distance value: 1747.71

```julia
using JuMP, Ipopt
model = Model(with_optimizer(Ipopt.Optimizer))

x = [1, 1.5, 2.5, 2.5, 4]
y = [5, 7, 9, 6, 7]
shipments = [100, 150, 300, 200, 400]

@variable(model, x0)
@variable(model, y0)

@NLobjective(model, Min, sum(shipments[i]*sqrt((x0-x[i])^2+(y0-y[i])^2) for i in 1:5))

optimize!(model)
```

```julia
println("Objective Value: ", objective_value(model))
println("x0: ", value(x0))
println("y0: ", value(y0))
```

```
Objective Value: 1747.7096891328363
x0: 3.0679192152498205
y0: 7.076400277458581
```

c)



This does seem correct because the warehouse is located closer to the customers with higher shipment weights.