Homework 6

Max Wagner

February 25, 2015

251.2 .

Protein = (0.5 x Hay) + (1 x Oats) + (2 x Feed) + (6 x HighP)

Carbs = (92 x Hay) + (4 x Oats) + (0.5 x Feed) + (1 x HighP)

Roughage = (5 x Hay) + (2 x Oats) + (1 x Feed) + (2.5 x HighP)

Cost = (1.8 x Hay) + (3.5 x Oats) + (0.4 x Feed) + (1 x HighP)

Protein > 40

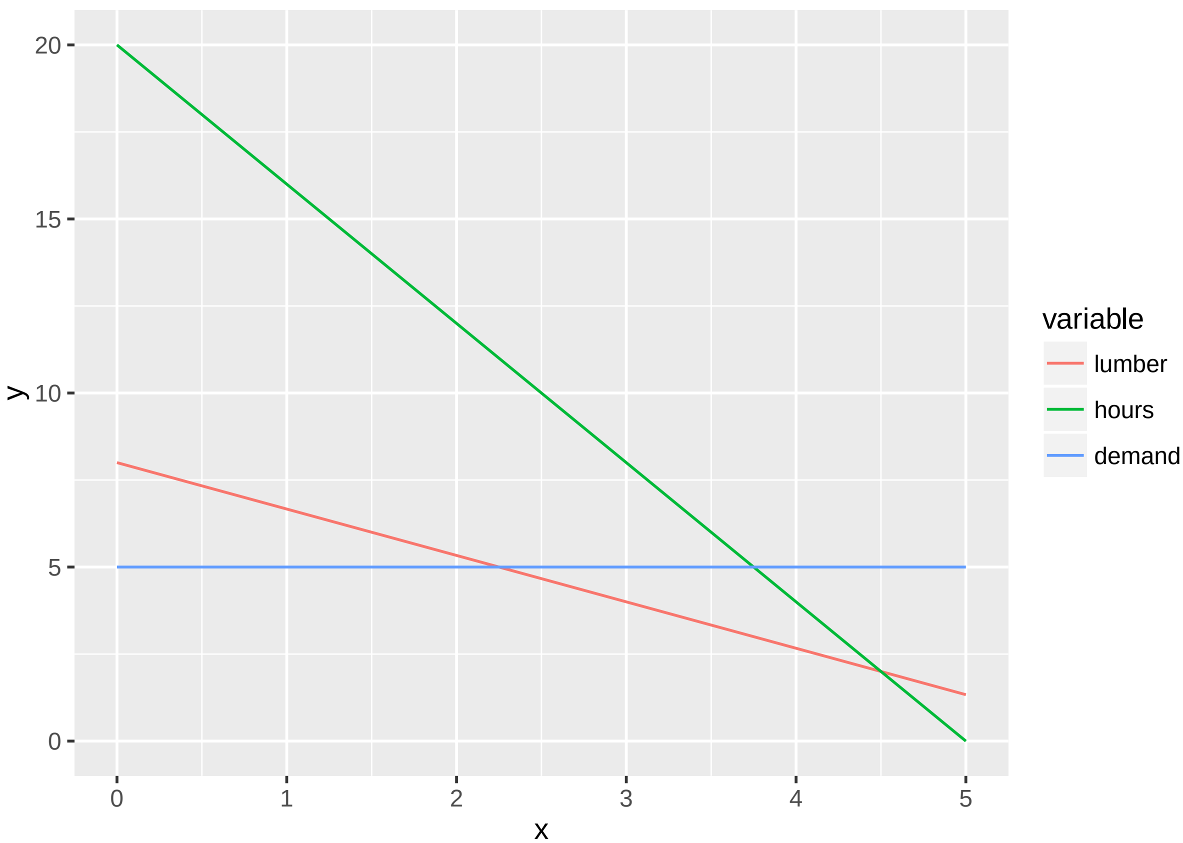
Carbs > 20

Roughage > 45

The model would optimize a diet where the constraits are met, and the price is the lowest.

264.6 Let's put them in terms of y, and graph.

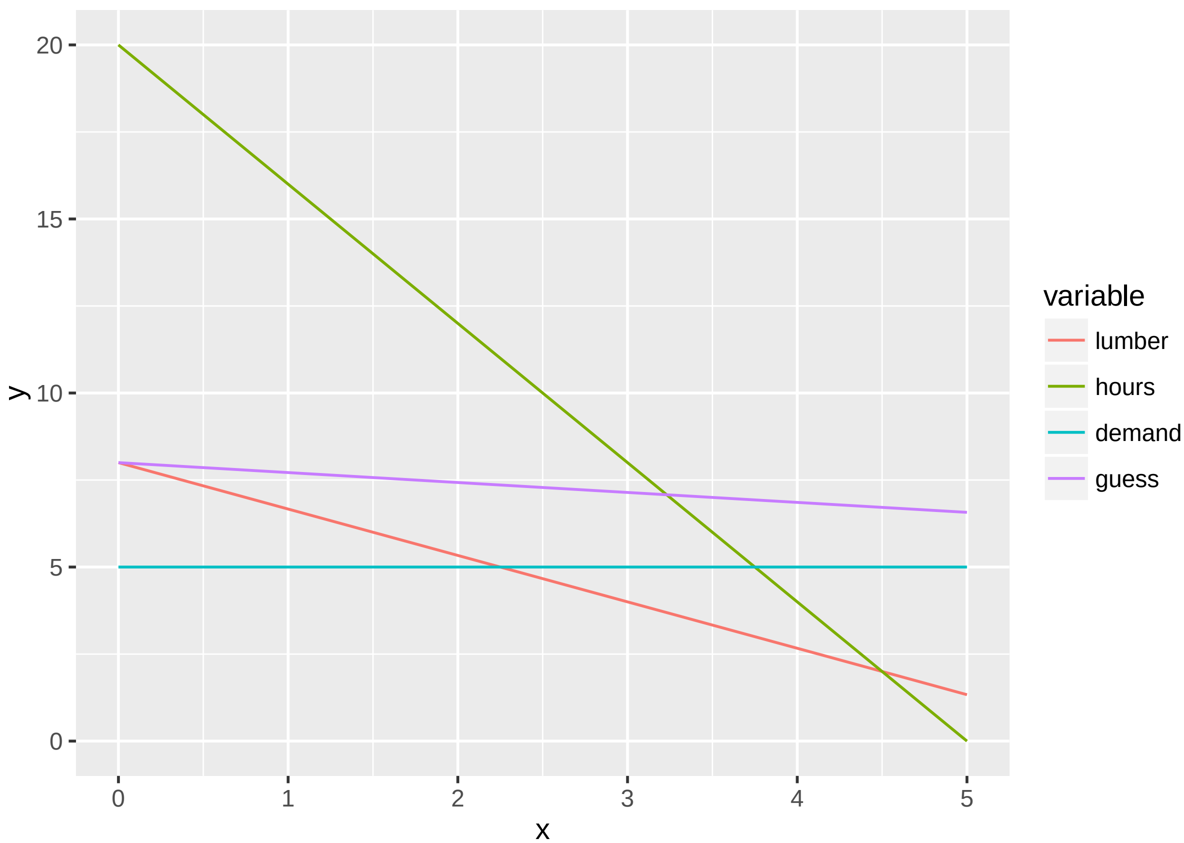
```
library(reshape2)
library(ggplot2)

x <- seq(0,5,by=0.5)
graph1 <- data.frame(x = x, lumber = (48 - (8*x)) / 6, hours = 20 - (4*x), demand = 5)
graph1 <- melt(graph1, "x", value.name="y")
ggplot(graph1, aes(x=x, y=y, color=variable)) + geom_line()
```



The triangular area formed by y=5 and 8x + 6y = 48 is the area we can focus on. From the slope of 10x + 35y, we can see that we should look at (0,8).

More graphing to check that 35*8 = 280 fits the solution

```
x <- seq(0,5,by=0.1)
graph2 <- data.frame(x = x, lumber = (48 - (8*x)) / 6, hours = 20 - (4*x), demand = 5, guess = (280 - (
graph2 <- melt(graph2, id.vars="x", value.name="y")
ggplot(graph2, aes(x=x, y=y, color=variable)) + geom_line()
```



268.6 We can use the graphs from above to sort of check our math in this section. I'll start with listing the varibles that we will eventually set to 0. We know y cannot be 0, so we are left with x, c1, c2, and c3.

$$8x + 6y + c1 \leq 48$$
$$4x + y + c2 \leq 20$$
$$y + c3 \geq 5$$

x, c1 gives the point (0, 8)

$$6y \leq 48$$
$$y + c2 \leq 20$$
$$y - c3 \geq 5$$

x, c2 gives an infeasable point.

x, c3 gives the point (0, 5)

$$6y + c1 \leq 48$$
$$y + c2 \leq 20$$
$$y \geq 5$$

c1, c2 gives an infeasable point.

c1, c3 gives the point (2.25, 5)

$$8x + 6y \leq 48$$
$$4x + y + c2 \leq 20$$
$$y \geq 5$$

c2, c3 gives an infeasable point.

The above feasable points give the potential solutions of (0, 5), (0, 8), (2.25, 5) which corresponds with 10x + 35y values of 175, 280, and 197.5. Again like in the previous section, (0, 8) gives the maximized solution.

278.6 .

```
library(lpSolveAPI)
x <- make.lp(2, 2)
y <- x
set.column(x, 1, c(1, 2))
set.column(x, 2, c(3, 4))
```