Assignment 3: Linear Programming project

Francis Vo, Soo-Hyun Yoo

November 6, 2012

1 Problem 1: mmmm . . . pork

1.1 Mathematical Model

1.1.1 Objective Function

$$\begin{aligned}\max(&8 \cdot \text{ham}_f + 12 \cdot \text{ham}_r + 11 \cdot \text{ham}_o + \\ &4 \cdot \text{bellies}_f + 12 \cdot \text{bellies}_r + 7 \cdot \text{bellies}_o + \\ &4 \cdot \text{picnics}_f + 13 \cdot \text{picnics}_r + 9 \cdot \text{picnics}_o)\end{aligned}$$

1.1.2 Constraints

$$\begin{aligned}\text{ham}_f + \text{ham}_r + \text{ham}_o &\leq 480 \\ \text{bellies}_f + \text{bellies}_r + \text{bellies}_o &\leq 400 \\ \text{picnics}_f + \text{picnics}_r + \text{picnics}_o &\leq 230 \\ \text{ham}_r + \text{bellies}_r + \text{picnics}_r &\leq 420 \\ \text{ham}_o + \text{bellies}_o + \text{picnics}_o &\leq 250\end{aligned}$$

1.2 Standard Form

1.2.1 Objective Function

$$\begin{aligned}\max(&8 \cdot \text{ham}_f + 12 \cdot \text{ham}_r + 11 \cdot \text{ham}_o + \\ &4 \cdot \text{bellies}_f + 12 \cdot \text{bellies}_r + 7 \cdot \text{bellies}_o + \\ &4 \cdot \text{picnics}_f + 13 \cdot \text{picnics}_r + 9 \cdot \text{picnics}_o)\end{aligned}$$

1.2.2 Constraints

$$\begin{aligned}
\text{ham}_f + \text{ham}_r + \text{ham}_o + \text{ham}_{\text{remain}} &= 480 \\
\text{bellies}_f + \text{bellies}_r + \text{bellies}_o + \text{bellies}_{\text{remain}} &= 400 \\
\text{picnics}_f + \text{picnics}_r + \text{picnics}_o + \text{picnics}_{\text{remain}} &= 230 \\
\text{ham}_r + \text{bellies}_r + \text{picnics}_r + \text{smoke}_{\text{reg}} &= 420 \\
\text{ham}_o + \text{bellies}_o + \text{picnics}_o + \text{smoke}_{\text{over}} &= 250 \\
\text{ham}_{\text{remain}},\ \text{bellies}_{\text{remain}},\ \text{picnics}_{\text{remain}},\ \text{smoke}_{\text{reg}},\ \text{smoke}_{\text{over}} &\geq 0
\end{aligned}$$

1.3 Matrix Form

We wish to find $\max(cx)$, where

$$c = \begin{pmatrix} 8 & 14 & 11 & 4 & 12 & 7 & 4 & 13 & 9 \end{pmatrix},$$

and

$$x = \begin{pmatrix} \text{ham}_f \\ \text{ham}_r \\ \text{ham}_o \\ \text{bellies}_f \\ \text{bellies}_r \\ \text{bellies}_o \\ \text{picnics}_f \\ \text{picnics}_r \\ \text{picnics}_o \end{pmatrix} \geq 0.$$

Also,

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \text{ham}_f \\ \text{ham}_r \\ \text{ham}_o \\ \text{bellies}_f \\ \text{bellies}_r \\ \text{bellies}_o \\ \text{picnics}_f \\ \text{picnics}_r \\ \text{picnics}_o \end{pmatrix} \leq \begin{pmatrix} 480 \\ 400 \\ 230 \\ 420 \\ 250 \end{pmatrix}.$$

1.4 Solution

Total net profit: \$10,910

	fresh	smoked on regular time	smoked on overtime
hams	440	0	40
bellies	0	400	0
picnics	0	20	210

1.5 Using an LP-solver

We used the GNU Linear Programming Kit to solve this problem. The help page tells us that we should give a model file (provided in the next section) to glpsol, which we do by running: `glpsol -m pork.mod -o pork.sol`.

1.6 Code

```
/* Decision variables */

var ham_f >=0;       /* ham */
var ham_r >=0;       /* ham */
var ham_o >=0;       /* ham */

var bellies_f >=0;  /* bellies */
var bellies_r >=0;  /* bellies */
var bellies_o >=0;  /* bellies */

var picnics_f >=0;  /* picnics */
var picnics_r >=0;  /* picnics */
var picnics_o >=0;  /* picnics */


/* Objective function */
maximize z: 8 * ham_f + 12 * ham_r + 11 * ham_o + 4 * bellies_f + 12 * bellies_r + 7 *
    bellies_o + 4 * picnics_f + 13 * picnics_r + 9 * picnics_o;


/* Constraints */

s.t. Ham             : ham_f + ham_r + ham_o <= 480;
s.t. Bellies         : bellies_f + bellies_r + bellies_o <= 400;
s.t. Picnics         : picnics_f + picnics_r + picnics_o <= 230;
s.t. Smoke_Regular   : ham_r + bellies_r + picnics_r <= 420;
s.t. Smoke_Overtime  : ham_o + bellies_o + picnics_o <= 250;

end;
```

pork.mod

2 Problem 2: least squares isnt good enough for me

2.1 Mathematical Model

2.1.1 Objective function

$\min(t)$

2.1.2 Constraints

Given a set of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$,

$$\begin{aligned} |x_i - b| &\leq t, \text{ and} \\ |ax_i + by_i - c| &\leq t, \end{aligned}$$

for $1 \leq i \leq n$.

2.2 Standard Form

2.2.1 Objective Function

$\min(t)$

2.2.2 Constraints

Given a set of points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$,

$$\begin{aligned} x_i - b + v_i &= t \\ x_i - b &\geq -t \\ ax_i + by_i - c + z_i &= t \\ ax_i + by_i - c &\geq -t \\ v_i, z_i &\geq 0 \end{aligned}$$

for $1 \leq i \leq n$.

2.3 Solution

The solution for the specific problem is as follows:

$$a = -8.8$$

$$b = 5.5$$

$$c = 12$$

This best fit line is shown in Figure 1.

2.4 Plot

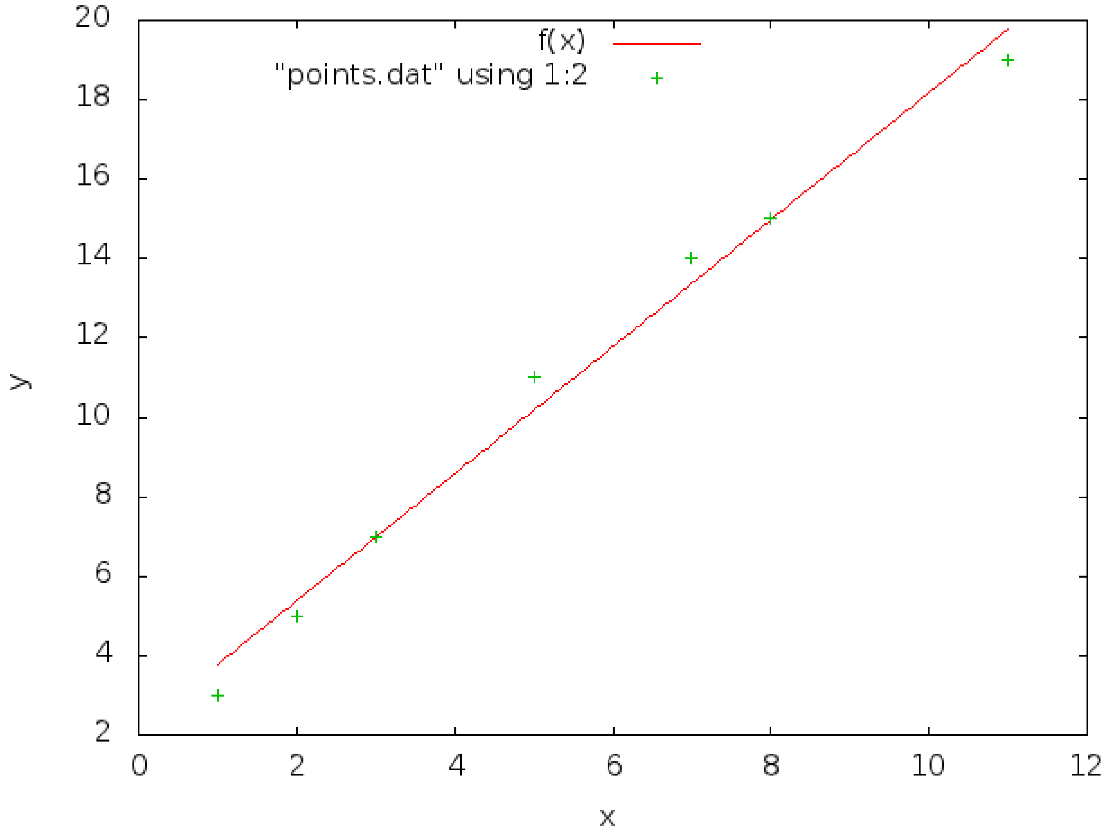

Figure 1: Points and best fit line.

2.5 Code

```
/* Decision variables */

var a; var b; var c; var t;

/* Objective function */
minimize z: t;

/* Constraints */

/*s.t. point_y_high : 19-3+a <= t;
s.t. point_y_low : 19-3+a >= -t;*/

/* For each point make sure b is set right
 * without this a=b=c=t=0
 */
s.t. point_x_1_high : 1  - b <=  t;
s.t. point_x_1_low  : 1  - b >= -t;
s.t. point_x_2_high : 2  - b <=  t;
s.t. point_x_2_low  : 2  - b >= -t;
s.t. point_x_3_high : 3  - b <=  t;
s.t. point_x_3_low  : 3  - b >= -t;
s.t. point_x_4_high : 5  - b <=  t;
s.t. point_x_4_low  : 5  - b >= -t;
s.t. point_x_5_high : 7  - b <=  t;
s.t. point_x_5_low  : 7  - b >= -t;
s.t. point_x_6_high : 8  - b <=  t;
s.t. point_x_6_low  : 8  - b >= -t;
s.t. point_x_7_high : 10 - b <=  t;
s.t. point_x_7_low  : 10 - b >= -t;

/* minimizes the maximum absolute deviation */
s.t. point_1_high : a*(1)  + b*(3)  - c <=  t;
s.t. point_1_low  : a*(1)  + b*(3)  - c >= -t;
s.t. point_2_high : a*(2)  + b*(5)  - c <=  t;
s.t. point_2_low  : a*(2)  + b*(5)  - c >= -t;
s.t. point_3_high : a*(3)  + b*(7)  - c <=  t;
s.t. point_3_low  : a*(3)  + b*(7)  - c >= -t;
s.t. point_4_high : a*(5)  + b*(11) - c <=  t;
s.t. point_4_low  : a*(5)  + b*(11) - c >= -t;
s.t. point_5_high : a*(7)  + b*(14) - c <=  t;
s.t. point_5_low  : a*(7)  + b*(14) - c >= -t;
s.t. point_6_high : a*(8)  + b*(15) - c <=  t;
s.t. point_6_low  : a*(8)  + b*(15) - c >= -t;
s.t. point_7_high : a*(10) + b*(19) - c <=  t;
s.t. point_7_low  : a*(10) + b*(19) - c >= -t;

end
```

bestFit.mod