GAMS Exercises

Outline

- Introduction to GAMS
- Computing basic DEA models by GAMS

Good Programming Practices

- Comment
- Separate data and routines
- Write routines separately for testing and debugging
- Save file periodically with different name to allow yourself to go back to pervious versions you have tried
- Name variables and files to help with organization

Resources

- Bruce McCarl's User Guide

  *www.**gams**.com/dd/docs/bigdocs/**gams**2002/**mccar**
  lgamsuser**guide**.pdf*

- Help in the top bar of GAMS

Components of a Program

- Header
- Define
  - Sets
  - Tables
  - Parameters
  - Variables
  - Equations
  - Model
- Execution
- Display results

Header

- Text, specific formatting not required
- Use $ontext and $offtext command to define header area
- Name of the program
- Name of the programmer
- Date created
- Citations to references used when writing code
- Brief summary of data and variables

Define Sets

- Strength of GAMS
- State the (index) (“descriptor of the set”) (/range/)
- i index of firms /i1*i89/
- Note the range can be a discrete list /Electricity,Length,Customers/
- Can define subsets big(i) where now big could be only the big firms
- End all sections with an “;”

Define Tables

- Multi-dimensional tables are possible, but here we focus on 2-D tables; see McCarl's guide for higher dimensions
- (table name) (index for the vertical dimension , index for the horizontal dimension)
- Example: data(i,j)
- Then include the data below either directly or read in the data and always end with ";"
- Remember reading in the data will allow your program to be reused more easily

Define Parameters

- Parameters (matrix, vector, scalar);
  * note GAMS used the term parameters to mean data that is read in or written values, not variables determined by the optimization process
  - [parameter name(index for the vertical dimension , index for the horizontal dimension) ‘description’]
  - Example: x(i,l) 'inputs of firm i‘
  - End all sections with an “;”

Map Parameters

- This can be done anywhere in the program after defining the parameters
  - [parameter name(index for the vertical dimension , index for the horizontal dimension) = table name(index for the vertical dimension , index for the horizontal dimension)]
  - Example: x(i,inp) = data(i,inp);
  - End all sections with an “;”

Define Variables

- Variables (matrix, vector, scalar);
  - [variable name(index for the vertical dimension , index for the horizontal dimension) ‘description’]
  - Example: w(i,inp) input weights
  - End all sections with an “;”
  - Can also use the command “Positive Variables”

Define Equations

- Equations (matrix, vector, scalar);
  - [equation name(indices) ‘description’]
  - Example: conv(i,h) convexity
  - End all sections with an “;”
  - Can also use the command “Positive Variables”

Specify Equations

- This can be done anywhere in the program, but I prefer to do it right after defining the equations
  - [equation name(indices) .. equation]
  - Example:

conv(i,h)..

d(i) - d(h) =g= (sum(outp,p(i,outp)*y(h,outp))- sum(inp, w(i,inp)*x(h,inp))) - (sum(outp,p(i,outp)*y(i,outp)) - sum(inp, w(i,inp)*x(i,inp)));

  - End all sections with an ";"

Define Model

- [model name ‘description’ /list equations/]
- Example: cnls model / all / ;
- End all sections with an “;”
- Can use all to mean include all equations that have been defined

Execution

- Loops can be used to execute multiple optimization programs as in the case of DEA

```
loop(iter,
   x0(inp) = x(inp, iter);
   y0(outp) = y(outp, iter);

   solve dea using lp maximizing eff;
   abort$(dea.modelstat<>1) "LP was not optimal";
   efficiency(iter) = eff.l;
);
```

- Solve cnls using nlp minimizing sse ;

Execution

```
loop(iter,
   x0(inp) = x(inp, iter);
   y0(outp) = y(outp, iter);

   solve dea using lp maximizing eff;
   abort$(dea.modelstat<>1) "LP was not optimal";
   efficiency(iter) = eff.l;
);
```

- Define the parameters that are changing between optimization problems within the loop

Execution

```
loop(iter,
   x0(inp) = x(inp, iter);
   y0(outp) = y(outp, iter);

   solve dea using lp maximizing eff;
   abort$(dea.modelstat<>1) "LP was not optimal";
   efficiency(iter) = eff.l;
);
```

- Solve
  - [(model name) using (type of optimization program) (objective criteria)]
  - Example: dea using lp maximizing eff;
  - End all sections with an “;”

Execution

```
loop(iter,
   x0(inp) = x(inp, iter);
   y0(outp) = y(outp, iter);

   solve dea using lp maximizing eff;
   abort$(dea.modelstat<>1) "LP was not optimal";
   efficiency(iter) = eff.l;
);
```

- Include flags to indicate problems
  - Example: abort$(dea.modelstat<>1) "LP was not optimal";
  - Not necessary but help to make sure you are really calculating what you expect

Execution

```
loop(iter,
   x0(inp) = x(inp, iter);
   y0(outp) = y(outp, iter);

   solve dea using lp maximizing eff;
   abort$(dea.modelstat<>1) "LP was not optimal";
   efficiency(iter) = eff.l;
);
```

- Write all relevent variables determined by the optimization problem
  - Example: efficiency(iter) = eff.l;
  - These will be overwritten in the next iteration of the loop, so we should record them
  - “varible.l” means write the current level of the variable

Show DEA models

- deaVRS
  - Reading in text files
  - Reading in excel files
  - Outputing results

Display

- Each time a GAMS file is executed it outputs a .lst file
- The command “Display” can be used to output information to the .lst file

option d:4:0:1;

display d.l, w.l,p.l;

- The option command “itemname:decimal:Rowitems:Columnitems”
- First specify the variable, the number of decimal places to display, the number of rows and the number of columns to display the data
- Zero indicated the number of rows will be determined by the data size

Reading Data

- Excel
- the xlimport command can be used to read data in from excel files
- Example

  $libinclude xlimport Y C:\YEMV.xls a1:d86
  $libinclude xlimport C C:\Ctot.xls a1:cg2
  $libinclude xlimport Z C:\ZEMV.xls a1:cg2

- The above command assumes the data files have been saved in the roor of drive C:

Exporting Results Data

- Excel
- the xldump command can be used to export data to excel files
- Example

$libinclude xldump Bvalue C:\StoNED_results.xls Beta a1:cw120

$libinclude xldump Dvalue C:\StoNED_results.xls Delta a1:cw120

$libinclude xldump Evalue C:\StoNED_results.xls Residuaalit a1:cw120

- The above command saves the results to the roor of drive C:

Short Routines

```
set r /rnk1*rnk1000/;
parameter rank(i);
alias (i,ii);
rank(i) = sum(ii$(esum(ii)>=esum(i)), 1);
parameter e2(r,i);
e2(r,i)=esum(i)$(rank(i)=ord(r));
option e2:4:0:1;
display e2;
```