Introduction to GAMS

G Cornelis van Kooten

GAMS

- Powerful software for solving LP, NLP, MLP and CGE models.
- Easy to use when setting up problems, especially in the case of very large problems: much easier than Excel.
- More powerful than Excel, Matlab, Maple, Mathematica, etc. for solving very large, non-linear programs
- Enables researcher to access a variety of powerful solvers using the same program. Solvers include Minos, CPlex, ConOpt, Lindo, XA, many other commercial solvers

GAMSIDE

- GAMS: General Algebraic Modeling System

plus

- IDE: Integrated Development Environment

  A Windows graphical interface to run GAMS

GAMSIDE

GAMS is used in two phases.

- First, one uses a text editor and creates a file which contains GAMS instructions.
- Second, one submits that file to GAMS which executes those instructions causing calculations to be done, solvers to be used and a solution file of the execution results to be created.

Two ways to do this.

- Traditional method – use a text editor set up the model then use DOS (or UNIX) command line instructions to find errors in and run the model.
- The GAMSIDE alternative. It is a graphical interface to create, debug, edit and run GAMS files.

Summary steps for using GAMS

1. Install GAMS and the IDE on your computer making an icon
2. Open the IDE through the icon 

3. Create a project by going to the file selection in the upper left corner.

What is a project? The GAMSIDE employs a "Project" file (1) to identify a location where all saved files are to be placed (to place files elsewhere use the save as dialogue) and where GAMS looks for files when executing, and (2) to which project files and program options are saved. It is recommended that you define a new project every time you wish to change the file storage directory.

Summary steps for using GAMS

4. Define a project name and location. Put it in a directory you want to use. Choose a suitable subdirectory and create a subdirectory with the name of the project. In turn, call your project by that name so that it is called *project.gpr*, where gpr stands for GAMS project.
5. Create or open an existing file of GAMS instructions
6. Prepare the file so you think it is ready for execution
7. Run the file with GAMS by clicking on the run button
8. Open and navigate around the output

A GAMS Program

INPUTS:

- SETS

  --declaration

  --assignment of members

- DATA (Parameters, Tables, Scalars)

  --declaration

  --assignment of values

A GAMS program (cont)

- VARIABLES
  --declaration
  --assignment of type

- Assignment of bounds and/or initial values (OPTIONAL)

A GAMS program (cont)

- EQUATIONS
  --declaration
  --definition

- MODEL and SOLVE Statements

- DISPLAY Statement (OPTIONAL)

Equations in GAMS

Two types of equations in GAMS:

1. Define parameters
2. Model equations (may be inequalities)

Not indexed: Objective is a single value

Z = sum((i,j), a(i,j)*x(i,j));

Indexed: Examples

R(i) = sum(j, c(j)*x(i,j));

R(item) = sum(city, c(city)*x(item,city));

Example problem: Minimize transportation costs

$$\min \sum_i \sum_j c_{ij} x_{ij} \ (\$)$$

$$s.t. \sum_j x_{ij} \le a_i \quad (\text{supply limit at plant } i)$$

$$\sum_i x_{ij} \ge b_j \quad (\text{satisfy demand at market } j)$$

```
SETS
i     plants   /Seattle, San-Diego/;
j     market /NY, Chicago, Kancity/;

PARAMETERS
a(i)    capacity at plant i
       / Seattle              350
         San-Diego            600 / ;

b(j)    demand at market j
       / NY                   325
         Chicago              300
         Kancity              275 / ;
```

TABLE $d(i, j)$ distance in ‘000s miles

	NY	Chicago	Kancity
Seattle	2.5	1.7	1.8
San-Diego	2.5	1.8	1.4

;

SCALAR f freight cost in $ per ‘000s miles /8/ ;

PARAMETER $c(i,j)$ transport cost in k$ per unit ;

$c(i,j) = f * d(i,j)/1000;$

VARIABLES

$x(i,j)$ shipment quantities

z total costs in k$;

Positive variable x;

EQUATIONS

supply(*i*) supply limit at plant i

demand(*j*) satisfy demand at market j

cost objective function is to minimize cost

;

cost.. z =E= sum((*i*,*j*), $c(i,j)*x(i,j)$);

supply(i).. sum(j, $x(i,j)$) =L= $a(i)$;

demand(j).. sum(*i*, $x(i,j)$) =G= $b(j)$;

MODEL transport /all/;

SOLVE transport using LP minimizing z;

DISPLAY x.l, x.m ;

x.l refers to level value of *x*, *x*.m to its marginal value;
To get shadow prices, you want supply.m and demand.m

Specify upper and lower bounds in lieu of constraints

x.up(j) = 50;
Same as an $x_j \leq 50$ constraint

y.lo("2")=20; $\rightarrow$ $y_2 \geq 20$ constraint
(Note reference to index 2)

cows.up("3") = 200 + ½ x.up("3");
$\rightarrow$ $cows_3 \leq 200 + ½ (50)$ constraint

Report Writing in GAMS

DISPLAY can be used after SOLVE statement in GAMS to print out calculation results:

1) variable name followed by .L → value of solution for that variable
2) variable name followed by .M → reduced cost associated with that variable
3) equation name followed by .L → left-hand side value of equation
4) equation name followed by .M → shadow price (dual variable) associated with equation

Dual slack variables = RHS of equation – associated .L value

Example displays level and marginal values of X and shadow price of CON1:

```
DISPLAY X.L, X.M, CON1.M;
```

PUT option and Report Writing

Define output file name and identify where output is to be written. If no path is given, default is project subdirectory and file is file name with .put extension

```
file Output /D:\work\output.txt/ ;
*Set page width
Output.pw=5000;
*Number of decimal places in output file
Output.nd=0;

Solve modelname using LP maximizing z;
```

Now comes the writing component. A comma is needed to create a comma separated value file, and a / is used to indicate a carriage return.

PUT option (cont)

```
put Output;
   put "Solver status ,", modelname.solvestat/;
   put "Model status ,", modelname.modelstat/;
   put "Objective value ,", z /;
   put "Period" "," “Price" "," “Quantity" ","/;
       loop(t, put t.tl "," price.l(t) "," quan.l(t) ", "/;);
   put/;
putclose Output;
```

Note how the index t needs to be identified in the loop with a .tl as opposed to only .l.

An example of how to write output is provided in the next two slides for an electricity grid model.

gamside: C:\Users\Kees van Kooten\Dropbox\Energy\Electricity\CapacityInvest\Science\battery.gpr - [C:\Users\Kees

File Edit Search Windows Utilities Model Libraries Help

BatterySize.gms

```
SETS g generating asset types /nuke, coal, GT, CC, Cgas, wind, solar/
     f(g) fossil fuel assets /coal, GT, CC, Cgas/
     r(g) renewable fuel assets /wind, solar /
     fn(g) fossil fuel plus nuclear /nuke, coal, GT, CC, Cgas/
     t hours in the year 2014 /1*8760/
     y years /2006*2015/
;
ALIAS(g,k);

SETS
tinit(t) first time period
tfin(t)  final time period
;
tinit(t)=yes$(ord(t) eq 1);
tfin(t)=yes$(ord(t) eq card(t));

SCALARS
  ctax    carbon tax $ per tCO2 /0/
  rate    discount rate across years for annualizing costs /0.05/
  effbat  roundtrip efficiency of the battery  /0.95/
  costbat cost of operating battery per MWh  /0/
;

* ---- IMPORT EXTRA DATA FROM SPREADSHEET ---*
* Build gdxxrw instructions file
$onecho > import.txt
par=Ywind rng=WindSW!A1:K8761 Rdim=1 Cdim=1
par=Ysolar rng=Solar!A1:K8761 Rdim=1 Cdim=1
par=Yload rng=Load!A1:K8761 Rdim=1 Cdim=1
$offecho

* Create and import GDX file with import data
$call gdxxrw.exe Alberta(2006To2015).xlsx o=ImportData.gdx @import.txt
$gdxin ImportData.gdx

* Need to declare parameters before loading them
PARAMETERS
  Ywind(t,y)    2006 to 2015 wind data in spreadsheet
  Ysolar(t,y)   2006 to 2015 solar data in spreadsheet
  Yload(t,y)    2006 to 2015 load data in spreadsheet
;

$load Ywind Ysolar Yload
$gdxin
```

```
*=== Export to Excel using GDX utilities

*=== First unload to GDX file (occurs during execution phase)
execute_unload 'results.gdx' Z2.L QuasiRent QuotaRent Welfare qflow.L qd.L qs.L PriceD.L PriceS.L

*=== Now write to variable levels to Excel file from GDX
execute 'gdxxrw.exe results.gdx o=results.xls var=Z2.l rng=well!a1'
execute 'gdxxrw.exe results.gdx o=results.xls par=QuasiRent rng=well!a2'
execute 'gdxxrw.exe results.gdx o=results.xls par=QuotaRent rng=well!a4'
execute 'gdxxrw.exe results.gdx o=results.xls par=Welfare rng=well!a6'
execute 'gdxxrw.exe results.gdx o=results.xls var=qflow.L rng=trade!a3'
execute 'gdxxrw.exe results.gdx o=results.xls var=qd.L rng=quant!a2:g12'
execute 'gdxxrw.exe results.gdx o=results.xls var=qs.L rng=quant!a14:g24'
execute 'gdxxrw.exe results.gdx o=results.xls var=PriceD.L rng=quant!a26:g36'
execute 'gdxxrw.exe results.gdx o=results.xls var=PriceS.L rng=quant!a38:g48'
```

```
MODEL Batstorage /all/;
file NoFossils /results.csv/;
NoFossils.nd=3;

loop (y,
  load(t) = Yload(t,y);
  wind(t) = Ywind(t,y);
  solar(t) = Ysolar(t,y);
  SOLVE Batstorage using LP minimizing cost;
  RevCarbon = ctax * sum((g,t), emit(g)*gen.l(g,t)) / 1000000 ;
  TotalCost = cost.l - RevCarbon + 2 * sum(fn, remove.l(fn))/1000
              + 0.15*(annual('coal')*cap.l('coal') + annual('GT')*cap.l('GT')
                      + annual('CC')*cap.l('CC'))/1000 ;
  Generation(g) = sum(t, gen.l(g,t))/1000000;
  Capacity(g) = smax(t, gen.l(g,t));
  decomm(fn) = remove.l(fn);
  addition(fn) = add.l(fn);
  carbon = sum((g,t), emit(g)*gen.l(g,t))/1000000;
  areasolar = panels.l * 0.0034/2.47;
  batpower = smax(t, discharge.l(t));
  batenergy = smax(t, v.l(t))/1000;

  DISPLAY TotalCost, RevCarbon, carbon, Generation, Capacity;

  put NoFossils 'Run on ' system.date ' using source file ' system.ifile///;
  put NoFossils 'MODEL: With Fossil Fuels'/;
  put NoFossils;
   put "Model status , " BatStorage.modelStat/;
   put "Solver status , " BatStorage.solvestat/;
   put "Total cost ($ mil) , " cost.l /;
   put "Carbon released ('000s tCO2) ," carbon/;
   put "Battery size (MW) , " batpower /;
   put "Battery size (GWh) , " batenergy /;
   put "No of 3.5 MW wind turbines , " turbines.l /;
   put "No of 0.005 MW solar panels , " panels.l /;
   put /;
   put "  , Generating, Generation "/;
   put "Asset, capacity (MW), (TWh), "/;
   loop(g, put g.tl "," Capacity(g) "," Generation(g) /);
   put //;
*   put " Movement into and out of battery (MW) and Battery Capacity (MWh) "/;
*   put "Hour, Battery, Charge, Discharge "/;
*   loop(t, put t.tl "," v.l(t) "," Charge.l(t) "," Discharge.l(t)/);
*   put //;
  putclose;
  NoFossils.ap=1;
);
```

```
file Out /Output3wind.csv/;
Out.nd=4;
put Out 'Battery problem output'/;
put Out;
 put "Model status , ", ',', Alberta.modelstat/;
 put "Solver status , ", ',', Alberta.solvestat/;
 put "Carbon tax , ", ",", taxEm/;
 put "Max wind turbine , ", ",", maxTurbine/;
 put "Max solar panel , ", ",", maxPanel/;
 put "Annual CO2 emissions (Mt), ", ',', TotalCO2/;
 put "Annual generating cost ($ mil), ", ',', Tcost /;
 put "Size of coGen plant , ", ',', coGencap.l/;
 put "Size of gas plant , ", ',', gascap.l/;
 put "Total installed capacity: Solar , ", ',', SolarCap /;
 put "Installed Wind Capacity (SE) , ", ',', WindCap("southeast") /;
 put "Installed Wind Capacity (SW), ", ',', WindCap("southwest") /;
 put "Installed Wind Capacity (North) , ", ',', WindCap("north") /;
 put "Battery size , ", ',', battery.l/;
 put /;
 put ", , , Wind, Wind, Wind,"/;
 put "Period, Load, Solar, SE, SW, N, CoGen, Gas, Storage, Charge, Discharge"/;
 loop(t, put t.tl "," demand(t) "," Tsolar(t) "," Twind(t, "southeast")","
         Twind(t, "southwest" ) "," Twind(t, "north")"," coGen.l(t) ","
         gas.l(t) "," v.l(t)"," charge.l(t) "," discharge.l(t)/);
 put /;

putclose Out;
```

Calling external Files from within GAMS

- $Include *externalfilename*

  where *externalfilename* can include full path or just file name if located in the current working directory as declared in GAMS project.

The file to include consists of GAMS code.

- The ‘master’ GAMS file treats the statements in the include file as a continuation of the program code precisely at the $Include point.
- Other commands for including external files are: $Batinclude, $Libinclude, $Sysinclude
- $Oninclude and $Offinclude turn the echo print of the include file in the LST file on and off (default is ‘on’)

Example from grid model

PARAMETER inflow(h) hourly inflow of water m3 per s
$Include inflow

PARAMETER demand(h) hourly demand for electricity
$Include demand2003

PARAMETER wpower(h) hourly wind power (MW)
$Include wpowers

Data are found in inflow.gms, demand2003.gms and wpowers.gms

Comma Separated Value (CSV) Files

- Spreadsheets and other programs (e.g., Matlab) can read and write CSV files
  - Commas separate fields
  - Text items can be in quotes
- $ondelim – tells GAMS that entries which follow are in CSV format
- $offdelim – tells GAMS to ‘turn off’ use of CSV format

Examples using CSV format

```
SETS
    K  index k   /K1*K3/
    J  index j   /J1, J2, J3/;
TABLE CSVEX1(K,J) Data in csv Format
$ondelim
dummy, J1, J2, J3
K1, 16, 34, 12
K2, 3, 11, 5
K3, 52, 16, 4
$Offdelim
```

Note use of ‘dummy’ as the ‘placeholder’ in a table (see also below)

Alternatively an external file can contain the table:

```
SETS
   K  index k  /K1*K3/
   J  index j  /J1*J3/;
TABLE CSVEX2(K,J) data in csv format
   $ondelim
   $Include externalcsvdatafile
   $offdelim
```

The file *externalcsvdatafile.gms* contains:

dummy, J1, J2, J3
K1, 16, 34, 12
K2, 3, 11, 5
K3, 52, 16, 4

Note: ‘dummy’ is needed in the csv format to use up the space over the set elements defining the table rows – in the first row, first column.

Consider this example in GAMS and Excel

$$\text{Max} \quad Z = 3x_1 + 2x_2 + 0.5x_3$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 \leq 10$$

$$x_1 - x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

Three equations; x_3 is not constrained ('free'). The GAMS setup and solution are provided on the next slides.

GAMS Model

```
$Title McCarl & Spreen Chapter 3 Example

SETS
  i    index 1  /1*3/
  j    index 2  /1, 2/
;

PARAMETER b(j) rhs constraint values
  / 1    10
    2     3  /
;

TABLE a(j,i) Table of technical constraints

        1      2      3
  1     1      1      1
  2     1     -1      0
;
```

```
PARAMETER c(i) coefficients in obj
 /1    3
  2    2
  3   0.5/
;

VARIABLES
  z     objective value
  x(i)  activities
;

EQUATIONS
OBJ      Objective function
CON(j)   Constraint
;
```

```
* ---------------------------------------------------------
*    Construction of the actual LP model
* ---------------------------------------------------------

OBJ ..  z =E= sum(i, c(i)*x(i));

CON(j).. sum(i, a(j,i)*x(i)) =L= b(j);

MODEL McSpreen /all/;

x.lo('1')=0; x.lo('2')=0;

SOLVE McSpreen using LP maximizing z;
```

gamside: C:\Documents and Settings\kooten.UVIC\My Documents\Classes\Programming\GAMS\Programs\class_examples.g...

File Edit Search Windows Utilities Help

Solution to previous GAMS problem

SinkCapacity

McSreen_Chap3_example.gms | McSreen_Chap3_example.lst

- Compilation
- Equation Listing
- Equation
- Column Listing
- Column
- Model Statistics
- Solution Report
- SolEQU
- SolVAR

```
               S O L V E      S U M M A R Y

     MODEL   McSpreen            OBJECTIVE  z
     TYPE    LP                  DIRECTION  MAXIMIZE
     SOLVER  CPLEX               FROM LINE  48

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      3 UNBOUNDED
**** OBJECTIVE VALUE                26.5000

 RESOURCE USAGE, LIMIT          0.015      1000.000
 ITERATION COUNT, LIMIT         2          10000

GAMS/Cplex    Apr 21, 2006 WIN.CP.CP 22.2 031.034.041.VIS For Cplex 10.0
Cplex 10.0.1, GAMS Link 31

Presolve found the problem infeasible or unbounded.
Rerunning with presolve turned off.
Dual infeasible or unbounded. Switching to primal to aid diagnosis.
Model has an unbounded ray.

**** ERRORS/WARNINGS IN VARIABLE x(3)
     1 error(s): 1 Unbounded variable
```

121: 1

Start | C:\Documents and ... | gamside | Microsoft PowerPoin... | 60% | 4:07 PM

R-program calling GAMS

```
library(triangle)
set.seed(312) #Set the seed which is any integer to allow us to repeat results

# ------------------ Parameters set for current R simulation ------------------
iter <- 100  # number of random numbers to generate
crops <- 10 # number of activities or crops
drift <- 0.002 # monthly drift of price over growing season of 7 months
volatil <- rnorm(crops, 0.4, 0.01) # see Turvey for explanation
# ----------------------------------------------------------------------

# ----------------Model parameters for input into GAMS:-------------------------------
# File name to enter GAMS: "scalars.gms"
# "delta" is insurance loading factor =1 actuarially sound, =0.5 50% subsidy; "area" is total
# area of farm; "targetR" is target revenue; "states" is number of iterations in simulation

X<-c('delta=0.5;','area=1500;','targetR=50000;','states=100;')
dim(X)<-c(4,1)
write.table(X, file='scalars.gms', col.names=F, row.names=F, quote=FALSE, sep='')
# ----------------------------------------------------------------------
```

```
#Crop names
crop.name <- c('wheat', 'durum', 'oat', 'barley', 'flax', 'canola', 'pea', 'lentil', 'other', 'sf')

# prices $/bu except lentil & other in $/lb (from Sask Crop Plannning Guide 2013)
pr <- c(7.00, 7.35, 2.90, 5.40, 12.60, 12.00, 9.00, 0.22, 0.35, 57.97)

# Average yield and stan dev. bu/ac from Sask Ag. Crop District 6B Stat Can 2012
yld <- c(35.5, 36.6, 76.1, 44.4, 20.6, 25.2, 26.2, 1167.7, 548.3, 1.0)
styld <- c(8.2, 6.6, 12.4, 10.3, 4.7, 7.7, 8.7, 464.0, 323.0, 0.001)

# Crop insurance payment for 70% coverage level ($/ac) (Sask Crop Plannning Guide 2013)
InsPremium <- c(10.50, 9.94, 11.02, 11.81, 15.10, 22.24, 11.23, 22.40, 16.69, 0.17)
# Cost ($/acre) of planting and harvesting crops from Sask Crop Plannning Guide 2013)
ct <- c(214.62, 208.17, 205.64, 208.10, 198.28, 276.96, 219.81, 234.48, 195.2, 55.8)
cost <- ct + InsPremium

# Observed area in crops from 2012 Census ('000s acres) Actual for Crop District 6A
obser <- c(717.2, 93.8, 61.2, 130.9, 27.5, 664.6, 208.1, 199.1, 26.0, 120.0)
```

```
# Farm-level observed with farm of 1500 ac
obs <- c(480, 60, 40, 100, 20, 450, 125, 125, 20, 80)

Input <- data.frame(cbind(cost, pr, yld,obs))
row.names(input) <- crop.name
write.csv(input, file='input.csv')

price <- array(0, dim=c(iter, crops))
yield <- array(0, dim=c(iter, crops))
revenue <- array(0, dim=c(iter, crops))

for (i in 1:iter) {
      price[i,] <- pr*exp((drift - 0.5*volatil^2)*7/12 + volatil*rnorm(crops)*(7/12)^0.5)
      yield[i,] <- rnorm(crops, yld, styld)
      revenue[i,] <- price[i,]*yield[i,]   #gross revenue
}
Revenue <-
setNames(data.frame(revenue),
c('wheat','durum','oat','barley','flax','canola','pea','lentil','other','sf'))
```

```
# GAMS function
write.csv(revenue, 'revenue.csv')
cmd = paste("C:\\GAMS\\win64\\24.1\\gams.exe","FarmSimPMP.gms", "lo=2")
system(cmd)

# Read output
gr1<-read.csv('REV.csv', header=FALSE)
gr2<-read.csv('VAR.csv', header=FALSE)
gr3<-read.csv('GRIP.csv', header=FALSE)
gr4<-read.csv('SEMI.csv', header=FALSE)
new.name<-c('rev','stdev','wheat','durum','oat','barley','flax','canola','pea','lentil','other','sf')
sol <- setNames(data.frame(cbind(gr1,gr2,gr3,gr4)), c('RevBase','VarBase','GRIP','semi-var'))
row.names(sol)<-new.name
sol
```

gamside: U:\kooten\Documents\Students\TimBogle\BiLevel\ForestScience\FS_BiLevel.gpr - [U:\kooten\Documents\Agriculture\WholeFarmInsurance\Model\Canada

File Edit Search Windows Utilities Model Libraries Help

inventory

FarmSimPMP.gms | RPGTradeModel.gms

```
* Whole Farm Insurance Model
* Following ideas in papeers by Calum Turvey, Keithe Coble and others
* Copyright G. Cornelis van Kooten and Esther Broere

SETS
         crop     crops /wheat,durum,oat,barley,flax,canola,pea,lentil,other,sf/
         iter     iterations in Monte Carlo Simulation /1*100/

alias(crop, r);

SCALARS delta, area, targetR, states;
$include scalars.gms

TABLE input(crop, *) Input values created in R file
$ondelim
$include input.csv
$offdelim
;

* -------------- CONTENTS OF INPUT MATRIX CREATED IN R: -------------------
* cost - variable cost in $ per acre
* pr - mean price of the various crops in $ per tonne
* yld - average yield by crop in tonnes per acre
* obs - observed acres in each of the crops with total acres equal 1000

TABLE carlo(iter, crop) Monte Carlo revenue simulations
$ondelim
$include revenue.csv
$offdelim
;

* ----------- Observed net revenue per acre by crop for historic data
PARAMETERS xrev(crop) 'net return per acre by crop (average over random states)'
           n1         for variance
```

1: 1 | Insert

Start

gamside: U:\kooten\Documents\Students\TimBogle\BiLevel\ForestScience\FS_BiLevel.gpr - [U:\kooten\Documents\Agriculture\WholeFarmInsurance\Model\Canada

File Edit Search Windows Utilities Model Libraries Help

inventory

FarmSimPMP.gms | RPGTradeModel.gms

```
* ----------- Observed net revenue per acre by crop for historic data
PARAMETERS xrev(crop) 'net return per acre by crop (average over random states)'
           n1         for variance
           n2         for skewness
;

xrev(crop) = sum(iter, carlo(iter,crop))/states;
n1 = 1/states;
n2 = states/((states-1)*(states-2));

* ----------- CONSTRUCTION OF THE MODEL EQUATIONS
VARIABLES
          rev1       Objective for 1st stage of PMP
          rev2       Objective for 2nd stage of PMP
          V1         Variance value
          x(crop)    Optimal number of acres in each crop
          rev(iter)  Net revenue for each random state
          avrev      Average net revenue across all random states
          Skew       skewness value
          GR(iter)   Revenue for each random state w crop specific insurance
          Z(crop)    Insurance premium for each crop
;

POSITIVE VARIABLES  x ;

EQUATIONS
          variance       minimize variance of yield
          revenu(iter)   Net revenue for each iteration with optimal land use
          land           Total land use constraint
          average        Calculate average revenue across all random states
          obj1           Net revenue objective for 1st stage of PMP
          obj2           Net revenue objective for 2nd stage of PMP
          calib(crop)    Calibration constraints
```

1: 1 | Insert

Start

gamside: U:\kooten\Documents\Students\TimBogle\BiLevel\ForestScience\FS_BiLevel.gpr - [U:\kooten\Documents\Agriculture\WholeFarmInsurance\Model\Canada

File Edit Search Windows Utilities Model Libraries Help

inventory

FarmSimPMP.gms | RPGTradeModel.gms

```
          calib(crop)     Calibration constraints
;

variance.. V1 =E= n1*sum(iter, sqr(0.001*(rev(iter)-avrev)));

revenu(iter)..  rev(iter) =E= sum(crop, carlo(iter,crop)*x(crop));
average..       avrev =E= sum(iter, rev(iter))/states;

land..      sum(crop, x(crop)) =E= area;

obj1..      rev1 =E= sum(crop, x(crop)*(input(crop,'yld')*input(crop,'pr') -
                        input(crop,'cost')));

calib(crop)..   x(crop) =L= input(crop,'obs')+0.01;

MODEL pmp /obj1, land, calib/

OPTION QCP=CPLEX;

SOLVE pmp using LP maximizing rev1;

* Assume TC = m0 x + 0.5 m1 x^2
PARAMETERS m1(crop)  Slope parameter for MC curve
           m0(crop)  Intercept parameter for MC curve
;

m1(crop) = 2*calib.m(crop)/input(crop,'obs');
m0(crop) = input(crop,'cost') - 0.5*m1(crop)*input(crop,'obs');

EQUATIONS
       target         Revenue target
       GrossRev(iter) Gross revenue insurance w crop specific revenue protect
;
```

1: 1 | Insert

Start

gamside: U:\kooten\Documents\Students\TimBogle\BiLevel\ForestScience\FS_BiLevel.gpr - [U:\kooten\Documents\Agriculture\WholeFarmInsurance\Model\Canada

File Edit Search Windows Utilities Model Libraries Help

inventory

FarmSimPMP.gms | RPGTradeModel.gms

```
obj2..     rev2 =E= sum(crop, input(crop,'pr')*input(crop,'yld')*x(crop)
                  - m0(crop)*x(crop) - 0.5*m1(crop)*x(crop)*x(crop));

target..   rev2 =G= targetR;

GrossRev(iter).. GR(iter) =E= rev(iter)
                  + sum(crop, max((Z(crop)-rev(iter)),0)*x(crop))
                  - delta*sum(crop, max((Z(crop)-xrev(crop)),0)*x(crop));

MODEL baseRev /obj2, land/;

MODEL baseVar /variance, obj2, revenu, average, target, land/

MODEL GrossRevIns /variance, obj2, revenu, GrossRev, average, target, land/;

*OPTION QCP=MINOS5;
OPTION NLP=CONOPT;
*OPTION NLP=MINOS5;
OPTION DNLP=SNOPT;
*OPTION DNLP=CONOPT;
*OPTION DNLP=MINOS5;

* ----------Minimize Variance s.t. revenue target------------------------------
SOLVE baseRev using QCP maximizing rev2;
*
* Write solution
*
file RevOut /REV.csv/;
put RevOut;
  put rev2.l /
  put 0.0 /
*  put V1.l /
  loop (crop, put x.l(crop) /;
```

1: 1 Insert

Start

gamside: U:\kooten\Documents\Students\TimBogle\BiLevel\ForestScience\FS_BiLevel.gpr - [U:\kooten\Documents\Agriculture\WholeFarmInsurance\Model\Canada

File Edit Search Windows Utilities Model Libraries Help

inventory

FarmSimPMP.gms RPGTradeModel.gms

```
SOLVE baseRev using QCP maximizing rev2;
*
* Write solution
*
file RevOut /REV.csv/;
put RevOut;
  put rev2.l /
  put 0.0 /
*  put V1.l /
  loop (crop, put x.l(crop) /;
);
putclose RevOut;

SOLVE baseVar using NLP minimizing V1;
*
* Write solution
*
file VarOut /VAR.csv/;
put VarOut;
  put rev2.l /
  put V1.l /
  loop (crop, put x.l(crop) /;
);
putclose VarOut;
* ---------------------------------------------------------------------------

* ---------- Gross Revenue insurance minimizing variance ----------------------
SOLVE GrossRevIns using DNLP minimizing V1;
*
* Write solution
*
file GRIPout /GRIP.csv/;
put GRIPout;
  put rev2.l /
```

1: 1 Insert

IntroGAMS.pptx - Microsoft PowerPoint

Start

```
  put rev2.l /
  put V1.l /
  loop (crop, put x.l(crop) /;
);
putclose GRIPout;
* ---------------------------------------------------------------------------

* ---------------------------------------------------------------------------
* Semi-variance model:
* -------------- Base case as above -----------------------------------------
PARAMETERS
         standev   standard deviation
;

standev = sqrt(V1.l);

EQUATIONS
          skewness      maximize skewness
;
skewness.. Skew =E= n2*sum(iter, power(((rev(iter)-avrev))/standev,3));

MODEL semivariance /variance, obj2, skewness, revenu, average, target, land/;
SOLVE semivariance using NLP maximizing Skew;
*
* Write solution
*
file SkewOut /SEMI.csv/;
put SkewOut;
  put rev2.l /
  put V1.l /
  loop (crop, put x.l(crop) /;
);
putclose SkewOut;
* ---------------------------------------------------------------------------
```