

# Introduction to GAMS

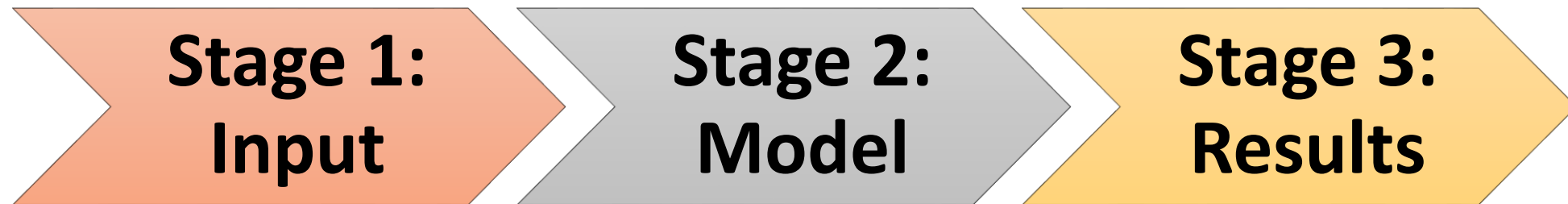
Imane Helmy

German University in Cairo

# What is GAMS?

The GAMS (General Algebraic Modeling System) software was developed to allow us to solve simultaneous equation systems, with or without the optimization of some objective functions.

# Overall Structure of GAMS Model: A quick comparison with STATA



**Stata**

Data (.dta)

Do-files (.do)

Log files (.log)

**Project  
Directory (cd)**

**GAMS**

Data

gms (.gms)  
Include (.inc)

Listing (.lst)  
GDx (.gdx)

**Project File  
(.gpr)  
Directory to  
include all files**

# The basic components of a GAMS model

Type	Component
<b>Stage 1: Inputs</b>	<b>Sets:</b> Declaration and Assignment of members
	<b>Parameters</b> Declaration and Assignment of values
	<b>Data</b> Assignment
<b>Stage 2: Model</b>	<b>Variables</b> Declaration and Assignment of type
	<b>Equations</b> Declaration and Definition
	<b>Model and Solve Statements</b>
	<b>Display Statements</b> (optional)
<b>Stage 3: Outputs</b>	<b>Echo Prints</b>
	<b>Reference Maps</b>
	<b>Equation Listings</b>
	<b>Status Reports</b>
	<b>Solution Reports</b>

# Overall Structure of GAMS Model: Stage 1-Input

**SAM Data.**

**SETS** declaration and definition.

**PARAMETERS** declaration and definition.

**Data** assignment.

# Overall Structure of GAMS Model: Stage 1-Input

- **SAM Data:** Explore your data- Egypt 2010/11 SAM

	1	2	3	4	5	6	7	8	9	10
	Activities	Commodities	Factors of production	Enterprises	Households	Government	Taxes	Savings or Investment	Rest of the world	Total
1 Activities		2,281,021								2,281,021
2 Commodities	941,156				1,005,613	131,004		246,449	282,223	2,606,446
3 Factors of production	1,339,865								1,869	1,341,734
4 Enterprises			552,319			97,039			524	649,881
5 Households			750,275	290,825		6,592			83,763	1,131,455
6 Government				84,843					1,316	147,915
7 Taxes		-37,290		82,911	16,134					61,755
8 Savings or Investment				191,254	107,995	-86,921			34,121	246,449
9 Rest of the world		362,715	39,140	47	1,714	201				403,816
10 Total	2,281,021	2,606,446	1,341,734	649,881	1,131,455	147,915	61,755	246,449	403,816	

Reference: Adapted Al-Riffai, P. et al. (2016). A disaggregated social accounting matrix: 2010/11 for policy analysis in Egypt. Egypt SSP Working Paper 2. Washington, DC and Cairo, Egypt: International Food Policy Research Institute

# Overall Structure of GAMS Model: Stage 1-Input

- **SETS** declaration and definition:

$S = \{commodities, activities, factors\}$

## Syntax:

Sets

Declare

```
a "activities" / manuf, agri, services/  
f "factors" / labor, capital, land/ ;
```

Define

## OR:

Sets

```
a   activities/  
M   manuf  
Agr Agriculture/ ;
```

- Set[s] is the keyword that indicates that this is a set statement.
- Set\_name is the internal name of the set. Optional **explanatory text** may be used.
- The list of set elements is delimited by forward slashes. Each element in a set must be separated from other elements by a comma or by an end-of-line, and each element is separated from any associated text by a blank.

# Overall Structure of GAMS Model: Stage 1-Input

- **SETS** declaration and definition:

Superset

```
Set i "all sectors" / light-ind, food+agr, heavy-ind, services /  
t(i) "traded sectors" / light-ind, food+agr, heavy-ind /  
Nt(i) "non-traded sectors" / services /;
```

**t(i) is a subset:** each member of the set *t* must also be a member of the superset *i*

**Domain checking:** order of declaration and definition is important: the membership of *i* must be known before *t* is defined



# Overall Structure of GAMS Model: Stage 1-Input

- **PARAMETERS** declaration and definition:

-Parameters are the elements in the equations that will not change after a simulation such as elasticity, share coefficients...etc.

**Syntax:**

```
Set j "markets" / mexico-df, guadalaja /;
```

```
Parameter dd(j) "distribution of demand" / mexico-df 55, guadalaja 15 /;
```

-Parameters are often defined by their value at the base year (SAM).

-A common way to define these variables is to add an "o" after the variable name

```
Yo Production base year data
```

# Overall Structure of GAMS Model: Stage 1-Input

- **PARAMETERS** declaration and definition:
  - Zero is the default value for all parameters: You only need to include the nonzero entries.
  - A scalar is regarded as a parameter that has no domain
- **Syntax:** Scalar  $f$  “freight in dollars per case per thousand miles” /90/;
- We can declare parameter  $d$  and specify its domain by tables ( plant  $i$ , market  $j$ ).

**Syntax:**

```
Table d(i,j) distance in thousands of miles
      new-york    chicago    topeka
seattle      2.5      1.7      1.8
san-diego    2.5      1.8      1.4 ;
```

# Overall Structure of GAMS Model: Stage 1-Input

- **Data Entry:**

```
TABLE SAM(*,*) Social accounting matrix
```

	LD	KD	SAL	CAP	F	AGR	MAN	SER	ACC
LD						300	100	200	
KD						100	150	100	
SAL	600								
CAP		210			70				
F		140							
AGR			162	21		50	150	90	27
MAN			108	84		20	150	90	173
SER			270	105		30	75	120	
ACC			60	70	70				
TOT	600	350	600	280	140	500	625	600	200

- **Data Assignment:**

```
DIVO = SAM('CAP', 'F');
ITO = SAM('TOT', 'ACC');
SFO = SAM('ACC', 'F');
SHO(h) = SAM('ACC', h);
YFO = SAM('TOT', 'F');
YHO(h) = SAM('TOT', h);
XSO(j) = SAM('TOT', j);
CO(i, h) = SAM(i, h);
DIO(i, j) = SAM(i, j);
INVO(i) = SAM(i, 'ACC');
KDO(j) = SAM('KD', j);
LDO(j) = SAM('LD', j);
```

Reference: Adapted from Robichaud, V. (2017). Introduction to GAMS. Partnership for Economic Policy (PEP)

**Value of dividends (DIVO) is given in table SAM at the intersection of row 'CAP' and column 'F'.**

# Overall Structure of GAMS Model: Stage 1-Input

- From Excel to GAMS (GDX):

## **Syntax:**

```
$call.gdxrw inputFile {outputFile} {options} [symbols]
```

```
$call " .gdxrw i=stg1_data.xls o=data_in.gdx index=Layout!A4"
```

# General Rules

- Declare then use.
- A good practice is to end commands with a semicolon (;) to avoid unexpected compilation errors.
- Volume vs. values.
- GAMS is NOT case sensitive (Capital and small letters are not distinguished).
- Main mathematical functions are described below:

Multiplication	*	Equality in an operation	=	Logarithm	LOG(.)
Subtraction	-	Summation	SUM(set domain, element)	Maximum	MAX(.,.)
Addition	+	Product	PROD(set domain, element)	Minimum	MIN(.,.)
Division	/	Absolute Value	ABS(.)		
Exponent	**	Exponential	EXP(.)		

Reference: Adapted from Robichaud, V. (2017). Introduction to GAMS. Partnership for Economic Policy (PEP)

# SPOT

## THE ERROR

Set

$I(f)$  "labour factors" / skilled labour, unskilled labour /  
 $f$  "factors" / labor, capital, land/

-The membership of  $f$  must be known before  $i$  is defined.

- A semi-colon is missing.

Set

$f$  "factors" / labor, capital, land/  
 $I(f)$  "labour factors" / skilled labour, unskilled labour /;

# The basic components of a GAMS model

Type	Component
<b>Stage 1: Inputs</b>	<b>Sets:</b> Declaration and Assignment of members
	<b>Parameters</b> Declaration and Assignment of values
	<b>Data</b> Assignment
<b>Stage 2: Model</b>	<b>Variables</b> Declaration and Assignment of type
	<b>Equations</b> Declaration and Definition
	<b>Model and Solve Statements</b>
	<b>Display Statements</b> (optional)
<b>Stage 3: Outputs</b>	<b>Echo Prints</b>
	<b>Reference Maps</b>
	<b>Equation Listings</b>
	<b>Status Reports</b>
	<b>Solution Reports</b>

# Overall Structure of GAMS Model: Stage 2-Model

**VARIABLES** declaration.

**EQUATIONS** declaration and definition.

**Calibration and Closures.**

**MODEL** solve



# Overall Structure of GAMS Model: Stage 2-Model

- Variables declaration:

**Syntax:** VARIABLES

P(i) Price of commodity i;

Positive Variables u, v, e;

Keyword	Default Lower Bound	Default Upper Bound
free (default)	-inf	+inf
positive or nonnegative	0	+inf
negative	-inf	0
binary	0	1
integer	0	+inf

# Overall Structure of GAMS Model: Stage 2-Model

- **Equations declaration:**

**Syntax:** EQUATIONS

UTILITY objective function

BUDCONS budget constraint

;

- **Equations definition:**

UTILITY.. U =E= PROD(C, Q(C)\*\*beta(C));

BUDCONS.. y =E= SUM(C, p(C)\*Q(C));

# Overall Structure of GAMS Model: Stage 2-Model

- **Equations definition:**

**Syntax:** UTILITY.. U =E= PROD(C, Q(C)\*\*beta(C));  $\longrightarrow$   $U = \prod_{c \in C} q_c^{\beta_c}$   
BUDCONS.. y =E= SUM(C, p(C)\*Q(C));  $\longrightarrow$   $\sum_{c \in C} p_c q_c = y$

## Syntax for defining equations:

1. Name is stated followed by two dots.
2. Equation is defined, ending with a semicolon.
3. The =E= between the left hand side and the right hand side of the equation means "equals to" as opposed to =G= (greater than) or =L= (less than).
4. Each equation can only be defined once but the variables can be used in several equations and on both sides of the same equation.
5. The order in which the equations are defined does not matter as all equations are solved simultaneously.

# Overall Structure of GAMS Model: Stage 2-Model

- **Calibration:**

## **Syntax:**

`w0 =1;`

`LO(I) =LO(I)/w0;`

# Overall Structure of GAMS Model: Stage 2-Model

- **Variable Initialization:**

- Assign values to variables declared using the suffix **.L** (for level)-initial value:

**Syntax:**

```
YH.L(h)=YHO(h);
```

- For fixing the exogenous variables, the suffix **.FX** is used

**Syntax:**

- Capital is sector-specific:

```
KS.FX=KSO;
```

# Overall Structure of GAMS Model: Stage 2-Model

- **Closures:**

- Researchers select closures based on:

1. Characteristics of the economy.
2. Literature and previous studies on the country.
3. Research Objectives.

**Sensitivity Analysis: changing closures might affect your results!**

# Overall Structure of GAMS Model: Stage 2-Model

- **Solve Model**

1. Defines the model by the **MODEL** statement, followed by a name and an optional brief description then the list of equations to be resolved between slashes, and ends with a semicolon.

When all equations are to be solved, use the keyword **ALL** instead of the complete list of equations.

2. The second line defines the procedure to be used to resolve the model. The **SOLVE** statement is followed by the name of the model, **USING** and the procedure to be used to resolve the model. The procedure to be used is determined by the type of model to be solved.

e.g. linear programs can be solved using the **LP** procedure, while non-linear programs can be solved using **NLP** or **CNS** (Constrained non-linear system).

```
MODEL UTILMAX /ALL/;
```

```
SOLVE UTILMAX MAXIMIZING U USING NLP;
```

# Overall Structure of GAMS Model: Stage 2-Model

- **Execute the Model:**

- GAMS code is written and saved with the extension **.gms**.

- Include Files: **Syntax:** `$INCLUDE stg1_expt.inc`

- To run the code: choose the run command in the file menu OR press F9 OR click on the red arrow button.





# SPOT THE ERROR

## Equations

Income.  $w*LS+r*Ks=Inc;$

-Declare then define.

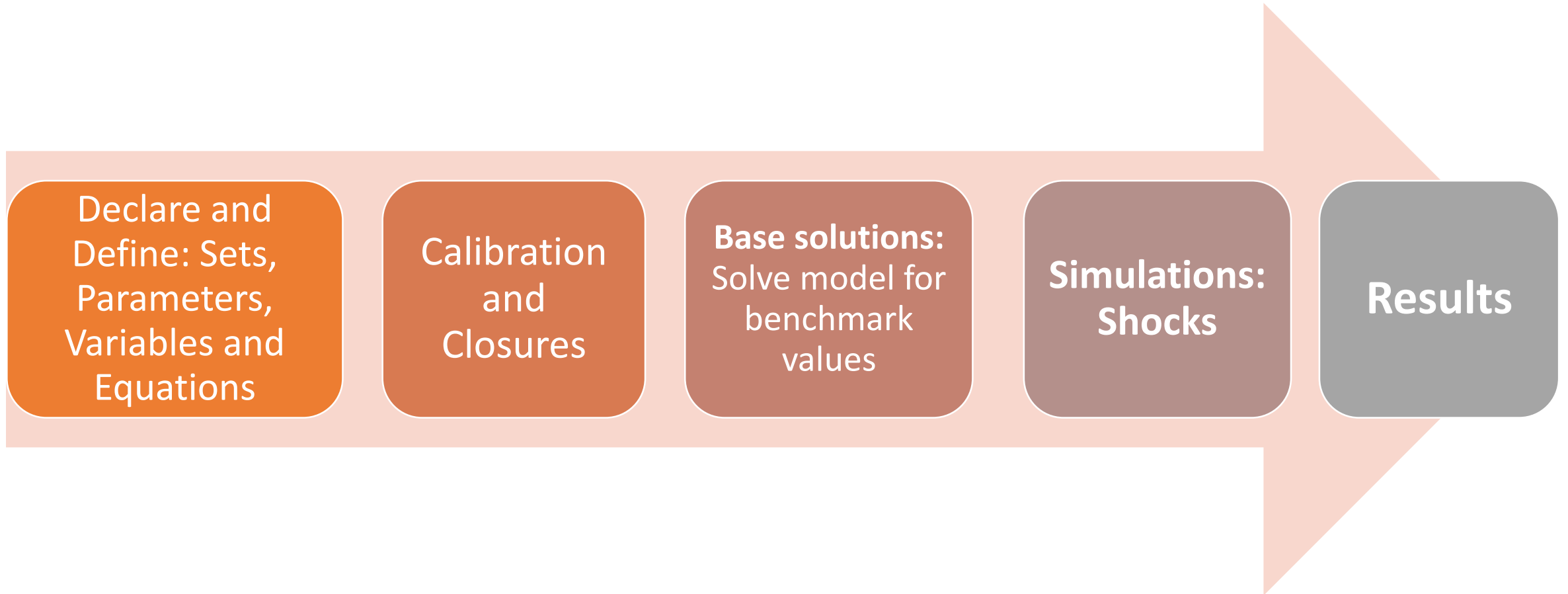
-A missing dot can make a difference!

-Relational Operator: =E=

## Equations

Income  
Income..  $w*LS+r*Ks=E=Inc;$

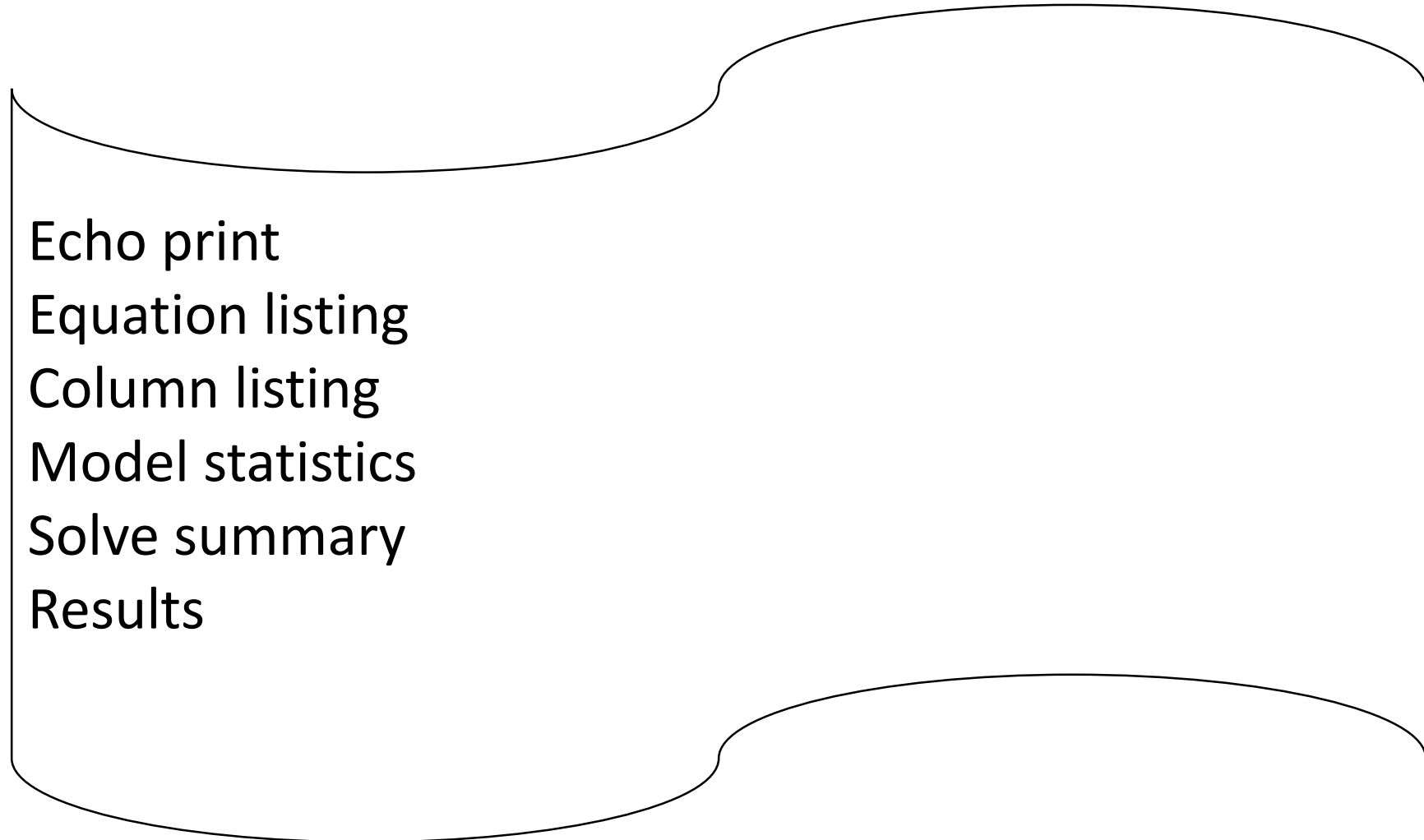
# A quick Recap...



# The basic components of a GAMS model

Type	Component
<b>Stage 1: Inputs</b>	<b>Sets:</b> Declaration and Assignment of members
	<b>Parameters</b> Declaration and Assignment of values
	<b>Data</b> Assignment
<b>Stage 2: Model</b>	<b>Variables</b> Declaration and Assignment of type
	<b>Equations</b> Declaration and Definition
	<b>Model and Solve Statements</b>
	<b>Display Statements</b> (optional)
<b>Stage 3: Outputs</b>	<b>Echo Prints</b>
	<b>Reference Maps</b>
	<b>Equation Listings</b>
	<b>Status Reports</b>
	<b>Solution Reports</b>

# Overall Structure of GAMS Model: Stage 3-Results



# Overall Structure of GAMS Model: Stage 3-Results

- The output file created has the extension **.lst**.
  - Echo print of the initial program**: reproduction of the model with line numbers on the left hand side for further reference in the file
  - Equation listing**
  - Column listing**
  - Model statistics**
  - Solve summary**
  - Results**

# Overall Structure of GAMS Model: Stage 3-Results

- **Equations Listing:**

-At the end of the equation, the value of the left hand side is written between parenthesis (LHS= " ").

-When the value of the left hand side differs from the one on the right hand side, GAMS add four asterisks (\*\*\*\*).

```
----- XSEQ  =E=  Value added demand in industry j (Leontief)
XSEQ(AGR) ..  VA(AGR) - 0.8*XS(AGR) =E= 0 ; (LHS = 0)
XSEQ(MAN) ..  VA(MAN) - 0.4*XS(MAN) =E= 0 ; (LHS = 0)
XSEQ(SER) ..  VA(SER) - 0.5*XS(SER) =E= 0 ; (LHS = 0)
```

# Overall Structure of GAMS Model: Stage 3-Results

```
MODEL STATISTICS

BLOCKS OF EQUATIONS      22      SINGLE EQUATIONS      58
BLOCKS OF VARIABLES     21      SINGLE VARIABLES      58
NON ZERO ELEMENTS      193      NON LINEAR N-Z      92
DERIVATIVE POOL        20      CONSTANT POOL        24
CODE LENGTH            195
FIXED EQUATIONS        58      FREE VARIABLES        58

GENERATION TIME      =  0.000E+0000 SECONDS      4 MB  24.6.1 r55820 WEX-WEI
```

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

MODEL      AUTA
TYPE      CNS
SOLVER     CONOPT          FROM LINE  334

**** SOLVER STATUS      1 Normal Completion
**** MODEL STATUS      16 Solved
```

# Overall Structure of GAMS Model: Stage 3-Results

- **Solution Listing:**

For each equation and each variable, GAMS indicates the lower and upper bounds and the level

```
---- VAR C Consumption of commodity i by type h households
```

	LOWER	LEVEL	UPPER
AGR.SAL	-INF	162.000	+INF
AGR.CAP	-INF	21.000	+INF
MAN.SAL	-INF	108.000	+INF
MAN.CAP	-INF	84.000	+INF
SER.SAL	-INF	270.000	+INF
SER.CAP	-INF	105.000	+INF

Reference: Adapted from Robichaud, V. (2017). Introduction to GAMS.  
Partnership for Economic Policy (PEP)





# Errors

- **Compilation errors:** syntax or coding errors caused by simple mistakes
  - The explanatory error message text will help diagnose the problem and correct it.
  - GAMS will put four asterisks (\*\*\*\*) below the line where a mistake has been detected followed by a dollar sign (\$) and a number corresponding to a specific error.

```
136   alpha(i)           = WO*LDO(i) / {PVAO(i) *VAO(i);  
****                                     $8
```

```
306   SOLVE AUTA USING CNS;  
****                               $257
```

## Error Messages

```
8   ')' expected  
257   Solve statement not checked because of previous errors
```



# Errors

## Common Cause of Compilation Error

Closing parentheses, or brackets are missing.

The two dots .. are missing in the equation definition.

The equation type (eg. =L=) is missing in the body of the equation.

The equation has been declared, but not defined.

A statement followed by another statement is not terminated with ;.

GAMS cannot find a set with this name. Check for typos in the set name and set elements that are referenced without quotes.

The referenced set element cannot be found” Check for typos, omissions in the set declaration...etc

A domain error. The wrong set is referenced for the respective index position.

The name used here was already used for another identifier.

Something is wrong with the model specification. This is often a consequential error of another error.

Surplus closing parentheses, square brackets or braces.



# Errors

- **Execution errors**

- Usually caused by illegal arithmetic operations such as division by zero or taking the log of a negative number.
- Reasons include: wrong data or equations, calibration issues.

**One error can generate many consequence error messages so concentrate on fixing the first error and run the model!**

# Additional Options

- The text following the **\$TITLE** command will appear as a header on every page of the output listing. A sub-title can also be added to this header using the command **\$STITLE**.
- GAMS will not read lines beginning with an asterisk (\*).
- The dollar control option **\$ontext** marks the beginning of the comment block and the option **\$offtext** marks the end.

```
$TITLE      MODEL AUTA
$STITLE     AUTARKY WITHOUT GOVERNMENT

* Model of a closed economy without government producing 3 goods using
* 2 factors owned by 2 types of households.
```

# Example: Transportation Problem

```
$title A Transportation Problem (TRANSPORT,SEQ=1)
```

```
$onText
```

```
This problem finds a least cost shipping schedule that meets  
requirements at markets and supplies at factories.
```

```
Dantzig, G B, Chapter 3.3. In Linear Programming and Extensions.  
Princeton University Press, Princeton, New Jersey, 1963.
```

```
This formulation is described in detail in:  
Rosenthal, R E, Chapter 2: A GAMS Tutorial. In GAMS: A User's Guide.  
The Scientific Press, Redwood City, California, 1988.
```

```
The line numbers will not match those in the book because of these  
comments.
```

```
Keywords: linear programming, transportation problem, scheduling  
$offText
```

```
Set
```

```
  i 'canning plants' / seattle, san-diego /  
  j 'markets'         / new-york, chicago, topeka /;
```

```
Parameter
```

```
  a(i) 'capacity of plant i in cases'  
      / seattle    350  
      / san-diego  600 /
```

```
  b(j) 'demand at market j in cases'  
      / new-york   325  
      / chicago    300  
      / topeka     275 /;
```

```
Table d(i,j) 'distance in thousands of miles'
```

	new-york	chicago	topeka
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4;

```
Scalar f 'freight in dollars per case per thousand miles' / 90 /;
```

# Example: Transportation Problem

```
Table d(i,j) 'distance in thousands of miles'
      new-york  chicago  topeka
seattle      2.5      1.7      1.8
san-diego    2.5      1.8      1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;

Variable
  x(i,j) 'shipment quantities in cases'
  z      'total transportation costs in thousands of dollars';

Positive Variable x;

Equation
  cost      'define objective function'
  supply(i) 'observe supply limit at plant i'
  demand(j) 'satisfy demand at market j';

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;
```

# GAMS Installation

- Download Demo/Student version (sufficient for small models) :  
<http://www.gams.com/download/>
- Installation Notes for Mac OS X:  
[https://www.gams.com/25.1/docs/UG\\_MAC\\_INSTALL.html](https://www.gams.com/25.1/docs/UG_MAC_INSTALL.html)
- Installation Notes for Windows:  
[https://www.gams.com/25.1/docs/UG\\_WIN\\_INSTALL.html](https://www.gams.com/25.1/docs/UG_WIN_INSTALL.html)

# References

- Robichaud, V. (2017). Introduction to GAMS. Partnership for Economic Policy (PEP).
- GAMS Development Corporation. GAMS Documentation. GAMS Development Corporation: Washington, D.C.
- McCarl, B., Meeraus, A., Eijk, P. van der, Bussieck, M., Dirkse, S., & Pete, S. *McCarl GAMS User Guide*. GAMS Development Corporation: Washington, D.C.
- Rosenthal, R. (2007). *GAMS — A User's Guide*. GAMS Development Corporation: Washington, D.C.



Thanks for your attention