# Dynamic Structural Models for Policy Evaluation

**Sergi Quintana**

Universitat Autònoma de Barcelona

Barcelona School of Economics

**BSE Summer School, 2024**

#bsesummer

## Session 0: Introduction to Mata

BSE Summer School, 2024

## Session Outline

## What is Mata?

Mata is a full-blown programming language that compiles what you type into bytecode, optimizes it, and executes it fast.

Behind the scenes, many of Stata's commands are implemented using Mata.

Some usefull resources:

- User's Manual
- Other resources: Mata Introduction , External Notes

## Where to get help

Similar than in Stata, we can search for help while using Mata:

- **help mata command**: will look for the documentation of the specific command.
- **search mata word**: will look for the exact command that contains that word.
- **view search mata word**: will display the result in the viewer.

## From Stata to Mata

From Stata to Mata:

- **st_data()**: loads a copy of the current Stata dataset.
- **st_view()**: makes a matrix that is a view of the current Stata dataset. As oposed to **st_data()** it does not copy the information, so it is more efficient.

From Mata to Stata:

- **st_addvar()**: adds a variable to Stata dataset.
- **st_store()**: asign values to existing Stata variable.

## Matrix Operations

- Declare a matrix: **A = (1,2 \ 3,4)**.
- Declare an empty matrix: **B = J(2,3,.)**
- Index a matrix. **A[1,1] = 4**
- Transpose of a matrix: **A'**.
- Sum two matrices : **A+B**.
- Multiply: **A*B**.
- Element-wise multiplication: **A:*B**

Barcelona School of Economics SUMMER SCHOOL

#bsesummer

## Exercices

**Exercice 1.**

Using **webuse auto, clear**:

- Move the variable price from Stata to Mata.
- Compute price squared.
- Replace the 4rth element of price squared with the value 10000.
- Move this variable back to Stata as a new variable.

## Exercices

**Exercice 2.**

Create in Mata:

- a scalar a that contains the number 2, a matrix A with 4 rows and 4 columns and all the elements being 2 and a matrix B of the same dimension, which has the value 2 on the diagonal and 0 otherwise.

Do the following:

- Multiply the scalar a with matrix A and store the result in matrix C. Calculate the sum of matrix C and the inverse of matrix B and store it in matrix D. Combine matrices A, B, C and D into one 8x8 matrix E. Create a vector b that contains the first, third and fifth element of the first row of E.

## Exercices

- Add 20 rows to matrix A with the following values: In column 1, start with the value 1 in the first new row and increase in increments of 1 to the value 20, the second column starts with the value 2 and increases in increments of 1 to the value 21, the third column starts with the value 3 and increases in increments of 1 to the value 22 and the fourth column starts with the value 4 and increases in increments of 1 to the value 23.

## Loops

The different types of loops are:

- **while**. Executes a section of code as long as an expression is true.
- **for**. Executes a section of code for several iterations.
- **do**. Executes a section of code one or more times, until an expression is not satisfied. It can be seen as a **while** loop that will at least perform an iteration.

## Logical Conditions

The statements are:

- **if**: Evaluates an expression and executes a section of code whenever it is true.
- **else if**: Introduces another condition to be evaluated after the **if**.
- **else**: A section of the code that will be executed if the other logical conditions are false.

## Functions

Mata allows to create new functions.

- Functions can take arguments.
- Functions can return outputs.
- Functions can take declarations.
    - Element type: transmorphic, numeric, real, complex, string, pointer.
    - Organizational type: matrix, vector, rowvector, colvector, scalar

## Exercices

**Exercice Loop**

- Using a loop, generate a 10x20 matrix where if the sum of the column index and the row index is less than ten, the entry is the squared sum of the sum of the index of the column and row, and otherwise is 0.
- Generate a 10x15 matrix and as long as there are more rows than columns, sum the last two columns and drop the last column.

## Exercices

**Exercice Functions**

Define a function that returns a vector of n independent random variables drawn from the Bernoulli distribution given success probability theta. **Hint:** You might want to use the function **uniform()**.

## Exercices

- Using the **PracticalSession0_data.dta**, find the OLS estimate using mata of a regression of log wage on age, age2, exper, exper2, educ, female.
- Find the standard errors.
- Find the t-statistic and the p-value. **Hint:** Use the function **ttail()**.
- Write a function that displays all this values for any regression.

## Optimization

Mata comes with a built-in optimizer **optimize_init()**.

- **S = optimize_init()** defines the problem.
- Now we can modify the default options of the problem, for example **optimize_init_which(S, "max")** defines a maximization problem.
- **optimize_init_evaluator(S, func)** defines the function to optimize.
- **optimize_init_params(S,1)** sets the initial value.
- **xmax = optimize(S)** returns the maximum.