Report to Users

Alan Riley

Vice President, Software Development
StataCorp LP

2006 German Stata Users Group meeting, Mannheim, Germany
Outline

1. Stata Press

2. Stata 9

3. New development
   - Stata 9.1
   - Stata 9.2 - Mata structures
   - Stata 9.2 - work faster
Most active year ever

- Stata Journal indexed
- Two revised editions of existing books
- Four new books published
- Seven books in progress
Stata Journal

- 6th year of publication
- Special edition - Stata 20th anniversary
- Now indexed

Thomson Scientific citation indexes

- Science Citation Index Expanded
- CompuMath Citation index
Stata Journal

- 6th year of publication
- Special edition - Stata 20th anniversary
- Now indexed

Thomson Scientific citation indexes

- Science Citation Index Expanded
- CompuMath Citation index
More than doubled number of books published

Revised editions, 2005

- Regression Models for Categorical Dependent Variables Using Stata, 2nd Edition
  by J. Scott Long, Jeremy Freese

- Maximum Likelihood Estimation with Stata, 3rd Edition
  by William Gould, Jeffrey Pitblado, William Sribney
More than doubled number of books published

Revised editions, 2005

- Regression Models for Categorical Dependent Variables Using Stata, 2nd Edition
  by J. Scott Long, Jeremy Freese

- Maximum Likelihood Estimation with Stata, 3rd Edition
  by William Gould, Jeffrey Pitblado, William Sribney
More than doubled number of books published

New books, 2005

- **Data Analysis With Stata**
  by Ulrich Kohler and Frauke Kreuter

- **Multilevel and Longitudinal Modeling Using Stata**
  by Sophia Rabe-Hesketh and Anders Skrondal

- **A Gentle Introduction to Stata**
  by Alan Acock

- **An Introduction to Stata for Health Researchers**
  by Svend Juul
Forthcoming books, 2006

- **An Introduction to Modern Econometrics Using Stata**  
  by Christopher F. Baum

- **Generalized Linear Models and Extensions, 2nd Edition**  
  by James Hardin, Joseph Hilbe

- **A Guide to Stochastic Frontier Models: Specification and Estimation**  
  by Subal Kumbhakar, Hung-Jen Wang

- **An Introduction to Forecasting Time Series Using Stata**  
  by Robert Yaffee

- **The 123s of Survey Statistics with Stata**  
  by Nicholas Winter

- **Applied Microeconometrics Using Stata**  
  by A. Colin Cameron, Pravin K. Trivedi
Forthcoming books, 2007

- **Data Management Using Stata**
  by Michael Mitchell
- Released April 2005
- 20th anniversary
- Largest release ever
Stata 1, January 1985
- 44 commands
- 175 pages of documentation

Stata 8, January 2003
- over 600 commands
- 4652 pages of documentation

Stata 9, April 2005
- over 700 commands including new matrix language Mata
- 6413 pages of documentation
## Stata 9

### Stata 1, January 1985
- 44 commands
- 175 pages of documentation

### Stata 8, January 2003
- over 600 commands
- 4652 pages of documentation

### Stata 9, April 2005
- over 700 commands including new matrix language Mata
- 6413 pages of documentation
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Commands</th>
<th>Pages of Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stata 1, January 1985</td>
<td>44 commands</td>
<td>175 pages</td>
<td></td>
</tr>
<tr>
<td>Stata 8, January 2003</td>
<td>over 600 commands</td>
<td>4652 pages</td>
<td></td>
</tr>
<tr>
<td>Stata 9, April 2005</td>
<td>over 700 commands including new matrix language Mata</td>
<td>6413 pages</td>
<td></td>
</tr>
</tbody>
</table>
Ongoing development

- Continued release-as-we-go strategy
- Stata 9.1
- Stata 9.2
  - Mata structures
  - Work faster
Multiple log files
Faster survey linearization
More stored estimation results
New Mata functions (permutation, string, regular expression, binary I/O)
Sized PNG and TIFF exported graphs
adoupdate
And more...
Mata structures

Set of variables tied together under a single name

```
struct structname {
    declaration(s)
}
```

Example

```
struct mystruct {
    real scalar n1, n2
    real matrix x
}
```
Mata structures

Set of variables tied together under a single name

```
struct structname {
    declaration(s)
}
```

Example

```
struct mystruct {
    real scalar n1, n2
    real matrix x
}
```
Mata structures

struct myresult {
    real scalar   yoverx
    real scalar   xovery
}

struct myresult scalar myfunc(real scalar x, real scalar y)
{
    struct myresult scalar   res

    res.yoverx = y/x
    res.xovery = x/y

    return(res)
}

...  
struct myresult scalar results  
...

results = myfunc(3, 4)
You can have vectors and matrices of structures

```
struct mystruct scalar      t
struct mystruct vector     t
struct mystruct rowvector  t
struct mystruct colvector  t
struct mystruct matrix     t
```

```
t[2,3].n1
```

Structures can contain vectors and matrices

```
t[2,3].x[9,2]
```
You can have vectors and matrices of structures

```
struct mystruct scalar    t
struct mystruct vector   t
struct mystruct rowvector t
struct mystruct colvector t
struct mystruct matrix   t
```

t[2,3].n1

Structures can contain vectors and matrices

```
t[2,3].x[9,2]
```
Structures can contain other structures

```mata
struct myresult {
    real scalar yoverx
    real scalar xovery
}

struct someresults {
    struct myresult scalar res1, res2
}

... 
struct someresults scalar myres
...

myres.res1 = myfunc(3, 4)
myres.res2 = myfunc(5, 6)
```
Mata structures

Advantages of structures

- Organization
- Convenience (return multiple results)
- Abstraction (handles)
Moore’s Law

- Computer processing power doubles every 18 months
- Max transistors per chip has doubled every 24 months
- To maintain, industry must improve at rate of 1% per week
Work faster – work in parallel

- new ‘flavor’ of Stata capable of performing symmetric multiprocessing (SMP)
- same capabilities as Stata/SE, but faster due to parallelization of central routines
- for dual core, multicore, or multiprocessor computers

Difference between ‘processor’ and ‘core’

- processor: central processing unit, or CPU
- core: computation engine of a CPU with integer and floating point processing units
Stata/MP

Work faster – work in parallel

- new ‘flavor’ of Stata capable of performing symmetric multiprocessing (SMP)
- same capabilities as Stata/SE, but faster due to parallelization of central routines
- for dual core, multicore, or multiprocessor computers
- http://www.stata.com/statamp/

Difference between ‘processor’ and ‘core’

- processor: central processing unit, or CPU
- core: computation engine of a CPU with integer and floating point processing units
Design requirements

- 100% compatible with Stata/SE, Intercooled Stata, and Small Stata
- No end-user programming necessary to obtain speed ups
- No changes necessary to do-files, user-written programs, or datasets
- Priority given to estimation commands
Supports 2 to 32 processors or cores on

- Macintosh OSX (Intel)
- 32-bit Windows
- 64-bit Windows (x86-64)
- 64-bit Windows (Itanium)
- 32-bit Linux
- 64-bit Linux (x86-64)
- 64-bit Linux (Itanium)
- 64-bit Solaris (Sparc)
**Perfection, in theory**

- 100% efficiency is twice as fast on 2 processors/cores
- Speed doubles for every doubling of number of processors
- Execution time halves for every doubling of number of processors

**Amdahl’s Law**

F: sequential/non-parallelizable fraction
N: number of processors
Maximum speed up: \[
\frac{1}{F + \frac{1-F}{N}}
\]
Perfection, in theory

- 100% efficiency is twice as fast on 2 processors/cores
- Speed doubles for every doubling of number of processors
- Execution time halves for every doubling of number of processors

Amdahl’s Law

\[ F: \text{sequential/non-parallelizable fraction} \]
\[ N: \text{number of processors} \]
Maximun speed up: \[ \frac{1}{F + \frac{1-F}{N}} \]
How much faster?

- Median speed up (overall)
  - 72% efficiency
  - 2 CPUs: 1.4
  - 3 CPUs: 1.75
  - 4 CPUs: 2.0

- Median speed up (estimation commands)
  - 88% efficiency
  - 2 CPUs: 1.7
  - 3 CPUs: 2.3
  - 4 CPUs: 2.8
How much faster?

- Median speed up (overall)
  - 72% efficiency
  - 2 CPUs: 1.4
  - 3 CPUs: 1.75
  - 4 CPUs: 2.0

- Median speed up (estimation commands)
  - 88% efficiency
  - 2 CPUs: 1.7
  - 3 CPUs: 2.3
  - 4 CPUs: 2.8
Observed performance region over all commands

Observed upper bound

Median performance (estimation)

Median performance (all commands)

Observed lower bound

Speed compared to single processor

Number of processors
Comments on median results

- half of commands run faster
- some even faster than theory due to cache effects
- half of commands run slower
- some not sped up at all
  - inherently sequential/impossible to parallelize (time series)
  - no effort made to parallelize (graph, xtmixed)
Methods

- Open/MP API
- Core algorithms
  - generate, replace
  - $X'X$
  - Inverses
  - ‘Summers’
  - Solvers
- Modifications to individual important internal routines
- Almost 400 sections of code modified
Stata/MP - All commands

The diagram shows the runtime (percent of single processor) of Stata/MP across different numbers of processors. The x-axis represents the number of processors, ranging from 1 to 16. The y-axis represents the runtime, ranging from 0 to 100 percent.

- **Boundary of all commands**
- **Median runtime**
- **Theoretical maximum**

The theoretical maximum runtime decreases as the number of processors increases, indicating improved parallel processing performance. The boundary of all commands shows the maximum possible runtime, while the median runtime gives a central tendency of the runtime data.
Stata/MP - Estimation commands

Runtime (percent of single processor) vs. Number of processors

- Boundary of all commands
- Median runtime
- Theoretical maximum
Stata/MP - arima

Percentage of single processor time vs. Number of Processors

- Observed
- Modeled
- Perfect scaling
gllamm, i() geqs() link(ologit) family(binom)
gllamm, i() geqs() link(logit) family(binom) nocons

The graph shows the percentage of single processor time against the number of processors. The observed data is represented by dots, the modeled data by a dashed line, and the perfect scaling by a solid line. As the number of processors increases, the percentage of single processor time decreases.
gllamm, i()
gllamm, i() eqs(cons w)