Package ‘SLbench’

November 3, 2010

Type Package

Title Collection of Useful Operations for Benchmarking the Starlab

Version 0.1-0

Date 2010-08-20

Author Jonathan Paul Olmsted <jpolmsted@gmail.com>

Maintainer Jonathan Paul Olmsted <jpolmsted@gmail.com>

Description This package is a collection of functions that were used by the author to conduct benchmarking of the computing resources available to the Political Science Department at the University of Rochester. It covers an array of tests and is meant to be used to time both single-process computations and multi-process (distributed) computations.

License GPL-2

LazyLoad no

Depends R (>= 2.9.2), wnominate (>= 0.94), pscl (>= 0.59), foreach (>= 1.3.0), mvtnorm (>= 0.9-92), SuppDists (>= 1.1-8), Matrix (>= 0.999375-43)

R topics documented:

SLbench-package *Collection of Useful Operations for Benchmarking the Starlab*

Description

This package is a collection of functions that were used by the author to conduct benchmarking of the computing resources available to the Political Science Department at the University of Rochester.

Details

Package:	SLbench
Type:	Package
Version:	0.1-0
Date:	2010-08-20
License:	GPL-2
LazyLoad:	yes
Depends:	R (>= 2.9.2)
Imports:	wnominate (>= 0.94), pscl (>= 0.59), foreach (>= 1.3.0), mvtnorm (>= 0.9-92), SuppDists (>= 1.1-8), Matrix (>=

The package `SLbench` provides a number of functions of the form `test*` which are intended to be called numerous times. By calling the same functions (which each take between 1 and 10 minutes to complete on the base machine) many times, a parallelizable "long job" is created. The purpose is to allow the same "long job" to be run in different computing environments (e.g. alternative BLAS libraries, single-process vs. multi-process) and preserve comparability of the performance results.

Author(s)

Jonathan Paul Olmsted, <`jpolmsted@NOSPAM.gmail.com`>, `www.rochester.edu/college/gradstudents/jolmsted/`

Examples

```
## Code Not Run
## system.time(testStandard())
## system.time(testDisk())
## system.time(testRNG())
## system.time(testIPE())
## system.time(testOptim())
## system.time(testMatrix())
```

`dfAgency` *David Lewis' Agency Creation Dataset*

Description

The data.frame '`dfAgency`' lists the counts of agencies created in each year from 1946 to 1995. In addition, many contextual variables are included. See below for a full description.

Format

A data frame with 50 observations on the following 43 variables.

`secterm` a numeric vector

`tyear` a numeric vector

`war` a numeric vector

`year` a numeric vector

`congress` a numeric vector

`demhouse` a numeric vector

`demsen` a numeric vector

`dempres` a numeric vector

`majorh` a numeric vector

`majors` a numeric vector

`approv` a numeric vector

`unemploy` a numeric vector

`vetoes` a numeric vector

`divided` a numeric vector

`publicl` a numeric vector

`hdem` a numeric vector

`hrep` a numeric vector

`counexec` a numeric vector

`counsec` a numeric vector

`counreor` a numeric vector

`counpma` a numeric vector

`counleg` a numeric vector

`y1983` a numeric vector

`Length` a numeric vector

`reppres` a numeric vector

`clinton` a numeric vector

`counesec` a numeric vector

- y1997 a numeric vector
- y1996 a numeric vector
- bush a numeric vector
- reagan a numeric vector
- carter a numeric vector
- ford a numeric vector
- nixon a numeric vector
- johnson a numeric vector
- kennedy a numeric vector
- eisen a numeric vector
- truman a numeric vector
- pcthmaj a numeric vector
- pctsmaj a numeric vector
- smmaj a numeric vector
- trend a numeric vector
- total a numeric vector

Source

Lewis, David. http://people.vanderbilt.edu/~david.lewis/data.htm.

References

Lewis, David. 2003. *Presidents and the Politics of Agency Design*. Stanford, CA: Stanford University Press.

Examples

```
data(dfAgency)
summary(dfAgency)
```

testIPE *Test Ideal Point Estimation Speed*

Description

This function performs ideal point estimation using a canned procedure and standard dataset.

Usage

```
testIPE()
```

Details

This package uses the canned routine from the `wnominate` package which uses a zig-zag algorithm for the Conditional Likelihood Estimation of ideal points. It is based on the Normal-Normal (Utility-Errors) model of vote choice.

Value

NULL

Note

This function was designed to finish evaluating in about 6 minutes on the author's machine with the default installation of R.

Author(s)

Jonathan Paul Olmsted, `<jpolmsted@NOSPAM.gmail.com>`, `www.rochester.edu/college/gradstudents/jolmsted/`

See Also

`wnominate`

Examples

```
## Code Not Run
## system.time(testIPE())
```

`testMatrix` *Test Matrix Operation Speeds*

Description

This function performs a series of matrix operations.

Usage

```
testMatrix()
```

Details

First, the function sets the psuedo-random number generator seed. Second, it creates a 2,000 by 2,000 matrix populated by draws from a Normal distribution. Third, it performs a series of operations, including: inversion, transposition, multiplication, and then inversion (again).

Value

NULL

Note

This function was designed to finish evaluating in about 1.5 minutes on the author's machine with the default installation of R.

Author(s)

Jonathan Paul Olmsted <jpolmsted@NOSPAM.gmail.com> www.rochester.edu/college/gradstudents/jolmsted/

Examples

```
## Code Not Run
## system.time(testMatrix())
```

---

testOptim *Test Optimization Speeds*

Description

The function `testOptim` estimates a Weibull event count model where the dependent variable is the number of agencies created during a presidency.

Usage

```
testOptim()
```

Details

Describe the model.

Value

NULL

Note

This function was designed to finish evaluation in just over 2 minutes on the author's machine.

Author(s)

Jonathan Paul Olmsted, <jpolmsted@NOSPAM.gmail.com>, www.rochester.edu/college/gradstudents/jolmsted/

Examples

```
## Code Not Run
## system.time(testOptim())
```

testRNG *Test Random Number Generation*

Description

This function draws a series of pseudo-random numbers from several different distributions.

Usage

```
testRNG()
```

Details

The function `testRNG` takes various draws from a univariate standard Normal distribution, a multivariate Normal distribution, a Beta distribution, and a uniform distribution.

Value

NULL

Note

This function was designed to take roughly 5 minutes to complete on the author's personal machine using a standard installation of R.

Author(s)

Jonathan Paul Olmsted, `<jpolmsted@NOSPAM.gmail.com>`, `www.rochester.edu/college/gradstudents/jolmsted/`

Examples

```
## Code Not Run
## system.time(testRNG())
```

testRW *Test Read-Write Speeds*

Description

This function performs an intensive disk-read and disk-write task. A trivial computation is performed with the results repeatedly written to and read from disk.

Usage

```
testRW()
```

Details

First, the function `testRW` initializes a vector of 4,000 random draws from a normal distribution. Then, the function iterates 6,000 times over the following procedure: (1) save the current vector to disk, (2) load the previously saved vector, and (3) concantenate the previous vector and a new draw from a standard normal distribution.

Value

NULL

Warning

This function has the side-effect. A file named 'draws.Rda' will be saved in the current working directory. The user must have write-permissions to the working directory.

Note

This function was designed to finish evaluating in about 1 minute on the author's machine with the default installation of R.

Author(s)

Jonathan Paul Olmsted, `<jpolmsted@NOSPAM.gmail.com>`, `www.rochester.edu/college/gradstudents/jolmsted/`

Examples

```
## Code Not Run
## system.time(testRW())
```

testStandard *Test Standard Benchmarking Script Speeds*

Description

This function performs an array of operations used to benchmark speeds.

Usage

```
testStandard()
```

Details

The function `testStandard` simply runs the most recent standard benchmarking script from Simon Urbanek. As of this version of this package, this is the script ending in `*25.R`.

Value

NULL

Note

This function was designed to finish evaluating in about 9.5 minutes on the author's machine with the default installation of R.

Author(s)

Jonathan Paul Olmsted, `<jpolmsted@NOSPAM.gmail.com>`, `www.rochester.edu/college/gradstudents/jolmsted/`

References

Urbanek, Simon. 2008. R Benchmarks, `http://r.research.att.com/benchmarks/`.

Examples

```
## Code Not Run
## system.time(testStandard())
```

Index