

R: a nearly-Lisp

Christophe Rhodes

Teclo Networks AG

April 6, 2011

Outline

Introduction

Examples

Repeated Measurement
Trellis Graphics

R and Lisp

The logo for teclo NETWORKS. The word "teclo" is written in a large, lowercase, sans-serif font. Below it, the word "NETWORKS" is written in a smaller, uppercase, sans-serif font, with each letter spaced out. The entire logo is rendered in a light blue color. To the left of the logo is a large, stylized blue graphic consisting of several overlapping, curved lines that suggest motion or a network structure.

Outline

Introduction

Examples

Repeated Measurement
Trellis Graphics

R and Lisp

The logo for teclo NETWORKS features a large, stylized blue 'e' on the left. To its right, the word 'teclo' is written in a bold, lowercase, sans-serif font. Below 'teclo', the word 'NETWORKS' is written in a smaller, uppercase, sans-serif font, with wide letter spacing.

teclo
NETWORKS

Introduction

History and Background: R

“a free software environment for statistical computing and graphics”



Introduction

History and Background: R

“a free software environment for statistical computing and graphics”

- ▶ “free”:
 1. you don't have to pay for it;
 2. you are (broadly) free to modify it for your own purposes;
 3. you don't get to whine at the R developers if it doesn't work for you (unless you pay for support).

Introduction

History and Background: R

“a free software environment for statistical computing and graphics”

- ▶ “free”:
 1. you don't have to pay for it;
 2. you are (broadly) free to modify it for your own purposes;
 3. you don't get to whine at the R developers if it doesn't work for you (unless you pay for support).
- ▶ “statistical computing”
 1. modelling, tests, time-series analysis, classification, clustering, and so on
 2. typical strength: vector computations on large datasets, provided with BLAS and LAPACK

Introduction

History and Background: R

“a free software environment for statistical computing and graphics”

- ▶ “free”:
 1. you don't have to pay for it;
 2. you are (broadly) free to modify it for your own purposes;
 3. you don't get to whine at the R developers if it doesn't work for you (unless you pay for support).
- ▶ “statistical computing”
 1. modelling, tests, time-series analysis, classification, clustering, and so on
 2. typical strength: vector computations on large datasets, provided with BLAS and LAPACK
- ▶ “graphics”
 1. many predefined graphical facilities;
 2. publication-quality output.

Introduction

History and Background: R

Abbreviated timeline:

- ▶ S (John Chambers): similar to but not exactly like Scheme
- ▶ Public R release in 1993; Free Software in 1995
- ▶ Core group formed in 1997
- ▶ R version 1.0.0 released in 2000
- ▶ biannual releases continue

teclo
NETWORKS

Introduction

History and Background: R

Abbreviated timeline:

- ▶ S (John Chambers): similar to but not exactly like Scheme
- ▶ Public R release in 1993; Free Software in 1995
- ▶ Core group formed in 1997
- ▶ R version 1.0.0 released in 2000
- ▶ biannual releases continue

Compare with:

- ▶ S-PLUS (commercial release of S)
- ▶ SAS, Stata, JAGS
- ▶ Scilab, Octave, Matlab
- ▶ Gnuplot, Spreadsheets

Introduction

History and Background: R

Web:

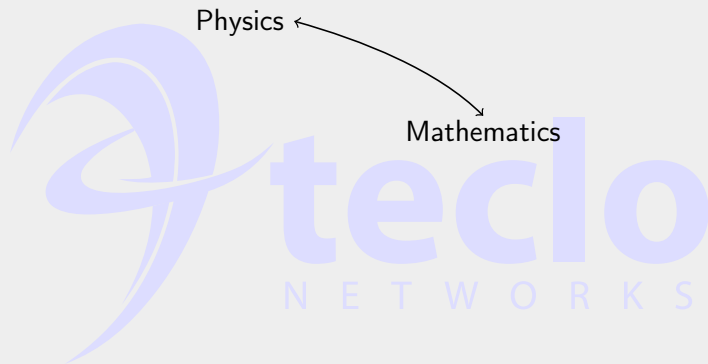
- ▶ R home page: <http://www.r-project.org/>
- ▶ Emacs Speaks Statistics: <http://ess.r-project.org/>
- ▶ Comprehensive R Archive Network:
<http://cran.r-project.org/>
- ▶ R Journal: <http://journal.r-project.org/>
- ▶ StackOverflow:
<http://stackoverflow.com/questions/tagged/r>
- ▶ RSeek: <http://www.rseek.org/>

Mail / News:

- ▶ R help: r-help@r-project.org /
gmane.comp.lang.r.general
- ▶ ESS help: ess-help@stat.math.ethz.ch /
gmane.emacs.ess.general

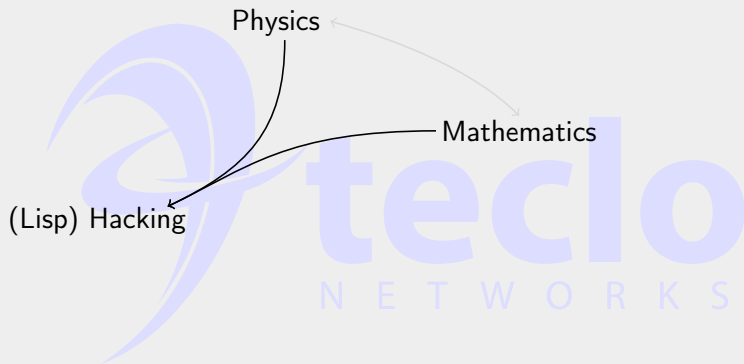
Introduction

History and Background: Me



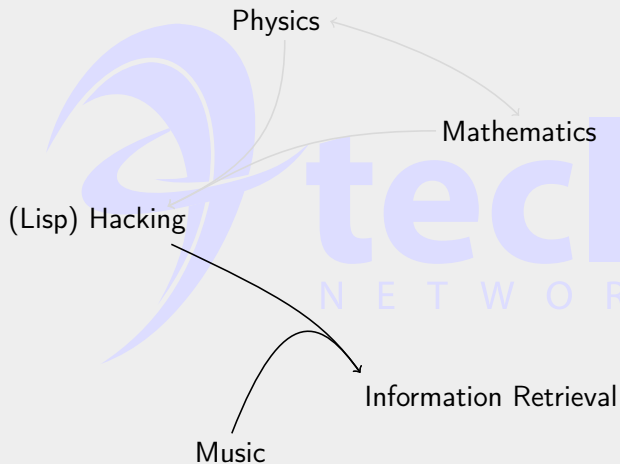
Introduction

History and Background: Me



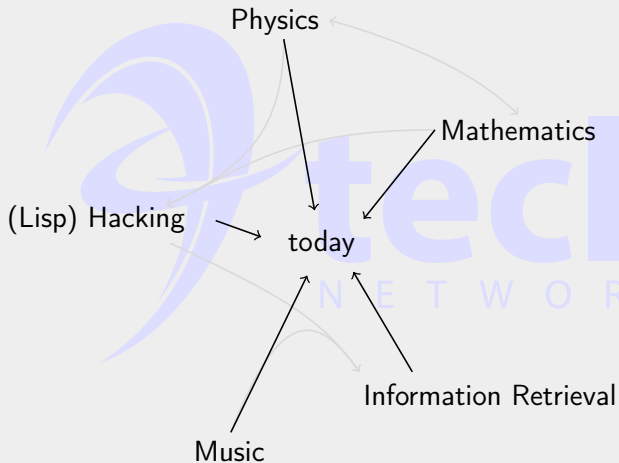
Introduction

History and Background: Me



Introduction

History and Background: Me



Introduction

R syntax

Very close to the original S:

- ▶ constants: numeric (1, 3:5, 4.2) and text ("foo")
- ▶ operators: arithmetic (+, *, %*%) and logical (<, &, %in%)
- ▶ function calls:
 - ▶ `seq(1,10)`
 - ▶ `seq(from=1, 10)`
 - ▶ `seq(to=10, from=1)`
 - ▶ `seq(1, 10, by=1)`
- ▶ assignment: `<-` (also `=`)
- ▶ loop constructs: `while`, `for`
- ▶ conditional expressions: `if` (but see also `ifelse`)

R data types

- ▶ vectors
 - ▶ character
 - ▶ numeric
 - ▶ double
 - ▶ integer
 - ▶ complex
 - ▶ logical
- ▶ list (generic vectors, dotted pairs)
- ▶ data frames
- ▶ attributes

Introduction

R semantics

Function calls and scope:

- ▶ lexical binding
- ▶ (abbreviatable) keyword arguments
- ▶ lazy argument evaluation
- ▶ split-horizon scoping
- ▶ copy-on-write modification
 - ▶ `<<-` to override
- ▶ first-class environments, argument to `eval`

Outline

Introduction

Examples

Repeated Measurement
Trellis Graphics

R and Lisp

The logo for teclo NETWORKS features a large, stylized blue 'e' on the left. To its right, the word 'teclo' is written in a lowercase, bold, sans-serif font. Below 'teclo', the word 'NETWORKS' is written in a smaller, uppercase, spaced-out sans-serif font. The entire logo is rendered in a light blue color.

teclo
NETWORKS

Outline

Introduction

Examples

Repeated Measurement

Trellis Graphics

R and Lisp

The logo for teclo NETWORKS features a large, stylized blue 'e' on the left. To its right, the word 'teclo' is written in a bold, lowercase, blue sans-serif font. Below 'teclo', the word 'NETWORKS' is written in a smaller, uppercase, blue sans-serif font, with each letter spaced out.

teclo
NETWORKS

Examples

Repeated Measurement

Motivation:

- ▶ Simple example
- ▶ Introduction to functionality



Examples

Repeated Measurement

Motivation:

- ▶ Simple example
- ▶ Introduction to functionality
- ▶ Single most useful thing to know about measurement

Examples

Repeated Measurement

Setup:

- ▶ Some quantity that we want to measure;
- ▶ Measurement is noisy.
 - ▶ could be 'random noise' in our equipment;
 - ▶ could be other systematic effects;

More specifically:

- ▶ measure: $\{ \} \rightarrow \mu + \epsilon$
- ▶ $\epsilon \sim D(0, \sigma^2)$
- ▶ $Cov(\epsilon_i, \epsilon_j) = 0$

Examples

Repeated Measurement

What do we expect when we take a measurement?

- ▶ a value somewhere near the 'true' value;
- ▶ but could be a long way away;
- ▶ in general, don't even know how much noise there is.

Everyone knows what to do: take more measurements and average...

Examples

Repeated Measurement

What do we expect when we take a measurement?

- ▶ a value somewhere near the 'true' value;
- ▶ but could be a long way away;
- ▶ in general, don't even know how much noise there is.

Everyone knows what to do: take more measurements and average...

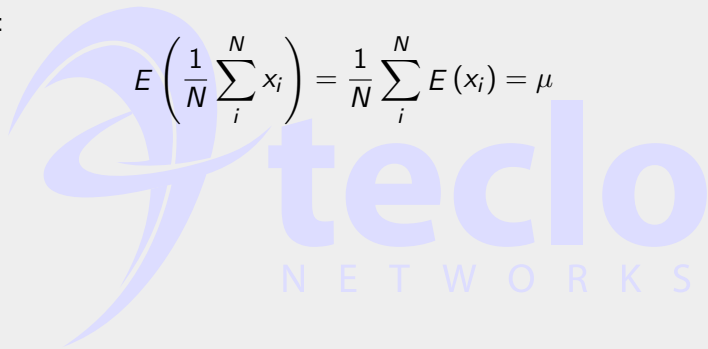
- ▶ ...but why?

Examples

Repeated Measurement

We expect that the average we compute is, on average, the true value:

$$E\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N}\sum_i^N E(x_i) = \mu$$



Examples

Repeated Measurement

We expect that the average we compute is, on average, the true value:

$$E\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N}\sum_i^N E(x_i) = \mu$$

What is the variance about this true value?

$$\text{Var}\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N^2}\text{Var}\left(\sum_i^N x_i\right) = \frac{1}{N^2}\times N\sigma^2 = \frac{\sigma^2}{N}$$

Examples

Repeated Measurement

We expect that the average we compute is, on average, the true value:

$$E\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N}\sum_i^N E(x_i) = \mu$$

What is the variance about this true value?

$$\text{Var}\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N^2}\text{Var}\left(\sum_i^N x_i\right) = \frac{1}{N^2}\times N\sigma^2 = \frac{\sigma^2}{N}$$

Standard deviation of the average scales as $\frac{1}{\sqrt{N}}$

Examples

Repeated Measurement

We expect that the average we compute is, on average, the true value:

$$E\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N}\sum_i^N E(x_i) = \mu$$

What is the variance about this true value?

$$\text{Var}\left(\frac{1}{N}\sum_i^N x_i\right) = \frac{1}{N^2}\text{Var}\left(\sum_i^N x_i\right) = \frac{1}{N^2}\times N\sigma^2 = \frac{\sigma^2}{N}$$

Standard deviation of the average scales as $\frac{1}{\sqrt{N}}$

[wait a minute, this was meant to be a talk about R]

Outline

Introduction

Examples

Repeated Measurement

Trellis Graphics

R and Lisp

The logo for teclo NETWORKS features a large, stylized blue 'e' on the left. To its right, the word 'teclo' is written in a bold, lowercase, sans-serif font. Below 'teclo', the word 'NETWORKS' is written in a smaller, uppercase, sans-serif font, with each letter spaced out.

Examples

Trellis Graphics

Motivation:

- ▶ Clear display of complex, multivariate information
- ▶ Rapid experimentation
- ▶ Adequate defaults, hooks everywhere

Examples

Trellis Graphics

Motivation:

- ▶ Clear display of complex, multivariate information
- ▶ Rapid experimentation
- ▶ Adequate defaults, hooks everywhere
- ▶ Teach how not to lie with statistics
- ▶ Defeat 'bad graph of the week' syndrome

Examples

Trellis Graphics

Distinct graphical and graphing system, originally for S+:

- ▶ Multipanel Conditioning
- ▶ Banking to 45°
- ▶ Automation
- ▶ Customization

Becker, R. A. and Cleveland, W. S., *S-PLUS Trellis Graphics User's Manual*, Seattle: MathSoft, Inc., Murray Hill: Bell Labs, 1996.

Examples

Trellis Graphics: Multipanel Conditioning

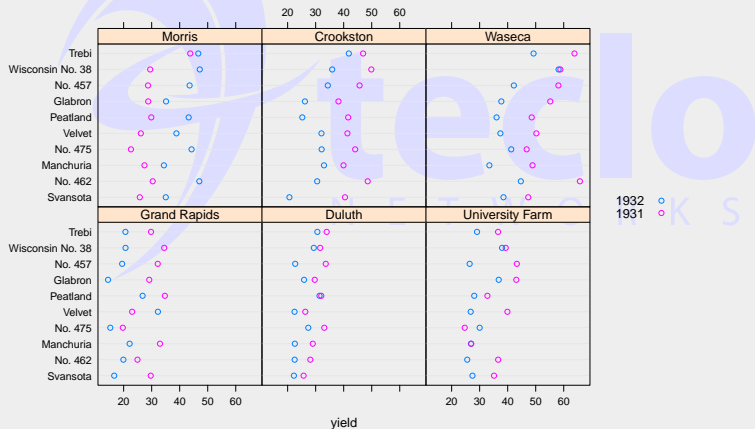
```
dotplot(variety~yield|site, data = barley, groups = year,  
        key = simpleKey(levels(barley$year), space = "right"),  
        xlab = "yield")
```



Examples

Trellis Graphics: Multipanel Conditioning

```
dotplot(variety~yield|site, data = barley, groups = year,  
        key = simpleKey(levels(barley$year), space = "right"),  
        xlab = "yield")
```



Examples

Trellis Graphics: Banking to 45°

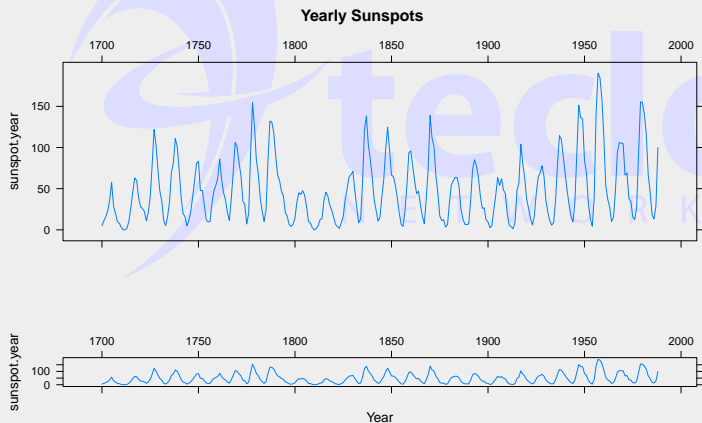
```
xyplot(sunspot.year)  
xyplot(sunspot.year, aspect="xy")
```



Examples

Trellis Graphics: Banking to 45°

```
xyplot(sunspot.year)  
xyplot(sunspot.year, aspect="xy")
```



Examples

Trellis Graphics: Automation

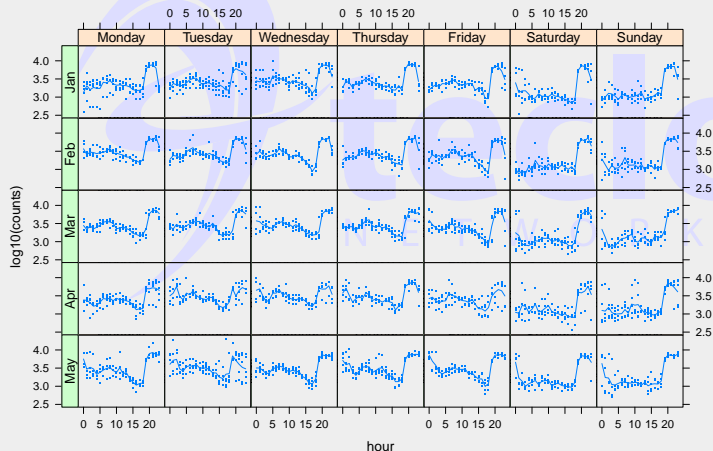
```
useOuterStrips(  
  xyplot(log10(counts)~hour|weekday+month, data=biocAccess,  
         type=c("p", "a"), as.table=TRUE, pch=".", cex=2))
```



Examples

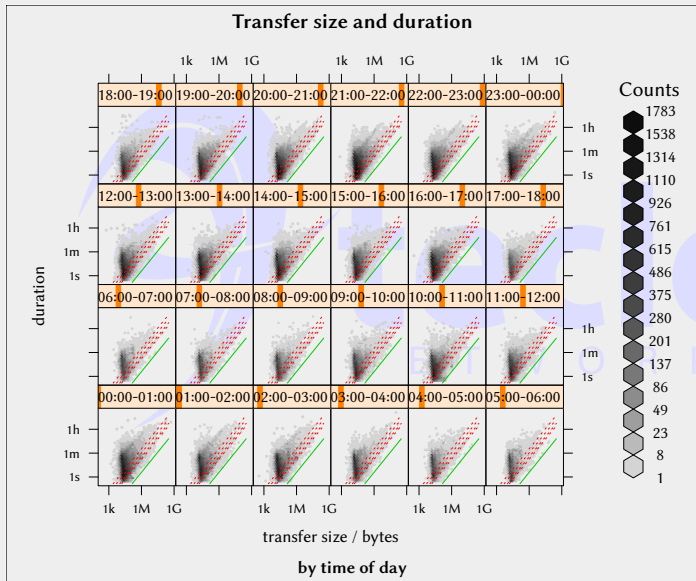
Trellis Graphics: Automation

```
useOuterStrips(  
  xyplot(log10(counts)~hour|weekday+month, data=biocAccess,  
        type=c("p", "a"), as.table=TRUE, pch=".", cex=2))
```



Examples

Trellis Graphics: Customization



Trellis Graphics

Lattice

Implementation of Trellis concepts in R:

- ▶ 'high-level' functions: `xyplot`, `bwplot`, `densityplot`
- ▶ *panel* functions: `panel.xyplot`, `panel.bwplot`, `panel.densityplot`
- ▶ utility functions: `useOuterStrips`, `resizePanels`

Sarkar, D., *Lattice: Multivariate Data Visualization with R*, New York: Springer Science+Business Media, 2008.

Outline

Introduction

Examples

Repeated Measurement

Trellis Graphics

R and Lisp

The logo for teclo NETWORKS. It features a large, stylized blue 'e' on the left. To its right, the word 'teclo' is written in a bold, lowercase, blue sans-serif font. Below 'teclo', the word 'NETWORKS' is written in a smaller, uppercase, blue sans-serif font, with wide letter spacing.

teclo
NETWORKS

R and Lisp

Lisp-like features in R

analogues to read, eval, print

- ▶ parse:
 - ▶ apply to character vector or file;
 - ▶ returns a list of expressions, optionally decorated with source location information;
- ▶ eval
 - ▶ takes (list of) expressions as argument;
 - ▶ returns value of last expression;
 - ▶ optional (lexical) environment argument, defaults to local
- ▶ deparse
 - ▶ string representation of object
 - ▶ approximately inverse of parse

R and Lisp

Lisp-like features in R

`with-output-to-string`:

- ▶ `capture.output`
- ▶ also similar stream concept to/from a variety of sources



R and Lisp

Lisp-like features in R

`with-output-to-string`:

- ▶ `capture.output`
- ▶ also similar stream concept to/from a variety of sources

`setf`:

- ▶ `x <- 3`
- ▶ `foo(x) <- 3`
 - ▶ `'foo<-'(x, value=3)`

teclo
NETWORKS

R and Lisp

Lisp-like features in R

`with-output-to-string`:

- ▶ `capture.output`
- ▶ also similar stream concept to/from a variety of sources

`setf`:

- ▶ `x <- 3`
- ▶ `foo(x) <- 3`
 - ▶ `'foo<-'(x, value=3)`

conditions, handlers and restarts:

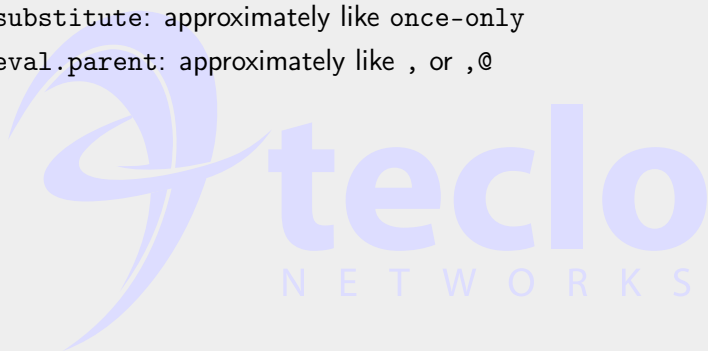
- ▶ `try`
- ▶ `tryCatch`
- ▶ `withCallingHandlers`
 - ▶ `signalCondition`
 - ▶ `simpleError`, `simpleWarning`, `simpleMessage`
 - ▶ `withRestarts`, `findRestart`, `invokeRestart`

R and Lisp

Lisp-like features in R

Macros:

- ▶ `substitute`: approximately like `once-only`
- ▶ `eval.parent`: approximately like `,` or `,@`



R and Lisp

Lisp-like features in R

Macros:

- ▶ `substitute`: approximately like `once-only`
- ▶ `eval.parent`: approximately like `,` or `,@`

```
withRetryRestart <- function(description, expr) {  
  call <- substitute(expr)  
  retry <- TRUE  
  while(retry) {  
    retry <- FALSE  
    withRestarts(eval.parent(call),  
                 retry=list(description=description,  
                             handler=function() retry <-< TRUE))  
  }  
}
```

R and Lisp

Lisp-like features in R

Macros:

- ▶ `substitute`: approximately like `once-only`
- ▶ `eval.parent`: approximately like `,` or `,@`

```
withRetryRestart <- function(description, expr) {  
  call <- substitute(expr)  
  retry <- TRUE  
  while(retry) {  
    retry <- FALSE  
    withRestarts(eval.parent(call),  
                 retry=list(description=description,  
                             handler=function() retry <-< TRUE))  
  }  
}
```

```
withRetryRestart("retry SLIME interactive evaluation request",  
                 value <- eval(parse(text=string), envir=globalenv()))
```

R and Lisp

Lisp-like features in R: SLIME

SLIME: Superior Lisp Interaction Mode for Emacs.

- ▶ started in 2003 for CMUCL only;
- ▶ it was too easy to port to other Common Lisps



R and Lisp

Lisp-like features in R: SLIME

SLIME: Superior Lisp Interaction Mode for Emacs.

- ▶ started in 2003 for CMUCL only;
- ▶ it was too easy to port to other Common Lisps

Vague server (swank) backends for:

- ▶ Ikarus, Kawa, Larceny, MIT Scheme, R6RS
- ▶ Goo
- ▶ Jolt
- ▶ Ruby

(other client implementations: Climacs, Vim, MCLIDE)

R and Lisp

Lisp-like features in R: SLIME

SLIME: Superior Lisp Interaction Mode for Emacs.

- ▶ started in 2003 for CMUCL only;
- ▶ it was too easy to port to other Common Lisps

Vague server (swank) backends for:

- ▶ Ikarus, Kawa, Larceny, MIT Scheme, R6RS
- ▶ Goo
- ▶ Jolt
- ▶ Ruby

(other client implementations: Climacs, Vim, MCLIDE)

swankr: <http://common-lisp.net/~crhodes/swankr/>

- ▶ Lots doesn't work. Please help!

R and Lisp

Lisp-like features in R: SLIME

SLIME: Superior Lisp Interaction Mode for Emacs.

- ▶ started in 2003 for CMUCL only;
- ▶ it was too easy to port to other Common Lisps

Vague server (swank) backends for:

- ▶ Ikarus, Kawa, Larceny, MIT Scheme, R6RS
- ▶ Goo
- ▶ Jolt
- ▶ Ruby

(other client implementations: Climacs, Vim, MCLIDE)

swankr: <http://common-lisp.net/~crhodes/swankr/>

- ▶ Lots doesn't work. Please help!

Any questions?