

O Sistema R e Computação Estatística

Paulo Justiniano Ribeiro Junior

LEG: Laboratório de Estatística e Geoinformação

Universidade Federal do Paraná

<mailto:paulojus@ufpr.br>

<http://www.leg.ufpr.br/~paulojus>

1 O projeto R

para

computação estatística

1.1 O que é o R?

R é um sistema para computação estatística e gráficos. Consiste de uma linguagem mais um ambiente de operação com gráficos, um *debugger*, acesso a certas funções do sistema e capacidade de rodar comandos armazenados em arquivos (*script*).

Influenciado por duas linguagens: S e Scheme, com aparência semelhante ao S e implementações e semânticas internas similares ao Scheme.

Linguagem interpretada, programação modular via funções, interfaces com C, C++ e Fortran, implementando uma diversidade de métodos estatísticos.

Sítio oficial: <http://www.r-project.org>

Área de *download* e espelhos: <http://cran.r-project.org>

2 O contexto:

Estatística computacional e
o desenvolvimento da linguagem S
e seus *dialetos*

2.1 Computação Estatística: 1980's

Principalmente *Fortran* ou PL/I (*SAS*).

Computação em “batch” (*SAS*, *BMDP*, *SPSS*, *Genstat*) com restrições de plataforma.

Alguns pequenos sistemas interativos (*GLIM*, *Minitab*).

Recursos gráficos limitados – impressão técnica via microfilmes, etc.

Soluções individuais em pesquisa (p.e. *GLIM* macros).

2.2 Computação Estatística: 1990's

Disseminação de PC's e algumas estações para pesquisa e ensino em alguns locais.

início: bons gráficos, “postscript”, muitos terminais mono.

fim: bons gráficos, postscript até 1000×1000 pixels, ainda muitos terminais mono.

C começa a ser mais usado por ser mais adaptável que *Fortran*.

Ainda *SAS*, *SPSS* etc. para programas em “batch”.

S começa a ter impacto em pesquisa e ensino.

2.3 Computação Estatística: 2000's

Velocidade de processamento.

Cores largamente disponíveis, usualmente 24-bits.

“Geração vídeo-game” agora na universidade.

Poucas pessoas sequer pensam em escrever programas completos para ideias de pesquisa: O paradigma é: crie protótipo e distribua em linguagens de alto nível tais como *S*, *MatLab*, *Ox*, *Gauss*, ...

Fortran ainda usado em computação científica, mas *C* e *C++* usualmente preferidos.

Alguns advogam o uso de *Java*. *SAS* ainda usado como “pseudo-batch”.

Ferramentas especializadas *Perl*, *Python*, *Web browsers*.

Tendência p/ *XML* (eXtensible Markup Language) com *MathML*.

2.4 A linguagem S

Inicialmente trabalho do Dr. John M. Chambers do *Bell Laboratories* (antiga *AT&T*, atualmente *Lucent Technologies*).

Ganhador em 1998 do prestigioso prêmio *Association for Computing Machinery Award for Software Systems* por, nas palavras da citação:

pelo sistema S, que mudou para sempre a forma como pessoas analisam, visualizam e manipulam dados.

Durante a última década foi o principal veículo para disponibilizar novas metodologias estatísticas aos usuários finais.

S tem uma longa história: o sistema gráfico remonta a 1976.

J. Chambers agora é *Bell Labs Fellow*, membro do R *core team* e trabalha no Omegahat, que pode então ser considerado o sucessor da linguagem S.

2.5 História da linguagem S

Nome da linguagem oscilou e os “sabores” de S são conhecidos pelas cores das capas dos livros que tiveram J. Chambers como co-autor:

- S1 1984 *brown book*: linguagem baseada em macros.
- S2 1988 *blue book*: extensões por usuários como primeiras classes.
- S3 1991 *white book*: estrutura de classes, funcionalidade estatística.
- S4 1998 *green book*: sistema de classes mais rígido.

Tudo era programas Unix escritos em *C* e *Fortran*.

S-PLUS produzido em 1988 em Seattle pela *Statistical Sciences* que em 1993 adquiriu direitos de exclusividade de mercado sobre S e se fundiu com a *Mathsoft*. Em 2001 separaram-se e tornou-se *Insightful*.

S não é (ou era) visto pelos desenvolvedores como um sistema estatístico, mas sim como um ambiente interativo para gráficos e análise de dados, um sistema para se fazer estatística dentro dele.

2.6 S-PLUS

S-PLUS disponível para um limitado espectro de plataformas (Unix, DOS, Windows).

Versão para LINUX somente em 1998, e não para Macintoshes.

Versão UNIX baseada em S4 desde 1998, para Windows a partir de 2001.

S-PLUS muito usado para ensino de estatística a nível de pós-graduação.

Embora também usado para cursos de serviço, teve menor impacto para ensino a nível de graduação.

Licenças acadêmicas caras.

Atualmente tem feito muito sucesso em setores comerciais (finanças, indústria farmacêutica, etc.).

3 O que é o R ?

3.1 História do R

R é um sistema originalmente escrito por Ross Ihaka e Robert Gentleman da *University of Auckland* no começo dos anos 90.

Ao usuário parece um dialeto da linguagem S mas internamente é baseado em ideias de *Scheme* (um membro da “família” LISP).

Muito parecido com S3.

Provavelmente iniciado como um projeto de pesquisa, mas usado em Auckland para cursos básicos em Macintoshes com 2Mb de memória.

Artigo de R&R na Computer Sciences em 1996.

Em 1997 outros se envolveram e criou-se um *core team* com acesso ao código.

Havia versão para Windows, usuários de Linux alavancaram o desenvolvimento, não havia versão de *S-PLUS*.

3.2 Curiosidades sobre o R

Por que o nome "R"?

Exemplo típico de *humor-net*:

Inicialmente versão reduzida da linguagem S, portanto faz sentido usar uma letra precedente no alfabeto.

Ross e **R**obert — simples coincidência ...?

1995: R&R lançam código sob GPL.

Período coincidente com o "boom" de "código aberto" motivado pelo LINUX.

Não havia softwares estatísticos para LINUX - só linguagens.

3.3 Como está o R no momento

Primeira versão não-beta (1.0) lançada em 29 de fevereiro de 2000.

Versão atual: 2.1.0.

Sistema disponível com código aberto.

Distribuído segundo termos da GNU—GPL2.

Disponível no formato compilado (binários) e/ou fontes + *scripts* de compilação.

Multiplataforma: compila em Windows, Linux, Mac, Unix, FreeBSD, etc.

Tipicamente duas versões por ano.

R-patched e *R-devel* atualizados diariamente.

3.4 The R FOUNDATION

R Foundation: criada em 2003.

Citando o R em publicações:

```
> citation()
```

To cite R in publications use:

```
R Core Team (2026). R: A Language and Environment for Statistical Computing.  
R Foundation for Statistical Computing, Vienna, Austria.doi:10.32614/R.manuals  
<https://doi.org/10.32614/R.manuals><https://www.R-project.org/>.
```

E leia com **muita** atenção no final da mensagem:

```
We have invested a lot of time and effort in creating R, please cite it when  
using it for data analysis. See also 'citation("pkgname")' for citing R packages
```

3.5 R vs S-PLUS

Co-existência dos sistemas (nem sempre pacífica).

S-PLUS é comercial, R é gratuito.

R é mais leve, requer menos hardware, S-PLUS é monolítico e R tem um pequeno núcleo e extensões.

S-PLUS com GUI “oficial”.

Performance comparável, embora R seja mais tolerante a código “mal escrito” que pode fazer o S-PLUS “travar”.

No início R com mais bugs, porém mais rapidamente corrigidos.

Ambos têm excelente qualidade gráfica, c/ limitações em 3D e gráficos dinâmicos.

Pesquisadores com ênfase em computação estatística migraram do S para o R.

J. Chambers é membro do R *core team*.

Tendência atual é mais de colaboração do que competição.

R *vs* S-PLUS (cont.)

S (como C) usa *static scoping*.

R (como Scheme) usa *lexical scoping*.

Consequências práticas:

1. incompatibilidades entre códigos.
2. tratamento de variáveis livres em funções.
3. objetos em vários arquivos (S) *vs* arquivo único (*workspace* - R).
4. velocidade.
5. riscos de perda de trabalho (*crashes*).
6. outras diferenças.

4 Uso do R

Algumas Questões

4.1 Para que o R é usado?

Impossível dizer pois é livremente disponível.

Listas e páginas-web dão alguma ideia do uso.

Palavras de um influente membro do R *core team*:

One of my main motivations for being involved is a (perhaps the) major use, to provide a first-class statistical system to students and researchers in the third world.

Atualmente usado para análises estatísticas de larga escala.

Aplicações em *micro-arrays* - THE BIOCONDUCTOR PROJECT.

Pesquisadores em várias companhias estão desenvolvendo seus sistemas a partir do R.

Ambiente de desenvolvimento, implementação e teste de novas metodologias estatísticas através dos *pacotes*.

4.2 Alguns recursos

Uso típico - linha de comando.

Mas há muito além disto ...

- Rcgi, Rweb, JGR.
- interfaces TCL/TK.
- Rsciview, Rcmdr.
- ... outros projetos RGUIs em franca atividade.

Alguns recursos (cont.)

Pacotes: + de 300, atualizações frequentes.

Metodologias recentes e/ou em desenvolvimento.

Dinâmica de "Patch" e checagem diária de pacotes.

Disponível como biblioteca compartilhada (*shared library*) e/ou estática (*static library*).

Interfaces com programas e linguagens, possibilidades diversas via integração com outros recursos.

Embedding reserva ao R o que ele tem de melhor: capacidade de produzir análises estatísticas e gráficos.

4.3 R é um projeto atípico

R não tem um líder e se baseia no consenso entre o R *core team*.

Há áreas de "expertise" entre os membros.

Deferência especial com os "fundadores".

Core team: *modus operandi* e diretrizes, encontros regulares (*DSC e UseR meetings*) e aparentemente excelente relacionamento.

4.4 Alguns tópicos “difícies”

Ross Ihaka teve longa disputa com a universidade para “liberar” seu trabalho com o R.

O direito de se construir um sistema comercial baseado no R não é claro.

A propriedade do código-fonte não é bem definida.

Por ex. R usa algoritmos estatísticos da *RSS*, com licença sob o entendimento de que o projeto não é comercial.

Projetos livres são enormemente trabalhosos:

- Usuários demandam: funcionamento como esperam e reparos.

- Têm o hábito de reportar/perguntar antes de ler manuais.

- Usuários que mais demandam provavelmente usam para ganhos comerciais.

Possível solução (como em LINUX) é prover suporte para produto gratuito.

Compatibilidade de versões e dificuldades com *entranhas* dos sistemas operacionais.

4.5 Alguns pontos fortes do projeto

R é largamente usado por grupos em países onde um sistema comercial é proibitivo e roda bem em hardware “quase obsoleto”.

Listas (R-help, R-packages, R-announce e R-devel).

Quase todo contato por internet.

Fácil adição de novos aspectos pelo usuário.

Possibilidades didáticas.

Encontrou um bug: arrume e prossiga!

Mais aspectos de orientação a objetos nas novas versões.

Sinergia com DBMS's & mais uso/integração via XML.

4.6 Alguns pontos fortes do projeto

Ênfase em compatibilidade com várias plataformas.

Disponibilidade de documentação e materiais.

Desenvolve senso de apreciação pelo desenho de software e suporte.

Competência, orientação e atitudes do *R Core-Team*.

Velocidade na disponibilização e divulgação de novas metodologias.

Padrão de qualidade na manutenção do projeto.

5 Usando o R

Comentários

5.1 Estrutura Atual

Pacotes:

- **base:**
base*, datasets, drDevices, graphics*, grid, methods*, splines, stats*, stats4, tcltk, tools, utils*
* indica pacotes que são carregados automaticamente ao iniciar o R.
- **recommended:**
boot, cluster, foreign, KernSmooth, lattice, mgcv, nlme, rpart, survival, bundle VR (nnet, MASS, spatial)
- **contributed packages**
fontes: CRAN, OMEGAHAT, BIOCONDUCTOR.
- *unofficial packages*

Pacotes disponíveis:

- para listar: `library()`
- para carregar: `library(pacote)`

5.2 R, (X)emacs e ESS

emacs: editor genérico com facilidades para diversas linguagens.

ess: **e**macs **s**peaks **s**tatistics.

módulo para integrar e facilitar o uso de programas estatísticos com (x)emacs.

suporte para: R, S-PLUS, SAS, Stata, BUGS.

para carregar coloque em `.emacs/init.el`:

```
(require 'ess-site)
```

5.3 *Demos*

`> demo(graphics)`

`> demo(plotmath)`

`> demo(image)`

`> demo(persp)`

5.4 Interfaces TCL/TK

```
> require(tcltk)
  > demo(tktttest)
  > demo(tkdensity)
  > demo(tkcanvas)

> require(geoR)
  > data(s100)
  > vg <- variog(s100, max.dist=1)
  > fit <- eyefit(vg)
```

5.5 Rodando em modo *batch*

automatização de tarefas e análises.

análises longas (estudos de simulação).

R CMD BATCH script.R &

saída *default* em script.Rout.

comando *completo*:

```
R CMD BATCH [options] infile [outfile]
```

6 Uma visão pessoal e institucional

6.1 Como comecei e porque uso o R

1998 – S e início da geoS – *S-PLUS* Ambiente Unix.

Dificuldades

Rotinas numéricas e Bayesianas.

Evitar *loops*.

uso de memória.

velocidade.

Soluções

Programação eficiente (em S).

Transcrição de partes do código para C.

R.

Outras Motivações: Sistemas LINUX, código aberto, custos, e perspectivas na volta ao Brasil.

Mudança inicialmente *subversiva*, posteriormente largamente adotada.

6.2 Uso do R no DEST/UFPR

Como parte de Projeto De Recursos Computacionais no Apoio ao Ensino e Pesquisa.

Concepção: projeto de baixo custo com aproveitamento de hardware obsoleto, modelo cliente–servidor, com uso exclusivo de programas gratuitos (e de preferência com código aberto), administração facilitada.

Básico: Linux + OpenOffice + R + L^AT_EX.

Vantagens: distribuição livre, integração, multi-plataforma, arquivos de comandos (*scripts*).

6.3 Projetos

LEG : Laboratório de Estatística e Geoinformação

- <http://www.est.ufpr.br/leg>
- `geoR` e `geoRglm`.
- `aRT`: API R-Terralib.
- `myR`.
- `Rcitrus`.
- parte do projeto URR (Ultimate Research Resources).
- pacote com funções de apoio ao ensino.

Parcerias

- Rede SAUDAVEL (Parceria DPI/INPE, Fiocruz, LESTE/UFMG, LEG/UFPR).
- FUNDECITRUS.
- CESO - DUKE.

7 Material de apoio didático/científico:

Pacotes e Sweave

7.1 Construindo pacotes

Modelos para pacote.

Aprenda com os outros!

Estrutura organizada.

Testes e documentação.

Pacotes oficiais e não oficiais.

Ideal para divulgação de trabalhos de pesquisa.

Ideal para instrumentos de apoio didático, produção de cursos e materiais como livros, apostilas, etc.

7.2 Sweave

Integração de R com L^AT_EX.

Conceito de Ciência Reprodutível.

Preparação de artigos, livros, apostilas, etc.

Documentos dinâmicos.

8 Conclusões

8.1 Características fundamentais do R

R é uma linguagem para manipular objetos. Portanto temos que entender:

que tipo de linguagem é.

de que forma manipula objetos e que tipo de manipulação.

que tipos de objetos.

O R pode fazer *isto ou aquilo*? — pergunta errada!

Como pode fazer *isto ou aquilo*?

Isto já foi feito antes?

O que eu preciso para fazer o que desejo? Preciso de programação de baixo nível?

A principal característica é: poderosa ferramenta para transferência de tecnologia.

R (S) é uma linguagem baseada em funções: não há subrotinas.