R语言—介绍

数理信息学院
汤银才

R介绍

- 为什么要学习R语言
- R的资源
  - 资料与文献
  - 网站
  - 统计分析软件包
- R的运行平台
  - R_GUI
  - R_Commander
  - R_WinEdt
  - R_ESS+XEmacs

- 语言/软件使用示例
  - John Verzani, Simple R – Using R for Introductory Statistics, 2002
    http://www.math.csi.cuny.edu/Statistics/R/simpleR/index.html
  - J.H. Maindonald, Using R for Data Analysis and Graphics – An Introduction, 2001
  - Julian J. Faraway, Practical Regression and ANOVA Uing R, 2002
    http://www.stat.lsa.umich.edu/~faraway/
    快速入门：http://www.stat.lsa.umich.edu/~faraway/stat500/introR.html
  - S. Chatterjee, A.S. Hadi & B. Price, Regression Analysis by Example, 3rd Ed., John Wiley & Sons, 2000
    http://www.ilr.cornell.edu/~hadi/RABE/
    (印影版，中国统计出版社，2003)

R的无私奉献者

Ross Ihaka **Robert Gentleman** **Bill Venables**

为什么要学习R语言

- R是
  - 一个开放(GPL)的统计编程环境
  - 一种语言，是S语言(由AT&T Bell实验室的Rick Becker, John Chambers,Allan Wilks开发)的一种方言(dialect) 之一，另一则为S-plus.
  - 一种软件，是集统计分析与图形直观显示于一体的统计分析
- R作为一个计划(project)，最早(1995年)是由Auckland大学统计系的Robert Gentleman和Ross Ihaka开始编制，目前由R核心开发小组(R Development Core Team – 以后用R DCT表示)维护，他们完全自愿、工作努力负责，并将全球优秀的统计应用软件打包提供给我们。我们可以通过R计划的网站(http://www.r-project.org)了解有关R的最新信息和使用说明，得到最新版本的R软件和基于R的应用统计软件包.

- R是完全免费的！！ 而S-Plus尽管是非常优秀的统计分析软件，可是你需要支付一笔$US .
- R可以在运行于UNIX, Windows和Macintosh的操作系统上 .
- R嵌入了一个非常实用的帮助系统.
- R具有很强的作图能力.
- 我们将R程序容易地移植到S-Plus程序中，反之S的许多过程直接或稍作修改用于R.
- 通过R语言的许多内嵌统计函数，很容易学习和掌握R语言的语法.
- 我们可以编制自己的函数来扩展现有的R语言(这就是为什么它在不断等级完善!!)
- .....

网站—R的资源

- **R主页：** http://www.r-project.org
- **CRAN (Comprehensive R Archive Network),** http://cran.r-project.org
- **CRAN的镜像站点** http://cran.r-project.org/mirrors.html
- UCLA提供的关于R与S-Plus的联接，具有搜索功能 http://statcomp.ats.ucla.edu/splus/default.htm
- 李东风主页提供了R的Windows版本 http://cn.math.pku.edu.cn/teachers/lidf/index.html
- 如果使用FTP软件(如Cuteftp)则推荐使用(匿名访问) ftp.u-aizu.ac.jp

统计分析软件包

- CRAN提供了许多便于统计分析的宏包：
  http://cran.r-project.org/src/contrib/PACKAGES.html
  - stable -- 稳妥(分布)广义回归分析
  - tseries – 时间序列分析
  - VaR – 风险值分析
  - matrix – 矩阵运算
  - cinterface – C与R的接口
  - foreign – 读写由S, Minitab, SAS, SPSS, Stata等软件的数据
  - normix – 混合正态分布分析
  - nortest – 正态分布的Anderson-Darling检验
  - MCMCpack – 基于Gibbs抽样的MCMC抽样方法
  - fracdiff – 分数差分模型的极大似然估计

还有很多……

- 参考资料

随软件所附pdf文档(help->manuals),随版本更新:

- W.N. Venables, D.M. Smith and the R DCT: Introduction to R -- Notes on R: A Programming Environment for Data Analysis and Graphics, 2003. http://bayes.math.montana.edu/Rweb/Rnotes/R.html
- R DCT, The R Environment for Statistical Computing and Graphics -- Reference Index,2003.
- R DCT, R Data Import/Export, 2003.
- R DCT, R Language Definition,2003
- R DCT, Writing R Extensions,2003

- Kurt Hornik, R FAQ, Version 1.8-1, 2003-10-07
- B. D. Ripley, R for Windows FAQ, Version for rw1080
- R Html Help, Statistical Data Analysis

其它PDF/HTML文件：

- Kickstarting R, http://cran.r-project.org/doc/contrib/Lemon-kickstart/
- R examples, Alison Gibbs, http://www.utstat.toronto.edu/alisong/Teaching/Winter04/Sta248/Rex.html

- *Ko-Kang Wang, Introduction to R for Statisticians,2004
- J.H. Maindonald, Using R for Data Analysis and Graphics – An Introduction, 2001
- J.H. Maindonald, Using S-PLUS for Data Analysis and Graphics, 2001
- Emmanuel Paradis, R for beginners
- Jonathan Baron, R reference card
- Bret Larget, R for Introductory Statistics, 2002
- W. N. Venables and B. D. Ripley，Modern Applied Statistics with S，4th Ed., 2002
  http://www.stats.ox.ac.uk/pub/MASS4/

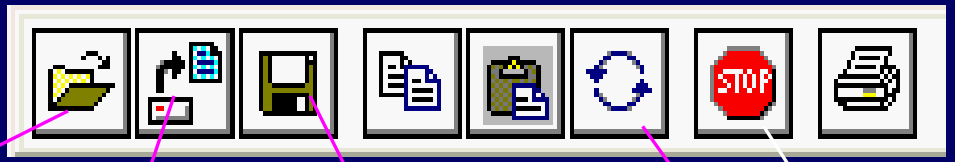
近期畅销书:

- Peter Dalgaard, Introductory Statistics with R, Springer, 2002
- John Maindonald, John Braun, Data Analysis and Graphics Using R -- An Example-based Approach, Cambridge University Press,2003
- John Fox, An R and S-Plus Companion to Applied Regression, Sage Publications, Inc., 2002

R的运行平台-1

- R_GUI
  - 启动R,我们看到R GUI (graphic user's interface)的主窗口, 它由三部分组成
    - 主菜单
    - 工具条
    - R console (R的运行窗口)
  - R console
    - 你的主要工作是在这里通过发布命令来完成的,包括数据集的建立,数据的分析,作图等.
    - 在这里你可以得到在线帮助
      - help.start()　HTML格式的关于R的帮助文件
      - help()　得到相应函数的帮助,例如help(plot)
      - demo()　得到R提供的几个示例
    - q()　退出R
    - 同Matlab类似，用右shift键可以重现以前的命令

- 工具条介绍

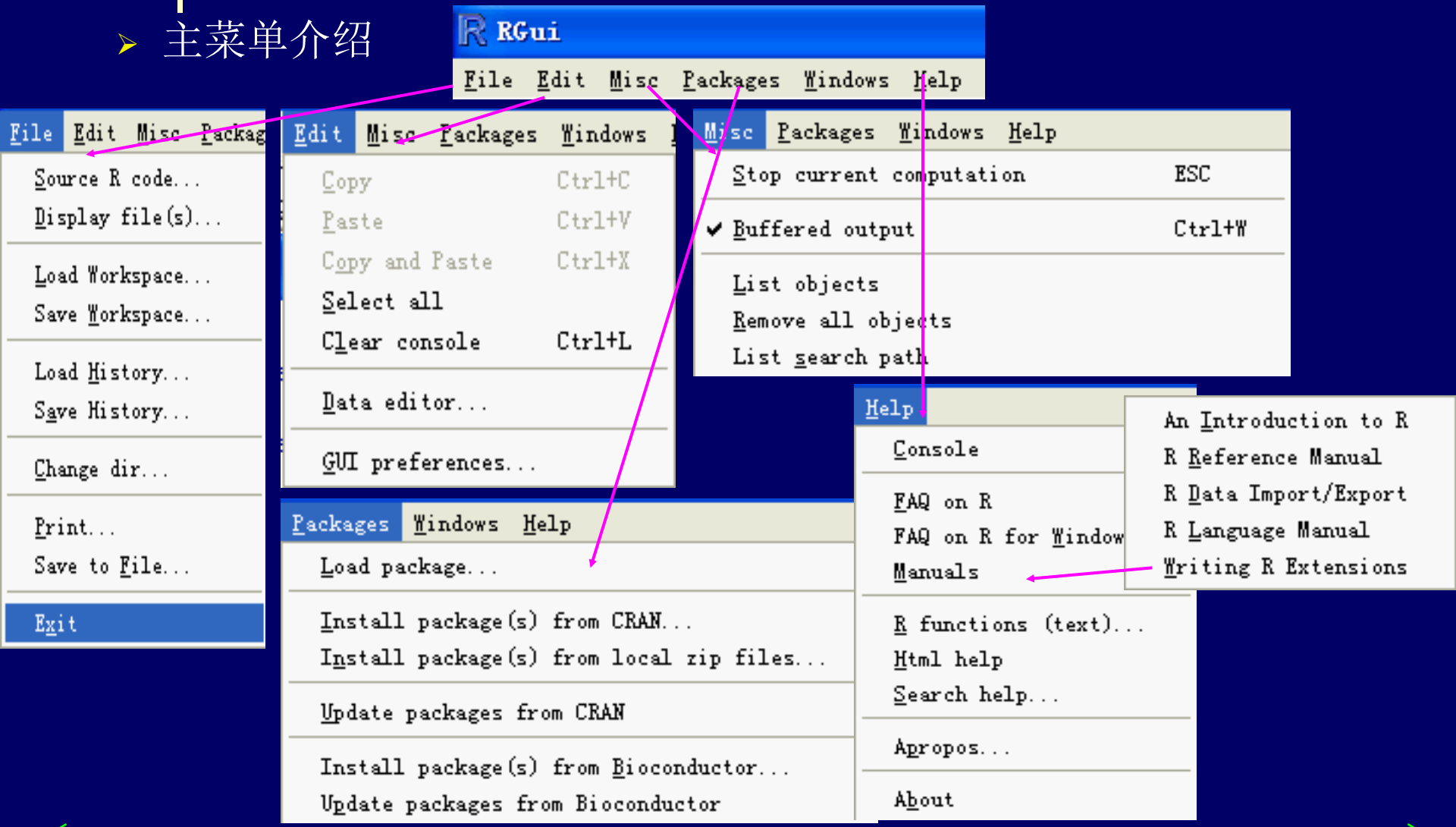


Source R code Load image Save image Copy and Paste

Stop current computation

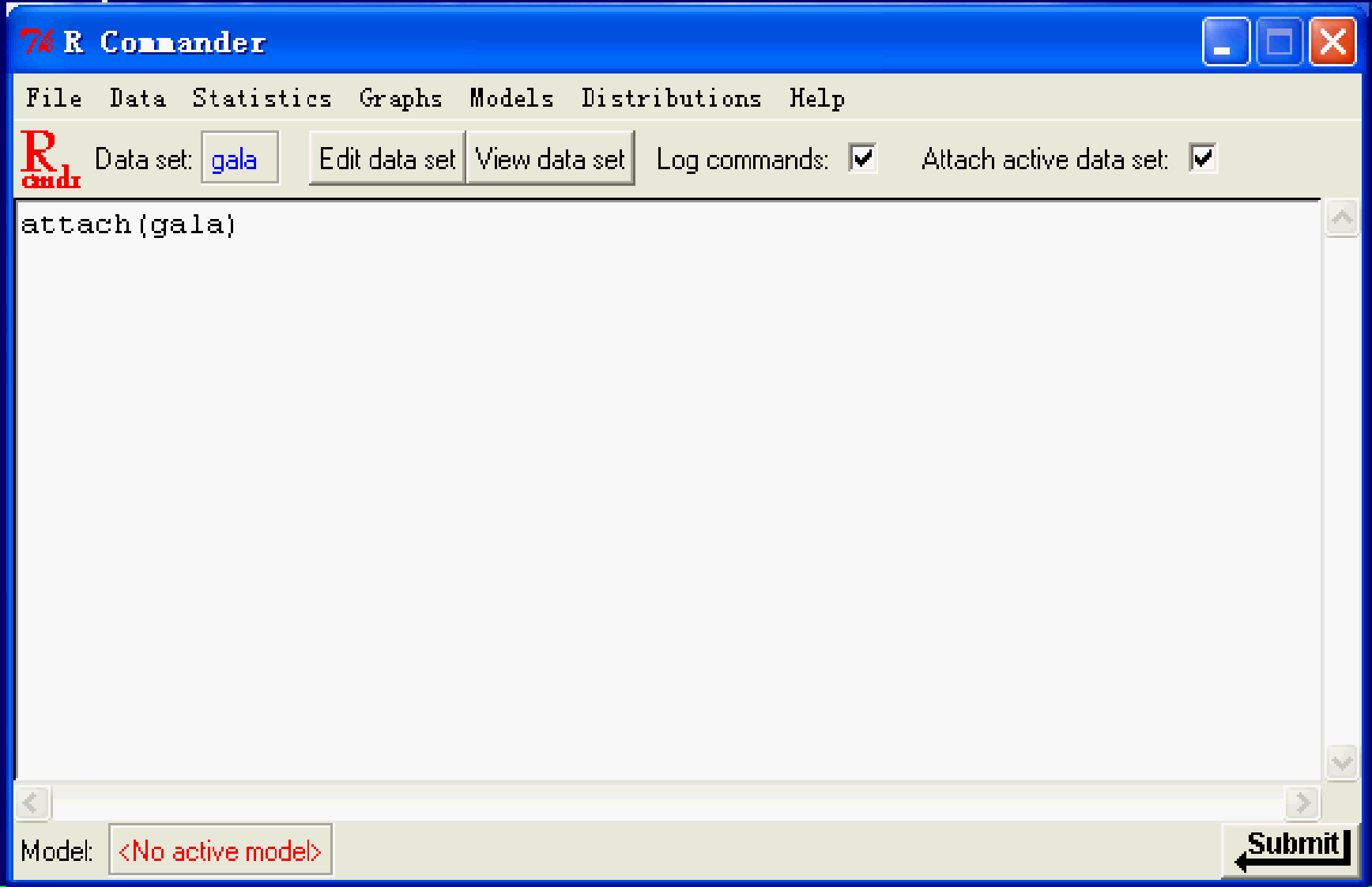
- Source R code　　执行R文件(*.R或*.r)
- Save image　　保存工作空间,文件名为*.RData
- Load image　　打开已有的工作空间
- Stop current computation　　中止当前计算(由于超时等原因)

- 主菜单介绍


RGui
File Edit Misc Packages Windows Help
File Edit Misc Packag
Source R code...
Display file(s)...
Load Workspace...
Save Workspace...
Load History...
Save History...
Change dir...
Print...
Save to File...
Exit
Edit Misc Packages Windows
Copy Ctrl+C
Paste Ctrl+V
Copy and Paste Ctrl+X
Select all
Clear console Ctrl+L
Data editor...
GUI preferences...
Misc Packages Windows Help
Stop current computation ESC
Buffered output Ctrl+W
List objects
Remove all objects
List search path
Packages Windows Help
Load package...
Install package(s) from CRAN...
Install package(s) from local zip files...
Update packages from CRAN
Install package(s) from Bioconductor...
Update packages from Bioconductor
Help
Console
FAQ on R
FAQ on R for Window
Manuals
R functions (text)...
Html help
Search help...
Apropos...
About
An Introduction to R
R Reference Manual
R Data Import/Export
R Language Manual
Writing R Extensions

R的运行平台-2

- R_Commander (帮助文件)
  - 作者: John Fox (jfox@mcmaster.ca)
  - 版本：Rcmdr Version 0.9-0
  - 安装
    - Rcmdr是R的一个宏包，它还需要宏包car的支持
    - 在加载Rcmdr宏包之前，需要先安装宏包car
    - Rcmdr仅在单文件RGui (SDI)下工作，这可通过RGui的Edit=>GUI preferences…进行设置(注：好象均可?)
  - 运行
    - 安装宏包car
    - 在RGui下通过Packages=>Load packages…加载Rcmdr宏包

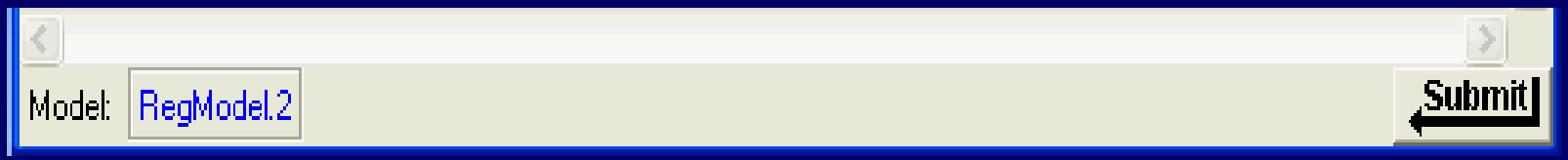

R Commander
File Data Statistics Graphs Models Distributions Help
Data set: gala
Edit data set
View data set
Log commands:
Attach active data set:
attach(gala)
Model: <No active model>
Submit

- 功能
  - R Commander是一个交互式菜单/对话框系统(menu/dialog-box interfaces)，用于进行数据的读、写、转换及常用的统计分析．作者还添加了线性与广义线性模型等统计分析工具．
- 结构与使用(具体见Rcmdr的Getting Started部分)
  - R Commander窗口由一些菜单及按钮组成．
  - 菜单的下方是一个log/script窗口
  - 通过菜单的对话框将命令发送到RGUI中，以完成某一项统计分析．这些命令同时在log窗口中显示出来．它们可以被重新编辑修改，并可通过Rcmdr窗口右下角的Submit按钮再一次发送给R执行
  - 命令指向一个当前的或者活动的数据集.一旦读入一个新的数据集，它就是活动的．注：通过Data=>Import data加载数据集，其名字及路径不能含有中文,而中R GUI中是允许的!!

- R Commander的菜单

R Commander 菜单树(Menu Tree) (点击)

- R Commander的信息反馈(information field)



R的运行平台-3

- R_WinEdt – 安装与使用
  - 作者: Uwe Ligges (ligges@statistik.uni-dortmund.de)
  - 版本：RWinEdt Version 1.6.0
  - 必备软件
    - R: http://cran.r-project.org/
    - WinEdt5.x: http://www.winedt.com/ (或CTeX2.3.x)
    - R-WinEdt: http://cran.r-project.org/contrib/extra/winedt/
  - R_WinEdt的安装
    - 将R_WinEdt压缩软件释放到WinEdt的plugins\R-WinEdt目录下
    - 双击install.exe文件进行进行安装,由此会在WinEdt目录下自动生成R.ini(此目录还有winedt.exe!)
    - 在此我们假定大家已安装了CTeX2.3.x,并进行缺省安装， 也即WinEdt在 C:\CTeX\WinEdt 目录下

- R_WinEdt的设置
  - 设置桌面快击键:

    为了同时可以使用LaTeX/CTeX和R_WinEdt，在桌面上复制一个WinEdt快击键，并命名为R_WinEdt;
  - 右击R_WinEdt快击键，选择属性，然后在“目标(T)”中输入

    C:\CTeX\WinEdt\WinEdt -C="R-WinEdt" -e=r.ini
  - (并不必须)如果每次运行R都想加载某个/些软件包或函数，则可对R主目录下etc下的.Rprofile进行修改,如每次加载simple宏包就可使用下面的设置

```
library(simple)
options(editor="\"c:/program files/winedt/winedt\"
-c=\"R-WinEdt-edit\" -e=r.ini -V")
```

R_WinEdt 界面

R-Editor - [C:\R\eigenes\scatterplot3d\R\scatterplot3d.R]

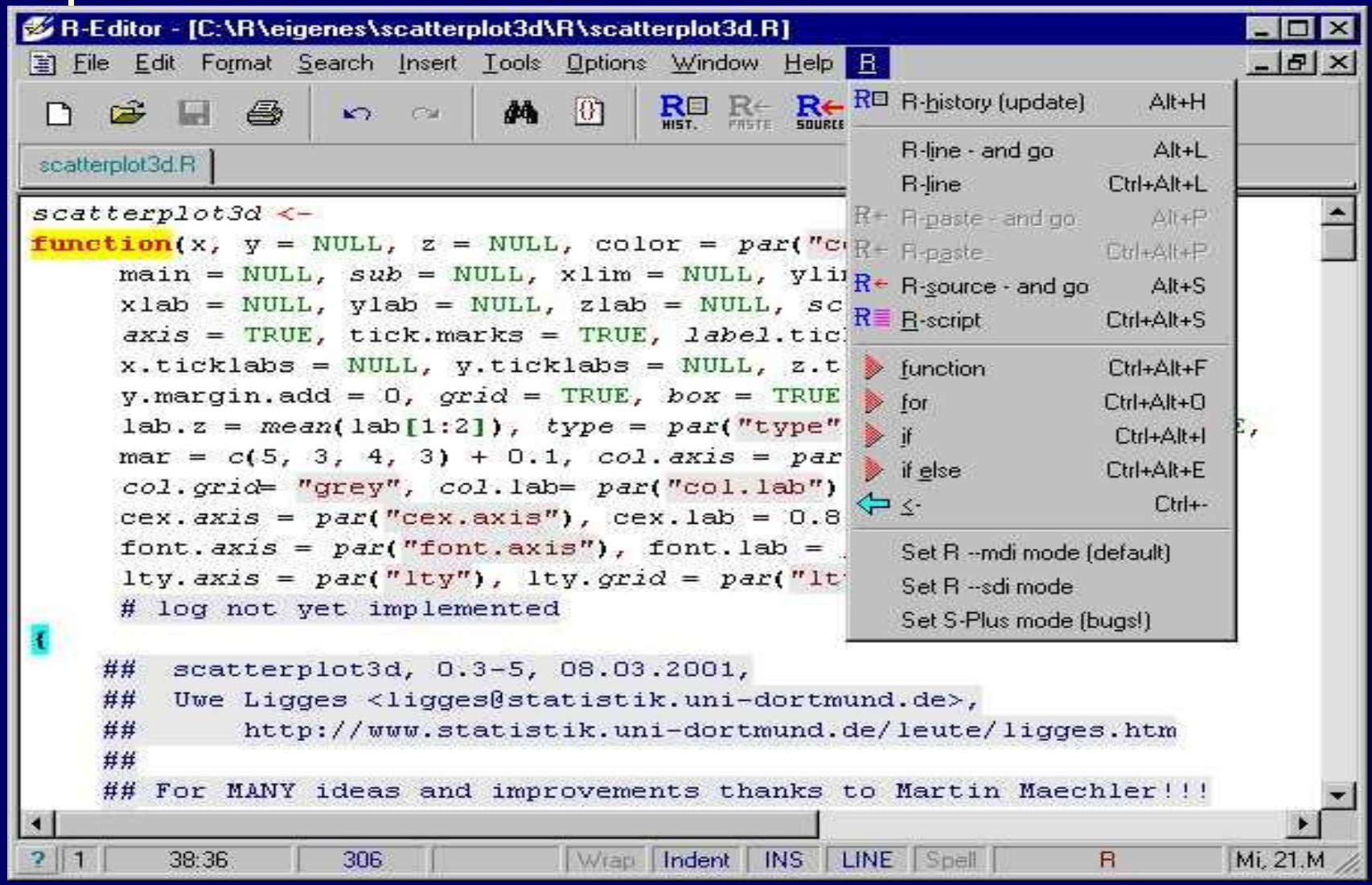
File Edit Format Search Insert Tools Options Window Help R

scatterplot3d.R

```
    mem.par <- par(mar = mar)
    x.scal <- y.scal <- z.scal <- 1
    xlabel <- if (!missing(x)) deparse(substitute(x))
    ylabel <- if (!missing(y)) deparse(substitute(y))
    zlabel <- if (!missing(z)) deparse(substitute(z))
    ## verification, init, ...
    if(highlight.3d && !missing(color))
        warning(message = "color is ignored when  highlight.3d = TRUE")
    if(length(x) < 2) stop("Minimal required length of x is 2 !")

    ## color as part of 'x' (data.frame or list):
    if(!is.null(d <- dim(x)) && (length(d) == 2) && (d[2] >= 4))
        color <- x[,4]
    else if(is.list(x) && !is.null(x$color))
        color <- x$color

    ## convert 'anything' -> vector
    xyz <- xyz.coords(x=x, y=y, z=z, xlab=xlabel, ylab=ylabel, zlab=zlabel,
                      log=log)
    if(is.null(xlab)) { xlab <- xyz$xlab; if(is.null(xlab)) xlab <- "" }
    if(is.null(ylab)) { ylab <- xyz$ylab; if(is.null(ylab)) ylab <- "" }
    if(is.null(zlab)) { zlab <- xyz$zlab; if(is.null(zlab)) zlab <- "" }

    if(length(color) == 1)
        color <- rep(color, length(xyz$x))
```

? 1 52:1 306 Wrap Indent INS LINE Spell R Mi, 21.Mrz.2001 15:12

R_WinEdt 菜单


R-Editor - [C:\R\eigenes\scatterplot3d\R\scatterplot3d.R]
File Edit Format Search Insert Tools Options Window Help R
scatterplot3d.R
R-history (update) Alt+H
R-line - and go Alt+L
R-line Ctrl+Alt+L
R-paste - and go Alt+P
R-paste Ctrl+Alt+P
R-source - and go Alt+S
R-script Ctrl+Alt+S
function Ctrl+Alt+F
for Ctrl+Alt+O
if Ctrl+Alt+I
if else Ctrl+Alt+E
<- Ctrl+-
Set R --mdi mode (default)
Set R --sdi mode
Set S-Plus mode (bugs!)
scatterplot3d <-
function(x, y = NULL, z = NULL, color = par("c
main = NULL, sub = NULL, xlim = NULL, ylim
xlab = NULL, ylab = NULL, zlab = NULL, sc
axis = TRUE, tick.marks = TRUE, label.tic
x.ticklabs = NULL, y.ticklabs = NULL, z.t
y.margin.add = 0, grid = TRUE, box = TRUE
lab.z = mean(lab[1:2]), type = par("type"
mar = c(5, 3, 4, 3) + 0.1, col.axis = par
col.grid= "grey", col.lab= par("col.lab")
cex.axis = par("cex.axis"), cex.lab = 0.8
font.axis = par("font.axis"), font.lab =
lty.axis = par("lty"), lty.grid = par("lt
# log not yet implemented
{
scatterplot3d, 0.3-5, 08.03.2001,
Uwe Ligges <ligges@statistik.uni-dortmund.de>,
http://www.statistik.uni-dortmund.de/leute/ligges.htm
##
For MANY ideas and improvements thanks to Martin Maechler!!!
38:36
306
Wrap Indent INS LINE Spell R
Mi, 21.M

- R_WinEdt的特点
  - - 与RGui共同运行
  - - 具有WinEdt的强大功能 (如Delimiter检查, 高级搜索, 书签, 宏, 缩进与注释的对齐等)
  - - 语法高亮显示(Syntax-Highlighting)
  - - 同时可以编辑多个R程序
  - - 设置简单快速的按钮与快击键
  - - 将窗口中的代码(script)发送到R中运行
  - - WinEdt中选中的代码(code)发送到R中运行
  - - 单行代码(script)发送到R中运行
  - - 及时更新历史命令记录文件 .Rhistory，以便重复使用旧的命令
  - - 提供衫的模块,如: for(_ in _){_}

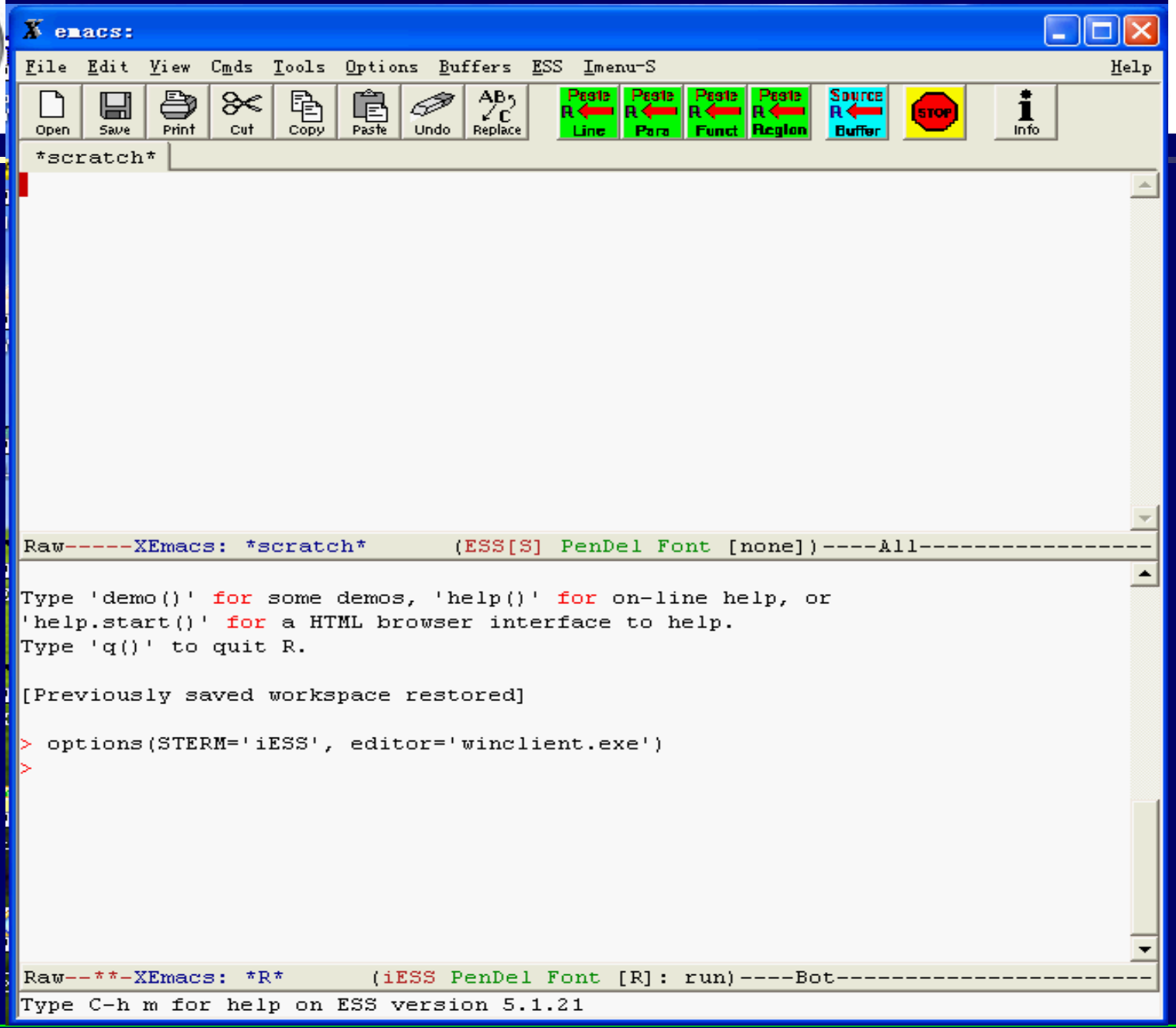
R_WinEdt – 菜单与热键

# Command	Hot Key	Menu Icon	说明
# Brackets Check	Ctrl+F12	{}	括号配对检查
# R History	ALT+H	R HIST.	保存历史记录
# R-line - and go	ALT+L		单行发送
# R-line	Ctrl+ALT+L		单行发送并返回
# R<- R-paste - and go	ALT+P	R<-PASTE	选中后发送
# R<- R-paste	Ctrl+ALT+P		选中后发送并返回
# R<- R-source - and go	ALT+S	R<-SOURCE	R文件发送(先打开)
# R<- R-script	Ctrl+ALT+S	R SCRIPT	R文件发送并返回
# function	Ctrl+Alt+F		生成函数框架
# for	Ctrl+Alt+O		生成for循环框架
# if	Ctrl+Alt+I		生成if框架
# ifelse	Ctrl+Alt+E		生成ifesle框架
# <-	Ctrl+-		生成赋值符号

R的运行平台-4

- R_ESS+XEmacs – 安装与使用 (作者: John Fox)
  - 关于ESS与XEmacs
    - Emacs是一个功能强大广为使用的编程器,可以进行配对检查,语法高亮显示,查错(debugging)等.

      For some Unix/Linux users, Emacs is more a way of life than an editor.
    - ESS(Emacs Speaks Statistics)可以为许多统计软件定制相应的编程环境,包括R, S-Pplus, SAS, Stata, Lisp-Stat等.
    - Emacs主要有二个版本: GNU Emacs 和 XEmacs.
    - 由于使用了ESS,因此多个统计分析软件可并存于同一平台.
    - John Fox的ESS设置提供了一个主窗口,上下二个子窗口: 上面的用于输入R的源文件/代码,下面的用于显示这些R程序/代码在XEmacs运行后的输入与输出结果.(点击此处查看)
    - XEmacs和ESS均是免费的!


emacs:
File Edit View Cmds Tools Options Buffers ESS Imenu-S Help
Open Save Print Cut Copy Paste Undo Replace
Paste R Line
Paste R Para
Paste R Funct
Paste R Region
Source R Buffer
STOP
Info
scratch
Raw-----XEmacs: *scratch* (ESS[S] PenDel Font [none])----All----------
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.
[Previously saved workspace restored]
> options(STERM='iESS', editor='winclient.exe')
>
Raw--**-XEmacs: *R* (iESS PenDel Font [R]: run)----Bot----------
Type C-h m for help on ESS version 5.1.21

- 必备软件与安装
  - R: http://cran.r-project.org/
  - Xmacs with Ess: http://www.xemacs.org/Download/win32/ 先下载Xemacs的安装器(installer) setup.exe并运行，接着进行net安装或下载Xemacs及必要的宏包。必要的宏包包括
    - ess
    - dired
    - ediff
    - edit-utils
    - efs
    - fsf-compat
    - mail-lib
    - pc
    - speedbar
    - vc
    - xemacs-base
    - xemacs-i586-pcwin32
  - 若你已下载了上述宏包，则可以选择本地安装(local directory)

XEmacs Setup

Select packages to install

Prev | Curr | Exp | Full/Part

Current	New	Src?	Package
	1.13	n/a	Sun
	1.10	n/a	ada
	1.20	n/a	apel
	1.24	n/a	auctex
	1.12	n/a	bbdb
	1.16	n/a	c-support
	1.14	n/a	calc
	1.16	n/a	calendar
	1.23	n/a	cc-mode
	1.13	n/a	cookie
	1.12	n/a	crisp
	1.14	n/a	debug
	1.09	n/a	dired

= click to choose action, (p) = previous version, (x) = experimental

去掉宏包, 点一下New一栏下对应的版本号; 此版本号即被Skip所代替.

< Back | Next > | Cancel

- R_ESS+XEmacs的设置
  - 我们假定R与XEmacs的目录分别为
    - R: C:\Program Files\R\rwxxxx\ (xxxx为版本号)
    - Xemacs: C:\Program Files\Xemacs
  - 检查系统的home目录(WinXP/2000/NT下可在DOS方式下用命令 set HOME 检查. 在此假定为c:\
  - 在home目录下建立子目录(用DOS命令mkdir) .xemacs
  - 从John Fox的主页下载设置文件fox-ess-config.zip，地址为：http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/ESS/index.html
  - 将解压文件init.el改名为Rinit.el并复制到 .xemacs下

- 将解压文件function.xpm, line.xpm, para.xpm, region.xpm, source.xpm和stop.xpm复制到目录

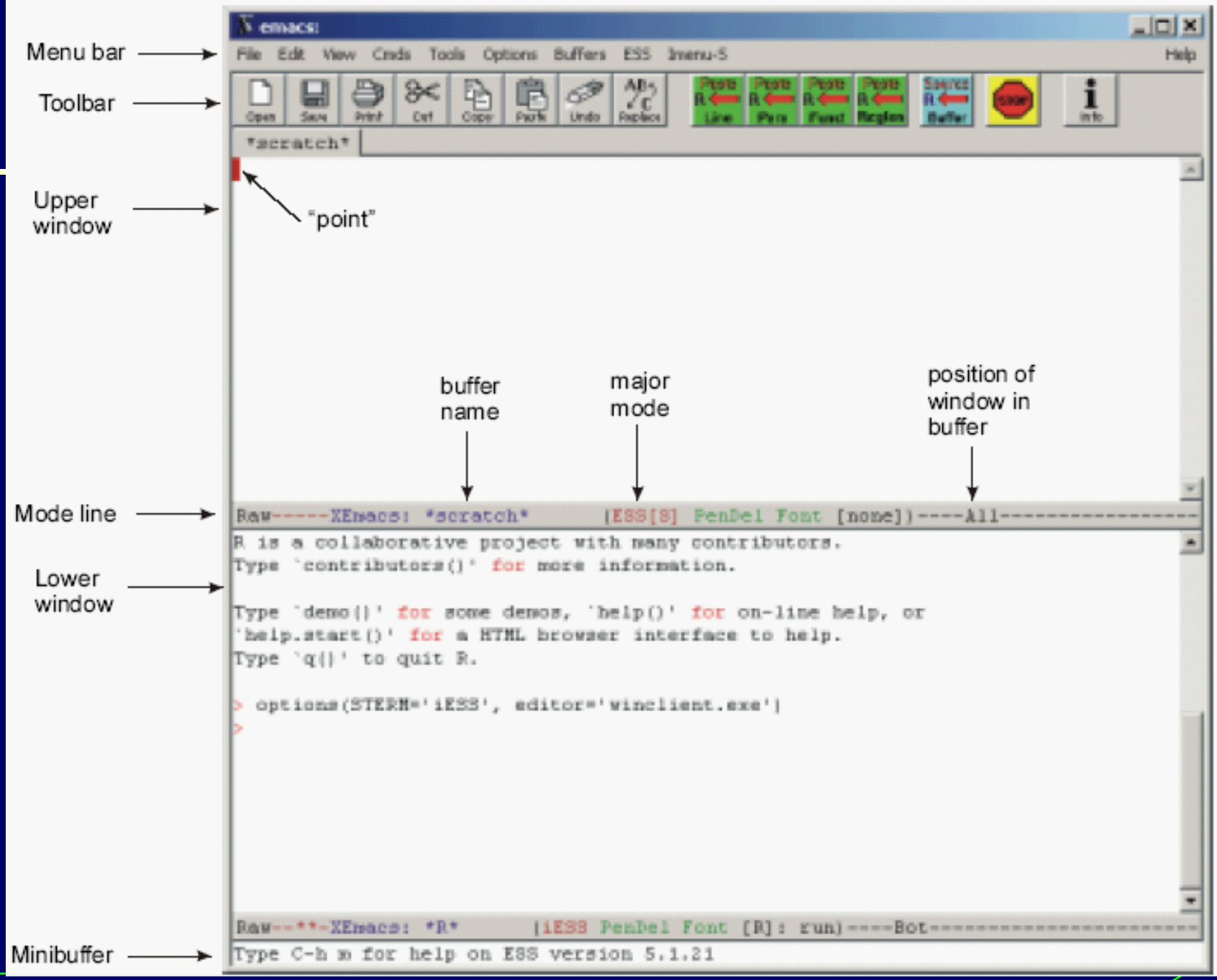
  c:\Program Files\Xemacs\XEmacs-xx.y.z\etc\toolbar\

  (xx.y.z为Xemacs的版本号),由此在Xemacs中建立R的工具条.
- (若没有)添加系统搜索路径: c:\Program Files\R\rwxxxx\bin

  (在WinXP/2000/NT中可由控制面板=>系统=>高级=>环境变量 中新建/添加).

  另一方法： 用纯文本修改文件Rini.el中的关于R的搜索，即 (setq-default inferior-R-program-name “c:/Program Files/rwxxxx/bin/rterm.exe”) ----也即只要去掉之前的分号(;).
- 修改桌面图标XEmacs的属性: 在快捷方式的”目标”下添加 -q -l “c:\.xemacs\Rinit.el“, 在”起始位置”添加存放文件的目录.

- ESS+XEmacs下R的使用 ---了解XEmacs视窗
  - 不是问题的问题:
    - 每次打开R_XEmacs可能会弹出信息 “Initialization complete”, 只要点击OK键就行了.
    - 安装后初次使用会在屏幕的底部(“minibuffer”中)出现当前窗口无法打开 .Rhistory 的信息. 此记录历史命令的文件会在正常退出R与ESS时建立(但只在下面窗口>提示符后直接输入一个或多个命令后!)


Menu bar
Toolbar
Upper window
"point"
buffer name
major mode
position of window in buffer
Mode line
Lower window
Minibuffer
emacs
File Edit View Cmds Tools Options Buffers ESS Inemu-S Help
scratch
Raw-----XEmacs: *scratch* (ESS[S] PenDel Font [none])----All----------
R is a collaborative project with many contributors.
Type 'contributors()' for more information.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.
> options(STERM='iESS', editor='winclient.exe')
>
Raw--**-XEmacs: *R* (iESS PenDel Font [R]: run)----Bot----------
Type C-h m for help on ESS version 5.1.21

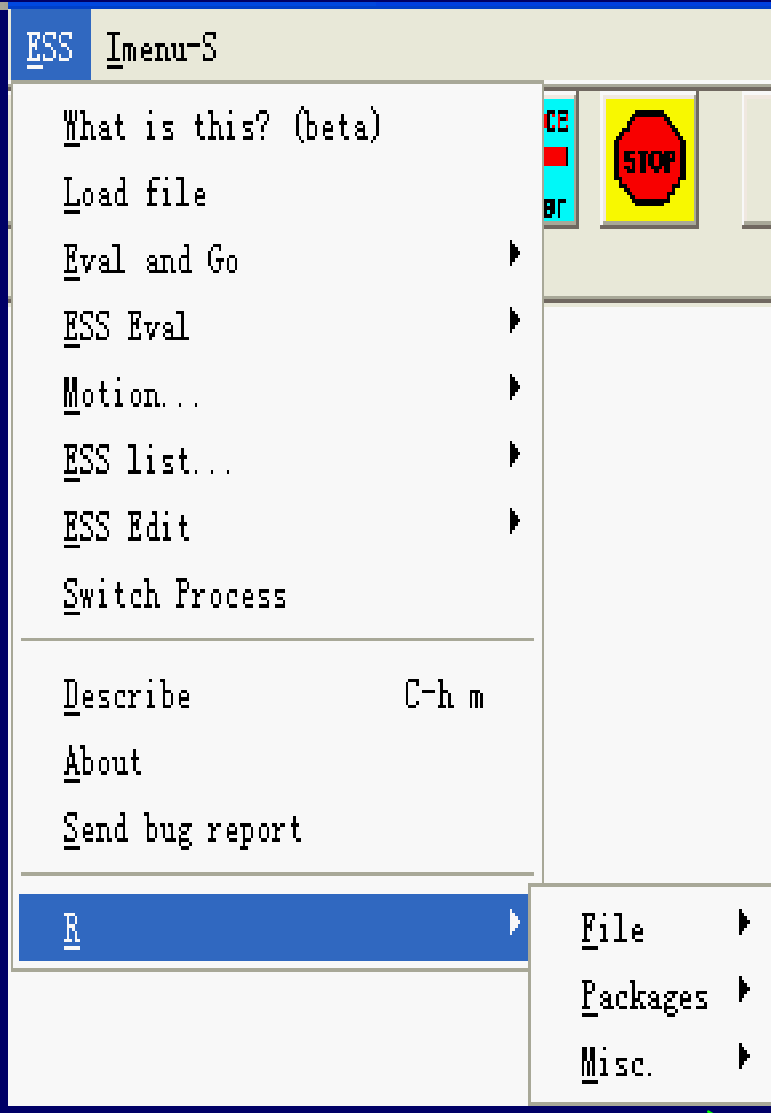
- 视窗介绍
  - John Fox将Xemacs主窗口(实际上称为frame)分为二个子窗口,上下分开(见上图所示).
  - 在Xemacs主窗口的上方还有一个菜单条(menu bar)和一个工具条(toolbar),其内容会随缓冲区(buffer)主模式(major mode)而变化.你可以通过光标在上下子窗口中转换观察菜单与工具条的变化.
  - J. Fox为R提供了专门的工具条与菜单,非常便于使用(后面介绍).
  - 上面的子窗口显示*scratch*缓冲区,其中你可以输入R命令.顾名思义, *scratch*缓冲并非永久性的,当然你可以将其内容保存在文件中.
  - 下面的子窗口显示R经Emacs运行的结果,在此缓冲区中你也可以直接输入R命令.

- 每一子窗口下均有一个模式行(mode line),显示的缓冲的状态,其中包括:1)缓冲的名字 2)缓冲的主模式(或次模式) 3)窗口在缓冲中的位置.
  - 上面的子窗口包含*scratch* buffer,处于ESS[S]主模式(主模式由文件的扩展名所决定. S指S语言,当打开扩展名为 .r, .R, .s, .S, .q的S语言源代码时,均处于ESS[S]主模式中),显示所有(ALL)缓冲内容—现在是空的.
  - 下面的子窗口包含*R* buffer, 处在iESS (inferior ESS – 之所以你它是inferior,因为它是受Emacs控制的!), 目前处在缓冲的底部(Bot). 你可以滑动游标观察其位置的变化.
- 指针(point): Emacs的光标,它与通常的光标还是有所差异,不过John Fox在ESS模式下按Windows的习惯进行了修改.
- 在窗口的下方是一个单行的小窗口,称为minibuffer(小缓冲),主要用于信息显示,其次若你要输入Emacs命令,它们会显示在minibuffer中,最后一命令(如搜索和替换)的响应在此完成.

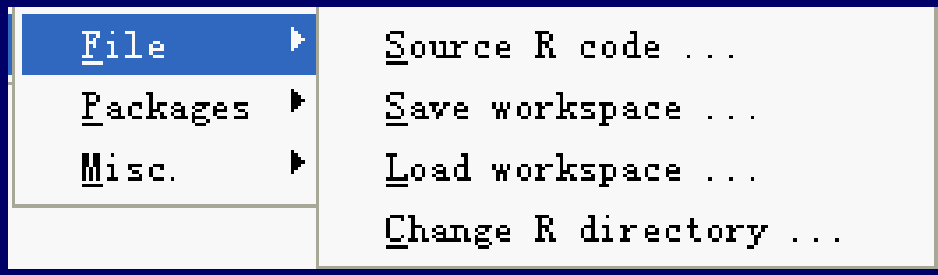
- ESS+XEmacs下R的使用
  --- R_XEmacs视窗

  - Paste to R(粘贴至R)按钮 --- 将上面子窗口中的命令送到下面子窗口中执行
    - Paste line　　执行光标所在行
    - Paste paragraph　　执行光标所在段落(其中不含空行)
    - Paste function　　执行已定义的光标所在的函数
    - Paste region　　执行用鼠标标记的区域
    - Source to R　　执行上面子窗口(缓冲区)中的所有命令
      - 对于R源文件在执行之前会重新保存更新的代码
      - 对于*scratch* buffer中的代码保存后才能执行
      - (不同于前的几种方式)R文件中的源代码的执行过程不在下面的子窗口中显示出来! 注意:首次使用会在minibuffer中出现Process to load into: R,按回车键继续
    - Stop　　中止当前计算(如由于超时)

- R子菜单
  - 如果光标/指针须在R的源文件中,则会出现ESS菜单,进行ESS会出现John Fox特地为R设计的R子菜单.
  - R子菜单本身包括三个子菜单
    - File 文件
    - Packages 宏包
    - Misc 其它
  - 它们的功能与R GUI的类似

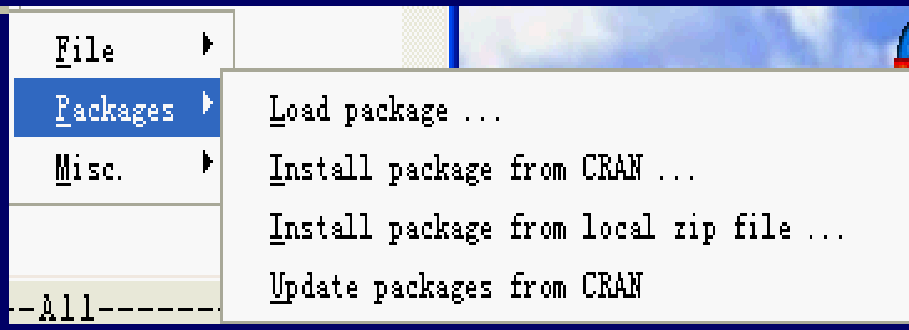


- ESS=>R=>File菜单



- Source R code… 打开对话框以选择源文件
- Save workspace… 将R的工作空间存入文件
- Load workspace… 加载保存的工作空间
- Change R directory… 改变工作空间(指R运行过程空间,并非指上面子窗口的源文件目录)

- ESS=>R=>packages菜单



- Load package…
  加载R宏包,纳入搜索目录
- Install package from CRAN…
  从CRAN处选择．下载．安装宏包(需要internet连接！)
- Install package from local zip file…
  从本地计算机或网络宏包(zip压缩文件)
- Update packages from CRAN…
  从CRAN处更新宏包 – 搜索所有安装宏包的新版本,下载并安装

- ESS=>R=>Misc菜单

  - Remove all objects… 删除R工件空间中所有的对象
  - List objects… 列出R工作空间中的对象
  - Display path… 列出R的搜索路径

- File=>Exit R/Xemacs菜单

  - Quit saving R workspace 在当前的R目录中保存工作空间
  - Just quit 不保存R工作空间直接退出R,系统会提醒你是否真的不保存修改的缓冲

R的语法与数据结构

- 语法
  - 符号
    - > 命令或运算提示符
    - + 续行符
  - 基本算术运算
    - + 加号
    - - 减号
    - * 乘号
    - / 除号
    - ^ 乘方
  - 赋值符
    - = 或 <-

- 求助符
  - ?
  - help()

例子：

```
>3+5
>3-5
>3/5
>3^5
>x=5
>?plot
>help(plot)
```

- 向量
  - 向量是R中最为基本的类型
  - 一个向量中元素的类型必须相同，包括
    - 数值型
      - 整型
      - 单精度实型
      - 双精度实型
    - 逻辑型
    - 复值型
    - 字符型

- 建立向量的方法(函数)
  - seq() 或 :　　若向量(序列)具有较为简单的规律
  - rep()　　若向量(序列)具有较为复杂的规律
  - c()　　若向量(序列)没有什么规律

例子：

```
>1:10
>seq(1,10,by=0.5)
>seq(1,10,length=21)
>rep(2:5,2)
>rep(2:5,rep(2,4))
>x=c(42,7,64,9)
>length(x)
```

- 向量运算中的循环法则(recycling rule)

```
>1:2+1:4
```

$$\begin{bmatrix}1\\2\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\end{bmatrix}=\begin{bmatrix}1\\2\\1\\2\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\end{bmatrix}=\begin{bmatrix}2\\4\\4\\6\end{bmatrix}$$

```
>1:4+1:7
```

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\end{bmatrix}=\begin{bmatrix}1\\2\\3\\4\\1\\2\\3\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\end{bmatrix}=\begin{bmatrix}2\\4\\6\\8\\6\\8\\10\end{bmatrix}$$

- 向量的下标(index)与向量子集(元素)的提取
  - 正的下标　　提取向量中对应的元素
  - 负的下标　　去掉向量中对应的元素
  - 逻辑运算　　提出向量中元素的值满足条件的元素

  注：R中向量的下标从1开始，这与通常的统计或数学软件一致而象C语言等计算机高级语言的向量下标则从0开始！

例子：

```
>x = c(42,7,64,9)
>x[1]
>x[-2]
>x[c(1,4)]
```

```
>x>10                    #值大于10的元素逻辑值
 [1] TRUE  FALSE  TRUE  FALSE
>x[x>10]                 #值大于10的元素
 [1] 42  64
>x[x<40&x>10]
> #产生(0,1)上100个均匀分布随机数
>y = runif(100, min = 0, max = 1)
>sum(y<0.5)          #值小于0.5的元素的个数
 [1] 47
>sum(y[y<0.5])     # 值小于0.5的元素的值的和
 [1] 10.84767
```

- 数据框架(data frame)
  - 许多数据集是数据框架的形式出现
  - 一个数据框架就是将许多向量组合起来的一个对象，它是二维的，通常其列表示变量，其行表示观测。
  - 建立数据框架的方法(函数)
    - 若你的数据本身保存在一个文件中，则可以使用
      - read.table()　仅接受带有分界符的ASCII数据

        如果数据是电子报表的形式，则采用下面的两种变型
      - read.csv()　先将数据另存为带逗号的数据(Comma Seperated values)
      - read.delim()　先将数据另存为用tab作为分界符的数据

      注：若数据集很大(如1,000,000观测x200变量)，则可由ODBC联接由数据库读入.

- 若你在R中建立了一些向量并试图想由它们生成框架，则可以使用data.frame()，但需要同时cbind().

例子：

```
>x=c(42,7,64,9)
>y=1:4
>z.df=data.frame(cbind(INDEX = y, VALUE = x))
>z.df
  INDEX VALUE
1     1    42
2     2     7
3     3    64
4     4     9
```

注 ☹ .df 只是为了提醒自己z.df是一个数据框架

☹ INDEX和VALUE是重新命名的向量名字

- 数据框架子集的提出取
  - 格式　foo[row, column]
    其中　foo　　数据框架的名称
    row　　需要提出取的行号
    column　　需要提出取的行号
  - 例子(续上一例)

```
> z.df[1,]                # 提取第一个观测(第一行)
  INDEX VALUE
  1     1    42
> z.df[,1]                #提取第一列(第一个变量的值)
[1] 1 2 3 4
> z.df[2,1]               #提取第二行第一列的元素
[1] 2
```

- 列表(list)
  - 复杂的数据分析时，仅有向量与数据框架还不够
  - 有时需要生成包含不同类型的对象
  - R的列表(list)就是包含任何类型的对象

例子:

```
>foo = list(x = 1:6, y = matrix(1:4, nrow = 2))
>foo
 $x
 [1] 1 2 3 4 5 6
 $y
      [,1] [,2]
 [1,]    1    3
 [2,]    2    4
```

- 列表子集的提出取
  - 提取一个子对象如foo的x,下面三种方式等价
    > foo$x
    >foo[1]
    >foo[[1]]
    例子

>foo$y

>foo[2]

>foo[[2]]

>foo[[1]][2]

>foo$y[2]

>foo$y[4]

- 条件语句
  - 作用: 避免除零或负数的对数等数学问题
  - 形式1:
    if (条件) 表达式1 else 表达式2
  - 形式2 – 常优于形式1!
    ifelse(条件, yes, no)

试比较下面的三个结果:

```
>x = c(6:-4)
> sqrt(ifelse(x >= 0, x, NA))
> ifelse(x >= 0, sqrt(x), NA)
>if (x >= 0) sqrt(x) else NA
```

- 循环(loops)
  - for()　　#若知道终止条件
    for (变量 in 向量) 表达式
  - while()　　#若无法知道运行次数
    while(条件) 表达式
  - 两者通常可以转换
    例1—试比较两种方法
    >for (i in 1:5) print (1:i)
    >i=1
    >while(i <= 5) {
    + print(1:i)
    + i = i+1
    + }

例2 – 见Ko-Kang Wang's "R Programming Workshop", pp6-8

- Suppose we generate a pseudo DNA microarray and we want to do an ANOVA on it. First we generate some factors for Array (a), Treatments (t) and Genes (g). Then generate some normal random numbers for the logged foreground intensity. Then we put into an aov() function for each gene – this is where the loop is good for. Note that you will get different answer when you try it, because of the random numbers generated.

程序如下(使用for循环，也可改用while循环):DNA_anova.R

注：R控制面板中显示符号>和+,而源程序中是不需要的！

```
> n = 3044
> a = c(rep(1, 2 * n), rep(2, 2 * n))
> t = c(rep(1, n), rep(2, n), rep(2, n), rep(1, n))
> g = rep(rep(1:1522, rep(2, 1522)), 4)
> y = rnorm(4 * n, mean = 4.13, sd = 0.75)
> ybar = data.frame(A = factor(a), G = factor(g),
+                   T = factor(t), Intensity = y)
> attach(ybar)
> ybar[1:10,]    #  查看ybar的前10行
> res.mat = matrix(0, 1522, 8, byrow = TRUE)
> coef.mat = matrix(0, 1522, 4, byrow = TRUE)
> for(i in 1:1522) {
+       gene.aov = aov(Intensity ~ A + T + A * T,
+       sub = G == i)
+       res.mat[i, ] = residuals(gene.aov)    # 保存ANOVA分析的残差
+       coef.mat[i, ] = coef(gene.aov)        # 保存ANOVA分析的方差系数
+ }
> res.mat[1:10, ]                  #查看残差的前10行
> coef.mat[1:10, ]                 #查看方差系数的前10行
```

- 向量化(vectorization)
  - 循环(loops)很有用,但如果能将一组命令向量化，则应尽量避免循环，原因在于
    - C是一种编译语言，其效率是很高的；R则是一种解释语言。在计算时，通常C要比R快100倍。
    - 在R中充分使用向量化，因为R会立即调用C进行运算，因而大大提高计算的效率！
  - 例子 ---见Ko-Kang Wang's "R Programming Workshop", pp9-11 (Gamma函数作图)
    - Vector_1.R　没有使用向量和循环的源程序
    - Vector_2.R　使用for循环改进后的源程序
    - Vector_3.R　使用向量化后源程序

- 函数
  - 函数是一系列语句的组合，在R中可以写出自己的函数
  - 形式: 变量名 = function(变量列表) 函数体
  - 函数引用: 变量名(变量的值)
  - 函数可以递归引用，但不提倡!
  - 例子 – 使用gamma函数求n!

```
>factorial = function(n) {
 + if (n>=0) gamma(n+1)
 + else print("Please input a positive integer!")
+ }
>factorial(6)
>factorial(-6)
```

- 用于处理错误的函数 – 用于处理用户输入不正确的类型而可能出现的错误
  - warning()　　若错误不严重以至影响整个计算
  - stop()　　若错误可能导致计算中止
  - print()　　显示必要的信息
  - formatC()　　数值作为字符串输出
  - cat()　　字符串联，可以插入\n(换行)及\t(tab键)
  - paste()　　字符粘贴(非字符型自动转换)

例子：

```
>cat("R", "is","a good", "software.\n")
>formatC(1/3, format = "f", digits = 4)
> formatC(1/3, format = "e", digits = 4)
>paste(1:12)  # 与as.character(1:12)等价
>paste("A", 1:6, sep = "")
>paste("today is", date())
```

R中的编程

- 编程的重要性
  - 一个统计软件(包)应该包罗万象—所有统计方法或技术，可以完成所有的任务？？
  - 这是一种错误的观点！！
    - 太贵！！
    - 费时！！
    - SAS是一个世界上最为优秀的统计分析软件之一，但……！！当然SAS具体编程能力，而SPSS更糟！
    - 通过编程可以实现(开发)满足自己需要的函数或宏包

- 好的编程习惯
  - 为了他人，更为你本人! 你的程序应该具有
    - 可读性(readability)
    - 可理解性(understandability)
  - 习惯之一：行前缩进(Indentation),在此推荐使用软件
    - WinEdt => 已开发了基于WinEdt的R使用平台
    - (X)Emacs =>已开发了基于(X)Emacs的R使用平台
    - UltraEdit
  - 习惯之二：增加注释(Commenting),它是你的帮助

    R中使用#作为注释语句的开始.
  - 习惯之三：变量的命名,使用意义明确的名字，切忌使用人或宠物的名字

例子 – 缩进

```
> for(i in 1:1522) {
+             gene.aov <- aov(Intensity ~ A + T + A * T,
+     sub = G == i)
+     res.mat[i, ] <- residuals(gene.aov)
+     coef.mat[i, ] <- coef(gene.aov)
+ }
```

或

```
>for(i in 1:1522)
> {
+     gene.aov <- aov(Intensity ~ A + T + A * T,
+     sub = G == i)
+     res.mat[i, ] <- residuals(gene.aov)
+     coef.mat[i, ] <- coef(gene.aov)
+ }
```

- 实例演示
  - 1. Julian Faraway提供的一个简短的入门

    Introduction to R
    - 使用 R GUI
    - 使用 R Commander
    - 使用 R_XEmacs

    Further reference: Moore's *The Basic Practice of Statistics, Second Edition* (Freeman, 2000),
  - 2. John Verzani 提供的统计学入门

    SimpleR --- Using R for Introductory Statistics
    - 使用R_WinEdt

- 实例演示
  - 1. Julian Faraway提供的一个简短的入门

    Introduction to R
    - 使用 R GUI
    - 使用 R Commander
    - 使用 R_XEmacs

    Further reference: Moore's *The Basic Practice of Statistics, Second Edition* (Freeman, 2000),

- 2. SimpleR –
  - A sample session involving regression(18.1)
  - t-test (18.2)
  - A simulation example (18.3)

- 3. John Verzani 提供的统计学入门
  SimpleR --- Using R for Introductory Statistics
  (Chatpers 2 -15)

SimpleR

- Introduction
- Data
  - The data is stored in R as a vector
  - R Basics: Graphical Data Entry Interfaces
  - R Basics: Accessing Data
- Univariate Data
  - Classification:
    - categorical,
    - discrete numeric
    - continuous numeric.
  - Categorical data
    - Using tables

- Factors
- Bar charts
- Pie Charts

- Numerical data
  - Numerical measures of center and spread
    mean, var, sd, median, fivenum, summary
  - Resistant measures of center and spread
    median, trimmed mean, IQR, mad
  - Stem-and-leaf charts : cut, apropos
  - Histograms : hist(data, breaks=)
  - Box plots: library(), data()
  - Frequency polygons: simple.freqpoly()
  - Densities : kernel density estimator with option bw

- Bivariate Data
  - Handling bivariate categorical data: table(), prop.table()
  - Plotting tabular data: barplot()
  - Handling bivariate data: categorical vs. numerical: , or ~
  - Bivariate data: numerical vs. numerical
    - Comparing two distributions with plots: boxplot(), ……
    - Using scatterplots to compare relationships : plot()
    - Linear regression: simple.lm(), abline()
    - Residual plots
    - Correlation coefficients: cor()
    - Locating points

- Resistant regression
  resistance vs. robustness
- Using rlm or lqs for resistant regression

- Multivariate Data
  - Storing multivariate data in data frames: data.frame()
    !!To import outside data into R -- *R Data Import/Export*
  - Accessing data in data frames
    - Individually (by variables)
    - As an array -- [row,column]
    - As an list -- $ or [[position]], [[name]] (can be unambiguous!)

- Manipulating data frames: stack and unstack
- *Using R's model formula notation*
- Ways to view/visualize multivariate data (Section 5.5)
- The lattice package

- Random Data
  - Random number generators in R-- the ``r'' functions
    - Uniform -- runif(#,min=0,max=1)
    - Normal -- rnorm(#,mean=0,sd=1)
    - Binormial -- rbinom(#,n,p) / two ways: curve or points?
    - Exponential -- rexp(#,rate=1/mean)
  - Sampling with and without replacement using sample

- A bootstrap sample
- d, p and q functions
  - d – theoretical densities
  - p – cumulative functions / lower.tail=F or T
  - q – quantiles
- Standardizing, scale and *z* scores

- **Simulations**
  - **The CLT**
  - **Normal plots**
    - qqnorm or qplot
    - Qqline – draw a reference line

Using simple.sim and functions

Example – Check CLT of binormial observations

- You need to write a function that generates one of your random numbers, and then give it to simple.sim.
- Write the function f

```
> f = function(n=100,p=.5) {
  + S = rbinom(1,n,p)
  + (S- n*p)/sqrt(n*p*(1-p))
  + }
```

- n is the first variable given to the function, by default it is 100, p is the second by default it is p=.5. Now we would call simple.sim as

```
> simple.sim(1000,f,100,.5)
```

- Exploratory Data Analysis (EDA)

  -- using simple.eda() to produce several graphs

  Examples (from package Simple):

  1. homedata
  2. CEO salaries
- CI estimation/Hypothesis testing using (chap. 9 &10)
  - CLT
  - proportion test : prop.test()
  - Z-test (with std known): simple.z.test (user defined)
  - t-test : t.test(), [option:alt="greater"/"less"]
  - CI fo the median: wilcox.test(X, conf.int=TRUE)

- Two-sample tests
  - Two-sample tests for proportion: prop.test()
  - Two-sample t-tests
    - equal variances : var.equal=TRUE
    - unequal variances – Welch two-sample t-test
  - Matched samples : paired t-tests
    t.test(x,y,paired=TRUE)
  - Resistant two-sample tests: wilcox.test()

- Chi square tests
  - The chi-squared distribution
  - The chi-squared goodness of fit tests

    chisq.text(freq, p=probs)
  - The chi-squared tests of independence

  -- if two rows in a contingency table are ``independent'' (Pearson's Chi-squared test)
  - The chi-squared tests of homogeneity

  -- if the rows come from the same distribution

  (Pearson's Chi-squared test)

  with [[‘exp’]] – to see expected counts

- Regression Analysis
  - Simple linear regression model – usual commands
    - Plot(x,y)
    - Abline(lm(y~x))
    - Lm(y~x)
    - lm.result=simple.lm(x,y);
    - Lm.result=lm(y~x);

      summary(lm.result)

      coef(lm.result)

      resid(lm.result)

      fitted(lm.result)

      predict(lm.result,)

  - Testing the assumptions of the model
  - Statistical inference about sigma and the coefficients

par(mfrow=c(2,2))

Plot(lm.result)

- Residuals vs. fitted
- Normal qqplot
- Sclale-location – sqrt(std resid)
- Cook's distance

- Multiple Linear Regression
  - lm(z ~ x+y)
  - lm(z ~ x+y -1) # no intercept beta_0
  - summary(lm(z ~ x+y))

    Example 1: Sale prices of homes

    Example 2: Quadratic regression

    > dist = c(253, 337,395,451,495,534,574)

    > height = c(100,200,300,450,600,800,1000)

    > lm.2 = lm(dist ~ height + I(height^2))

    - lm.3 = lm(dist ~ height + I(height^2) + I(height^3))
  - Model choose through graphs and testing

*Easy plotting/ page 89.

- Analysis of Variance (ANOVA)
  - one-way analysis of variance
    oneway.test() -- under normality assumption
    kruskal.test() -- nonparametric test without normality

Example: Scholarship Grading, page 89

```
> x = c(4,3,4,5,2,3,4,5)
> y = c(4,4,5,5,4,5,4,4)
> z = c(3,4,2,4,5,5,4,4)
> scores = data.frame(x,y,z);    boxplot(scores)
> oneway.test(values ~ ind, data=scores, var.equal=T)
> df = stack(data.frame(x,y,z)) # prepare the data
> oneway.test(values ~ ind, data=df,var.equal=T)
> anova(lm(values ~ ind, data=df))  [aov()=anova(lm())]
> kruskal.test(values ~ ind, data=df)
```