Introduction · 统计工具分 supervised & unsupervised | $E(Y-\hat{Y})^2 = E[f(X)+\varepsilon-\hat{f}(X)]^2 = [f(X)-\hat{f}(X)]^2 + Var(\varepsilon)$ | 有监督有 y_i，无监督无 y_i (cluster.

Subset Selection Lasso Least Squares Generalized Additive Models Trees Bagging Boosting SVM ⟶ Interpretability High to Low. | SVM 提前 ⟵ Flexibility Low

Free Software: run for any purpose | study how the program work | adapt it to needs (Access to the source code) | redistribute copies | improve program, release it, benefit whole

help.start("A") help("A") ?A (精确) help.search() 返回相关 apropos() 返回相关obj 模糊) find("A") 返回包, example("A")

Working with R | R.version.string | getwd() | setwd() | dir("c:\\temp") 显示目录下文件 | memory.size() t: max f: use NA: 内存
memory.limit() 显示或改变内存限制 | ls(), object() 显示工作空间的项目 | object.size(x) x的大小.

object("package:base") base 中所有的object | search() 搜当前加载的包 | rm() 清除 | rm(list=ls(all=TRUE)) 全部清除

history(#) 显示最后#步命令 | savehistory("myfile") 保存历史命令命名为myfile | save.image("myfile") 保存工作空间.

q() 退出 | .libPaths() 库路径 | library() 显示已安装包 | install.packages(c("A","B")) 安装包 | %/% 取整 %% 取余 119%%7==0 判断整除

log() exp() sign() abs()
log(x, base=a) log10() log2()
?log 显示 log全部 option
gamma(), beta() c() 向量
R中基本unit是vector. length(x)
2*x+3 数乘, 5:1*x 按顺序乘
x+1:5 按顺序加, log(x) 逐个求
xx[1:4] 取前4号.
xx[xx>"a"] 取出>"a"的
u <- xx > "a" 返回 FALSE. TRUE.
xx[u] 取出 TRUE 位置值.
xx[c(1,4)] 调用1和4号.
xx[-c(2,3,5)] 调用除2,3,5号
ones <- rep(1,10) 10个1的向量
even <- seq(from=2, to=20, by=2)
2到20步长2的向量.
trend <- 1981:2005 生成年份向量
c(ones, even) 连接向量
cumsum(x) 各个数累加值
cumprod(x) 各个数累乘值
min(z[,1], z[,2]) 求全数字最小
pmin(z[,1], z[,2]) 对应位置最小
返回向量.
nlm(function(x) return( ), 8)
x: 自变量, 8: 起始值,
nlm 求函数最小值
返回 minimum, estimate 极值点
gradient 极值处梯度
interations 迭代次数.
D(expression(exp(x^2)), "x")
求导, "x" 求导对象 deriv3
integrate(function(x) x^2, 0, 1)
积分, 0,1 下限, 上限.
x*y 对应位乘法.
t(x) %*% y = crossprod(x,y) (-)(|)
x %*% t(y) = tcrossprod(x,y) (|)(-)
t(x) 转置. = outer(x,y)
z <- matrix(c(...), nrow=3,
ncol=2, byrow=F)
生成3行2列 F: 按列排
dim(A) 维数(行,列)
nrow(A) 行数 ncol(A) 列数
A[1,] 取1st行, A[,1] 取1st列
A[2,2] 取2行2列 (都返回向量)
class(D) 向量返回内容类别. 非向量返回格式类别
E <- A[2,2, drop=F] 强制返回量
A <- A[,-2] 返回除第2列的阵
det(A) A的行列式值.
eigen(A) A的特征根.

solve(A)): 求A逆
diag(2,3,4) 对角2, 3行4列
rep(c(1,2), c(3,3)) 重复3次1 3次2
svd(A) 奇异值分解 qr(A) QR分解
cbind(A,B) 左右联 rbind 上下联
runif(20) 长度20的均匀分布向量
slove(matrix(c(1,-1,1,1), nrow=2), c(2,4))
解 $\begin{pmatrix}1&1\\-1&1\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}2\\4\end{pmatrix}$
union(x,y) 并集, intersect(x,y) 交集
setdiff(x,y) (x-y), c%in%y 查c在y内
setequal(x,y) 检查x y是否相同
choose(n,k) = C_n^k
z <- 3.5-8i (虚数)
Re(z) 实部 Im(z) 虚部 Mod(z) 模长
Conj(z) 共轭. is.complex 判别.
most number are rounded to an accuracy of 53 binary digits.
除用相同公式算否则2个浮点数不会相等. sqrt(2)*sqrt(2) == 2 (F)
sqrt(2)*sqrt(2) ≠ 2 但减其他正常
x <- 0.3-0.2, y <- 0.1, x==y (F)
identical(x,y) F all.equal(x,y) T
命名: 区分大小写, 不以数/符号开头
不能有空格. 中间随意
③ Data Structures.
R可处理的数据类型: numeric, character, logical, complex, raw
R中数据容器: scalars, vectors, matrices, arrays, dataframe, list
attributes(A) 返回A的各项目名, A的类型
向量仅可装(数, 字符, 逻辑)
同一个向量仅可含一种类型
矩阵仅可装(数, 字符, 逻辑)
同矩阵用相同类型.
matrix(c(..), nrow=.., ncol=..,
byrow=T/F, dimnames=list(row,col))
y[1:2, c(3,4)] 取1-2行, 3和4列.
z <- as.vector(y) 转为向量
dim(z) <- c(4,5) 转为4x5矩阵
str(z) => int [1:5, 1:4] 1 2 3 4 ...
z <- (1:24) dim1 <- c("A1","A2")
dim2 <- c("B1","B2","B3")
dim3 <- c("C1","C2","C3","C4")
z <- array(1:24, c(2,3,4),
dimnames = list(dim1, dim2, dim3))
A, B嵌在C1, C2, C3, C4里
aperm(A, c(2,3,1)) (调位)
(=) A变成 c(3,4,2) 维

数据框不同列可放不同类型
A <- c(1,2,3,4) B <- c(a,b,c,d)
cbind 变矩阵, 1,2,3,4 变字符.
data.frame(A,B) 者同时保留名字
names(A) 显示框内名字
A[1,2] 或 A[c("name1","name2")] 取2
不可 A[name1] 返回 data.frame.
A$name1 返回向量
factor(数据, ordered=TRUE)
有排序因子. 默认按字母表
factor(status, order=TRUE,
levels=c("Poor","Improved","Exc"))
指定排序. P<I<E.
status2 <- c("P","I","E")
status3 <- factor(status2, levels
= c("P","I","E"), labers=c("C","B",
"A")) 返回 [1] CBA | Levels: CBA
不在labers中的会被miss.
status3[4] <- "P" × 因子不能这样
转回向量加好后加回
! NOT & AND | OR
A&&B 仅B中true的去比.
A||B 仅B中true的去比
isTRUE(6>4) TRUE.
identical('Twins', 'twins') FALSE
xor(5==6, 'FALSE') TRUE (或!)
xor(!isTRUE(TRUE), 6>1) TRUE
xor(4>=9, 8!==8.0) FALSE.
8!=8.0 F. (T!=F)==!(6==7) (T)
T&c(T,F,F) (T F F)
T&&c(T,F,F) (T) T||c(T,F,F) (T)
T|c(T,F,F) (T T T)
5>8||6!=8&&4>3.9 (T) 先&&
aaa <- gl(2,5) class(aaa) factor
mode(aaa) numeric
typeof(aaa) integer.
List 不要求每一列数目相同
R中最complex的 data types
Lst <- list(name="Fred", wife
= "Mary", child.ages=c(4,7,9))
Lst[[2]] 取出"Mary", Lst[[3]][2]: 7
Lst[["name"]]: Fred = Lst$name
Z$c <- "K" 增加/编辑项.
Z$b <- NULL 删除项b
x$a 模糊找. x[[""]] 精确找
x[[, exact=false]] 模糊
Formulas: f <- y~x
NA 有分 integer... 类.
NAN 属于 NA. NAN: not a number
attach: 绑定到某数据上

detach 解除绑定.
with(mtcars, { }) 括号中绑定mtcars
{ } 中 <- 不保存, <<- 保存
Data Input.
Mydata <- data.frame(age=numeric(0), ...)
My data <- edit(My data) 窗口编辑
x <- numeric(10) Data.entry(x): 输入10个
a <- scan() 输数不输字, 回车一次, 放空回车结束 (空格/换行/制表分割) → "\t"
rt <- scan(file="~\\rt.txt", sep="\n")
回车分割 同行合并 默认空格分.
rt <- scan(file=file.choose, sep="\t")
scan(... what=double(), skip=1)
双精度, 跳过前一行
read.table(file, header=T/F, sep=" ",
row.names=" ") header: 是否1st作names.
colClasses 决定每列 class = c("integer" ...)
skip: 从头跳 n行. stringsAsFactors = FALSE
nrow: 读 n行. 禁止文字转因子
文件路径用\\, 可读URL
read.delim(file) 默认1st作表头, 空格分
url = url("http: ...") ds = readLine(url)
read.xlsx("... .xlsx", 1, header=T, as.data.frame=T)
read.dta("... .dta") 从 Stata
read.table("clipboard") 从剪贴板读
data() 读内建data,
data(package=.packages(all.available=T))
读外部内置, try(data(package="M")) 看M中数据名, history(Inf) 看全部历史(窗口)
loadhistory(file) 保存历史
pdf("~\\1.pdf") data(mtcars)
plot(mtcars$wt~mtcars$mpg)
dev.off 将图存入pdf
write.table(data, file="...txt", sep=" ",
quote=FALSE, append=FALSE, na="NA")
↳ 防覆盖
save(data, file="....Rdata")
load("....Rdata")
save(list=ls(all=TRUE), file="all.Rdata")
writeClipboard(data) 写入剪贴板.
file.exists("... .txt") 确认文件存在F.
data.frame 改名: names(A)[2] <- "B"
A$B <- factor(A$B, Levels=c(1,2), labels=c("male", "female")) 将数据1,2用male.female表示.
data$sumx <- data$x1 + data$x2
attach(data); data$sumx <- x1+x2 detach(data)
data <- transform(data, sumx=x1+x2) 三行同样

seq(from=0.04, by=0.01, length=11)
seq(0.04 , by=0.01, along=N)
0.04 开始, N个, 步长0.01
最后一个值超出范围则停止,不超出.
seq(1.4, 2.1, 0.3) : [1] 1.4, 1.7, 2.0
Sequence(c(4,3,4,4,4,5))
生成 1:4, 1:3, 1:4 ......
rep(1:4, 2) 重复2次 1:4
rep(1:4, each=2) 11223344
rep(1:4, each=2, times=3)
11223344 11223344 11223344
rep(1:4, c(4,1,4,2))
1x4, 2x1, 3x4, 4x2 次.
rep(☐ , ☐)
重复体 单个为次数/向量对应
gl(最大, 重复次数, 总长度)
生成的是levels,长度不够再重
Soft <- gl(3,2,6, labels=c("A","B","C"))
Temp <- gl(2,3,6, labels=("D","E"))
data.frame(Temp, Soft)
修改变量:
A$B[A$C >=55 & A$E <=75] <- "X"
B可以已有也可新建. 也可用:
A <- within(A, {B <- NA | B[C>1] <- "X" | B[F<9] <- "K"})
可以保存新建量返回, with在{}中新建的不保存(用<<-保存
重命名变量: fix(A) (窗口)
names(A)[2] <- "B"
names(A)[1:2] <- c("A","B")
缺失值NA, 不可能值NaN
A==NA 永远是FALSE, 用is.na(A) 返回T/F的阵或vector.
x <- c(1,2,NA,3) y <- x[1]+x[2]+x[3]+x[4]
print(y) [1] NA 有NA算完全NA.
y <- sum(x, na.rm=TRUE) NA转0
bad <- is.na(x) x[!bad] 不显示NA
x <- c(1,2,NA) Y <- c("a",NA,"b")
good <- complete.cases(X,Y)
返回TFF, X[good]返回1 y[good]返"a"
na.omit() 删除含NA的行(row)
complete.case(A) 对A的每行查
NULL: 不存在, 空. Z <- NULL 新建
NULL可添入东西, NA不可 NULL无mode
NULL的length为0, NA的为1
Dates → Date class
Times → POSIXct or POSIXlt
Dates存从1970-01-01开始的天数,
Times存从1970-01-01开始的秒数
x <- as.Date("1970-01-01") unclass为0
POSIXct是一串数字, POSIXlt是list
x <- Sys.time() 获取系统时间(ct型)
date() 返回"Wed Mar 09 23:49:16 2016"
today <- Sys.Date() 只返日期
format(today, format="%M %d %Y")
format(today, format="%y")
2个Date型相减可得日期差
difftime(today, dob, units="weeks")
求差多几周 myformat <- "%m/%d/%y"
as.Date(A$B, myformat) 转格式
可将factor转为Date
排序 A[order(A$B),] 按age排序 小到大

attach(A) B <- A[order(gender, -age),]
detach(A) 先排性别, 再年龄大到小排
myvars <- names(A) %in% c("B","C")
返回T.F. 判断A各项名是否是B或C
A[!myvars] 或 A[c(-1,-5)] 剔除
选择变量: A[which(A$X=="M" & A$Y>30),]
which返回c(....)代入A取数据.
A <- subset(B, C>=35 | C<80, select =c(X,Y)) 选出B中C≥35或C<80的条目的X,Y数据
A <- subset(B, k=="M" & c>50, select =X:Y. 选出X到Y之间全部数据
A[sample(1:nrow(A), 3, replace=F),]
抽A中的样本(范围, 次数, 是否放回)
sample返回一个向量.
加总: aggregate(price~cut, dia, mean)

	cut	price
1	Fair	45
2	Good	39

返回以cut分类的dia数据中price的mean.
aggregate(price~cut+color, dia, mean)
按cut、color交叉分类求price均值
aggregate(cbind(price, car)~cut, dia, mean)
以cut分类返回price, car的mean
merge(A, B)

A:

a	b	c
m	n	h
o	p	Q
G	F	L

B:

a	b	d
m	n	x
z	p	Y
o	p	x

merge(A, B) ⇒

a	b	c	d
m	n	h	x
o	p	Q	x

merge(A, B, all=T) ⇒

a	b	c	d
m	n	h	x
o	p	Q	x
Z	P	NA	Y
G	F	L	NA

merge(x=A, y=B, by.x =c("m","n"), by.y = c("m","n")) 仅m对n一致才拼接
(plyr包) join(x=A, y=B, by=c("m","n"))
rbind 上下连dataframe时不用使AB列名顺序对应 会自动匹配 但需一样
重命名matrix行列名. rownames.
colnames(A) <- c("..", ".. ")
(reshape包): melt(A, id=(c("ID","Time")))

A:

ID	Time	X1	X2
1	1	5	6
1	2	3	5
2	1	6	4
2	2	2	4

→ md:

ID	Time	Vari...	value
1	1	X1	5
1	2	X1	3
2	1	X1	6
2	2	X1	2
1	1	X2	6
1	2	X2	5
2	1	X2	1
2	2	X2	4

ID	Var..	1	2
1	X1	5	3
1	X2	6	5
2	X1	6	2
2	X2	1	4

← cast(md, ID+Variable~Time) 把time值当分类

Time	X1	X2
1	5.5	3.5
2	2.5	4.5

← cast(md, Time~variable, mean)

country	developed	GDP2000	GDP2005	GDP2010
China	0	5000	5500	5010
USA	1	6000	6500	6010
Japan	1	7000	7500	7001

long <- reshape(Data, idvar="country", varying=list(names(Data)[3:5]), v.names="GDP", timevar="year", times=c(2000,2005,2010), direction="long")
long[order(long$country),]

	country	developed	year	GDP
China.2000	china	0	2000	5000
China.2005	china	0	2005	5500
China.2010	china	0	2010	5010
Japan.2000	Japan	1	2000	7000
Japan.2005	Japan	1	2005	7500
Japan.2010	Japan	1	2010	7010
USA.2000	USA	1	2000	6000
USA.2005	USA	1	2005	6500
USA.2010	USA	1	2010	6010

wide <- reshape(long, v.names="GDP", idvar="country", timevar="year", direction="wide") 恢复md rowname(wide) <- NULL
回到md但GDP.2000带上.

paste("A", c("B","C","D"), c("E","F","G"))
"ABE" "ACF" "ADG".
paste("A", "B", "C", sep="/") "A/B/C",
x <- "Q", Y <- 25 paste("A", x, "B", Y) "AQB25"
sprintf("A %s B %s", X, Y) "A Q B 25"
sprintf("A %s B %s C %s", c("Q","M"), c("H",3,"F",9), Y) AQBHCY AMB3CY AQBFCY AMB9CY
(stringr包) str_split(string=A$B, pattern="-") 去掉A$B中的"-" (List)
Reduce(rbind, A) 将list中du按行合为矩阵
str_sub(string=A$B, start=1, end=3)
取前3个字符.
A[str_sub(string=A$B, start=4, end=4)==1,]
抽出A中B的第4个字符为1的条目
str_detect(string=A$B, pattern="k") 找出带有k的条目 返回位置向量, 区分大小写
str_detect(string=A$B, "k") 也可用 忽略
str_detect(string=A$B, ignore.case("j"))
str_split(string=A, pattern="H", n=2) 看到H分割, 分2段, H分后去掉. 以第一个H为准.
str_trim(A) 去掉开头和结尾空格和\n.
str_extract(string=A, pattern="k") 提k, 无k的条目返回NA, 有k整句提出.
[0-9]代表0-9数, [0-9][0-9][0-9][0-9] 4位数
[0-9]{4}代表4位数 \\d 数
\\d{1,3}, 1、2、3个数相连
"^\\d{4}"开头是4位数 "\\d{4}$"结尾是4位数,
"^\\d{4}$"开头结尾都4位数. str_replace 仅一次
str_replace_all(string=A, pattern="\\d", replacement="x") 全部替换成x
unlist(str_split(Sys.time(), " "))[2]
for(A) 命令 或 for(A){ }可换行
A: (i in 1:4) (i in seq_along(x)) → 1:x的长度
(letter in x) → 用x中的字符按顺序

for(i in 1:100){iTotal <- iTotal+i}
cat("Sum of 1-100:", iTotal, "\n", sep="")

szSymbols <- c("MSFT","GOOG","AAPL" ...)
for(SymbolName in szSymbols){
Cat(SymbolName, "\n", seq="")

x <- matrix(1:6, 2, 3)
for(i in seq_len(nrow(x))){
for(j in seq_len(ncol(x))){
print(x[i,j]) } }

count <- 0 while(count < 10){print(count)
count <- count+1}

i <- 1; iTotal <- 0 while(i<=100){iTotal <- iTotal+i i <- i+1}

z <- 5 while(z>=3 && z<=10){print(z)
coin <- rbinom(1,1,0.5) # 伯努利分布 p=0.5
if(coin==1){z <- z+1} else {z <- z-1}}

i <- 1 iTotal <- 0 repeat {iTotal <- iTotal+i
i <- i+1 if(i<=100) next else break}

a <- 3 | if(a==1){ | print("a==1") | }else{print("a!=1")} }else 合体

grade <- as.factor(c("grade1", "grade2")) |
if(!is.factor(grade)){ | grade <- as.factor(grade) | }else{ | print("Grade already is a factor") | } 判断是否因子 是输, 不是转.

a <- 4 if(a==1){ | print("a==1") | }else if(a==2){ | print("a==2") | }else{ | print("Not 1&2")} 多条件

x <- matrix(1:6, 2, 3) | ifelse(x>=0, sqrt(x), NA)
若x≥0开方, 否则出NA 可对矩阵逐个

n <- 1 | switch(n, print("option1"), print("option2"), print("option3")) 输出 option1

ccc <- c("b", "QQ", "a", "A", "bb")
for(ch in ccc) cat(ch, ":", switch(EXPR=ch, a=1, b=2:3), "\n") EXPR固定开关, ch被判断元素, 若循环到a输1, b输2 3

用户函数 my <- function(a1, a2 ...){ ... }
函数可作为参数传递到另一个函数, 可嵌套
colmean <- function(y){ | nc <- ncol(y) | means <- numeric(nc) | for(i in 1:nc){ | means[i] <- mean(y[,i]) | } | means | } 有NA出NA
colmean <- function(y, removeNA=TRUE){ | nc <- ncol(y) | means <- numeric(nc) | for(i in 1:nc){ | means[i] <- mean(y[,i], na.rm=removeNA) | } | means | } 可选是否除NA.
median(x) 中位数.

parboth <- function(a,b){ | c <- pmax(a,b) | d <- pmin(a,b) | answer <- list(median(c), median(d)) | names(answer)[[1]] <- "..." | names(answer)[[2]] <- "..." | return(answer) | } 这样可return多结果

twosam <- function(y1, y2){ | n1 <- length(y1); n2 <- length(y2) | yb1 <- mean(y1); yb2 <- mean(y2) | s1 <- var(y1); s2 <- var(y2) | s <- ((n1-1)*s1+(n2-1)*s2)/(n1+n2-2) | tst <- (yb1-yb2)/sqrt(s*(1/n1+1/n2)) | tst} T检验 sd: 标准差 可选na.rm

args() 返回函数参数.
lm(data=mydata, y~x, model=F, 1:100)
= lm(y~x, mydata, 1:100, model=F)
...写在前面不可按位置匹配