

## Data manipulation

### Vectors:

```
v<-scan() v<-1:n v<-c(1,3,4) v<-c(nm1=3,nm2=5,...)
v<-seq(lo,hi,step) v<-seq(length=n,from=lo,by=step)
v<-rep(expr,times=5) v<-rep(expr,each=5)
boolv=v>3 & v<5 strv=c("vr","br")
length(v) names(v) names(v)<-c("p1","p2"..)
v[1] v{2:4} v[v>3] v[boolv] v[-exclidxvec] v[c("p1","p2")]
factvec=factor(c("f1","f2","f1",...))
v=ordered(c("fr","th"),levels=c("fr","se"),labels=c())
```

### Matrices:

```
m<-matrix(v,nrow,ncol) m<-matrix(c(1,2,3,4,5,6),2,3)
diag(vec) col(matrix) row(m) A[col(A)>=row(A)]
dim(m)<-c(row,col,...) dimnames(m)<-list(c(),c()...)
m[vec,vec] m[,vec] m[vec,] matrix ops: %*% %/% %o%
eigen(sqrn) $values, $vectors svd(m) $u $v $d
solve(m) solve(m,b)
rowconcat(w/recycle): rbind(m1,m2) colconcat: cbind(m1,m2)
aperm(m,c(newdim1,newdim2,...))
```

### Lists/objects:

```
li<-list(na1=obj,na2=obj,...) li<-c(li,na5=obj)
li$na1 li$na1[2] li[[2]] li[[2]][3]
attributes(li) length(li)
objects: class(o) names(o) methods(fun) func(obj)=func.class(o)
Data frames: list of vectors with names, matrix index too
df$fred df$fred[3] df[, "fred"] df[3:9,c("fred","bob")]
attach(df) detach(df) summary(df)
df<-data.frame(vecs,facrs,...[row.names=,col.names=])
crosstabbing: attach(df);table(fact1[,fact2],...) gives cts while
tapply(var,fact|list(fac1,fac2),func) gives mean/std etc
```

## Programming

**Special values:** NaN Inf -Inf TRUE FALSE

**Operators:** %% (modulo) %/% (int division)

**Assignment:** <- -> = \_ as in C returns val

**Logical:** !, & | vectorwise, &&, || scalar with partial evaluation

### Typcasting:

as(o,"numeric"|"character"|"logical"|"vector" matrix")

is(obj, as above), also as.numeric(), is.numeric()

**recycling:** 2\*c(1,2)+c(3,4,5,6)+1 = (6,9,8,11)

**Common functions:** round trunc floor ceiling, abs, sign, log, log10, sqrt, exp, sin, asin cosh log2 log(x,base) gamma lgamma sum prod cumsum cumprod cummax cummin range mean var std quantile(x,pvec) duplicated(c(1 2 1 4 2))=F F T F T unique union intersect setdiff

### Functions:

```
name<-func(unnam1,unarg2,named1=def1,...) {.... return(3); }
call as: name(1,2) or name(na=2,1,2) or name(1,2,3)
for (i in vector) { } or stat; while (boolexpr) statmt
repeat {} stmt; if (condition) statement else statement
ifelse(boolvec,trueval,falseval),
apply(m,veckeekeepdims,function[,funcargs])
sweep(m,veckeekeepdims,var[,op=-]) does op using var
Strings: paste(v1,v2,sep="","collapse=") (collapse bet els)
substrng(text,firt,last=1000000)
grep(regex,strvec) returns index of matches
regexpr(regex,vec) returns -1 if false, posi if true
cat("str1","str2","n") outputs string str=format(num,digits=n,
justfy="left"|"right", trim=TRUE, big.mark="")
```

## Files & Objects

**Objects in memory:** search() ls([n]"pth"|"regexp"), find("var")  
rm(obj) rm(list=ls(), library(), edit(obj),  
save(obj1,obj2,...,file="/..."), ,save.image(file=""), load(fn)

**File directories:** getwd() setwd("dir") , unlink("file"),  
list.files(path="","pattern=")

**Scripts:** history(), save.history(file=""), edit("file"),  
sink(file="","append=FALSE) for current coms,  
source(scrfn,echo=TRUE,print.eval=TRUE)

### Reading dataframes:

```
df=read.table(file="/..URL",header=FALSE,sep="",
quote="\""",row.names=strvec[colon]colname,na.strings=str
vec, skip=n, strip.white=FALSE,blank.lines.skip=TRUE,
comment.char="#"") also read.csv(), write.table(obj,"file")
```

## Graphics

```
plot(x,y,type="p[oint]||[ines]|n[one]|b[oth]|s[tep]",main="title",s
ub="subtitle",xlab="","ylab=""," logx|y|xy,
xlim=2vec,ylim=2vec)
```

also: hist(x), bar(x), sunflowerplot(x,y) piechart(x), boxplot(x),  
coplot(x,y,z), interaction.plot(f1,f2,y), qqplot(x), qqline(x)

**options:** add=FALSE, axes=TRUE, type="p","l","o", xlim=v,  
ylim=, xlab=,ylab=,main=,sub=""," pch=0|..|25|"x", pty="s"|"m"

**lowlevel:** points(x,y),lines(x,y),text(x,y,""), arrows(x0,y0,x1,y1),  
abline(h=y,v=x,reg=mod,coef=v), legend(x,y,legend), rug(x)

**params:** ?par, incl options, bg="col" cex\*=txtsz, col\*="color",  
font\*, las=txtor ,lty=1-6, lwd=width,mfcol(c(nr,nc)), ps=pts,  
\*= has .axis, .lab, .main, .sub also options above for par  
op<=par();par(...);par(op); par(mfrow=c(2,2) 2x2 multiplot seq  
pairs(df) (matrix scatter) persp(x,y,z) contour(x,y,z) image(x,y,z)  
3-D grid data

windowgmt: x11(), pdf() dev.list() dev.cur() dev.set(n) dev.off(n)

## Models

**formulas:** var|log(v)~v1+v2-v3+v1:v2+v1\*v3+log(v5)+I(v1\*v2)  
-1 excludes constant, \* gives interaction+indiv terms  
(var1+var2+var3)^2 poly(var1,2) or poly(var1,var2,3)

### To create models:

```
obj<-lm(formula,data=df,na.action=na.omit,subset=vec)
obj<-glm(...,family=binomial|Gamma|Gaussian|poisson|
inverse.Gaussian)
```

```
library(nls);nls(Var~exprVar,data=,start=namvec,trace=T)
```

```
library(mgcv);obj<-gam(var~s(var1)+s(var2)+var3,...)
```

```
library(mda);obj<-mars(xmatrix,y,degree=n)
```

```
library(rpart);obj<-rpart(formula,...)
```

```
library(nnet);obj<-nnet(...,size=n) obj<-lowess(x,y)
```

**To analyze:** print, plot, summary, predict(obj,newdata), fitted,  
residuals, coefficients, plot(dep~indep)

## Probability

```
sample(vector,num) sample(1:n,num) density(x,bw=0.1)
```

```
d*(x,params,log=FALSE) gives prob(pdf) at x
```

```
p*(x,params) gives CDF at x q*(x,params) gives inverse CDF
```

```
r*(n,params) generates n random# fitdistr(x,"*",knownparam=)
```

\* may be: norm mean=,sd= lnorm meanlog=,sdlog=

beta shape1=,shape2= binom size=,prob=

cauchy location=,scale= chisq df= exp rate=

f df1=,df2= gamma shape=,rate= geom. prob=

hyper m=,n=,k= logis location=,scale=

nbinom size=,prob= pois lambda= t df=

unif min=,max= weibull shape=,scale= Wilcox m=,n=