Parts of The R Book

January 16, 2012

Table 1: Statistical Models (p.323)

The explanatory variables	Models choose
1. All continuous	Regression
2. All categorical	ANOVA
3. Both (mixed)	ANCOVA

The response variables	Models choose
1. Continuous	Normal regression,ANOVA or ANCOVA
2. Proportion	Logistic regression
3. Count	Log-liner models
4. Binary	Binary logistic analysis
5. Time and death	Survival analysis

Maximum likelihood: given the data and given our choice of model, what values of the parameters of that model make the observed data most likely?

The principle of Parsimony (Occam's razor) :

- models should have as few parameters as possible;
- linear models should be preferred to non-linear models;
- experiments relying on few assumptions should be preferred to those relying on many;
- models should be pared down until they are *minimal adequate*;
- simple explanations should be preferred to complex explanations.

So, we would only include an explanatory variable in a model if it significantly improved the fit of the model. Parsimony says that, other things being equal, we prefer:

- a model with n-1 parameters to a model with n parameters;

- a model with k-1 explanatory variables to a model with k explanatory variables;
- a linear model to a model which is curved;
- a model without a hump to a model with a hump;
- a model without interactions to a model containing interactions between factors.

Parsimony requires that the model should be as simple as possible. This means that the model should not contain any redundant parameters or factor levels. We achieve this by fitting a maximal model and then simplifying it by following one or more of these steps:

- remove non-significant interaction terms;
- remove non-significant quadratic or other non-linear terms;
- remove non-significant explanatory variables;
- group together factor levels that do not differ from one another;
- in ANCOVA, set non-significant slopes of continuous explanatory variables to zero.

All the above are subject, of course, to the caveats that the simplification make good scientific sense and do not lead to significant reductions in explanatory powered.

Just as there is no perfect model, so there may be no optimal scale of measurement for a model. Suppose, for example, we had a process that had Poisson errors with multiplicative effects among the explanatory variables. Then, one must choose between three different scales, each of which optimizes one of three different properties: 1. the scale of $\sqrt{y}$ would give constancy of variance; 2. the scale of $y^{2/3}$ would give approximately normal errors; 3. the scale of $ln(y)$ would give additivity.

Types of statistical models: the null model; the minimal adequate model; the current model; the maximal model; and the saturated model.

Table 2: Statistical modeling involves the selection of minimal adequate model from a potentially large set of more complex models, using stepwise model simplification.

Model	Interpretation
Saturated model	One parameter for every data point Fit: perfect Degrees of freedom: none Explanatory power of the model: none
Maximal model	Contains all (p) factors, interactions and covariates that might be of any interest. Many of the model's terms are likely to be insignificant. Degrees of freedom: $n - p - 1$ Explanatory power of the model: it depends
Minimal adequate model	A simplified model with $0 \leq p' \leq p$ parameters Fit: less than the maximal model, but not significantly so Degrees of freedom: $n - p' - 1$ Explanatory power of the model: $r^2 = SSR/SSY$
Null model	Just one parameter, the overall mean $\bar{y}$ Fit: none; $SSE = SSY$ Degrees of freedom: $n - 1$ Explanatory power of the model: none

Table 4: Examples of R model formula. In a model formula, the function I (upper case i) stands for 'as is' and is used for generating sequencces I(1:10) or calculating quadratic terms I(xˆ2)

Models	Model formula	Comments
Null	$y \sim 1$	1 is the intercept in regression models, but here it is the overall mean y
Regression	$y \sim x$	x is a continuous explanatory variable
Regression through origin	$y \sim x - 1$	Do not fit an intercept
One-way ANOVA	$y \sim sex$	sex is a two-level categorical variables
One-way ANOVA	$y \sim sex - 1$	as above, but do not fit an intercept (give two means rather than a mean and a difference)
Two-way ANOVA	$Y \sim sex + genotype$	genotype is a four-level categorical variable

continuous...

Models	Model formula	Comments
Factorial ANOVA	$y \sim N * P * K$	N,P,K are two level factors to be fitted along with all their interactions
Three-way ANOVA	$y \sim N * P * K - N : P : K$	as above, but don't fit the three-way interaction
Analysis of covariance	$y \sim x + sex$	A common slope for y against x but with two intercepts, one for each sex
Analysis of covariance	$y \sim x * sex$	two slopes and two intercepts
Nested ANOVA	$y \sim a/b/c$	Factor c nested within factor b within factor a
Split-plot ANOVA	$Y \sim a * b * c + Error(a/b/c)$	A factorial experiment but with three plot sizes and three different error variances, one for each plot size.
Multiple regression	$y \sim x + z$	Two continuous explanatory variables, flat surface fit
Multiple regression	$y \sim x * z$	Fit an quadratic term as well $(x + z + x : z)$
Multiple regression	$y \sim x + I(x^2) + z + I(z^2)$	Fit a quadratic term for both x and z
Multiple regression	$y = poly(x, 2) + z$	Fit a quadratic polynomial for x and linear z
Multiple regression	$y \sim (x + z + w)^2$	Fit three variables plus all their interactions up to two-way
Non-parametric model	$y \sim s(x) + s(z)$	y is a function of smoothed x and z in a generalized additive model
Transformed response and explanatory variables	$log(y) \sim I(1/x) + sqrt(z)$	All three variables are transformed in the model

- : indicates deletion of an explanatory variable in the model
* : indicates inclusion of explanatory variables and interactions
/ : indicates nesting of explanatory variables in the model
+ : indicates inclusion of an explanatory variable in the model (not addition)
| : indicates conditioning (not 'or'), so that $y \sim x|z$ is read as 'y as a function of x given z'.

Box-Cox Transformations

Sometimes it is not clear from theory what the optimal transformation of the response variable should be. I this circumstances, the Box-Cox transformation

Table 3: Model simplification process.

Step	Procedure	Explanation
1	Fit the maximal model	Fit all the factors, interactions and covariants of interest. Note the residues deviance. If you are using Poisson or binomial errors, check for overdispersion and rescale if necessary.
2	Begin model simplification	Inspect the parameter estimates using the R function `summary`. Remove the least significant terms first, using `update -`, starting with the highest-order interactions.
3	If the deletion causes an insignificant increase in deviance	Leave the term out of the model. Inspect the parameter values again. Remove the least significant terms remaining.
4	If the deletion causes a significant increase in deviance	Put the term back in the model using `update +`. These are the statistically significant terms as assessed by deletion from the maximal model.
5	Keep removing terms from the model	Repeat steps 3 or 4 until the model contain nothing but significant terms. This is the minimal adequate model. If none of the parameters is significant, then the minimal adequate is the null model.

offers a simple empirical solution. The idea is to find the power transformation, λ(lambda), that maximizes the likelihood when a specified set of explanatory variables is fitted to

$$\frac{y^\lambda - 1}{\lambda}$$

as the response. For the case $\lambda = 0$ the Box-Cox transformation is defined as $log(y)$.

Table 5: Summary of statistical models in R

Models	Summary
lm	fits a linear model with normal errors and constant variance; generally this is used for regression analysis using continuous explanatory variables.
aov	fits analysis of variance with normal errors, constant variance and the identity link; generally used for categorical explanatory variables or ANCOVA with a mix of categorical and continuous explanatory variables.
glm	fits generalized linear models to data using categorical or continuous explanatory variables, by specifying one of a family of error structures (e.g. Poisson for count data or binomial for proportion data) and a particular link function.
gam	fits generalized additive models to data with one of a family of error structures in which the continuous explanatory variables can (optionally) be fitted as arbitrary smoothed functions using non-parametric smoothers rather than specific parametric functions.
lme	and lmer fit linear mixed-effects models with specified mixtures of fixed effects and random effects and allow for the specification of correlation structure amongst the explanatory variables and autocorrelation of the response variables (e.g. time series effects with repeated measures). lmer allows for non-normal errors and non-constant variance with the same error family as GLM.
nls	fit a non-linear regression model via least squares, estimating the parameters of s specified non-linear function.
nlme	fits a specified non-linear function in a mixed-effects model where the parameters of the non-linear function are assumed to be random effects; allows for the specification of correlation structure amongst the explanatory variables and autocorrelation of the response variable (e.g. time series effects with repeated measures).
loess	fits a local regression model with one or more continuous explanatory variables using non-parametric techniques to produce a smoothed model surface.

Models	Summary
tree	fits a regression tree model using binary recursive partitioning whereby the data are successively split along coordinate axes of the explanatory variables so that at any node, the split is chosen that maximally distinguishes the response variable in the left and right branches. With a categorical response variable, the tree is called a classification tree, and the model used for classification assumes that the response variable follows a multinomial distribution.

For most of these models, a range of generic functions can be used to obtain information about the model. The most used are as follows:

summary produces parameter estimates and standard errors from lm and ANOVA tables from aov. You can use summary.lm or summary.aov to get the alternative form of output (an ANOVA table or a table of parameter estimates and standard errors).

plot produces diagnostic plots for model checking, including residuals against fitted values, influence tests, etc.

anova is a wonderfully useful function for comparing different models and producing ANOVA tables.

updata is used to modify the last model fit; it saves both typing effort and computing time.

coef gives the coefficients (estimate parameters) from the model.

fitted gives the fitted values.

resid gives the residuals.

predict use information from the fitted model to produce smooth functions for plotting a line through the scatterplot of your data.

Optional arguments in model-fitting functions: subset, weights, data, offset, na.action.

Akaike's Information Criterion (AIC) is known in the statistics trade as a penalized log-likelihood. If you have a model for which log-likelihood value can be obtained, then

$$AIC = -2 \times log - likelihood + 2(p + 1)$$

where p is the number of parameters in the model, and 1 is added for the estimated variance (you can call this another parameter if you wanted to). When comparing two models, *the smaller the AIC, the better the fit.*

Table 6: Useful non-linear functions

Name		Equation
Asymptotic functions		
	Michaelis-Menten	$y = \frac{ax}{1+bx}$
	2-parameter asymptotic exponential	$y = a(1 - e^{-bx})$
	3-parameter asymptotic exponential	$y = a - be^{-cx}$
S-shaped functions		
	2-parameter logistic	$y = \frac{e^{a+bx}}{1+e^{a+bx}}$
	3-parameter logistic	$y = \frac{a}{1+be^{-cx}}$
	4-parameter logistic	$y = a + \frac{b-a}{1+e^{(c-x)/d}}$
	Weibull	$y = a - be^{-(cx^d)}$
	Gompertz	$y = ar^{-be^{-cx}}$
Humped curves		
	Ricker curve	$y = axe^{-bx}$
	First-order compartment	$y = ke^{(-e_a x - e^{-e_b x})}$
	Bell-shaped	$y = ae^{-\|bx\|^2}$
	Biexponential	$y = ae^{bx} - ce^{-dx}$

Non-linear Regression

What we mean in this case by 'non-linear' is not that the relationship is curved (it was curved in the case of polynomial regressions, but that was linear model), but that the relationship cannot be transformation of the response variable or the explanatory variable or both.

In R, the main difference between linear models and non-linear models is that we have to tell R the exact nature of the equation as part of the model formula when we use non-linear modelling. In place of lm we write nls (non-linear squares). Then, istead of $y \sim x$, we write $y \sim a - b * exp(-c * x)$ to spell out the precise nonlinear model we want R to fit the data.

Changing the look of graphics

Many of the changes that you want to make to the look of your graphics will involve the use of the graphic parameters function, `par`. Other changes, however, can be made through alterations to the arguments to high-level functions such as `plot, points,lines, axis,title` and `text`.

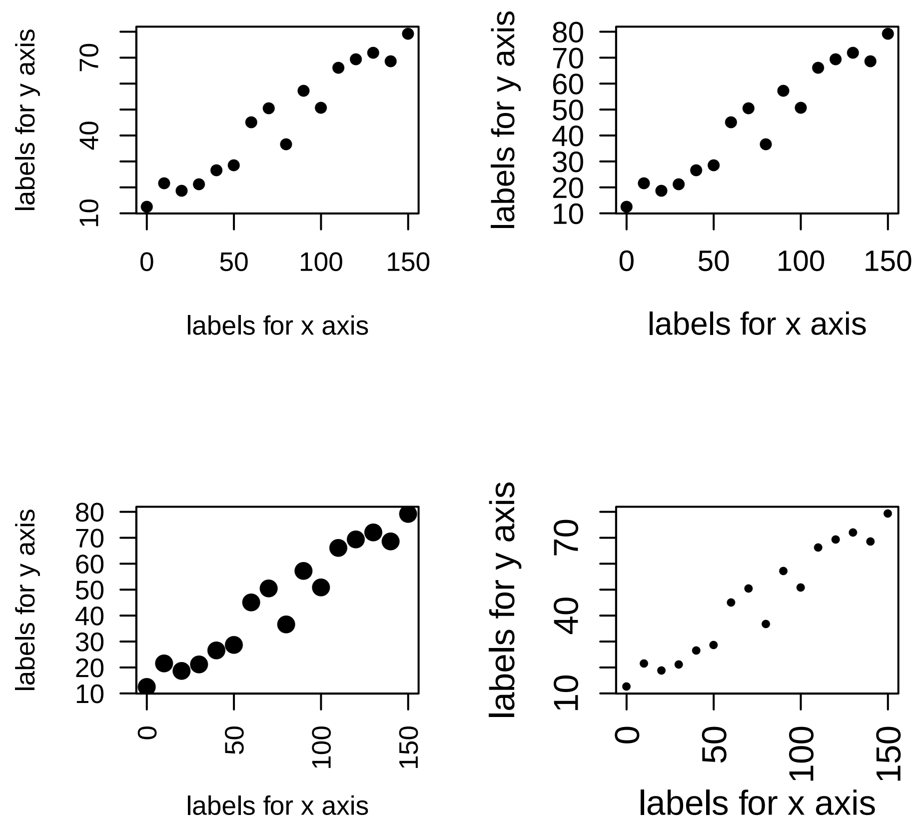
las determines the orientation of the numbers on the tick marks; 0: parallel with axis, 1: horizonal, 2: vertical with axis, 3: vertical. las=1 is better.

cex determines the size of plotting characters (pch);

cex.lab determines the size of the text labels on the axes;

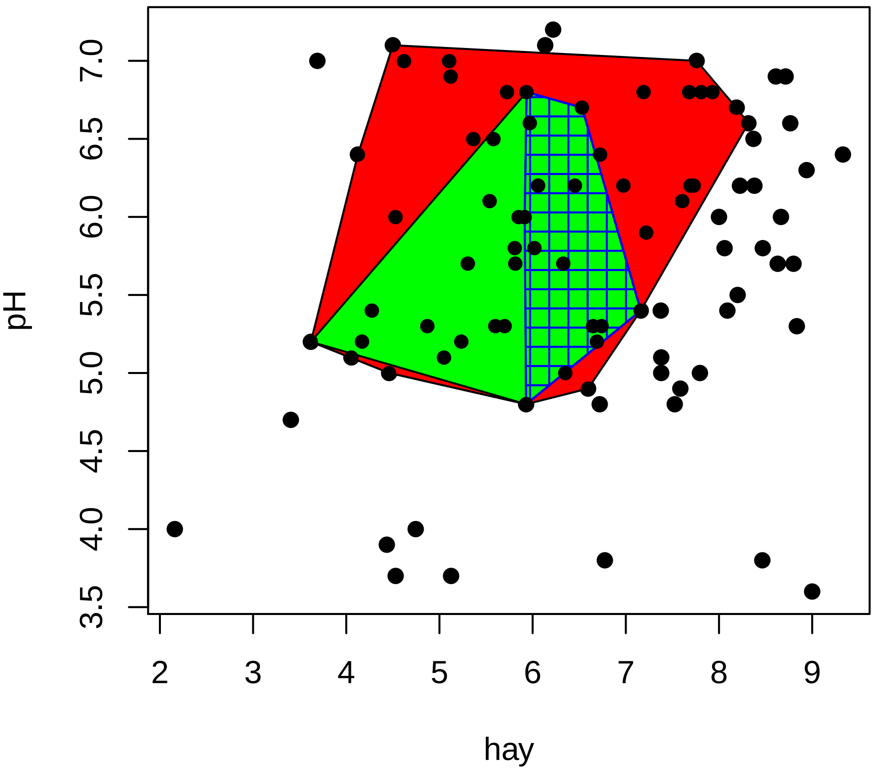
cex.axis   determines the size of the numbers on the tick marks.

```
> par(mfrow=c(2,2))
> x=seq(0,150,10); y=16+x*0.4+rnorm(length(x),0,6)
> plot(x,y,pch=16,xlab="labels for x axis",ylab="labels for y axis")
> plot(x,y,pch=16,xlab="labels for x axis",ylab="labels for y axis",
+ las=1,cex.lab=1.2,cex.axis=1.1)
> plot(x,y,pch=16,xlab="labels for x axis",ylab="labels for y axis",
+ las=2,cex=1.5)
> plot(x,y,pch=16,xlab="labels for x axis",ylab="labels for y axis",
+ las=3, cex=0.7,cex.lab=1.3,cex.axis=1.3)
```



Shading. default values: density=NULL; angle=45; border=NULL; col=NA; lty=par("lty", ...)

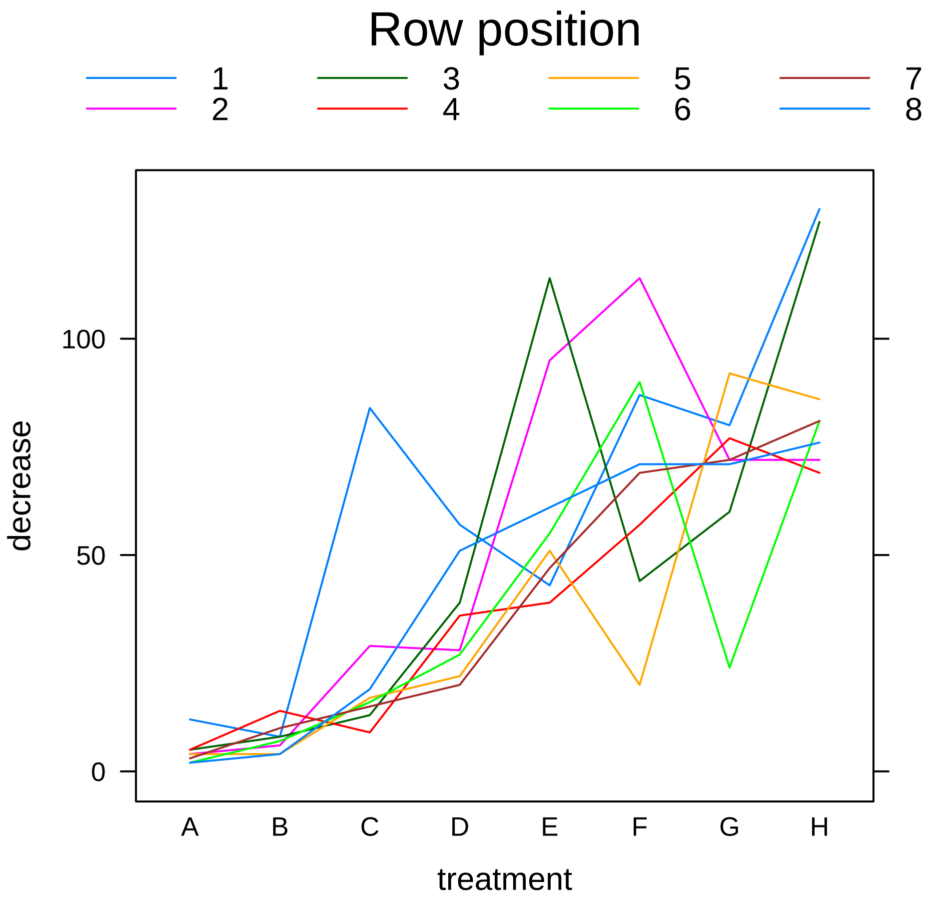
```
> res=read.table("F:\\2011fall courses\\stat571\\R\\therbook\\pgr.txt",header=T)
> attach(res)
> par(mfrow=c(1,1))
> plot(hay,pH)
> x=hay[FR>5]
> y=pH[FR>5]
> polygon(x[chull(x,y)],y[chull(x,y)],col="red")
> x=hay[FR>10]
> y=pH[FR>10]
> polygon(x[chull(x,y)],y[chull(x,y)],col="green")
> x=hay[FR>20]
> y=pH[FR>20]
> polygon(x[chull(x,y)],y[chull(x,y)],density=10,angle=90,col="blue")
> polygon(x[chull(x,y)],y[chull(x,y)],density=10,angle=0,col="blue")
> points(hay,pH,pch=16)
```



Logarithmic axes: You can transform the variables inside the plot function (e.g. plot(log(y)~x)) or you can plot the untransformed variables on logarithmically scaled axes (e.g. log="x", log="y", log="xy").

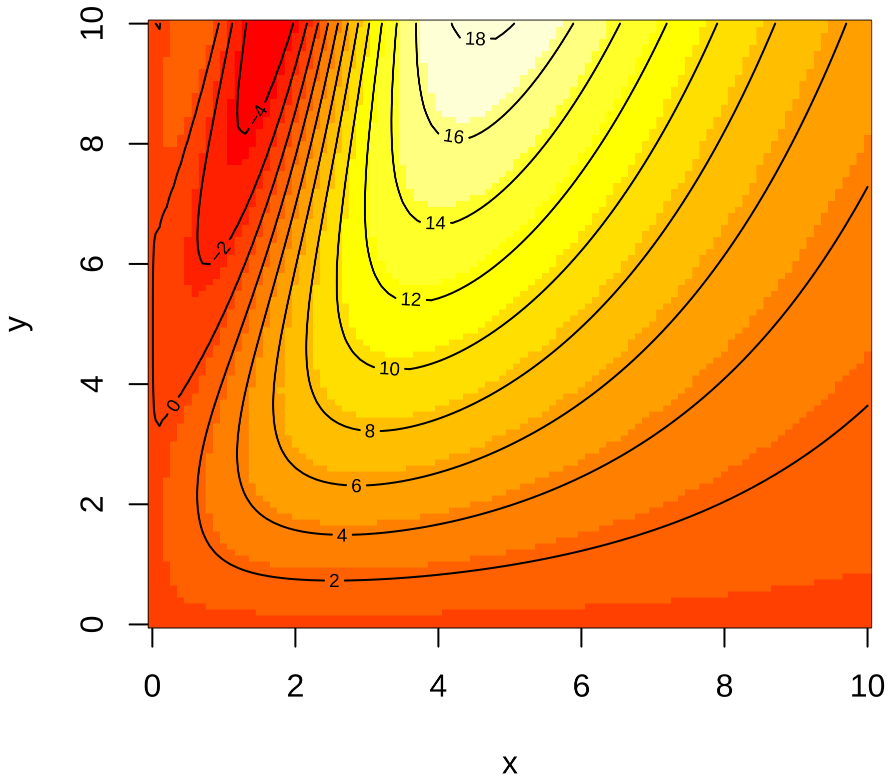
Axis labels containing subscripts and superscripts, you need to use `expression` function. plot(1:10, 1:10, ylab=expression(r^2), xlab=expression(x[i]), type="n").

```
> library(lattice)
> bwplot(decrease ~ treatment, OrchardSprays, groups = rowpos,
+ panel = "panel.superpose", #each group be drawn in a different colour
+ panel.groups = "panel.linejoin", #dots joined by lines for each member of the group
+ xlab = "treatment",
+ key = list(lines = Rows(trellis.par.get("superpose.line"), c(1:7, 1)),
+ text = list(lab = as.character(unique(OrchardSprays$rowpos))), columns = 4, title = "Ro
```



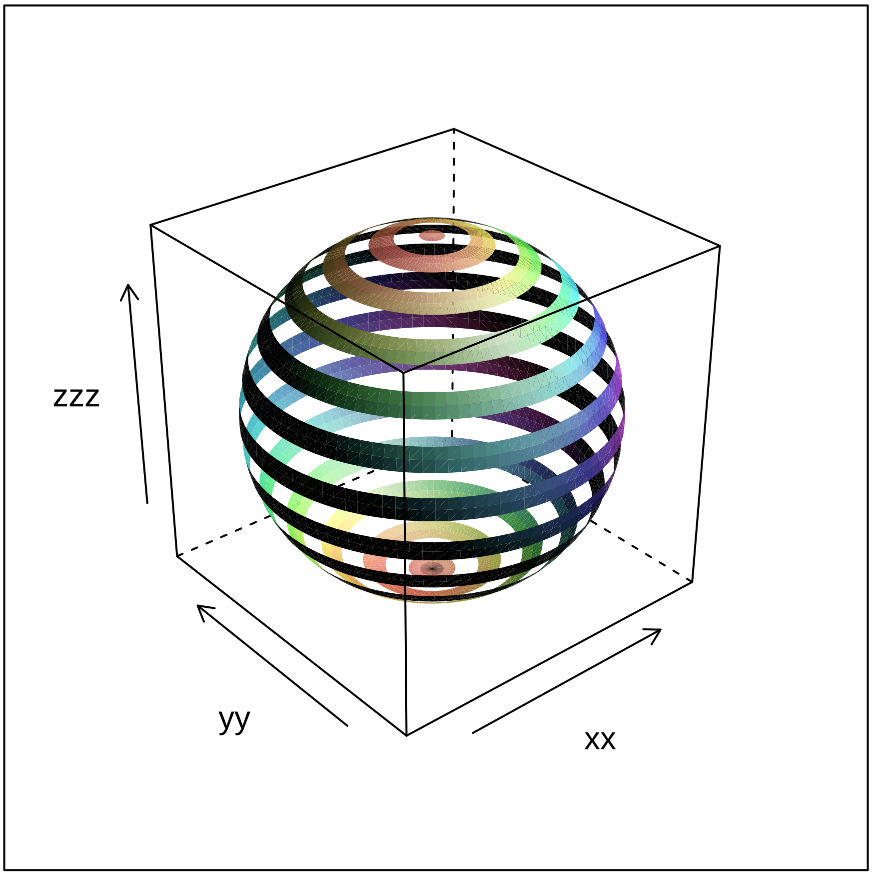
3-D plots: use the package "akima"

```
> library(akima)
> x=seq(0,10,.1);y=seq(0,10,.1)
> func=function(x,y)3*x*exp(.1*x)*sin(y*exp(-.5*x))
> image(x,y,outer(x,y,func))
> contour(x,y,outer(x,y,func),add=T)
```



Another example.

```
> library(lattice)
> n=50;tx=matrix(seq(-pi,pi,len=2*n),2*n,n)
> ty=matrix(seq(-pi,pi,len=n)/2,2*n,n,byrow=T)
> xx=cos(tx)*cos(ty);yy=sin(tx)*cos(ty);zz=sin(ty);zzz=zz;zzz[,1:12*4]=NA
> wireframe(zzz~xx*yy,shade=T,light.source=c(3,3,3))
```



An alphabetical tour of the graphics parameters

Which graphics attributes are changed with the par functions, which can be changed inside the plot function, and which stand alone? If you need to use par, then the graphics parameters should be altered ***before*** you use the first plot function. It is a good idea to save a copy of the default parameters settings so that you can be changed back at the end of the session to their default values. $default.parameters = par(no.readonly = T); par()...; par(default.parameters)$

When writing functions, you need to know things about the current plotting region. To inspect the current values of any of the graphics parameters (par), type the name of the option in double quotes. For instance:

par("usr") to see the current limits of the x and y axes. x-min, x-max, y-min, y-max.

par("mar") to see the sizes of the margins.

Parameter	In plot?	Default values	Meaning
adj	*	centred	justification of text
ann	*	TRUE	annotate plots with axis and overall titles?
ask		FALSE	pause before new graph?
bg	*	"transparent"	background style or colour
bty		full box	type of box drawn around the graph
cex	*	1	character expansion: enlarge if > 1, reduce if < 1
cex.axis	*	1	magnification for axis notation
cex.lab	*	1	magnification for label notation
cex.main	*	1.2	main title character size
cex.sub	*	1	sub-title character size
cin		0.1354167, 0.1875000	character size (wideth, height) in inches
col	*	"black"	colors() to see range of colours
col.axis		"black"	colour for graph axes
col.lab	*	"black"	colour for graph labels
col.main	*	"black"	colour fir main heading
col.sub	*	"black"	colour for sub-heading
cra		13, 18	character size (width, height) in rasters (pixels)
crt		0	rotation of single characters in degress (see srt)
csi		0.1875	character height in inches
cxy		0.02255379, 0.03452245	character size (width, height) in user-defined units
din		7.166666, 7.156249	size of the graphic device (width, height) in inches
family	*	"sans"	font styles: serif, sans, mono, and symbol
fg		"black"	colour for objects such as axes and boxes in the foreground
fig		0,1,0,1	coordinates of the figure region within the display region: c(x1,x2,y1,y2)
fin		7.16666, 7.156249	dimensions of the figure region (width, height) in inches
font	*	1	regular=1, bold=2, italics=3,bold&italics=4
font.axis	*	1	font in which axis is numbered
font.lab	*	1	font in which labels are written
font.main	*	1	font for main heading
font.sub-title	*	1	font for sub-heading
gamma		1	correction for hsv colours
hsv		1 1 1	values (range[0,1]) for hue, saturation and value of colour
lab		5 5 7	number of tick marks on the x axis, y axis and size of labels

continuous...

Parameter	In plot?	Default values	Meaning
las		0	orientation of axis numbers, use las=1 for publication
lend		“round”	style for the ends of lines; could be “square” or “butt”
lheight		1	height of a line of text used to vertically space multi-line text
ljoin		“round”	style for joining two lines; could be “mitre” or “bevel”
lmitre		10	controls when mitred line joins are automatically converted into bevelled line joins
log	*	neither	which axes to log: log=”x”, log=”y”, log=”xy”
lty	*	“solid”	line type (e.g. dashed: lty=2)
lwd	*	1	width of lines on a graph
mai		0.95625, 0.76875, 0.76875, 0.39375	margin sizes in inches for c(bottom, left, top, right)
mar		5.1, 4.1, 4.1, 2.1	margin sizes in numbers of lines for c(bottom, left, top, right)
mex		1	margin expansion specifies the size of font used to convert between “mar” and “mai”, and between “oma” and “omi”
mfcol		1 1	number of graphs per page, produced by columnwise
mfrow		1 1	multiple graphs per page. mfrow=c(2, 3) gives 2 rows, 3 columns, drawn row-wise
mfg		1 1 1 1	which figure in an array of figures is to be drawn next (if setting) or is being drawn (if enquiring); the array must already have been set by mfcolor mfrow
mgp		3 1 0	margin line (in mex units) for the axis title, axis labels and axis line
new		FALSE	to draw another plot on top of the existing plot, set new=TRUE so that plot does not wipe the slate clean
oma		0 0 0 0	size of the outer margins in lines of text. c(bottom, left, top, right)
omd		0 1 0 1	size of the outer margins in normalized device coordinate (NDC) units, expressed as a fraction (in [0,1]) of the device region. c(bottom, left, top, right)
omi		0 0 0 0	size of the outer margins in inches. c(bottom, left, top, right)

continuous...

Parameter	In plot?	Default values	Meaning
pch	*	1	plotting symble; e.g. pch=16
pin		6.004166, 5.431249	current plot dimensions (width, height), in inches
plt		0.1072675, 0.9450582, 0.1336245, 0.8925764	coordinates of the plot region as fractions of the current figure region c(x1, x2, y1, y2)
ps		12	point size of text and asymbols
pty		"m"	type of plot region to be used: pty="s" generate a square plotting region, "m" stands for maximal.
srt	*	0	string rotation in degrees
tck		tcl=-0.5	big tick marks (grid-lines); to use this set tcl=NA
tcl		-0.5	tick marks outside the frame
tmag		1.2	enlargement of text of the main title relative to the other annotating text of the plot.
type	*	"p"	plot type. r.g. type="n" to produce blank axes.
usr		set by the last plot function	extremes of the user-defined coorfinates of the plottiong region
xaxp, yaxp		0 1 5	tick marks for kog axes: xmin, xmax, and number of intervals
xaxs, yaxs		"r"	pretty x axis intervals
xaxt, yaxt		"s"	x axis type: use xaxt="n" to set up the axis but not plot it.
xlab, ylab	*	label for the x axis	xlab="labels"
xlim, ylim	*	pretty	user control of x axis scaling: xlim=c(0,1)
xlog, ylog		FALSE	is the x axis on a log scale? log="x", log="y", log="xy"
xpd		FALSE	the way plotting is clipped: if FALSE all plotting is clipped to the plot region; if TRUE all plottiong is clipped to the figure region, and if NA all plotting is clipped to the device region.
yaxp		0 1 5	tick marks for kog axes: ymin, ymax, and number of intervals

See more at other books.