

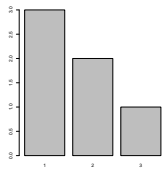
Theory III: graphics in R

- the plot function
- graphical parameters
- histograms and boxplot
- plots of surfaces

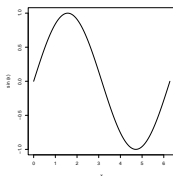
1

The generic plot command

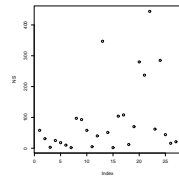
```
> plot(factor(c(1,2,3,1,1,2)))
```



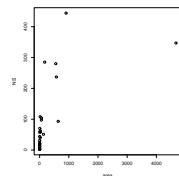
```
> plot(sin,0,pi)
```



```
> plot(NS)
```



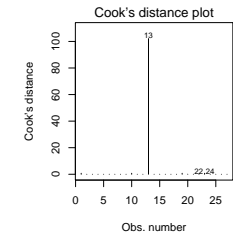
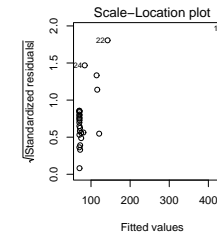
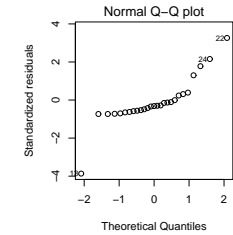
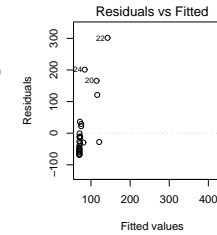
```
> plot(area,NS)
```



2

plot on lm object:

```
> fit=lm(NS~area)
> par(mfrow=c(2,2))
> plot(fit)
```



3

Graphical parameters: can be set using `par()` or within `plot()`

type of plot (line, points,...)	type	plotting character: <code>pch</code>
labels: <code>xlab</code> , <code>ylab</code>		limits for axes: <code>xlim</code> , <code>ylim</code>
shape of plot region: <code>pty</code>		colour: <code>col</code>
multiple plots: <code>mfrow</code>		style for text on axes: <code>las</code>

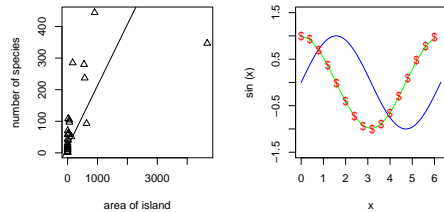
and *many* others (`help(par)`)

Consistency: graphical parameter same effect in all graphics functions

4

Example: graphical parameters and `abline()`, `points()` and `lines()`

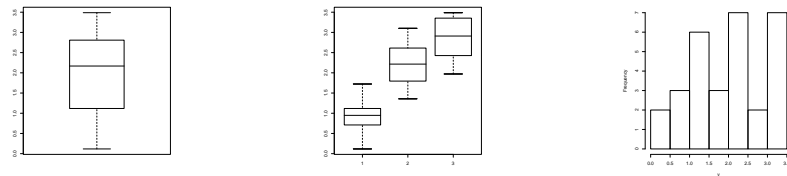
```
> par(mfrow=c(1,2),pty="s")
> plot(area,NS,xlab="area of island",ylab="number of species")
> abline(fit) #add regression line for lm object fit
> plot(sin,0,2*pi,ylim=c(-1.5,1.5),col="blue")
> x=seq(0,2*pi,0.4)
> lines(x,cos(x),col="green") #add line to current plot
> points(x,cos(x),pch="$",col="red") #add points to current plot
```



Histograms and boxplots

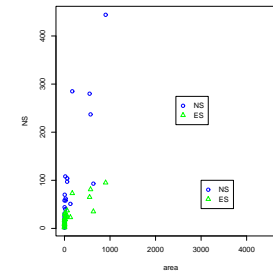
Synthetic data:

```
> group=rep(c(1,2,3),rep(10,3))
> y=group+rnorm(30,sd=0.5)
> boxplot(y) > boxplot(y~factor(group)) >hist(y)
```



Adding a legend

```
> plot(area,NS,pch=1,col="blue")
> points(area,ES,pch=2,col="green")
> legend(3000,100,legend=c("NS","ES"),pch=c(1,2),
+ col=c("blue","green"))
> lc=locator(1) #interactive graphics - click at position in plot
> legend(lc,legend=c("NS","ES"),pch=c(1,2),col=c("blue","green"))
```



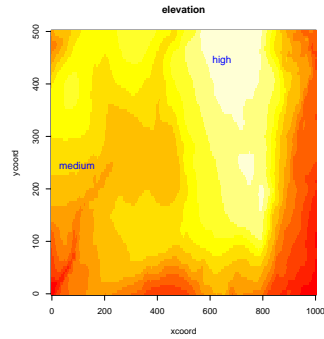
Plots of surfaces: image, contour, persp

Data for surface:

```
> elevation=read.table("elevdata.txt",header=T)
> names(elevation)
[1] "x" "y" "elev"
> attach(elevation)
> xcoord=unique(x) #the unique values in x
> ycoord=unique(y) #if not sorted we may use sort(unique(y))
> length(xcoord)
[1] 201
> length(ycoord)
[1] 101
```

Example: `image()` and `text()`

```
> elevmat=matrix(elev,ncol=101,byrow=T)
> image(xcoord,ycoord,elevmat)
> text(xcoord[c(20,130)],ycoord[c(50,90)],
+ c("medium","high"),col="blue",cex=1.2)
```



9

11

Exporting plots: `postscript` (latex), `dev.copy2eps`, `pdf`,
`win.metafile` (word), `png` (web)

Example: plot to postscript file

```
> postscript(file="xy.ps")
> plot(x,y)
> dev.off()
or
> plot(x,y)
> dev.copy2eps(file="xy.ps")
```

Windows metafile:

```
> win.metafile(file="xy.wmf")
> plot(x,y)
> dev.off()
```

10

Other useful functions

- `segments` draw line segments
- `symbols` draw symbols of varying size
- `identify` (interactive) identifies point in a scatterplot:

```
> plot(x,y)
> identify(x,y) #use left mouse button to identify
                #points; e.g. first point
[1] 1 #terminates when middle or right button is pressed
```

12