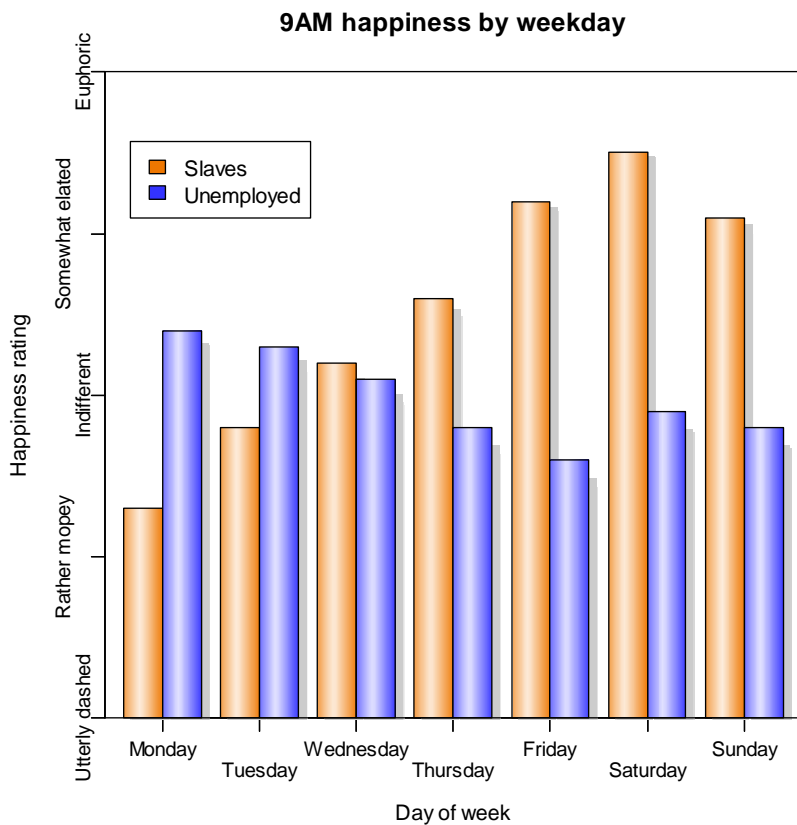


```

# get some extra room on the left
par(mar=c(5,5,4,2))
# make up some happiness data, as so many seem to do
happyday<-data.frame(Monday=c(2.3,3.4),Tuesday=c(2.8,3.3),Wednesday=c(3.2,3.1),
Thursday=c(3.6,2.8),Friday=c(4.2,2.6),Saturday=c(4.5,2.9),Sunday=c(4.1,2.8))
happylabels<-c("Utterly dashed","Rather mopey","Indifferent","Somewhat elated",
"Euphoric")
barp(happyday,names.arg=names(happyday),legend.lab=c("Slaves","Unemployed"),
legend.pos="e",col=c("#ee7700","#3333ff"),main="9AM happiness by weekday",
xlab="Day of week",ylab="Happiness rating",ylim=c(1,5),staxx=TRUE,staxy=TRUE,
height.at=1:5,height.lab=happylabels,cex.axis=0.9,cylindrical=TRUE,
shadow=TRUE)

```

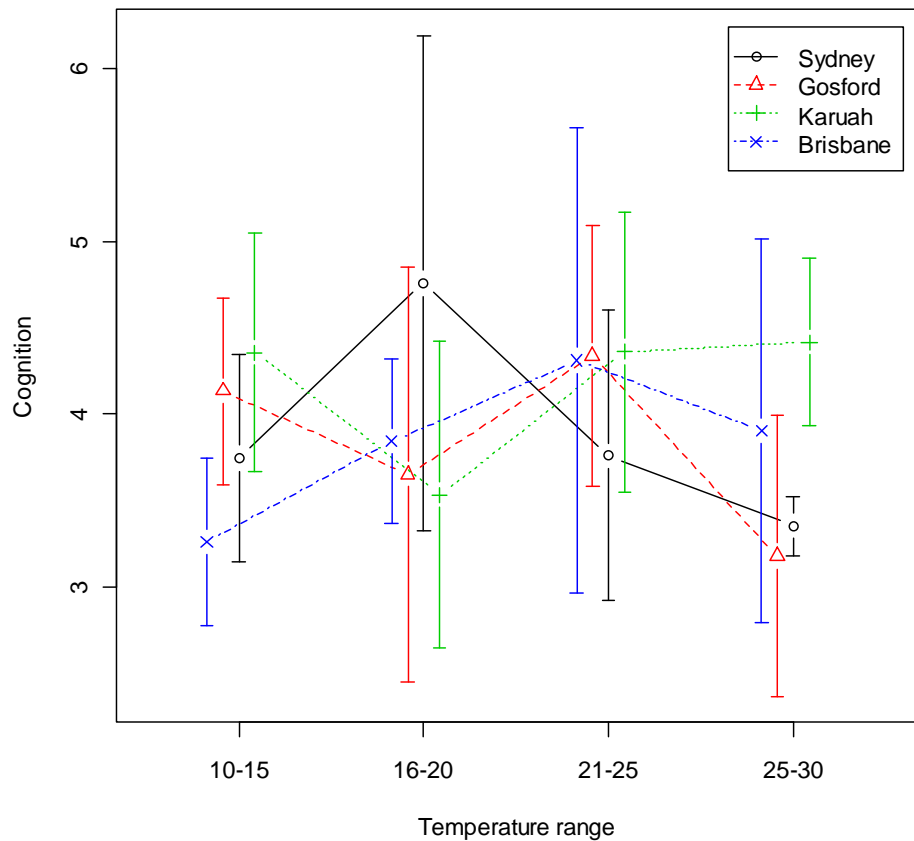


```

test.df<-data.frame(a=rnorm(80)+4,b=rnorm(80)+4,c=rep(LETTERS[1:4],each=20),
d=rep(rep(letters[1:4],each=4),5))
# first use the default values
brkdn.plot("a", "c", "d", test.df, pch=1:4, col=1:4)
# now jazz it up a bit using medians and median absolute deviations
# and some enhancements
bp<-brkdn.plot("a", "c", "d", test.df, main="Test of the breakdown plot",
mct="median", md="mad", xlab="Temperature range", ylab="Cognition",
xaxlab=c("10-15", "16-20", "21-25", "25-30"), pch=1:4, lty=1:4, col=1:4)
es<-emptyspace(bp)
legend(es, legend=c("Sydney", "Gosford", "Karuah", "Brisbane"), pch=1:4,
col=1:4, lty=1:4, xjust=0.5, yjust=0.5)

```

Test of the breakdown plot

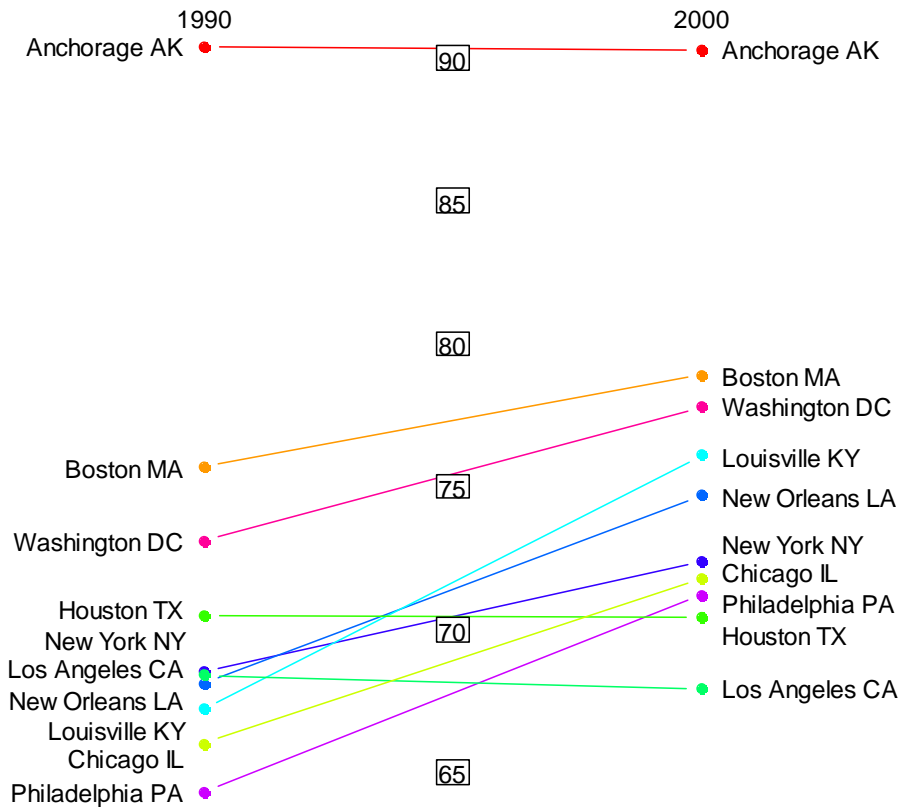


```

# percentage of those over 25 years having completed high school
# in 10 cities in the USA in 1990 and 2000
educattn<-matrix(c(90.4,90.3,75.7,78.9,66,71.8,70.5,70.4,68.4,67.9,
67.2,76.1,68.1,74.7,68.5,72.4,64.3,71.2,73.1,77.8),ncol=2,byrow=TRUE)
rownames(educattn)<-c("Anchorage AK","Boston MA","Chicago IL",
"Houston TX","Los Angeles CA","Louisville KY","New Orleans LA",
"New York NY","Philadelphia PA","Washington DC")
colnames(educattn)<-c(1990,2000)
bumpchart(educattn,main="Rank for high school completion by over 25s")
# now show the raw percentages and add central ticks
bumpchart(educattn,rank=FALSE,
main="Percentage high school completion by over 25s",col=rainbow(10))
# margins have been reset, so use
par(xpd=TRUE)
boxed.labels(1.5,seq(65,90,by=5),seq(65,90,by=5))
par(xpd=FALSE)

```

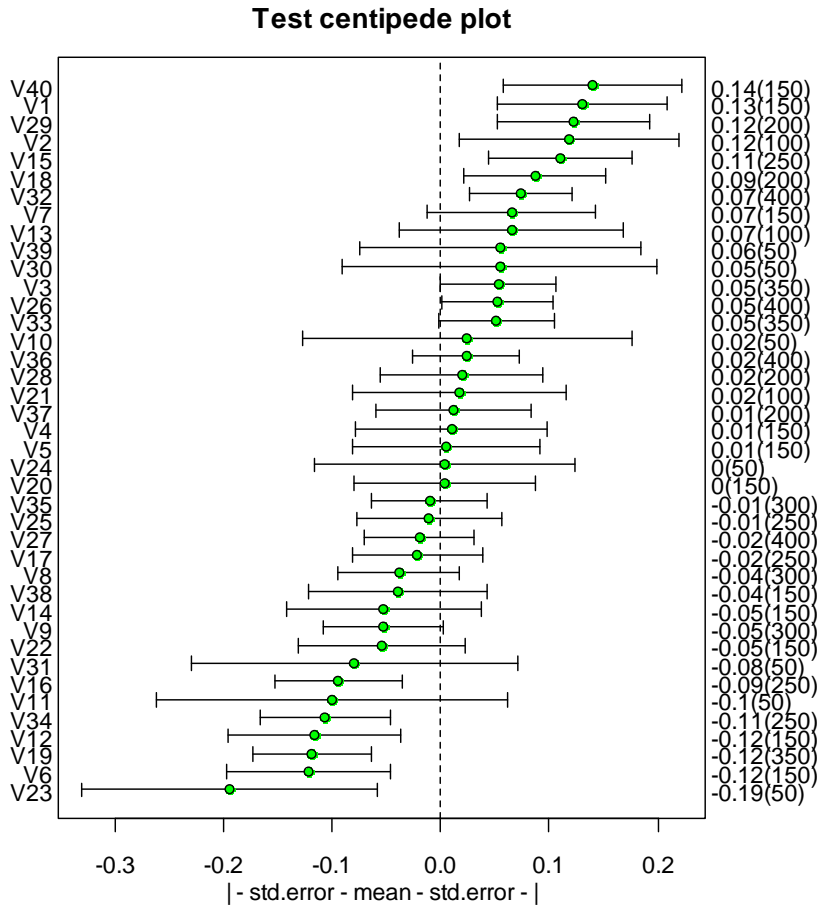
Percentage high school completion by over 25s



```

testcp<-list("",40)
for(i in 1:40) testcp[[i]]<-rnorm(sample(1:8,1)*50)
segs<-get.segs(testcp)
centipede.plot(segs,main="Test centipede plot",vgrid=0)

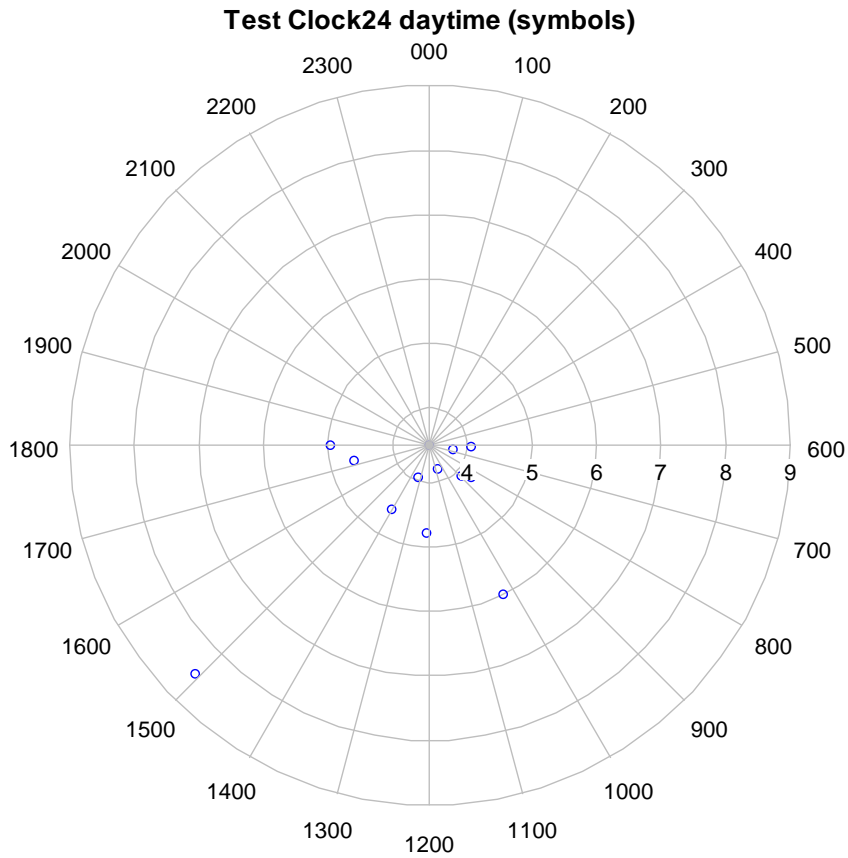
```



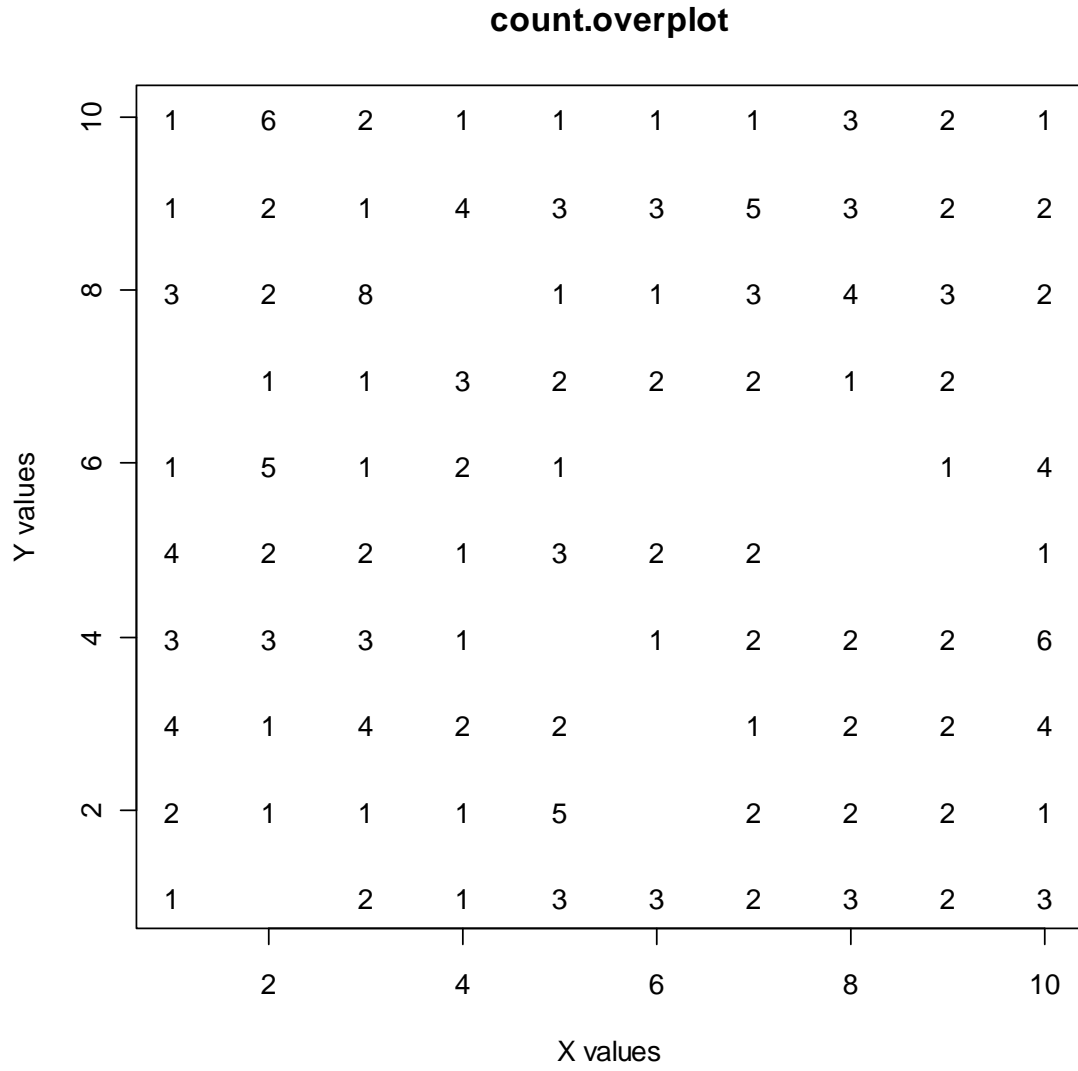
```

testlen<-rnorm(24)*2+5
testpos<-0:23+rnorm(24)/4
clock24.plot(testlen,testpos,main="Test Clock24 (lines)",show.grid=FALSE,
line.col="green",lwd=3)
if(dev.interactive()) par(ask=TRUE)
# now do a 'daylight' plot
clock24.plot(testlen[7:19],testpos[7:19],
main="Test Clock24 daytime (symbols)",
point.col="blue",rp.type="s",lwd=3)

```



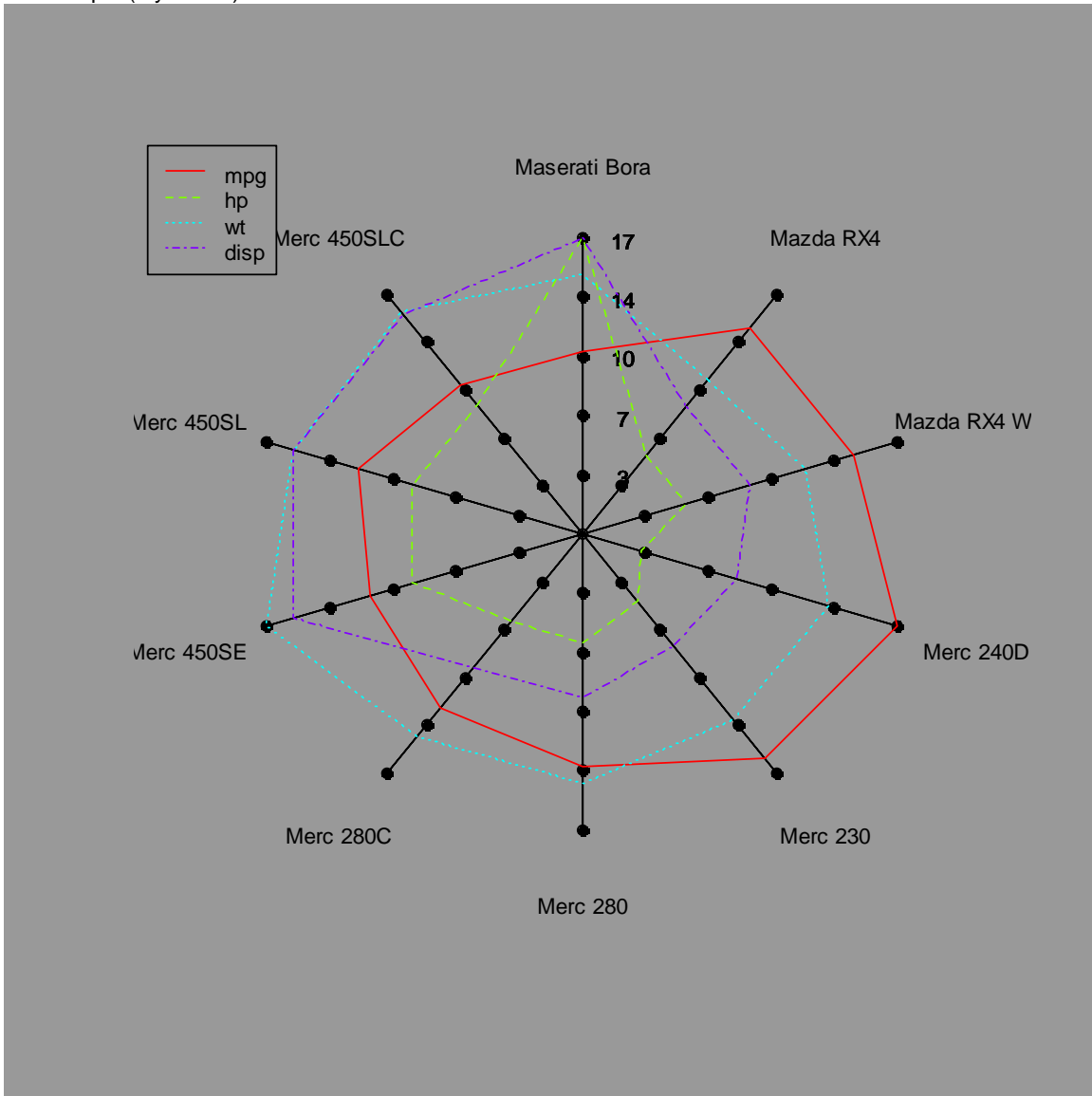
```
xy.mat<-cbind(sample(1:10,200,TRUE),sample(1:10,200,TRUE))
count.overplot(xy.mat,main="count.overplot",
xlab="X values",ylab="Y values")
```



```

data(mtcars)
mysubset<-mtcars[substr(dimnames(mtcars)[[1]],1,1)=="M",c("mpg","hp","wt","disp")]
diamondplot(mysubset)

```

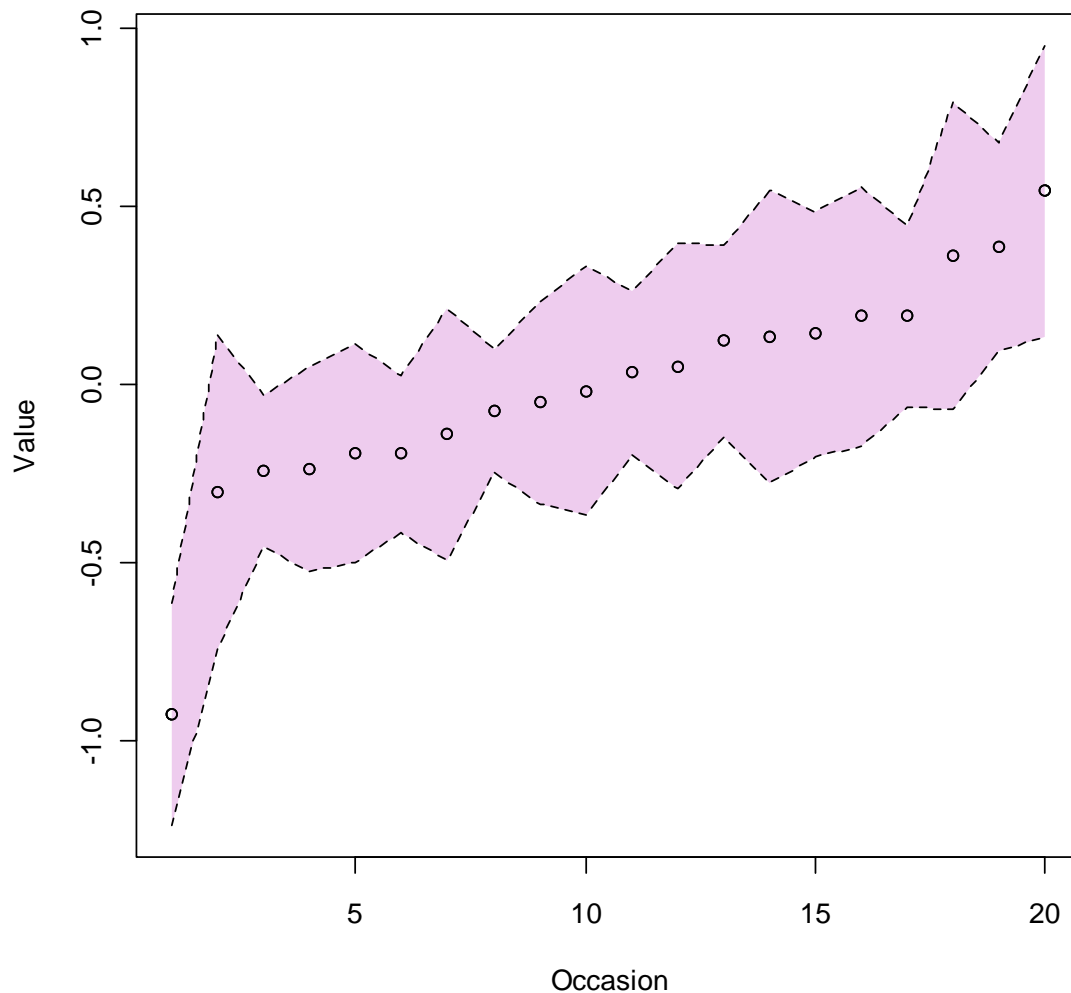


```

disptest<-matrix(rnorm(200),nrow=20)
disptest.means<-rowMeans(disptest)
row.order<-order(disptest.means)
se.disptest<-unlist(apply(disptest,1,std.error))
plot(disptest.means[row.order],main="Dispersion as error bars",
ylim=c(min(disptest.means-se.disptest),max(disptest.means+se.disptest)),
xlab="Occasion",ylab="Value")
dispersion(1:20,disptest.means[row.order],se.disptest[row.order])
plot(disptest.means[row.order],main="Dispersion as confidence band",
ylim=c(min(disptest.means-se.disptest),max(disptest.means+se.disptest)),
xlab="Occasion",ylab="Value")
dispersion(1:20,disptest.means[row.order],se.disptest[row.order],type="l",
fill="#eeccee",lty=2)
# remember to redraw the points
points(disptest.means[row.order])

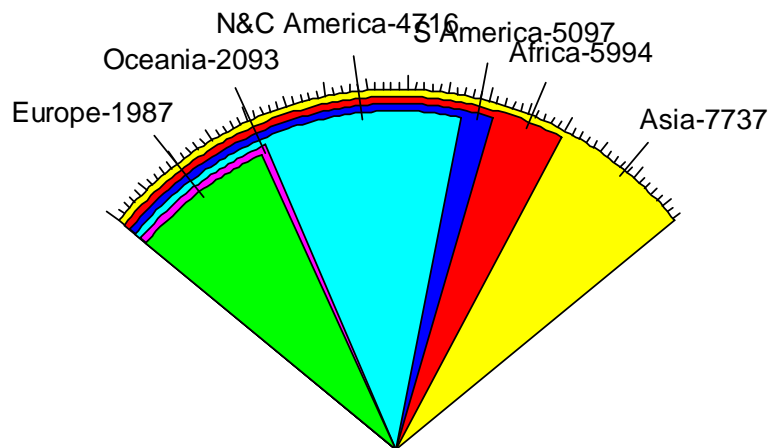
```

Dispersion as confidence band



```
iucn.df<-data.frame(area=c("Africa","Asia","Europe","N&C America",  
"S America","Oceania"),threatened=c(5994,7737,1987,4716,5097,2093))  
fan.plot(iucn.df$threatened,  
labels=paste(iucn.df$area,iucn.df$threatened,sep="-"),  
main="Threatened species by geographical area",ticks=276)
```

Threatened species by geographical area

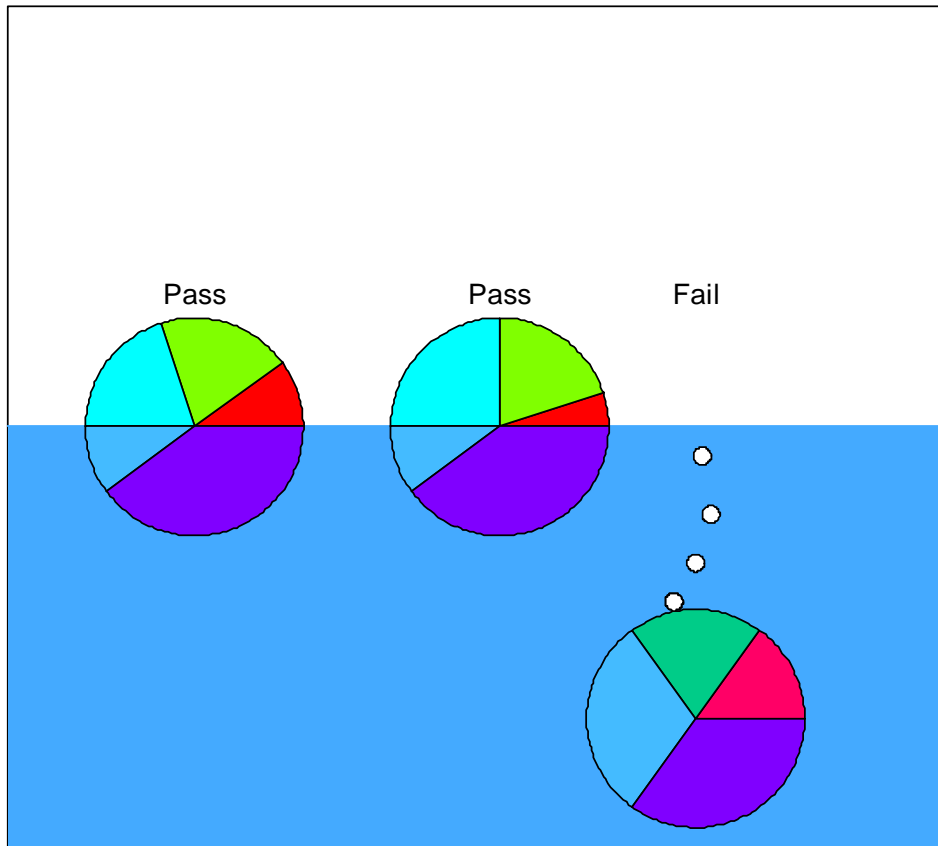


```

plot(1:5,type="n",main="Floating Pie test",xlab="",ylab="",axes=FALSE)
box()
polygon(c(0,0,5.5,5.5),c(0,3,3,0),border="#44aaff",col="#44aaff")
floating.pie(1.7,3,c(2,4,4,2,8),radius=0.5,
col=c("#ff0000","#80ff00","#00ffff","#44bbff","#8000ff"))
floating.pie(3.1,3,c(1,4,5,2,8),radius=0.5,
col=c("#ff0000","#80ff00","#00ffff","#44bbff","#8000ff"))
floating.pie(4,1.5,c(3,4,6,7),radius=0.5,
col=c("#ff0066","#00cc88","#44bbff","#8000ff"))
draw.circle(3.9,2.1,radius=0.04,col="white")
draw.circle(3.9,2.1,radius=0.04,col="white")
draw.circle(3.9,2.1,radius=0.04,col="white")
draw.circle(4.2,3,radius=0.04,col="white")
draw.circle(4.07,2.55,radius=0.04,col="white")
draw.circle(4.03,2.85,radius=0.04,col="white")
text(c(1.7,3.1,4),c(3.7,3.7,3.7),c("Pass","Pass","Fail"))

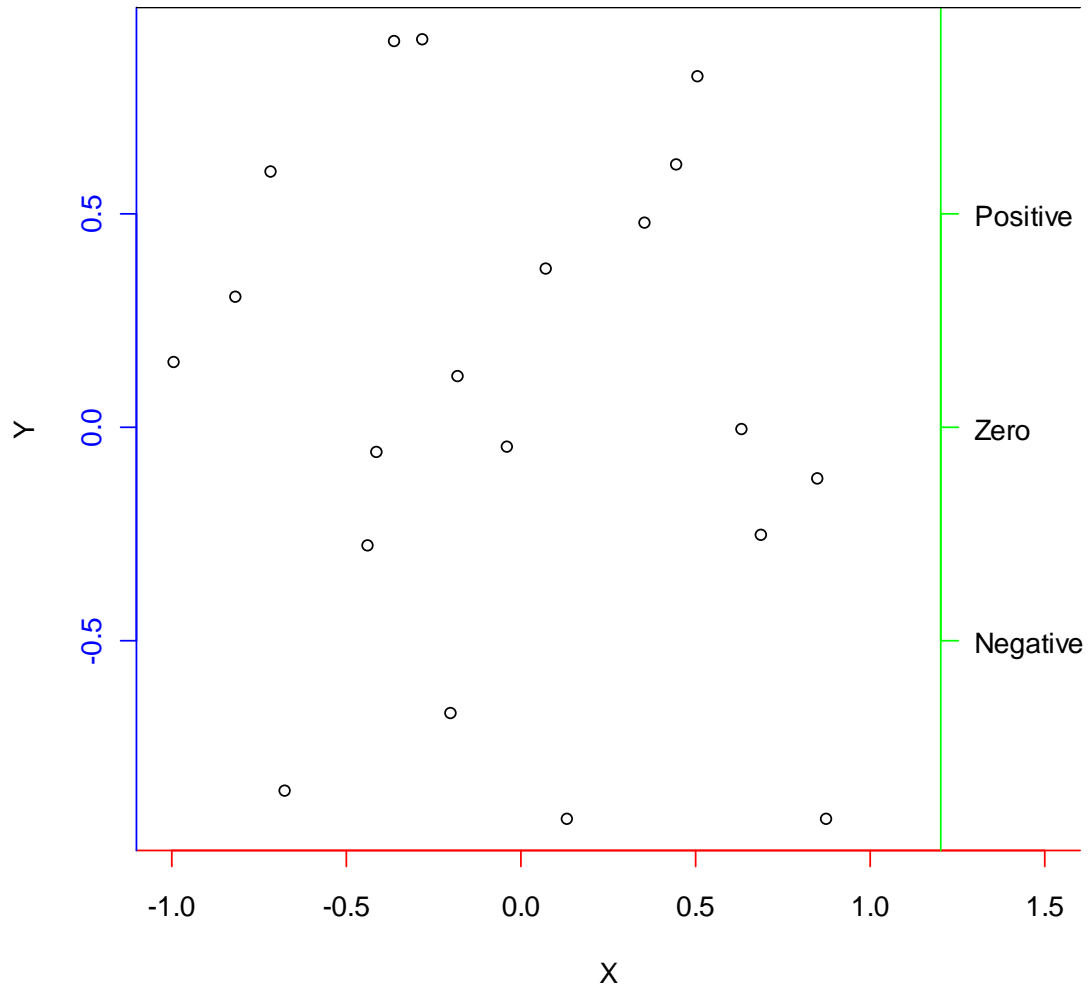
```

Floating Pie test



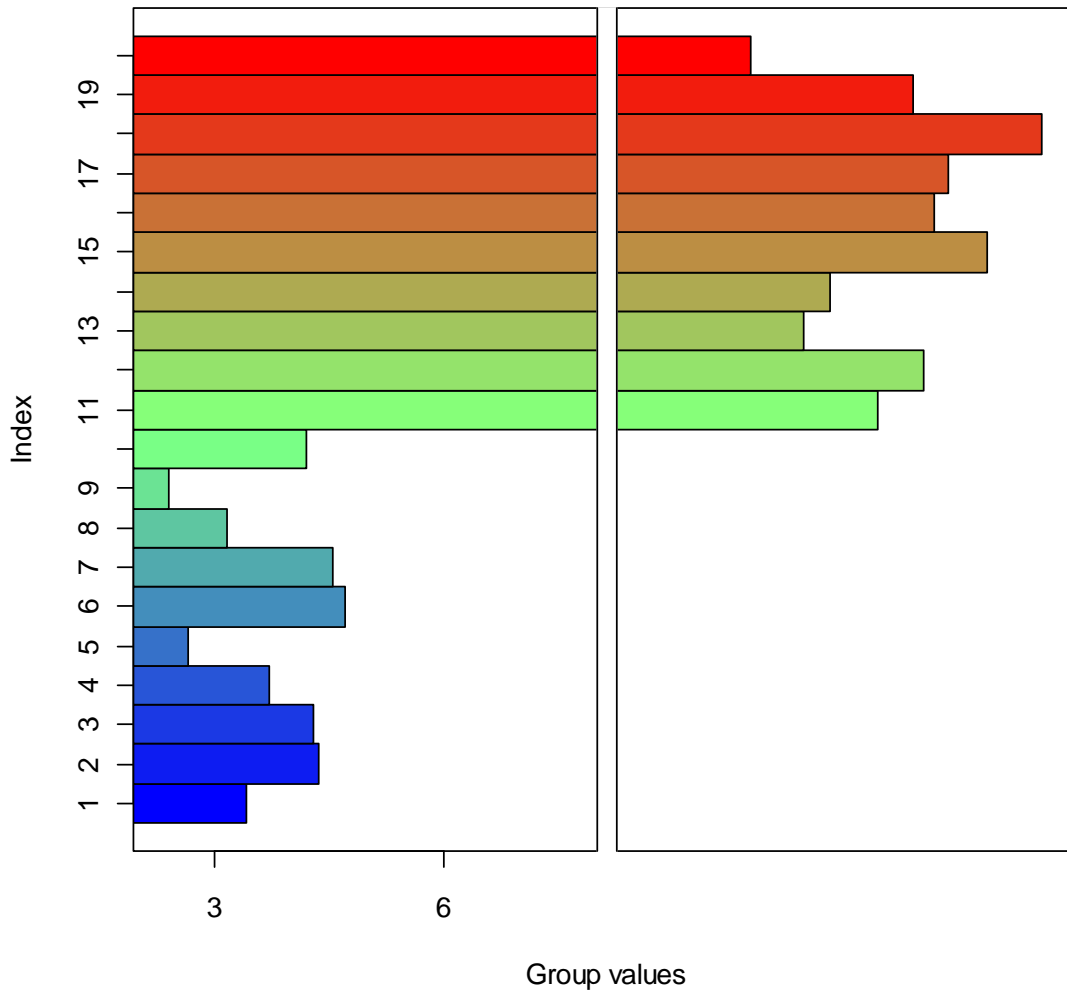
```
plot(runif(20,-1,1),runif(20,-1,1),xlim=c(-1,1.5),main="Demo of fullaxis",
     xlab="X",ylab="Y",axes=FALSE)
fullaxis(1,col="red")
fullaxis(2,col="blue",col.axis="blue")
fullaxis(4,at=c(-0.5,0,0.5),labels=c("Negative","Zero","Positive"),pos=1.2,
     col="green",las=1)
# add a top line to complete the "box"
xylim<-par("usr")
segments(xylim[1],xylim[4],xylim[2],xylim[4])
```

Demo of fullaxis



```
twogrp<-c(rnorm(10)+4,rnorm(10)+20)
gap.barplot(twogrp,gap=c(8,16),xlab="Index",ytics=c(3,6,17,20),
ylab="Group values",main="Barplot with gap")
gap.barplot(twogrp,gap=c(8,16),xlab="Index",ytics=c(3,6,17,20),
ylab="Group values",horiz=TRUE,main="Horizontal barplot with gap")
```

Horizontal barplot with gap

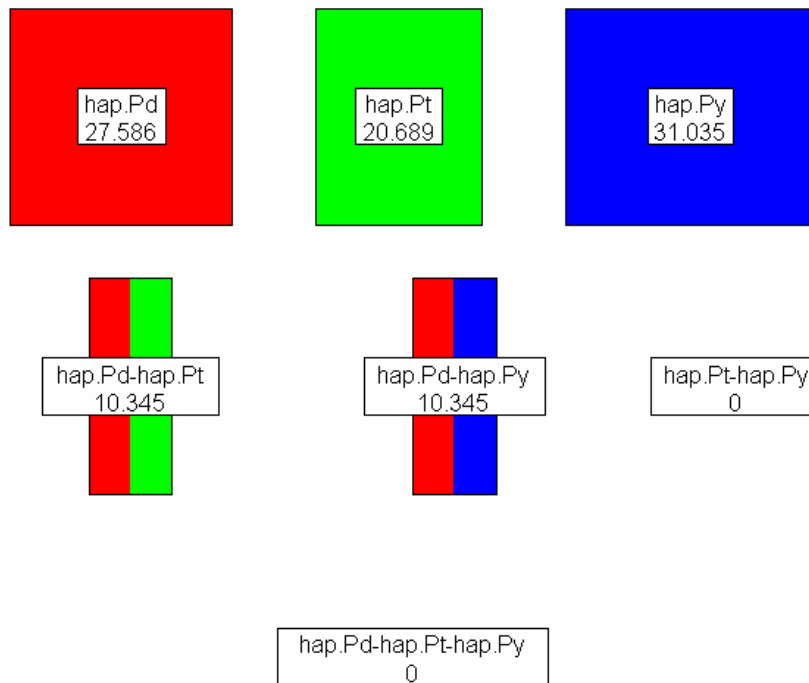


```

# this example is from a haplotype mapping problem submitted by Mao Jianfeng
## Not run:
hapIntList<-
getIntersectList(3,xnames=c("hap.Pd","hap.Pt","hap.Py"))
# enter the data as follows:
# Number of elements in hap.Pd - 1: 27.586
# Number of elements in hap.Pt - 1: 20.689
# Number of elements in hap.Py - 1: 31.035
# Number of elements in hap.Pd-hap.Pt - 1: 10.345
# Number of elements in hap.Pd-hap.Py - 1: 10.345
# Number of elements in hap.Pt-hap.Py - 1: 0
# Number of elements in hap.Pd-hap.Pt-hap.Py - 1: 0
# Total number of elements - 1: 100
## End(Not run)
hapIntList<-structure(list(structure(c(27.586, 20.689, 31.035),
.Names = c("hap.Pd","hap.Pt","hap.Py")),
structure(c(10.345, 10.345, 0),
.Names = c("hap.Pd-hap.Pt","hap.Pd-hap.Py","hap.Pt-hap.Py")),
structure(0, .Names = "hap.Pd-hap.Pt-hap.Py"),100),
class = "intersectList")
intersectDiagram(hapIntList)

```

Intersection Diagram

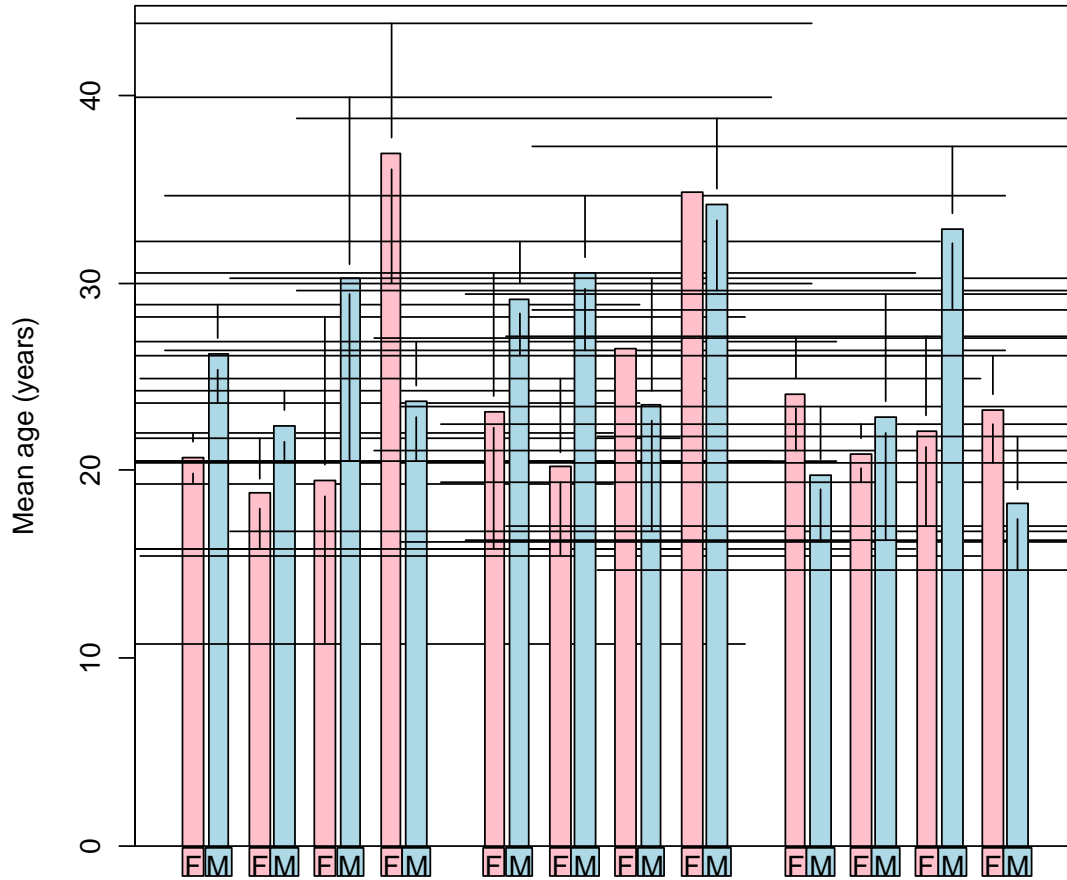


```

test.df<-data.frame(Age=rnorm(100,25,10),
Sex=sample(c("M","F"),100,TRUE),
Marital=sample(c("D","M","S","W"),100,TRUE),
Employ=sample(c("Full Time","Part Time","Unemployed"),100,TRUE))
test.col<-list(Overall="gray",Employ=c("#1affd8","#caeec","#f7b3cc"),
Marital=c("mediumpurple","orange","tan","lightgreen"),Sex=c("pink","lightblue"))
hierobarp(formula=Age~Sex+Marital+Employ,data=test.df,ylab="Mean age (years)",
main="Show only the final breakdown",errbars=TRUE,col=test.col$Sex)

```

Show only the final breakdown

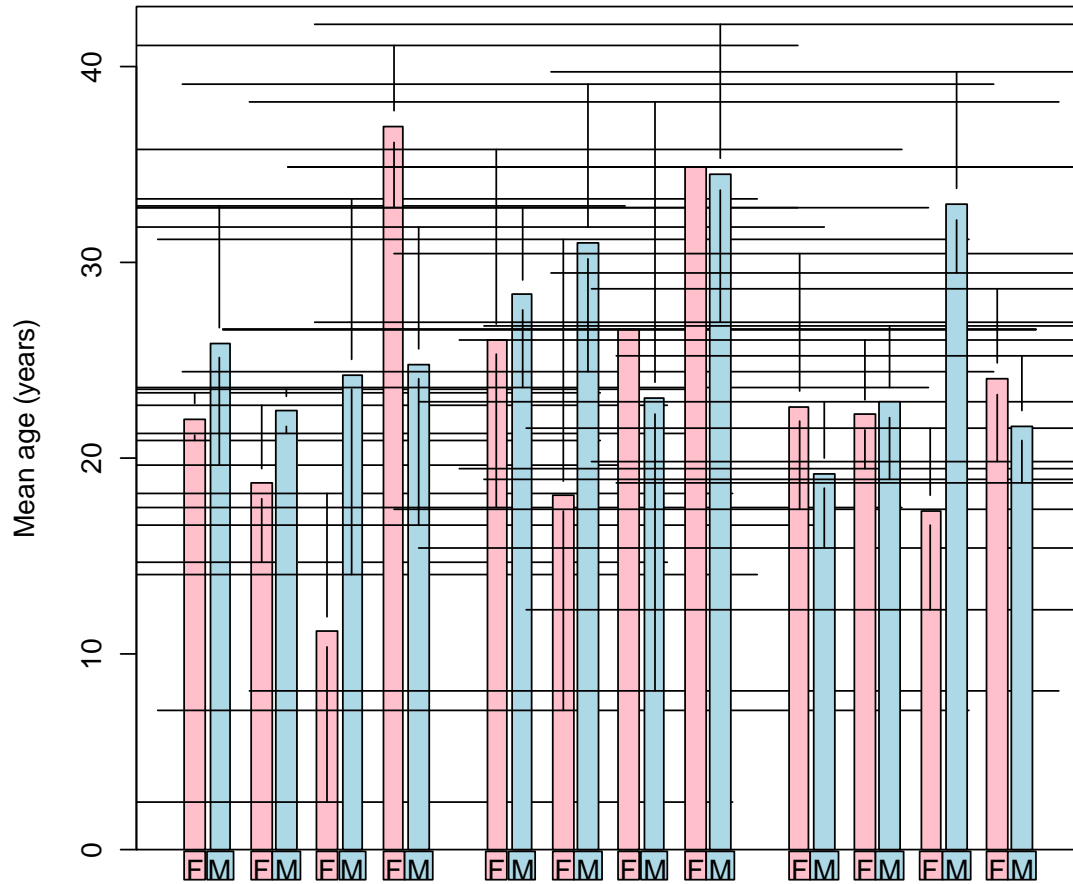


```

# set up functions for 20 and 80 percentiles - must be offsets, not limits
q20<-function(x,na.rm=TRUE) return(mean(x)-quantile(x,probs=0.2,na.rm=TRUE))
q80<-function(x,na.rm=TRUE) return(quantile(x,probs=0.8,na.rm=TRUE)-mean(x))
# show the asymmetric dispersion measures
hierobarp(formula=Age~Sex+Marital+Employ,data=test.df,ylab="Mean age (years)",
main="Use median and quantiles for dispersion",mct=median,lmd=q20,umd=q80,
errbars=TRUE,col=test.col$Sex)

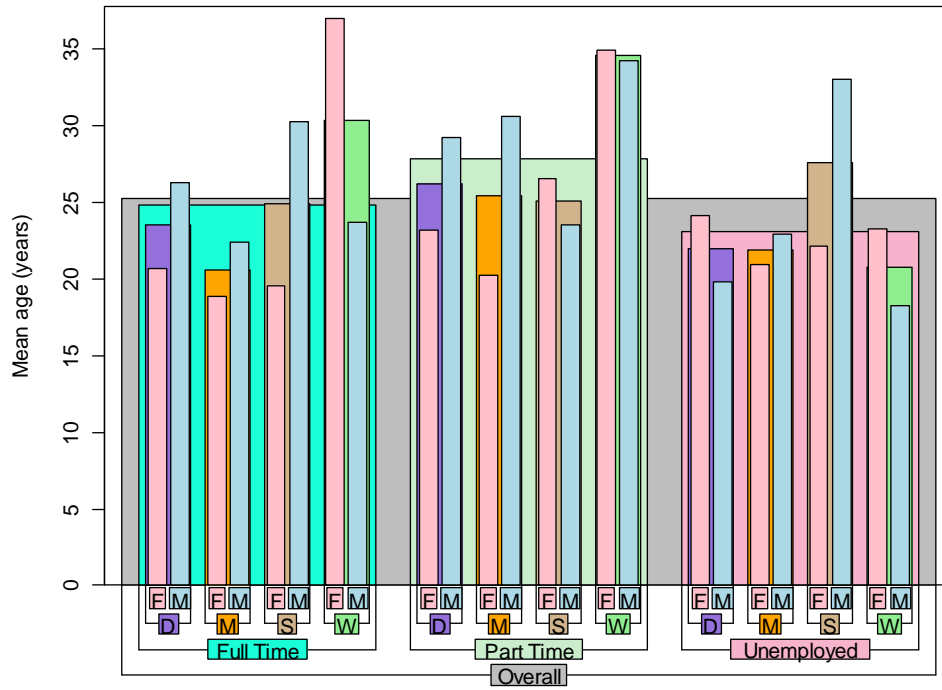
```

Use median and quantiles for dispersion



```
# start a wide plot window for this one
x11(width=10)
hierobarp(formula=Age~Sex+Marital+Employ,data=test.df,ylab="Mean age (years)",
main="Show the entire hierarchical breakdown",col=test.col,showall=TRUE,
showbrklab=TRUE,mar=c(5,4,4,8))
```

Show the entire hierarchical breakdown

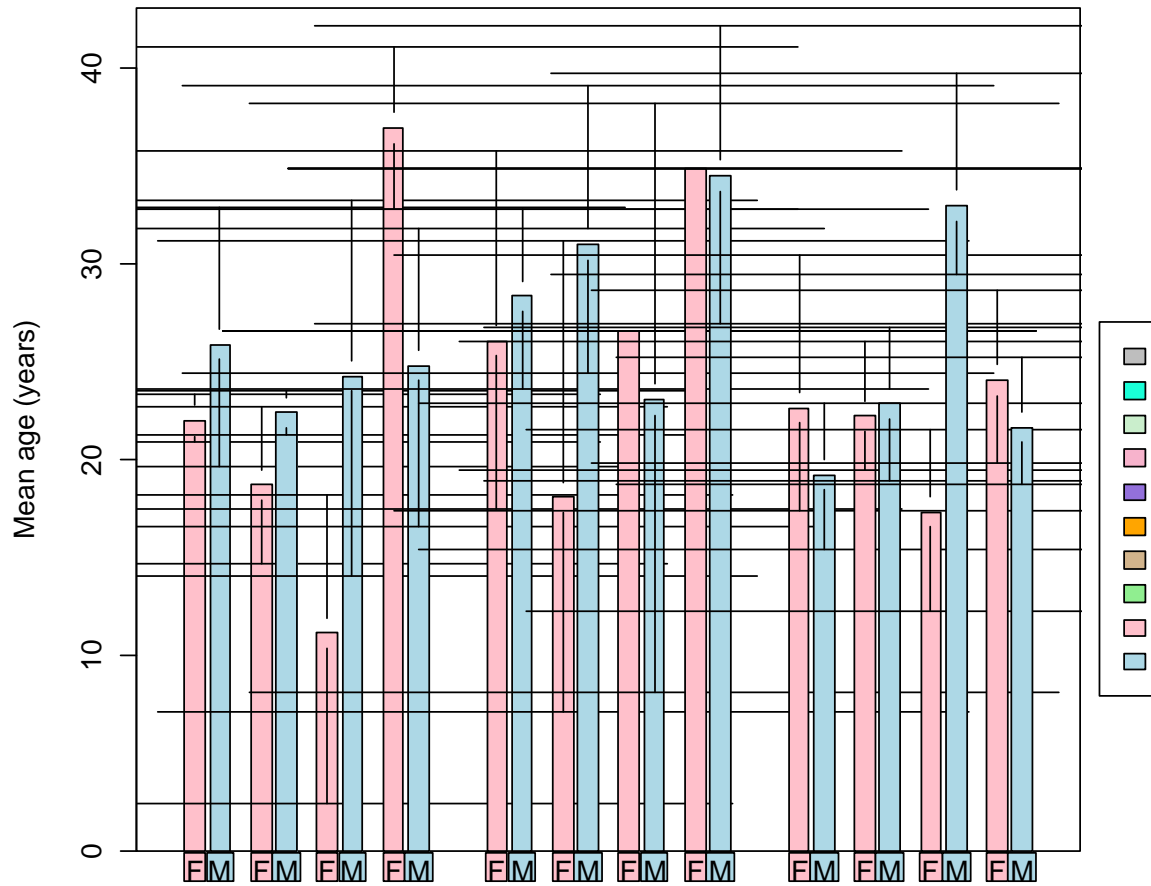


```

# example of a legend that might be included, needs a lot of space
par(xpd=TRUE)
legend(1.02,27,c("Overall", "Full time", "Part time", "No work", "Divorced",
"Married", "Single", "Widowed", "Female", "Male"),
fill=unlist(test.col))
par(xpd=FALSE,mar=c(5,4,4,2))

```

Use median and quantiles for dispersion

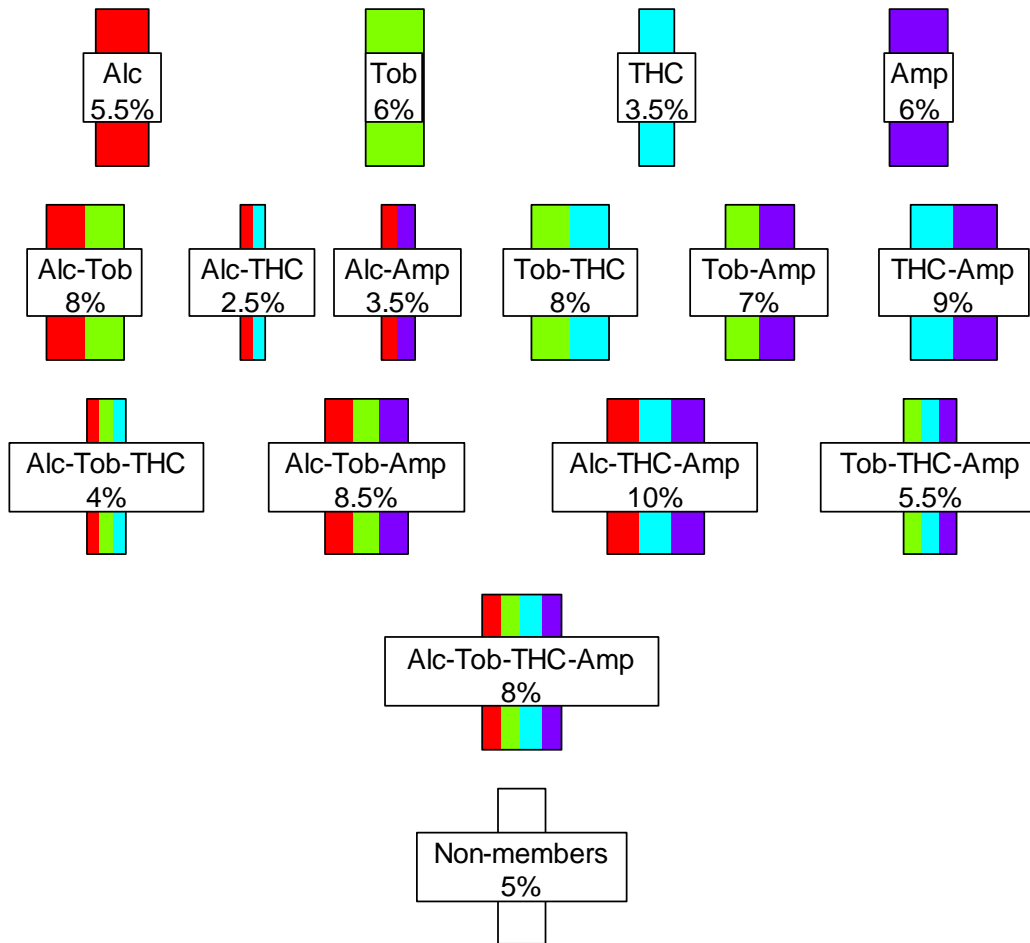


```

# create a matrix where each row represents an element and
# a 1 (or TRUE) in each column indicates that the element is a member
# of that set.
druguse<-matrix(c(sample(c(0,1),200,TRUE),
sample(c(0,1),200,TRUE),
sample(c(0,1),200,TRUE),
sample(c(0,1),200,TRUE))),ncol=4)
colnames(druguse)<-c("Alc","Tob","THC","Amp")
druglist<-makeIntersectList(druguse)
# first display it as counts
intersectDiagram(druglist)
# then as percent with non.members, passing the initial matrix
intersectDiagram(druguse,pct=TRUE,show.nulls=TRUE)

```

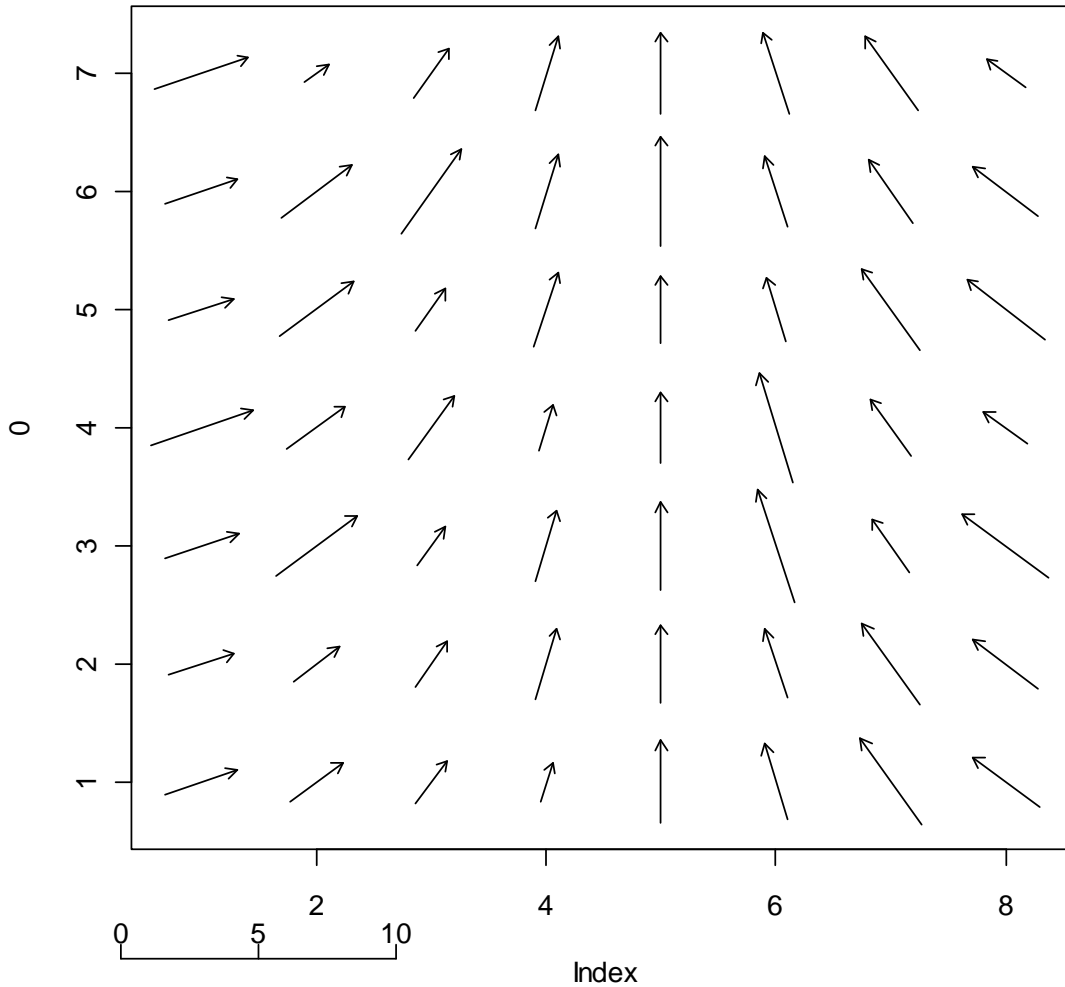
Intersection Diagram



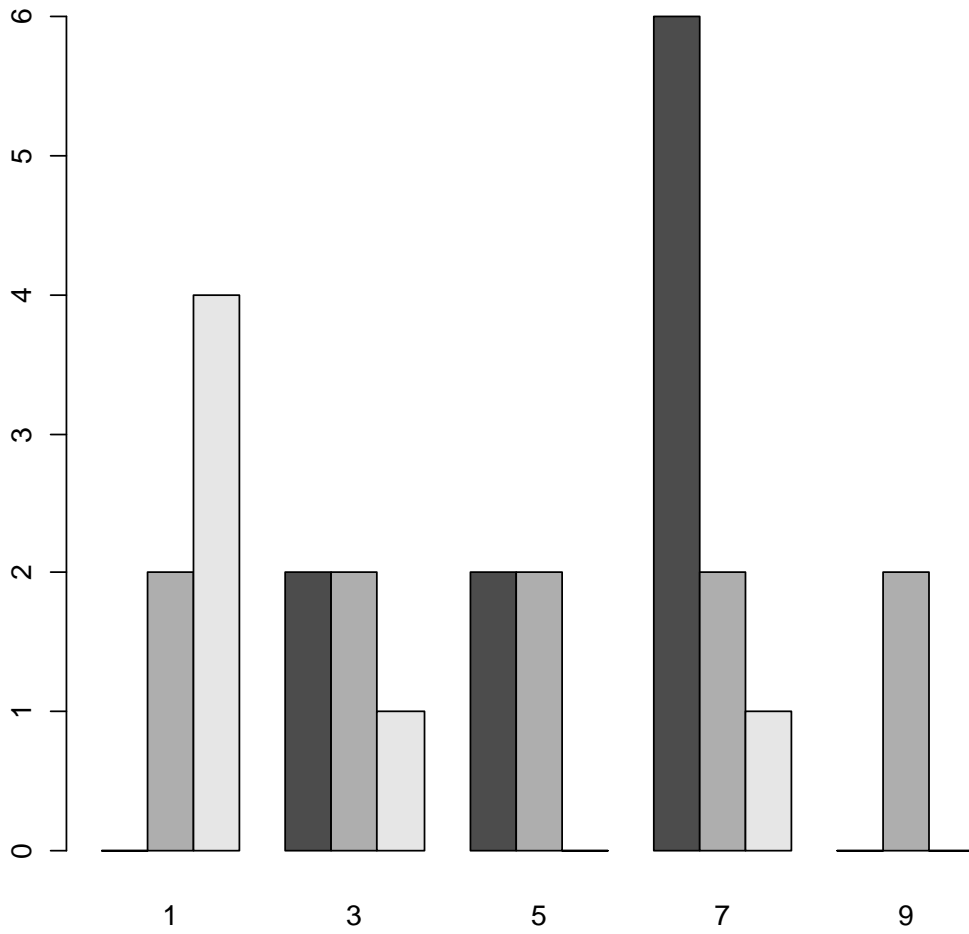
```

# manufacture a matrix of orientations in radians
o<-matrix(rep(pi*seq(0.1,0.8,by=0.1),7),ncol=8,byrow=TRUE)
m<-matrix(rnorm(56)+4,ncol=8,byrow=TRUE)
# get an empty plot of approximately 1:1 aspect ratio
plot(0,xlim=c(0.7,8.3),ylim=c(0.7,7.3),type="n")
vectorField(o,m,vecspec="rad")
# the scaling usually has to be worked out by trial and error
lengthKey(0.3,-0.5,c(0.5,10),0.24)

```



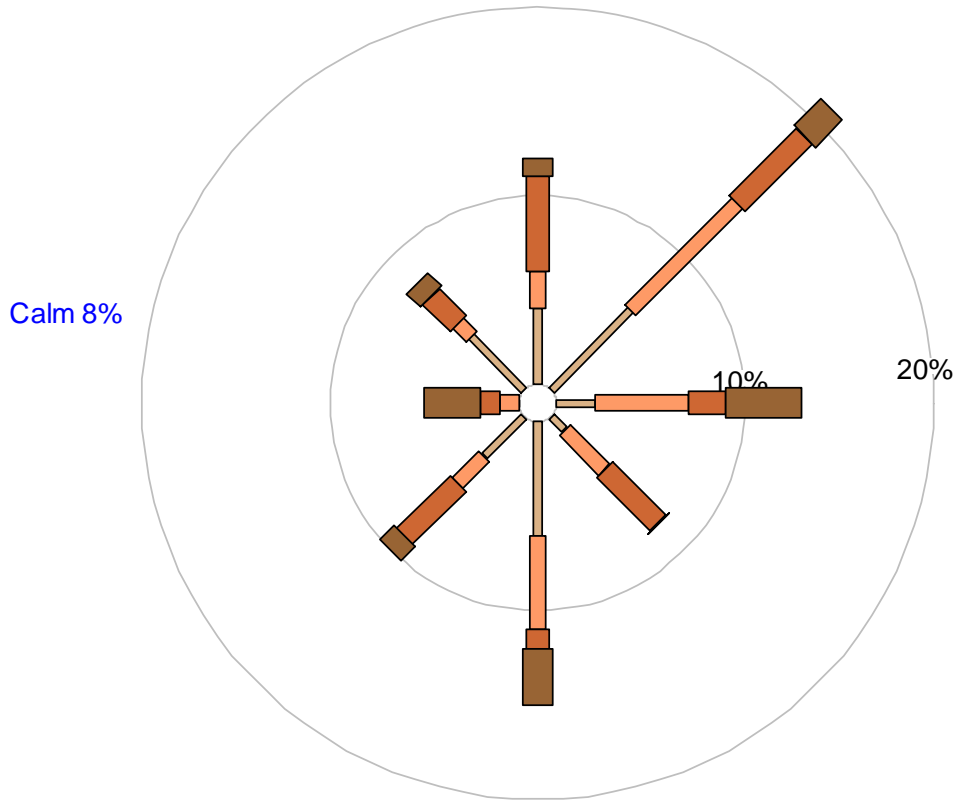
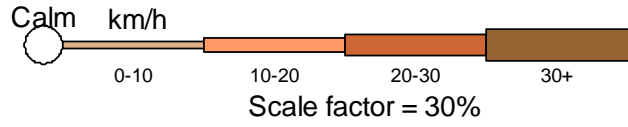
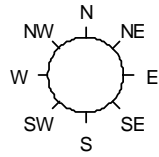
```
l <- list(runif(10)*10,1:10,c(1,1,1,1,4,8))  
multhist(l)
```



```

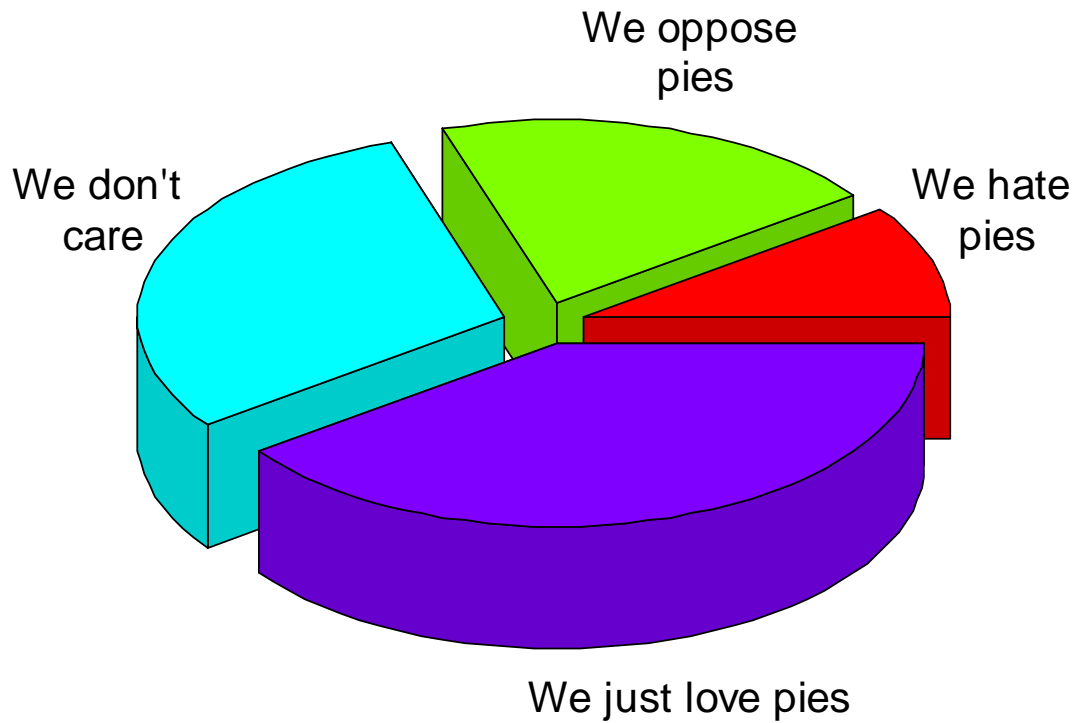
windagg<-matrix(c(8,0,0,0,0,0,0,0,4,6,2,1,6,3,0,4,2,8,5,3,5,2,1,1,
5,5,2,4,1,4,1,2,1,2,4,0,3,1,3,1),nrow=5,byrow=TRUE)
oz.windrose(windagg)

```



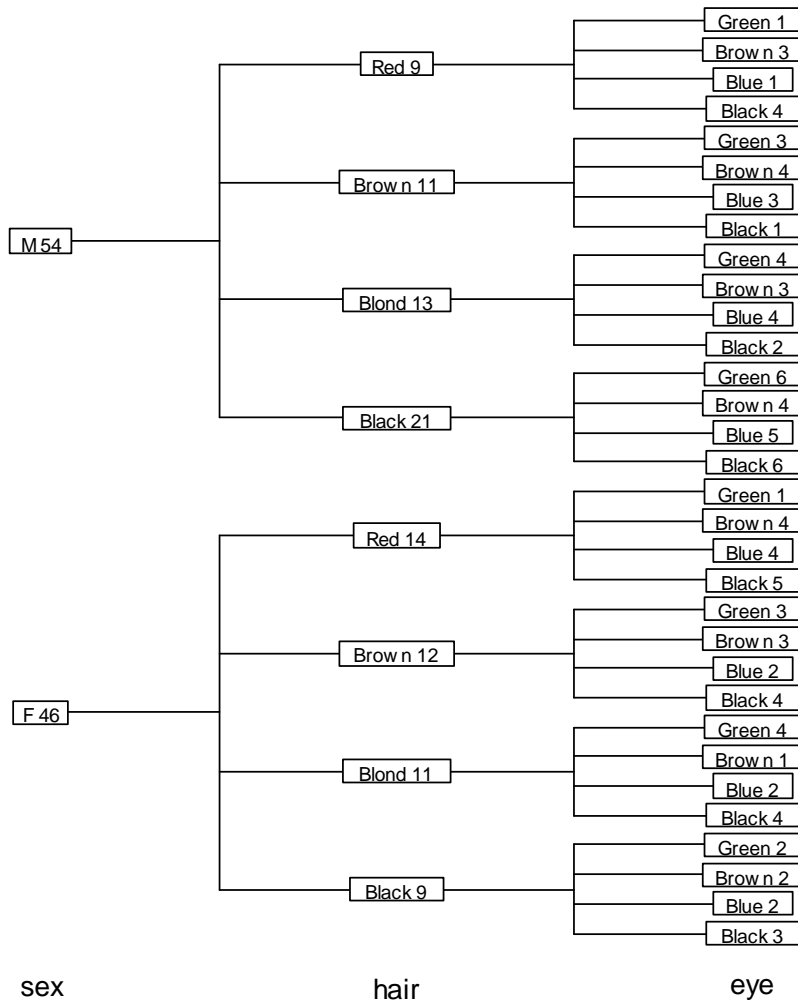
```
pieval<-c(2,4,6,8)
pielabels<-
c("We hate\n pies", "We oppose\n pies", "We don't\n care", "We just love pies")
pie3D(pieval,radius=0.9,labels=pielabels,explode=0.1,main="3D PIE OPINIONS")
```

3D PIE OPINIONS

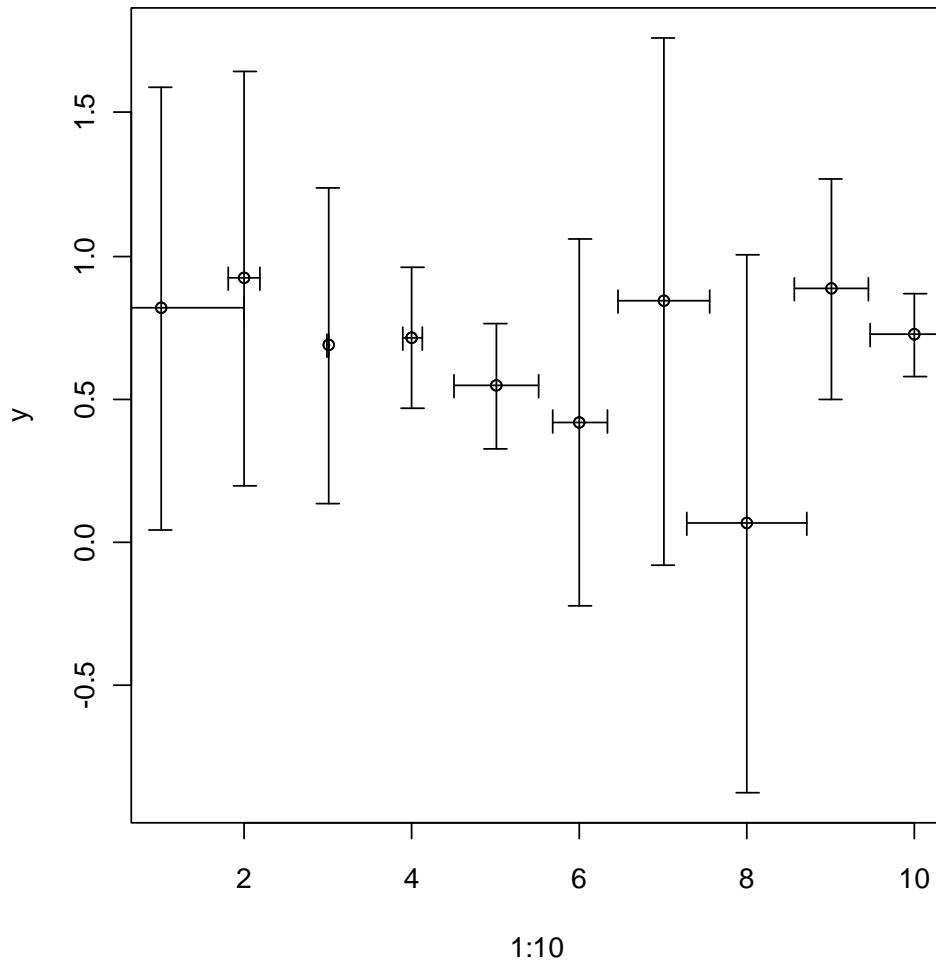


```
sex<-sample(c("M","F"),100,TRUE)
hair<-sample(c("Blond","Black","Brown","Red"),100,TRUE)
eye<-sample(c("Blue","Black","Brown","Green"),100,TRUE)
charac<-data.frame(sex=sex,hair=hair,eye=eye)
charalist<-makeDendrite(charac)
plot.dendrite(charalist,names(charac),main="Test dendrogram",cex=0.8)
```

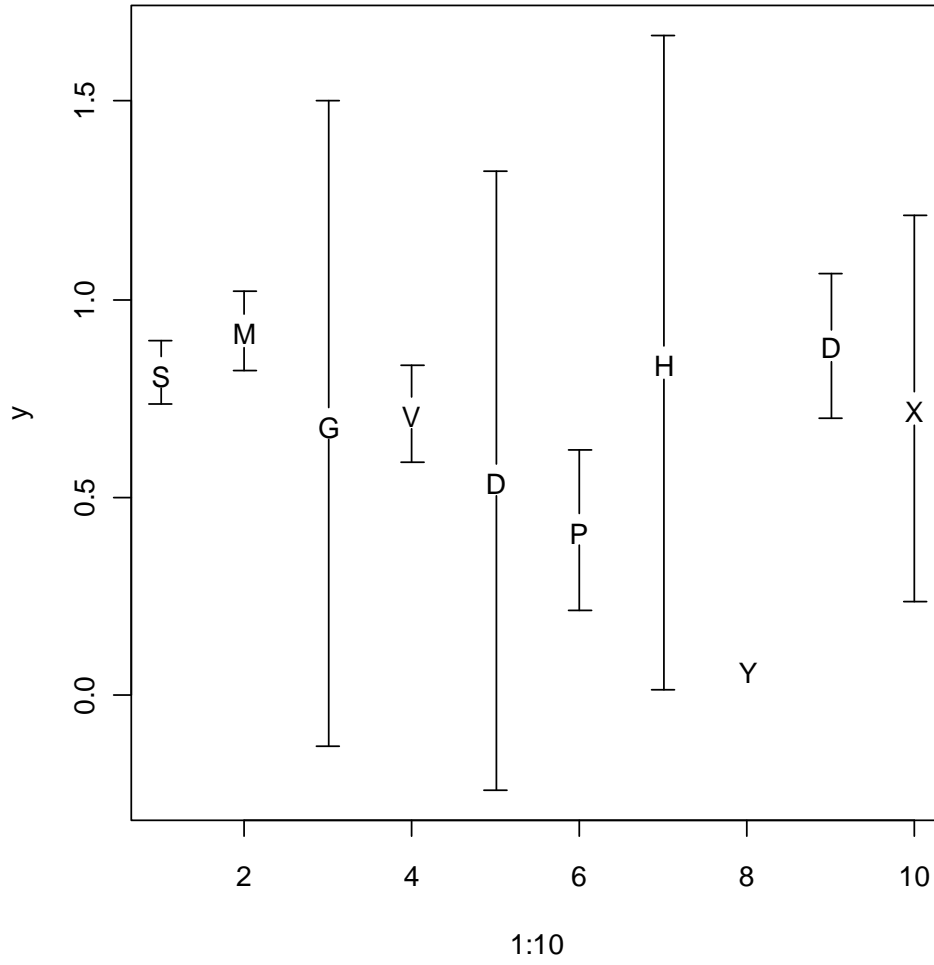
Test dendrogram



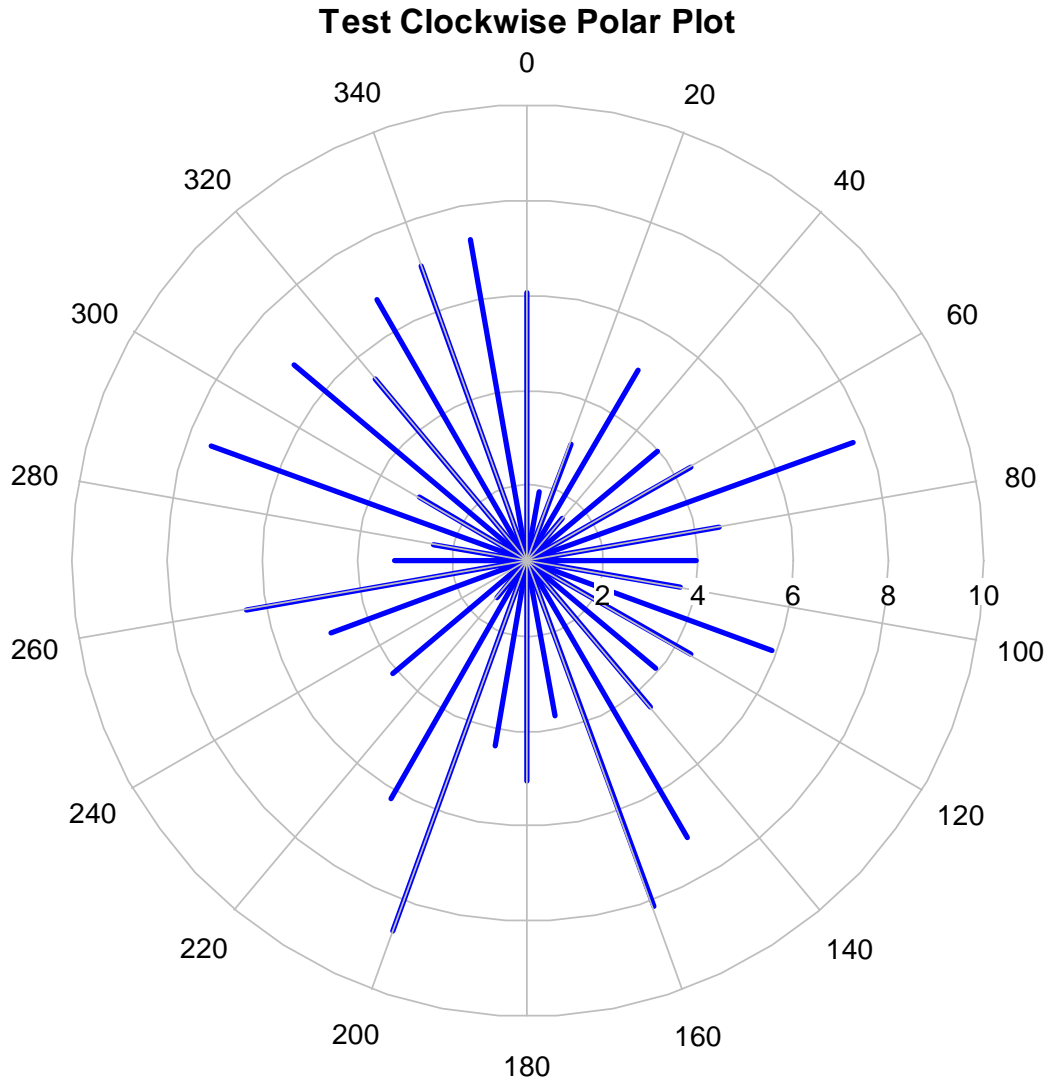
```
y<-runif(10)
err<-runif(10)
plotCI(1:10,y,err)
plotCI(1:10,y,err,2*err,lwd=2,col="red",scol="blue")
err.x<-runif(10)
err.y<-runif(10)
plotCI(1:10,y,err.y,pt.bg=par("bg"),pch=21)
plotCI(1:10,y,err.x,pt.bg=par("bg"),pch=21,err="x",add=TRUE)
data(warpbreaks)
attach(warpbreaks)
wmeans<-by(breaks,tension,mean)
wsd<-by(breaks,tension,sd)
```



```
labs<-sample(LETTERS,replace=TRUE,size=10)
plotCI(1:10,y,err,pch=NA,gap=0.02)
text(1:10,y,labs)
```



```
testlen<-c(rnorm(36)*2+5)
testpos<-seq(0,350,by=10)
polar.plot(testlen,testpos,main="Test Polar Plot",lwd=3,line.col=4)
polar.plot(testlen,testpos,main="Test Clockwise Polar Plot",
start=90,clockwise=TRUE,lwd=3,line.col=4)
# reset the margins
par(mar=c(5,4,4,2))
```

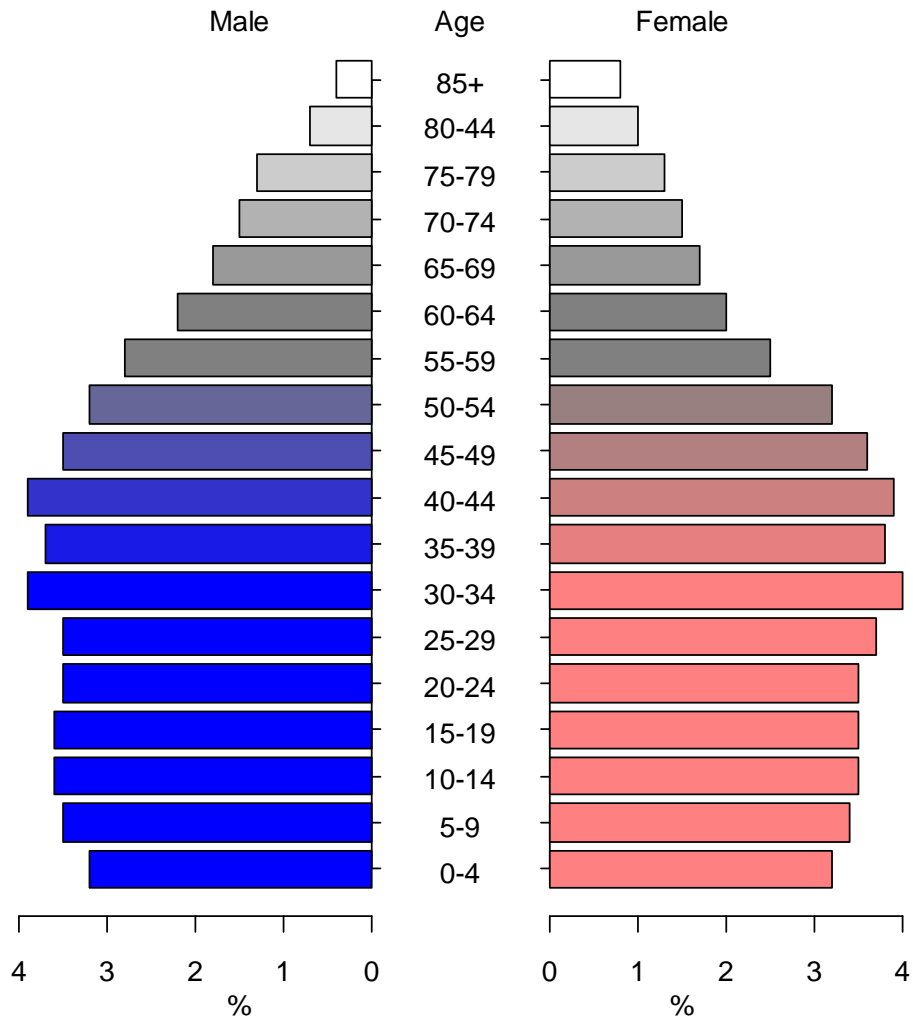


```

xy.pop<-c(3.2,3.5,3.6,3.6,3.5,3.5,3.9,3.7,3.9,3.5,3.2,2.8,2.2,1.8,
1.5,1.3,0.7,0.4)
xx.pop<-c(3.2,3.4,3.5,3.5,3.5,3.7,4,3.8,3.9,3.6,3.2,2.5,2,1.7,1.5,
1.3,1,0.8)
agelabels<-c("0-4", "5-9", "10-14", "15-19", "20-24", "25-29", "30-34",
"35-39", "40-44", "45-49", "50-54", "55-59", "60-64", "65-69", "70-74",
"75-79", "80-44", "85+")
xycol<-color.gradient(c(0,0,0.5,1),c(0,0,0.5,1),c(1,1,0.5,1),18)
xxcol<-color.gradient(c(1,1,0.5,1),c(0.5,0.5,0.5,1),c(0.5,0.5,0.5,1),18)
par(mar=pyramid.plot(xy.pop,xx.pop,labels=agelabels,
main="Australian population pyramid 2002",xycol=xycol,xxcol=xxcol))

```

Australian population pyramid 2002

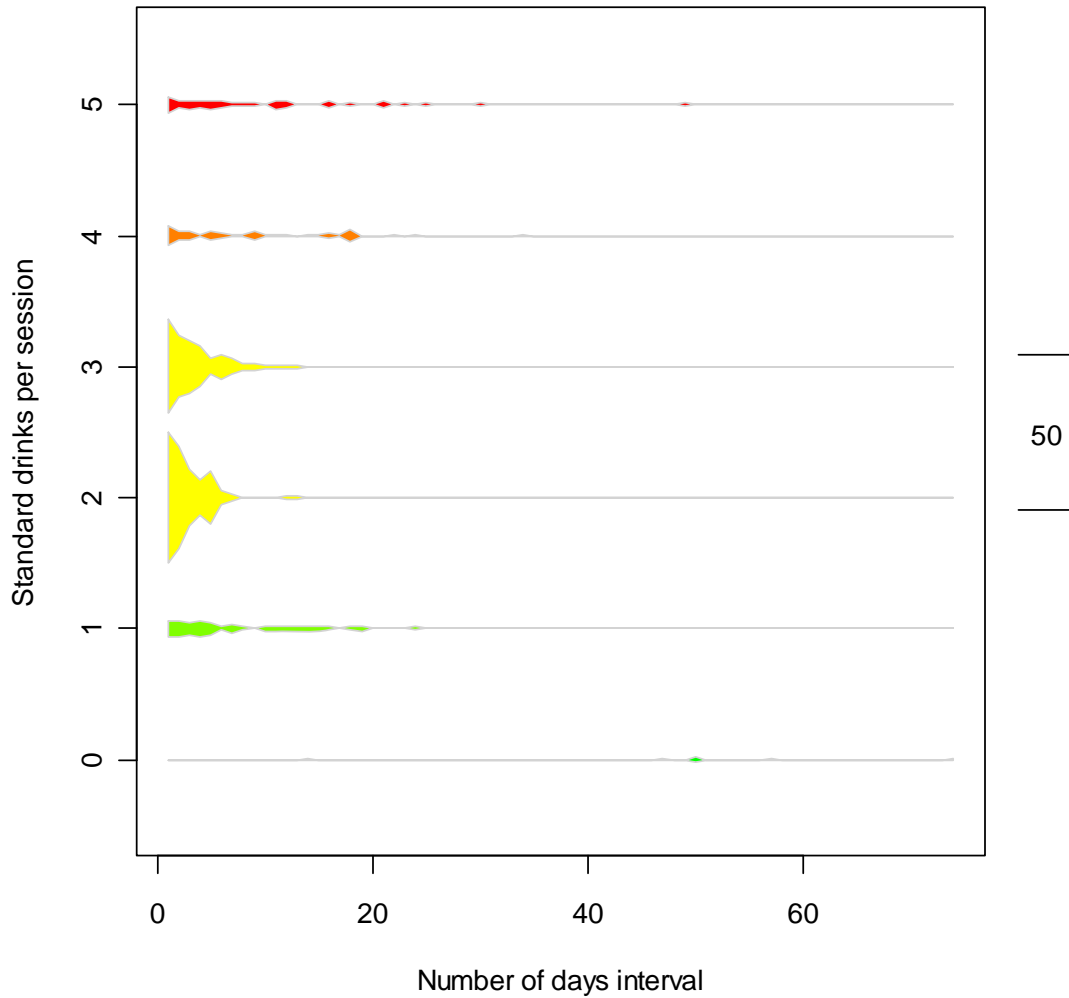


```

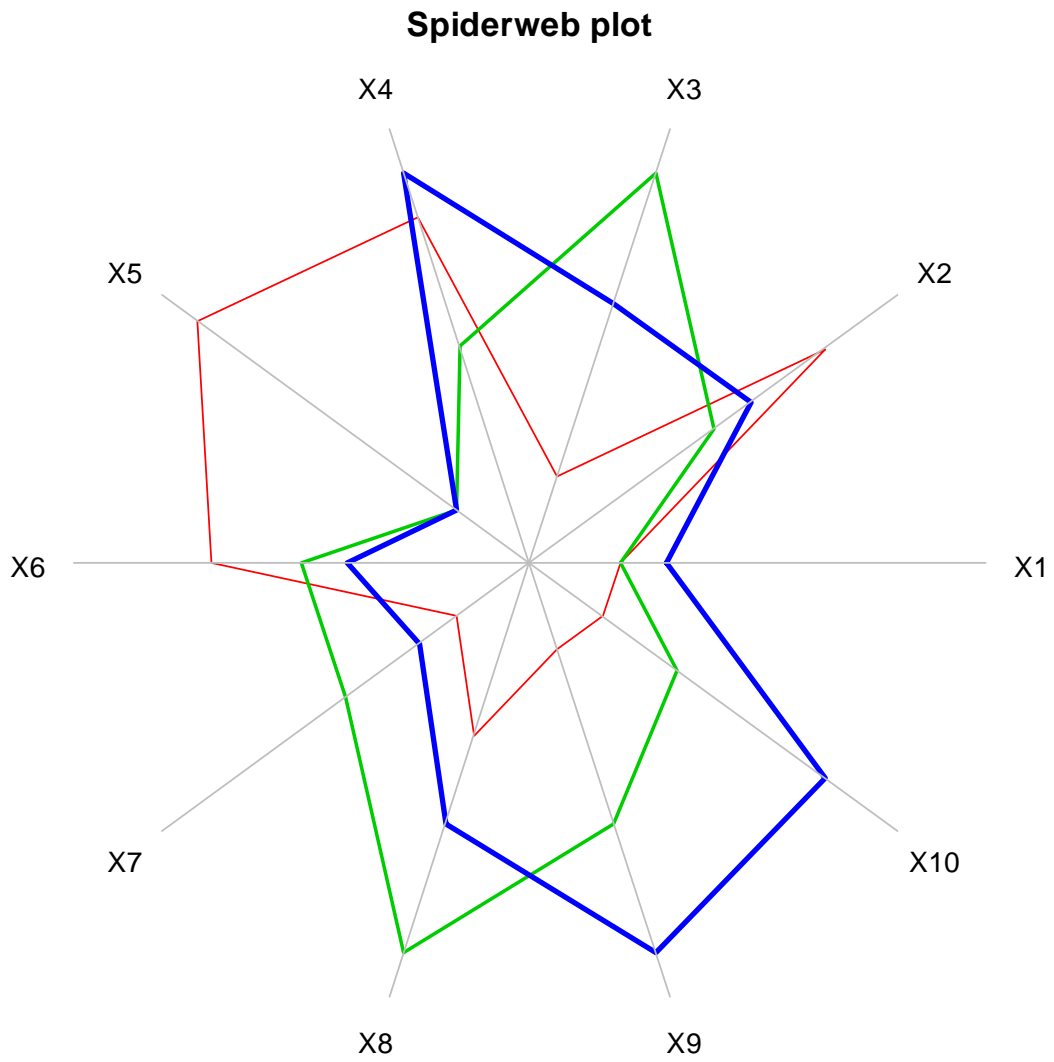
# first a moderate drinker with frequent bigger sessions
qnt<-sample(0:5,365,TRUE,prob=c(0.02,0.1,0.4,0.3,0.1,0.08))
qtdates<-seq(as.Date("2007-01-01"),as.Date("2007-12-31"),by=1)
qt.plot(qnt,as.numeric(qtdates),xlab="Number of days interval",
ylab="Standard drinks per session")
# now add monthly bigger sessions and notice how this
qnt[c(30,60,90,120,150,180,210,240,270,300,330,360)]<-rep(4:5,length.out=12)
qt.plot(qnt,as.numeric(qtdates),xlab="Number of days interval",
ylab="Standard drinks per session")

```

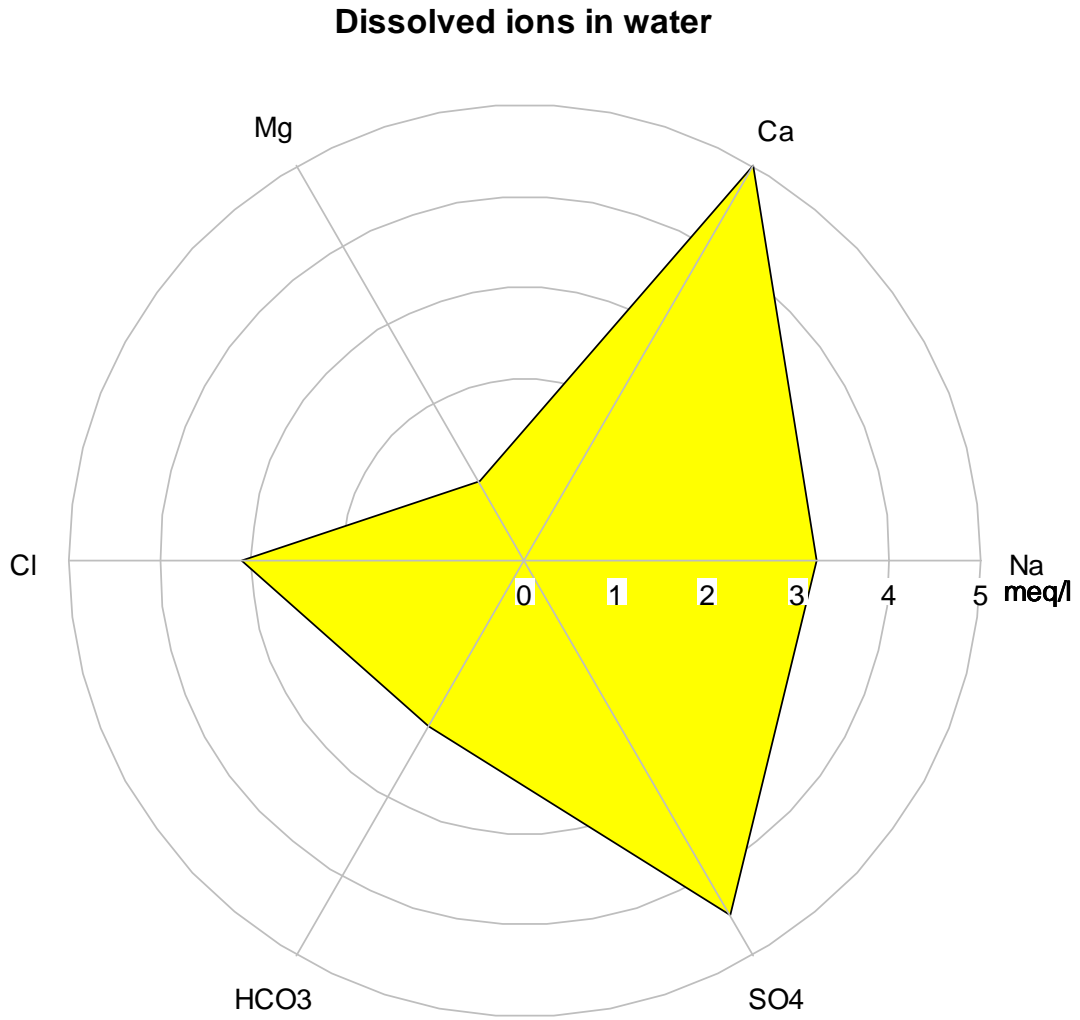
Quantity x interval



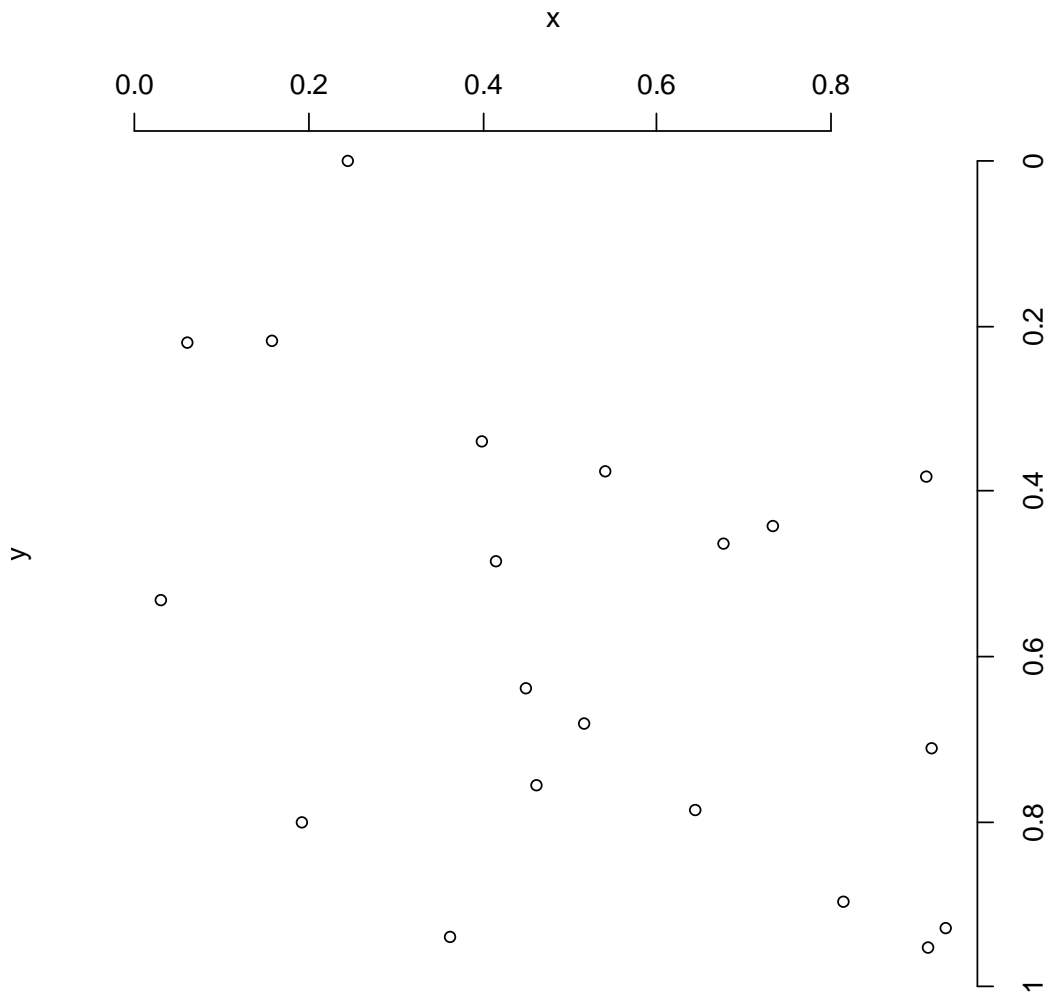
```
posmat<-matrix(sample(2:9,30,TRUE),nrow=3)
radial.plot(posmat,labels=paste("X",1:10,sep=""),rp.type="p",
main="Spiderweb plot",line.col=2:4,show.grid=FALSE,lwd=1:3,
radial.lim=c(0,10))
```



```
ions<-c(3.2,5,1,3.1,2.1,4.5)
ion.names<-c("Na","Ca","Mg","Cl","HCO3","SO4")
radial.plot(ions,labels=ion.names,rp.type="p",main="Dissolved ions in water",
grid.unit="meq/l",radial.lim=c(0,5),poly.col="yellow")
par(xpd=oldpar$xpd,mar=oldpar$mar,pty=oldpar$pty)
# reset the margins
par(mar=c(5,4,4,2))
```



```
x <- runif(20)
y <- runif(20)
revaxis(x,y,yside=4)
```

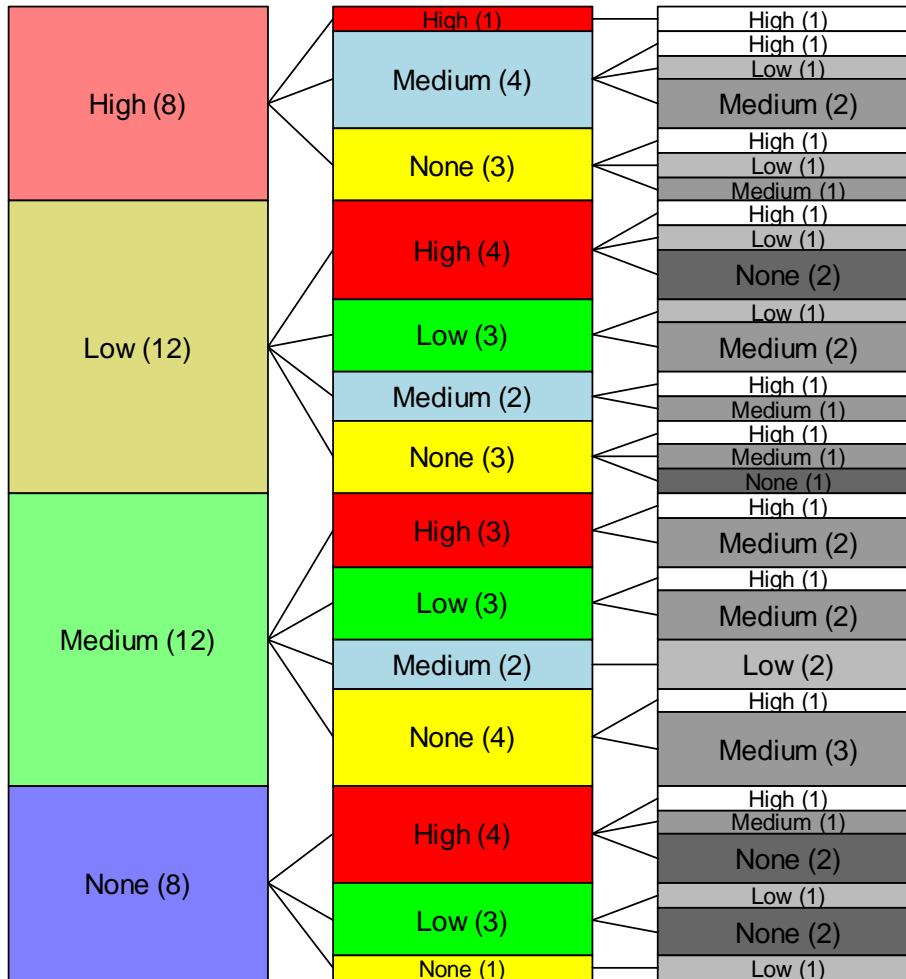


```

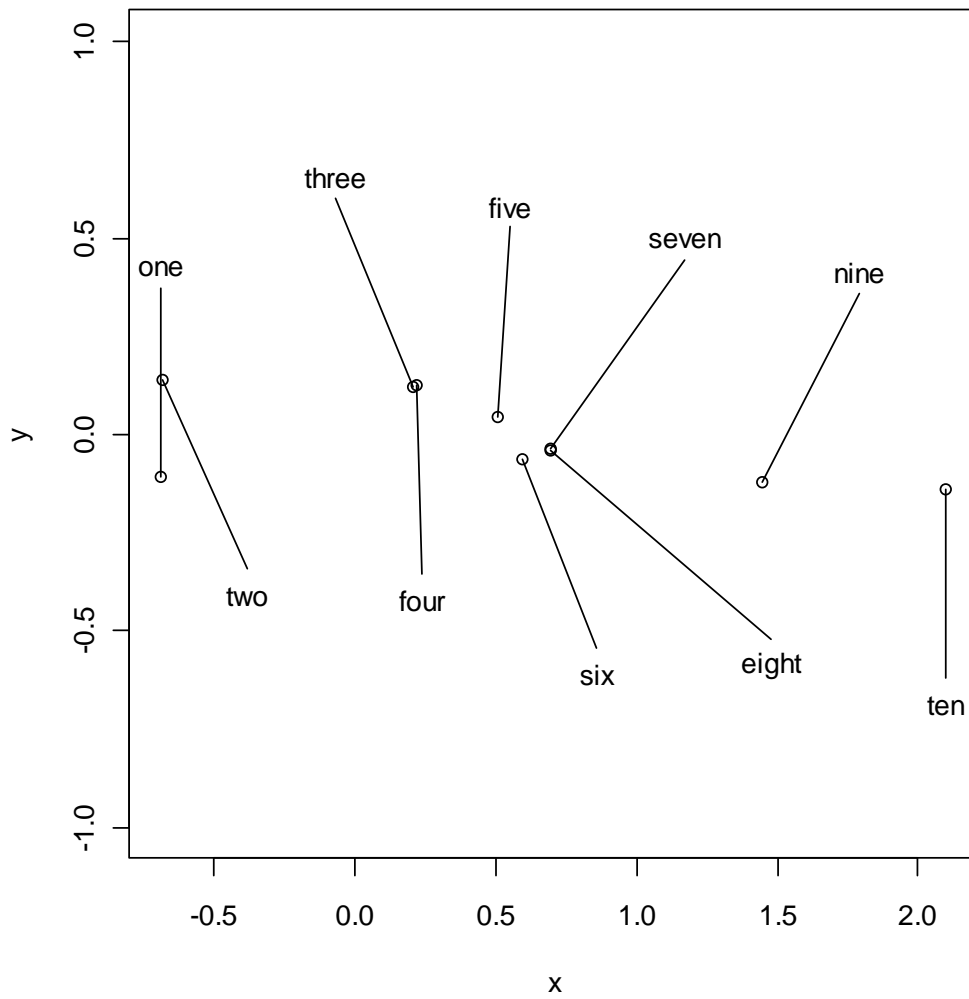
cat1<-sample(c("None","Low","Medium","High"),40,TRUE)
cat2<-sample(c("None","Low","Medium","High"),40,TRUE)
cat3<-sample(c("None","Low","Medium","High"),40,TRUE)
hcats<-data.frame(cat1,cat2,cat3)
bhcol<-list(c("#ff8080","#dddd80","#80ff80","#8080ff"),
c("red","green","lightblue","yellow"),
c("#ffffff","#bbbbbb","#999999","#666666"))
sizetree(hcats,col=bhcol,main="Hierarchical count chart")

```

Hierarchical count chart



```
# spread labels out in the x dimension using defaults
x<-sort(rnorm(10))
y<-rnorm(10)/10
plot(x,y,ylim=c(-1,1),type="p")
nums<-c("one","two","three","four","five","six","seven","eight","nine","ten")
spread.labels(x,y,nums)
```

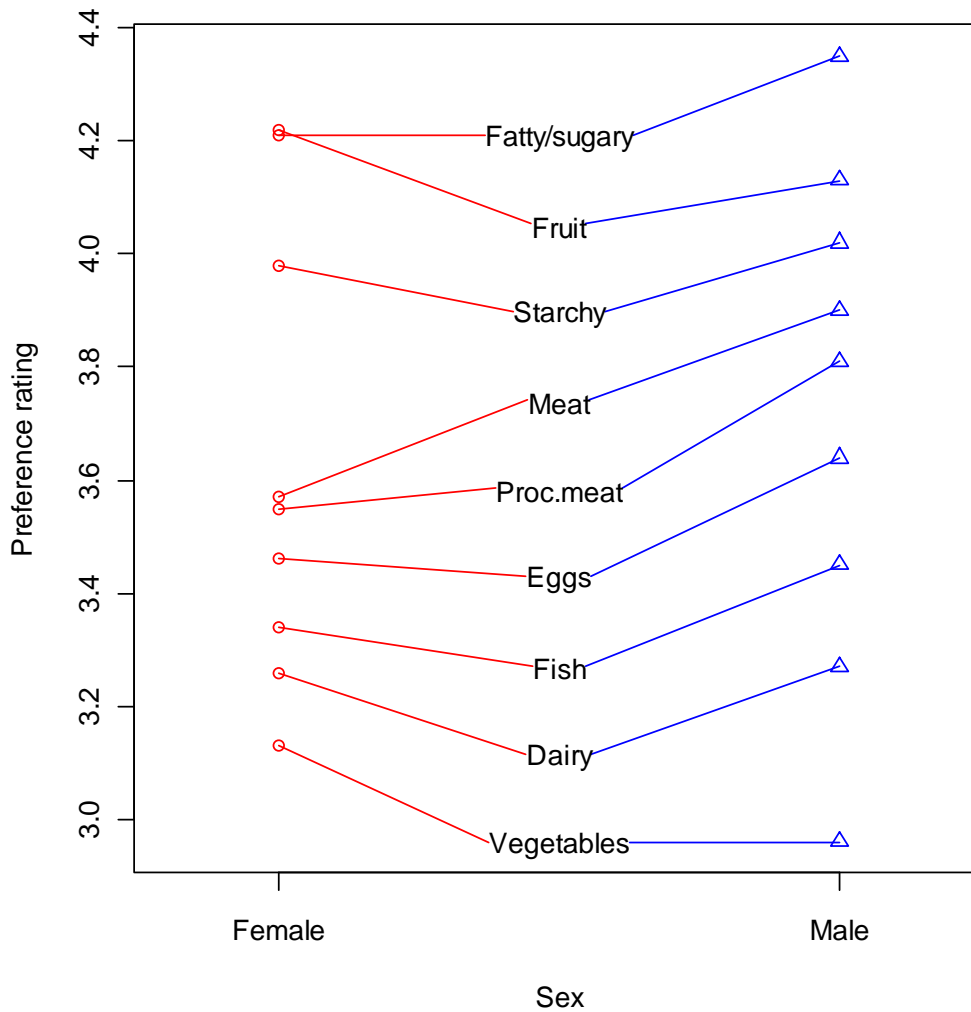


```

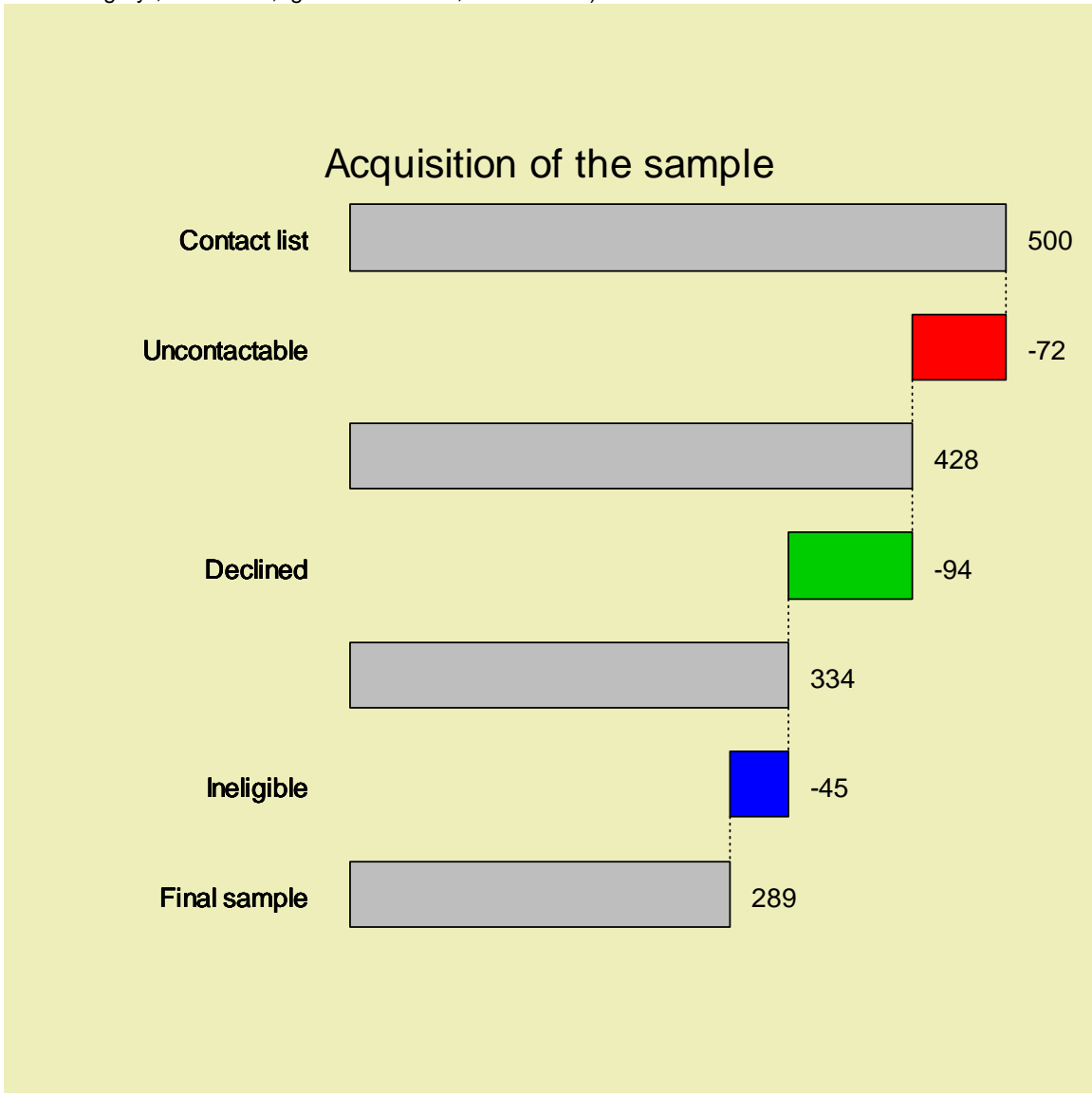
fpkids<-data.frame(Food=c("Fatty/sugary","Fruit","Starchy","Meat",
"Proc.meat","Eggs","Fish","Dairy","Vegetables"),
Female=c(4.21,4.22,3.98,3.57,3.55,3.46,3.34,3.26,3.13),
Male=c(4.35,4.13,4.02,3.9,3.81,3.64,3.45,3.27,2.96))
plot(rep(1,9),fpkids$Female,xlim=c(0.8,2.2),
ylim=range(c(fpkids$Female,fpkids$Male)),xlab="Sex",xaxt="n",
ylab="Preference rating",main="Children's food preferences by sex",
col="red")
axis(1,at=1:2,labels=c("Female","Male"))
points(rep(2,9),fpkids$Male,col="blue",pch=2)
spread.labels(rep(1:2,each=9),c(fpkids$Female,fpkids$Male),
fpkids$Food,between=TRUE,lincol=c("red","blue"))

```

Children's food preferences by sex

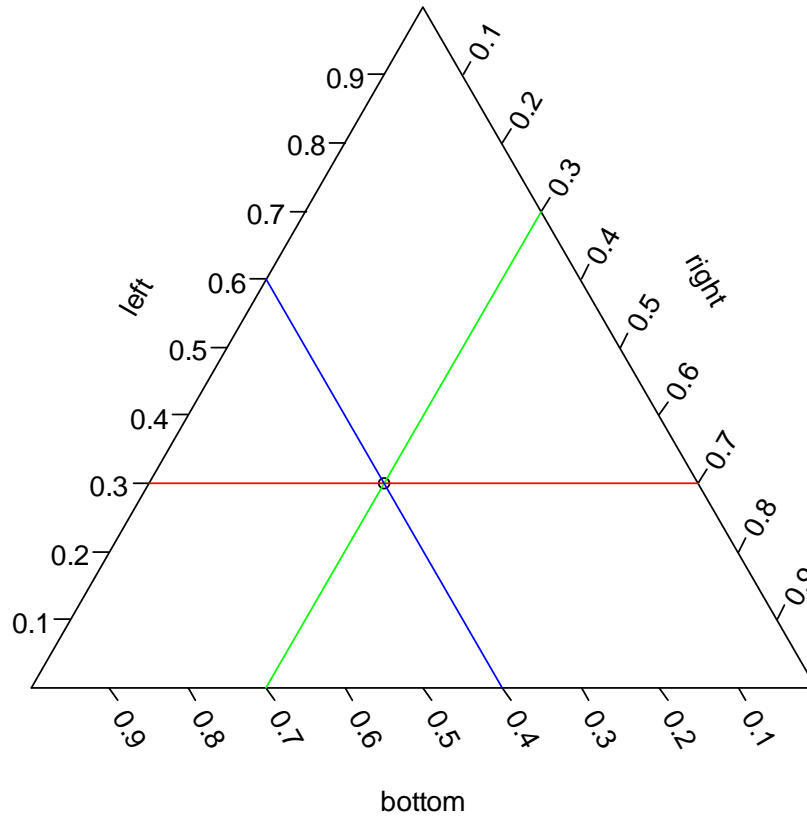


```
sample_size<-c(500,-72,428,-94,334,-45,289)
totals<-c(TRUE,FALSE,TRUE,FALSE,TRUE,FALSE,TRUE)
labels<-c("Contact list","Uncontactable","","Declined","","Ineligible",
"Final sample")
staircase.plot(sample_size,totals,labels,main="Acquisition of the sample",
total.col="gray",inc.col=2:4,bg.col="#eeeebb",direction="s")
```



```
triax.return<-triax.plot(data.frame(bottom=0.4,right=0.3,left=0.3),
main="Triax ablines",no.add=FALSE)
triax.abline(l=0.3,col="red")
triax.abline(r=0.3,col="green")
triax.abline(b=0.4,col="blue")
par(triax.return$oldpar)
```

Triax ablines

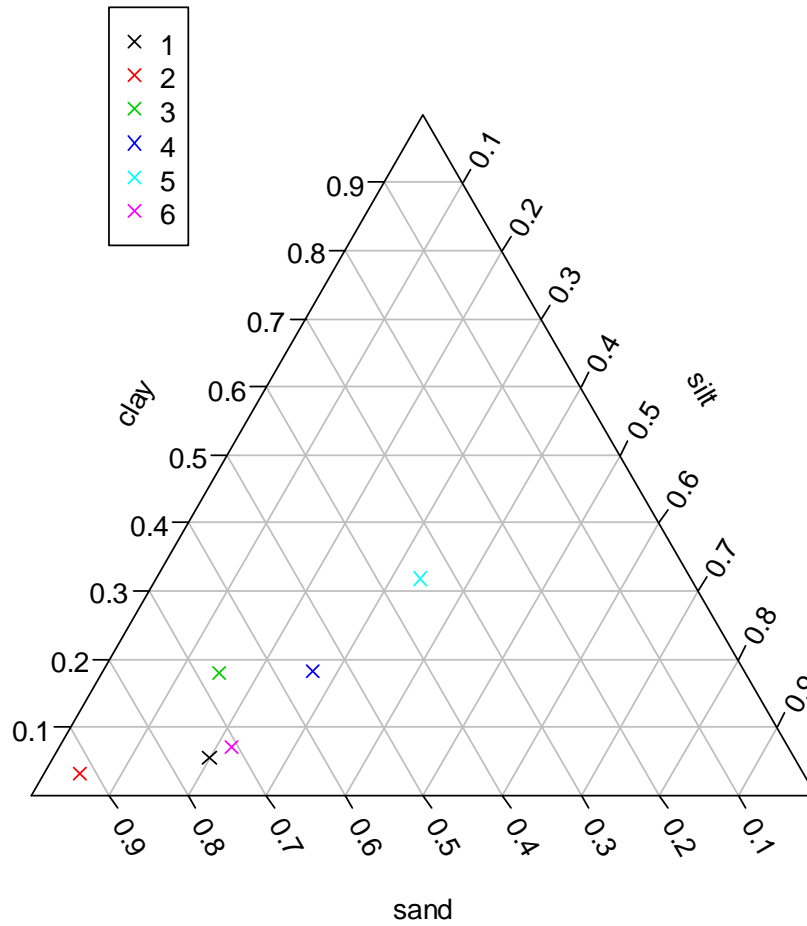


```

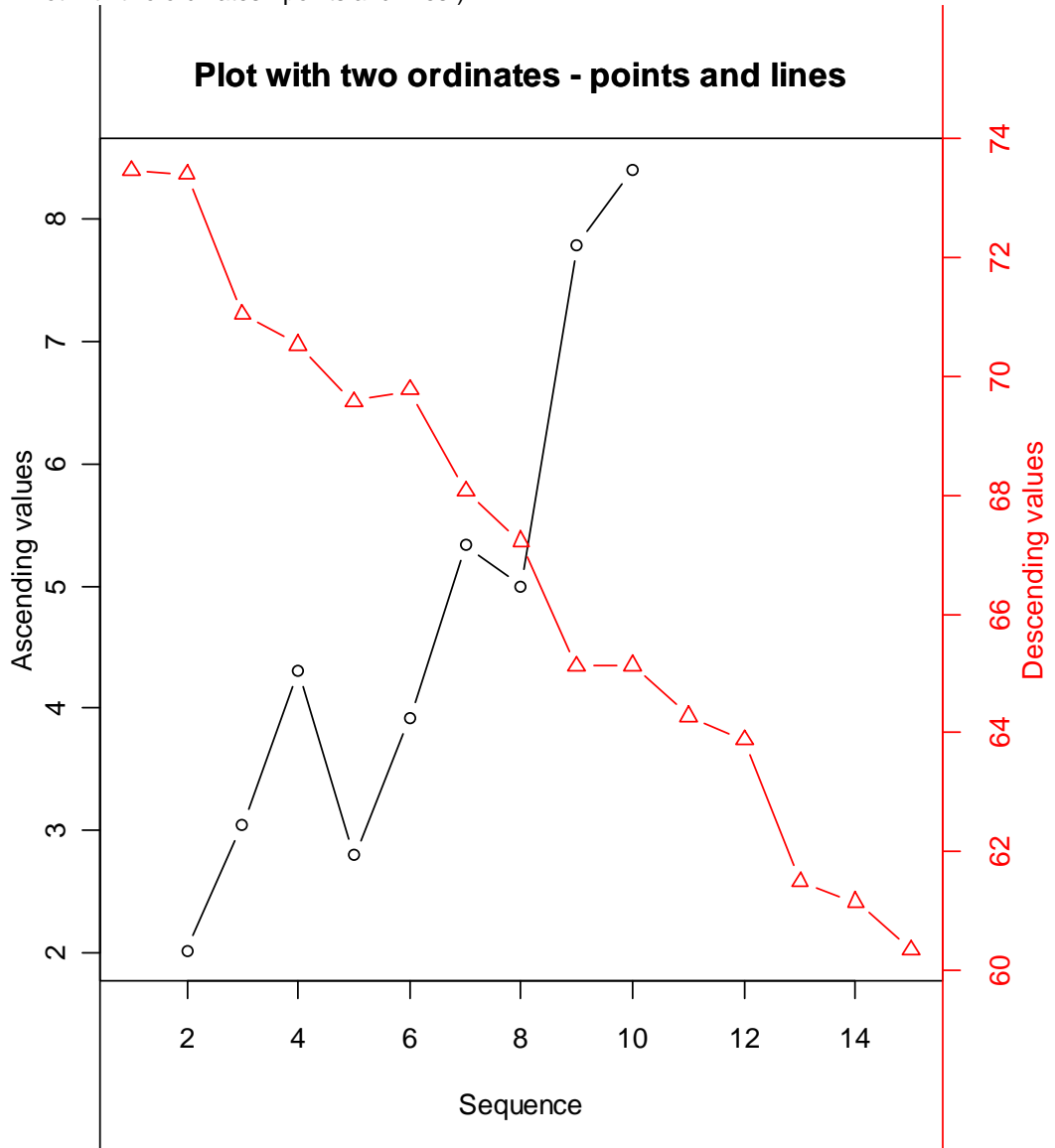
data(soils)
triax.plot(soils[1:10,],main="DEFAULT")
triax.plot(soils[1:10,],main="PERCENTAGES (Counterclockwise axes)",
tick.labels=list(l=seq(10,90,by=10),r=seq(10,90,by=10),b=seq(10,90,by=10)),
pch=3,cc.axes=TRUE)
triax.return<-triax.plot(soils[1:6,],main="GRID AND LEGEND",
show.grid=TRUE,show.legend=TRUE,col.symbols=1:6,pch=4)
# triax.plot changes a few parameters
par(triax.return$oldpar)

```

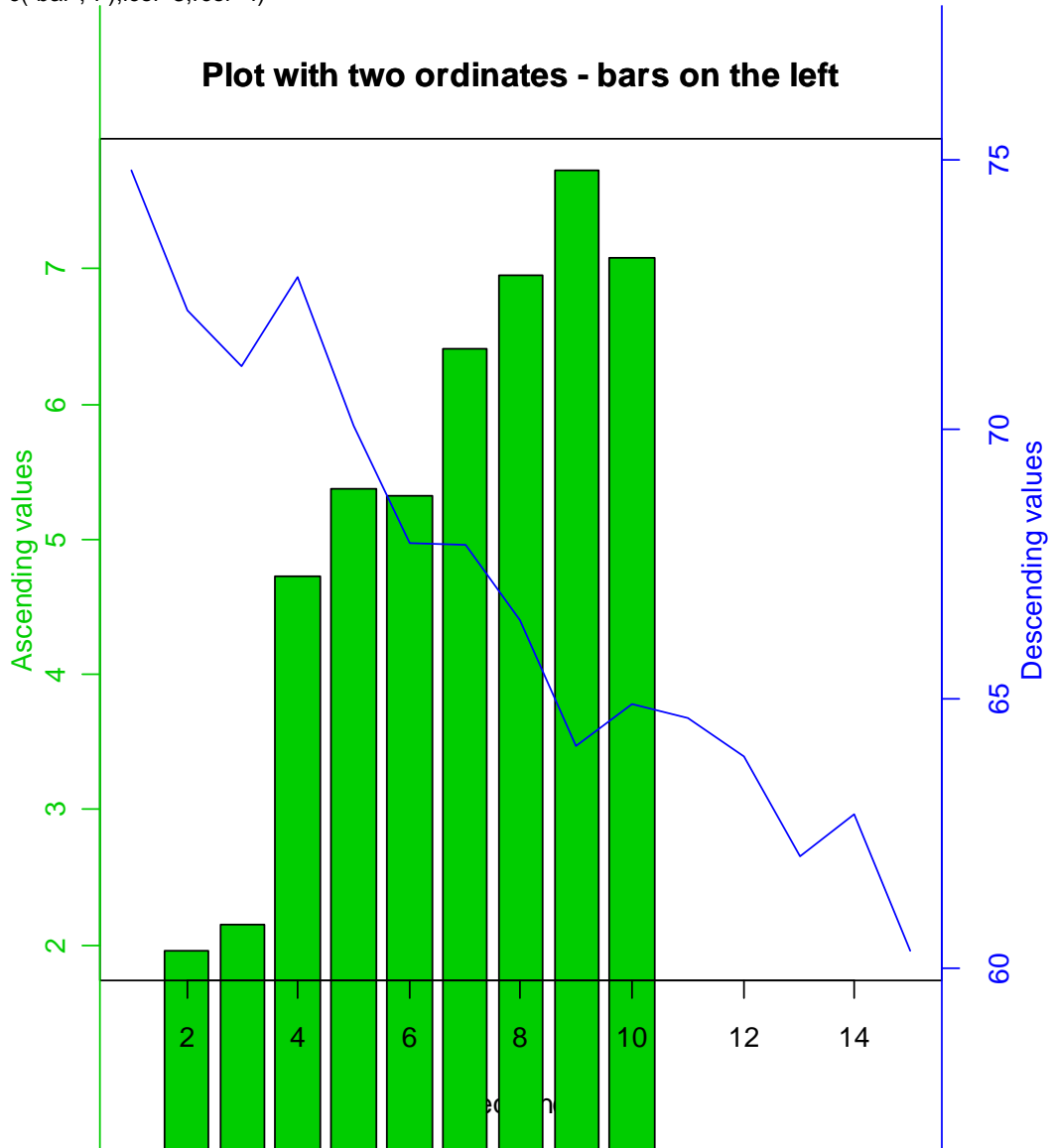
GRID AND LEGEND



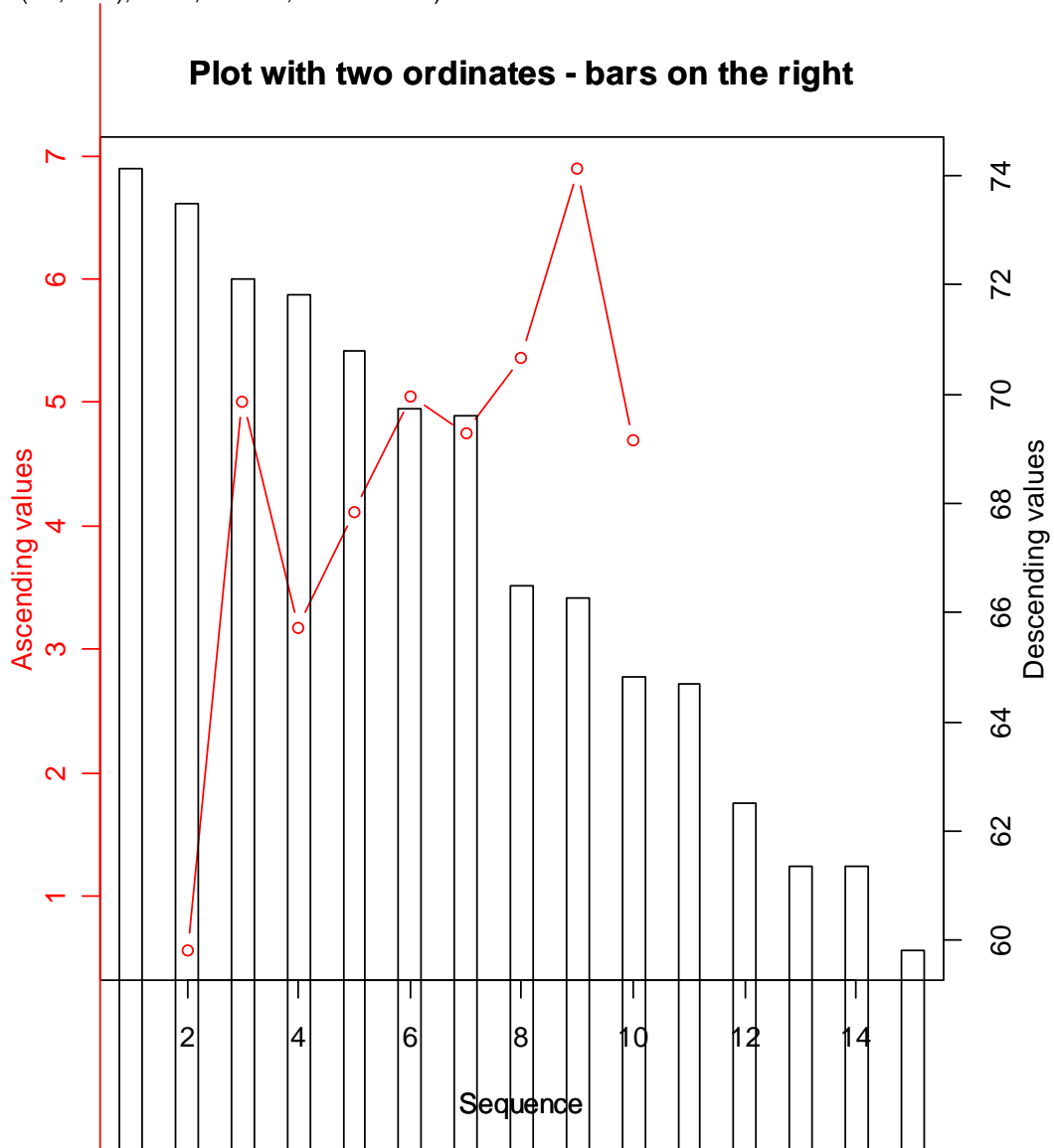
```
twoord.plot(2:10,seq(3,7,by=0.5)+rnorm(9),
1:15,rev(60:74)+rnorm(15),xlab="Sequence",
ylab="Ascending values",rylab="Descending values",
main="Plot with two ordinates - points and lines")
```



```
twoord.plot(2:10,seq(3,7,by=0.5)+rnorm(9),
1:15,rev(60:74)+rnorm(15),xlab="Sequence",
ylab="Ascending values",rylab="Descending values",
main="Plot with two ordinates - bars on the left",
type=c("bar","l"),lcol=3,rcol=4)
```



```
twoord.plot(2:10,seq(3,7,by=0.5)+rnorm(9),
1:15,rev(60:74)+rnorm(15),xlab="Sequence",
ylab="Ascending values",rylab="Descending values",
main="Plot with two ordinates - bars on the right",
type=c("b","bar"),lcol=2,rcol=NA,halwidth=0.2)
```



```
xhist<-hist(rnorm(100),plot=FALSE)
xdens<-dnorm(seq(-2,2,by=0.05))
twoord.plot(xhist$mids,xhist$counts,seq(-2,2,by=0.05),
xdens,type=c("bar","l"),lcol=4,rcol=2,ylab="Counts",
rylab="Density",main="Histogram and density curve",
halfwidth=0.2)
```

