LECTURE 2
PLOTTING DATA
Pairs plot

(p24)

```r
fuel <- read.table("fuel.txt",header=T)
pairs(fuel[,c("Weight","Disp","Mileage")])
```
XY plot showing a Factor:

(p26)

```r
fuel <- read.table("fuel.txt",header=T)

str(fuel)

attach(fuel,pos=3)

levels(Type)
chf <- as.numeric(Type)
plot(Weight,Mileage,pch=as.numeric(Type))
legend(3300,35,
legend=c("compact","large","medium","small","sporty","van"),
pch=1:6)

plot(Weight,Mileage,type="n");text(Weight,Mileage,Type)
legend(3300,35,
```
legend=c("compact","large","medium","small","sporty","van"),
pch=c("1","2","3","4","5","6"))

chq <- factor(Type)
levels(chq) <- c("c","l","m","s","p","v")
chch <- as.character(chq)
plot(Weight,Mileage,type="n");text(Weight,Mileage,chch)
legend(3300,35,
legend=c("compact","large","medium","small","sporty","van"),
pch=c("c","1","m","s","p","v"))
Coplot and Lattice plots

(p29)

```r
fuel <- read.table("fuel.txt", header=T)
str(fuel)
attach(fuel, pos=3)
levels(Type)
chf <- as.numeric(Type)
chq <- factor(Type)
levels(chq) <- c("c","l","m","s","p","v")
chch <- as.character(chq)

#

coplot(Mileage ~ Weight | Type, pch= chch)
```
library(lattice)
xyplot(Mileage~Weight | Type)

Page 29 Notes:

Note that the use of symbols with each panel shows the order of the panels by Type.
Page 31 Notes:

Note the use of R function `rev` to produce a closed polygon.
Solution to the Exercise on page 34.

```r
fuel <- read.table("fuel.txt", header=T)
str(fuel)
#___________________________________________________
attach(fuel, pos=3)
levels(Type)
chf <- as.numeric(Type)
chq <- factor(Type)
levels(chq) <- c("c", "l", "m", "s", "p", "v")
chch <- as.character(chq)

library(lattice)

listc <- c("Compact", "Large", "Medium", "Small", "Sporty", "Van")

xyplot(Mileage~Weight | Type, panel=panel.superpose, groups=Type,
       key=list(text=list(listc),
                points=list(pch=trellis.par.get("superpose.symbol")$pch[1:6]),
                columns=6))
```
The different characters within each box are almost irrelevant, due to the internal labelling of panels. Note the legend produced on top and the use of `columns=6` in the `key` option to string the key out in a single row, rather than the default of a single column. The graphic produced is on the web site under ’Examples’ : Lecture 2 ”p34.pdf”.

For further information use ”library(lattice);help(xyplot)” inside R.
options(digits=3);library(lattice)
heath <- read.table("heath.dat",header=T)

heath$Heath <- factor(heath$Heath,labels=c("Dry(h)","Wet(h)"))
heath$Seed <- factor(heath$Seed,
  labels=c("Dry(s)","None(s)","Wet(s)"))
heath$Sample <- factor(rep(rep(1:3,each=3),24))
heath$Repl <- factor(rep(rep(1:2,each=54),2))

trellis.device(color=F)
dot.line <- trellis.par.get("dot.line")
dotp <- dotplot(Seed ~ AcaciaR/AcaciaN|Heath*Enclosure,data=heath,
   groups=Transect,
   panel = function(x, y, subscripts, ...) {
     panel.abline(h = y, col = dot.line$col,lty = dot.line$lty)
     panel.superpose(x, y, subscripts, ...)})

print(dotp)
Note the use of `color = F` in `device.trellis()` to produce symbols rather than colours in the graphic.

**DOTPLOT**

If you simply type `help(dotplot)` in R you will get a message saying that the function ’dotplot’ is being replaced by ’dotchart’, a simpler version.

The function ’dotplot’ as per page 35 still works however, since there are two libraries in R that use the term `dotplot`. The one on page 35 uses ’dotplot’ from the lattice library. The defunct ’dotplot’ is from the base library. Thus the analysis on page 35 holds.
Interacting with graphics

This will be done in class interactively!!

(page 37–38)