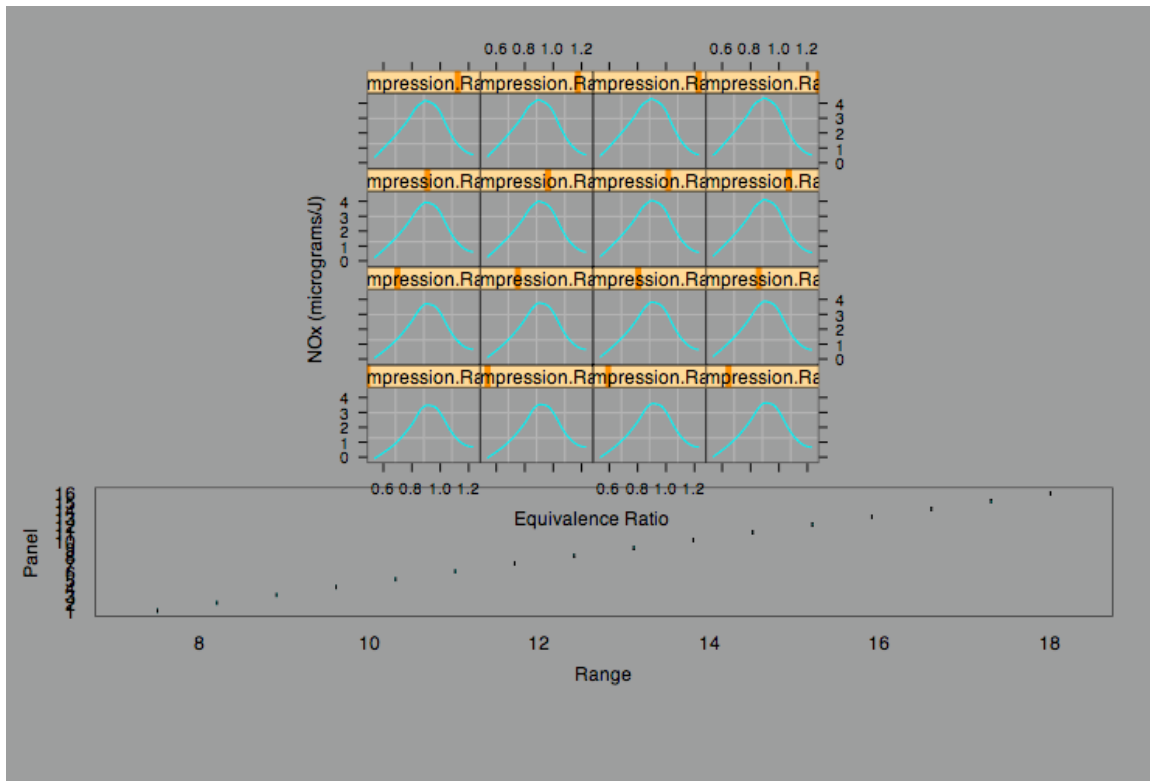


Ch 4: Level Plots of Data. The galaxy data.

Today: More Ch 4: Level Plots, Contour Plots, Wireframe Plots, with Galaxy Data Set

But First – a look at the program that produced the following coplot (ethanol data)



This is produced from

```
> book.4.13 <-
+ function()
+ {
+   attach(ethanol)
+   eth.m <- loess(NOx ~ C * E, span = 1/3, degree = 2,
+     parametric = "C", drop.square = "C", family="s")
+   C.marginal <- seq(min(C),max(C),length=16)
+   E.marginal <- seq(min(E),max(E),length=50)
+   CE.marginal <- list(C=C.marginal,E=E.marginal)
+   CE.grid <- expand.grid(CE.marginal)
+   eth.fit <- predict(eth.m,CE.grid)
+   Compression.Ratio <- CE.grid$C
+   ans <- xyplot(eth.fit ~ CE.grid$E | Compression.Ratio,
+     panel = function(x, y) {
```

```

+           panel.grid(h=2)
+           panel.xyplot(x, y, type = "l")
+       },
+       aspect="xy",
+       layout = c(4, 4),
+       xlab = "Equivalence Ratio",
+       ylab = "NOx (micrograms/J)")
+ detach()
+ print(ans, position=c(0,.3,1,1), more=T)
+
+ ans <- plot(shingle(seq(min(ethanol$C), max(ethanol$C), length=16)),
+           sub = list("Figure 4.13",cex=.8),
+           scales=list(x=list(cex=.7),y=list(cex=.7)),
+           xlab = "Compression Ratio")
+ print(ans, position=c(0,.1,1,.45))
+ invisible()
+ }

```

Note the command

```
eth.fit <- predict(eth.m,CE.grid)
```

where eth.fit was a loess fit based on the original data.

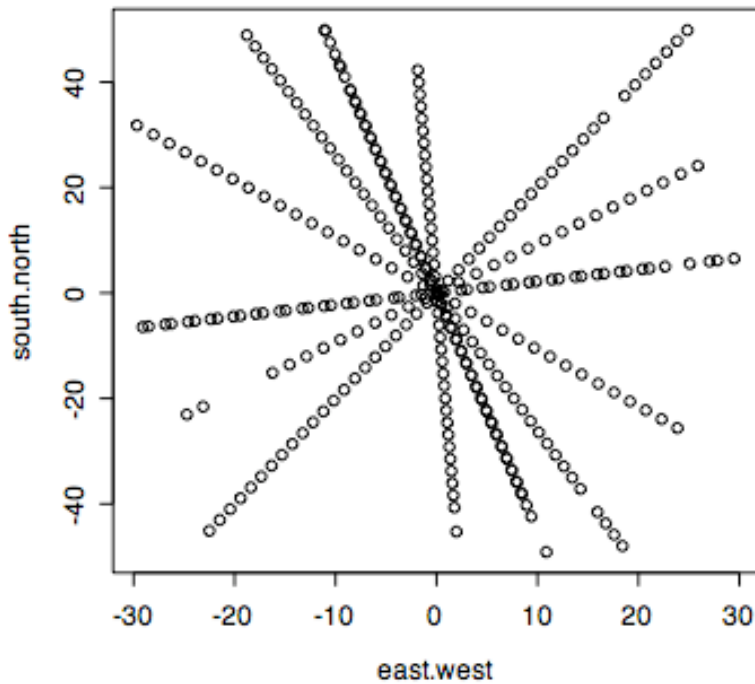
This is what generates the NOX values at the ER-CR grid (as opposed to observed NOX at observed ER-CR values)

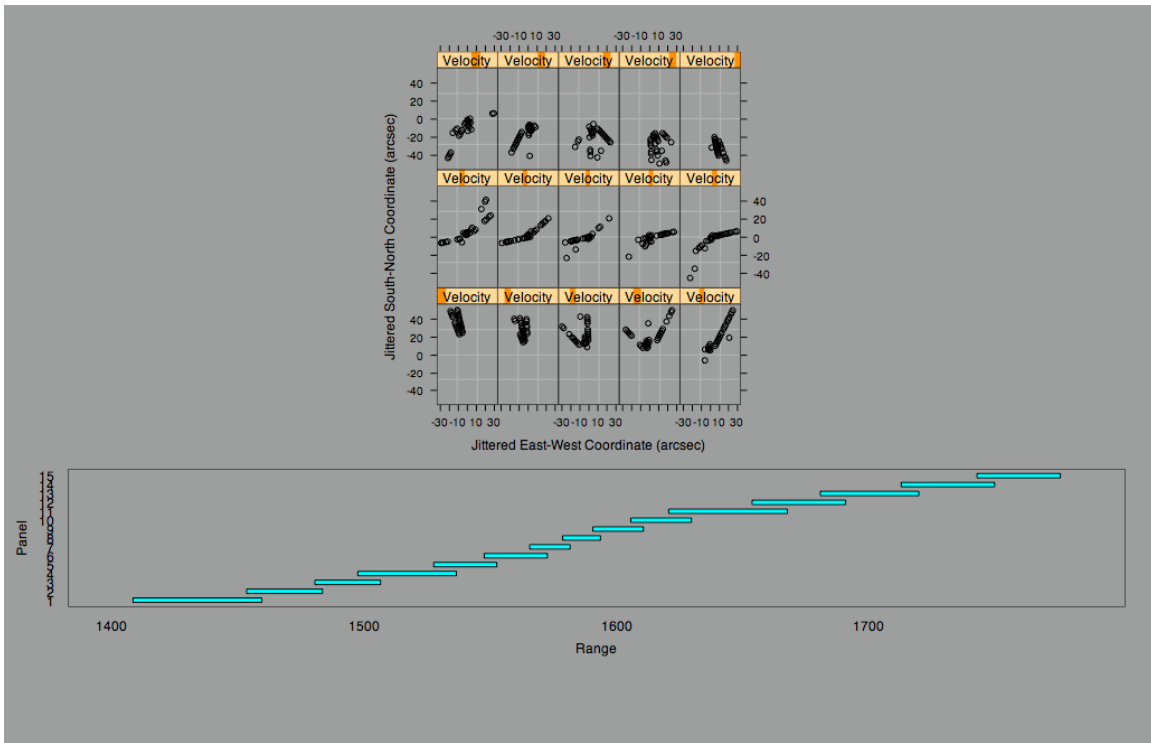
The galaxy data: 415 measurements of the velocity of a galaxy. the direction of the sighting is carefully noted. (slit angle and radial position just give the same information in a different way.)

	east.west	south.north	slit.angle	radial.position	velocity
3	8.46278858	-38.17317200	102.5	39.099998	1769
4	7.96497774	-35.92769241	102.5	36.799999	1749
5	7.46716690	-33.68221283	102.5	34.500001	1749
6	6.96935558	-31.43673134	102.5	32.200000	1758
...
413	-9.09046364	41.00443268	102.5	-42.000000	1456
414	-9.58827496	43.24991226	102.5	-44.299999	1448
415	-10.04279804	45.30013657	102.5	-46.400002	1409
416	-10.54060936	47.54561615	102.5	-48.700001	1416
417	-11.06006432	49.88872528	102.5	-51.099999	1429

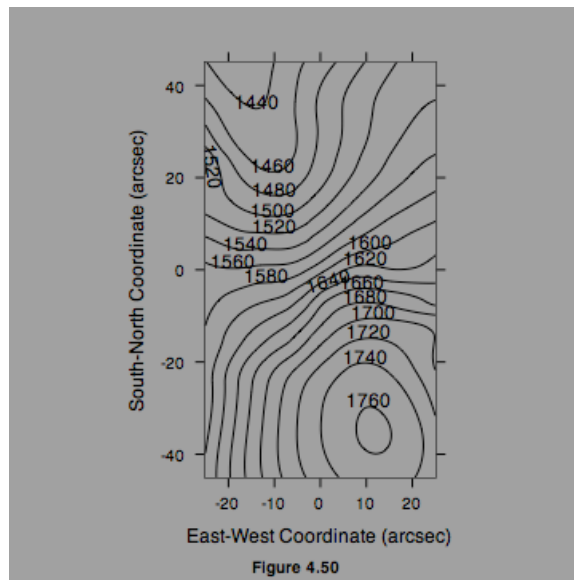
(Note the distinct feature of angular data – but not an issue here).

This is trivariate data but note the structure of the direction coordinates.





Another way to look at this data is the Contour Plot.



This is slightly modified from the "book.4.50()" function. I had to delete the reference to "labels" and also the plot of the intermediate contours. Here is the actual program for the above plot – The **bold** shows the deleted parts.

```
> book.4.50
function()
{
```

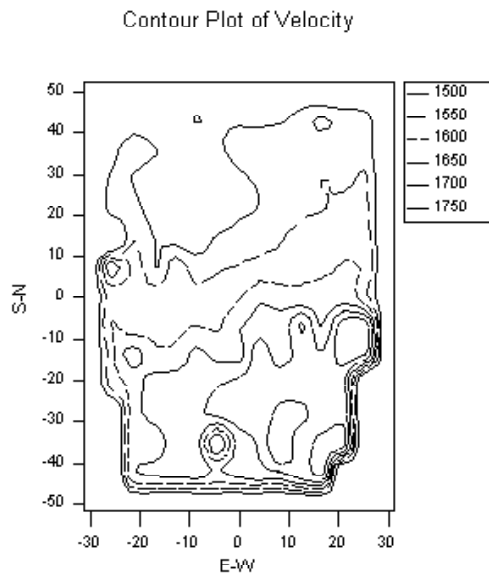
```

attach(galaxy)
galaxy.marginal <- list(east.west = seq(-25, 25,length=101),
  north.south = seq(-45, 45,length=181))
galaxy.grid <- expand.grid(galaxy.marginal)
gal.m <- loess(velocity ~ east.west * north.south, span = 0.25,
  degree = 2, normalize = F, family = "symmetric")
galaxy.fit <- predict(gal.m, galaxy.grid)
x <- range(galaxy.marginal$east.west)
y <- range(galaxy.marginal$north.south)
ans <- contourplot(galaxy.fit ~ galaxy.grid$east.west * galaxy.grid$north.south,
  at = seq(1435, 1755, by = 40),
  panel = function(..., at) {
    #ref <- trellis.par.get("reference.line")
    #panel.contourplot(..., #labels = F,
    #   at = at,
    #   lty = ref$lty, lwd = ref$lwd, col = ref$col)
    panel.contourplot(..., #labels = F,
      at = seq(1460, 1740, by = 40))
    panel.contourplot(...,
      at = seq(1440, 1760, by = 40))
  },
  aspect = diff(range(y))/diff(range(x)),
  sub = list("Figure 4.50",cex=.8),
  xlab = "East-West Coordinate (arcsec)",
  ylab = "South-North Coordinate (arcsec)")
detach()
ans
}

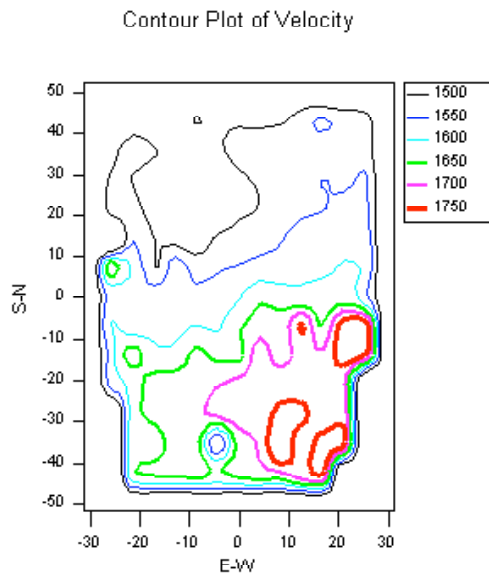
```

Note the necessity for the elevation labels (the velocity in this example). Colour is another way to do it.

Here is a MINITAB version



This can be improved with colour control and line-width control.



With more smoothing the MINITAB output would look like the Cleveland plot. Use of two colours is best, as long as a legend is given with elevations tied to colours.

Back to text ...

It is instructive to think about the mechanics of the contour plot. The data is not a smooth surface, but rather points on a rough surface. You need to smooth the surface first, and this is at a grid of points, not the data points. Then for each grid-square, you need to estimate where a contour line crosses this square. See pp 240-244 for a discussion of this. (This is important because you would not otherwise understand how the contour plot is constructed.)

Level Plots of smoothed surfaces. (pp 245 ff)

Here is the program:

```
book.4.55 <-  
function()  
{  
  attach(galaxy)  
  galaxy.marginal <- list(east.west = seq(-25, 25,length=101),  
    north.south = seq(-45, 45,length=181))  
  galaxy.grid <- expand.grid(galaxy.marginal)  
  gal.m <- loess(velocity ~ east.west * north.south, span = 0.25,  
    degree = 2, normalize = F, family = "symmetric")  
  galaxy.fit <- predict(gal.m, galaxy.grid)  
  n.level <- 48  
  given <- seq(min(galaxy.fit),max(galaxy.fit),length=n.level+1)  
  Velocity <- shingle(as.vector(galaxy.fit),  
    cbind(given[-(n.level+1)],given[-1]))  
  ans <- xyplot(galaxy.grid$n ~ galaxy.grid$e | Velocity,  
    layout = c(8, 6),  
    panel = function(x, y) {  
      panel.grid(v=2, h=2)  
      panel.xyplot(x, y, pch = ".")  
    },  
    aspect=diff(range(galaxy.grid$n))/diff(range(galaxy.grid$e)),  
    xlab = "East-West Coordinate (arcsec)",  
    ylab = "South-North Coordinate (arcsec)")  
  detach()  
  print(ans, position=c(0,.3,1,1), more=T)  
  
  attach(galaxy)  
  galaxy.marginal <- list(east.west = seq(-25, 25,length=101),  
    north.south = seq(-45, 45,length=181))  
  galaxy.grid <- expand.grid(galaxy.marginal)  
  gal.m <- loess(velocity ~ east.west * north.south, span = 0.25,  
    degree = 2, normalize = F, family = "symmetric")
```

```

galaxy.fit <- predict(gal.m, galaxy.grid)
given <- seq(min(galaxy.fit),max(galaxy.fit),length=49)
ans <- plot(shingle(as.vector(galaxy.fit),
  intervals = cbind(given[-length(given)], given[-1])),
  cex=.7,
  scales=list(x=list(cex=.7),y=list(at=seq(10,40,10),cex=.7)),
  sub = list("Figure 4.55",cex=.8),
  ylab="",
  xlab = "Velocity (km/sec)")
detach()
print(ans, position=c(0,.05,1,.35))
invisible()
}

```

However, it needs to be modified to avoid error messages.

Look at Fig 4.55

Do you really understand this plot?

1. Why are there 48 plots when Fig 4.44 on p 231 had only 15?
2. Why do some of the lines in Fig 4.55 seem to have greater width than others?
3. Where are the data points that we saw in Fig 4.44?

Wireframe plots:

