Visualizing in R – advanced plotting

Norsk statistikermøte, Halden, 10. juni 2013

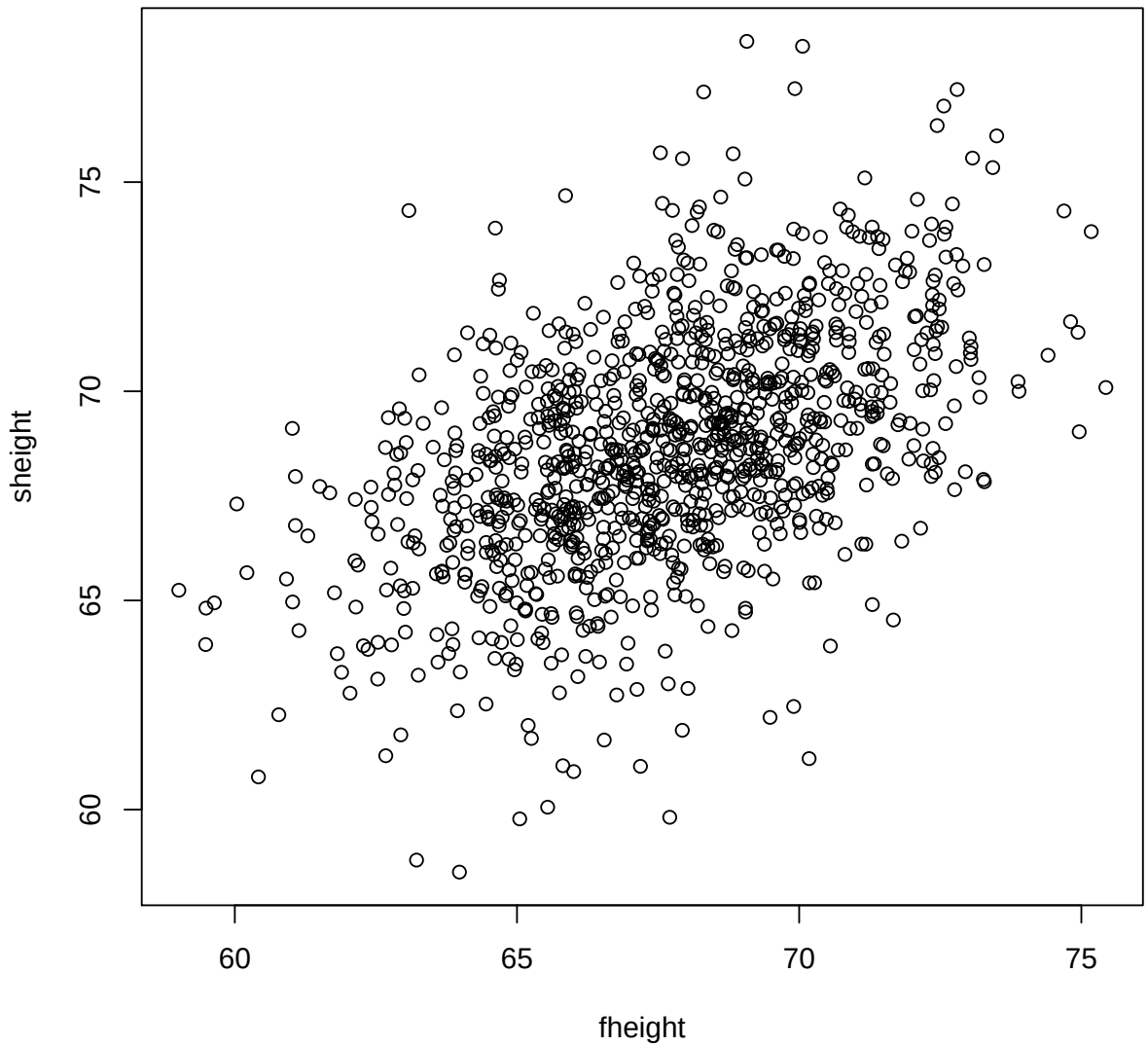
Elisabeth Orskaug
Thordis Thorarinsdottir

Norsk Regnesentral

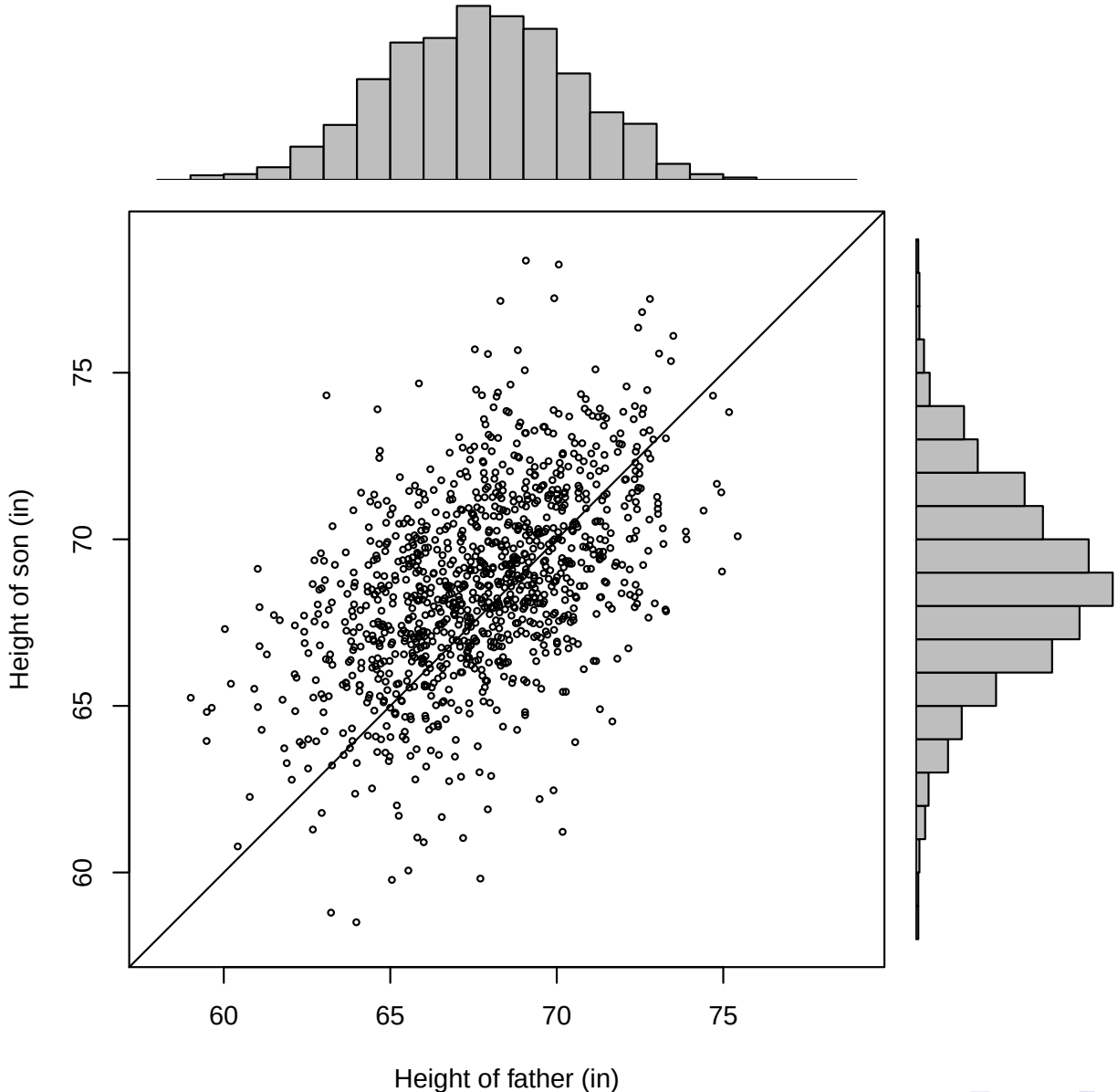
2. Displaying multivariate data

- Scatterplots and pairwise scatter plots
- Parallel coordinate plots

A classic bivariate example



Marginal distributions add information value

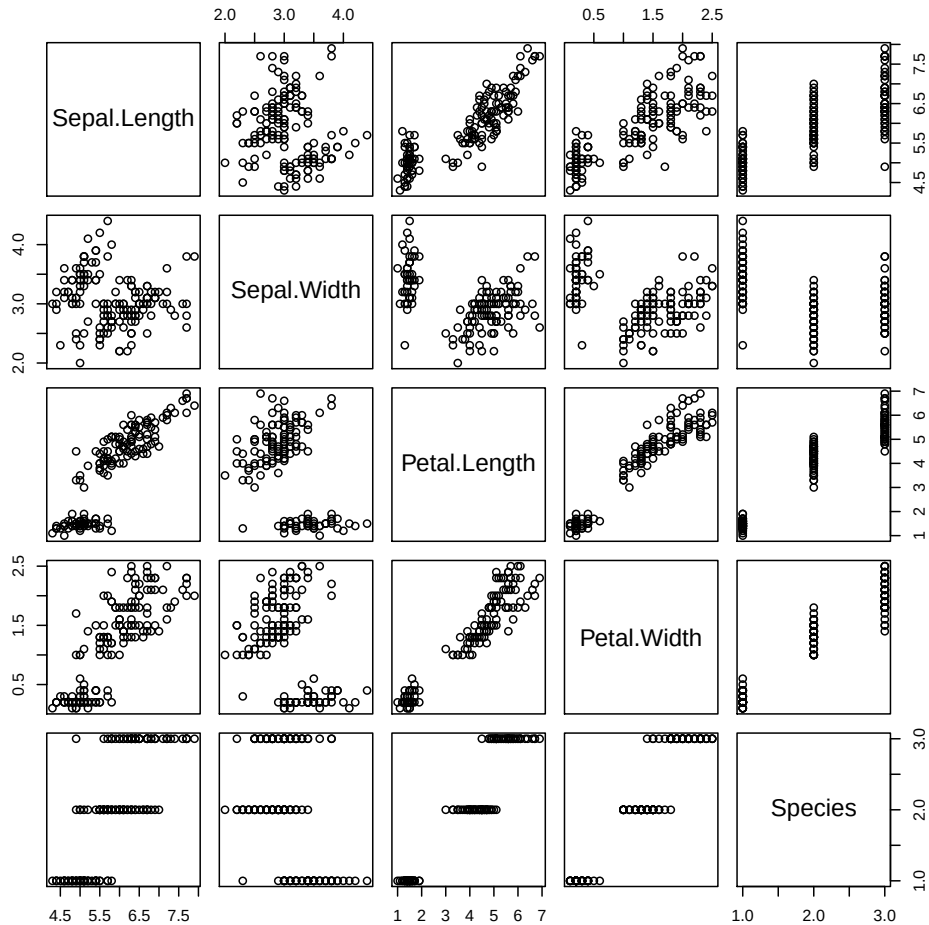


We use a 2×2 layout for this plot

```
pdf(...)
zones=matrix(c(2,0,1,3), ncol=2, byrow=TRUE)
layout(zones, widths=c(4/5,1/5), heights=c(1/5,4/5))
limit <- c(58,79)
fhist = hist(father.son$fheight, breaks=58:79, plot=FALSE)
shist = hist(father.son$sheight, breaks=58:79, plot=FALSE)
top = max(c(shist$counts, fhist$counts))
par(mar=c(4,4,1,1))
plot(father.son, xlim=limit, ylim=limit, cex=0.5,
     xlab="Height of father (in)", ylab="Height of son (in)")
abline(a=0, b=1)
par(mar=c(0,4,1,1))
barplot(fhist$counts, axes=FALSE, ylim=c(0, top), space=0)
par(mar=c(4,0,1,1))
barplot(shist$counts, axes=FALSE, xlim=c(0, top), space=0,
        horiz=TRUE)
dev.off()
```

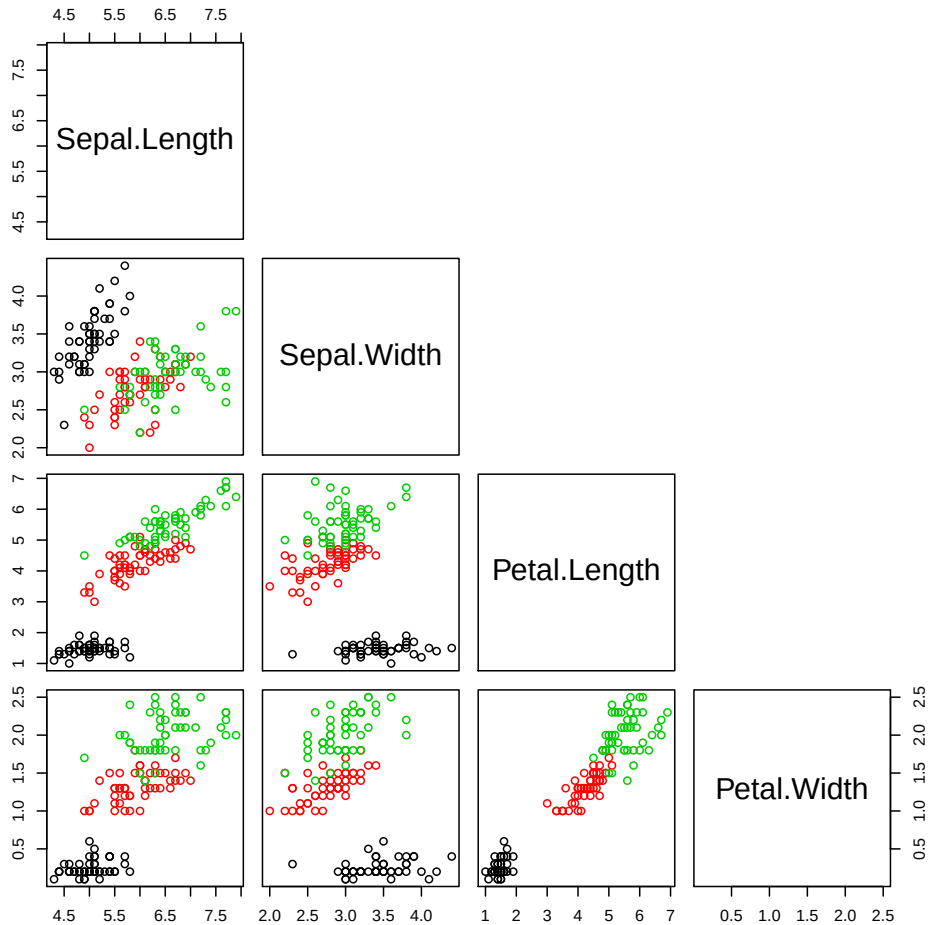
In higher dimensions, we use pairwise scatterplots

```
pairs(iris)
```



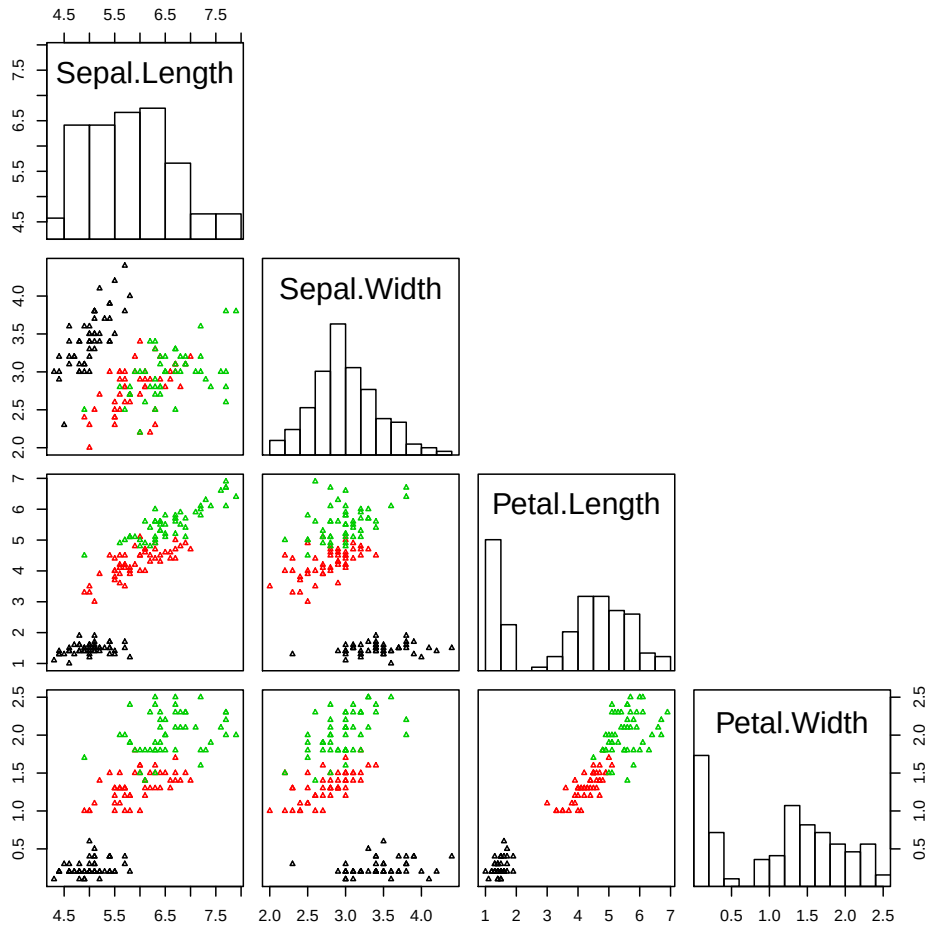
We only need one panel

```
pairs(iris[1:4],
      upper.panel=NULL,
      col=iris$Species)
```



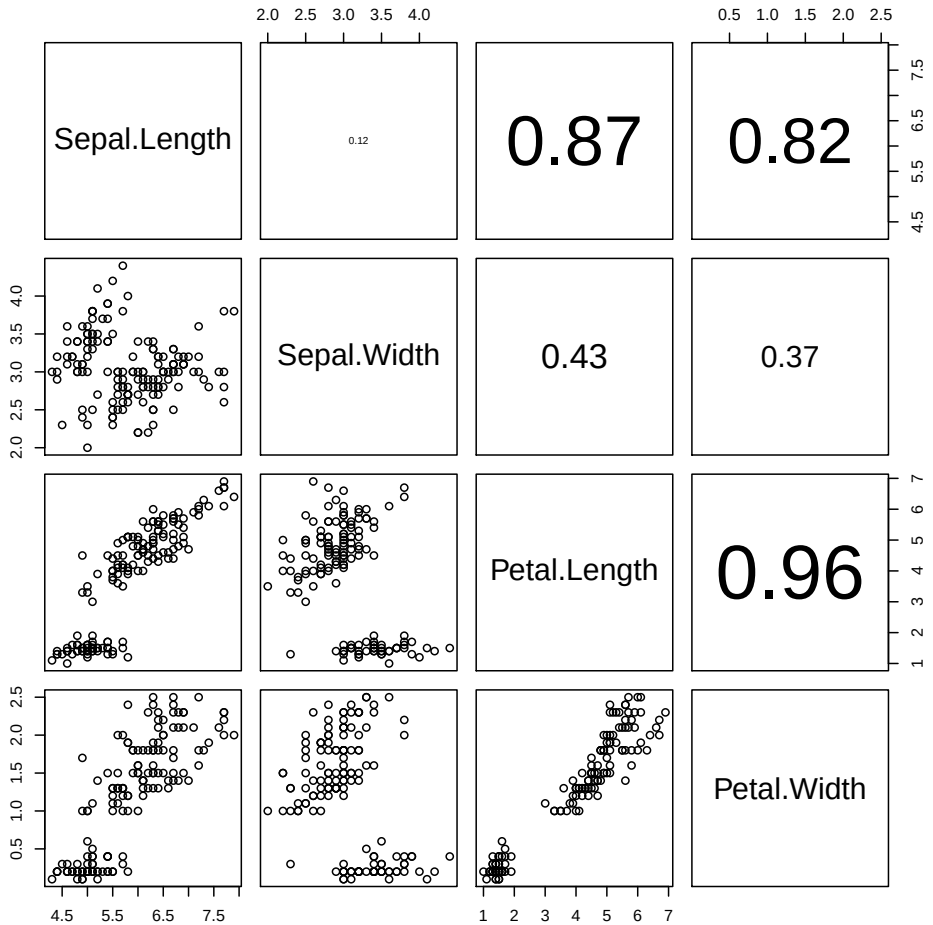
Marginal histograms can be added

```
panel.hist <- function(x, ...)
{
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks
    nB <- length(breaks)
    y <- h$counts
    y <- y/max(y)
    rect(breaks[-nB], 0,
         breaks[-1], y)
}
pairs(iris[1:4], col=iris$Species,
      cex = 0.5, pch = 24,
      diag.panel = panel.hist,
      upper.panel=NULL)
```

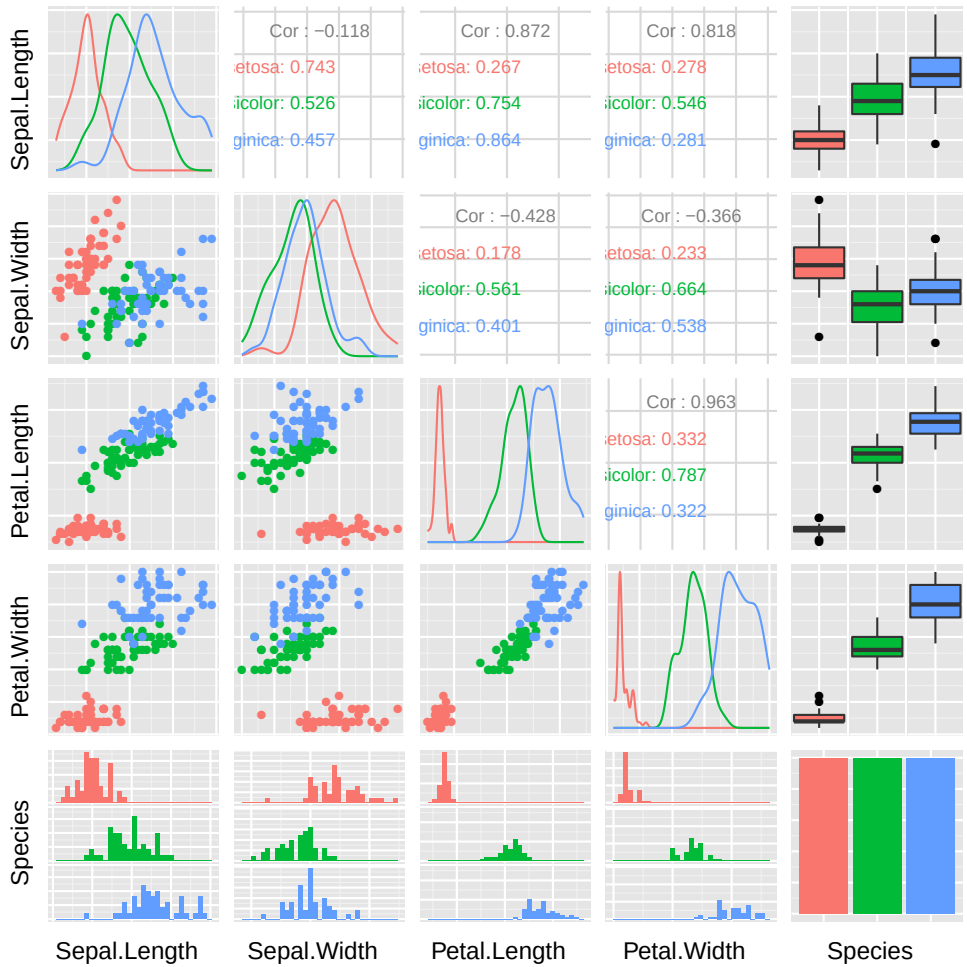


We can include the correlation

```
panel.cor <- function(x, y,
                      digits = 2,
                      prefix = "")
{
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789),
                  digits = digits)[1]
    txt <- paste0(prefix, txt)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt,
         cex = cex.cor * r)
}
pairs(iris[1:4],
      upper.panel = panel.cor)
```



Alternatively, we can use `ggpairs()`



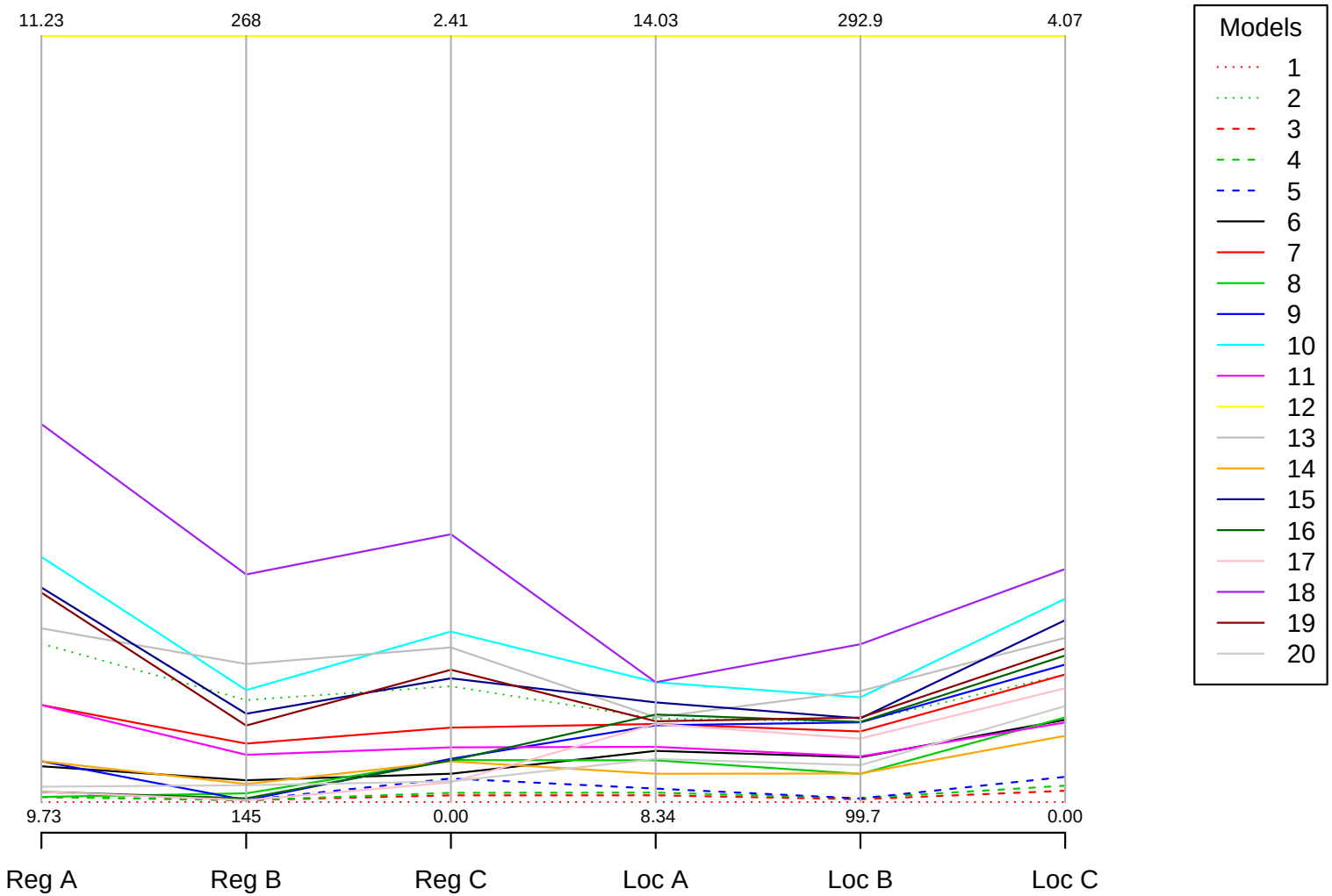

```
library(GGally)
ggpairs(iris, colour="Species", axisLabels="none")
```

Large tables are cumbersome, especially in presentations

Section 2.2

	Regional Score			Local Score		
Model	A	B	C	A	B	C
1	9.73	144.9	0.000	8.34	99.7	0.000
2	10.04	161.3	0.364	8.96	120.0	0.678
3	9.74	145.2	0.021	8.39	100.4	0.060
4	9.74	145.2	0.029	8.41	100.7	0.087
5	9.75	145.2	0.074	8.44	100.6	0.134
6	9.80	148.4	0.089	8.72	111.0	0.437
7	9.92	154.3	0.234	8.92	117.5	0.678
8	9.74	146.3	0.132	8.65	106.8	0.450
9	9.81	145.2	0.136	8.91	119.8	0.731
10	10.21	162.9	0.536	9.23	126.1	1.081
11	9.92	152.5	0.172	8.75	111.2	0.423
12	11.23	268.1	2.409	14.03	292.9	4.073
13	10.07	167.1	0.486	8.97	127.7	0.873
14	9.81	147.8	0.128	8.55	106.9	0.352
15	10.15	159.1	0.389	9.08	120.8	0.967
16	9.75	145.5	0.131	8.99	119.9	0.779
17	9.75	145.2	0.062	8.92	115.7	0.605
18	10.47	181.5	0.842	9.23	139.5	1.239
19	10.14	157.2	0.416	8.94	121.0	0.817
20	9.76	147.6	0.065	8.66	109.0	0.510

Parallel coordinate plot offers a graphical view



The code is very simple

```
library(MASS)

pdf(...)
par(mar=c(5,4,4,8)+0.1, xpd=TRUE)
parcoord(scores, lty=lines, col=colors, var.label=TRUE)
legend("topright", inset=c(-0.2,0), legend=c(1:20), lty=lines,
       col=colors, title="Models")
dev.off()
```

`scores` is a data frame with 20 lines and 6 columns, where the column names are “Reg A”, “Reg B”,...