Explore the data

Anja Bråthen Kristoffersen

Biomedical Research Group

UNIVERSITY OF OSLO

Probability distributions

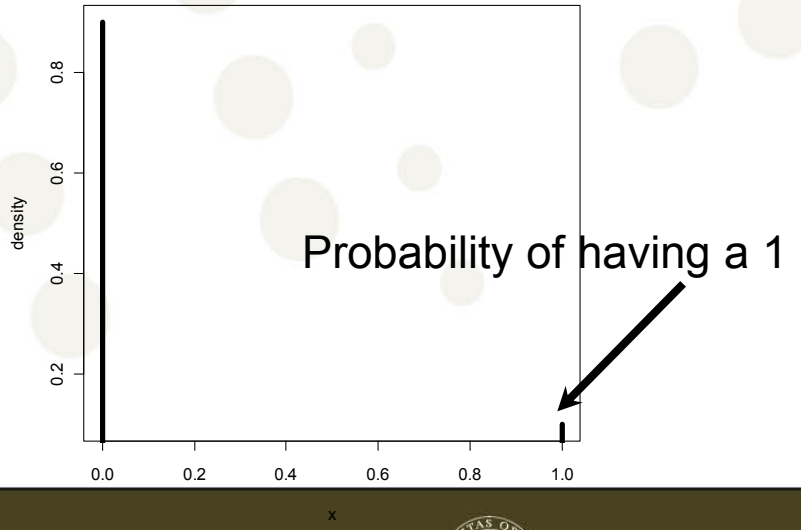
Can be either discrete or continuous (uniform, bernoulli, normal, etc)

Defined by a density function, *p*(x) or *f*(x)

Bernoulli distribution Be(p)

Flip a coin (T=0, H=1). Probability of H is 0.1.

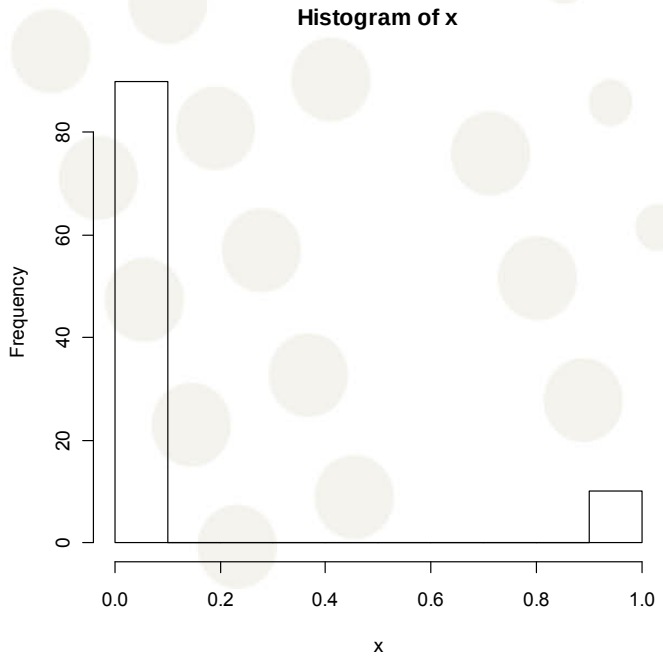
```
x <- 0:1
f <- dbinom(x, size=1, prob=0.1)
plot(x,f,xlab="x",ylab="density",type="h",lwd=5)
```



Probability distributions

Random sampling
Generate 100 observations from a Be(0.1)

```
x<-rbinom(100, size=1, prob=0.1)
hist(x)
```

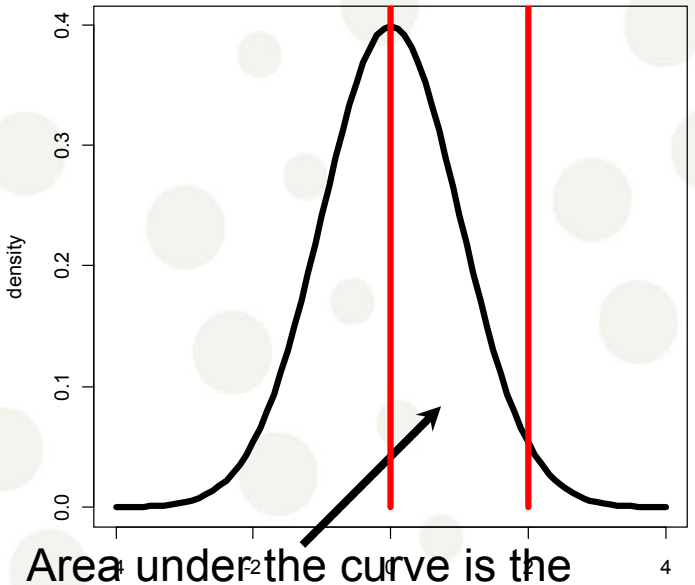


Probability distributions

Normal distribution $N(\mu,\sigma^2)$

μ is the mean and σ^2 is the variance

```
x <- seq(-4, 4, 0.1)
f <- dnorm(x, mean=0, sd=1)
plot(x, f, xlab="x", ylab="density", lwd=5, type="l")
lines(c(0, 0), c(0, 0.5), col=2, lwd=5)
lines(c(2, 2), c(0, 0.5), col=2, lwd=5)
```

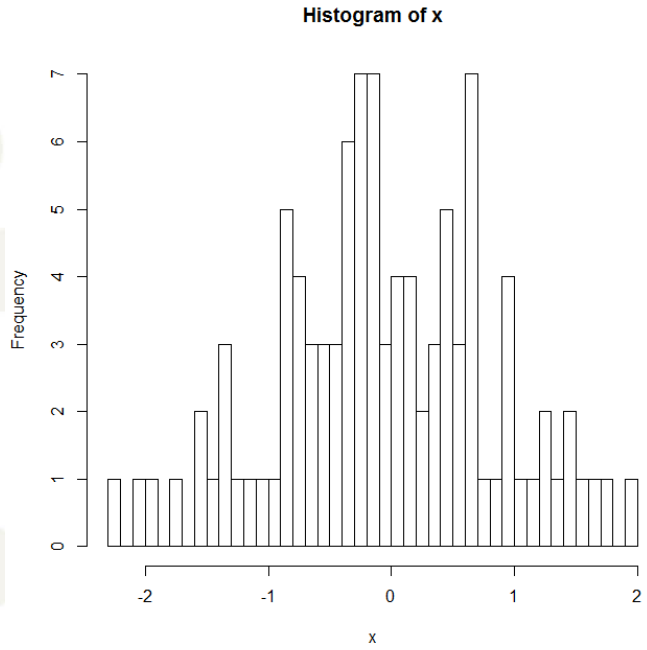


Area under the curve is the probability of having an observation between 0 and 2.

Probability distributions

Random sampling
Generate 100 observations from a N(0,1)

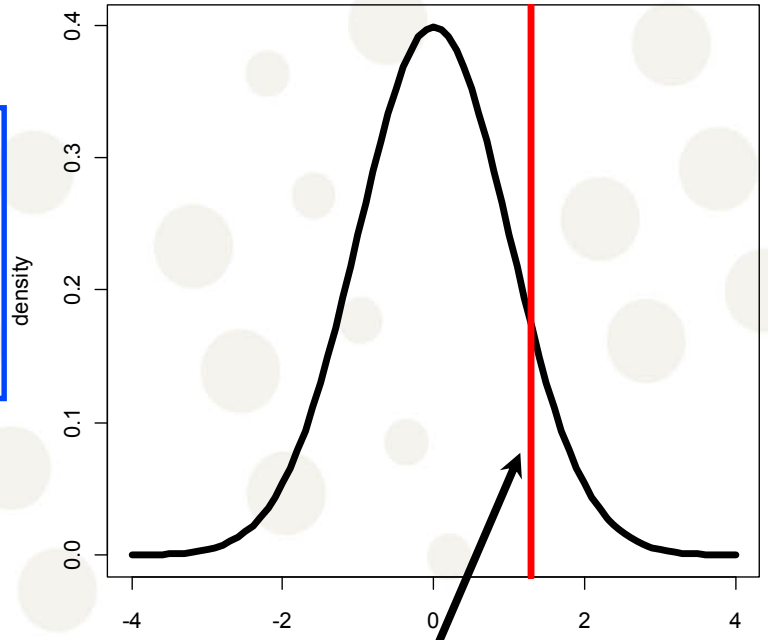
```
x <- rnorm(100, mean = 0, sd =1)
hist(x)
hist(x,breaks= 50)
```



Quantiles

(Theoritical) Quantiles: The *p*-quantile is the value with the property that there is a probability *p* of getting a value less than or equal to it.

```
q90 <- qnorm(0.90, mean = 0, sd = 1)
x <- seq(-4, 4, 0.1)
f <- dnorm(x, mean=0, sd=1)
plot(x, f, xlab="x",ylab= "density", type="l", lwd=5)
abline(v = q90, col = 2, lwd = 5)
```



90% of the prob. (area under the curve) is on the left of red vertical line.

The 50% quantile is the same as the median

Descriptive Statistics

Empirical Quantiles: The *p*-quantile is the value with the property that *p*% of the observations are less than or equal to it.

Empirical quantiles can easily be obtained in R.

```
x<-rnorm(100, mean = 0, sd = 1)
quantile(x)
```

```
       0%        25%        50%        75%       100%
-2.2719255 -0.6088466 -0.0594199  0.6558911  2.5819589
```

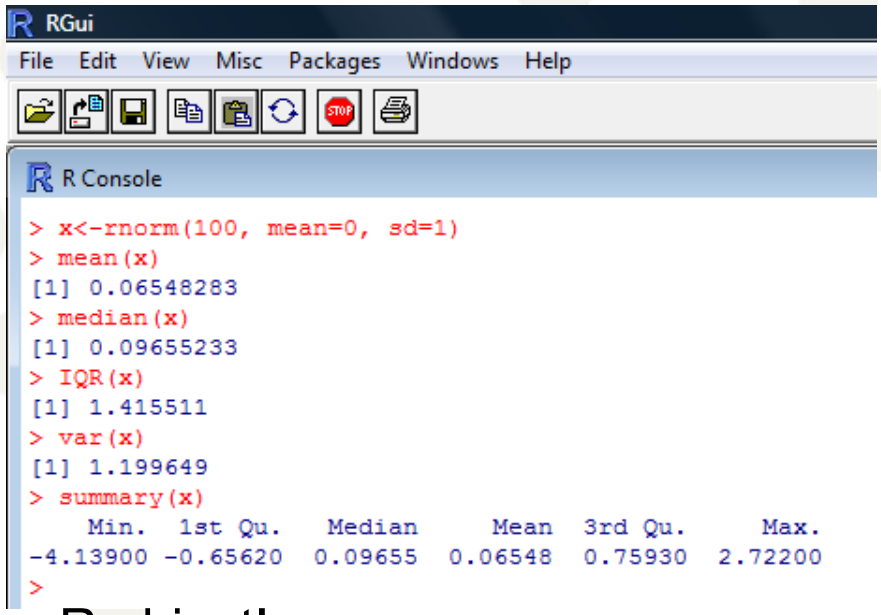
```
quantile(x, probs = c(0.1, 0.2, 0.9))
```

```
       10%        20%        90%
-1.1744996 -0.8267067  1.3834892
```

Descriptive Statistics

We often need to quickly 'quantify' a data set. This can be done using a set of summary statistics (mean, median, variance, standard deviation)

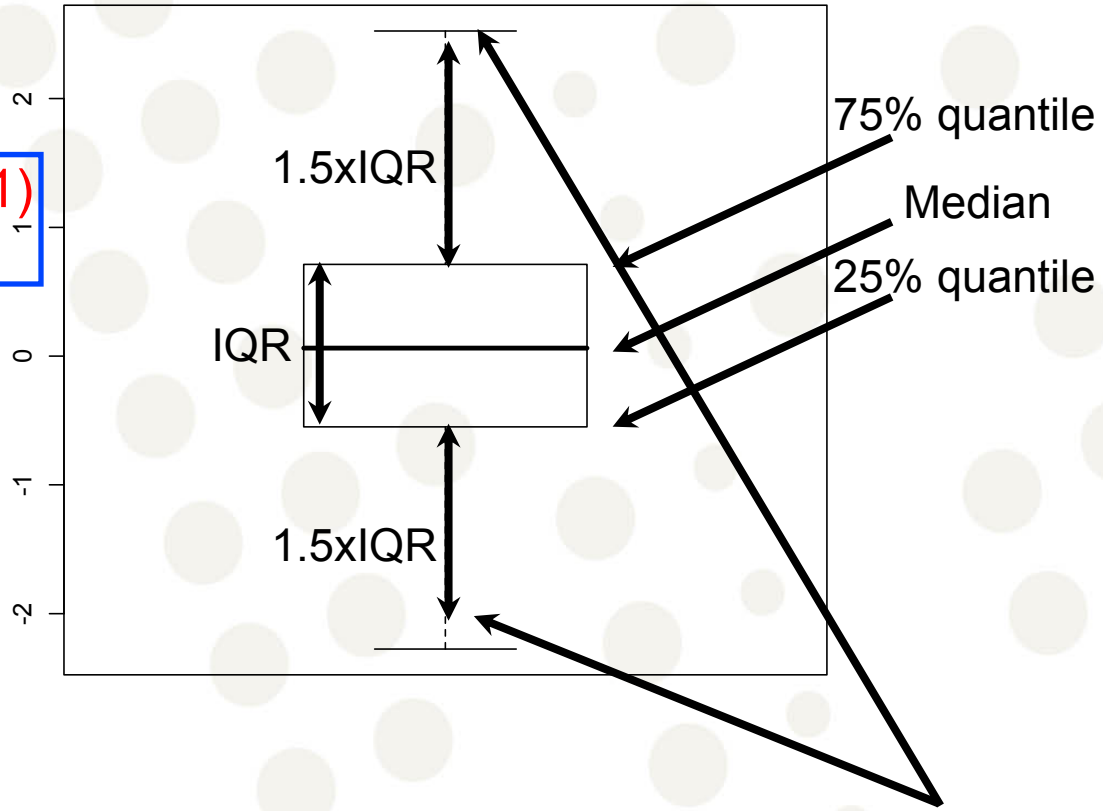
```
x <- rnorm(100, mean=0, sd=1)
mean(x)
median(x)
IQR(x)
var(x)
sd(x)
summary(x)
```



summary() can be used for almost any R object!
R is object oriented (methods/classes).

Descriptive Statistics - Box-plot

```
x <- rnorm(100, mean=0, sd=1)
boxplot(x)
```



IQR= 75% quantile -25% quantile= Inter Quantile Range

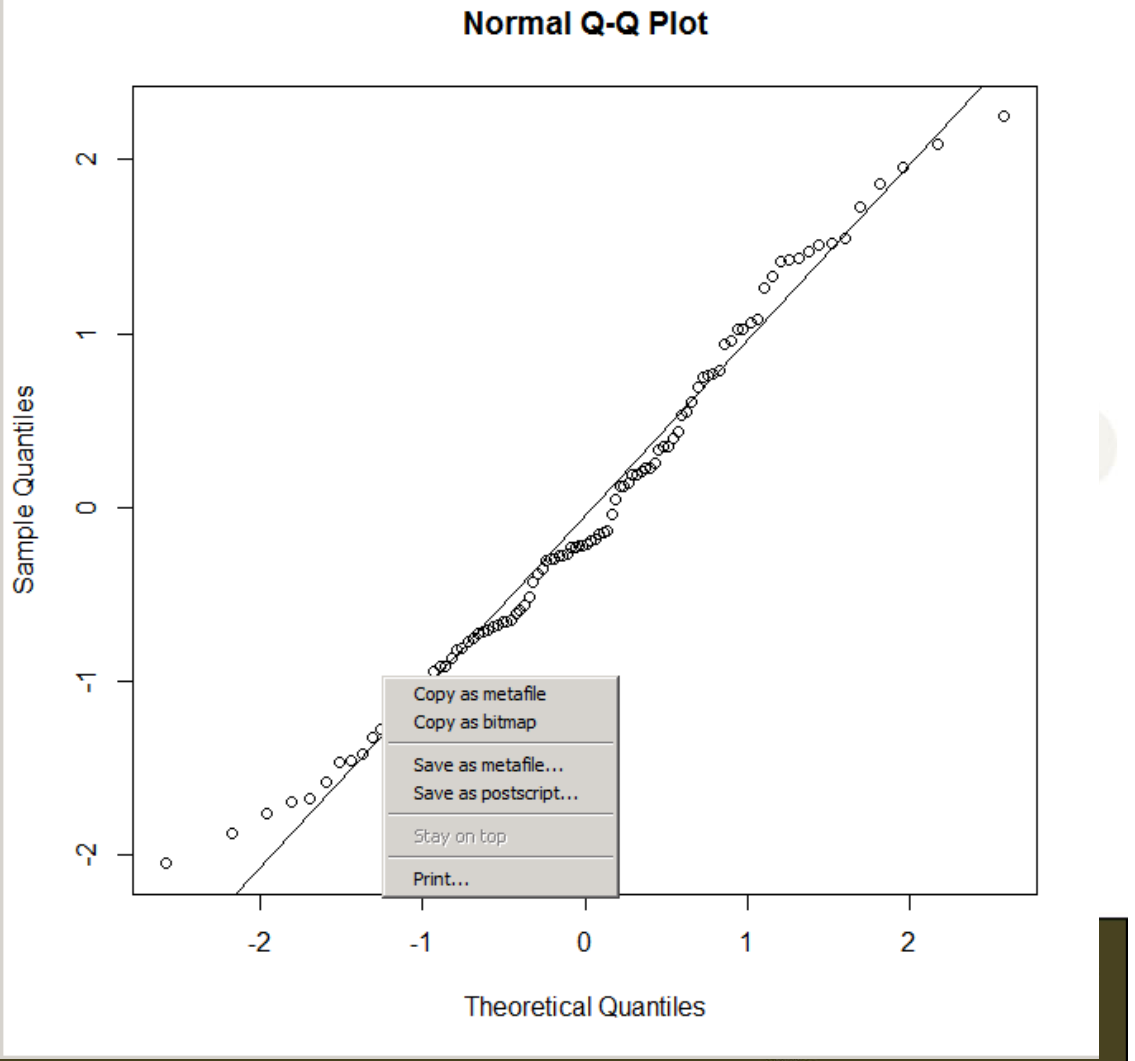
Everything above or below are considered outliers

QQ-plot

- Many statistical methods make some assumption about the distribution of the data (e.g. Normal).
- The quantile-quantile plot provides a way to visually verify such assumptions.
- The QQ-plot shows the theoretical quantiles versus the empirical quantiles. If the distribution assumed (theoretical one) is indeed the correct one, we should observe a straight line with gradient equal to 1.

QQ-plot

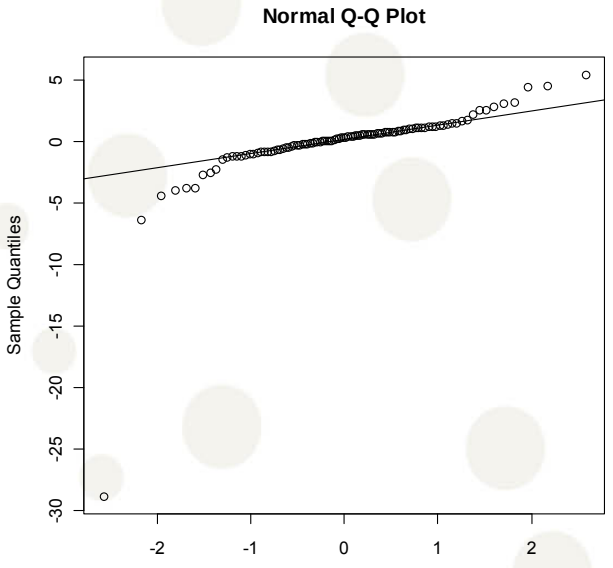
```
x <- rnorm(100, mean=0, sd=1)
qqnorm(x)
qqline(x)
```



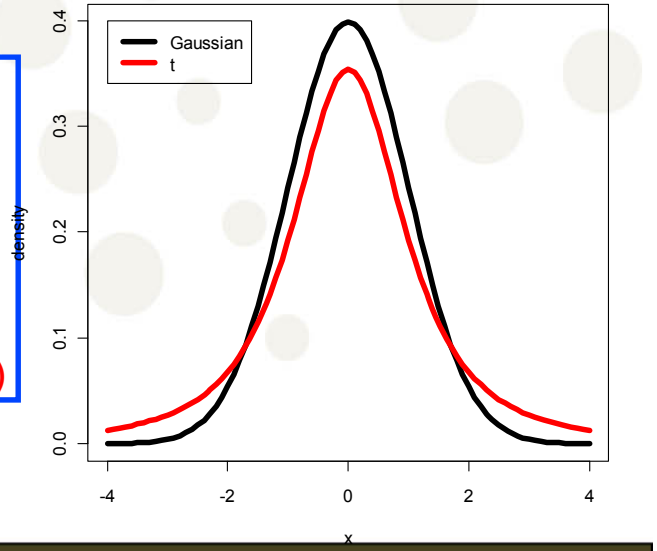
QQ-plot

```
x<-rt(100,df=2)
qqnorm(x)
qqline(x)
```

Clearly the *t* distribution with two degrees of freedom is different from the Normal!



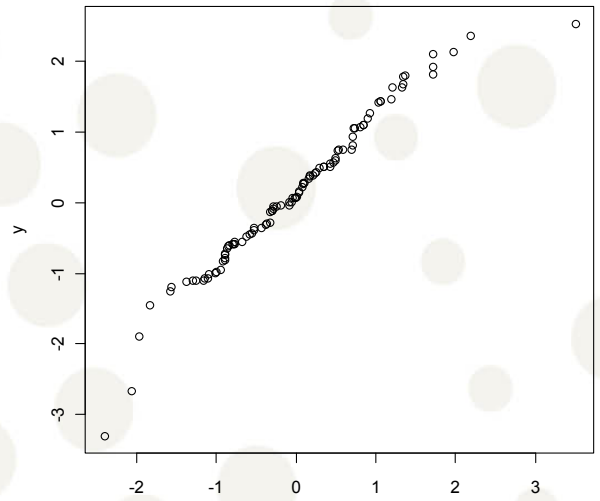

```
x <- seq(-4, 4, 0.1)
f1 <- dnorm(x, mean = 0, sd = 1)
f2 <- dt(x, df = 2)
plot(x, f1, xlab = "x", ylab = "density", lwd = 5, type = "l")
lines(x, f2, xlab = "x", ylab = "density", lwd = 5, col = 2)
legend(-4, 0.4, c("Gaussian", "t"), col = c(1, 2), lty = c(1, 1), lwd = 5)
```



QQ-plot

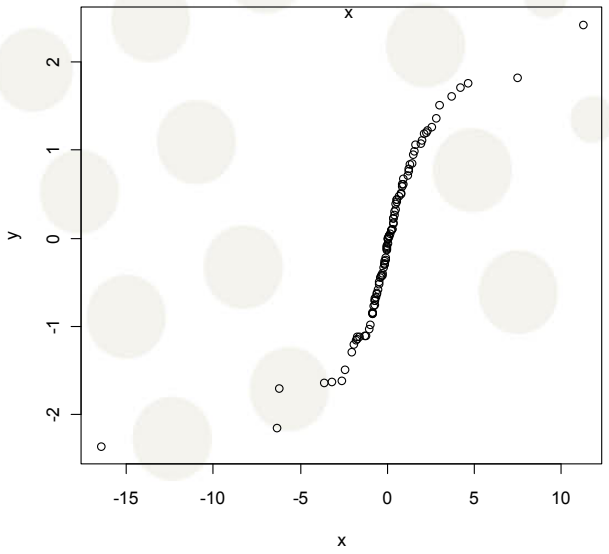
Comparing two samples

```
x<-rnorm(100, mean=0, sd=1)
y<-rnorm(100, mean=0, sd=1)
qqplot(x,y)
```




```
x<-rt(100, df=2)
y<-rnorm(100, mean=0, sd=1)
qqplot(x,y)
```

Ex: Try with different values of df.



Main idea behind quantile normalization

read.data()

- If your dataset contains columns name, use header = TRUE
  - Othervice all data will be read as text not numeric
- How are they separated, preferable with tabulator, use sep = "\t“
  - Othervice you will have problem with text containing space
- R use “.” as decimal points, if your txt file use ”,” use dec = ”,”

head(dat)

```
> dat <- read.table(file = "M:/Undervisning/Statistikk/DLBCLpatientDataNEW.txt", header =TRUE, sep="\t")
> head(dat)
  DLBCL.sample..LYM.number. Analysis.Set Follow.up..years. Status.at.follow.up Subgroup IPI.Group
1                         2     Training               4.0               Alive      GCB       Low
2                         4     Training               4.9               Alive      GCB    Medium
3                         6     Training               5.6               Alive      GCB       Low
4                         7     Training              12.1               Alive      GCB    Medium
5                         8     Training               0.6                Dead      ABC    Medium
6                        11     Training               0.3                Dead      GCB      High
  Germinal.center.B.cell.signature Lymph.node.signature Proliferation.signature  BMP6 MHC.class.II.signature
1                             0.28                -0.07                   -0.56  0.46                    0.57
2                             1.01                -1.15                   -1.04  0.23                    0.63
3                             0.83                -2.11                    0.52 -0.28                    0.38
4                             0.89                -1.33                    0.01 -0.64                    0.93
5                             0.27                -1.56                    1.56 -0.67                   -2.50
6                            -0.05                 0.06                   -0.68 -0.38                   -2.32
  Outcome.predictor.score
1                   -0.23
2                   -0.38
3                    0.20
4                   -0.41
5                    1.25
6                    0.44
```

summary(dat)

```
> summary(dat)
 DLBCL.sample..LYM.number.     Analysis.Set Follow.up..years.
 Min.   :  1.00            Training  :160   Min.   : 0.000
 1st Qu.: 91.75            Validation: 80   1st Qu.: 0.900
 Median :177.50                             Median : 2.800
 Mean   :190.29                             Mean   : 4.411
 3rd Qu.:284.25                             3rd Qu.: 7.100
 Max.   :439.00                             Max.   :21.800
 Status.at.follow.up     Subgroup     IPI.Group
 Alive:102           ABC     : 73   High   : 32
 Dead :138           GCB     :115   Low    : 82
                     Type III: 52   Medium :108
                                    missing:  1
                                    NA's   : 17

 Germinal.center.B.cell.signature Lymph.node.signature Proliferation.signature
 Min.   :-2.61000                 Min.   :-2.6500      Min.   :-1.700000
 1st Qu.:-0.91000                 1st Qu.:-0.8675      1st Qu.:-0.410000
 Median :-0.16000                 Median : 0.0600      Median :-0.010000
 Mean   :-0.03062                 Mean   : 0.0065      Mean   : 0.005958
 3rd Qu.: 0.86000                 3rd Qu.: 0.8675      3rd Qu.: 0.412500
 Max.   : 2.48000                 Max.   : 2.9800      Max.   : 2.180000
      BMP6          MHC.class.II.signature Outcome.predictor.score
 Min.   :-1.87000   Min.   :-3.020000      Min.   :-1.700000
 1st Qu.:-0.65250   1st Qu.:-0.537500      1st Qu.:-0.537500
 Median :-0.13500   Median : 0.125000      Median :-0.085000
 Mean   :-0.04362   Mean   :-0.006083      Mean   :-0.003208
 3rd Qu.: 0.49250   3rd Qu.: 0.680000      3rd Qu.: 0.522500
 Max.   : 2.69000   Max.   : 1.890000      Max.   : 2.360000
>
```

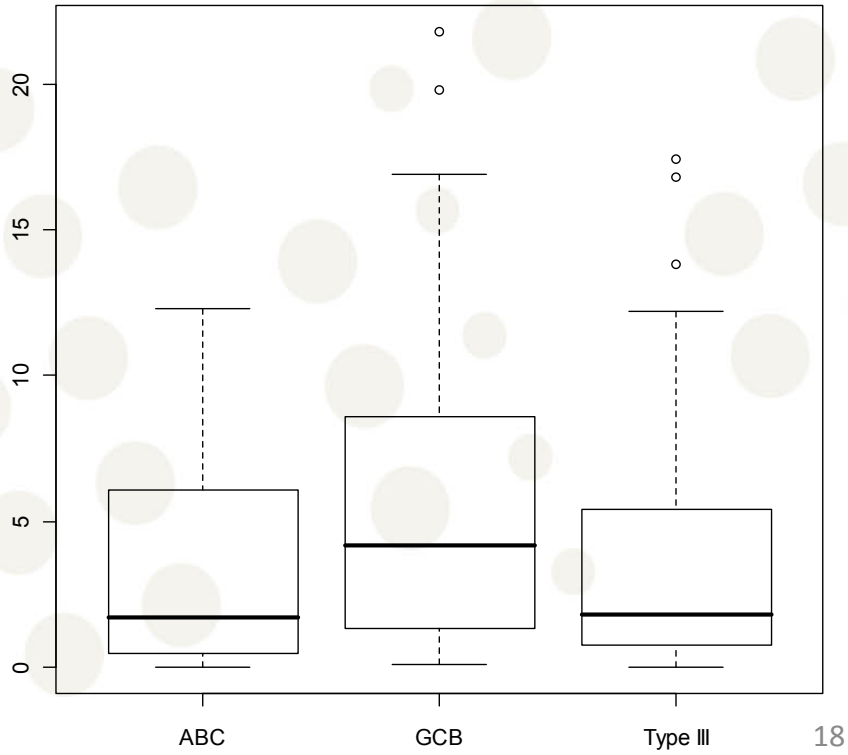
Extract the subgroup ABC

```
> datABC <- dat[dat$Subgroup=="ABC",]
> dim(datABC)
[1] 73 12
```

Boxplot: follow up time for each subgroup

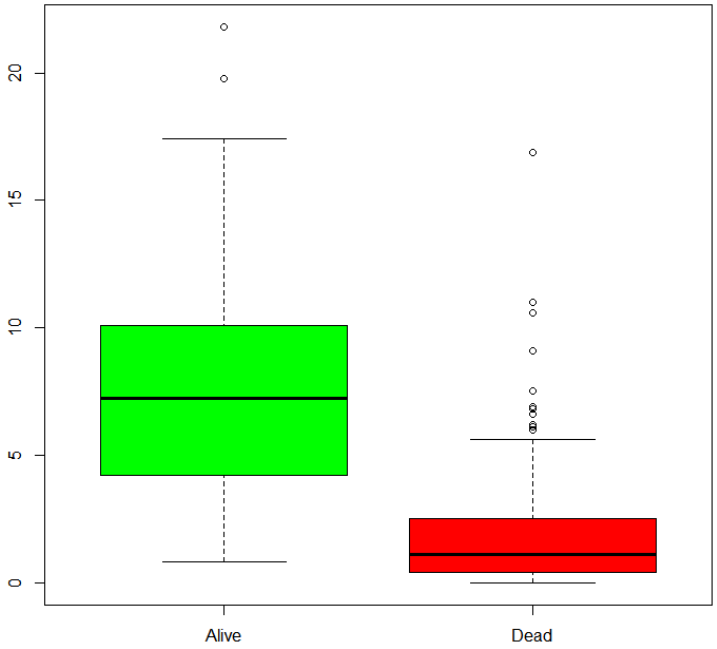
```
boxplot(Follow.up..years. ~ Subgroup, data = dat)
```

The boxplot function can be used to display several variables at a time!



Boxplot: follow up time for each status group

```
boxplot(Follow.up..years. ~ Status.at.follow.up, data = dat,
        col = c("green", "red"))
```



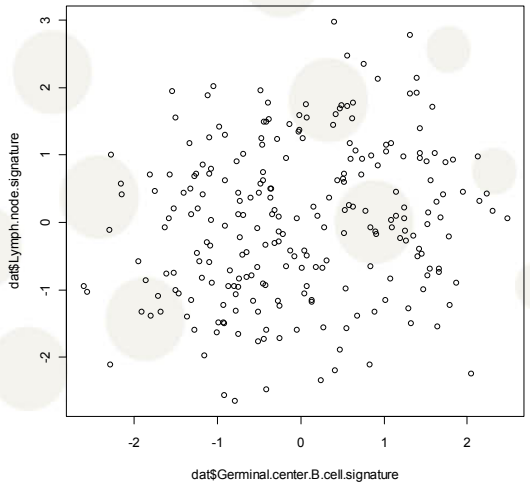
Scatter plots

Biological data sets often contain several variables
So they are multivariate.

Scatter plots allow us to look at two variables at a time.

```
plot(dat$Germinal.center.B.cell.signature, dat$Lymph.node.signature)
cor(dat$Germinal.center.B.cell.signature, dat$Lymph.node.signature)
[1] 0.1633608
```

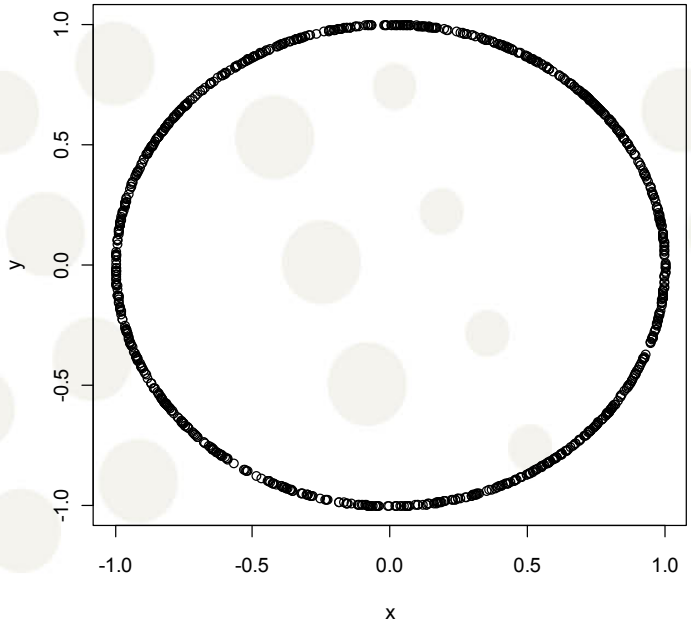
This can be used
to assess independence!



Scatter plots vs. correlations

Correlation is only good for linear dependence.

```
# Quick comment on correlation
theta <- runif(1000, 0, 2*pi)
x <- cos(theta)
y <- sin(theta)
plot(x, y)
cor(x, y)
```



What is the correlation?

[1] 0.01274363

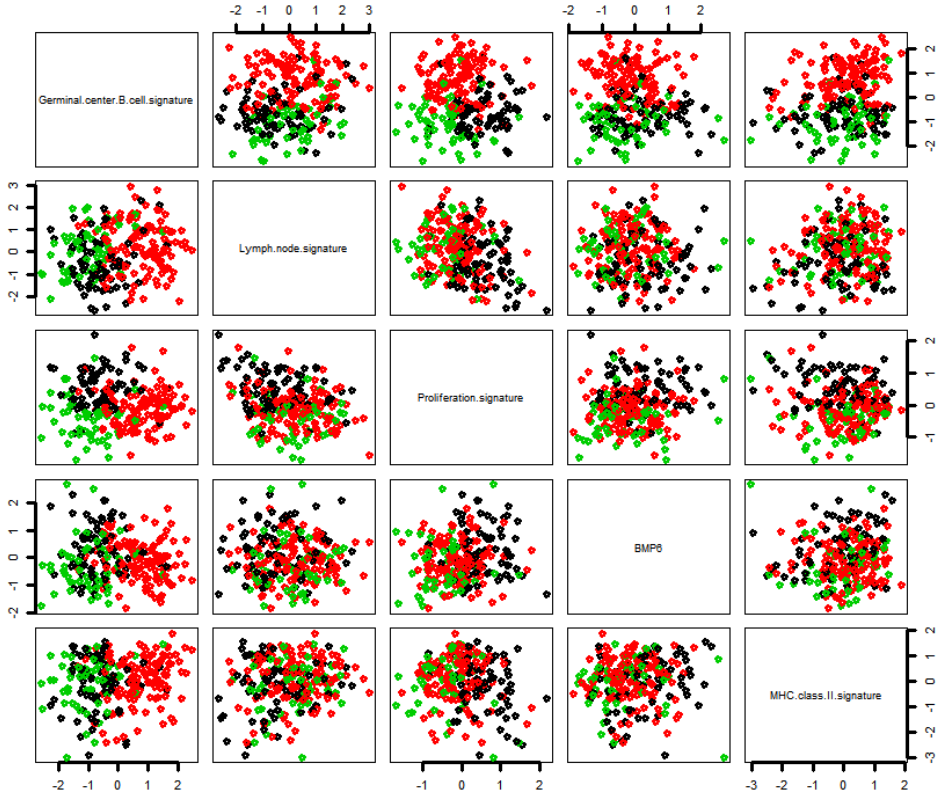
Trellis graphics

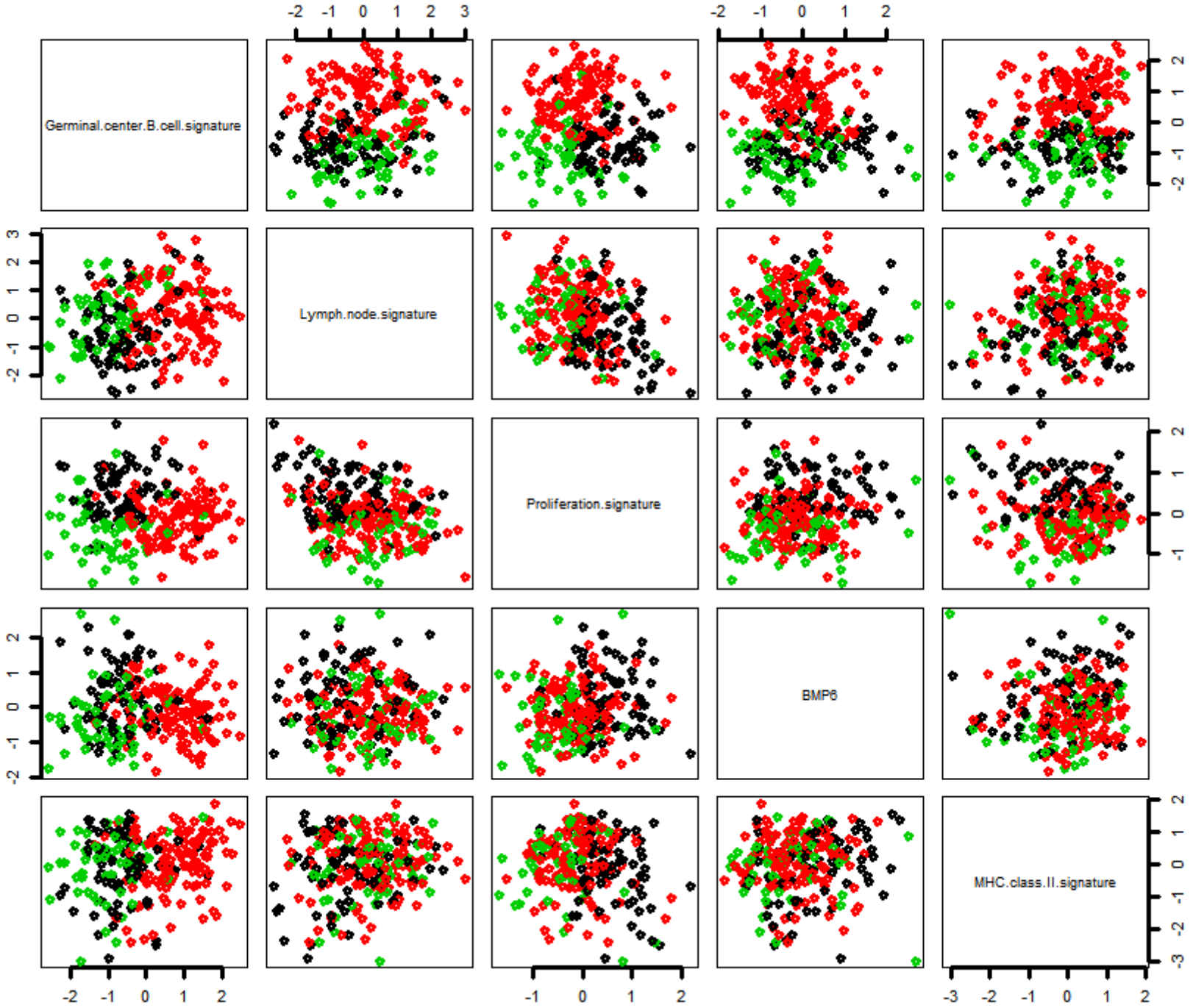
Trellis Graphics is a family of techniques for viewing complex, multi-variable data sets.

```
pairs(dat[,7:11],
      col = dat[,5],
      lwd = 3)
```

Note that the plotting symbol is enlarged.

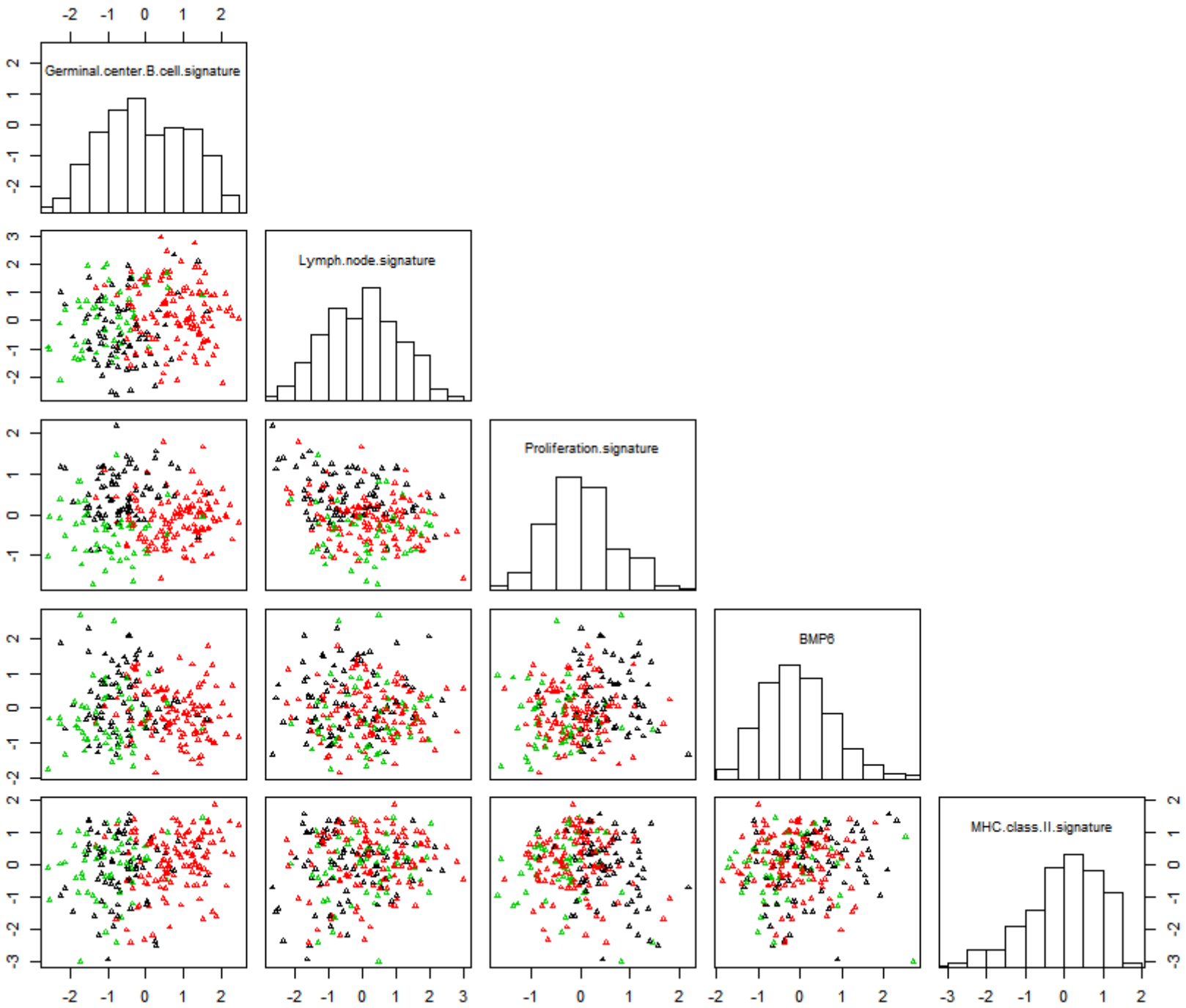
Many more possibilities in the ‘lattice’ package




Germinal.center.B.cell.signature
Lymph.node.signature
Proliferation.signature
BMP6
MHC.class.II.signature

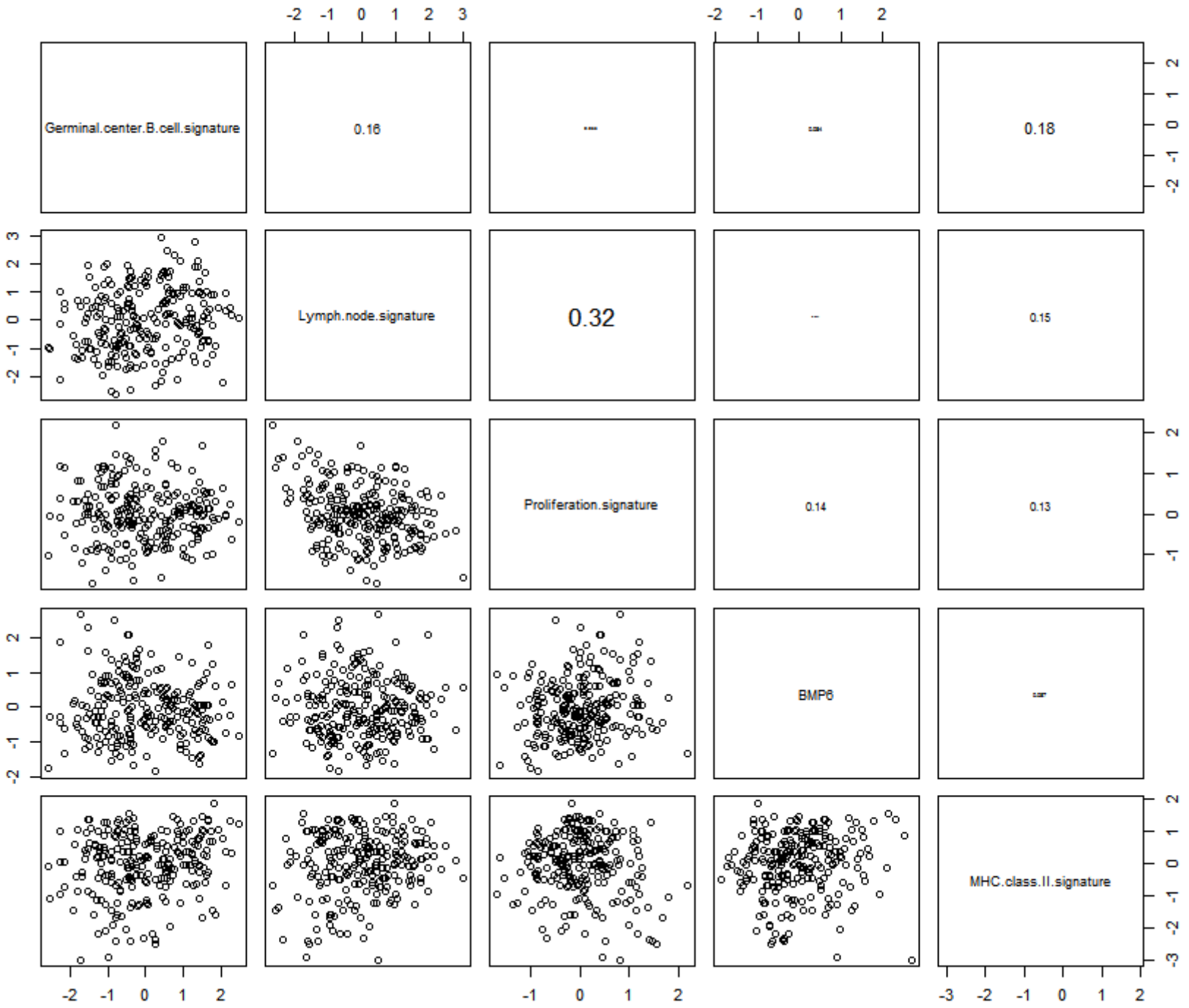
Trellis graphics
with histograms on the diagonal

```
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(dat[,7:11], col = dat[,5], cex = 0.5, pch = 24,
        diag.panel = panel.hist, upper.panel=NULL)
```


Germinal.center.B.cell.signature
Lymph.node.signature
Proliferation.signature
BMP6
MHC.class.II.signature

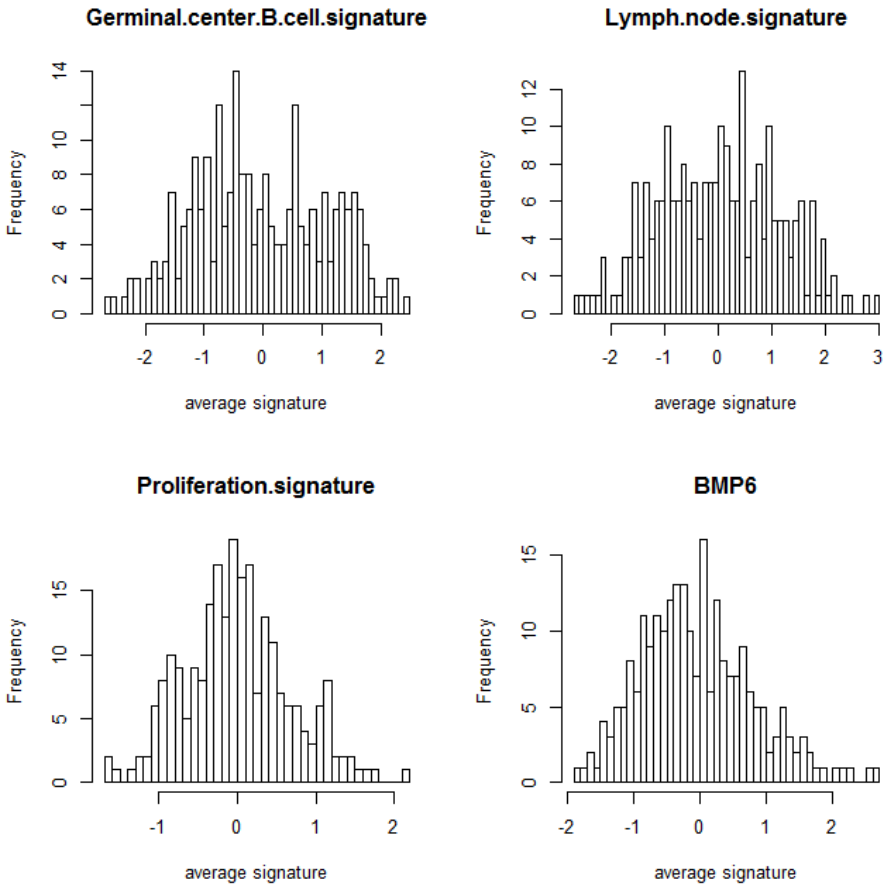
Trellis graphics with correlation in upper panel

```
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(dat[,7:11], upper.panel = panel.cor)
```


Germinal.center.B.cell.signature
0.16
0.18
Lymph.node.signature
0.32
0.15
Proliferation.signature
0.14
0.13
BMP6
MHC.class.II.signature

histogram

```
par(mfrow=c(2,2))
hist(dat[,7], 50, main = names(dat)[7], xlab="average signature")
hist(dat[,8], 50, main = names(dat)[8], xlab="average signature")
hist(dat[,9], 50, main = names(dat)[9], xlab="average signature")
hist(dat[,10], 50, main = names(dat)[10], xlab="average signature")
```



Summary

- Plotting should be the first step in any statistical analysis!
- Extremely Important
- Good modeling starts and ends with plotting
- R provides a great framework for plotting