

Introduction to Geostatistics

3. Descriptive graphs, and statistics

Edzer Pebesma

`edzer.pebesma@uni-muenster.de`

Institute for Geoinformatics (ifgi)
University of Münster

April 27, 2010



Why graphics are important

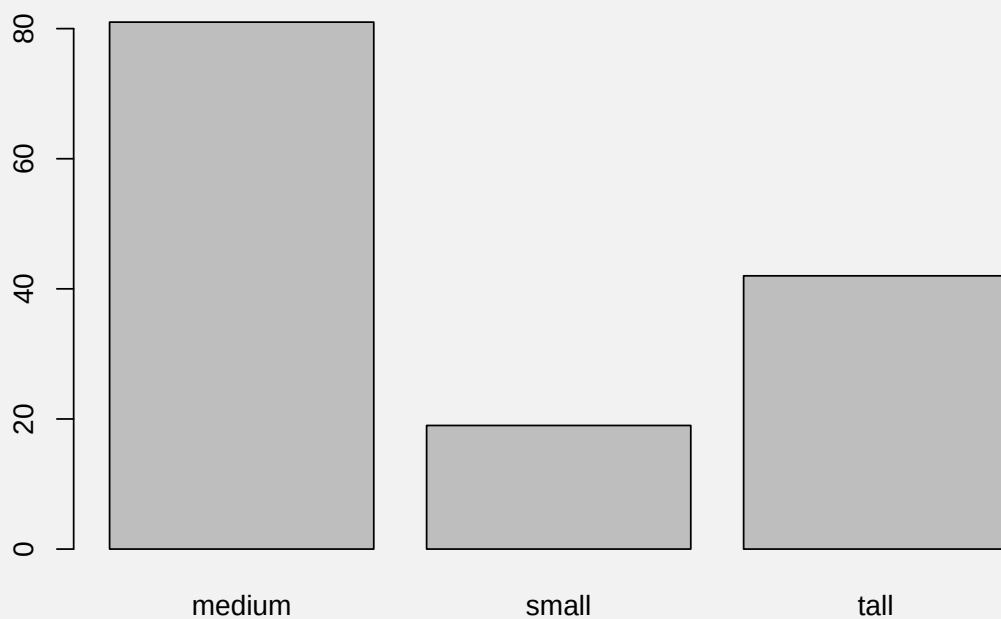
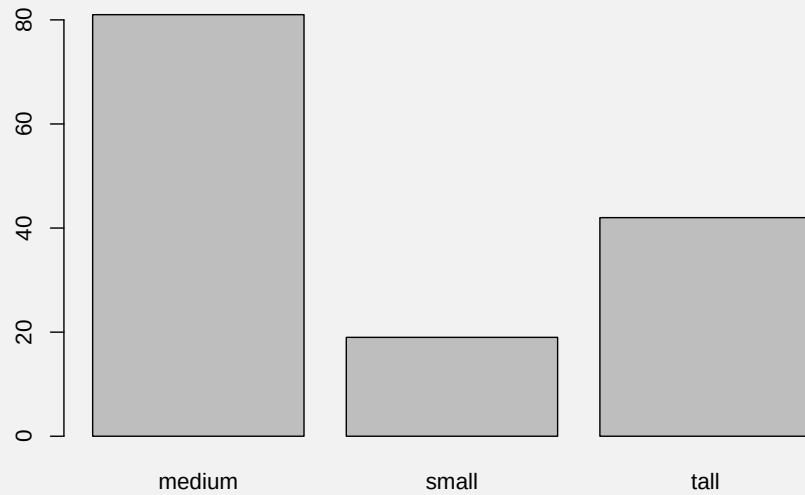
- ▶ a picture is worth more than a thousand words
- ▶ the other brain half
- ▶ dense information content, but the reader determines how much is read
- ▶ graphics need annotation (speech, written text), about
 - ▶ what it is about
 - ▶ what the the essential message is



Univariate graphs: bar chart

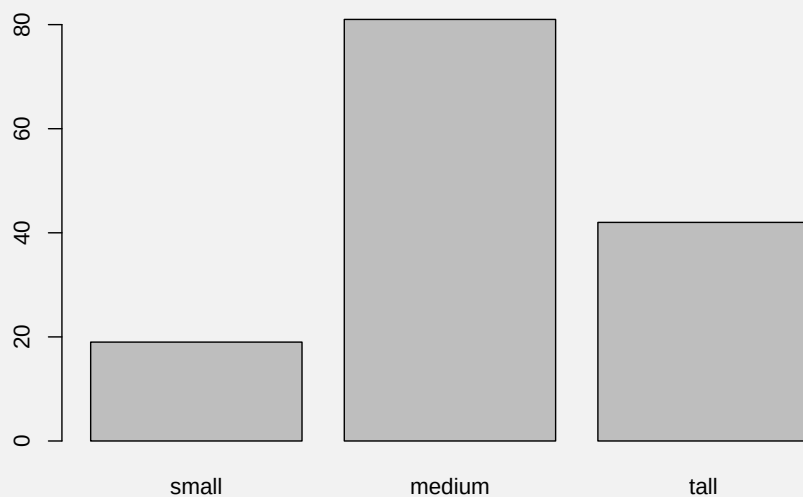
Discrete (nominal, ordinal) data:

```
> load("students2010.rdata")  
> attach(students)  
> IAm1 = factor(as.character(IAm))  
> plot(IAm1)
```



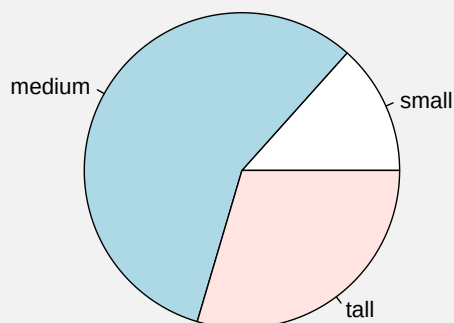
Univariate graphs: bar chart 2

```
> IAm2 = as.character(IAm1)
> IAm2 = factor(IAm2, levels = c("small", "medium", "tall"))
> plot(IAm2)
```



Univariate graphs: pie chart

```
> pie(table(IAm2))
```

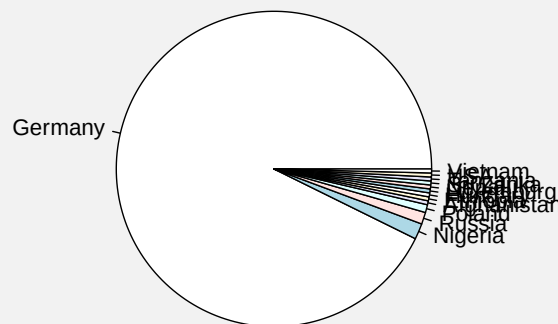


Note that medium is more than small + tall

Univariate graphs: pie chart

Pie charts do not always work well.

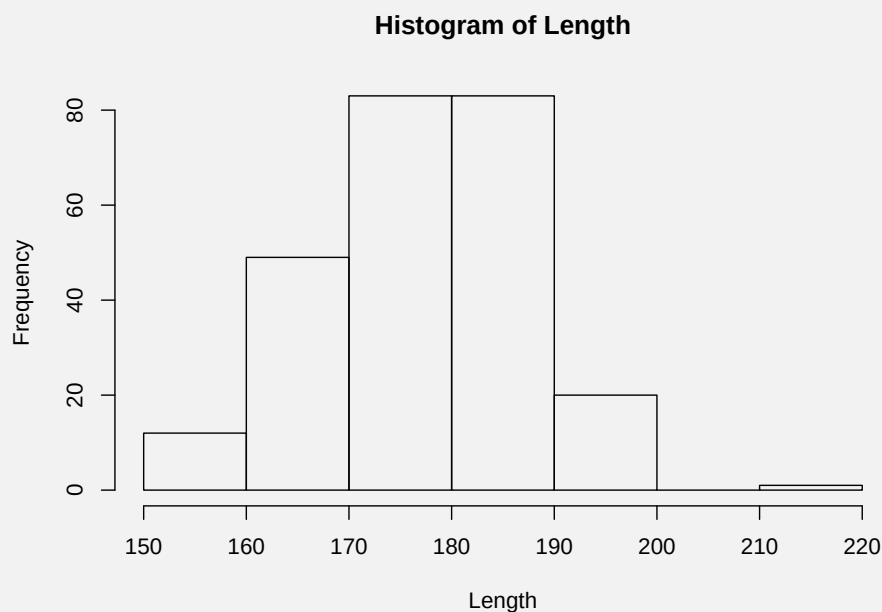
```
> pie(table(CountryOfBirth))
```



Univariate graphs: histogram

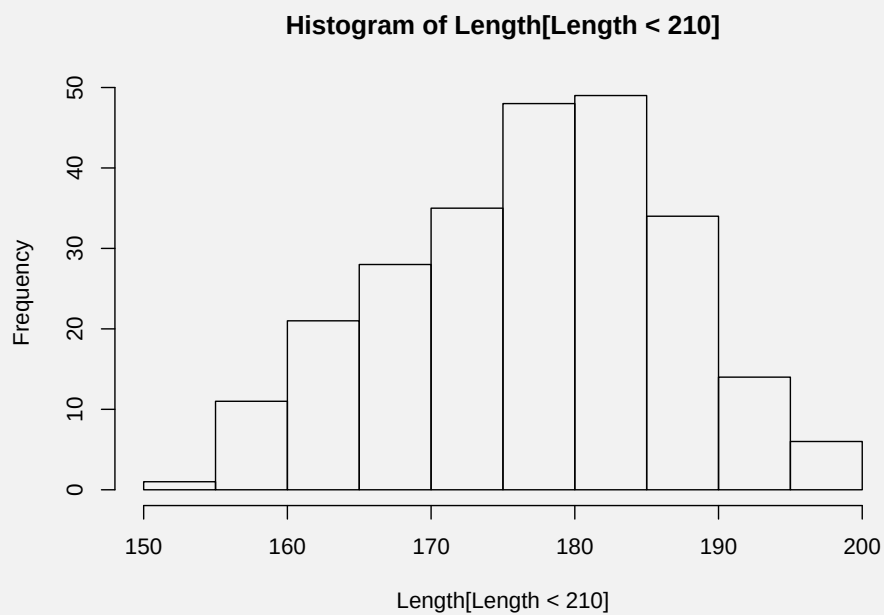
Histograms show the *distribution* of continuous data:

```
> hist(Length)
```



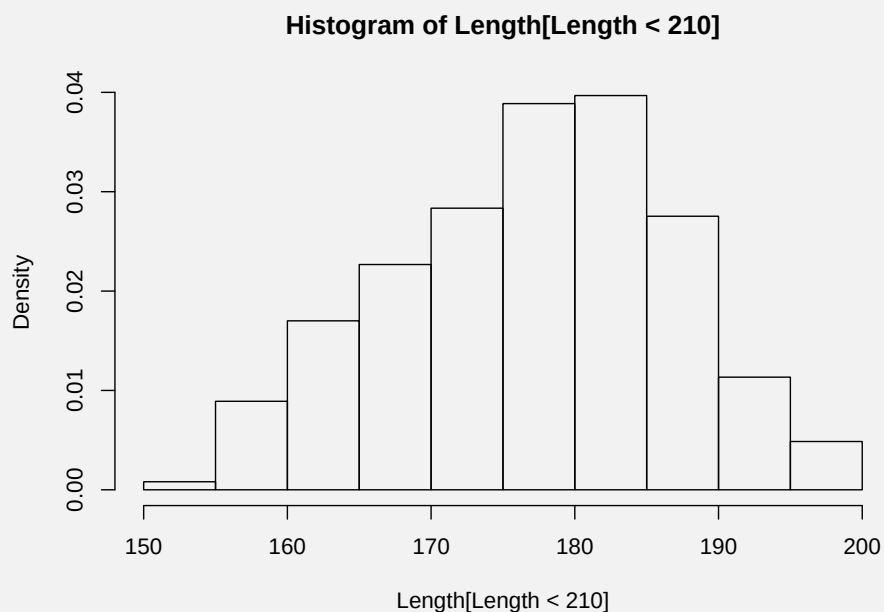
Univariate graphs: annotated histogram

```
> hist(Length[Length < 210])
```



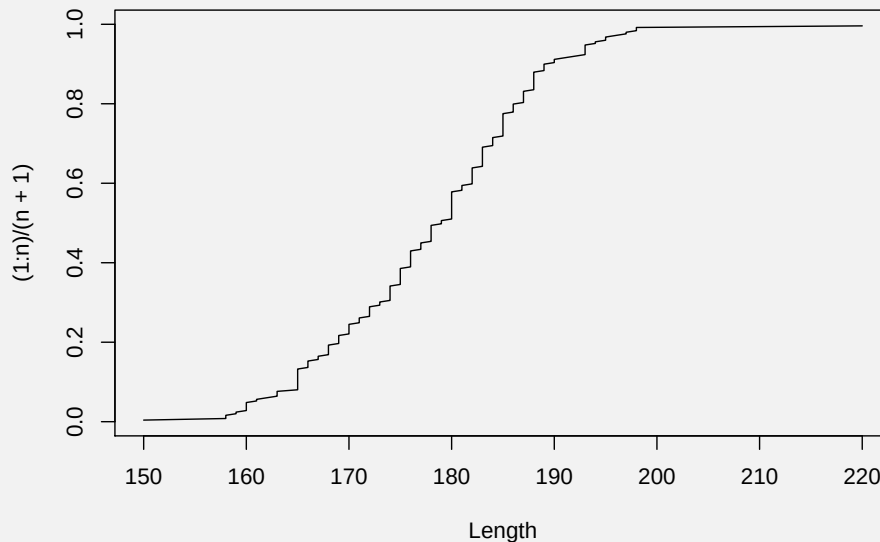
Histogram with fractions

```
> hist(Length[Length < 210], freq = FALSE)
```



Cumulative frequency curve

```
> n = length(Length)
> qqplot(Length, (1:n)/(n + 1), type = "l")
```

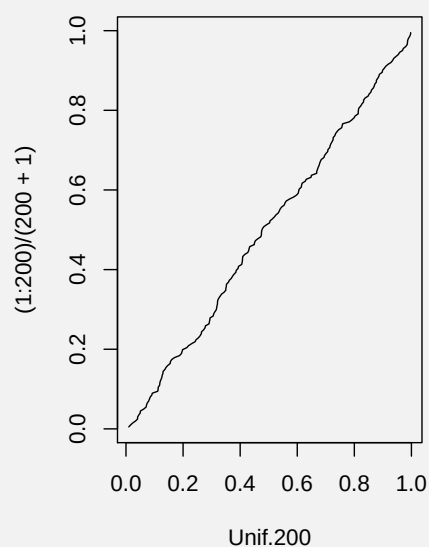
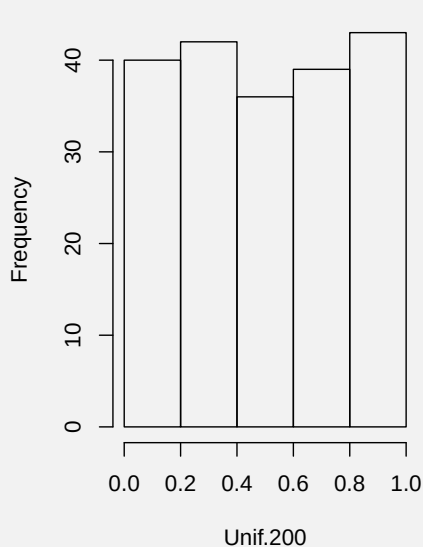


Cumulative frequency curve

When does this curve plot a straight line?

```
> Unif.200 = runif(200)
> n = length(Unif.200)
> par(mfrow = c(1, 2))
> hist(Unif.200, 5)
> qqplot(Unif.200, (1:200)/(200 + 1), type = "l")
```

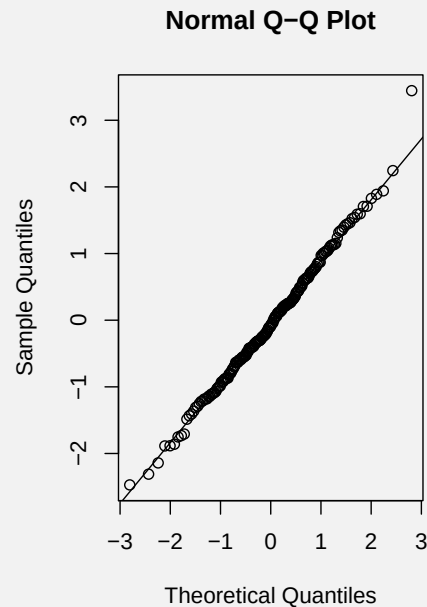
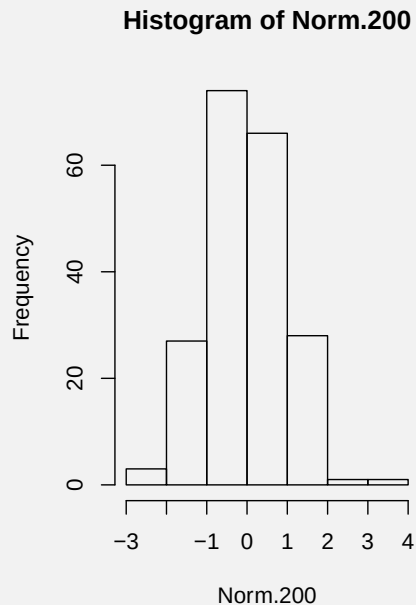
Histogram of Unif.200



Normal probability plots: 1

When does the normal curve plot a straight line?

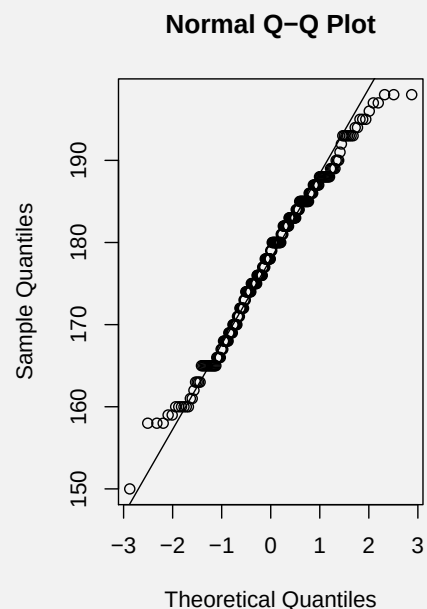
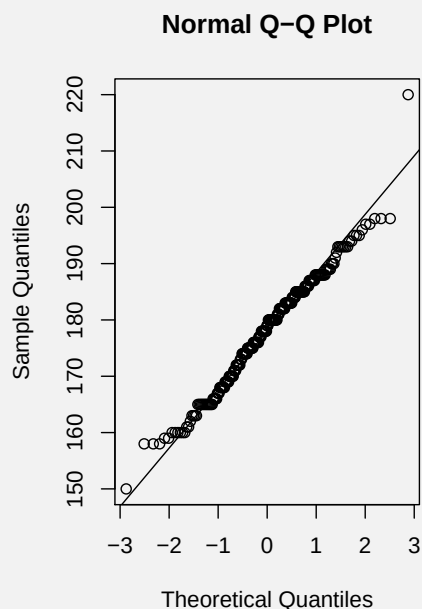
```
> Norm.200 = rnorm(200)
> par(mfrow = c(1, 2))
> hist(Norm.200, 5)
> qqnorm(Norm.200)
> qqline(Norm.200)
```



Normal probability plots: 2

Does **Length** follow a normal distribution?

```
> par(mfrow = c(1, 2))
> qqnorm(Length)
> qqline(Length)
> qqnorm(Length[Length < 210])
> qqline(Length[Length < 210])
```



Intermezzo: S formula's

```
> Weight ~ Length
```

```
Weight ~ Length
```

```
> class(Weight ~ Length)
```

```
[1] "formula"
```

Formula's are a language construct that specify *dependency*:
 $y \sim x$ means y depends on x . The following are e.g. equivalent:

```
> plot(x, y)
```

```
> plot(y ~ x)
```

Note that `plot` is a *generic*. The actual plot instance called depends on the class of its first argument.

```
> methods(plot)
```

```
[1] plot.acf*          plot.data.frame*   plot.Date*
[4] plot.decomposed.ts* plot.default        plot.dendrogram*
[7] plot.density       plot.ecdf           plot.factor*
[10] plot.formula*      plot.hclust*        plot.histogram*
[13] plot.HoltWinters*  plot.isoreg*        plot.lm
[16] plot.medpolish*    plot.mlm            plot.POSIXct*
[19] plot.POSIXlt*      plot.ppr*           plot.prcomp*
[22] plot.princomp*     plot.profile.nls*   plot.spec
[25] plot.spec.coherency plot.spec.phase     plot.stepfun
[28] plot.stl*          plot.table*         plot.ts
```

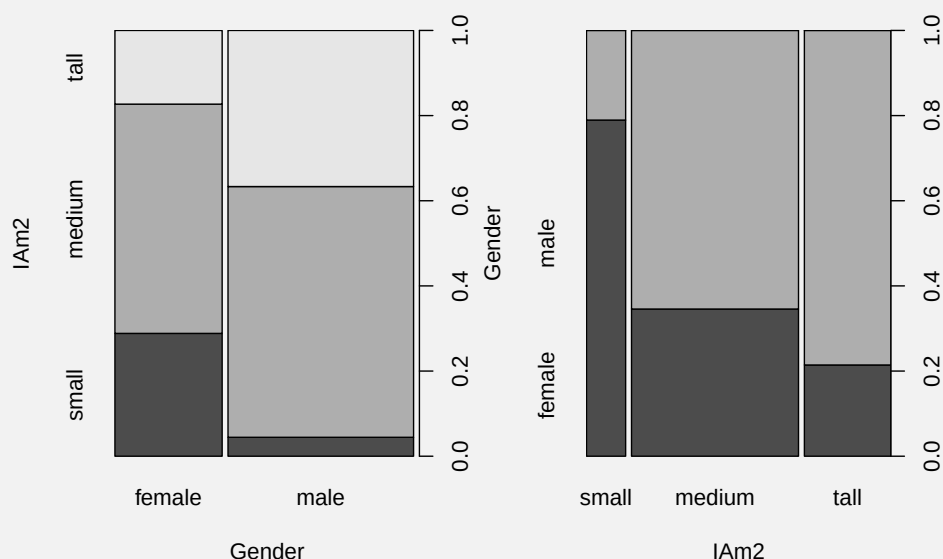


Bivariate graphs: 2 factors

```
> par(mfrow = c(1, 2))
```

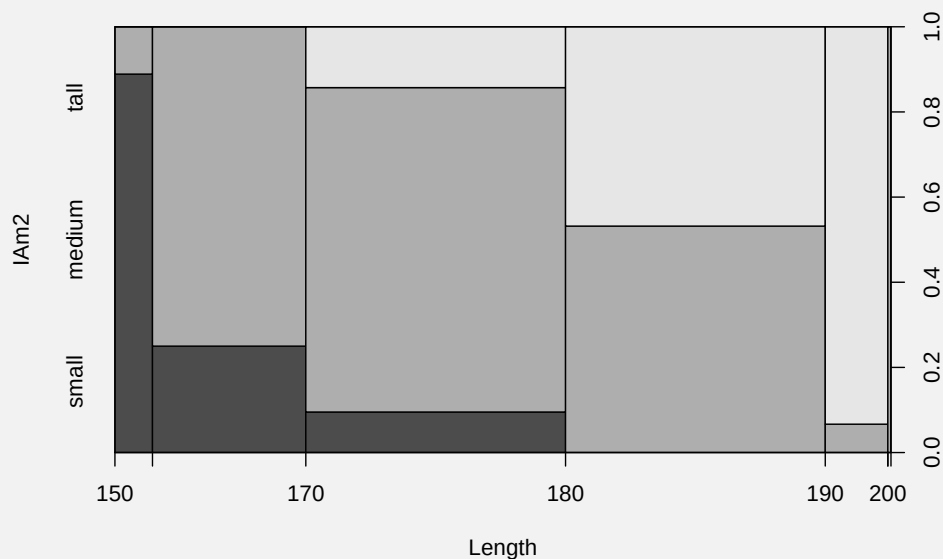
```
> plot(IAm2 ~ Gender)
```

```
> plot(Gender ~ IAm2)
```



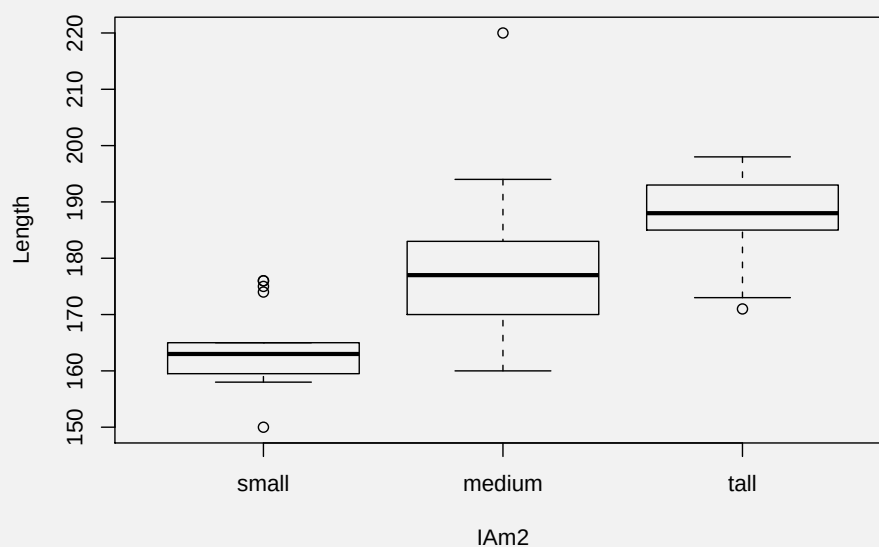
Bivariate graphs: x numerical, y factor

```
> plot(IAm2 ~ Length)
```



Bivariate graphs: x factor, y numerical

```
> plot(Length ~ IAm2)
```

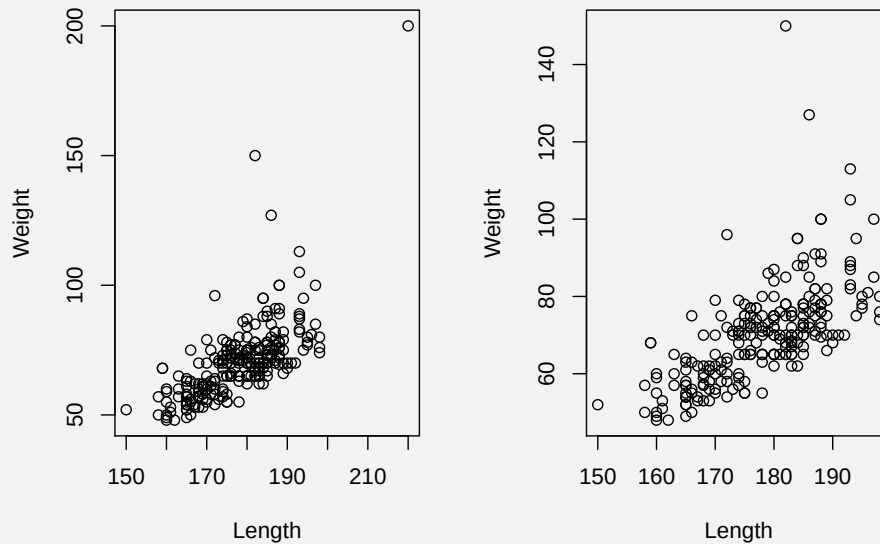


See also range in `?boxplot`

Bivariate graphs: scatter plot

Two numerical variables:

```
> par(mfrow = c(1, 2))  
> plot(Weight ~ Length)  
> plot(Weight ~ Length, students[Length < 200, ])
```

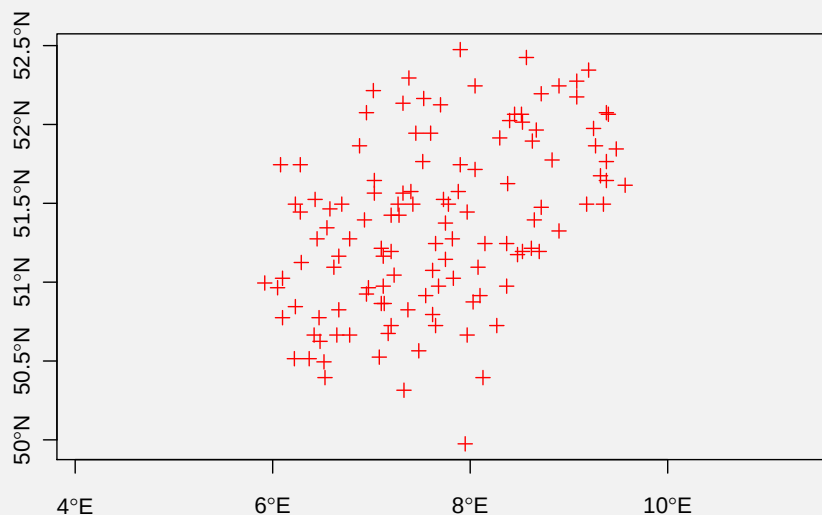


Bivariate graphs: geographical map

Two numerical variables: x- and y-coordinate. Features: points, lines, polygons, grids.

Axes have special properties, e.g. aspect ratio

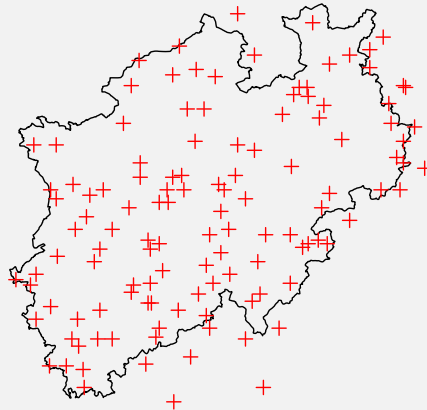
```
> library(sp)  
> load("katha/.RData")  
> plot(wetter.ll, axes = TRUE, col = "red")
```



Bivariate graphs: geographical map

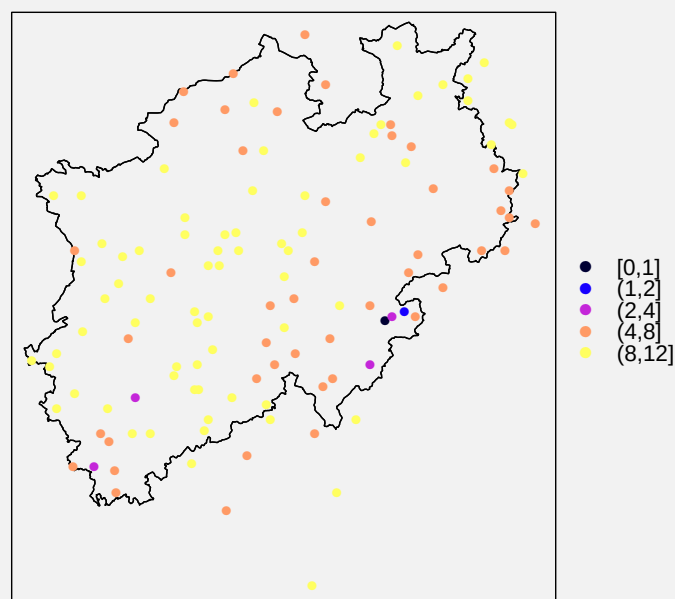
On maps, axes are often replaced by useful reference features

```
> library(sp)
> plot(NRW)
> plot(wetter.ll, add = TRUE, col = "red")
```



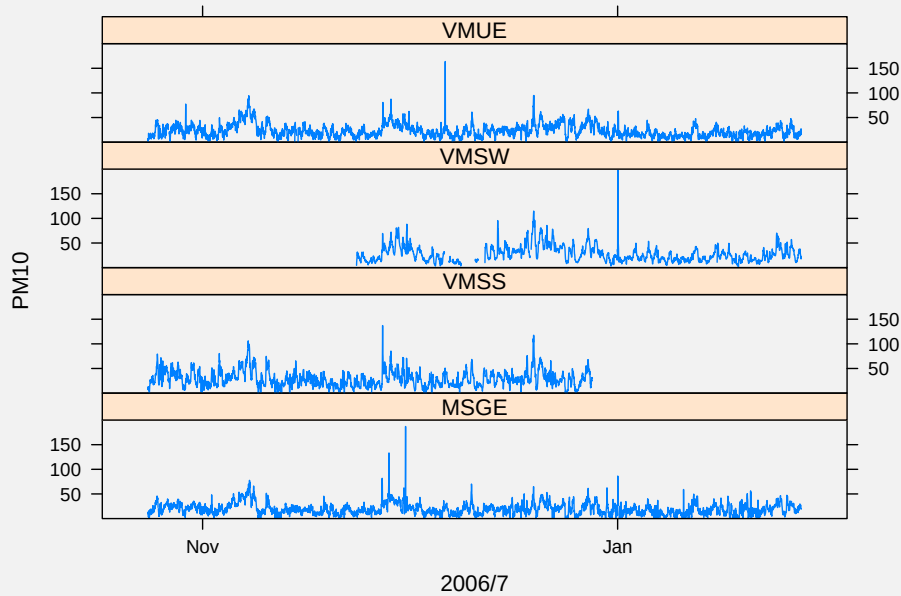
Trivariate graphs: maps with features

```
> library(sp)
> spplot(wetter.ll, "MESSWERT")
```



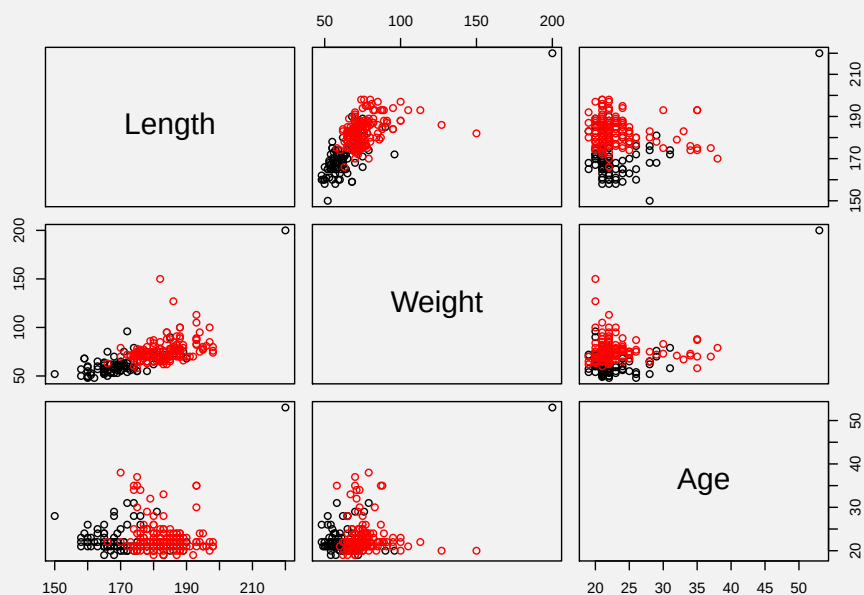
Time series graphs: PM10

```
> library(lattice)
> load("pm10.RData")
> xyplot(values ~ time | Station, pm10tso, type = "l",
+       layout = c(1, 4), xlab = "2006/7", ylab = "PM10",
+       ylim = c(0, 200))
```



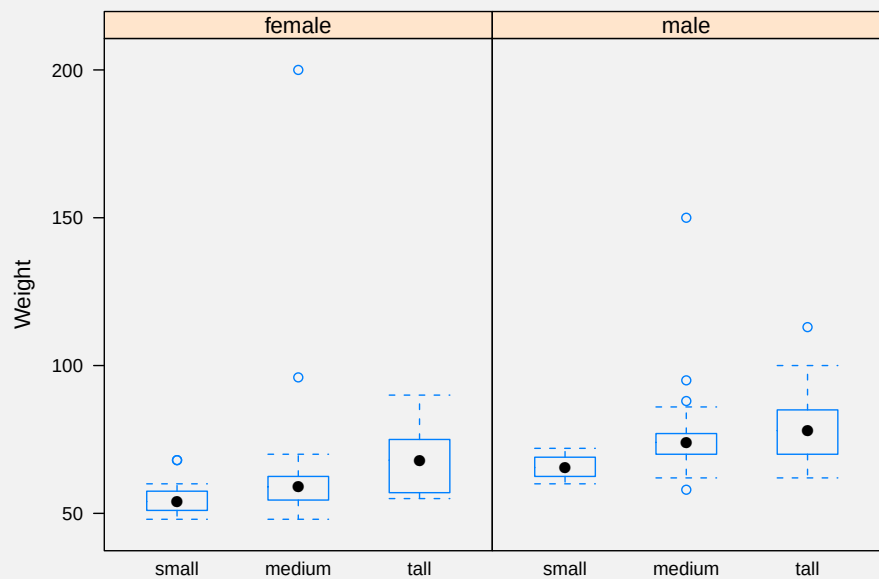
Scatter plot matrix: students

```
> plot(students[c("Length", "Weight", "Age")], col = as.numeric(students$Gender))
```



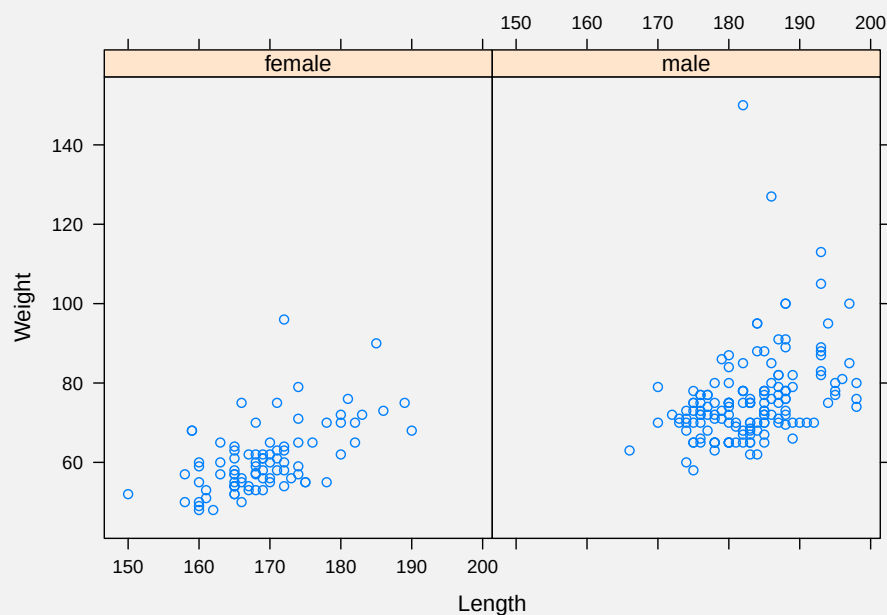
Trivariate graphs: 2. students

```
> library(lattice)
> bwplot(Weight ~ IAm2 | Gender, students)
```



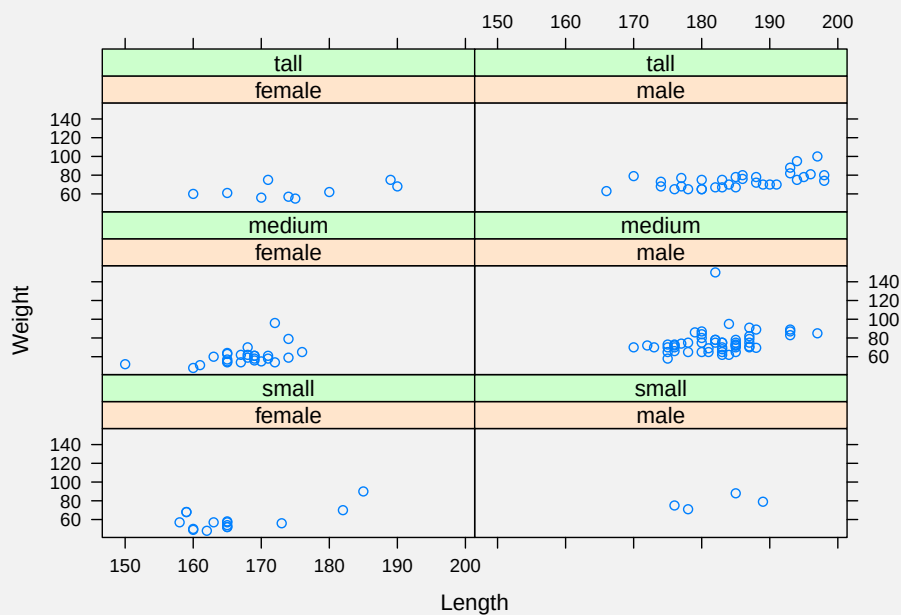
Trivariate graphs 3: students

```
> library(lattice)
> check = Length < 220
> xyplot(Weight ~ Length | Gender, students[check, ])
```



Multivariate graphs 2: students

```
> library(lattice)
> xyplot(Weight ~ Length | Gender + IAm2, students[check,
+      ])
```



Lattice xyplot examples



Why graphics are important

- ▶ a picture is worth more than a thousand words
- ▶ the other brain half
- ▶ dense information content, but the reader determines how much is read
- ▶ graphics need annotation (speech, written text), about
 - ▶ what it is about
 - ▶ what the the essential message is

