UNIVERSITÄT BERN

Statistical methods for archaeological data analysis I: Basic methods

03 - Explorative statistics & graphical display

Martin Hinz

Institut für Archäologische Wissenschaften, Universität Bern

04.03.2025

You can download a pdf of this presentation.

Loading data for the following steps

download data

- muensingen_fib.csv

Read the Data on Muensingen Fibulae

```
muensingen <- read.csv2("https://raw.githubusercontent.com/BernCoDALab/smada/refs/
head(muensingen)
```

```
##    X Grave Mno FL BH BFA FA CD BRA ED FEL  C   BW  BT FEW Coils Length
## 1  1   121 348 28 17   1 10 10   2  8   6 20  2.5 2.6 2.2     4     53
## 2  2   130 545 29 15   3  8  6   3  6  10 17 11.7 3.9 6.4     6     47
## 3  3   130 549 22 15   3  8  7   3 13   1 17  5.0 4.6 2.5    10     47
## 4  8   157  85 23 13   3  8  6   2 10   7 15  5.2 2.7 5.4    12     41
## 5 11   181 212 94 15   7 10 12   5 11  31 50  4.3 4.3  NA     6    128
## 6 12   193 611 68 18   7  9  9   7  3  50 18  9.3 6.5  NA     4    110
##   fibula_scheme
## 1             B
## 2             B
## 3             B
## 4             B
## 5             C
## 6             C
```

Cross tables (contingency tables)

For summary of data:

```
my_table <- table(muensingen$fibula_scheme, muensingen$Grave)
my_table
```

```
##
##      6 23 31 44 48 49 61 68 80 91 121 130 157 181 193
##    A 1  1  1  1  0  0  0  0  0  0    0    0    0    0    0
##    B 0  0  0  0  1  1  2  1  1  1    1    2    1    0    0
##    C 0  0  0  0  0  0  0  0  0  0    0    0    0    1    1
```

```
addmargins(my_table)
```

```
##
##         6 23 31 44 48 49 61 68 80 91 121 130 157 181 193 Sum
##    A    1  1  1  1  0  0  0  0  0  0    0    0    0    0    0   4
##    B    0  0  0  0  1  1  2  1  1  1    1    2    1    0    0  11
##    C    0  0  0  0  0  0  0  0  0  0    0    0    0    1    1   2
##    Sum  1  1  1  1  1  1  2  1  1  1    1    2    1    1    1  17
```

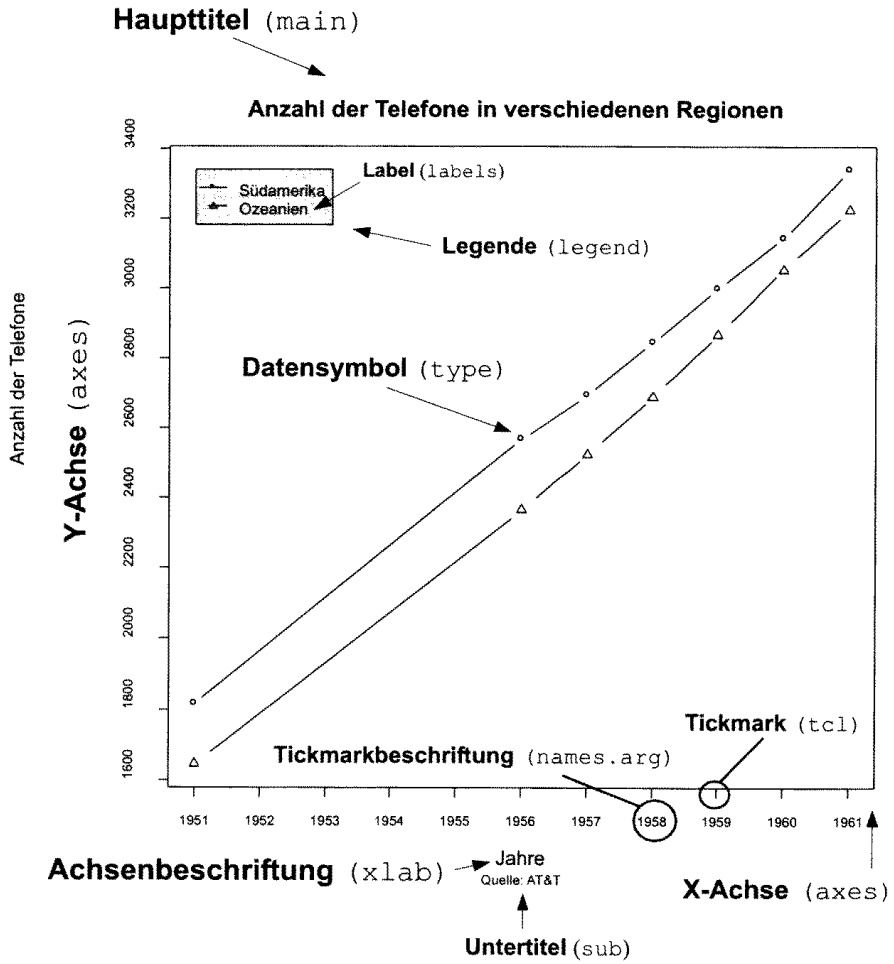
Basics about charts

Principles for good charts according to E. Tufte:

(The Visual Display of Quantitative Information. Cheshire/ Connecticut: Graphics Press, 1983)

- „Graphical exellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.”
- Data-ink ratio = „proportion of a graphic’s ink devoted to the non-redundant display of data-information“ (kein chartjunk!)
- „Graphical excellence is often found in simplicity of design and complexity of data. “

- after Müller-Scheeßel



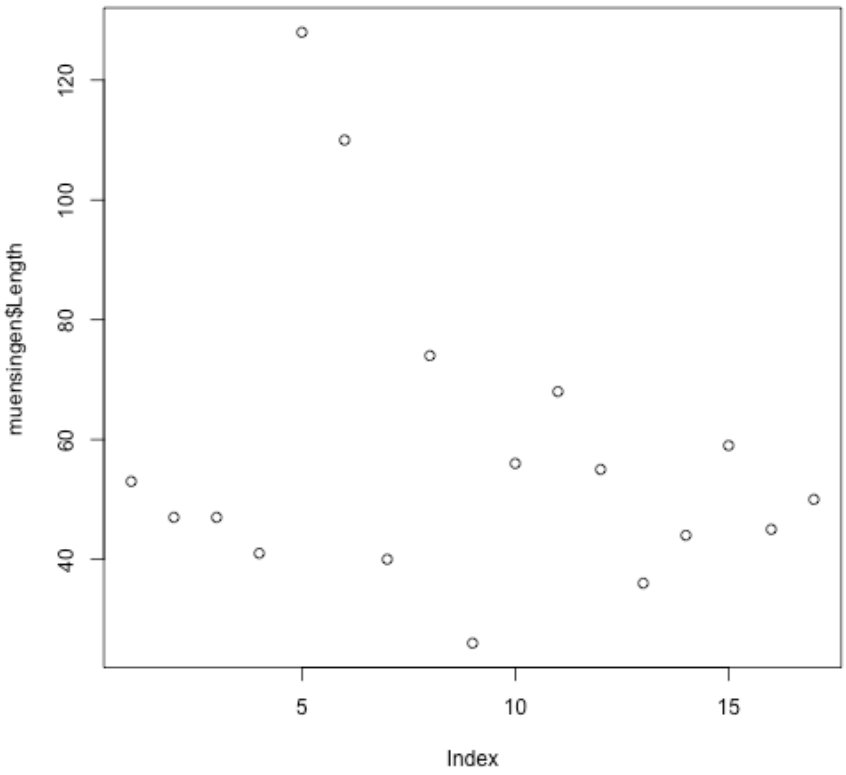
Plot [1]

Basic drawing function of R:

```
plot(muensingen$Length)
```

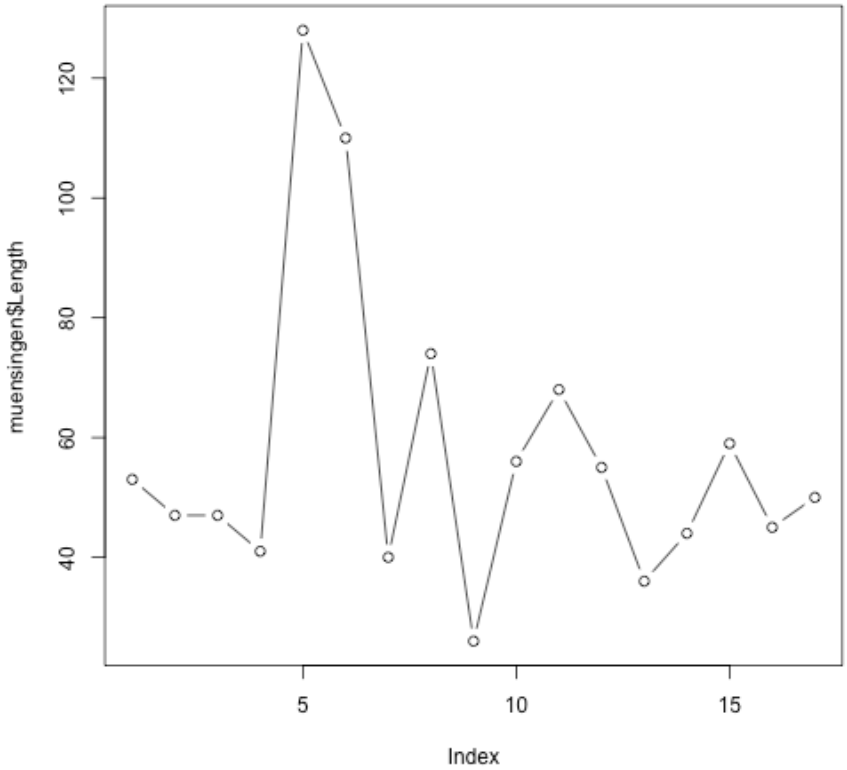
options:

- p – points (default)
- l – solid line
- b – line with points for the values
- c – line with gaps for the values
- o – solid line with points for the values
- h – vertical lines up to the values
- s – stepped line from value to value
- n – empty coordinate system



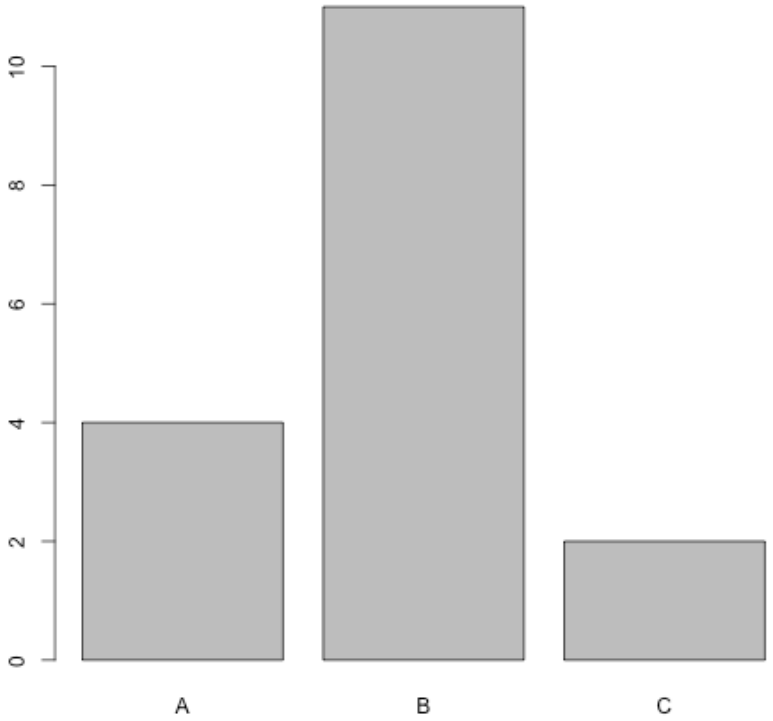
Plot [2]

```
plot(muensingen$Length,type="b")
```



Intelligent system: automatic determination of variable type, drawing of the appropriate chart

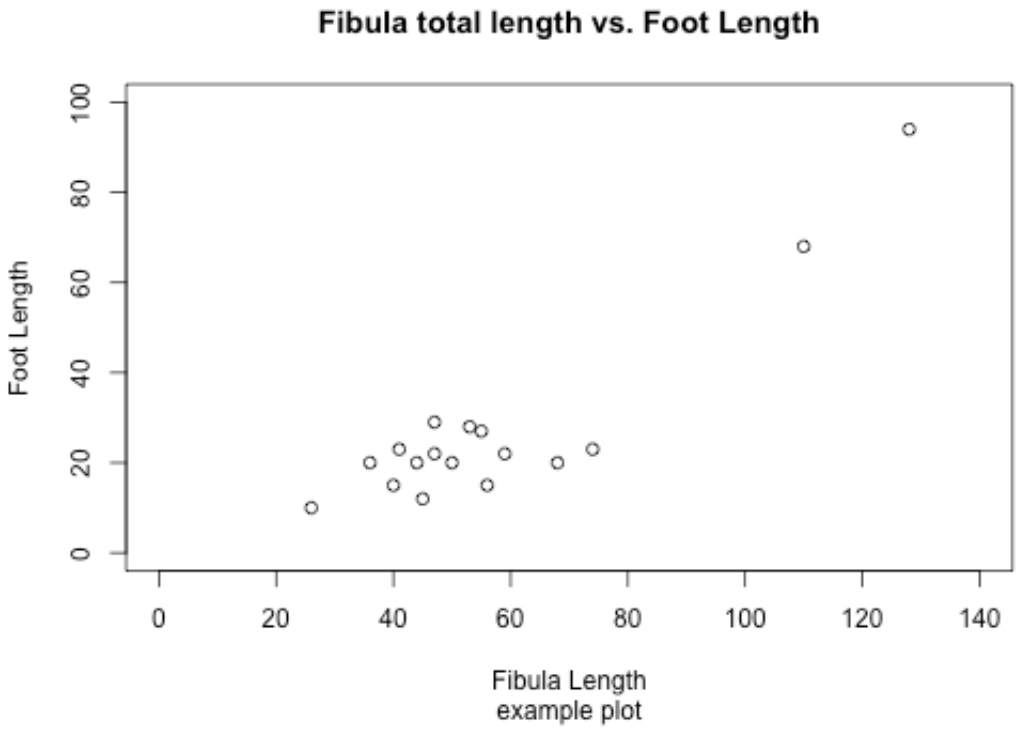
```
plot(as.factor(muensingen$fibula_scheme))
```



Plot [3]

Enhancing the plot with optional components & Text

```
plot(muensingen$Length, muensingen$FL,
     xlim=c(0, 140), # limits of the x axis
     ylim = c(0, 100), # limits of the y axis
     xlab = "Fibula Length", # label of the y axis
     ylab = "Foot Length", # label of the x axis
     main = "Fibula total length vs. Foot Length", # main title
     sub="example plot" # subtitle
     )
```



Plot [4]

Plot do a lot for you:

- Opens a window for display
- Determines the optimal size of the frame of reference
- Draws the coordinate system
- Draws the values

Gives a „handle“ back for further additions to the plot, e.g.:

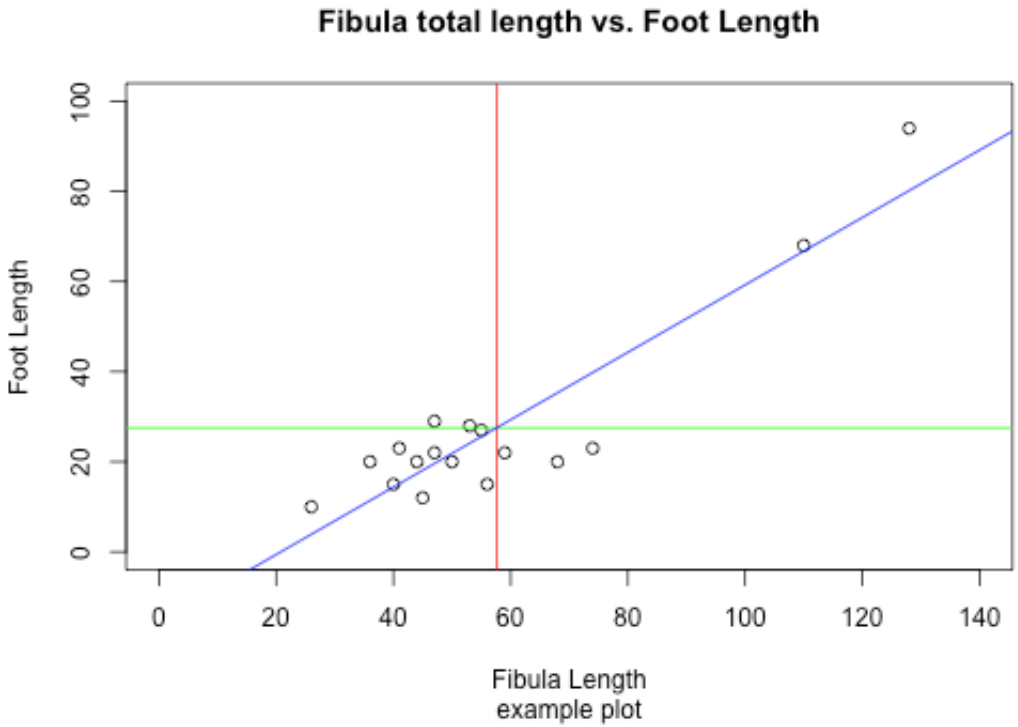
- lines – additional lines to an existing plot
- points – additional points to an existing plot
- abline – additional special lines to an existing plot
- text – additional text on choosen position to an existing plot

Additional possiblities for “decorations”: ? par

Plot [5]

Add additional elements: Drawing lines

```
abline(v = mean(muensingen$Length), col = "red")          # draw a red vertical lin
abline(h = mean(muensingen$FL), col = "green")            # draw a green vertical l
abline(lm(FL~Length, data = muensingen), col = "blue")    # draw a blue diagonal li
```



Export the graphics

With the GUI:

Export → Save as...

With the commando line: As vector file

```
dev.copy2eps(file="test.eps")
dev.copy2pdf(file="test.pdf")
```

```
savePlot(filename="test.tif", type="tiff")
```

Possible are “png”, “jpeg”, “tiff”, “bmp” SavePlot can save sometimes also vector files (dependent on operation system and installation)

Pie chart [1]

The classical one – but also with R not much better...

Used to display proportions, suitable for nominal data

$$a_i = \frac{n_i}{N} * 360^\circ$$

Disadvantages:

- Color selection can influence the perception (red is seen larger then gray)
- Small differences are not easy visible

totally No-Go: 3d-pies!!!

Pie chart [2]

I eat pie...

The pieces »viel zu wenig«, »etwas zu wenig« und »gerade richtig« have exactly the same size, the piece »viel zu viel« is a bit smaller.

source: http://www.lrz-muenchen.de/~wlm

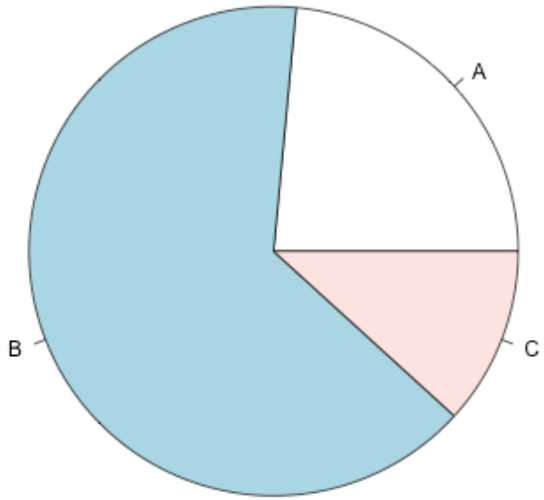
Pie chart [3]

Data are a vector of counts

```
table(muensingen$fibula_scheme)
```

```
##
##  A  B  C
##  4 11  2
```

```
pie(table(muensingen$fibula_scheme))
```



Color palette:

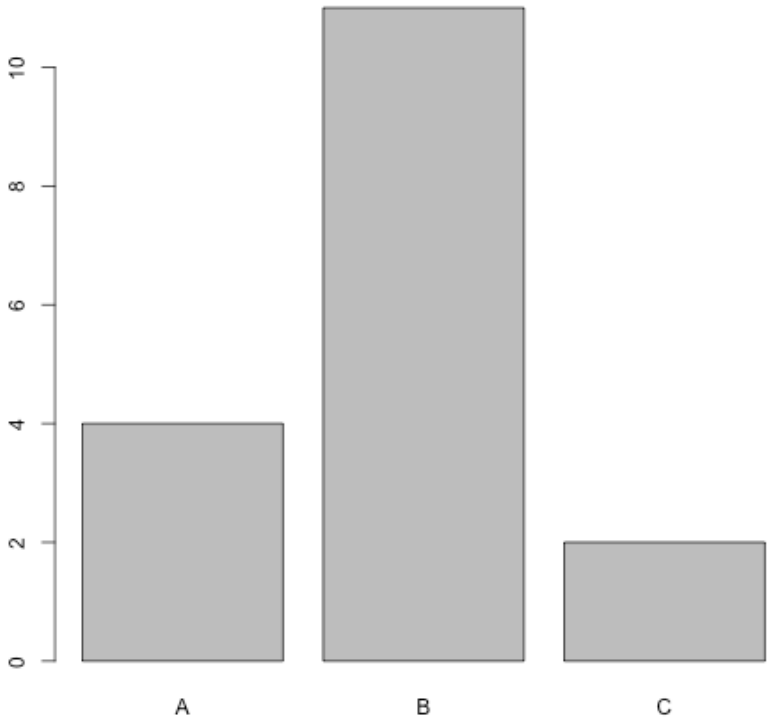
The standard palette is pastel, if you prefer another:

```
pie(table(muensingen$fibula_scheme),
    col=c("red","green","blue"))
```

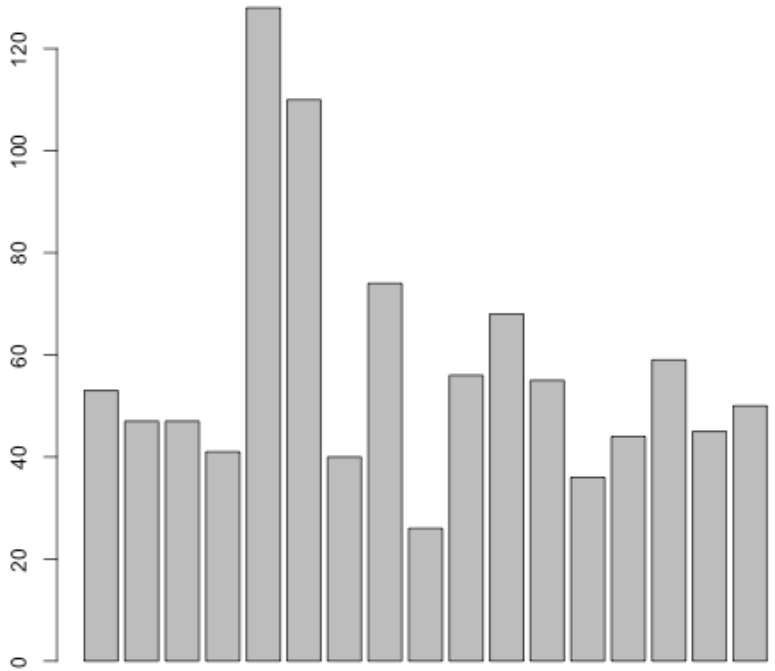
Bar plot [1]

Generally the better alternative... Bar plots are suitable for display of proportions as well as for absolute data. They can be used for every level of measurement.

```
barplot(table(muensingen$fibula_scheme))
```



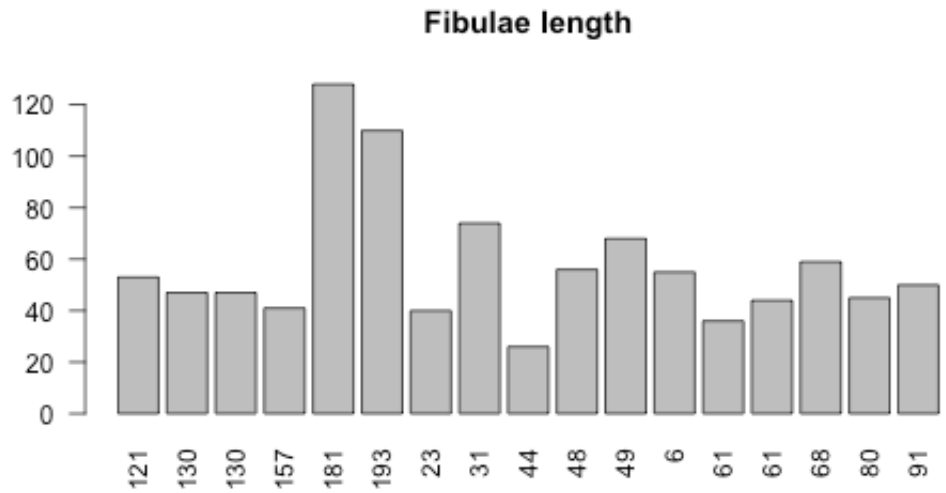

```
barplot(muensingen$Length)
```



Bar plot [2]

With names:

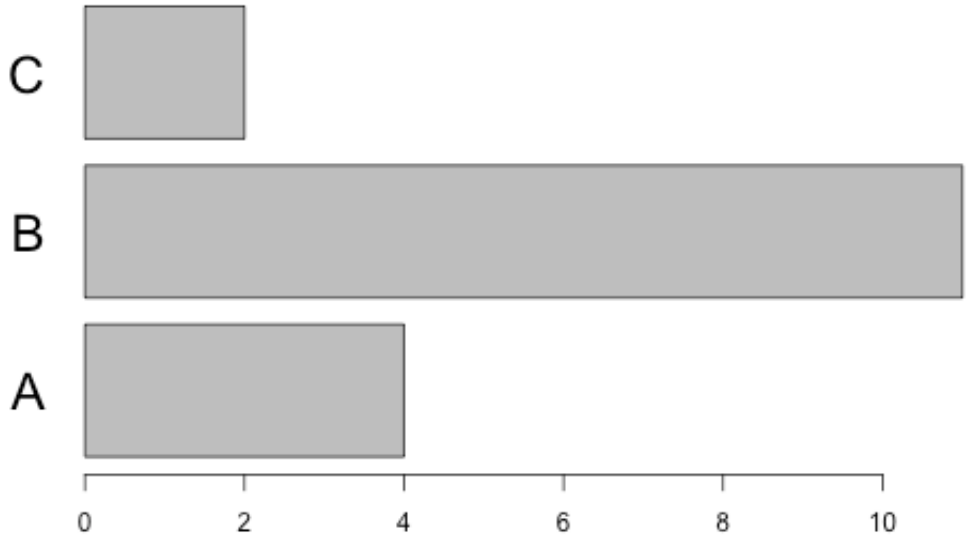
```
par(las=2)                              # turn labels 90°
barplot(muensingen$Length,              # plot fibulae length
        names.arg=muensingen$Grave)     # with names of the graves
title("Fibulae length")                 # add title
```



Bar plot [3]

Horizontal:

```
par(las=1)                                # turn labels back again
barplot(table(muensingen$fibula_scheme), # Plot counts fibulae scheme
        horiz=T,                          # horizontal
        cex.names=2)                      # make the labels bigger
```



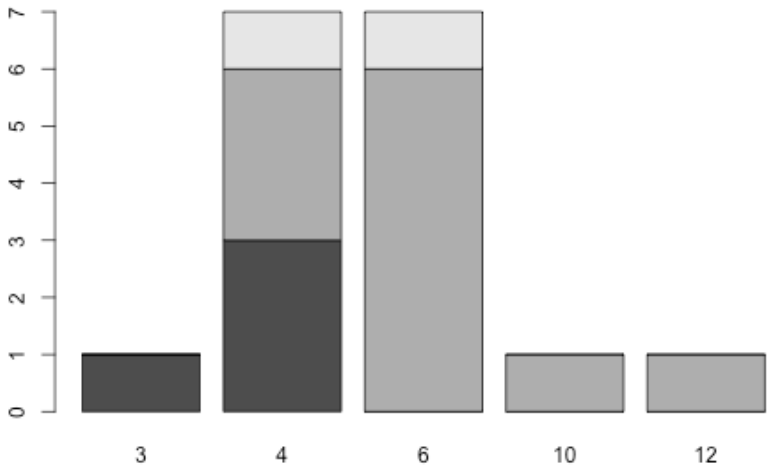
Bar plot [4]

Display of counts

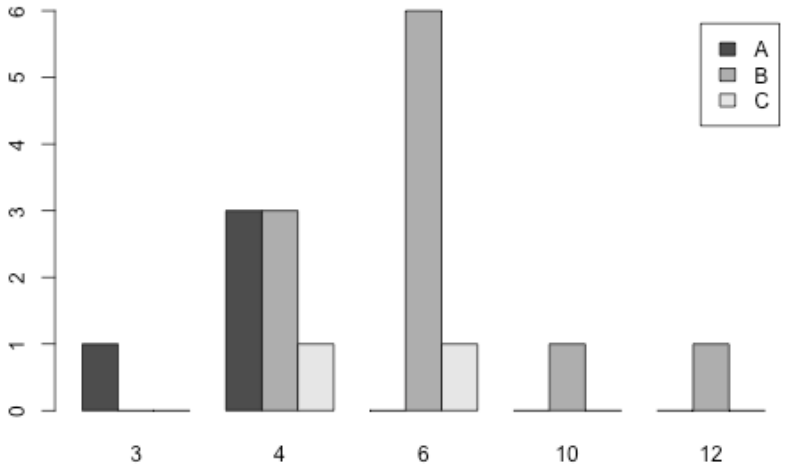
```
my_new_table <- table(muensingen$fibula_scheme,
                      muensingen$Coils)
my_new_table
```

```
##
##      3 4 6 10 12
##   A 1 3 0  0  0
##   B 0 3 6  1  1
##   C 0 1 1  0  0
```

```
barplot(my_new_table)
```




```
barplot(my_new_table, beside=T, legend.text=T)
```



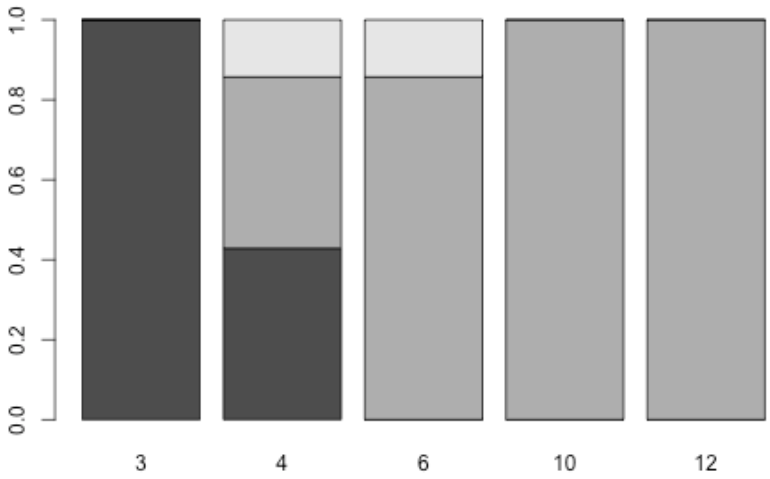
Bar Plot [5]

Display of proportions

```
table.prop<-prop.table(my_new_table,2)
table.prop
```

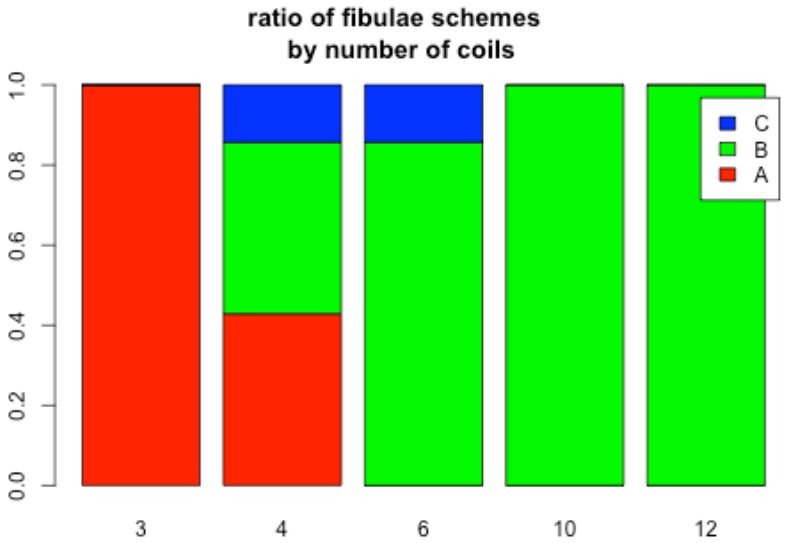
```
##
##              3         4         6        10        12
##    A 1.0000000 0.4285714 0.0000000 0.0000000 0.0000000
##    B 0.0000000 0.4285714 0.8571429 1.0000000 1.0000000
##    C 0.0000000 0.1428571 0.1428571 0.0000000 0.0000000
```

```
barplot(table.prop)
```




```
tmp<-barplot(table.prop,
             legend.text=T,   # add a legend
             col=rainbow(3)   # make it more colorful
             )

# add a title
title("ratio of fibulae schemes \n by number of coils",
      outer=TRUE,            # outside the plot area
      line=- 3)              # on line -3 above
```

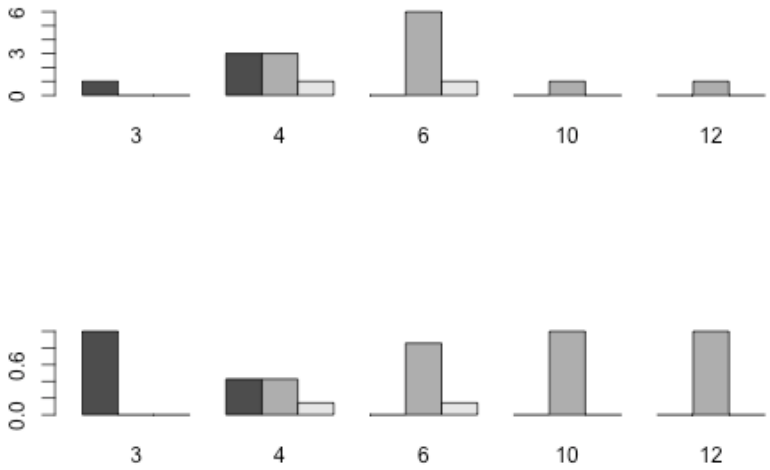


Bar Plot [6]

Problems with bar plots – and also with many other charts

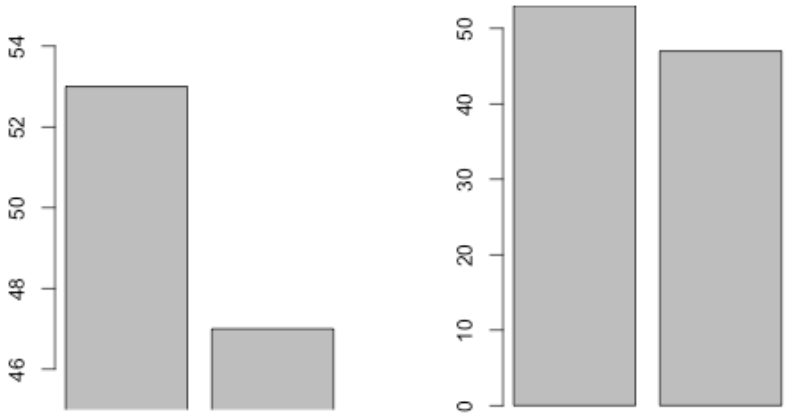
Percent vs. count: percents often distort the relations

```
par(mfrow=c(2,1))
barplot(my_new_table,beside=T)
barplot(table.prop,beside=T)
```



Scales: the choosen limits of the axes can distort the relations

```
par(mfrow=c(1,2))
barplot(muensingen$Length[1:2],xpd=F,ylim=c(45,55))
barplot(muensingen$Length[1:2],xpd=F)
```




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

Box-plot (Box-and-Whiskers-Plot)

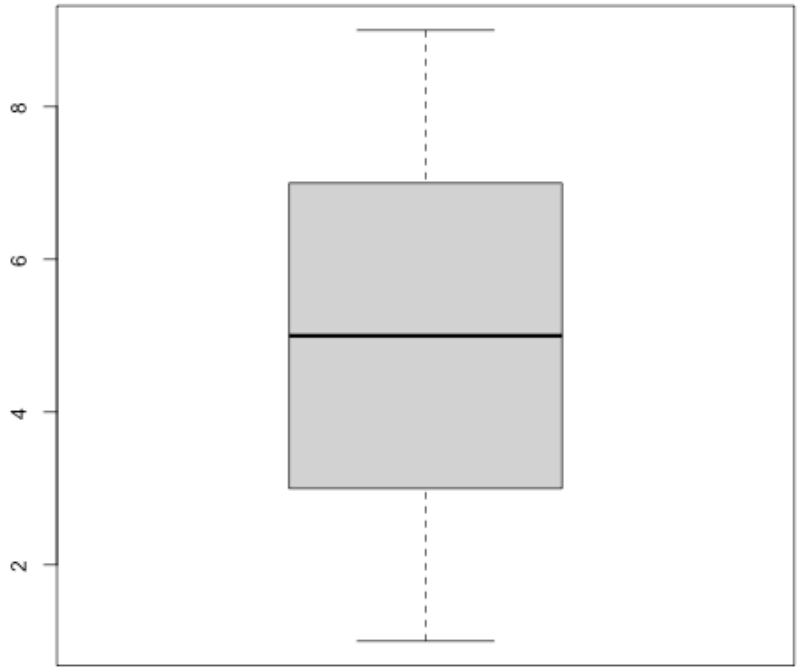
One of the best (my precious)!

Used to display the distribution of values in a data vector of metrical (interval, ratio) scale

```
1  2  3  4  5  6  7  8  9
___|___|___|___|___
```

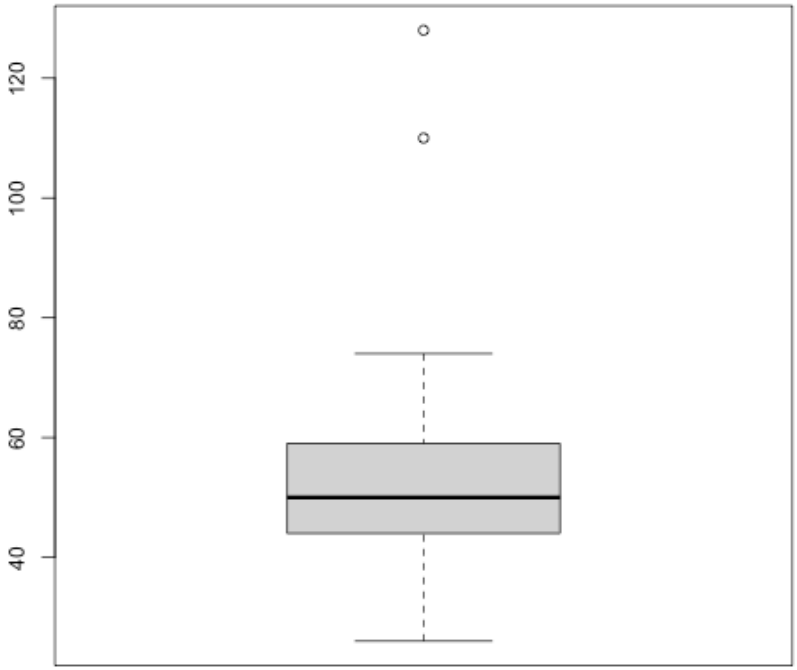
- thick line: mean
- Box: the inner both quantiles
- Whisker: last value < than 1.5 times the distance of the inner quantile

```
boxplot(1:9)
```

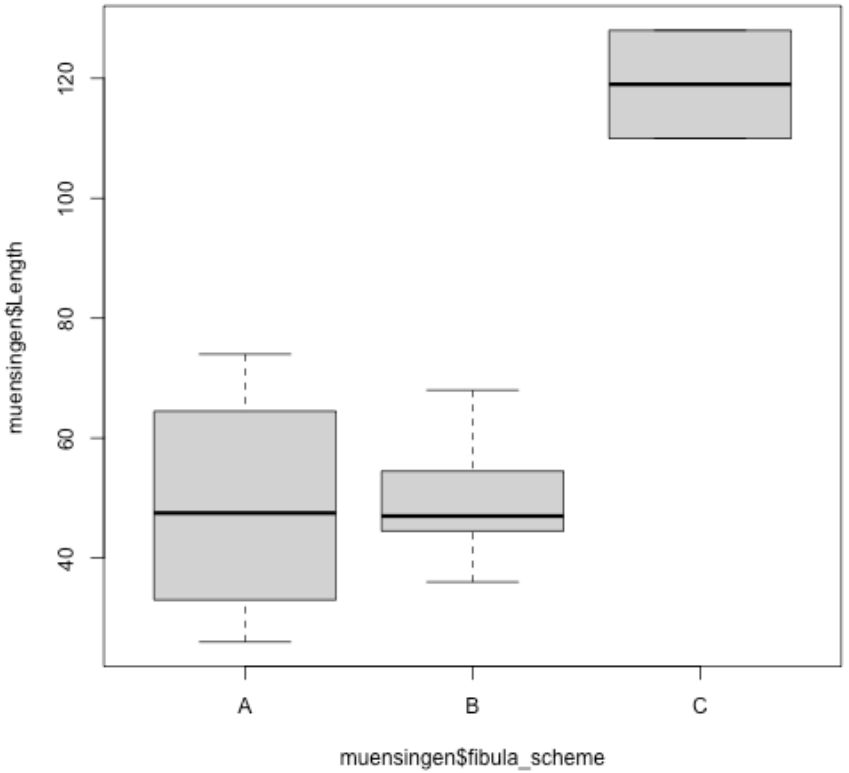


Box Plot [2]

```
boxplot(muensingen$Length)
```



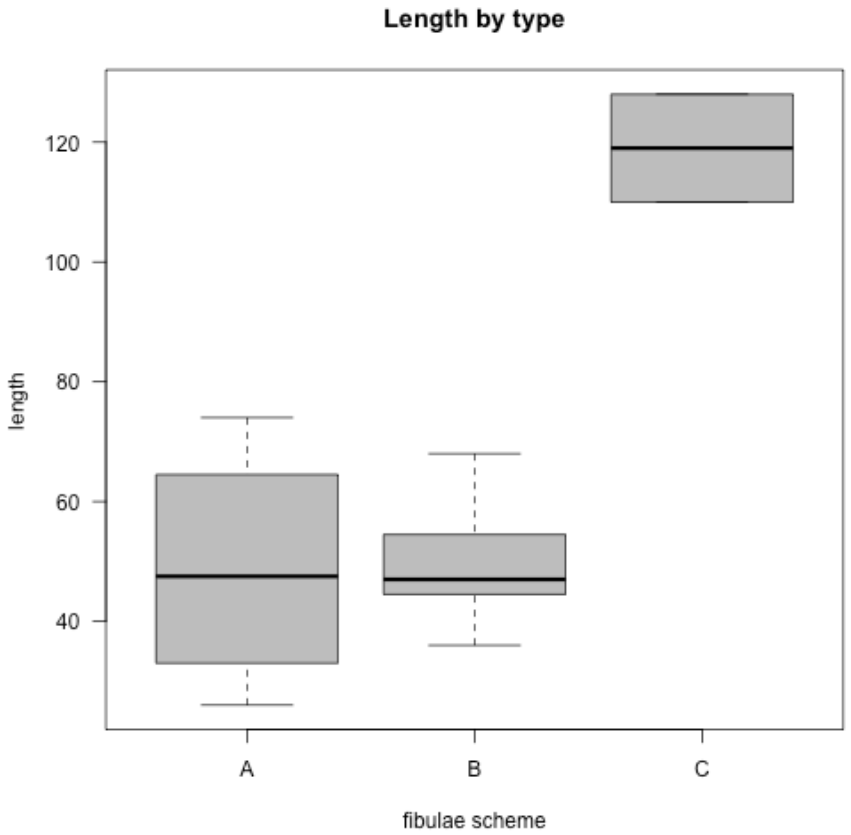

```
boxplot(muensingen$Length ~
          muensingen$fibula_scheme)
```



Box Plot [3]

More beautiful:

```
par(las=1)
boxplot(Length ~ fibula_scheme,
        data = muensingen,
        main = "Length by type",
        col="grey",
        xlab="fibulae scheme",
        ylab= "length"
        )
```

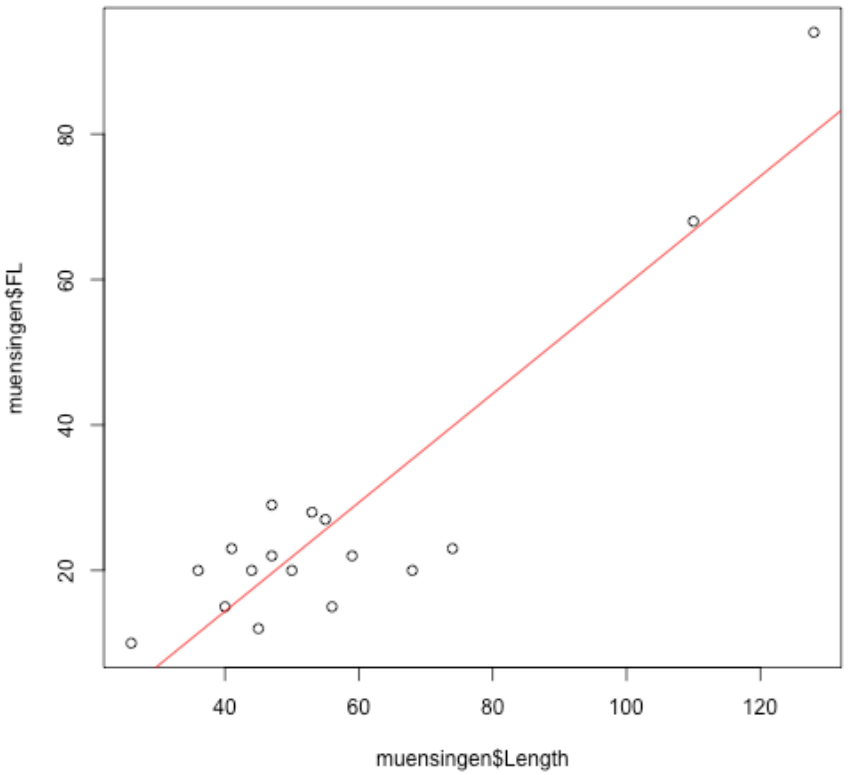


Scatterplot [1]

For 2 variables

Used to display a variable in relation to another one. Generally for all scales suitable, but for nominal and ordinal scale other charts are often better.

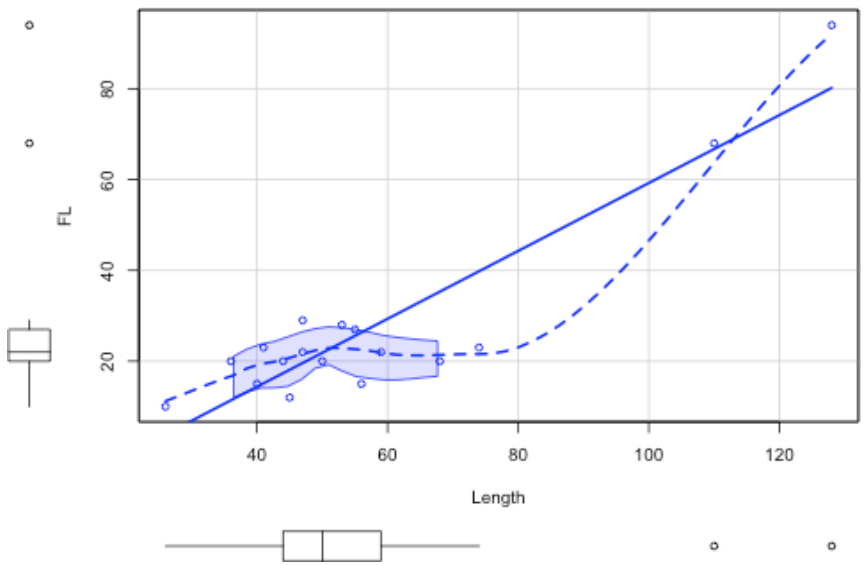
```
plot(muensingen$Length, muensingen$FL)
abline(
  lm(muensingen$FL~muensingen$Length),
  col="red")
```



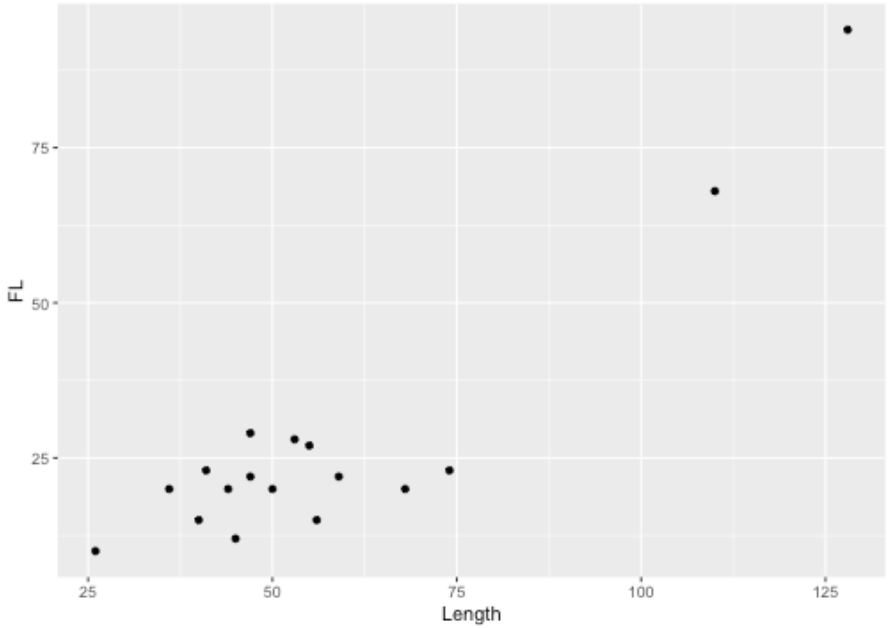
Scatterplot [2]

Call additional libraries:

```
library(car) # library for regression analysis
scatterplot(FL ~ Length, data = muensingen)
```



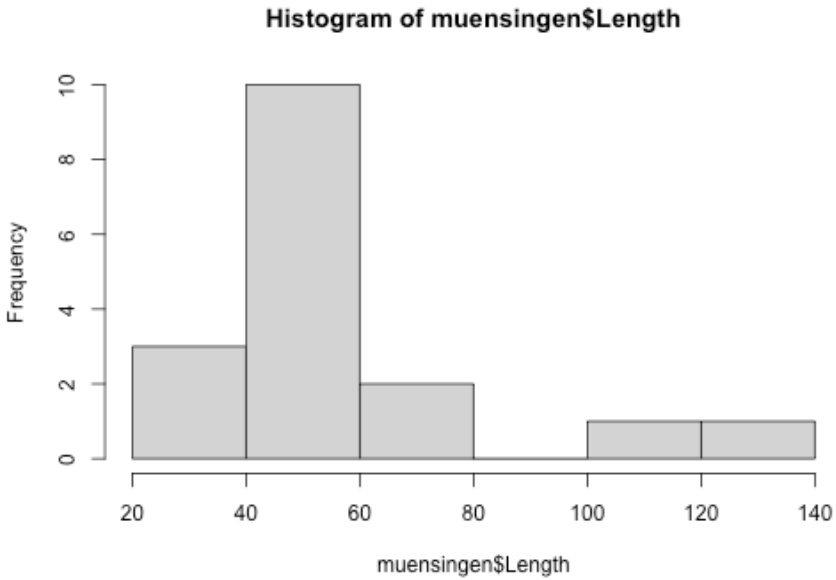

```
library(ggplot2) # advanced plots library
b<- ggplot(muensingen,aes(x=Length,y=FL))
graph<-b + geom_point()
show(graph)
```



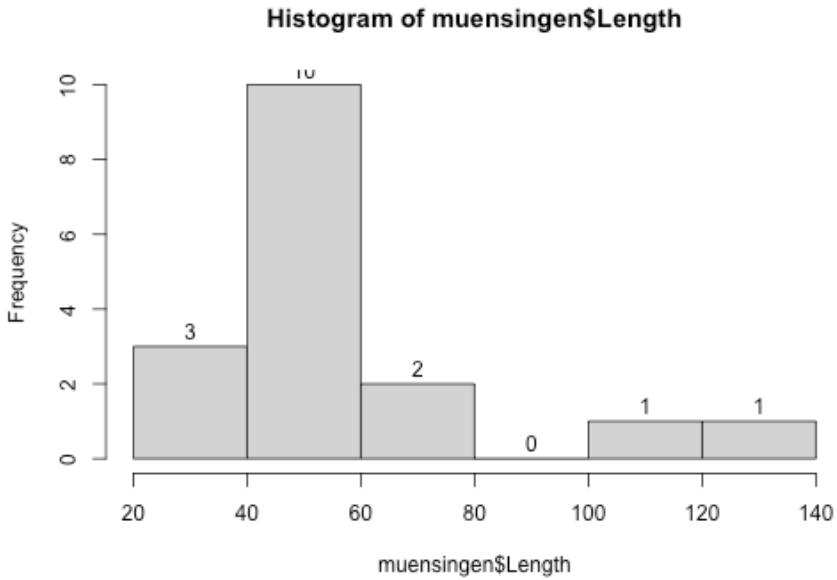
Histogramm [1]

Used for classified display of distributions Data reduction vs. precision: Display of count values of classes of values

```
hist(muensingen$Length)
```



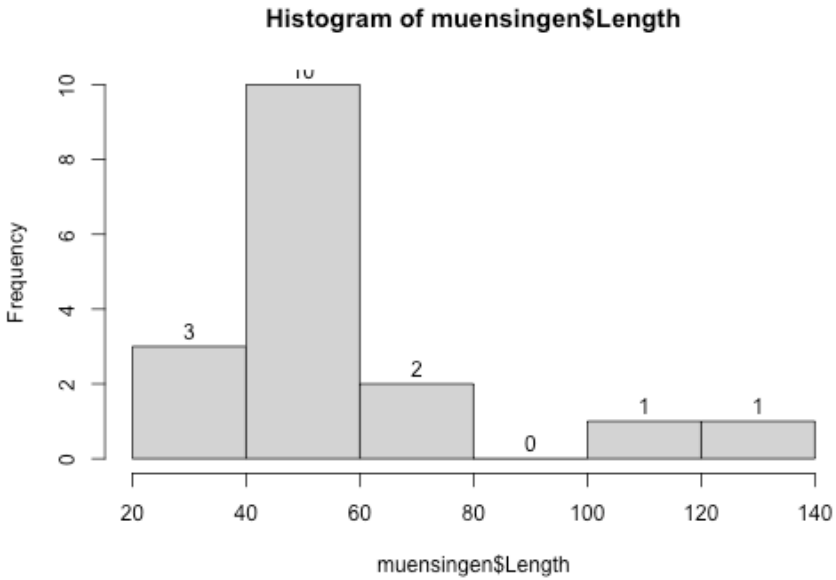

```
hist(muensingen$Length, labels = T)
```



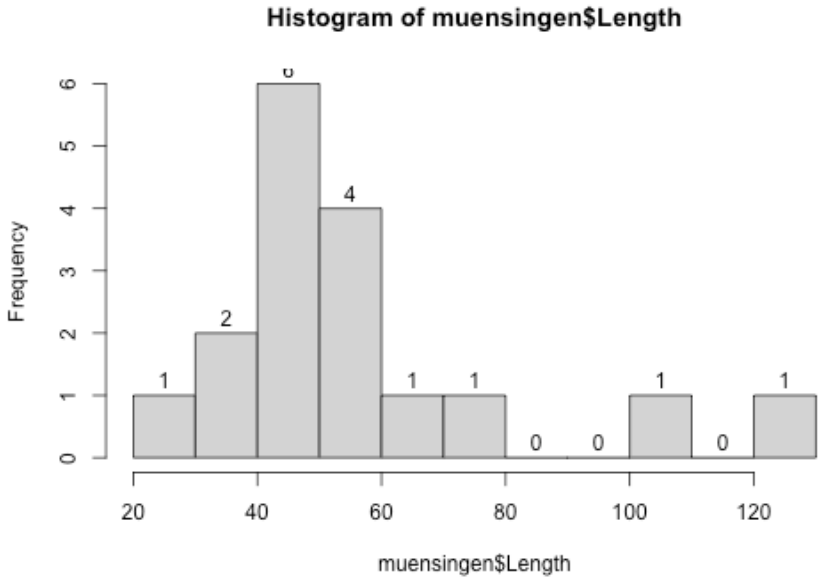
Histogramm [2]

Custom breaks of classes

```
hist(muensingen$Length,
     labels = T)
```



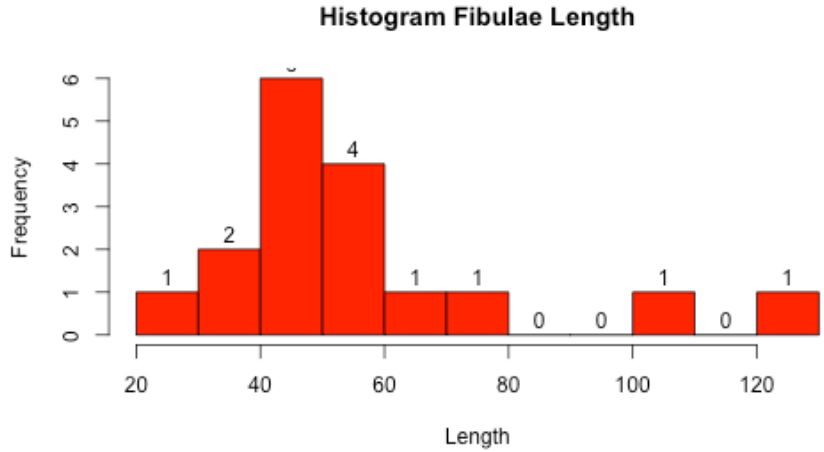

```
hist(muensingen$Length,
     labels = T,
     breaks = 10)
```



Histogramm [3]

More beautiful

```
hist(muensingen$Length,breaks=10,
     labels=T,
     col="red",
     xlab="Length",
     main="Histogram Fibulae Length")
```



Disadvantages:

- Data reduction vs. precision → loss of information
- Actual display depends strongly on the choosen class width

steam-and-leaf chart

An attempt to overcome the disadvantages of a histogram

Is not very often used. Scales like histograms.

```
stem(muensingen$Length)
```

```
##
##   The decimal point is 2 digit(s) to the right of the |
##
##   0 | 34444
##   0 | 5555566677
##   1 | 13
```

Advantage:

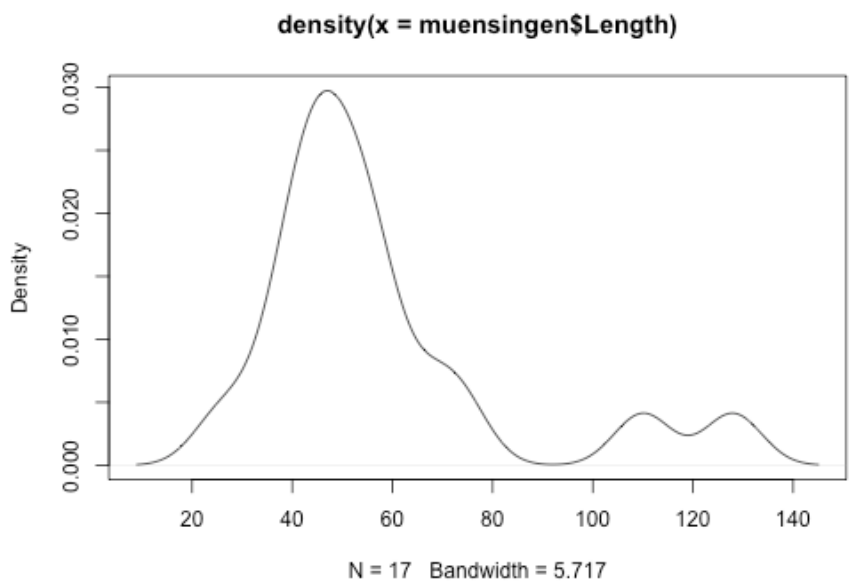
- Information about the distribution inside the classes and the absolute values are (partly) visible.

kernel smoothing (kernel density estimation)

Another attempt to overcome the disadvantages of a histogram

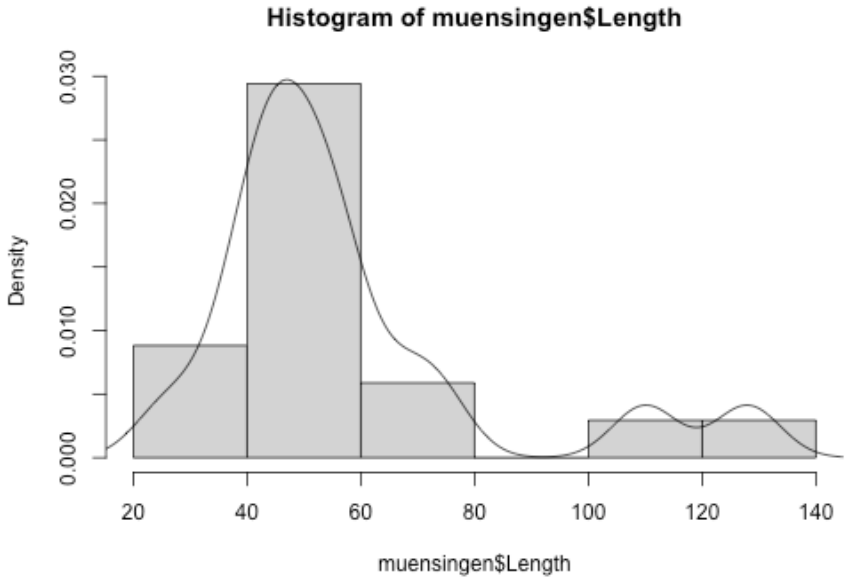
The distribution of the values is considered and a distribution curve is calculated. Continuous distributions are better displayed, without artificial breaks. Scales like histograms.

```
plot(density(muensingen$Length))
```



Histogram and kernel-density-plot together

```
hist(muensingen$Length, prob=T)
lines(density(muensingen$Length))
```

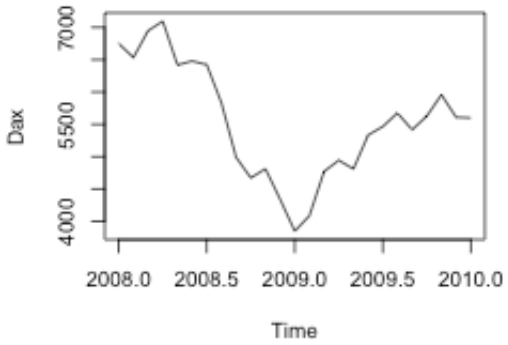


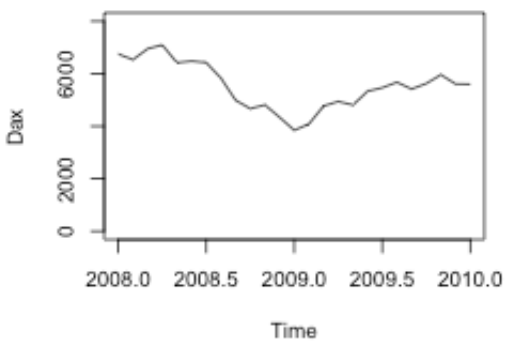
Style of charts

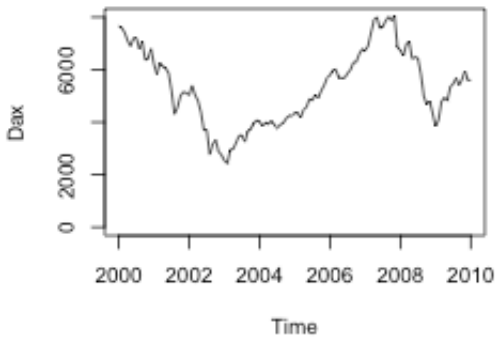
Stay honest!

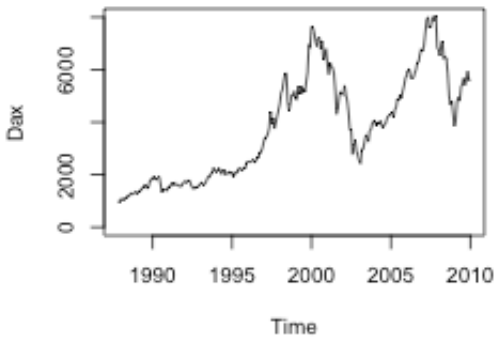
- dax.csv

Choice of display has a strong influence on the statement.









Style of charts

Stay honest!

Choice of display has a strong influence on the statement.

Clear layout!

Minimise Ratio of ink per shown information!

Use the suitable chart for the data!

Consider nominal-ordinal-interval-ratio scale

Suggestions for charts

What to display	suitable	not suitable
Parts of a whole: few	Pie chart, stacked bar plot	
Parts of a whole: few	Stacked bar plot	
Multiple answers (ties)	Horizontal bar plot	Pie chart, stacked bar plot
Comparison of different values of different variables	Grouped bar plot	
Comparison of parts of a whole	Stacked bar plot	
Comparison of developments	Line chart	
Frequency distribution	Histogram, kernel density plot	
Correlation of two variables	scatterplot	