Course "Empirical Evaluation in Informatics"

Data analysis techniques

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Samples and populations
- The mean
- The variability
- Comparing samples
  - significance test, confidence interval
- Bootstrap
- Simple relationships of two variables
  - Plots, log-Scales
  - Correlation, linear models
  - local models (loess)

"Empirische Bewertung in der Informatik"

Techniken der Datenanalyse

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
http://www.inf.fu-berlin.de/inst/ag-se/

- Stichproben und Grundgesamtheiten
- Der Mittelwert
- Die Variabilität
- Vergleich von Stichproben
  - Signifikanztest, Vertrauensbereich
- Bootstrap
- Einfache Beziehungen zwischen zwei Variablen
  - Plots, log-Skalen
  - Korrelation, lineare Modelle
  - lokale Modelle (loess)

First note: samples and populations

- At the start of a statistical analysis, we usually have some subset ("sample", "*Stichprobe*") of all possible values of some kind ("population", "*Grundgesamtheit*")
  - e.g. data for a size 50 subset of all FUB Informatics students
- The goal of analysis is making valid statements about the population on the basis of
  - the sample alone (*frequentist approach*) or
  - the sample plus prior beliefs about the population (*Bayesian approach*)

Warning: sampling is difficult

- Both approaches will work well only if the sample is representative
    - that is, each member of the population had the same chance of being in the sample

- Obtaining a representative sample is very difficult
    - Often the boundaries of the population are unclear
        - Is a guest student a member?
        - Is a Nebenfach-student a member? etc.
    - It is unknown how to sample randomly with even chances
        - e.g. just catching people when passing the foyer is insufficient
    - Often the member we picked for our sample will refuse to cooperate

- So all conclusions must be considered with care
    - The conclusions are only "estimates"

Again: Possible tasks of data analysis

- Measure a variable
- Compare two (or more) variables
- Model a relationship

Measure a variable: what does the mean mean?

- Given: a set of measurements of the variable
- So we have a sample of a population. Which population?

- **Case 1**: There is a single "true" value and we have a set of measurements with errors.
  - e.g. 10 measurements of the length of the same road
  - Case **a)**: We are perhaps interested in the true value only, not in the population of measurements
    - The sample mean is an estimate of the true value
  - Case **b)**: But maybe we try to understand the measurement method, not the road.
    - (e.g. for research on software inspection techniques)

    Then we are interested in the population, not the true value
    - The *error* in the measurements is what we want to characterize

What does the mean mean? (2)

- Case 1: There is a single "true" value and we have a set of measurements with errors. [...]

- **Case 2**: There is a stochastic variable (i.e. it has variability) and we have a sample of its values
  - e.g. each person's age in a sample from a population of people
  - We are interested in the "average" or "expected" case
    - The sample mean is an estimate of the mean age
  - There is a true value of the mean age of the population,
    but not a true value of the age of the population
    - The age of the population can be partially characterized by looking at the mean plus the *variation* of the age

What we need

- Estimates of the "expected" value of the variable
  - mean, median, mode, etc. (measures of "location")
- Estimates of the variation ("variance") of the variable
  - standard deviation, median absolute deviation, quantile ranges, etc. (measures of "scale")
- Estimates of the error in the estimates
  - e.g. standard error of the mean, confidence limits

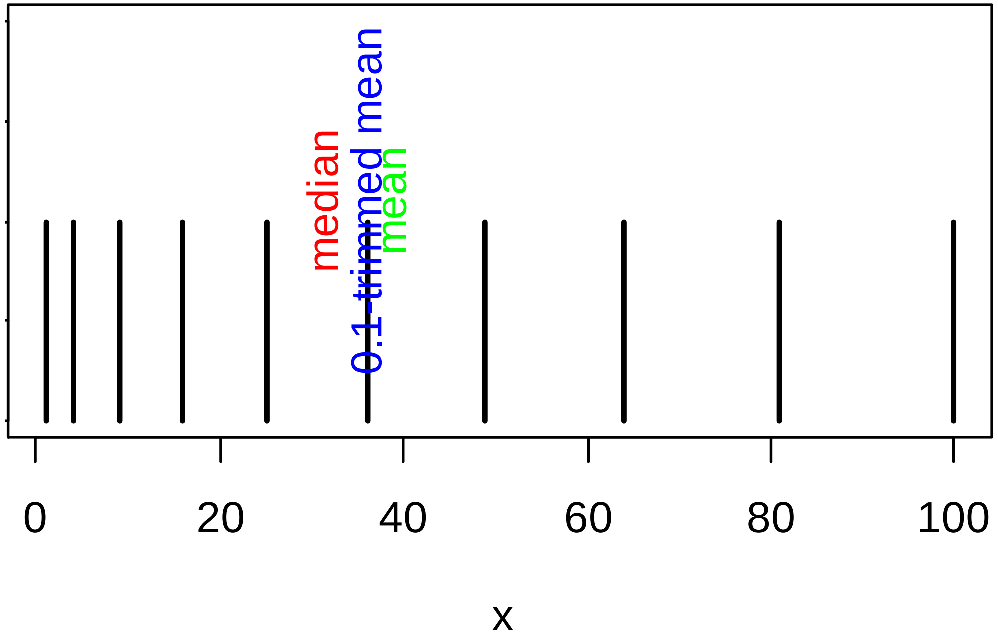
- Note: There are different ways of defining "error", too
  - They lead to different measures and methods
  - They are appropriate in different situations
  - But most of this is beyond the scope of this lecture

Estimators for expected value

- Arithmetic mean
  - Most common
  - Can be used only on a difference scale or ratio scale
- Median (the 50/50 cut point)
  - Required if all we have is an ordinal scale
  - Also useful if we want to be robust against few extreme values
    - Ignores distance; inefficient (i.e. much information remains unused)
- Mode (the most frequent value)
  - Required if we only have nominal data (unordered)
  - Sometimes useful for ordinal scales with few values
- Trimmed mean (leave out a top/bottom fraction of the data points)
  - Robust against outliers, without ignoring distance
- M-estimators
  - very advanced technique, robust and efficient

Expected value estimation example

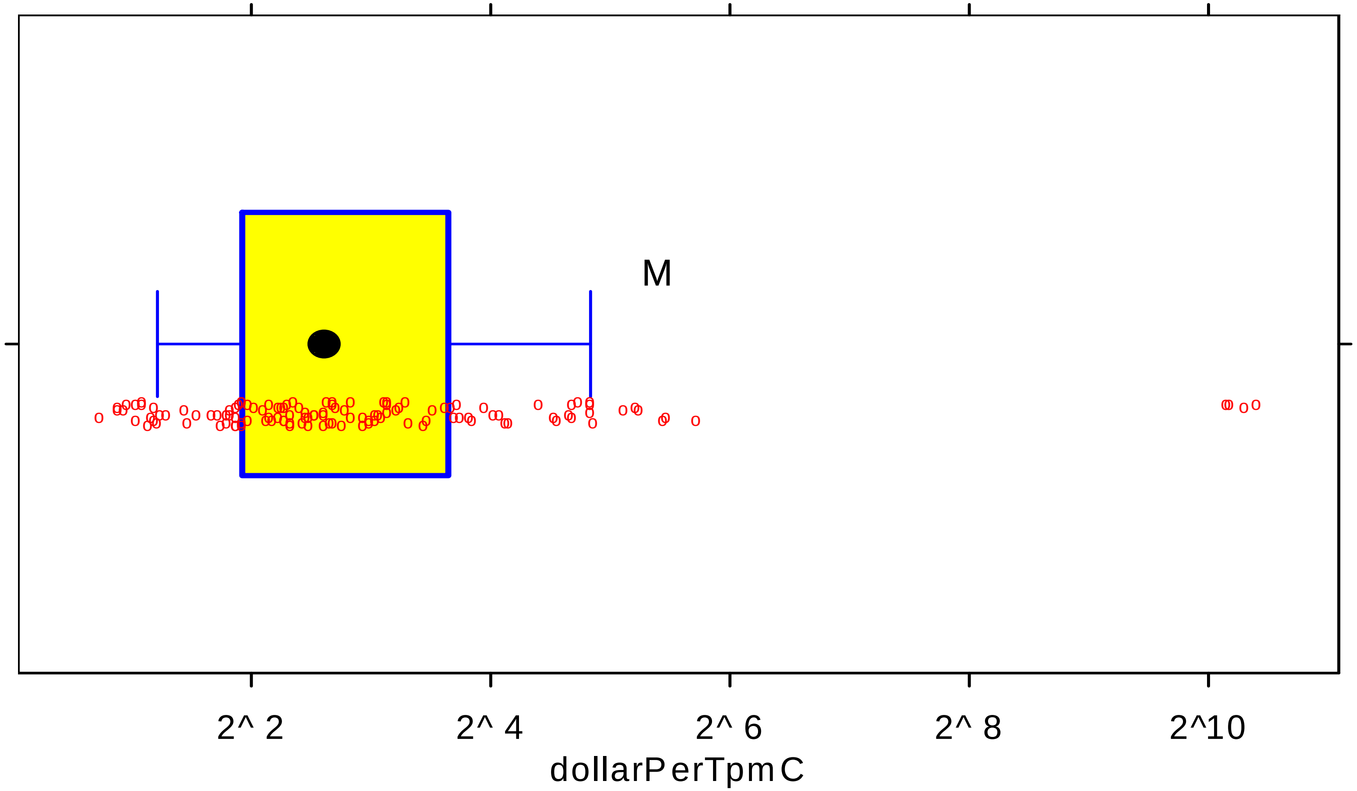
- x=(1:10)^2=
  c(1,4,9,16,25,36,
   49,64,81,100)
- median(x)=
  (25+36)/2=
  30.5
- mean(x,tr=0.1)=
  mean(c(4,9,16,
   25,36,49,64,81)
  =35.5
- mean(x)=38.5



- Base plot: plot(x, rep(1, length(x)), type="h")

Expected value estimation example (2)

- From the TPC data:



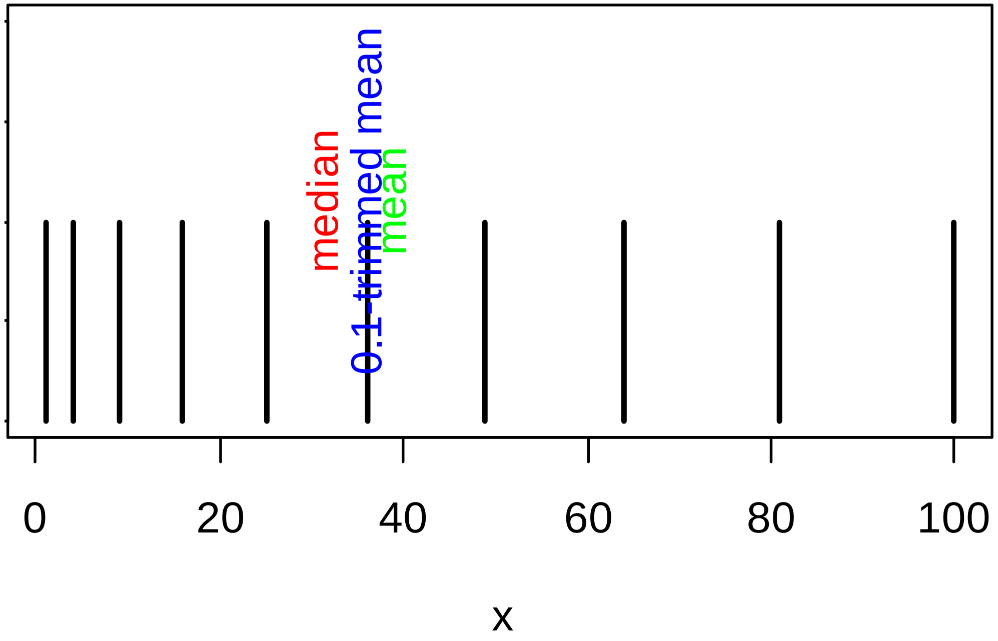
- median=6.1
- 0.1-trimmed mean=8.5
- mean=48

Estimators for variation

- Standard deviation
  - mean distance of a value from the mean
  - R: sd(x) or sqrt(var(x)) or mean(abs(mean(x)-x))
- Median absolute deviation
  - median distance of a value from the median
  - R: mad(x, constant=1) or median(abs(median(x)-x))
  - normal-consistent estimate is mad(x)
    - (i.e. equal to sd(x) for large samples from normal distributions)
    - less efficient estimator than std.dev., but robust to outliers
- Interquartile range
  - difference of the 0.75 and 0.25 quantiles
  - R: IQR(x) or diff(quantiles(x, c(0.75,0.25)))
  - normal-consistent estimate is IQR(x)/1.349
  - Note: interquartile range is related to the median, (not to the trimmed mean)

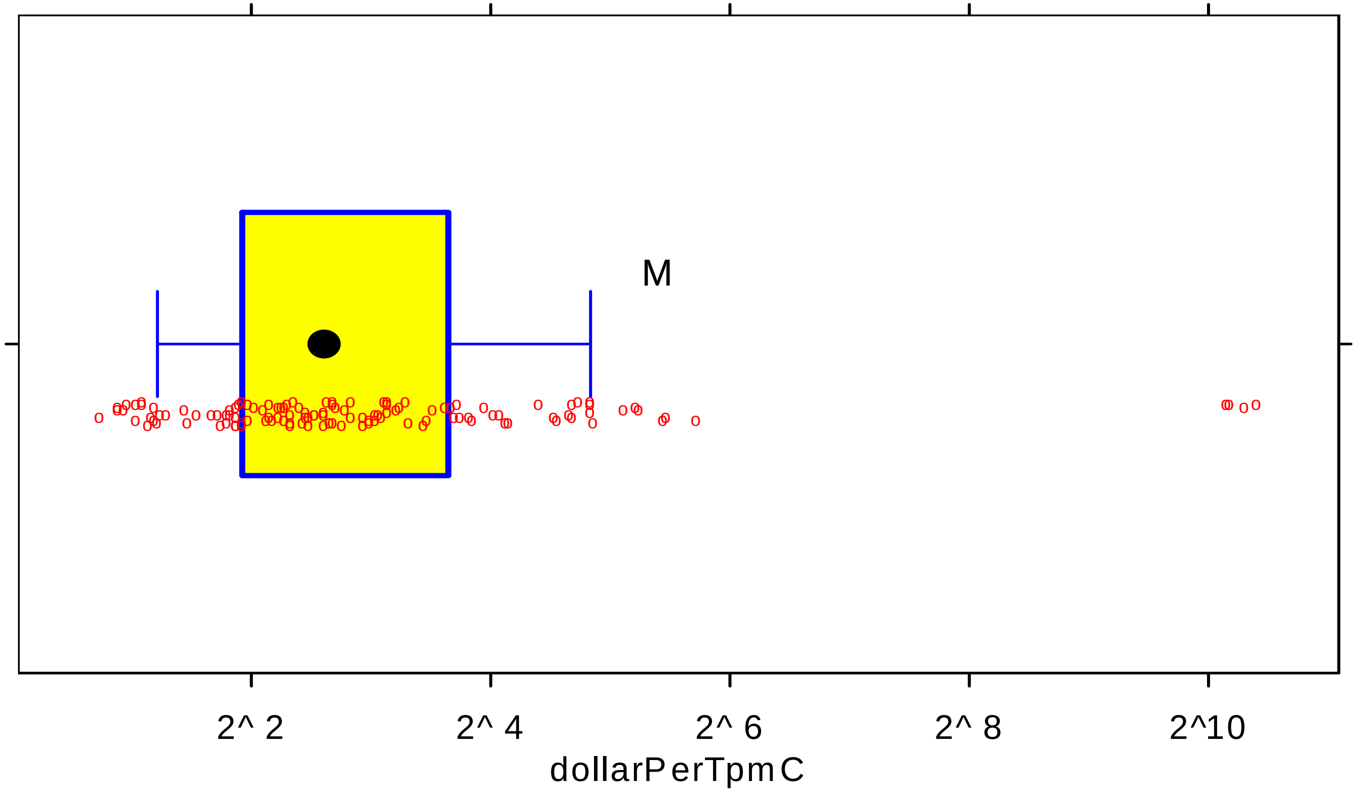
Variation estimation example

- x=(1:10)^2= c(1,4,9,16,25,36, 49,64,81,100)
- sqrt(var(x))= sd(x)= 34
- mad(x)= 36
- IQR(x)/1.349= 37
- mad(x,const=1)= 24
- IQR(x)= 49.5



Variation estimation example (2)

- From the TPC data: x = dollarPerTpmC

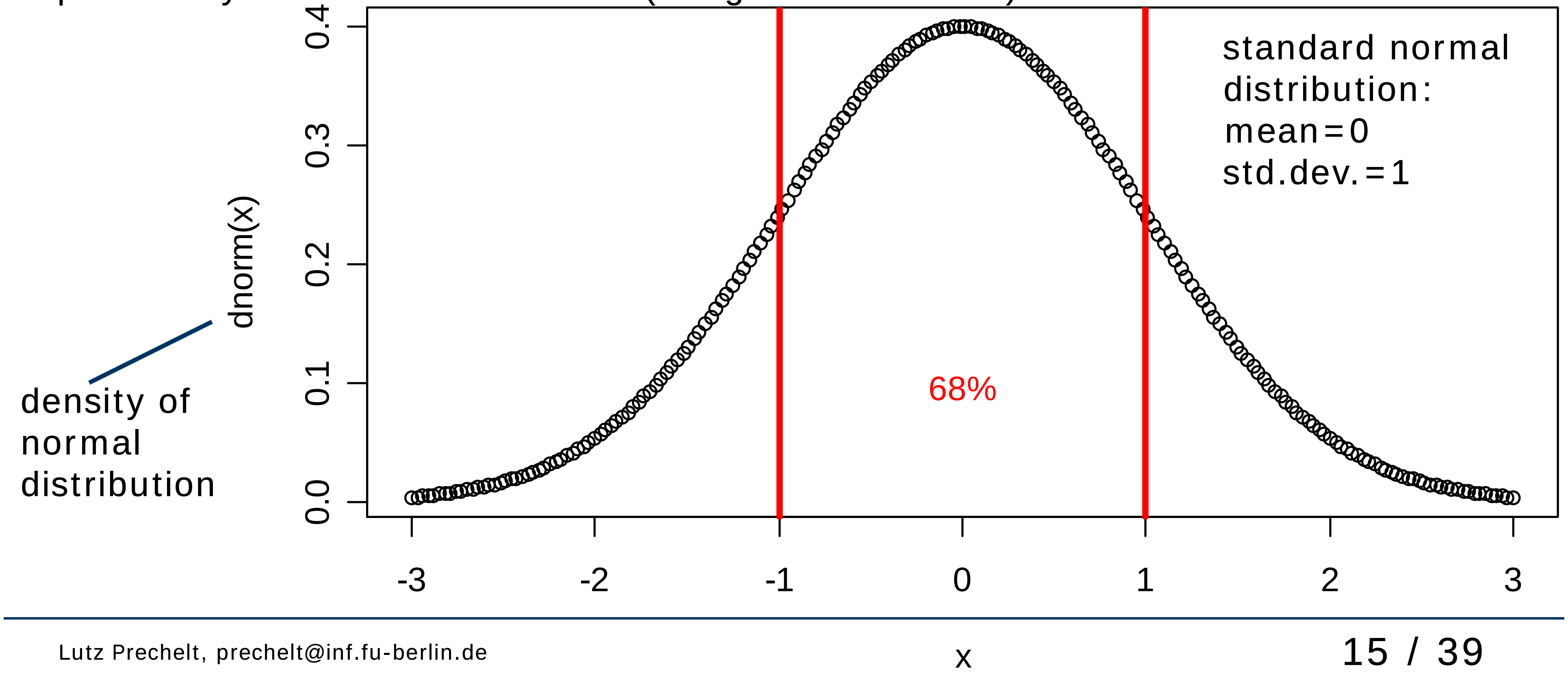


- sd(x) = 214
- mad(x) = 4.1
- IQR(x)/1.349 = 6.5

The standard normal ("Gaussian") distribution

- 68%/95%/99.7% of all values fall within 1/2/3 standard deviations around the mean
  - R: pnorm(1)-pnorm(-1)=0.683
  - pnorm(1:3)-pnorm(-1:-3) = 0.683 0.954 0.997

probability of normal distribution (integral over dnorm)



density of normal distribution

Estimators for error: standard error

- Standard error (se, stderr) of the mean
  - is the standard deviation of the mean-estimates that are based on samples of size N from the same distribution
  - R: se = sd(x)/sqrt(length(x)) = sqrt(var(x)/length(x))

- The best way of expressing estimated errors is by means of a confidence interval:
  - e.g. with 68% probability, the true mean will be in the range mean-se…mean+se
    - so we have 68% confidence the mean will be in this range
    - [mean-se,mean+se] is called a 68% confidence interval for the mean
  - [mean-2*se,mean+2*se] is a 95% confidence interval for the mean, etc.

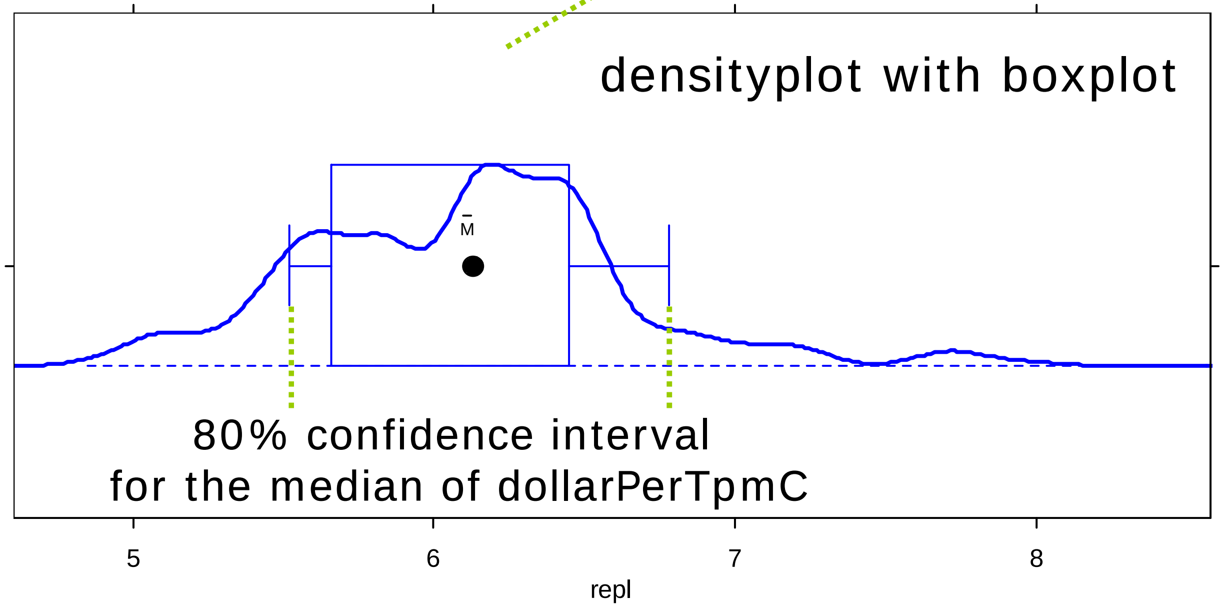
- TPC dollarPerTpmC: mean=48, std.err=19

Estimators for error: bootstrap

- Generally, estimating errors and confidence intervals is mathematically very challenging
  - std.err of the mean is one of the few simpler exceptions
- One possible replacement for strong theory is bootstrapping
  - More formally known as Bootstrap resampling
- Bootstrapping means simulating many trials by
  - treating the sample as if it was the population
  - computing many replicates of the statistic of interest
  - and observing the variation.
- However, for many kinds of statistics, further considerations are required
  - in particular, compensating for bias
  - again, this is beyond the scope of this lecture

Bootstrap example

- We bootstrap the median of dollarPerTpmC:
  - xx = tpc$dollarPerTpmC
  - repl = replicate(1000, median(sample(xx, replace=T)))
  - mean(xx)=48, se_{mean}=19, median(xx)=6.1, $\text{se}_{\text{median}}$=sd(repl)=0.54
  - bias = mean(repl)-median(xx) = -0.02
- R support: library(boot)


densityplot with boxplot

80% confidence interval
for the median of dollarPerTpmC

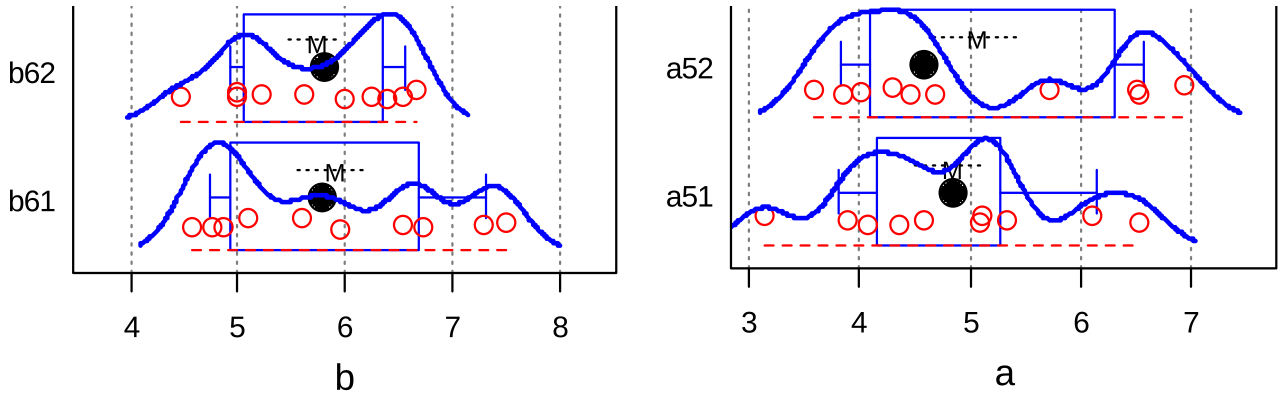
Compare two or more samples

- We often want to compare two or more different samples of a variable (e.g. from 2 experiment groups)
- Essentially what we want is a confidence interval for the difference of the means
  - rather than the much more common, but much less informative p-value (as produced by a significance test)
  - The meaning of the p-value is this:
    - If there is in fact really no difference between the groups...
    - ...then the probability of obtaining a difference at least as large as the one you have seen is p.
  - If p is small, the difference is called "statistically significant"
    - (which basically tells you that the sample was large enough)
- If the samples are both from a normal distribution, the R procedure *t.test* computes such an interval
  - iff you are sure that both distributions have the same variance, set var.equal=TRUE; makes the test more efficient

Example: Comparing two pure normal distributions

- for each block of two pairs of samples (bottom to top):
  - n=10,50,50, μ_b=6,6,5.1, μ_a=5,5,5, σ=1,1,0.2
  - t-test, assuming unequal variance

p-value
80% confidence interval

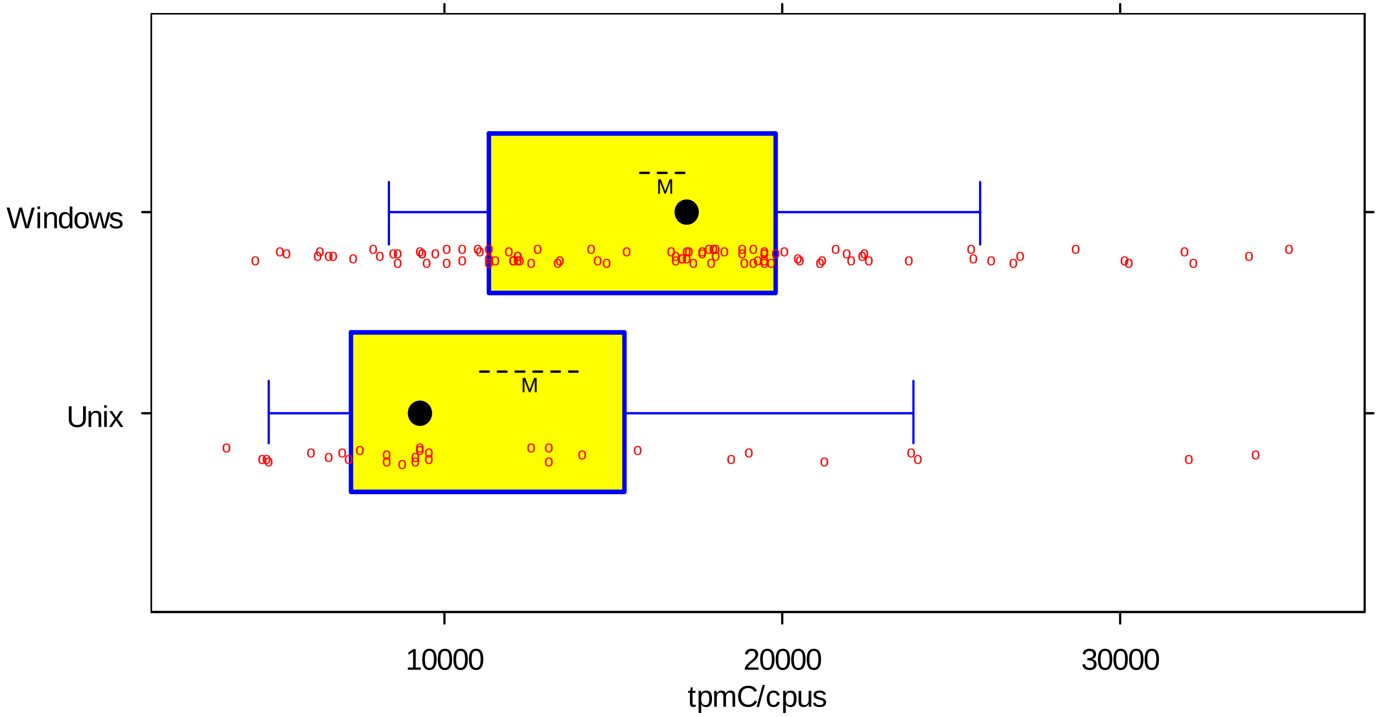


p=0.176 0.05...1.28

p=0.036 0.44...1.70

Example: Comparing tpmC per processor

- Now consider the tpmC performance per processor:
  - How large is the Windows/Unix difference and its 95% confidence interval?



Example,
using normal distribution theory

- x = (tpc$tpmC/tpc$cpus)[tpc$ostype=="Windows"]
- y = (tpc$tpmC/tpc$cpus)[tpc$ostype=="Unix"]

- t.test(x,y): **df = 43.62, p-value = 0.016**
  alternative hypothesis: true difference in means is not equal to 0
  95 percent confidence interval: 803 7258
  sample estimates: mean(x)=16544, mean(y)=12514

- or, assuming equal variances in the populations:
- t.test(x,y,var.equal=T): **df = 125, p-value = 0.0079**
  95 percent confidence interval: 1078 6983

Example, using bootstrap

- Bootstrapping is a general method for computing conf. interv.
  - making fewer assumptions (in particular: no normality needed)
- library(boot)
- dat = cbind(c(x,y), c(rep(1,length(x)),rep(0,length(y))))

-
  ```
  bb=boot(dat, function(d,i) mean(d[i,1][d[i,2]==1])-
                             mean(d[i,1][d[i,2]==0]),
          R=1000)
  ```
- boot.ci(bb)

-
  ```
  Intervals :
  Level      Normal              Basic
  95%   ( 953, 7195 )   (1094, 7446 )
  Level     Percentile            BCa
  95%   ( 615, 6967 )   ( 406, 6884 )
  ```

```
t-test:
803 7258
```

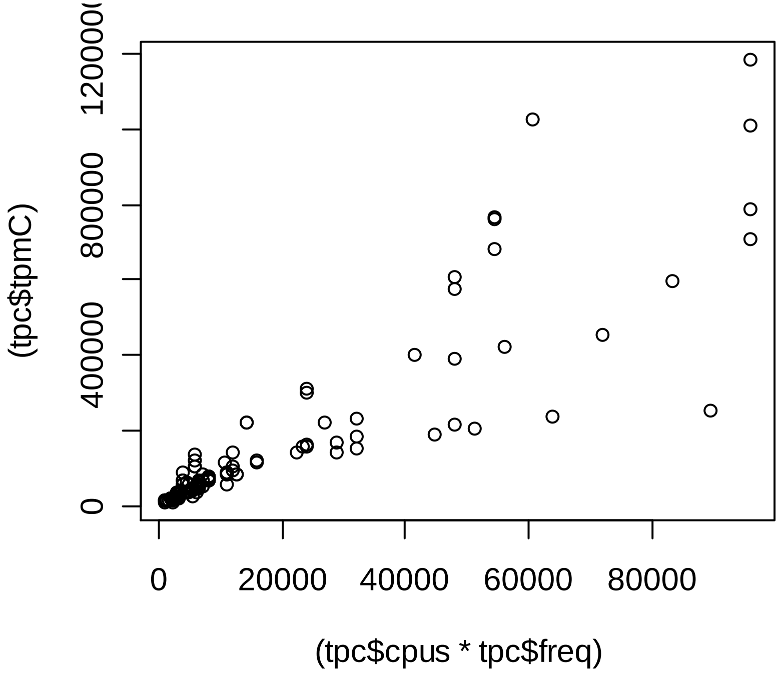
- When in doubt, the BCa interval ("bias-corrected and accelerated") may be your safest bet

Model a relationship

- Often we want to know whether there is a relationship between two or more variables
  - and what this relationship is
  - Its nature may be causal or purely correlational

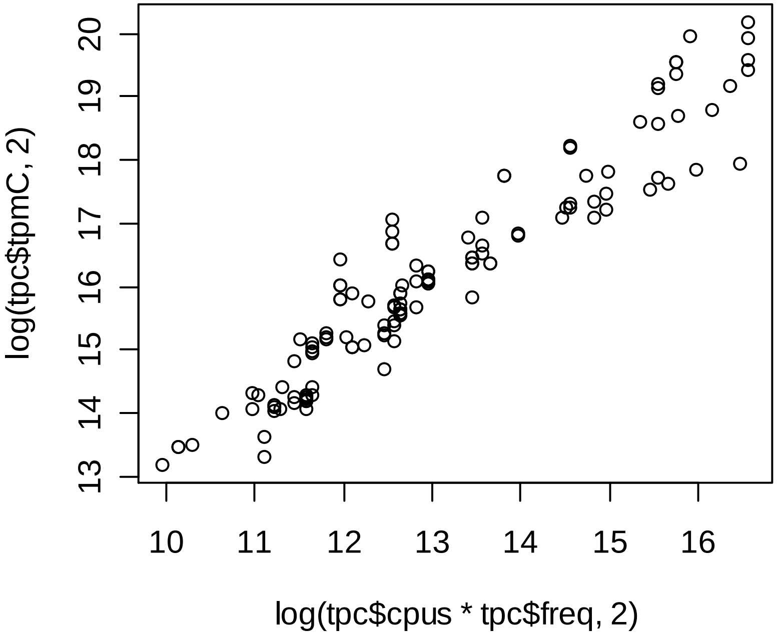
- The basic case is two variables on a ratio scale

- The basic approach is the scatter plot
  - Example: tpmc vs. total clock speed
  - plot(cpus*freq, tpmC)
  - Is there a relationship? Probably yes, but the data cluster too much near the small values
  - Let us use a log scale instead



Log-log scale scatter plot, correlation

- plot(log(cpus*freq,2), log(tpmC,2))
- Yes, there is is quite obviously a strong linear relationship between these parameters
- The strength can be quantified by means of the correlation coefficient r
  - cor(log(cpus*freq,2), log(tpmC,2)) = 0.95
  - Watch out: Correlation is sensitive to the scale:
  - cor(cpus*freq, tpmC) = 0.88
  - Note: The computation assumes that the deviations from the relationship follow a normal distribution
    - So the non-log cor is not valid in this case

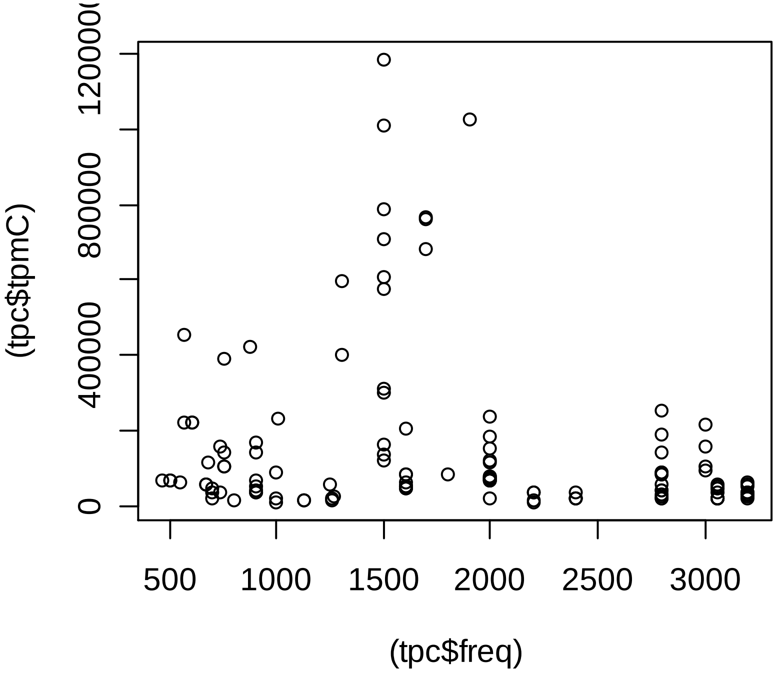


More on correlation

- cor(log(cpus*freq,2), log(tpmC,2)) = 0.95
- cor(cpus*freq, tpmC) = 0.88

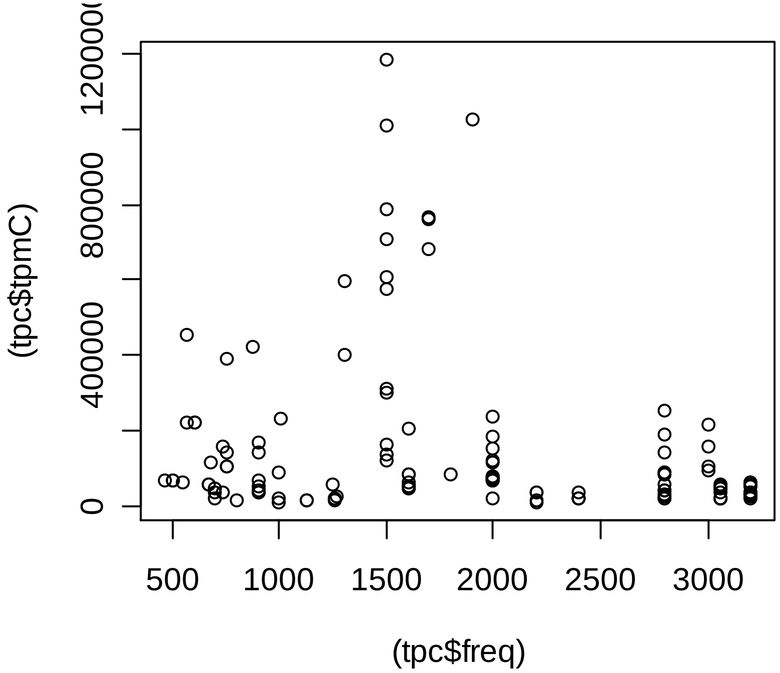
- You can ignore scale entirely by using rank correlation:
  - cor(rank(cpus*freq), rank(tpmC)) = 0.94
    - uses rank numbers instead of actual data values (for data on less than a difference scale, this is the only allowed way)

- For less nice examples (with outliers), the results can be quite different
  - cor(freq, tpmC) = -0.195
  - cor(rank(freq), rank(tpmC)) = -0.28
  - because the normality assumption is violated



Confidence interval for the correlation coefficient

  - cor(log(cpus*freq,2), log(tpmC,2)) = 0.95
  - cor(cpus*freq, tpmC) = 0.88
- Again we use the Bootstrap:
  - xx = cbind(log(cpus*freq,2), log(tpmC,2))
    bb=boot(xx, function(d,i) cor(d[i,1], d[i,2]), R=1000)
    boot.ci(bb)
  - 95% BCa interval: 0.929 0.964
- The other example:
  - cor(freq, tpmC) = -0.195
  - xx = cbind(freq, tpmC)
    . . .
  - 95% BCa interval: -0.285 -0.099



Note: Impressing laymen

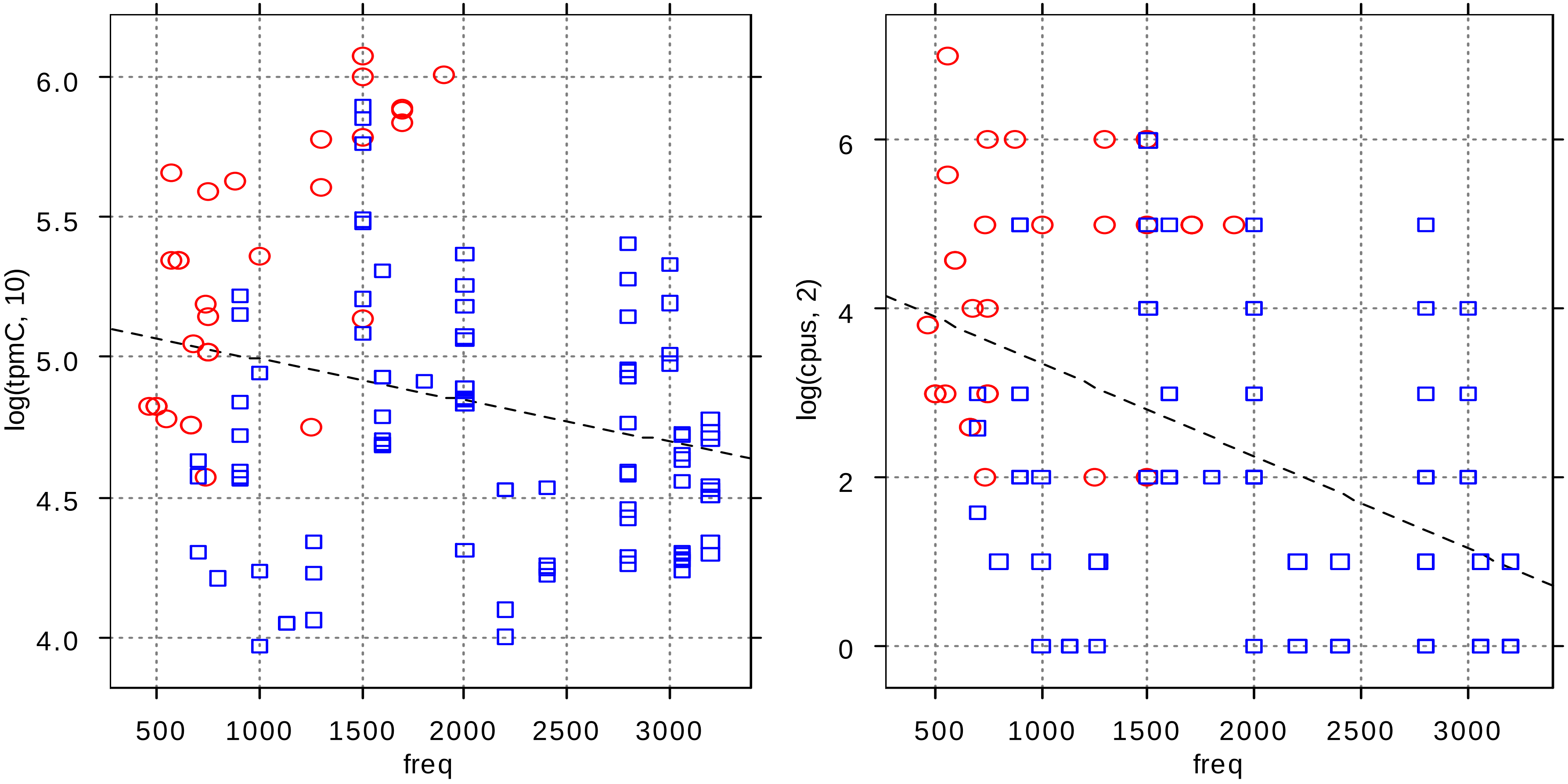
- Some studies contain statements like this:
  - *"The Pearson correlation coefficient is significant at level alpha = 0.05"*
  - This talks about a hypothesis test against the null hypothesis that r = 0
- This sounds impressive, but means nothing more than that there *may* be some correlation (however small)
  - precisely: it means that if there is no correlation at all in the population, it is unlikely (<5%) to obtain such samples
    - Hence if you had previous grounds to believe in correlation, the data does not suggest you need to drop that belief
  - In most cases this is of very little interest
- When you see such a statement, the best reaction is usually to be very heavily unimpressed

Correlation and causation

- Warning: **Remember that a correlation need not indicate causality**
  - cor(freq, tpmC) = -0.285...-0.099 (95% ci)
    means that increasing processor clock rate correlates with a *decreasing* rate of transactions per minute
    - This correlation can clearly not be causal: everything else the same, a faster clock would *increase* the transaction rate
  - So?
    - You need to know enough about your data:
  - The real reason is that the faster-clock (Windows) systems tend to have much fewer processors than the slower-clock (Unix) systems
    - The decreasing transaction rate is a property of the tpc data set, not of the clock frequency

freq and tpmC versus freq and cpus

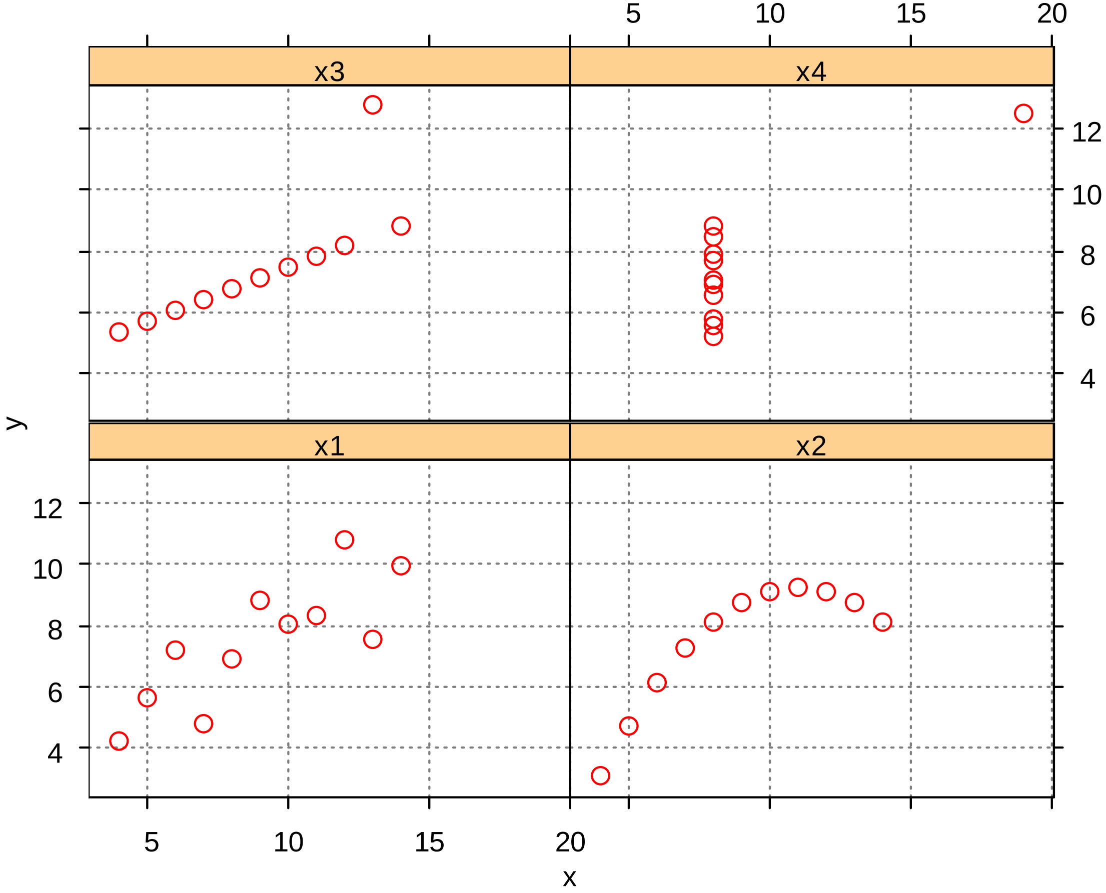
- xyplot(log(cpus,2)~freq, data=tpc,
  panel=panel.superpose, groups=ostype)



Problems with summary statistics

- A further warning: The correlation, even in conjunction with other summary statistics, does not tell much about the nature of a relationship

- The following plots all share the same correlation (0.82), means (x=9, y=7.5) and standard deviations (x=3.3, y=2)
  - data(anscombe)
    - 'stack' for repackaging
    - xyplot



Describing the relationship between x and y

- Since the correlation coefficient does not provide enough information, a scatter plot is usually advisable

- Where appropriate(!), a linear regression line can be used to visualize a trend in the data
  - use panel.lmline or type="r" with panel.xyplot
  - the function that computes the regression is lm
    - lm: "linear model"

- lm can also compute regressions for more than one predictor variable or results other than straight lines
  - linear models are the most important technique of professional statisticians
  - Again, this is beyond the scope of this lecture

Attention with linear models!

- Assume we have a sample of pairs (x,y) and we assume there is a systematic relationship (linear, for now)
  - Case 1: For any x, there is a single "true" value of y
    - Case 1A: Our x are precise, but the y are measurements with errors (and those errors have normal distribution!)
    - Case 1B: The x have errors as well
  - Case 2: The relationship is stochastic. For any x, there is a single expected value of y, but actual values do vary
    - Case 2A: Our x are measured precisely, but the y may have errors
    - Case 2B: Our y are measured precisely, but the x have errors
    - Case 2C: Both x and y are measured with errors

- The standard linear regression formula makes assumptions that are met only by cases 1A and 2A
  - 1B and 2C require advanced theoretical knowledge!
  - So be careful what you do

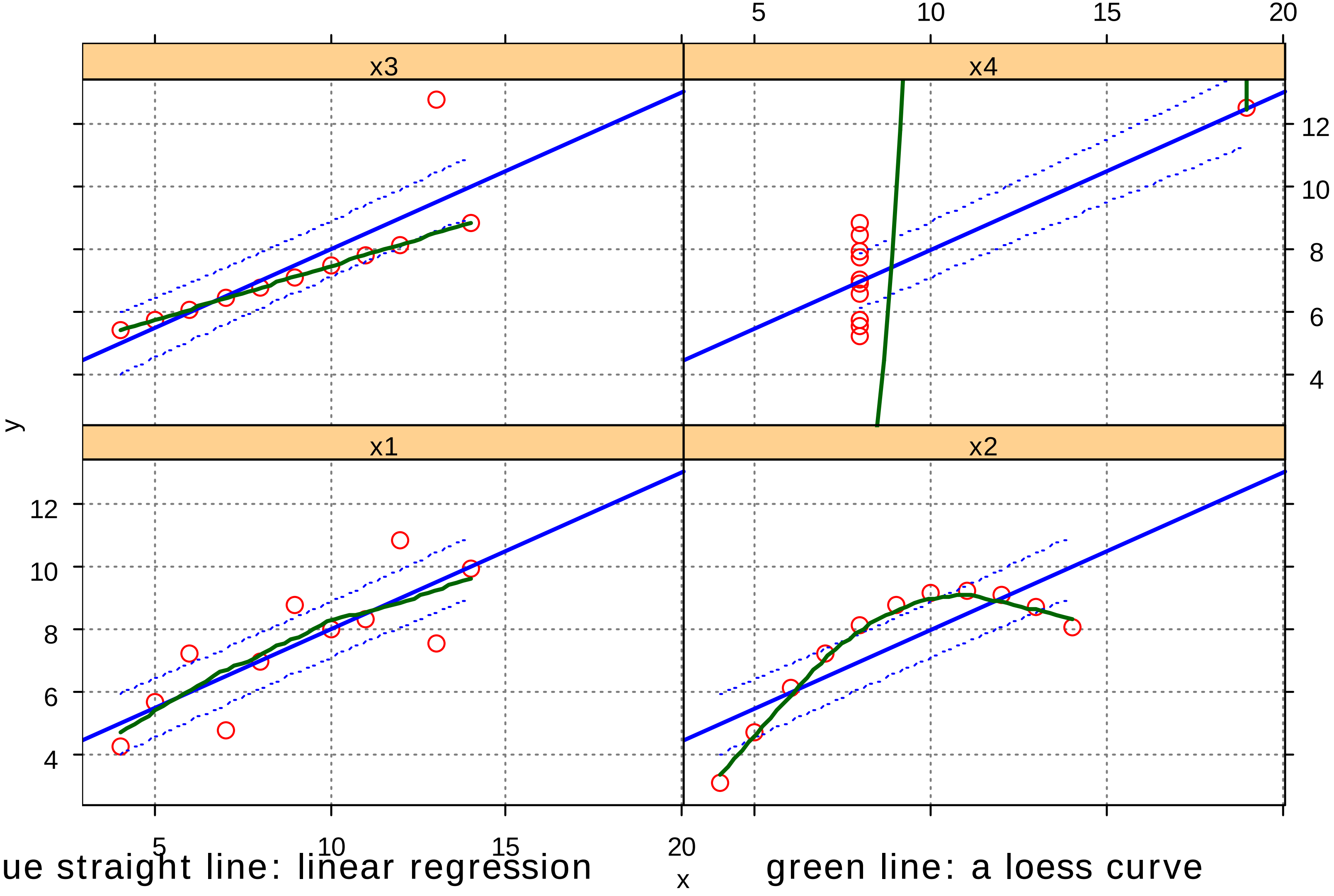
Non-linear trends

- Often a straight regression line is not a suitable fit
- If we know a suitable fitting function f, there are two approaches:
  - Transform the data, using the inverse of f, so that the data fit with a straight line
  - or fit a curve rather than a straight line
- Transforming the data may also lead to a more uniform distribution of the data points
  - See the logarithmic transformations we have used

Local trends

- If no appropriate curve function can be found or we do not want to assume a specific kind, we can fit a local regression
    - *loess* = locally weighted regression
    - at each point of the line, we perform a linear regression, but far-away points are weighted less heavily
    - Parameter *span* controls weighting and ignoring of points
    - use e.g. *panel.loess* for plotting

Example: Loess curves



blue straight line: linear regression

green line: a loess curve

Things not covered

- In many cases, numerical linear models are insufficient to characterize the given data
  - Then advanced techniques such as nonlinear numerical models (e.g. *neural networks*) or partially qualitative models (e.g. *classification trees*) may help
- In particular, the data may have temporal aspects
  - Then topics such as *time series analysis*, *random effects models*, and *survival analysis* become relevant
- Or we are looking for a measure that can only be described by a yet unknown combination of our variables
  - *Factor analysis, principal component analysis*
- In many cases, the data to be analyzed is incomplete
  - *"missing data":* an important, often difficult, and subtle matter

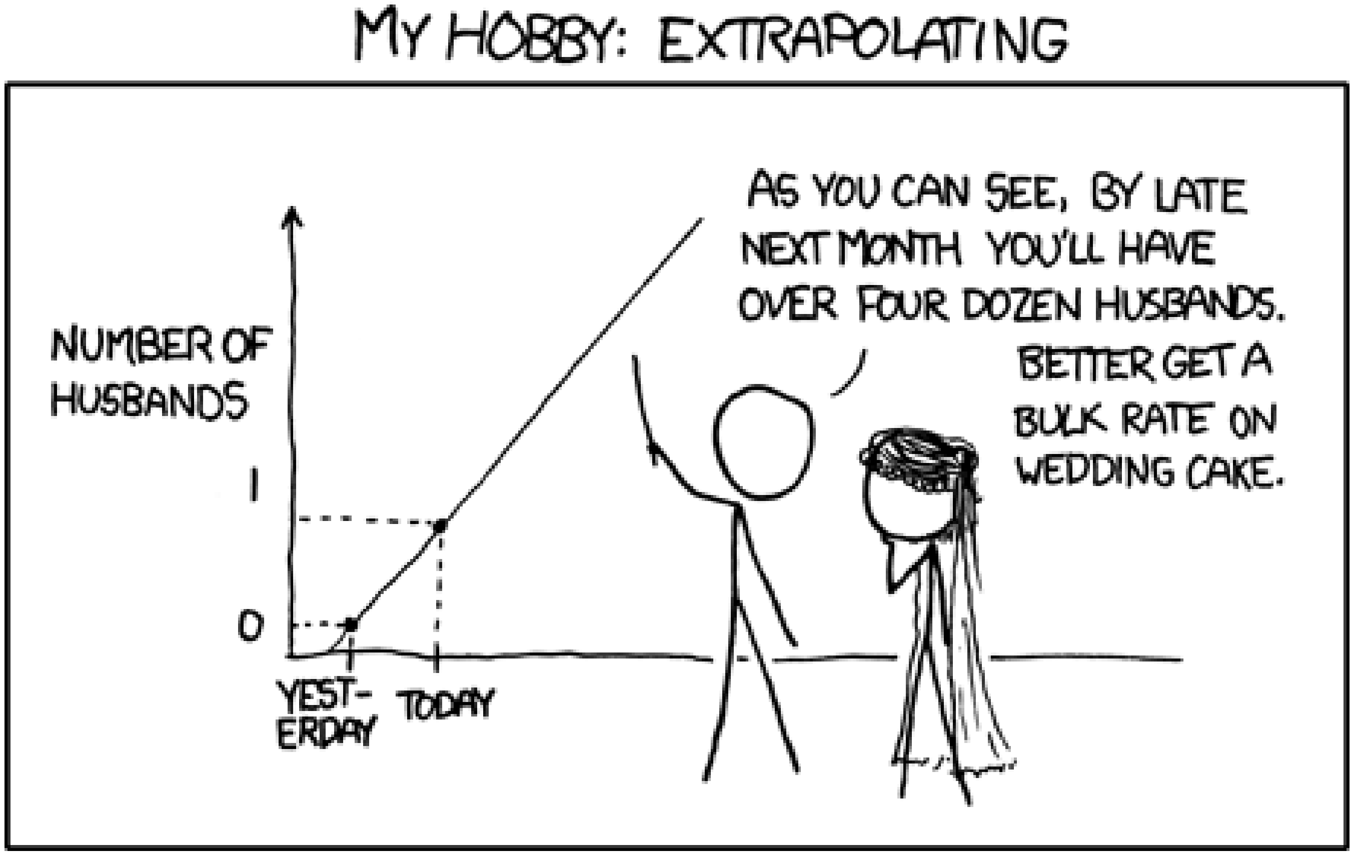
- ...and many others

Final note: Statistics is difficult

- The techniques presented here only scratch the surface of statistical data analysis
  - In some cases, they are sufficient
  - If not, try to get help from a professional statistician

- Rules of thumb:
  - Stick to what you really understand!
  - Beware of ignored assumptions!
    - Violations may be OK, but you need to think about it
  - Back your numbers up by informative plots!
    - Plots produce much higher credibility than bare numbers
    - And are not as likely to be grossly misinterpreted

Thank you!


MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS. BETTER GET A BULK RATE ON WEDDING CAKE.

http://xkcd.com/605/