Course "Empirical Evaluation in Informatics"

Introduction to data analysis

Lutz Prechelt
Freie Universität Berlin, Institut für Informatik

- Conclusions are not obvious
- Possible tasks of data analysis:
  - Exploring
  - Measuring, Comparing
  - Modeling for prediction
  - Modeling for understanding
- Quality criteria
- Steps:
  - make data available
  - validate
  - explore
  - analyze

"Empirische Bewertung in der Informatik"

Datenanalyse: Einführung

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik

- Schlüsse sind oft nicht klar
- Mögliche Aufgaben
  - Erkunden
  - Messen, Vergleichen
  - Modellieren zur Vorhersage
  - Modellieren für Verständnis
- Qualitätskriterien
- Schritte:
  - Daten verfügbar machen
  - Validieren
  - Erkunden
  - Analysieren

Drawing conclusions from data

Consider the following statements:

- "The average consumption of **A** is 2.9 for the C++ group, but 5.7 for the Java group."
  - Conclusion?
    How clear is this conclusion?
- "The average consumption of **B** is 0.9 for the C++ group, but 0.38 for the Java group."
  - Conclusion?
    How clear is this conclusion?
- "The average consumption of **A** is 2.9 for the C++ group (standard deviation 6.3),
  but 5.7 for the Java group (std.dev. 11.1)."
  - Conclusion?
    How clear is this conclusion?

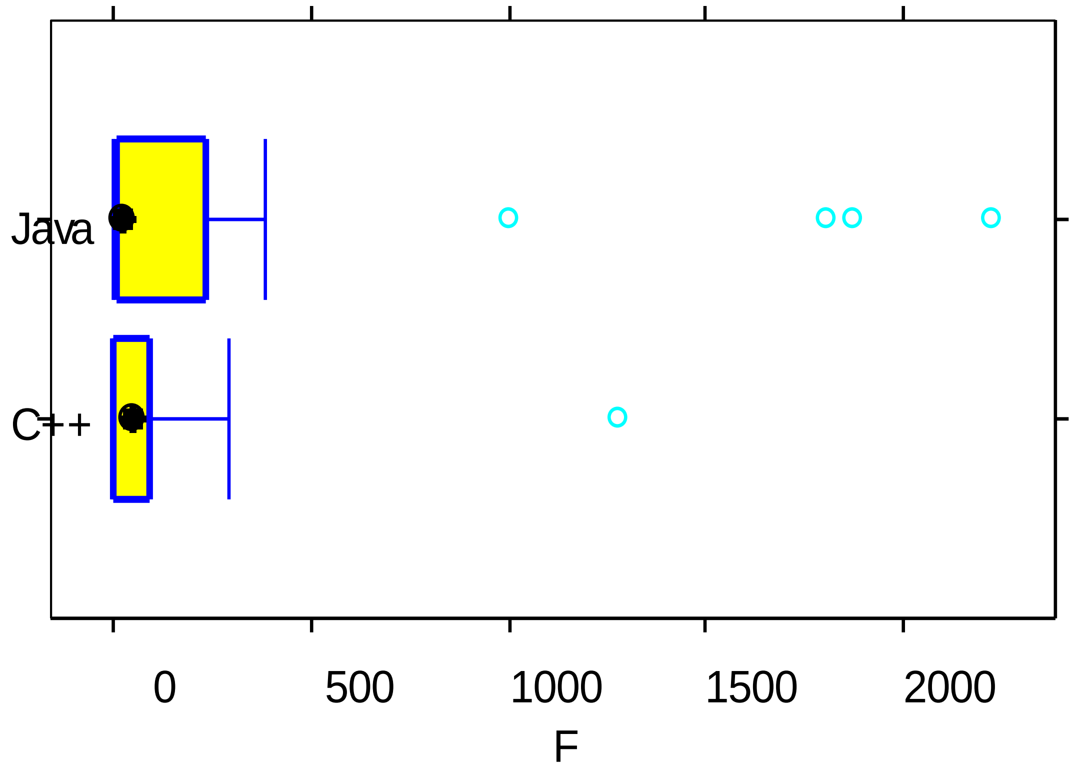
Drawing conclusions from data (2)

- "There is no significant difference in the average consumption of **D** between the C++ group and the Java group ($p=0.36$)"
  - Conclusion?
    How clear is this conclusion?

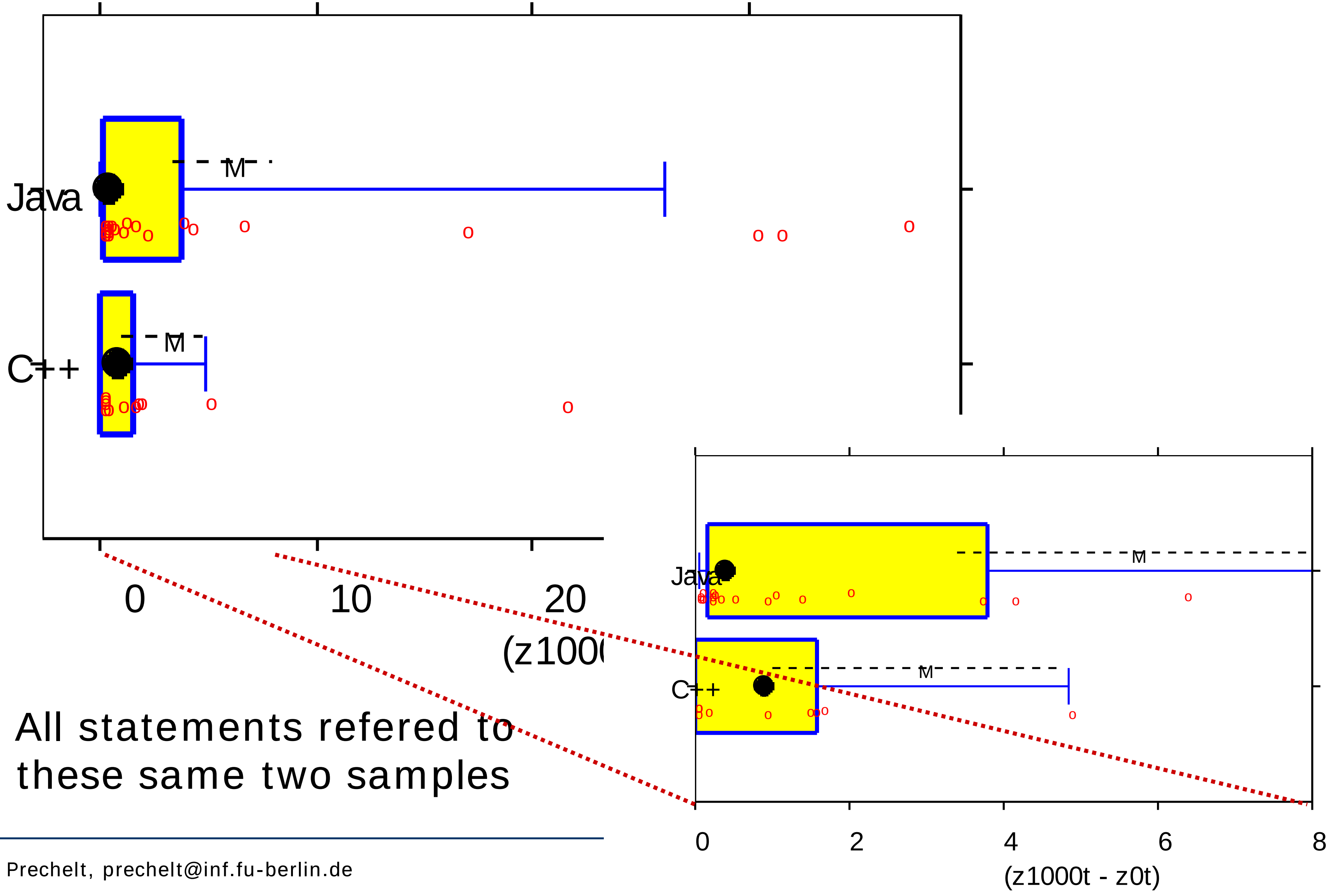
- These statements all refer to the same pair of groups from the same quasi-experiment:
  - C++: $N=11$
  - Java: $N=24$

Drawing conclusions from data (3)

- "Except for a few more outliers in the (much larger) Java group, the boxplots for the consumption of **F** in the Java vs. C++ group look fairly similar"
  - Conclusion?
    How clear is
    this conclusion?

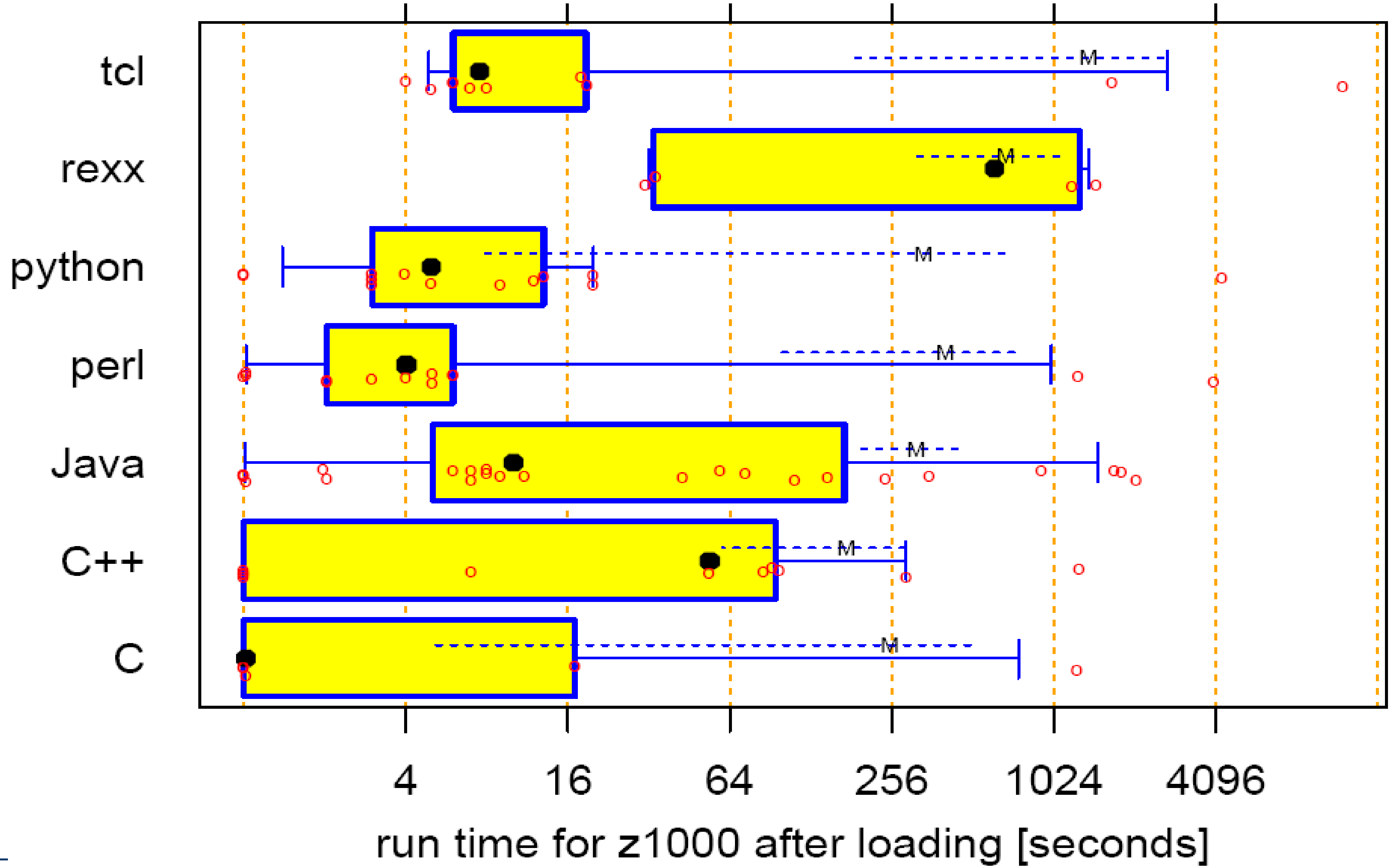


What we have looked at:



- All statements refered to these same two samples

You had even seen this data previously



Lutz Prechelt, prechelt@inf.fu-berlin.de

The problem: Conclusions are not obvious

What we learn from this example:

- Looking at only one aspect of real data is not enough
- Looking at the whole is difficult
  - We can often see certain tendencies
  - but is not always clear what they mean
    - or if they mean anything at all
- ➔ Great care must be taken when analyzing data

- Note we have only compared the values of <u>two</u> different samples of <u>one</u> variable!
- Analysis becomes much more difficult for more complex situations
  - such as comparing relationships between several variables

Possible tasks of data analysis

There are several different kinds of general goal when analyzing data:

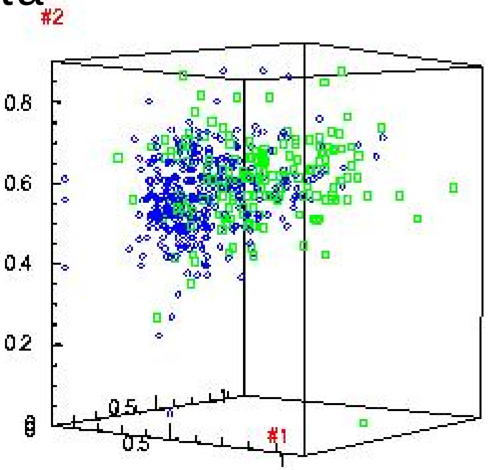
1. Exploring something
2. Measuring something
3. Modeling something for explanation
4. Modeling something for prediction
5. Comparing two or more somethings

- See also http://www.itl.nist.gov/div898/handbook/
  - *The NIST/SEMATECH e-Handbook of Statistical Methods*
  - Note this uses the perspective of analog domains (like manufacturing), not digital domains (like software)

Let's look at each goal:

1 Exploring something

- You do not know in advance what to expect in the data
- You try to get an overview of the data you have and to find interesting structure in the data
  - distributions of samples of individual variables
  - relationships between samples of variables
  - salient characteristics; unexpected characteristics
- Typical goals:
  - creating hypotheses for later investigation
  - finding artifacts, problems, peculiarities in the data
- This is called "Exploratory data analysis" (EDA)
  - almost always a good idea when starting any data analysis
  - it is more an attitude and work style than a concrete task or goal
    - e.g. emphasizing visualization



2 Measuring something

- You know exactly what aspect of an object you are interested in
  - e.g. the number of defects in a particular design

- But the characteristics of the object may make it difficult to measure that aspect precisely:
  - random fluctuations in the measurements (stochastic error)
    - e.g. because your defect detection/estimation method is unreliable (e.g. because it is performed by a human being)
  - systematic measurement error
    - e.g. because certain kinds of defects are almost always overlooked
  - corruption of individual data points
    - e.g. because some part of a defect list has been lost

2 Measuring something (2)

The goals:

1. Determining the measurement value
   - if derived from a sample, this is called a 'point estimate'
2. Determining the expected structure, size, and direction of the error components
   - {stochastic, systematic, corruption}
   - in order to produce a precise and accurate estimate of the aspect of interest

- These activities are particularly important at the very beginning of each data analysis
  - when validating the input data
- or when the measurement itself is the aim of the study
- and also during modeling (see below)

3 Modeling something for explanation

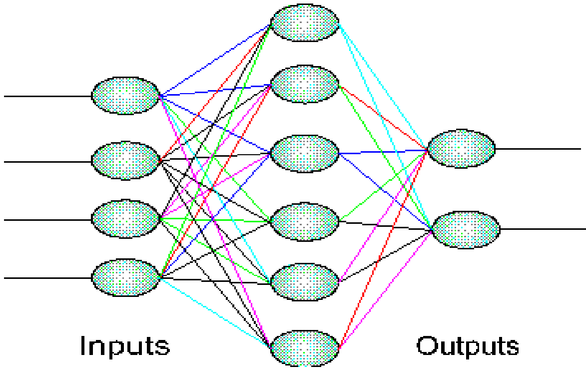
- You want to describe the mechanism that has produced the data
  - "replace" many numbers by a small data generation rule
  - so that the rule makes sense in your domain
  - Example: *"WorkTime = UnderstandingTime + ConstructionTime. UnderstandingTime is 17 minutes per requirements document page on average. ConstructionTime is ..."*

$$P_K = \frac{\pi^2 \cdot EI}{S_K^2}$$

- Such models are the quantitative ingredients of theories
  - (most theory elements will be qualitative)
- Theories are fundamental for progress in software engineering methods
  - Once a theory has been validated, it tells you where the most progress can be made
  - and provides a framework of thinking for practitioners

4 Modeling something for prediction

Freie Universität Berlin

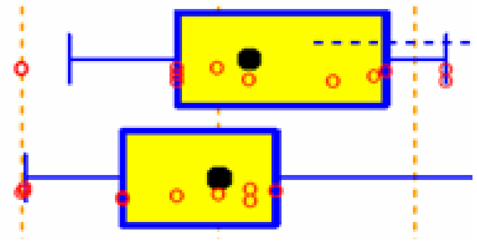


- You consider your data to be examples
  - inputs and outputs
- You want to find out how to predict output values given the input values
  - by multivariate statistics, machine learning
  - despite the automation, it involves a lot of manual analysis
- Prediction models are often much more complex than explanation models
  - because no interpretation of the model is required

- Rarely important for evaluation
- But useful for project management
  - cost estimation, scheduling, staffing, quality management etc.

5 Comparing something

- You have two or more "things" and want to compare them with respect to one or more attributes
  - which is larger, smaller, faster, ...



- This is a rather typical scenario in evaluation
  - in particular with experiments
  - but often also for surveys, benchmarking, etc.

- Notes
  1. Comparing is very similar to measuring
     - namely measuring a difference or ratio
  2. Comparing is the main realm of significance testing and confidence intervals

How to do it?

1. Exploring something
2. Measuring something
3. Modeling something for explanation
4. Modeling something for prediction
5. Comparing something

- **Next week**, we will shortly talk about some techniques for performing these tasks
- Today, we only wanted to understand
  - the tasks themselves
  - their differences

Quality criteria for data analysis

- Data analysis has to support the primary quality attributes of the empirical study overall:
  - credibility and relevance
- It can usually not do much for relevance
  - Exception: Exploratory quantitative studies
- To support credibility, the following properties are required. Data analysis must be
  - **correct**: Data has not been mis-collected nor mis-processed and we trust the analysis (and hence its results)
  - **illustrative ("anschaulich")**: It is easy to understand (a) what the results say, (b) how they came to be from the data
    - The analysis makes us understand the data itself
  - **informative**: The analysis reports results that are relevant and helpful for answering the study question

Problems with the quality attributes

- Correctness:
  - Complex analyses almost always require assumptions that can not be fully validated
    - e.g. normality or even just representativeness
    - → **understand the assumptions of your analyses!**
  - Weaknesses in the data may be pronounced by the analysis
- Illustrativeness:
  - "Most illustrative" is a different thing to different people
- Informativeness:
  - The more detailed and the more validated the results are, the harder to understand they tend to become
  - But difficulties in understanding reduce the informativeness
  - Hence, there is a tradeoff between precision and simplicity
    - Trade off very consciously! (Perhaps report in multiple formats)

Where are we?

Freie Universität Berlin

- Conclusions are not obvious
- Possible tasks of data analysis:
  - Exploring
  - Measuring, Comparing
  - Modeling for prediction
  - Modeling for understanding

- Quality criteria
- Steps:
  - make data available
  - validate
  - explore
  - analyse

We will now look at:

- these four tasks and
- some practical advice for performing them

Make data available

- Data can be collected in many different ways
  - by hand on paper
  - by hand with some collection software tool
  - automatically by some mechanism
  - or data may already exist

- Initially, the data is often not in a form directly suitable for the analysis software
  - May need encoding (e.g. anonymize personal information)
  - May need collection (when it comes from different sources)
  - May need collating (when it is distributed over many files)
  - May need syntactical processing (to match a target format)

Make data available (2)

- Manual data-collection work is error-prone
- Manual data-transformation work is error-prone
- Use automation whereever possible
  - Manual processes make *individual* mistakes: hard to catch
  - Automated processes make *systematic* mistakes: can be found by testing
    - Starting over from scratch is cheap!
  - Furthermore, automation scripts serve as an audit trail
- Double-check your data whenever possible

Make data available (3): Encoding

- Principles:
  - Remove superfluous information
    - Keep only relevant (or potentially relevant) information
    - This helps avoid confusion-based mistakes
  - Choose analysis-friendly representations
    - Perhaps encode redundantly in more than one form
  - Redundancy helps find many kinds of mistakes

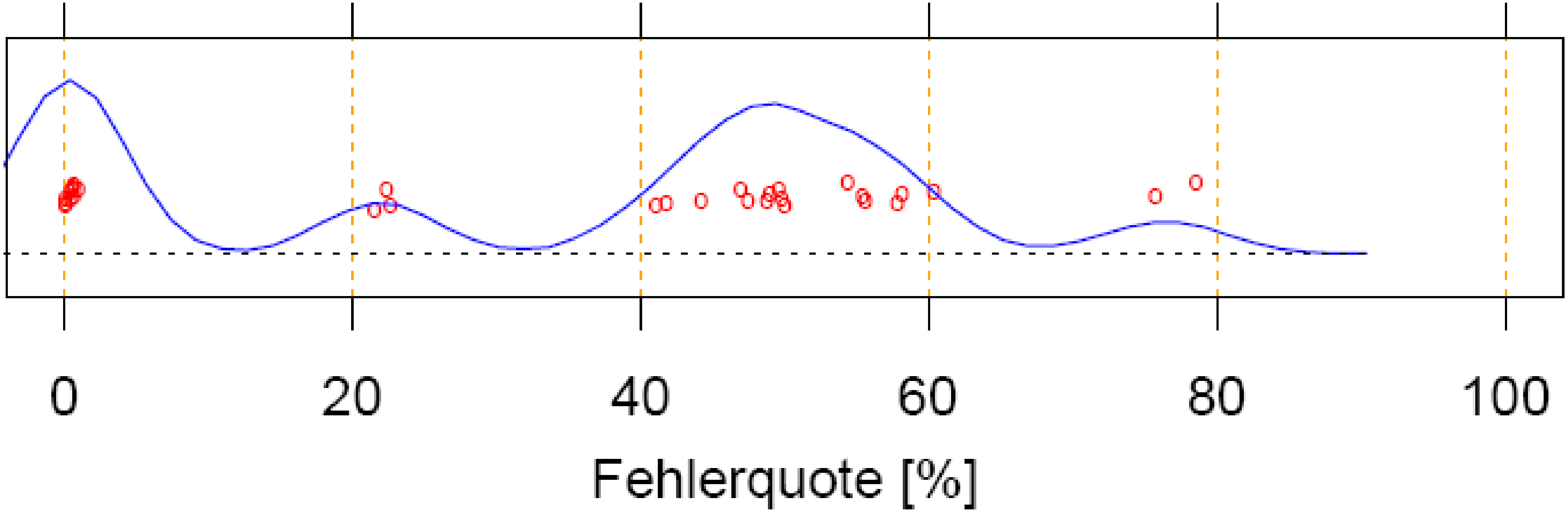
- Making data anonymous:
  - Just removing all names etc. makes many analyses impossible
    - because many relationships between data records are lost
  - Better solution: Pseudonyms
    - Consistently replace Meier, Müller, Huber, Schmidt by subj1, subj2, subj3, subj4
    - Then throw away the mapping
    - Or collect your data in this form right from the start

Validate data

- Typical problems when making data available:
  - Some data is lost
  - Some data is corrupted
  - Some data is confused or mis-labeled
- In the validation step, we try to recognize these events:
  - Always compare actual and expected data counts
  - Check for impossible or unlikely values
    - continuous data: very low, very high values
    - discrete data: very frequent, very rare values
  - Check the consistency of any redundant information
    - Having redundancy is a very good idea!
  - Hand-check a few random data points against the earliest possible form of the data source they come from
    - and keep checking if any problems were found
      - errors tend to cluster; repairing errors may introduce new ones

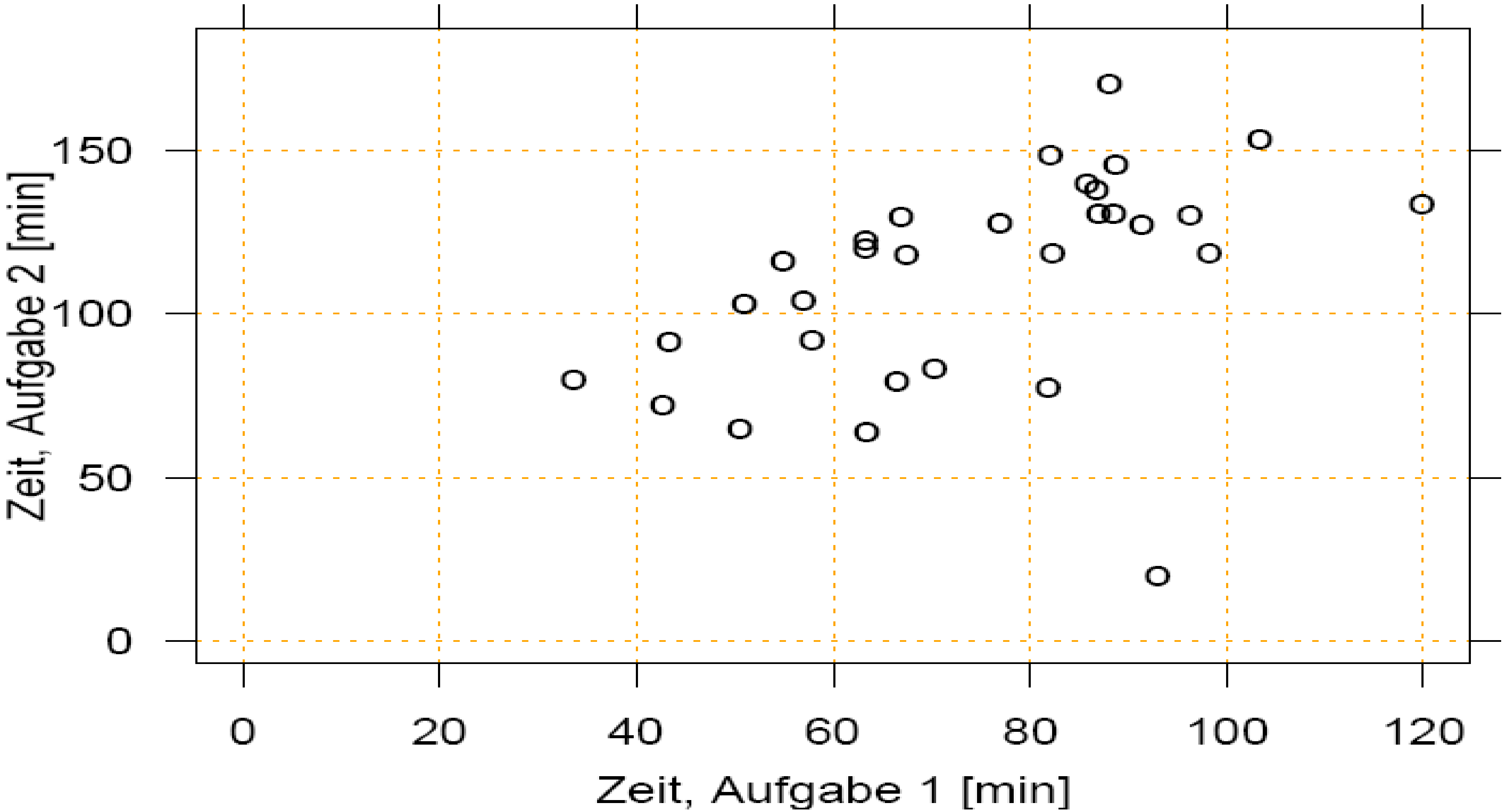
Validation: example

- Are these data correct?
  - they were computed and typed-in by two persons
    - half each
- If not, what may have gone wrong?



Validation: example (2)

- Are these manually collected data correct?
  - What would you do to find out?



Explore data

- Get an overview of the whole dataset
- Look at individual variables
- Look at pairs of variables
- Quick-check specific expectations, if any

- We use an example data set to illustrate the ideas:
  http://www.tpc.org/tpcc/results/tpcc_results.txt (as of 2004-04)
  - TPC: Transaction Processing Council
  - tpmC: Transactions-per-minute (type C)
    - an RDBMS benchmark
  - The data set *tpc* used here is *tpcc_results.txt* after a number of encoding steps

- Concrete commands for the steps in R syntax are shown

Explore:
Get an overview of the dataset

- Is all data in one set or are there multiple connected sets?
  - e.g. one set describing experimental subjects and another containing four records of observations for each subject
  - If there were multiple data sets, we would need JOIN operations for some analyses (like in a relational database)
    - merge() in R; in this case we do not

- How many observations are there?
  - tpc = read.delim("tpcc_results.txt")
  - nrow(tpc) → *127*

- Which variables of which types are there?
  - names(tpc) → *tpmC, dollarPerTpmC, cpus, frontEnds, cputype, freq, ostype, tpmon,* and several others
  - sapply(tpc, class) → *tpmC:numeric, cputype:factor*, etc.

Types of variables

- The R type of a variable is related to its scale type:

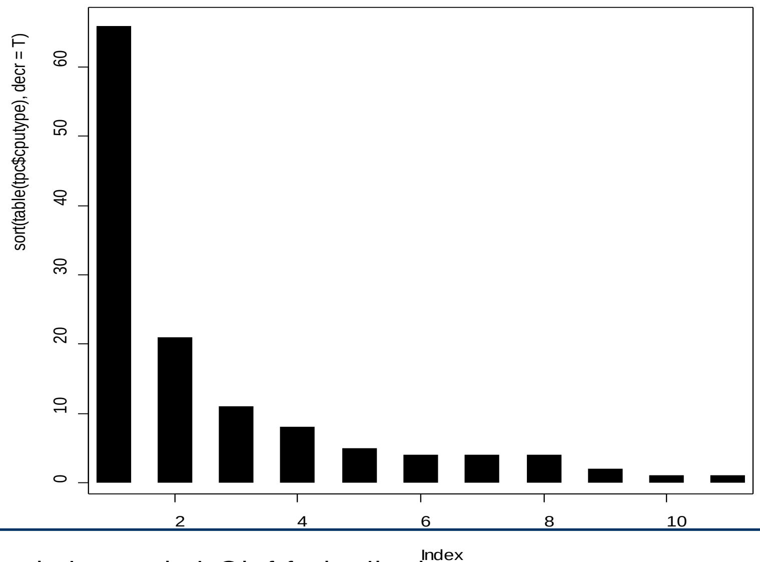
- Nominal variables:
  - *factor* (or *logical*)
  - often used to segment the dataset into parts
- Ordinal variables:
  - *ordered* (a special kind of *factor*)
- Variables on difference scales:
  - *numeric*
  - for times and dates: *POSIXct* etc.
- Variables on ratio scales:
  - *numeric*

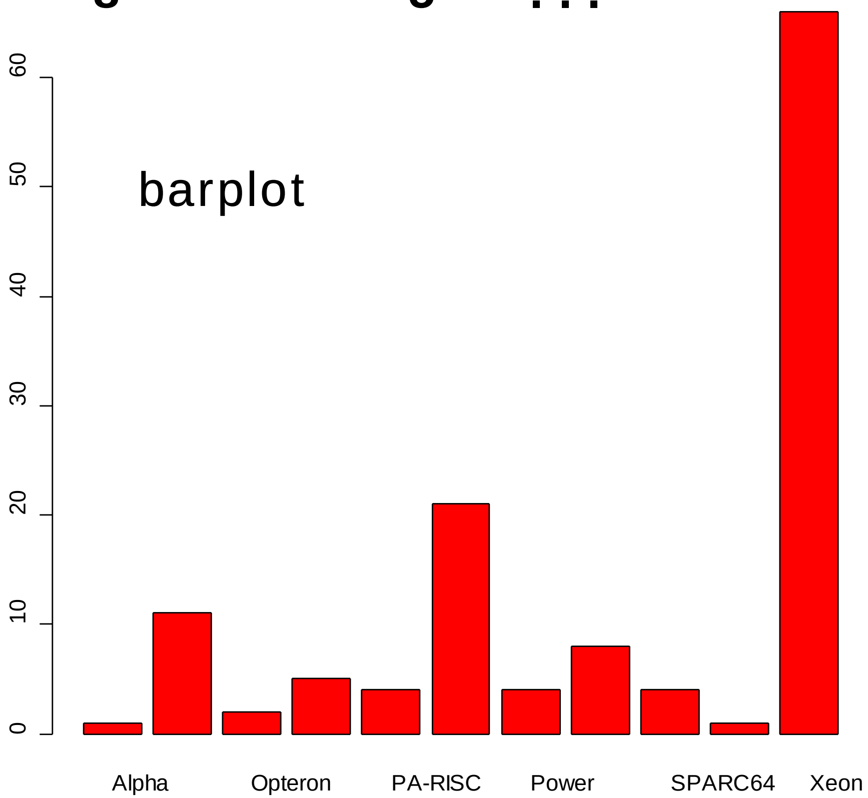
Explore: Look at individual variables

- For nominal and ordinal data:
  - Review levels and frequencies of the factor
  - e.g. sort(table(tpc$cputype)):

```
 Xeon    Pentium3    Itanium2  RS64       other ...
   66          21          11     8           5   ...
```

  - Perhaps visualize the proportions graphically
  - e.g. plot(tpc$cputype)





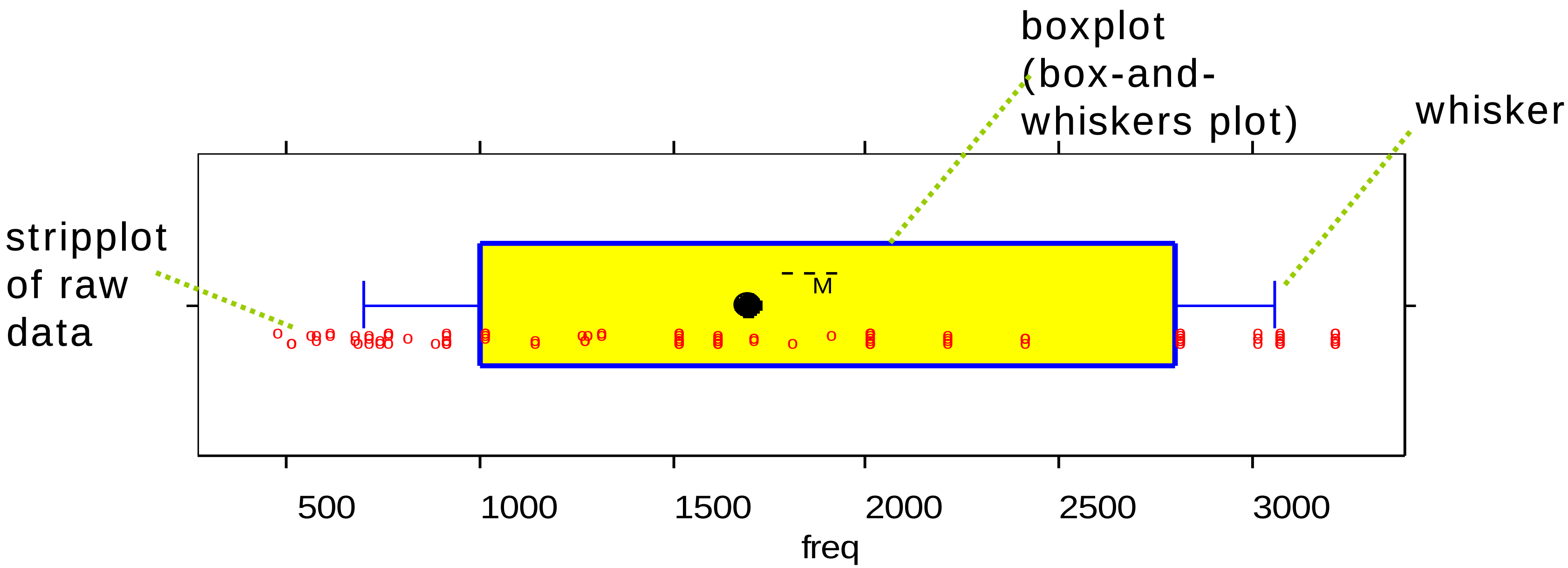
Freie Universität Berlin

Explore: Look at individual variables (2)

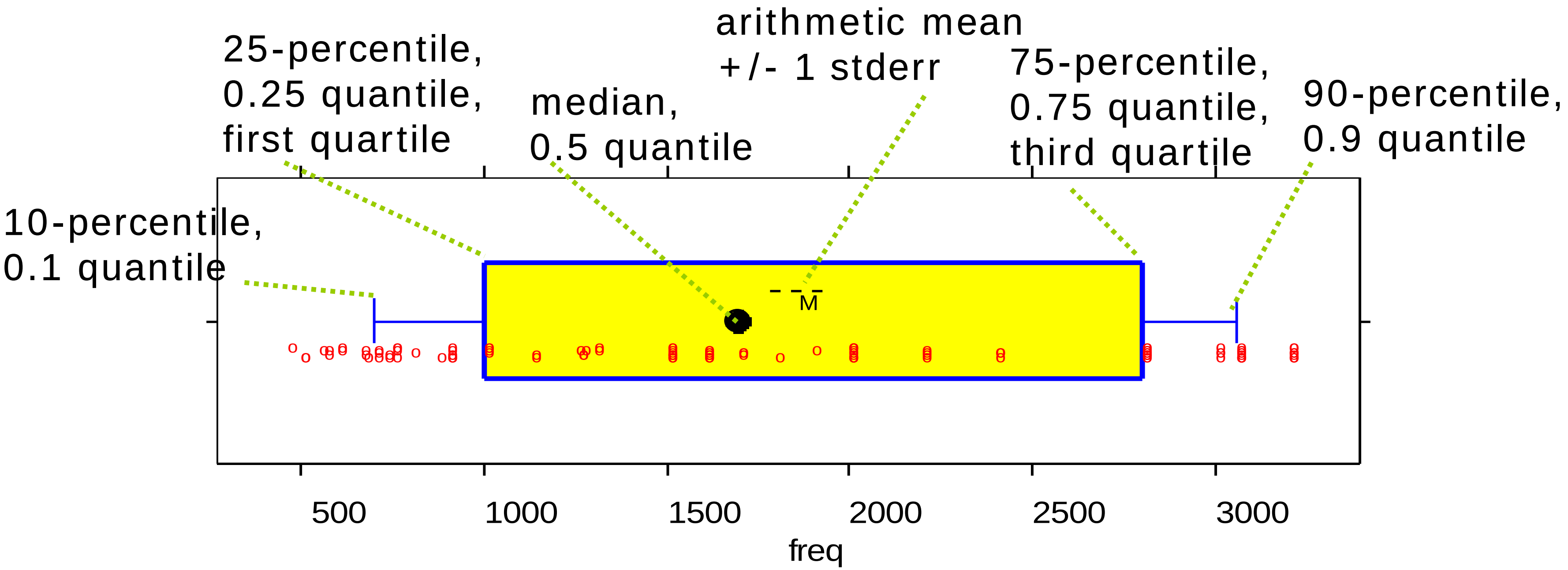
- For numerical data: Review the distribution
  - Numerically as a summary
    - e.g. summary(tpc$freq):

```
Min. 1st Qu.  Median   Mean 3rd Qu.    Max.
464    1000    1700    1860    2800    3200
```

  - Graphically as a barplot, stripplot, boxplot, density plot or combination thereof

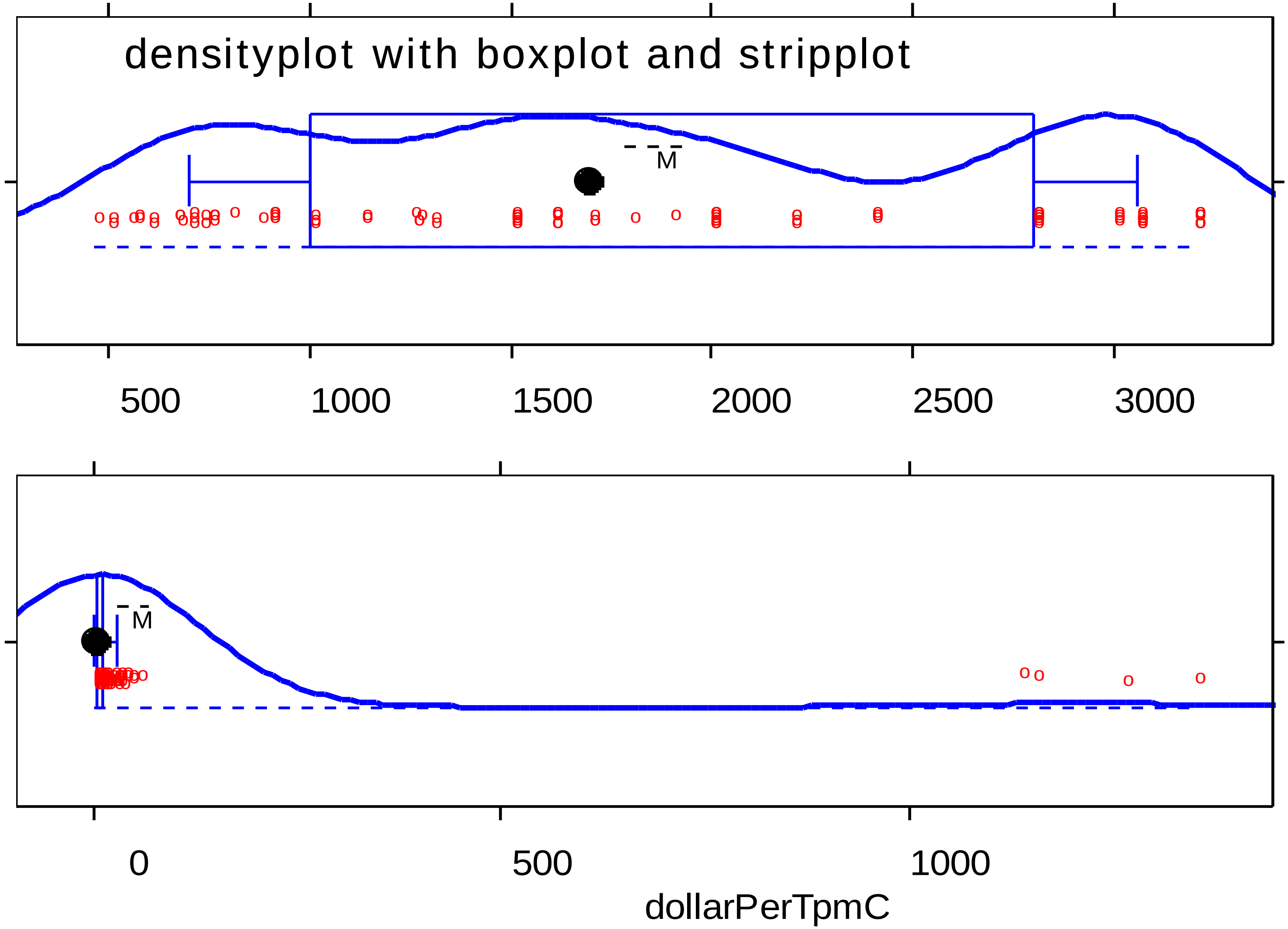


Boxplot explained



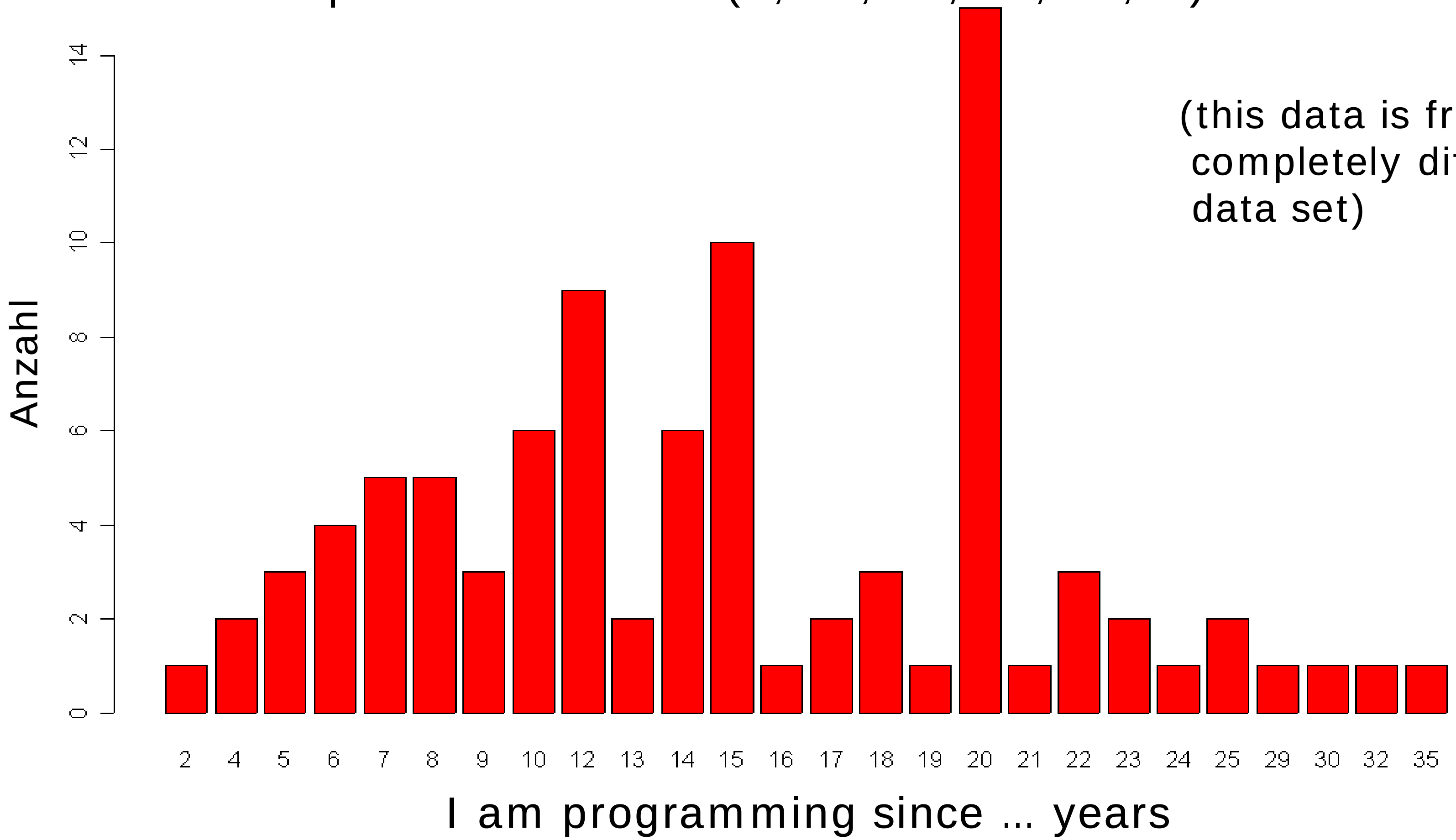
- The above is a flexible, non-standard type of boxplot
  - library(agsemisc); bwplot(..., panel=panel.bwstrip)
- Conventionally, whiskers extend up to 2 iqr beyond the box
  - and end at a data point; iqr: interquartile range (box width)
- Conventionally, stripplot and meanplot are missing
  - except for "outliers" (data values beyond the whiskers)

Explore: Look at individual variables (3)



Explore:
Look at individual variables (4)

- Look for "unnatural" phenomena:
  - in this case: "round" numbers (10, 12, 15, 20) are suspiciously more frequent than others (9, 11, 13, 19, 21, ...)

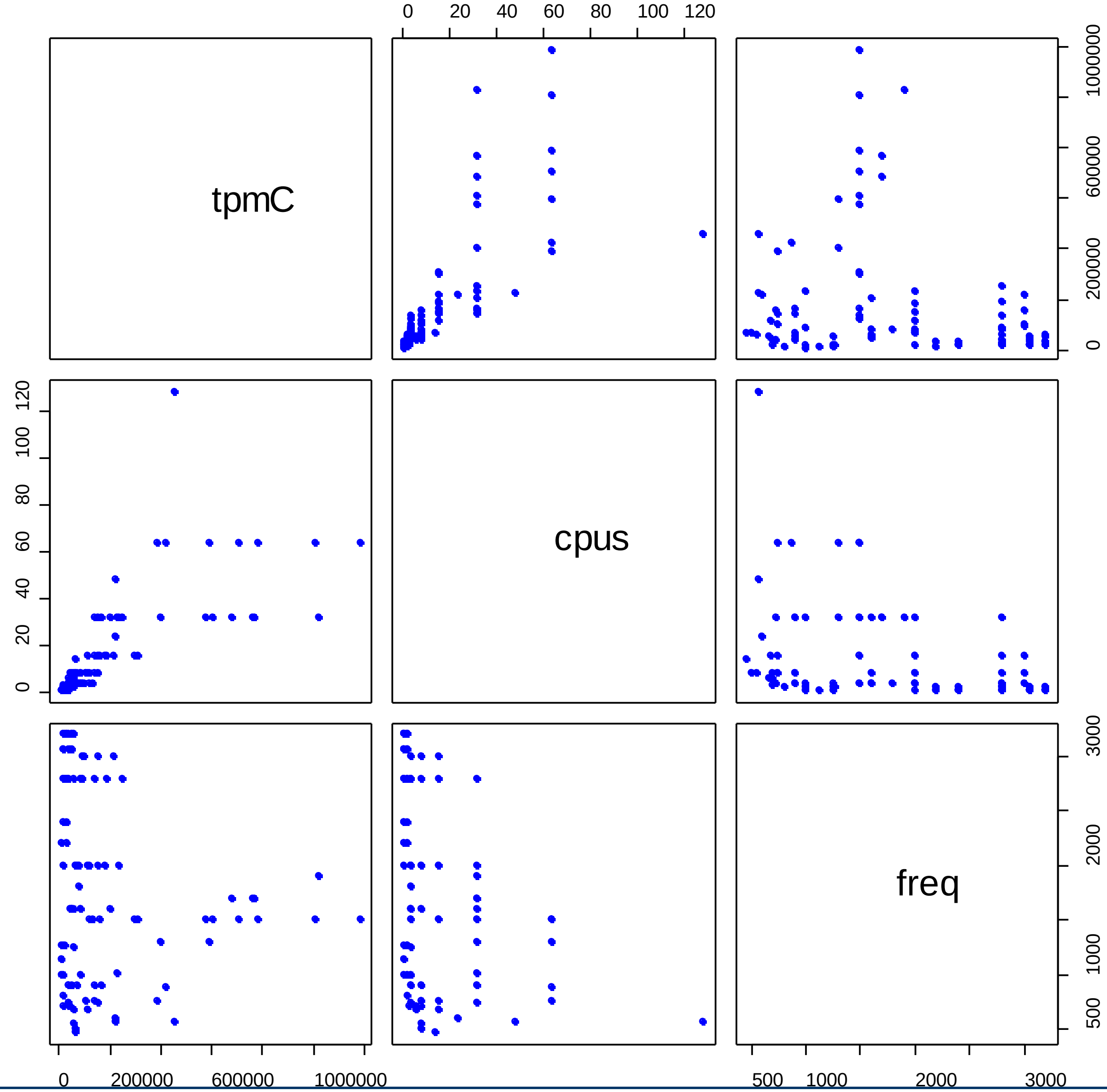


(this data is from a completely different data set)

Explore: Look at pairs of variables

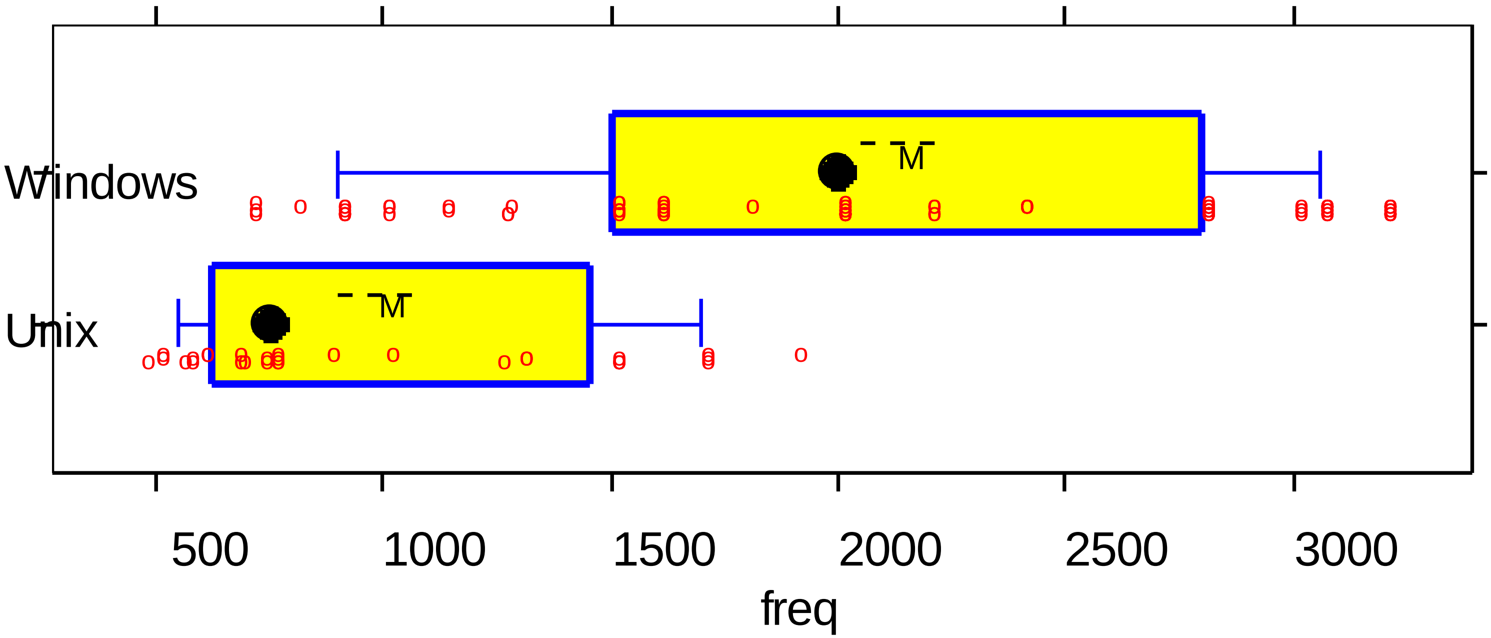
- For the numerical variables only:
  - a scatter plot matrix

```
x = tpc[ ,c("tpmC",
  "cpus", "freq")]
  pairs(x)
```



Explore:
Quick-check specific expectations

- expectation: "The Unix machines have lower clock rates"
  - (because of their RISC architectures at that time)
  - Approach: Comparative boxplots

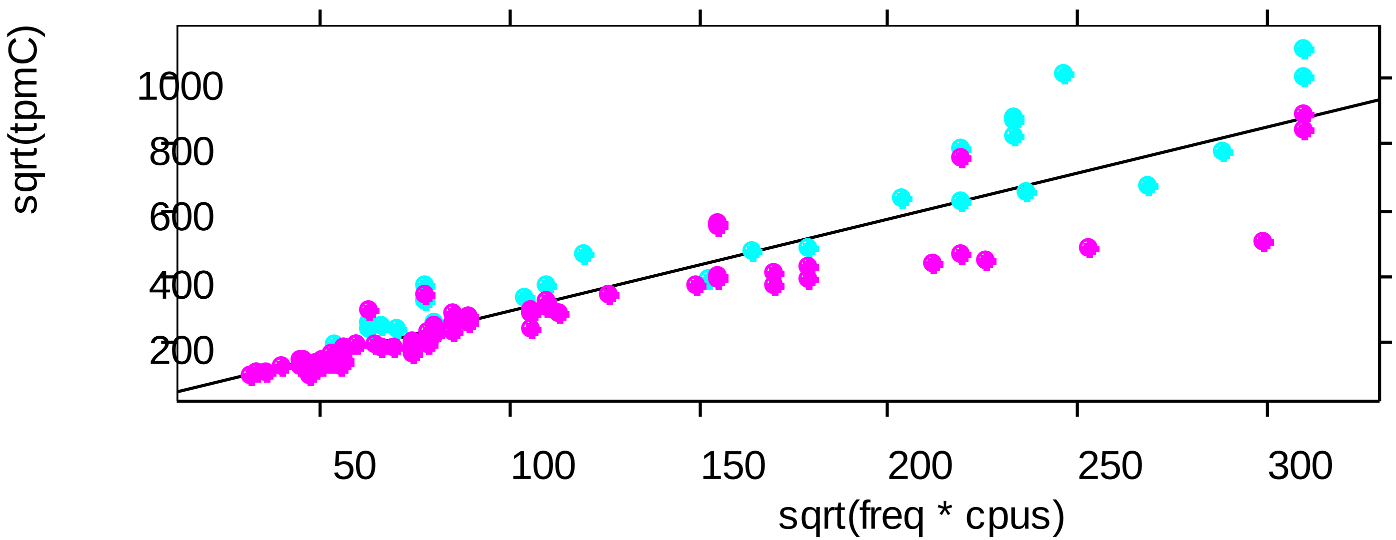


Explore:
Quick-check specific expectations (2)

Freie Universität Berlin

- expectation: "Faster clocks and more CPUs lead to proportionally higher tpmC"
  - Approach: Scatterplot of tpmC versus freq*cpus
  - `xyplot(tpmC ~ freq*cpus, data=tpc)`
  - `xyplot(sqrt(tpmC) ~ sqrt(freq*cpus), data=tpc, groups=ostype, panel=function(...) { panel.lmline(...); panel.superpose(...) } )`

Unix
Windows



Perform analysis: measure, model, or compare

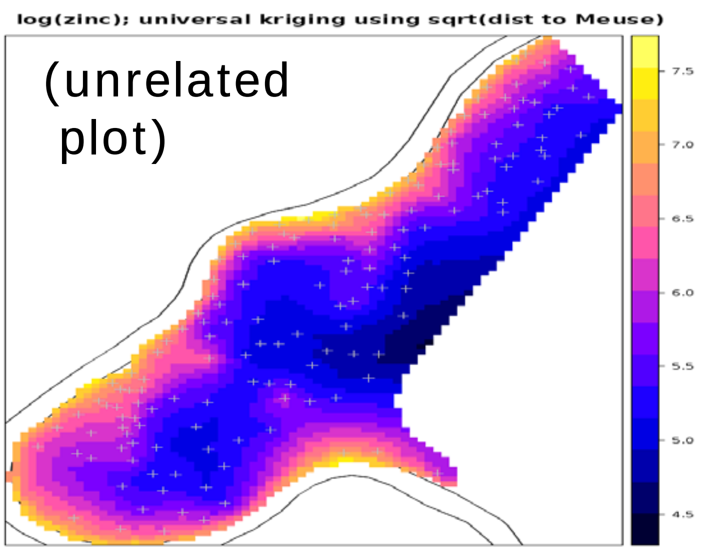
- In principle, exploration and analysis both use the same techniques
  - Exploration tends to prefer graphical visualization techniques
    - e.g. a scatterplot
    - Because they are quick to review and understand
    - Because unexpected characteristics are easily seen
  - Analysis tends to prefer quantitative, numerical techniques
    - e.g. a correlation coefficient
    - ...because they are "more precise"
      - Attention: The precision can be misleading!
    - ...because they focus on one aspect
      - but that can be a disadvantage, too. Use visualization as well.

- Techniques will be covered in more detail in next week's presentation

A note on plotting in R

R has three groups of plotting operations:

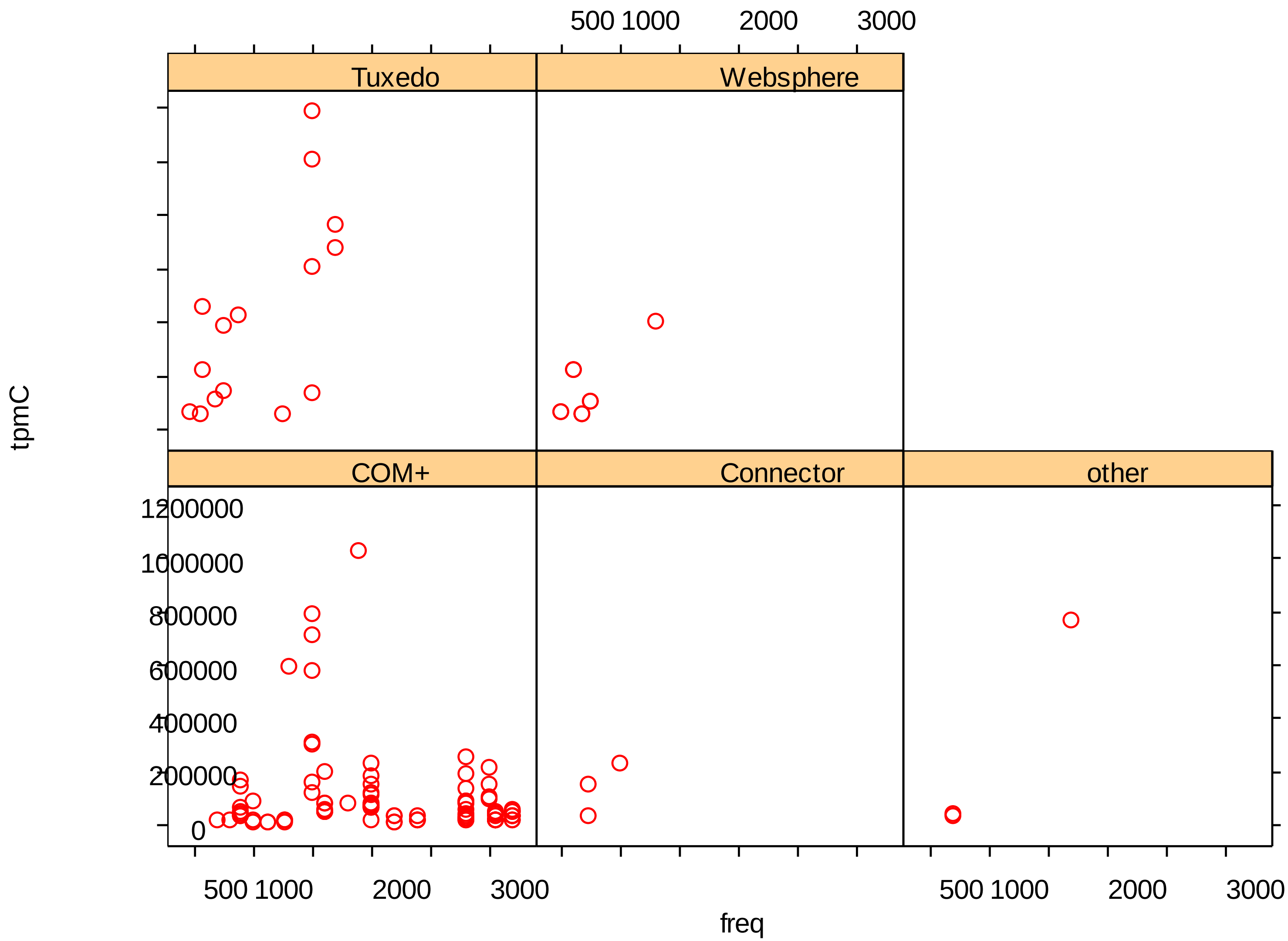
- The **basic** group consists of high-level operations for producing complete plots
  - *plot, boxplot*, and others
- and low-level operations for adding to plots
  - *points, lines, text* etc.



- The other is known as ***Lattice*** (formerly *Trellis*) and uses high-level operations for producing complete plots
  - *xyplot, bwplot*, etc. (*panel.xyplot* etc. do the actual work)
- and low-level operations for use in panel functions
  - *lpoint, llines, ltext* etc.
- Lattice specializes in producing many plots at once:
  - *library(lattice); xyplot(tpmC~freq|tpmon, data=tpc)*
- The third one, called **grid**, is the basis for Lattice
  - very flexible, cumbersome

furthermore: **ggplot2**

"lattice" example:
xyplot(tpmC~freq|tpmon, data=tpc)



Thank you!