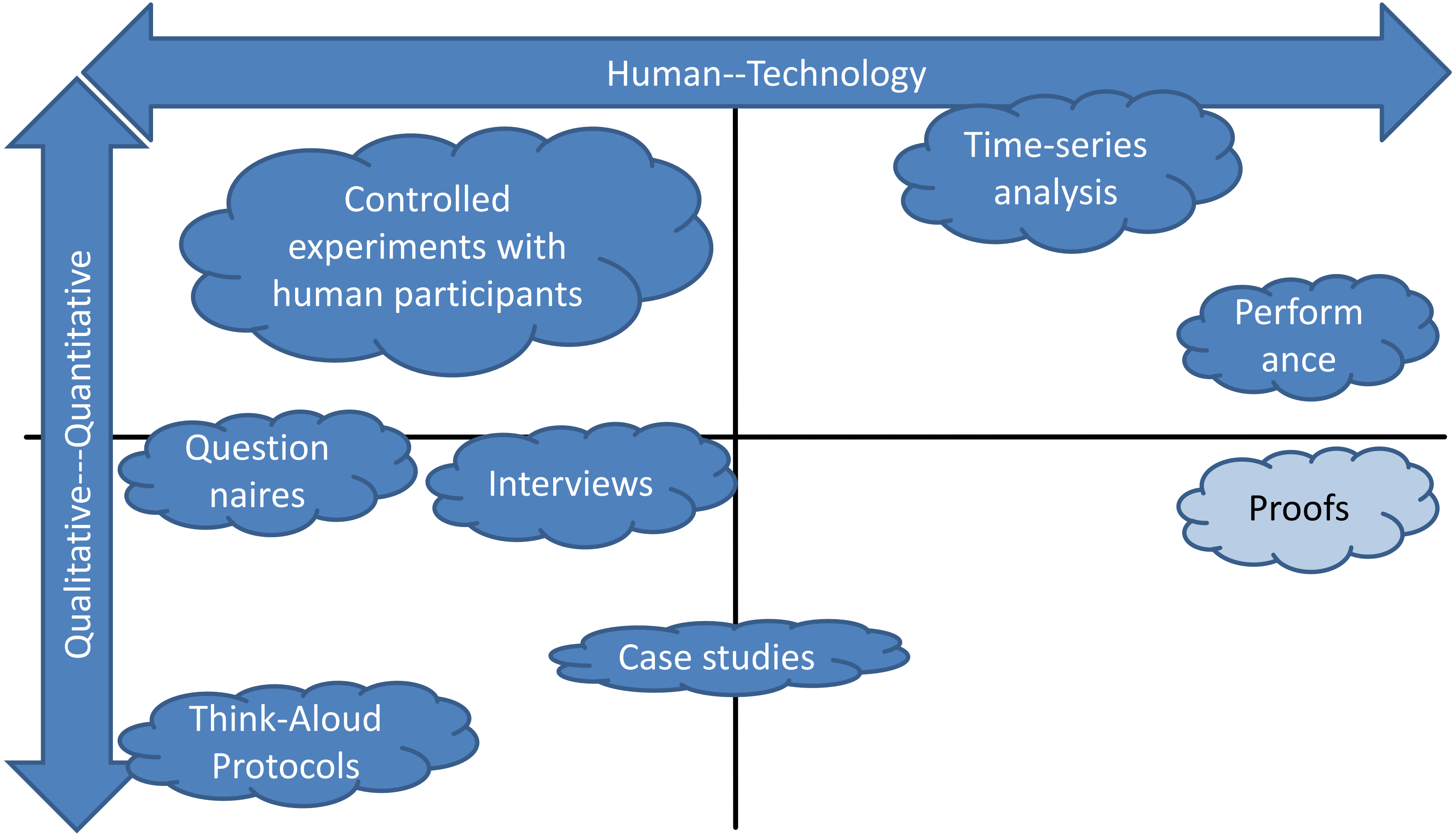
Empirical Software Engineering

Performance Messungen

Overview


Human--Technology
Qualitative---Quantitative
Controlled experiments with human participants
Time-series analysis
Perform ance
Question naires
Interviews
Proofs
Case studies
Think-Aloud Protocols

Task

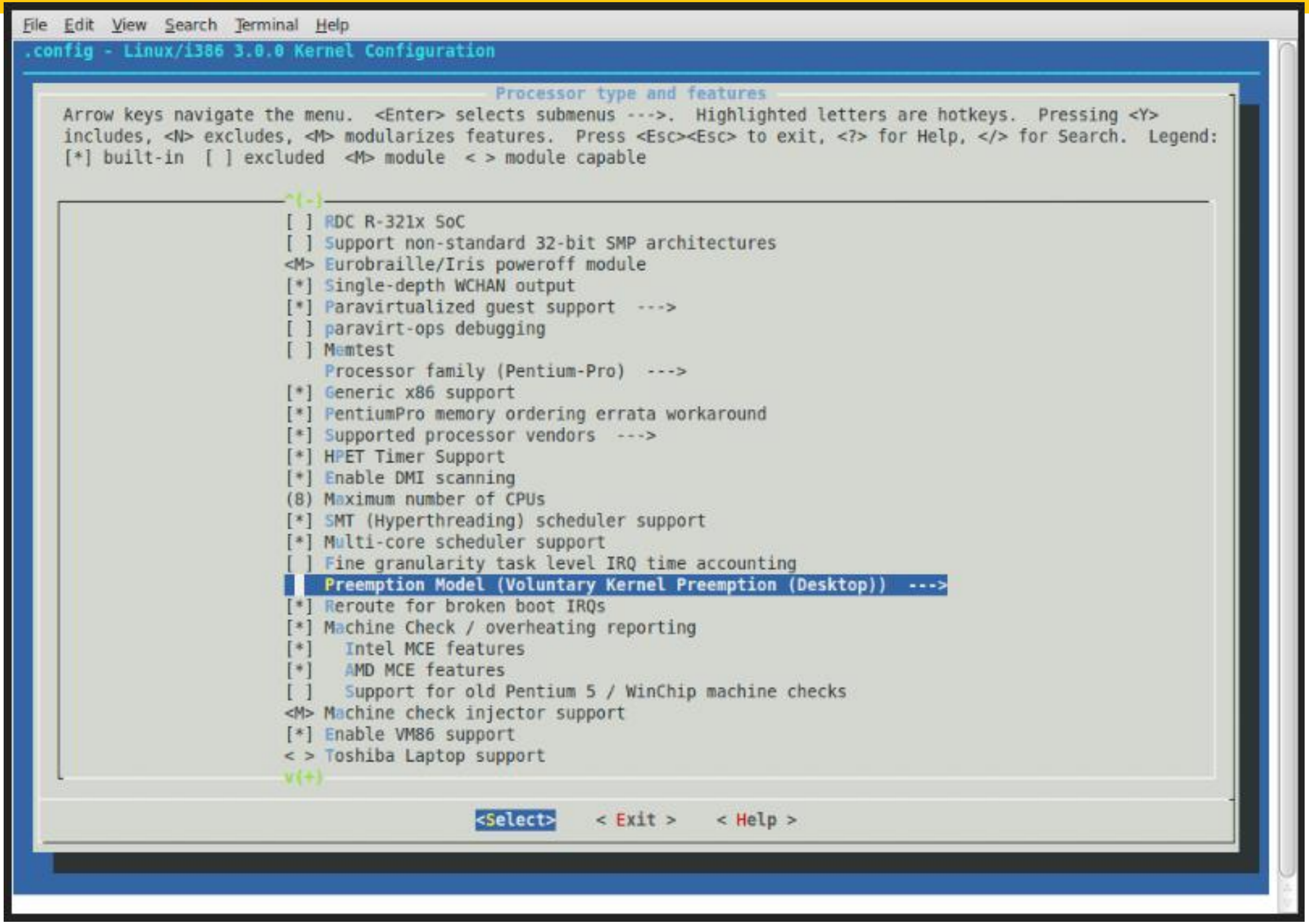
- Determine the fastest sorting algorithm
  - Group 1: Merge sort vs. quick sort (http://rosettacode.org/wiki/Sorting_algorithms/Quicksort#Java)
  - Group 2: Mergesort recursive vs. Mergesort iterative
  - Group 3: Mergesort Java vs. Mergesort Python
  - Group 4: Mergesort Java vs. Mergesort C
  - Link to all merge sorts: https://www.geeksforgeeks.org/iterative-merge-sort/
- Put the results on a poster
- Do you trust the results of your colleagues?

Goals

- Understand difficulties of performance analyses
- Evaluate performance analyses
- Get a first impression of statistical tests

Why Performance Analysis?

- Compare alternatives
- Understand influence of a configuration option
- System tuning
- Understand relative performance (over time)
- Understand absolute performance for single case
- Set expectations (e.g., min/optimal system requirements for PC games)
- Analyze system behavior


File Edit View Search Terminal Help
.config - Linux/i386 3.0.0 Kernel Configuration
Processor type and features
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
^(-)
[ ] RDC R-321x SoC
[ ] Support non-standard 32-bit SMP architectures
<M> Eurobraille/Iris poweroff module
[*] Single-depth WCHAN output
[*] Paravirtualized guest support --->
[ ] paravirt-ops debugging
[ ] Memtest
Processor family (Pentium-Pro) --->
[*] Generic x86 support
[*] PentiumPro memory ordering errata workaround
[*] Supported processor vendors --->
[*] HPET Timer Support
[*] Enable DMI scanning
(8) Maximum number of CPUs
[*] SMT (Hyperthreading) scheduler support
[*] Multi-core scheduler support
[ ] Fine granularity task level IRQ time accounting
Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
[*] Reroute for broken boot IRQs
[*] Machine Check / overheating reporting
[*] Intel MCE features
[*] AMD MCE features
[ ] Support for old Pentium 5 / WinChip machine checks
<M> Machine check injector support
[*] Enable VM86 support
< > Toshiba Laptop support
v(+)
<Select> < Exit > < Help >

Analysis Techniques

- Measurement
  - No simplifying assumptions
  - Most trustworthy results
  - Inflexible, one selected system
- Simulation
  - Abstraction
  - Flexible
- Analytical modeling
  - Mathematical description of system
  - Strong abstraction, results are often unrealistic
  - Especially for early validation

Benchmark

- Exectute existing programs on existing hardware components in realistic environment (no simulation)
- Measure performance, memory consumption, etc.
- Can be automated
- No human influence

Benchmark - Examples

- 3DMark (Graphics chip)
- TCP-H (Datawarehouse)
- TCP-C (On-line transaction processing)
- Sintel (Video encoder)

What Can We Measure?

- Execution time
- CPU cycles
- MIPS (Million instructions per second)
- MFLOPS (Million floating-point operations per second)
- SPEC (System Performance Evaluation Cooperative)
- QUIPS (Quality improvements per second)
- Transactions per second

- What criteria should a good metric fulfill?
- Are the presented metrics good metrics?
  - Execution time
  - CPU cycles
  - MIPS (Million instructions per second)
  - MFLOPS (Million floating-point operations per second)
  - SPEC (Standard Performance Evaluation Cooperative)
  - QUIPS (Quality improvements per second)
  - Transactions per second

Criteria

Criterion	Execution time	CPU Cycles	MIPS	MFLOPS	SPEC	QUIPS	Transactions/second
Linearity	+	-	-	+	-	+	+
Reliability	+	-	-	-	-	-	+
Repeatability	+	+	+	+	+	+	+
Easy to measure	+	+	+	+	+	+	+
Consistency	+	+	-	-	+	+	+
Independence	+	+	+	-	-	+	+

Confounding Parameters

- Influence measurement result systematically or unsystematically
- Examples:
  - Background processes
  - Differences in hardware
  - Differences in temperature
  - Input data, random?
  - Heap size
  - System interrupts
  - Parallel execution in single/multicore systems
  - Garbage collector

- How can we control the influence of a confounding parameters

Typical: Best Measurement

- Repeat measurement
- Best, second best, or worst measurement

- Bsp: Execution time
- R: Read einlesen
  - data <- read.csv("rt.csv", header=TRUE, sep = ";", dec = ".")
  - header: Do variables have heading
  - sep: Separator for data entries
  - dec: Decimal point/comma
  - rt <- data[,’time’]
  - min(rt)/max(rt)

Arithmetic Mean

- Repeat measurement
- Compute mean:

$$\bar{x}_{arithm} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \ldots + x_n}{n}$$

- R:
  - mean(rt)

Median

- Value that is in the middle
- Robust against outliers
- R:
  - median(rt)

- Even number of measurements:
  - Arithmetic mean of the two middle values
  - Use one of the two middle values

Median or Arithmetic Mean?

- Median, if:
  - Ordinal Data*
  - Few measurement values
  - Non-normal distribution
  - Outliers

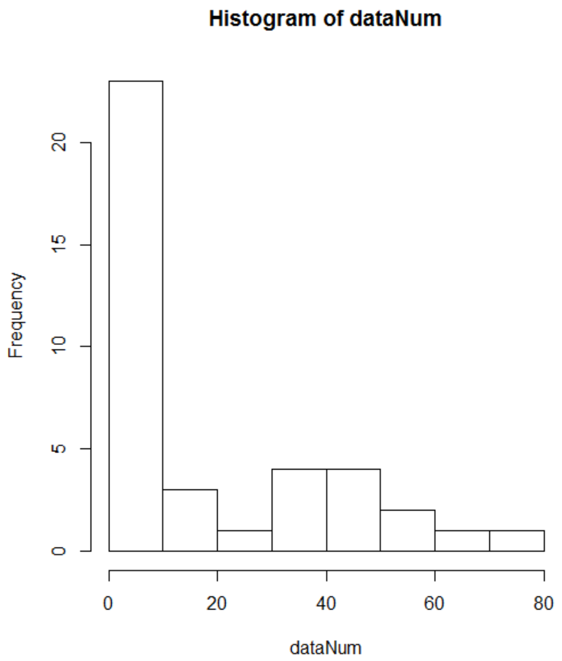
- *Scale types
  - Nominal (z.B. Gender)
  - Ordinal (z.B. Ranking)
  - Metric (z.B. Temperature, response time)

Look At Data

- Go swim in the data!
- Get an overview
- Look at how data are distributed
- Are there outliers?

Histograms

- Frequency of measurement values in fixed buckets

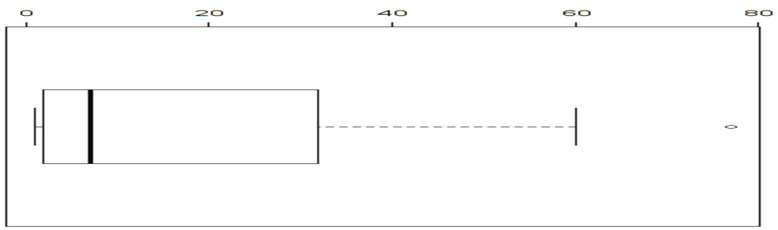


- R
  - rtNum <-as.numeric(unlist(rt))
  - hist(rt)

Boxplots

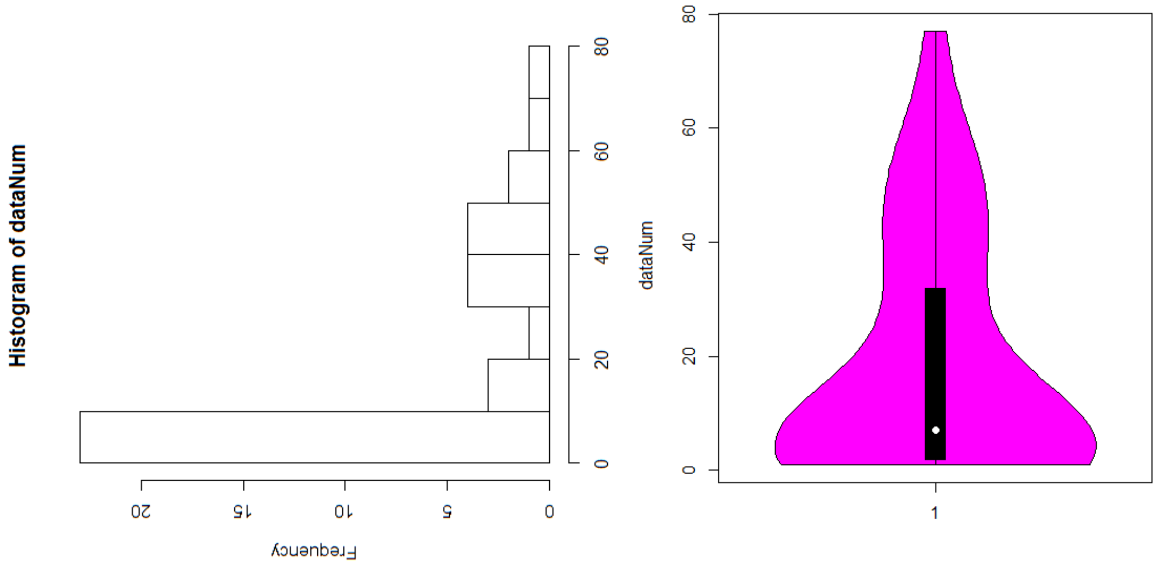
- Median as thick line
- Quartiles as box (50% of all values within the box)
- Whiskers (-> box and whisker plots)
- Outliers as dots

- R: boxplot(rt)



Violin-Plot

- Like box plots, but show additionally the distribution of data
- R:
  - install.packages("vioplot")
  - library(vioplot)
  - vioplot(rtNum)



Measurement Model

- $y = \tau + \varepsilon$

- y: Observed value
- τ: True value
- ε: Error

- Population: greek letters
- Sample: german letters

Error Model

- True mean: 10

- 1 random error, influence of +/- 1
- Measurement: 9 (50%) and 11 (50%)

- 2 random errors, each +/- 1
- Measurement: 8 (25%), 10 (50%), 12 (25%)

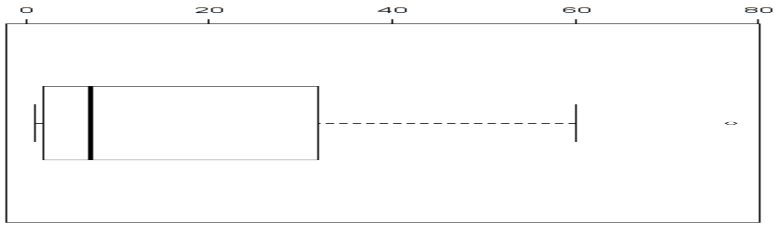
- 3 random errors, each +/- 1
- Measurement: 7 (12.5%), 9 (37.5%), 11 (37.5%), 13 (12.5%)

- N random errors, each +/- 1
- Normal distribution

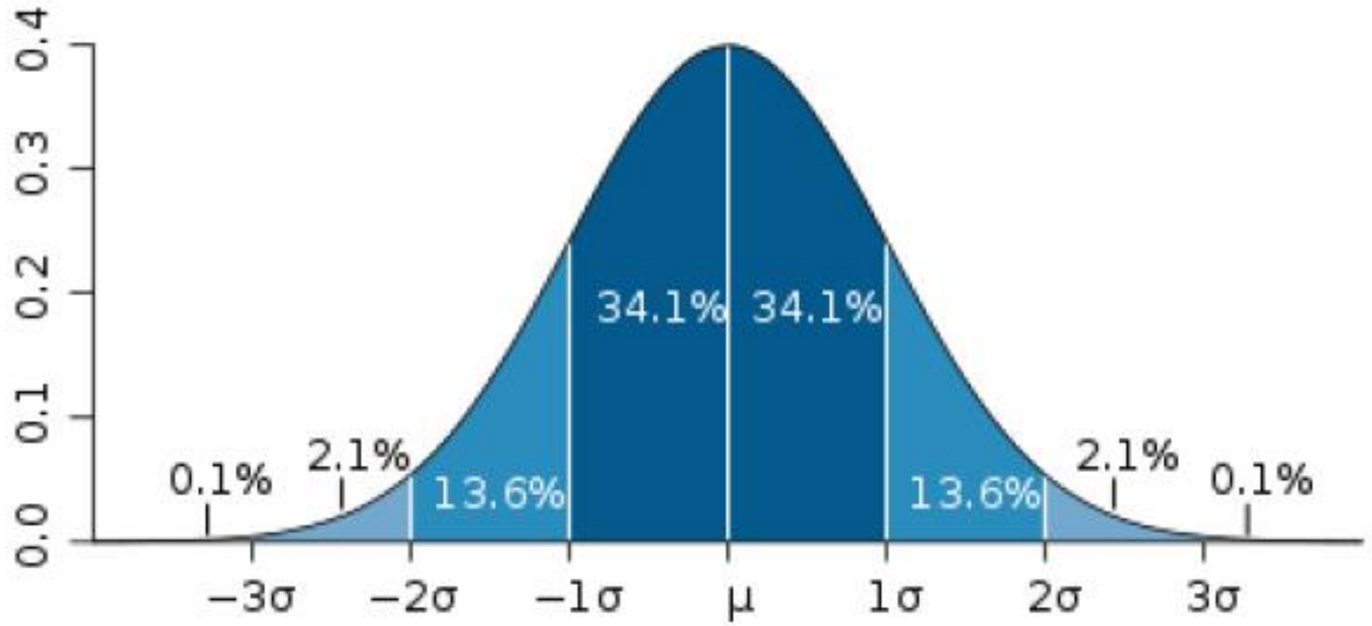
bution

Dispersion

- Mean: 45.55
- Boxplot



Standard Deviation



Standard Deviation

- R:
  - sd(rtNum)
  - 21.55
- Mean: 45.55

- 45.55 - 21.55 = 24 –> 45.55 (34 % of measurement values)
- 45.55 + 21.55 = 67.1 (34% of measurement values)

Use cases for Standard Deviation

- Define outlier
- Define giftedness
- Announce the discovery of the Higgs-Boson

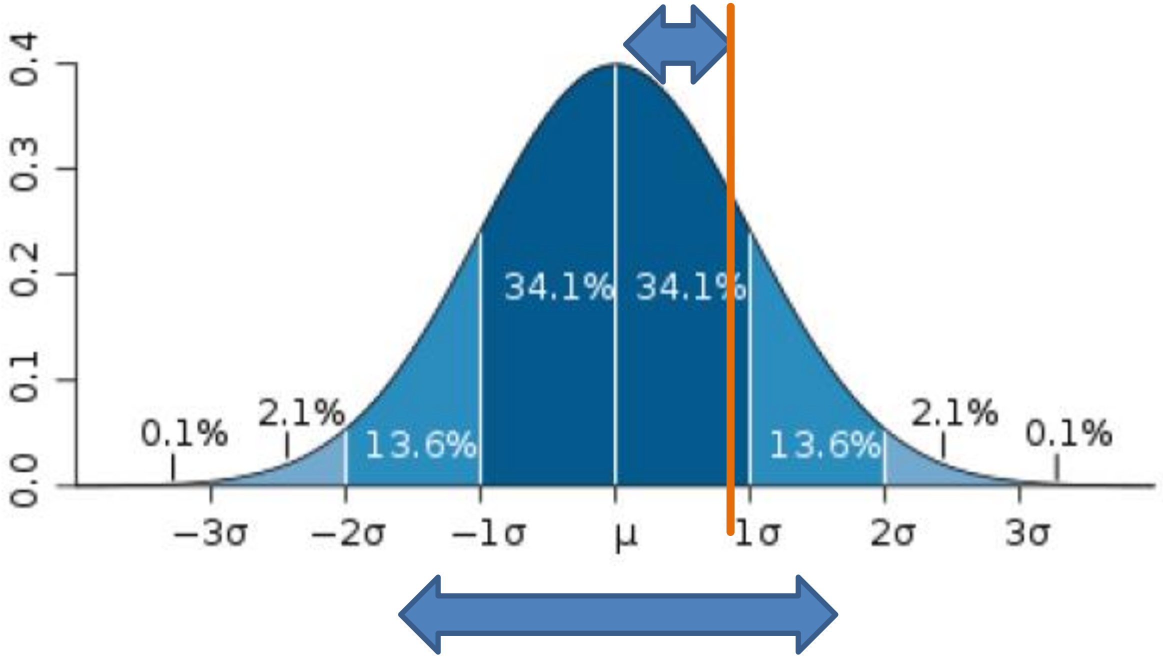
Variance

- Is the squared standard deviation

Accuracy vs. Precision

Accuracy:
Deviation of observed mean from true mean

Important when measuring response time



Precision:
Dispersion around mean

Cause of measurement errors is unclear

Random vs. Systematic Errors

- Systematic errors: Errors of the epxeriment/measurement methods
  - CPU speed: measurement during different tempertatures
  - State not resetted for second measurement
  - Low variance, or constant variance for all measurements
  - Need to be excluded during design, which requires practice and experience

  → Affect accuracy
- Random errors:
  - Cannot be controlled
  - Requires statistical methods

  → Affect precision

Significance Tests

- To evaluate whether an observed result appeared rather randomly or not

T-Test

- Designed by Student (William Sealy Gosset)
- Comparision of two measurements

Null hypothesis (H_0)	**Alternative hypothesis (H_1)**
Statistical hypotheses	
Measurements do not differ, i.e., they come from the same population	Data of both measurements are from different populations
Formal: $H_0 : \bar{x}_1 = \bar{x}_2$	Formal: $H_1 : \bar{x}_1 \neq \bar{x}_2$

T-Test: Result

- Determines probability of observed result, under the assumption that the null hypothesis is valid -> conditional probability
- If probability is smaller than:

  very very significant
  very significant
  typical significance level
  for exploraty/initial studies

  - 0.001
  - 0.01
  - 0.05
  - 0.10

  null hypothesis must be wrong
- Significance level must be defined in advance!

T-Test: Conclusion

- What does significant result mean?
- Is null hypothesis incorrect? -> No
- Is alternative hypotheses correct? -> No

- There is no evidence that the null hypothesis is valid (thus, I can only make statements about the null hypothesis)
- Writing a report:
  - Reject/could not reject null hypothesis
  - Never: Confirmation of null or alternative hypothesis

T-Test by Hand (1)

- Computation of test value

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\hat{\sigma}_{(\bar{x}_1 - \bar{x}_2)}}$$

rt.csv:
t = 1.522

$$\hat{\sigma}_{(\bar{x}_1 - \bar{x}_2)} = \sqrt{\frac{\sum_{i=1}^{n_1}(x_{i1} - \bar{x}_1)^2 + \sum_{i=1}^{n_2}(x_{i2} - \bar{x}_2)^2}{(n_1 - 1) + (n_2 - 1)}} \bullet \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

T-Test by Hand (2)

- Degrees of freedom, df
  - for t-Test: $n_1 + n_2 - 2$ (in this example: 11)
- Table with t distribution (e.g., wikipedia)

$$t_{\alpha/2, df=11} = 2{,}201$$

- Comparison with calculated value (t_{emp} = 1.522)
  - is $t_{emp} > t_{\alpha/2, df=11}$?
  - no, so not significant

One-tailed vs. Two-tailed

- Two-tailed:
  - No assumption about direction of effect (e.g., which of two UIs is more usable
  - Compute half of significance level
- One-tailed:
  - Assumption that one UI is more usable
  - No need to cut significance level in half

$$t_{\alpha,df=11} = 1,796$$

T-Test: R

- t.test(rt1, rt2)
- Output:

```
Welch Two Sample t-test

data:  dataPC1 and dataPC2
t = 1.5222, df = 10.566, p-value = 0.1573
alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:
 -5.095727 27.583584

sample estimates:
mean of x mean of y
 50.74243  39.49850
```

- p value: conditional probability of having observed result under the assumption that nul hypothesis is valid
- If p value is smaller than defined significance level, result is significant and null hypothesis can be rejected

Types of errors

Decision

H_1

β error;
Type-2 error

T-Test: Variants

- T test for independent samples:
  - Creation of samples must not be dependent on each other
  - E.g., random assignment of participants to one or the other sample
- T test for dependent samples:
  - Creation of samples depends on each other
  - E.g., in a within-subjects design, or when spouses are distributed to different samples

T-Test: Prerequisits

- Metric scale type
- Normally distributed data (e.g., Shapiro-Wilk)
- Or: sample size >= 30

Mann-Whitney-U

- Non-parametric test
- Ordinal data (or non-normal distributed metric data)
- Computation of test:

$$U = n_1 \bullet n_2 + \frac{n_1(n_1+1)}{2} - T_1$$

  - r_i : Ranks in the sample

$$T = \sum_{i=1}^{n} r_i$$

Goals

- Understand difficulties of performance analyses
- Evaluate performance analyses
- Get a first impression of statistical tests

Literature

- David Lilja. *Measuring Computer Performance: A practitioner's guide.* Cambridge University Press. 2000.
- Performance-Paper
- Beliebiges Statistikbuch

Assignment

- Read excerpts of the following papers:
  - How Do Professional Developers Comprehend Software? (Section II, skim Section III)
  - An Experiment About Static and Dynamic Type Systems (Section 4, skim Section 5)
- What do you think of the experiment
  - What would you do in the same way? Why?
  - What would you do differently? Why?