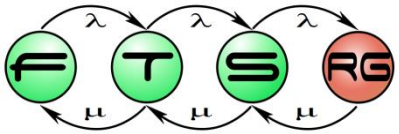
Parametrisierung der Modelle: Regression, Benchmarking

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Ansätze für Leistungsanalyse

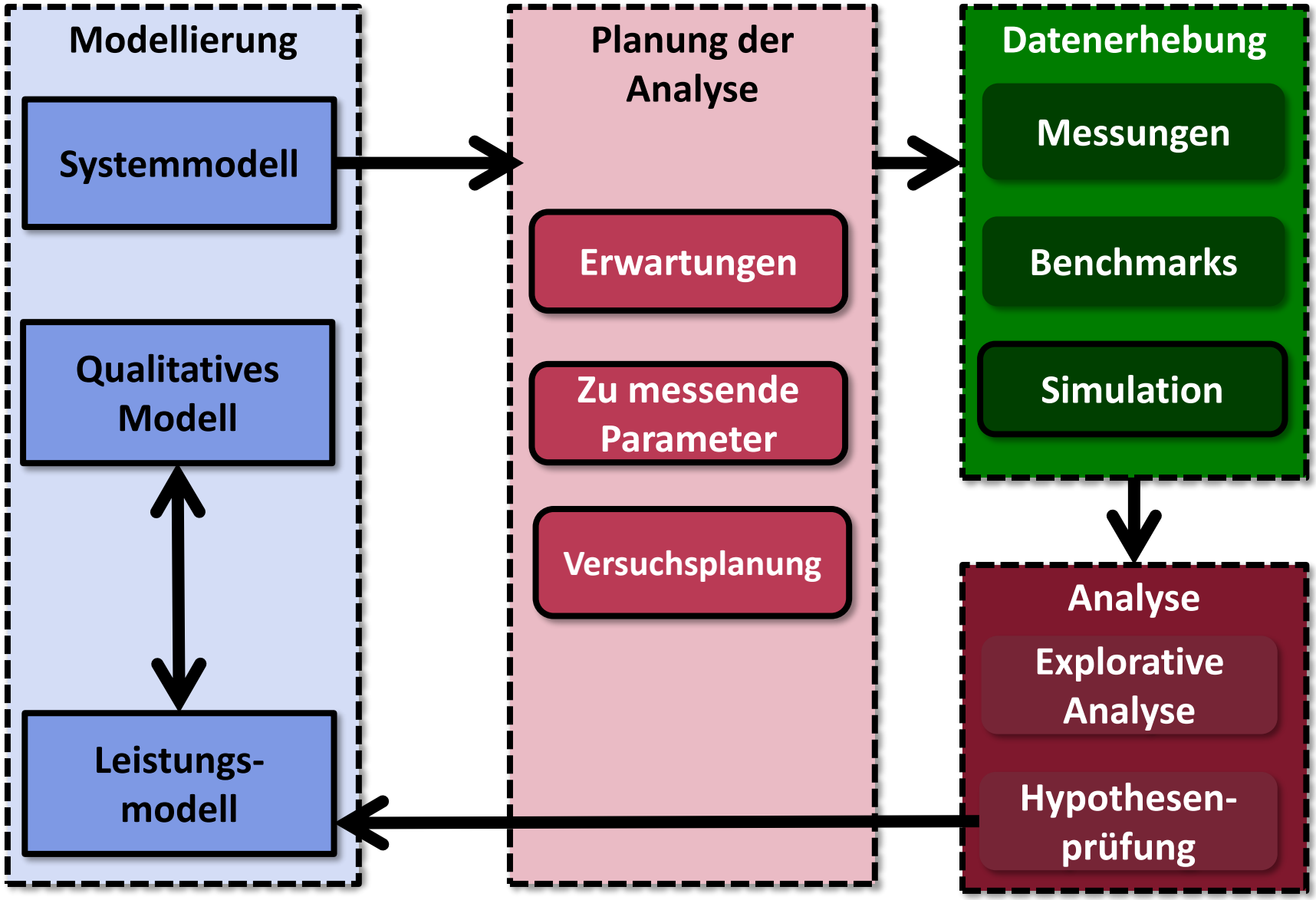
Belastungsprobe

- „Synthetische," einfache Last
- Erkundung des Grenzdurchsatzes
- Vergleich verschiedener Versionen eines Systems
- Untersuchung der Überlastung

Benchmarking

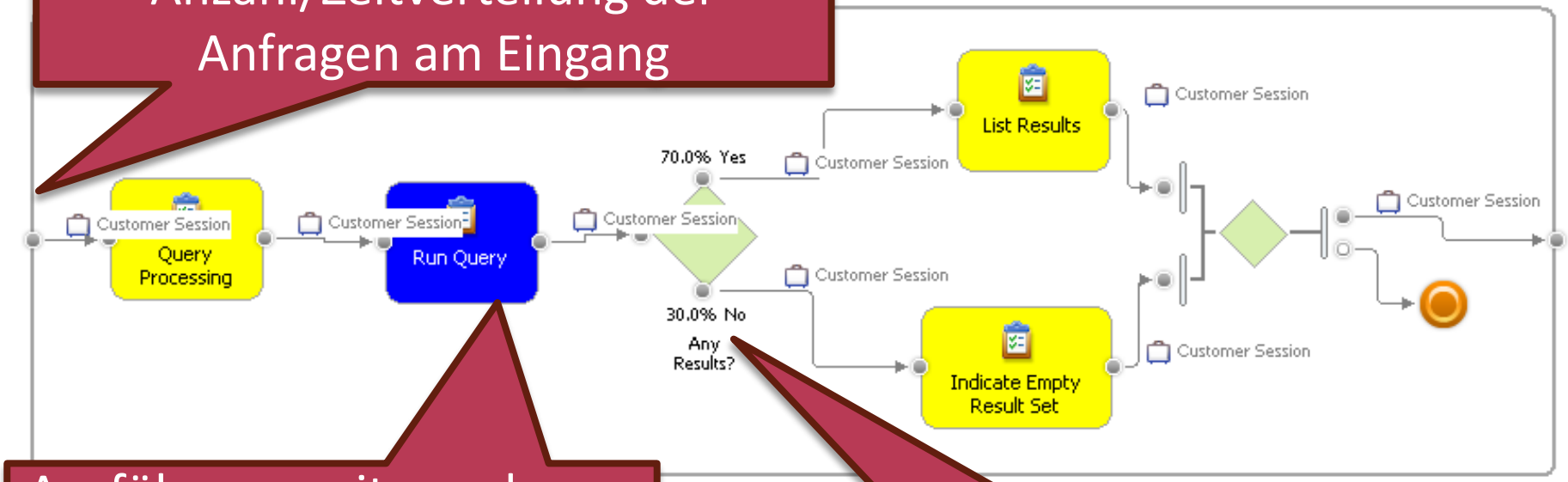
- Anhand reeller Anwendung
- Mehr komplexe Umgebung und Last
- Objektiver Vergleich verschiedener Systeme
- Im stabilen Operationsbereich

Vom Systemmodell zum Leistungsmodell



Grundfrage

- Können wir die Leistungsparameter richtig abschätzen?



Anzahl/Zeitverteilung der Anfragen am Eingang

Ausführungszeit gegebener Aufgabe an gegebener Ressource

Erwartungswert der Entscheidungswahrscheinlichkeit/ -Häufigkeit

Glaubhaftigkeit der Daten

- Empfindlichkeitsanalyse
  - Wie empfindlich sind die **Ausgangsparameter** auf die Änderungen der **Eingangsparameter**
  - (Anzahl/Kapazität der Ressourcen, Entscheidungen der Benutzer) → (Antwortzeit/Durchsatz des Prozesses)
  - „parameter sweep”: Analyse eines Parameters auf einem gegebenen Intervall

  → Wie genau sollen wir welche Parameter abschätzen?
- Faustregel: Glaubhaftigkeit der Daten
  - Die Ungewissheit der Messungen nimmt mit der Wurzel der Anzahl der Messungen ab

MATHEMATISCHE ABSCHÄTZUNG: REGRESSIONSMETHODEN

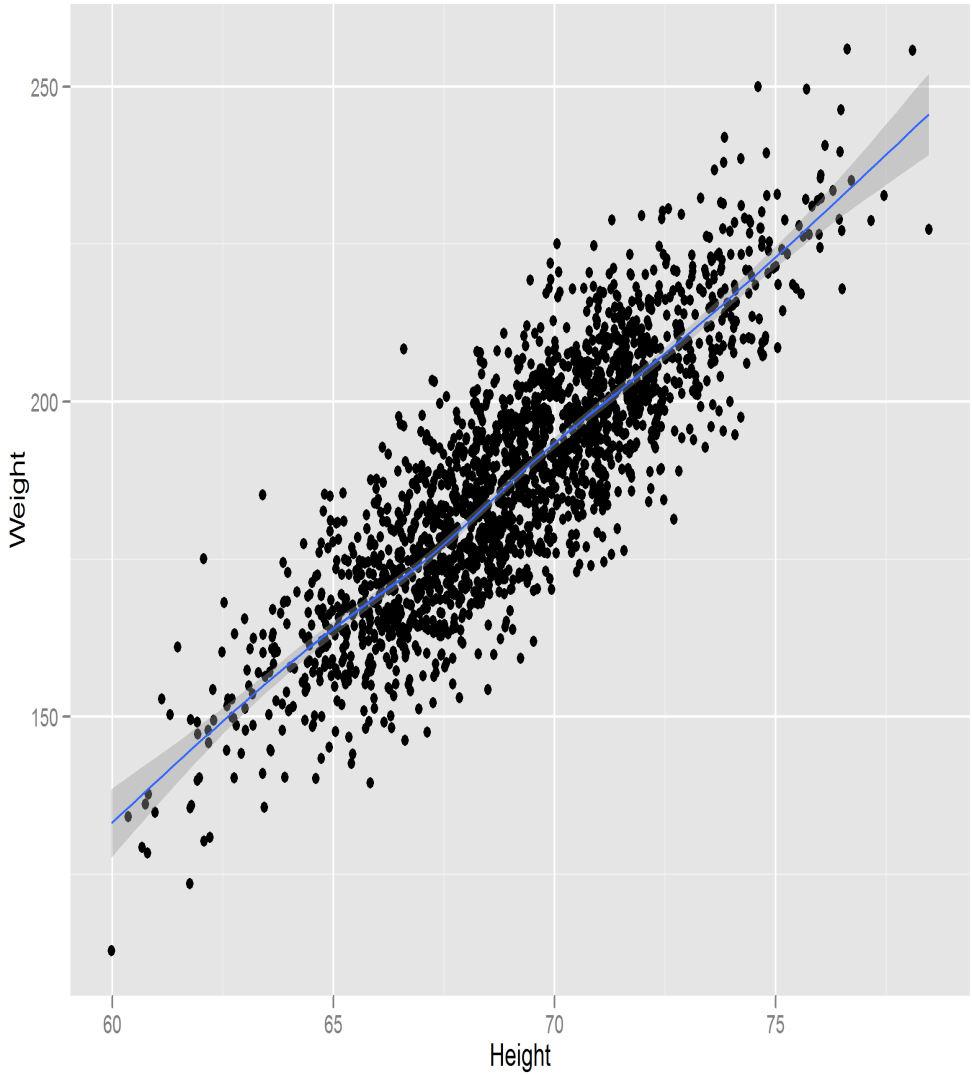
Aufgabe

- Ein System hat viele mögliche Parameter
- Wir möchten manche von denen abschätzen
  - schwierig/teuer zu messen
  - wurde nicht gemessen / kann nicht gemessen werden
- Wir möchten abschätzen/vorhersagen
  - Es ist noch nicht passiert, soll vorhergesagt werden
  - Es gab die Eingabe nicht (z.B. Benutzeranzahl noch zu niedrig)
  - Effekt kann erst später erfahren werden (z.B. die Antwortzeit erhöht sich nur während der Bedienung)
  - Wie gut können wir uns unsere Annahmen/Abschlüssen verlassen?

Regression

Funktion f

- <u>Eingabe:</u>
  Werte der Attributen
- <u>Ausgabe:</u>
  beste Annäherung der Beobachtungen
- „Faustregel”
- Beispiel:
  gemeinsame Verteilung der Gewicht und Höhe passt eigentlich auf eine Gerade



Regressionsmethoden

- Grundprinzip:

$$Y_t = f(\bullet) + \varepsilon_t$$

Wahrscheinlichkeits-variable

Annäherung

Fehler

Vorhergesagtes Ereignis

$$Y = f(X_1, X_2, \ldots, X_n)$$

Beobachtbare Variablen

- Durchschnittlicher Fehler (mean error)

$$ME = \frac{\sum_{t=1}^{n} (Y_t - F_t)}{n}$$

Abgeschätzter Wert

Gemessener Wert

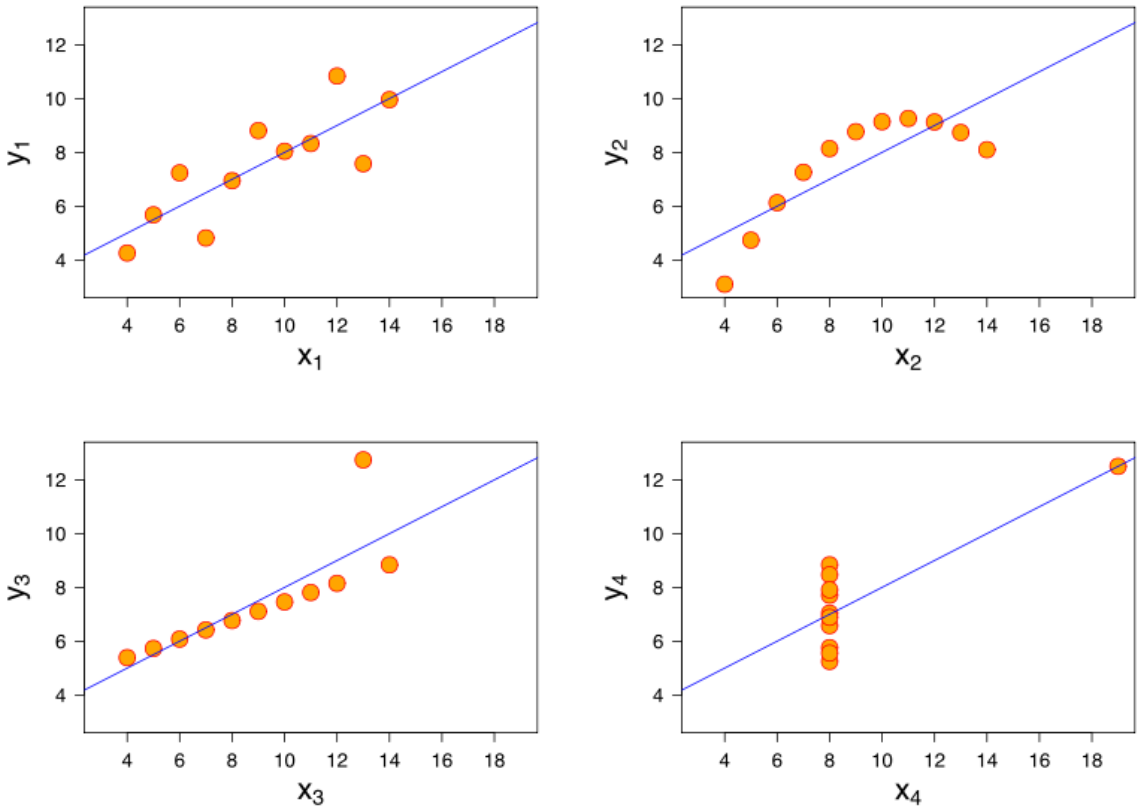
Lineare Regression

- Anpassung einer einfachen linearen Funktion an den Daten
  - Erwartet keine grosse Änderung in dem Verhalten des Systems $Y = a + bX$
- Methode der kleinsten Quadrate
  - Wir suchen die Parameter *a,b* (hier: a – Verschiebung, b – Steilheit), wo $SSE = \sum_{t=1}^{n} \varepsilon_t^{\,2} = \sum_{t=1}^{n} (Y_t - F_t)^2$ minimal ist (Sum of Squared Errors)
- Ziel: $\sum_{t=1}^{n} (Y_t - F_t)^2 = \sum_{t=1}^{n} \left[Y_t - (a + bX_t) \right]^2$ zu minimieren

Lineare Regression

- Die am besten passende Gerade
- **ABER:**

  Anscombe's quartet
  - Grundsätzlich vershiedene Daten
  - Gleiche Regressions-Gerade



Lineare Regression (Fortsetzung)

- (Quadrat des) Korrelationskoeffizient(es)
  - Stärke des Zusammenhanges zwischen den abgeschätzten und tatsächlichen Werten einer Variable
  - Ein Wert zwischen 0 und 1
  - 0: kein Zusammenhang
  - -1 oder 1: Funktionsähnlicher Zusammenhang
    - -1 .. 1 (Richtung des Zusammenhanges: inverse .. direkte)

$$R^2 = \frac{\sum_{t=1}^{n}\left(F_t - \overline{Y}\right)^2}{\sum_{t=1}^{n}\left(Y_t - \overline{Y}\right)^2}$$

- Beispiel:
  E-mail-Dienstleistung, 8 Wochen lang Spitzenlast gemessen

Woche	1	2	3	4	5	6	7	8
Max. Last (email/Min)	420	410	437	467	448	460	507	514

Wie kann die Last abgeschätzt werden?

Wie hoch ist der Korrelationskoeffizient?

Lineare Regression (Beispiel)

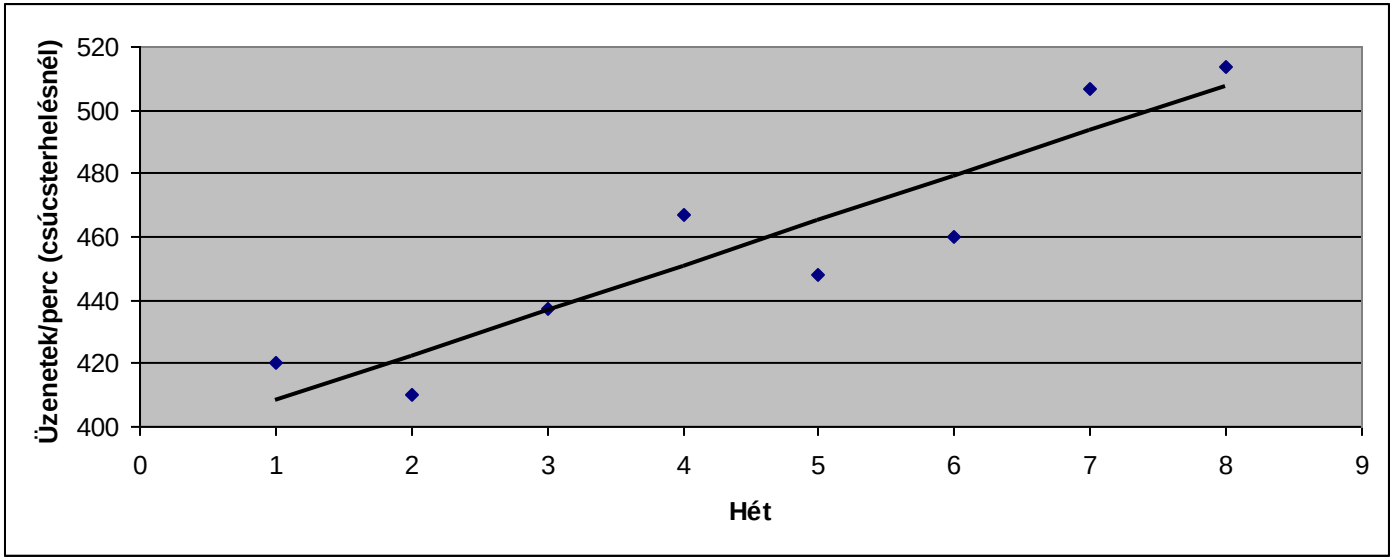
Mit der Methode der kleinsten Quadrate

$Y=393.98+14.20X$

Korrelationskoeffizient:

$R^2=0.855$

Messwert	Vorhergesagt
420	408,18
410	422,38
437	436,58
467	450,78
448	464,98
460	479,18
507	493,38
514	507,58
	521,78



Zusammenhang zwischen Zweier Variablen

- Nehmen wir an, dass es ein linearer Zusammenhang zw. der Anzahl der Benutzer und der Anzahl der verschickten Mails besteht. (z.B. anhand der Logs)

Bejelentkezett felh. átlagos száma (1 óra alatt)	2450	2765	2241	2860	3011	2907	3209
Átl. terhelés (kimenő+bejövő emailek/óra)	19257	20488	18152	21450	21077	20639	22142

- Lineare Regression, Methode der kleinsten Quadrate:

AnzahlDerMails = f(AnzahlDerBenutzer)

Y=9480.48 + 3.95X, R^2=0.937 – *starker Zusammenhang*

Nichtlineare Methoden

- Exponentielle Annäherung: $Y_t = a \times b^t$
  - passt gut an den Zuwachs des Webverkehrs
- Umordnung der Gleichung: $\log Y_t = \log a + t \log b$

$$\log Yt = Y', \log a = a', \log b = b'$$

$$Y' = a' + b't$$

- Methode der kleinsten Quadrate passt schon
- Z.B. Werte der grössten gemessenen Last seien gegeben

  Wie hohe Spitzenlast wird zum Jahresende erwartet?

Hónap	1	2	3	4	5	6	7	8	9	10
Max. kérések/sec (Y_t)	1035	1100	1160	1250	1350	1555	1770	1950	2210	2630
ln (Y_t)	6.942	7.003	7.056	7.13	7.207	7.349	7.478	7.575	7.7	7.874

Beispiel: Exponentielle Last

- Schätzfunktion: $Y_t = a \times e^{bt}$
- Methode der kleinsten Quadrate (auf der linearen Funktion)

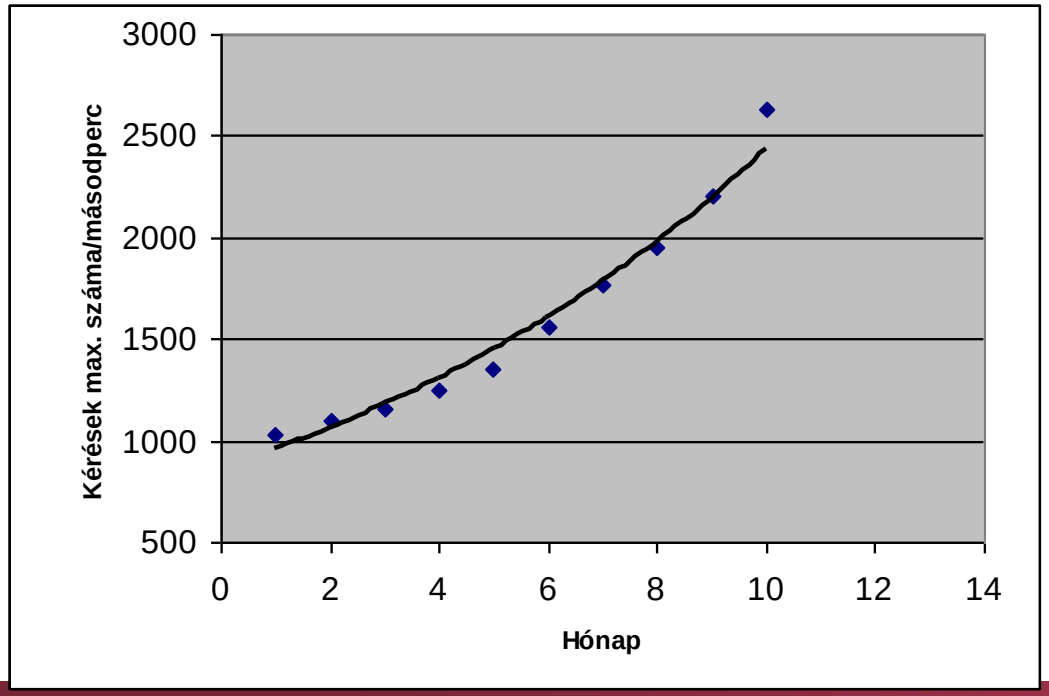
$$Y' = a' + b't, a' = 6.717, b' = 0.110, a = e^{a'}$$

- Ergebnis:

$$Y_t = 826.33 \times e^{0.11t}$$

- 12. Monat:

$Y_t = 3093.3$



Methode der gleitenden Mittelwerte

- Hilfreich bei kurzfristiger Vorhersage
- Liefert die Werte nur einzeln
- Erwarteter Wert: der Durchschnitt der letzten *n* Werte

$$F_{t+1} = \frac{\sum_{i=t}^{t-n+1} Y_i}{n}$$

wo Y_t ist der im Zeitpunkt *t* gemessene Wert

F_{t+1} der erwartete Wert

n ist typischerweise zwischen 3 und 10

(sonst wäre der Fehler zu groß)

Exponentielles Gleitfenster

- Liefert die Werte nur einzeln
- Als Durschnitt der bereits gemessenen Werte
  - spätere Messungen (und Messfehler) mit grösseren Gewichten
- Hilfreich bei kurzfristiger Vorhersage
  - (warum exponentiell?)

$$F_{t+1} = F_t + \alpha\left(Y_t - F_t\right)$$

wo F_t : ist der für Zeitpunkt t vorhergesagte Wert

Y_t : der im t. Zeitpunkt gemessene Wert

$Y_t - F_t$: Messfehler im t. Period

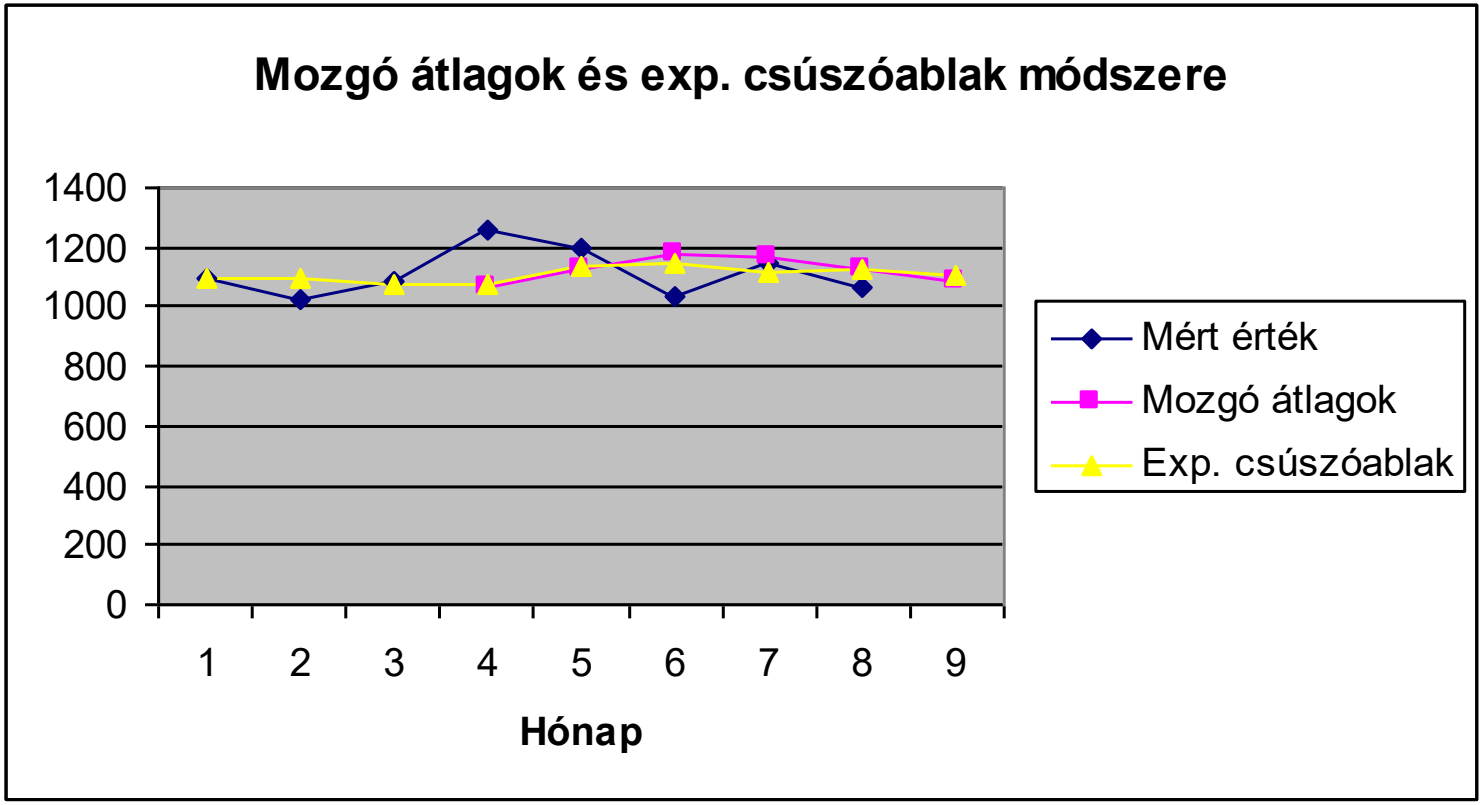
α : Gewicht ($0 \leq \alpha \leq 1$, inder Praxis $0.05 \leq \alpha \leq 0.3$)

Vergleich der zwei Methoden

- Gegebene Bandbreiteanforderungen, Vorhersage mittels der zwei Methoden

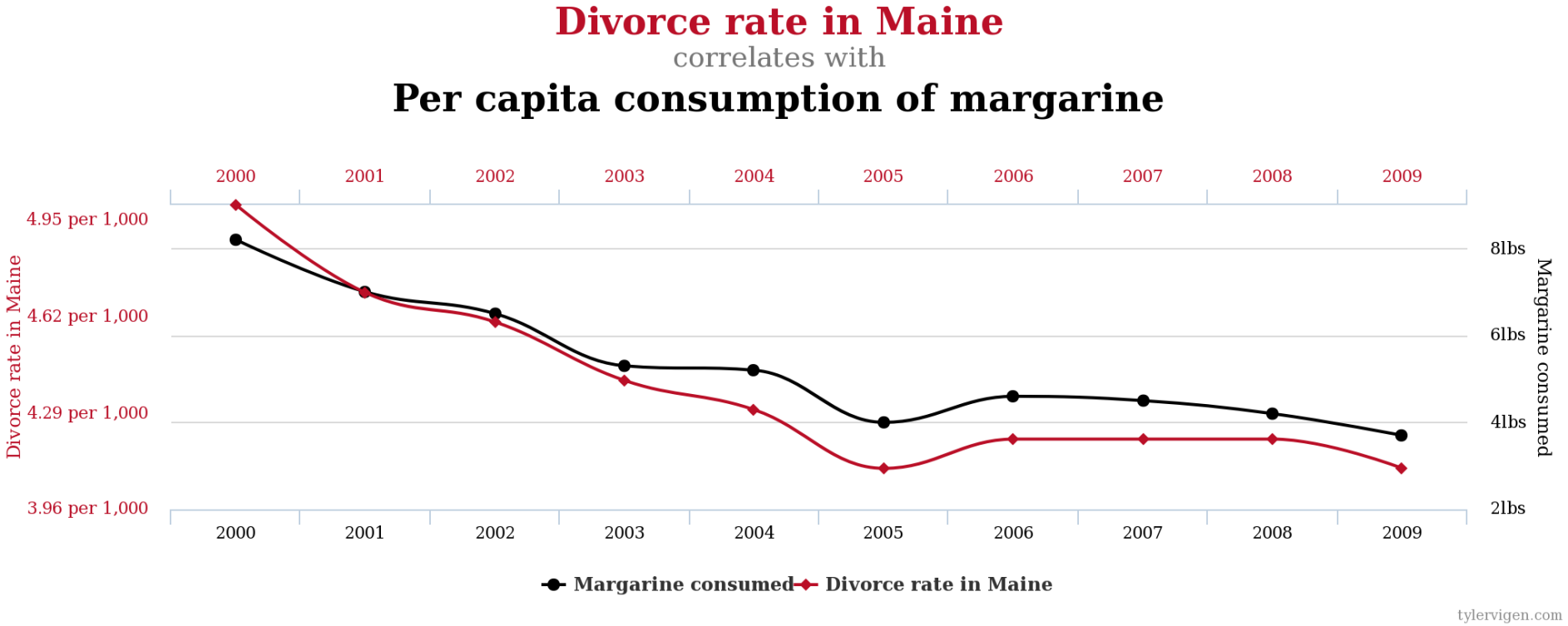
Hónap	Sávszélesség igény	Mozgó átlagok módszere (n=3)	Exp. csúszóablak (α = 0.3)
1	1100		1100.00
2	1020		1100.00
3	1090		1076.00
4	1255	1070.0000	1080.20
5	1195	1121.6667	1132.64
6	1039	1180.0000	1151.35
7	1145	1163.0000	1117.64
8	1066	1126.3333	1125.85
9		1083.3333	1107.90

Vergleich der zwei Methoden



Wichtig zu Merken!

Kausalität (gemeinsame Ursache) **!= Korrelation** (gemeinsames Vorkommen)



Beispiel aus der Informatik: viele Benutzer →

hohe Auslastung <u>UND</u> hohe Antwortzeit

WARUM BENCHMARKING?

Warum benchmark?

Benchmarking - Definition

- Wikipedia

> *„In **computing**, a benchmark is the **act of running** a computer program, a set of programs, or other operations, in order to **assess the relative performance** of an object, normally by running a number of **standard tests** and trials against it.”*

Benchmarking ist

- das **Ausführen** eines **Programmes** (von mehreren Programmen oder von anderen Operationen)
- **mit standardisierten Tests** oder Eingaben,
- um **die relative Performanz** eines Objektes zu **bestimmen.**

Benchmarking

- Ziele: Vergleich der Leistung von Software- oder Hardwarekomponenten
  - Entscheidungsunterstützung
    - Welche Komponente soll gekauft/installiert werden?
    - Was kann das existierende System leisten? (Schwächen/Stärken)
    - Was können die Konkurrenten?
  - Leistungstesten
    - Soll die Leistung erhöht werden? Wo? (bei Entwicklung)
    - Ist eine gegebene Einstellung optimal?
    - Hat eine Einstellung tatsächliche Auswirkung auf die Gesamtleistung?

Erwartungen

- Wiederholbarkeit
  - Repeatability
- Reproduzierbarkeit
  - Reproducilibity
- Relevanz
- Konformität mit Normen/Vereinbarungen
- Verallgemeinerter Anwendungsfall
  - Das Ergebnis sei für den Durchschnittsbenutzer interpretierbar

STANDARD BENCHMARKS

SPEC, TPC-C, …

SPEC Benchmarks

- http://www.spec.org/benchmarks.html
  - Standard Performance Evaluation Corp.
- Ressource- und Anwendungsbenchmark
  - CPU
  - Anwendungen
  - Mailserver
  - Webserver usw.
- Benchmark: bestellbare Dienstleistung

SPEC CPU2006

- CPU-intensiv
- CINT2006
  - Rechenintensiv, mit Ganzzahlen
- CFP2006
  - Mit Gleitkommazahlen

- Ergebnisse: http://spec.org/cpu2006/results/
  - Test Sponsor (vendor), System Name (product)
  - Processor: enabled cores, enabled chips, cores/chip, threads/core
  - Results: base, peak

CINT2006 und CFP2006 Leistungsgeneratoren

- CINT2006 :

400.perlbench	C	Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence
462.libquantum	C	Physics / Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

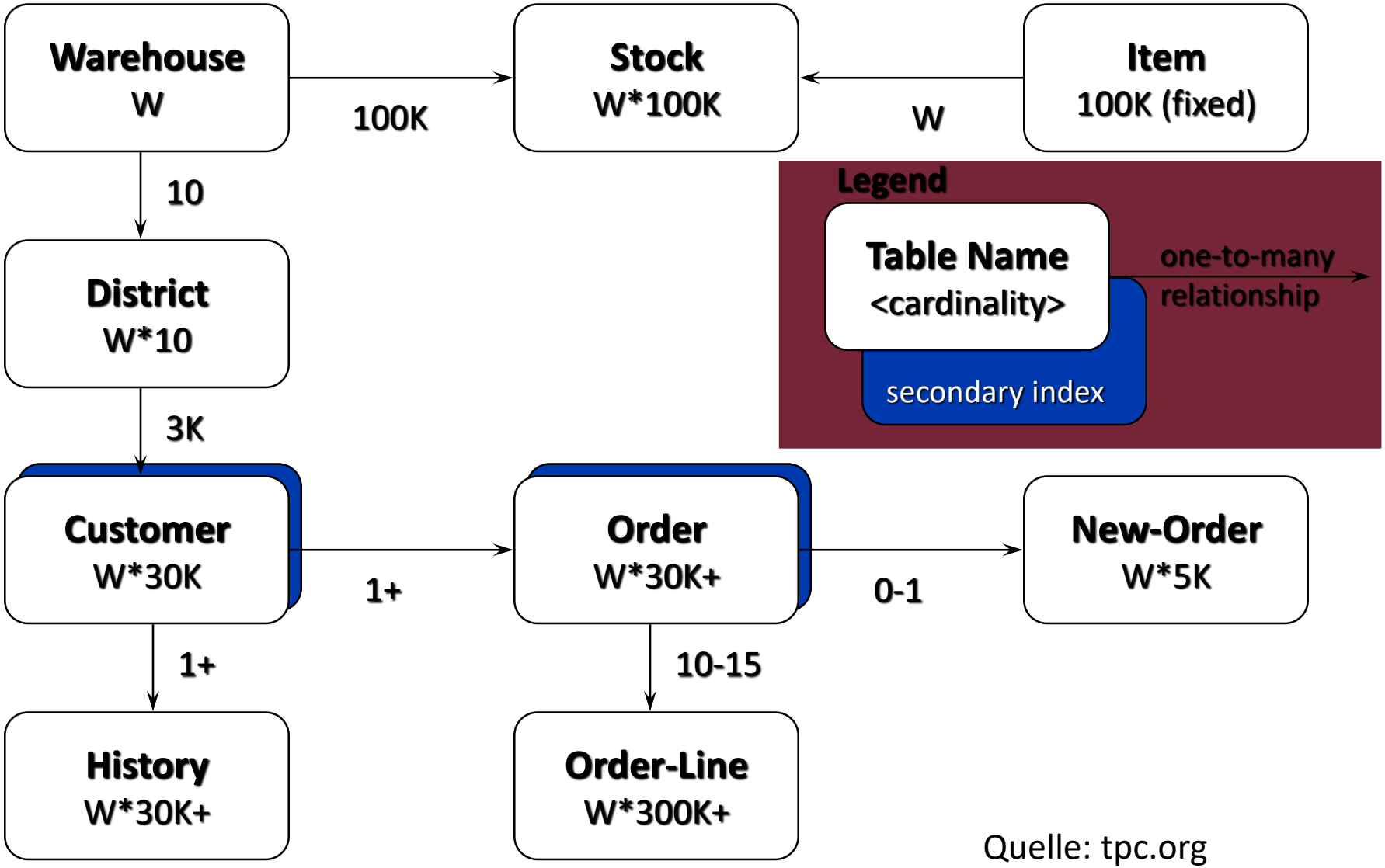
- CFP2006:

410.bwaves	Fortran	Fluid Dynamics
416.gamess	Fortran	Quantum Chemistry
433.milc	C	Quantum Chromodynamics
434.zeusmp	Fortran	Fluid Dynamics
435.gromacs	C, Fortran	Molecular Dynamics
436.cactusADM	C, Fortran	General Relativity
437.leslie3d	Fortran	Fluid Dynamics
444.namd	C++	Molecular Dynamics
447.deall	C++	Finite Element Anal.
450.soplex	C++	Linear Programming
453.povray	C++	Image Ray-tracing
454.calculix	C, Fortran	Structural Mechanics
459.GemsFDTD	Fortran	Electromagnetics
465.tonto	Fortran	Quantum Chemistry
470.lbm	C	Fluid Dynamics
481.wrf	C, Fortran	Weather
482.sphinx3	C	Speech Recognition

Das TPC-Benchmark

- Beobachtung von Datenbanksystemen
  - RDBMS+OS+HW
- Messungsumgebung
  - Musterdatenbank: Kunden und Bestellungen
  - 5 Arten von Transakt. (Abfrage/Änderung) gemischt
  - Obergrenze für die Laufzeit
  - Realitätsnahe Bedingungen: ACID Transaktionen, Bedenkzeit der Benutzer
    (Atomarität, Konsistenz, Isolation und Dauerhaftigkeit)
- Gemessene Daten
  - Durchsatz (tpmC) (*transaction per minute*)
  - „Effizienz” ($/tpmC)

TPC-C Übersicht



Quelle: tpc.org

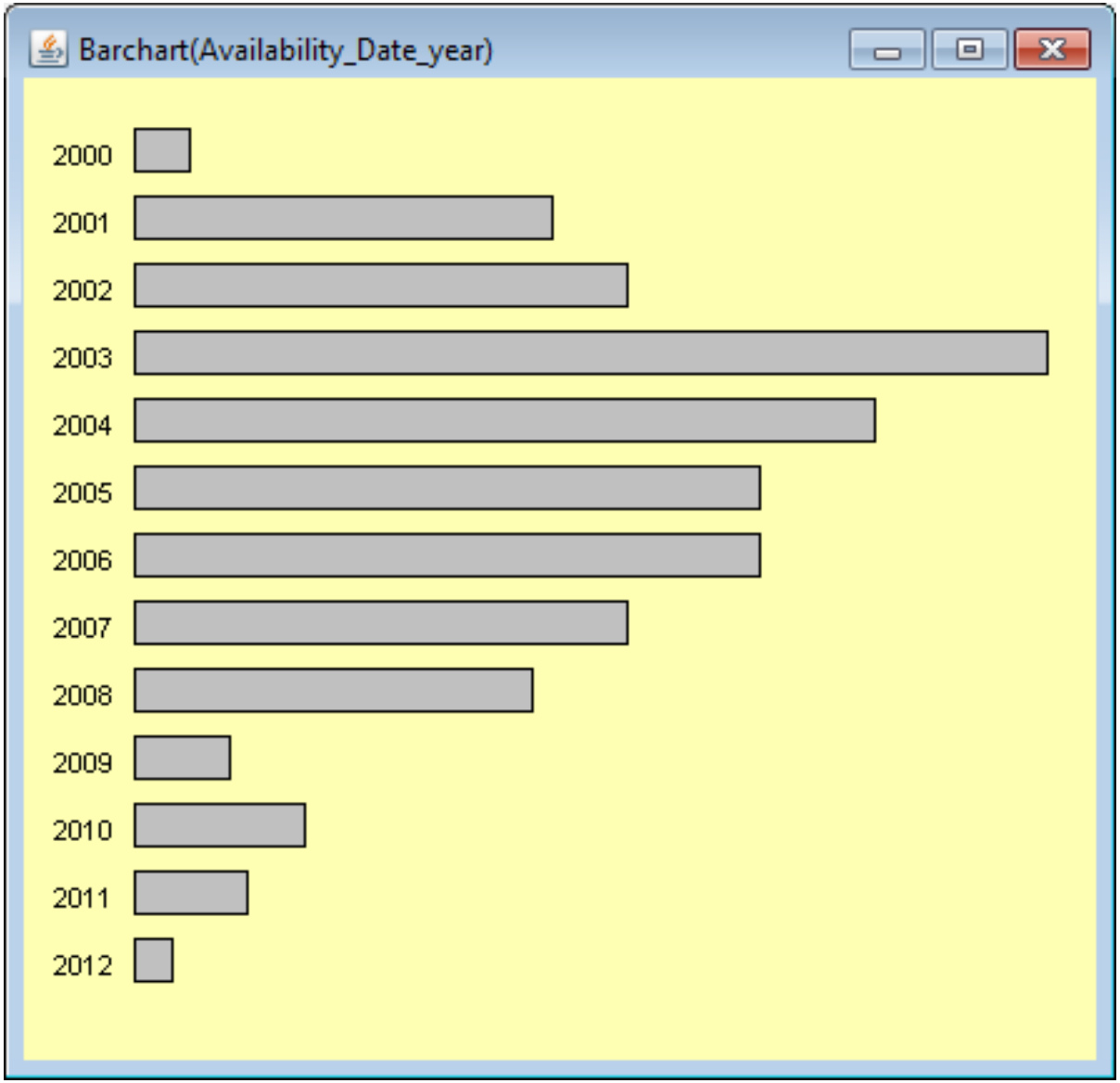
Vor der Analyse: Reinigung der Daten

- Ausgangsdatensatz:

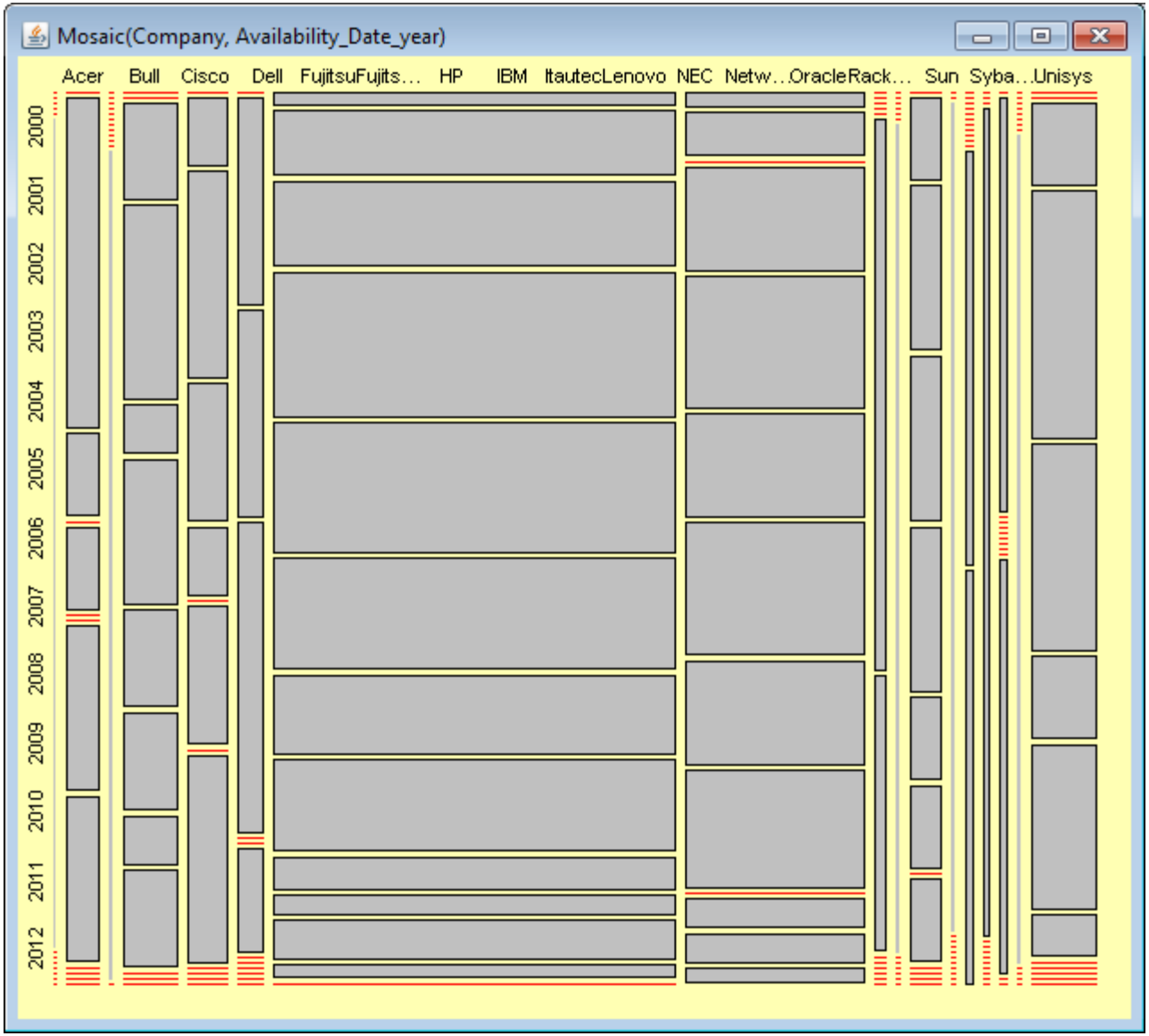
	A	B	C	D	E	F	G	H	I	J	K
1	TPC-C BENCHMARK RESULTS										
2	These results are valid as of date 6/12/2012 10:04:24 PM										
3											
4			TPC-C Results - Revision 5.X								
5											
6	Company	System	Spec. Revision	tpmC	Price/Perf	Total Sys. Cost	Currency	Database Software	Operating System	TP Monitor	Server CPU Type
7	Acer	Altos R710	5.5	66543	12.42	826507.55	AUD	Microsoft SQL Server	Microsoft Windows Serv	Microsoft CO	Intel Xeon - 3.6 GHz
8	Bull	Bull Escal	5.9	6085166	2.81	17127928	USD	IBM DB2 9.5	IBM AIX 5L V5.3	Microsoft CO	IBM POWER6 - 5.0
9	Bull	Bull Escal	5.9	629159	2.49	1566664	USD	IBM DB2 9.5 Enterpri	IBM AIX 5L V5.3	Microsoft CO	IBM POWER6 - 4.2
10	Bull	Bull Escal	5.8	1616162	3.54	5716286	USD	IBM DB2 9.1	IBM AIX 5L V5.3	Microsoft CO	IBM POWER6 - 4.7
11	Bull	Bull Escal	5.8	404462	3.51	1417121	USD	Oracle Database 10g	IBM AIX 5L V5.3	Microsoft CO	IBM POWER6 - 4.7

- Überflüssige Daten:
  - Zeilen (z.B. die ersten und letzten Zeilen, die nicht zum Ergebnis gehören)
  - Spalten (z.B. „Server CPU Type” interessant?)
  - z.B. Kosten in verschiedenen Währungen
  - Dezimalkomma vs. Dezimalpunkt
  - *Fujitsu* vs. *Fujitsu-Siemens* (zu vereinigen?)

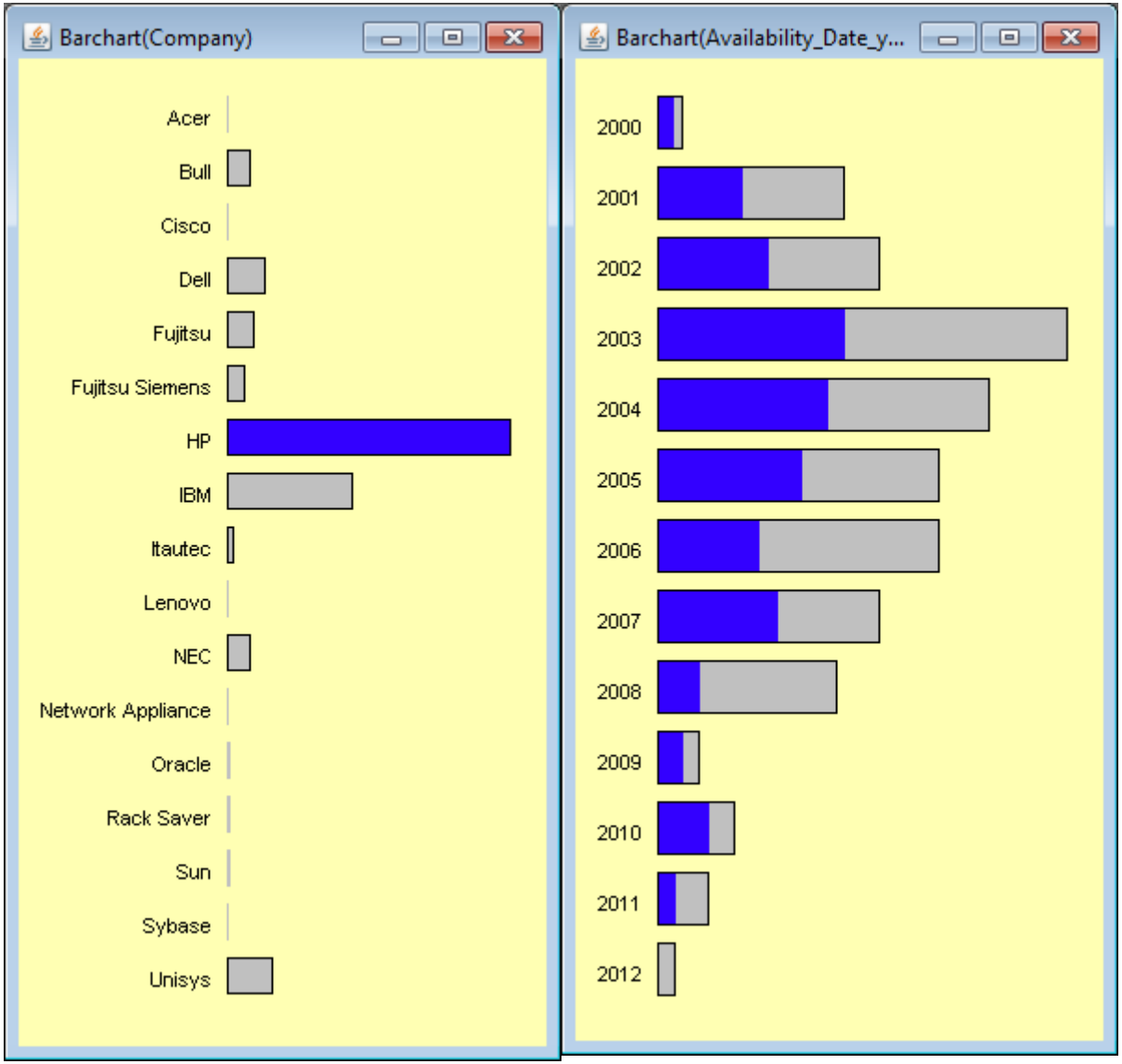
Welche Jahren werden im Benchmark betrachtet?



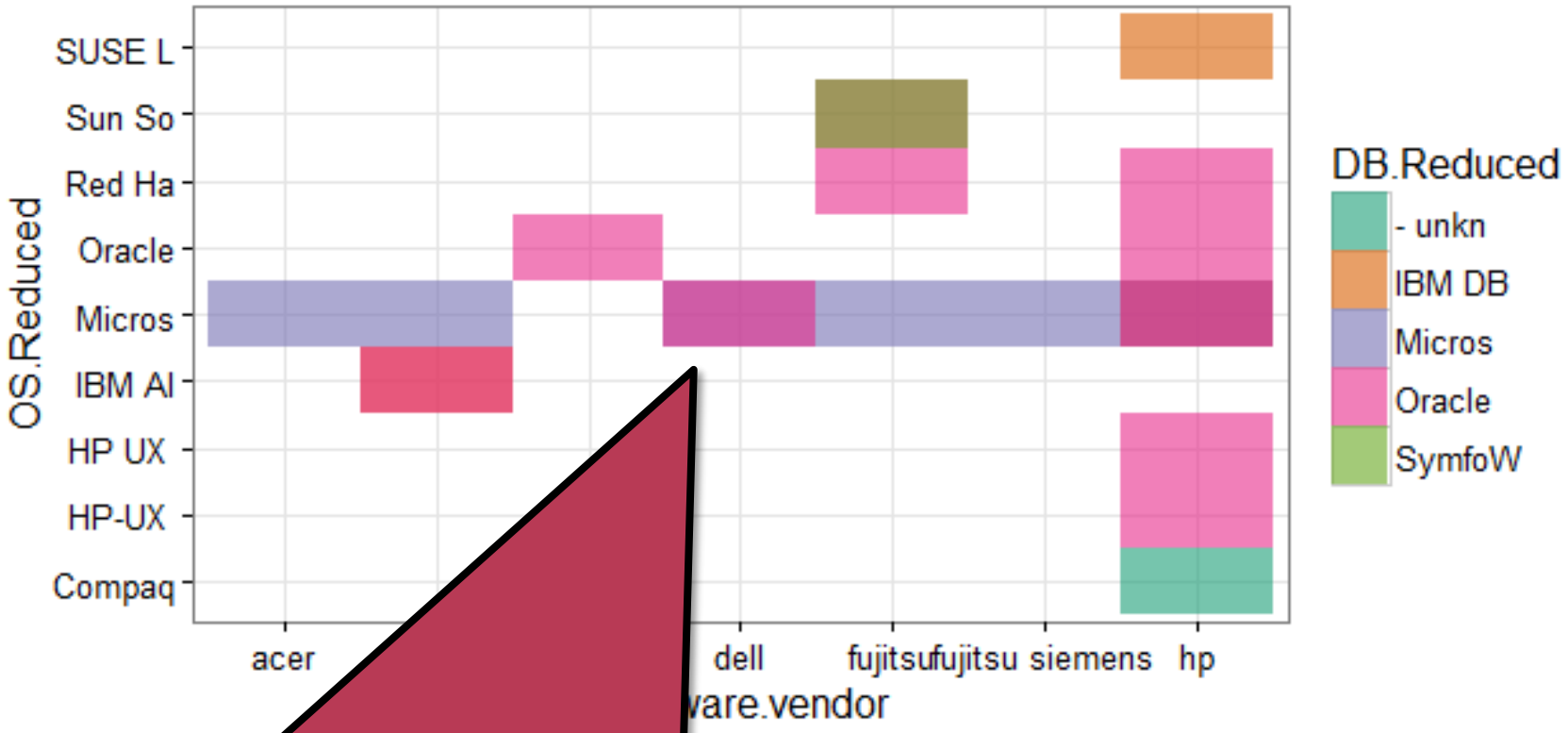
In welchen Jahren waren die Lieferanten aktiv?



In welchen Jahren waren die Lieferanten aktiv?

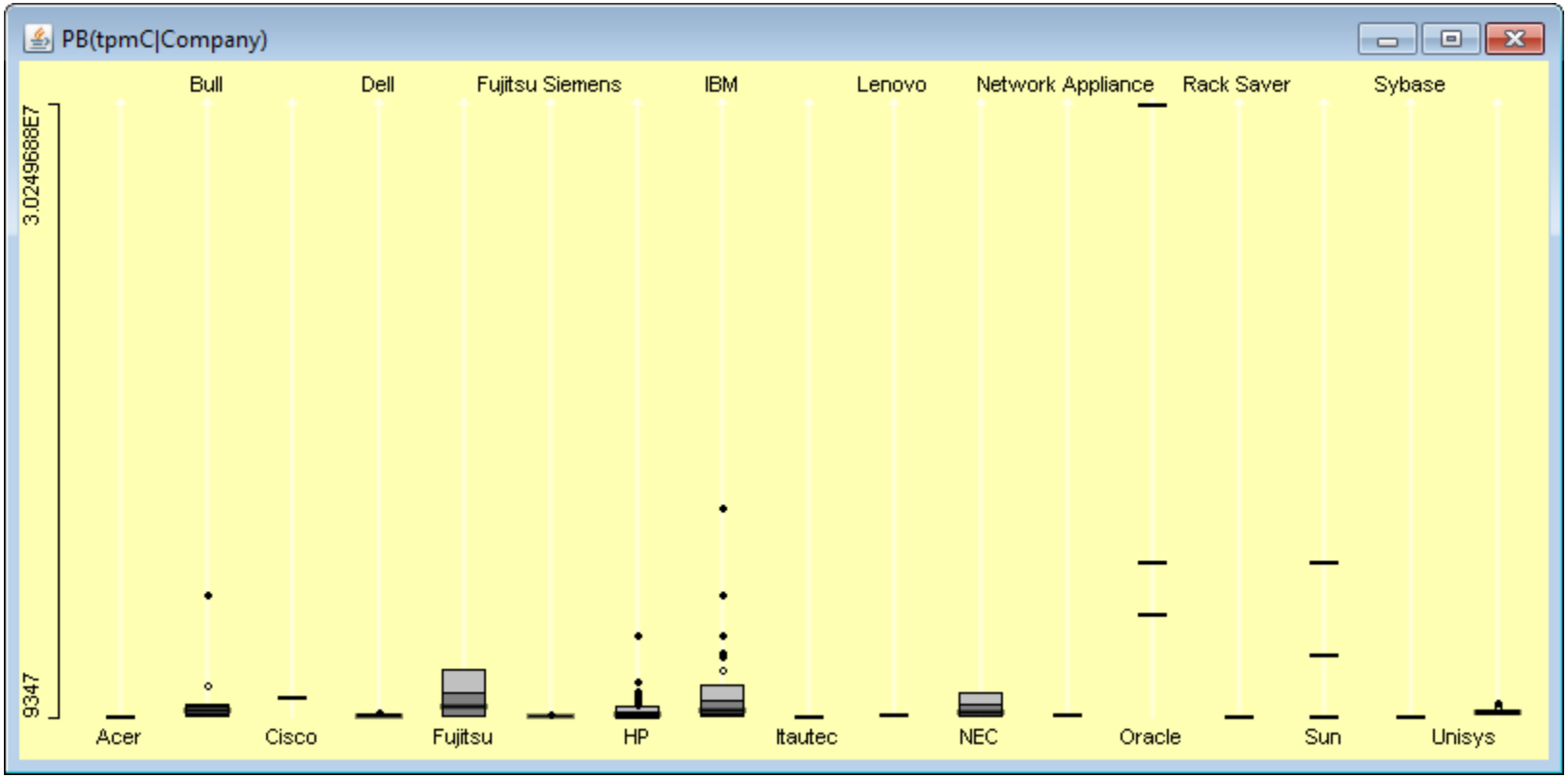


Gemessene Konfigurationsvariationen

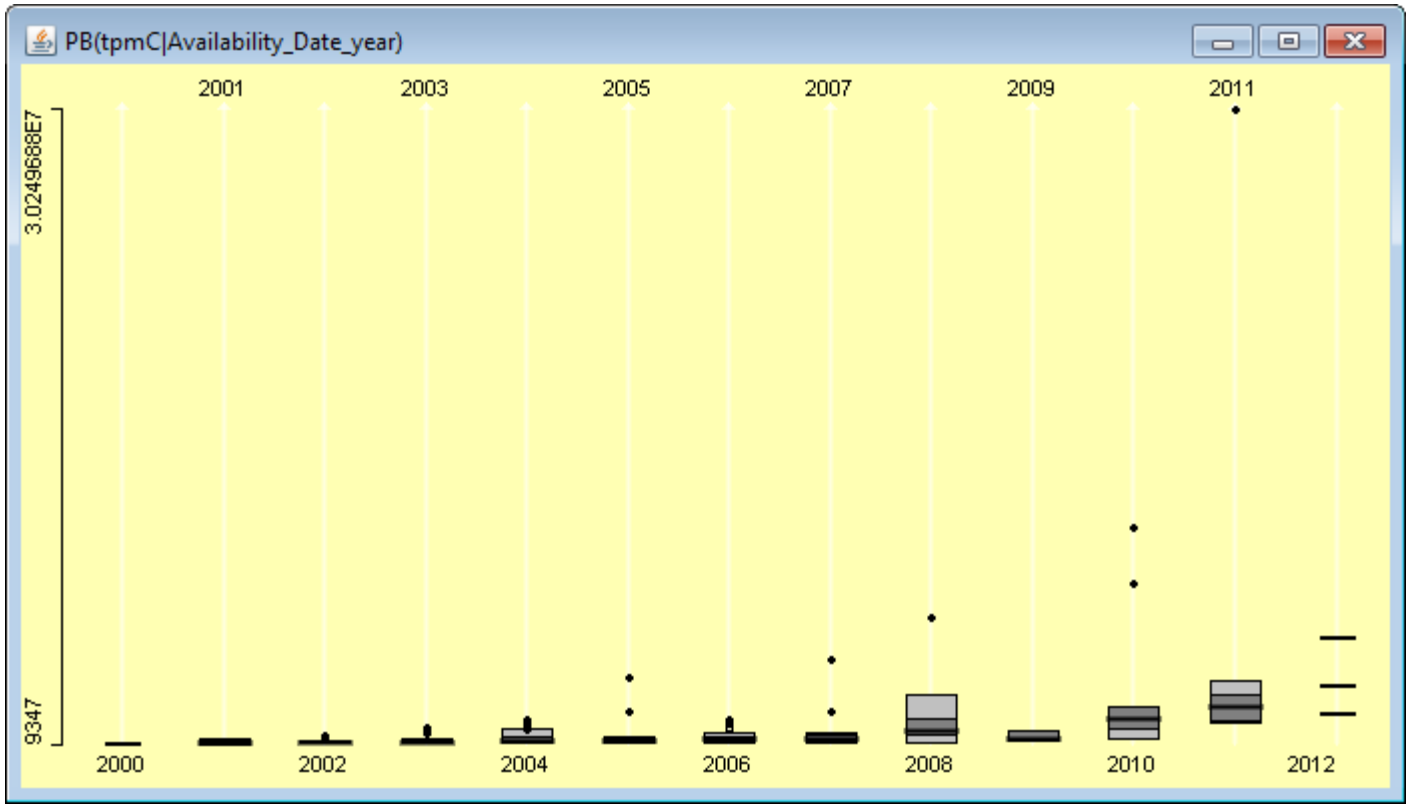


Datenbanksysteme können an mehreren Betriebssystemen laufen

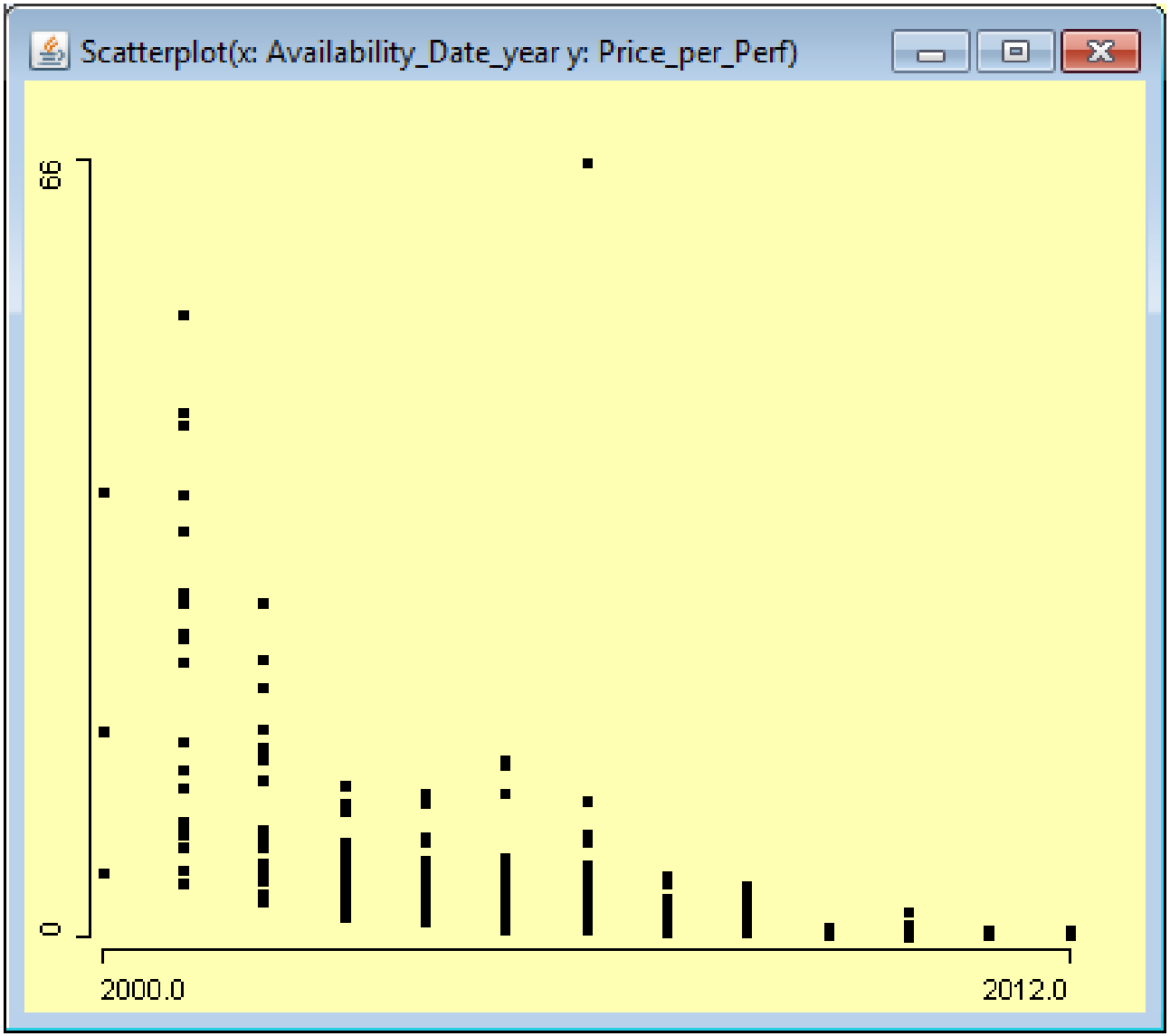
Gemessene Werte der Leistungsmetriken



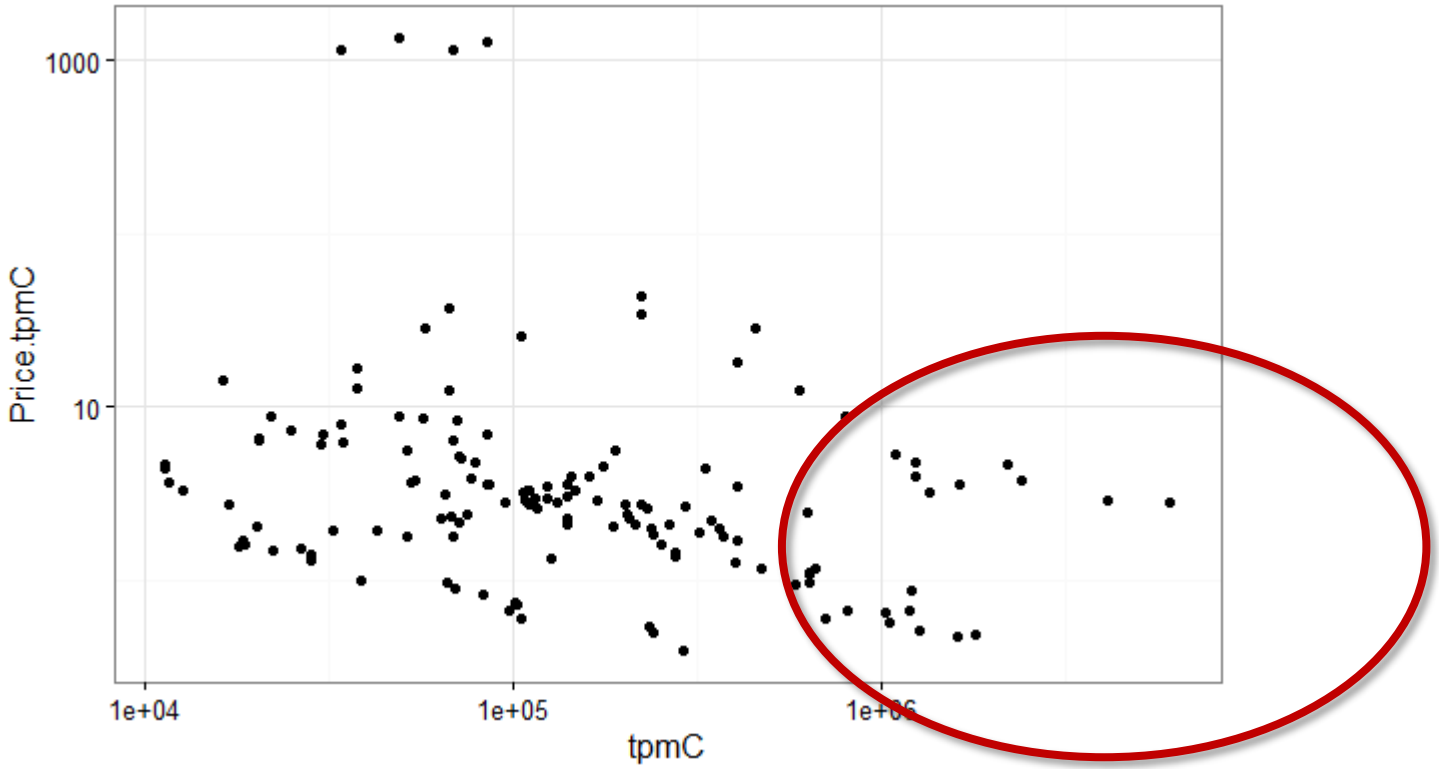
Wie hat sich Leistung mit der Zeit verändert?



Wie haben sich die Preise verändert?

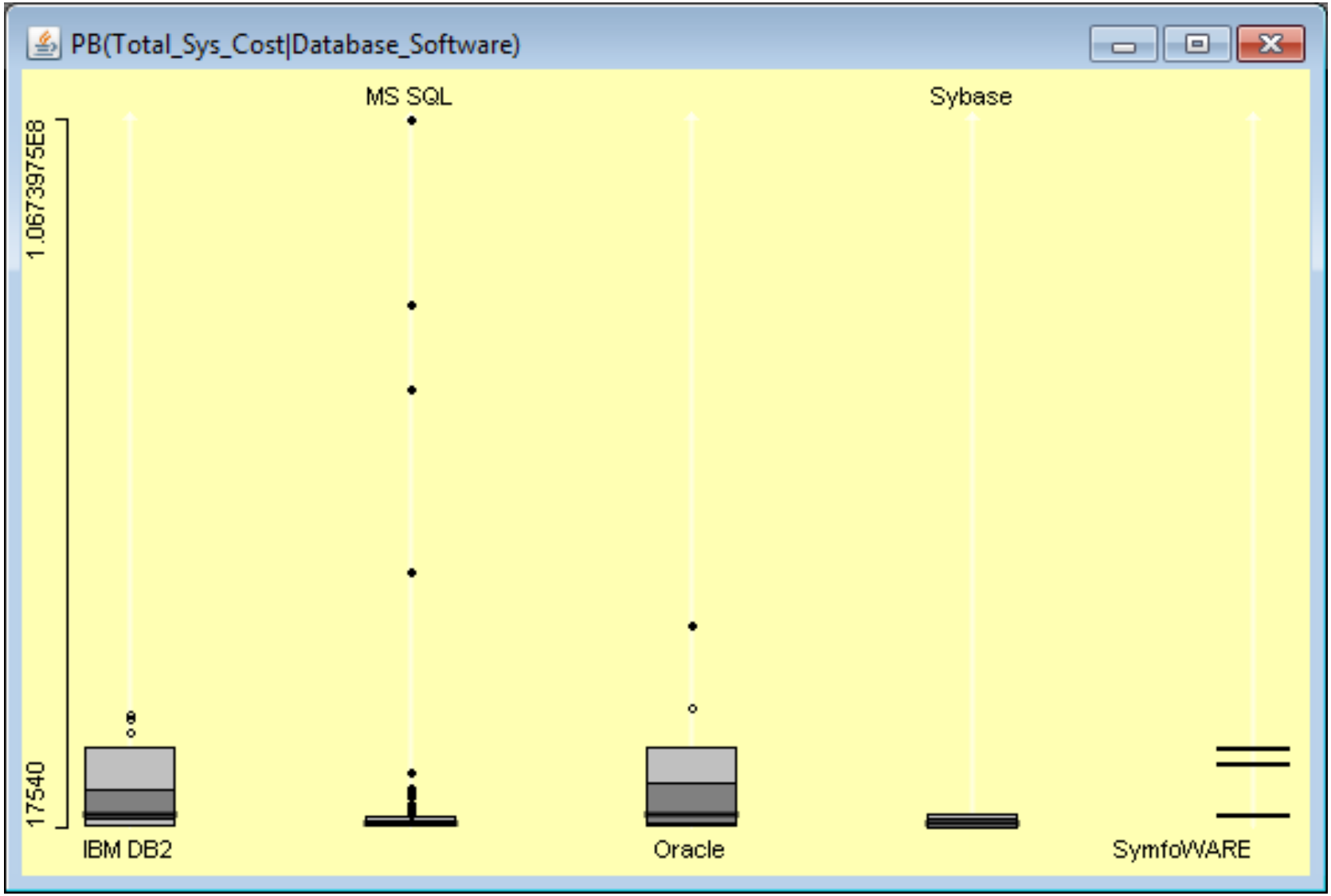


Benchmark Ergebnisse

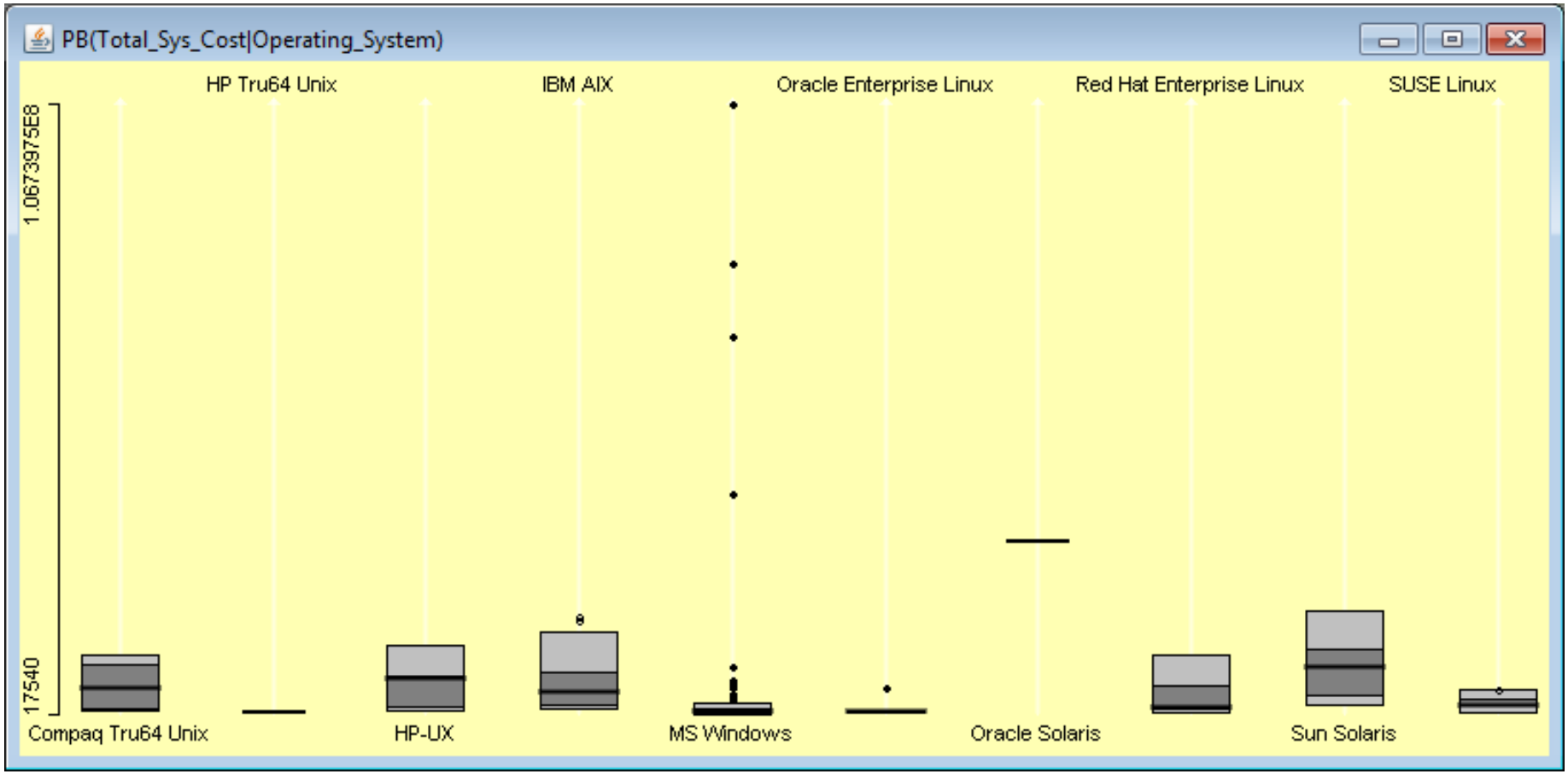


Logarithmische Skalierung?

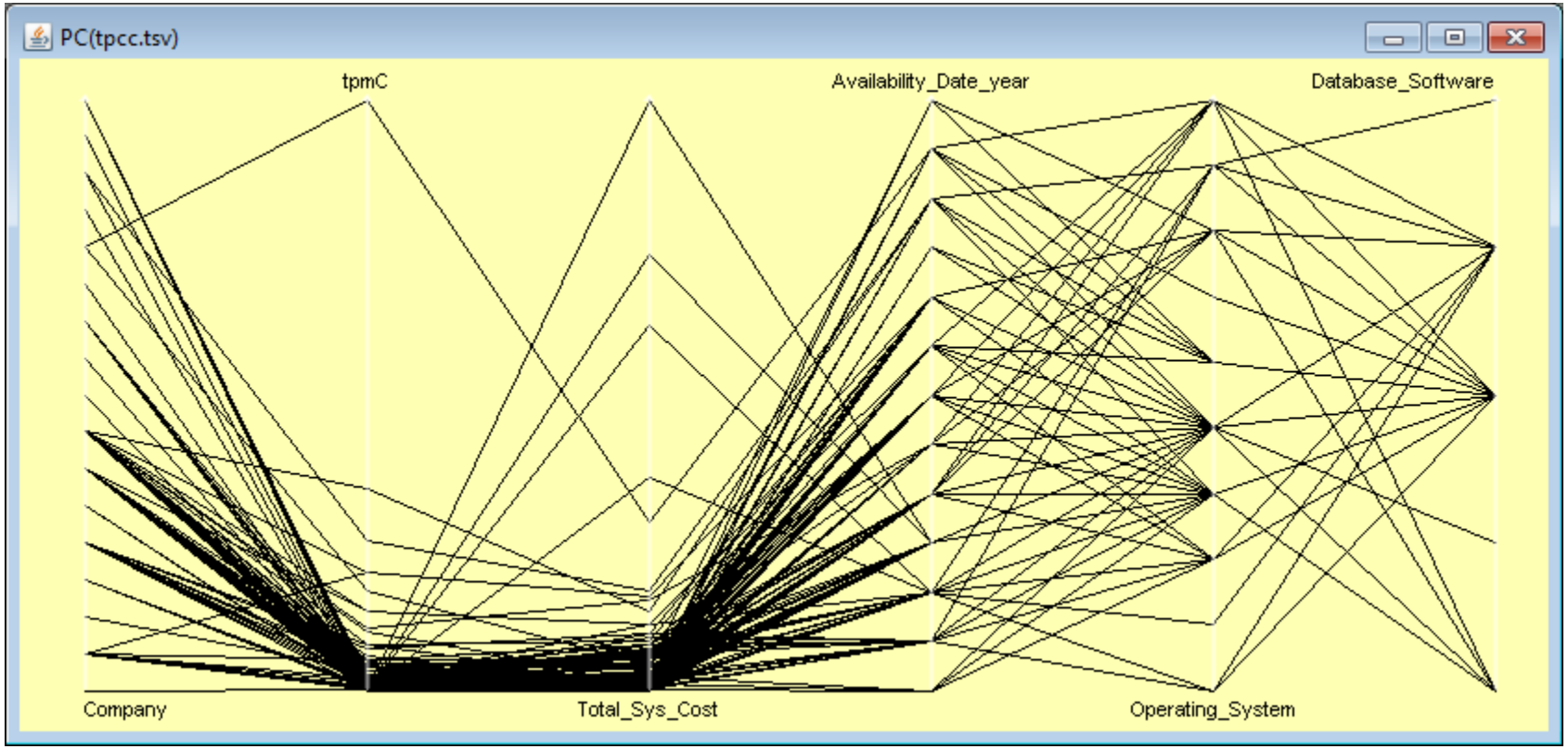
DB-Management SW wählen



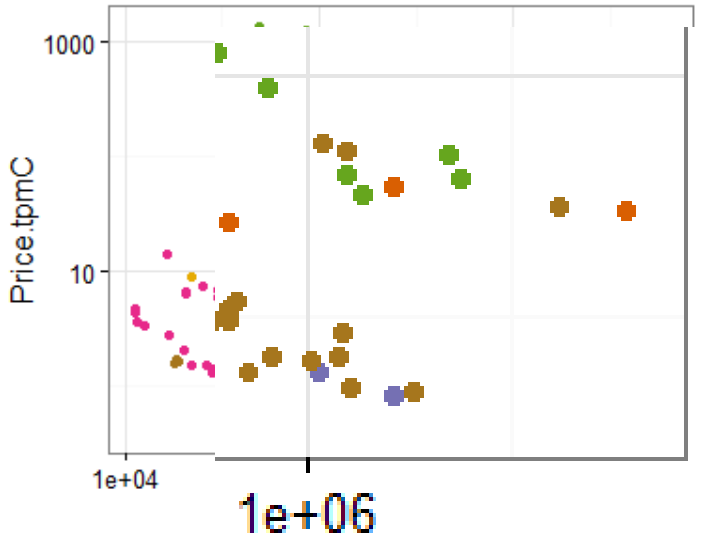
OS wählen

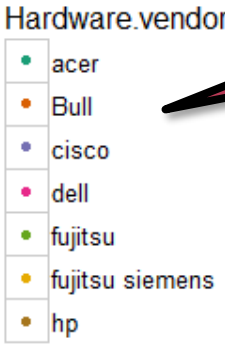

PB(Total_Sys_Cost|Operating_System)
HP Tru64 Unix
IBM AIX
Oracle Enterprise Linux
Red Hat Enterprise Linux
SUSE Linux
1.0673975E8
17540
Compaq Tru64 Unix
HP-UX
MS Windows
Oracle Solaris
Sun Solaris

Das ,,big picture''

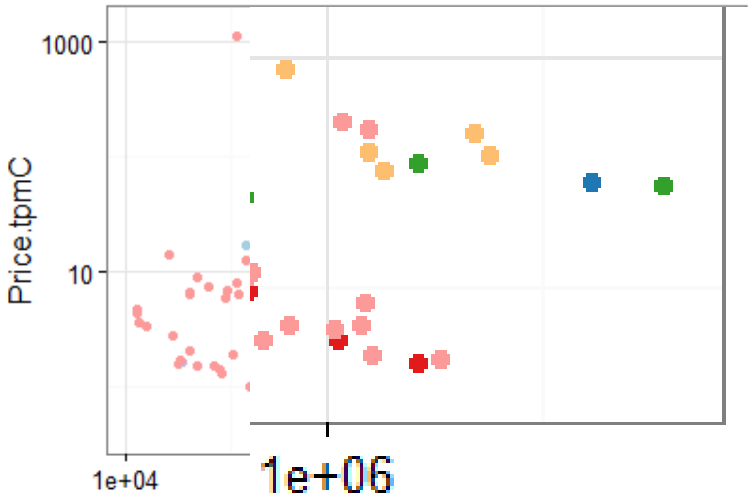


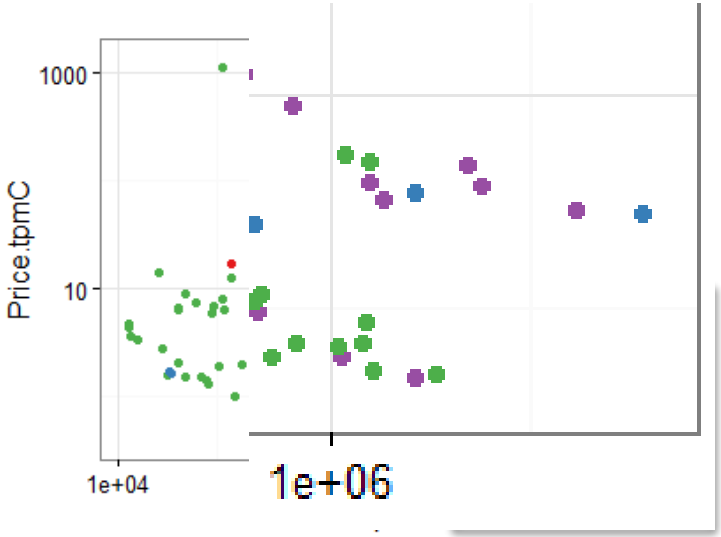
Benchmark Ergebnisse

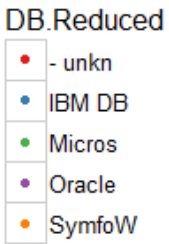




Das Spitzelnfeld ist eher vielfältig







Es gibt weder ein bestes OS, noch eine beste DB-Konfiguration