

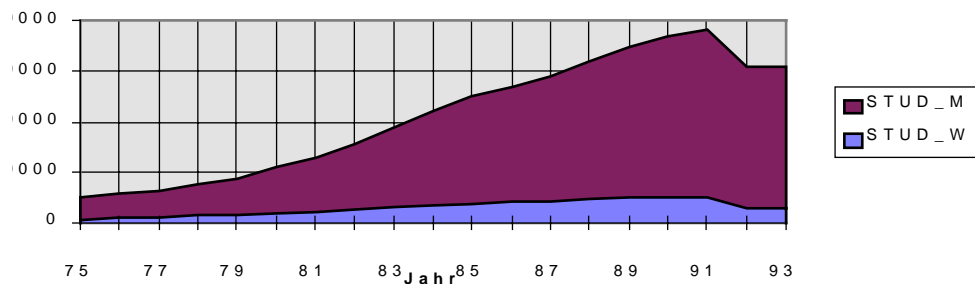
Visualisieren von Daten

-

Techniken und Programme

JAHR	STUD_GES	STUD_W	STUD_M	ERST_GES	ERST_W	ERST_M
1975	5003	682	4321	1439	209	1230
1976	5820	832	4988	1491	247	1244
1977	6374	970	5404	1525	263	1262
1978	7558	1418	6140	2156	449	1707
...						
1992	30889	2837	28052	5005	381	4624
1993	31005	2958	28047	4345	320	4025

Studentenzahlen Informatik



Visualisieren von Daten

Universität Osnabrück
- Rechenzentrum -
Frank Elsner
F.Elsner@rz.uni-osnabrueck.de
Albrechtstraße 28
D-49076 Osnabrück

Stand: 09/22/97
Version: 1.4

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Einleitung.....	1
Charakterisierung von Daten	3
Beschreibung der Datenquelle	3
Dimensionen von Definitions- und Wertebereich	3
Beispiele für Beobachtungen	4
Gitter	4
Techniken zur Visualisierung und Beispiele	6
Visualisierungsprogramme	8
Anwendungsgebiete.....	8
Funktionsumfang	8
Benutzerschnittstellen	10
Literaturhinweise	11

Einleitung

In diesem Skript erhalten Sie eine Einführung in Techniken zur Visualisierung numerischer Daten.

Die zentrale Aufgabe der Visualisierung besteht darin, große Mengen numerischer Daten (Zahlenfriedhöfe?) in einer **aussagekräftigen grafischen Form** darzustellen.

Data visualization is about comprehension, not graphics. Think of it as a range of techniques that enable you to display abstract numerical data and statistics in graphical form.[2]

Im Zusammenhang mit der Visualisierung wird häufig die folgende einprägsame Formel verwendet:

*1 p = 100 w
One picture is worth a thousand words.*

Bei der zunehmenden Fülle der zu verarbeitenden Daten wie z.B. bei der Wettervorhersage bieten Visualisierungstechniken oft die einzige Möglichkeit, in den Daten **Muster** oder **Strukturen** zu erkennen und hieraus Zusammenhänge abzuleiten. Beim Durchforsten riesiger Datenmengen ist es oft die legendäre "Nadel im Heuhaufen", die gefunden werden will.

Die Vorteile der grafischen Darstellung von Daten gegenüber einer tabellarischen Darstellung beruhen auf der bemerkenswerten Fähigkeit des menschlichen Gehirns zur Verarbeitung grafischer Informationen:

With half the neurons in the brain dedicated to visual processing, images provide the greatest mental bandwidth. Thus, by offering a picture of the data and its internal relationships, visualization makes it easier for you to understand information that's too complex to perceive numerically [2].

Die Visualisierung der Daten steht in unmittelbarem Zusammenhang mit der Analyse der Daten, da Visualisierung und Analyse einander ergänzen. Die **Visualisierung und Analyse wissenschaftlich-technischer Daten** wird zusammengefaßt als **visuelle Datenanalyse** bezeichnet, wobei hierbei vorausgesetzt wird, daß sowohl visuelle als auch mathematisch/statistische Auswertungen komplementär eingesetzt werden.

Der Vorteil der visuellen Datenanalyse besteht darin, daß sich grafische Informationen und Zahlen ergänzen. Es wird z.B. häufig zunächst eine **explorative Datenanalyse** durchgeführt, bei der sich (hoffentlich!) augenfällige Hinweise auf Zusammenhänge ergeben. Im Anschluß an die explorative Datenanalyse erfolgt eine analytische Untersuchung (statistische oder mathematische Datenanalyse), an die sich erneut eine visuelle Datenanalyse mit modifizierten (z.B. Skalierung) oder reduzierten Daten (z.B. Teilmenge) anschließen kann.

Das folgende kleine Beispiel soll den Unterschied zwischen tabellarischen und grafischen Informationen verdeutlichen. Die folgende Tabelle enthält Datenmaterial über die Entwicklung der Studentenzahlen in Deutschland im Fach Informatik, wobei jeweils nach Frauen und Männern und Erstsemestern und Studentenzahlen gesamt unterschieden wird.

JAHR	STUD_GES	STUD_W	STUD_M	ERST_GES	ERST_W	ERST_M
1975	5003	682	4321	1439	209	1230
1976	5820	832	4988	1491	247	1244
1977	6374	970	5404	1525	263	1262
1978	7558	1418	6140	2156	449	1707
...						
1992	30889	2837	28052	5005	381	4624
1993	31005	2958	28047	4345	320	4025

Abb. 1: Studentenzahlen im Fach Informatik

Das folgende Liniendiagramm stellt den zeitlichen Verlauf in visueller Form dar. Die grafische Darstellung unterstützt den Betrachter z.B. beim Vergleich der Werte für weibliche und männliche Studenten oder bei der Beurteilung langfristiger Trends.

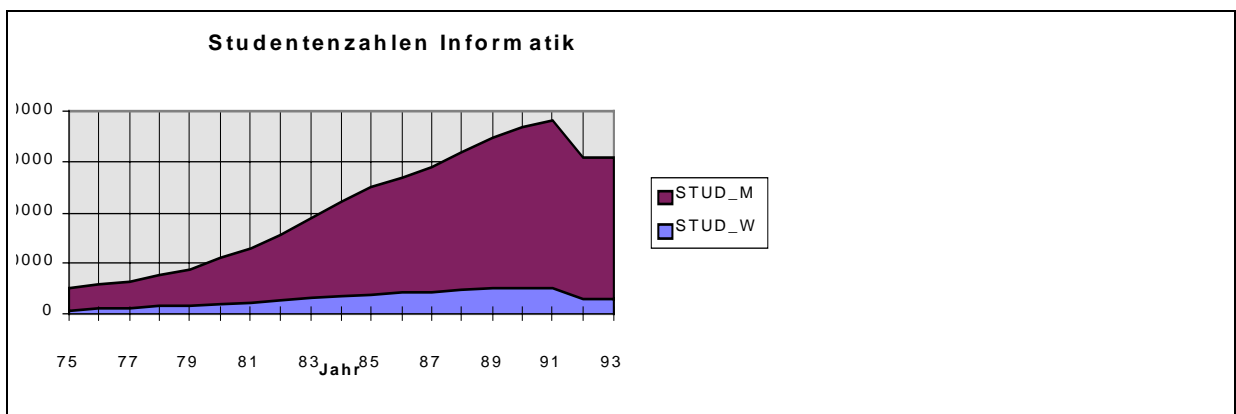


Abb. 2: Liniendiagramm (Excel für Windows 5.0)

Ein komplexes Szenario in der **Umwelt-Meßtechnik** könnte z.B. folgendermaßen aussehen:

Das städtische Umweltamt der (fiktiven) Stadt Schlackendorf läßt rund um die Uhr jeweils alle 5 Minuten an verschiedenen Standorten Luftproben nach 40 Kriterien auswerten (Temperatur, Kohlenmonoxid, Windstärke usw.). Die Auswertung der Daten soll u.a. folgenden Fragen beantworten:

Wann und wo wurden die behördlichen Grenzwerte überschritten? Besteht ein Zusammenhang zwischen Temperatur und Kohlenmonoxidgehalt? Fallen in der Nähe der von der Lokalpresse scharf attackierten (fiktiven) Petrowerke GAU Chemie anscheinend zufällig, aber ungewöhnlich oft, Meßstationen aus? [3]

Charakterisierung von Daten

In diesem Kapitel erhalten Sie einen Überblick, wie Sie eine Datenquelle beschreiben und in ein Visualisierungsprogramm einlesen können.

Beschreibung der Datenquelle

Eine visuelle Datenanalyse beginnt mit dem Einlesen (Importieren) des maschinenlesbar vorliegenden Datenmaterials in das Visualisierungsprogramm¹.

Hierzu erstellen Sie zunächst eine **Beschreibung des Datenmaterials**, die häufig als **Vorspann** oder **Meta-Information** (Daten **über** Daten) (*meta data, header information, data dictionary*) bezeichnet wird. Meta-Daten und eigentliche Daten befinden sich i.d.R. in unterschiedlichen Dateien bzw. in einer Datenbank. Mögliche Informationen lauten:

- Dateiname der Datenquelle
- Anzahl der Beobachtungen (Fälle)
- Bezeichnung und Bedeutung der unabhängigen Variablen pro Beobachtung
- Bezeichnung und Bedeutung der abhängigen Variablen pro Beobachtung
- Layout der Datenquelle
(z.B. Anordnung der Variablen in Spalten, Trennzeichen zwischen Datenwerten, Reihenfolge der Variablen)
- Maßeinheit und ggf. Meßgenauigkeit, Datentyp, Grenzen des Definitions- und Wertebereichs, Kennzeichnung von fehlenden Werten

Dimensionen von Definitions- und Wertebereich

Die Dimension des Definitionsbereiches n ist die Anzahl unabhängiger Variablen pro Beobachtung. Analog ist die Dimension des Wertebereiches m die Anzahl abhängiger Variablen pro Beobachtung.

Beobachtungen, deren Variablen numerisch sind, lassen sich als (Beobachtungs-) Punkte $(\mathbf{x}_i, \mathbf{y}_i)$, $i=1, \dots, k$, im \mathbb{R}^{n+m} interpretieren, wobei (x_1, \dots, x_n) für die unabhängigen und $\mathbf{y}=(y_1, \dots, y_m)$ für die abhängigen Variablen steht.

Beobachtungen, für die $n=m=1$ gilt, lassen sich z.B. als Punkte in der Ebene darstellen, Beobachtungen mit $n=3$ $m=3$ als räumliche Vektoren.

Der Zusammenhang zwischen den unabhängigen Variablen \mathbf{x} und den abhängigen Variablen \mathbf{y} wird entweder über eine bekannte **Funktion oder Parametrisierung** f mit $\mathbf{y}=f(\mathbf{x})$ oder über eine **Messung** hergestellt. Die spezielle Dimension 0 wird für ei-

¹ Das Datenmaterial muß vorher auf irgendeine Art und Weise in maschinenlesbare Form gebracht werden, sei es durch Meßgeräte, die Datensätze maschinenlesbar abspeichern oder durch Erfassen des Datenmaterials mit einem Editor oder per OCR. Das Resultat ist in jedem Fall eine Datei, die häufig als Rohdaten-Datei (*raw data file*) bezeichnet wird.

nen fehlenden Definitionsbereich D verwendet, d.h. es gibt nur Beobachtungen mit abhängigen Variablen, die in einer willkürlichen Reihenfolge sortiert werden.

Bei einem funktionalen Zusammenhang liegt i.d.R. ein **kontinuierlicher** Definitionsbereich D vor und damit bei stetigen Funktionen auch ein kontinuierlicher Wertebereich R , bei einzelnen Meßdaten (ohne Interpolation) i.d.R. ein **diskreter** Definitionsbereich D und damit ein diskreter Wertebereich R vor.

Beispiele für Beobachtungen

Ein Beispiel für eine Beobachtung mit ausschließlich einer abhängigen Variablen ist das Ergebnis einer Wahl, z.B. die von einem Wähler gewählte Partei $\langle xyz \rangle$ bei der Bundestagswahl. In diesem Fall ist die Dimension des Definitionsbereiches 0 (kategoriale Daten).

Ein weiteres Beispiel sind Beobachtungen (Meßwerte), die an einer Wetterstation für Temperatur, Windstärke und Niederschlag ermittelt werden. Gegeben sind Zeitpunkt der Messung und Ort (geografische Länge und Breite) als unabhängige Variablen.

Visualisierungsprogramme unterstützen i.d.R. mehrere Methoden, um Datenquellen einzulesen. In Zeilen angeordnete Beobachtungen können häufig direkt eingelesen werden, wobei die erste Zeile die Variablennamen enthalten kann (Beispiele: Excel für Windows 5.0, SPSS für Windows 6.01).

Gitter

Die Werte für die unabhängigen Variablen definieren ein Gitter im n -dimensionalen Raum (n : Dimension des Definitionsbereiches), das sich häufig auf räumliche oder zeitliche Koordinatensysteme (Länge x , Breite y , Tiefe z , Zeitpunkt t) beziehen läßt:

Gitterstrukturen	Skizze	Beispiel
Reguläres Gitter (<i>regular grid</i>) (2D: Quadrat oder Rechteck, 3D: Würfel oder Quader)	<pre> +-----+ +-----+ +-----+ </pre> (+: Meßpunkt)	Messung physikalischer Größen wie Druck oder Temperatur auf einem regelmäßigen Gitter

<p>Deformiertes Gitter (transformiertes regelmäßiges Gitter) (<i>deformed regular grid</i>)</p> <p>(2D: Dreieck, Raute, ... 3D: Quader, Prisma, ...)</p>	<pre>+---+---+ \ \ \ +---+---+</pre> <p>(+: Meßpunkt)</p>	<p>Messung der Druckverteilung auf einer gekrümmten Oberfläche (z.B. auf einem Kotflügel)</p>
<p>unregelmäßig verteilte, diskrete Meßpunkte (<i>ungridded, scatterplot</i>)</p>	<pre>+ + + + + + + + +</pre> <p>(+: Meßpunkt)</p>	<p>Messung von Temperatur und Niederschlag in einigen ausgewählten Städten</p>

Abb. 3: Typen von Gittern

Die Geometrie des Gitters und die verwendeten 1D- (Linien), 2D- (Flächen) oder 3D- (Volumen) Elemente wie z.B. Punkte auf einem Intervall, Eckpunkte von Rechtecken oder Quadern bestimmen u.a. die möglichen Interpolationsmethoden, um diskrete Messungen kontinuierlich fortzusetzen.

Techniken zur Visualisierung und Beispiele

In diesem Kapitel erhalten Sie einen Überblick, wie Beobachtungen in Abhängigkeit von der Dimension von Definitions- und Wertebereich visuell dargestellt werden können.

Die folgende Tabelle enthält Visualisierungstechniken für numerische Variablen, die Sie abhängig von der Dimension des Definitionsbereichs n und der Dimension des Wertebereichs m anwenden können. Zu unterscheiden ist jeweils zwischen kontinuierlichen Bereichen (funktionaler Zusammenhang) und diskreten Bereichen (Messung).

Typ	n/m	Visualisierungstechnik für funktionalen oder interpolierten Zusammenhang	Visualisierungstechnik für diskrete Beobachtungen (Meßwerte)	Beispiele
0/1	n=0, m=1 (i, y_i)	Liniendiagramm (x-Achse kategorial)	Streudiagramm, Balkendiagramm, Tortendiagramm, Boxplot	Wahlergebnis nach Parteien
0/2	n=0, m=2 (i, y_1, y_2)	x-y-z-Streudiagramm (x-y-Achse kategorial)	gestapeltes Balkendiagramm, gruppiertes Balkendiagramm	Umsätze nach Region und Produkt
1/1	n=1, m=1 (x, y)	Graph einer Funktion $y=f(x)$	x-y Streudiagramm, Histogramm	Niederschlag nach Monat, Umsatz nach Monat
1/2	n=1, m=2 (x, y_1, y_2)	Überlagerte Graphen für (y_1, y_2), d.h. $y = (y_1, y_2) = (f_1(x), f_2(x))$	überlagerte x-y Streudiagramme	Niederschlag und Temperatur nach Monat
2/1	n=2, m=1 (x_1, x_2, y)	Funktionsgebirge von $y=f(x_1, x_2)$, farbige Konturlinien, farbige Konturflächen	x-y-z-Streudiagramm, x-y-Streudiagramm, mit Pseudo-Farbe für z,	Topologische Karte (Farbe repräsentiert Höhe über NN)

2/2	$n=2, m=2$ (x_1, x_2, y_1, y_2)	Funktionsgebirge mit Pseudo-Farben (<i>heat mapping</i>) Stromlinien Vektoren	x-y-z-Streudiagramm mit Glyphen (z.B. Vektoren), x-y-z-Streudiagramm mit Pseudo-Farben für 4. Dimension, Vektoren	Geschwindigkeit v , $v=(v_1, v_2)$, in der Ebene
3/1	$n=3, m=1$ (x_1, x_2, x_3, y)	Transparente Kontur- flächen (Isoflächen) 3D-Schnitte	x-y-z-Streudiagramm mit Pseudo-Farben	Temperatur- verteilung in einem Volumen
3/3	$n=3, m=3$ $(x_1, x_2, x_3,$ $y_1, y_2, y_3)$	Stromlinien	x-y-z-Streudiagramm mit Glyphen (z.B. Vektoren)	Windstärke als 3D-Vektor im Raum

Abb. 4: Visualisierungstechniken

Die in der Tabelle verwendeten Begriffe **Pseudo-Farben** und **Glyphen** haben folgende Bedeutung:

Pseudo-Farben

Ein bestimmter Farbton repräsentiert einen numerischen Wert oder einen Bereich von Werten. Falls die Farbe rot (heiß) zur Darstellung großer Werte und die Farbe blau (kalt) für niedriger Werte verwendet wird, wird diese Technik auch als **Heat Mapping** bezeichnet.

Glyphen

Als Verallgemeinerung von (Vektor-) Pfeilen werden graphische Symbole wie z.B. Flaggen, Kugeln oder Quader verwendet, deren Größe, Ausrichtung und/oder Farbe numerische Werte oder einen Bereich von Werten repräsentiert.

Visualisierungsprogramme

In diesem Kapitel erhalten Sie einen Überblick über Visualisierungsprogramme.

Anwendungsgebiete

Programme zur Visualisierung von Daten lassen sich (mit Überschneidungen) in folgende Anwendungsgebiete einteilen:

Anwendungsgebiet)	englische Bezeichnung	Beispiele
Computeralgebrasystem mit integrierter Grafik	<i>symbolic math package</i>	Mathematica Maple Derive Macsyma Axiom MuPAD
(interaktive) naturwissenschaftlich-technische Datenanalyse und Datenvisualisierung	<i>(interactive) scientific data analysis</i>	Matlab Octave Data Explorer Origin PV-Wave Stanford Graphics Mathcad
statistische Datenanalyse und Visualisierung	<i>statistics package</i>	SPSS Statistica SAS
Tabellenkalkulation mit integrierter Geschäftsgrafik	<i>business packages (spreadsheet and presentation graphics)</i>	Excel Harvard Graphics Lotus 1-2-3
spezielle Anwendungsgebiete wie z.B. Chemie, Geographie, Prozeßdatenverarbeitung, CAD/CAM/FEM	<i>special purpose</i>	ARC/Info AutoCAD SAS/GIS

Abb. 5: Anwendungsgebiete

Funktionsumfang

Ein Programm, das numerische Daten verarbeitet und visuell darstellt, läßt sich funktional in folgende Aufgabenbereiche gliedern:

Aufgabenbereich	englische Bezeichnung	Stichworte
Einlesen von (Roh-) Datenquellen	<i>Reading raw data</i>	ASCII freies Format bzw. festes Format, binäre Daten
Importieren von Daten im (programm-) internen oder in einem Fremdformat	<i>Importing system files or reading database or spreadsheet formats</i>	Formate wie Excel .XLS, dBase .DBF
Auswählen von Spalten, Filtern, Verbinden und Transformieren von Daten	<i>Selecting columns, filtering, merging and transforming data</i>	Auswahl einer Teilmenge, Einteilung in Klassen, Aggregieren
Zusammenfassen und einfaches Analysieren von Daten	<i>Summarizing and analyzing data</i>	Regression, Statistiken wie Mittelwert oder emp. Standardabweichung
Darstellen in 2D	<i>Plotting 2D graphics</i>	Streudiagramm, Linienzug, Fehlerbalken, Tortendiagramm
Darstellen in 3D	<i>Plotting 3D graphics</i>	Funktionsgebirge, Kurven und Flächen im Raum, Konturlinien, Schnitte
Darstellen in nD	<i>Plotting nD graphics</i>	Vektoren im Raum, Glyphen, Stromlinien, Pseudo-Farben (<i>Heat Mapping</i>)
Animieren in 2D/3D	<i>Animate</i>	Vor-/Rücklauf, Standbild, Anzahl <i>frames</i>
Interaktives Bearbeiten	<i>Interactive manipulation</i>	Zoom, Perspektive, Projektion, Rotation, Schnitt, Einfärben, Transparenz, Schattierung, Masierung
Hinzufügen von Gestaltungselementen	<i>Adding layout elements</i>	Titel, Legende, Anmerkungen, Achseneinteilung und -beschriftung, Gitternetz
Exportieren von Grafiken in Austauschformate zur Weiterverarbeitung	<i>Exporting to metafile formats</i>	Formate wie TIFF, EPS, CGM, WMF oder BMP
Ausdrucken von Grafiken auf Plotter, Drucker etc.	<i>Printing to output devices</i>	PostScript, HP LaserJet, HP Deskjet, Plotter
Spezielle Anforderungen	<i>Special requirements</i>	Manipulation in Realzeit

Abb. 6: Aufgabenbereiche

Benutzerschnittstellen

Programme (bzw. Funktionsbibliotheken) zur Visualisierung von Daten lassen sich hinsichtlich ihrer Benutzerschnittstelle folgendermaßen klassifizieren:

Benutzerschnittstelle	Beschreibung	Beispiel
Einbetten von Funktionsaufrufen in eine Programmiersprache (i.d.R. C oder Fortran)	Der Benutzer programmiert ein vollständiges (Rahmen-) Programm (<i>driver program</i>) und verwendet vorgefertigte Grafik-Funktionen der jeweiligen Bibliothek.	NAG Graphics Libr. GKS-2D/3D Libr., OpenGL PHIGS (PEX) Libr.
interpretierte Kommandosprache (Dialog) (<i>command language</i>)	Der Benutzer gibt im Dialog oder im Batch mit dem System Kommandos ein und erhält die gewünschten Ergebnisse (Text) bzw. Grafiken angezeigt bzw. als Grafik-Datei zur weiteren Bearbeitung.	Maple Mathematica gnuplot PV-Wave CL
Menüsystem (<i>spreadsheet interface, WIMP - Windows, Icons, Mice, Pointers</i>)	Der Benutzer arbeitet mit einer grafischen Benutzeroberfläche, die in einem Datenfenster die aktuellen Datensätze tabellarisch anzeigt. Die weitere Verarbeitung erfolgt durch Auswahl von Menüpunkten oder Piktogrammen, ggf. unterstützt durch "Assistenten".	Excel SPSS Origin
Visuelle Programmierung (graphischer Editor für Datenfluß-Netzwerke und Werkzeug-Bibliothek) (<i>visual programming</i>)	Der Benutzer konstruiert in einem Grafik-Editor ein visuelles Programm (Datenfluß-Netzwerk mit Symbolen und Verbindungslinien). Jedes Symbol im Netzwerk repräsentiert ein Werkzeug (wie z.B. Einlesen, Einfärben, Berechnen von Isolinien, Ausgeben), jede Verbindung zwischen Symbolen repräsentiert einen Datenfluß.	Data Explorer apE AVS IRIS Explorer

Abb. 7: Benutzerschnittstellen

Literaturhinweise

1. *J.M. Chambers, W. S. Cleveland, B. Kleiner, P. A. Tukey:*
Graphical Methods for Data Analysis.
Wadsworth Int. Group and Duxbury Press, Boston, 1987
2. *BYTE, April 1993:*
State of the Art - Visualization,
Seite 120-149.
3. *Precision Visuals:*
Visuelle Daten-Analyse.
4. *Betsy Coming:*
Designing and Producing Effective Graphs with SAS/Graph Software I
SAS Observations, 1. Quarter 1994.
5. *Betsy Coming:*
Designing and Producing Effective Graphs with SAS/Graph Software II
SAS Observations, 2. Quarter 1994.
6. **Hersteller-Dokumentation der genannten Visualisierungsprogramme**