INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Fortgeschrittenenpraktikum

Web-basierte dynamische Visualisierung klinischer Daten

Marta Petrova

Aufgabensteller: Prof. Dr. Heinz-Gerd Hegering
Prof. Dr. Albrecht Neiß

Betreuer: Michael Scholz
Martin Daumer
Michael Brenner
Helmut Reiser

Abgabetermin: August 2002

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Am Sylvia Lawry Centre for Multiple Sclerosis Research (SLCMSR) an der Technischen Universität München wird anhand einer weltweit einmaligen Datensammlung von klinischen Studien und Registerdaten zur Multiplen Sklerose ein neuer Ansatz in der Therapieforschung begangen. Ein wesentliches Ziel des Zentrums ist die Entwicklung von "virtuellen Placebo-Patienten", die es gestatten, die Wirksamkeit neuer Medikamente ohne echte Placebogruppen nachzuweisen. Die virtuellen Placebo-Gruppen müssen mit Hilfe statistischer Methoden aus der Datensammlung errechnet werden. Danach sollen die Daten der verschiedenen Placebo-Gruppen analysiert und visualisiert werden. Das Sylvia Lawry Centre setzt Macromedia ColdFusion als Web-Application Server ein [DEVCF], der in der neuen Version MX auch J2EE kompatibel ist. Durch die Integration der Flash-Technologie wurde gleichzeitig eine umfangreiche Graphikbibliothek geschaffen, mit der schnell Charts realisiert werden. Mit dieser Technologie wird das Centre der Öffentlichkeit und den weltweit verteilten Experten des wissenschaftlichen Beirats aktuelle Einblicke in die weltweit größte Datenbank zu Multiplen Sklerose (MS) Studien geben. Die Daten sollen dabei nicht "roh", sondern in Form von statistischen Visualisierungen dargestellt und über ein Web-Interface zugänglich gemacht werden. Dabei sollen die Daten zum Zeitpunkt des Zugriffs aus der Datenbank abgerufen, vorverarbeitet und aggregiert werden. Diese Daten müssen dann automatisiert in Form interaktiver Visualisierungstools dargestellt werden.

1.2 Aufgabenstellung

Die Aufgabe dieses Fortgeschrittenen-Praktikums besteht darin, eine Bibliothek von webbasierten Tools zur Datenaggregation und -vorverarbeitung, sowie zur intelligenten, automatisierten und interaktiven Visualisierung von

grundlegenden Datenstrukturen, deskriptiven Statistiken und Drill-Down-Objekten auf dem Web-Application Server ColdFusion MX entwickeln. Diese Bibliothek soll folgende Aufgaben lösen:

1. Daten bereinigen
2. Datenstrukturen analysieren
   - Datentyp festlegen
   - Verteilungsmuster erkennen
   - Statistische Größen berechnen (z.B. Median, Mittelwerte, Quantile, Regressionsparameter, Konfidenzintervalle)
3. Diskretisierung der Daten
4. Daten zur Visualisierung über die CF-Funktion CFCHART aufbereiten
5. Benutzergesteuerte Anpassung der Datenauswahl und -darstellung implementieren

1.3 Überblick über die Ausarbeitung

Innerhalb der Ausarbeitung zum Fortgeschrittenen-Praktikum wird anfangs die Anforderungsanalyse beschrieben, es folgt eine kurze Beschreibung einiger statistischerBegriffe, die für das Praktikum relevant sind. Anschließend folgt eine Beschreibung der Architektur von ColdFusion MX. In Kapitel 5 folgt die Beschreibung des Entwurfs und der Funktionalitäten der Bibliothek.

Kapitel 2

Anforderungsanalyse

Wie bereits in der Einleitung geschildert, besteht die Aufgabe dieses Fortgeschrittenen-Praktikums in der Entwicklung einer Bibliothek von webbasierten Tools, mit deren Hilfe Daten in einer statistisch adäquaten Form visualisiert werden können. Damit die in den Daten enthaltene Information übersichtlich und graphisch richtig dargestellt wird, benutzt man die deskriptive Statistik [WC99]. Die deskriptive Statistik hat die Aufgabe, empirisch gewonnene Daten zu ordnen, durch bestimmte Maßzahlen zusammenzufassen und graphisch oder tabellarisch darzustellen. Der Benutzer erhält als Output die graphische Darstellung der Daten. Alle Berechnungen, Datenvorverarbeitungen und Aggregationen sollen im Hintergrund transperent für den Benutzer ablaufen. Die Programm-Bibliothek soll über die im folgenden beschriebenen Funktionalitäten verfügen.

2.1 Datenabfragen

Die erste Aufgabe ist die Abfrage der Daten aus der Datenbank. Dabei werden die Daten in "roher" Form von der Datenbank rausgenommen.

2.2 Daten bereinigen

Es kann passieren, dass die Daten in der Datenbank bereinigt werden müssen. Der Grund dafür kann sein, dass es fehlerhafte Daten gibt (falsch getippte Daten), Daten mit fehlenden Werten (Datenbankfelder mit nicht eingegebenem Wert), Daten mit nicht passendem Datentyp. Diese Daten müssen erkannt und entfernt werden. Da im konkreten Anwendungsfall eine "saubere" Datenbank zur Verfügung steht, reduziert sich hier der Aufwand. Relevant bleibt u.a. der Umgang mit fehlenden Werten.

2.3 Datenstrukturen analysieren

Eine weitere wichtige Funktionalität der Bibliothek soll die Datenanalyse sein. Zu der Datenanalyse gehört auch die Festlegung des Datentyps, der für die spätere Visualisierung der Daten wichtig ist. Wenn es sich z.B. um Textdaten handelt, dann impliziert das, dass die Daten qualitativ (kategorial) sind und dass sie mit Hilfe eines Diagramms mit festen Kategorien dargestellt werden. Ein weiterer wichtiger Punkt bei der Datenanalyse ist die Erkennung der Verteilungsmuster. Es sollte schon vor der Datenvisualisierung klar sein, ob es sich um eine symmetrische (z.B. Normalverteilung), eine rechtsschiefe oder eine linksschiefe Verteilung handelt. Zur Datenanalyse gehört auch die Berechnung verschiedener statistischer Größen, wie Median, Mittelwert, Quantile oder Regressionsparameter, die unterschiedlich gut für unterschiedliche Verteilungstypen geeignet sind. Dabei sollen alle genannten statistischen Größen berechnet werden und bei Bedarf wird die geeignetste Größe benutzt.

2.4 Diskretisierung der Daten

Die Daten sollen bei Bedarf sinnvoll in Klassen (Kategorien) eingeteilt werden. Diese Einteilung hängt wieder vom Datentyp ab. Wenn die Daten aus stetigen Messgrößen bestehen, können die Daten klassifiziert werden. Dazu teilt man den gesamten Wertebereich der Daten in Intervalle ein, die Klassen genannt werden. Die Breite der einzelnen Klassen ist gleich. Ziel der Klassifizierung ist es, einerseits die tabellarische und graphische Darstellung übersichtlicher zu gestalten, ohne andererseits zuviel an Information zu verlieren. Bei qualitativen Daten sind die Klassen durch die Datenausprägungen vorgegeben.

2.5 Daten zur Visualisierung über die CF-Funktion CFCHART aufbereiten

Eine zentralle Aufgabe besteht in der Aufbereitung der Daten in eine Form, die von der ColdFusion Funktion CHCHART, die für das Zeichnen von Diagrammen in ColdFusion zuständig ist, nicht fehlinterpretiert werden. Zum Beispiel unterscheiden die graphischen Funktionen von ColdFusion keine diskreten Daten. Dabei kann es sehr leicht zu falscher Visualisierung der Daten kommen (siehe Abb. 4.3).

2.6 Benutzergesteuerte Anpassung der Datenauswahl und -darstellung implementieren

Der letzte Punkt der Anforderungen ist die Implementierung einer graphischen Benutzeroberfläche. Die Benutzeroberfläche muss leicht bedienbar sein. Der Benutzer muss auf einen Blick alle in der Datenbank vorhandenen Attribute sehen. Von diesen Daten kann er auswählen, welche er visualisieren möchte. Zur Visualisierung der Daten wurden im Rahmen des Fortgeschrittenen Praktikums prototypisch zwei Typen von Diagrammen implementiert: das Histogramm und der Scatter-Plot. Beim Histogramm sollen zusätzliche Funktionalitäten wie Achsenskalierung, Drill-Down Funktionalität (per linken Mausklick auf einen Histogrammbalken gelangt man zum Histogramm der ausgewählten Klasse), Änderung der Klassenanzahl und Anzeigen des Konfidenzintervalls implementiert werden. Beim Scatter-Plot sollen zusätzlich die Regressionsgerade und den Konfidenzbereich dargestellt werden.

Kapitel 3

Kurze Einführung in die Statistik

In diesem Kapitel werden einige Begriffe der Statistik beschrieben, die später bei der Implementierung eine wichtige Rolle spielen.

3.1 Grundlegende Begriffe

Der Gegenstand einer Untersuchung, die Beobachtungseinheit (Patient, Organ usw.), ist durch eine Reihe von Eigenschaften oder Variablen, die Merkmale, gekennzeichnet [FH99]. Diese können verschiedene Eigenschaften oder Zahlenwerte, die Ausprägungen, annehmen. Die ermittelten Ausprägungen der Merkmale sind die Daten. Man unterscheidet qualitative und quantitative Merkmale. Qualitative Merkmale sind jene, die primär nicht durch Zahlenangaben erfassbar sind (typische Beispiele: Geschlecht, Blutgruppe). Quantitative Merkmale sind durch Zahlenwerte erfassbar und angebbar (Körpergröße, Gewicht).

3.1.1 Absolute/ Relative Häufigkeiten

Sei A ein qualitatives Merkmal mit den Ausprägungen $A_1, ...A_n$. In einer Stichprobe vom Umfang n habe man die Ausprägung A_i mit der absoluten Häufigkeit n_i beobachtet $(i = 1, 2, ..., k)$. Bei n_0 Beobachtungseinheiten aus der Stichprobe fehle die Angabe zum Merkmal A. Dann ist offenbar

$$n_1 + n_2 + ... + n_k = n - n_0.$$

Unter den relativen Häufigkeiten, genauer den adjustierten relativen Häufigkeiten, versteht man

$$h_i = \frac{n_i}{n - n_0} \qquad (i = 1, 2, ..., k).$$

Der Zusatz “adjustiert“ soll betonen, dass man bei der Berechnung nur diejenigen Beobachtungseinheiten berücksichtigt, bei denen die Angaben zum Merkmal A tatsächlich vorliegen. Meist werden die relativen Häufigkeiten in Prozent angegeben:

$$h_i = h_i \times 100\%$$

also z. B. 30 % statt 0.3.

3.1.2 Arithmetischer Mittelwert

Der arithmetische Mittelwert ist die am häufigsten gebrauchte Lagemaßzahl. Man erhält sie als durchschnittlichen Wert, indem man alle Werte $x_1, x_2, ..., x_n$ addiert und durch die Gesamtzahl der Daten n dividiert:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

3.1.3 Empirischer Median

Zur Berechnung des empirischen Medians müssen die Daten der Größe nach geordnet werden, d. h. man geht von der Urliste der Daten $x_1, x_2, ..., x_n$ zur Rangliste $x_{(1)} \leq x_{(2)} \leq ... \leq x_{(n)}$ über, indem man die Daten der Größe nach ordnet, (i) heißt Rangzahl. Die Rangzahl gibt den Platz auf der Rangliste an. (1) ist die Rangzahl des kleinsten Wertes, (n) ist die Rangzahl des größten Wertes. Der empirische Median ist der Wert “in der Mitte“ der Rangliste, d. h. die Hälfte der Messwerte sind kleiner bzw. größer als der Median.

$$\tilde{x} = x_{(\frac{n+1}{2})} \quad \text{falls} \quad n \quad \text{ungerade,}$$

und

$$\tilde{x} = x_{(\frac{n}{2})} \quad \text{falls} \quad n \quad \text{gerade.}$$

3.1.4 Empirische Varianz

Die Varianz s^2 bei quantitativen Merkmalen ist die mittlere quadratische Abweichung der Einzelwerte vom Mittelwert und beträgt

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

3.1.5 Empirische Standardabweichung

Die Standardabweichung ist die Quadratwurzel aus der Varianz

$$s = +\sqrt{s^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

3.1.6 Regression

Die Ermittlung der funktionalen Abhängigkeit verschiedener Variablen nennt man Regression. Bei der linearen Regression von Y auf X geht man davon aus, dass zwischen den beiden Merkmalen ein linearer Zusammenhang der Form

$$Y = b_0 + b_1 X$$

besteht. Die Abweichung der tatsächlich festgestellten Wertepaare von der durch die Gleichung beschriebenen Geraden führt man auf den Einfluss nicht erfasster Störgrößen zurück. Es stellt sich die Aufgabe, b_0 und b_1 vernünftig aus den Daten zu schätzen. Dieses Problem wurde mathematisch von C. F. Gauß (Methode der kleinsten Quadrate) gelöst. Man erhält die Schätzwerte b_1 bzw. b_0, die aus den Daten mithilfe der Formeln

$$b_1 = \frac{\frac{1}{n-1} \cdot \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

bzw.

$$b_0 = \bar{y} - b_1 \bar{x}$$

berechnet werden.
Die Gerade

$$y = b_0 + b_1 x$$

heißt (empirische) Regressionsgerade der Regression von Y auf X; b_1, der Anstieg der Regressionsgeraden, heißt (empirischer) Regressionskoeffizient.

3.1.7 Normalverteilung

Eine stetige Zufallsvariable X heißt mit Erwartungswert μ und Varianz σ^2 normalverteilt, wenn die Wahrscheinlichkeit dafür, dass X höchstens gleich x ist, durch das Integral der Gaußschen Fehlerfunktion gegeben ist, in Formeln:

$$P(X \leq x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \int_{-\infty}^{x} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \, dt$$

Hierfür schreibt man abkürzend $X : N(\mu, \sigma^2)$. $F(x) = P(X \leq x)$ ist die Verteilungsfunktion der Normalverteilung. Deren erste Ableitung

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$$

ist die Dichtefunktion der Normalverteilung. Das Bild der Dichtefunktion ist die bekannte Glockenkurve (Abbildung 3.1).

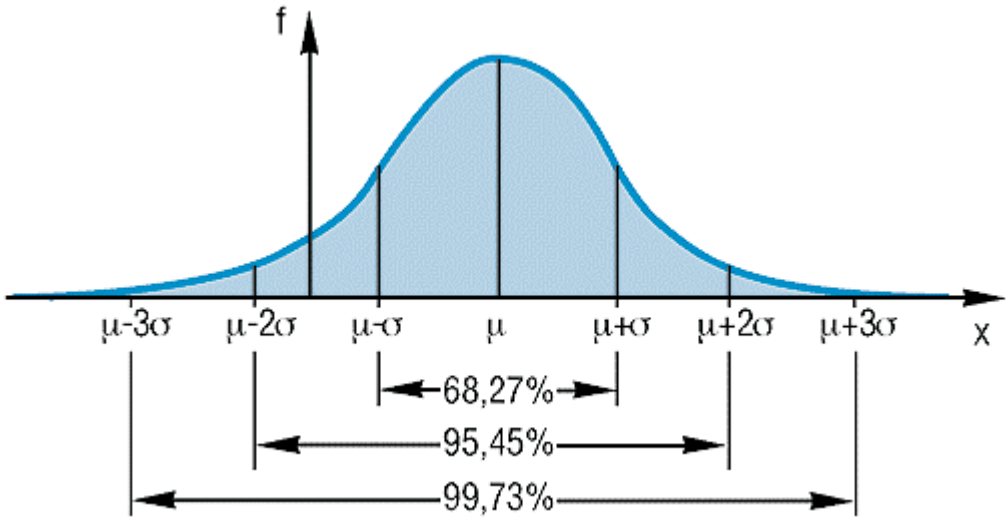

Abbildung 3.1: Dichtefunktion der Normalverteilung $N(\mu, \sigma^2)$

Die Verteilungsfunktion der Normalverteilung hat einen sigmoiden (s-förmigen) Kurvenverlauf (Abbildung 3.2).

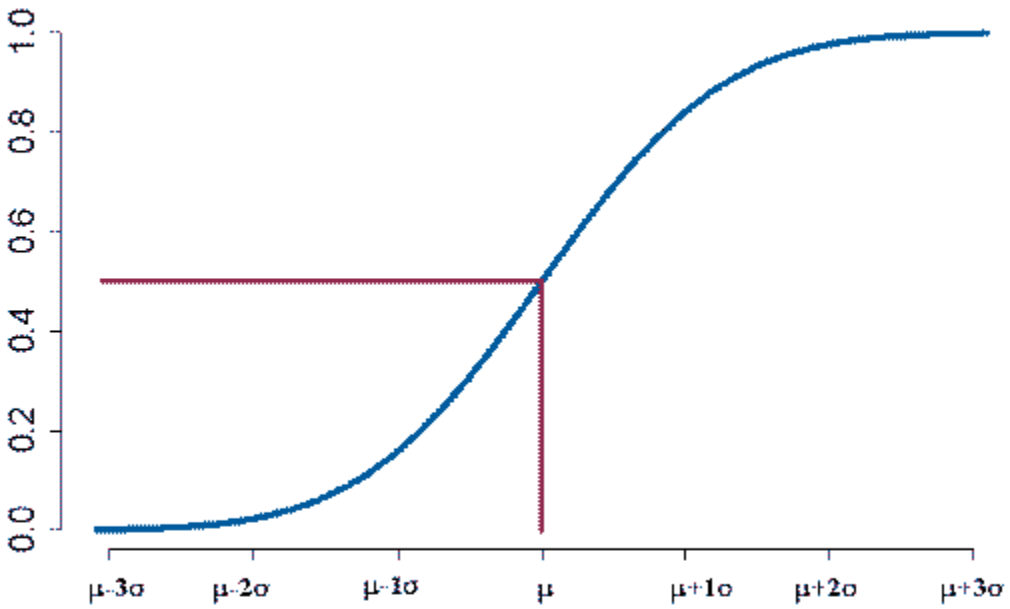

Abbildung 3.2: Verteilungsfunktion der Normalverteilung $N(\mu, \sigma^2)$

3.1.8 Konfidenzintervall

Der unbekannte Erwartungswert μ einer Normalverteilung $N(\mu, \sigma^2)$ wird durch den Mittelwert aus einer zufälligen Stichprobe geschätzt. Zu dem Mittelwert lässt sich ein Intervall, das sogenannte Konfidenzintervall, angeben, das den unbekannten Erwartungswert μ mit einer vorgegebenen Konfidenzwahrscheinlichkeit $1 - \alpha$ enthält. Die Intervallgrenzen t_u bzw. t_o berechnet man aus den Formeln

$$t_u = \bar{x} - \frac{\sigma}{\sqrt{n}}.z_{1-\frac{\alpha}{2}},$$

$$t_o = \bar{x} + \frac{\sigma}{\sqrt{n}}.z_{1-\frac{\alpha}{2}}.$$

Dabei ist σ die Standardabweichung der betrachteten Normalverteilung. n ist der Stichprobenumfang und $z_{1-\frac{\alpha}{2}}$ das $(1-\frac{\alpha}{2})$-Quantil der Standardnormalverteilung.

3.2 Visualisierungstechniken

Für die Visualisierung der Daten werden zwei grundlegende Typen von Diagrammen verwendet: das Histogramm und Scatter-Plot.

3.2.1 Das Histogramm

Das Histogramm dient zur graphischen Darstellung für die Häufigkeitsverteilung quantitativer Merkmale (Abbildung 3.3). Die Daten werden der Größe nach in Klassen eingeteilt und diese auf einer Grundlinie aufgetragen. Über jeder Klasse wird ein Rechteck gezeichnet, dessen Flächeninhalt proportional zur relativen Häufigkeit der auf die Klasse entfallenden Elemente ist. Ist die Anzahl der Daten groß genug und bei stetigen Merkmalen die Klassenbreite klein genug, so entspricht die Form des Histogramms der Dichtefunktion für die stetige Zufallsvariablen bzw. der Wahrscheinlichkeitsfunktion für diskrete Zufallsvariablen.

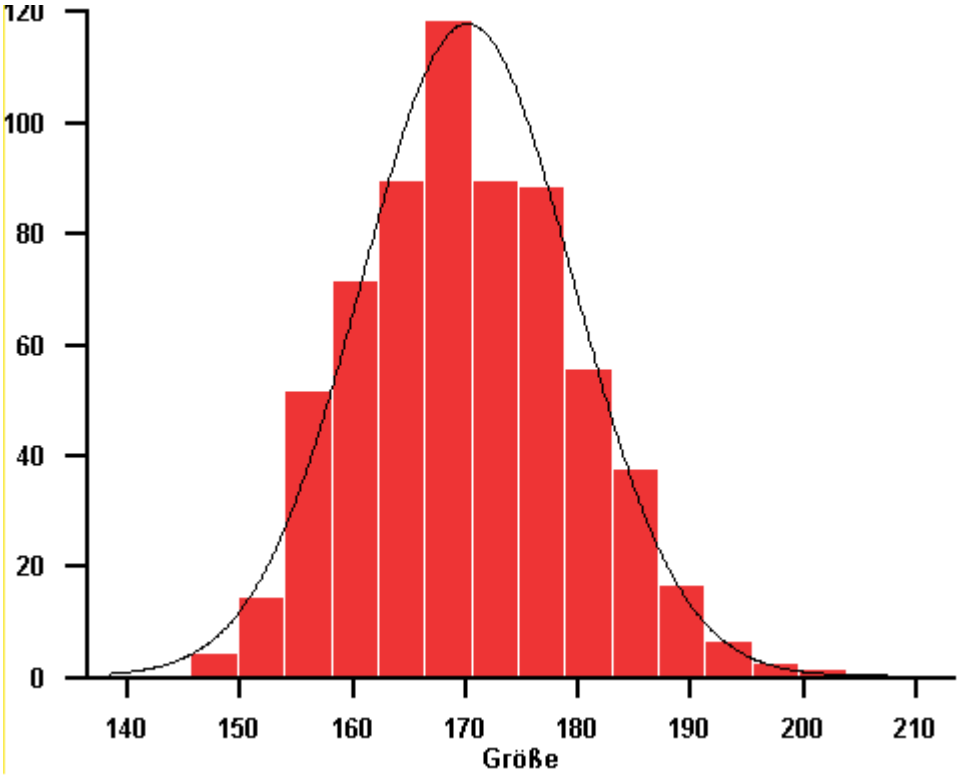

Abbildung 3.3: Histogramm mit Normalverteilungsdichte des Merkmals “Größe“

Das Histogramm selbst ist nicht fest definiert, so daß eine Aufgabe darin besteht, variable Parameter (wie z.B. Anzahl der Balken, Skalierung der x-Achse) für den Benutzer einfach veränderbar zu gestalten.

3.2.2 Scatter-Plot

Die Punktwolke dient zur graphischen Veranschaulichung des Zusammenhangs zweier stetiger Merkmale, die an n Beobachtungseinheiten erfasst wurden. Jede Beobachtungseinheit liefert genau einen Punkt für die Punktwolke. Der Zusammenhang zwischen zwei Merkmale wird mit der Hilfe der Regression untersucht (Abbildung 3.4). Mit der linearen Regression wird z.B. versucht diejenige Gerade zu bestimmen, welche den kleinsten Abstand zu den Punkten hat, d.h. die Lage der Punkte soll so genau wie möglich durch eine lineare Funktion (Gerade) beschrieben werden.

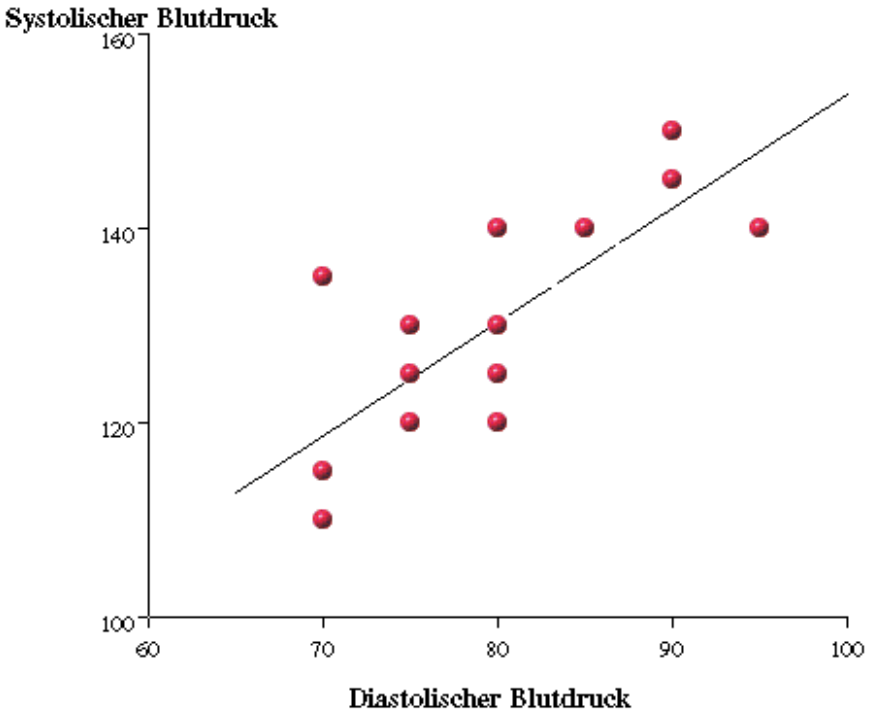

Abbildung 3.4: Scatter-Plot und Regressionsgerade

Kapitel 4

ColdFusion MX

Bei ColdFusion MX handelt sich um die derzeit neueste Version des Macromedia ColdFusion Servers, die zu Beginn des Praktikums noch als Beta-Version unter dem Code-Namen NEO vorlag. In diesem Kapitel werden einige grundsätzliche Merkmale der Architektur von ColdFusion genannt, sowie eine kurze Beschreibung der graphischen Funktionen gegeben, die für dieses Praktikum von großer Bedeutung sind.

4.1 Einleitung

Durch eingängige Skripterstellungsfunktionen, Möglichkeiten zur Verbindung mit Unternehmensdaten und integrierten Such- und Diagrammerstellungsfunktionen können Entwickler mit ColdFusion MX dynamische Webseiten, Content-Publishing-Systeme, E-Commerce-Seiten und anderes bereitstellen. Dem ColdFusion MX Server liegt ein auf der JRUN-Technologie basierender J2EE-Server zugrunde [NEO]. ColdFusion MX kann sowohl als Stand-alone-Server eingesetzt werden, als auch mit Java Application Servern, wie dem IBM WebSphere Application Server oder auch Servern von Sun iPlanet, zusammenarbeiten. Die ColdFusion MX-Umgebung unterstützt die Betriebssysteme Windows, Linux und Unix, kann Internetstandards und Komponentenmodelle integrieren, wie XML, Webdienste, Java, .NET/COM und COBRA.

4.2 Programmieren mit ColdFusion und CFML

ColdFusion Anwendungen bestehen aus einer Ansammlung von Templates oder Seiten, die die ColdFusion Markup Sprache (CFML) verwenden [CFREF]. CFML ist eine leicht zu erlernende Sprache, die mehr als 75 Tags sowie mehr als 240 eingebaute Funktionen umfasst. Die Syntax von ColdFusion ähnelt der von HTML und XML. Es werden wieder Tags verwendet, um Daten zu verarbeiten. ColdFusion erlaubt auch den Entwicklern, die

Sprachumgebung zu erweitern, indem sie ihre eigene Custom Tags oder benutzerdefinierte Funktionen (UDF) entwickeln oder COM, C++ und Java Komponenten integrieren. Neu in ColdFusion MX sind auch die ColdFusion Komponenten (CFC). Die CFC's sind wiederverwendbare Anwendungskomponenten, die von den ColdFusion Seiten abgerufen werden. Sie werden benutzt, um die Web-Anwendungen auf eine objektbasierte Weise zu organisieren.

4.3 Die Architektur von ColdFusion MX

ColdFusion MX ist J2EE kompatibel. Er baut dabei auf einer neuen Architektur auf, die die Verlässlichkeit und Skalierbarkeit der Java-Plattform übernimmt, jedoch nicht deren Komplexität.

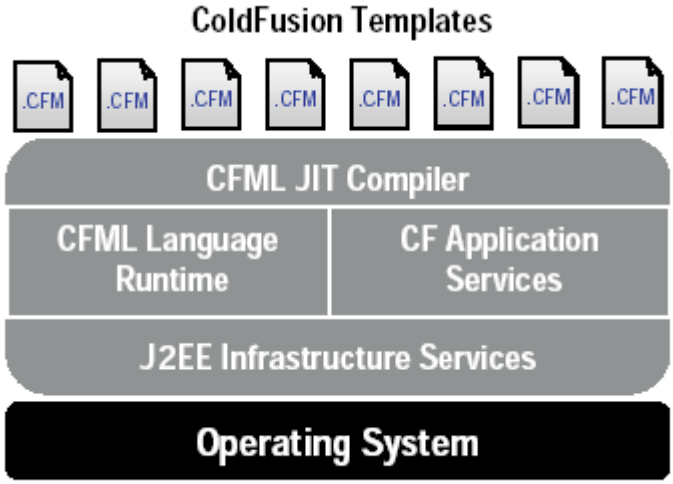

Abbildung 4.1: Die Architektur von dem ColdFusion MX Server

ColdFusion MX besteht aus vier Subsystemen (Abbildung 4.1):

- Java 2 Enterpise Edition (J2EE) Infrastructure Services
- CFML Just-In-Time (JIT) Compiler
- CFML Language Runtime
- CFML Application Services

4.3.1 Infrastructure Services

Die Infrastrukturdienstleistungen, die mit ColdFusion MX kommen, werden durch eine eingebettete Version von Macromedia JRun zur Verfügung gestellt. Die J2EE Engine in JRun entspricht dem J2EE Standard. Die Infrastrukturdienstleistungen umfassen die Datenbankanbindungen, die Sicherheitsdienste, das Interagieren mit dem Betriebssystem und das Steuern der HTTP, FTP und POP Protokolle.

4.3.2 CFML JIT Compiler

Anstatt p-Code (preprocessed Code), wie in den früheren Versionen von ColdFusion, erzeugt der CFML JIT Compiler in ColdFusion MX Java Bytecode. Fordert der ColdFusion Server ein ColdFusion Template zum ersten Mal an, wird dieses Template von dem CFML JIT Compiler in Java Bytecode übersetzt und im Cache gespeichert. Dieser Code wird dann auf dem Server von der CFML Language Runtime ausgeführt. Im Template enthaltenen Instruktionen wie Datenbankanfragen oder Formatierungen werden auch von dem CFML Language Runtime durchgeführt.

4.3.3 CFML Language Runtime und CFML Application Services

Der CFML Language Runtime umfasst die Verarbeitung aller ColdFusion Tags und Funktionen. Der Language Runtime bearbeitet auch die Interaktionen mit den Application Services, wie z.B. Charting und Graphing oder Ganztextsuche.

4.4 Charting und Graphing in ColdFusion MX

Die Fähigkeit, Daten in einem Diagramm anzuzeigen, kann die Dateninterpretation enorm vereinfachen. Anstelle einer einfachen Tabelle mit numerischen Daten werden die Daten anhand von Bar-Charts, Pie-Charts, Linien oder anderen Diagrammarten mit Farben, Untertiteln, mit zwei- oder dreidimensionaler Darstellung visualisiert. Der **cfchart** Tag zusammen mit **cfchartseries** und **cfchartdata** liefern viele unterschiedliche Diagrammarten. Mittels der Tagattribute lassen sich die Diagramme individuell anpassen. ColdFusion unterstützt 11 verschiedene Arten von Diagrammen in zwei oder drei Dimensionen. Die folgende Abbildung 4.2 zeigt ein zweidimensionales Beispiel aller Diagrammarten. (Bemerkung: Im zweidimensionalen Fall erscheinen das Bar und das Zylinder Diagramm gleich, sowie der Konus und die Pyramide.)

Abbildung 4.2: Diagrammarten

4.4.1 Erzeugen eines einfachen Diagramms

Um ein Diagramm zu erstellen, wird der **cfchart** Tag zusammen mit mindestens einem **cfchartseries** Tag eingesetzt, der wiederum einen oder mehrere **cfchartdata** Tags umfassen kann.

- **cfchart**: Spezifiziert den Container, in dem das Diagramm erscheint. Dieser Container definiert die Höhe, Breite, Hintergrundfarbe, Beschriftung und andere Eigenschaften des Diagramms. Jeder **cfchart** Tag beinhaltet mindestens einem **chchartseries** Tag
- **cfchartseries**: Spezifiziert eine Datenbankanfrage, die die Daten an das Diagramm liefert und/oder einen oder mehrere **cfchartdata** Tags, die einzelne Datenpunkte spezifizieren. Spezifiziert die Diagrammart, die Farben für das Diagramm und andere optionale Attribute
- **cfchartdata**: Spezifiziert optional einzelne Datenpunkte für den **cfchartseries** Tag

Der folgende Code zeigt einen Ausschnitt aus dem Programmcode des Fortgeschrittenen-Praktikums, wie man ein Histogramm erstellen kann:

```
<cfchart
   format="flash"
   chartHeight="#height#"
   chartWidth="#width#"
   font="Arial"
   fontSize="11"
   gridLines="3"
   show3d="no"
   showLegend="no"
   xAxisTitle="#attributes.beispiel#"
   yAxisTitle="#attributes.ylabel#"
   dataBackgroundColor="#d_d_blau#"
   >

   <cfchartseries
      type="bar"
      seriesColor="#gruen#"
      >
         <cfloop index="j" from="1" to="#arraylen(array_items_histogramm)#">
            <cfchartdata
               item="#array_items_histogrammm[j]#"
               value="#array_values_histogrammm[j]#"
            >
         </cfloop>
   </cfchartseries>

</cfchart>
```

`#Height#`, `#width#`,`#attributes.beispiel#`, `#attributes.ylabel#`, `#d_d_blau#` und `#gruen#` sind vordefinierte globale Variablen. `#array_items_histogrammm[j]#` und `#array_values_histogrammm[j]#` sind Arrays mit den Punkten für die X-Achse und deren Werte.
Für Datenbankanfragen wird der Tag **cfquery** benutzt. Er spezifiziert unteranderem den Anfragennamen, die Datenquelle, und ob die Anfrage gecacht werden soll. In dem **cfquery** Tag sind SQL Anweisungen eingeschlossen, mit denen die Datenbankanfrage spezifiziert wird. Die entsprechende Datenbankanfrage für den obigen Code sieht folgendermaßen aus:

```
<cfquery name="all_datasets" datasource="#dsn#"
         cachedwithin="#CreateTimeSpan(0,1,0,1)#">
   SELECT #column#
```

```
  FROM #table#
  WHERE (#colum# is not null)
</cfquery>
```

`#dsn#`, `#column#` und `#table#` sind wieder vordefinierte globale Variablen.

4.4.2 Nachteile der graphischen Funktionen von ColdFusion MX

Wenn man die bereitstehenden Funktionen von ColdFusion MX für Visualisierung von Daten benutzt, merkt man sehr schnell, dass die graphische Darstellung für viele Fälle nicht sehr adäquat ist. Z.B in Abbildung 4.3 ist das Problem der nicht äquidistanten Skalierung der X-Achse vor allem im Bereich zwischen 70 und 80 leicht zu erkennen.

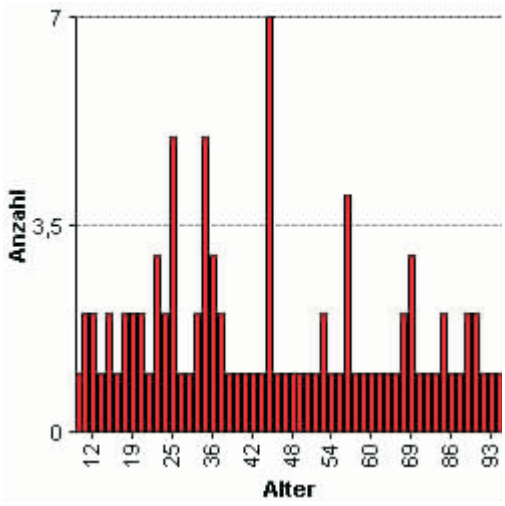

Abbildung 4.3: In dieser Form unbrauchbare Standardausgabe einer kontinuirlichen Datenreihe von Alterswerten mit ColdFusion MX

Um ein Histogramm zu erstellen, das z.B. die Datengruppierung nach dem Alter und auch noch den Prozentanteil jeder Altersgruppe darstellt (siehe Abb. 4.4), ist eine manuelle Vorverarbeitung der Daten erforderlich. Deshalb wäre hier ein Tool sehr hilfreich, das dies automatisiert und mächtige analytische Werkzeuge zur Datenuntersuchung und Datenanalyse liefert. Das genau ist das Ziel dieses Fortgeschrittenen Praktikums.

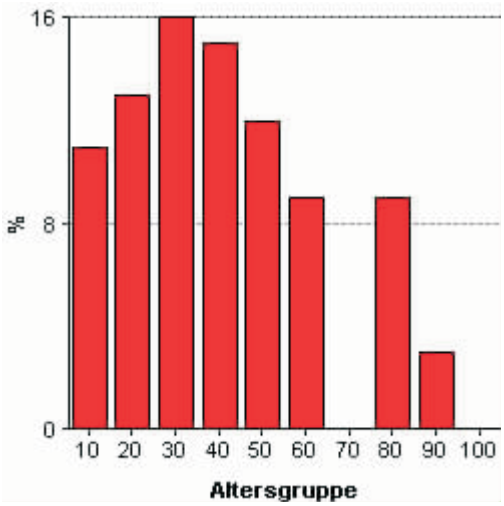

Abbildung 4.4: Sinnvolles Histogramm über die Altersverteilung- hier per Hand aus den Daten von Abb. 4.3 erstellt.

Kapitel 5

Die Bibliothek und ihre Komponenten

In diesem Kapitel werden die verschiedene Komponenten der Bibliothek, die im Rahmen des Fortgeschrittenen-Praktikums implementiert worden sind, und deren Funktionalitäten beschrieben. Die Bibliothek besteht aus ColdFusion Templates (CFML-Dateien) und ColdFusion Komponenten (CFC). Die Komponenten sind erweiterbare und ersetzbare Bestandteile der Bibliothek, die die Anforderungen an die Bibliothek aus Kapitel 2 erfüllen, wie die Vorverarbeitung der Daten, Aufbereitung der Daten für Visualisierung und das Berechnen verschiedener statistischer Größen. Sie enthalten eine oder mehrere Methoden, die in den Templates aufgerufen werden. In den ColdFusion Templates werden die von den Komponenten vorbereitenden Daten visualisiert.
Zentrale Rolle in der Bibliothek spielt die Datei `index.cfm`. Sie holt sich alle verfügbare Attribute aus der Datenbank und stellt diese Variablen dem Benutzer zur Auswahl zur Verfügung. Wenn der Benutzer ausgewählt hat, welche Daten visualisiert werden sollen, übergibt die Datei `index.cfm` die ausgewählten Daten dem Template `datentest.cfm`. Das Template prüft den Datentyp und von dem Datentyp abhängend ruft es das entsprechende ColdFusion Template auf. Das Template ruft die ColdFusion Komponente mit ihrer Methode auf und übergibt ihr die notwendigen Parameter, wie Datenbanktabelle, Datenbankattribut und andere. Die ColdFusion Komponente gibt die Ergebnisse ihrer Methoden dem ColdFusion Template zurück und dieses visualisiert die Daten mit Hilfe der graphischen Funktionen von ColdFusion. Das Template ruft nachher wieder die Datei `index.cfm` auf. Z.B. der Benutzer wählt das Datenbankattribut "Age of First Visit" (siehe Abb. 5.10). Das Template `index.cfm` ruft das Template `datentest.cfm`, das den Datentyp prüft, stellt fest, dass es sich um einen numerischen Datentyp handelt, und damit um eine quantitative Variable, und ruft das Template `histogramm.cfm`, das von seiner Seite die Komponente `histogramm.cfc`

ruft. Die Aufgabe der Komponente ist im Kapitel 5.1, Punkt 4 beschrieben. Die folgenden Diagramme zeigen den Zusammenhang zwischen den einzelnen ColdFusion Komponenten und Templates (siehe Abb. 5.1) und ein UML Aktivitätsdiagramm der Bibliothek (siehe Abb.5.2).

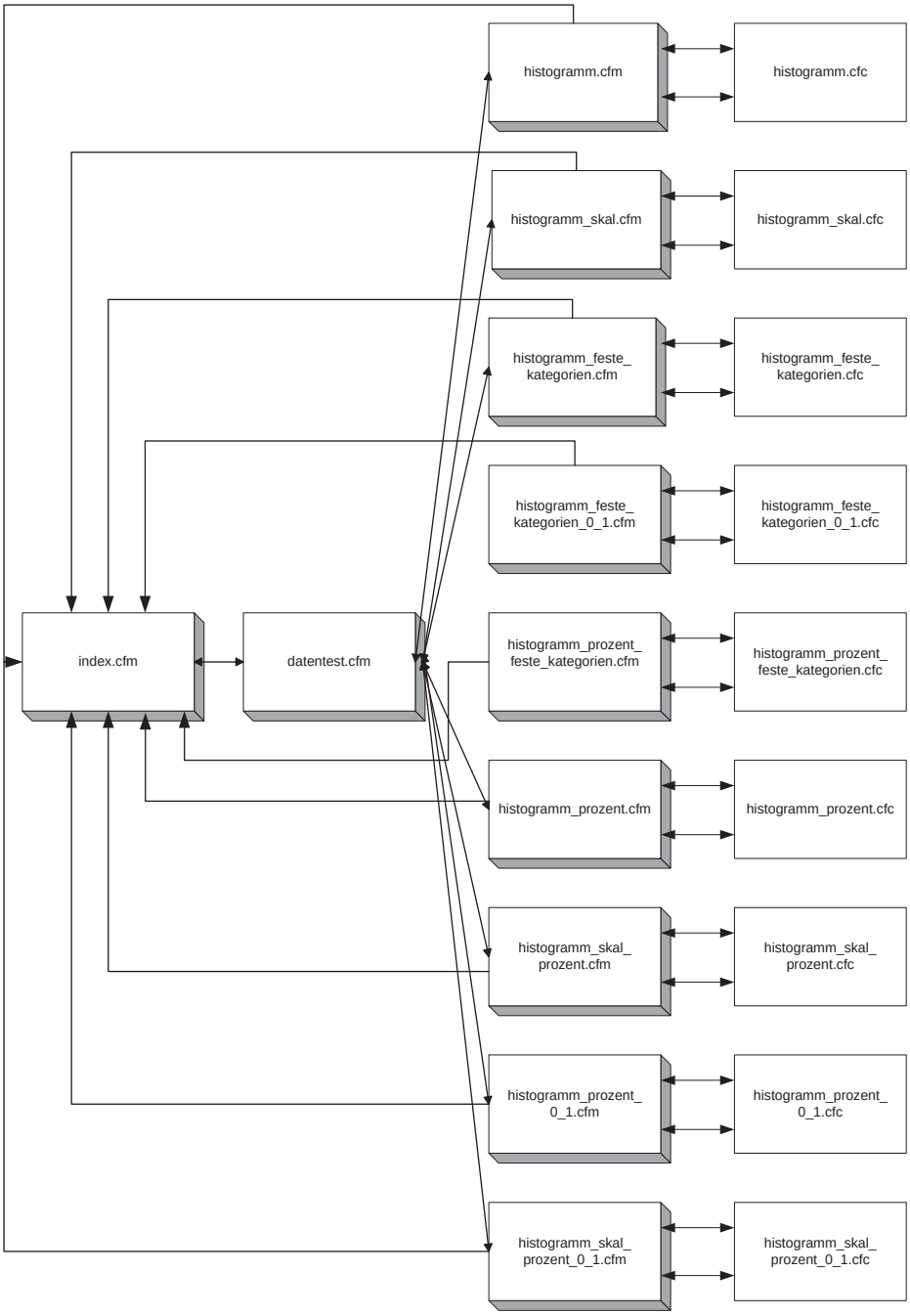

Abbildung 5.1: Zusammenhang zwischen den einzelnen Komponenten in der Bibliothek

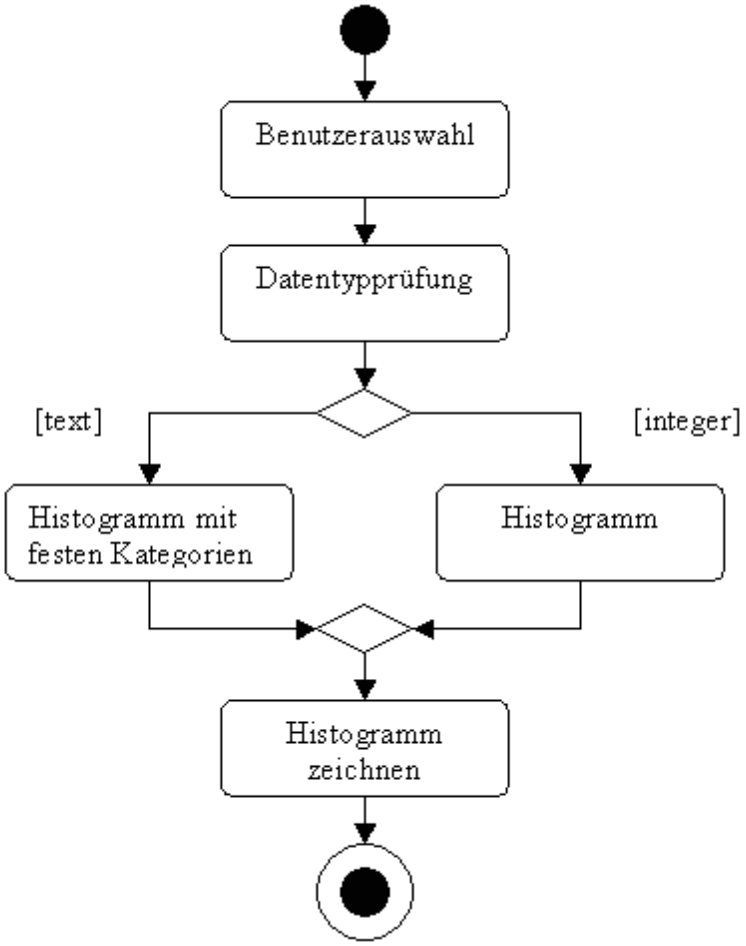

Abbildung 5.2: UML Aktivitätsdiagramm der Bibliothek

5.1 Funktionalitätsbeschreibung der einzelnen Komponenten der Bibliothek

Wenn die Bibliothek gefordert ist, bestimmte Daten zu visualisieren, werden die Daten zuerst einer Datentypprüfung unterzogen. Es wird zwischen textuellen, numerischen und booleschen Daten unterschieden. Von dem Datentyp wird weiter bestimmt, ob die Daten qualitativ oder quantitativ sind und anhand dieser Information werden die entsprechenden ColdFusion Templates und Komponenten aufgerufen.
Zur besseren Erläuterung der Funktionalitäten der Bibliothek folgt eine kurze Beschreibung der verschiedenen Komponenten (CFCs) und Templates

(CFML-Dateien). Es wird zwischen sechs verschiedenen Fällen unterschieden:

1. Darstellung qualitativer Daten (z.B. Geschlecht): Diese Daten werden von der ColdFusion Komponente `histogramm_feste_kategorien.cfc` bearbeitet. Die Komponente besteht aus einer Methode `freq`, die als Input den Datenbanknamen, die Datenbanktabelle, das Datenbankattribut, den Datentyp und einen Selektor (ob alle Datensätze oder nur ein bestimmter Teil von ihnen bearbeitet werden muß) bekommt. Die Komponente führt die Datenbankanfrage durch, gruppiert die Daten in Klassen (die Klassen sind von allen Datenausprägungen gebildet), berechnet die Summe der Datensätze für jede Klasse und übergibt die Ergebnisse ihrer Berechnungen dem Template `histogramm_feste_kategorien.cfm`, das die Daten mit Hilfe der ColdFusion Funktion `chchart` in Form eines Balkendiagramms darstellt (Abbildung 5.3).

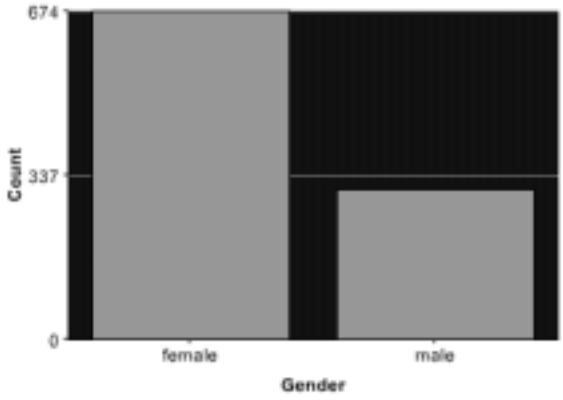

Abbildung 5.3: Qualitative Daten (Geschlecht)

2. Darstellung des Zusammenhangs zwischen qualitativen und qualitativen Daten (z.B. zwischen Verlauf der Krankheit und Geschlecht): wird von der ColdFusion Komponente `histogramm_prozent_feste_ketegorien.cfc` bearbeitet. Die Eingabeparameter sind wie in Punkt 1, zusätzlicher Parameter ist nur ein zweites Datenbankattribut. Die Komponente gruppiert die Daten des ersten Datenbankattributs in Klassen, für jede Klasse berechnet sie den prozentuellen Anteil der Daten des zweiten Datenbankattributs und übergibt die Ergebnisse dem Template `histogramm_prozent_feste_kategorien.cfm`, das die Daten graphisch als gestapeltes Balkendiagramm darstellt (Abbildung 5.4).

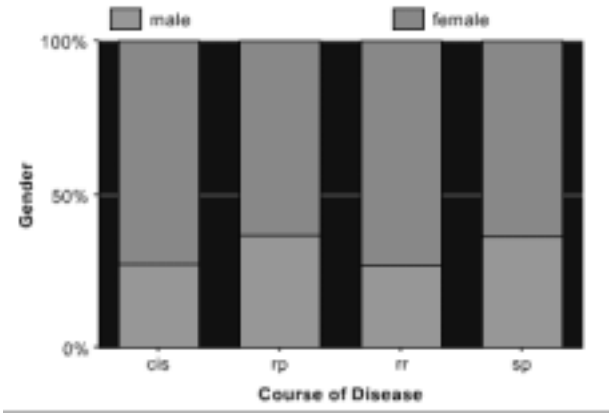

Abbildung 5.4: Zusammenhang zwischen qualitativen und quantitativen Daten (zwischen Verlauf der Krankheit und Geschlecht)

3. Darstellung des Zusammenhangs zwischen qualitativen und booleschen Daten (z.B. zwischen Geschlecht und Enhancement): wird von der Komponente `histogramm_feste_kategorien_0_1.cfc` bearbeitet. Die Komponente hat die selben Aufgaben wie die Komponente in Punkt 2, nur das zweite Datenbankattribut ist eine 0/1 Variable. Zusätzlich wird auch die Variabilität in den Messungen als Konfidenzintervall berechnet und dargestellt. Die Daten werden von dem Template `histogramm_feste_kategorien_0_1.cfm` als Balkendiagramm visualisiert (Abbildung 5.5).

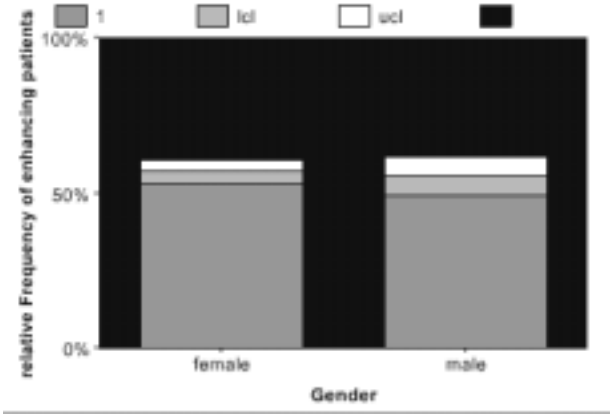

Abbildung 5.5: Zusammenhang zwischen qualitativen und booleschen Daten (zwischen Geschlecht und Enhancement)

4. Quantitative Daten (z.B. Alter): Bei den quantitativen Daten sind mehr Funktionalitäten möglich als bei den qualitativen.
   Die erste Funktionalität ist die benutzergesteuerte Klassenbildung. Der Benutzer gibt dabei die Anzahl der Balken für das Histogramm vor. Die Daten werden von der ColdFusion Komponente `histogramm.cfc` bearbeitet. Die Komponente besteht aus einer Methode `group`, die als Input den Datenbanknamen, die Datenbanktabelle, das Datenbankattribut, den Datentyp, einen Selektor und die gewünschte Anzahl der Klassen bekommt. Die Komponente bildet die Klassen, berechnet die Summe der Datensätze für jede Klasse und übergibt die Ergebnisse ihrer Berechnungen dem Template `histogramm.cfm`, das die Daten mit Hilfe der ColdFusion Funktion `cfchart` in Form eines Histogramms darstellt (Abbildung 5.6).

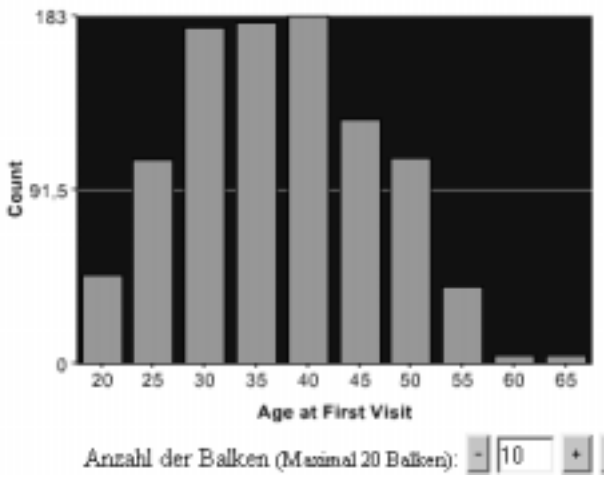

Abbildung 5.6: Quantitative Daten (Alter)

Andere mögliche Funktionalität ist die Skalierung der X-Achse. Das ist Aufgabe der Komponente `histogramm_skal.cfc`. Diese bekommt als Eingabe die selben Parameter wie die Komponente `histogramm.cfc`, zusätzliche Parameter sind die Skalierungsparameter "Von", "Bis" und "Schrittweite". Von den benutzerdefinierten Skalierungsparameter berechnet die Komponente die Anzahl der Klassen und ihre Größe. Die Darstellung ist Aufgabe des Templates `histogramm_skal.cfm`, die Daten sind als Histogramm dargestellt (Abbildung 5.7).

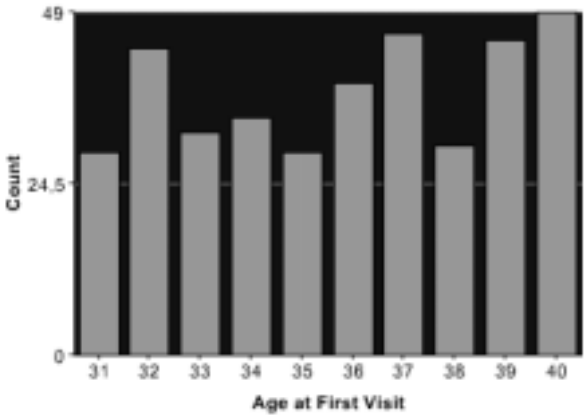

Abbildung 5.7: Skalierung der X-Achse bei quantitativen Daten

Die letzte Funktionalität ist die Drill-Down-Funktionalität. Per linken Mausklick auf einem Balken gelangt der Benutzer in das Histogramm der gewünschten Gruppe. Diese Aufgabe wird von der Komponente `histogramm_drill_down.cfc` erfüllt. Z.B. klickt man auf dem Histogramm von Abb. 5.6 auf dem Balken für die Altersgruppe 35, erhält man ein Histogramm ähnlich des Histogramms von Abb. 5.7.

5. Darstellung des Zusammenhangs zwischen quantitativen und qualitativen Daten (z.B. zwischen Alter und Geschlecht): Die Funktionalitäten sind dieselben wie in Punkt 4. Für jede Klasse auf der X-Achse sind die Daten für die Y-Achse als gestapelte Balken dargestellt. Die Drill-Down-Funktionalität wird nicht unterstützt, weil abhängend davon auf welchem gestapelten Balken geklickt wird, gelangt der Benutzer in verschiedene Histogramme für die selbe Klasse, was nicht gewünscht ist. Der Benutzer kann wieder die Balkenanzahl und die Skalierungsparameter für die X-Achse eingeben. Diese Aufgaben sind von den Komponenten `histogramm_prozent.cfc` und `histogramm_skal_prozent.cfc` erfüllt (Abbildung 5.8).

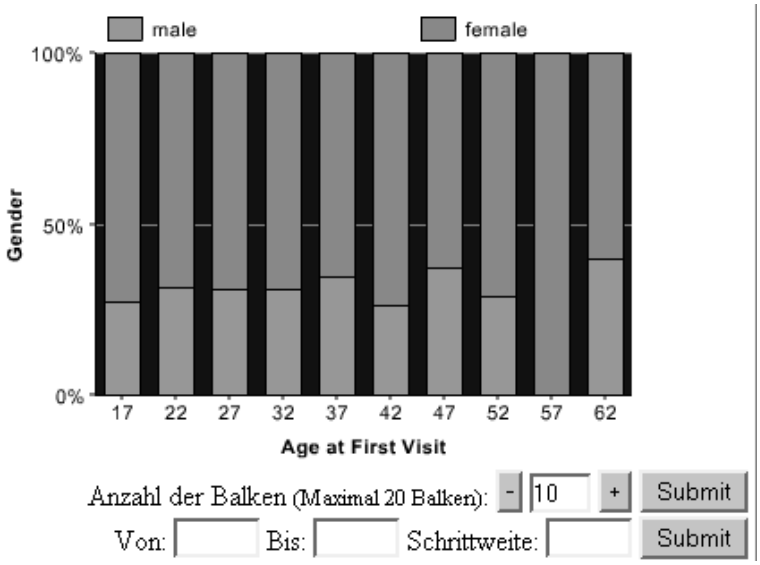

Abbildung 5.8: Zusammenhang zwischen quantitativen und qualitativen Daten (zwischen Alter und Geschlecht)

6. Darstellung des Zusammenhangs zwischen quantitativen und booleschen Daten (z.B. zwischen Alter und Enhancement): Wie in Punkt 5 und 3. Das zweite Datenbankattribut ist eine 0/1 Variable. Die Variabilität in den Messungen wird auch berechnet und dargestellt. Zuständige Komponenten sind `histogramm_prozent_0_1.cfc` und `histogramm_skal_prozent_0_1.cfc` (Abbildung 5.9).

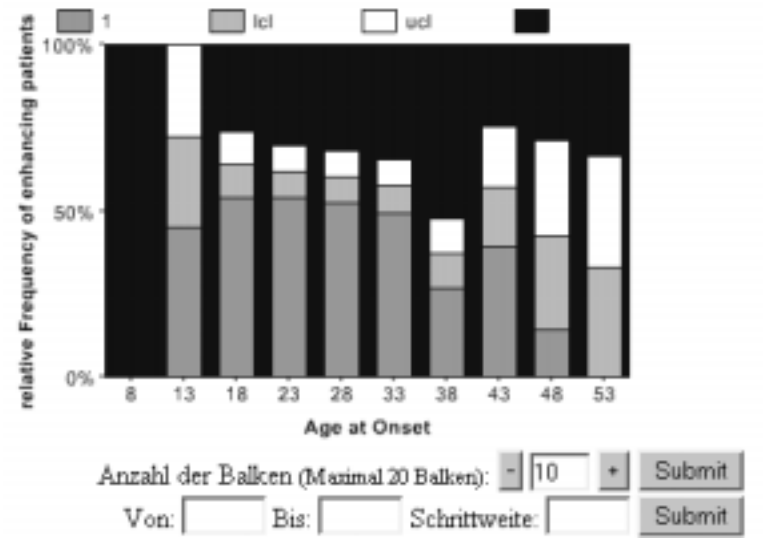

Abbildung 5.9: Zusammenhang zwischen quantitativen und booleschen Daten (zwischen Alter und Enhancement)

5.2 Funktionalitätsbeschreibung der Benutzerschnittstelle

Die graphische Benutzeroberfläche sollte möglichst leicht bedienbar sein. Der Benutzer bekommt auf einen Blick alle verfügbaren Attribute von der Datenbank und er kann auswählen, welche Daten visualisiert werden müssen. Er kann Variablen nur für die X-Achse auswählen, für die beiden Achsen und er kann auch einen Selektor auswählen, ob alle Beobachtungen, die ersten Beobachtungen für diese Variable oder die letzten Beobachtungen angezeigt werden sollen. Als Voreinstellung werden die ersten Beobachtungen dargestellt. Wenn der Benutzer seine Wahl getroffen hat und dann den Button "Histogramm anzeigen" anklickt, werden die Daten in Form eines Histogramms dargestellt (siehe Abb. 5.10).

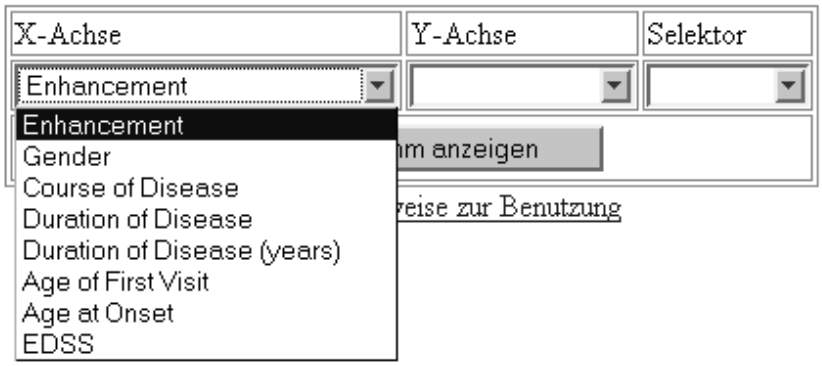

Abbildung 5.10: Die Benutzerschnittstelle

Bei quantitativen Daten für die X-Achse stehen mehrere Funktionalitäten zur Verfügung (siehe Abb. 5.11):

- Anzahl der Balken verändern: Der Benutzer kann die gewünschte Anzahl der Balken angeben und dann mit dem "Submit" Button bestätigen. Die maximal erlaubte Balkenanzahl ist vorgegeben.

- Mit den "+" und "-" Tasten kann der Benutzer die Balkenanzahl entweder erhöhen oder mindern.

- Von:... Bis:... Schrittweite:...: Es ist möglich, dass nur einen Teilbereich der X-Skala angezeigt wird. Der Benutzer muss den gewünschten Startwert, Endwert und die Schrittweite angeben und dann mit dem "Submit" Button bestätigen.

- Drill-Down-Funktionalität: Über die Drill-Down-Funktionalität - d.h. per linken Mausklick auf den zu betrachtenden Balken - gelangt man zum Histogramm der ausgewählten Gruppe.

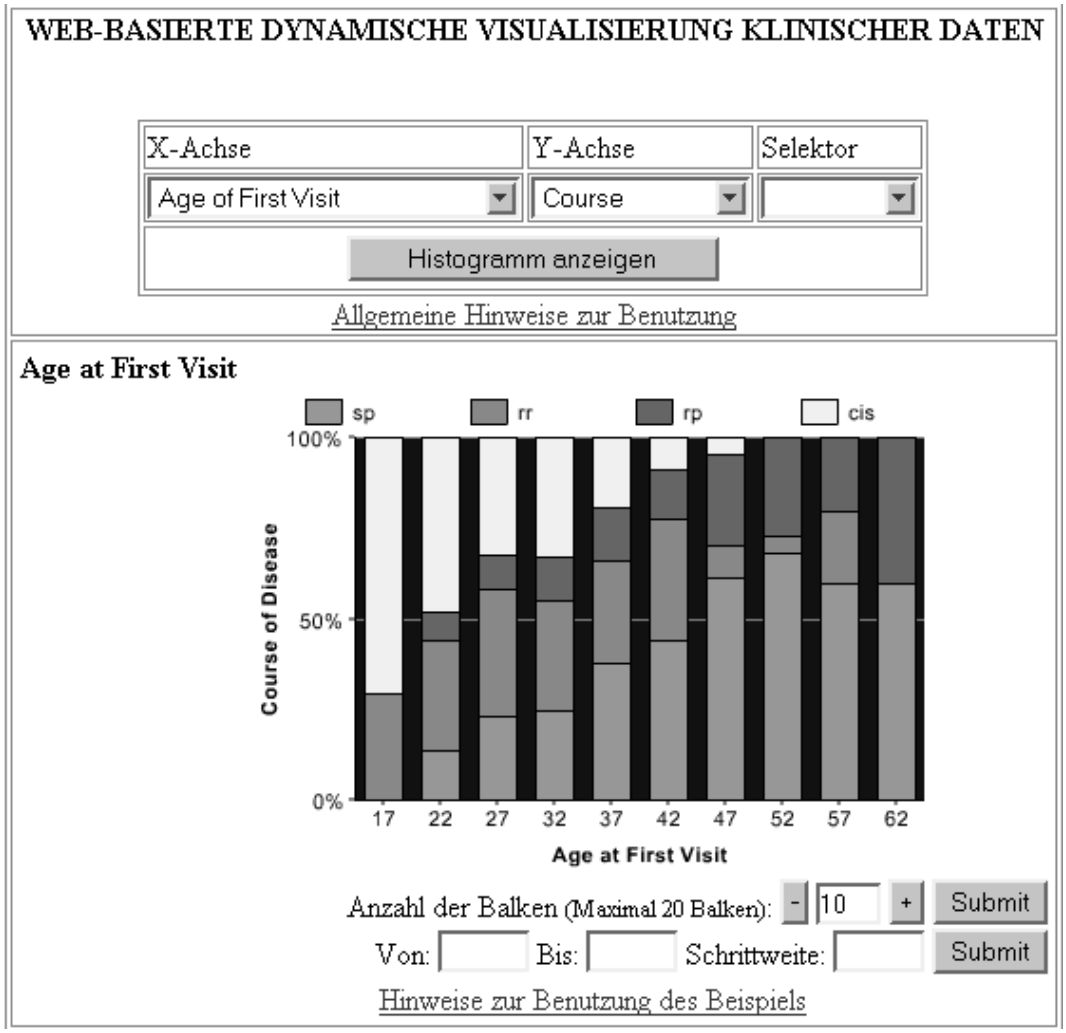

Abbildung 5.11: Beispiel für den Zusammenhang zwischen quantitativen und qualitativen Daten

Kapitel 6

Zusammenfassung und Ausblick

Mit diesem Praktikum wurde eine Bibliothek für Datenvorverarbeitung und Datenvisualisierung entwickelt. Die Komponenten der Bibliothek verbessern die eingebauten graphischen Funktionen von ColdFusion MX bei der Darstellung statistischer Daten, indem sie die Daten intelligent vorbereiten, analysieren und dann in einer statistisch adäquaten Form ausgeben. Hierbei wurde auch eine graphische Benutzeroberfläche entworfen, um eine schnelle Auswahl und Darstellung der Daten zu ermöglichen. Nach dem Abschluß des Fortgeschrittenen-Praktikums werden die Ergebnisse der Datenvisualisierung mit gängiger statistischer Software geprüft und validiert und dem Sylvia Lawry Centre zur Verfügung gestellt. Ziel des Fortgeschrittenen-Praktikums war auch die Vorbereitung der Daten und ihre Visualisierung in Form eines Scatter-Plots. Dabei sollten auch das Konfidenzintervall und die Regressionsgerade berechnet und dargestellt werden. Der Teil, der die Daten für die Visualisierung mit ColdFusion aufbereitet, das Konfidenzintervall und die Regressionsgerade berechnet, ist implementiert worden. Es hat sich aber herausgestellt, dass man mit den graphischen Funktionen von ColdFusion zwar einen Scatter-Plot darstellen kann, aber wenn das Konfidenzintervall und die Regressionsgerade dazu gezeichnet werden, wird das ganze Bild nicht sehr sauber. Besonders bei vielen Daten sind die Geraden zu unregelmäßig (siehe Abb. 6.1) und das Programm wird sehr langsam.

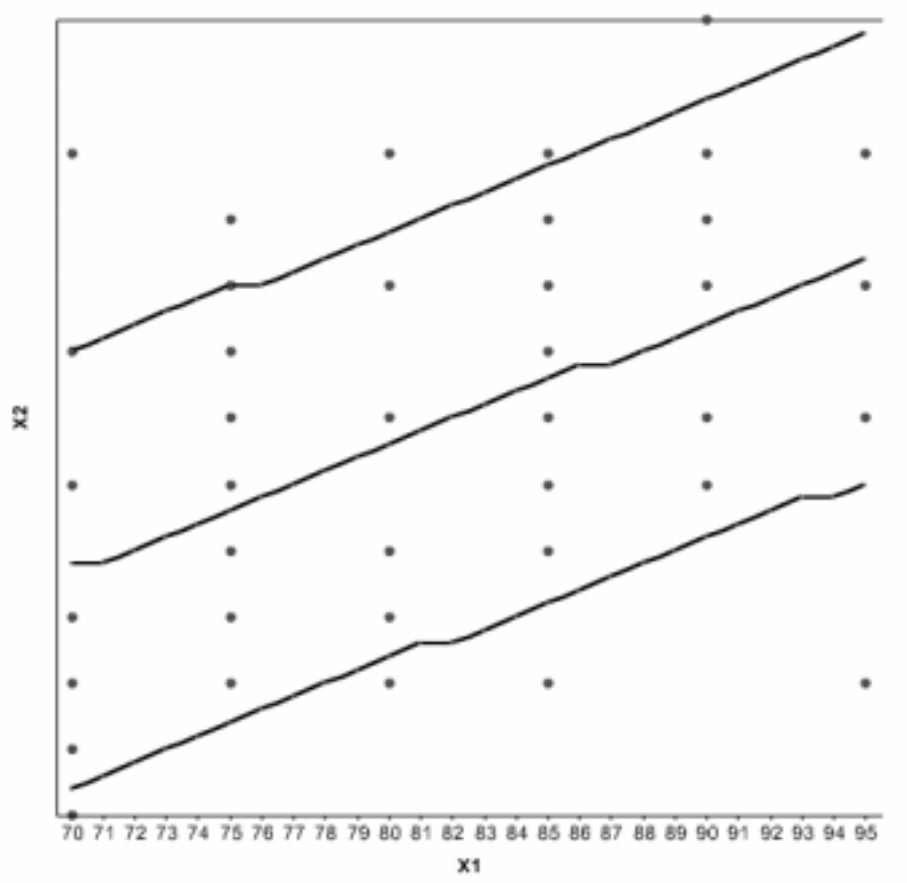

Abbildung 6.1: Scatter-Plot mit Konfidenzintervall und Regressionsgerade

Eine weitere gewünschte Erweiterung der Bibliothek wäre die Implementierung einer Komponente, die bei qualitativen Daten für jedes Merkmal ein getrenntes Bild zeichnet und diese Bilder auf einer Seite darstellt, damit die verschiedene Merkmale leicht vergleichbar sind.

Literaturverzeichnis

[FH99] Fassl, Horst. *Einführung in die medizinische Statistik.* Leipzig:Barth, 1999.

[WJ92] Werner, Jürgen. *Biomathematik und Medizinische Statistik.* Urban & Schwarzenberg, 1992.

[WC99] Weiß, Christel. *Basiswissen medizinische Statistik.* Springer, 1999.

[NEO] *NEO Architecture Overview.* http://www.macromedia.com/software/coldfusion/whitepapers/pdf/NeoArchWP.pdf

[DEVCF] *Developing Macromedia ColdFusion MX Applications with ColdFusion Markup Language (CFML).* http://download.macromedia.com/pub/coldfusion/documentation/cfmx_dev_cf_apps.pdf

[CFREF] *CFML Reference.* http://download.macromedia.com/pub/coldfusion/documentation/cfmx_cfml_reference.pdf