

Automatisierte Unterstützung für Prüfungen in Statistiksoftwarekursen im Humanmedizinstudium

Rainer Muche, Semra Kocak, Evelyn Jäckel,
Brigitte Janz, Beate Einsiedler

Universität Ulm
Institut für Biometrie
Schwabstraße 13
89075 Ulm
rainer.muche@uni-ulm.de

Zusammenfassung

Im „Lehrprojekt Biometrie“ an der Universität Ulm [6] wird jeweils im Sommersemester das Pflichtseminar im PC-Pool anhand der Statistiksoftware SAS-Analyst [4] gelehrt. Ein vergleichbarer Seminardurchlauf im Wintersemester wird wegen zu geringer PC-Pool-Kapazitäten im Seminarraum anhand von kleinen Aufgaben mit dem Taschenrechner durchgeführt. Dieser Kurs wird Semester begleitend an jedem der 6 Termine anhand von Kurztests abgeprüft. Diese Prüfungsform hat sich bewährt, da so eine kontinuierliche Mitarbeit der Studierenden über das gesamte Semester erreicht wird. Deshalb soll im PC-Kurs dieses auch angeboten werden.

Die Umsetzung dieser Vorgehensweise birgt zusätzliche Probleme. Unter anderem muss das Zeitproblem in der Übung gelöst werden sowie die Prüfungssituation formal eingehalten werden (Abschreib-Verhinderung). Außerdem ergibt sich ein Mehraufwand für die Korrektur, wenn alle Studierenden eigene Prüfungsdaten bekommen.

Um diesen Problemen zu begegnen, sollen folgende Punkte umgesetzt werden:

- Durch Zufallsziehung für jeden Studierenden einen eigenen Datensatz mit Nebenbedingungen bzgl. der Auswertbarkeit
- Zugriffsmöglichkeiten während der Prüfung
- Eingabemodul für Eingabe der Lösungen der Studierenden
- SAS-Auswertung der Musterlösung für jeden Studierenden
- Programm zum Vergleich der Musterlösung und Studierendeneingabe
- Eventuell (Halb-)automatisierte Punktvergabe

Das vorgesehene System ist mit mehreren Software-Tools entwickelt worden, wobei die wesentlichen Aspekte in SAS realisiert wurden.

Schlüsselworte: Prüfungen, Zufallsziehung, PROC IMPORT, ARRAY, IMPORT-Prozedur

1 Einleitung

An der Universität Ulm [6] wird jeweils im Sommersemester das Pflichtseminar Biometrie im PC-Pool anhand der Statistiksoftware SAS-Analyst [4] gelehrt. Ein vergleichbarer Seminardurchlauf im Wintersemester wird wegen zu geringer PC-Pool-Kapazitäten im Seminarraum anhand von kleinen Aufgaben mit dem Taschenrechner durchgeführt. Dieser Kurs im Winter wird Semester begleitend an jedem von insgesamt

6 Terminen anhand von Kurzttests abgeprüft, der PC-Kurs bisher an zwei Terminen nach jeweils 3 Seminarterminen. Die Prüfungsform am Ende jedes Termins hat sich über Jahre bewährt, da eine kontinuierliche Mitarbeit der Studierenden über das gesamte Semester erreicht wird. Deshalb soll im PC-Kurs dies auch angeboten werden. Der für Studierende und Dozenten zusätzliche Aufwand soll durch möglichst viel Automatisierung aufgefangen werden.

2 Ausgangssituation der Lehr- und Prüfungssituation

Die Einführung von Statistiksoftware-PC-Kursen im Pflichtseminar Biometrie im Humanmedizinstudium wurde an der Universität Ulm im so genannten „Lehrprojekt Biometrie“ über die letzten Jahre umgesetzt. Dabei sollten folgende Ziele erreicht werden:

Ziele der Nutzung von Statistiksoftware im Pflichtkurs Biometrie / Q1 :

- Anpassung des Kurses Biometrie an die neue ÄAppO (seit 2003)
- Steigenden Ansprüchen der Studierenden an Lehrmethoden entgegenkommen
- Die Einbindung moderner Medien
- Praxisnahe Umsetzung der vermittelten statistischen Kenntnisse
- Leistungs- und Motivationssteigerung durch Praxisbezug

Statistik Software SAS – Analyst

Bei der Auswahl der Statistiksoftware, die im Seminar eingesetzt werden sollte, sind viele Kriterien untersucht worden. Eine genaue Beschreibung der Auswahl findet sich in [3]. Ein wesentlicher Aspekt zur relativ einfachen und schnellen Einarbeitung der Studierenden war das Vorhandensein einer einfachen, benutzerfreundlichen, maus- und menügesteuerten Oberfläche. Die Wahl fiel auf die Oberfläche SAS-Analyst, die ohne zusätzliche Kosten mit SAS (bis einschließlich SAS Version 9.2) mitgeliefert wird. Ein weiterer wesentlicher Aspekt bei der Auswahl war das vorhandene Expertenwissen im Institut über diese Oberfläche [4].

Relevanter Datensatz

Zur Verfügung gestellt werden den Studierenden in den Übungen die Daten aus der Studie: *"Prevalence of Type 2 Diabetes mellitus and Impaired Glucose Regulation in Caucasian Children and Adolescents with Obesity living in Germany"* [7]. Die Idee bei der Auswahl dieser Studie als Grundlage der Lehre sind folgende Aspekte:

- Es handelt sich um reale Daten der Universitätskinderklinik Ulm.
- Die grundlegenden Krankheiten Adipositas und Diabetes mellitus sind Volkskrankheiten und sollten den Studierenden im 7. Semester bekannt sein.
- Das Patientenkollektiv besteht aus Kindern und Jugendlichen, dies könnte eher einen emotionalen Bezug zur Fragestellung erzeugen.
- Der für die Lehre reduzierte Datensatz ist mit 219 Beobachtungen und 80 Variablen so groß, dass der Einsatz von Statistiksoftware zur Auswertung evident ist.

3 Überlegungen zur Veränderung der Prüfungssituation

Im PC-Kurs Biometrie wurde zu Beginn entschieden, die Durchführung der Prüfungen anhand von Auswertungen am PC erfolgen zu lassen [6]. Dies ist im Sinne von OSCE-Prüfungen im Medizinstudium, in denen die Fähigkeiten der Studierenden (Skills) geprüft werden. Da aber noch nicht klar war, wie schnell die Studierenden mit dem System zurecht kommen, wurden zwei separate Prüfungstermine nach 3 bzw. 6 von 6 Übungsterminen angesetzt. Folgende Überlegungen führten zur Veränderung der Prüfungssituation im PC-Kurs Biometrie:

Überlegungen zur Überarbeitung der Prüfungssituation:

- Wir haben mit der Prüfungsdurchführung des Seminars im Wintersemester mit jeweiligem Kurztest am Ende eines jeden der 6 Seminartermine gute Erfahrungen gemacht. Die Studierenden haben eine Lernzielkontrolle, wissen wie sie bezüglich des Scheinerhalts stehen und können entsprechend die Zeit der Vor- und Nachbereitung individueller einteilen. Sie bereiten sich durch die Teilprüfungen aber vor allem auf jeden Seminartermin relativ gut vor, so dass der Lernerfolg über das Semester besser erreicht wird als mit einer Prüfung am Ende des Semesters. Das ist im Fach Biometrie besonders wichtig, da der Stoff jeder einzelnen Übung auf dem Inhalt der vorhergehenden aufbaut.
- Mit den zwei zusätzlichen Prüfungsterminen im PC-Kurs haben wir dagegen schlechte Erfahrungen gemacht. In den Sommersemestern 2006 und 2007 war dies einer der Hauptgründe, dass die Studierenden den fakultativ angebotenen PC-Kurs nicht gewählt haben [8]. Außerdem gab es aus der Durchführung des Lehrprojektes im SS 2008 mehrere Hinweise, dass die Studierenden sich erst zum ersten Prüfungstermin konkret vorbereitet und dadurch schon viel Übungszeit verloren haben.
- Deshalb ergab sich fast natürlich die Fragestellung, ob man nicht die Vorteile der Lehre am PC mit den Vorteilen der direkten Prüfung am Ende einer jeden Übung verbinden kann.

Um einigermaßen reliable und faire Prüfungen in einem PC-Pool durchführen zu können, sind allerdings einige Maßnahmen zu treffen. Die wichtigste dabei ist wohl, dass ein Abschreiben von Tischnachbarn verhindert wird und gleiche, faire Aufgaben für alle Studierende gestellt werden.

Daraus folgte für uns, dass für jeden Studierenden ein „eigener“ Datensatz zur Verfügung gestellt wird, der in der Prüfung am PC genutzt wird. Um die Prüfung nicht an einem komplett anderen inhaltlichen Projekt durchführen zu müssen, haben wir uns entschieden, die Prüfungsdatensätze durch Zufallsziehung aus dem Diabetes-Übungsdatensatz zu erzeugen. Diese Bereitstellung und die dadurch große Vielzahl an individuellen Lösungen mit größerem Aufwand für die Korrektur der Prüfungen ergab eine Notwendigkeit für eine möglichst weitgehende Automatisierung der Prüfung. In den folgenden Abschnitten wird die Umsetzung dieser Automatisierung beschrieben.

4 Programmelemente und Programmstruktur

Folgendes Organigramm zeigt den umgesetzten Ablauf der Prüfungen im PC-Kurs und deren Weiterverarbeitung bis zur Ergebnisdarstellung.

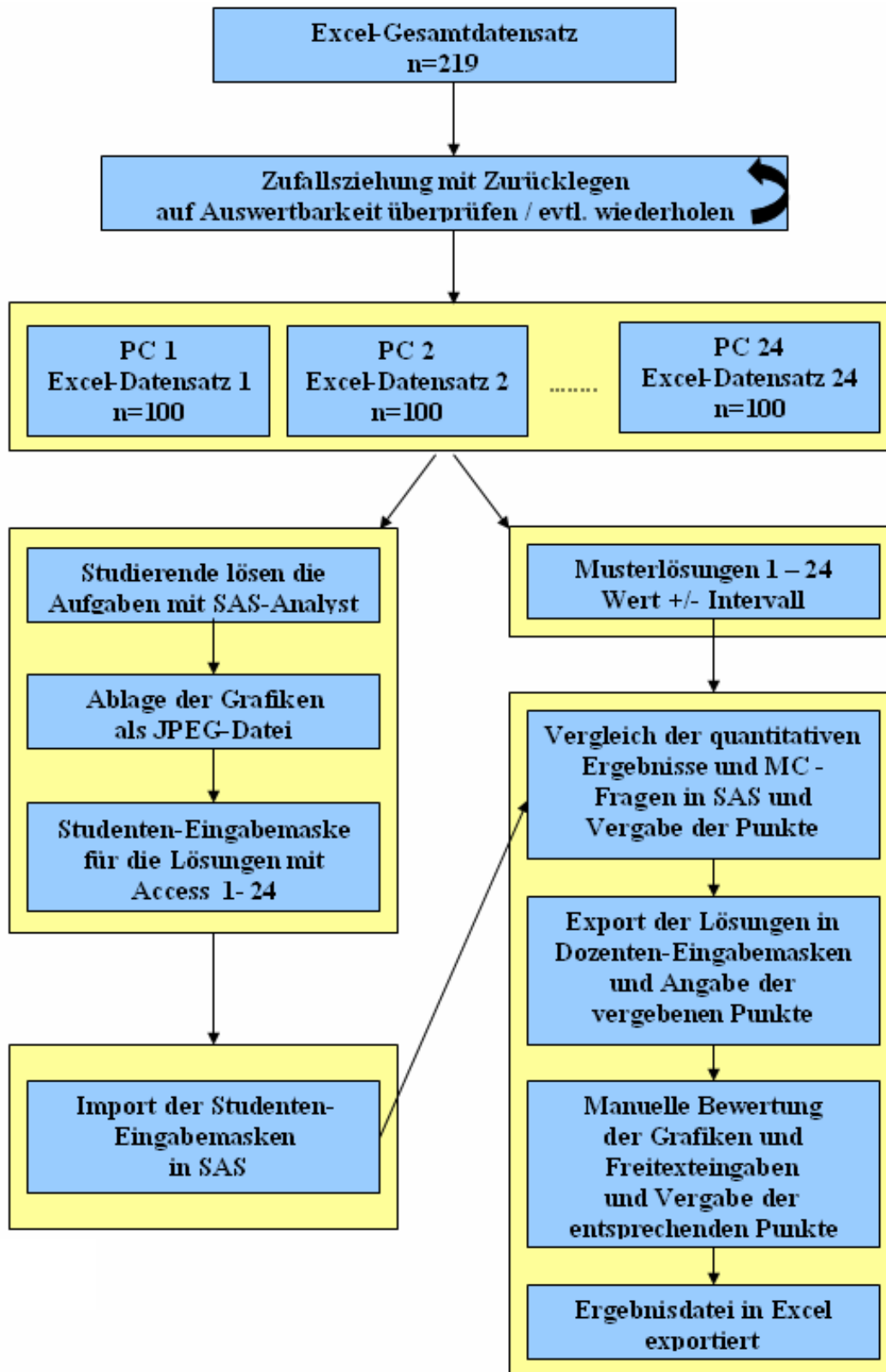


Abbildung 1: Ablauf der Unterstützung der Prüfung im PC-Kurs Biometrie

Folgende Software-Tools werden dabei eingesetzt:

- MS-Excel: Datenhaltung
 Dateneingabe und –ausgabe
- MS-Access: Eingabemasken für Ergebnisse und Punktvergabe
- SAS: 3 Makros / SAS-Programme:
1. Datenziehung (mit Zurücklegen) für jeden Studierenden
Kontrolle der Auswertbarkeit
zur Verfügungstellung von Masken
 2. Einlesen der einzelnen Datensätze und Erstellung Musterlösung
Einlesen der Ergebnisse der Studierenden
Punktvergabe für quantitative Lösungen und MC-Aufgaben
Auslesen für Punktvergabe für Freitexte und Grafiken Ergebnisse
 3. Auslesen aller Informationen für Ergebnislisting und Auswertung
der Prüfungsreliabilität [5]

5 Beschreibung der SAS-Programme

In diesem Abschnitt werden die Erzeugung der Prüfungsdaten mit Zufallsziehung und die Erzeugung der Musterlösung anhand umfangreicher SAS-Programme beschrieben. Eine genauere Beschreibung findet sich in [2].

5.1 Erzeugung der Prüfungsdaten

Die folgende Abbildung 2 beschreibt die Erzeugung der individuellen Prüfungsdaten. Dazu wird im ersten Schritt mit PROC IMPORT der Originaldatensatz eingelesen. Eine Makro-Schleife über 24 Durchgänge (PC-Pool mit 24 Rechnern) wird geöffnet. Im ersten Schritt der Schleife werden die Access-Eingabemasken für die Ergebnisse benannt und in einen Ordner für jeden Studierenden geschrieben. Somit steht für jeden Studierenden und jede der 6 Prüfungen eine leere Eingabemaske (Access-mdb) zur Verfügung.

Im nächsten Schritt werden durch Zufallsziehung mit Zurücklegen für jeden Studierenden die Prüfungsdaten mit je 100 Beobachtungen erzeugt [1]. Wir haben uns für das Zurücklegen entschieden, da so mehr unterschiedliche Datensätze erzeugt werden können. Mit dem Statement `set ziehung point = select nobs = total` wird angegeben, dass die bereits bestehende SAS-Datei "ziehung" in die nun entstandene SAS-Datei "gezogen" eingefügt wird. Allerdings wird mit der Option `point=` nur die Beobachtung selektiert, die dem Wert der Variable `select` entspricht. Mit der Option `nobs=` wird eine Konstante erstellt, in der die Anzahl der Datensätze in der SAS-Datei "ziehung" aufgelistet ist.

```
Data gezogen (drop=i);  
  select = ceil(ranuni(0)*total);  
  set ziehung point = select nobs = total;  
  i+1;  
  if i>100 then stop;  
run;
```

Da die Datensätze für die Auswertungen ähnliche Eigenschaften haben sollen, werden problematische Datensituationen ausgeschlossen (z.B. fast nur fehlende Werte in einer Variable oder äußerst schief verteilte Merkmale) und erst nach einer Überprüfung freigegeben und in den jeweiligen Prüfungsordner als MS-Excel-Datei gespeichert.

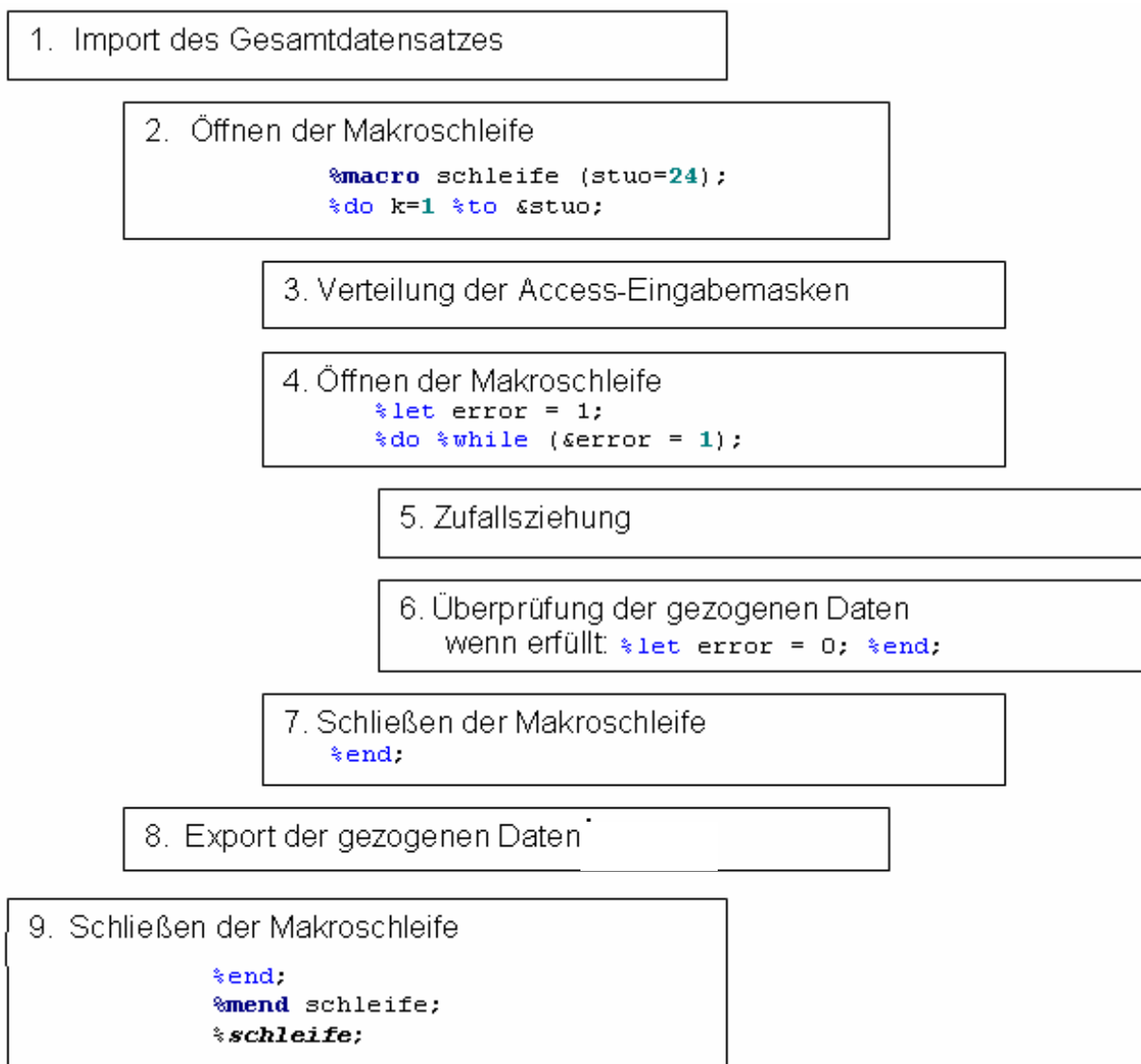


Abbildung 2: SAS-Makrostruktur für Erzeugung der Prüfungsdaten für 24 Studierende

5.2 Erzeugung der Musterlösungen:

Für die Erstellung der individuellen Musterlösungen werden die Tabelleninhalte der Eingabemasken (s. Kapitel 6) importiert und mit der in SAS erstellten Musterlösung verglichen. Um diesen Prozess zu vollziehen, wird an erster Stelle eine zweite Makroschleife geöffnet. Diese Makroschleife läuft ebenfalls von $&k = 1$ bis $&k = 24$ und importiert zunächst die einzelnen, zufällig erstellten Datensätze, die den Studierenden zum Erzielen der jeweiligen Ergebnisse behilflich waren. Mit dem nun im jeweiligen Durchlauf importierten Datensatz wird zunächst eine Musterlösung erstellt. Dies wird im Folgenden anhand der ersten Aufgabe des zweiten Testates erklärt.

Importieren:

Hier werden die Datensätze importiert und als SAS-Datei "testat2" abgespeichert (siehe Programmausschnitt). Der Importfile ist der in Kapitel 3.1.4 dargestellte Exportfile. Mit der zusätzlichen Funktion `Memosize=255` soll erreicht werden, dass alle Werte, die aus den Access-Feldern ausgelesen werden, die Maximallänge 255 haben dürfen. Da die Felder für die Interpretationen im Access-Feld als Memofeld deklariert und die Länge der Eingaben auf 255 Zeichen festgelegt wurde, muss diese Länge auch in SAS übernommen werden, damit die Texteingaben nicht abgeschnitten und komplett übernommen werden.

```
PROC IMPORT OUT= WORK.testat2
            DATATABLE= "tbl Kurztest2"
            DBMS=ACCESS REPLACE;
            DATABASE="S:\PC_&k.\Kurztest2_&k..mdb";
            SCANMEMO=YES;
            Memosize=255;
            USEDATE=NO;
            SCANTIME=YES;
RUN;
```

Die Programmsegmente, die zum Lösen der Aufgaben und zum Erzielen der Ergebnisse notwendig sind, werden nun ausgeführt und die einzelnen Ergebnisse in die entsprechende Makrovariable für die Musterlösungen der einzelnen Aufgaben übergeben. Um den Studierenden eventuelle Auf- und Abrundungen ihrer erzielten Ergebnisse zu gewährleisten, ohne dass diese vom System als falsch erkannt werden, wurden Intervalle der entsprechenden Ergebnisse vordefiniert. Die Musterergebnisse werden nun mit dem vordefinierten Wert addiert beziehungsweise subtrahiert und ebenfalls als Makrovariablen deklariert, um ein Lösungsintervall festzulegen. Im Folgenden kann verglichen werden, ob der erzielte Wert des Studierenden sich in diesem Intervall befindet. Die entsprechenden Punkte können dann vergeben werden.

Musterergebnisse und die entsprechenden Intervalle:

Zuerst werden mit Proc Means die Werte der erwünschten deskriptiven Auswertung erzielt. Da die Werte nicht direkt dem Output entnommen und weiterverarbeitet werden können, müssen zunächst mit Output Out= die Werte in eine separate SAS-Datei umgeleitet werden, damit mit dieser erstellten SAS-Datei weitergearbeitet werden kann.

Im folgenden Code wird angegeben, dass die Resultate der Proc Means Prozedur (im Genauen: Mittelwert, Standardabweichung, Minimum, 25%-Quartil, Median, 75%-Quartil, Maximum und Anzahl fehlender Werte der Variable Geburtsgewicht) in einer SAS-Datei „testat“ abgespeichert werden sollen.

```
proc means data=testat2 noprint; var GEBGWG;
output out=testat
mean=mean std=std min=min q1=q1 median=median q3=q3 max=max
nmiss=nmiss;
run;
```

Im nächsten Schritt wird eine weitere SAS-Datei „_testat2_“ geöffnet und die Ergebnisse der eben erstellten SAS-Daten „testat“ eingefügt. Da diese Werte in den restlichen Schritten öfters gebraucht werden, ist es sinnvoll, diese Werte mit der call symput Anweisung einer Makrovariablen zu übergeben. Bei dieser Anweisung wird in der Klammer angegeben, wie die Makrovariable in Hochkommata benannt werden soll. Mit der put Anweisung wird hingegen angegeben, welcher Wert dieser Makrovariable übergeben wird. Damit die Ergebnisse nachher überschaubar bleiben, wird mit der 8.2 Anweisung der übergebene Wert auf 2 Nachkommastellen und insgesamt 8 Stellen reduziert.

```
data testat2_;
set testat;
call symput ('mean',put(mean,8.2));
call symput ('std',put(std,8.2));
call symput ('min',put(min,8.2));
call symput ('q1',put(q1,8.2));
call symput ('median',put(median,8.2));
call symput ('q3',put(q3,8.2));
call symput ('max',put(max,8.2));
call symput ('missing',nmiss);
run;
```

Da im selben Data-Step nicht auf die neu erstellten Makrovariablen zugegriffen werden kann, wird eine weitere SAS-Datei „_1_Interval“ erstellt und die bereits bestehende SAS-Datei „_testat2_“ mit der SET-Anweisung eingefügt. Nun wird eine neue Variable generiert, welche die untere Grenze des Intervalls darstellen soll. Für die genaue Bestimmung wird der Wert der Makrovariable mit dem vordefinierten Wert subtrahiert und die neu erstellte Variable ebenfalls mit der CALL SYMPUT-Anweisung einer Makrovariablen übergeben. Die gleiche Prozedur wird für die obere Grenze des Intervalls durchgeführt, allerdings wird hier die Makrovariable mit dem vordefinierten Wert addiert. Im darauf folgenden Schritt werden die Intervalle zur Punktevergabe herangezogen. Für die deskriptive Statistik der ersten Aufgaben im zweiten Testat sind die Intervalle auf [Musterergebnis-20, Musterergebnis+20] festgelegt worden.

```
data _1_Interval; set _testat2_;
u_mean=((&mean)-20);
call symput ('u_mean',u_mean);
o_mean=((&mean)+20);
call symput ('o_mean',o_mean);
...
run;
```


Da das gesamte Programm sich in einer Makroschleife befindet und diese jeweiligen Werte für jeden einzelnen Durchlauf der 24 Durchläufe erstellt werden, werden im nächsten Punkt zwei SAS-Dateien generiert. In der ersten SAS-Datei „xe“ werden alle Makrovariablen der genauen quantitativen Musterergebnisse mit der dazugehörigen PC-Nummer aufgelistet. Mit der If-Schleife werden bei jedem Durchlauf die erzielten Ergebnisse mit der entsprechenden PC-Nummer in die SAS-Datei „erg“ eingetragen. Diese SAS-Datei „erg“ wird in den darauf folgenden Schritten mit den erzielten Ergebnissen der Studierenden zusammengefügt und als eine Excel-Tabelle der Ergebnisse dargestellt. Dies wird ermöglicht, indem die einzelnen Eingabemasken der Studierenden importiert und ebenfalls mit einer If-Schleife zu einer SAS-Datei „loesung“ mit der dazugehörigen PC-Nummer aufgelistet werden. Diese beiden SAS-Dateien werden nun mit der Prozedur SORT nach der PC-Nummer sortiert und in einer weiteren SAS-Datei `Data ende; merge erg loesung; by PC; run;` zusammengefügt. Als Resultat liegt die SAS-Datei "ende", wie in Abbildung 3 dargestellt, vor. Wenn durch Fehlen eines Studierenden bei einem Kurztest eine PC-Station nicht besetzt wird, werden die erstellten Musterergebnisse von SAS in dieser Tabelle dargestellt, jedoch die Lösungen des Studierenden als fehlende Werte gekennzeichnet.

```
Data xe;
PC = &k;
Aufg 1 mean = &mean;
Aufg_1_std_ = &std;
...
run;

%if &k=1 %then %do;
    data erg; set xe; run;
%end;
%if &k>1 %then %do;
    data erg; set erg xe; run;
%end;
```

	Musterlösung Aufgabe 1a	Musterlösung Aufgabe 1b	Ergebnis Studierender Aufgabe 1a	Ergebnis Studierender Aufgabe 1b
PC_1	Lösung SAS	Lösung SAS		Lösung Studierender	Lösung Studierender	
PC_2	Lösung SAS	Lösung SAS		Lösung Studierender	Lösung Studierender	
.						
PC_24	Lösung SAS	Lösung SAS		Lösung Studierender	Lösung Studierender	

Abbildung 3: Datensatz mit Musterlösungen und Studierendenlösungen

In der zweiten SAS-Datei „xeint“ werden nur die oberen und unteren Grenzen der Intervalle zusammengefügt. Diese SAS-Datei „xeint“ wird ebenfalls nach der PC-Num-

mer sortiert und mit der SAS-Datei „loesung“, welche die erzielten Ergebnisse der Studierenden beinhaltet, zusammengefügt. Schließlich liegt nun die SAS-Datei „punkte“ vor, welche im nächsten Schritt für die Punktevergabe benötigt wird.

```
Data xeint;
PC = &k;
Aufg 1 mean u = &u mean;
Aufg_1_mean_o = &o_mean;
...
run;
```

Punktevergabe:

Die Werte der SAS-Datei „punkte“ werden zunächst in Arrays übergeben, wobei loesu das untere Intervall und loeso das obere Intervall darstellen. In den Klammern muss die genaue Anzahl der Arrays beziehungsweise der Variablen vordefiniert werden. Die Variablen der Ergebnisse der Studierenden werden in den Arrays erg1 und die jeweiligen Punkte in punktel1 abgespeichert. Die Ergebnisse bei denen keine Rundungsfehler akzeptiert werden, werden in einem zweiten Array erg2 und die dazu complimentierenden Musterlösungen in der Array-Kette exakt abgespeichert. Die hierzu gehörenden Punkte werden in punkte2 übergeben.

```
data punkte;
array loesu (7) Aufg 1 mean u Aufg 1 std u Aufg_1_q1_u
Aufg 1 median u Aufg_1_q3_u Aufg_3_Perc0_u
Aufg 3 Perc1 u;
array loeso (7) Aufg 1 mean o Aufg 1 std o Aufg_1_q1_o
Aufg 1 median o Aufg_1_q3_o Aufg_3_Perc0_o
Aufg 3 Perc1 o;
array erg1 (7) Aufg 1 mean Aufg 1 std Aufg_1_q1
Aufg 1 median Aufg_1_q3 Aufg_3_Perc0
Aufg 3 Perc1;
array punktel1 (7)p1 p2 p4 p5 p6 p9 p10;
array exakt (13) Aufg_1_min_ Aufg_1_max_ Aufg_1_mis_Aufg_3_Freq0_
Aufg_3 Freq1
Aufg 3 mis Aufg 4 a Aufg 4 b Aufg 4_c_ Aufg_4_d_
Aufg 4 e Aufg 4 f Aufg 4 abs 2 3 ;
array erg2 (13) Aufg 1 min Aufg 1 max K2 lmiss
Aufg 3 Freq0 Aufg 3 Freq1 K2 3miss
K2 4gp11 K2 4gp21 K2 4gp12 K2_4gp22
K2 4gp13 K2_4gp23 K2_4;
array punkte2 (13) p3 p7 p8
p11 p12 p13
p14 p15 p16 p17
p18 p19 p20;

run;
```

1. Do-Schleife (Intervalle):

Die Abfrage soll von Array 1 – Array 7 laufen und richtet sich danach, ob die Ergebnisse größer gleich den unteren Intervallwerten und kleiner gleich den oberen Intervallwerten sind. Dabei dürfen die Ergebnisse keine fehlenden Werte darstellen. Wenn dies zutrifft, werden die einzelnen Punkte auf 1, andernfalls auf 0 gesetzt.

```
do i=1 to 7;
  if erg1(i) >= loesu(i) and
  erg1(i) <= loeso(i) and erg1(i) ne . then do;
  punkte1(i) = 1; end;
  else do; punkte1(i) = 0; end;
end;
```

2. Do-Schleife (Exakter Wert):

Die Abfrage soll von Array 1 – Array 13 laufen und überprüft, ob die Ergebnisse der Studierenden den Lösungs-Ergebnissen entsprechen, wobei die Ergebnisse ebenfalls keine fehlenden Werte darstellen dürfen. Wenn dies zutrifft, werden die einzelnen Punkte auf 1, andernfalls auf 0 gesetzt.

```
do l=1 to 13;
  if erg2(l) = exakt(l) and erg2(l) ne . then do;
  punkte2(l) = 1; end;
  else do; punkte2(l) = 0; end;
end;
```

Letztlich werden die zusammengehörigen Punkte pro Aufgabe aufsummiert (siehe unten) und in eine entsprechenden Variable übergeben. In diesem Falle wird jede Angabe innerhalb einer Aufgabe tatsächlich mit einem Punkt bewertet. In den anderen Testaten kann es jedoch auch vorkommen, dass eine Angabe mit zwei oder mehr Punkten bewertet werden kann. In diesem Fall wird der entsprechende Punkt mit zwei oder mehr multipliziert, bevor es der Variable übergeben wird (z.B. punkte = (p1*2)+(p2*2)). Dies bedeutet, wenn null Punkte erzielt wurden, bleibt auch nach der Multiplikation der Wert auf null.

```
punkt_Aufg_1 = p1 + p2 + p3 + p4 + p5 + p6 + p7 + p8;
punkt_Aufg_3 = p9 + p10 + p11 + p12;
punkt_Aufg_3_mis = p13;
punkt_Aufg_4 = p14 + p15 + p16 + p17 + p18 + p19 + p20;
```

6 Eingabemasken für Studierende und Dozenten

Eingabemaske der Studierenden:

Die Studierenden sollen ihre quantitativen Ergebnisse und MC-Lösungen in eine Access-Eingabemaske eingeben (siehe Abbildung 4). Dies kann entweder manuell oder per Copy-Paste erfolgen. Hier ergab sich jedoch das Problem, dass die Darstellung der Dezimalzahlen einerseits mit Punkt und andererseits mit Komma repräsentiert werden können. Dieses Problem wurde damit gelöst, dass beim Auslesen der einzelnen Access-Felder in SAS alle Werte einheitlich mit einem Punkt dargestellt werden. In dem folgenden Code wird verdeutlicht, wie diese Umwandlung in unserem Programm durchgeführt wird. Das eingelesene Feld der Access Eingabemaske stellt hier die Variable AccessFeld dar. Der Wert wird bis zum Erscheinen eines Punktes oder eines Kommas ausgelesen und wird einer Variable Namens var1a übergeben. Alle Werte, die nach dem Komma oder dem Punkt folgen, werden einer weiteren Variable var1b übergeben. Im

nächsten Schritt wird eine neue Variable erstellt und die Werte der Variablen var1a und var1b zusammengefügt sowie mit einem Punkt getrennt.

```
var1a=scan (AccessFeld,1,"; ."); /*teil vor dem komma in var1a;*/  
var1b=scan (AccessFeld,2,"; ."); /*nachkommastellen in var1b;*/  
NeueVariable = compress (var1a||'.'||var1b);
```

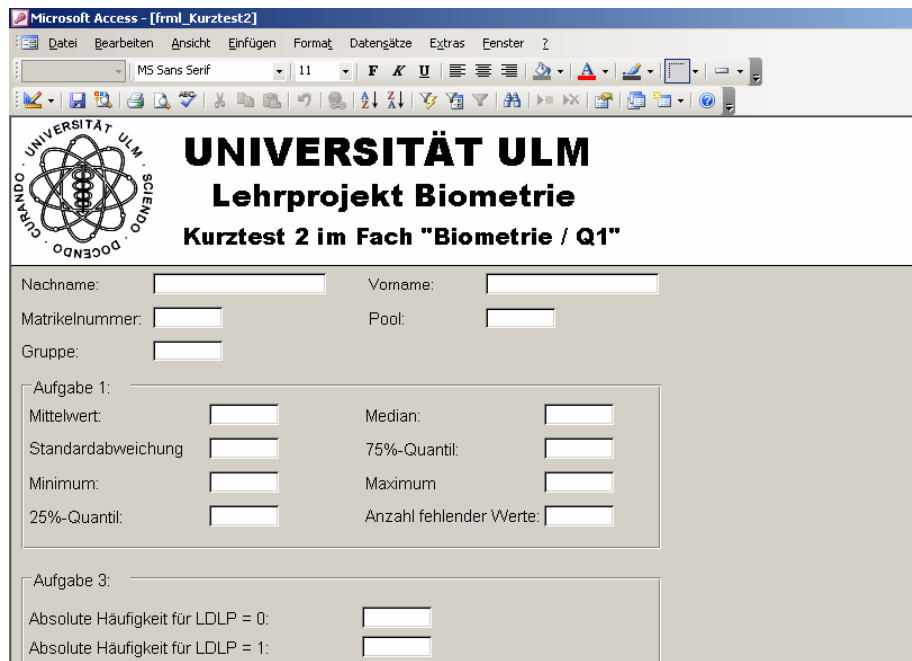


Abbildung 4: Eingabemaske für Ergebnisse für Studierende

Die Aufgabe 2 ist in dieser Eingabemaske nicht vorhanden, da es sich hier um eine Aufgabe zur Erzeugung einer Grafik handelt. Die Studierenden bekommen in ihren Aufgabenstellungen genau erläutert, wie sie bei solchen Aufgaben vorgehen sollen. In diesem Fall wird die entstandene Grafik als JPEG oder GIF Datei in ihren individuellen Verzeichnissen abgespeichert.

Nachdem die Studierenden die Felder in ihren Eingabemasken ausgefüllt haben, werden nach Abspeichern diese Werte der dahinter liegenden Tabelle übergeben. Falls die Studierenden das Fenster mit der Access-Eingabemaske ohne Speichern verlassen möchten, wird vom System ein Dialogfenster geöffnet mit der Nachfrage, ob die Daten abgespeichert werden sollen. In diesem Dialogfenster sollte dann auf "ja" gedrückt werden, andernfalls werden alle Daten verworfen. Nun ist nach einem Kurzttest für jeden Studierenden solch eine individuelle Tabelle erstellt worden, welche in SAS importiert werden kann.

Nach Beendigung eines Kurztestes werden alle aktuellen Daten aus dem gemeinsamen Verzeichnis „S“ auf einen weiteren Datenträger (USB-Stick) überschrieben. Hier ist es wichtig, dass die Datenträger mit den leeren Eingabemasken und der Datenträger mit den nun gefüllten Eingabemasken unterschiedliche Datenträger darstellen, da das spätere Auslesen sonst durch doppelt auftretende Verzeichnisnamen zu Fehlermeldungen führen könnte. Ein weiterer wichtiger Punkt ist die nachträgliche Leerung des gemeinsamen Verzeichnisses „S“ nach jedem Kurzttest.

Dozentenmaske

Die eingegebenen Werte der Studierenden werden nun in eine Dozenten-Access-Eingabemaske übertragen und die erworbenen Punkte jeweils in die entsprechenden neuen Felder übertragen (siehe Markierungen in der Abbildung 5). Die Punktfelder für die Texteingaben und Grafiken werden hierbei auf 0 gesetzt, um die manuelle Bewertung durchzuführen, die im nächsten Abschnitt erläutert wird. Ziel dieser Dozenten-Eingabemasken ist, zum einen die Kontrolle über die Punktevergabe und zum anderen die Möglichkeit zusätzlich manuelle Punkte zu vergeben, wie im Falle der Interpretationen und Grafiken.

Abbildung 5: Eingabemaske für Ergebnisse für Dozenten

Beurteilung der Grafiken und Freitexte

Nach Erstellung der Grafiken kann mit rechtem Mausklick diese Graphik als JPEG oder GIF-Datei in einem angegebenen Verzeichnis abgespeichert werden. Je nach Grafik ist meist nur eines möglich, entweder JPEG oder GIF. In diesem Fall sollten die Studierenden die genauen Angaben der Aufgabenstellungen entnehmen und verfolgen. Die Grafiken sollten in ihrem entsprechenden Verzeichnis mit genau angegebenen Namen abgespeichert werden. Die durch die Musterlösung entstandenen Grafiken werden ebenfalls mit dem Code `filename grafout "S:\PC_&k.\Aufgabe2Loesung.gif"` in die jeweiligen Verzeichnisse (`Aufgabe2_Lösung.gif`) geleitet. Die Musterlösungen stellen im Allgemeinen nur GIF-Dateien her, da der Unterschied beider Grafik-Formate sehr gering ist. Damit könnten nun beide Grafiken von den Dozenten verglichen und entsprechend bewertet werden. Die vergebenen Punkte werden in der Dozenteneingabemaske im entsprechenden Feld für die Punkte dieser Grafik eingetragen.

7 Ausgabe der Ergebnisse

Als letzter Schritt werden alle Ergebnisse (Punkte, Lösungen und Musterlösungen) in einer SAS-Datei zusammengefasst und für die weitere Bearbeitung zum Listing der Punkte und Noten für die Studierenden und weitere Auswertungen als MS-Excel-Datei ausgelesen.

Die Prüfungen sollen natürlich fair, objektiv, reliabel und valide sein, um die einzelnen Studierenden gerecht zu bewerten. Zur Untersuchung dieser Eigenschaften können verschiedene statistische Auswertungen herangezogen werden. Es gibt dazu entsprechende Vorschläge, die diese Auswertungen festlegen sowie ein im Institut erstelltes Auswertungsprogramm [5]. Für dieses Programm sind die Ergebnisse in standardisierter Form zu übergeben.

8 Testläufe

Für den Testlauf wurden für alle Testate die jeweiligen Zufallsziehungen und die gleichzeitige Erstellung der Eingabemasken ausgeführt. Diese Daten wurden auf einen Datenträger (USB-Stick) geladen und über den Dozenten-PC im PC-Raum auf das gemeinsame Verzeichnis „S“ kopiert. Des Weiteren wurden über verschiedene PCs die Datensätze mit SAS-Analyst aufgerufen und die erzielten Lösungen in die Eingabemasken eingetragen. Bei einigen Testdaten wurden beabsichtigt Fehler aufgeführt, um die darauf folgenden Auswirkungen zu beurteilen. Beim Durchlauf dieser Schritte wurden die in der Abbildung 6 aufgeführten Punkte getestet. Die Folgen der beabsichtigten Fehler sind in der Abbildung 7 aufgelistet.

Anforderung	Ja	Nein
Zugriff der PCs auf das gemeinsame Verzeichnis „S“	✓	
Aufrufen der entsprechenden Datensätze mit SAS-Analyst	✓	
Öffnen der Access-Eingabemasken	✓	
Füllen der einzelnen Felder in den Eingabemasken	✓	
Abspeichern der Lösungen	✓	
Musterlösung der gezogenen Datensätze	✓	
Import der Eingabemasken	✓	
Texteingaben in Integer umwandeln	✓	
Vergleich Ergebnis und Musterlösung	✓	
Vergabe der Punkte	✓	
Abspeichern der Grafiken	✓	
Export in die Dozenten-Eingabemasken	✓	

Abbildung 6: Testläufe der Anforderungen an das System

Mögliche Fehler	Vom System abgedeckt	Vom System nicht abgedeckt
Matrikelnummer vergessen einzutragen	✓	
Falsche PC-Nummer eingetragen		✓
Vor den Ergebnissen Leerzeichen eingetippt	✓	
Dezimalzahlen zum einen mit . zum anderen mit , eingetragen	✓	

Abbildung 7: Testläufe der möglicherweise auftretenden Fehler

Da in den Access-Eingabemasken die Matrikelnummer als Primärschlüssel verwendet wird, wird der Studierende beim Versuch zu speichern darauf aufmerksam gemacht, dass dies ohne die Matrikelnummer nicht möglich ist. Somit wurde erreicht, dass jede Eingabemaske eindeutig einem Studierenden zugeordnet werden kann. Wenn beim Eintragen der Lösungen in den Eingabemasken wenige Leerzeichen entstanden sein sollten, werden diese beim Einlesen der Felder ignoriert. Die Ergebnisse können nun in den Feldern entweder mit der Hand eingetippt und hierbei die Dezimalzahl mit einem „.“ gekennzeichnet oder aus dem Output kopiert und eingefügt werden. Jedoch wird hier die Dezimalzahl mit einem „.“ gekennzeichnet, da SAS-Analyst mit einem amerikanischen Codepage arbeitet. Bei der Umwandlung der Werte in Integer werden diese jedoch vereinheitlicht.

Falls ein Studierender eine falsche PC-Nummer eingibt und diese Nummer gleichzeitig von einem anderen Studierenden ebenfalls eingegeben wurde, wird beim Exportieren der Ergebnisdateien in die Dozenten-Eingabemasken die erste Version überschrieben. Da bei der Export-Prozedur der Dozenten-Access-Eingabemasken die REPLACE-Option verwendet wurde, wird die erste Datei mit der doppelt aufgeführten PC-Nummer erstellt, jedoch beim nächsten Durchlauf durch die Datei mit der identischen PC-Nummer überschrieben. Dies bedeutet, dass beim Auswerten der Dozenten-Eingabemasken eventuell der Verlust einer Lösung nicht auffallen würde.

Jedoch wird am Ende jeder Musterlösung eine Tabelle erstellt, in dem pro Aufgabe und PC-Nummer (PC1 - PC24) alle Musterlösungen und alle vom Studierenden erzielten Ergebnisse als Excel-Datei dargestellt werden. In dieser Tabelle werden alle Formen aufgeführt, die der Prüfende genau analysieren sollte. Bei eventuell doppelt aufgeführten PC-Nummern besteht für den Prüfenden die Möglichkeit der Zurückverfolgung und ggf. entweder manuellen Bewertung oder bei vorhandener Sitzordnung die nachträgliche Abänderung dieser Nummer. Sollte es zu nachträglichen Abänderung der PC-Nummer kommen, kann das ganze Programm ein weiteres Mal gestartet werden.

Diskussion und Ausblick

Mit den vorliegenden Programmen stehen Tools zur Verfügung, im Statistiksoftwarekurs Biometrie an der Universität Ulm Prüfungen Semester begleitend als Kurztests am Ende einer jeden Übung durchzuführen. Die Umsetzung der Vorgaben und die Programme sind vor Beginn des Sommersemesters 2009 gerade fertig geworden. Entsprechende Tests zeigen, dass der Ablauf wie geplant funktionieren kann. In wie weit die Studierenden mit dieser Prüfungsform zu Recht kommen und ob sich die Abläufe in Bezug auf die zeitliche Durchführung, Akzeptanz durch die Studierenden und bei der Korrektur der Prüfungen bewähren, wird im folgenden Sommersemester evaluiert.

Literatur

- [1] Boudreaux D., Cranford K. (1995): Simple Random Sampling and Subsetting Strategies Using SAS Software. Observations 3rd quarter 1995
- [2] Kocak S. (2009): Programmierung eines Prüfungstools mit SAS im PC-Kurs Biometrie. BSc-Arbeit Hochschule Ulm, Med. Dokumentation und Informatik
- [3] Muche R., Babik T. (2008): Auswahl und Einbindung einer Statistiksoftware im „Lehrprojekt Biometrie“ an der Universität Ulm, GMS Medizinische Informatik, Biometrie und Epidemiologie, Vol. 4(1), Verfügbar unter: <http://www.egms.de/pdf/journals/mibe/2008-4/mibe000061.pdf>
- [4] Muche R., Habel A., Rohlmann F. (2000): Medizinische Statistik mit SAS-Analyst. Springer Verlag, Heidelberg
- [5] Muche R., Janz B., Einsiedler B. (2009): Quantitative Analysen medizinischer Prüfungen mittels eines (teil-) automatisierten SAS-Programms. KSFE-Proceedings der 13. KSFE-Tagung, Halle/Saale. Shaker Verlag
- [6] Muche R., Weirather-Herrlein S., Wildt M., Radlinger K., Seefried K. Jordan J. (2005): Dokumentation des Lehrprojektes „Biometrie“ an der Universität Ulm. Shaker Verlag, Aachen
- [7] Wabitsch M., Hauner H., Hertrampf M., Muche R., Hay B., Mayer H., Debatin K.M., Heinze E. (2004): Prevalence of Type 2 Diabetes mellitus and Impaired Glucose Regulation in Caucasian Children and Adolescents with Obesity living in Germany. Int. J. Obesity 28, 307-313
- [8] Wegerer J. (2008): Evaluation des Lehrprojektes Biometrie an der Universität Ulm zum Sommersemester 2008, Studienarbeit des Universitätsklinikum Ulm, Schule für Medizinische Dokumentation