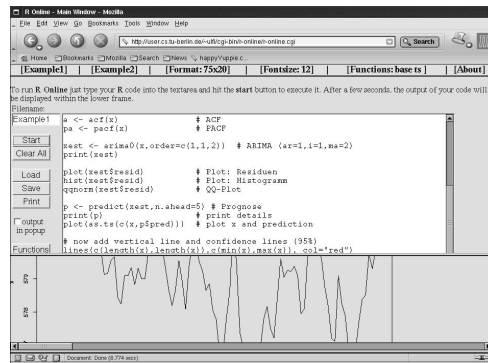


R-Online

Ulf Bartel, Karl Bartel
Technische Universität Berlin

15. Oktober 2002



Zusammenfassung

R-Online ist eine einfache Programmierumgebung fuer die statistische Programmiersprache R[3]. R-Online soll einen einfachen Einstieg in das Programmieren mit R bieten. Da sich das Programm komplett ueber einen Browser bedienen laesst, ist es fuer den Benutzer nicht notwendig, irgendwelche Software lokal bei sich zu installieren. Hinzu kommt, dass R unter der GNU GPL als freie Software erhaeltlich ist.

1 Projektziel

Ziel des R-Online Projektes ist es, einen einfachen Einstieg in die Programmierung mit der statistischen Programmiersprache R zu bieten. Das Programm sollte ohne Einarbeitung

oder komplizierte Installation auf jedem Rechner lauffaehig sein. Um das Lernen der Sprache R zu erleichtern, muessen ausserdem eine Dokumentation und leicht verstaendliche Beispiele in R-Online integriert werden.

Das Projekt richtet sich ausschliesslich an Einsteiger. Fuer groessere Projekte wird R-Online weder gedacht noch geeignet sein.

2 Verwendete Techniken

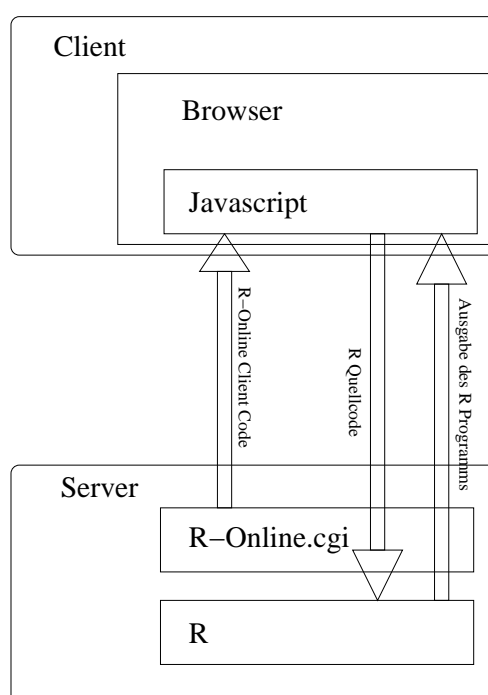


Abbildung 1: prinzipielle Funktionsweise

Client: Damit keine lokale Installation notwendig wird, mussten wir fuer den Client entweder Java oder Javascript verwenden. Java ist deutlich maechtiger als Javascript, also koennten mehr nuetzliche Funktionen, wie z.B. Syntaxeinfarbung, in R-Online verwendet werden. Javascript hingegen stellt weniger Anforderungen an den Client-Rechner und ist in nahezu allen bekannten Browsern standardmaessig integriert. Wir entschieden uns fuer Javascript, um moeglichst vielen Leuten den Zugang zu R-Online zu ermoeglichen.

Server: Ausser der R Installation wird auf dem Server ein Skript benoetigt, das den Javascript-Code und die HTML-Seiten fuer den Client liefert. Zur Erzeugung dieser dy-

namischen Seiten waere eigentlich jede gaengige Programmiersprache geeignet. Besonders praktisch erschienen uns Skriptsprachen (wie z.B. Php, Pyhton oder Perl), da sie einen schnellen Entwicklungsprozess ermoeöglichen und ohne Neukompilation auf unterschiedlichen Webservern lauffaehig sind. Wir waehlten Perl, da wir mit Perl am besten vertraut sind.

Weiterhin werden einige Standardprogramme wie ghostscript ('gs') und die netpbm-tools zur Bildkonvertierung benoetigt, die eigentlich auf allen Unixsystemen verfuegbar sein sollten.

3 Wiederverwendung von Quelltext aus anderen Projekten

Da wir schon ausgiebige Erfahrungen bei der CGI-Programmierung mit Perl gesammelt haben, haben wir uns entschieden, unser Projekt sauber von Grund auf neu zu schreiben. Bei der Integration von R orientierten wir uns an dem Quellcode von R-Web[11]. Dabei sind uns eklatante Sicherheitsprobleme aufgefallen. So ist es unter R-Web zum Beispiel moeglich, mit dem R-Befehl 'exec' (entspricht dem UNIX-Systemaufruf) beliebige Programme auf dem Server auszufuehren. Auf eine Mail an den Autor von R-Web, die die Sicherheitsproblematik schilderte, haben wir keine Antwort erhalten. Dies ist nicht verwunderlich, da das Projekt schon seit langem nicht mehr gepflegt wird. Bei R-Online entschieden wir uns, alle sicherheitskritischen R-Befehle vor der Ausfuehrung des R-Programms aus dem Quelltext zu entfernen.

4 Projektbeginn und auftauchende Probleme

Obwohl Javascript standardisiert ist, werden viele Javascript-Funktionen in manchen Browsern falsch oder gar nicht interpretiert. Daher kam es des oefteren vor, dass Programmteile, die auf unserer Entwicklungsplattform (Debian GNU/Linux mit Mozilla) einwandfrei funktionierten, nach einem Kompatibilitaetscheck mit dem Internet Explorer und Netscape 4.x (jeweils Windows und MacOS X), neu implementiert werden mussten.

Vergleichbare Probleme gab es bei der Darstellung der fuer das User Interface benutzten HTML-Seiten. Es war nicht moeglich, auf allen Browsern eine abolut identische Oberflaeche

zu erreichen.

5 Das fertige Programm

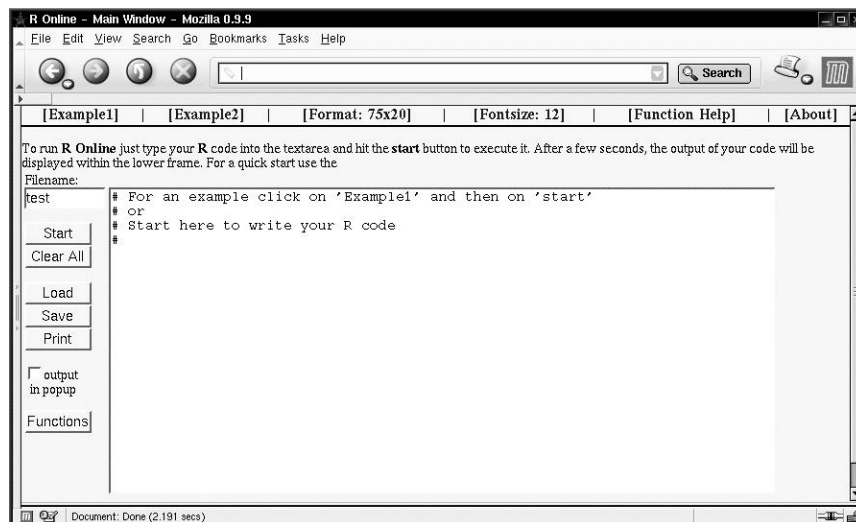


Abbildung 2: R-Online im Browser

5.1 Bedienung

Wie aus Abbildung 2 zu erkennen ist, besteht die Oberfläche zum grössten Teil aus dem Editorfeld zur Programmeingabe. Auf der linken Seite befinden sich:

- Der “Start”-Button zum Ausführen des Programms
- Ein “Clear-All”-Button zum Löschen des Editorfeldes
- “Load” und “Save” zum Laden/Speichern des Programms auf/von der eigenen Festplatte
- Der “Print”-Button zum Ausdrucken des Quelltextes
- Das “Functions”-Menue, welches vordefinierte Funktionen zum schnellen Einstieg in die Sprache R enthaelt.

Ueber die Links am oberen Rand erreicht man:

- Beispiele (einfache, eigenstaendige Programme)
- Menues, in denen die Groesse der Schrift und des Editorfeldes individuell eingestellt werden kann
- Funktionserklaerungen zu allen Funktionen des base- und ts-Paketes
- Einen Erklaerungstext zu R-Online mit Link zur Homepage

5.2 Eine sehr kurze Einfuehrung

Damit man ein Gefuehl fuer die Sprache bekommt, kann man als erstes einmal die beiden integrierten Beispiele laden und starten. So bekommt man schnell einen ersten Eindruck ueber die Syntax und die Maechtigkeit von R. Zur schnellen Erstellung der ersten eigenen Programme bietet das 'Functions'-Menue einige vorgefertigte Routinen bzw. Aufrufe.

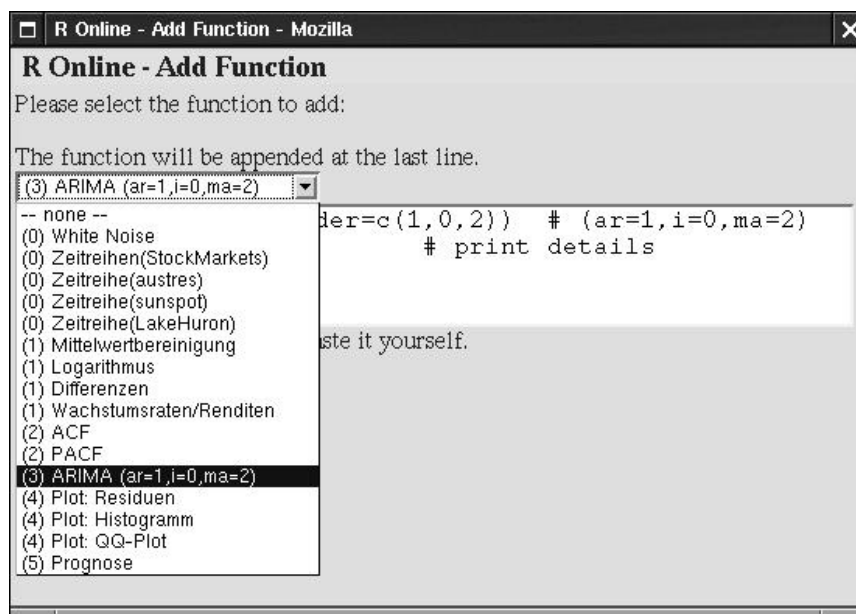


Abbildung 3: Vordefinierte R-Funktionen

Dabei gliedern sich diese Funktionen in sechs unterschiedliche Kategorien:

- (0) Zeitreihen
- (1) Vortransformation der Zeitreihe
- (2) Modellspezifikation

- (3) Modell-Schaetzung
- (4) Modell-Check
- (5) Prognose

Man kann jetzt nacheinander aus jeder Kategorie etwas auswahlen, und dann evtl. die Parameter anpassen. Durch erneutes Starten des Programmes kann man sich dann davon ueberzeugen, ob der Austausch von Funktionen oder das Aendern von Parametern eine verbesserte Prognoseguete zur Folge hat.

In den folgenden Screenshots (Abb. 4-7) kann man einige der Moeglichen Diagrammartentypen sehen. Sie stammen alle aus dem ersten Beispiel. Das Ergebnis der Analyse ist in Abbildung 7 zu sehen.

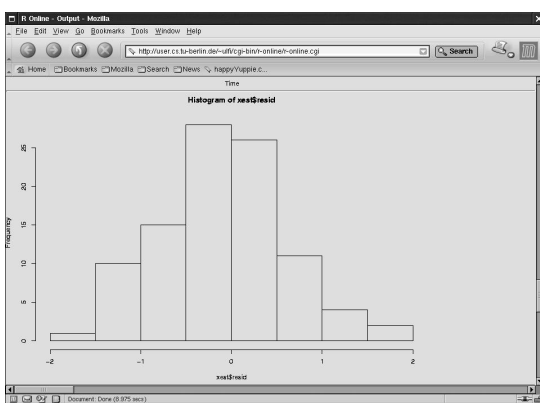


Abb. 4: Histogramm

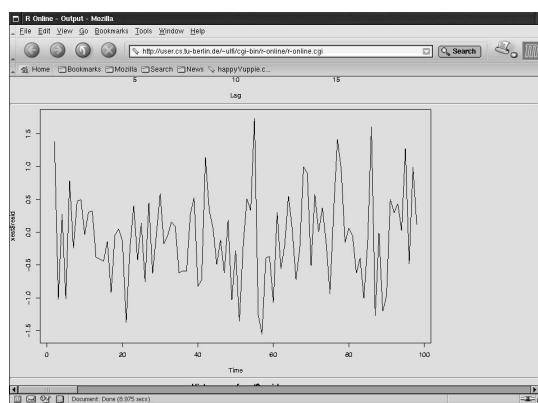


Abb. 5: Residuen

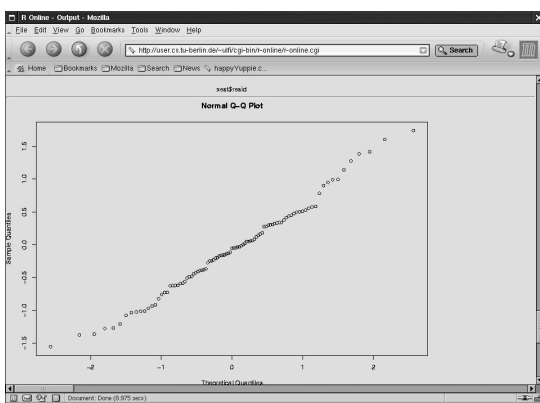


Abb. 6: QQ-Plot

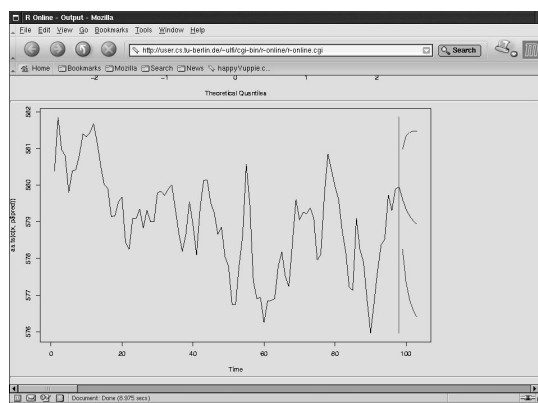


Abb. 7: Prognose mit Konfidenzintervall

Als weitere gute Lektuere empfehle ich 'Time Series Model for Changjiang River Monthly Flow at Hankou'[14], in der die wichtigsten Methoden der Zeitreihenanalyse, sowie saisonale Modelle anhand von kleinen R-Programmen erklart werden.

Mittlerweile wurde R-Online in 'Frequently Asked Questions on R'[7] (R-FAQ) von Kurt Hornik in die Kategorie 'R Web Interfaces' aufgenommen.

6 Modularisierung

6.1 Trennung des Editors vom R-Kern

Als alle wichtigen Funktionen implementiert waren, wurde R-Online in ein Perl-Modul ('online_editor.pm') und ein cgi-Skript ('r-online.cgi') getrennt. Das Perl-Modul enthaelt alle Grundfunktionen des Editors, waehrend das cgi-Skript die R-spezifischen Teile enthaelt (z.B. das Aufrufen des R-Interpreters, Beispiele und vordefinierte Funktionen). Durch Aenderungen des cgi-Skriptes ('r-online.cgi') laesst sich R-online an eigene Beduerfnisse anpassen. Damit kann man die Oberflaeche z.B. inhaltlich auf den Themenbereich einer Veranstaltung abstimmen. Sogar komplette andere Programmiersprachen lassen sich leicht integrieren. Da alle sprachunabhaengigen Funktionen ausgegliedert sind und der Quelltext gut dokumentiert ist, benoetigt man dazu nur grundlegende Perl-Kenntnisse.

6.2 Benutzung des Editors fuer andere Programmiersprachen

Um zu demonstrieren, dass sich der Editor von R-Online problemlos auch fuer andere Programmiersprachen als R benutzen laesst, wurde je eine Version fuer Perl [12] und Opal [13] erstellt. Die Grundfunktionen des Editors bleiben unveraendert. Es wurden nur die Beispiele, die vordefinierten Funktionen und die Hilfetexte angepasst und ein anderer Interpreter eingebunden.

6.3 Online-Editor fuer Perl

Perl ist eine Programmiersprache, die einerseits die wichtigsten Programmierbefehle wie Schleifen, Verzweigungen, etc. enthaelt und andererseits aber auch die Moeglichkeit bietet, leicht Such- und Ersetzungsoperationen wie in einem Editor durchzufuehren. Diese Moeglichkeiten machen Perl zu einer der beliebtesten Sprachen fuer dynamische Web-Seiten und zum Parsen von Text-Dateien.

Ein Web-basierter Perl Editor (Abbildung 8) ermoeglicht es Anfaengern die Sprache kennenzulernen, ohne Perl zu installieren. Durch Benutzung der vorgefertigten Funktionen und

Beispiele gelangt man schnell und ohne das Lesen langer Hilfetexte zu ersten funktionierenden Perl-Programmen. Fuer grosse Projekte ist der Perl-Editor jedoch nicht geeignet, da er z.B. aus Sicherheitsgruenden keine Moeglichkeit bietet, von einem Perl-Programm aus direkt auf Dateien zuzugreifen. Zur Gewaehrleistung der Sicherheitsaspekte wird automatisch das Perl-Modul 'Safe.pm' eingebunden.

Aber auch fortgeschrittene Perl-Programmierer haben mit dem Editor die Moeglichkeit, das Funktionieren, z.B. von regulaeren Ausdruecken, schnell und einfach zu testen.

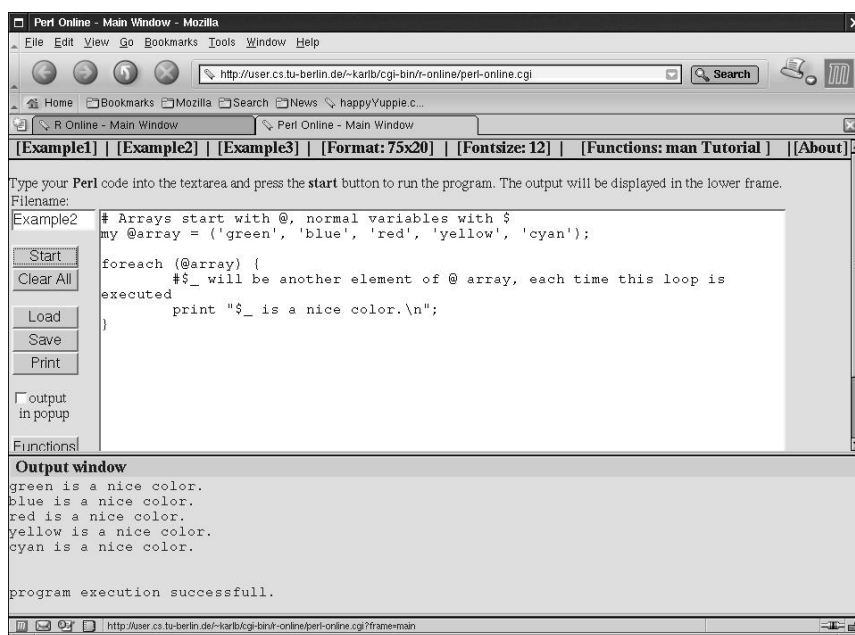


Abbildung 8: Der Perl-Editor (perl-online.cgi)

6.4 Online-Editor fuer Opal

Opal ist eine funktionale Programmiersprache, die an der TU-Berlin entwickelt wird. Da im Rahmen der Veranstaltung Informatik 1 das Programmieren in Opal unterrichtet wird, beoetigen viele Erstsemestler eine Entwicklungsumgebung fuer Opal. Die Opal-Installation ist jedoch haeufig problematisch und nur auf einem Unix-System moeglich.

Ein Browser-basierter Opal-Editor, mit dem man zuhause Opal-Programme entwickeln kann, bietet deshalb fuer viele Erstsemestler eine gute Moeglichkeit zur Bewaeltigung der Übungsaufgaben. Die eingebauten Beispiele und Funktionen, welche an den Inhalt der Informatik-1-Veranstaltung angelehnt sind, sollten den Einstieg zusaetzlich erleichtern.

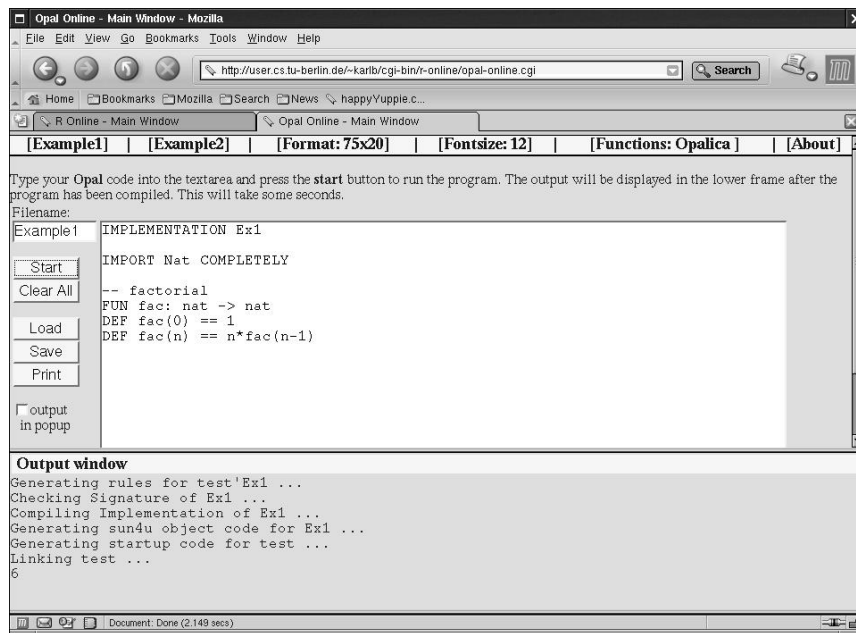


Abbildung 9: Die Opal-Version des Editors

7 Rueckblick

Abschliessend kann man erkennen, dass unsere anfaenglichen Ziele nahezu komplett erreicht wurden. Es hat sich herausgestellt, dass wir den Arbeitsaufwand realistisch eingeschaezt haben. Zusaetzlich zu den urspruenglichen Zielen ist uns eine Modularisierung des Programms gelungen, die neue Anwendungsmoeglichkeiten eroeffnet. In Erinnerung an unseren Studiumsanfang, hoffen wir, speziell den Erstsemestlern der TU-Berlin, mit dem Opal Editor den Einstieg zu erleichtern.

Die vielen Probleme, die aus dem unterschiedlichen Verhalten der einzelnen Browser und Betriebssysteme resultieren, haette man durch den Einsatz Java verhindern koennen. Erweiterungen wie z.B. Syntaxhighlighting waeren dadurch moeglich geworden. Eventuell waere der Einsatz von Java deswegen guenstiger gewesen.

Externe Entwickler konnten wir fuer unser Projekt leider nicht gewinnen. Eine Zusammenarbeit mit den Mitgliedern des Opal@Home[15]-Projekt es war aufgrund der unterschiedlichen Anforderungen und Ziele bedauerlicherweise nicht moeglich.

8 Zukunft von R-Online

Da der Funktionsumfang von R-Online bereits vollstaendig ist, liegt der Schwerpunkt zukuenftiger Entwicklungen in der Verbesserung der Browserkompatibilitaet sowie der Fehlerbereinigung. Eine Liste der bekannten Fehler und Inkompatibilitaeten findet man auf der Projekthomepage. Auch die Schnittstelle zwischen dem Editor und dem Sprachen-Backend koennte evtl. ueberarbeitet werden.

Wir hoffen, das sich externe Entwickler finden, die weitere Sprachmodule entwickeln.

Anhang

Homepage

Die Hompage dieses Projekts ist <http://www.osvisions.org/r-online/>. Dort ist auch immer die aktuellste Version erhaeltlich. Eine aktuelle funktionsfaehige Version von R-Online steht unter [2] zum Testen bereit. Die Module fuer Perl und Opal befinden sich als weitere Beispiele im gleichen Source-Paket. Online testen kann man sie unter [12] und [13].

Installation

Will man nur die eingebauten Beispiele oder vordefinierten Funktionen modifizieren, bzw. Aenderungen an der Oberflaeche vornehmen, reicht eine einfache Installation:

- r-online.cgi und online_editor.pm in den lokalen cgi-Pfad kopieren
- evtl. den Ort des Perl-Interpreters anpassen (in der Regel nicht noetig)
- die Variable *\$THISFILE* (in r-online.cgi) mit der URL des Scripts belegen.

Fuer eine komplette Installation sind weiterhin folgende Schritte durchzufuehren:

- komplette R-Installation (siehe [3]), die vom Webserver aus erreichbar ist
- ghostscript (gs) sowie die Programme pstopnm, ppmtogif, pnmcrop und pnmflip aus dem netpbm Packet

- Anpassung der im Skript gekennzeichneten Pfade fuer diese Programme.

Anmerkungen zur Installation an der TU-Berlin:

Im Informatik-Netz (cs) der Uni werden die Homeverzeichnisse vom Webserver unter einem anderen Pfad importiert. D.h., dass eine Datei, die unter `/home/u/ulfi/www/cgi-bin/` liegt, auf dem Webserver unter `/export/mdd0/all/u/ulfi/www/cgi-bin/` zu finden ist. Ausserdem ist das auf den Solaris Rechnern verfuegbare `gmake` zu `alt`, um R bzw. das `netpbm`-Paket zu kompilieren. Abhilfe schafft eine aktuelle Version von <http://www.gnu.org/software/make/make.html>.

Literatur

- [1] <http://www.osvisions.org/r-online/>
- [2] <http://user.cs.tu-berlin.de/~ulfi/cgi-bin/r-online/r-online.cgi>
- [3] <http://www.r-project.org/>
- [4] <http://cran.r-project.org/>
- [5] <http://stat.ethz.ch/R-alpha/R-patched/doc/html/>
- [6] <http://www.perl.org>
- [7] <http://www.ci.tuwien.ac.at/~hornik/R/>
- [8] <http://www.insightful.com/products/default.asp>
- [9] <http://www.aptech.com/>
- [10] <http://www.xplore-stat.de/>
- [11] <http://www.math.montana.edu/Rweb/>
- [12] <http://user.cs.tu-berlin.de/~karlb/cgi-bin/r-online/perl-online.cgi>
- [13] <http://user.cs.tu-berlin.de/~karlb/cgi-bin/r-online/opal-online.cgi>
- [14] <http://www.cs.uiowa.edu/~rqin/s156project.htm>
- [15] <http://opaeditor.berlios.de/>