

Datenanalyse mit R

SoSe 2020

Christina Bogner

Version vom 24. April 2020

Contents

1	Vorwort	5
1.1	Organisatorisches	5
1.2	Sinn und Unsinn dieses Skripts	6
2	Der Kurs	7
2.1	Zuordnung zum Modul und Leistungsnachweis	7
2.2	Lernziele des Kurses	7
2.3	Was mir im Umgang miteinander wichtig ist	7
3	Erste Schritte in	9
3.1	Was ist R?	9
3.2	Was ist RStudio?	10
3.3	RStudio Cloud	10
3.4	Inhalt der live Einführung und der Übungen	10
4	Daten in R	13
4.1	Datenstrukturen erzeugen	13
4.2	Arten von Daten in R	15
4.3	Objekt, sag mir wer du bist	15
4.4	Datenlücken, Fehlschläge etc.	17
4.5	Inhalt der live Einführung und der Übungen	17
5	Aufgabensammlung	19
5.1	Erste Schritte	19
5.2	Laden und Speichern von Daten	20
5.3	Automatisieren: Skripte & Funktionen	20
5.4	Daten visualisieren	21
5.5	Effizientes Programmieren	22

Chapter 1

Vorwort

“And honey, we’re gonna do it in style”

— Fools Garden

1.1 Organisatorisches

Die Coronaviruspandemie verändert unser Leben und unser Lernen. Die UzK bittet Lehrende, zumindest zu Beginn des SoSe 2020 auf digitale Lernformen umzusteigen. Daher wird dieser Kurs als ein Onlinekurs beginnen. Abhängig von der (sehr dynamischen) Lage werden wir im weiteren Kursverlauf das Format anpassen. Bitte seien Sie nachsichtig, wenn nicht alles so klappt, wie in Präsenzveranstaltungen. Wir müssen aktuell alle sehr viel dazu lernen in Sachen digitale Lehre. Sie können sicher sein, dass das Geographische Institut bemüht ist, die Lehre so effizient wie möglich weiter laufen zu lassen, damit Sie in Ihrem Studium fortfahren können.

In dieser Veranstaltung werden wir folgende Werkzeuge verwenden:

1. **ILIAS**: die Online-Lernplattform der UzK. Entweder sind Sie bereits automatisch in dem Kurs registriert oder werden von mir per Hand angemeldet.
2. **Campuswire**: die Live-Chatplattform dient der allgemeinen Kommunikation und der Selbstorganisation des Lernens. Verwenden Sie diese, um Fragen mit Ihren Kommilitonen und mir zu diskutieren. Sie sollten eine Einladungsmail zu Campuswire erhalten haben.
3. **Zoom**: die Videokonferenz-Software werden wir für live Einführungen nutzen. Die Anmeldemodalitäten sind auf den Kursseiten in ILIAS erklärt.

1.2 Sinn und Unsinn dieses Skripts

Dieses Skript ist ein lebendiges Begleitdokument des Kurses. Es wird laufend angepasst und aktualisiert.

Ich nutze verschiedene Farbkästen, um wichtige Stellen hervorzuheben:

Infoblock
Achtung, wichtig!
Beispielblock
Lernziele
Zusammenfassung

Chapter 2

Der Kurs

2.1 Zuordnung zum Modul und Leistungsnachweis

Dieser Kurs gehört zum Modul *Fachmethodik I* oder *Fachmethodik II* und ist aus 4 SWS Praktikum und 2 SWS Seminar aufgebaut. Das wichtigste Ziel besteht darin, Ihnen einen sicheren Umgang mit R beizubringen.

Den Leistungsnachweis bildet ein benoteter Praktikumsbericht.

2.2 Lernziele des Kurses

- Daten für Analysen vorbereiten
- eigene wiederverwendbare Skripte schreiben
- eigene Funktionen schreiben
- einfache Datenanalysen durchführen
- Daten visualisieren
- Ergebnisse reproduzierbar im Praktikumsbericht darstellen

2.3 Was mir im Umgang miteinander wichtig ist

- Pünktlichkeit bei live und Präsenzsitzungen
- Gute Vorbereitung durch erledigen der blenden learning Einheiten und Hausaufgaben
- Respektieren anderer Meinungen
- Offenheit gegenüber neuen Sichtweisen, Themen und Methoden
- Geduld mit sich selbst und den anderen

Chapter 3

Erste Schritte in

- Layout und Bedeutung einzelner Fenster in RStudio kennen
- Anweisungen aus dem Skript an die Konsole schicken
- R als Taschenrechner benutzen
- erste Funktionen aufrufen
- Objekte mit eckigen Klammern [] ansprechen
- R-Hilfeseiten aufrufen

3.1 Was ist R?

R ist eine Programmiersprache für Datenanalyse und statistische Modellierung. Es ist frei verfügbar (*open source software*) und neben Python einer der am meisten benutzten Programmiersprachen zur Datenanalyse und -visualisierung. R wurde von Ross Ihaka und Robert Gentleman 1996 veröffentlicht Ihaka and Gentleman (1996). Es gibt für R eine Vielzahl von Zusatzpaketen, die die Funktionalität und die Einsatzmöglichkeiten enorm erweitern.

Sie können R für Ihren Computer auf der offiziellen R-Seite <https://www.r-project.org/> herunterladen und installieren. Auch die Pakete finden Sie dort unter CRAN (*The Comprehensive R Archive Network*). Auf den CRAN-Seiten finden Sie sogen. CRAN Task Views, eine Übersicht über Pakete in verschiedenen Themenbereichen. Für den Umweltbereich sind folgende Paketsammlungen besonders relevant:

- Environmetrics: Analyse von Umweltdaten
- Multivariate: Multivariate Statistik
- Spatial: Analyse von räumlichen Daten
- TimeSeries: Zeitreihenanalyse

Zu Beginn des Kurses, werden wir jedoch nicht auf Ihren lokalen Rechnern arbeiten, sondern in einer Cloud (s.u.). Das ermöglicht einen schnelleren Einstieg in R und bietet eine live Unterstützung durch den Dozenten beim Pro-

grammieren. Daher biete ich zu diesem frühen Zeitpunkt im Kurs keine Unterstützung bei der Installation. Für die ganz Ungeduldigen, gibt es hier eine kurze *Einleitung zur Installation*

3.2 Was ist RStudio?

RStudio Desktop ist eine Entwicklungsumgebung für R. Sie können die *open source* Version kostenlos für Ihren Rechner hier herunterladen.

Es gibt eine live Einführung in RStudio im Kurs. Zusätzlich können Sie hier ein Video dazu ansehen.

3.3 RStudio Cloud

Zu Beginn des Kurses werden wir in der RStudio Cloud arbeiten. Sie sollten eine Einladungsmail zu unserem Kurs in der Cloud bekommen haben. Ich werde in der Cloud Projekte für Sie anlegen (*assignment*), die Skripte, Arbeitsanweisungen etc. beinhalten. Wenn Sie auf so ein Assignment klicken, wird für Sie automatische ein Kopie des Projekts erstellt, in der Sie dann arbeiten können.

Der große Vorteil der Cloud ist, dass ich direkt in Ihre Projekte eingreifen kann, wenn es mal zu Fehlern kommt. Während ich in Ihrem Projekt arbeite, werden Sie kurz aus der R-Sitzung ausgeloggt, da die Cloud kein gleichzeitiges Arbeiten unterstützt. Nehmen Sie sich etwas Zeit, um die Cloud und die darin enthaltenen Tutorials kennen zu lernen.

Sowohl in der RStudio Cloud als auch in einer lokalen Installation, ist Ihr RStudio so aufgebaut wie in Abbildung 3.1.

3.4 Inhalt der live Einführung und der Übungen

- Überblick über RStudio
- R als Taschenrechner
- einfache Funktionen aufrufen
- Zuordnungen (*assignments*)
- Notation mit eckigen Klammern [] (*array*-Notation)
- Hilfeseiten aufrufen

Funktionen, die wir in der Session nutzen werden:

Funktion	Bedeutung	Beispielaufruf
<code>pi</code>	Zahl pi	<code>pi</code>
<code>sin</code>	Sinus	<code>sin(2)</code>
<code>cos</code>	Cosinus	<code>cos(2)</code>
<code>sqrt</code>	Quadratwurzel	<code>sqrt(2)</code>

Funktion	Bedeutung	Beispielaufruf
<code>c</code>	(<i>concatenate</i>) Fügt Daten zu einem Vektor zusammen	<code>c(1,2,3,4)</code>
<code>help.start</code>	Öffnet ein Browser-Fenster mit diversen Handbüchern	<code>help.start()</code>
<code>help.search</code>	Sucht nach einem Begriff in Hilfe-Dateien	<code>help.search('time')</code>
<code>??</code>	alias <code>help.search</code>	<code>??time</code>
<code>help</code>	Sucht nach einer Funktion	<code>?mean</code>
<code>?</code>	alias <code>help()</code>	<code>?mean</code>
<code>mean</code>	Mittelwert	<code>mean(c(1,2,3,4))</code>
<code>var</code>	Varianz	<code>var(c(1,2,3,4))</code>
<code>sd</code>	Standardabweichung	<code>sd(c(1,2,3,4))</code>
<code>sum</code>	Summe	<code>sum(c(1,2,3,4))</code>
<code>vector</code>	Generiert einen Vektor	<code>vector(length=3, mode='numeric')</code>

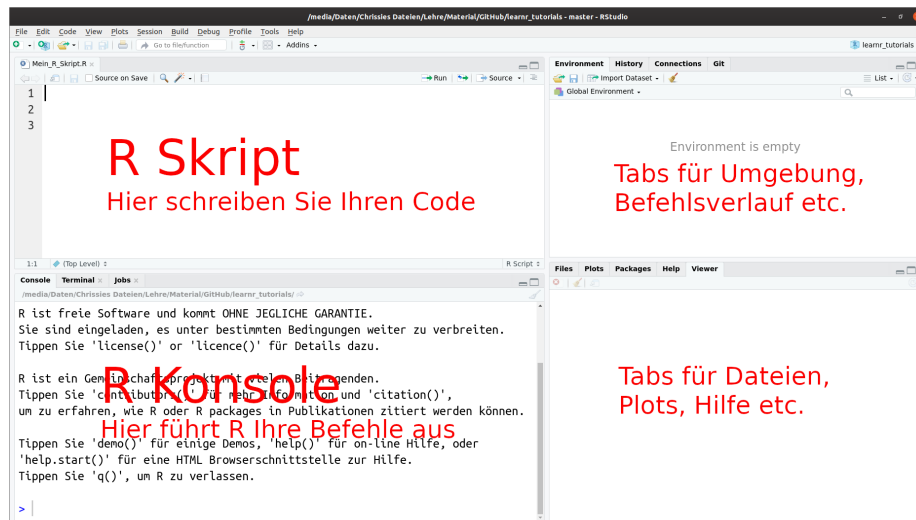


Figure 3.1: Aufbau von RStudio

Chapter 4

Daten in R

- Daten einlesen mit `read.table`
- Datenstrukturen erstellen
- Typen von Daten in R abfragen
- Daten speichern mit `write.table`

4.1 Datenstrukturen erzeugen

In R gibt es unterschiedliche Datenobjekte. Es ist wichtig, sich über die Struktur (oder Typ) des Datenobjekts Gedanken zu machen. Denn diese bestimmt, was mit einem Objekt gemacht werden kann und ob Funktionen damit richtig umgehen können. Schließlich ist es nicht egal, ob es sich bei einem Objekt um ein numerisches Objekt oder einfach Text (*character*) handelt.

Die wichtigsten Datentypen sind

- **Vektoren:** hier gruppiert man gleichartige Elemente, z.B. Zahlen. Auch eine einzelne Zahl (ein Skalar) wird von R wie ein Vektor behandelt.
- **Matrizen:** zweidimensionale (Zeilen und Spalten) Datentabellen mit gleichartigen Elementen.
- **Listen:** können beliebige Elemente beliebiger Länge enthalten.
- **Dataframes:** zweidimensionale Datentabellen, die beliebige Elemente enthalten können. Die Spalten der Dataframes müssen allerdings gleichartige Elemente enthalten. Dataframes sind eine Unterart von Listen.

Neben diesen Hauptstrukturen gibt es

- **Factor:** ein besonderer Vektor für kategorielle Variablen

Um diese Datenstrukturen zu erzeugen, gibt es jeweils eine Funktion mit gleichlautendem Namen.

```
# Vektor erzeugen
my_vect = vector(length = 3, mode = 'numeric')
my_vect
```

```
## [1] 0 0 0
```

```
# Matrix erzeugen
my_matrix = matrix(data = c(1:(3*4)), nrow = 3, ncol = 4)
my_matrix
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
```

```
# Dataframe erzeugen
my_dataframe = data.frame('Spalte_1' = rep('Text', 10),
                           'Spalte_2' = 1:10)
my_dataframe
```

```
##      Spalte_1 Spalte_2
## 1      Text         1
## 2      Text         2
## 3      Text         3
## 4      Text         4
## 5      Text         5
## 6      Text         6
## 7      Text         7
## 8      Text         8
## 9      Text         9
## 10     Text        10
```

```
# Liste erzeugen
my_list = list('Schachtel_1' = 3, 'Schachtel_2' = my_dataframe,
               'Schachtel_3' = 'Noch mehr Text')
my_list
```

```
## $Schachtel_1
## [1] 3
##
## $Schachtel_2
##      Spalte_1 Spalte_2
## 1      Text         1
## 2      Text         2
```

```
## 3      Text      3
## 4      Text      4
## 5      Text      5
## 6      Text      6
## 7      Text      7
## 8      Text      8
## 9      Text      9
## 10     Text     10
##
## $Schachtel_3
## [1] "Noch mehr Text"
```

```
# Factor erzeugen
my_factor = factor(c('R', 'RStudio', 'Cloud', 'Cloud', 'R', 'R'))
my_factor
```

```
## [1] R      RStudio Cloud  Cloud   R        R
## Levels: Cloud R RStudio
```

4.2 Arten von Daten in R

Die Datenstrukturen `vector`, `data.frame` usw. können unterschiedliche Arten von Daten enthalten.

Name	Beispiele
raw	3A, FE
logical	TRUE, FALSE
integer	1, 42, -3
numeric/double	3, 2.81, 6.032e23
complex	1.2+2.2i
character	"foo"

4.3 Objekt, sag mir wer du bist

Um die Struktur und/oder Datenart abzufragen, verwendet man `class`, `typeof`, `mode` und `storage.mode`.

```
class(my_vect)
```

```
## [1] "numeric"
```

```
typeof(my_vect)
```

```
## [1] "double"
```

```
class(my_dataframe)
```

```
## [1] "data.frame"
```

```
typeof(my_dataframe)
```

```
## [1] "list"
```

Mit `str` kann man das Innenleben eines Objekts anzeigen. Das ist besonders wichtig nach dem Einlesen von Daten, um das Ergebnis des Einlesens zu kontrollieren. Dabei kontrolliert man, dass z.B. alle numerischen Spalten auch als Zahlen eingelesen wurden und nichts schief gegangen ist.

```
str(my_dataframe)
```

```
## 'data.frame':  10 obs. of  2 variables:
## $ Spalte_1: Factor w/  1 level "Text": 1 1 1 1 1 1 1 1 1 1
## $ Spalte_2: int  1 2 3 4 5 6 7 8 9 10
```

Weitere Funktionen, die Auskunft über Objekte geben sind `length`, sinnvoll auf nur Vektoren und Listen, und `dim`, sinnvoll auf zweidimensionalen Datenobjekten. Wenn Sie versuchen, `dim` auf einem Vektor aufzurufen, gibt es `NULL` (s.u.), weil Vektoren keine Dimensionen haben. Wenn Sie `length` auf einem `data.frame` aufrufen, bekommen Sie die Anzahl der Dimensionen, nämlich 2. Das sind keine besonders spannenden Informationen .

```
length(my_vect)
```

```
## [1] 3
```

```
dim(my_vect)
```

```
## NULL
```

```
length(my_dataframe)
```

```
## [1] 2
```

```
dim(my_dataframe)
```

```
## [1] 10  2
```


4.4 Datenlücken, Fehlschläge etc.

Datenlücken werden in R mit `NA` kodiert, Fehlschläge bei Berechnungen mit `NaN` (not a number) und Vektoren der Länge 0 mit `NULL`. Letzteres wird häufig beim Aufruf von Funktionen benutzt, wenn man bestimmte Parameter ausschalten möchte. Die Benutzung muss aber immer in der Hilfe zur jeweiligen Funktion nachgeschlagen werden.

4.5 Inhalt der live Einführung und der Übungen

- Daten einlesen und `data.frame` erstellen: Aufgabe 5.2.1

Funktionen, die wir in der Session nutzen werden:

Funktion	Bedeutung	Beispielaufruf
<code>read.table</code>	Liest Daten aus einer Datei ein.	<code>read.table(file='Daten.txt', header=TRUE)</code>
<code>ls</code>	Zeigt den Inhalt des Workspaces.	<code>ls</code>
<code>head</code>	Zeigt den ersten Teil eines Objekts.	<code>head(x)</code>
<code>tail</code>	Zeigt den letzten Teil eines Objekts.	<code>tail(x)</code>
<code>str</code>	Zeigt die Struktur (Innenleben) eines Objekts an	<code>str(my_dataframe)</code>
<code>length</code>	Gibt die Länge eines Objekts.	<code>length(x)</code>
<code>dim</code>	Gibt die Dimension eines Objekts (Reihenfolge: Zeilen, Spalten)	<code>dim(x)</code>
<code>seq</code>	Erstellt eine regelmäßige Reihe.	<code>seq(from=-2, to=4, by=0.1)</code>
<code>data.frame</code>	Erstellt eine Datentabelle.	<code>data.frame(x,y,z)</code>
<code>colnames, rownames</code>	Benennt Spalten bzw. Zeilen eines Datenobjekts.	<code>colnames(x)</code>
<code>rm</code>	Löscht Objekte aus dem Workspace.	<code>rm(x)</code>
<code>summary</code>	Fasst ein Objekt zusammen.	<code>summary(x)</code>
<code>table</code>	Erstellt eine Häufigkeitstabelle.	<code>table(x)</code>

Funktion	Bedeutung	Beispielaufruf
<code>which</code>	Gibt die TRUE-Indices eines logischen Objekts.	<code>which(LETTERS == 'R')</code>
<code>history</code>	Zeigt die Liste mit ausgeführten Befehlen der Session.	<code>history</code>
<code>write.table</code>	Speichert Datenobjekte als Tabelle ab.	<code>write.table(x, file='Tabelle.txt')</code>
<code>save.image</code>	Speichert den Workspace.	<code>save.image(file='RSession.Rdata')</code>
<code>savehistory</code>	Speichert die History.	<code>savehistory(file='Myhistory.Rhistory')</code>

Chapter 5

Aufgabensammlung

5.1 Erste Schritte

5.1.1 Ars Haushaltsbuch

Der angehende Datenanalyst Ar Stat möchte dem Rat seiner Mutter folgen und ein Haushaltsbuch anlegen. Als erstes möchte er sich einen Überblick über seine Ausgaben in der Uni-Mensa verschaffen und erstellt die folgende Tabelle:

1. Wie viel hat Ar insgesamt in der Woche ausgegeben?
2. Wie viel hat er im Schnitt pro Tag ausgegeben?
3. Wie stark schwanken seine Ausgaben?

Leider hat Ar sich beim übertragen der Daten vertippt. Er hat am Dienstag seine Freundin zum Essen eingeladen und 7,95 € statt 2,90 € ausgegeben.

4. Korrigieren Sie Ars Fehler.
5. Wie verändern sich die Ergebnisse aus den Teilaufgaben 1 bis 3 Warum?

Table 5.1: Ars Mensaausgaben

Wochentag	Ausgaben
Montag	2,57
Dienstag	2,90
Mittwoch	2,73
Donnerstag	3,23
Freitag	3,90

5.2 Laden und Speichern von Daten

5.2.1 Bestandesaufnahme

Ar Stat arbeitet als HiWi in der AG Ökosystemforschung und soll im Nationalpark Eifel eine Bestandsaufnahme durchführen. Er notiert den BHD (Brusthöhendurchmesser) und die Art der Bäume.

1. Lesen Sie den Datensatz `BHD.txt` ein und ordnen Sie ihn der Variable `BHD` zu.
2. Erstellen Sie einen Vektor `a` mit Baumnummern. Von welcher Art sind die Elemente des Vektors `a`?
3. Fügen Sie die Datensätze `BHD` und `a` zu einem `data.frame` zusammen und benennen Sie die Spalten sinnvoll.
4. Löschen Sie den Vektor `a`.
5. Lesen Sie den Datensatz `Art.txt` ein und ordnen Sie ihn der Variablen `art` zu.
6. Fügen Sie die Art in den `data.frame` ein.
7. Erstellen Sie eine Tabelle mit der Anzahl der jeweiligen Arten. Nutzen Sie die Funktion ‘`table`’.
8. Speichern Sie die Tabelle mit `write.table`.
9. Welche Baumart hat im Mittel einen größeren Brusthöhendurchmesser?

5.3 Automatisieren: Skripte & Funktionen

5.3.1 R-Hausaufgaben

An dem Kurs “Einführung in R” nehmen 49 Studierende teil. Der Leistungsnachweis besteht aus Hausaufgaben, die insgesamt mit 100 Punkten bewertet werden. Ab 50 Punkten gilt der Kurs als bestanden.

1. Lesen Sie den Datensatz `R-HAs.txt`, der die Endpunkte enthält, ein.
2. Ermitteln Sie, wie viele Teilnehmer bestanden und wie viele nicht bestanden haben.

5.3.2 Fledermäuse, die Zweite

Wir beschäftigen uns erneut mit den Fledermäusen.

1. Lesen Sie den korrigierten(!) Datensatz `\texttt{Fledermaus_cor.txt}` ein.
2. Schreiben Sie eine Funktion, die den Entwicklungsstand der Tiere klassifiziert. Nutzen Sie dazu die ad hoc Regel: Individuum < 5 cm ist ein Jungtier, sonst erwachsen.
3. Erstellen Sie eine ordinal-skalierte Variable `alter` mit dem Entwicklungsstand der Tiere.
4. Schreiben Sie eine Funktion, die die Mittelwerte der Größe für weibliche und männliche Individuen berechnet.

5. Berechnen Sie die Mittelwerte der Größe und runden Sie auf 2 Nachkommastellen.

5.3.3 Unfaire Klausur?

Ar belegt im 4. Semester die Veranstaltung “Spaß mit R”. Bei der Klausur gibt es 2 Aufgabengruppen mit jeweils 60 Punkten. Aufgabengruppe 1 wird an Studierende auf ungeraden Sitzplätzen und Aufgabengruppe 2 an Studierende auf geraden Sitzplätzen ausgegeben.

1. Lesen Sie den Datensatz `Klausurpunkte.txt` ein.
2. Überprüfen Sie Ars Vermutung, dass die Aufgabengruppe 1 im Schnitt leichter war als Aufgabengruppe 2 (d.h. in der Gruppe 1 im Schnitt mehr Punkte erzielt wurden).

5.4 Daten visualisieren

5.4.1 Lange Bramke (Harz)

Im Harz wurden über eine längere Zeit Niederschlag, Abfluss und Temperatur gemessen.

1. Laden Sie den Datensatz `Data.dat`.
2. Stellen Sie die Temperatur in einem Scatterplot dar. Welche Darstellungsart (Argument `type` in `plot`) erscheint Ihnen am sinnvollsten?
3. Beschriften Sie die Graphik und fügen Sie einen Titel hinzu.
4. Speichern Sie die Graphik als pdf ab.
5. Stellen Sie die Niederschläge in einem Diagramm dar. Wählen Sie einen geeigneten Darstellungstyp mit `type` (Tipp: geben Sie für die Hilfe `?plot` in die Console ein).

5.4.2 Temperatur-Datensatz

1. Laden Sie den Temperatur-Datensatz aus Zuur et al. (2009).
2. Berechnen Sie die Monatsmittelwerte für alle Stationen, sowie die Standardabweichungen.
3. Stellen Sie die Monatsmittel der Temperatur in einem Barplot dar.
4. Beschriften Sie die Graphik sinnvoll.
5. Fügen Sie die Standardabweichungen zu den einzelnen Balken hinzu.

5.4.3 Artenvielfalt in Grasländern

Sie erhalten Daten aus dem Grasland-Monitoring im Yellowstone Nationalpark und dem National Bison Range (USA). Das Ziel des Monitorings ist die Untersuchung möglicher Änderungen der Biodiversität und des Zusammenhang mit Umweltfaktoren. Biodiversität wurde durch die Anzahl unterschiedlicher

Arten quantifiziert. Insgesamt haben die Forscher ca. 90 Arten in 8 Transekten kartiert. Die Aufnahmen wurden alle 4 bis 10 Jahre wiederholt. Insgesamt liegen 58 Beobachtungen vor. Die Daten sind in der Datei **Vegetation2.xls** gespeichert.

1. Laden Sie den Datensatz in R und sehen Sie sich das Ergebnis genau mit **str**, **head** und **tail** an. Diese Aufgabe dient dazu, das Einlesen von Excel-Dateien zu erarbeiten. Tipp: eine mögliche Bibliothek, die dabei helfen kann, wäre **xlsx**.
2. Berechnen Sie den Mittelwert und die Standardabweichung der Artenzahl (Variable **R**) pro Transekt.
3. Plotten Sie die Artenzahl gegen die Variable **BARES0IL** (Anteil von unbewachsenem Boden).
4. Benutzen Sie unterschiedliche Symbole pro Transekt, erstellen Sie eine Legende.
5. Beschriften Sie die Graphik sinnvoll und speichern Sie sie als pdf ab, ohne die Maus zu benutzen.

5.4.4 Tracerversuche

Im Waldstein wurden Tracerversuche mit dem Farbstoff Brilliant Blue durchgeführt und die gefärbten Bodenprofile *binärisiert* (d.h. in ein schwarz-weiß Bild umgewandelt). Schwarze Pixel stellen gefärbten Boden und weiße ungefärbten dar. Aus diesen Binärbildern wurde anschließend eine Reihe von Kenngrößen berechnet.

1. Lesen Sie die Datei `\texttt{Waldstein2005_ind.txt}` ein. Die Tiefe eines Profils ist 579 Pixel und es liegen 6 Profile untereinander in der Spalte d.
2. Berechnen Sie die 5%, 50% und 95% Quantile des Färbeanteils (Index d) der 6 Profile.
3. Stellen Sie den Median des Anteils der Färbung mit der Tiefe dar und fügen Sie die Quantile als transparente Fläche hinzu (Tipp: **polygon**).

5.5 Effizientes Programmieren

5.5.1 Lagerungsdichten

Auf 10 verschiedenen landwirtschaftlichen Feldern wurden im Oberboden je 25 Stechzylinder entnommen.

1. Lesen Sie den Datensatz `r file.name("Bodendaten.txt")` ein.
2. Bestimmen Sie die mittlere Lagerungsdichte pro Feld.

5.5.2 Temperatur-Datensatz, revisited

1. Laden Sie den Temperatur-Datensatz aus Zuur et al. (2009), Datei `Temperatur.csv`.
2. Berechnen Sie die Jahresmittelwerte je Station. (Tipp: Hilfe von `tapply` genau lesen!)

Bibliography

- Ihaka, R. and Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314.
- Zuur, A. F., Ieno, E., and Meesters, E. (2009). *A Beginner's Guide to R*. Springer.