Data Science für die Humangeographie: Ein pragmatischer Einstieg mit R

Konzeption quantitativer Forschung

Till Straube
straube@geo.uni-frankfurt.de

Institut für Humangeographie
Goethe-Universität Frankfurt

Wintersemester 2020/21

Contents

Terminüberblick

Alle Sitzungen finden von 13 bis 16h c.t. statt

Datum	Sitzung	Inhalt
2. November 2020	1	Vorbesprechung
9. November 2020	2	Erste Schritte
16. November 2020	3	Text: Anderson 2008
23. November 2020	4	Datenstrukturen
30. November 2020	5	Visualisierungen
7. Dezember 2020	6	Text: Shelton et al. 2014
14. Dezember 2020	7	Geodaten
11. Januar 2021	9	Choroplethen
18. Januar 2021	10	Text: Chandra 2014
25. Januar 2021	11	HTML-Tabellen
1. Februar 2021	12	Web Scraping
8. Februar 2021	13	Text: Straube 2021
15. Februar 2021	13	Präsentationen
31. März 2021		*Abgabe Exposé*

Online-Ressourcen

R Tutorials und eBooks

- **R for Data Science**
  https://r4ds.had.co.nz/
  Ausführliches Handbuch, Fokus auf Data Science
- **RStudio Cloud Primers**
  https://rstudio.cloud/learn/primers/1
- **Swirl**
  https://swirlstats.com/students.html
  Interaktives Tutorial als R-Paket, mit verschiedenen Lektionen
- **Quick-R**
  https://www.statmethods.net/r-tutorial/index.html
  Überblickartiges Tutorial, kurz und bündig
- **RStudio Cheat Sheets**
  https://www.rstudio.com/resources/cheatsheets/
  Einseitige Cheat Sheets zu verschiedenen Themen
- **Google's R Style Guide**
  https://google.github.io/styleguide/Rguide.xml
  Regeln für leserlichen R Code

Inspiration für Visualisierungen

- **R Graph Gallery**
  https://www.r-graph-gallery.com/
  Viele Beispiele für verschiedenste Visualisierungen
- **DDJ Katalog**
  http://katalog.datenjournalismus.net/#/
  Portfolio Datenjournalismus, leider etwas veraltet

- **Subreddits**
  https://www.reddit.com/r/dataisbeautiful
  https://www.reddit.com/r/DataArt/
  https://www.reddit.com/r/MapPorn/

Spezialthemen

- **Tutorial Reguläre Ausdrücke**
  https://danielfett.de/en/tutorials/tutorial-regulare-ausdrucke/
  Deutschsprachige Einführung zu regulären Ausdrücken

1 Vorbesprechung

1.1 Überblick

1.1.1 Seminar im Curriculum

- Dieses Seminar ist Bestandteil des Moduls BA3.
- Das Projektseminar besteht aus zwei Teilen über zwei Semester:
  - Konzeption quantitativer Forschung (Wintersemester)
  - Analyse quantitativer Daten (Sommersemester)
- Im Winter gibt es 12 inhaltliche Termine (davon 4x Textarbeit).
- Im Sommer wird das Seminar mit den selben Teilnehmer*innen fortgeführt.

1.1.2 Lernziele für das Wintersemester

Sie können…

- einfache Skripte in R eigenständig erstellen.
- Datensätze in vielfältigen Formaten visualisieren.
- Online-Ressourcen gezielt einsetzen.
- Möglichkeiten der Datenbeschaffung identifizieren.
- epistemologische Verschiebungen durch Data Science wiedergeben.

1.1.3 Technische Anforderungen

- Es sind keine Vorkenntnisse in R erforderlich.
- Sie brauchen einen Laptop, mit dem Sie gut arbeiten können.
- Wir benutzen die RStudio Cloud als Plattform.
- Sie brauchen einen ruhigen Arbeitsplatz.

1.1.4 Unterstützung im Corona-Semester

- Die Uni bietet einen “Semesterlaptop” an.
- Bei Bedarf kann ich gerne versuchen, Arbeitsplätze im Seminarraum (PEG) anzubieten.
- Bitte kontaktieren Sie mich per E-Mail, falls Sie einen Arbeitsplatz regelmäßig in Anspruch nehmen wollen würden.

1.2 Seminarformat

- Das Seminar findet jede Woche Montags, 13–16h c.t. statt.
- Der Zoom-Link, den Sie per E-Mail erhalten haben, bleibt gleich.
- Wir machen um ca. 14:25h eine zehnminütige Pause.
- Für Textbesprechungen wird die Gruppe zweigeteilt.
- Dieses Seminar findet in verschiedenen Modi statt:

1.2.1 Input und Plenum

- Ich rede oder moderiere (mit Folien oder ohne)
- Sie hören mir und Ihren Kommiliton*innen aufmerksam zu
- Sie "melden" sich für Redebeiträge oder Fragen (Zoom-Funktion)
- Die*der Chat-Verantwortliche unterbricht mich bei Klärungsbedarf

1.2.2 Think-pair-share

- Sie bearbeiten eine Fragestellung in zufälligen Zweier-Konstellationen (Breakout-Session)
- Nach einer vorgegebenen Zeitspanne kehren Sie ins Plenum zurück
- Ich fordere Sie ggf. auf, Ergebnisse und offene Fragen mit der Gruppe zu teilen

1.2.3 Follow the recipe

- Ich teile ein unvollständiges Beispielprojekt.
- Wir gehen die Teilschritte nach und nach durch.
- Ich "habe den Plan", stelle aber immer wieder Fragen ans Plenum.
- Sie vollziehen die Schritte an Ihrer eigenen Kopie des Projekts nach.
- Die*der Chat-Verantwortliche unterbricht mich bei Klärungsbedarf

1.2.4 Hands-on session

- Sie bearbeiten praktische Aufgabenstellungen alleine.
- Dabei sind sie in zufälligen Dreier-Konstellationen (Breakout-Session).
- Bei Fragen oder Problemen wenden Sie sich zunächst an Ihre Kleingruppe.
- Falls Sie nicht weiterkommen, fordern Sie Hilfe an (Zoom-Funktion).
- Ich reagiere auf Hilfegesuche oder schaue in zufälligen Gruppen vorbei.

1.2.5 Share your work

- Ich wähle eine Teilnehmer*in zufällig aus.
- Die Person teilt ihren Bildschirm und berichtet von ihrer Bearbeitung eines Problems.
- Alle anderen unterstützen solidarisch durch aktives Nachvollziehen, Nachfragen und Hinweise.

1.3 Leistungsnachweise

1.3.1 Exposé

- Zum Ende des Wintersemesters geben Sie ein Exposé für ein Untersuchungsvorhaben für das Sommersemester ab.
- Sie können sich mit bis zu vier Personen zusammenschließen.

- Die Projektgruppe besteht dann verbindlich für das Sommersemester.
- Damit steigen aber auch die Anforderungen an Umfang, Detail und technischen Anspruch.
- Umfang für das Exposé: max. 15k Zeichen inkl. Leerzeichen, exkl. Literaturverzeichnis
- Als Abgabetermin haben wir den 31. März vereinbart.

1.3.1.1 Inhalte

- Einführung ins Thema
- Forschungsstand / Literaturüberblick
- Herleitung einer klar abgegrenzten (vorläufigen) Forschungsfrage
- Konkrete Datenquellen
- Ideen für Verfahren und Visualisierungen

1.3.1.2 Bewertungskriterien

Alle kriterien werden mit einer (runden) Schulnote bewertet. Der gewichtete Schnitt ergibt die Gesamtnote.

Kriterium	Gewichtung	Erläuterung
Ziterweise und Formatierung	10%	Der Text erfüllt formale Anforderungen an Wissenschaftlichkeit.
Ausdruck und Rechtschreibung	10%	Der Text ist sprachlich gelungen.
Roter Faden	10%	Der Text ist übersichtlich strukturiert und die Einzelteile greifen gut ineinander.
Literatur	10%	Die zitierten Quellen sind für eine Einführung ins Thema geeignet und werden gut zusammengefasst.
Theorie	10%	Relevante wissenschaftliche Perspektiven werden anhand von geeigneter Fachliteratur aufgezeigt.
Fragestellung	10%	Die Forschungsfrage ist für das Vorhaben geeignet und wird überzeugend hergeleitet.
Datenquellen	20%	Die Datenquellen sind geeignet und detailliert beschrieben.
Design	20%	Das Untersuchungsvorhaben ist nachvollziehbar beschrieben, und der technische Anspruch ist dem Projektseminar angemessen.

1.3.2 Anwesenheit

- Es besteht Anwesenheitspflicht.
- Für Ihre ersten zwei Fehltermine pro Semester brauche ich keine Entschuldigung (aber Sie sollten das ggf. mit ihrer Projektgruppe absprechen).

- Sie sind dann selbstständig für die Nacharbeit der behandelten Themen zuständig.
- Im Falle eines zusätzlichen Fehltermins brauche ich ein Attest und einen Nachweis über Nacharbeit.
- Zur Anwesenheit gehört…
  - uneingeschränkte Aufmerksamkeit über die komplette Veranstaltungsdauer,
  - aktive Mitarbeit an Beispielen,
  - Bearbeitung von Übungsaufgaben,
  - aktive Beteiligung an Diskussionen.
- Eine eingeschaltete Kamera macht das allen Beteiligten leichter!

1.4 Lehrphilosophie

- Die folgenden vier "Säulen" habe ich mal im Rahmen einer Fortbildung als meine Lehrphilosophie definiert.
- Sie spiegeln meinen eigenen Anspruch an meine Lehre wider und sind als Vorschlag für ein gutes Miteinander zu verstehen.
- Begreifen Sie das gerne auch als Ermunterung, Aspekte hiervon einzufordern, wenn sie in der Veranstaltung zu kurz kommen.

1.4.1 Transparenz

- Erforderliche Leistungen und Bewertungskriterien sind vorab bekannt.
- Termine und Regelungen werden in der Vorbereitungssitzung verbindlich vereinbart.
- Aktuelle Lehrmaterialien stehen online durchgängig zur Verfügung.

1.4.2 Praktische Übungen

- Eigenständige Anwendung steht im Vordergrund.
- Verfahren und Techniken werden mit Beispielen und Übungen erarbeitet.
- Die perfekte Aufgabe ist immer ein bisschen "zu schwer".
- Toleranz für Frustration ist eine wichtige Fähigkeit und lässt sich trainieren.

1.4.3 Geschützte Räume

- Alle können sich im Plenum respektiert und sicher fühlen. Verletzendes Verhalten wird benannt.
- Es gibt einen vertrauensvollen Rahmen für ehrlichen Austausch.
- Frustrationen und Momente des Scheiterns werden ernst genommen und konstruktiv bearbeitet.

1.4.4 Kritische Reflexion

- Auch Teilnehmende, die kein weiterführendes Interesse an der Anwendung quantitativer Verfahren haben, sind im Seminar gut aufgehoben.
- Verfahren werden kontextualisiert, ihre Limitationen werden aufgezeigt.
- Kritische Forschung zu quantitativen Praktiken wird besprochen.

2 Erste Schritte

2.1 Vorbereitung

- Machen Sie sich einen kostenlosen Account auf https://rstudio.cloud
- Treten Sie dem Seminar-Workspace bei. (Sie erhalten eine Einladung per E-Mail.)
- Optional/alternativ: installieren Sie R und RStudio auf Ihrem Computer.

2.2 Lernziele für diese Sitzung

Sie können…

- Rechenoperatoren einsetzen.
- Variablen zuweisen.
- Funktionen aufrufen.
- Hilfe zu Funktionen anzeigen.
- die wichtigsten Variablentypen bestimmen.
- zwischen Variablentypen konvertieren.

2.3 Operatoren

Zunächst stellen wir fest, dass man die R-Konsole ganz banal als Taschenrechner benutzen kann:

```
1 + 4
```

```
## [1] 5
```

```
8 / 3
```

```
## [1] 2.666667
```

```
(2.45 + 3.5) * 7
```

```
## [1] 41.65
```

Die Zeichen +, −, ∗ usw. heißen in der Informatik Operatoren oder Infixe (weil sie immer zwischen zwei Werten stehen).

2.4 Variablen

Variablen funktionieren so, dass man einem *Wert* einen Namen gibt. Die Zuweisung folgt dabei dem Schema `NAME <- WERT`:

```
x <- 5
```

Nach einer erfolgreichen Variablenzuweisung gibt die Konsole *keine* Rückmeldung, sondern nur bei Fehlern.

`x` steht jetzt für die Zahl fünf. Mit dieser Variable können wir jetzt genauso rechnen wie mit einer Zahl:

```
x + 3
```

```
## [1] 8
```

Auch die Zuweisung von Variablen kann Rechenoperationen und andere Variablen enthalten:

```
y <- (x * 2) - 1
print(y)
```

```
## [1] 9
```

Der Befehl `print(y)` ist dabei ganz einfach die Anweisung an die Konsole, den Wert für `y` auszugeben. Das passiert zwar auch, wenn man nur `y` eingibt, aber `print(y)` (oder `print(x)`, `print(1 + 1)`, usw.) ist die formal korrekte Schreibweise.

Der Wert einer Variable kann auch verändert werden. Dafür weisen wir ihr einfach einen neuen Wert zu:

```
x <- 20
print(x)
```

```
## [1] 20
```

Eine Besonderheit ist, dass der alte Wert der Variable auch innerhalb der Zuweisung eines neuen Werts benutzt werden darf. Das kann in einem Script sehr praktisch sein. Wenn wir `x` also um 0,5 erhöhen wollen, sieht das so aus:

```
x <- x + 0.5
print(x)
```

```
## [1] 20.5
```

Dabei wird als Dezimaltrennzeichen ausschließlich der Punkt verwendet.

2.5 Konstanten

Manche benannten Werte sind schon in R eingebaut:

```
print(pi)
```

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

Diese Werte heißen üblicherweise "Konstanten" – allerdings lassen sie sich in R auch überschreiben!

```
pi <- 3
print(pi)
```

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

2.6 Funktionen

Mit `print()` haben wir schon unsere erste *Funktion* kennengelernt. R stellt uns eine Vielzahl von verschiedenen Funktionen zur Verfügung, und sie werden immer nach dem gleichen Schema benutzt: `FUNKTIONSNAME(PARAMETER)`.

Parameter (auf Englisch auch "arguments") sind die Werte, die als Input an die Funktion übergeben werden. Je nach Funktion können das auch mehrere Werte sein, die dann durch Kommas getrennt werden. So nimmt die Funktion `max()`, die den Maximalwert bestimmt, beliebig viele Zahlen als Parameter:

```
max(1, 2, 2, 5, 4, 3)
```

```
## [1] 5
```

Die Funktion `round()` hat als optionalen Parameter die Anzahl der Nachkommastellen, auf die gerundet werden soll. Wenn er nicht angegeben wird, nimmmt dieser Parameter immer den Wert 0 an:

```
round(4.567)
```

```
## [1] 5
```

Aber er lässt sich auch spezifizieren:

```
round(4.567, digits = 2)
```

```
## [1] 4.57
```

Dabei sind die folgenden Ausdrücke identisch:

```
round(4.567, digits = 2)
```

```
## [1] 4.57
```

```
round(4.567, 2)
```

```
## [1] 4.57
```

```
round(digits = 2, 4.567)
```

```
## [1] 4.57
```

Was Funktionen genau machen und welche Parameter sie dabei nehmen, ist in der R-Dokumentation sehr ausführlich (und auf den ersten Blick recht kompliziert) beschrieben. Ganz am Ende der Hilfeseite finden sich oft Beipsiele. Die Hilfe zu einer Funktion kann mit folgendem Befehl aufgerufen werden:

```
?max
```

Notiz am Rande: Auch die Infix-Operatoren +, −, ∗, usw. sind eigentlich nur verkürzte Schreibweisen von Funktionen. Mit “backticks” (‘) lassen sie sich in vollwertige Funktionen zurückverwandeln:

```
`+`(2, 2)
```

```
## [1] 4
```

2.7 Strings

R kann nicht nur mit Zahlen umgehen, sondern auch mit Text. Ein *String* ist eine Aneinanderreihung von Buchstaben, und wird mit einfachen oder doppelten Anführungszeichen umschlossen:

```
print("Hello, World!")
```

```
## [1] "Hello, World!"
```

Auch Variablen können Strings als Wert haben:

```
name <- "Hase"
```

Es gibt auch Funktionen, die Strings als Parameter nehmen. `paste` fügt Strings aneinander:

```
paste("Mein Name ist", name)
```

```
## [1] "Mein Name ist Hase"
```

2.8 Datentypen

Den *Typ* einer Variable oder eines Wertes bestimmen wir durch den Befehl `str()`:

```
str(name)
```

```
##  chr "Hase"
```

```
str(10)
```

```
##  num 10
```

Dabei steht `chr` („character“) für Strings und `num` („numeric“) für Zahlen.

Ein weiterer Variablentyp ist `logi` („logical“), der prinzipiell nur die Werte `TRUE` oder `FALSE` annehmen kann. Dieser Typ heißt auch Boolsche Variabel:

```
str(FALSE)
```

```
##  logi FALSE
```

Soweit es ein eindeutiges Ergebnis gibt, kann R mit den entsprechenden Befehlen Werte vom einen in den anderen Typ umwandeln:

```
as.numeric("1000")
```

```
## [1] 1000
```

```
as.character(x)
```

```
## [1] "20.5"
```

```
as.logical(0)
```

```
## [1] FALSE
```

Kann R einen Wert nicht umwandeln, dann kommt dabei `NA` raus (mit einer Warnung):

```
as.numeric("Hallo!")
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA
```

`NA` („not available/assigned“) ist dabei ein besonderer Wert, den jeder Variablentyp annehmen kann.

2.9 Aufgaben

2.9.1 Rechnen

Lösen Sie folgende Rechenaufgaben mit Hilfe von R:

- 4 plus 10
- 8 mal 12
- 4 minus 7
- 3 hoch 18
- 4,5 geteilt durch die Summe von 5 und 8
- Quadratwurzel aus 101
- Kubikwurzel aus 12

2.9.2 Variablen

Weisen Sie den Variablen a bis g folgende Werte zu:

a) TRUE

b) 2
c) Ihren Namen
d) Die Quadratwurzel aus b
e) 8 $\frac{1}{4}$
f) Das vierfache von e
g) Die aktuelle Uhrzeit mit Datum und Zeitzone (automatisch generiert)

2.9.3 Datentypen

Bestimmen Sie die Typen der Variablen a bis g.

Finden Sie je zwei Beispiele für die Umwandlung…

- von `numeric` zu `character`
- von `numeric` zu `logical`
- von `character` zu `logical`
- von `character` zu `numeric`
- von `logical` zu `character`
- von `logical` zu `numeric`
- von `character` zu `Date`
- von `Date` zu `numeric`

(`Date` ist kein eigentlicher Datentyp, aber erfüllt an dieser Stelle denselben Zweck.)

2.9.4 Swirl

Folgen Sie den Anleitungen, um Swirl zu installieren: https://swirlstats.com/students.html

Absolvieren Sie Lektion 1 („Basic Building Blocks“).

2.9.5 Recherche

Recherchieren Sie:

- Welche Funktion gibt den absoluten Wert einer Zahl aus? (z.B. -4 ergibt 4, 8 ergibt 8)
- Welche Konstanten sind in R „eingebaut“?
- Wie bestimmt man den „Rest“ einer Division? (z.B. 40 geteilt durch 7 hat den Rest 5)
- In der Statistik wird zwischen stetigen und diskreten Variablen unterschieden. Welche äquivalente Unterscheidung nimmt R vor?

2.9.6 Kniffliges

Lösen Sie die folgenden Probleme:

- Durch welchen Ausdruck lässt sich eine Zahl auf die nächste *gerade* Zahl runden? (z.B. 18,9 auf 18,0 oder 21,2 auf 22,0)
- Durch welchen Ausdruck lässt sich eine Zahl auf die nächste *halbe* Zahl abrunden? (z.B. 18,9 auf 18,5 oder 21,2 auf 21,0)
- Absolvieren Sie in die Lektion 8 („Logic“).
- Machen Sie sich mit der Funktion `xor()` vertraut. Finden Sie einen Ausdruck, der `xor()` simuliert, aber nur aus Infix-Operatoren besteht.
- Was bedeutet „strong“ bzw „weak typing“? Wie ist R hier einzuordnen?

- Was sind funktionale Programmiersprachen? Welche Eigenschaften von R sind funktional, welche nicht?
- Starten Sie den R Track in Excercism
- Richten Sie sich ein IDE *außer* RStudio für einen R Workflow ein.

3 Text: Anderson 2008

3.1 Lesetext

Anderson, Chris. 2008. *The End of Theory: The Data Deluge Makes the Scientific Method Obsolete.* URL: https://www.wired.com/2008/06/pb-theory/ (zugegriffen: 11. Juli 2017).

3.2 Fragen an den Text

1. Um welche Art von Text handelt es sich? Wer ist der Autor, und an wen wendet er sich?
2. Was ist das zentrale Anliegen des Texts? Welche Entwicklungen werden beschrieben?
3. Mit welchen Begriffen würden wir diese Phänomene heute beschreiben?
4. Aus heutiger Perspektive: Hatte der Autor recht? Warum / warum nicht?
5. In welchen Punkten stimmen Sie dem Autor zu? Wie würden Sie den Text problematisieren?