

# Analyse von Beziehungsmatrizen mit dem Statistikprogramm R

## Installieren von R

Home-page: <http://www.r-project.org/> (diese finden Sie auch schnell durch Eingabe von **R** in Google). Dann geht man unter: download: CRAN; man wählt den Server der ETHZ. Wählt Betriebssystem, dann „base“, dann „Download R 2.x.x for Windows“ (wenn man auf Windows installiert).

## Einlesen einer Beziehungsmatrix in R

Achten Sie darauf, dass Variablenamen keine Lehrschräge enthalten. Die Excel-Daten-Tabelle darf keine leere Spalten, Zeilen oder Zellen enthalten. Fehlende Daten sind durch NA zu ersetzen. Fügen Sie die Tabelle in den Editor ein und speichern Sie die Textdatei in ein Verzeichnis ab (möglichst ein einfaches, um Fehler zu vermeiden). Wir nehmen an, wir hätten die Textdatei unter „MeineDaten.txt“ im Verzeichnis „c:“ abgespeichert. Dann müssen wir in **R**, wenn wir den Daten in **R** den Namen „bez“ geben wollen, den folgenden Befehl eingeben:

```
bez = read.table("c : /MeineDaten.txt", header = T).  
enter
```

Achtung: **R** unterscheidet zwischen Gross- und Kleinbuchstaben. **header=T(RUE)** gibt an, dass die erste Zeile die Variablenamen enthält. Achten Sie darauf, dass das Querzeichen nach vorne schaut - und nicht wie üblich den Backslash verwenden. Die Tabelle kann in **R** durch den Befehl

```
bez
```

und enter angeschaut werden.

## Faktoren und Levels

Es verbessert die Lesbarkeit der Resultate, wenn man in R bei nominalskalierten Variablen mit Faktoren und benannten Ausprägungen arbeitet. *Faktoren* sind Variablen, die vom System als nominalskaliert betrachtet werden. *Levels* sind die Ausprägungen der nominalskalierten Variable. Will man eine Variable in einen Faktor umwandeln, z.B. die Variable „genre“ mit den Ausprägungen „mann“ und „frau“ so schreibt man fürs Dataframe „bez“:

```
bez$genre=factor(bez$genre, labels=(„mann“,„frau“))
```

Achtung: es wird dabei der kleinsten Zahl „mann“ zugeordnet und der zweitengrößten Zahl „frau“. Ist im Dataframe bisher jedoch den Mädchen die 1 zugeordnet, so muss man die Reihenfolge der Aufzählung umkehren: **labels=(„frau“,„mann“)**

## Berechnung von Korrelationen

Wir möchten die Pearson-Korrelation zwischen der Variable V1 und V2 berechnen. Dann schreiben wir

```
cor(bez$V1,bez$V2)
```

Für die Spearman-Korrelation schreiben wir:

```
cor(bez$V1,bez$V2,method= „spearman“)
```

Für Kendalls  $\tau$  schreiben wir:

```
cor(bez$V1,bez$V2,method= „kendall“)
```

## Installation des Paketes beznetz

Um das Paket „beznetz“ (für Beziehungsnetz) zu installieren, müssen Sie dieses in das Verzeichnis „libraries“ kopieren. Dieses befindet sich im Verzeichnis, in dem Sie **R** installiert haben, unter „R“ (aus dem Internet [sozio.logik.ch](http://sozio.logik.ch) herunterkopieren, entzippen und hineinkopieren. Achtung: oft wird beim Entzippen ein Verzeichnis gemacht, das den Namen der zip-Datei hat. Sie dürfen nicht dieses Verzeichnis in „libraries“ kopieren, sondern das Verzeichnis, das in diesem Falle ein Unterverzeichnis „beznetz“ ist). Das Paket wird eingebunden durch „`library(beznetz)`“. Sie müssen das Paket für jede **R**-Session neu einbinden. Die vom Paket angebotenen Befehle können mit

```
help(beznetz)
```

angeschaut werden - mit den entsprechenden Gebrauchsanleitungen.

## Hinzufügen der Spalten- oder Zeilensummen der Beziehungsmatrix

Sie können den Befehl:

```
ZeilenSumme(bez)
```

eingeben (s. für Details unter **R** „`help(ZeilenSumme)`“). Es wird die Zeilensumme unter dem Namen „ZeilenSUM“ der Datenmatrix bez beigefügt. Es kann auch ein anderer Namen, z.B. „sitzenNeben“ eingegeben werden durch

```
ZeilenSumme(bez, name=\sitzenNeben\)
```

Es werden automatisch die ersten *n* Spalten zeilenweise aufsummiert (*n* = Anzahl Personen des Netzes). Sollten nicht diese Spalten summiert werden, kann man die Spalten spezifizieren. Mit „`c(1:5, 17, 18, 23:25)`“ würden die 1. bis zur 5. die 17., die 18. und dann die 23. bis 25. Spalte verwendet.

```
SpaltenSumme(bez)
```

Es wird die Spaltensumme unter dem Namen „SpaltenSUM“ der Datenmatrix bez beigefügt (s. für Details im beznetz-Paket mit „`help(SpaltenSumme)`“). Es werden automatisch die ersten *n* Spalten spaltenweise aufsummiert (*n* = Anzahl Personen des Netzes). Sollten nicht diese Spalten summiert werden, kann man die Spalten spezifizieren. Mit „`c(1:12, 17)`“ würden die 1. bis zur 12. und die 17. verwendet. Es müssen *n* Spalten angegeben werden. Es kann auch ein anderer Name angegeben werden z.B. durch „`name=\beliebt\`“ - in diesem Falle würde der Name „beliebt“ der Spaltensummenspalte zugewiesen.

## Berechnung von Rangmitteln

Möchte man die Notendurchschnitte der Mädchen mit den Notendurchschnitten der Knaben vergleichen (oder die Noten anderer Untergruppen oder die Beliebtheit nach Untergruppen), so kann man folgenden Befehl eingeben (Noten sind ordinalskaliert!)

```
RangMittel(bez$genre, bez$durchschnitt)
```

wenn die Gruppierungsvariable „genre“ und die zu untersuchende Variable „durchschnitt“ heisst .

## Ordnen nach Gruppenvariable

Wollen Sie eine Beziehungsmatrix nach einer Gruppenvariable (z.B. Geschlecht oder Altersgruppen, etc.) ordnen, so können Sie

```
ordnen(bez, gruppen=bez$genre)
```

Dabei werden die ersten *n* Spalten als Teile der Beziehungsmatrix eingelesen (*n* = Anzahl Personen des Beziehungsnetzes = Anzahl Zeilen des Dataframes „bez“).

Sollten nicht die resten Spalten die Beziehungsmatrix enthalten, können z.B. die Spalten `c(5:10,15,7:18)` mit

```
ordnen(bez, c(5:10,15,7:18), gruppen=bez$genre)
```

eingelassen werden (5:10 bedeutet 5, 6, 7, 8, 9, 10). Es ist darauf zu achten, dass gleichviele Spalten angegeben werden, wie es Zeilen gibt.

## Aufteilen nach Gruppenvariable

Wollen wir eine Beziehungsmatrix nach einer Gruppenvariable (z.B. Geschlecht oder Altersgruppen, etc) aufteilen (es erscheinen  $k^2$  Beziehungsmatrizen,  $k$  = Anzahl Ausprägungen der Gruppenvariable, es werden die Intrateilgruppenbeziehungen sowie die Extragruppenbeziehungen dargestellt), so können Sie

```
aufteilen(bez, gruppen=bez$genre)
```

eingeben. Dabei werden die ersten  $n$  Spalten als Teile der Beziehungsmatrix eingelesen ( $n$  = Anzahl Personen des Beziehungsnetzes = Anzahl Zeilen des Dataframes „bez“). Siehe für Alternativen unter „Ordnen nach Gruppenvariablen“.

Möchten Sie auf eine der Teilmatrizen Zugriff haben so müssen Sie die Liste mit den Matrizen benennen, z.B. durch `b`:

```
b:=aufteilen(bez, gruppen=bez$genre)
```

Mit

```
b[[3]]
```

haben Sie Zugriff auf die dritte Matrix der Liste.

## Berechnung von Intra-Extra-Beziehungshäufigkeiten von Teilgruppen

Wollen wir die Extra- und Intra-Teilgruppenbeziehungen gemäss einer Gruppenvariable (z.B. Geschlecht oder Altersgruppen, etc) berechnen und ihr Verhältnis pro Gruppe so können Sie

```
IntraExtra(bez, gruppen=bez$genre)
```

eingeben. Dabei werden die ersten  $n$  Spalten als Teile der Beziehungsmatrix eingelesen ( $n$  = Anzahl Personen des Beziehungsnetzes = Anzahl Zeilen des Dataframes „bez“). Siehe für Alternativen unter „Ordnen nach Gruppenvariablen“.

## 0.1 Analyse nach Untergruppen

Möchte man z.B. Korrelationen nur für die Knaben berechnen oder die Korrelationen von Mädchen und Knaben vergleichen, so kann man die Daten mit einem Basis-R-Befehl wie folgt aufsplitten:

```
getrennt=split(bez, bez$genre)
```

`getrennt` ist eine Liste, welche die Daten getrennt nach Knaben und Mädchen enthält. Man erhält sie, indem man „`getrennt`“ schreibt und dann enter drückt. Mit `getrennt[[1]]` erhält man das Teildatenset des ersten Geschlechtes. Sind die Ausprägungen benannt (als levels eines Faktors, z.B. mit „frau“, so kann man die Daten der Mädchen durch „`getrennt$frau`“ erhalten. Man kann einer Teildatenbasis einen Namen geben, z.B. durch `maedchen=getrennt[[2]]` und dann mit dem Datenset `maedchen` arbeiten wie mit einer normalen Dataframe (z.B. Korrelationen berechnen).