

## Dokumentation

### Erzeugung der Matrizen

Zunächst soll eine Funktion implementiert werden, welche eine  $m \times n$ -Matrix erzeugt und diese mit Nullen und Einsen füllt. Die Wahrscheinlichkeit der 1sen beträgt  $p$  und analog die W'keit von 0en  $1 - p$ . Jeder Matrizenpunkt  $X_{j,k}$ , mit  $1 \leq j \leq m$  und  $1 \leq k \leq n$ , entspricht einer  $\text{Ber}(p)$ -verteilten Zufallsvariable.

Hierfür wurde eine Matrix mit der Funktion *matrix* erzeugt, welche einen Vektor, sowie die Zeilen- und Spaltengröße übergeben bekommt. Der Vektor wurde, da  $\text{Ber}(p)$ -verteilt, mit *rbinom* erzeugt, welches die Größe  $(m \times n)$ , 1 (nur Nullen und Einsen) und die W'keit  $p \in [0, 1]$  übergeben bekommt.

```
Zuordnung <- function(m,n,p) return(matrix(rbinom(m*n,1,p),n,m))
```

### Links-Rechts-Verbindung

Als nächstes sollen Matrizen wie oben beschrieben auf eine Links-Rechts-Verbindung geprüft werden können. Als Rückgabe soll *TRUE* für eine gefundenen Verbindung und *FALSE* andernfalls zurückgeliefert werden. Eine Links-Rechts-Verbindung besteht, wenn von links nach rechts über die Matrix "gewandert" werden kann, indem man nur nach oben, links, rechts oder unten geht, also nur zu benachbarte Punkten, und dabei auf Einsen trifft.

Es wurde eine Tiefensuche implementiert, welche schon auf der zu untersuchenden Matrix arbeitet und schon besuchte Punkte mit einer 0 markiert. Das ist praktisch, da die anderen Ausgangsnutzen auch nicht betretbar sind. Aus Effizienzgründen wurde mit Pointern gearbeitet, um die Matrix nicht immer kopieren zu müssen. Die Funktion *newPointer* wurde nicht selbst implementiert ([http://www.stat.berkeley.edu/~paciorek/computingTips/Pointers\\_passing\\_reference\\_.html](http://www.stat.berkeley.edu/~paciorek/computingTips/Pointers_passing_reference_.html)).

Die Funktion *Verbunden* erzeugt einen Pointer auf die Matrix und lässt dann die Tiefensuche auf alle linken Einträge der Matrix laufen, die eine 1 beinhalten. Wird dabei ein Weg von links nach rechts gefunden, gibt *Verbunden* *TRUE* und sonst *FALSE* zurück.

Die Tiefensuche schaut zunächst, ob sie schon rechts angekommen ist. Wenn ja, dann ist die Suche erfolgreich gewesen. Andernfalls wird in folgender Reihenfolge geschaut, wo der Weg rekursiv weiter gehen kann: rechts, links, unten, oben. Kann ein Weg nicht weiter verfolgt werden, wird Backtracking angewandt und an der letzten Abzweigung fortgesetzt. Ist auch der letzte mögliche Punkt nicht mehr weiterverfolgbar, so wird *FALSE* zurückgegeben.

```
Verbunden <- function(a) {
  besucht <- newPointer(a)
  for(i in 1:dim(besucht$value)[1]) {
    if(besucht$value[i,1]) {
```

```

        if (Tiefensuche(besucht, c(i, 1)))
            return(TRUE)
    }
}
return(FALSE)
}

```

## Relative Häufigkeit

Zuletzt soll eine Funktion *RelativeHaeufigkeit* implementiert werden, welche den Mittelwert mit *mean* bei *b*-facher Durchführung (*replicate*) beim Überprüfen auf LR-Verbundenheit bestimmt. Hierfür werden die Matrizengröße ( $m \times n$ ) und die Wahrscheinlichkeit für die Ber(p)-Verteilung benötigt.

```

RelativeHaeufigkeit <- function(m, n, p, b) {
  return(mean(replicate(b, Verbunden(Zuordnung(m, n, p)))))
}

```

Im Folgenden wurden Plots erzeugt, um die relative Häufigkeit bei unterschiedlichen Matrizengrößen zu visualisieren. Für *b* wurde 1000 gewählt, um eine einigermaßen gute Approximation zu bekommen. Für *p* wurde die Wahrscheinlichkeit  $\frac{1}{10}$ ,  $\frac{2}{10}$  bis  $\frac{10}{10}$  gewählt. Dafür wurde mit *Werte1bis10* ein Vektor erzeugt.

```

Werte1bis10 <- function(nm, b) {
  p <- c(1:10)/10
  ergebnis <- c(1:10)
  for(i in c(1:10))
    ergebnis[i] <- RelativeHaeufigkeit(nm, nm, p[i], b)

  return(ergebnis)
}

```

**n=m=5**

```
plot(Werte1bis10(5,1000),c(1:10)/10, ylab="Wahrscheinlichkeit p",  
xlab="Relative Haeufigkeit P(A)")
```

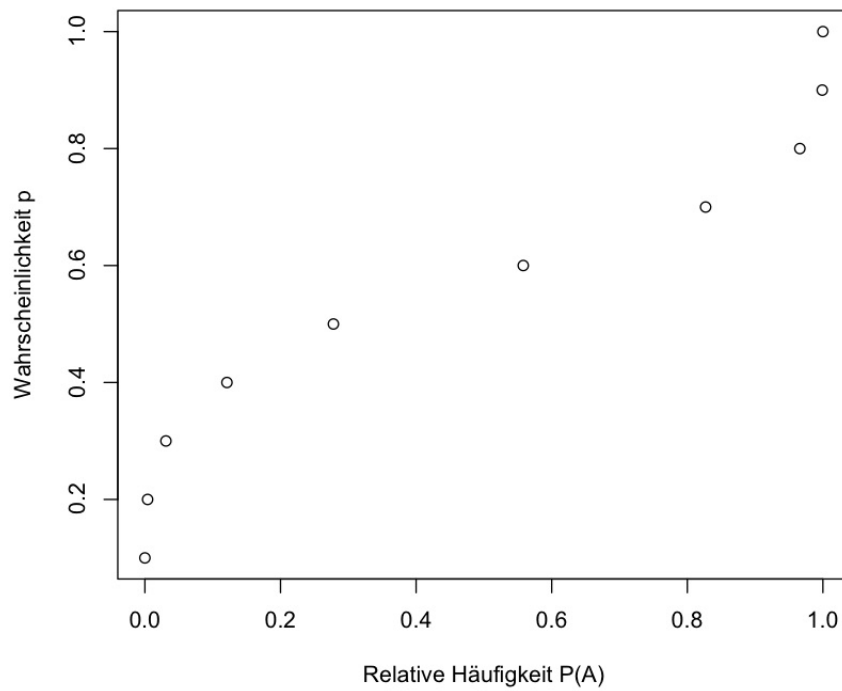


Abbildung 1: n=m=5

**n=m=10**

```
plot(Werte1bis10(10,1000),c(1:10)/10, ylab="Wahrscheinlichkeit p",  
xlab="Relative Haeufigkeit P(A)")
```

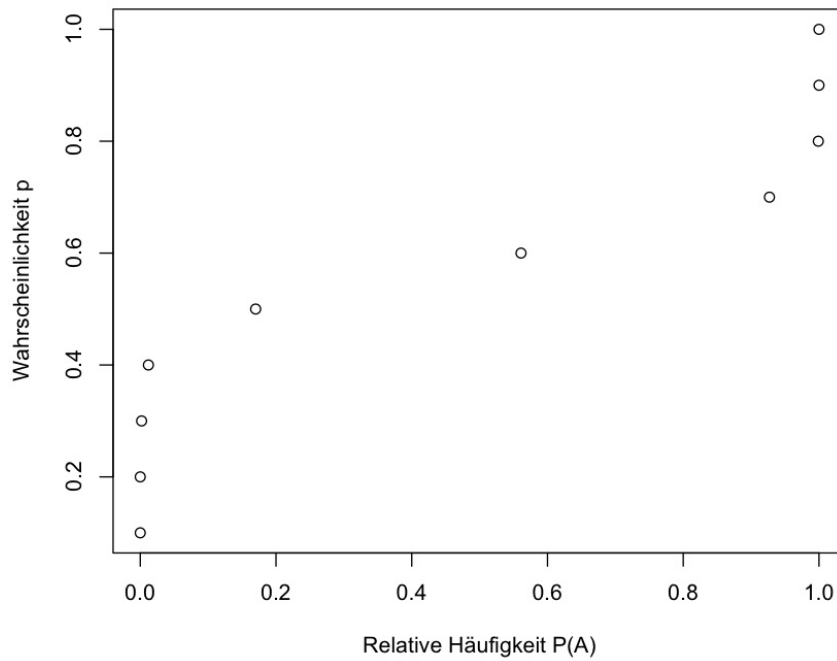


Abbildung 2: n=m=10

**n=m=20**

```
plot(Werte1bis10(20,1000),c(1:10)/10, ylab="Wahrscheinlichkeit p",  
xlab="Relative Haeufigkeit P(A)")
```

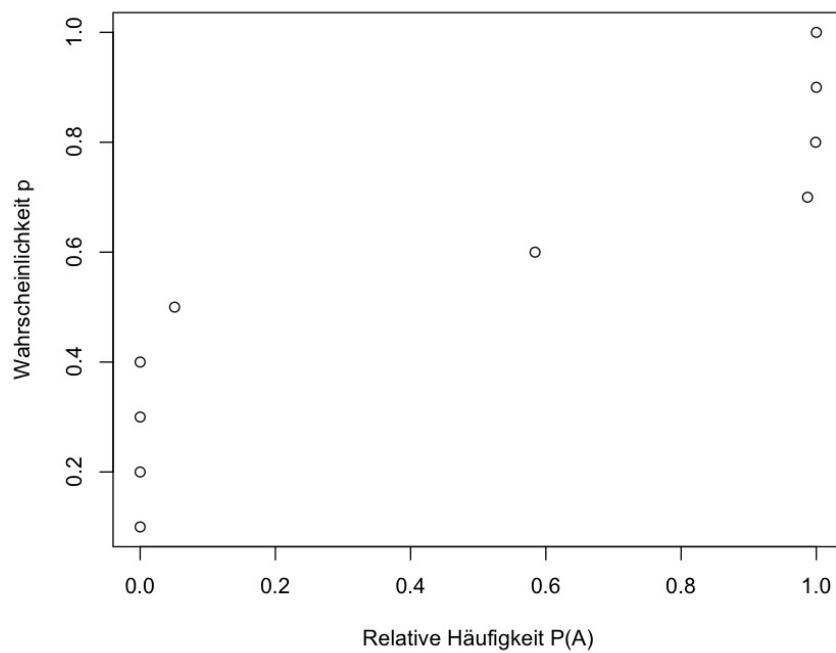


Abbildung 3: n=m=20

Es fällt auf, dass sich die Werte bei größeren Matrizen an den Randbereichen sammeln, also um die relative Häufigkeit 0 und 1. Außerdem ist bei einer Wahrscheinlichkeit unter 0,5 die relative Häufigkeit eher niedrig und darüber Sprungartig groß. Ab  $p \approx 0,7$  und  $n=m$  groß genug geht die relative Häufigkeit gegen 1.