

**Herzlich willkommen zur
Vorlesung Statistik für Informatiker, SS 2021**

Matthias Birkner

Kapitel 0

Auftakt

0.1 Organisatorisches

Homepage der Vorlesung:

<http://www.staff.uni-mainz.de/birkner/StatInfo21/>

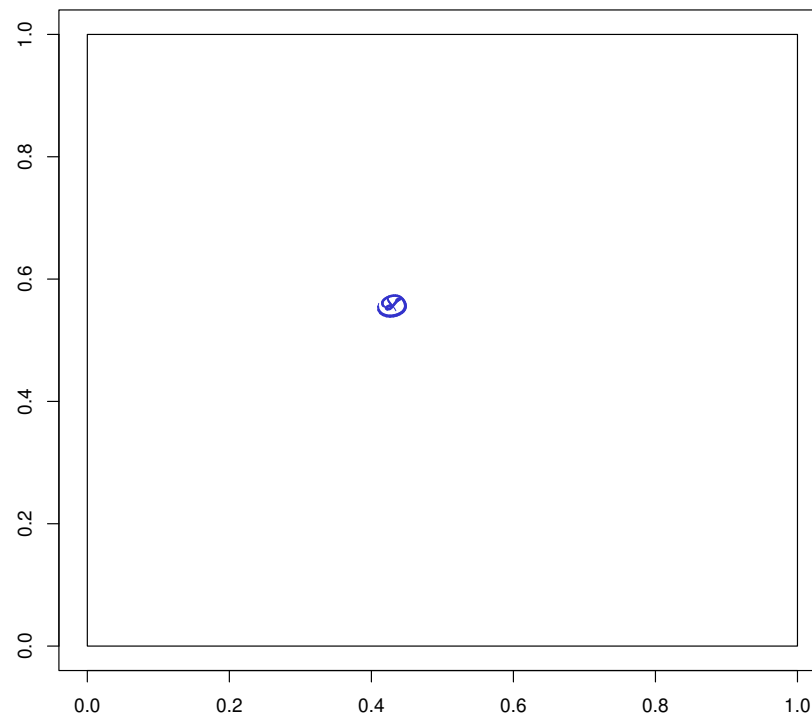
(dort werden die Folien/Notizen, benutzter R-Code/Datensätze, etc. stehen)

Übungen (betreut von Iulia Dahmer), Moodle:

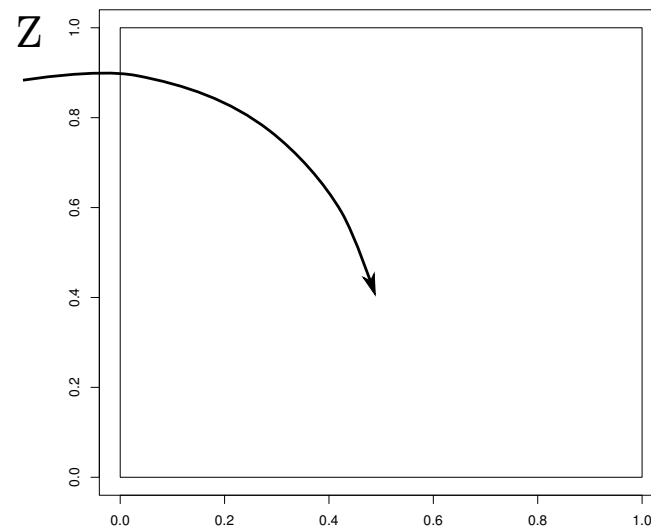
<https://lms.uni-mainz.de/moodle/course/view.php?id=45924>

0.2 Ein Startbeispiel

Sei $Z = (X, Y)$ ein „rein zufällig“ gewählter Punkt im Einheitsquadrat $S = \{(x, y) : 0 \leq x, y \leq 1\}$ (geschrieben in kartesischen Koordinaten).



Einheitsquadrat S

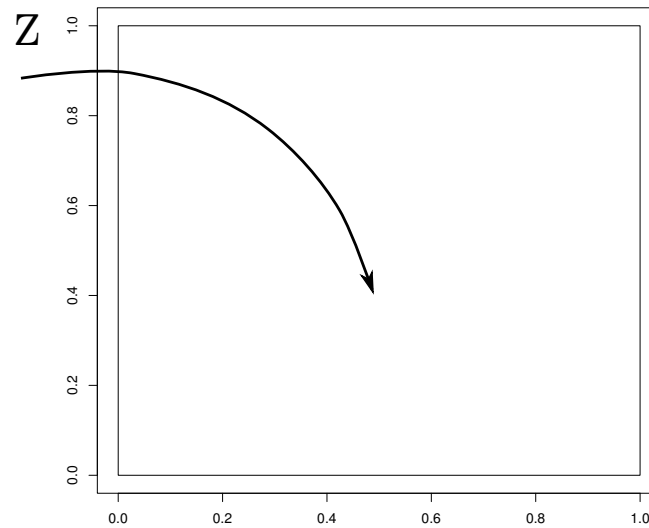


Für die praktische Implementierung stellen wir uns etwa vor, dass S in (sehr kleine) disjunkte Quadrate („Pixel“) zerlegt wird, und dass wir unter allen möglichen Pixeln eines wählen, wobei jedes dieselbe Chance hat, gezogen zu werden.

Eine Möglichkeit, dies in R zu implementieren, wäre

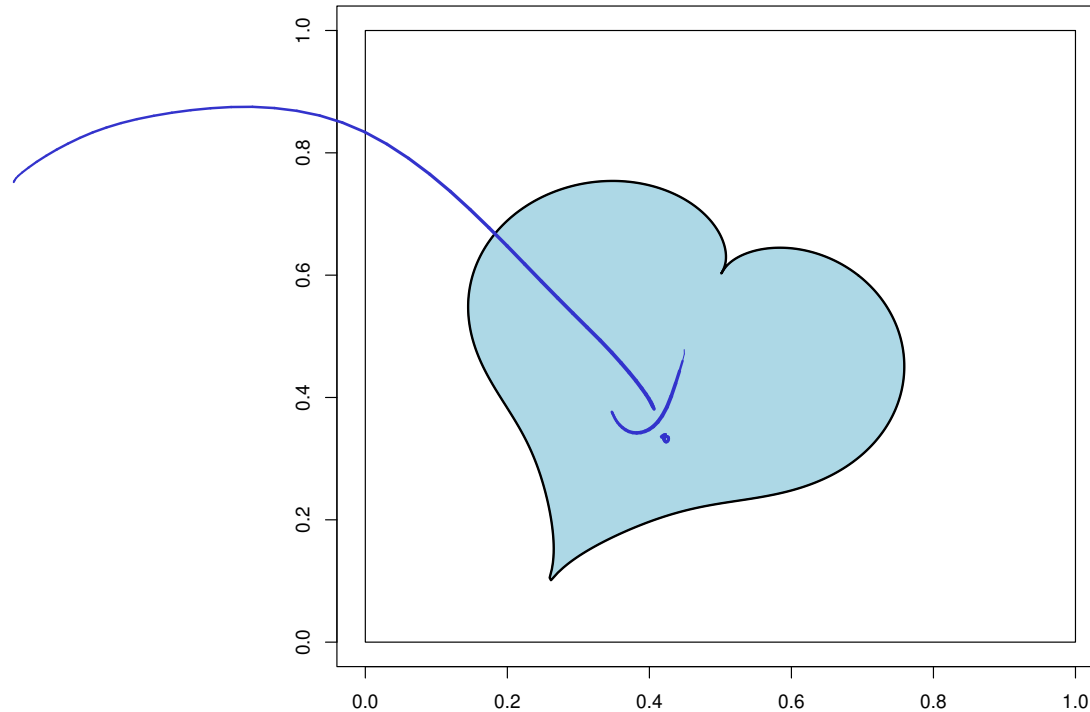
```
Z <- c(runif(1),runif(1))
```

Hierbei generiert der Befehl `runif(1)` eine (Pseudo-)Zufallszahl im Einheitsintervall $[0, 1]$.



Wir nennen Z eine Zufallsvariable,
die möglichen Werte $S (= \{(x,y) : 0 \leq x,y \leq 1\})$
bilden den Wertebereich.

Sei $B \subset S$ eine gewisse Teilmenge.



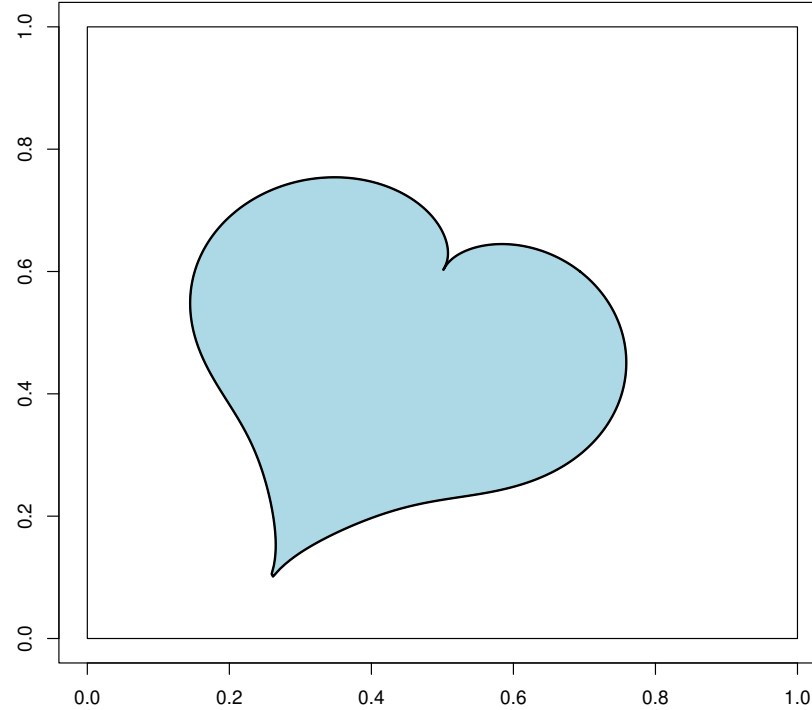
Einheitsquadrat und Teilmenge B

Wir können dann das Ereignis

$$\{Z \in B\}$$

(„ Z nimmt einen Wert in B an“)

betrachten.



Je nach Ausgang des zufälligen Experiments (für das Computerexperiment hängt dies vom internen Zustand des Pseudo-Zufallsgenerators und damit implizit vom gewählten “random seed” ab) wird Z in B liegen oder nicht, d.h.

das Ereignis $\{Z \in B\}$ tritt ein oder nicht.

Wie wahrscheinlich ist das Ereignis $\{Z \in B\}$?

$$P(\{Z \in B\}) = \frac{\text{Anzahl Pixel in } B}{\text{Anzahl Pixel in } S} = \frac{\text{Fläche von } B}{\text{Fläche von } S}$$

"1"

$$= \text{Fläche von } B$$

(engl. "probability" ---)

Beschreibung mittels Indikatorfunktion

$B \subset S$, die Indikatorfunktion von B ist

$$\mathbb{1}_B(z) = \begin{cases} 1, & z \in B \\ 0, & z \notin B \end{cases}$$

Bilde eine neue/weitere $W := \mathbb{1}_B(Z)$

(Wertebereich von W ist $\{0, 1\}$)

(W heißt auch eine Indikatorvariable)

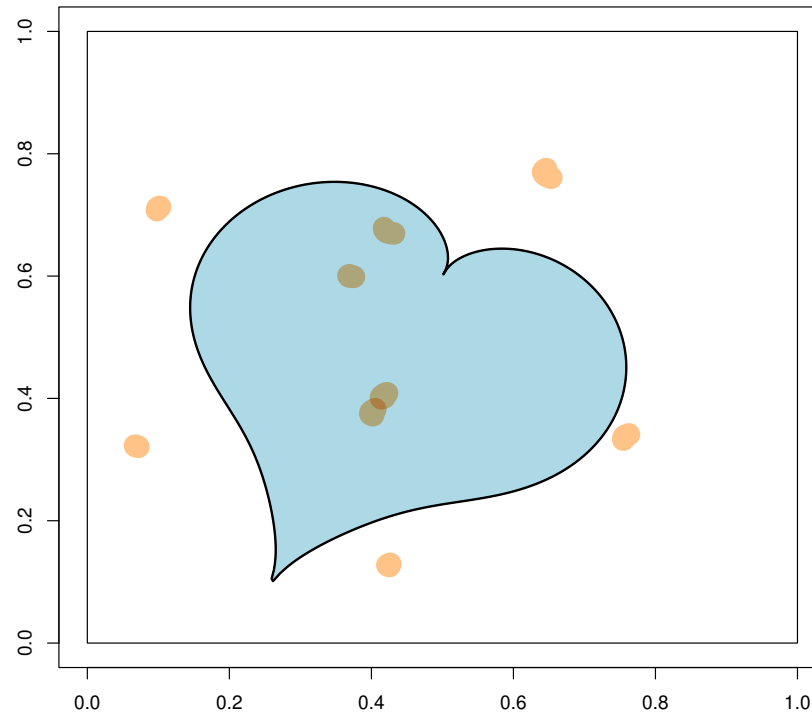
$$\{z \in B\} = \{W = 1\}, \quad \{W = 0\} = \{z \in B^c\}$$

$$\begin{aligned} & \text{(mit } B^c = S \setminus B \\ & \quad = \{z \in S : z \notin B\}) \end{aligned}$$

Insbesondere:

$$P(\{W = 1\}) = P(\{z \in B\}) = \frac{\text{Fläche von } B}{1}$$

0.3 Anwendung: eine einfache Monte Carlo-Methode zur Integration



Sei $p = P(\{Z \in B\}) = \text{Fläche von } B$. Wir können den Zufall verwenden, um p (wenigstens approximativ) zu bestimmen.

Stellen wir uns vor, wir wiederholen obiges Zufallsexperiment n -mal, wobei der Zufall „jedes mal neu wirkt“.

Seien $z_1, z_2, \dots, z_n$ die Ergebnisse dieser n Experimente
(Jargon: „unabhängige Kopien von z “),

$$W_i := \mathbb{1}_B(z_i), \quad i = 1, 2, \dots, n$$

$$\tilde{M}_n := \sum_{i=1}^n W_i \quad (\text{Zählt, wieviele der } n \text{ zufälligen Punkte in } B \text{ liegen})$$

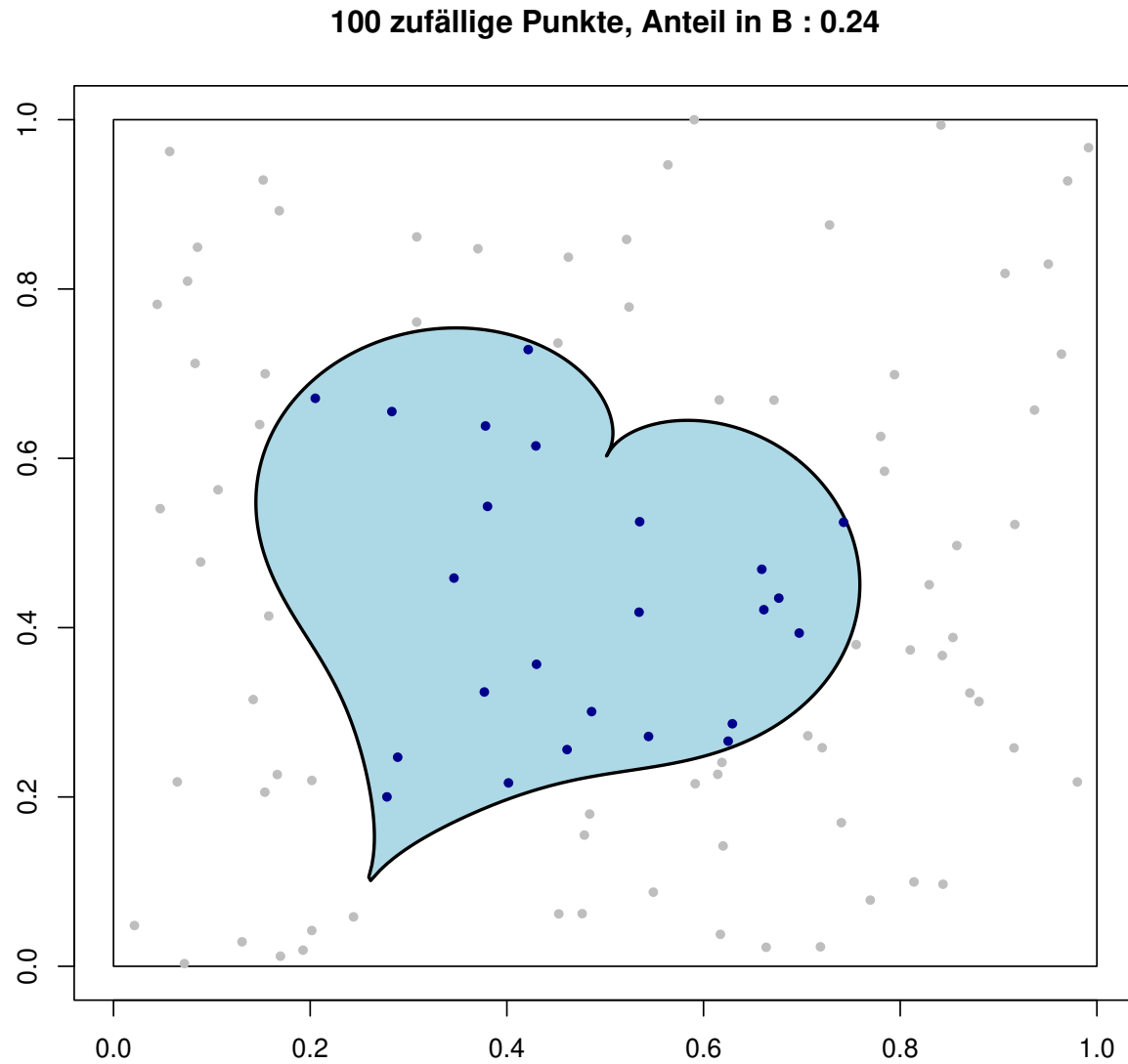
$$M_n := \frac{1}{n} \sum_{i=1}^n W_i \quad (= \frac{1}{n} \tilde{M}_n) \quad \text{„empirischer Anteil“ der Punkte in } B$$

ist ein sehr plausibler „Schätzwert“ für p

Wertebereich von M_n ist

$$\left\{ 0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1 \right\}.$$

$$\begin{aligned} & (= P(\{z \in B\})) \\ & = \text{Fläche von } B \end{aligned}$$



Visualisierung eines solchen Versuchs für $n = 100$.

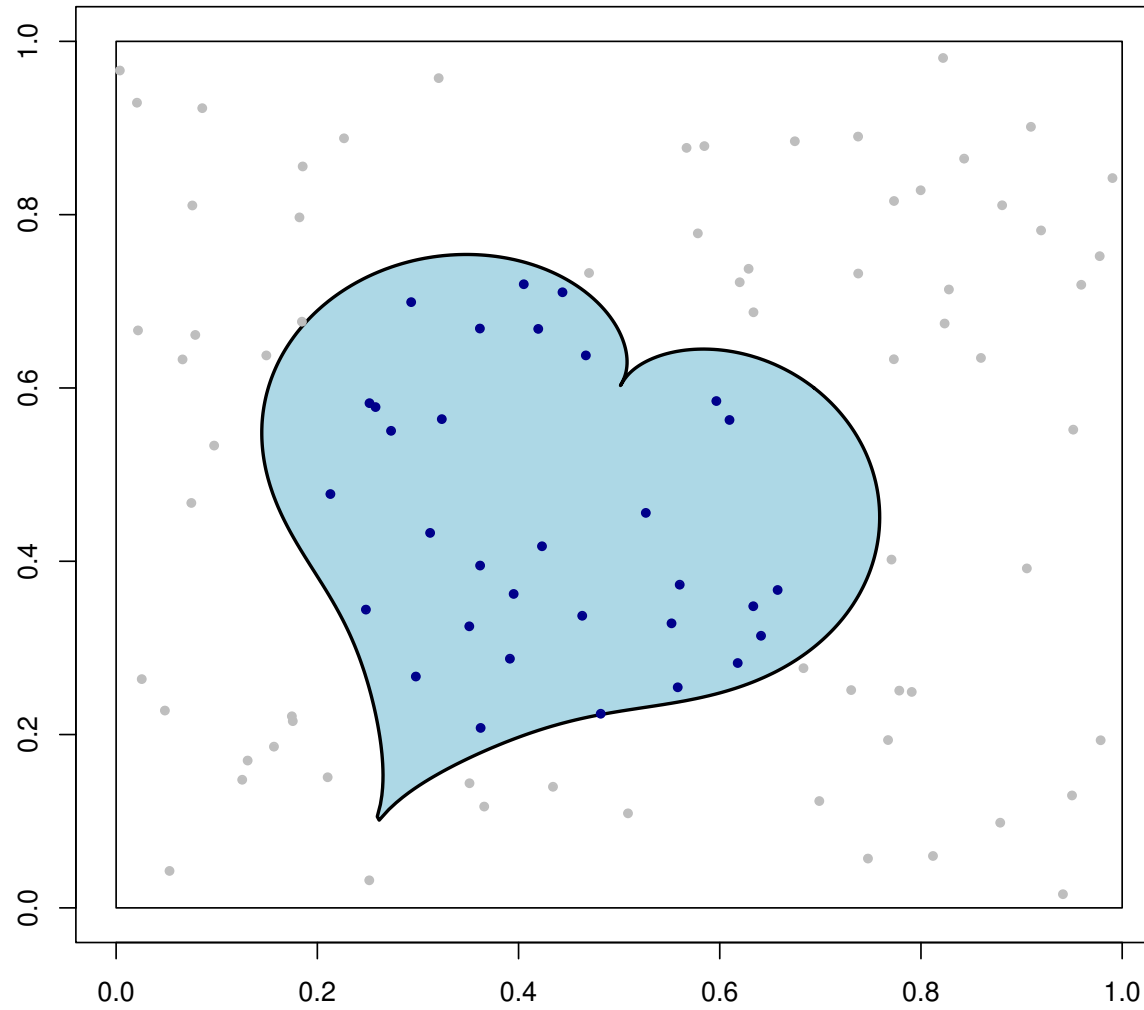
Bemerkung.

Man kann dies (nämlich die Simulation von M_n und Ausgabe des berechneten Werts) als ein einfaches Beispiel eines sogenannten Monte Carlo-Algorithmus betrachten:

Das Verfahren bestimmt zwar nicht genau den Wert von p , wir werden aber quantifizieren können, wie (un-)wahrscheinlich es ist, dass es um mehr als ein vorgegebenes ε „daneben liegt“.

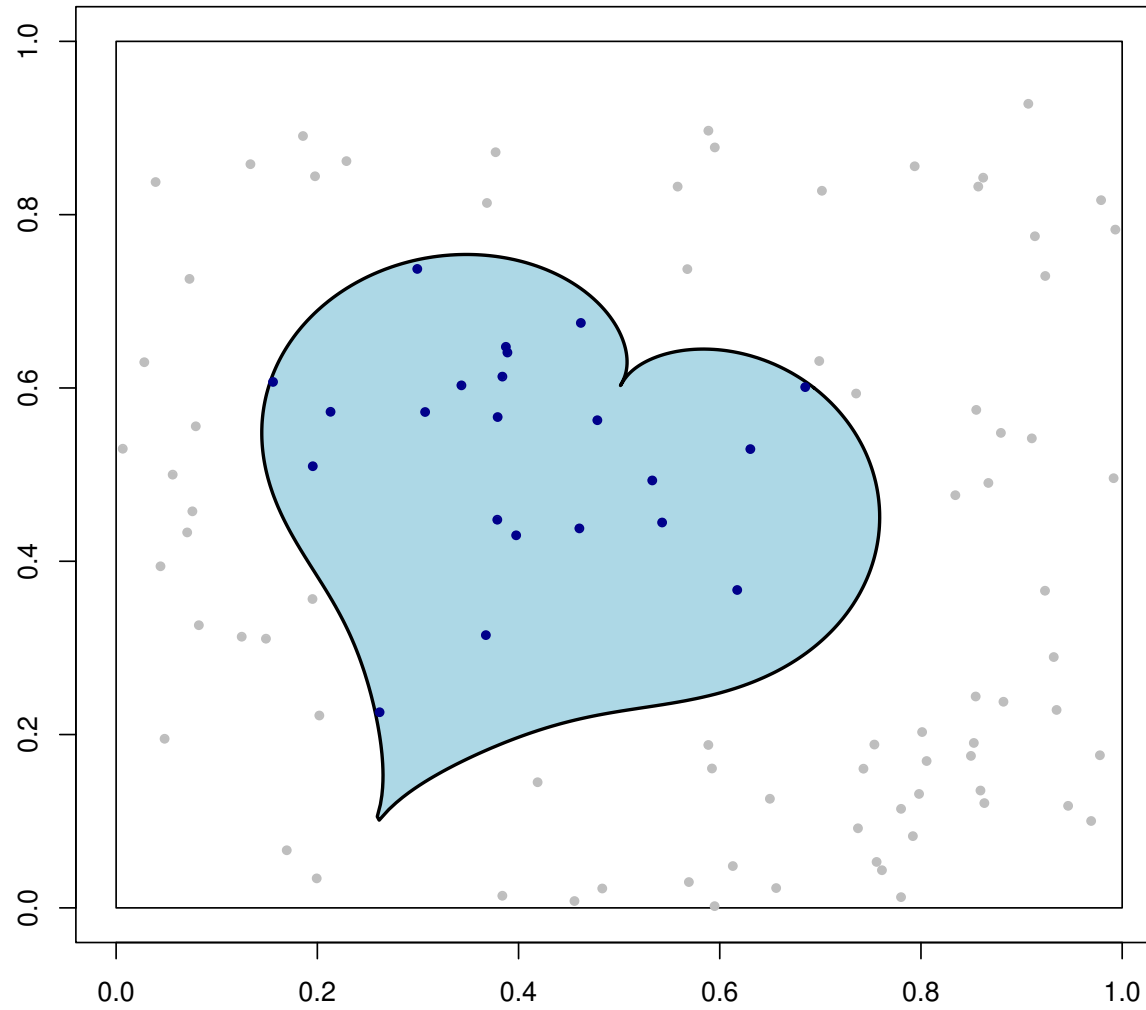
Eine Wiederholung mit $n = 100$ zufälligen Punkten in $[0, 1]^2$

100 zufällige Punkte, Anteil in B : 0.32



Noch eine Wiederholung mit $n = 100$ zufälligen Punkten in $[0, 1]^2$

100 zufällige Punkte, Anteil in B : 0.22



Verteilung von $\tilde{M}_n$

$$\tilde{M}_n = \sum_{i=1}^n \mathbb{1}_B(z_i)$$

(zählt, wieviele der n zufälligen Punkte in B landen)

(Wertebereich $\{0, 1, \dots, n\}$)

($p = P(\{z \in B\})$)

$$P(\{\tilde{M}_n = k\}) = \binom{n}{k} p^k (1-p)^{n-k}$$

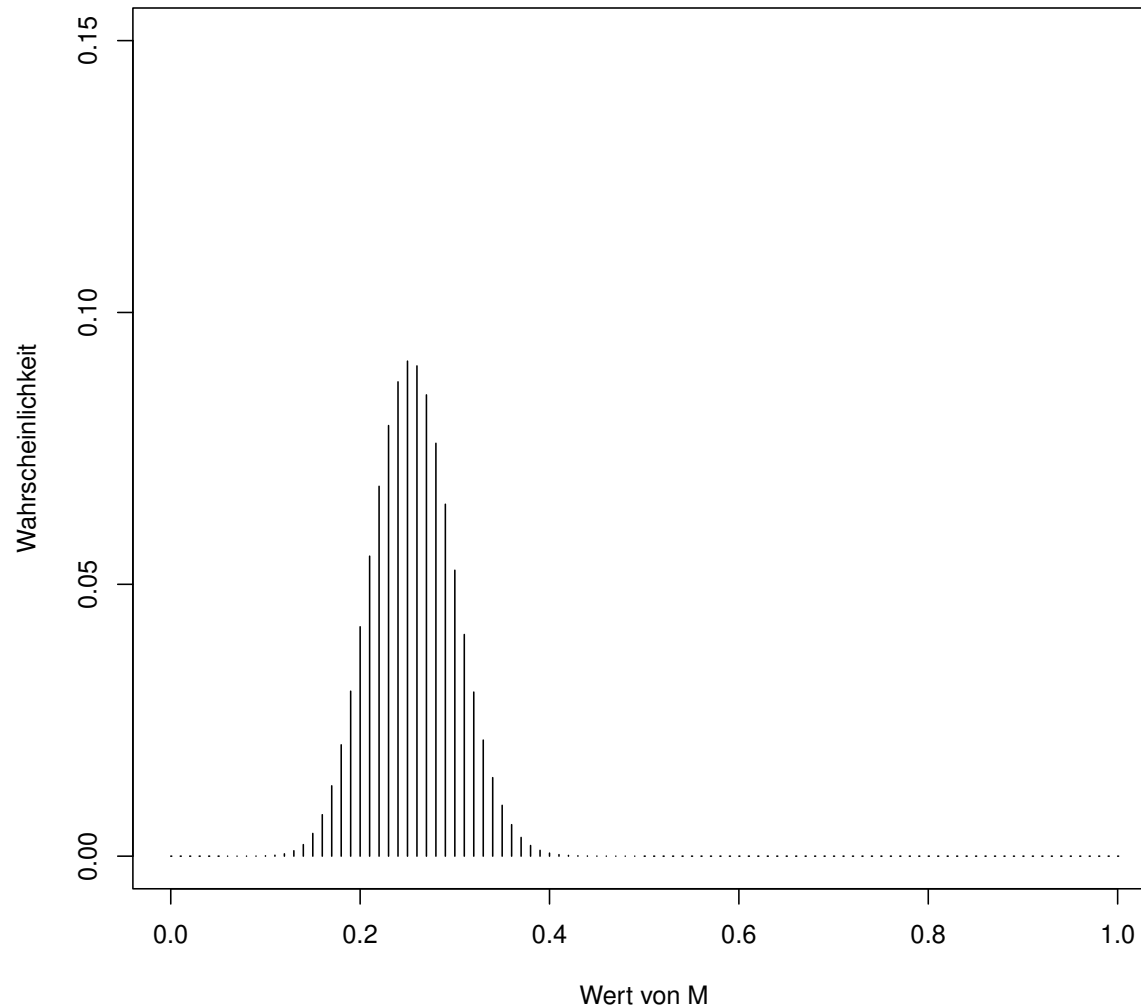
$$= \frac{n!}{k!(n-k)!} = \frac{n(n-1) \dots (n-k+1)}{k \cdot (k-1) \dots 2 \cdot 1}$$

($\tilde{M}_n$ ist binomialverteilt mit Parametern n und p)

Ant. Möglichkeiten,
 k "Treffer" auf
 n Würfe zu verteilen

$$P(\{z_1 \in B, z_2 \in B, \dots, z_k \in B, \\ z_{k+1} \notin B, z_{k+2} \notin B, \dots, z_n \notin B\})$$

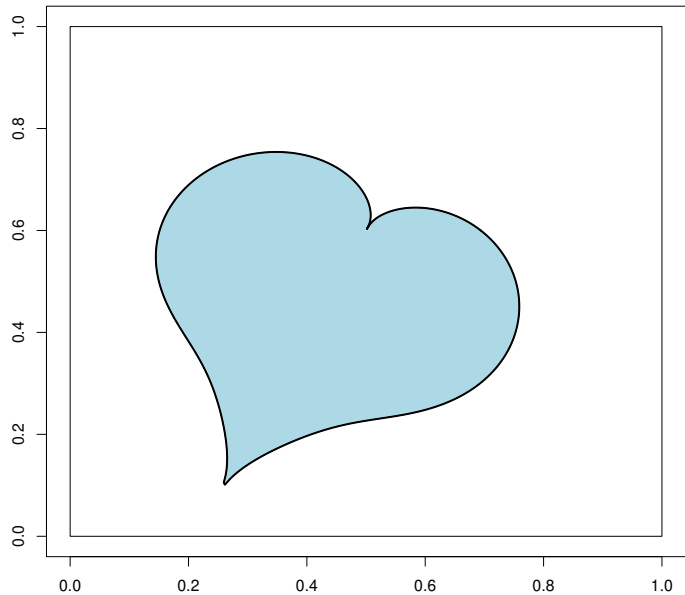
Damit ergibt sich für die Verteilungsgewichte von M_{100} folgendes (Balken-)Diagramm



(an die Stelle k/n zeichnen wir einen Balken der Höhe $P(\{M_n = \frac{k}{n}\})$)

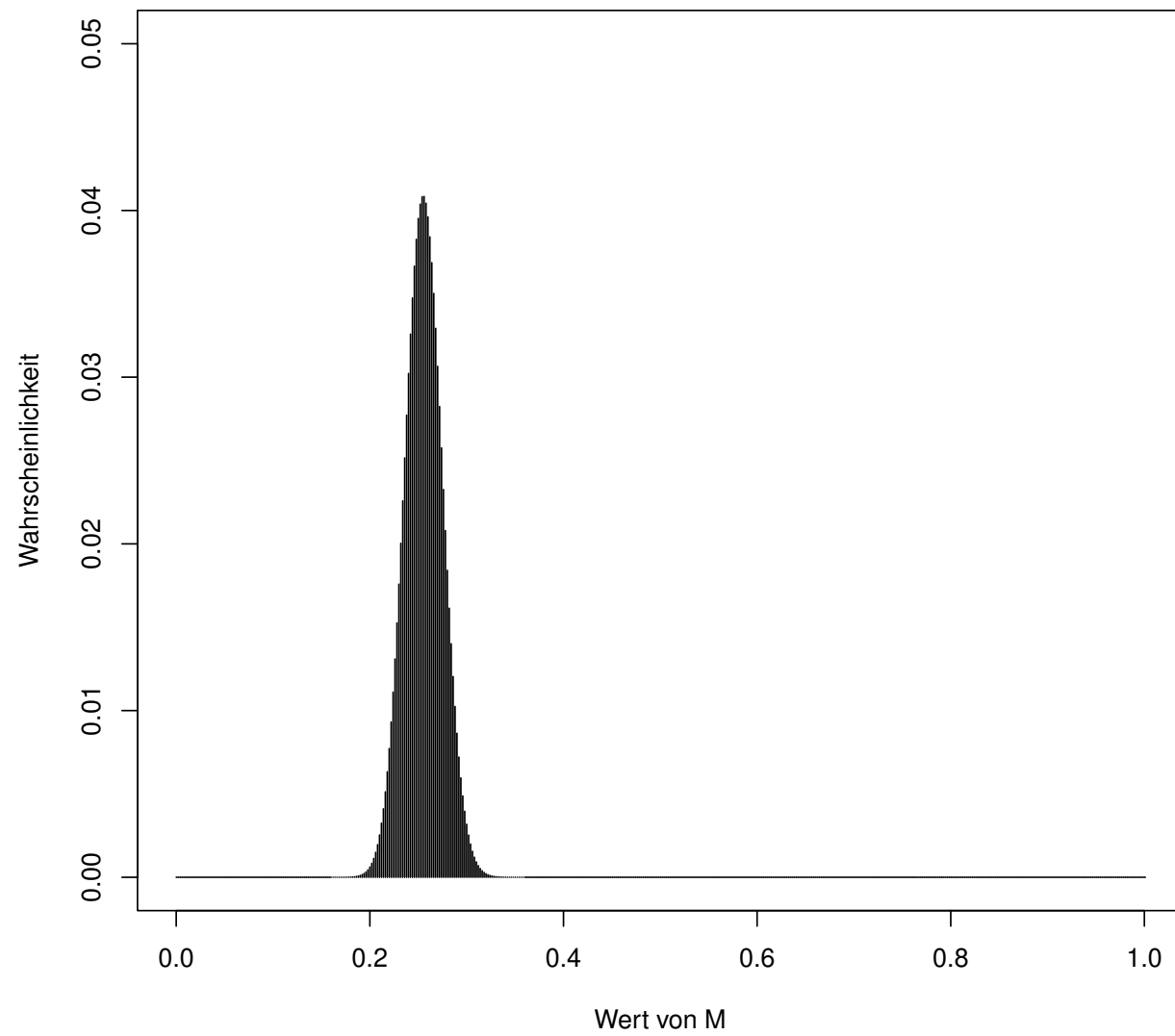
Bemerkung. Tatsächlich ist der Wert von p in diesem Beispiel

$$p = \int_{-\pi}^{\pi} \frac{1}{98} \left(2 - 2 \sin(w) + \frac{\sin(w) \sqrt{|\cos(w)|}}{\sin(w) + 7/5} \right)^2 dw \approx 0.25557221...$$

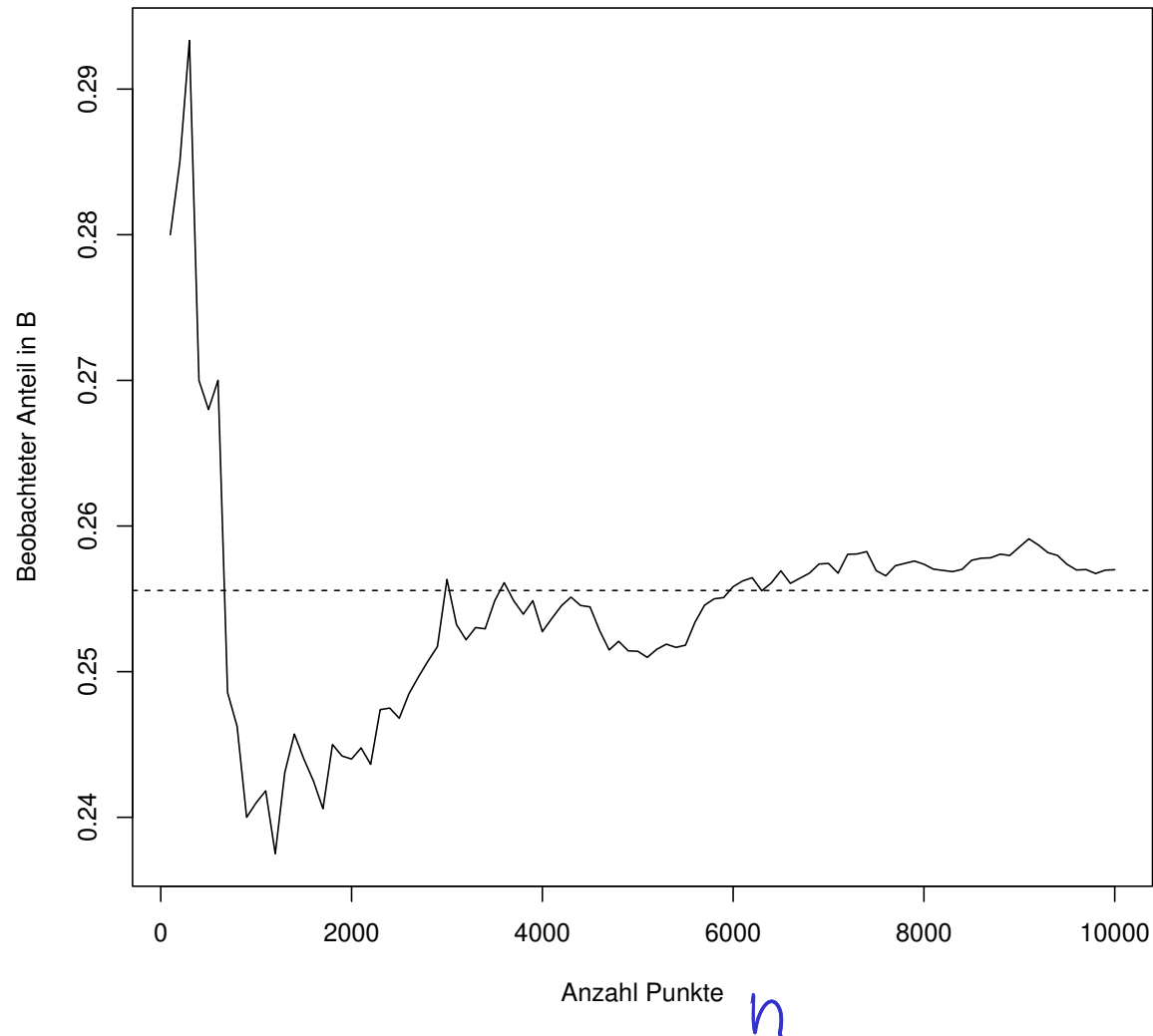


Durch Erhöhung von n können wir die Genauigkeit der Approximation erhöhen.

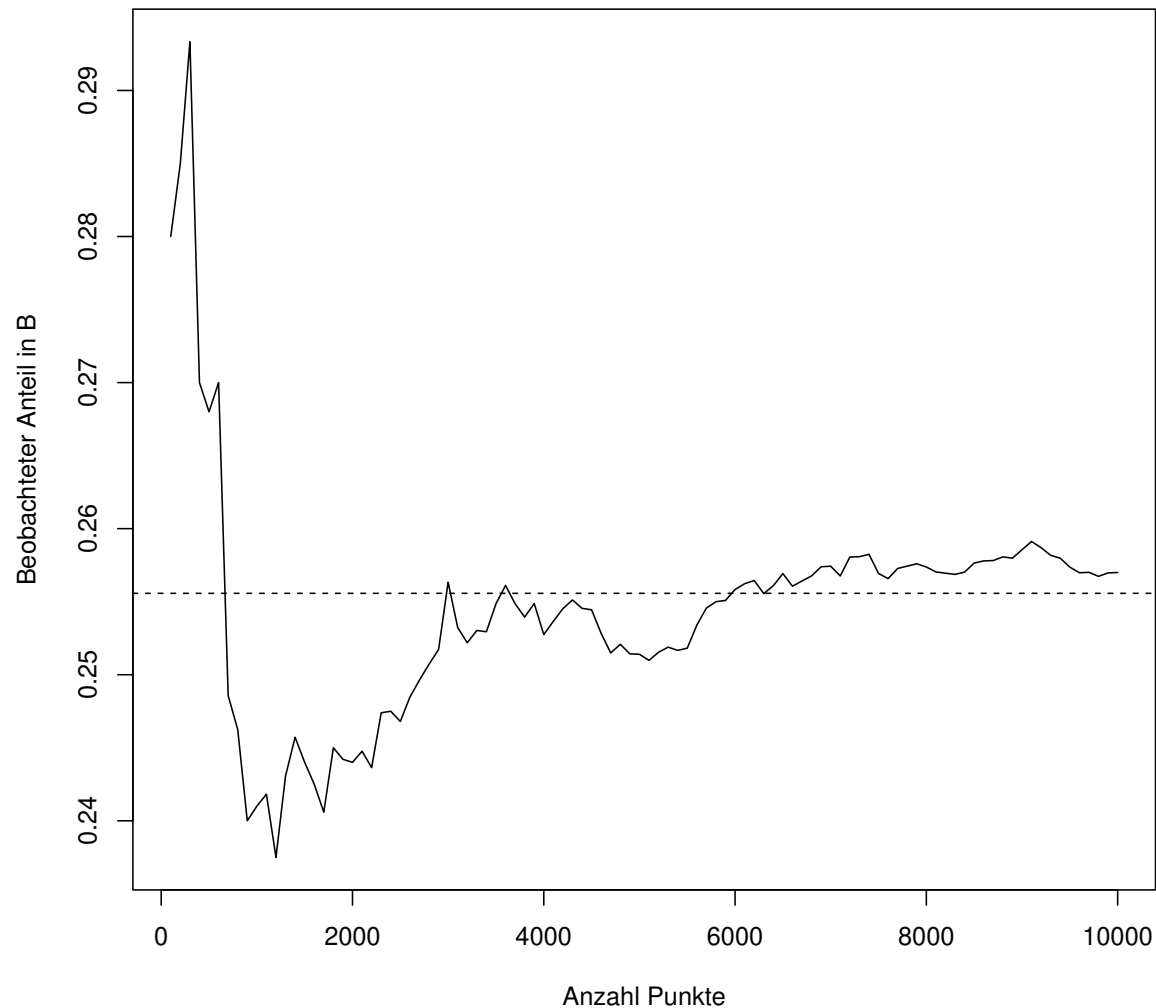
Wir werden dies später quantifizieren können, für den Moment betrachten wir das Histogramm der Verteilungsgewichte von M_{500} .



Betrachten wir zum Abschluss (für eine Folge von $Z_1, Z_2, \dots, Z_{10000}$) die Folge der M_n als Funktion von n :



(die gestrichelte Linie ist der Wert p)



Wir werden im Lauf der Vorlesung sehen: Für $n \rightarrow \infty$ konvergiert M_n (in geeignetem Sinne) gegen p . Dies folgt aus dem sogenannten *Gesetz der großen Zahlen*.

0.4 Erste Schritte mit R

Was ist R?

- R ist eine für die Statistik und für stochastische Simulation entwickelte Programmiersprache, zudem sind viele statistische Standardverfahren bereits in R implementiert oder als Zusatzpaket verfügbar.
- R hat eine sehr aktive Benutzer- und Entwicklergemeinschaft, die nahezu alle Bereiche der Statistik und viele Anwendungsbereiche (z.B. Populationsgenetik, Finanzmathematik) überdeckt.
- R ist frei verfügbar unter der GNU general public license, für (nahezu) alle Rechnerarchitekturen erhältlich:

<http://www.r-project.org/>

Literatur zu R

„Standardreferenz“:

W.N. Venables et al, *An Introduction to R*,
<http://cran.r-project.org/manuals.html>

Zur ersten Einführung auch die Hinweise zu R auf der Vorlesungs-Homepage.

Weiterführend (z.T. wesentlich über den Stoff der Vorlesung hinaus):

- Günther Sawitzki, *Einführung in R*, <http://sintro.r-forge.r-project.org/>
- William N. Venables, Brian D. Ripley, *Modern applied statistics with S* („Standardwerk“, UB Lehrbuchsammlung)
- Lothar Sachs and Jürgen Hedderich, *Angewandte Statistik – Methodensammlung mit R* (E-Book, UB)
- Christine Duller, *Einführung in die nichtparametrische Statistik mit SAS und R : ein anwendungsorientiertes Lehr- und Arbeitsbuch* (E-Book, UB)
- Helge Toutenburg, Christian Heumann, *Deskriptive Statistik : Eine Einführung in Methoden und Anwendungen mit R und SPSS* (E-Book, UB)
- Uwe Ligges, *Programmieren mit R* (E-Book, UB)

R installieren, starten, anhalten

Installation: Windows, Mac OS: Binaries von <http://www.r-project.org/> (siehe Link Download, Packages, **CRAN** dort)

Linux: Für die meisten Distributionen gibt es fertige Pakete (entweder bei **CRAN** oder im Repository der Distribution)

Bei Fragen oder Problemen: Übungsgruppen

R starten: Windows, Mac OS: Icon (ggf. aus Menu) anklicken,
Linux/Unix: > R auf einer Konsole (oder z.B. mit ESS aus Emacs heraus).

R beenden: q() (fragt, ob Daten gespeichert werden sollen)

laufende Rechnungen unterbrechen: CTRL-C

Hinweis: **RStudio** <https://www.rstudio.com/> bietet eine integrierte Entwicklungsumgebung für R [für Privatanwender frei verfügbar].

Einige einfache (und wichtige) R-Befehle

Mathematische Operatoren
und Funktionen:

`+`, `-`, `*`, `/`, `^`, `exp`, `sin`, `log`, etc.

Hilfe aufrufen:

`help(Befehl)` oder `?Befehl`

Online-Hilfe starten:

`help.start()`

Variable einen Wert zuweisen:

`x <- 5`

`y <- "Zeichenkette"`

`z <- TRUE`

`w <- -2+7.5i`

Vektor erzeugen, Elementzugriff:

`v <- c(1,2,3.1415,-17); v[3]`

Rezyklierungs-Regel bei Vektoren:

`v+2` liefert `3,4,5.1415,-15`

Liste erzeugen, Elementzugriff:

`l <- list(1.2, "text", -5, FALSE);`

`l[[2]]`

(Pseudo-)Zufallszahl generieren:

`runif`

Demos starten (und bestaunen):

`demo()`

Einlesen eines Skripts:

`source`

Ausgabe in Datei umlenken:

`sink`

Grafikausgaben:

`plot, ...`

Grafik„geräte“ öffnen:

`x11, postscript, pdf,`

`dev.copy2eps, dev.copy2pdf, ...`

Grafikparameter setzen: `par`