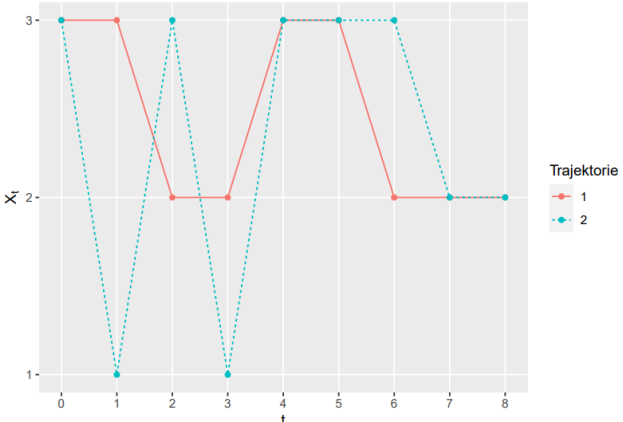


**Wiederholung WAHR**

<p><b>Unbedingte Wahrscheinlichkeit</b></p>	<p><math>P(X_6 = 1)</math> Keine Informationen über die bisherigen tatsächlich realisierten Zustände bekannt.</p> <p><b>Berechnung:</b> Es gibt <math>\binom{10}{5}</math> Möglichkeiten 5 schwarze und 5 weiße Kugeln anzuordnen. Die 2. Kugel soll schwarz sein <math>\rightarrow \binom{9}{4}</math> Möglichkeiten</p> <p><b>Laplace-Wahrscheinlichkeit</b></p> $P(X_2 = \text{schwarz}) = \frac{\binom{9}{4}}{\binom{10}{5}} = \frac{9!}{4! \cdot 5!} = \frac{9! \cdot 5!}{10! \cdot 4!} = \frac{5}{10} = \frac{1}{2}$ <p><b>Satz der totalen Wahrscheinlichkeit</b></p> $P(X_2 = s) = P(X_2 = s   X_1 = s) \cdot P(X_1 = s) + P(X_2 = s   X_1 = w) \cdot P(X_1 = w)$ $= \frac{4}{9} \cdot \frac{1}{2} + \frac{5}{9} \cdot \frac{1}{2}$ $= \frac{1}{2}$
<p><b>Bedingte Wahrscheinlichkeit</b></p>	<p><math>P(X_6 = 1   X_5 = x_5, X_4 = x_4, \dots, X_0 = x_0)</math> Information über alle vorherigen Zustände bekannt.</p> <p>Wissen wir nur den letzten Zustand so lässt sich nur folgende bedingte W'keit berechnen: <math>P(X_6 = 1   X_5 = x_5)</math></p> <p>Diese beiden W'keiten stimmen in der Regel nicht überein! Die Berechnung auf Basis von allen vorherigen Zuständen ist nützlicher jedoch auch aufwändiger.</p> <p><b>Berechnung:</b> Analog Unbedingte W'keit <math>\rightarrow</math> Achtung, Vorwissen über Zustände spielt eine Rolle</p> <p><b>Nochmals anschauen – Beispiel suchen</b></p>
<p><b>Allgemein</b></p>	<p>Stochastische Prozesse = Familien von Zufallsvariablen <b>Diskret (in Tagen / Schritten) oder kontinuierlich</b> <b>Deterministisch</b></p> <p>Zustandsraum S (endlich oder höchstens abzählbar unendlich) S={0,300}      S={0,1,2,...}      S=[0, ∞)</p> <p>Vier Typen von stochastischen Prozessen:</p> <ul style="list-style-type: none"> <li>• Zeitdiskret und zustandsdiskret</li> <li>• Zeitdiskret und zustandskontinuierlich</li> <li>• Zeitkontinuierlich und zustandsdiskret</li> </ul>

	<ul style="list-style-type: none"> <li>• Zeitkontinuierlich und zustandskontinuierlich</li> </ul>
<p><b>Zeit- und zustandsdiskrete stochastische Prozesse</b></p>	<p>Annahme: Indexmenge <math>\mathcal{T} = \mathbb{N}_0</math></p>  <p>Zwei Trajektorien eines zeitdiskreten Prozesses mit drei Zuständen</p>
<p><b>R-Code</b></p>	<p>Dichtefunktion ploten:</p> $f(x) = \begin{cases} \frac{3}{2}(1 - x^2), & \text{falls } 0 \leq x \leq 1 \\ 0, & \text{sonst} \end{cases}$ <pre>f &lt;- function(z) (3/2*(1 - z^2))*((z&gt;=0)&amp;(z&lt;=1)) x &lt;- seq(-2, 2, 0.01) plot(x, f(x), type = "l", main = "Dichte f(x)") plot(func)</pre>

Verteilung	Anwendung	R-Code	Wertebereich, Erwartungswert & Varianz	Funktion P(X=x)
Bernoulli $X \sim \text{Bin}(1, p)$	Indikator ob ein Ereignis eintritt (z.B. Münzwurf)  Zwei möglichen Ausgängen: Erfolg (Ja) und Misserfolg (Nein)	<code>dbinom(x, size = 1, prob = p)</code> ➔ Analog Binomial, <u>size = 1</u>	{0, 1} $E(X) = p$ $\text{Var}(X) = p \cdot (1 - p)$	$P(X = 1) = p$ $P(X = 0) = 1 - p$
Binomial $X \sim \text{Bin}(n, p)$	Anzahl Erfolge in n unabhängigen Bernoulli-Versuchen (z.B. Sport-Toto)  (ohne zurücklegen)	<code>dbinom(x = k, size = n, prob = p)</code>  <code>pbinom(x, size = n, prob = p)</code> # Kum. Verteilungsfunktion <code>qbinom(alpha, size = n, prob = p)</code> # Quantilfunktion <code>rbinom(n = 1, size = 32, prob = 0.659)</code> # Zufallszahlen  W'keit dass mehr als 25: <code>sum(dbinom(26:32, size = 32, prob = 0.659))</code> <code>1 - pbinom(25, size = 32, prob = 0.659)</code>  Zwischen 20 und 25: <code>sum(dbinom(20:25, size = 32, prob = 0.659))</code>	{0, 1, ..., n} $E(X) = n \cdot p$ $\text{Var}(X) = n \cdot p \cdot (1 - p)$	$P(X = k) = \binom{n}{k} \cdot p^k (1 - p)^{n-k}$  $p = W'$ keit für einen Erfolg  $p^k (1 - p)^{n-k}$ W'keit für ein Ereignis mit k Erfolgen und n - k Misserfolgen  $\binom{n}{k}$ Anzahl Ereignisse mit k Erfolgen und n Versuchen
Geometrische $X \sim \text{Geom}(p)$	Anzahl Misserfolge bis zum ersten Erfolg  Eigenschaft: Gedächtnislosigkeit, d.h. Wahrscheinlichkeit dafür, dass der erste Erfolg noch mindestens j weitere Fehlversuche lang auf sich warten lässt, ist unabhängig davon, wie viele Misserfolge man schon beobachtet hat.	<code>dgeom(k, prob = p)</code>  Kumulative Verteilung: <code>pgeom(x, prob = p)</code> <= x Misserfolgen <code>1 - pgeom(x, prob = p)</code> > x Misserfolgen  <code>qgeom(0.9, prob = 1/6)</code> # Quantile <code>rgeom(1, prob = 1/6)</code> # Zufallszahl (Anzahl Fehlwürfe)  Zwischen 1 -4: <code>sum(dgeom(1:4, prob = 1/6))</code>	{0, 1, ..., ∞} $E(X) = (1 - p) / p$ $\text{Var}(X) = (1 - p) / p^2$	$P(X = k) = p \cdot (1 - p)^k$  $p = W'$ keit für einen Erfolg $k = \text{Anzahl Misserfolge}$
Negative Binomial $X \sim \text{NBin}(r, p)$	Anzahl Misserfolge bis zum r-ten Erfolg (z.B. bis 3 Gewinne bei Rubbellos)	<code>dnbinom(x = , size = r, prob = p)</code>  <code>pnbinom(q = , size = r, prob = p)</code> # Kum. Verteilungsfunktion <code>qnbinom(p = , size = r, prob = p)</code> # Quantilfunktion <code>rnbinom(n = , size = r, prob = p)</code> # Zufallszahlen	{0, 1, ..., ∞} $E(X) = r(1 - p) / p$ $\text{Var}(X) = r(1 - p) / p^2$	$P(X = k) = \binom{k + r - 1}{k} p^r (1 - p)^k$  $p = W'$ keit für einen Erfolg $k = \text{Anzahl Misserfolge}$ $r = \text{Anzahl Erfolge}$
Hypergeometrische $X \sim \text{Hyp}(m, n, k)$	Anzahl Elemente j, in einer Stichprobe der Grösse k (z.B. Qualitätskontrolle) Kann auch für mit zurücklegen verwendet werden	<code>dhyper(x = , m = m, n = n, k = k)</code>  <code>phyper(q = , m = m, n = n, k = k)</code> # Kum. Verteilungsfunktion <code>qhyper(p = , m = m, n = n, k = k)</code> # Quantilfunktion <code>rhyper(nn = , m = m, n = n, k = k)</code> # Zufallszahlen	{0, 1, ..., min(k, m)} $E(X) = k \cdot \frac{m}{m+n}$ $\text{Var}(X) = k \cdot \frac{m}{m+n} \left(1 - \frac{m}{m+n}\right) \cdot \frac{(n+m-k)}{(m+n-1)}$	$P(X = j) = \frac{\binom{m}{j} \binom{n}{k-j}}{\binom{n+m}{k}}$  $m = \text{Anzahl defekter Schrauben} = 27$ $n = \text{Anzahl intakter Schrauben} = 973$ $k = \text{Stichprobengrösse} = 30$
Poisson $X \sim \text{Pois}(\lambda)$	Anzahl Ereignisse, wobei deren Eintreffrate $\lambda$ konstant ist (z.B. Unfälle, Kunden)	<code>dpois(x = k, lambda = lambda)</code>  <code>ppois(q, lambda = lambda)</code> # Kum. Verteilungsfunktion <code>qpois(p = 0.1, lambda = 32)</code> # Quantile W'keit 30:35 Kunden am Schalter <code>sum(dpois(x = 30:35, lambda = 32))</code>	{0, 1, ..., ∞} $E(X) = \lambda$ $\text{Var}(X) = \lambda$	$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$  $\lambda = \text{Rate, mit welcher die Ereignisse, in einer vorgegebenen Zeiteinheit oder Gebiet eintreffen}$

Grafische Darstellung der Verteilung in R :  
`plot(k, pk, type = "h", main = "Bin(n=32, p=0.659)")`

Verteilung	Anwendung	Dichte / Kum. Verteilungsfunktion	Erwartungswert / Varianz	R-Code
Uniform $X \sim \text{Unif}([a, b])$	Wenn alle Werte gleich wahrscheinlich sind	Dichte $f(x) = \begin{cases} 0 & \text{falls } x < a \\ 1/(b-a) & \text{falls } a \leq x \leq b \\ 0 & \text{falls } x > b \end{cases}$ Kum. Verteilungsfunktion: $F(x) = \begin{cases} 0 & \text{falls } x < a \\ \frac{x-a}{b-a} & \text{falls } a \leq x \leq b \\ 1 & \text{falls } x > b \end{cases}$	$E(X) = (a + b) / 2$ Entspricht der Mitte des Intervalls [a, b])  $\text{Var}(X) = (b - a)^2 / 12.$ Die Varianz erhöht sich, wenn die Intervalllänge (b - a) grösser wird.	<code>dunif(x = x, min = a, max = b) # Dichtefunktion</code>  # Kum. Verteilungsfunktion <code>P(X ≤ x) : punif(q = x, min = a, max = b)</code> <code>P(x ≤ X ≤ y) : punif(q=y,min=a,max=b) - punif(q=x,min=a,max=b)</code> <code>P(X &gt; x) : 1 - punif(q = x, min = a, max = b)</code>  <code>qunif(p = alpha, min = a, max = b) # Quantilfunktion</code> <code>runif(n = n, min = a, max = b) # Zufallszahlen</code>
Exponential $X \sim \text{Exp}(\lambda)$	Dauer von zufälligen Zeitintervallen (zwischen 2 Poisson-Ergebnissen)  Gedächtnislosigkeit	Dichte $f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$ Kum. Verteilungsfunktion $F(x) = P(X \leq x) = \begin{cases} 1 - e^{-\lambda x} & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$	$E[X] = \frac{1}{\lambda}$ Erwartete Wartezeit zwischen zwei Ereignissen  $\text{Var}(X) = 1/\lambda^2.$	<code>dexp(x = x, rate = lambda) # Dichtefunktion</code>  # Kum. Verteilungsfunktion <code>P(X ≤ x) : pexp(q = x, rate = lambda)</code> <code>P(x ≤ X ≤ y) : pexp(q = y, rate = lambda) - pexp(q = x, rate = lambda)</code> <code>P(X &gt; x) : 1 - pexp(q = x, rate = lambda)</code> <code>qexp(p = alpha, rate = lambda) # Quantilfunktion</code> <code>rexp(n = n, rate = lambda) # Zufallszahlen</code>
Weibull $X \sim \text{Weibull}(\lambda, \beta)$	Dauer von zufälligen Zeitintervallen mit Berücksichtigung der Lebensphase	Dichte $f(x) = \begin{cases} \lambda \cdot \beta \cdot (\lambda \cdot x)^{\beta-1} \cdot e^{-(\lambda x)^\beta} & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$ Kum. Verteilungsfunktion $F(x) = \begin{cases} 1 - e^{-(\lambda x)^\beta} & \text{falls } x \geq 0 \\ 0 & \text{falls } x < 0 \end{cases}$	$E[X] = \frac{1}{\lambda} \Gamma(1 + 1/\beta)$  $\text{Var}(X) = \frac{1}{\lambda^2} [\Gamma(1 + 2/\beta) - \Gamma^2(1 + 1/\beta)]$ Eigenschaften: Für $\beta = 1$ entspricht die Weibullverteilung der Exponentialverteilung.	# Dichtefunktion <code>dweibull(x, shape = beta, scale = 1/lambda)</code>  # Kum. Verteilungsfunktion <code>P(X ≤ x) : pweibull(q, shape = beta, scale = 1/lambda)</code>  <code>qweibull(p, shape = beta, scale = 1/lambda) # Quantilfunktion</code> <code>rweibull(n, shape = beta, scale = 1/lambda) # Zufallszahlen</code>
Gamma $X \sim G(\alpha, \beta)$	Summe von exponentialverteilten Zufallsvariablen, Kosten	Dichte $f(y) = e^{-\lambda y} \cdot y^{\alpha-1} \cdot \frac{\lambda^\alpha}{\Gamma(\alpha)} \quad \text{mit } y > 0$	$E(X) = \frac{\alpha}{\beta}$ $\text{Var}(X) = \frac{\alpha}{\beta^2}$  $\alpha > 0$ ist der Formparameter. $\lambda > 0$ ist die Rate. bzw. $1/\lambda$ ist der Skalenparameter	<code>dgamma(x, shape = alpha, rate = lambda) # Dichtefunktion</code>  # Kum. Verteilungsfunktion <code>P(X ≤ x) : pgamma(q, shape = alpha, rate = lambda)</code>  <code>qgamma(p, shape = alpha, rate = lambda) # Quantilfunktion</code> <code>rgamma(n, shape = alpha, rate = lambda) # Zufallszahlen</code>
Normal / Gauss $X \sim N(\mu, \sigma^2)$	Abweichung von (Mess-) Werten, Näherung bei grossen Stichproben	Dichte $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(x - \mu)^2}{\sigma^2}\right)$ Kum. Verteilungsfunktion $F(x) = P(X \leq x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{1}{2} \frac{(z - \mu)^2}{\sigma^2}\right) dz$	$E(X) = \mu$ $\text{Var}(X) = \sigma^2$	<code>dnorm(x = x, mean = mu, sd = sigma) # Dichtefunktion</code> # Kum. Verteilungsfunktion <code>P(X ≤ x) : pnorm(x, mean = mu, sd = sigma)</code> <code>P(a ≤ X ≤ b) = P(X ≤ b) - P(X ≤ a) :</code> <code>pnorm(b, mean = mu, sd = sigma) - pnorm(a, mean = mu, sd = sigma)</code> <code>P(X &gt; x) : 1 - pnorm(x, mean = mu, sd = sigma)</code> Oder: <code>pnorm(x, mean = .., sd = .., lower.tail = F)</code>  <code>qnorm(alpha, mean = mu, sd = sigma) # Quantilfunktion</code>
Lognormal $X \sim \log N(\mu, \sigma^2)$	Rechtsschiefe Messdaten mit nur positiven Werten	Dichte $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(\log(x) - \mu)^2}{\sigma^2}\right)$	$E(X) = e^{(\mu + \frac{\sigma^2}{2})}$ $\text{Var}(X) = e^{(2\mu + \sigma^2)}(e^{\sigma^2} - 1)$ Für die logarithmierte Zufallsvariable gilt: $E(\log(X)) = \mu$ und $\text{Var}(\log(X)) = \sigma^2$	<code>lnorm(.., meanlog = mu, sdlog = sigma) -&gt; log. Zufallsvariable</code>  Mit welcher W'keit fliegt eine Person mehr als 20000 Meilen? <code>P(X &gt; 20000) 1-plnorm(20000, 8.94, 0.977)</code>

**Markov-Ketten**

**Markov-Ketten**

$$P(X_{t+1} = x_{t+1} | X_t = x_t)$$

Bedingte Verteilung: Die zukünftige Entwicklung des Prozesses hängt nur vom augenblicklichen Zustand ab, aber nicht von vergangenen Zuständen (egal ob bekannt oder nicht) → Der Prozess hat kein Gedächtnis.

Sofern die gesamte Vergangenheit wichtig ist, handelt es sich nicht um einen Markov-Prozess.

Man kann einen Prozess markovsch machen, indem man den Zustandsraum vergrößert / Teil der Vergangenheit in die Definition des Zustandsraumes aufnimmt.

$X_{n+1}$  ist nur von  $X_n$  abhängig,  $X_n$  ist aber wiederum abhängig von  $X_{n-1}$  → somit ist auch  $X_{n+1}$  von  $X_{n-1}$  abhängig

**Homogen:**

Wenn für alle t gilt:  $P(X_{t+1} = j | X_t = i) = P(X_1 = j | X_0 = i)$

→ Übergangswahrscheinlichkeit hängt nicht von Zeit ab

**Inhomogen:**

Vom Zeitpunkt abhängig (werden nicht behandelt)

**Übergangsmatrix:**

Endlicher Zustandsraum mit N Zuständen →  $N^2$  mögliche Übergänge

- Zusammenfassung in quadratische Matrix P
- Zeile i: augenblicklicher Zustand
- Spalte j: (potenzieller) nächster Zustand
- $0 \leq p_{ij} \leq 1$  für alle i, j ∈ S
- Überprüfung: Zeilensummen = 1 (Spaltensummen ≠ 1)
- Stochastische Matrix

**Übergangsdiagramm:**

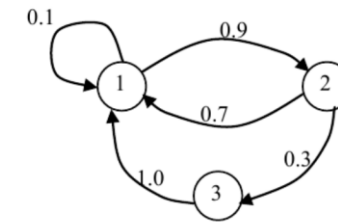
- Jeder Zustand: Kreis
- Übergangsw'keiten: Pfeile zwischen den Kreisen
- Pfeile werden weggelassen, wenn W'keit = 0

**Vorgehen:**

1. Wie sieht der Zustandsraum aus?
2. Handelt es sich um eine Markov-Kette?
3. Wenn ja, ist sie homogen?
4. Bestimme die Übergangsmatrix und den Übergangsgraphen

Beispiel:

Übergangsgraph zur Matrix  $P = \begin{pmatrix} 0.1 & 0.9 & 0 \\ 0.7 & 0 & 0.3 \\ 1 & 0 & 0 \end{pmatrix}$ :



**R-Code:**

**Logistic Map**

```

# Parameter
x0_1 <- 0.9
x0_2 <- 0.9
r <- 3.76
Tmax <- 100

# Wenn wir für r einen Wert zwischen 3.57 und 4 verwenden,
# verhält sich das System chaotisch.

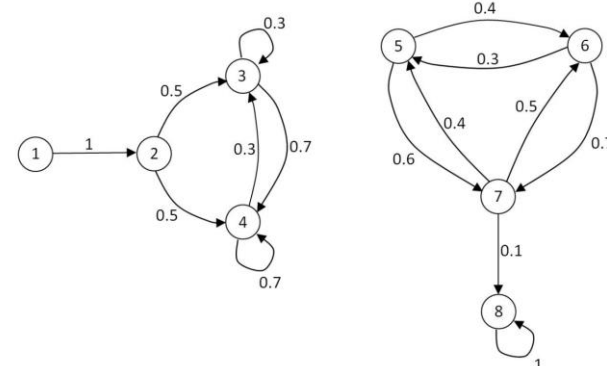
# Funktion
logistic_map <- function(r, x0, Tmax){
  x <- x0
  out <- rep(NA, Tmax)
  out[1] <- x # x0 an 1. Position einfügen

  for (i in 2:Tmax) { # Ab 2 da 1. Position = x0
    out[i] <- (r*out[(i-1)] * (1 - out[(i-1)]))
  }
  return(out)
}

# Plots
par(mfrow = c(2,1))
plot(x = NA, y = NA, xlim = c(0, Tmax), ylim = c(0,1), xlab = "x", ylab = "y")
lines(logistic_map(r, x0_1, Tmax), col = 1)
lines(logistic_map(r, x0_2, Tmax), col = 2)

plot(x = NA, y = NA, xlim = c(0, Tmax), ylim = c(0,1), xlab = "x", ylab = "y")
lines(logistic_map(r, x0_1, Tmax), col = 1)
lines(logistic_map(r, x0_2+1e-7, Tmax), col = 2)
  
```

	<p><b>Random Walk</b></p> <pre># Funktion random_walk &lt;- function(x0, Tmax){   x &lt;- x0   epsilon &lt;- rnorm(Tmax)   out &lt;- rep(NA, Tmax)   out[1] &lt;- x    for (i in 2:Tmax) {     out[i] &lt;- out[(i-1)] + epsilon[i]   }    return(out) }</pre>
<p><b>Zustandsverteilungen</b></p>	<p>Wahrscheinlichkeit, mit der sich der Prozess zum Zeitpunkt t im Zustand i befindet:  <math>\pi_i(t) = P(X_t = i)</math></p> <p>Zustandsvektor <math>\vec{\pi}(t)</math> gibt genau die Verteilung der diskreten Zufallsvariable <math>X_t</math> an (keine bedingte Verteilung, hängt vom Startzustand ab)</p> <p><b>t-Schritt-Übergangsmatrix:</b>          Übergangswahrscheinlichkeiten aus einem Zustand zum Zeitpunkt 0 zu einem Zustand zum Zeitpunkt t, beschreibt die zeitliche Entwicklung einer Wahrscheinlichkeitsverteilung, nicht aber die zeitliche Entwicklung einer Realisierung (Trajektorie)  <math>\vec{\pi}(t) = \vec{\pi}(0) * P^t \rightarrow P^t = t - \text{Schritt} - \text{Übergangsmatrix}</math>          Zeilenvektor = Zeilenvektor * Matrix (Lin. Algebra: Spaltenvektor)</p> <p>Anfangszustand:</p> <ul style="list-style-type: none"> <li>- Muss nicht deterministisch sein</li> <li>- Verteilung ist bekannt</li> </ul> <p>Spezialfall:  <math>t = 0 \rightarrow P^0 = I</math> (Einheitsmatrix)</p> <p><b>Erreichbarkeit</b>          Ein Zustand <math>j \in S</math> heisst vom Zustand <math>i \in S</math> aus erreichbar, wenn ein <math>t \in \mathbb{N}_0</math> existiert, so dass die Wahrscheinlichkeit, sich nach t Schritten im Zustand j zu befinden, wenn man sich aktuell im Zustand i befindet, grösser als 0 ist.  <math>P(X_{s+t} = j   X_s = i) &gt; 0</math> oder <math>(P^t)_{ij} &gt; 0</math>          erreichbar: <math>i \rightarrow j</math>          nicht erreichbar: <math>i \nrightarrow j</math> Somit: <math>j \neq i</math> (absorbierender Zustand)          wechselseitig erreichbar: <math>i \leftrightarrow j</math></p>

	<p>Absorbierender Zustand: Zustand den man niemals wieder verlässt  <math>\rightarrow i</math> ist absorbierend, wenn: <math>P_{ii} = 1</math></p> <p><b>Klassifikation von Zuständen</b>          Zwei Zustände liegen in derselben Klasse, wenn: <math>i \leftrightarrow j</math></p> <p>Klasseneigenschaften:          Ein Zustand <math>i \in S</math> heisst rekurrent: <math>P(R_j &lt; \infty   X_0 = j) = 1</math>  <math>\rightarrow</math> wenn man bei Start in j sicher wieder nach j zurückkehrt          Ein Zustand <math>i \in S</math> heisst transient: <math>P(R_j &lt; \infty   X_0 = j) &lt; 1</math>  <math>\rightarrow</math> wenn man bei Start in j mit positiver Wahrscheinlichkeit nicht wieder nach j zurückkehrt  <math>\rightarrow</math> <b>Innerhalb einer Klasse sind alle Zustände rekurrent oder transient</b></p> <p><b>Irreduzibel / Reduzibel</b>          Wenn jeder Zustand einer Markov-Kette von jedem anderen Zustand aus erreichbar ist <math>\rightarrow</math> ansonsten reduzibel          Wichtige Eigenschaft für die Untersuchung des Langzeitverhaltens.</p> <p>Bsp.:</p>  <p>Wir zerlegen <math>S = \{1, 2, \dots, 8\}</math> in Klassen wechselseitig erreichbarer Zustände:  <math>S = \{1\} \cup \{2\} \cup \{3, 4\} \cup \{5, 6, 7\} \cup \{8\}</math>,</p> <p>wobei die Klassen <math>\{1\}</math>, <math>\{2\}</math> und <math>\{5, 6, 7\}</math> transient und die Klassen <math>\{3, 4\}</math> und <math>\{8\}</math> rekurrent sind.</p> <p>Da S in mehrere Klassen zerfällt, ist die Markov-Kette reduzibel.</p> <p><b>Periode eines Zustandes</b>          Anzahl Schritte, nach denen eine Rückkehr in einen Zustand minimal möglich ist.          Ist <math>d \in \mathbb{N}</math> die grösste Zahl, für die gilt, dass  <math>P(X_n = i   X_0 = i) = 0</math> oder <math>(P^n)_{ii} = 0</math></p>
--	---

für alle  $n$ , die nicht durch  $d$  teilbar sind, so nennt man  $d$  die Periode des Zustands  $i$ . Ist  $d = 1$ , so nennt man den Zustand  $i$  aperiodisch, ist  $d > 1$ , so heisst der Zustand periodisch.

Die Umkehrung gilt nicht: Wenn alle Diagonalelemente der Übergangsmatrix Null sind, kann die Markov-Kette sowohl periodisch als auch aperiodisch sein.

Wie erkennt man periodische Prozesse?

Am einfachsten geht das im Übergangsgraphen. Man verfolgt die Übergangspfeile von einem Zustand über die anderen Zustände zu sich selbst zurück. Dies sind gerichtete Zyklen. Falls die Längen aller diese Zyklen ein Vielfaches einer Zahl  $d > 1$  sind, ist der Prozess periodisch.

Ist ein Diagonalelement der irreduziblen Markov-Kette ungleich 0 so ist diese aperiodisch.

Sind alle Elemente gleich 0 so kann man nicht in einem Schritt in einen Zustand zurückkehren (erst nach 2 Schritten).

**R-Code:**

Matrizenmultiplikationen: `%**%`

Potenzen einer Matrix: `%^%` Paket: `expm`

- Bei gleichzeitiger Verwendung Klammern setzen
- Ohne Klammern: Matrizen werden komponentenweise multipliziert / potenziert.

Überprüfung stochastische Matrix:

`rowSums(P5) = 1`

Vektor-Matrix-Multiplikation:

`t(c(1,0,0)) %**% P5` (zuerst Vektor transponieren)

**Bsp. aus AB 3:** Berechnen und plotten der Wahrscheinlichkeit, an der  $k$ -ten Ampel rot zu haben, nullten Ampel grün ( $k = 1, 2, \dots, 30$ )

```
P <- matrix(c(0.4, 0.6, 0.9, 0.1), ncol=2, byrow=TRUE)
Pt <- c(1,0) # Nullte Ampel grün, Anfangszustand
pRot <- rep(0, 30)
for (i in 1:30) {
  Pt <- Pt %**% P
  pRot[i] <- Pt[2]
}
plot(pRot, type="l");points(pRot);abline(h=0.4)
```

Plotten Sie  $\bar{\pi}(t)$  jeweils für die Anfangsverteilung  $\bar{\pi}(0) = (1, 0)$  und  $\bar{\pi}(0) = (0, 1)$  gegen die Zeit  $t$ .

```
library("expm")
P <- matrix(c(0, 1, 1, 0), ncol = 2, byrow = T)
tt <- 1:10
verteilung <- function(tt, P, start) {
  sapply(tt, function(i) (P %**% i)[start,])
}
```

```
d1 <- verteilung(tt, P, start = 1) # start = 2: Start bei Parameter 2
plot(tt, d1[1,], type = 'o', col = 'red', ylab = "pi(t)")
lines(tt, d1[2,], type = 'o', col = 'green')
```

**Bsp. aus AB 4:** Wahrscheinlichkeit in 4 zu landen bei Start in 3:

```
library(expm)
(P %**% 3)[1,4] # Zeile 1, Spalte 4 (Start in eins Ziel in 4)
```

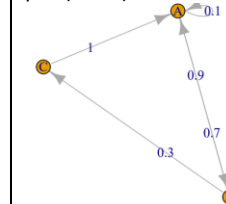
**Paket markovchain**

```
library(markovchain)
P <- rbind(c(0.1, 0.9, 0), c(0.7, 0, 0.3), c(1, 0, 0))
kette <- new("markovchain", states = c("A", "B", "C"),
            transitionMatrix = P)
```

`kette`

`summary(kette)` # weitere Infos über Kette (z.B. Aufteilung in Klassen, Angabe irreduzibel / reduzibel, absorbierende Zustände)

`plot(kette)` # Übergangsgraphen



`rmarkovchain(n = 50, object = kette, t0 = "A")` #Simulation

- $X_0$  in Vektor integrieren: `include.t0=TRUE`

Ist nur eine Anfangsverteilung vorgegeben, muss erst ein Startzustand gezogen werden:

```
pi_0 <- c(0.1, 0.7, 0.2)
zustaende <- states(kette) # Vektor mit Namen
x_0 <- sample(zustaende, 1, prob = pi_0)
rmarkovchain(50, kette, t0 = x_0)
```

Potenzen und Produkte können in diesem Paket direkt mit `*` und `^` berechnet werden.

Extrahieren einer Übergangsmatrix:

`kette@transitionsMatrix` oder `(kette^5)@transitionMatrix`

Periode bestimmen:

`period(ehrenfest)`

Schätzung der Übergangsmatrix:

`markovchanFit(x)$estimate` # **Schätzung! entspricht nicht den wahren Werten**

	<p>Wenn Nullen in der geschätzten Matrix ein Problem sind: Laplace Korrektur:  <code>markovchainFit(x, method = "laplace", laplacian = 0.1)\$estimate</code></p>
<p><b>Aufenthaltsdauer für endlichen Zeithorizont</b></p>	<p>Aufenthaltsdauer entspricht der Anzahl Zeitschritten, während derer der Prozess im gesuchten Zustand ist.</p> <ul style="list-style-type: none"> <li>➔ Entspricht einer Zufallsvariabel</li> <li>➔ Wird durch eine Verteilung beschrieben</li> <li>➔ Erwartete Dauer hängt vom Anfangszustand ab</li> <li>➔ Als absolute Zahl oder besser als Anteil</li> </ul> <p>Anzahl Aufenthalte im Zustand j:</p> $Z_j(T) = \sum_{i=0}^T Y_i$ <p>mit</p> $Y_i = \begin{cases} 1, & X_i = j \\ 0, & X_i \neq j \end{cases}$ <p>Es gilt</p> $E(Y_i) = 1 \cdot P(Y_i = 1) + 0 \cdot P(Y_i = 0) = \pi_j(i)$ <p>und damit</p> $E(Z_j(T)) = E\left(\sum_{i=0}^T Y_i\right) = \sum_{i=0}^T E(Y_i) = \sum_{i=0}^T \pi_j(i)$ <p>Jeder Zeitschritt ist <math>\pi_j(t) = \vec{\pi}(t) * \vec{e}_j</math> vom Erwartungswert          Erwartungswert: <math>E(Z_j(T)) = \vec{\pi}(0) * M(T) * \vec{e}_j</math></p> <p><b>Aufenthaltsdauermatrix M (T)</b>          Erwartete Aufenthalte entspricht der Zeilensumme (T + 1)          Bei einer festen Anfangsverteilung:          Vektor der erwartenden Aufenthalte <math>\vec{N} = \vec{\pi}(0) * M(T)</math></p> <p><b>R-Code:</b>          Beispiel: Das erste Paket wurde mit einem Fehler geliefert, Betrachtung der nächsten 10 Pakete. Wie viele Pakete haben im Mittel einen Fehler?  <code>library("expm")</code></p> <pre>M &lt;- P %^% 0 for (i in 1:10) {   M &lt;- M + P %^% i } M M[1,1] - 1 #Matrizen-Elemente auswählen (hier: Zeile 1, Spalte 1)</pre>

	<p>Beispiel: Berechnung M (T), T = 4:  <code>library(expm)</code>  <code>P &lt;- rbind(c(0.9,0.1), c(1,0))</code>  <code>M4 &lt;- diag(c(1,1))</code>  <code>for (k in 1:4) {</code>  <code style="padding-left: 2em;">M4 &lt;- M4 + (P %^% k)</code>  <code>}</code>  <code>M4</code></p> <p>Im Zustand 2 defekt:          Bei Start in 1 erhalten wir also</p> $E(Z_2(4)) = \vec{\pi}(0) \cdot M(4) \cdot \vec{e}_2'$ $= (1 \ 0) \cdot \begin{pmatrix} 4.6281 & 0.3719 \\ 3.7190 & 1.2810 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $= 0.3719,$ <p>was wir natürlich auch direkt aus M(4) hätten ablesen können.          Man beachte, dass die Zeilensummen von M(4) beide T + 1 = 5 sind!</p>
<p><b>Kostenmodelle für endlichen Zeithorizont</b></p>	<p><b>Zustandsabhängige Kosten</b>          Gesamtkosten K (Zufallsvariabel):</p> $K = \sum_{t=0}^T K_t$ <p>Zusammenfassung der einzelnen Zustände in einem Kostenvektor <math>\vec{c}</math></p> <p>Erwartete Kosten zum Zeitpunkt t: <math>E(K_t) = \vec{\pi}(t) * \vec{c}</math>          Erwartete Gesamtkosten: <math>E(K) = \vec{\pi}(0) * M(T) * \vec{c}</math></p> <p>Bsp.:</p> <p><b>Beispiel:</b> Eine Versicherung plant, eine Reiseversicherung anzubieten. Die Versicherung soll während einer Reisezeit von 7 Tagen 100 CHF pro Krankheitstag bezahlen. Die Bedingung ist, dass der Versicherungsnehmer gesund ist am Tage des Reiseantritts.</p> <p><b>Wie teuer muss die Police sein, damit die Versicherung keinen Verlust macht?</b></p> <p>Wir setzen folgendes <b>markovsches Krankheitsmodell</b> voraus: Eine Person, die heute gesund ist, ist morgen krank mit einer Wahrscheinlichkeit von 0.05. Eine kranke Person ist mit einer Wahrscheinlichkeit von 0.6 am folgenden Tag wieder gesund.</p> <p>Der Prozess hat den Zustandsraum <math>S = \{1, 2\}</math> mit (1 = gesund, 2 krank)</p>



Die Übergangsmatrix ist dann

$$P = \begin{pmatrix} 0.95 & 0.05 \\ 0.6 & 0.4 \end{pmatrix}.$$

Der Kostenvektor ist:

$$\vec{c} = (c_1 \ c_2) = (0 \ 100).$$

Die Anfangsverteilung ist gegeben durch  $\vec{\pi}(0) = (1, 0)$ , denn die Versicherung soll ja nur für Personen gelten, die bei Antritt der Reise gesund sind.

Der Erwartungswert der Gesamtkosten für den Zeitraum  $\{0, 1, 2, \dots, 6\}$  ist dann gegeben durch

$$\begin{aligned} E(K) &= \vec{\pi}(0) \cdot M(T) \cdot \vec{c}' \\ &= (1 \ 0) \cdot M(T) \cdot \begin{pmatrix} 0 \\ 100 \end{pmatrix} \end{aligned}$$

Mit

$$M(T) = \sum_{t=0}^6 P^t = \begin{pmatrix} 6.5798 & 0.4202 \\ 5.0423 & 1.9577 \end{pmatrix}$$

ergibt sich:

$$E(K) = (1 \ 0) \cdot \begin{pmatrix} 6.5798 & 0.4202 \\ 5.0423 & 1.9577 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 100 \end{pmatrix} = 42.02$$

Die Versicherung muss also mindestens 42.02 CHF für die Police verlangen, wenn sie keinen Verlust machen will.

### Übergangsabhängige Kosten

Übergangskosten Matrix U: Treten bei einem Übergang keine Kosten an, so ist das entsprechende Matricelement = 0.

Erwartungswert:  $E(K_{t,t+1}) = \vec{\pi}(t) * \vec{c}$

Gesamtkosten:  $E(K) = \vec{\pi}(0) * M(T - 1) * \vec{c}$

Zur Berechnung der erwarteten Anzahl Sprünge von einem Zustand in einen anderen verwendet man eine Kostenmatrix, die aus lauter Einsen besteht mit Ausnahme der Hauptdiagonalen:

$$U = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Man sieht also, dass mit den Konzepten von zustands- und übergangsabhängigen Kosten eine ganze Reihe von interessierenden Prozesseigenschaften behandelt werden können, zumindest was deren Erwartungswert angeht.

Anfangswert = 0 -> kein Sprung in diesen Zustand

Komponenten von  $\vec{c}$ : Zeilensumme der Matrix  $P*U$  komponentenweise multiplizieren (R:  $P*U$ , ohne %\*%)

Gesucht: Anzahl Sprünge von Zustand 2 in 3, Kostenmatrix U:

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{Bei unabhängigem Ausgangszustand wäre die gesamte 3. Spalte = 1}$$

### R-Code:

#### Durchschnittliche Kosten ermitteln:

```
1. Funktion für M
getM <- function(P, Tmax) {
  M <- diag(dim(P)[1])
  for (i in 1:Tmax) {
    M <- M + P %%^ i
  }
  M
}
```

```
2. Kosten berechnen, genügend grosses Tmax wählen!
(k1 <- t(c(1,0,0,0,0)) %*% getM(P, Tmax=1000) %*% kosten)
Startvektor
```

### Stationäre Eigenschaften

#### Verhalten nach Einschwingphase (Transiente Eigenschaften):

#### Asymptotische Verteilung oder auch (eindeutige) Grenzverteilung

Gibt es eine Zustandsverteilung, so dass unabhängig vom

Anfangszustand gilt:  $\pi(\infty) = \lim_{t \rightarrow \infty} \pi_j(t)$

→ Da sich nichts mehr ändern darf: stets stationär

#### Stationäre Verteilung:

Sofern sich in den nächsten Schritten nichts mehr ändert (Potenzen vergrößern bis sich nichts mehr ändert -> annähernder Grenzwert gefunden).

$$\vec{\pi}^* = \vec{\pi}^* \cdot P$$

Eine stationäre Verteilung muss nicht zwingend asymptotisch sein.

#### Nicht eindeutige Grenzverteilung:

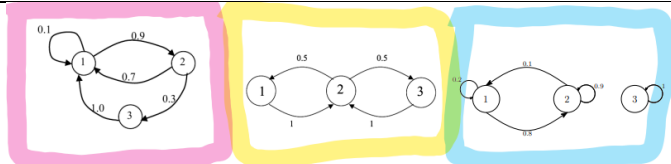
Sofern die Grenzverteilung von der Anfangsverteilung abhängig ist.

Bsp.: Reduzible Markov-Kette kann mehrere Grenzverteilungen haben (je nach Anfangsverteilung, alle Verteilungen sind jedoch auch stationäre Verteilungen).

#### Aperiodische, irreduzible Markov-Kette mit endl. Zustandsraum:

- Eindeutige stationäre Verteilung
- Eindeutige Grenzverteilung = stationäre Verteilung
- Jede Markov-Kette konvergiert gegen ihre stationäre Verteilung – unabhängig von der Startverteilung





**Beispiel 1** (links): **Aperiodisch und irreduzibel**. Konvergenz gegen Grenzverteilung, unabhängig von Startverteilung.  
**Beispiel 2** (mitte): **Periodisch**. Markov-Kette konvergiert nicht.  
**Beispiel 3** (rechts): **Reduzibel**. Markov-Kette konvergiert für jede Startverteilung, aber Grenzverteilung hängt von der Startverteilung ab.

**Eigenwerte und stationäre Verteilungen**

Eine stationäre Verteilung ( $\vec{\pi}^*$ ) ist ein linker Eigenvektor zum Eigenwert 1. Somit ist ein linker Eigenvektor von P ein rechter Eigenvektor von P'

Ein Eigenvektor ( $\vec{\pi}^*$ ) ist nur dann eine stationäre Verteilung, wenn zusätzlich

- $\sum_{i=1}^N \pi_i^* = 1$  und
- $\pi_i^* \geq 0$  für alle  $i = 1, \dots, N$ .

Wenn  $\pi^*$  nur Komponenten mit dem selben Vorzeichen hat, können wir dies stets erreichen, in dem wir mit der **Summe der Komponenten normieren**.

**EVEN mit wechselnden Vorzeichen** lassen sich nicht zu einer Zustandsverteilung machen!

Die Suche nach einer stationären Verteilung reduziert sich also auf die **Berechnung einer linken Eigenvektors von P bzw. rechten Eigenvektors von P' zum Eigenwert 1**, der nur Komponenten mit dem selben Vorzeichen hat.

Wenn von Startverteilung abhängig -> Grenzverteilung nicht eindeutig.

**R-Code:**

**Grenzverteilung finden:**

$P^t$  ausprobieren: `P %%% t` (immer grössere Zahl wählen bis eingeschwungen)

**Stationäre Verteilung:**

$$\vec{\pi}^* = \left( \frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{4} \right)$$

`P <- rbind(c(0, 1, 0), c(0.5, 0, 0.5), c(0, 1, 0))`

`v <- c(1/4, 1/2, 1/4)`

`t(v) %%% P`

→ Ergebnis analog v

Potenzen bilden  $P^t$  -> es existieren zwei Matrizen t gerade & t ungerade -> es existiert keine Grenzverteilung,  $P^t$  wechselt zwischen zwei Matrizen, diese Markov-Kette ist periodisch mit  $d = 2$

**Eigenwerte und stationäre Verteilungen**

`P <- rbind(c(0.4, 0.6), c(0.9, 0.1))` -> alle Werte positiv: Markov-Kette aperiodisch und irreduzibel, eine stationäre Verteilung `eigen(t(P))`

→ Komponenten beide +, Summe nicht = 1. Beheben, indem wir durch Summe teilen:

`v <- eigen(t(P))$vectors[,1]`

`v <- v/sum(v)`

`v`

-> rechter EV zu P' zum EW 1 -> für linken EV von P: transponieren

`all.equal()` -> Vergleiche innerhalb einer Toleranz, Überprüfung ob  $EW == 1$ , anstelle von  $== 1$

**Wie hoch ist (auf lange Sicht) der durchschnittliche Gewinn?**

`gewinne <- rowSums(P*U)`

`gewinne`

`t(pi_stat) %%% gewinne` # Anstelle von pi\_stat der eigenv. V (siehe oben) anwenden

**markovchain-Paket:**

`library("markovchain")`

`P -> Matrix erstellen`

`kette <- new("markovchain", transitionMatrix=P)`

`steadyStates(kette)`

Falls mehrere stationäre Verteilungen möglich, Matrix wird ausgegeben:

Alle Zeilenvektoren sind stationäre Verteilungen, aber auch gewichtete Summen, sofern die Gewichte nicht negativ sind und sich zu 1 addieren:

`v <- 0.25 * stat[1,] + 0.75 * stat[2,]`

`v`

`t(v) %%% mc@transitionMatrix`

**Asymptotische Besetzung**

Für aperiodische und irreduzible Markov-Ketten gilt mit W'keit 1:

$$\vec{\hat{\pi}} = \vec{\pi}(\infty)$$

d.h. für fast jede Realisation des Prozesses ist die asymptotische Besetzung gleich der eindeutigen Grenzverteilung (Ergodensatz).

Bildet man den Mittelwert einer Funktion  $g$ , die von den Zuständen abhängt, so kann man entweder eine einzelne Trajektorie verfolgen und die relevante Grösse über einen sehr langen Zeithorizont mitteln (Zeitmittel) oder aber die Grösse zu einem festen Zeitpunkt (nachdem die Kette sich eingeschwungen hat) als Mittelwert über sehr viele Trajektorien bestimmen (Scharmittel).

**BEISPIEL ??**

**Asymptotische Kostenmodelle**

Für Prozesse mit unterschiedlich langen Laufzeiten

**Asymptotische Durchschnittskosten**

Voraussetzung: eindeutige Besetzungsverteilung  $\vec{\hat{\pi}}$ , die unabhängig

	<p>vom Anfangszustand ist (aperiodische und irreduzible Ketten erfüllen dies immer).</p> <p>Zeitmittel / Scharmittel: Kosten pro Zeitschritt: <math>E(\bar{K}) = \bar{\pi}(\infty) * \bar{c}</math></p> <p><b>R-Code:</b> <b>Beispiel AB 6:</b> Wie viel darf dieser Kaffee pro Kunde und Einkauf kosten? Durch den Kaffee ändert sich die Verteilung. Betrachten Sie dazu das asymptotische Kundenverhalten. (P &lt;- matrix(c(0.8, 0.2, 0.2, 0.8), ncol = 2, byrow = T))</p> <p># Langfristiger durchschnittlicher Gewinn: (es &lt;- eigen(t(P))) (ev &lt;- es\$eigenvectors[,1])/sum(es\$eigenvectors[,1]))</p> <p># Gewinnverteilung cost &lt;- c(10,0) ev %*% cost # Gewinn pro Woche: CHF 5.-</p> <p># Veränderung des Kundenverhaltens durch Kaffee: (Pmit &lt;- matrix(c(0.9, 0.1, 0.2, 0.8), ncol = 2, byrow = T))</p> <p>(es &lt;- eigen(t(Pmit))) es &lt;- es\$eigenvectors[,1]/sum(es\$eigenvectors[,1]) es</p> <p># Kostenrechnung c = (10-x,0) erwartete Gewinn: <math>(2/3 \cdot 1/3) * (10-x) = 2/3(10-x)</math> dieses Resultat mit dem Gewinn ohne Kaffee gleichsetzen und nach x auflösen</p> <p>Wie lange dauert es, bis die Massnahme greift? library(expm) zeit &lt;- 0:15 p &lt;- sapply(zeit, function(z) c(0.5,0.5) %*% (Pmit %^z)) plot(zeit, p[1,], col="green", type="b", ylim=c(0,1), xlab="Woche", ylab="p") lines(zeit, p[2,], col="red", type="b") abline(h=ev)</p>
--	---

	<p><b>Diskontierte Gesamtkosten</b> <math>E(K_\alpha) = \bar{\pi}(0) \cdot (I - \alpha \cdot P)^{-1} \cdot \bar{c}</math> Wobei <math>\alpha = \frac{1}{1+z} \rightarrow z</math> kann z. B. einen Zinssatz 0.02 sein Allgemein: <math>CLV(C, r, \alpha) = C \cdot \frac{1}{1 - \alpha r}</math> <math>CLV(600, 0.7, 1) = \frac{600}{1 - 0.7} = 2000</math></p> <p>Ohne Diskontierung:</p> <p>Mit Diskontierung: <math>CLV(600, 0.7, 0.95) = \frac{600}{1 - 0.95 \cdot 0.7} = 1791.045</math></p> <p>Cramersche Regel: <math>\begin{pmatrix} a &amp; b \\ c &amp; d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d &amp; -b \\ -c &amp; a \end{pmatrix}</math></p> <p><b>R-Code:</b> <b>Durchschnittlicher Gesamtgewinn mit Diskontierung</b> P &lt;- matrix(c(1,0, 0, 0.35, 0.55, 0.1, 0.3, 0.1, 0.6), ncol = 3, byrow = T) cost &lt;- c(0, 600, 720) pi_0 &lt;- c(0,1,0) (clv &lt;- t(pi_0) %*% solve(diag(3)- 0.95*P) %*% cost)</p>
<b>Markov Chain Monte Carlo</b>	<p>Detailed-Balance-Bedingung: Ein Zustand erhält genauso viele Teilchen wie er abgibt</p> <p>Wahrscheinlichkeit für einen Sprung von i nach j, sodass die Detailed-Balance-Bedingung erfüllt ist: Bedingung (3.2) kann leicht erfüllt werden, indem man 2 Fälle unterscheidet, <math>(\alpha_{ij}, \alpha_{ji} \in [0, 1])</math>:</p> <ul style="list-style-type: none"> <li><math>\pi_j \cdot q_{ji} \geq \pi_i \cdot q_{ij}</math>: Setze <math>\alpha_{ij} = 1</math> und <math>\alpha_{ji} = \frac{\pi_i \cdot q_{ij}}{\pi_j \cdot q_{ji}}</math>,</li> <li><math>\pi_j \cdot q_{ji} &lt; \pi_i \cdot q_{ij}</math>: Setze <math>\alpha_{ji} = 1</math> und <math>\alpha_{ij} = \frac{\pi_j \cdot q_{ji}}{\pi_i \cdot q_{ij}}</math>.</li> </ul> <p>Zusammenfassend haben wir also für <math>i \neq j</math>:</p> $\alpha_{ij} = \min \left\{ 1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \right\}.$ <p><b>Markov Chain Monte Carlo in der Bayes-Statistik</b> <math>\pi_i = P(\Theta = \theta_i   X = x) = \frac{P(X = x   \Theta = \theta_i) P(\Theta = \theta_i)}{P(X = x)}</math></p> <p><b>R-Code</b> <b>Erstellung Markov-Kette mit korrekter Grenzverteilung:</b> A &lt;- rbind(c(1, 1, 0, 1), c(1/2, 1, 1, 0), #A: Matrix der Sprung W'keit c(0, 1/2, 1, 1), c(1/8, 0, 1/2, 1)) Q &lt;- rbind(c(0, 0.5, 0, 0.5), c(0.5, 0, 0.5, 0), c(0, 0.5, 0, 0.5), c(0.5, 0, 0.5, 0))</p>

```

P <- A * Q
P

diag(P) <- 1 - rowSums(P) # Diagonale ergänzen, dass Zeilensum. = 1
P

c(1, 2, 4, 8) %*% P

v <- eigen(t(P))$vectors[,1] # Kontrolle: linker EV von P zum EW 1
v <- Re(v) / sum(Re(v))
v

c(1,2,4,8)/15 # vergleichen

Bestimmen wie viele Zeitschritte es braucht, bis stationäre Verteilung erreicht wird:
Elemente aus Burn-in Phase werden ignoriert.
Beispiel:
Zufallszahlen erzeugen, deren Dichte im Intervall [0, π] die Form der Sinusfunktion hat und die sonst 0 ist. Wir können den Metropolis-Algorithmus wie folgt in R implementieren:
unn_density <- function(x) {
  ifelse (x < 0 | x > pi, 0, sin(x))
}
simu_mcmc <- function(steps, sd = 0.5) {
  mcmc <- rep(NA, steps)
  # Step 1 (Initialisierung)
  x <- 0.2
  for (i in 1:length(mcmc)) {
    # Step 2 (Vorschlag)
    x_star <- rnorm(1, mean = x, sd = sd)
    # Step 3 (Vorschlagsdichte Q kürzt sich raus)
    alpha <- min(1, unn_density(x_star) / unn_density(x))
    x <- sample(c(x_star, x), prob = c(alpha, 1-alpha), size = 1)
    mcmc[i] <- x #Wir merken uns den Zustand
  }
  mcmc
}

Beispiel Satz von Bayes:
set.seed(2)
# Wir halten es für möglich, dass Kopf häufiger vorkommt, und nehmen folgende A-Priori-Verteilung an
prior <- function(theta) {
  ifelse(theta < 0.8, theta, 0.8 + (0.8-theta))
}

# p(X | Theta) = Wahrscheinlichkeit, dass man Kopf wirft

```

```

likelihood <- function(theta) dbinom(x = 2, size = 4, prob = theta)

theta <- 0.5 # Wir starten in einem beliebigen Zustand
mcmc <- rep(0, 100000)
mcmc[1] <- 100 * theta + 1 # von 1 bis 101
pold <- prior(theta) * likelihood(theta)
for (t in 2:length(mcmc)){
  if (mcmc[t-1] == 1) { # Vorschlag eines neuen Zustandes
    mcnew <- sample(1:2,1)
  } else if (mcmc[t-1] == 101) {
    mcnew <- sample(100:101,1)
  } else {
    mcnew <- sample(c(mcmc[t-1]-1,mcmc[t-1]+1),1)
  }
  theta_new <- (mcnew - 1) / 100
  pnew <- prior(theta_new) * likelihood(theta_new)
  alpha_ij <- pnew / pold
  if (runif(1) < alpha_ij) { # Annahmew'keit min(alpha_ij, 1)
    pold <- pnew
    mcmc[t] <- mcnew
  } else{
    mcmc[t] <- mcmc[t-1]
  }
}

Beispiel R-Code Histogramm Burn-In etc. evtl. auch auf R Skript

```

**Zeitkontinuierliche Markov-Prozesse**

Kontinuierlicher Zeit	Zeitkontinuierlicher Markov-Prozess:
	<ul style="list-style-type: none"> <li>- Zeitparameter t ist konstant</li> <li>- Zustandsraum weiterhin diskret und endlich</li> <li>- Gedächtnislosigkeit, jedoch ohne festgelegte Folge vom betrachteten Zeitpunkt (hängt lediglich vom momentanen Zustand s ab)</li> </ul> $X(t), t \geq 0 = P(X(t+s) = j   X(s) = i)$ <ul style="list-style-type: none"> <li>- Prozess homogen (nicht von der Zeit abhängig)</li> <li>- <math>P(X(t+s) = j   X(s) = i) = P(X(t) = j   X(0) = i)</math></li> <li>- Beschreibung des Markov-Prozesses: Angabe der Sprungwahrscheinlichkeit &amp; Angabe Wartezeitparameter</li> </ul> <p><b>Übergangswahrscheinlichkeit <math>p_{ij}</math> + Wartezeitparameter <math>\lambda</math></b>  <math>p_{ii} = 0 \rightarrow</math> für alle i, wenn der Zustand gewechselt wird, kann nicht im selben Zustand verblieben werden.            Wartezeitparameter / Rate mit der i verlassen wird:  <math>\lambda_i = \frac{1}{E(T_i)}</math> (Inverse der mittleren Wartezeit)</p>

**Aufenthaltsdauer / Wartezeitverteilung**

Welche Verteilung hat die Aufenthaltsdauer in einem bestimmten Zustand?

- Annahme: Seit dem Sprung in den jetzigen Zustand ist eine Zeit  $t > 0$  vergangen
- Wahrscheinlichkeit, mit der man Zustand verlässt, ist unabhängig von der Zeit  $t$

$$\frac{P(t < T \leq t + \Delta t)}{P(T > t)}$$

- Für kleine  $\Delta t \rightarrow c\Delta t$
- Exponentialverteilten Zufallszahlen  $\exp(c)$
- $E(T) = \frac{1}{\lambda} \rightarrow c = \lambda$
- Nach bestimmter Wartezeit hat die noch verbleibende Wartezeit dieselbe Verteilung wie die Wartezeit

→ Übergangsrate  $r_{ij} = p_{ij}\lambda_i$

**Simulation:**

1. Startzustand  $x_0 = i$  &  $t_0 = 0$
2. Zufällige Wartezeit  $T_i$  auf den ersten Sprung  $Exp(\lambda_i)$ -verteilte Zufallszahl
3. Zustand  $x_i$  nach erstem Sprung gem. Wahrscheinlichkeiten  $p_{ij}$  in der  $i$ -ten Zeile der Übergangsmatrix
4. Nach  $k$  Sprüngen im Zustand  $j$ : Wartezeit  $T_{k+1}$  auf Sprung  $k + 1$  wieder als  $Exp(\lambda_j)$ -verteilte Zufallszahl

**R-Code:**

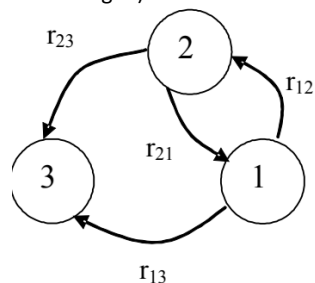
Paket: markovchain  
rctmc() -> für Simulation

**Ratenmatrix R**

Alle Übergangsraten zusammen ergeben Ratenmatrix R.  
Hauptdiagonalelemente von R = 0

**Ratendiagramm:**

Übergänge = 0 werden weggelassen (keine Pfeile, welche auf sich selbst zeigen)



**Generatormatrix Q**

Diagonalelemente der Ratenmatrix durch  $(-\lambda_i)$  ersetzen

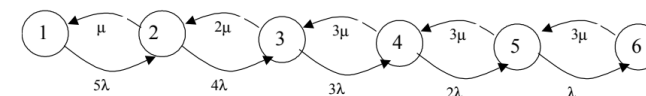
- In jeder Zeile ist die Summe der Matrixelemente = 0

$$Q = \begin{pmatrix} -\lambda & \lambda \\ \mu & -\mu \end{pmatrix}$$

**Beispiel:**

- fünf identischen, unabhängigen Maschinen
- Ausfallrate  $\lambda$  / Reparaturrate  $\mu$
- 3 Monteure; an einer defekten Maschine kann jeweils nur ein Monteur arbeiten.
- Zustand 1: alle Maschinen intakt
- Zustand 2: ist eine Maschine defekt
- Zustand 6: alle fünf Maschinen defekt

Das Ratendiagramm sieht wie folgt aus:



Die Ratenmatrix:

$$\begin{pmatrix} 0 & 5\lambda & 0 & 0 & 0 & 0 \\ \mu & 0 & 4\lambda & 0 & 0 & 0 \\ 0 & 2\mu & 0 & 3\lambda & 0 & 0 \\ 0 & 0 & 3\mu & 0 & 2\lambda & 0 \\ 0 & 0 & 0 & 3\mu & 0 & \lambda \\ 0 & 0 & 0 & 0 & 3\mu & 0 \end{pmatrix}$$

Die Generatormatrix:

$$\begin{pmatrix} -5\lambda & 5\lambda & 0 & 0 & 0 & 0 \\ \mu & -\mu - 4\lambda & 4\lambda & 0 & 0 & 0 \\ 0 & 2\mu & -2\mu - 3\lambda & 3\lambda & 0 & 0 \\ 0 & 0 & 3\mu & -3\mu - 2\lambda & 2\lambda & 0 \\ 0 & 0 & 0 & 3\mu & -3\mu - \lambda & \lambda \\ 0 & 0 & 0 & 0 & 3\mu & -3\mu \end{pmatrix}$$

Wie gross ist die Wahrscheinlichkeit, vom Zustand 2 aus in den Zustand 3 zu wechseln?

$$r_{ij} = \lambda_i \cdot p_{ij}$$

$$p_{23} = \frac{r_{23}}{\lambda_2} = \frac{6}{8} = 0.75$$

**R-Code:**

**Welchen Anteil der Zeit ist das Telefon besetzt?**

```
Q <- matrix(c(-0.5, 0.5, # Besetzt: 30/60
             1/3, -1/3), ncol = 2, byrow = T) # nicht Besetzt: 20/60
es <- eigen(t(Q))
ev <- es$vectors[,2]
ev / sum(ev) #0.4 0.6
```

```
library(expm)
expm(Q * 1000) #Nur zur Probe
```

	<p><b>Q, U, R</b></p> <pre>Q &lt;- matrix(c(-5, 1, 3, 1,               10, -11, 0, 1,               0, 0, -3, 3,               5, 5, 5, -15), ncol = 4, byrow = T) es &lt;- eigen(t(Q)) ev &lt;- es\$eigenvalues[,4]/sum(es\$eigenvalues[,4]) ev</pre> <p># Anzahl der Zustandswechsel</p> <pre>U &lt;- matrix(1, nrow = 4, ncol = 4) diag(U) &lt;- 0 U</pre> <p># Rekonstruktion von R aus Q</p> <pre>R &lt;- Q diag(R) &lt;- 0</pre> <p># Berechnung der Kosten</p> <pre>ev %*% rowSums(R*U)</pre>
<p><b>Zeitabhängige Zustandsverteilungen</b></p>	<p><b>Teilchenmodell</b></p> <ul style="list-style-type: none"> <li>- Voraussetzungen: Homogen, diskreter Zustandsraum &amp; durch Ratenmatrix beschrieben</li> <li>- Startzustand: Sehr grosse Anzahl M Teilchen gem. Anfangsverteilung zur Zeit 0 auf die einzelnen Zustände verteilen. Im Zustand i befindet sich: <math>m_i(0) = M\pi_i(0)</math></li> <li>- Dynamik: Untersuchung der Veränderung der Anzahl Teilchen im Zustand 1 über einen kleinen Zeitschritt. Bilanzgleichung (Zu- &amp; Abfluss der Teilchen): <math>\Delta t(q_{21}m_2(0) + q_{31}m_3(0) + \dots + q_{N1}m_N(0) + q_{11}m_1(0))</math></li> <li>- Grenzübergang zu infinitesimalen Zeitschritten: Differentialgleichung: <math>\frac{d}{dt} \vec{\pi}(t) = \vec{\pi}(t) Q \rightarrow \vec{\pi}(t) = \vec{\pi}(0) \exp(Q t)</math></li> <li>- Andere Möglichkeit: <math>\vec{\pi}(t) = \vec{\pi}(0) * \left( I + Q * \left( \frac{t}{n} \right)^n \right)</math></li> </ul> <p>Grenzübergang: <math>\vec{\pi}(t) = \vec{\pi}(0) \exp(Q t)</math></p>

	<p><b>Beispiel / R-Code:</b></p> <p><b>Aufgabe:</b> Gegeben sei eine Generatormatrix</p> $Q = \begin{pmatrix} -4 & 3 & 1 \\ 2 & -8 & 6 \\ 0 & 1 & -1 \end{pmatrix} \frac{1}{\text{Minuten}}$ <p>Plotten Sie <math>\vec{\pi}(t)</math> in Abhängigkeit von t, wenn Sie im Zustand 1 starten.</p> <p>Matrixexponentialfunktion: Paket expm, Funktion expm( )</p> <pre>library(expm) Q &lt;- matrix( c (-4 , 3 , 1 , 2 , -8 , 6 , 0 , 1 , -1 ), ncol=3 , byrow=T) time &lt;- seq(0, 3, 0.05) vert &lt;- t(sapply(time, function(z) c(1,0,0) %*% expm(Q*z) )) matplot(time, vert, type="l", lwd=2, xlab="Zeit [Minuten]",         ylab="pi") grid()</pre>
<p><b>Langzeitverhalten eines Markov-Prozesses</b></p>	<p><b>Grenzwertverhalten</b></p> <ol style="list-style-type: none"> <li>1. Unter welchen Umständen existiert dieser Grenzwert?</li> <li>2. Ist dieser Grenzwert eindeutig?</li> <li>3. Wie kann der Grenzwert berechnet werden?</li> </ol> <p><b>Voraussetzungen:</b></p> <ul style="list-style-type: none"> <li>- Asymptotische Verteilung / irreduzibel</li> <li>- Eindeutige stationäre Verteilung (ändert sich nicht mehr)</li> </ul> <p><b>Bilanzgleichung:</b></p> <ul style="list-style-type: none"> <li>- In jeden Zustand fließen genauso viele Teilchen hinein wie hinaus</li> <li>- <math>\vec{\pi} * Q = \vec{0}</math></li> <li>- Keine eindeutige Lösung</li> <li>- Durch Normierung auf Gesamtwahrscheinlichkeit 1 ergibt sich stationäre Verteilung</li> </ul> <p><b>Asymptotische Verteilung:</b></p> <ul style="list-style-type: none"> <li>- t diskret: linker EV zum EW 1 von P bzw. rechter EV zum EW 1 von P'</li> <li>- t kontinuierlich: linker EV zum EW 0 von Q bzw. rechter EV zum EW 0 von Q'</li> </ul> <p><b>Ableitung durch Eigenwertanalyse:</b></p> <p><b>Aufgabe:</b> Gegeben sei eine Generatormatrix</p> $Q = \begin{pmatrix} -4 & 3 & 1 \\ 2 & -8 & 6 \\ 0 & 1 & -1 \end{pmatrix} \frac{1}{\text{Minuten}}$ <ul style="list-style-type: none"> <li>• Warum gibt es eine eindeutige Grenzverteilung?</li> <li>• Bestimmen Sie die Grenzverteilung.             <ul style="list-style-type: none"> <li>➔ Prozess ist irreduzibel, daher gibt es eine eindeutige Grenzverteilung</li> <li>➔ Bestimmung der Grenzverteilung:</li> </ul> </li> </ul>

	<p><b>R-Code:</b></p> <pre>es &lt;- eigen(t(Q)) vv &lt;- es\$vector[,3] ev &lt;- vv/sum(vv) ev  matplot(time, vert, type="l", lwd=2, xlab="Zeit [Minuten]", ylab="pi") grid() abline(h = ev, col = 1:3)</pre>
<b>Kostenmodelle asymptotisch</b>	<p>Unendlicher Zeithorizont</p> <p><b>Stationärer Grenzfall:</b></p> <p>Mittlere Kosten: <math>E\left(\frac{dK_t}{dt}\right) = \vec{\pi} * \vec{c}'</math> (<math>c'</math> = Spaltenvektor)</p> <p><math>c</math> = Kosten pro Zeit</p> <p>Asymptotische Durchschnittskosten (Zustandskosten diskret und kontinuierlich): <math>\vec{\pi} * \vec{c}'</math></p> <p>Bei Übergangskosten <math>c'</math> gegeben durch Zeilensummen von</p> <ul style="list-style-type: none"> <li><math>P \circ U</math> (t diskret)</li> <li><math>R \circ U</math> (t kontinuierlich)</li> </ul>

**Punktprozesse**

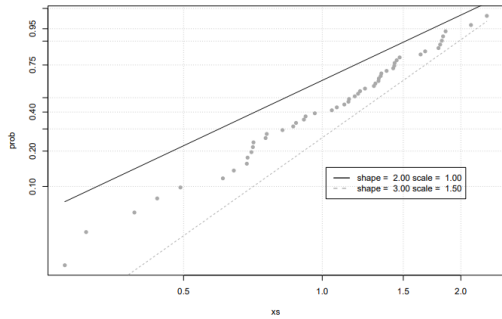
<b>Abfolge von einzelnen Ereignissen</b>	<p><b>Hierarchie:</b></p> <ul style="list-style-type: none"> <li>- Punktprozesse (Folge von zufälligen Wartezeiten)</li> <li>- Erneuerungsprozesse (Wartezeiten sind iid)</li> <li>- Poisson-Prozesse (Wartezeiten sind iid-exponentialverteilt)</li> </ul>
<b>Einfache Punktprozesse</b>	<p><u>Nur eine Sorte von (gleichartigen) Ereignissen</u></p> <ul style="list-style-type: none"> <li>- Eintrittszeitpunkt (nur 1 Mal)</li> <li>- Kontinuierliche Zeit</li> <li>- Wahrscheinlichkeit, dass 2 Ereignisse exakt zum gleichen Zeitpunkt eintreffen: 0</li> </ul> <p>Beschreibung durch:</p> <ul style="list-style-type: none"> <li>- Zwischenankunftszeit / Wartezeit: Aussage über Zeit zwischen zwei Ereignissen <math>T_n = S_n - S_{n-1}</math> (Zufallsvariable)</li> <li>- Anzahl Ereignisse <math>N(t)</math> / Zählprozess: Zählen wie oft eine Ankunft stattgefunden hat <ul style="list-style-type: none"> <li>- Monoton steigend, Werte sind ganzzahlig</li> </ul> </li> </ul> <p>➔ Zähl- &amp; Punktprozesse: Äquivalente Beschreibungen</p>

	<p>Ein <b>Punktprozess</b> (a) mit Eintrittszeiten <math>S_1, S_2, \dots</math> und Zwischenankunftszeiten <math>T_1, T_2, \dots</math> sowie der zugehörige <b>Zählprozess</b> (b):</p> <p>Es gibt Prozesse, bei denen eine stochastische Abhängigkeit zwischen verschiedenen Zwischenankunftszeiten besteht.</p>
<b>Poisson-Prozess</b>	<p>Art von Erneuerungsprozess</p> <p><b>Voraussetzungen:</b></p> <ul style="list-style-type: none"> <li>- Rate <math>\lambda &gt; 0</math>, wenn Zeiten zwischen den Ereignissen iid-exponentialverteilt sind</li> <li>- Alle Wartezeiten mit derselben Verteilung <math>\exp(\lambda)</math></li> <li>- Exponentialverteilte Wartezeit ist eine starke Annahme, gleichbedeutend mit Gedächtnislosigkeit des Prozesses (Zukunft hängt nicht von der bisherigen Wartezeit ab)</li> </ul> <p><b>Verteilungsfunktion der Zufallsvariable T (Wartezeiten):</b></p> <ul style="list-style-type: none"> <li>- Wahrscheinlichkeit, dass Zufallsvariable X einen Wert kleiner gleich x annimmt</li> </ul> <p>Die <b>Verteilungsfunktion</b> einer <math>Exp(\lambda)</math>-Verteilung ist gegeben als</p> $F_T(t) = P(T \leq t) = \begin{cases} 1 - e^{-\lambda t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$ <p>Ihre Ableitung, die <b>Dichtefunktion</b> ist gegeben durch:</p> $f_T(t) = \begin{cases} \lambda e^{-\lambda t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$ <p>Der <b>Erwartungswert</b> ist</p> $E(T) = \int_{-\infty}^{\infty} t f(t) dt = \dots = \frac{1}{\lambda}$ <p><b>Wie wahrscheinlich ist es, dass innerhalb eines nächsten kleinen Zeitschritts <math>\Delta t</math> das nächste Ereignis stattfindet?</b></p> <p>➔ Gesucht: Bedingte Wahrscheinlichkeit</p> <ul style="list-style-type: none"> <li>- Wenn <math>\Delta t \rightarrow 0 \Rightarrow \lambda</math></li> <li>- Für kleine <math>\Delta t</math> gilt: <math>\lambda \Delta t</math></li> </ul> <p>Konstante Lambda beschreibt die Rate, mit der etwas passiert, falls noch kein Ereignis stattgefunden hat</p> <p>Sei <math>(N(t))_{t \geq 0}</math>: <math>P(N(t) = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}</math></p>

	<p><b>Addition von Ereignisströmen:</b>                  Unabhängige Poisson-Prozesse mit Raten <math>\lambda_1</math> und <math>\lambda_2</math>                  Zählprozess: <math>N(t) = N_1(t) + N_2(t)</math>                  Poisson-Prozess: <math>\lambda = \lambda_1 + \lambda_2</math>                  Verteilungsfunktion: <math>F_T(t) = 1 - \exp(-(\lambda_1 + \lambda_2)t)</math></p> <p><b>Ausdünnen eines Poisson-Prozesses</b>  <math>\lambda_1 = p * \lambda</math></p> <p>Zählung der Ereignisse, welche nicht in <math>N(t)</math> registriert sind:  <math>\lambda_2 = (1 - p) * \lambda</math>                  Ereignisse aus <math>N_2(t)</math> = Ereignisse welche nicht in <math>N_1(t)</math> enthalten sind (Prozesse sind unabhängig)</p> <p>Simulation eines PP mit festem Zeithorizont</p> <ol style="list-style-type: none"> <li>1. Ziehen der auftretenden Ereignisse im Intervall <math>(0, T]</math> aus einer Poissonverteilung</li> <li>2. Ziehen von <math>k</math> Zufallszahlen unabhängig aus einer Gleichverteilung <math>(0, T]</math></li> <li>3. Gleichverteilten Zufallszahlen ordnen und in die Eintrittszeitpunkte einsetzen</li> </ol> <p><b>R-Code:</b>  <b>Wie gross ist die Wahrscheinlichkeit pFull, dass ein Bus mit 15 Sitzplätzen nicht alle Wartenden mitnehmen kann?</b>  <code>pfull &lt;- 1-ppois(15, lambda = 30*0.5)</code>  <code>pfull</code></p>
<p><b>Erneuerungsprozesse</b></p>	<p>Beschreiben Erneuerung von ausfallenden Bauteilen (wie viele Teile waren bisher defekt?) -&gt; Lebensdauerverteilung</p> <p>Wartezeit für nächstes Ereignis: Lebensdauer des Bauteils</p> <p>Erwartete Restlebensdauer: Hängt davon ab, wie lange dieses schon im Einsatz ist (Exp.verteilung nicht geeignet, da gedächtnislos)</p> <p>Verteilungsfunkt. Lebensdauer: W'keit, dass ein Bauteil bis zum Zeitpunkt <math>t</math> kaputt geht:  <math>S(t) = 1 - F(t)</math></p> <p>W'keit kaputt in Zeitintervall: <math>\frac{F(t + \Delta t) - F(t)}{1 - F(t)}</math></p> <p>Ausfallrate <math>h(t)</math>: <math>\frac{\frac{F(t + \Delta t) - F(t)}{\Delta t}}{1 - F(t)}</math></p> <p>Bedingte Dichte: <math>h(t) \Delta t</math></p> <p><b>Weibull-Verteilung:</b></p>

	<p>Überlebensfunktion <math>S(t)</math> <math>S(t) = \exp(-(\lambda t)^\beta), t \geq 0</math></p> <p>Verteilungsfunktion <math>F(t)</math> <math>F(t) = 1 - \exp(-(\lambda t)^\beta)</math></p> <p>Dichte <math>f(t)</math> <math>f(t) = \lambda \beta (\lambda t)^{\beta-1} e^{-(\lambda t)^\beta}</math></p> <p>Ausfallrate <math>h(t)</math> <math>h(t) = \lambda \beta (\lambda t)^{\beta-1}</math></p> <p>Erwartungswert <math>E(T)</math> <math>E(T) = \frac{1}{\lambda} \Gamma(1 + \frac{1}{\beta})</math></p> <p><math>\beta = 1</math>: Exponentialverteilung  <math>\beta &gt; 1</math>: <math>h(t)</math> ist monoton steigend (Bauteile altern)  <math>\beta &lt; 1</math>: <math>h(t)</math> ist monoton fallend (Kinderkrankheiten nehmen ab)</p> <p><b>Langzeitverhalten von Erneuerungsprozessen</b>                  Gesetz der grossen Zahlen:  <math>\lim_{N \rightarrow \infty} \frac{S_N}{N} = \tau</math></p> <p>Auf Erneuerungsprozess anwenden:                  Dies gilt mit <math>W'</math>keit = 1  <math>\lim_{t \rightarrow \infty} \frac{N(t)}{t} = \frac{1}{\tau}</math> Entspricht der mittleren Rate</p> <p>Für kleine <math>t</math> weicht diese Grösse stark von der Rate 1 ab. Erst für grosse <math>t</math> konvergiert diese Grösse gegen die Rate 1.</p> <p><b>R-Code:</b></p> <p>Paket Renext <code>fweibull()</code> – numerische Schätzung</p> <p>Dichte: <code>dweibull()</code> <code>shape: beta</code></p> <p>Verteilungsfunkt.: <code>pweibull()</code> <code>scale: 1/lambda</code></p> <p>Quantile: <code>qweibull()</code></p> <p>Zufallszahlen: <code>rweibull()</code></p> <p><code>library(car)</code>  <code>set.seed(43627)</code>  <code>x &lt;- rweibull(50, shape = 3, scale = 1.5)</code>  <code>qqPlot(x, distribution = "weibull", shape = 3)</code>  <code>qqPlot(x, distribution = "weibull", shape = 1)</code></p> <p>Weibull-Plot:                  Wenn die Daten aus einer Weibull-Verteilung stammen, liegen die Punkte ungefähr auf einer Geraden. Parameter müssen vorher nicht geschätzt werden.  <code>library(Renext)</code></p> <p><code>weibplot(x)</code>  <code>weibplot(x, shape = c(2, 3), scale = c(1, 1.5))</code></p>
--	---





Bestimmung von Parametern:  
`est <- fweibull(x)`  
`est$estimate`  
`weibplot(x, shape = est$estimate[1], scale = est$estimate[2])`

**Kumulative Prozesse**

Systeme, bei denen Kosten und Gewinne auflaufen

- ➔ Wenn Gewinne als negative Kosten interpretiert werden kann man beides in generalistischem Kostenprozess  $K(t)$  integrieren.  $K(t)$  kann steigen und fallen.
- ➔  $K(t)$  : zeitkontinuierlich und wertekontinuierlich
- ➔ Eine Folge von iid-bivariaten Zufallszahlen

Asymptotische Kostenrate ergibt sich aus:  
 Erwartungswerte der Zykluskosten / Erw.werte der Zykluszeit

$$\lim_{t \rightarrow \infty} \frac{K(t)}{t} = \frac{E(K)}{E(T_a)}$$

stochastische Abhängigkeit der beiden  
Größen hier irrelevant

Grösse könnte auch analytisch berechnet werden.