Kapitel 9

Survival-Analyse

Die Survival-Analyse modelliert Überlebenszeiten (Hosmer Jr, Lemeshow & May, 2008; Klein & Moeschberger, 2003). Diese geben allgemein an, wieviel Zeit bis zum Eintreten eines bestimmten Ereignisses verstrichen ist und sollen hier deshalb gleichbedeutend mit *Ereigniszeiten* sein. Es kann sich dabei etwa um die Zeitdauer handeln, die ein Patient nach einer Behandlung weiter am Leben ist, um die verstrichene Zeit, bis ein bestimmtes Bauteil im Gebrauch einen Defekt aufweist, oder um die Dauer, die ein Kleinkind benötigt, um ein vordefiniertes Entwicklungsziel zu erreichen – z. B. einen Mindestwortschatz besitzt. Bei der Analyse von Überlebenszeiten kann sowohl die Form ihres grundsätzlichen Verlaufs von Interesse sein, als auch inwiefern ihr Verlauf systematisch von Einflussgrößen abhängt.

Die folgenden Auswertungen verwenden Funktionen des Pakets `survival`, das im Basisumfang von R enthalten ist. Seine Anwendung wird vertiefend in Harrell Jr (2015) behandelt.

9.1 Verteilung von Ereigniszeiten

Überlebenszeiten lassen sich äquivalent durch verschiedene Funktionen beschreiben, die jeweils andere Eigenschaften ihrer Verteilung hervortreten lassen: Die Überlebenszeit selbst sei T mit Werten ≥ 0 und einer – hier als stetig vorausgesetzten – Dichtefunktion $f(t)$.[1] Die zugehörige Verteilungsfunktion $F(t) = P(T \leq t)$ liefert die Wahrscheinlichkeit, mit der T höchstens den Wert t erreicht. Die monoton fallende Survival-Funktion $S(t) = P(T > t) = 1 - F(t)$ gibt an, mit welcher Wahrscheinlichkeit die Überlebenszeit größer als t ist. Dabei wird $S(0) = 1$ vorausgesetzt, zum Zeitpunkt $t = 0$ soll das Ereignis also noch nicht eingetreten sein. Schließlich drückt die *Hazard*-Funktion $\lambda(t)$ die unmittelbare Ereignisrate zum Zeitpunkt t aus.

$$\lambda(t) = \lim_{\Delta_t \to 0^+} \frac{P(t \leq T < t + \Delta_t \,|\, T \geq t)}{\Delta_t} = \frac{P(t \leq T < t + \Delta_t)/\Delta_t}{P(T > t)} = \frac{f(t)}{S(t)}, \quad t \geq 0$$

$\lambda(t)$ ist bei kleiner werdender Intervallbreite Δ_t der Grenzwert für die bedingte Wahrscheinlichkeit pro Zeiteinheit, dass das Ereignis unmittelbar eintritt, wenn es bis zum Zeitpunkt t noch nicht eingetreten ist. Bezeichnet $\#\text{Personen} \,|\, \text{Ereignis} \in [t, t + \Delta_t)$ die Anzahl der Personen, bei denen das Ereignis im Intervall $[t, t + \Delta_t)$ eintritt und $\#\text{Personen} \,|\, \text{Ereignis} \geq t$ die Anzahl der

[1] $f(t) = \lim_{\Delta_t \to 0^+} \frac{P(t \leq T < t + \Delta_t)}{\Delta_t}$ in Abhängigkeit von der Zeit t und der Intervallbreite Δ_t.

Personen, bei denen das Ereignis zum Zeitpunkt t noch eintreten kann, lässt sich das hazard auf empirischer Ebene wie folgt formulieren:[2]

$$\hat{\lambda}(t) = \frac{\#\text{Personen} \,|\, \text{Ereignis} \in [t, t + \Delta_t)}{\#\text{Personen} \,|\, \text{Ereignis} \geq t} \cdot \frac{1}{\Delta_t}$$

$\hat{\lambda}(t)$ gibt also an, bei welchem Anteil verbleibender Personen ohne Ereignis bis zum Zeitpunkt t das Ereignis pro Zeiteinheit eintritt. Bei Werten der monoton steigenden kumulativen Hazard-Funktion $\Lambda(t) = -\ln S(t)$ handelt es sich um das bis zum Zeitpunkt t kumulierte Risiko, dass sich das Ereignis unmittelbar ereignet. Umgekehrt gilt $S(t) = \mathrm{e}^{-\Lambda(t)}$.

Hazard- und Survival-Funktion bedingen einander. Nimmt man eine über die Zeit konstante Ereignisrate $\lambda(t) = \frac{1}{b}$ an (wie etwa beim radioaktiven Zerfall), impliziert dies eine exponentiell verteilte Überlebenszeit T (Abschn. 9.5). Die bedingte Wahrscheinlichkeit, dass ein noch nicht eingetretenes Ereignis unmittelbar auf t folgt, wäre damit unabhängig von der bereits verstrichenen Zeit t. Mit der Annahme, dass die logarithmierte Ereignisrate linear von t abhängt ($\ln \lambda(t) = \alpha + \rho\, t$ mit y-Achsenabschnitt α und Steigung ρ), ergibt sich entsprechend eine Gompertz-Verteilung von T. Analog führt die Annahme, dass die logarithmierte Ereignisrate linear mit der logarithmierten Zeit zusammenhängt ($\ln \lambda(t) = \alpha + \rho \ln t$), zu einer Weibull-Verteilung von T (Abschn. 9.5). Bei einem positivem ρ würde in beiden Fällen das hazard mit der Zeit ansteigen, was oft in Situationen angemessen ist, in denen das Eintreten des Ereignisses mit kontinuierlichen Reifungs- oder Abnutzungsprozessen zusammenhängt.

9.2 Zensierte und gestutzte Ereigniszeiten

Um festzustellen, wann ein Zielereignis eintritt, werden die untersuchten Objekte über eine gewisse Zeit hinweg beobachtet – etwa wenn bei aus einer stationären Behandlung entlassenen Patienten mit Substanzmissbrauch erhoben wird, ob sich innerhalb eines Zeitraums ein Rückfall ereignet. Meist weisen empirisch erhobene Überlebenszeiten dabei die Besonderheit auf, dass von einigen Beobachtungseinheiten die Zeit bis zum Eintreten des Ereignisses unbekannt bleibt, was spezialisierte statistische Modelle notwendig macht.

Der Erhebungszeitraum ist oft begrenzt, so dass nicht für alle Untersuchungseinheiten das Ereignis auch tatsächlich innerhalb des Beobachtungszeitraums auftritt. Für solche *rechts-zensierten* Daten ist also nur bekannt, dass die Überlebenszeit den letzten Beobachtungszeitpunkt überschreitet, nicht aber ihr exakter Wert. Eine andere Ursache für rechts-zensierte Daten kann ein frühzeitiger dropout aus der Studie nach einem Umzug oder bei schwindender Motivation zur Teilnahme sein. *Links-zensierte* Daten entstehen, wenn das Ereignis bekanntermaßen bereits an einem unbekannten Zeitpunkt vor Beginn der Erhebung eingetreten ist. Daten werden als *links-gestutzt* bezeichnet, wenn sich das Ereignis bei manchen potentiellen Beobachtungseinheiten bereits vor Erhebungsbeginn ereignet und sie deswegen nicht mehr in der Studie berücksichtigt werden können. Während die Häufigkeit zensierter Beobachtungen in der Stichprobe bekannt ist, fehlt über gestutzte Daten jede Information.

[2] $\#\text{Personen} \,|\, \text{Ereignis} \geq t$ ist die Größe des *risk set* bzw. die Anzahl der Beobachtungsobjekte *at risk* zum Zeitpunkt t.

Wichtig für die Survival-Analyse ist die Annahme, dass der zur Zensierung führende Mechanismus unabhängig von Einflussgrößen auf die Überlebenszeit ist, Beobachtungsobjekte mit zensierter Überlebenszeit also kein systematisch anderes hazard haben. Diese Bedingung wäre etwa dann erfüllt, wenn zensierte Daten dadurch entstehen, dass eine Studie zu einem vorher festgelegten Zeitpunkt endet, bis zu dem nicht bei allen Untersuchungseinheiten das Ereignis eingetreten ist. Bewirkt eine Ursache dagegen sowohl das nahe bevorstehende Eintreten des Ereignisses selbst als auch den Ausfall von Beobachtungsmöglichkeiten, wäre die Annahme nicht-informativer Zensierung verletzt. So könnte eine steigende zeitliche Beanspruchung im Beruf bei Patienten mit Substanzmissbrauch einerseits dazu führen, dass die Bereitschaft zur Teilnahme an Kontrollterminen sinkt, andererseits könnte sie gleichzeitig die Wahrscheinlichkeit eines Rückfalls erhöhen. Wenn selektiv Beobachtungseinheiten mit erhöhtem hazard nicht mehr beobachtet werden können, besteht die Gefahr verzerrter Schätzungen des Verlaufs der Überlebenszeiten.

9.2.1 Zeitlich konstante Prädiktoren

Survival-Daten beinhalten Angaben zum Beobachtungszeitpunkt, zu den Prädiktoren der Überlebenszeit sowie eine Indikatorvariable dafür, ob das Ereignis zum angegebenen Zeitpunkt beobachtet wurde. Für die Verwendung in späteren Analysen sind Beobachtungszeitpunkt und Indikatorvariable zunächst in einem Survival-Objekt zusammenzuführen, das Informationen zur Art der Zensierung der Daten berücksichtigt. Dies geschieht für potentiell rechts-zensierte Daten mit `Surv()` aus dem Paket `survival` in der folgenden Form:

```
> Surv(⟨Zeitpunkt⟩, ⟨Status⟩)
```

Als erstes Argument ist ein Vektor der Zeitpunkte $t_i > 0$ zu nennen, an denen das Ereignis bei den Objekten i eingetreten ist. Bei rechts-zensierten Beobachtungen ist dies der letzte bekannte Zeitpunkt, zu dem das Ereignis noch nicht eingetreten war. Die dabei implizit verwendete Zeitskala hat ihren Ursprung 0 beim Eintritt in die Untersuchung. Das zweite Argument ist eine numerische oder logische Indikatorvariable, die den Status zu den genannten Zeitpunkten angibt – ob das Ereignis also vorlag (1 bzw. `TRUE`) oder nicht (0 bzw. `FALSE` bei rechts-zensierten Beobachtungen).[3]

Für die folgende Simulation von Überlebenszeiten soll eine Weibull-Verteilung mit Annahme proportionaler hazards bzgl. der Einflussfaktoren (Abschn. 9.4) zugrunde gelegt werden (Abb. 9.1). Dafür sei der lineare Effekt der Einflussgrößen durch $\boldsymbol{X\beta} = \beta_1 X_1 + \cdots + \beta_j X_j + \cdots + \beta_p X_p$ gegeben (ohne absoluten Term β_0, s. Abschn. 12.9.1). Hier soll dafür ein kontinuierlicher Prädiktor sowie eine kategoriale UV mit 3 Stufen verwendet werden, wobei beide Variablen nicht über die Zeit variieren. Zusätzlich sei die Schichtung hinsichtlich des Geschlechts berücksichtigt.

```
> N   <- 180                                          # Anzahl Personen
> P   <- 3                                            # Anzahl Stufen UV
> sex <- factor(sample(c("f", "m"), N, replace=TRUE))     # Geschlecht
> X   <- rnorm(N, 0, 1)                               # kont. Prädiktor
> IV  <- factor(rep(LETTERS[1:P], each=N/P))          # UV Faktor
```

[3] Für Intervall-zensierte Daten vgl. `?Surv`. Vergleiche Abschn. 9.2.2 für zeitabhängige Prädiktoren und Fälle, in denen mehrere Ereignisse pro Beobachtungsobjekt möglich sind.

```
# Effekte der UV-Stufen: 1. Stufe = baseline -> Effekt 0
> IVeff <- c(0, -1, 1.5)

# zusammengefasster Effekt der Einflussgrößen mit zufälligem Fehler
> Xbeta <- 0.7*X + IVeff[unclass(IV)] + rnorm(N, 0, 2)
```

Weiter sei $U \sim \mathcal{U}(0,1)$ eine gleichverteilte Zufallsvariable auf dem Intervall $[0,1]$. Weibull-verteilte Überlebenszeiten können dann als Realisierung von $T = (-\ln(U)\, b^a\, \mathrm{e}^{-\boldsymbol{X}\boldsymbol{\beta}})^{\frac{1}{a}}$ simuliert werden (Abschn. 9.5). Die hier getroffene Wahl $a = 1.5$ führt dazu, dass $\ln \lambda(t)$ linear mit $\ln t$ ansteigt. Die Simulation von Überlebenszeiten mit anderen kumulativen Hazard-Funktionen $\Lambda(t)$ unter Annahme proportionaler hazards erfolgt allgemein mit $\Lambda^{-1}(-\ln(U)\, \mathrm{e}^{-\boldsymbol{X}\boldsymbol{\beta}})$ (Bender, Augustin & Blettner, 2005).

```
# Weibull-Verteilung zur Charakterisierung des baseline-hazards
> weibA <- 1.5                                          # Formparameter
> weibB <- 100                                          # Skalierungsparameter
> U     <- runif(N, 0, 1)                               # gleichverteilte Var

# Überlebenszeiten - aufrunden für t > 0
> eventT <- ceiling((-log(U)*(weibB^weibA)*exp(-Xbeta))^(1/weibA))
> obsLen <- 120                                         # Beobachtungsdauer

# stelle kumulierte Verteilung der Überlebenszeiten dar
> plot(ecdf(eventT), xlim=c(0, 200), main="Kumulative
+      Überlebenszeit-Verteilung", xlab="t", ylab="F(t)")

> abline(v=obsLen, col="blue", lwd=2)                   # Untersuchungsende
> text(obsLen-5, 0.2, adj=1, labels="Ende Beobachtungszeit")
```

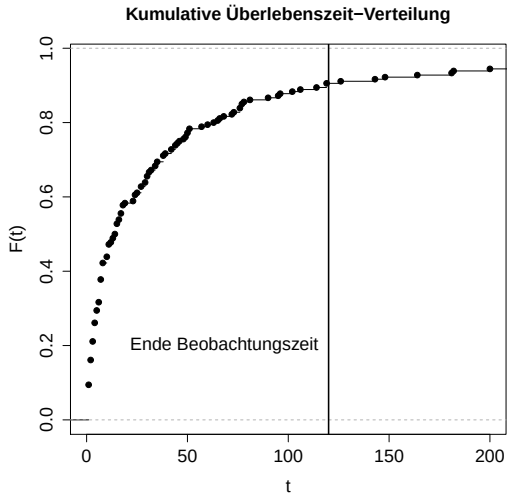

Abbildung 9.1: Kumulative Verteilung simulierter Survival-Daten mit Weibull-Verteilung

Der zur rechts-Zensierung führende Prozess soll hier ausschließlich das geplante Ende des Beobachtungszeitraums sein. Alle nach dem Endpunkt der Erhebung liegenden Überlebenszeiten werden daher auf diesen Endpunkt zensiert. Entsprechend wird das Ereignis nur beobachtet, wenn die Überlebenszeit nicht nach dem Endpunkt liegt.

```
> censT  <- rep(obsLen, N)                          # Zensierungszeit
> obsT   <- pmin(eventT, censT)                     # zensierte Ü-Zeit
> status <- eventT <= censT                         # Ereignis-Status
> dfSurv <- data.frame(obsT, status, sex, X, IV)    # Datensatz
> library(survival)                                 # für Surv()
> Surv(obsT, status)                                # Survival-Objekt
[1] 63 25 73 120+ 4 39 4 1 10 11 2 120+ 7 29 95     # ...
```

In der (hier gekürzten) Ausgabe des mit `Surv()` gebildeten Survival-Objekts sind zensierte Überlebenszeiten durch ein + kenntlich gemacht.

9.2.2 Daten in Zählprozess-Darstellung

Überlebenszeiten lassen sich samt der zugehörigen Werte für relevante Prädiktoren auch in der *Zählprozess*-Darstellung notieren. Für jedes Beobachtungsobjekt unterteilt diese Darstellung die Zeit in diskrete Intervalle (Start, Stop], für die jeweils angegeben wird, ob sich ein Ereignis im Intervall ereignet hat. Die Intervalle sind links offen und rechts geschlossen, zudem muss Stop > Start gelten, so dass keine Intervalle der Länge 0 auftauchen. Der Aufruf von `Surv()` erweitert sich dafür wie folgt:

```
> Surv(⟨Start⟩, ⟨Stop⟩, ⟨Status⟩)
```

⟨`Start`⟩ gibt den Beginn eines Beobachtungsintervalls an. ⟨`Stop`⟩ ist wieder der beobachtete Ereignis-Zeitpunkt bzw. bei rechts-zensierten Daten der letzte bekannte Zeitpunkt, zu dem kein Ereignis vorlag. Intervalle für unterschiedliche Personen dürfen dabei unterschiedliche Grenzen besitzen. Anders als in der in Abschn. 9.2.1 vorgestellten Darstellung kann die hier implizit verwendete Zeitskala flexibel etwa das Alter eines Beobachtungsobjekts sein, die (vom Eintritt in die Untersuchung abweichende) Zeit seit einer Diagnose oder die absolute kalendarische Zeit. Ist das Ereignis im Intervall (Start, Stop] eingetreten, ist ⟨`Status`⟩ gleich 1 (bzw. `TRUE`), sonst 0 (bzw. `FALSE` bei rechts-zensierten Beobachtungen).

Durch die explizite Angabe des Beginns eines Beobachtungszeitraums gibt die Zählprozess-Darstellung Auskunft darüber, vor welchem Zeitpunkt keine Informationen darüber vorliegen, ob sich das Ereignis bereits einmal ereignet hat. Die Darstellungsform ist damit für links-gestutzte Daten geeignet. Tabelle 9.1 zeigt die Darstellung am Beispiel von vier Beobachtungsobjekten aus zwei Gruppen, deren Lebensalter zu Beginn und zu Ende einer zehnjährigen Untersuchung ebenso erfasst wurde wie der Ereignis-Status zu Ende der Untersuchung.

Die Zählprozess-Darstellung erlaubt es auch Erhebungssituationen abzubilden, in denen der Status von Beobachtungsobjekten an mehreren Zeitpunkten t_m ermittelt wird. So könnte etwa bei regelmäßigen Kontrollterminen nach einer Operation geprüft werden, ob eine bestimmte Krankheit wieder diagnostizierbar ist. Der Aufruf von `Surv()` hat dann die Form wie bei links-gestutzten Beobachtungen, wobei pro Beobachtungsobjekt mehrere (Start, Stop] Intervalle

Tabelle 9.1: Zählprozess-Darstellung für vier links-gestutzte und teilweise rechts-zensierte Beobachtungen mit Überlebenszeiten (Lebensalter) $\{47, 38^+, 41, 55\}$ aus zwei Gruppen in einer zehnjährigen Untersuchung

Start	Stop	Gruppe	Status
37	47	Treatment	1
28	38	Treatment	0
31	41	Control	1
45	55	Control	1

vorliegen. Für den Untersuchungszeitpunkt t_m ist dabei ⟨`Start`⟩ der letzte zuvor liegende Untersuchungszeitpunkt t_{m-1}, während ⟨`Stop`⟩ t_m selbst ist. Für den ersten Untersuchungszeitpunkt t_1 ist t_0 der Eintritt in die Untersuchung, etwa das zugehörige kalendarische Datum oder das Alter eines Beobachtungsobjekts.

Wiederkehrende Ereignisse

Da der Ereignis-Status in der Zählprozess-Darstellung am Ende mehrerer Zeitintervalle erfasst werden kann, ist es auch möglich Ereignisse abzubilden, die sich pro Untersuchungseinheit potentiell mehrfach ereignen. Die spätere Analyse wiederkehrender Ereignisse muss dabei berücksichtigen, ob diese unabhängig voneinander eintreten, oder etwa in ihrer Reihenfolge festgelegt sind.

Bei mehreren Beobachtungszeitpunkten muss der Datensatz für die Zählprozess-Darstellung im Long-Format vorliegen, d. h. für jede Untersuchungseinheit mehrere Zeilen umfassen (Abschn. 3.3.9). Pro Beobachtungszeitpunkt t_m ist für jede Untersuchungseinheit eine Zeile mit den Werten t_{m-1}, t_m, den Werten der Prädiktoren sowie dem Ereignis-Status zu t_m in den Datensatz aufzunehmen. Zusätzlich sollte jede Zeile den Wert eines Faktors ⟨`ID`⟩ enthalten, der das Beobachtungsobjekt identifiziert (Tab. 9.2).

Tabelle 9.2: Zählprozess-Darstellung für zwei Beobachtungsobjekte aus zwei Gruppen am Ende von je drei zweijährigen Beobachtungsintervallen mit potentiell mehrfach auftretenden Ereignissen

ID	Start (t_{m-1})	Stop (t_m)	Gruppe	Status
1	37	39	Treatment	0
1	39	41	Treatment	1
1	41	43	Treatment	1
2	28	30	Control	0
2	30	32	Control	1
2	32	34	Control	0

Liegen die Daten im (⟨`Zeitpunkt`⟩, ⟨`Status`⟩)-Format vor, können sie mit `survSplit()` aus

dem Paket `survival` in Zählprozess-Darstellung mit mehreren Zeitintervallen pro Beobachtungsobjekt gebracht werden.

```
> survSplit(⟨Modellformel⟩, cut=⟨Grenzen⟩, start="⟨Name⟩",
+           id="⟨Name⟩", zero=⟨Startzeit⟩, data=⟨Datensatz⟩)
```

Als erstes Argument ist eine Modellformel zu übergeben, deren linke Seite ein mit `Surv()` erstelltes Objekt ist (Abschn. 9.2). Die rechte Seite der Modellformel definiert die Variablen, die im erstellten Datensatz beibehalten werden – meist `.` stellvertretend für alle Variablen des Datensatzes `data`. `cut` legt die linken Grenzen t_{m-1} der neu zu bildenden Intervalle fest, jedoch ohne die erste Grenze t_0.[4] `start` ist der Name der zu erstellenden Variable der linken Intervallgrenzen t_{m-1}. Soll eine Variable hinzugefügt werden, die jedem Intervall das zugehörige Beobachtungsobjekt zuordnet, muss deren Name für `id` genannt werden. Mit `zero` lässt sich in Form eines Vektors oder als – dann für alle Beobachtungen identische – Zahl angeben, was der Zeitpunkt t_0 des Beginns der Beobachtungen ist.

```
> library(survival)                                # für survSplit()
> dfSurvCP <- survSplit(Surv(obsT, status) ~ ., cut=seq(30,90,by=30),
+                       start="start", id="ID", zero=0, data=dfSurv)

# sortiere nach Beobachtungsobjekt und linken Intervallgrenzen
> idxOrd <- order(dfSurvCP$ID, dfSurvCP$start)
> head(dfSurvCP[idxOrd, ], n=7)
  sex          X IV ID start obsT status
1   f -1.3130607  A  1     0   30      0
2   f -1.3130607  A  1    30   60      0
3   f -1.3130607  A  1    60   63      1
4   m -0.1384786  A  2     0   25      1
5   m -0.3846335  A  3     0   30      0
6   m -0.3846335  A  3    30   60      0
7   m -0.3846335  A  3    60   73      1
```

Zeitabhängige Prädiktoren

In der Zählprozess-Darstellung ist es möglich, die Werte von zeitlich variablen Prädiktoren in die Daten aufzunehmen (Tab. 9.3). Eine spätere Analyse setzt dabei voraus, dass der Wert eines zeitabhängigen Prädiktors zum Zeitpunkt t_m nur Informationen widerspiegelt, die bis t_m vorlagen – aber nicht später. Eine zeitlich rückwirkende Kategorisierung von Untersuchungseinheiten in verschiedene Gruppen auf Basis ihres Verhaltens zu Studienende wäre etwa demnach unzulässig. Ebenso sind meist zeitabhängige Variablen problematisch, die weniger Einflussgröße bzw. Prädiktor, sondern eher Effekt oder Indikator eines Prozesses sind, der zu einem bevorstehenden Ereignis führt.

[4] Bei nicht wiederkehrenden Ereignissen ist die Einteilung der Gesamtbeobachtungszeit in einzelne, bündig aneinander anschließende Intervalle beliebig: Die einzelne Beobachtung im (⟨`Zeitpunkt`⟩, ⟨`Status`⟩)-Format `(10, TRUE)` ist sowohl äquivalent zu den zwei Beobachtungen in Zählprozess-Darstellung `(0, 4, FALSE)`, `(4, 10, TRUE)` als auch zu den drei Beobachtungen `(0, 2, FALSE)`, `(2, 6, FALSE)`, `(6, 10, TRUE)`.

Tabelle 9.3: Zählprozess-Darstellung für zeitabhängige Prädiktoren bei drei Beobachtungsobjekten mit Überlebenszeiten $\{3, 4^+, 2\}$

ID	Start (t_{m-1})	Stop (t_m)	Temperatur	Status
1	0	1	45	0
1	1	2	52	0
1	2	3	58	1
2	0	1	37	0
2	1	2	41	0
2	2	3	56	0
2	3	4	57	0
3	0	1	35	0
3	1	2	61	1

9.3 Kaplan-Meier-Analyse

9.3.1 Survival-Funktion schätzen

Die Kaplan-Meier-Analyse liefert als nonparametrische Maximum-Likelihood-Schätzung der Survival-Funktion eine Stufenfunktion $\hat{S}(t)$, deren Stufen bei den empirisch beobachteten Überlebenszeiten liegen. Sie wird samt der punktweisen Konfidenzintervalle mit `survfit()` aus dem Paket `survival` berechnet.

```
> survfit(⟨Modellformel⟩, type="kaplan-meier", conf.type="⟨CI-Typ⟩")
```

Als erstes Argument ist eine Modellformel zu übergeben, deren linke Seite ein mit `Surv()` erstelltes Objekt ist (Abschn. 9.2). Die rechte Seite der Modellformel ist entweder der konstante Term 1 für eine globale Anpassung, oder besteht aus (zeitlich konstanten) Faktoren. In diesem Fall resultiert für jede Faktorstufe bzw. Kombination von Faktorstufen g eine separate Schätzung $\hat{S}_g(t)$. Das Argument `type` ist bei der Voreinstellung `"kaplan-meier"` zu belassen. Mit `conf.type` wird die Transformation für $S(t)$ angegeben, auf deren Basis die nach zugehöriger Rücktransformation gewonnenen Konfidenzintervalle für $S(t)$ konstruiert sind: Mit `"plain"` erhält man Intervalle auf Basis von $S(t)$ selbst. Geeigneter sind oft die durch `"log"` erzeugten Intervalle, die auf dem kumulativen hazard $\Lambda(t) = -\ln S(t)$ basieren. Mit `"log-log"` ergeben sich die Intervalle aus dem logarithmierten kumulativen hazard $\ln \Lambda(t) = \ln(-\ln S(t))$. Auf `"none"` gesetzt unterbleibt die Berechnung von Konfidenzintervallen.

```
# Schätzung von S(t) ohne Trennung nach Gruppen
> library(survival)                              # für survfit()
> KM0 <- survfit(Surv(obsT, status) ~ 1,  type="kaplan-meier",
+                conf.type="log", data=dfSurv)

# Schätzung von S(t) getrennt nach Gruppen
> (KM <- survfit(Surv(obsT, status) ~ IV, type="kaplan-meier",
+                conf.type="log", data=dfSurv))
```

```
Call: survfit(formula = Surv(obsT, status) ~ IV, data = dfSurv,
    type = "kaplan-meier", conf.type = "log")

     records n.max n.start events median 0.95LCL 0.95UCL
IV=A      60    60      60     55   13.5       8      31
IV=B      60    60      60     49   33.0      17      50
IV=C      60    60      60     59    5.5       4      11
```

Die Ausgabe gibt – ggf. getrennt für jede Gruppe – Auskunft über die Anzahl der Beobachtungen (`records`), die Anzahl der Ereignisse (`events`) und unter `median` eine Schätzung für den Median der Überlebenszeit (das 50%-Quantil der geschätzten Survival-Funktion $\hat{S}(t)$) gefolgt von den Grenzen des zugehörigen Konfidenzintervalls (`0.95LCL`, `0.95UCL`).

Analog zum geschätzten Median der Überlebenszeit ermittelt `quantile(⟨survfit-Objekt⟩, probs=⟨Quantile⟩)` beliebige Quantile von $\hat{S}(t)$ – also Schätzungen für die Zeitpunkte, zu denen bei einem bestimmten Anteil der Beobachtungsobjekte ein Ereignis aufgetreten ist. In der Voreinstellung `conf.int=TRUE` erhält man zusätzlich die Grenzen des jeweiligen Konfidenzintervalls für ein Quantil.

```
# Quantile der Überlebenszeit mit Grenzen der Konfidenzintervalle
> quantile(KM0, probs=c(0.25, 0.5, 0.75), conf.int=TRUE)
$quantile
25%  50%  75%
4.0 14.5 47.0

$lower
25%  50%  75%
  3   10   35

$upper
25%  50%  75%
 7    19   68
```

Über `print(⟨survfit-Objekt⟩, print.rmean=TRUE)` erhält man auch die geschätzte mittlere Überlebenszeit.

```
> print(KM0, print.rmean=TRUE)
     n  events *rmean *se(rmean) median 0.95LCL 0.95UCL
180.00  163.00  32.64       2.89  14.50   10.00   19.00
* restricted mean with upper limit =  120
```

`summary(⟨survfit-Objekt⟩, times=⟨Vektor⟩)` gibt detailliert Auskunft über die Werte von $\hat{S}(t)$ zu den unter `times` genannten Zeiten t. So lässt sich etwa das geschätzte 100-Tage Überleben berechnen. Fehlt `times`, umfasst die Ausgabe alle tatsächlich aufgetretenen Ereigniszeiten t_i.

```
# Werte der geschätzten Survival-Funktion für 20, 50, 100 Tage
> summary(KM0, times=c(20, 50, 100))        # ohne Trennung nach Gruppen
Call: survfit(formula = Surv(obsT, status) ~ 1, data = dfSurv,
    type = "kaplan-meier", conf.type = "log")
```

```
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   20     75     105    0.417  0.0367       0.3505        0.495
   50     43      34    0.228  0.0313       0.1741        0.298
  100     22      19    0.122  0.0244       0.0826        0.181
```

Die Ausgabe nennt unter `time` die vorgegebenen Ereignis-Zeitpunkte t_i, unter `n.risk` die Anzahl der Personen, die vor t_i noch kein Ereignis hatten, unter `n.event` die Anzahl der Ereignisse zu t_i, unter `survival` die Schätzung $\hat{S}(t_i)$, unter `std.err` die geschätzte Streuung des Schätzers $\hat{S}(t_i)$ sowie in den letzten beiden Spalten die Grenzen des punktweisen Konfidenzintervalls für $\hat{S}(t_i)$.

9.3.2 Survival, kumulative Inzidenz und kumulatives hazard darstellen

Die grafische Darstellung von $\hat{S}(t)$ mit `plot(`⟨`survfit-Objekt`⟩`)` (Abb. 9.2) zeigt zusätzlich die Konfidenzintervalle. Die geschätzte kumulative Inzidenz $1 - \hat{S}(t)$ erhält man mit dem Argument `fun=function(x) { 1-x }`. Für das geschätzte kumulative hazard $\hat{\Lambda}(t) = -\ln \hat{S}(t)$ ist beim Aufruf von `plot()` das Argument `fun="cumhaz"` zu verwenden.

```
> plot(KM0, main=expression(paste("KM-Schätzer ", hat(S)(t),
+      " mit CI")), xlab="t", ylab="Survival", lwd=2)

> plot(KM0, main=expression(paste("KM-Schätzer 1-", hat(S)(t), " mit CI")),
+      xlab="t", ylab="kumulative Inzidenz", fun=function(x) {1-x}, lwd=2)

> plot(KM,  main=expression(paste("KM-Schätzer ", hat(S)[g](t), lty=1:3,
+      " für Gruppen")), xlab="t", ylab="Survival", lwd=2, col=1:3)

> legend(x="topright", lty=1:3, col=1:3, lwd=2, legend=LETTERS[1:3])

> plot(KM0, main=expression(paste("KM-Schätzer ", hat(Lambda)(t))),
+      xlab="t", ylab="kumulatives hazard", fun="cumhaz", lwd=2)
```

9.3.3 Log-Rank-Test auf gleiche Survival-Funktionen

`survdiff()` aus dem Paket `survival` berechnet den Log-Rank-Test, ob sich die Survival-Funktionen in mehreren Gruppen unterscheiden.[5]

```
> survdiff(⟨Modellformel⟩, rho=0, data=⟨Datensatz⟩)
```

Als erstes Argument ist eine Modellformel der Form ⟨`Surv-Objekt`⟩ `~` ⟨`Faktor`⟩ zu übergeben. Stammen die dabei verwendeten Variablen aus einem Datensatz, ist dieser unter `data` zu nennen. Das Argument `rho` kontrolliert, welche Testvariante berechnet wird. Mit der Voreinstellung 0 ist dies der Mantel-Hänszel-Test.

[5] Für eine exakte Alternative vgl. `logrank_test()` aus dem Paket `coin` (Hothorn, Hornik, van de Wiel & Zeileis, 2008).

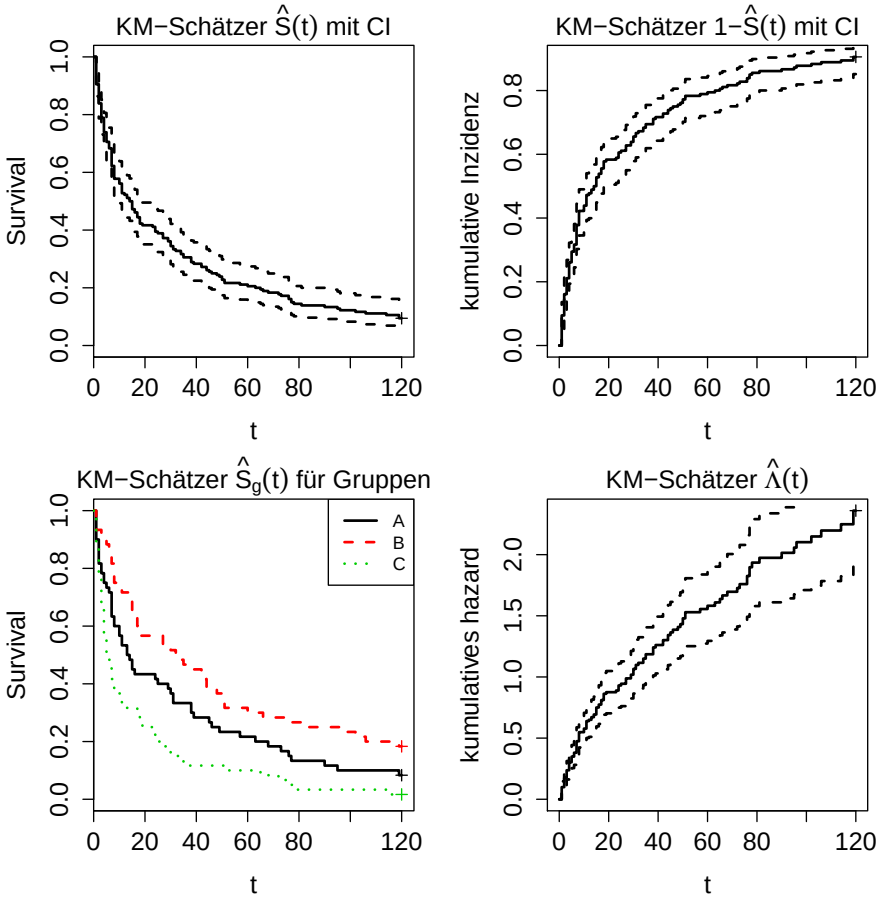

Abbildung 9.2: Kaplan-Meier-Schätzungen: Survival-Funktion $\hat{S}(t)$ sowie kumulative Inzidenz $1-\hat{S}(t)$ ohne Berücksichtigung der Gruppen (mit Konfidenzintervallen), separate Schätzungen $\hat{S}_g(t)$ getrennt nach Gruppen g sowie die geschätzte kumulative Hazard-Funktion $\hat{\Lambda}(t)$ (mit Konfidenzintervallen)

```
> library(survival)                                  # für survdiff()
> survdiff(Surv(obsT, status) ~ IV, data=dfSurv)
Call:
survdiff(formula = Surv(obsT, status) ~ IV, data = dfSurv)

      N Observed Expected (O-E)^2/E (O-E)^2/V
IV=A 60       55     54.6   0.00351   0.00552
IV=B 60       49     72.3   7.50042  14.39151
IV=C 60       59     36.2  14.43764  20.10283

Chisq= 23.9 on 2 degrees of freedom, p= 6.55e-06
```

Die Ausgabe nennt in der Spalte `N` die jeweilige Gruppengröße, unter `Observed` die Anzahl der beobachteten Ereignisse f_g^o pro Gruppe g, unter `Expected` die dort unter der Nullhypothese erwartete Anzahl von Ereignissen f_g^e, unter `(O-E)^2/E` den Wert für $\frac{(f_g^o - f_g^e)^2}{f_g^e}$ und unter

`(O-E)^2/V` $\frac{(f_g^o - f_g^e)^2}{\hat{\sigma}_D}$, wobei $\hat{\sigma}_D$ die geschätzte Varianz von $D = f_g^o - f_g^e$ ist. Die letzte Zeile liefert den Wert der asymptotisch χ^2-verteilten Teststatistik sowie die zugehörigen Freiheitsgrade mit dem p-Wert.

Bei geschichteten (stratifizierten) Stichproben lässt sich der Test auch unter Berücksichtigung der Schichtung durchführen. Dazu ist als weiterer Vorhersageterm auf der rechten Seite der Modellformel `strata(`⟨`Faktor`⟩`)` hinzuzufügen, wobei für dessen Gruppen keine Parameter geschätzt werden.

```
> survdiff(Surv(obsT, status) ~ IV + strata(sex), data=dfSurv)
Call:
survdiff(formula = Surv(obsT, status) ~ IV + strata(sex), data = dfSurv)

      N Observed Expected (O-E)^2/E (O-E)^2/V
IV=A 60       55     54.7   0.00185   0.00297
IV=B 60       49     72.1   7.42735  14.28570
IV=C 60       59     36.2  14.41091  20.32224

Chisq= 23.9  on 2 degrees of freedom, p= 6.45e-06
```

9.4 Cox proportional hazards Modell

Ein semi-parametrisches Regressionsmodell für zensierte Survival-Daten ist das Cox proportional hazards (PH) Modell, das den Einfluss von kontinuierlichen Prädiktoren ebenso wie von Gruppierungsfaktoren auf die Überlebenszeit einbeziehen kann. Das Modell lässt sich auf Daten anpassen, die aus unterschiedlichen Beobachtungszeiträumen hervorgegangen sind. Es macht keine spezifischen Voraussetzungen für die generelle Verteilungsform von Überlebenszeiten, basiert aber auf der Annahme, dass der Zusammenhang der p Prädiktoren X_j mit der logarithmierten Ereignisrate linear ist, sich also mit dem bekannten Regressionsmodell beschreiben lässt. Für die Form des Einflusses der X_j gelten insgesamt folgende Annahmen:

$$\begin{aligned}
\ln \lambda(t) &= \ln \lambda_0(t) + \beta_1 X_1 + \cdots + \beta_p X_p &&= \ln \lambda_0(t) + \boldsymbol{X\beta} \\
\lambda(t) &= \lambda_0(t)\, \mathrm{e}^{\beta_1 X_1 + \cdots + \beta_p X_p} &&= \lambda_0(t)\, \mathrm{e}^{\boldsymbol{X\beta}} \\
S(t) &= S_0(t)^{\exp(\boldsymbol{X\beta})} &&= \exp\left(-\Lambda_0(t)\, \mathrm{e}^{\boldsymbol{X\beta}}\right) \\
\Lambda(t) &= \Lambda_0(t)\, \mathrm{e}^{\boldsymbol{X\beta}}
\end{aligned}$$

Dabei spielt $\ln \lambda_0(t)$ die Rolle des absoluten Terms β_0 im GLM, entsprechend schließt die Abkürzung $\boldsymbol{X\beta}$ keinen absoluten Term ein (Kap. 8). $\lambda_0(t)$ ist das baseline hazard, also die für alle Beobachtungseinheiten identische Hazard-Funktion, wenn alle Einflussgrößen X_j gleich Null sind. Die Form dieser Funktion – und damit die generelle Verteilung von Überlebenszeiten (etwa Exponential- oder Weibull-Verteilung, s. Abschn. 9.1, 9.5) – bleibt unspezifiziert, weshalb dies der nonparametrische Teil des Modells ist.[6]

[6] Aus diesem Grund ist der Aussagebereich eines angepassten Cox PH-Modells auch auf die in der Stichprobe tatsächlich vorliegenden Überlebenszeiten begrenzt, eine Extrapolation über die maximal beobachtete Überlebenszeit hinaus also unzulässig.

Ein exponenziertes Gewicht e^{β_j} ist der multiplikative Faktor, mit dem sich die Ereignisrate verändert, wenn der Prädiktor X_j um eine Einheit wächst. Dies ist das *hazard ratio*, also das relative hazard nach Erhöhung von X_j um eine Einheit zu vorher. Die prozentuale Veränderung der Ereignisrate ist $100(e^{\beta_j} - 1)$. Bei $e^{\beta_j} = 1.3$ wäre jede zusätzliche Einheit von X_j mit einer um 30% höheren Ereignisrate assoziiert, bei $e^{\beta_j} = 0.6$ mit einer um 40% niedrigeren Rate. Das Modell impliziert die Annahme, dass das relative hazard zweier Personen i und j mit Prädiktorwerten $\boldsymbol{x}_i$ und $\boldsymbol{x}_j$ unabhängig vom baseline hazard $\lambda_0(t)$ sowie unabhängig vom Zeitpunkt t konstant ist.

$$\frac{\lambda_i(t)}{\lambda_j(t)} = \frac{\lambda_0(t)\, e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{\lambda_0(t)\, e^{\boldsymbol{x}_j'\boldsymbol{\beta}}} = \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}}}{e^{\boldsymbol{x}_j'\boldsymbol{\beta}}} = e^{(\boldsymbol{x}_i - \boldsymbol{x}_j)'\,\boldsymbol{\beta}}$$

Das hazard von Person i ist demnach proportional zum hazard von Person j mit dem über die Zeit konstanten Proportionalitätsfaktor $e^{(\boldsymbol{x}_i - \boldsymbol{x}_j)'\,\boldsymbol{\beta}}$. Diese Annahme proportionaler hazards von Personen mit unterschiedlichen Prädiktorwerten bedeutet, dass z. B. das hazard von Personen mit einer bestimmten Behandlungsmethode stets im selben Verhältnis zum hazard von Personen mit einer anderen Behandlungsmethode steht. Die hazards dürfen sich also nicht mit der Zeit annähern, stattdessen muss eine Methode gleichmäßig besser sein. Anders gesagt darf keine Interaktion $t \times \boldsymbol{X}$ vorliegen.

Das Cox PH-Modell wird mit `coxph()` aus dem Paket `survival` angepasst.

```
> coxph(⟨Modellformel⟩, ties="⟨Methode⟩", data=⟨Datensatz⟩)
```

Als erstes Argument ist eine Modellformel zu übergeben, deren linke Seite ein mit `Surv()` erstelltes Objekt ist (Abschn. 9.2). Die rechte Seite der Modellformel besteht aus kontinuierlichen Prädiktoren oder Faktoren, die zeitlich konstant oder – bei Daten in Zählprozess-Darstellung (Abschn. 9.2.2) – auch zeitabhängig sein können. Stammen die in der Modellformel verwendeten Variablen aus einem Datensatz, ist dieser unter `data` zu nennen. Zudem sind zwei besondere Vorhersageterme möglich: `strata(⟨Faktor⟩)` sorgt dafür, dass ein stratifiziertes Cox PH-Modell angepasst wird – mit einer separaten Baseline-Hazard-Funktion $\lambda_{0_g}(t)$ für jede Stufe g des Faktors. Der Term `cluster` ist in Abschn. 9.4.5 beschrieben.

Das Modell berücksichtigt für die Anpassung die in Rangdaten transformierten, ggf. zensierten Ereignis-Zeitpunkte, was es robust gegen Ausreißer macht. Gleichzeitig ist deshalb aber gesondert über das Argument `ties` zu spezifizieren, wie bei Bindungen, also identischen Ereignis-Zeitpunkten und damit uneindeutigen Rängen, vorzugehen ist. Voreinstellung ist `"efron"`, für eine früher häufig gewählte Methode ist das Argument auf `"breslow"` zu setzen. Eine exakte, aber rechenintensive Behandlung von Bindungen erhält man mit `"exact"`. Durch die Rangtransformation hat die Länge der Intervalle zwischen aufgetretenen Ereignissen keinen Einfluss auf die Parameterschätzungen.

Das Cox PH-Modell soll hier für die in Abschn. 9.2.1 simulierten Daten mit zeitlich konstanten Prädiktoren und höchstens einmal auftretenden Ereignissen angepasst werden.

```
> library(survival)                                  # für coxph()
> (fitCPH <- coxph(Surv(obsT, status) ~ X + IV, data=dfSurv))
Call:
coxph(formula = Surv(obsT, status) ~ X + IV, data = dfSurv)
```

```
      coef exp(coef) se(coef)     z       p
X    0.491     1.634   0.0869  5.66 1.5e-08
IVB -0.406     0.666   0.1968 -2.06 3.9e-02
IVC  0.579     1.784   0.1912  3.03 2.5e-03

Likelihood ratio test=56  on 3 df, p=4.1e-12  n= 180, number of events= 163

# äquivalenter Aufruf mit Daten in Zählprozess-Darstellung
> coxph(Surv(start, obsT, status) ~ X + IV, data=dfSurvCP)    # ...
```

Die Ausgabe nennt in der Spalte `coef` die geschätzten Koeffizienten $\hat{\beta}_j$, deren exponenzierte Werte $e^{\hat{\beta}_j}$ in der Spalte `exp(coef)` stehen. Für den kontinuierlichen Prädiktor X_j ist $e^{\hat{\beta}_j}$ der Änderungsfaktor für die geschätzte Ereignisrate $\hat{\lambda}(t)$ (also das hazard ratio), wenn X_j um eine Einheit wächst. Die Zeilen `IVB` und `IVC` sind so zu interpretieren, dass `A` als erste Stufe des Faktors `IV` als Referenzgruppe verwendet wurde (Abschn. 2.6.5), so dass Dummy-codierte Variablen für die Stufen `B` und `C` verbleiben (Treatment-Kontraste, s. Abschn. 12.9.2). Für `IVB` und `IVC` ist $e^{\hat{\beta}_j}$ daher jeweils der multiplikative Änderungsfaktor für $\hat{\lambda}(t)$ verglichen mit Gruppe `A`. Die geschätzten Streuungen $\hat{\sigma}_{\hat{\beta}}$ der $\hat{\beta}_j$ stehen in der Spalte `se(coef)`, die Werte der Wald-Statistik $z = \frac{\hat{\beta}}{\hat{\sigma}_{\hat{\beta}}}$ in der Spalte `z` und die zugehörigen p-Werte in der Spalte `p`. Dieser Wald-Test setzt voraus, dass z unter der Nullhypothese asymptotisch standardnormalverteilt ist.

Die letzte Zeile berichtet die Ergebnisse des Likelihood-Quotienten-Tests des Gesamtmodells gegen das Modell ohne Prädiktoren. Teststatistik ist die Devianz-Differenz beider Modelle mit der Differenz ihrer Freiheitsgrade als Anzahl der Freiheitsgrade der asymptotisch gültigen χ^2-Verteilung. Zusätzliche Informationen liefert `summary(`⟨`coxph-Objekt`⟩`)`.

```
> summary(fitCPH)
Call:
coxph(formula = Surv(obsT, status) ~ X + IV, data = dfSurv)

n= 180, number of events= 163

        coef exp(coef) se(coef)      z Pr(>|z|)
X    0.49123   1.63433  0.08685  5.656 1.55e-08 ***
IVB -0.40612   0.66623  0.19675 -2.064  0.03901 *
IVC  0.57890   1.78407  0.19117  3.028  0.00246 **

    exp(coef) exp(-coef) lower .95 upper .95
X      1.6343     0.6119    1.3785    1.9376
IVB    0.6662     1.5010    0.4531    0.9797
IVC    1.7841     0.5605    1.2266    2.5950

Concordance= 0.683  (se = 0.027 )
Rsquare= 0.268   (max possible= 1 )
Likelihood ratio test= 56.05  on 3 df,   p=4.103e-12
```

```
Wald test            = 54.98  on 3 df,   p=6.945e-12
Score (logrank) test = 54.86  on 3 df,   p=7.365e-12
```

Neben den bereits erwähnten Informationen enthält die Ausgabe zusätzlich die Konfidenzintervalle für die exponenzierten Schätzungen $e^{\hat{\beta}_j}$ in den Spalten `lower .95` und `upper .95`. Die Konkordanz ist der Anteil an allen Paaren von Beobachtungsobjekten, bei denen das Beobachtungsobjekt mit empirisch kürzerer Überlebenszeit auch ein höheres vorhergesagtes hazard besitzt. Liegen keine Bindungen vor, ist die Konkordanz daher gleich Kendalls τ von T und $\hat{\lambda}$ (Abschn. 10.3.1). Der unter `Rsquare` angegebene pseudo-R^2-Wert ist jener nach Cox & Snell (Abschn. 8.1.3). Die Ergebnisse des Likelihood-Quotienten-, Wald- und Score-Tests beziehen sich alle auf den Test des Gesamtmodells gegen jenes ohne Prädiktoren.

9.4.1 Anpassungsgüte und Modelltests

Aus einem von `coxph()` zurückgegebenen Objekt lassen sich weitere Informationen zur Anpassungsgüte extrahieren, darunter der Wert des Informationskriteriums AIC mit `extractAIC()` oder die pseudo-R^2-Werte nach McFadden, Cox & Snell und Nagelkerke (Abschn. 8.1.3). Dafür enthält das Objekt in der Komponente `loglik` einen Vektor mit den maximierten geschätzten likelihoods des Modells ohne Prädiktoren und des angepassten Modells.

```
> library(survival)                              # für coxph()
> extractAIC(fitCPH)                             # AIC
[1] 3.000 1399.438

> LL0 <- fitCPH$loglik[1]           # log-likelihood 0-Modell
> LLf <- fitCPH$loglik[2]           # log-likelihood angepasstes Modell

> as.vector( 1 - (LLf / LL0))                    # R^2 McFadden
[1] 0.03866744

> as.vector( 1 - exp((2/N) * (LL0 - LLf)))       # R^2 Cox & Snell
[1] 0.2675625

# R^2 Nagelkerke
> as.vector((1 - exp((2/N) * (LL0 - LLf))) / (1 - exp(LL0)^(2/N)))
[1] 0.2676477
```

Da der hier im Modell berücksichtigte Faktor `IV` mit mehreren Parametern β_j assoziiert ist, muss seine Signifikanz insgesamt über einen Modellvergleich getestet werden. Dazu dient ein Likelihood-Quotienten-Test, der auf der asymptotisch χ^2-verteilten Devianz-Differenz zweier hierarchischer Modelle mit demselben Kriterium beruht (Abschn. 8.1.5): Der Prädiktorensatz des eingeschränkten Modells ⟨`fitR`⟩ ist dabei vollständig im Prädiktorensatz des umfassenderen Modells ⟨`fitU`⟩ enthalten, das zusätzlich noch den Faktor berücksichtigt. Der Test erfolgt dann mit `anova(`⟨`fitR`⟩`,` ⟨`fitU`⟩`)`.

```
# eingeschränktes Modell ohne Faktor IV
> fitCPH1 <- coxph(Surv(obsT, status) ~ X, data=dfSurv)
```

```
> anova(fitCPH1, fitCPH)                          # LQ-Modelltest für IV
Analysis of Deviance Table
Cox model: response is  Surv(obsT, status)
Model 1: ~ X
Model 2: ~ X + IV
   loglik  Chisq Df P(>|Chi|)
1 -708.98
2 -696.72  24.52  2 4.738e-06 ***
```

9.4.2 Survival-Funktion und baseline hazard schätzen

Analog zur Kaplan-Meier-Analyse (Abschn. 9.3.1) schätzt `survfit(⟨coxph-Objekt⟩)` die Survival-Funktion $\hat{S}_{\overline{\boldsymbol{x}}}(t)$ im Cox PH-Modell für ein pseudo-Beobachtungsobjekt, das als Prädiktorwerte den jeweiligen Mittelwert für die gegebene Stichprobe besitzt (kurz: $\overline{\boldsymbol{x}}$). Der Median der geschätzten Überlebenszeit findet sich in der Ausgabe unter `median`, beliebige Quantile von $\hat{S}_{\overline{\boldsymbol{x}}}(t)$ erhält man mit `quantile(⟨coxph-Objekt⟩, probs=⟨Quantile⟩)`.

```
> library(survival)                             # für survfit(), basehaz()
> (CPH <- survfit(fitCPH, conf.type="log"))
Call: survfit(formula = fitCPH)

records  n.max  n.start  events  median  0.95LCL  0.95UCL
    180    180      180     163      15       11       19

# Quantile der Überlebenszeit ohne Grenzen der Konfidenzintervalle
> quantile(CPH, probs=c(0.25, 0.5, 0.75), conf.int=FALSE)
25% 50% 75%
  5  15  42
```

Berücksichtigt das Modell Faktoren, ist die mittlere pseudo-Beobachtung $\overline{\boldsymbol{x}}$ kaum zu interpretieren, da für sie die Mittelwerte der dichotomen Indikatorvariablen gebildet werden (Abschn. 12.9.2). Diese Mittelwerte entsprechen damit keiner tatsächlich vorhandenen Gruppe. Oft ist es dann angemessener, an das Argument `newdata` von `survfit()` einen Datensatz zu übergeben, der neue Daten für Variablen mit denselben Namen, und bei Faktoren zusätzlich denselben Stufen wie jene der ursprünglichen Prädiktoren enthält. In diesem Fall berechnet `survfit()` für jede Zeile des Datensatzes die Schätzung $\hat{S}(t)$. Auf diese Weise kann $\hat{S}(t)$ etwa für bestimmte Gruppenzugehörigkeiten oder Werte anderer Prädiktoren ermittelt werden.

In der von `survfit()` zurückgegebenen Liste stehen die Werte für t in der Komponente `time`, jene für $\hat{S}(t)$ in der Komponente `surv`. Dabei ist `surv` eine Matrix mit so vielen Zeilen, wie es Werte für t gibt und so vielen Spalten, wie neue Beobachtungsobjekte (Zeilen von `newdata`) vorhanden sind.

```
# Datensatz: 2 Frauen mit Prädiktor X=-2 in Gruppe A bzw. in C
> dfNew <- data.frame(sex=factor(c("f", "f"), levels=levels(dfSurv$sex)),
+                     X=c(-2, -2),
+                     IV=factor(c("A", "C"), levels=levels(dfSurv$IV)))
```

```
# wende angepasstes Cox PH-Modell auf neue Daten an
> CPHnew <- survfit(fitCPH, newdata=dfNew)
```

Die grafische Darstellung von $\hat{S}_{\overline{\boldsymbol{x}}}(t)$ bzw. von $\hat{S}(t)$ für die Beobachtungen in `newdata` erfolgt mit `plot(`⟨`survfit-Objekt`⟩`)` (Abb. 9.3). Die geschätzte kumulative Inzidenz $1 - \hat{S}(t)$ erhält man mit dem Argument `fun=function(x) { 1-x }`. Für die geschätzte kumulative Hazard-Funktion $\hat{\Lambda}(t) = -\ln \hat{S}(t)$ ist beim Aufruf von `plot()` das Argument `fun="cumhaz"` zu verwenden.

```
# Darstellung geschätztes S(t) für mittlere pseudo-Beobachtung
> plot(CPH, main=expression(paste("Cox PH-Schätzung ", hat(S)(t),
+      " mit CI")), xlab="t", ylab="Survival", lwd=2)

# füge geschätztes S(t) für neue Daten hinzu
> lines(CPHnew$time, CPHnew$surv[ , 1], lwd=2, col="blue")
> lines(CPHnew$time, CPHnew$surv[ , 2], lwd=2, col="red")
> legend(x="topright", lwd=2, col=c("black", "blue", "red"),
+        legend=c("pseudo-Beobachtung", "sex=f, X=-2, IV=A",
+                 "sex=f, X=-2, IV=C"))
```

Das geschätzte kumulative baseline hazard erhält man mit `basehaz(`⟨`coxph-Objekt`⟩`)` aus dem Paket `survival` (Abb. 9.3). Die Schätzung $\hat{\Lambda}_{\overline{\boldsymbol{x}}}(t)$ kann dabei mit dem Argument `centered=TRUE` für ein pseudo-Beobachtungsobjekt berechnet werden, das für die vorliegende Stichprobe gemittelte Prädiktorwerte $\overline{\boldsymbol{x}}$ besitzt. Setzt man dagegen `centered=FALSE`, bezieht sich das Ergebnis $\hat{\Lambda}_0(t)$ i. S. einer echten baseline bei Treatment-Kontrasten auf ein Beobachtungsobjekt in der Referenzgruppe eines Faktors, für das alle Prädiktorwerte gleich 0 sind. Das Ergebnis ist ein Datensatz mit den Variablen `hazard` für $\hat{\Lambda}_0(t)$ und `time` für t.

Um das geschätzte kumulative hazard $\hat{\Lambda}(t)$ für beliebige Werte der Prädiktoren zu ermitteln, ist $\hat{\Lambda}_0(t) \cdot \mathrm{e}^{\boldsymbol{X}\hat{\boldsymbol{\beta}}}$ zu bilden. Soll $\hat{\Lambda}(t)$ nicht für die Referenzgruppe, sondern für eine andere Stufe j eines Faktors berechnet werden, vereinfacht sich der Ausdruck bei Treatment-Kontrasten zu $\hat{\Lambda}_0(t) \cdot \mathrm{e}^{\hat{\beta}_j}$.

```
# exponenzierte geschätzte Koeffizienten = Faktoren für hazard
> expCoef <- exp(coef(fitCPH))

# kumulatives hazard für Referenzgruppe A und X=0
> Lambda0A <- basehaz(fitCPH, centered=FALSE)
> Lambda0B <- expCoef[2]*Lambda0A$hazard      # kumulatives hazard B
> Lambda0C <- expCoef[3]*Lambda0A$hazard      # kumulatives hazard C

# stelle kumulatives hazard für Gruppe A dar
> plot(hazard ~ time, main=expression(paste("Cox PH-Schätzung ",
+      hat(Lambda)[g](t), " pro Gruppe")), type="s", ylim=c(0, 5),
+      xlab="t", ylab="kumulatives hazard", lwd=2, data=Lambda0A)

# füge kumulatives hazard für Gruppe B und C hinzu
> lines(Lambda0A$time, Lambda0B, lwd=2, col="red")
```

```
> lines(Lambda0A$time, Lambda0C, lwd=2, col="green")
> legend(x="bottomright", lwd=2, col=1:3, legend=LETTERS[1:3])
```

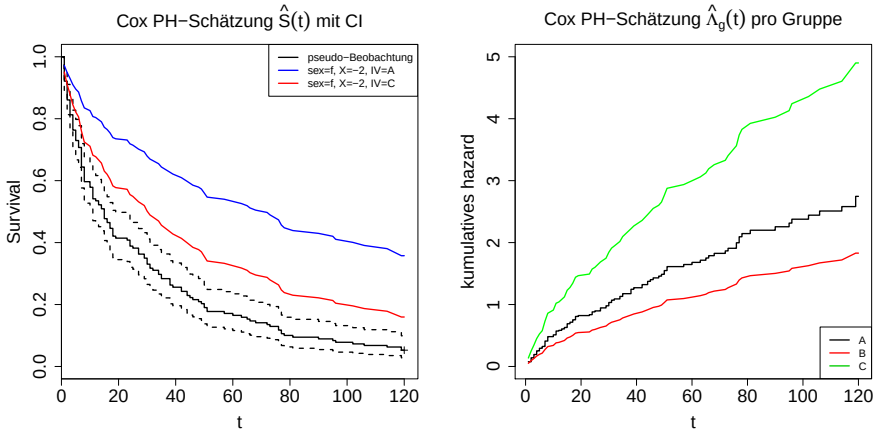

Abbildung 9.3: Cox PH-Schätzungen: Survival-Funktion $\hat{S}_{\overline{x}}(t)$ mit Konfidenzintervallen für ein pseudo-Beobachtungsobjekt mit mittleren Prädiktorwerten $\overline{x}$ sowie kumulative Hazard-Funktion $\hat{\Lambda}_g(t)$ getrennt nach Gruppen g

9.4.3 Modelldiagnostik

Um die Angemessenheit des Cox PH-Modells für die gegebenen Daten beurteilen zu können, sollten drei Aspekte der Modellanpassung geprüft werden: Die Annahme proportionaler hazards, das Vorhandensein besonders einflussreicher Beobachtungen sowie die Linearität von $\ln \lambda(t)$ bzgl. der kontinuierlichen Prädiktoren.

Aus der Annahme proportionaler hazards folgt, dass $\ln(-\ln(S(t)))$ linear mit $\ln t$ ist. Diese Voraussetzung lässt sich in einem Diagramm prüfen, das $\ln(-\ln(\hat{S}_g(t_i)))$ gegen $\ln t_i$ aufträgt, wobei $\hat{S}_g(t_i)$ die geschätzten Kaplan-Meier Survival-Funktionen der Überlebenszeit für die Stufen g eines Faktors sind (Abb. 9.4). Dafür ist separat für jeden Prädiktor ein Kaplan-Meier Modell anzupassen und mit `plot(⟨KM-Modell⟩, fun="cloglog")` darzustellen. Damit sich die PH-Annahme bzgl. kontinuierlicher Prädiktoren auf diese Weise prüfen lässt, müssen diese zunächst in Gruppen eingeteilt werden (Abschn. 2.6.7).

Die Survival-Funktionen sollten im Diagramm linear mit $\ln t_i$ ansteigen und zudem parallel verlaufen. Im Spezialfall des Modells, das für T eine Exponentialverteilung annimmt (Abschn. 9.5), sollte die Steigung von $\hat{S}_g(t_i)$ gleich 1 sein. Ist die PH-Annahme offensichtlich für einen Prädiktor verletzt, besteht eine Strategie darin, bzgl. dieses Prädiktors zu stratifizieren – bei kontinuierlichen Variablen nach Einteilung in geeignete Gruppen.

```
> library(survival)                              # für survfit(), cox.zph()

# teile X per Median-Split in zwei Gruppen
> dfSurv <- transform(dfSurv, Xcut=cut(X, breaks=c(-Inf, median(X), Inf)))
> KMiv   <- survfit(Surv(obsT, status) ~ IV,     # KM-Schätzungen für IV
+                   type="kaplan-meier", data=dfSurv)
```

```
> KMxcut <- survfit(Surv(obsT, status) ~ Xcut,  # KM-Schätzungen für X
+                   type="kaplan-meier", data=dfSurv)

# Diagramme ln(-ln(S(t))) gegen ln t
> plot(KMiv, fun="cloglog", main="cloglog-Plot für IV1",
+      xlab="ln t", ylab=expression(ln(-ln(hat(S)[g](t)))),
+      col=c("black", "blue", "red"), lty=1:3)

> legend(x="topleft", col=c("black", "blue", "red"), lwd=2,
+        lty=1:3, legend=LETTERS[1:3])

> plot(KMxcut, fun="cloglog", main="cloglog-Plot für Xcut",
+      xlab="ln t", ylab=expression(ln(-ln(hat(S)[g](t)))),
+      col=c("black", "blue"), lty=1:2)

> legend(x="topleft", col=c("black", "blue"), lwd=2,
+        legend=c("lo", "hi"), lty=1:2)
```

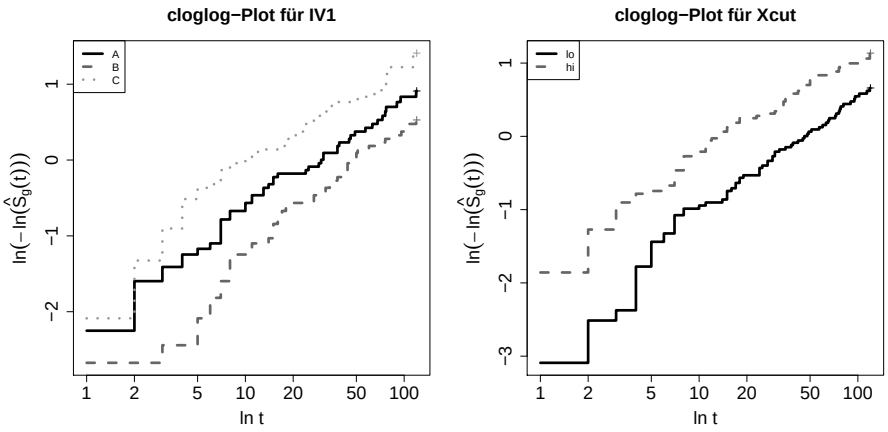

Abbildung 9.4: Beurteilung der Annahme proportionaler hazards bzgl. jedes Prädiktors anhand der Linearität von $\ln(-\ln(\hat{S}_g(t_i)))$ mit $\ln t_i$

Ähnlich wie in der Regressionsdiagnostik (Abschn. 6.5) kann sich die Beurteilung der Voraussetzungen des Cox PH-Modells auch auf die Verteilung von Residuen stützen, insbesondere auf die Schönfeld- und Martingal-Residuen. Beide erhält man mit `residuals(`⟨`coxph-Objekt`⟩`, type="`⟨`Typ`⟩`")`, wobei `type` auf `"scaledsch"` bzw. auf `"martingale"` zu setzen ist. Das Ergebnis ist eine Matrix, die für jede Beobachtung (Zeilen) das Residuum bzgl. jedes Prädiktors (Spalten) enthält.

`cox.zph(`⟨`coxph-Objekt`⟩`)` berechnet ausgehend von der Korrelation der Schönfeld-Residuen mit einer Transformation der Überlebenszeit für jeden Prädiktor sowie für das Gesamtmodell einen χ^2-Test der Nullhypothese, dass die Annahme proportionaler hazards stimmt.

```
> (czph <- cox.zph(fitCPH))
          rho  chisq     p
```

```
X       -0.0959 1.5316 0.216
IVB      0.1001 1.6013 0.206
IVC      0.0216 0.0761 0.783
GLOBAL       NA 3.2264 0.358
```

Das von `cox.zph()` ausgegebene Objekt lässt sich an `plot()` übergeben, um die Schönfeld-Residuen für jeden Prädiktor gegen eine Transformation der Überlebenszeit darzustellen (Abb. 9.5). Die Diagramme enthalten zur Verdeutlichung des Verlaufs eine Spline-Interpolation (Abschn. 16.1.2) inkl. des zugehörigen Bereichs von ±2 Standardfehlern. Gibt es eine systematische Variation der Residuen in Abhängigkeit von der Überlebenszeit, ist das ein Hinweis darauf, dass die Annahme proportionaler hazards verletzt ist.

```
> par(mfrow=c(2, 2))                    # Platz für 4 panels
> plot(czph)                            # Residuen und splines
```

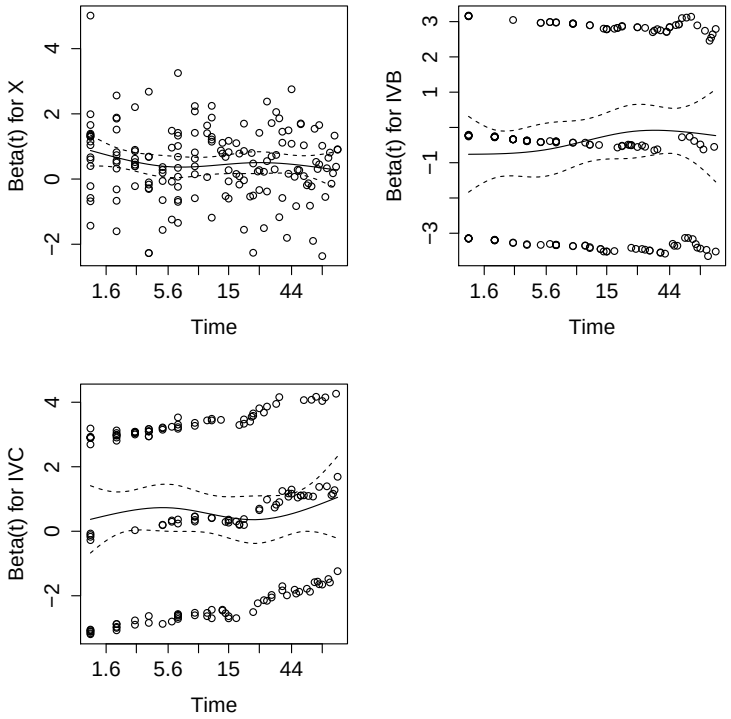

Abbildung 9.5: Beurteilung der Annahme proportionaler hazards bzgl. jedes Prädiktors anhand skalierter Schönfeld-Residuen

Einflussreiche Beobachtungen können ähnlich wie in der linearen Regression über die von ihnen verursachten Änderungen in den geschätzten Parametern $\hat{\beta}_j$ diagnostiziert werden (Abschn. 6.5.1). Das standardisierte Maß DfBETAS erhält man für jede Beobachtung und jeden Prädiktor, indem man im Aufruf von `residuals()` das Argument `type="dfbetas"` wählt (Abb. 9.6).

```
# Matrix der standardisierten Einflussgrößen DfBETAS
> dfbetas <- residuals(fitCPH, type="dfbetas")
> plot(dfbetas[ , 1], type="h", main="DfBETAS für X",
+      ylab="DfBETAS", lwd=2)
```

```
> plot(dfbetas[ , 2], type="h", main="DfBETAS für IV-B",
+      ylab="DfBETAS", lwd=2)

> plot(dfbetas[ , 3], type="h", main="DfBETAS für IV-C",
+      ylab="DfBETAS", lwd=2)
```

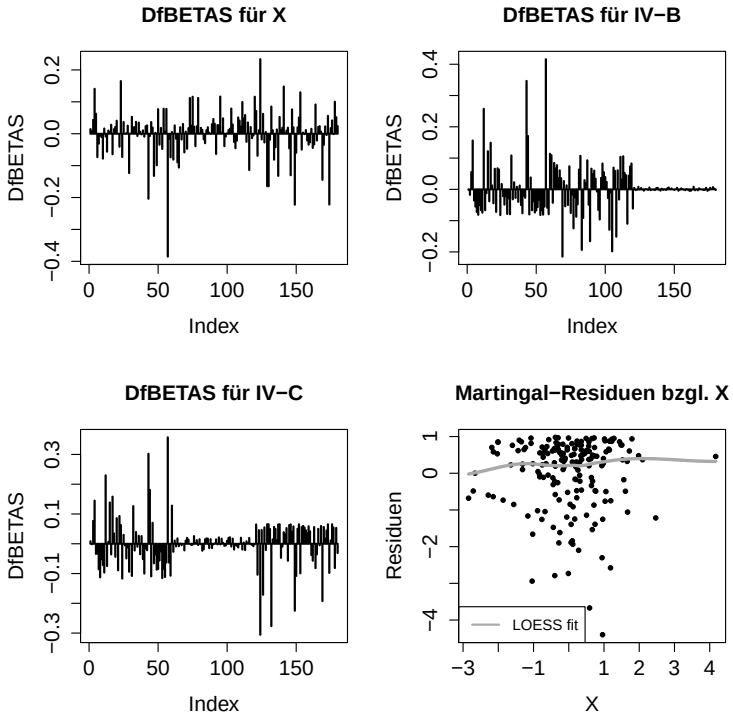

Abbildung 9.6: Diagnostik einflussreicher Beobachtungen anhand der DfBETAS-Werte für jeden Prädiktor sowie Beurteilung der Linearität bzgl. des kontinuierlichen Prädiktors

Laut Modell sollte der Zusammenhang von $\ln \lambda(t)$ mit den Prädiktoren X_j linear sein. Inwieweit die Daten mit dieser Annahme konsistent sind, lässt sich über den Verlauf der Martingal-Residuen in Abhängigkeit von den Werten der kontinuierlichen X_j einschätzen (Abb. 9.6). Zur grafischen Beurteilung der Verteilung ist es dabei hilfreich, einen nonparametrichen LOESS-Glätter (Abschn. 16.1.3) einzuzeichnen, der horizontal verlaufen sollte.

```
> resMart <- residuals(fitCPH, type="martingale")
> plot(dfSurv$X, resMart, main="Martingal-Residuen bzgl. X",
+      xlab="X", ylab="Residuen", pch=20)

# zeichne zusätzlich LOESS-Glätter ein
> lines(loess.smooth(dfSurv$X, resMart), lwd=2, col="blue")
> legend(x="bottomleft", col="blue", lwd=2, legend="LOESS fit")
```

9.4.4 Vorhersage und Anwendung auf neue Daten

Wie im GLM liefert `predict(⟨coxph-Modell⟩, type="⟨Typ⟩")` für jedes Beobachtungsobjekt i Schätzungen verschiedener Kennwerte. Mit dem Argument `type="risk"` erhält man die hazard ratios $e^{(\boldsymbol{x}_i - \overline{\boldsymbol{x}})'\hat{\boldsymbol{\beta}}}$ zu einem pseudo-Beobachtungsobjekt, das als Prädiktorwerte den jeweiligen Mittelwert jedes Prädiktors aus der Stichprobe besitzt (kurz: $\overline{\boldsymbol{x}}$, s. Abschn. 9.4.2). Für stratifizierte Modelle werden diese Mittelwerte dabei pro Schicht gebildet, es sei denn man setzt das Argument `reference="sample"`. Für die Werte des linearen Prädiktors $\boldsymbol{X}\hat{\boldsymbol{\beta}}$ selbst ist `type="lp"` zu verwenden.

Zusätzlich akzeptiert das Argument `newdata` von `predict()` einen Datensatz, der neue Daten für Variablen mit denselben Namen, und bei Faktoren zusätzlich denselben Stufen wie jene der ursprünglichen Prädiktoren enthält. Als Ergebnis erhält man die vorhergesagten hazard ratios für die neuen Prädiktorwerte (Abschn. 6.4).

`survexp()` aus dem Paket `survival` ermittelt auf Basis eines Cox PH-Modells die Vorhersage der Survival-Funktion $\hat{S}(t_i)$ in Abhängigkeit von Prädiktorwerten.

```
> survexp(⟨Modellformel⟩, ratetable="⟨coxph-Modell⟩", data=⟨Datensatz⟩)
```

Als erstes Argument ist die rechte Seite einer Modellformel mit den Prädiktoren anzugeben, für deren Kombination von Werten jeweils der Wert der Survival-Funktion geschätzt werden soll. Als Basis dient ein Cox PH-Modell, das für `ratetable` zu übergeben ist. Dieses Modell liefert auch die Ereigniszeiten t_i, für die $\hat{S}(t_i)$ bestimmt wird. Der Datensatz mit Daten für die Variablen der Modellformel wird von `data` akzeptiert.

Die zurückgegebene Liste enthält in der Komponente `time` die Ereigniszeiten t_i und in der Komponente `surv` die Werte von $\hat{S}(t_i)$. Ohne Trennung nach Gruppen oder anderen Prädiktoren (Modellformel `~ 1`) ist `surv` ein Vektor. Stehen Prädiktoren in der Modellformel (`~ X + IV1 + ...`), ist `surv` eine Matrix mit je einer Spalte für jede Kombination g von Faktorstufen und Werten der kontinuierlichen Prädiktoren. Die zugehörigen Werte von $\hat{S}_g(t_i)$ stehen in den Zeilen.

```
# Vorhersage für Gesamtgruppe ohne Trennung nach Prädiktoren
> Shat1 <- survexp(~ 1, ratetable=fitCPH, data=dfSurv)
> with(Shat1, head(data.frame(time, surv), n=4))
  time      surv
1    1 0.9053526
2    2 0.8372556
3    3 0.7858015
4    4 0.7342081

# Vorhersage getrennt nach Gruppen von IV
> Shat2 <- survexp(~ IV, ratetable=fitCPH, data=dfSurv)
> with(Shat2, head(data.frame(time, surv), n=4))
  time      IV.A      IV.B      IV.C
1    1 0.9195093 0.9455425 0.8510060
2    2 0.8582419 0.9028127 0.7507122
3    3 0.8104261 0.8686281 0.6783505
```

```
4     4 0.7613120 0.8326912 0.6086210
```

9.4.5 Erweiterungen des Cox PH-Modells

Das Cox PH-Modell lässt sich auf verschiedene Situationen erweitern:

- Zeitvariierende Kovariaten X können über die Zählprozess-Darstellung von Daten (Abschn. 9.2.2) repräsentiert und etwa in Interaktionstermen $X \times t$ in die Modellformel von `coxph()` aufgenommen werden. Dafür ist es notwendig, die Gesamt-Beobachtungsintervalle aller Personen mit `survSplit()` an allen vorkommenden Ereigniszeiten in Teilintervalle zu zerlegen.
- Pro Beobachtungsobjekt potentiell mehrfach auftretende Ereignisse lassen sich ebenfalls in Zählprozess-Darstellung speichern. In der Modellformel von `coxph()` kann dann ein Vorhersageterm `cluster(`⟨`ID`⟩`)` hinzugefügt werden, wobei der Faktor ⟨`ID`⟩ codiert, von welchem Beobachtungsobjekt ein Intervall stammt. Dies bewirkt eine robuste Schätzung der Kovarianzmatrix der Parameterschätzer. Alternativ können wiederkehrende Ereignisse durch Stratifizierung analysiert werden, wobei die Beobachtungen mit jeweils dem ersten, zweiten, dritten, ... Ereignis ein Stratum bilden.
- Penalisierte Cox-Modelle können mit der Funktion `coxnet()` aus dem Paket `glmnet` (Abschn. 6.6.2) sowie mit dem Paket `coxphf` (Ploner & Heinze, 2013) angepasst werden.
- Für Hinweise zur Auswertung mit *frailty* (Duchateau & Janssen, 2007) oder *competing risks* (Beyersmann, Allignol & Schumacher, 2012) Modellen vgl. den Abschnitt *Survival Analysis* der CRAN Task Views (Allignol & Latouche, 2014).

9.5 Parametrische proportional hazards Modelle

Bei spezifischen Vorstellungen über die Verteilung der Überlebenszeit T kommen auch parametrische Regressionsmodelle in Betracht, die sich unter Beibehaltung der Annahme proportionaler hazards als Spezialfälle des Cox PH-Modells ergeben (Abschn. 9.4). Für exponential- oder Weibull-verteilte Überlebenszeiten gibt es dabei zwei äquivalente Möglichkeiten, das lineare Regressionsmodell zu formulieren: Zum einen wie im Cox PH-Modell für das logarithmierte hazard, zum anderen für die logarithmierte Überlebenszeit. Bei der zweiten Darstellung spricht man von einem *accelerated failure time* Modell (AFT). Da in parametrischen Modellen das hazard voll spezifiziert ist, lassen sie sich anders als Cox PH-Modelle auch zur Vorhersage jenseits des letzten beobachteten Ereignisses nutzen.

9.5.1 Darstellung über die Hazard-Funktion

Spezialfälle des Cox PH-Modells ergeben sich, wenn für das baseline hazard $\lambda_0(t)$ eine Verteilung angenommen wird, die mit einer Exponential- oder Weibull-Verteilung von T korrespondiert. Die logarithmierte Ereignisrate soll wie im Cox PH-Modell linear von den Prädiktoren X_j abhängen. Das baseline hazard $\lambda_0(t)$ ist dabei der Verlauf der Ereignisrate für ein Beobachtungsobjekt,

für das alle X_j gleich 0 sind. Die exponenzierten Parameter e^{β_j} geben wie im Cox PH-Modell den Änderungsfaktor für die Ereignisrate (also das hazard ratio) an, wenn ein Prädiktor um 1 wächst.

Bei Annahme einer Exponentialverteilung von T mit Erwartungswert $E(T) = b > 0$ und Varianz b^2 ergibt sich die Dichtefunktion $f(t) = \lambda(t)\, S(t)$ aus der konstanten Hazard-Funktion $\lambda(t) = \lambda = \frac{1}{b}$ und der Survival-Funktion $S(t) = \mathrm{e}^{-\frac{t}{b}}$. Die kumulative Hazard-Funktion ist $\Lambda(t) = \frac{t}{b}$. Oft wird die Exponentialverteilung auch mit der Grundrate λ als $f(t) = \lambda\, \mathrm{e}^{-\lambda t}$ bzw. $S(t) = \mathrm{e}^{-\lambda t}$ und $\Lambda(t) = \lambda t$ formuliert. In `rexp()` ist mit dem Argument `rate` λ gemeint. Durch den Einfluss der X_j ergibt sich dann als neue Grundrate $\lambda' = \lambda\, \mathrm{e}^{\boldsymbol{X\beta}}$. Insgesamt resultieren aus der Spezialisierung des Cox PH-Modells folgende Modellvorstellungen, wobei die Abkürzung $\boldsymbol{X\beta}$ keinen absoluten Term β_0 einschließt:

$$\begin{aligned}
\lambda(t) &= \tfrac{1}{b}\, \mathrm{e}^{\boldsymbol{X\beta}} &&= \lambda\, \mathrm{e}^{\boldsymbol{X\beta}} \\
\ln \lambda(t) &= -\ln b + \boldsymbol{X\beta} &&= \ln \lambda + \boldsymbol{X\beta} \\
S(t) &= \exp\left(-\tfrac{t}{b}\, \mathrm{e}^{\boldsymbol{X\beta}}\right) &&= \exp\left(-\lambda\, t\, \mathrm{e}^{\boldsymbol{X\beta}}\right) \\
\Lambda(t) &= \tfrac{t}{b}\, \mathrm{e}^{\boldsymbol{X\beta}} &&= \lambda\, t\, \mathrm{e}^{\boldsymbol{X\beta}}
\end{aligned}$$

Die Dichtefunktion einer Weibull-Verteilung kann unterschiedlich formuliert werden. `rweibull()` verwendet den Formparameter $a > 0$ für das Argument `shape` und den Skalierungsparameter $b > 0$ für `scale`. Für $a > 1$ steigt das hazard mit t, für $a < 1$ sinkt es, und für $a = 1$ ist es konstant. Die Exponentialverteilung ist also ein Spezialfall der Weibull-Verteilung für $a = 1$. b ist die *charakteristische Lebensdauer*, nach der $1 - \frac{1}{\mathrm{e}} \approx 63.2\%$ der Ereignisse aufgetreten sind ($S(b) = \frac{1}{\mathrm{e}}$). Die Dichtefunktion $f(t) = \lambda(t)\, S(t)$ ergibt sich mit dieser Wahl aus der Hazard-Funktion $\lambda(t) = \frac{a}{b}\, (\frac{t}{b})^{a-1}$ und der Survival-Funktion $S(t) = \exp(-(\frac{t}{b})^a)$. Die kumulative Hazard-Funktion ist $\Lambda(t) = (\frac{t}{b})^a$ mit der Umkehrfunktion $\Lambda^{-1}(t) = (b\, t)^{\frac{1}{a}}$. Der Erwartungswert ist $E(T) = b\, \Gamma(1 + \frac{1}{a})$.

Analog zur Exponentialverteilung lässt sich die Weibull-Verteilung auch mit $\lambda = \frac{1}{b^a}$ formulieren, so dass $\lambda(t) = \lambda\, a\, t^{a-1}$, $S(t) = \exp(-\lambda\, t^a)$ und $\Lambda(t) = \lambda\, t^a$ gilt. Durch den Einfluss der X_j ergibt sich dann $\lambda' = \lambda\, \mathrm{e}^{\boldsymbol{X\beta}}$. Insgesamt impliziert das Weibull-Modell folgende Zusammenhänge:

$$\begin{aligned}
\lambda(t) &= \tfrac{a}{b}\, (\tfrac{t}{b})^{a-1}\, \mathrm{e}^{\boldsymbol{X\beta}} &&= \lambda\, a\, t^{a-1}\, \mathrm{e}^{\boldsymbol{X\beta}} \\
\ln \lambda(t) &= \ln\left(\tfrac{a}{b}\, (\tfrac{t}{b})^{a-1}\right) + \boldsymbol{X\beta} &&= \ln \lambda + \ln a + (a-1)\, \ln t + \boldsymbol{X\beta} \\
S(t) &= \exp\left(-(\tfrac{t}{b})^a\, \mathrm{e}^{\boldsymbol{X\beta}}\right) &&= \exp\left(-\lambda\, t^a\, \mathrm{e}^{\boldsymbol{X\beta}}\right) \\
\Lambda(t) &= (\tfrac{t}{b})^a\, \mathrm{e}^{\boldsymbol{X\beta}} &&= \lambda\, t^a\, \mathrm{e}^{\boldsymbol{X\beta}}
\end{aligned}$$

9.5.2 Darstellung als accelerated failure time Modell

Das betrachtete Exponential- und Weibull-Modell lässt sich äquivalent auch jeweils als lineares Modell der logarithmierten Überlebenszeit formulieren (accelerated failure time Modell, AFT).

$$\ln T = \boldsymbol{X\gamma} + \boldsymbol{z} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$$

Dabei ist ϵ ein Fehlerterm, der im Weibull-Modell einer Typ-I (Gumbel) Extremwertverteilung folgt und durch $\sigma = \frac{1}{a}$ skaliert wird. Sind t_i zufällige Überlebenszeiten aus einer

Weibull-Verteilung, sind damit $\ln t_i$ zufällige Beobachtungen einer Extremwertverteilung mit Erwartungswert $\mu = \ln b$. Mit $a = 1$ ergibt sich als Spezialfall das Exponential-Modell.

Ein Parameter γ_j ist im AFT-Modell das über t konstante Verhältnis zweier Quantile von $S(t)$, wenn sich der Prädiktor X_j um eine Einheit erhöht. Ein exponenzierter Parameter e^{γ_j} gibt analog den Änderungsfaktor für die Überlebenszeit bei einer Änderung von X_j um eine Einheit an. Zwischen dem Parameter β_j in der Darstellung als PH-Modell und dem Parameter γ_j in der Formulierung als AFT-Modell besteht für Weibull-verteilte Überlebenszeiten die Beziehung $\beta_j = -\frac{\gamma_j}{a}$, für den Spezialfall exponentialverteilter Überlebenszeiten also $\beta_j = -\gamma_j$.

9.5.3 Anpassung und Modelltests

AFT-Modelle können mit `survreg()` aus dem Paket `survival` angepasst werden.

```
> survreg(⟨Modellformel⟩, dist="⟨Verteilung⟩", data=⟨Datensatz⟩)
```

Als erstes Argument ist eine Modellformel zu übergeben, deren linke Seite ein mit `Surv()` erstelltes Objekt ist (Abschn. 9.2). Dabei ist sicherzustellen, dass alle Ereignis-Zeitpunkte $t_i > 0$ sind und nicht (etwa durch Rundung) Nullen enthalten. Die rechte Seite der Modellformel kann neben – zeitlich konstanten – kontinuierlichen Prädiktoren und Faktoren als besonderen Vorhersageterm `strata(⟨Faktor⟩)` umfassen. Dieser sorgt dafür, dass ein stratifiziertes Modell angepasst wird, das eine separate Baseline-Hazard-Funktion $\lambda_{0_g}(t)$ für jede Stufe g des Faktors beinhaltet. Stammen die in der Modellformel verwendeten Variablen aus einem Datensatz, ist dieser unter `data` zu nennen. Das Argument `dist` bestimmt die für T angenommene Verteilung – mögliche Werte sind unter Annahme proportionaler hazards `"weibull"` oder `"exponential"`.[7]

Die Modelle sollen hier für die in Abschn. 9.2.1 simulierten Daten mit zeitlich konstanten Prädiktoren und höchstens einmal auftretenden Ereignissen angepasst werden.

```
> library(survival)                          # für survreg()
> fitWeib <- survreg(Surv(obsT, status) ~ X + IV, dist="weibull",
+                    data=dfSurv)
```

Wald-Tests der Parameter sowie einen Likelihood-Quotienten-Test des Gesamtmodells erhält man mit `summary(⟨survreg-Objekt⟩)`.

```
> summary(fitWeib)                           # Parameter- und Modelltests
Call:
survreg(formula = Surv(obsT, status) ~ X + IV, data = dfSurv,
    dist = "weibull")
             Value Std. Error     z        p
(Intercept)  3.423     0.1703 20.10 7.45e-90
X           -0.632     0.1024 -6.17 6.65e-10
IVB          0.504     0.2449  2.06 3.94e-02
IVC         -0.778     0.2327 -3.34 8.29e-04
Log(scale)   0.216     0.0608  3.56 3.73e-04
```

[7] Lässt man die Annahme proportionaler hazards fallen, kommen auch weitere Verteilungen für T in Betracht, über die `?survreg` Auskunft gibt.

```
Scale= 1.24

Weibull distribution
Loglik(model)= -695.9   Loglik(intercept only)= -726.9
        Chisq= 62.03 on 3 degrees of freedom, p= 2.2e-13
Number of Newton-Raphson Iterations: 5
n= 180
```

Die Ausgabe ist weitgehend analog zu jener von `summary(⟨coxph-Objekt⟩` (Abschn. 9.4), wobei in der Spalte `Value` die geschätzten AFT-Parameter $\hat{\gamma}_j = -\hat{\beta}_j \cdot \hat{a}$ genannt werden.[8] Die Schätzung $\hat{a}$ des Formparameters der Weibull-Verteilung ist unter `Scale` aufgeführt. Der zugehörige Wald-Test mit der H_0: $\ln a = 0$ steht in der Zeile `Log(scale)`. Eine Alternative hierzu ist der Likelihood-Quotienten-Test des eingeschränkten Modells ⟨`fitR`⟩ mit Exponentialverteilung ($a = 1$, also $\ln a = 0$) gegen das umfassendere Modell ⟨`fitU`⟩ mit Weibull-Verteilung mittels `anova(⟨fitR⟩, ⟨fitU⟩)` (Abschn. 8.1.5).

```
# eingeschränktes Modell mit a=1 -> Exponentialverteilung
> fitExp <- survreg(Surv(obsT, status) ~ X + IV, dist="exponential",
+                   data=dfSurv)

> anova(fitExp, fitWeib)                    # LQ-Modelltest
   Terms Resid. Df    -2*LL Test Df Deviance     Pr(>Chi)
1 X + IV       176 1405.946      NA       NA           NA
2 X + IV       175 1391.839    =  1 14.10752 0.0001726517
```

Da der hier im Modell berücksichtigte Faktor `IV` mit mehreren Parametern γ_j assoziiert ist, muss seine Signifikanz insgesamt über einen Modellvergleich getestet werden. Dazu dient wie beim Vergleich der Modelle mit Exponential- und Weibull-Verteilung ein Likelihood-Quotienten-Test zweier hierarchischer Modelle.

```
# eingeschränktes Modell ohne Faktor IV
> fitR <- survreg(Surv(obsT, status) ~ X, dist="weibull", data=dfSurv)
> anova(fitR, fitWeib)                      # LQ-Modelltest für Faktor IV
   Terms Resid. Df    -2*LL Test Df Deviance     Pr(>Chi)
1      X       177 1418.773      NA       NA           NA
2 X + IV       175 1391.839  +IV  2 26.93433 1.416721e-06
```

9.5.4 Survival-Funktion schätzen

Die geschätzte Verteilungsfunktion $\hat{F}(t)$ für ein mit `survreg()` angepasstes Modell ermittelt `predict()` (Abschn. 6.4). Die meist stattdessen betrachtete geschätzte Survival-Funktion ergibt sich daraus als $\hat{S}(t) = 1 - \hat{F}(t)$.

[8] Für die $\hat{\beta}_j$ ergibt sich 0.51 (`X`), −0.41 (`IVB`) und 0.63 (`IVC`) – also Schätzungen, die hier denen des Cox PH-Modells sehr ähnlich sind (s. S. 338).

```
> predict(⟨survreg-Objekt⟩, newdata=⟨Datensatz⟩, type="quantile",
+         p=⟨Quantile⟩, se=TRUE)
```

Als erstes Argument ist ein `survreg`-Objekt zu übergeben. Das Argument `newdata` erwartet einen Datensatz, der neue Daten für Variablen mit denselben Namen, und bei Faktoren zusätzlich denselben Stufen wie jene der ursprünglichen Prädiktoren im `survreg`-Objekt enthält. Mit dem Argument `type="quantile"` liefert `predict()` für jede Zeile in `newdata` für das Quantil $p \in (0, 1)$ den Wert $\hat{F}^{-1}(p)$. Dies ist die Überlebenszeit t_p, für die bei den in `newdata` gegebenen Gruppenzugehörigkeiten und Prädiktorwerten $\hat{F}(t_p) = p$ gilt. Dafür ist an `p` ein Vektor mit Quantilen zu übergeben, deren zugehörige Werte von $\hat{F}(t)$ gewünscht werden. Den geschätzten Median der Überlebenszeit erfährt man etwa mit `p=0.5`, während für einen durchgehenden Funktionsgraphen von $\hat{F}(t)$ bzw. $\hat{S}(t)$ eine fein abgestufte Sequenz im Bereich $(0, 1)$ angegeben werden muss. Setzt man `se=TRUE`, erhält man zusätzlich noch die geschätzte Streuung für t_p.

Mit `se=TRUE` ist das zurückgegebene Objekt eine Liste mit den Werten von $\hat{F}^{-1}(p)$ in der Komponente `fit` und den geschätzten Streuungen in der Komponente `se.fit`. Umfasst `newdata` mehrere Zeilen, sind `fit` und `se.fit` Matrizen mit einer Zeile pro Beobachtungsobjekt und einer Spalte pro Quantil.[9]

```
# Datensatz: 2 Männer mit Prädiktor X=0 in Gruppe A bzw. in C
> dfNew <- data.frame(sex=factor(c("m", "m"), levels=levels(dfSurv$sex)),
+                     X=c(0, 0),
+                     IV=factor(c("A", "C"), levels=levels(dfSurv$IV)))

# geschätzte Werte von F^(-1)
> percs <- (1:99)/100                          # Perzentile = Quantile
> FWeib <- predict(fitWeib, newdata=dfNew, type="quantile",
+                  p=percs, se=TRUE)

# stelle geschätzte Survival-Funktion S(t) statt F(t) dar -> 1-percs
# zunächst für Beobachtungsobjekt 1
> matplot(cbind(FWeib$fit[1, ],
+               FWeib$fit[1, ] - 2*FWeib$se.fit[1, ],
+               FWeib$fit[1, ] + 2*FWeib$se.fit[1, ]), 1-percs,
+         type="l", main=expression(paste("Weibull-Fit ", hat(S)(t),
+         " mit SE")), xlab="t", ylab="Survival", lty=c(1, 2, 2),
+         lwd=2, col="blue")

# für Beobachtungsobjekt 2
> matlines(cbind(FWeib$fit[2, ],
+                FWeib$fit[2, ] - 2*FWeib$se.fit[2, ],
+                FWeib$fit[2, ] + 2*FWeib$se.fit[2, ]), 1-percs,
+          col="red", lwd=2)

> legend(x="topright", lwd=2, lty=c(1, 2, 1, 2),
+        col=c("blue", "blue", "red", "red"),
```

[9] Abschnitt 9.4.2 demonstriert die analoge Verwendung von `newdata` in `survfit()`.

```
+        legend=c("sex=m, X=0, IV=A", "+- 2*SE",
+                 "sex=m, X=0, IV=C", "+- 2*SE"))
```

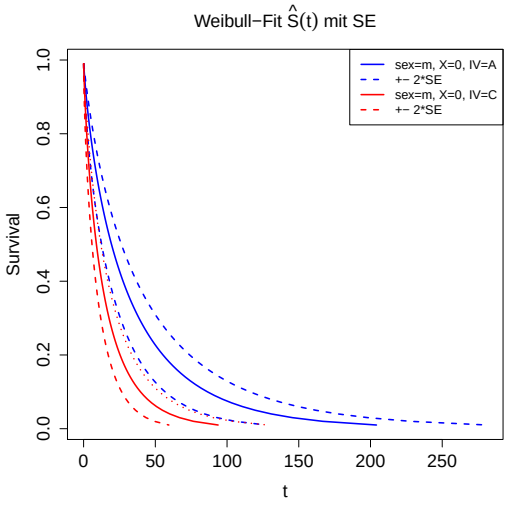

Abbildung 9.7: Schätzung der Survival-Funktion $\hat{S}(t)$ aus Weibull-Modell für zwei männliche Personen mit Prädiktorwert $X = 0$ aus Gruppe `A` bzw. `C`