PH 716 Applied Survival Analysis

Part V: Cox Proportional Hazards Model

Zhiyang Zhou (zhou67@uwm.edu, zhiyanggeezhou.github.io)

2024/Mar/04 14:17:38

Assumptions for Cox proportional hazards (PH) model

- Observed $\widetilde{T}_i = \tilde{t}_i$ and $\Delta_i = \delta_i$ (event indicator)
- T_i are independent across i, given $x_{i1}, \ldots, x_{ip}$
- The independent and non-informative censoring
- $\lambda_{T_i}(t) = \lambda(t \mid x_{i1}, \ldots, x_{ip}) = \lambda_0(t) \exp(\sum_{j=1}^p x_{ij}\beta_j)$, or equiv. $\ln \lambda_{T_i}(t) = \ln \lambda_0(t) + \sum_{j=1}^p x_{ij}\beta_j$
  - $\lambda_0(t)$ (the baseline hazard): obtained when all covariates are zeros and left unspecified
    - A semi-parametric generalized linear model: nonparmetric baseline hazard + paramatric proportion
  - Proportional hazards: the HR between any two individuals, say $\lambda_{T_{i_1}}(t)/\lambda_{T_{i_2}}(t) = \exp(\sum_{j=1}^p x_{i_1 j}\beta_j - \sum_{j=1}^p x_{i_2 j}\beta_j)$, is constant over time
  - No intercept β_0
  - Interpretation of β_j: $\exp(\beta_j)$ is the HR associated with one-unit change of the jth covariate, fixing everything else

Weibull regression as a special case of PH models

- Recall the Weibull regression: $\ln T_i = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j + \sigma\varepsilon_i$ with $\varepsilon_i \overset{\text{iid}}{\sim} F_{\varepsilon_i}(\epsilon) = 1 - \exp(-\exp\epsilon)$
  - $S_{T_i}(t) = \exp[-\{t/\exp(\beta_0 + \sum_{j=1}^p x_{ij}\beta_j)\}^{1/\sigma}] \Rightarrow \lambda_{T_i}(t) = (1/\sigma)t^{1/\sigma - 1}\exp\{(-\beta_0 - \sum_{j=1}^p x_{ij}\beta_j)/\sigma\}$
- $\lambda_{T_i}(t) = \lambda_0(t)\exp(\sum_{j=1}^p x_{ij}\beta_j^*)$ if $\lambda_0(t) = (1/\sigma)t^{1/\sigma-1}\exp(-\beta_0/\sigma)$ and $\beta_j^* = -\beta_j/\sigma$, $j = 1, \ldots, p$
- The only continuous-time model that is both a PH and an AFT model

Partial likelihood (assuming no tied failure time)

- The observed-data likelihood $L(\boldsymbol{\beta}, \lambda_0) = \prod_i \lambda_{T_i}(\tilde{t}_i)^{\delta_i} S_{T_i}(\tilde{t}_i)$ relying on both $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_j]^\top$ and unspecified $\lambda_0(\cdot)$
- Further assumptions
  - K and only K distinct, ordered failure times, say $t_1 < \cdots < t_K$
  - No tied failure time: for each k, there is one and only one individual, say subject i_k, who fails at t_k
  - Risk set $\mathcal{R}(t) = \{i : \widetilde{T}_i \geq t\}$: the set of individuals who are known to survive just prior to time t
- Rephrase $L(\boldsymbol{\beta}, \lambda_0)$:

$$L(\boldsymbol{\beta}, \lambda_0) = \prod_{i=1}^n \lambda_{T_i}(\tilde{t}_i)^{\delta_i} S_{T_i}(\tilde{t}_i) = \prod_{i=1}^n \left\{ \frac{\lambda_{T_i}(\tilde{t}_i)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \lambda_{T_\ell}(\tilde{t}_i)} \right\}^{\delta_i} \times \left\{ \sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \lambda_{T_\ell}(\tilde{t}_i) \right\}^{\delta_i} \times S_{T_i}(\tilde{t}_i)$$

- Take the partial likelihood (i.e., the first term of the above $L(\boldsymbol{\beta}, \lambda_0)$)

$$pL(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left\{ \frac{\lambda_{T_i}(\tilde{t}_i)}{\sum_{k \in \mathcal{R}(\tilde{t}_i)} \lambda_{T_k}(\tilde{t}_i)} \right\}^{\delta_i} = \prod_{i=1}^{n} \left\{ \frac{\exp(\sum_{j=1}^{p} x_{ij}\beta_j)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)} \right\}^{\delta_i} = \prod_{k=1}^{K} \frac{\exp(\sum_{j=1}^{p} x_{i_k j}\beta_j)}{\sum_{\ell \in \mathcal{R}(t_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)}$$

  as a surrogate of $L(\boldsymbol{\beta}, \lambda_0)$ in estimating $\boldsymbol{\beta}$
  - Cox (1972) argued that $pL(\boldsymbol{\beta})$ contained almost all the information about $\boldsymbol{\beta}$
  - Extensive evidence, both theoretical and numerical, supported this argument in the past few decades
  - $\frac{\exp(\sum_{j=1}^{p} x_{i_k j}\beta_j)}{\sum_{\ell \in \mathcal{R}(t_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)}$: the probability of selecting a particular person (here subject i_k) from the risk set at time t_k

- Log-partial likelihood

$$p\ell(\boldsymbol{\beta}) = \ln pL(\boldsymbol{\beta}) = \sum_{k=1}^{K} \left\{ \sum_{j=1}^{p} x_{kj}\beta_j - \ln \sum_{\ell \in \mathcal{R}(t_k)} \exp\left(\sum_{j=1}^{p} x_{\ell j}\beta_j \right) \right\}$$

Ex. 5.1 The calculation of partial likelihood

i	$\tilde{t}_i$	δ_i	x_i
1	9	1	4
2	8	0	5
3	6	1	7
4	10	1	3

- Follow the following definition (without reordering failure times) and fill in the table

$$pL(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left\{ \frac{\exp(\sum_{j=1}^{p} x_{ij}\beta_j)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)} \right\}^{\delta_i}$$

i	$\tilde{t}_i$	δ_i	x_i	$\mathcal{R}(\tilde{t}_i)$	$\left\{ \frac{\exp(x_i\beta)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \exp(x_\ell\beta)} \right\}^{\delta_i}$
1	9	1	4		
2	8	0	5		
3	6	1	7		
4	10	1	3		

Ex. 5.2 The calculation of partial likelihood: comparison of two groups

- Covariate x_i indicating the group label

i	$\tilde{t}_i$	δ_i	x_i	$\mathcal{R}(\tilde{t}_i)$	$\frac{\exp(x_i\beta)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_i)} \exp(x_\ell\beta)}$
1	4	0	0		
2	7	1	0		
3	8	0	0		
4	9	1	0		

i	$\tilde{t}_i$	δ_i	x_i	$\mathcal{R}(\tilde{t}_i)$	$\frac{\exp(x_i\beta)}{\sum_{\ell\in\mathcal{R}(\tilde{t}_i)}\exp(x_\ell\beta)}$
5	10	0	0		
6	3	1	1		
7	5	1	1		
8	5	0	1		
9	6	1	1		
10	8	0	1		

```
library(survival)
data = data.frame(
  tte = c(4,7,8,9,10,3,5,5,6,8),
  delta = c(0,1,0,1,0,1,1,0,1,0),
  x = c(0,0,0,0,0,1,1,1,1,1)
)
fit = coxph(Surv(tte,delta)~x, data = data)
summary(fit)
```

- $\exp(\beta)$ is the hazard ratio of $group = 1$ against $group = 0$, fixing covariates other than (if any). It implies that one jumps from $group = 0$ to $group = 1$ the hazard would be inflated by $(\exp(\beta) - 1) \times 100\%$.
- Is there any difference between the survival of the two groups? There are at least four p-values. Which one shall we refer to?
- What are meanings of other digits in the output?
- What if there are more covariates?

Ex. 5.3. Leukemia data (with tied event/failure times)

```
survival::leukemia
```

Partial likelihood (Cox's modification)

- Assumptions
  - K and only K distinct, ordered failure times, say $t_1 < \cdots < t_K$
  - d_k failures at time t_k: there are d_k individuals, say subject $i_{k,1}, \ldots, i_{k,d_k}$, who fail at t_k
- Accordingly

$$pL(\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\prod_{i\in\{i_{k,1},\ldots,i_{k,d_k}\}} \exp(\sum_j x_{ij}\beta_j)}{\sum_{D(d_k)\subset\mathcal{R}(t_k)} \prod_{i\in D(d_k)} \exp(\sum_j x_{ij}\beta_j)}$$

  - $D(d_k) \subset \mathcal{R}(t_k)$: a subset of $\mathcal{R}(t_k)$ containing d_k subjects, i.e., a set of d_k individuals who are at risk at t_k
- Labeled as `exact` by `survival::coxph`

Partial likelihood (Breslow's approximation)

- Keeping the assumptions for the Cox's modification
- Substituting $\{\sum_{\ell\in\mathcal{R}(t_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)\}^{d_k}$ for the denominator of the Cox's modification

$$pL(\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{\prod_{i\in\{i_{k,1},\ldots,i_{k,d_k}\}} \exp(\sum_j x_{ij}\beta_j)}{\{\sum_{\ell\in\mathcal{R}(t_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)\}^{d_k}}$$

- Default tie-handling method in SAS

Partial likelihood (Efron's approximation)

- Keeping the assumptions for the Cox's modification
- Substitute $\{\sum_{\ell\in\mathcal{R}(t_k)}\exp(\sum_{j=1}^{p}x_{\ell j}\beta_j)\}^{d_k}$ for the denominator of Cox's modification

$$pL(\boldsymbol{\beta})=\prod_{k=1}^{K}\frac{\prod_{i\in\{i_{k,1},\ldots,i_{k,d_k}\}}\exp(\sum_j x_{ij}\beta_j)}{\prod_{m=1}^{d_k}\{\sum_{\ell\in\mathcal{R}(t_k)}\exp(\sum_{j=1}^{p}x_{\ell j}\beta_j)-\frac{m-1}{d_k}\sum_{i\in\{i_{k,1},\ldots,i_{k,d_k}\}}\exp(\sum_j x_{ij}\beta_j)\}}$$

- Default tie-handling method by `survival::coxph`

Summary of handling ties

- With no ties, all approximation options give exactly the same results
- With only a few ties, all approximations yield pretty much the same results
- With many ties (relative to the number at risk), both of Breslow's and Efron's approximations yield coefficients $\boldsymbol{\beta}$ that are biased toward 0.
- Computing time of Cox's method is substantially longer than that of approximate methods. But it is not a big issue with today's hardwares.
- The Efron's approximation almost always works better than the Breslow's method, without consuming more time.

Revisit Ex. 5.3. Leukemia data (with tied event/failure times)

```
library(survival)
data = survival::leukemia
fit1 = coxph(Surv(time,status)~x, data = data)
fit2 = coxph(Surv(time,status)~x, data = data, ties = 'efron')
fit3 = coxph(Surv(time,status)~x, data = data, ties = 'breslow')
fit4 = coxph(Surv(time,status)~x, data = data, ties = 'exact')
c(coef(fit1), coef(fit2), coef(fit3), coef(fit4))
```

CIs and hypothesis tests for HRs

- Suppose the HR of interest is the one associated with the one-unit increase of the jth covairate, i.e., $\exp(\beta_j)$
- $\text{var}\{\exp(\hat{\beta}_j)\}\approx\exp(2\hat{\beta}_j)\text{var}(\hat{\beta}_j)$ (delta method)
  - Hence $\text{se}(\exp(\hat{\beta}_j))\approx\exp(\hat{\beta}_j)\text{se}(\hat{\beta}_j)$
- 95% CI for $\exp(\beta_j)$
  - $\exp(\hat{\beta}_j)\pm\Phi^{-1}(.975)\times\text{se}(\exp(\hat{\beta}_j))$
    - $\Phi^{-1}(.975)$ (≈ 1.96): the .975 quantile of $N(0,1)$
  - $\exp(\hat{\beta}_j\pm\Phi^{-1}(.975)\times\text{se}(\hat{\beta}_j))$ (preferred; why?)
- Hypothesis test for $H_0:\exp(\beta_j)=1$ (i.e., $\beta_j=0$) vs. H_1 : otherwise.
  - Wald test statistic: $\hat{\beta}_j/\text{se}(\hat{\beta}_j)\approx N(0,1)$ under H_0
    - Equivalent to checking whether $\exp(\hat{\beta}_j\pm\Phi^{-1}(.975)\times\text{se}(\hat{\beta}_j))$ covers 1
- LRT to compare two nested models
  - Model 1 nested to Model 2

    * Model 1: $\lambda(t \mid x_{i1}, \ldots, x_{ip}) = \lambda_0(t) \exp(\sum_{j=1}^{p} x_{ij}\beta_j)$
    * Model 2: $\lambda(t \mid x_{i1}, \ldots, x_{ip}, x_{i,q+1}, \ldots, x_{i,p+q}) = \lambda_0(t) \exp(\sum_{j=1}^{p+q} x_{ij}\beta_j)$
  - H_0 : Model 1 is correct (i.e., $\beta_{p+1} = \cdots = \beta_{p+q} = 0$) vs. H_1 : Model 2 is correct
  - Test statistic: $2(\ln L_{\text{Model2}} - \ln L_{\text{Model1}}) \approx \chi^2(q)$ under H_0

Ex. 5.4. Nursing home data

- Variables:
  - ID: Patient ID
  - lstay: Length of stay of a resident (in days)
  - age: Age of a resident
  - trt: Nursing home assignment (1: receive treatment, 0: control)
  - gender: Gender (1:male, 0:female)
  - marstat: Marital status (1: married, 0: not married)
  - hlstat: Health status (2: second best, 5: worst)
  - cens: Censoring indicator (1:censored, 0: discharged)

```
options(digits=4)
library(survival)
data = read.csv("NursingHome.csv")
data$event <- 1-data$cens
head(data)
data$trt = factor(data$trt) # not necessary because it is of two levels
data$gender = factor(data$gender) # not necessary because it is of two levels
data$marstat = factor(data$marstat) # not necessary because it is of two levels
data$hlstat = factor(data$hlstat) # necessary because it is of more than two levels

fit1 <- coxph(Surv(lstay,event) ~ trt + age + gender + marstat + hlstat, data=data)
summary(fit1)

# Testing if trt is necessary against the full model
fit2 <- coxph(Surv(lstay,event) ~ age + gender + marstat + hlstat, data=data)
anova(fit1, fit2)
summary(fit2)

# Testing if trt, age and marstat are necessary against the full model
fit3 <- coxph(Surv(lstay,event) ~ gender + hlstat, data=data)
anova(fit1, fit3)
summary(fit3)
```

Estimating the baseline hazard

- Have to maximize the likelihood $L(\beta, \lambda_0)$ instead of the partial likelihood $pL(\beta)$
  - Assuming $\lambda_0(\cdot)$ as piecewise constant between uncensored failure time, Breslow (1972) proved that
    * $L(\beta, \lambda_0)$ and $pL(\beta)$ share the identical maximizer, say $\hat{\beta}$, with respect to β
    * The maximizer of $L(\beta, \lambda_0)$ with respect to λ_0, say $\hat{\lambda}_0$, satisfies that

$$\hat{\Lambda}_0(t) = \sum_{k:t_k \le t} \frac{d_k}{\sum_{\ell \in \mathcal{R}(t_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\hat{\beta}_j)}$$

      · $\hat{\Lambda}_0(t)$: Breslow estimator of the baseline cumulative hazard rate, reducing to the NA estimator (Lecture Note Part II) if all covariates are zeros
      · d_k: # of events at t_k
      · $\mathcal{R}(t_k)$: the risk set at t_k

- $\widehat{S}_{T_i}(t) = \exp\{-\hat{\Lambda}_0(t)\}^{\exp(\sum_{j=1}^{p} x_{ij}\hat{\beta}_j)} = \widehat{S}_0(t)^{\exp(\sum_{j=1}^{p} x_{ij}\hat{\beta}_j)}$
  - $\widehat{S}_0(t) = \exp\{-\hat{\Lambda}_0(t)\}$: estimated baseline survival function

Ex. 5.4. Revisit the nursing home data

```
options(digits=4)
library(survival)
data.ex54 = read.csv("NursingHome.csv")
data.ex54$event <- 1-data.ex54$cens
data.ex54$marstat = factor(data.ex54$marstat) # not necessary because it is of two levels
data.ex54$hlstat = factor(data.ex54$hlstat) # necessary because it is of more than two levels
fit.ex54 <- coxph(Surv(lstay,event) ~ marstat + hlstat, data=data.ex54)
## P.S. note the mandantory scaling of covariates in `survival::coxph`

# baseline hazard and survival
baseline <- basehaz(fit.ex54, centered = FALSE)
names(baseline)[1] = 'cum.haz'
baseline$surv = exp(-baseline$cum.haz)
baseline

# Plot the survival function with given values of covariates
newdata.ex54 <- data.frame(
  marstat = factor(c(0,0,1,1)),
  hlstat = factor(c(2,5,2,5))
)
newdata.ex54
cox.predicted.survival <- survfit(fit.ex54, newdata=newdata.ex54)
plot(
  cox.predicted.survival, lty=1:4, col=1:4, lwd=2,
  xlab="Survival Time", ylab="Estimated Probability"
)
legend(
  "topright",
  c(
    "Not married, health status second best",
    "Not married, health status worst",
    "Married, health status second best",
    "Married, health status worst"
  ),
  lty=1:4, col=1:4, lwd=2
)
```

Model checking

- Cox-Snell residuals: assessing the overall fit of the final model
- Martingale residuals: determining the functional form of a covariate included in the model
- Deviance residuals: detecing outliers
- Schoenfeld residuals residuals: checking the appropriateness of the PH assumption

Cox-Snell residuals

- Inverse cdf theorem: arbitrary r.v. X with cdf $F_X(x) = \Pr(X \leq x) \Rightarrow F_X(X) \sim U(0,1)$

- It follows that $T_i \overset{\text{independent}}{\sim} S_{T_i}(\cdot) \Rightarrow S_{T_i}(T_i) \overset{\text{iid}}{\sim} U(0,1) \Rightarrow \Lambda_{T_i}(T_i) = -\ln S_{T_i}(T_i) \overset{\text{iid}}{\sim} \exp(1)$
- Cox-Snell residuals: $r_{i,\text{CS}} = \widehat{\Lambda}_{T_i}(\widetilde{T}_i)$
  - $\widehat{\Lambda}_{T_i}(\cdot)$: estimated $\Lambda_{T_i}(\cdot)$ given by the Cox PH model
  - $\{(r_{i,\text{CS}}, \Delta_i) : i = 1, \ldots, n\}$ is a right-censored dataset
    - $r_{i,\text{CS}} = \min(H_i, \widehat{\Lambda}_{T_i}(C_i)) \Leftarrow \widetilde{T}_i = \min(T_i, C_i)$ and monotonically ascending $\Lambda_{T_i}(\cdot)$
      - $H_i = \widehat{\Lambda}_{T_i}(T_i) \approx \Lambda_{T_i}(T_i) \Rightarrow \Lambda_{H_i}(t) \approx t$
  - $\widehat{\Lambda}_{H_i,\text{NA}}(t) \approx t$
    - $\widehat{\Lambda}_{H_i,\text{NA}}(\cdot)$: NA estimator of $\Lambda_{H_i}(\cdot)$ based on $\{(r_{i,\text{CS}}, \Delta_i) : i = 1, \ldots, n\}$
- Cox-Snell residual plot
  - For uncensored subjects
    - Compare $r_{i,\text{CS}}$ to exp(1) samples via Q-Q plot
    - Or, plot $\widehat{\Lambda}_{H_i,\text{NA}}(\tilde{t}_i)$ against $\tilde{t}_i$
  - Used to diagnose poor model fit
  - No insight into how model assumptions are violated

Ex. 5.5 [KM, Example 11.1]

- This multi-center acute leukemia study consists of 137 patients with acute myelocytic leukemia (AML) or acute lymphoblastic leukemia (ALL) aged 7 to 52 from March 1, 1984 to June 30, 1989 at four institutions.
- The disease-free survival time (`t2`) on study is defined as time (in days) to relapse or death
- `d3` is the disease free survival indicator: 1 - Dead or Relapsed, 0 - Alive Disease Free.
- Focus on effects of the following 9 covariates on disease-free survival:
  - `z1`: Patient age in years.
  - `z2`: Donor age in years.
  - `z3`: Patient sex: 1 - Male, 0 - Female.
  - `z4`: Doner sex: 1 - Male, 0 - Female.
  - `z5`: Patient Cytomegalovirus (CMV) status: 1 - CMV positive, 0 - CMV negative.
  - `z6`: Donor CMV status: 1 - CMV positive, 0 - CMV negative.
  - `z7`: Waiting time to transplant in days.
  - `z8`: French–American–British classification (FAB): 1 - FAB Grade 4 or 5 and AML, 0 - otherwise.
  - `z10`: Methotrexate (MTX): used as a Graft-Versus-Host-Prophylactic 1 - Yes, 0 - No.

```
options(digits=4)
library(survival)
# model fitting
data.ex55 = read.csv("bmt.csv")
fit.ex55 <- coxph(Surv(t2,d3) ~ z1+z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55)

# Cox-Snell residual
r.cs = data.ex55$d3-residuals(fit.ex55, type='martingale') # Cox-Snell

# Cox-Snell residual plot
set.seed(2024)
exp.rnd = rexp(10000)
qqplot(
  x = exp.rnd, y = r.cs[as.logical(data.ex55$d3)],
  xlab = "Theoretical Quantiles", ylab = "Sample Quantiles"
)
```

```r
qqline(r.cs[as.logical(data.ex55$d3)], distribution = qexp)
# Or
cum.haz.r.cs <- basehaz(coxph(Surv(r.cs, d3)~1, data=data.ex55), centered = FALSE)
plot(
  x=cum.haz.r.cs[,2], y=cum.haz.r.cs[,1],
  xlab='t', ylab='Cumulative hazard of r.cs'
)
abline(a=0,b=1,col='red')
```

Martingale residuals

- Martingale residuals: $r_{i,\mathrm{M}} = \Delta_i - r_{i,\mathrm{CS}}$
  - [KM, pp. 360] Zero-sum: $\sum_i r_{i,\mathrm{M}} = 0$
  - Estimated excess number of events seen in the data but not predicted by the model
    - Positive $r_{i,\mathrm{M}}$: the patient died sooner than expected
    - Negative $r_{i,\mathrm{M}}$: the patient lived longer than expected (or were censored)
  - Analogous to the residuals in linear models
    - But asymmetric and unbounded from below
- Examining the best functional form for a given covariate
  1. Partition covariates into two parts:
     - $x_{i2}, \ldots, x_{ip}$: for which we know their proper functional form, say $f_2(\cdot), \ldots, f_p(\cdot)$, respectively
     - x_{i1}: a single covariate for which there is a potential functional form $f_1(\cdot)$
  2. If $f_1(\cdot)$ is best for x_{i1} and x_{i1} is independent of other covariates, then fit the Cox PH model without the jth covariate $\lambda_{T_i}(t) = \lambda_0(t) \exp\{\sum_{j=2}^{p} f_j(x_{ij})\beta_j\}$ and compute martingale residuals $r_{i,\mathrm{M}}$
  3. Confirm f_1 via the scatterplot of $r_{i,\mathrm{M}}$ against x_{i1} with a fitted loess (locally estimated scatterplot smoothing) line
     - If the fitted loess line is linear, then no transformation of x_{i1} is needed; otherwise, a discretized version/transformation of x_{i1} is indicated
  4. Fitting $\lambda_{T_i}(t) = \lambda_0(t) \exp\{\sum_{j=2}^{p} f_j(x_{ij})\beta_j\} \exp\{f_1(x_{i1}\beta_1)\}$ and check the scatterplot of updated $r_{i,\mathrm{M}}$ against x_{i1} with a fitted loess (locally estimated scatterplot smoothing) line
     - If the fitted loess line is overlapping the x-axis, then no transformation of x_{i1} is needed.
- Theoretical notes:
  - Why would the martingale residuals reveal the correct functional forms of covariates?
    - Because $E(r_{i,\mathrm{M}}) \approx (n_D/n)\{f_1(x_{i1}) - C\}$ [KM, pp. 362]
      - n_D/n: the ratio of total number of events to total number of subjects
      - C: a constant
  - Why is the residual bearing such a name?
    - Martingale: a stochastic process $M(t)$ such that $\mathrm{E}\{M(t)\} = 0$ and $\mathrm{E}\{M(t) \mid M(s)\} = M(s)$ for all $s < t$
    - $r_{i,\mathrm{M}}$ obtained by evaluating a martingale at $\tilde{t}_i$
  - The zero-sum of martingale residuals is specific for Cox PH model with the Breslow estimator for the baseline hazard

Revisit Ex. 5.5

```r
options(digits=4)
library(survival)
# [DM, pp. 208] a function to add the smooth curve and confidence limits
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve
  # make list of x values
```

```
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))
  # Then fit loess function through the points (xx, yy)
  # at the listed values
  yy.xx <- predict(loess(yy ~ xx), se=T,
  newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
}

# model fitting without z1
data.ex55 = read.csv("bmt.csv")
fit.ex55 <- coxph(Surv(t2,d3) ~ z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55, ties = 'exact')

# Martingale residual plot (for the model without z1) vs. z1
r.m = residuals(fit.ex55, type='martingale')
sum(r.m)
plot(
  x=data.ex55$z1, y=r.m,
  main = 'Martingale residuals \n (for the model without z1) \n versus z1')
smoothSEcurve(r.m, data.ex55$z1)
## indicating a cubic function?

# model fitting with a cubic function of z1
fit.ex55.1 <- coxph(Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55)

# Martingale residual plot (for the model with a cubic function of z1) vs. z1
r.m.1 = residuals(fit.ex55.1, type='martingale')
plot(
  x=data.ex55$z1, y=r.m.1,
  main = 'Martingale residual \n (for the model with a cubic function of z1) \n versus z1')
smoothSEcurve(r.m.1, data.ex55$z1)
```

Deviance residuals

- Outlier: an observation for which the outcome is not sufficiently well predicted by the fitted model
- Deviance residuals: $r_{i,\mathrm{D}} = \mathrm{sign}(r_{i,M})\sqrt{-2\{r_{i,M} + \delta_i \ln(\delta_i - r_{i,M})\}}$
  - Symmetrically distributed with expected value 0 (if the fitted model is correct); de-skewed/transformed martingale residuals
    - $r_{i,\mathrm{D}} = 0 \Leftrightarrow r_{i,M} = 0$
    - Inflating $r_{i,\mathrm{D}}$ when $r_{i,M}$ is close to 1
    - Shrinking large negative $r_{i,M}$
  - Analogous to the deviance in GLMs
- Detecting outliers: plotting $r_{i,\mathrm{D}}$ against $\sum_{j=1}^{p} x_{ij}\hat{\beta}_j$ (called linear predictors or risk scores)
  - With moderate (or less) censoring, this plot should look like randomly-distributed noise without discernible pattern
  - Large absolute values of deviance residuals indicating observations that are poorly explained by the model, potentially pointing to outliers or influential points
    - 95% of absolute deviance residuals ≤ 2

* 99.7% of absolute deviance residuals ≤ 3

Revisit Ex. 5.5

```
options(digits=4)
library(survival)
# [DM, pp. 208] a function to add the smooth curve and confidence limits
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve
  # make list of x values
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))
  # Then fit loess function through the points (xx, yy)
  # at the listed values
  yy.xx <- predict(loess(yy ~ xx), se=T,
  newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
}

# model fitting
fit.ex55.1 <- coxph(
  Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10,
  data=data.ex55,
  x = T
)

# Two ways to calculate linear predictors
risk.score.1 = fit.ex55.1$x %*% coef(fit.ex55.1)
risk.score.2 = fit.ex55.1$linear.predictors
sum((risk.score.1-risk.score.2)^2) # seems distinct?

# Deviance residual plot vs. risk scores
r.d = residuals(fit.ex55.1, type='deviance')
plot(
  x=risk.score.1, y=r.d,
  main = 'Deviance residuals \n versus risk scores')
smoothSEcurve(yy=r.d, xx=risk.score.1)

# Potential outliers
(1:nrow(data.ex55))[abs(r.d) > 2]
sum(abs(r.d) > 2)/nrow(data.ex55)
sum(abs(r.d) > 3)/nrow(data.ex55)
```

Schoenfeld residuals

- Schoenfeld residuals: for uncensored subject i and the jth covariate,

$$r_{ij,\mathrm{S}} = x_{ij} - \bar{x}_{\cdot j}$$

$-\ \bar{x}_{\cdot j} = \sum_{k \in \text{uncensored subjects}} w_{kj} z_{kj}$ with weights $w_{kj} = \frac{\exp(\sum_{j=1}^{p} x_{kj}\beta_j)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_k)} \exp(\sum_{j=1}^{p} x_{\ell j}\beta_j)}$

- Investigating the PH assumption: plotting $r_{ij,\mathrm{S}}$ versus the covariate x_{ij} for the j covariate
  - Points are centered at zero if the PH assumption holds
- Theoretical note [DM, Sec. 7.2.2]:
  - Schoenfeld residuals are components of the score function
  - $\Rightarrow \sum_{i\in\text{uncensored subjects}} r_{ij,\mathrm{S}} = 0$ for each j

Revisit Ex. 5.5

```
options(digits=4)
library(survival)
# [DM, pp. 208] a function to add the smooth curve and confidence limits
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve
  # make list of x values
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))
  # Then fit loess function through the points (xx, yy)
  # at the listed values
  yy.xx <- predict(loess(yy ~ xx), se=T,
  newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
}

# model fitting
fit.ex55.1 <- coxph(
  Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10,
  data=data.ex55,
  x=T
)

# Schoenfeld residual plot
r.s = residuals(fit.ex55.1, type='schoenfeld')
par(mfrow=c(2,2))
for (j in c(1,4,9)){
  plot(
    x=fit.ex55.1$x[data.ex55$d3==1,j], y=r.s[,j],
    main = paste0('Schoenfeld residuals \n versus ', colnames(fit.ex55.1$x)[j]),
    xlab = paste0(colnames(fit.ex55.1$x)[j]),
    ylab = 'Schoenfeld residuals'
  )
  smoothSEcurve(yy=r.s[,j], xx=fit.ex55.1$x[data.ex55$d3==1,j])
}
```