Analyse des durées de vie avec le logiciel R

Ségolen Geffray

Des outils ainsi que des données pour l'analyse des durées de vie sont disponibles dans les packages

`survival`

`MASS`

Il est nécessaire de charger ce package au début de **chaque** nouvelle session `R` par l'instruction suivante :

```
library(survival)
```

NB1 : pour voir quels sont les packages disponibles dans `R` :

```
library()
```

NB2 : pour voir ce que contient un package donné (**après** l'avoir chargé si nécessaire) :

```
help(package="survival")
```

Enumérons les fonctions que nous étudions aujourd'hui :

- `Surv` : *crée un "objet de survie"*
- `survfit` : *fournit une estimation de la fonction de survie (méthode actuarielle ou Kaplan-Meier...)*
- `survdiff` : *éxécute le test du log-rank*
- `coxph` : *ajuste un modèle de Cox*
- `cox.zph` : *teste l'ajustement d'un modèle de Cox aux données*
- `residuals` : *détermine différents types de résidus après ajustement d'un modèle de Cox aux données*

et, bien sûr, nous avons besoin de

- `data` : *pour charger un ou plusieurs jeux de données*
- `str` : *expose la structure d'un objet R*
- `summary` : *pour obtenir un résumé détaillé des résultats demandés*
- `print` : *pour obtenir un résumé court des résultats demandés*
- `par(mfrow=c(1,2))` : *pour couper la fenêtre graphique en deux*
- `plot` : *pour faire un graphique*

- `lines` : *pour rajouter des points ou des lignes sur un graphique déjà existant*
- `abline` : *pour rajouter une droite sur un graphique déjà existant*
- `legend` : *pour rajouter une légende sur un graphique déjà existant*
- `lowess` : *ajuste une courbe de régression lissée*
- ...

Les objects de survie

Les objets de survie sont créés au moyen de la fonction `Surv(time,status)` du package `survival`. Pour créer des données censurées à droite, cette fonction a besoin de 2 arguments :

- `time` : *durée réellement observée*
- `status` : *indicatrice qui vaut 0 ou 1 (resp FALSE ou TRUE, resp 1 ou 2) selon que l'observation correspond à une censure ou non*

L'instruction de base pour créer un object de survie à partir des colonnes `time` et `status` du data frame `mydata` est donc :

```
mydata.surv=Surv(time,status,data=mydata)
```

Lorsqu'il y a plus de 2 modalités pour `status` ou que les modalités ne sont pas parmi celles citées plus haut, il faut préciser quelles modalités correspondent aux observations non censurées, par exemple :

```
mydata.surv=Surv(time,status!=0,data=mydata)
```

Lorsque l'on travaille avec des covariables dépendantes du temps, l'object de survie est créé de la façon suivante :

```
mydata.surv=Surv(start,stop,event,data=mydata)
```

où la période totale d'observation d'un sujet est découpé en intervalles dont `start` et `stop` sont respectivement le début et la fin et où `event` est la version "time-dependent" de l'indicatrice `status` et est égale à 1 une fois que le décès (ou l'évènement) s'est produit.
NB : on peut aussi se servir de la fonction `Surv` pour créer des données censurées par intervalle ou censurées à gauche. Pour plus de détails, voir :

```
help(Surv)
```

Estimation de la fonction de survie

L'estimation d'une fonction de survie à partir d'un objet de survie (i.e. à partir de données censurées) se fait au moyen de la fonction `survfit` du package `survival`. L'instruction de base est :

```
survfit(Surv(time,status),data=mydata)
```

On peut souhaiter estimer la survie parmi des sous groupes déterminés par un ou plusieurs facteurs catégoriels :

```
survfit(Surv(time, status)~factor,data=mydata)
```

par exemple :

```
survfit(Surv(time,status)~sex,data=mydata)          (2 sous-groupes selon le sexe)
survfit(Surv(time,status)~sex+treat,data=mydata)    (4 sous-groupes selon le sexe et le traitement)
```

Des options possibles sont les suivantes :

```
survfit(formula,                                                  (ce qu'on fait avec les données)
        data,                                                     (le jeu de données que l'on considère)
        conf.int=.95,                                             (le niveau de confiance des intervalles de confiance)
        type=c("kaplan-meier","fleming-harrington"),              (l'estimateur de la fonction de survie employé)
        error=c("greenwood","tsiatis"),                           (l'estimateur de la variance employé)
        conf.type=c("log","log-log","plain","none")               (la formule des intervalles de confiance)
        )
```

Les valeurs par défaut sont `conf.int=.95,type="kaplan-meier",error="greenwood",conf.type="log"`.

L'estimateur de Kaplan-Meier de la fonction de survie à partir d'un échantillon $(T_i, D_i)_{i=1,\dots,n}$ d'observations éventuellement censurées est donné par :

$$\widehat{S}_{KM}(t) = \prod_{i=1}^{n} \left(1 - \frac{I(T_i \leq t, D_i = 1)}{Y(T_i)}\right)$$

où $Y(t)$ représente le nombre de sujets à risque (i.e. ni censurés, ni décédés) à l'instant t et où $I(A)$ représente la fonction indicatrice de l'évènement A et vaut 1 si l'évènement A est réalisé et 0 sinon. Une autre écriture de l'estimateur de Kaplan-Meier est la suivante. Ordonnons les durées observées non-censurées : $T'_1 < \ldots, T'_K$ où K est le nombre de durées non-censurées distinctes et posons $T'_{K+1} = \infty$. Notons d_j le nombre de morts dans l'intervalle $[T'_j, T'_{j+1}[$. Soit n_j le nombre d'individus à risque (i.e. ni morts, ni censurés) à l'instant T'_j. L'estimateur de Kaplan-Meier se définit alors comme :

$$\widehat{S}_{KM}(t) = \prod_{j=1}^{k} \left(\frac{n_j - d_j}{n_j}\right) \quad \text{où } k \text{ est défini par } T'_k \leq t < T'_{k+1}$$
$$\widehat{S}_{KM}(t) = 1 \quad \text{pour } t < T'_1.$$

L'estimateur de Fleming-Harrington (très sensible aux ex aequo) de la fonction de survie est donné par :

$$\widehat{S}_{FH}(t) = \exp\left(-\widehat{\Lambda}(t)\right)$$

où $\widehat{\Lambda}$ représente l'estimateur de Nelson-Aalen de Λ :

$$\widehat{\Lambda}(t) = \sum_{i=1}^{n} \frac{I(T_i \leq t, D_i = 1)}{Y(t_i)}$$

que l'on peut aussi écrire sous la forme suivante :

$$\widehat{\Lambda}(t) = \sum_{j=1}^{k} \frac{d_j}{n_j} \quad \text{où } k \text{ est défini par } T'_k \leq t < T'_{k+1}$$
$$\widehat{\Lambda}(t) = 0 \quad \text{pour } t < T'_1.$$

L'estimateur de Tsiatis (ou d'Aalen) de la variance de $\widehat{\Lambda}$ est donné par :

$$\widehat{\mathrm{Var}}_T\left(\widehat{\Lambda}(t)\right) = \sum_{j=1}^{k} \frac{d_j}{n_j^2} \quad \text{où } k \text{ est défini par } T'_k \leq t < T'_{k+1}$$
$$\widehat{\mathrm{Var}}_T\left(\widehat{\Lambda}(t)\right) = 0 \quad \text{pour } t < T'_1.$$

L'estimateur de Greenwood de la variance de $\widehat{\Lambda}$ est donné par :

$$\widehat{\mathrm{Var}}_G\left(\widehat{\Lambda}(t)\right) = \sum_{j=1}^{k} \frac{d_j}{n_j(n_j - d_j)} \quad \text{où } k \text{ est défini par } T'_k \leq t < T'_{k+1}$$
$$\widehat{\mathrm{Var}}_G\left(\widehat{\Lambda}(t)\right) = 0 \quad \text{pour } t < T'_1.$$

On note par $\widehat{\mathrm{se}}(X)$ l'estimateur de l'écart-type d'une variable aléatoire X défini par $\widehat{\mathrm{se}}(X) = \left(\widehat{\mathrm{Var}}(X)\right)^{1/2}$.

Les intervalles de confiance pour $\Lambda(t)$ permettent d'obtenir ceux pour $S(t)$. Les intervalles de confiance pour $\Lambda(t)$ sont donnés par :

$$\Lambda(t) \in \left[\widehat{\Lambda}(t) \pm z_{1-\alpha/2}.\, \widehat{\mathrm{se}}(\widehat{\Lambda}(t))\right]$$

où $z_{1-\alpha/2}$ est le quantile d'ordre $1-\alpha/2$ de la loi normale centrée et réduite i.e. $\mathbb{P}[\mathcal{N}(0,1) \leq z_{1-\alpha/2}] = 1-\alpha/2$. La formule `conf.type="log"` (log-survival scale) pour les intervalles de confiance pour $S(t)$ est obtenue par exponentiation des intervalles de confiance pour $\Lambda(t)$ sur la base de la formule $S(t) = \exp(-\Lambda(t))$:

$$S(t) \in \left[\exp\left(-\widehat{\Lambda}(t) \pm z_{1-\alpha/2}.\, \widehat{\mathrm{se}}(\widehat{\Lambda}(t))\right)\right].$$

La formule `conf.type="plain"` (plain scale) pour les intervalles de confiance pour $S(t)$ est obtenue par delta-méthode à partir des intervalles de confiance pour $\Lambda(t)$:

$$S(t) \in \left[\exp\left(-\widehat{\Lambda}(t)\right) \pm z_{1-\alpha/2}.\, \exp\left(-\widehat{\Lambda}(t)\right).\, \widehat{\mathrm{se}}(\widehat{\Lambda}(t))\right].$$

La formule `conf.type="log-log"` (log-hazard scale) pour les intervalles de confiance pour $S(t)$ est obtenue par double exponentiation à partir des intervalles de confiance pour $\log(\Lambda(t))$:

$$S(t) \in \left[\exp\left(-\exp\left(\log\left(-\log\left(\widehat{S}(t)\right)\right) \pm z_{1-\alpha/2}.\, \frac{\widehat{\mathrm{se}}(\widehat{\Lambda}(t))}{\widehat{\Lambda}(t)}\right)\right)\right].$$

NB : la fonction `survfit` peut être utilisée pour déterminer une prédiction de la fonction de survie après ajustement d'un modèle de Cox. Pour plus de détails, voir :

```
help(survfit)
```

Test d'une différence de survie entre plusieurs sous-groupes ou échantillons

La comparaison de la survie dans plusieurs groupes peut s'effectuer au moyen du test du log-rank ou du test de Wilcoxon.
NB1 : Le test du log-rank est plus performant lorsque les deux courbes de survie ne se croisent pas.
NB2 : lorsque les taux de hasard instantané sont proportionnels, le log-rank est le "meilleur" test que l'on puisse effectuer.

Le test d'une différence de survie statistiquement significative entre plusieurs sous-groupes ou échantillons se fait dans le logiciel `R` au moyen de la fonction `survdiff` du package `survival`. L'instruction de base pour un test sur un traitement est :

```
survdiff(Surv(time,status)~treatment,data=mydata)          (test du log-rank)
survdiff(Surv(time,status)~treatment,data=mydata,rho=1)    (test de Wilcoxon)
```

On peut souhaiter stratifier le test, par exemple pour tester la différence de survie en fonction du traitement reçu par des patients de différentes institutions dans le cadre d'une étude multi-centrique mais sans tester l'effet éventuellement associé à l'institution :

```
survdiff(Surv(time,status)~treatment+strata(inst),data=mydata)
```
(test du log-rank à plusieurs échantillons et à plusieurs strates)

Pour plus de détails, voir :

```
help(survdiff)
```

Ajustement d'un modèle de régression à hasards proportionnels sur des données

Rappelons que le modèle de Cox à hasards proportionnels s'écrit sous la forme :

$$\begin{aligned}\lambda(t/\mathbf{Z}) &= \lambda_0(t).\ \exp(\beta'.\mathbf{Z}) \\ &= \lambda_0(t).\ \exp(\beta^{(1)}Z^{(1)} + \cdots + \beta^{(p)}Z^{(p)})\end{aligned}$$

où $\mathbf{Z} = (Z^{(1)}, \ldots, Z^{(p)})'$ est un vecteur de longueur p de covariables que l'on n'autorise pas ici à dépendre du temps. Le vecteur $\beta = (\beta^{(1)}, \ldots, \beta^{(p)})'$ est un vecteur de paramètres. La fonction λ_0 est la fonction de hasard instantané dite "baseline".

L'ajustement d'un modèle de régression à hasards proportionnels sur des données (i.e. l'estimation des paramètres $\beta^{(1)}, \ldots, \beta^{(p)}$) se fait au moyen de la fonction `coxph` du package `survival`. L'instruction de base est :

```
coxph(formula,
      data,
      method=c("efron","breslow","exact"))
```

L'option `method` spécifie la méthode employée pour traiter les ex aequo. S'il n'y a aucun ex aequo (pour les durées) dans les données, alors toutes les méthodes sont équivalentes. De nombreux programmes de régression pour le modèle de Cox utilisent la méthode de Breslow par défaut mais pas `R`. Le logiciel `R` utilise par défaut la méthode d'Efron qui est plus précise et néanmoins moins gourmande en temps de calcul que la méthode de Breslow. La méthode exacte calcule la vraisemblance exacte partielle (ce qui revient à un modèle conditionnel logistique). S'il y a un trop grand nombre d'ex aequo dans les données, le temps de calcul sera excessif. Pourquoi a-t-on besoin d'approximer la vraisemblance de Cox? La raison est la suivante. Lorsqu'il n'y a pas d'ex aequo, lorsque le patient n°1 est le premier à mourir, lorsque le patient n°2 est le second à mourir, la vraisemblance partielle ressemble à :

$$\left(\frac{r_1}{\sum_i r_i}\right)\left(\frac{r_2}{\sum_{i>1} r_i}\right)\cdots$$

où r_i est le "risk score" pour l'individu i et est défini par :

$$r_i = \exp(\beta'.\mathbf{Z_i}).$$

Dans la réalité, les durées de vie sont vraisemblablement des variables continues (ce qui entraîne qu'il ne devrait pas exister d'ex aequo). Mais, pour des raisons pratiques, les observations sont discrétisées de sorte que les durées observées peuvent comporter des ex aequo. Par exemple, supposons que 5 patients soient visités une fois par jour (le soir) et les deux premiers soient morts à la fin de la même journée. Si les données avaient été plus précises, on saurait si les deux premiers termes de la vraisemblance sont :

$$L_{1,2} = \frac{r_1}{r_1 + r_2 + r_3 + r_4 + r_5}\frac{r_2}{r_2 + r_3 + r_4 + r_5}$$

ou bien

$$L_{1,2} = \frac{r_2}{r_1 + r_2 + r_3 + r_4 + r_5} \frac{r_1}{r_1 + r_3 + r_4 + r_5}.$$

Remarquons que le numérateur du produit est inchangé mais pas le dénominateur. L'approximation de Breslow est la suivante :

$$L_{1,2} \simeq \frac{r_1 r_2}{\left(\sum_{i=1}^{5} r_i\right)^2}.$$

Cela sous-estime la vraisemblance mais simplifie notablement le problème.

L'approximation d'Efron est la suivante :

$$L_{1,2} \simeq \frac{r_1 r_2}{(r_1 + r_2 + r_3 + r_4 + r_5)(\frac{r_1}{2} + \frac{r_2}{2} + r_3 + r_4 + r_5)}$$

puisque r_1 et r_2 ont chacun 50% de chances d'être la bonne valeur.

La méthode de la vraisemblance partielle exacte fournit la valeur :

$$L_{1,2} = \frac{r_1 r_2}{\sum_{i \neq j} r_i r_j}.$$

Voyons maintenant quelles instructions permettent d'ajuster un modèle de Cox sur des données gràce au logiciel `R`. Dans l'exemple suivant, un modèle de Cox est ajusté sur l'objet de survie `mydata.surv` avec en prédicteurs les covariables `cov1` et `cov2` : les effets simples dûs respectivement à `cov1` et `cov2` sont estimés.

```
cox.mydata = coxph(mydata.surv~cov1+cov2, data=mydata)
```

Dans l'exemple suivant, un modèle de Cox est ajusté sur l'objet de survie `myobject.surv` avec en prédicteurs les covariables `cov1` et `cov2` : les effets simples dûs respectivement à `cov1` et `cov2` sont estimés ainsi que l'interaction entre `cov1` et `cov2`.

```
cox.mydata = coxph(mydata.surv~cov1*cov2, data=mydata)
```

Dans l'exemple suivant, un modèle de Cox est ajusté sur l'objet de survie `myobject.surv` avec en prédicteur la covariable `cov1` : l'effet simple dû à `cov1` est estimé mais l'effet dû à `cov2` n'est pas estimé (le modèle est seulement ajusté dans chaque strate au moyen du "baseline hazard" qui est ici propre à chaque strate).

```
cox.mydata = coxph(mydata.surv~cov1+strata(cov2), data=mydata)
```

En plus des estimateurs du vecteur de paramètres β, la fonction `coxph` fournit un test par covariable et un test global. Le test du χ^2 pour la $j^{\text{ème}}$ covariable consiste à tester l'hypothèse nulle $H_0 : \beta^{(j)} = 0$ contre $H_1 : \beta^{(j)} \neq 0$. Le test global du χ^2 consiste à tester l'hypothèse nulle $H_0 : \beta = (0, \ldots, 0)'$ contre $H_1 : \beta \neq (0, \ldots, 0)'$.

Calcul des résidus après ajustement d'un modèle de Cox

De nombreuses procédures de vérification des hypothèses du modèle de Cox sont basées sur des quantités que l'on appelle résidus. Leurs valeurs sont calculés pour chaque individu. Ces résidus ont la particularité d'avoir un comportement connu, du moins approximativement, lorsque les hypothèses du modèle sont remplies. Pour définir les résidus en question, introduisons quelques notations.

Supposons que l'on dispose d'un échantillon de taille n noté $(T_i, D_i, \mathbf{Z_i})_{i=1,\ldots,n}$ où T_i est la durée observée pour le $i^{ème}$ patient, D_i est l'indicatrice associée et $\mathbf{Z_i} = (Z_i^{(1)}, \ldots, Z_i^{(p)})'$ est le vecteur de covariables du $i^{ème}$ patient. L'estimateur de Breslow (ou Tsiatis ou Link) de la fonction de hasard cumulée dite "baseline" que l'on note, pour $t \geq 0$, $\Lambda_0(t) = \int_0^t \lambda_0(u)du$ est défini, pour $t \geq 0$, par :

$$\widehat{\Lambda}_0(t) = \sum_{i=1}^{n} \frac{I(T_i \leq t, D_i = 1)}{\sum_{j=1}^{n} I(T_j \geq T_i).\exp(\widehat{\beta}'.\mathbf{Z_j})}.$$

Introduisons $\bar{\mathbf{Z}}(\beta,t) = (\bar{Z}^{(1)}(\beta,t),\ldots,\bar{Z}^{(p)}(\beta,t))$ le vecteur moyenne pondérée à l'instant t des vecteurs de covariables des sujets à risque à l'instant t dont la $j^{\text{ème}}$ coordonnée est donnée par :

$$\bar{Z}^{(j)}(\beta,t) = \sum_{k=1}^{n} \frac{I(T_k \geq t)\exp(\beta'.\mathbf{Z_k})}{\sum_{l=1}^{n} I(T_l \geq T_k)\exp(\beta'.\mathbf{Z_l})}\, Z_k^{(j)}.$$

Les résidus de `Schoenfeld` calculés par le logiciel `R` se présentent sous la forme d'une matrice à p colonnes qui a autant de lignes qu'il y a d'observations non-censurées dans les données. Les lignes sont ordonnées par durées de vie croissantes. Les résidus de Schoenfeld mesurent la distance entre le vecteur covariable des sujets et la moyenne pondérée des vecteurs covariables des sujets à risque. Les résidus de Schoenfeld servent à évaluer la tendance au cours du temps et donc à tester l'hypothèse des hasards proportionnels (qui dit que le log-ratio ne doit pas dépendre du temps). Le résidu correspondant à la covariable j et à la $i^{\text{ème}}$ durée non-censurée est donné par :

$$\widehat{Scho}_i^{(j)} = Z_i^{(j)} - \bar{Z}^{(j)}(\widehat{\beta}, T_i).$$

Les résidus de Schoenfeld sont liés au score de la façon suivante :

$$\sum_i \widehat{Scho}_i^{(j)} = \frac{\partial \log L(\beta)}{\partial \beta_j}\Big|_{\widehat{\beta}} = 0.$$

Les résidus du `score` se présentent sous la forme d'une matrice $n \times p$ avec une ligne par individu et une colonne par covariable. Les composants de cette matrice s'expriment de la façon suivante :

$$\widehat{Scor}_i^{(j)} = D_i\,(Z_i^{(j)} - \bar{Z}^{(j)}(\widehat{\beta}, T_i)) - \sum_{k=1}^{n} I(T_k \leq T_i)(Z_i^{(j)} - \bar{Z}^{(j)}(\widehat{\beta}, T_k))\frac{\exp(\widehat{\beta}'.\mathbf{Z}_k)}{\sum_{l=1}^{n} I(T_l \geq T_k)\exp(\widehat{\beta}'.\mathbf{Z}_l)}.$$

Les résidus `martingale` se présentent sous la forme d'un vecteur $\widehat{\mathbf{M}} = (\widehat{M}_1,\ldots,\widehat{M}_n)'$ dont la $i^{\text{ème}}$ coordonnée est donnée par :

$$\widehat{M}_i = D_i - \widehat{\Lambda}_0(T_i).\exp(\widehat{\beta}'.\mathbf{Z_j}).$$

La quantité $\widehat{M}_i$ représente la différence entre la nature observée du $i^{\text{ème}}$ évènement et la nature que l'on attendait en théorie compte tenu de la durée de suivi, des covariables et du modèle ajusté. Les $\widehat{M}_i$ sont de moyenne nulle, compris entre $-\infty$ et 1, négatifs lorsque les durées sont inférieures à celles attendues en théorie, asymptotiquement non corrélés et présentent une forte tendance à l'asymétrie. Le tracé des $\widehat{M}_i$ en fonction des covariables incluses dans le modèle permet de détecter la non-linéarité i.e. une forme fonctionnelle mal spécifiée dans la partie paramétrique du modèle. Le tracé des $\widehat{M}_i$ en fonction des durées de vie ou des rangs des durées de vie permet de détecter une influence du temps.
NB : pour les variables catégorielles, la non-linéarité n'est pas un problème.

Le résidu `deviance` est un vecteur $\widehat{\mathbf{Dev}} = (\widehat{Dev}_1,\ldots,\widehat{Dev}_n)'$ dont la $i^{\text{ème}}$ coordonnée est donnée par :

$$\widehat{Dev}_i = \text{sign}(\widehat{M}_i).\sqrt{-2\widehat{M}_i - 2D_i\log(D_i - \widehat{M}_i)}.$$

Les $\widehat{Dev}_i$ ont une distribution un peu moins asymétrique que les $\widehat{M}_i$ mais ne sont pas de moyenne nulle. Ils sont peu utilisés en pratique.

Le vecteur indice `dfbeta` pour le patient i (noté $\Delta_i\beta = (\Delta_i\beta^{(1)},\ldots,\Delta_i\beta^{(p)})'$) approxime le changement opéré dans l'estimation du vecteur covariable par le retrait de l'observation n°i. Sa $j^{\text{ème}}$ coordonnée est :

$$\Delta_i\beta^{(j)} = \widehat{\beta}^{(j)} - \widehat{\beta}^{(j)}_{(-i)}.$$

Le vecteur indice `dfbetas` pour le patient i (noté $s\Delta_i\beta = (s\Delta_i\beta^{(1)},\ldots,s\Delta_i\beta^{(p)})'$) approxime ce même changement mais normalisé par l'écart-type de l'estimateur. Sa $j^{\text{ème}}$ coordonnée est :

$$s\Delta_i\beta^{(j)} = \frac{\widehat{\beta}^{(j)} - \widehat{\beta}^{(j)}_{(-i)}}{\widehat{\text{se}}(\widehat{\beta}^{(j)})}.$$

L'instruction `R` qui permet d'effectuer le calcul des différents résidus et indices est la suivante :

```
residuals(cox.mydata,
          type=c("martingale","deviance","score","schoenfeld","dfbeta","dfbetas","scaledsch"),
          )
```

NB : les résidus "scaled Schoenfeld" sont également déterminés par la fonction `cox.zph` (voir ci-dessous).

Vérification des hypothèses du modèle de Cox

Hypothèse des hasards proportionnels
Le rapport des risques instantanés pour deux sujets i et j (ou hazards ratio ou risque relatif des sujets de caractéristique $\mathbf{Z}_i$ par rapport aux sujets de caractéristique $\mathbf{Z}_j$) ne doit pas dépendre du temps :

$$\frac{\lambda(t/\mathbf{Z}_i)}{\lambda(t/\mathbf{Z}_j)} = \frac{\exp(\beta'.\mathbf{Z}_i)}{\exp(\beta'.\mathbf{Z}_j)}.$$

Le test de l'hypothèse des hasards proportionnels dans le modèle de Cox se fait au moyen de la fonction `cox.zph` du package `survival`. L'instruction de base est :

```
test.mydata<-cox.zph(cox.mydata,
                     transform=c("km","rank","identity"),
                     global=TRUE )
print(test.mydata)     affiche les résultats des tests
plot(test.mydata)      trace les résidus scaled Schoenfeld  en fonction du temps
```

En entrée, `cox.mydata` représente le résultat d'un ajustement du modèle de Cox sur des données réalisé avec `coxph`. L'option `transform` indique comment les temps de survie doivent être transformés avec d'effectuer le test. Pour des données censurées à droite, la valeur par défaut est `transform="km"`. L'option `global=TRUE` indique qu'un test du χ^2 global sera effectué en plus de tests du χ^2 variable par variable. Le test du χ^2 pour la $j^{\text{ème}}$ covariable consiste à tester l'hypothèse nulle $H_0 : \beta^{(j)}(t) = \beta^{(j)}$ contre $H_1 : \beta^{(j)}(t) \neq \beta^{(j)}$ tandis que le test global du χ^2 consiste à tester l'hypothèse nulle $H_0 : \beta(t) = \beta$ contre $H_1 : \beta(t) \neq \beta$ où $\beta(t) = (\beta^{(1)}(t), \ldots, \beta^{(p)}(t))'$ représente le vecteur des paramètres que l'on autorise à dépendre du temps t.

Hypothèse de log-linéarité
Dans le modèle de Cox, la relation entre le risque instantané et les covariables est log-linéaire i.e. $\log(\lambda(t/\mathbf{Z})$ est une fonction linéaire de $\mathbf{Z}$, à savoir :

$$\log(\lambda(t/\mathbf{Z}) = \log(\lambda_0(t)) + \beta^{(1)}Z^{(1)} + \cdots + \beta^{(p)}Z^{(p)}.$$

Le tracé des résidus `martingale` en fonction de variables explicatives incluses dans le modèle peut être utilisé pour indiquer si certaines variables ont besoin d'être transformées avant d'être incorporées dans le modèle. Pour cela, on ajoute une courbe lissée sur les points obtenus. La forme fonctionnelle est alors suggérée par la forme de la courbe lissée. Ainsi, une croissance lente de la courbe suggèrera une transformation logarithme ou racine. A l'inverse, une croissance rapide suggèrera une transformation puissance avec une puissance supérieure à 1.
Par ailleurs, le tracé des résidus `martingale` en fonction de variables explicatives non incluses dans le modèle peut être utilisé pour indiquer si certaines variables devraient être incluses dans le modèle, ce qui est le cas si une dépendance apparait.

Recherche de sujets influents (ou marginaux)
Cette recherche se limite, en l'état actuel des connaissances, à la représentation graphique des indices `dfbetas` en fonction de l'indice des sujets. Des points extrêmes ou isolés doivent faire examiner en détails les sujets correspondants à la recherche d'une erreur de cotation ou d'une configuration exceptionnelle des variables explivatives. Il pourra parfois être intéressant d'extraire provisoirement ces sujets de l'échantillon et de recalculer le modèle : une variabilité importante des estimations fera craindre une instabilité importante et jettera un doute sur les résultats.