

# Récupérer et exécuter le test OLS de Grambsch-Therneau

Marc Thévenin

2023-06-13

Comment récupérer la variante du test de Grambsch-Therneau implémentée au package **survival** avant son passage à la v3. Permet de s'assurer une reproductibilité avec les autres applications statistiques (Stata, Sas, Python) avec des durées discrètes.

Package	Fonctions
<b>survival</b>	<code>coxph - coxzph.old</code>

## Champ d'application

- Modèle de Cox (analyse des durées)
- Diagnostic sur l'hypothèse de proportionalité des risques

## Problématique

- Depuis le passage à la version 3 du package **survival** en 2020, le test OLS sur les résidus de Schoenfeld a été supprimé et remplacé par le test GLS. Le second est la version originelle du test proposé en 1994 par P.Grambsch et T.Therneau.
- En présence d'évènements simultanés (durée discrète), les résultats affichés par les deux versions peuvent fortement varier.
- Lorsqu'on utilise le modèle de Cox avec des durées discrètes, situation très courante dans les sciences sociales, je préconise l'utilisation de la version dite *simplifiée* (OLS).
  - Le modèle de Cox est une méthode à durée continue, les conditions de validité du test GLS ne sont établies avec des événements mesurés simultanément.
  - Contrainte de reproductibilité: **Stata**, **Sas**, **Python** (**lifelines**, **statsmodels**)

## Récupération et exécution du test OLS

- J'ai récupéré le script de la fonction dans les archives du CRAN. Elle a simplement été renommée `cox.zphold()`.
- Pour charger la fonction, il suffit d'appliquer `source()` qui joue le même rôle que `library()`.
  - Directement sur le dépôt github:
 

```
* source("https://raw.githubusercontent.com/mthevenin/analyse_duree/main/cox.zphold.R")
```
  - Si le script a été enregistré en local:
 

```
* source("local_path/cox.zphold.R")
```

## Exemple

---

### Listing 1 Récupération des données

---

```
library(readr)
trans <- read.csv("https://raw.githubusercontent.com/mthevenin/analyse_duree/master/bases/trans.csv")
```

---



---

### Listing 2 Estimation d'un modèle de Cox

---

```
library(survival)
coxfit = coxph(formula = Surv(stime, died) ~ year + age + surgery, data = trans)
```

---



---

### Listing 3 Récupération et exécution du test OLS

---

```
source("https://raw.githubusercontent.com/mthevenin/analyse_duree/main/cox.zphold/cox.zphold.R")
cox.zphold(coxfit)
```

---

	rho	chisq	p
year	0.159	1.96	0.1620
age	0.109	1.15	0.2845
surgery	0.251	3.96	0.0465
GLOBAL	NA	7.99	0.0462