

Introduction à l'analyse Biographique des avec R

Correction TPI: analyse sorties du premier emploi

Marc Thevenin

2023-11-28

Table of contents

1	Packages	2
2	La base et la construction des variables d'analyse	3
2.1	La base	3
2.2	Construction des variables d'analyse	5
3	Analyse non paramétrique: Kaplan Meier et tests du log-rank	6
3.1	Estimation	6
3.2	Comparaison de fonctions de survie	9
3.2.1	Test du log-rank (niveau de diplôme)	13
3.2.2	Comparaison des RMST (niveau de diplôme)	14
4	Analyse semi-paramétrique: modèle à risques proportionnel de Cox	16
4.1	Estimation	17
4.2	Test de l'hypothèse de risques proportionnels	18
4.2.1	Test sur les résidus de Schoenfeld	18
4.2.2	Modèle de Cox et interaction avec la durée	20
4.3	Introduction d'une variable dynamique	21
4.3.1	Retour sur l'estimation du modèle de Cox	22
4.3.2	Construction de la TVC	24
4.3.3	Estimation du modèle	25
5	Modèle (logistique) à temps discret	26
5.1	Transformation de la base	26
5.2	Parmamétrisation de la durée et Estimation du modèle	29
5.2.1	Fonction continue de la durée	29

5.2.2	Forme discrète de la durée	33
6	Risque concurrent	35
6.1	Incidence cumulée et test de gray	35
7	Modèle multinomial	39
8	Mise en forme de la base	40
8.1	On conserve les premières séquences = emploi	41
8.2	Age à la 1ere séquence d'inactivité: retraite et autres causes de sortie	42
8.3	On conserve une ligne par individu	46

```
options(scipen=999) # empêcher le format scientifique
options(show.signif.stars=FALSE)
```

1 Packages

Analyse standard

Le package **survival** permet de réaliser un grand nombre d'analyses des durées pour un événement unique (risques non concurrents et non récurrent): **Kaplan Meier**, **modèles semiparamétrique** (Cox), **modèle paramétriques**.....

Analyse à durée discrète/groupée

Pour les modèles à durée discrète/groupée, on utilisera la fonction **glm**, intégrée à R. La fonction **uncount** du package **tydir** permet d'allonger, si nécessaire, la base en format long. La fonction **survsplit** du package peut, sous certaines conditions, réaliser cette opération.

Amélioration des outputs

- Graphique: Pour l'esthétique, on peut également utiliser le package **survminer** (ex: fonction de survie de type **ggplot2** et présentation des risk-ratio après un modèle de Cox). Le package **survminer** permet aussi de faire des tests du log-rank qui comparent des fonctions de survie 2 à 2 à partir de variables à plus de deux modalités.
- Outputs:
 - **jtools**: pour les modèles estimés avec la fonction **glm** (modèles à temps discret) ou **coxph**: package **RecordLinkage**, et fonction **summ** du package **jtools**. Pas d'utilisation possible pour la fonction **multinom** (risque concurrent avec un modèle logistique multinomial).
 - **gtsummary**: fonction **tbl_regression**, très répandue. Permet de sortir un output pour le modèle multinomial, mais en format long,

Installation des packages

```
#install.packages("survival")
#install.packages("survminer")
#install.packages("survRM2")
#install.packages("rms")
#install.packages("tidyr")
#install.packages("gtools")
#install.packages("jtools")
#install.packages("RecordLinkage")
#install.packages("gtsummary")
#install.packages("cmprsk")
#install.packages("nnet")

library(survival)
library(survminer)
library(survRM2)
library(tidyr)
library(gtools)
library(jtools)
library(RecordLinkage)
library(gtsummary)
library(cmprsk)
library(nnet)
```

2 La base et la construction des variables d'analyse

2.1 La base

Base **tp_activite**: Analyse de la durée de la **première séquence d'emploi(s)** de femmes nées avant 1951.

```
act <- read.csv("D:/D/Marc/SMS/FORMATIONS/2022/Durée2/tp/bio et entourage/sortie emploi [t
```

Il s'agit de la base "prête à l'analyse". A l'origine les données sont dans une base en format *âges-séquences* qui a du être mise en forme. Un programme de mise en forme, avec étapes prudentes, est donné à la fin du document.

Variables

Variable	
ident	Identifiant de la personne
diplome	Niveau de diplôme <ul style="list-style-type: none"> • 1 = inférieur au bac • 2 = bac • 3 = supérieur au bac
gene	Génération <ul style="list-style-type: none"> • 1 = avant 1940 • 2 = 1940-1940 • 3 = 1945-1950
csp	Csp représentative <ul style="list-style-type: none"> • artisane ou agricultrice • cadre • employée • ouvrière • profession intermédiaire
enf	Avoir eu un enfant pendant la période d'observation <ul style="list-style-type: none"> • 0: non • 1: oui
typinact	type de sortie de l'emploi <ul style="list-style-type: none"> • 0: pas de sortie de l'emploi • inactivité-retour au foyer • Autres (chômage, maladie...)
aanenf	Age à la de naissance de l'enfant <ul style="list-style-type: none"> • 0: pas d'enfant
ageact	Age à la naissance Age au premier emploi

Variable	
ageinact	Age à la sortie de l'emploi <ul style="list-style-type: none"> • 0: pas de sortie de l'emploi
ageret	Age à la retraite <ul style="list-style-type: none"> • Age à la sortie • 0: pas à la retraite au moment de l'enquête
age_enq	Age au moment de l'enquête <ul style="list-style-type: none"> • Age à la retraite

2.2 Construction des variables d'analyse

Variable censure-évènement

```
act$d = ifelse(act$ageinact>0, 1, 0)
```

Variable de durée

Rappel sur la notion de censure: la durée d'observation est inférieure à la durée d'exposition au risque.

Ici, on doit aussi prendre en compte la situation inverse. Des femmes ont toujours occupé un emploi et sont sorties du marché du travail uniquement au moment de la retraite. On ne peut plus les considérer comme exposées au risque après la retraite sinon la durée d'observation est supérieure à la durée d'exposition. On peut confondre ces situations comme des censures à droite (0) si on suppose que l'âge à la prise de la retraite n'est pas informative. Quelque part on fait une hypothèse d'indépendance entre le passage à la retraite et les autres causes de sorties. Potentiellement discutable, mais le passage à la retraite va se situer a des âges plutôt élevé.

Principe:

- Le début de l'exposition au risque est *ageact*
- On doit calculer la fin de l'exposition: sortie de l'emploi hors retraite , à la retraite ou simplement sortie de l'observation avec une censure à droite (âge à l'enquête).
- On calcule la durée. Avec des données discrètes/groupées, je prends toute la longueur de la durée et non la simple différence entre 2 points: fin - début + 1. On évite d'observer des évènements avec $t = 0$ et on facilite certaines manipulations, en particulier celles relatives aux covariables non fixes issues d'autres bases biographiques.

```
act$fin = ifelse(act$d==1, act$ageinact, ifelse(act$ageret>0, act$ageret, act$age_enq))
act$dur = act$fin - act$ageact + 1
```

3 Analyse non paramétrique: Kaplan Meier et tests du log-rank

3.1 Estimation

Fonctions `survfit` `survdif`

```
km = survfit(Surv(dur,d)~1, data=act)
summary(km)
```

Call: `survfit(formula = Surv(dur, d) ~ 1, data = act)`

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
2	1437	31	0.978	0.00383	0.971	0.986
3	1406	70	0.930	0.00674	0.917	0.943
4	1336	68	0.882	0.00850	0.866	0.899
5	1266	80	0.827	0.00999	0.807	0.846
6	1186	57	0.787	0.01081	0.766	0.808
7	1129	68	0.740	0.01158	0.717	0.763
8	1060	69	0.691	0.01219	0.668	0.716
9	991	54	0.654	0.01256	0.630	0.679
10	937	47	0.621	0.01281	0.596	0.647
11	888	56	0.582	0.01303	0.557	0.608
12	832	28	0.562	0.01310	0.537	0.588
13	804	30	0.541	0.01316	0.516	0.568
14	773	31	0.519	0.01320	0.494	0.546
15	741	23	0.503	0.01321	0.478	0.530
16	717	24	0.487	0.01321	0.461	0.513
17	691	16	0.475	0.01320	0.450	0.502
18	673	7	0.470	0.01319	0.445	0.497
19	665	13	0.461	0.01318	0.436	0.488
20	647	11	0.453	0.01317	0.428	0.480
21	633	8	0.448	0.01315	0.422	0.474
22	623	9	0.441	0.01314	0.416	0.468
23	612	7	0.436	0.01313	0.411	0.463
24	600	9	0.429	0.01311	0.405	0.456
25	586	6	0.425	0.01310	0.400	0.452
26	577	4	0.422	0.01309	0.397	0.449

27	567	4	0.419	0.01308	0.394	0.446
28	554	7	0.414	0.01307	0.389	0.440
29	535	5	0.410	0.01306	0.385	0.436
30	514	9	0.403	0.01305	0.378	0.429
31	485	4	0.400	0.01305	0.375	0.426
32	453	3	0.397	0.01305	0.372	0.423
33	422	8	0.389	0.01307	0.365	0.416
34	380	3	0.386	0.01309	0.361	0.413
35	344	8	0.377	0.01316	0.352	0.404
36	313	5	0.371	0.01323	0.346	0.398
37	277	3	0.367	0.01328	0.342	0.394
38	240	4	0.361	0.01341	0.336	0.388
39	206	6	0.351	0.01369	0.325	0.378
40	176	6	0.339	0.01407	0.312	0.367
41	144	6	0.325	0.01461	0.297	0.354
42	112	6	0.307	0.01546	0.278	0.339
43	84	2	0.300	0.01593	0.270	0.333
44	59	3	0.285	0.01738	0.252	0.321
45	40	1	0.277	0.01835	0.244	0.316

```
km
```

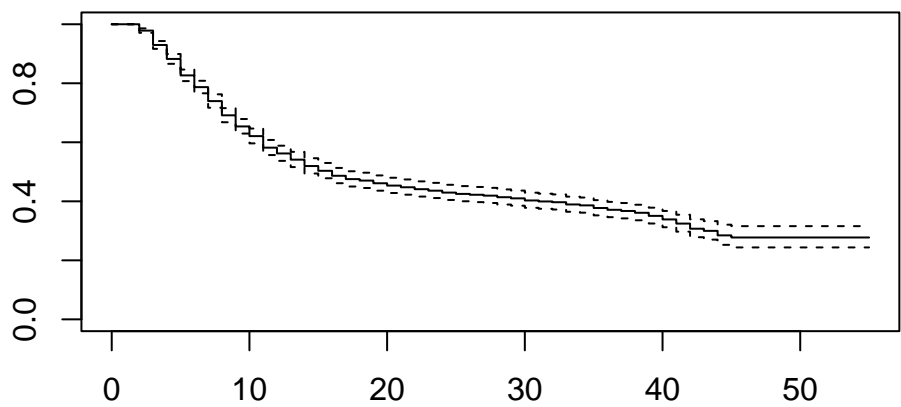
```
Call: survfit(formula = Surv(dur, d) ~ 1, data = act)
```

```

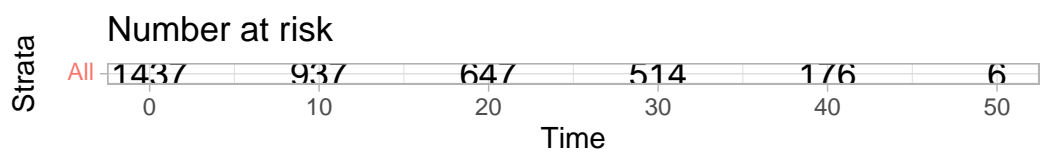
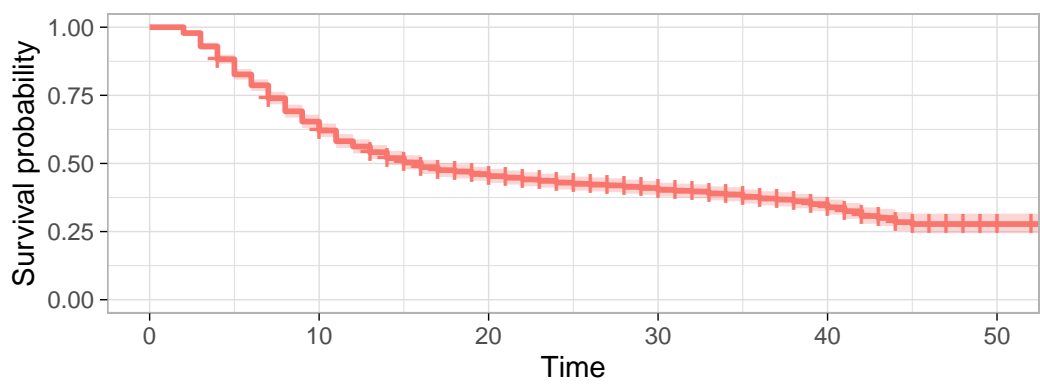
      n events median 0.95LCL 0.95UCL
[1,] 1437    919    16      14      18

```

```
plot(km)
ggsurvplot(km, risk.table=TRUE, ggtheme=theme_light(),)
```



Strata + All



3.2 Comparaison de fonctions de survie

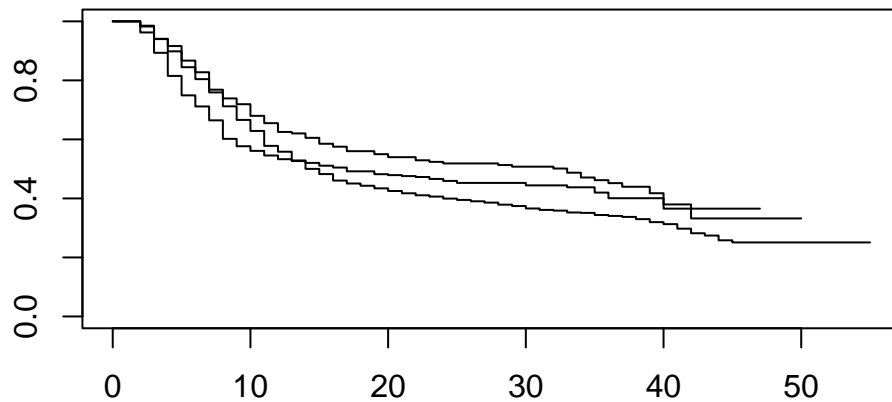
Niveau de diplome

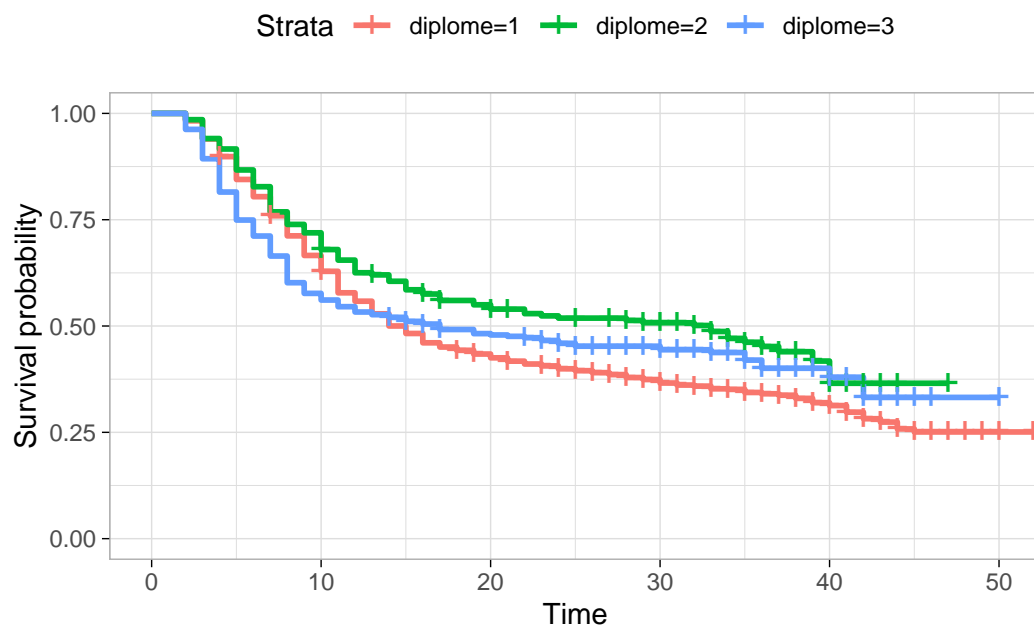
```
km = survfit(Surv(dur,d)~diplome, data=act)
# summary(km)
km
```

Call: `survfit(formula = Surv(dur, d) ~ diplome, data = act)`

	n	events	median	0.95LCL	0.95UCL
diplome=1	915	626	15	13	16
diplome=2	203	110	33	17	40
diplome=3	319	183	17	11	33

```
plot(km)
ggsurvplot(km, ggtheme=theme_light(),)
```





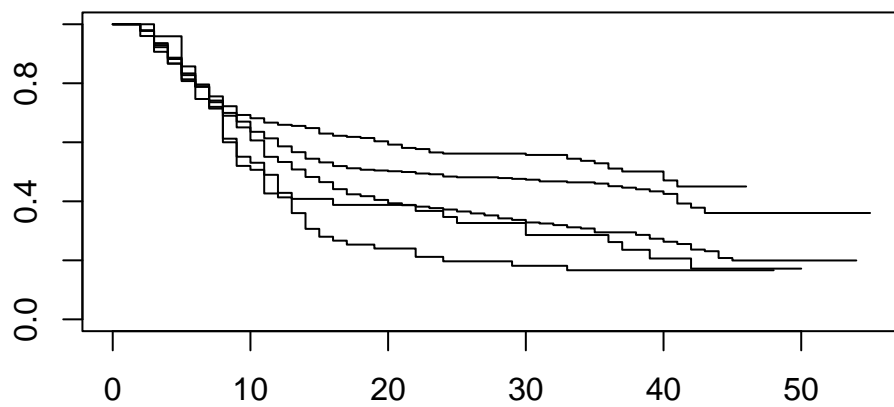
CSP

```
km = survfit(Surv(dur,d)~csp, data=act)
# summary(km)
km
```

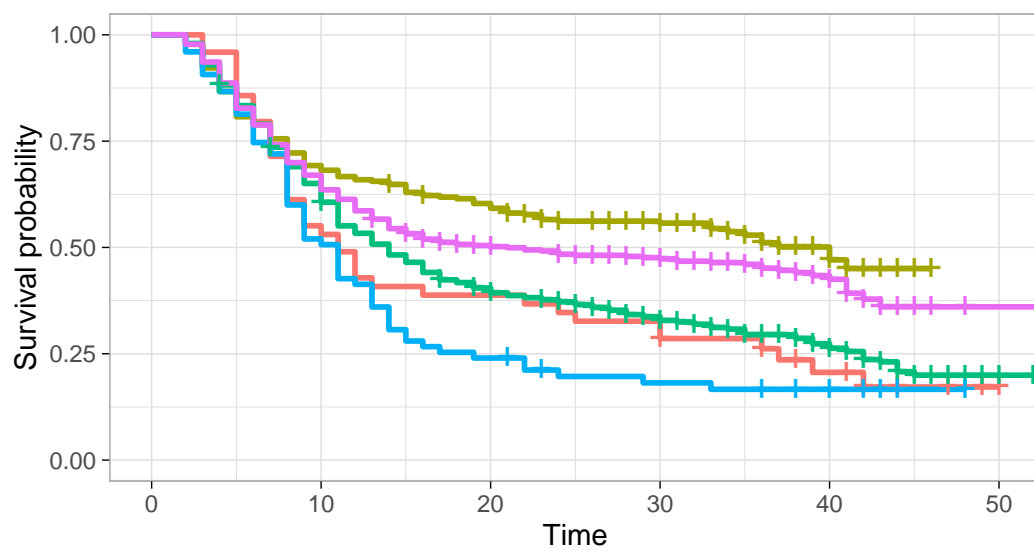
Call: `survfit(formula = Surv(dur, d) ~ csp, data = act)`

	n	events	median	0.95LCL	0.95UCL
csp=Artisans agr	49	39	11	8	25
csp=Cadres	270	129	40	33	NA
csp=Employes	637	461	14	12	16
csp=Ouvriers	75	62	11	8	13
csp=Profs interm	406	228	21	14	37

```
plot(km)
ggsurvplot(km, ggtheme=theme_light(),)
```



ata + csp=Artisans agr + csp=Cadres + csp=Employes + csp=Ouvriers + csp=



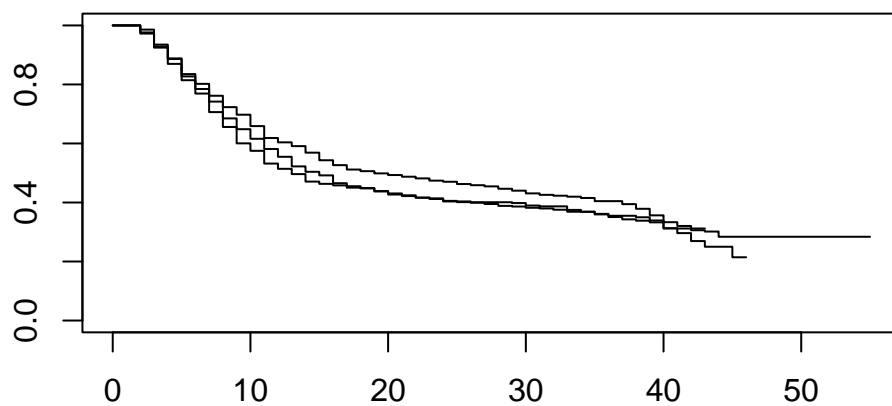
Générations

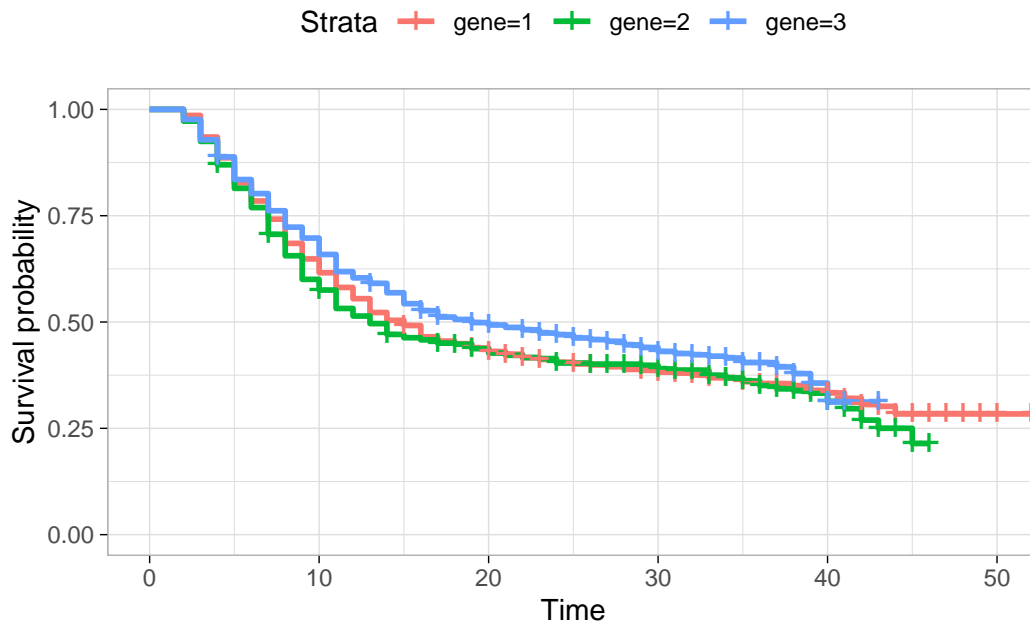
```
km = survfit(Surv(dur,d)~gene, data=act)
# summary(km)
km
```

Call: survfit(formula = Surv(dur, d) ~ gene, data = act)

	n	events	median	0.95LCL	0.95UCL
gene=1	492	335	15	13	18
gene=2	399	266	13	11	18
gene=3	546	318	19	16	27

```
plot(km)
ggsurvplot(km, ggtheme=theme_light(),)
```





3.2.1 Test du log-rank (niveau de diplome)

Attention: sensible à l'hypothèse de risques proportionnels (constance des risks ratios dans le temps => cf modèle de Cox) Hypothèse nulle : les fonctions de survie sont homogènes => par déduction rapports des risques toujours égaux à 1 / Hypothèse alternative: les fonctions de survie ne sont pas homogènes. La probabilité reportée (p-value) est appelée "risque de première espèce"

Test conseillé: utiliser l'option `rho=1`, pas sensible à la distribution des censures à droite, dit test de **Peto-Peto**.

```
survdif(Surv(dur,d)~diplome,data=act, rho=0)
```

Call:

```
survdif(formula = Surv(dur, d) ~ diplome, data = act, rho = 0)
```

	N	Observed	Expected	(O-E) ² /E	(O-E) ² /V
diplome=1	915	626	594	1.7463	5.196
diplome=2	203	110	138	5.6401	6.949
diplome=3	319	183	187	0.0994	0.131

Chisq= 7.9 on 2 degrees of freedom, p= 0.02

```
survdif(Surv(dur,d)~diplome,data=act, rho=1)
```

Call:

```
survdif(formula = Surv(dur, d) ~ diplome, data = act, rho = 1)
```

	N	Observed	Expected	(O-E) ² /E	(O-E) ² /V
diplome=1	915	420.8	411.3	0.223	0.88
diplome=2	203	75.7	95.3	4.027	6.68
diplome=3	319	141.6	131.5	0.763	1.34

Chisq= 7.1 on 2 degrees of freedom, p= 0.03

Si plus de deux modalités, il est pertinent de tester les fonctions de survie 2 à 2: fonction `pairwise_survdif` (même syntaxe que `surdif`)

```
pairwise_survdif(Surv(dur,d)~diplome,data=act, rho=1)
```

Pairwise comparisons using Peto & Peto test

data: act and diplome

1	2
2	0.031 -
3	0.516 0.031

P value adjustment method: BH

3.2.2 Comparaison des RMST (niveau de diplome)

RMST: Restricted mean of survival time.

- Intéressant pour des démographies, on compare des espérances de survie (séjour) partielles.
- La durée est bornée au moment du dernier évènement observé. Lorsqu'on compare plusieurs courbes, on prend celle où la durée du dernier évènement est la plus courte.
- Encore peu diffusé (hélas) en sciences sociales.
- Défaut de la fonction R: créer une variable nommée *arm* identique à la variable discrete, mais codée (0,1).

Package `survRM2` indépendant de `survival`.

Exemple pour comparer les 2 premiers niveaux de diplome (inférieur au bac versus bac).

```
rmst12=act[act$diplome!=3,]
rmst12$arm=ifelse(rmst12$diplome==1,1,0)
a=rmst2(rmst12$dur, rmst12$d, rmst12$arm)
print(a)
```

The truncation time, tau, was not specified. Thus, the default tau 47 is used.

Restricted Mean Survival Time (RMST) by arm

	Est.	se	lower .95	upper .95
RMST (arm=1)	23.041	0.582	21.901	24.181
RMST (arm=0)	27.250	1.309	24.683	29.816

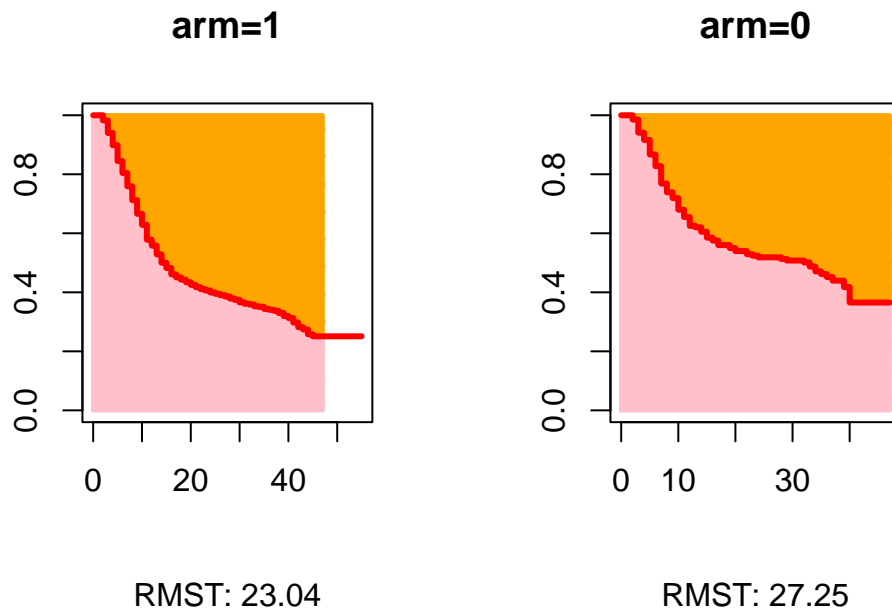
Restricted Mean Time Lost (RMTL) by arm

	Est.	se	lower .95	upper .95
RMTL (arm=1)	23.959	0.582	22.819	25.099
RMTL (arm=0)	19.750	1.309	17.184	22.317

Between-group contrast

	Est.	lower .95	upper .95	p
RMST (arm=1)-(arm=0)	-4.209	-7.017	-1.401	0.003
RMST (arm=1)/(arm=0)	0.846	0.760	0.940	0.002
RMTL (arm=1)/(arm=0)	1.213	1.056	1.393	0.006

```
plot(a)
```



PS: *c'est la comparaison qui renvoyait un msg d'erreur mercredi après-midi. La fatigue a surement jouée son rôle, j'avais laissé la valeur 3 dans `ifelse` alors qu'il n'y avait plus d'observation pour cette modalité.*

4 Analyse semi-paramétrique: modèle à risques proportionnel de Cox

Rappel: l'hypothèse PH signifie que le rapport des risques est constant pendant la durée d'observation.

Avec une seule covariable X , un modèle à risque proportionnel s'écrit:

$$h(t_i|X) = h_0(t_i) \times e^{b \times X}$$

Soit 2 observations A et B. Le rapport des risques entre A et B s'écrit:

$$\frac{h(t_i|X_A)}{h(t_i|X_B)} = \frac{e^{b \times X_A}}{e^{b \times X_B}} = e^{b \times (X_A - X_B)}$$

Fonctions `coxph` `coxzph` `survsplit`

4.1 Estimation

Penser à mettre les covariables en facteurs si nécessaire (ou `factor(nom variable)` dans le modèle) et prévoir les changements de références (fonction `relevel`).

```
act$gene = as.factor(act$gene)
act$gene = relevel(act$gene, ref = "2")
act$csp = as.factor(act$csp)
act$csp = relevel(act$csp, ref = "Cadres")
act$diplome = as.factor(act$diplome)
act$diplome = relevel(act$diplome, ref = 2)

coxfit = coxph(Surv(dur,d) ~ gene + csp + diplome, data=act)
summary(coxfit)
```

Call:

```
coxph(formula = Surv(dur, d) ~ gene + csp + diplome, data = act)
```

n= 1437, number of events= 919

	coef	exp(coef)	se(coef)	z	Pr(> z)
gene1	-0.13753	0.87151	0.08337	-1.650	0.09904
gene3	-0.19451	0.82324	0.08356	-2.328	0.01992
cspArtisans agr	0.79870	2.22264	0.18885	4.229	0.000023455201
cspEmployes	0.72507	2.06488	0.11937	6.074	0.0000000001248
cspOuvriers	1.05506	2.87214	0.17004	6.205	0.0000000000548
cspProfs interm	0.35252	1.42265	0.11755	2.999	0.00271
diplome1	0.04667	1.04778	0.11013	0.424	0.67171
diplome3	0.37376	1.45319	0.12612	2.964	0.00304

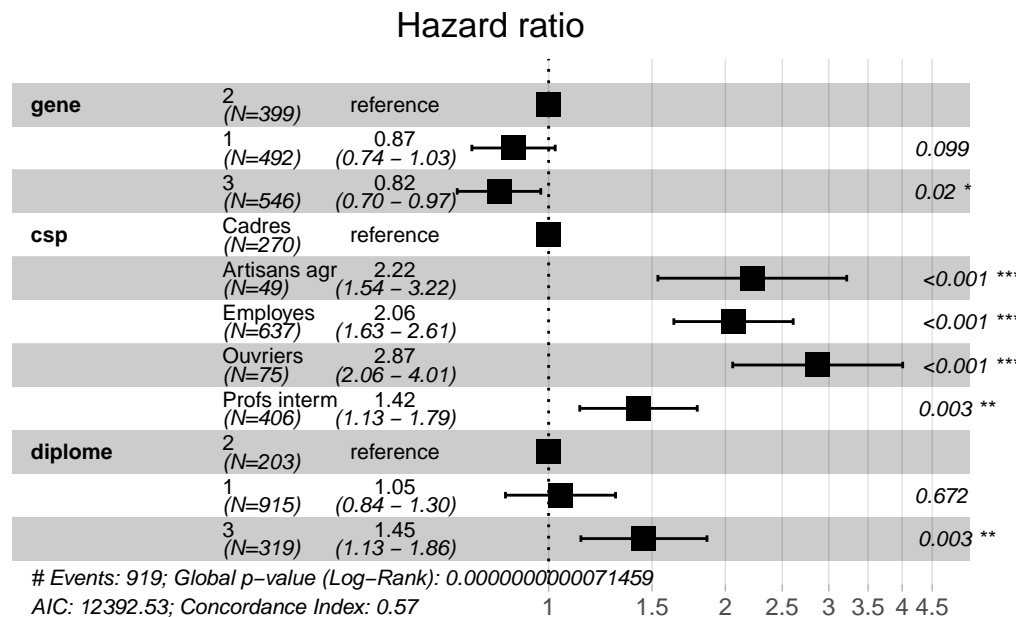
	exp(coef)	exp(-coef)	lower .95	upper .95
gene1	0.8715	1.1474	0.7401	1.0262
gene3	0.8232	1.2147	0.6989	0.9697
cspArtisans agr	2.2226	0.4499	1.5350	3.2183
cspEmployes	2.0649	0.4843	1.6341	2.6092
cspOuvriers	2.8721	0.3482	2.0581	4.0082
cspProfs interm	1.4227	0.7029	1.1299	1.7913
diplome1	1.0478	0.9544	0.8444	1.3002
diplome3	1.4532	0.6881	1.1349	1.8607

Concordance= 0.574 (se = 0.01)

Likelihood ratio test= 69.18 on 8 df, p=0.000000000007
Wald test = 66.89 on 8 df, p=0.000000000002
Score (logrank) test = 68.01 on 8 df, p=0.000000000001

```
ggforest(coxfit)
```

Warning in .get_data(model, data = data): The `data` argument is not provided.
Data will be extracted from model fit.



4.2 Test de l'hypothèse de risques proportionnels

4.2.1 Test sur les résidus de Schoenfeld

v3 - test exact (gls): La fonction utilisée est `cox.zph` **v2 - test simplifié (ols):** la fonction utilisée est `cox.zphold` (à récupérer et à charger). Je continue de conseiller à utiliser cette version du test.

Le test peut utiliser plusieurs formes paramétriques de la durée. Par défaut la fonction utilise $f(t) = 1 - S(t)$, soit le complémentaire de l'estimateur de Kaplan-Meier (option `transform="km"`). Autres fonctions $f(t) = t$ (`transform="identity"`).

Avec la v2:

```
source("D:/D/Marc/SMS/FORMATIONS/2022/Durée2/a distribuer/cox.zphold.R")
```

```
cox.zphold(coxfit)
```

	rho	chisq	p
gene1	-0.0112	0.1161	0.73328057
gene3	0.0307	0.8624	0.35305484
cspArtisans agr	0.0481	2.1434	0.14318292
cspEmployes	0.0458	2.0459	0.15261533
cspOuvriers	0.0294	0.8157	0.36643658
cspProfs interm	-0.0151	0.2196	0.63935110
diplome1	0.0036	0.0121	0.91247404
diplome3	-0.0958	8.8535	0.00292533
GLOBAL	NA	42.3314	0.00000117

```
cox.zphold(coxfit, transform="identity")
```

	rho	chisq	p
gene1	-0.0309	0.889	0.34586
gene3	0.0275	0.691	0.40578
cspArtisans agr	0.0339	1.068	0.30143
cspEmployes	0.0291	0.826	0.36339
cspOuvriers	-0.0066	0.041	0.83949
cspProfs interm	-0.0357	1.224	0.26857
diplome1	-0.0162	0.245	0.62091
diplome3	-0.0779	5.861	0.01548
GLOBAL	NA	24.836	0.00166

Avec la v3:

```
cox.zph(coxfit, terms=FALSE)
```

	chisq	df	p
gene1	0.00478	1	0.945
gene3	0.60359	1	0.437
cspArtisans agr	1.10591	1	0.293
cspEmployes	15.45130	1	0.0000846587

cspOuvriers	0.89068	1	0.345
cspProfs interm	8.30586	1	0.004
diplome1	23.54565	1	0.0000012198
diplome3	32.99485	1	0.0000000092
GLOBAL	43.35937	8	0.0000007516

```
cox.zph(coxfit, terms=FALSE, transform="identity")
```

	chisq	df	p
gene1	1.116	1	0.29083
gene3	1.947	1	0.16294
cspArtisans agr	0.859	1	0.35392
cspEmployes	10.786	1	0.00102
cspOuvriers	0.138	1	0.70987
cspProfs interm	7.114	1	0.00765
diplome1	8.941	1	0.00279
diplome3	16.348	1	0.000053
GLOBAL	26.963	8	0.00072

4.2.2 Modèle de Cox et interaction avec la durée

- Prérequis: les variables discrètes doivent être sous forme d'indicateurs (0,1).
- Permet de modifier le modèle, de le présenter sans l'hypothèse de constance du rapport de risque et d'enrichir l'interprétation.
- Attention à l'interprétation du terme d'interaction, ce n'est pas un rapport de risque mais un rapport de rapports de risque.

On peut directement introduire cette interaction, avec le choix de la paramétrisation de la durée, dans la fonction `coxph`.

```
act$dipl1 = ifelse(act$diplome==1, 1,0)
# act$dipl2 = ifelse(act$diplome==2, 1,0) Pas nécessaire car c'est la référence
act$dipl3 = ifelse(act$diplome==3, 1,0)

coxfit2 = coxph(Surv(dur, d) ~ gene + csp + dipl1 + dipl3 + tt(dipl3), data = act, tt = function(t) {
summary(coxfit2)
```

Call:

```
coxph(formula = Surv(dur, d) ~ gene + csp + dipl1 + dipl3 + tt(dipl3),
      data = act, tt = function(x, t, ...) x * t)
```

n= 1437, number of events= 919

	coef	exp(coef)	se(coef)	z	Pr(> z)
gene1	-0.12615	0.88148	0.08345	-1.512	0.13060
gene3	-0.19547	0.82245	0.08356	-2.339	0.01932
cspArtisans agr	0.79385	2.21189	0.18867	4.208	0.000025819776
cspEmployes	0.71162	2.03729	0.11872	5.994	0.000000002046
cspOuvriers	1.05756	2.87934	0.16977	6.229	0.000000000468
cspProfs interm	0.33875	1.40320	0.11734	2.887	0.00389
dipl1	0.04848	1.04968	0.11011	0.440	0.65973
dipl3	0.88508	2.42317	0.17238	5.134	0.000000282986
tt(dipl3)	-0.05013	0.95111	0.01259	-3.980	0.000068799501

	exp(coef)	exp(-coef)	lower .95	upper .95
gene1	0.8815	1.1345	0.7485	1.0381
gene3	0.8225	1.2159	0.6982	0.9688
cspArtisans agr	2.2119	0.4521	1.5281	3.2016
cspEmployes	2.0373	0.4908	1.6144	2.5710
cspOuvriers	2.8793	0.3473	2.0643	4.0161
cspProfs interm	1.4032	0.7127	1.1149	1.7660
dipl1	1.0497	0.9527	0.8459	1.3025
dipl3	2.4232	0.4127	1.7284	3.3972
tt(dipl3)	0.9511	1.0514	0.9279	0.9749

Concordance= 0.59 (se = 0.01)

Likelihood ratio test= 88.29 on 9 df, p=0.000000000000004

Wald test = 83.36 on 9 df, p=0.000000000000003

Score (logrank) test = 84.6 on 9 df, p=0.000000000000002

4.3 Introduction d'une variable dynamique

On regardera le cas d'une variable de type binaire.

Question: quel est l'effet de la naissance d'un (premier) enfant sur le risque de sortie de l'emploi?

Estimation du modèle en considérant la naissance comme une variable fixe (variable enf)

```
coxfit3 = coxph(Surv(dur, d) ~ gene + csp + diplome + enf, data = act)
summary(coxfit3)
```

Call:

```
coxph(formula = Surv(dur, d) ~ gene + csp + diplome + enf, data = act)
```

n= 1437, number of events= 919

	coef	exp(coef)	se(coef)	z	Pr(> z)
gene1	-0.11143	0.89455	0.08354	-1.334	0.182228
gene3	-0.20483	0.81479	0.08359	-2.450	0.014273
cspArtisans agr	0.68793	1.98959	0.18911	3.638	0.000275
cspEmployes	0.67699	1.96794	0.12006	5.639	0.0000000171327
cspOuvriers	0.96649	2.62871	0.17056	5.667	0.0000000145567
cspProfs interm	0.32627	1.38579	0.11792	2.767	0.005660
diplome1	0.02963	1.03007	0.11052	0.268	0.788620
diplome3	0.39445	1.48357	0.12655	3.117	0.001827
enf	0.81318	2.25506	0.12314	6.604	0.0000000000401

	exp(coef)	exp(-coef)	lower .95	upper .95
gene1	0.8945	1.1179	0.7594	1.0537
gene3	0.8148	1.2273	0.6917	0.9598
cspArtisans agr	1.9896	0.5026	1.3734	2.8823
cspEmployes	1.9679	0.5081	1.5553	2.4900
cspOuvriers	2.6287	0.3804	1.8818	3.6722
cspProfs interm	1.3858	0.7216	1.0998	1.7461
diplome1	1.0301	0.9708	0.8295	1.2792
diplome3	1.4836	0.6740	1.1577	1.9012
enf	2.2551	0.4434	1.7715	2.8706

Concordance= 0.605 (se = 0.01)

Likelihood ratio test= 123.5 on 9 df, p=<0.0000000000000002

Wald test = 107.3 on 9 df, p=<0.0000000000000002

Score (logrank) test = 111.3 on 9 df, p=<0.0000000000000002

4.3.1 Retour sur l'estimation du modèle de Cox

La base qui sert à l'estimation est splittée aux temps d'évènement.

Pour transformer la base d'origine on utilise la fonction `survsplit`. Au préalable on doit créer un vecteur donnant tous les temps d'évènement observés (fonction `cut`).

Récupération des durées d'évènement

```
cut= unique(act$dur[act$d == 1])
```

Transformation de la base

La nouvelle base est nommée *tv*. Il est nécessaire d'avoir une variable qui indique le début de l'intervalle de temps entre deux évènements (ici on va l'appeler *dur0*).

```
tv = survSplit(data = act, cut = cut, end = "dur", start = "dur0", event = "d")
```

Estimation du modèle

Avec R, lorsque la base est splittée, on ne renseigne pas la durée mais le début et la fin de chaque intervalle pour utiliser la fonction `coxph`.

On vérifie que le modèle estimé avec cette base splittée est identique au précédent.

```
coxfit = coxph(Surv(dur0, dur, d) ~ gene + csp + diplome + enf, data = tv)
summary(coxfit)
```

Call:

```
coxph(formula = Surv(dur, d) ~ gene + csp + diplome + enf, data = act)
```

n= 1437, number of events= 919

	coef	exp(coef)	se(coef)	z	Pr(> z)
gene1	-0.11143	0.89455	0.08354	-1.334	0.182228
gene3	-0.20483	0.81479	0.08359	-2.450	0.014273
cspArtisans agr	0.68793	1.98959	0.18911	3.638	0.000275
cspEmployes	0.67699	1.96794	0.12006	5.639	0.0000000171327
cspOuvriers	0.96649	2.62871	0.17056	5.667	0.0000000145567
cspProfs interm	0.32627	1.38579	0.11792	2.767	0.005660
diplome1	0.02963	1.03007	0.11052	0.268	0.788620
diplome3	0.39445	1.48357	0.12655	3.117	0.001827
enf	0.81318	2.25506	0.12314	6.604	0.0000000000401

	exp(coef)	exp(-coef)	lower .95	upper .95
gene1	0.8945	1.1179	0.7594	1.0537
gene3	0.8148	1.2273	0.6917	0.9598
cspArtisans agr	1.9896	0.5026	1.3734	2.8823
cspEmployes	1.9679	0.5081	1.5553	2.4900
cspOuvriers	2.6287	0.3804	1.8818	3.6722
cspProfs interm	1.3858	0.7216	1.0998	1.7461
diplome1	1.0301	0.9708	0.8295	1.2792
diplome3	1.4836	0.6740	1.1577	1.9012
enf	2.2551	0.4434	1.7715	2.8706

```

Concordance= 0.605 (se = 0.01 )
Likelihood ratio test= 123.5 on 9 df, p=<0.0000000000000002
Wald test = 107.3 on 9 df, p=<0.0000000000000002
Score (logrank) test = 111.3 on 9 df, p=<0.0000000000000002

```

4.3.2 Construction de la TVC

Comme on dispose de l'information sur l'âge à la naissance de l'enfant:

Si *enfant*==1 & *age*>=*aanenf*, *tvc*=1, 0 sinon.

L'âge de la répondante sur chaque année d'observation n'existe pas dans la base puisqu'on a une variable de durée. On doit construire cette variable pour comparer l'âge de la personne sur toute la période d'observation à l'âge à la naissance du premier enfant.

```
tvc$age= tvc$ageact + tvc$dur0
```

```

tvc$tvc = tvc$enf
tvc$tvc = ifelse(tvc$tvc==1 & tvc$age>=tvc$aanenf,1,0)
head(tvc, n=12)

```

	ident	diplome	gene	csp	enf	typinact	aanenf	ageact	ageinact	ageret
1	101	1	1	Ouvriers	1	2	26	14	26	0
2	101	1	1	Ouvriers	1	2	26	14	26	0
3	101	1	1	Ouvriers	1	2	26	14	26	0
4	101	1	1	Ouvriers	1	2	26	14	26	0
5	101	1	1	Ouvriers	1	2	26	14	26	0
6	101	1	1	Ouvriers	1	2	26	14	26	0
7	101	1	1	Ouvriers	1	2	26	14	26	0
8	101	1	1	Ouvriers	1	2	26	14	26	0
9	101	1	1	Ouvriers	1	2	26	14	26	0
10	101	1	1	Ouvriers	1	2	26	14	26	0
11	101	1	1	Ouvriers	1	2	26	14	26	0
12	101	1	1	Ouvriers	1	2	26	14	26	0

	age_enq	fin	dipl1	dipl3	dur0	dur	d	age	tvc
1	61	26	1	0	0	2	0	14	0
2	61	26	1	0	2	3	0	16	0
3	61	26	1	0	3	4	0	17	0
4	61	26	1	0	4	5	0	18	0
5	61	26	1	0	5	6	0	19	0
6	61	26	1	0	6	7	0	20	0
7	61	26	1	0	7	8	0	21	0
8	61	26	1	0	8	9	0	22	0

9	61	26	1	0	9	10	0	23	0
10	61	26	1	0	10	11	0	24	0
11	61	26	1	0	11	12	0	25	0
12	61	26	1	0	12	13	1	26	1

Remarque: important

La naissance d'un enfant peut avoir lieu après la sortie d'activité. Cela pose un problème car avec les modèles de durée on stoppe l'observation après l'évènement. Par principe la cause précède toujours l'effet, mais dans ce cas on observe la cause après. Il y a un risque que l'estimation soit biaisée.

Lorsque que la cause est observée avant l'évènement, on parle d'effet d'adaptation, lorsque la manifestation de la cause est observée après l'évènement on parle d'effet d'anticipation. Il n'en reste pas moins que la cause réelle est toujours antérieure à l'évènement, mais elle n'est pas observée. Certains modèles tente de résoudre ce problème, mais ils sont particulièrement complexes (et peu diffusés).

On n'a pas ce problème avec l'impact des greffes (analyse transplantation), car l'évènement étudié est complètement absorbant (décès).

4.3.3 Estimation du modèle

```
coxfit3 = coxph(Surv(dur0, dur, d) ~ gene + csp + diplome + tv, data = tv)
summary(coxfit3)
```

Call:

```
coxph(formula = Surv(dur0, dur, d) ~ gene + csp + diplome + tv,
      data = tv)
```

n= 27742, number of events= 919

	coef	exp(coef)	se(coef)	z	Pr(> z)
gene1	-0.05154	0.94977	0.08364	-0.616	0.53777
gene3	-0.21045	0.81022	0.08354	-2.519	0.01176
cspArtisans agr	0.79742	2.21980	0.18920	4.215	0.0000250048
cspEmployes	0.69243	1.99857	0.12107	5.719	0.0000000107
cspOuvriers	1.01370	2.75579	0.17075	5.937	0.0000000029
cspProfs interm	0.32316	1.38149	0.11890	2.718	0.00657
diplome1	0.09618	1.10095	0.11041	0.871	0.38370
diplome3	0.38124	1.46410	0.12728	2.995	0.00274
tv	1.13565	3.11318	0.08035	14.134	< 0.0000000000000002

exp(coef) exp(-coef) lower .95 upper .95

gene1	0.9498	1.0529	0.8062	1.1190
gene3	0.8102	1.2342	0.6879	0.9543
cspArtisans agr	2.2198	0.4505	1.5320	3.2163
cspEmployes	1.9986	0.5004	1.5764	2.5338
cspOuvriers	2.7558	0.3629	1.9720	3.8511
cspProfs interm	1.3815	0.7239	1.0943	1.7440
diplome1	1.1010	0.9083	0.8867	1.3669
diplome3	1.4641	0.6830	1.1409	1.8789
tv	3.1132	0.3212	2.6596	3.6442

Concordance= 0.664 (se = 0.01)

Likelihood ratio test= 284.7 on 9 df, p=<0.0000000000000002

Wald test = 264.9 on 9 df, p=<0.0000000000000002

Score (logrank) test = 275.8 on 9 df, p=<0.0000000000000002

5 Modèle (logistique) à temps discret

- Par définition ce n'est pas un modèle à risque proportionnel, mais à odds proportionnels. Toutefois en situation de rareté, l'Odds converge vers une probabilité, qui est une mesure du risque (ici une probabilité conditionnelle).
- Le modèle à temps discret est de type paramétrique.
- Il est moins contraignant que le modèle de Cox si l'hypothèse de proportionnalité n'est pas respectée, car le modèle est ajusté par une fonction de la durée.
- La base de données doit être transformée ici en format long (cf `survsplit`): aux temps d'observation ou sur des intervalles de temps.

Avec un lien logistique, le modèle à temps discret, avec seulement des covariables fixes, peut s'écrire:

$$\log \left[\frac{P(Y_t = 1 \mid Y_{t-1} = 0, X_k)}{1 - P(Y_t = 1 \mid Y_{t-1} = 0, X_k)} \right] = a_0 + \sum_p a_p f(t_p) + \sum_k b_k X_k$$

5.1 Transformation de la base

Allongement de la base et variables d'analyse

On va utiliser la fonction `uncount` du package `tidyr` qui va répliquer les observations selon la valeur de la variable de durée. Avant, on génère une variable miroir de la variable *mois* qui sera supprimée avec l'exécution d'`uncount`, et une variable `x=1` pour créer la variable de durée sous forme de compteur.

```
act$dur2 = act$dur
act$x=1
td = uncount(data=act,dur)
head(td, n=13)
```

	ident	diplome	gene	csp	enf	typinact	aanenf	ageact	ageinact	ageret
1	101	1	1 Ouvriers	1		2	26	14	26	0
2	101	1	1 Ouvriers	1		2	26	14	26	0
3	101	1	1 Ouvriers	1		2	26	14	26	0
4	101	1	1 Ouvriers	1		2	26	14	26	0
5	101	1	1 Ouvriers	1		2	26	14	26	0
6	101	1	1 Ouvriers	1		2	26	14	26	0
7	101	1	1 Ouvriers	1		2	26	14	26	0
8	101	1	1 Ouvriers	1		2	26	14	26	0
9	101	1	1 Ouvriers	1		2	26	14	26	0
10	101	1	1 Ouvriers	1		2	26	14	26	0
11	101	1	1 Ouvriers	1		2	26	14	26	0
12	101	1	1 Ouvriers	1		2	26	14	26	0
13	101	1	1 Ouvriers	1		2	26	14	26	0

	age_enq	d	fin	dipl1	dipl3	dur2	x
1	61	1	26	1	0	13	1
2	61	1	26	1	0	13	1
3	61	1	26	1	0	13	1
4	61	1	26	1	0	13	1
5	61	1	26	1	0	13	1
6	61	1	26	1	0	13	1
7	61	1	26	1	0	13	1
8	61	1	26	1	0	13	1
9	61	1	26	1	0	13	1
10	61	1	26	1	0	13	1
11	61	1	26	1	0	13	1
12	61	1	26	1	0	13	1
13	61	1	26	1	0	13	1

Variable de durée (compteur)

Remarque: variante possible avec dplyr, je fais encore du R à l'ancienne

```
td$t = ave(td$x,td$ident, FUN=cumsum)
head(td, n=13)
```

	ident	diplome	gene	csp	enf	typinact	aanenf	ageact	ageinact	ageret
1	101	1	1 Ouvriers	1		2	26	14	26	0
2	101	1	1 Ouvriers	1		2	26	14	26	0
3	101	1	1 Ouvriers	1		2	26	14	26	0
4	101	1	1 Ouvriers	1		2	26	14	26	0
5	101	1	1 Ouvriers	1		2	26	14	26	0
6	101	1	1 Ouvriers	1		2	26	14	26	0
7	101	1	1 Ouvriers	1		2	26	14	26	0
8	101	1	1 Ouvriers	1		2	26	14	26	0
9	101	1	1 Ouvriers	1		2	26	14	26	0
10	101	1	1 Ouvriers	1		2	26	14	26	0
11	101	1	1 Ouvriers	1		2	26	14	26	0
12	101	1	1 Ouvriers	1		2	26	14	26	0
13	101	1	1 Ouvriers	1		2	26	14	26	0

	age_enq	d	fin	dipl1	dipl3	dur2	x	t
1	61	1	26	1	0	13	1	1
2	61	1	26	1	0	13	1	2
3	61	1	26	1	0	13	1	3
4	61	1	26	1	0	13	1	4
5	61	1	26	1	0	13	1	5
6	61	1	26	1	0	13	1	6
7	61	1	26	1	0	13	1	7
8	61	1	26	1	0	13	1	8
9	61	1	26	1	0	13	1	9
10	61	1	26	1	0	13	1	10
11	61	1	26	1	0	13	1	11
12	61	1	26	1	0	13	1	12
13	61	1	26	1	0	13	1	13

Variable évènement-censure

On remplace les valeurs de la variable évènement/censure d de sorte à ce que la sortie de l'emploi soit codée 0 avant la sortie effective .

```
td$d[td$t<td$dur]=0
head(td, n=13)
```

	ident	diplome	gene	csp	enf	typinact	aanenf	ageact	ageinact	ageret
1	101	1	1 Ouvriers	1		2	26	14	26	0
2	101	1	1 Ouvriers	1		2	26	14	26	0
3	101	1	1 Ouvriers	1		2	26	14	26	0
4	101	1	1 Ouvriers	1		2	26	14	26	0

5	101	1	1	Ouvriers	1	2	26	14	26	0
6	101	1	1	Ouvriers	1	2	26	14	26	0
7	101	1	1	Ouvriers	1	2	26	14	26	0
8	101	1	1	Ouvriers	1	2	26	14	26	0
9	101	1	1	Ouvriers	1	2	26	14	26	0
10	101	1	1	Ouvriers	1	2	26	14	26	0
11	101	1	1	Ouvriers	1	2	26	14	26	0
12	101	1	1	Ouvriers	1	2	26	14	26	0
13	101	1	1	Ouvriers	1	2	26	14	26	0

	age	enq	d	fin	dipl1	dipl3	dur2	x	t
1	61	0	26		1	0	13	1	1
2	61	0	26		1	0	13	1	2
3	61	0	26		1	0	13	1	3
4	61	0	26		1	0	13	1	4
5	61	0	26		1	0	13	1	5
6	61	0	26		1	0	13	1	6
7	61	0	26		1	0	13	1	7
8	61	0	26		1	0	13	1	8
9	61	0	26		1	0	13	1	9
10	61	0	26		1	0	13	1	10
11	61	0	26		1	0	13	1	11
12	61	0	26		1	0	13	1	12
13	61	1	26		1	0	13	1	13

5.2 Parmamétrisation de la durée et Estimation du modèle

5.2.1 Fonction continue de la durée

- $a_0 + \sum_p a_p f(t_p)$ sera la baseline du risque.
- Il faut trouver une fonction qui ajuste le mieux les données. Classiquement on utilise des polynômes d'ordre 1,2 ou 3 (dit "effet quadratique"). [Remarque sur la méthode des splines].
- On estime des modèles avec seulement la fonction de la durée, on peut utiliser le critère AIC ou BIC pour choisir le meilleur ajustement. La valeur la moins élevée donne le meilleur ajustement (la différence est significative à partir de -2).
- On estime ensuite le modèle avec les covariables sélectionnées.

Choix de la fonction

- $f(t) = a_1 \times t$
- $f(t) = a_1 \times t + a_2 \times t^2$

```

td$t2 = td$t^2
td$t3 = td$t^3

fit1 = glm(d ~ t, data=td, family="binomial")
summ(fit1)

```

MODEL INFO:

Observations: 29241
 Dependent Variable: d
 Type: Generalized linear model
 Family: binomial
 Link function: logit

MODEL FIT:

$\chi^2(1) = 118.42$, $p = 0.00$
 Pseudo- R^2 (Cragg-Uhler) = 0.02
 Pseudo- R^2 (McFadden) = 0.01
 AIC = 8053.95, BIC = 8070.51

Standard errors: MLE

	Est.	S.E.	z val.	p
(Intercept)	-2.94	0.05	-55.60	0.00
t	-0.04	0.00	-10.31	0.00

```

fit2 = glm(d ~ t + t2, data=td, family="binomial")
summ(fit2)

```

MODEL INFO:

Observations: 29241
 Dependent Variable: d
 Type: Generalized linear model
 Family: binomial
 Link function: logit

MODEL FIT:

$\chi^2(2) = 120.96$, $p = 0.00$
 Pseudo- R^2 (Cragg-Uhler) = 0.02

Pseudo- R^2 (McFadden) = 0.01
AIC = 8053.41, BIC = 8078.26

Standard errors: MLE

	Est.	S.E.	z val.	p
(Intercept)	-3.03	0.08	-39.97	0.00
t	-0.02	0.01	-1.57	0.12
t2	-0.00	0.00	-1.57	0.12

```
fit3 = glm(d ~ t + t2 + t3, data=td, family="binomial")
summ(fit3)
```

MODEL INFO:

Observations: 29241

Dependent Variable: d

Type: Generalized linear model

Family: binomial

Link function: logit

MODEL FIT:

$\chi^2(3) = 242.27$, $p = 0.00$

Pseudo- R^2 (Cragg-Uhler) = 0.03

Pseudo- R^2 (McFadden) = 0.03

AIC = 7934.09, BIC = 7967.23

Standard errors: MLE

	Est.	S.E.	z val.	p
(Intercept)	-3.70	0.10	-35.40	0.00
t	0.19	0.02	8.46	0.00
t2	-0.01	0.00	-11.15	0.00
t3	0.00	0.00	11.77	0.00

En comparant les critères d'information (ici AIC), le choix peut se porter sur la forme cubique.

A savoir: la forme cubique n'est pas sans défaut (ou plus généralement des polynômes de degré supérieur à 2), elle est sensible aux outliers, ce qui est le cas ici. Le graphique a été tronqué car le risque ne cesse d'augmenter jusqu'à une valeur proche de 1 au durée élevée.

Il est préférable d'utiliser une méthode de lissage de type "spline cubique" moins violente (ne pas paniquer, c'est même implémenté dans excel).

Estimation du modèle

```
fit = glm(d ~ t + t2 + t3 + gene + csp + diplome, data=td, family="binomial")
summ(fit, digits=4)
```

MODEL INFO:

Observations: 29241

Dependent Variable: d

Type: Generalized linear model

Family: binomial

Link function: logit

MODEL FIT:

$\chi^2(11) = 314.6576$, $p = 0.0000$

Pseudo- R^2 (Cragg-Uhler) = 0.0439

Pseudo- R^2 (McFadden) = 0.0385

AIC = 7877.7075, BIC = 7977.1074

Standard errors: MLE

	Est.	S.E.	z val.	p
(Intercept)	-4.2477	0.1762	-24.1067	0.0000
t	0.1956	0.0225	8.7112	0.0000
t2	-0.0139	0.0012	-11.2060	0.0000
t3	0.0002	0.0000	11.7065	0.0000
gene1	-0.1500	0.0856	-1.7517	0.0798
gene3	-0.2243	0.0852	-2.6335	0.0084
cspArtisans agr	0.8127	0.1934	4.2021	0.0000
cspEmployes	0.7376	0.1218	6.0545	0.0000
cspOuvriers	1.1018	0.1747	6.3063	0.0000
cspProfs interm	0.3558	0.1197	2.9727	0.0030
diplome1	0.0527	0.1122	0.4701	0.6383
diplome3	0.3607	0.1285	2.8061	0.0050

5.2.2 Forme discrète de la durée

- Il s'agit d'introduire la variable de durée dans le modèle comme une variable catégorielle (factor).
- Pas conseillé si beaucoup de points d'observation, ce qui est le cas ici, et surtout si présence de points d'observation sans évènement.
- A l'inverse, si peu de points d'observation, la paramétrisation avec une durée continue n'est pas conseillé.
- La correction de la non proportionnalité peut être plus compliquée (non traité).

(*exemple)

```
glm(d ~ factor(t) + gene + csp + diplome, data=td, family="binomial")
```

On va supposer que l'on ne dispose que de 4 points d'observations. Pour l'exemple, on va créer ces points à partir des quartiles de la durée, et conserver pour chaque personne seulement une observation par intervalle.

Sélection de la dernière observation dans chaque intervalle

```
td$ct4 = quantcut(td$t)
```

On va générer un compteur et un total d'observations en stratifiant *ident* + *ct4*

```
td$n = ave(td$x, td$ident, td$ct4, FUN=cumsum)
td$N = ave(td$x, td$ident, td$ct4, FUN=sum)
```

On conserve la dernière observation dans la strate.

```
td2 = subset(td, n==N)
```

Estimation du modèle

Ici la variable *ct4* a été généré en format de type factor. Attention, si ce n'est pas le cas préciser le format dans le modèle ou le changer au préalable.

```
fit = glm(d ~ ct4 + gene + csp + diplome, data=td2, family=binomial)
summ(fit, digits=4)
```

MODEL INFO:

Observations: 3925

Dependent Variable: d

Type: Generalized linear model
 Family: binomial
 Link function: logit

MODEL FIT:

$\chi^2(11) = 133.9036$, $p = 0.0000$
 Pseudo- R^2 (Cragg-Uhler) = 0.0506
 Pseudo- R^2 (McFadden) = 0.0313
 AIC = 4162.3075, BIC = 4237.6089

Standard errors: MLE

	Est.	S.E.	z val.	p
(Intercept)	-1.9252	0.1690	-11.3941	0.0000
ct4(6,13]	0.5377	0.0921	5.8386	0.0000
ct4(13,24]	0.0070	0.1113	0.0633	0.9495
ct4(24,55]	-0.1592	0.1276	-1.2471	0.2123
gene1	-0.0889	0.0977	-0.9103	0.3627
gene3	-0.2829	0.0966	-2.9289	0.0034
cspArtisans agr	0.9551	0.2248	4.2492	0.0000
cspEmployes	0.8135	0.1348	6.0357	0.0000
cspOuvriers	1.2206	0.2049	5.9580	0.0000
cspProfs interm	0.3966	0.1312	3.0237	0.0025
diplome1	0.1046	0.1249	0.8374	0.4023
diplome3	0.3870	0.1434	2.6995	0.0069

Probabilités estimées à partir du modèle avec la durée seulement

Durées	p
0 à 6 ans	0.21
6 à 13 ans	0.31
13 à 24 ans	0.20
24 à 55 ans	0.17

Interprétation:

Les risques (probabilités) sont estimés sur des intervalles assez long, leur valeur est donc plus élevées qu’avec des durées “continues”: le risque de sortie de l’activité est de 21% jusqu’à 6 ans, toute chose égale par ailleurs.

6 Risque concurrent

Le package est **cmprsk**. Les graphiques sont plutôt affreux, en particulier en raison de la position de la légende. La fonction, assez récente, *ggcompetingrisks* de **survminer** résoud le problème

6.1 Incidence cumulée et test de gray

Le package est **cmprsk**. Les graphiques sont plutôt affreux, en particulier en raison de la position de la légende. La fonction, assez récente, *ggcompetingrisks* de **survminer** résoud le problème

```
# ic et test de gray

act$typinact = as.factor(act$typinact)
ic = cuminc(act$dur, act$typinact)
ic
```

Estimates and Variances:

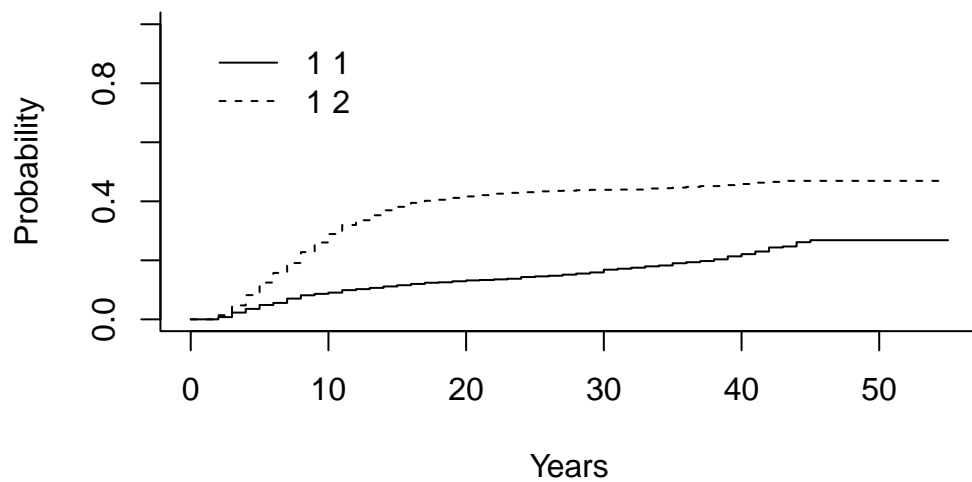
\$est

		10	20	30	40	50
1	1	0.0905531	0.1319179	0.1683808	0.2209676	0.2681875
1	2	0.2891226	0.4165783	0.4390128	0.4586536	0.4693039

\$var

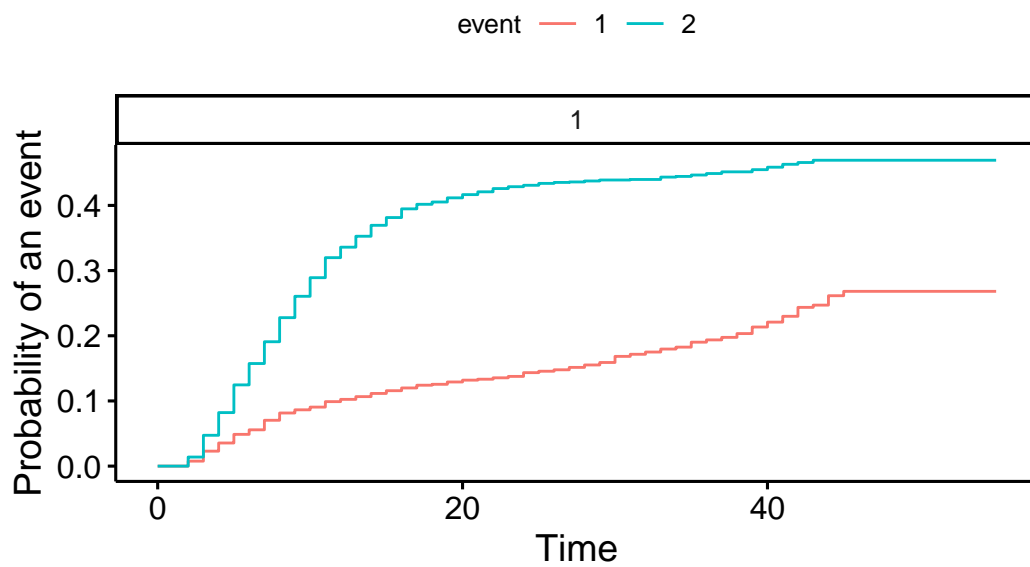
		10	20	30	40	50
1	1	0.00005722054	0.00007953895	0.00009870348	0.0001503380	0.0002954601
1	2	0.00014266865	0.00016866085	0.00017137870	0.0001846356	0.0002059581

```
plot(ic)
```



```
ggcompetingrisks(fit = ic)
```

Cumulative incidence functions



On compare les courbes en ajoutant l'option `groupe`, et ici on indique une version du test avec une pondération de type *Peto-Peto* avec `rho=1`. Pour le graphique avec `ggcompetingrisks` on a le choix entre avoir un graphique par valeur de la variable dont les valeurs sont comparées (graphique par défaut) ou un seul graphique avec les courbes empilées (option `multiple_panels = F`).

Le résultat du test de Gray est affiché automatiquement avec l'option `group`.

```
ic = cuminc(act$dur, act$typinact, group=act$gene, rho=1)
ic
```

Tests:

	stat		pv	df
1	17.20981	0.00018320456	2	
2	21.10132	0.00002617622	2	

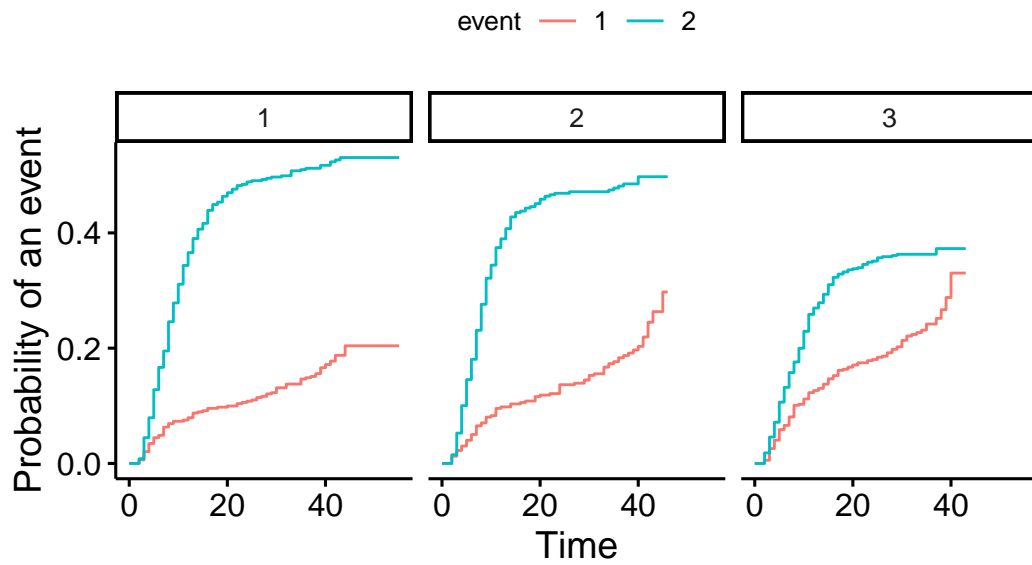
Estimates and Variances:

\$est		10	20	30	40	50
2	1	0.08285888	0.11855967	0.1527784	0.2031056	NA
1	1	0.07317073	0.09963677	0.1312357	0.1715027	0.2041405
3	1	0.11186919	0.17086452	0.2136215	0.3303424	NA
2	2	0.34406102	0.45849678	0.4714256	0.4975086	NA
1	2	0.31097561	0.46976008	0.4969744	0.5173007	0.5307891
3	2	0.22926316	0.33783443	0.3629199	0.3728012	NA

\$var		10	20	30	40	50
2	1	0.0001907296	0.0002629734	0.0003287464	0.0004731326	NA
1	1	0.0001378102	0.0001821243	0.0002333418	0.0003029241	0.0004136609
3	1	0.0001819926	0.0002593861	0.0003156173	0.0026766240	NA
2	2	0.0005659435	0.0006240847	0.0006269723	0.0006856356	NA
1	2	0.0004347758	0.0005050285	0.0005082562	0.0005121396	0.0005289099
3	2	0.0003235388	0.0004092405	0.0004250576	0.0005095554	NA

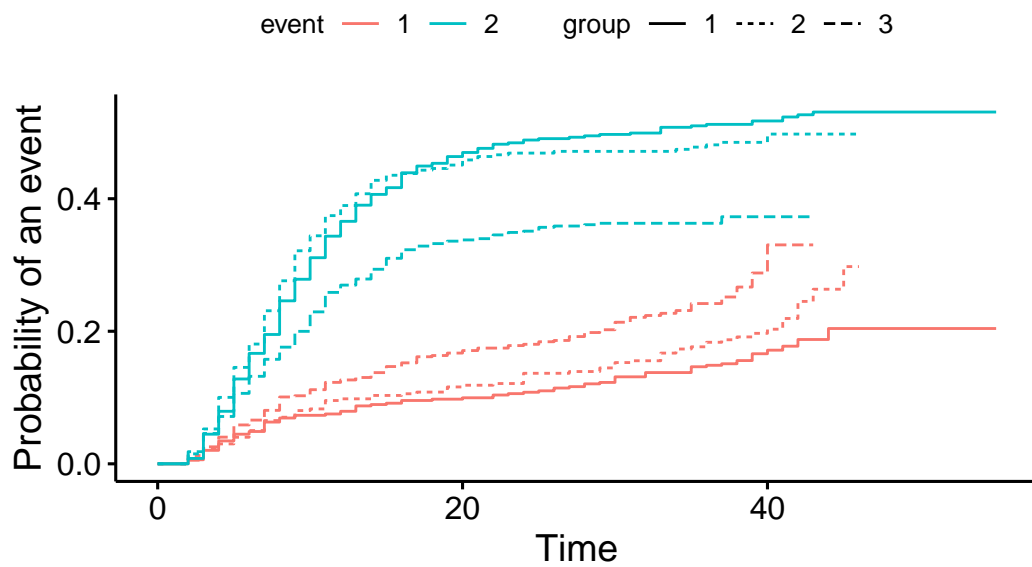
```
ggcompetingrisks(fit = ic)
```

Cumulative incidence functions



```
ggcompetingrisks(fit = ic, multiple_panels = F)
```

Cumulative incidence functions



7 Modèle multinomial

Bienvenue dans le monde de la librairie **nnet** et de la fonction très vieillissante **multinom**.

```
act$T = act$dur
td = uncount(act,T)

td$gene = as.factor(td$gene)
td$gene = relevel(td$gene, ref = "2")
td$csp = as.factor(td$csp)
td$csp = relevel(td$csp, ref = "Cadres")
td$diplome = as.factor(td$diplome)
td$diplome = relevel(td$diplome, ref = 2)

td$typinact = relevel(td$typinact, ref = "0")

td$x=1
td$t = ave(td$x, td$ident, FUN=cumsum)
td$t2 = td$t*td$t

td$e = ifelse(td$t<td$dur,0, td$typinact)

competfit = multinom(formula = e ~ t + t2 + gene + csp + diplome, data = td)
```

```
# weights:  48 (33 variable)
initial value 40536.633413
iter  10 value 14213.686287
iter  20 value 9354.248105
iter  30 value 7218.939115
iter  40 value 6284.031278
iter  50 value 6082.437268
iter  60 value 6063.479237
iter  70 value 6061.798193
iter  70 value 6061.798160
iter  70 value 6061.798124
final value 6061.798124
converged
```

```
competfit
```

Call:

```
multinom(formula = e ~ t + t2 + gene + csp + diplome, data = td)
```

Coefficients:

	(Intercept)	t	t2	gene1	gene3	cspArtisans	agr
1	-11.781028	0.22091891	0.001039854	-1.22398547	1.4399719		-1.1751145
2	-4.909453	-0.06316511	0.002056065	-0.29440093	0.1828764		0.1316466
3	-4.071496	0.07604300	-0.004399414	-0.06812543	-0.3894873		0.9829043

	cspEmployes	cspOuvriers	cspProfs	interm	diplome2	diplome3
1	-0.2460749	-0.4955051		-0.1859166	0.9395357	1.2619093
2	0.5398095	0.8388623		0.1839159	0.2704196	0.8580221
3	0.7762950	1.1251805		0.3917102	-0.1998322	0.0398104

Residual Deviance: 12123.6

AIC: 12189.6

PS: toujours du mal à sortir proprement avec Quarto l'output en utilisant gtsummary (normalement ok dans un script). Mais l'output n'est pas top non plus, les Y_k sont empilées en ligne alors qu'il est préférable de les mettre en colonne.

8 Mise en forme de la base

Base additionnelle mef

La base utilisée pour les analyses a été construite à partir d'une base individus-séquences. Pour obtenir les variables d'analyses (durée, évènement-censure) une mise en forme a été nécessaire. Il s'agit d'une méthode parmi d'autres, qui n'utilise que la base d'origine, et basée sur les comptages et doubles comptages des séquences pour conserver les informations nécessaires.

Extrait de la base d'origine avec des situations types.

```
mef <- read.csv("D:/D/Marc/SMS/FORMATIONS/2022/Durée2/a distribuer/mef.csv")
head(mef, n=29)
```

	ident	age_enq	ageact	act_lab	act
1	102	65	16	Activité	1
2	102	65	24	Activité	1
3	102	65	26	Activité	1
4	102	65	27	inactivité autre, foyer	4
5	102	65	60	retraite	7
6	113	62	14	études, formation	5
7	113	62	19	Activité	1

8	113	62	38	Activité	1
9	113	62	49	Activité	1
10	113	62	50	Activité	1
11	127	50	19	Activité	1
12	127	50	26	chomage	2
13	127	50	27	Activité	1
14	127	50	28	Activité	1
15	127	50	29	Activité	1
16	127	50	30	Activité	1
17	132	63	17	Activité	1
18	132	63	25	Activité	1
19	132	63	26	Activité	1
20	132	63	27	Activité	1
21	132	63	30	Activité	1
22	132	63	34	Activité	1
23	132	63	35	Activité	1
24	132	63	46	Activité	1
25	132	63	47	Activité	1
26	132	63	49	Activité	1
27	132	63	56	Activité	1
28	132	63	57	Activité	1
29	132	63	60	retraite	7

- **ident=102:** Premier emploi à 16 ans, sortie définitive de l'emploi à 27 ans jusqu'à la retraite.
- **ident=113:** La première séquence n'est pas l'emploi (études/formation pro). Premier emploi à 19 ans. Toujours en activité au moment de l'enquête.
- **ident=127:** Premier emploi à 19 ans, suivi d'une période de chômage d'un an à l'âge de 26 ans. Toujours en activité depuis la reprise de l'emploi.
- **ident=132:** Premier emploi à 17 ans, et a toujours travaillé jusqu'à la retraite. Etait déjà à la retraite au moment de l'enquête: retraite à 60 ans et enquête à 63 ans.

8.1 On conserve les premières séquences = emploi

On supprime la première séquence de *ident=113* comme le début de l'exposition est l'âge au premier emploi.

Idée: on repère si la première séquence observée est un emploi. Si ce n'est pas le cas, on supprime la ligne (*tact=0*).

```

mef$act2 = ifelse(mef$act == 1, 1, 0)
mef$tact = ave(mef$act2, mef$ident, FUN=cumsum)
mef = subset(mef, mef$tact>0)

mef$tact = NULL

```

8.2 Age à la 1ere séquence d'inactivité: retraite et autres causes de sortie

Même principe, mais on devra faire un double comptage pour la situation où après une période d'activité succède une période d'inactivité qui sera suivie d'une nouvelle séquence d'emploi, puis d'une nouvelle séquence de non emploi. C'est le cas pour *ident=127*. Avec **ttinact=1** on repère donc la première sortie de l'emploi.

```

mef$inact = ifelse(mef$act != 1, 1, 0)
mef$tinact = ave(mef$inact, mef$ident, FUN=cumsum)
mef$ttinact = ave(mef$tinact, mef$ident, FUN=cumsum)
head(mef, n=29)

```

	ident	age_enq	ageact		act_lab	act	act2	inact	tinact	ttinact
1	102	65	16		Activité	1	1	0	0	0
2	102	65	24		Activité	1	1	0	0	0
3	102	65	26		Activité	1	1	0	0	0
4	102	65	27	inactivité autre, foyer		4	0	1	1	1
5	102	65	60		retraite	7	0	1	2	3
7	113	62	19		Activité	1	1	0	0	0
8	113	62	38		Activité	1	1	0	0	0
9	113	62	49		Activité	1	1	0	0	0
10	113	62	50		Activité	1	1	0	0	0
11	127	50	19		Activité	1	1	0	0	0
12	127	50	26		chomage	2	0	1	1	1
13	127	50	27		Activité	1	1	0	1	2
14	127	50	28		Activité	1	1	0	1	3
15	127	50	29		Activité	1	1	0	1	4
16	127	50	30		Activité	1	1	0	1	5
17	132	63	17		Activité	1	1	0	0	0
18	132	63	25		Activité	1	1	0	0	0
19	132	63	26		Activité	1	1	0	0	0
20	132	63	27		Activité	1	1	0	0	0
21	132	63	30		Activité	1	1	0	0	0
22	132	63	34		Activité	1	1	0	0	0

23	132	63	35	Activité	1	1	0	0	0
24	132	63	46	Activité	1	1	0	0	0
25	132	63	47	Activité	1	1	0	0	0
26	132	63	49	Activité	1	1	0	0	0
27	132	63	56	Activité	1	1	0	0	0
28	132	63	57	Activité	1	1	0	0	0
29	132	63	60	retraite	7	0	1	1	1

On reporte l'âge à la retraite si la retraite est la première séquence d'inactivité (ageinact avec act=7), ou l'âge à la sortie de l'emploi pour une autre cause (ageinact avec act différent de 7).

```
mef$ageret = ifelse(mef$ttinact==1 & mef$act==7, mef$ageact,0)
mef$ageinact = ifelse(mef$ttinact==1 & mef$act!=7, mef$ageact,0)
head(mef, n=29)
```

	ident	age_enq	ageact		act_lab	act	act2	inact	tinact	ttinact
1	102	65	16		Activité	1	1	0	0	0
2	102	65	24		Activité	1	1	0	0	0
3	102	65	26		Activité	1	1	0	0	0
4	102	65	27	inactivité autre, foyer		4	0	1	1	1
5	102	65	60		retraite	7	0	1	2	3
7	113	62	19		Activité	1	1	0	0	0
8	113	62	38		Activité	1	1	0	0	0
9	113	62	49		Activité	1	1	0	0	0
10	113	62	50		Activité	1	1	0	0	0
11	127	50	19		Activité	1	1	0	0	0
12	127	50	26		chomage	2	0	1	1	1
13	127	50	27		Activité	1	1	0	1	2
14	127	50	28		Activité	1	1	0	1	3
15	127	50	29		Activité	1	1	0	1	4
16	127	50	30		Activité	1	1	0	1	5
17	132	63	17		Activité	1	1	0	0	0
18	132	63	25		Activité	1	1	0	0	0
19	132	63	26		Activité	1	1	0	0	0
20	132	63	27		Activité	1	1	0	0	0
21	132	63	30		Activité	1	1	0	0	0
22	132	63	34		Activité	1	1	0	0	0
23	132	63	35		Activité	1	1	0	0	0
24	132	63	46		Activité	1	1	0	0	0
25	132	63	47		Activité	1	1	0	0	0
26	132	63	49		Activité	1	1	0	0	0

27	132	63	56	Activité	1	1	0	0	0
28	132	63	57	Activité	1	1	0	0	0
29	132	63	60	retraite	7	0	1	1	1

	ageret	ageinact
1	0	0
2	0	0
3	0	0
4	0	27
5	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	26
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	60	0

On reporte les âges sur toutes les lignes en faisant un total, les âges avant la sortie étant toujours égaux à 0.

```
mef$ageret = ave(mef$ageret, mef$ident, FUN=sum)
mef$ageinact = ave(mef$ageinact, mef$ident, FUN=sum)
head(mef, n=29)
```

	ident	age_enq	ageact	act_lab	act	act2	inact	tinact	ttinact
1	102	65	16	Activité	1	1	0	0	0

2	102	65	24	Activité	1	1	0	0	0
3	102	65	26	Activité	1	1	0	0	0
4	102	65	27	inactivité autre, foyer	4	0	1	1	1
5	102	65	60	retraite	7	0	1	2	3
7	113	62	19	Activité	1	1	0	0	0
8	113	62	38	Activité	1	1	0	0	0
9	113	62	49	Activité	1	1	0	0	0
10	113	62	50	Activité	1	1	0	0	0
11	127	50	19	Activité	1	1	0	0	0
12	127	50	26	chomage	2	0	1	1	1
13	127	50	27	Activité	1	1	0	1	2
14	127	50	28	Activité	1	1	0	1	3
15	127	50	29	Activité	1	1	0	1	4
16	127	50	30	Activité	1	1	0	1	5
17	132	63	17	Activité	1	1	0	0	0
18	132	63	25	Activité	1	1	0	0	0
19	132	63	26	Activité	1	1	0	0	0
20	132	63	27	Activité	1	1	0	0	0
21	132	63	30	Activité	1	1	0	0	0
22	132	63	34	Activité	1	1	0	0	0
23	132	63	35	Activité	1	1	0	0	0
24	132	63	46	Activité	1	1	0	0	0
25	132	63	47	Activité	1	1	0	0	0
26	132	63	49	Activité	1	1	0	0	0
27	132	63	56	Activité	1	1	0	0	0
28	132	63	57	Activité	1	1	0	0	0
29	132	63	60	retraite	7	0	1	1	1
ageret ageinact									
1	0	27							
2	0	27							
3	0	27							
4	0	27							
5	0	27							
7	0	0							
8	0	0							
9	0	0							
10	0	0							
11	0	26							
12	0	26							
13	0	26							
14	0	26							
15	0	26							
16	0	26							

17	60	0
18	60	0
19	60	0
20	60	0
21	60	0
22	60	0
23	60	0
24	60	0
25	60	0
26	60	0
27	60	0
28	60	0
29	60	0

8.3 On conserve une ligne par individu

On génère un compteur sur les séquences, et on conserve seulement une ligne par personne.

```
mef$x=1
mef$n = ave(mef$x, mef$ident, FUN=cumsum)
mef = subset(mef, mef$n==1)
mef = mef[c("ident", "age_enq", "ageact", "act", "ageret", "ageinact")]
head(mef)
```

	ident	age_enq	ageact	act	ageret	ageinact
1	102	65	16	1	0	27
7	113	62	19	1	0	0
11	127	50	19	1	0	26
17	132	63	17	1	60	0