

# Modules StatMed 1 & 2

Étude d'association et analyse de survie en oncologie

[masedki.github.io](https://masedki.github.io)

Université Paris-Saclay & NewMed





# Mesure d'association et régression logistique

- Fixer le vocabulaire
- Pratiquer sur un jeu de données
- Accessoirement installer une ou plusieurs librairies sur R

```
require(riskCommunicator) ## on peut utiliser library()  
require(Epi)  
data("framingham")  
?framingham  
summary(framingham)
```

## Un test basique : lien entre deux variables binaires

```
attach(framingham)
tab = table(DIABETES, HOSPMI)
chisq.test(tab, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 94.57, df = 1, p-value < 2.2e-16
```

# Risk Relatif (RR) et Odds Ratio (OR)

Table	Maladie (Oui)	Maladie(Non)
FR (Oui)	a	b
FR(Non)	c	d

- a, b, c et d sont des comptages
- Le RR est donné par

$$\frac{\frac{a}{a+b}}{\frac{c}{c+d}}$$

- L'OR

$$\frac{\frac{a}{b}}{\frac{c}{d}}$$

# Le tableau de contingence sous R

```
table(DIABETES, HOSPMI)
```

```
##           HOSPMI
## DIABETES      0      1
##           0 10061  1036
##           1   412   118
```

```
table(1-DIABETES, 1-HOSPMI)
```

```
##
##           0      1
##    0   118   412
##    1  1036 10061
```

# RR et OR sous R

```
twoby2(1-DIABETES, 1-HOSPMI)
```

```
## 2 by 2 table analysis:
```

```
## -----
```

```
## Outcome      : 0
```

```
## Comparing    : 0 vs. 1
```

```
##
```

```
##           0           1      P(0) 95% conf. interval
```

```
## 0   118    412  0.2226    0.1892    0.2601
```

```
## 1 1036 10061  0.0934    0.0881    0.0989
```

```
##
```

```
##                                     95% conf. interval
```

```
##                               Relative Risk: 2.3848    2.0133    2.8248
```

```
##                               Sample Odds Ratio: 2.7814    2.2447    3.4465
```

```
##                               Probability difference: 0.1293    0.0955    0.1670
```

```
##
```

```
##                               Asymptotic P-value: 0.0000
```

```
## -----
```

# L'OR se généralise via la régression logistique

```
m = glm(HOSPMI~DIABETES, data=framingham, family = binomial)
summary(m)

##
## Call:
## glm(formula = HOSPMI ~ DIABETES, family = binomial, data = framingham)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.27330    0.03263 -69.671  <2e-16 ***
## DIABETES      1.02296    0.10939   9.351  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 7521.2  on 11626  degrees of freedom
## Residual deviance: 7447.5  on 11625  degrees of freedom
## AIC: 7451.5
##
## Number of Fisher Scoring iterations: 5
```



# Lien avec l'OR

On note  $Y$  (la maladie) et  $X$  (le facteur de risque). La régression logistique repose sur

$$\mathbb{P}(Y = 1|X = x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

De manière équivalente

$$\ln \left[ \frac{\mathbb{P}(Y = 1|X = x)}{\mathbb{P}(Y = 0|X = x)} \right] = \beta_0 + \beta_1 x$$

```
exp(coefficients(m))
```

```
## (Intercept)    DIABETES  
##    0.1029719    2.7814175
```

# Un autre facteur de risque quantitatif

```
m1 = glm(HOSPMI~BMI, data=framingham, family = binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = HOSPMI ~ BMI, family = binomial, data = framingham)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.186340   0.191206 -16.664  < 2e-16 ***
## BMI          0.037494   0.007146   5.247 1.55e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 7483.8  on 11574  degrees of freedom
## Residual deviance: 7457.3  on 11573  degrees of freedom
##   (52 observations effacées parce que manquantes)
## AIC: 7461.3
##
## Number of Fisher Scoring iterations: 5
```

# Examen de l'OR

```
exp(coefficients(m1))
```

```
## (Intercept)          BMI  
## 0.04132285 1.03820547
```

```
exp(confint(m1))
```

```
## Attente de la réalisation du profilage...
```

```
##                2.5 %      97.5 %  
## (Intercept) 0.02843493 0.06017651  
## BMI        1.02366643 1.05275247
```

# Modèle logistique multivarié

```
mv = glm(HOSPMI~AGE + SEX + CURSMOKE + CIGPDAY + DIABETES + BMI,  
        data=framingham, family = binomial)  
summary(mv)
```

```
##  
## Call:  
## glm(formula = HOSPMI ~ AGE + SEX + CURSMOKE + CIGPDAY + DIABETES +  
##     BMI, family = binomial, data = framingham)  
##  
## Coefficients:  
##             Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -3.0451918  0.3250160  -9.369  < 2e-16 ***  
## AGE          0.0253844  0.0035392   7.172 7.37e-13 ***  
## SEX         -1.1949699  0.0698267 -17.113  < 2e-16 ***  
## CURSMOKE     0.3732739  0.1025009   3.642 0.000271 ***  
## CIGPDAY     -0.0007418  0.0038573  -0.192 0.847499  
## DIABETES     0.8679602  0.1153942   7.522 5.41e-14 ***  
## BMI          0.0355116  0.0080000   4.439 9.04e-06 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
##     Null deviance: 7423.6  on 11497  degrees of freedom  
## Residual deviance: 6912.2  on 11491  degrees of freedom  
## (129 observations effacées parce que manquantes)  
## AIC: 6926.2  
##  
## Number of Fisher Scoring iterations: 5
```

# Les OR ajustés

```
exp(coefficients(mv))
```

```
## (Intercept)          AGE          SEX    CURSMOKE    CIGPDAY    DIABETES
## 0.04758719  1.02570935  0.30271308  1.45248216  0.99925847  2.38204688
##          BMI
## 1.03614966
```

```
exp(confint(mv))
```

```
## Attente de la réalisation du profilage...
```

```
##          2.5 %    97.5 %
## (Intercept) 0.02517437 0.09002621
## AGE         1.01862767 1.03286052
## SEX         0.26374600 0.34681552
## CURSMOKE    1.18703547 1.77418282
## CIGPDAY     0.99166910 1.00678269
## DIABETES    1.89353407 2.97769202
## BMI         1.01992144 1.05241935
```

# Données de survie

```
require(survival)
```

```
## Le chargement a nécessité le package : survival
```

```
data(cancer, package="survival")  
?lung  
head(lung)
```

##	inst	time	status	age	sex	ph.ecog	ph.karno	pat.karno	meal.cal	wt.loss
## 1	3	306	2	74	1	1	90	100	1175	NA
## 2	3	455	2	68	1	0	90	90	1225	15
## 3	3	1010	1	56	1	0	90	90	NA	15
## 4	5	210	2	57	1	1	90	60	1150	11
## 5	1	883	2	60	1	0	100	90	NA	0
## 6	12	1022	1	74	1	1	50	80	513	0

# Analyse de survie : ses ingrédients

L'analyse de survie permet de modéliser le **temps qui s'écoule jusqu'à ce qu'un évènement se produise**, de comparer le temps écoulé jusqu'à l'évènement entre différents groupes ou de déterminer l'effet de variables quantitatives sur le temps écoulé jusqu'à l'évènement.

## Vocabulaire classique

- Censure
- Fonction de survie
- Estimation par Kaplan-Meier
- Risque instantané (*hazard* en anglais)

# Notion de censure

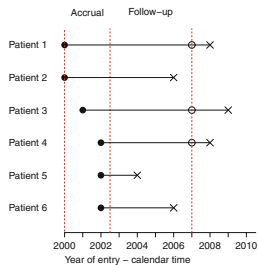


Figure 1: Temps réel

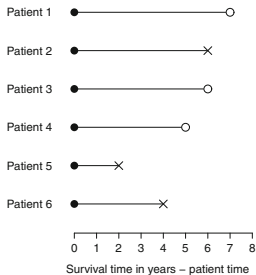


Figure 2: Résumé

Patient	Survtime	Status
1	7	0
2	6	1
3	6	0
4	5	0
5	2	1
6	4	1

Figure 3: Données



# Fonction de survie

La fonction de survie est la probabilité qu'un individu survive (ou la probabilité que l'évènement qui nous intéresse ne se produise pas) jusqu'au temps  $t$  inclus. C'est la probabilité que l'évènement (par exemple, le décès) ne se soit pas encore produit.

$$S(t) = \mathbb{P}(T > t).$$

On a  $0 \leq S(t) \leq 1$  car c'est une probabilité et  $T \geq 0$  (c'est un temps).

# Son estimation par Kaplan-Meier

Pour les curieux (c'est un estimateur de  $S(t)$ )

$$\widehat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

- $n_i$  nombre de patients à risque au temps  $t_i$
- $d_i$  le nombre d'évènements au temps  $t_i$

# Risque instantané (*hazard*) : encore un truc abstrait !!

C'est le taux de décès instantané. Il s'agit de la probabilité que, étant donné qu'un sujet a survécu jusqu'à l'instant  $t$ , il succombe dans le petit intervalle de temps suivant, divisé par la longueur de cet intervalle. Formellement

$$h(t) = \lim_{\delta \rightarrow 0} \frac{\mathbb{P}(t < T < t + \delta | T > t)}{\delta}.$$

# Il est préférable de visualiser

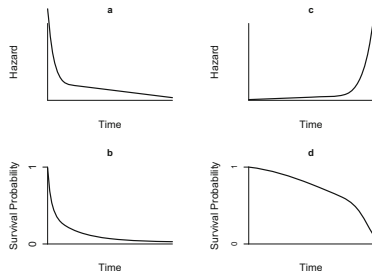


Figure 4: Deux fonctions de risque instantané

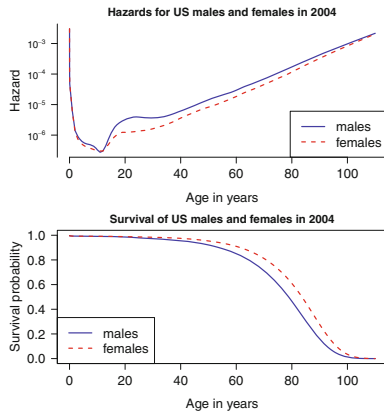


Figure 5: Décès aux USA jusqu'en 2004

# La librairie survival fait le job

```
s <- Surv(lung$time, lung$status)
class(s)
```

```
## [1] "Surv"
```

```
s[1:10]
```

```
## [1] 306 455 1010+ 210 883 1022+ 310 361 218 166
```

```
head(lung)
```

```
##   inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss
## 1    3  306      2  74   1        1        90       100     1175      NA
## 2    3  455      2  68   1         0        90        90     1225     15
## 3    3 1010      1  56   1         0        90        90        NA     15
## 4    5  210      2  57   1         1        90        60     1150     11
## 5    1  883      2  60   1         0       100        90        NA      0
## 6   12 1022      1  74   1         1        50        80     513      0
```

# Notre première courbe de Kaplan-Meier

```
sfit <- survfit(Surv(time, status)~1, data=lung) ## aucune stratification
sfit
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = lung)
##
##           n events median 0.95LCL 0.95UCL
## [1,] 228      165      310      285      363
```

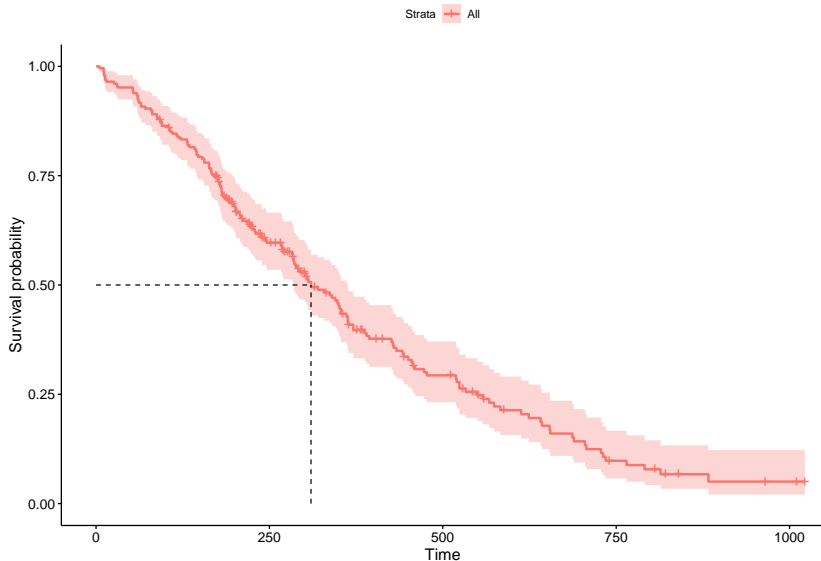
```
summary(sfit, times=seq(0, 1000, 250))
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = lung)
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    0     228      0   1.0000  0.0000   1.0000    1.000
##   250     115     89   0.5967  0.0333   0.5349    0.666
##   500      41     49   0.2933  0.0351   0.2320    0.371
##   750       9     23   0.0979  0.0266   0.0575    0.167
##  1000        2      4   0.0503  0.0228   0.0207    0.123
```

```
## Commande à tester : plot(sfit)
```

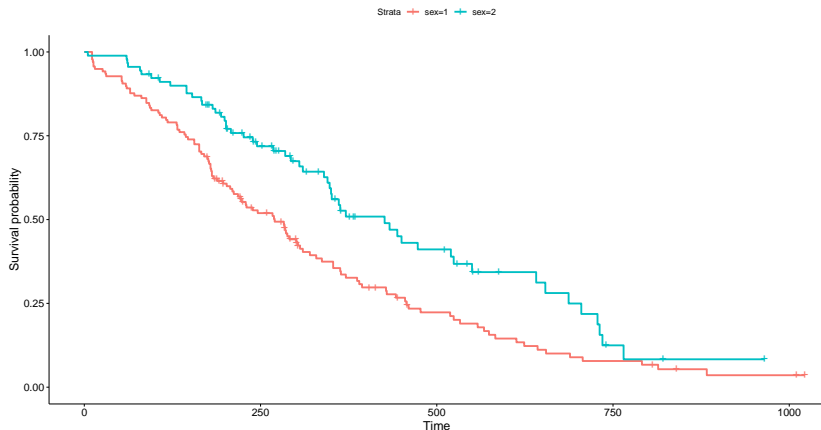
# Une librairie pour améliorer les figures

```
require(survminer)
ggsurvplot(sfit, surv.median.line = "hv")
```



# Effet d'une variable binaire comme un traitement, sexe ou facteur de risque

```
sfit1 <- survfit(Surv(time, status)~sex, data=lung)  
ggsurvplot(sfit1)
```

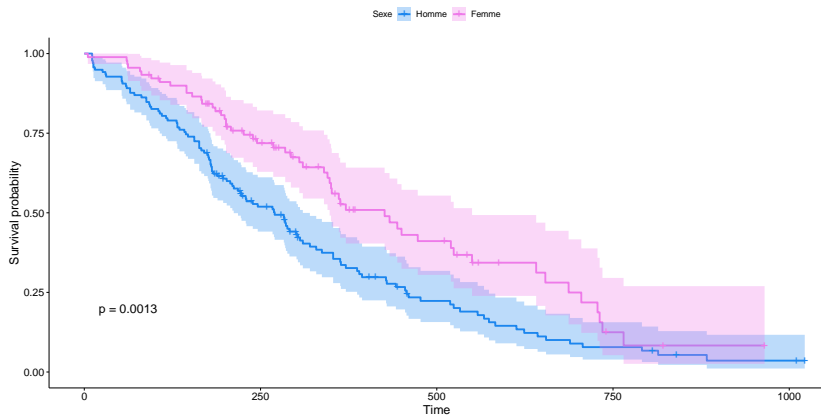




# Ajout d'infos sur la figure : test du log-rank

```
ggsurvplot(sfit1, conf.int=TRUE, pval=TRUE, risk.table=FALSE,  
  legend.labs=c("Homme", "Femme"), legend.title="Sexe",  
  palette=c("dodgerblue2", "orchid2"),  
  title="Courbes de Kaplan-Meier pour la survie au Cancer du poumon",  
  risk.table.height=.15)
```

Courbes de Kaplan-Meier pour la survie au Cancer du poumon



# Quantifier l'effet d'une variable sur la survie

```
survdif(Surv(time, status)~sex, data=lung)
```

```
## Call:
## survdif(formula = Surv(time, status) ~ sex, data = lung)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=1 138      112      91.6      4.55      10.3
## sex=2  90       53      73.4      5.68      10.3
##
## Chisq= 10.3 on 1 degrees of freedom, p= 0.001
```

- Le test du log-rank fonctionne comme à un test d'indépendance du khi-deux.
- Une hypothèse de modélisation :

L'analyse de survie compare les fonctions de survie de différents groupes. Si vous avez suivi les deux groupes jusqu'à ce que tout le monde meure, les deux courbes de survie se termineront à 0%, mais un groupe pourrait avoir survécu en moyenne beaucoup plus longtemps que l'autre. L'analyse de survie y parvient en comparant les risques instantanés à différents moments de la période d'observation. **L'analyse de survie ne suppose pas que le risque est constant, mais que le rapport des risques entre les groupes est constant dans le temps.**

La régression des risques instantanés proportionnels (*Proportional Hazards*), également appelée régression de Cox, est l'approche la plus courante pour évaluer l'effet de différentes variables sur la survie.

# Le modèle Cox PH

$$\ln [h(t)] = \ln [h_0(t)] + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

- Une seule variable d'exposition catégorielle ( $x_1 = 1$  et  $x_1 = 0$ )

$$h(t) = h_0(t) \times e^{\beta_1 x_1}$$

- On peut estimer le Hazard Ratio, en comparant les individus exposés aux individus non exposés au temps  $t$  :

$$HR(t) = \frac{h_1(t)}{h_0(t)} = e^{\beta_1}.$$

On retrouve le HR du modèle de Cox qui est constant dans le temps.

# Comment ça marche sous R

```
coxfit <- coxph(Surv(time, status)~sex, data=lung)
summary(coxfit)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex, data = lung)
##
##      n= 228, number of events= 165
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## sex -0.5310      0.5880   0.1672 -3.176  0.00149 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## sex      0.588      1.701   0.4237   0.816
##
## Concordance= 0.579 (se = 0.021 )
## Likelihood ratio test= 10.63 on 1 df,  p=0.001
## Wald test               = 10.09 on 1 df,  p=0.001
## Score (logrank) test = 10.33 on 1 df,  p=0.001
```

# Interprétation

La colonne  $\exp(\text{coef})$  contient  $e^{\beta_1}$ . Il s'agit du HR, l'effet multiplicatif de cette variable sur le taux de risque (pour chaque unité d'augmentation de cette variable). Ainsi, pour une variable catégorielle comme le sexe, passer de l'homme (base) à la femme entraîne une réduction du risque d'environ 40%. On peut également inverser le signe de la colonne coef et prendre  $\exp(0.531)$ , ce qui peut être interprété comme le fait d'être un homme entraîne une augmentation du risque de 1.7 fois, ou comme le fait que les hommes meurent à un taux par unité de temps environ 1.7 fois supérieur à celui des femmes (les femmes meurent à un taux par unité de temps 0.588 fois supérieur à celui des hommes).

## À retenir :

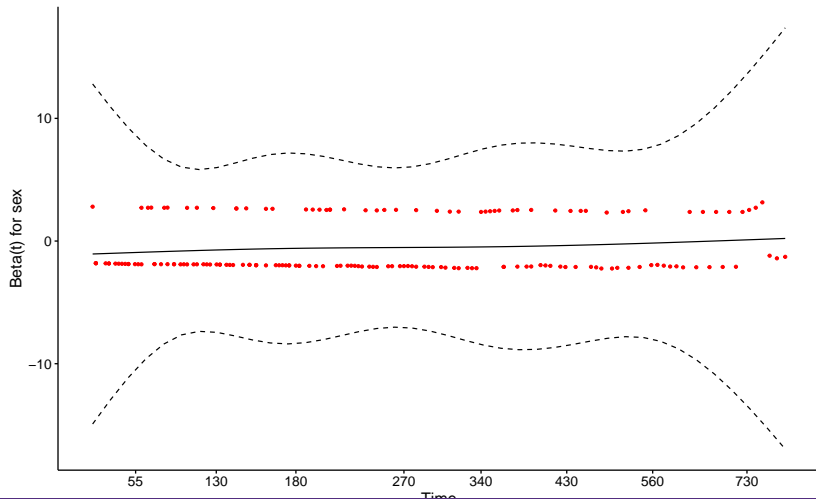
- $HR = 1$  : pas d'effet
- $HR > 1$  : augmentation du risque
- $HR < 1$  : réduction du risque (protection)

# Vérifications de validité du modèle de Cox PH

```
test.ph <- cox.zph(coxfit)  
ggcoxzph(test.ph)
```

Global Schoenfeld Test p: 0.09063

Schoenfeld Individual Test p: 0.0906



# Cox PH multivarié

```
coxfitmv <- coxph(Surv(time,status)~sex+age+ph.ecog+ph.karno+pat.karno+meal.cal+wt.loss,  
                 data=lung)  
coxfitmv
```

```
## Call:  
## coxph(formula = Surv(time, status) ~ sex + age + ph.ecog + ph.karno +  
##       pat.karno + meal.cal + wt.loss, data = lung)  
##  
##              coef exp(coef) se(coef)      z      p  
## sex          -5.509e-01  5.765e-01  2.008e-01 -2.743 0.00609  
## age           1.065e-02  1.011e+00  1.161e-02  0.917 0.35906  
## ph.ecog       7.342e-01  2.084e+00  2.233e-01  3.288 0.00101  
## ph.karno      2.246e-02  1.023e+00  1.124e-02  1.998 0.04574  
## pat.karno    -1.242e-02  9.877e-01  8.054e-03 -1.542 0.12316  
## meal.cal      3.329e-05  1.000e+00  2.595e-04  0.128 0.89791  
## wt.loss      -1.433e-02  9.858e-01  7.771e-03 -1.844 0.06518  
##  
## Likelihood ratio test=28.33 on 7 df, p=0.0001918  
## n= 168, number of events= 121  
##      (60 observations effacées parce que manquantes)
```

# Bonus : données TCGA

**L'Atlas du génome du cancer (TCGA)** est le fruit d'une collaboration entre le **National Cancer Institute (NCI)** et le **National Human Genome Research Institute (NHGRI)**, qui a permis de recueillir de nombreuses données cliniques et génomiques sur 33 types de cancer.

```
# Install the main RTCGA package
#BiocManager::install("RTCGA")
# Install the clinical and mRNA gene expression data packages
#BiocManager::install("RTCGA.clinical")
#BiocManager::install("RTCGA.mRNA")
require(RTCGA)
require(RTCGA.clinical)
dim(BRCA.clinical)
```

```
## [1] 1098 3703
```



# Bonus : extraction des données

Nous allons utiliser la fonction **survivalTCGA()** du package **RTCGA** pour extraire les informations de survie des données cliniques. Pour ce faire, elle examine le statut vital (mort ou vivant) et crée une variable temporelle qui est soit le nombre de jours avant le décès. On lui donne une liste d'ensembles de données cliniques à extraire, et un vecteur de caractères de variables à extraire. Examinons le cancer du sein, le cancer de l'ovaire et le glioblastome multiforme. Extrayons simplement le type de cancer (**admin.disease\_code**).

```
clin <- survivalTCGA(BRCA.clinical, OV.clinical, GBM.clinical,  
                    extract.cols="admin.disease_code")  
# Show the first few lines  
head(clin)
```

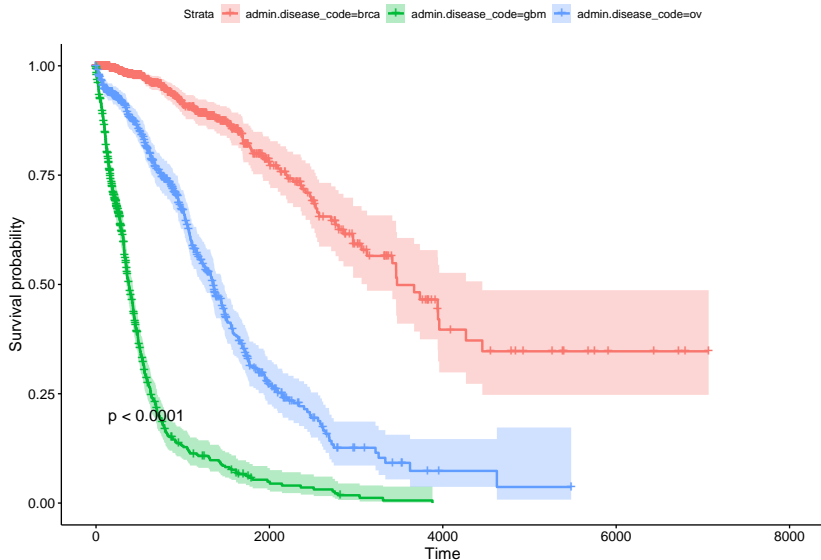
##	times	bcr_patient_barcode	patient.vital_status	admin.disease_code
## 379.31.0	3767	TCGA-3C-AAAU	0	brca
## 379.31.0.1	3801	TCGA-3C-AALI	0	brca
## 379.31.0.2	1228	TCGA-3C-AALJ	0	brca
## 379.31.0.3	1217	TCGA-3C-AALK	0	brca
## 379.31.0.4	158	TCGA-4H-AAAK	0	brca
## 379.31.0.5	1477	TCGA-5L-AATO	0	brca

```
table(clin$admin.disease_code)
```

```
##  
## brca  gbm  ov  
## 1098  595  576
```

# Les fonctions de survie pour les 3 cancers

```
sfit_tgca <- survfit(Surv(times, patient.vital_status)~admin.disease_code, data=clin)  
ggsurvplot(sfit_tgca, conf.int=TRUE, pval=TRUE)
```



# Interpréter le risque par rapport au cancer du sein

```
cox_tcga <- coxph(Surv(times, patient.vital_status)~admin.disease_code, data=clin)
cox_tcga
```

```
## Call:
## coxph(formula = Surv(times, patient.vital_status) ~ admin.disease_code,
##       data = clin)
##
##               coef exp(coef) se(coef)      z      p
## admin.disease_codegbm  2.8875   17.9476   0.1129 25.57 <2e-16
## admin.disease_codeov   1.5470    4.6973   0.1153 13.42 <2e-16
##
## Likelihood ratio test=904.3  on 2 df, p=< 2.2e-16
## n= 2269, number of events= 847
```

# Table of Contents