

## DATA, R INTRO, AND EXERCISES

The course material, including this file, is found on <https://multi-state-book.github.io/barcelona2024/>

### R packages

We will be working with the following packages:

- `survival`: Main package for survival analysis
- `mets`: Analysis of **M**ultivariate **E**vent **T**imes
- `pseudo`: Computes Pseudo-Observations for Modeling
- `geepack`: Generalized Estimating Equation Package
- (`ggplot2`: Create Elegant Data Visualisations Using the Grammar of Graphics)

#### `survival`

<https://CRAN.R-project.org/package=survival>

We will use the following functions:

Function	Description
<code>Surv()</code>	Creates a survival object, to be used as a response variable in a model formula
<code>survfit()</code>	Computes Nelson-Aalen (NA), Kaplan-Meier (KM) and Aalen-Johansen (AJ) estimators
<code>survdiff()</code>	Calculates logrank test
<code>summary()</code>	Summary of a survival curve. If argument <code>times = x</code> is added only this time-point <code>x</code> is listed
<code>print()</code>	Prints a short summary of a survival curve. Restricted mean survival time (RMST) and expected number of years lost (YL) until <code>x</code> can be calculated using <code>rmean = x</code>
<code>plot()</code>	Plots NA ( <code>fun="cumhaz"</code> ), KM, and AJ estimators
<code>coxph()</code>	Fits Cox models and Fine-Gray models
<code>finegray()</code>	Prepares competing risks data (Fine-Gray model) to used before <code>coxph()</code> for competing risks data

Kaplan-Meier estimator:

```
summary(km <- survfit(Surv(days,status!=0)~tment,data=pb3))
```

Cox model:

```
summary(coxph(Surv(days,status!=0)~tment,data=pb3))
```

Fine-Gray model:

```
pb3$fstatus <- factor(pb3$status, 0:2, labels=c("cens", "trans", "death"))  
pdata <- finegray(Surv(days,fstatus) ~ .,data=pb3, etype="trans")  
summary(coxph(Surv(fgstart,fgstop,fgstatus)~tment, weight=fgwt, data=pdata))
```

**mets**

<https://cran.r-project.org/package=mets>

We will only use a few functions from the package for recurrent events analysis. Illustrations will be giving during the lectures.

Function	Description
recurrentMarginal()	Cook-Lawless estimator
recreg()	Ghosh-Lin model

**pseudo**

<https://cran.r-project.org/package=pseudo>

The package consists of four functions for computing pseudo observations (POs) from different non-parametric estimators.

Function	Description
pseudosurv()	based on Kaplan-Meier estimator
pseudomean()	RMST based on the Kaplan-Meier estimator
pseudoci()	based on Aalen-Johansen estimator
pseudoyl()	years lost (YL) based on the Aalen-Johansen estimator

All functions need a *time variable*, a *status variable*, and a *time scalar* or *time vector*.

For pseudosurv and pseudomean, the status variable should be binary (0,1) with 0 meaning censoring and 1 an event.

For pseudoci and pseudoyl, the status variable should be categorical (0, 1, 2, ...) and ordered such that 0 again means censoring (no event) and 1, 2, ... the different event types. NB: it should *NOT* be a factor variable.

When the POs have been created and merged with the original data and potentially reshaped as long format (in case with multiple time points), the function geese() (see next section) from package 'geepack' is used to fit the PO models.

Calculate POs at single time point (year 2) and add to data

```
pb3$followup <- pb3$days/365.25
pb3$fail      <- as.numeric(with(pb3, status>0))

po2 <- pseudosurv(pb3$followup, pb3$fail, tmax = 2)
pb3$po2 <- as.vector(po2$pseudo)
```

Calculate POs at multiple time points (year 1, 2, and 3) and reshape data to long format

```
potsurv <- pseudosurv(pb3$followup, pb3$fail, tmax = 1:3)
longpb3 <- NULL
for(it in 1:length(potsurv$time)){
  longpb3 <- rbind(longpb3,
                   cbind(pb3,
                          pseudo = 1-potsurv$pseudo[,it],
                          tpseudo = potsurv$time[it],
                          id       = 1:nrow(pb3)))
}
longpb3 <- longpb3[order(longpb3$id),]
```

geepack

<https://cran.r-project.org/package=geepack>

From this package we use the generalized estimating equation (GEE) function `geese()` to fit models for pseudo observations.

For a single time point:

```
geese(eo2 ~ tment, data = pb3, id = id, mean.link = "identity")
```

For multiple time points:

```
geese(pseudo~as.factor(tpseudo) + tment, id=id, data=longpb3,
      mean.link="cloglog", corstr="independence")
```

The following link functions are supported: identity, logit, probit, cloglog, log, and inverse.

*NB:* Defining two summary functions for summarizing a geese fit (one without `exp(est)` one that does)

```
posumm<-function(pofit,d=6){
  round(cbind(
    Est   = pofit$beta,
    SD    = sqrt(diag(pofit$vbeta)),
    lo.ci = pofit$beta-1.96*sqrt(diag(pofit$vbeta)),
    up.ci = pofit$beta+1.96*sqrt(diag(pofit$vbeta)),
    Wald  = (pofit$beta/sqrt(diag(pofit$vbeta)))^2,
    PVal  = 2-2*pnorm(abs(pofit$beta/sqrt(diag(pofit$vbeta))))),d)
}

posummExp<-function(pofit,d=6){
```

```
round(cbind(  
  est      = pofit$beta,  
  SD       = sqrt(diag(pofit$vbeta)),  
  exp.est  = exp(pofit$beta),  
  exp.lo.ci = exp(pofit$beta-1.96*sqrt(diag(pofit$vbeta))),  
  exp.up.ci = exp(pofit$beta+1.96*sqrt(diag(pofit$vbeta))),  
  PVal     = 2-2*pnorm(abs(pofit$beta/sqrt(diag(pofit$vbeta))))),d)  
)
```

## Data descriptions

`pbc3.csv`

*The PBC3 trial in liver cirrhosis*

Variable	Description
id	patient id
unit	hospital
days	follow-up time in days (time since randomisation)
status	0 = censoring, 1 = transplantation, 2 = death without transplantation
tment	0 = placebo, 1 = CyA
sex	0 = female, 1 = male
age	age (years)
bili	bilirubin (micromoles/L)
alb	albumin (g/L)
stage	disease stage: 2 = I-II, 3 = III, 4 = IV

## `cphholter.csv`

### *The Copenhagen Holter study*

Variable name	Description
id	patient id
timedeath	follow-up time (days)
death	0 = alive, 1 = dead
timeafib	time to atrial fibrillation (days); missing if afib = 0
afib	0 = no atrial fibrillation, 1 = atrial fibrillation
timestroke	time to stroke (days); missing if stroke = 0
stroke	0 = no stroke, 1 = stroke
sex	0 = female, 1 = male
age	age (years)
smoker	current smoker: 0 = no, 1 = yes
esvea	excessive supra-ventricular ectopic activity: 0 = no, 1 = yes
chol	cholesterol (mmol/L)
diabet	diabets mellitus: 0 = no, 1 = yes
bmi	body mass index (kg/m <sup>2</sup> )
aspirin	aspirin use: 0 = no, 1 = yes
probnp	NT-proBNP (pmol/L)
sbp	systolic blood pressure (mmHg)

## **affektive.csv**

### *Recurrent episodes in affective disorders*

NB: All patients start in state 1 (in hospital).

Variable name	Description
id	patient id
episode	number of affective episodes
state	Status at time start: 0 = no current affective episode, 1 = current affective episode
start	start time in state (months)
stop	last time seen in state (months)
status	status at time stop: 0 = transition to state 0 1 = transition to state 1 2 = transition to death 3 = censoring
prev	'start' of time to next transition to state 1, even if in state 1
bip	0 = unipolar, 1 = bipolar
sex	0 = female, 1 = male
age	age (years)
year	year of initial episode

## Day 1

### Ex 1

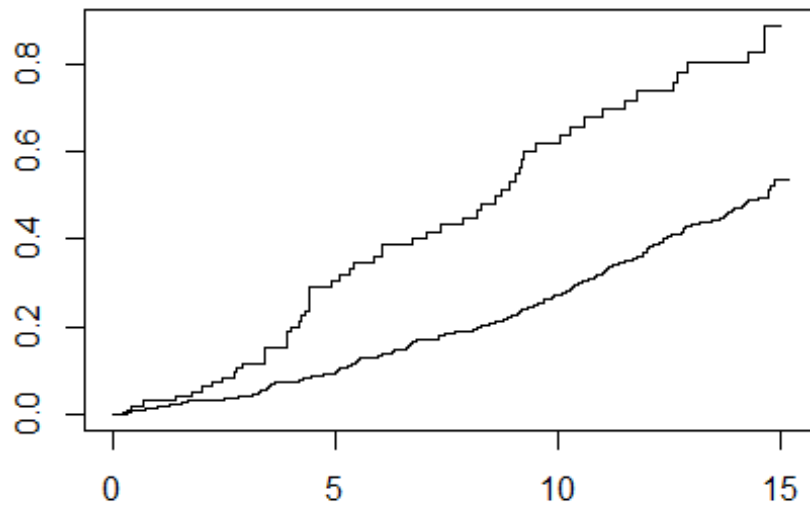
Consider the Copenhagen Holter study and estimate non-parametrically the cumulative hazards for stroke-free survival for subjects with and without ESVEA. Compare the two using the logrank test.

```
# Note that the time variables, timestroke and timedead, are
# measured in days. We will first convert them to years for easier
# interpretations.
chs_data <- read.csv("cphholter.csv")
chs_data$timestroke <- chs_data$timestroke/365.25
chs_data$timedead <- chs_data$timedead/365.25

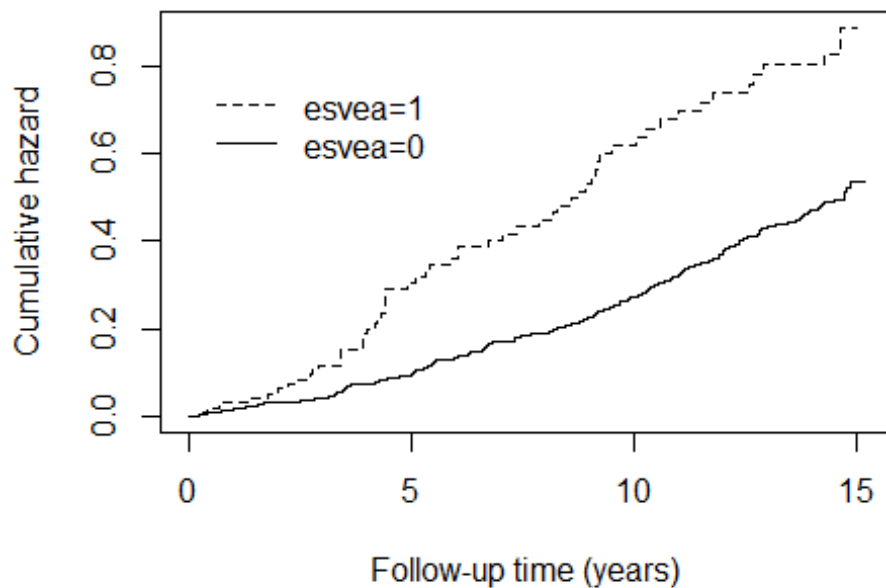
# Create composite end-point of stroke or death
chs_data$timestrokeordeath <- ifelse(chs_data$stroke == 1,
                                     chs_data$timestroke,
                                     chs_data$timedead)
chs_data$strokeordeath <- ifelse(chs_data$stroke == 1,
                                1,
                                chs_data$death)

# Cumulative hazards with or without ESVEA:
# Nelson-Aalen estimator
library(survival)
sfit <- survfit(Surv(timestrokeordeath,strokeordeath)~esvea,data=chs_data)
plot(sfit,fun="cumhaz")
```





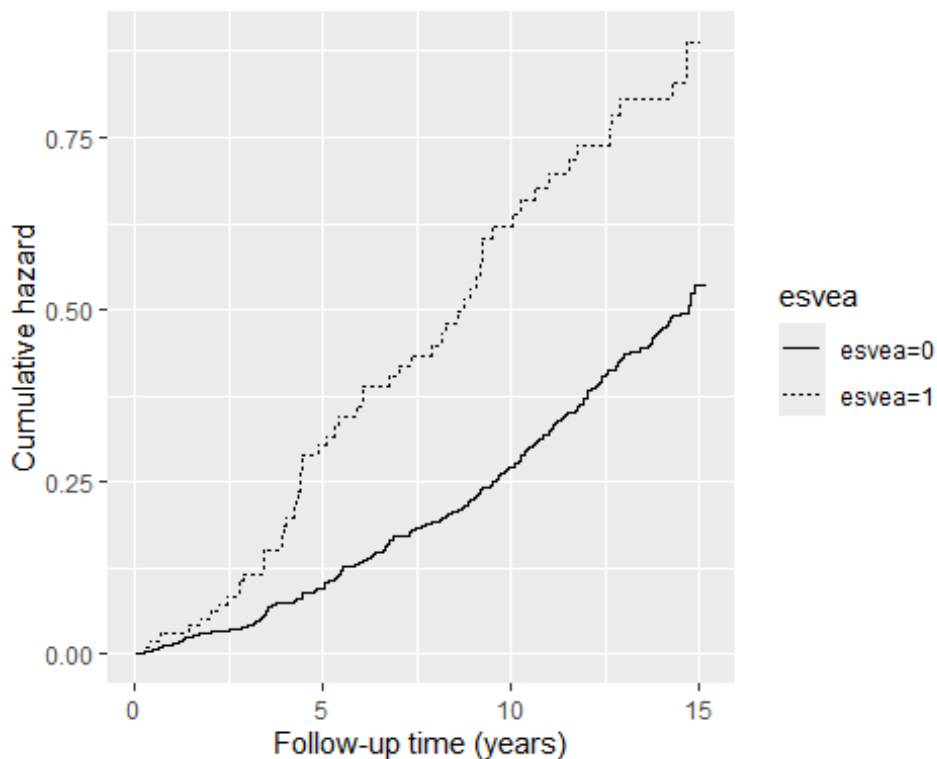
```
# Improving plot
plot(sfit, fun="cumhaz",
      lty=c(1,2),
      xlab="Follow-up time (years)",
      ylab="Cumulative hazard")
legend(0, .8, c("esvea=1", "esvea=0"), lty=c(2,1), bty='n')
```



```
# using ggplot to 'improve'
library(ggplot2)

# Need a data frame
naadata <- data.frame(
  time = sfit$time,
  cumhaz = sfit$cumhaz,
  esvea = c(rep(names(sfit$strata)[1], sfit$strata[1]),
            rep(names(sfit$strata)[2], sfit$strata[2])))

ggplot(data = naadata) +
  geom_step(aes(x = time, y = cumhaz, linetype = esvea)) +
  xlab("Follow-up time (years)") +
  ylab("Cumulative hazard")
```



```
# Logrank test
survdif(Surv(timestrokeordeath, strokeordeath) ~ esvea, data = chs_data)
```

Call:

```
survdif(formula = Surv(timestrokeordeath, strokeordeath) ~ esvea,
  data = chs_data)
```

	N	Observed	Expected	(O-E) <sup>2</sup> /E	(O-E) <sup>2</sup> /V
esvea=0	579	230	253.4	2.17	18.6
esvea=1	99	57	33.6	16.37	18.6

Chisq= 18.6 on 1 degrees of freedom, p= 2e-05

## Ex 2

Repeat the previous exercise, now looking instead at the competing end-points stroke and death without stroke.

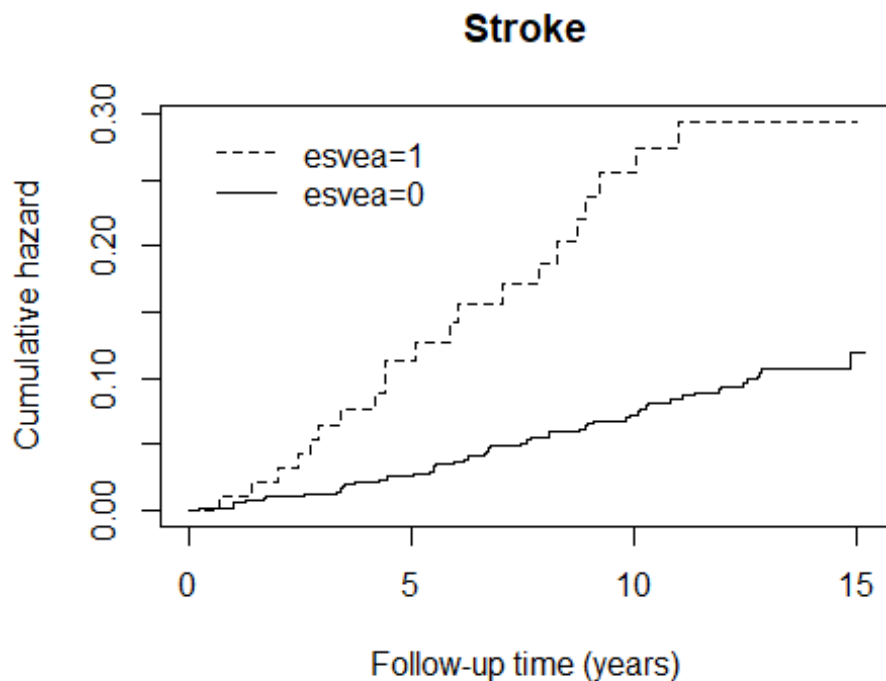
```
library(survival)

# Stroke - censor for death
# but we can use the define time variable timestrokeordeath
naa_stroke <- survfit(Surv(timestrokeordeath, stroke) ~ esvea, data = chs_data)
plot(naa_stroke, fun="cumhaz", main="Stroke",
  lty=c(1,2),
```

```

xlab="Follow-up time (years)",
ylab="Cumulative hazard")
legend(0, .3, c("esvea=1", "esvea=0"), lty=c(2,1), bty='n')

```



```

# Logrank test
survdif(Surv(timestrokeordeath, stroke) ~ esvea, data = chs_data)

Call:
survdif(formula = Surv(timestrokeordeath, stroke) ~ esvea, data = chs_data)

      N Observed Expected (O-E)^2/E (O-E)^2/V
esvea=0 579      52   64.19      2.32     19.2
esvea=1  99      21    8.81     16.87     19.2

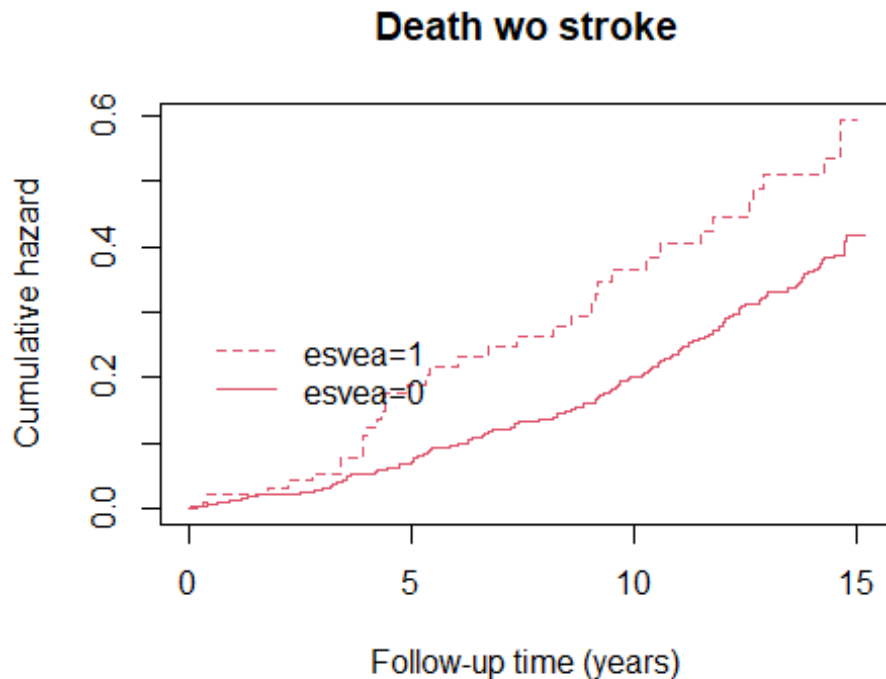
Chisq= 19.2 on 1 degrees of freedom, p= 1e-05

# Death without stroke: need to create indicator which censors at stroke
# but we can use the define time variable timestrokeordeath
chs_data$death_wo_stroke <- ifelse(chs_data$stroke == 1, 0, chs_data$death)

naa_dws <- survfit(Surv(timestrokeordeath, death_wo_stroke) ~ esvea, data =
chs_data)
plot(naa_dws, fun="cumhaz", main="Death wo stroke",
lty=c(1,2), col=c(2,2),
xlab="Follow-up time (years)",

```

```
ylab="Cumulative hazard")
legend(0, .3, c("esvea=1", "esvea=0"), lty=c(2,1), col=c(2,2), bty='n')
```



```
# Logrank test
survdif(Surv(timestrokeordeath, death_wo_stroke) ~ esvea, data = chs_data)
```

Call:

```
survdif(formula = Surv(timestrokeordeath, death_wo_stroke) ~
  esvea, data = chs_data)
```

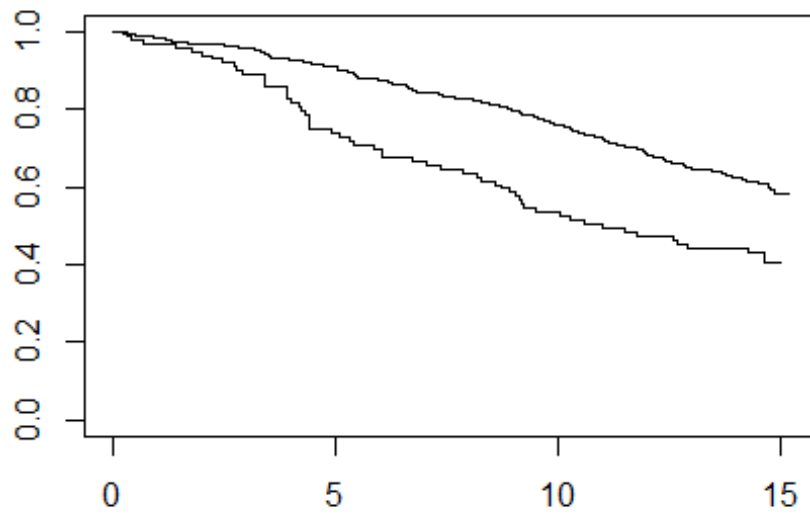
	N	Observed	Expected	$(O-E)^2/E$	$(O-E)^2/V$
esvea=0	579	178	189.3	0.669	5.8
esvea=1	99	36	24.7	5.115	5.8

Chisq= 5.8 on 1 degrees of freedom, p= 0.02

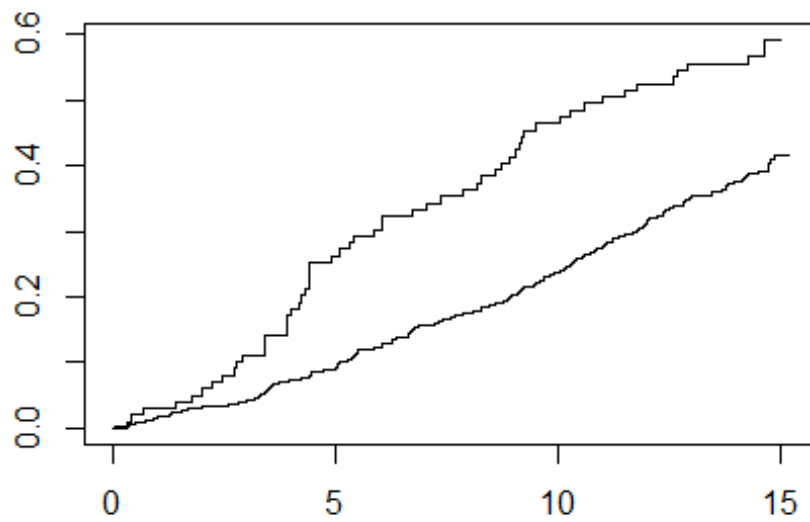
### Ex 3

Estimate non-parametrically the probabilities of stroke-free survival for subjects with and without ESVEA.

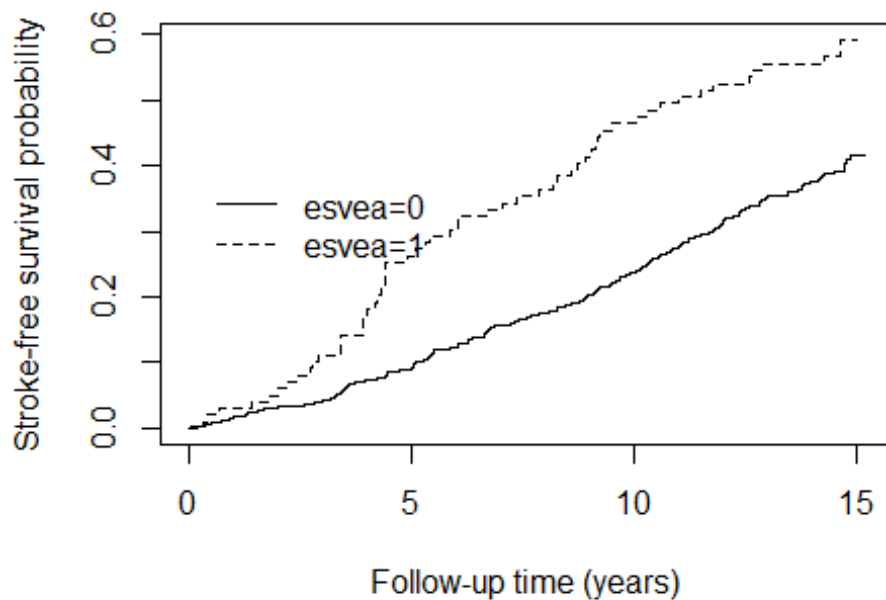
```
# Kaplan-Meier estimate of the survival functions
# We already have the fit from ex 1.1!
plot(sfit)
```



```
# or failure  
plot(sfit, fun="event")
```



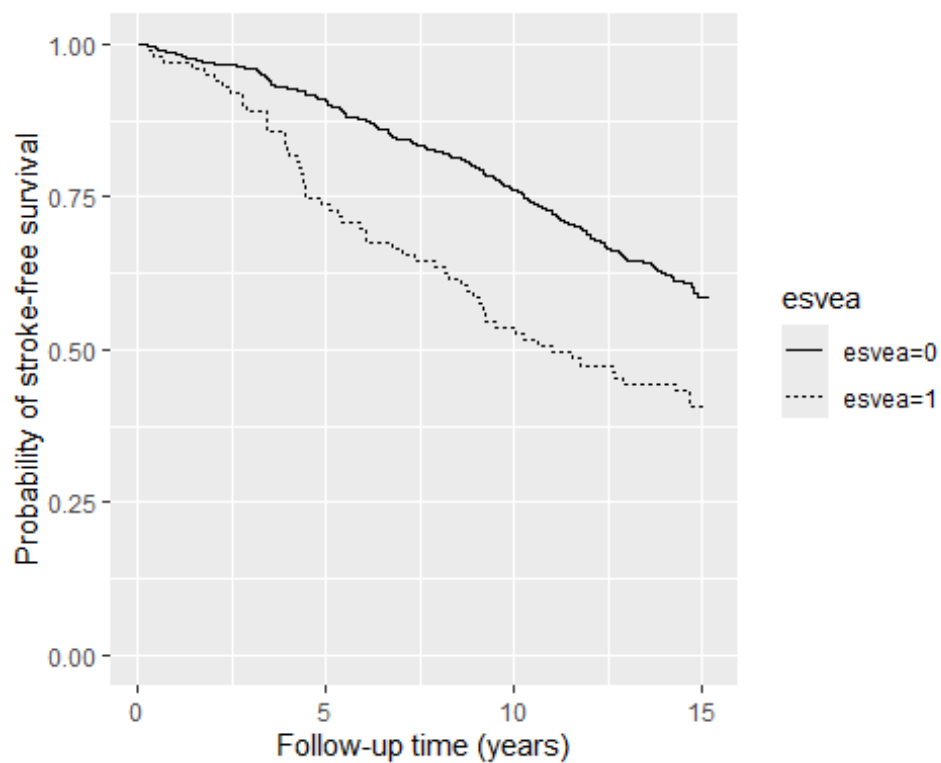
```
plot(sfit,fun="event",  
     lty=c(1,2),  
     xlab="Follow-up time (years)",  
     ylab="Stroke-free survival probability")  
legend(0, .4, c("esvea=0", "esvea=1"), lty=c(1,2), bty='n')
```



```
# using ggplot to 'improve'
kmdata <- data.frame(
  time = sfit$time,
  surv = sfit$surv,
  esvea = c(rep(names(sfit$strata)[1], sfit$strata[1]),
            rep(names(sfit$strata)[2], sfit$strata[2])))

# Plotting the Kaplan-Meier estimate
ggplot(data = kmdata) +
  geom_step(aes(x = time, y = surv, linetype = esvea)) +
  ylim(c(0,1)) +
  xlab("Follow-up time (years)") +
  ylab("Probability of stroke-free survival")
```





```
library(survminer)
```

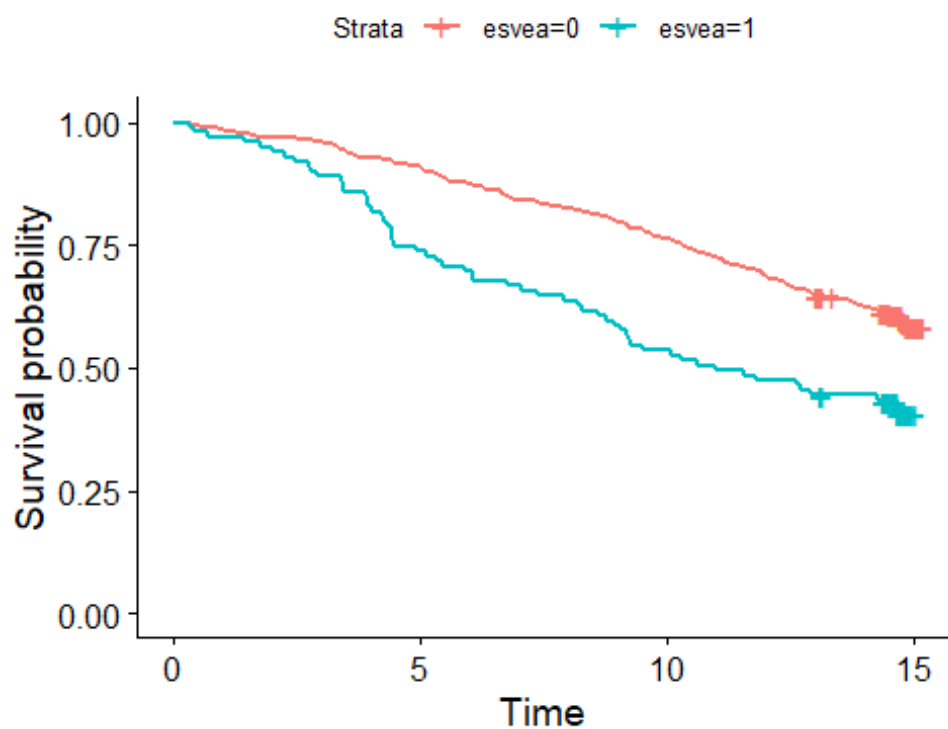
```
Loading required package: ggpubr
```

```
Attaching package: 'survminer'
```

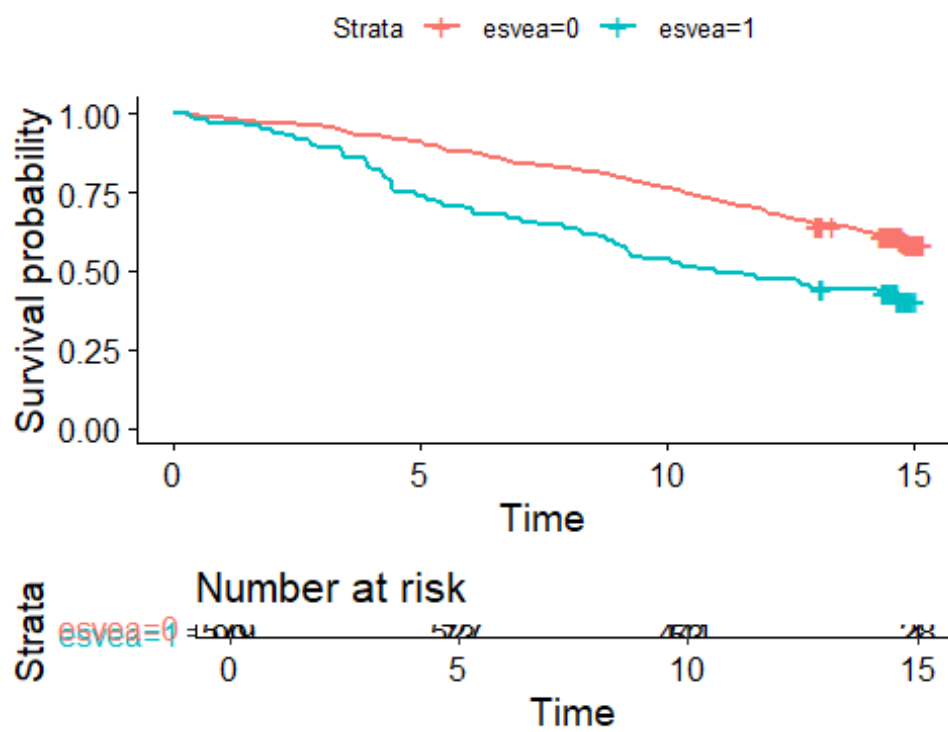
```
The following object is masked from 'package:survival':
```

```
myeloma
```

```
ggsurvplot(sfit)
```



```
ggsurvplot(sfit, risk.table = TRUE)
```



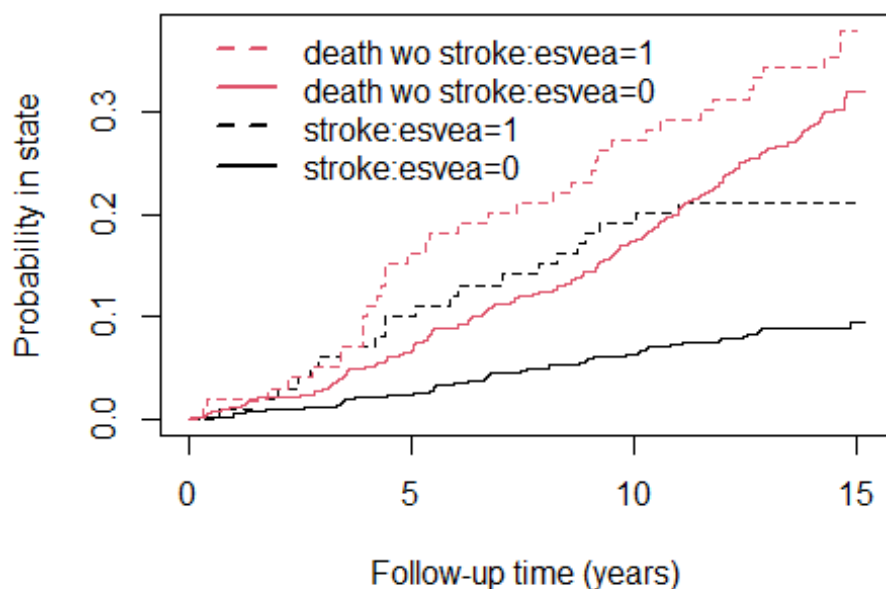
## Ex 4

Estimate non-parametrically the cumulative incidences of stroke and death without stroke for subjects with and without ESVEA.

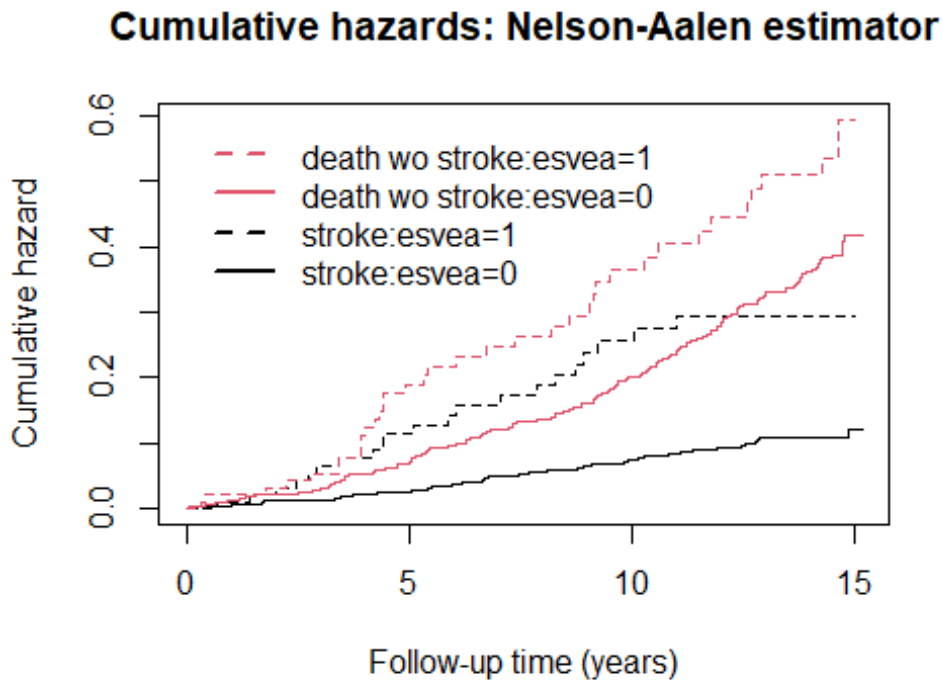
```
# Creating event variable, 0 = censored, 1 = stroke, 2 = death w/o stroke
chs_data$event <- with(chs_data,
  ifelse(death_wo_stroke == 0,
    stroke,
    death_wo_stroke*2)
)

# Survfit creates both competing risks
crfit <- survfit(Surv(timestrokeordeath, factor(event)) ~ esvea, data=chs_data)
plot(crfit,
  main="Cumulative incidences: Aalen-Johansens estimator",
  lty=c(1,2,1,2), col=c(1,1,2,2),
  xlab="Follow-up time (years)",
  ylab="Probability in state")
legend(0, .4,
  c("death wo stroke:esvea=1", "death wo stroke:esvea=0",
    "stroke:esvea=1", "stroke:esvea=0"),
  lty=c(2,1,2,1), col=c(2,2,1,1),
  lwd=2, bty='n')
```

### Cumulative incidences: Aalen-Johansens estimator

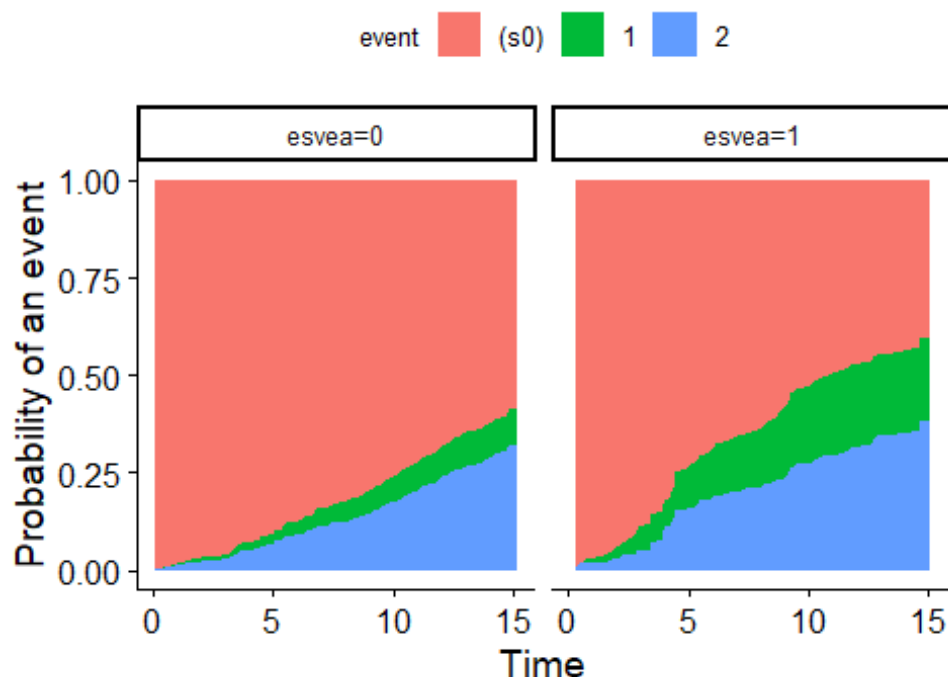


```
# Btw
# Nelson-Aalen, again
plot(crfit, fun="cumhaz",
     main="Cumulative hazards: Nelson-Aalen estimator",
     lty=c(1,2,1,2), col=c(1,1,2,2),
     xlab="Follow-up time (years)",
     ylab="Cumulative hazard")
legend(0, .6,
      c("death wo stroke:esvea=1", "death wo stroke:esvea=0",
        "stroke:esvea=1", "stroke:esvea=0"),
      lty=c(2,1,2,1), col=c(2,2,1,1),
      lwd=2, bty='n')
```



```
# Stacked plot
library(survminer)
ggcompetingrisks(crfit)
```

## Cumulative incidence functions



Estimate also the 10-year restricted mean stroke-free survival times and the expected number of years lost due to stroke or death without stroke for subjects with and without ESVEA.

```
print(sfit,rmean=10)
```

```
Call: survfit(formula = Surv(timestrokeordeath, strokeordeath) ~ esvea,
  data = chs_data)
```

	n	events	rmean*	se(rmean)	median	0.95LCL	0.95UCL
esvea=0	579	230	8.97	0.0949	NA	NA	NA
esvea=1	99	57	7.71	0.3051	11	8.93	NA

\* restricted mean with upper limit = 10

```
print(crfit,rmean=10)
```

```
Call: survfit(formula = Surv(timestrokeordeath, factor(event)) ~ esvea,
  data = chs_data)
```

	n	nevent	rmean	se(rmean)*
esvea=0, (s0)	579	0	8.9691353	0.09488854
esvea=1, (s0)	99	0	7.7107419	0.30513218
esvea=0, 1	579	52	0.2888054	0.05384486
esvea=1, 1	99	21	0.9491078	0.22774760
esvea=0, 2	579	178	0.7420592	0.08273372

```
esvea=1, 2      99      36 1.3401503 0.25872072  
  *restricted mean time in state (max time = 10 )
```

## Day 2

### Ex 1

Consider the data from the Copenhagen Holter study and the composite end-point stroke-free survival. Fit a Cox model and estimate the hazard ratio between subjects with or without ESVEA.

```
chs_data <- read.csv("cphholter.csv")
chs_data$timestroke <- chs_data$timestroke/365.25
chs_data$timedeath <- chs_data$timedeath/365.25

# Create composite end-point of stroke or death
chs_data$timestrokeordeath <- ifelse(chs_data$stroke == 1, chs_data$timestroke,
chs_data$timedeath)
chs_data$strokeordeath <- ifelse(chs_data$stroke == 1, 1, chs_data$death)

library(survival)
# Option method = "breslow" added as the ties-handling approach.
# This is the default in SAS and we have used that in the book
summary(coxph(Surv(timestrokeordeath,strokeordeath)
~ esvea, method = "breslow", data=chs_data))
```

Call:

```
coxph(formula = Surv(timestrokeordeath, strokeordeath) ~ esvea,
      data = chs_data, method = "breslow")
```

n= 678, number of events= 287

	coef	exp(coef)	se(coef)	z	Pr(> z )	
esvea	0.6285	1.8747	0.1482	4.241	2.23e-05	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
esvea	1.875	0.5334	1.402	2.507

Concordance= 0.546 (se = 0.012 )

Likelihood ratio test= 15.79 on 1 df, p=7e-05

Wald test = 17.98 on 1 df, p=2e-05

Score (logrank) test = 18.58 on 1 df, p=2e-05

### Ex 2

Fit a Cox model as before, now also adjusting for sex, age, and systolic blood pressure.

```
summary(coxph(Surv(timestrokeordeath,strokeordeath)
~ esvea + sex + age + sbp, method = "breslow", data=chs_data))
```

```
Call:
coxph(formula = Surv(timestrokeordeath, strokeordeath) ~ esvea +
      sex + age + sbp, data = chs_data, method = "breslow")

n= 675, number of events= 285
(3 observations deleted due to missingness)
```

	coef	exp(coef)	se(coef)	z	Pr(> z )	
esvea	0.318284	1.374767	0.152587	2.086	0.0370	*
sex	0.577585	1.781731	0.126946	4.550	5.37e-06	***
age	0.076658	1.079673	0.009362	8.189	2.64e-16	***
sbp	0.005152	1.005165	0.002438	2.113	0.0346	*

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	exp(coef)	exp(-coef)	lower .95	upper .95
esvea	1.375	0.7274	1.019	1.854
sex	1.782	0.5613	1.389	2.285
age	1.080	0.9262	1.060	1.100
sbp	1.005	0.9949	1.000	1.010

```
Concordance= 0.672 (se = 0.016 )
Likelihood ratio test= 99.45 on 4 df, p=<2e-16
Wald test = 104.1 on 4 df, p=<2e-16
Score (logrank) test = 110 on 4 df, p=<2e-16
```

### Ex 3

Consider the data from the Copenhagen Holter study and fit Cox models for the **cause-specific hazards** for the two outcomes (1) stroke and (2) death without stroke including ESVEA, sex, age, and systolic blood pressure. Compare to previous exercise.

```
# Stroke indicator variable already censors for death
summary(coxph(Surv(timestrokeordeath, stroke)
              ~ esvea + sex + age + sbp, method = "breslow", data = chs_data))
```

```
Call:
coxph(formula = Surv(timestrokeordeath, stroke) ~ esvea + sex +
      age + sbp, data = chs_data, method = "breslow")
```

```
n= 675, number of events= 72
(3 observations deleted due to missingness)
```

	coef	exp(coef)	se(coef)	z	Pr(> z )	
esvea	0.702407	2.018606	0.269968	2.602	0.00927	**
sex	0.491881	1.635389	0.248634	1.978	0.04789	*
age	0.078980	1.082183	0.019054	4.145	3.4e-05	***
sbp	0.011340	1.011404	0.004651	2.438	0.01477	*



```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
esvea      2.019      0.4954      1.189      3.426
sex         1.635      0.6115      1.005      2.662
age         1.082      0.9241      1.043      1.123
sbp         1.011      0.9887      1.002      1.021

Concordance= 0.728 (se = 0.028 )
Likelihood ratio test= 41.22 on 4 df,   p=2e-08
Wald test              = 43.26 on 4 df,   p=9e-09
Score (logrank) test = 47.2 on 4 df,   p=1e-09

# Death without stroke: need to create indicator which censors at stroke
chs_data$death_wo_stroke <- ifelse(chs_data$stroke == 1, 0, chs_data$death)
summary(coxph(Surv(timestrokeordeath,death_wo_stroke)
             ~ esvea + sex + age + sbp, method = "breslow", data=chs_data))

Call:
coxph(formula = Surv(timestrokeordeath, death_wo_stroke) ~ esvea +
      sex + age + sbp, data = chs_data, method = "breslow")

n= 675, number of events= 213
(3 observations deleted due to missingness)

      coef exp(coef) se(coef)      z Pr(>|z|)
esvea 0.160081  1.173605 0.186795 0.857   0.391
sex   0.605281  1.831766 0.147665 4.099 4.15e-05 ***
age   0.076073  1.079041 0.010759 7.071 1.54e-12 ***
sbp   0.002956  1.002960 0.002867 1.031   0.303
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
esvea      1.174      0.8521      0.8138      1.692
sex         1.832      0.5459      1.3714      2.447
age         1.079      0.9267      1.0565      1.102
sbp         1.003      0.9970      0.9973      1.009

Concordance= 0.657 (se = 0.019 )
Likelihood ratio test= 64.37 on 4 df,   p=3e-13
Wald test              = 67.44 on 4 df,   p=8e-14
Score (logrank) test = 70.93 on 4 df,   p=1e-14

```

## Ex 4

Reproduce the results in Table 2.14 (as many as time allows).

```

affective <- read.csv("affective.csv")
library(survival)
# Cox model for episode 1-4 separately, time since diagnosis
coxph(Surv(start, stop, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 1 & state == 0))

```

Call:

```

coxph(formula = Surv(start, stop, status == 1) ~ bip, data = subset(affective,
  episode == 1 & state == 0), method = "breslow")

```

	coef	exp(coef)	se(coef)	z	p
bip	0.3552	1.4264	0.2500	1.421	0.155

Likelihood ratio test=1.89 on 1 df, p=0.1692  
n= 116, number of events= 99

```

coxph(Surv(start, stop, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 2 & state == 0))

```

Call:

```

coxph(formula = Surv(start, stop, status == 1) ~ bip, data = subset(affective,
  episode == 2 & state == 0), method = "breslow")

```

	coef	exp(coef)	se(coef)	z	p
bip	0.1890	1.2080	0.2604	0.726	0.468

Likelihood ratio test=0.51 on 1 df, p=0.4751  
n= 91, number of events= 82

```

coxph(Surv(start, stop, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 3 & state == 0))

```

Call:

```

coxph(formula = Surv(start, stop, status == 1) ~ bip, data = subset(affective,
  episode == 3 & state == 0), method = "breslow")

```

	coef	exp(coef)	se(coef)	z	p
bip	-0.1175	0.8891	0.3005	-0.391	0.696

Likelihood ratio test=0.16 on 1 df, p=0.6936  
n= 74, number of events= 62

```

coxph(Surv(start, stop, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 4 & state == 0))

```

Call:

```

coxph(formula = Surv(start, stop, status == 1) ~ bip, data = subset(affective,
  episode == 4 & state == 0), method = "breslow")

```

	coef	exp(coef)	se(coef)	z	p
bip	1.1500	3.1581	0.3536	3.252	0.00114

Likelihood ratio test=9.93 on 1 df, p=0.001623

n= 56, number of events= 47

# AG model, time since diagnosis

```
coxph(Surv(start, stop, status == 1) ~ bip, method = "breslow",
      data = subset(affective, state == 0))
```

Call:

```
coxph(formula = Surv(start, stop, status == 1) ~ bip, data = subset(affective,
  state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.36593	1.44186	0.09448	3.873	0.000107

Likelihood ratio test=14.24 on 1 df, p=0.0001612

n= 626, number of events= 542

# PWP model, time since diagnosis

```
coxph(Surv(start, stop, status == 1)~strata(episode)+bip, method = "breslow",
      data = subset(affective, state == 0))
```

Call:

```
coxph(formula = Surv(start, stop, status == 1) ~ strata(episode) +
  bip, data = subset(affective, state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.2418	1.2736	0.1121	2.157	0.031

Likelihood ratio test=4.54 on 1 df, p=0.03312

n= 626, number of events= 542

# Cox model for episode 1-4 separately, gap time

```
affective$wait <- with(affective, stop - start)
coxph(Surv(wait, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 1 & state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ bip, data = subset(affective,
  episode == 1 & state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.3991	1.4905	0.2487	1.605	0.109

Likelihood ratio test=2.39 on 1 df, p=0.1222

n= 116, number of events= 99

```
coxph(Surv(wait, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 2 & state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ bip, data = subset(affective,
  episode == 2 & state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.2165	1.2418	0.2579	0.84	0.401

Likelihood ratio test=0.68 on 1 df, p=0.41

n= 91, number of events= 82

```
coxph(Surv(wait, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 3 & state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ bip, data = subset(affective,
  episode == 3 & state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	-0.1114	0.8946	0.2867	-0.389	0.698

Likelihood ratio test=0.15 on 1 df, p=0.6953

n= 74, number of events= 62

```
coxph(Surv(wait, status == 1) ~ bip, method = "breslow",
      data = subset(affective, episode == 4 & state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ bip, data = subset(affective,
  episode == 4 & state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.5964	1.8155	0.3183	1.874	0.061

Likelihood ratio test=3.31 on 1 df, p=0.06905

n= 56, number of events= 47

# AG model, gap time

```
coxph(Surv(wait, status == 1) ~ bip, method = "breslow",
      data = subset(affective, state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ bip, data = subset(affective,
  state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.12555	1.13377	0.09445	1.329	0.184

Likelihood ratio test=1.74 on 1 df, p=0.1875  
n= 626, number of events= 542

# PWP model, gap time

```
coxph(Surv(wait, status == 1) ~ strata(episode) + bip, method = "breslow",  
      data = subset(affective, state == 0))
```

Call:

```
coxph(formula = Surv(wait, status == 1) ~ strata(episode) + bip,  
      data = subset(affective, state == 0), method = "breslow")
```

	coef	exp(coef)	se(coef)	z	p
bip	0.02781	1.02820	0.10040	0.277	0.782

Likelihood ratio test=0.08 on 1 df, p=0.7821  
n= 626, number of events= 542

## Day 3

### Ex 1

Fit Fine-Gray models for the cumulative incidences of stroke and death without stroke adjusting for ESVEA, sex, age, and systolic blood pressure. Are the (adjusted) associations between ESVEA and the cumulative incidences statistically significant?

```
chs_data <- read.csv("cphholter.csv")
chs_data$timestroke <- chs_data$timestroke/365.25
chs_data$timedeath <- chs_data$timedeath/365.25

# Create composite end-point of stroke or death
chs_data$timestrokeordeath <- ifelse(chs_data$stroke == 1,
                                     chs_data$timestroke,
                                     chs_data$timedeath)
chs_data$strokeordeath <- ifelse(chs_data$stroke == 1, 1,
                                 chs_data$death)
# Death without stroke: need to create indicator
# which censors at stroke
chs_data$death_wo_stroke <- ifelse(chs_data$stroke == 1, 0,
                                    chs_data$death)

# We need a variable with one level for each possible outcome:
# event variable, 0 = censored, 1 = stroke, 2 = death w/o stroke
chs_data$event <- factor(
  with(chs_data, ifelse(death_wo_stroke == 0, stroke,
                        death_wo_stroke*2)))

# Stroke
fg_stroke_data<-finegray(Surv(timestrokeordeath,event) ~ .,
                        etype=1, data=chs_data)
summary(coxph(Surv(fgstart,fgstop,fgstatus)
              ~esvea + sex + age + sbp, weight=fgwt,
              data=fg_stroke_data))
```

Call:

```
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ esvea + sex +
      age + sbp, data = fg_stroke_data, weights = fgwt)
```

```
n= 888, number of events= 72
(4 observations deleted due to missingness)
```

	coef	exp(coef)	se(coef)	robust se	z	Pr(> z )	
esvea	0.593921	1.811075	0.271675	0.275526	2.156	0.031116	*
sex	0.379189	1.461099	0.248427	0.243020	1.560	0.118684	
age	0.063347	1.065397	0.019072	0.018469	3.430	0.000604	***

```
sbp    0.010629  1.010686 0.004608  0.004196 2.533 0.011305 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	exp(coef)	exp(-coef)	lower .95	upper .95
esvea	1.811	0.5522	1.0554	3.108
sex	1.461	0.6844	0.9074	2.353
age	1.065	0.9386	1.0275	1.105
sbp	1.011	0.9894	1.0024	1.019

```
Concordance= 0.699 (se = 0.029 )
Likelihood ratio test= 30.7 on 4 df, p=4e-06
Wald test = 37.72 on 4 df, p=1e-07
Score (logrank) test = 34.45 on 4 df, p=6e-07, Robust = 24.82 p=5e-05
```

(Note: the likelihood ratio and score tests assume independence of observations within a cluster, the Wald and robust score tests do not).

```
# Death wo stroke
fg_death_data <- finegray(Surv(timestrokeordeath,event) ~ .,
                           etype=2,
                           data=chs_data)
summary(coxph(Surv(fgstart,fgstop,fgstatus)
              ~ esvea + sex + age + sbp, weight=fgwt,
              data=fg_death_data))
```

```
Call:
coxph(formula = Surv(fgstart, fgstop, fgstatus) ~ esvea + sex +
      age + sbp, data = fg_death_data, weights = fgwt)
```

```
n= 1172, number of events= 213
(10 observations deleted due to missingness)
```

	coef	exp(coef)	se(coef)	robust se	z	Pr(> z )
esvea	-0.006269	0.993751	0.188059	0.193559	-0.032	0.974164
sex	0.530219	1.699304	0.148362	0.146047	3.630	0.000283 ***
age	0.066495	1.068756	0.010812	0.010673	6.230	4.65e-10 ***
sbp	0.001601	1.001602	0.002927	0.002917	0.549	0.583197

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	exp(coef)	exp(-coef)	lower .95	upper .95
esvea	0.9938	1.0063	0.6800	1.452
sex	1.6993	0.5885	1.2763	2.262
age	1.0688	0.9357	1.0466	1.091
sbp	1.0016	0.9984	0.9959	1.007

```
Concordance= 0.636 (se = 0.019 )
```

```
Likelihood ratio test= 46.38 on 4 df, p=2e-09
Wald test            = 50.97 on 4 df, p=2e-10
Score (logrank) test = 49.87 on 4 df, p=4e-10, Robust = 41 p=3e-08
```

(Note: the likelihood ratio and score tests assume independence of observations within a cluster, the Wald and robust score tests do not).

## Ex 2

Consider the data on recurrent episodes in affective disorder. Estimate non-parametrically the mean number of episodes,  $\mu(t)$ , in  $[0, t]$  for unipolar and bipolar patients, taking the mortality into account. Estimate, incorrectly, the same mean curves by treating death as censoring and compare with the correct curves.

```
affective <- read.csv("affective.csv")

library(survival)
library(mets)

Loading required package: timereg

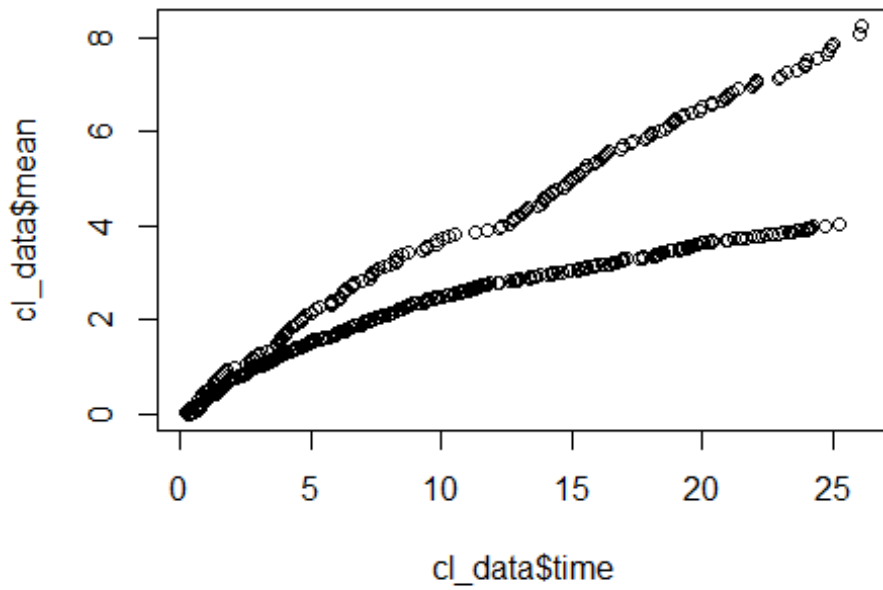
# status: 1=event, 2=death
affective$hosp<-ifelse(affective$status==1,1,0)
affective$death<-ifelse(affective$status==2,1,0)

xr0 <- phreg(Surv(prev,stop,hosp)~cluster(id),
             data=subset(affective,bip==0))
dr0 <- phreg(Surv(prev,stop,death)~cluster(id),
             data=subset(affective,bip==0))
xr1 <- phreg(Surv(prev,stop,hosp)~cluster(id),
             data=subset(affective,bip==1))
dr1 <- phreg(Surv(prev,stop,death)~cluster(id),
             data=subset(affective,bip==1))
out0 <- recurrentMarginal(xr0,dr0)
out1 <- recurrentMarginal(xr1,dr1)

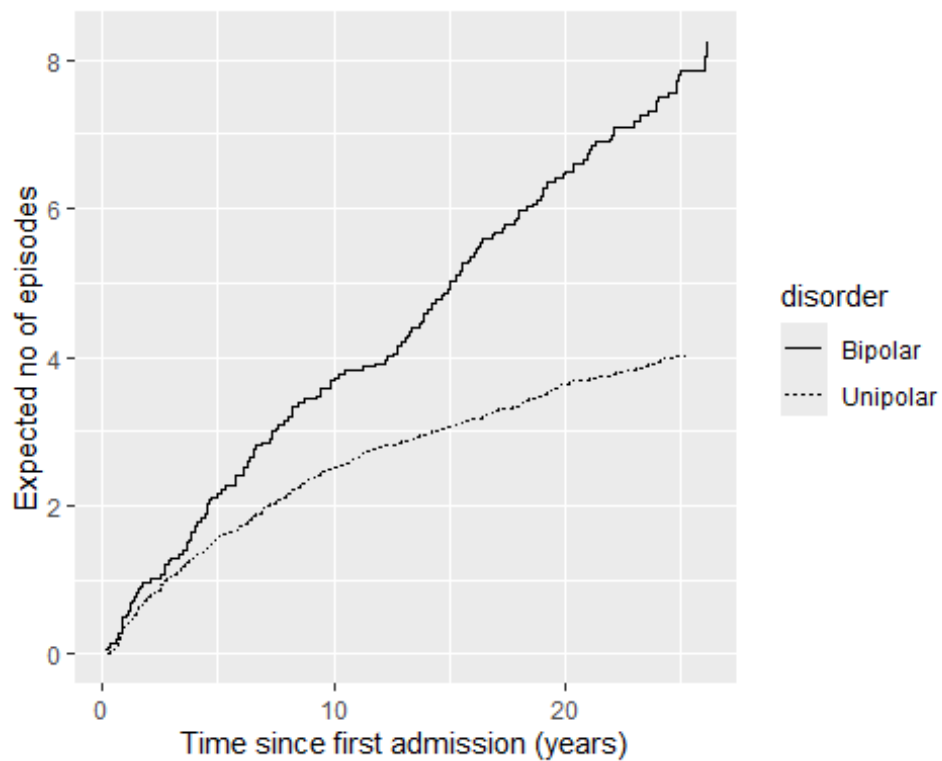
cl_data <- data.frame(
  time = c(out0$time,out1$time)/12,
  mean = c(out0$mu,out1$mu),
  disorder = c(rep("Unipolar",length(out0$time)),
               rep("Bipolar",length(out1$time))),
  type="CL")

plot(cl_data$time, cl_data$mean)
```





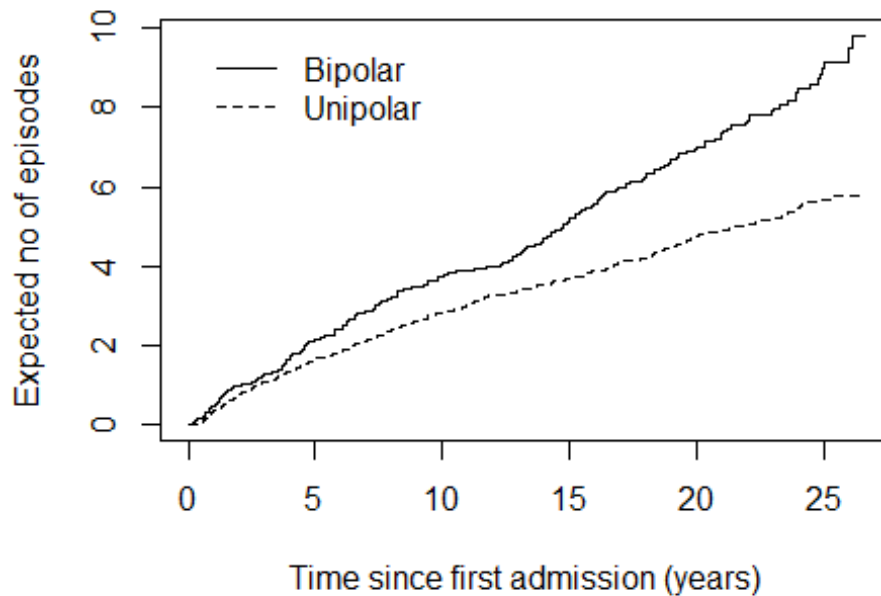
```
library(ggplot2)
ggplot(data = cl_data) +
  geom_step(aes(x = time, y = mean, linetype = disorder)) +
  xlab("Time since first admission (years)") +
  ylab("Expected no of episodes")
```



```
naafit <- survfit(Surv(prev/12, stop/12, status == 1) ~ bip,
                  data = affective)

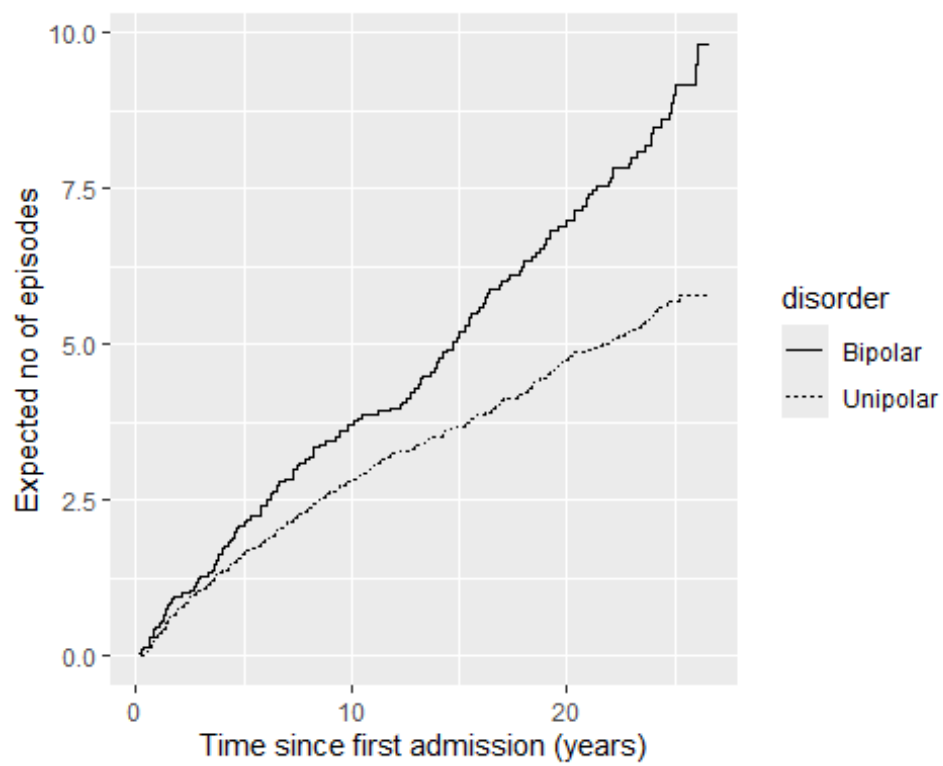
plot(naafit, fun="cumhaz",
     main="Nelson-Aalen estimator overestimates",
     lty=c(2,1),
     xlab="Time since first admission (years)",
     ylab="Expected no of episodes")
legend(0, 10, c("Bipolar", "Unipolar"), lty=c(1,2), bty='n')
```

## Nelson-Aalen estimator overestimates

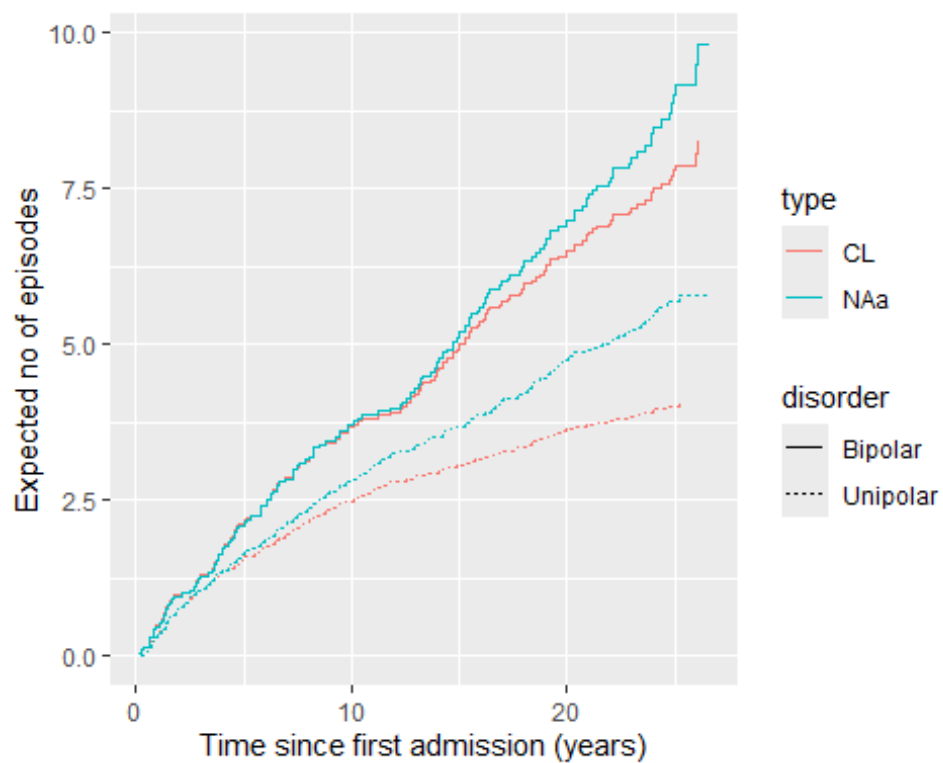


```
# Collecting the data for plotting with ggplot
naa_data <- data.frame(
  time = naafit$time,
  mean = naafit$cumhaz,
  disorder = c(rep("Unipolar", naafit$strata[[1]]),
               rep("Bipolar" , naafit$strata[[2]])),
  type="NAa")

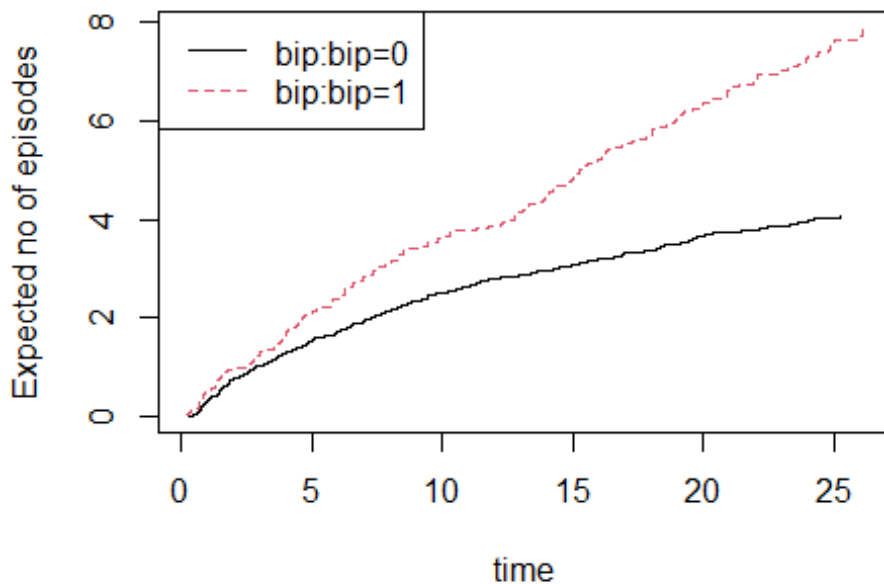
ggplot(data = naa_data) +
  geom_step(aes(x = time, y = mean, linetype = disorder)) +
  xlab("Time since first admission (years)") +
  ylab("Expected no of episodes")
```



```
# Comparing
ggplot(data = rbind(cl_data, naa_data)) +
  geom_step(aes(x = time, y = mean,
                linetype = disorder, color = type)) +
  xlab("Time since first admission (years)") +
  ylab("Expected no of episodes")
```



```
### updated mets:
xr <- phreg(Surv(prev/12,stop/12,hosp)
  ~ strata(bip) + cluster(id), data=affectedive)
dr <- phreg(Surv(prev/12,stop/12,death)
  ~ strata(bip) + cluster(id), data=affectedive)
out <- recurrentMarginal(xr,dr)
bplot(out,ylab="Expected no of episodes",col=1:2)
```



### Ex 3

Continuing the previous exercise, fit a Ghosh-Lin model for the expected number of episodes,  $\mu(t)$ , in  $[0, t]$  taking the mortality into account and adjusting for initial diagnosis (bipolar vs. unipolar) and calendar year of diagnosis. Fit a LWYY model for the same expectations by treating death as censoring and compare with the correct analysis.

```
library(mets)
# Ghosh-Lin
table(affective$year)

  59  60  61  62  63
365 303 323 152 144

subaff<- subset(affective,!is.na(prev))
recreg(Event(prev, stop, status)
       ~ bip + factor(year) + cluster(id),
       cause = 1, cens.code = 3, death.code = 2,
       data = subaff)
```

Call:  
 recreg(formula = Event(prev, stop, status) ~ bip + factor(year) +  
 cluster(id), data = subaff, cause = 1, death.code = 2, cens.code = 3)

```

n events
661    542

119 clusters
coeffients:
      Estimate      S.E.    dU^-1/2 P-value
bip      0.683507    0.194159  0.095201  0.0004
factor(year)59  0.349757  0.350467  0.154568  0.3183
factor(year)60  0.291375  0.351911  0.159275  0.4077
factor(year)61  0.139938  0.320329  0.157839  0.6622
factor(year)62 -0.343467  0.363702  0.186840  0.3450

exp(coeffients):
      Estimate      2.5%  97.5%
bip      1.98081  1.35386  2.8981
factor(year)59  1.41872  0.71380  2.8198
factor(year)60  1.33827  0.67142  2.6674
factor(year)61  1.15020  0.61392  2.1550
factor(year)62  0.70931  0.34774  1.4468

# LWYY
summary(fit.LWYY <- coxph(Surv(prev, stop, status == 1)
      ~ bip + factor(year) + cluster(id),
      data = subaff, ties = "breslow"))

Call:
coxph(formula = Surv(prev, stop, status == 1) ~ bip + factor(year),
      data = subaff, ties = "breslow", cluster = id)

n= 661, number of events= 542

      coef exp(coef) se(coef) robust se      z Pr(>|z|)
bip      0.44411    1.55911  0.09563    0.18405  2.413  0.0158 *
factor(year)59  0.26666    1.30559  0.15551    0.33974  0.785  0.4325
factor(year)60  0.14986    1.16167  0.15987    0.36006  0.416  0.6773
factor(year)61  0.07438    1.07722  0.15816    0.31572  0.236  0.8138
factor(year)62 -0.38749    0.67876  0.18711    0.36209 -1.070  0.2846
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

      exp(coef) exp(-coef) lower .95 upper .95
bip      1.5591    0.6414    1.0870    2.236
factor(year)59  1.3056    0.7659    0.6708    2.541
factor(year)60  1.1617    0.8608    0.5736    2.353
factor(year)61  1.0772    0.9283    0.5802    2.000
factor(year)62  0.6788    1.4733    0.3338    1.380

Concordance= 0.576 (se = 0.022 )

```

Likelihood ratio test= 39.5 on 5 df,  $p=2e-07$   
Wald test = 13.26 on 5 df,  $p=0.02$   
Score (logrank) test = 40.02 on 5 df,  $p=1e-07$ , Robust = 11.08  $p=0.05$

(Note: the likelihood ratio and score tests assume independence of observations within a cluster, the Wald and robust score tests do not).



## Day 4

```
pbc3 <- read.csv("pbc3.csv")

pbc3$followup <- pbc3$days/365.25
pbc3$fail <- as.numeric(pbc3$status>0)

# The following packages are needed:
library(survival)
library(pseudo) # also loads package geepack
```

### Ex 1

Calculate the pseudo observations (POs) based on Kaplan-Meier at year 2 and year 3 (separately), and add these to the PBC3 data.

```
# Add survival POs and failure POs (1-PO) at t=2 and 3 years:
po2 <- pseudosurv(pbc3$followup, pbc3$fail, tmax = 2)
po3 <- pseudosurv(pbc3$followup, pbc3$fail, tmax = 3)

pbc3$po2<-as.vector(po2$pseudo)
pbc3$po3<-as.vector(po3$pseudo)

pbc3$epo2<-as.vector(1-po2$pseudo)
pbc3$epo3<-as.vector(1-po3$pseudo)
```

### Ex 2

Estimate, separately for year 2 and 3, the risk difference between the two treatments using POs and the 'identity' link function.

```
# Defining two summary functions for summarizing a geese fit
# (one without `exp(est)` one that does)

posumm<-function(pofit,d=6){
  round(cbind(
    Est = pofit$beta,
    SD = sqrt(diag(pofit$vbeta)),
    lo.ci = pofit$beta-1.96*sqrt(diag(pofit$vbeta)),
    up.ci = pofit$beta+1.96*sqrt(diag(pofit$vbeta)),
    Wald = (pofit$beta/sqrt(diag(pofit$vbeta)))^2,
    PVal = 2-2*pnorm(abs(pofit$beta/sqrt(diag(pofit$vbeta))))),d)
}

posummExp<-function(pofit,d=6){
  round(cbind(
    est = pofit$beta,
    SD = sqrt(diag(pofit$vbeta)),
    exp.est = exp(pofit$beta),
```

```

exp.lo.ci = exp(pofit$beta-1.96*sqrt(diag(pofit$vbeta))),
exp.up.ci = exp(pofit$beta+1.96*sqrt(diag(pofit$vbeta))),
PVal      = 2-2*pnorm(abs(pofit$beta/sqrt(diag(pofit$vbeta))))),d)
}

```

```

posumm(geese(epo2 ~ tment, data = pbc3,
            id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.168128	0.029897	0.109529	0.226726	31.624011	0.00000
tment	-0.013582	0.041682	-0.095278	0.068114	0.106185	0.74453

```

posumm(geese(epo3 ~ tment, data = pbc3,
            id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.249897	0.037810	0.175789	0.324006	43.68169	0.000000
tment	-0.020715	0.052479	-0.123573	0.082144	0.15581	0.693044

### Ex 3

Same as 2. while adjusting for alb and log2(bili).

```

posumm(geese(epo2 ~ tment + alb + log2(bili),
            data = subset(pbc3, !is.na(alb)),
            id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.053060	0.167305	-0.274858	0.380977	0.100580	0.751135
tment	-0.049809	0.036125	-0.120613	0.020995	1.901108	0.167954
alb	-0.010199	0.003281	-0.016630	-0.003769	9.664663	0.001878
log2(bili)	0.113757	0.016050	0.082300	0.145214	50.237704	0.000000

```

posumm(geese(epo3 ~ tment + alb + log2(bili),
            data = subset(pbc3, !is.na(alb)),
            id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.165786	0.256215	-0.336394	0.667967	0.418687	0.517593
tment	-0.065200	0.048826	-0.160900	0.030500	1.783148	0.181763
alb	-0.012608	0.005130	-0.022663	-0.002553	6.039716	0.013988
log2(bili)	0.128303	0.018275	0.092484	0.164122	49.288959	0.000000

### Ex 4

Repeat 2. and 3. using the 'log' link function, i.e., targeting the risk ratio.

```

posummExp(geese(epo2 ~ tment, data = pbc3,
            id = id, mean.link = "log"))

```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.783032	0.177824	0.168128	0.118651	0.238235	0.000000
tment	-0.084236	0.258724	0.919214	0.553587	1.526325	0.744739

```
posummExp(geese(epo3 ~ tment, data = pbc3,
  id = id, mean.link = "log"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.386706	0.151304	0.249897	0.185767	0.336166	0.000000
tment	-0.086532	0.219335	0.917107	0.596648	1.409682	0.693198

```
posummExp(geese(epo2 ~ tment + alb + log2(bili),
  data = subset(pbc3, !is.na(alb)),
  id = id, mean.link = "log"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.894021	0.713575	0.150466	0.037156	0.609318	0.007948
tment	-0.493523	0.189439	0.610472	0.421126	0.884951	0.009182
alb	-0.064764	0.015694	0.937289	0.908897	0.966569	0.000037
log2(bili)	0.512934	0.061160	1.670185	1.481509	1.882889	0.000000

```
posummExp(geese(epo3 ~ tment + alb + log2(bili),
  data = subset(pbc3, !is.na(alb)),
  id = id, mean.link = "log"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.223108	0.621253	0.294314	0.087094	0.994563	0.048979
tment	-0.378839	0.152266	0.684656	0.507997	0.922749	0.012846
alb	-0.056832	0.013415	0.944753	0.920236	0.969923	0.000023
log2(bili)	0.405878	0.046036	1.500620	1.371147	1.642318	0.000000

## Ex 5

Repeat 2. and 3. using the 'cloglog' link function, i.e., targeting the hazard ratio.

```
posummExp(geese(epo2 ~ tment, data = pbc3,
  id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.692406	0.195243	0.184076	0.125546	0.269893	0.000000
tment	-0.092097	0.282826	0.912017	0.523910	1.587630	0.744704

```
posummExp(geese(epo3 ~ tment, data = pbc3,
  id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.246376	0.175301	0.287545	0.203932	0.405439	0.000000
tment	-0.099531	0.252245	0.905262	0.552152	1.484189	0.693151

```
posummExp(geese(eps2 ~ tment + alb + log2(bili),
               data = subset(pbc3, !is.na(alb)),
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-2.049934	1.285382	0.128743	0.010365	1.599059	0.110756
tment	-0.717637	0.359783	0.487904	0.241035	0.987615	0.046082
alb	-0.098561	0.032454	0.906141	0.850297	0.965652	0.002390
log2(bili)	0.788586	0.132718	2.200284	1.696315	2.853980	0.000000

```
posummExp(geese(eps3 ~ tment + alb + log2(bili),
               data = subset(pbc3, !is.na(alb)),
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-0.716708	1.207647	0.488357	0.045790	5.208443	0.552864
tment	-0.584313	0.315704	0.557489	0.300264	1.035068	0.064195
alb	-0.089456	0.028021	0.914428	0.865562	0.966054	0.001411
log2(bili)	0.588922	0.096463	1.802044	1.491607	2.177091	0.000000

## Ex 6

Calculate the POs at year 1, 2, 3, and 4 in 'one go' and create a data set of long format and estimate a joint model using the 'cloglog' link function and tment as the only covariate.

```
potsurv <- pseudosurv(pbc3$followup, pbc3$fail, tmax = 1:4)
longpbc3 <- NULL
for(it in 1:length(potsurv$time)){
  longpbc3 <- rbind(longpbc3,
                    cbind(pbc3,
                          pseudo = 1-potsurv$pseudo[,it],
                          tpseudo = potsurv$time[it],
                          id = 1:nrow(pbc3)))
}
longpbc3 <- longpbc3[order(longpbc3$id),]

# Now for the joint model
posummExp(geese(pseudo~as.factor(tpseudo)+tment, id=id,
               data=longpbc3, mean.link="cloglog", corstr="independence"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-0.752126	0.165353	0.471363	0.340882	0.651789	0.000005
as.factor(tpseudo)1	-1.740793	0.201299	0.175381	0.118204	0.260215	0.000000
as.factor(tpseudo)2	-0.958403	0.141929	0.383505	0.290374	0.506505	0.000000
as.factor(tpseudo)3	-0.516039	0.111410	0.596880	0.479793	0.742542	0.000004
tment	-0.054187	0.219043	0.947255	0.616614	1.455192	0.804614

```
# Cox model for comparison
summary(coxph(Surv(followup, status!=0)~tment, data=pbc3))
```

Call:

```
coxph(formula = Surv(followup, status != 0) ~ tment, data = pbc3)
```

n= 349, number of events= 90

	coef	exp(coef)	se(coef)	z	Pr(> z )
tment	-0.05874	0.94295	0.21092	-0.278	0.781

	exp(coef)	exp(-coef)	lower .95	upper .95
tment	0.943	1.06	0.6237	1.426

Concordance= 0.517 (se = 0.029 )

Likelihood ratio test= 0.08 on 1 df, p=0.8

Wald test = 0.08 on 1 df, p=0.8

Score (logrank) test = 0.08 on 1 df, p=0.8

## Ex 7

Repeat 6 but now adjusted for alb and log2(bili).

```
posummExp(geese(pseudo~as.factor(tppseudo) + tment + alb + log2(bili), id=id,  
               data=subset(longpbc3, !is.na(alb)),  
               mean.link="cloglog", corstr="independence"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.024057	1.046447	0.359135	0.046185	2.792614	0.327775
as.factor(tppseudo)1	-2.138191	0.288764	0.117868	0.066926	0.207585	0.000000
as.factor(tppseudo)2	-1.182857	0.201594	0.306402	0.206391	0.454874	0.000000
as.factor(tppseudo)3	-0.744680	0.171730	0.474886	0.339164	0.664920	0.000014
tment	-0.486139	0.257217	0.614997	0.371471	1.018170	0.058759
alb	-0.075213	0.023167	0.927546	0.886371	0.970633	0.001168
log2(bili)	0.672780	0.086925	1.959678	1.652695	2.323681	0.000000

# Cox model for comparison

```
summary(coxph(Surv(followup,status!=0)~ tment + alb + log2(bili), data=pbc3))
```

Call:

```
coxph(formula = Surv(followup, status != 0) ~ tment + alb + log2(bili),  
      data = pbc3)
```

n= 343, number of events= 88

(6 observations deleted due to missingness)

	coef	exp(coef)	se(coef)	z	Pr(> z )
tment	-0.57434	0.56308	0.22447	-2.559	0.0105 *
alb	-0.09089	0.91312	0.02164	-4.199	2.68e-05 ***
log2(bili)	0.66510	1.94469	0.07442	8.937	< 2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	exp(coef)	exp(-coef)	lower .95	upper .95
tment	0.5631	1.7760	0.3627	0.8742
alb	0.9131	1.0951	0.8752	0.9527
log2(bili)	1.9447	0.5142	1.6807	2.2501

Concordance= 0.819 (se = 0.022 )

Likelihood ratio test= 120 on 3 df, p=<2e-16

Wald test = 110.5 on 3 df, p=<2e-16

Score (logrank) test = 133.8 on 3 df, p=<2e-16

## Day 5

### Ex 1

Estimate, separately for year 3 and 4, the RMST difference between the two treatments using POs and the 'identity' link function.

```
pb3$rmst3<-pseudomean(pbc3$followup, pbc3$fail, tmax = 3)
pb3$rmst4<-pseudomean(pbc3$followup, pbc3$fail, tmax = 4)

posumm(geese(rmst3 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	2.605424	0.063887	2.480205	2.730642	1663.149903	0.000000
tment	0.071596	0.085036	-0.095074	0.238266	0.708881	0.399816

```
posumm(geese(rmst4 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	3.297363	0.096715	3.107801	3.486925	1162.368669	0.000000
tment	0.075160	0.130654	-0.180921	0.331242	0.330927	0.565114

### Ex 2

Same, while adjusting for 'alb' and 'log2(bili)'.

```
posumm(geese(rmst3 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	2.825598	0.346121	2.147201	3.503995	66.644608	0.000000
tment	0.147813	0.072936	0.004858	0.290769	4.107123	0.042703
alb	0.022512	0.006811	0.009162	0.035862	10.924256	0.000949
log2(bili)	-0.243093	0.031982	-0.305778	-0.180408	57.773091	0.000000

```
posumm(geese(rmst4 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	3.818862	0.547757	2.745258	4.892466	48.606238	0.000000
tment	0.196950	0.111177	-0.020958	0.414858	3.138187	0.076479
alb	0.033267	0.010709	0.012277	0.054256	9.650113	0.001893
log2(bili)	-0.403272	0.044917	-0.491308	-0.315235	80.608697	0.000000

### Ex 3

Calculate the POs based on Aalen-Johansen for both event types at year 2 and year 3 (separately) and add to the PBC3 data.

```
cipo2 <- pseudoci(pbc3$followup, pbc3$status, tmax = 2)
cipo3 <- pseudoci(pbc3$followup, pbc3$status, tmax = 3)

pbc3$trans.po2 <- as.vector(cipo2$pseudo[[1]])
pbc3$death.po2 <- as.vector(cipo2$pseudo[[2]])

pbc3$trans.po3 <- as.vector(cipo3$pseudo[[1]])
pbc3$death.po3 <- as.vector(cipo3$pseudo[[2]])
```

### Ex 4

Estimate, separately for year 2 and 3, the risk difference between the two treatments using POs for transplantation. Use the 'identity' link function.

```
posumm(geese(trans.po2 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.070125	0.020601	0.029748	0.110502	11.587463	0.000664
tment	-0.023905	0.026712	-0.076261	0.028451	0.800848	0.370840

```
posumm(geese(trans.po3 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.078677	0.022228	0.035111	0.122244	12.528588	0.000401
tment	-0.014037	0.030609	-0.074030	0.045956	0.210316	0.646520

### Ex 5

Same as 4. now adjusting for 'alb' and 'log2(bili)'.

```
posumm(geese(trans.po2 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))
```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	-0.178474	0.114327	-0.402554	0.045606	2.437004	0.118502
tment	-0.028900	0.024515	-0.076949	0.019150	1.389699	0.238456
alb	0.000121	0.002174	-0.004140	0.004382	0.003109	0.955532
log2(bili)	0.053038	0.013827	0.025937	0.080138	14.713819	0.000125

```
posumm(geese(trans.po3 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))
```



	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	-0.193465	0.130058	-0.448379	0.061449	2.212727	0.136877
tment	-0.020205	0.028480	-0.076026	0.035617	0.503289	0.478058
alb	0.000064	0.002489	-0.004815	0.004943	0.000655	0.979587
log2(bili)	0.058787	0.014397	0.030569	0.087006	16.672809	0.000044

## Ex 6

Repeat 4. and 5. using the 'cloglog' link function.

```
posummExp(geese(trans.po2 ~ tment, data = pbc3,
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-2.621343	0.304712	0.072705	0.040012	0.132112	0.000000
tment	-0.429430	0.484557	0.650880	0.251790	1.682535	0.375492

```
posummExp(geese(trans.po3 ~ tment, data = pbc3,
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-2.501708	0.294418	0.081945	0.046016	0.145927	0.000000
tment	-0.203986	0.447241	0.815474	0.339399	1.959339	0.648319

```
posummExp(geese(trans.po2 ~ tment + alb + log2(bili),
               data = subset(pbc3, !is.na(alb)),
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-6.629307	2.389897	0.001321	0.000012	0.142972	0.005539
tment	-0.873859	0.573256	0.417338	0.135682	1.283671	0.127415
alb	-0.005764	0.054623	0.994253	0.893306	1.106606	0.915961
log2(bili)	0.790978	0.215573	2.205553	1.445500	3.365248	0.000243

```
posummExp(geese(trans.po3 ~ tment + alb + log2(bili),
               data = subset(pbc3, !is.na(alb)),
               id = id, mean.link = "cloglog"))
```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-5.749723	2.085688	0.003184	0.000053	0.189802	0.005838
tment	-0.623817	0.509801	0.535895	0.197301	1.455563	0.221085
alb	-0.000347	0.046248	0.999653	0.913024	1.094502	0.994012
log2(bili)	0.648014	0.170812	1.911740	1.367826	2.671941	0.000148

## Ex 7

Calculate the POs at year 1, 2, 3, and 4. Create a data set with long format and estimate a joint model using the 'cloglog' link function and 'tment' as the only covariate.

```

pot <- pseudoci(pbc3$followup, pbc3$status, tmax = 1:4)
longpbc3 <- NULL
for(it in 1:length(pot$time)){
  longpbc3 <- rbind(longpbc3,
    cbind(pbc3, trans.po = pot$pseudo[[1]][,it],
          pbc3, death.po = pot$pseudo[[2]][,it],
          tpseudo = pot$time[it],
          id = 1:nrow(pbc3)))
}
longpbc3 <- longpbc3[order(longpbc3$id),]

# Now for the joint model:
posummExp(geese(trans.po ~ as.factor(tpseudo)+tment, id=id, data=longpbc3,
  mean.link="cloglog", corstr="independence"))

```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-1.939509	0.272918	0.143775	0.084211	0.245468	0.000000
as.factor(tpseudo)1	-1.874043	0.377045	0.153502	0.073310	0.321411	0.000001
as.factor(tpseudo)2	-0.831661	0.224073	0.435326	0.280595	0.675382	0.000206
as.factor(tpseudo)3	-0.619153	0.198090	0.538400	0.365164	0.793822	0.001774
tment	-0.078428	0.376746	0.924569	0.441821	1.934782	0.835095

## Ex 8

Same as 7. now adjusting for 'alb' and 'log2(bili)'.

```

posummExp(geese(trans.po ~ as.factor(tpseudo) + tment + alb + log2(bili), id=id,
  data=subset(longpbc3, !is.na(alb)),
  mean.link="cloglog", corstr="independence"))

```

	est	SD	exp.est	exp.lo.ci	exp.up.ci	PVal
(Intercept)	-4.265182	1.776156	0.014049	0.000432	0.456614	0.016334
as.factor(tpseudo)1	-2.077288	0.505531	0.125269	0.046508	0.337413	0.000040
as.factor(tpseudo)2	-0.739156	0.222242	0.477517	0.308896	0.738184	0.000881
as.factor(tpseudo)3	-0.613836	0.208422	0.541271	0.359751	0.814379	0.003228
tment	-0.248456	0.406118	0.780004	0.351886	1.728987	0.540682
alb	-0.012020	0.039560	0.988052	0.914336	1.067712	0.761257
log2(bili)	0.561370	0.132258	1.753072	1.352755	2.271855	0.000022

## Ex 9

Estimate the difference between treatments of years lost due to transplantation before year 3 and 4 (separately) using POs and the 'identity' link function.

```

# First calculate POs:
yl3 <- pseudoyl(pbc3$followup, pbc3$status, tmax = 3)
yl4 <- pseudoyl(pbc3$followup, pbc3$status, tmax = 4)
pbc3$trans.yl3 <- as.vector(yl3$pseudo[[1]])
pbc3$death.yl3 <- as.vector(yl3$pseudo[[2]])

```

```

pbc3$trans.yl4<-as.vector(yl4$pseudo[[1]])
pbc3$death.yl4<-as.vector(yl4$pseudo[[2]])

posumm(geese(trans.yl3 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.143046	0.041216	0.062262	0.223829	12.045356	0.000519
tment	-0.056386	0.051029	-0.156404	0.043632	1.220955	0.269173

```

posumm(geese(trans.yl4 ~ tment, data = pbc3,
             id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	0.240163	0.063211	0.116269	0.364057	14.435287	0.000145
tment	-0.051134	0.081499	-0.210872	0.108604	0.393658	0.530383

## Ex 10

Same adjusted for 'alb' and 'log2(bili)'.

```

posumm(geese(trans.yl3 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	-0.292596	0.214510	-0.713035	0.127844	1.860550	0.172561
tment	-0.062970	0.045824	-0.152786	0.026845	1.888339	0.169389
alb	-0.000748	0.004088	-0.008761	0.007265	0.033470	0.854839
log2(bili)	0.100151	0.026282	0.048639	0.151663	14.521475	0.000139

```

posumm(geese(trans.yl4 ~ tment + alb + log2(bili),
             data = subset(pbc3, !is.na(alb)),
             id = id, mean.link = "identity"))

```

	Est	SD	lo.ci	up.ci	Wald	PVal
(Intercept)	-0.610439	0.337125	-1.271203	0.050325	3.278715	0.070184
tment	-0.065538	0.073204	-0.209017	0.077942	0.801524	0.370638
alb	0.000268	0.006436	-0.012347	0.012883	0.001733	0.966798
log2(bili)	0.182172	0.040055	0.103663	0.260680	20.684436	0.000005