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

Kaplan-Meier estimator:

```
summary(km <- survfit(Surv(days,status!=0)~tment,data=pbc3))
```

Cox model:

```
summary(coxph(Surv(days,status!=0)~tment,data=pbc3))
```

Fine-Gray model:

```
pbc3$fstatus  <- factor(pbc3$status, 0:2, labels=c("cens", "trans", "death"))
pdata <- finegray(Surv(days,fstatus) ~ .,data=pbc3, 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

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

po2 <- pseudosurv(pbc3$followup, pbc3$fail,tmax = 2)
pbc3$po2<-as.vector(po2$pseudo)
```

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

```
potsurv <- pseudosurv(pbc3$followup, pbc3$fail,tmax = 1:3)
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),]
```

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

For multiple time points:

```
geese(pseudo~as.factor(tpseudo) + tment, id=id, data=longpbc3,
      mean.link="cloglog", corstr="independence"))
```

The following link functions are supported: `identit`, `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^2)
aspirin	aspirin use: 0 = no, 1 = yes
probnp	NT-proBNP (pmol/L)
sbp	systolic blood pressure (mmHg)

affective.csv

Recurrent episodes in affective disorders

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

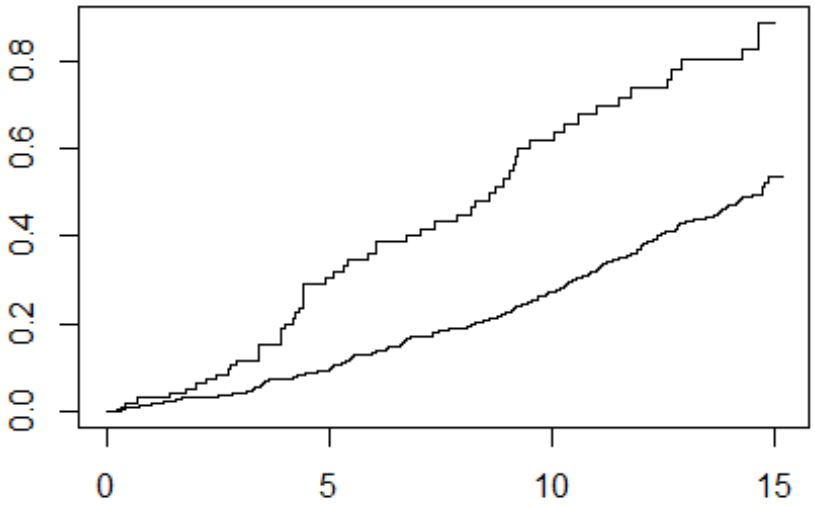
Variable name	Description
id	patient id
episode	number of affective episodes
state	Status at time start: 0 = no current affective episode, 1 = current affective episode
start	start time in state (months)
stop	last time seen in state (months)
status	status at time stop: 0 = transition to state 0 1 = transition 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

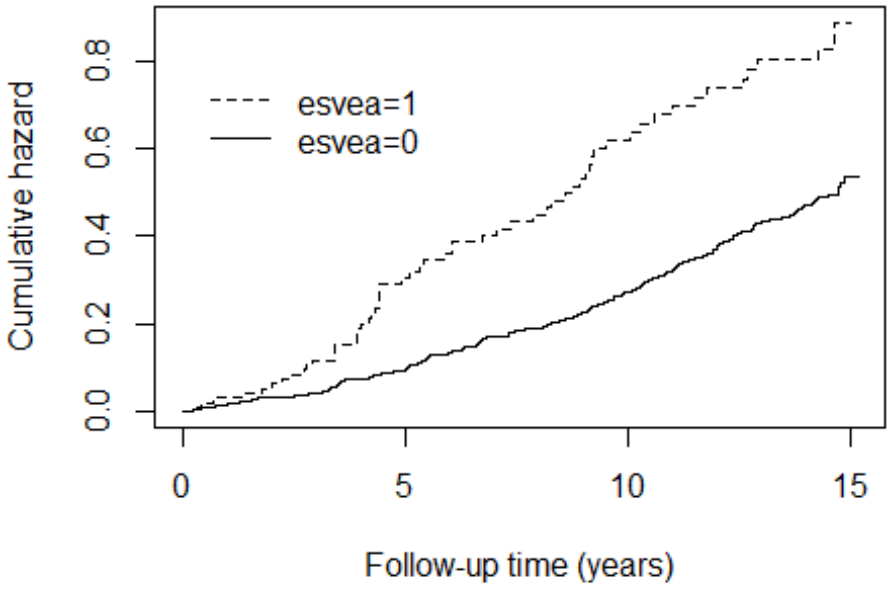
Ex 1

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.

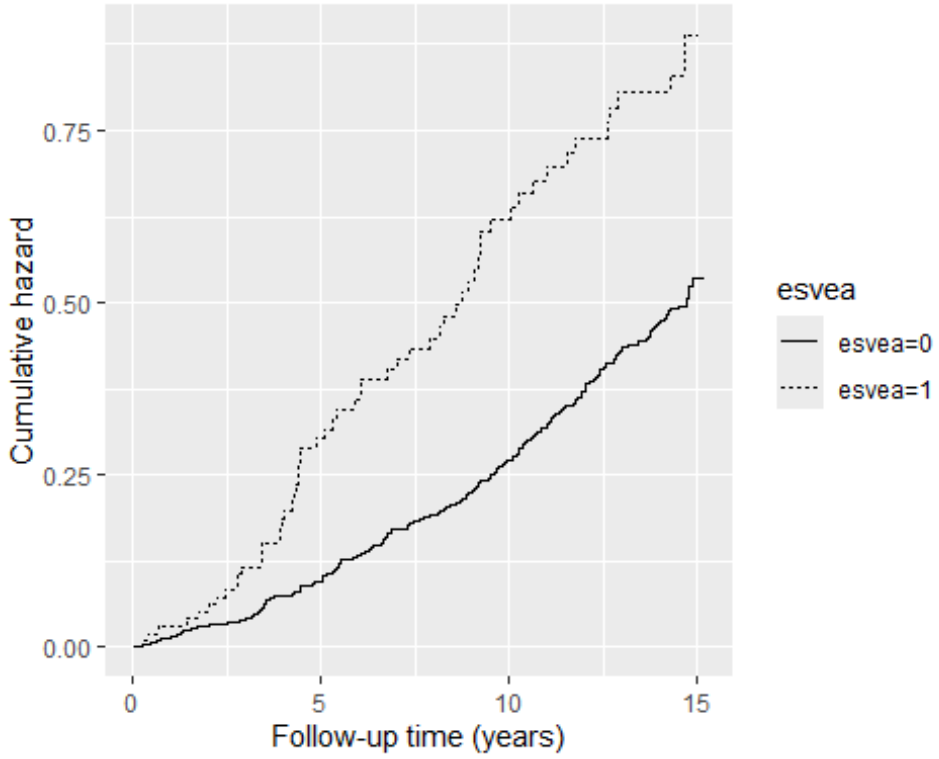
```
# Note that the time variables, timestroke and timedeath, are
# measured in days. We will first convert them to years for easier
# interpretations.
chs_data <- read.csv("cphholter.csv")
chs_data$timestroke <- chs_data$timestroke/365.25
chs_data$timedeath  <- chs_data$timedeath/365.25

# 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)

# 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')
```




```r
# 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")
```



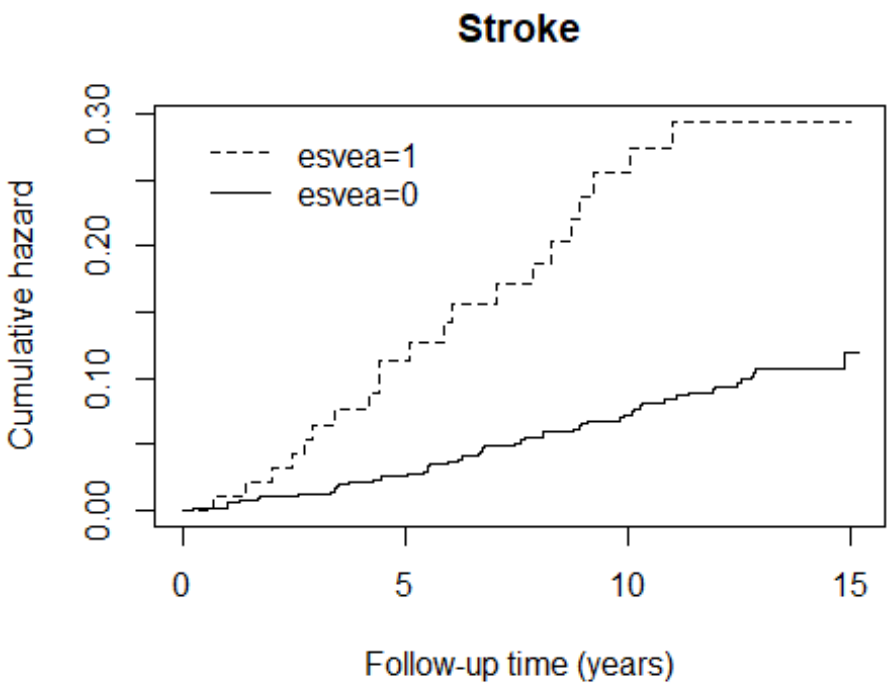

```
# Logrank test
survdiff(Surv(timestrokeordeath, strokeordeath) ~ esvea, data = chs_data)

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

          N Observed Expected (O-E)^2/E (O-E)^2/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.

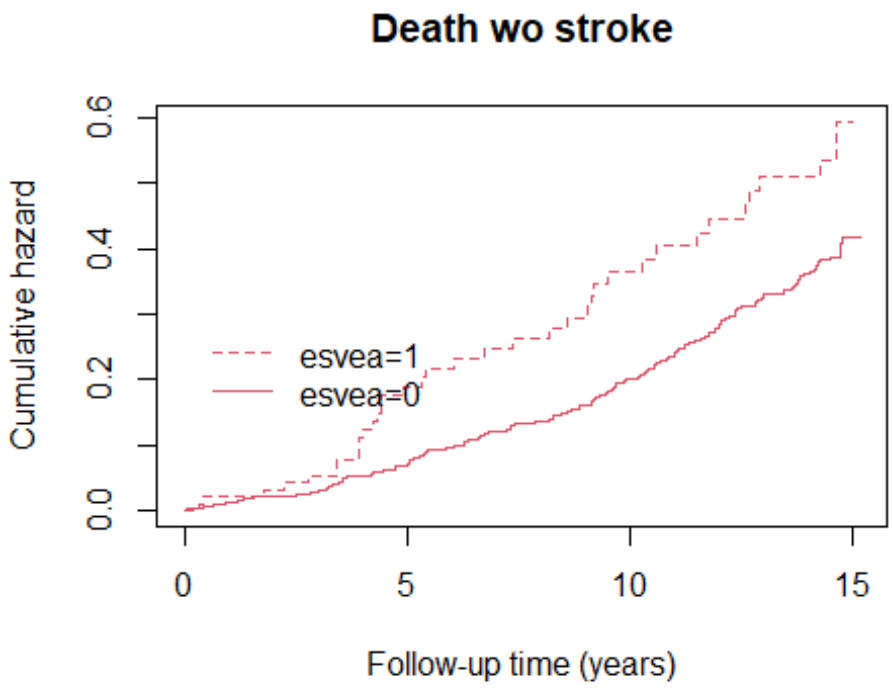
```
library(survival)

# 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
survdiff(Surv(timestrokeordeath, stroke) ~ esvea, data = chs_data)

Call:
survdiff(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')
```

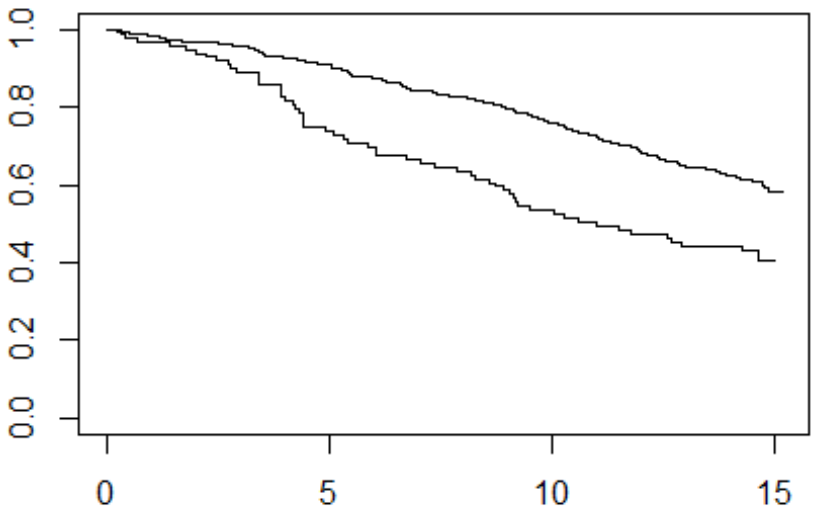



```
# Logrank test
survdiff(Surv(timestrokeordeath, death_wo_stroke) ~ esvea, data = chs_data)

Call:
survdiff(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
```

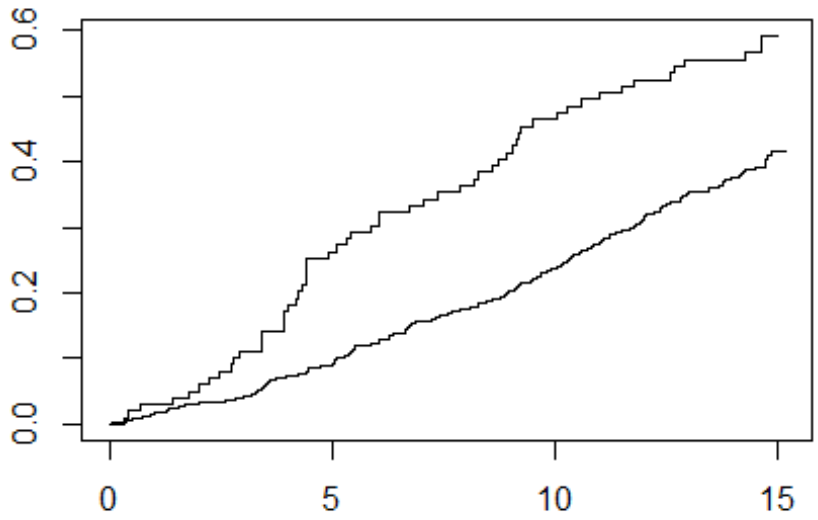
Ex 3

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

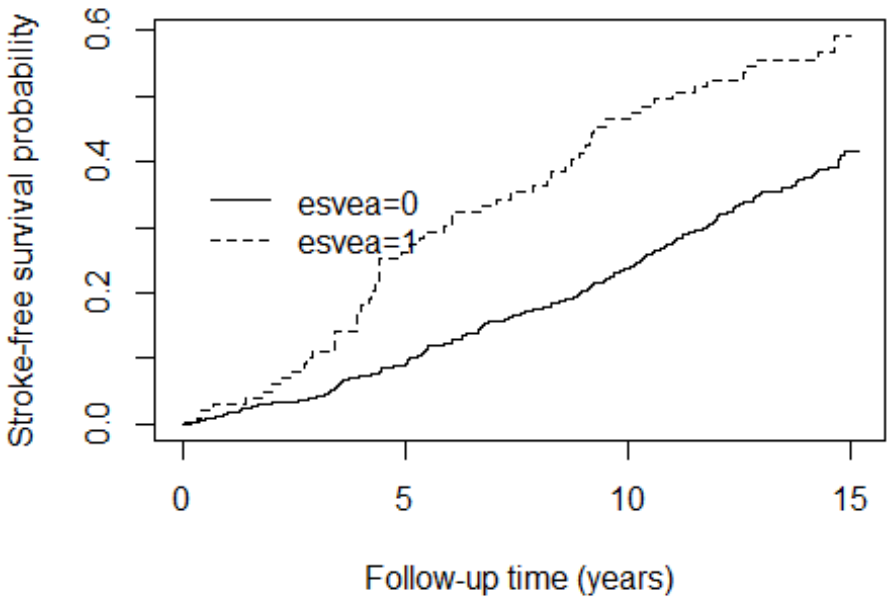
```
# Kaplan-Meier estimate of the survival functions
# We already have the fit from ex 1.1!
plot(sfit)
```



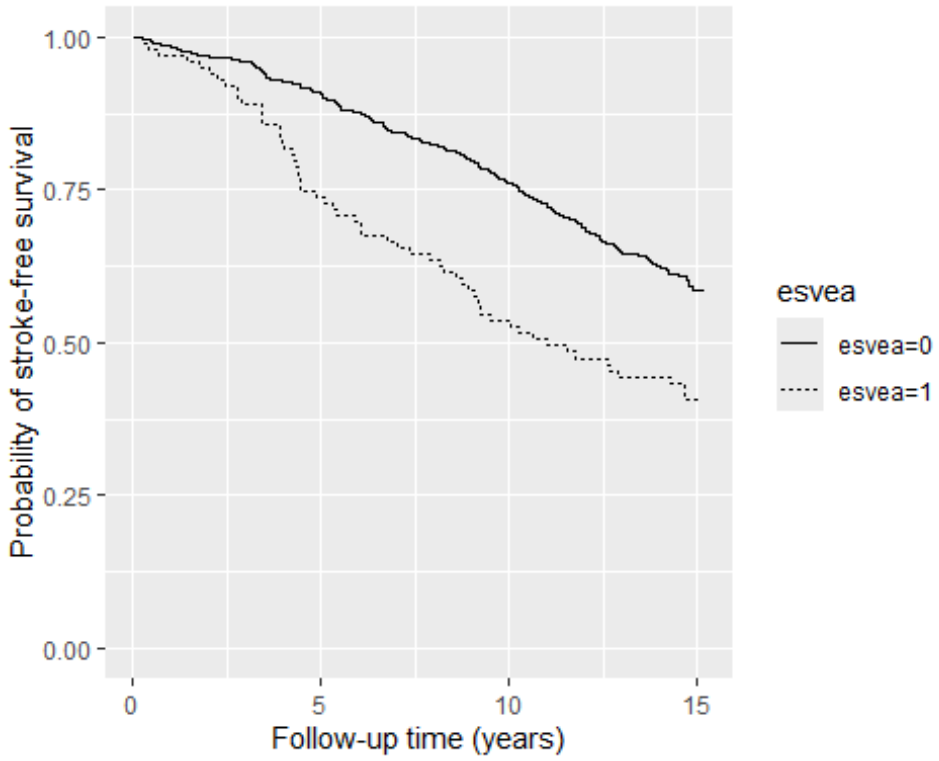

```
# or failure
plot(sfit,fun="event")
```




```
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')
```




```r
# 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")
```



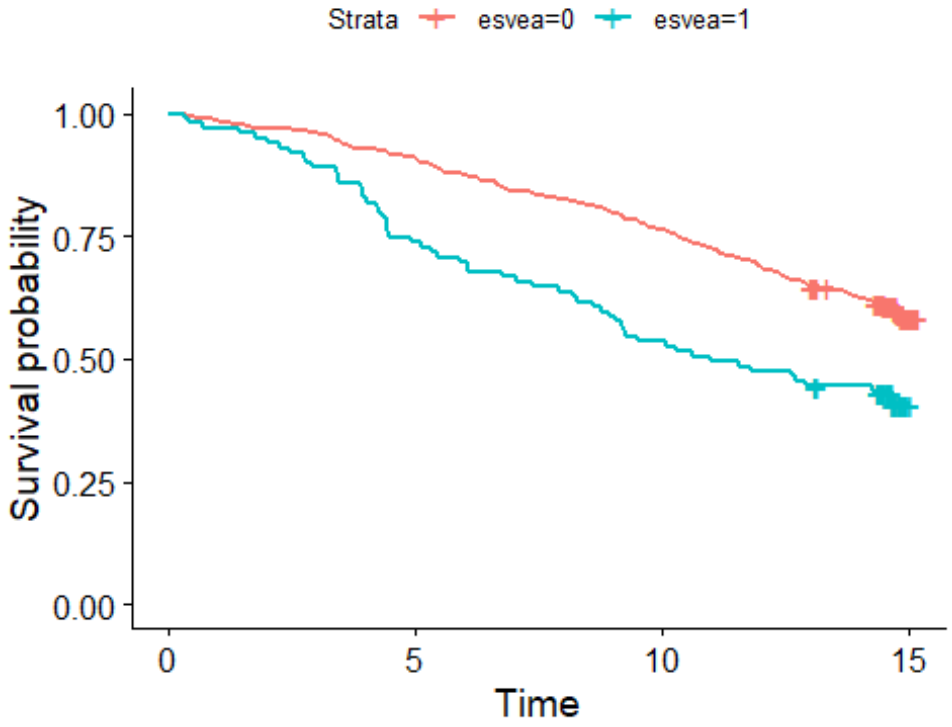

```
library(survminer)
```

```
Loading required package: ggpubr
```

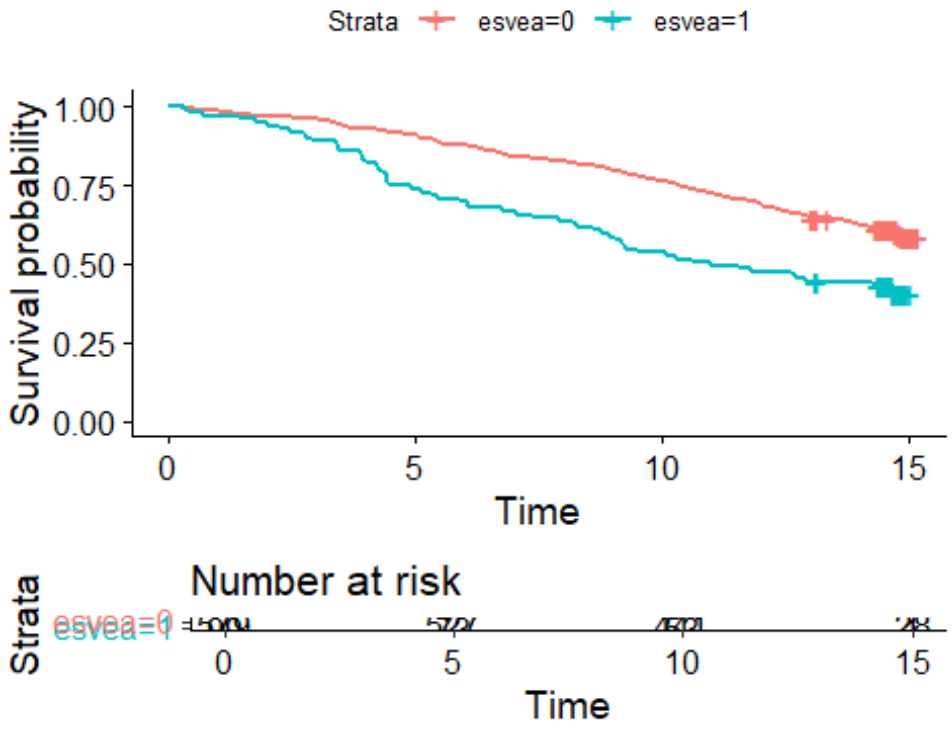
```

Attaching package: 'survminer'
```

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

    myeloma
```

```
ggsurvplot(sfit)
```



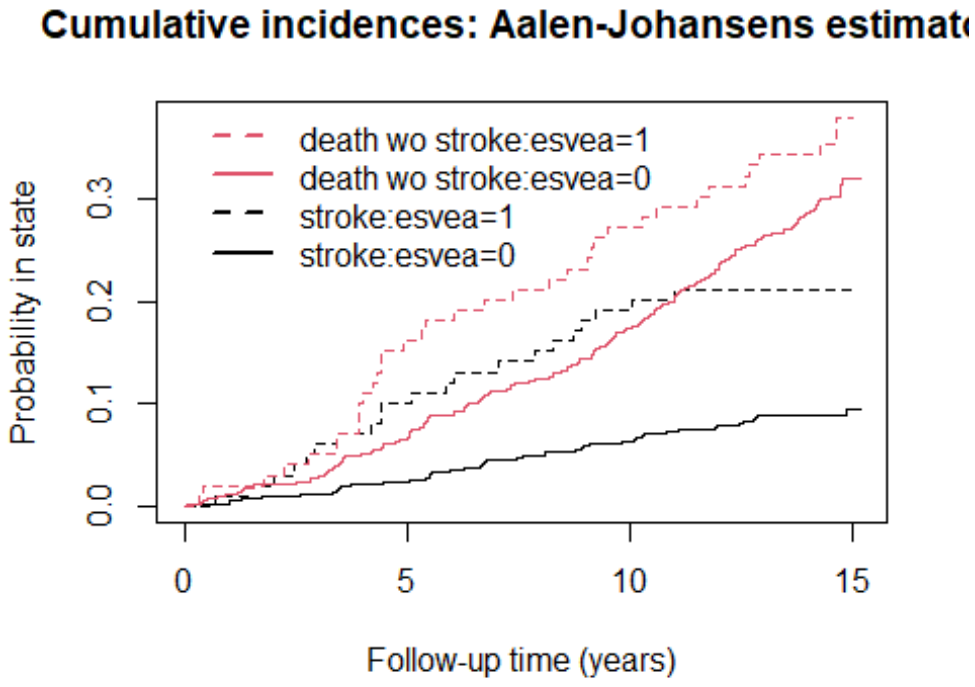

```
ggsurvplot(sfit,risk.table = TRUE)
```



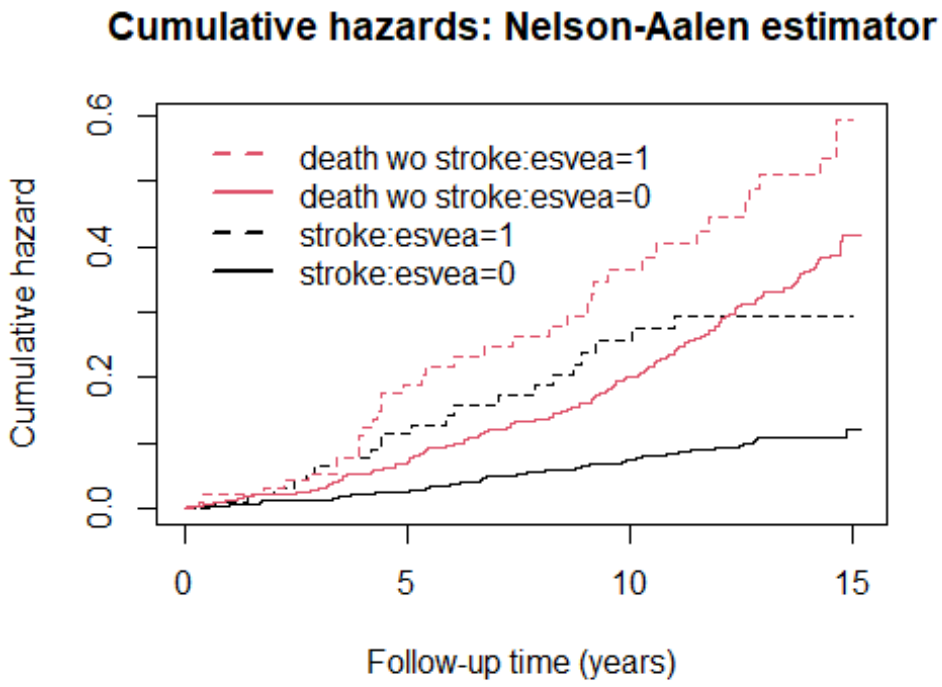
Ex 4

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

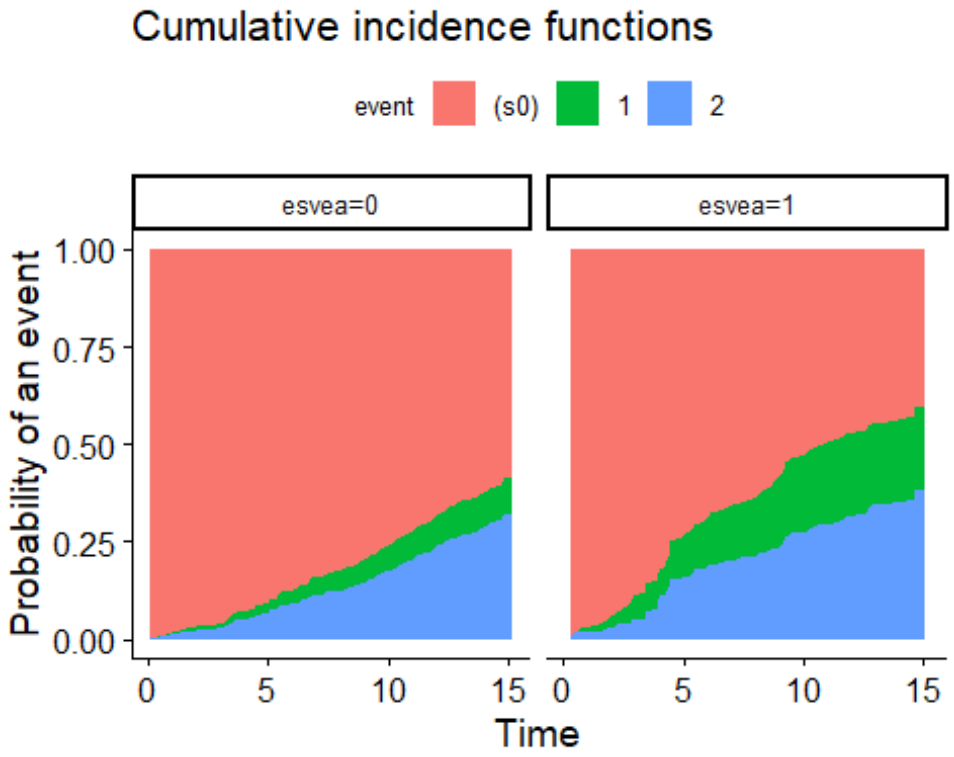
```
# Creating event variable, 0 = censored, 1 = stroke, 2 = death w/o stroke
chs_data$event <- with(chs_data,
                       ifelse(death_wo_stroke == 0,
                              stroke,
                              death_wo_stroke*2)
                       )

# 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')
```



```r
# 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')
```




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



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.

Ex 2

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

Any changes in conclusion regarding effect of treatment?

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.

Ex 4

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

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?

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.

Ex 3

Continuing the previous exercise, fit Ghosh-Lin models 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, incorrectly, LWYY models for the same expectations by treating death as censoring and compare with the correct analysis.

Day 4

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.

Ex 2

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

Ex 3

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

Ex 4

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

Ex 5

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

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.

Ex 7

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

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.

Ex 2

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

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.

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.

Ex 5

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

Ex 6

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

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.

Ex 8

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

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.

Ex 10

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