Examples of output from plotting functions

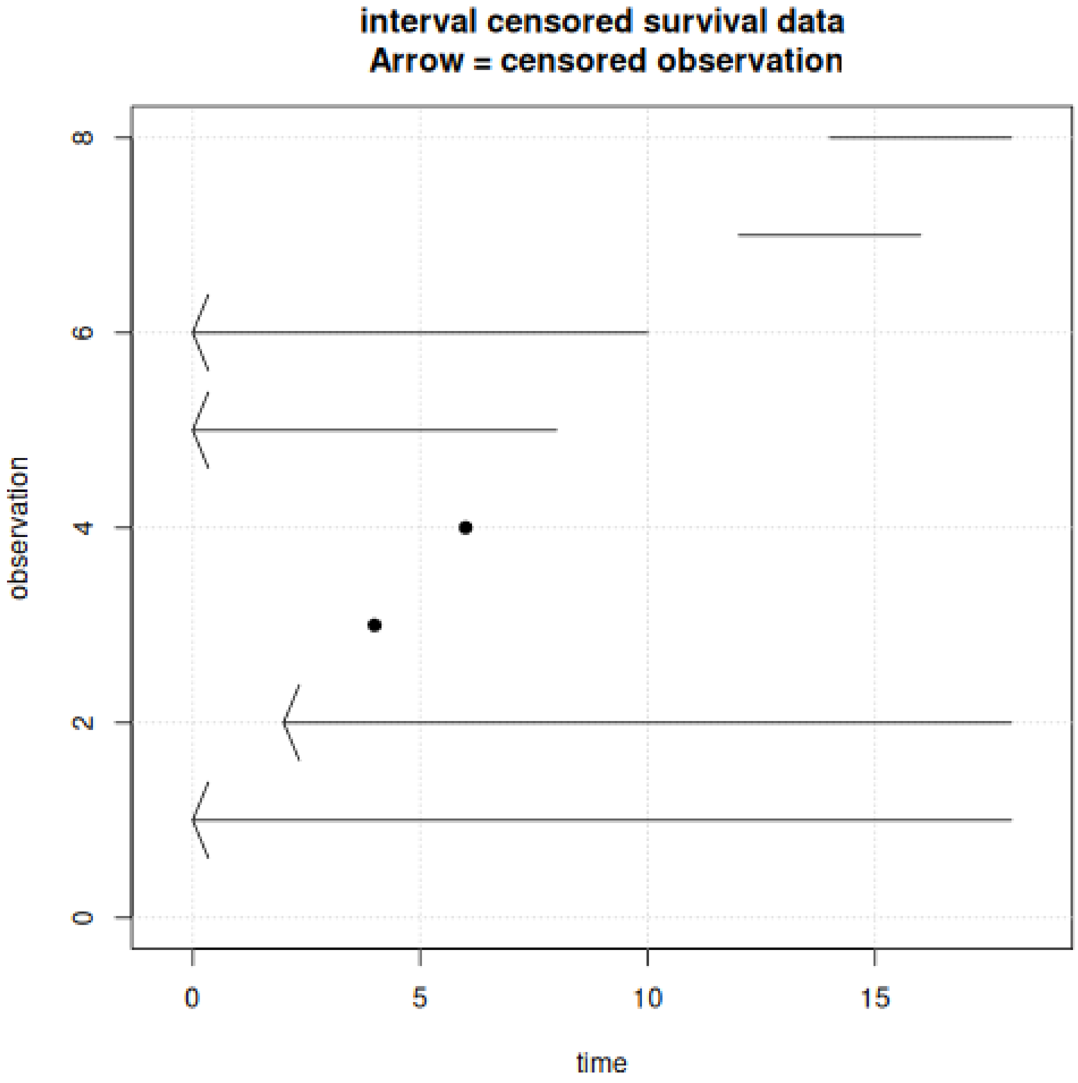
C Dardis

November 25, 2024

Some minimal examples showing the output of plots from the examples.

1 plotSurv

```
library("survMisc")

## Loading required package:  survival
## Registered S3 method overwritten by 'survMisc':
##   method    from
##   plot.Surv survival

df0 <- data.frame(t1=c(0, 2, 4, 6, NA, NA, 12, 14),
                  t2=c(NA, NA, 4, 6, 8, 10, 16, 18))
s1 <- Surv(df0$t1, df0$t2, type="interval2")
plot(s1, l=2)
```



2 autoplot.Ten

The ’autoplot’ function is a generic S3 method used by ’ggplot2’.

2.1 Simple examples

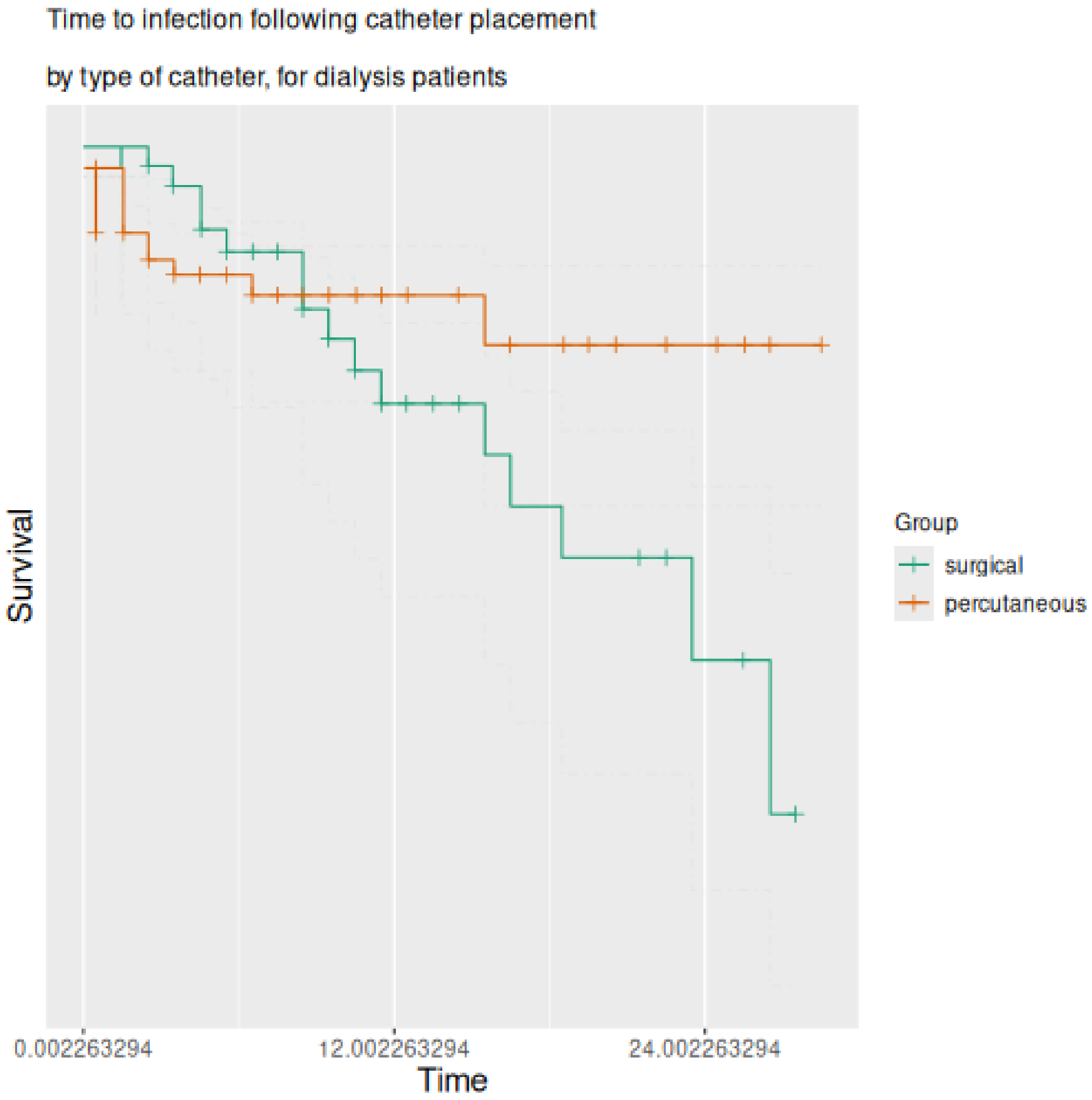
```
data("kidney", package="KMsurv")
t1 <- ten(survfit(Surv(time, delta) ~ type, data=kidney))
autoplot(t1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

Now, we increase the line size and use jitter to prevent overlap; we also make the relative size of the table larger.

```
print(autoplot(t1, type="fill", survLineSize=2, jitter="all"), tabHeight=0.35)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

A more customized example follows. Note that we return only the element marked 'plot' from the result (which is a list with two elements).

```
autoplot(t1, timeTicks="months",
         type="CI", jitter="all",
         legLabs=c("surgical", "percutaneous"),
         title="Time to infection following catheter placement \n
by type of catheter, for dialysis patients",
titleSize=10, censSize=2)$plot
```



Here we assign the result in order to modify the y axis.

```
str(a1 <- autoplot(t1), max.level=1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

## check the output is what we want
a1$plot + ggplot2::scale_y_continuous(limits=c(0.8, 1), name="Survival")

## Error:  object 'a1' not found

## this is one simple way
```

```
a1 <- autoplot(t1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

suppressMessages(a1$plot <- a1$plot +
                     ggplot2::scale_y_continuous(limits=c(0.8, 1), name="Survival"))

## Error:  object 'a1' not found

a1

## Error:  object 'a1' not found

## or we can assign them as follows
a1 <- autoplot(t1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

ls(a1$plot$scales$scales[[3]]$super$super)

## Warning in ls(a1$plot$scales$scales[[3]]$super$super):  'a1$plot$scales$scales[[3]]$super$super'
converted to character string
## Error in as.environment(pos):  no item called "a1$plot$scales$scales[[3]]$super$super"
on the search list

is.environment(a1$plot$scales$scales[[3]]$super$super$limits)

## Error:  object 'a1' not found

is.null(a1$plot$scales$scales[[3]]$super$super$limits)

## Error:  object 'a1' not found

a1$plot$scales$scales[[3]]$super$super$limits <- c(0.8, 1)

## Error:  object 'a1' not found

a1

## Error:  object 'a1' not found
```

2.2 Modifying the legend

Reordering the legend labels (example with 3 groups).

```
data("bmt", package="KMsurv")
b1 <- ten(Surv(time=t2, event=d3) ~ group, data=bmt)
autoplot(b1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

```
autoplot(b1, legOrd=c(1, 3, 2))

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

Here we also re-label the legend.

```
autoplot(b1, legOrd=c(3, 2, 1), legLabs=letters[1:3])

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

Now, let’s put the legend inside the plot itself.

```
a2 <- autoplot(b1)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

## ensure this is what we want
a2$plot + ggplot2::theme(legend.position=c(0.75, 0.75))

## Error:  object ’a2’ not found

a2$plot <- a2$plot + ggplot2::theme(legend.position=c(0.75, 0.75))

## Error:  object ’a2’ not found

a2

## Error:  object ’a2’ not found
```

2.3 One group only

A number of options for plotting a line with just one group.

```
t2 <- ten(survfit(Surv(time=time, event=delta) ~ 1, data=kidney))
autoplot(t2, legLabs="")$plot

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

autoplot(t2, legend=FALSE)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

2.4 Using confidence bands

Here we change the default pointwise confidence intervals to bands.

```
data("rectum.dat", package="km.ci")
t3 <- ten(survfit(Surv(time, status) ~ 1, data=rectum.dat))
## change confidence intervals to confidence bands
ci(t3, how="nair", tL=1, tU=40)

## Warning in (function (..., list = character(), package = NULL, lib.loc = NULL, :
data set ’critical.value.nair.95’ not found

##      cg   t     S      Sv     SCV lower upper
##   1:  1 1.1  0.93 0.00033 0.00039  0.85  0.97
##   2:  1 1.1  0.92 0.00037 0.00044  0.83  0.96
##   3:  1 1.2  0.91 0.00041   5e-04  0.82  0.95
##   4:  1 1.3   0.9 0.00043 0.00053  0.82  0.95
##   5:  1 1.4   0.9 0.00045 0.00056  0.81  0.95
##  ---
## 155:  1  36 0.095 0.00042   0.048 0.044  0.17
## 156:  1  36  0.09   4e-04    0.05 0.041  0.16
## 157:  1  39 0.085 0.00038   0.054 0.037  0.16
## 158:  1  40  0.08 0.00036   0.057 0.034  0.15
## 159:  1  40 0.075 0.00034   0.062 0.031  0.14

autoplot(t3, type="fill", alpha=0.6, legend=FALSE)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero
```

2.5 More customization

If the output of ’autoplot.ten’ is assigned, it can be modified in place. The list elements are ggplot2 objects which can be altered as usual.

```
## manually changing the output
t4 <- ten(survfit(Surv(time, delta) ~ type, data=kidney))
(a4 <- autoplot(t4, type="CI", alpha=0.8, survLineSize=2)$plot)

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

## change default colors
suppressMessages(a4 + list(
                          ggplot2::scale_color_manual(values=c("red", "blue")),
                          ggplot2::scale_fill_manual(values=c("red", "blue"))))

## Error:  object ’a4’ not found

## change limits of y-axis
suppressMessages(a4 + ggplot2::scale_y_continuous(limits=c(0, 1)))

## Error:  object ’a4’ not found
```

3 autoplot.StratTen

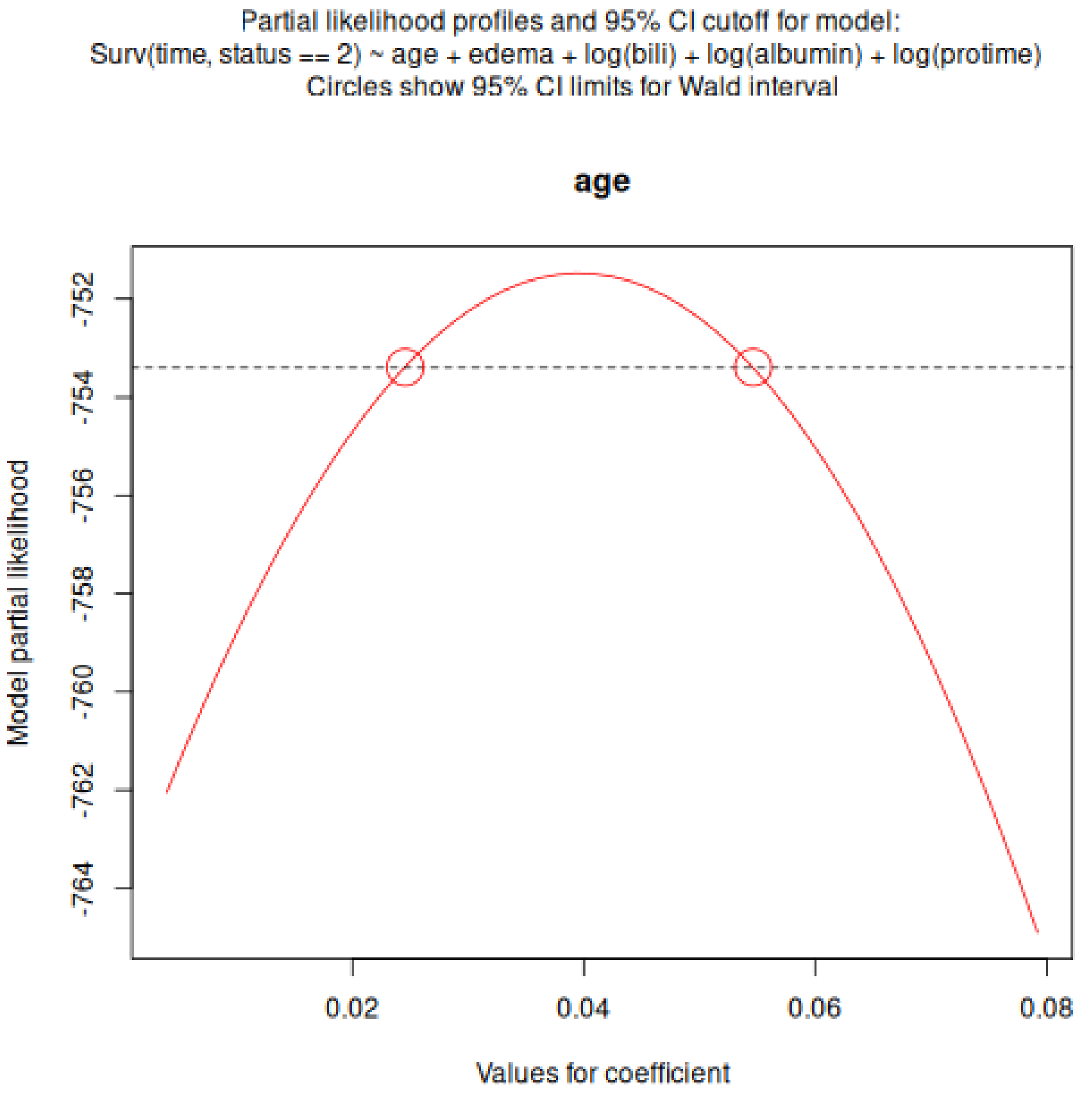
An example of the plots from a stratified model:

```
data("pbc", package="survival")
t1 <- ten(Surv(time, status==2) ~ trt + strata(edema), data=pbc, abbNames=FALSE)
suppressWarnings(str(a1 <- autoplot(t1), max.level=1))

## Error in if (is.nan(times1[1])) times1 <- times:  argument is of length zero

a1

## Error:  object ’a1’ not found
```

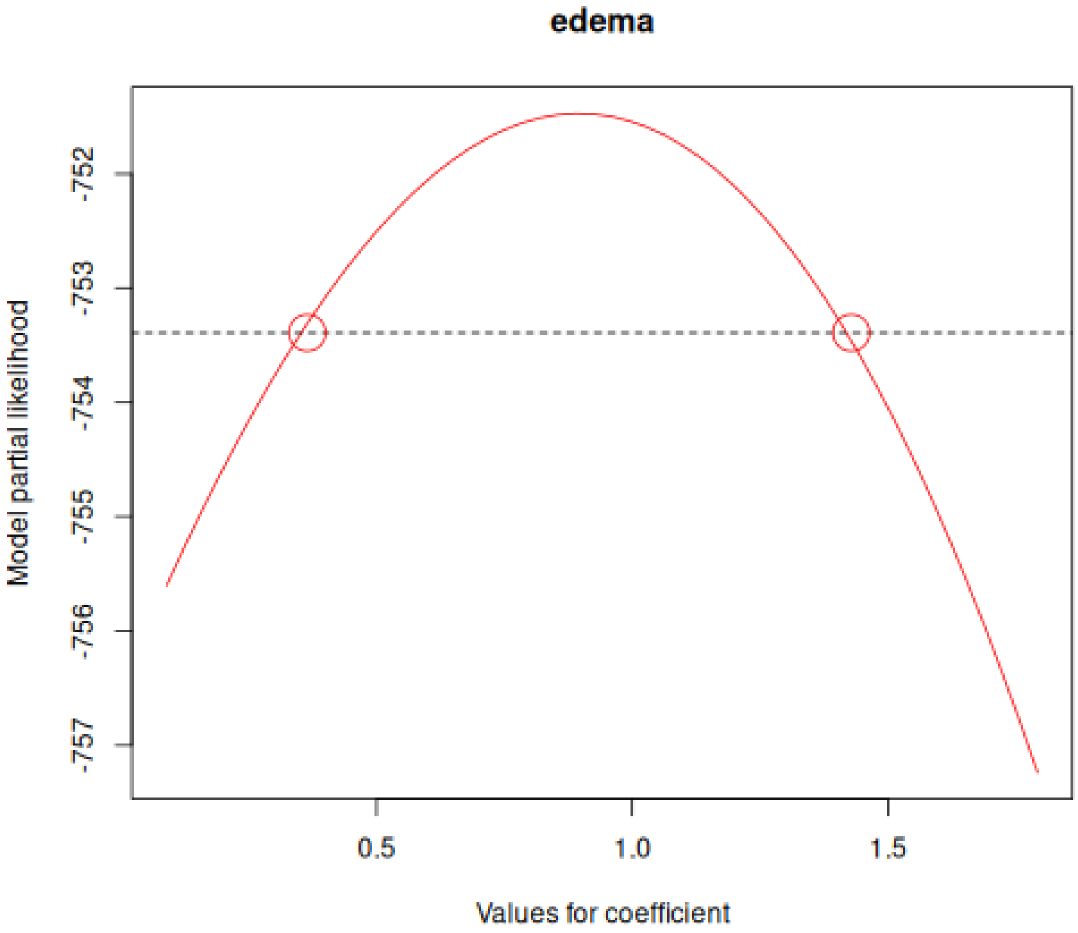
4 profLik

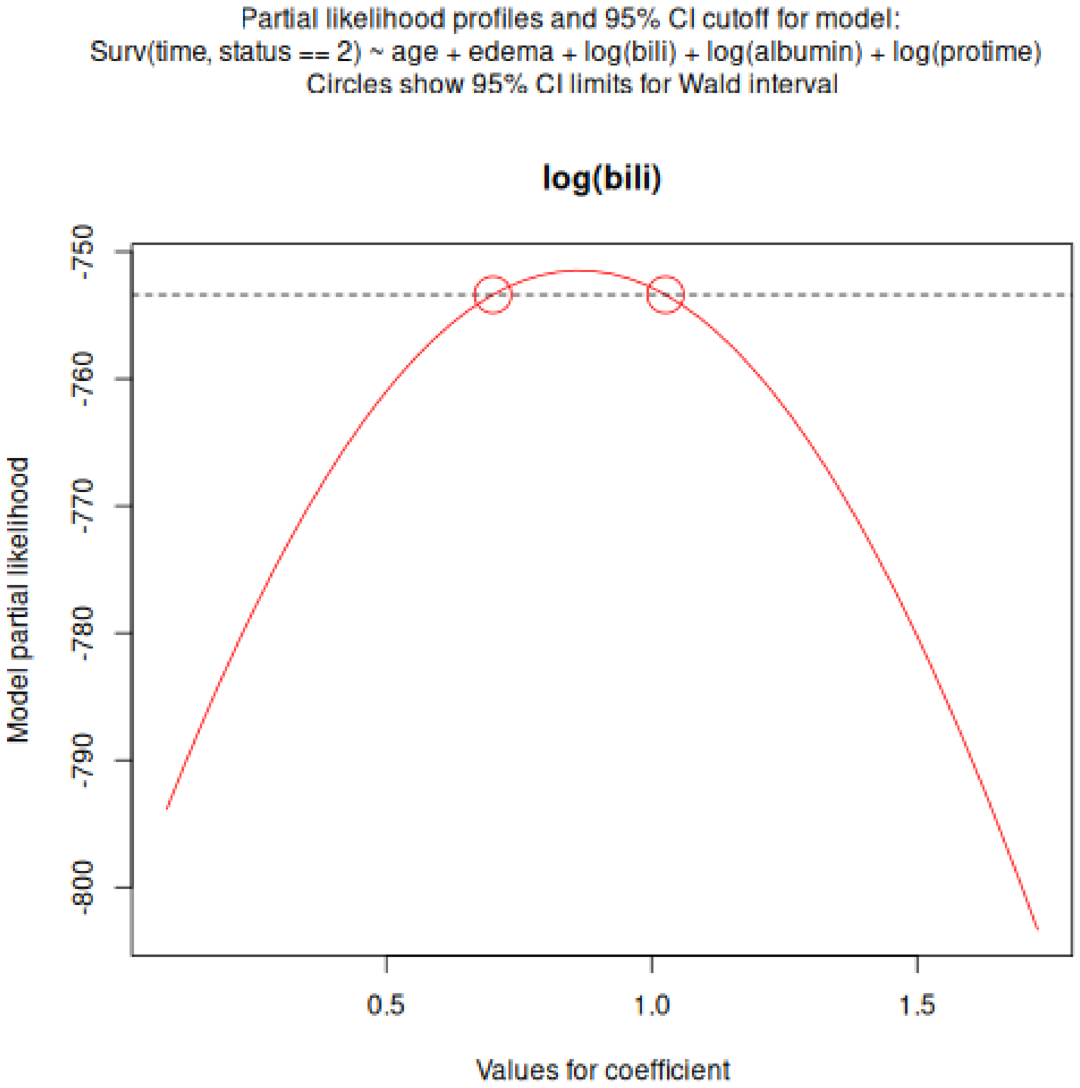
Plotting profile likelihood.

```
data("pbc", package="survival")
c1 <- survival::coxph(formula = Surv(time, status == 2) ~ age + edema + log(bili) +
                          log(albumin) + log(protime), data = pbc)
profLik(c1, col="red", devNew=FALSE)
```


Partial likelihood profiles and 95% CI cutoff for model:
Surv(time, status == 2) ~ age + edema + log(bili) + log(albumin) + log(protime)
Circles show 95% CI limits for Wald interval
age
Model partial likelihood
-752
-754
-756
-758
-760
-762
-764
0.02
0.04
0.06
0.08
Values for coefficient

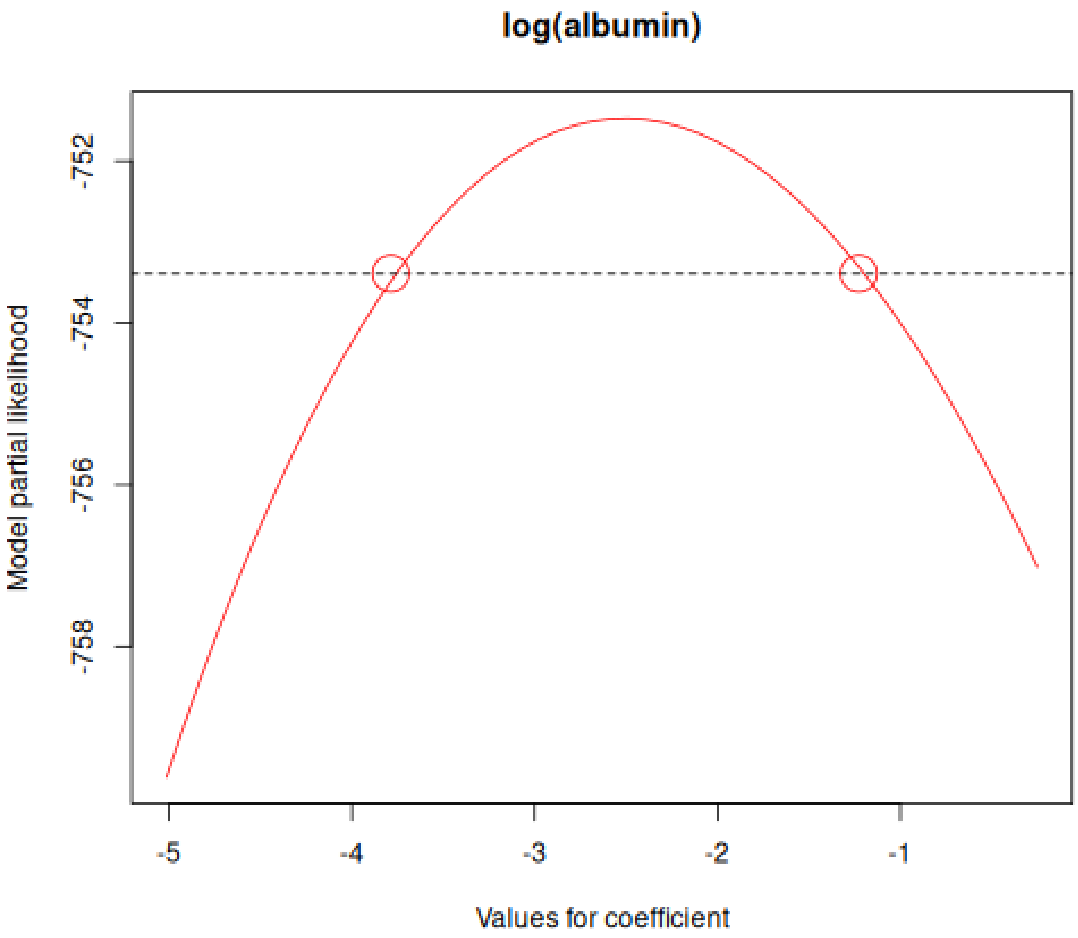
Partial likelihood profiles and 95% CI cutoff for model:
Surv(time, status == 2) ~ age + edema + log(bili) + log(albumin) + log(protime)
Circles show 95% CI limits for Wald interval



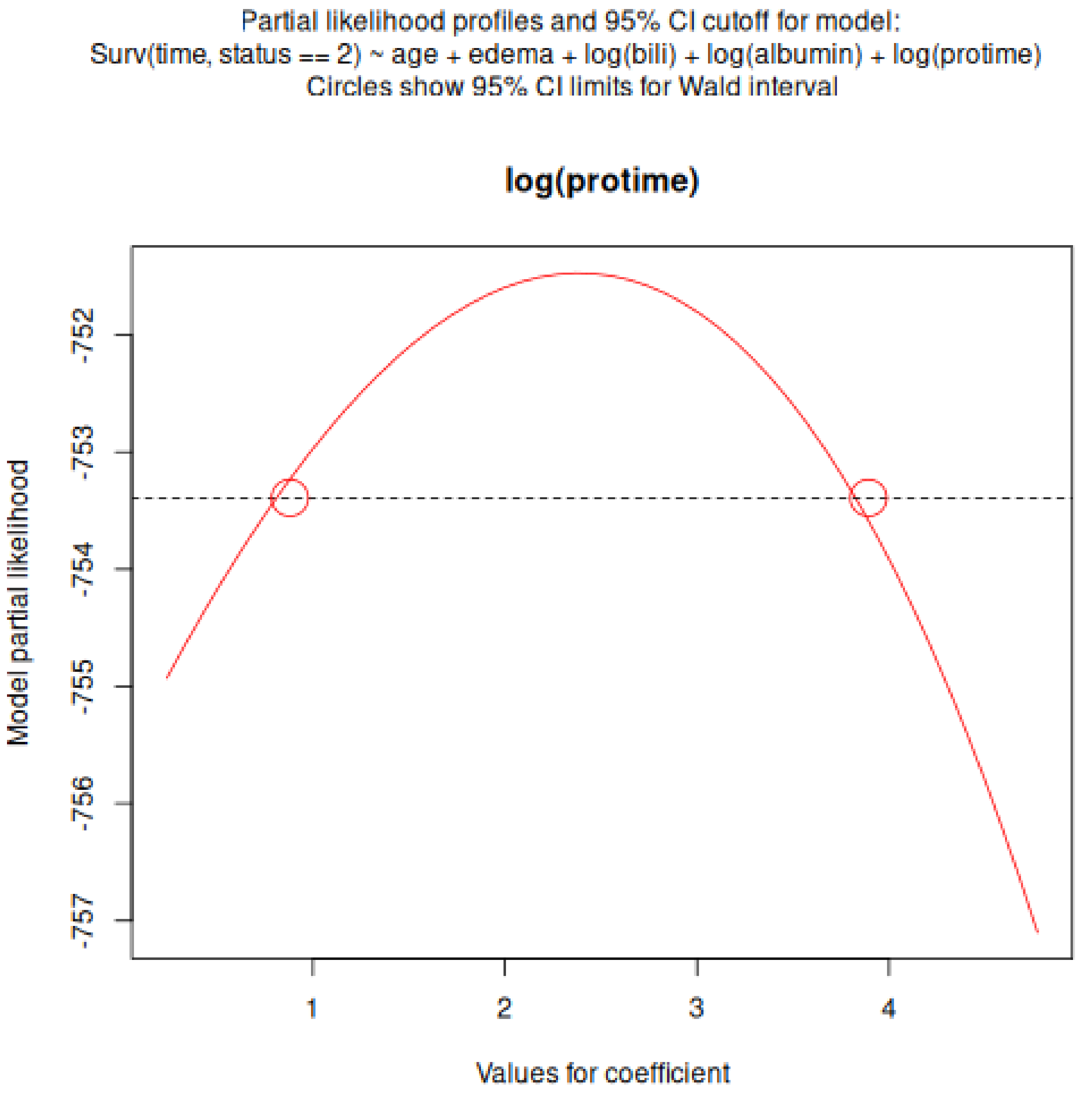


Partial likelihood profiles and 95% CI cutoff for model:
Surv(time, status == 2) ~ age + edema + log(bili) + log(albumin) + log(protime)
Circles show 95% CI limits for Wald interval

Partial likelihood profiles and 95% CI cutoff for model:
Surv(time, status == 2) ~ age + edema + log(bili) + log(albumin) + log(protime)
Circles show 95% CI limits for Wald interval



log(albumin)


Partial likelihood profiles and 95% CI cutoff for model:
Surv(time, status == 2) ~ age + edema + log(bili) + log(albumin) + log(protime)
Circles show 95% CI limits for Wald interval
log(protime)
Model partial likelihood
-752
-753
-754
-755
-756
-757
1
2
3
4
Values for coefficient