Examples of output from plotting functions

C Dardis

November 13, 2026

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

1 autoplot.Ten

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

1.1 Simple examples

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

## Error in `ten()`:
## !  could not find function "ten"

autoplot(t1)

## Error in `autoplot()`:
## !  could not find function "autoplot"
```

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 `autoplot()`:
## !  could not find function "autoplot"
```

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

## Error in `autoplot()`:
## !  could not find function "autoplot"
```

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

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

## Error in `autoplot()`:
## !  could not find function "autoplot"

## 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 `autoplot()`:
## !  could not find function "autoplot"

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 `autoplot()`:
## !  could not find function "autoplot"

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()`:
## !  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
```

1.2 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)
```

```
## Error in `ten()`:
## !  could not find function "ten"
```

```
autoplot(b1)
```

```
## Error in `autoplot()`:
## !  could not find function "autoplot"
```

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

```
## Error in `autoplot()`:
## !  could not find function "autoplot"
```

Here we also re-label the legend.

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

```
## Error in `autoplot()`:
## !  could not find function "autoplot"
```

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

```
a2 <- autoplot(b1)

## Error in `autoplot()`:
## !  could not find function "autoplot"

## 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
```

1.3 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))

## Error in `ten()`:
## !  could not find function "ten"

autoplot(t2, legLabs="")$plot

## Error in `autoplot()`:
## !  could not find function "autoplot"

autoplot(t2, legend=FALSE)

## Error in `autoplot()`:
## !  could not find function "autoplot"
```

1.4 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))
```

```
## Error in ‘ten()‘:
## !  could not find function "ten"

## change confidence intervals to confidence bands
ci(t3, how="nair", tL=1, tU=40)

## Error in ‘ci()‘:
## !  could not find function "ci"

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

## Error in ‘autoplot()‘:
## !  could not find function "autoplot"
```

1.5 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))

## Error in ‘ten()‘:
## !  could not find function "ten"

(a4 <- autoplot(t4, type="CI", alpha=0.8, survLineSize=2)$plot)

## Error in ‘autoplot()‘:
## !  could not find function "autoplot"

## 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
```

2 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)

## Error in ‘ten()‘:
## !  could not find function "ten"

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

## Error in ‘autoplot()‘:
## !  could not find function "autoplot"

a1

## Error:
## !  object ’a1’ not found
```

3 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)

## Error in ‘profLik()‘:
## !  could not find function "profLik"
```