

The km.plot and sync.ylab.widths functions

Greg Cicconetti

September 23, 2014

Contents

1 Example 1: Juxtaposing two Kaplan-Meier graphics and their At Risk tables	1
1.1 Create the graphic components	2
1.2 Syncing the widths of the figures	7
1.3 Assembling the page and discussion	7

1 Example 1: Juxtaposing two Kaplan-Meier graphics and their At Risk tables

Intialize a session:

```
remove(list=ls())
require(figures2)
require(survival)
require(ggplot2)
require(plyr)
require(scales)
default.settings()
```

A data set included in the figures2 package is loaded. Note that this data contains one row per subject and has a censor column and a centime column.

```
data(km.data)
working.df <- km.data
head(working.df)
```

```
  CENTREID SUBJID AGE SEX   TRTGRP COUNTRY      REGION CENSOR
1    58785     1  64  M Treatment  Canada North America    0
2    58785     2  73  M Treatment  Canada North America    0
3    58785     3  39  M  Placebo  Canada North America    0
4    58785     4  63  M Treatment  Canada North America    0
5    58785     5  59  M Treatment  Canada North America    0
6    58785     6  73  M  Placebo  Canada North America    0
  CENTIME.DAY CENTIME.MON
1         1504      49.41273
2         1534      50.39836
3         1485      48.78850
4         1415      46.48871
5         1451      47.67146
6         1469      48.26283
```

The Goal: Create Kaplan-Meier plots for Males and Females and juxtapose these.

1.1 Create the graphic components

The `km.plot` function returns a list of objects. The first two objects are ggplot objects for the Kaplan-Meier graphic and corresponding at risk table.

```
km.M <- km.plot(parent.df = subset(working.df, SEX=="M"),
  censor.col = "CENSOR",
  centime.col = "CENTIME.MON",
  category.col = "TRTGRP",
  category.palette = c("red", "blue"),
  at.risk.palette = c("red","blue"),
  linetype.palette = c("solid","dotted"),
  x.label = "Time Since Randomization (months)",
  x.limits=c(-3,48),
  x.ticks=seq(0,48,6),
  y.limits=c(0,.25),
  y.ticks=seq(0,.25,.05))
```

Inspecting the Kaplan-Meier Graphic:

```
km.M[[1]]
```

Inspecting the At Risk Table:

```
km.M[[2]]
```

A similar call for the females:

```
km.F <- km.plot(parent.df = subset(working.df, SEX=="F"),
  censor.col = "CENSOR",
  centime.col = "CENTIME.MON",
  category.col = "TRTGRP",
  category.palette = c("red", "blue"),
  at.risk.palette = c("red","blue"),
  linetype.palette = c("solid","dotted"),
  x.label = "Time Since Randomization (months)",
  x.limits=c(-3,48),
  x.ticks=seq(0,48,6),
  y.limits=c(0,.25),
  y.ticks=seq(0,.25,.05))
```

First object:

```
km.F[[1]]
```

Second object:

```
km.F[[2]]
```

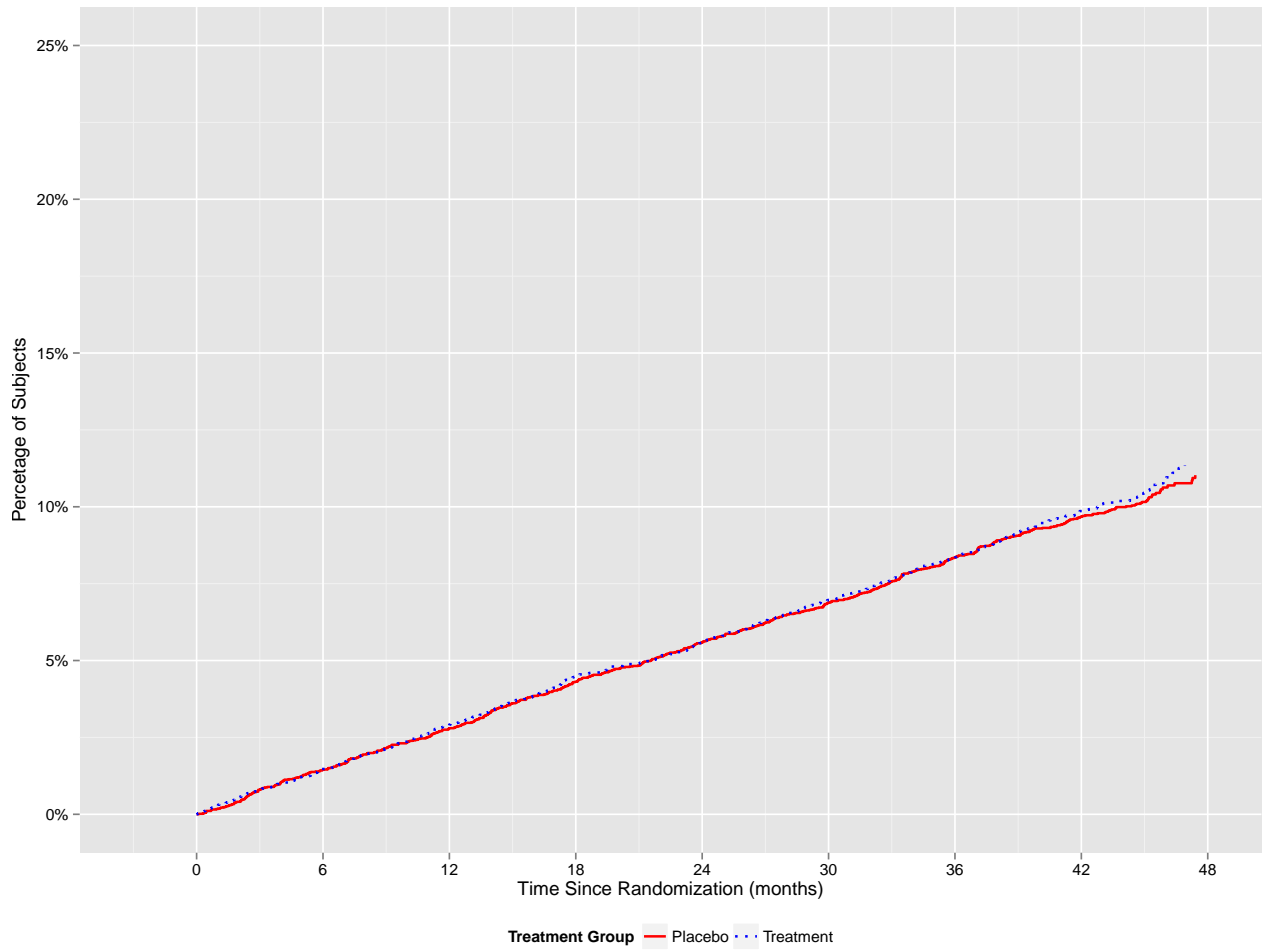


Figure 1: Kaplan-Meier Graphic for Males

	Baseline	12 weeks	24 weeks	36 weeks	48 weeks	60 weeks	72 weeks	84 weeks	96 weeks
Treatment	6426	6239	6097	5963	5852	5731	5587	4652	470
Placebo	6435	6261	6127	5995	5868	5751	5593	4670	471

Number of Subjects

Figure 2: Corresponding At Risk Table for Males

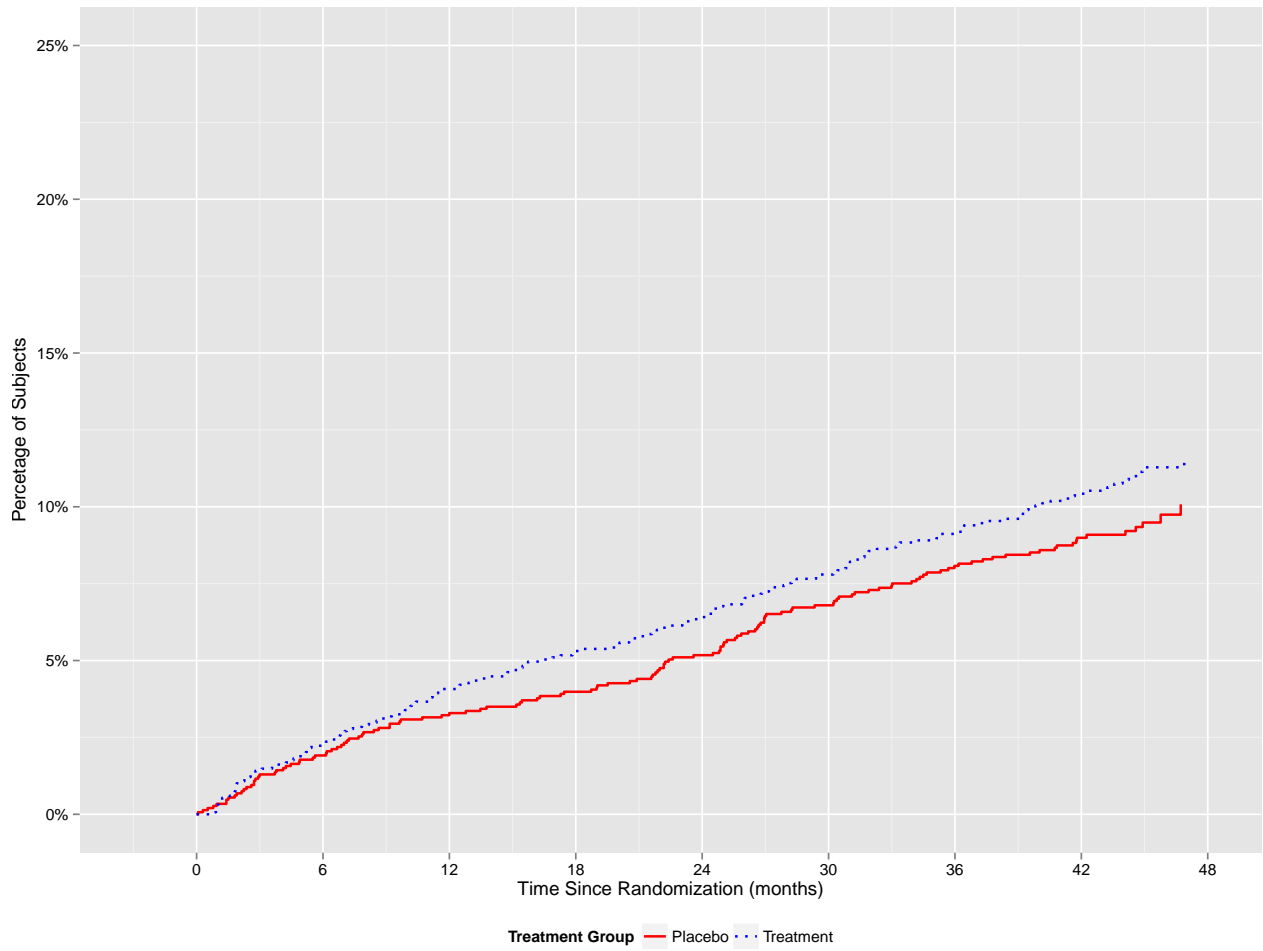


Figure 3: Kaplan-Meier Graph for Females

Number at Risk	Treatment	1494	1438	1400	1377	1356	1328	1298	1051	107
	Placebo	1473	1429	1402	1381	1349	1313	1281	1017	115
		Number of Subjects								

Figure 4: Number at Risk Table for Females

1.2 Syncing the widths of the figures

In this step we combine tasks of aligning the y axes of the KM curves & tables. The `sync.ylab.widths` function takes a list of ggplot objects and returns a list of the same length contain gtable objects. These can objects are different than ggplot objects. They can be visually inspected with `grid.draw` (grid package) and can be processed by the `build.page` function (as this is merely a wrapper for the `grid.arrange` function (gridExtra package)).

In this example, the list of ggplot objects being supplied to `sync.ylab.widths` are manipulations of the graphs already reviewed. In particular:

- legends have been suppressed
- margins have been altered; use this to manipulate the space between graphics.
- tick mark colors have been changed

```
comeback.M <- sync.ylab.widths(list(
  km.M[[1]]+
    ggtitle("Kaplan Meier-Plot of Time to\nFirst MACE: Males") +
    guides(color=F, linetype=F),
  km.M[[2]]+labs(x=NULL, y="At Risk")
))

comeback.F <- sync.ylab.widths(list(
  km.F[[1]]+
    ggtitle("Kaplan Meier-Plot of Time to\nFirst MACE: Females") +
    guides(color=F, linetype=F) +
    theme(axis.ticks.y=element_line(color="white")) +
    labs(y=NULL) +
    scale_y_continuous(labels=NULL, limits=c(0,.25), breaks=seq(0,.25,.05)),
  km.F[[2]]+
    labs(x=NULL, y=NULL) +
    scale_y_discrete(labels=NULL))
)
```

1.3 Assembling the page and discussion

One needs to iterate with minor changes until the final product is visually appealing. One needs to consider the dimensions allocated to the `interior.h` and `interior.w` below as well. E.g., this allocation seems to work well for the data set. However, if the treatment labels were shorter, the allocation of `interior.w` would need to change. Similarly if more arms were included, more rows in the At Risk table would demand a different distribution of `interior.h`.

In manipulating the plot margins keep in mind that overlapping graphics can obscure the edges of adjacent graphics/tables. E.g., note above that `c(-3, 48)` was passed to `x.limits` instead of `c(0, 48)`. With the latter in use, portions of the text in the At Risk table were obscured.

```
build.page(interior.h = c(.8, .2),
           interior.w = c(.6, .4),
           ncol=2, nrow=2,
           interior = list(comeback.M[[1]], comeback.F[[1]],
                          comeback.M[[2]], comeback.F[[2]]))
```

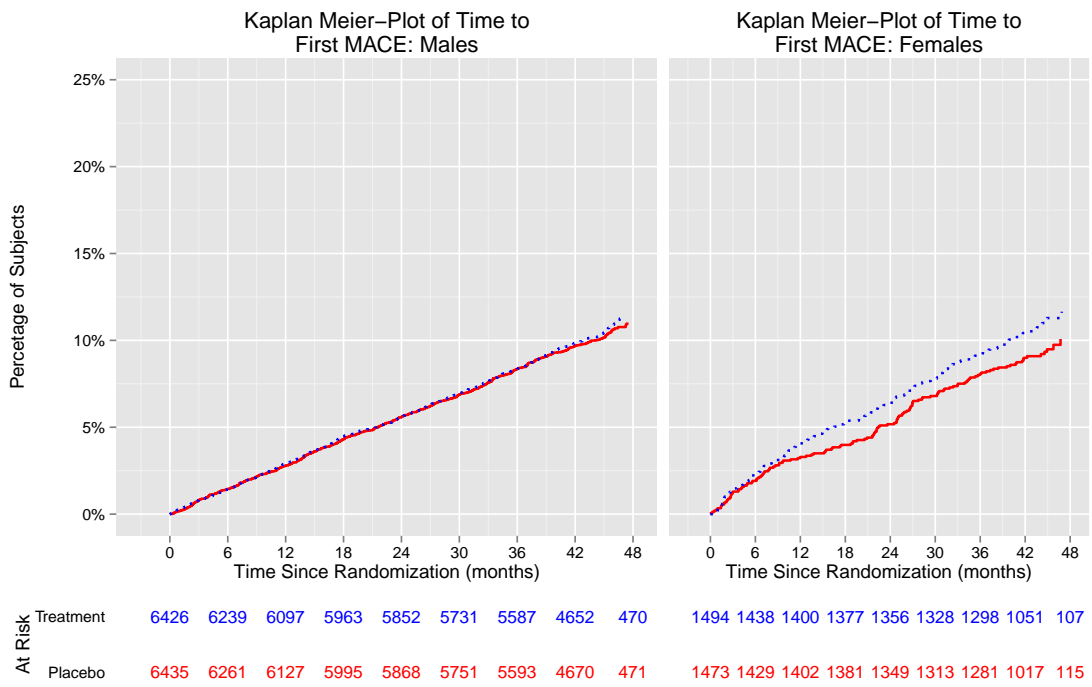


Figure 5: A Figure that arranges four graphics