



INTEGO
GROUP

Paper ST16

How to Customize Survival Curves in R and Add Interactivity

Oleksandr Babych

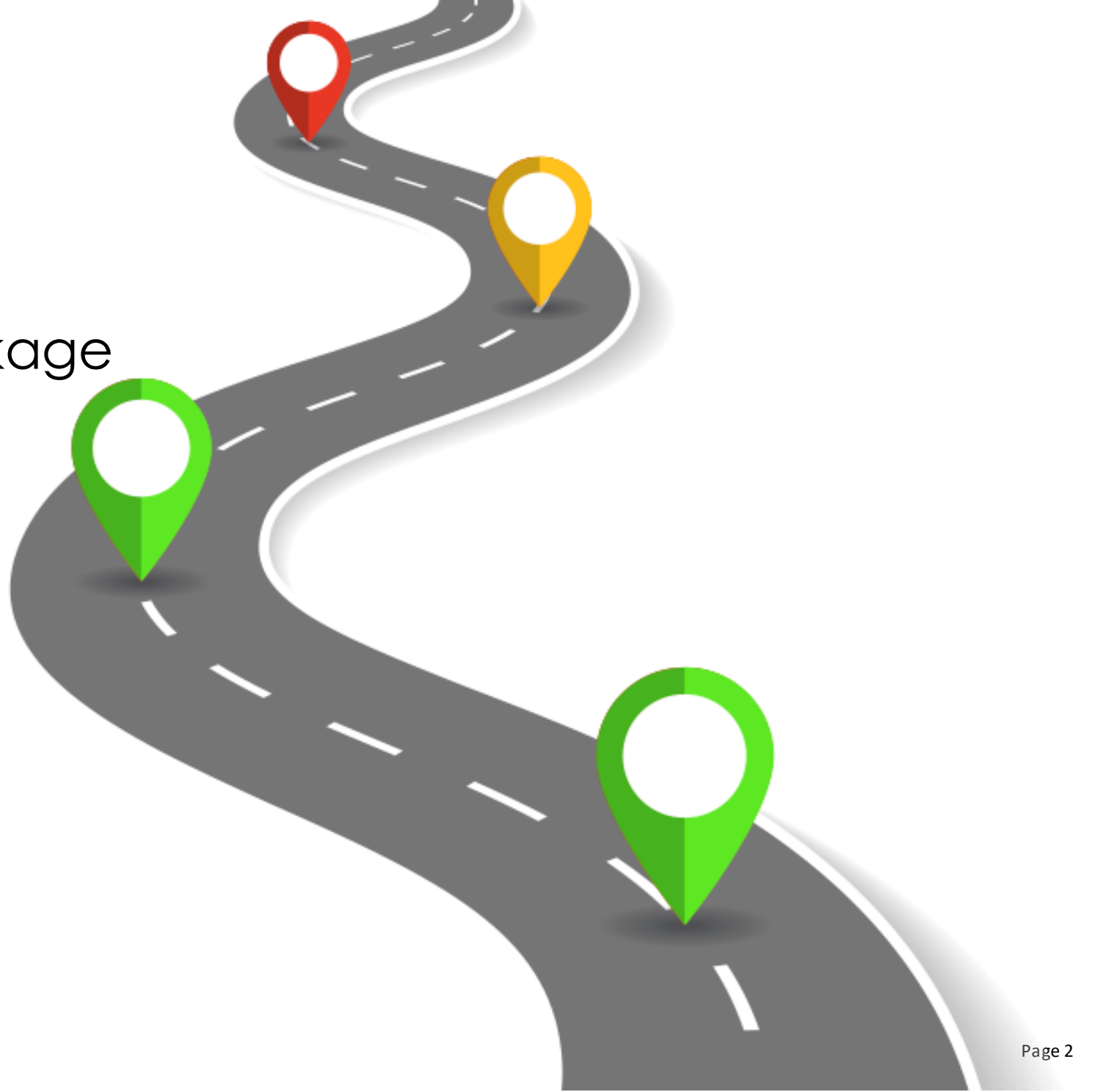


Orlando 2023



AGENDA

- Introduction: survival package
- Draw a KM curve in R
- Add interactivity
- Summary



Survival package

Main functions:

- `Surv()` – creates survival object
- `survfit()` - creates survival curves using the Kaplan-Meier method

	USUBJID	SUBJID	SEX	AGE	AGEU	PARAMCD	PARAM	AVAL	AVALU	CNSR
1	01-001	001	1	74	YEARS	PFS	Progression Free Survival	306	DAYS	0
2	01-002	002	1	68	YEARS	PFS	Progression Free Survival	455	DAYS	0
3	01-003	003	1	56	YEARS	PFS	Progression Free Survival	1010	DAYS	1
4	01-004	004	1	57	YEARS	PFS	Progression Free Survival	210	DAYS	0
5	01-005	005	1	60	YEARS	PFS	Progression Free Survival	883	DAYS	0
6	01-006	006	1	74	YEARS	PFS	Progression Free Survival	1022	DAYS	1
7	01-007	007	2	68	YEARS	PFS	Progression Free Survival	310	DAYS	0
8	01-008	008	2	71	YEARS	PFS	Progression Free Survival	361	DAYS	0
9	01-009	009	1	53	YEARS	PFS	Progression Free Survival	218	DAYS	0
10	01-010	010	1	61	YEARS	PFS	Progression Free Survival	166	DAYS	0

```
surv_obj <- Surv(time = ADTTE$AVAL, event = ADTTE$CNSR == 0)
head(surv_obj, 10)
```

```
[1] 306 455 1010+ 210 883 1022+ 310 361 218 166
```

Create survival curves

```
surv_fit <- survfit(formula = Surv(time = AVAL, event = CNSR == 0) ~ 1, data = ADTTE)
surv_fit <- survfit(surv_obj ~ 1)
```

Investigate the survfit object

- `summary(surv_fit, times = c(0,10,20,40,80))`

time	n.risk	n.event	survival	std.err	lower	95% CI	upper	95% CI
0	228	0	1.000	0.00000		1.000		1.000
10	227	1	0.996	0.00438		0.987		1.000
20	220	7	0.965	0.01219		0.941		0.989
40	217	3	0.952	0.01419		0.924		0.980
80	205	12	0.899	0.01995		0.861		0.939

- `print(surv_fit, print.rmean = TRUE)`

n	events	rmean*	se(rmean)	median	0.95LCL	0.95UCL
228	165	376	19.7	310	285	363

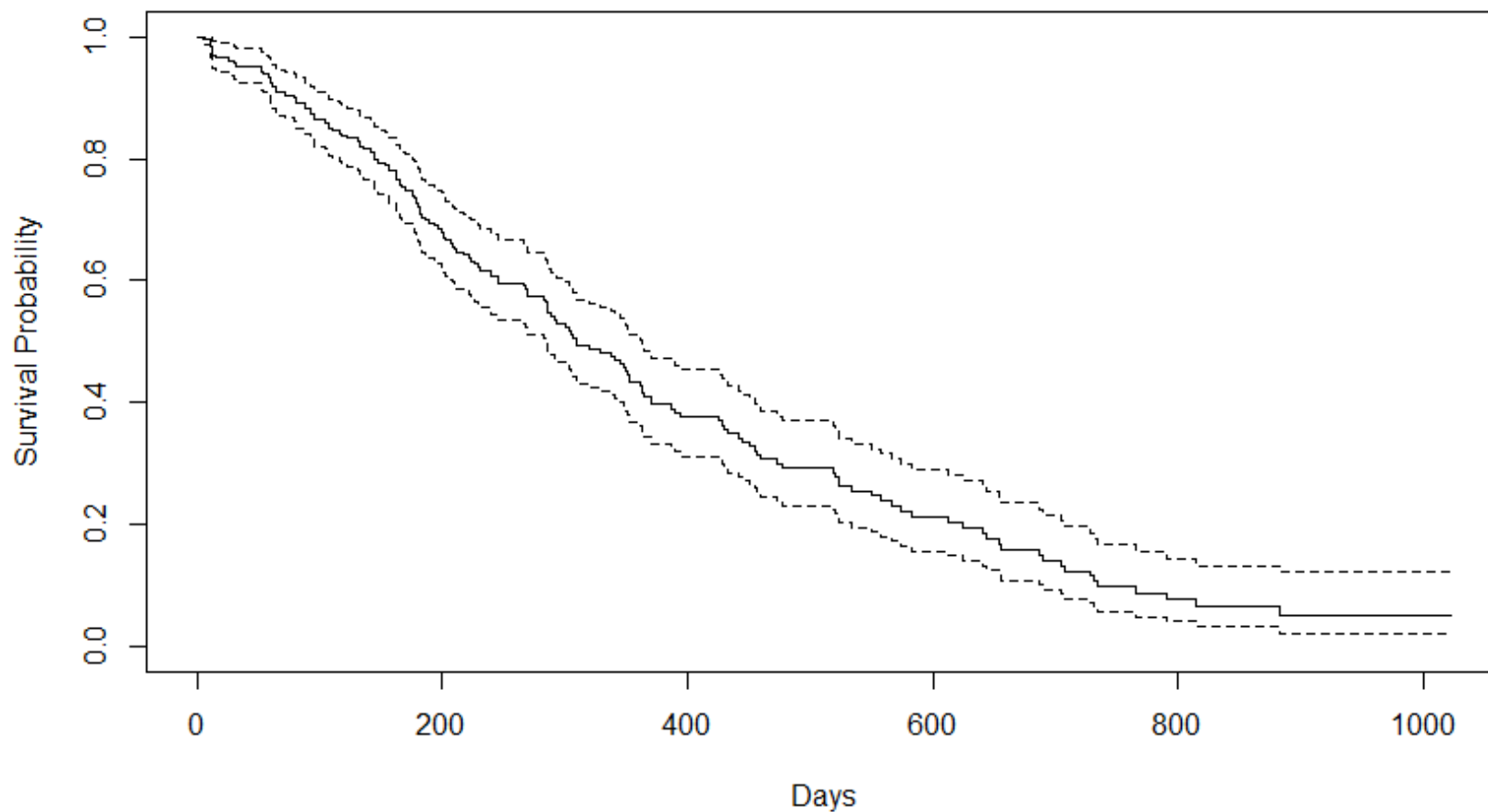
* restricted mean with upper limit = 1022

Investigate the survfit object (cont.)

- `str(surv_fit)`

```
List of 16
 $ n      : int 228
 $ time   : num [1:186] 5 11 12 13 15 26 30 31 53 54 ...
 $ n.risk  : num [1:186] 228 227 224 223 221 220 219 218 217 215 ...
 $ n.event : num [1:186] 1 3 1 2 1 1 1 1 2 1 ...
 $ n.censor : num [1:186] 0 0 0 0 0 0 0 0 0 0 ...
 $ surv    : num [1:186] 0.996 0.982 0.978 0.969 0.965 ...
 $ std.err  : num [1:186] 0.0044 0.00885 0.00992 0.01179 0.01263 ...
 $ cumhaz   : num [1:186] 0.00439 0.0176 0.02207 0.03103 0.03556 ...
 $ std.chaz : num [1:186] 0.00439 0.0088 0.00987 0.01173 0.01257 ...
 $ type     : chr "right"
 $ logse    : logi TRUE
 $ conf.int  : num 0.95
 $ conf.type: chr "log"
 $ lower     : num [1:186] 0.987 0.966 0.959 0.947 0.941 ...
 $ upper     : num [1:186] 1 1 0.997 0.992 0.989 ...
 $ call      : language survfit(formula = Surv(time = AVAL, event = CNSR == 0) ~ 1, data = ADTTE)
 - attr(*, "class")= chr "survfit"
```

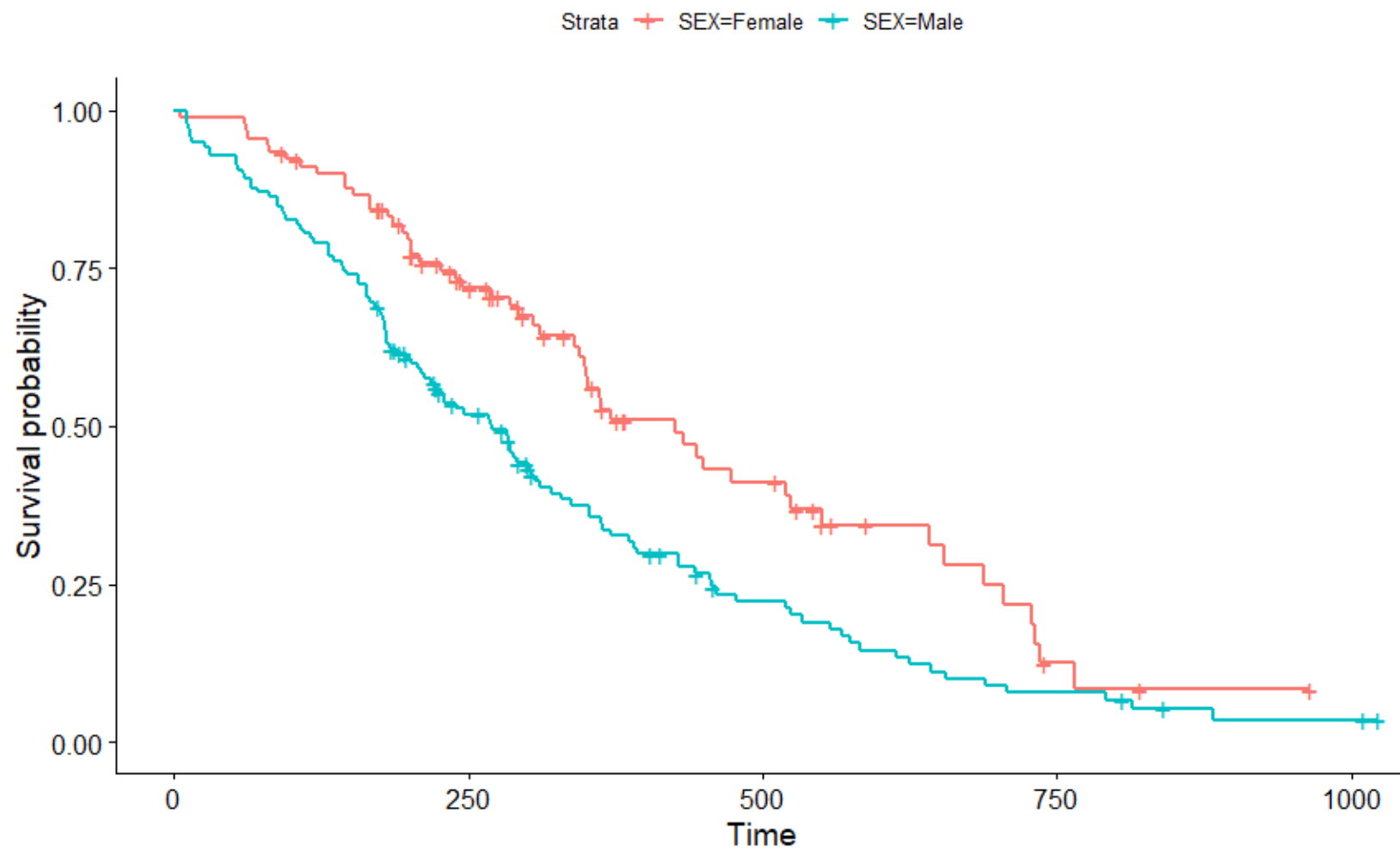
Kaplan-Meier Survival Curve



```
# Basic KM plot
plot(surv_fit,
     conf.int = TRUE,
     mark.time = FALSE,
     main= "Kaplan-Meier Survival Curve",
     xlab = "Days",
     ylab="Survival Probability"
)

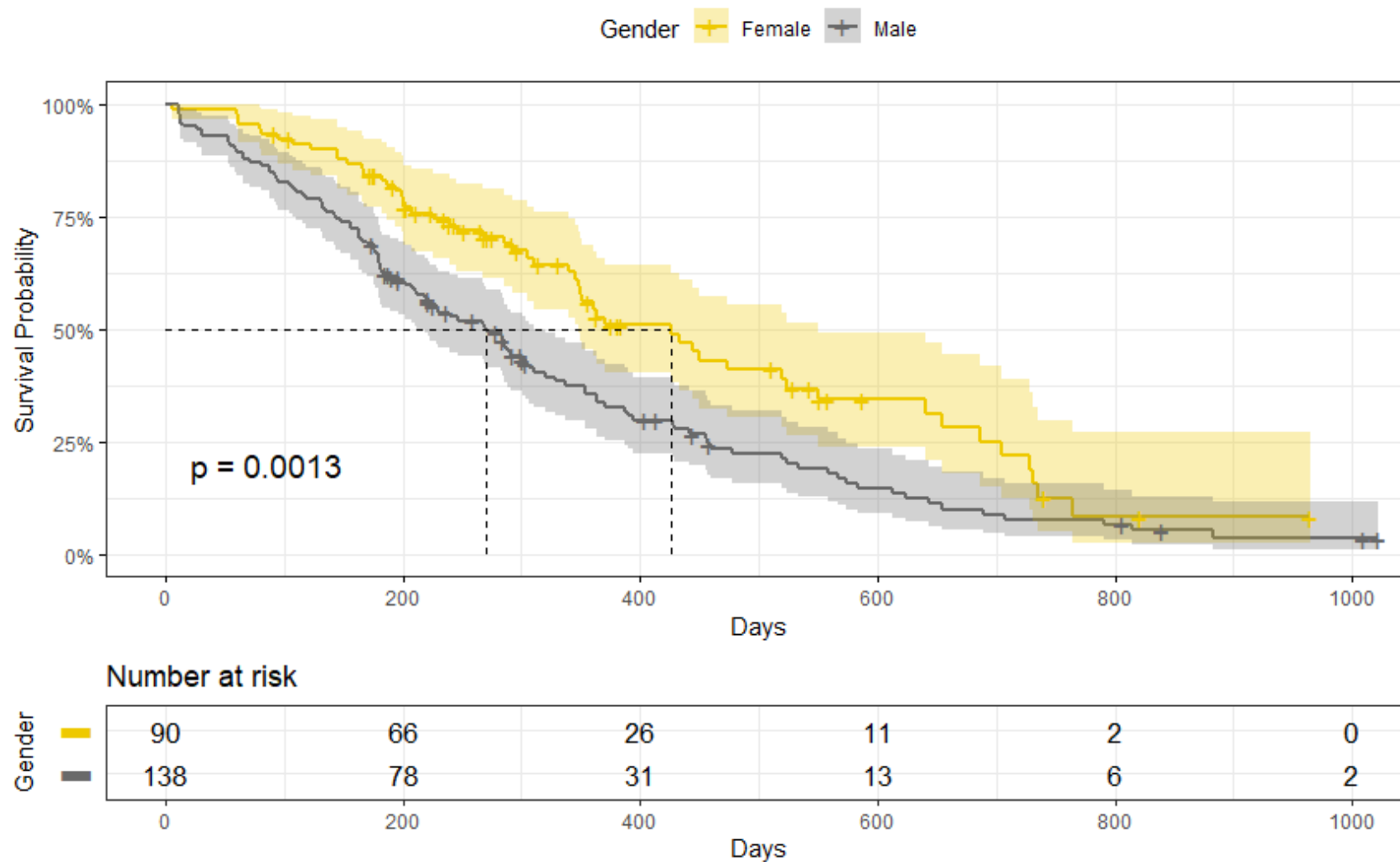
# survfit object which contains survival curves
# plots the confidence interval
# removes '+' sign for every event
# figure title
# x-axis label
# y-axis
```

Basic survminer KM plot



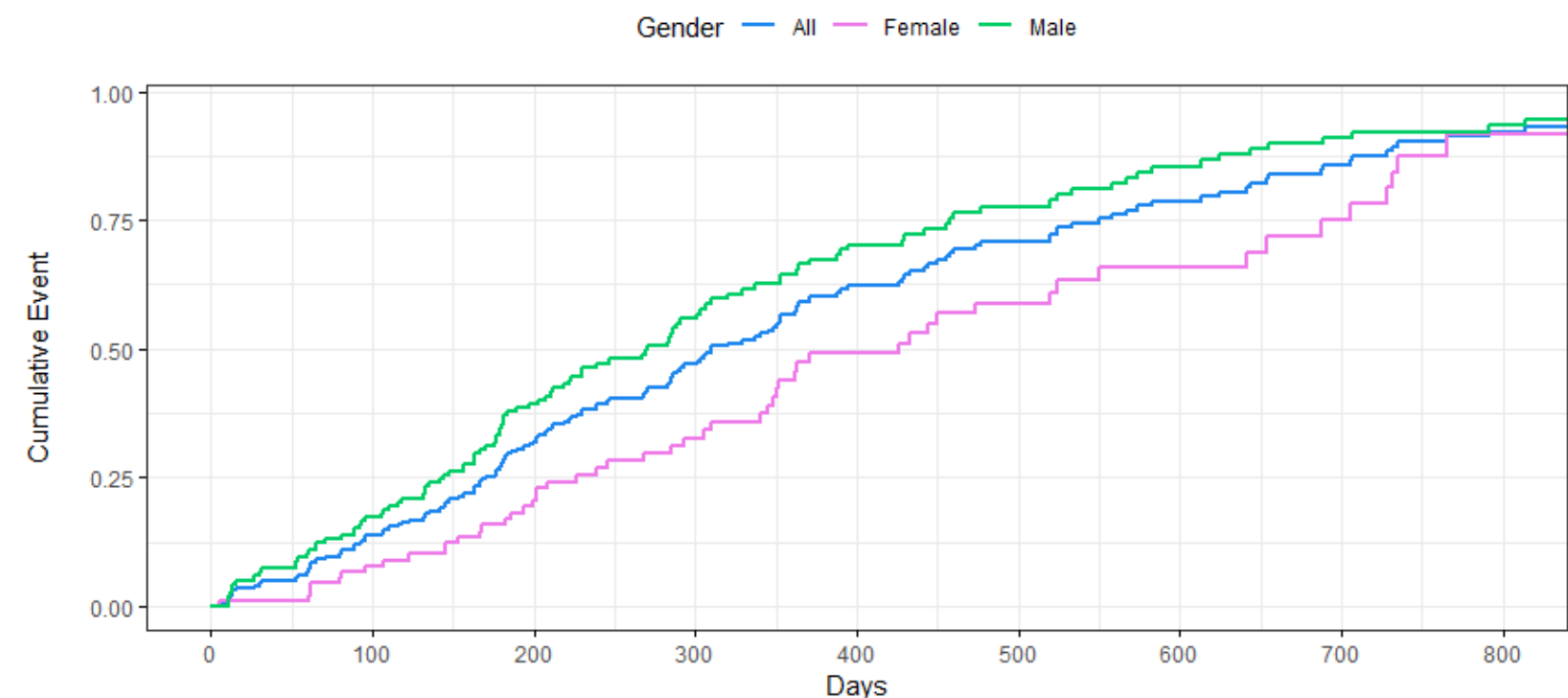
```
library(survminer)
surv_fit2 <- survfit(surv(time = AVAL, event = CNSR == 0) ~ SEX, data = ADTTE)
ggsurvplot(surv_fit2)
```

Advanced survminer KM plot



```
ggsurvplot(
  surv_fit2,
  risk.table = TRUE,
  conf.int = TRUE,
  pval = TRUE,
  surv.median.line = "hv",
  surv.scale = "percent",
  break.time.by = 200,
  risk.table.y.text.col = TRUE,
  risk.table.y.text = FALSE,
  xlab = "Days",
  ylab = "Survival Probability",
  legend.title = "Gender",
  legend.labs = c("Female", "Male"),
  palette = c("gold2", "grey41"),
  ggtheme = theme_bw()
)
```

KM plot of cumulative events



```
ggsurvplot(  
  surv_fit2,  
  fun = "event",  
  add.all = TRUE,  
  cumevents = TRUE,  
  censor = FALSE,  
  xlim = c(0,800),  
  break.time.by = 100,  
  xlab = "Days",  
  ylab= "Cumulative Event",  
  legend.title = "Gender",  
  legend.labs = c("All","Female", "Male"),  
  palette = c("dodgerblue2", "orchid2",  
              "springgreen3"),  
  ggtheme = theme_bw()  
)
```

Cumulative number of events

Gender	All	0	31	72	101	126	138	148	156	163
	Female	0	7	18	27	37	42	45	48	53
	Male	0	24	54	74	89	96	103	108	110
		0	100	200	300	400	500	600	700	800

R interactive graphs libraries



```
library(plotly)
```

- JavaScript
- free
- can convert ggplot2 to plotly

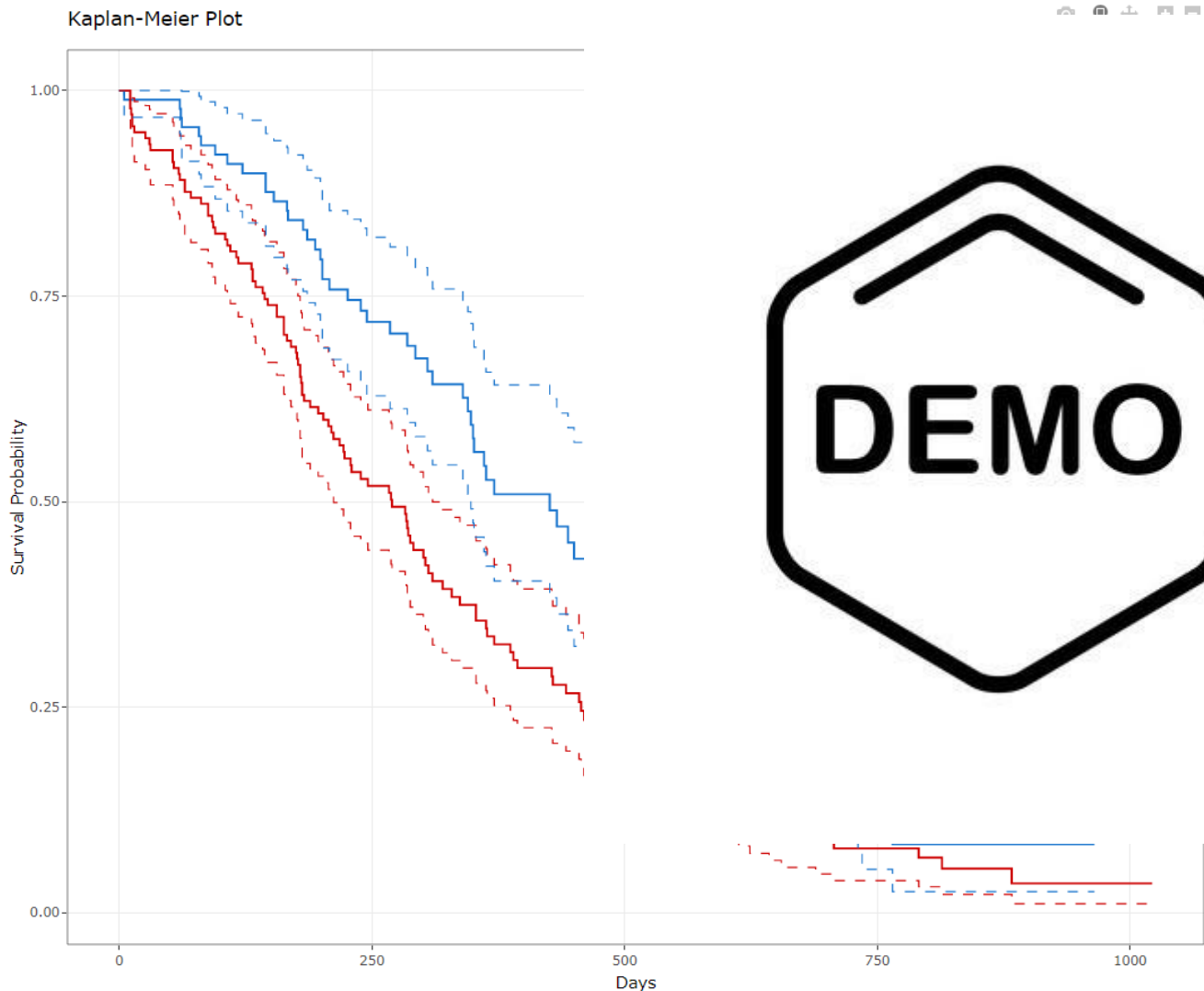


```
library(highcharter)
```

- JavaScript
- not free
- looks beautiful

Both inspired by well-known ggplot2 R package and the Grammar of Graphics concept.

Plotly interactive KM graph

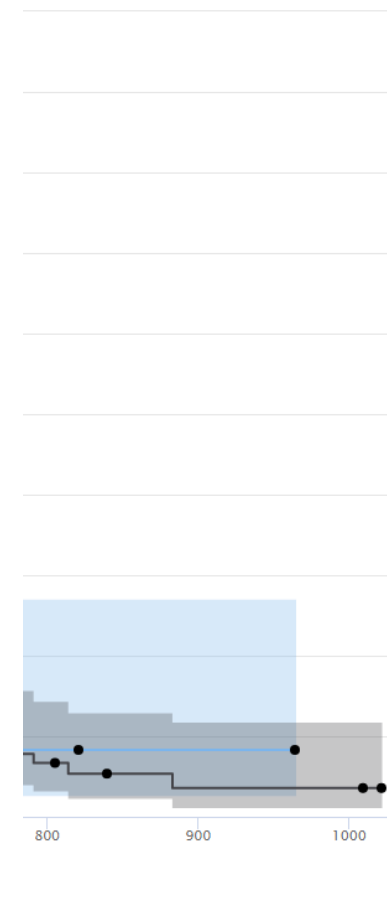
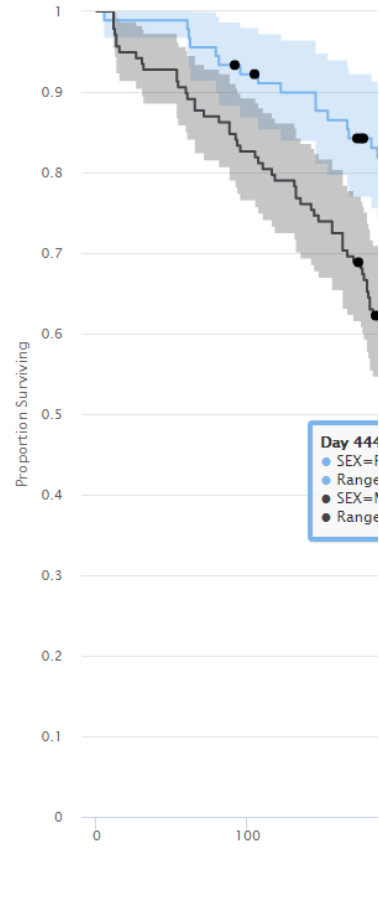


```
[plotly)
[GGally)

lot <- ggsurv(surv_fit2,
  CI = TRUE,
  plot.cens = FALSE,
  surv.col = c("dodgerblue3","red3"),
  size.ci = 0.3,
  xlab = "Days",
  ylab = "Survival Probability",
  main = "Kaplan-Meier Plot") +
  _bw()

/(int_kmplot)
```

Highcharts interactive graph



```
hchart(surv_fit2, ranges = TRUE,
       tooltip = list(headerFormat = "<b>Day {point.key}</b> <br>"))
```

PRESENTATION TAKEAWAYS

- ✓ Easy to create a KM plot in R
- ✓ R syntax is very straightforward
- ✓ Simple customization
- ✓ Interactive graphs
- ✓ More information
- ✓ Looks beautiful



THANK YOU

Oleksandr Babych
Intego Group LLC
Grzybowska 78, 00-844 Warsaw, Poland
Work Phone: +1 407.512.1006 (ext. 2449)
Email: oleksandr.babych@intego-group.com
Web: www.intego-group.com

