Perform contrasts in linear regression and survival analysis. Then moving on to examples of some KM plots

Eamonn O'Brien

03 September, 2016

1 Introduction

Showing how to perform contrasts in linear regression and survival analysis. Then moving on to examples of some KM plots

2 linear regression and contrasts (t dist)

```
n <- 200
intercept <- 100
noise <- 5
# Hyp the effect of age on outcome, a unit change in outcome over the whole age range
# so ~0.02 effect on outcome for each increase in age of 1 year
effect.of.age <- 1/(65-18)
effect.of.sex <- 5         # Hypothesize the effect of sex on outcome
effect.of.treatment <- 1   # Hypothesize the treatment effect on outcome

# Hypothesize baseline version of response on outcome,
# a unit increase results in 0.5 increase in outcome,
# strong predictor so 5 over the baseline range
effect.of.baseline <- 1/2
                                # random error

# covariate effects
baseline <- runif(n,100,110)                # baseline effect
treat <- 1*(runif(n)<0.5)                   # randomised treatment effect
fact <- c(rep("a", n/4), rep("b", n/4), rep("c", n/4), rep("d", n/4))  # Generate grp
age <- sample(x=18:65, size=n, replace=TRUE) # Generate age covariate

# create the outcome, with random error
 Y <- 2*(fact=="b")+  3*(fact=="c")+  4*(fact=="d") +
   effect.of.age*age + effect.of.baseline*baseline +
   effect.of.treatment*treat + rnorm(n, intercept, noise)

# prepare for analysis
d <- data.frame(Y=Y, baseline=baseline, treat=treat,
                fact=factor(fact),  age=age)

dd <- datadist(d, data=d)
options(datadist="dd")
```

3 my function for contrasts (t dist)

```
    lincom <- function (object, ref, comp) {

       var <- vcov(object, regcoef.only = TRUE, intercepts = "none")

       c1<-var[, grepl(comp, colnames(var))  ]
       var1<-c1[ grepl(comp, names(c1))  ]

       c1<-var[, grepl(ref, colnames(var))  ]
       var2<-c1[ grepl(ref, names(c1))  ]

       c1<-var[, grepl(comp, colnames(var))  ]
       cov<-c1[ grepl(ref, names(c1))  ]


       comp1<-object$coef[grepl(comp,   names(object$coef ))   ][[1]]
       ref1<- object$coef[grepl(ref,    names(object$coef ))   ][[1]]

       df <- qt(.975,  object$df.residual) # from harrell contrast.rms

       namez<-c("comp v", "ref", "est", "Lower95%CI", "Upper95%CI")
       res<-c(comp,ref,(comp1-ref1),
               (comp1-ref1+c(-1,1)*
                           df*sqrt(var1[[1]]+var2[[1]]-2*cov[[1]])))

              res <- c( res[1:2], sprintf(fmt="%.15s", res[3:5]))

       names(res)<-namez
       return(res)

}
```

4 variance covariance matrix

```
    f <- ols(Y ~ baseline + fact + age + treat, d)
    print(kable(vcov(f)))
```

	Intercept	baseline	fact=b	fact=c	fact=d	age	treat
Intercept	160.0253254	-1.5118198	0.2359462	-1.7434087	-0.3005340	-0.0200311	0.3129343
baseline	-1.5118198	0.0144645	-0.0082782	0.0105553	-0.0030402	-0.0000980	-0.0059740
fact=b	0.2359462	-0.0082782	1.0661274	0.5236146	0.5370853	0.0010321	0.1200462
fact=c	-1.7434087	0.0105553	0.5236146	1.0527267	0.5251232	0.0021944	0.0375927
fact=d	-0.3005340	-0.0030402	0.5370853	0.5251232	1.0478274	0.0013337	0.0753072
age	-0.0200311	-0.0000980	0.0010321	0.0021944	0.0013337	0.0007042	-0.0001280
treat	0.3129343	-0.0059740	0.1200462	0.0375927	0.0753072	-0.0001280	0.5336941

5 confidence intervals

```
d$fact <- relevel( d$fact, ref="a")           # no need for this first time
f <- ols(Y ~ baseline + fact + age + treat, d)
print(kable(confint(f)))
```

	2.5 %	97.5 %
Intercept	69.4505190	119.3509509
baseline	0.3194223	0.7938415
fact=b	-0.2892479	3.7837529
fact=c	3.5959314	7.6432534
fact=d	2.8102450	6.8481381
age	-0.0497585	0.0549194
treat	-0.5556698	2.3260792

6 my function, compare level c to level d

```
print(kable (lincom(f, ref="fact=d", comp= "fact=c") ) )
```

comp v	fact=c
ref	fact=d
est	0.7904008318357
Lower95%CI	-1.230933864824
Upper95%CI	2.8117355284956

7 relevel to get a contrast comparing level c to level d

```
d$fact <- relevel( d$fact, ref="d")
f <- ols(Y ~ baseline + fact + age + treat, d)
print(kable(confint(f)    ))
```

	2.5 %	97.5 %
Intercept	74.2449068	124.2149463
baseline	0.3194223	0.7938415
fact=a	-6.8481381	-2.8102450
fact=b	-5.0931219	-1.0707562
fact=c	-1.2309339	2.8117355
age	-0.0497585	0.0549194
treat	-0.5556698	2.3260792

8 Harrell's contrast function

```
    x <- contrast(f, list(fact="d"), list(fact="c")  )
    print(x, latex=TRUE, file='')
```

```
  baseline age treat   Contrast     S.E.     Lower    Upper     t Pr(>|t|)
1 105.0554  41     0 -0.7904008 1.024845 -2.811736 1.230934 -0.77   0.4415

Error d.f.= 193

Confidence intervals are 0.95 individual intervals
```

9 survival analysis and contrasts (z dist)

```
n <- 1000
set.seed(731)
age <- 50 + 12*rnorm(n)
label(age) <- "Age"
grp <- factor(sample(c('a','b','c','d'), n,
                     rep=TRUE, prob=c(.25, .25,.25,.25)))
cens <- 15*runif(n)

h <- .02*exp(.04*(age-50)+.4*(grp=='b')+.6*(grp=='c')+.8*(grp=='d'))

dt <- -log(runif(n))/h
label(dt) <- 'Follow-up Time'

e <- ifelse(dt <= cens,1,0)
dt <- pmin(dt, cens)
units(dt) <- "Year"
dd <- datadist(age, grp)
options(datadist='dd')

S <- Surv(dt,e)
f <- cph(S ~ (age) + grp, x=TRUE, y=TRUE)
# cox.zph(f, "rank")             # tests of PH
print(kable(anova(f)))
```

	Chi-Square	d.f.	P
age	87.18197	1	0.0000000
grp	17.47482	3	0.0005643
TOTAL	96.77207	4	0.0000000

```
#print(kable(f))
```

10 variance covariance matrix

```
print(kable(vcov(f)))
```

	age	grp=b	grp=c	grp=d
age	0.0000313	0.0001215	0.0001762	0.0001134
grp=b	0.0001215	0.0504216	0.0310630	0.0308016
grp=c	0.0001762	0.0310630	0.0467012	0.0311192
grp=d	0.0001134	0.0308016	0.0311192	0.0490332

11 Another function for survival analysis contrasts

```
lincom.cph <- function (object, ref, comp) {

  var <- vcov(object, regcoef.only = TRUE, intercepts = "none")

  c1<-var[, grepl(comp, colnames(var))  ]
  var1<-c1[ grepl(comp, names(c1))  ]

  c1<-var[, grepl(ref, colnames(var))  ]
  var2<-c1[ grepl(ref, names(c1))  ]

  c1<-var[, grepl(comp, colnames(var))  ]
  cov<-c1[ grepl(ref, names(c1))  ]


  comp1<-object$coef[grepl(comp,   names(object$coef ))   ][[1]]
  ref1<- object$coef[grepl(ref,    names(object$coef ))   ][[1]]

  namez<-c("comp v", "ref", "HR", "Lower95%CI", "Upper95%CI")
  res<-c (comp,ref,exp(comp1-ref1),
          exp(comp1-ref1+c(-1,1)*
                1.96*sqrt(var1[[1]]+var2[[1]]-2*cov[[1]])) )

  res <- c( res[1:2], sprintf(fmt="%.5s", res[3:5]))

  names(res)<-namez
  return(res)

}
```

12 my function, compare level c to level d

```
print(kable(lincom.cph(f, ref="grp=d", comp="grp=c")))
```

comp v	grp=c
ref	grp=d
HR	0.986
Lower95%CI	0.689
Upper95%CI	1.412

13 Harrell function, compare level c to level d

```
print(kable(summary(f, grp=c('d'), est.all=F) ))
```

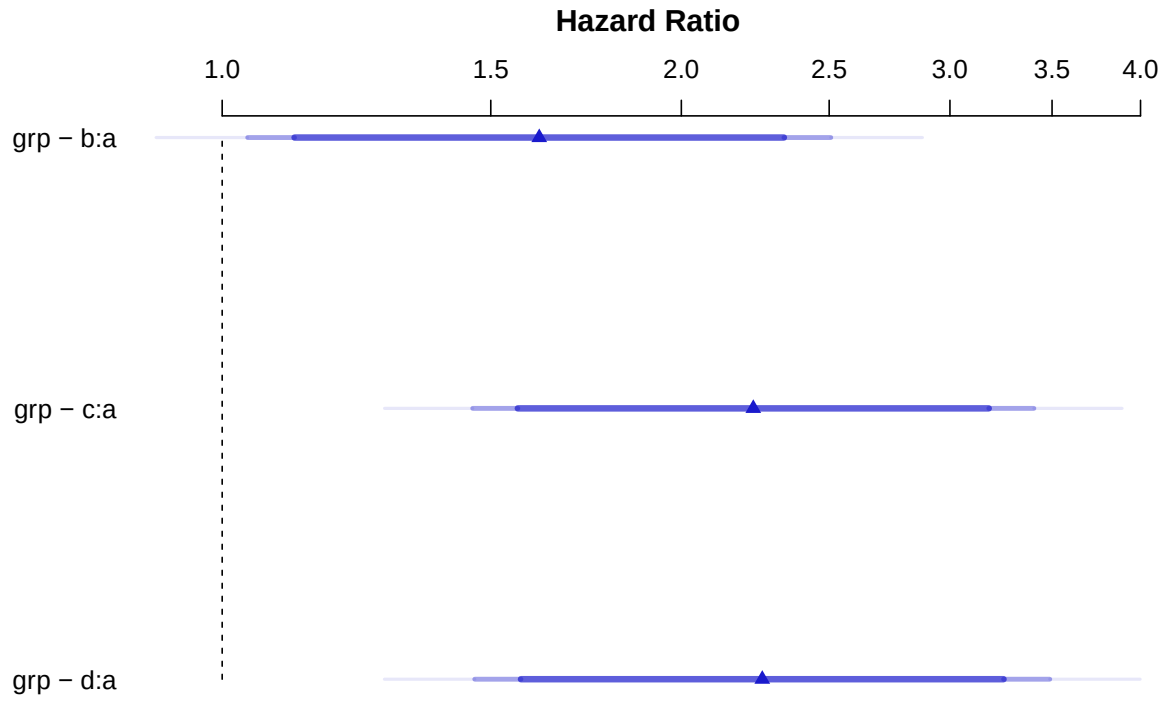
	Low	High	Diff.	Effect	S.E.	Lower 0.95	Upper 0.95	Type
grp - a:d	4	1	NA	-0.8153094	0.2214344	-1.2493128	-0.3813060	1
Hazard Ratio	4	1	NA	0.4425024	NA	0.2867017	0.6829689	2
grp - b:d	4	2	NA	-0.3366706	0.1945547	-0.7179908	0.0446497	1
Hazard Ratio	4	2	NA	0.7141441	NA	0.4877312	1.0456615	2
grp - c:d	4	3	NA	-0.0135435	0.1830191	-0.3722544	0.3451674	1
Hazard Ratio	4	3	NA	0.9865478	NA	0.6891789	1.4122263	2

14 Harrell again

```
      x <- contrast(f, list(grp="c"), list(grp="d")  )
      print(x, X=F, fun=exp  )
```

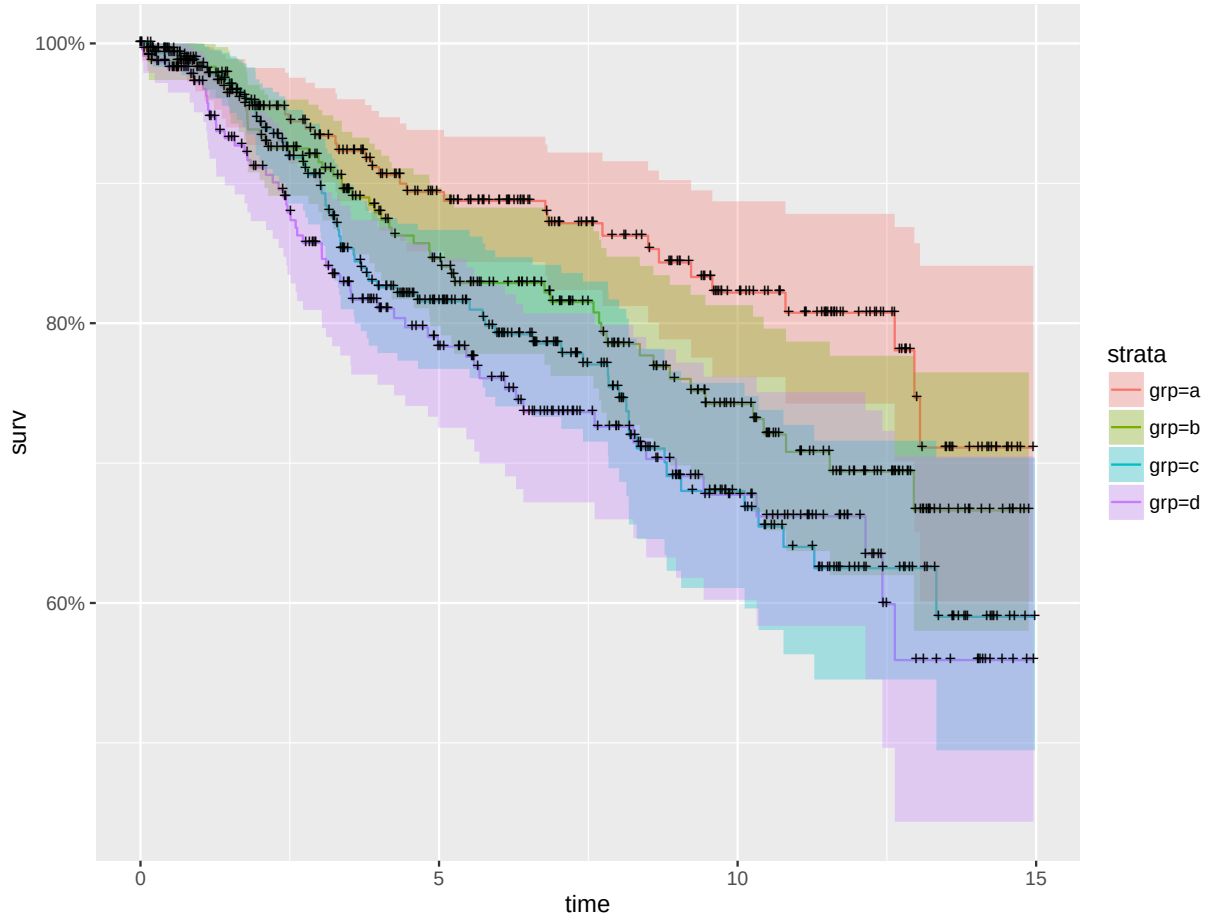
```
       age  Contrast S.E.     Lower    Upper     Z Pr(>|z|)
1 48.80065 0.9865478   NA 0.6891789 1.412226 -0.07    0.941

Confidence intervals are 0.95 individual intervals
```

```
      plot( summary(f, grp=c('a'), est.all=F), log=T )
```

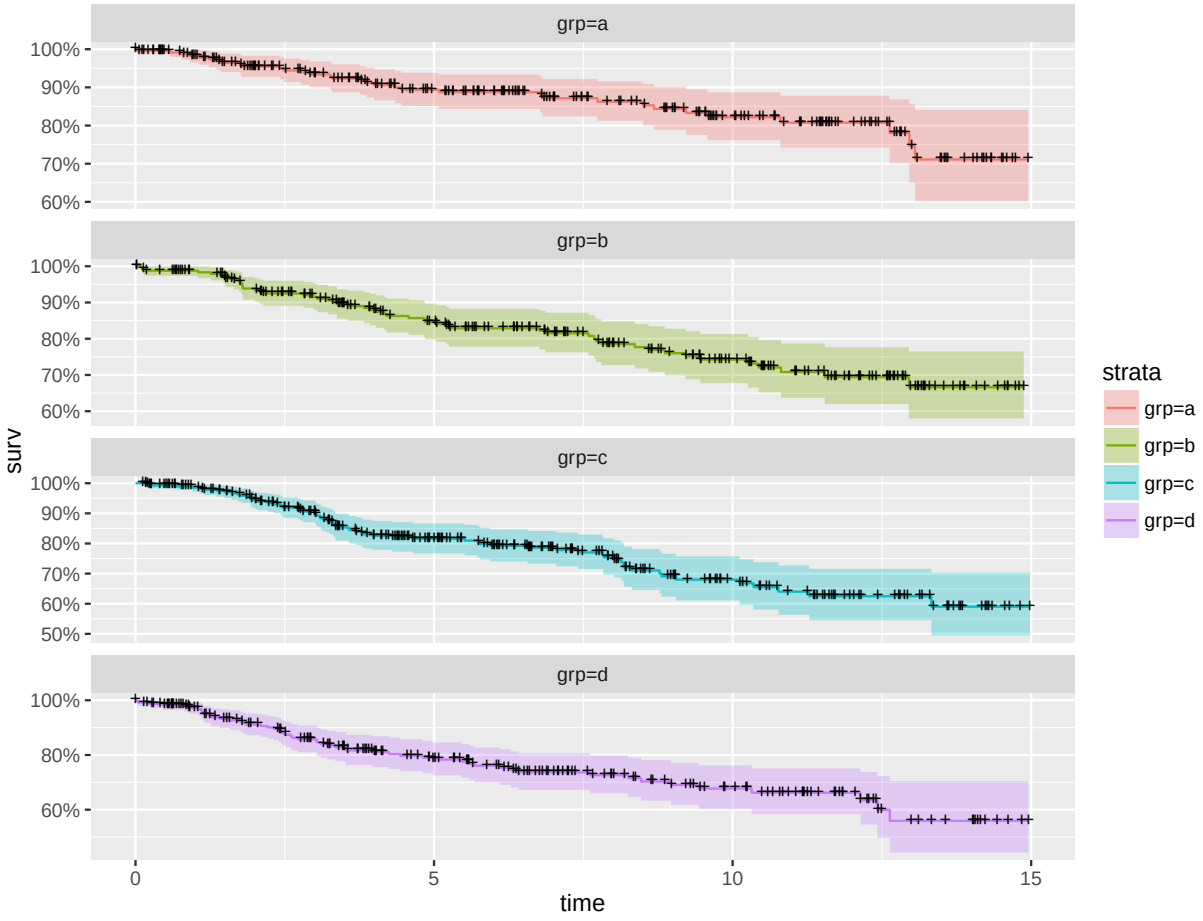


15 Explore simple survival plots

```
library(ggfortify)
library(survival)
f <- survfit(S ~   grp )
autoplot(f)
```

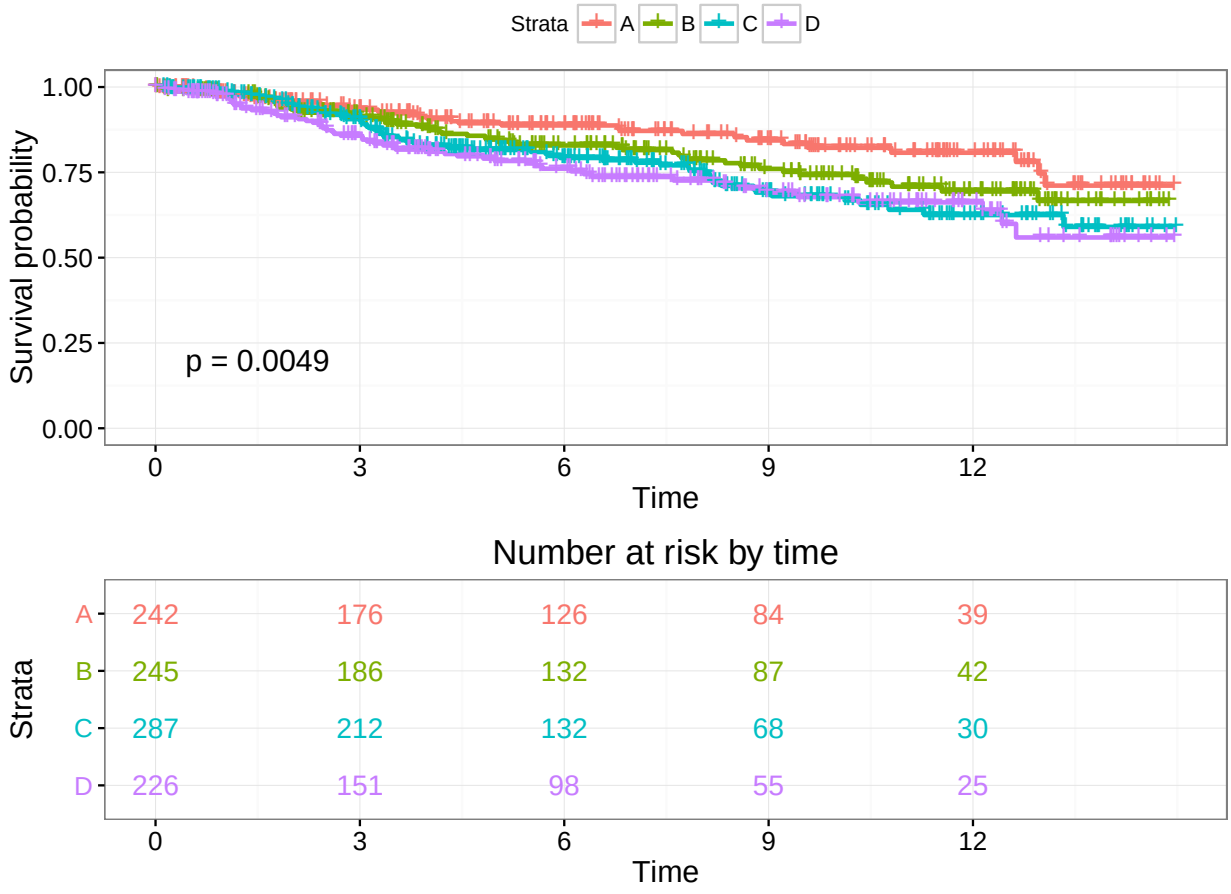



```
autoplot(f, facets = TRUE, nrow=4)
```


grp=a
grp=b
grp=c
grp=d
100%
90%
80%
70%
60%
50%
surv
0
5
10
15
time
strata
grp=a
grp=b
grp=c
grp=d

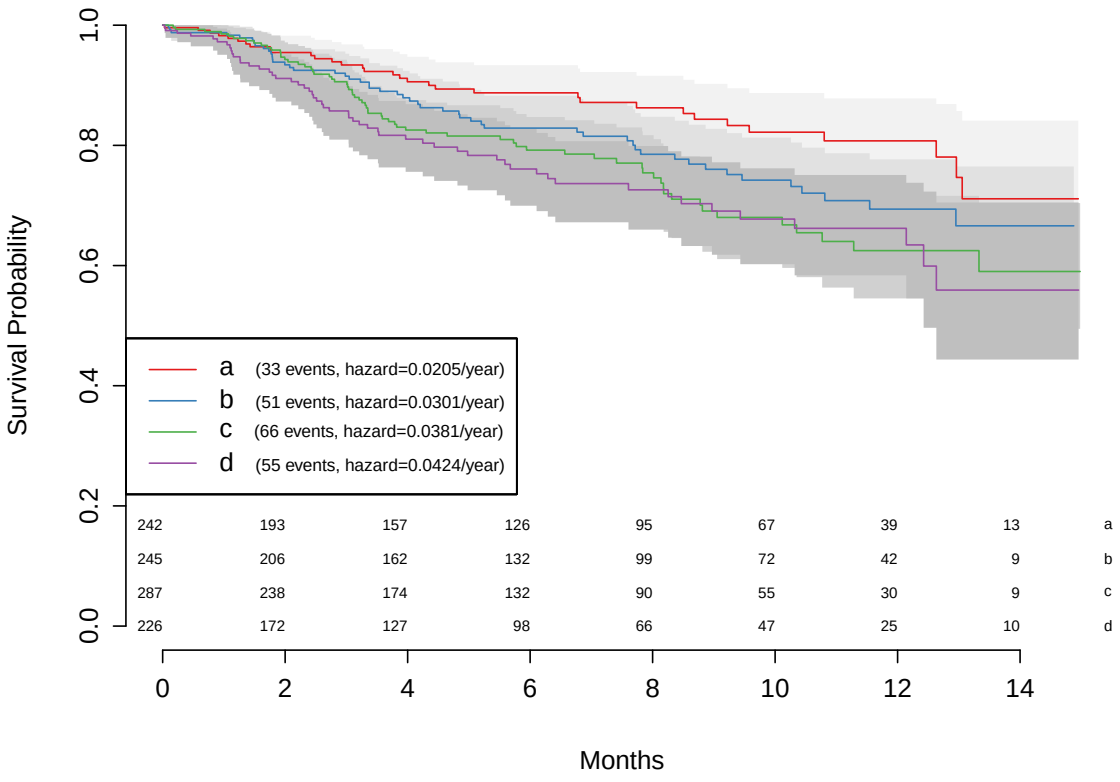
16 Explore simple survival plots

```
library("survminer")
ggsurvplot(f, pval = TRUE,
           break.time.by = 3,
           risk.table = TRUE,
           risk.table.col = "strata",
           risk.table.height = 0.5,
           ggtheme = theme_bw(),
           legend.labs = c("A", "B", "C", "D"))
```



17 Explore simple survival plots

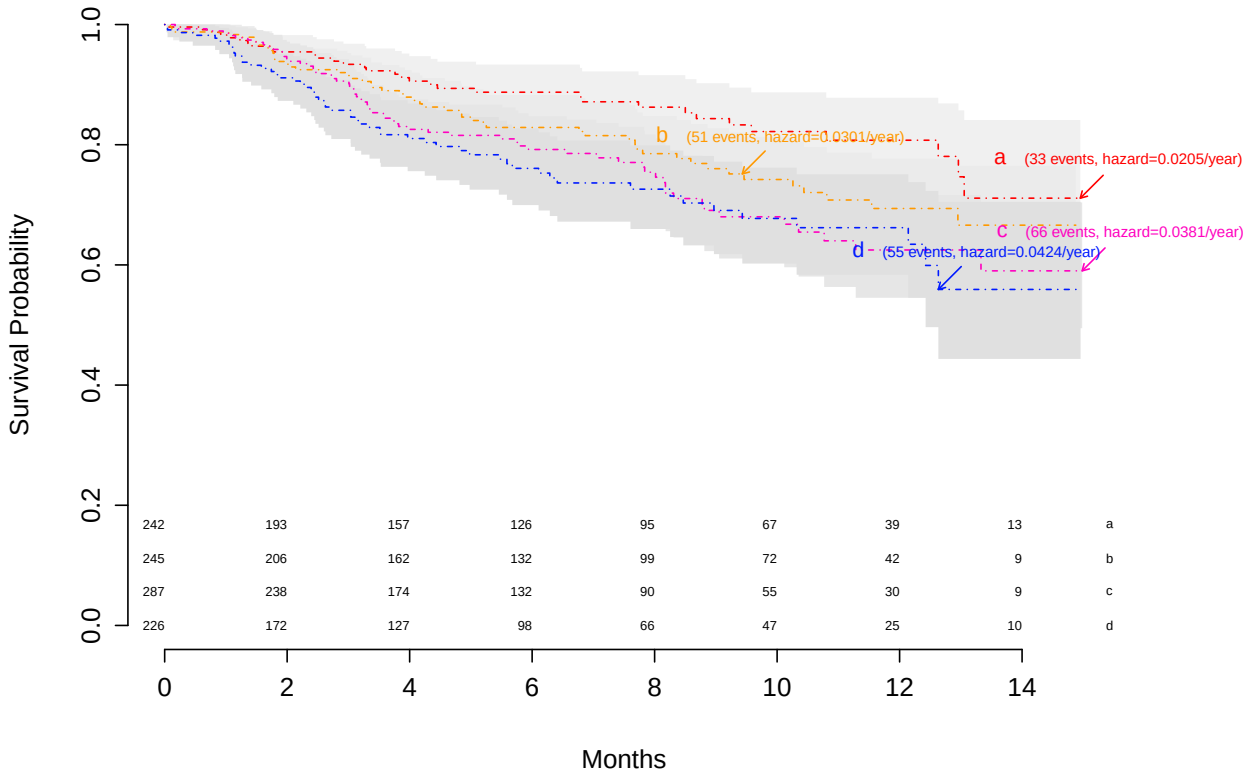
```
f2 <- npsurv(S ~ grp)

x <- c("#e41a1c","#377eb8","#4daf4a","#984ea3")

 #x <- rainbow(40)
 #x <- x[c(1,5,36,27)]


survplot(f2,  n.risk=TRUE, levels.only=T, conf.int=T,
         aehaz=TRUE,
         conf=c("bands"),
         col.fill=gray(seq(.95, .75, length=4)),
         #col.fill= c(rgb(0,0,0,0.1)),
         type=c("kaplan-meier"),
         lty=1,  col=x, xlab="Months", abbrev.label=T,
         label.curves = list(keys = "lines"), #bty='n',
         y.n.risk= 0, cex.n.risk=.6, time.inc=2)
```



```
      #lines(f2,  col= x)
```

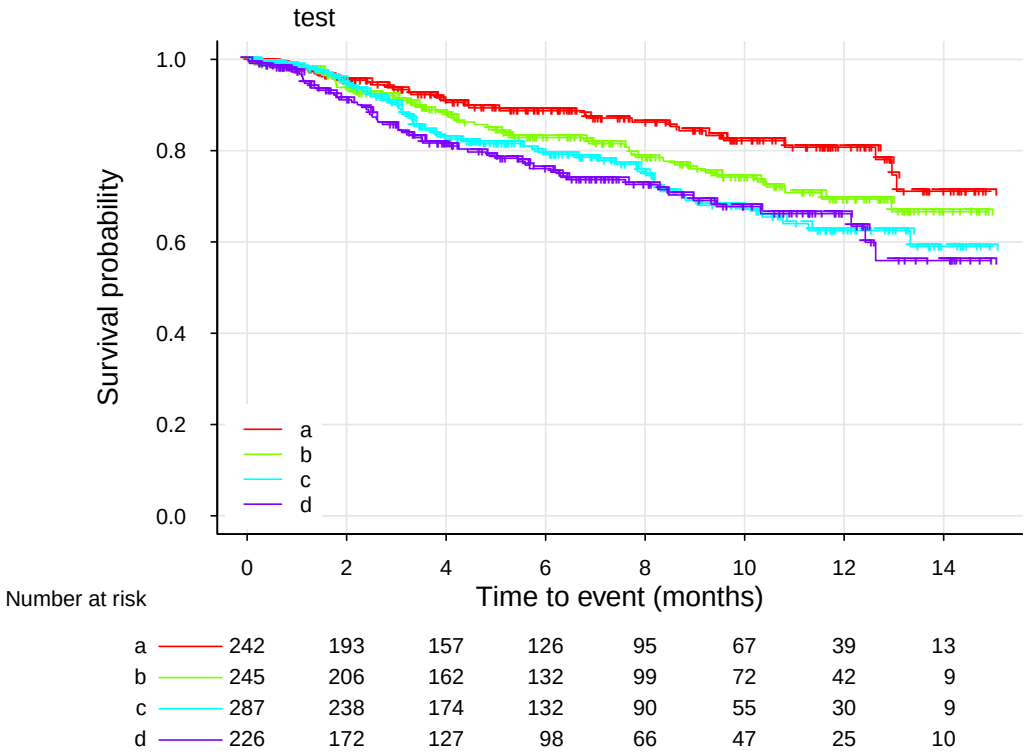
18 Explore simple survival plots

```
x <- rainbow(40)
x <- x[c(1,5,36,27)]

survplot(f2,  n.risk=TRUE, levels.only=T, conf.int=T,
         aehaz=TRUE,
         conf=c("bands"),
         col.fill=gray(seq(.94, .88, length=4)),
         #col.fill= c(rgb(0,0,0,0.1)),
         type=c("kaplan-meier"),
         lty=4,  col=x,
         xlab="Months", abbrev.label=T,
         label.curves=list(method="arrow", cex=.8),
         #label.curves=list(keys="lines"),
         y.n.risk= 0, cex.n.risk=.5, time.inc=2)
```



19 Explore simple survival plots

```
      setwd(wd3)
      source("Alternative KM plot.R", echo = F) # note this code is stored elsewhere
      setwd(wd)


kmplot(f2, mark='¬', simple=F,
  xaxis.at=seq(0,14,2),
  xaxis.lab=seq(0,14,2), # n.risk.at
  lty.surv=c(1,1), lwd.surv=1, col.surv=rainbow(4), # survival.curves
       lty.ci=0, lwd.ci=.7, col.ci=rainbow(4), #0 ci not plottrd
   group.names=c('a','b','c','d'),
    group.order=c(1,2,3,4), # order of appearance in the n.risk.at table and legend.
    extra.left.margin=6, label.n.at.risk=T, draw.lines=TRUE,
    cex.axis=0.8, xlab='Time to event (months)', ylab='Survival probability', # labels
    grid=TRUE, lty.grid=1, lwd.grid=1, col.grid=grey(.9),
    legend=T, loc.legend='bottomleft',
    cex.lab=1.1, xaxs='r', bty='L', las=1, tcl=-.2  # other parameters passed to plot()
)
title(main='test', adj=.1, font.main=1, line=0.5, cex.main=1)
```



```r
 # exponential distribution hazard rate estimates
print(f2$numevents/f2$exposure, digits=4)
```

```
  grp=a   grp=b   grp=c   grp=d
0.02050 0.03011 0.03811 0.04239
```

20 Computing Environment

```
R version 3.2.2 (2015-08-14)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 8 x64 (build 9200)

locale:
[1] LC_COLLATE=English_United Kingdom.1252
[2] LC_CTYPE=English_United Kingdom.1252
[3] LC_MONETARY=English_United Kingdom.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United Kingdom.1252

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets
[7] methods   base

other attached packages:
 [1] survminer_0.2.2    ggfortify_0.2.0    rethinking_1.58
 [4] rstan_2.11.1       StanHeaders_2.11.0 reshape_0.8.5
 [7] rms_4.5-0          SparseM_1.7        Hmisc_3.17-4
[10] ggplot2_2.1.0      Formula_1.2-1      survival_2.39-5
[13] lattice_0.20-33    knitr_1.14

loaded via a namespace (and not attached):
 [1] zoo_1.7-13          splines_3.2.2       colorspace_1.2-6
 [4] htmltools_0.3.5     stats4_3.2.2        loo_0.1.6
 [7] yaml_2.1.13         chron_2.3-47        DBI_0.5
[10] foreign_0.8-66      RColorBrewer_1.1-2  matrixStats_0.50.2
[13] multcomp_1.4-6      plyr_1.8.4          stringr_1.1.0
[16] MatrixModels_0.4-1  munsell_0.4.3       gtable_0.2.0
[19] mvtnorm_1.0-5       codetools_0.2-14    coda_0.18-1
[22] evaluate_0.9        latticeExtra_0.6-28 inline_0.3.14
[25] quantreg_5.26       TH.data_1.0-7       Rcpp_0.12.6
[28] acepack_1.3-3.3     scales_0.4.0        formatR_1.4
[31] gridExtra_2.2.1     digest_0.6.10       stringi_1.1.1
[34] polspline_1.1.12    dplyr_0.5.0         grid_3.2.2
[37] tools_3.2.2         sandwich_2.3-4      magrittr_1.5
[40] tibble_1.2          cluster_2.0.3       tidyr_0.6.0
[43] MASS_7.3-45         Matrix_1.2-2        data.table_1.9.6
[46] assertthat_0.1      rmarkdown_1.0       R6_2.1.3
[49] rpart_4.1-10        nnet_7.3-12         nlme_3.1-128
```

This took 6.11 seconds to execute.