Survival.RMD

Sarah Speroff

11/18/2021

Data prep

```
library(survival)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggfortify)
library(survminer)
```

```
## Loading required package: ggpubr

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```
library(RColorBrewer)
library(ggthemes)
```

```
setwd("/Users/speroffs/Desktop/A.Poculata-Multiple-stressors-thesis-analysis-/Survival")
```

```
# Load data for GRAPHICAL PARAMETERS
graph.dat <- read.csv("Full_ daily mortality tracker - Censored.CoxPH.w.controls.csv",
                      header = TRUE)

# Load data for MODEL CREATION (1 death added to each treatment to prevent infinite coeff)
surv.dat <- read.csv("2Full_ daily mortality tracker - Censored.CoxPH.w.controls.csv",
                      header = TRUE)
```

```
# relevel to compare all to control
surv.dat$Treatment <- factor(surv.dat$Treatment, levels = c("amb.ctrl", "PDS", "PLS", "NDS", "NLS", "PDI

graph.dat$Treatment <- factor(graph.dat$Treatment, levels = c("amb.ctrl", "PDS", "PLS", "NDS", "NLS", "I
```

```
# as factor for df 1
surv.dat$Plastic <- as.factor(surv.dat$Plastic)
surv.dat$Light <- as.factor(surv.dat$Light)
surv.dat$Food <- as.factor(surv.dat$Food)
surv.dat$Treatment <- as.factor(surv.dat$Treatment)
surv.dat$Sym.State <- as.factor(surv.dat$Sym.State)

# for df 2
graph.dat$Plastic <- as.factor(graph.dat$Plastic)
graph.dat$Light <- as.factor(graph.dat$Light)
graph.dat$Food <- as.factor(graph.dat$Food)
graph.dat$Treatment <- as.factor(graph.dat$Treatment)
graph.dat$Sym.State <- as.factor(graph.dat$Sym.State)
```

Initial Data visualization

```
# Create a survival model to assess mortality over time by treatment with modeling data
treat.mod <- coxph(Surv(Time, Death) ~Treatment, data=surv.dat)

# summarize the model
summary(treat.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Treatment, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                 coef exp(coef) se(coef)     z Pr(>|z|)
## TreatmentPDS  3.1032   22.2688   0.7747 4.006 6.18e-05 ***
```

```
## TreatmentPLS  2.9889   19.8639   0.7818 3.823 0.000132 ***
## TreatmentNDS  1.8373    6.2795   0.9129 2.013 0.044155 *
## TreatmentNLS  1.8067    6.0901   0.9129 1.979 0.047806 *
## TreatmentPDF  2.8564   17.3992   0.7906 3.613 0.000303 ***
## TreatmentPLF  3.5334   34.2409   0.7561 4.674 2.96e-06 ***
## TreatmentNDF  3.9850   53.7854   0.7712 5.167 2.37e-07 ***
## TreatmentNLF  3.7926   44.3700   0.7874 4.817 1.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## TreatmentPDS    22.269    0.04491     4.878    101.65
## TreatmentPLS    19.864    0.05034     4.291     91.95
## TreatmentNDS     6.279    0.15925     1.049     37.58
## TreatmentNLS     6.090    0.16420     1.018     36.45
## TreatmentPDF    17.399    0.05747     3.695     81.94
## TreatmentPLF    34.241    0.02920     7.780    150.69
## TreatmentNDF    53.785    0.01859    11.864    243.84
## TreatmentNLF    44.370    0.02254     9.482    207.63
##
## Concordance= 0.781  (se = 0.02 )
## Likelihood ratio test= 79.43  on 8 df,   p=6e-14
## Wald test            = 42.75  on 8 df,   p=1e-06
## Score (logrank) test = 76.68  on 8 df,   p=2e-13
```

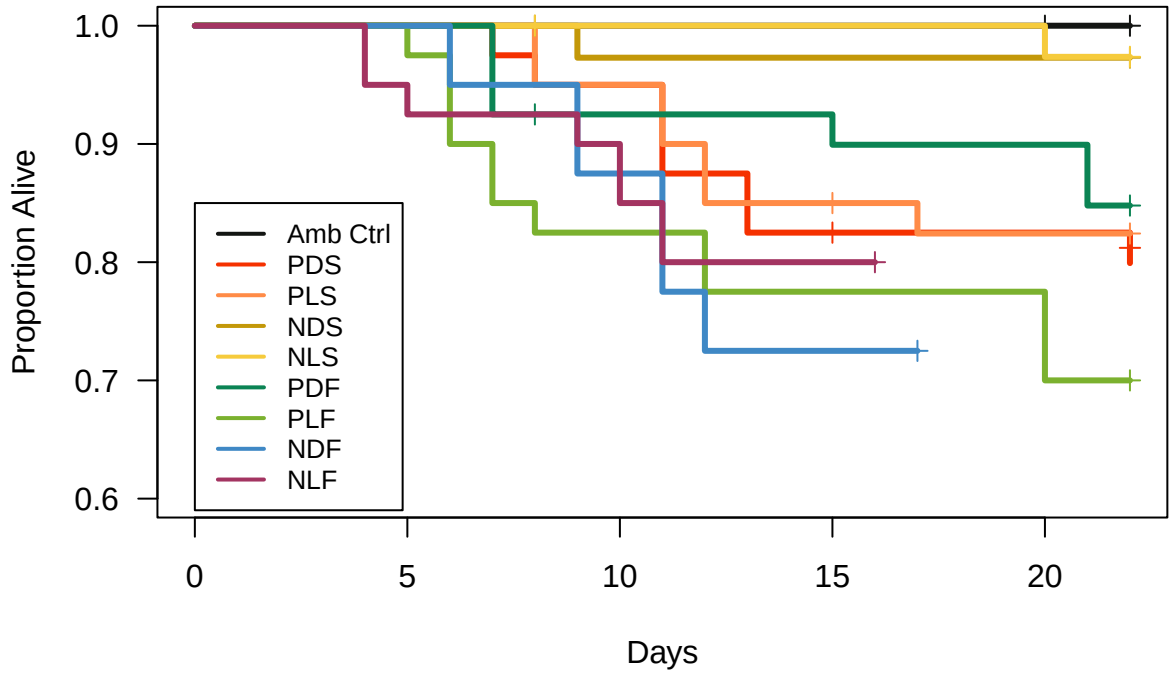
```
# Perform Log-rank test on our treat.mod
treat_diff <- survdiff(Surv(Time, Death) ~ Treatment, data=surv.dat)
treat_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Treatment, data = surv.dat)
##
##                       N Observed Expected (O-E)^2/E (O-E)^2/V
## Treatment=amb.ctrl 162        2    26.70    22.848    37.169
## Treatment=PDS       42       10     6.21     2.313     2.573
## Treatment=PLS       42        9     6.24     1.220     1.357
## Treatment=NDS       42        3     6.45     1.845     2.062
## Treatment=NLS       42        3     6.65     2.000     2.242
## Treatment=PDF       42        8     6.26     0.481     0.535
## Treatment=PLF       42       14     5.66    12.277    13.539
## Treatment=NDF       42       13     4.05    19.791    21.723
## Treatment=NLF       42       10     3.78    10.237    11.228
##
##  Chisq= 76.4  on 8 degrees of freedom, p= 3e-13
```

```
# make color palette to plot survival function
new.pal <- c("#131513", # control
             "#F53100", # pds
             "#FF8B47", # pls
             "#C39809", # nds
             "#F6CB3C", # nls
             "#0B8454", # pdf
             "#7BB12F", # plf
```

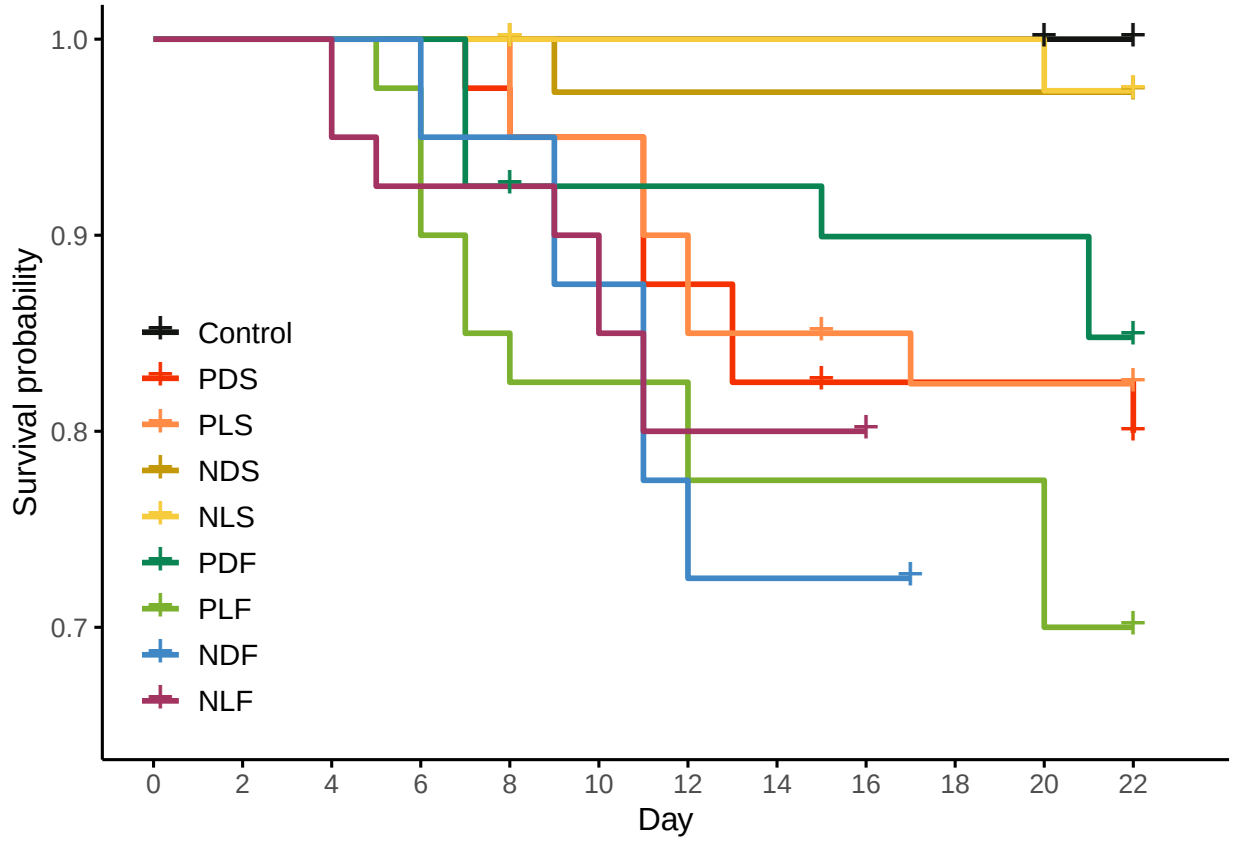
```
              "#3F88C5", # ndf
              "#A33461") # nlf
```

```
# write the same model using the graphing data, and plot that
treat <- survfit(Surv(Time, Death) ~Treatment, data=graph.dat)

plot(treat, conf.int=FALSE,
     xlab = "Days",
     xlim = c(0,22),
     ylab = "Proportion Alive",
     ylim = c(0.6,1.0),
     mark.time=T,
     pch = 3,
     las=1, lwd = 3,
     col=new.pal)
legend(0,0.85, legend=c("Amb Ctrl", "PDS", "PLS", "NDS", "NLS", "PDF", "PLF", "NDF", "NLF"),
       lty=1, lwd=2, cex=0.8,col=new.pal)
```




```
# Same plot using the ggplot function
Km.surv.fig <- ggsurvplot(treat, # survival function
                conf.int = FALSE, # remove confidence intervals
                pval = FALSE, # do not include p value on plot
                surv.scale = "default", # or %
                legend.labs = c("Control", "PDS", "PLS", "NDS", "NLS", "PDF", "PLF", "NDF", "NLF"),
                legend = c(0.1, 0.35), font.legend = 11, legend.title ="", # legend parameters
```

```
                xlab = "Day", xlim = c(0, 23), break.x.by = 2, font.x = 12, # x axis parameters
                ylim = c(0.65,1), font.y= 12, # y axis parameters
                censor.size = 6,
                ggtheme = theme_classic2(),
                # title = "A.",
                palette = c("#131513","#F53100", "#FF8B47","#C39809","#F6CB3C",
                            "#0B8454", "#7BB12F","#3F88C5","#A33461"))
Km.surv.fig
```



Single variable Hazards

PLASTIC

```
# First we will compute *univariate* models for each predictor variable to see how the impact survival
```

```r
# Plastic
plastic.mod <- coxph(Surv(Time, Death) ~ Plastic, data=surv.dat)
```

```r
summary(plastic.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                  coef exp(coef) se(coef)     z Pr(>|z|)
## PlasticPlastic 0.9583    2.6074   0.2381 4.024 5.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic     2.607     0.3835     1.635     4.158
##
## Concordance= 0.616  (se = 0.029 )
## Likelihood ratio test= 16.3  on 1 df,   p=5e-05
## Wald test            = 16.2  on 1 df,   p=6e-05
## Score (logrank) test = 17.47  on 1 df,   p=3e-05
```

```r
# HR plastic:no plastic (3.2244)
# plastic increased the hazard by 222.4%
#   1-exp(coef)
#   1-3.2244 = -2.2244 * 100 = 222.44
# being in no plastic reduced the hazard by 69%
#   1-exp(-coef)
#   1-0.3101 = 0.69 * 100 = 69
```

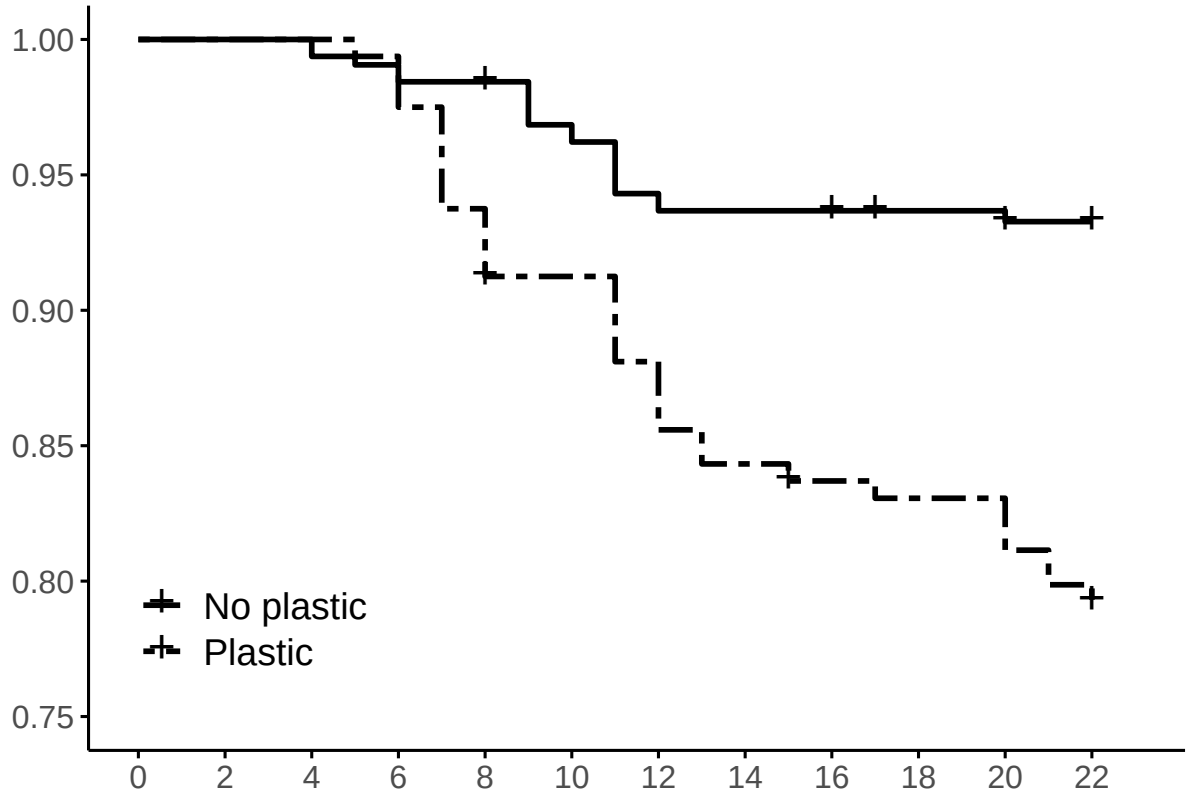
```r
# Perform Log-rank test on our plastic.mod
plastic_diff <- survdiff(Surv(Time, Death) ~ Plastic, data=surv.dat)
plastic_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Plastic, data = surv.dat)
##
##                      N Observed Expected (O-E)^2/E (O-E)^2/V
## Plastic=NoPlastic 330       31     47.6       5.8      17.4
## Plastic=Plastic   168       41     24.4      11.3      17.4
##
##  Chisq= 17.4  on 1 degrees of freedom, p= 3e-05
```

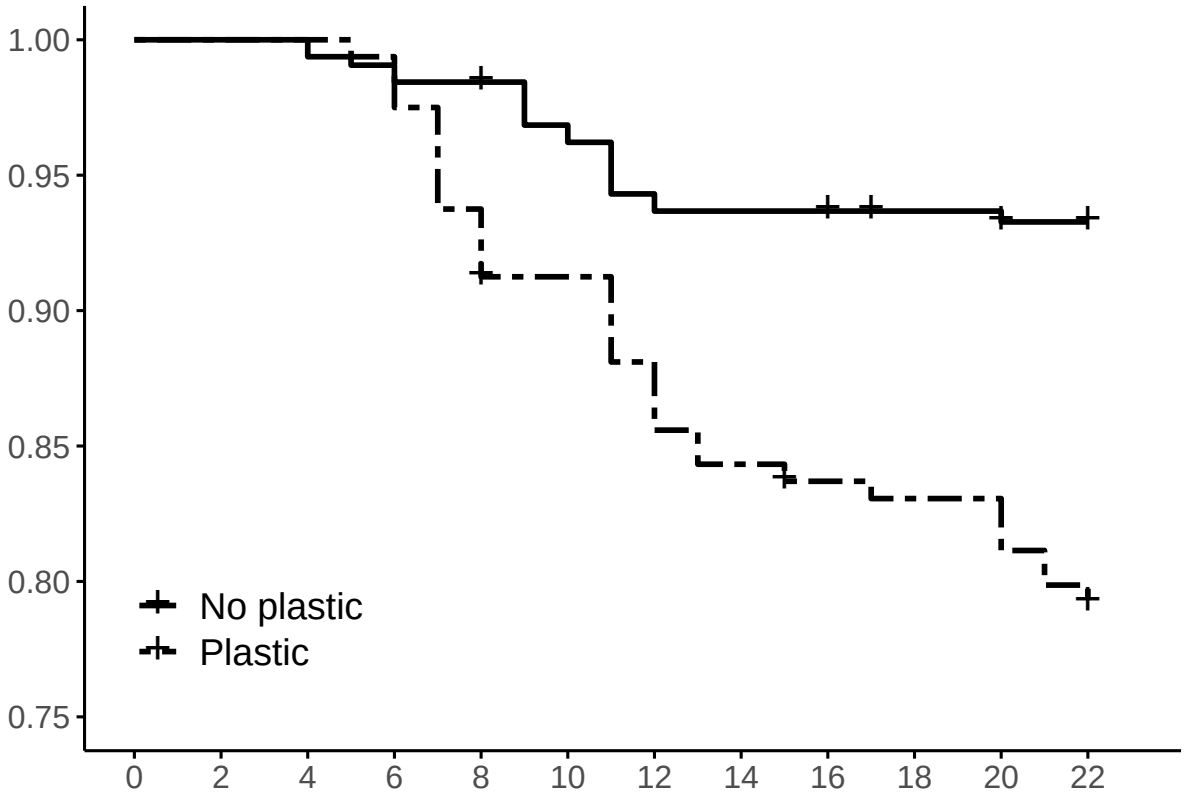
```r
# X^2 = 19.7
# df = 1
# p = 9x10^-6
```

```r
# rewrite the model with graphing data, and plot the kaplan meier curve for pastic
plastic <- survfit(Surv(Time, Death) ~ Plastic, data = graph.dat)
```

```
km.plastic.fig <-
  ggsurvplot(plastic,
          conf.int = FALSE, pval = FALSE, surv.scale = "default", # or %
           legend.labs = c("No plastic", "Plastic"),
           legend.title = "", legend = c(0.15, 0.2), font.legend = 14,
           xlab = "", xlim = c(0, 23),break.x.by = 2, font.x = 14,
           ylab = "", ylim = c(0.75,1),font.y= 14,
           font.tickslab = 12,
           censor.size = 6,
           ggtheme = theme_classic2(),
           linetype = c(1,6),
           palette = c("black","black"))

km.plastic.fig
```




```
#Save plot
ggsave(file = "/Users/speroffs/Desktop/km.plastic.fig.jpg",
       print(km.plastic.fig),
       width = 8, height = 6, units = "in", dpi=300)
```



FOOD

```
# Food
food.mod <- coxph(Surv(Time, Death) ~ Food, data=surv.dat)
```

```
summary(food.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Food, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                 coef exp(coef) se(coef)     z Pr(>|z|)
## FoodStarved -0.02979   0.97065  0.24781 -0.12    0.904
##
##             exp(coef) exp(-coef) lower .95 upper .95
## FoodStarved    0.9706       1.03    0.5972     1.578
##
## Concordance= 0.508  (se = 0.028 )
## Likelihood ratio test= 0.01  on 1 df,   p=0.9
## Wald test            = 0.01  on 1 df,   p=0.9
## Score (logrank) test = 0.01  on 1 df,   p=0.9
```

```r
# HR starved:fed <- (0.8864)
# being starved reduced the hazard by 11.36%
#   1-exp(coef)
#   1-0.8864 = 0.1136 * 100 = 11.36
# being fed increased the hazard by 12.8%
#   1-exp(-coef)
#   1-1.128 = -0.128 * 100 = -12.8
```

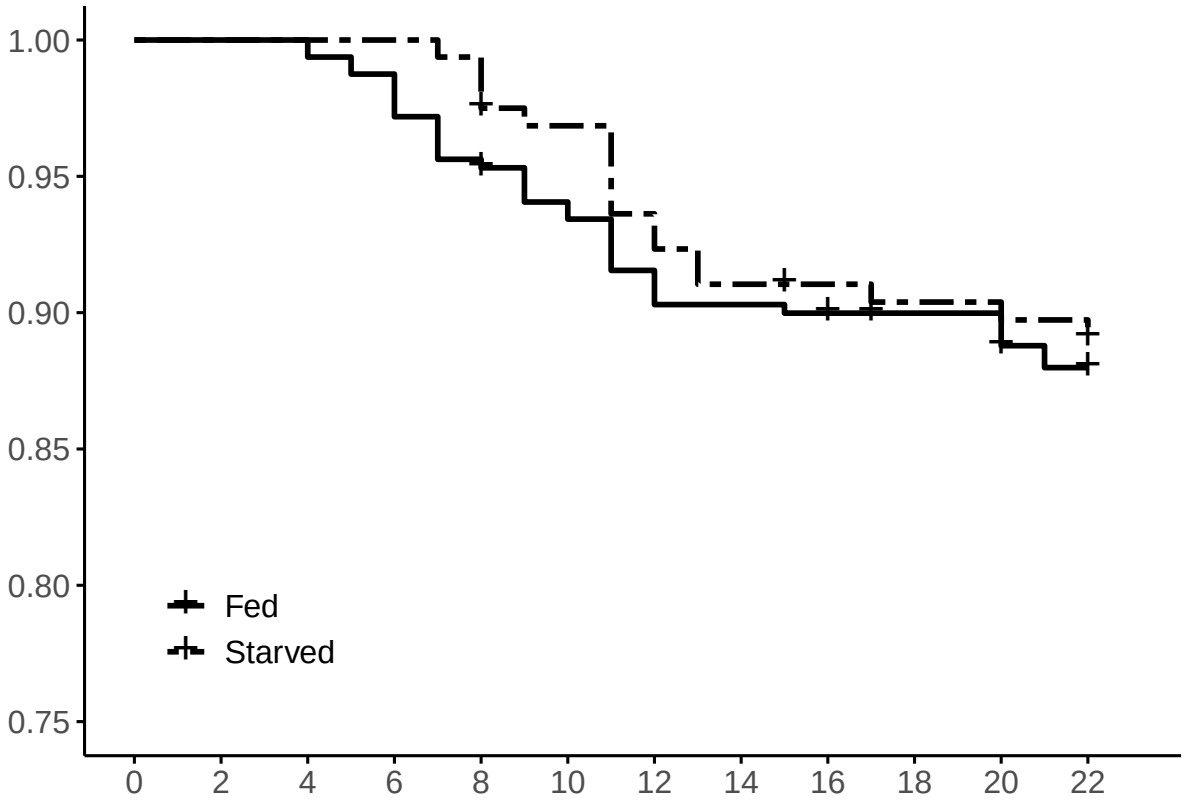
```r
# Perform Log-rank test on our foodf.mod
food_diff <- survdiff(Surv(Time, Death) ~ Food, data=surv.dat)
food_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Food, data = surv.dat)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## Food=Fed      330       47     46.5   0.00643    0.0185
## Food=Starved 168       25     25.5   0.01170    0.0185
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

```r
# X^2 = 0.2
# df = 1
# p = 0.7
```

```r
# kaplan meier curve
food <- survfit(Surv(Time, Death) ~ Food, data = graph.dat)
```

```r
# plot
km.food.fig <-
  ggsurvplot(food,
          conf.int = FALSE, pval = FALSE, surv.scale = "default", # or %
           legend.labs = c("Fed", "Starved"), legend.title = "", legend = c(0.15, 0.2), font.legend = 1
           xlab = "", xlim = c(0, 23),break.x.by = 2, font.x = 12,
           ylab = "", ylim = c(0.75,1),font.y= 14,
           font.tickslab = 12,
           censor.size = 6,
           ggtheme = theme_classic2(),
           linetype = c(1,6),
           palette = c("black","black"))

km.food.fig
```



FOOD: Fed vs Starved in No Plastic / Light treatment

```
# Fed vs Starved: for **NoPlastic/Light**

# filter out observations
# New df with just Plastic/Dark/Fed variables
FS1.dat <- filter(surv.dat, Treatment == "NLF" | Treatment == "NLS")
```

```
# cox model for fs1
fs1.coxm <- coxph(Surv(Time, Death) ~ Food, data=FS1.dat)
```

```
## Warning in coxph.fit(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 1 ; coefficient may be infinite.
```

```
summary(fs1.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Food, data = FS1.dat)
##
##   n= 84, number of events= 13
##
##                  coef  exp(coef)   se(coef)      z Pr(>|z|)
```

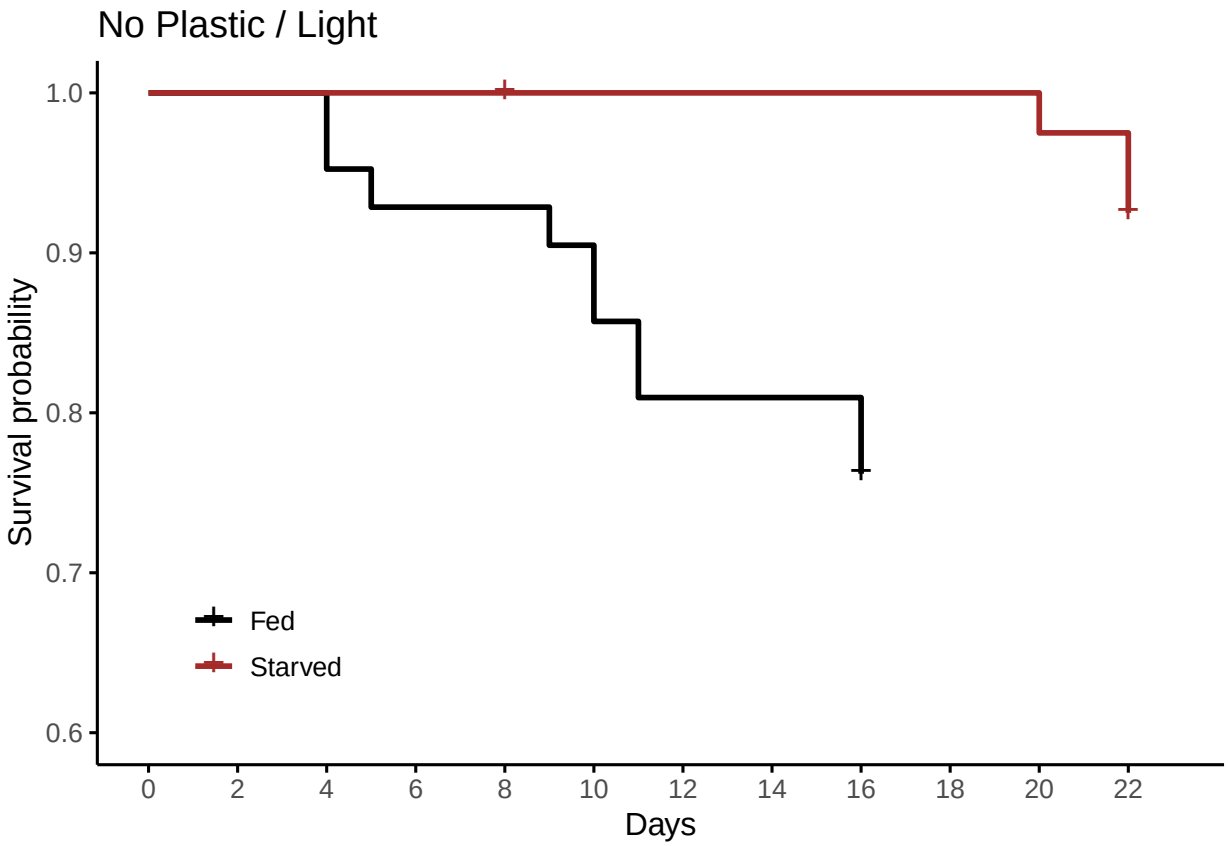
```
## FoodStarved -2.029e+01  1.542e-09  7.722e+03 -0.003    0.998
##
##             exp(coef) exp(-coef) lower .95 upper .95
## FoodStarved 1.542e-09  648562506         0       Inf
##
## Concordance= 0.731  (se = 0.024 )
## Likelihood ratio test= 14.71  on 1 df,   p=1e-04
## Wald test            = 0  on 1 df,   p=1
## Score (logrank) test = 10.87  on 1 df,   p=0.001
```

```
# Perform Log-rank test
fs1_diff <- survdiff(Surv(Time, Death) ~ Food, data=FS1.dat)
fs1_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Food, data = FS1.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Food=Fed      42       10     4.82      5.56      10.9
## Food=Starved 42        3     8.18      3.28      10.9
##
##  Chisq= 10.9  on 1 degrees of freedom, p= 0.001
```

```
# surv fit to plot
fs1 <- survfit(Surv(Time, Death) ~ Food, data = FS1.dat)
```

```
# plot
fs1.plot <- ggsurvplot(fs1, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Fed", "Starved"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "brown"),
           title = "No Plastic / Light")

fs1.plot
```



FOOD: Fed vs Starved in Plastic / Dark treatment

```
# Fed vs Starved: for **Plastic / Dark **

# filter out observations
# New df with just Plastic/Dark/Fed variables
FS2.dat <- filter(surv.dat, Treatment == "PDF" | Treatment == "PDS")
```

```
# cox model for fs2
fs2.coxm <- coxph(Surv(Time, Death) ~ Food, data=FS2.dat)
```

```
summary(fs2.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Food, data = FS2.dat)
##
##   n= 84, number of events= 18
##
##                coef exp(coef) se(coef)     z Pr(>|z|)
## FoodStarved 0.2407    1.2721   0.4744 0.507    0.612
##
##             exp(coef) exp(-coef) lower .95 upper .95
```

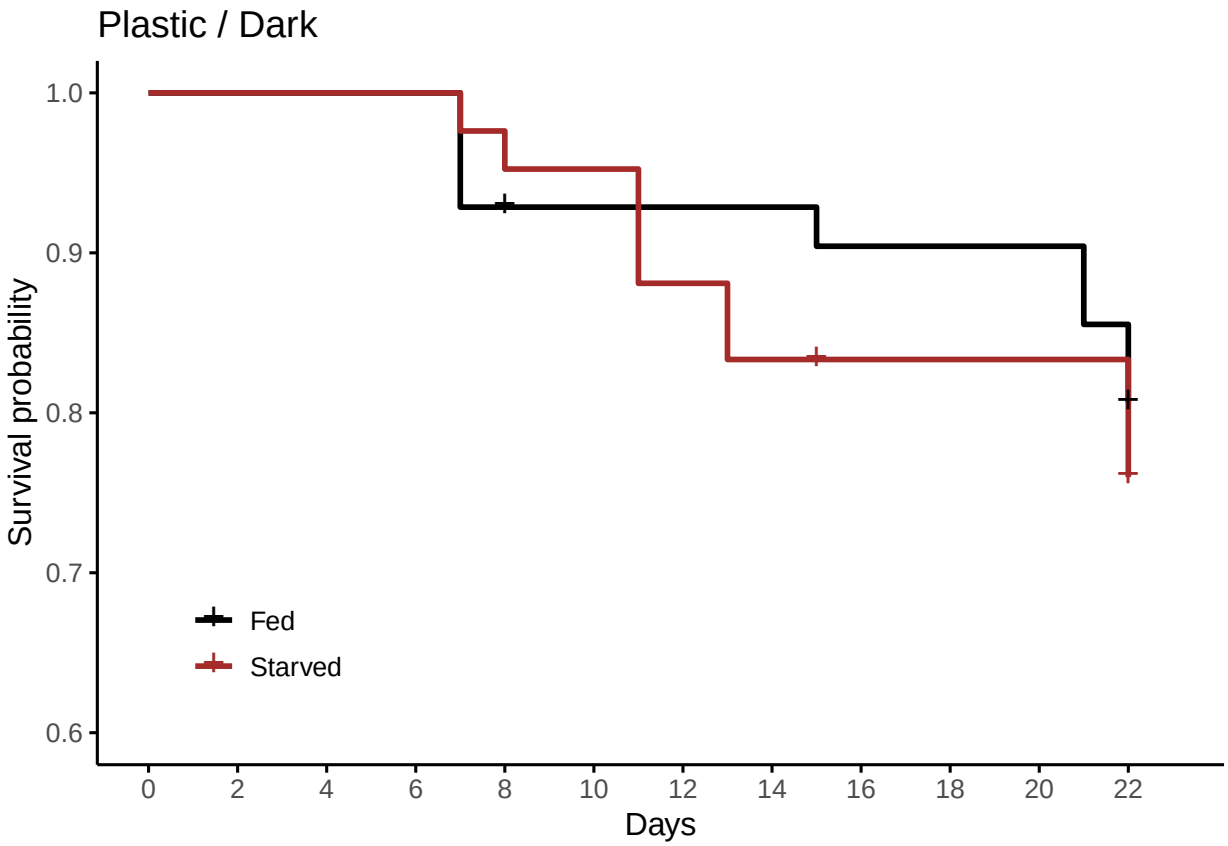
```
## FoodStarved     1.272     0.7861     0.502     3.224
##
## Concordance= 0.53  (se = 0.06 )
## Likelihood ratio test= 0.26  on 1 df,   p=0.6
## Wald test            = 0.26  on 1 df,   p=0.6
## Score (logrank) test = 0.26  on 1 df,   p=0.6
```

```
# Perform Log-rank test
fs2_diff <- survdiff(Surv(Time, Death) ~ Food, data=FS2.dat)
fs2_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Food, data = FS2.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Food=Fed      42        8     9.06     0.124     0.259
## Food=Starved 42       10     8.94     0.126     0.259
##
##  Chisq= 0.3  on 1 degrees of freedom, p= 0.6
```

```
# surv fit
fs2 <- survfit(Surv(Time, Death) ~ Food, data = FS2.dat)
```

```
# plot
fs2.plot <- ggsurvplot(fs2, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Fed", "Starved"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "brown"),
           title = "Plastic / Dark")

fs2.plot
```



FOOD: Fed vs Starved in No Plastic / Dark treatment

```
# Fed vs Starved: for ** No Plastic / Dark **

# filter out observations
# New df with just Plastic/Dark/Fed variables
FS3.dat <- filter(surv.dat, Treatment == "NDF" | Treatment == "NDS")
```

```
# cox model for fs3
fs3.coxm <- coxph(Surv(Time, Death) ~ Food, data=FS3.dat)
```

```
summary(fs3.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Food, data = FS3.dat)
##
##   n= 84, number of events= 16
##
##                 coef exp(coef) se(coef)      z Pr(>|z|)
## FoodStarved -2.64122   0.07127  1.03800 -2.545   0.0109 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

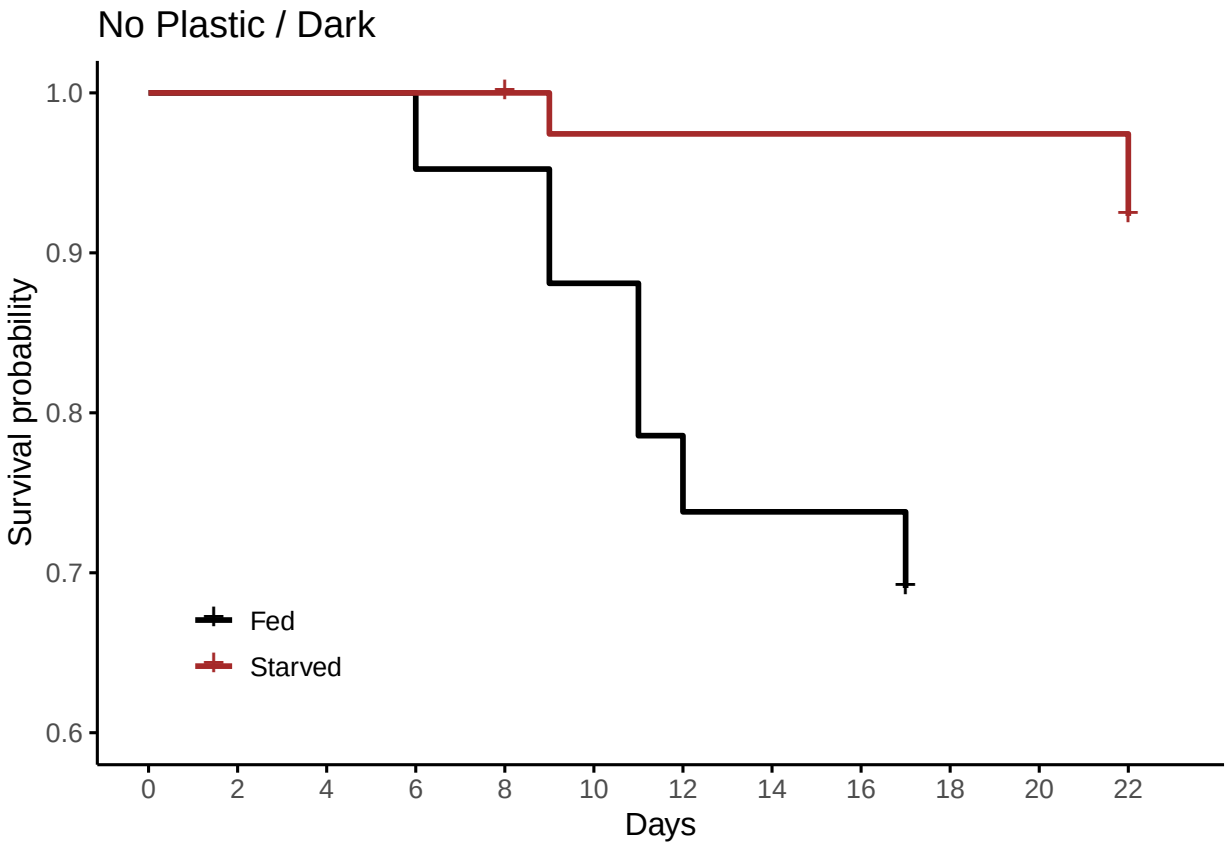
```
##
##             exp(coef) exp(-coef) lower .95 upper .95
## FoodStarved   0.07127      14.03  0.009319    0.5451
##
## Concordance= 0.713  (se = 0.04 )
## Likelihood ratio test= 13.12  on 1 df,   p=3e-04
## Wald test            = 6.47  on 1 df,   p=0.01
## Score (logrank) test = 11.22  on 1 df,   p=8e-04
```

```
# Perform Log-rank test
fs3_diff <- survdiff(Surv(Time, Death) ~ Food, data=FS3.dat)
fs3_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Food, data = FS3.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Food=Fed      42       13     6.83      5.58      11.2
## Food=Starved 42        3     9.17      4.15      11.2
##
##  Chisq= 11.2  on 1 degrees of freedom, p= 8e-04
```

```
# surv fit
fs3 <- survfit(Surv(Time, Death) ~ Food, data = FS3.dat)
```

```
# plot
fs3.plot <- ggsurvplot(fs3, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Fed", "Starved"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "brown"),
           title = "No Plastic / Dark")

fs3.plot
```



FOOD: Fed vs Starved in Plastic / Light treatment

```
# Fed vs Starved: for ** Plastic / Light **

# filter out observations
# New df with just Plastic/Dark/Fed variables
FS4.dat <- filter(surv.dat, Treatment == "PLF" | Treatment == "PLS")
```

```
# cox model for fs4
fs4.coxm <- coxph(Surv(Time, Death) ~ Food, data=FS4.dat)
```

```
summary(fs4.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Food, data = FS4.dat)
##
##   n= 84, number of events= 23
##
##                coef exp(coef) se(coef)      z Pr(>|z|)
## FoodStarved -0.5354    0.5854   0.4273 -1.253     0.21
##
##             exp(coef) exp(-coef) lower .95 upper .95
```

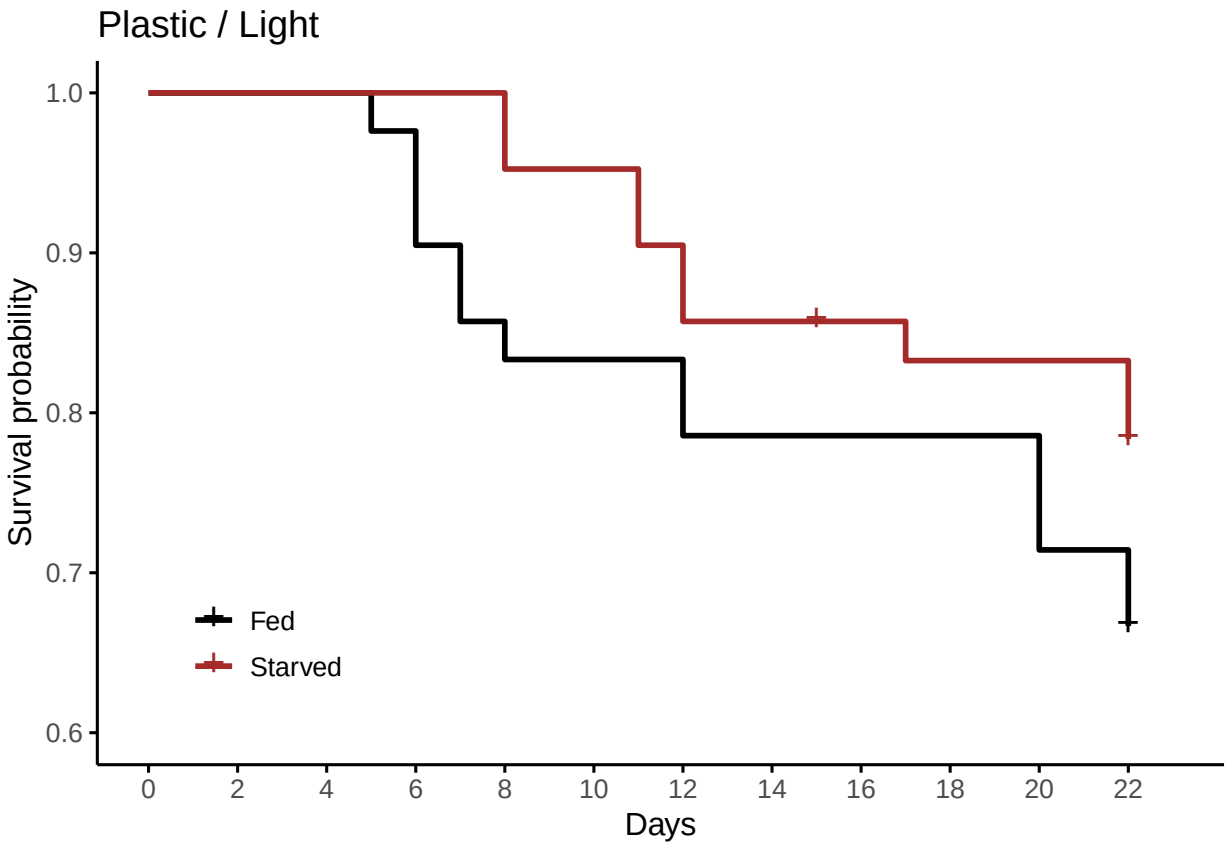
```
## FoodStarved    0.5854      1.708    0.2534     1.353
##
## Concordance= 0.572  (se = 0.052 )
## Likelihood ratio test= 1.61  on 1 df,   p=0.2
## Wald test            = 1.57  on 1 df,   p=0.2
## Score (logrank) test = 1.61  on 1 df,   p=0.2
```

```
# Perform Log-rank test
fs4_diff <- survdiff(Surv(Time, Death) ~ Food, data=FS4.dat)
fs4_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Food, data = FS4.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Food=Fed      42       14       11     0.815      1.61
## Food=Starved 42        9       12     0.748      1.61
##
##  Chisq= 1.6  on 1 degrees of freedom, p= 0.2
```

```
# surv fit
fs4 <- survfit(Surv(Time, Death) ~ Food, data = FS4.dat)
```

```
# plot
fs4.plot <- ggsurvplot(fs4, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Fed", "Starved"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "brown"),
           title = "Plastic / Light")

fs4.plot
```




```
# can we combine fs1-fs4 onto a single plot, or can we facet them?
```

Light

```
# Light
light.mod <- coxph(Surv(Time, Death) ~ Light, data=surv.dat)
```

```
summary(light.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Light, data = surv.dat)
##
##   n= 498, number of events= 72
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## LightLight -0.6552    0.5194   0.2363 -2.772  0.00557 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## LightLight    0.5194      1.925    0.3268    0.8253
##
```

```
## Concordance= 0.577  (se = 0.03 )
## Likelihood ratio test= 7.46  on 1 df,   p=0.006
## Wald test            = 7.68  on 1 df,   p=0.006
## Score (logrank) test = 7.96  on 1 df,   p=0.005
```

```
# HR light:dark <- (0.5073)
# being in the light reduced the hazard by 49.27%
#   1-exp(coef)
#   1-0.5073 = 0.4927 * 100 = 49.27
# being in the dark increased the hazard by 97.1%
#   1-exp(-coef)
#   1-1.971 = -0.971 * 100 = 97.1
```

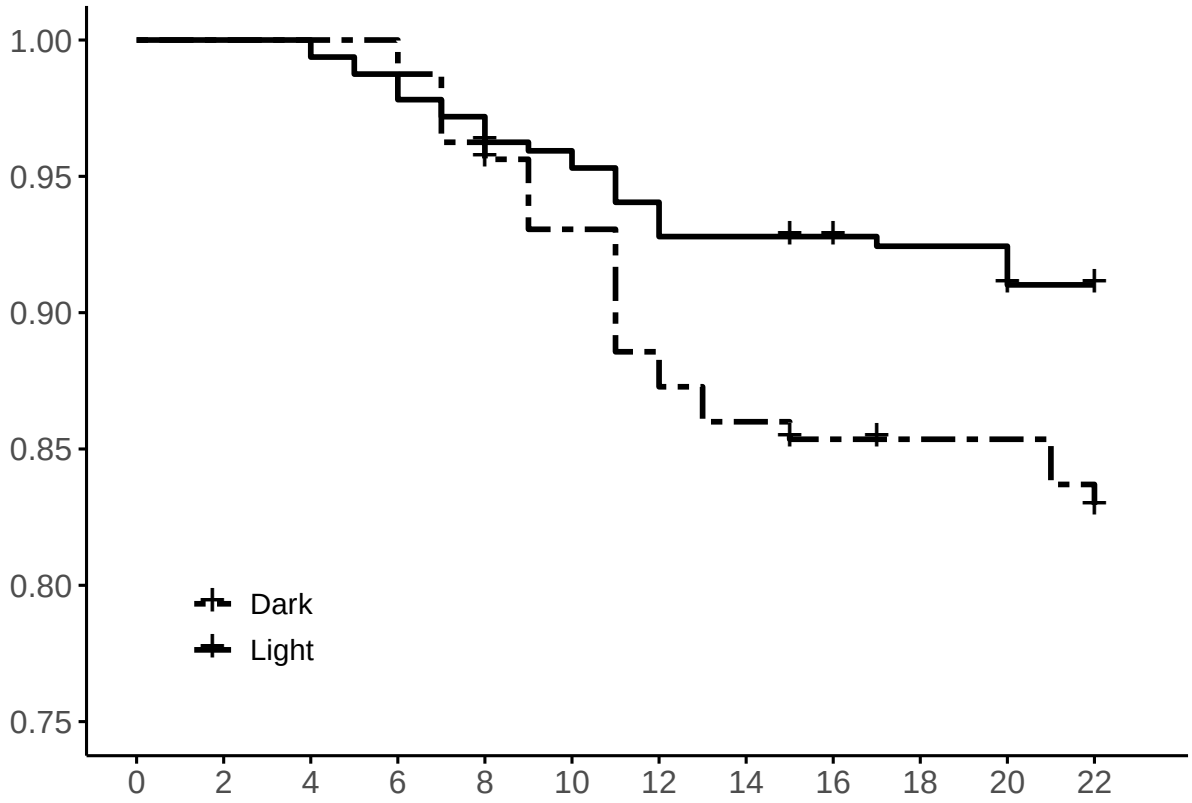
```
# Perform Log-rank test on our light.mod
light_diff <- survdiff(Surv(Time, Death) ~ Light, data=surv.dat)
light_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Light, data = surv.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Light=Dark   168       34       23      5.29      7.91
## Light=Light 330       38       49      2.48      7.91
##
##  Chisq= 7.9  on 1 degrees of freedom, p= 0.005
```

```
# X^2 = 6.4
# df = 1
# p = 0.01
```

```
# kaplan meier curve
light <- survfit(Surv(Time, Death) ~ Light, data = graph.dat)
```

```
# plot
km.light.fig <-
  ggsurvplot(light,
          conf.int = FALSE, pval = FALSE, surv.scale = "default", # or %
           legend.labs = c("Dark", "Light"), legend.title = "", legend = c(0.15, 0.2), font.legend = 11
           xlab = "", xlim = c(0, 23),break.x.by = 2, font.x = 12,
           ylab = "", ylim = c(0.75,1),font.y= 12,
           font.tickslab = 12,
           censor.size = 6,
           ggtheme = theme_classic2(),
           linetype = c(6,1),
           palette = c("black","black"))

km.light.fig
```



Light: Apo frags

```
# Dark vs Light: for apo frags

# filter out observations
# New df with just apo frags
apo.dat <- filter(surv.dat, Sym.State == "Apo")
```

```
# cox model for apo
apo.coxm <- coxph(Surv(Time, Death) ~ Light, data=apo.dat)
```

```
summary(apo.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Light, data = apo.dat)
##
##   n= 249, number of events= 46
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## LightLight -0.6638    0.5149   0.2955 -2.247   0.0247 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

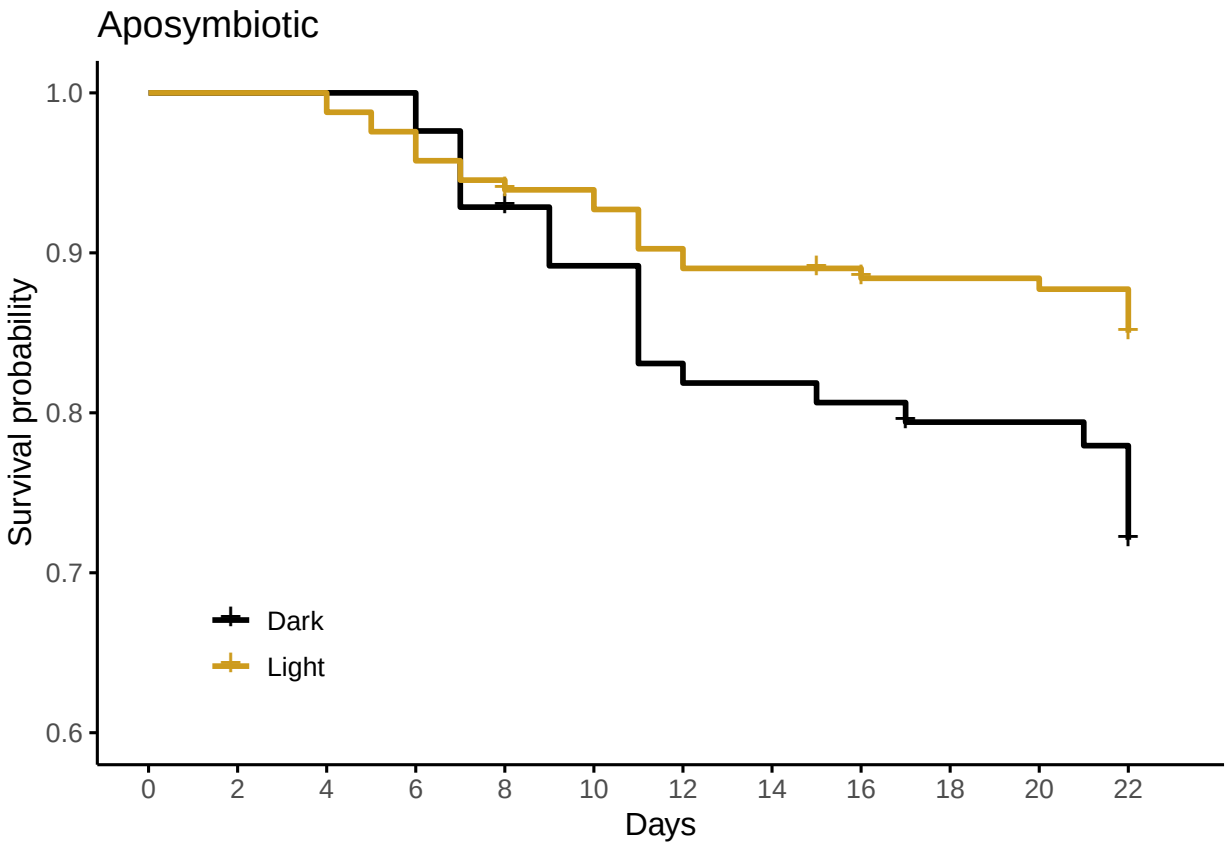
```
##
##             exp(coef) exp(-coef) lower .95 upper .95
## LightLight    0.5149      1.942    0.2886    0.9188
##
## Concordance= 0.576  (se = 0.037 )
## Likelihood ratio test= 4.91  on 1 df,   p=0.03
## Wald test            = 5.05  on 1 df,   p=0.02
## Score (logrank) test = 5.23  on 1 df,   p=0.02
```

```
# Perform Log-rank test
apo_diff <- survdiff(Surv(Time, Death) ~ Light, data=apo.dat)
apo_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Light, data = apo.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Light=Dark    84       22     14.9      3.43      5.18
## Light=Light 165       24     31.1      1.64      5.18
##
##  Chisq= 5.2  on 1 degrees of freedom, p= 0.02
```

```
# surv fit
afit <- survfit(Surv(Time, Death) ~ Light, data = apo.dat)
```

```
# plot
a.plot <- ggsurvplot(afit, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Dark", "Light"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "goldenrod3"),
           title = "Aposymbiotic")

a.plot
```



Light: Sym frags

```
# Dark vs Light: for sym frags

# filter out observations
# New df with just apo
sym.dat <- filter(surv.dat, Sym.State == "Sym")
```

```
# cox model for sym
sym.coxm <- coxph(Surv(Time, Death) ~ Light, data=sym.dat)
```

```
summary(sym.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Light, data = sym.dat)
##
##   n= 249, number of events= 26
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## LightLight -0.6400    0.5273   0.3940 -1.624    0.104
##
##            exp(coef) exp(-coef) lower .95 upper .95
```

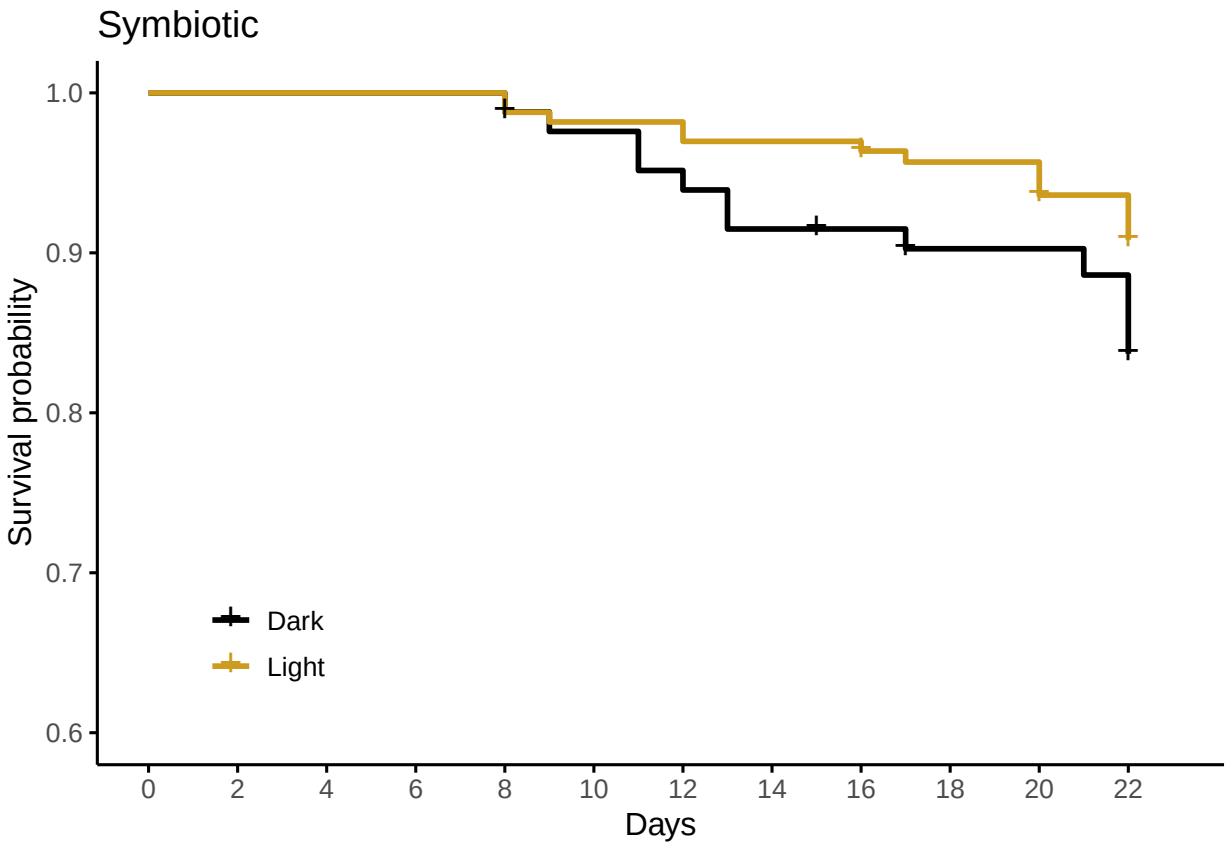
```
## LightLight    0.5273      1.897    0.2436     1.141
##
## Concordance= 0.579  (se = 0.049 )
## Likelihood ratio test= 2.55  on 1 df,   p=0.1
## Wald test            = 2.64  on 1 df,   p=0.1
## Score (logrank) test = 2.73  on 1 df,   p=0.1
```

```
# Perform Log-rank test
sym_diff <- survdiff(Surv(Time, Death) ~ Light, data=sym.dat)
sym_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Light, data = sym.dat)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Light=Dark    84       12     8.12     1.852      2.73
## Light=Light 165       14    17.88     0.841      2.73
##
##  Chisq= 2.7  on 1 degrees of freedom, p= 0.1
```

```
# surv fit
sfit <- survfit(Surv(Time, Death) ~ Light, data = sym.dat)
```

```
# plot
s.plot <- ggsurvplot(sfit, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Dark", "Light"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("black", "goldenrod3"),
           title = "Symbiotic")

s.plot
```



Temperature

```
# Temperature
temperature.mod <- coxph(Surv(Time, Death) ~ Temperature, data=surv.dat)
```

```
summary(temperature.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Temperature, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                       coef exp(coef) se(coef)    z Pr(>|z|)
## TemperatureElevated  3.0495   21.1049   0.7175 4.25 2.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## TemperatureElevated      21.1    0.04738     5.172     86.12
##
## Concordance= 0.674  (se = 0.013 )
## Likelihood ratio test= 51.39  on 1 df,   p=8e-13
```

```
## Wald test            = 18.06  on 1 df,   p=2e-05
## Score (logrank) test = 37.04  on 1 df,   p=1e-09
```

```
# HR elevated:ambient <- ????
```

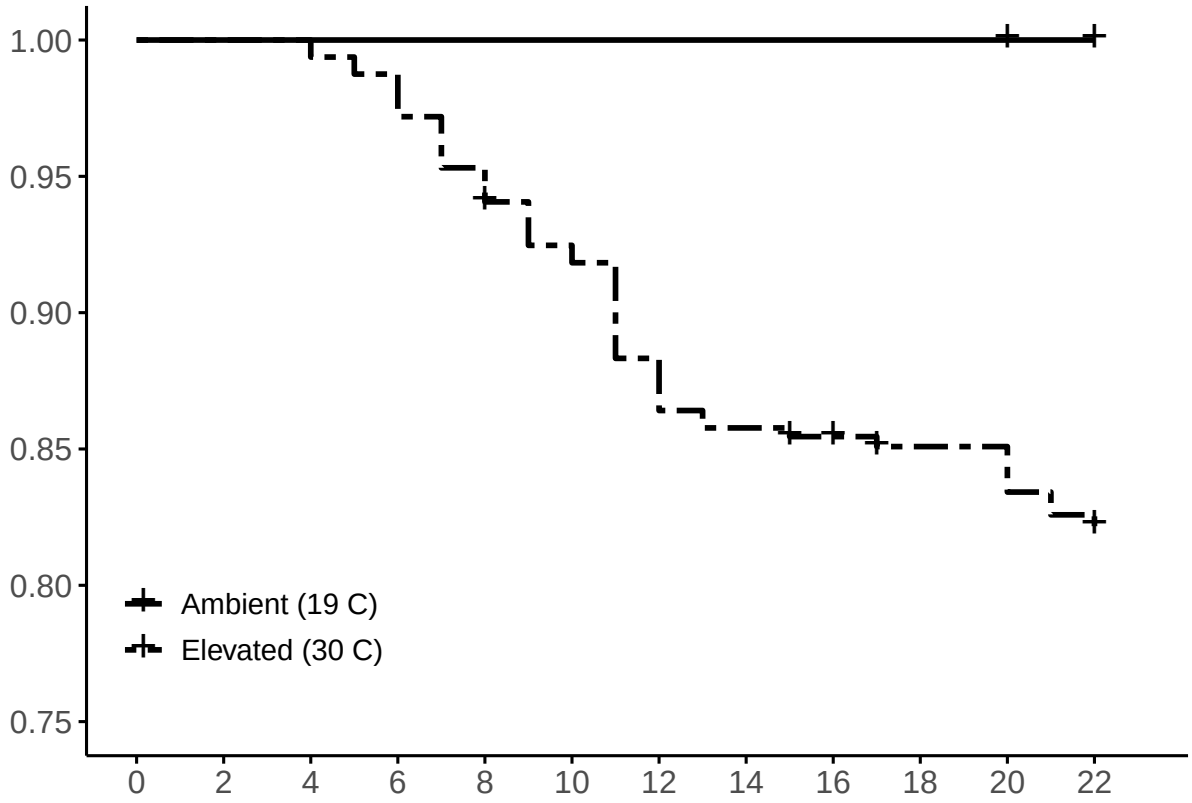
```
# Perform Log-rank test on our temperature.mod
temperature_diff <- survdiff(Surv(Time, Death) ~ Temperature, data=surv.dat)
temperature_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Temperature, data = surv.dat)
##
##                         N Observed Expected (O-E)^2/E (O-E)^2/V
## Temperature=Ambient   162        2     26.7      22.8      37.2
## Temperature=Elevated 336       70     45.3      13.5      37.2
##
##  Chisq= 37.2  on 1 degrees of freedom, p= 1e-09
```

```
# X^2 = 31.3
# df = 1
# p = 2x10^-8
```

```
# kaplan meier curve
temperature <- survfit(Surv(Time, Death) ~ Temperature, data = graph.dat)
```

```
# plot
km.temp.fig <-
  ggsurvplot(temperature,
          conf.int = FALSE, pval = FALSE, surv.scale = "default", # or %
           legend.labs = c("Ambient (19 C)", "Elevated (30 C)"), legend.title = "", legend = c(0.15, 0.1
           xlab = "", xlim = c(0, 23),break.x.by = 2, font.x = 12,
           ylab = "", ylim = c(0.75,1),font.y= 12,
           font.tickslab = 12,
           censor.size = 6,
           ggtheme = theme_classic2(),
           linetype = c(1,6),
           palette = c("black","black"))

km.temp.fig
```



Temperature: ambient vs elevated controls

```
# Ambient vs elevated controls

# filter out observations
# New df with just apo
tmp.dat <- filter(surv.dat, Treatment == "NLF" | Treatment == "amb.ctrl")
```

```
# cox model for temperature controls
tmp.coxm <- coxph(Surv(Time, Death) ~ Temperature, data=tmp.dat)
```

```
## Warning in coxph.fit(X, Y, istrat, offset, init, control, weights = weights, :
## Loglik converged before variable 1 ; coefficient may be infinite.
```

```
summary(tmp.coxm)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Temperature, data = tmp.dat)
##
##   n= 204, number of events= 12
##
##                          coef exp(coef)  se(coef)     z Pr(>|z|)
```

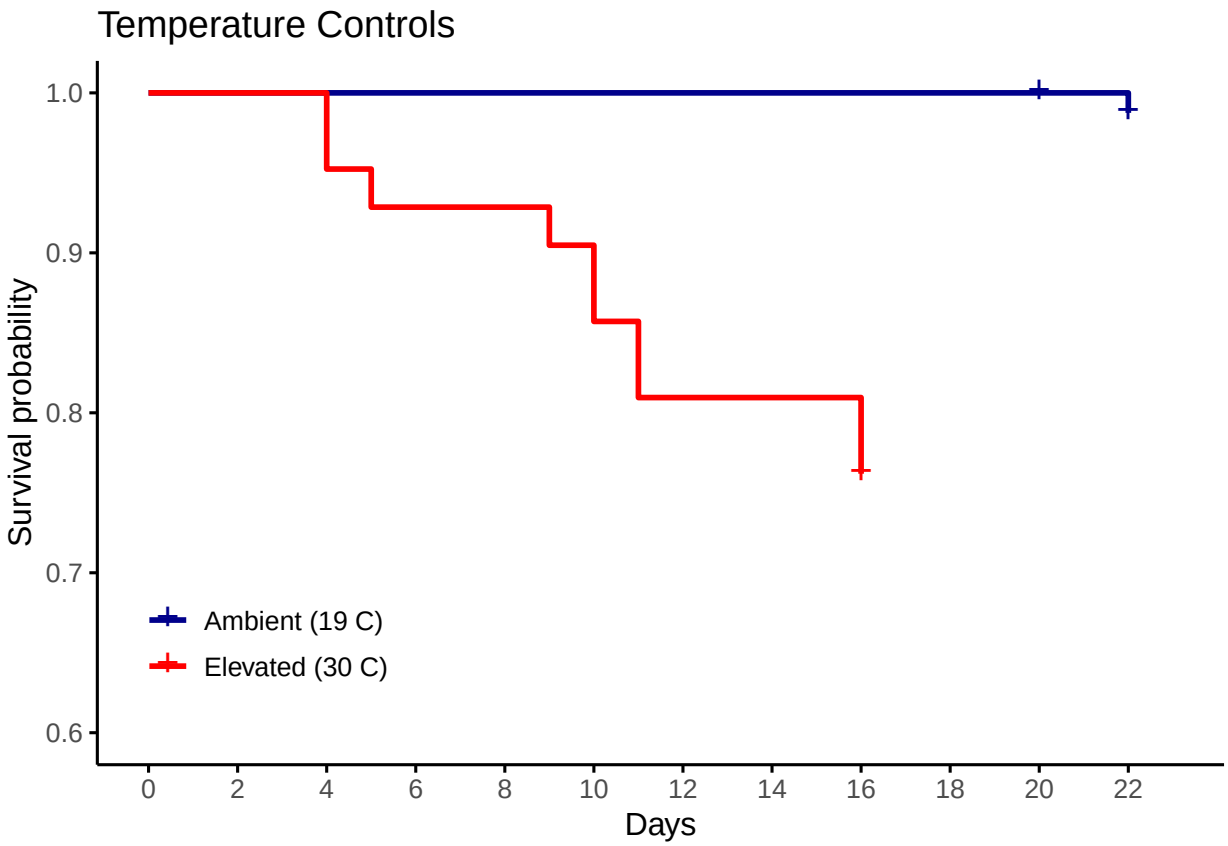
```
## TemperatureElevated 2.136e+01 1.896e+09 6.604e+03 0.003    0.997
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## TemperatureElevated 1.896e+09  5.276e-10         0       Inf
##
## Concordance= 0.853  (se = 0.037 )
## Likelihood ratio test= 33.49  on 1 df,   p=7e-09
## Wald test            = 0  on 1 df,   p=1
## Score (logrank) test = 43.29  on 1 df,   p=5e-11
```

```
# Perform Log-rank test
tmp_diff <- survdiff(Surv(Time, Death) ~ Temperature, data=tmp.dat)
tmp_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Temperature, data = tmp.dat)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## Temperature=Ambient  162        2    10.11       6.5        43
## Temperature=Elevated  42       10     1.89      34.7        43
##
##  Chisq= 43  on 1 degrees of freedom, p= 5e-11
```

```
# surv fit
tfit <- survfit(Surv(Time, Death) ~ Temperature, data = tmp.dat)
```

```
# plot
t.plot <- ggsurvplot(tfit, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Ambient (19 C)", "Elevated (30 C)"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.6,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("darkblue", "red"),
           title = "Temperature Controls")

t.plot
```



Sym State

```
# Sym.State
symst.mod <- coxph(Surv(Time, Death) ~ Sym.State, data=surv.dat)
```

```
summary(symst.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Sym.State, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                 coef exp(coef) se(coef)      z Pr(>|z|)
## Sym.StateSym -0.6319    0.5316   0.2454 -2.575     0.01 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## Sym.StateSym    0.5316      1.881    0.3286    0.8599
##
## Concordance= 0.586  (se = 0.028 )
## Likelihood ratio test= 6.92  on 1 df,   p=0.009
```

```
## Wald test            = 6.63  on 1 df,   p=0.01
## Score (logrank) test = 6.85  on 1 df,   p=0.009
```

```
# HR sym:apo <- 0.4293
# being sym reduced the hazard by 57.07%
#   1-exp(coef)
#   1-0.4293 = 0.5707 * 100 = 57.07
# being apo increased the hazard by 132.9%
#   1-exp(-coef)
#   1-2.329 = -1.329 * 100 = 132.9
```

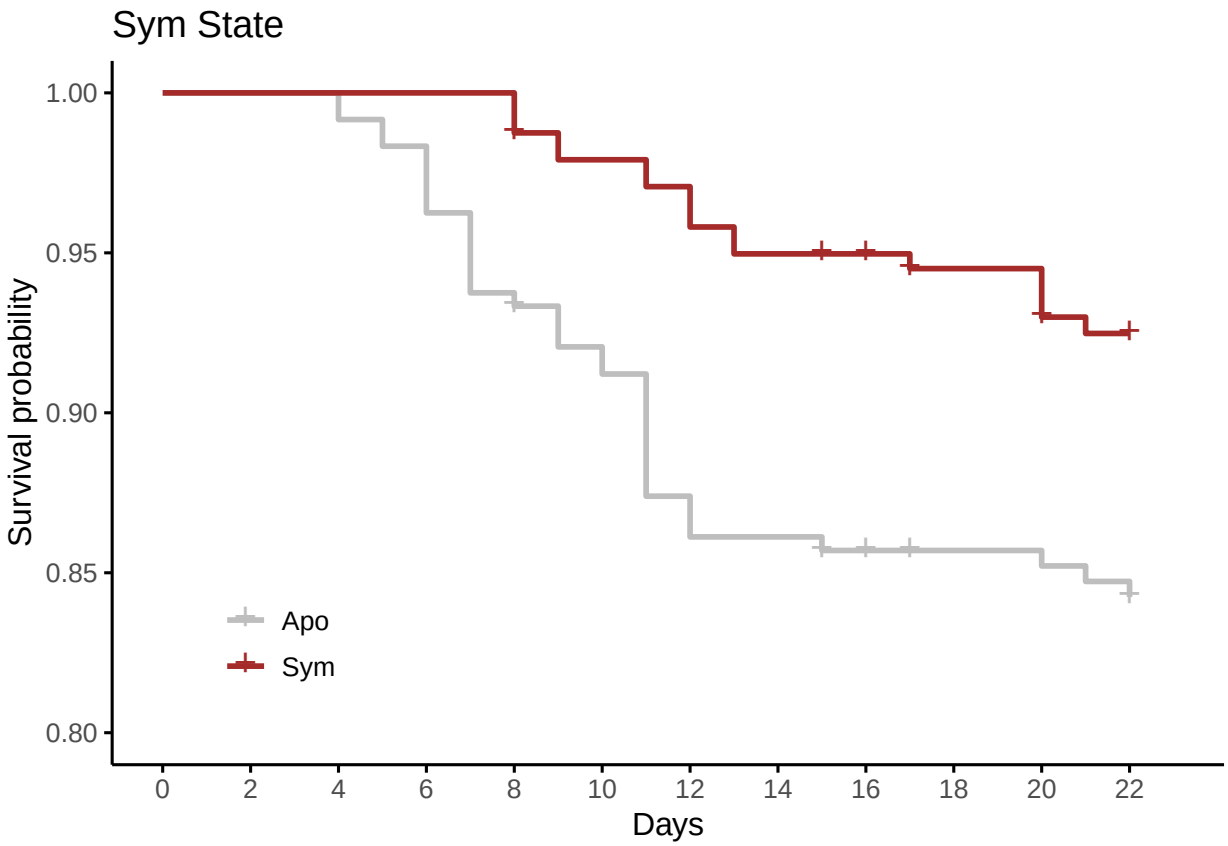
```
# Perform Log-rank test on our temperature.mod
sym_diff <- survdiff(Surv(Time, Death) ~ Sym.State, data=surv.dat)
sym_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Sym.State, data = surv.dat)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## Sym.State=Apo 249       46       35      3.47      6.87
## Sym.State=Sym 249       26       37      3.28      6.87
##
##  Chisq= 6.9  on 1 degrees of freedom, p= 0.009
```

```
# X^2 = 8.8
# df = 1
# p = 0.003
```

```
# surv fit
symfit <- survfit(Surv(Time, Death) ~ Sym.State, data = graph.dat)
```

```
# plot
s.plot <- ggsurvplot(symfit, conf.int = FALSE,
           pval = FALSE,
           surv.scale = "default", # or %
           legend.labs = c("Apo", "Sym"),
           legend.title = "",
           legend = c(0.15, 0.2),
           font.legend = 10,
           xlab = "Days",
           xlim = c(0, 23),
           break.x.by = 2,
           font.x = 12,
           ylim = c(0.8,1),
           font.y= 12,
           censor.size = 5,
           ggtheme = theme_classic2(),
           palette = c("gray", "brown"),
           title = "Sym State")

s.plot
```




```
# lets see if we can do a better pairwise analysis for all treatments
library(emmeans)
lsmeans(treat.mod, pairwise~Treatment)
```

```
## $lsmeans
##  Treatment lsmean    SE  df asymp.LCL asymp.UCL
##  amb.ctrl    0.00 0.000 Inf    0.0000      0.00
##  PDS         3.10 0.775 Inf    1.5848      4.62
##  PLS         2.99 0.782 Inf    1.4566      4.52
##  NDS         1.84 0.913 Inf    0.0481      3.63
##  NLS         1.81 0.913 Inf    0.0175      3.60
##  PDF         2.86 0.791 Inf    1.3069      4.41
##  PLF         3.53 0.756 Inf    2.0516      5.02
##  NDF         3.99 0.771 Inf    2.4735      5.50
##  NLF         3.79 0.787 Inf    2.2494      5.34
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
##
## $contrasts
##  contrast       estimate    SE  df z.ratio p.value
##  amb.ctrl - PDS  -3.1032 0.775 Inf  -4.006  0.0020
##  amb.ctrl - PLS  -2.9889 0.782 Inf  -3.823  0.0042
##  amb.ctrl - NDS  -1.8373 0.913 Inf  -2.013  0.5344
##  amb.ctrl - NLS  -1.8067 0.913 Inf  -1.979  0.5581
##  amb.ctrl - PDF  -2.8564 0.791 Inf  -3.613  0.0092
```

```
##  amb.ctrl - PLF  -3.5334 0.756 Inf  -4.674  0.0001
##  amb.ctrl - NDF  -3.9850 0.771 Inf  -5.167  <.0001
##  amb.ctrl - NLF  -3.7926 0.787 Inf  -4.817  0.0001
##  PDS - PLS        0.1143 0.459 Inf   0.249  1.0000
##  PDS - NDS        1.2659 0.658 Inf   1.923  0.5977
##  PDS - NLS        1.2965 0.658 Inf   1.969  0.5649
##  PDS - PDF        0.2468 0.474 Inf   0.520  0.9999
##  PDS - PLF       -0.4302 0.414 Inf  -1.039  0.9821
##  PDS - NDF       -0.8818 0.438 Inf  -2.013  0.5342
##  PDS - NLF       -0.6894 0.466 Inf  -1.480  0.8650
##  PLS - NDS        1.1516 0.667 Inf   1.727  0.7297
##  PLS - NLS        1.1822 0.667 Inf   1.773  0.7001
##  PLS - PDF        0.1325 0.486 Inf   0.273  1.0000
##  PLS - PLF       -0.5445 0.427 Inf  -1.274  0.9389
##  PLS - NDF       -0.9961 0.451 Inf  -2.211  0.3986
##  PLS - NLF       -0.8037 0.478 Inf  -1.683  0.7573
##  NDS - NLS        0.0306 0.816 Inf   0.037  1.0000
##  NDS - PDF       -1.0191 0.677 Inf  -1.505  0.8535
##  NDS - PLF       -1.6961 0.636 Inf  -2.666  0.1602
##  NDS - NDF       -2.1477 0.654 Inf  -3.286  0.0283
##  NDS - NLF       -1.9553 0.673 Inf  -2.907  0.0869
##  NLS - PDF       -1.0498 0.677 Inf  -1.551  0.8314
##  NLS - PLF       -1.7267 0.636 Inf  -2.714  0.1428
##  NLS - NDF       -2.1783 0.654 Inf  -3.333  0.0243
##  NLS - NLF       -1.9859 0.673 Inf  -2.953  0.0768
##  PDF - PLF       -0.6770 0.443 Inf  -1.527  0.8429
##  PDF - NDF       -1.1286 0.467 Inf  -2.418  0.2741
##  PDF - NLF       -0.9361 0.493 Inf  -1.899  0.6145
##  PLF - NDF       -0.4516 0.404 Inf  -1.119  0.9716
##  PLF - NLF       -0.2591 0.434 Inf  -0.598  0.9996
##  NDF - NLF        0.1924 0.423 Inf   0.455  1.0000
##
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 9 estimates
```

```
#doesnt look like any of the pairwise comparisons for the coxph model are significant?
```

```
# run multiple univariate analyses at once - does not take into account *interaction* between variables

covariates <- c("Plastic", "Food",  "Light", "Temperature", "Sym.State")
univ_formulas <- sapply(covariates,
                        function(x) as.formula(paste('Surv(Time, Death)~', x)))

univ_models <- lapply( univ_formulas, function(x){coxph(x, data = surv.dat)})

# Extract data
univ_results <- lapply(univ_models,
                       function(x){
                          x <- summary(x)
                          p.value<-signif(x$wald["pvalue"], digits=2)
                          wald.test<-signif(x$wald["test"], digits=2)
                          beta<-signif(x$coef[1], digits=2);#coefficient beta
                          HR <-signif(x$coef[2], digits=2);#exp(beta)
                          HR.confint.lower <- signif(x$conf.int[,"lower .95"], 2)
```

```
                          HR.confint.upper <- signif(x$conf.int[,"upper .95"],2)
                          HR <- paste0(HR, " (",
                                       HR.confint.lower, "-", HR.confint.upper, ")")
                          res<-c(beta, HR, wald.test, p.value)
                          names(res)<-c("beta", "HR (95% CI for HR)", "wald.test(z)",
                                        "p.value")
                          return(res)
                          return(exp(cbind(coef(x),confint(x))))
                         })
res <- t(as.data.frame(univ_results, check.names = FALSE))
as.data.frame(res)
```

```
##              beta HR (95% CI for HR) wald.test(z) p.value
## Plastic      0.96      2.6 (1.6-4.2)           16 5.7e-05
## Food        -0.03     0.97 (0.6-1.6)         0.01     0.9
## Light       -0.66   0.52 (0.33-0.83)          7.7  0.0056
## Temperature     3        21 (5.2-86)           18 2.1e-05
## Sym.State   -0.63   0.53 (0.33-0.86)          6.6    0.01
```

Multivariate Statistical Analysis

```
# a full model that contains all of our parameters
multiv.mod <- coxph(Surv(Time, Death) ~ Plastic + Light + Food + Temperature + Sym.State,
                    data = surv.dat)
```

```
summary(multiv.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature + Sym.State, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                          coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic        0.03426   1.03485  0.25277  0.136 0.892182
## LightLight            0.12752   1.13600  0.23951  0.532 0.594446
## FoodStarved          -0.92275   0.39742  0.25707 -3.590 0.000331 ***
## TemperatureElevated   3.55853  35.11168  0.75387  4.720 2.35e-06 ***
## Sym.StateSym         -0.71454   0.48942  0.24578 -2.907 0.003646 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic          1.0349    0.96632    0.6306    1.6984
```

```
## LightLight             1.1360    0.88028    0.7104    1.8166
## FoodStarved            0.3974    2.51621    0.2401    0.6578
## TemperatureElevated   35.1117    0.02848    8.0123  153.8672
## Sym.StateSym           0.4894    2.04324    0.3023    0.7923
##
## Concordance= 0.777  (se = 0.025 )
## Likelihood ratio test= 73.58  on 5 df,   p=2e-14
## Wald test            = 40.37  on 5 df,   p=1e-07
## Score (logrank) test = 66.12  on 5 df,   p=7e-13
```

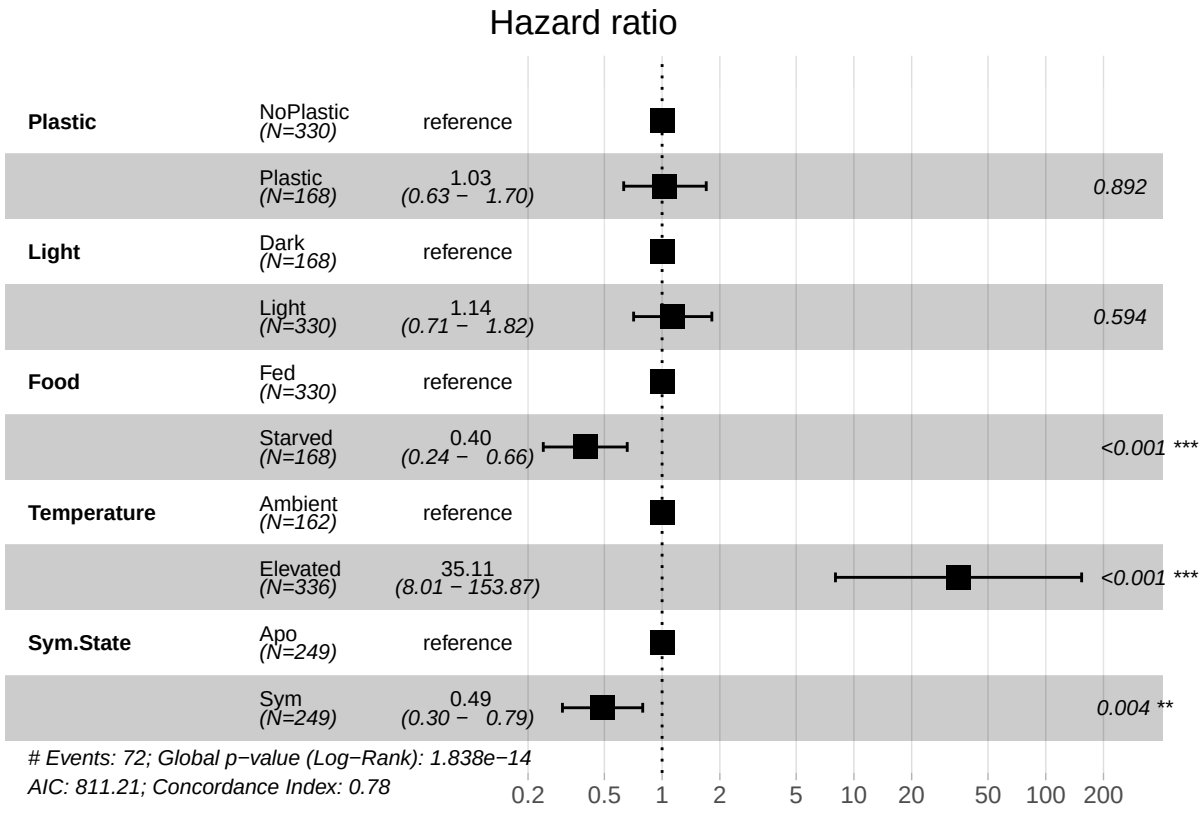
```
# Perform Log-rank test on our multiv.mod
multiv_diff <- survdiff(Surv(Time, Death) ~ Plastic + Light + Food + Temperature + Sym.State,
                    data = surv.dat)
multiv_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature + Sym.State, data = surv.dat)
##
##                                                                                N
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo 21
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym 21
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo 21
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym 21
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Apo 81
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Sym 81
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo 21
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym 21
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo 21
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym 21
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo   21
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym   21
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo   21
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym   21
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo   21
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym   21
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo   21
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym   21
##                                                                           Observed
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo       10
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym        3
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo        1
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym        2
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Apo        1
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Sym        1
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo        8
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym        2
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo        1
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym        2
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo          6
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym          2
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo          5
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym          5
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo          9
```

```
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym          5
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo          5
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym          4
##                                                                                    Expected
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo     1.81
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym     2.24
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo     3.22
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym     3.23
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Apo    13.39
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Sym    13.30
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo     1.70
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym     2.08
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo     3.22
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym     3.43
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo       2.96
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym       3.31
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo       3.16
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym       3.05
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo       2.43
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym       3.23
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo       3.08
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym       3.16
##                                                                                    (O-E)^2/E
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo   36.9839
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym    0.2610
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo    1.5301
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym    0.4684
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Apo   11.4696
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Sym   11.3783
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo   23.3618
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym    0.0031
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo    1.5301
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym    0.5941
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo      3.1249
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym      0.5156
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo      1.0692
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym      1.2493
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo     17.7453
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym      0.9687
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo      1.2001
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym      0.2215
##                                                                                    (O-E)^2/V
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo  38.76314
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym   0.27574
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo   1.62875
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym   0.49867
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Apo  14.36777
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient , Sym.State=Sym  14.22267
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo  24.47336
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym   0.00328
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo   1.62875
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym   0.63435
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Apo     3.31126
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated, Sym.State=Sym     0.54948
```

```
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Apo     1.13689
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated, Sym.State=Sym     1.32536
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Apo    18.66154
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated, Sym.State=Sym     1.03041
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Apo     1.27386
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated, Sym.State=Sym     0.23550
##
##  Chisq= 118  on 17 degrees of freedom, p= <2e-16
```

```
# Let's create a forest plots to show the output of our model
ggforest(multiv.mod, data=surv.dat)
```




```
# now lets take a look how how our model changes if we remove phenotype, since sym state is not an envi
m1 <- coxph(Surv(Time, Death) ~ Plastic + Light + Food + Temperature, data=surv.dat)
summary(m1)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                       coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic     0.03931   1.04009  0.25273  0.156 0.876398
## LightLight         0.11173   1.11821  0.23943  0.467 0.640745
```

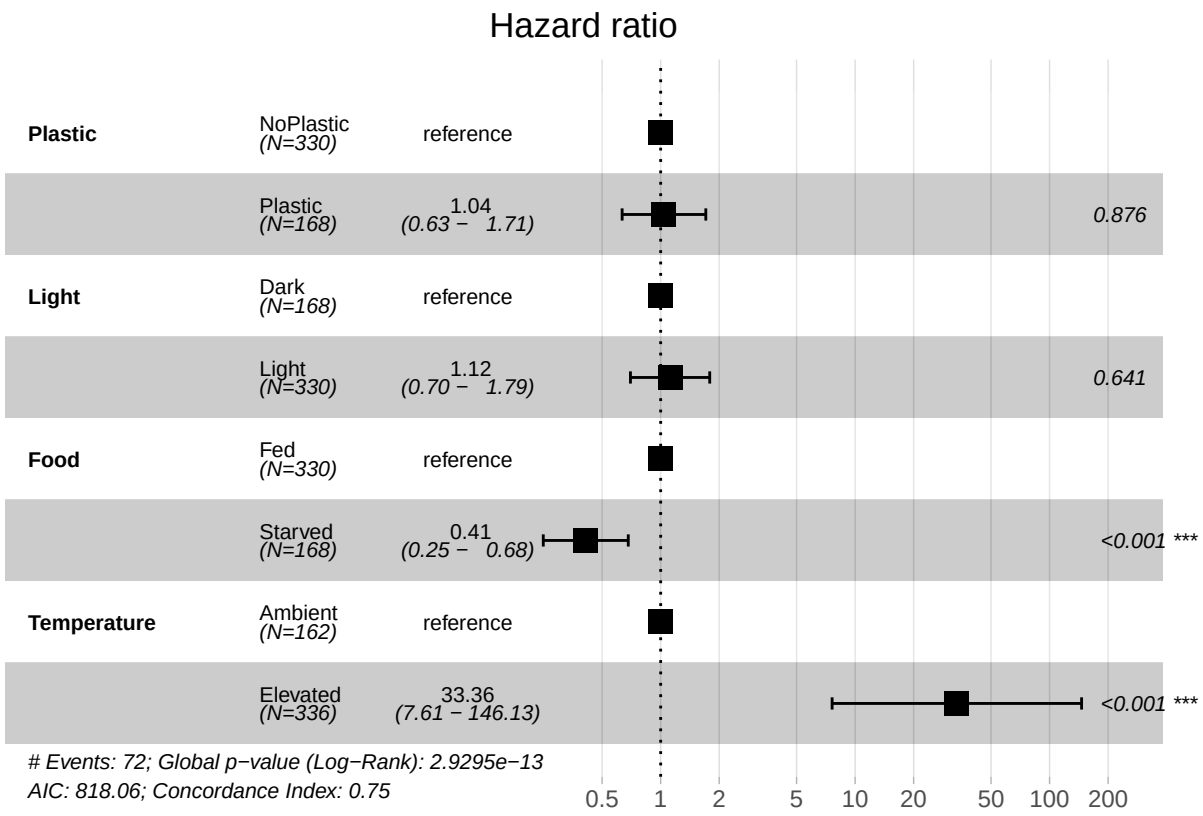
```
## FoodStarved         -0.88672   0.41200  0.25671 -3.454 0.000552 ***
## TemperatureElevated  3.50724  33.35619  0.75370  4.653 3.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic          1.040    0.96145    0.6338    1.7069
## LightLight              1.118    0.89428    0.6994    1.7878
## FoodStarved             0.412    2.42716    0.2491    0.6814
## TemperatureElevated    33.356    0.02998    7.6143  146.1253
##
## Concordance= 0.75  (se = 0.023 )
## Likelihood ratio test= 64.73  on 4 df,   p=3e-13
## Wald test            = 31.86  on 4 df,   p=2e-06
## Score (logrank) test = 57.4  on 4 df,   p=1e-11
```

```
# Perform Log-rank test
m1_diff <- survdiff(Surv(Time, Death) ~ Plastic + Light + Food + Temperature,data = surv.dat)
m1_diff
```

```
## Call:
## survdiff(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature, data = surv.dat)
##
##                                                                    N Observed
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated  42       13
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated  42        3
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient  162        2
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated  42       10
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated  42        3
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated    42        8
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated    42       10
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated    42       14
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated    42        9
##                                                                   Expected
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated     4.05
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated     6.45
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient     26.70
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated     3.78
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated     6.65
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated       6.26
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated       6.21
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated       5.66
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated       6.24
##                                                                   (O-E)^2/E
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated    19.791
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated     1.845
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient     22.848
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated    10.237
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated     2.000
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated       0.481
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated       2.313
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated      12.277
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated       1.220
```

```
##                                                                  (O-E)^2/V
## Plastic=NoPlastic, Light=Dark , Food=Fed    , Temperature=Elevated    21.723
## Plastic=NoPlastic, Light=Dark , Food=Starved, Temperature=Elevated     2.062
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Ambient     37.169
## Plastic=NoPlastic, Light=Light, Food=Fed    , Temperature=Elevated    11.228
## Plastic=NoPlastic, Light=Light, Food=Starved, Temperature=Elevated     2.242
## Plastic=Plastic, Light=Dark , Food=Fed    , Temperature=Elevated       0.535
## Plastic=Plastic, Light=Dark , Food=Starved, Temperature=Elevated       2.573
## Plastic=Plastic, Light=Light, Food=Fed    , Temperature=Elevated      13.539
## Plastic=Plastic, Light=Light, Food=Starved, Temperature=Elevated       1.357
##
##  Chisq= 76.4  on 8 degrees of freedom, p= 3e-13
```

```
ggforest(m1, data=surv.dat)
```




```
PLFT.fig <- ggforest(m1, data=surv.dat) # save as figure
```

```
# and now we can compare those two models using the likelihood ratio test
anova(multiv.mod, m1, test="LRT")
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(Time, Death)
##  Model 1: ~ Plastic + Light + Food + Temperature + Sym.State
##  Model 2: ~ Plastic + Light + Food + Temperature
```

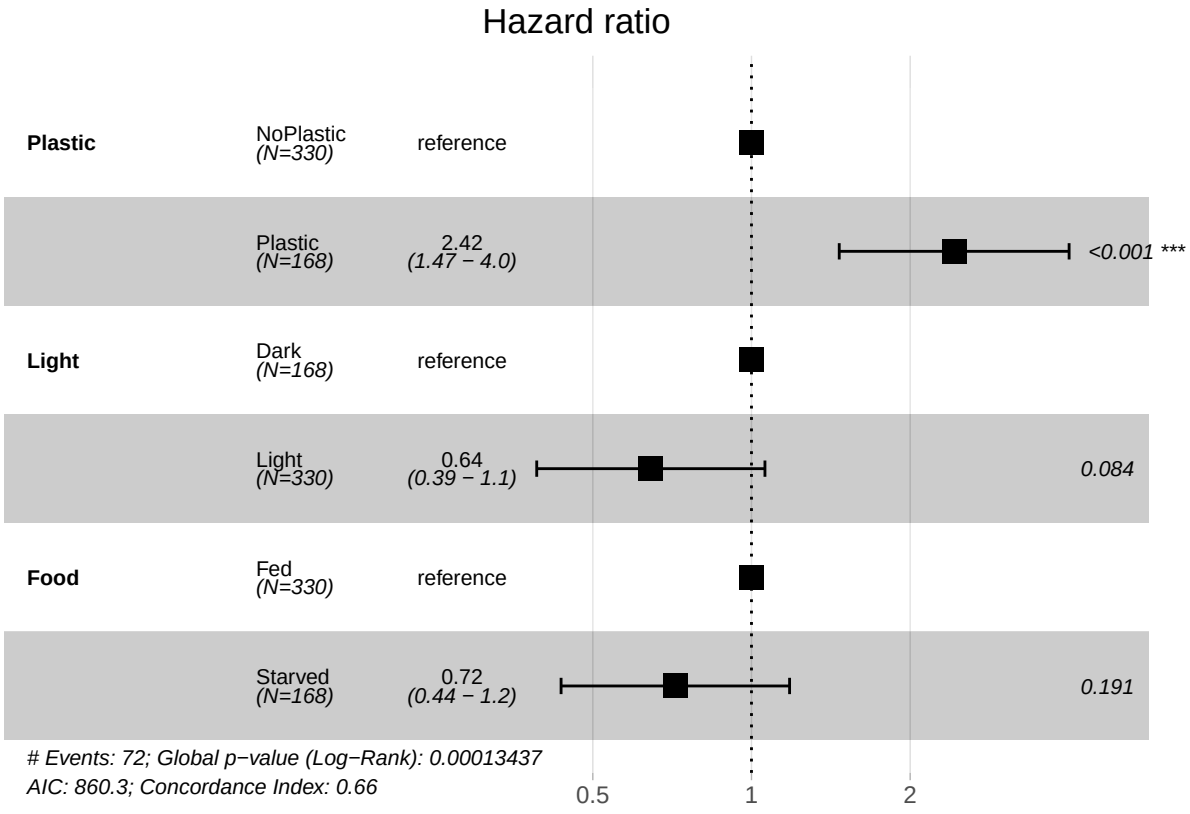
```
##    loglik  Chisq Df P(>|Chi|)
## 1 -400.61
## 2 -405.03 8.8497  1  0.002931 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# a high p value shows that we don't have evidence to say that the full model is better than the reduce
# while a low (and significant p value) tells us that our full model is not made better by dropping the
```

```
# We see that temperature has a very high effect here, what does our data look like when we remove it?
tmp.mod <- coxph(Surv(Time, Death) ~ Plastic + Light + Food, data = surv.dat)
summary(tmp.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food, data = surv.dat)
##
##   n= 498, number of events= 72
##
##                   coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic  0.8853    2.4237   0.2561  3.457 0.000547 ***
## LightLight     -0.4397    0.6443   0.2543 -1.729 0.083831 .
## FoodStarved    -0.3327    0.7170   0.2546 -1.306 0.191389
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic    2.4237     0.4126    1.4672     4.004
## LightLight        0.6443     1.5522    0.3914     1.061
## FoodStarved       0.7170     1.3947    0.4353     1.181
##
## Concordance= 0.662  (se = 0.028 )
## Likelihood ratio test= 20.49  on 3 df,   p=1e-04
## Wald test            = 21.05  on 3 df,   p=1e-04
## Score (logrank) test = 22.56  on 3 df,   p=5e-05
```

```
ggforest(tmp.mod, data=surv.dat)
```



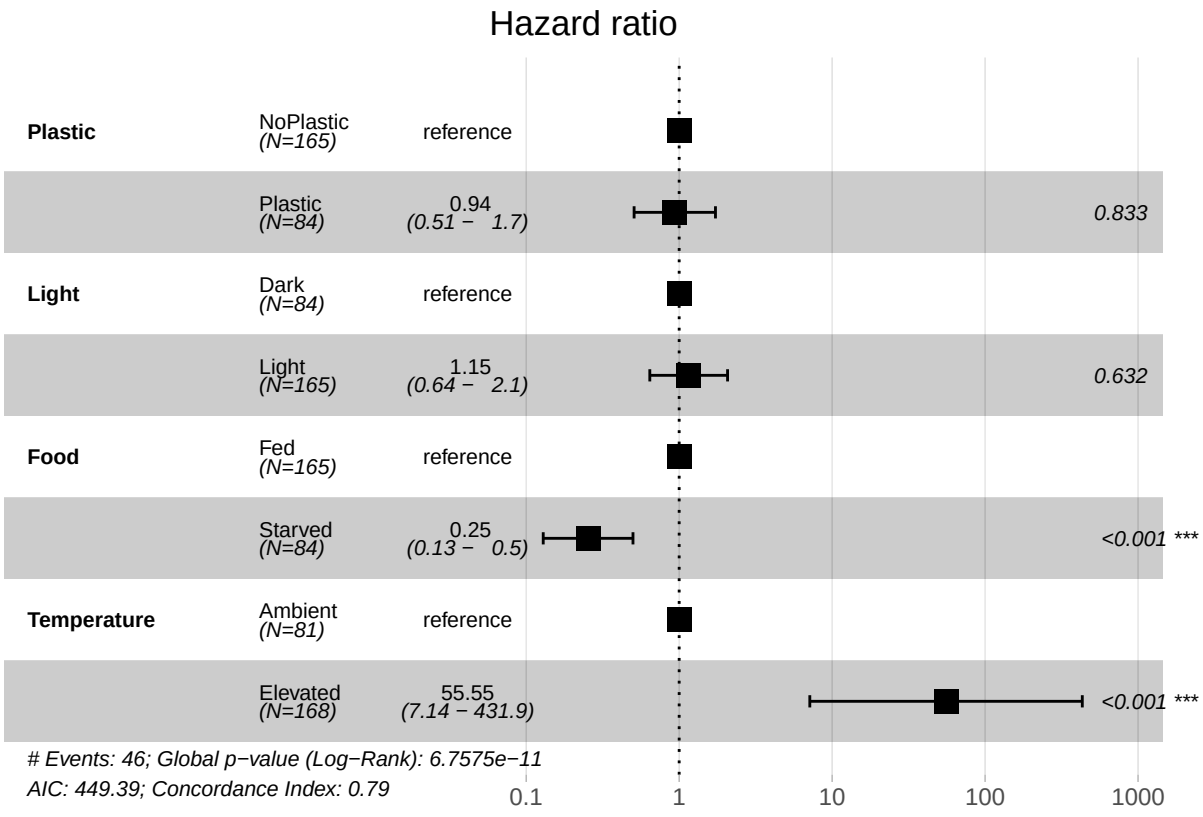

```
PLF.fig <- ggforest(tmp.mod, data=surv.dat) # save as figure
```

```
# And finally, we'll break it down by sym state, starting off with our Apo frags
# write a model exactly like the full model, but using the apo.dat data frame
apo.mod <- coxph(Surv(Time, Death) ~ Plastic + Light + Food + Temperature, data = apo.dat)
summary(apo.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature, data = apo.dat)
##
##   n= 249, number of events= 46
##
##                         coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic      -0.06604   0.93610  0.31244 -0.211 0.832608
## LightLight           0.14308   1.15382  0.29865  0.479 0.631885
## FoodStarved         -1.37105   0.25384  0.34469 -3.978 6.96e-05 ***
## TemperatureElevated  4.01722  55.54664  1.04640  3.839 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic         0.9361     1.0683    0.5074    1.7269
## LightLight             1.1538     0.8667    0.6426    2.0718
## FoodStarved            0.2538     3.9395    0.1292    0.4988
```

```
## TemperatureElevated   55.5466     0.0180    7.1443  431.8742
##
## Concordance= 0.788  (se = 0.028 )
## Likelihood ratio test= 53.48  on 4 df,   p=7e-11
## Wald test            = 28.12  on 4 df,   p=1e-05
## Score (logrank) test = 52.99  on 4 df,   p=9e-11
```

```
# and plot
ggforest(apo.mod, data=apo.dat)
```

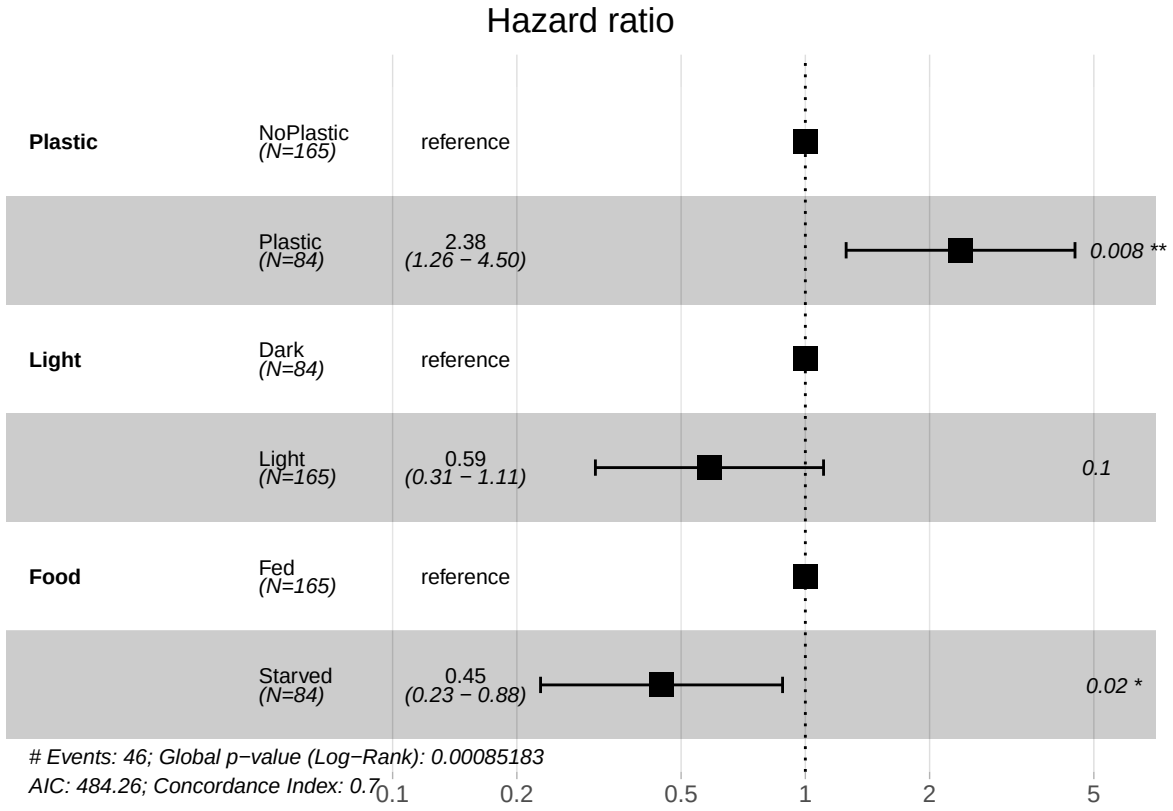



```
# now remove temperature (since it's so influential)
apo.mod2 <- coxph(Surv(Time, Death) ~ Plastic + Light + Food , data = apo.dat)
summary(apo.mod2)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food, data = apo.dat)
##
##   n= 249, number of events= 46
##
##                  coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic  0.8659    2.3771   0.3257  2.659  0.00785 **
## LightLight     -0.5348    0.5858   0.3247 -1.647  0.09951 .
## FoodStarved    -0.8021    0.4484   0.3444 -2.329  0.01987 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##                exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic    2.3771     0.4207    1.2555    4.5007
## LightLight        0.5858     1.7072    0.3100    1.1069
## FoodStarved       0.4484     2.2302    0.2283    0.8807
##
## Concordance= 0.702  (se = 0.034 )
## Likelihood ratio test= 16.61  on 3 df,   p=9e-04
## Wald test            = 17.59  on 3 df,   p=5e-04
## Score (logrank) test = 18.48  on 3 df,   p=3e-04
```

```
ggforest(apo.mod2, data = apo.dat)
```

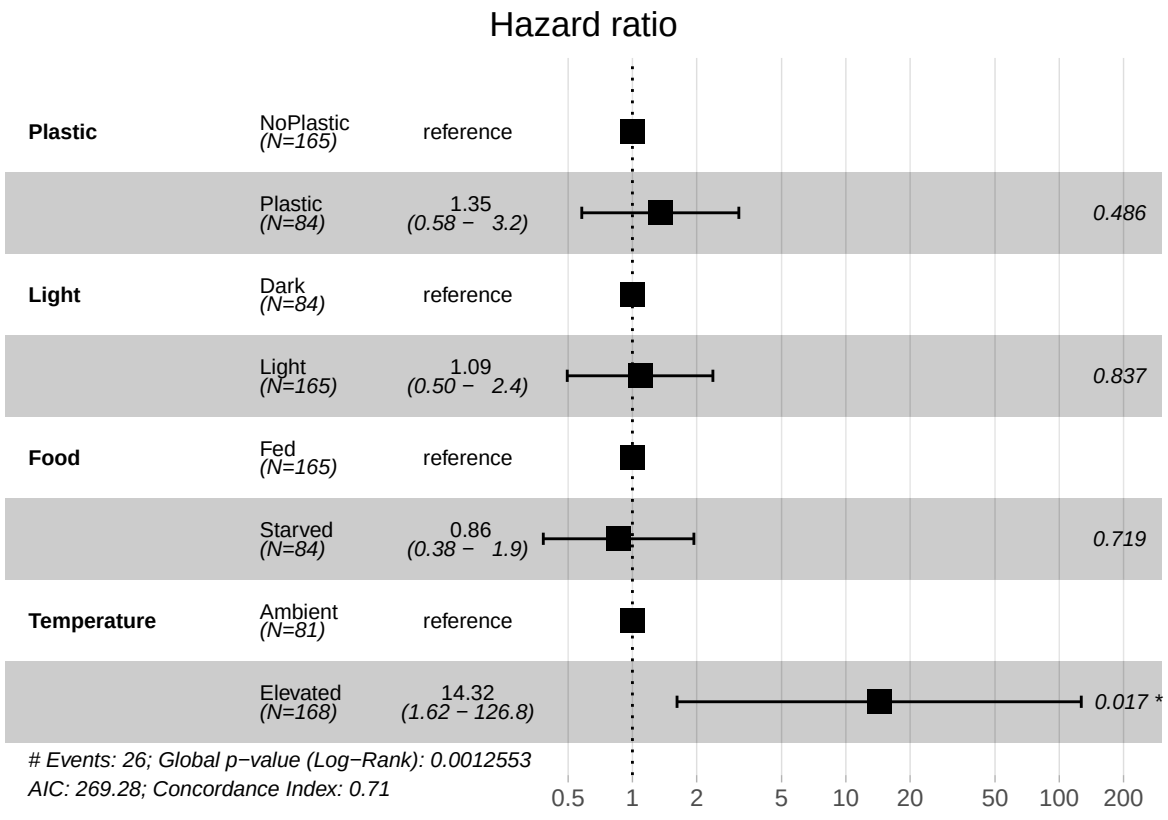



```
# And we will repeat those same steps for our sym fragments
sym.mod <- coxph(Surv(Time, Death) ~ Plastic + Light + Food + Temperature, data = sym.dat)
summary(sym.mod)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food +
##     Temperature, data = sym.dat)
##
##   n= 249, number of events= 26
##
##                      coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic     0.3008    1.3510   0.4323  0.696   0.4865
```

```
## LightLight           0.0826    1.0861   0.4008  0.206   0.8367
## FoodStarved         -0.1491    0.8615   0.4145 -0.360   0.7190
## TemperatureElevated  2.6619   14.3235   1.1126  2.393   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                     exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic         1.3510    0.74019    0.5790     3.152
## LightLight             1.0861    0.92072    0.4951     2.383
## FoodStarved            0.8615    1.16081    0.3823     1.941
## TemperatureElevated   14.3235    0.06982    1.6181   126.790
##
## Concordance= 0.706  (se = 0.038 )
## Likelihood ratio test= 17.96  on 4 df,   p=0.001
## Wald test            = 8.01  on 4 df,   p=0.09
## Score (logrank) test = 13.98  on 4 df,   p=0.007
```

```
ggforest(sym.mod, data=sym.dat)
```

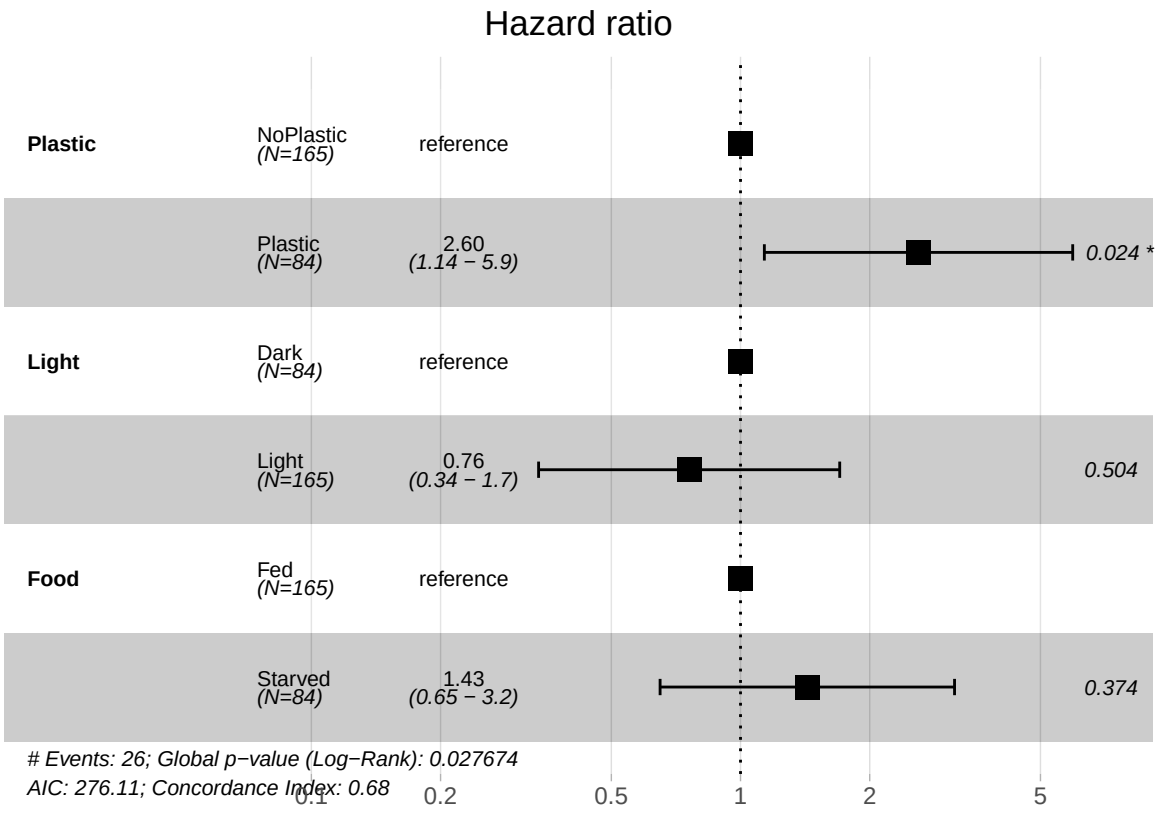



```
# now remove temperature
sym.mod2 <- coxph(Surv(Time, Death) ~ Plastic + Light + Food , data = sym.dat)
summary(sym.mod2)
```

```
## Call:
## coxph(formula = Surv(Time, Death) ~ Plastic + Light + Food, data = sym.dat)
```

```
##
##   n= 249, number of events= 26
##
##                    coef exp(coef) se(coef)      z Pr(>|z|)
## PlasticPlastic  0.9552    2.5991   0.4221  2.263   0.0236 *
## LightLight     -0.2757    0.7591   0.4122 -0.669   0.5036
## FoodStarved     0.3584    1.4311   0.4030  0.889   0.3738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                exp(coef) exp(-coef) lower .95 upper .95
## PlasticPlastic    2.5991     0.3848    1.1364     5.944
## LightLight        0.7591     1.3174    0.3384     1.703
## FoodStarved       1.4311     0.6988    0.6495     3.153
##
## Concordance= 0.676  (se = 0.05 )
## Likelihood ratio test= 9.13  on 3 df,   p=0.03
## Wald test            = 8.8  on 3 df,   p=0.03
## Score (logrank) test = 9.65  on 3 df,   p=0.02
```

```
ggforest(sym.mod2, data = sym.dat)
```



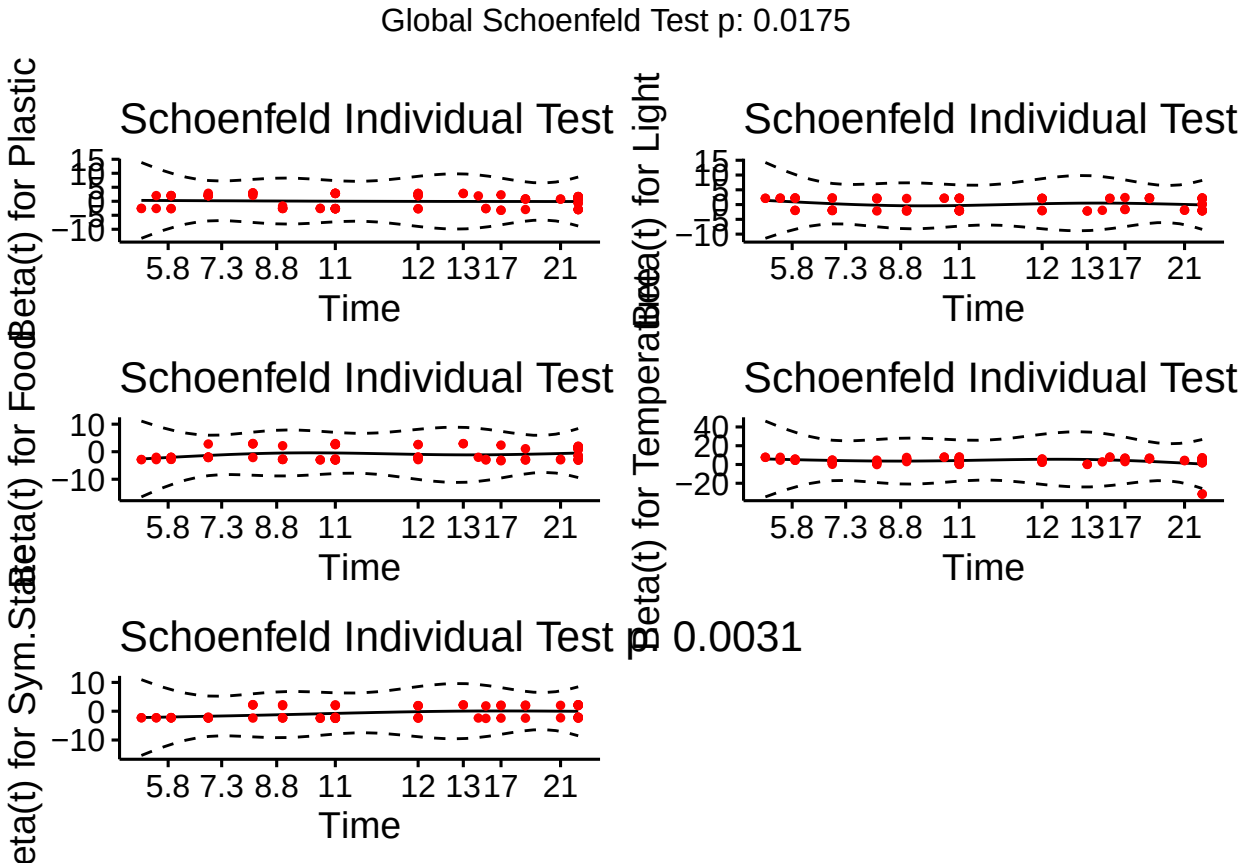
Testing model assumptions

```
# We'll start out with our first full model that includes all of our stress parameters and sym state, m

# Let's test for proportional hazards
all.hazard <- cox.zph(multiv.mod)
print(all.hazard)
```

```
##                chisq df      p
## Plastic       1.1350  1 0.2867
## Light         0.0766  1 0.7819
## Food          1.1981  1 0.2737
## Temperature   2.3381  1 0.1262
## Sym.State     8.7224  1 0.0031
## GLOBAL       13.7186  5 0.0175
```

```
# If the p vale for this test is significant for any of the co-variates (ie the variable p value is les
```

```
# Let's plot the assumptions graph
ggcoxzph(all.hazard)
```

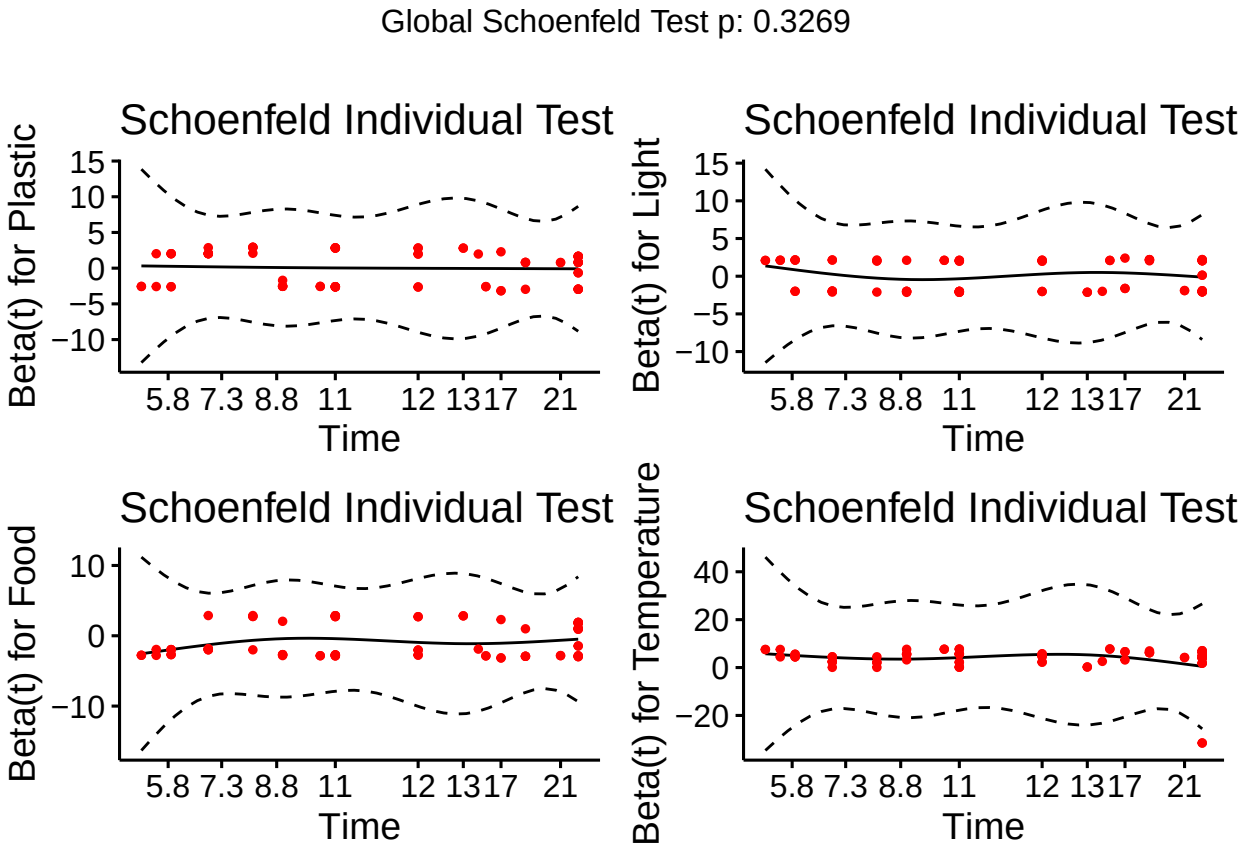


```
# lets test our reduced model that has all of our stress parameters
treat.hazard <- cox.zph(m1)
print(treat.hazard)
```

```
##                chisq df    p
## Plastic     1.1083  1 0.29
## Light       0.0826  1 0.77
## Food        1.2546  1 0.26
## Temperature 2.3529  1 0.13
## GLOBAL      4.6347  4 0.33
```

```
# Yes! so even though removing sym.state from our cox model showed that it was worse (in the LRT anova)
```

```
# and plot
ggcoxzph(treat.hazard)
```




```
# If we see a pattern with the variable over time, that is an indication that the assumptions cannot be
```

```
# and for our full model without temperature
tmp.hazard <- cox.zph(tmp.mod)
tmp.hazard
```

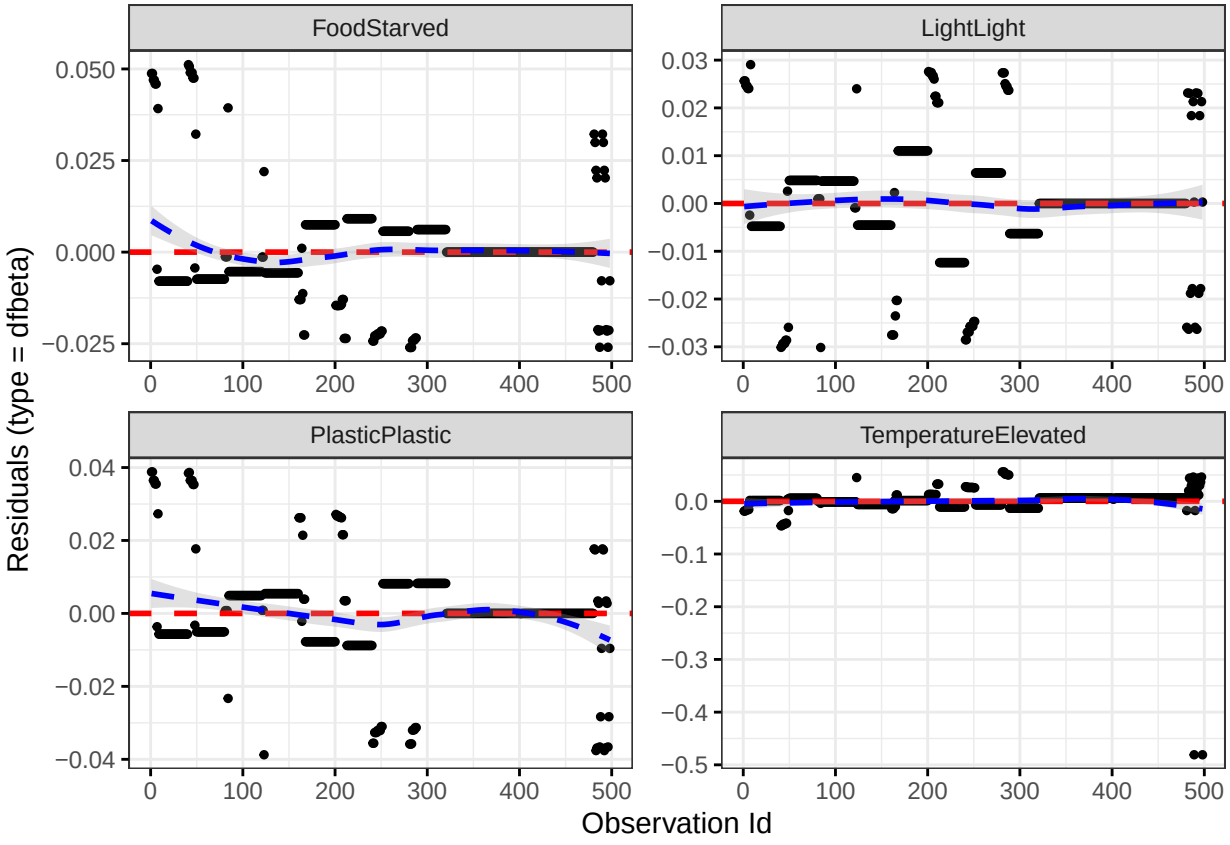
```
##         chisq df     p
```

```
## Plastic 0.036  1 0.850
## Light   0.308  1 0.579
## Food    3.696  1 0.055
## GLOBAL  3.765  3 0.288
```

```
# Let's take a deeper look at our m1 model (all stress parameters without sym state)

# estimated changes in the regression coefficients upon deleting each observation in turn
ggcoxdiagnostics(m1, type = "dfbeta",
                 linear.predictions = FALSE, ggtheme = theme_bw())
```

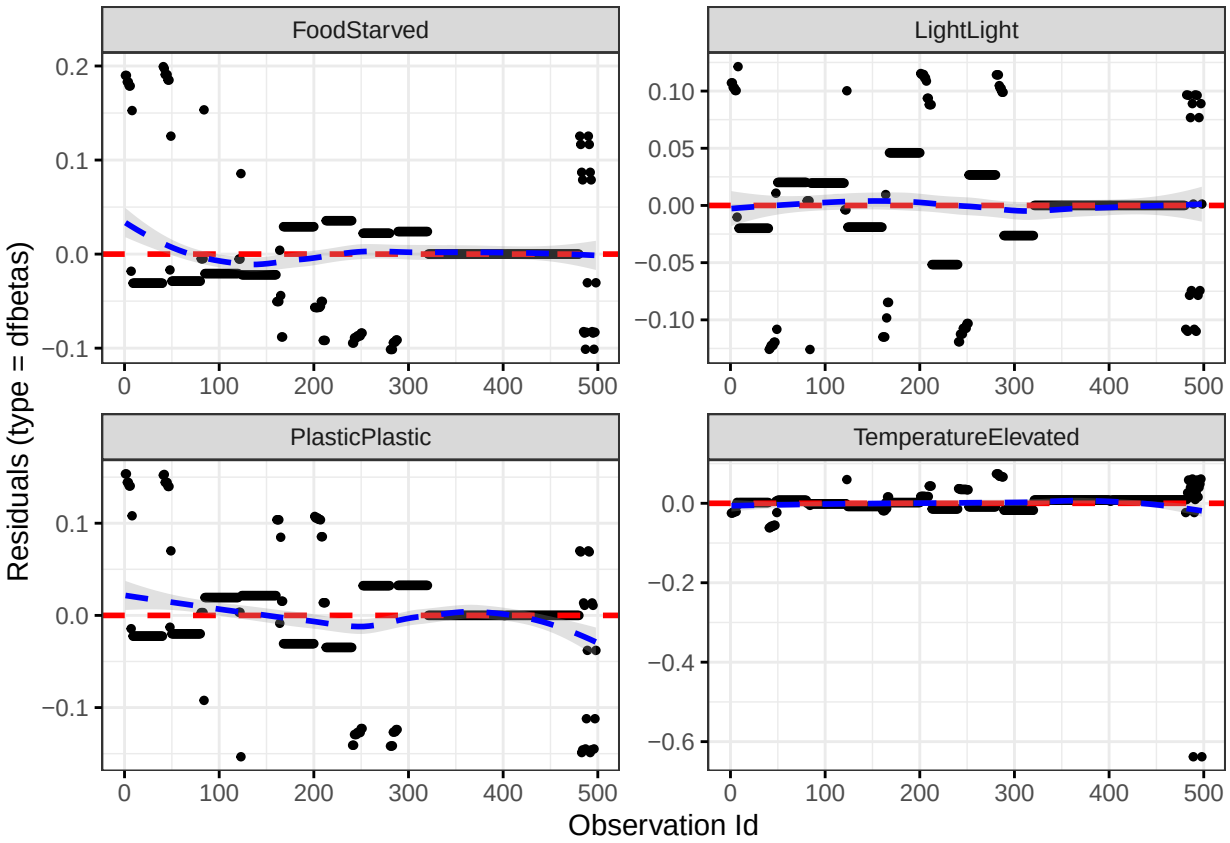
```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## `geom_smooth()` using formula 'y ~ x'
```



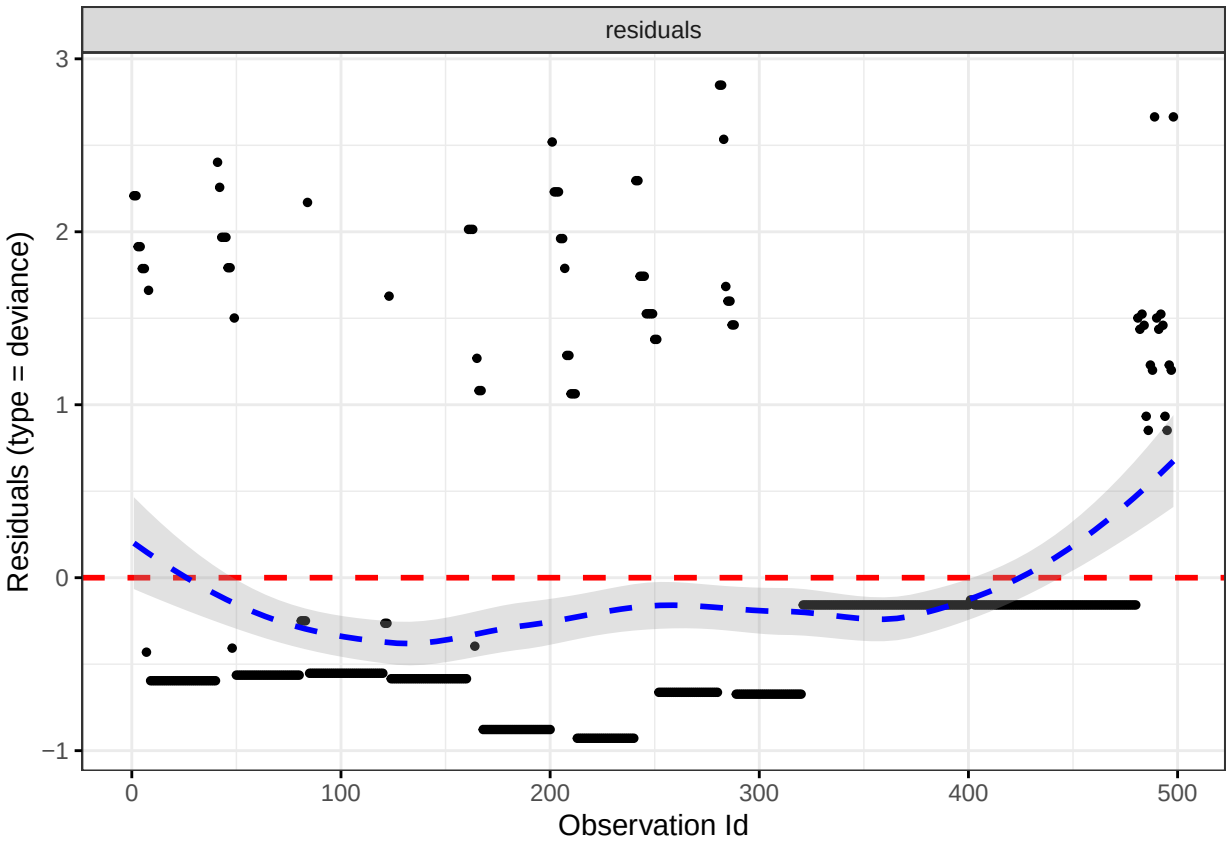

```
# estimated changes in the coefficients divided by their standard errors
ggcoxdiagnostics(m1, type = "dfbetas",
                 linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
#  check outliers by visualizing the deviance residuals. The deviance residual is a normalized transfor
ggcoxdiagnostics(m1, type = "deviance",
                 linear.predictions = FALSE, ggtheme = theme_bw())
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# not very symmetrical around 0
```

```
# can we do the same for our single model parameters
plastic.hazard <- cox.zph(plastic.mod)
plastic.hazard
```

```
##         chisq df    p
## Plastic   0.1  1 0.75
## GLOBAL    0.1  1 0.75
```

```
light.hazard <- cox.zph(light.mod)
light.hazard
```

```
##        chisq df    p
## Light  0.501  1 0.48
## GLOBAL 0.501  1 0.48
```

```
food.hazard <- cox.zph(food.mod)
food.hazard
```

```
##        chisq df     p
## Food    3.39  1 0.066
## GLOBAL  3.39  1 0.066
```

```
temp.hazard <- cox.zph(temperature.mod)
temp.hazard
```

```
##             chisq df    p
## Temperature  2.62  1 0.11
## GLOBAL       2.62  1 0.11
```

```
symst.hazard <- cox.zph(symst.mod)
symst.hazard ## only one that differs over time, not proportional!
```

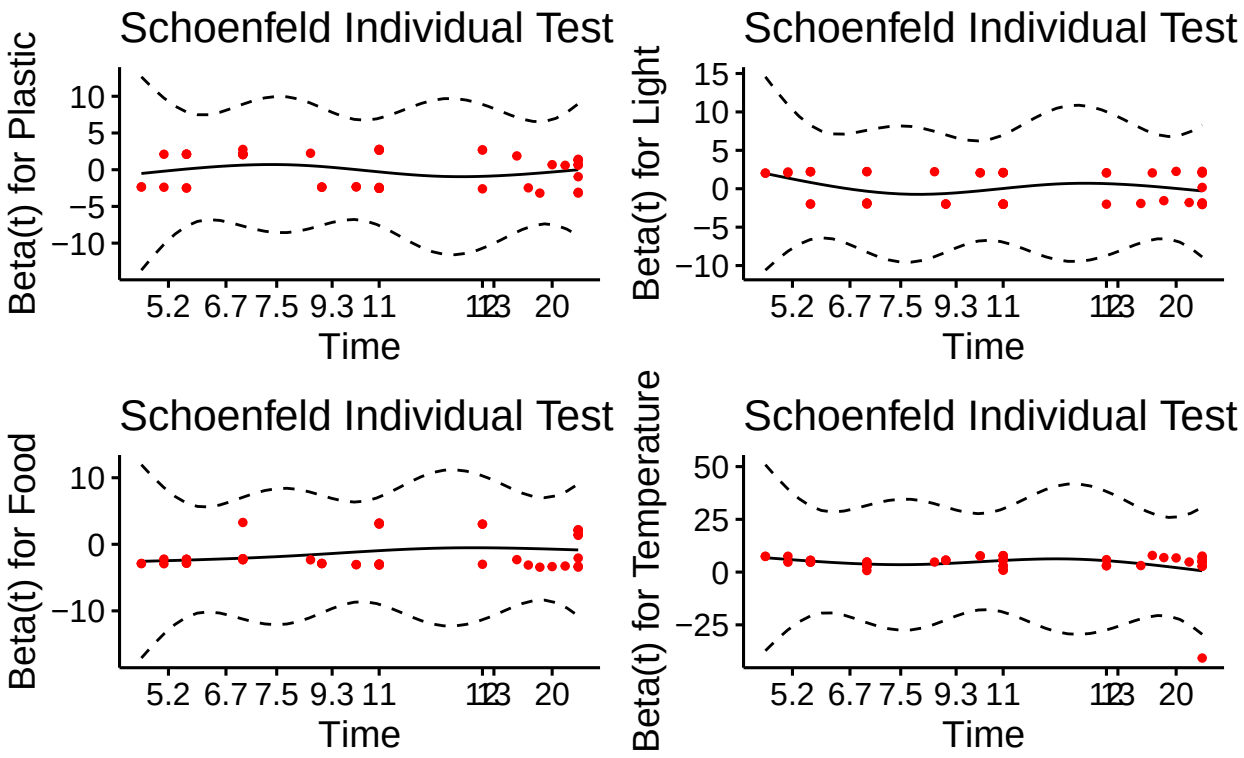
```
##           chisq df      p
## Sym.State  9.14  1 0.0025
## GLOBAL     9.14  1 0.0025
```

```
# let's then assess our parameter estimates by sym state
apo.hazard <- cox.zph(apo.mod)
apo.hazard # all good
```

```
##             chisq df     p
## Plastic     0.932  1 0.334
## Light       0.562  1 0.453
## Food        3.127  1 0.077
## Temperature 1.232  1 0.267
## GLOBAL      5.852  4 0.210
```

```
ggcoxzph(apo.hazard)
```

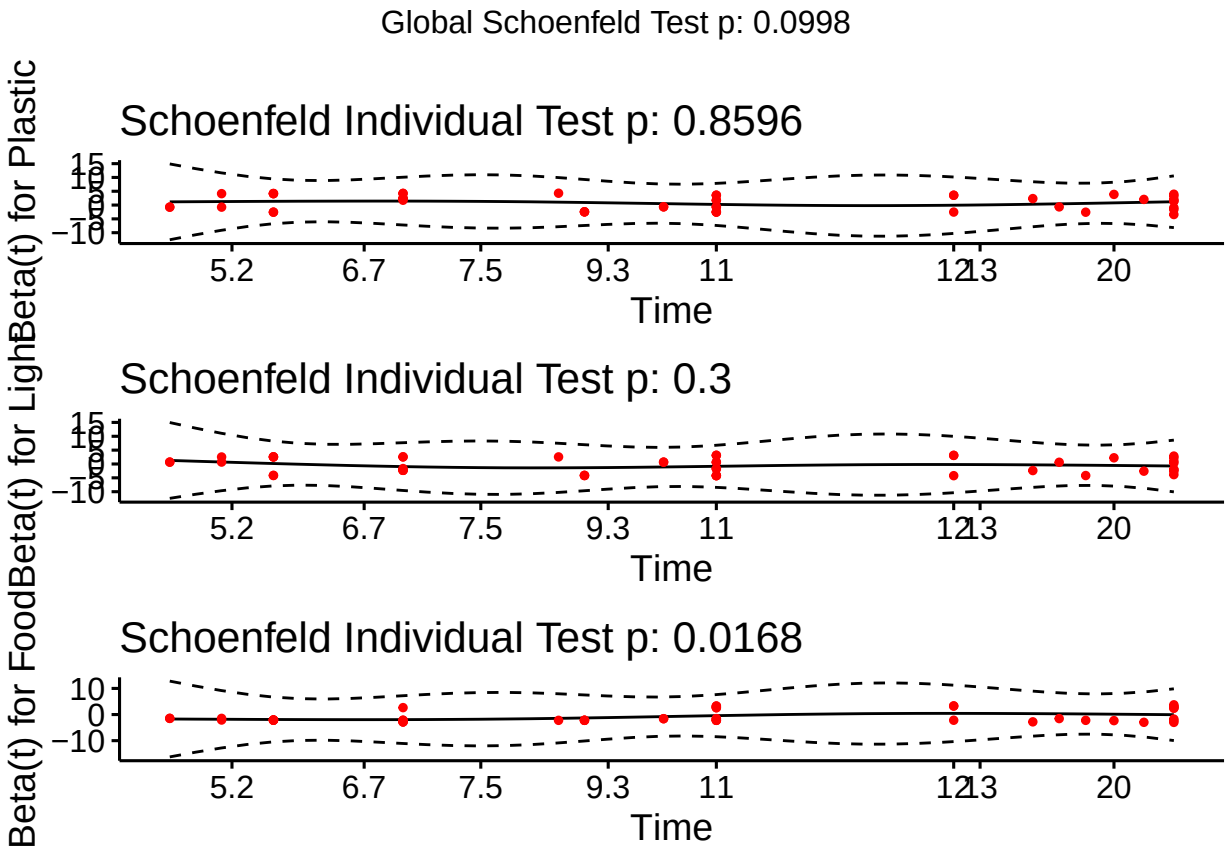
Global Schoenfeld Test p: 0.2105




```
apo2.hazard <- cox.zph(apo.mod2)
apo2.hazard # all good
```

```
##          chisq df     p
## Plastic 0.0313  1 0.860
## Light   1.0741  1 0.300
## Food    5.7132  1 0.017
## GLOBAL  6.2560  3 0.100
```

```
ggcoxzph(apo2.hazard)
```

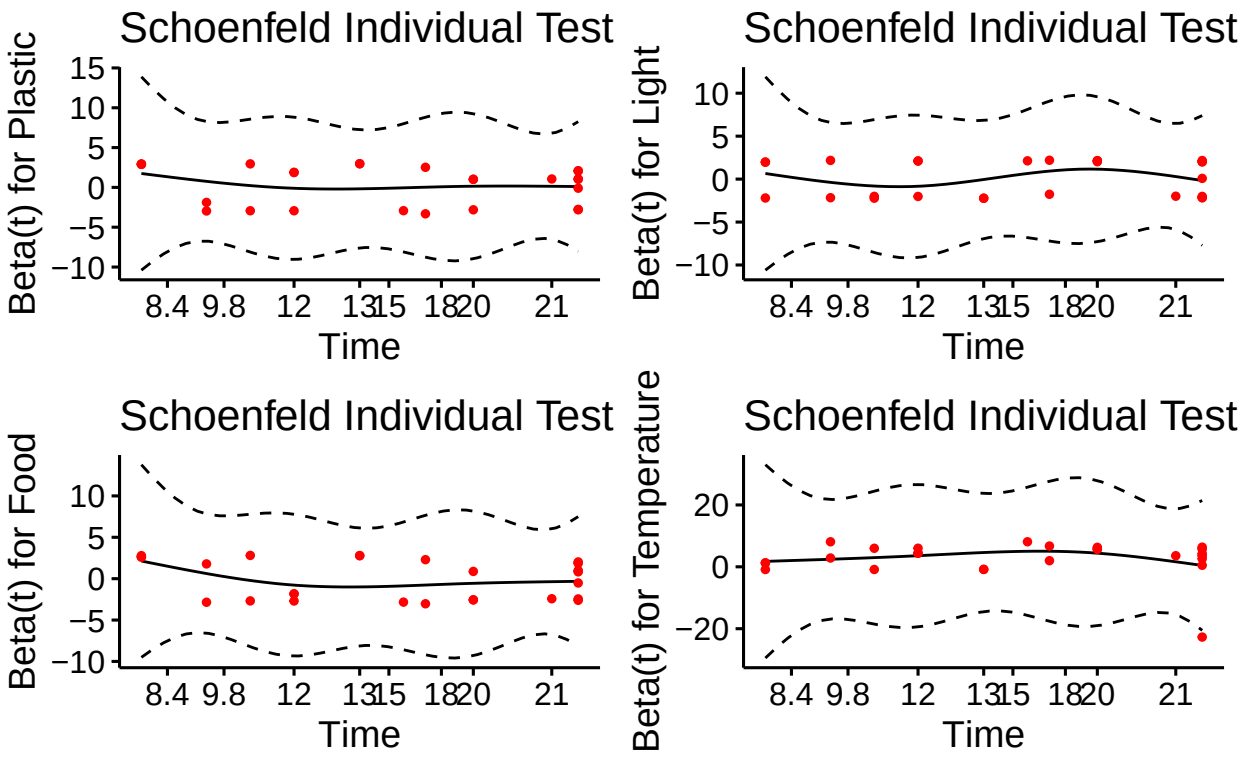



```
# sym
sym.hazard <- cox.zph(sym.mod)
sym.hazard #nice
```

```
##             chisq df    p
## Plastic     0.691  1 0.41
## Light       0.114  1 0.74
## Food        1.849  1 0.17
## Temperature 1.182  1 0.28
## GLOBAL      3.356  4 0.50
```

```
ggcoxzph(sym.hazard)
```

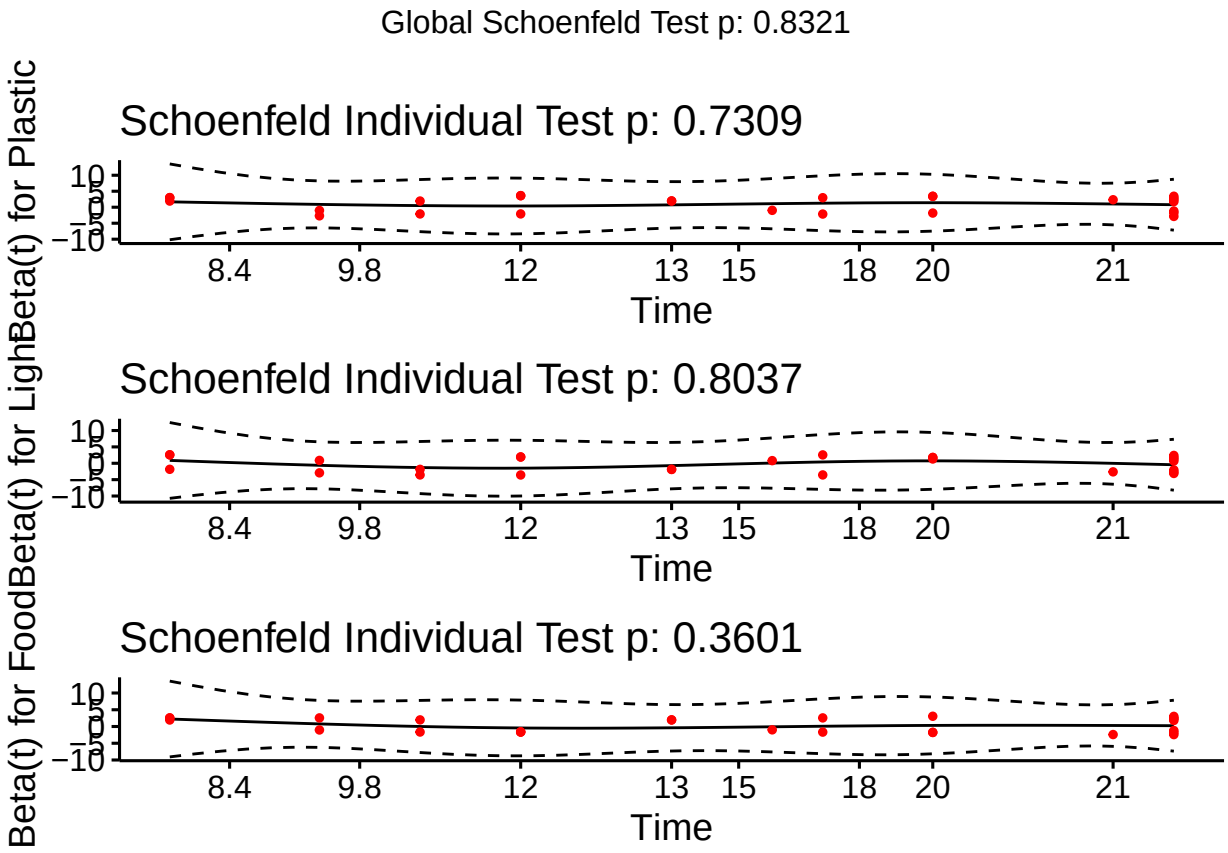
Global Schoenfeld Test p: 0.5001




```
sym2.hazard <- cox.zph(sym.mod2)
sym2.hazard #nice
```

```
##             chisq df    p
## Plastic    0.1183  1 0.73
## Light      0.0618  1 0.80
## Food       0.8377  1 0.36
## GLOBAL     0.8723  3 0.83
```

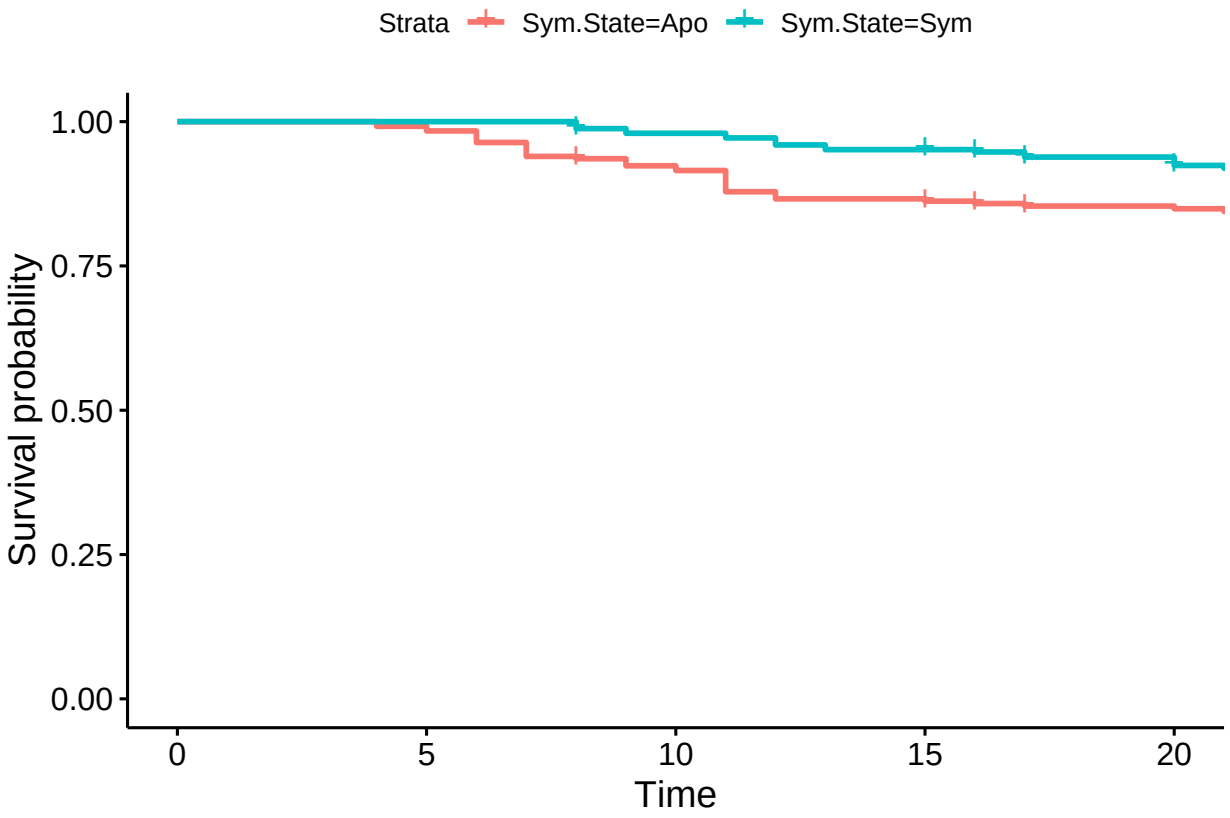
```
ggcoxzph(sym2.hazard)
```



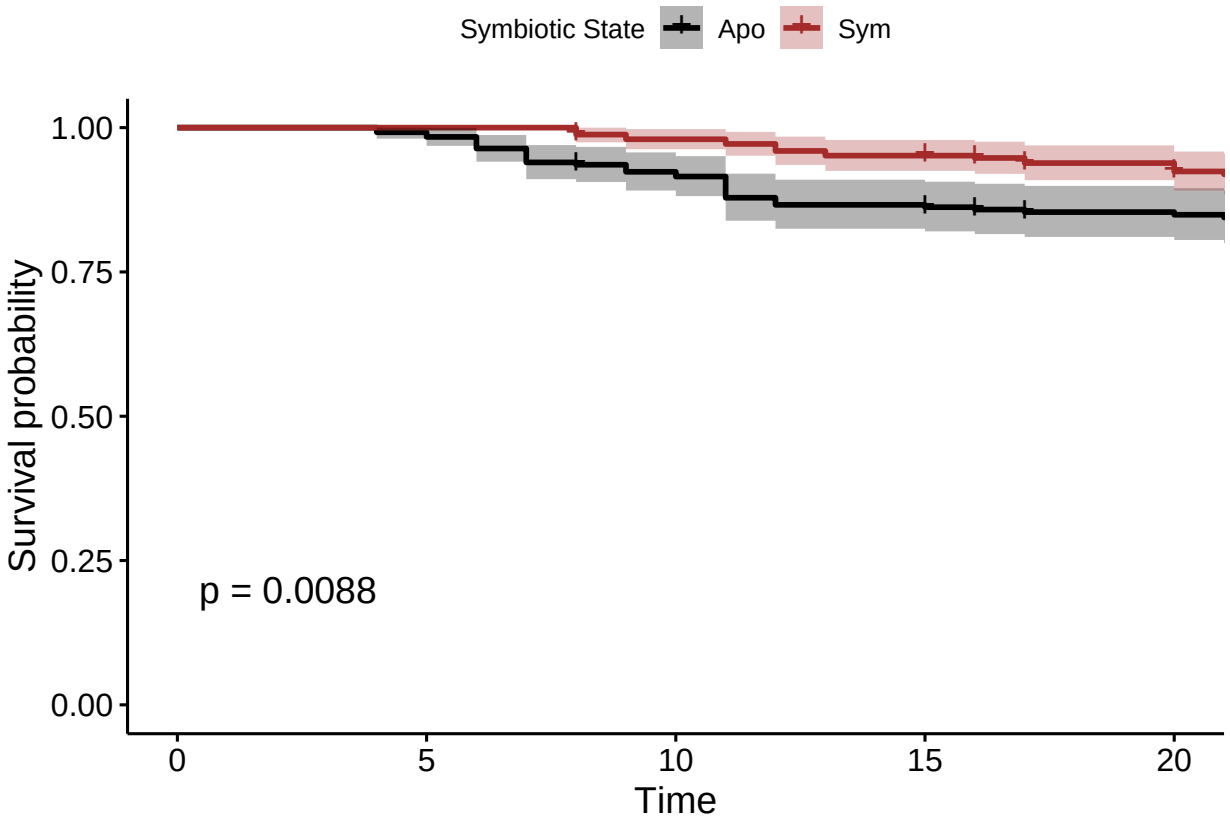

```
treat.col <- palette(brewer.pal(n=9, name = "Set2"))
```

```
## Warning in brewer.pal(n = 9, name = "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```

```
# Let's visualize a model using ggplot
fit.1 <- survfit(Surv(Time, Death)~Sym.State, data=surv.dat)
ggsurvplot(fit.1)
```



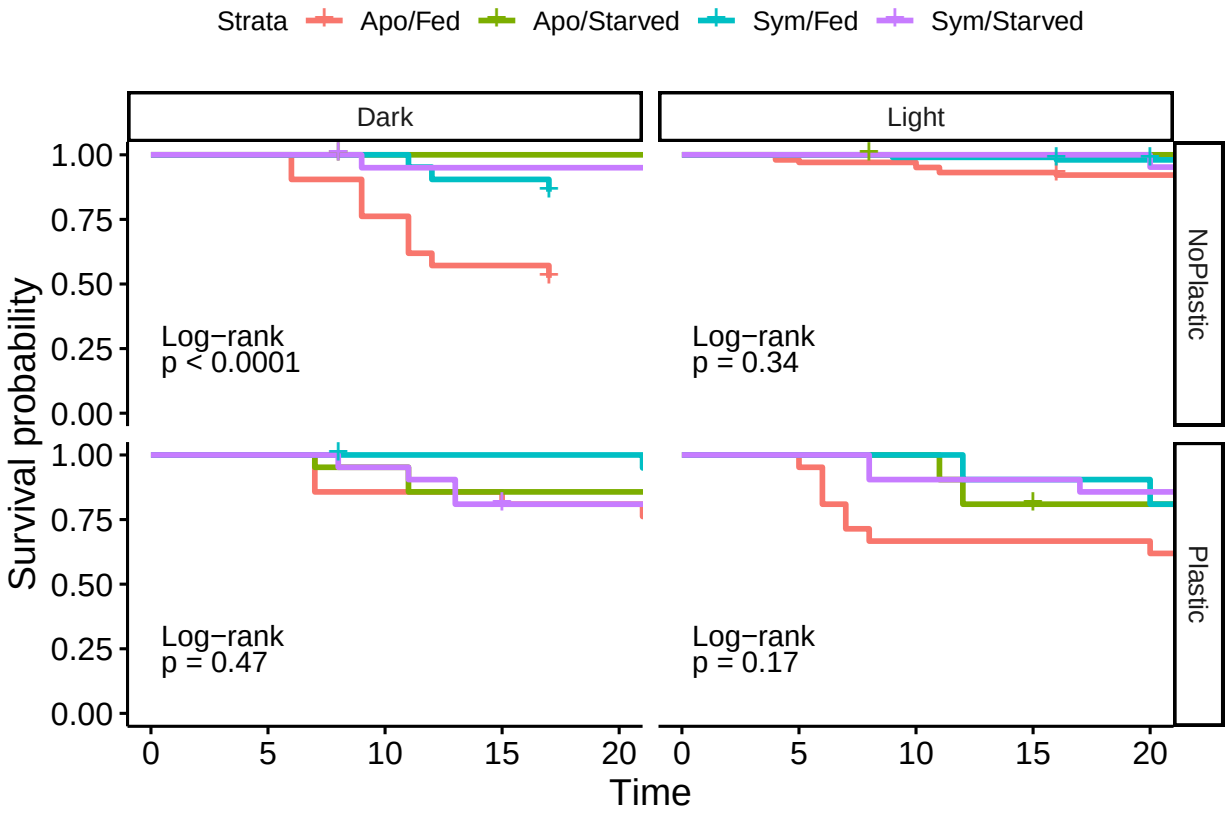

```
ggsurvplot(fit.1, conf.int = TRUE,
           pval = TRUE,
           legend.labs = c("Apo", "Sym"),
           legend.title = "Symbiotic State",
           palette = c("black", "brown"))
```



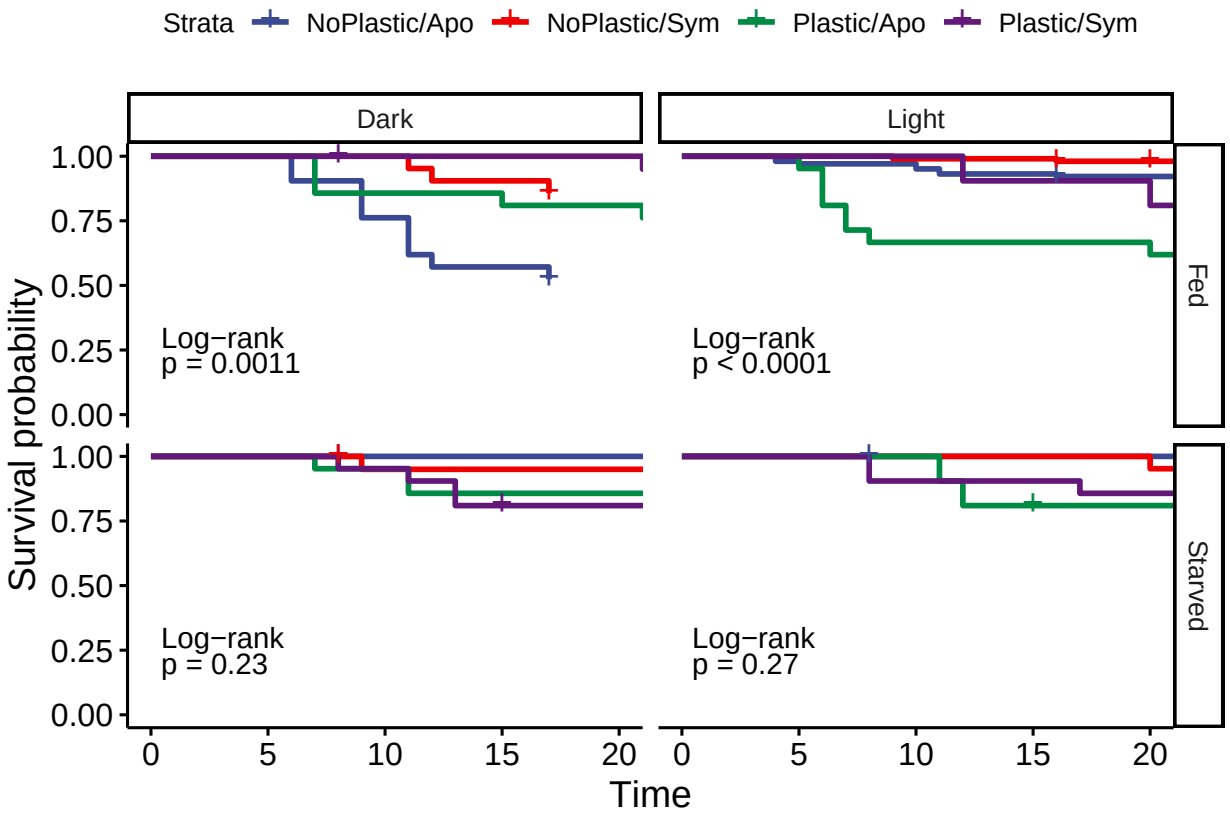

```
#ggsurvplot(survfit(all.mod), data=surv.dat, palette = treat.col, ggtheme = theme_minimal())


# food & sym. state
sig.fit <- survfit(Surv(Time, Death) ~ Sym.State+Food, data=surv.dat)
p <- ggsurvplot_facet(sig.fit, surv.dat,
                  facet.by= c("Plastic", "Light"),
                  short.panel.labs = TRUE,
                  palette = "aas",
                  pval = TRUE,
                  pval.method = TRUE,
                  legend.labs = c("Apo/Fed", "Apo/Starved", "Sym/Fed", "Sym/Starved"))
```

```
## Warning: ‘as.tibble()‘ was deprecated in tibble 2.0.0.
## Please use ‘as_tibble()‘ instead.
## The signature and semantics have changed, see ‘?as_tibble‘.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.

## Warning: ‘select_()‘ was deprecated in dplyr 0.7.0.
## Please use ‘select()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```

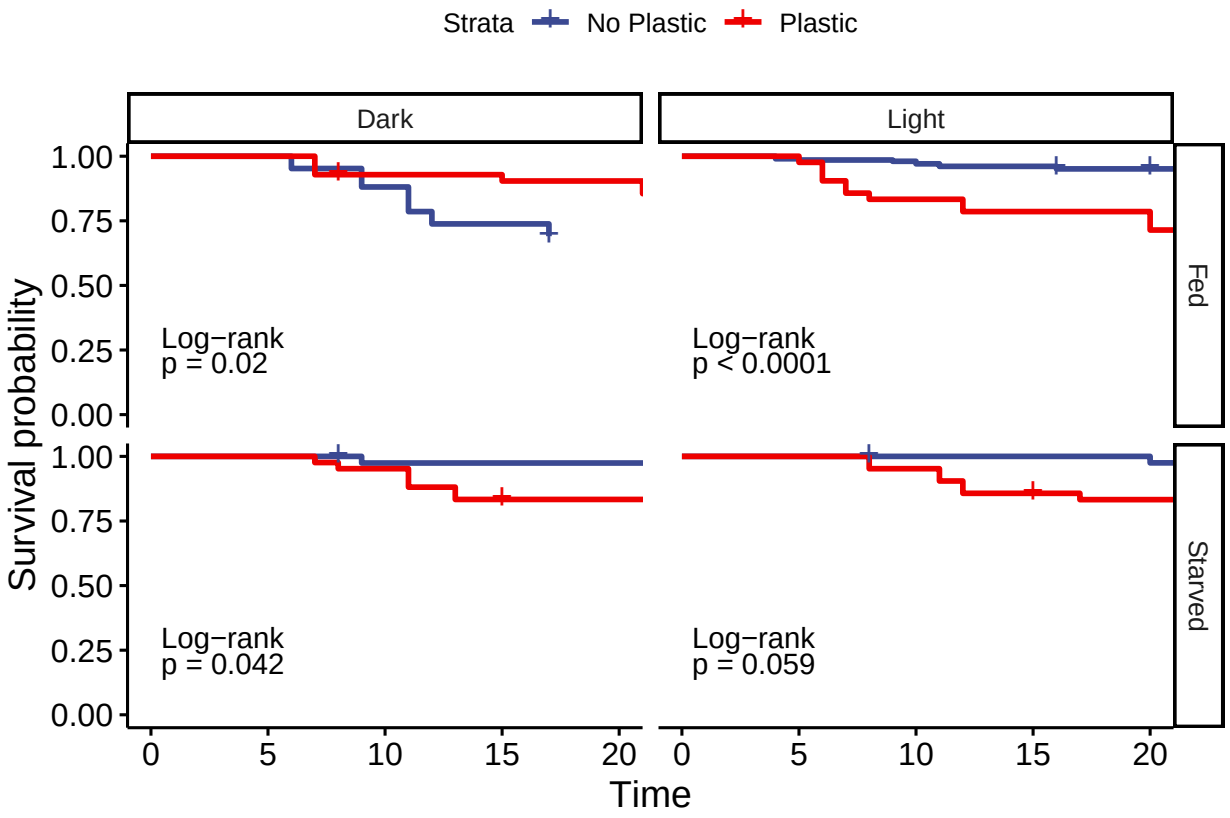
```
p
```



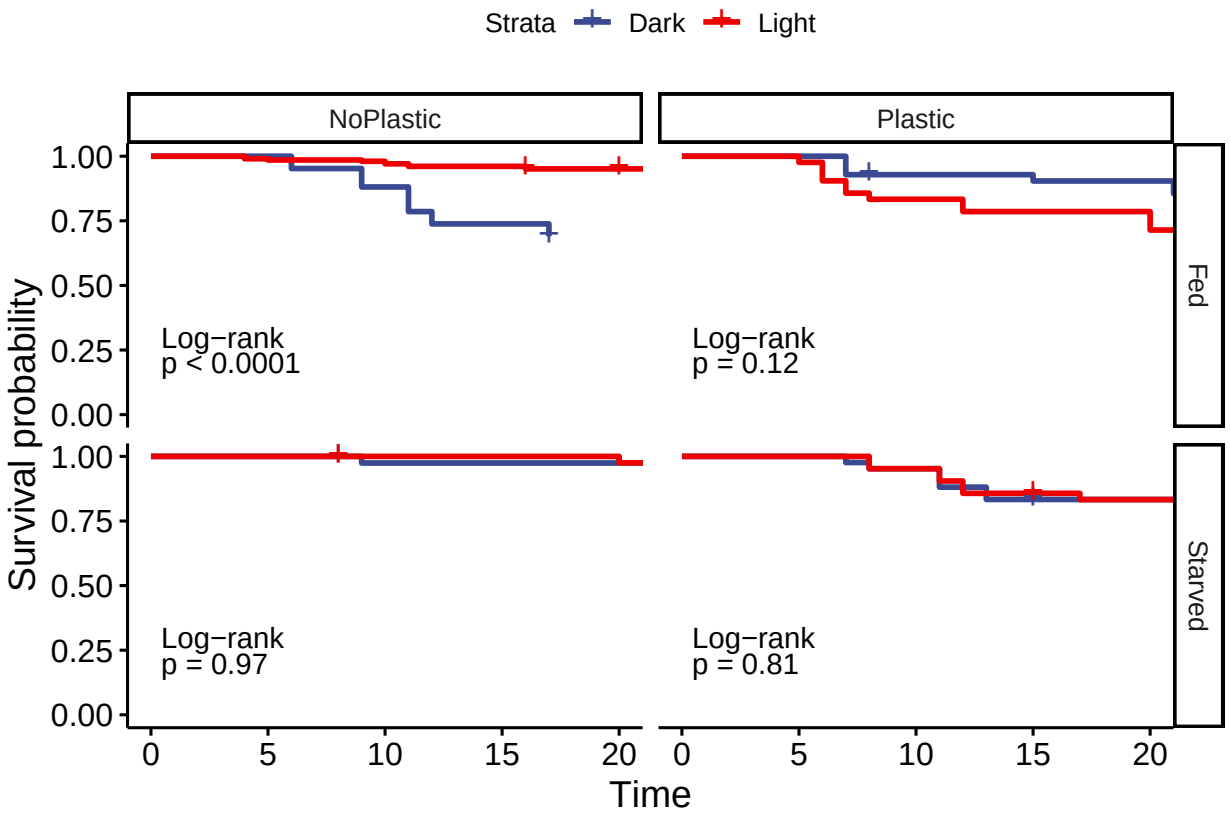

```
# p + lims(y=c(0.25,1.0))


# facet plot with all 4 variables
fit <- survfit( Surv(Time, Death) ~ Plastic + Sym.State, data = surv.dat )
ggsurvplot_facet(fit, surv.dat,
                 facet.by = c("Food", "Light"),
                 palette = "aaas",
                 pval = TRUE, pval.method = TRUE,
                 short.panel.labs = TRUE,
                 legend.labs = c("NoPlastic/Apo", "NoPlastic/Sym", "Plastic/Apo", "Plastic/Sym"))
```



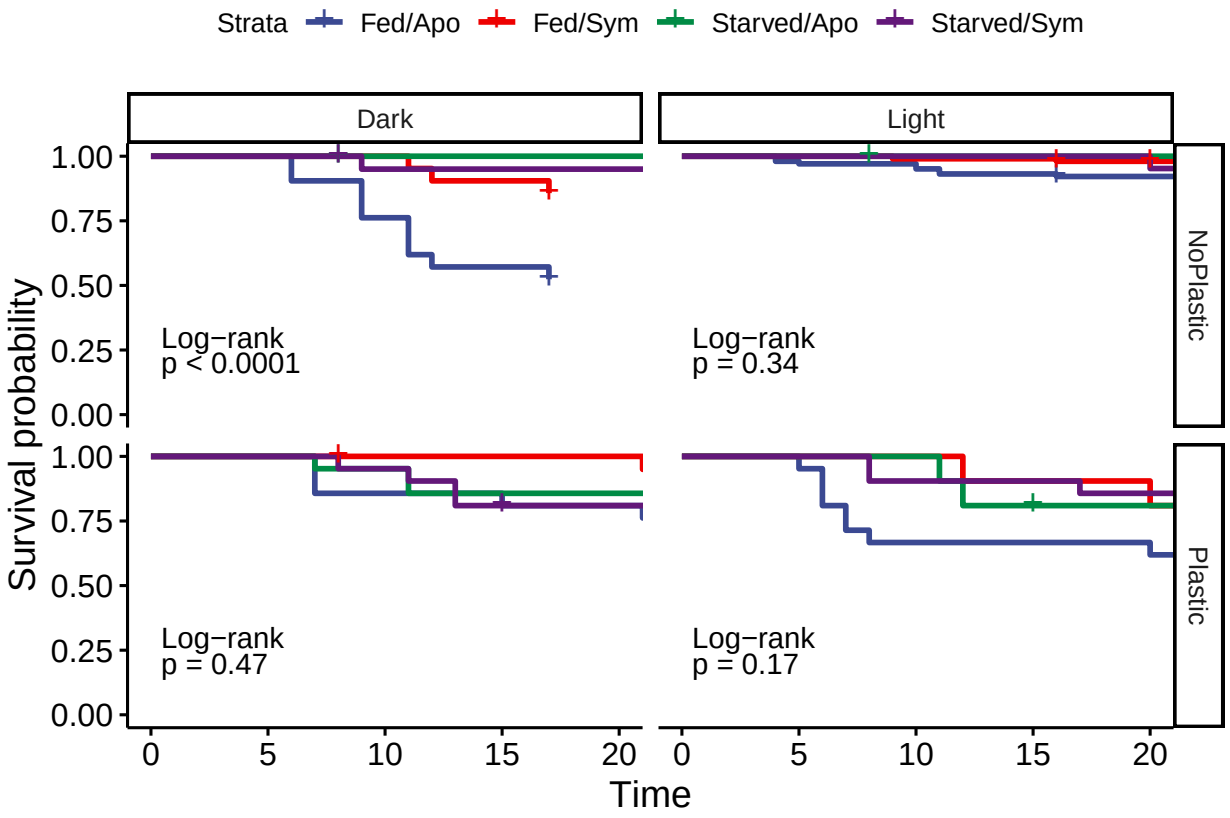

```
# facet plot not including sym state
fit2 <- survfit( Surv(Time, Death) ~ Plastic, data = surv.dat )
ggsurvplot_facet(fit2, surv.dat,
                 facet.by = c("Food", "Light"),
                 palette = "aaas",
                 pval = TRUE, pval.method = TRUE,
                 short.panel.labs = TRUE,
                 legend.labs = c("No Plastic", "Plastic"))
```



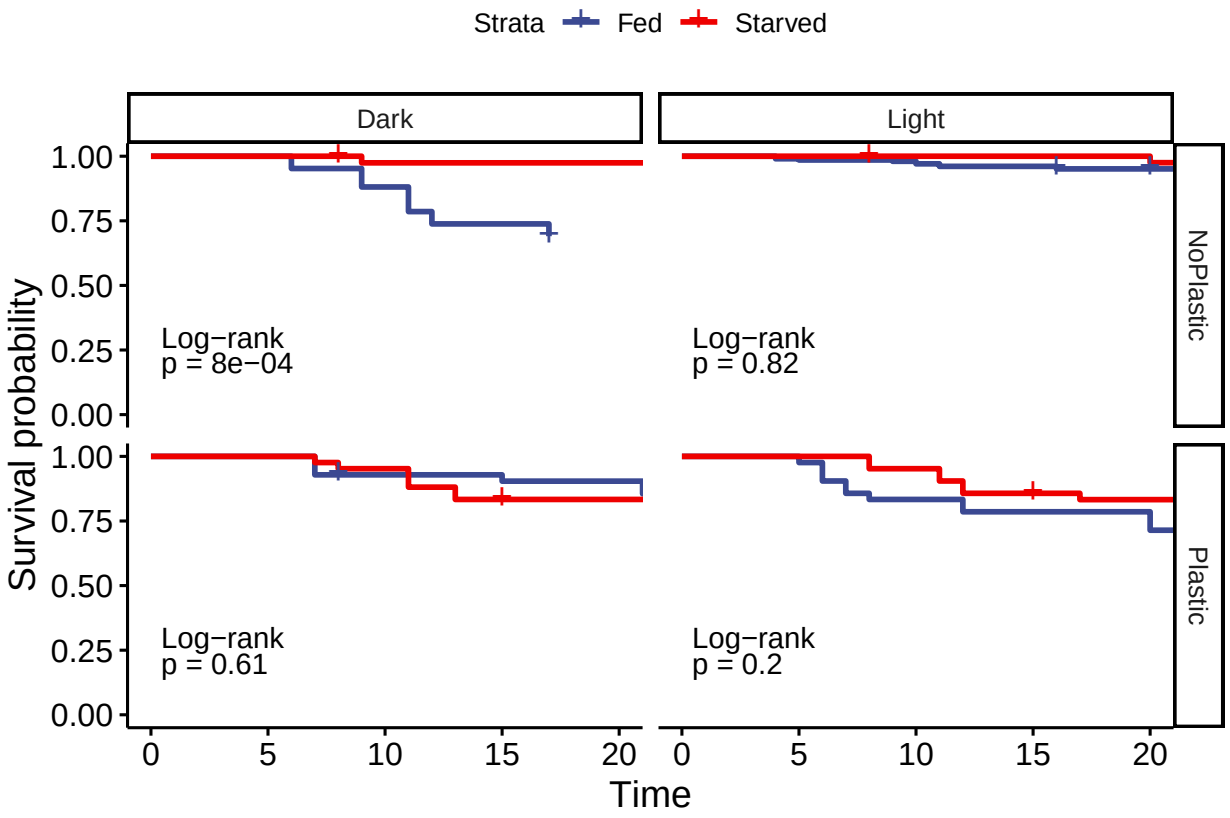

```
# facet plot not including sym state
fit3 <- survfit( Surv(Time, Death) ~ Light, data = surv.dat )
ggsurvplot_facet(fit3, surv.dat,
                 facet.by = c("Food", "Plastic"),
                 palette = "aaas",
                 pval = TRUE, pval.method = TRUE,
                 short.panel.labs = TRUE)
```




```
# facet plot with all 4 variables
fit4 <- survfit( Surv(Time, Death) ~ Food + Sym.State, data = surv.dat )
ggsurvplot_facet(fit4, surv.dat,
                 facet.by = c("Plastic", "Light"),
                 palette = "aaas",
                 pval = TRUE, pval.method = TRUE,
                 short.panel.labs = TRUE,
                 legend.labs = c("Fed/Apo", "Fed/Sym", "Starved/Apo", "Starved/Sym"))
```




```
# facet plot sym state by plastic and light
fit5 <- survfit( Surv(Time, Death) ~ Food, data = surv.dat )
ggsurvplot_facet(fit5, surv.dat,
                 facet.by = c("Plastic", "Light"),
                 palette = "aaas",
                 pval = TRUE, pval.method = TRUE,
                 short.panel.labs = TRUE)
```


Strata
Fed
Starved
Dark
Light
NoPlastic
Plastic
Survival probability
1.00
0.75
0.50
0.25
0.00
Log−rank
p = 8e−04
Log−rank
p = 0.82
Log−rank
p = 0.61
Log−rank
p = 0.2
0
5
10
15
20
Time