

Survival of C.elegans with three different media

Jonas Gehrlein

12 nov 2018

code for fitting Kaplan-Meier and log-rank test and for displaying survival curves for each type of media

```
data <- read.table(here('data', 'Worm_4days.csv'), header = TRUE, sep = ";", dec = ",")
head(data)
```

```
##   i..Time Status Group Replicate
## 1      4      1  FG13          1
## 2      6      1  FG13          1
## 3      6      1  FG13          1
## 4      8      1  FG13          1
## 5      8      1  FG13          1
## 6     10      1  FG13          1
```

here() should show that your position is in the folder 7.semester else select the active project to be 7.semester in the upper right corner.

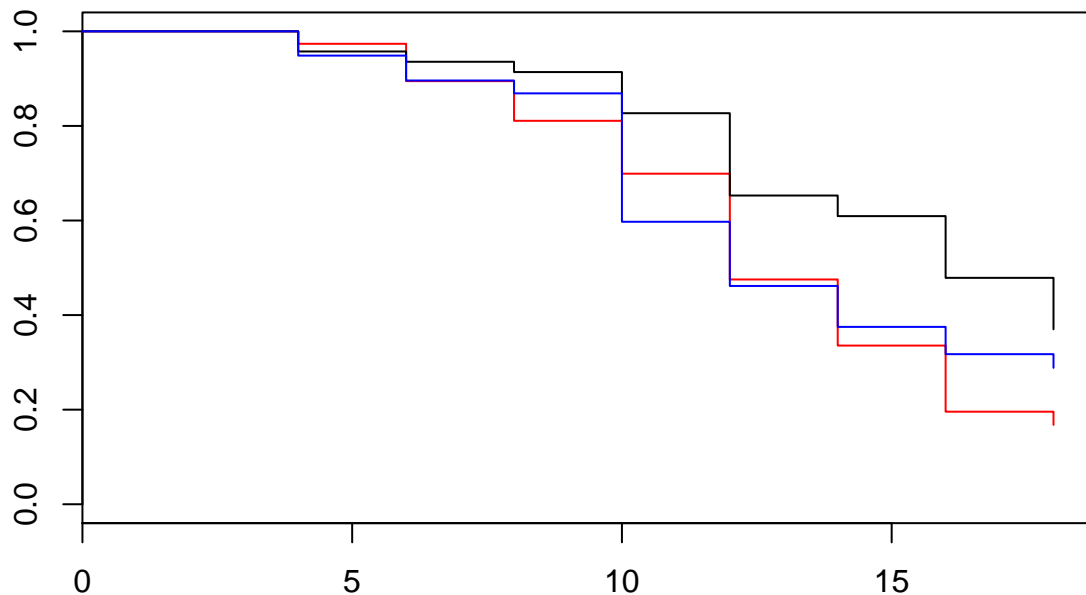
Or make a new .Rproj file in the folder 7.semester. There is a problem with the numbers of the worms so we change that with col.names()

```
colnames(data)<-c("Time", "status", "media", 'Replicate')
head(data)
```

```
##   Time status media Replicate
## 1    4      1  FG13          1
## 2    6      1  FG13          1
## 3    6      1  FG13          1
## 4    8      1  FG13          1
## 5    8      1  FG13          1
## 6   10      1  FG13          1
```

Then we create a survival object with the survival package and make a kaplan-meier curve

```
data$Survobj <- with(data, Surv(Time, event = status))
km <- survfit(Survobj ~ media, data = data, conf.type = "log-log", error = "greenwood")
plot(km, col = c("red", "black", "blue"))
```



Shows errorbars and lineplot based on one of Anders papers <https://onlinelibrary.wiley.com/doi/full/10.1111/accel.12165> This is done by extracting the surviving proportion, standard errors and time from `survfit()`. and adding a startpoint where the survival is 100% at 0 hours.

```
df_fly_13 <- data.frame(c(0,km$time[1:8]),c(1,km$urv[1:8]),c(0,km$std.err[1:8]))
colnames(df_fly_13) <- c('Time','Surv','Std.error')
df_fly_20 <- data.frame(c(0,km$time[9:16]),c(1,km$urv[9:16]),
                        c(0,km$std.err[9:16]))
colnames(df_fly_20) <- c('Time','Surv','Std.error')
df_OP50 <- data.frame(c(0,km$time[17:24]),c(1,km$urv[17:24]),
                      c(0,km$std.err[17:24]))
colnames(df_OP50) <- c('Time','Surv','Std.error')

plot(df_fly_13$Time[2:9],df_fly_13$Surv[2:9], pch = 16, cex = 1.2, xlab = 'Heatstress (hours)',
     ylab = 'Surviving fraction',
     main = expression('Survival heat stress for'~italic(C.elegans)),
     xlim = c(0,22), ylim = c(0,1), xaxp = c(0,22,11))
lines(df_fly_13$Time,df_fly_13$Surv)
arrows(df_fly_13$Time, df_fly_13$Surv-df_fly_13$Std.error, df_fly_13$Time,
       df_fly_13$Surv+df_fly_13$Std.error, length=0.05, angle=90, code=3, col = 'black')
```

```
## Warning in arrows(df_fly_13$Time, df_fly_13$Surv - df_fly_13$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped
```

```
points(df_OP50$Time[2:9],df_OP50$Surv[2:9], pch = 1, cex = 1.2)
lines(df_OP50$Time,df_OP50$Surv, lty = 2)
arrows(df_OP50$Time, df_OP50$Surv-df_OP50$Std.error,
       df_OP50$Time, df_OP50$Surv+df_OP50$Std.error,
       length=0.05, angle=90, code=3, col = 'black')
```

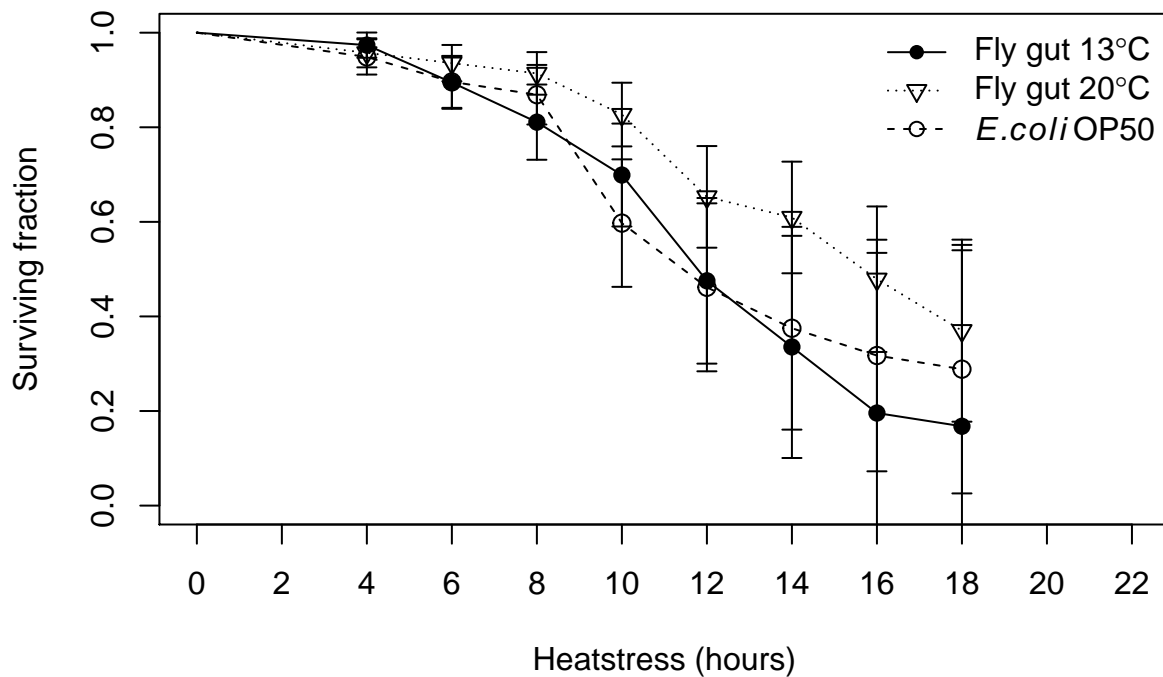
```
## Warning in arrows(df_OP50$Time, df_OP50$Surv - df_OP50$Std.error,
## df_OP50$Time, : zero-length arrow is of indeterminate angle and so skipped
```

```
points(df_fly_20$Time[2:9],df_fly_20$Surv[2:9], pch = 6)
lines(df_fly_20$Time,df_fly_20$Surv,lty = 3)
arrows(df_fly_20$Time, df_fly_20$Surv-df_fly_20$Std.error, df_fly_20$Time,
       df_fly_20$Surv+df_fly_20$Std.error, length=0.05, angle=90, code=3, col = 'black')
```

```
## Warning in arrows(df_fly_20$Time, df_fly_20$Surv - df_fly_20$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped
```

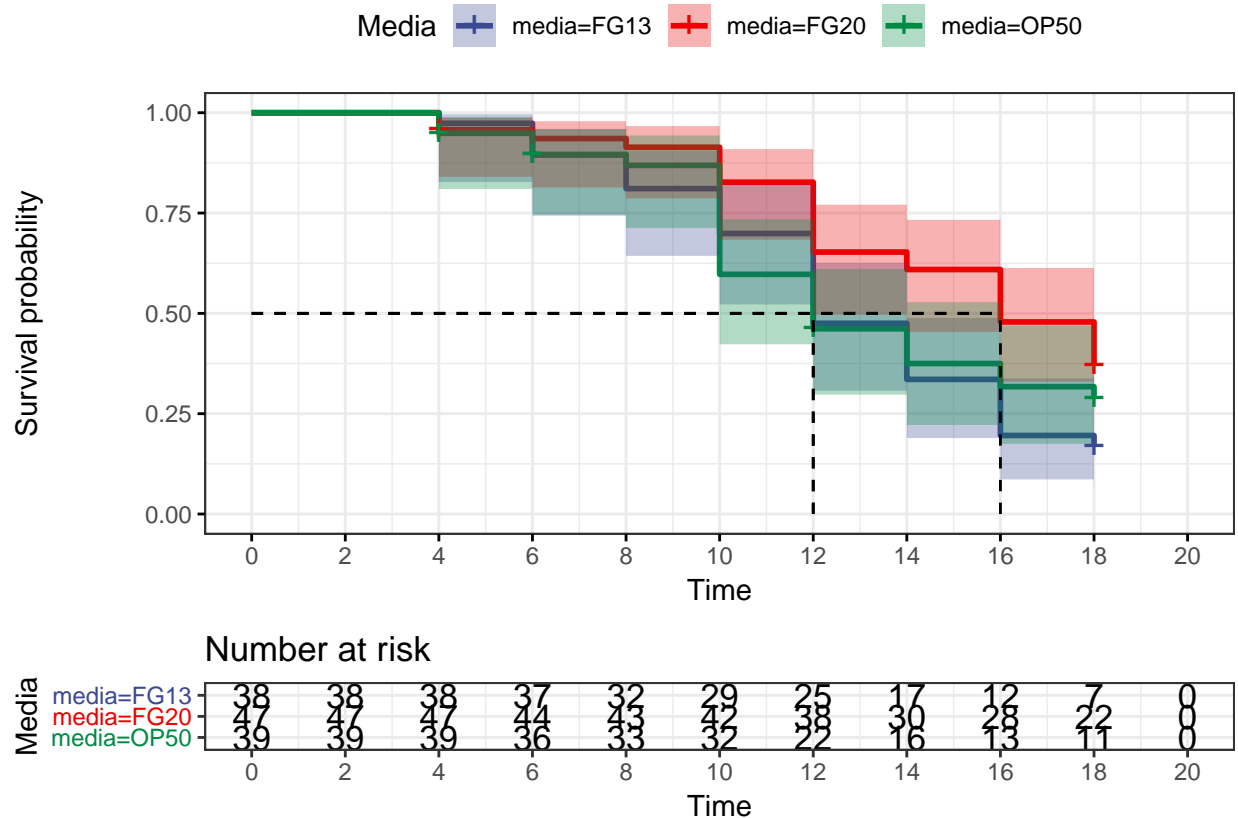
```
legend('topright', pch = c(16,6,1),lty = c(1,3,2),
       legend = c(expression('Fly gut 13'*degree*C),
                    expression('Fly gut 20'*degree*C),
                    expression(italic(E.coli) ~ OP50)), bty = 'n')
```

Survival heat stress for *C.elegans*



Then we create a normal Kaplan-Meier curve

```
ggsurvplot(km,data = data, conf.int = TRUE, ggtheme = theme_bw(),
  risk.table = 0.25, palette = 'aaas', surv.median.line = 'hv',
  legend.title = 'Media',break.x.by = 2)
```



Now we test for difference between the curves with both log-rank and gehan-wilcoxon and with an cox proportional hazard model

```
survdifff(Survobj ~media, data = data, rho = 0)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = data, rho = 0)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 38       30   23.0      2.115    3.498
## media=FG20 47       29   38.4      2.280    5.052
## media=OP50 39       26   23.6      0.238    0.396
##
##  Chisq= 5.6  on 2 degrees of freedom, p= 0.06
```

```
survdifff(Survobj ~media, data = data, rho = 1)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = data, rho = 1)
##
```

```
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 38      21.1      16.7      1.149      2.42
## media=FG20 47      18.6      26.0      2.101      5.80
## media=OP50 39      20.0      17.0      0.531      1.13
##
## Chisq= 5.9  on 2 degrees of freedom, p= 0.05
```

```
fit <- coxph(Survobj ~ media, data = data)
summary(fit)
```

```
## Call:
## coxph(formula = Survobj ~ media, data = data)
##
## n= 124, number of events= 85
##
##           coef exp(coef) se(coef)      z Pr(>|z|)
## mediaFG20 -0.6010     0.5482  0.2622 -2.293   0.0219 *
## mediaOP50 -0.1943     0.8234  0.2685 -0.724   0.4693
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## mediaFG20    0.5482      1.824    0.3280    0.9165
## mediaOP50    0.8234      1.214    0.4865    1.3936
##
## Concordance= 0.576 (se = 0.037 )
## Rsquare= 0.044 (max possible= 0.997 )
## Likelihood ratio test= 5.57 on 2 df,  p=0.06
## Wald test               = 5.44 on 2 df,  p=0.07
## Score (logrank) test = 5.57 on 2 df,  p=0.06
```

We then try to split up in replicates to see if there are any differences between scorers

```
d1 <- data[data$Replicate == 1,]
d2 <- data[data$Replicate == 2,]
d1$Survobj <- with(d1, Surv(d1$Time, event = d1$status))
d2$Survobj <- with(d2, Surv(d2$Time, event = d2$status))
```

Then we make kaplan-meier curves and dose response curves

```
km1 <- survfit(Survobj ~ media, data = d1, conf.type = "log-log")
km2 <- survfit(Survobj ~ media, data = d2, conf.type = "log-log")
df_fly_13 <- data.frame(c(0, km1$time[1:7]), c(1, km1$surv[1:7]),
                        c(0, km1$std.err[1:7]))
colnames(df_fly_13) <- c('Time', 'Surv', 'Std.error')
df_fly_20 <- data.frame(c(0, km1$time[8:15]), c(1, km1$surv[8:15]),
                        c(0, km1$std.err[8:15]))
colnames(df_fly_20) <- c('Time', 'Surv', 'Std.error')
df_OP50 <- data.frame(c(0, km1$time[16:24]), c(1, km1$surv[16:24]),
                      c(0, km1$std.err[16:24]))
colnames(df_OP50) <- c('Time', 'Surv', 'Std.error')
```

```

plot(df_fly_13$Time[2:9],df_fly_13$Surv[2:9], pch = 16, cex = 1.2, xlab = 'Heatstress (hours)',
     ylab = 'Surviving fraction',
     main = expression('Survival heat stress for'~italic(C.elegans)),
     xlim = c(0,22), ylim = c(0,1), xaxp = c(0,22,11))
lines(df_fly_13$Time,df_fly_13$Surv)
arrows(df_fly_13$Time, df_fly_13$Surv-df_fly_13$Std.error, df_fly_13$Time,
       df_fly_13$Surv+df_fly_13$Std.error, length=0.05, angle=90, code=3, col = 'black')

```

```

## Warning in arrows(df_fly_13$Time, df_fly_13$Surv - df_fly_13$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped

```

```

points(df_OP50$Time[2:9],df_OP50$Surv[2:9], pch = 1, cex = 1.2)
lines(df_OP50$Time,df_OP50$Surv, lty = 2)
arrows(df_OP50$Time, df_OP50$Surv-df_OP50$Std.error,
       df_OP50$Time, df_OP50$Surv+df_OP50$Std.error,
       length=0.05, angle=90, code=3, col = 'black')

```

```

## Warning in arrows(df_OP50$Time, df_OP50$Surv - df_OP50$Std.error,
## df_OP50$Time, : zero-length arrow is of indeterminate angle and so skipped

```

```

points(df_fly_20$Time[2:9],df_fly_20$Surv[2:9], pch = 6)
lines(df_fly_20$Time,df_fly_20$Surv,lty = 3)
arrows(df_fly_20$Time, df_fly_20$Surv-df_fly_20$Std.error, df_fly_20$Time,
       df_fly_20$Surv+df_fly_20$Std.error, length=0.05, angle=90, code=3, col = 'black')

```

```

## Warning in arrows(df_fly_20$Time, df_fly_20$Surv - df_fly_20$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped

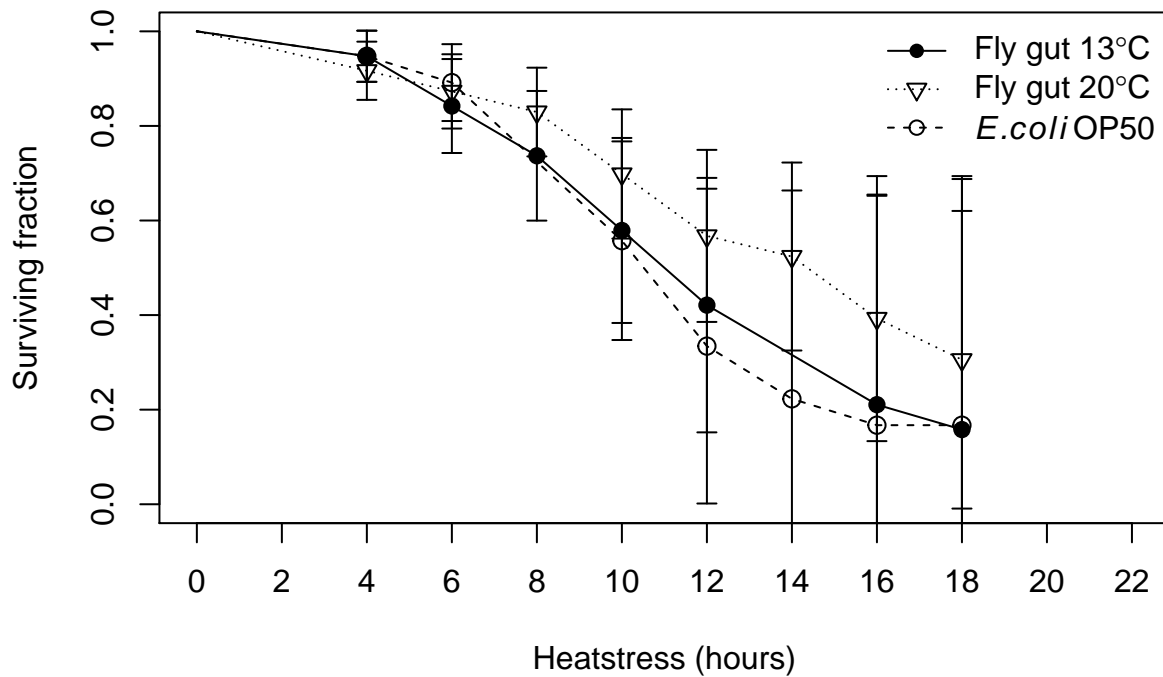
```

```

legend('topright', pch = c(16,6,1),lty = c(1,3,2),
     legend = c(expression('Fly gut 13'*degree*C),
                 expression('Fly gut 20'*degree*C),
                 expression(italic(E.coli) ~ OP50)), bty = 'n')

```

Survival heat stress for *C.elegans*



```
df_fly_13 <- data.frame(c(0,km2$time[1:7]),c(1,km2$urv[1:7]),c(0,km2$std.err[1:7]))
colnames(df_fly_13) <- c('Time','Surv','Std.error')
df_fly_20 <- data.frame(c(0,km2$time[8:12]),c(1,km2$urv[8:12]),
                        c(0,km2$std.err[8:12]))
colnames(df_fly_20) <- c('Time','Surv','Std.error')
df_OP50 <- data.frame(c(0,km2$time[13:20]),c(1,km2$urv[13:20]),
                      c(0,km2$std.err[13:20]))
colnames(df_OP50) <- c('Time','Surv','Std.error')

plot(df_fly_13$Time[2:9],df_fly_13$Surv[2:9], pch = 16, cex = 1.2, xlab = 'Heatstress (hours)',
      ylab = 'Surviving fraction',
      main = expression('Survival heat stress for'-italic(C.elegans)),
      xlim = c(0,22), ylim = c(0,1), xaxp = c(0,22,11))
lines(df_fly_13$Time,df_fly_13$Surv)
arrows(df_fly_13$Time, df_fly_13$Surv-df_fly_13$Std.error, df_fly_13$Time,
       df_fly_13$Surv+df_fly_13$Std.error, length=0.05, angle=90, code=3, col = 'black')
```

```
## Warning in arrows(df_fly_13$Time, df_fly_13$Surv - df_fly_13$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped
```

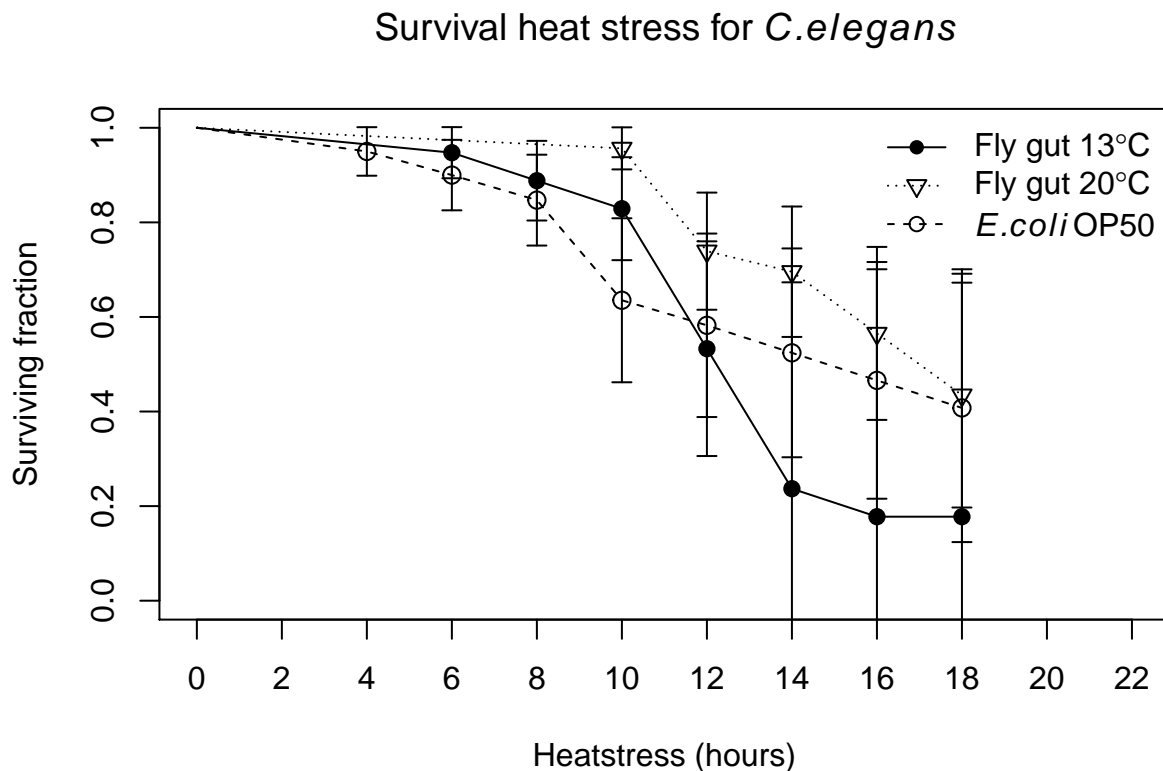
```
points(df_OP50$Time[2:9],df_OP50$Surv[2:9], pch = 1, cex = 1.2)
lines(df_OP50$Time,df_OP50$Surv, lty = 2)
arrows(df_OP50$Time, df_OP50$Surv-df_OP50$Std.error,
       df_OP50$Time, df_OP50$Surv+df_OP50$Std.error,
       length=0.05, angle=90, code=3, col = 'black')
```

```
## Warning in arrows(df_OP50$Time, df_OP50$Surv - df_OP50$Std.error,
## df_OP50$Time, : zero-length arrow is of indeterminate angle and so skipped
```

```
points(df_fly_20$Time[2:9],df_fly_20$Surv[2:9], pch = 6)
lines(df_fly_20$Time,df_fly_20$Surv,lty = 3)
arrows(df_fly_20$Time, df_fly_20$Surv-df_fly_20$Std.error, df_fly_20$Time,
      df_fly_20$Surv+df_fly_20$Std.error, length=0.05, angle=90, code=3, col = 'black')
```

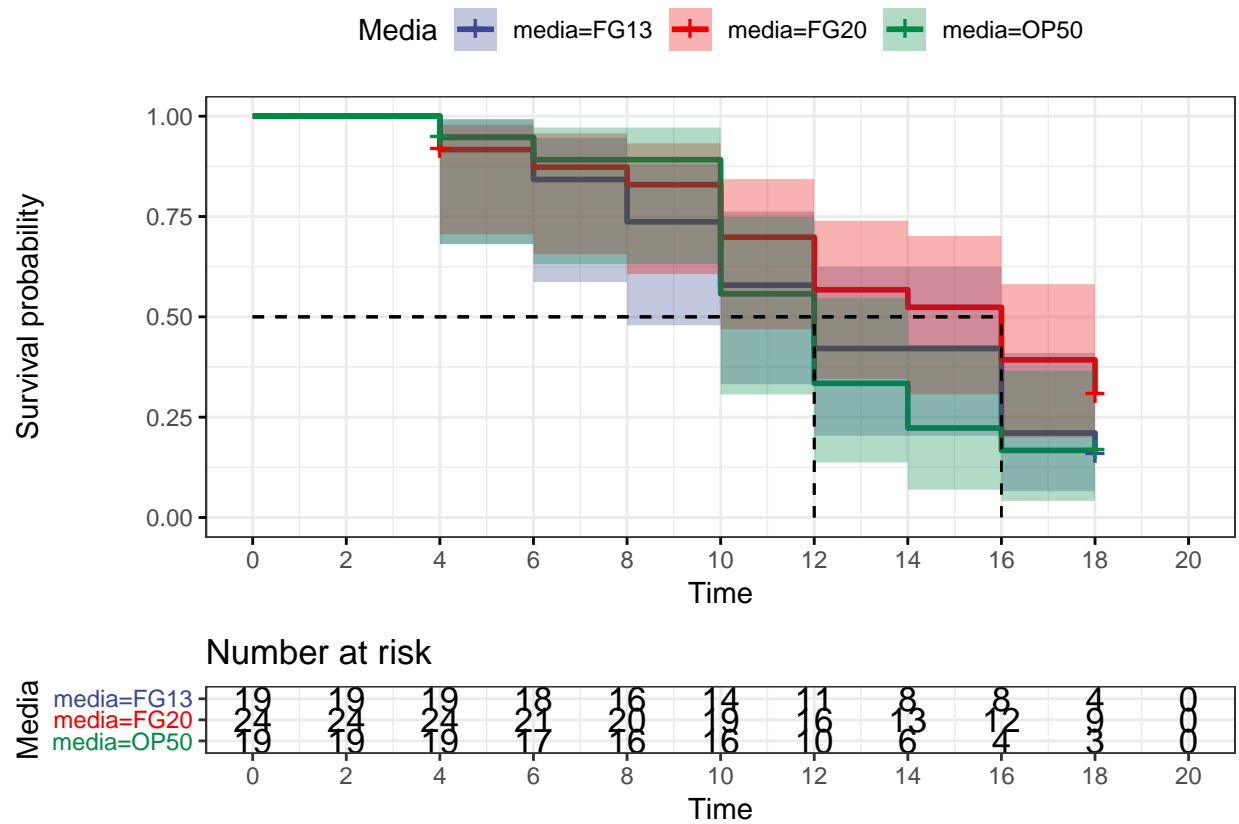
```
## Warning in arrows(df_fly_20$Time, df_fly_20$Surv - df_fly_20$Std.error, :
## zero-length arrow is of indeterminate angle and so skipped
```

```
legend('topright', pch = c(16,6,1),lty = c(1,3,2),
      legend = c(expression('Fly gut 13'*degree*C),
                    expression('Fly gut 20'*degree*C),
                    expression(italic(E.coli) ~ OP50)), bty = 'n')
```

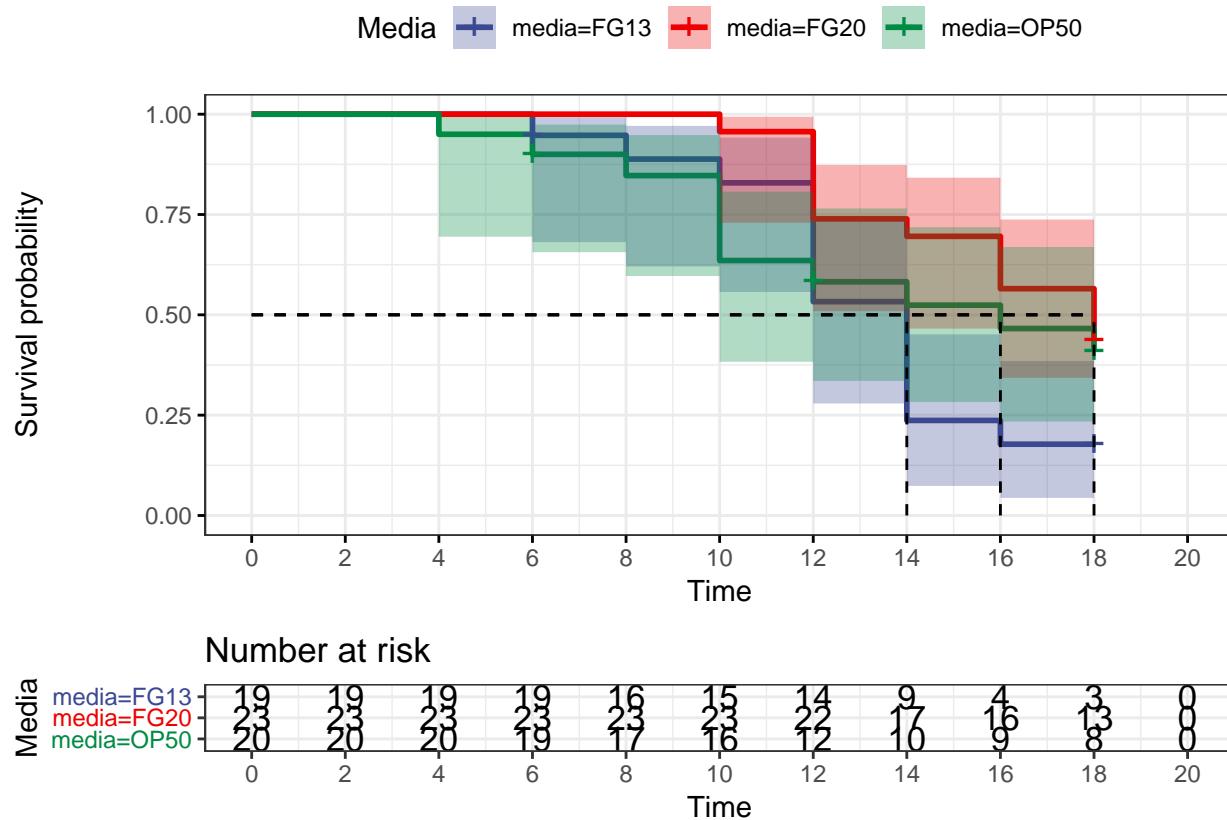


Then kaplan meier

```
ggsurvplot(km1,data = data, conf.int = TRUE, ggtheme = theme_bw(),
            risk.table = 0.25, palette = 'aaas', surv.median.line = 'hv',
            legend.title = 'Media',break.x.by = 2)
```

```
ggsurvplot(km2,data = data, conf.int = TRUE, ggtheme = theme_bw(),
  risk.table = 0.25, palette = 'aaas', surv.median.line = 'hv',
  legend.title = 'Media',break.x.by = 2)
```



and at last some tests

```
survdifff(Survobj ~ media, data = d1, rho = 0)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = d1, rho = 0)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 19      16    14.1    0.244    0.431
## media=FG20 24      16    20.3    0.915    2.014
## media=OP50 19      15    12.5    0.480    0.815
##
## Chisq= 2.1  on 2 degrees of freedom, p= 0.4
```

```
survdifff(Survobj ~ media, data = d1, rho = 1)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = d1, rho = 1)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 19    10.8    9.54    0.162    0.368
## media=FG20 24    10.3   13.14    0.613    1.687
## media=OP50 19    10.6    9.03    0.282    0.613
##
## Chisq= 1.7  on 2 degrees of freedom, p= 0.4
```

```
fit <- coxph(Survobj ~media, data = d1)
summary(fit)
```

```
## Call:
## coxph(formula = Survobj ~ media, data = d1)
##
## n= 62, number of events= 47
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## mediaFG20 -0.41066  0.66321  0.35443 -1.159  0.247
## mediaOP50  0.07651  1.07951  0.36076  0.212  0.832
##
##               exp(coef) exp(-coef) lower .95 upper .95
## mediaFG20    0.6632    1.5078    0.3311    1.328
## mediaOP50    1.0795    0.9263    0.5323    2.189
##
## Concordance= 0.557 (se = 0.052 )
## Rsquare= 0.035 (max possible= 0.995 )
## Likelihood ratio test= 2.2 on 2 df,  p=0.3
## Wald test               = 2.13 on 2 df,  p=0.3
## Score (logrank) test = 2.16 on 2 df,  p=0.3
```

```
survdifff(Survobj ~media, data = d2, rho = 0)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = d2, rho = 0)
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 19      14      9.32  2.355719  3.72463
## media=FG20 23      13     17.77  1.282531  2.84312
## media=OP50 20      11     10.91  0.000743  0.00122
##
## Chisq= 4.3 on 2 degrees of freedom, p= 0.1
```

```
survdifff(Survobj ~media, data = d2, rho = 1)
```

```
## Call:
## survdifff(formula = Survobj ~ media, data = d2, rho = 1)
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## media=FG13 19    10.58    7.27    1.510    2.95
## media=FG20 23    8.40    12.74    1.475    4.04
## media=OP50 20    8.98    7.96    0.131    0.27
##
## Chisq= 4.6 on 2 degrees of freedom, p= 0.1
```

```
fit <- coxph(Survobj ~media, data = d2)
summary(fit)
```

```
## Call:
## coxph(formula = Survobj ~ media, data = d2)
```

```

##
##   n= 62, number of events= 38
##
##           coef exp(coef) se(coef)      z Pr(>|z|)
## mediaFG20 -0.8083    0.4456   0.3905 -2.070   0.0385 *
## mediaOP50 -0.4944    0.6099   0.4073 -1.214   0.2248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## mediaFG20    0.4456      2.244    0.2073    0.958
## mediaOP50    0.6099      1.639    0.2746    1.355
##
## Concordance= 0.603 (se = 0.053 )
## Rsquare= 0.066 (max possible= 0.989 )
## Likelihood ratio test= 4.23 on 2 df,  p=0.1
## Wald test              = 4.36 on 2 df,  p=0.1
## Score (logrank) test = 4.53 on 2 df,  p=0.1

```