Biostatistics (MATH11230)

Vanda Inácio

2022-11-07

In this supplementary file I start by showing show how to fit a Weibull distribution to a sample that contains censored event times. The `survreg` function from the `survival` package fits a Weibull model with two parameters: intercept, say β_0 and a scale parameter σ. The two parameters we use λ and α are related to β_0 and σ by

$$\lambda = e^{-\beta_0/\sigma}, \qquad \alpha = \frac{1}{\sigma}.$$

Remember that the survival function of the Weibull distribution (as we have it in slide 11 of 'Parametric estimators of the survival function') is given by

$$S(t; \lambda, \alpha) = e^{-\lambda t^{\alpha}}, \lambda > 0, \quad \alpha > 0.$$

We will analyse the classic `veteran` dataset available in the package survival. For more information about the dataset, please type `help(veteran)`.

```
require(survival)
fit_weibull <- survreg(Surv(time, status) ~ 1, dist = "weibull", data = veteran)
summary(fit_weibull)
```

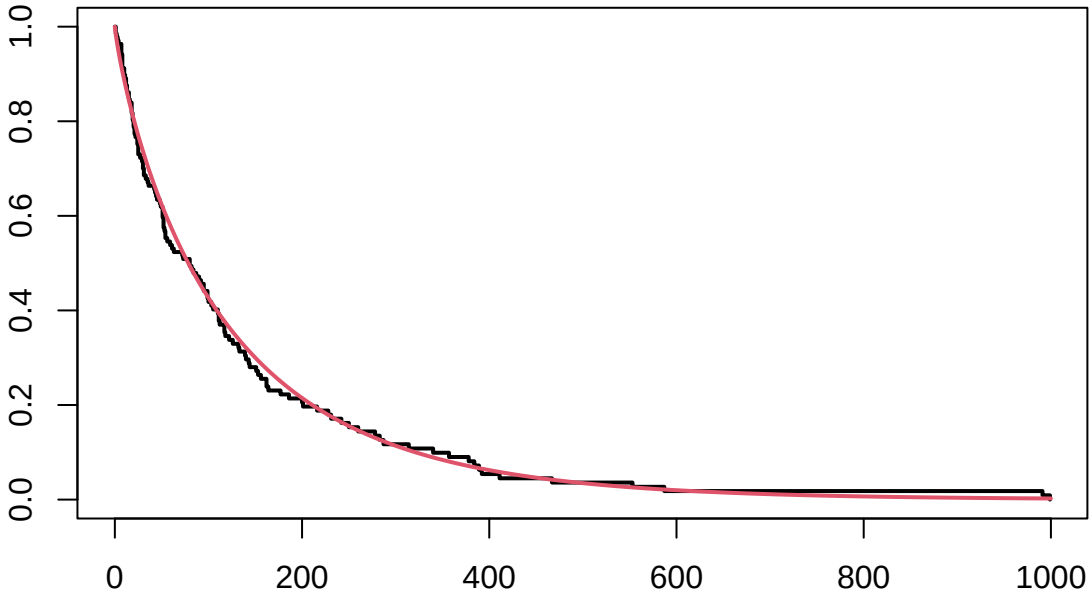
```
##
## Call:
## survreg(formula = Surv(time, status) ~ 1, data = veteran, dist = "weibull")
##              Value Std. Error     z      p
## (Intercept) 4.7931     0.1078 44.46 <2e-16
## Log(scale)  0.1601     0.0669  2.39  0.017
##
## Scale= 1.17
##
## Weibull distribution
## Loglik(model)= -748.1   Loglik(intercept only)= -748.1
## Number of Newton-Raphson Iterations: 6
## n= 137
```

```
alpha <- 1/ fit_weibull$scale
lambda <- exp(- fit_weibull$coefficients/fit_weibull$scale)

y <- seq(0, 1000, by = 1)
est_surv <- exp(-lambda * y^alpha)

#Kaplan-Meier estimate
km_fit <- survfit(Surv(time, status) ~ 1, conf.type = "log-log", data = veteran)

plot(km_fit, conf.int = FALSE, lwd = 2)
lines(y, est_surv, col = 2, lwd = 2)
```

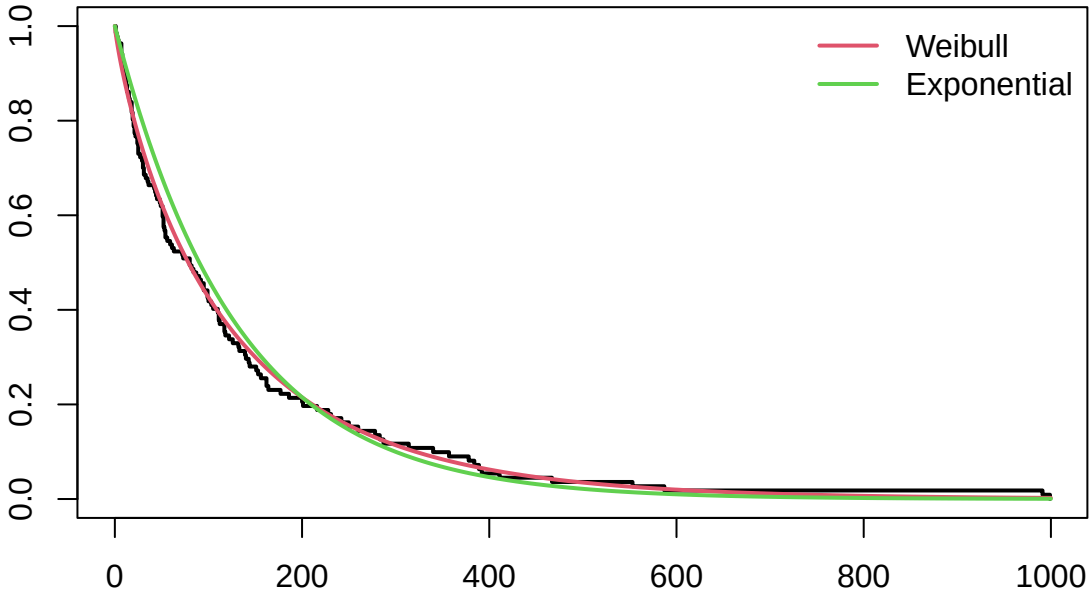


Let us compare with the fit of an exponential distribution.

```
theta_est <- sum(veteran$status)/sum(veteran$time)
theta_est
```

```
## [1] 0.00768169
```

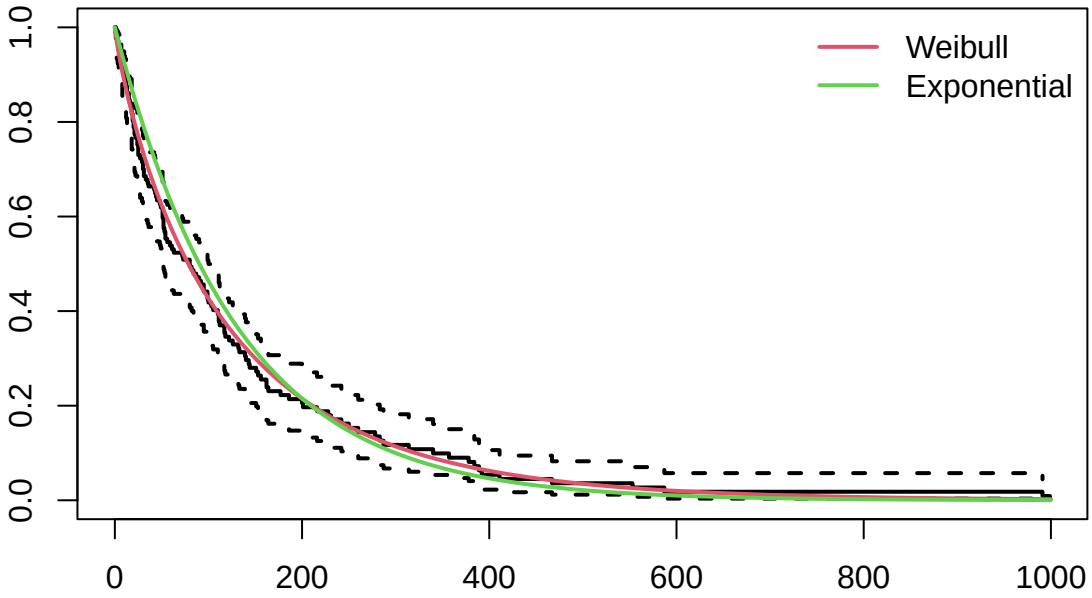
```
est_surv_exp <- exp(-theta_est*y)
plot(km_fit, conf.int = FALSE, lwd = 2)
lines(y, est_surv, col = 2, lwd = 2)
lines(y, est_surv_exp, col = 3, lwd = 2)
legend("topright", col = c(2, 3),
       lty = c(1,1), lwd = c(2, 2),
       c("Weibull", "Exponential"), bty = "n")
```



Although the two fits are similar, when we include the pointwise confidence bands of the Kaplan-Meier estimate, we see that for times before 100 or so, the exponential estimate of the survival distribution is not

included in the bands. Nevertheless, in this case, both parametric models seem to follow quite nicely the Kaplan-Meier estimate.

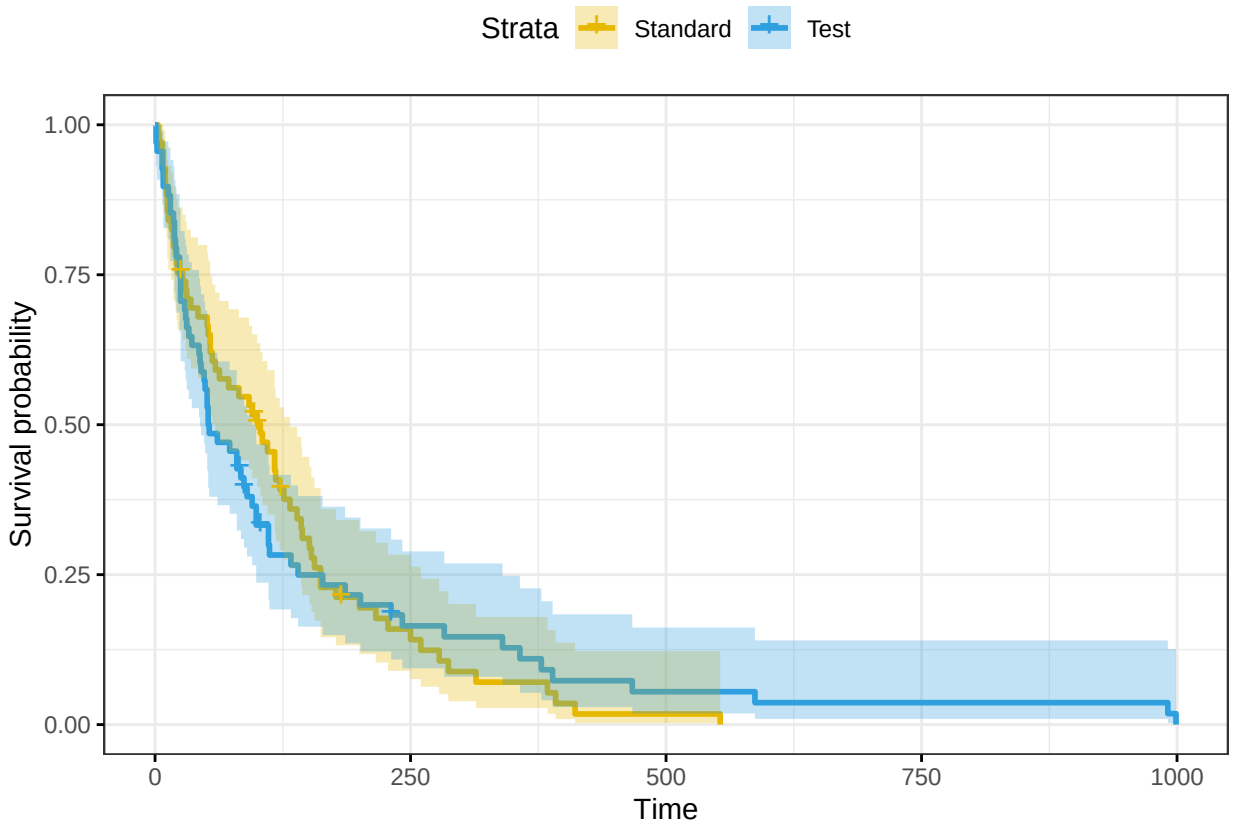
```
plot(km_fit, conf.int = TRUE, lwd = 2)
lines(y, est_surv, col = 2, lwd = 2)
lines(y, est_surv_exp, col = 3, lwd = 2)
legend("topright", col = c(2, 3),
       lty = c(1,1), lwd = c(2, 2),
       c("Weibull", "Exponential"), bty = "n")
```



We now move to the comparison of survival curves part.

```
require(survival)
fit_veteran <- survfit(Surv(time, status) ~ trt, data = veteran)

require(survminer)
p <- ggsurvplot(fit_veteran,
data = veteran,
size = 1,
palette = c("#E7B800", "#2E9FDF"),
conf.int = TRUE,
legend.labs =  c("Standard", "Test"),
ggtheme = theme_bw()
)
print(p)
```



We now reproduce the results of the toy example and also illustrate the use of the `survdiff` function.

```
time <- c(3.1, 6.8, 9, 9, 11.3, 16.2, 8.7, 9, 10.1, 12.1, 18.7, 23.1)
status <- c(1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0)
group <- c(rep(1, 6), rep(2, 6))
res_toy <- survdiff(Surv(time, status) ~ group)
res_toy
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ group)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## group=1 6        4     2.57     0.800      1.62
## group=2 6        3     4.43     0.463      1.62
##
##  Chisq= 1.6  on 1 degrees of freedom, p= 0.2
```

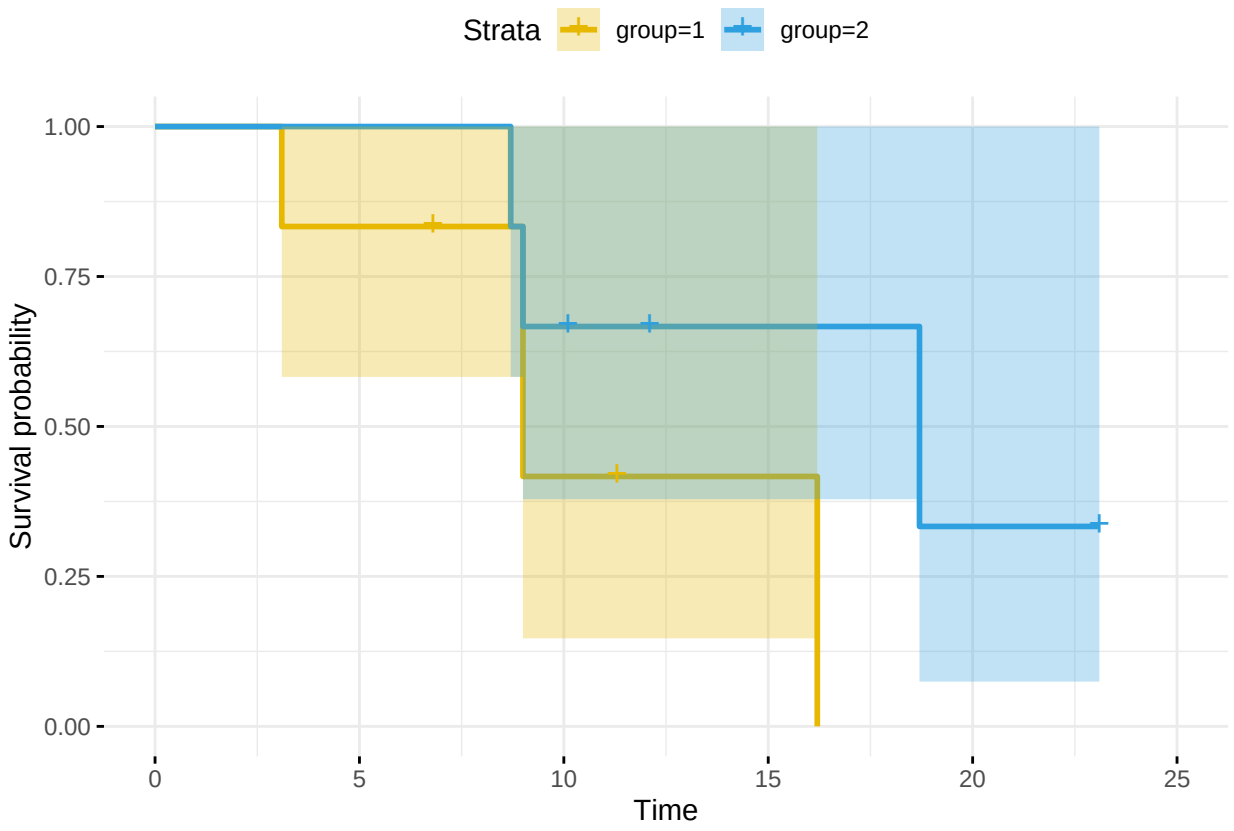
```
#observed number events across all timepoints
o <- 4
#expected number events across all timepoints
e <- (6/12) + (4/10) + (12/9) + (1/3)
#variance  of the U_L statistic
v <- (6*6*11)/((12^2)*11) + (4*6*9)/((10^2)*9) +
     (4*5*3*6)/((9^2)*8) + (1*2*1*2)/((3^2)*2) + 0
((o-e)^2)/v
```

```
## [1] 1.620508
```

```
#p-value
pchisq(((o-e)^2)/v, df = 1, lower.tail = FALSE)
```

```
## [1] 0.2030209
```

```
require(survminer)
toy_data <- data.frame("time" = time, "status" = status, "group" = group)
fit_toy <- survfit(Surv(time, status) ~ group, data = toy_data)
ggsurvplot(fit_toy,
           conf.int = 0.95,
           palette = c("#E7B800", "#2E9FDF"),
           ggtheme = theme_minimal(),
           data = toy_data
           )
```



Let us now compare the log-rank test and its corresponding weighted versions using the veteran dataset.

```
res_veteran <- survdiff(Surv(time, status) ~ trt, data = veteran)
res_veteran
```

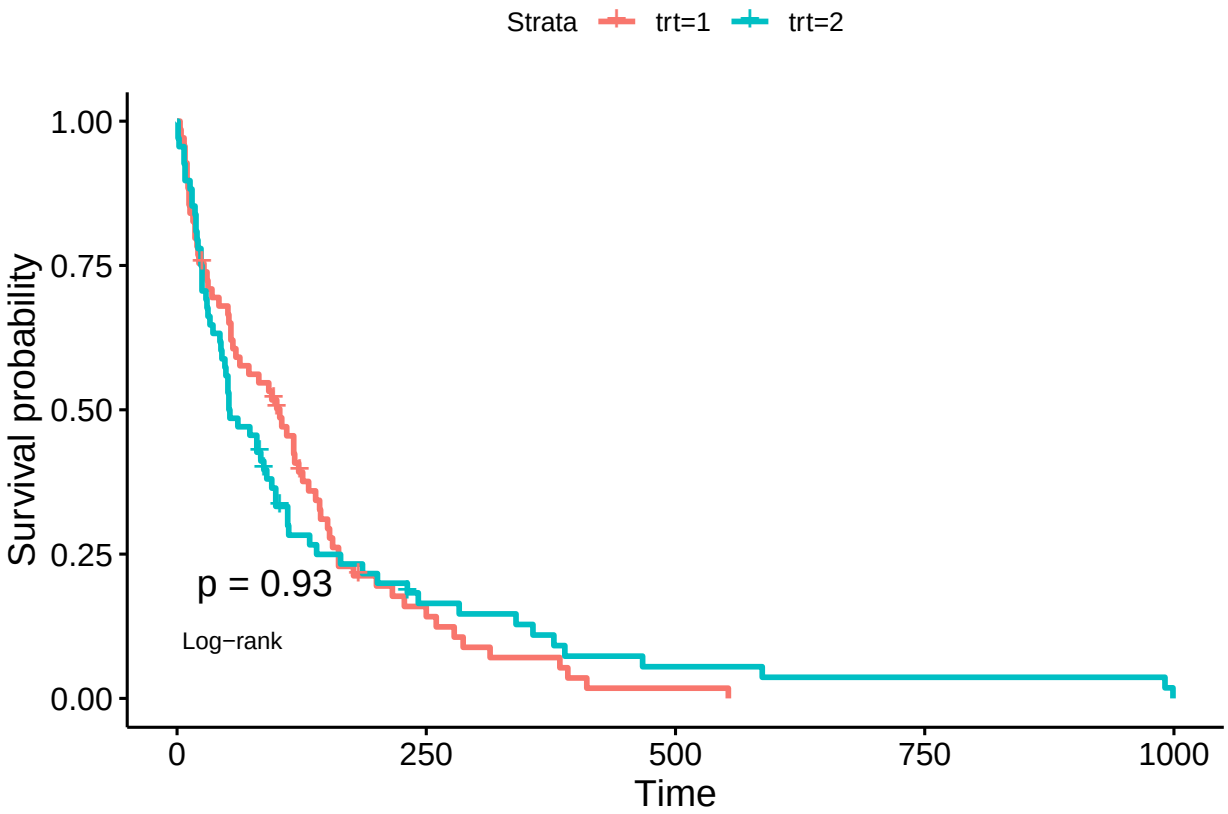
```
## Call:
## survdiff(formula = Surv(time, status) ~ trt, data = veteran)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## trt=1 69       64     64.5   0.00388   0.00823
## trt=2 68       64     63.5   0.00394   0.00823
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

```
res_veteran_peto <- survdiff(Surv(time, status) ~ trt, data = veteran, rho = 1)
res_veteran_peto
```

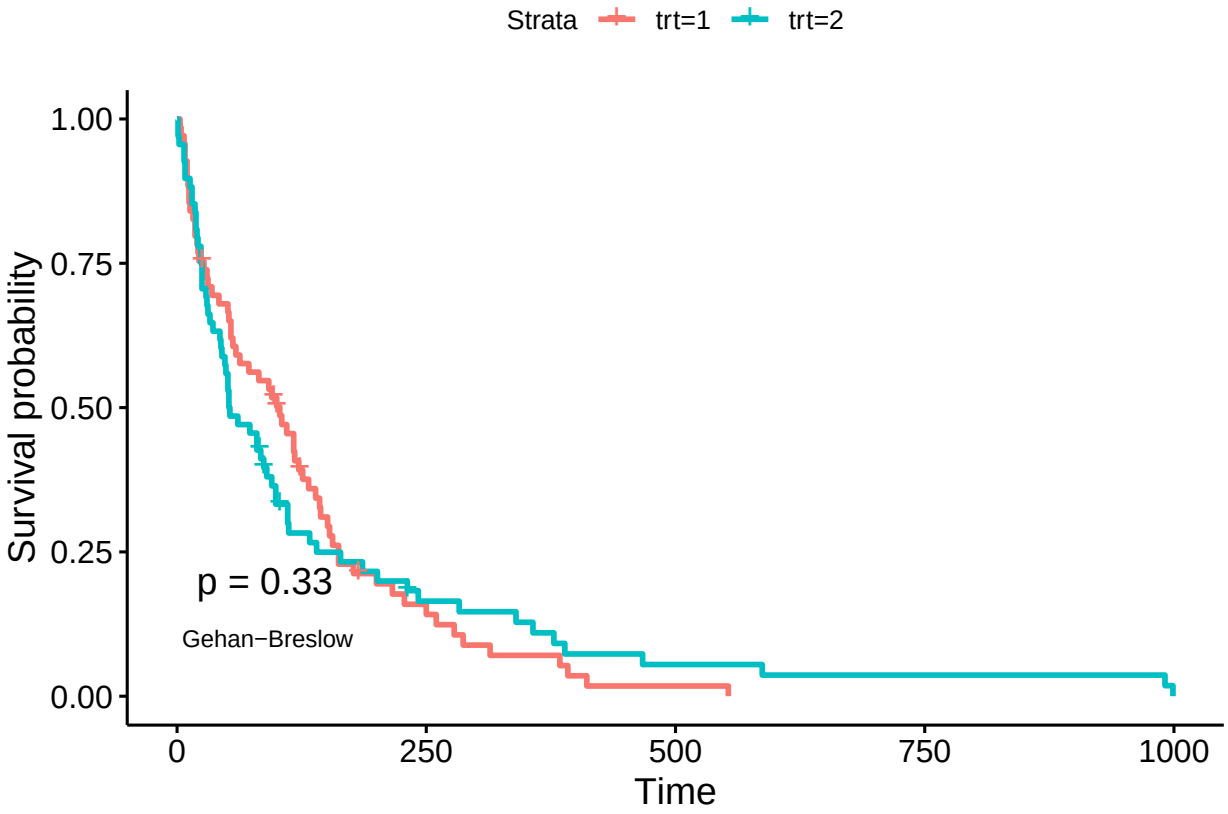
```
## Call:
```

```
## survdiff(formula = Surv(time, status) ~ trt, data = veteran,
##     rho = 1)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## trt=1 69     32.2     35.4     0.279     0.871
## trt=2 68     35.2     32.1     0.308     0.871
##
##  Chisq= 0.9  on 1 degrees of freedom, p= 0.4
```

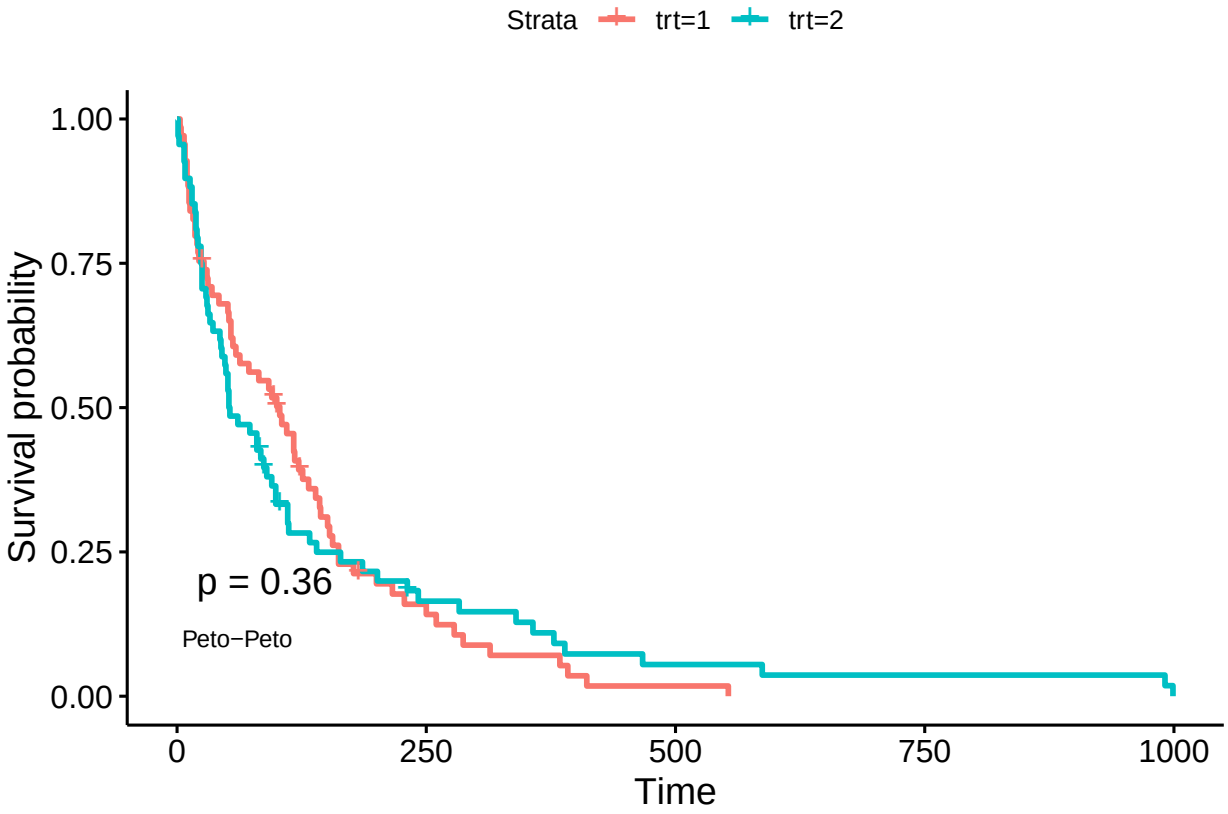
```
#p-value in the plot corresponds to log rank test
ggsurvplot(fit_veteran,
           data = veteran,
           pval = TRUE,
           pval.method = TRUE,
           log.rank.weights = "1",
           pval.method.coord = c(5, 0.1),
           pval.method.size = 3)
```




```
#p-value in the plot corresponds to Gehan-Breslow test
ggsurvplot(fit_veteran,
           data = veteran,
           pval = TRUE,
           pval.method = TRUE,
           log.rank.weights = "n",
           pval.method.coord = c(5, 0.1),
           pval.method.size = 3)
```




```
#p-value in the plot corresponds to Peto test
ggsurvplot(fit_veteran,
           data = veteran,
           pval = TRUE,
           pval.method = TRUE,
           log.rank.weights = "S1",
           pval.method.coord = c(5, 0.1),
           pval.method.size = 3)
```



To illustrate the extension of the log rank test for the comparison of multiple survival curves let us use the veteran dataset again but now comparing the survival experience for the different cell types (1=squamous, 2=smallcell, 3=adeno, 4=large).

```
res_veteran_multiple <- survdiff(Surv(time, status) ~ celltype, data = veteran)
res_veteran_multiple
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ celltype, data = veteran)
##
##                      N Observed Expected (O-E)^2/E (O-E)^2/V
## celltype=squamous   35       31     47.7      5.82     10.53
## celltype=smallcell  48       45     30.1      7.37     10.20
## celltype=adeno      27       26     15.7      6.77      8.19
## celltype=large      27       26     34.5      2.12      3.02
##
##  Chisq= 25.4  on 3 degrees of freedom, p= 1e-05
```

```
#Reconstructing the test statistic manually
U <- res_veteran_multiple$obs[1:3] - res_veteran_multiple$exp[1:3]
V <- res_veteran_multiple$var[1:3, 1:3]
t(U)%*%solve(V)%*%U
```

```
##          [,1]
## [1,] 25.4037
```