

Calculation of hazard ratios of parametric survival models in R
- A tutorial [Draft]

Wilmar Igl, PhD

Sunday 7th January, 2018/ 11:07

Contents

| | | |
|----------|---|-----------|
| 1 | Overview | 3 |
| 2 | Cautionary Notes | 3 |
| 2.1 | Differences between software | 3 |
| 2.2 | Disclaimer | 3 |
| 3 | Data example | 4 |
| 4 | Calculation of an (average) hazard ratio based on the Cox PH model | 4 |
| 5 | Calculation of the hazard ratio of proportional hazards survival models | 5 |
| 5.1 | Exponential Distribution | 5 |
| 5.1.1 | Application | 5 |
| 5.1.2 | Theory | 5 |
| 5.2 | Weibull distribution | 6 |
| 5.2.1 | Application | 6 |
| 5.2.2 | Theory | 7 |
| 5.3 | Gompertz Distribution | 9 |
| 5.3.1 | Application | 9 |
| 5.3.2 | Theory | 9 |
| 6 | Calculation of an (average) hazard ratio for accelerated failure time models | 10 |
| 6.1 | Weighted average hazard ratios | 10 |
| 6.1.1 | sAHR | 10 |
| 6.1.2 | gAHR | 11 |
| 6.1.3 | AHR | 11 |
| 6.1.4 | Alternative overall effect sizes | 11 |
| 6.2 | Log-logistic Distribution | 12 |
| 6.2.1 | Application | 12 |
| 6.2.2 | Theory | 17 |
| 6.3 | Log-normal Distribution | 18 |
| 6.3.1 | Application | 18 |
| 6.3.2 | Theory | 23 |
| 6.4 | Generalized Gamma Distribution | 24 |
| 6.4.1 | Application | 24 |
| 6.4.2 | Theory | 29 |
| | References | 30 |
| 7 | Software information | 31 |

1 Overview

This tutorial aims to support the interpretation of parametric time-to-event models by explaining how to calculate the hazard ratio, which is a conventional effect size to evaluate clinical relevance of treatment effects.

The following distributions, which are considered most important for health economic analyses, are implemented in the `flexsurv` package (cf `?flexsurvreg`):

1. `exp`: exponential (PH model)
2. `weibull`: Weibull (AFT model/PH model)
3. `gompertz`: Gompertz (PH model)
4. `llogis`: log-logistic (AFT model)
5. `lnorm`: lognormal (AFT model)
6. `gengamma`: Generalized gamma (AFT model)

A single, constant hazard ratio can only be calculated for distributions, for which the hazard ratio does not depend on time and assumes proportional hazards. For the exponential, Weibull (if PH parametrisation is used), and Gompertz distribution, a single, constant hazard ratio can be computed, but not for the remaining accelerated failure time (AFT) models.

2 Cautionary Notes

2.1 Differences between software

Distributions, which are used for parametric time-to-event models may use various parametrizations, which are different between R and other statistical analysis systems, eg SAS, but also within R packages, which may lead to misinterpretation of results.

Special attention is recommended when using the Weibull distribution, which is different between:

- R vs SAS
- R
 - `stats::dweibull()`
 - `survival::survreg(..., dist="weibull")`
 - `flexsurv::flexsurvreg(..., dist="weibull")`
 - `eha::weibreg()`

The differences concern the implementation as PH vs. AFT model, but also the definition of shape and scale. Therefore, careful study of the documentation of functions is recommended (see also Zhang [2016]).

2.2 Disclaimer

This program¹ is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation;

This program is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details. You can retrieve a copy of the GNU public licence version 3 online².

¹Program is used here in the sense of literate programming, which combines natural language and computer language in an integrated program or document, respectively.

²<https://www.gnu.org/licenses/gpl-3.0.en.html>

3 Data example

The *Breast Cancer Survival Data* (German Breast Cancer Study Group, 1984-1989) from the *flexsurv* package with survival times of 686 patients with primary node-positive breast cancer is used as a data example:

```
data(bc)
#help(bc)
bc$recyrs <- bc$rectime/365.25 # years
bc$group <- factor(bc$group, levels=c("Poor", "Medium", "Good"))
```

4 Calculation of an (average) hazard ratio based on the Cox PH model

The hazard ratio based on the standard Cox PH model.

```
library(survival)
fit.coxph <- coxph(formula = Surv(recyrs, censrec) ~ group, data=bc)
print(fit.coxph)

## Call:
## coxph(formula = Surv(recyrs, censrec) ~ group, data = bc)
##
##              coef exp(coef) se(coef)      z      p
## groupMedium -0.778      0.459   0.130 -5.98 2.3e-09
## groupGood   -1.618      0.198   0.165 -9.83 < 2e-16
##
## Likelihood ratio test=114 on 2 df, p=0
## n= 686, number of events= 299

hr.coxph <- exp(coef(fit.coxph))["groupGood"]
cat("The Hazard Ratio (Good:Poor) is ", round(hr.coxph, 4), ".")

## The Hazard Ratio (Good:Poor) is  0.1983 .
```

The (averaged) hazard ratio based on the weighted Cox PH model (using Prentice & Kalbfleisch weights with censoring correction and robust variance estimation) [Schemper et al., 2009].

```
library(coxphw)
fit.coxphw <- coxphw(formula = Surv(recyrs, censrec) ~ group, template="AHR", data=bc)
print(fit.coxphw)

## coxphw(formula = Surv(recyrs, censrec) ~ group, data = bc, template = "AHR")
##
## Model fitted by weighted estimation (AHR template)
##
##              coef se(coef) exp(coef) lower 0.95 upper 0.95      z
## groupMedium -0.8002188 0.1421561 0.4492306  0.3399898  0.5935713 -5.629154
## groupGood   -1.7732565 0.1822368 0.1697792  0.1187857  0.2426637 -9.730506
##
##              p
## groupMedium 1.810959e-08
## groupGood   0.000000e+00
##
## Wald Chi-square=102.1684 on 2df, p=0, n=686

oc.coxphw <- concord(fit.coxphw)
hr.coxphw <- oc.coxphw["groupGood", "concordance prob."]
cat("The Hazard Ratio (Good:Poor) is ", round(hr.coxphw, 4), ".")

## The Hazard Ratio (Good:Poor) is  0.1451 .
```

5 Calculation of the hazard ratio of proportional hazards survival models

5.1 Exponential Distribution

5.1.1 Application

The HR, which compares treatment i with a reference treatment, can be computed from the estimates for treatment i from the table of coefficients as follows:

$$HR_i = \exp(\beta_i) \quad (1)$$

```
library(flexsurv)
fit.exp <- flexsurvreg(formula = Surv(reccyrs, censrec) ~ group, dist="exp", data=bc)
print(fit.exp)

## Call:
## flexsurvreg(formula = Surv(reccyrs, censrec) ~ group, data = bc,
##   dist = "exp")
##
## Estimates:
##           data mean  est      L95%      U95%      se      exp(est)
## rate           NA    0.2812  0.2389  0.3309  0.0233      NA
## groupMedium  0.3338  -0.7195 -0.9721 -0.4669  0.1289  0.4870
## groupGood    0.3338  -1.5375 -1.8566 -1.2184  0.1628  0.2149
##           L95%      U95%
## rate           NA      NA
## groupMedium  0.3783  0.6269
## groupGood    0.1562  0.2957
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -830.9014, df = 3
## AIC = 1667.803

hr.exp <- exp(coef(fit.exp)["groupGood"])
cat("The Hazard Ratio (Good:Poor) is ", round(hr.exp, 4), ".")

## The Hazard Ratio (Good:Poor) is 0.2149 .
```

Therefore, the hazard ratio of patients in the good prognostic group to die is 0.2149 compared to patients in the poor prognostic group, ie about an 79% reduction of the hazard.

5.1.2 Theory

For transparency the derivation is given below:

First, the documentation for the time-to-event regression model is retrieved using `?flexsurv::flexsurvreg`. Second, the name (stem) of the exponential distribution function is used, and the respective density distribution function retrieved using `?dexp`. The documentation for the exponential density function does not give the parametrization for the hazard function $h(t)$, but can be taken from the literature as constant function $h(t) = \lambda$. Considering that the hazard ratio is calculated as (0=control, 1=experimental):

$$HR = \frac{h_1(t)}{h_0(t)} \quad (2)$$

(3)

The hazard function is (rate parameter λ):

$$h(t) = \lambda \quad (4)$$

The hazard function in the control group (=0) is:

$$h_0(t) = \lambda_0 \quad (5)$$

The hazard function in the experimental group (=1) is:

$$h_1(t) = \lambda_1 \quad (6)$$

The hazard ratio is:

$$\text{HR} = \frac{\lambda_1}{\lambda_0} \quad (7)$$

As described in `?flexsurvreg` the location parameter of the exponential distribution, which is implemented in `dgompertz` is `rate` and the covariates represent treatment effects on the log of the rate parameter.

Therefore, the rate values can be retrieved from the table of coefficients by ($\beta_0 = \text{intercept}$):

$$\lambda_0 = \exp(\beta_0) \quad (8)$$

$$\lambda_1 = \exp(\beta_0 + \beta_1) \quad (9)$$

Therefore, the hazard ratio (experimental: control) is calculated as:

$$\text{HR} = \frac{\lambda_1}{\lambda_0} \quad (10)$$

$$= \frac{\exp(\beta_0 + \beta_1)}{\exp(\beta_0)} \quad (11)$$

$$= \exp(\beta_1) \quad (12)$$

5.2 Weibull distribution

5.2.1 Application

The HR, which compares treatment i with a reference treatment, can be computed from the estimates for treatment i from the table of coefficients as follows:

$$\text{HR}_i = \exp(-\beta_i)^a \quad (13)$$

```
# AFT model
fit.wbl.aft <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="weibull", data=bc)
print(fit.wbl.aft)

## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##   dist = "weibull")
##
## Estimates:
##      data mean  est    L95%    U95%    se    exp(est)  L95%
## shape      NA  1.3797  1.2548  1.5170  0.0668     NA      NA
## scale      NA  3.3964  3.0180  3.8223  0.2047     NA      NA
## groupMedium 0.3338  0.5986  0.4127  0.7845  0.0948  1.8196  1.5110
## groupGood  0.3338  1.2122  0.9661  1.4583  0.1256  3.3609  2.6277
##
##      U95%
## shape      NA
## scale      NA
## groupMedium 2.1913
## groupGood  4.2988
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -811.7372, df = 4
## AIC = 1631.474
```

```

# Note: coef (below) gives parameters on log scale, and therefore
# need to be exponentiated (also shape!). However, in the output of the
# model fit, scale and shape are already exponentiated, but not the
# treatment effect estimates.
hr.aft <- exp(-coef(fit.wbl.aft)["groupGood"])^exp(coef(fit.wbl.aft)["shape"])
# hr.aft
cat("The Hazard Ratio (Good:Poor) is ", round(hr.aft, 4) , ".")

## The Hazard Ratio (Good:Poor) is 0.1878 .

# PH model (for comparison)
fit.wbl.ph <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="weibullPH", data=bc)
print(fit.wbl.ph)

## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##             dist = "weibullPH")
##
## Estimates:
##           data mean  est      L95%      U95%      se      exp(est)
## shape           NA    1.3797  1.2548  1.5170  0.0668         NA
## scale           NA    0.1851  0.1482  0.2312  0.0210         NA
## groupMedium    0.3338  -0.8259 -1.0807 -0.5711  0.1300  0.4378
## groupGood      0.3338  -1.6724 -1.9943 -1.3505  0.1642  0.1878
##           L95%      U95%
## shape           NA         NA
## scale           NA         NA
## groupMedium    0.3394  0.5649
## groupGood      0.1361  0.2591
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -811.7372, df = 4
## AIC = 1631.474

hr.ph <- exp(coef(fit.wbl.ph)["groupGood"])
cat("The Hazard Ratio (Good:Poor) is ", round(hr.ph, 4) , ".")

## The Hazard Ratio (Good:Poor) is 0.1878 .

```

Therefore, the hazard ratio of patients in the good prognostic group to die is 0.1878 compared to patients in the poor prognostic group, ie about an 81% reduction of the hazard.

5.2.2 Theory

For transparency the derivation is given below for the AFT parametrization ³:

First, the documentation for the time-to-event regression model is retrieved using `?flexsurv::flexsurvreg`. Second, the name (stem) of the Weibull distribution function is used, and the respective density distribution function retrieved using `?dweibull`. The documentation for the Weibull density function gives the parametrization for the density function $f()$ and the cumulative density function $F()$. Considering that the hazard ratio is calculated as (0=control, 1=experimental):

$$\text{HR} = \frac{h_1(t)}{h_0(t)} \quad (14)$$

$$(15)$$

The hazard function is:

$$h(t) = \frac{f(t)}{S(t)} \quad (16)$$

³provided by Daniel Dalevi and extended by Wilmar Igl

The survival function $S(t)$ is $S(t) = 1 - F(t)$.

The hazard function $h(t)$ therefore is the density function $f(t)$ (shape parameter a , scale parameter b , replacing x by t):

$$f(t) = \frac{a}{b} \cdot \left(\frac{t}{b}\right)^{a-1} \cdot \exp\left(-\left(\frac{t}{b}\right)^a\right) \quad (17)$$

divided by the survival function $S(t)$:

$$S(t) = 1 - F(t) \quad (18)$$

$$= 1 - (1 - \exp(-\left(\frac{t}{b}\right)^a)) \quad (19)$$

$$= \exp(-\left(\frac{t}{b}\right)^a) \quad (20)$$

resulting in $h(t)$:

$$h(t) = \frac{a}{b} \cdot \left(\frac{t}{b}\right)^{a-1} \quad (21)$$

The hazard function in the control group (=0) is:

$$h_0(t) = \frac{a_0}{b_0} \cdot \left(\frac{t}{b_0}\right)^{a_0-1} \quad (22)$$

The hazard function in the experimental group (=1) is:

$$h_1(t) = \frac{a_1}{b_1} \cdot \left(\frac{t}{b_1}\right)^{a_1-1} \quad (23)$$

Assuming the same shape of Weibull distributions in treatment groups ($a_0 = a_1 = a$), the hazard ratio is:

$$\text{HR} = \frac{h_1(t)}{h_0(t)} \quad (24)$$

$$= \frac{a}{a} \cdot \frac{b_0}{b_1} \cdot \left(\frac{t}{b_1}\right)^{a-1} \cdot \left(\frac{b_0}{t}\right)^{a-1} \quad (25)$$

$$= \frac{b_0}{b_1} \cdot \left(\frac{b_0}{b_1}\right)^{a-1} \quad (26)$$

$$= \left(\frac{b_0}{b_1}\right)^a \quad (27)$$

As described in `?flexsurvreg` the location parameter of the Weibull distribution, which is implemented in `dweibull` is `scale` and the covariates represent treatment effects on this scale parameter.

Therefore, the scale and shape values can be retrieved from the table of coefficients by ($\beta_0 = \text{intercept}$):

$$b_0 = \exp(\beta_0) \quad (28)$$

$$b_1 = \exp(\beta_0 + \beta_1) \quad (29)$$

Therefore, the hazard ratio (experimental: control) is calculated as:

$$\text{HR} = \left(\frac{b_0}{b_1}\right)^a \quad (30)$$

$$= \left(\frac{\exp(\beta_0)}{\exp(\beta_0 + \beta_1)}\right)^a \quad (31)$$

$$= \left(\frac{1}{\exp(\beta_1)}\right)^a \quad (32)$$

$$= \exp(-\beta_1)^a \quad (33)$$

5.3 Gompertz Distribution

5.3.1 Application

The HR, which compares treatment i with a reference treatment, can be computed from the estimates for treatment i from the table of coefficients as follows:

$$\text{HR}_i = \exp(-\beta_i)^a \quad (34)$$

```
fit.gmp <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="gompertz", data=bc)
print(fit.gmp)

## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##   dist = "gompertz")
##
## Estimates:
##           data mean  est      L95%      U95%      se      exp(est)
## shape           NA   0.1315  0.0568   0.2063   0.0381      NA
## rate            NA   0.2231  0.1799   0.2767   0.0245      NA
## groupMedium    0.3338  -0.7847 -1.0403  -0.5290  0.1304   0.4563
## groupGood      0.3338  -1.6219 -1.9451  -1.2988  0.1649   0.1975
##              L95%      U95%
## shape           NA      NA
## rate            NA      NA
## groupMedium    0.3534   0.5892
## groupGood      0.1430   0.2729
##
## N = 686,  Events: 299,  Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -825.1811, df = 4
## AIC = 1658.362

hr.gmp <- exp(coef(fit.gmp)["groupGood"])
cat("The Hazard Ratio (Good:Poor) is ", round(hr.gmp, 4) , ".")

## The Hazard Ratio (Good:Poor) is  0.1975 .
```

Therefore, the hazard ratio of patients in the good prognostic group to die is 0.1975 compared to patients in the poor prognostic group, ie about an 80% reduction of the hazard.

5.3.2 Theory

For transparency the derivation is given below:

First, the documentation for the time-to-event regression model is retrieved using `?flexsurv::flexsurvreg`. Second, the name (stem) of the Gompertz distribution function is used, and the respective density distribution function retrieved using `?dgompertz`. The documentation for the Gompertz density function gives the parametrization for the hazard function $h(t)$. Considering that the hazard ratio is calculated as (0=control, 1=experimental):

$$\text{HR} = \frac{h_1(t)}{h_0(t)} \quad (35)$$

$$(36)$$

The hazard function is (shape parameter a , rate parameter b):

$$h(t) = b \cdot \exp(a \cdot t) \quad (37)$$

The hazard function in the control group (=0) is:

$$h_0(t) = b_0 \cdot \exp(a_0 \cdot t) \quad (38)$$

The hazard function in the experimental group (=1) is:

$$h_1(t) = b_1 \cdot \exp(a_1 \cdot t) \quad (39)$$

Assuming the same shape of Weibull distributions in treatment groups ($a_0 = a_1 = a$), the hazard ratio is:

$$\text{HR} = \frac{h_1(t)}{h_0(t)} \quad (40)$$

$$= \frac{b_1}{b_0} \cdot \frac{\exp(a \cdot t)}{\exp(a \cdot t)} \quad (41)$$

$$= \frac{b_1}{b_0} \quad (42)$$

As described in `?flexsurvreg` the location parameter of the Gompertz distribution, which is implemented in `dgompertz` is `rate` and the covariates represent treatment effects on the log of the rate parameter.

Therefore, the rate values can be retrieved from the table of coefficients by ($\beta_0 = \text{intercept}$):

$$b_0 = \exp(\beta_0) \quad (43)$$

$$b_1 = \exp(\beta_0 + \beta_1) \quad (44)$$

Therefore, the hazard ratio (experimental: control) is calculated as:

$$\text{HR} = \frac{b_1}{b_0} \quad (45)$$

$$= \frac{\exp(\beta_0 + \beta_1)}{\exp(\beta_0)} \quad (46)$$

$$= \exp(\beta_1) \quad (47)$$

6 Calculation of an (average) hazard ratio for accelerated failure time models

6.1 Weighted average hazard ratios

This section will describe the calculation of a weighted average hazard ratios as proposed by Schemper et al. [2009]. They give three different definitions for the weighted average hazard ratio:

1. sAHR: simple average hazard ratio
2. gAHR: geometric average hazard ratio
3. AHR: average hazard ratio with weights by Kalbfleisch and Prentice (cf “odds of concordance”)

6.1.1 sAHR

The sAHR [Schemper et al., 2009] is intuitive, but not symmetric in $h_0(t)$ and $h_1(t)$ and therefore not recommended:

$$\text{sAHR} = \int \frac{h_1(x)}{h_0(x)} \cdot w(t) \cdot f(t) dt \quad (48)$$

```
## Definition of "simple" weighted average hazard ratio (using S(t) as weights)
## cf Schemper, 2009, sAHR definition
calcAHRsmp <- function(x, h0, h1, ft, wt){
  integrand <- function(x){h1(x)/h0(x) * wt(x) * ft(x) }
  ahr <- integrate(integrand, lower=0, upper=x)$value
  return(ahr)
}
```

6.1.2 gAHR

The gAHR [Schemper et al., 2009] does not suffer from the disadvantages of the sAHR, but represents a geometric average hazard ratio.

$$\text{gAHR} = \exp \left[\int \log \left(\frac{h_1(x)}{h_0(x)} \right) \cdot w(t) \cdot f(t) dt \right] \quad (49)$$

```
## Definition of weighted average hazard ratio using S(t) as weights
## cf Schemper, 2009, gAHR definition
calcAHRgeo <- function(x, h0, h1, ft, wt){
  integrand <- function(x){ log(h1(x)/h0(x)) * wt(x) * ft(x) }
  logahr <- integrate(integrand, lower=0, upper=x)$value
  ahr <- exp(logahr)
  return(ahr)
}
```

6.1.3 AHR

The AHR with Kalbfleisch & Prentice weights is the preferred definition for the average hazard ratio by Schemper et al. [2009], because it allows the interpretation of the hazard ratio as *odds of concordance* (OC). The OC is the probability $P(T_1 < T_0)$ that a randomly chosen survival time T_1 from group G_1 is smaller than a randomly chosen survival time T_0 from group G_0 . It is an intuitive and elementary non-parametric measure of effect size, characterizing the degree of separation of the distributions of the survival times of two groups [Schemper et al., 2009, p. 2476]. Schemper et al. [2009] conclude that the AHR_{OC} (odds of concordance) definition is a “close surrogate” to the AHR_{WCOX} (Weighted Cox), which uses $S(t)$ as weights, but is computationally simpler.

$$\text{AHR}_{OC} = \frac{\int S_0(t) \cdot f_1(t) dt}{\int S_1(t) \cdot f_0(t) dt} \quad (50)$$

```
## Definition of weighted average hazard ratio using weights
## by Prentice & Kalbfleisch (Schemper, 2009, Appendix A)
calcAHRoc <- function(x, f0, f1, S0, S1){
  integrand1 <- function(x){ S0(x) * f1(x) }
  term1 <- integrate(integrand1, lower=0, upper=x)$value

  integrand2 <- function(x){ S1(x) * f0(x) }
  term2 <- integrate(integrand2, lower=0, upper=x)$value

  ahr <- term1 / term2
  return(ahr)
}
```

6.1.4 Alternative overall effect sizes

Royston and Parmar [2013] remain “unconvinced” by proposals from Schemper et al. [2009] and recommend as effect size to describe treatment effects in data, where the proportional hazards assumption is violated, the differences of the *Restricted Mean Survival Times* (RMST)⁴, which is, however, beyond the scope of this tutorial.

⁴An R package to calc RMST is available here: <https://github.com/scientific-computing-solutions/RMST>

6.2 Log-logistic Distribution

6.2.1 Application

Model parameters and functions Model parameters and functions are calculated.

```
fit.llg <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="llogis", data=bc)
print(fit.llg)
```

```
## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##   dist = "llogis")
##
## Estimates:
##           data mean  est    L95%   U95%   se    exp(est)  L95%
## shape           NA  1.7546  1.5959  1.9291  0.0849     NA      NA
## scale           NA  2.3352  2.0369  2.6771  0.1628     NA      NA
## groupMedium    0.3338  0.6653  0.4655  0.8652  0.1020  1.9451  1.5928
## groupGood      0.3338  1.2882  1.0564  1.5201  0.1183  3.6264  2.8760
##           U95%
## shape           NA
## scale           NA
## groupMedium    2.3754
## groupGood      4.5726
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -798.8476, df = 4
## AIC = 1605.695
```

```
plot(fit.llg, col=c("red","green","blue"))
legend("topright", fill=c("red","green","blue"), legend=c("Good", "Medium", "Poor"))
```

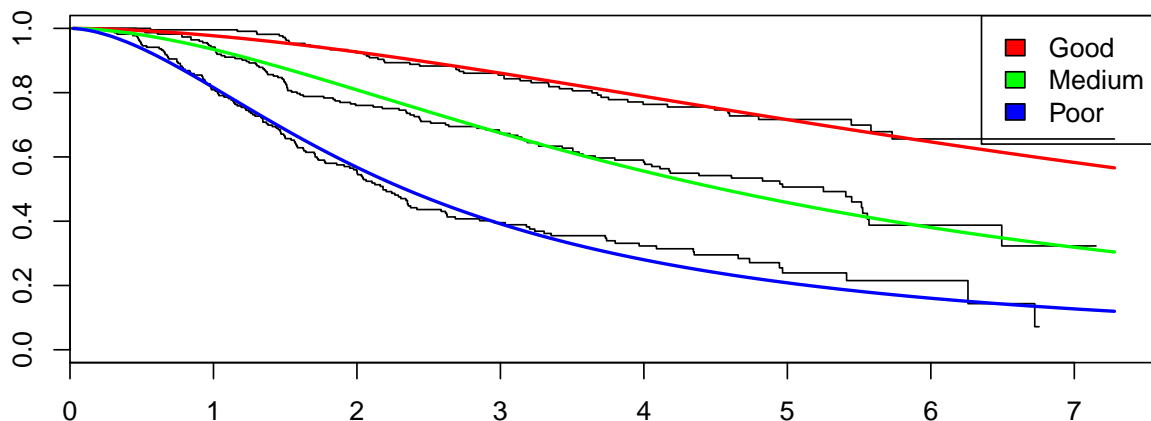


Figure 1: Estimated survival probabilities from fitted survival model with log-logistic distribution

Functions are parametrized with the estimated coefficients for the reference group (ie “poor” prognosis) and comparison group (ie “good” prognosis):

```

# Probability Density Functions
f0 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"])){
  dllogis(x=x, shape=shape.a, scale=scale.b)
}

f1 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"]+coef(fit.llg)["groupGood"])){
  dllogis(x=x, shape=shape.a, scale=scale.b)
}

ft <- function(x){ (f0(x) + f1(x))}

# Survival functions
S0 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"])){
  pllogis(q=x, shape=shape.a, scale=scale.b, lower=FALSE)
}

S1 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"]+coef(fit.llg)["groupGood"])){
  pllogis(q=x, shape=shape.a, scale=scale.b, lower=FALSE)
}

St <- function(x){ ( S0(x) * f0(x) + S1(x) * f1(x) ) / ( f0(x) + f1(x) ) }
## St <- function(x){ (S0(x) + S1(x))/2} ## assuming same number of events

# Hazard functions
h0 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"])){
  hllogis(x=x, shape=shape.a, scale=scale.b)
}

h1 <- function(x,
               shape.a=exp(coef(fit.llg)["shape"]),
               scale.b=exp(coef(fit.llg)["scale"]+coef(fit.llg)["groupGood"])){
  hllogis(x=x, shape=shape.a, scale=scale.b)
}

```

sAHR The sAHR (“simple” definition) is:

```
## sAHR for max/2 FU period (50%)
sahr.maxt50p <- calcAHRsmp(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt50p)

## [1] 0.09791712

## gAHR for max FU period (100%)
sahr.maxt100p <- calcAHRsmp(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt100p)

## [1] 0.1768691
```

```
ff <- function(x){calcAHRsmp(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

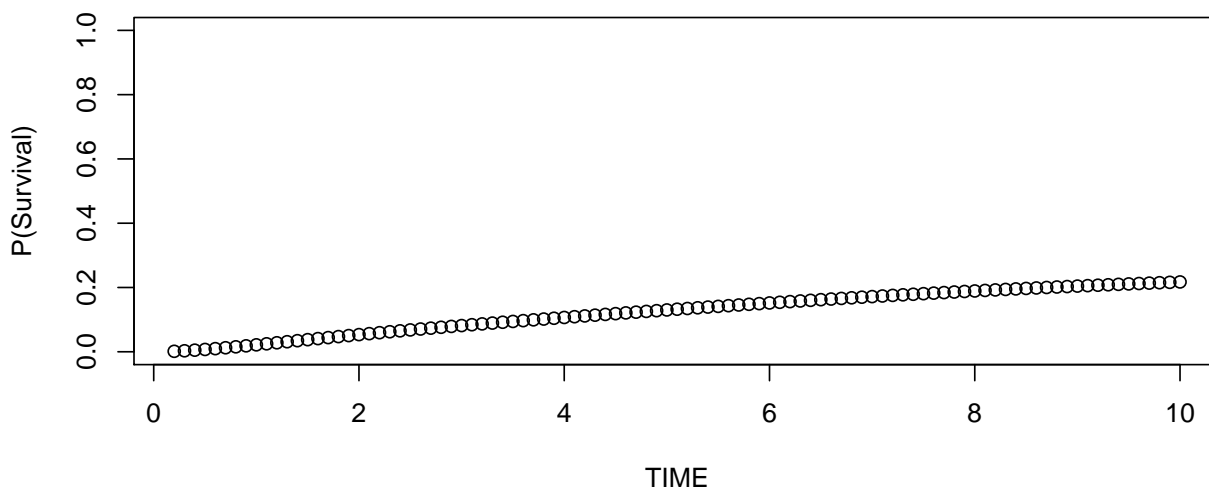


Figure 2: AHR(simple) for log-logistic distribution over time

gAHR The AHR (“geometric” definition) is:

```
## gAHR for max/2 FU period (50%)
gahr.maxt50p <- calcAHRgeo(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt50p)

## [1] 0.312349

## gAHR for max FU period (100%)
gahr.maxt100p <- calcAHRgeo(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt100p)

## [1] 0.2514565
```

```
ff <- function(x){calcAHRgeo(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

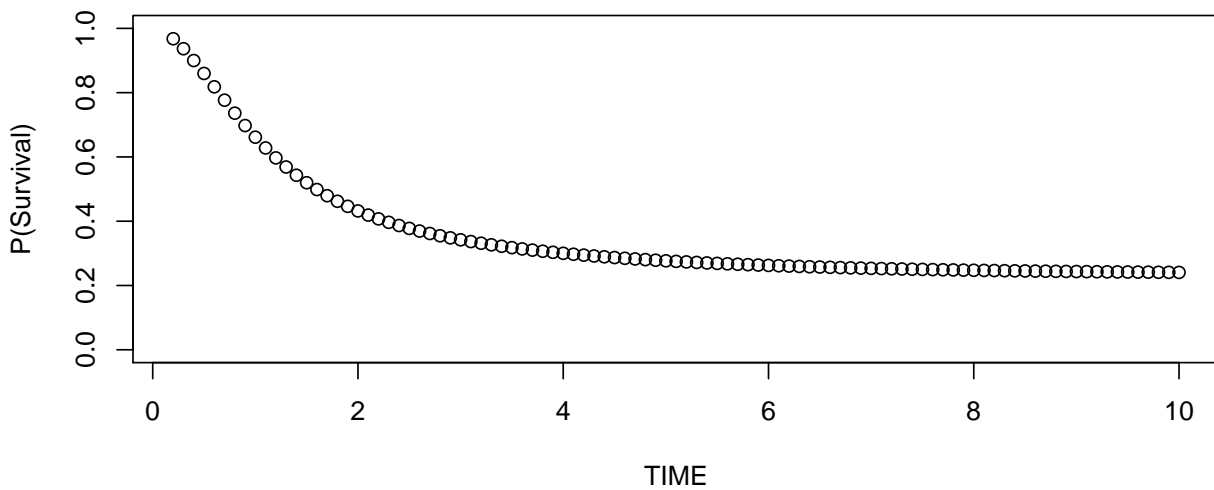


Figure 3: AHR(geo) for log-logistic distribution over time

AHR(OC) The AHR (“odds of concordance”) is:

```
## AHR for max/2 FU period (50%)
ahr.maxt50p <- calcAHRoc(x=max(bc$recyrs)/2, f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)/2), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt50p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 4 years
## the average hazard ratio (odds of concordance) is 0.1593 .

## AHR for max FU period (100%)
ahr.maxt100p <- calcAHRoc(x=max(bc$recyrs), f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt100p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 7 years
## the average hazard ratio (odds of concordance) is 0.1939 .
```

```
ff <- function(x){calcAHRoc(x, f0, f1, S0, S1)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

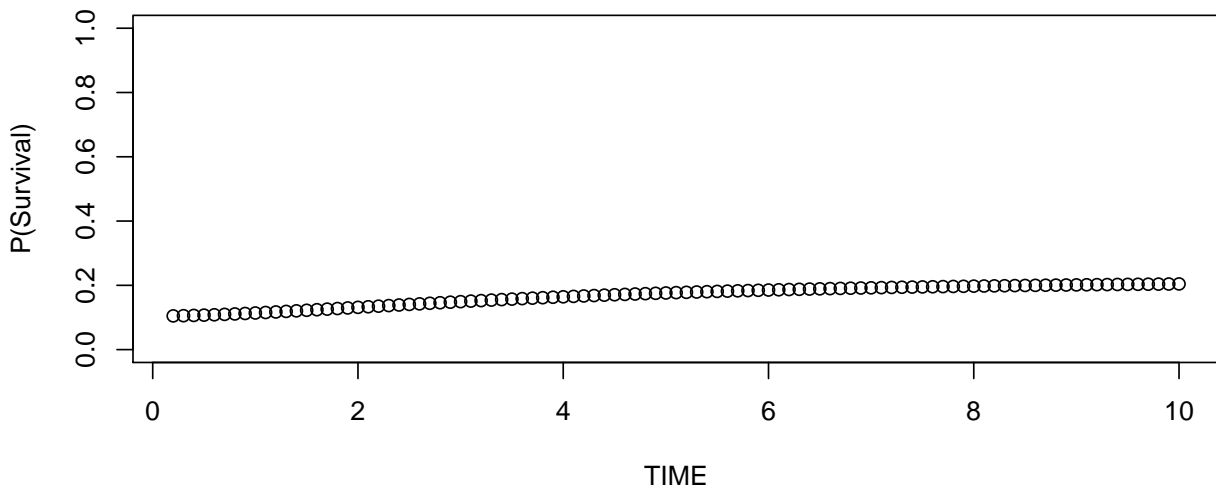


Figure 4: AHR(OC) for log-logistic distribution over time

6.2.2 Theory

The *probability density function* (density function) of the log-logistic distribution with *shape* parameter $a > 0$ and *scale* parameter $b > 0$ is (cf `?dllogis`):

$$f(x|a, b) = \frac{a}{b} \cdot \left(\frac{x}{b}\right)^{a-1} / \left(1 + \left(\frac{x}{b}\right)^a\right)^2 \quad (51)$$

The *cumulative density function* (probability function) is (cf `?pllogis`)

$$F(x|a, b) = 1 - 1 / \left(1 + \left(\frac{x}{b}\right)^a\right) \quad (52)$$

The *survival function* is (cf `?pllogis(..., lower=FALSE)`):

$$S(x|a, b) = 1 - F(x|a, b) \quad (53)$$

$$= 1 / \left(1 + \left(\frac{x}{b}\right)^a\right) \quad (54)$$

The *hazard function* is (cf `?hllogis`):

$$h(x|a, b) = \frac{a}{b} \cdot \left(\frac{x}{b}\right)^{a-1} / \left(1 + \left(\frac{x}{b}\right)^a\right) \quad (55)$$

The hazard function can also be calculated by `dllogis(...)/pllogis(..., lower=FALSE)`.

6.3 Log-normal Distribution

6.3.1 Application

Model parameters and functions Model parameters and functions are calculated.

```
fit.lnm <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="lnorm", data=bc)
print(fit.lnm)
```

```
## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##   dist = "lnorm")
##
## Estimates:
##           data mean  est    L95%   U95%   se    exp(est)  L95%
## meanlog           NA  0.8719  0.7332  1.0107  0.0708     NA      NA
## sdlog            NA  0.9748  0.8938  1.0630  0.0431     NA      NA
## groupMedium      0.3338  0.6443  0.4444  0.8442  0.1020  1.9046  1.5595
## groupGood        0.3338  1.2829  1.0557  1.5100  0.1159  3.6070  2.8740
##
##           U95%
## meanlog           NA
## sdlog            NA
## groupMedium      2.3262
## groupGood        4.5269
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -790.5603, df = 4
## AIC = 1589.121
```

```
plot(fit.lnm, col=c("red","green","blue"))
legend("topright", fill=c("red","green","blue"), legend=c("Good", "Medium", "Poor"))
```

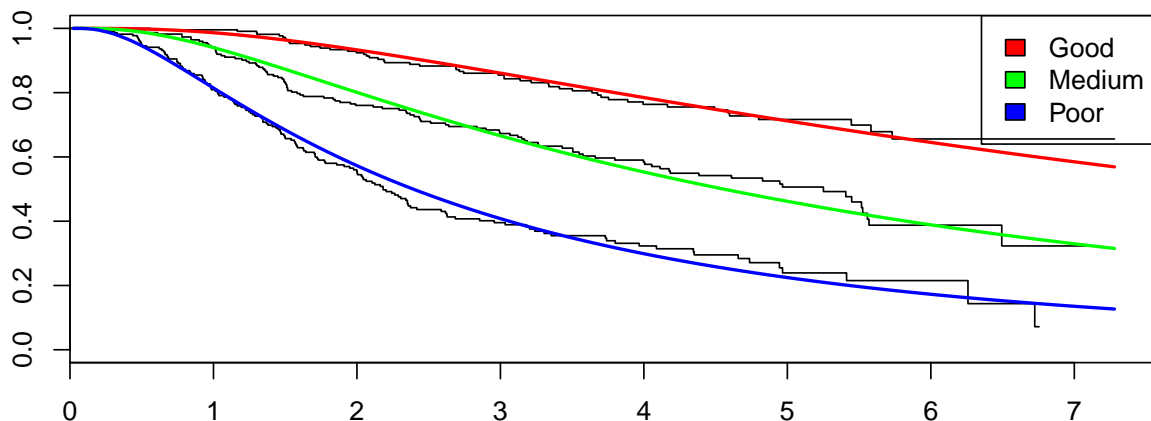


Figure 5: Estimated survival probabilities from fitted survival model with log-normal distribution

Functions are parametrized with the estimated coefficients for the reference group (ie “poor” prognosis) and comparison group (ie “good” prognosis):

```

# Probability Density Functions
f0 <- function(x,
               meanlog = coef(fit.lnm)["meanlog"],
               sdlog    = exp(coef(fit.lnm)["sdlog"])){
  dlnorm(x=x, meanlog=meanlog, sdlog=sdlog)
}

f1 <- function(x,
               meanlog = coef(fit.lnm)["meanlog"]+coef(fit.lnm)["groupGood"],
               sdlog    = exp(coef(fit.lnm)["sdlog"])){
  dlnorm(x=x, meanlog=meanlog, sdlog=sdlog)
}

ft <- function(x){ (f0(x) + f1(x))}

# Survival functions
S0 <- function(x,
               meanlog = coef(fit.lnm)["meanlog"],
               sdlog    = exp(coef(fit.lnm)["sdlog"])){
  plnorm(q=x, meanlog=meanlog, sdlog=sdlog, lower=FALSE)
}

S1 <- function(x,
               meanlog = coef(fit.lnm)["meanlog"]+coef(fit.lnm)["groupGood"],
               sdlog    = exp(coef(fit.lnm)["sdlog"])){
  plnorm(q=x, meanlog=meanlog, sdlog=sdlog, lower=FALSE)
}

St <- function(x){ ( S0(x) * f0(x) + S1(x) * f1(x) ) / ( f0(x) + f1(x) ) }
## St <- function(x){ (S0(x) + S1(x))/2} ## assuming same number of events

# Hazard functions
h0 <- function(x, f=f0, S=S0){f(x)/S(x)}
h1 <- function(x, f=f1, S=S1){f(x)/S(x)}

```

sAHR The sAHR (“simple” definition) is:

```
## sAHR for max/2 FU period (50%)
sahr.maxt50p <- calcAHRsmp(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt50p)

## [1] 0.09965374

## gAHR for max FU period (100%)
sahr.maxt100p <- calcAHRsmp(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt100p)

## [1] 0.1772884
```

```
ff <- function(x){calcAHRsmp(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

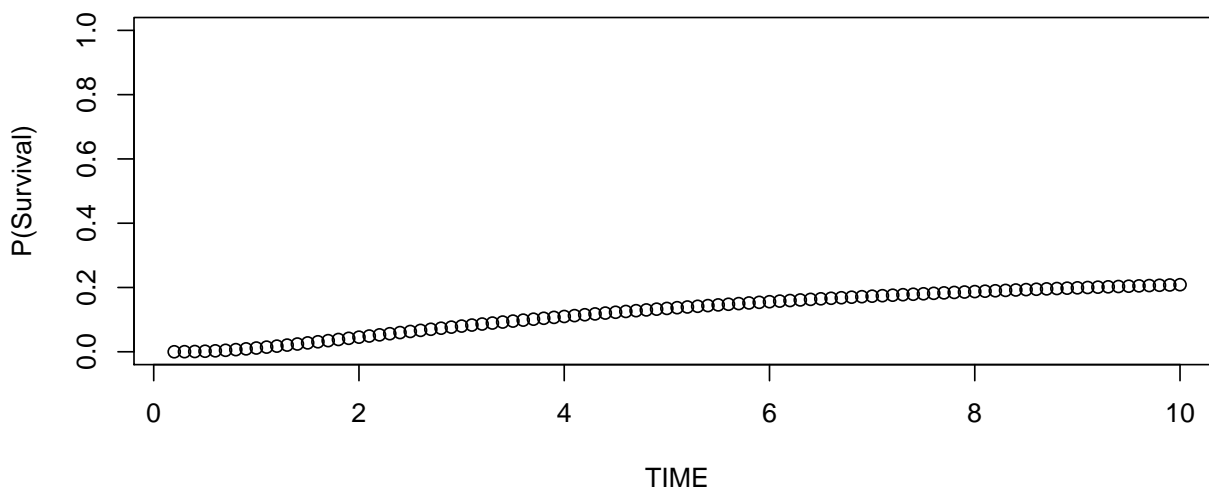


Figure 6: AHR(simple) for log-normal distribution over time

gAHR The AHR (“geometric” definition) is:

```
## gAHR for max/2 FU period (50%)
gahr.maxt50p <- calcAHRgeo(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt50p)

## [1] 0.292448

## gAHR for max FU period (100%)
gahr.maxt100p <- calcAHRgeo(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt100p)

## [1] 0.2346106
```

```
ff <- function(x){calcAHRgeo(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

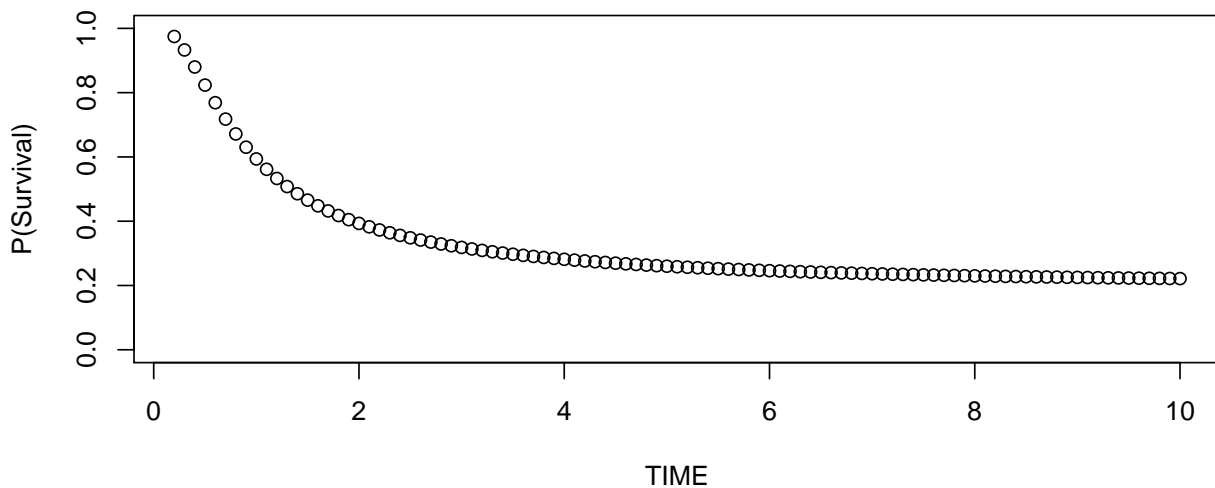


Figure 7: AHR(geo) for log-normal distribution over time

AHR(OC) The AHR (“odds of concordance”) is:

```
## AHR for max/2 FU period (50%)
ahr.maxt50p <- calcAHRoc(x=max(bc$recyrs)/2, f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)/2), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt50p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 4 years
## the average hazard ratio (odds of concordance) is 0.1606 .

## AHR for max FU period (100%)
ahr.maxt100p <- calcAHRoc(x=max(bc$recyrs), f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt100p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 7 years
## the average hazard ratio (odds of concordance) is 0.1974 .
```

```
ff <- function(x){calcAHRoc(x, f0, f1, S0, S1)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

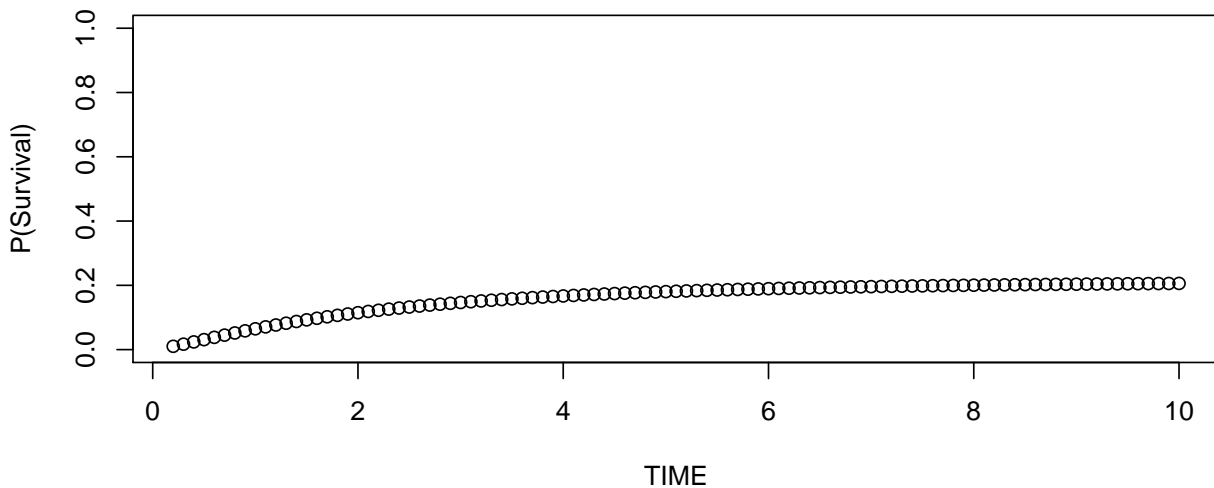


Figure 8: AHR(OC) for log-normal distribution over time

6.3.2 Theory

The *probability density function* (PDF) of the log-normal distribution with μ as the mean and σ as the standard deviation of the logarithm is (cf `?dlnorm`):

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma \cdot x} \cdot \exp\left(-\frac{(\log(x) - \mu)^2}{2 \cdot \sigma^2}\right) \quad (56)$$

The formula for the *cumulative density function* (CDF) (cf `?plnorm`) is not given. The formula for the *survival function* (1-CDF) (cf `?plnorm(..., lower=FALSE)`) is not given. The formula for the *hazard function* is not given (cf `dlnorm(...)/plnorm(..., lower=FALSE)`).

6.4 Generalized Gamma Distribution

6.4.1 Application

Model parameters and functions Model parameters and functions are calculated.

```
fit.ggm <- flexsurvreg(formula = Surv(recyrs, censrec) ~ group, dist="gengamma", data=bc)
print(fit.ggm)
```

```
## Call:
## flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,
##   dist = "gengamma")
##
## Estimates:
##           data mean  est      L95%      U95%      se      exp(est)
## mu           NA    0.6552  0.4276  0.8829  0.1161      NA
## sigma        NA    1.0561  0.9622  1.1591  0.0502      NA
## Q            NA   -0.5926 -1.0589 -0.1263  0.2379      NA
## groupMedium  0.3338  0.6338  0.4299  0.8376  0.1040  1.8847
## groupGood    0.3338  1.2831  1.0638  1.5023  0.1119  3.6076
##           L95%      U95%
## mu           NA      NA
## sigma        NA      NA
## Q            NA      NA
## groupMedium  1.5370  2.3109
## groupGood    2.8973  4.4921
##
## N = 686, Events: 299, Censored: 387
## Total time at risk: 2111.978
## Log-likelihood = -787.3636, df = 5
## AIC = 1584.727
```

```
plot(fit.ggm, col=c("red","green","blue"))
legend("topright", fill=c("red","green","blue"), legend=c("Good", "Medium", "Poor"))
```

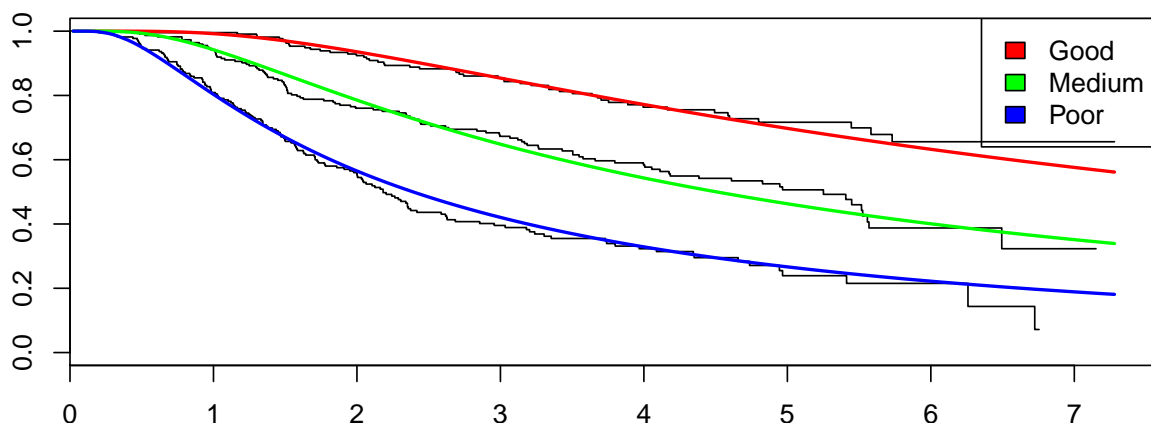


Figure 9: Estimated survival probabilities from fitted survival model with generalized gamma distribution

Functions are parametrized with the estimated coefficients for the reference group (ie “poor” prognosis) and comparison group (ie “good” prognosis):

```

# Probability Density Functions
f0 <- function(x,
  mu      = coef(fit.ggm)["mu"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  dgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
}

f1 <- function(x,
  mu      = coef(fit.ggm)["mu"]+coef(fit.ggm)["groupGood"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  dgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
}

ft <- function(x){ (f0(x) + f1(x))}

# Cumulative Density Functions
S0 <- function(x,
  mu      = coef(fit.ggm)["mu"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  pgengamma(q=x, mu=mu, sigma=sigma, Q=Q, lower=FALSE)
}

S1 <- function(x,
  mu      = coef(fit.ggm)["mu"]+coef(fit.ggm)["groupGood"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  pgengamma(q=x, mu=mu, sigma=sigma, Q=Q, lower=FALSE)
}

St <- function(x){ ( S0(x) * f0(x) + S1(x) * f1(x) ) / ( f0(x) + f1(x) ) }
## St <- function(x){ (S0(x) + S1(x))/2} ## assuming same number of events

## Hazard functions
h0 <- function(x,
  mu      = coef(fit.ggm)["mu"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  hgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
}

h1 <- function(x,
  mu      = coef(fit.ggm)["mu"]+coef(fit.ggm)["groupGood"],
  sigma  = exp(coef(fit.ggm)["sigma"]),
  Q       = coef(fit.ggm)["Q"]){
  hgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
}

```

sAHR The sAHR (“simple” definition) is:

```
## sAHR for max/2 FU period (50%)
sahr.maxt50p <- calcAHRsmp(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt50p)

## [1] 0.1131397

## gAHR for max FU period (100%)
sahr.maxt100p <- calcAHRsmp(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(sahr.maxt100p)

## [1] 0.2209214
```

```
ff <- function(x){calcAHRsmp(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

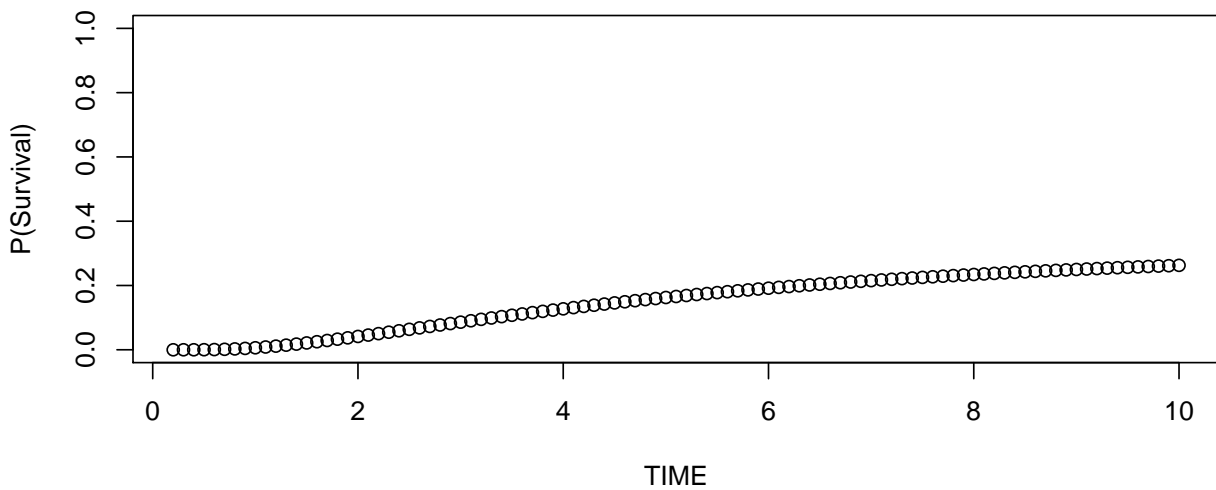


Figure 10: AHR(simple) for gen-gamma distribution over time

gAHR The AHR (“geometric” definition) is:

```
## gAHR for max/2 FU period (50%)
gahr.maxt50p <- calcAHRgeo(x=max(bc$recyrs)/2, h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt50p)

## [1] 0.2555621

## gAHR for max FU period (100%)
gahr.maxt100p <- calcAHRgeo(x=max(bc$recyrs), h0=h0, h1=h1, ft=ft, wt=St)
print(gahr.maxt100p)

## [1] 0.2217868
```

```
ff <- function(x){calcAHRgeo(x=x, h0=h0, h1=h1, ft=ft, wt=St)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

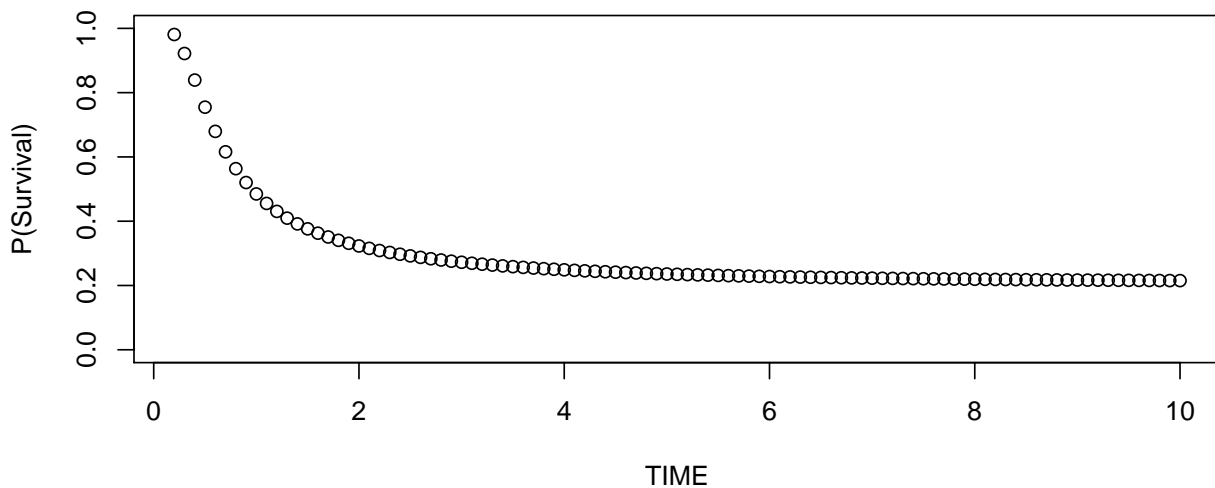


Figure 11: AHR(geo) for gen-gamma distribution over time

AHR(OC) The AHR (“odds of concordance”) is:

```
## AHR for max/2 FU period (50%)
ahr.maxt50p <- calcAHRoc(x=max(bc$recyrs)/2, f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)/2), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt50p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 4 years
## the average hazard ratio (odds of concordance) is 0.1711 .

## AHR for max FU period (100%)
ahr.maxt100p <- calcAHRoc(x=max(bc$recyrs), f0=f0, f1=f1, S0=S0, S1=S1)
cat("The average HR (good:poor) for patients with a follow-up period of",
    round(max(bc$recyrs)), "years",
    "\nthe average hazard ratio (odds of concordance) is ",
    round(ahr.maxt100p, 4), ".")

## The average HR (good:poor) for patients with a follow-up period of 7 years
## the average hazard ratio (odds of concordance) is 0.2271 .
```

```
ff <- function(x){calcAHRoc(x, f0, f1, S0, S1)}
tt <- seq(0.2,10,0.1)
yy <- sapply(tt, FUN=ff)
plot(tt, yy, ylim=c(0,1), xlab="TIME", ylab="P(Survival)")
```

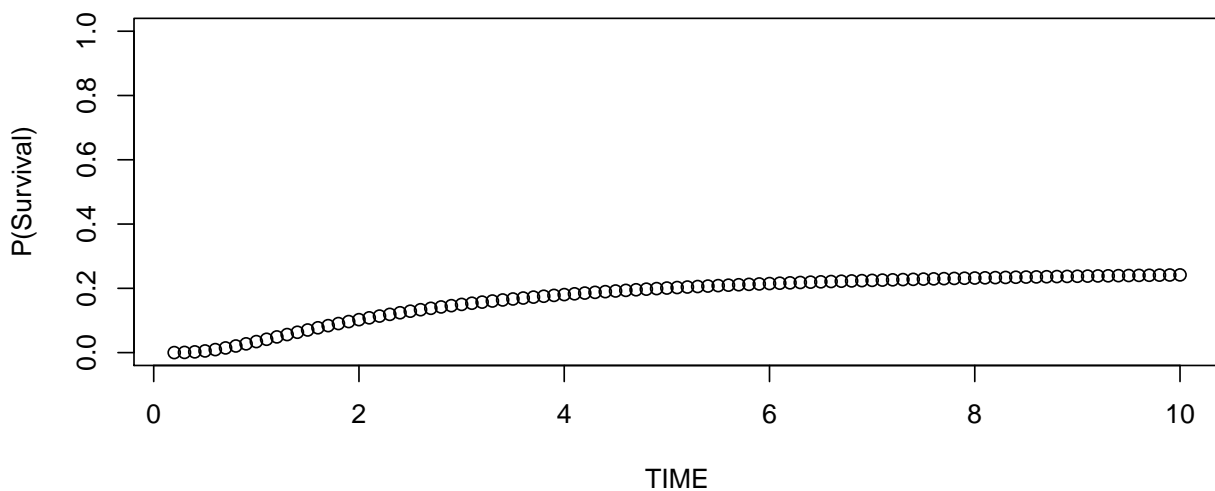


Figure 12: AHR(OC) for gen-gamma distribution over time

6.4.2 Theory

The *probability density function* of the generalized gamma function with ‘*location*’ parameter μ and ‘*scale*’ parameter σ is (cf `?dgengamma`):

$$f(x|\mu, \sigma, Q) = |Q| \cdot (Q^{-2})^{Q^{-2}} / (\sigma \cdot x \cdot \Gamma(Q^{-2})) \cdot \exp(Q^{-2} \cdot (Q \cdot w - \exp(Q \cdot w))) \quad (57)$$

$$g \sim \Gamma(Q^{-2}, 1) \quad (58)$$

$$w = \log(Q^2 \cdot g) / Q \quad (59)$$

$$x = \exp(\mu + \sigma \cdot w) \quad (60)$$

The formula for the *cumulative density function* (CDF, cf `?pgengamma`) is not given. The formula for the *survival function* (1-CDF, cf `?pgengamma(..., lower=FALSE)`) is not given. The formula for the *hazard function* (cf `?hgengamma`) is not given.

The parametrization of the `dgengamma` function (and related functions) is different from the original parametrization [Stacy, 1962]. An R function with the original parametrization is available in the `flexsurv` package (cf `?dgengamma.orig`).

The generalized gamma distribution simplifies to the log-normal, Weibull, and gamma distributions with the following parameterisations (cf `?dgengamma` (see Cox et al. [2007]):

```
# Don't run:
# dgengamma(x, mu, sigma, Q=0)      = dlnorm(x, mu, sigma)
# dgengamma(x, mu, sigma, Q=1)     = dweibull(x, shape=1/sigma, scale=exp(mu))
# dgengamma(x, mu, sigma, Q=sigma) = dgamma(x, shape=1/sigma^2, rate=exp(-mu) / sigma^2)
```

References

- Christopher Cox, Haitao Chu, Michael F. Schneider, and Alvaro Muoz. Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. *Statistics in Medicine*, 26(23):4352–4374, October 2007. ISSN 0277-6715. doi: 10.1002/sim.2836.
- Patrick Royston and Mahesh KB Parmar. Restricted mean survival time: an alternative to the hazard ratio for the design and analysis of randomized trials with a time-to-event outcome. *BMC Medical Research Methodology*, 13:152, December 2013. ISSN 1471-2288. doi: 10.1186/1471-2288-13-152. URL <https://doi.org/10.1186/1471-2288-13-152>.
- Michael Schemper, Samo Wakounig, and Georg Heinze. The estimation of average hazard ratios by weighted Cox regression. *Statistics in Medicine*, 28(19):2473–2489, August 2009. ISSN 1097-0258. doi: 10.1002/sim.3623.
- E. W. Stacy. A Generalization of the Gamma Distribution. *The Annals of Mathematical Statistics*, 33(3): 1187–1192, September 1962. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177704481. URL <https://projecteuclid.org/euclid.aoms/1177704481>.
- Zhongheng Zhang. Parametric regression model for survival data: Weibull regression model as an example. *Annals of Translational Medicine*, 4(24), December 2016. ISSN 2305-5839. doi: 10.21037/atm.2016.08.45. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5233524/>.

7 Software information

```
## R version 3.2.3 (2015-12-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Linux Mint 18.2
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] coxphw_4.0.0  flexsurv_1.1  survival_2.41-3  ggplot2_2.2.1
## [5] xtable_1.8-2  knitr_1.18
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.14      magrittr_1.5      splines_3.2.3
## [4] munsell_0.4.3     colorspace_1.3-2  lattice_0.20-33
## [7] quadprog_1.5-5    rlang_0.1.6       highr_0.6
## [10] stringr_1.2.0     plyr_1.8.4        tools_3.2.3
## [13] grid_3.2.3        gtable_0.2.0      lazyeval_0.2.1
## [16] tibble_1.4.1      Matrix_1.2-3      RColorBrewer_1.1-2
## [19] mstate_0.2.10     deSolve_1.20      evaluate_0.10.1
## [22] stringi_1.1.6     compiler_3.2.3    pillar_1.0.1
## [25] muhaz_1.2.6       scales_0.5.0      mvtnorm_1.0-6
##
## Document compile time: 0.2207131 minutes
```