

8. Survival analysis in HTA

Gianluca Baio

Department of Statistical Science | University College London

- ✉ g.baio@ucl.ac.uk
- 🌐 <https://gianluca.statistica.it/>
- 🌐 <https://egon.stats.ucl.ac.uk/research/statistics-health-economics/>
- 🔗 <https://github.com/giabaio>
- 🔗 <https://github.com/StatisticsHealthEconomics>
- 🐦 [@gianlubaio](https://twitter.com/gianlubaio)

Bayesian Methods in Health Economics, Lausanne

- Survival data in health economic evaluations
 - Extrapolation from clinical trials
 - Parametric models
- Issues with Bayesian modelling
 - Parameterisations
 - Convergence
 - Alternative methods for Bayesian inference
- PSA in survival analysis
 - Propagating the relevant uncertainty

References

 *The BUGS Book*, chapter 11.11

 [Book website](#)

 NICE DSU Evidence Synthesis Technical Support Document Series – TSD14

 NICE DSU Evidence Synthesis Technical Support Document Series – TSD16

 NICE DSU Evidence Synthesis Technical Support Document Series – TSD19

 *Doing Bayesian Data Analysis*, chapter 14

 [Book website](#)

Relevant quantities

- Outcome is **time to event** $T > 0$, a continuous random variable with sample space (support) $[0, \infty)$
- Relevant quantities are: the **density** and **cumulative distribution** functions

$$F(t) = \Pr(T \leq t) = \int_0^t f(u) du$$

the "survivor" (often referred to as "survival") function

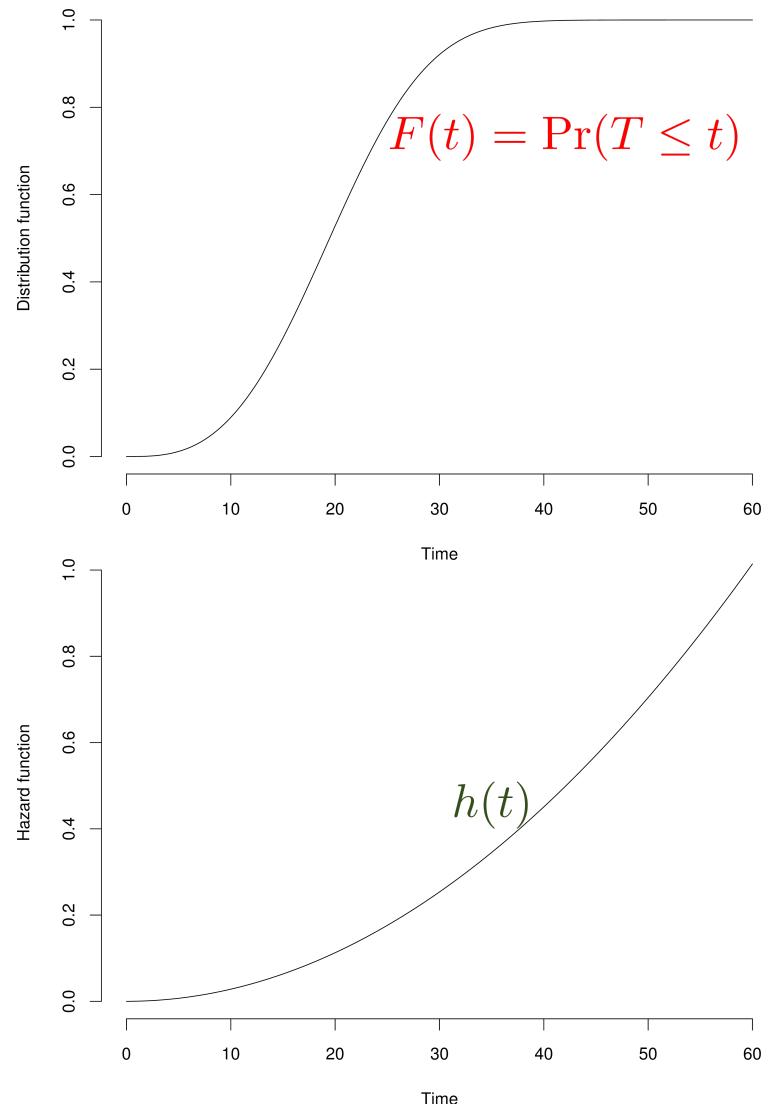
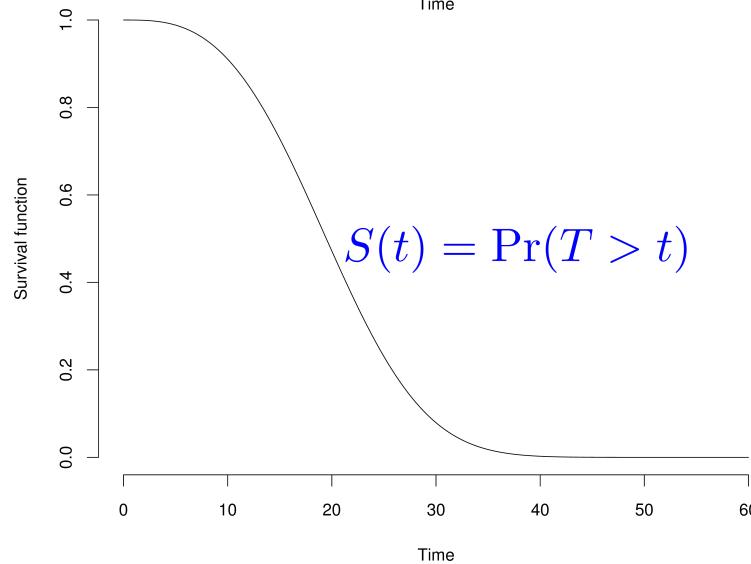
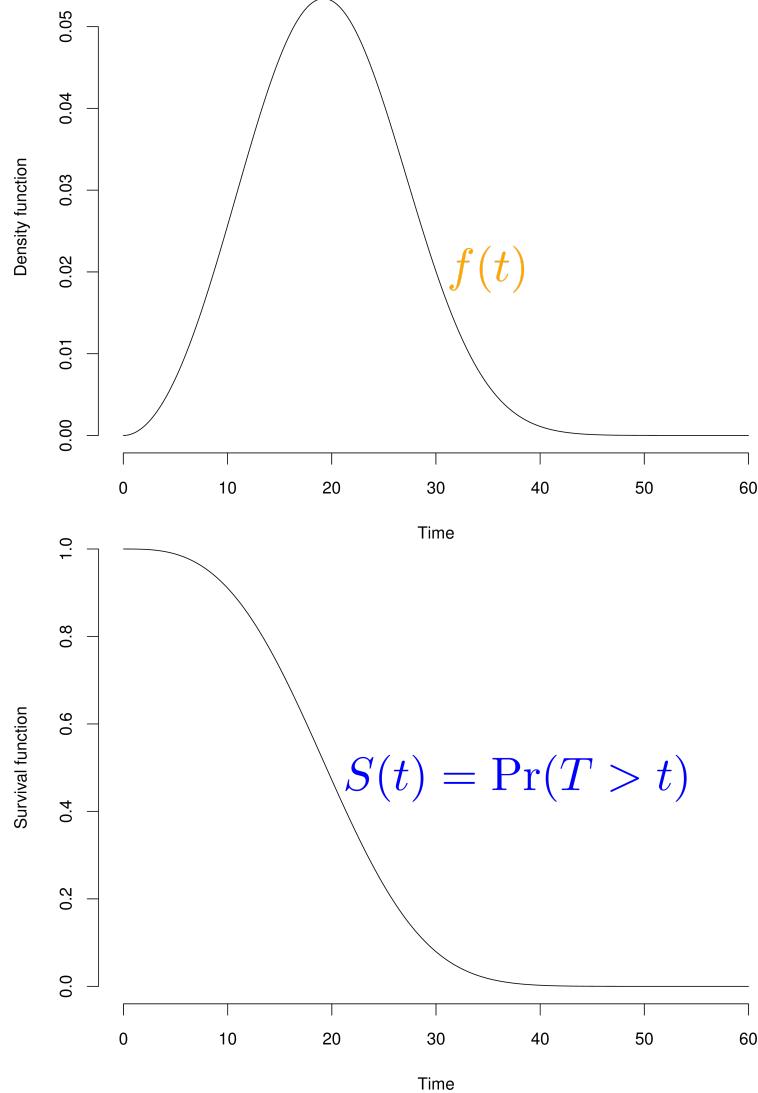
$$S(t) = 1 - F(t) = \Pr(T > t)$$

the "hazard" and the "cumulative hazard" functions

$$h(t) = \lim_{\Delta \rightarrow 0} \frac{\Pr(t \leq T \leq t + \Delta \mid T > t)}{\Delta} \quad \text{and} \quad H(t) = \int_0^t h(u) du$$

- Since $F(t)$, $S(t)$, $f(t)$, $h(t)$ and $H(t)$ are all connected, specifying one of them is sufficient to fully characterise the survival model

Survival analysis



Recap

- 1 The density function is the derivative wrt time of the cumulative function:

$$f(t) = \frac{d}{dt} F(t) = F'(t)$$

- 2 The "instantaneous" hazard function is a conditional probability, so can be specified as the ratio of the joint probability to the marginal probability of the conditioning event:

$$h(t) = \frac{\lim_{\Delta \rightarrow 0} \Pr(t \leq T \leq t + \Delta)/\Delta}{\Pr(T > t)} = \frac{f(t)}{S(t)}$$

- 3 The derivative wrt time of the survival function is minus the density function:

$$S'(t) = \frac{d}{dt} S(t) = \frac{d}{dt} [1 - F(t)] = -f(t)$$

- 4 The hazard function can be represented as the ratio of the derivative to the actual survival function:

$$h(t) = -\frac{S'(t)}{S(t)}$$

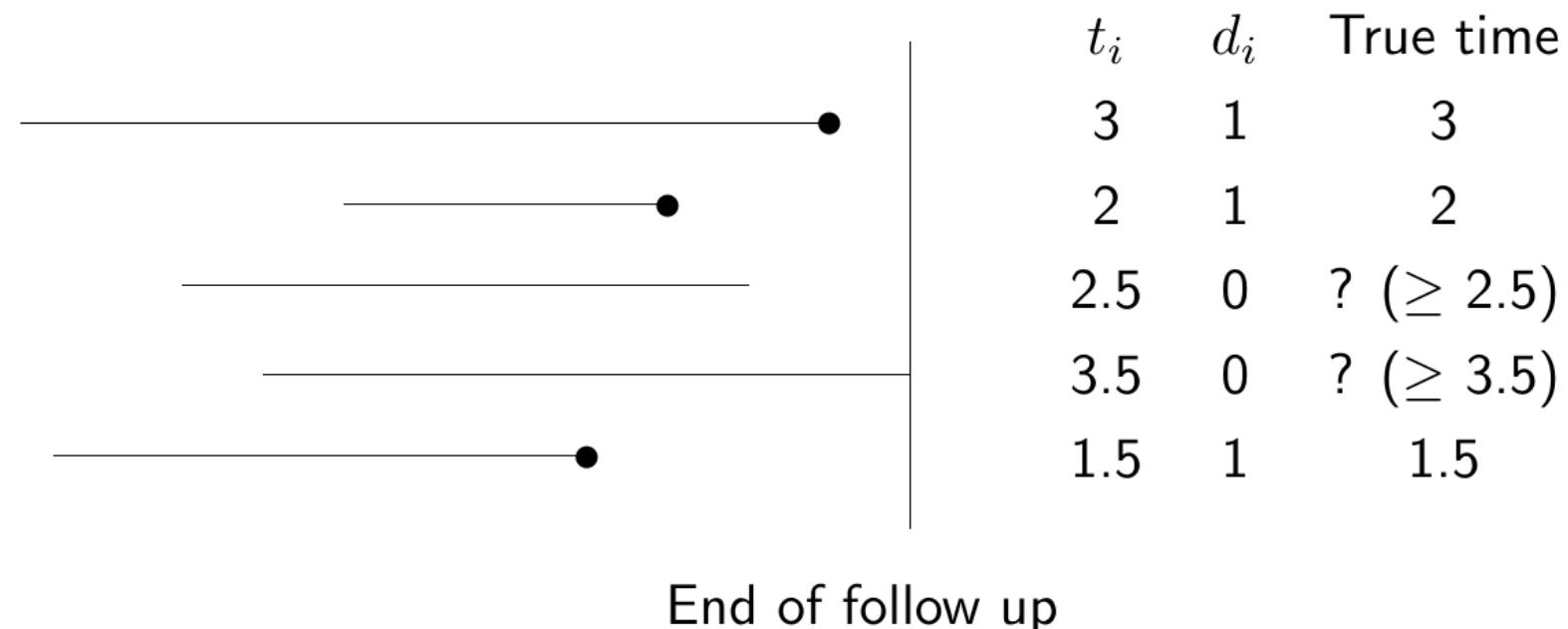
- 5 The cumulative hazard function is minus the log-survival:

$$H(t) = \int_0^t h(u) du = \int_0^t -\frac{S'(u)}{S(u)} du = -\log(S(t))$$

Censoring

In order to account for censoring, we need to specify the data collection

- $t_i > 0$: **observed** time to event for subject i
- $d_i = 0, 1$: an indicator of censoring
 - If $d_i = 0$, then the i -th subject did not experience the event. In this case, t_i is the **partially observed** time
 - If $d_i = 1$, then the observed time is the "true" (**fully observed**) one



Modelling

Broadly speaking, there are two wide “families” of survival models:

I Semi-parametric Survival Models (eg Cox Proportional Hazard, splines, ...)

- Model directly the hazard function $h(t)$
- Distribution of survival time unknown
- Less consistent with theoretical $S(t)$ (typically step function)
- + Does not rely on distributional assumptions
- + Baseline hazard not necessary for estimation of hazard ratio

Modelling

Broadly speaking, there are two wide “families” of survival models:

1 Semi-parametric Survival Models (eg Cox Proportional Hazard, splines, ...)

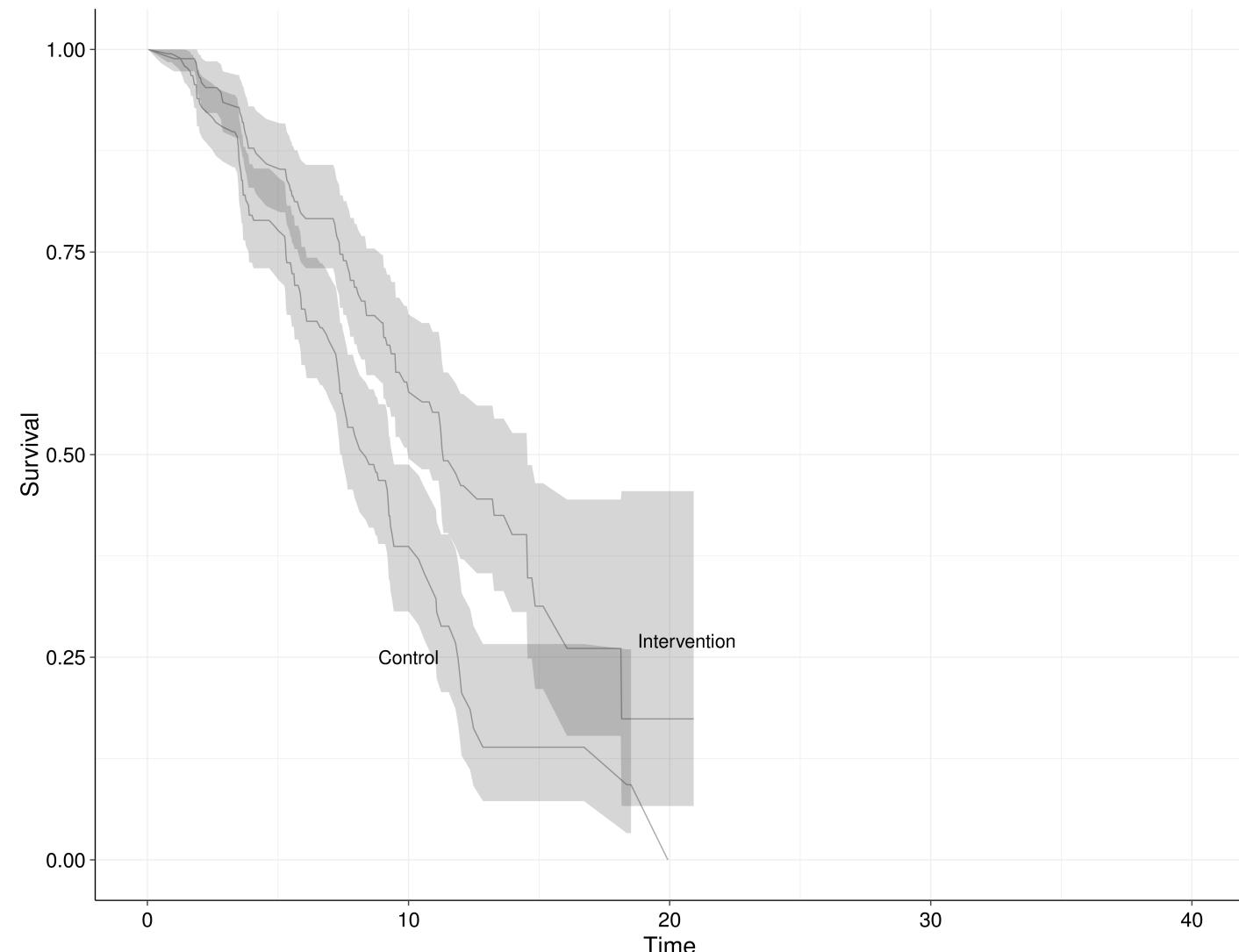
- Model directly the hazard function $h(t)$
- Distribution of survival time unknown
- Less consistent with theoretical $S(t)$ (typically step function)
- + Does not rely on distributional assumptions
- + Baseline hazard not necessary for estimation of hazard ratio

2 Parametric Survival Models (eg Weibull, Exponential, ...)

- Model directly the time-to-event t , using a suitable parametric distribution
- + Completely specified $h(t)$ and $S(t)$
- + More consistent with theoretical $S(t)$
- + Time-quantile prediction possible
- Assumption on underlying distribution

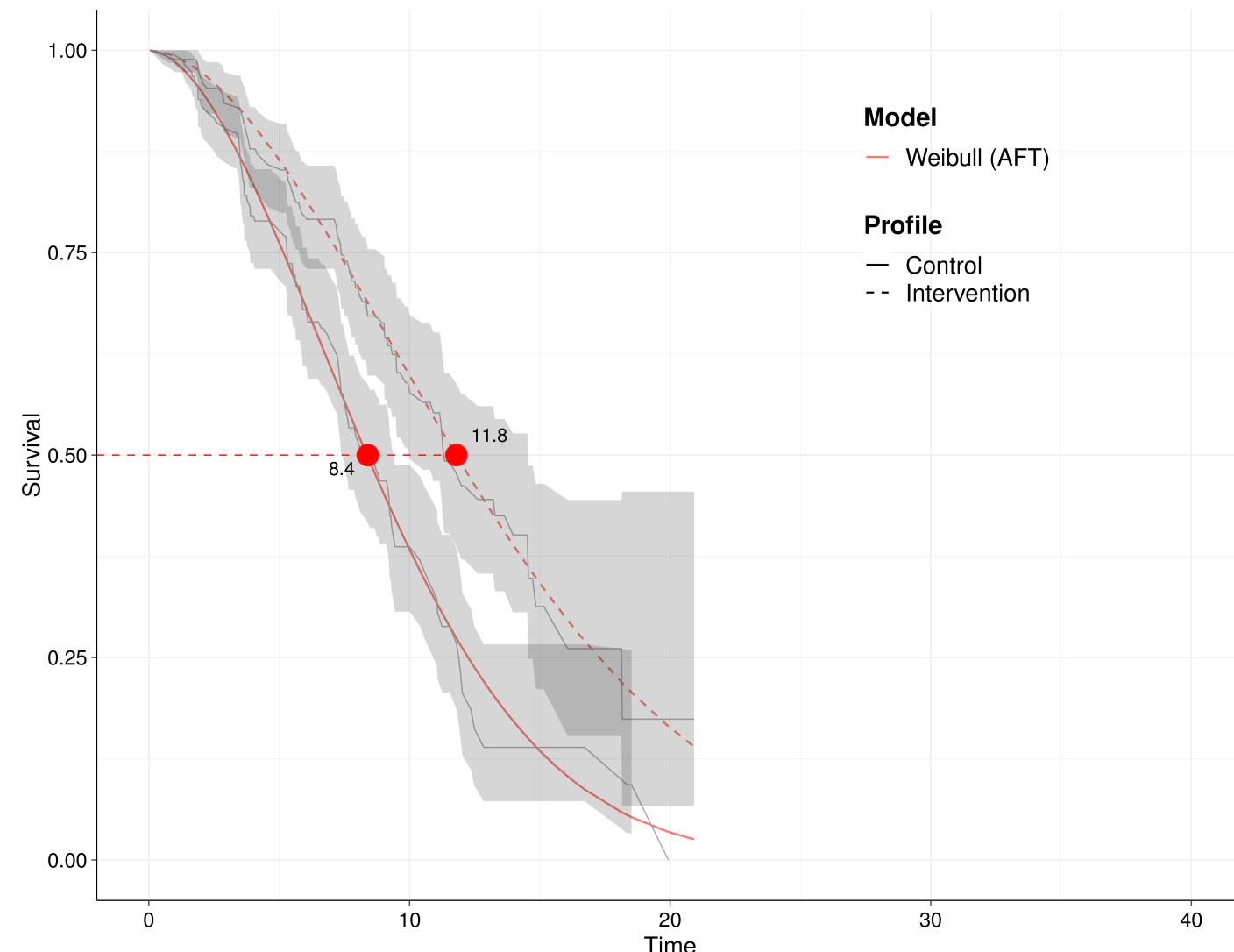
Survival analysis in HTA

Trial data – Kaplan-Meier curves



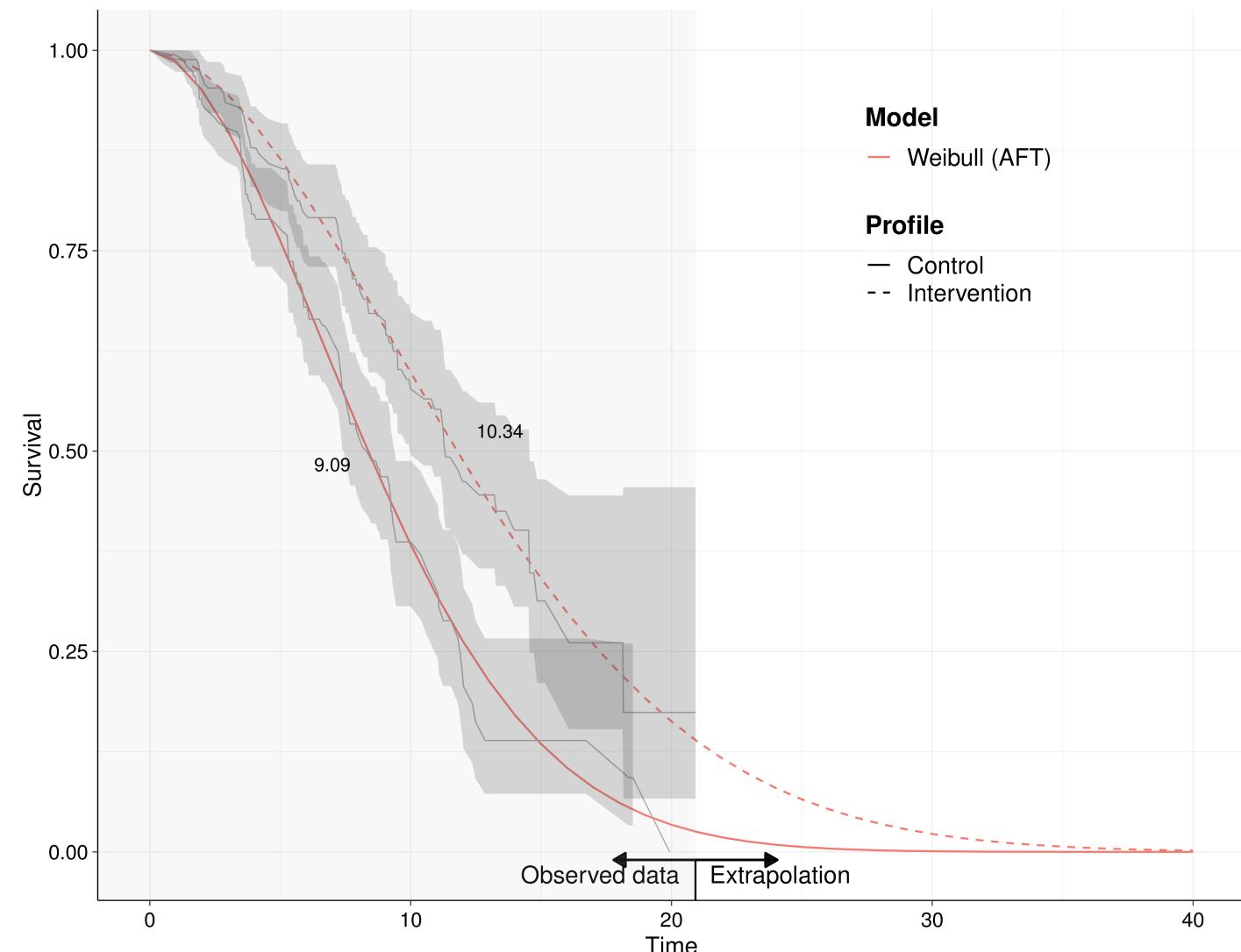
Survival analysis in HTA

Median time: $t : S(t) = 0.5$



Survival analysis in HTA

$$\text{Mean time: } \int_0^{\infty} S(t)dt$$



General structure

$$t \sim f(\mu(\mathbf{x}), \alpha(\mathbf{x})), \quad t \geq 0$$

- \mathbf{x} = vector of covariates (potentially influencing survival)
- $\mu(\mathbf{x})$ = **location** parameter
 - Scale or mean – usually main objective of the (biostats!) analysis
 - Typically depends on the covariates \mathbf{x}
- $\alpha(\mathbf{x})$ = **ancillary** parameters
 - Shape, variances, etc
 - May depend on \mathbf{x} , but often assume they don't (see NICE TSD 14)
- **NB:** $S(t)$ and $h(t)$ are functions of $\mu(\mathbf{x}), \alpha(\mathbf{x})$

General structure

$$t \sim f(\mu(\mathbf{x}), \alpha(\mathbf{x})), \quad t \geq 0$$

- \mathbf{x} = vector of covariates (potentially influencing survival)
- $\mu(\mathbf{x})$ = **location** parameter
 - Scale or mean – usually main objective of the (biostats!) analysis
 - Typically depends on the covariates \mathbf{x}
- $\alpha(\mathbf{x})$ = **ancillary** parameters
 - Shape, variances, etc
 - May depend on \mathbf{x} , but often assume they don't (see NICE TSD 14)
- **NB:** $S(t)$ and $h(t)$ are functions of $\mu(\mathbf{x}), \alpha(\mathbf{x})$
- Typically use generalised linear model

$$g(\mu_i) = \beta_0 + \sum_{j=1}^J \beta_j x_{ij} [+ \dots]$$

– since $t > 0$, usually, $g(\cdot) = \log$

- In a Bayesian setting, complete by putting suitable priors on β and α (more on this later...)

Survival analysis in HTA

For example... (see   survHE paper + NICE TSD 14)

Data model	Location parameter	Ancillary parameter
$t_i \sim \text{Exponential}(\mu_i)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	—
$t_i \sim \text{Weibull}(\mu_i, \alpha)$	Scale: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{logNormal}(\mu_i, \alpha)$	log-mean: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	log-sd: $\alpha \sim \text{Uniform}(0, 5)$
$t_i \sim \text{logLogistic}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gamma}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gompertz}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gen Gamma}(\mu_i, \alpha)$	Location: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	$\alpha = (\sigma, q)$ Scale: $\sigma \sim \text{Gamma}(0.1, 0.1)$ Shape: $q \sim \text{Normal}(0, 100)$
$t_i \sim \text{Gen F}(\mu_i, \alpha)$	Location: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	$\alpha = (\sigma, q, p)$ Scale: $\sigma \sim \text{Gamma}(0.1, 0.1)$ Shape(1): $p \log(p) \sim \text{Normal}(0, 0.5)$ Shape(2): $q \sim \text{Normal}(0, 2.5)$

A R package for survival analysis in HTA

Objective: Simplify and standardise commands to fit survival analysis

- Can do MLE + bootstrap to get (possibly rough-ish!) estimates from the joint distribution of the parameters
- Can also do Bayesian models to get (usually better!) estimates from the joint **posterior** distribution of the parameters
 - INLA: Super fast (comparable to MLE), but currently supports only a restricted range of models
 - MCMC: Slower, but more generalisable – survHE produces and saves the model code + data & initial values, so the user can customise them

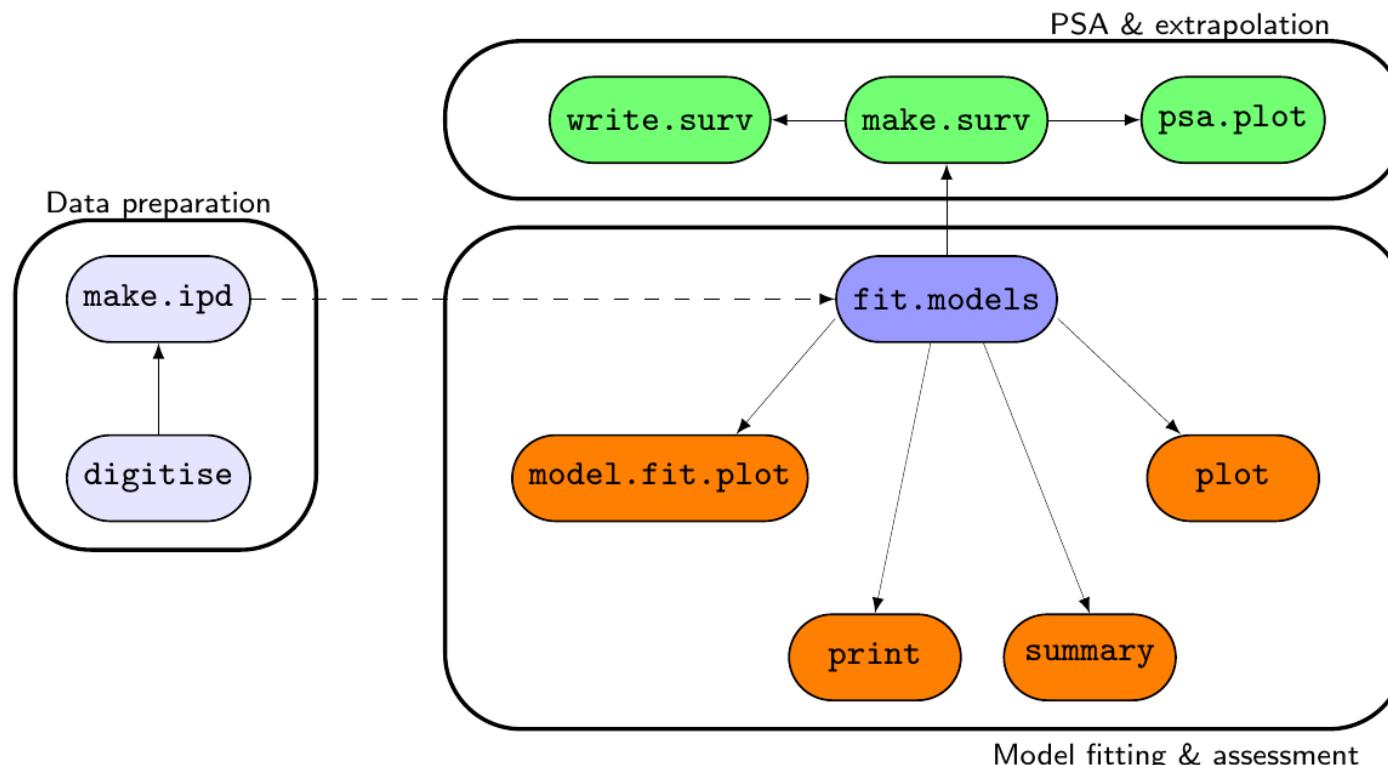
A R package for survival analysis in HTA

Objective: Simplify and standardise commands to fit survival analysis

- Can do MLE + bootstrap to get (possibly rough-ish!) estimates from the joint distribution of the parameters
- Can also do Bayesian models to get (usually better!) estimates from the joint **posterior** distribution of the parameters
 - INLA: Super fast (comparable to MLE), but currently supports only a restricted range of models
 - MCMC: Slower, but more generalisable – survHE produces and saves the model code + data & initial values, so the user can customise them
- Automatically produces specialised graphs
 - Survival curves + model fitting statistics (AIC, BIC, DIC)
- Can produce a full **PSA** characterisation of the parameters **and** the survival curves
 - These can be used directly in the economic model!

A R package for survival analysis in HTA

Objective: Simplify and standardise commands to fit survival analysis



 <https://github.com/giabaio/survHE>

 [survHE webpage](#)

Running the model

Exploring the object

Estimates (1)

Estimates (2)

Estimates (3)

```
> # Loads the package (can also use the GitHub development version
> # see http://www.statistica.it/gianluca/software/survhe/)
> library(survHE)
>
> # Defines the 'model formula'
> formula=Surv(TIME,EVENT)~1
>
> # Fits the model on the data for the Intervention group only using a bunch of distributions
> m.int=fit.models(formula,data=subset(data,treatment=="Intervention"),
+                   distr=c("exp","weibull","lnorm","llogis","gompertz","gengamma"))
>
> # Fits the model on the data for the Control group only using a bunch of distributions
> m.ctr=fit.models(formula,data=subset(data,treatment=="Comparator"),
+                   distr=c("exp","weibull","lnorm","llogis","gompertz","gengamma"))
```

Fitting parametric models in R with survHE

Running the model

Exploring the object

Estimates (1)

Estimates (2)

Estimates (3)

```
> # Explores the object 'm.int' to see what's inside...
> lapply(m.int,names)
```

```
$models
[1] "Exponential"    "Weibull (AFT)"   "log-Normal"      "log-Logistic"   "Gompertz"        "Gen. Gamma"
```

```
$model.fitting
[1] "aic" "bic" "dic"
```

```
$method
NULL
```

```
$misc
[1] "time2run"     "formula"       "data"          "model_name"    "km"
```

```
> # Shows model fitting statistics (eg AIC)
> m.int$model.fitting$aic
```

```
[1] 3031.740 3021.513 3032.696 3024.756 3027.801 3021.369
```

Fitting parametric models in R with survHE

Running the model

Exploring the object

Estimates (1)

Estimates (2)

Estimates (3)

```
> # Prints the model estimates for model 1 (Exponential)
> print(m.int,mod=1)
```

Model fit for the Exponential model, obtained using Flexsurvreg
(Maximum Likelihood Estimate). Running time: 0.024 seconds

mean	se	L95%	U95%
rate 0.00193397	0.000133775	0.00168877	0.00221477

Model fitting summaries

Akaike Information Criterion (AIC)....: 3031.740
Bayesian Information Criterion (BIC)...: 3035.508

```
> # Prints the model estimates for model 2 (Weibull)
> print(m.int,mod=2)
```

Model fit for the Weibull AF model, obtained using Flexsurvreg
(Maximum Likelihood Estimate). Running time: 0.011 second

	mean	se	L95%	U95%
shape	1.22191	0.0668217	1.09772	1.36016
scale	511.30594	28.9447135	457.60930	571.30343

Model fitting summaries

Akaike Information Criterion (AIC)....: 3021.513
Bayesian Information Criterion (BIC)...: 3029.050

[Running the model](#)[Exploring the object](#)[Estimates \(1\)](#)[Estimates \(2\)](#)[Estimates \(3\)](#)

```
> # Prints the model estimates for model 3 (log-Normal), using 3 digits precision  
> print(m.int,mod=3,digits=3)
```

Model fit for the log-Normal model, obtained using Flexsurvreg
(Maximum Likelihood Estimate). Running time: 0.009 seconds

	mean	se	L95%	U95%
meanlog	5.83	0.0693	5.7	5.97
sdlog	1.11	0.0550	1.0	1.22

Model fitting summaries
Akaike Information Criterion (AIC)....: 3032.696
Bayesian Information Criterion (BIC)...: 3040.232

[Running the model](#)[Exploring the object](#)[Estimates \(1\)](#)[Estimates \(2\)](#)[Estimates \(3\)](#)

```
> # Can also show the output printout from the original inferential engine ('flexsurv' in this case...)
> print(m.int,mod=3,original=TRUE)
```

Call:

```
flexsurvreg(formula = Surv(TIME, EVENT) ~ 1, data = data, dist = "lnorm")
```

Estimates:

	est	L95%	U95%	se
meanlog	5.8317202	5.6958328	5.9676076	0.0693316
sdlog	1.1059558	1.0032757	1.2191447	0.0549826

N = 320, Events: 209, Censored: 111

Total time at risk: 108068

Log-likelihood = -1514.348, df = 2

AIC = 3032.696

Fitting parametric models in R with survHE

Plotting (1)

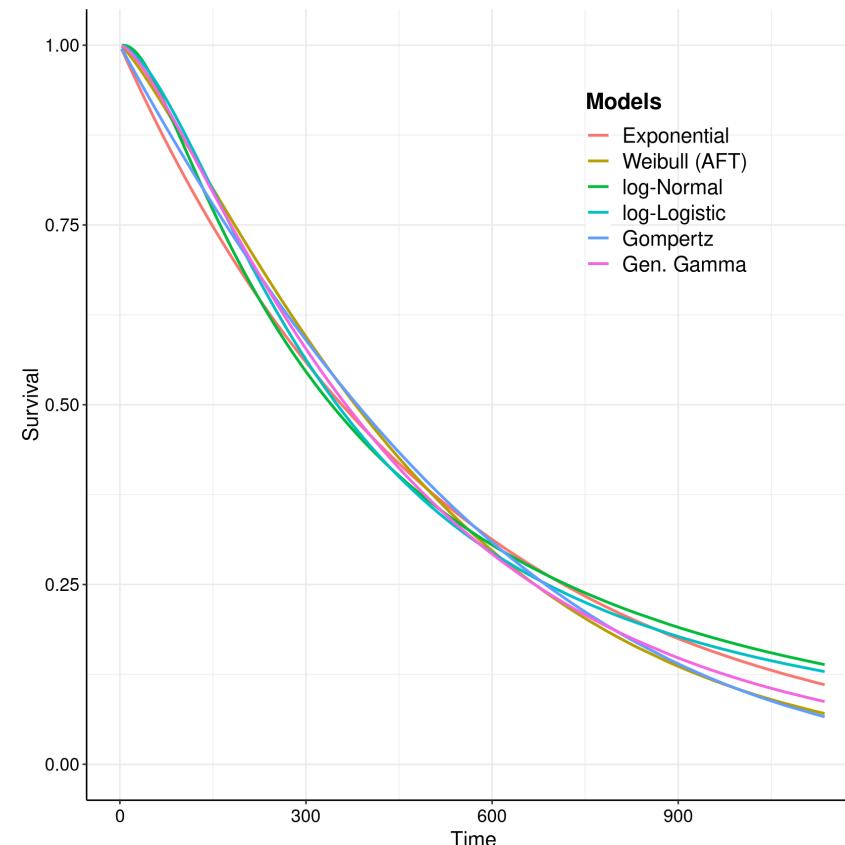
Plotting (2)

Plotting (3)

Model fit

...and more!

```
> plot(m.int) # Basic plot function (based on 'ggplot2')
```



Fitting parametric models in R with survHE

Plotting (1)

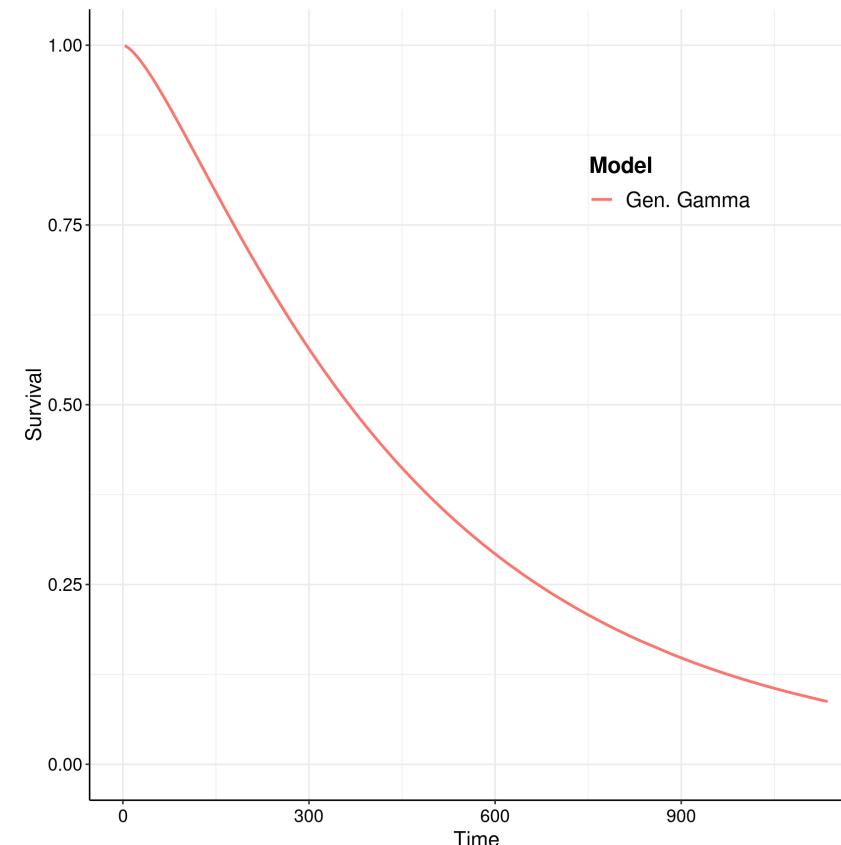
Plotting (2)

Plotting (3)

Model fit

...and more!

```
> plot(m.int,mods=6) # Selects only the 6th model (Generalised Gamma)
```



Fitting parametric models in R with survHE

Plotting (1)

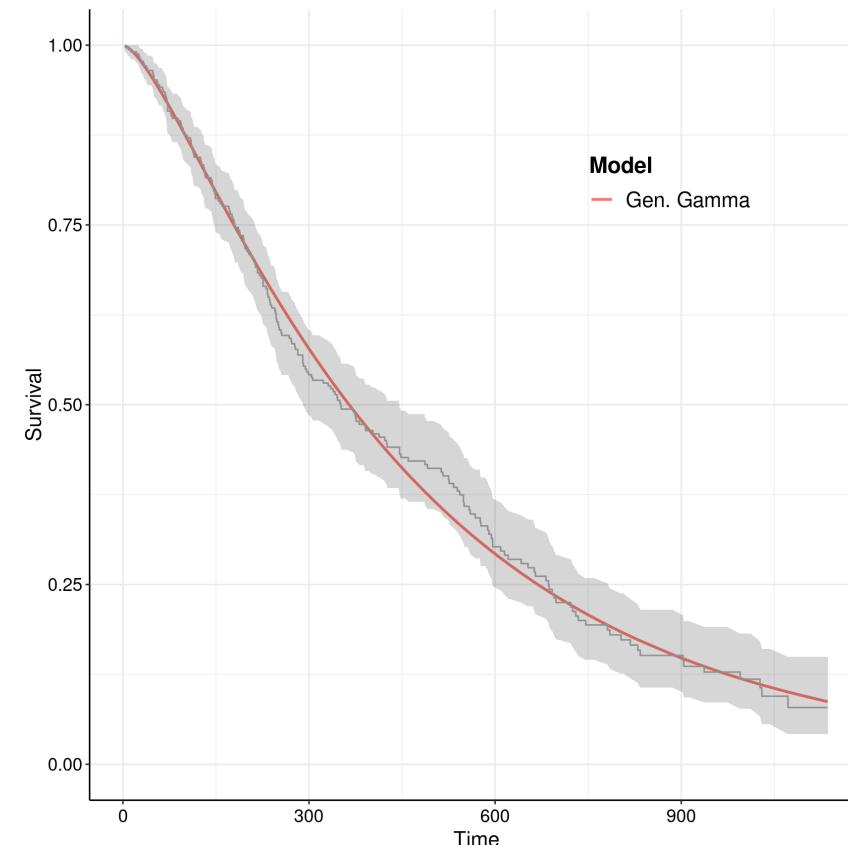
Plotting (2)

Plotting (3)

Model fit

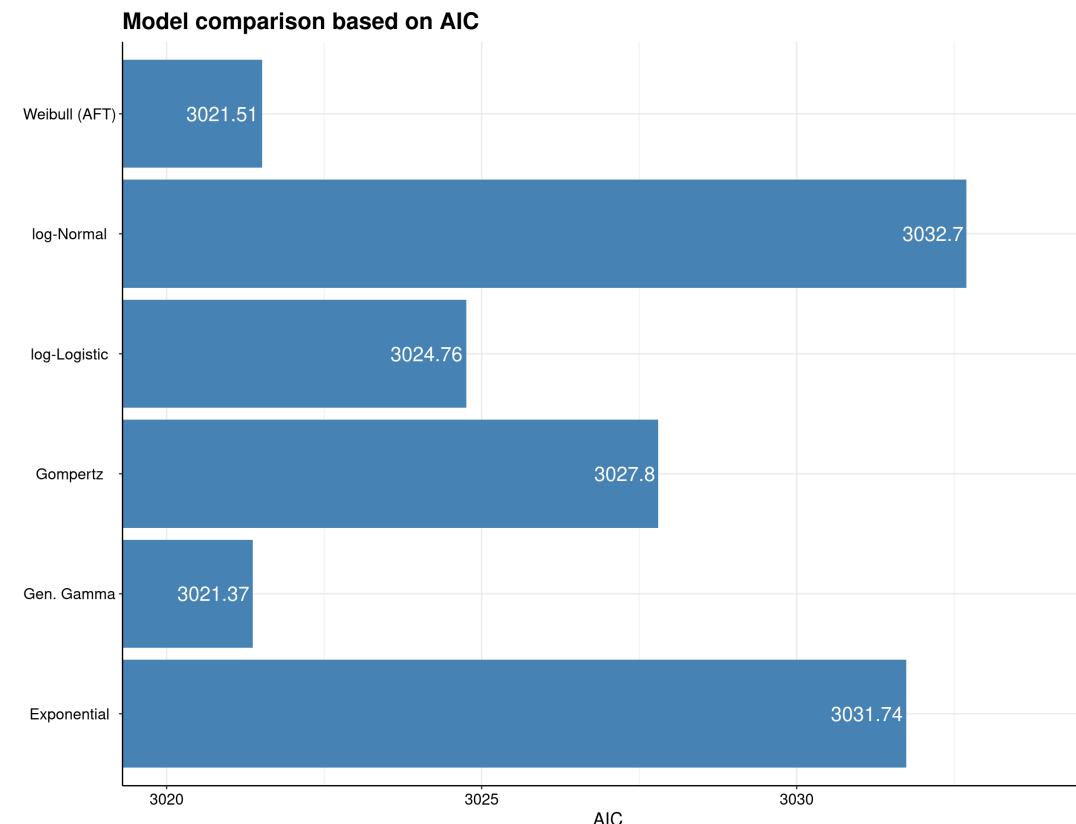
...and more!

```
> plot(m.int,mods=6,add.km=TRUE) # Selects only the 6th model (Generalised Gamma) and adds Kaplan-Maier curve
```



[Plotting \(1\)](#)[Plotting \(2\)](#)[Plotting \(3\)](#)[Model fit](#)[...and more!](#)

```
> model.fit.plot(m.int)
```



[Plotting \(1\)](#)[Plotting \(2\)](#)[Plotting \(3\)](#)[Model fit](#)[...and more!](#)

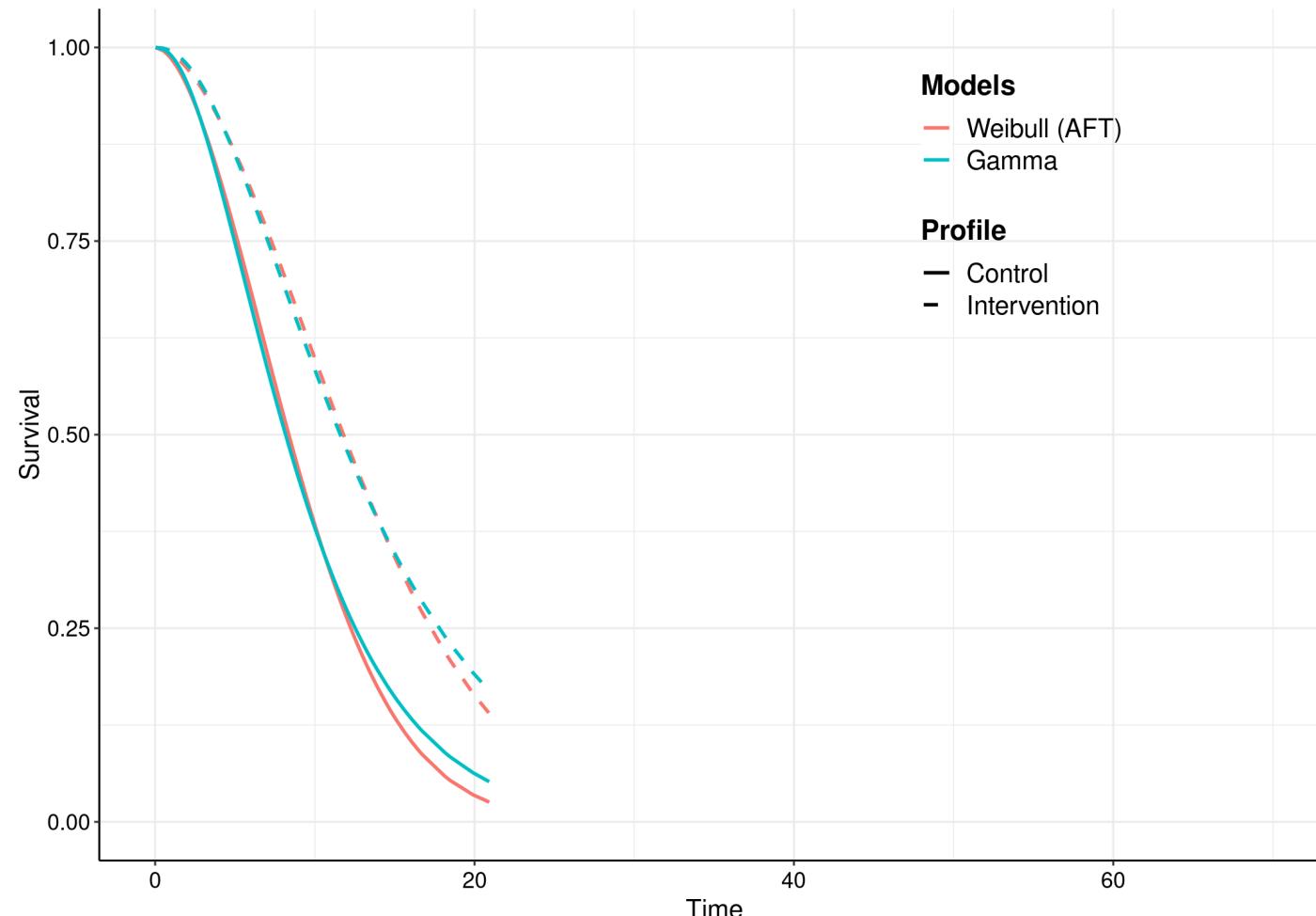
Many more options are shown and explained in the survHE  paper and  webpage

A recipe for disaster?...

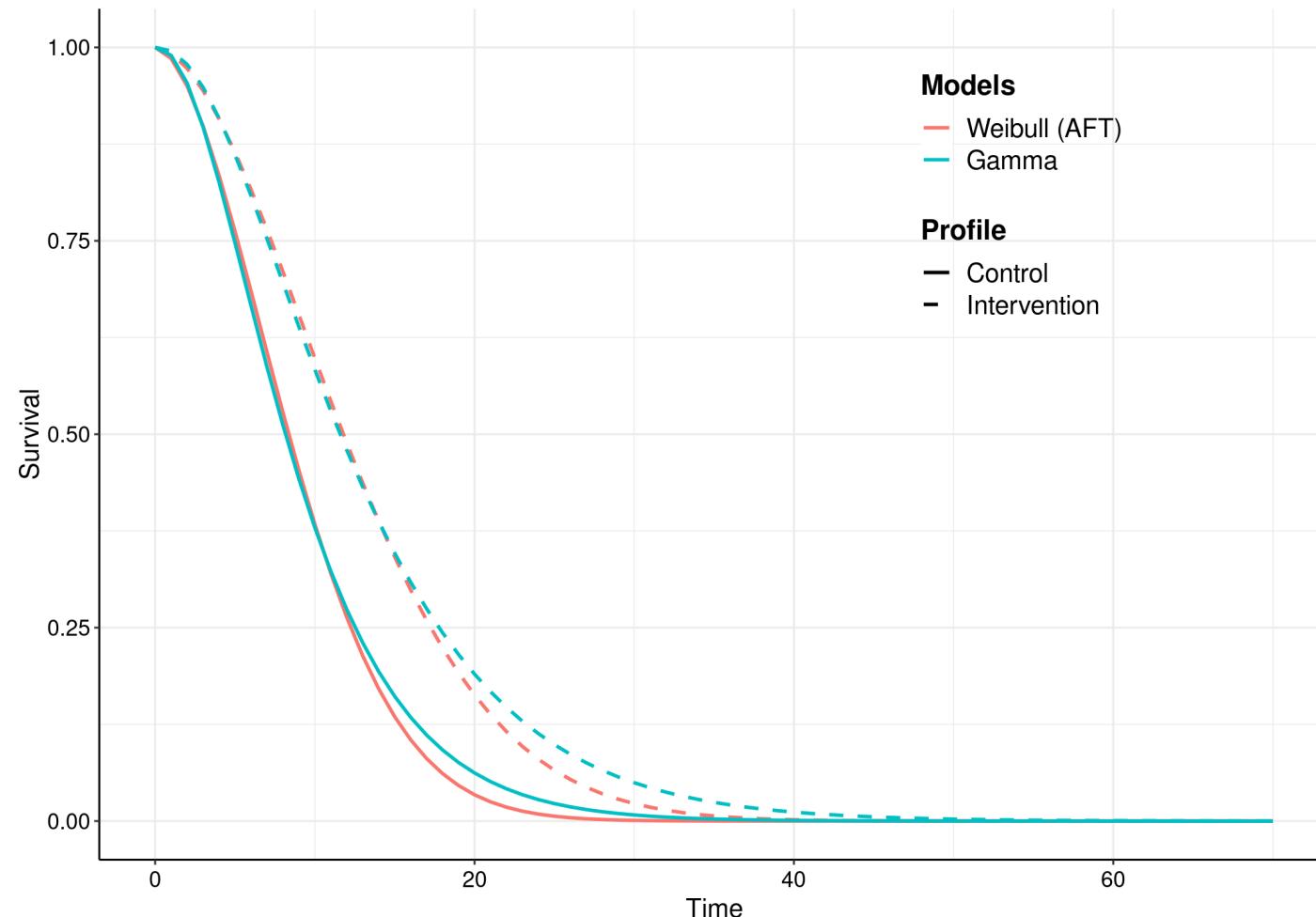


British diners would be locked up for crimes such as dipping pizza in ketchup, if despairing Italians chefs could police their food

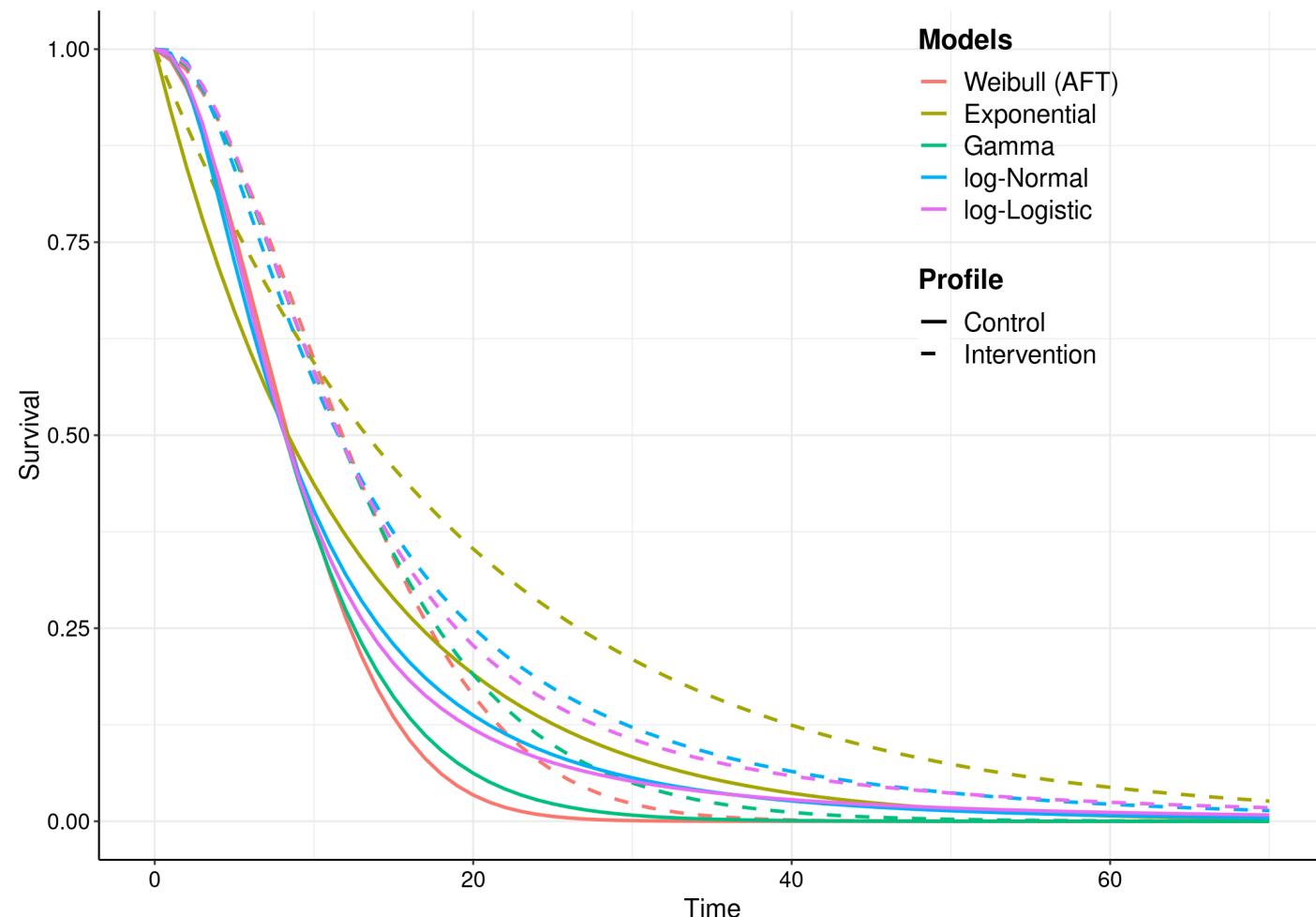
A recipe for disaster?...



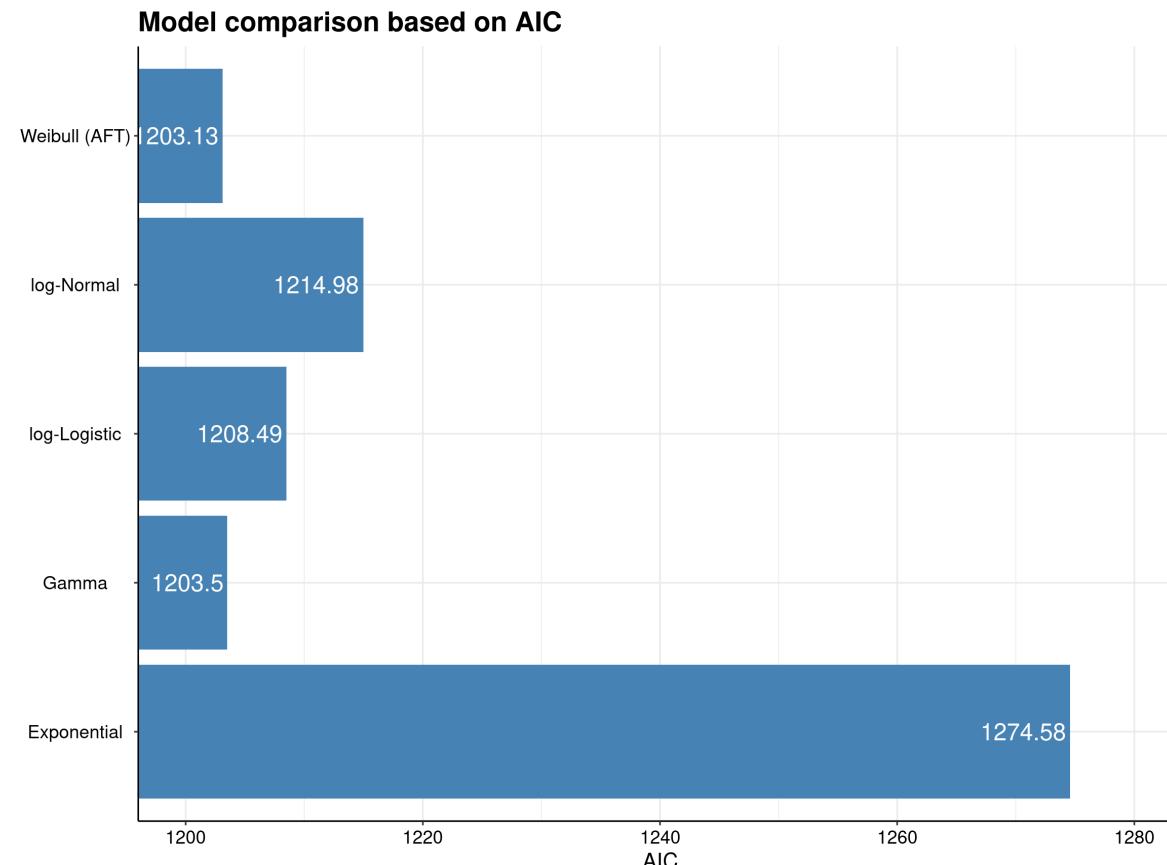
A recipe for disaster?...



A recipe for disaster?...



A recipe for disaster?...



- NB: Any *IC can only tell us about model fit for the observed data!
- Extrapolation (like missing data) is based on (virtually) untestable assumptions

Extrapolation

A recipe for disaster?...

Survival model inputs

Trial 1: Some outcome of interest, parametric survival models

variable	Coef.	Std. Err.	z	P>z	95% Conf. Interval
Treatment effect	0.47	0.13	0.01	0.00	
Intercept	3.29	0.08	36.67	0.00	

Exponential model

time in weeks

Trial 2 Data: OS KM data & Weibull regression for parameters

t	Active		t	Control		Regression analysis		
	Ln(t)	Ln(-Ln(S(t)))		Ln(t)	Ln(-Ln(S(t)))	Active	Control	
0	0	1	0	0.49	0.99	-5.18	-0.71	-5.24
1	0.52	0.99	1	0.85	0.99	0.66	-4.48	-0.16
2	1.93	0.99	1	1.44	0.96	0.80	-4.07	0.37
3	2.23	0.96	1	1.44	0.96	0.95	-3.78	0.37
3	2.59	0.96	3	2.59	0.97	1.13	-3.56	0.95
3	3.08	0.97	3	2.79	0.97	1.26	-3.37	1.02
4	3.61	0.97	3	2.89	0.96	1.47	-3.22	1.06
4	4.36	0.96	3	3.38	0.96	1.50	-3.08	1.22
4	4.49	0.96	4	3.57	0.95	1.57	-2.96	1.27
5	4.79	0.95	4	4.10	0.95	1.59	-2.85	1.41
5	4.92	0.94	4	4.13	0.94	1.66	-2.75	1.42
5	5.28	0.94	4	4.23	0.94	1.71	-2.66	1.44
6	5.51	0.93	4	4.23	0.93	1.71	-2.58	1.44
6	5.51	0.93	4	4.23	0.93	1.79	-2.50	1.44
6	5.97	0.92	5	4.69	0.92	1.96	-2.45	1.55
7	7.08	0.92	5	5.31	0.92	2.04	-2.36	1.67
7	7.70	0.91	5	5.64	0.91	2.06	-2.30	1.73
8	7.84	0.90	6	5.87	0.90	2.12	-2.24	1.77
8	8.36	0.90	7	6.56	0.90	2.14	-2.18	1.88
8	8.46	0.89	7	6.79	0.89	2.18	-2.13	1.91
9	8.89	0.89	7	7.05	0.89	2.23	-2.08	1.95
9	9.34	0.88	8	7.51	0.88	2.24	-2.03	2.02
9	9.41	0.88	8	7.84	0.88	2.28	-1.98	2.06
10	9.77	0.87	8	8.10	0.87	2.28	-1.93	2.09
10	9.80	0.87	8	8.30	0.87	2.31	-1.89	2.12
10	10.10	0.86	8	8.39	0.86	2.32	-1.85	2.13
10	10.20	0.85	8	8.46	0.86	2.36	-1.80	2.14
11	10.59	0.85	9	8.62	0.85	2.38	-1.77	2.15
11	10.79	0.84	9	8.66	0.85	2.44	-1.73	2.16
12	11.51	0.84	9	9.15	0.84	2.50	-1.69	2.21
12	12.16	0.83	9	9.15	0.84	2.60	-1.65	2.21
13	13.41	0.83	9	9.31	0.83	2.71	-1.62	2.23
15	15.02	0.82	9	9.31	0.83	2.73	-1.58	2.23
15	15.28	0.81	9	9.34	0.82	2.74	-1.55	2.23
16	15.54	0.81	10	9.64	0.81	2.77	-1.52	2.27
16	16.00	0.80	10	9.80	0.81	2.78	-1.49	2.26
16	16.20	0.80	10	9.97	0.80	2.63	-1.46	2.30
17	16.95	0.79	10	10.07	0.80	2.84	-1.43	2.31
17	17.18	0.79	10	10.16	0.79	2.85	-1.40	2.32
17	17.26	0.78	10	10.20	0.79	2.87	-1.37	2.32
18	17.70	0.78	10	10.26	0.78	2.87	-1.34	2.33
18	17.70	0.77	10	10.30	0.78	2.89	-1.31	2.33
18	17.93	0.76	11	10.52	0.77	2.90	-1.29	2.35

Weibull model

time in weeks

variable	Coef.	Std. Err.	z	P>z
Treatment effect	0.35	0.09	0.89	0.00
Intercept	4.02	0.03	48.50	0.00
ln_p	0.67	0.05	9.01	0.00

variance covariance matrix

Trt_eff	Int	ln_p
0.01	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

Used to sample lambda and gamma for both treatment arms and the hazard ratio in PSA

Cholesky decomposition Std normal Correlated parameters matrix

variates	parameters
-0.0654	0.33
0.00	4.02
0.00	0.63

Used to sample lambda for both treatment arms and the hazard ratio in PSA

Cholesky decomposition Std normal Correlated parameters matrix

variates	parameters
0.08 -0.05	0.02
0.00 0.05	0.1160
0.00 0.02	-0.8316

Extrapolation using survHE

Code

Processing (1)

Processing (2)

Plotting (1)

Plotting (2)

```
> # Creates an object 'extr' with estimated values for the survival curve
> # Uses the output from the 'survHE' object named 'm.int'
> # Considers the 1st distribution (Exponential)
> # Makes extrapolation for times 0-5000
> # Does only 1 simulation
> extr=make.surv(m.int,mod=1,t=seq(0,5000),nsim=1)
```

Extrapolation using survHE

Code **Processing (1)** Processing (2) Plotting (1) Plotting (2)

```
> # inspects elements of the object 'extr'  
> names(extr)
```

```
[1] "S"      "sim"     "nsim"    "mat"     "des.mat" "times"
```

```
> # number of simulated curves (one per treatment arm)  
> length(extr$S)
```

```
[1] 1
```

Extrapolation using survHE

Code Processing (1) **Processing (2)** Plotting (1) Plotting (2)

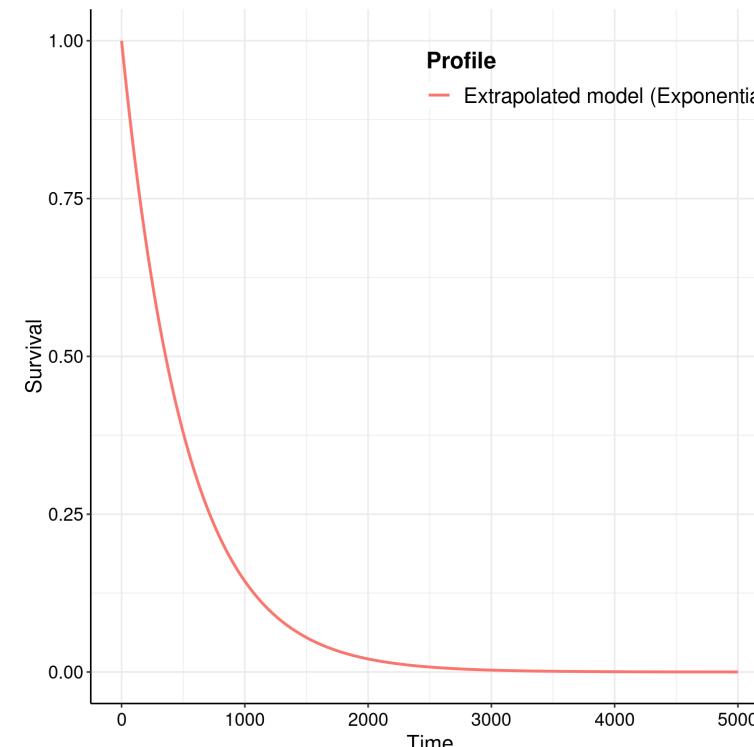
```
> # simulated survival curve for the first of the nsim (=1 in this case...) simulations  
> extr$S
```

```
[[1]]  
# A tibble: 5,001 × 2  
  t      S  
  <int> <dbl>  
1 0     1  
2 1     0.998  
3 2     0.996  
4 3     0.994  
5 4     0.992  
6 5     0.990  
7 6     0.988  
8 7     0.987  
9 8     0.985  
10 9    0.983  
# ... with 4,991 more rows
```

Extrapolation using survHE

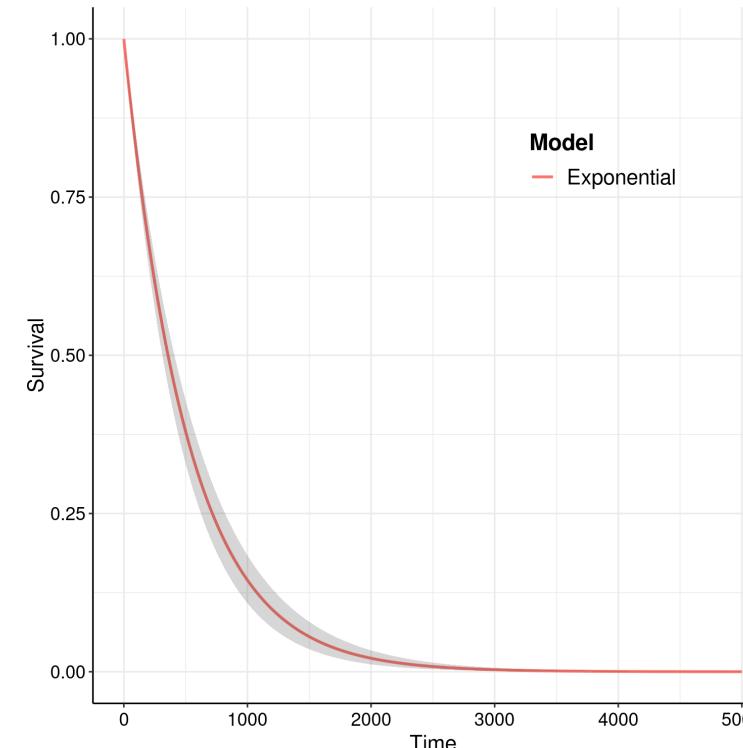
Code Processing (1) Processing (2) **Plotting (1)** Plotting (2)

```
> # Uses a specialised function to plot the extrapolated survival curve  
> psa.plot(extr,labs="Extrapolated model (Exponential)")
```



[Code](#)[Processing \(1\)](#)[Processing \(2\)](#)[Plotting \(1\)](#)[Plotting \(2\)](#)

```
> # Or directly using the 'plot' method - this time using 1000 simulations from the parameter distributions...
> plot(m.int,nsim=1000,mods=1,t=seq(0,5000))
```



... To be or not to be (Bayesian)?...

- For more complex models, MLE-based estimates may fail to converge
 - This may be an issue for multi-parameter models, where limited data (not compounded by relevant prior information) are not enough to fit all the model parameters
 - **NB:** you would normally need to fit more complex models for cases where the survival curves are "strange" and so the usual parametric models fail to provide sufficient fit
- When there is strong correlation among the survival parameters, the results of the uncertainty analysis may be (strongly) biased under a more simplistic frequentist model
 - This matters most in health economics, because this bias carries over the economic modelling, optimal decision making and assessment of the impact of parametric uncertainty!
 - A full Bayesian approach propagates directly correlation and uncertainty in the model parameters through to the survival curves and the economic model

... To be or not to be (Bayesian)?...

Model fit for the Generalised F model, obtained using Flexsurvreg
(Maximum Likelihood Estimate). Running time: 1.157 seconds

	mean	se	L95%	U95%
mu	2.29139696	0.0798508	2.13489e+00	2.44790e+00
sigma	0.58729598	0.0725044	4.61076e-01	7.48069e-01
Q	0.84874994	0.2506424	3.57500e-01	1.34000e+00
P	0.00268265	0.0902210	6.33197e-32	1.13655e+26
as.factor(arm)1	0.34645851	0.0877892	1.74395e-01	5.18522e-01

Model fit for the Generalised F model, obtained using Stan
(Bayesian inference via Hamiltonian Monte Carlo). Running time: 26.692 seconds

	mean	se	L95%	U95%
mu	2.256760	0.3455163	1.1897086	3.0865904
sigma	0.507861	0.0762112	0.3608566	0.6582047
Q	0.700062	0.3358360	0.0786118	1.3880582
P	1.131968	0.5837460	0.3908284	2.634276
as.factor(arm)1	0.345516	0.0865904	0.1745665	0.5176818

... To be or not to be (Bayesian)?...

- For more complex models, MLE-based estimates may fail to converge
 - This may be an issue for multi-parameter models, where limited data (not compounded by relevant prior information) are not enough to fit all the model parameters
 - NB: you would normally need to fit more complex models for cases where the survival curves are "strange" and so the usual parametric models fail to provide sufficient fit
- When there is strong correlation among the survival parameters, the results of the uncertainty analysis may be (strongly) biased under a more simplistic frequentist model
 - This matters most in health economics, because this bias carries over the economic modelling, optimal decision making and assessment of the impact of parametric uncertainty!
 - **A full Bayesian approach propagates directly correlation and uncertainty in the model parameters through to the survival curves and the economic model**

- In theory, coding up a survival model in standard Bayesian software (eg BUGS or JAGS) is not that complicated
 - **NB:** although they differ in how they take care of censoring, so some care is needed!
- **BUT:** Gibbs sampling can struggle with survival models
 - Compilation and running time can be rather long
 - Because the main outcome t has missing values in the data (the censored times), it is generally useful to set initial values for t , in a clever way to avoid problems in running the MCMC
 - Convergence may be difficult to reach even with relatively simple models (eg Weibull Proportional Hazard)

- In theory, coding up a survival model in standard Bayesian software (eg BUGS or JAGS) is not that complicated
 - **NB:** although they differ in how they take care of censoring, so some care is needed!
- **BUT:** Gibbs sampling can struggle with survival models
 - Compilation and running time can be rather long
 - Because the main outcome t has missing values in the data (the censored times), it is generally useful to set initial values for t , in a clever way to avoid problems in running the MCMC
 - Convergence may be difficult to reach even with relatively simple models (eg Weibull Proportional Hazard)
- survHE uses alternative modes of Bayesian inference to overcome/limit these issues
 -  **Integrated Nested Laplace Approximation (INLA)**
 - Very fast and accurate, but at present can only run a limited number of survival models
 -  **Hamiltonian Monte Carlo (HMC)**
 - MCMC algorithm, slightly cleverer than Gibbs sampling (for some models...)
 - **Very efficient** for survival analysis
 - Can implement virtually *any* model – rstan allows easy-ish building blocks to define new sampling distributions (accounting for censoring etc...)

Bayesian survival analysis using survHE

Data model	Location parameter	Ancillary parameter
$t_i \sim \text{Exponential}(\mu_i)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	—
$t_i \sim \text{Weibull}(\mu_i, \alpha)$	Scale: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{logNormal}(\mu_i, \alpha)$	log-mean: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	log-sd: $\alpha \sim \text{Uniform}(0, 5)$
$t_i \sim \text{logLogistic}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gamma}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gompertz}(\mu_i, \alpha)$	Rate: $\mu_i = \exp\left(\beta_0 + \sum_{j=1}^J \beta_j x_{ij}\right)$	Shape: $\alpha \sim \text{Gamma}(0.1, 0.1)$
$t_i \sim \text{Gen Gamma}(\mu_i, \boldsymbol{\alpha})$	Location: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	$\boldsymbol{\alpha} = (\sigma, q)$ Scale: $\sigma \sim \text{Gamma}(0.1, 0.1)$ Shape: $q \sim \text{Normal}(0, 100)$
$t_i \sim \text{Gen F}(\mu_i, \boldsymbol{\alpha})$	Location: $\mu_i = \beta_0 + \sum_{j=1}^J \beta_j x_{ij}$	$\boldsymbol{\alpha} = (\sigma, q, p)$ Scale: $\sigma \sim \text{Gamma}(0.1, 0.1)$ Shape(1): $\log(p) \sim \text{Normal}(0, 0.5)$ Shape(2): $q \sim \text{Normal}(0, 2.5)$

with $\boldsymbol{\beta} = \beta_0, \dots, \beta_J \stackrel{iid}{\sim} \text{Normal}(0, v)$, for suitable v

Running the model

Checks Output

```
> # Calls the 'survHE' function 'fit.models' to run the model
> # 1. using HMC as inferential engine (with the option: 'method="hmc"'')
> # 2. specifying a Weibull distribution for the data ('distr="weibull"'')
> m.hmc=fit.models(Surv(TIME,EVENT)~as.factor(treatment),
+                     data=dat,
+                     distr="weibull",
+                     method="hmc"
+ )
```

Bayesian survival analysis using survHE

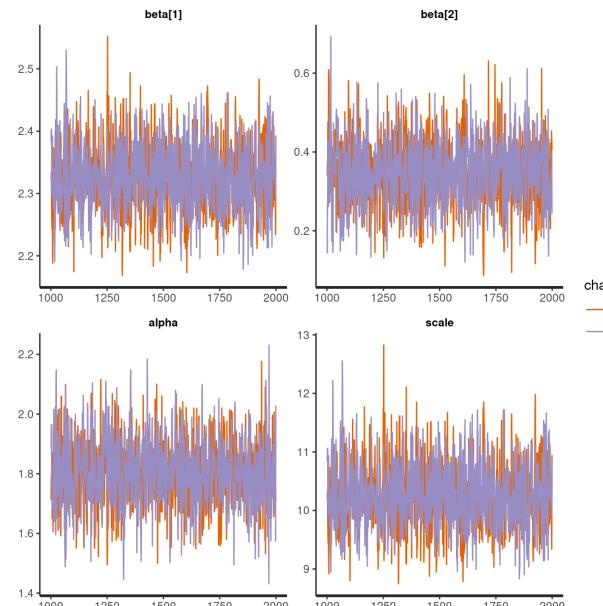
HMC

Running the model

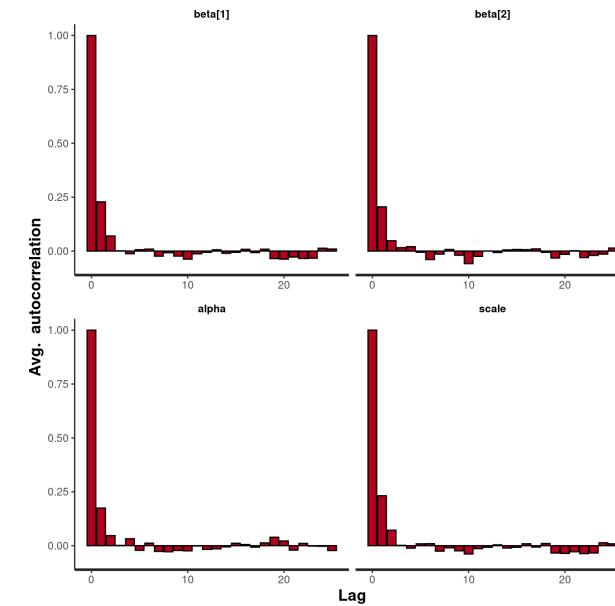
Checks

Output

```
> # Check convergence of the model using standard 'rstan' plots  
> rstan::traceplot(m.hmc$models[[1]])
```



```
> # Check convergence of the model using standard 'rstan' plots  
> rstan::stan_ac(m.hmc$models[[1]])
```



Bayesian survival analysis using survHE

HMC

Running the model

Checks

Output

```
> # Shows the parameter estimates in the formatting of  
> print(m.hmc,digits=4)
```

Model fit for the Weibull AF model, obtained using Stan (Bayesian Hamiltonian Monte Carlo). Running time: 2.4915 seconds

	mean	se	L95%	U95%		mean	se_mean	sd	2.5%	25%	50%
shape	1.8031	0.11252	1.5800	2.0240	beta[1]	2.33	0.00	0.05	2.22	2.29	2.32
scale	10.2544	0.56387	9.2307	11.4348	beta[2]	0.35	0.00	0.08	0.19	0.29	0.35
as.factor(treatment)Intervention	0.3488	0.08335	0.1854	0.5107	alpha	1.80	0.00	0.11	1.58	1.73	1.80

Model fitting summaries

Akaike Information Criterion (AIC)....: 1205.155
Bayesian Information Criterion (BIC)...: 1220.777
Deviance Information Criterion (DIC)...: 1202.951

```
> # ...or in 'rstan' original formatting (adding the c  
> print(m.hmc,digits=2, original=TRUE)
```

Inference for Stan model: WeibullAF.
Stan chain (chain 1) started at step 1
Sampling for Stan chain (chain 1) took 2.4915 seconds
Chain 1 finished at step 2000
2000 chains were used to calculate the posterior summary statistics
Chain 1 took 2.4915 seconds
Total sampling time was 2.4915 seconds
Total execution time was 2.4915 seconds

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%
shape	2.33	0.00	0.05	2.22	2.29	2.32	2.36	2.40
scale	0.35	0.00	0.08	0.19	0.29	0.35	0.40	0.45
alpha	1.80	0.00	0.11	1.58	1.73	1.80	1.93	2.00
lp__	-600.26	0.04	1.20	-603.40	-600.78	-599.94	-599.70	-599.50

Samples were drawn using NUTS(diag_e) at Fri Jan 7 15:45:20 2017
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (Rhat>1).

Running the model

Comparing the results

Graphical comparison

```
> # Calls the 'survHE' function 'fit.models' to run the model
> # 1. using INLA as inferential engine (with the option: 'method="inla"'')
> # 2. specifying a Weibull distribution for the data ('distr="weibull"'')
> m.inla=fit.models(Surv(TIME,EVENT)~as.factor(treatment),
+                     data=dat,distr="weibull",
+                     method="inla"
+ )
```

Bayesian survival analysis using survHE

INLA

Running the model

Comparing the results

Graphical comparison

```
> # INLA output with parameters estimates  
> print(m.inla,digits=4)
```

```
> # HMC output with parameters estimates  
> print(m.hmc,digits=4)
```

Model fit for the Weibull AF model, obtained using INLA (Bayesian Model Inference via the Integrated Nested Laplace Approximation). Running time: 1.809 seconds

	mean	se	L95%	U95%
shape	1.7656	0.11423	1.5506	1.9869
scale	10.2737	0.60318	9.1417	11.5265
as.factor(treatment)Intervention	0.3491	0.08526	0.1886	0.5139

Model fitting summaries

Akaike Information Criterion (AIC)....: 1205.370
Bayesian Information Criterion (BIC)...: 1220.991
Deviance Information Criterion (DIC)...: 1206.533

	mean	se	L95%	U95%
shape	1.8031	0.11252	1.5800	1.8031
scale	10.2544	0.56387	9.2307	10.2544
as.factor(treatment)Intervention	0.3488	0.08335	0.1854	0.3488

Model fitting summaries

Akaike Information Criterion (AIC)....: 1205.155
Bayesian Information Criterion (BIC)...: 1220.777
Deviance Information Criterion (DIC)...: 1202.951

Bayesian survival analysis using survHE

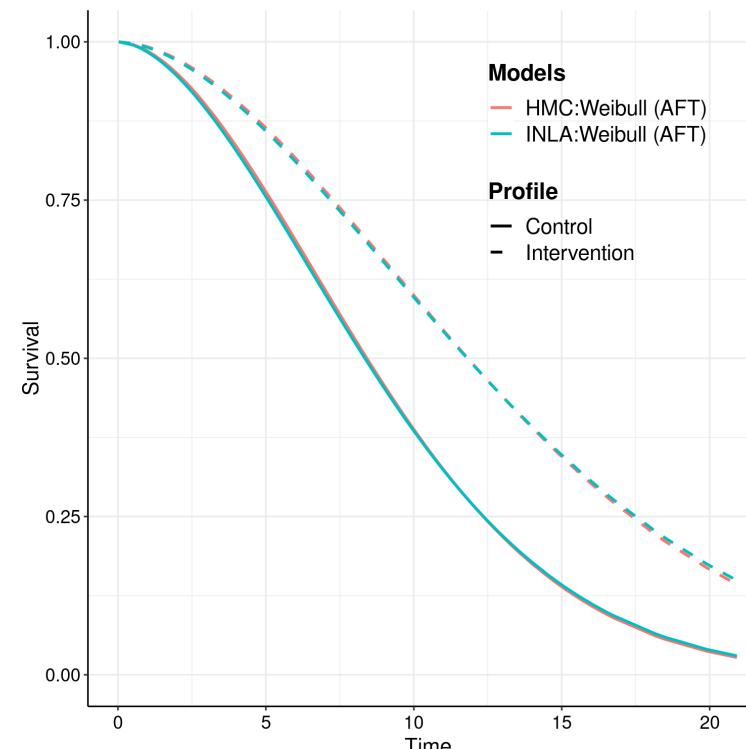
INLA

Running the model

Comparing the results

Graphical comparison

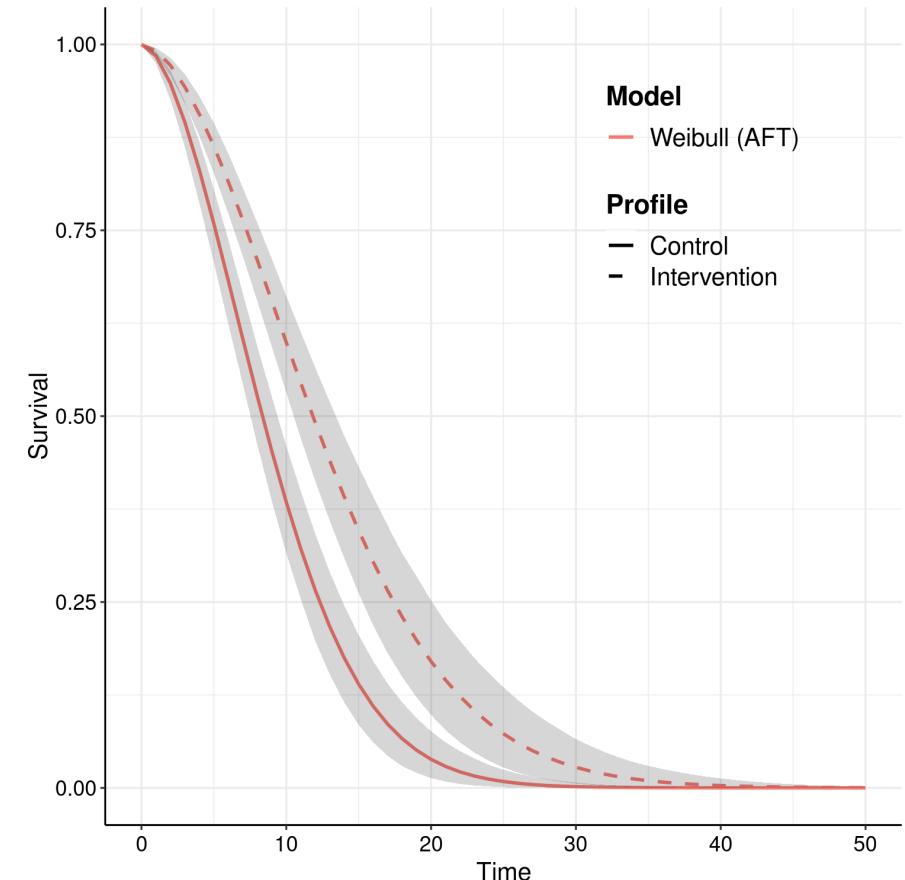
```
> plot(INLA=m.inla,HMC=m.hmc,lab.profile=c("Control","Intervention"))
```



Bayesian survival analysis and PSA

- survHE uses the simulations produced by the **full joint posterior distribution** of all the model parameters
 - If the parameters are not highly correlated, these will look pretty much like the bootstrap-based simulations of flexsurv
 - **BUT** if the they are correlated, they won't be the same as the bootstrap!

```
> # Uses the 'survHE' method 'plot' to do the  
> # extrapolation for 1000 simulations from  
> # the joint posterior distribution, over  
> # a time horizon of 0-50  
> plot(m.hmc,  
+       nsim=1000,  
+       t=seq(0,50),  
+       lab.profile=c("Control", "Intervention")  
+ )
```



Set up/interventions

- ICD (Implantable Cardioverter Defibrillators) compared to anti-arrhythmic drugs (AAD) for prevention of sudden cardiac death in people with cardiac arrhythmia

Set up/interventions

- ICD (Implantable Cardioverter Defibrillators) compared to anti-arrhythmic drugs (AAD) for prevention of sudden cardiac death in people with cardiac arrhythmia

Data

- Individual data from cohort of 535 UK cardiac arrhythmia patients implanted with ICDs between 1991 and 2002
- Meta-analysis of three (non-UK) RCTs providing published HRs – Relatively short-term follow-up: approximately 75% people, followed for less than 5 years, maximum 10 years
- UK population mortality statistics by age, sex, cause of death

Set up/interventions

- ICD (Implantable Cardioverter Defibrillators) compared to anti-arrhythmic drugs (AAD) for prevention of sudden cardiac death in people with cardiac arrhythmia

Data

- Individual data from cohort of 535 UK cardiac arrhythmia patients implanted with ICDs between 1991 and 2002
- Meta-analysis of three (non-UK) RCTs providing published HRs – Relatively short-term follow-up: approximately 75% people, followed for less than 5 years, maximum 10 years
- UK population mortality statistics by age, sex, cause of death

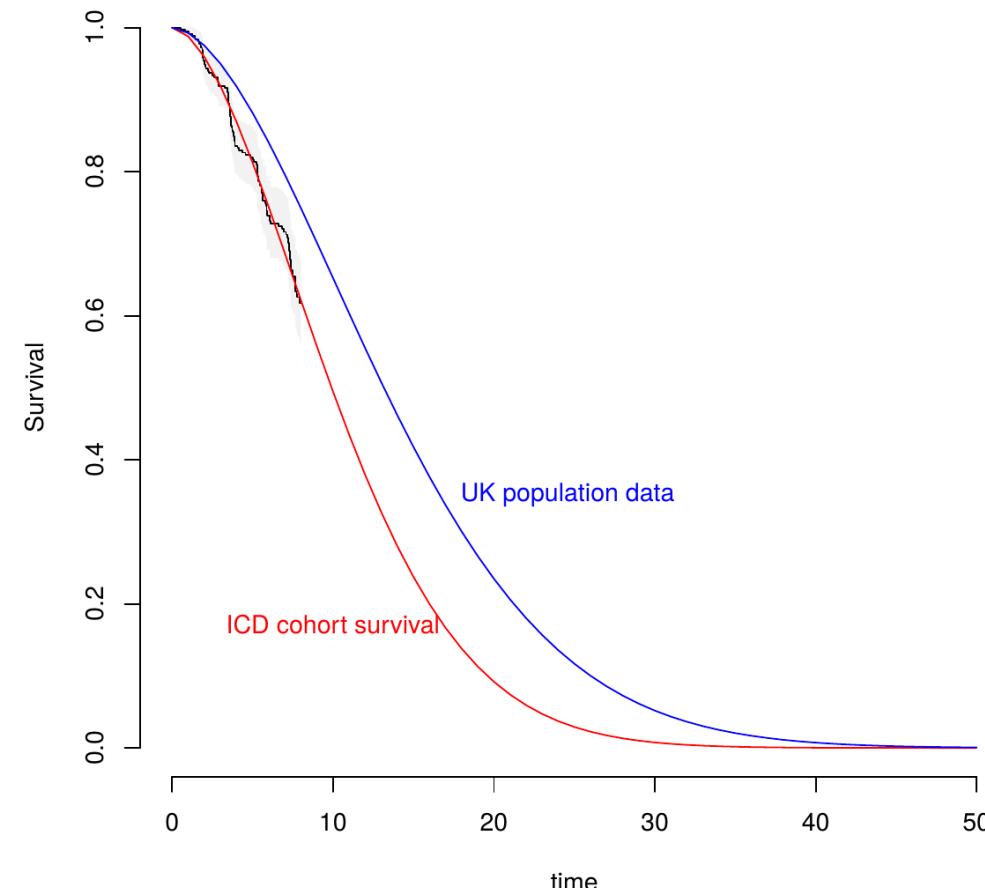
Objective

- Estimate the survival curve over the lifetime of ICD and AAD patients in UK
- Extrapolate the output to inform the wider economic model

Example: ICD & Cardiac death

Basic idea

Use UK population data (matched by age/sex) to "anchor" the ICD population at risk



Basic idea

Use UK population data (matched by age/sex) to "anchor" the ICD population at risk

- Perhaps the easiest way to do this is to relate the hazard between the two populations – eg **proportional hazard (PH)** model

$$h_{\text{ICD}}(t) = e^{\beta} h_{\text{UK}}(t) \quad \Leftrightarrow \quad \text{HR} = \frac{h_{\text{ICD}}(t)}{h_{\text{UK}}(t)} = e^{\beta} = \text{Constant}$$

- Relatively easy to model – but probably very unrealistic!
 - ICD patients are at (much?) greater risk of arrhythmia death
 - If the proportion of deaths caused by arrhythmia changes over time, we would induce bias, because we would be extrapolate a constant HR for all causes mortality

Example: ICD & Cardiac death

Basic idea

Use UK population data (matched by age/sex) to "anchor" the ICD population at risk

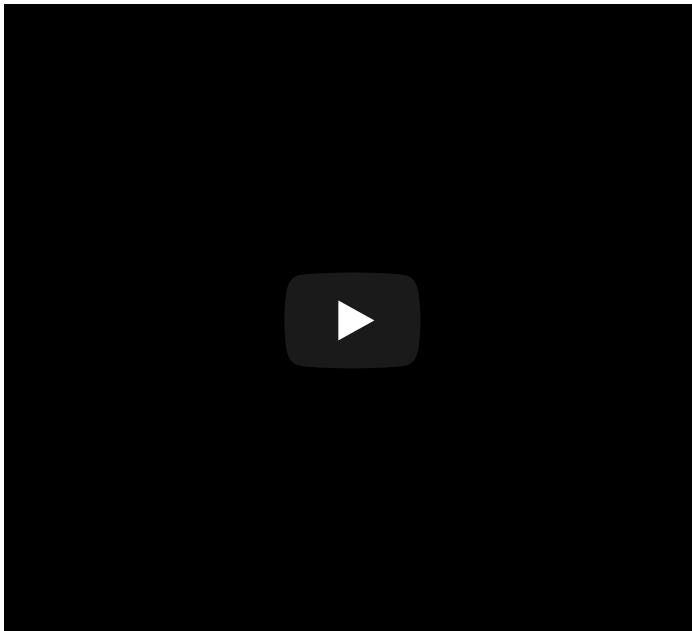
- Perhaps the easiest way to do this is to relate the hazard between the two populations – eg **proportional hazard (PH)** model

$$h_{\text{ICD}}(t) = e^{\beta} h_{\text{UK}}(t) \quad \Leftrightarrow \quad \text{HR} = \frac{h_{\text{ICD}}(t)}{h_{\text{UK}}(t)} = e^{\beta} = \text{Constant}$$

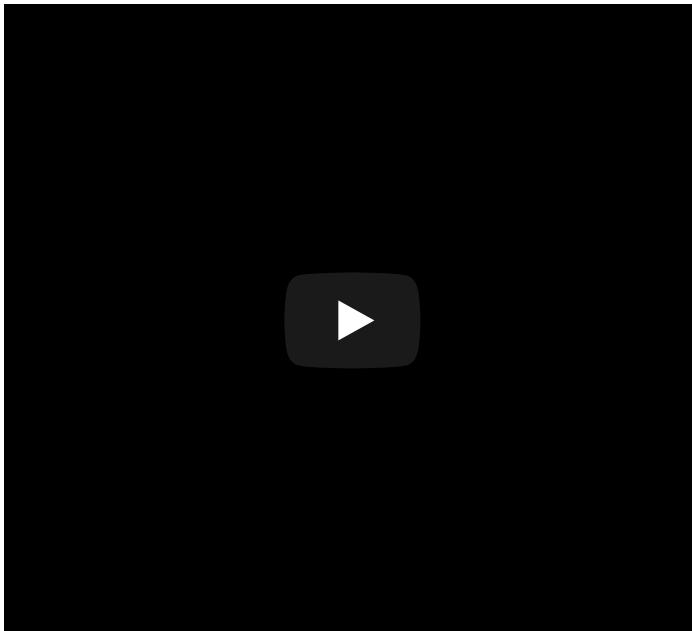
- Relatively easy to model – but probably very unrealistic!
 - ICD patients are at (much?) greater risk of arrhythmia death
 - If the proportion of deaths caused by arrhythmia changes over time, we would induce bias, because we would be extrapolate a constant HR for all causes mortality
- Formally account for multiple mortality causes (**Poly-Weibull** model  **Demiris et al, 2015**):

$$\begin{aligned} h_{\text{ICD}}(t) &= h_{\text{ICD}}^{\text{arr}}(t) + h_{\text{ICD}}^{\text{othe}}(t) \\ &= e^{\beta} h_{\text{UK}}^{\text{arr}}(t) + h_{\text{UK}}^{\text{othe}}(t) \\ &= e^{\beta} \alpha_1 \mu_1 t^{\alpha_1-1} + \alpha_2 \mu_2 t^{\alpha_2-1} \end{aligned}$$

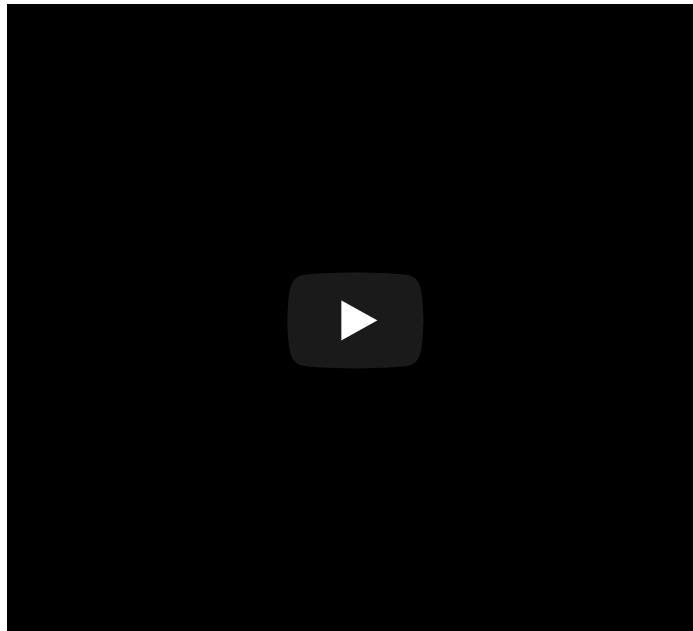
- This assumes that
 - Arrhythmia hazard is **proportional** to matched UK population
 - Other causes hazard is **identical** to matched UK population



- To set up a full Bayesian model including a reasonable specification of the priors can be a hard task
- Often people claim that they have "no prior information".



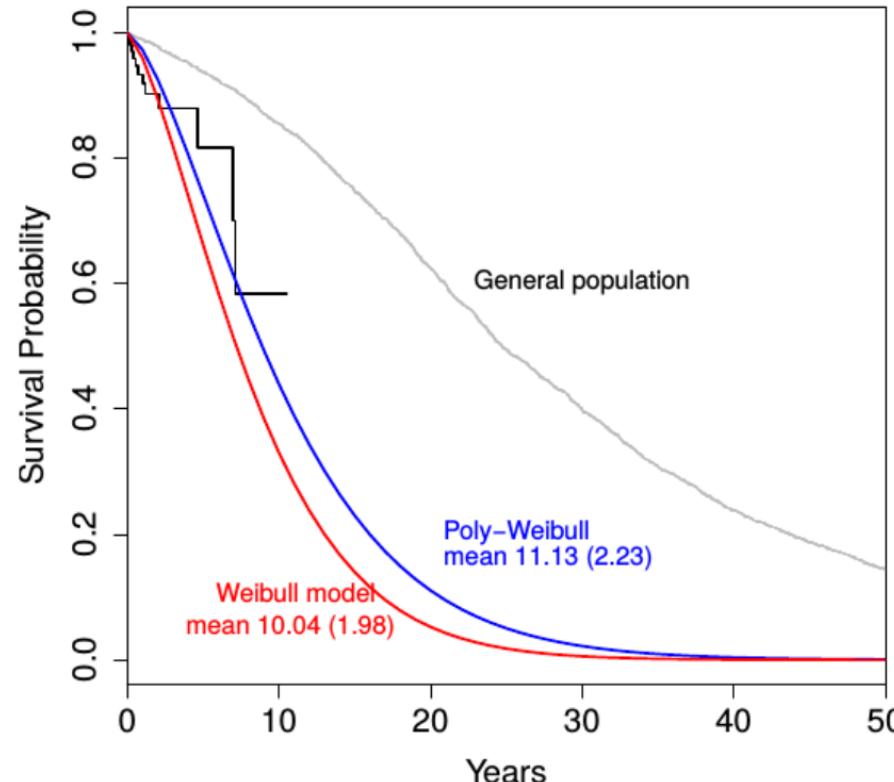
- To set up a full Bayesian model including a reasonable specification of the priors can be a hard task
- Often people claim that they have "no prior information". **But: don't they?...**



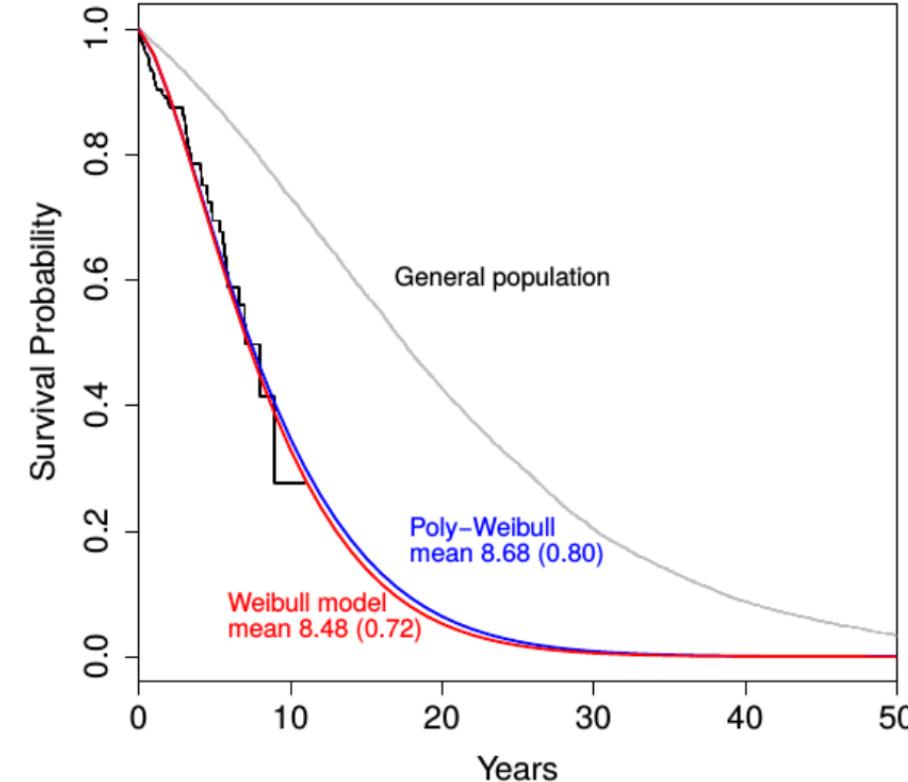
- To set up a full Bayesian model including a reasonable specification of the priors can be a hard task
- Often people claim that they have "no prior information". **But: don't they?...**
- In the ICD case, age at entry is around 60 – we **know** that people won't survive more than 60 more years
 - Setting a prior for the scale $\mu_i \sim \text{Uniform}(0, 100)$ implies that the prior mean survival of the resulting Weibull distribution is
$$\mu_i \Gamma\left(1 + \frac{1}{\alpha}\right) < 60$$
- Can also include some knowledge on the shape α and the coefficient β to limit their variations in reasonable ranges...

Results

Women



Men



- Ignoring cause-specific mortality (**Weibull**) results in larger bias, especially for females (because the arrhythmia proportion of deaths does vary over time in that subgroup)

 Next lecture