ACCELERATED FAILURE TIME MODEL

Dr. Aric LaBarr
Institute for Advanced Analytics

MODEL STRUCTURE

Accelerated Failure Time Model

- The accelerated failure time (AFT) model is a regression that relates covariates (independent variables) to the event time *T*.
- The AFT model is a parametric model – depends on knowledge of the underlying distribution of the data.

$$T_i = e^{\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i}$$

Accelerated Failure Time Model

- We can transform this model into a linear regression model by taking the natural log of both sides of the equation:

$$T_i = e^{\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i}$$

- The equation now becomes:

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Accelerated Failure Time Model

- We can transform this model into a linear regression model by taking the natural log of both sides of the equation:

$$T_i = e^{\beta_0+\beta_1 x_{i,1}+\cdots+\beta_k x_{i,k}+\sigma e_i}$$

- The equation now becomes:

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Ensures positive predictions of *T*

Accelerated Failure Time Model

- We can transform this model into a linear regression model by taking the natural log of both sides of the equation:

$$T_i = e^{\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i}$$

- The equation now becomes:

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Variables used to predict *T*

Accelerated Failure Time Model

- We can transform this model into a linear regression model by taking the natural log of both sides of the equation:

$$T_i = e^{\beta_0+\beta_1 x_{i,1}+\cdots+\beta_k x_{i,k}+\sigma e_i}$$

- The equation now becomes:

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Variance of the errors

Accelerated Failure Time Model

- We can transform this model into a linear regression model by taking the natural log of both sides of the equation:

$$T_i = e^{\beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i}$$

- The equation now becomes:

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Errors in the model

Accelerated Failure Time Model

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Errors in the model

- The errors in the AFT model can follow many different distributions.
- Assumptions:
  - Specify correct distribution of errors
  - Constant Mean
  - Constant Variance (σ)
  - Independence across observations

Accelerated Failure Time Model

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Variables used to predict *T*

- If there is no censoring in the data, traditional OLS could estimate the parameters.
- If there is censoring, maximum likelihood estimation could estimate the parameters (MOST LIKELY SCENARIO).

AFT Model – SAS

```
proc lifereg data=Survival.Recid;
   model week*arrest(0) = fin age race wexp mar paro prio /
                             dist=lnormal;
run;
```

AFT Model – SAS

The LIFEREG Procedure

Model Information	
Data Set	SURVIVAL.RECID
Dependent Variable	Log(week)
Censoring Variable	arrest
Censoring Value(s)	0
Number of Observations	432
Noncensored Values	114
Right Censored Values	318
Left Censored Values	0
Interval Censored Values	0
Number of Parameters	9
Name of Distribution	Lognormal
Log Likelihood	-322.6945851

Number of Observations Read	432
Number of Observations Used	432

AFT Model – SAS

Fit Statistics	
-2 Log Likelihood	645.389
AIC (smaller is better)	663.389
AICC (smaller is better)	663.816
BIC (smaller is better)	700.005

Fit Statistics (Unlogged Response)	
-2 Log Likelihood	1366.469
Lognormal AIC (smaller is better)	1384.469
Lognormal AICC (smaller is better)	1384.896
Lognormal BIC (smaller is better)	1421.085

Algorithm converged.

AFT Model – SAS

Type III Analysis of Effects			
Effect	**DF**	**Wald Chi-Square**	**Pr > ChiSq**
fin	1	4.3657	0.0367
age	1	2.9806	0.0843
race	1	1.8824	0.1701
wexp	1	2.2466	0.1339
mar	1	2.4328	0.1188
paro	1	0.1092	0.7411
prio	1	5.8489	0.0156

AFT Model – SAS

Analysis of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.2677	0.4617	3.3628	5.1726	85.44	<.0001
fin	1	0.3428	0.1641	0.0212	0.6645	4.37	0.0367
age	1	0.0272	0.0158	-0.0037	0.0581	2.98	0.0843
race	1	-0.3632	0.2647	-0.8819	0.1556	1.88	0.1701
wexp	1	0.2681	0.1789	-0.0825	0.6187	2.25	0.1339
mar	1	0.4604	0.2951	-0.1181	1.0388	2.43	0.1188
paro	1	0.0559	0.1691	-0.2756	0.3873	0.11	0.7411
prio	1	-0.0655	0.0271	-0.1186	-0.0124	5.85	0.0156
Scale	1	1.2946	0.0990	1.1145	1.5038		

AFT Model – SAS

Analysis of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.2677	0.4617	3.3628	5.1726	85.44	<.0001
fin	1	0.3428	0.1641	0.0212	0.6645	4.37	0.0367
age	1	0.0272	0.0158	-0.0037	0.0581	2.98	0.0843
race	1	-0.3632	0.2647	-0.8819	0.1556	1.88	0.1701
wexp	1	0.2681	0.1789	-0.0825	0.6187	2.25	0.1339
mar	1	0.4604	0.2951	-0.1181	1.0388	2.43	0.1188
paro	1	0.0559	0.1691	-0.2756	0.3873	0.11	0.7411
prio	1	-0.0655	0.0271	-0.1186	-0.0124	5.85	0.0156
Scale	1	1.2946	0.0990	1.1145	1.5038		

AFT Model – R

```
recid.aft.ln <- survreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                        data = recid, dist = 'lognormal')

summary(recid.aft.ln)
```

AFT Model – R

```
## Call:
## survreg(formula = Surv(week, arrest == 1) ~ fin + age + race +
##     wexp + mar + paro + prio, data = recid, dist = "lognormal")
##               Value Std. Error     z       p
## (Intercept)  4.2677     0.4617  9.24 < 2e-16
## fin          0.3428     0.1641  2.09 0.03667
## age          0.0272     0.0158  1.73 0.08427
## race        -0.3632     0.2647 -1.37 0.17006
## wexp         0.2681     0.1789  1.50 0.13391
## mar          0.4604     0.2951  1.56 0.11882
## paro         0.0559     0.1691  0.33 0.74108
## prio        -0.0655     0.0271 -2.42 0.01559
## Log(scale)   0.2582     0.0764  3.38 0.00073
##
## Scale= 1.29
##
## Log Normal distribution
## Loglik(model)= -683.2   Loglik(intercept only)= -697.9
##  Chisq= 29.35 on 7 degrees of freedom, p= 0.00012
## Number of Newton-Raphson Iterations: 4
## n= 432
```

INTERPRETATION

AFT Model Parameter Interpretation

- If a parameter estimate is **positive**, increases in that variable **increase** the expected survival time.
- If a parameter estimate is **negative**, increases in that variable **decrease** expected survival times.
- If a parameter estimate is **zero**, increases in that variable have **no impact** on expected survival times.

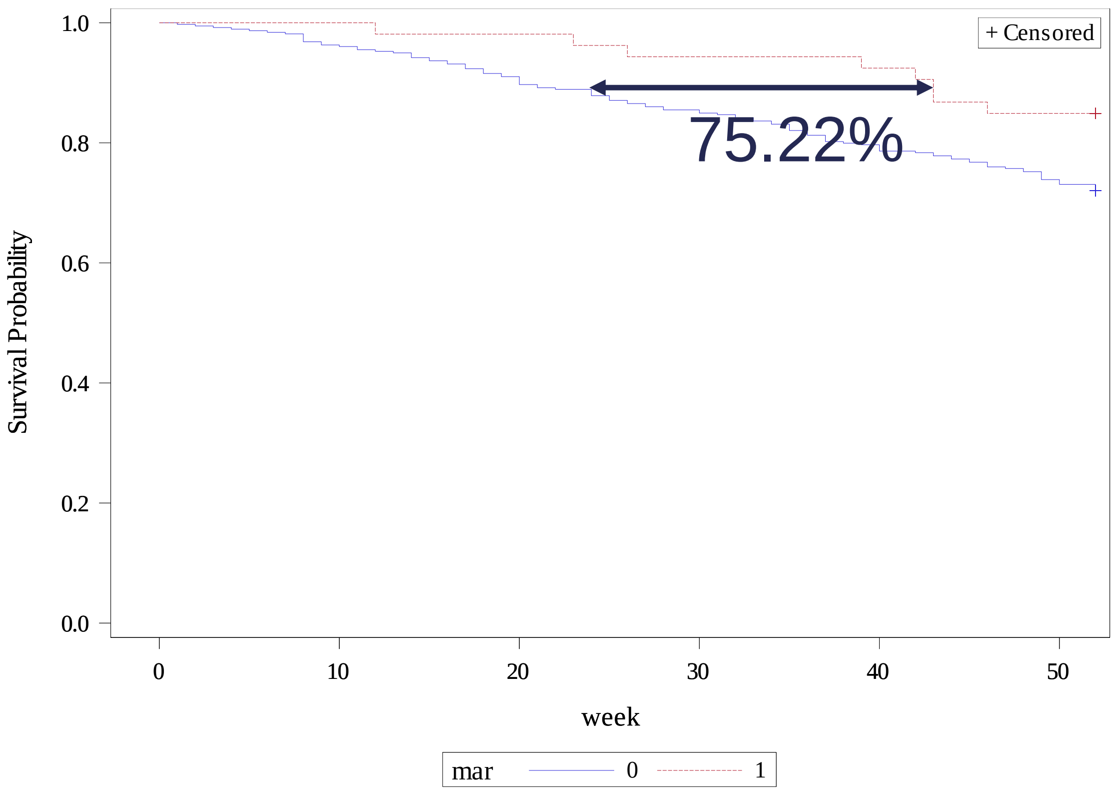
- $100 \times (e^{\beta} - 1)$ is the % increase in the expected survival time for each one-unit increase in the variable.

Recidivism Parameter Interpretation

Variable	β Estimate	$100\left(e^{\beta}-1\right)$
Financial Aid	0.3319	39.36%
Age at Release	0.0333	3.39%
Marital Status	0.5609	75.22%
Prior Convictions	-0.0743	-7.16%

Recidivism Parameter Interpretation

Variable	β Estimate	$100(e^{\beta} - 1)$
Marital Status	0.5609	75.22%



ERROR DISTRIBUTIONS

Model Assumptions

Parametric Models

- AFT models are **parametric** – we assume failure time has a particular structure and distribution.
- Kaplan-Meier estimation is **nonparametric** – makes no assumption on failure time.

- Parametric methods allow for more detailed/precise estimation than nonparametric methods **IF** the distribution is specified correctly.
  - Ex: Easier to estimate medians, survival & hazard functions.

Accelerated Failure Time Model

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Errors in the model

- The errors in the AFT model can follow many different distributions.
- Assumptions:
  - **Specify correct distribution of errors**
  - Constant Mean
  - Constant Variance (σ)
  - Independence across observations

Variance (Scale) vs. Rate

$$\log T_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \sigma e_i$$

Variance of the errors

- Variance (also called scale in survival analysis) describes the spread of the distribution of errors.
- Another common form is the inverse of the scale, called the **rate**: $\lambda = 1/\sigma$.
- If σ is small, then events are not spread out → events happening close to one another → higher rate of events, or λ is large.

ERROR DISTRIBUTIONS

Common Distributions

Alternative Distributions

- We will focus on the distribution of failure time *T* (not on the error itself) since this is what we input into software.
- Distributions are commonly checked two ways:
  1. Graphically
  2. Statistical Tests

- We will go over some commonly used distributions for survival data, but there is **no guarantee** that your data will adequately match just one of the distributions here, or even any of them at all.

Exponential Distribution

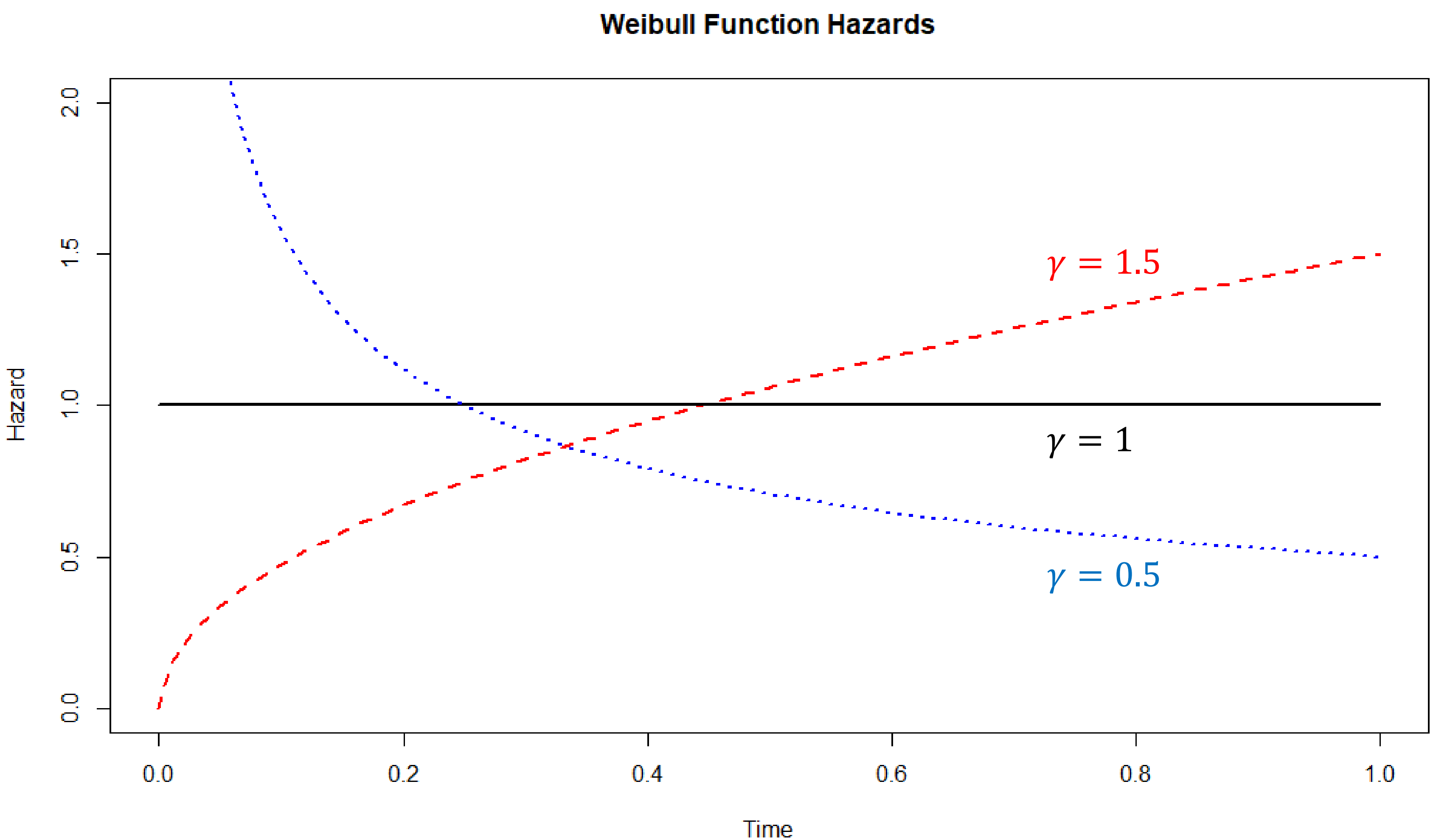
- Simplest distribution is the **exponential distribution** – constant hazard that doesn't depend on time.
  - Survival function: $S(t) = e^{-\lambda t}$
  - Hazard function: $h(t) = \lambda$

- Constant hazard commonly used when failures are completely random:
  - Light bulbs
  - Electronics
  - Etc.

Weibull Distribution

- Most commonly used distribution is the **Weibull** distribution, which has an additional *shape* parameter γ.
  - Survival function: $S(t) = e^{-(\lambda t)^{\gamma}}$
  - Hazard function: $h(t) = \lambda\gamma(\lambda t)^{\gamma-1}$
- The shape parameter γ determines whether the hazard increases or decreases with time:
  - $\gamma > 1$: hazard **increasing** with time (Ex: aging parts "wear out")
  - $\gamma < 1$: hazard **decreasing** with time (Ex: post-surgery complications)

Weibull Distribution Hazards



Exponential vs. Weibull

- Hazard for Weibull is constant when $\gamma = 1$.
- Weibull distribution **IS** the exponential distribution when $\gamma = 1$!

- Both R and SAS can test this.
  - R: Log(scale) p-value → testing if H_0: $\log\left(\frac{1}{\gamma}\right) = 0$
  - SAS: Lagrange Multiplier Test for Scale p-value → testing if H_0: $\gamma = 1$

Exponential vs. Weibull

- Hazard for Weibull is constant when $\gamma = 1$.
- Weibull distribution **IS** the exponential distribution when $\gamma = 1$!

- Both R and SAS can test this.
  - R: Log(scale) p-value → testing if $\boxed{H_0\colon \log\left(\frac{1}{\gamma}\right) = 0}$
  - SAS: Lagrange Multiplier Test for Scale p-value → testing if $\boxed{H_0\colon \gamma = 1}$

SAME THING!

Exponential vs. Weibull

- Hazard for Weibull is constant when $\gamma = 1$.
- Weibull distribution **IS** the exponential distribution when $\gamma = 1$!

- Both R and SAS can test this.
  - R: Log(scale) p-value → testing if $H_0\colon \log\left(\frac{1}{\gamma}\right) = 0$
  - SAS: Lagrange Multiplier Test for Scale p-value → testing if $H_0\colon \gamma = 1$

WAIT WHAT?!?!?
ISN'T γ SHAPE?

Note on Parameterization

- With the scale vs. rate or shape vs. scale thing, there are a couple of different ways to write the Weibull distribution, and they're all fairly common.
  - `?survreg`: "There are multiple ways to parameterize a Weibull distribution. The `survreg` function embeds it in a general location-scale family, which is a different parameterization than the `rweibull` function, and often leads to confusion."
  - `proc lifereg` documentation: "The Weibull with *Scale=1* is an exponential distribution."

Matching up the parameterization

R	SAS	Parameter
	`proc lifereg` "Weibull Shape"	γ
`survreg` "scale"	`proc lifereg` "scale"	$1/\gamma$
`survreg` "intercept"	`proc lifereg` "intercept"	$-\log\lambda$

Exponential vs. Weibull – SAS

```
proc lifereg data=Survival.Recid;
   model week*arrest(0) = fin age race wexp mar paro prio /
                                       dist=exponential;
run;
```

Exponential vs. Weibull – SAS

Analysis of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.0507	0.5860	2.9021	5.1993	47.78	<.0001
fin	1	0.3663	0.1911	-0.0083	0.7408	3.67	0.0553
age	1	0.0556	0.0218	0.0128	0.0984	6.48	0.0109
race	1	-0.3049	0.3079	-0.9085	0.2986	0.98	0.3220
wexp	1	0.1467	0.2117	-0.2682	0.5617	0.48	0.4882
mar	1	0.4270	0.3814	-0.3205	1.1745	1.25	0.2629
paro	1	0.0826	0.1956	-0.3007	0.4660	0.18	0.6726
prio	1	-0.0857	0.0283	-0.1412	-0.0302	9.15	0.0025
Scale	0	1.0000	0.0000	1.0000	1.0000		
Weibull Shape	0	1.0000	0.0000	1.0000	1.0000		

Lagrange Multiplier Statistics		
Parameter	Chi-Square	Pr > ChiSq
Scale	24.9302	<.0001

Exponential vs. Weibull – R

```
recid.aft.w <- survreg(Surv(week, arrest == 1) ~
                       fin + age + race + wexp + mar + paro + prio,
                       data = recid, dist = 'weibull')

summary(recid.aft.w)
```

Exponential vs. Weibull – R

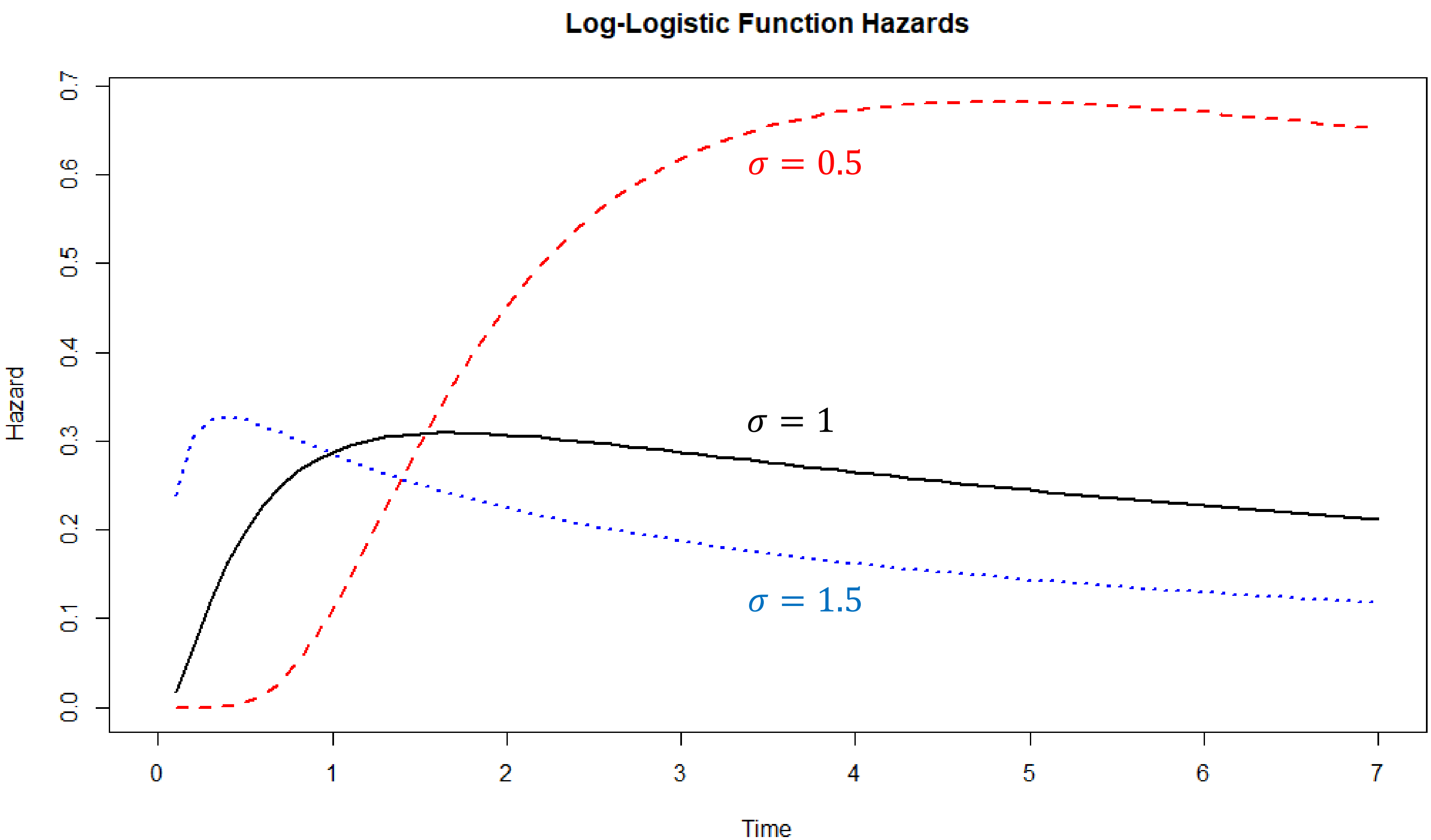
```
## Call:
## survreg(formula = Surv(week, arrest == 1) ~ fin + age + race +
##     wexp + mar + paro + prio, data = recid, dist = "weibull")
##               Value Std. Error     z       p
## (Intercept)  3.9901     0.4191  9.52 < 2e-16
## fin          0.2722     0.1380  1.97 0.04852
## age          0.0407     0.0160  2.54 0.01096
## race        -0.2248     0.2202 -1.02 0.30721
## wexp         0.1066     0.1515  0.70 0.48196
## mar          0.3113     0.2733  1.14 0.25473
## paro         0.0588     0.1396  0.42 0.67355
## prio        -0.0658     0.0209 -3.14 0.00167
## Log(scale)  -0.3391     0.0890 -3.81 0.00014
##
## Scale= 0.712
##
## Weibull distribution
## Loglik(model)= -679.9   Loglik(intercept only)= -696.6
##  Chisq= 33.42 on 7 degrees of freedom, p= 2.2e-05
## Number of Newton-Raphson Iterations: 6
## n= 432
```

Other Distributions

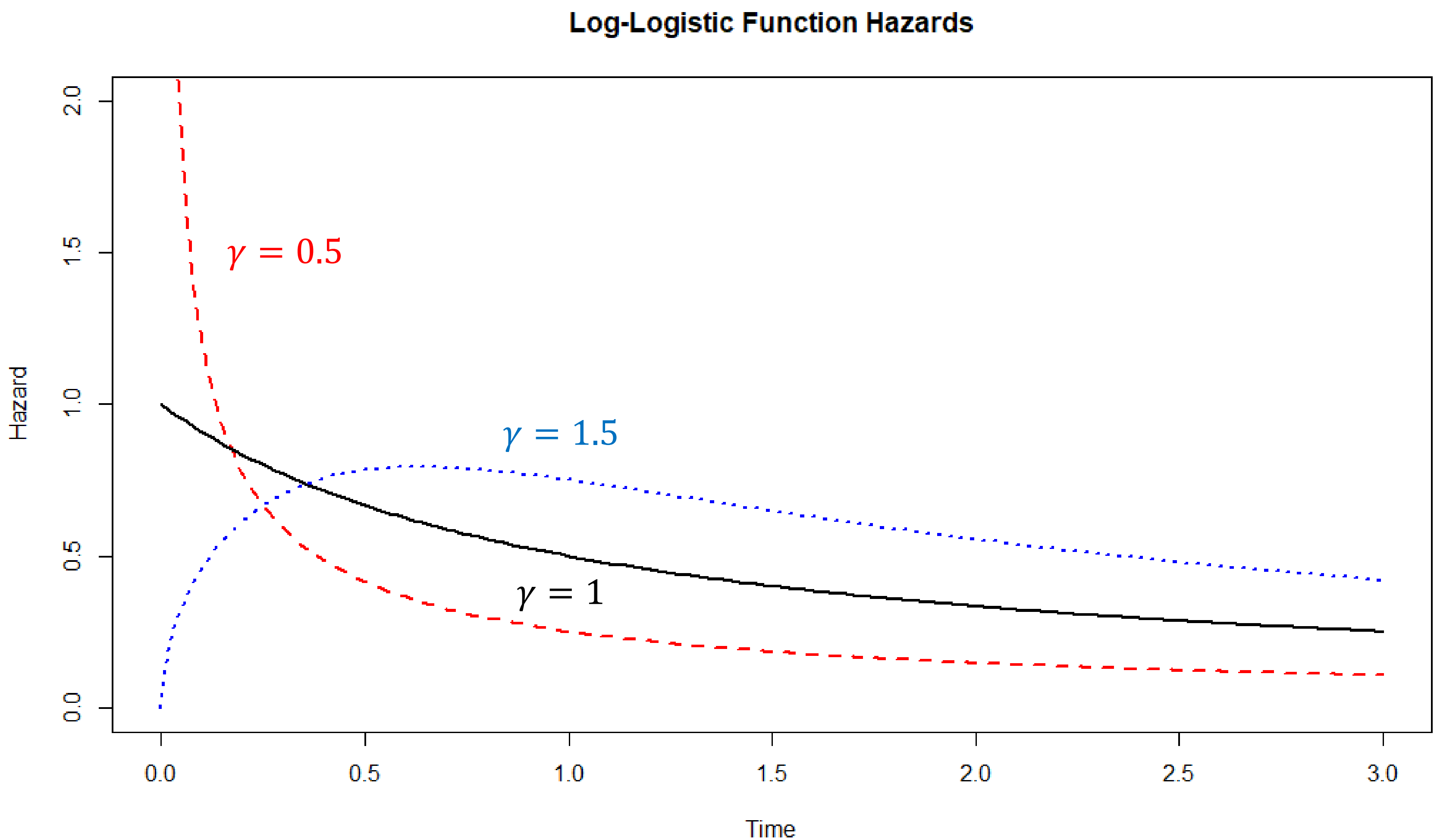
- **Log-Normal Distribution:** If T has a log-normal distribution, then ε follows a normal distribution.
  - IF NO CENSORING, log-normal AFT = linear regression with $y = \log(T)$ are equivalent.

- **Log-Logistic Distribution:** Allows hazard to increase then decrease if $\gamma > 1$.
  - Log-logistic AFT model is just an ordinal logistic regression model!

Log-Normal Hazard

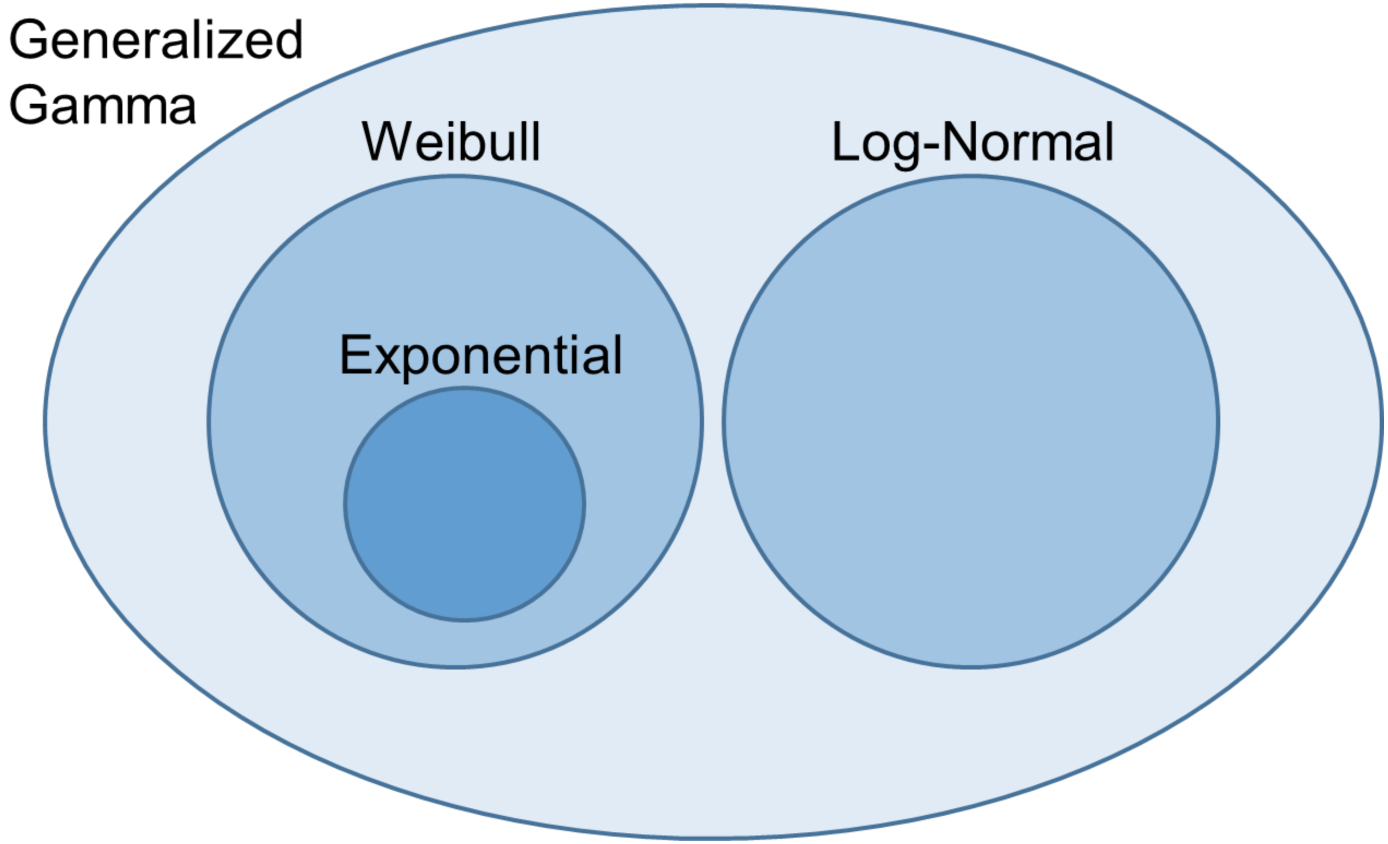


Log-Logistic Hazard



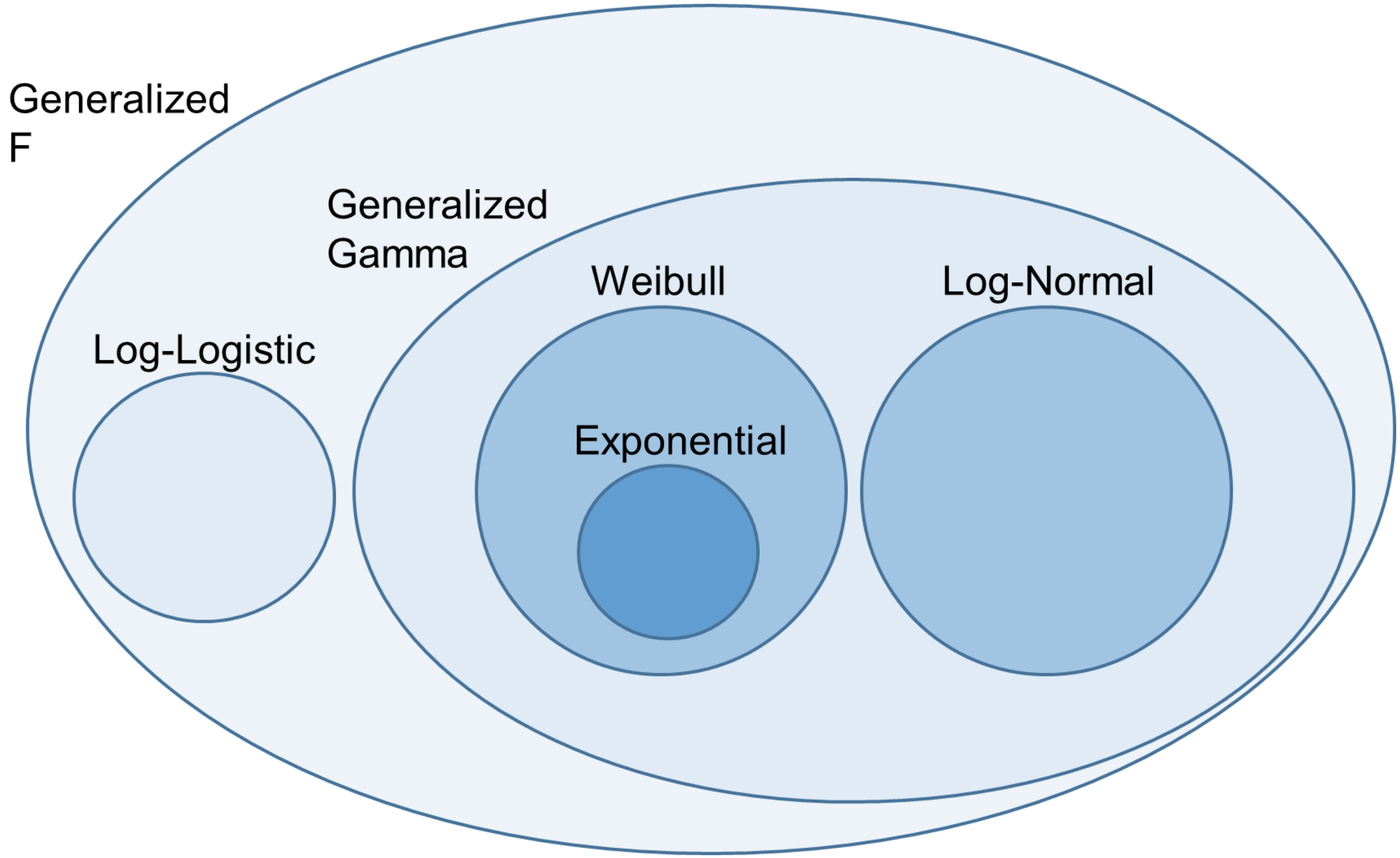
Other Distributions

- **Generalized Gamma Distribution:** Includes log-normal and Weibull as special cases.



Other Distributions

- **Generalized F Distribution:** Includes log-logistic and generalized gamma as special cases.



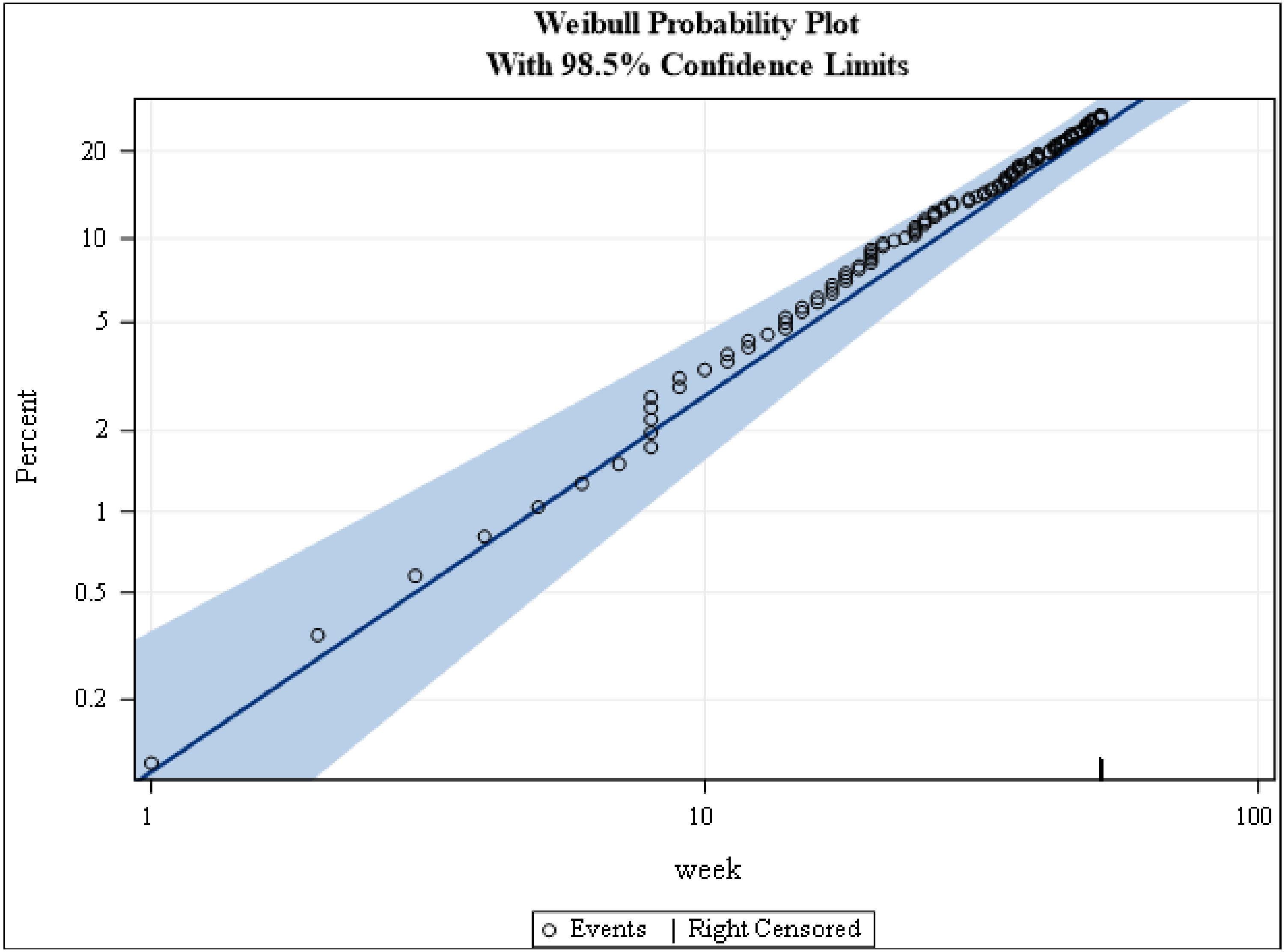
Checking Distributions – SAS

```
proc lifereg data=Survival.Recid;
   model Week*arrest(0) = fin age race wexp mar paro prio /
                              alpha=0.015 dist=weibull;
   probplot;
run;
```

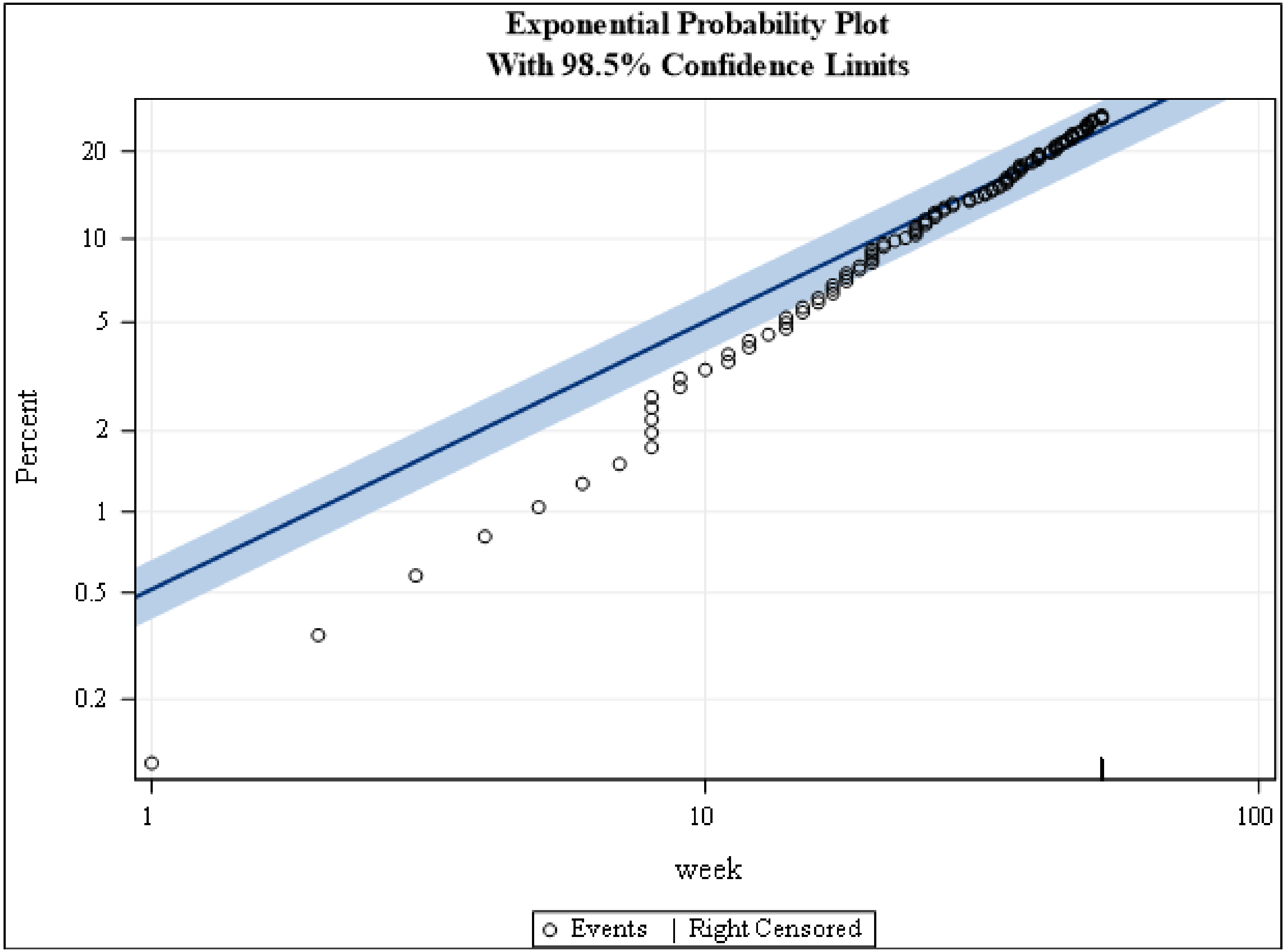
Checking Distributions – SAS

Analysis of Maximum Likelihood Parameter Estimates							
Parameter	**DF**	**Estimate**	**Standard Error**	**95% Confidence Limits**		**Chi-Square**	**Pr > ChiSq**
Intercept	1	3.9901	0.4191	3.1687	4.8115	90.65	<.0001
fin	1	0.2722	0.1380	0.0018	0.5426	3.89	0.0485
age	1	0.0407	0.0160	0.0093	0.0721	6.47	0.0110
race	1	-0.2248	0.2202	-0.6563	0.2067	1.04	0.3072
wexp	1	0.1066	0.1515	-0.1905	0.4036	0.49	0.4820
mar	1	0.3113	0.2733	-0.2244	0.8469	1.30	0.2547
paro	1	0.0588	0.1396	-0.2149	0.3325	0.18	0.6735
prio	1	-0.0658	0.0209	-0.1069	-0.0248	9.88	0.0017
Scale	1	0.7124	0.0634	0.5983	0.8482		
Weibull Shape	1	1.4037	0.1250	1.1789	1.6713		

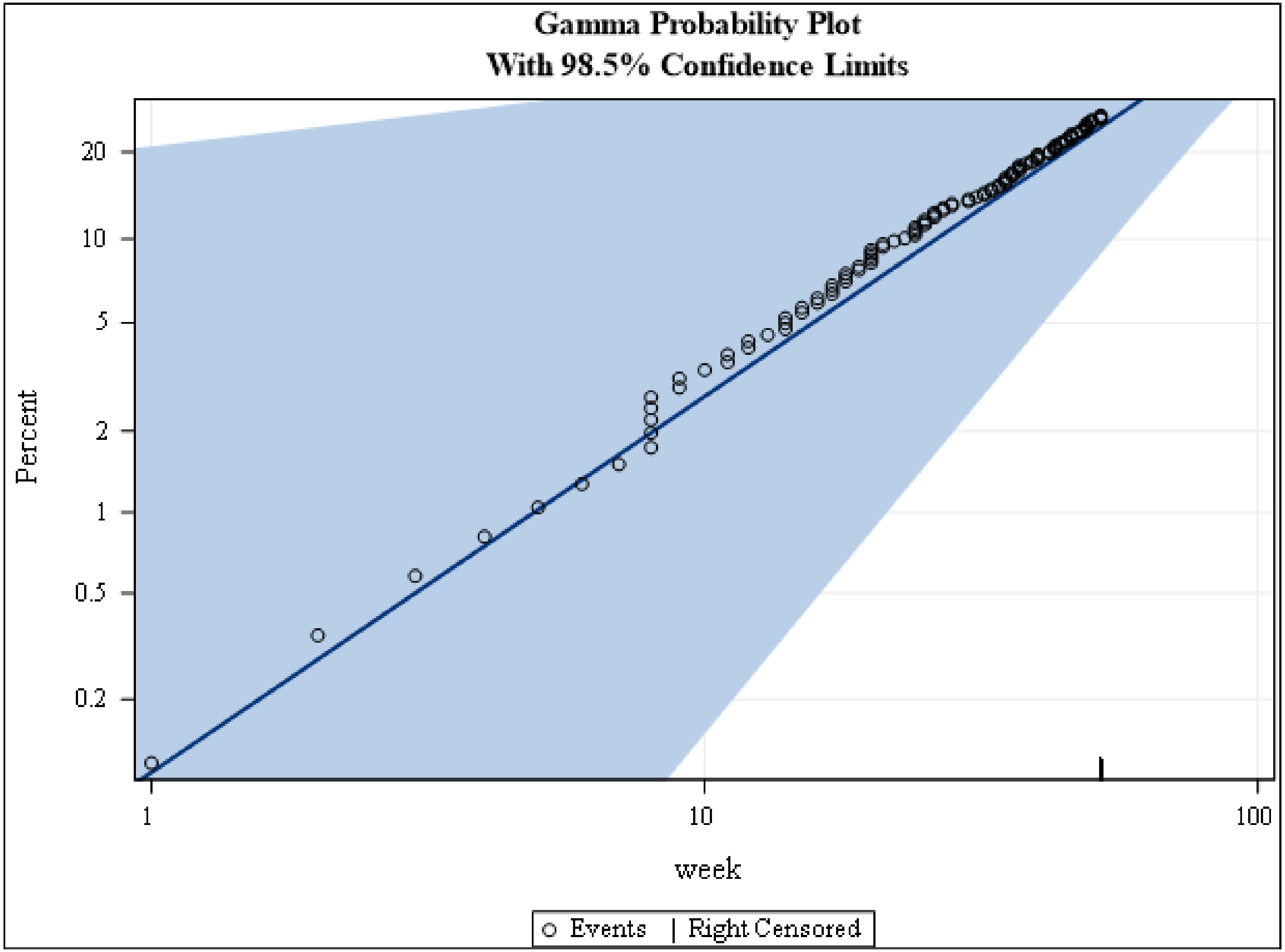
Checking Distributions – SAS



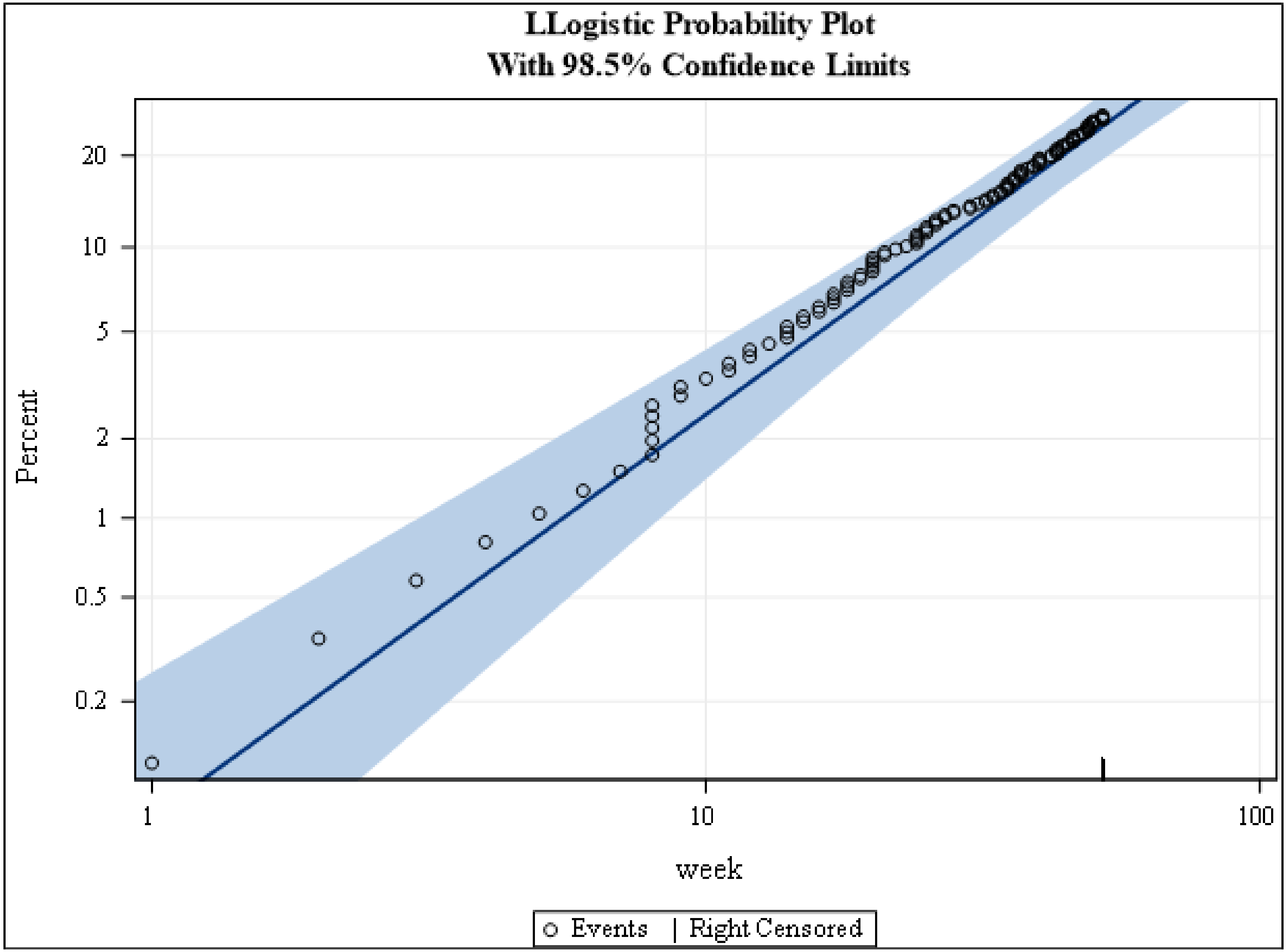
Checking Distributions – SAS



Checking Distributions – SAS



Checking Distributions – SAS

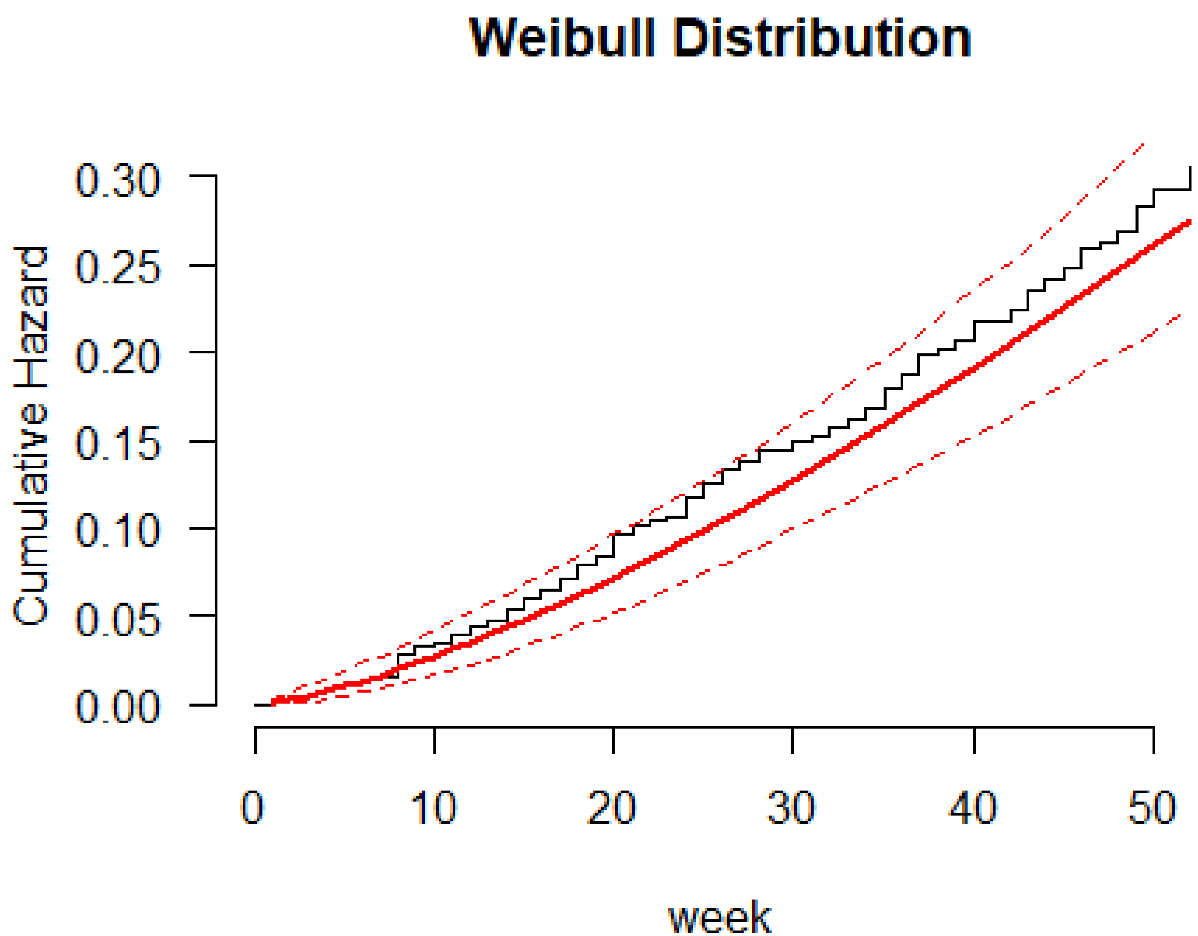


Checking Distributions – R

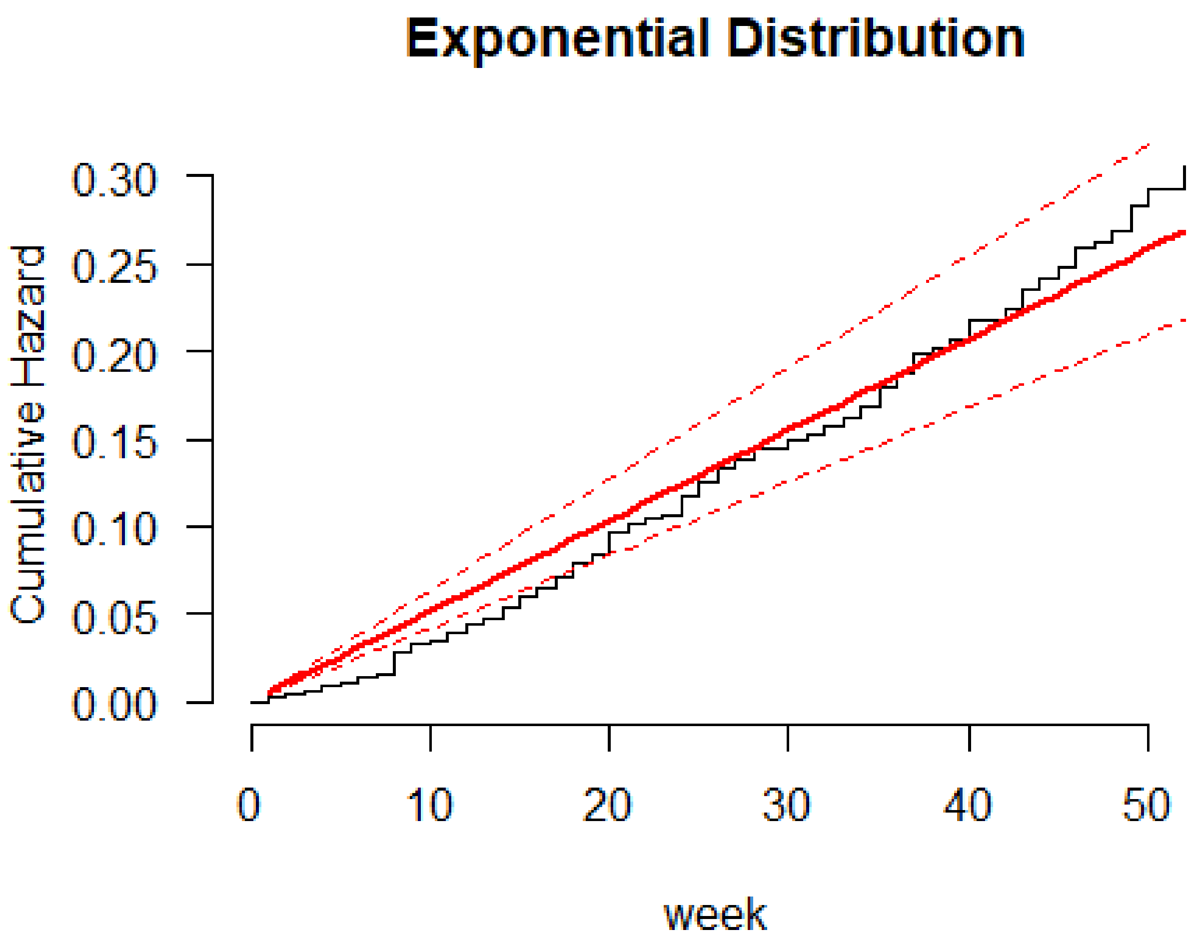
```
recid.aft.w <- flexsurvreg(Surv(week, arrest == 1) ~
                           fin + age + race + wexp +
                           mar + paro + prio,
                          data = recid, dist = "weibull")

plot(recid.aft.w, type = "cumhaz", ci = TRUE, conf.int = FALSE,
     las = 1, bty = "n", xlab = "week", ylab = "Cumulative Hazard",
     main = "Weibull Distribution")
```

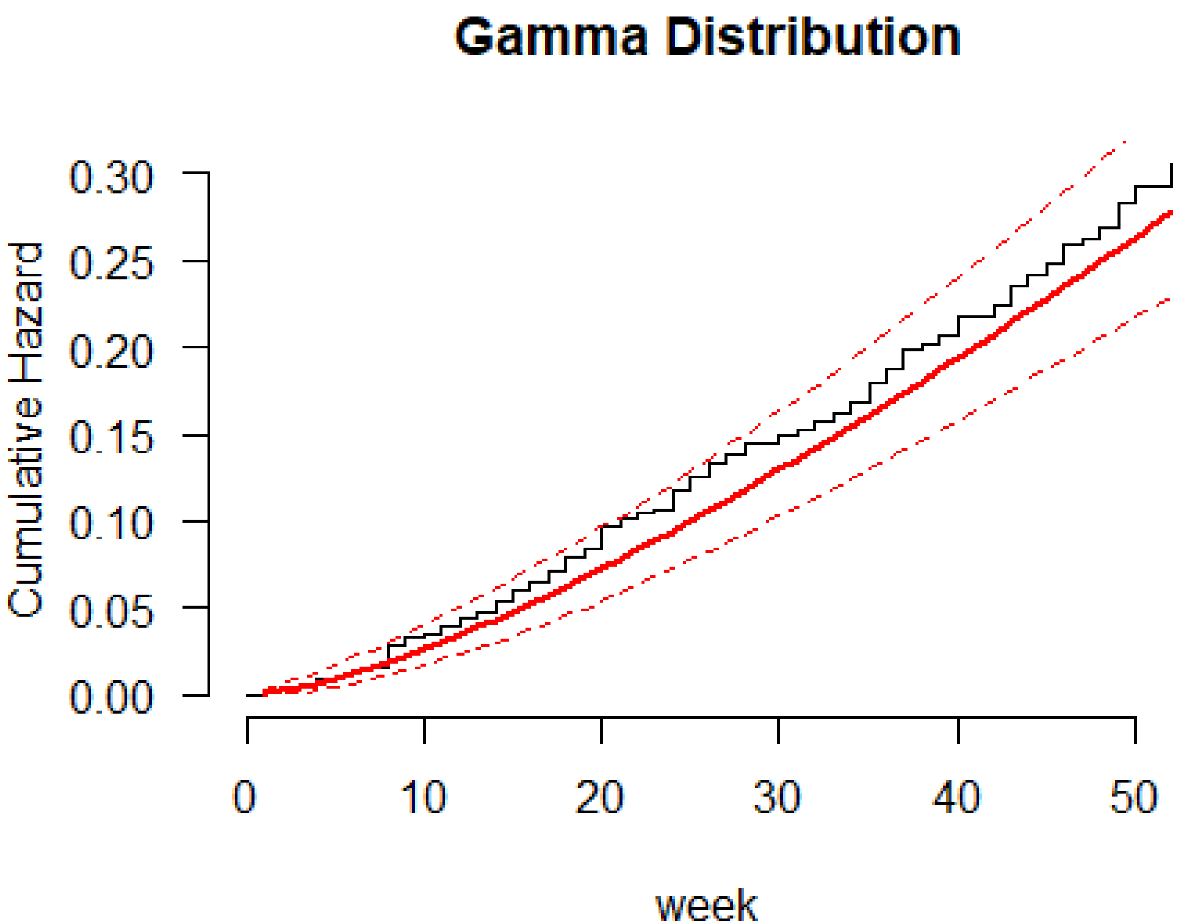
Checking Distributions – R



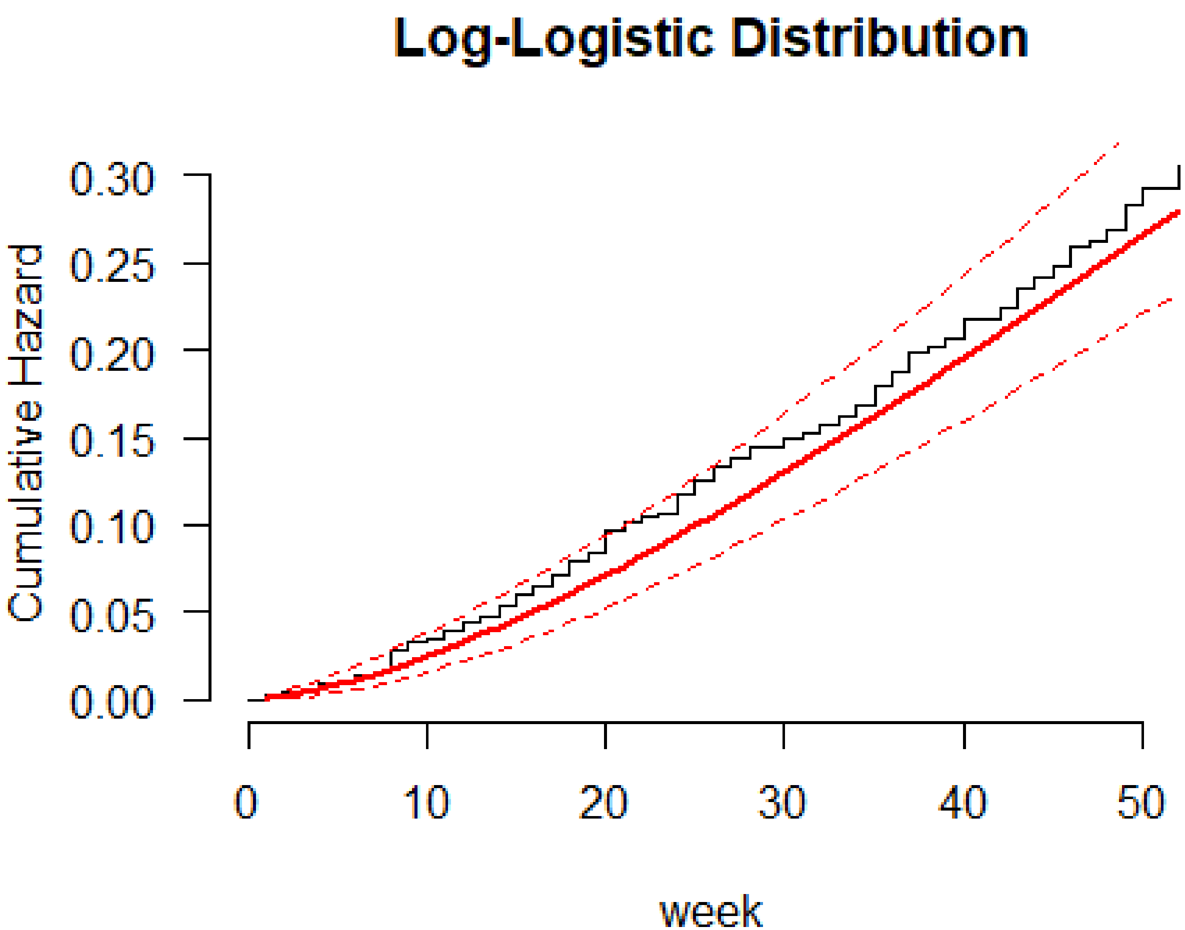
Checking Distributions – R



Checking Distributions – R



Checking Distributions – R



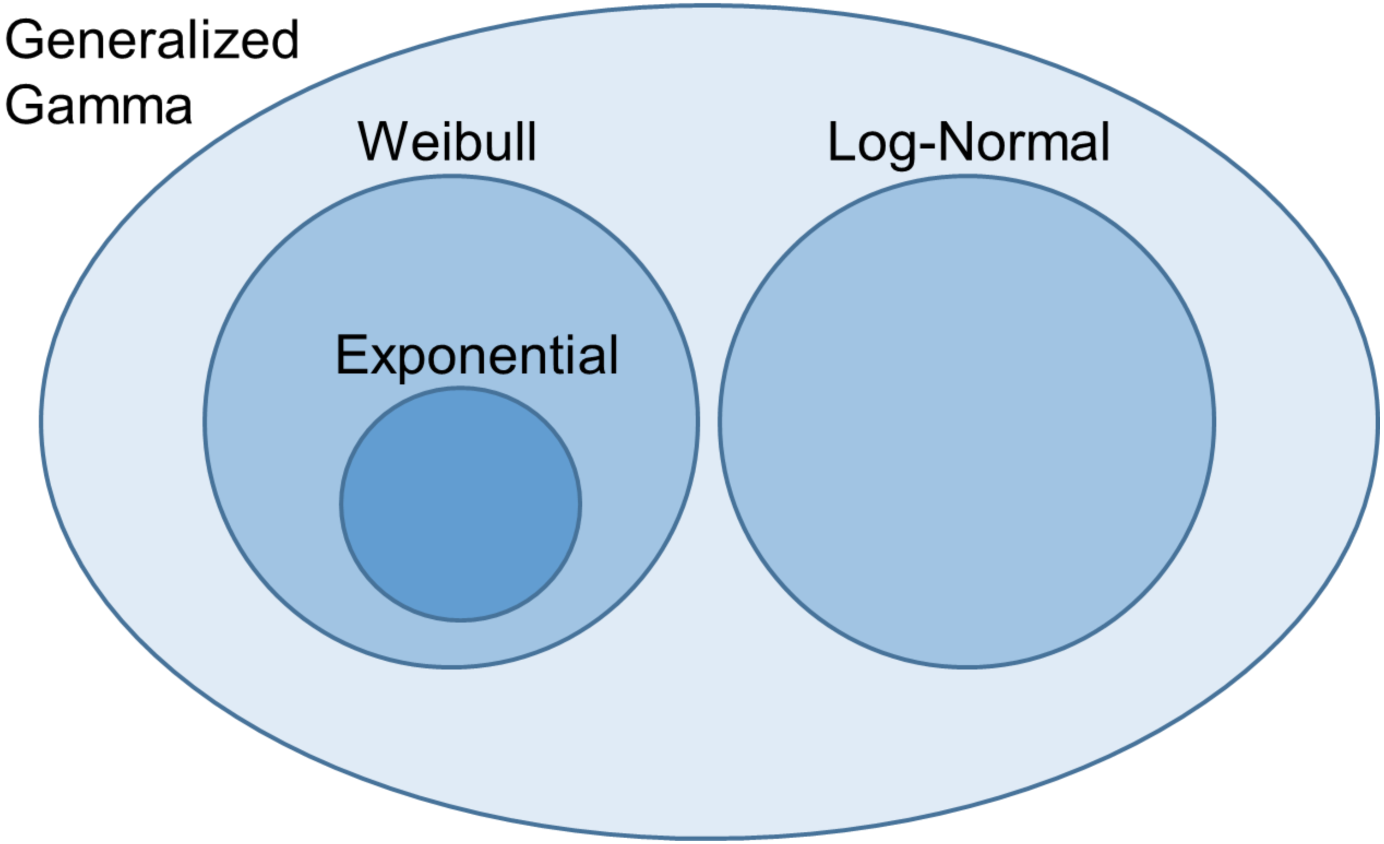
Alternative Distributions

- We will focus on the distribution of failure time T (not on the error itself) since this is what we input into software.
- Distributions are commonly checked two ways:
  1. Graphically
  2. Statistical Tests ← **OH YEAH!**

- We will go over some commonly used distributions for survival data, but there is **no guarantee** that your data will adequately match just one of the distributions here, or even any of them at all.

Nested Distributions!

- **Generalized Gamma Distribution:** Includes log-normal and Weibull as special cases.



Goodness-of-Fit Tests

- Since these models are nested within the generalized gamma, we can use the **likelihood ratio test**.
- Likelihood Ratio Test:

$$\mathrm{LRT} = -2(\log L_{Nested} - \log L_{Full})$$

- Typically, use **full model** (all variables) since we don't know which p-values are correct.

Goodness-of-Fit Tests – SAS

Fit Statistics	
-2 Log Likelihood	651.652
AIC (smaller is better)	667.652
AICC (smaller is better)	667.992
BIC (smaller is better)	700.199

Fit Statistics (Unlogged Response)	
-2 Log Likelihood	1372.732
Exponential AIC (smaller is better)	1388.732
Exponential AICC (smaller is better)	1389.072
Exponential BIC (smaller is better)	1421.279

Goodness-of-Fit Tests

- Here are the log-likelihood values for the models we can compare in SAS:

Log-Likelihood Value	Implied Distribution
-686.37	Exponential
-679.92	Weibull
-683.23	Log-Normal
-679.92	Generalized Gamma

Goodness-of-Fit Tests

- Here are the likelihood ratio test values for the comparisons to the generalized gamma:

Comparison	LRT	P-value	Conclusion	Winner
Exponential vs. Generalized Gamma	12.9	0.0016	Gamma > Exponential	Gamma
Weibull vs. Generalized Gamma	0.00	1.00	Gamma = Weibull	Weibull
Log-Normal vs. Generalized Gamma	6.62	0.0101	Gamma > Log-Normal	Gamma

Goodness-of-Fit Tests – SAS

```
data GOF;
   Exp = -686.37;
   Weib = -679.92;
   LNorm = -683.23;
   GGam = -679.92;

   LRT1 = -2*(Exp - GGam);
   LRT2 = -2*(Weib - GGam);
   LRT3 = -2*(LNorm - GGam);

   P_Value1 = 1 - probchi(LRT1,2);
   P_Value2 = 1 - probchi(LRT2,1);
   P_Value3 = 1 - probchi(LRT3,1);
run;

proc print data=GOF;
       var LRT1-LRT3 P_Value1-P_Value3;
run;
```

Goodness-of-Fit Tests – SAS

```
data GOF;
   Exp = -686.37;
   Weib = -679.92;
   LNorm = -683.23;
   GGam = -679.92;

   LRT1 = -2*(Exp - GGam);
   LRT2 = -2*(Weib - GGam);
   LRT3 = -2*(LNorm - GGam);

   P_Value1 = 1 - probchi(LRT1,2);
   P_Value2 = 1 - probchi(LRT2,1);
   P_Value3 = 1 - probchi(LRT3,1);
run;

proc print data=GOF;
       var LRT1-LRT3 P_Value1-P_Value3;
run;
```

How did I get d.f.?

Goodness-of-Fit Tests – R

```
like.e <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                data = recid, dist = "exp")$loglik
like.w <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                data = recid, dist = "weibull")$loglik
like.ln <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                data = recid, dist = "lnorm")$loglik
like.g <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                data = recid, dist = "gamma")$loglik
like.ll <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                data = recid, dist = "llogis")$loglik
like.f <- flexsurvreg(Surv(week, arrest == 1) ~
                        fin + age + race + wexp + mar + paro + prio,
                        data = recid, dist = "genf")$loglik
```

Goodness-of-Fit Tests – R

```
pval.e.g <- 1 - pchisq((-2*(like.e-like.g)), 2)
pval.w.g <- 1 - pchisq((-2*(like.w-like.g)), 1)
pval.ln.g <- 1 - pchisq((-2*(like.ln-like.g)), 1)
pval.g.f <- 1 - pchisq((-2*(like.g-like.f)), 1)
pval.ll.f <- 1 - pchisq((-2*(like.ll-like.f)), 1)

Tests <- c('Exp vs. Gam', 'Wei vs. Gam', 'LogN vs. Gam', 'Gam vs. F',
           'LogL vs. F')

P_values <- c(pval.e.g, pval.w.g, pval.ln.g, pval.g.f, pval.ll.f)

cbind(Tests, P_values)
```

Goodness-of-Fit Tests – R

```
##      Tests             P_values
## [1,] "Exp vs. Gam"   "0.00172559564523367"
## [2,] "Wei vs. Gam"   "1"
## [3,] "LogN vs. Gam" "0.0110221983305441"
## [4,] "Gam vs. F"     "0.10886091147540 2"
## [5,] "LogL vs. F"    "0.11827642224585 3"
```

?

PREDICTING SURVIVAL & EVENT TIMES

Making Predictions

- AFT models assume a distribution for *T*, meaning that we expect event times to behave in a certain way.

- **IF WE ASSUME CORRECT DISTRIBUTION** we can predict quantiles, survival probabilities, event times, survival curves, and changes in expected values as predictor variable values change.

Example Predictions

- Median survival time:
  - Find *t* such that $\hat{S}_i(t) = 0.5$
- The time by which *q%* of people with the same values for predictor variables have the event:
  - Find *t* such that $\hat{S}_i(t) = 1 - q$
- 20 week predicted survival probability:
  - $\hat{S}_i(20)$

- **CAREFUL:** $\hat{S}_i(t)$ is entirely determined by the distribution used so estimates WON'T be the same across different distributions.

Predicted Survival Quantiles – SAS

```
proc lifereg data=Survival.Recid outest=Beta;
   model Week*arrest(0) = fin age prio / dist=weibull;
   output out=recid_q p=quan quantile=(0.25, 0.50, 0.75)
          std_err=se;
run;

proc print data=recid_q;
       var week _PROB_ quan se;
run;
```

Predicted Survival Quantiles – SAS

Obs	week	_PROB_	quan	se
1	20	0.25	52.688	5.824
2	20	0.50	98.728	12.518
3	20	0.75	161.958	24.851
4	17	0.25	24.180	3.553
5	17	0.50	45.308	6.206
6	17	0.75	74.325	10.759
7	25	0.25	17.891	3.917
8	25	0.50	33.524	6.933
9	25	0.75	54.994	11.420
10	52	0.25	64.227	7.687
11	52	0.50	120.349	16.907
12	52	0.75	197.427	33.286
13	52	0.25	35.955	3.831
14	52	0.50	67.372	7.297
15	52	0.75	110.521	14.179

⋮

Predicted Survival Probability at *t* – SAS

```
proc lifereg data=Survival.Recid outest=Beta;
   model Week*arrest(0) = fin age prio / dist=weibull;
   output out=recid_s xbeta=lp cdf=cdistfunc;
run;

data recid_s;
   set recid_s;
   survprob = 1 - cdistfunc;
run;

proc print data=recid_s;
   var week survprob;
run;
```

Predicted Survival Probability at *t* – SAS

$\hat{S}_i(\text{week})$

Obs	week	survprob
1	20	0.92858
2	17	0.83891
3	25	0.63152
4	52	0.80732
5	52	0.61736
6	52	0.73121
7	23	0.92604
8	52	0.72034
9	52	0.58915
10	52	0.71430
11	52	0.80822
12	52	0.89821

⋮

Predicted Survival Probability at *t* – SAS

```
%predict(outest=Beta, out=recid_s, xbeta=lp, time=10);

proc print data=_PRED_;
   var week survprob t prob;
run;
```

Predicted Survival Probability at *t* – SAS

$\hat{S}_i(\text{week})$ $\hat{S}_i(10)$

Obs	week	survprob	t	prob
1	20	0.92858	10	0.97232
2	17	0.83891	10	0.91985
3	25	0.63152	10	0.88039
4	52	0.80732	10	0.97895
5	52	0.61736	10	0.95320
6	52	0.73121	10	0.96937
7	23	0.92604	10	0.97635
8	52	0.72034	10	0.96792
9	52	0.58915	10	0.94878
10	52	0.71430	10	0.96711
11	52	0.80822	10	0.97906
12	52	0.89821	10	0.98939

⋮

Predicted Change in Event Time – SAS

```
proc lifereg data=Survival.Recid outest=Beta;
   model Week*arrest(0) = fin age prio / dist=weibull;
   output out=recid_e xbeta=lp cdf=cdistfunc;
run;

data _null_;
   set Beta;
   call symput('shape', 1/_SCALE_);
   call symput('beta_fin', fin);
run;
```

Predicted Change in Event Time – SAS

```
data recid_e;
   set recid_e;
   if arrest = 0 then delete;
   if fin = 1 then delete;
   survprob = 1 - cdistfunc;

   lp_new = lp + &beta_fin;

   newtime = squantile('weibull', survprob, &shape, exp(lp_new));
   diff = newtime - week;
run;

proc print data=recid_e;
        var week survprob lp lp_new newtime diff;
run;
```

Predicted Change in Event Time – SAS

Obs	survprob	lp	lp_new	newtime	week	diff
1	0.92858	4.85409	5.10359	25.6678	20	5.6678
2	0.83891	4.07520	4.32470	21.8176	17	4.8176
3	0.63152	3.77398	4.02349	32.0847	25	7.0847
4	0.92604	4.96795	5.21745	29.5179	23	6.5179
5	0.65952	4.23682	4.48633	47.4854	37	10.4854
6	0.85065	4.51972	4.76922	32.0847	25	7.0847
7	0.74107	4.19276	4.44227	35.9349	28	7.9349
8	0.88437	3.79972	4.04922	12.8339	10	2.8339
9	0.96906	4.26256	4.51207	7.7003	6	1.7003
10	0.71555	4.73282	4.98232	66.7362	52	14.7362
11	0.85883	5.23623	5.48573	62.8860	49	13.8860
12	0.73207	4.59322	4.84273	55.1857	43	12.1857

⋮

Predicted Survival Quantiles – R

```
recid.aft.w <- survreg(Surv(week, arrest == 1) ~
                          fin + age + prio, data = recid,
                          dist = 'weibull')

survprob.75.50.25 <- predict(recid.aft.w, type = "quantile",
                                   se.fit = TRUE,
                               p = c(0.25, 0.5, 0.75))

head(survprob.75.50.25$fit)
```

```
##          [,1]      [,2]      [,3]
## [1,] 52.68849  98.72758 161.95827
## [2,] 24.17956  45.30760  74.32514
## [3,] 17.89085  33.52383  54.99438
## [4,] 64.22717 120.34873 197.42682
## [5,] 35.95471  67.37185 110.52057
## [6,] 48.95457  91.73097 150.48064
```

Predicted (Mean) Event Times – R

```
p.time.mean <- predict(recid.aft.w, type = "response",
                       se.fit = TRUE)

head(p.time.mean$fit, n = 10)
```

```
##  [1] 128.26394  58.86229  43.55317 156.35349  87.52751
       119.17415 143.73152
##  [8] 115.26040  81.92984 113.19494
```

Predicted Survival Probability at *t* – R

```
survprob.actual <- 1 - psurvreg(recid$week,
                                mean = predict(recid.aft.w,
                                type = "lp"),
                                scale = recid.aft.w$scale,
                                distribution = recid.aft.w$dist)

head(survprob.actual, n = 10)
```

```
##  [1] 0.9285822 0.8389085 0.6315234 0.8073231 0.6173609
       0.7312118 0.9260438
##  [8] 0.7203354 0.5891529 0.7143008
```

Predicted Survival Probability at *t* – R

```
survprob.10wk <- 1 - psurvreg(10,
                              mean = predict(recid.aft.w,
                                             type = "lp"),
                              scale = recid.aft.w$scale,
                              distribution = recid.aft.w$dist)

head(survprob.10wk)
```

```
## [1] 0.9723202 0.9198457 0.8803901 0.9789527 0.9531961 0.9693657
```

Predicted Change in Event Time – R

```
new_time <-  qsurvreg(1 - survprob.actual,
                      mean = predict(recid.aft.w, type = "lp") +
                      coef(recid.aft.w)['fin'],
                      scale = recid.aft.w$scale,
                      distribution = recid.aft.w$dist)

recid$new_time <- new_time
recid$diff <- recid$new_time - recid$week

head(data.frame(recid$week, recid$new_time, recid$diff))
```

```
##   recid.week recid.new_time recid.diff
## 1         20       25.66776   5.667764
## 2         17       21.81760   4.817600
## 3         25       32.08471   7.084706
## 4         52       66.73619  14.736188
## 5         52       66.73619  14.736188
## 6         52       66.73619  14.736188
```