Carleton University
School of Mathematics and Statistics

Technical Report of a Survival Analysis Using a Parametric Accelerated Failure Time Model

Hua Ye

October 2009

In this paper: 1) run a complete data analysis using a parametric Accelerated Failure Time model; 2) select the appropriate log-location-scale family for the survival time T_i; 3) assess the fit and perform inference on the selected model. Software for analyzing is R.

Contents

Purpose

1) Run a complete data analysis using a parametric Accelerated Failure Time (AFT) model;
2) Select the appropriate log-location-scale family for the survival time T_i;
3) Assess the fit and perform inference.

Data description

The data consists of 137 patients who participated in a clinical trial for two treatment regimens for lung cancer.

We consider the data set from a study designed to assess the effect of a new treatment on the survival time of patients with lung cancer. The *TIME* variable contains survival time in days of after a treatment. The variable *STATUS* has a value of 1 for those events at time, and has a value of 0 for those right censored.

The covariates included in the analyses are:

(i) *trt*: 1=standard treatment, 2=new treatment;
(ii) *ctype*: cell type, 1=squamous, 2=smallcell, 3=adeno, 4=large;
(iii) *dtime*: days from diagnosis to randomization;
(iv) *age*: in years at the time of a treatment;
(v) *prior*: prior therapy, 0=no, 1=yes.

1 Import data into R

Before starting the data analysis, we need to load the *survival* library in R. We can do this by running *library(survival)*.

The data is stored in the text file *stat_5603-a2d1.txt*. To import the data set I use the command:

```
> a<-read.table("m:/stat_5603-a2d1.txt", header=T)
```

The *header=T* option tells R that the variable names are stored in the first row of the data set.

2 Apply a parametric AFT model

One of the interests of survival analysis is to understand the relationship between time to failure and other covariates measured at the studied subjects. This can be done by using parametric regression models.

2.1 AFT model

Let T_i be a random variable denoting the failure time for the *ith* subject, and let x_{i1}, x_{i2}, …, x_{ip} be the values of p covariates for that same subject. An AFT model is then

$$\log T_i = b_0 + b_1 x_{i1} + ... + b_p x_{ip} + s e_i$$

where e_i is a random disturbance term, and $b_0, ..., b_p$ and s are parameters to be estimated. As for the natural log transformation of T_i, many popular survival distributions *T's* have the property $Y = logT$ where Y is from a location-scale family, and this transformation also ensures that predicted values of T are positive.

In other words, the AFT model can be specified as: $\log T_i = X_i^T \underline{b} + s e_i$

If there are no censored data, we can readily estimate this model by Ordinary Least Squares (OLS). Simply generate a new variable, $Y = logT$, and use the linear regression model with Y as the dependent variable. Survival data, however, typically have at least some censored observations, and these are difficult to handle with OLS[1]. Alternatively, we can use Maximum Likelihood Estimation (MLE) with different distribution assumption on e . For each of the distribution of ε, there is a corresponding distribution for T.

Distribution of T	**Distribution of** $e = \frac{Y-m}{s} = \frac{\log T - m}{s} = \frac{\log T - X^T \underline{b}}{s}$
Weibull	Extreme value (2 parameters)
Exponential	Extreme value (1 parameter)
Gamma	Log-gamma
Log-logistic	Logistic
Log-normal	Normal

Table 1 Corresponding distributions

Note that the AFT model is named for the distribution of T rather than the distribution of ε or $logT$.

[1] Allison, P. 1995. Survival Analysis Using the SAS System: A Practical Guide. SAS Publishing

2.2 Use R to select and fit the AFT model

The *survreg()* function in R produces the estimates of parametric regression models with censored survival data using the method of maximum likelihood.

The output of *survreg()* function gives the estimated regression coefficients and their standard errors.

- The value of *scale* refers to the estimate of s .
- The values of *(Intercept), trt, dtime, age, prior, factor(ctype)2, factor(ctype)3,* and *factor(ctype)4* refer to the estimates of $b_0,...,b_p$.
- It also gives loglik(MLE) for the full and baseline models.

Note that the covariate *ctype* (cell type) is categorical and we let R treat it as *factor(ctype)*.

The outputs of *survreg()* function in R are presented as follows:

(i) Lognormal distribution

```
> lognormalreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="lognormal")
> summary(lognormalreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "lognormal")
                 Value Std. Error       z        p
(Intercept)     4.38092    0.47326  9.2569 2.10e-20
trt            -0.13812    0.14528 -0.9507 3.42e-01
dtime          -0.00851    0.00735 -1.1574 2.47e-01
age             0.00949    0.00679  1.3968 1.62e-01
prior          -0.39952    0.17339 -2.3042 2.12e-02
factor(ctype)2 -0.76868    0.18869 -4.0738 4.63e-05
factor(ctype)3 -0.83456    0.21617 -3.8608 1.13e-04
factor(ctype)4  0.00824    0.21388  0.0385 9.69e-01
Log(scale)     -0.20538    0.06297 -3.2618 1.11e-03

Scale= 0.814

Log Normal distribution
Loglik(model)= -679   Loglik(intercept only)= -695.8
        Chisq= 33.44 on 7 degrees of freedom, p= 2.2e-05
Number of Newton-Raphson Iterations: 4
n= 137
```

(ii) Weibull distribution

```
> weibullreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="weibull")
> summary(weibullreg)
```

```
Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "weibull")
                  Value Std. Error      z        p
(Intercept)     4.74873    0.47248 10.051 9.12e-24
trt             0.02747    0.14091  0.195 8.45e-01
dtime          -0.01185    0.00669 -1.773 7.63e-02
age             0.00623    0.00624  0.999 3.18e-01
prior          -0.34389    0.16119 -2.133 3.29e-02
factor(ctype)2 -0.71940    0.18555 -3.877 1.06e-04
factor(ctype)3 -0.84020    0.21051 -3.991 6.57e-05
factor(ctype)4 -0.17067    0.20307 -0.840 4.01e-01
Log(scale)     -0.30203    0.06756 -4.470 7.81e-06

Scale= 0.74

Weibull distribution
Loglik(model)= -685.1   Loglik(intercept only)= -699.1
        Chisq= 28.05 on 7 degrees of freedom, p= 0.00022
Number of Newton-Raphson Iterations: 5
n= 137
```

(iii) Exponential distribution

```
> exponentialreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="exponential")
> summary(exponentialreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "exponential")
                  Value Std. Error       z        p
(Intercept)     4.74052    0.62521  7.5823 3.39e-14
trt            -0.01167    0.18703 -0.0624 9.50e-01
dtime          -0.01199    0.00887 -1.3522 1.76e-01
age             0.00639    0.00843  0.7573 4.49e-01
prior          -0.34060    0.21793 -1.5629 1.18e-01
factor(ctype)2 -0.74161    0.24636 -3.0103 2.61e-03
factor(ctype)3 -0.87388    0.27973 -3.1240 1.78e-03
factor(ctype)4 -0.16525    0.27240 -0.6066 5.44e-01

Scale fixed at 1

Exponential distribution
Loglik(model)= -693.7   Loglik(intercept only)= -703
        Chisq= 18.66 on 7 degrees of freedom, p= 0.0093
Number of Newton-Raphson Iterations: 4
n= 137
```

(iv) Gaussian distribution

```
> gaussianreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="gaussian")
> summary(gaussianreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "gaussian")
                Value Std. Error      z        p
(Intercept)     80.499    38.5887  2.086 0.036971
trt             -2.084    11.8533 -0.176 0.860438
dtime           -0.473     0.5991 -0.789 0.429839
age              0.944     0.5547  1.702 0.088723
prior          -24.196    14.1289 -1.713 0.086802
factor(ctype)2 -58.126    15.4088 -3.772 0.000162
factor(ctype)3 -63.493    17.6481 -3.598 0.000321
factor(ctype)4  -7.925    17.4347 -0.455 0.649412
Log(scale)       4.195     0.0628 66.783 0.000000

Scale= 66.4

Gaussian distribution
Loglik(model)= -726   Loglik(intercept only)= -738.8
        Chisq= 25.6 on 7 degrees of freedom, p= 0.00059
Number of Newton-Raphson Iterations: 4
n= 137
```

(v) Logistic distribution

```
> logisticreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="logistic")
> summary(logisticreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "logistic")
                Value Std. Error      z        p
(Intercept)     69.715    33.6410  2.072 0.038236
trt             -8.297    10.4606 -0.793 0.427696
dtime           -0.568     0.5003 -1.135 0.256387
age              1.000     0.4842  2.065 0.038946
prior          -19.682    12.3729 -1.591 0.111670
factor(ctype)2 -47.398    13.9527 -3.397 0.000681
factor(ctype)3 -54.061    15.6453 -3.455 0.000549
factor(ctype)4  -2.299    15.9405 -0.144 0.885345
Log(scale)       3.547     0.0753 47.107 0.000000

Scale= 34.7

Logistic distribution
Loglik(model)= -720.1   Loglik(intercept only)= -733.7
        Chisq= 27.18 on 7 degrees of freedom, p= 0.00031
Number of Newton-Raphson Iterations: 4
n= 137
```

(vi) Loglogistic distribution

```
> loglogisticreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="loglogistic")
> summary(loglogisticreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "loglogistic")
                Value Std. Error       z        p
(Intercept)     4.2612    0.47967   8.884 6.47e-19
trt            -0.1533    0.14816  -1.035 3.01e-01
dtime          -0.0099    0.00697  -1.420 1.56e-01
age             0.0123    0.00689   1.781 7.50e-02
prior          -0.3713    0.17998  -2.063 3.91e-02
factor(ctype)2 -0.7796    0.19479  -4.002 6.28e-05
factor(ctype)3 -0.8755    0.22339  -3.919 8.88e-05
factor(ctype)4 -0.0284    0.21006  -0.135 8.93e-01
Log(scale)     -0.7396    0.07253 -10.198 2.03e-24

Scale= 0.477

Log logistic distribution
Loglik(model)= -681.5   Loglik(intercept only)= -699.4
        Chisq= 35.77 on 7 degrees of freedom, p= 8e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

3 Model selection

3.1 Look for models with graphic methods

Graphical methods are useful for summarizing information and suggesting possible Models. They also provide ways to check assumptions concerning the form of a lifetime distribution and its relationship to covariates[2].

Use R, we find that the most possible fitted more may be Weibull and log normal. More over, it looks like **the log normal model** is a little bit better than **the Weibull model** and the log-logistic model.

3.1.1 Weibull lifetime model

We see that, plots of $\log(-\log(\hat{S}_j(t))$ versus *log(t)* are roughly linear, then Weibull lifetime model is suggested.

[2] LAWLESS, J. 2002. Statistical Models for Lifetime Data and Methods. Wiley-Interscience.

```
> fit.km=survfit(Surv(time,status)~1,data=a,conf.type="none")
> plot(log(fit.km$time[1:360]),log(-log(fit.km$surv[1:360])), ylab='log(-
log(K-M estimate)', xlab='log(t)', cex.main=0.8,cex.lab=0.8, cex.axis=0.7)
```

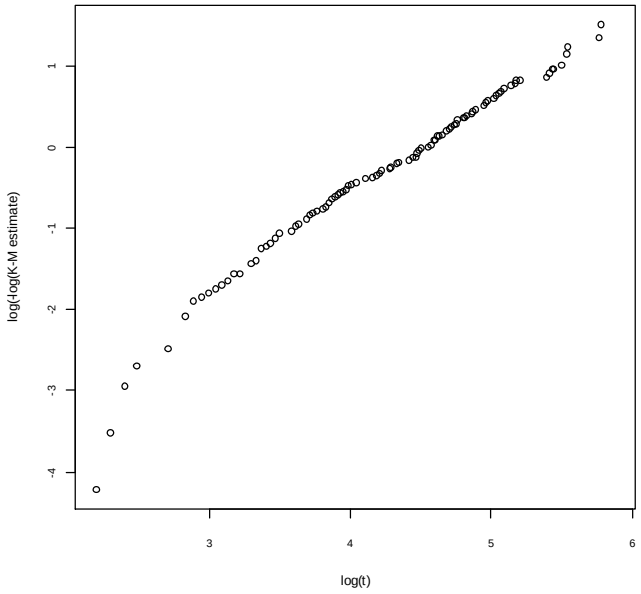

Figure 1 Weibull probability plots for the whole data

```
> fit.km.bytrt=survfit(Surv(time,status)~trt,data=a,conf.type="none")
> plot(log(fit.km.bytrt$time[1:360]), log(-
log(fit.km.bytrt$surv[1:360])),pch=1:2, ylab='log(-log(K-M estimate)',
xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -3, cex=.8, pch=1:2,c("standard treatment"," new treatment"))
```

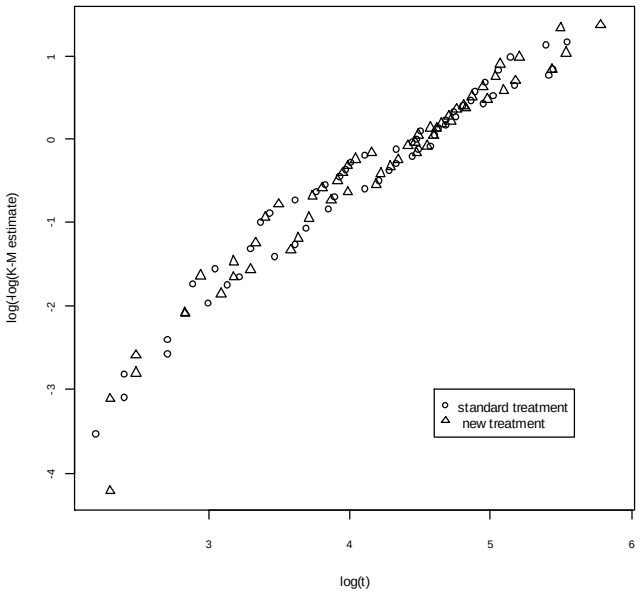

Figure 2 Weibull probability plots for the data with different treatment

```
> fit.km.byctype=survfit(Surv(time,status)~ctype,data=a,conf.type="none")
> plot(log(fit.km.byctype$time[1:360]), log(-
log(fit.km.byctype$surv[1:360])),pch =1:4, ylab='log(-log(K-M estimate)',
xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -2, cex=.8, pch =1:4,c("squamous","smallcell","adeno","large"))
```

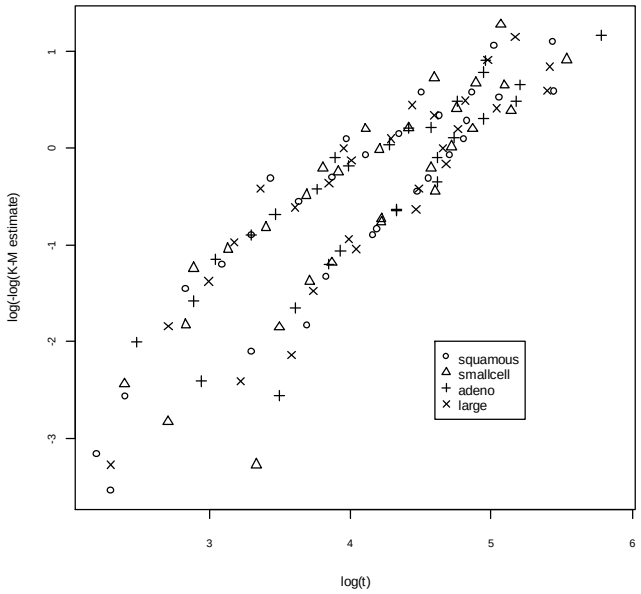

Figure 3 Weibull probability plots for the data with different cell type

3.1.2 Log normal lifetime model

We see that, plots of $qnorm(1-\hat{S}_j(t))$ versus *log(t)* are roughly linear, then log normal lifetime model is suggested.

```
> fit.km=survfit(Surv(time,status)~1,data=a,conf.type="none")
> plot(log(fit.km$time[1:360]), qnorm(1-fit.km$surv[1:360]), ylab='qnorm(1-
K-M extimate)', xlab='log(t)', cex.main=0.8,cex.lab=0.8, cex.axis=0.7)
```

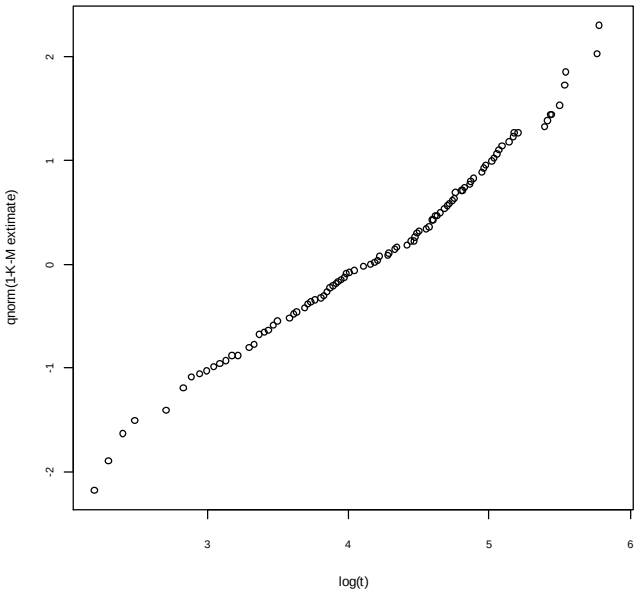

Figure 4 Log normal probability plots for the whole data

```
> fit.km.bytrt=survfit(Surv(time,status)~trt,data=a,conf.type="none")
> plot(log(fit.km.bytrt$time[1:360]), qnorm(1-
fit.km.bytrt$surv[1:360]),pch=1:2, ylab='qnorm(1-K-M extimate)',
xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -1, cex=.8, pch=1:2,c("standard treatment"," new treatment"))
```

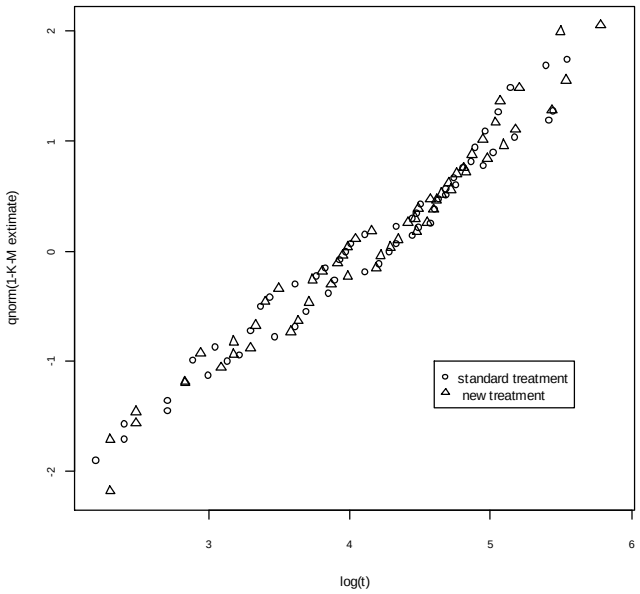

Figure 5 Log normal probability plots for the data with different treatment

```
> fit.km.byctype=survfit(Surv(time,status)~ctype,data=a,conf.type="none")
> plot(log(fit.km.byctype$time[1:360]), qnorm(1-
fit.km.byctype$surv[1:360]),pch=1:4, ylab='qnorm(1-K-M extimate)',
xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -1, cex=.8, pch =1:4,c("squamous","smallcell","adeno","large"))
```

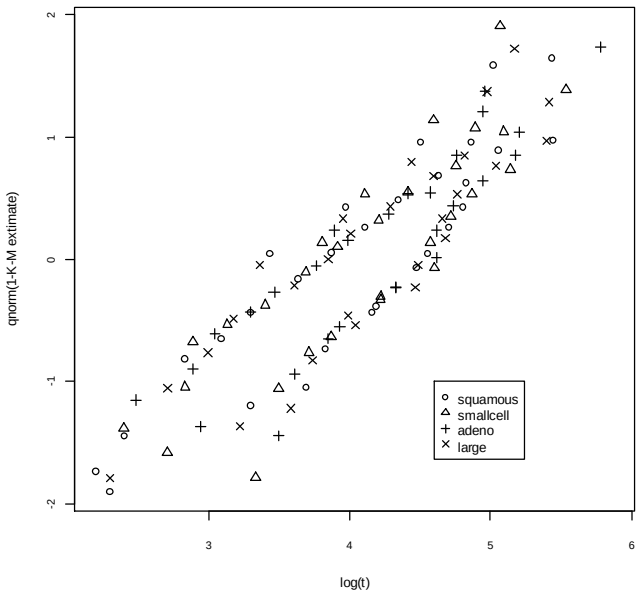

Figure 6 Log normal probability plots for the data with different cell type

3.1.3 Log-logistic lifetime model

We see that, plots of $\log((1-\hat{S}_j(t)/\hat{S}_j(t))$ versus *log(t)* are roughly linear, and its shape is like the one of the log normal time model. Then log normal lifetime model is considered.

```
> fit.km=survfit(Surv(time,status)~1,data=a,conf.type="none")
> plot(log(fit.km$time[1:360]),log((1-fit.km$surv[1:360])/
fit.km$surv[1:360]), ylab='log((1-K-M estimate)/K-M estimate)',
xlab='log(t)', cex.main=0.8,cex.lab=0.8, cex.axis=0.7)
```

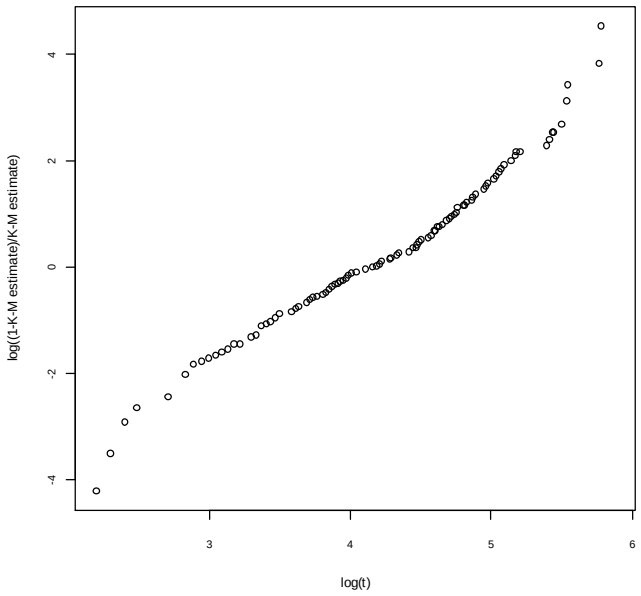

Figure 7 Log-logistic probability plots for the whole data

```
> fit.km.bytrt=survfit(Surv(time,status)~trt,data=a,conf.type="none")
> plot(log(fit.km.bytrt$time[1:360]), log((1-fit.km.bytrt
$surv[1:360])/fit.km.bytrt$surv[1:360]),pch=1:2, ylab='log((1-K-M
estimate)/K-M estimate)', xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -1, cex=.8, pch=1:2,c("standard treatment"," new treatment"))
```

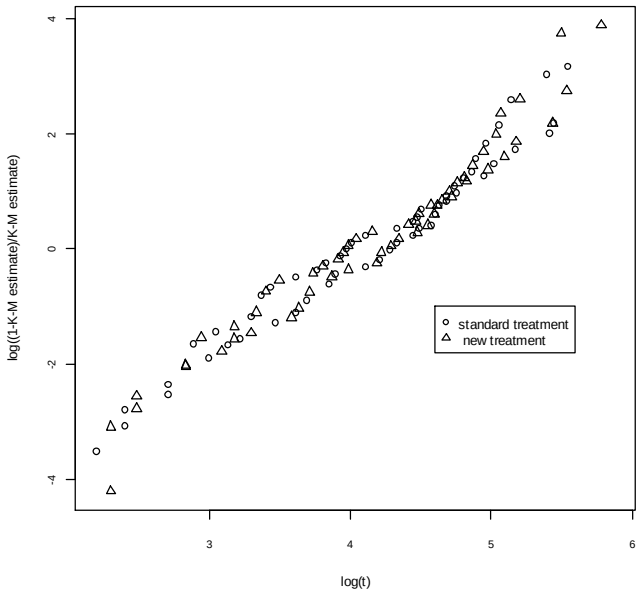

Figure 8 Log-logistic probability plots for the data with different treatment

```
> fit.km.byctype=survfit(Surv(time,status)~ctype,data=a,conf.type="none")
> plot(log(fit.km.byctype$time[1:360]), log((1-
fit.km.byctype$surv[1:360])/fit.km.byctype$surv[1:360]),pch=1:4, ylab='
log((1-K-M estimate)/K-M estimate)', xlab='log(t)', cex.lab=0.8,cex.axis=0.7)
> legend(4.6, -1, cex=.8, pch =1:4,c("squamous","smallcell","adeno","large"))
```

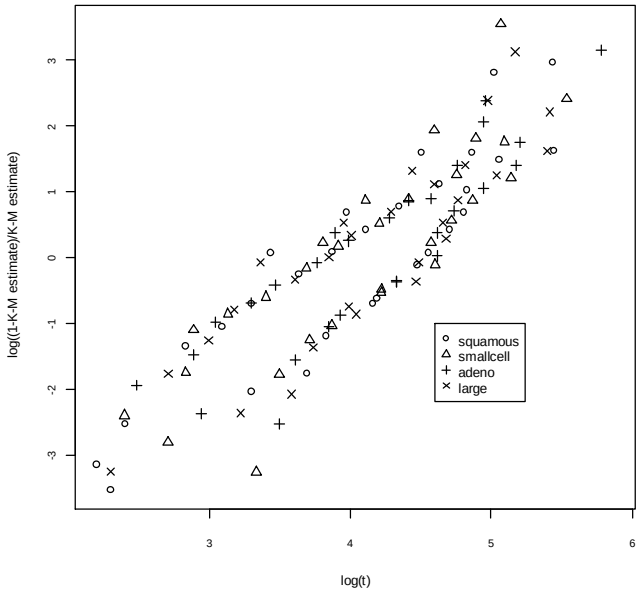

Figure 9 Log-logistic probability plots for the data with different cell type

3.2 Akaike Information Criterion

One method for comparing models is to use the Akaike Information Criterion[3] (AIC). This is defined as: *AIC = -2logL + 2p.*

Here *p* is the number of parameters fit in the model.

While the *survreg()* function does not directly give us this value, we can obtain it through the information we are given:

The ANOVA[4] function provides -2*loglik(MLE)* for each model, which can also be obtained from the summaries of R.

```
> anova(lognormalreg,weibullreg,exponentialreg,gaussianreg,logisticreg,
loglogisticreg,test="Chisq")
                                  Terms Resid. Df    -2*LL Test Df  Deviance    P(>|Chi|)
1 trt + dtime + age + prior + factor(ctype)       128 1358.093      NA        NA           NA
2 trt + dtime + age + prior + factor(ctype)       128 1370.195    =  0 -12.10269           NA
3 trt + dtime + age + prior + factor(ctype)       129 1387.375    = -1 -17.18013 3.399733e-05
4 trt + dtime + age + prior + factor(ctype)       128 1452.022    =  1 -64.64613           NA
5 trt + dtime + age + prior + factor(ctype)       128 1440.144    =  0  11.87803           NA
6 trt + dtime + age + prior + factor(ctype)       128 1363.063    =  0  77.08073           NA
```

[3] Model Selection, http://www.webpages.uidaho.edu/~brian/stat401ch13_01.pdf.
[4] Parametric Regression in Survival Analysis, http://www.stat.ncu.edu.tw/teacher/Tsengyk/Handout2a.htm.

In this case, *p*'s of different models are almost the same. From the above information, we know that the **log normal** model yields the smallest AIC.

Thus, the **log normal** model may fit the data better than other models.

3.3 Likelihood-ratio test

The approach of AIC is often used in model selection while the models are not nested. However, AIC does not consider the sample size. It might have good properties for smaller sample size *n*. On the other hand, AIC might suggest a non-reasonable model[5]. Moreover, picking the model with the smallest AIC requires the model in the suite with the best overall statistical properties and parameter balance[6].

Another method is the likelihood-ratio test for comparing nested models.

A model is said to be nested within another model if the first model is a special case of the second. More precisely, model A is nested within model B if A can be obtained by imposing restrictions on the parameters in B. For example, the exponential model is nested within both the Weibull and the Gamma models.

For example, in this case, we calculate the likelihood ratio test as:

H_0: Exponential model

H_1: Weibull model

The log-likelihood for Exponential model is -693.7 and the log-likelihood for Weibull model is -685.1. The likelihood-ratio Chi-square statistic is:

$$-2(l(\underline{q}_0) - l(\hat{\underline{q}}_n)) = -2(-693.7-(-685.1)) = 17.2.$$

Clearly, we can reject the null hypotheses, that is to say the Weibull model fits the data better than the Exponential model. By the likelihood ratio test, when the models are nested, the **Weibull** model may fit the data better.

[5] Chapter 6. Modelling Method for Survival Analysis, http://www.stat.nuk.edu.tw/wongkf_html/survival06.pdf.
[6] Model Selection, http://www.webpages.uidaho.edu/~brian/stat401ch13_01.pdf.

3.4 Model fit for selecting model

Model fit can be evaluated based upon a graphical comparison between empirical Kaplan–Meier (K-M)) survival curves and fitted or "predicted" survival curves generated from the selected AFT model[7].

First of all, the general Kaplan–Meier survival curve can be obtained in R:

```
> fit.km=survfit(Surv(time,status)~1,data=a,conf.type="none")
> plot(fit.km,main="Clinical trial for two treatment regimens for lung cancer",
ylab='K-M estimate', xlab='Time (days)', cex.main=0.8,cex.lab=0.7, cex.axis=0.7)
```

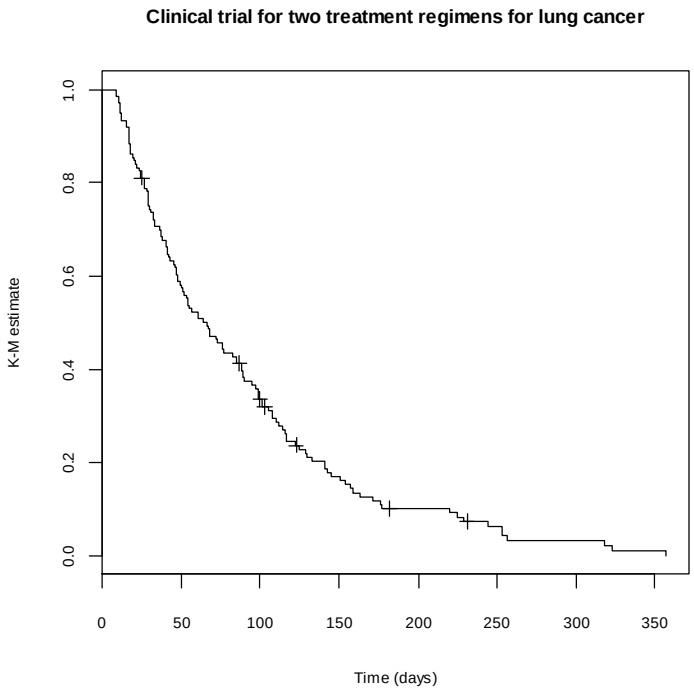

Figure 10 General K-M survival curve

3.4.1 Categorize the data in terms of treatment

When plotting K-M survival curves, we choose to categorize treatment (the variable is *trt: 1=standard treatment, 2=new treatment*) to conduct the graphical comparison.

```
> fit.km.bytrt=survfit(Surv(time,status)~trt,data=a,conf.type="none")
> plot(fit.km.bytrt,main="Clinical trial for two treatment regimens for lung
cancer by treatment",ylab='K-M estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,col=c("black","red"),lty=1:2)
> legend(230, .96,lty=1:2, cex=.7,col=c("black","red"),c("standard
treatment"," new treatment"))
```

[7] Swindell, W. 2009. Accelerated Failure Time Models Provide a Useful Statistical Framework for Aging Research, Experimental Gerontology 44 (2009): 190–200.

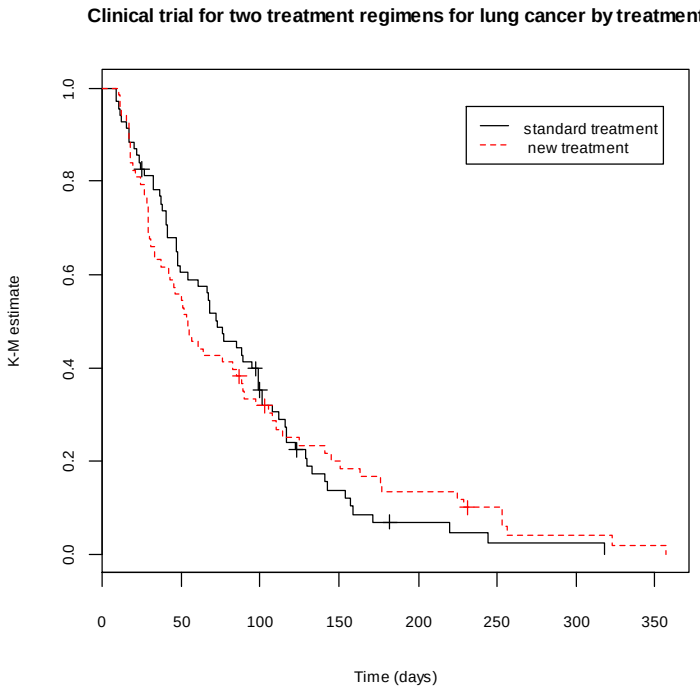

Figure 11 K-M survival curves by treatment

Then, add the fitted or "predicted" survival curves generated from the selected AFT model to compare.

- According to log-location-scale families, $S(t_i) = S_0(\frac{\log t_i - X_i^T \underline{b}}{s_i})$ where for the log normal regression $S_0(w) = 1 - \Phi(w)$ and $w = \frac{\log t_i - X_i^T \underline{b}}{s_i}$.
- According to log-location-scale families, for the Weibull distribution of *T*, the survival function $S(t_i) = S_0(\frac{\log t_i - X_i^T \underline{b}}{s_i})$ where $S_0(w) = \exp(-\exp(w))$.

Consider two patients with the following covariate combinations:

- X_1=(1, treatment = 1, days from diagnosis to randomization = 3, age = 60, prior = 0, cell type = 4)
- X_2=(1, treatment = 2, days from diagnosis to randomization = 3, age = 60, prior = 0, cell type = 4)

In R, they can be expressed as:

```
> fit.km.bytrt=survfit(Surv(time,status)~trt,data=a,conf.type="none")
> plot(fit.km.bytrt,main="Clinical trial for two treatment regimens for lung
cancer by treatment",ylab='Survival probability estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,lty=1:6)

> x1<-c(1,1,3,60,0,0,0,1)
> x2<-c(1,2,3,60,0,0,0,1)
> t <- seq(0,360,0.1)
>  sx1 <- 1-pnorm((log(t)-x1%*%lognormalreg$coeff)/lognormalreg$scale)
>  sx2 <- 1-pnorm((log(t)-x2%*%lognormalreg$coeff)/lognormalreg$scale)
> wx1 <- exp(-exp(log(t)-x1%*%weibullreg$coeff)/weibullreg$scale)
> wx2 <- exp(-exp(log(t)-x2%*%weibullreg$coeff)/weibullreg$scale)
> lines(t,sx1,lty=3)
> lines(t,sx2,lty=4)
> lines(t,wx1,lty=5)
> lines(t,wx2,lty=6)
> legend(230, .96,lty=1:6, cex=.7, c("K-M standard treatment","K-M
for new treatment", "X1 by log normal","X2 by log normal","X1 by
Weibull","X2 by Weibull"))
```

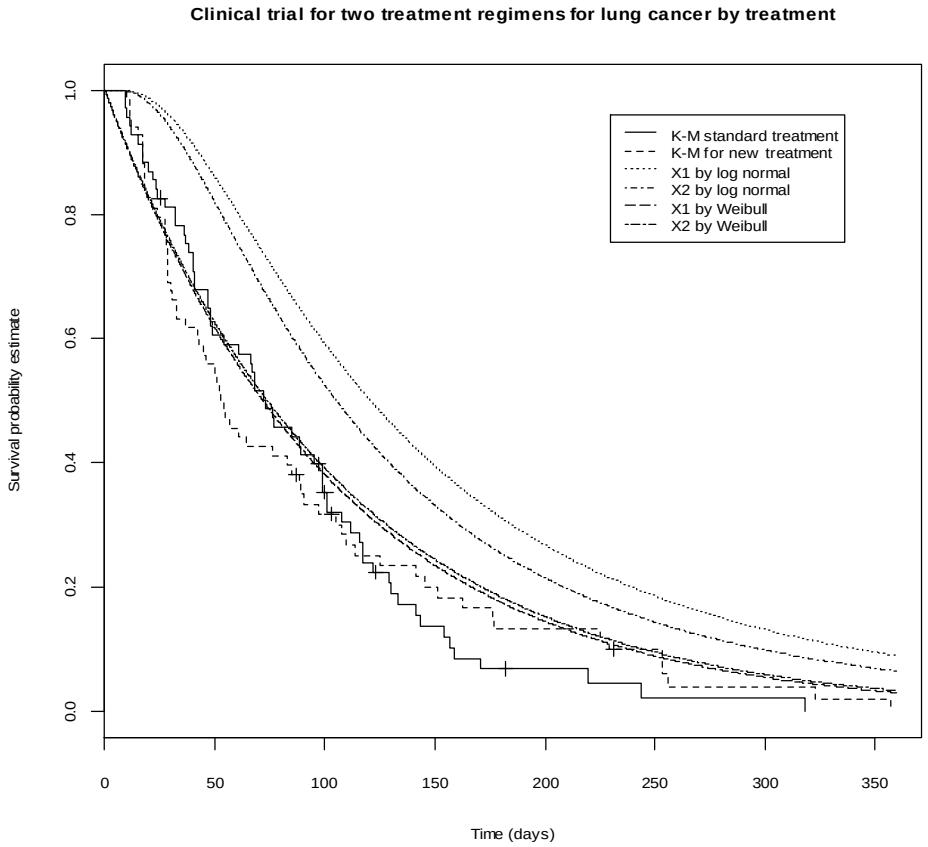

Figure 12 Graphical comparison between K-M curves and fitted curves by treatment

From the above graphical comparison, we can see that the **Weibull** regression model may fit the data **better** than the log normal regression model.

3.4.2 Categorize the data in terms of cell type

In order to conduct the graphical comparison, we choose to categorize cell type:

- The variable is *ctype: 1=squamous, 2=smallcell, 3=adeno, 4=large.*

```
> fit.km.byctype=survfit(Surv(time,status)~ctype,data=a,conf.type="none")
> plot(fit.km.byctype,main="Clinical trial for two treatment regimens for lung cancer
by cell type",ylab='K-M estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,col=c(1:4),lty=1:4)
> legend(260, .96,lty=1:4, cex=.7,col=c(1:4),c("squamous","smallcell","adeno","large"))
```

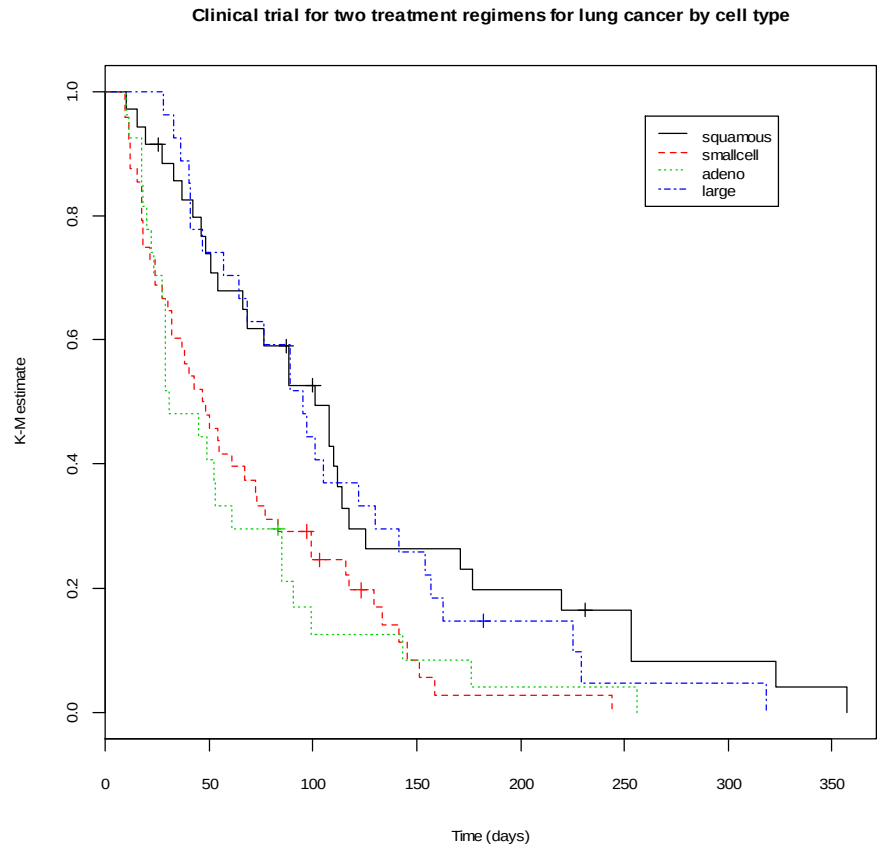

Figure 13 K-M survival curves by cell type

We can use the data of two other patients with the following covariate combinations:

- X_3=(1, treatment = 1, days from diagnosis to randomization = 10, age = 50, prior = 0, cell type = 3)
- X_4=(1, treatment = 2, days from diagnosis to randomization = 10, age = 50, prior = 0, cell type = 3)

In R, it can be expressed as:

```
> fit.km.byctype=survfit(Surv(time,status)~ctype,data=a,conf.type="none")
> plot(fit.km.byctype,main="Clinical trial for two treatment regimens for
lung cancer by cell type",ylab='K-M estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7, lty=1:8)

> x3<-c(1,1,10,50,0,0,1,0)
> x4<-c(1,2,10,50,0,0,1,0)
> t <- seq(0,360,0.1)
> sx3 <- 1-pnorm((log(t)-x3%*%lognormalreg$coeff)/lognormalreg$scale)
> sx4 <- 1-pnorm((log(t)-x4%*%lognormalreg$coeff)/lognormalreg$scale)
> wx3 <- exp(-exp(log(t)-x3%*%weibullreg$coeff)/weibullreg$scale)
> wx4 <- exp(-exp(log(t)-x4%*%weibullreg$coeff)/weibullreg$scale)
> lines(t,sx3,lty=5)
> lines(t,sx4,lty=6)
> lines(t,wx3,lty=7)
> lines(t,wx4,lty=8)
> legend(230, .96,lty=1:8, cex=.7, c("K-M for squamous","K-M for
smallcell","K-M for adeno","K-M for large", "X3 by log normal","X4 by log
normal","X3 by Weibull","X4 by Weibull"))
```

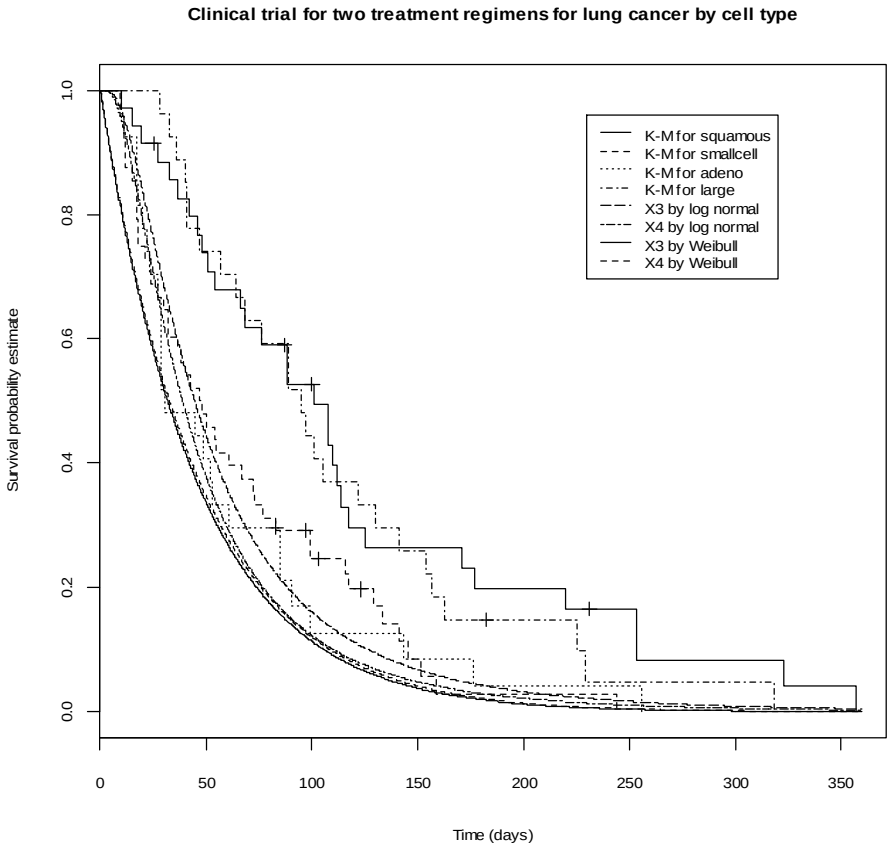

Figure 14 Graphical comparison between K-M curves and fitted curves by cell type

From the above graphical comparison, we can see that the **log normal** regression model may fit the data **better** than the Weibull regression model.

3.4.3 Result of iteration

After repeating several rounds of the graphical comparison, the **log normal** regression AFT model is selected for the data analysis in this case.

3.5 Residual analysis

(*I skip it* ☺.)

4 Fit the log normal regression AFT model

4.1 Variable selection

It is necessary to determine which variables should be included in the fitted AFT model. Variable selection is performed using **a forward and backward stepwise procedure**[8] that searches all possible models to determine which model minimized the AIC:

$$AIC = -2logL + 2p.$$

In this case, the selecting criterion is based on the ***z* statistics** associated with estimated regression parameters, which are equivalent to the common **chi-square wald test statistics**[9].

1) To begin, we fit the full model with all covariates included.

```
> lognormalreg<-
survreg(Surv(time,status)~trt+dtime+age+prior+factor(ctype),data=a,d
ist="lognormal")
> summary(lognormalreg)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    factor(ctype), data = a, dist = "lognormal")
                Value Std. Error       z        p
(Intercept)     4.38092    0.47326  9.2569 2.10e-20
```

[8] Swindell, W. 2009. Accelerated Failure Time Models Provide a Useful Statistical Framework for Aging Research, Experimental Gerontology 44 (2009): 190–200.
[9] Parametric Regression in Survival Analysis, http://www.stat.ncu.edu.tw/teacher/Tsengyk/Handout2a.htm.

```
trt             -0.13812    0.14528 -0.9507 3.42e-01
dtime           -0.00851    0.00735 -1.1574 2.47e-01
age              0.00949    0.00679  1.3968 1.62e-01
prior           -0.39952    0.17339 -2.3042 2.12e-02
factor(ctype)2  -0.76868    0.18869 -4.0738 4.63e-05
factor(ctype)3  -0.83456    0.21617 -3.8608 1.13e-04
factor(ctype)4   0.00824    0.21388  0.0385 9.69e-01
Log(scale)      -0.20538    0.06297 -3.2618 1.11e-03

Scale= 0.814

Log Normal distribution
Loglik(model)= -679   Loglik(intercept only)= -695.8
        Chisq= 33.44 on 7 degrees of freedom, p= 2.2e-05
Number of Newton-Raphson Iterations: 4
n= 137
```

2) Since the variable *factor(ctype)4 (cell type is large)* is associated with the least significant *z* statistic and the largest *p* value, it is excluded from the model. This gives the following fit:

```
> attach(a)
> squamous <- ctype==1
> smallcell <- ctype==2
> adeno <- ctype==3
> large <- ctype==4
> lognormalreg.fit1<-
survreg(Surv(time,status)~trt+dtime+age+prior+smallcell+adeno,data=a
,dist="lognormal")
> summary(lognormalreg.fit1)

Call:
survreg(formula = Surv(time, status) ~ trt + dtime + age + prior +
    smallcell + adeno, data = a, dist = "lognormal")
                Value Std. Error      z        p
(Intercept)    4.38656    0.45007  9.746 1.91e-22
trt           -0.13857    0.14481 -0.957 3.39e-01
dtime         -0.00854    0.00732 -1.166 2.43e-01
age            0.00947    0.00678  1.397 1.62e-01
prior         -0.39941    0.17337 -2.304 2.12e-02
smallcellTRUE -0.77237    0.16261 -4.750 2.04e-06
adenoTRUE     -0.83826    0.19370 -4.328 1.51e-05
Log(scale)    -0.20534    0.06296 -3.262 1.11e-03

Scale= 0.814

Log Normal distribution
Loglik(model)= -679   Loglik(intercept only)= -695.8
        Chisq= 33.44 on 6 degrees of freedom, p= 8.6e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

3) The next variable to be removed from the model is *trt (treatment)*.

```
> lognormalreg.fit2<-
survreg(Surv(time,status)~+dtime+age+prior+smallcell+adeno,data=a,di
st="lognormal")
> summary(lognormalreg.fit2)

Call:
survreg(formula = Surv(time, status) ~ +dtime + age + prior +
    smallcell + adeno, data = a, dist = "lognormal")
                Value Std. Error     z        p
(Intercept)    4.21693    0.41499 10.16 2.95e-24
dtime         -0.00887    0.00734 -1.21 2.26e-01
age            0.00877    0.00676  1.30 1.95e-01
prior         -0.39329    0.17384 -2.26 2.37e-02
smallcellTRUE -0.74962    0.16139 -4.64 3.40e-06
adenoTRUE     -0.85805    0.19328 -4.44 9.02e-06
Log(scale)    -0.20188    0.06296 -3.21 1.34e-03

Scale= 0.817

Log Normal distribution
Loglik(model)= -679.5   Loglik(intercept only)= -695.8
        Chisq= 32.52 on 5 degrees of freedom, p= 4.7e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

4) The 3rd variable to be removed from the model is *dtime (days from diagnosis to randomisation)*.

```
> lognormalreg.fit3<-
survreg(Surv(time,status)~+age+prior+smallcell+adeno,data=a,dist="lo
gnormal")
> summary(lognormalreg.fit3)

Call:
survreg(formula = Surv(time, status) ~ +age + prior + smallcell +
    adeno, data = a, dist = "lognormal")
                Value Std. Error     z        p
(Intercept)    4.16169     0.4145 10.04 1.01e-23
age            0.00878     0.0068  1.29 1.96e-01
prior         -0.47898     0.1595 -3.00 2.68e-03
smallcellTRUE -0.75753     0.1621 -4.67 2.95e-06
adenoTRUE     -0.83760     0.1935 -4.33 1.50e-05
Log(scale)    -0.19683     0.0630 -3.13 1.77e-03

Scale= 0.821

Log Normal distribution
Loglik(model)= -680.2   Loglik(intercept only)= -695.8
        Chisq= 31.07 on 4 degrees of freedom, p= 3e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

5) The 4th variable to be removed from the model is *age (in years)*.

```
> lognormalreg.fit4<-
survreg(Surv(time,status)~+prior+smallcell+adeno,data=a,dist="lognor
mal")
> summary(lognormalreg.fit4)

Call:
survreg(formula = Surv(time, status) ~ +prior + smallcell + adeno,
    data = a, dist = "lognormal")
              Value Std. Error     z        p
(Intercept)    4.674      0.123 37.90 0.00e+00
prior         -0.496      0.160 -3.10 1.96e-03
smallcellTRUE -0.739      0.163 -4.55 5.49e-06
adenoTRUE     -0.842      0.195 -4.32 1.55e-05
Log(scale)    -0.190      0.063 -3.01 2.60e-03

Scale= 0.827

Log Normal distribution
Loglik(model)= -681.1   Loglik(intercept only)= -695.8
        Chisq= 29.41 on 3 degrees of freedom, p= 1.8e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

6) The 5th variable to be removed from the model is *prior (prior therapy 0=no, 1=yes)*.

```
> lognormalreg.fit5<-
survreg(Surv(time,status)~+smallcell+adeno,data=a,dist="lognormal")
> summary(lognormalreg.fit5)

Call:
survreg(formula = Surv(time, status) ~ +smallcell + adeno, data = a,
    dist = "lognormal")
              Value Std. Error     z        p
(Intercept)    4.485      0.111 40.51 0.000000
smallcellTRUE -0.662      0.167 -3.97 0.000071
adenoTRUE     -0.745      0.199 -3.74 0.000186
Log(scale)    -0.153      0.063 -2.43 0.015033

Scale= 0.858

Log Normal distribution
Loglik(model)= -685.7   Loglik(intercept only)= -695.8
        Chisq= 20.17 on 2 degrees of freedom, p= 4.2e-05
Number of Newton-Raphson Iterations: 4
n= 137
```

But here we can see that: after the variable *prior* is excluded, the value of AIC increases.

So, we should keep *prior*.

At this stage, we should also check if the variables that have been removed from the model earlier, namely, *age, dtime*, *trt*, *factor(ctype)4* can enter the model. After several **backward** rounds check, such as:

```
> lognormalreg.fit6<-
survreg(Surv(time,status)~prior+smallcell+adeno+large,data=a,dist="lognormal
")
> lognormalreg.fit7<-
survreg(Surv(time,status)~trt+prior+smallcell+adeno,data=a,dist="lognormal")
> lognormalreg.fit8<-survreg(Surv(time,status)~
dtime+prior+smallcell+adeno,data=a,dist="lognormal")
> lognormalreg.fit9<-
survreg(Surv(time,status)~trt+prior+smallcell+adeno+large,data=a,dist="logno
rmal")
…….
…….
…….
```

Finally, it appears that:

- ***Prior***, ***smallcell*** *(cell type)*, ***adeno*** *(cell type)* are perhaps the only important variables in this data set.

And the selected model has the following performance.

```
> lognormalreg.fit4<-
survreg(Surv(time,status)~+prior+smallcell+adeno,data=a,dist="lognor
mal")
> summary(lognormalreg.fit4)

Call:
survreg(formula = Surv(time, status) ~ +prior + smallcell + adeno,
    data = a, dist = "lognormal")
              Value Std. Error     z        p
(Intercept)    4.674      0.123 37.90 0.00e+00
prior         -0.496      0.160 -3.10 1.96e-03
smallcellTRUE -0.739      0.163 -4.55 5.49e-06
adenoTRUE     -0.842      0.195 -4.32 1.55e-05
Log(scale)    -0.190      0.063 -3.01 2.60e-03

Scale= 0.827

Log Normal distribution
Loglik(model)= -681.1   Loglik(intercept only)= -695.8
        Chisq= 29.41 on 3 degrees of freedom, p= 1.8e-06
Number of Newton-Raphson Iterations: 4
n= 137
```

Note:

We can use R function *stepAIC* to obtain the same result more readily:

```
> library(mass)
> attach(a)
```

```
> squamous <- ctype==1
> smallcell <- ctype==2
> adeno <- ctype==3
> large <- ctype==4
> lognormalreg.fit1<-
survreg(Surv(time,status)~trt+dtime+age+prior+smallcell+adeno,data=a,
dist="lognormal")
> stepAIC(lognormalreg.fit1)
Start:  AIC=1374.09
Surv(time, status) ~ trt + dtime + age + prior + smallcell +
    adeno

            Df    AIC
- trt        1 1373.0
- dtime      1 1373.5
- age        1 1374.0
<none>         1374.1
- prior      1 1377.3
- adeno      1 1389.6
- smallcell  1 1392.9

Step:  AIC=1373.01
Surv(time, status) ~ dtime + age + prior + smallcell + adeno

            Df    AIC
- dtime      1 1372.5
- age        1 1372.7
<none>         1373.0
- prior      1 1376.0
- adeno      1 1389.4
- smallcell  1 1391.0

Step:  AIC=1372.46
Surv(time, status) ~ age + prior + smallcell + adeno

            Df    AIC
- age        1 1372.1
<none>         1372.5
- prior      1 1379.2
- adeno      1 1388.0
- smallcell  1 1390.7

Step:  AIC=1372.12
Surv(time, status) ~ prior + smallcell + adeno

            Df    AIC
<none>         1372.1
- prior      1 1379.4
- adeno      1 1387.6
- smallcell  1 1389.3
Call:
survreg(formula = Surv(time, status) ~ prior + smallcell + adeno,
    data = a, dist = "lognormal")

Coefficients:
  (Intercept)         prior smallcellTRUE     adenoTRUE
    4.6737875    -0.4960756    -0.7391344    -0.8421581

Scale= 0.8272893

Loglik(model)= -681.1   Loglik(intercept only)= -695.8
        Chisq= 29.41 on 3 degrees of freedom, p= 1.8e-06
n= 137
```

4.2 Influence analysis for the AFT model fit

From the information obtained in 4.1, we know that:

	Parameter	Value	Std.Error	z	p
Full model	b_0 Intercept	4.38092	0.47326	9.2569	2.10e-20
	b_1 trt	-0.13812	0.14528	-0.9507	3.42e-01
	b_2 dtime	-0.00851	0.00735	-1.1574	2.47e-01
	b_3 age	0.00949	0.00679	1.3968	1.62e-01
	b_4 prior	-0.39952	0.17339	-2.3042	2.12e-02
	b_5 factor(ctype)2	-0.76868	0.18869	-4.0738	4.63e-05
	b_6 factor(ctype)3	-0.83456	0.21617	-3.8608	1.13e-04
	b_7 factor(ctype)4	0.00824	0.21388	0.0385	9.69e-01
	s^* Log(scale)	-0.20538	0.06297	-3.2618	1.11e-03
Reduced model	b_0 Intercept	4.674	0.123	37.90	0.00e+00
	b_4 **prior**	-0.496	0.160	-3.10	1.96e-03
	b_5 **smallcellTRUE**	-0.739	0.163	-4.55	5.49e-06
	b_6 **adenoTRUE**	-0.842	0.195	-4.32	1.55e-05
	s^* Log(scale)	-0.190	0.063	-3.01	2.60e-03

Table 2 Parameters of the full model and the reduced model

We also know that:

	Loglik(model)
Full model	-679
Reduced model	-681.1

Table 3 Log-likelihood for the full model and the reduced model

Assuming our diagnostics give some credibility to our chosen model, we can do some **inference analysis**. Let $\underline{q} = [\underline{q_1}, \underline{q_2}] = [\underline{b}, s]$.

1) We can test: $H_0: b_1 = b_2 = ... = b_7 = 0$

   When $b_1 = b_2 = ... = b_7 = 0$, we get:

```
> lognormalreg0<-survreg(Surv(time,status)~1,data=a,dist="lognormal")
> summary(lognormalreg0)

Call:
survreg(formula = Surv(time, status) ~ 1, data = a, dist =
"lognormal")
             Value Std. Error     z     p
(Intercept)  4.1069      0.080 51.30 0.000
Log(scale)  -0.0752      0.063 -1.19 0.233

Scale= 0.928
```

```
Log Normal distribution
Loglik(model)= -695.8   Loglik(intercept only)= -695.8
Number of Newton-Raphson Iterations: 5
n= 137
```

$$\Lambda_0 = 2\,(l(\hat{q}) - l(\tilde{q}))$$

$$= 2\,(l(\hat{b}_0, \hat{b}_1, ..., \hat{b}_7, \hat{s}) - l(\tilde{b}_0, 0,0,0,0,0,0,0, \tilde{s}))$$

$$= 2\,(\text{loglik}_1 - \text{loglik}_0)$$

$$= 2\,(-679 + 695.8)$$

$$= 33.6$$

Since:

```
> 1-pchisq(33.6,7)
[1] 2.046140e-05                    ## p-value, d.f.= 9-2 = 7
```

The P-value is much smaller than 0.05. This means that: at the 95% confidence level, we may reject H_0, and find strong evidence that $b_1, b_2, ...b_7$ may not be equal to 0 at the same time.

2) Let $\hat{q}$ refer to the parameters of the full model, and $\tilde{q}$ be for the reduced model.

We can test: H_0: $b_1 = b_2 = b_3 = b_7 = 0$

Use this method, thus,

$$\Lambda_1 = 2\,(l(\hat{q}) - l(\tilde{q}))$$

$$= 2\,(l(\hat{b}_0, \hat{b}_1, ..., \hat{b}_7, \hat{s}) - l(\tilde{b}_0, 0,0,0, \tilde{b}_4, \tilde{b}_5, \tilde{b}_6, 0, \tilde{s}))$$

$$= 2\,(\text{loglik}_1 - \text{loglik}_2)$$

$$= 2\,(-679 + 681.1)$$

$$= 4.2$$

Since:

```
> 1-pchisq(4.2,4)
[1] 0.3796149                       ## p-value, d.f.= 9-5 = 4
```

The P-value is bigger than 0.05. This means that: at the 95% confidence level, there is **hardly** any log-likelihood difference between the full model and the reduced model, and we find no evidence against the model only ***Prior, cell type = smallcell, cell type = adeno*** included.

Also, we can use this approach with **a forward and backward stepwise procedure** to conduct variable selection, and we can find the same results as the way using the AIC.

5 Get inferences for the survivor functions at a point from the reduced model

5.1 Obtain the covariance matrix Cov($\hat{\underline{b}}, \hat{s}$) for the reduced model

There are no built-in functions in R that will provide inferences for the AFT model. To get inferences for the survivor function at a point, we need the covariance matrix Cov($\hat{\underline{b}}, \hat{s}$). Unfortunately R gives the covariance matrix for Cov($\hat{\underline{b}}, \hat{s}^* = \log \hat{s}$).

To get the covariance matrix Cov($\hat{\underline{b}}, \hat{s}$),[10] we simply multiply all the elements in the last row and the last column by $\hat{s}$.

```
> lognormalreg.fit4$var
               (Intercept)          prior smallcellTRUE   adenoTRUE    Log(scale)
(Intercept)    0.0152053916 -9.890744e-03 -1.293597e-02 -0.0133528751  2.014160e-04
prior         -0.0098907439  2.565813e-02  4.022582e-03  0.0050969879 -1.076527e-05
smallcellTRUE -0.0129359702  4.022582e-03  2.644518e-02  0.0121832655 -5.124566e-05
adenoTRUE     -0.0133528751  5.096988e-03  1.218327e-02  0.0379706351 -1.067674e-04
Log(scale)     0.0002014160 -1.076527e-05 -5.124566e-05 -0.0001067674  3.963683e-03
```

```
   > covar<-lognormalreg.fit4$var
   > coninfr<-covar
   > coninfr[,5]<-covar[,5]*lognormalreg.fit4$scale
   > coninfr[5,]<-covar[5,]*lognormalreg.fit4$scale
```

The covariance matrix Cov($\hat{\underline{b}}, \hat{s}$) is:

```
   > coninfr
               (Intercept)          prior smallcellTRUE     adenoTRUE    Log(scale)
(Intercept)    0.0152053916 -9.890744e-03 -1.293597e-02 -0.0133528751  1.666293e-04
prior         -0.0098907439  2.565813e-02  4.022582e-03  0.0050969879 -8.905996e-06
smallcellTRUE -0.0129359702  4.022582e-03  2.644518e-02  0.0121832655 -4.239498e-05
adenoTRUE     -0.0133528751  5.096988e-03  1.218327e-02  0.0379706351 -8.832750e-05
Log(scale)*    0.0001666293 -8.905996e-06 -4.239498e-05 -0.0000883275  3.279112e-03
```

5.2 Get inferences of median time-to-events

The median time-to-events (TTE's) [11] for an individual with covariate vector ***X*** under the log normal regression AFT model is $T_i = \exp(X_i^T \hat{\underline{b}})$.

[10] Introduction to Survival Analysis Using R, http://www.stat.ncu.edu.tw/teacher/Tsengyk/Handout2b.doc.

5.2.1 Check the effects of prior therapy with squamous cell type

Consider two patients with the following covariate combinations:

- X_1=(1, prior = 0, cell type = *squamous*)
- X_2=(1, prior = 1, cell type = *squamous*)

This means that the two patients with *squamous* cell type only have different conditions on prior therapy: X_2 uses prior therapy and X_1 does not.

In R, this can be expressed as:

```
> x1<-c(1,0,0,0)
> x2<-c(1,1,0,0)
> medx1 <- exp(x1%*%lognormalreg.fit4$coeff)
> matrix(c(x1,0),1,5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
> c(exp(x1%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x1,0),1,5)%*%coninfr%*%matrix(c(x1,0),5,1))),
medx1,
exp(x1%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x1,0),1,5)%*%con
infr%*%matrix(c(x1,0),5,1))))
[1] 84.10782 107.10262 136.38413

> medx2<-exp(x2%*%lognormalreg.fit4$coeff)
> c(exp(x2%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x2,0),1,5)%*%coninfr%*%matrix(c(x2,0),5,1))),
medx2,
exp(x2%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x2,0),1,5)%*%con
infr%*%matrix(c(x2,0),5,1))))
[1] 49.06405 65.21646 86.68640
```

From the above, we obtain 95% confidence intervals for subject X_1:

Lower CI	Median TTE	Upper CI
84.10782	107.10262	136.38413

Table 4 95% confidence intervals of median TTE for subject X_1

We obtain 95% confidence intervals for subject X_2:

Lower CI	Median TTE	Upper CI
49.06405	65.21646	86.68640

Table 5 95% confidence intervals of median TTE for subject X_2

5.2.2 Check the effects of prior therapy with small cell type

Consider two patients with the following covariate combinations:

- X_3=(1, prior = 0, cell type = *small cell*)
- X_4=(1, prior = 1, cell type = *small cell*)

[11] Introduction to Survival Analysis Using R, http://www.stat.ncu.edu.tw/teacher/Tsengyk/Handout2b.doc.

```
> x3<-c(1,0,1,0)
> x4<-c(1,1,1,0)
> medx3 <- exp(x3%*%lognormalreg.fit4$coeff)
> matrix(c(x3,0),1,5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    1    0    0
> c(exp(x3%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x3,0),1,5)%*%coninfr%*%matrix(c(x3,0),5,1))),
medx3,
exp(x3%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x3,0),1,5)%*%con
infr%*%matrix(c(x3,0),5,1))))
[1] 39.98289 51.14441 65.42174

> medx4<-exp(x4%*%lognormalreg.fit4$coeff)
> c(exp(x4%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x4,0),1,5)%*%coninfr%*%matrix(c(x4,0),5,1))),
medx4,
exp(x4%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x4,0),1,5)%*%con
infr%*%matrix(c(x4,0),5,1))))
[1] 22.21560 31.14263 43.65685
```

From the above, we obtain 95% confidence intervals for subject X_3:

Lower CI	Median TTE	Upper CI
39.98289	51.14441	65.42174

Table 6 95% confidence intervals of median TTE for subject X_3

We obtain 95% confidence intervals for subject X_4:

Lower CI	Median TTE	Upper CI
22.21560	31.14263	43.65685

Table 7 95% confidence intervals of median TTE for subject X_4

5.2.3 Check the effects of prior therapy with adeno cell type

Consider two patients with the following covariate combinations:

- X_5=(1, prior = 0, cell type = *adeno*)
- X_6=(1, prior = 1, cell type = *adeno*)

```
> x5<-c(1,0,0,1)
> x6<-c(1,1,0,1)
> medx5 <- exp(x5%*%lognormalreg.fit4$coeff)
> matrix(c(x5,0),1,5)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    1    0
> c(exp(x5%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x5,0),1,5)%*%coninfr%*%matrix(c(x5,0),5,1))),
medx5,
exp(x5%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x5,0),1,5)%*%con
infr%*%matrix(c(x5,0),5,1))))
[1] 33.54016 46.13765 63.46670
```

```
> medx6<-exp(x6%*%lognormalreg.fit4$coeff)
> c(exp(x6%*%lognormalreg.fit4$coeff-
1.96*sqrt(matrix(c(x6,0),1,5)%*%coninfr%*%matrix(c(x6,0),5,1))),
medx6,
exp(x6%*%lognormalreg.fit4$coeff+1.96*sqrt(matrix(c(x6,0),1,5)%*%con
infr%*%matrix(c(x6,0),5,1))))
[1] 18.75190 28.09394 42.09010
```

From the above, we obtain 95% confidence intervals for subject X_5:

Lower CI	Median TTE	Upper CI
33.54016	46.13765	63.46670

Table 8 95% confidence intervals of median TTE for subject X_5

We obtain 95% confidence intervals for subject X_6:

Lower CI	Median TTE	Upper CI
18.75190	28.09394	42.09010

Table 9 95% confidence intervals of median TTE for subject X_6

According these outputs we obtain, the summary table can be set up:

	Squamous	*Small Cell*	*Adeno*
Prior = 0	107.10262	51.14441	46.13765
Prior = 1	65.21646	31.14263	28.09394

Table 10 Median TTE for subject X_1 to X_6

6 Plot survival curves from the reduced model

As we did in 3.4, in this case, according to log-location-scale families,

$S(t_i) = S_0(\frac{\log t_i - X_i^T \underline{b}}{s_i})$ where for log normal regression $S_0(w) = 1 - \Phi(w)$ and

$w = \frac{\log t_i - X_i^T \underline{b}}{s_i}$.

Consider two patients with the following covariate combinations:

- X_1=(1, prior = 0, cell type = *squamous*)
- X_2=(1, prior = 1, cell type = *squamous*)
- X_3=(1, prior = 0, cell type = *small cell*)
- X_4=(1, prior = 1, cell type = *small cell*)

- X_5=(1, prior = 0, cell type = *adeno*)
- X_6=(1, prior = 1, cell type = *adeno*)

In R, this can be expressed as:

```
> x1<-c(1,0,0,0)
> x2<-c(1,1,0,0)
> x3<-c(1,0,1,0)
> x4<-c(1,1,1,0)
> x5<-c(1,0,0,1)
> x6<-c(1,1,0,1)
> t <- seq(0,360,0.1)
> sx1 <- 1-pnorm((log(t)-
x1%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> sx2 <- 1-pnorm((log(t)-
x2%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> sx3 <- 1-pnorm((log(t)-
x3%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> sx4 <- 1-pnorm((log(t)-
x4%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> sx5 <- 1-pnorm((log(t)-
x5%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> sx6 <- 1-pnorm((log(t)-
x6%*%lognormalreg.fit4$coeff)/lognormalreg.fit4$scale)
> plot(t,sx1, main="Survival curves using the reduced log normal AFT
model",ylab='Survival Probibility',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,type='l')
> lines(t,sx2,lty=2)
> lines(t,sx3,lty=3)
> lines(t,sx4,lty=4)
> lines(t,sx5,lty=5)
> lines(t,sx6,lty=6)
> legend(200, .93,lty=1:6, cex=.7, c("Subject X1: no prior,
squamous","Subject X2: prior, squamous","Subject X3: no prior, small
cell","Subject X4: prior, small cell","Subject X5: no prior, adeno","Subject
X6: prior, adeno"))
```

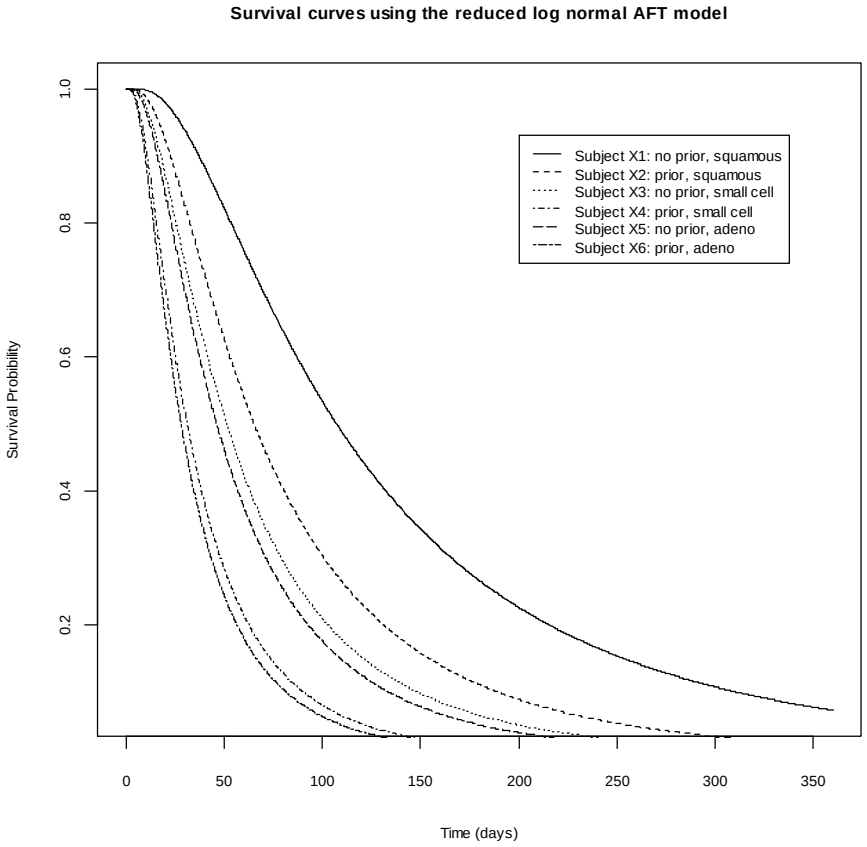

Figure 15 Survival curves using the reduced log normal AFT model

7 Perform diagnostic analyses to evaluate the adequacy of model fit

Model fit was evaluated based upon a graphical comparison between empirical Kaplan–Meier survival curves and fitted or "predicted" survival curves generated from the final AFT model.

7.1 Evaluate model fit by *cell type*

```
> plot(fit.km.byctype,main="Clinical trial for two treatment regimens for
lung cancer by cell type",ylab='K-M estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,col=c(1:10),lty=1:10)
> lines(t,sx1, col=c(5), lty=5)
> lines(t,sx2, col=c(6), lty=6)
> lines(t,sx3, col=c(7), lty=7)
> lines(t,sx4, col=c(8), lty=8)
> lines(t,sx5, col=c(9), lty=9)
> lines(t,sx6, col=c(10), lty=10)
> legend(200, .93,lty=1:10, col=c(1:10), cex=.7,
c("squamous","smallcell","adeno","large","Subject X1: no prior,
squamous","Subject X2: prior, squamous","Subject X3: no prior, small
```

```
cell","Subject X4: prior, small cell","Subject X5: no prior, adeno","Subject
X6: prior, adeno"))
```

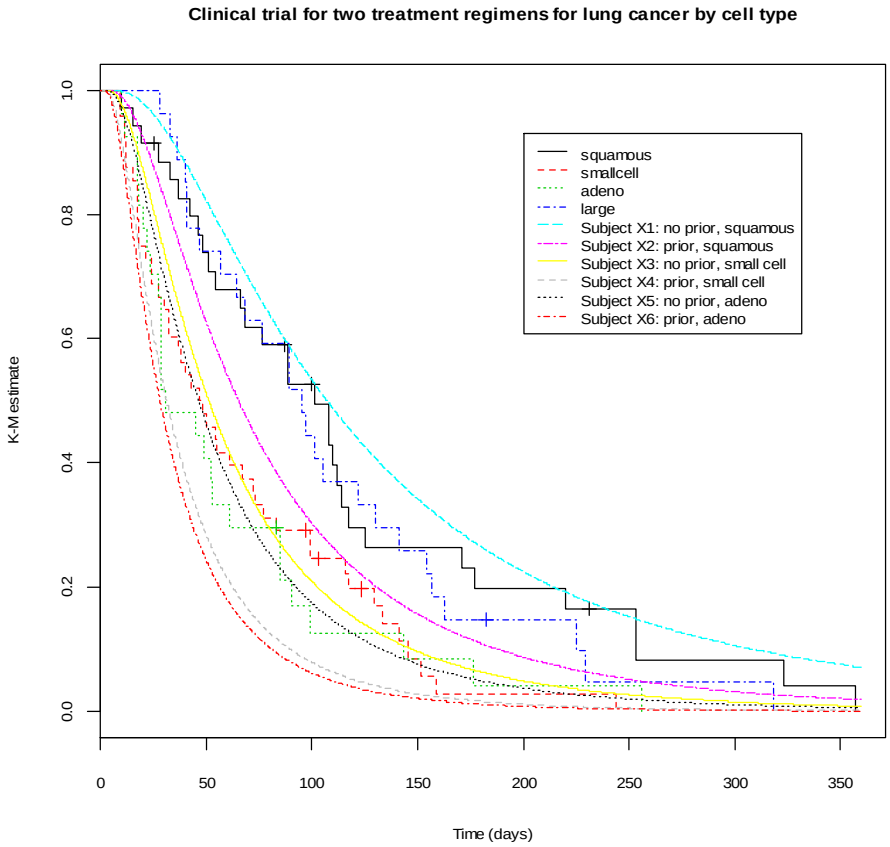

Figure 16 Graphical comparison between K-M curves and final fitted curves by cell type

From this figure, we can see that: the K-M curve of *squamous* roughly falls into the interval between X_1 and X_2; X_3 and X_5 fit the K-M curves of *smallcell* and *adeno*. Details will be presented in the Summary Report.

7.2 Evaluate model fit by *prior*

```
> fit.km.byprior=survfit(Surv(time,status)~prior,data=a,conf.type="none")
> plot(fit.km.byprior,main="Clinical trial for two treatment regimens for
lung cancer by prior",ylab='K-M estimate',xlab='Time
(days)',cex.main=0.8,cex.lab=0.7,cex.axis=0.7,col=c(1:2),lty=1:2)
> lines(t,sx1, col=c(5), lty=5)
> lines(t,sx2, col=c(6), lty=6)
> lines(t,sx3, col=c(7), lty=7)
> lines(t,sx4, col=c(8), lty=8)
> lines(t,sx5, col=c(9), lty=9)
> lines(t,sx6, col=c(10), lty=10)
> legend(200, .93,lty=c(1,2,5,6,7,8,9,10), col=c(1,2,5,6,7,8,9,10), cex=.7,
c("no prior","prior","Subject X1: no prior, squamous","Subject X2: prior,
```

```
squamous","Subject X3: no prior, small cell","Subject X4: prior, small
cell","Subject X5: no prior, adeno","Subject X6: prior, adeno"))
```

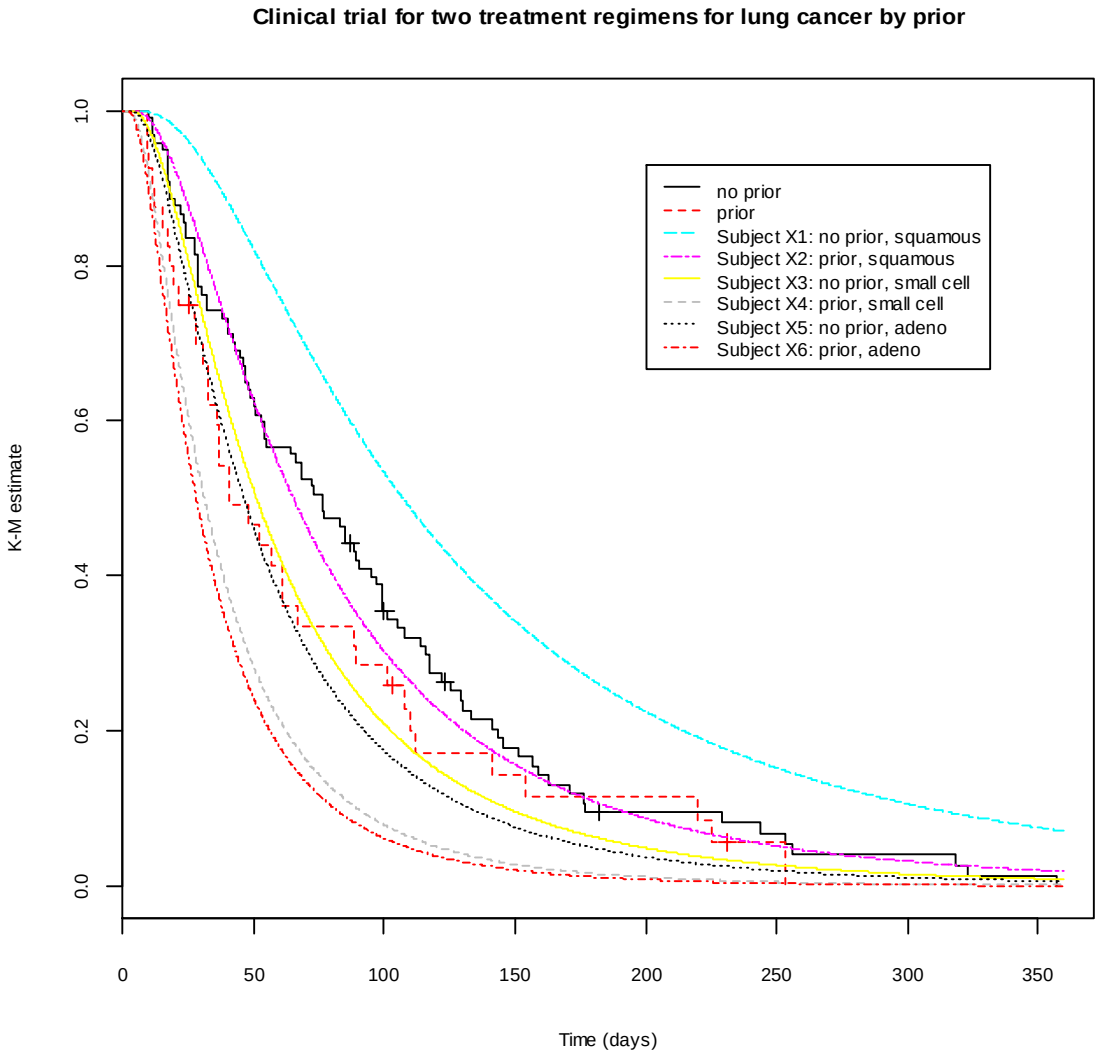

Figure 17 Graphical comparison between K-M curves and final fitted curves by *prior*

From this figure, we can see that: only X_2 seems to fit the K-M curve with *prior*. Details will be presented in the Summary Report.