

# Introduction to survival analysis (2)

Yue Jiang

Duke University

# A disclaimer

Today's (and last time's) lectures are introductory surface level treatments of survival analysis. We focus on applications and use cases -- there are no theoretical results presented (even for important subjects like variance estimation).

There is much to discuss regarding survival analysis both theoretically and in application. In STA 440, we will focus on using and implementing commonly used methods to tackle real-world datasets instead of focusing on theoretical considerations.

# The DIG Trial

## The New England Journal of Medicine

© Copyright, 1997, by the Massachusetts Medical Society

VOLUME 336

FEBRUARY 20, 1997

NUMBER 8



THE EFFECT OF DIGOXIN ON MORTALITY AND MORBIDITY IN PATIENTS WITH  
HEART FAILURE

THE DIGITALIS INVESTIGATION GROUP\*

Investigators compared the **primary outcome** of the number of days from the start of the study to either death or hospitalization from worsening heart failure.

# Representing survival data

Underlying data:

- $T$ : Failure time, a non-negative random variable
- $C$ : Censoring time, a non-negative random variable Observed data for individual  $i$ :
- $Y_i$ :  $(T_i \wedge C_i)$ , the minimum of  $T_i$  and  $C_i$
- $\delta_i$ :  $1_{(T_i \leq C_i)}$ , whether we observe a failure

If  $\delta_i = 0$ , then we have **right-censoring**: the survival time is longer than the censoring time.

Commonly, we assume  $C_i$  are *i.i.d.* random variables with some distribution and that the censoring mechanism is *independent* of the failure mechanism.

# Characterizing continuous $T$

- Density function:  $f(t) = \lim_{\Delta t \rightarrow 0^+} \frac{P(t \leq T < t + \Delta t)}{\Delta t}$
- Distribution function:  $F(t) = P(T \leq t) = \int_0^t f(s) ds$
- Survival function:  $S(t) = P(T > t) = 1 - F(t)$
- Hazard function:  $\lambda(t) = \lim_{\Delta t \rightarrow 0^+} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}$
- Cumulative hazard function:  $\Lambda(t) = \int_0^t \lambda(s) ds$

Knowing one is equivalent to knowing the others.

# Hazard distributions

Exponential distribution:

- $f(t) = \lambda e^{-\lambda t}$  (don't get the rate parameter  $\lambda$  confused with the hazard)
- $F(t) = 1 - e^{-\lambda t}$
- $S(t) = e^{-\lambda t}$
- $\lambda(t) = \lambda$
- $\Lambda(t) = \lambda t$

What do you notice about the hazard for survival times that have an exponential distribution?

# Hazard distributions

Weibull distribution:

- $f(t) = p\lambda^p t^{p-1} e^{-(\lambda t)^p}$
- $F(t) = 1 - e^{-(\lambda t)^p}$
- $S(t) = e^{-(\lambda t)^p}$
- $\lambda(t) = p\lambda^p t^{p-1}$
- $\Lambda(t) = (\lambda t)^p$

When the shape parameter  $p$  is 1, then we have the exponential distribution.

The hazard increases monotonically over time if  $p > 1$  and decreases monotonically if  $p < 1$  (is this reasonable to assume?).

# Hazard distributions

Plotting Weibull hazard with  $\lambda = 1$  and various shape parameters  $p$ :



# Hazard distributions

Log-normal distribution:

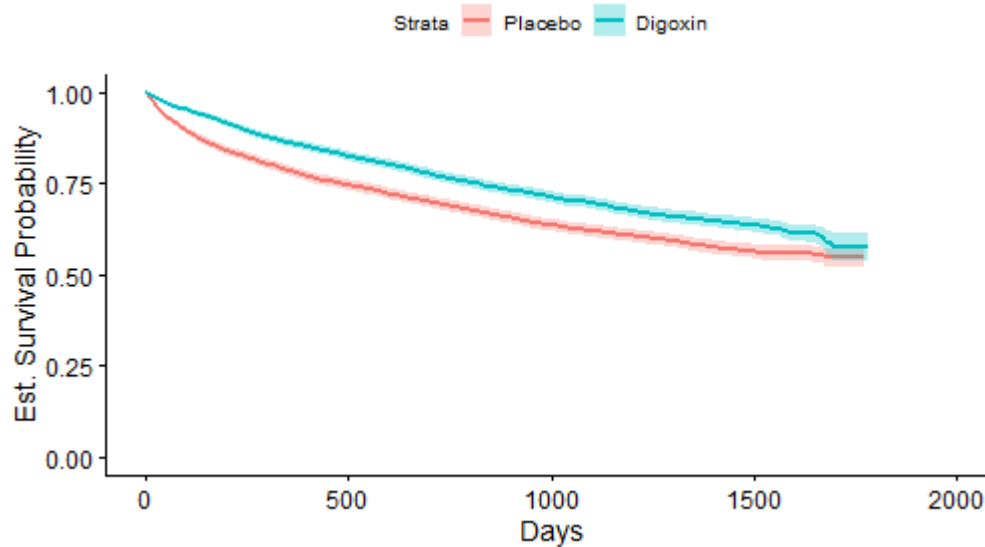
- $f(t) = \frac{1}{t\sigma} \phi \left( \frac{\log(t) - \mu}{\sigma} \right)$
- $F(t) = \Phi \left( \frac{\log(t) - \mu}{\sigma} \right)$
- $S(t) = 1 - F(t)$
- $\lambda(t) = f(t)/S(t)$
- $\Lambda(t) = \int_0^t \lambda(s) ds$

# Hazard distributions

Plotting log-normal hazard with  $\mu = 0$  and  $\sigma^2 = 1$ :

# Review: comparing multiple groups

```
library(survminer)
ggsurvplot(survfit(Surv(DWHFDAYS, DWHF) ~ TRTMT, data = dig),
  xlab = "Days", ylab = "Est. Survival Probability",
  ylim = c(0, 1),
  conf.int = T, censor = F,
  legend.labs = c("Placebo", "Digoxin"))
```



# Review: comparing multiple groups

```
survdiff(Surv(DWHFDAYS, DWHF) ~ TRTMT, data = dig)
```

```
## Call:
## survdiff(formula = Surv(DWHFDAYS, DWHF) ~ TRTMT, data = dig)
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## TRTMT=0 3403      1291      1126      24.1      46.6
## TRTMT=1 3397      1041      1206      22.5      46.6
##
##  Chisq= 46.6  on 1 degrees of freedom, p= 9e-12
```

How might we adjust for potential confounders? Is there any way to create a predictive model for survival time?

# An accelerated failure time model

An **accelerated failure time** (AFT) model assumes

$$\log(T_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon_i$$

where  $\epsilon_i$  are commonly assumed to be i.i.d. and follow some specified distribution.

There is a one-to-one relationship between the distribution of  $T$  and the assumed error distribution in the AFT model.

For instance, if  $\epsilon$  has a normal distribution, then  $T$  has a log-normal distribution. If  $\epsilon$  has a gen. EV distribution, then  $T$  has a Weibull distribution, etc.

In software packages, these models are often fit by specifying the distribution of  $T$ .

# An accelerated failure time model

Note that we can also write the AFT model as

$$\begin{aligned} T_i &= \exp(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi}) e^{\epsilon_i} \\ &= e^{\beta_0} e^{\beta_1 x_{1i}} e^{\beta_2 x_{2i}} \cdots e^{\beta_p x_{pi}} e^{\epsilon_i} \end{aligned}$$

Covariates in an AFT model have a multiplicative effect on *time*. For instance, if  $\beta_k = 0.4$ , then  $\exp(\beta_k) \approx 1.5$ . Holding all else equal, an individual with covariate  $x_k$  one unit greater than another is expected to survive approximately 1.5 times longer than the other.

# Fitting an AFT model

```
library(survival)
aft_e <- survreg(Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
                 data = dig, dist = "exponential")
summary(aft_e)

##
## Call:
## survreg(formula = Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
##         data = dig, dist = "exponential")
##              Value Std. Error      z      p
## (Intercept)  6.62850    0.07495 88.43 < 2e-16
## TRTMT        0.29856    0.04167  7.17 7.7e-13
## EJF_PER      0.04037    0.00239 16.92 < 2e-16
## PREVMI      -0.04978    0.04364 -1.14  0.25
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= -20511.5   Loglik(intercept only)= -20680.3
##      Chisq= 337.51 on 3 degrees of freedom, p= 7.6e-73
## Number of Newton-Raphson Iterations: 5
## n=6799 (1 observation deleted due to missingness)
```

# Fitting an AFT model

```
library(survival)
aft_w <- survreg(Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
                 data = dig, dist = "weibull")
summary(aft_w)
```

```
##
## Call:
## survreg(formula = Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
##         data = dig, dist = "weibull")
##              Value Std. Error      z      p
## (Intercept)  6.58614    0.10266 64.15 < 2e-16
## TRTMT        0.39910    0.05768  6.92 4.5e-12
## EJF_PER      0.05263    0.00337 15.61 < 2e-16
## PREVMI       -0.06661    0.06001  -1.11  0.27
## Log(scale)   0.31844    0.01891 16.84 < 2e-16
##
## Scale= 1.37
##
## Weibull distribution
## Loglik(model)= -20351.2   Loglik(intercept only)= -20504.9
##      Chisq= 307.43 on 3 degrees of freedom, p= 2.5e-66
## Number of Newton-Raphson Iterations: 5
## n=6799 (1 observation deleted due to missingness)
```



# A proportional hazards model

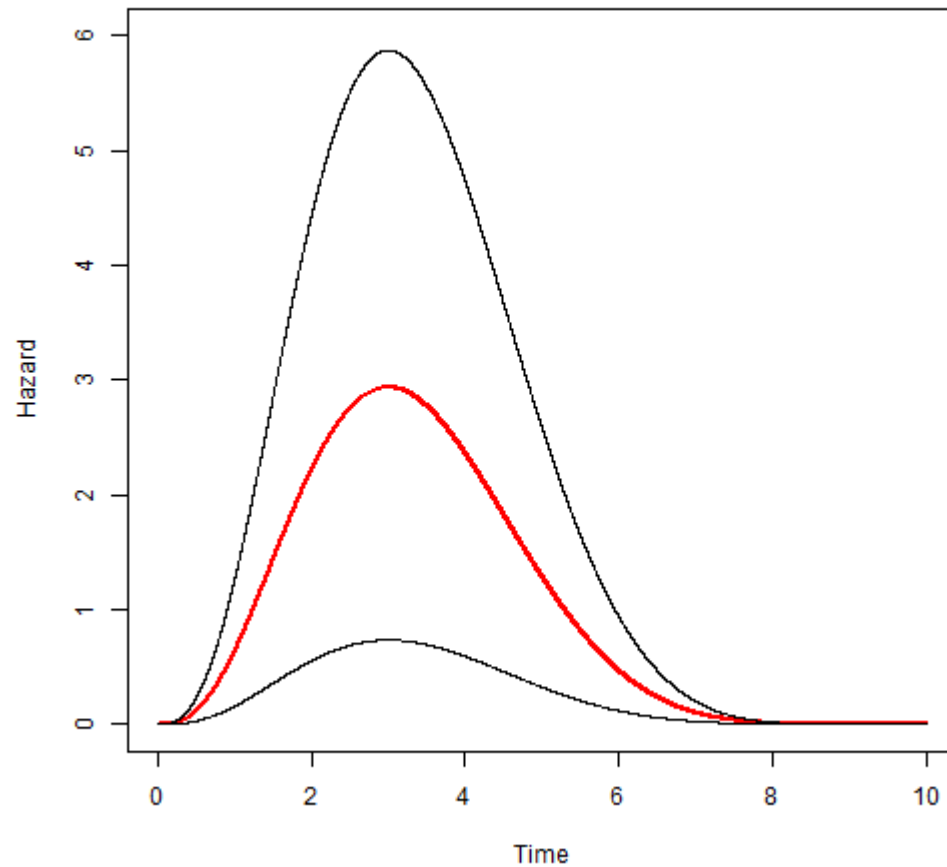
$$\lambda(t) = \lambda_0(t) \exp(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)$$

where the **baseline hazard** is assumed to have some distribution (or maybe not!...more on that in just a bit).

Covariates in a PH model have a multiplicative effect on *hazards*. For instance, if  $\beta_k = 0.4$ , then  $\exp(\beta_k) \approx 1.5$ . Holding all else equal, an individual with covariate  $x_k$  one unit greater than another is expected to have approximately 1.5 times the *hazard* of the event than the other.

Would you rather have a higher linear predictor or a lower linear predictor in a PH model? How does this compare to the AFT model?

# A proportional hazards model



# Fitting a PH model

```
library(eha)
ph_ln <- phreg(Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
               data = dig, dist = "lognormal")
ph_ln
```

```
## Call:
## phreg(formula = Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,
##       data = dig, dist = "lognormal")
##
## Covariate           W.mean      Coef Exp(Coef)   se(Coef)     Wald p
## (Intercept)                4.480
## TRTMT              0.518    -0.290    0.748    0.042    0.000
## EJF_PER            29.455    -0.038    0.963    0.002    0.000
## PREVMI              0.645     0.048    1.049    0.044    0.271
##
## log(scale)                15.421
## log(shape)               -1.358
##
## Events                2332
## Total time at risk      6092212
## Max. log. likelihood    -20339
## LR test statistic       303.79
## Degrees of freedom      3
```

# The Cox proportional hazards model

$$\lambda(t) = \lambda_0(t) \exp(\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)$$

In a parametric survival model (such as ones we've seen), the survival times are assumed to follow a specific distribution, which is a fairly strong assumption.

Often times, we only care about the  $\beta$  terms and not so much the  $\lambda_0$ . Using the concept of partial likelihood, Cox (1972) found that we can "separate" inference for the  $\beta$  terms from  $\lambda_0$ .

The Cox model is a *semiparametric* survival model;  $\lambda_0(t)$  is left completely unspecified, with no assumptions made on its shape. (A semi-parametric version of the AFT model also exists, but isn't very popular).

# The Cox proportional hazards model

- By far the most commonly used regression model for survival data
- Attractive interpretations using hazard ratios
- Can be extended to accommodate time-varying covariates, recurrent events, etc.
- Fewer assumptions than fully parametric models, but still requires PH assumption
- Can compare nested models by using partial likelihood ratio statistic, which has asymptotic  $\chi^2$  distribution (df = difference in number of parameters)

# Fitting the Cox PH model

```
coxml <- coxph(Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,  
              data = dig)  
summary(coxml)
```

```
## Call:  
## coxph(formula = Surv(DWHFDAYS, DWHF) ~ TRTMT + EJF_PER + PREVMI,  
##       data = dig)  
##  
##      n= 6799, number of events= 2332  
##      (1 observation deleted due to missingness)  
##  
##              coef exp(coef)  se(coef)      z Pr(>|z|)  
## TRTMT      -0.289745  0.748454  0.041666  -6.954 3.55e-12 ***  
## EJF_PER    -0.038061  0.962654  0.002381 -15.985 < 2e-16 ***  
## PREVMI      0.047921  1.049087  0.043638   1.098  0.272  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
##              exp(coef) exp(-coef) lower .95 upper .95  
## TRTMT           0.7485      1.3361    0.6898    0.8121  
## EJF_PER          0.9627      1.0388    0.9582    0.9672  
## PREVMI           1.0491      0.9532    0.9631    1.1428  
##
```

# Fitting the Cox PH model

The strength of the Cox model is that we can ignore estimation of  $\lambda_0$  completely (it doesn't matter for valid inference on the  $\beta$ s).

If we want to estimate survival probabilities, then we must estimate the baseline hazard. A non-parametric estimator (the **Breslow estimator**) is implemented by the `basehaz` function in the `survival` package (confusingly, this is of the *cumulative* hazard). It is given by:

$$\hat{\Lambda}_0(t) = \sum_{i:t_{(i)} < t} \frac{1}{\sum_{j \in R_i} \exp(\mathbf{X}_j \beta)}$$

We can then estimate survival in the Cox model by:

$$\hat{S}(t|\mathbf{X}) = \exp(-\exp(\mathbf{X}\beta)\hat{\Lambda}_0(t))$$

# Cox model diagnostics

Recall the inverse CDF result: if  $T_i$  has survival function  $S_i(t)$ , then  $S_i(T_i)$  should have a uniform distribution on  $(0, 1)$  and  $\Lambda_i(T_i)$  should follow  $\text{Exp}(1)$  distribution.

Thus, if the model is correct, then the estimated cumulative hazard  $\hat{\Lambda}$  at observed times should be a censored sample from  $\text{Exp}(1)$ . These  $\hat{\Lambda}_i(Y_i)$  are known as **Cox-Snell residuals**.

Plotting  $\log(-\log \hat{S}(Y_i))$  against  $\log(Y_i)$  should thus follow a straight line through the origin at a 45-degree angle.

Although Cox-Snell residuals can also be used for other models (e.g., checking whether distribution specified in AFT model is appropriate), they're not too useful in practice (for a variety of reasons).



# Cox model diagnostics

To assess PH assumption, we can examine **Schoenfeld residuals**. Intuitively, each Schoenfeld residual is the difference between the observed covariate and the expected covariate for each observed failure, conditioned on the risk set at that time.

In plotting Schoenfeld residuals vs. survival times, we expect to see them randomly distributed around 0.

# Cox model diagnostics

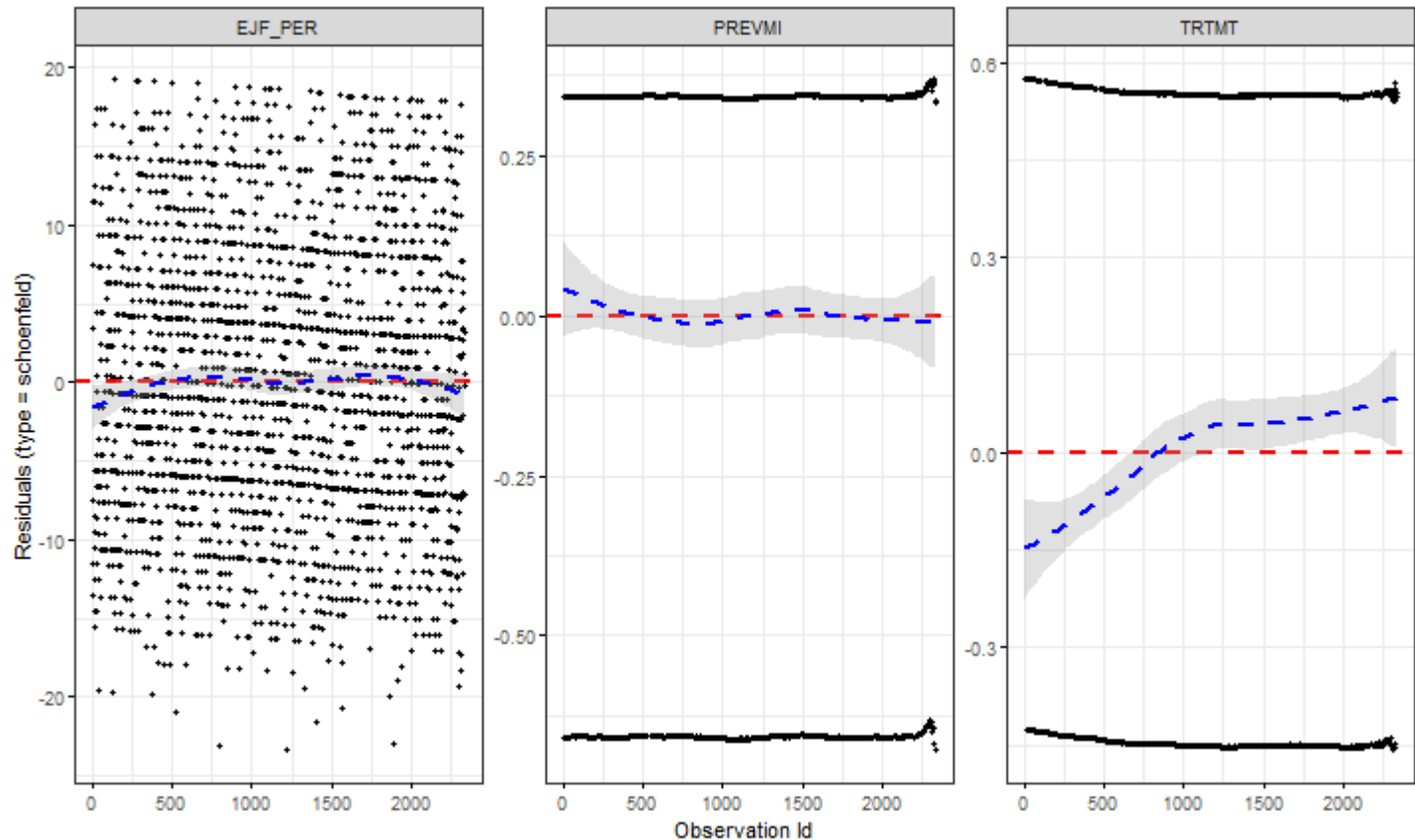
**Martingale residuals** are based on the difference between observed number of events up until time  $Y_i$  and the expected count based on the fitted model.

**Deviance residuals** are a normalized transformation of the martingale residuals that correct their skewness. They should be randomly distributed around mean 0 with a variance of 1.

In practice, these residuals are useful for finding potential outliers: negative values "lived longer than expected" and positive values "died too soon."

# Cox model diagnostics

```
library(survminer)
ggcoxdiagnostics(coxm1, type = "schoenfeld")
```



# Cox model diagnostics

```
library(survminer)
ggcoxdiagnostics(coxm1, type = "deviance", linear.predictions = F)
```

