Survival analysis and Multiple imputation

CMED6040 – Session 4

Tim Tsang (matklab@hku.hk)

School of Public Health
The University of Hong Kong

6 June 2023

Session 4 learning objectives

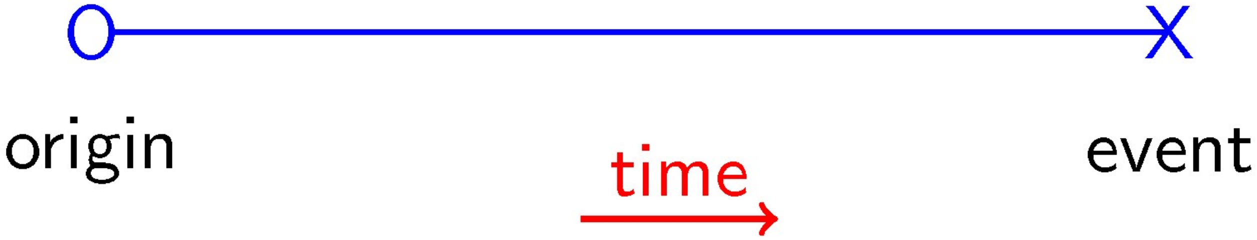
After this session, students should be able to

- Apply and plot the Kaplan-Meier estimator
- Apply proportional hazards regression models
- Apply parametric accelerated failure time models
- Analyse data in the presence of interval censoring

Survival analysis

Recap on survival data

- Survival data, or time-to-event data, record the time from a well-defined starting point (time origin) until the occurrence of a particular event (end point).
- When the end point is death, the data are literally survival data.



Recap on survival data

- Definition of origin and event needs to be very clear
- For example, in a study we want to estimate the incubation period (time from infection to onset)
- The origin is date of enrollment or date of infection?
- The event is date of onset (presence of at least one symptoms or two symptoms)?

Survival and hazard functions

- $F(t) = \Pr(T \leq t)$ is the cumulative density function of event occurs. The survival function is the probability of surviving event-free beyond time t:
$$S(t) = \Pr(T > t) = 1 - F(t)$$
- Probability takes values between 0 and 1, so S(t) takes values between 0 and 1
- For a survival function, $\Pr(T = 1) = \Pr(T = 2) = 0.5$
- Hence, $F(0.99) = P(T \leq 0.99) = 0$
- $F(1.01) = P(T \leq 1.01) = P(T = 1) = 0.5$
- $F(1.99) = P(T \leq 1.99) = P(T = 1) = 0.5$
- $F(2.01) = P(T \leq 2.01) = P(T = 1) + P(T = 2) = 1$

Survival and hazard functions

- The hazard function is the rate at which the event occurs at time *t* conditional on it not having yet occurred

$$h(t) = \frac{\Pr(t \leq T \leq t + \delta t | T \geq t)}{\delta t}$$

$$= \frac{\Pr(t \leq T \leq t + \delta t, T \geq t)}{\Pr(T \geq t)\delta t} = \frac{\Pr(t \leq T \leq t + \delta t)}{\Pr(T \geq t)\delta t}$$

$$= \frac{f(t)}{S(t)} = -\frac{d}{dt}\log(S(t))$$

Remark: $\frac{d}{dt}\log\big(S(t)\big) = \frac{S'(t)}{S(t)} = -\frac{f(t)}{S(t)}$

- *h*(*t*) is non-negative

Recap on censoring

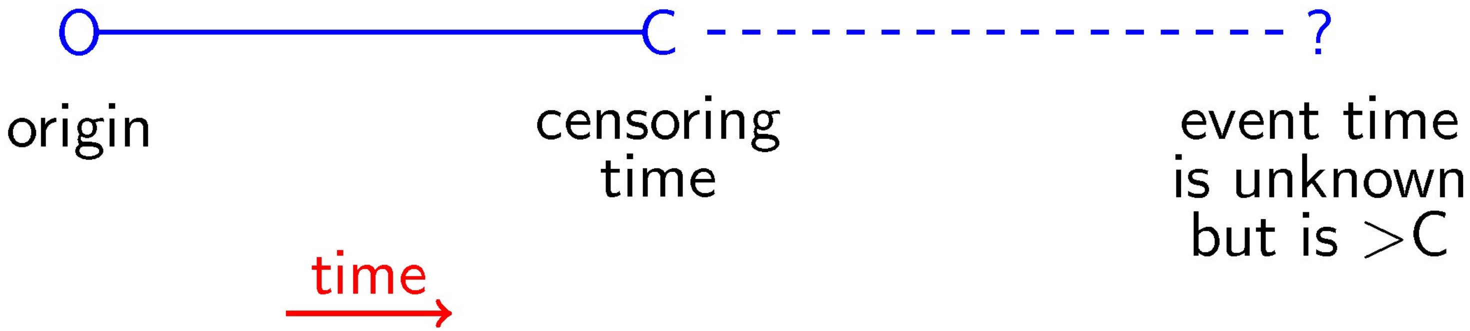
- An important concept in survival analysis is ‘censoring’.
- We will typically not follow all trial subjects until they die (or whatever the event under consideration is)
- Some patients are alive at the end of study or drop out
- Sometimes the event may never occur, e.g. cancer, if patient experiences remission

Censoring

- When we cannot observe the time of occurrence of an event, we may still obtain partial information of the form:

“the event had not occurred by time C”

- This is formally known as “right-censoring” since the right-hand end of the lifetime is unknown.



Example revisited

A histochemical marker (here ‘HPA’) discriminates between tumours that have metastasized and those that have not – can it predict survival?

23	47	69	70*	71*	100*	101*
148	181	198*	208*	212*	224*	
5	8	10	13	18	24	26
26	31	35	40	41	48	50
59	61	68	71	76	105*	107*
109*	113	116	118	143	154*	162*
188*	212*	217*	225*			

Blue (lines 1–2) – patients with HPA negative tumours; Red (lines 3–7) – HPA positive tumours. *indicate no event occurs

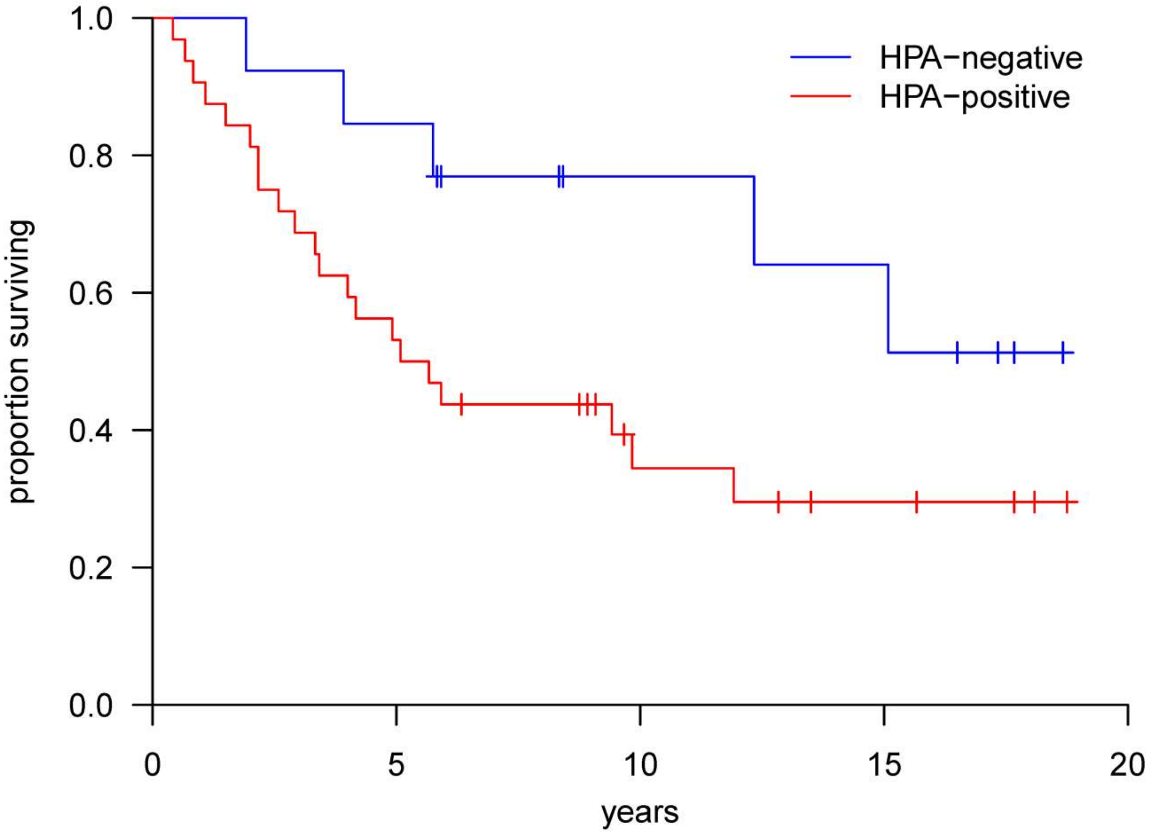
Kaplan-Meier estimator

- The survival function can be estimated with the Kaplan-Meier (KM) estimator.
- The KM estimator takes into account right-censoring.
- A plot of the Kaplan-Meier estimate is a series of horizontal steps of declining magnitude which approximate the true survival function (an underlying curve).
- Can use package “survival” in R

Defining a survival object in R

- package: survival
- Surv(*time*, *time2*, *event*, *type*)
  - *time* is the follow-up time (right censored data); or starting time (interval censored data)
  - *time2* is not used (right censored data); or ending time (interval censored data)
  - *event* defines the outcome: 1 = event occured at *time*; 0 = right censored (right censored data); 2 = left censored; 3 = interval censored (interval censored data)
  - *type* specifies the type of censoring, e.g. “right”, “left”, “interval”
- survfit() to obtain KM estimates
  - using the survival object as dependent variable

Kaplan-Meier estimates for 2 groups



```
hpa.km2 <- survfit(Surv(time,
event, type="right")~staining,
data=hpa)

plot(hpa.km2, col=c(4,2),
conf.int=F)
```

Patients with positive staining had worse prognosis.

Comparing Kaplan-Meier estimates for 2 groups

- Are the Kaplan-Meier curves for two groups *significantly* different?
- The log-rank test gives the relevant p-value.
- For the HPA data it can be run using

```
survdiff(Surv(time, event, type="right")~staining, data=hpa)
```

- For the HPA data the p-value is 0.06, so the survival functions are not significantly different between the two groups

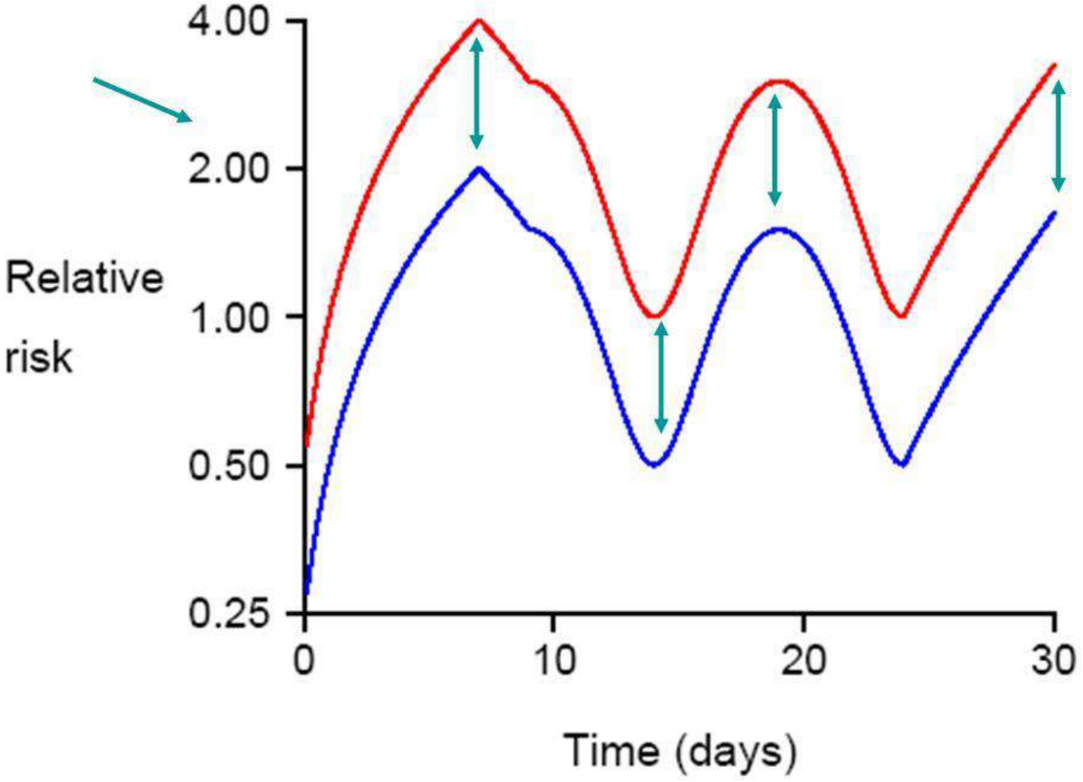
Cox model

The proportional hazards (Cox) model

- Regression model for survival data
- Each explanatory variable has a proportional effect on the hazard of death, compared to the hazard in the reference group.
- The absolute risk (hazard) may change over time, but the proportional difference between groups should stay the same.
- $h_1(t) = exp(\beta_1)h_0(t)$ for group 1 vs group 0.
  - $exp(\beta_1)$ is the hazard ratio (similar to relative risk).
- General form: $h_i(t) = exp(\beta \boldsymbol{x}_i)h_0(t)$
- Need at least 10 observed events per factor in a multivariable model (right-censored events don't count)

The proportional hazards assumption

- The PH model assumes that the relative risk between the 2 groups is constant through time, regardless of changes in absolute risk.
- No assumption on the baseline hazard



Data requirements and model checking

- Can check the PH assumption with a “complementary log-log”-scaled KM plot (look for parallel lines between groups)
  - PH assumption: $h_1(t) = ch_0(t) \; \leftrightarrow H_1(t) = cH_0(t)$, where $H(t)$ is the cumulative hazard
  - $\rightarrow -\log S_1(t) = -c \log S_0(t) \;\; (slide\ 6) \;\; \rightarrow \log(-\log S_1(t)) = \log(c) + \log(-\log S_0(t))$
  - if PH assumption holds, we expect a vertical shift when plotting $\log(-\log S(t))$ against t or log t
- Residual plot (Schoenfeld residuals over time)
- Can also check the PH assumption with a time dependent covariate (if it is significant, the relative hazard isn't constant through time).

Fitting a model

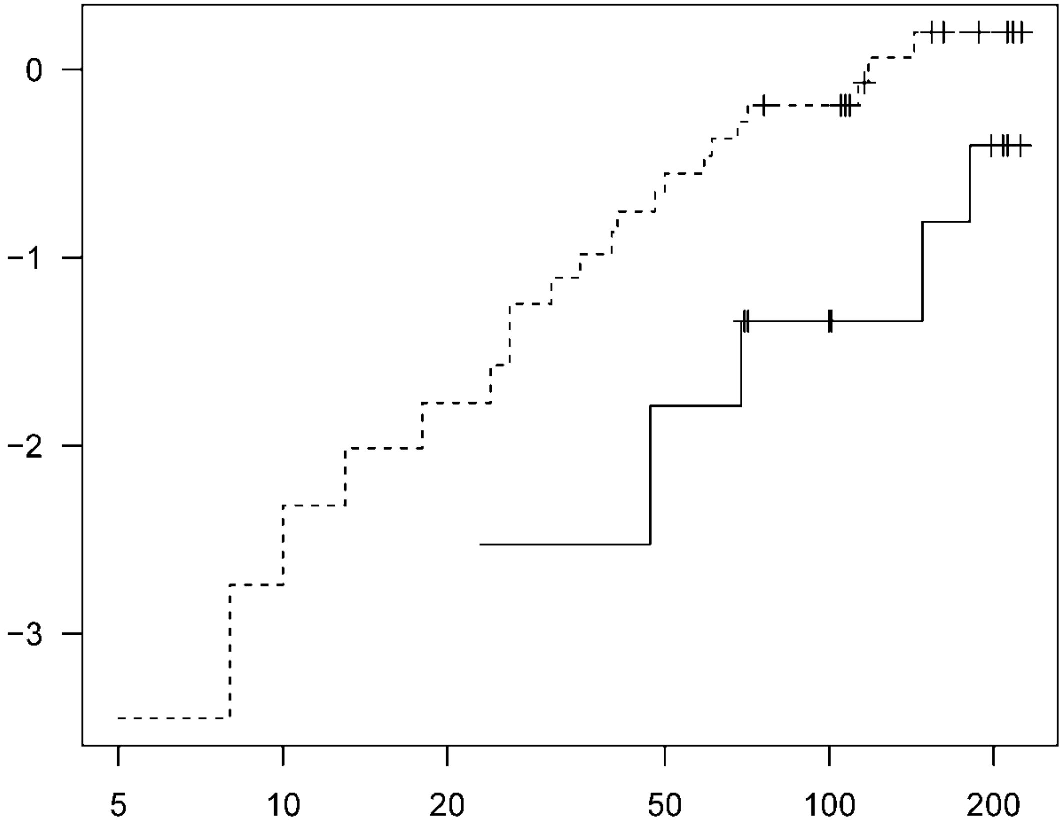
```
hpa.cox <- coxph(Surv(time, event)~staining, data=hpa)

hpa.cox

summary(hpa.cox)


           coef exp(coef) se(coef)     z Pr(>|z|)

staining 0.9093    2.4827   0.5009 1.815   0.0695 .

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


         exp(coef) exp(-coef) lower .95 upper .95

staining     2.483     0.4028    0.9302     6.626
```

Positive HPA staining insignificantly associated with increased risk of death (HR 2.48; 95% CI: 0.93 to 6.63; p-value: 0.07)

Model diagnostics

- To test the key assumption of proportional hazards

```
plot(hpa.km2, fun="cloglog", lty=1:2, mark.time=T)
```



Looks more or less parallel except near the end

Diagnostics

- cox.zph() creates interactions with time for testing the PH assumption
- based on Schoenfeld residuals
  - specific to each covariate
  - based on the partial likelihood

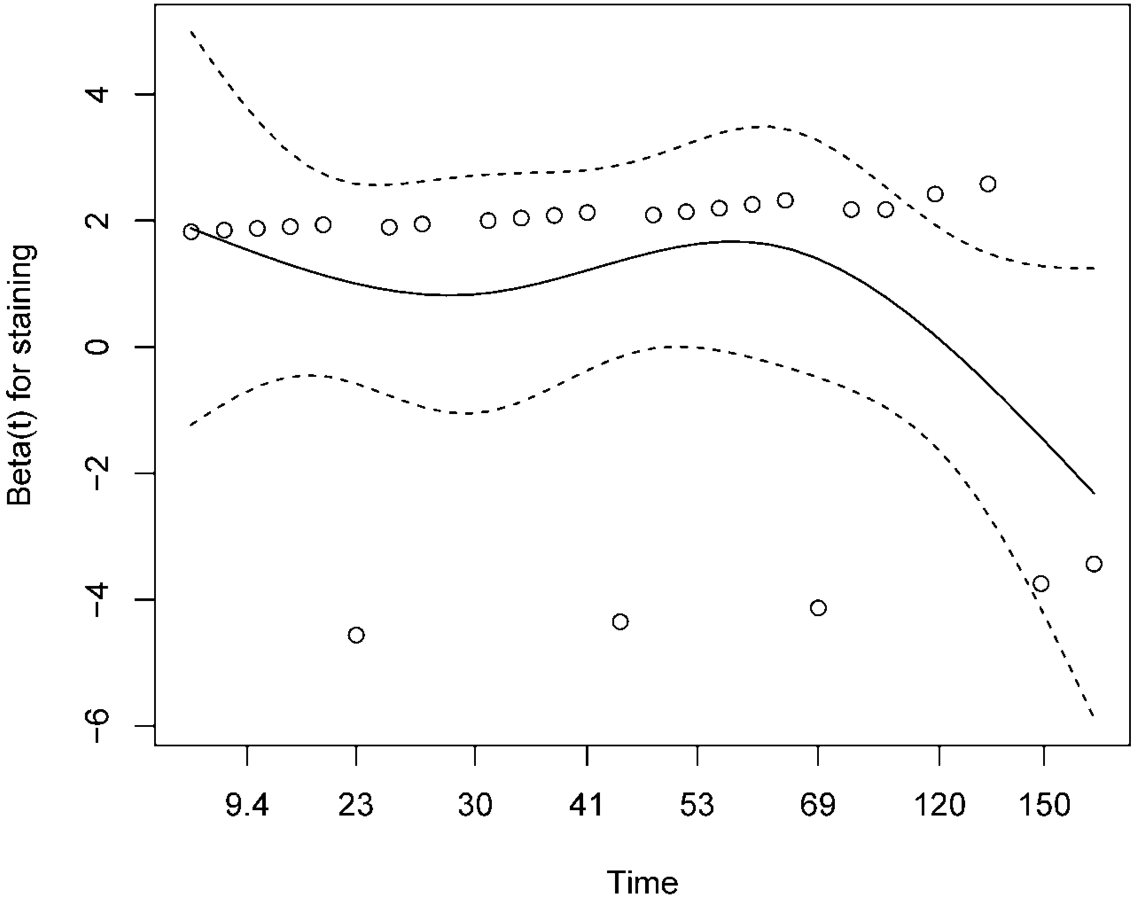
```
hpa.cox.zph <- cox.zph(hpa.cox)

hpa.cox.zph
```

```
         chisq  df    p

staining  1.32  1   0.25
```

- No significant interaction with time

Diagnostics

```
plot(hpa.cox.zph)
```



No obvious pattern over time

If PH assumption fails…

- Perform a stratified analysis

- Include interactions with time

- Include time-varying covariates

Accelerated failure time models

Accelerated failure time (AFT) model

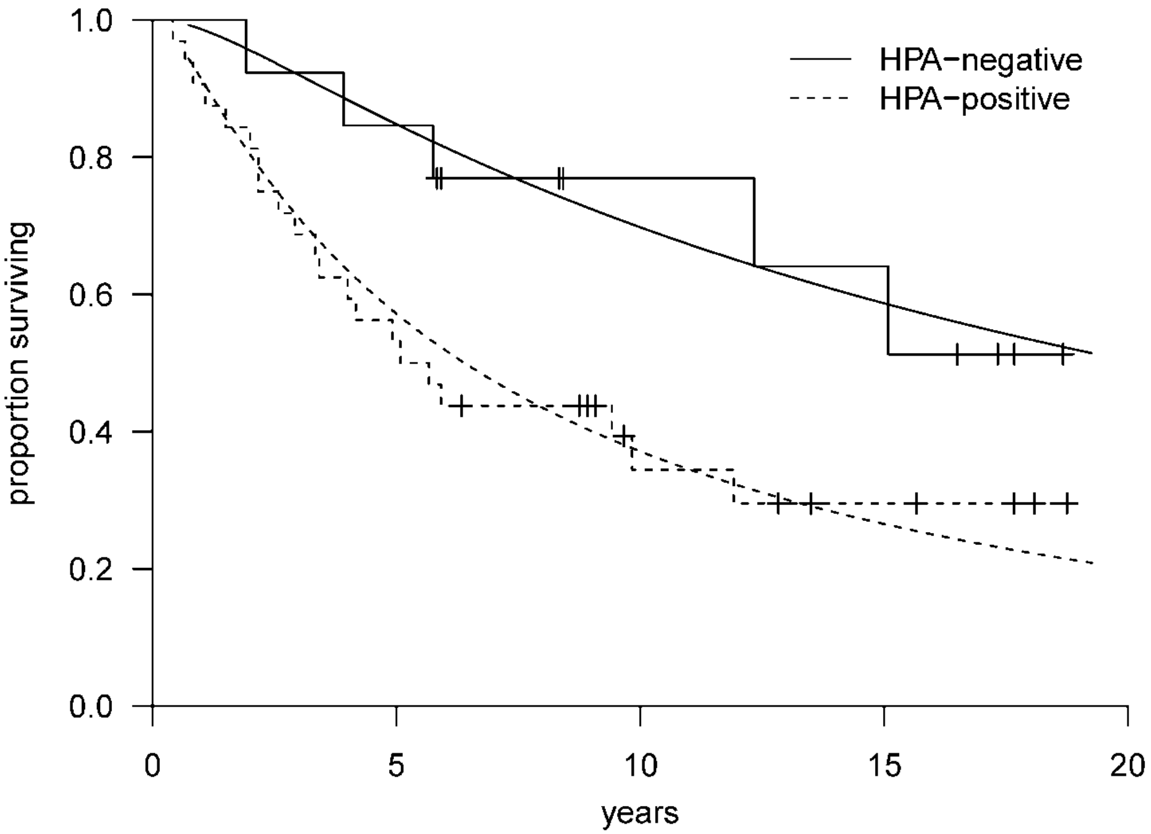
- In survival analysis, the parametric AFT model is an alternative to the proportional hazards model.
- $\log T = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon$, where *T* is the survival time, and ε follows some error distribution.
  - Some common distributions for *T*: lognormal, weibull, exponential
- exp(β_i) is the 'acceleration factor' associated with x_i
- Acceleration factors are the proportional increase (deceleration) or decrease (acceleration) in the median time to event.
  - Median is the preferred summary measure for survival data because survival time is usually right-skewed
- survreg() to fit parametric model in R

Fitting a model

```
hpa.aft1 <- survreg(Surv(time, event)~staining,
data=hpa, dist="lognormal")
hpa.aft1
Coefficients:
(Intercept)    staining
   5.491726   -1.151172
Scale= 1.359451

exp(coef(hpa.aft1)[2])
exp(confint(hpa.aft1))
```

Positive HPA staining associated with faster death (AF 0.32; 95% CI: 0.11 to 0.88)

Fitting a model



```
plot(hpa.km2, lty=1:2)
curve(1-plnorm(x, meanlog=hpa.aft1$coef[1], sdlog=hpa.aft1$scale), add=TRUE, lty=1)
curve(1-plnorm(x, meanlog=hpa.aft1$coef[1]+hpa.aft1$coef[2], sdlog=hpa.aft1$scale),
add=TRUE, lty=2)
```

Model comparison and checking

- Compare AICs or log-likelihoods of alternative parametric models

- Plot fitted curves against K-M estimates

AFT vs PH models

- PH models:
  - semi-parametric model (non-parametric part: $h_0(t)$; parametric part: $exp(\beta \boldsymbol{x}_i)$)
  - more widely used
  - flexible by not restricting shape of baseline hazard
  - interpretation in terms of higher / lower risk of event
- AFT models:
  - parametric model
  - give more precise estimates if they fit well (more powerful)
  - interpretation in terms of acceleration / deceleration of time to event

Censoring and truncation

Six individuals infected with HIV via transfusion

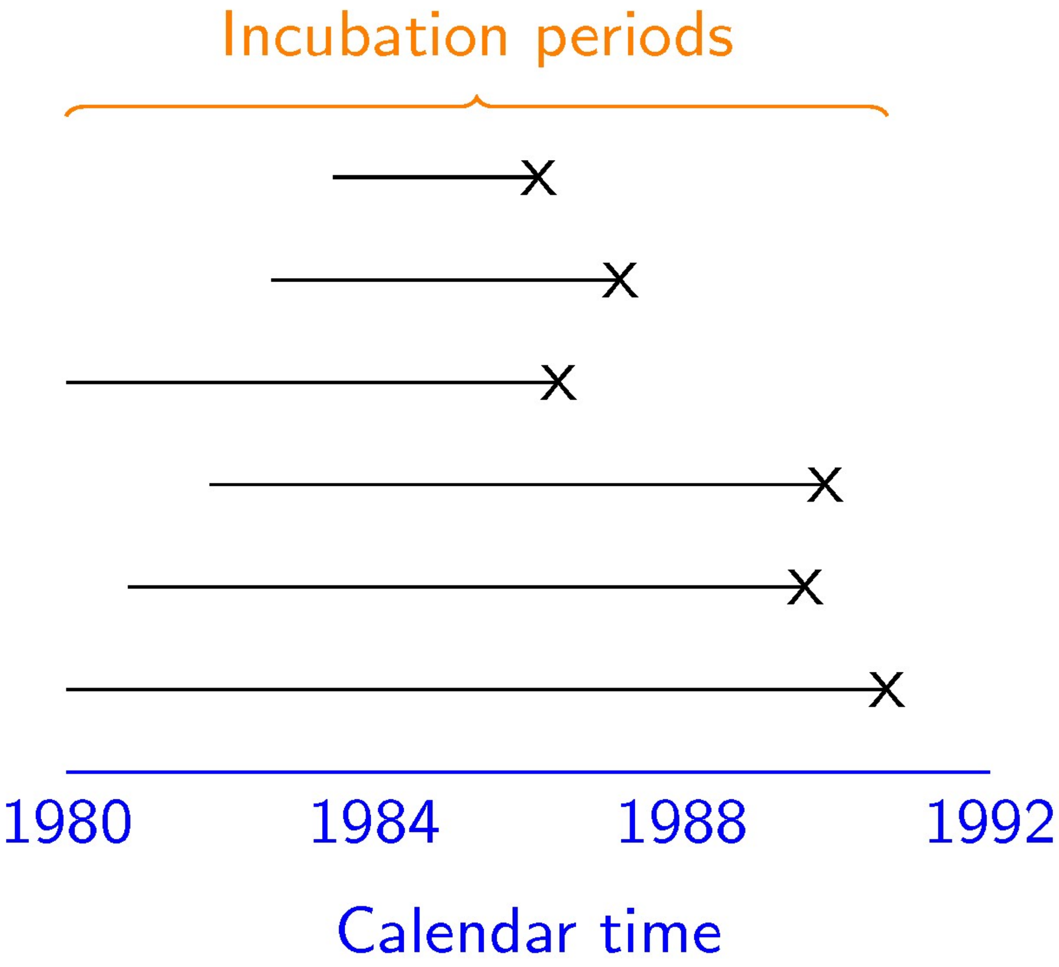


Figure: Infection and AIDS onset date exactly known.

Right censoring – analysis at 1988

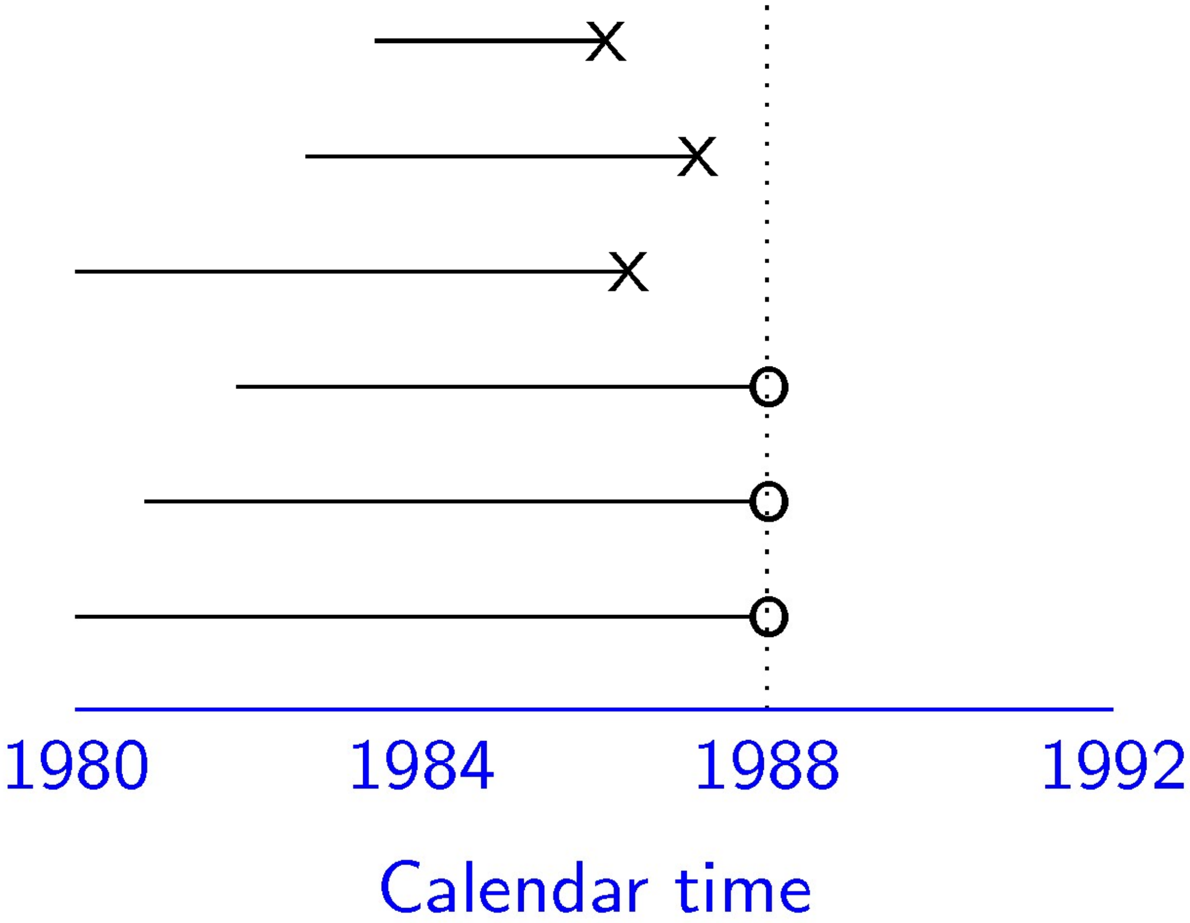


Figure: Some HIV-infected patients hadn't progressed to AIDS.

Right truncation – analysis at 1988

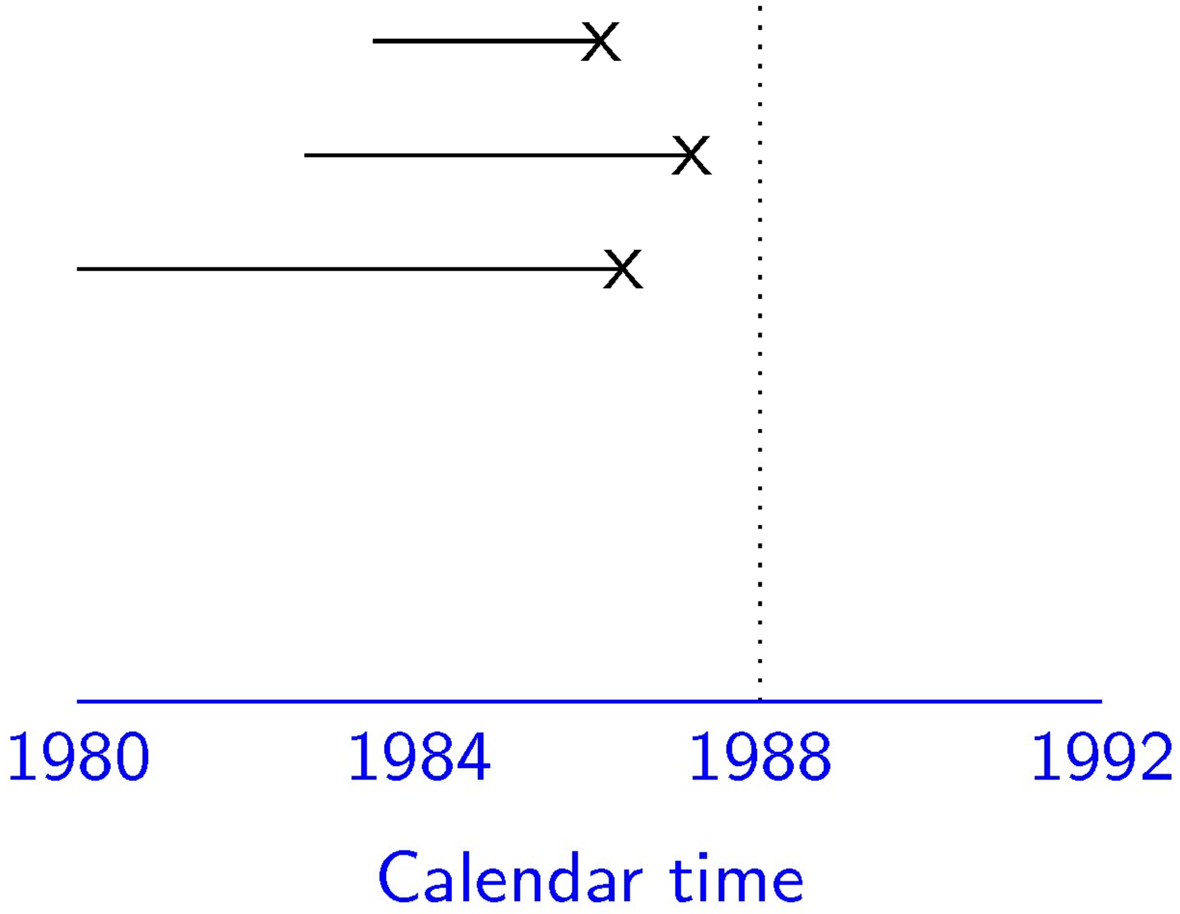


Figure: If we never knew about the patients without AIDS by 1988

Censoring vs truncation

- Under right censoring, we knew about the individuals with HIV but without AIDS at the analysis date.
  - We knew that there were some long incubation periods, but we didn't know exactly how long they were.
- Under right truncation, we never knew about the individuals not yet suffering from AIDS
  - They didn't appear in our dataset

Interval censoring – exact date of AIDS onset unknown

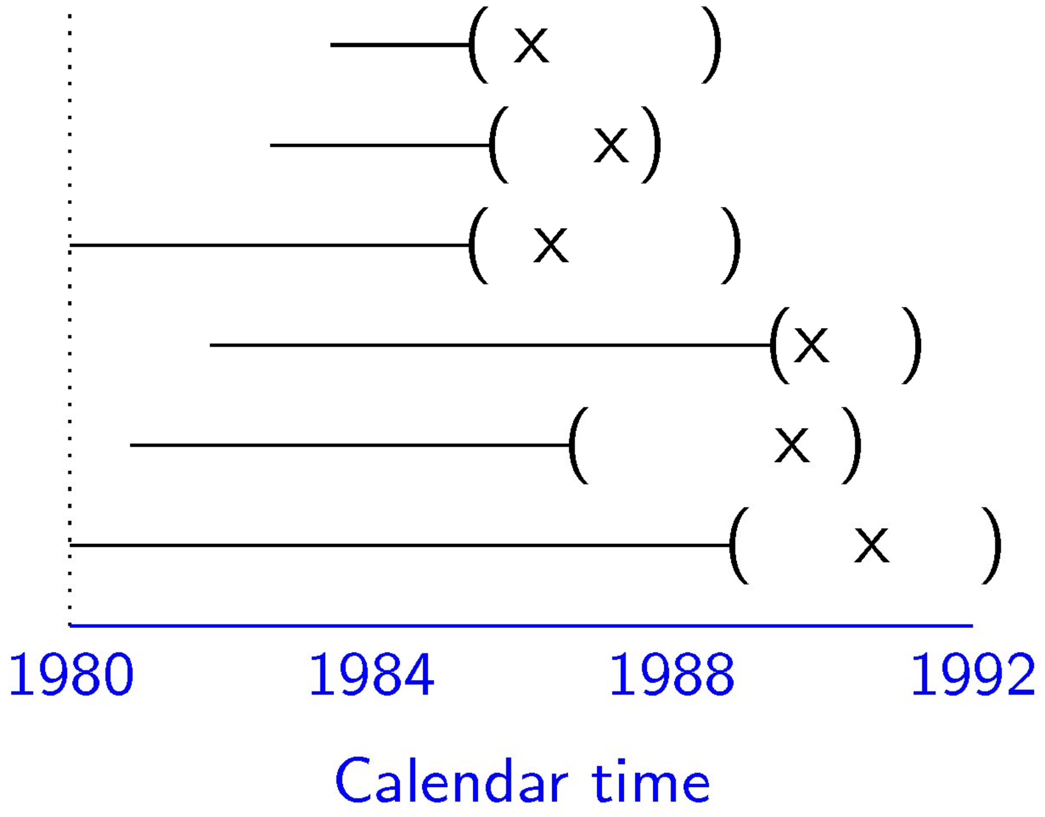


Figure: We don't know the exact date of AIDS onset.

Interval censoring

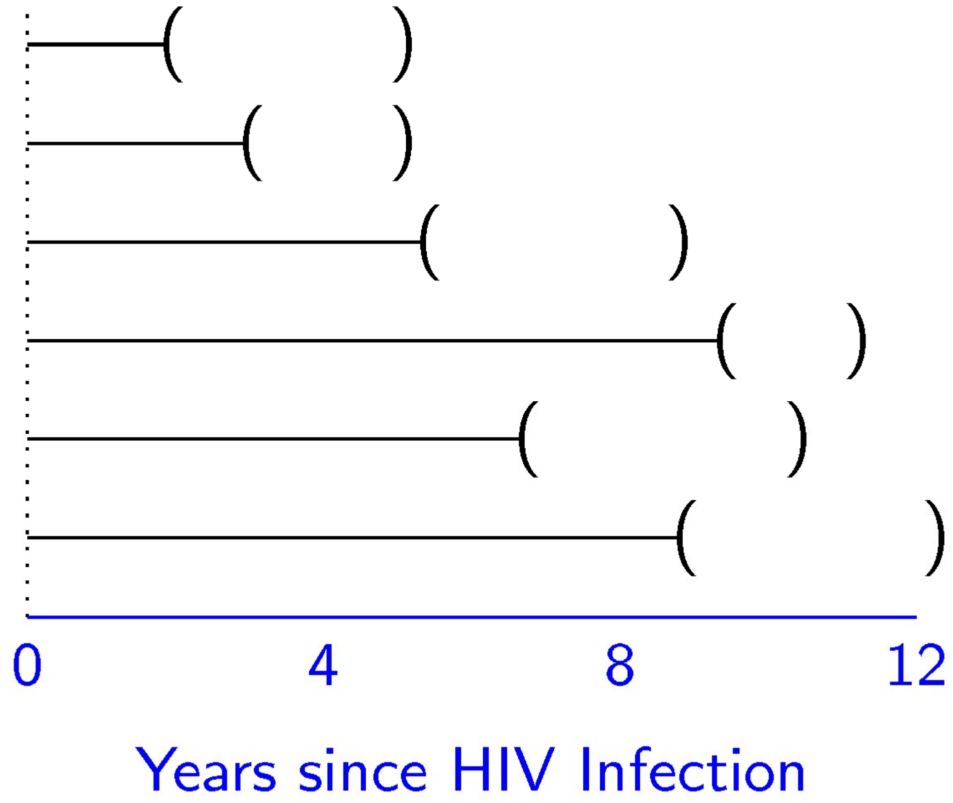


Figure: All we observe is the interval during which the event occurred.

Analysis of censored data

For parametric inference (with right censoring only), likelihood function is

$$l(\theta) = \prod_{i=1}^{n} f(t_i|\theta)^{\delta_i}\left(1 - F(t_i|\theta)\right)^{1-\delta_i}$$

where $f(\cdot)$ is the (assumed known) pdf, $F(\cdot)$ is the cdf, and δ_i is the event indicator ($\delta_i = 1$ for uncensored, $\delta_i = 0$ for censored).

Likelihood values for censored data

- Consider the value of the likelihood from each individual observation $i = 1, \cdots, n$.
- For exactly-observed times t_i, the value was $f(t_i)$.
- For right-censored times c_i, the value is $\{1 - F(t_i)\}$.
- For times t_i with right-truncation at *T*, the value is given by $f(t_i)/F(T)$.
- For times censored on the interval (L_i, R_i) the value is given by $\{F(R_i) - F(L_i)\}$.

Fitting parametric models

- To fit parametric (e.g. lognormal) distributions to censored or truncated data, simply multiply together the values from each observation.
- Then maximise the likelihood to estimate the distribution.
- Note that we won't get the same answer if we simply take the midpoint of any censoring intervals.
- R command `survreg` can handle interval-censored data.
  - syntax like:

```
survreg(Surv(timeL,timeR,event=3,type="interval")~1)
```

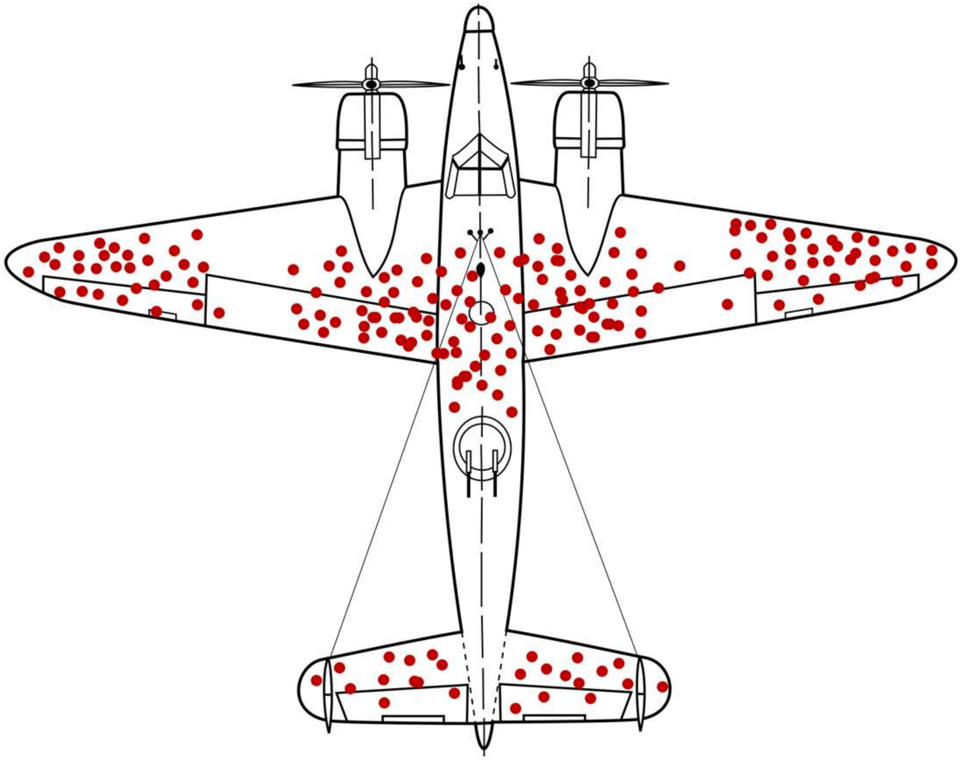
Review

- Survival analysis handled data with right censoring, right truncation or interval censoring
- Cox regression model allows flexible analysis without specifying the baseline hazard
- Accelerated failure time model is a parametric method as an alternative way to analyze survival data

Multiple imputation

Survivorship bias

During World War II, the statistician Abraham Wald examined the damage done to aircraft that had returned from missions and recommended adding armor to the areas that showed the least damage. The bullet holes in the returning aircraft represented areas where a bomber could take damage and still fly well enough to return safely to base.



From Wikipedia

Types of missing data

- Missing completely at random (MCAR)
  - Missingness is independent of all other variables
  - e.g. Accidental loss of data
- Missing at random (MAR)
  - Missingness is independent of unobserved variables
  - e.g. Loss of contact of recovered patients
- Missing not at random (MNAR)
  - Missingness is dependent on unobserved variables
  - e.g. Patients with high BMI less likely to respond to treatment, but the study did not record the BMI

Methods for missing data

- Complete case analysis
  - May lose substantial information
  - Only valid for MCAR
- Nearest neighbor imputation
  - Impute missing value from most similar subject
  - Valid for MCAR
- Mean imputation
  - Impute missing values by the mean of the variable
  - Reduce variation in the data
  - May not maintain associations between variables
  - Valid for MCAR

Methods for missing data

- Regression imputation
  - Predict missing value from regression model, like linear regression or GLM
  - Valid for MAR
- Inverse probability weighting
  - Estimate the probability of response based on some external knowledge
  - Weight the observed data using the inverse probability
  - Valid for MAR
- Multiple imputation
  - Single imputation treats imputed values as actual responses
  - Multiple imputation accounts for variability / uncertainty in the imputed data
  - Valid for MAR

Example

Complete case analysis (default for lm, glm and gam)

```
mvc <- read.csv(”YOUR PATH/mvc.csv")

mvc.miss <- mvc

mvc.miss$age[1:10] <- NA

lm.miss <- lm(MVC~age+height, data=mvc.miss)

summary(lm.miss)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -466.576    441.785  -1.056  0.29994
age           -6.197      1.755  -3.532  0.00145 **
height         6.386      2.352   2.716  0.01120 *
…
Residual standard error: 80.57 on 28 degrees of freedom
  (10 observations deleted due to missingness)
```

Example

- Mean imputation

```
mvc.miss$age[1:10] <- NA
mvc.miss$age[1:10] <- mean(mvc.miss$age, na.rm=T)
```

- Regression

```
mvc.miss$age[1:10] <- NA
lm.impute <- lm(age~height, data=mvc.miss)
mvc.miss$age[1:10] <- predict(lm.impute,
mvc.miss)[1:10]
```

Multiple imputations

- Impute missing variable to form *m* complete datasets

- Perform the analysis to obtain estimates from each of the *m* datasets

- Combine the *m* estimates to obtain the overall estimates
  - practically *m* = 10, 20 or 50 is sufficient

Multiple imputation in R

- package: Hmisc
- Construct *n.impute* imputed datasets:
- transcan(*formula*, *n.impute*, *shrink*, *data*, *imputed*)
  - *formula* for imputation (not model fitting), should be like $\sim y + x_1 + x_2$
  - *n.impute* is the number of imputations
  - *shrink* = *T* to avoid overfitting for imputation
  - *data* should refer to a data frame
  - *imputed* = *T* to save the imputed values

Multiple imputation example (transcan)

```
mvc.miss <- mvc

mvc.miss$age[1:5] <- mvc.miss$height[6:10] <- NA

require(Hmisc)

mvc.impute <- transcan(~MVC + age + height, n.impute=50,

shrink=T, data=mvc.miss, imputed=T)

mvc.impute$imputed$age
```

```
         1        2        3        4        5        6        7        8
1 47.00000 43.25055 37.06274 47.00000 37.06274 47.00000 47.88888 47.09355
2 43.28446 44.03811 55.77490 58.23814 55.69904 34.82969 47.00000 39.79569
3 41.82874 51.03184 36.48883 47.00000 47.00000 31.91915 50.50301 31.91915
4 47.00000 47.00000 47.00000 47.00000 34.67497 34.28352 47.00000 34.68664
5 56.84882 60.40867 61.09107 60.97119 61.32033 58.98520 63.66946 55.72228
         9       10       11       12       13       14       15       16
1 47.00000 42.50380 34.25330 34.36322 47.00000 44.51422 47.00000 34.56837
2 55.69904 41.35587 37.12405 31.24612 41.17544 55.46087 41.35587 36.99878
3 31.06728 41.82874 47.00000 40.21853 51.03184 47.00000 51.03184 40.22834
4 43.18182 43.69245 47.00000 46.24437 47.00000 47.00000 47.00000 47.00000
5 61.72836 61.56117 58.98520 61.56117 63.41946 63.66946 60.75684 62.32885
```

Multiple imputation example (aregImpute)

```
mvc.impute.areg <- aregImpute(~MVC + age + height, n.impute=50,
data=mvc.miss)
mvc.impute.areg$imputed$age
```

```
[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]
1   47   53   32   35   31   35   31   62   31    39    31    55    55    32    32    53    53    32    31
2   40   34   32   31   47   47   61   31   35    47    50    39    39    31    49    48    31    47    47
3   40   31   39   53   53   49   55   34   31    34    49    38    55    31    34    35    51    32    55
4   40   31   34   35   55   34   49   34   31    53    40    55    50    31    31    38    53    41    49
5   53   31   31   31   32   53   65   62   62    34    53    53    32    58    65    37    47    32    65
  [,20] [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
1    35    53    31    50    47    39    55    38    31    32    31    49    32    35    31    32    47    39
2    47    31    32    35    50    47    32    47    32    47    31    47    53    55    61    53    32    34
3    53    32    53    39    31    41    38    55    41    31    49    38    32    55    49    55    55    40
4    40    55    31    51    39    53    38    34    35    55    35    34    32    55    32    41    32    53
5    65    62    62    53    62    32    31    65    65    65    32    65    47    58    61    65    31    37
```

Transcan vs aregImpute

- Non-integer output (Transcan) vs Integer output (aregImpute)
- `transcan` function performs imputation using single regression models for each variable with missing values.
- Integer input in regression does not ensure integer prediction
- `aregImpute` function uses additive regression, bootstrapping, and predictive mean matching to impute missing values.
- When the original data is integer, the `aregImpute` function uses predictive mean matching, which matches the predicted values from the regression model to the observed values in the dataset.

Combining estimates from multiple imputation

- The final MI estimate Q from m imputations is given by

  $$\bar{Q} = \frac{1}{m} \sum_{i=1}^{m} \hat{Q}^{(i)}$$

  with variance

  $$T = \bar{U} + \left(1 + \frac{1}{m}\right) B$$

  where B and $\bar{U}$ are between- and within-imputation variances:

  $$B = \frac{1}{m-1} \sum_{i=1}^{m} (\hat{Q}^{(i)} - \bar{Q})^2$$

  $$\bar{U} = \frac{1}{m} \sum_{i=1}^{m} \hat{U}^{(i)}$$

- $(\bar{Q} - Q)/\sqrt{T}$ follows a t-distribution

R function for the above process

- package: Hmisc
- refit the model based on the imputed datasets
- Obtain the overall estimates
- fit.mult.impute(*formula*, *fitter*, *transcan*, *data*)
  - *formula* should be like $y{\sim}x_1 + x_2$
  - *fitter* is the model to fit the data, e.g. lm, glm (for glm, the family option can be included)
  - *transcan* class object created in the last slide
  - *data* should refer to a data frame

Multiple imputation example

```
mvc.lm.impute <- fit.mult.impute(MVC ~ age + height, lm,
mvc.impute, data=mvc.miss)
summary(mvc.lm.impute)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -782.622    367.133  -2.132 0.039556 *
age           -4.003      1.430  -2.799 0.008004 **
height         7.563      2.027   3.730 0.000623 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 86.96 on 38 degrees of freedom
Multiple R-squared:  0.4233,    Adjusted R-squared:  0.3929
F-statistic: 13.94 on 2 and 38 DF,  p-value: 2.876e-05
```

Multiple imputation example

```
• mvc.lm.impute.areg <- fit.mult.impute(MVC ~ age + height, lm,
  mvc.impute.areg,data=mvc.miss)
• summary(mvc.lm.impute.areg)

• Coefficients:
•             Estimate Std. Error t value Pr(>|t|)
• (Intercept) -651.802    432.211  -1.508  0.13981
• age           -3.907      1.536  -2.544  0.01516
• height         6.761      2.430   2.782  0.00836
•
• Residual standard error: 95.72 on 38 degrees of freedom
• Multiple R-squared:  0.3165,   Adjusted R-squared:  0.2805
• F-statistic: 8.797 on 2 and 38 DF,  p-value: 0.0007252
```