Survival Analysis and Machine Learning

Joseph Hogan Ann Mwangi Richard Mugo
Amos Okutse Allison DeLong Ziv Shkedy

Brown University, Moi University, University of Hasselt

14-17 July 2025
Eka Hotel Conference Center
AMPATH
Eldoret, Kenya

Funding: Fogarty Institute of the NIH, U Hasselt

Basics of event-time data (survival data)

What kinds of endpoints are event time data?

- Studies of mortality – time to death
- Studies of disease process – time to progression, time to recovery
- Studies with combined outcomes – time to MI or heart failure

Sometimes called **survival data** or **failure time data**

Overview

- Description of a study of HIV-related mortality
- What are event-time data and what makes them different?
- Representing and summarizing event-time data
- Regression models for event-time data

Bloomfield *et al. BMC Infectious Diseases* 2014, **14**:284
http://www.biomedcentral.com/1471-2334/14/284

BMC
Infectious Diseases

RESEARCH ARTICLE Open Access

Blood pressure level impacts risk of death among HIV seropositive adults in Kenya: a retrospective analysis of electronic health records

Gerald S Bloomfield[1,2,3*], Joseph W Hogan[4,5], Alfred Keter[5,6], Thomas L Holland[7], Edwin Sang[5,6], Sylvester Kimaiyo[5,6] and Eric J Velazquez[1,2,3]

Study of risk factors for HIV-related mortality

Longitudinal study of 50,000 people living with HIV (PLWH) in Kenya between 2005 and 2010

Key objective: Assess the effect of blood pressure on mortality

Secondary objectives:

- Assess impact of other markers of co-morbidities (diabetes risk, liver disease)
- Estimate short- and long-term survival rates

Study of risk factors for HIV-related mortality

Primary endpoint: Time from enrollment to death

Covariates (predictors, features):

- SBP, DBP
- Gender
- BMI
- Hemoglobin (marker of CVD risk, stroke etc.)
- Creatinine (kidney function marker)
- CD4 count at enrollment (immune function marker)
- ART status at enrollment
- WHO disease stage at enrollment (1 to 4)
- Marital status (yes/no)
- Urban/rural
- Advanced HIV (yes/no)

Table 1 Summary of characteristics by gender

Variable	Women (n = 36616)	Men (n = 12859)	Overall (n = 49475)
Age, median (IQR), years	32 (26-39)	38 (31-46)	33 (27-41)
Age category, No. (%), years			
<25	7473 (20.4)	812 (6.3)	8285 (16.8)
25-34	15227 (41.6)	4183 (32.5)	19410 (39.2)
35-44	9049 (24.7)	4362 (33.9)	13411 (27.1)
45-54	3739 (10.2)	2447 (19.0)	6186 (12.5)
55-64	948 (2.6)	803 (6.2)	1751 (3.5)
≥65	180 (0.5)	252 (2.0)	432 (0.9)
SBP, median (IQR), mmHg	110 (100-120)	110 (100-120)	110 (100-120)
DBP, median (IQR), mmHg	70 (60-72)	70 (60-79)	70 (60-74)
SBP category, No. (%), mmHg			
<100	3822 (10.4)	1083 (8.4)	4905 (9.9)
100-119	21145 (57.8)	6444 (50.1)	27589 (55.8)
120-139	10145 (27.7)	4488 (34.9)	14633 (29.6)
≥140	1504 (4.1)	844 (6.6)	2348 (4.8)
DBP category, No. (%), mmHg			
<60	1748 (4.8)	571 (4.4)	2319 (4.7)
60-79	27702 (75.7)	9107 (70.8)	36809 (74.4)
80-89	5905 (16.1)	2620 (20.4)	8525 (17.2)
≥90	1261 (3.4)	561 (4.4)	1822 (3.7)

BMI, median (IQR), kg/m^{2a}	21.5 (19.3-24.0)	20.1 (18.4-21.9)	21.0 (19.0-23.5)
BMI category, No. (%), kg/m^{2a}			
<18.5	5348 (16.8)	2855 (25.9)	8203 (19.1)
18.5 - <25	20645 (64.8)	7495 (67.9)	28140 (65.6)
25 - <30	4716 (14.8)	595 (5.4)	5311 (12.4)
≥30	1173 (3.7)	97 (0.9)	1270 (3.0)
Hemoglobin, median (IQR), g/dL[b]	11.8 (10.2-13.1)	13.8 (11.9-15.3)	12.2 (10.5-13.7)
Creatinine, median (IQR), µmol/L[c]	60 (51-71.4)	76 (64.5-89)	63.8 (53-77)
CD4 count, median (IQR), cells/mm^{3d}	413 (296-581)	363 (271-502)	399 (288-561)
CD4 category, no. (%), cells/mm^{3d}			
200-350	9705 (37.4)	4019 (46.8)	13,724 (39.8)
>350	16224 (62.6)	4566 (53.2)	20790 (60.2)
ART naïve at enrollment, No. (%)	33345 (91.1)	11645 (90.6)	44990 (90.9)
WHO stage at enrollment, No. (%)[e]			
Stage 1	17304 (60.5)	4803 (48.0)	22107 (57.2)
Stage 2	6647 (23.2)	2658 (26.6)	9305 (24.1)
Stage 3	4666 (16.3)	2547 (25.5)	7213 (18.7)
Urban	16970 (46.4)	6165 (47.9)	23135 (46.8)
Married/living with partner[f]	19045 (54.0)	9127 (73.1)	28172 (59.0)

Data used for this workshop

We created a *synthetic dataset* having 5000 individual records

- Distribution of variables mimics that in the actual study
- None of the individual records is identical to an actual patient record
- Missing data patterns have been preserved

R code for importing and summarizing data

```
# Import the data
# load the data into your working directory
load("Workshop2025.Rdata")
```

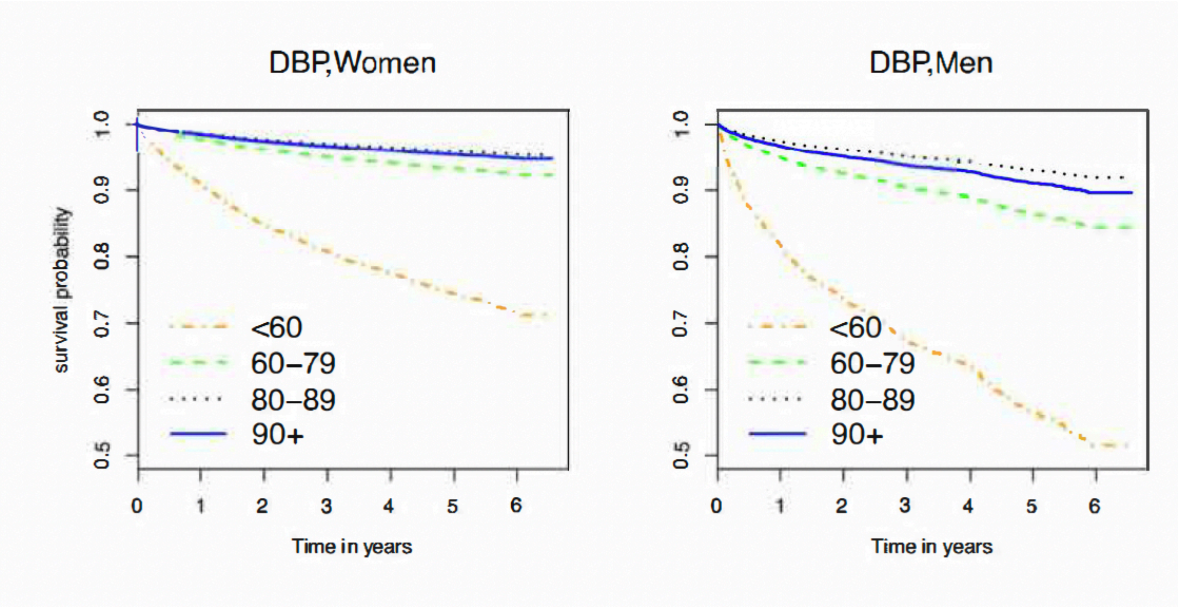
```
# Explore the data structure using str()
str(htndat)
View(htndat)
```

Refer to R code for ways to prepare summaries of the data using `gtsummary`

Summaries of mortality data

- Survival curves
- Hazard functions
- Event rates per person-time follow up
- Medians, percentiles

Survival curves



Event rates

Table 2 Unadjusted mortality rates per 100 person years by blood pressure groups for men and women with and without advanced HIV

Characteristic	Person years	Women events	Mortality rate (95% CI)	Person years	Men events	Mortality rate (95% CI)
Advanced HIV[a]						
Systolic blood pressure						
SBP <100 mmHg	3447	113	3.3 (2.7-3.9)	816	57	7.0 (5.4-9.1)
SBP 100-119 mmHg	18648	364	2.0 (1.8-2.2)	5966	219	3.7 (3.2-4.2)
SBP 120-139 mmHg	7954	116	1.5 (1.2-1.7)	4001	96	2.4 (2.0-2.9)
SBP ≥140 mmHg	1095	23	2.1 (1.4-3.2)	590	21	3.6 (2.3-5.5)
Diastolic blood pressure						
DBP <60 mmHg	1164	73	6.3 (5.0-8.0)	356	56	15.8 (12.1-20.5)
DBP 60-79 mmHg	23988	467	1.9 (1.8-2.1)	8142	283	3.5 (3.1-3.9)
DBP 80-89 mmHg	4913	58	1.2 (0.9-1.5)	2372	45	1.9 (1.4-2.5)
DBP ≥90 mmHg	1079	18	1.7 (1.1-2.6)	504	9	1.8 (0.9-3.4)

Important features of event time data

- All values are **positive** – cannot have negative time
- Some people have **partial information** due to **censoring**

Partial information and censoring

Notation

$$\begin{aligned} T &= \text{actual time to an event} \\ T^* &= \text{observed follow-up time} \\ \Delta &= \mathbb{I}(T = T^*) \end{aligned}$$

Types of observed information about T

- Full observed; e.g., $T = 45$
- Right censored; e.g., $T > 30$
- Interval censored; e.g., $26 < T < 40$

Our examples: T is either *observed* or *right-censored*

Representations of event history data

Follow up time and observation indicator

T^*	Δ	Interpretation
5	1	$T = 5$
8	1	$T = 8$
10	0	$T > 10$
12	1	$T = 12$
15	0	$T > 15$

Those with $\Delta = 0$ are **right-censored**

Information about T is not fully missing; it's **partially observed**

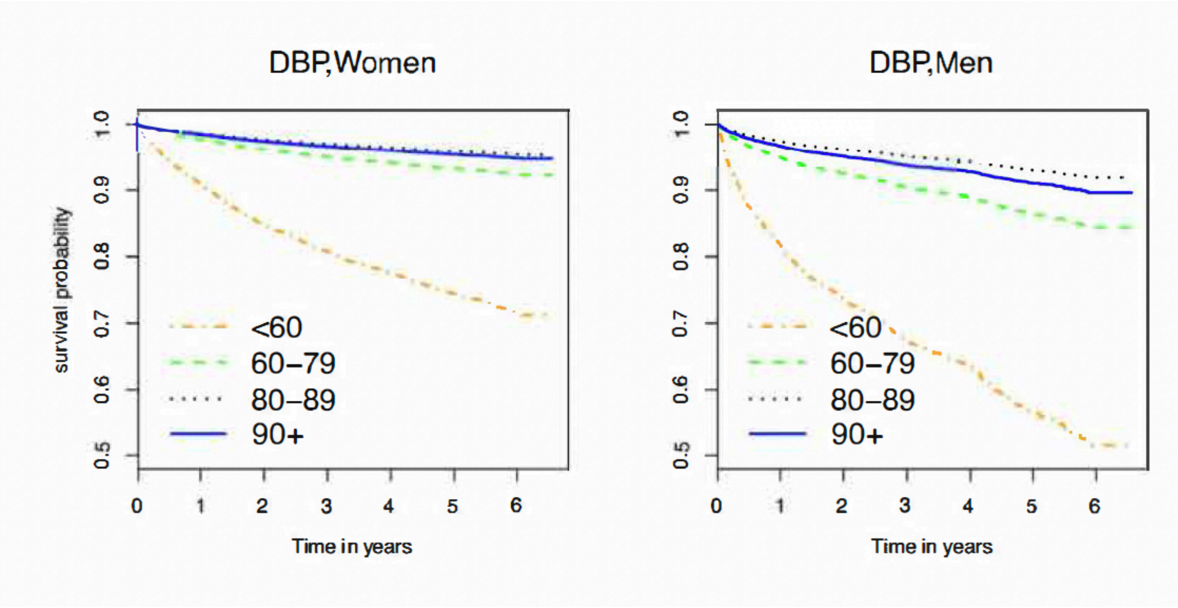
Summaries of event history data: Survivor function

Survivor function:

- proportion event-free as a function of time
- decreases over time as more events occur

$$S(t) \;=\; P(T \leq t)$$

Survival curves

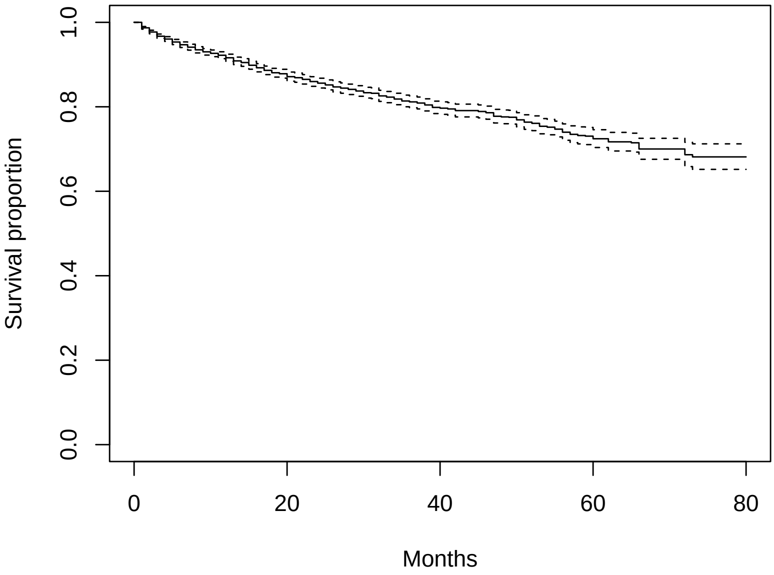


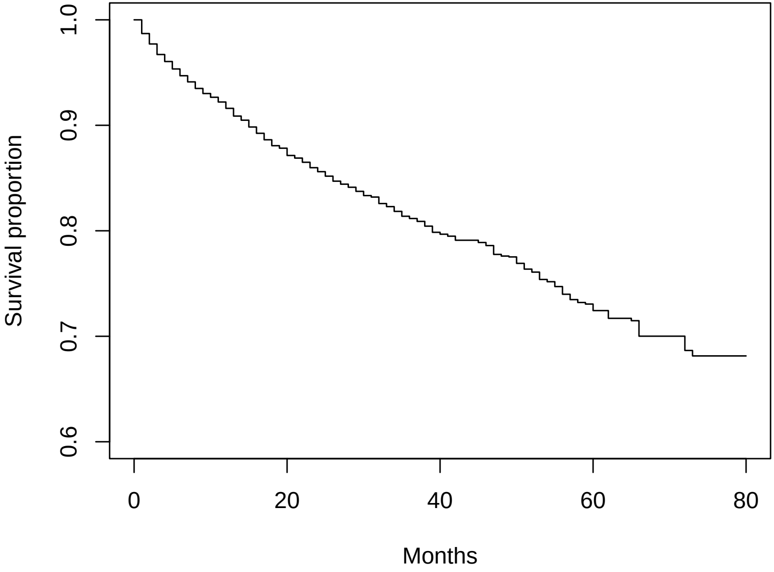
Calculating $\hat{S}(t)$: Kaplan-Meier Estimator

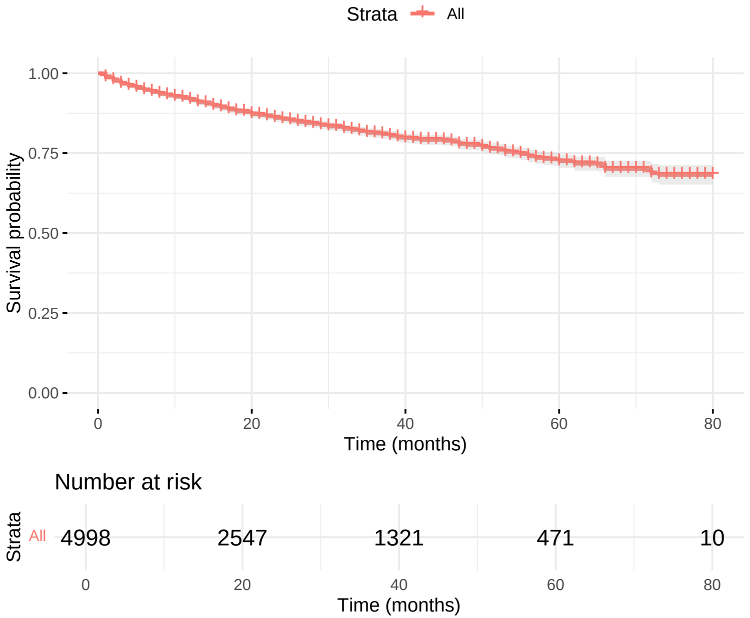
T^*	Δ	at risk	events	$\hat{S}(t)$
0		8	0	1.0
5	1	8	1	$(1-1/8) = .83$
8	1	7	1	$.83 \times (1-1/7) = .71$
10	0	6	0	.71
12	1	5	1	$.71 \times (1-1/5) = .57$
15	0	4	0	.57
16	0	3	0	.57
22	1	2	1	$.57 \times (1-1/2) = .29$
26	0	1	0	.29

Survival curves for HTN data

```
view(htndat)
view(htndat[c("survmonth", "event")])

# Create the survival object

surv_obj <- with(htndat, Surv(time = survmonth, event = event))

# Explanation:
#   - 'time'  = months from baseline to death or censoring
#   - 'event' = 1 if death occurred, 0 if censored

# Fit overall KM curve
km_overall <- survfit(surv_obj ~ 1, data = htndat)
print(km_overall)

# basic plot
plot(km_overall, xlab="Months", ylab="Survival proportion")
```


Survival proportion
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
Months


Survival proportion
1.0
0.9
0.8
0.7
0.6
0
20
40
60
80
Months


Strata
All
1.00
0.75
0.50
0.25
0.00
Survival probability
0
20
40
60
80
Time (months)
Number at risk
Strata
All
4998
2547
1321
471
10
0
20
40
60
80
Time (months)

Properties of the Kaplan-Meier estimator

- It's *nonparametric* – does not depend on a model
- It accounts for censoring, but under some assumptions
  - 'non-informative' censoring
  - event rate after censoring is equal that for those who remain in follow-up

'Non-informative censoring' is the assumption underlying all of our analyses today. It may not always hold in practice.

Summaries of event history data

Because of censoring, summaries like mean, SD, median, etc. cannot be calculated directly from the data

Typical approach

1. Calculate an estimate of $S(t)$
2. Use that estimate to derive various summaries

Example: Calculating the median

$$\text{median}(T) \quad = \quad \text{value of } t \text{ such that } S(t) = .5$$

Check understanding

What fraction survived to 20 months?

What fraction survived to 5 years?

What is the 75th percentile of the survival times?

What is the median survival time?

R code for basic summaries from KM curve

```
# summarize KM probs at each time
summary(km_overall)

summary(km_overall, times = 12)

quantile(km_overall, probs=c(.05, .10, .25, .5))

tbl_survfit(km_overall, probs=c(.05, .10, .25, .5))
```

R code for basic summaries from KM curve

```
> summary(km_overall)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   4998      65    0.987 0.00160        0.984        0.990
    2   4394      44    0.977 0.00217        0.973        0.981
    3   4210      43    0.967 0.00263        0.962        0.972
    4   4066      28    0.960 0.00290        0.955        0.966
    5   3955      29    0.953 0.00316        0.947        0.960
    6   3849      26    0.947 0.00338        0.940        0.954

   15   2947      21    0.898 0.00482        0.889        0.908
   16   2852      19    0.892 0.00498        0.883        0.902
   17   2764      19    0.886 0.00514        0.876        0.896
   18   2680      17    0.881 0.00528        0.870        0.891

   65    318       1    0.715 0.01142        0.693        0.738
   66    292       6    0.700 0.01266        0.676        0.725
   72    155       3    0.687 0.01464        0.658        0.716
   73    133       1    0.681 0.01541        0.652        0.712
```

Quantiles output

Output from

```
tbl_survfit(km_overall, probs=c(.05, .10, .25, .5))
```

Characteristic	5.0% Percentile	10% Percentile	25% Percentile	50% Percentile
Overall	6.0 (5.0, 7.0)	15 (13, 17)	55 (51, 60)	— (—, —)

Summaries of event history data: Hazard function

Hazard function:

- event *rate* as a function of time
- rate of event at t among those who have not yet had the event

$$h(t) \approx P(T = t \,|\, T \geq t)$$

Relationship between hazard function and survival function

$$\begin{aligned} h(t) &= -\frac{dS(t)/dt}{S(t)} \\ &\approx \frac{\text{slope of survival curve at } t}{\text{proportion survived up to } t} \end{aligned}$$

Calculating $\hat{h}(t)$

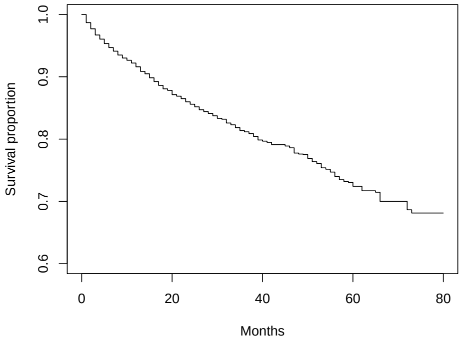
T^*	Δ	at risk	events	$\hat{h}(t)$
0		8	0	—
5	1	8	1	$1/8 = .13$
8	1	7	1	$1/7 = .14$
10	0	6	0	0
12	1	5	1	$1/5 = .20$
15	0	4	0	0
16	0	3	0	0
22	1	2	1	$1/2 = .50$
26	0	1	0	0

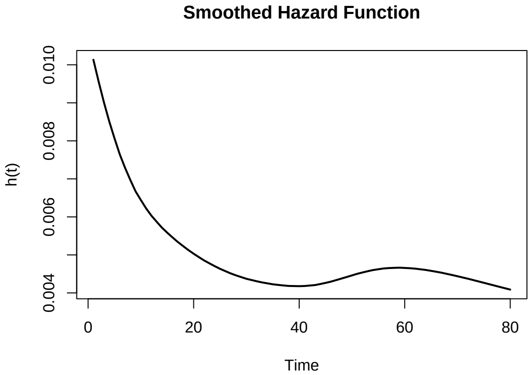
In real data, there can be more than one event at some time points – hence the hazard is not always increasing over time.

R code for calculating and plotting hazard function

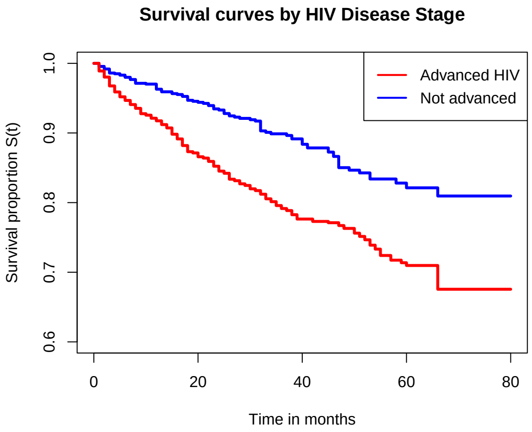
```
# Hazard functions
fit.hazard <- bshazard(surv_obj ~ 1, data = htndat)

# Extract smoothed hazard and log-hazard
time_vals  <- fit.hazard$time
hazard     <- fit.hazard$hazard
log_hazard <- log(hazard)

# Plot hazard with confidence bands
plot(time_vals, hazard, type = "l", lwd = 2,
     xlab = "Time", ylab = "h(t)", main = "Smoothed Hazard Function")

polygon(c(fit.hazard$time, rev(fit.hazard$time)),
        c(fit.hazard$upper, rev(fit.hazard$lower)),
        col = rgb(0, 0, 1, 0.2), border = NA)

# Plot log-hazard
plot(time_vals, log_hazard, type = "l", lwd = 2,
     xlab = "Time", ylab = "log h(t)", main = "Smoothed Log-Hazard Function")
```

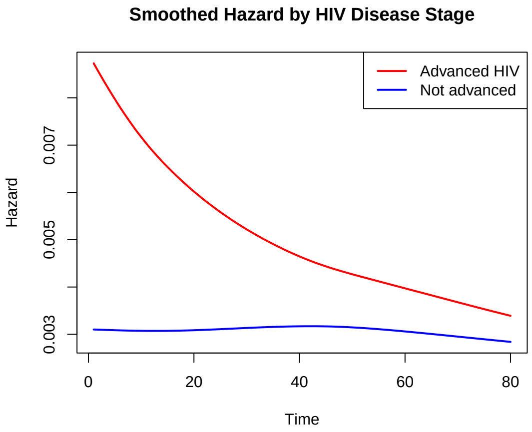
Hazard function vs survival function





Hazard function vs survival function





R code for calculating stratified survival curves

```
# Overall KM survival plot by advanced HIV status
km_adv <- survfit(surv_obj ~ factor(adv_HIV), data = htndat)

plot(km_adv, lwd=3, col=c("blue","red"), ylim=c(.6,1) ,
     xlab="Time in months", ylab="Survival proportion S(t)",
     main="Survival curves by HIV Disease Stage")

legend("topright",
       legend = c("Advanced HIV", "Not advanced"),
       col    = c("red", "blue"),
       lwd    = 2)
```

R code for calculating stratified hazard curves

```
## Fit smoothed hazard models for each level
fit_adv    <- bshazard(Surv(time = survmonth, event = event) ~ 1,
                       data = htndat[htndat$adv_HIV == 1, ])
fit_notadv <- bshazard(Surv(time = survmonth, event = event) ~ 1,
                       data = htndat[htndat$adv_HIV == 0, ])

## Common y-limits so the curves share the same scale
ylim_vals <- range(c(fit_adv$hazard, fit_notadv$hazard))

## Plot the hazard functions
plot(fit_adv$time, fit_adv$hazard, type = "l", lwd = 2, col = "red",
     xlab = "Time", ylab = "Hazard",
     main = "Smoothed Hazard by HIV Disease Stage",
     ylim = ylim_vals)

lines(fit_notadv$time, fit_notadv$hazard, lwd = 2, col = "blue")

legend("topright",
       legend = c("Advanced HIV", "Not advanced"),
       col    = c("red", "blue"), lwd    = 2)
```

Regression model for survival data

Accelerated failure time model (AFT)

$$\log T = \mu + \alpha X + \sigma\epsilon$$

where X is a covariate and ϵ is an error term

- Models the survival time directly
- α is the effect of X on **survival time**
- Works well for parametric models (e.g., log-normal)
- Harder to fit without parametric assumptions

Regression model for survival data

Proportional hazards model (PH)

$$\log h(t) = \log h_0(t) + \beta X$$

or

$$h(t) = h_0(t)\exp(\beta X)$$

where X is a covariate and $h_0(t)$ is a 'baseline hazard' function

- Models the hazard function, not survival time
- Very flexible – do not have to model $h_0(t)$ in many cases
- β is the effect of X on **event rate**

Proportional hazards model with binary covariate

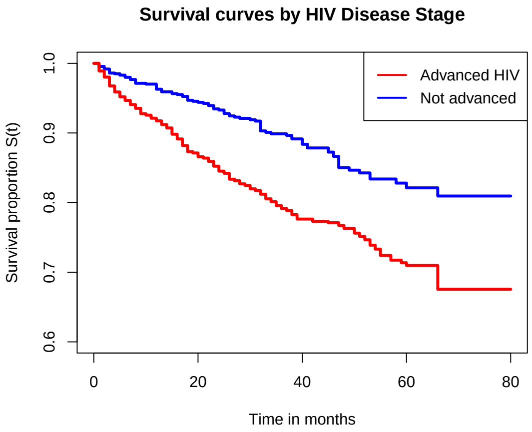
Model specification

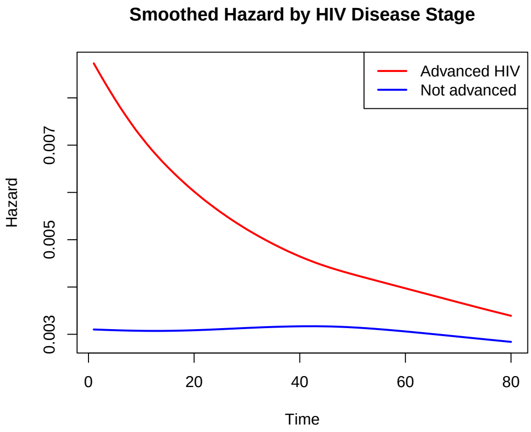
$$\begin{aligned} \log h(t) &= \log h_0(t) + \beta X \\ X &= \text{1 advanced HIV, 0 if not} \end{aligned}$$

Interpretation

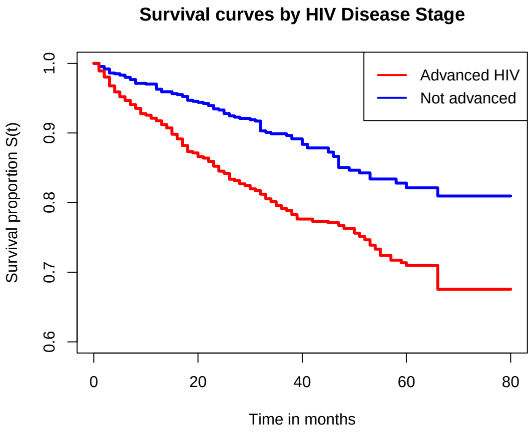
$$\begin{aligned} \log h_0(t) &= \text{log hazard function when } X = 0 \\ \log h_0(t) + \beta &= \text{log hazard function when } X = 1 \\ \beta &= \text{difference in log hazard functions,} \\ & \quad \textit{assumed constant over time} \end{aligned}$$

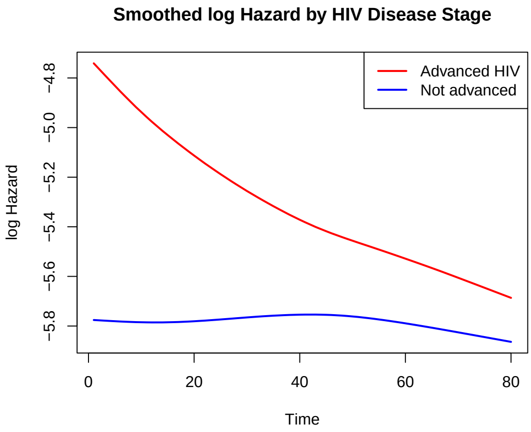
Hazard function vs survival function





Log hazard function vs survival function





Fitted model

Model specification

$$\begin{aligned} \log h(t) &= \log h_0(t) + \beta X \\ X &= \text{1 advanced HIV, 0 if not} \end{aligned}$$

```
> univ_cox <- coxph( Surv(survmonth, event) ~ adv_HIV,
                              data = htndat, ties = "efron")
> summary(univ_cox)

  n= 3038, number of events= 398
   (1960 observations deleted due to missingness)

        coef exp(coef) se(coef)    z Pr(>|z|)
adv_HIV 0.697     2.008    0.116 6.01 1.86e-09 ***
```

log hazard ratio $\hat{\beta} = .697$

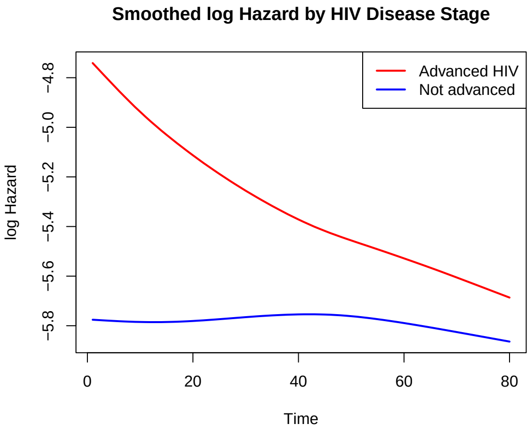
hazard ratio $\exp(.697) = 2.008$

Visualizing the model

$$\log h(t) = \log h_0(t) + \beta X$$
$$X = 1 \text{ advanced HIV, 0 if not}$$

log hazard ratio $\hat{\beta} = .697$

- How to interpret $\hat{\beta}$?
- Is this an appropriate model?



Examining proportional hazards assumption

Proportional hazards assumption can be assessed using **Schoenfeld residual plots**

These use a special type of residual to get an estimate of the HR as a function of time

$$\hat{\beta}(t) \quad = \quad \text{HR as a function of time}$$

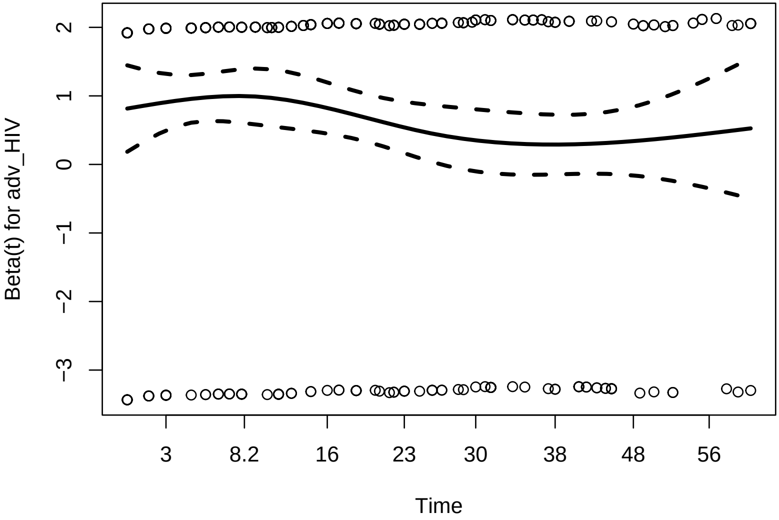
If PH assumption is met, $\hat{\beta}(t)$ will be a constant function of time (flat line)

R code for assessing PH assumption

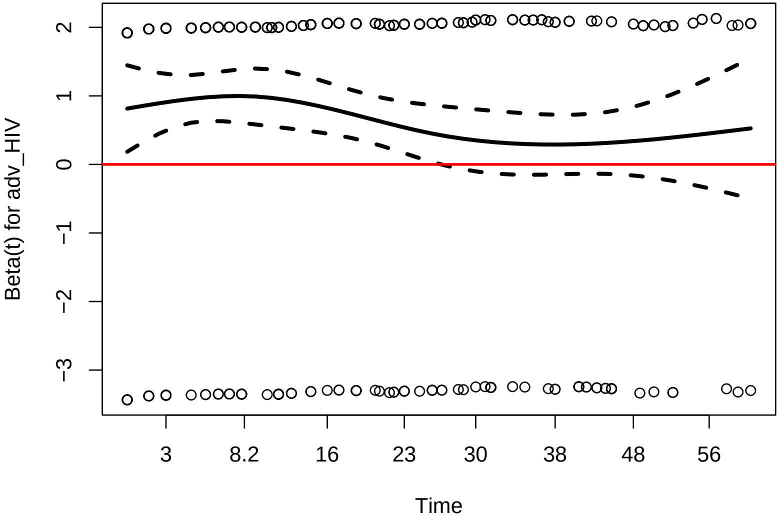
This code tests the PH assumption for advanced HIV effect using the fitted model `univ_cox`

```
## Using Schoenfeld residuals
> plot(cox.zph(univ_cox), lwd=3)

## Using a test of hypothesis (H0: proportional hazards holds)
> cox.zph(univ_cox)

        chisq df     p
adv_HIV  3.65  1 0.056
GLOBAL   3.65  1 0.056
```

Schoenfeld residual plot of $\hat{\beta}(t)$



Schoenfeld residual plot of $\hat{\beta}(t)$



Elaborate to allow hazard ratio to change over time

Add time-by-covariate interaction

$$\log h(t) \;=\; \log h_0(t) + \beta X + \delta(X \cdot t)$$

Interpretation

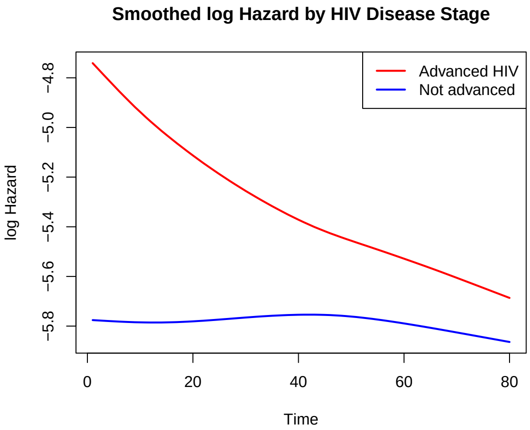
- $\beta + \delta t =$ log HR at time t
- $\beta =$ log HR near $t = 0$

Time-varying hazard ratio: fitted model

$$\log h(t) = \log h_0(t) + \beta X + \delta(X \cdot t)$$

$$\hat{\beta} = .973$$
$$\hat{\delta} = -.013$$

- How to interpret $\hat{\beta}$?
- How to calculate HR at different times?



R code for time-varying HR

```
> tv_cox <- coxph(surv_obj ~ adv_HIV + tt(adv_HIV),
                  data = htndat,
                  tt = function(x, t, ...) x * t)

> summary(tv_cox)

Call:
coxph(formula = surv_obj ~ adv_HIV + tt(adv_HIV), data = htndat,
    tt = function(x, t, ...) x * t)

  n= 3038, number of events= 398
   (1960 observations deleted due to missingness)

                  coef exp(coef)  se(coef)      z Pr(>|z|)
adv_HIV       0.972570  2.644733  0.188536  5.159 2.49e-07 ***
tt(adv_HIV) -0.013329  0.986759  0.006883 -1.937   0.0528 .
---
```

Regression with continuous covariate

Example: model effect of SBP on mortality hazard

$$\log h(t) = \log h_0(t) + \beta \, \mathrm{SBP}$$

Important questions

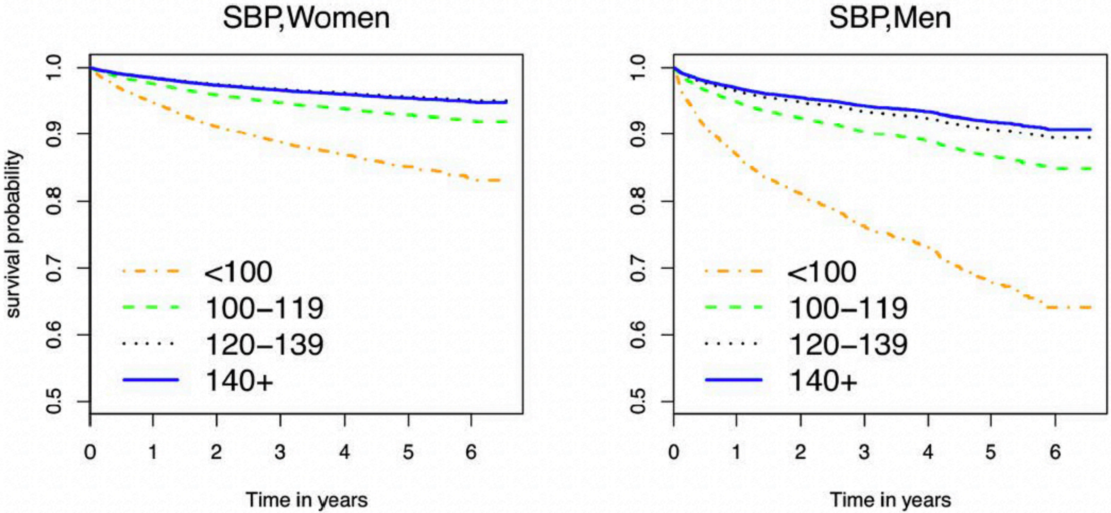
- How to interpret β
- Is the true effect of SBP linear?
- How to make the model more flexible

Summaries of SBP effect

Table 2 Unadjusted mortality rates per 100 person years by blood pressure groups for men and women with and without advanced HIV

Characteristic	Person years	Women events	Mortality rate (95% CI)	Person years	Men events	Mortality rate (95% CI)
Advanced HIV[a]						
Systolic blood pressure						
SBP <100 mmHg	3447	113	3.3 (2.7-3.9)	816	57	7.0 (5.4-9.1)
SBP 100-119 mmHg	18648	364	2.0 (1.8-2.2)	5966	219	3.7 (3.2-4.2)
SBP 120-139 mmHg	7954	116	1.5 (1.2-1.7)	4001	96	2.4 (2.0-2.9)
SBP ≥140 mmHg	1095	23	2.1 (1.4-3.2)	590	21	3.6 (2.3-5.5)

Summaries of SBP effect



Model building process

1. Fit model with linear effect

$$\log h(t) = \log h_0(t) + \beta \, \mathrm{SBP}$$

2. Examine residual plot to see if linearity is reasonable

3. Fit new model with nonlinear effect, using splines

$$\log h(t) = \log h_0(t) + f(\mathrm{SBP})$$

where f is a function that is estimated from the data

Model with linear effect: R code

```
## ----model1----------------------------------------------------
# Center SBP at 120 for interpretability and save data in dat
dat <- htndat %>%
  dplyr::mutate(sbp_c120 = SBP - 120)

# Fit Cox model with SBP only
model1 <- coxph(
  Surv(survmonth, event) ~ sbp_c120,
  data     = dat,
  ties     = "efron"
)
summary(model1)
# There is evidence of effect of SBP on time to death
```

Model with linear effect: Output

```
> summary(model1)
Call:
coxph(formula = Surv(survmonth, event) ~ sbp_c120, data = dat,
    ties = "efron")

  n= 4998, number of events= 749

             coef exp(coef) se(coef)      z Pr(>|z|)
sbp_c120 -0.02246   0.97779  0.00286 -7.853 4.05e-15 ***
---
         exp(coef) exp(-coef) lower .95 upper .95
sbp_c120    0.9778      1.023    0.9723    0.9833
```

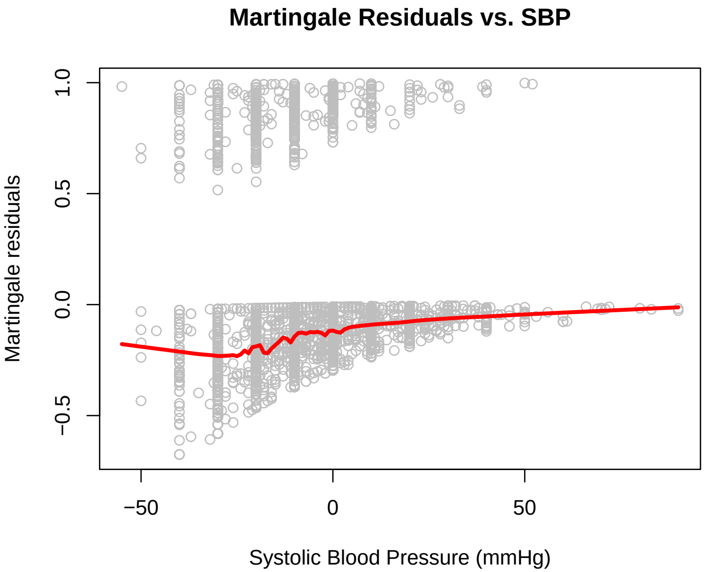
Parameter estimates

log hazard ratio	$\hat{\beta} = -.022$
hazard ratio	$\exp(-.022) = .98$

Martingale residual plots: R code

```
## ----Martingale------------------------------------------------
# Martingale residuals using model 1
htndat$mart_resid <- residuals(model1, type = "martingale")

## ----MartingalePlot--------------------------------------------
# Plot Martingale vs. linear predictor, SBP
plot(
  htndat$SBP-120,
  htndat$mart_resid,
  xlab = "Systolic Blood Pressure (mmHg)",
  ylab = "Martingale residuals",
  main = "Martingale Residuals vs. SBP",
  col="gray"
)

lines(
  lowess(htndat$SBP-120, htndat$mart_resid, f = 0.20),
  col = "red",
  lwd = 3
)
```

Martingale residual plot



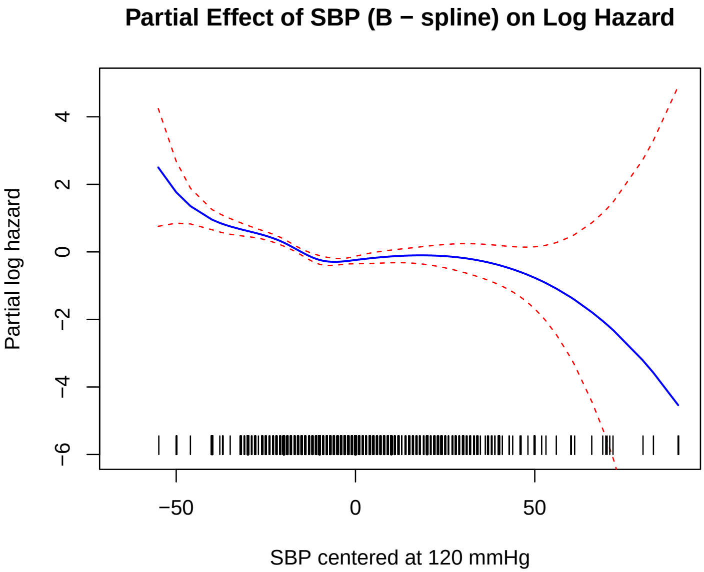
Regression spline for SBP: R code

```
## ----splineReg------------------------------------------------------
# choose interior knots at the .05, .35, .65, .95 quantiles of sbp_c120

###
knots <- quantile(dat$sbp_c120, probs = c(0.20, 0.50, 0.80))

# fit Cox model with a 3-df B-spline for sbp_c120 plus adv_HIV as a covariate
model_spline <- coxph(
  Surv(survmonth, event) ~ bs(sbp_c120, knots = knots, degree = 3),
  data     = dat,
  ties     = "efron"
)
summary(model_spline)
```

Plotting regression spline: R code

```
## ----------------------------------------------------------------------
# Visualize the partial effects of B-spline terms
termplot(
  model_spline,
  terms          = 1,           # first term is the bs(sbp_c150, ...) spline
  se             = TRUE,
  rug            = TRUE,
  col.term       = "blue",
  col.se         = "red",
  col.res        = "darkgray",
  main           = "Partial Effect of SBP (B - spline) on Log Hazard",
  xlab           = "SBP centered at 120 mmHg",
  ylab           = "Partial log hazard",
  ylim=c(-6,5)
)
```

Effect of SBP using cubic b-spline regression spline



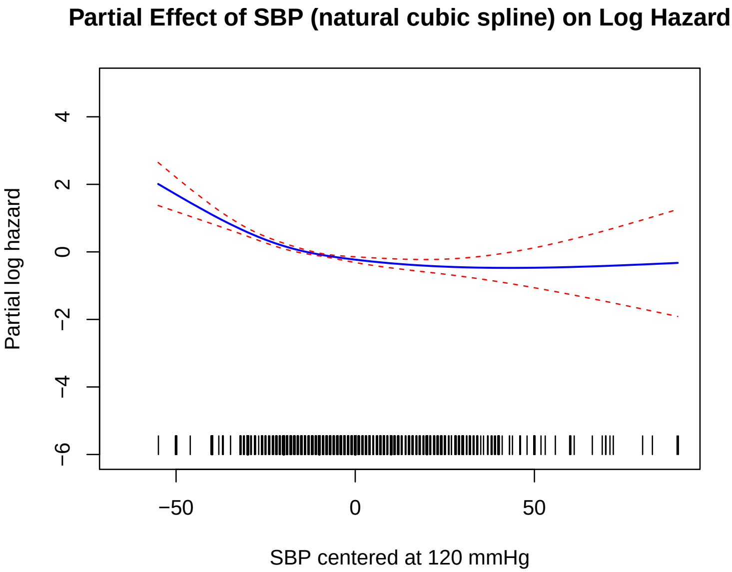
Another smoothing approach - natural spline

```
# fit Cox model with a 3-df natural spline for sbp_c120 plus adv_HIV as a covariate
model_spline <- coxph(
  Surv(survmonth, event) ~ ns(sbp_c120, df=3),
  data     = dat,
  ties     = "efron"
)
summary(model_spline)
```

Comparison

- bs() fits cubic smoothing spline - may have erratic tail behavior
- ns() fits natural cubic spline - constrains tail behavior

Effect of SBP using regression spline



Fitting models with other predictors

SBP and advanced HIV status

$$\log h(t) \quad = \quad \log h_0(t) + \beta_1 \text{SBP} + \beta_2 \text{Adv}$$

Add interaction to see if SBP effect differs by Adv HIV

$$\log h(t) \quad = \quad \log h_0(t) + \beta_1 \text{SBP} + \beta_2 \text{Adv} + \beta_3 (\text{SBP} \times \text{Adv})$$

Complex model to combine ideas we have discussed

- Add age and gender to the model
- Use spline term for SBP and age
- Allow hazard for advanced HIV to vary with time

$$\log h(t) \quad = \quad \log h_0(t) + f_1(\text{SBP}) + f_2(\text{age}) + \beta_1 \text{male.gender} + \beta_2 \text{Adv}$$

Fitting models with other predictors

SBP and advanced HIV status

$$\log h(t) = \log h_0(t) + \beta_1 \text{SBP} + \beta_2 \text{Adv}$$

```
> summary(model3)
Call:
coxph(formula = Surv(survmonth, event) ~ sbp_c120 + as.factor(adv_HIV),
    data = dat, ties = "efron")

  n= 3038, number of events= 398
   (1960 observations deleted due to missingness)

                         coef  exp(coef)  se(coef)      z  Pr(>|z|)
sbp_c120            -0.016011   0.984117  0.003723 -4.301 1.70e-05 ***
as.factor(adv_HIV)1  0.686845   1.987435  0.116028  5.920 3.23e-09 ***
---

                    exp(coef) exp(-coef) lower .95 upper .95
sbp_c120               0.9841     1.0161     0.977    0.9913
as.factor(adv_HIV)1    1.9874     0.5032     1.583    2.4949
```

Fitting models with other predictors

Add interaction to see if SBP effect differs by Adv HIV

$$\log h(t) \;=\; \log h_0(t) + \beta_1 \mathrm{SBP} + \beta_2 \mathrm{Adv} + \beta_3(\mathrm{SBP} \times \mathrm{Adv})$$

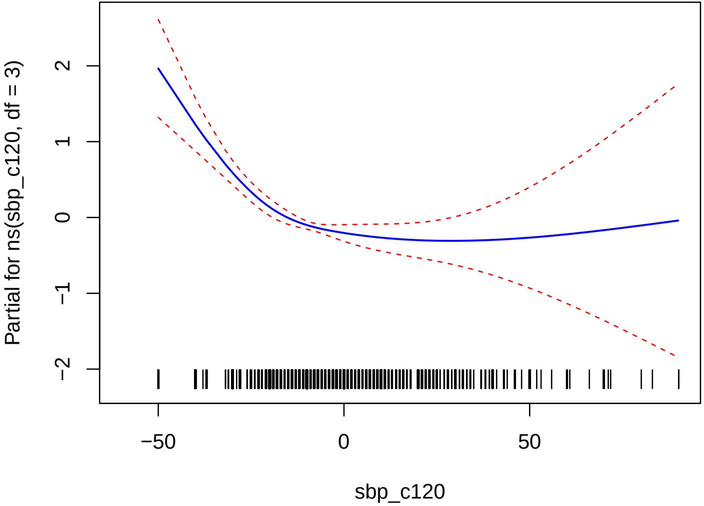
```
Call:
coxph(formula = Surv(survmonth, event) ~ sbp_c120 * as.factor(adv_HIV),
    data = dat, ties = "efron")

  n= 3038, number of events= 398
   (1960 observations deleted due to missingness)

                                 coef exp(coef)  se(coef)      z Pr(>|z|)
sbp_c120                    -0.010847  0.989212  0.007533 -1.440    0.150
as.factor(adv_HIV)1          0.609250  1.839052  0.151291  4.027 5.65e-05 ***
sbp_c120:as.factor(adv_HIV)1 -0.006764  0.993259  0.008665 -0.781    0.435
```

Complex model to combine ideas we have discussed

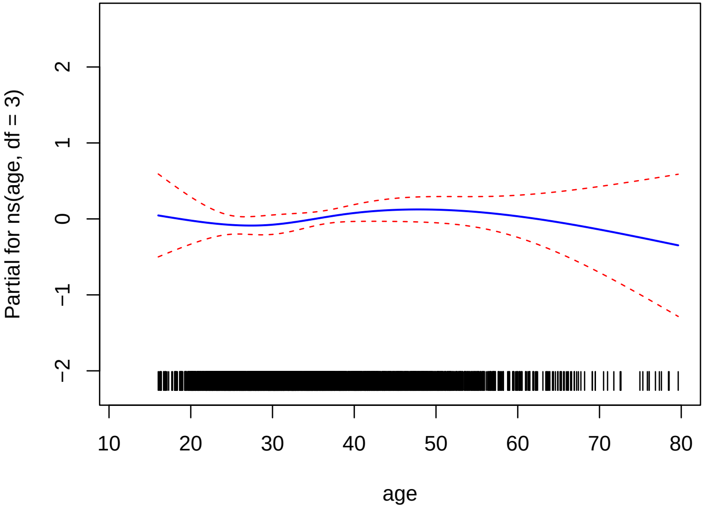
$$\log h(t) = \log h_0(t) + f_1(\text{SBP}) + f_2(\text{age}) + \beta_1 \text{male.gender} + \beta_2 \text{Adv}$$

```
coxph(formula = Surv(survmonth, event) ~ ns(sbp_c120, df = 3) +
    ns(age, df = 3) + as.factor(male.gender) + as.factor(adv_HIV),
    data = dat, ties = "efron")

  n= 3038, number of events= 398
   (1960 observations deleted due to missingness)

                             coef exp(coef)  se(coef)      z Pr(>|z|)
ns(sbp_c120, df = 3)1   -1.921020  0.146457  0.310251 -6.192 5.95e-10 ***
ns(sbp_c120, df = 3)2   -4.609319  0.009959  0.988404 -4.663 3.11e-06 ***
ns(sbp_c120, df = 3)3   -1.296022  0.273618  0.931644 -1.391    0.164
ns(age, df = 3)1         0.368627  1.445748  0.251386  1.466    0.143
ns(age, df = 3)2        -0.270376  0.763092  0.657682 -0.411    0.681
ns(age, df = 3)3        -0.293124  0.745930  0.488120 -0.601    0.548

as.factor(male.gender)1  0.630648  1.878828  0.109065  5.782 7.37e-09 ***
as.factor(adv_HIV)1      0.569175  1.766808  0.118548  4.801 1.58e-06 ***
```

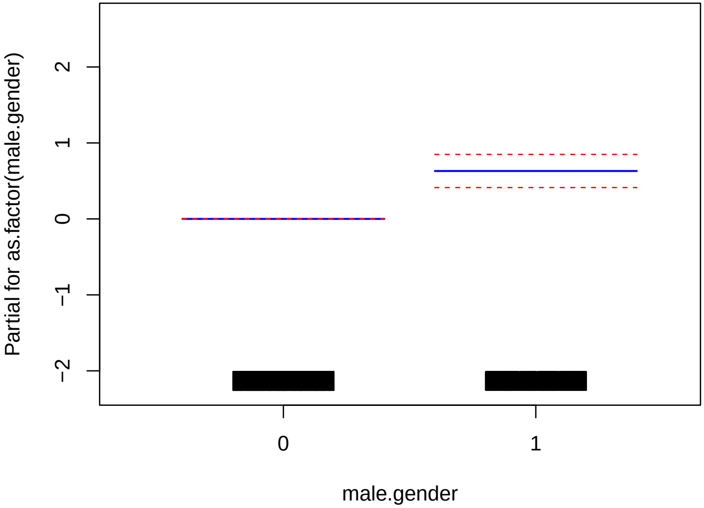
SBP effect



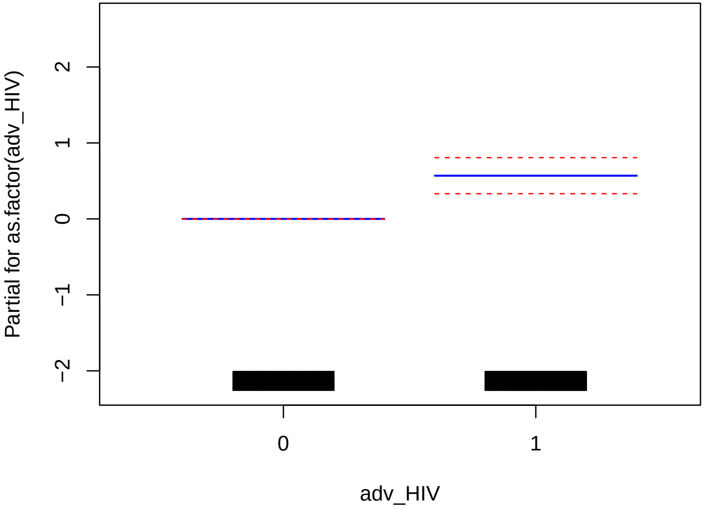
Age effect



Male gender effect



Advanced HIV effect



Exercises for day 1

1. Create stratified survival curves
   1. Stratify by gender
   2. Stratify by marital status
   3. Assess whether hazards look proportional
2. Fit a regression model to assess the effect of hemoglobin on survival
   1. Use a linear trend
   2. Use a natural regression spline
   3. Include gender and advanced HIV status as covariates
   4. Add SBP as a covariate

Session 4: Prediction

Please install R package pec

Prediction from proportional hazards regression

- Predictions in terms of survival probabilities
- Generating predictions for those in the dataset
- Summaries of model calibration to compare models
  - Calibration plot
  - Brier score
- Out of sample prediction

Predictions in terms of survival probabilities

Simple PH regression with single predictor

$$\log h(t \mid X) \;=\; \log h_0(t) + \beta X$$

- Usual regression prediction: predict the outcome
- Survival analysis: censoring makes this difficult
- Instead: predict individual-specific survival function

Predictions in terms of survival probabilities

Simple PH regression with single predictor

$$\log h(t \mid X) = \log h_0(t) + \beta X$$

Prediction from PH model

For a specific value $X = x^*$, can generate survival curve

$$\hat{S}(t \mid x^*) = \hat{P}(T > t \mid X = x^*)$$

Example: If X represents SBP, can generate entire survival curve for any specific value of SBP

$$\begin{aligned} \hat{S}(t \mid 120) &= \hat{P}(T > t \mid X = 120) \\ \hat{S}(t \mid 80) &= \hat{P}(T > t \mid X = 80) \end{aligned}$$

Generating and comparing $\hat{S}(t \mid X)$ between models

We will work with 4 models of survival

$$\log h(t \mid X) = \log h_0(t) + \beta\,(\text{SBP})$$

$$\log h(t \mid X) = \log h_0(t) + f_1(\text{SBP})$$

$$\begin{aligned}\log h(t \mid X) &= \log h_0(t) + f_1(\text{SBP}) + f_2(\text{age}) \\ &\quad + \beta_1 \text{male.gender} + \beta_2 \text{advHIV}\end{aligned}$$

$$\begin{aligned}\log h(t \mid X) &= \log h_0(t) + f_1(\text{SBP}) + f_2(\text{DBP}) + f_3(\text{age}) \\ &\quad + \beta_1 \text{male.gender} + \beta_2 \text{advHIV} + \beta_3 \text{arvNaive}\end{aligned}$$

Compare 4-year survival between models 1 and 2

$$\log h(t \mid X) = \log h_0(t) + \beta\,(\mathrm{SBP})$$

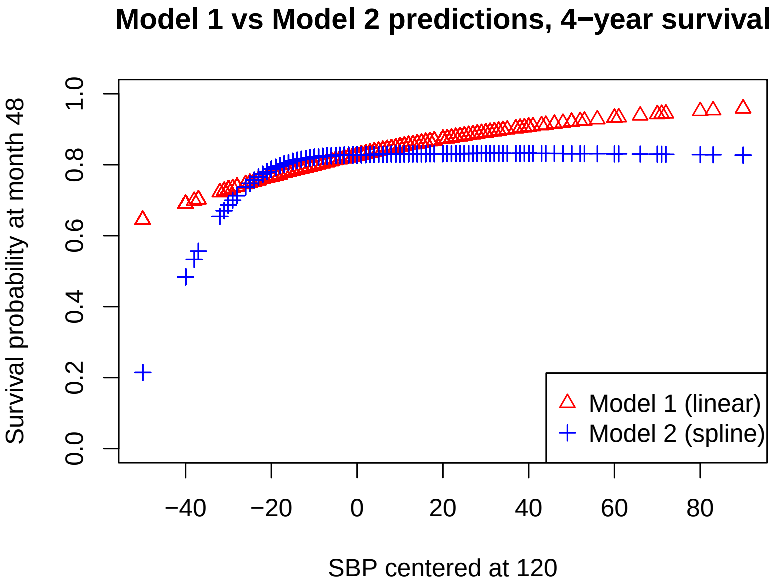
$$\log h(t \mid X) = \log h_0(t) + f_1(\mathrm{SBP})$$

- Use R code on next slide to generate predicted probabilities for each individual in the dataset
- Plot predictions as a function of SBP
- Compare predictions between models

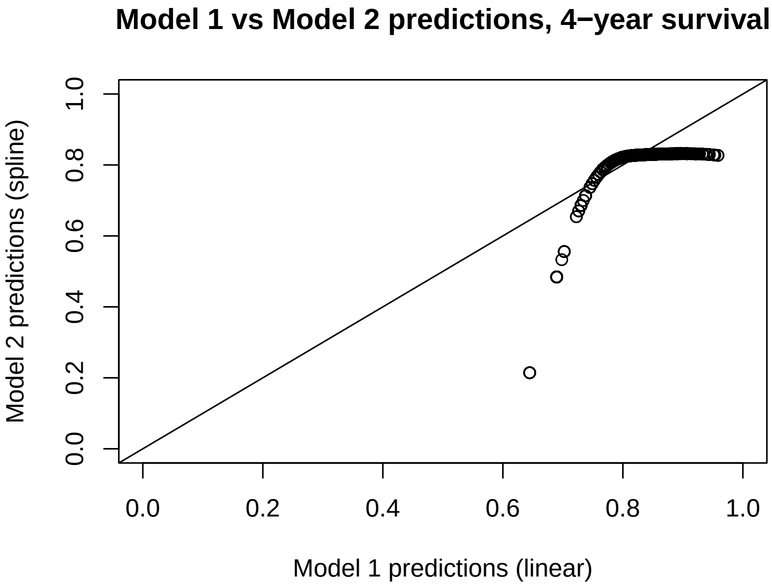
R code for generating predicted probabilities

```
model1 <- coxph(
  Surv(survmonth, event) ~ sbp_c120,
  data    = dat,
  ties    = "efron",x=TRUE
)

# Predict survival curves for model 1
model1_pred <- survfit(model1, newdata = dat)

# Use summary to get survival at 48 months
model1_survprob <- summary(model1_pred, times = 48)

# The result includes a flat list:
# Use the 'records' field to reshape into matrix
model1_survmatrix <- matrix(model1_survprob$surv, ncol = 1, byrow = TRUE)

# Add predictions to data frame
dat$s48_model1 <- model1_survmatrix[, 1]
```

Compare predicted probabilities by SBP



Compare predicted probabilities between models 1 and 2

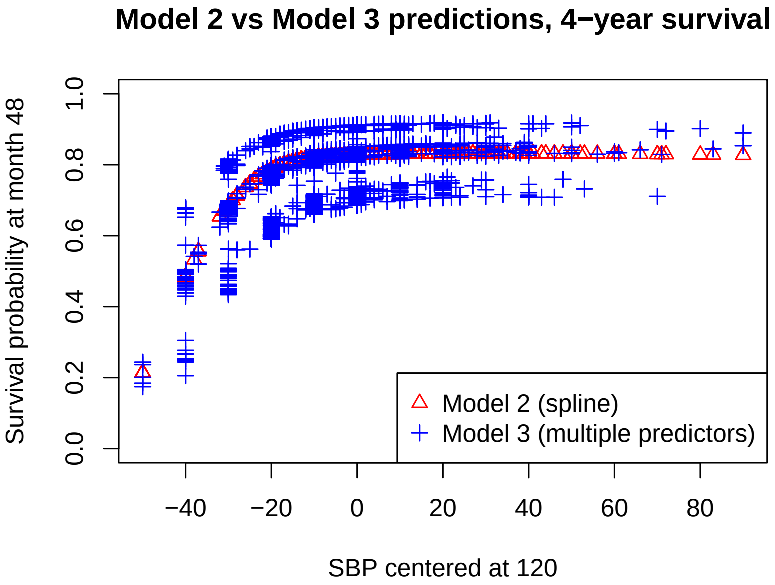


Compare 4-year survival between models 2 and 3

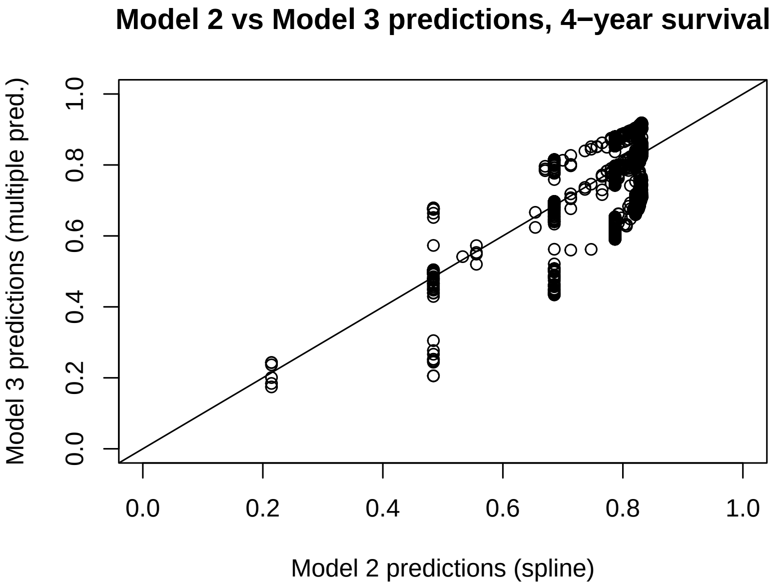
$$\log h(t \mid X) = \log h_0(t) + f_1(\text{SBP})$$

$$\begin{aligned} \log h(t \mid X) &= \log h_0(t) + f_1(\text{SBP}) + f_2(\text{age}) \\ &\quad + \beta_1 \text{male.gender} + \beta_2 \text{advHIV} \end{aligned}$$

Compare predicted probabilities by SBP



Compare predicted probabilities between models 2 and 3



Assessing and comparing model fit using calibration plots

Calibration plots compare the model-based predictions to observed data

Challenge: Because of censoring, cannot compare predicted survival time to observed survival time.

Solution: Group the model-based survival predictions into quantiles, and compare predicted to observed survival probability **within each quantile**

Illustration

For model 3, the deciles of predicted 4-year survival are found as follows.

```
> quantile(dat$s48_model3, probs=seq(0,1,.1))

       0%       10%       20%       30%       40%       50% 
0.1743208 0.6736697 0.7191268 0.7776377 0.7978039 0.8208215 

      60%       70%       80%       90%      100% 
0.8349416 0.8499603 0.8795266 0.9029255 0.9187028 
```

There are about 300 observations within each decile

- Step 1: Calculate average **predicted** value of $\widehat{S}(48 \mid X)$ within each decile
- Step 2: Use K-M curve to estimate **observed** value of $\widehat{S}(48)$ within each decile
- Step 3: Plot observed vs. predicted within each decile.

A model that fits well should have good agreement between **observed** and **predicted** survival probabilities

Use calibration function in R to make plots

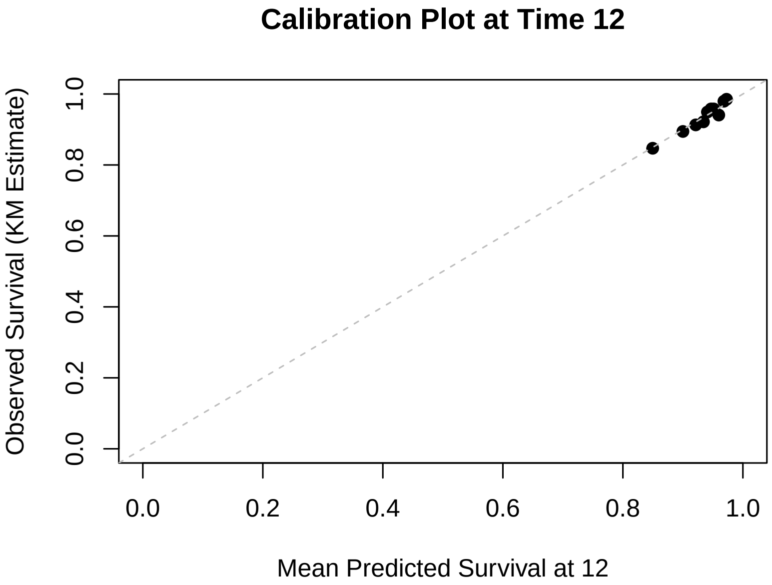
Calibration plots for month 48 predictions

```
calibrate_survival_model(t_star=48,
model=model3,
data_calib=dat,
time_var="survmonth",
status_var="event",
pct=.10)

calibrate_survival_model(48, model4, dat, "survmonth", "event", pct=.10)
```

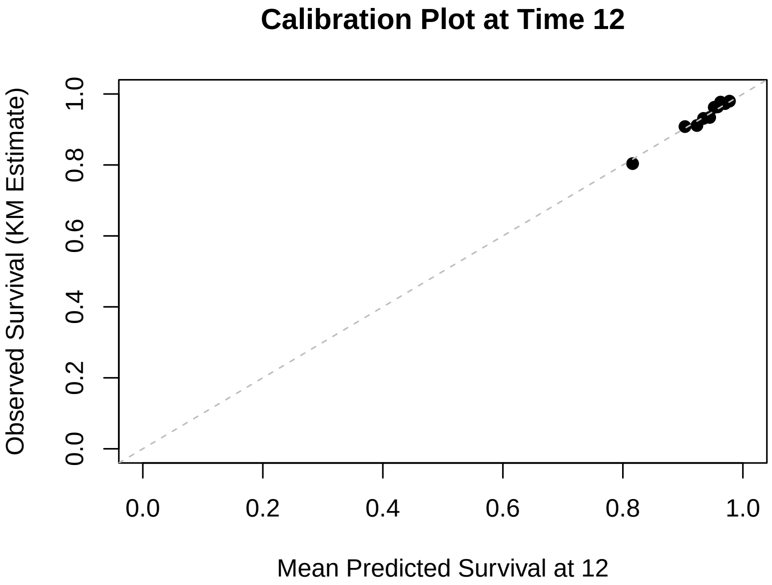
Calibration plots for month 72 predictions

```
calibrate_survival_model(72, model3, dat, "survmonth", "event", pct=.10)
calibrate_survival_model(72, model4, dat, "survmonth", "event", pct=.10)
```

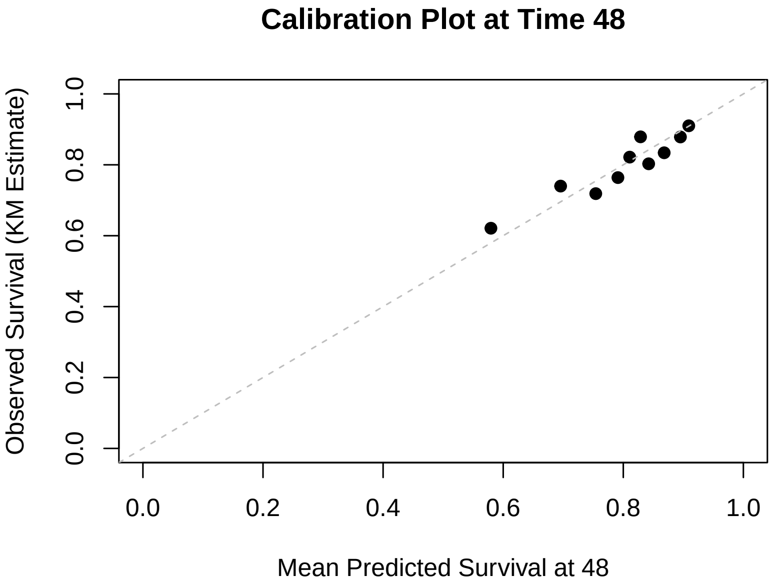
Calibration: Model 3, Month 12



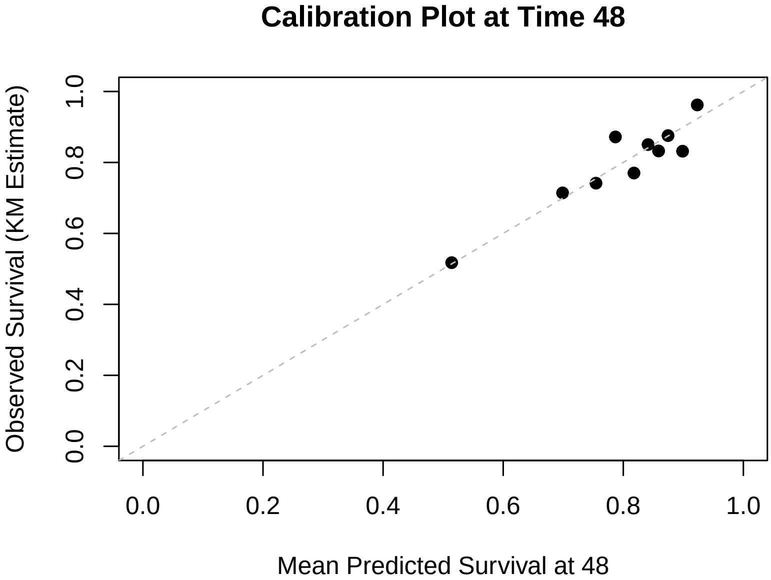
Calibration: Model 4, Month 12



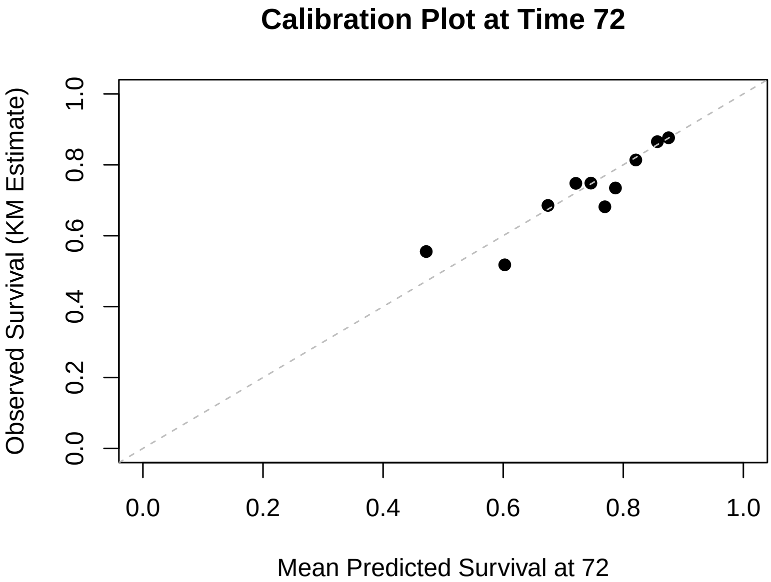
Calibration: Model 3, Month 48



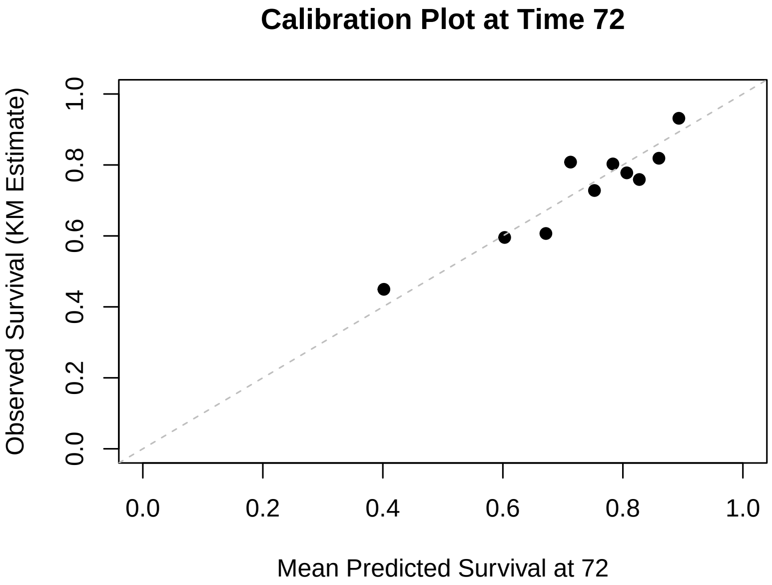
Calibration: Model 4, Month 48



Calibration: Model 3, Month 72



Calibration: Model 4, Month 72



Quantifying model fit: Brier score

The *Brier score* uses individual outcomes to quantify model fit. But it has limitations

- It only uses those with observed events
- It handles censoring by reweighting

The formula for the Brier score is constructed like *mean square error*

$$B(t) = (1/n) \sum_i w_i(t) \left\{ \mathbb{I}(T_i \leq t, \Delta_i = 1) - \hat{S}(t \mid X_i) \right\}^2$$

$$\begin{aligned} w_i(t) &= \text{censoring weight} \\ \mathbb{I}(T_i \leq t, \Delta_i = 1) &= 1 \text{ if person } i \text{ had event before } t, \\ & \quad\; 0 \text{ if not} \\ \hat{S}(t \mid X_i) &= \text{predicted survival probability at } t \end{aligned}$$

Quantifying model fit: Brier score

Time-specific Brier score assesses fit at specific times

Integrated Brier score averages over times, and is a measure of overall model fit

The Brier scores can be compared across models. Lower scores are better (lower mean squared error)

These scores can also be used for assessing **external validation**. More on that tomorrow.

R code for Brier score

```
brier <- pec(object = list("model2"=model2, "model3"=model3, "model4"=model4),
             formula = Surv(survmonth, event) ~ 1,
             data = dat,
             times = c(12,24,36,48,60,72),
             cens.model = "marginal",
             split.method = "loocv")

# time-specific brier scores
summary(brier, times=c(12,24,36,48,60,72))

# integrated brier score
crps(brier)
```

Comparing Brier scores across the models

```
> summary(brier, times=c(12,24,36,48,60,72))

 AppErr
  time n.risk Reference model2 model3 model4
1   12   1964     0.061  0.060  0.059  0.058
2   24   1377     0.106  0.105  0.102  0.099
3   36    920     0.139  0.136  0.132  0.126
4   48    595     0.162  0.158  0.155  0.147
5   60    273     0.186  0.186  0.182  0.173
6   72     73     0.199  0.200  0.196  0.191

> # integrated brier score
> crps(brier)
Integrated Brier score (crps):

          IBS[0;time=80)
Reference          0.133
model2             0.131
model3             0.129
model4             0.124
```

Questions on calibration and model assessment

- Add one or more variables to model 4, call it model 5
- Use calibration to see if the model fits better
- Check the overall fit using Brier scores

Case Study 2: Missing Data, Random Forest

Missing predictors in regression

Many ways to handle missing predictors

- Drop all cases with missing predictors
  - Easy
  - Throwing away important data
- Impute new values (multiple imputation)
  - Generates a filled-in dataset
  - Requires a model for imputation
  - Usually have to do multiple imputation to account for uncertainty of the imputed values

Missing predictors in regression

- Inverse probability weighting
  - Use cases with complete data
  - Weight the cases in inverse proportion to the probability of being 'complete'
  - Requires a model for probability of being complete
  - Need care when validating to new data
- Missing indicators
  - Add an indicator for whether a covariate is missing
  - Allows use of all cases
  - Can be ok for prediction: Having missing information can be predictive
  - Not recommended if interest is in effect of covariates

Missing predictors in regression

Each of the methods above can be valid approaches for handling missing predictors

Focus for today is use of the **missing indictor** method, for illustration

Should not be viewed as an endorsement of this method for all circumstances!

Some notation for missing data

For a covariate X,

$$
\begin{aligned}
R &= 1 \text{ if observed, } 0 \text{ if missing} \\
M &= 1 \text{ if missing, } 0 \text{ if observed} \\
&= 1 - R
\end{aligned}
$$

Missing data - missing indicator method

For discrete covariates, we add another category for 'missing'

Example: If X has two levels, 0 and 1, but some values are missing, we create a new version:

$$X^* = \begin{cases} -1 & \text{if missing} \\ 0 & \text{HIV-negative} \\ 1 & \text{HIV-positive} \end{cases}$$

For regression models, we just include the 3-factor variable instead of the two-factor variable.

Missing data - missing indicator method

For continuous covariates, we parameterize the model using the missing data indicators.

Example: model with BMI and gender, where BMI is missing for some people

$$\log h(t) = \log h_0(t) + \beta_1 \text{Gender} + \beta_2(\text{BMI} \times R) + \beta_3 M$$

For prediction, the coefficient β_3 is the effect of BMI being missing

- If $R = 1$

$$\log h(t) = \log h_0(t) + \beta_1 \text{Gender} + \beta_2 \text{BMI}$$

- If $R = 0$, this means $M = 1$

$$\log h(t) = \log h_0(t) + \beta_1 \text{Gender} + \beta_3$$

Adding a category for a categorical variable

Example: Model with SBP, DBP, and advanced HIV status (missing)

Note: The reference category is 0 (not advanced HIV)

```
coxph(formula = Surv(survmonth, event) ~ ns(SBP, df = 3) + ns(DBP,
    df = 3) + as.factor(adv_HIV), data = train, ties = "efron",
    x = TRUE)

  n= 1683, number of events= 233
   (867 observations deleted due to missingness)

                         coef exp(coef)  se(coef)      z Pr(>|z|)
ns(SBP, df = 3)1    -1.239957  0.289397  0.411568 -3.013 0.002589 **
ns(SBP, df = 3)2    -4.248157  0.014291  1.280427 -3.318 0.000907 ***
ns(SBP, df = 3)3    -1.066975  0.344048  1.285571 -0.830 0.406560
ns(DBP, df = 3)1    -1.233406  0.291299  0.402542 -3.064 0.002184 **
ns(DBP, df = 3)2    -4.945258  0.007117  1.030517 -4.799  1.6e-06 ***
ns(DBP, df = 3)3    -2.661150  0.069868  1.485188 -1.792 0.073166 .
as.factor(adv_HIV)1  0.581275  1.788317  0.151356  3.840 0.000123 ***
```

Creating the new categorical variable

```
# create new category for adv_HIV to reflect missingness
# setting to -1 if missing

> dat$adv_HIV_new = dat$adv_HIV
> dat$adv_HIV_new = ifelse(is.na(dat$adv_HIV), -1, dat$adv_HIV)

> table(dat$adv_HIV_new)

  -1    0    1
1418 1055 1778
```

Adding a category for a categorical variable

Example: Use missing indicator to create 3rd category for advanced HIV

Note: The reference category is -1 (Advanced HIV missing)

```
coxph(formula = Surv(survmonth, event) ~ ns(SBP, df = 3) + ns(DBP,
    df = 3) + as.factor(adv_HIV_new), data = train, ties = "efron",
    x = TRUE)

  n= 2550, number of events= 401

                           coef exp(coef) se(coef)      z Pr(>|z|)
ns(SBP, df = 3)1       -1.16263   0.31266  0.33110 -3.511 0.000446 ***
ns(SBP, df = 3)2       -3.61930   0.02680  1.17352 -3.084 0.002041 **
ns(SBP, df = 3)3       -1.46454   0.23118  1.34290 -1.091 0.275456
ns(DBP, df = 3)1       -1.64559   0.19290  0.30055 -5.475 4.37e-08 ***
ns(DBP, df = 3)2       -4.42399   0.01199  0.74373 -5.948 2.71e-09 ***
ns(DBP, df = 3)3       -2.00089   0.13521  1.07236 -1.866 0.062059 .
as.factor(adv_HIV_new)0 -0.78773   0.45488  0.15190 -5.186 2.15e-07 ***
as.factor(adv_HIV_new)1 -0.20616   0.81370  0.10854 -1.899 0.057505 .
```

Adding a missing data indicator for continuous variable

- Create missing data indicator

$$\begin{aligned} M &= 1 \text{ if } X \text{ missing} \\ &= 0 \text{ if } X \text{ observed} \end{aligned}$$

- Set missing values to 0 because $RX = 0$ when $R = 0$

Example using `hgb`

```
dat$hgb_miss = is.na(dat$hgb_centered)
dat$hgb_recode = ifelse( is.na(dat$hgb_centered), 0, dat$hgb_centered)
```

Using missing indicator for continuous variable

```
all:
coxph(formula = Surv(survmonth, event) ~ SBP + DBP + log_creat_centered,
    data = train, ties = "efron", x = TRUE)

  n= 1932, number of events= 279
   (618 observations deleted due to missingness)

                        coef exp(coef)  se(coef)      z Pr(>|z|)
SBP                -0.012921  0.987163  0.004806 -2.689  0.00717 **
DBP                -0.038429  0.962300  0.006981 -5.505  3.7e-08 ***
log_creat_centered  0.294038  1.341835  0.191327  1.537  0.12433
---
```

Using missing indicator for continuous variable

```
Call:
coxph(formula = Surv(survmonth, event) ~ SBP + DBP + log_creat_recode +
    log_creat_miss, data = train, ties = "efron", x = TRUE)

  n= 2550, number of events= 401

                        coef exp(coef)  se(coef)      z Pr(>|z|)
SBP                -0.012170  0.987904  0.003973 -3.063  0.00219 **
DBP                -0.042956  0.957954  0.005856 -7.335 2.22e-13 ***
log_creat_recode    0.304529  1.355986  0.191038  1.594  0.11092
log_creat_missTRUE  0.276188  1.318096  0.109515  2.522  0.01167 *
```

What is machine learning?

ML has many definitions. It depends on who's defining it!

For the purposes of this workshop:

Machine learning refers to the process of developing, validating and using an algorithm or **rule** *R that receives information $\boldsymbol{X}$ as input and generates a prediction $\hat{Y}$ as output. The process typically is optimized for a specific use of the prediction, such as for decision making or medical diagnosis.*

$$\boxed{X_1, X_2, \ldots, X_k} \quad \longrightarrow \quad R(X_1, X_2, \ldots, X_k) \quad \longrightarrow \quad \hat{Y}$$

In other words, it's the process of **learning** a rule to translate the inputs $\boldsymbol{X}$ into an output $\hat{Y}$.

Broad classes of machine learning

- Supervised learning
- Unsupervised learning
- Semi-supervised learning

Machine learning: Goals and uses

- Prediction of a continuous value (e.g., blood pressure)
- Prediction of a binary endpoint (e.g., viral failure)
- Prediction of an event time (e.g., time to death)
- Classification and diagnosis (e.g., high risk versus low risk for disease)
- Forming clusters of units or individuals whose features are similar

Machine learning

$$\boxed{X_1, X_2, \ldots, X_k} \longrightarrow R(X_1, X_2, \ldots, X_k) \longrightarrow \hat{Y}$$

- The X inputs can be simple (age, sex, etc.) or complex (clinical notes, images).
- When they are complex, they must be rendered in a way that can be handled by the algorithm
- The algorithm or rule $R(X_1, X_2, \ldots, X_k)$ can be
  - A fitted statistical model that generates predictions
  - A complex algorithm that processes the inputs
  - A series of if-then statements
  - etc.

Training and testing steps to validation

Building and validating a prediction model typically requires two steps

1. Training step: Fit the model to a sample of data that are representative of a target population
2. Test step: Use new data to see how well the model performs

We want a model that performs well on the *test* data.

Bias and Variance

Two key contributors to model performance are **bias** and **variance**

- **Bias:** We want predictions that are close to the true value of the outcome we are trying to predict
- **Variance:** We want predictions that do not vary too much around the true value.

A model that performs well on new data will balance these two

Split sample (validation set) approach

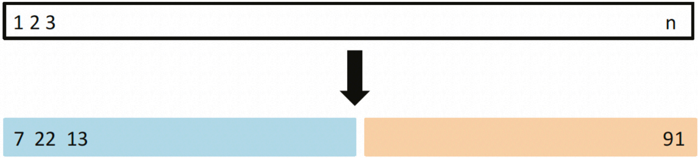
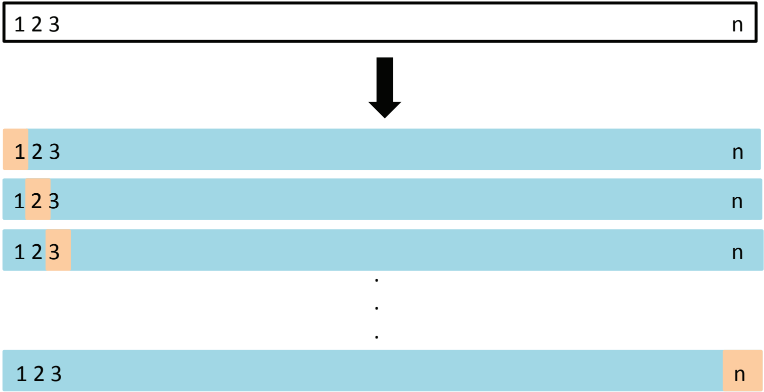


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

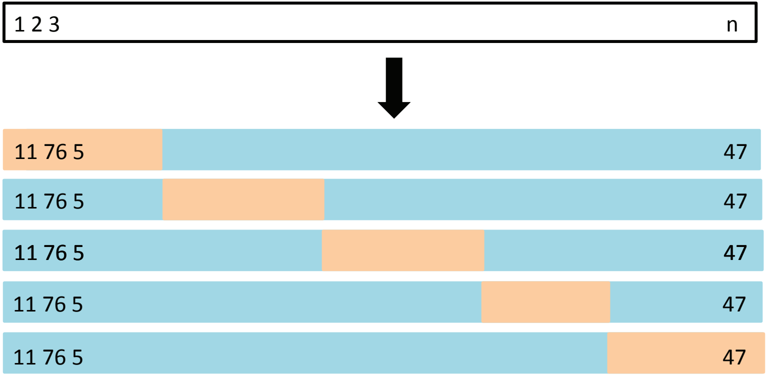
Calculate MSE using the validation dataset

Leave-one-out cross validation



- Fit separate model to each blue dataset
- Calculate $[Y - \widehat{f}(X)]^2$ on the single left-out observation for each dataset
- Average these to get the MSE

K-fold cross validation



- Fit separate model to each blue dataset
- Calculate fold-specific MSE on the **sample** of left-out observations
- Average these fold-specific MSE to get the overall MSE

Comparing the methods

Split sample

- Easy to implement
- Usually requires a large dataset
- Relies on a single split – results may vary depending on the specific way the data were split

Leave-one-out

- Also easy to implement
- Better for smaller datasets
- Tends to produce accurate estimates of MSE
- Requires refitting the model n times - can be very computationally intensive
  - This is not required if using linear regression

K-fold

- Slightly more complicated to implement - but R can do it easily
- Good compromise between split sample and leave-one-out

Code for test-train split

```
# use 3/5 of the data for the training sample (60%)
set.seed(20250716)
train_id <- sample(seq_len(nrow(dat)), floor( (3/5)*nrow(dat) ) )

# Split data into train and test sets
train <- dat[train_id, ]
test  <- dat[-train_id, ]

> nrow(dat)
[1] 4251
> nrow(test)
[1] 1701
> nrow(train)
[1] 2550
```

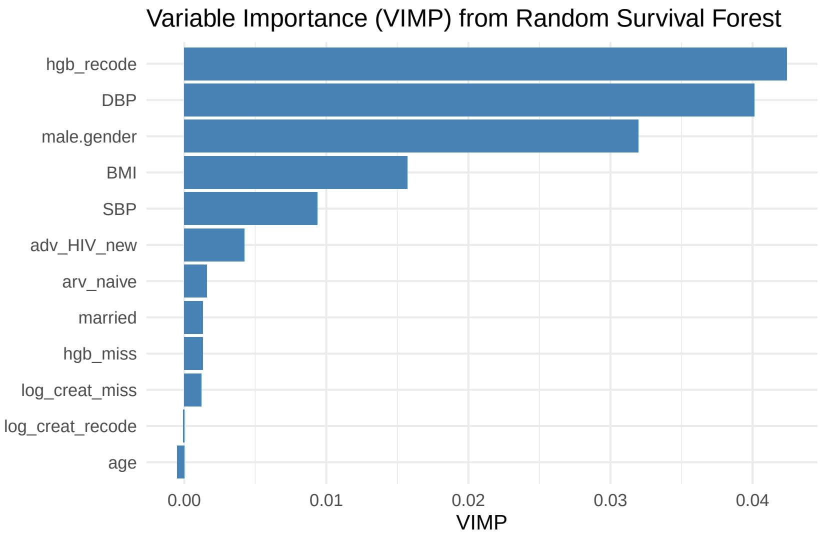
Regression model with multiple variables

```
model6 <- coxph(
  Surv(survmonth, event) ~
    as.factor(male.gender)
  + as.factor(adv_HIV_new)
  + as.factor(arv_naive)
  + as.factor(married)
  + ns(SBP, df=3)
  + ns(DBP, df=3)
  + ns(age, df=3)
  + ns(BMI, df=3)
  + log_creat_recode + log_creat_miss
  + hgb_recode + hgb_miss,
  data    = train,
  ties    = "efron", x=TRUE
)
```

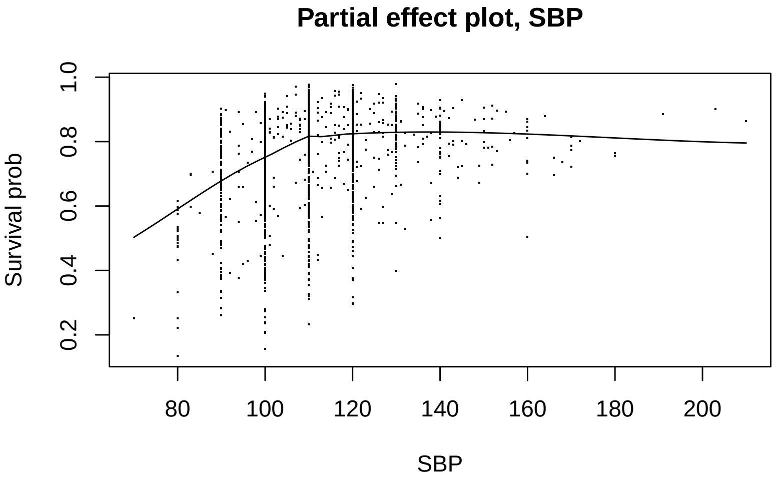
Random forest (ML) with multiple variables

```
# Fit the random forest model on the training data set
rsf_fit <- randomForestSRC::rfsrc(Surv(survmonth, event) ~
                                    as.factor(male.gender)
                                  + as.factor(adv_HIV_new)
                                  + as.factor(arv_naive)
                                  + as.factor(married)
                                  + SBP
                                  + DBP
                                  + age
                                  + BMI
                                  + log_creat_recode + log_creat_miss
                                  + hgb_recode + hgb_miss,
                                  data = train,
                                  nsplit = 10,
                                  ntree = 500,
                                  importance = "permute",
                                  na.action = "na.impute")
```

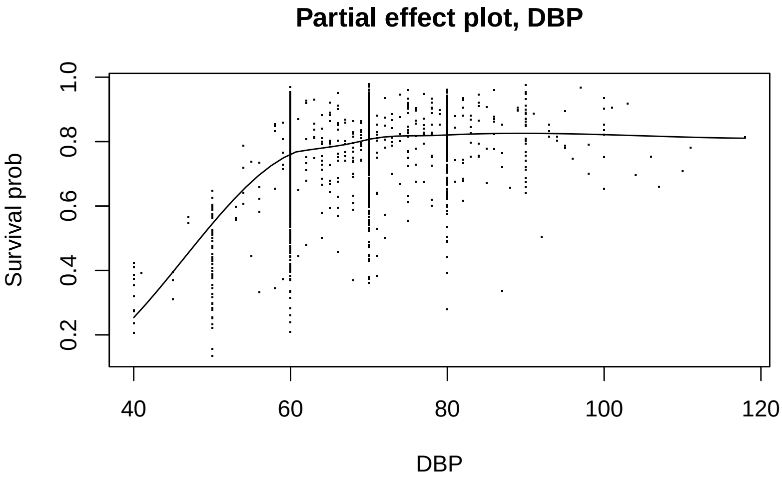
Variable Importance Plot



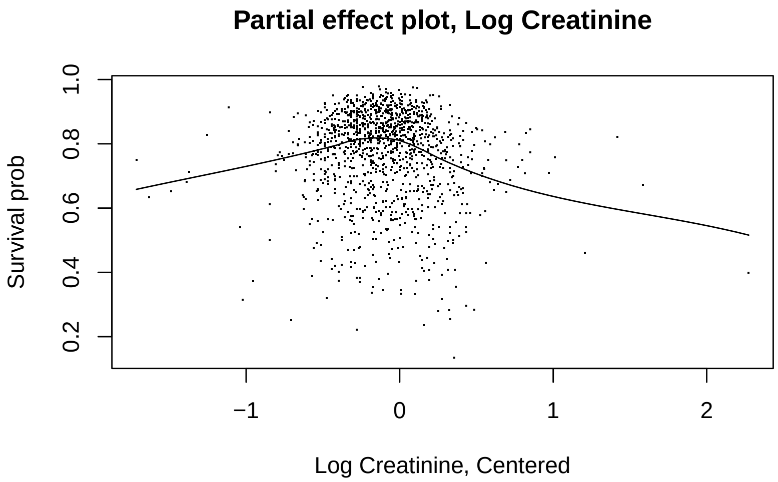
Partial effect plots



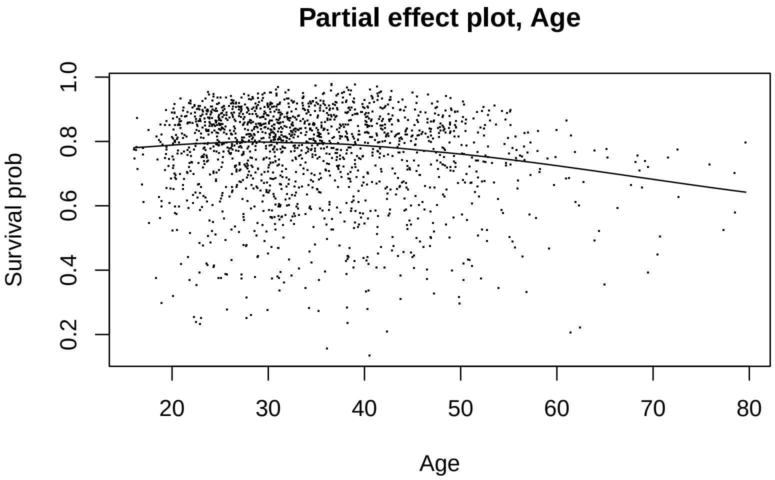
Partial effect plots



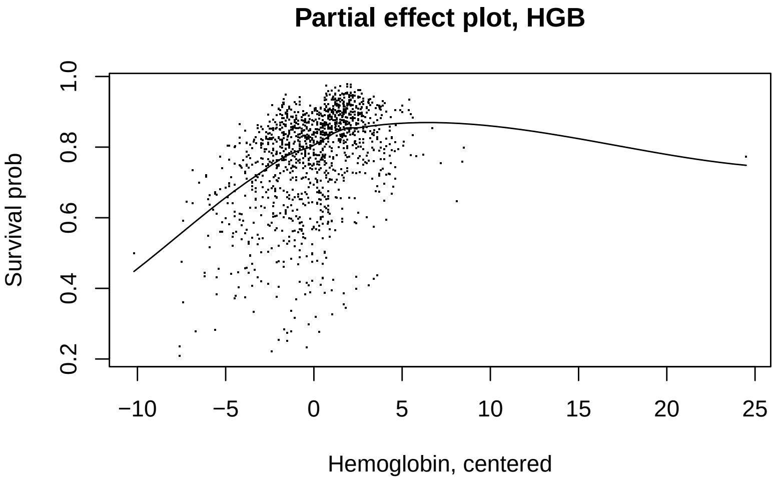
Partial effect plots



Partial effect plots



Partial effect plots



Brier scores: Test data

```
 AppErr
  time n.risk Reference    RF Model6
1   12   1073     0.068 0.064  0.065
2   24    762     0.115 0.108  0.106
3   36    490     0.147 0.132  0.129
4   48    321     0.166 0.151  0.149
5   60    144     0.188 0.176  0.172
6   72     52     0.200 0.189  0.189

>
> # integrated brier score
> crps(brier)

Integrated Brier score (crps):

          IBS[0;time=80)
Reference          0.138
RF                 0.129
Model6             0.126
>
```

Brier scores: Training data

```
  AppErr
  time n.risk Reference    RF Model6
1   12   1563     0.082 0.058  0.075
2   24   1090     0.129 0.084  0.115
3   36    730     0.158 0.094  0.137
4   48    444     0.184 0.102  0.156
5   60    185     0.215 0.104  0.178
6   72     45     0.219 0.108  0.184

>
> # integrated brier score
> crps(brier)

Integrated Brier score (crps):

          IBS[0;time=80)
Reference          0.154
RF                 0.086
Model6             0.132
```