Event History Analysis in R

Day 2: Multivariate Analyses

Constantin Manuel Bosancianu

Doctoral School of Political Science
Central European University, Budapest
bosancianu@icloud.com

November 3, 2015

Outline

Yesterday was more theoretical - basic concepts in survival analysis, and a few descriptive summaries of the data.

Today I'll try going for an ambitious target.

- ✓ Cox Proportional Hazards (Cox PH) model
- ✓ Investigating the PH assumption
- ✓ Going past the PH assumption[1]

[1] Stratified Cox PH model and the extended Cox model.

Reason

Why is Cox PH so important?

Without giving away too much of the ending, it should be said that it’s the “OLS model” of survival analysis:

- ✓ Simple in its mechanics
- ✓ Robust to slight violations in its assumptions (particularly the PH one)[2]
- ✓ Much better than a logistic regression model, because it uses information on survival time as well

[2] There are a few other benefits, but it’s best to leave them until we get to that section.

Cox Proportional Hazards model

Cox PH model

So far, we've been able to compare survivor probabilities and hazard rates between distinct groups in our sample (e.g. men vs. women, countries from South America vs. everywhere else).

The Cox PH model is intended to provide additional controls when judging these relationships, since it can estimate hazard rates for groups in the sample, *after adjusting for any number of categorical and continuous time-invariant predictors*.

Roughly speaking, the Cox PH model is to the log-rank test what OLS regression is to the t-test.

Cox PH model

Let's use X to denote a set (vector) of independent variables, $X_1, X_2, \dots X_k$.

The formula for the Cox PH model is

$$h(t, X) = h_0(t) \times e^{\sum\limits_{i=1}^{k} \beta_i \times X_i} \tag{1}$$

The model estimates the hazard rate for an individual at time t, based on a "baseline hazard function", $h_0(t)$, and the values on the vector of independent variables.

Cox PH model

Two interesting features of the model:

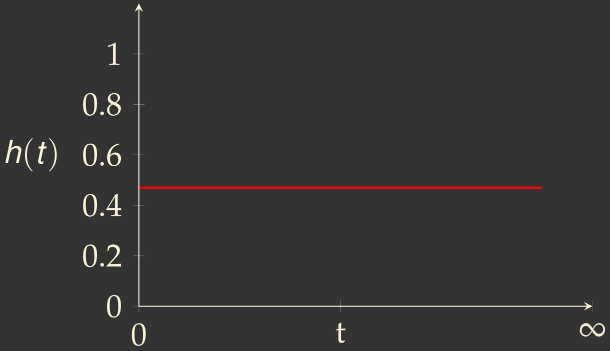
- ✓ $h_0(t)$ is a function that depends on time, but not on the Xs
- ✓ $e^{\sum\limits_{i=1}^{k} \beta_i \times X_i}$ depends on the Xs, but not on time[3]

If $X_1 = X_2 = \cdots = X_k = 0$, then $h(t, X) = h_0(t) \times e^0 = h_0(t)$, which is why it's called the "baseline function".

[3] The Xs for now are time-invariant variables, e.g. the continent location for a country. If you want to use time-variant variables as well, the PH assumption no longer holds ("extended" Cox model).

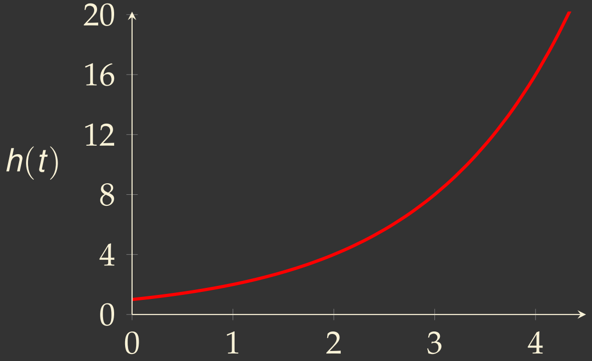
The baseline function

$h_0(t)$ is interesting in itself–it can take any functional form.



Constant hazard function

The baseline function



Weibull hazard function

Other functions are also possible, such as the decreasing Weibull, or the lognormal.

The baseline function

Because of its ability to handle multiple forms for the baseline hazard function, the Cox PH model is a “semiparametric” model.

Other benefits (apart from those mentioned already):

- ✓ Due to its robustness, it’s a “safe” choice (Kleinbaum & Klein, 2012, p. 111)
- ✓ The exponential part guarantees we never get negative predicted hazard rates, which are mathematically impossible

Estimates

No point in going into the estimation procedure which is a flavour of Maximum Likelihood (partial ML).

The estimates that you get from the model, though, are (indirectly) hazard ratios.

$$\widehat{HR} = \frac{\widehat{h}(t, X^*)}{\widehat{h}(t, X^\dagger)} \tag{2}$$

Here, X^* and $X^\dagger$ denote two sets of values on the explanatory variables for two observations.

Estimates

Let's say that $X^* = 1$ denotes a country from Latin America from my data set, while $X^\dagger = 0$ refers to all other countries.

$$\widehat{HR} = \frac{\widehat{h}(t, X^*)}{\widehat{h}(t, X^\dagger)} = \frac{\cancel{\widehat{h}_0^*(t)} \times e^{\sum_{i=1}^{k} \hat{\beta}_i \times X_i^*}}{\cancel{\widehat{h}_0^\dagger(t)} \times e^{\sum_{i=1}^{k} \hat{\beta}_i \times X_i^\dagger}} \tag{3}$$

Essentially, the Cox PH model needs the PH assumption for those hazard functions to cancel each other out. The baseline hazard is not even estimated.[4]

[4] Because of the PH assumption, $h_0^* = h_0^\dagger \times \theta$, where θ is a constant. The θ disappears as well when we take the logarithm of the quantity.

Estimates

$$\widehat{HR} = \frac{e^{\sum_{i=1}^{k} \hat{\beta}_i \times X_i^*}}{e^{\sum_{i=1}^{k} \hat{\beta}_i \times X_i^\dagger}} = e^{\sum_{i=1}^{k} \hat{\beta}_i (X_i^* - X_i^\dagger)} \tag{4}$$

In my case, $X^* = 1$ and $X^\dagger = 0$, which means the equation reduces to $\widehat{HR} = e^{\beta}$.

The coefficients we get from the model are the βs, but we can get to hazard ratios by computing e^{β} and thus getting the hazard ratio.

Hazard ratios

They are like “odds ratios” in logistic regression. A HR larger than one denotes a higher hazard rate for the first group than the second.

Suppose the coefficient in my model would have been 2. $\widehat{HR}$=7.3890561, which means drug users would be at a much higher risk of dying compared to non-users.

If $\beta > 0$, the $HR > 1$. If $\beta < 0$, then $HR < 1$.

Practical Cox PH

I won’t use the same data as yesterday; it’s a sexy topic, but I couldn’t get any model to show something interesting *and* statistically significant.

Today we’ll rely on a data set (from Hosmer & Lemeshow, 1999) comprised of 100 HIV infected persons, tracking their survival time.

```
path <- "https://stats.idre.ucla.edu/stat/r/examples/asa/hmohiv.csv"
hmohiv <- read.table(path, sep = ",", header = TRUE)
```

Practical Cox PH

```
hmohiv |>
  glimpse()
```

```
Rows: 100
Columns: 7
$ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
$ time    <int> 5, 6, 8, 3, 22, 1, 7, 9, 3, 12, 2, 12, 1, 15, 34, 1, 4, 19, 3,~
$ age     <int> 46, 35, 30, 30, 36, 32, 36, 31, 48, 47, 28, 34, 44, 32, 36, 36~
$ drug    <int> 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0,~
$ censor  <int> 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0,~
$ entdate <chr> "5/15/1990 ", "9/19/1989 ", "4/21/1991 ", "1/3/1991 ", "9/18/1~
$ enddate <chr> "10/14/1990 ", "3/20/1990 ", "12/20/1991 ", "4/4/1991 ", "7/19~
```

Practical Cox PH

```r
library(Zelig)
z.out <- zelig(Surv(time, censor) ~ drug + age,
               model = "coxph", data = hmohiv)
```

Unfortunately, the model does not work in `Zelig`, although it was available in previous versions (e.g. 3.5.x).

Keep following it, though, it's bound to appear again soon.

Practical Cox PH: `survival`

```
colnames(hmohiv)[5] <- "death" # the variable name was odd
model.1 <- coxph(Surv(time, death) ~ drug + age, data = hmohiv)
summary(model.1)

Call:
coxph(formula = Surv(time, death) ~ drug + age, data = hmohiv)

  n= 100, number of events= 80

        coef exp(coef) se(coef)     z Pr(>|z|)
drug 1.01670   2.76405  0.25622 3.968 7.24e-05 ***
age  0.09714   1.10202  0.01864 5.211 1.87e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     exp(coef) exp(-coef) lower .95 upper .95
drug     2.764     0.3618     1.673     4.567
age      1.102     0.9074     1.062     1.143

Concordance= 0.711  (se = 0.033 )
Likelihood ratio test= 39.13  on 2 df,   p=3e-09
Wald test            = 36.13  on 2 df,   p=1e-08
Score (logrank) test = 38.39  on 2 df,   p=5e-09
```

`Surv()` creates the `survival` object which is used as a response variable in the analysis.

Practical Cox PH: `survival`

Breaking down the results for the drug use dummy indicator.

- `coef`=1.0166984 is the estimated coefficient (a logged hazard ratio) for the dummy indicator;
- `exp(coef)`=2.7640538 is the hazard ratio for the variable;
- `se(coef)`=0.256217 is the standard error for the coefficient;
- `z`=3.9681137 is the z value ($\frac{\beta}{SE}$);
- `Pr(>|z|)`=7.2443761×10^{-5} is the p value associated with the coefficient.

Practical Cox PH: `survival`

The output presents a range of other pieces of information.

- ✓ Lower and upper CIs for the hazard ratio;
- ✓ LR test: compare the model with a null model (the baseline hazard rate) in terms of fit;
- ✓ R^2 value for the model fit (a Nagelkerke type of R^2).

Wald test is generally considered less reliable than the LR one.

Model fit: `concordance`

`Concordance` is a specific measure for the Cox PH model.

It denotes the percentage of pairs (A,B) of observations where the hazard of A is greater than that of B, and A gets the event before B.

It isn't very accurate through: it doesn't indicate *by how much sooner* A gets the event faster than B.

Cox PH: predictions

We could also obtain expected hazard ratios, with the `predict()` function.

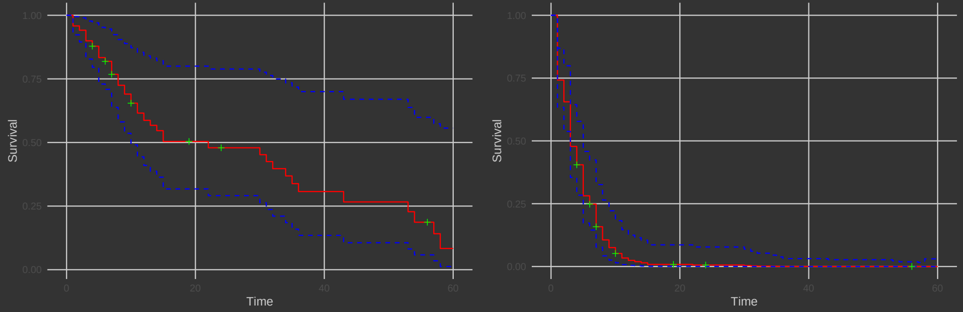
```r
predict(model.1, type = "risk", se.fit = TRUE) # hazard ratio
predict(model.1, type = "lp", se.fit = TRUE) # linear predictor
predict(model.1, type = "expected", se.fit = TRUE)
# expected number of events
```

The linear predictor is the $\sum_{i=1}^{i} \beta_i X_i$ quantity. Predictions of the “expected” type incorporate the baseline hazard rate as well.

Cox PH: predictions

We can also plot expected survival rates for particular "types" of individual in the data.

```r
plot(survfit(model.1, newdata = data.frame(drug = 1, age = 50)),
     xlab = "Time", ylab = "Survival")
```



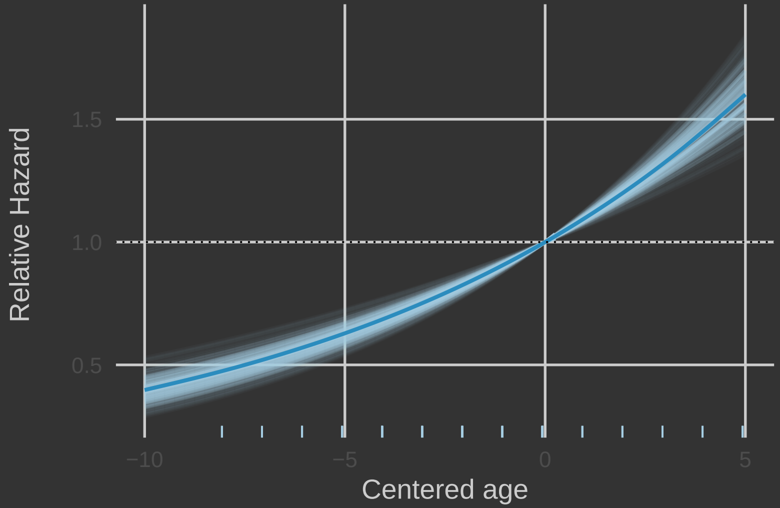
Left: `age`=20. Right: `age`=40. Comparison for drug users.

Cox PH: predictions

An alternative is the `simPH` package, which essentially fills the gap which `Zelig` left.

```r
library(simPH)
hmohiv <- hmohiv |> mutate(cent_age = age - mean(age, na.rm = TRUE))
model.2 <- coxph(Surv(time, death) ~ drug + cent_age, data = hmohiv)
sim.2 <- coxsimLinear(model.2, b = "cent_age", Xj = seq(-10, 8, by = 0.25),
                      nsim = 250)
# The smaller the "by=" interval, the smoother the graph
simGG(sim.2, xlab = "Centered age", type = "lines")
```

Cox PH: predictions



Cox PH: predictions

```r
sim.3 <- coxsimLinear(model.2, b = "drug", Xj = 0:1,
                      nsim = 250)
graph2 <- simGG(sim.3, xlab = "Drug use", type = "lines", method = "lm")
graph2 + scale_x_continuous(limits = c(0, 1),
    breaks = c(0, 1), labels = c("No drug use", "Drug use"))
```

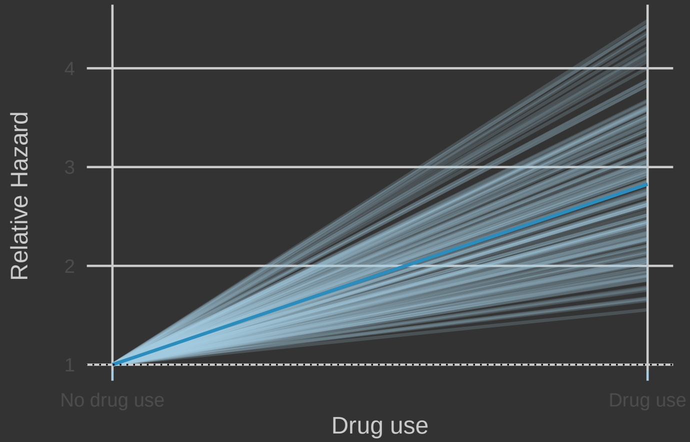
The option `method="lm"` is needed to make sure that the lines are straight, and don't get smoothed.

The `simGG()` function produces a `ggplot2` object, so we can modify a lot of things about it.[5]

[5] Remember the cool graph themes available in the `ggthemes` package.

Cox PH: predictions

We can obtain similar predictions for dichotomous variables.



Assessing the PH assumption

PH assumption

The assumption can be checked, to see if the Cox PH model is appropriate under the circumstances.

Three ways of checking:

- ✓ Graphically (through log–log curves)
- ✓ Through a goodness-of-fit (GOF) test
- ✓ Relying on time-dependent variables

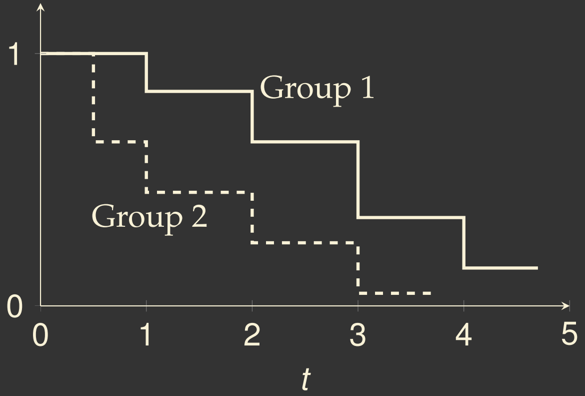
PH check: graphical

What we want to see is parallel log–log curves: $ln(-ln(\widehat{S}))$.

$\widehat{S} \in (0; 1)$, which means $-ln(\widehat{S}) \in (0; \infty)$, which means $ln(-ln(\widehat{S})) \in (-\infty; \infty)$.

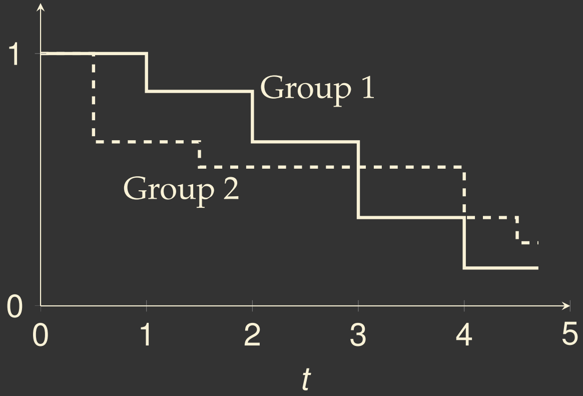
We have a hazard function from the Cox model, which we can convert to a survival function through some mathematical trasformations.

PH check: graphical



Roughly parallel lines mean the PH assumption is met

PH check: graphical



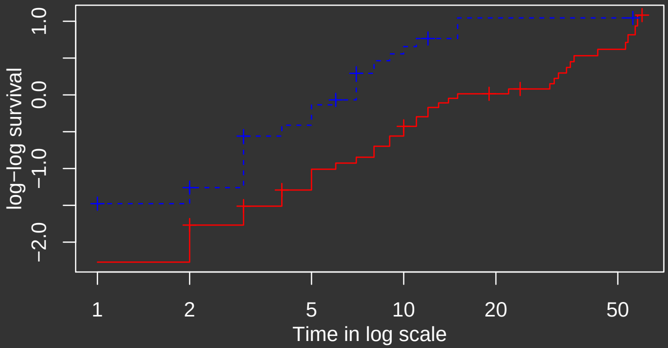
Crossing, converging or diverging lines mean the PH assumption is not met

PH check: graphical

Done for Kaplan-Meier survival estimates, one variable at a time.

```r
# This is not a Cox PH model
model.1 <- survfit(Surv(time, death) ~ drug, data = hmohiv)
plot(model.1, fun = "cloglog", lty = c("solid", "dashed"),
     col = c("red", "blue"), xlab = "Time in log scale",
     ylab = "log-log survival")
```

PH check: graphical



The PH assumption appears to be satisfied with respect to drug use

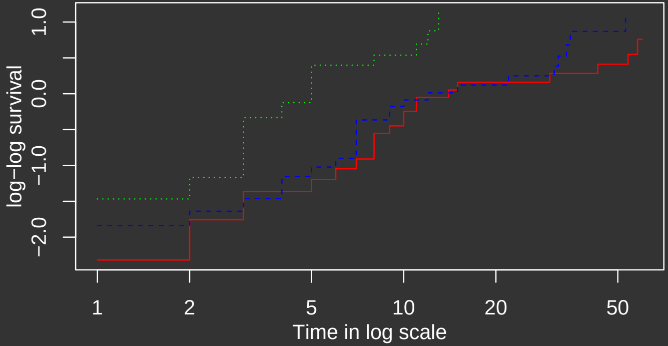
PH check: graphical

We can follow the same strategy with respect to continuous variables, but we have to split them into *k* categories.

```
quantile(hmohiv$age, c(0.33, 0.66)) # find ages at those precise
# locations in the age distribution
hmohiv$agecat[hmohiv$age <= 32] <- 0
hmohiv$agecat[hmohiv$age >= 33 & hmohiv$age <= 38] <- 1
hmohiv$agecat[hmohiv$age >= 39] <- 2
model.1 <- survfit(Surv(time, death) ~ agecat, data = hmohiv)
plot(model.1, fun = "cloglog", lty = c("solid", "dashed", "dotted"),
     col = c("red", "blue", "green"), xlab = "Time in log scale",
     ylab = "log-log survival")
```

It's good to choose a manageable number of categories, e.g. 2 or 3.

PH check: graphical



The PH assumption appears to be satisfied with respect to age too

Graphical checks: problems

The graphical check approach has a few inherent problems (Kleinbaum & Klein, 2012, pp. 172–174):

- ✓ Deciding whether lines are parallel is a subjective decision
- ✓ Deciding how to split variables into groups is also subjective
- ✓ Difficult strategy with reduced samples (some groups could be very small)

PH check: graphical

A second graphical approach is to plot expected vs. observed survival curves.

First stage involves running a Cox PH model and obtaining expected curves based on the values of the predictor.

The second stage is to obtain Kaplan–Meier curves as we did above.

We then plot the two sets of curves and inspect how closely they overlap.

GOF checks

They are appealing, because they provide a single digit, with an associated p value, on the basis of which to make a decision.

Based on Schoenfeld residuals, which is computed for each observation which experiences an event, for each of the DVs in the model.

Residuals are computed as $X_i - \mathit{weighted}(X_{\sim i})$, where X_i is observation's i value on the X variable, $X_{\sim i}$ are the values for all other observations, and the weights are those observations' hazard rates.

GOF checks: procedure

Three-stage procedure:

- ✓ Compute Schoenfeld residuals for each observation
- ✓ Construct a variable that orders survival times, from lowest to highest
- ✓ Correlate this variable with the residuals

If PH assumption is valid, there should be no correlation observed.

GOF checks: procedure

R does all of this for you automatically, with a slight twist: it uses scaled Schoenfeld residuals.[6]

```
model.1 <- coxph(Surv(time, death) ~ drug + age, data = hmohiv)
cox.zph(model.1, transform = "km", global = FALSE)

        chisq df    p
drug 0.000158  1 0.99
age  0.017640  1 0.89

# The "global" option does a global chi-square test
# in addition to one for each variable.
```

We want to see low correlations (ρ), and p values above 0.05.

[6] In the vast majority of cases, the results should be the same. The scaling procedure involves adjusting the residuals by the covariance matrix of the residuals.

Time-dependent checks

The procedure involves using a function of time, and interacting this with the predictors.

If the PH assumption is valid, the coefficient on the interaction should not be statistically significant.

$$h(t, X) = h_0(t) \times e^{\beta X + \delta(X \times g(t))} \tag{5}$$

$g(t) = t$, or $g(t) = log(t)$, or many others.

Time-dependent checks: R

```r
model.2 <- coxph(Surv(time, death) ~ drug + age + tt(drug), data = hmohiv,
                 tt=function(x, t, ...) x * t)
```

The function I chose here is simply $g(t) = t$.

The model has to be run with the `tt` argument. If you try computing by hand a $drug \times time$ interaction, the model will have convergence errors due to multicollinearity.

Time-dependent checks: R

```
summary(model.2); rm(model.2)

Call:
coxph(formula = Surv(time, death) ~ drug + age + tt(drug), data = hmohiv,
    tt = function(x, t, ...) x * t)

  n= 100, number of events= 80

             coef exp(coef) se(coef)      z Pr(>|z|)
drug      1.19633   3.30796  0.34992  3.419 0.000629 ***
age       0.09487   1.09951  0.01892  5.013 5.35e-07 ***
tt(drug) -0.02828   0.97212  0.03878 -0.729 0.465940
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

         exp(coef) exp(-coef) lower .95 upper .95
drug        3.3080     0.3023     1.666     6.568
age         1.0995     0.9095     1.059     1.141
tt(drug)    0.9721     1.0287     0.901     1.049

Concordance= 0.708  (se = 0.034 )
Likelihood ratio test= 39.81  on 3 df,   p=1e-08
Wald test            = 36.35  on 3 df,   p=6e-08
Score (logrank) test = 38.95  on 3 df,   p=2e-08
```

Stratified Cox model

Stratification

An extension to the case when one of the variables is clearly shown not to abide by the PH assumption.

The solution essentially involves running models separately for each category of the variable–stratifying based on the variable.

Unfortunately, everything was pretty good with the previous model, so there's not much we can do there.

Alternative data

Variable	Description	Values
ID	Identification code	1-628
AGE	Age at enrollment	Years
BECKTOTA	Beck depression score at admission	0-54
HERCOC	Heroin/cocaine use during 3 months prior to admission	1=both; 2=only heroin; 3=only cocaine; 4=neither
IVHX	Drug use history at admission	1=never; 2=previous; 3=recent
NDRUGTX	No. of prior drug treatments	0-40
RACE	Subject's race	0=White; 1=other
TREAT	Treatment randomization	0=short; 1=long
SITE	Treatment site	0=A; 1=B
LOT	Length of treatment	No. of days
TIME	Time to return to drug use	No. of days[7]
CENSOR	Returned to drug use	1=yes; 0=otherwise

[7] Measured starting from admission time.

Stratification

```r
df_uis <- read.table("../02-data/03-uissurv.dat", quote = "\"",
                     comment.char = "")
df_uis <- df_uis |>
  rename(id = 1, age = 2, becktota = 3, hercoc = 4, ivhx = 5,
         ndrugtx = 6, race = 7, treat = 8, site = 9, lot = 10,
         time = 11, censor = 12) |>
  mutate(becktota = if_else(becktota == ".", NA_character_, becktota),
         hercoc = if_else(hercoc == ".", NA_character_, hercoc),
         ivhx = if_else(ivhx == ".", NA_character_, ivhx),
         race = if_else(race == ".", NA_character_, race),
         age = if_else(age == ".", NA_character_, age)) |>
  mutate_at(.vars = vars(becktota, hercoc, ivhx, race, age),
            .funs = as.numeric)
model.3 <- coxph(Surv(time, censor) ~ age + becktota + ivhx + treat +
                   site + lot, data = df_uis, na.action = na.omit)
```

Stratification

```
Call:
coxph(formula = Surv(time, censor) ~ age + becktota + ivhx +
    treat + site + lot, data = df_uis, na.action = na.omit)

  n= 591, number of events= 475
   (37 observations deleted due to missingness)

               coef  exp(coef)   se(coef)       z Pr(>|z|)
age      -0.0196807  0.9805117  0.0079033  -2.490   0.0128 *
becktota  0.0040371  1.0040453  0.0047896   0.843   0.3993
ivhx      0.3014561  1.3518257  0.0570247   5.286 1.25e-07 ***
treat     0.1520115  1.1641736  0.0942117   1.614   0.1066
site      0.4972081  1.6441247  0.1071282   4.641 3.46e-06 ***
lot      -0.0092848  0.9907582  0.0008012 -11.588  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

         exp(coef) exp(-coef) lower .95 upper .95
age         0.9805     1.0199    0.9654    0.9958
becktota    1.0040     0.9960    0.9947    1.0135
ivhx        1.3518     0.7397    1.2089    1.5117
treat       1.1642     0.8590    0.9679    1.4003
site        1.6441     0.6082    1.3327    2.0283
lot         0.9908     1.0093    0.9892    0.9923

Concordance= 0.735  (se = 0.011 )
Likelihood ratio test= 180.9  on 6 df,   p=<2e-16
Wald test            = 159.6  on 6 df,   p=<2e-16
Score (logrank) test = 162.4  on 6 df,   p=<2e-16
```

Stratification

```
cox.zph(model.3, transform = "km", global = FALSE)

             chisq df      p
age          0.299  1 0.5847
becktota     2.575  1 0.1085
ivhx         0.038  1 0.8455
treat        9.882  1 0.0017
site         3.046  1 0.0809
lot        207.034  1 <2e-16
```

We clearly have a problem here with the `length of treatment` variable.

Stratified Cox

The model is a simple variation on the standard Cox model.

$$h_g(t, X) = h_{0g}(t) \times e^{\sum_{i=1}^{k} \beta_i X_i} \tag{6}$$

Here g designates the strata. We have separate baseline hazards for the different categories (strata).

Stratified Cox: practical

A simple extension of the R code as well.

```r
quantile(df_uis$lot, c(0.33, 0.66)) # examine cutoff points

  33%   66%
55.00 94.82

df_uis$lotcat[df_uis$lot < 55] <- 0
df_uis$lotcat[df_uis$lot >= 55 & df_uis$lot <= 94] <- 1
df_uis$lotcat[df_uis$lot > 94] <- 2
model.4 <- coxph(Surv(time, censor) ~ age + becktota + ivhx + treat +
                   site + strata(lotcat), data = df_uis, na.action = na.omit)
```

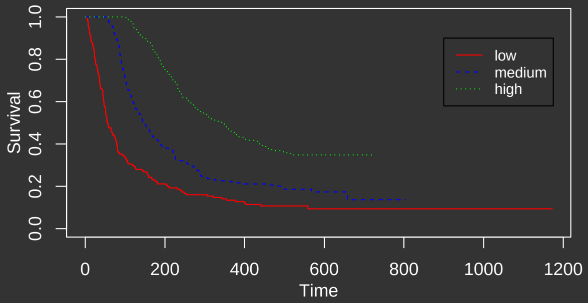
Stratified Cox: practical

```
Call:
coxph(formula = Surv(time, censor) ~ age + becktota + ivhx +
    treat + site + strata(lotcat), data = df_uis, na.action = na.omit)

  n= 591, number of events= 475
   (37 observations deleted due to missingness)

              coef exp(coef)  se(coef)      z Pr(>|z|)
age      -0.021194  0.979029  0.008056 -2.631  0.00852 **
becktota  0.005115  1.005128  0.004791  1.068  0.28563
ivhx      0.242769  1.274774  0.056782  4.275 1.91e-05 ***
treat     0.020763  1.020980  0.100144  0.207  0.83575
site      0.210432  1.234211  0.112827  1.865  0.06217 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

         exp(coef) exp(-coef) lower .95 upper .95
age          0.979     1.0214    0.9637    0.9946
becktota     1.005     0.9949    0.9957    1.0146
ivhx         1.275     0.7845    1.1405    1.4248
treat        1.021     0.9795    0.8390    1.2424
site         1.234     0.8102    0.9894    1.5397

Concordance= 0.559  (se = 0.014 )
Likelihood ratio test= 24.2  on 5 df,   p=2e-04
Wald test            = 24.01  on 5 df,   p=2e-04
Score (logrank) test = 24.13  on 5 df,   p=2e-04
```

Stratified Cox: practical

Each stratum will have the same coefficients on the βs, but their baseline hazard rates are probably different.

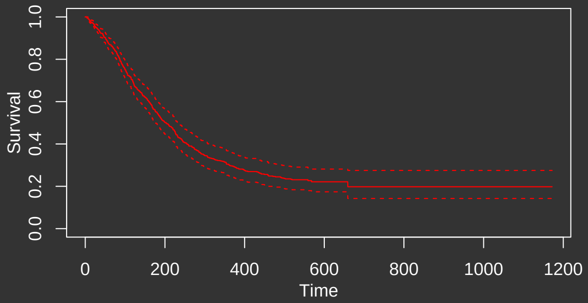
```r
plot(survfit(model.4))
```



Stratified Cox: practical

The “un-stratified” model will make no such distinction between basic hazard rates.

```
plot(survfit(model.3))
```



Stratified Cox

With multiple variables, they are added in the same `strata` field, e.g. `strata(IV1+IV2+...)`.

R will create categories based on them, e.g. 2 categories for `IV1` and 3 for `IV2` ⇒ 6 categories.

Assumption: there is no interaction between strata and the estimated parameters: $\beta_{1_{(1)}} = \beta_{1_{(2)}} = \beta_{1_{(3)}} = \beta_{1_{(4)}} = \dots$

Stratified Cox

It’s safe to test the assumption, with a series of interactions.

```
rm(model.3, model.4)
# Recode the LoT variable into only two categories, for convenience
quantile(df_uis$lot, 0.5) # median is 84

50%
 84

df_uis <- df_uis |> mutate(lotcat = if_else(lot < 84, 0, 1))
# Model with stratification
model.4.1 <- coxph(Surv(time, censor) ~ age + becktota + ivhx + treat +
                     site + strata(lotcat), data = df_uis, na.action = na.omit)
# Model with interactions
model.4.2 <- coxph(Surv(time, censor) ~ age + becktota + ivhx + treat +
                     site + age * lotcat + becktota * lotcat + ivhx * lotcat +
                     treat * lotcat + site * lotcat, data = df_uis,
                   na.action = na.omit)
```

Stratified Cox

```
round(summary(model.4.1)$coefficients,
      digits = 3)

           coef exp(coef) se(coef)      z Pr(>|z|)
age      -0.019     0.981    0.008 -2.394    0.017
becktota  0.007     1.007    0.005  1.437    0.151
ivhx      0.261     1.298    0.057  4.595    0.000
treat    -0.144     0.866    0.093 -1.543    0.123
site      0.079     1.082    0.106  0.741    0.459
```

Stratified Cox

```
round(summary(model.4.2)$coefficients,
      digits = 3)

                  coef exp(coef) se(coef)      z Pr(>|z|)
age             -0.042     0.959    0.012 -3.584    0.000
becktota         0.004     1.004    0.007  0.594    0.552
ivhx             0.433     1.543    0.084  5.154    0.000
treat            0.088     1.092    0.133  0.661    0.509
site             0.608     1.837    0.152  3.992    0.000
lotcat          -1.299     0.273    0.548 -2.371    0.018
age:lotcat       0.035     1.036    0.016  2.214    0.027
becktota:lotcat  0.005     1.005    0.010  0.506    0.613
ivhx:lotcat     -0.242     0.785    0.115 -2.105    0.035
treat:lotcat    -0.359     0.698    0.187 -1.921    0.055
site:lotcat     -0.788     0.455    0.210 -3.750    0.000
```

Stratified Cox

Likelihood ratio (LR) test: $-2 * LL_{M2} - (-2 * LL_{M1})$ is distributed χ^2, with $k_{M2} - k_{M1}$ degrees of freedom, where k is the number of predictors of the models.

```
lrtest(model.4.1, model.4.2)

Likelihood ratio test

Model 1: Surv(time, censor) ~ age + becktota + ivhx + treat + site + strata(lotcat)
Model 2: Surv(time, censor) ~ age + becktota + ivhx + treat + site + age *
    lotcat + becktota * lotcat + ivhx * lotcat + treat * lotcat +
    site * lotcat
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   5 -2387.5
2  11 -2664.1  6 553.18  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The coefficients do vary between strata.

Extended Cox model

Extended Cox

We can partition the vector of coefficients into two:

- ✓ X^* which only contains time-invariant variables: $X_1, X_2, \ldots X_k$
- ✓ $X^\dagger$ which only contains time-variant variables: $X_1(t), X_2(t), \ldots X_j(t)$

$$h(t, X(t)) = h_0(t) \times e^{\sum\limits_{i=1}^{k} \beta X_i^* + \sum\limits_{i=1}^{j} \delta X_i^\dagger(t)} \tag{7}$$

Even though $X_i^\dagger$ is time-variant, the estimated δ is the same for all time periods.

`heart` data

Variable	Description	Values
START	Time at which program was entered	-
STOP	Time when program was exited	-
EVENT	Death status	1=dead; 0=censored;
AGE	Age at acceptance into program	Years - 48
YEAR	Year of acceptance	Year - Nov 1, 1967
SURGERY	Previous surgery	1=yes; 0=no
TRANSPLANT	If ind. had transplant	1=yes; 2=no

Stanford Heart Transplant Study (1967-1974, N=103)

Extended Cox: practical

69 (67%) patients received a transplant and 75 (73%) patients died (45 recipients of new hearts and 30 non-recipients).

There is a problem, though: people who died sooner never got the chance to get a transplant, since it takes time to find a donor.

This might make transplants appear overly effective.

Extended Cox: practical

The `heart` data is available in the `survival` package.

```
data(heart)
heart |> glimpse()

Rows: 172
Columns: 8
$ start      <dbl> 0, 0, 0, 1, 0, 36, 0, 0, 0, 51, 0, 0, 0, 12, 0, 26, 0, 0, 1~
$ stop       <dbl> 50, 6, 1, 16, 36, 39, 18, 3, 51, 675, 40, 85, 12, 58, 26, 1~
$ event      <dbl> 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0,~
$ age        <dbl> -17.15537303, 3.83572895, 6.29705681, 6.29705681, -7.737166~
$ year       <dbl> 0.1232033, 0.2546201, 0.2655715, 0.2655715, 0.4900753, 0.49~
$ surgery    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ transplant <fct> 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0,~
$ id         <dbl> 1, 2, 3, 3, 4, 4, 5, 6, 7, 7, 8, 9, 10, 10, 11, 11, 12, 13,~
```

We don’t have a time variable, but `Surv()` can handle a beginning and end indicator, which we have in the data.

Extended Cox: practical

We don't have to worry about clustering at the level of individual; the software knows what observation to pick for inclusion in the likelihood function.

This is a new way of arranging the data, but it's the only way `survival` knows how to handle time-varying predictors.

```r
model.5 <- coxph(Surv(start, stop, event) ~ transplant + surgery + age + year,
                 data = heart)
```

Extended Cox: practical

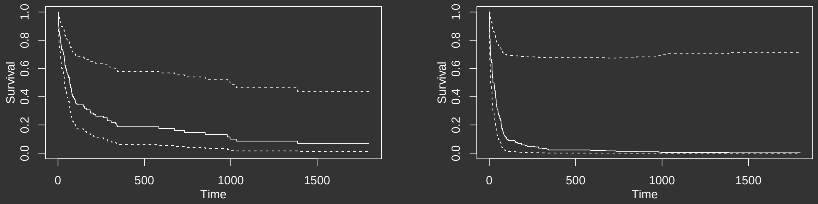
```
Call:
coxph(formula = Surv(start, stop, event) ~ transplant + surgery +
    age + year, data = heart)

  n= 172, number of events= 75

                coef exp(coef) se(coef)      z Pr(>|z|)
transplant1 -0.01025   0.98980  0.31375 -0.033   0.9739
surgery     -0.63721   0.52877  0.36723 -1.735   0.0827 .
age          0.02717   1.02754  0.01371  1.981   0.0476 *
year        -0.14635   0.86386  0.07047 -2.077   0.0378 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

            exp(coef) exp(-coef) lower .95 upper .95
transplant1    0.9898     1.0103    0.5352    1.8307
surgery        0.5288     1.8912    0.2574    1.0860
age            1.0275     0.9732    1.0003    1.0555
year           0.8639     1.1576    0.7524    0.9918

Concordance= 0.636  (se = 0.033 )
Likelihood ratio test= 15.11  on 4 df,   p=0.004
Wald test            = 14.49  on 4 df,   p=0.006
Score (logrank) test = 15.03  on 4 df,   p=0.005
```

Extended Cox: practical

```r
# Respondents of 38 years old, with no transplant
plot(survfit(model.5, newdata = data.frame(transplant = 0, age = -10,
                                           surgery = 0, year = 0)),
     xlab = "Time", ylab = "Survival")
# Respondents of 68 years old, with no transplant
plot(survfit(model.5, newdata = data.frame(transplant = 0, age = 20,
                                           surgery = 0, year = 0)),
     xlab = "Time", ylab = "Survival")
```

Respondents are similar in other respects: no past experience of surgery, and admitted to the program in November 1967.

Extended Cox: practical



Left: `age`=38. Right: `age`=68.

Clearly, older individuals have lower survival rates.

Concluding remarks

The Cox PH model is quite a good choice for the initial steps in the analysis.

It's robust, can accomodate a range of situations (some not covered here), and is relatively simple to run.

Other more complex models await: accelerated failure time models (AFT), frailty models etc.

Thank you for the kind attention!

References

Hosmer, D. W., Jr., & Lemeshow, S. (1999). *Applied Survival Analysis: Regression Modeling of Time to Event Data*. New York, NY: Wiley.

Kleinbaum, D. G., & Klein, M. (2012). *Survival Analysis: A Self-Learning Text*. New York, NY: Springer.