COX REGRESSION MODEL

Dr. Aric LaBarr
Institute for Advanced Analytics

PROPORTIONAL HAZARDS

Proportional Hazards Model

- Alternative to modeling failure time is to model hazards.
- **Proportional hazard (Cox Regression) model:** model the log of the hazard directly:

$$\log h(t) = \log h_0(t) + \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k}$$

- Hazard function is:

$$h(t) = h_0(t) e^{\beta_1 x_{i,1} + \cdots + \beta_k x_{i,k}}$$

- Predictions shift the hazard rather than directly shifting the failure time like in the AFT model.

Proportional Hazards Model – R

```
recid.ph <- coxph(Surv(week, arrest == 1) ~ fin + age + race +
                  wexp + mar +paro + prio, data = recid)

summary(recid.ph)
```

Proportional Hazards Model – R

```
## Call:
## coxph(formula = Surv(week, arrest == 1) ~ fin + age + race +
##     wexp + mar + paro + prio, data = recid)
##
##   n= 432, number of events= 114
##
##          coef exp(coef) se(coef)      z Pr(>|z|)
## fin  -0.37942   0.68426  0.19138 -1.983  0.04742 *
## age  -0.05744   0.94418  0.02200 -2.611  0.00903 **
## race  0.31390   1.36875  0.30799  1.019  0.30812
## wexp -0.14980   0.86088  0.21222 -0.706  0.48029
## mar  -0.43370   0.64810  0.38187 -1.136  0.25606
## paro -0.08487   0.91863  0.19576 -0.434  0.66461
## prio  0.09150   1.09581  0.02865  3.194  0.00140 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Proportional Hazards Model – R

```
##        exp(coef) exp(-coef) lower .95 upper .95
## fin       0.6843     1.4614    0.4702    0.9957
## age       0.9442     1.0591    0.9043    0.9858
## race      1.3688     0.7306    0.7484    2.5032
## wexp      0.8609     1.1616    0.5679    1.3049
## mar       0.6481     1.5430    0.3066    1.3699
## paro      0.9186     1.0886    0.6259    1.3482
## prio      1.0958     0.9126    1.0360    1.1591
##
## Concordance= 0.64  (se = 0.027 )
## Likelihood ratio test= 33.27  on 7 df,   p=2e-05
## Wald test            = 32.11  on 7 df,   p=4e-05
## Score (logrank) test = 33.53  on 7 df,   p=2e-05
```

Hazard Ratio

- If a parameter estimate is positive, increases in that variable increase the expected hazard.
  - **Increase** the rate/risk of failure
- If a parameter estimate is negative, increases in that variable decrease expected hazard.
  - **Decrease** in the rate/risk of failure
- $100 \times \left(e^{\beta} - 1\right)$ is the % increase in the expected hazard for each one-unit increase in the variable.
- e^{β} is the hazard ratio – the ratio of the hazards for each one-unit increase in the variable.

Recidivism Parameter Interpretation

Variable	β Estimate	$100(e^{\beta} - 1)$
Financial Aid	-0.347	-29.3%
Age at Release	-0.067	-6.5%
Prior Convictions	0.097	10.2%

MORE ON PMLE

Semiparametric Models

- In AFT and PH models, estimation depends on some distributional assumption around either the failure time or the baseline hazard.
- However, in PH models, Cox noticed that the likelihood can be split into two pieces:
  - 1st piece: depends on $h_0(t)$ and the parameters
    - Treat as non-parametric (no assumptions about form or distribution)
  - 2nd piece: **only** depends on the parameters
    - Treat as parametric (know the form)
- This is why it is called a **semiparametric** model.

Cox Regression Model

- Using the semiparametric model approach, we can basically ignore ever estimating anything about the baseline hazard $h_0(t)$ – the **Cox regression model**.

- Basically, Cox disregarded the first piece of the likelihood and maximized the second piece – still a PH model.

Partial Likelihood Estimation

- This is the more important piece of the work done by Sir David Cox in his original article.
- Estimates are obtained by maximizing the **partial likelihood** – only one piece that depends on the predictors, not the entire thing.
  - Done based on ranks of failure times – don't depend on baseline hazard.
  - All we care about is ratios between hazards.

Too Much Info on PMLE

- Since estimation for Cox regression uses ranks, ties can be problematic.
- Common methods to construct an appropriate partial likelihood for breaking ties: Efron (R default), Breslow (SAS default), exact
- If there are a new/no ties any of these would work just fine.
- Safe to go with Efron because it does better for higher numbers of ties.

DIAGNOSTICS

Linearity

Residual Plots

- Martingale residual plots in **R** are useful for checking linearity of predictors by plotting them vs. the predictor.
  - Similar to looking for residual patterns in linear regression revealing lack of linearity.

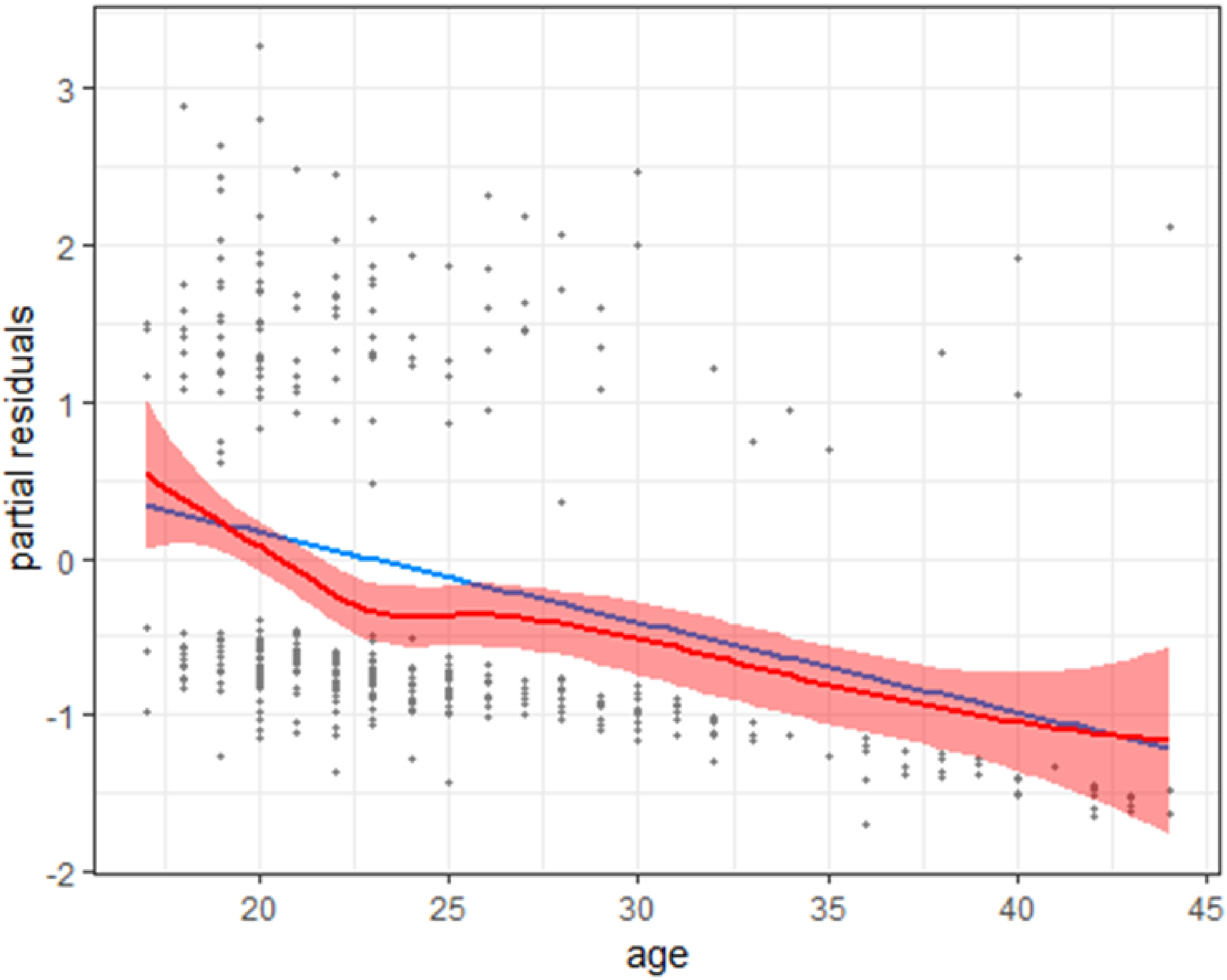
- Cumulative martingale residual plots in **SAS** compared to the predictor (or time) can also be used for determining linearity.

Linearity – R

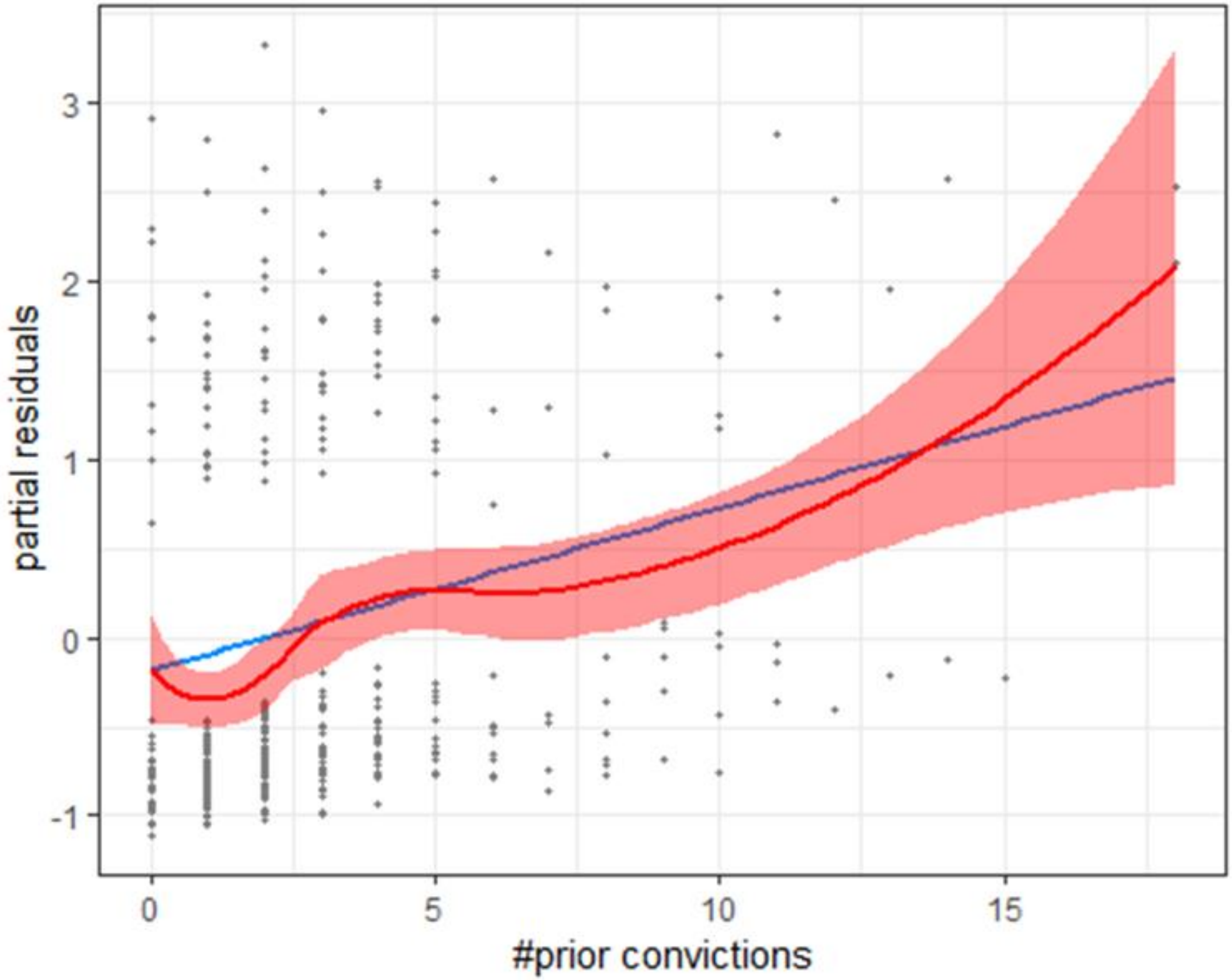
```
visreg(recid.ph, "age", xlab = "age", ylab = "partial residuals",
       gg = TRUE, band = FALSE) +
  geom_smooth(col = "red", fill = "red") + theme_bw()

visreg(recid.ph, "prio", xlab = "#prior convictions",
       ylab = "partial residuals",
       gg = TRUE, band = FALSE) +
  geom_smooth(col = "red", fill = "red") + theme_bw()
```

Linearity – R



Linearity – R



DIAGNOSTICS

Tests for Proportional Hazards

Schoenfeld Residuals

- Schoenfeld residuals are best used for investigating relationships with time for predictor variables since they are calculated on a per variable basis.
- You can plot these residuals against **functions** of time or the more popular technique would be to test the correlation between these residuals and **functions** of time.
- Which functions?
    - Common examples: t, $\log(t)$, K-M estimate, etc.

Proportional Hazard Test – R

```
recid.ph.zph <- cox.zph(recid.ph, transform = ...)
recid.ph.zph
```

Fill with one of: “km”, “identity”, “log”, or “rank”

Proportional Hazard Test – R

“identity”

```
##              rho   chisq        p
## fin      0.02161  0.0562 0.812654
## age     -0.27357 12.0614 0.000515
## race    -0.11497  1.4861 0.222824
## wexp     0.22643  6.9348 0.008453
## mar      0.07648  0.7544 0.385086
## paro    -0.03211  0.1220 0.726831
## prio    -0.00939  0.0109 0.916881
## GLOBAL        NA 18.1561 0.011285
```

“log”

```
##              rho   chisq        p
## fin      0.06391  0.4914 0.483319
## age     -0.28482 13.0738 0.000299
## race    -0.09576  1.0311 0.309895
## wexp     0.20238  5.5398 0.018589
## mar      0.08934  1.0293 0.310329
## paro     0.00942  0.0105 0.918399
## prio     0.05576  0.3840 0.535460
## GLOBAL        NA 17.6783 0.013509
```

AUTOMATIC SELECTION TECHNIQUES

Automatic Selection Techniques

- One of the benefits of PROC PHREG is the automatic selection techniques that it employs.
- Has similar selection techniques as PROC LOGISTIC:
  - Best
  - Forward
  - Backward
  - Stepwise

Automatic Selection Techniques – R

```r
stepAIC(coxph(Surv(week, arrest == 1) ~ fin + age + race + wexp +
              mar +paro + prio, data = recid))
```

⋮

```
##             coef exp(coef) se(coef)      z       p
## fin  -0.36020   0.69753  0.19049 -1.891 0.05864
## age  -0.06042   0.94137  0.02085 -2.897 0.00376
## mar  -0.53312   0.58677  0.37276 -1.430 0.15266
## prio  0.09751   1.10243  0.02722  3.583 0.00034
##
## Likelihood ratio test=31.41  on 4 df, p=2.528e-06
## n= 432, number of events= 114
```

PREDICTIONS

Estimating Survival Curves

- Once we've obtained parameter estimates from the partial likelihood, we can plug it into the full likelihood and nonparametrically estimate the remaining piece.
  - Think combining partial MLE and Kaplan-Meier…

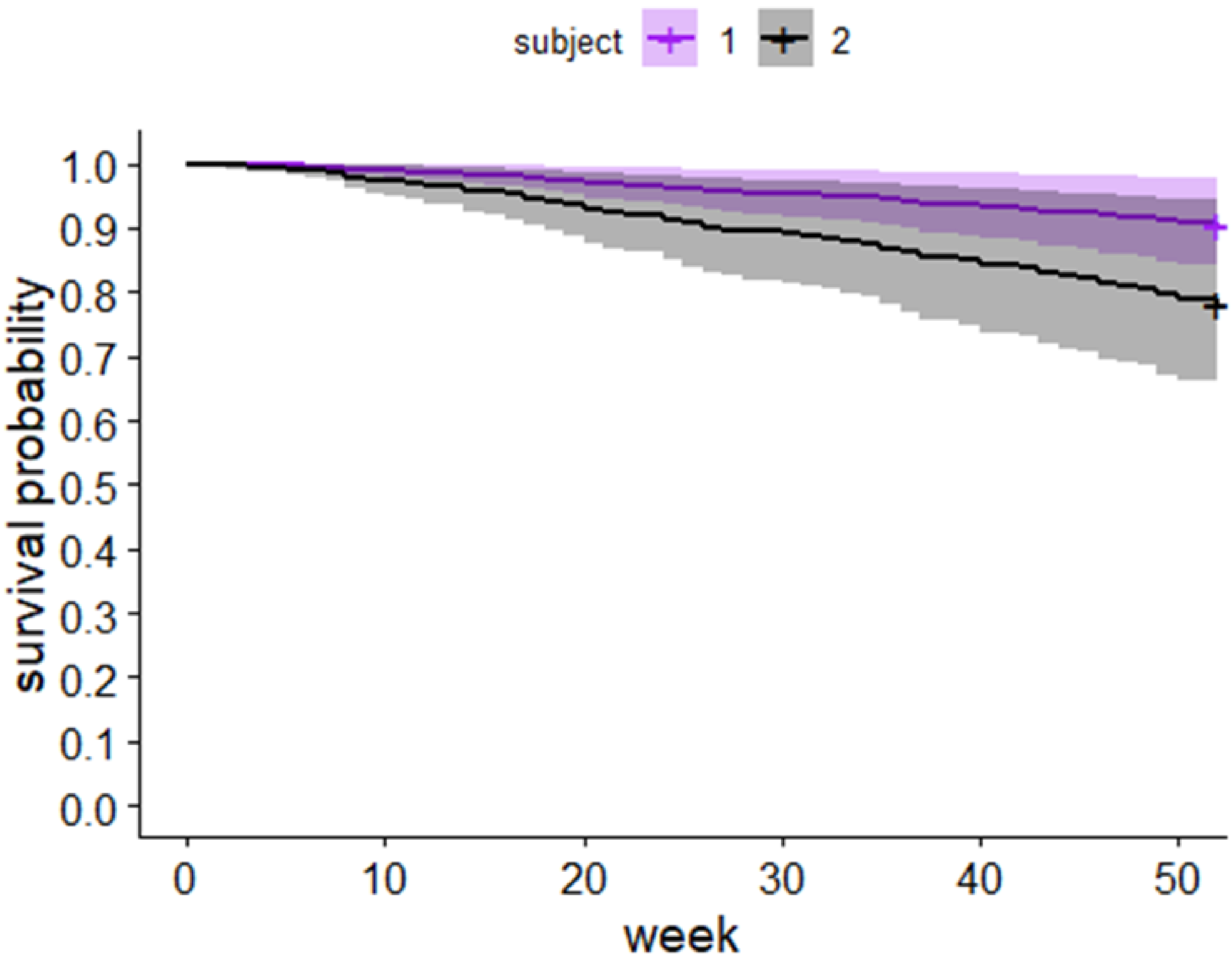
- Now we can estimate survival curves for predefined predictor values (combinations of the x's).

Estimated Survival Curves – R

```
newdata <- data.frame(fin = c(1, 0), age = 30, race = 0,
                      wexp = c(1, 0), mar = 0, paro = 0,
                      prio = c(0, 4))

ggsurvplot(survfit(recid.ph, newdata), data = newdata,
           break.y.by = 0.1, palette = c("purple", "black"),
           ylab = "Survival Probability", xlab = "week",
           legend.labs = c("1", "2"), legend.title = "subject")
```

Estimated Survival Curves – R



MODEL ASSESSMENT

Concordance

- What is “risk” in this context?
  - Risk: $\hat{\beta}_1 x_{i,1} + \cdots + \hat{\beta}_k x_{i,k}$
  - Piece of the model dealing with the predictors

- Example:
  - Person 1: event at $t = 3$ and $\hat{\beta}_1 x_{i,1} + \cdots + \hat{\beta}_k x_{i,k} = 1.5$
  - Person 2: event (or censored) at $t = 7$ and $\hat{\beta}_1 x_{i,1} + \cdots + \hat{\beta}_k x_{i,k} = 0.3$
  - Concordant pair since person with higher “risk” score had the event first.

Concordance – R

```
concordance(recid.ph)
```

```
## Call:
## concordance.coxph(object = recid.ph)
##
## n= 432
## Concordance= 0.6403 se= 0.02666
## discordant concordant     tied.x     tied.y    tied.xy
##      27242      15291         49        111          0
```

NON-PROPORTIONAL HAZARD MODELS

Time-dependent coefficients

Time Dependent Coefficients

- Models up until this point have assumed that predictors have a constant effect, β, on the target variable.
- In PH models, we assume effects are **constant over time**, so that the hazard ratio is independent of time.
- What if this didn't hold true and the effect of the predictor variable could change across time?
  - Example: Does age have a constant effect throughout the study?
- These effects, $\beta(t)$, are called **time-dependent coefficients**.

Time Dependent Coefficients

- If your software of choice tells you that you need one of these, what do you do?
- Need to add these time-dependent coefficients, but luckily SAS and R can easily do this for you.

$$\log h(t) = \log h_0(t) + \beta_1 x_{i,1} + \beta_2(t) x_{i,2}$$

Time Dependent Coefficients – R

```
recid.ph.tdc <- coxph(Surv(week, arrest == 1) ~ fin + race +
                      wexp + mar + paro + age + tt(age),
                      data = recid,
                      tt = function(x, time, ...){x*log(time)})

summary(recid.ph.tdc)
```

Time Dependent Coefficients – R

```
##               coef exp(coef) se(coef)      z Pr(>|z|)
## fin       -0.36196   0.69631  0.19073 -1.898  0.05773 .
## race       0.26275   1.30050  0.30677  0.857  0.39171
## wexp      -0.28437   0.75249  0.20529 -1.385  0.16598
## mar       -0.36769   0.69233  0.38055 -0.966  0.33394
## paro      -0.16886   0.84462  0.19353 -0.873  0.38290
## age        0.11703   1.12415  0.06521  1.795  0.07270 .
## tt(age)   -0.05777   0.94387  0.02177 -2.653  0.00798 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## fin          0.6963     1.4361    0.4791     1.012
## race         1.3005     0.7689    0.7128     2.373
## wexp         0.7525     1.3289    0.5032     1.125
## mar          0.6923     1.4444    0.3284     1.460
## paro         0.8446     1.1840    0.5780     1.234
## age          1.1242     0.8896    0.9893     1.277
## tt(age)      0.9439     1.0595    0.9044     0.985
```

Interpretation

- Let's use our example with age having a time-dependent coefficient:

$$\beta_{\text{age}}(t) = 0.173 - 0.077 \times \log(\text{week})$$

- Initially, it seems for short periods of time (low week number), being older is actually worse since the coefficient is positive (0.173).
- However, as time goes on, this effect decreases (-0.077) to the point of being better to be older after week 1.

NON-PROPORTIONAL HAZARD MODELS

Time-dependent Variables

Time Dependent Variables

- Similar to time-dependent coefficients, **time-dependent variables** have the actual value of the predictor variable (rather than its effect) change over time.
- Time *independent* variable examples:
  - Age (at entry)
  - Race
- Time *dependent* variable examples:
  - Employment status
  - Blood pressure

Counting Process Structure

- For time-dependent variables, it is necessary to split the *time* column of your data set into separate *start* and *stop* columns.
- This is known as the **counting process** structure/layout to your data.

- This is NEEDED for R to do the analysis.
- SAS will do this for you!

Counting Process Example

- Person 1 has an event at time = 9, but their value of *x* changes after time = 5.
- Observe Person 1 until end of time = 5, after which they are censored:

Person	Start	Stop	x	Event
1	0	5	3	0

- Create a “new” person starting after time = 5 who is the *exact same* as Person 1, but with new *x* value:

Person	Start	Stop	x	Event
1	0	5	3	0
1	5	9	7	1

Counting Process Example

- Create a “new” person starting after time = 5 who is the *exact same* as Person 1, but with new *x* value:

Person	Start	Stop	x	Event
1	0	5	3	0
1	5	9	7	1

- We observe this “new” person until either *x* changes again or their tenure ends (whichever comes first).

Time-Dependent Variables – R

```
recid_long.ph <- coxph(Surv(start, stop, arrested == 1) ~ fin
                       + age + race + wexp + mar + paro + prio
                       + employed, data = recid_long)

summary(recid_long.ph)
```

Time-Dependent Variables – R

```
##                coef exp(coef) se(coef)      z Pr(>|z|)
## fin       -0.35672   0.69997  0.19113 -1.866  0.06198 .
## age       -0.04634   0.95472  0.02174 -2.132  0.03301 *
## race       0.33866   1.40306  0.30960  1.094  0.27402
## wexp      -0.02555   0.97477  0.21142 -0.121  0.90380
## mar       -0.29375   0.74546  0.38303 -0.767  0.44314
## paro      -0.06421   0.93781  0.19468 -0.330  0.74156
## prio       0.08514   1.08887  0.02896  2.940  0.00328 **
## employed -1.32832   0.26492  0.25072 -5.298 1.17e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## fin         0.7000     1.4286    0.4813    1.0180
## age         0.9547     1.0474    0.9149    0.9963
## race        1.4031     0.7127    0.7648    2.5740
## wexp        0.9748     1.0259    0.6441    1.4753
## mar         0.7455     1.3414    0.3519    1.5793
## paro        0.9378     1.0663    0.6403    1.3735
## prio        1.0889     0.9184    1.0288    1.1525
## employed    0.2649     3.7747    0.1621    0.4330
```

?