Applied survival analysis

Constantin T Yiannoutsos, Ph.D.

January 22, 2016

Contents of today's lecture

Model Selection in Survival Analysis

Suppose we have a censored survival time that we want to model as a function of a (possibly) set of covariates. Two important questions are:

- How to decide which covariates to use
- How to decide if the final model fits well

To address these topics, we'll consider a new example:

Survival of Atlantic Halibut - Smith et al

Obs #	*Survival Time* (min)	*Censoring* Indicator	*Tow Duration* (min.)	Diff in *Depth*	*Length* of Fish (cm)	*Handling* Time (min.)	Total *log(catch)* ln(weight)
100	353.0	1	30	15	39	5	5.685
109	111.0	1	100	5	44	29	8.690
113	64.0	0	100	10	53	4	5.323
116	500.0	1	100	10	44	4	5.323
⋮							

Process of Model Selection

Collett (Section 3.6) has an excellent discussion of various approaches for model selection. In practice, model selection proceeds through a combination of

- knowledge of the science
- trial and error, common sense
- automatic variable selection procedures
  - forward selection
  - backward selection
  - stepwise seletion

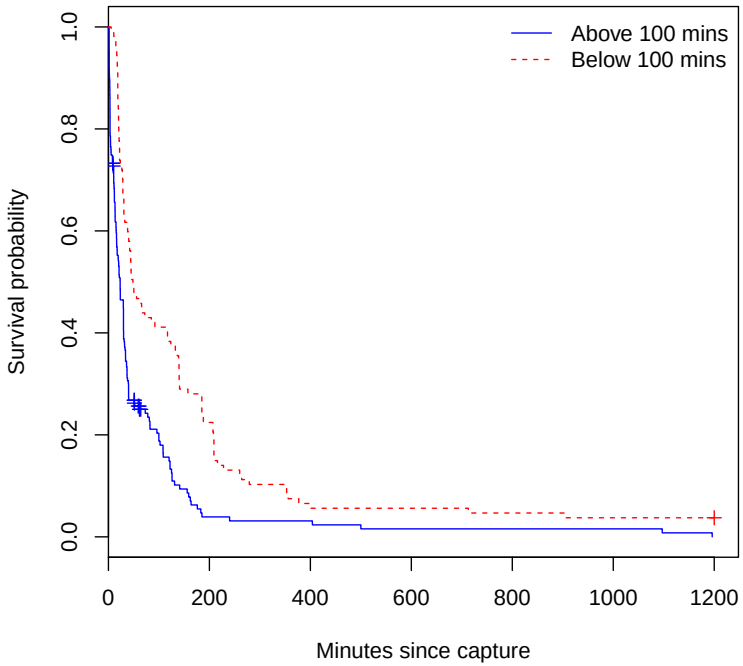
Many advocate the approach of first doing a univariate analysis to "screen" out potentially significant variables for consideration in the multivariate model (see Collett).

Let's start with this approach!

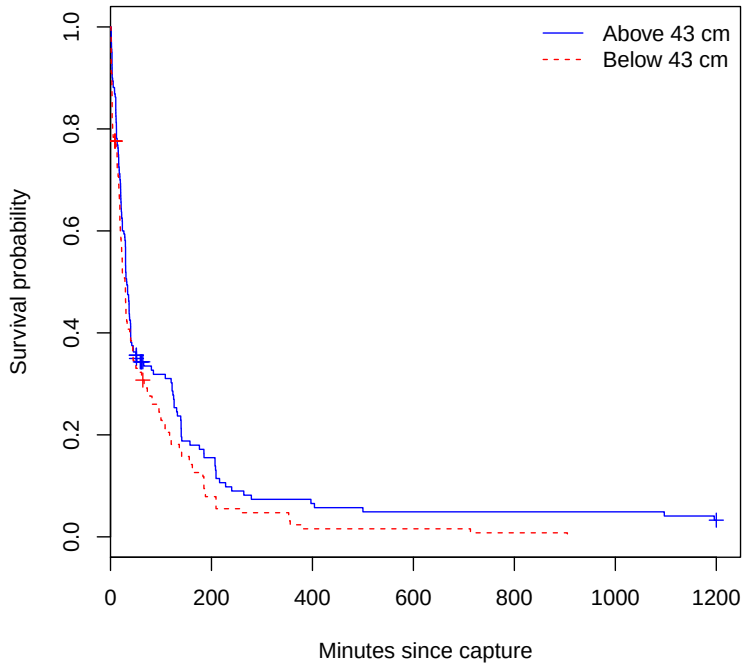
Univariate KM plots of Atlantic Halibut survival

Note: Continuous variables have been dichotomized at the median.

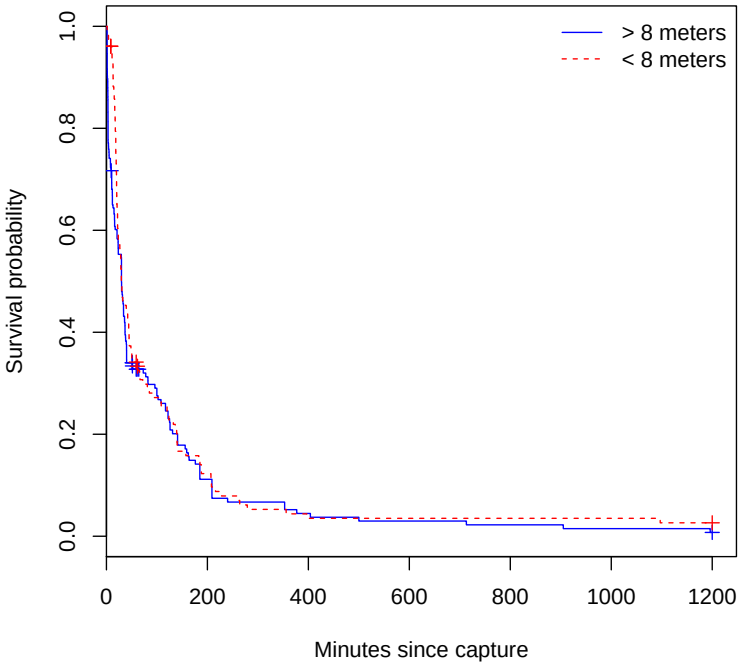
Tow duration (mins)



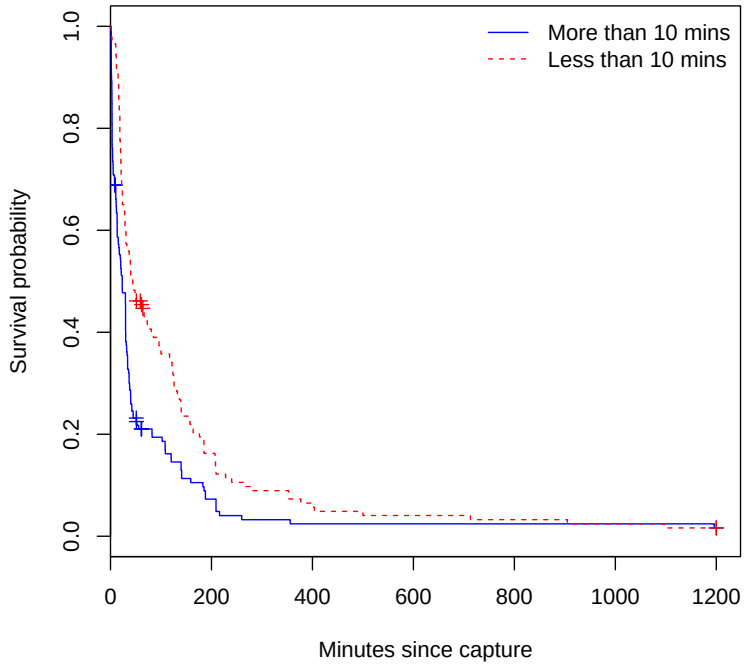
length (cm)



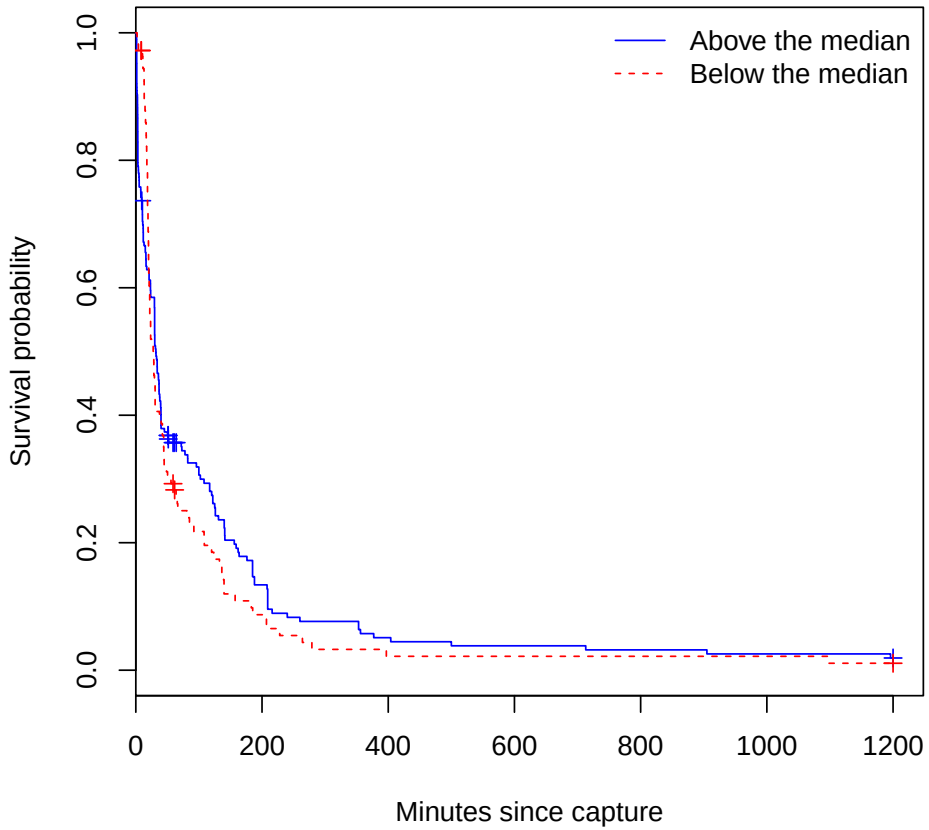
Depth (meters)



handling (mins)



log-catch



Which covariates look like they might be important?

Automatic Variable selection procedures in Stata, R and SAS

Statistical Software:

- Stata: `sw` command before `cox` command
- SAS: `selection=` option on model statement of `proc phreg`
- R: `stepAIC` command.

Options:

(1) forward

(2) backward

(3) stepwise (forward or backward in Stata and R)

(4) best subset (SAS only, using `score` option)

One drawback of these options is that they can only handle variables one at a time. When might that be a disadvantage?

Collett's Model Selection Approach

Collett **Section 3.6.1**

This approach assumes that all variables are considered to be on an equal footing, and there is no *a priori* reason to include any specific variables (like treatment).

(1) Fit a univariate model for each covariate, and identify the predictors significant at some level p_1, say 0.20.

(2) Fit a multivariate model with all significant univariate predictors, and use *backward* selection to eliminate non-significant variables at some level p_2, say 0.10.

(3) Starting with final step (2) model, consider each of the non-significant variables from step (1) using *forward* selection, with significance level p_3, say 0.10.

(4) Do final pruning of main-effects model (omit variables that are non-significant, add any that are significant), using *stepwise* regression with significance level p_4. At this stage, you may also consider adding interactions between any of the main effects currently in the model, under the hierarchical principle.

Collett recommends using a likelihood ratio test for all variable inclusion/exclusion decisions.

Model selection in R

R fits various models applying the AIC criteria described by Collett:

$$\text{minimize AIC} = -2 \log(\hat{L}) + (k * q)$$

where q is the number of unknown parameters in the model and α is typically between 2 and 6 (they suggest $k = 3$).

The model is then chosen which minimizes the AIC (similar to maximizing log-likelihood, but with a penalty for number of variables in the model)

R command for Forward selection

Forward Selection

We use the "forward" option. Each step is carried out so that the Akaike Information Criterion (Aic) is minimized. The R command is as follows:

```
fit.stepForw = stepAIC(fitmin,scope = list(lower = formula(fitmin),
                                           upper = formula(fitmax)),
                       direction = "forward",k = 3,trace = 1)
summary(fit.stepForw)
```

Where `fitmin` is the empty Cox model (i.e., a model without covariates) and `fitmax` is the saturated model (i.e., the Cox model with all possible covariates in it).

Output

The results are given in the following output:

```
Call:
coxph(formula = Surv(survtime, censor) ~ handling + logcatch +
    towdur + length, data = halibut)

  n= 294, number of events= 273

              coef exp(coef)  se(coef)      z Pr(>|z|)
handling  0.054947  1.056485  0.009870  5.567 2.59e-08 ***
logcatch -0.183373  0.832458  0.050982 -3.597 0.000322 ***
towdur    0.007744  1.007774  0.002017  3.839 0.000123 ***
length   -0.036960  0.963715  0.010028 -3.686 0.000228 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

          exp(coef) exp(-coef) lower .95 upper .95
handling     1.0565     0.9465    1.0362    1.0771
logcatch     0.8325     1.2013    0.7533    0.9199
towdur       1.0078     0.9923    1.0038    1.0118
length       0.9637     1.0377    0.9450    0.9828

Concordance= 0.683  (se = 0.02 )
Rsquare= 0.25   (max possible= 1 )
Likelihood ratio test= 84.5   on 4 df,   p=0
Wald test            = 90.71  on 4 df,   p=0
Score (logrank) test = 94.51  on 4 df,   p=0
```

R cmmand for Backward selection

Backward Selection We use the "backward" option. Each step is carried out so that the Akaike Information Criterion (Aic) is minimized. The R command is as follows:

```
fit.backward = stepAIC(fitmax,scope = list(lower = formula(fitmin),
                                           upper = formula(fitmax)),
                       direction = "backward", k = 3,trace = 1)
summary(fit.backward)
```

Output

```
Call:
coxph(formula = Surv(survtime, censor) ~ towdur + length + handling +
    logcatch, data = halibut)

  n= 294, number of events= 273

              coef exp(coef)  se(coef)      z Pr(>|z|)
towdur    0.007744  1.007774  0.002017  3.839 0.000123 ***
length   -0.036960  0.963715  0.010028 -3.686 0.000228 ***
handling  0.054947  1.056485  0.009870  5.567 2.59e-08 ***
logcatch -0.183373  0.832458  0.050982 -3.597 0.000322 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

          exp(coef) exp(-coef) lower .95 upper .95
towdur       1.0078     0.9923    1.0038    1.0118
length       0.9637     1.0377    0.9450    0.9828
handling     1.0565     0.9465    1.0362    1.0771
logcatch     0.8325     1.2013    0.7533    0.9199

Concordance= 0.683  (se = 0.02 )
Rsquare= 0.25   (max possible= 1 )
Likelihood ratio test= 84.5   on 4 df,   p=0
Wald test            = 90.71  on 4 df,   p=0
Score (logrank) test = 94.51  on 4 df,   p=0
```

R command for Stepwise selection

The stepwise procedure has a forward and a backward version.

The only difference is whether one starts with the empty or the saturated model. In both cases the model is refit and all variables that are not to be included in the model are discarded.

Forward and Backward stepwise selection

Forward stepwise Selection

We use the "both" option. Each step is carried out so that the Akaike Information Criterion (Aic) is minimized. The R command is as follows:

```
fit.stepBack = stepAIC(fitmin, scope = list(lower = formula(fitmin),
                                            upper = formula(fitmax)),
                       direction = "both", k = 3, trace = 1)
summary(fit.stepBack)
```

Backward stepwise Selection

```
fit.stepBack = stepAIC(fitmax, scope = list(lower = formula(fitmin),
                                            upper = formula(fitmax)),
                       direction = "both", k = 3, trace = 1)
summary(fit.stepBack)
```

Where `fitmin` is the empty Cox model (i.e., a model without covariates) and `fitmax` is the saturated model (i.e., the Cox model with all possible covariates in it).

Output: Backward stepwise selection

The results are given in the following output:

```
Call:
coxph(formula = Surv(survtime, censor) ~ towdur + length + handling +
    logcatch, data = halibut)

  n= 294, number of events= 273

              coef exp(coef)  se(coef)      z Pr(>|z|)
towdur    0.007744  1.007774  0.002017  3.839 0.000123 ***
length   -0.036960  0.963715  0.010028 -3.686 0.000228 ***
handling  0.054947  1.056485  0.009870  5.567 2.59e-08 ***
logcatch -0.183373  0.832458  0.050982 -3.597 0.000322 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

          exp(coef) exp(-coef) lower .95 upper .95
towdur       1.0078     0.9923    1.0038    1.0118
length       0.9637     1.0377    0.9450    0.9828
handling     1.0565     0.9465    1.0362    1.0771
logcatch     0.8325     1.2013    0.7533    0.9199

Concordance= 0.683  (se = 0.02 )
Rsquare= 0.25   (max possible= 1 )
Likelihood ratio test= 84.5   on 4 df,   p=0
Wald test            = 90.71  on 4 df,   p=0
Score (logrank) test = 94.51  on 4 df,   p=0
```

Output: Forward stepwise selection

```
Call:
coxph(formula = Surv(survtime, censor) ~ handling + logcatch +
    towdur + length, data = halibut)

  n= 294, number of events= 273

              coef exp(coef)  se(coef)      z Pr(>|z|)
handling  0.054947  1.056485  0.009870  5.567 2.59e-08 ***
logcatch -0.183373  0.832458  0.050982 -3.597 0.000322 ***
towdur    0.007744  1.007774  0.002017  3.839 0.000123 ***
length   -0.036960  0.963715  0.010028 -3.686 0.000228 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

          exp(coef) exp(-coef) lower .95 upper .95
handling     1.0565     0.9465    1.0362    1.0771
logcatch     0.8325     1.2013    0.7533    0.9199
towdur       1.0078     0.9923    1.0038    1.0118
length       0.9637     1.0377    0.9450    0.9828

Concordance= 0.683  (se = 0.02 )
Rsquare= 0.25   (max possible= 1 )
Likelihood ratio test= 84.5   on 4 df,   p=0
Wald test            = 90.71  on 4 df,   p=0
Score (logrank) test = 94.51  on 4 df,   p=0
```

Notes

- When the halibut data was analyzed with the forward, backward and stepwise options, the same final model was reached. However, this will not always be the case.

- Variables can be forced into the model using the `lockterm` option in Stata and the `include` option in SAS. Any variables that you want to force inclusion of must be listed first in your model statement.

- Stata uses the Wald test for both forward and backward selection, although it has an option to use the likelihood ratio test instead (`lrtest`). SAS uses the score test to decide what variables to add and the Wald test for what variables to remove.

Assessing overall model fit

How do we know if the model fits well?

- Always look at univariate plots (Kaplan-Meiers) Construct a Kaplan-Meier survival plot for each of the important predictors, like the ones shown at the beginning of these notes.
- Check proportionality assumption (this will be the topic of the next lecture)
- **Check residuals!**
  - **(a)** generalized (Cox-Snell)
  - **(b)** martingale
  - **(c)** deviance
  - **(d)** Schoenfeld
  - **(e)** weighted Schoenfeld

Using residuals to test model fit

Residuals for survival data are slightly different than for other types of models, due to the censoring.

Before we start talking about residuals, we need an important basic result:

Inverse CDF:
If T_i (the survival time for the i-th individual) has survivorship function $S_i(t)$, then the transformed random variable $S_i(T_i)$ (i.e., the survival function evaluated at the actual survival time T_i) should be from a uniform distribution on $[0, 1]$, and hence $-\log[S_i(T_i)]$ should be from a unit exponential distribution.

More mathematically,

$$\begin{aligned} \text{If} \quad T_i &\sim S_i(t) \\ \text{then} \quad S_i(T_i) &\sim \mathit{Uniform}[0,1] \\ \text{and} \quad -\log S_i(T_i) &\sim \mathit{Exponential}(1) \end{aligned}$$

The fact that $S(T_i) \sim U[0,1]$ results from the definition of the survival function, $S(x) = P(T > x) = 1 - P(T \leq x)$.

Now consider that $P(S(T) \leq x) = P(T \geq S^{-1}(x))$ by the fact that $g^{-1}g(x) = x$, where $g^{-1}(x)$ is the unique inverse of x, and because $S(\cdot)$ is non-increasing (so we reverse the sign of the inequality).

Thus, $P(S(T) \leq x) = S(S^{-1}(x)) = x$ which is the definition of a uniformly-distributed random variable.

The fact that $-\log S_i(T_i) \sim \mathit{Exponential}(1)$ results from the Inverse CDF Theorem.

Generalized (Cox-Snell) Residuals

The implication of the last result is that if the model is correct, the estimated cumulative hazard for each individual at the time of their death or censoring should be like a censored sample from a unit exponential. This quantity is called the *generalized* or *Cox-Snell* residual.

Here is how the generalized residual might be used. Suppose we fit a PH model:

$$S(t;Z) = [S_0(t)]^{\exp(\beta Z)}$$

or, in terms of hazards:

$$\begin{aligned} \lambda(t;Z) &= \lambda_0(t)\exp(\beta Z) \\ &= \lambda_0(t)\exp(\beta_1 Z_1 + \beta_2 Z_2 + \cdots + \beta_k Z_k) \end{aligned}$$

After fitting, we have:

- $\hat{\beta}_1, \ldots, \hat{\beta}_k$
- $\hat{S}_0(t)$

So, for each person with covariates $\mathbf{Z}_i$, we can get

$$\hat{S}(t; \mathbf{Z}_i) = [\hat{S}_0(t)]^{\exp(\beta \mathbf{Z}_i)}$$

This gives a predicted survival probability at each time t in the dataset (see notes from the previous lecture).

Then we can calculate

$$\hat{\Lambda}_i = -\log[\hat{S}(T_i; Z_i)]$$

In other words, first we find the predicted survival probability at the actual survival time for an individual, then log-transform it.

Example: Nursing home data

Say we have

- a single male
- with actual duration of stay of 941 days ($X_i = 941$)

We compute the entire distribution of survival probabilities for single males, and obtain $\hat{S}(941) = 0.260$.

$$-\log[\hat{S}(941, \text{single male})] \quad = \quad -\log(0.260) = 1.347$$

We repeat this for everyone in our dataset. These should be like a censored sample from an exponential (1) distribution if the model fits the data well.

Cox-Snell residuals in the nursing home data example

Based on the properties of a unit exponential model

- plotting $-\log(\hat{S}(t))$ vs t should yield a straight line
- plotting $\log[-\log S(t)]$ vs $\log(t)$ should yield a straight line through the origin with slope=1.

To convince yourself of this, start with $S(t) = e^{-\lambda t}$ and calculate $\log[-\log S(t)]$. What do you get for the slope and intercept?

(Note: this does not necessarily mean that the underlying distribution of the original survival times is exponential!)

Obtaining residuals from R

- Fit a Cox PH model with `coxph` comand
- Use the `residuals` command
- Use the `type` option to specify the type of the residual you want to obtain. R provides the following types of residuals:
  - martingale residuals
  - deviance residuals
  - score residuals
  - schoenfeld and scaled Schoenfeld residuals
  - dfbeta, dfbetas
  - partial residuals

Obtaining Cox-Snell residuals from R

We obtain Cox-Snell residuals from R as follows:

- Use the `predict` command with the `csnell` option
- Define a survival dataset using the Cox-Snell residuals as the "pseudo" failure times
- Calculate the estimated KM survival
- Take the $\log[-\log(S(t))]$ based on the above
- Generate the log of the Cox-Snell residuals
- Graph $\log[-\log S(t)]$ vs $\log(t)$

Cox-Snell residuals in the halibut data ample

- Fit the Cox model:

```
fit = coxph( Surv(survtime,censor) ~ towdur + handling + length
> + logcatch, data = halibut)
summary(fit)
```

- Define the residuals

```
halibut$csres = halibut$censor - residuals(fit,type = "martingale")
```

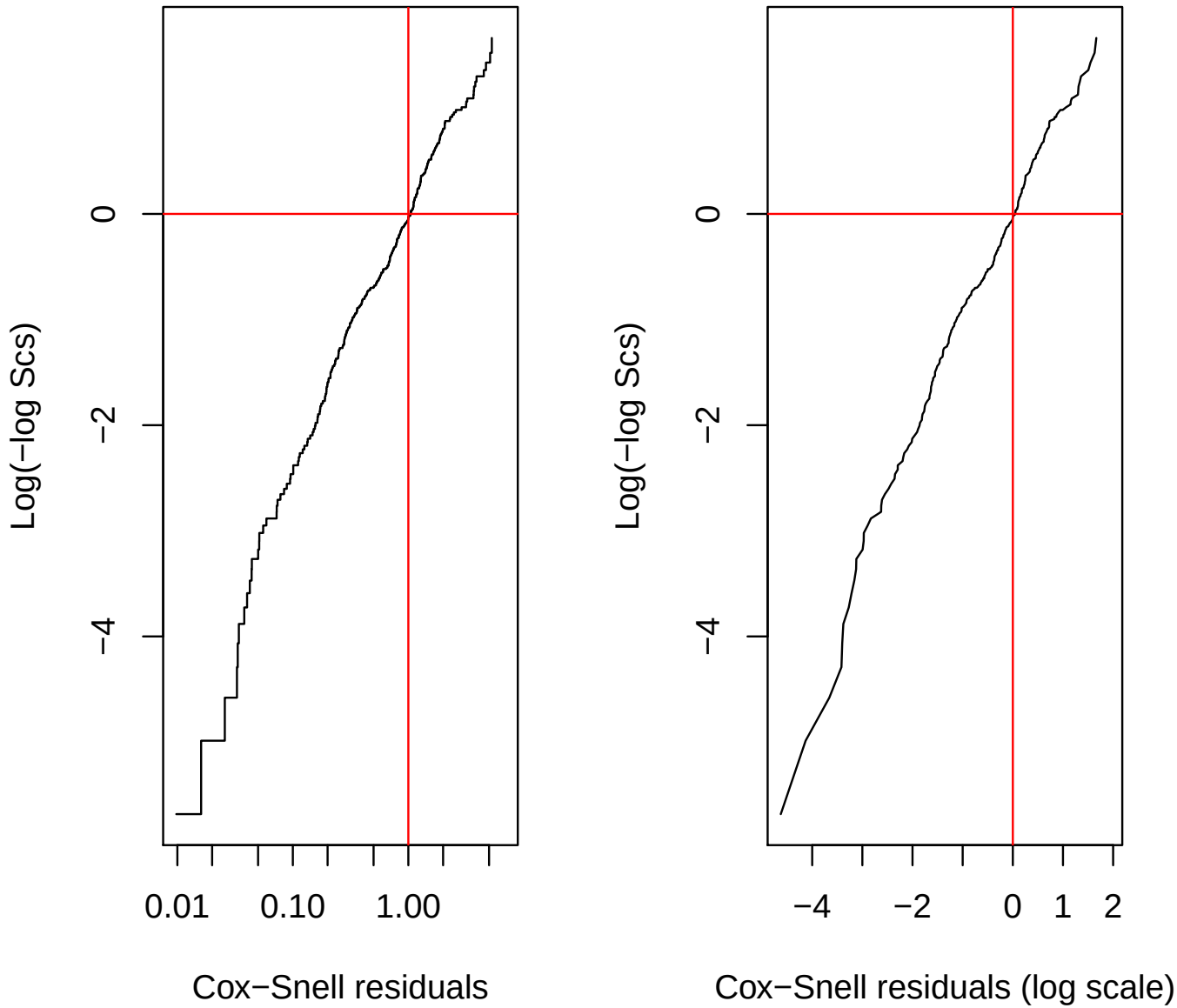
- Assess the fit

```
fitcs = survfit(Surv(csres,censor) ~ 1,data = halibut)
```

- Plot the results

```
plot(fitcs,fun = "cloglog",conf.int = F,mark.time = F,
     xlab = "Cox-Snell residuals (log scale)",ylab = "Log(-log Scs)")
abline(h = 0,col = "red")
abline(v = 1,col = "red") # Due to log scale!
```

Cox-Snell residuals in the halibut data example



Notes

The fit from the Cox-Snell residuals is fairly good, which suggests that the model fits adequately.

Nevertheless, Allison states that the "Cox-Snell residuals... are not very informative for Cox models estimated by partial likelihood."

Martingale Residuals

(See Fleming and Harrington, p.164)

Martingale residuals are defined for the i-th individual as:

$$r_i = \delta_i - \hat{\Lambda}(T_i)$$

Properties of the Martingale residuals

The Martingale residuals have the following properties:

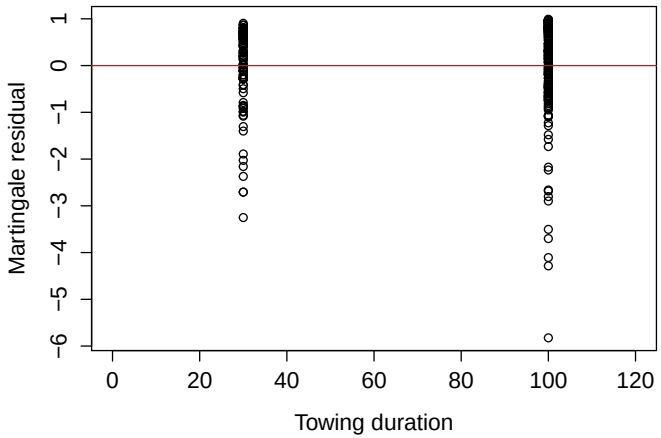
- r_i's have mean 0
- range of r_i's is between $-\infty$ and 1
- approximately uncorrelated (in samples)
- **Interpretation:** - the residual r_i can be viewed as the difference between the observed number of deaths (0 or 1) for subject i between time 0 and T_i, and the expected numbers based on the fitted model.

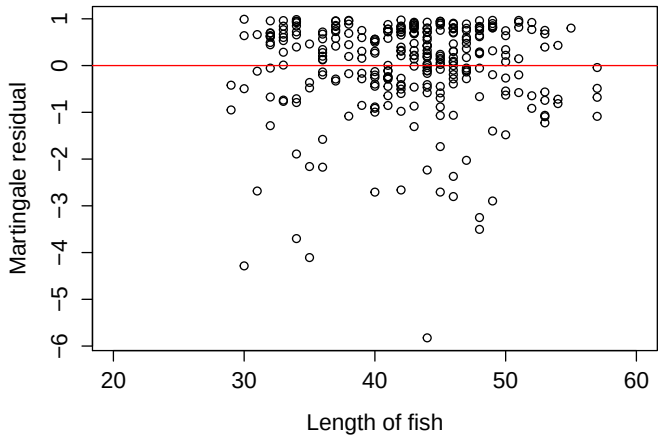
Obtaining martingale residuals with R

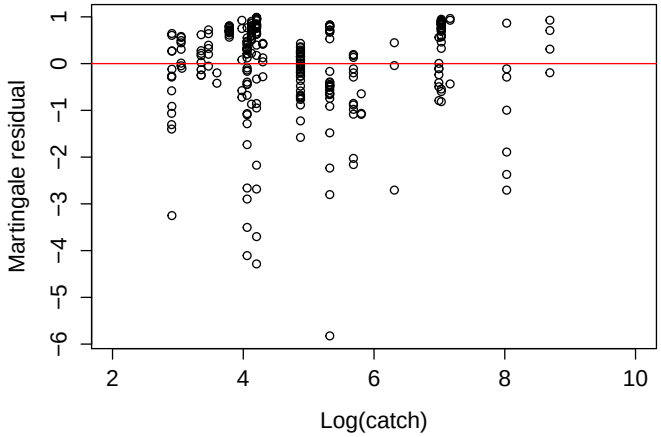
Martingale residuals are obtained through the `residuals` command (using `type="martingale"`).

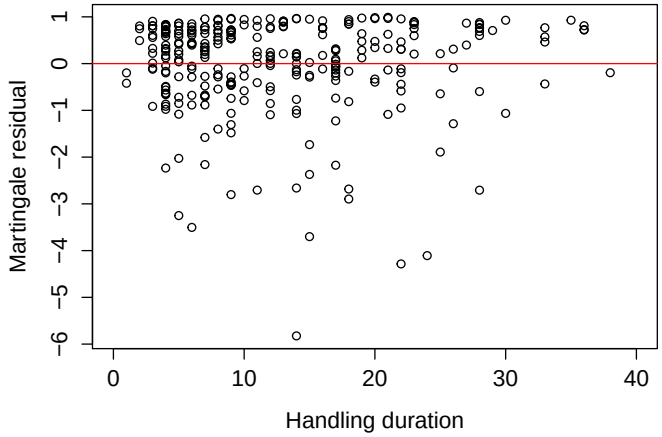
Once the martingale residual is created, you can plot it versus the predicted log HR (i.e., $\beta \mathbf{Z}_i$), or any of the individual covariates.

Martingale residuals in the halibut data example

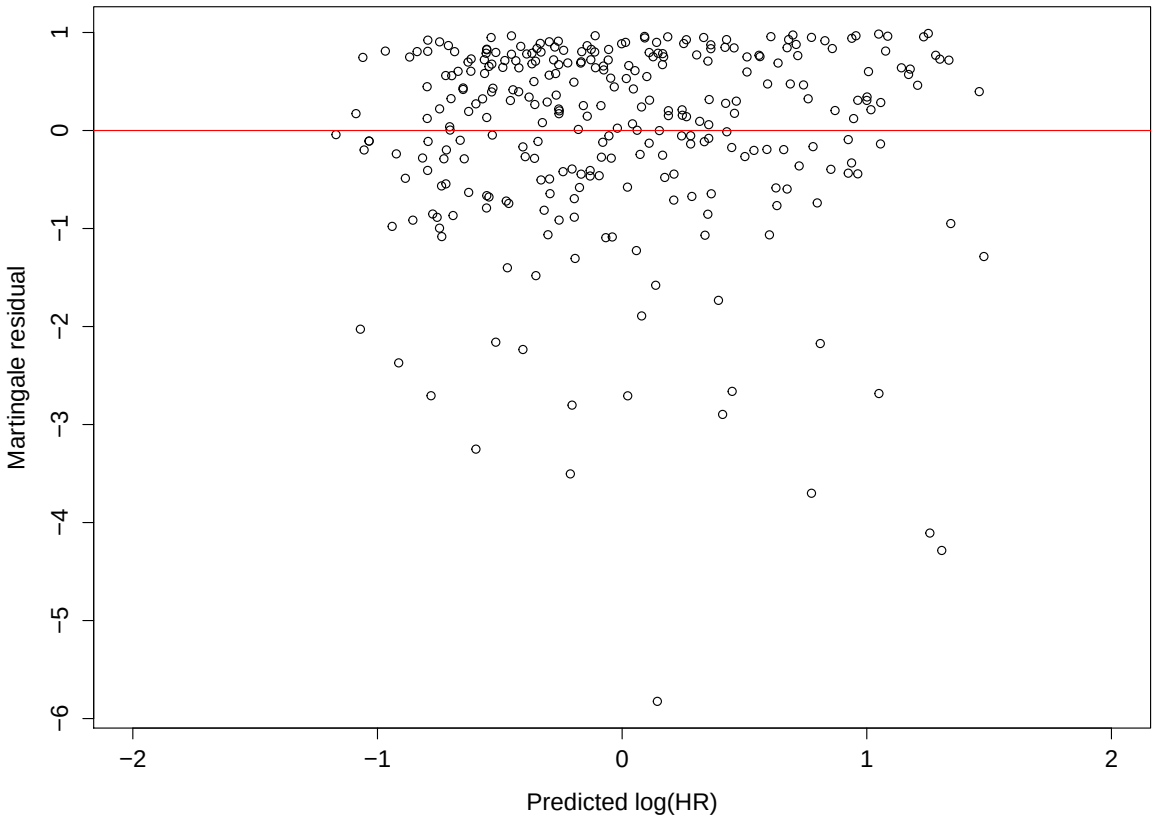








Martingale residuals against the predicted log hazard ratio



Deviance Residuals

One problem with the martingale residuals is that they tend to be asymmetric.

A solution is to use **deviance residuals**. For person i, these are defined as a function of the martingale residuals (r_i):

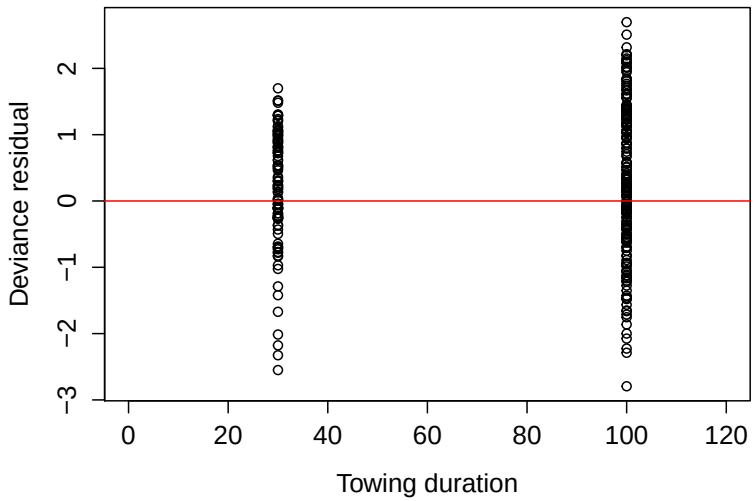
$$\hat{D}_i = \text{sign}(\hat{r}_i)\sqrt{-2[\hat{r}_i + \delta_i log(\delta_i - \hat{r}_i)]}$$

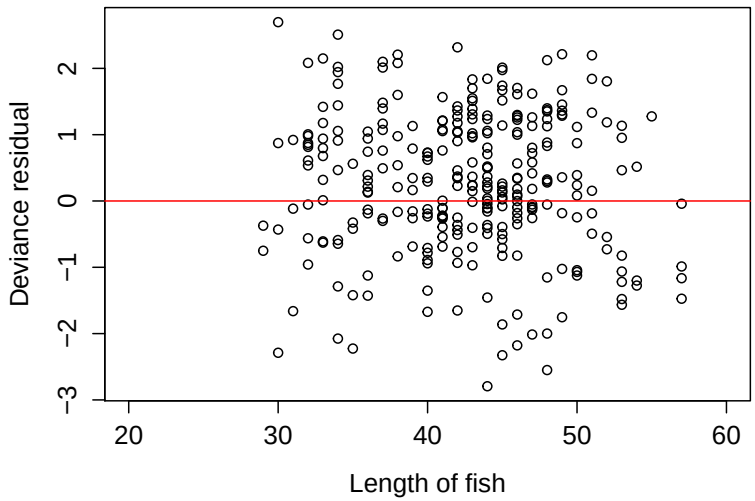
Obtaining deviance residuals in R

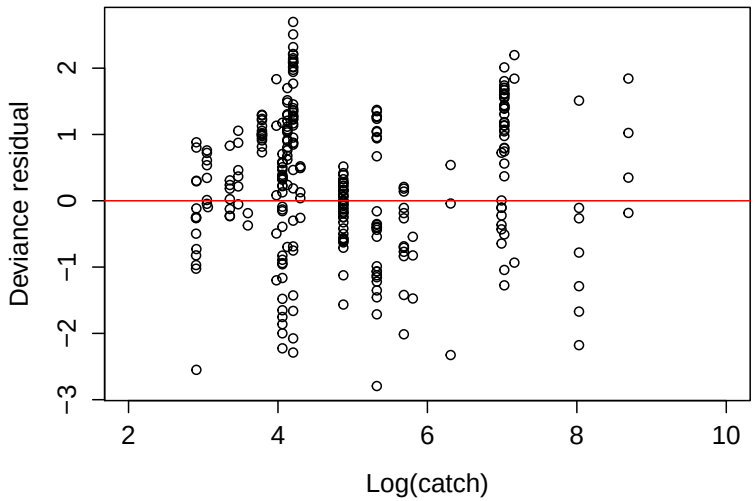
In R, the deviance residuals are generated using `residuals` command with the `type="deviance"` option:
The deviance residuals can then be plotted versus the predicted log(HR) or the individual covariates, as shown for the Martingale residuals.

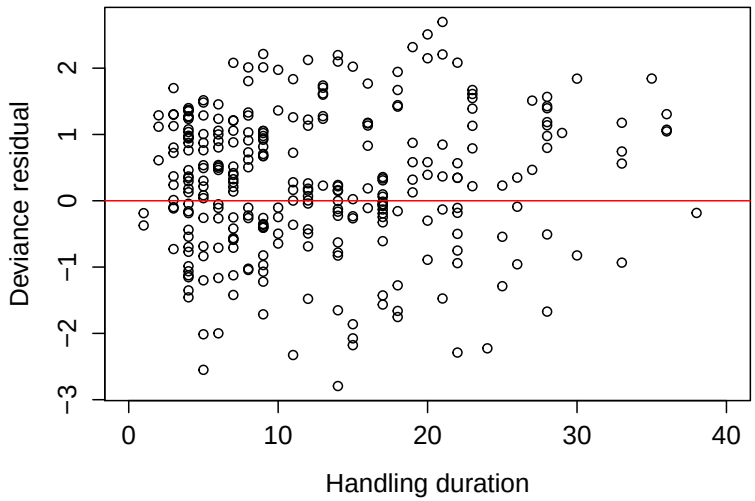
Deviance residuals behave much like residuals from OLS regression (i.e., mean=0, s.d.=1). They are negative for observations with survival times that are smaller than expected.

Deviance Residuals

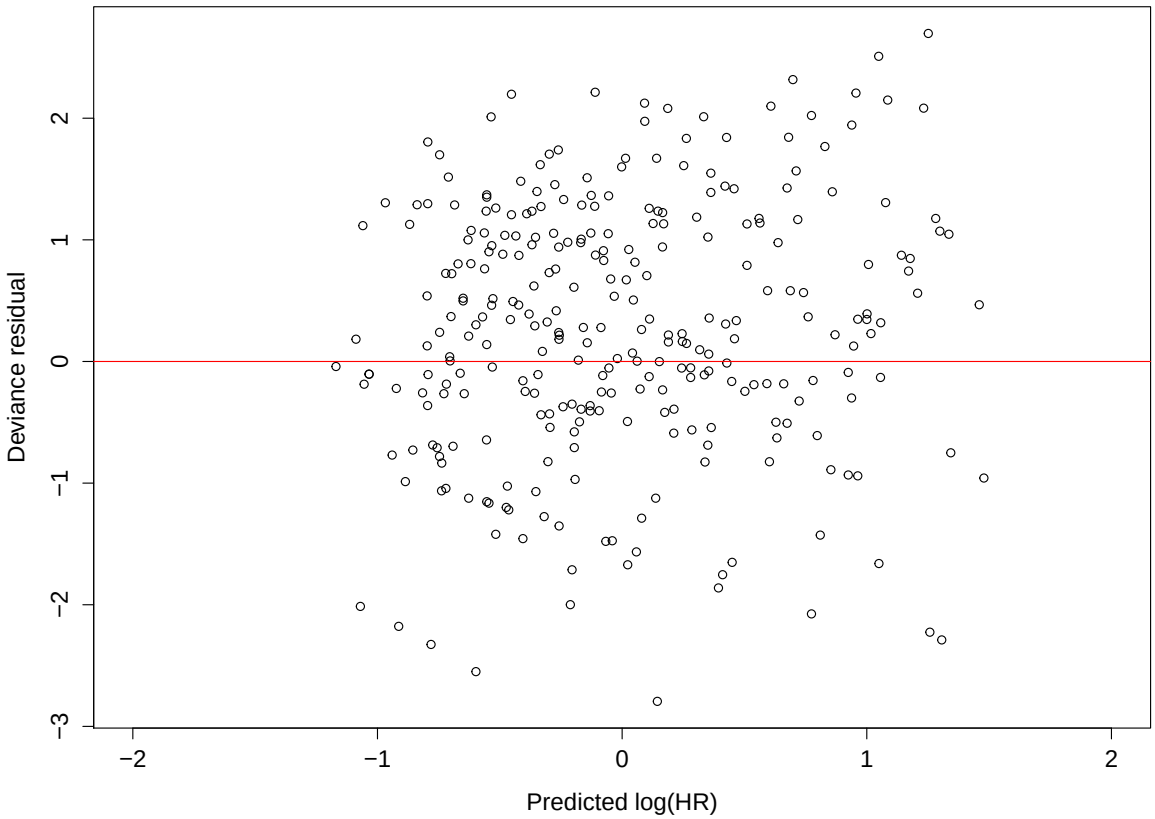








Deviance residuals versus predicted log hazard ratio



Schoenfeld residuals

These are defined at each observed failure time as:

$$r_{ij}^{s} = Z_{ij}(t_i) - \bar{Z}_j(t_i)$$

where

$$\bar{Z}_j(t_i) = \frac{\sum_{\ell \in \mathcal{R}(\tau_j)} Z_\ell e^{\beta Z_\ell}}{\sum_{\ell \in \mathcal{R}(\tau_j)} e^{\beta Z_\ell}}$$

is the average value of the covariates at each failure time.

Notes

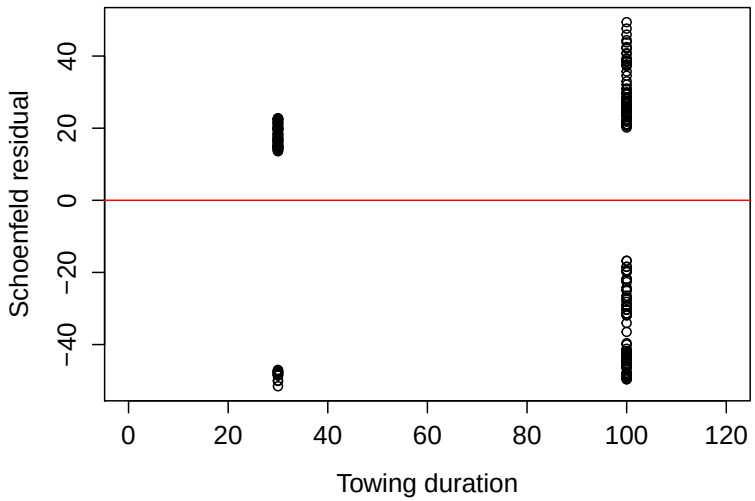
The following are some comments about the Schoenfeld residuals:

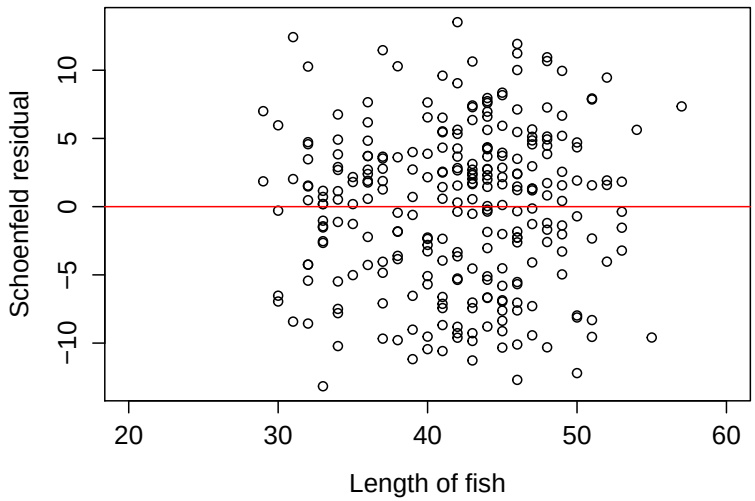
- represent the difference between the observed covariate and the average over the risk set at that time
- calculated for each covariate
- not defined for censored failure times.
- useful for assessing time trend or lack or proportionality, based on plotting versus event time
- sum to zero, have expected value zero, and are uncorrelated (in samples)

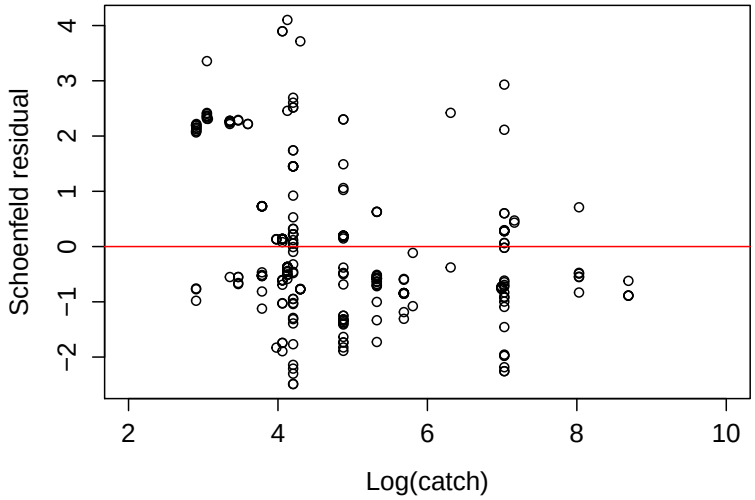
‘

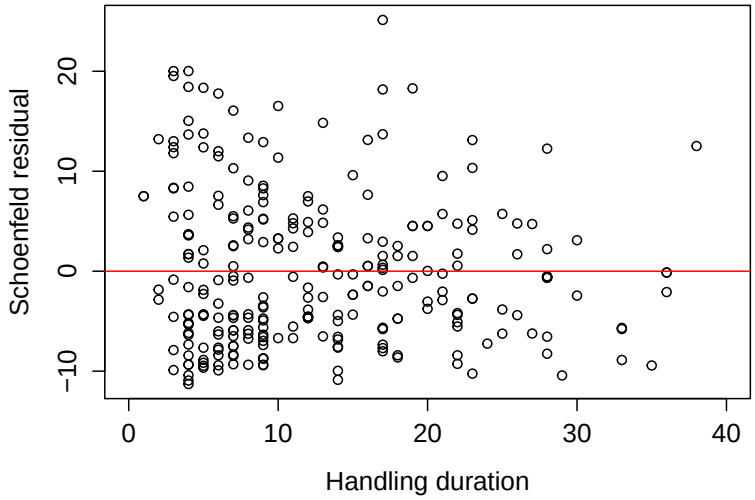
In R, the Schoenfeld residuals are generated in the `residuals` command itself, using the `type="schoenfeld"` option.

Schoenfeld residuals in the halibut data example









Weighted Schoenfeld Residuals

These are actually used more often than the previous unweighted version, because they are more like the typical OLS residuals (i.e., symmetric around 0).

They are defined as:

$$r_{ij}^{w} \quad = \quad n\widehat{V}\ r_{ij}^{s}$$

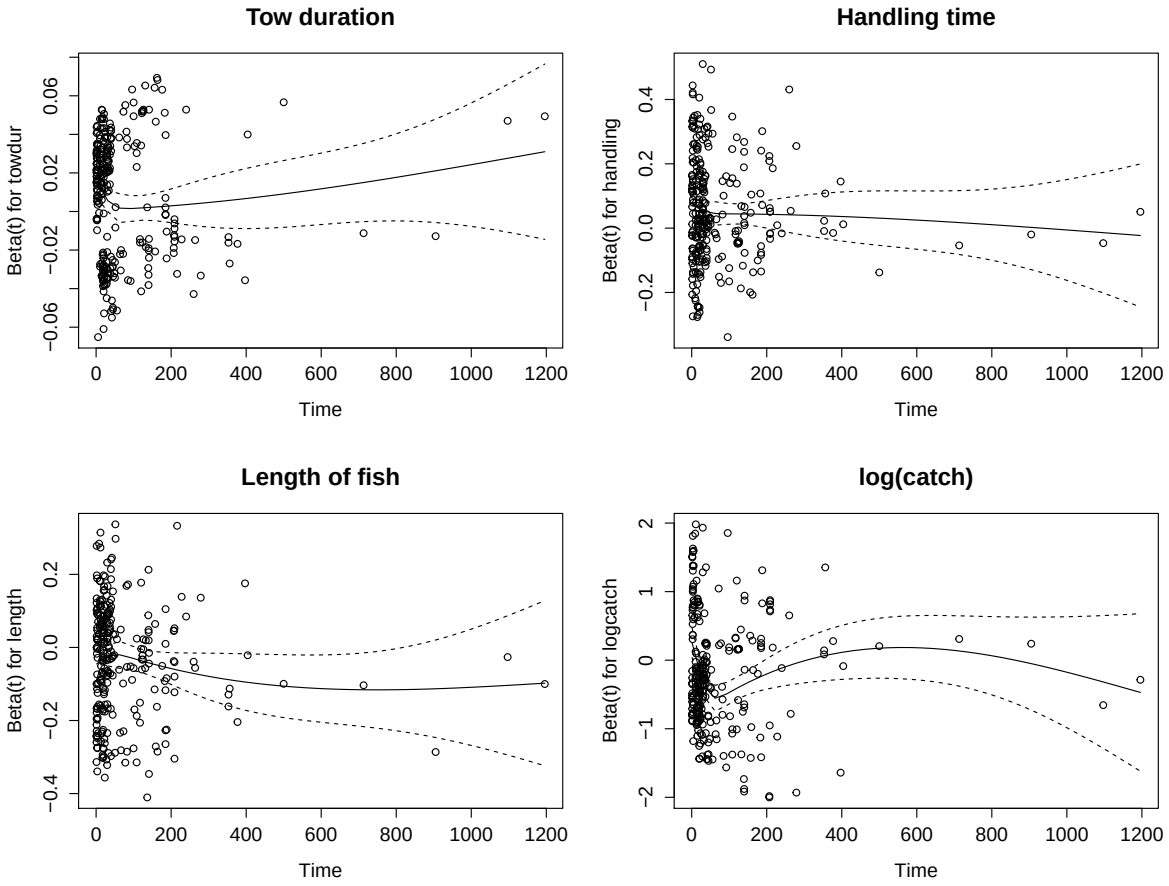
where $\widehat{V}$ is the estimated variance of $\hat{\boldsymbol{\beta}}$. The weighted residuals can be used in the same way as the unweighted ones to assess time trends and lack of proportionality.

Obtaining the weighted Schoenfeld residuals from R

In R, use the command `cox.zph` with the option.

```
cox.zph(fit,transform = "identity")
```

This results in the following plots:



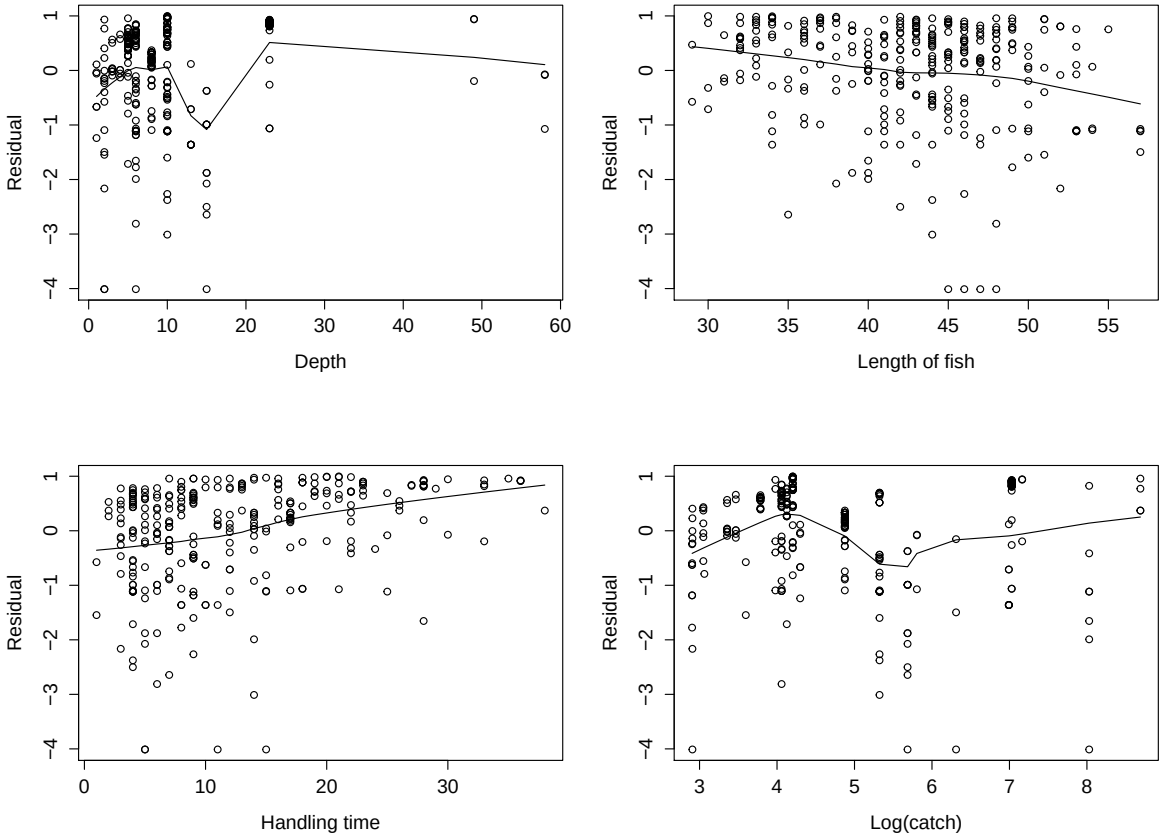
Using Residual plots to explore relationships

If you calculate martingale or deviance residuals without any covariates in the model and then plot against covariates, you obtain a graphical impression of the relationship between the covariate and the hazard.

In R, it is easy to do this (also possible in stata using the "estimate" option)

Residual plots

We plot the martingale residuals versus each of the covariates. This produces the following plots:



Can we improve the model?

The plots appear to have some structure, which indicate that we could be leaving something out. It is always a good idea to check for interactions.

In this case, there are several important interactions. I used a backward selection model forcing all main effects to be included, and considering all pairwise interactions.

Backward elimination with pairwise interactions

The results from this analysis are as follows:

```
Call:
coxph(formula = Surv(survtime, censor) ~ towdur + depth + length +
    handling + logcatch + towdepth + lengthdepth + handlingdepth +
    towlength + towhandling, data = halibut)

  n= 294, number of events= 273

                   coef  exp(coef)   se(coef)      z Pr(>|z|)
towdur        -0.0756235  0.9271652  0.0174010 -4.346 1.39e-05 ***
depth          0.1249947  1.1331424  0.0639359  1.955 0.050583 .
length        -0.0775731  0.9253594  0.0255046 -3.042 0.002354 **
handling       0.0045787  1.0045892  0.0321858  0.142 0.886876
logcatch      -0.2241951  0.7991592  0.0715613 -3.133 0.001731 **
towdepth       0.0029236  1.0029279  0.0004994  5.854 4.79e-09 ***
lengthdepth   -0.0060456  0.9939727  0.0013568 -4.456 8.36e-06 ***
handlingdepth -0.0041262  0.9958823  0.0011834 -3.487 0.000489 ***
towlength      0.0011826  1.0011833  0.0003540  3.340 0.000836 ***
towhandling    0.0011146  1.0011152  0.0003556  3.134 0.001723 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Interpretation: Handling alone doesn't seem to affect survival, unless it is combined with a longer towing duration or shallower trawling depths.