Cox Proportional-Hazards Regression for Survival Data in R

An Appendix to *An R Companion to Applied Regression*, third edition

John Fox & Sanford Weisberg

last revision: 2018-09-28

Abstract

Survival analysis examines and models the time it takes for events to occur, termed *survival time*. The *Cox proportional-hazards regression model* is the most common tool for studying the dependency of survival time on predictor variables. This appendix to Fox and Weisberg (2019) briefly describes the basis for the Cox regression model, and explains how to use the **survival** package in R to estimate Cox regressions.

1 Introduction

Survival analysis examines and models the time it takes for events to occur. The prototypical such event is death, from which the name "survival analysis" and much of its terminology derives, but the ambit of application of survival analysis is much broader. Essentially the same methods are employed in a variety of disciplines under various rubrics—for example, "event-history analysis" in sociology and "failure-time analysis" in engineering. In this appendix, therefore, terms such as *survival* are to be understood generically.

Survival analysis focuses on the distribution of survival times. Although there are well known methods for estimating unconditional survival distributions, most interesting survival modeling examines the relationship between survival and one or more predictors, usually termed *covariates* in the survival-analysis literature. The subject of this appendix is the Cox proportional-hazards regression model introduced in a seminal paper by Cox, 1972, a broadly applicable and the most widely used method of survival analysis. The **survival** package in R (Therneau, 1999; Therneau and Grambsch, 2000) fits Cox models, as we describe here, and most other commonly used survival methods.[1]

As is the case for the other on-line appendices to *An R Companion to Applied Regression*, we assume that you have read the *R Companion* and are therefore familiar with R.[2] In addition, we assume familiarity with Cox regression. We nevertheless begin with a review of basic concepts, primarily to establish terminology and notation. The second section of the appendix takes up the Cox proportional-hazards model with time-independent covariates. Time-dependent covariates are introduced in the third section. A fourth and final section deals with diagnostics.

[1]The **survival** package is one of the "recommended" packages that are included in the standard R distribution. The package must be loaded via the command `library("survival")`.

[2]Most R functions used but not described in this appendix are discussed in Fox and Weisberg (2019). All the R code used in this appendix can be downloaded from `http://tinyurl.com/carbook` or via the `carWeb()` function in the **car** package.

2 Basic Concepts and Notation

Let T represent survival time. We regard T as a random variable with cumulative distribution function $P(t) = \Pr(T \leq t)$ and probability density function $p(t) = dP(t)/dt$.[3] The more optimistic *survival function* $S(t)$ is the complement of the distribution function, $S(t) = \Pr(T > t) = 1 - P(t)$. A fourth representation of the distribution of survival times is the *hazard function*, which assesses the instantaneous risk of demise at time t, conditional on survival to that time:

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{\Pr\left[(t \leq T < t + \Delta t) | T \geq t\right]}{\Delta t} \\ &= \frac{f(t)}{S(t)} \end{aligned}$$

Models for survival data usually employ the hazard function or the log hazard. For example, assuming a constant hazard, $h(t) = \nu$, implies an exponential distribution of survival times, with density function $p(t) = \nu e^{-\nu t}$. Other common hazard models include

$$\log h(t) = \nu + \rho t$$

which leads to the *Gompertz distribution* of survival times, and

$$\log h(t) = \nu + \rho \log(t)$$

which leads to the *Weibull distribution* of survival times. (See, for example, Cox and Oakes, 1984, Sec. 2.3, for these and other possibilities.) In both the Gompertz and Weibull distributions, the hazard can either increase or decrease with time; moreover, in both instances, setting $\rho = 0$ yields the exponential model.

A nearly universal feature of survival data is *censoring*, the most common form of which is *right-censoring*: Here, the period of observation expires, or an individual is removed from the study, before the event occurs—for example, some individuals may still be alive at the end of a clinical trial, or may drop out of the study for various reasons other than death prior to its termination. A case is *left-censored* if its initial time at risk is unknown. Indeed, the same case may be both right and left-censored, a circumstance termed *interval-censoring*. Censoring complicates the likelihood function, and hence the estimation, of survival models.

Moreover, conditional on the value of any covariates in a survival model and on an individual's survival to a particular time, censoring must be independent of the future value of the hazard for the individual. If this condition is not met, then estimates of the survival distribution can be seriously biased. For example, if individuals tend to drop out of a clinical trial shortly before they die, and therefore their deaths go unobserved, survival time will be over-estimated. Censoring that meets this requirement is *noninformative*. A common instance of noninformative censoring occurs when a study terminates at a predetermined date.

3 The Cox Proportional-Hazards Model

Survival analysis typically examines the relationship of the survival distribution to covariates. Most commonly, this examination entails the specification of a linear-like model for the log hazard. For example, a parametric model based on the exponential distribution may be written as

$$\log h_i(t) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

[3] If you're unfamiliar with calculus, the essence of the matter here is that areas under the density function $p(t)$ represent probabilities of death in a given time interval, while the distribution function $P(t)$ represents the probability of death by time t.

or, equivalently,

$$h_i(t) = \exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik})$$

that is, as a linear model for the log-hazard or as a multiplicative model for the hazard. Here, i is a subscript for case, and the xs are the covariates. The constant α in this model represents a kind of baseline log-hazard, because $\log h_i(t) = \alpha$, or $h_i(t) = e^{\alpha}$, when all of the xs are zero. There are similar parametric regression models based on the other survival distributions described in the preceding section.[4]

The *Cox model*, in contrast, leaves the baseline hazard function $\alpha(t) = \log h_0(t)$ unspecified:

$$\log h_i(t) = \alpha(t) + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

or, again equivalently,

$$h_i(t) = h_0(t) \exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik})$$

This model is *semi-parametric* because while the baseline hazard can take any form, the covariates enter the model linearly. Consider, now, two cases i and i' that differ in their x-values, with the corresponding linear predictors

$$\eta_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik}$$

and

$$\eta_{i'} = \beta_1 x_{i'1} + \beta_2 x_{i'2} + \cdots + \beta_k x_{i'k}$$

The hazard ratio for these two cases,

$$\begin{aligned}\frac{h_i(t)}{h_{i'}(t)} &= \frac{h_0(t)e^{\eta_i}}{h_0(t)e^{\eta_{i'}}} \\ &= \frac{e^{\eta_i}}{e^{\eta_{i'}}}\end{aligned}$$

is independent of time t. Consequently, the Cox model is a *proportional-hazards model.*

Remarkably, even though the baseline hazard is unspecified, the Cox model can still be estimated by the *method of partial likelihood*, developed by Cox (1972) in the same paper in which he introduced what came to called the Cox model. Although the resulting estimates are not as efficient as maximum-likelihood estimates for a correctly specified parametric hazard regression model, not having to make arbitrary, and possibly incorrect, assumptions about the form of the baseline hazard is a compensating virtue of Cox's specification. Having fit the model, it is possible to extract an estimate of the baseline hazard (see below).

3.1 The `coxph()` Function

The Cox proportional-hazards regression model is fit in R with the `coxph()` function, located in the **survival** package:

```
library("survival")
args(coxph)

function (formula, data, weights, subset, na.action, init, control,
    ties = c("efron", "breslow", "exact"), singular.ok = TRUE,
    robust = FALSE, model = FALSE, x = FALSE, y = TRUE, tt, method = ties,
    ...)
NULL
```

[4]The `survreg()` function in the **survival** package fits the exponential model and other parametric *accelerated failure time* models. Because the Cox model is now used much more frequently than parametric survival regression models, we will not describe `survreg()` in this appendix. Enter `?survreg` and see Therneau (1999) for details.

Most of the arguments to `coxph()`, including `data`, `weights`, `subset`, `na.action`, `singular.ok`, `model`, `x` and `y`, are familiar from `lm()` (see Chapter 4 of the *Companion*, especially Section 4.9). The `formula` argument is a little different. The right-hand side of the formula for `coxph()` is the same as for a linear model.[5] The left-hand side is a survival object, created by the `Surv()` function. In the simple case of right-censored data, the call to `Surv()` takes the form `Surv(`*`time, event`*`)`, where *`time`* is either the event time or the censoring time, and *`event`* is a dummy variable coded `1` if the event is observed or `0` if the case is censored. See Section 4 below and, more generally, the on-line help for `Surv()` for other possibilities.

Among the remaining arguments to `coxph()`:

- `init` (initial values) and `control` are technical arguments: See the on-line help for `coxph()` for details.
- `method` indicates how to handle cases that have tied (i.e., identical) survival times. The default `"efron"` method is generally preferred to the once-popular `"breslow"` method; the `"exact"` method is much more computationally intensive.
- If `robust` is `TRUE`, `coxph()` calculates robust coefficient-variance estimates. The default is `FALSE`, unless the model includes non-independent cases, specified by the `cluster()` function in the model formula. We do not describe Cox regression for clustered data in this appendix.

3.2 An Illustration: Recidivism

The `Rossi` data set in the **carData** package contains data from an experimental study of recidivism of 432 male prisoners, who were observed for a year after being released from prison (Rossi et al., 1980). The following variables are included in the data; the variable names are those used by Allison (1995), from whom this example and variable descriptions are adapted:

- `week`: week of first arrest after release, or censoring time.
- `arrest`: the event indicator, equal to `1` for those arrested during the period of the study and `0` for those who were not arrested.
- `fin`: a factor, with levels `"yes"` if the individual received financial aid after release from prison, and `"no"` if he did not; financial aid was a randomly assigned factor manipulated by the researchers.
- `age`: in years at the time of release.
- `race`: a factor with levels `"black"` and `"other"`.
- `wexp`: a factor with levels `"yes"` if the individual had full-time work experience prior to incarceration and `"no"` if he did not.
- `mar`: a factor with levels `"married"` if the individual was married at the time of release and `"not married"` if he was not.
- `paro`: a factor coded `"yes"` if the individual was released on parole and `"no"` if he was not.
- `prio`: number of prior convictions.

[5] There are, however, special functions `cluster()` and `strata()` that may be included on the right side of the model formula. The `cluster()` function is used to specify non-independent cases (such as several individuals in the same family), and the `strata()` function may be used to divide the data into sub-groups with potentially different baseline hazard functions, as explained in Section 5.1.

- `educ`: education, a categorical variable coded numerically, with codes `2` (grade 6 or less), `3` (grades 6 through 9), `4` (grades 10 and 11), `5` (grade 12), or `6` (some post-secondary).[6]
- `emp1`–`emp52`: factors coded `"yes"` if the individual was employed in the corresponding week of the study and `"no"` otherwise.

We read the data file into a data frame, and print the first few cases (omitting the variables `emp1` – `emp52`, which are in columns 11–62 of the data frame):

```
library("carData")
Rossi[1:5, 1:10]

  week arrest fin age  race wexp         mar paro prio educ
1   20      1  no  27 black   no not married  yes    3    3
2   17      1  no  18 black   no not married  yes    8    4
3   25      1  no  19 other  yes not married  yes   13    3
4   52      0 yes  23 black  yes     married  yes    1    5
5   52      0  no  19 other  yes not married  yes    3    3
```

Thus, for example, the first individual was arrested in week 20 of the study, while the fourth individual was never rearrested, and hence has a censoring time of 52.

Following Allison, a Cox regression of time to rearrest on the time-constant covariates is specified as follows:

```
mod.allison <- coxph(Surv(week, arrest) ~
    fin + age + race + wexp + mar + paro + prio,
    data=Rossi)
mod.allison

Call:
coxph(formula = Surv(week, arrest) ~ fin + age + race + wexp +
    mar + paro + prio, data = Rossi)

                  coef exp(coef) se(coef)     z      p
finyes         -0.3794    0.6843   0.1914 -1.98 0.0474
age            -0.0574    0.9442   0.0220 -2.61 0.0090
raceother      -0.3139    0.7306   0.3080 -1.02 0.3081
wexpyes        -0.1498    0.8609   0.2122 -0.71 0.4803
marnot married  0.4337    1.5430   0.3819  1.14 0.2561
paroyes        -0.0849    0.9186   0.1958 -0.43 0.6646
prio            0.0915    1.0958   0.0286  3.19 0.0014

Likelihood ratio test=33.3  on 7 df, p=2e-05
n= 432, number of events= 114
```

The `summary()` method for Cox models produces a more complete report:

```
summary(mod.allison)

Call:
coxph(formula = Surv(week, arrest) ~ fin + age + race + wexp +
```

[6]Following Allison (2010), `educ` is not used in the examples reported below. We reinvite the reader to redo our examples adding `educ` as a predictor.

```
    mar + paro + prio, data = Rossi)

  n= 432, number of events= 114

                  coef exp(coef) se(coef)     z Pr(>|z|)
finyes         -0.3794    0.6843   0.1914 -1.98   0.0474
age            -0.0574    0.9442   0.0220 -2.61   0.0090
raceother      -0.3139    0.7306   0.3080 -1.02   0.3081
wexpyes        -0.1498    0.8609   0.2122 -0.71   0.4803
marnot married  0.4337    1.5430   0.3819  1.14   0.2561
paroyes        -0.0849    0.9186   0.1958 -0.43   0.6646
prio            0.0915    1.0958   0.0286  3.19   0.0014

               exp(coef) exp(-coef) lower .95 upper .95
finyes             0.684      1.461     0.470     0.996
age                0.944      1.059     0.904     0.986
raceother          0.731      1.369     0.399     1.336
wexpyes            0.861      1.162     0.568     1.305
marnot married     1.543      0.648     0.730     3.261
paroyes            0.919      1.089     0.626     1.348
prio               1.096      0.913     1.036     1.159

Concordance= 0.64  (se = 0.027 )
Rsquare= 0.074   (max possible= 0.956 )
Likelihood ratio test= 33.3  on 7 df,   p=2e-05
Wald test            = 32.1  on 7 df,   p=4e-05
Score (logrank) test = 33.5  on 7 df,   p=2e-05
```

- The column marked `z` in the output records the ratio of each regression coefficient to its standard error, a Wald statistic which is asymptotically standard normal under the hypothesis that the corresponding β is zero. The coefficients for the covariates `age` and `prio` (prior convictions) have very small p-values, while the coefficient for `fin` (financial aid—the focus of the study) has a p-value only slightly less than 0.05.

- The exponentiated coefficients in the second column of the first panel (and in the first column of the second panel) of the output are interpretable as multiplicative effects on the hazard. Thus, for example, holding the other covariates constant, an additional year of `age` reduces the weekly hazard of rearrest by a factor of $e^{b_2} = 0.944$ on average—that is, by 5.6 percent. Similarly, each prior conviction increases the hazard by a factor of 1.096, or 9.6 percent.

- The likelihood-ratio, Wald, and score chi-square statistics at the bottom of the output are asymptotically equivalent tests of the omnibus null hypothesis that all of the βs are zero. In this instance, the test statistics are in close agreement, and the omnibus null hypothesis is soundly rejected.

The `Anova()` function in the **car** package has a method for `"coxph"` objects, by default computing Type-II likelihood-ratio tests for the terms in the model:

```
library("car")
Anova(mod.allison)

Analysis of Deviance Table (Type II tests)
     LR Chisq Df Pr(>Chisq)
```

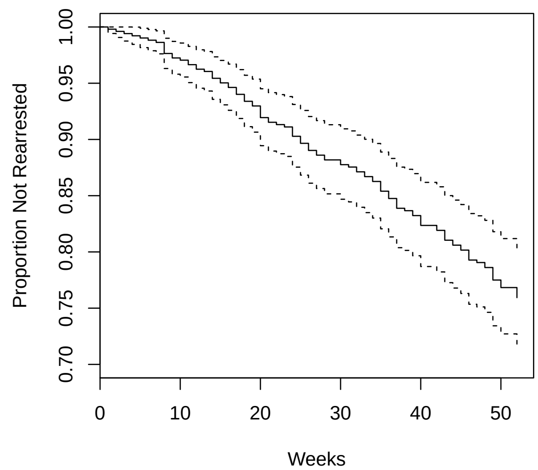

Figure 1: Estimated survival function $\widehat{S}(t)$ for the Cox regression of time to rearrest on several predictors. The broken lines show a point-wise 95-percent confidence envelope around the survival function.

```
fin      3.99  1      0.0459
age      7.99  1      0.0047
race     1.13  1      0.2888
wexp     0.50  1      0.4794
mar      1.43  1      0.2316
paro     0.19  1      0.6655
prio     8.98  1      0.0027
```

Having fit a Cox model to the data, it is often of interest to examine the estimated distribution of survival times. The `survfit()` function estimates $S(t)$, by default at the mean values of the covariates. The `plot()` method for objects returned by `survfit()` graphs the estimated surivival function, along with a point-wise 95-percent confidence band. For example, for the model just fit to the recidivism data:

```
plot(survfit(mod.allison), ylim=c(0.7, 1), xlab="Weeks",
    ylab="Proportion Not Rearrested")
```

This command produces Figure 1. The limits for the vertical axis, set by `ylim=c(0.7, 1)`, were selected after examining an initial plot.

Even more cogently, we may wish to display how estimated survival depends upon the value of a covariate. Because the principal purpose of the recidivism study was to assess the impact of financial aid on rearrest, we focus on this covariate. We construct a new data frame with two rows, one for each value of `fin`; the other covariates are fixed to their average values. For a dummy covariate, such as the contrast associated with `race`, the average value is the proportion coded 1 in the data set—in the case of `race`, the proportion of non-blacks (cf., the discussion of predictors effect displays in Section 4.3 of the *R Companion.*). This data frame is passed to `survfit()` via the `newdata` argument:

```
Rossi.fin <- with(Rossi, data.frame(fin=c(0, 1),
    age=rep(mean(age), 2), race=rep(mean(race == "other"), 2),
```

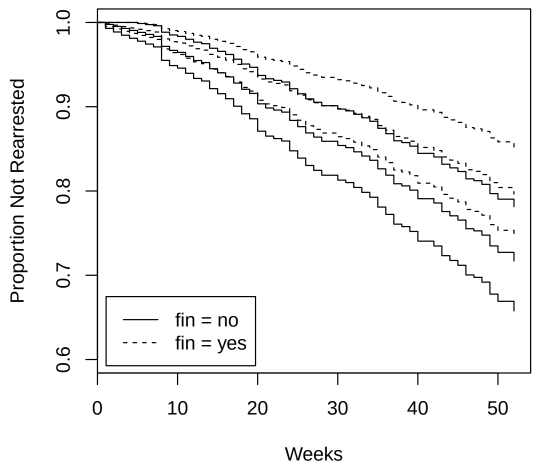

Figure 2: Estimated survival functions for those receiving (`fin` = `yes`) and not receiving (`fin` = `no`) financial aid. Other covariates are fixed at their average values. Each estimate is accompanied by a point-wise 95-percent confidence envelope.

```
     wexp=rep(mean(wexp == "yes"), 2), mar=rep(mean(mar == "not married"), 2),
     paro=rep(mean(paro == "yes"), 2), prio=rep(mean(prio), 2)))
plot(survfit(mod.allison, newdata=Rossi.fin), conf.int=TRUE,
    lty=c(1, 2), ylim=c(0.6, 1), xlab="Weeks",
    ylab="Proportion Not Rearrested")
legend("bottomleft", legend=c("fin = no", "fin = yes"), lty=c(1 ,2), inset=0.02)

Warning messages:
1: In model.frame.default(Terms2, newdata, xlev = object$xlevels) :
  variable 'fin' is not a factor
2: In model.frame.default(Terms2, newdata, xlev = object$xlevels) :
  variable 'race' is not a factor
3: In model.frame.default(Terms2, newdata, xlev = object$xlevels) :
  variable 'wexp' is not a factor
4: In model.frame.default(Terms2, newdata, xlev = object$xlevels) :
  variable 'mar' is not a factor
5: In model.frame.default(Terms2, newdata, xlev = object$xlevels) :
  variable 'paro' is not a factor
```

The `survfit()` command generates warnings because we supplied numerical values for factors (e.g., the proportion of non-blacks for the factor `race`), but the computation is performed correctly. We specified two additional arguments to `plot()`: `lty=c(1, 2)` indicates that the survival function for the first group (i.e., for `fin` = `no`) will be plotted with a solid line, while that for the second group (`fin` = `yes`) will be plotted with a broken line; `conf.int=TRUE` requests that confidence envelopes be drawn around each estimated survival function (which is not the default when more than one survival function is plotted). We used the `legend()` function to place a legend on the plot.[7] The

[7]The `plot()` method for `"survfit"` objects can also draw a legend on the plot, but separate use of the `legend()`

resulting graph, which appears in Figure 2, shows the higher estimated "survival" of those receiving financial aid, but the two confidence envelopes overlap substantially, even after 52 weeks.

4 Time-Dependent Covariates

The `coxph()` function handles time-dependent covariates by requiring that each time period for an individual appear as a separate "case"—that is, as a separate row (or *record*) in the data set. Consider, for example, the `Rossi` data frame, and imagine that we want to treat weekly employment as a time-dependent predictor of time to rearrest. As is often the case, however, the data for each individual appears as a single row, with the weekly employment indicators as 52 columns in the data frame, with names `emp1` through `emp52`; for example, for the first person in the study:

```
Rossi[1, ]
```

```
  week arrest fin age  race wexp         mar paro prio educ emp1 emp2 emp3 emp4 emp5
1   20      1  no  27 black   no not married  yes    3    3   no   no   no   no   no
  emp6 emp7 emp8 emp9 emp10 emp11 emp12 emp13 emp14 emp15 emp16 emp17 emp18 emp19
1   no   no   no   no    no    no    no    no    no    no    no    no    no    no
  emp20 emp21 emp22 emp23 emp24 emp25 emp26 emp27 emp28 emp29 emp30 emp31 emp32
1    no  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
  emp33 emp34 emp35 emp36 emp37 emp38 emp39 emp40 emp41 emp42 emp43 emp44 emp45
1  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
  emp46 emp47 emp48 emp49 emp50 emp51 emp52
1  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>  <NA>
```

The employment indicators are missing after week 20, when individual 1 was rearrested.

To put the data in the requisite form, we need one row for each non-missing period of observation. To perform this task, we have written a function named `unfold()`; the function is included with the script file for this appendix, and takes the following arguments:[8]

- `data`: A data frame to be "unfolded" from "wide" to "long" format.
- `time`: The column number or quoted name of the event/censoring-time variable in `data`.
- `event`: The quoted name of the event/censoring indicator variable in `data`.
- `cov`: A vector giving the column numbers of the time-dependent covariate in `data`, or a list of vectors if there is more than one time-dependent covariate.
- `cov.names`: A character string or character vector giving the name(s) to be assigned to the time-dependent covariate(s) in the output data set.
- `suffix`: The suffix to be attached to the name of the time-to-event variable in the output data set; defaults to `".time"`.
- `cov.times`: The observation times for the covariate values, including the start time. This argument can take several forms:

function provides greater flexibility. Legends, line types, and other aspects of constructing graphs in R are described in Chapter 9 of the *R Companion*.

[8]This is a slightly simplified version of the `unfold()` function in the **RcmdrPlugin.survival** package, which adds survival-analysis capabilities to the R Commander graphical user interface to R (see Fox, 2017). The `Rossi` data set is also included in the **RcmdrPlugin.survival** package. Our ability to deal with a time-dependent covariate in this relatively simple manner depends on the regular structure of the `Rossi` data; for a more general treatment of time-dependent covariates for `coxph()` models, see the vignette "Using Time Dependent Covariates" in the **survival** package, which may be accessed by the command `vignette("timedep", package="survival")`.

  - The default is the vector of integers from zero to the number of covariate values (i.e., containing one more entry—the initial time of observation—than the length of each vector in `cov`).
  - An arbitrary numerical vector with one more entry than the length of each vector in `cov`.
  - The columns in the input data set that give the (potentially different) covariate observation times for each individual. There should be one more column than the length of each vector in `cov`.

- `common.times`: A logical value indicating whether the times of observation are the same for all individuals; defaults to `TRUE`.
- `lag`: Number of observation periods to lag each value of the time-dependent covariate(s); defaults to `0`. The use of `lag` is described later in this section.

Thus, to unfold the `Rossi` data, we enter:

```
Rossi.2 <- unfold(Rossi, time="week",
    event="arrest", cov=11:62, cov.names="employed")
```

```
> Rossi.2[1:50, ]
```

```
     start stop arrest.time week arrest fin age  race wexp         mar paro prio educ employed
1.1      0    1           0   20      1  no  27 black   no not married  yes    3    3       no
1.2      1    2           0   20      1  no  27 black   no not married  yes    3    3       no
. . .
1.19    18   19           0   20      1  no  27 black   no not married  yes    3    3       no
1.20    19   20           1   20      1  no  27 black   no not married  yes    3    3       no
2.1      0    1           0   17      1  no  18 black   no not married  yes    8    4       no
2.2      1    2           0   17      1  no  18 black   no not married  yes    8    4       no
. . .
2.16    15   16           0   17      1  no  18 black   no not married  yes    8    4       no
2.17    16   17           1   17      1  no  18 black   no not married  yes    8    4       no
3.1      0    1           0   25      1  no  19 other  yes not married  yes   13    3       no
3.2      1    2           0   25      1  no  19 other  yes not married  yes   13    3       no
. . .
3.13    12   13           0   25      1  no  19 other  yes not married  yes   13    3       no
```

Once the data set is constructed, it is simple to use `coxph()` to fit a model with time-dependent covariates. The right-hand-side of the model is essentially the same as before, but both the start and end times of each interval are specified in the call to `Surv()`, in the form `Surv(`*`start`*`, `*`stop`*`, `*`event`*`)`. Here, *`event`* is the time-dependent version of the event indicator variable, equal to `1` only in the time-period during which the event occurs. For the example:

```
mod.allison.2 <- coxph(Surv(start, stop, arrest.time) ~
    fin + age + race + wexp + mar + paro + prio + employed,
    data=Rossi.2)
summary(mod.allison.2)

Call:
coxph(formula = Surv(start, stop, arrest.time) ~ fin + age +
    race + wexp + mar + paro + prio + employed, data = Rossi.2)

  n= 19809, number of events= 114

                  coef exp(coef) se(coef)     z Pr(>|z|)
finyes         -0.3567    0.7000   0.1911 -1.87   0.0620
```

```
age           -0.0463    0.9547   0.0217 -2.13   0.0330
raceother     -0.3387    0.7127   0.3096 -1.09   0.2740
wexpyes       -0.0256    0.9748   0.2114 -0.12   0.9038
marnot married  0.2937    1.3414   0.3830  0.77   0.4431
paroyes       -0.0642    0.9378   0.1947 -0.33   0.7416
prio           0.0851    1.0889   0.0290  2.94   0.0033
employedyes   -1.3283    0.2649   0.2507 -5.30  1.2e-07

               exp(coef) exp(-coef) lower .95 upper .95
finyes             0.700      1.429     0.481     1.018
age                0.955      1.047     0.915     0.996
raceother          0.713      1.403     0.388     1.308
wexpyes            0.975      1.026     0.644     1.475
marnot married     1.341      0.745     0.633     2.842
paroyes            0.938      1.066     0.640     1.374
prio               1.089      0.918     1.029     1.152
employedyes        0.265      3.775     0.162     0.433

Concordance= 0.708  (se = 0.027 )
Rsquare= 0.003   (max possible= 0.066 )
Likelihood ratio test= 68.7  on 8 df,   p=9e-12
Wald test            = 56.1  on 8 df,   p=3e-09
Score (logrank) test = 64.5  on 8 df,   p=6e-11
```

4.1 Lagged Covariates

The last analysis suggests that time-dependent employment covariate has an apparently large effect, as the hazard of rearrest is smaller by a factor of $e^{-1.3282} = 0.265$ (i.e., a decline of 73.5 percent) during a week in which the former inmate was employed. As Allison (2010) points out, however, the direction of causality here is ambiguous, because a person cannot work when he is in jail. One way of addressing this problem is to use instead a lagged value of employment, from the *previous* week for example. The `unfold()` function can easily provide lagged time-dependent covariates:

```
Rossi.3 <- unfold(Rossi, "week", "arrest", 11:62, "employed", lag=1)
mod.allison.3 <- coxph(Surv(start, stop, arrest.time) ~
    fin + age + race + wexp + mar + paro + prio + employed,
    data=Rossi.3)
summary(mod.allison.3)

Call:
coxph(formula = Surv(start, stop, arrest.time) ~ fin + age +
    race + wexp + mar + paro + prio + employed, data = Rossi.3)

  n= 19377, number of events= 113

                  coef exp(coef) se(coef)     z Pr(>|z|)
finyes        -0.3513    0.7038   0.1918 -1.83  0.06703
age           -0.0498    0.9514   0.0219 -2.27  0.02297
raceother     -0.3215    0.7251   0.3091 -1.04  0.29837
wexpyes       -0.0476    0.9535   0.2132 -0.22  0.82321
marnot married  0.3448    1.4116   0.3832  0.90  0.36831
paroyes       -0.0471    0.9540   0.1963 -0.24  0.81038
```

```
prio           0.0920    1.0964   0.0288  3.19  0.00140
employedyes   -0.7869    0.4553   0.2181 -3.61  0.00031

              exp(coef) exp(-coef) lower .95 upper .95
finyes            0.704      1.421     0.483     1.025
age               0.951      1.051     0.911     0.993
raceother         0.725      1.379     0.396     1.329
wexpyes           0.953      1.049     0.628     1.448
marnot married    1.412      0.708     0.666     2.992
paroyes           0.954      1.048     0.649     1.402
prio              1.096      0.912     1.036     1.160
employedyes       0.455      2.197     0.297     0.698

Concordance= 0.67  (se = 0.027 )
Rsquare= 0.002   (max possible= 0.067 )
Likelihood ratio test= 47.2  on 8 df,   p=1e-07
Wald test            = 43.4  on 8 df,   p=7e-07
Score (logrank) test = 46.4  on 8 df,   p=2e-07
```

The coefficient for the now-lagged employment indicator still has a small p-value, but the estimated effect of employment, though substantial, is much smaller than before: $e^{-0.7869} = 0.455$ (or a decrease of 54.5 percent).

5 Model Diagnostics

As is the case for a linear or generalized linear model (see Chapter 8 of the *R Companion*), it is desirable to determine whether a fitted Cox regression model adequately describes the data. We will briefly consider three kinds of diagnostics: for violation of the assumption of proportional hazards; for influential data; and for nonlinearity in the relationship between the log hazard and the covariates. All of these diagnostics use the `residuals()` method for `"coxph"` objects, which calculates several kinds of residuals, along with some quantities that are not normally thought of as residuals. Details are in Therneau (1999).

5.1 Checking Proportional Hazards

Tests and graphical diagnostics for proportional hazards may be based on the *scaled Schoenfeld residuals*; these can be obtained directly as `residuals(model, "scaledsch")`, where `model` is a `"coxph"` model object. The matrix returned by `residuals()` has one column for each covariate in the model. More conveniently, the `cox.zph()` function calculates tests of the proportional-hazards assumption for each covariate, by correlating the corresponding set of scaled Schoenfeld residuals with a suitable transformation of time [the default is based on the *Kaplan-Meier estimate* of the survival function, $K(t)$].

We will illustrate these tests with a scaled-down version of the Cox regression model fit to the recidivism data in Section 3.2, eliminating the covariates whose coefficients had large p-values:[9]

```
mod.allison.4 <- coxph(Surv(week, arrest) ~ fin + age + prio,
    data=Rossi)
mod.allison.4
```

[9]It is possible that a covariate whose coefficients has a large p-value when its effect is, in essence, *averaged* over time nevertheless has a important *interaction* with time, which manifests itself as nonproportional hazards. We leave it to the reader to check for this possibility using the model fit originally to the recidivism data.

```
Call:
coxph(formula = Surv(week, arrest) ~ fin + age + prio, data = Rossi)

          coef exp(coef) se(coef)     z       p
finyes -0.3470    0.7068   0.1902 -1.82 0.06820
age    -0.0671    0.9351   0.0209 -3.22 0.00129
prio    0.0969    1.1017   0.0273  3.56 0.00038

Likelihood ratio test=29.1  on 3 df, p=2e-06
n= 432, number of events= 114
```

The coefficient for financial aid, which is the focus of the study, now has a two-sided p-value greater than 0.05; a one-sided test is appropriate here, however, because we expect the coefficient to be negative, so there is still marginal evidence for the effect of this covariate on the time of rearrest.

As mentioned, tests for the proportional-hazards assumption are obtained from `cox.zph()`, which computes a test for each covariate, along with a global test for the model as a whole:

```
cox.zph(mod.allison.4)

            rho   chisq      p
finyes -0.00657 0.00507 0.9433
age    -0.20976 6.54147 0.0105
prio   -0.08004 0.77288 0.3793
GLOBAL       NA 7.13046 0.0679
```

There is, therefore, strong evidence of non-proportional hazards for `age`, while the global test (on 3 degrees of freedom) has a p-value slightly in excess of 0.05. These tests are sensitive to linear trends in the hazard.

Plotting the object returned by `cox.zph()` produces graphs of the scaled Schoenfeld residuals against transformed time (see Figure 3):

```
par(mfrow=c(2, 2))
plot(cox.zph(mod.allison.4))
```

Interpretation of these graphs is greatly facilitated by smoothing, for which purpose `cox.zph()` uses a smoothing spline, shown on each graph by a solid line; the broken lines represent $\pm$ 2-standard-error envelopes around the fit. Systematic departures from a horizontal line are indicative of non-proportional hazards. The assumption of proportional hazards appears to be supported for the covariates `fin` (which is, recall, a two-level factor, accounting for the two bands in the graph) and `prio`, but there appears to be a trend in the plot for `age`, with the `age` effect declining with time; this effect was also detected in the test reported above.

One way of accommodating non-proportional hazards is to build interactions between covariates and time into the Cox regression model; such interactions are themselves time-dependent covariates. For example, based on the diagnostics just examined, it seems reasonable to consider a linear interaction of time and `age`; using the previously constructed `Rossi.2` data frame:

```
mod.allison.5 <- coxph(Surv(start, stop, arrest.time) ~
    fin + age + age:stop + prio,
    data=Rossi.2)
mod.allison.5

Call:
coxph(formula = Surv(start, stop, arrest.time) ~ fin + age +
```

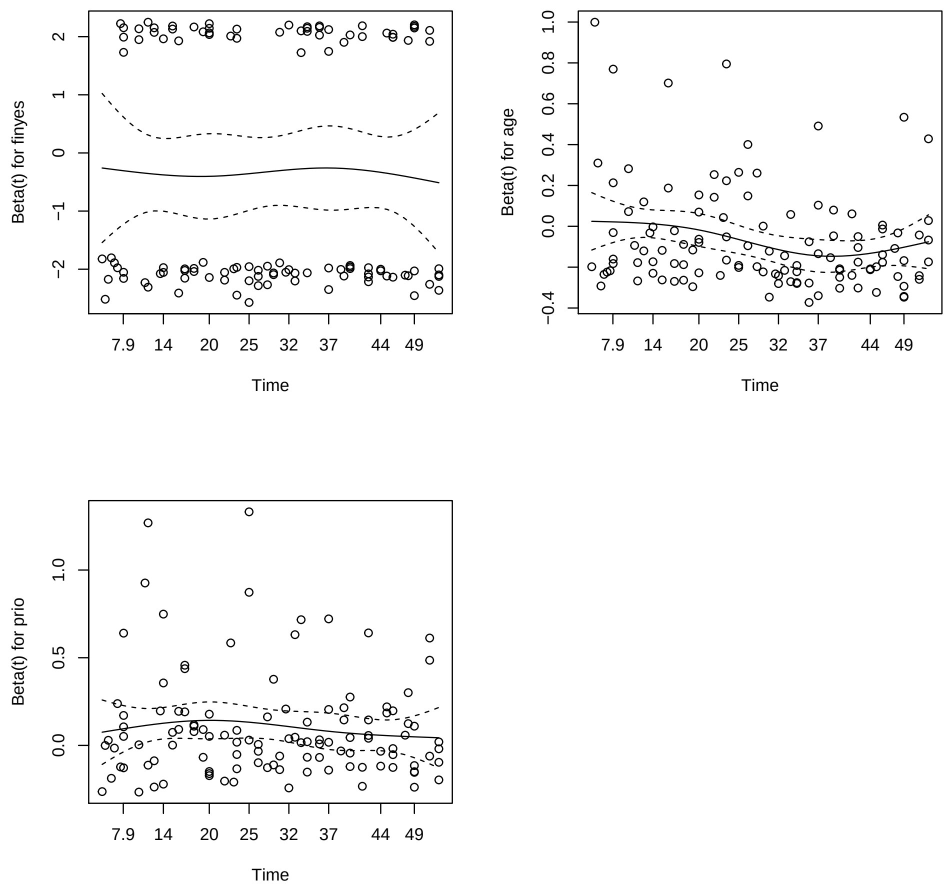

Figure 3: Plots of scaled Schoenfeld residuals against transformed time for each covariate in a model fit to the recidivism data. The solid line is a smoothing spline fit to the plot, with the broken lines representing a ± 2-standard-error band around the fit.

```
    age:stop + prio, data = Rossi.2)

              coef exp(coef) se(coef)     z       p
finyes   -0.34856   0.70570  0.19023 -1.83 0.06690
age       0.03228   1.03280  0.03943  0.82 0.41301
prio      0.09818   1.10316  0.02726  3.60 0.00032
age:stop -0.00383   0.99617  0.00147 -2.61 0.00899

Likelihood ratio test=36  on 4 df, p=3e-07
n= 19809, number of events= 114
```

As expected, the coefficient for the interaction is negative with a small p-value: The effect of `age` declines with time.[10] The model does not require a "main-effect" term for `stop` (i.e., time); such a term would be redundant, because the time effect is the baseline hazard.

An alternative to incorporating an interaction in the model is to divide the data into *strata* based on the value of one or more covariates. Each stratum is permitted to have a different baseline hazard function, while the coefficients of the remaining covariates are assumed to be constant across strata. An advantage of this approach is that we do not have to assume a particular form of interaction between the stratifying covariates and time. A disadvantage is the resulting inability to examine the effects of the stratifying covariates. Stratification is most natural when a covariate takes on only a few distinct values, and when the effect of the stratifying variable is not of direct interest. In our example, `age` takes on many different values, but we can create categories by arbitrarily dissecting the variable into class intervals. After examining the distribution of `age`, we decided to define four intervals: 19 or younger; 20 to 25; 26 to 30; and 31 or older. We use the `recode()` function in the **car** package to categorize `age`:[11]

```
Rossi$age.cat <- recode(Rossi$age, " lo:19=1; 20:25=2; 26:30=3; 31:hi=4 ")
xtabs(~ age.cat, data=Rossi)

age.cat
  1   2   3   4
 66 236  66  64
```

Most of the individuals in the data set are in the second age category, 20 to 25, but because this is a reasonably narrow range of ages, we did not feel the need to sub-divide the category.

A stratified Cox regression model is fit by including a call to the `strata()` function on the right-hand side of the model formula. The arguments to this function are one or more stratifying variables; if there is more than one such variable, then the strata are formed from their cross-classification. In the current illustration, there is just one stratifying variable:

```
mod.allison.6 <- coxph(Surv(week, arrest) ~
    fin + prio + strata(age.cat), data=Rossi)
mod.allison.6

Call:
coxph(formula = Surv(week, arrest) ~ fin + prio + strata(age.cat),
    data = Rossi)

          coef exp(coef) se(coef)     z      p
```

[10] That is, initially, `age` has a positive partial effect on the hazard (given by the `age` coefficient, 0.032), but this effect gets progressively smaller with time (at the rate −0.0038 per week), becoming negative after about 10 weeks.

[11] An alternative is to use the standard R function `cut()`: `cut(Rossi$age, c(0, 19, 25, 30, Inf))`. See Chapter 2 of the *R Companion*.

```
finyes -0.3408    0.7112   0.1902 -1.79 0.0732
prio    0.0941    1.0986   0.0270  3.48 0.0005

Likelihood ratio test=13.4  on 2 df, p=0.001
n= 432, number of events= 114
```

```
cox.zph(mod.allison.6)
```

```
           rho  chisq     p
finyes -0.0183 0.0392 0.843
prio   -0.0771 0.6859 0.408
GLOBAL      NA 0.7299 0.694
```

There is no evidence of non-proportional hazards for the remaining covariates.

5.2 Influential Cases

Specifying the argument `type=dfbeta` to `residuals` produces a matrix of estimated changes in the regression coefficients upon deleting each case in turn; likewise, `type=dfbetas` produces the estimated changes in the coefficients divided by their standard errors (cf., Sections 8.3 and 8.6.2 of the *R Companion* for influence diagnostics for linear and generalized linear models).

For example, for the model regressing time to rearrest on financial aid, age, and number of prior offenses:

```
dfbeta <- residuals(mod.allison.4, type="dfbeta")
par(mfrow=c(2, 2))
for (j in 1:3) {
    plot(dfbeta[, j], ylab=names(coef(mod.allison.4))[j])
    abline(h=0, lty=2)
    }
```

The index plots produced by these commands appear in Figure 4. Comparing the magnitudes of the largest dfbeta values to the regression coefficients suggests that none of the cases are terribly influential individually, even though some of the dfbeta values for `age` are large compared with the others.[12]

5.3 Nonlinearity

Nonlinearity—that is, an incorrectly specified functional form in the parametric part of the model—is a potential problem in Cox regression as it is in linear and generalized linear models (see Sections 8.4.2 and 8.6.3 of the *R Companion*). The *martingale residuals* may be plotted against covariates to detect nonlinearity, and may also be used to form component-plus-residual (or partial-residual) plots, again in the manner of linear and generalized linear models.

For the regression of time to rearrest on financial aid, age, and number of prior arrests, let us examine plots of martingale residuals and partial residuals against the last two of these covariates; nonlinearity is not an issue for financial aid, because this covariate is a dichotomous factor:

```
par(mfrow=c(2, 2))
res <- residuals(mod.allison.4, type="martingale")
X <- as.matrix(Rossi[, c("age", "prio")]) # matrix of covariates
par(mfrow=c(2, 2))
```

[12]As an exercise, the reader may wish to identify these cases and, in particular, examine their ages.

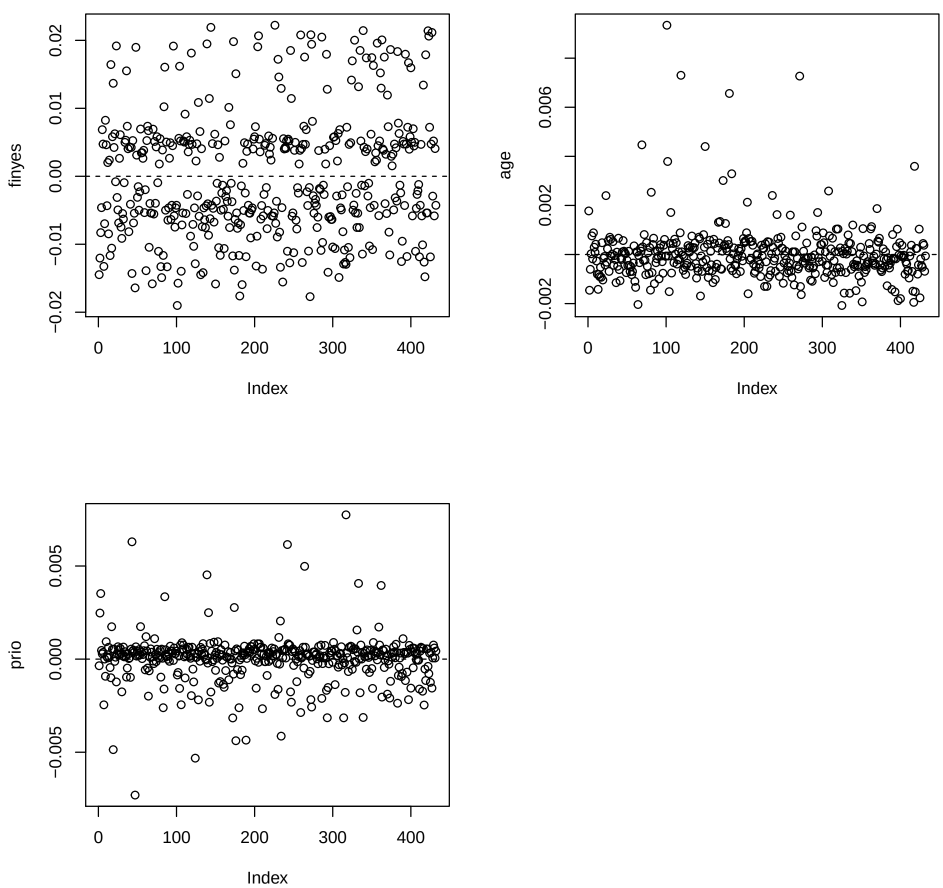

Figure 4: Index plots of dfbeta for the Cox regression of time to rearrest on `fin`, `age`, and `prio`.

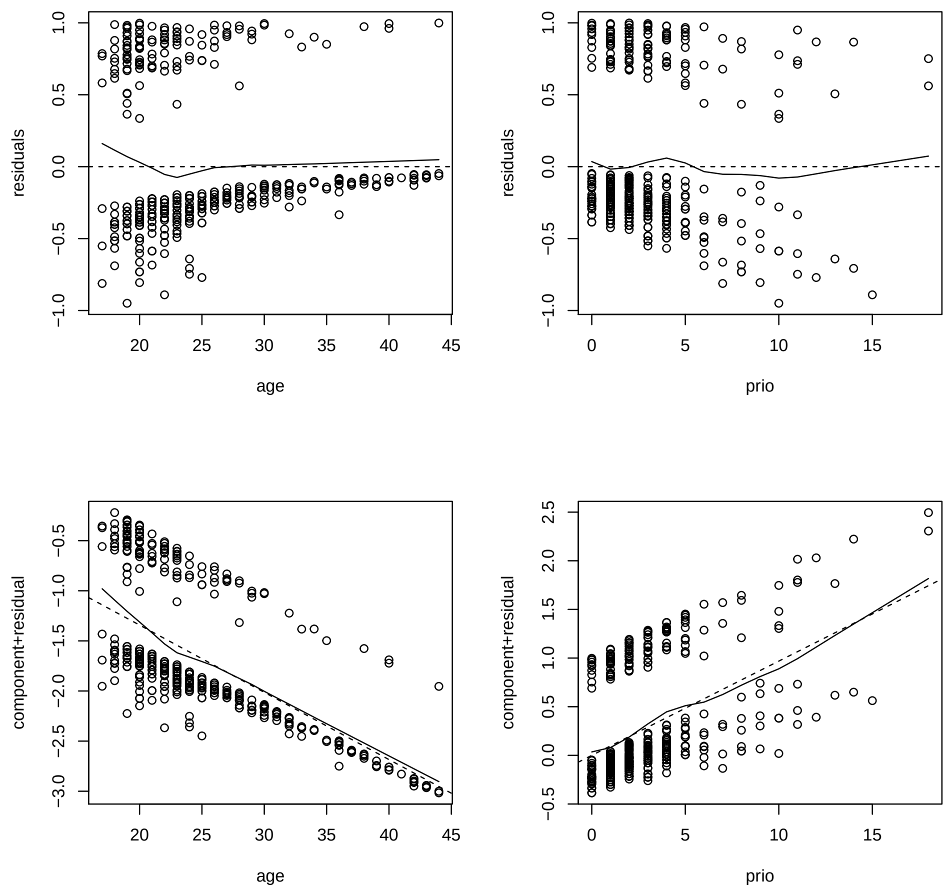

Figure 5: Martingale-residual plots (top) and component-plus-residual plots (bottom) for the covariates `age` and `prio`. The broken lines on the residual plots are at the vertical value 0, and on the component-plus-residual plots are fit by linear least-squares; the solid lines are fit by local linear regression (lowess).

```
for (j in 1:2) {  # residual plots
    plot(X[, j], res, xlab=c("age", "prio")[j], ylab="residuals")
    abline(h=0, lty=2)
    lines(lowess(X[, j], res, iter=0))
    }

b <- coef(mod.allison.4)[c(2,3)]  # regression coefficients
for (j in 1:2) {  # component-plus-residual plots
    plot(X[, j], b[j]*X[, j] + res, xlab=c("age", "prio")[j],
        ylab="component+residual")
    abline(lm(b[j]*X[, j] + res ~ X[, j]), lty=2)
    lines(lowess(X[, j], b[j]*X[, j] + res, iter=0))
    }
```

The resulting residual and component-plus-residual plots appear in Figure 5. As in the plots of Schoenfeld residuals, smoothing these plots is also important to their interpretation; The smooths in Figure 5 are produced by local linear regression using the `lowess` function; setting `iter=0` selects a non-robust smooth, which is generally advisable in plots that may be banded. Nonlinearity, it appears, is slight here.

6 Complementary Reading and References

There are many texts on survival analysis: Cox and Oakes (1984) is a classic (if now slightly dated) source, coauthored by the developer of the Cox model. As mentioned, the running example in this appendix is adapted from Allison (2010), who presents a highly readable introduction to survival analysis based on the SAS statistical package, but nevertheless of general interest. Allison (2014) is a briefer treatment of the subject by the same author. Another widely read and wide-ranging text on survival analysis is Hosmer et al. (2008). The book by Therneau and Grambsch (2000) is also worthy of mention here; Therneau is the author of the **survival** package for R, and the text, which focusses on relatively advanced topics, develops examples using both the **survival** package and SAS. Extensive documentation for the **survival** package may be found in Therneau (1999); although the **survival** package has continued to evolve since 1999, this technical report remains a useful source of information about it, as do the several vignettes in the **survival** pacakge; see `vignette(package="survival")`.

References

Allison, P. D. (2010). *Survival Analysis Using SAS: A Practical Guide.* SAS Institute, Cary NC, second edition.

Allison, P. D. (2014). *Event History and Survival Analysis.* Sage, Thousand Oaks CA, second edition.

Cox, D. R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B*, 34:187–220.

Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data.* Chapman and Hall, London.

Fox, J. (2017). *Using the R Commander: A Point-and-Click Interface for R.* CRC/Chapman Hall, Boca Raton, FL.

Fox, J. and Weisberg, S. (2019). *An R Companion to Applied Regression.* Sage, Thousand Oaks, CA, third edition.

Hosmer, Jr., D. W., Lemeshow, S., and May, S. (2008). *Applied Survival Analysis: Regression Modeling of Time-to-Event Data.* Wiley, New York, second edition.

Rossi, P. H., Berk, R. A., and Lenihan, K. J. (1980). *Money, Work and Crime: Some Experimental Results.* Academic Press, New York.

Therneau, T. M. (1999). A package for survival analysis in S. Technical report, Mayo Foundation, Rochester MN.

Therneau, T. M. and Grambsch, P. M. (2000). *Modeling Surival Data: Extending the Cox Model.* Springer, New York.