Montogue

Tutorial ST2
Survival Analysis with R

Lucas Monteiro Nogueira

• Summary •	
Problem 1	Introduction and Kaplan-Meier estimation
Problem 2	Cox proportional hazards – Model selection
Problem 3	Cox proportional hazards – Testing the proportional hazards assumption and obtaining a basic plot
Problem 4	Cox proportional hazards – Model diagnostics
Problem 5	Cox proportional hazards – Stratification and time-varying covariates
Problem 6	Parametric survival analysis – Weibull distribution

PROBLEMS

▶ Datasets

Download dataset *hipData.xlsx* in our Google Drive folder.

▶ Problem 1 – Introduction and Kaplan-Meier estimation

This tutorial uses the *hipData* dataset. This fictitious dataset, which is loosely based on Cleves *et al.* (2010), describes a clinical trial for an inflatable device that reportedly protects the elderly from hip fracture when they fall. 120 subjects ages 61 to 84 were included in the trial; half of the subjects are men and the other half are women. 60 subjects – 30 men and 30 women – were given the hip device, whereas the other 60 did not use the device. The subjects were then followed for 3 to 23 months to evaluate whether the incidence of hip fracturing is indeed lower in subjects that use the device. The *hipData* file contains 120 datapoints and 7 columns, namely:

id → This is the ID of the subjects involved in the study; it ranges from 1 to 120.

fracture → This column indicates whether the failure event – i.e., a hip fracture – was registered in the follow-up period; a 1 indicates that a hip fracture occurred, whereas a 0 indicates that a hip fracture did not occur. 65 of the 120 subjects registered fractured hips over the course of the study.

age → This column indicates the age of the subjects, which, as mentioned above, ranges from 61 to 84, with a mean value close to 72.

calcium → This column contains the average intravenous concentration of a bone-fortifying drug that was assigned to subjects with the expectation that it, in conjunction with the hip device, would reduce the incidence of fractures. We will not use this covariate in this particular tutorial.

male → This indicator variable equals 0 for female subjects and 1 for male ones.

ttf → This variable, which stands for "time-to-failure," indicates the follow-up time. As mentioned above, subjects were followed for 3 to 23 months, with a mean value equal to 11.5 months.

To load the dataset, save it in your working folder and use the command *read.table*:

```
> hip <- read.table("hipData.txt", header = TRUE)
```

The basis of all models developed in a R survival analysis the *Surv* object, which converts censored survival data to a format compatible with commands such as *coxph* or *survfit*. The first argument of *Surv* is the

follow-up time, which in the present case is *ttf*, and the second argument is the failure event, which in the present case is *fracture*:

```
> Y <- Surv(hip$ttf,hip$fracture)
```

To create a Kaplan-Meier estimate, we appeal to R's *survfit* function. Confidence intervals obtained through the delta method are prone to yield values above one and below zero, so we opt for a log-log transformation by adding the argument *log-log* to the *survfit* call:

```
> km.estim <- survfit(Y ~ 1, conf.type="log-log")
```

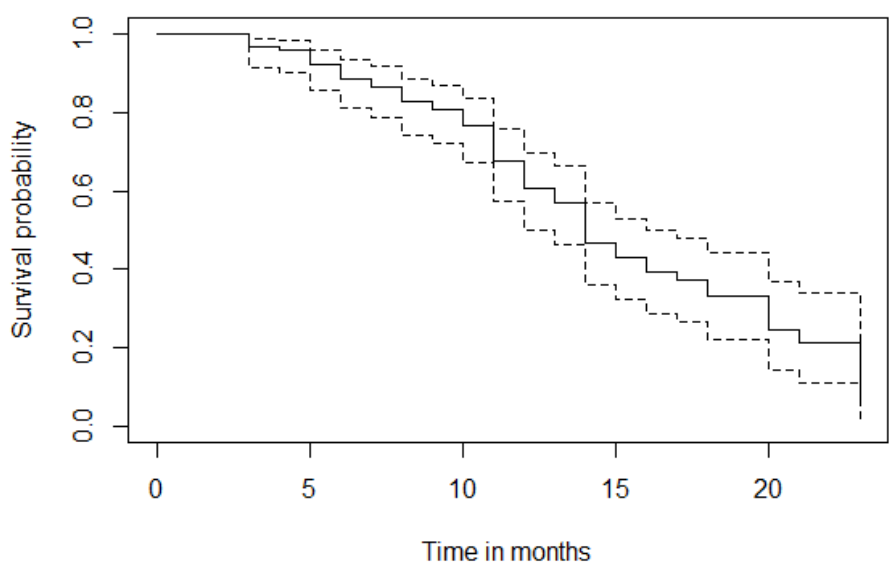
Typing the structure name yields some details of the estimate:

```
> km.estim
Call: survfit(formula = Y ~ 1, conf.type = "log-log")

       n events median 0.95LCL 0.95UCL
[1,] 120     65     14      13      16
```

These basic stats indicate that 65 of the 120 subjects experienced a hip fracture, and the mean survival time was 14 months with a confidence interval ranging from 13 to 16. Next, we can use the *plot* command to produce a survival curve for the KM estimator:

```
> plot(km.estim, conf.int=T, xlab="Time in months",
ylab="Survival probability", lwd=1.5)
```



We proceed to create a Kaplan-Meier model with *protect* as a covariate:

```
> km.estim.ptct <- survfit(Y ~ protect, data=hip)
```

Next, we implement a log-rank test with *survdiff* to assess the significance of the *protect* covariate:

```
> survdiff(Y ~ protect, data=hip)
Call:
survdiff(formula = Y ~ protect, data = hip)

            N Observed Expected (O-E)^2/E (O-E)^2/V
protect=0  60       45     35.3      2.67      6.51
protect=1  60       20     29.7      3.17      6.51

 Chisq= 6.5  on 1 degrees of freedom, p= 0.01
```

The *p*-value is small enough to indicate a statistically significant difference between survival of individuals that used the hip device and those that did not. Next, let us prepare some survival probabilities for select follow-up times:

```
> timeValues = c(1,2,5,10,15,20,23)
> summary(km.estim.ptct, times = timeValues)
Call: survfit(formula = Y ~ protect, data = hip)
                protect=0
time n.risk n.event survival std.err  lower 95% CI upper 95% CI
   1     60       0    1.000  0.0000       1.0000        1.000
   2     60       0    1.000  0.0000       1.0000        1.000
   5     57       5    0.916  0.0362       0.8473        0.989
```

```
  10      45       10    0.740  0.0578        0.6349         0.863
  15      18       22    0.311  0.0649        0.2069         0.468
  20       6        5    0.150  0.0605        0.0679         0.331
  23       2        3    0.000     NaN            NA            NA
                protect=1
time n.risk n.event survival std.err lower 95% CI upper 95% CI
   1      60        0    1.000  0.0000        1.0000         1.000
   2      60        0    1.000  0.0000        1.0000         1.000
   5      51        4    0.931  0.0332        0.8686         0.999
  10      31        6    0.800  0.0578        0.6945         0.922
  15      18        6    0.615  0.0800        0.4764         0.794
  20       6        3    0.417  0.1118        0.2470         0.706
  23       2        1    0.209  0.1578        0.0474         0.919
```

As highlighted above, the probability of survival at 10-months follow-up is 80% for those who were assigned the hip device and 74% for controls. At 20-months follow-up, we estimate that subjects who received the hip device had a survival probability of 41.7%, whereas controls had a survival probability of 15%.

▶ Problem 2 – Cox proportional hazards – Model selection

We now turn to a semiparametric analysis of the *hipData* dataset with Cox regression. Although we are interested in evaluating the performance of the hip device, factors such as the subjects' age and use of the bone-fortifying drug mentioned earlier may have confounding effects. We shall ignore the influence of the drug, but we will include *age* as an additional predictor.

We press on to prepare three Cox proportional hazards regression models. In Model A, the only predictor is *protect*; in Model B, the predictors are *protect* and *age*; in Model C, the predictors are *protect*, *age*, and an interaction term *protect:age*. Let us prepare the three models:

```
> Y <- Surv(hip$ttf, hip$fracture)
> modelA <- coxph(Y ~ protect, data=hip)
> modelB <- coxph(Y ~ protect + age, data=hip)
> modelC <- coxph(Y ~ protect + age + protect:age,
data=hip)
```

Then, we use the *summary* command to list the results of each Cox model.

```
> summary(modelA)
Call:
coxph(formula = Y ~ protect, data = hip)

  n= 120, number of events= 65

           coef exp(coef) se(coef)      z Pr(>|z|)
protect -0.6877    0.5028   0.2699 -2.548   0.0108 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

        exp(coef) exp(-coef) lower .95 upper .95
protect    0.5028      1.989    0.2962    0.8532

Concordance= 0.558  (se = 0.037 )
Likelihood ratio test= 6.97  on 1 df,   p=0.008
Wald test            = 6.49  on 1 df,   p=0.01
Score (logrank) test = 6.75  on 1 df,   p=0.009
```

In Model A, we see that the *protect* covariate is associated with a negative coefficient (≈ -0.688) and a hazard of ≈ 0.5, which indicates that subjects who wear the hip device have about half the hazard of fracturing their hips relatively to individuals who do not. The significance level, $p \approx 0.01$, is reasonable.

Let's summarize Model B:

```
> summary(modelB)
Call:
```

```
coxph(formula = Y ~ protect + age, data = hip)

  n= 120, number of events= 65
           coef exp(coef) se(coef)      z Pr(>|z|)
protect -0.8819    0.4140   0.2783 -3.169 0.001530 **
age      0.0879    1.0919   0.0238  3.693 0.000222 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

        exp(coef) exp(-coef) lower .95 upper .95
protect     0.414     2.4154     0.240    0.7143
age         1.092     0.9159     1.042    1.1440

Concordance= 0.655  (se = 0.039 )
Likelihood ratio test= 20.83  on 2 df,   p=3e-05
Wald test            = 19.21  on 2 df,   p=7e-05
Score (logrank) test = 19.67  on 2 df,   p=5e-05
```

This particular model corroborates the fact that wearing the hip device reduces the hazard for hip fracture and further indicates that increasing age is associated with a greater hazard for hip fracture; specifically, the exponentiated coefficient associated with *age* is about 1.09, which suggests that increasing *age* by one unit raises the hazard for hip fracture by 9%. Further, note that the significance levels of the two predictors ($p \approx 0.0015$ for *protect* and $p \approx 0.0002$ for *age*) are very good.

Finally, let's summarize Model C:

```
> summary(modelC)
Call:
coxph(formula = Y ~ protect + age + protect:age, data =
hip)

  n= 120, number of events= 65

               coef exp(coef) se(coef)      z Pr(>|z|)
protect     -2.65438   0.07034  3.56970 -0.744  0.45713
age          0.07884   1.08204  0.02995  2.632  0.00848 **
protect:age  0.02393   1.02421  0.04795  0.499  0.61778
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

             exp(coef) exp(-coef) lower .95 upper .95
protect        0.07034    14.2162 6.437e-05    76.868
age            1.08204     0.9242 1.020e+00     1.147
protect:age    1.02421     0.9764 9.323e-01     1.125

Concordance= 0.657  (se = 0.04 )
Likelihood ratio test= 21.08  on 3 df,   p=1e-04
Wald test            = 18.19  on 3 df,   p=4e-04
Score (logrank) test = 20.39  on 3 df,   p=1e-04
```

This third model confirms Model B's indications that *protect* is associated with a negative coefficient and *age* is associated with a small positive coefficient, but in this case the significance levels for *protect* and the newly introduced *protect:age* interaction term are very poor. Further, *protect* is associated with a very large negative coefficient (≈ -2.65), which leads to a strikingly small hazard of 0.07. The low significance levels and the suspicious coefficient for *protect* discourage us from working with Model C any further.

Instead of selecting models on the basis of summary statistics only, we can assess nested models using a (partial) likelihood ratio test. The log-likelihoods for the three models are obtained as follows:

```
> logLik(modelA)
'log Lik.' -247.2021 (df=1)
> logLik(modelB)
'log Lik.' -240.2717 (df=2)
> logLik(modelC)
'log Lik.' -240.1464 (df=3)
```

Log likelihood ratio tests are primarily intended for nested models; accordingly, we can compare model B with model A and model C with model B. The likelihood ratio statistic is given by

$$2\times\left(l\left(\hat{\beta}\right)_{\text{full model}} - l\left(\hat{\beta}\right)_{\text{reduced model}}\right)$$

That is, the statistic equals two times the difference between the log likehood statistic for the "full" model and the value for the "reduced" model. Comparing Model B with Model A, we type

```
> 2*(-240.2717 + 247.2021)
[1] 13.8608
```

This statistic is then entered into a chi-square distribution with degrees of freedom equal to the difference between the d.f.'s of the two models, that is, *df* = 2 – 1 = 1:

```
> pchisq(13.8608, df=1, lower.tail=F)
[1] 0.0001968621
```

This very small value indicates that *protect* also belongs in the model if *age* is included.

In a second comparison, we compare Model C with Model B to assess the significance of the interaction term *protect:age*. The likelihood statistic and the significance level are calculated as follows:

```
> 2*(-240.1464 + 240.2717)
[1] 0.2506
> pchisq(0.2506, df=1, lower.tail=F)
[1] 0.6166529
```

This high *p*-value indicates that *protect:age* does not belong in the model when *protect* and *age* are included. These findings, in conjunction with the summary statistics obtained earlier, suggest that Model A is too simplistic and Model C is overstuffed, so we shall stick with Model B in the sequel.

Using analysis of variance

Instead of performing the tedious calculations outlined above, we can compare nested models directly with R's *anova* (analysis of variance) command. To compare Model B with Model A, for instance, we type

```
> anova(modelB, modelA)

Analysis of Deviance Table
 Cox model: response is  Y
 Model 1: ~ protect + age
 Model 2: ~ protect
   loglik  Chisq Df P(>|Chi|)
1 -240.27
2 -247.20 13.861  1 0.0001969 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1
```

The significance level output by R, highlighted in red, is similar to the one obtained in our manual calculations.

▶ Problem 3 – Cox proportional hazards – Testing the proportional hazards assumption and obtaining a basic plot

In Problem 2, we worked with Cox regression models but did not evaluate whether the models satisfy the proportional hazards assumption. The usual code to test the proportional hazards assumptions is to use the *cox.zph* function with two arguments, namely (1) the model to be tested, and (2) the *transform* = *rank* command, which requests that ranked survival times be tested against the Schoenfeld residuals rather than the actual survival times, which is the default. The output is shown below.

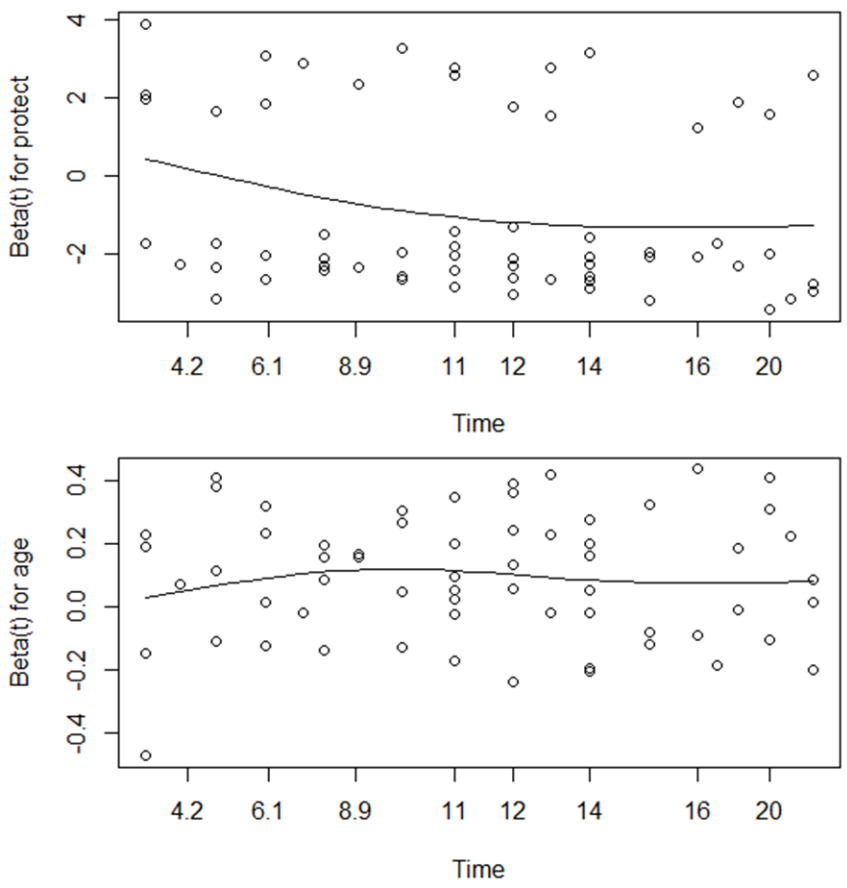
```
> cox.zph(modelB, transform=rank)
```

```
         chisq df     p
protect 3.007  1 0.083
age     0.106  1 0.745
GLOBAL  3.015  2 0.221
```

The *p*-values are rather high, therefore we cannot reject the proportional hazards assumption for *protect*, *age*, or the global model. We can also graph Schoenfeld residuals by wrapping the *cox.zph* command in a *plot* call:

```
> par(mfrow=c(2,1))
> plot(cox.zph(modelB, transform=rank), se=F,
var='protect')
> plot(cox.zph(modelB, transform=rank), se=F, var='age')
```

If the PH assumption is satisfied, the fitted curve should look horizontal because the Schoenfeld residuals would be independent of survival time. As shown below, this is approximately the case for *age* but not for *protect*, whose fitted curve slopes downward.



Now that we have established a basic regression model to investigate, we can proceed to create some Cox-adjusted survival curves. Suppose, for instance, that we are interested in plotting the survival curves for a subject with age equal to 72 – which is the mean age of the subjects in the *hipData* dataset – with *protect* set to 0 (not assigned the hip device) and 1 (assigned the hip device). We first create two *data.frame* objects:

```
> withoutHip <- data.frame(protect=0, age=72)
> withHip <- data.frame(protect=1, age=72)
```

Then, we appeal to the *plot* command with a *survfit* call nested within:

```
> plot(survfit(modelB, newdata=withoutHip), conf.int=F,
col=c("Red"), lwd=2, main="Adjusted survival for
age=72\nwith and without hip device", xlab="Time")
```

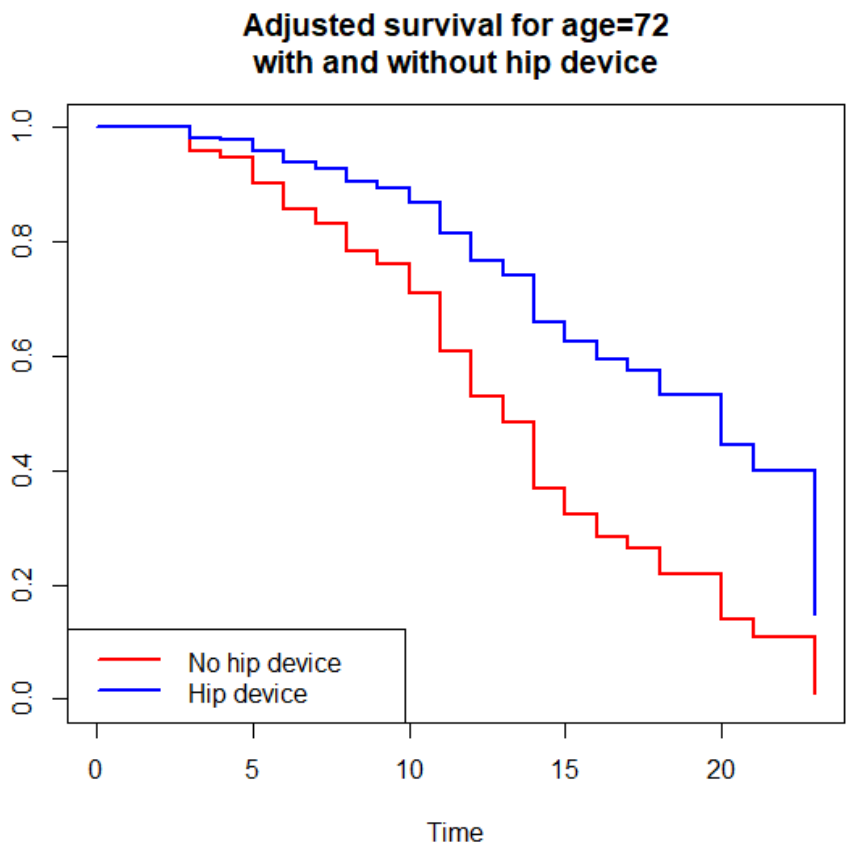
To superpose the second curve over the first, we use the *lines* command:

```
> lines(survfit(modelB, newdata=withHip), conf.int=F,
col=c("Blue"), lwd=2)
```

Lastly, we add a legend:

```
> legend("bottomleft", legend=c("No hip device", "Hip
device"), col=c("Red","Blue"), lwd=2)
```

The final plot is shown below. As can be seen, the population that was assigned the hip device had a consistently better survival profile than the population that did not use the device, especially in the follow-up range of 15 to 20 months.



▶ Problem 4 – Cox proportional hazards – Model diagnostics

Two simple diagnostics of interest are (1) investigating martingale residuals, and (2) assessing the effect of outliers.

Martingale residuals can be plotted by first creating a *residuals* object with *type* set to *martingale*, and then using the *plot* command. Although routinely used after fitting, martingale residual plots can also be useful *before* using Cox regression, in that they may suggest which covariates should be in the model and the form they should take. In the context at hand, we can use martingale residuals to check if using a log transformation for *age* would be a good idea. We begin by fitting a null model:

```
> null.model <- coxph(Y ~ 1)
```

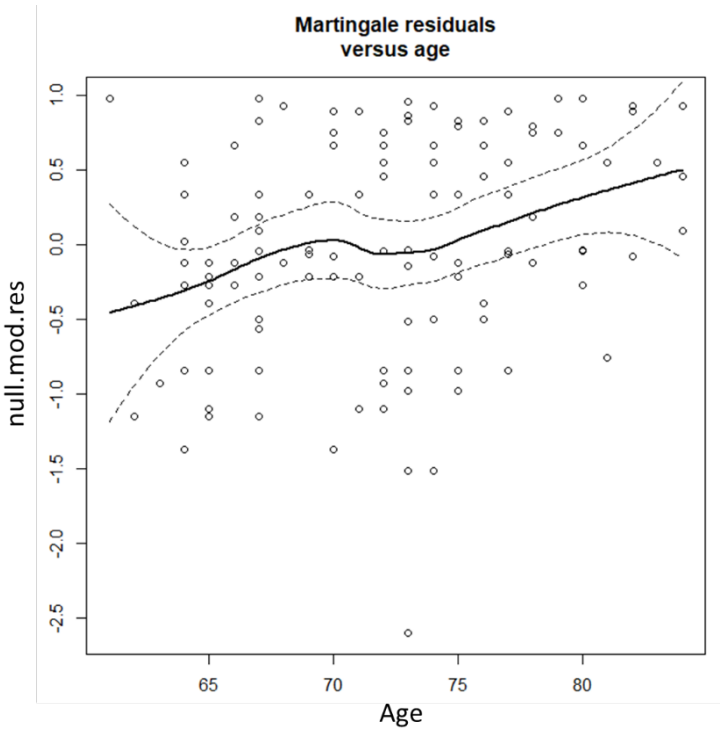
Then, we create a *residuals* object and resort to the usual *plot* command:

```
> null.mod.res <- residuals(null.model, type="martingale")
> plot(null.mod.res ~ hip$age)
```

Lastly, we evoke the *smoothSEcurve* function, which is given in the Additional Information section, to create a loess smooth curve for the data:

```
> smoothSEcurve(null.mod.res, hip$age)
> title("Martingale residuals\nversus age")
```

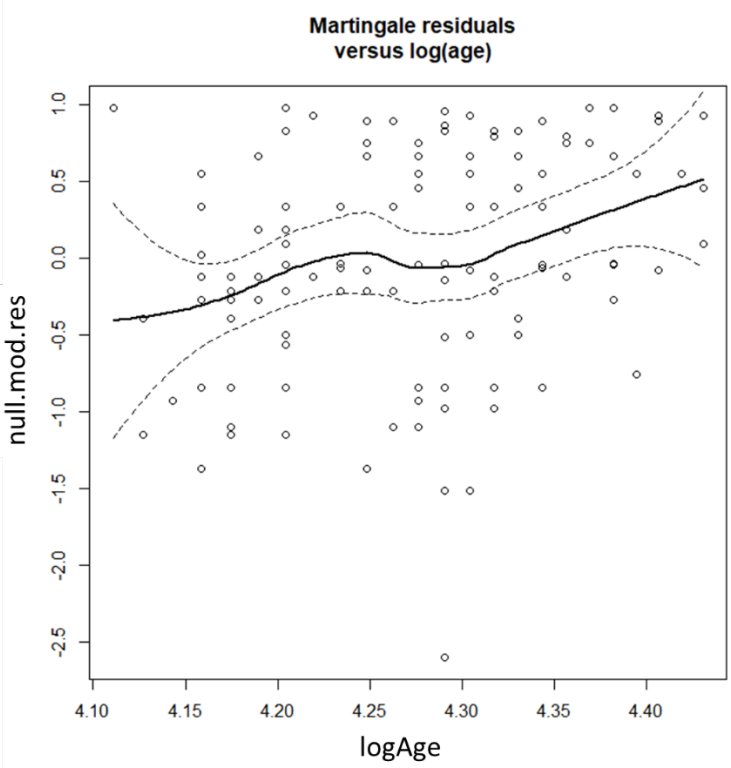
The ensuing plot is shown on the next page.



As can be seen, there are hints of a nonlinear behavior for the *age* covariate, so we may proceed to check if *log(age)* would be a better choice.

```
> logAge <- log(hip$age)
> plot(null.mod.res ~ logAge)
> smoothSEcurve(null.mod.res, logAge)
> title("Martingale residuals\nversus log(age)")
```

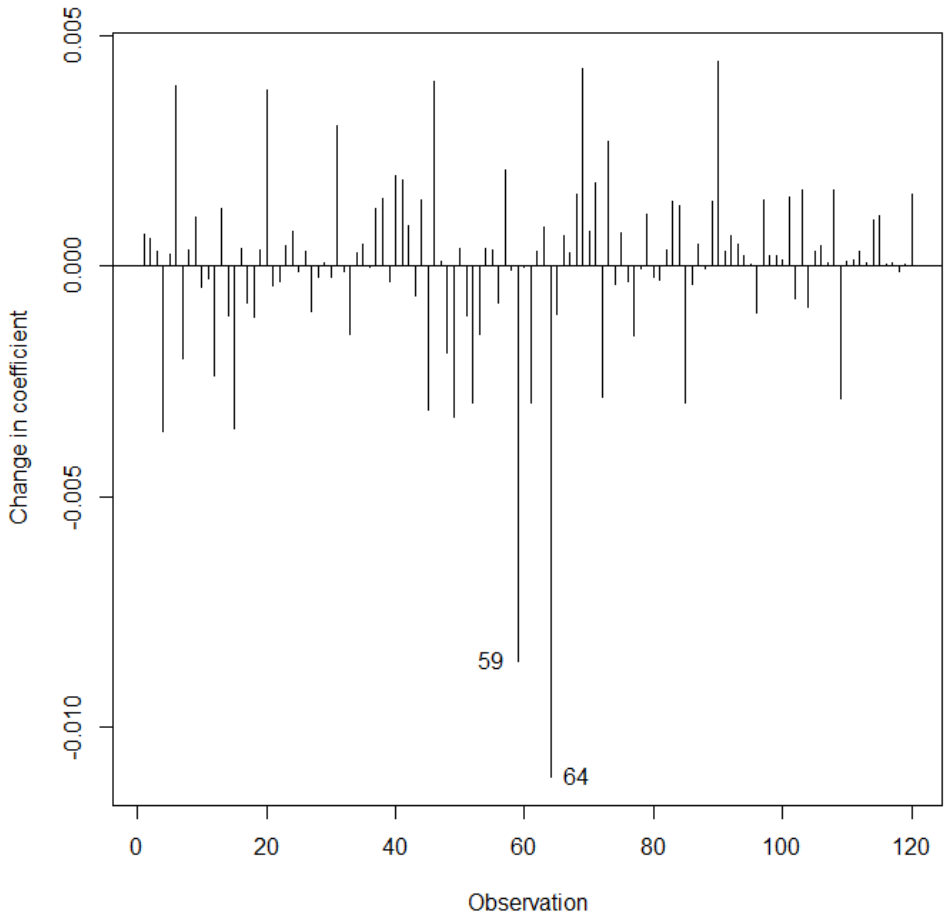
The ensuing plot is shown below. Clearly, a *log* transformation does little to straighten the loess curve, so we shall stick with an untransformed *age* covariate.



Next, let us verify the influence of outliers. We can assess the change in coefficient produced by a given observation using the *dfbeta* option in the *residuals* command and then plotting this object.

```
> modelB.dfbeta.res <- residuals(modelB, type="dfbeta")
> n.obs <- length(hip$ttf)
> index.obs <- 1:n.obs
> plot(modelB.dfbeta.res[,2] ~ index.obs, type="h",
xlab="Observation", ylab="Change in coefficient")
> abline(h=0)
```

We add a *[,2]* to the *modelB.dfbeta.res* call because we are interested in the *age* covariate. We choose *h* under *type* because we want the residuals to be plotted as spikes. Lastly, the *abline* command adds a horizontal line through 0. The plot is shown below.



Clearly, there are two observations that, if excluded, would appreciably reduce the coefficient associated with the *age* covariate. We can identify these coefficients with the mouse by initializing R's locator feature with the command

```
> identify(modelB.dfbeta.res[,2] ~ index.obs)
```

and then left-clicking on the two spikes. In doing so, we find that the two spikes are associated with observations 59 and 64. Referring to the *hipData* dataset, we see that observation 59 is a 61-year-old woman who was followed for a mere 3 months and received a hip implant, but nonetheless fractured her hip. Observation 64, in turn, is an 80-year old male that was followed for a longer period (20 months) than most subjects and also registered a fractured hip.

▶ Problem 5 – Cox proportional hazards – Stratification and time-varying covariates

Given the fact that men and women are subject to different risk patterns for bone disease, a reasonable course of action would be to stratify our model for gender. This can be easily done by adding a *strata(male)* option to the formula in the *coxph* statement; let's name the gender-stratified model as *modelB.strat*:

```
> modelB.strat <- coxph(Y ~ protect + age + strata(male),
data=hip)
```

Let's summarize the statistics:

```
> summary(modelB.strat)
Call:
coxph(formula = Y ~ protect + age + strata(male), data =
hip)

  n= 120, number of events= 65

            coef exp(coef) se(coef)      z Pr(>|z|)
protect -0.81287   0.44358  0.27819 -2.922 0.003478 **
age      0.08972   1.09387  0.02364  3.795 0.000147 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

        exp(coef) exp(-coef) lower .95 upper .95
protect    0.4436     2.2544    0.2571    0.7652
```

```
age         1.0939     0.9142    1.0443    1.1457

Concordance= 0.664  (se = 0.041 )
Likelihood ratio test= 20.37  on 2 df,   p=4e-05
Wald test            = 18.74  on 2 df,   p=9e-05
Score (logrank) test = 19.42  on 2 df,   p=6e-05
```

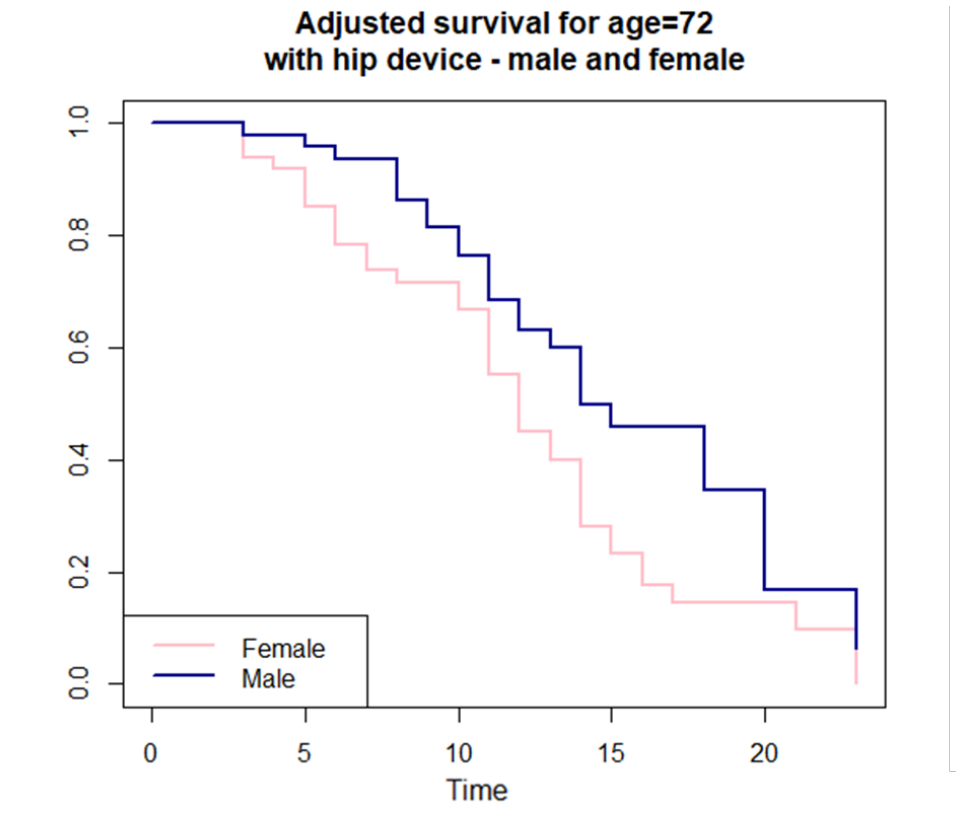
In this stratified model, *protect* and *age* covariates are still highly significant, but the coefficients are slightly different. The differences are listed in the following table.

	Nonstratified	Stratified
protect coef.	−0.8819	−0.81287
exp(protect)	0.4140	0.44358
protect p-value	0.001530	0.003478
age coef.	0.0879	0.08972
exp(age)	1.0919	1.09387
age p-value	0.000222	0.000147

Judging from the coefficients, the effect of *protect* – that is, wearing a hip device or not – is sensibly different when adjusting for gender, whereas the effect of *age* is more or less the same. With this in mind, let us plot survival curves for individuals aged 72 that were assigned the hip device, this time adjusting for gender.

```
> withHipFemale <- data.frame(protect=0, age=72, male=0)
> withHipMale <- data.frame(protect=0, age=72, male=1)

> plot(survfit(modelB.strat, newdata = withHipFemale),
conf.int=F, main="Adjusted survival for age=72\nwith hip
device - male and female", xlab="Time", lwd=2, col="Pink")
> lines(survfit(modelB.strat, newdata = withHipMale),
conf.int=F, lwd=2, col="DarkBlue")
> legend("bottomleft", legend=c("Female","Male"),
col=c("Pink","DarkBlue"), lwd=2)
```



The plot indicates that men who were assigned the hip device fared slightly better than women who wore the device, especially in the 5 to 10-month follow-up period and later on in the 15 to 20-month follow-up period.

One of the main drawbacks of a 'typical' Cox regression model is that it assumes the values of covariates to be set at some initial time and remain constant thereafter. This assumption can be relaxed if we allow for temporal variation in the covariates. For instance, the hip device studied hitherto may suffer material wear and lose efficiency with prolonged use; we can represent this hypothesis by including a *time transfer function* in the model. To implement this in R, we add a *tt()* term

to the regression formula and define the time transfer function at the end of the *coxph* call. In the present case, we suggest that *protect* has a time-dependent component of the form $tt(t) = x \times log(t)$, that is:

```
> modelB.tt <- coxph(Y ~ protect + tt(protect) + age,
tt=function(x,t,...) x*log(t), data=hip)
```

Summarizing the new model:

```
> summary(modelB.tt)

Call:
coxph(formula = Y ~ protect + tt(protect) + age, data =
hip,
    tt = function(x, t, ...) x * log(t))

  n= 120, number of events= 65

                coef exp(coef) se(coef)      z Pr(>|z|)
protect      1.31545   3.72645  1.20307  1.093  0.27421
tt(protect) -0.95195   0.38599  0.51292 -1.856  0.06346 .
age          0.08992   1.09408  0.02372  3.792  0.00015 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

Note that, in this particular model, the significances of protect and the time-varying covariate are rather low. The fitted function is $\beta(t) = 1.315 - 0.952log(t)$, which indicates that, in this formulation, the effectiveness of the hip device shows a decreasing trend over the course of the follow-up period.

▶ Problem 6 – Parametric survival analysis – Weibull distribution

The most widely used parametric approach to survival analysis is the Weibull distribution. The Weibull distribution has survival function

$$S(t) = \exp\left(-\exp\left(-\frac{\mu}{\sigma}\right)t^{1/\sigma}\right)$$

where t is time, μ is the mean parameter, and σ is the scale parameter. Taking a complementary log-log transformation $g(u) = log(-log(u))$ of the Weibull survival function and adjusting brings to

$$\log\left[-\log\left(S(t_i)\right)\right] = -\frac{\mu}{\sigma} + \frac{1}{\sigma}\log(t_i)$$

Thus, for a dataset that can be represented by a Weibull distribution, a plot of $y_i = log(-log(t_i))$ versus the logarithm of time should yield a straight line. Further, the linear fit yields estimates of the scale parameter, which is the reciprocal of the line slope, and the mean parameter, which can be obtained by noting that the intercept of the line equals $-\mu/\sigma$.

Turning to the hip device at hand, we first obtain a Kaplan-Meier estimate of the survival distribution:

```
> result.km <- survfit(Y ~ 1)
> survEst <- result.km$surv
> survTime <- result.km$time
```

Then, we perform the pertaining transformations:

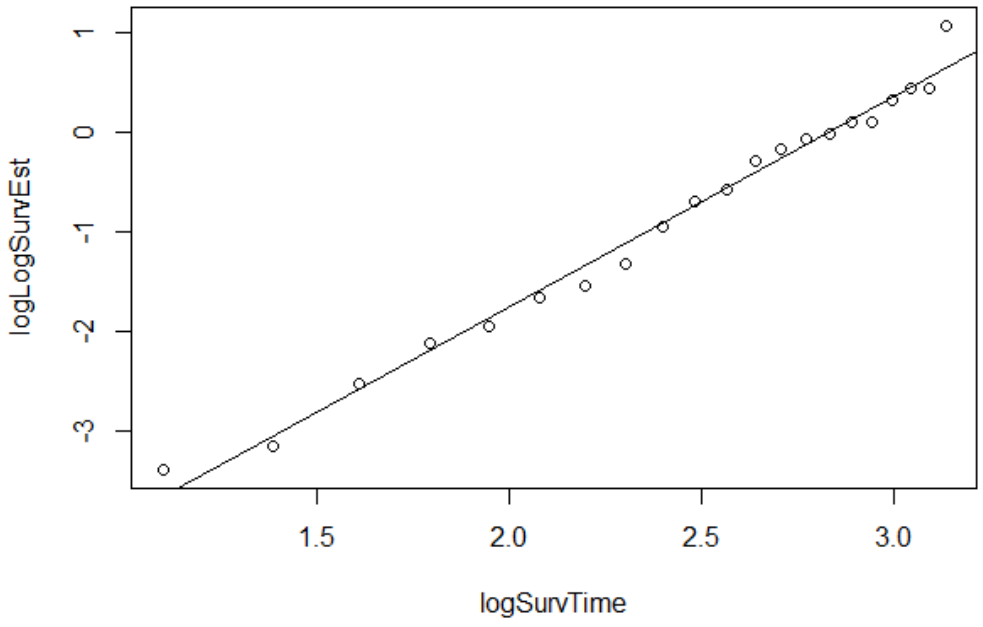
```
> logLogSurvEst <- log(-log(survEst))
> logSurvTime <- log(survTime)
```

Lastly, we plot *logLogSurvEst* versus *logSurvTime* and set up a linear fit:

```
> plot(logLogSurvEst ~ logSurvTime)
> result.lm <- lm(logLogSurvEst ~ logSurvTime)
> abline(result.lm)
```

The ensuing plot is shown below. As can be seen, there is a reasonable correspondence between the data points and the linear fit,

therefore we surmise that the hip device data can be represented by a Weibull distribution.



Fitting a Weibull distribution to survival data is similar to fitting a Cox regression model; the main difference is that instead of using the *coxph* function, we wrap the survivor object with the *survreg* function and specify *weibull* as the distribution of choice. For a model with *protect* as the only predictor, we'd write

```
> model.wbll <- survreg(Y ~ protect, dist="weibull",
data=hip)
```

Then, applying the *summary* command yields the statistics of the fitted model:

```
> summary(model.wbll)

Call:
survreg(formula = Y ~ protect, data = hip, dist =
"weibull")
              Value Std. Error     z       p
(Intercept)  2.7184     0.0614 44.30 <2e-16
protect      0.2753     0.1133  2.43  0.015
Log(scale)  -0.8882     0.0992 -8.96 <2e-16

Scale= 0.411

Weibull distribution
Loglik(model)= -232.9   Loglik(intercept only)= -236.3
     Chisq= 6.65 on 1 degrees of freedom, p= 0.0099
Number of Newton-Raphson Iterations: 7
n= 120
```

The Weibull distribution has scale parameter σ = 0.411 and coefficient γ = 0.2753 for the *protect* covariate. Importantly, the ratio $-\gamma/\sigma$ should yield the corresponding regression coefficient in the Cox model associated with the same data; mathematically,

$$\beta = -\frac{\gamma}{\sigma}$$

In the case at hand,

$$\beta = -\frac{0.2753}{0.411} = -0.670$$

We can check this relationship by fitting the same data to a Cox regression:

```
> model.cox <- coxph(Y ~ protect, data=hip)
Call:
coxph(formula = Y ~ protect, data = hip)

  n= 120, number of events= 65
```

```
           coef exp(coef) se(coef)      z Pr(>|z|)
protect -0.6877    0.5028   0.2699 -2.548   0.0108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

        exp(coef) exp(-coef) lower .95 upper .95
protect    0.5028      1.989    0.2962    0.8532

Concordance= 0.558  (se = 0.037 )
Likelihood ratio test= 6.97  on 1 df,   p=0.008
Wald test            = 6.49  on 1 df,   p=0.01
Score (logrank) test = 6.75  on 1 df,   p=0.009
```

The Cox model yields a coefficient of −0.6877 for *protect*, which is reasonably close to our estimate of −0.670.

ADDITIONAL INFORMATION

The *smoothSEcurve* function used in Problem 4 is given below.

```
smoothSEcurve <- function(yy, xx) {
  xx.list <- min(xx) + ((0:100)/100)*(max(xx)-min(xx))
  yy.xx <- predict(loess(yy~xx), se=T,
                   newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit~xx.list, lwd=2)
  lines(yy.xx$fit - qt(0.975,yy.xx$df)*yy.xx$se.fit ~
xx.list, lty=2)
  lines(yy.xx$fit + qt(0.975,yy.xx$df)*yy.xx$se.fit ~
xx.list, lty=2)
}
```

REFERENCES

- CLEVES, M., GUTIERREZ, R.G., GOULD, W. and MARCHENKO, Y.V. (2010). *An Introduction to Survival Analysis Using Stata*. 3rd edition. College Station: Stata Press.
- KLEINBAUM, D.G. and KLEIN, M. (2012). *Survival Analysis: A Self-Learning Text*. 3rd edition. Berlin/Heidelberg: Springer.
- MOORE, D.F. (2016). *Applied Survival Analysis Using R*. Berlin/Heidelberg: Springer.