

TMA4275; Project 3

Ola Rasmussen

Contents

Problem 1:	2
Introduction	2
a)	2
b)	4
c)	8
d)	10
e)	14
Problem 2:	16
Introduction	16
a)	16
b)	17
c)	21

Problem 1:

Introduction

In this problem we will study a tire failure known as tread and belt separation (TBS) in Kristov et al (2002).

a)

Here we will use the R function *coxph* to redo the two Cox regression analyses reported in Tables 2 and 3 in Kristov et al (2002), and then discuss the importance of the various covariates.

```
tire <- read.csv("Filer/tire.txt", sep = "")

mod <- coxph(Surv(Survival, Censor) ~ Age + Wedge + Inter + EB2B +
  Peel + Carbon + WxP, tire)
summary(mod)
```

```
## Call:
## coxph(formula = Surv(Survival, Censor) ~ Age + Wedge + Inter +
##      EB2B + Peel + Carbon + WxP, data = tire)
##
##      n= 34, number of events= 11
##
##              coef  exp(coef)    se(coef)      z Pr(>|z|)
## Age          2.116e+00  8.302e+00  1.452e+00  1.457  0.14499
## Wedge       -1.037e+01  3.140e-05  4.684e+00 -2.214  0.02684 *
## Inter       -1.130e+01  1.232e-05  4.692e+00 -2.409  0.01599 *
## EB2B        -1.405e+01  7.896e-07  8.529e+00 -1.647  0.09946 .
## Peel        -3.621e+01  1.887e-16  1.381e+01 -2.623  0.00873 **
## Carbon     -5.485e+01  1.515e-24  3.392e+01 -1.617  0.10591
## WxP          2.234e+01  5.058e+09  8.973e+00  2.490  0.01277 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## Age          8.302e+00  1.205e-01  4.821e-01  1.430e+02
## Wedge        3.140e-05  3.184e+04  3.238e-09  3.045e-01
## Inter        1.232e-05  8.119e+04  1.249e-09  1.215e-01
## EB2B         7.896e-07  1.266e+06  4.338e-14  1.437e+01
## Peel         1.887e-16  5.300e+15  3.345e-28  1.064e-04
## Carbon       1.515e-24  6.600e+23  2.022e-53  1.135e+05
## WxP          5.058e+09  1.977e-10  1.164e+02  2.198e+17
##
```

```
## Concordance= 0.935 (se = 0.045 )
## Likelihood ratio test= 27.81 on 7 df, p=2e-04
## Wald test = 10.13 on 7 df, p=0.2
## Score (logrank) test = 21.03 on 7 df, p=0.004
```

```
modsign <- coxph(Surv(Survival, Censor) ~ Wedge + Inter + Peel +
  WxP, tire)
summary(modsign)
```

```
## Call:
## coxph(formula = Surv(Survival, Censor) ~ Wedge + Inter + Peel +
##      WxP, data = tire)
##
##      n= 34, number of events= 11
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## Wedge -1.007e+01  4.224e-05  4.196e+00 -2.400  0.01638 *
## Inter -7.331e+00  6.549e-04  2.731e+00 -2.685  0.00726 **
## Peel -2.868e+01  3.513e-13  1.056e+01 -2.715  0.00663 **
## WxP  1.919e+01  2.160e+08  7.088e+00  2.707  0.00678 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## Wedge 4.224e-05  2.368e+04 1.132e-08 1.576e-01
## Inter 6.549e-04  1.527e+03 3.102e-06 1.383e-01
## Peel  3.513e-13  2.847e+12 3.584e-22 3.442e-04
## WxP  2.160e+08  4.629e-09 1.999e+02 2.335e+14
##
## Concordance= 0.89 (se = 0.054 )
## Likelihood ratio test= 19.75 on 4 df, p=6e-04
## Wald test = 10.8 on 4 df, p=0.03
## Score (logrank) test = 16.32 on 4 df, p=0.003
```

We see here that Age and WxP are the only covariates that are positive, i.e. they extend the life of the wheel a lot more than the other covariates.

b)

Now we will write our own R code, using the Breslow estimator, to estimate the integrated hazard rate for a component with all covariate values equal to unity, $A_0(t, x_0)$, for each of the two models considered in a). We will then compare them by plotting them. This can be seen in Figure 1. We will also find the corresponding estimated survival functions and visualize them in Figure 2. The Breslow estimator is given by,

$$\hat{A}_0(t) = \sum_{T_j \leq t} \frac{1}{\sum_{l \in \mathfrak{R}_j} r(\hat{\beta}, \mathbf{x}_l(T_j))} = \sum_{T_j \leq t} \frac{1}{\sum_{l \in \mathfrak{R}_j} \exp(\hat{\beta}^T \cdot \mathbf{x}_l(T_j))}, \quad (1.1)$$

where \mathfrak{R}_j is all the individuals who are alive just before T_j . To handle ties, we will take the average of the combinations happening at the same time. But if the covariates are fixed, as they are in our problem, we may use,

$$\hat{A}(t|\mathbf{x}_0) = \hat{A}_0(t) \cdot r(\hat{\beta}, \mathbf{x}_0) = \hat{A}_0(t) \cdot \exp(\hat{\beta}^T) = \sum_{T_j \leq t} \frac{\exp(\hat{\beta}^T)}{\sum_{l \in \mathfrak{R}_j} \exp(\hat{\beta}^T \cdot \mathbf{x}_l(T_j))}. \quad (1.2)$$

which is plotted in Figure 1. The corresponding survival function is given by,

$$\hat{S}(t|\mathbf{x}_0) = \prod_{j|T_j \leq t} (1 - \Delta \hat{A}(T_j|\mathbf{x}_0)), \quad (1.3)$$

which is plotted in Figure 2.

```
tireo <- tire[order(tire$Survival), ]
betasmod <- as.numeric(coef(mod))
betasmodsign <- c(as.numeric(modsign$coefficients))

hata <- function(tireo, betas) {
  Amod <- c()
  tireoc <- tireo[tireo$Censor == 1, ]$Survival
  for (i in unique(tireoc)) {
    Amoddenom <- c()
    denom <- c()
    tireotime <- tireo[tireo$Survival >= i, ]
    for (j in 1:nrow(tireotime)) {
      if (i %in% tireoc[duplicated(tireoc)]) {
        ties = c()
        for (k in 1:2) {
          ties <- c(ties, exp(sum(betas * c(as.numeric(tireotime[j,
            1:(ncol(tireotime) - 2)]))))))
        }
      }
      denom <- c(denom, sum(ties)/2)
    }
  }
  Amod <- c(Amod, Amoddenom/denom)
}
```

```

    } else {
      denom <- c(denom, exp(sum(betas * c(as.numeric(tireotime[j,
        1:(ncol(tireotime) - 2)])))))
    }
  }
  Amoddenom <- c(Amoddenom, sum(denom))
  Amod <- c(Amod, exp(sum(c(t(betas))))/Amoddenom)
}
return(cumsum(Amod))
}

```

```

plot(unique(tireo[tireo$Censor == 1, ]$Survival), hatA(tireo,
  betasmod), type = "s", lwd = 3, ylim = c(0, 0.5), col = "blue",
  main = "Breslow estimators", xlab = "Time", ylab = "Value")
lines(unique(tireo[tireo$Censor == 1, ]$Survival), hatA(tireo[,
  c(2, 3, 5, 7, 8, 9)], betasmodsign), type = "s", lwd = 3,
  col = "red")

```

```

plot(unique(tireo[tireo$Censor == 1, ]$Survival), cumprod(1 -
  hatA(tireo, betasmod)), type = "s", lwd = 3, xlim = c(0,
  1.3), ylim = c(0.4, 1), col = "blue", main = "Estimated survival functions",
  xlab = "Time", ylab = "Value")
lines(unique(tireo[tireo$Censor == 1, ]$Survival), cumprod(1 -
  hatA(tireo[, c(2, 3, 5, 7, 8, 9)], betasmodsign)), type = "s",
  lwd = 3, col = "red")

```

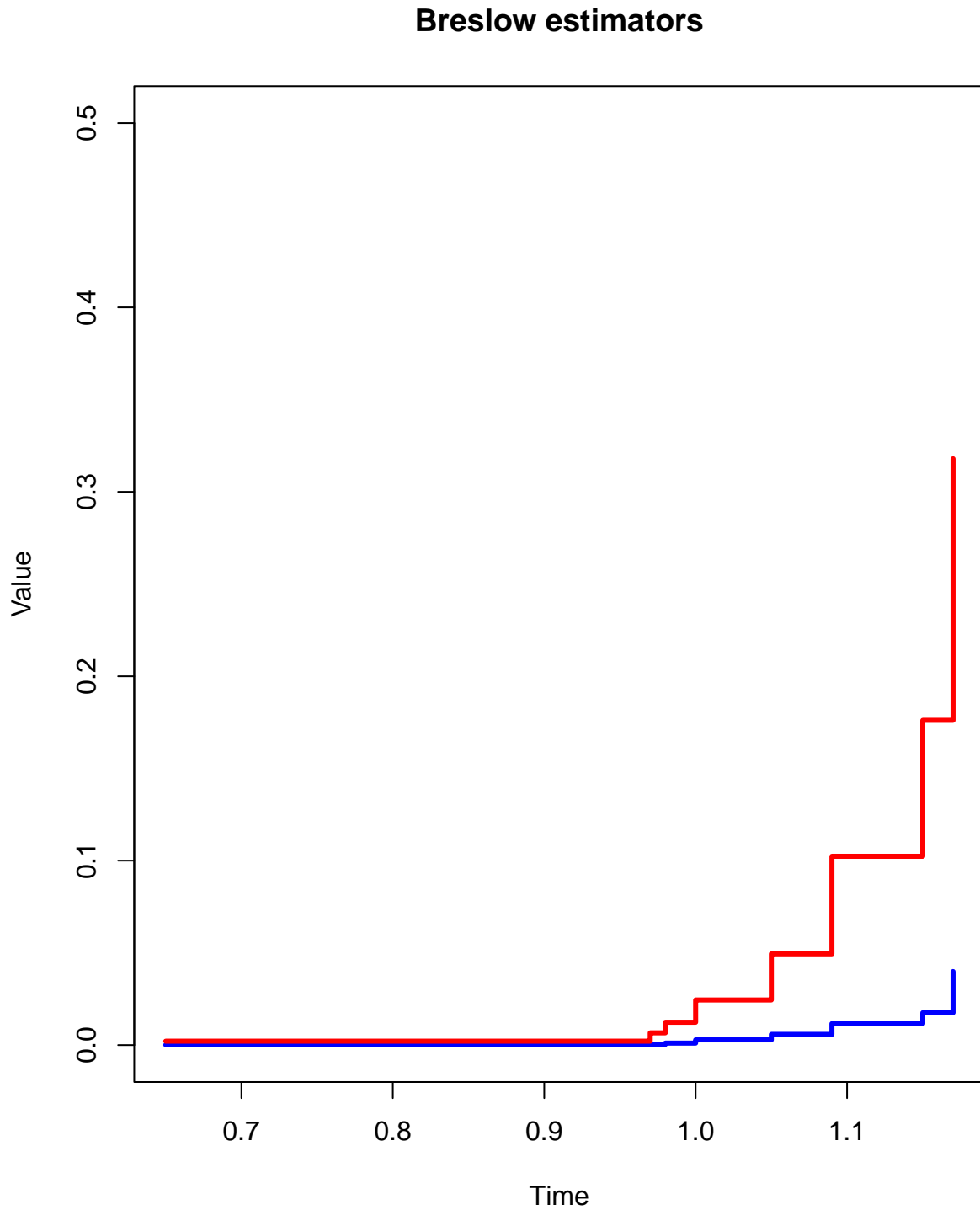


Figure 1: Here we have plotted the Breslow estimators for the two models. The blue line is for the full model, and the red is for the significant model. We see that using the significant model, the tires tend to fail at a faster rate than the full model.

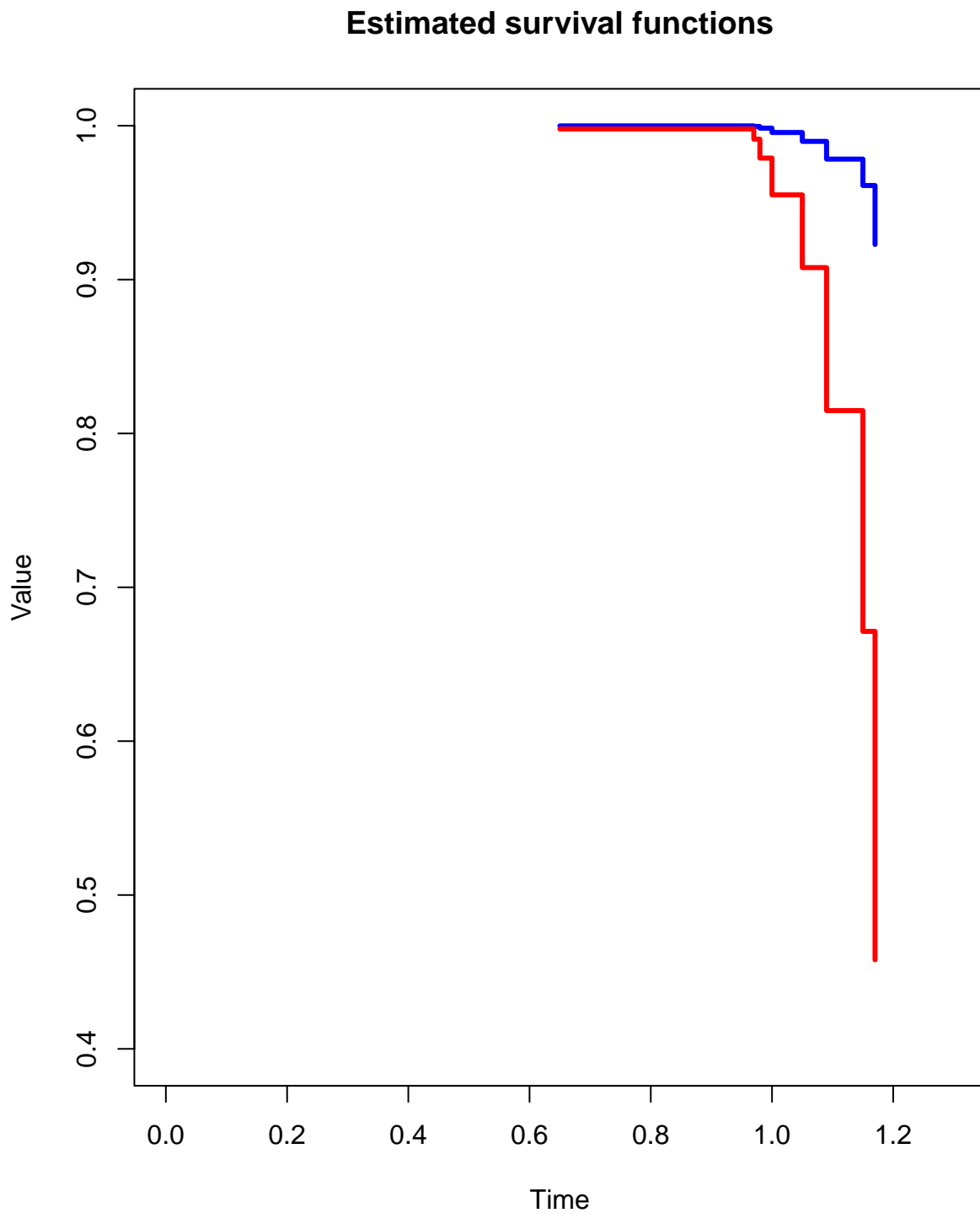


Figure 2: Here we have plotted the corresponding estimated survival functions for the two models. The blue line is for the full model, and the red is for the significant model. We see also here that the tires tend to fail at a faster rate than the full model.

c)

Now, using the R function *survreg*, we will redo the two analysis in a) using a Weibull regression model. Then we will transform the estimated parameter values so that they have the same interpretation as for the Cox regression models.

```
modwei <- survreg(Surv(Survival, Censor) ~ Age + Wedge + Inter +
  EB2B + Peel + Carbon + WxP, tire)
summary(modwei)
```

```
##
## Call:
## survreg(formula = Surv(Survival, Censor) ~ Age + Wedge + Inter +
##      EB2B + Peel + Carbon + WxP, data = tire)
##              Value Std. Error      z      p
## (Intercept) -5.1867      2.8477  -1.82 0.0686
## Age          -0.0560      0.0717  -0.78 0.4346
## Wedge         0.6181      0.2331   2.65 0.0080
## Inter         0.6779      0.3058   2.22 0.0266
## EB2B          0.7351      0.5293   1.39 0.1649
## Peel          1.9758      0.7806   2.53 0.0114
## Carbon        2.7978      2.0750   1.35 0.1775
## WxP           -1.2388      0.4759  -2.60 0.0092
## Log(scale)   -2.8196      0.2580 -10.93 <2e-16
##
## Scale= 0.0596
##
## Weibull distribution
## Loglik(model)= 10.1   Loglik(intercept only)= -5
##  Chisq= 30.03 on 7 degrees of freedom, p= 9.4e-05
## Number of Newton-Raphson Iterations: 11
## n= 34
```

```
modsignwei <- survreg(Surv(Survival, Censor) ~ Wedge + Inter +
  Peel + WxP, tire)
summary(modsignwei)
```

```
##
## Call:
## survreg(formula = Surv(Survival, Censor) ~ Wedge + Inter + Peel +
##      WxP, data = tire)
##              Value Std. Error      z      p
## (Intercept) -1.360      0.316  -4.30 1.7e-05
## Wedge        0.571      0.184   3.10 0.00195
## Inter        0.468      0.145   3.23 0.00122
```



```
## Peel          1.627          0.452   3.60 0.00032
## WxP           -1.094          0.292  -3.74 0.00018
## Log(scale)   -2.818          0.246 -11.44 < 2e-16
##
## Scale= 0.0597
##
## Weibull distribution
## Loglik(model)= 6.6   Loglik(intercept only)= -5
##  Chisq= 23.16 on 4 degrees of freedom, p= 0.00012
## Number of Newton-Raphson Iterations: 10
## n= 34
```

```
weib.coef.ph <- -modwei$coef[2:8]/modwei$scale
coxph.coef <- mod$coef
weib.coef.ph.sign <- -modsignwei$coef[2:5]/modsignwei$scale
coxph.coef.sign <- modsign$coef

data.frame(weib.coef.ph, coxph.coef)
```

```
##      weib.coef.ph coxph.coef
## Age      0.9396072   2.116461
## Wedge    -10.3649662 -10.368572
## Inter    -11.3681128 -11.304487
## EB2B     -12.3272315 -14.051701
## Peel     -33.1336060 -36.206472
## Carbon   -46.9191863 -54.846595
## WxP       20.7746273  22.344167
```

```
data.frame(weib.coef.ph.sign, coxph.coef.sign)
```

```
##      weib.coef.ph.sign coxph.coef.sign
## Wedge      -9.557762    -10.072221
## Inter      -7.837299     -7.331024
## Peel     -27.254334    -28.677270
## WxP       18.317976     19.190982
```

We see that in the full models, all the estimated values are pretty similar except for Age and Carbon, but in the significant models they are all very similar.

d)

The estimated survival functions formula is given by,

$$\begin{aligned} S(t) &= e^{-\int_0^t \alpha(s) ds} \\ &= e^{-\int_0^t \alpha_0(s) r(\beta) ds} \\ &= e^{-r(\beta) \int_0^t \alpha_0(s) ds} \\ &= e^{-r(\beta) H(t)} \\ &= e^{-r(\beta) \cdot (-\log(1-F(t)))} \\ &= e^{r(\beta) \cdot \log(1-F(t))} \\ &= e^{\beta^T} \cdot \text{pweibull}(t, \text{shape}, \text{scale}, \text{lower}=F, \text{log}=T) \end{aligned}$$

where $H(t)$ is the baseline hazard rate and is found in the documentation of the Weibull functions in R. This can be seen plotted in Figure 3 for both the full model and the significant model.

```
t = seq(0, 1.3, 0.01)
plot(t, exp(exp(sum(weib.coef.ph)) * pweibull(t, shape = 1/(modwei$scale),
  scale = exp(as.numeric(modwei$coef[1])), lower = F, log = T)),
  type = "l", ylim = c(0, 1), col = "blue", main = "Survival functions for the Weibull",
  xlab = "Time", ylab = "Value")
lines(t, exp(exp(sum(weib.coef.ph.sign)) * pweibull(t, shape = 1/(modsignwei$scale),
  scale = exp(as.numeric(modsignwei$coef[1])), lower = F, log = T)),
  col = "red")
```

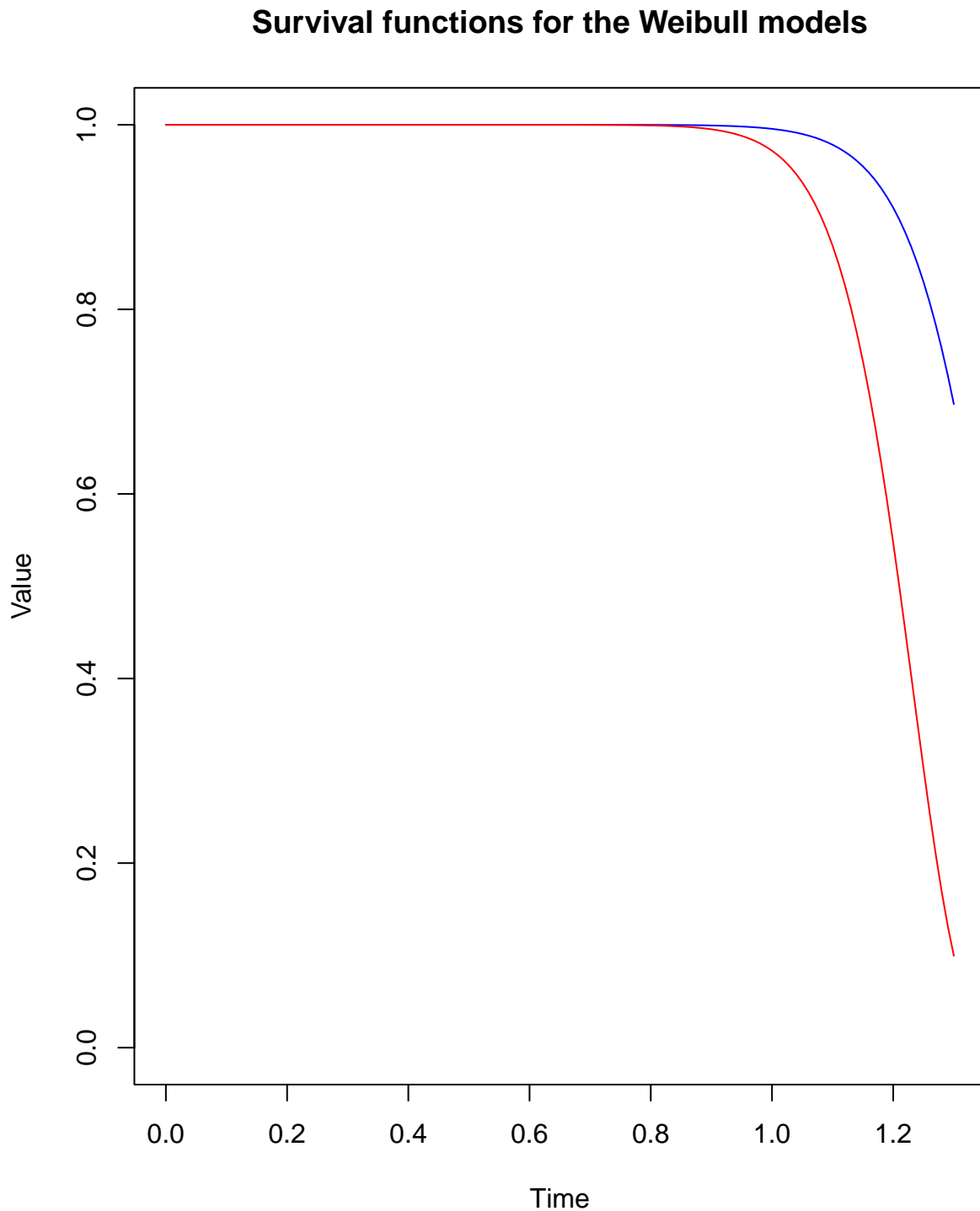


Figure 3: Here we have plotted the estimated survival functions for the two Weibull models. The blue line is for the full model, and the red is for the significant model. We see also here that the tires tend to fail at a faster rate using the significant model.

The martingale residuals are given by,

$$\widehat{M}_i = \widehat{M}_i(\tau) = N_i(\tau) - \widehat{\Lambda}_i(\tau), \quad (1.4)$$

where $\widehat{\Lambda}_i(t)$, the estimated cumulative intensity process, is given by,

$$\widehat{\Lambda}_i(\tau) = \sum_{T_j \leq \tau} \frac{Y_i(T_j) r(\widehat{\beta}, \mathbf{x}_i(T_j))}{\sum_{l \in \mathcal{R}_j} r(\widehat{\beta}, \mathbf{x}_l(T_j))}. \quad (1.5)$$

I struggled with writing this into code, but I want to explain that I understand the formula. We are finding the martingale residuals to the 11 uncensored individuals in the dataset, i.e. the i 's. $N_i(\tau) = 1$ when $\tau = 1.3$, because all the tires have failed before that time. $Y_i(T_j)$ is 1 right up to the moment tire i fails at one of the T_j 's, so we don't have to add the rest after the survival time of that individual, i.e. τ in Eq. 1.5 becomes the failure time of individual i . The denominator is handled in the same way that I handled the denominator in the Breslow estimator. I will include the code below to show what I have.

```
martingale_resid <- function(tireo, betas) {
  tireoc <- tireo[tireo$Censor == 1, ]$Survival
  Y <- as.data.frame(matrix(nrow = length(tireoc), ncol = length(unique(tireoc))))
  lambda <- c()

  for (individ in 1:length(tireoc)) {
    Amod <- c()
    for (i in unique(tireoc)) {
      Amoddenom <- c()
      denom <- c()
      tireotime <- tireo[tireo$Survival >= i, ]
      for (j in 1:nrow(tireotime)) {
        if (i %in% tireoc[duplicated(tireoc)]) {
          ties = c()
          for (k in 1:2) {
            ties <- c(ties, exp(sum(betas * c(as.numeric(tireotime[j,
              1:(ncol(tireotime) - 2)])))))
          }
          denom <- c(denom, sum(ties)/2)
        } else {
          denom <- c(denom, exp(sum(betas * c(as.numeric(tireotime[j,
            1:(ncol(tireotime) - 2)])))))
        }
      }
      Amoddenom <- c(Amoddenom, sum(denom))
      Amod <- c(Amod, 1/Amoddenom)
    }
  }

  lambda <- c(lambda, sum((exp((sum(c(t(betas) * as.numeric(tireo[tireo$Censor ==
```

```

        1, ][individ, 1:(ncol(tireo) - 2]])))) * Amod) *
        Y[individ, ]))
    }
    return(lambda)
}

# plot(1:11, 1 - martingale_resid(tireo, weib.coef.ph))

```

Now we will check if an exponential regression model give a good fit. The pdf of an exponential distribution is,

$$f(t; \lambda) = \lambda e^{-\lambda t},$$

and the pdf of a weibull distribution is,

$$f(t; a, \sigma) = \frac{a}{\sigma} \left(\frac{t}{\sigma} \right)^{a-1} e^{-\left(\frac{t}{\sigma} \right)^a}.$$

```

ourshape <- 1/(modwei$scale)
ourshapesign <- 1/(modsignwei$scale)
cat("a =", ourshape, "for the full model", "\na =", ourshapesign,
    "for the significant model")

## a = 16.76976 for the full model
## a = 16.74865 for the significant model

```

We see here that the exponential distribution is just a special case of the Weibull distribution when $a = 1$ and $\sigma = \frac{1}{\lambda}$. In our two cases, the shapes for the full and significant models are 16.77 and 16.75 respectively. So we can conclude with that the exponential distribution would not give a good fit because the shape, a , of our Weibull distribution is a lot bigger than 1.

e)

When using backward elimination process, we first begin with fitting the full model, and then subsequently removing the variable with the highest p-value, until we have only significant variables left (p-value smaller than 0.05).

```
modweielim1 <- survreg(Surv(Survival, Censor) ~ Age + Wedge +  
  Inter + EB2B + Peel + Carbon + WxP, tire)  
summary(modweielim1)$table
```

##		Value	Std. Error		z	p
##	(Intercept)	-5.18665280	2.84774030	-1.8213223	6.855788e-02	
##	Age	-0.05602984	0.07171521	-0.7812826	4.346363e-01	
##	Wedge	0.61807468	0.23306161	2.6519798	8.002133e-03	
##	Inter	0.67789344	0.30575589	2.2171068	2.661580e-02	
##	EB2B	0.73508678	0.52930931	1.3887660	1.649039e-01	
##	Peel	1.97579446	0.78062959	2.5310269	1.137291e-02	
##	Carbon	2.79784423	2.07499175	1.3483640	1.775413e-01	
##	WxP	-1.23881456	0.47589631	-2.6031186	9.237998e-03	
##	Log(scale)	-2.81957748	0.25797224	-10.9297707	8.305799e-28	

```
modweielim2 <- update(modweielim1, . ~ Wedge + Inter + EB2B +  
  Peel + Carbon + WxP)  
summary(modweielim2)$table
```

##		Value	Std. Error		z	p
##	(Intercept)	-5.8854917	2.8297908	-2.079833	3.754087e-02	
##	Wedge	0.7009827	0.2149054	3.261820	1.106994e-03	
##	Inter	0.7358756	0.3145812	2.339223	1.932390e-02	
##	EB2B	0.6057867	0.4664326	1.298766	1.940244e-01	
##	Peel	2.2313098	0.7854350	2.840858	4.499228e-03	
##	Carbon	3.3448884	2.0483723	1.632949	1.024796e-01	
##	WxP	-1.4071340	0.4660243	-3.019443	2.532397e-03	
##	Log(scale)	-2.7846417	0.2481281	-11.222595	3.158622e-29	

```
modweielim3 <- update(modweielim1, . ~ Wedge + Inter + Peel +  
  Carbon + WxP)  
summary(modweielim3)$table
```

##		Value	Std. Error		z	p
##	(Intercept)	-4.2518786	1.7523591	-2.426374	1.525054e-02	
##	Wedge	0.6153898	0.1787956	3.441863	5.777238e-04	
##	Inter	0.5549395	0.1604382	3.458898	5.423901e-04	
##	Peel	1.9023498	0.4594223	4.140743	3.461820e-05	

```
## Carbon      2.6742232  1.5481651   1.727350 8.410481e-02
## WxP         -1.2633457  0.2986307  -4.230461 2.332128e-05
## Log(scale) -2.8593035  0.2308582 -12.385538 3.129806e-35
```

```
modweielim4 <- update(modweielim1, . ~ Wedge + Inter + Peel +
  WxP)
summary(modweielim4)$table
```

```
##              Value Std. Error      z      p
## (Intercept) -1.3596818  0.3161733 -4.300433 1.704650e-05
## Wedge       0.5706588  0.1842206  3.097692 1.950340e-03
## Inter       0.4679363  0.1447041  3.233747 1.221777e-03
## Peel        1.6272560  0.4524421  3.596606 3.223957e-04
## WxP         -1.0936989  0.2923820 -3.740651 1.835445e-04
## Log(scale) -2.8183174  0.2462516 -11.444868 2.494828e-30
```

We find the same significant model as they do in Kristov et al (2002), which makes sense because they use the same method.

Problem 2:

Introduction

In this problem we will simulate many different datasets using the significant model we got in 1:e). We will later compare those datasets with the original.

a)

We will now make a function that simulates survival times from our model. We will simulate using the probability integral transform method. We have the survival function of the Weibull distribution,

$$S_i(t) = e^{-A(t)} = e^{-r(\beta; x)A_0(t)} = e^{-\exp(\beta^T x_i) \left(\frac{t_i}{\sigma}\right)^a},$$

where σ is the scale, and a is the shape. This gives us,

$$u_i = 1 - S_i(t) = 1 - e^{-\exp(\beta^T x_i) \left(\frac{t_i}{\sigma}\right)^a},$$

$$\left(\frac{t_i}{\sigma}\right)^a = -\frac{1}{\exp(\beta^T x_i)} \ln(1 - u_i),$$

$$t_i = \left(-\frac{\sigma^a}{\exp(\beta^T x_i)} \ln(1 - u_i) \right). \quad (2.1)$$

This equation is used to sample the new survival times. The censoring times will be samples using `rexp`. With rate 0.95 we get a similar amount of censored survival times to the original dataset, as seen below the function.

```
sim2a <- function(tire, modweielim4, seed = 98) {
  weib.coef.ph.sign <- -modsignwei$coef[2:5]/modsignwei$scale
  set.seed(seed)
  logu <- log(1 - runif(nrow(tire)))
  a <- 1/(modweielim4$scale)
  sigma <- exp(as.numeric(modweielim4$coef[1]))
  K <- c()
  for (i in 1:nrow(tire)) {
    K <- c(K, exp(sum(t(weib.coef.ph.sign) * as.numeric(tire[i,
      c(2, 3, 5, 7)]))))
  }
  # Eq. 2.1:
  t <- (-((sigma^a)/K) * logu)^(1/a)
  censoredtimes <- rexp(nrow(tire), 0.95)
  tildeTT <- c()
  D <- c()
  for (i in 1:nrow(tire)) {
```



```

      tildeTT[i] <- min(t[i], censoredtimes[i])
      ifelse(t[i] <= censoredtimes[i], D[i] <- 1, D[i] <- 0)
    }
    tiresim <- data.frame(Survival = t, Censor = D)
    return(tiresim)
  }
  d <- c()
  for (i in 1:500) {
    d <- c(d, sim2a(tire, modweielim4, i)$Censor)
  }
  cat("Number of censored times in the original dataset:  ", length(tire$Censor[tire$Censor == 1]), "\nNumber of censored times in 500 simulated datasets: ", length(d[d == 1])/500)

```

```

## Number of censored times in the original dataset:    23
## Number of censored times in 500 simulated datasets:  23.002

```

b)

Here we will make a function that automatically finds the significant model using the backward elimination process.

```

fun2b <- function(full_model) {
  model <- full_model
  finished <- F
  while (finished == F) {
    model_summary <- data.frame(t(summary(model)$table[, 4]))
    model_summary <- model_summary[, -c(1, ncol(model_summary))]
    if (max(model_summary) > 0.05) {
      model_summary <- model_summary[-max.col(model_summary)]
      model <- update(model, as.formula(paste(".*", paste(colnames(model_summary), collapse = "+"))))
    } else {
      final_model <- update(model, as.formula(paste(".*", paste(colnames(model_summary), collapse = "+"))))
      finished <- T
    }
  }
  return(summary(final_model)$table)
}

```

```

simdat1 <- tire
simdat1[, 8:9] <- sim2a(tire, modweielim4)
simbestreg1 <- fun2b(survreg(Surv(Survival, Censor) ~ Age + Wedge +
  Inter + EB2B + Peel + Carbon + WxP, simdat1))
mod1 <- survreg(as.formula(paste("Surv(Survival, Censor) ~",
  paste(rownames(simbestreg1[2:(nrow(simbestreg1) - 1), ]),
    collapse = "+"))), simdat1)
simreg1 <- summary(mod1)

```

The formula for the estimated intensity process is,

$$\hat{\lambda}_i(t) = Y_i(t)r(\hat{\beta}; x_i)\alpha_0(t),$$

for individual i . $\alpha_0(t) = \frac{a}{\sigma^a}t^{a-1}$ is the baseline hazard rate of a Weibull distribution. We will test the first, middle, and last case in the simulated dataset. Here i got pretty weird results, so bad in fact that they are uncomparable to the original hazard rate. Unfortunately don't have the time to find out what i did wrong.

```

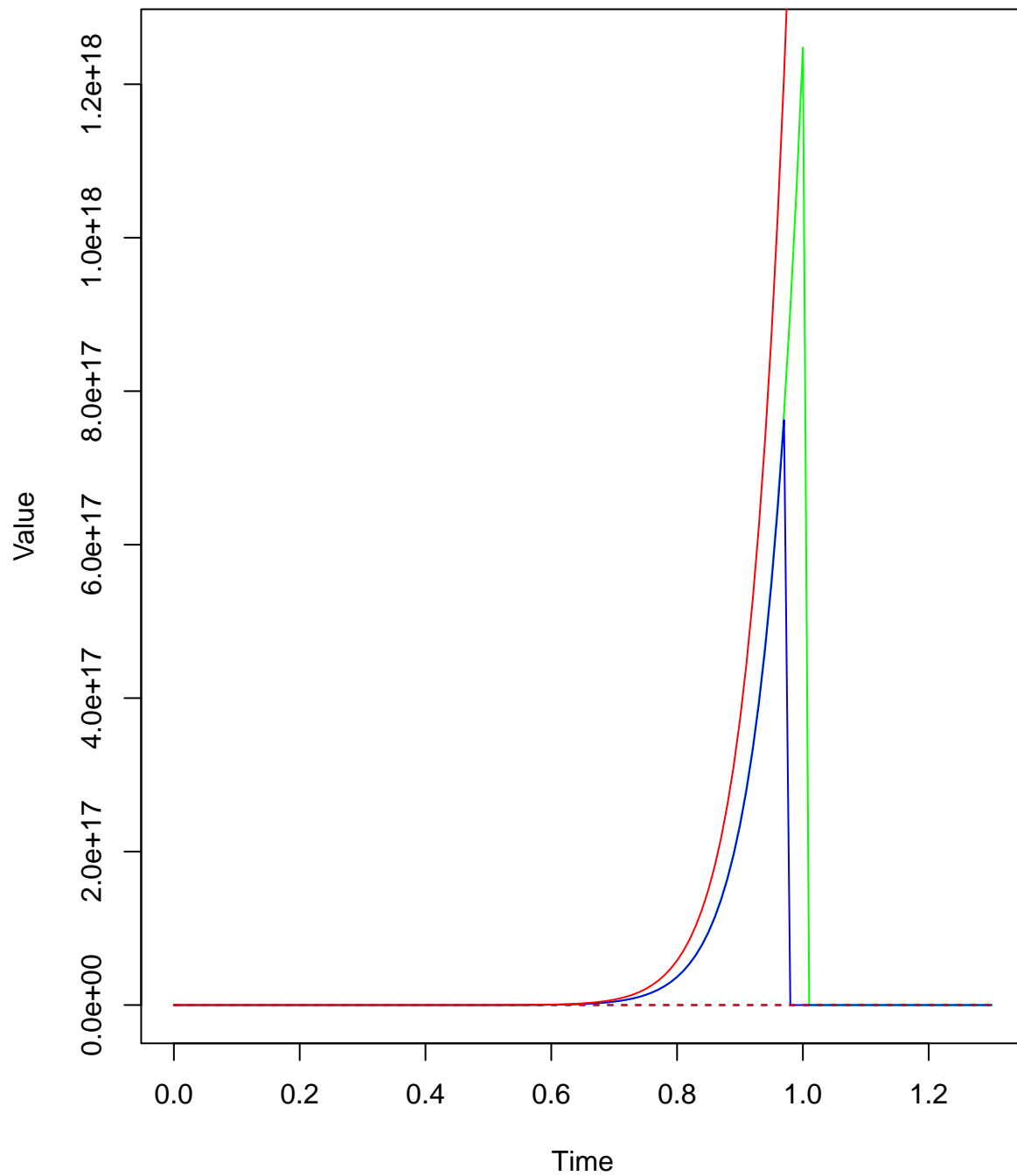
Y11 <- stepfun(simdat1$Survival[1], 1:0, right = T)
Y117 <- stepfun(simdat1$Survival[17], 1:0, right = T)
Y134 <- stepfun(simdat1$Survival[34], 1:0, right = T)
a1 <- 1/(mod1$scale)
sigma1 <- exp(as.numeric(mod1$coef[1]))
Y1 <- stepfun(tire$Survival[1], 1:0, right = T)
Y17 <- stepfun(tire$Survival[17], 1:0, right = T)
Y34 <- stepfun(tire$Survival[34], 1:0, right = T)
a <- 1/(modsignwei$scale)
sigma <- exp(as.numeric(modwei$coef[1]))

plot(t, Y11(t) * exp(sum(simreg1$coef[-1] * simdat1[1, c(2, 3,
  5, 7)])) * ((1/(mod1$scale))/(exp(as.numeric(mod1$coef[1]))^(1/(mod1$scale)))) *
  t^((1/(modwei$scale)) - 1), type = "l", col = "green", main = "Intensity processes",
  xlab = "Time", ylab = "Value")
lines(t, Y1(t) * exp(sum(modweielim4$coef[-1] * tire[1, c(2,
  3, 5, 7)])) * ((1/(modsignwei$scale))/(exp(as.numeric(modsignwei$coef[1]))^(1/(modsignwei$scale)))) *
  t^((1/(modsignwei$scale)) - 1), lty = 2, type = "l", col = "green")
lines(t, Y117(t) * exp(sum(simreg1$coef[-1] * simdat1[17, c(2,
  3, 5, 7)])) * ((1/(mod1$scale))/(exp(as.numeric(mod1$coef[1]))^(1/(mod1$scale)))) *
  t^((1/(modwei$scale)) - 1), type = "l", col = "blue")
lines(t, Y17(t) * exp(sum(modweielim4$coef[-1] * tire[17, c(2,
  3, 5, 7)])) * ((1/(modsignwei$scale))/(exp(as.numeric(modsignwei$coef[1]))^(1/(modsignwei$scale)))) *
  t^((1/(modsignwei$scale)) - 1), lty = 2, type = "l", col = "blue")
lines(t, Y134(t) * exp(sum(simreg1$coef[-1] * simdat1[34, c(2,
  3, 5, 7)])) * ((1/(mod1$scale))/(exp(as.numeric(mod1$coef[1]))^(1/(mod1$scale)))) *
  t^((1/(modwei$scale)) - 1), type = "l", col = "red")

```

```
lines(t, Y34(t) * exp(sum(modweielim4$coef[-1] * tire[34, c(2,
  3, 5, 7)])) * ((1/(modsignwei$scale))/(exp(as.numeric(modsignwei$coef[1]))^(1/(modsi
  t^((1/(modsignwei$scale)) - 1), lty = 2, type = "l", col = "red")
```

Intensity processes



```

simdat2 <- tire
simdat2[, 8:9] <- sim2a(tire, modweielim4, 1)
simbestreg2 <- fun2b(survreg(Surv(Survival, Censor) ~ Age + Wedge +
  Inter + EB2B + Peel + Carbon + WxP, simdat2))
simreg2 <- summary(survreg(as.formula(paste("Surv(Survival, Censor) ~",
  paste(rownames(simbestreg2[2:(nrow(simbestreg2) - 1), ]),
    collapse = "+"))), simdat1))
simdat3 <- tire
simdat3[, 8:9] <- sim2a(tire, modweielim4, 2)
simbestreg3 <- fun2b(survreg(Surv(Survival, Censor) ~ Age + Wedge +
  Inter + EB2B + Peel + Carbon + WxP, simdat3))
simreg3 <- summary(survreg(as.formula(paste("Surv(Survival, Censor) ~",
  paste(rownames(simbestreg3[2:(nrow(simbestreg3) - 1), ]),
    collapse = "+"))), simdat1))

```

c)

Now to end this problem, we will simulate many datasets, find for each of the seven covariates the proportion of the fitted models that includes the covariate, and finally plot all the estimated survival functions for each of the fitted models. The plots can be found in Figure 4. We will here test 500 iterations.

```
t <- seq(0, 2, 0.01)
S <- as.data.frame(matrix(0, nrow = 500, ncol = 7))
colnames(S) <- c("Age", "Wedge", "Inter", "EB2B", "Peel", "Carbon",
  "WxP")
covariates <- c()
scales <- c()
intercept <- c()
datasets <- tire
it <- (1:514)[-c(15, 125, 154, 202, 203, 235, 296, 321, 328,
  352, 355, 374, 414, 463)]
l = 1
for (i in it) {
  datasets[, 8:9] <- sim2a(tire, modweielim4, i)
  simbestreg <- fun2b(survreg(Surv(Survival, Censor) ~ Age +
    Wedge + Inter + EB2B + Peel + Carbon + WxP, datasets))
  covariates <- c(covariates, rownames(simbestreg[-c(1, nrow(simbestreg)),
    ]))
  simreg <- (survreg(as.formula(paste("Surv(Survival, Censor) ~",
    paste(rownames(simbestreg[2:(nrow(simbestreg) - 1), ]),
      collapse = "+"))), datasets))
  beta <- -simreg$coef[-1]/simreg$scale
  scales <- c(scales, simreg$scale)
  intercept <- c(intercept, simreg$coef[1])
  for (k in 1:length(beta)) {
    if (names(beta)[k] == "Age") {
      S[l, 1] <- beta[k]
    } else if (names(beta)[k] == "Wedge") {
      S[l, 2] <- beta[k]
    } else if (names(beta)[k] == "Inter") {
      S[l, 3] <- beta[k]
    } else if (names(beta)[k] == "EB2B") {
      S[l, 4] <- beta[k]
    } else if (names(beta)[k] == "Peel") {
      S[l, 5] <- beta[k]
    } else if (names(beta)[k] == "Carbon") {
      S[l, 6] <- beta[k]
    } else if (names(beta)[k] == "WxP") {
      S[l, 7] <- beta[k]
    }
  }
}
```

```

    }
  }
  l = l + 1
}

```

```

cat("Age appeared in", paste(length(covariates[covariates ==
"Age"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nWedge appeared in", paste(length(covariates[covariates ==
"Wedge"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nInter appeared in", paste(length(covariates[covariates ==
"Inter"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nEB2B appeared in", paste(length(covariates[covariates ==
"EB2B"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nPeel appeared in", paste(length(covariates[covariates ==
"Peel"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nCarbon appeared in", paste(length(covariates[covariates ==
"Carbon"])/length(it) * 100, "%", sep = ""), "of the models,",
"\nWxP appeared in", paste(length(covariates[covariates ==
"WxP"])/length(it) * 100, "%", sep = ""), "of the models,")

```

```

## Age appeared in 12.6% of the models,
## Wedge appeared in 86% of the models,
## Inter appeared in 95.4% of the models,
## EB2B appeared in 11.2% of the models,
## Peel appeared in 100% of the models,
## Carbon appeared in 9.6% of the models,
## WxP appeared in 98.6% of the models,

```

This makes sense considering we are simulating from a distribution that is only dependent on Wedge, Inter, Peel, and WxP. We see that Peel is the most important covariate because it appears in all of our tested models.

```

plot(t, exp(exp(sum(S[1, ])) * pweibull(t, shape = 1/(scales[1]),
scale = exp(as.numeric(intercept[1])), lower = F, log = T)),
type = "l", ylim = c(0, 1), col = "blue", main = "Survival functions of all the data",
xlab = "time", ylab = "Value")
for (i in 2:500) {
  lines(t, exp(exp(sum(S[i, ])) * pweibull(t, shape = 1/(scales[i]),
scale = exp(as.numeric(intercept[i])), lower = F, log = T)),
col = "blue")
}
lines(t, exp(exp(sum(weib.coef.ph.sign)) * pweibull(t, shape = 1/(modsignwei$scale),
scale = exp(as.numeric(modsignwei$coef[1])), lower = F, log = T)),
lwd = 3, col = "red")

```

Survival functions of all the datasets

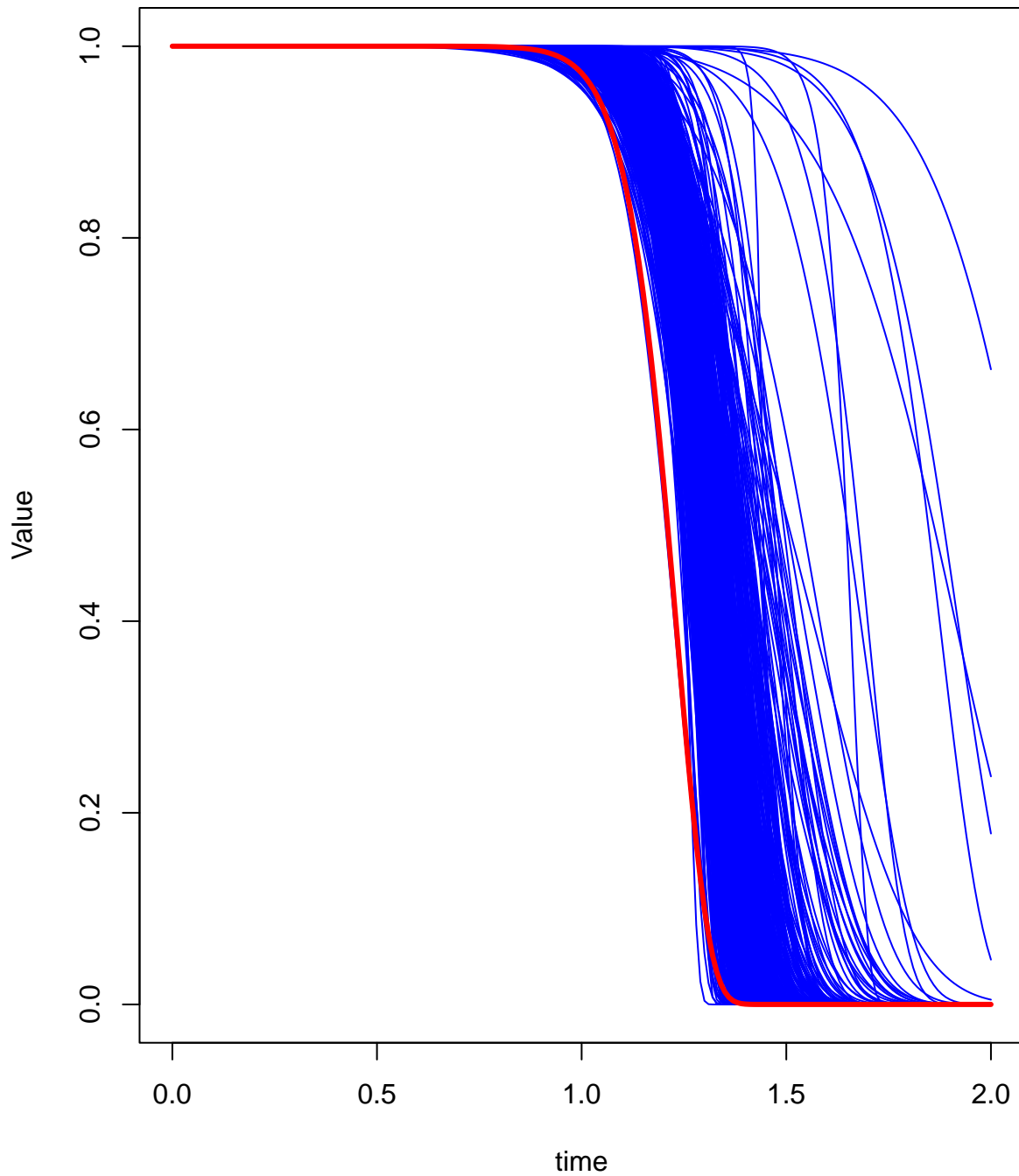


Figure 4: Here we see all the simulated survival functions. The red line is the survival function of the true dataset, while the blue lines are the survival functions of the simulated datasets. We can see that the simulated survival times are a bit skewed to the right, but not by much.