Survival Analysis Project

Contents

1 Introduction

As proposed in the description of the assignment, I choose to study the pbc dataset which is a biomedical dataset included into the R survival package. That data is based on a study on the disease "Primary Biliary Cholangitis" conducted by the Mayo Clinic during 12 years.

In addition to the dataset, I also used the references below to get an understanding of the data themselves (i.e. like domain experts in a "real" project):

- https://www.dovemed.com/diseases-conditions/primary-biliary-cirrhosis/
- https://en.wikipedia.org/wiki/Primary_biliary_cholangitis

2 Dataset Overview

The following command prints a basic description of the dataset:

```
help(pbc)
```

The dataset consists of the following fields:

- time : the Time To Event to be studied (in days),
- status : the status at the endpoint (censored, transplant or dead; i.e. "Event" occurred)),
- trt : treatment received by the patient, one of D-penicillamine, placebo or none,
- 17 other variables (e.g. sex, stage of the disease, a.s.o..).

The participant that received no treatment (trt=none above) did not participate in the clinical trial, but consented to have basic measurements recorded and to be followed for survival. Concretely speaking, it means that 8 of the 17 variables listed above are not available for them.

The command "`table(pbc$trt, exclude=NULL)`" indicates that :

- 158 participants received D-penicillamine,
- 154 participants received a placebo,
- 106 participants received no treatment (i.e. basic measurements).

The data set can be inspected through the following command:

```
str(pbc)
```

It shows that except for the sex variable, all categorical variables are represented as plain numerical values and not R factors (i.e. if used "as is", R will not interpret them as categorical variables).

3 Dataset Preparation (Categorical Variables)

The following code converts time to event from days to years and turns all categorical variables to R factors with “human readable” labels :

```
dat.full = pbc
dat.full$timeYears <- dat.full$time / 365.25

# convert factor variables
dat.full$ascites <- factor(dat.full$ascites)
dat.full$edema   <- factor(dat.full$edema  )
dat.full$hepato  <- factor(dat.full$hepato )
dat.full$spiders <- factor(dat.full$spiders)
dat.full$stage   <- factor(dat.full$stage  )
dat.full$status  <- factor(dat.full$status )
dat.full$trt     <- factor(dat.full$trt    )

dat.full$sex  <- factor(dat.full$sex, levels=c('m','f'), labels = c("male", "female"))

levels(dat.full$ascites)[levels(dat.full$ascites)=="0"] <- "absence"
levels(dat.full$ascites)[levels(dat.full$ascites)=="1"] <- "presence"

levels(dat.full$hepato)[levels(dat.full$hepato)=="0"] <- "absence"
levels(dat.full$hepato)[levels(dat.full$hepato)=="1"] <- "presence"

levels(dat.full$spiders)[levels(dat.full$spiders)=="0"] <- "absence"
levels(dat.full$spiders)[levels(dat.full$spiders)=="1"] <- "presence"

levels(dat.full$edema)[levels(dat.full$edema)=="0"] <- "none"
levels(dat.full$edema)[levels(dat.full$edema)=="0.5"] <- "managed"
levels(dat.full$edema)[levels(dat.full$edema)=="1"] <- "edema"

levels(dat.full$status)[levels(dat.full$status)=="0"] <- "censored"
levels(dat.full$status)[levels(dat.full$status)=="1"] <- "transplant"
levels(dat.full$status)[levels(dat.full$status)=="2"] <- "dead"

levels(dat.full$trt)[levels(dat.full$trt)=="1"] <- "D-penicillamine"
levels(dat.full$trt)[levels(dat.full$trt)=="2"] <- "placebo"

# mark patients outside of the trial as “control”
tmp <- addNA(dat.full$trt)
levels(tmp) <- c(levels(dat.full$trt), "control")
dat.full$trt <- tmp
remove("tmp")

# transplant are considered as censored
dat.full$event <- 0 + (dat.full$status == "dead")

# boolean indicating if the observation is in the trial or not
dat.full$trial=(dat.full$trt!="control")
```

This code also creates a binary event indicator (0 = censored, 1 = event occurred) that will be used as an input to all survival analysis functions.

Note: transplant patients are considered as censored as they quit the trial without having experienced the event.

Finally we perform a "sanity check" on the time to event and censoring information:

```
> summary(dat.full$status)
  censored transplant       dead
       232          25        161
> summary(dat.full$timeYears)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1123  2.9920  4.7360  5.2510  7.1550 13.1300
```

As there are no NA, the labels are those defined at the previous step, and the time are positive ranging from 0.11 to 13.13 years, we can start the survival analysis.

4 Descriptive statistics

The following commands prints some descriptive statistics about the dataset:

```
> summary(dat.full)
       id              time             status
 Min.   :  1.0   Min.   :  41   censored  :232
 1st Qu.:105.2   1st Qu.:1093   transplant: 25
 Median :209.5   Median :1730   dead      :161
 Mean   :209.5   Mean   :1918
 3rd Qu.:313.8   3rd Qu.:2614
 Max.   :418.0   Max.   :4795

              trt           age            sex          ascites
 D-penicillamine:158   Min.   :26.28   male  : 44   absence :288
 placebo        :154   1st Qu.:42.83   female:374   presence: 24
 control        :106   Median :51.00                NA's    :106
                       Mean   :50.74
                       3rd Qu.:58.24
                       Max.   :78.44

     hepato        spiders        edema           bili
 absence :152   absence :222   none   :354   Min.   : 0.300
 presence:160   presence: 90   managed: 44   1st Qu.: 0.800
 NA's    :106   NA's    :106   edema  : 20   Median : 1.400
                                             Mean   : 3.221
                                             3rd Qu.: 3.400
                                             Max.   :28.000

     chol           albumin          copper          alk.phos
 Min.   : 120.0   Min.   :1.960   Min.   :  4.00   Min.   :  289.0
 1st Qu.: 249.5   1st Qu.:3.243   1st Qu.: 41.25   1st Qu.:  871.5
 Median : 309.5   Median :3.530   Median : 73.00   Median : 1259.0
 Mean   : 369.5   Mean   :3.497   Mean   : 97.65   Mean   : 1982.7
 3rd Qu.: 400.0   3rd Qu.:3.770   3rd Qu.:123.00   3rd Qu.: 1980.0
 Max.   :1775.0   Max.   :4.640   Max.   :588.00   Max.   :13862.4
 NA's   :134                      NA's   :108      NA's   :106

     ast              trig           platelet        protime
 Min.   : 26.35   Min.   : 33.00   Min.   : 62.0   Min.   : 9.00
 1st Qu.: 80.60   1st Qu.: 84.25   1st Qu.:188.5   1st Qu.:10.00
 Median :114.70   Median :108.00   Median :251.0   Median :10.60
 Mean   :122.56   Mean   :124.70   Mean   :257.0   Mean   :10.73
 3rd Qu.:151.90   3rd Qu.:151.00   3rd Qu.:318.0   3rd Qu.:11.10
 Max.   :457.25   Max.   :598.00   Max.   :721.0   Max.   :18.00
 NA's   :106      NA's   :136      NA's   :11      NA's   :2

  stage       timeYears          event          trial
 1   : 21   Min.   : 0.1123   Min.   :0.0000   Mode :logical
 2   : 92   1st Qu.: 2.9918   1st Qu.:0.0000   FALSE:106
 3   :155   Median : 4.7365   Median :0.0000   TRUE :312
 4   :144   Mean   : 5.2506   Mean   :0.3852   NA's :0
 NA's:  6   3rd Qu.: 7.1554   3rd Qu.:1.0000
            Max.   :13.1280   Max.   :1.0000
```

5 Survival Time (Kaplan-Meier Estimator)

5.1 Overall Survival Time

The following commands prints the overall survival time using the Kaplan-Meier estimator:

```
> fit.KM=survfit(Surv(timeYears, event) ~ 1, data = dat.full)
> fit.KM
```

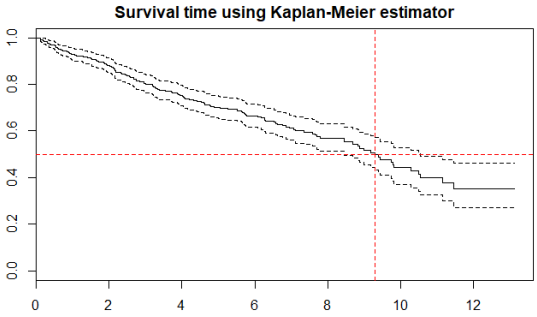
```
Call: survfit(formula = Surv(timeYears, event) ~ 1, data = dat.full)

      n  events  median 0.95LCL 0.95UCL
 418.00  161.00    9.30    8.46   10.55
```

It reads out as follows:

- There is a total of 418 observations out of which 161 experienced the event,
- the median survival time (time at which the probability of experiencing the event is of 50%) is 9.3 years,
- The lower bound of the 95% confidence interval of the median survival time is 8.46 years and its upper bound is 10.55 years.

It can be confirmed graphically using the following commands:

```
> plot(fit.KM)
> abline(v=9.3, col=2, lty=2)
> abline(h=0.5, col=2, lty=2)
```



Note:

- the Nelson-Aalen estimator (obtained by adding type = "fh" to the survfit command above) gives very close results and are therefore not printed here,
- changing the confidence level to to log-log (using the parameter conftype = "log-log") also does not cause any significant changes to the confidence level boundaries.

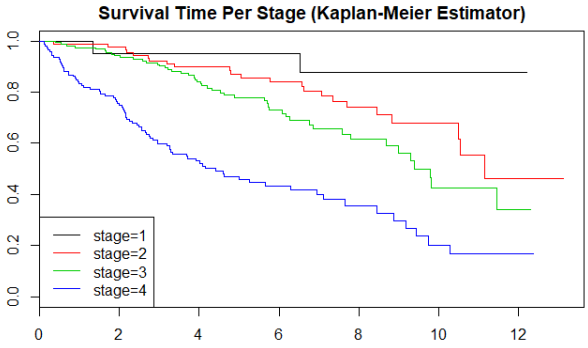
The following command prints the overall survival rate at the end of the study (along with it 95% confidence interval). After 12 years, 161 participants have experienced the event, 9 hasn't and the rest has been censored (418 - 9 - 161).

```
> summary(fit.KM, time=12)
Call: survfit(formula = Surv(timeYears, event) ~ 1, data = dat.full)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   12      9     161    0.353  0.0488         0.27        0.463
```

5.2 Survival Time By Stage

The following commands plot the survival time by stage using the Kaplan-Meier estimator:

```
fit.stage=survfit(Surv(timeYears, event) ~ stage, data = dat.full)
plot(fit.stage, col = 1:4)
legend("bottomleft", lty = 1, col = 1:4, legend = names(fit.stage$strata))
title(main="Survival Time Per Stage (Kaplan-Meier Estimator)")
```



It can be observed that the stage of the disease clearly separate the survival times.

6 Logrank Tests

6.1 Logrank Test According To Treatment

We can check the difference of survival time according the treatment which is represented as a categorical variable that can take one of the following values :

- D-penicillamine: patient received the treatment,
- placebo : people received a placebo,
- control : people outside of the trial but that are followed for comparison.

The R survdiff command compares the survival time of 2 (or more) groups using a statistical test where the null hypothesis is that there is no significant difference in the survival time between the groups.

In order to prove that the survival time of the groups are significantly different, the null hypothesis needs to be rejected, which mean that the p value of the test needs to be small, typically of a few percent at most.

The result of the test can be confirmed by plotting the survival times of each group on a single graph.

```
survdiff(Surv(timeYears, event) ~ trt, data = subset(dat.full, dat.full$trt!="control"))
```

```
Call:
survdiff(formula = Surv(timeYears, event) ~ trt, data = subset(dat.full,     dat.full$trt != "control"))

                       N Observed Expected (O-E)^2/E (O-E)^2/V
trt=D-penicillamine 158       65     63.2    0.0502     0.102
trt=placebo         154       60     61.8    0.0513     0.102

 Chisq= 0.1  on 1 degrees of freedom, p= 0.75
```

```
survdiff(Surv(timeYears, event) ~ trt, data = subset(dat.full, dat.full$trt!="placebo"))
```

```
Call:
survdiff(formula = Surv(timeYears, event) ~ trt, data = subset(dat.full, dat.full$trt != "placebo"))

                       N Observed Expected (O-E)^2/E (O-E)^2/V
trt=D-penicillamine 158       65     65.9    0.0128    0.0374
trt=control         106       36     35.1    0.0241    0.0374

 Chisq= 0  on 1 degrees of freedom, p= 0.847
```

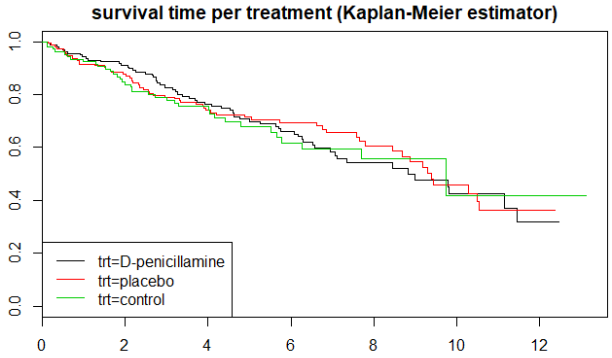
```
survdiff(Surv(timeYears, event) ~ trt, data = subset(dat.full, dat.full$trt!="D-penicillamine"))
```

```
Call:
survdiff(formula = Surv(timeYears, event) ~ trt, data = subset(dat.full, dat.full$trt != "D-
penicillamine"))

              N Observed Expected (O-E)^2/E (O-E)^2/V
trt=placebo 154       60     61.3    0.0268    0.0752
trt=control 106       36     34.7    0.0473    0.0752

 Chisq= 0.1  on 1 degrees of freedom, p= 0.784
```

As all p values are large, we can conclude that the survival time of the groups are not significantly different.

This is confirmed by the following plot that displays the survival time of each group :

```
fit=survfit(Surv(timeYears, event) ~ trt, data = dat.full)
plot(fit, col=1:3)
legend("bottomleft", lty = 1, col = 1:3, legend = names(fit$strata))
title(main="survival time per treatment (Kaplan-Meier estimator)")
```



In the scope of this study, from a statistical standpoint, those tests seems to show that D-penicillamine is not a proper cure for the disease. In a real world situation, domain experts would be required to confirm that hypothesis.

6.2 Logrank Test According To "Spiders" Covariate

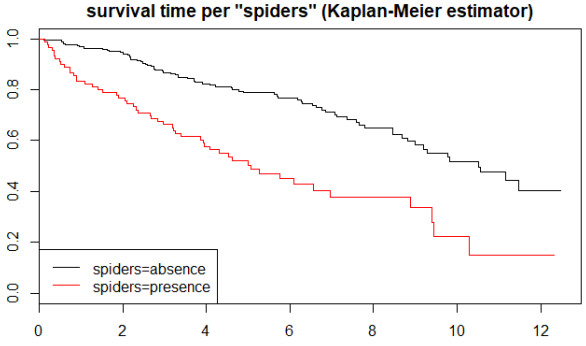
The spiders covariate is a categorical variable that separates well the survival time:

```
survdiff(Surv(timeYears, event) ~ spiders, data = dat.full[dat.full$trial,])
```

```
Call:
survdiff(formula = Surv(timeYears, event) ~ spiders, data = dat.full[dat.full$trial,
    ])

                  N Observed Expected (O-E)^2/E (O-E)^2/V
spiders=absence 222       73     98.1      6.43      30.3
spiders=presence 90       52     26.9     23.44      30.3

 Chisq= 30.3  on 1 degrees of freedom, p= 3.67e-08
```

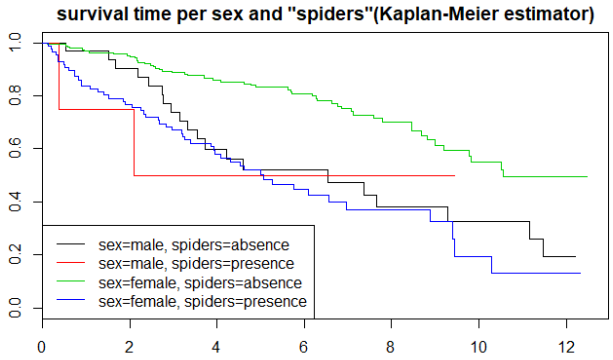
In this example, the groups are extremely well separated (p value is lower than 0.001) as this can be confirmed by plotting the survival time of each groups :

```
fit.spiders=survfit(Surv(timeYears, event) ~ spiders, data = dat.full[dat.full$trial,])
plot(fit.spiders, col = 1:2)
legend("bottomleft", lty = 1, col = 1:2, legend = names(fit.spiders$strata))
title(main='survival time per "spiders" (Kaplan-Meier estimator)')
```



Even if the logrank test above already proves that the 2 groups have different survival times, we can check if sex is a confounding factor:

```
fit.spiders=survfit(Surv(timeYears, event) ~ sex+spiders, data = dat.full[dat.full$trial,])
plot(fit.spiders, col = 1:4)
legend("bottomleft", lty = 1, col = 1:4, legend = names(fit.spiders$strata))
title(main='survival time per "spiders" (Kaplan-Meier estimator)')
```



As:

- the “absence” curve is above the “presence” curve for both the female and the male group,
- the 2 female curves (green and blue) are globally above the 2 male curves (black and red),

it suggests that the logrank test could be improved if stratified by the sex variable:

```
survdiff(Surv(timeYears, event) ~ spiders+strata(sex), data = dat.full[dat.full$trial,])
```

```
Call:
survdiff(formula = Surv(timeYears, event) ~ spiders + strata(sex), data = dat.full[dat.full$trial, ])

                  N Observed Expected (O-E)^2/E (O-E)^2/V
spiders=absence 222       73     99.3      6.99      35.3
spiders=presence 90       52     25.7     27.05      35.3

 Chisq= 35.3  on 1 degrees of freedom, p= 2.81e-09
```

As expected from the graphical analysis, we can see that adding a stratification to the logrank test causes the p value to decrease from 3.67e-08 to 2.81e-09.

7 Data Preparation (Semi Parametric Model)

In order to use semi parametric models, it is required that there are no missing values for all observations.

Out of the 418 observations (i.e. patients), 106 are only followed for basic measurements and for them 8 of the explanatory variables are not available. Consequently, those observations will not be used when building semi parametric models.

The code below displays the number of missing values in the remaining observations:

```
dat.trial = dat.full[dat.full$trial,]
dat.trial$trt = factor(dat.trial$trt)

for(v in names(dat.trial))
{
  nb_na = sum(is.na(dat.trial[[v]]))
  if (nb_na > 0)
  {
    cat(v, "=", nb_na, "\n")
  }
}
```

```
chol = 28
copper = 2
trig = 30
platelet = 4
```

The largest number (30) represents 9,6% of the total of the observations, which is significant but where it still make sense to approximate the missing values to be able to keep the variable.

7.1 Chol Variable

7.1.1 Filling The Missing Values Using A Linear Model

Since about 10% of the data are missing, a first attempt is made to check if it is possible to find a good approximation of the Chol variables from the other ones using a linear regression :

```
l=lm(chol_work~age+sex+ascites+hepato+spiders+edema+bili+albumin+copper+alk.phos+ast+trig
+platelet+protime+stage, data=dat.trial)
summary(l)
```

That attempt shows that only a few variables may properly explain the variable Chol (i.e. the variables that have a low p value as it allows to reject the null hypothesis that states the coefficient of the variable is equal to 0).

We can then build a second model only based on those variables :

```
l=lm(chol_work~edema+bili+ast+platelet, data=dat.trial)
summary(l)

# check noise gaussiannity
shapiro.test(l$residuals)

# check heteroschedasticity
plot(l$fitted.values, l$residuals)
```

```
Call:
lm(formula = chol_work ~ edema + bili + ast + platelet, data = dat.trial)

Residuals:
    Min      1Q  Median      3Q     Max
-544.15  -90.67  -35.21   49.52 1241.54

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    94.9623    45.9904   2.065 0.039880 *
edemamanaged -114.2614    41.8310  -2.731 0.006714 **
edemaedema   -241.2621    54.1936  -4.452 1.24e-05 ***
bili           21.7836     3.0634   7.111 1.00e-11 ***
ast             0.9231     0.2278   4.052 6.62e-05 ***
platelet        0.4351     0.1294   3.363 0.000882 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 195.2 on 274 degrees of freedom
  (32 observations deleted due to missingness)
Multiple R-squared:  0.3121,     Adjusted R-squared:  0.2995
F-statistic: 24.86 on 5 and 274 DF,  p-value: < 2.2e-16


       Shapiro-Wilk normality test

data:  l$residuals
W = 0.75706, p-value < 2.2e-16
```

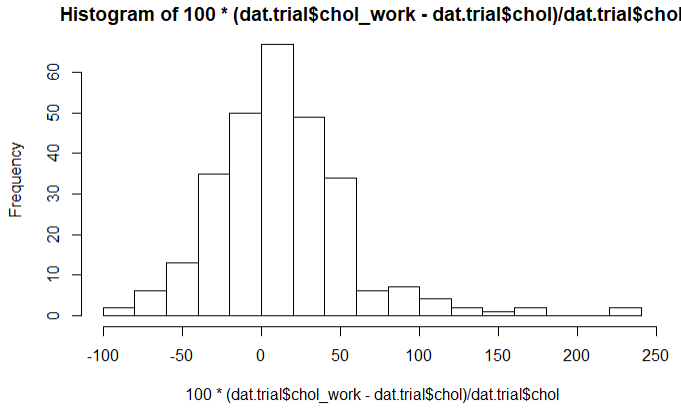
Despite low p values (for global F statistic and individual coefficients), that model is not good as the residuals are not Gaussian and the R squared value is not close to 1.

As a reminder of the linear model:

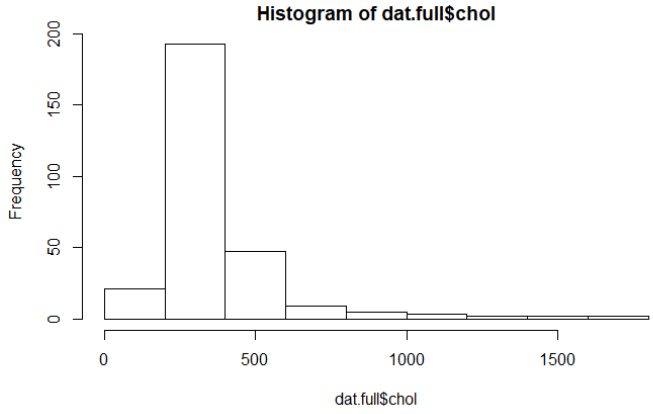
- The residuals should follow a Gaussian distribution (i.e. shapiro test should have a pvalue greater than 0.1),
- The residual should be heteroscedastic (the residuals should be homogeneously spread without any visible pattern),
- The R squared (and adjusted R squared) should be close to 1 (i.e. prediction should be close to the actual values),
- The p value of the F statistic should be low in order to reject the simplest model consisting of a single intercept value,
- The p value of each coefficient should be low in order to reject the hypothesis that this coefficient could be equal to 0 (i.e. taking that particular variable out of the model).

Below is the histogram of the prediction error in percentage:

```
dat.trial$chol_work=rep(NA, 312)
dat.trial$chol_work=predict(l, dat.trial)
hist(100*(dat.trial$chol_work-dat.trial$chol)/dat.trial$chol, breaks=12)
```



When compared to the distribution of the Chol variable, it does not make sense to opt for a complex model that exhibits an error from -50 to +50% compared to a simple median.



7.1.2 Filling The Missing Values Using The Median

Consequently, the median is preferred to fill the missing values of the Chol variable:

```
chol_median <- median(dat.trial$chol, na.rm=TRUE)
chol_median
chol_na <- is.na(dat.trial$chol)

dat.trial$chol_fxd = dat.trial$chol

dat.trial$chol_fxd[chol_na] <- chol_median
```

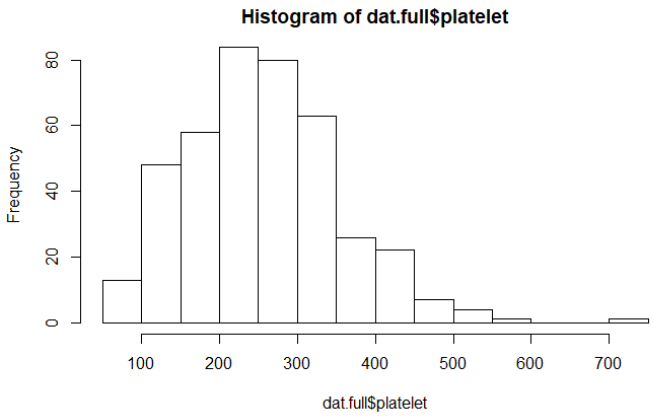
7.2 Trig Variable

Since about 10% of the data are missing for the Trig variable, a study similar to the Chol variable was made for the Trig variable but it also showed to that the median was the most appropriate way to fill the missing values :

```
trig_median <- median(dat.trial$trig, na.rm=TRUE)
trig_median
trig_na <- is.na(dat.trial$trig)

dat.trial$trig_fxd = dat.trial$trig

dat.trial$trig_fxd[trig_na] <- trig_median
```

7.3 Platelet Variable

Below is the histogram of the platelet variable:

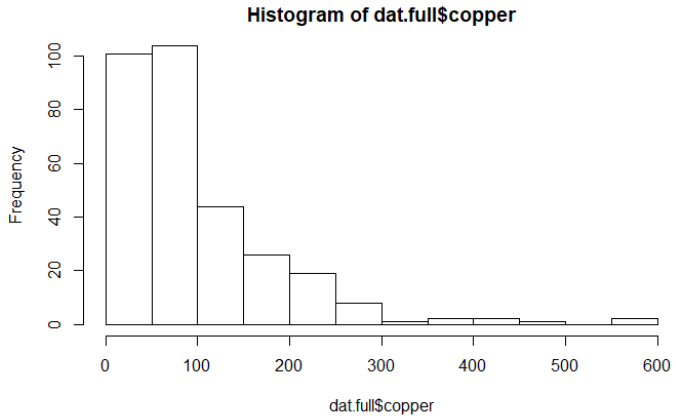


As less than 1% of the value are missing (4 out of 412) for that variable and that its distribution is Gaussian like, the median can be used to fill the missing values :

```
platelet_median <- median(dat.trial$platelet, na.rm=TRUE)
platelet_median
platelet_na <- is.na(dat.trial$platelet)

dat.trial$platelet_fxd = dat.trial$platelet

dat.trial$platelet_fxd[platelet_na] <- platelet_median
```

7.4 Copper Variable

Below is the histogram of the copper variable:



As less than 1% of the value are missing (2 out of 412) for that variable and its data is relatively grouped, the median can be used to fill the missing values:

```
copper_median <- median(dat.trial$copper, na.rm=TRUE)
copper_median
copper_na <- is.na(dat.trial$copper)

dat.trial$copper_fxd = dat.trial$copper

dat.trial$copper_fxd[copper_na] <- copper_median
```

8 Cox Proportional Hazards Models (Semi Parametric Models)

8.1 Automatic Model Selection

The code below automatically builds a model that minimizes the value of the AIC, starting from the model with all possible covariates (i.e. the "full" model) and dropping variables as long as the AIC value decreases (i.e. as long as the model improves).

```
M.full=coxph(Surv(timeYears, event) ~ trt+age+sex+ascites+hepato+spiders+edema+ bili+
        albumin+alk.phos+ast+protime+stage+chol_fxd+trig_fxd+ copper_fxd+platelet_fxd,
        data = dat.trial)

M.AIC <- step (M.full, trace=0)
summary(M.AIC)

fits <- list(M.full = M.full, M.AIC = M.AIC)
sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + bili +
    albumin + ast + protime + stage + copper_fxd, data = dat.trial)

  n= 312, number of events= 125

                   coef  exp(coef)   se(coef)      z Pr(>|z|)
age           0.0323646  1.0328940  0.0094917  3.410  0.00065 ***
edemamanaged  0.1168648  1.1239675  0.2861858  0.408  0.68301
edemaedema    0.9009883  2.4620350  0.3149228  2.861  0.00422 **
bili          0.0850507  1.0887723  0.0190612  4.462 8.12e-06 ***
albumin      -0.8044104  0.4473516  0.2584056 -3.113  0.00185 **
ast           0.0043078  1.0043171  0.0016708  2.578  0.00993 **
protime       0.3192571  1.3761050  0.1033727  3.088  0.00201 **
stage2        1.6434050  5.1727527  1.0768001  1.526  0.12696
stage3        1.9091057  6.7470523  1.0475320  1.822  0.06838 .
stage4        2.2671301  9.6516614  1.0413990  2.177  0.02948 *
copper_fxd    0.0026966  1.0027003  0.0009287  2.904  0.00369 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
age              1.0329     0.9682    1.0139    1.0523
edemamanaged     1.1240     0.8897    0.6414    1.9695
edemaedema       2.4620     0.4062    1.3281    4.5641
bili             1.0888     0.9185    1.0488    1.1302
albumin          0.4474     2.2354    0.2696    0.7423
ast              1.0043     0.9957    1.0010    1.0076
protime          1.3761     0.7267    1.1237    1.6852
stage2           5.1728     0.1933    0.6268   42.6870
stage3           6.7471     0.1482    0.8659   52.5745
stage4           9.6517     0.1036    1.2536   74.3092
copper_fxd       1.0027     0.9973    1.0009    1.0045

Concordance= 0.851  (se = 0.029 )
Rsquare= 0.47   (max possible= 0.983 )
Likelihood ratio test= 198  on 11 df,   p=0
Wald test            = 206.5  on 11 df,   p=0
Score (logrank) test = 327.3  on 11 df,   p=0

  M.full    M.AIC
1117.425  1103.95
```

As a preliminary step we check that the 2 global tests (Likelihood ratio test and Wald test) both reject their null hypothesis which is that all variables coefficients could be equal to 0. The test p values are both very low, which confirms that it is worth analyzing the other results.

The p value of the coefficients corresponds to the null hypothesis that states that the coefficient of the variable could be equal to 0 (i.e. do not have that variable in the model).

It can observed that:

- All continuous covariates has low p values that reject their null hypothesis,
- All categorical covariates has at least one low p value for their dummy variables.

Those observations confirm that all covariates are statistically significant in the model.

As a side note, it can also be observed that the dummy variables that do not have a low p value also have a very large confidence interval that typically cross 1, meaning that it cannot be strongly decided in which way the variable should contribute to the model (i.e. increase or decrease the hazard).

For each covariate, exp(coef) is the hazard ratio associated with one unit increase of the covariate:

- if exp(coef) is lower than 1, the hazard will decrease as the covariate increases,
- if exp(coef) is greater than 1, the hazard will increase as the covariate increases.

In the case of a categorical variable:

- the reference level is the one for which there is no dummy variable,
- the hazard ratio corresponds to moving from the reference level to the level of the dummy variable.

8.2 Schoenfeld Residuals

The code below tests for each covariate if its hazard is proportional or not (using a statistical test where the null hypothesis is hazard proportionally). If the p value is low, it means that hazard is not proportional.

This is an issue as the cox model is built on the assumption that each covariate impacts the hazard in a proportional manner.

Here are some possible ways to address that issue:

- If the covariate is categorical, consider using the variable as a stratification parameter rather than as a model parameter,
- If the covariate is continuous, find a transformation (i.e. function) that improves hazard proportionality, or alternatively consider splitting the covariate in classes of value and use a categorical variable instead,

Another potential way is to change the scope of the study and truncate it to the part of time where hazard is proportional (by censoring all events that happens outside of the new time window).

cox.zph(M.AIC)

```
                rho    chisq      p
age           0.00547  0.00393 0.9500
edemamanaged -0.18600  4.57611 0.0324
edemaedema   -0.04816  0.28541 0.5932
bili          0.17065  4.03611 0.0445
albumin      -0.01717  0.04612 0.8300
ast           0.01390  0.02077 0.8854
protime      -0.11390  1.76920 0.1835
stage2       -0.01802  0.04792 0.8267
stage3       -0.02566  0.09479 0.7582
stage4       -0.04735  0.31409 0.5752
copper_fxd   -0.00190  0.00042 0.9836
GLOBAL             NA 14.71145 0.1961
```

In the table above, we can see that those variables has a low p value :

- edemamananaged,
- bili.

We also know that in the medical domain, stratifying according to the severity of the disease is generally worth considering.

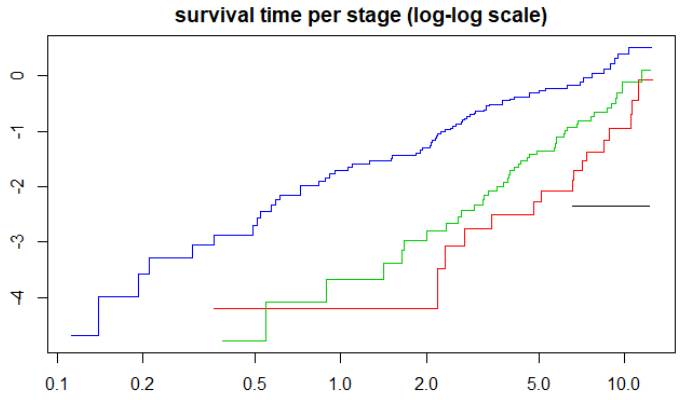
8.3 Stratifying According To The Stage Covariate

In this case, the severity of the disease is given by the stage covariate and even if the p values of its proportional tests are not low, the model itself do not exhibits good characteristics for the stage dummy variables (some of the p values are low and the confidence intervals are very large and cross 1).

Another way to check for hazard proportionality is to check if the survival curves for all level of the covariate are “parallel” in the log-log scale (i.e. compared to the reference level, all survival curves are identical but only shifted by a constant value equal to exp(coef)).

```
table(dat.trial$stage)

plot(survfit(Surv(timeYears, event) ~ stage, data = dat.trial), fun="cloglog", col = 1:4)
title(main='survival time per stage (log-log scale)')
```

```
  1   2   3   4
 16  67 120 109
```



The survival curves above do not have the same slope and their distance greatly varies over time, which indicates that their hazards functions are not proportional.

We can also note that the stage covariate also splits the observations in a relatively even manner (except for the first stage, but it still contain 5% of the observations).

Based on that, we try to change the stage covariate from a model covariate to a stratification one:

```
M.AIC_StrStage=coxph(Surv(timeYears, event) ~ age + edema + bili + albumin + ast + protime +
copper_fxd + strata(stage), data = dat.trial)
summary(M.AIC_StrStage)

fits <- list(M.full = M.full, M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage)
sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + bili +
   albumin + ast + protime + copper_fxd + strata(stage), data = dat.trial)

 n= 312, number of events= 125

                    coef  exp(coef)   se(coef)      z Pr(>|z|)
age            0.0320644  1.0325840  0.0095685  3.351 0.000805 ***
edemamanaged   0.1389586  1.1490765  0.2884608  0.482 0.630002
edemaedema     0.8202947  2.2711690  0.3176066  2.583 0.009802 **
bili           0.0941181  1.0986895  0.0205636  4.577 4.72e-06 ***
albumin       -0.7537003  0.4706219  0.2554608 -2.950 0.003174 **
ast            0.0037185  1.0037254  0.0017149  2.168 0.030128 *
protime        0.3032525  1.3542563  0.1036260  2.926 0.003429 **
copper_fxd     0.0025138  1.0025169  0.0009537  2.636 0.008394 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

              exp(coef) exp(-coef) lower .95 upper .95
age              1.0326     0.9684    1.0134    1.0521
edemamanaged     1.1491     0.8703    0.6528    2.0225
edemaedema       2.2712     0.4403    1.2187    4.2325
bili             1.0987     0.9102    1.0553    1.1439
albumin          0.4706     2.1248    0.2852    0.7765
ast              1.0037     0.9963    1.0004    1.0071
protime          1.3543     0.7384    1.1053    1.6592
copper_fxd       1.0025     0.9975    1.0006    1.0044

Concordance= 0.8  (se = 0.047 )
Rsquare= 0.352   (max possible= 0.958 )
Likelihood ratio test= 135.2  on 8 df,   p=0
Wald test            = 147.3  on 8 df,   p=0
Score (logrank) test = 204.5  on 8 df,   p=0

        M.full          M.AIC M.AIC_StrStage
     1117.4247      1103.9503       870.1467
```

The new model is statistically significant (except for one of the dummy variable of the edema covariate) and better than the previous one as its AIC is much lower.

Concretely speaking, it means that even if there is a unique set of coefficients for the model covariates there is one hazard baseline function per stage.

8.4 Stratifying By Other Covariates

8.4.1 Stratifying By Edema

One of the dummy variable of the edema categorical covariate shows poor statistical significance and poor hazard proportionality (edemamanaged in previous sections).

The code below adds the edema covariate as another stratification variable:

```
table(dat.trial$stage, dat.trial$edema)

M.AIC_StrStageEdema=coxph(Surv(timeYears, event) ~ age + bili + albumin + ast + protime +
copper_fxd + strata(stage, edema), data = dat.trial)
summary(M.AIC_StrStageEdema)

fits <- list(M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage,
M.AIC_StrStageEdema=M.AIC_StrStageEdema)
sapply(fits, AIC)
```

```
    none managed edema
  1   16       0     0
  2   62       4     1
  3  106      11     3
  4   79      14    16
Call:
coxph(formula = Surv(timeYears, event) ~ age + bili + albumin +
    ast + protime + copper_fxd + strata(stage, edema), data = dat.trial)

  n= 312, number of events= 125

                coef exp(coef)  se(coef)      z Pr(>|z|)
age         0.033781  1.034358  0.010102  3.344 0.000826 ***
bili        0.084188  1.087833  0.022377  3.762 0.000168 ***
albumin    -0.715172  0.489108  0.253928 -2.816 0.004856 **
ast         0.003813  1.003820  0.001722  2.214 0.026844 *
protime     0.345945  1.413325  0.106301  3.254 0.001136 **
copper_fxd  0.002419  1.002422  0.001007  2.401 0.016330 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

           exp(coef) exp(-coef) lower .95 upper .95
age           1.0344     0.9668    1.0141    1.0550
bili          1.0878     0.9193    1.0412    1.1366
albumin       0.4891     2.0445    0.2973    0.8045
ast           1.0038     0.9962    1.0004    1.0072
protime       1.4133     0.7076    1.1475    1.7407
copper_fxd    1.0024     0.9976    1.0004    1.0044

Concordance= 0.755  (se = 0.061 )
Rsquare= 0.241   (max possible= 0.92 )
Likelihood ratio test= 85.95  on 6 df,   p=2.22e-16
Wald test            = 88.9  on 6 df,   p=0
Score (logrank) test = 97.84  on 6 df,   p=0

             M.AIC      M.AIC_StrStage M.AIC_StrStageEdema
         1103.9503            870.1467            713.6171
```

Even if the model exhibits good statistical significance and a better AIC, it creates 12 different groups and for 5 of them there are either no data or very few observations (less than 5).

It means that the survival curves for those groups would be either impossible to build or based on a very low number of observations which could be an issue when performing prediction.

Even grouping the level “managed” and “edema” would no really improve the situation as there would still be 2 groups with either no data or very little observations (less than 5).

Consequently, it seems safer not to stratify by edema and keep it as a regular covariate.

8.4.2 Stratifying By Sex

The code below adds stratification by sex, which is a common practice in medical studies:

```
table(dat.trial$stage, dat.trial$sex)

M.AIC_StrStageSex=coxph(Surv(timeYears, event) ~ age + edema + bili + albumin + ast + protime +
copper_fxd + strata(stage, sex), data = dat.trial)
summary(M.AIC_StrStageSex)

fits <- list(M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage,
M.AIC_StrStageSex=M.AIC_StrStageSex)
sapply(fits, AIC)
```

```
    male female
  1    3     13
  2    6     61
  3   12    108
  4   15     94
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + bili +
    albumin + ast + protime + copper_fxd + strata(stage, sex),
    data = dat.trial)

  n= 312, number of events= 125

                   coef exp(coef)  se(coef)      z Pr(>|z|)
age            0.036021  1.036678  0.010672  3.375 0.000738 ***
edemamanaged   0.108358  1.114447  0.292633  0.370 0.711168
edemaedema     0.739274  2.094414  0.324238  2.280 0.022606 *
bili           0.100808  1.106064  0.021439  4.702 2.57e-06 ***
albumin       -0.779223  0.458762  0.257982 -3.020 0.002524 **
ast            0.003581  1.003587  0.001779  2.012 0.044188 *
protime        0.313600  1.368342  0.108830  2.882 0.003957 **
copper_fxd     0.002210  1.002213  0.001045  2.116 0.034342 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

              exp(coef) exp(-coef) lower .95 upper .95
age              1.0367     0.9646    1.0152    1.0586
edemamanaged     1.1144     0.8973    0.6280    1.9776
edemaedema       2.0944     0.4775    1.1094    3.9542
bili             1.1061     0.9041    1.0606    1.1535
albumin          0.4588     2.1798    0.2767    0.7606
ast              1.0036     0.9964    1.0001    1.0071
protime          1.3683     0.7308    1.1055    1.6937
copper_fxd       1.0022     0.9978    1.0002    1.0043

Concordance= 0.801  (se = 0.053 )
Rsquare= 0.342   (max possible= 0.94 )
Likelihood ratio test= 130.8  on 8 df,   p=0
Wald test            = 134.7  on 8 df,   p=0
Score (logrank) test = 194.3  on 8 df,   p=0

            M.AIC    M.AIC_StrStage M.AIC_StrStageSex
        1103.9503          870.1467          760.4019
```

Stratifying by sex gives similar results as stratifying by edema (good statistical model, better AIC) but again some groups with very few observations. For the same reasons as above, that stratification is not kept.

8.5 Bili Covariate

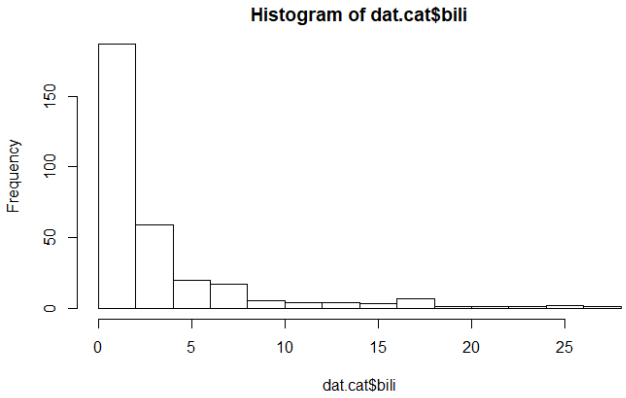
In the section above, the command cox.zph printed a very low p value for the bili covariate, which indicated that the covariate did not impact the hazard function in a proportional manner.

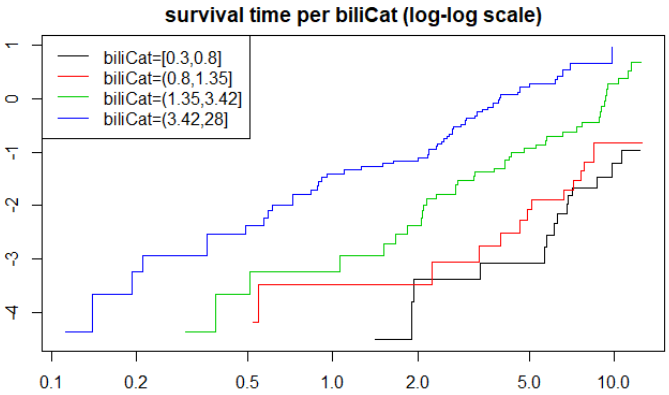
8.5.1 Changing The Continuous Bili Covariate To A Categorical Covariate

The code below displays the histogram of the continuous bili covariate, splits it into into 4 quartiles of approximately the same size and displays the survival curve according to those quartiles in the log-log scale.

```
hist(dat.trial$bili)

dat.trial$biliCat <- cut(dat.trial$bili, breaks=quantile(dat.trial$bili), include.lowest = TRUE)

table(dat.trial$biliCat)

fit.biliCat <- survfit(Surv(timeYears, event) ~ biliCat, data = dat.trial)
plot(fit.biliCat, fun="cloglog", col = 1:4)
legend("topleft", lty = 1, col = 1:4, legend = names(fit.biliCat$strata))
title(main='survival time per biliCat (log-log scale)')
```

```
[0.3,0.8]  (0.8,1.35] (1.35,3.42]   (3.42,28]
        90          66          78          78
```





The survival curves are not “parallel”, confirming that the hazard ratio is not proportional for that covariate.

The code below builds and displays the model where the continuous bili covariate is replaced by its categorical equivalent :

```
M.AIC_StrStage_BiliCat=coxph(Surv(timeYears, event) ~ age + edema + biliCat + albumin + ast +
protime + copper_fxd + strata(stage), data = dat.trial)
summary(M.AIC_StrStage_BiliCat)

fits <- list(M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage,
M.AIC_StrStage_BiliCat=M.AIC_StrStage_BiliCat)

sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + biliCat +
    albumin + ast + protime + copper_fxd + strata(stage), data =
dat.trial)

  n= 312, number of events= 125

                        coef exp(coef)  se(coef)      z Pr(>|z|)
age                 0.027649  1.028035  0.009292  2.976  0.00292 **
edemamanaged        0.333870  1.396362  0.287346  1.162  0.24527
edemaedema          0.964226  2.622756  0.313131  3.079  0.00207 **
biliCat(0.8,1.35]   0.142774  1.153469  0.389740  0.366  0.71412
biliCat(1.35,3.42]  1.005621  2.733604  0.333755  3.013  0.00259 **
biliCat(3.42,28]    1.636375  5.136514  0.364906  4.484 7.31e-06 ***
albumin            -0.793463  0.452276  0.248120 -3.198  0.00138 **
ast                 0.002641  1.002645  0.001767  1.495  0.13492
protime             0.326150  1.385623  0.105163  3.101  0.00193 **
copper_fxd          0.001937  1.001939  0.001003  1.931  0.05347 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    exp(coef) exp(-coef) lower .95 upper .95
age                    1.0280     0.9727    1.0095    1.0469
edemamanaged           1.3964     0.7161    0.7951    2.4524
edemaedema             2.6228     0.3813    1.4198    4.8450
biliCat(0.8,1.35]      1.1535     0.8669    0.5374    2.4760
biliCat(1.35,3.42]     2.7336     0.3658    1.4212    5.2581
biliCat(3.42,28]       5.1365     0.1947    2.5122   10.5021
albumin                0.4523     2.2110    0.2781    0.7355
ast                    1.0026     0.9974    0.9992    1.0061
protime                1.3856     0.7217    1.1275    1.7028
copper_fxd             1.0019     0.9981    1.0000    1.0039

Concordance= 0.8  (se = 0.047 )
Rsquare= 0.374   (max possible= 0.958 )
Likelihood ratio test= 145.9  on 10 df,   p=0
Wald test            = 139.7  on 10 df,   p=0
Score (logrank) test = 180.1  on 10 df,   p=0

                M.AIC         M.AIC_StrStage M.AIC_StrStage_BiliCat
            1103.9503               870.1467               863.4442
```

That new model has good statistical significance and better AIC.

On the survival curves, it can be observed that :

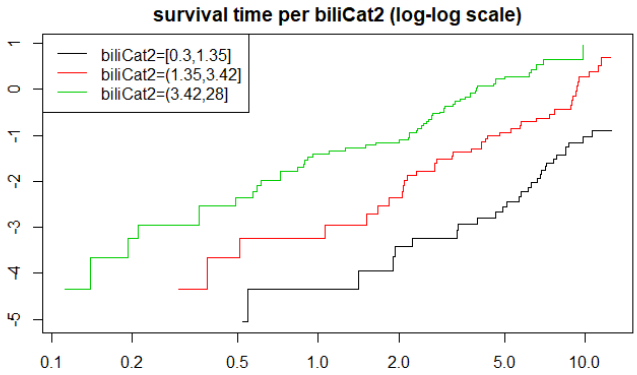
- The last 2 curves are “parallel”,
- the first 2 curves are not parallel to the other 2, but are close to each other and that their median would be also parallel to the other 2,

On the coxph output, it can be observed that the last 2 dummy variables has good statistics whereas the first one has a high p value (suggesting that the model would be better without).

Consequently, it makes sense to try to merge the first 2 quartiles into a single group.

```
biliQuant=quantile(dat.trial$bili)
dat.trial$biliCat2 <- cut(dat.trial$bili, breaks=biliQuant[-2], include.lowest = TRUE)

table(dat.trial$biliCat2)
fit.biliCat2 <- survfit(Surv(timeYears, event) ~ biliCat2, data = dat.trial)
plot(fit.biliCat2, fun="cloglog", col = 1:4)
legend("topleft", lty = 1, col = 1:3, legend = names(fit.biliCat2$strata))
title(main='survival time per biliCat2 (log-log scale)')
```

```
[0.3,1.35] (1.35,3.42]   (3.42,28]
       156          78          78
```



The 3 curves are now almost parallel, suggesting a proportional hazard ratio :

```
M.AIC_StrStage_BiliCat2=coxph(Surv(timeYears, event) ~ age + edema + biliCat2 + albumin + ast +
protime + copper_fxd + strata(stage), data = dat.trial)
summary(M.AIC_StrStage_BiliCat2)

cox.zph(M.AIC_StrStage_BiliCat2)

fits <- list(M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage,
M.AIC_StrStage_BiliCat2=M.AIC_StrStage_BiliCat2)

sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + biliCat2 +
    albumin + ast + protime + copper_fxd + strata(stage), data =
dat.trial)

  n= 312, number of events= 125

                          coef  exp(coef)   se(coef)      z Pr(>|z|)
age                   0.0274668  1.0278475  0.0092769  2.961 0.003069 **
edemamanaged          0.3359240  1.3992327  0.2872526  1.169 0.242227
edemaedema            0.9605428  2.6131146  0.3131704  3.067 0.002161 **
biliCat2(1.35,3.42]   0.9326028  2.5411145  0.2633297  3.542 0.000398 ***
biliCat2(3.42,28]     1.5625218  4.7708371  0.3005222  5.199    2e-07 ***
albumin              -0.7976869  0.4503695  0.2478485 -3.218 0.001289 **
ast                   0.0026495  1.0026530  0.0017666  1.500 0.133679
protime               0.3263342  1.3858785  0.1053337  3.098 0.001948 **
copper_fxd            0.0019677  1.0019697  0.0009997  1.968 0.049018 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                     exp(coef) exp(-coef) lower .95 upper .95
age                     1.0278     0.9729    1.0093     1.047
edemamanaged            1.3992     0.7147    0.7969     2.457
edemaedema              2.6131     0.3827    1.4144     4.828
biliCat2(1.35,3.42]     2.5411     0.3935    1.5166     4.258
biliCat2(3.42,28]       4.7708     0.2096    2.6472     8.598
albumin                 0.4504     2.2204    0.2771     0.732
ast                     1.0027     0.9974    0.9992     1.006
protime                 1.3859     0.7216    1.1274     1.704
copper_fxd              1.0020     0.9980    1.0000     1.004

Concordance= 0.799  (se = 0.047 )
Rsquare= 0.373   (max possible= 0.958 )
Likelihood ratio test= 145.8  on 9 df,   p=0
Wald test            = 139.6  on 9 df,   p=0
Score (logrank) test = 180  on 9 df,   p=0

                        rho  chisq      p
age                  0.0184 0.0429 0.8359
edemamanaged        -0.1505 3.1002 0.0783
edemaedema          -0.0531 0.3514 0.5533
biliCat2(1.35,3.42]  0.0497 0.3226 0.5701
biliCat2(3.42,28]    0.0881 1.1472 0.2841
albumin             -0.0494 0.3397 0.5600
ast                 -0.0418 0.1969 0.6572
protime             -0.1030 1.5133 0.2186
copper_fxd          -0.0878 0.9820 0.3217
GLOBAL                   NA 7.7405 0.5605
                M.AIC          M.AIC_StrStage M.AIC_StrStage_BiliCat2
            1103.9503               870.1467                861.5783
```

Indeed, this new model has better statistical significance for the bili covariate and better AIC confirming that it is worth merging the first 2 quartiles into a single group.

8.5.2 Applying A Transformation To The Bili Covariate

When looking at the histogram of the bili covariate, it can also be observed that its density decreases very rapidly and since its hazard ratio is not proportional, it suggests that it applying a log transformation to the variable could improve the model :

```
M.AIC_StrStage_BiliLog=coxph(Surv(timeYears, event) ~ age + edema + log(bili) + albumin + ast +
protime + copper_fxd + strata(stage), data = dat.trial)
summary(M.AIC_StrStage_BiliLog)

cox.zph(M.AIC_StrStage_BiliLog)

fits <- list(M.AIC = M.AIC, M.AIC_StrStage=M.AIC_StrStage,
M.AIC_StrStage_BiliLog=M.AIC_StrStage_BiliLog)

sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + log(bili) +
    albumin + ast + protime + copper_fxd + strata(stage), data =
dat.trial)

  n= 312, number of events= 125

                 coef exp(coef)  se(coef)      z Pr(>|z|)
age           0.031052  1.031539  0.009365  3.316 0.000914 ***
edemamanaged  0.213277  1.237727  0.285228  0.748 0.454616
edemaedema    0.810758  2.249613  0.313557  2.586 0.009719 **
log(bili)     0.705322  2.024499  0.125215  5.633 1.77e-08 ***
albumin      -0.700384  0.496395  0.251364 -2.786 0.005331 **
ast           0.002285  1.002288  0.001773  1.289 0.197546
protime       0.279478  1.322439  0.104896  2.664 0.007714 **
copper_fxd    0.001673  1.001674  0.001012  1.653 0.098392 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

             exp(coef) exp(-coef) lower .95 upper .95
age             1.0315     0.9694    1.0128    1.0506
edemamanaged    1.2377     0.8079    0.7077    2.1648
edemaedema      2.2496     0.4445    1.2168    4.1592
log(bili)       2.0245     0.4939    1.5839    2.5876
albumin         0.4964     2.0145    0.3033    0.8124
ast             1.0023     0.9977    0.9988    1.0058
protime         1.3224     0.7562    1.0767    1.6243
copper_fxd      1.0017     0.9983    0.9997    1.0037

Concordance= 0.804  (se = 0.047 )
Rsquare= 0.377   (max possible= 0.958 )
Likelihood ratio test= 147.8  on 8 df,   p=0
Wald test            = 146.1  on 8 df,   p=0
Score (logrank) test = 186.8  on 8 df,   p=0

                  rho   chisq      p
age           -0.0076 0.00739 0.9315
edemamanaged  -0.1737 3.93813 0.0472
edemaedema    -0.0772 0.72734 0.3937
log(bili)      0.1440 2.77530 0.0957
albumin       -0.0477 0.32440 0.5690
ast           -0.0215 0.04797 0.8266
protime       -0.1175 1.87167 0.1713
copper_fxd    -0.0651 0.52242 0.4698
GLOBAL            NA 9.23885 0.3225
               M.AIC         M.AIC_StrStage M.AIC_StrStage_BiliLog
            1103.9503               870.1467               857.6011
```

This model has good statistical significance and even better AIC than the categorical one. As such, applying a log transformation to the categorical covariate is the preferred solution.

9 AUC And ROC Curve

Thanks to the survivalROC R package, it is possible to compute the ROC curve (False Positive rate vs True Positive rate) and its associated AUC (Area Under the Curve) even with censoring information.

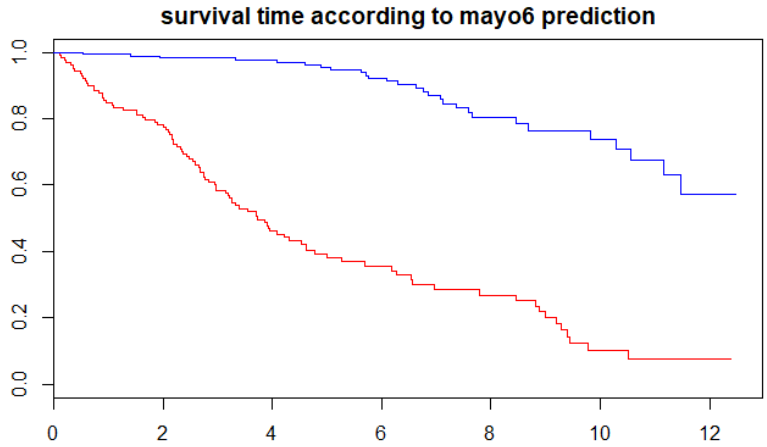
However, to compute a ROC curve it is required to have a single covariate. In order to have a single value for all observation, a simple semi parametric model is built based on the previous section:

```
M.full2=coxph(Surv(timeYears, event) ~ trt+age+sex+ascites+hepato+spiders+edema+
        log(bili)+albumin+alk.phos+ast+protime+stage+chol_fxd+trig_fxd+
        copper_fxd+platelet_fxd, data = dat.trial)
M.AIC3 <- step (M.full2, trace=0)
# summary(M.AIC3)

# model manually adjusted from M.AIC3
M.AIC4 <- coxph(Surv(timeYears, event) ~ age + edema + log(bili) + albumin + protime +
copper_fxd, data = dat.trial)
summary(M.AIC4)

fits <- list(M.full2 = M.full2, M.AIC3 = M.AIC3, M.AIC4 = M.AIC4)
sapply(fits, AIC)
```

```
Call:
coxph(formula = Surv(timeYears, event) ~ age + edema + log(bili) +
    albumin + protime + copper_fxd, data = dat.trial)

  n= 312, number of events= 125

                    coef  exp(coef)   se(coef)      z Pr(>|z|)
age            0.0311835  1.0316748  0.0086548  3.603 0.000315 ***
edemamanaged   0.1477070  1.1591732  0.2765198  0.534 0.593228
edemaedema     0.9087216  2.4811485  0.3081372  2.949 0.003187 **
log(bili)      0.8043185  2.2351727  0.1085080  7.413 1.24e-13 ***
albumin       -0.9296601  0.3946878  0.2404527 -3.866 0.000111 ***
protime        0.2497394  1.2836908  0.0870755  2.868 0.004130 **
copper_fxd     0.0020908  1.0020930  0.0009809  2.131 0.033054 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

              exp(coef) exp(-coef) lower .95 upper .95
age              1.0317     0.9693    1.0143    1.0493
edemamanaged     1.1592     0.8627    0.6742    1.9931
edemaedema       2.4811     0.4030    1.3563    4.5388
log(bili)        2.2352     0.4474    1.8070    2.7649
albumin          0.3947     2.5336    0.2464    0.6323
protime          1.2837     0.7790    1.0823    1.5226
copper_fxd       1.0021     0.9979    1.0002    1.0040

Concordance= 0.849  (se = 0.029 )
Rsquare= 0.48   (max possible= 0.983 )
Likelihood ratio test= 203.8  on 7 df,   p=0
Wald test            = 208.3  on 7 df,   p=0
Score (logrank) test = 298.5  on 7 df,   p=0

 M.full2   M.AIC3   M.AIC4
1104.479 1089.609 1090.084
```

The code below computes the single covariate for all observations (equal to the dot product of the covariates by the model coefficients), its associated survival ROC curve at a given point in time (5 years, arbitrary chosen) and separate low risk from high risk using a cutoff of 10% of false positive rate:

```
library(survivalROC)

dat.cov <- subset(dat.trial, select=c("age", "albumin", "protime", "copper_fxd"))

dat.cov$"log(bili)" <- log(dat.trial$bili)

dat.cov$edemamanaged <- 0+(dat.trial$edema=="managed")
dat.cov$edemaedema <- 0+(dat.trial$edema=="edema")

dat.cov <- dat.cov[, names(M.AIC4$coefficients)]

# consistency check
if ((length(names(M.AIC4$coefficients))==length(names(dat.cov))) &&
   all(names(M.AIC4$coefficients)==names(dat.cov))) {
  dat.trial$mayo6 = as.vector(M.AIC4$coefficients %*% t(dat.cov))

  ROC.6 <- survivalROC(Stime = dat.trial$timeYears,
                status = dat.trial$event,
                marker = dat.trial$mayo6,
                predict.time = 365.25 * 5,
                method="KM")
  cutoff <- with(ROC.6, min(cut.values[FP <= 0.10]))

  dat.trial$prediction <-
  ifelse(dat.trial$mayo6 <= cutoff,
        "low_risk", "high_risk")

  fit.KM <- survfit(Surv(timeYears, event) ~ prediction, data = dat.trial)
  plot(fit.KM, col = c("red", "blue"))
  title(main="survival time according to mayo6 prediction")
} else {
  cat("data error")
```



The survival time according to binary indicator derived from the synthetic covariate mayo6 allows to clearly separate the observations the observation between high risks and low risks.

Finally, the code below displays the ROC curve itself:

```
library(ggplot2)
ROC = data.frame(FP=ROC.6$FP,TP=ROC.6$TP)
ggplot(ROC, aes(FP, TP)) +
geom_line() +
theme_bw(base_size = 12)
```