Survival Analysis Final Project: Analysis of Patients with Primary Biliary Cirrhosis (PBC)

Christopher Rutherford

Fall 2019

Summary

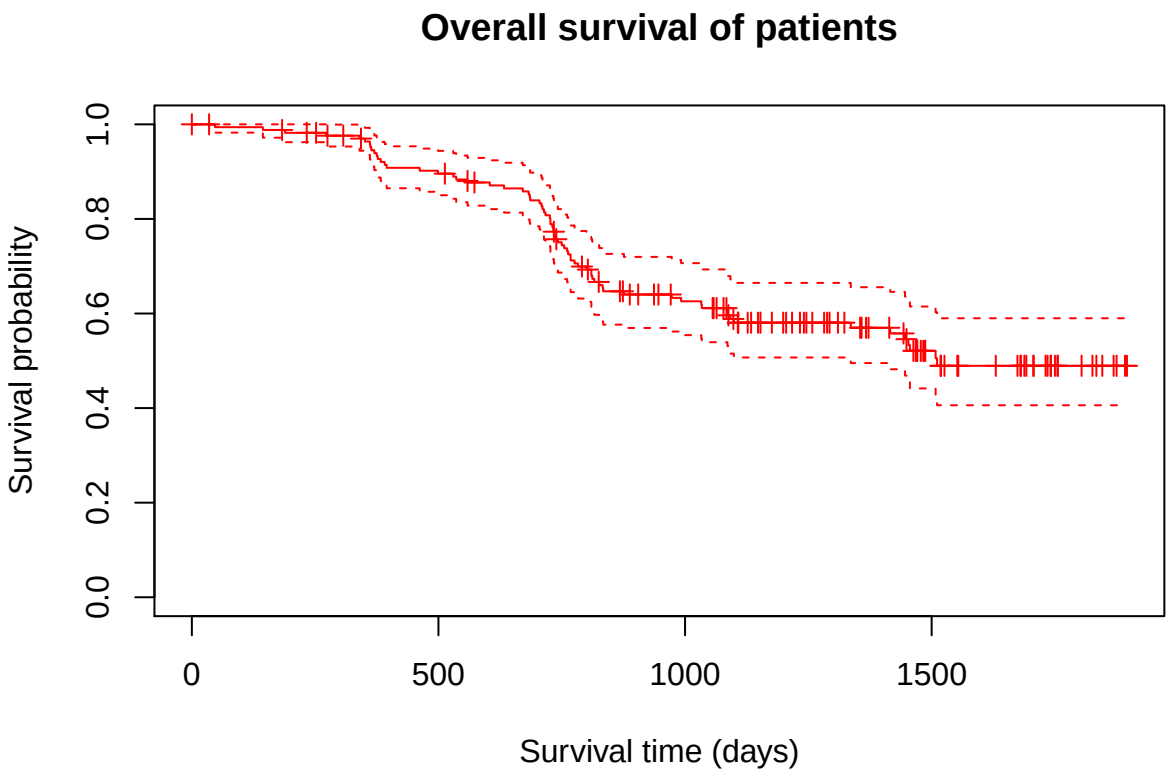
This study contains a sample of 170 patients who suffered from primary biliary cirrhosis, more recently known as primary biliary cholangitis (PBC). PBC is an autoimmune disease that slowly destroys the bile ducts in the liver overtime, possibly leading to cirrhosis and other problems throughout the body.1 The patients are split into two main groups – 86 who received a treatment of ursodeoxycholic acid (UDCA), and 84 who received a placebo. One patient who received UDCA is not included in the analysis, as they are censored at time 0. The study spans over a period of 1896 days. The covariates in this study include the treatment type, the stage of the disease, the bilirubin level at the study's start, and the patient's risk score. Bilirubin is a compound formed from the breakdown of red blood cells. The liver is responsible for filtering out this substance, so high levels typically indicate underlying problems with the liver. Of the 170 patients, 98 were censored - 39 placebo, 59 UDCA - while 72 had the event - 45 placebo, 27 UDCA. Statistical tests are considered significant if the P-value is below 0.05.

We begin by importing the data and plotting the survival curve.

```
udca_data<-udca1
attach(udca_data)
head(udca_data)
```

```
##   id trt stage bili riskscore futime status
## 1  1   1     1  1.0       5.1   1896      0
## 2  2   0     1  1.7       4.2   1456      1
## 3  3   1     0  0.5       3.4   1892      0
## 4  4   1     1  1.4       5.0    343      0
## 5  5   0     1  1.1       4.3   1875      0
## 6  6   1     1  1.4       5.9    768      1
```

```
udca_data$surv=Surv(time = futime,
                    event=status==1)
plot(x=udca_data$surv, col='red',
     main="Overall survival of patients",
     xlab="Survival time (days)", ylab="Survival probability",
     mark.time=T)
```



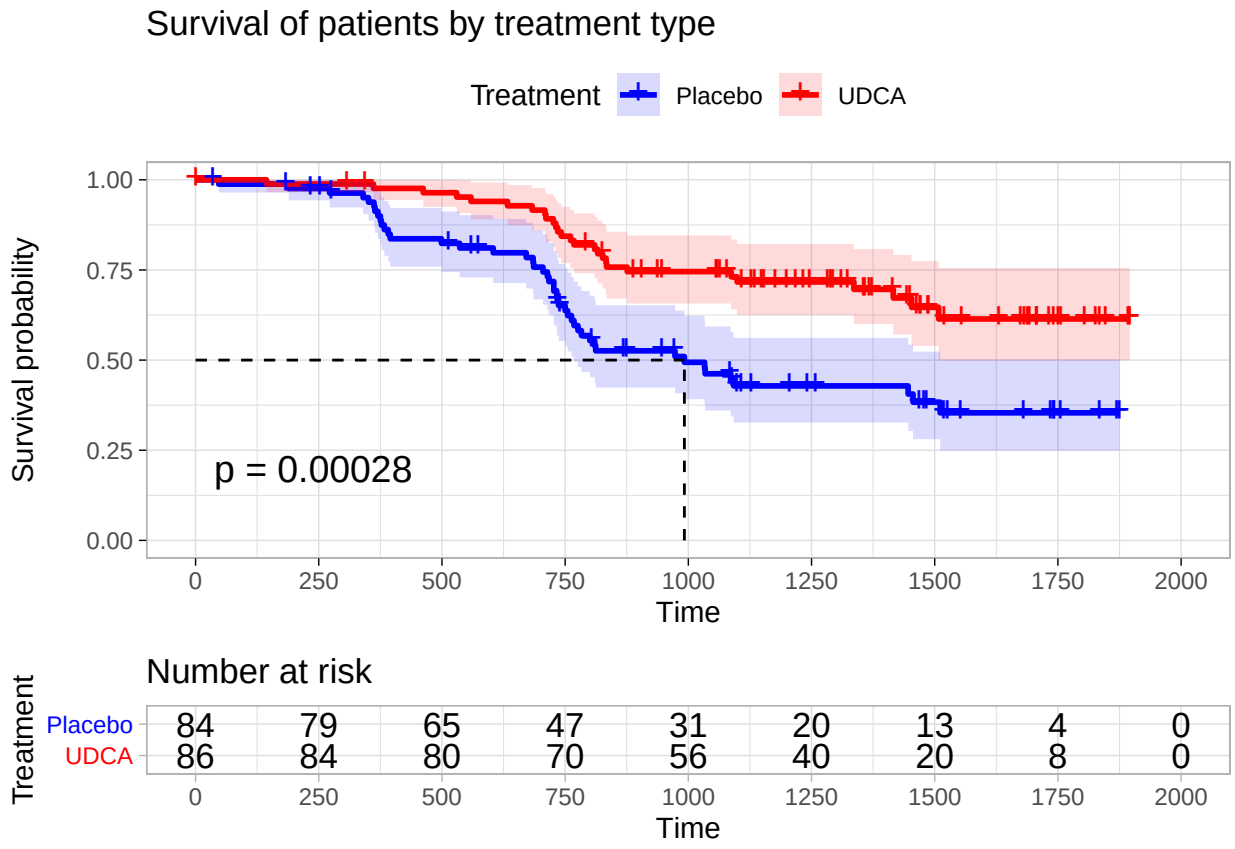
Statistical Methods and Case Studies

Preliminary Analysis

The primary covariate of interest is treatment in order to assess whether or not there is a significant difference between the treatment and a placebo. The mean survival time of placebo patients was 897 days, while those who received UDCA had a mean survival of 1170 days. Those who received the placebo had a median survival time of 992 days with a 95% confidence lower interval of (768, NA). Those who received the actual treatment did not reach a 50% survival rate by the end of the study, but the 95% confidence interval is (1508, NA). NA indicates that the upper confidence band did not drop below 50% survival before the study ended. Our null hypothesis is H_0: the true survival curves of these groups are the same. The alternative is Ha: the two survival curves are not the same. The log-rank test for the treatment variable yields a test statistic of $13.2 \sim \chi^2_1$ and a P-value of 3E-4, so treatment type has a significant effect on assessing survival outcome.

```
udca_trt=survfit(surv~trt,data = udca_data)
ggsurvplot(fit = udca_trt, data = udca_data,
           conf.int = T, conf.int.alpha=0.15,
           title = "Survival of patients by treatment type",

           censor=T, risk.table = T,
           surv.median.line = "hv",

           legend.labs=c("Placebo","UDCA"), legend.title="Treatment",
           break.time.by=250, pval = T,
           palette = c("blue","red"), ggtheme = theme_light())
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```




```
survdiff(surv~trt,
         data=udca_data)
```

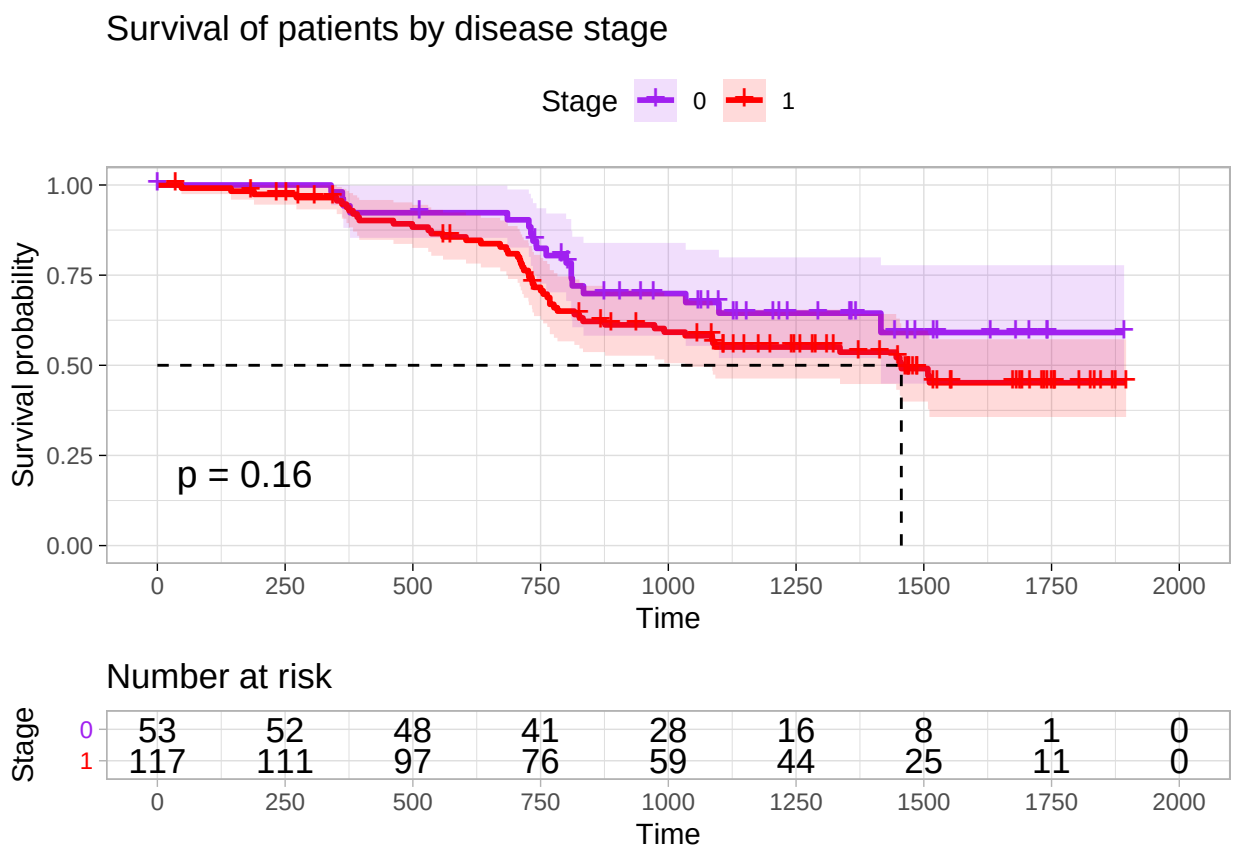
```
## Call:
## survdiff(formula = surv ~ trt, data = udca_data)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## trt=0 84       45     29.9      7.68      13.2
## trt=1 86       27     42.1      5.44      13.2
##
##  Chisq= 13.2  on 1 degrees of freedom, p= 3e-04
```

These curves begin to diverge from each other after about 350 days into the study. They maintain their distance from each other and, for the most part, are nearly parallel. It seems as though the benefit of the treatment is not very noticeable until about a year into the study. This visual test for a difference between these two survival curves is confirmed by the log-rank test, so we can reject the hypothesis that these curves are the same. We now look at the curves for the stage variable.

```
udca_stage=survfit(surv~stage,
                   data=udca_data)
```

```
ggsurvplot(fit = udca_stage, data = udca_data,
           title="Survival of patients by disease stage",
           pval = T, conf.int = T, conf.int.alpha=0.15,
           censor=T, surv.median.line = "hv",
           legend.labs = c("0","1"), legend.title = "Stage",
           palette = c("purple","red"),
           risk.table = T,
           ggtheme = theme_light(),
           break.time.by = 250)
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```




```
survdiff(surv~stage,
         data=udca_data)
```

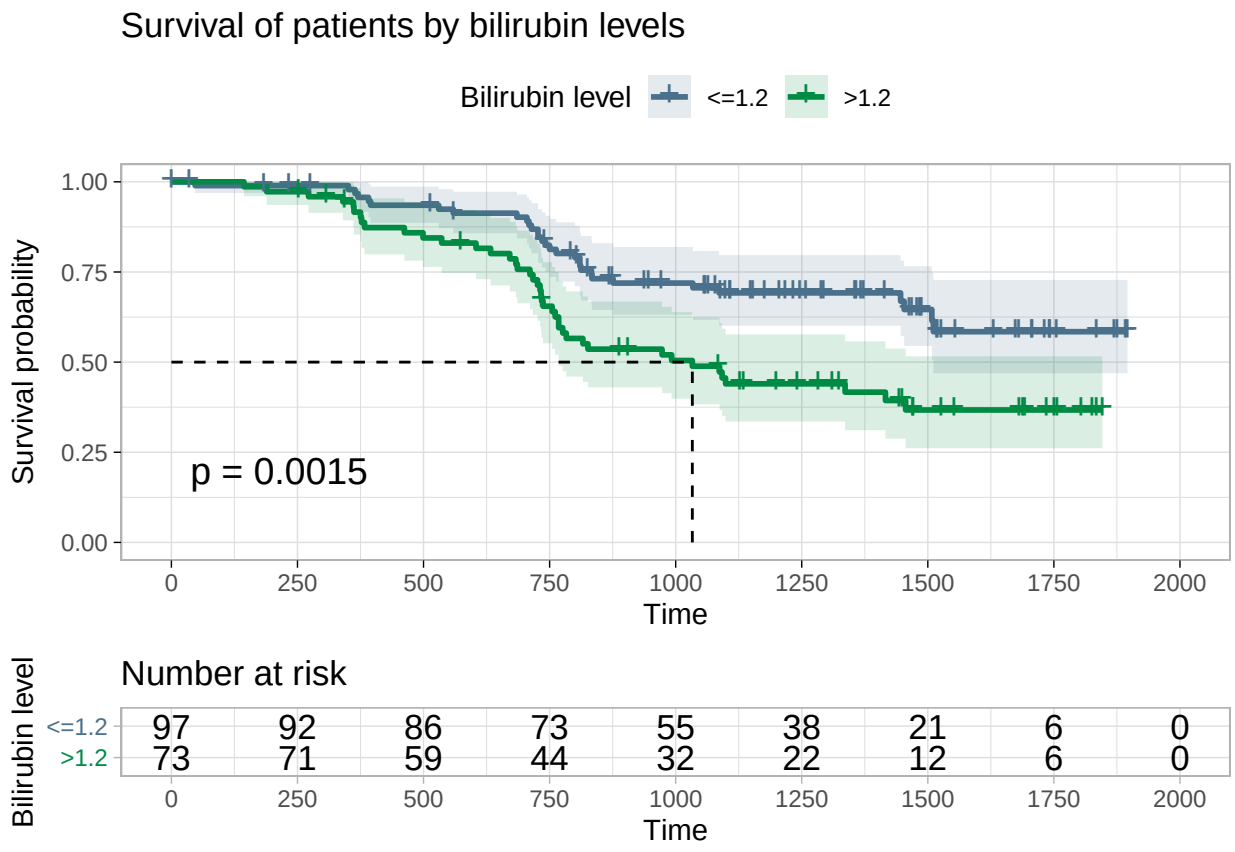
```
## Call:
## survdiff(formula = surv ~ stage, data = udca_data)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## stage=0   53       18     23.6     1.344      2.01
## stage=1 117       54     48.4     0.657      2.01
##
##  Chisq= 2  on 1 degrees of freedom, p= 0.2
```

Compared to the treatment curves, these are much closer and even have a bit of overlap at a few points. Additionally, the log-rank test yields a P-value of 0.16, which further indicates that these curves are not significantly different from each other.

We will continue to investigate survival curves of other covariates. A normal bilirubin level is considered to be less than or equal to 1.2 mg/dL6. As such, we will split the data into two groups – those with normal bilirubin levels and those with an above normal level. There are 97 patients with normal bilirubin levels and 73 with elevated bilirubin levels.

```
udca_bili = survfit(formula = surv ~ bili > 1.2,
                    data = udca_data)

ggsurvplot(fit = udca_bili, title = "Survival of patients by bilirubin levels",
  conf.int = T, conf.int.alpha = 0.15, censor = T,
  surv.median.line = 'hv', pval = T,
  legend.labs = c("<=1.2", ">1.2"), legend.title = "Bilirubin level",
  palette = c("skyblue4", "springgreen4"), risk.table = T,
  ggtheme = theme_light(),
  break.time.by = 250)
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```



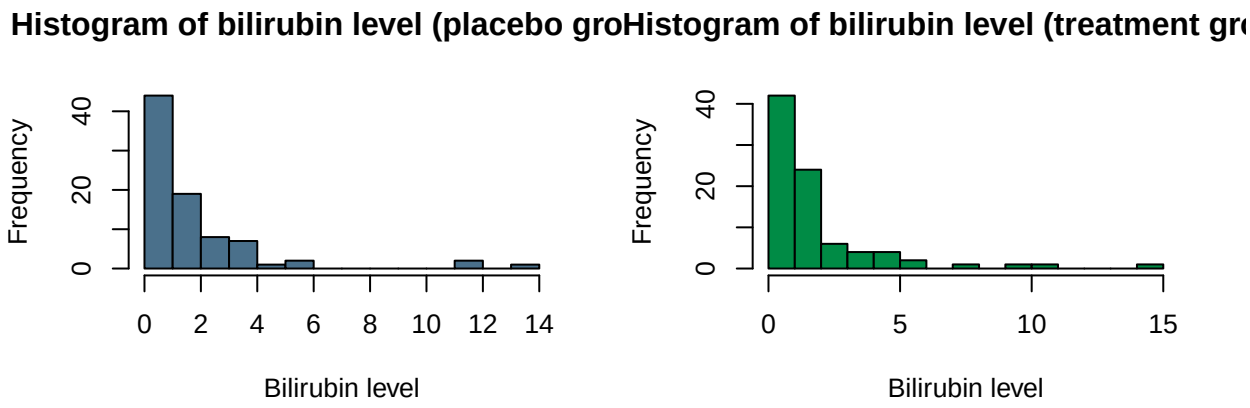

```
survdiff(formula = surv ~ bili > 1.2,
         data = udca_data)
```

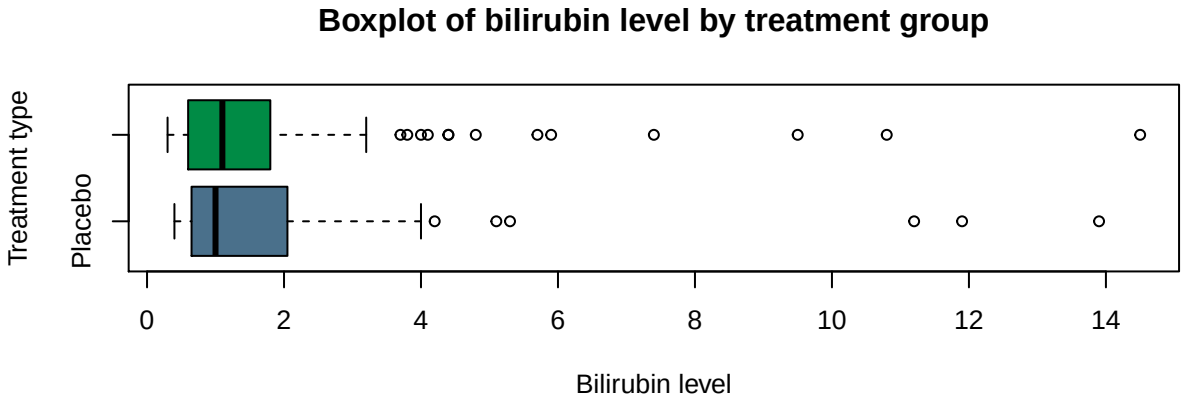
```
## Call:
## survdiff(formula = surv ~ bili > 1.2, data = udca_data)
##
##                     N Observed Expected (O-E)^2/E (O-E)^2/V
## bili > 1.2=FALSE 97       31     44.1      3.88        10
## bili > 1.2=TRUE  73       41     27.9      6.12        10
##
##  Chisq= 10  on 1 degrees of freedom, p= 0.002
```

According to the log-rank test, patients with abnormal bilirubin levels have a significantly different survival outcome than patients with normal levels (χ^2_1=10, p=0.002). Because PBC is typically associated with elevated bilirubin levels, we want to rule out the possibility that bilirubin level may be a confounding or interacting variable.

We will now visually and numerically assess the distribution of the bilirubin levels within each group.

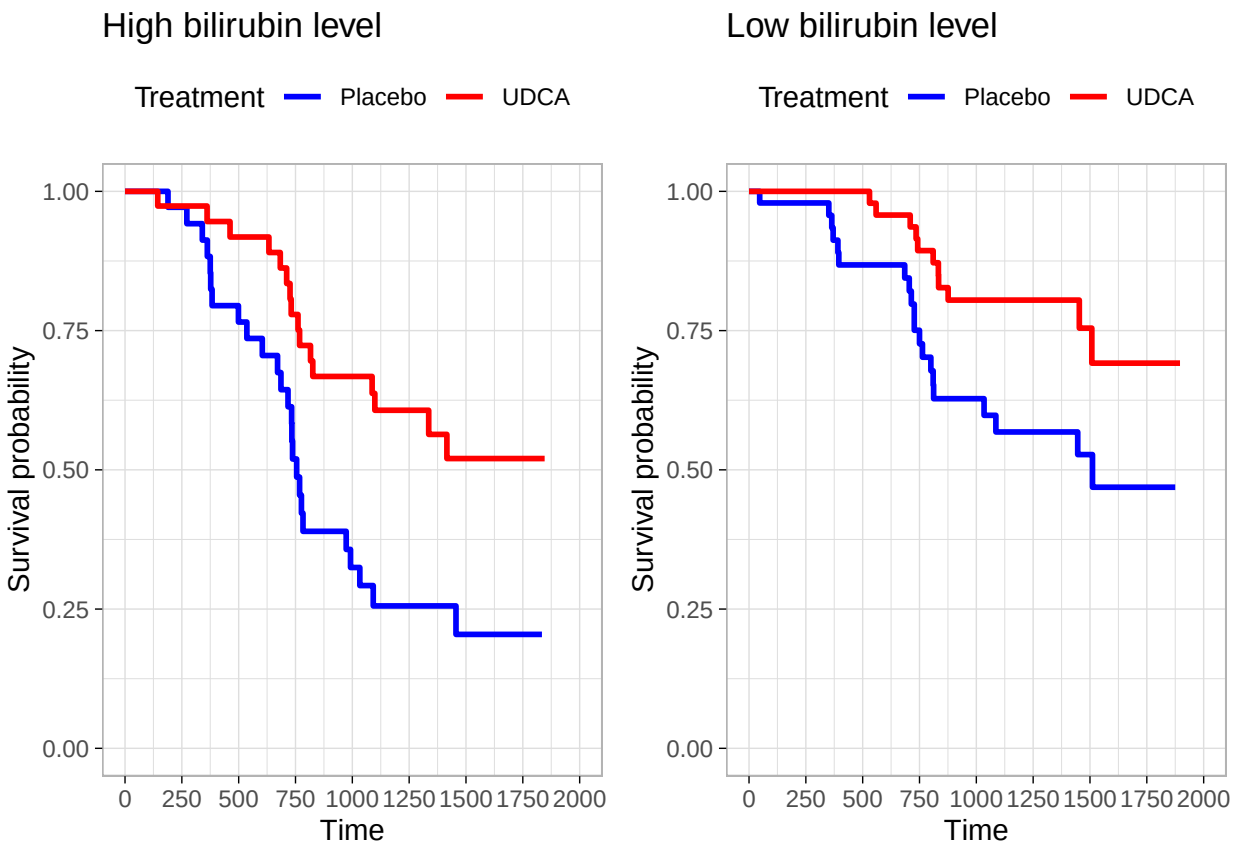
```
layout(matrix(c(1,2,3,3),ncol=2,byrow = T))
hist(x = udca_data[udca_data$trt == 0, ]$bili,
  breaks = 15,
  col = "skyblue4",
  xlab = "Bilirubin level",
  main = "Histogram of bilirubin level (placebo group)")
hist(x = udca_data[udca_data$trt==1,]$bili,
     breaks = 15,
     col = "springgreen4",
     xlab = "Bilirubin level",
     main = "Histogram of bilirubin level (treatment group)")
boxplot(formula = bili~as.factor(trt),
        data = udca_data,
        main = "Boxplot of bilirubin level by treatment group",
        ylab = "Treatment type",
        xlab = "Bilirubin level",
        names = c("Placebo","Treatment"),
        col = c("skyblue4","springgreen4"),
        horizontal = T)
```






```
bili_data_hi=udca_data[udca_data$bili>1.2,]
bili_data_lo=udca_data[udca_data$bili<=1.2,]
bili_hi=survfit(surv~trt,
                data=bili_data_hi)
bili_lo=survfit(surv~trt,
                data=bili_data_lo)
bili_lo_plot=ggsurvplot(fit = bili_hi,
           data=bili_data_hi,
           title="High bilirubin level",
           legend.labs=c("Placebo","UDCA"),
           legend.title="Treatment",
           break.time.by=250,
           censor=F,
           palette = c("blue","red"),
           ggtheme = theme_light())
bili_hi_plot=ggsurvplot(fit = bili_lo,
           data=bili_data_lo,
           title="Low bilirubin level",
           legend.labs=c("Placebo","UDCA"),
           legend.title="Treatment",
           break.time.by=250,
           censor=F,
           palette = c("blue","red"),
```

```
           ggtheme = theme_light())
bili_plot_list=list(bili_lo_plot, bili_hi_plot)
bili_plots=arrange_ggsurvplots(bili_plot_list)
```



Both the histogram and box plots make it quite apparent that bilirubin level is evenly distributed in each of the treatment groups. We also provide the five number summary of bilirubin level within each treatment group. The survival plots indicate higher bilirubin levels typically mean lower survival, but both plots have the treatment group living longer. There is little, if any interaction between treatment and bilirubin level.

```
summary(udca_data[udca_data$trt==0,4])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.400   0.675   1.000   1.826   2.025  13.900
```

```
summary(udca_data[udca_data$trt==1,4])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.300   0.600   1.100   1.886   1.775  14.500
```

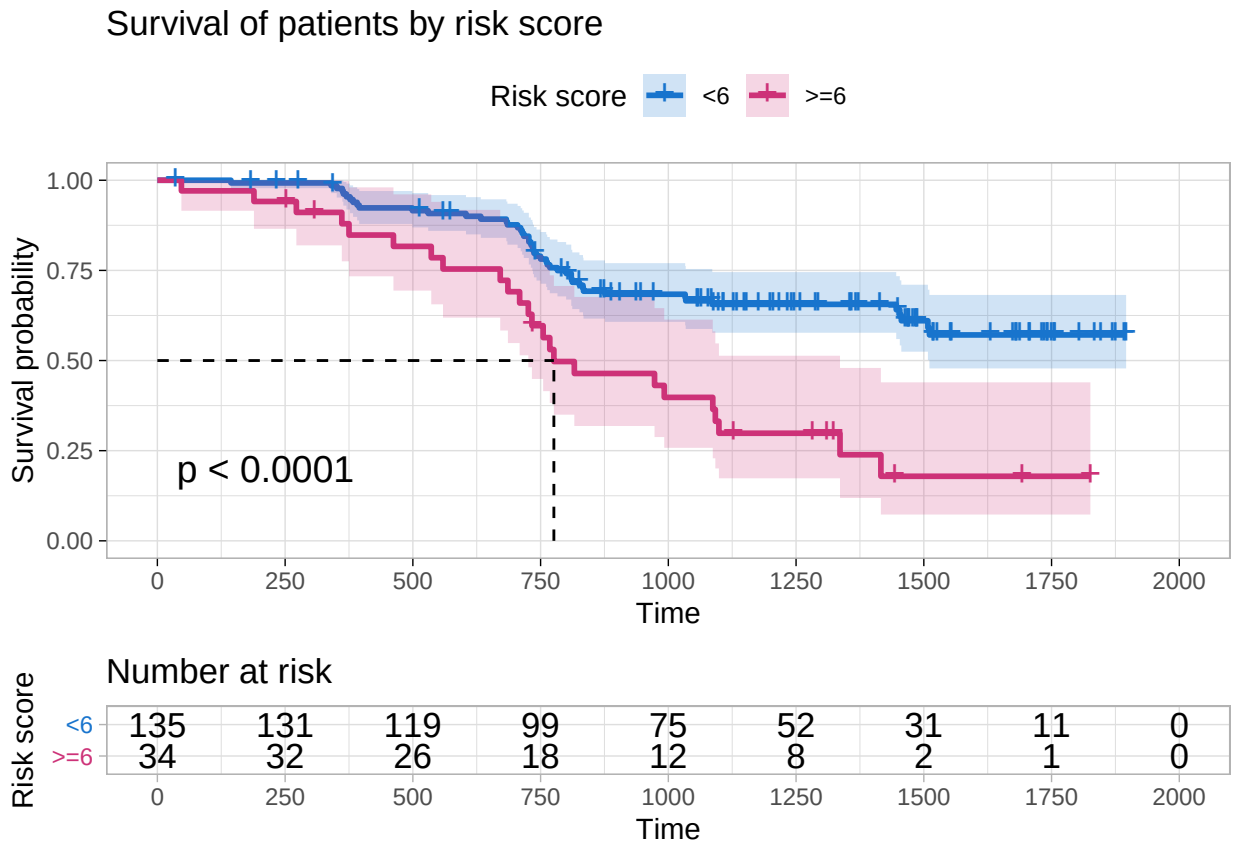
These summary statistics agree with the visual assessment and the distribution of bilirubin levels within the treatment groups is relatively identical.

Risk score is calculated from various factors such as age, bilirubin level, and other biological factors. A "high" risk score is considered to be 6 or larger, so the data is split into two groups based on this criterion. Those with a high risk score (>=6) have a median survival time of 776 days with a 95% confidence interval

of (726, 1336). Those with a low risk score did not reach 50% survival but have a 95% confidence interval for median survival time of (1511, NA).

```
udca_risk = survfit(formula=surv~riskscore>=6,
                    data=udca_data)
ggsurvplot(fit = udca_risk,
           title = "Survival of patients by risk score",
           conf.int = T, conf.int.alpha=0.2,
           censor = T,
           surv.median.line = 'hv', pval=T,
           legend.labs=c("<6",">=6"), legend.title="Risk score",
           palette = c("dodgerblue3","violetred3"),
           risk.table = T,
           ggtheme = theme_light(),
           break.time.by=250)
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```



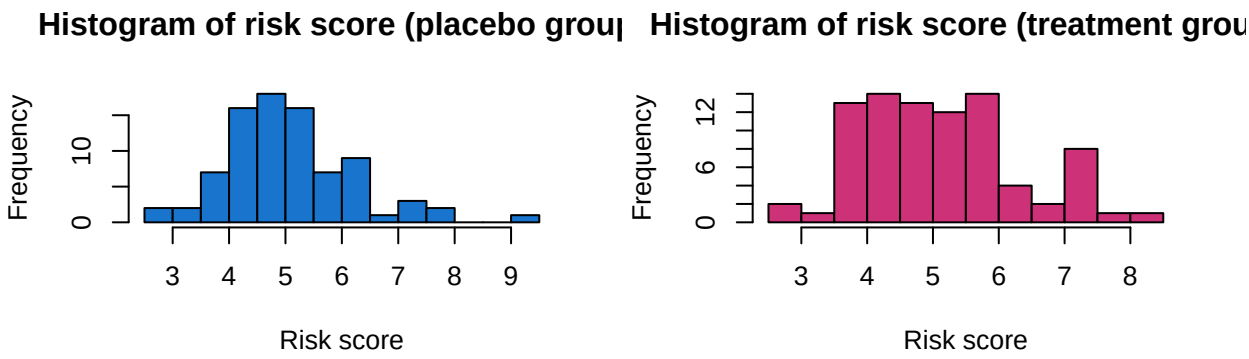

```
survdiff(formula = surv~riskscore>6,
         data = udca_data)
```

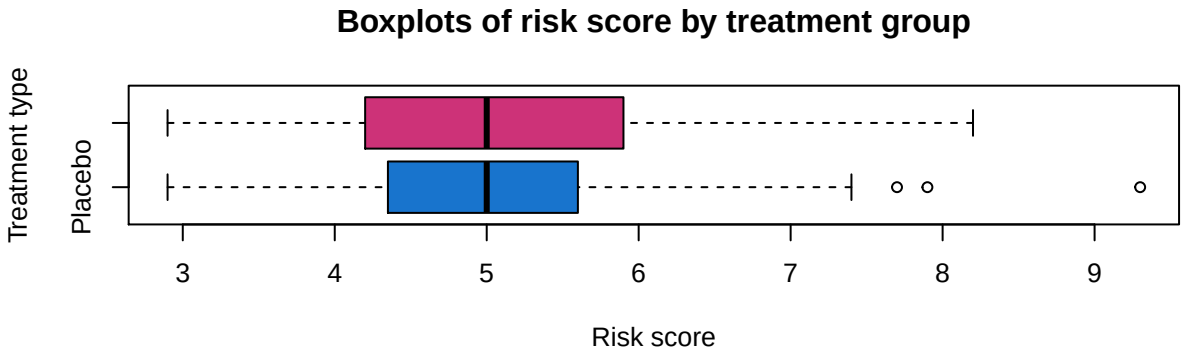
```
## Call:
## survdiff(formula = surv ~ riskscore > 6, data = udca_data)
```

```
##
## n=169, 1 observation deleted due to missingness.
##
##                         N Observed Expected (O-E)^2/E (O-E)^2/V
## riskscore > 6=FALSE 137       50     61.3      2.09      14.2
## riskscore > 6=TRUE   32       22     10.7     12.03      14.2
##
##  Chisq= 14.2  on 1 degrees of freedom, p= 2e-04
```

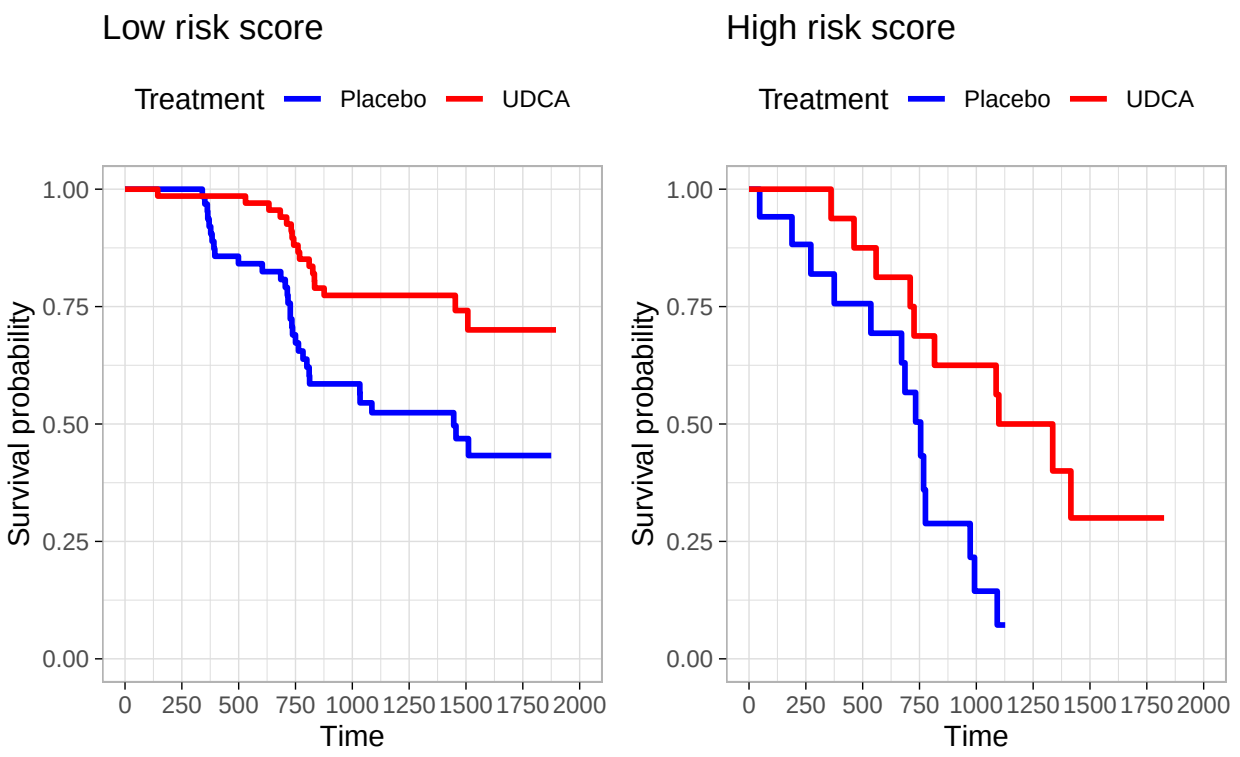
The log-rank test tells us that these curves are significantly different from each other (χ^2=14.2, p=2E-4). We now check to make sure this variable is not possibly confounding as well.

```
layout(matrix(c(1,2,3,3),ncol=2,byrow = T))
hist(x = udca_data[udca_data$trt==0,]$riskscore,
     breaks = 15,
     col="dodgerblue3",
     xlab="Risk score",
     main="Histogram of risk score (placebo group)",)
hist(x = udca_data[udca_data$trt==1,]$riskscore,
     breaks = 15,
     col="violetred3",
     xlab="Risk score",
     main="Histogram of risk score (treatment group)")
boxplot(formula=riskscore~as.factor(trt),
        data = udca_data,
        main="Boxplots of risk score by treatment group",
        ylab = "Treatment type",
        xlab="Risk score",
        names=c("Placebo","Treatment"),
        col=c("dodgerblue3","violetred3"),
        horizontal = T)
```






```
score_data_hi=udca_data[udca_data$riskscore>=6,]
score_data_lo=udca_data[udca_data$riskscore<6,]
score_hi=survfit(surv~trt,
                 data = score_data_hi)
score_lo=survfit(surv~trt,
                 data = score_data_lo)
score_lo_plot=ggsurvplot(fit = score_lo,
                         data=score_data_lo,
                         title="Low risk score",
                         legend.labs=c("Placebo","UDCA"),
                         legend.title="Treatment",
                         break.time.by=250,
                         censor=F,
                         palette = c("blue","red"),
                         ggtheme = theme_light())
score_hi_plot=ggsurvplot(fit = score_hi,
                         data=score_data_hi,
                         title="High risk score",
                         legend.labs=c("Placebo","UDCA"),
                         legend.title="Treatment",
                         break.time.by=250,
                         censor=F,
                         palette = c("blue","red"),
                         ggtheme = theme_light())
score_plot_list=list(score_lo_plot, score_hi_plot)
score_plots=arrange_ggsurvplots(score_plot_list)
```



The histograms and box plots both show that risk score is evenly distributed within the treatment groups, with slightly more variation within the treatment group. Additionally, there is not any noticeable interaction between risk score and treatment.

Modeling and Model Assessment

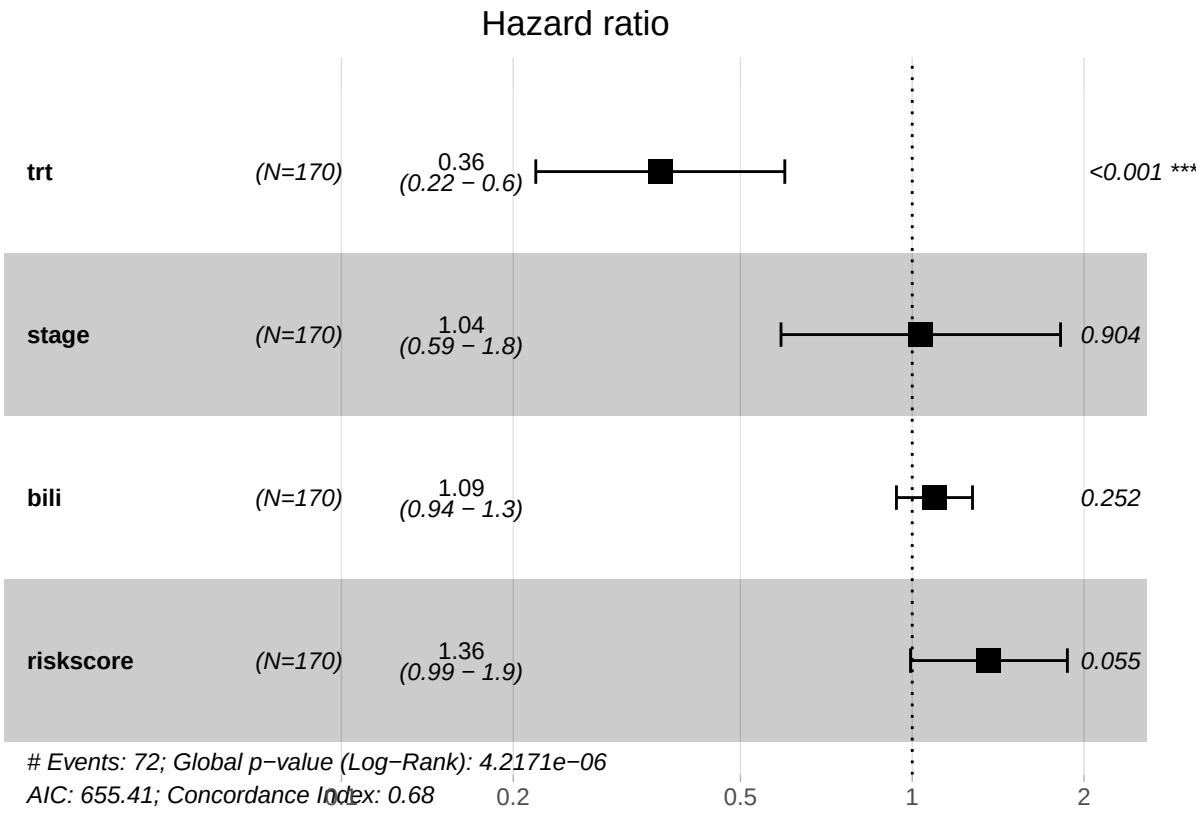
We now fit a Cox PH model to the data to asses which variables are the most useful for developing a model, adjusting for confounding variables, interaction, the PH assumption, and time-dependence as necessary.

```
fit1=coxph(surv~trt+stage+bili+riskscore,
           data = udca_data)
summary(fit1)
```

```
## Call:
## coxph(formula = surv ~ trt + stage + bili + riskscore, data = udca_data)
##
##   n= 169, number of events= 72
##    (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## trt       -1.01636   0.36191  0.25628 -3.966 7.31e-05 ***
## stage      0.03453   1.03514  0.28768  0.120   0.9045
## bili       0.08971   1.09385  0.07827  1.146   0.2518
## riskscore  0.30950   1.36275  0.16142  1.917   0.0552 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##           exp(coef) exp(-coef) lower .95 upper .95
## trt          0.3619     2.7631    0.2190    0.5981
## stage        1.0351     0.9661    0.5890    1.8192
## bili         1.0939     0.9142    0.9383    1.2752
## riskscore    1.3628     0.7338    0.9932    1.8699
##
## Concordance= 0.684  (se = 0.031 )
## Likelihood ratio test= 30.32  on 4 df,   p=4e-06
## Wald test            = 29.86  on 4 df,   p=5e-06
## Score (logrank) test = 31.47  on 4 df,   p=2e-06
```

```
ggforest(fit1, fontsize = 0.8,
         data = udca_data)
```



In this model, the coefficient for treatment is very significant in assessing the difference in hazard between the two treatment groups. Riskscore is close to significance while stage and bili are not very significant at all; their coefficients are close to 0. However, all three tests indicate that the model itself is significant. Patients in the placebo group have an average hazard rate $1/.36=2.76$ times greater than those receiving treatment. Every 1 point increase in stage or bili increases the hazard rate by only 1.04 or 1.09, respectively. We now assess the PH assumption of the model to determine which variables may violate the assumption.

```
cox.zph(fit1)
```

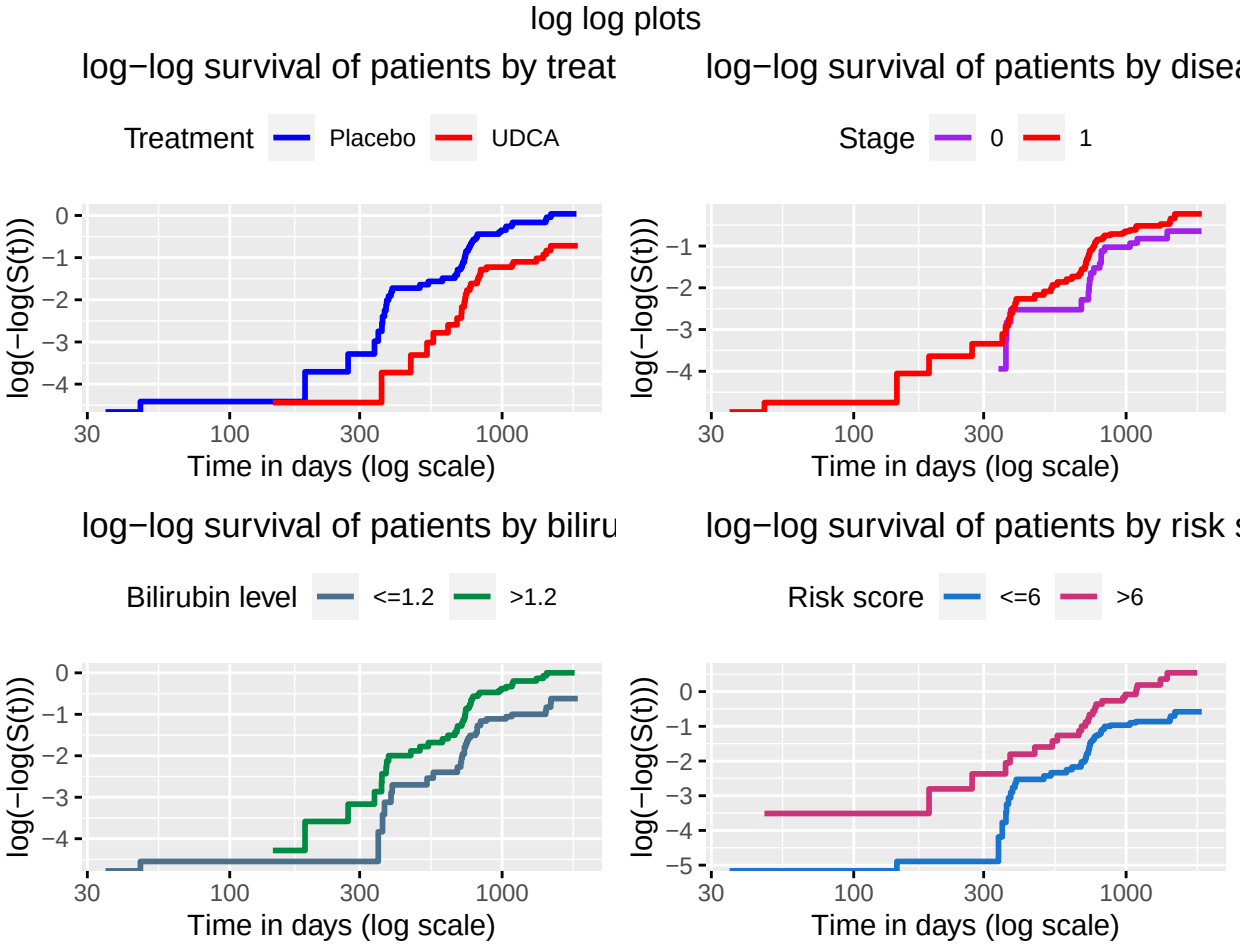
```
##           chisq df    p
## trt       0.587  1 0.44
```

```
## stage     0.100  1 0.75
## bili      1.275  1 0.26
## riskscore 0.312  1 0.58
## GLOBAL    2.620  4 0.62
```

Because the P-value is larger than 0.05 for each variable, along with the global P-value, we can conclude that this model and its variables likely satisfy the PH assumption. We can double check this assessment with log-log plots, plotting the observed plots against the expected plots, and evaluating the Schoenfeld residuals. Bilirubin level and risk score must be plotted as categorical variables for the log-log plots. They are split into the same groups as they were for the KM curves.

```
udca_trt_nozeros=survfit(surv~trt,
                         data=udca_data_nozeros)
loglogtrt=ggsurvplot(fit=udca_trt_nozeros,
           data=udca_data_nozeros,
           fun="cloglog",
           title="log-log survival of patients by treatment type",

           xlim=c(min(udca_trt_nozeros$time),
                  max(udca_trt_nozeros$time)),

           legend.labs=c("Placebo","UDCA"),
           legend.title="Treatment",
           xlab="Time in days (log scale)",

           palette = c("blue","red"),
           censor=F,
           ggtheme = theme_gray())

# log log plots for stage ----
udca_stage_nozeros=survfit(surv~stage,
                           data=udca_data_nozeros)
loglogstage=ggsurvplot(fit=udca_stage_nozeros,
           data=udca_data_nozeros,
           fun="cloglog",
           title="log-log survival of patients by disease stage",

           xlim=c(min(udca_stage_nozeros$time),
                  max(udca_stage_nozeros$time)),

           legend.labs=c("0","1"),
           legend.title="Stage",
           xlab="Time in days (log scale)",

           palette = c("purple","red"),
           censor=F,
           ggtheme = theme_gray())

# loglog plots for bilirubin levels ----
udca_bili_nozeros=survfit(surv~bili>1.2,
                          data=udca_data_nozeros)
loglogbili=ggsurvplot(fit=udca_bili_nozeros,
           data=udca_data_nozeros,
           fun="cloglog",
```

```
           title="log-log survival of patients by bilirubin level",

           xlim=c(min(udca_bili_nozeros$time),
                  max(udca_bili_nozeros$time)),

           legend.labs=c("<=1.2",">1.2"),
           legend.title="Bilirubin level",
           xlab="Time in days (log scale)",

           palette = c("skyblue4", "springgreen4"),
           censor=F,
           ggtheme = theme_gray())

# loglog plot for risk score ----
udca_risk_nozeros=survfit(surv~riskscore>=6,
                          data=udca_data_nozeros)
loglogrisk=ggsurvplot(fit=udca_risk_nozeros,
           data=udca_data_nozeros,
           fun="cloglog",
           title="log-log survival of patients by risk score",

           xlim=c(min(udca_bili_nozeros$time),
                  max(udca_bili_nozeros$time)),

           legend.labs=c("<=6",">6"),
           legend.title="Risk score",
           xlab="Time in days (log scale)",
           palette = c("dodgerblue3","violetred3"),
           censor=F,
           ggtheme = theme_gray())

# plot all log log plots from a list
logloglist=list(loglogtrt,
                loglogbili,
                loglogstage,
                loglogrisk)
arrange_ggsurvplots(x = logloglist,
                    nrow=2,
                    title = "log log plots")
```

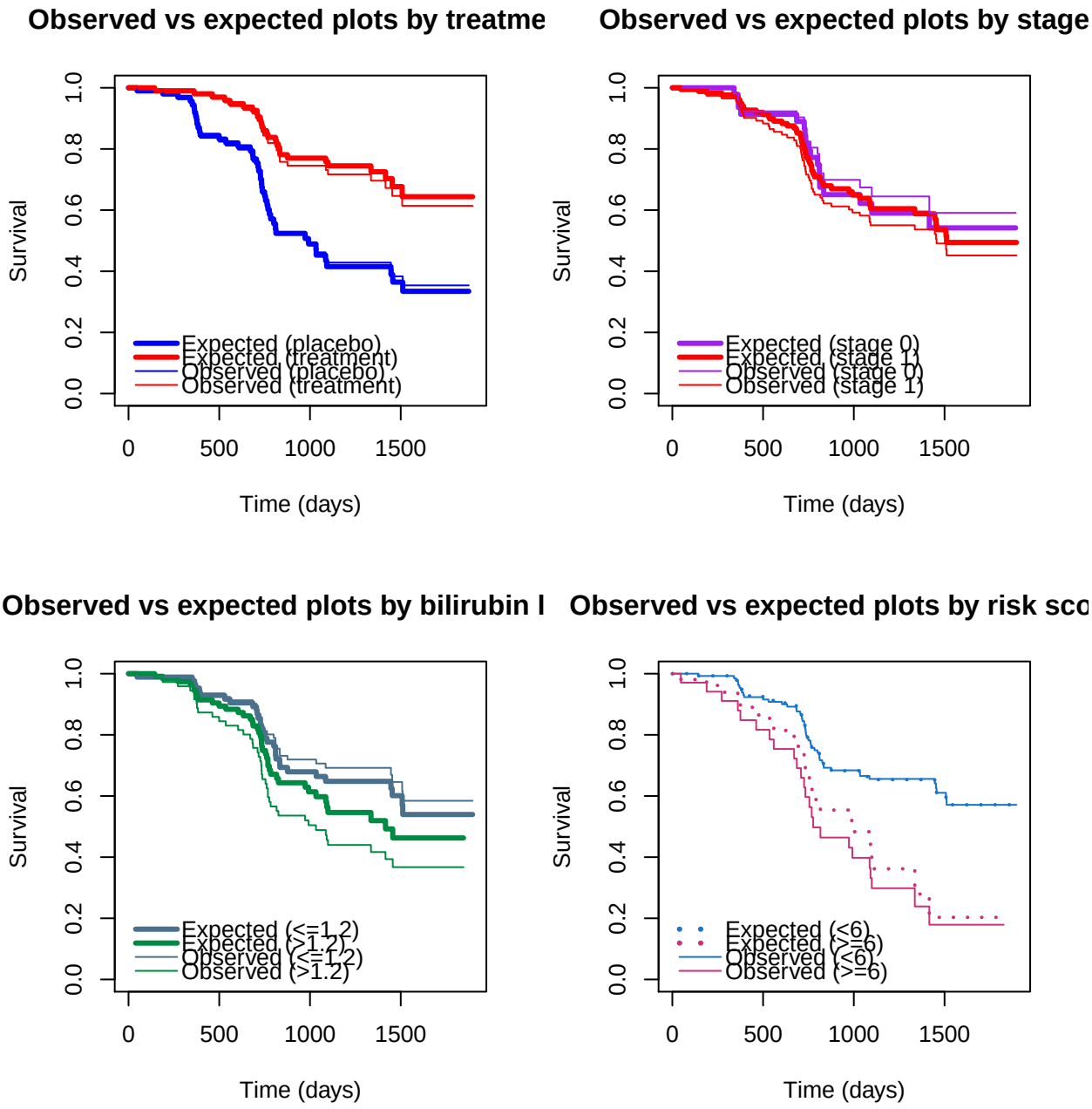


These plots provide some support for the goodness-of-fit test performed on the previous model, but despite stage's large P-value, it seems to be the least consistently parallel out of each variable, as the curves overlap as well. Stage may be in violation of the PH assumption, but the other variables seem to satisfy it. We now check observed vs. expected plots for each of these variables.

```
par(mfrow=c(2,2))
mean_df_1=data.frame(t(colMeans(x = udca_data[3:5],
                                na.rm = T,dims = 1)))
plot(survfit(coxph(surv~stage+bili+riskscore+strata(trt),
                   data = udca_data),
             newdata = mean_df_1),
     col=c("blue","red"),
     lwd=3,
     conf.int = F,
     main="Observed vs expected plots by treatment",
     xlab="Time (days)",
     ylab = "Survival")
par(new=T)
plot(udca_trt,
     col=c("blue","red"),
     axes=F)
legend("bottomleft",
       legend = c("Expected (placebo)","Expected (treatment)",
                  "Observed (placebo)","Observed (treatment)"),
```

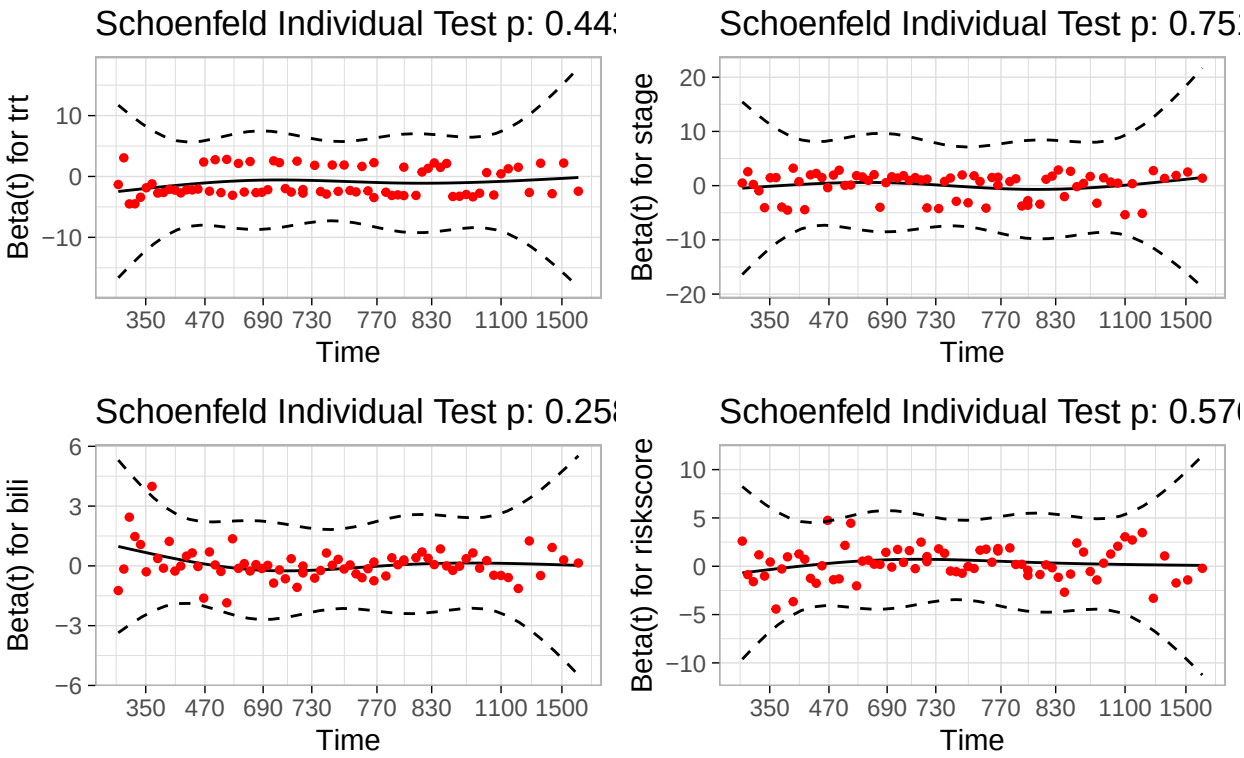
```
       lwd = c(3,3,1,1),
       col = c("blue","red","blue","red"),
       text.width = 0,
       x.intersp=0.2,
       y.intersp = 0.5,
       box.lty = 0)

# stage ----
mean_df_2=data.frame(t(colMeans(x = udca_data[c(2,4:5)],
                                na.rm = T,dims = 1)))
plot(survfit(coxph(surv~trt+bili+riskscore+strata(stage),
                   data = udca_data),
             newdata = mean_df_2),
     col=c("purple","red"),
     lwd=3,
     conf.int = F,
     main="Observed vs expected plots by stage",
     xlab="Time (days)",
     ylab = "Survival")
par(new=T)
plot(udca_stage,
     col=c("purple","red"),
     axes=F)
legend("bottomleft",
       legend = c("Expected (stage 0)","Expected (stage 1)",
                  "Observed (stage 0)","Observed (stage 1)"),
       lwd = c(3,3,1,1),
       col = c("purple","red","purple","red"),
       text.width = 0,
       x.intersp=0.2,
       y.intersp = 0.5,
       box.lty = 0)

# bilirubin ----
mean_df_3=data.frame(t(colMeans(x = udca_data[c(2,3,5)],
                                na.rm = T,dims = 1)))
plot(survfit(coxph(surv~trt+stage+riskscore+strata(bili>1.2),
                   data = udca_data),
             newdata = mean_df_3),
     col=c("skyblue4","springgreen4"),
     lwd=3,
     conf.int = F,
     main="Observed vs expected plots by bilirubin level",
     xlab="Time (days)",
     ylab = "Survival")
par(new=T)
plot(udca_bili,
     col=c("skyblue4","springgreen4"),
     axes=F)
legend("bottomleft",
       legend = c("Expected (<=1.2)","Expected (>1.2)",
                  "Observed (<=1.2)","Observed (>1.2)"),
       lwd = c(3,3,1,1),
```

```
       col = c("skyblue4","springgreen4","skyblue4","springgreen4"),
       text.width = 0,
       x.intersp=0.2,
       y.intersp = 0.5,
       box.lty = 0)

# riskscore ----
mean_df_4=data.frame(t(colMeans(x = udca_data[2:4],
                                na.rm = T,dims = 1)))
plot(survfit(coxph(surv~trt+stage+bili+strata(riskscore>=6),
                   data = udca_data),
             newdata = mean_df_4),
     col=c("dodgerblue3","violetred3"),
     lwd=2,
     lty = ("dotted"),
     conf.int = F,
     main="Observed vs expected plots by risk score",
     xlab="Time (days)",
     ylab = "Survival")
par(new=T)
plot(udca_risk,
     col=c("dodgerblue3","violetred3"),
     axes=F)
legend("bottomleft",
       legend = c("Expected (<6)","Expected (>=6)",
                  "Observed (<6)","Observed (>=6)"),
       lty = c("dotted","dotted","solid","solid"),
       lwd = c(3,3,1,1),
       col = c("dodgerblue3","violetred3","dodgerblue3","violetred3"),
       text.width = 0,
       x.intersp=0.2,
       y.intersp = 0.5,
       box.lty = 0)
```



The observed and expected plots for treatment are nearly identical, along with risk score, while bilirubin is parallel but not quite identical. The plots for stage cross over each other several times, further indicating that this variable may violate the PH assumption. We now check the Schoenfeld residuals to further assess which variables may violate the PH assumption. The Schoenfeld residual test tests H_0: $\rho = 0$ for each variable. If there is a relationship ($\rho \neq 0$) between the residuals and time, then the PH assumption is violated. The correlation for each variable is not significantly different from zero, but the previous plots hint that stage violates the PH assumption.

```
ggcoxzph(fit=cox.zph(fit1),
         ggtheme = theme_light())
```



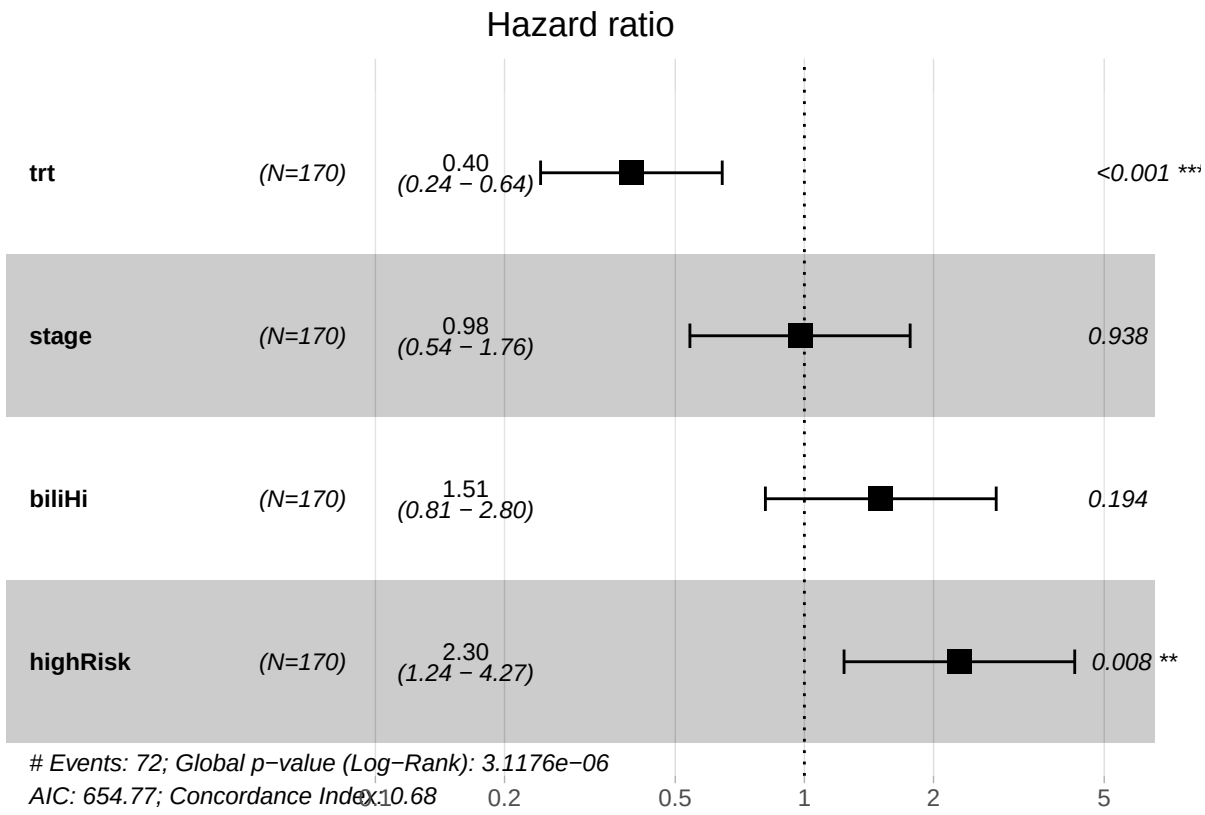
Because the P-values for each variable are above 0.05, we do not have enough evidence to reject the PH assumption for these variables based on the correlation alone. The global P-value is also large enough to not reject H_0. However, this does not tell us that these variables do satisfy the PH assumption.

We now create two new binary variables based on our continuous variables, bilirubin level and risk score. These will be split into two groups in the same manner as the KM curves. Bilirubin level is considered to be high if it is above 1.2, so this new variable will be 0 if bilirubin level is 1.2 or less and is 1 if bilirubin is above 1.2. For risk score, its indicator variable is 0 for risk scores below 6 and 1 for risk scores of 6 and above.

```
# add indicator column for high bilirubin level ----
for(i in 1:length(udca_data[,1])){
        if (udca_data$bili[i]>1.2){udca_data$biliHi[i]=1}
        else {udca_data$biliHi[i]=0}
}

# add indicator column for high risk score ----
# takes care of NA first then does the others
for(i in 1:length(udca_data[,1])){
        if (is.na(udca_data$riskscore[i])){udca_data$highRisk[i] = NA}
        else if (udca_data$riskscore[i] >= 6){udca_data$highRisk[i] = 1}
        else {udca_data$highRisk[i]=0}
}
udca_bili_bin=survfit(formula = surv~biliHi,
                      data = udca_data)
```

A full Cox model is created with these new indicator variables, and the output is provided below.

```
fit2=coxph(formula = surv~trt+stage+biliHi+highRisk,
           data = udca_data)
ggforest(fit2,fontsize = 0.8,
         data = udca_data)
```



Compared to the original model, the coefficient, standard error, and confidence intervals are all larger/wider for the variables for bilirubin level and risk score, as they are only indicators as opposed to continuous variables. However, the confidence intervals in this new model are not too wide, and the P-values are smaller for these variables compared to the first model.

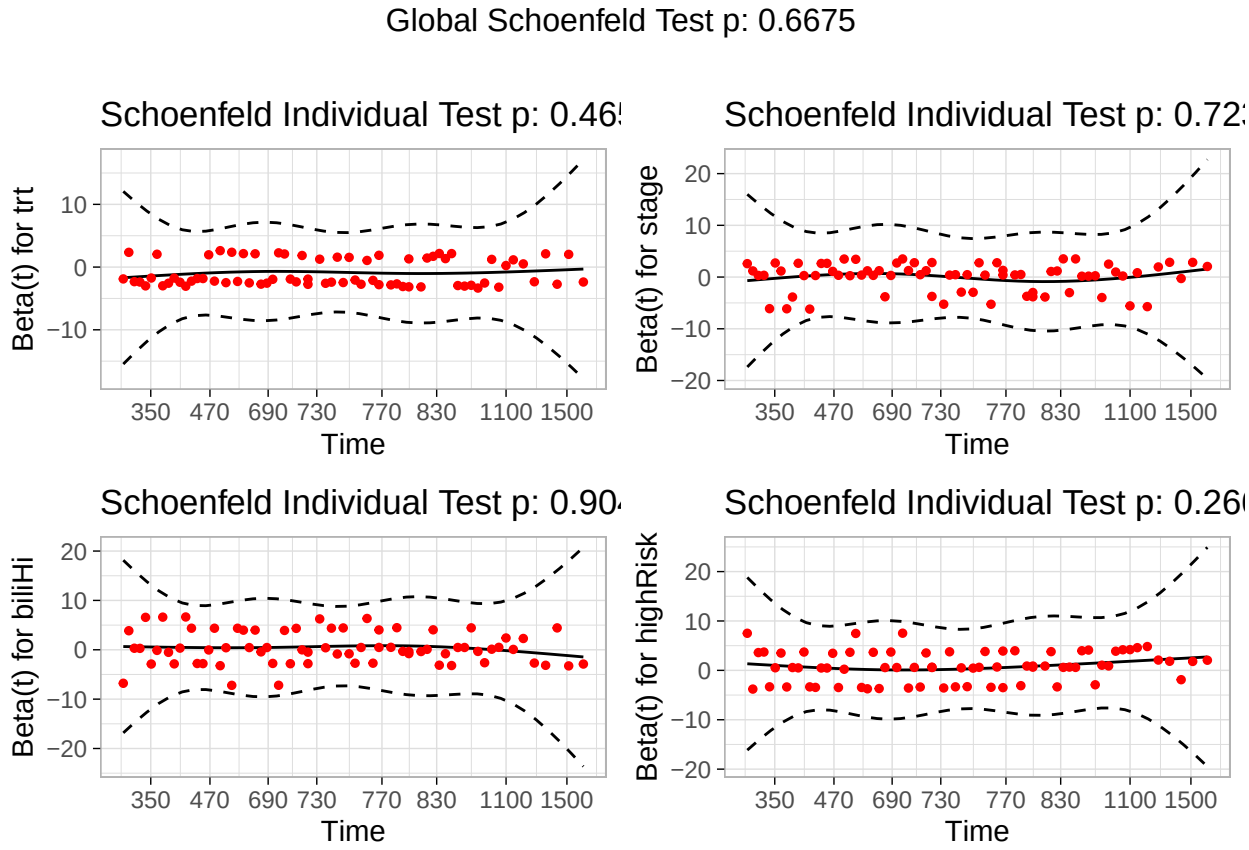
The P-values for biliHi and highRisk are both smaller than their continuous counterparts from the first full model, and risk can now be considered significant. This model may prove to be useful in certain situations, as it is simpler to assess the different average hazard rates for each variable. Those with a high risk score have, on average, 2.3 times the hazard than those with a low risk score. Those who received the placebo have an average hazard 2.5 times greater than those who received the treatment.

We now run diagnostics on this new model and compare it to the original model.

```
cox.zph(fit2)
```

```
##           chisq df    p
## trt      0.5318  1 0.47
## stage    0.1249  1 0.72
## biliHi   0.0145  1 0.90
## highRisk 1.2353  1 0.27
## GLOBAL   2.3728  4 0.67
```

```
ggcoxzph(fit = cox.zph(fit2),
         ggtheme = theme_light())
```



Once again, Schoenfeld residuals indicate that these variables do not violate the PH assumption.

Because the log-log plots seem to provide the most evidence against stage satisfying the PH assumption, we create a new model that is stratified on the stage variable. We create these models with both the continuous and indicator variables for bilirubin and risk score.

```
fit_stage_cont = coxph(formula = surv~trt+bili+riskscore+strata(stage),
                       data = udca_data)
fit_stage_indicator=coxph(formula = surv~trt+biliHi+highRisk+strata(stage),
                          data = udca_data)
summary(fit_stage_indicator)
```

```
## Call:
## coxph(formula = surv ~ trt + biliHi + highRisk + strata(stage),
##     data = udca_data)
##
##   n= 169, number of events= 72
##    (1 observation deleted due to missingness)
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## trt      -0.9250    0.3965   0.2494 -3.709 0.000208 ***
## biliHi    0.3852    1.4699   0.3146  1.224 0.220772
## highRisk  0.8187    2.2674   0.3180  2.575 0.010032 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## trt         0.3965     2.5220    0.2432    0.6464
## biliHi      1.4699     0.6803    0.7934    2.7231
## highRisk    2.2674     0.4410    1.2159    4.2284
##
## Concordance= 0.669  (se = 0.031 )
## Likelihood ratio test= 27.75  on 3 df,   p=4e-06
## Wald test            = 27.24  on 3 df,   p=5e-06
## Score (logrank) test = 29.64  on 3 df,   p=2e-06
```

```
summary(fit_stage_cont)
```

```
## Call:
## coxph(formula = surv ~ trt + bili + riskscore + strata(stage),
##     data = udca_data)
##
##   n= 169, number of events= 72
##    (1 observation deleted due to missingness)
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## trt       -1.00229   0.36704  0.25600 -3.915 9.04e-05 ***
## bili       0.08061   1.08395  0.07744  1.041   0.2979
## riskscore  0.31143   1.36538  0.16114  1.933   0.0533 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## trt           0.367     2.7245    0.2222    0.6062
## bili          1.084     0.9226    0.9313    1.2616
## riskscore     1.365     0.7324    0.9956    1.8725
##
## Concordance= 0.68  (se = 0.034 )
## Likelihood ratio test= 27.1  on 3 df,   p=6e-06
## Wald test            = 26.13  on 3 df,   p=9e-06
## Score (logrank) test = 27.73  on 3 df,   p=4e-06
```

The output for these models is almost identical to the original unstratified models. However, both the coefficient and standard error for bilirubin level and risk are larger in model 1, as there are only 2 possible values (0 and 1) as opposed to the continuous model. We now run diagnostics on these models to assess the PH assumption.

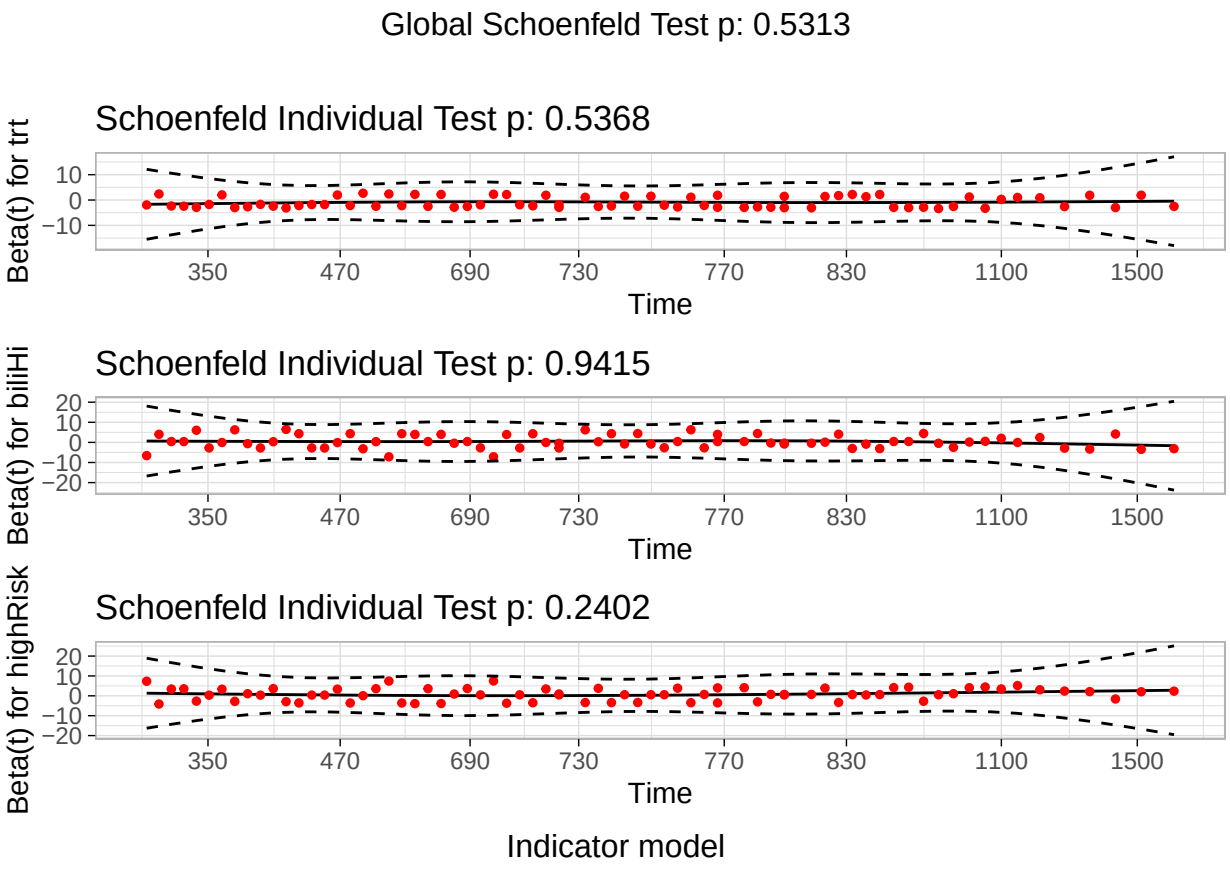
```
# assessment of stratified models ----
cox.zph(fit_stage_indicator)
```

```
##            chisq df    p
## trt      0.38157  1 0.54
## biliHi   0.00539  1 0.94
## highRisk 1.37943  1 0.24
## GLOBAL   2.20318  3 0.53
```

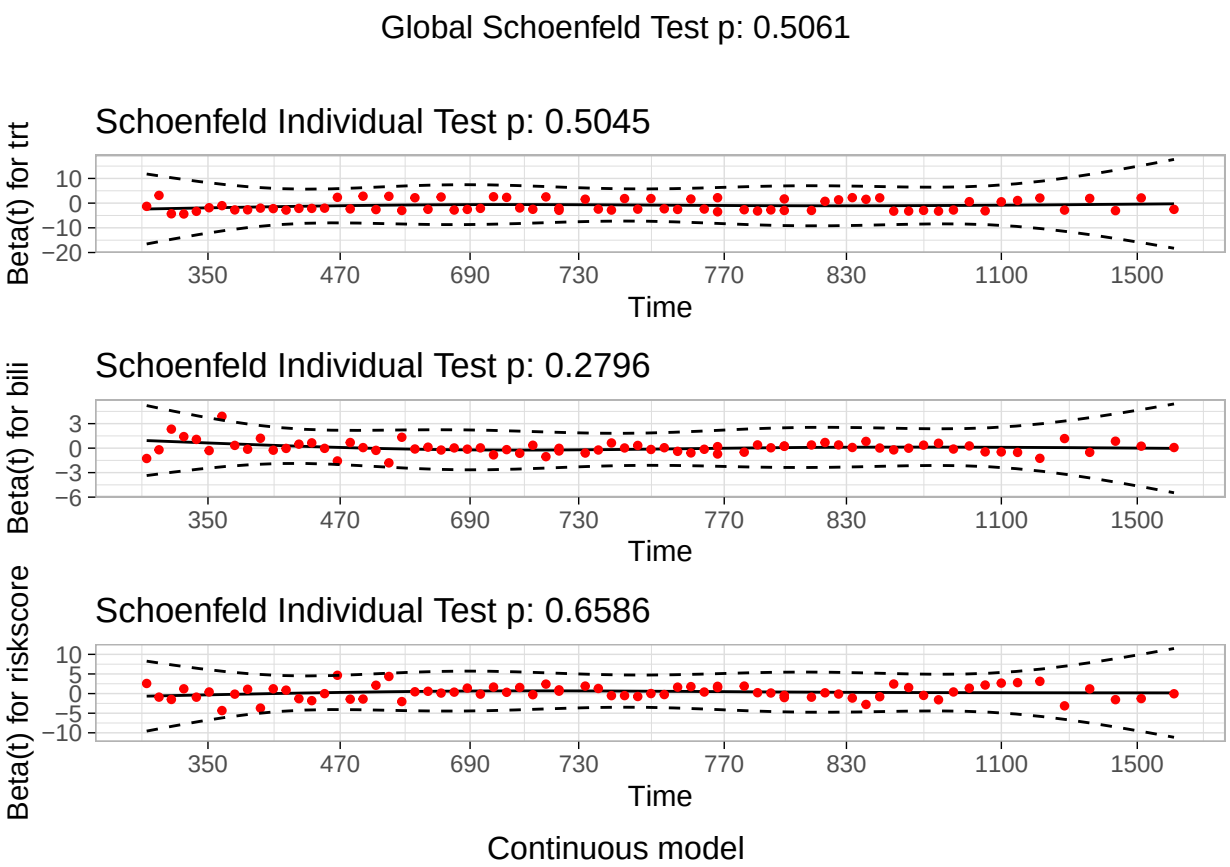
```
cox.zph(fit_stage_cont)
```

```
##            chisq df    p
## trt        0.445  1 0.50
## bili       1.169  1 0.28
## riskscore 0.195  1 0.66
## GLOBAL     2.333  3 0.51
```

```
ggcoxzph(fit = cox.zph(fit_stage_indicator),caption = "Indicator model",
         ggtheme = theme_light())
```




```
ggcoxzph(fit = cox.zph(fit_stage_cont),caption="Continuous model",
         ggtheme = theme_light())
```



For both models, we do not have enough evidence to reject the PH assumption for the global model or any of the variables. We now create two interactions models – one each for the indicator and continuous models. We want to determine if our current no-interaction model is more appropriate than separate models fitted for each stage group. We shift our focus to Model 1, the indicator version, first. The output and hazard model for the indicator interaction model are provided below.

```
fit_stage_interact_indicator=coxph(surv~(trt+biliHi+highRisk)*strata(stage),
                                   data = udca_data)
fit_stage_interact_cont=coxph(surv~(trt+bili+riskscore)*strata(stage),
                              data = udca_data)
summary(fit_stage_interact_indicator)
```

```
## Call:
## coxph(formula = surv ~ (trt + biliHi + highRisk) * strata(stage),
##     data = udca_data)
##
##   n= 169, number of events= 72
##    (1 observation deleted due to missingness)
##
##                                    coef exp(coef) se(coef)      z Pr(>|z|)
## trt                             -1.6588    0.1904   0.6432 -2.579  0.00991 **
## biliHi                           2.2350    9.3467   0.6862  3.257  0.00113 **
## highRisk                        -0.6109    0.5429   0.8688 -0.703  0.48196
## trt:strata(stage)stage=1         0.7802    2.1818   0.7045  1.107  0.26813
## biliHi:strata(stage)stage=1     -2.2227    0.1083   0.7713 -2.882  0.00395 **
## highRisk:strata(stage)stage=1    1.6909    5.4244   0.9378  1.803  0.07137 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                exp(coef) exp(-coef) lower .95 upper .95
## trt                               0.1904     5.2528   0.05396    0.6716
## biliHi                            9.3467     0.1070   2.43544   35.8709
## highRisk                          0.5429     1.8421   0.09889    2.9801
## trt:strata(stage)stage=1          2.1818     0.4583   0.54845    8.6797
## biliHi:strata(stage)stage=1       0.1083     9.2322   0.02389    0.4911
## highRisk:strata(stage)stage=1     5.4244     0.1844   0.86321   34.0865
##
## Concordance= 0.681  (se = 0.031 )
## Likelihood ratio test= 36.05  on 6 df,   p=3e-06
## Wald test            = 34.61  on 6 df,   p=5e-06
## Score (logrank) test = 39.1  on 6 df,   p=7e-07
```

```
summary(fit_stage_interact_cont)
```

```
## Call:
## coxph(formula = surv ~ (trt + bili + riskscore) * strata(stage),
##     data = udca_data)
##
##   n= 169, number of events= 72
##    (1 observation deleted due to missingness)
##
##                                    coef exp(coef) se(coef)      z Pr(>|z|)
## trt                            -0.85342   0.42595  0.50346 -1.695   0.0901 .
## bili                            0.03604   1.03670  0.15053  0.239   0.8108
## riskscore                       0.14471   1.15571  0.33453  0.433   0.6653
## trt:strata(stage)stage=1       -0.17203   0.84195  0.58425 -0.294   0.7684
## bili:strata(stage)stage=1       0.06516   1.06733  0.17897  0.364   0.7158
## riskscore:strata(stage)stage=1  0.18042   1.19772  0.38566  0.468   0.6399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                exp(coef) exp(-coef) lower .95 upper .95
## trt                                0.426     2.3477    0.1588     1.143
## bili                               1.037     0.9646    0.7718     1.392
## riskscore                          1.156     0.8653    0.5999     2.226
## trt:strata(stage)stage=1           0.842     1.1877    0.2679     2.646
## bili:strata(stage)stage=1          1.067     0.9369    0.7515     1.516
## riskscore:strata(stage)stage=1     1.198     0.8349    0.5624     2.551
##
## Concordance= 0.68  (se = 0.034 )
## Likelihood ratio test= 28.08  on 6 df,   p=9e-05
## Wald test            = 27.72  on 6 df,   p=1e-04
## Score (logrank) test = 29.62  on 6 df,   p=5e-05
```

```
anova(fit_stage_interact_indicator,fit_stage_indicator)
```

```
## Analysis of Deviance Table
##  Cox model: response is  surv
##  Model 1: ~ (trt + biliHi + highRisk) * strata(stage)
```

```
##  Model 2: ~ trt + biliHi + highRisk + strata(stage)
##    loglik Chisq Df P(>|Chi|)
## 1 -279.89
## 2 -284.04   8.3  3    0.0402 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit_stage_interact_cont,fit_stage_cont)
```

```
## Analysis of Deviance Table
##  Cox model: response is  surv
##  Model 1: ~ (trt + bili + riskscore) * strata(stage)
##  Model 2: ~ trt + bili + riskscore + strata(stage)
##    loglik  Chisq Df P(>|Chi|)
## 1 -283.88
## 2 -284.37 0.9838  3    0.8052
```

For the indicator model, our P-value is significant, which suggests that we can reject H0 (no interaction) in favor of using the interaction model. However, the P-value for the continuous model is quite large. We cannot reject H0, so the no-interaction model is acceptable for this case. The two preferred models seem to be the indicator model with interaction terms, and the continuous model with no interaction terms.

Conclusion

For a majority of the models, treatment remained a significant predicator in assessing survival and hazard ratios. Only when the models got more complex did it start to lose its significance, along with other variables as well. A parsimonious approach to this problem may prove to be more useful in properly assessing survival and hazard without becoming too complicated. Ultimately, the decision on whether to use the indicator or continuous version of the model depends on subsets of existing data of interest, or new data that may come along. For instance, a researcher with a sample of patients who all have normal or near normal bilirubin levels, with minimal variation, may prefer to use the continuous model in order to assess the hazard ratio between individuals in this group. Groups with a large amount of variation in these variables may prefer the indicator/binary model. Two subjects in this group with similar values may have a hazard ratio of nearly 1 when using the continuous model, but there would likely be greater interest in assessing the hazard ratio between a subject with a normal level vs. one with a high level.

References

1. The Mayo Clinic. (2019, March 9). Primary biliary cholangitis. Retrieved from https://www.mayoclinic.org/diseases-conditions/primary-biliary-cholangitis-pbc/symptoms-causes/syc-20376874.

2. T. M. Therneau and P. M. Grambsch, Modeling survival data: extending the Cox model. Springer, 2000.

3. K. D. Lindor, E. R. Dickson, W. P Baldus, R.A. Jorgensen, J. Ludwig, P. A. Murtaugh, J. M. Harrison, R. H. Weisner, M. L. Anderson, S. M. Lange, G. LeSage, S. S. Rossi and A. F. Hofman. Ursodeoxycholic acid in the treatment of primary biliary cirrhosis. Gastroenterology, 106:1284-1290, 1994.

4. Kleinbaum, D.G., and M. Klein. 2011. Survival Analysis: A Self-Learning Text. World Publishing Corporation.

5. Binet, J. L., Auquier, A. , Dighiero, G. , Chastang, C. , Piguet, H. , Goasguen, J. , Vaugier, G. , Potron, G. , Colona, P. , Oberling, F. , Thomas, M. , Tchernia, G. , Jacquillat, C. , Boivin, P. , Lesty, C. , Duault, M. T., Monconduit, M. , Belabbes, S. and Gremy, F. (1981), A new prognostic classification of chronic lymphocytic leukemia derived from a multivariate survival analysis. Cancer, 48: 198-206. https://doi.org/10.1002/1097-0142(19810701)48:1<198::AID-CNCR2820480131>3.0.CO;2-V

6. The Mayo Clinic. (2018, Nov 6). Bilirubin test. Retrieved from https://www.mayoclinic.org/tests-procedures/bilirubin/about/pac-20393041.

7. Boston University School of Public Health. (2016, June 3). Cox Proportional Hazards Regression Analysis. Retrieved from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Survival/BS704_Survival6.html.

8. Cox, D. R. (1984). Interaction. International Statistical Review, 1-24.

9. Healthline. High Bilirubin Levels: Symptoms and Causes of Elevated Bilirubin. Retrieved from https://www.healthline.com/health/high-bilirubin.

10. The Mayo Clinic. The Updated Natural History Model for Primary Biliary Cirrhosis. Retrieved from https://www.mayoclinic.org/medical-professionals/transplant-medicine/calculators/the-updated-natural-history-model-for-primary-biliary-cirrhosis/itt-20434724.