Webappendix for the article entitled Insights for quantifying the long-term benefit of immunotherapy using quantile regression

Bassirou Mboup, Christophe Le Tourneau, Aurélien Latouche

04/05/2020

This document makes an easier access to the supplementary material of the article entitled **Insights for quantifying the long-term benefit of immunotherapy using quantile regression**.

1) Data set

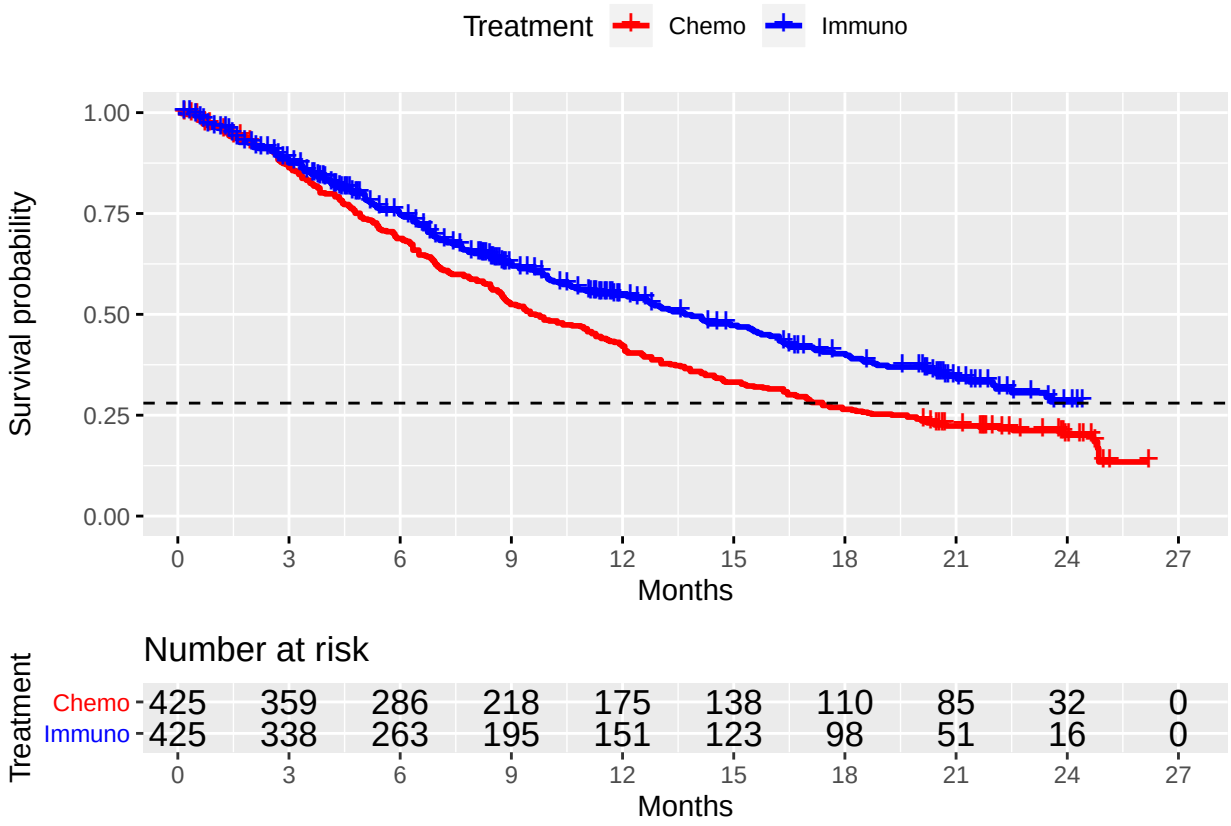
We use the algorithm of guyot. al 2012 to reconstructing individual-level time-to-event data based on the published Kaplan–Meier curves of the randomized controlled trial (Rittmeyer et al. 2017). This algorithm is available in code R based on the inverse of the Kaplan Meier equation.

After reconstruction, we get in this dataset the following variables.

- time : vector of observed failure times.
- event: vector of indicator of status (0 for censoring, 1 for type of event).
- tmt.arm.number: vector of treatment indicator (1 if treated and 0 otherwise).
- treatment.type: the type of treatment immonotherapy or chemotherapy

```
##        time event tmt.arm.number treatment.type
## 1 0.4059140     1              1   Atezolizumab
## 2 0.4059140     1              1   Atezolizumab
## 3 0.4059140     1              1   Atezolizumab
## 4 0.5599768     1              1   Atezolizumab
## 5 0.5599768     1              1   Atezolizumab
## 6 0.5599768     1              1   Atezolizumab
```

```
## Kaplan Meier curves
fit_KM <- survfit(Surv(time,event)~tmt.arm.number,data=data_ICI_Rittmeyer)

res <- ggsurvplot(fit_KM,data=data_ICI_Rittmeyer,
           risk.table=TRUE,
           conf.int=FALSE,
           xlim=c(0.4,27),
           palette =c("red","blue"),
           xlab="Months",
           risk.table.y.text.col=T,
            break.time.by=3,
          ggtheme = theme_grey() ,
           legend.title="Treatment",
           legend.labs=c("Chemo","Immuno")
            )
res$table <- res$table + theme(axis.line = element_blank())
res$plot <- res$plot+geom_hline(yintercept=0.28,lty=2)
print(res)
```



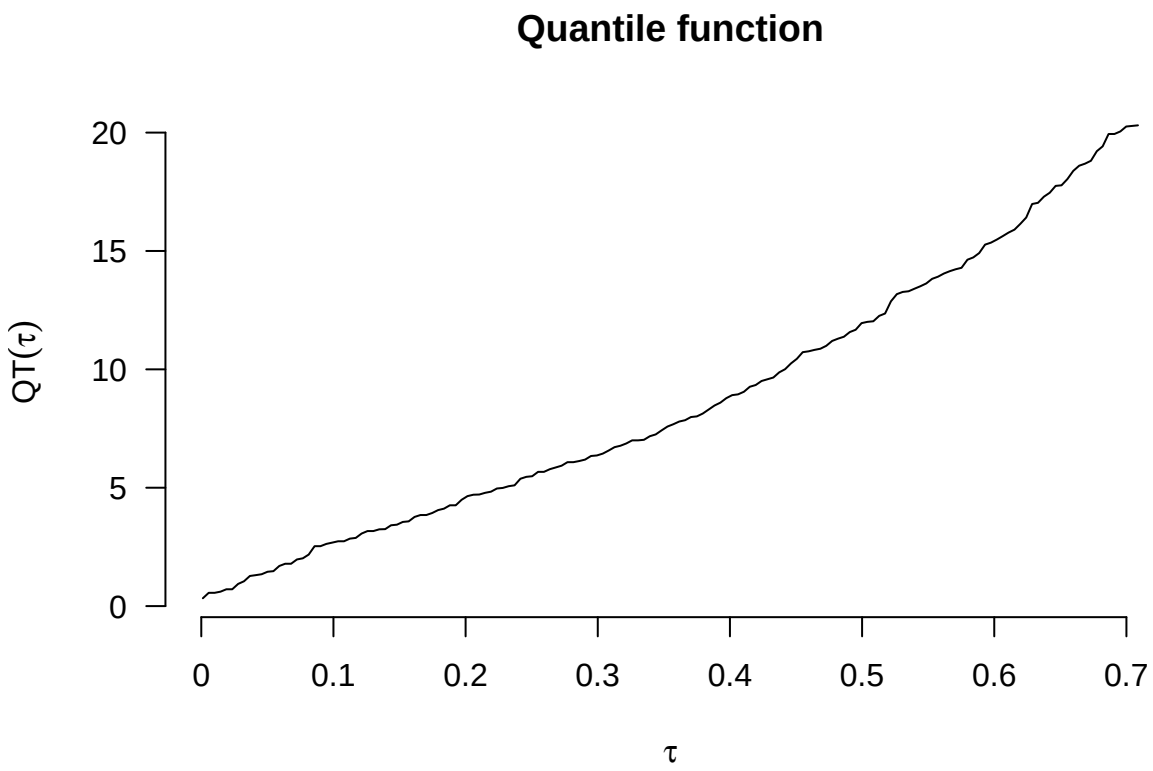
2) Application

```
library(quantreg)
x <- c(0.1, 0.2, 0.3, 0.4,0.5,0.6)
Rq <- crq(Surv(time,event)~tmt.arm.number,data=data_ICI_Rittmeyer,method="Pen")
result <- summary(Rq,taus=x)
result
```

```
##
## tau: [1] 0.1
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     2.70243  2.70243  2.81841  0.02959  91.33789  0.00000
## tmt.arm.number -0.01844 -0.04192  1.01991  0.27088  -0.06807  0.94573
##
## tau: [1] 0.2
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     4.15397  1.73010  4.69080  0.75529   5.49982  0.00000
## tmt.arm.number  0.88255 -0.22914  2.60054  0.72187   1.22259  0.22148
##
## tau: [1] 0.3
##
## Coefficients:
```

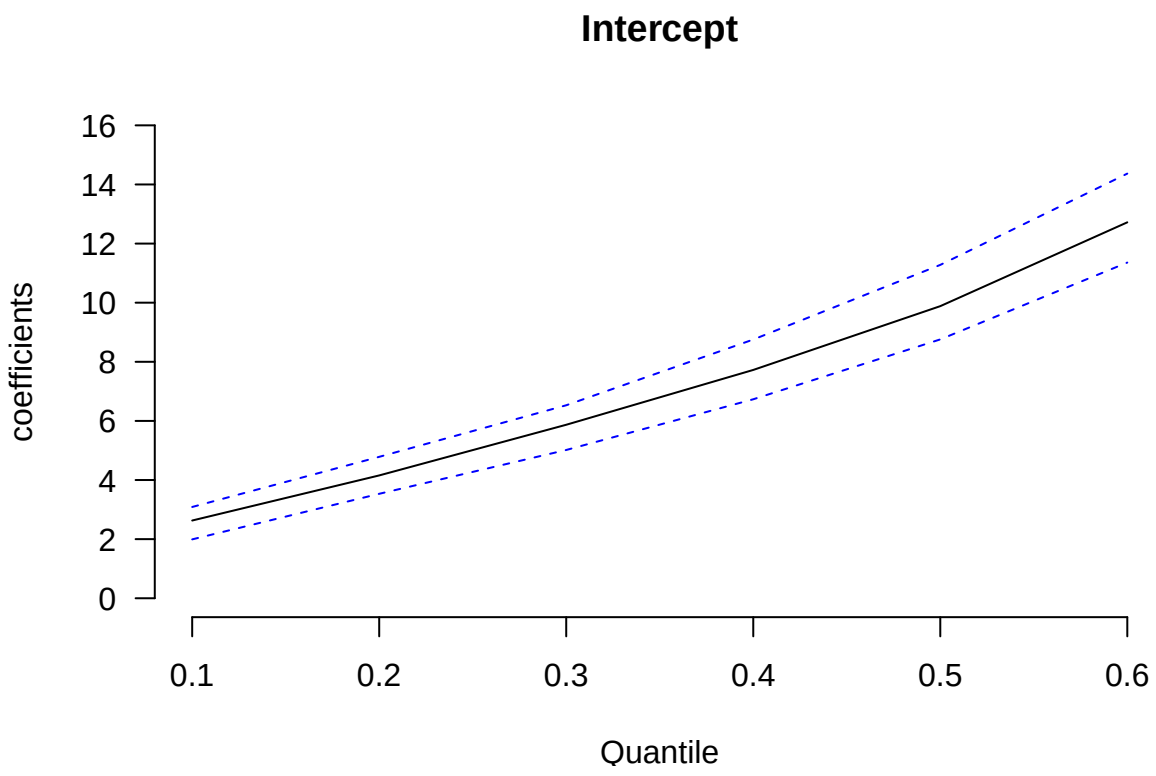
```
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     5.86294  5.86294  6.36690  0.12857  45.60287  0.00000
## tmt.arm.number  1.02312  0.51916  2.69376  0.55476   1.84427  0.06514
##
## tau: [1] 0.4
##
## Coefficients:
##                Value   Lower Bd Upper Bd Std Error T Value Pr(>|t|)
## (Intercept)    7.80078 4.60256  9.11294  1.15063   6.77959 0.00000
## tmt.arm.number 2.12340 0.26985  5.18194  1.25311   1.69451 0.09017
##
## tau: [1] 0.5
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     9.78031  7.79325  9.78031  0.50691  19.29381  0.00000
## tmt.arm.number  4.35360  2.94401  6.89136  1.00700   4.32336  0.00002
##
## tau: [1] 0.6
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)    12.69400 10.87159 13.20487  0.59524  21.32596  0.00000
## tmt.arm.number  5.46031  2.91633  8.39814  1.39845   3.90456  0.00009

## jack

## jack

## jack

## jack

## jack

## jack
```

```r
# Quantile function
tau <- Rq$sol["tau",][1:160]
q<- Rq$sol["Qhat",][1:160]
plot(tau,q,type="l",xlab = expression(tau),ylab = expression(QT(tau)),
     main="Quantile function",axes=FALSE)
axis(1,at=seq(from=0,to=0.7,by=0.1),labels=seq(from=0,to=0.7,by=0.1),las=1)
axis(2,at=seq(from=0,to=20,by=5),labels=seq(from=0,to=20,by=5),las=2)
```

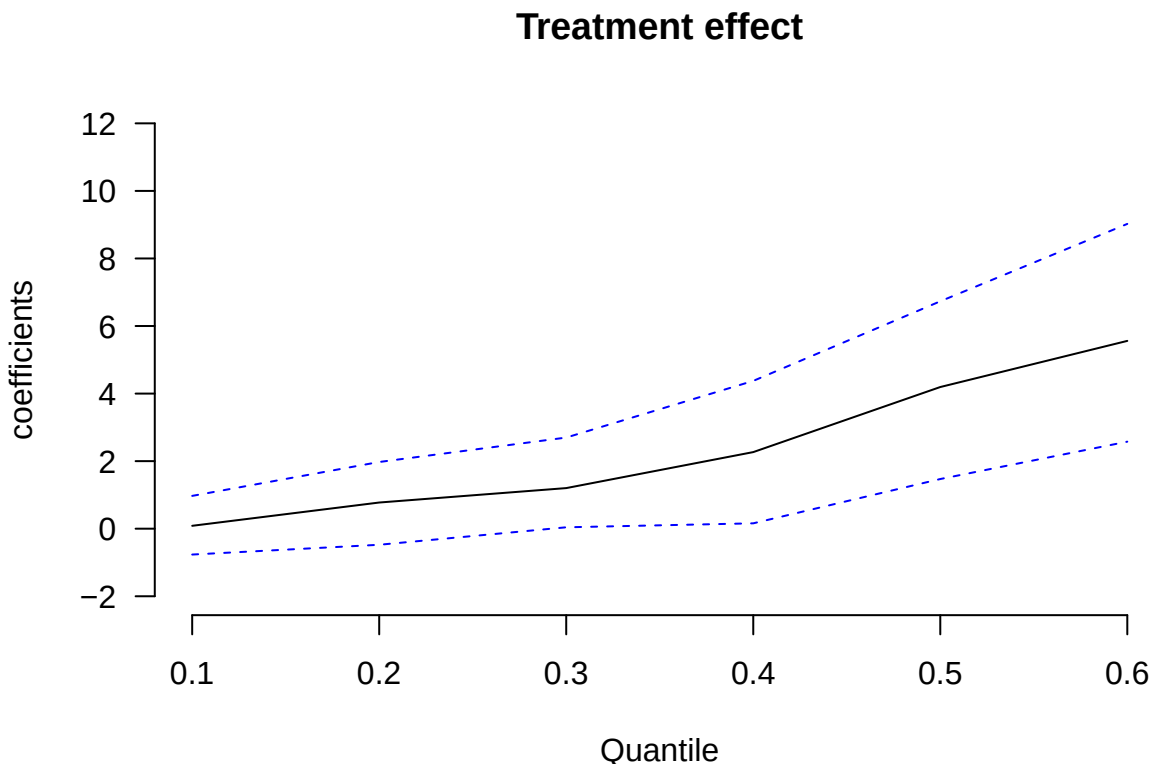


3) Resampling for coefficients

```
############ data for bootstrapping with 1000 replications ################
load("~/Desktop/Immunotherapy/Mat.RData")
load("~/Desktop/Immunotherapy/Moy.RData")
load("~/Desktop/Immunotherapy/Mat1.RData")
load("~/Desktop/Immunotherapy/Moy1.RData")

b <- matrix(data=NA,ncol=6,nrow = 1000)
for(i in 1:6){
b[,i] <-Mat1[,i][order(Mat1[,i])]
}

inf0 <- NA
for(i in 1:6){
inf0[i] <- b[,i][25]
}
sup0 <- NA
for(i in 1:6){
  sup0[i] <- b[,i][975]
}

plot(x,Moy1,type="l",ylim=c(0,16),ylab="coefficients",xlab ="Quantile",main="Intercept",axes = FALSE)
axis(1,at=seq(from=0.1,to=0.6,by=0.1),labels=seq(from=0.1,to=0.6,by=0.1),las=1)
axis(2,at=seq(from=0,to=16,by=2),labels=seq(from=0,to=16,by=2),las=2)
lines(x,sup0,col="blue",lty=2)
lines(x,inf0,col="blue",lty=2)
```

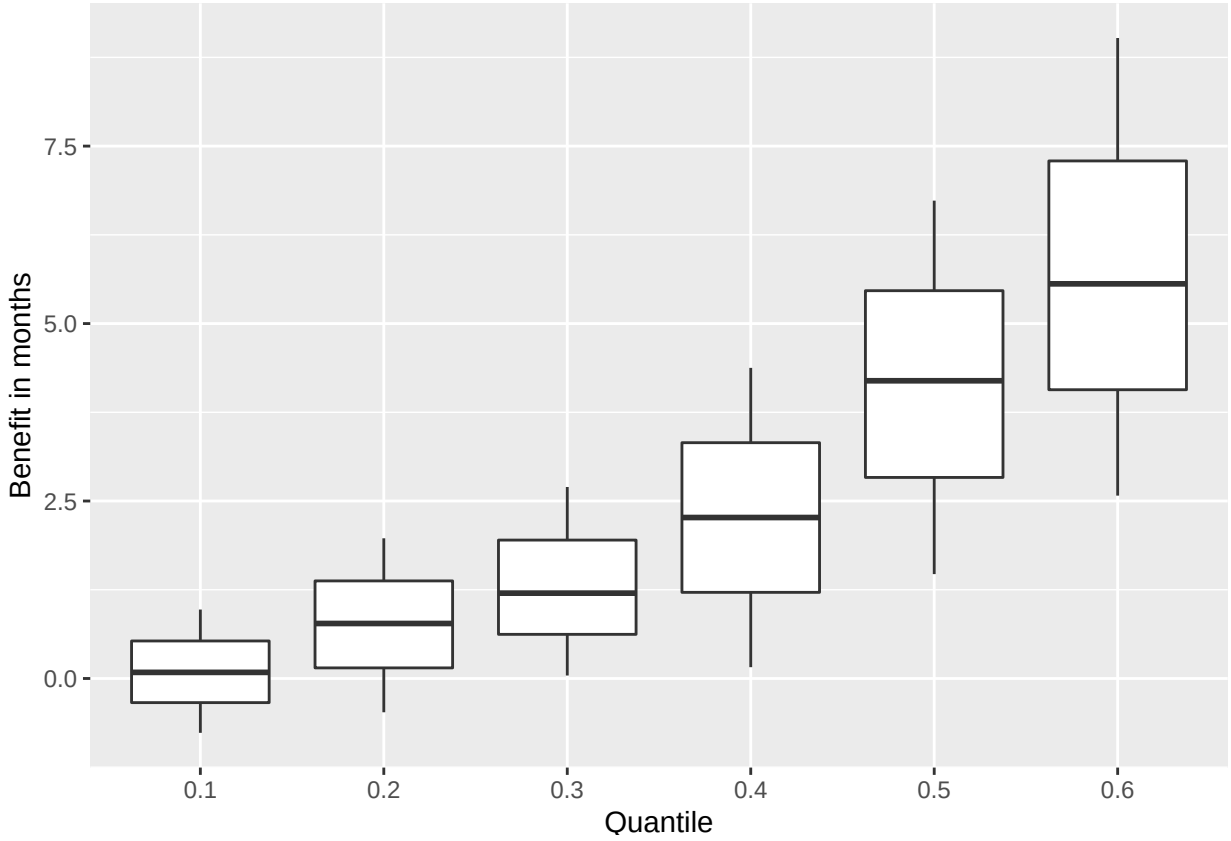



```
## Variation of the treatment effect as function of each quantile available and confidence
# intervals
b <- matrix(data=NA,ncol=6,nrow = 1000)
for(i in 1:6){
b[,i] <-Mat[,i][order(Mat[,i])]
}

inf <- NA
for(i in 1:6){
inf[i] <- b[,i][25]
}
sup <- NA
for(i in 1:6){
  sup[i] <- b[,i][975]
}


plot(x,Moy,type="l",ylim=c(-2,12),ylab="coefficients",xlab ="Quantile",main="Treatment effect",axes = F
axis(1,at=seq(from=0.1,to=0.6,by=0.1),labels=seq(from=0.1,to=0.6,by=0.1),las=1)
axis(2,at=seq(from=-2,to=12,by=2),labels=seq(from=-2,to=12,by=2),las=2)
lines(x,sup,col="blue",lty=2)
lines(x,inf,col="blue",lty=2)
```



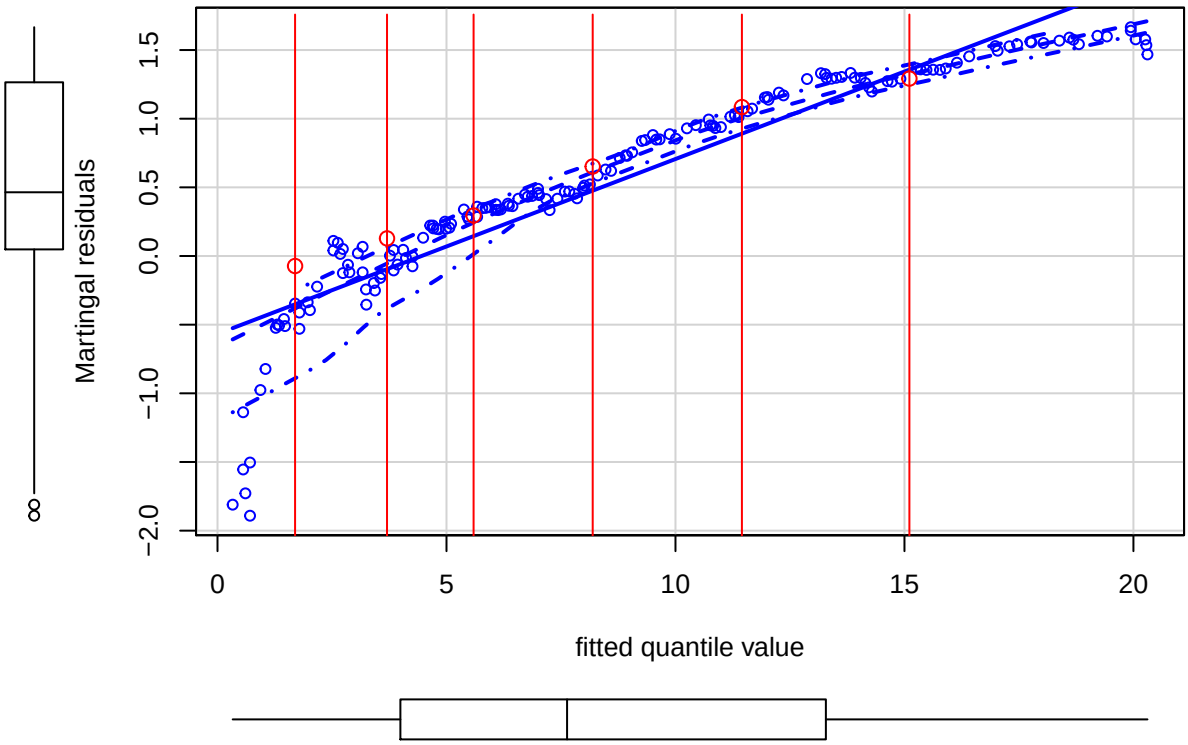
4) Benefit of treatment for each quantile

```
datafr <- rbind.data.frame(Moy,inf,sup)
names(datafr) <- c( "0.1", "0.2", "0.3", "0.4", "0.5","0.6")
p <- ggplot(stack(datafr), aes(x=factor(ind,levels=names(datafr)), y=values))+geom_boxplot()
p+labs(x="Quantile",y="Benefit in months")
```



5) Check linearity of the model

```
x <- c(0.1,0.2,0.3,0.4,0.5,0.6)
Rq <- crq(Surv(time,event)~tmt.arm.number,data=data_ICI_Rittmeyer,method="Pen")
tau <- Rq$sol["tau",][1:160]
obs <- unname(quantile(data_ICI_Rittmeyer$time,tau))
obss <- unname(quantile(data_ICI_Rittmeyer$time,x))
q<- Rq$sol["Qhat",][1:160]
q1 <- Rq$sol["Qhat",][c(24,46,69,91,114,136)]
residu <- obs-q
residus <- obss-q1
scatterplot(q,log(abs(residu)),xlab="fitted quantile value",ylab="Martingal residuals")
points(q1,log(abs(residus)),col='red')
for(i in 1:6){
  abline(v=q1[i],col="red")
}
```



6) Testing equality of two groups for a given quantile using survival Kaplan Meier function

For a given quantile, we can rely on 2-samples test derived for the median survival as detailed below. Testing for equality of median was derived in (Tang et al.,Chen et al.) Once the desired quantile is identified, the methodology can be adapted. These tests, designed for detecting the difference of the median survival times, can be readily extended to compare survival quantiles.

Therefore, let's assume that

$$\widehat{F}_1^{-1}(q) = \inf\{t : \widehat{F}_1(t) = 1 - \widehat{S}_1(t) \geq q\}$$
$$\widehat{F}_2^{-1}(q) = \inf\{t : \widehat{F}_2(t) = 1 - \widehat{S}_2(t) \geq q\}, \forall q \in [0, 1]$$

where $\widehat{S}_1$ and $\widehat{S}_2$ are respectively the estimated of the survival functions S_1 for non treated groups and S_2 for treated groups using the Kaplan Meier method, $widehat F_1$ and $\widehat{F}_2$ are right continuous, piece-wise constant estimators of F_1 and F_2 respectively.

Testing the equality of the quantile between the two groups is equivalent to testing the null hypothesis

$$H_0 : F_1^{-1}(q) = F_2^{-1}(q)$$
$$F_2\{F_1^{-1}(q)\} = q$$

As pointed out by Kosorok et al., $\sqrt{n}(F_2\{F_1^{-1}(q)\} - q)$ is asymptotically a zero-mean Gaussian process with variance σ^2.

We estimated the variance $\hat{\sigma}^2$ using re-sampling bootstrap method. The following statistic test

$$\frac{(\widehat{F}_2\{\widehat{F}_1^{-1}(q)\} - q)^2}{\sigma^2}$$

follows a χ^2-distribution with 1 degrees of freedom.

We applied this test in our data set at the quantile level 0.6 highly significant with a p-value $< 10^{-4}$ of $\%1.892087e^{-05}$, which indicates a significant difference at the quantile level 0.6 survival time between the two treatment groups.

R code

```
quantileTest <- function(time,event,treat,q,B=1000,seed=1234){
  set.seed(seed)
  Mesdon <- cbind.data.frame(time=time,event=event,treat=treat)
  fit1 <- survfit(Surv(Mesdon$time[Mesdon$treat==0],Mesdon$event[Mesdon$treat==0])~1,conf.type ="none")
  fit2 <- survfit(Surv(Mesdon$time[Mesdon$treat==1],Mesdon$event[Mesdon$treat==1])~1,conf.type = "none")
  F1.inv <- unname(quantile(fit1, prob = q))
  F2.inv <- unname(quantile(fit2, prob = q))

  # Calculate F2(F1.inv(p))
  Qp <- function(t1,c1, t2, c2) {
    fit1 <- survfit(Surv(t1, c1)~1, conf.type = "none")
    fit2 <- survfit(Surv(t2, c2)~1, conf.type = "none")
    F1.inv <- unname(quantile(fit1, prob=q))
    if (is.na(F1.inv)) {
      warning(paste0("Error"))
      F1.inv <- max(t1)
    }
    F2  <- stepfun(fit2$time, c(0, 1-fit2$surv)) #CDF of F2
    out <- F2(F1.inv) #F2(F1.inv(p))
    return(out)
  }

  Q <- Qp(Mesdon$time[Mesdon$treat==0],Mesdon$event[Mesdon$treat==0],
          Mesdon$time[Mesdon$treat==1],Mesdon$event[Mesdon$treat==1])

  # Bootstrap
  b.est <- numeric(B)
  for (i in 1:B) {
    boot1    <- sample(1:length(Mesdon$time[Mesdon$treat==0]),replace =TRUE)
    t1.boot  <- Mesdon$time[Mesdon$treat==0][boot1]
    c1.boot  <- Mesdon$event[Mesdon$treat==0][boot1]
    boot2    <- sample(1:length(Mesdon$time[Mesdon$treat==1]),replace = TRUE)
    t2.boot  <- Mesdon$time[Mesdon$treat==1][boot2]
    c2.boot  <- Mesdon$event[Mesdon$treat==1][boot2]
    b.est[i] <- Qp(t1.boot,c1.boot,t2.boot,c2.boot)
  }

  se   <- sd(b.est)
  Z<- (Q-q)^2/se^2
  pval <- 1-pchisq(Z,1)
```

```
  return(pval)
}


## Application of the test with our data at 0.6 quantile
quantileTest(time=data_ICI_Rittmeyer$time,event=data_ICI_Rittmeyer$event,
             treat = data_ICI_Rittmeyer$tmt.arm.number,q=0.6)
```

```
## [1] 1.20207e-05
```