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

Bassirou Mboup, Christophe Le Tourneau, Aurélien Latouche

06/07/2020

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) Importing the reconstructed data set

We use the algorithm of Guyot. al 2012 to reconstruct individual-level time-to-event data based on the published Kaplan–Meier curves of the randomized controlled trial (Rittmeyer et al. 2017).

The R code of the algorithm is available at https://www.mskcc.org/sites/default/files/node/137932/documents/2017-04-20-14-31-36/dataexample.zip

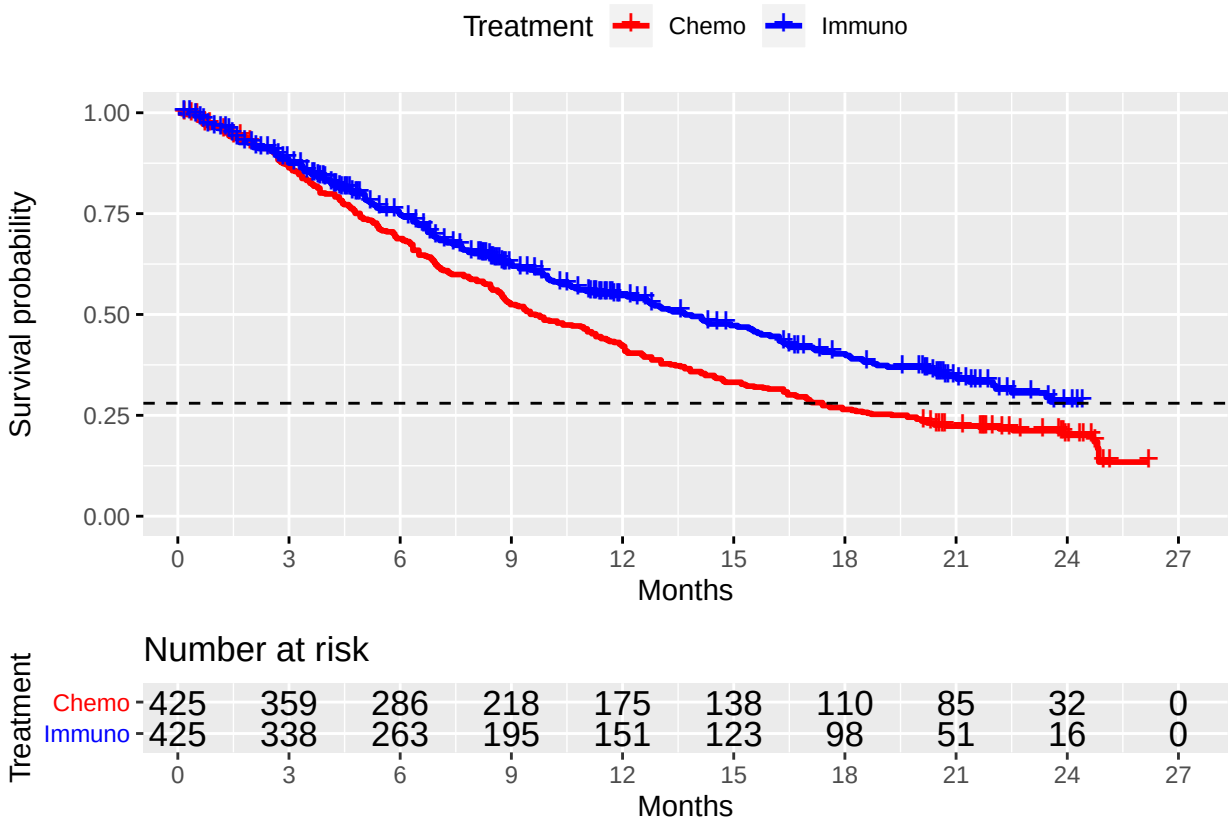
After reconstruction, we get in this dataset the following variables.

- time : vector of observed failure times e.g (death or censored).
- event: vector of indicator of status (1 for death and 0 for censoring).
- tmt.arm.number: vector of treatment indicator (binary with 1 for immunotherapy).
- treatment.type: the type of treatment (char) 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) Estimation of quantile regression coefficients

We generally observe n iid replicates of $(\widetilde{T},\ \delta,\ Z)$, denoted by $\{\widetilde{T}_i,\ \delta_i,Z_i,i=1,\ldots,n\}$, where $\widetilde{T}=\min(T,C)$ the observed failure time, $\delta=I_{(T\leq C)}$ the censoring indicator ($\delta=1$ if event and $\delta=0$ if censored) and C is the right censored time.

The regression parameters are estimated by minimizing an asymmetric linear loss function for a specific quantile, τ,

$$\widehat{\beta}(\tau)=\arg\min_{\beta\in\mathbb{R}^2}\sum_{i=1}^{n}\rho_\tau\left(\widetilde{T}_i-\min(C_i,X_i^T\beta)\right)$$

where $\beta=(\beta_0,\beta_1)$, $X=(1,Z)$ and $\rho_\tau(u)=u[\tau-I_1(u<0)]$ is the asymmetric linear loss function.

Two approaches were suggested to estimate censored quantile regression model however Koenker et al. shows that their properties were close.

As pointed out in Peng and Huang, it is difficult to produce an estimate of variance asymptotic of the estimates by quantile regression, because the covariance matrix of the $\sqrt{n}(\widehat{\beta}(\tau)-\beta(\tau))$ bound process involves an unknown density function. Bootstrapping procedures are also frequently used for variance estimation under quantile regression. Simple bootstrapping procedures based on resampling with replacement also seem to have satisfactory performance Portnoy et al. However, we used the simple bootstrap approach to estimate the variance as well as the confidence intervals of $\widehat{\beta}(\tau)=(\widehat{\beta_0}(\tau),\widehat{\beta_1}(\tau))$.

The procedure of the bootstrap is described as follows : we estimated the coefficient $\beta_1(\tau)$ by using the **crq** function in the **quantreg package R**. A number of estimates say $\{\hat{\beta}_1^{(b)}(\tau)\}_{b=1,\ldots,B}$ for each quantile τ, are obtained by repeating this procedure $B = 1000$ times. The variance of $\beta_1(\tau)$ can be estimated by the sample variance of $\{\hat{\beta}_1^{(b)}(\tau)\}_{b=1,\ldots,B}$. Finally, the confidence intervals for $\hat{\beta}_1(\tau)$ are based on the empirical percentiles of $\{\hat{\beta}_1^{(b)}(\tau)\}_{b=1,\ldots,B}$.

3) Application of quantile regression for survival data

In the following R output, test of the coefficients compares this coefficients to 0 for a given quantile τ. For the coefficient $\beta_1(\tau)$, this test rejects or not the equality of the two treatment groups for a given quantile τ. This test is based of the Wald test and the variance of the coefficients is obtained using resampling bootstraps method.

Thus, we do not reject the hypothesis of equality of the two groups at 0.1 quantile (P.value=0.9470) illustrated by the overlap of the curves at this quantile.

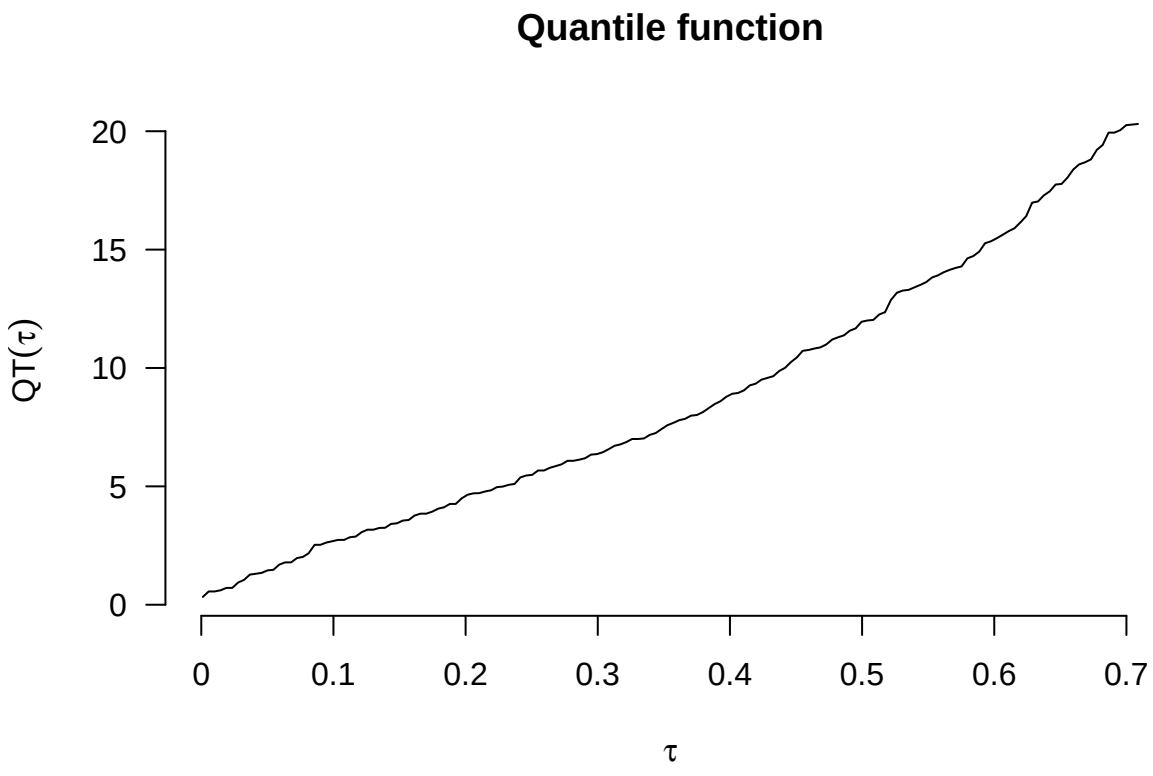
We reject this hypothesis at 0.6 quantile (P.value $< 10^{-4}$).

```
set.seed(123456)
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  1.75658  2.81841  0.27088   9.97646  0.00000
## tmt.arm.number -0.01844 -0.13442  0.95376  0.27760  -0.06642  0.94704
##
## tau: [1] 0.2
##
## Coefficients:
##                Value   Lower Bd Upper Bd Std Error T Value Pr(>|t|)
## (Intercept)    4.15397 3.30234  6.26304  0.75529   5.49982 0.00000
## tmt.arm.number 0.88255 0.05166  2.45479  0.61305   1.43960 0.14998
##
## tau: [1] 0.3
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     5.86294  5.75690  6.26086  0.12857  45.60287  0.00000
## tmt.arm.number  1.02312  0.55678  2.04722  0.38022   2.69086  0.00713
##
## 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 -0.49612  4.36664  1.24052   1.71169  0.08695
##
```

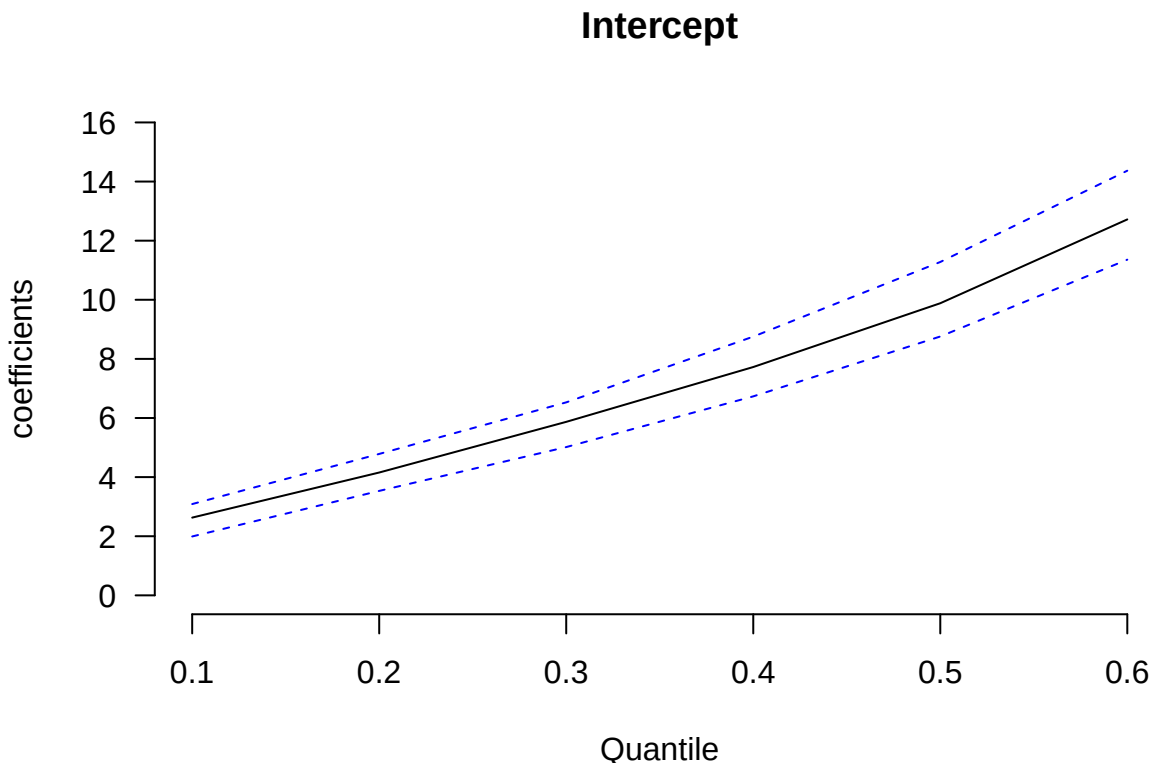
```
## tau: [1] 0.5
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)     9.78031  8.72650 10.71357  0.50691  19.29381  0.00000
## tmt.arm.number  4.35360  2.26212  6.58893  1.10380   3.94420  0.00008
##
## tau: [1] 0.6
##
## Coefficients:
##                Value    Lower Bd Upper Bd Std Error T Value  Pr(>|t|)
## (Intercept)    12.69400 10.75741 13.09069  0.59524  21.32596  0.00000
## tmt.arm.number  5.46031  3.04456  6.91123  0.98641   5.53551  0.00000

## 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)
```

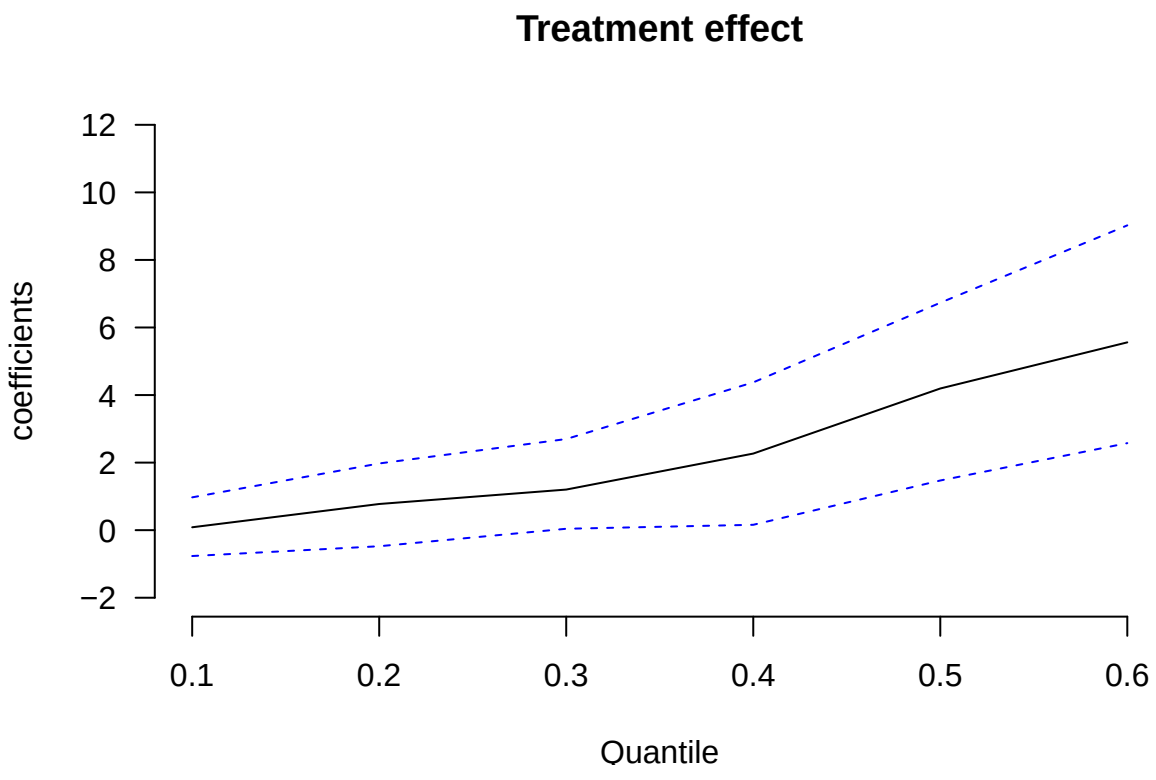


4) Resampling for confidence interval of coefficients

```
############ data for bootstrapping with 1000 replications ################
load("Mat.RData")
load("Moy.RData")
load("Mat1.RData")
load("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)
```



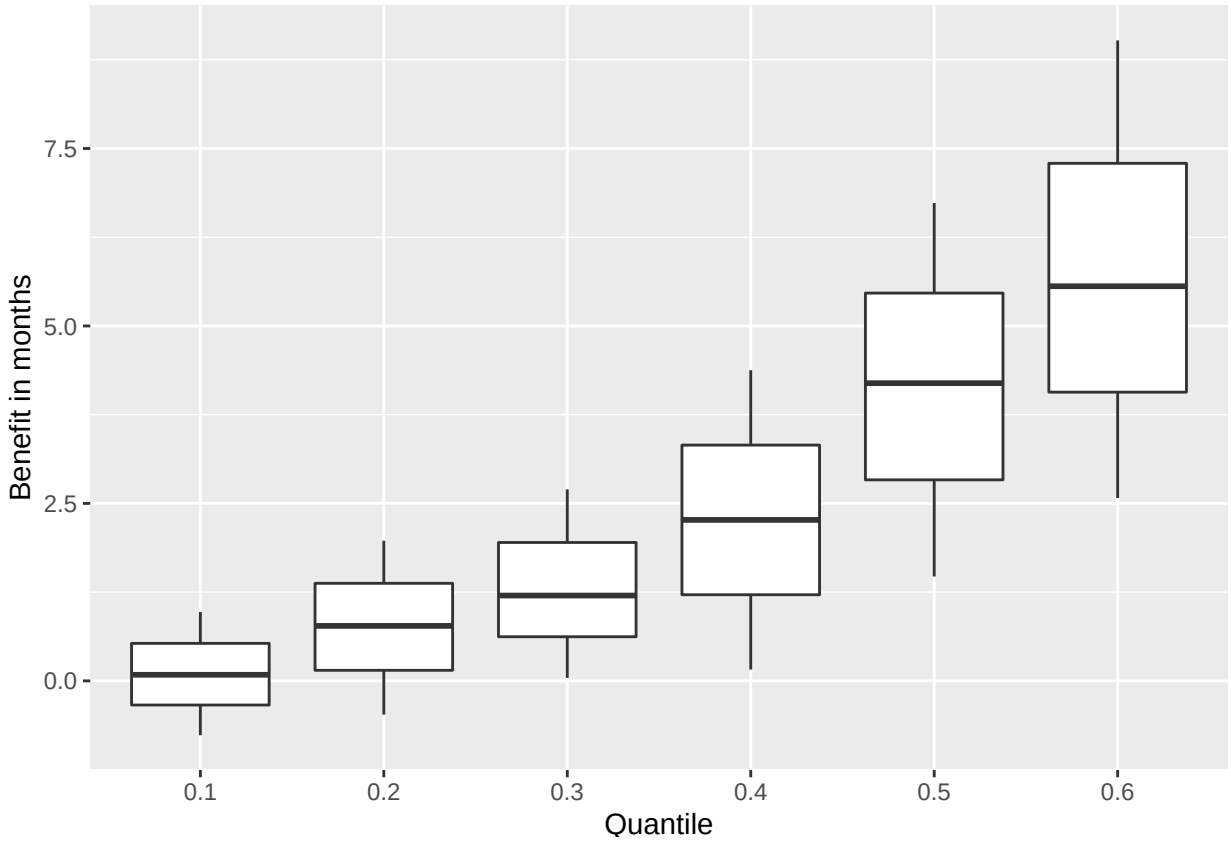
```r
## Variation of the treatment effect as function of each quantile available and confidence
# intervals
b <- matrix(data=NA,ncol=6,nrow = 1000)
for(i in 1:6){
b[,i] <-Mat[,i][order(Mat[,i])]
}

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)
```



5) Benefit of treatment for each quantile

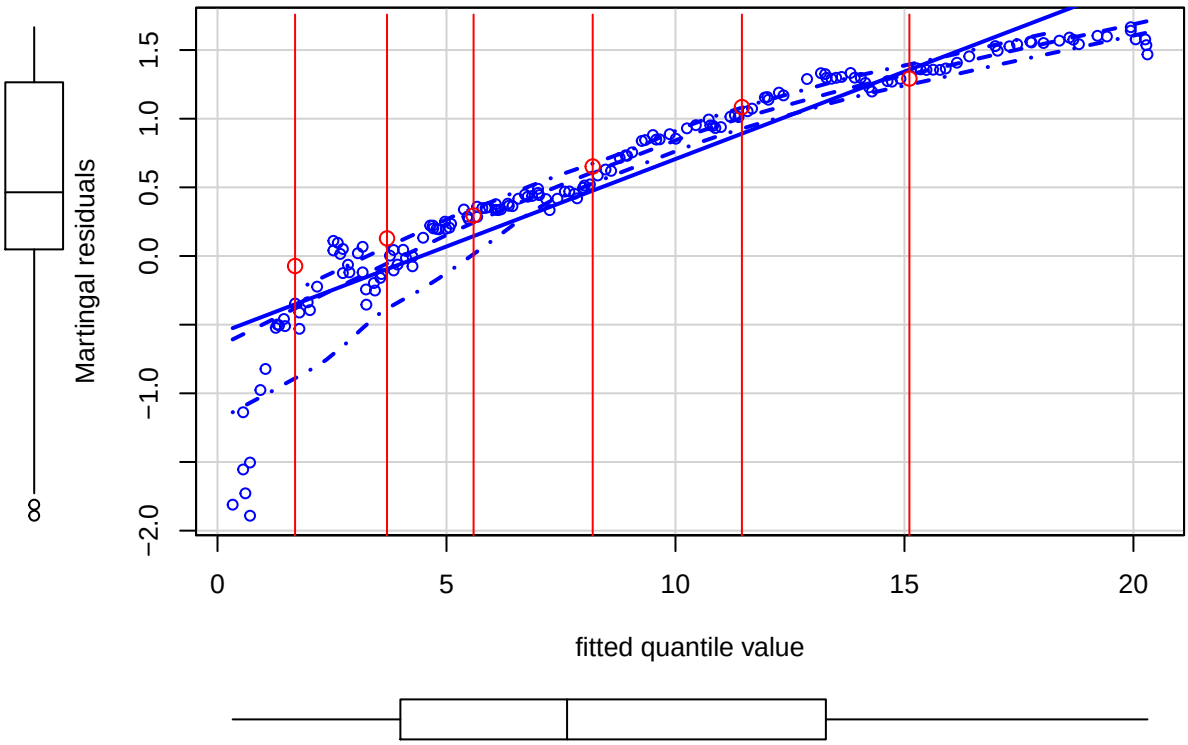
```r
datafr <- rbind.data.frame(Moy,inf,sup)
names(datafr) <- c( "0.1", "0.2", "0.3", "0.4", "0.5","0.6")
```

```
p <- ggplot(stack(datafr), aes(x=factor(ind,levels=names(datafr)), y=values))+geom_boxplot()
p+labs(x="Quantile",y="Benefit in months")
```



6) Visual check of the linearity assumption

```
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")
}
```



7) 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 Kaplan Meier estimate of the survival functions S_1 for untreated groups and S_2 for treated groups.

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 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
```