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

Bassirou Mboup, Christophe Le Tourneau, Aurélien Latouche

07/05/2020

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) 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)
```
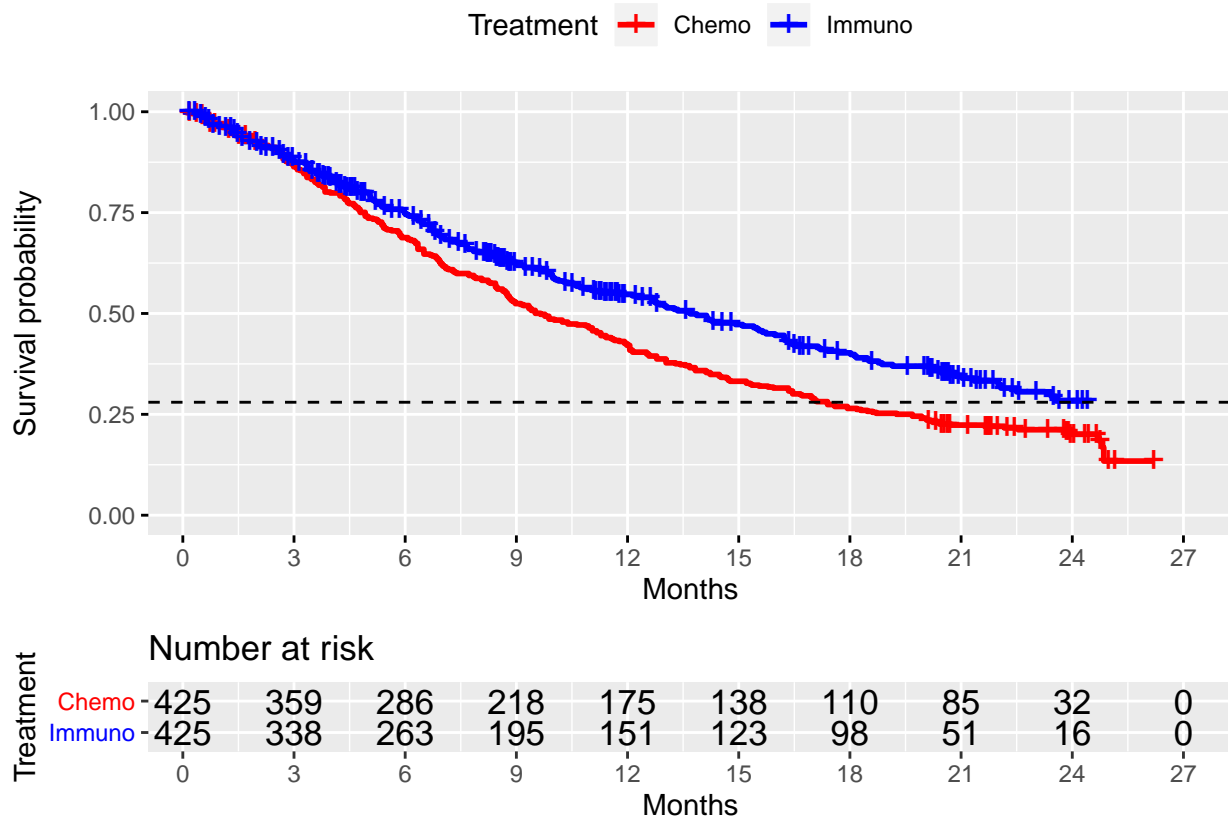

2) Application of quantile regression for survival data

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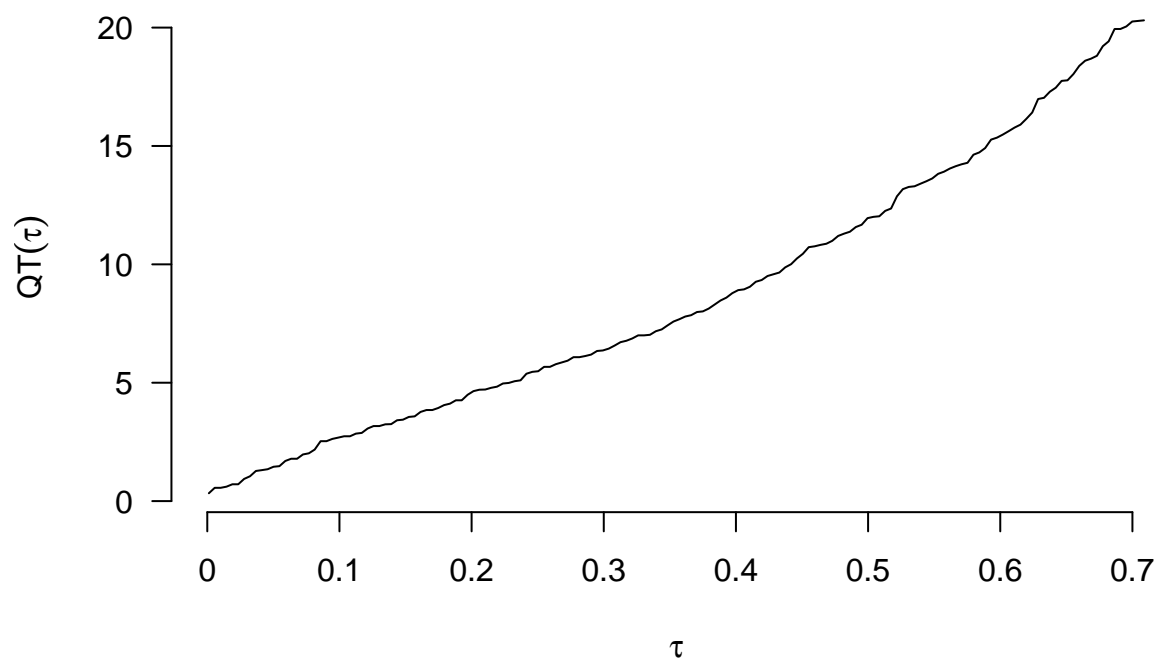
```r
# Quantile function
tau <- Rq$sol["tau",][1:160]
q<- Rq$sol["Qhat",][1:160]
plot(tau,q,type="l",xlab = expression(tau),ylab = expression(QT(tau)),
     main="Quantile function",axes=FALSE)
axis(1,at=seq(from=0,to=0.7,by=0.1),labels=seq(from=0,to=0.7,by=0.1),las=1)
axis(2,at=seq(from=0,to=20,by=5),labels=seq(from=0,to=20,by=5),las=2)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

Quantile function

3) Resampling for confidence interval of coefficients

```r
############ 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)
```
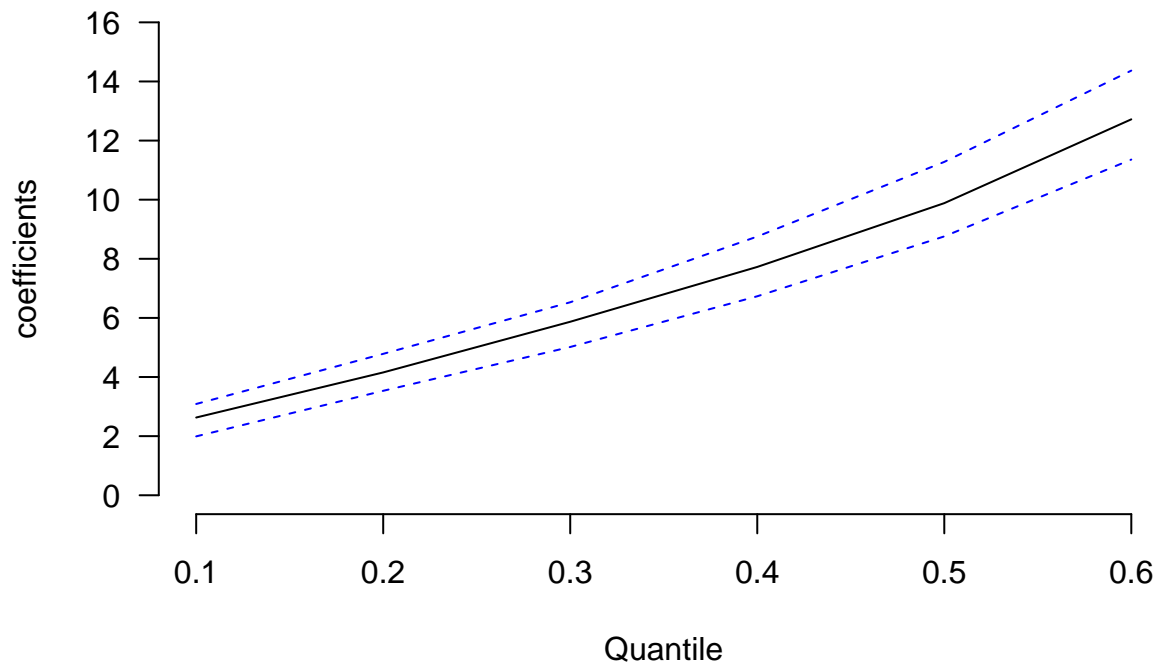
Intercept


```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)
```
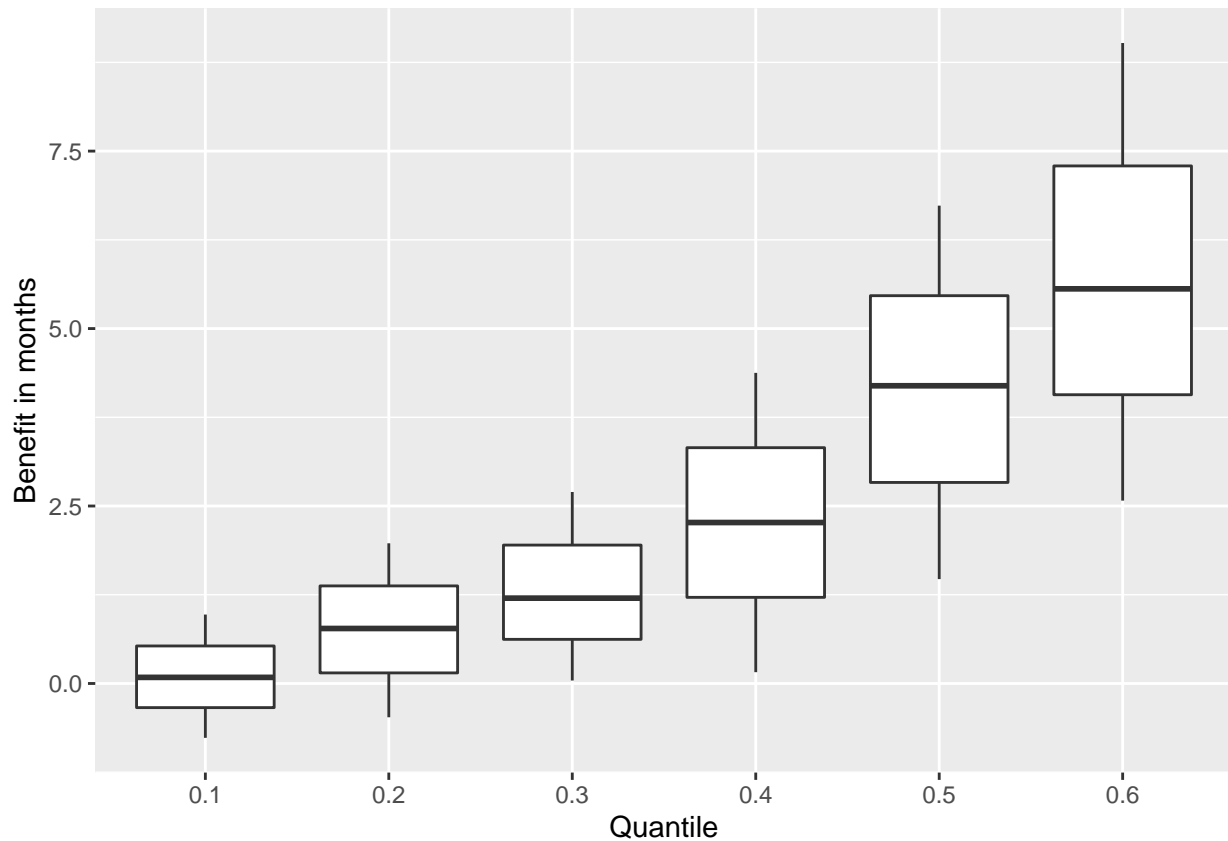
Treatment effect
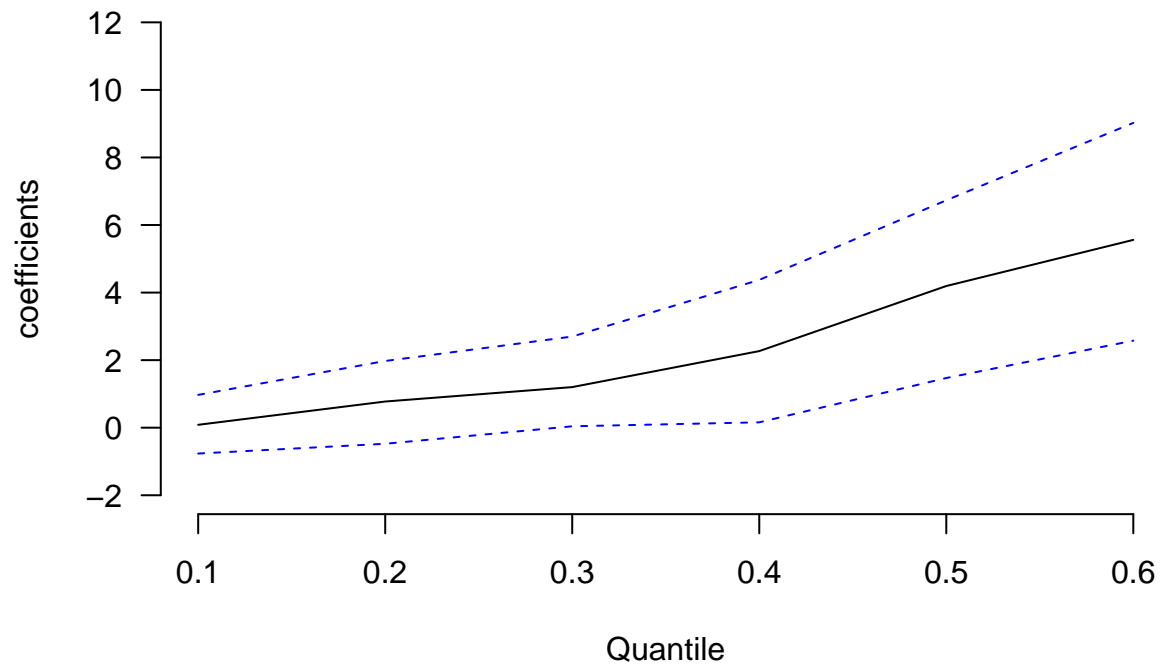
4) Benefit of treatment for each quantile

```
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) Visual check of the linearity assumption

```r
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")
}
```
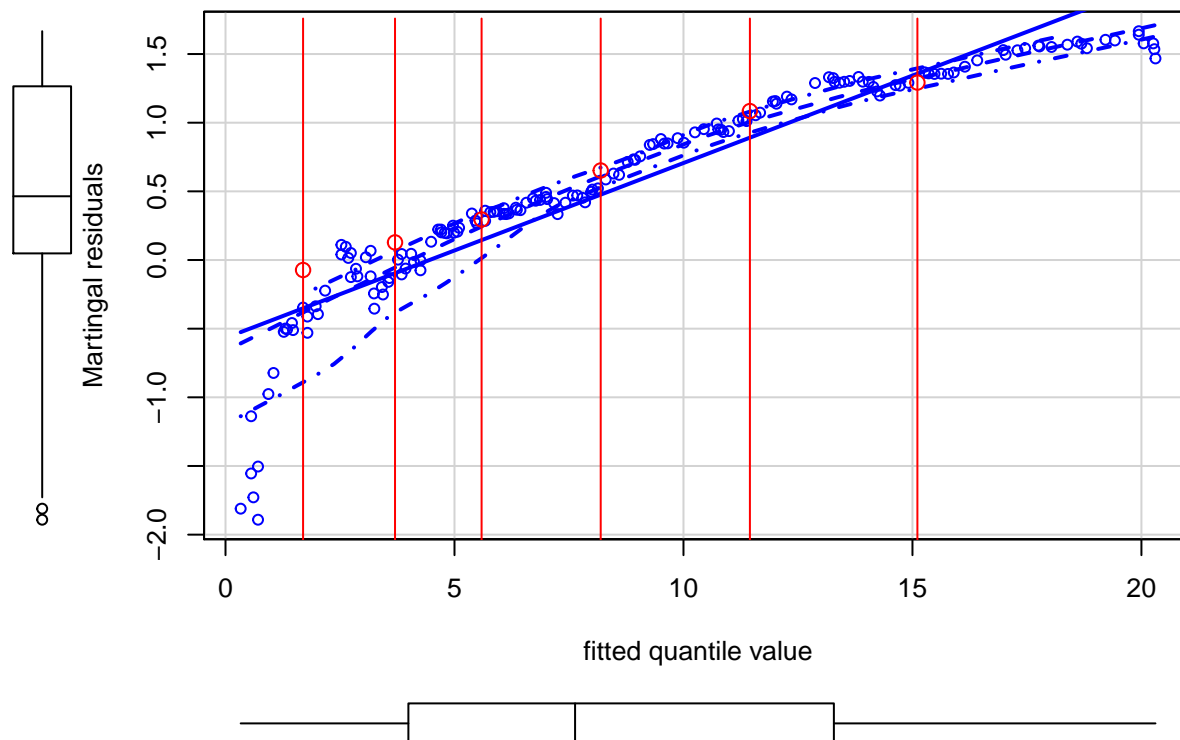
```r
load("ImmunoRit.RData")
library(knitr)
library(riskRegression)
```

```
## riskRegression version 2020.02.05
```

```r
db <- ImmunoRit
cox1<-coxph(Surv(time,event)~Treat, data=db,x=TRUE)

db2<-mutate(db,Treat=0)

pr<-predictCox(cox1,times=db2$time,newdata=db2,type="cumhazard")
cumhaz<-pr$cumhazard[1,]



beta1 <- log(0.59)
lamda <- cumhaz
beta2 <- log(0.74)
beta3 <- log(0.36)

genereTime <- function(x,lamda,trt,uni){
  return(-log(uni)/(lamda*exp(beta2*trt+beta1*x+beta3*x*trt)))
}

ttt <- ImmunoRit$Treat
n=850
```

```r
PDL1 <- rep(NA,n)
for(i in 1:n) {
  x <- -1
  while(x<0){
    uni <- runif(1,0,1)
    x <- uniroot((function(x)(genereTime(x,lamda[i],ttt[i],uni)-ImmunoRit$time[i]))),lower=-10,upper=10,

  }
  PDL1[i] <- x
}

summary(PDL1)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.006002  0.884266  2.014230  3.095343  4.651668 18.552966
```

```r
ImmunoRit$PDL1 <- PDL1

cox<-coxph(Surv(time,event)~Treat*PDL1,data=db,x=TRUE)
summary(cox)
```

```
## Call:
## coxph(formula = Surv(time, event) ~ Treat * PDL1, data = db,
##     x = TRUE)
##
##   n= 850, number of events= 555
##
##                   coef exp(coef)  se(coef)        z Pr(>|z|)
## Treat       -2.485766  0.083262  0.159190 -15.615  < 2e-16 ***
## PDL1        -0.044406  0.956566  0.002141 -20.743  < 2e-16 ***
## Treat:PDL1  -0.628000  0.533658  0.149829  -4.191 2.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## Treat        0.08326     12.010   0.06095    0.1137
## PDL1         0.95657      1.045   0.95256    0.9606
## Treat:PDL1   0.53366      1.874   0.39786    0.7158
##
## Concordance= 0.752  (se = 0.011 )
## Likelihood ratio test= 545.1  on 3 df,   p=<2e-16
## Wald test            = 483.6  on 3 df,   p=<2e-16
## Score (logrank) test = 682.7  on 3 df,   p=<2e-16
```

```r
Rq_1 <- crq(Surv(time,event)~Treat*PDL1,data=ImmunoRit,method="PengHuang")
result <- summary(Rq_1,tau=c(0.1,0.20,0.30,0.40,0.50,0.6))
kable(result[[6]][2]$coefficients)
```

	Value	Lower Bd	Upper Bd	Std Error	T Value	Pr(>|t|)
(Intercept)	1.185667	0.5896168	1.880427	0.3292943	3.6006290	0.0003174
Treat	1.092907	-1.0174670	3.394264	1.1254624	0.9710737	0.3315116
PDL1	2.663467	2.4792105	2.797365	0.0811633	32.8161600	0.0000000
Treat:PDL1	8.050838	7.2333462	12.013197	1.2193721	6.6024454	0.0000000