Treatment effect of 6xRCHOP21+2R versus 6xRCHOP21 of patients with DLBCL diagnosed between 2014 and 2018

Carolien C.H.M. Maas

July 2023

# Load packages

# remove previous history  
rm(list = ls(all.names=TRUE))  
  
# Load packages  
library(haven) # import STATA file  
library(lubridate) # adding weeks to dates  
library(mice) # imputation  
library(rms) # rcs  
library(survival) # Kaplan-Meier curve  
library(survminer) # plot Kaplan-Meier  
library(openxlsx) # write to Excel  
library(ggplot2) # plots  
library(RISCA) # ipw.log.rank  
library(ggpubr) # plots patient characteristics  
library(patchwork) # combine plots  
set.seed(100)

# Load data

DLBCL.orig <- haven::read\_dta(paste0(file.path, "Data/analyses\_1007.dta"))  
cat("Number of included patients:", nrow(DLBCL.orig), "\n")

## Number of included patients: 1715

##### Exclusion of patients

cat(" Missing stop date :" , sum(is.na(DLBCL.orig$txstop\_rit)), "\n",  
 "Missing start date:", sum(is.na(DLBCL.orig$txstart\_rchop21)), "\n")

## Missing stop date : 21   
## Missing start date: 1

DLBCL.raw <- DLBCL.orig[!is.na(DLBCL.orig$txstop\_rit),]  
DLBCL.raw <- DLBCL.raw[!is.na(DLBCL.raw$txstart\_rchop21),]

# Exclude 3 patients with incorrect stop date

## Incorrect stop date: 3

In total, 1691 patients are included of which received   
 6xRCHOP21: 757   
 6xRCHOP21+1R: 41   
 6xRCHOP21+2R: 893

##### Recode

Fix typo’s, i.e. txstop\_rit one year too many

Make gender binary 0/1

DLBCL.raw$geslacht <- DLBCL.raw$geslacht-1

If center is missing, it is not an academic center.

DLBCL.raw$center[is.na(DLBCL.raw$center)] <- 0

Define time at which patients are at risk, which is the end of the possible 8th cycle (second additional rituximab treatment).

# at risk at start of 6th/7th/8th cycle  
DLBCL.raw$at.risk <- as.Date(DLBCL.raw$txstop\_rit)   
  
# add 63 days for 6xRCHOP21  
# add 42 days for 6xRCHOP21+1R  
# add 21 days for 6xRCHOP21  
DLBCL.raw <- DLBCL.raw |>  
 mutate(  
 at.risk = case\_when(  
 RCHOP21==0 ~ txstop\_rit + lubridate::days(63),  
 RCHOP21==1 ~ txstop\_rit + lubridate::days(42),  
 RCHOP21==9 ~ txstop\_rit + lubridate::days(21)  
 ))  
  
# DLBCL.raw[DLBCL.raw$RCHOP21==0, "at.risk"] <- as.Date(DLBCL.raw$txstop\_rit)[DLBCL.raw$RCHOP21==0] + lubridate::days(21)  
# DLBCL.raw[DLBCL.raw$RCHOP21==1, "at.risk"] <- as.Date(DLBCL.raw$txstop\_rit)[DLBCL.raw$RCHOP21==1] - lubridate::days(21)

Define recurrence and survival time measured from the date at which the first additional rituximab cycle could have been administered (at.risk) and the date of event-free survival (efsdat).

DLBCL.raw$EFS.time <- as.numeric(difftime(DLBCL.raw$efsdat, DLBCL.raw$at.risk, unit="weeks"))/52  
DLBCL.raw$OS.time <- as.numeric(difftime(DLBCL.raw$osdat, DLBCL.raw$at.risk, unit="weeks"))/52

Exclude patients that had an event before the at.risk date

cat("Number of patients with event before stop date:", sum(DLBCL.raw$EFS.time<0), "\n",  
 "Number of patients with death before stop date:", sum(DLBCL.raw$OS.time<0), "\n")

## Number of patients with event before stop date: 128   
## Number of patients with death before stop date: 47

# OMIT patients with death before stop date  
DLBCL.raw <- DLBCL.raw[-which(DLBCL.raw$OS.time<0),]  
  
# add 6xRCHOP21 + 1R to the 6XRCHOP21 + 2R group  
table(DLBCL.raw$RCHOP21)

##   
## 0 1 9   
## 720 885 39

DLBCL.raw$RCHOP21[DLBCL.raw$RCHOP21==9] <- 1  
  
cat("Number of patients with event in first 30 days:", sum(DLBCL.raw$EFS.time<30/365&DLBCL.raw$RCHOP21==0), "\n",  
 "Number of patients with death in first 30 days:", sum(DLBCL.raw$OS.time<30/365&DLBCL.raw$RCHOP21==0), "\n")

## Number of patients with event in first 30 days: 64   
## Number of patients with death in first 30 days: 14

Make all variables numeric.

DLBCL.raw <- as.data.frame(lapply(DLBCL.raw, as.numeric, na.rm=FALSE))

Ensure categorical variables are considered as factors, such that dummies are created for each level in regression.

DLBCL.raw$sescat <- as.factor(DLBCL.raw$sescat)  
DLBCL.raw$stadium <- as.factor(DLBCL.raw$stadium)  
DLBCL.raw$region <- as.factor(DLBCL.raw$region)

### Describe data

to.be.imputed.DLBCL <- DLBCL.raw[, c("RCHOP21", "geslacht", "lft", "stadium",  
 "ipi\_ldh", "ipi\_extra", "ipi\_total", "ipi\_ps", "region",   
 "sescat", "malignancy", "center", "efsi", "EFS.time", "dood", "OS.time",  
 "jaar")]

### Impute missing data with single imputation

DLBCL <- mice::complete(mice::mice(to.be.imputed.DLBCL, m=1, print=FALSE))

Save the distributions of all variables, which is necessary for Harrell summaries.

dd <- rms::datadist(DLBCL)  
options(datadist="dd")

##### Define IPI score

If performance score, elevated LDH, or >1 extranodal site is missing, do not add to IPI score, e.g. set that item of the IPI score to 0.

DLBCL$ipi\_ps\_na <- DLBCL$ipi\_ps  
DLBCL$ipi\_ps\_na[is.na(DLBCL.raw$ipi\_ps)] <- 0  
DLBCL$ipi\_ldh\_na <- DLBCL$ipi\_ldh  
DLBCL$ipi\_ldh\_na[is.na(DLBCL.raw$ipi\_ldh)] <- 0  
DLBCL$ipi\_extra\_na <- DLBCL$ipi\_extra  
DLBCL$ipi\_extra\_na[is.na(DLBCL.raw$ipi\_extra)] <- 0  
table(DLBCL$ipi\_total)

##   
## 0 1 2 3   
## 378 437 506 323

Define some variables binary instead of categorical

DLBCL$stage2 <- as.numeric(DLBCL$stadium==2)  
DLBCL$stage3 <- as.numeric(DLBCL$stadium==3)  
DLBCL$stage4 <- as.numeric(DLBCL$stadium==4)  
for (regio in 1:9){  
 DLBCL[, paste0("region", regio)] <- as.numeric(DLBCL$region==regio)  
}  
DLBCL$ses.low <- as.numeric(DLBCL$sescat==0)  
DLBCL$ses.medium <- as.numeric(DLBCL$sescat==1)  
DLBCL$ses.high <- as.numeric(DLBCL$sescat==2)

# Propensity score model

1. RCHOP21 indicates whether the patient received 6xRCHOP21 or 6xRCHOP21+2R.
2. region indicates the region where the patient is treated.
3. The International Prognostic Index (IPI) consists of

* an indicator of age > 60 (lft)
* stage 3 or 4 (stadium)
* WHO score ECOG/Zubrod status 2, 3, or 4 (ipi\_ps)
* elevated serum LDH (ipi\_ldh)
* more than 1 extranodal site (ipi\_extra).

1. malignancy indicates if a patient had one or more previous malignancy
2. center indicates if the center that the patient is treated in was academic or not
3. ses indicates Social Economic Status, which is a proxy for comorbidity, and is divided into low, medium, and high.

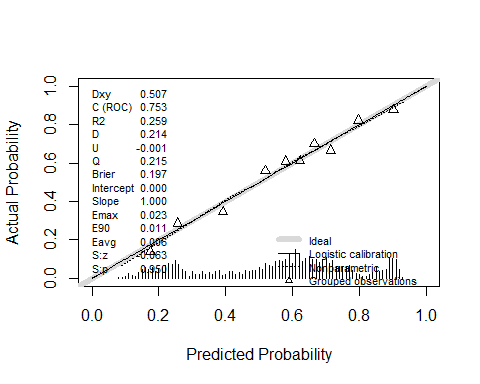
# fixed effects propensity score model  
PS.formula.FE <- RCHOP21~geslacht+lft+stadium+ipi\_ps\_na+ipi\_ldh\_na+ipi\_extra\_na+malignancy+region+center+ses.low+ses.high  
PS.model <- stats::glm(PS.formula.FE, family="binomial", data=DLBCL)  
# summary(PS.model)

The propensity score (), i.e. the predicted probability of treatment.

DLBCL$e.i <- predict(PS.model, type="response")

Assess the performance of the propensity score model.

rms::val.prob(DLBCL$e.i, DLBCL$RCHOP21, g=10)



## Dxy C (ROC) R2 D D:Chi-sq   
## 5.068513e-01 7.534256e-01 2.587204e-01 2.138655e-01 3.525949e+02   
## D:p U U:Chi-sq U:p Q   
## NA -1.216545e-03 1.591616e-12 1.000000e+00 2.150821e-01   
## Brier Intercept Slope Emax E90   
## 1.969624e-01 7.609287e-11 1.000000e+00 2.287073e-02 1.141416e-02   
## Eavg S:z S:p   
## 5.903087e-03 -6.252168e-02 9.501474e-01

png(file=paste0(file.path, "Results/PS.model.performance.png"), width=500, height=500, res=100)  
rms::val.prob(DLBCL$e.i, DLBCL$RCHOP21, g=10)

## Dxy C (ROC) R2 D D:Chi-sq   
## 5.068513e-01 7.534256e-01 2.587204e-01 2.138655e-01 3.525949e+02   
## D:p U U:Chi-sq U:p Q   
## NA -1.216545e-03 1.591616e-12 1.000000e+00 2.150821e-01   
## Brier Intercept Slope Emax E90   
## 1.969624e-01 7.609287e-11 1.000000e+00 2.287073e-02 1.141416e-02   
## Eavg S:z S:p   
## 5.903087e-03 -6.252168e-02 9.501474e-01

grDevices::dev.off()

## png   
## 2

# Write Propensity Score Model Table to Excel

##### Overlap assumption is satisfied

# orange: #EC7F12  
# purple: #9161BD  
# blue: #0D88AF  
overlap.assumption.plot <- ggplot2::ggplot(DLBCL, ggplot2::aes(x=e.i, fill=as.factor(RCHOP21)))+  
 ggplot2::geom\_histogram(color="#e9ecef", alpha=0.6, binwidth=0.02)+  
 ggplot2::theme\_light()+  
 ggplot2::theme(legend.title=ggplot2::element\_blank(), text=ggplot2::element\_text(size=12))+  
 ggplot2::scale\_fill\_manual(values=c("#EC7F12", "#9161BD"), labels=c("6xRCHOP21", "6xRCHOP21+2R"))+  
 ggplot2::ylab("Frequency")+  
 ggplot2::xlab("Propensity score")  
ggsave(filename=paste0(file.path, "Results/hist.PS.png"),  
 plot=overlap.assumption.plot,   
 width=8, height=6, dpi=300)

# Stabilized inverse propensity score weights

P.Z <- mean(DLBCL$RCHOP21)  
DLBCL$stabilized.w.i <- P.Z\*DLBCL$RCHOP21/DLBCL$e.i +  
 (1-P.Z)\*(1-DLBCL$RCHOP21)/(1-DLBCL$e.i)

Check balance before and after weighting using standardized mean difference

# define number of quantiles  
g <- 4  
main.titles <- c("0-1", "2", "3", "4-5")  
for (weights.name in c("stabilized.w.i")){ # stabilized.w.i.quantile  
 # covariates  
 vars <- c("geslacht",   
 "ipi\_ldh\_na",   
 "ipi\_ps\_na",   
 "ipi\_extra\_na",   
 "malignancy",   
 "center",  
 "lft",   
 paste0("region", 2:9),   
 paste0("stage", 2:4),   
 "ses.low", "ses.medium", "ses.high")  
 # construct a table  
 tabUnmatched <- tableone::CreateTableOne(vars=vars, strata="RCHOP21",   
 data=DLBCL, test=FALSE)  
 # show table with SMD  
 # print(tabUnmatched, smd=TRUE) # 736 - 890  
 # 11 covariates with important imbalance  
 # stats::addmargins(table(tableone::ExtractSmd(tabUnmatched) > 0.1))  
   
 # weighted data  
 weighted.data <- survey::svydesign(ids = ~ 1,   
 data = DLBCL,  
 weights = ~ DLBCL[, weights.name])  
 # construct a table  
 tabWeighted <- tableone::svyCreateTableOne(vars = vars,   
 strata = "RCHOP21",   
 data = weighted.data,   
 test = FALSE)  
 # show table with SMD  
 # print(tabWeighted, smd = TRUE) # 747.42 - 878.17  
 # 0 covariates with important imbalance  
 # stats::addmargins(table(tableone::ExtractSmd(tabWeighted) > 0.1))  
  
 # love plot for propensity scores based on all patients  
 love.df <- data.frame(names=c("Gender", "Elevated LDH", "Performance status",   
 "Extranodal sites", "Additional malignancies",   
 "Academic center", "Age",   
 paste("Region", 2:9),  
 paste("Stage", 2:4),  
 "Low SES", "Medium SES", "High SES"), # TODO: why SES no ref?  
 unweighted=as.numeric(tableone::ExtractSmd(tabUnmatched)),  
 weighted=as.numeric(tableone::ExtractSmd(tabWeighted)),  
 diff=tableone::ExtractSmd(tabUnmatched)-tableone::ExtractSmd(tabWeighted))  
 colnames(love.df) <- c("names", "unweighted", "weighted", "difference")  
 ordered.love.df <- love.df[order(love.df$difference, decreasing=FALSE),]  
 cat("Maximum Standardized Mean Difference:", max(ordered.love.df$weighted), "\n")  
   
 # ensure right order  
 ordered.love.df$names <- factor(ordered.love.df$names, levels=ordered.love.df$names)   
 ggsave(filename=paste0(file.path, "Results/Love.plot.all.png"),  
 plot=ggplot2::ggplot(data=ordered.love.df, ggplot2::aes(y=names, x=unweighted))+  
 ggplot2::geom\_point(x=ordered.love.df$unweighted, colour="#EC7F12", size=3)+  
 ggplot2::geom\_point(x=ordered.love.df$weighted, colour="#9161BD", size=3)+  
 ggplot2::geom\_vline(xintercept=0, linetype="dashed")+  
 ggplot2::ggtitle("Covariate balance for all patients")+  
 ggplot2::xlab("Standardized mean difference")+  
 ggplot2::ylab("")+  
 ggplot2::xlim(0, 1)+  
 ggplot2::theme\_classic()+  
 ggplot2::theme(plot.title=element\_text(hjust=0.5),  
 text=ggplot2::element\_text(size=17)),  
 width=8, height=6, dpi=300)  
   
 # plot before and after weighting in each risk stratum  
 for (i in 1:g){  
 # https://rdrr.io/cran/tableone/f/vignettes/smd.Rmd  
 tabUnmatched.q <- tableone::CreateTableOne(vars = vars,   
 strata="RCHOP21",   
 data = subset(DLBCL, DLBCL$ipi\_total==i-1),   
 test = FALSE)  
 weighted.data.q <- subset(weighted.data, DLBCL$ipi\_total==i-1)  
 tabWeighted.q <- tableone::svyCreateTableOne(vars = vars,  
 strata = "RCHOP21",  
 data = weighted.data.q,  
 test = FALSE)  
   
 # love plot for propensity scores based on all patients  
 love.df.q <- data.frame(names=c("Gender", "Elevated LDH", "Performance status",   
 "Extranodal sites", "Additional malignancies",   
 "Academic center", "Age",   
 paste("Region", 2:9),  
 paste("Stage", 2:4),  
 "Low SES", "Medium SES", "High SES"), # TODO: why SES no ref?  
 unweighted=as.numeric(tableone::ExtractSmd(tabUnmatched.q)),  
 weighted=as.numeric(tableone::ExtractSmd(tabWeighted.q)),  
 diff=tableone::ExtractSmd(tabUnmatched.q)-tableone::ExtractSmd(tabWeighted.q))  
 colnames(love.df.q) <- c("names", "unweighted", "weighted", "difference")  
 ordered.love.df.q <- love.df.q[order(love.df.q$difference, decreasing=FALSE),]  
   
 # ensure right order  
 ordered.love.df.q$names <- factor(ordered.love.df.q$names, levels=ordered.love.df.q$names)   
 ggsave(filename=paste0(file.path, "Results/Plots/Love.plot.", i, ".png"),  
 plot=ggplot2::ggplot(data=ordered.love.df.q, ggplot2::aes(y=names, x=unweighted))+  
 ggplot2::geom\_point(x=ordered.love.df.q$unweighted, colour="#EC7F12")+  
 ggplot2::geom\_point(x=ordered.love.df.q$weighted, colour="#9161BD")+  
 ggplot2::geom\_vline(xintercept=0, linetype="dashed")+  
 ggplot2::ggtitle(paste("Covariate balance for IPI score", main.titles[i]))+  
 ggplot2::xlab("Standardized mean difference")+  
 ggplot2::ylab("")+  
 ggplot2::xlim(0, 0.7)+  
 ggplot2::theme\_classic()+  
 ggplot2::theme(plot.title=element\_text(hjust=0.5)))  
 }  
   
 # before after weighting  
 png(file=paste0(file.path, "Results/Plots/Love.plot.quantiles.", weights.name, ".png"),  
 width=16, height=16, units="cm", res=300)  
 par(mar=rep(0, 4))  
 layout(matrix(1:4, ncol=2, byrow=TRUE))  
 for (i in 1:4){  
 plot(NA, xlim=0:1, ylim=0:1, xaxt="n", yaxt="n", bty="n")  
 img <- png::readPNG(paste0(file.path, "Results/Plots/Love.plot.", i, ".png"))  
 rasterImage(img, 0, 0, 1, 1)  
 }  
 grDevices::dev.off()  
}

## Maximum Standardized Mean Difference: 0.06450302

## Saving 5 x 4 in image  
## Saving 5 x 4 in image  
## Saving 5 x 4 in image  
## Saving 5 x 4 in image

# Compare PS in whole sample or risk-strata

compare.PS <- data.frame(ID=1:nrow(DLBCL), IPI.score=as.factor(DLBCL$ipi\_total),  
 all=DLBCL$e.i, risk.strata.PS=rep(NA, nrow(DLBCL)))  
for (i in 1:g){  
 PS.model.i <- stats::glm(PS.formula.FE, family="binomial",   
 data=DLBCL,  
 subset=(ipi\_total==i-1))  
  
 compare.PS[which(DLBCL$ipi\_total==i-1), "risk.strata.PS"] <- predict(PS.model.i, type="response")  
}  
compare.PS.plot <- ggplot(data=compare.PS, mapping=aes(x=all,   
 y=risk.strata.PS,   
 colour=IPI.score)) +  
 geom\_abline(intercept=0, slope=1, color="black") +  
 geom\_point() +  
 theme\_minimal() +   
 ylab("Propensity scores developed on each risk strata") +  
 xlab("Propensity scores developed on all patients") +  
 scale\_color\_discrete(name="IPI-score", labels=c("0-1", "2", "3", "4-5"))  
ggsave(filename=paste0(file.path, "Results/compare.PS.png"),   
 plot=compare.PS.plot, width=6, height=6, dpi=300)

# Recurrence and overall survival

### Set time-to-event

The event-free survival indicator (efsi) is 1 if the patient has a recurrence or if there is no response to initial treatment and second line treatment is administered, and 0 otherwise. The time until the event is the event-free survival time.

DLBCL$S.EFS <- survival::Surv(event=DLBCL$efsi, time=DLBCL$EFS.time)  
med.EFS.time <- stats::quantile(prodlim::prodlim(prodlim::Hist(EFS.time, efsi==1)~1,   
 data=DLBCL, reverse=TRUE))$quantiles.survival  
paste0(sprintf("%.2f", med.EFS.time[3, "quantile"]), " [IQR: ",  
 sprintf("%.2f", med.EFS.time[4, "quantile"]), "-",  
 sprintf("%.2f", med.EFS.time[2, "quantile"]), "]")

## [1] "4.41 [IQR: 3.80-5.29]"

Set the overall survival.

DLBCL$S.OS <- survival::Surv(event=DLBCL$dood, time=DLBCL$OS.time)  
med.FU.time <- stats::quantile(prodlim::prodlim(prodlim::Hist(OS.time, dood==1)~1,   
 data=DLBCL, reverse=TRUE))$quantiles.survival  
paste0(sprintf("%.2f", med.FU.time[3, "quantile"]), " [IQR: ",  
 sprintf("%.2f", med.FU.time[4, "quantile"]), "-",  
 sprintf("%.2f", med.FU.time[2, "quantile"]), "]")

## [1] "4.41 [IQR: 3.79-5.25]"

save(DLBCL, file=paste0(file.path, "Data/imputed.DLBCL.Rdata"), compress=TRUE)

### Helper functions

ylabs <- data.frame(outcome=c("EFS", "OS"),  
 label=c("EFS probability", "OS probability"))  
limit.year.EFS <- 5  
limit.year.OS <- 5  
fun.event <- function(lp, h0){  
 h <- h0\*exp(lp)  
 p <- 1-exp(-h)  
 return(p)  
}

### For each outcome, show:

1. Kaplan-Meier curves
2. If proportional hazard assumption is satisfied
3. Kaplan-Meier estimates, hazard ratios and RMST stratified by IPI score

DLBCL.OS <- DLBCL  
# OMIT negative event-times and first events  
EFS.limit <- 0  
DLBCL.EFS <- DLBCL[-which(DLBCL$EFS.time<EFS.limit),]  
for (weights.name in c("stabilized.w.i")){ # stabilized.w.i.quantile  
 for (outcome in ylabs$outcome){  
 # Survival outcome  
 S.temp <- eval(parse(text=paste0("DLBCL.", outcome, "$S.", outcome)))  
 limit.year <- eval(parse(text=(paste0("limit.year.", outcome))))  
   
 # Set the censoring of patients with follow-up time more than x years to "censored"  
 S.temp[S.temp[,1]>limit.year, 2] <- 0  
 # Set the follow-up time to x years for patients with follow-up time more than x years.  
 S.temp[S.temp[,1]>limit.year, 1] <- limit.year  
 assign(paste0("S.", outcome, ".limit"), S.temp)  
   
 # Kaplan-Meier plots  
 km <- survival::survfit(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),   
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 robust=TRUE, type="kaplan-meier", conf.type="log")  
 assign(paste0("km\_", outcome), km)  
  
 # Kaplan-Meier estimates at x-years  
 treat <- summary(km, times=limit.year)  
 all <- summary(survival::survfit(eval(parse(text=paste0("S.", outcome, ".limit ~ 1"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 robust=TRUE, type="kaplan-meier", conf.type="log"), times=limit.year)  
 # summary(km, times=limit.year)  
  
 # Log-rank test using stabilized weights is rejected,  
 # thus the survival between the two treatment groups is significantly different  
 LRT <- RISCA::ipw.log.rank(times=eval(parse(text=paste0("S.", outcome, ".limit[, 1]"))),   
 failures=eval(parse(text=paste0("S.", outcome, ".limit[, 2]"))),  
 variable=eval(parse(text=paste0("DLBCL.", outcome, "$RCHOP21"))),   
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))))  
   
 # Univariate Cox proportional hazards model with stabilized weights  
 cox <- survival::coxph(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 robust=TRUE)  
 assign(paste0("cox.", outcome), cox)  
  
 # Restricted Mean Survival Time Analysis  
 RMST <- survival:::survmean(survival::survfit(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 type="kaplan-meier", conf.type="log"),  
 rmean=limit.year)  
 dRMST <- diff(RMST$matrix[, "rmean"])  
 assign(paste0("dRMST.", outcome), dRMST)  
   
 # p-value RMST  
 # dRMST\_se <- sqrt(sum(RMST$matrix[, "se(rmean)"]^2/RMST$matrix[, "records"]))  
 # p.RMST <- 2\*pnorm(-abs(dRMST/dRMST\_se))  
   
 # Kaplan-Meier plot  
 LRT.text <- sprintf("%.3f", LRT$p.value)  
 HR.text <- sprintf("%.2f", exp(coef(cox)))  
 HR.text.lower <- sprintf("%.2f", exp(confint(cox)[, "2.5 %"]))  
 HR.text.upper <- sprintf("%.2f", exp(confint(cox)[, "97.5 %"]))  
 RMST.text <- sprintf("%.2f", dRMST)  
 RMST.text.lower <- sprintf("%.2f", dRMST-1.96\*sqrt(RMST$matrix[1, "se(rmean)"]^2+RMST$matrix[2, "se(rmean)"]^2))  
 RMST.text.upper <- sprintf("%.2f", dRMST+1.96\*sqrt(RMST$matrix[1, "se(rmean)"]^2+RMST$matrix[2, "se(rmean)"]^2))  
 # RMST.p <- sprintf("%.3f", p.RMST)  
 km\_plot <- survminer::ggsurvplot(km,  
 palette=c("#EC7F12", "#9161BD"),  
 ggtheme = theme(  
 axis.line = element\_line(colour = "black"),  
 panel.background = element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = element\_blank(), #remove major gridlines  
 panel.grid.minor = element\_blank(), #remove minor gridlines  
 legend.background = element\_rect(fill='transparent'), #transparent legend bg  
 legend.box.background = element\_rect(color = NA, fill='transparent') #transparent legend panel  
 ),  
 legend.labs=c("6xRCHOP21", "6xRCHOP21+2R"),  
 xlab="",  
 ylab=ylabs[ylabs$outcome==outcome, "label"],  
 risk.table=FALSE, conf.int="True",  
 robust=TRUE, pval=FALSE)$plot+  
 ggplot2::annotate("text", x=0, y=0.05,  
 label=paste0("p-value of weighted log-rank test = ",  
 LRT.text, "\n",  
 "HR = ", HR.text,  
 paste0(", 95% CI: [",  
 HR.text.lower, "; ",  
 HR.text.upper, "] \n"),  
 "\u0394RMST = ", RMST.text,  
 paste0(", 95% CI: [", RMST.text.lower,  
 "; ", RMST.text.upper, "]"), "\n"),  
 size=3.5, hjust=0) # TODO: size=4.5  
 # ggplot2::theme(axis.title.x=ggplot2::element\_blank(),  
 # axis.text.y=ggplot2::element\_text(size=12),  
 # axis.title.y=ggplot2::element\_text(size=12))  
 assign(paste0("km\_plot\_", outcome), km\_plot)  
 km\_table <- survminer::ggsurvplot(survival::survfit(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 robust=TRUE,  
 type="kaplan-meier",   
 conf.type="log"),  
 palette=c("#EC7F12", "#9161BD"),  
 ggtheme = theme(  
 axis.line = element\_line(colour = "black"),  
 panel.background = element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = element\_blank(), #remove major gridlines  
 panel.grid.minor = element\_blank(), #remove minor gridlines  
 legend.background = element\_rect(fill='transparent'), #transparent legend bg  
 legend.box.background = element\_rect(fill='transparent') #transparent legend panel  
 ),  
 legend.labs=c("-2R", "+2R"),  
 xlab="Time in years",  
 risk.table=TRUE,  
 robust=TRUE,  
 pval=FALSE)$table  
 assign(paste0("km\_table\_", outcome), km\_table)  
  
 # Write results to Excel  
 openxlsx::write.xlsx(x=data.frame(names=c("All patients", "6xRCHOP21", "6xRCHOP21+2R", "Difference", "P-value"),  
 KMest=c(round(c(all$surv, treat$surv)\*100, 2),   
 round(diff(treat$surv)\*100, 2),  
 LRT.text),  
 KMCI=c(paste0("[", round(c(all$lower, treat$lower)\*100, 2),  
 "; ", round(c(all$upper, treat$upper)\*100, 2),"]"),   
 paste0("[", round((diff(treat$surv)-1.96\*sqrt(sum((treat$std.err)^2)))\*100, 2),  
 "; ", round((diff(treat$surv)+1.96\*sqrt(sum((treat$std.err)^2)))\*100, 2),"]"),   
 ""),  
 Coxest=c("",  
 "(ref)",  
 HR.text,  
 "",  
 round(summary(cox)$robscore["pvalue"], 3)),  
 CoxCI=c("",  
 "",  
 paste0("[", HR.text.lower, "; ",  
 HR.text.upper, "]"),  
 "",  
 ""),  
 RMSTest=c("",  
 "",  
 "",  
 RMST.text,  
 ""),  
 RMSTCI=c("",  
 "",  
 "",  
 paste0("[", RMST.text.lower,  
 "; ", RMST.text.upper, "]"),  
 "")),  
 file=paste0(file.path, "Results/results.", outcome , ".", weights.name, ".xlsx"),  
 sheetName=outcome)  
 }  
  
 # PH assumption plot  
 png(file=paste0(file.path, "Results/Plots/PH.assumption.", weights.name, ".png"),  
 width=600, height=300)  
 par(mfrow=c(1, 2))  
 outcome <- "EFS"  
 plot(survival::cox.zph(survival::coxph(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 robust=TRUE)), main="a. Event-free survival")  
 outcome <- "OS"  
 plot(survival::cox.zph(survival::coxph(eval(parse(text=paste0("S.", outcome, ".limit ~ RCHOP21"))),  
 data=eval(parse(text=paste0("DLBCL.", outcome))),  
 weights=eval(parse(text=paste0("DLBCL.", outcome, "[, \"", weights.name, "\"]"))),  
 robust=TRUE)), main="b. Overall survival")  
 grDevices::dev.off()  
}  
  
# Kaplan-Meier curves all patients  
ggsave(filename=paste0(file.path, "Results/KM.EFS.OS.", weights.name, ".png"),  
 plot=ggpubr::ggarrange(km\_plot\_EFS,  
 km\_plot\_OS,  
 km\_table\_EFS,  
 km\_table\_OS,  
 nrow=2, ncol=2, align="h", labels=c("A", "B"),  
 common.legend=TRUE, legend="top", heights=c(1, 0.3)),  
 width=8, height=6, dpi=300)

# Subgroup analysis stratified by IPI score

for (weights.name in c("stabilized.w.i")){ # stabilized.w.i.quantile  
 for (outcome in ylabs$outcome){  
 # estimate treatment effect in each quantile  
 limit.year <- eval(parse(text=(paste0("limit.year.", outcome))))  
 KM\_est\_quantile <- c()  
 plot.KM <- list()  
 table.KM <- list()  
 HR\_quantile <- c()  
 HR\_quantile\_lower <- c()  
 HR\_quantile\_upper <- c()  
 RMST\_RCHOP21 <- c()  
 RMST\_quantiles <- c()  
 RMST\_quantiles\_lower <- c()  
 RMST\_quantiles\_upper <- c()  
 # RMST\_quantiles\_p <- c()  
 LRT\_stat\_quantile <- c()  
 LRT\_p\_quantile <- c()  
 for (ipi.i in 1:g){  
 # select data for only ith quantile  
 quantile.DLBCL <- eval(parse(text=paste0("DLBCL.", outcome, "[DLBCL.", outcome, "$ipi\_total==ipi.i-1,]")))  
 # quantile.DLBCL <- ordered.DLBCL[ordered.DLBCL$quantile.nr==ipi.i,]  
  
 # Survival outcome  
 S.quantile <- eval(parse(text=paste0("quantile.DLBCL$S.", outcome)))  
 # Set the censoring of patients with follow-up time more than x years to "censored"  
 S.quantile[S.quantile[,1]>limit.year, 2] <- 0  
 # Set the follow-up time to x years for patients with follow-up time more than x years.  
 S.quantile[S.quantile[,1]>limit.year, 1] <- limit.year  
 quantile.DLBCL$S.quantile <- S.quantile  
  
 # event-free survival at 5 years  
 km\_quantile <- survival::survfit(S.quantile~RCHOP21, data=quantile.DLBCL, robust=TRUE,  
 weights=quantile.DLBCL[, weights.name],  
 type="kaplan-meier", conf.type="log")  
 # save survival  
 treat\_quantile <- summary(km\_quantile, times = limit.year)  
 KM\_est\_quantile <- rbind(KM\_est\_quantile,  
 cbind(treat\_quantile$surv, treat\_quantile$std.err,  
 treat\_quantile$lower, treat\_quantile$upper))  
   
 # log-rank test  
 LRT\_quantile <- RISCA::ipw.log.rank(times=S.quantile[, 1], failures=S.quantile[, 2],  
 variable=quantile.DLBCL$RCHOP21,  
 weights=quantile.DLBCL[, weights.name])  
 LRT\_stat\_quantile <- c(LRT\_stat\_quantile, LRT\_quantile$statistic)  
 LRT\_p\_quantile <- c(LRT\_p\_quantile, LRT\_quantile$p.value)  
  
 # # Kaplan Meier plots for each group of IPI  
 # png(file=paste0(file.path, "Results/Plots/", outcome, ".", round(limit.year, 1), ".Q", i, ".", weights.name, ".emf"), width=500, height=500)  
 # show(survminer::ggsurvplot(km\_quantile, conf.int="True", risk.table=TRUE))  
 # grDevices::dev.off()  
   
 # univariate cox  
 cox\_quantile <- survival::coxph(S.quantile~RCHOP21, data=quantile.DLBCL,  
 weights=quantile.DLBCL[, weights.name],  
 robust=TRUE)  
 PH.assumption <- survival::cox.zph(cox\_quantile)$table["RCHOP21", "p"]  
  
 # save hazard ratios  
 HR\_quantile <- c(HR\_quantile, as.numeric(exp(coef(cox\_quantile)[1])))  
 HR\_quantile\_lower <- c(HR\_quantile\_lower, as.numeric(exp(confint(cox\_quantile)[1,1])))  
 HR\_quantile\_upper <- c(HR\_quantile\_upper, as.numeric(exp(confint(cox\_quantile)[1,2])))  
  
 # RMST  
 RMST\_quantile <- survival:::survmean(survival::survfit(eval(parse(text=paste0("S.", outcome, "~RCHOP21"))),  
 data=quantile.DLBCL, robust=TRUE,  
 weights=quantile.DLBCL[, weights.name],  
 type="kaplan-meier", conf.type="log"),  
 rmean=limit.year)  
   
 RMST\_RCHOP21 <- cbind(RMST\_RCHOP21, RMST\_quantile$matrix[, "rmean"])  
 dRMST\_Q <- diff(RMST\_quantile$matrix[, "rmean"])  
 RMST\_quantiles <- c(RMST\_quantiles, dRMST\_Q)  
 RMST\_quantiles\_lower <- c(RMST\_quantiles\_lower,   
 dRMST\_Q-1.96\*sqrt(RMST\_quantile$matrix[1, "se(rmean)"]^2+RMST\_quantile$matrix[2, "se(rmean)"]^2))  
 RMST\_quantiles\_upper <- c(RMST\_quantiles\_upper,   
 dRMST\_Q+1.96\*sqrt(RMST\_quantile$matrix[1, "se(rmean)"]^2+RMST\_quantile$matrix[2, "se(rmean)"]^2))  
 # dRMST\_se <- sqrt(sum(RMST\_quantile$matrix[, "se(rmean)"]^2/RMST\_quantile$matrix[, "records"]))  
 # RMST\_quantiles\_p <- c(RMST\_quantiles\_p, 2\*pnorm(-abs(dRMST\_Q/dRMST\_se)))  
   
 # make text for KM plots  
 LRT\_text\_quantile <- sprintf("%.2f", LRT\_quantile$p.value)  
 HR\_text\_quantile <- sprintf("%.2f", exp(coef(cox\_quantile)[1]))  
 HR\_text\_quantile\_lower <- sprintf("%.2f", exp(confint(cox\_quantile)[1,1]))  
 HR\_text\_quantile\_upper <- sprintf("%.2f", exp(confint(cox\_quantile)[1,2]))  
 RMST\_text\_quantile <- sprintf("%.2f", diff(RMST\_quantile$matrix[, "rmean"]))  
 RMST\_text\_quantile\_lower <- sprintf("%.2f",   
 diff(RMST\_quantile$matrix[, "rmean"])-1.96\*sqrt(RMST\_quantile$matrix[1, "se(rmean)"]^2+RMST\_quantile$matrix[2, "se(rmean)"]^2))  
 RMST\_text\_quantile\_upper <- sprintf("%.2f",   
 diff(RMST\_quantile$matrix[, "rmean"])+1.96\*sqrt(RMST\_quantile$matrix[1, "se(rmean)"]^2+RMST\_quantile$matrix[2, "se(rmean)"]^2))  
   
 # save Kaplan-Meier plots  
 plot.KM[[ipi.i]] <- survminer::ggsurvplot(km\_quantile, conf.int="True",  
 palette=c("#EC7F12", "#9161BD"),  
 ggtheme = ggplot2::theme(  
 axis.line = ggplot2::element\_line(colour = "black"),  
 panel.background = ggplot2::element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = ggplot2::element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = ggplot2::element\_blank(), #remove major gridlines  
 panel.grid.minor = ggplot2::element\_blank(), #remove minor gridlines  
 legend.background = ggplot2::element\_rect(fill='transparent'), #transparent legend bg  
 legend.box.background = ggplot2::element\_rect(km\_plot, fill='transparent') #transparent legend panel  
 ),  
 ylim=c(0, 1), risk.table=TRUE)$plot+  
 ggplot2::theme(legend.position = "none")+  
 ggplot2::ylab(ylabs[ylabs$outcome==outcome, "label"])+  
 ggplot2::xlab("")+  
 ggplot2::theme(plot.title=ggplot2::element\_text(hjust=0.5, size=10),  
 axis.text.x=ggplot2::element\_text(size=10),  
 axis.title.x=ggplot2::element\_text(size=10),  
 axis.text.y=ggplot2::element\_text(size=10),  
 axis.title.y=ggplot2::element\_text(size=10))+  
 ggplot2::ggtitle(paste0("(N=", nrow(quantile.DLBCL), ")"))   
   
 table.KM[[ipi.i]] <- survminer::ggsurvplot(survival::survfit(S.quantile~RCHOP21,   
 data=quantile.DLBCL,   
 robust=TRUE,  
 type="kaplan-meier",   
 conf.type="log"),  
 palette=c("#EC7F12", "#9161BD"),  
 ggtheme = theme(axis.line = element\_line(colour = "black"),  
 axis.text.y = ggplot2::element\_text(size=10),  
 axis.title.y = ggplot2::element\_text(size=10),  
 axis.line.y = ggplot2::element\_blank(),  
 axis.ticks.y = ggplot2::element\_blank(),  
 axis.line.x = ggplot2::element\_blank(),  
 axis.ticks.x = ggplot2::element\_blank(),  
 axis.text.x = ggplot2::element\_blank(),  
 axis.title.x = ggplot2::element\_blank(),  
 panel.background = ggplot2::element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = ggplot2::element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = ggplot2::element\_blank(), #remove major gridlines  
 panel.grid.minor = ggplot2::element\_blank(), #remove minor gridlines  
 legend.background = ggplot2::element\_rect(fill='transparent'), #transparent legend bg  
 legend.box.background = ggplot2::element\_rect(fill='transparent') #transparent legend panel  
 ),  
 legend.labs=c("-2R", "+2R"),  
 xlab="Time in years",  
 risk.table=TRUE,  
 risk.table.title=element\_blank(),  
 risk.table.fontsize=3,  
 robust=TRUE, pval=FALSE)$table  
 }  
  
 # combine Kaplan-Meier estimates into one data frame  
 KM\_est\_quantile <- cbind(KM\_est\_quantile, rep(c(0, 1), g))  
 KM\_est\_quantile <- cbind(KM\_est\_quantile, rep(1:g, each=2))  
 colnames(KM\_est\_quantile) <- c("estimate", "sd", "lower", "upper", "treatment", "risk.group")  
 KM\_est\_quantile <- as.data.frame(KM\_est\_quantile)  
 assign(paste0("KM\_est\_quantile\_", outcome), KM\_est\_quantile)  
  
 # combine hazard ratio"s into one data frame  
 HR\_df\_quantile <- data.frame(risk.group=1:g, estimate=HR\_quantile,   
 lower=HR\_quantile\_lower, upper=HR\_quantile\_upper)  
  
 # combine RMST into one data frame  
 RMST\_df\_quantile <- data.frame(risk.group=1:g, estimate=RMST\_quantiles,   
 lower=RMST\_quantiles\_lower, upper=RMST\_quantiles\_upper)  
 assign(paste0("RMST.df.quantile.", outcome), RMST\_df\_quantile)  
 linear.RMST <- summary(lm(estimate ~ risk.group, data=RMST\_df\_quantile))$coefficients  
 assign(paste0("RMST\_RCHOP21\_", outcome), RMST\_RCHOP21)  
   
 # Calculate the absolute difference in event-free survival, i.e. benefit and harm.  
 diff <- c()  
 j <- 1  
 for (i in seq(from=1, to=nrow(KM\_est\_quantile), by=2)){  
 diff <- rbind(diff, KM\_est\_quantile[i+1,]-KM\_est\_quantile[i,])  
  
 # standard deviation of difference is square root of sum of variances  
 diff[j, "sd"] <- sqrt(KM\_est\_quantile[i+1, "sd"]^2+KM\_est\_quantile[i, "sd"]^2)  
 diff[j, "lower"] <- diff[j, "estimate"]-1.96\*diff[j, "sd"]  
 diff[j, "upper"] <- diff[j, "estimate"]+1.96\*diff[j, "sd"]  
 j <- j + 1  
 }  
 diff <- as.data.frame(diff)  
 diff$risk.group <- 1:g  
 linear.KM <- summary(lm(estimate ~ risk.group, data=diff))$coefficients  
 ARD\_df\_quantile <- data.frame(risk.group=1:g, estimate=diff$estimate\*100,   
 lower=diff$lower\*100, upper=diff$upper\*100)  
  
 # Write results to Excel  
 results.df.q <- data.frame(cbind(c(rep("ARD", 4), rep("HR", 4), rep("RMST", 4)),  
 rbind(ARD\_df\_quantile,  
 HR\_df\_quantile,  
 RMST\_df\_quantile)))  
 results.df.q$text <- paste0(sprintf("%.2f", results.df.q$estimate), " [",   
 sprintf("%.2f", results.df.q$lower), "; ",   
 sprintf("%.2f", results.df.q$upper), "]")  
 openxlsx::write.xlsx(results.df.q,  
 file=paste0(file.path, "Results/ARD.HR.RMST.", outcome , ".", weights.name, ".xlsx"),  
 sheetName=outcome)  
  
 # Kaplan-Meier plots  
 KM.plots <- ggpubr::ggarrange(plot.KM[[1]]+ggplot2::labs(tag="A"),  
 plot.KM[[2]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 plot.KM[[3]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 plot.KM[[4]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 ncol=4, nrow=1,  
 widths=c(1, 0.8, 0.8, 0.8),  
 align="h")  
   
 # Kaplan-Meier tables  
 table.KM.plots <- ggpubr::ggarrange(table.KM[[1]],  
 table.KM[[2]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 table.KM[[3]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 table.KM[[4]]+ggpubr::rremove("y.axis")+  
 ggpubr::rremove("ylab")+  
 ggpubr::rremove("y.text")+  
 ggpubr::rremove("y.ticks"),  
 ncol=4, nrow=1,  
 widths=c(1, 0.8, 0.8, 0.8),  
 align="h")  
   
 # plot for absolute reduction in survival  
 avg.diff <- diff(summary(eval(parse(text=paste0("km\_", outcome))),  
 times = limit.year)$surv)  
 diff.plot <- ggplot2::ggplot(data=diff,  
 ggplot2::aes(x=factor(risk.group), y=estimate))+  
 ggplot2::geom\_hline(yintercept=0)+  
 ggplot2::geom\_hline(yintercept=avg.diff, linetype="dashed", linewidth=1.2)+  
 ggplot2::geom\_errorbar(ggplot2::aes(ymin=lower, ymax=upper),  
 position=ggplot2::position\_dodge(0.9), width=0)+  
 ggplot2::geom\_point(size=5, shape=18, color="#EC7F12")+  
 ggplot2::scale\_y\_continuous(limits=c(min(diff$lower, 0),  
 max(diff$upper)),  
 labels=scales::percent\_format(accuracy=1))+  
 ggplot2::theme(text=ggplot2::element\_text(size=12),  
 axis.line.y = element\_line(colour = "black"),  
 panel.background = element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = ggplot2::element\_blank(),  
 panel.grid.minor = ggplot2::element\_blank())+  
 ggplot2::ylab("ARD") +   
 ggplot2::theme(axis.title.x=ggplot2::element\_blank(),  
 axis.text.x=ggplot2::element\_blank(),  
 axis.ticks.x=ggplot2::element\_blank())+  
 ggplot2::labs(tag="C")  
   
 # create plot for hazard ratios  
 cox <- eval(parse(text=paste0("cox.", outcome)))  
 linear.HR <- summary(lm(estimate ~ risk.group, data=HR\_df\_quantile))$coefficients  
 hazard.plot <- ggplot2::ggplot(data=HR\_df\_quantile, ggplot2::aes(x=risk.group, y=estimate))+  
 ggplot2::geom\_hline(yintercept=1)+  
 ggplot2::geom\_hline(yintercept=exp(coef(cox)), linetype="dashed", linewidth=1.2)+  
 ggplot2::geom\_errorbar(ggplot2::aes(ymin=lower, ymax=upper),  
 position=ggplot2::position\_dodge(0.9), width=0)+  
 ggplot2::geom\_point(size=5, shape=18, color="#EC7F12")+  
 ggplot2::scale\_x\_discrete(limits=as.factor(1:g))+  
 ggplot2::scale\_y\_continuous(limits=c(min(HR\_df\_quantile$lower),   
 max(HR\_df\_quantile$upper, 1)),  
 labels=scales::number\_format(accuracy=0.01))+  
 ggplot2::theme(text=ggplot2::element\_text(size=12),  
 axis.title.x=ggplot2::element\_blank(),  
 axis.text.x=ggplot2::element\_blank(),  
 axis.ticks.x=ggplot2::element\_blank(),  
 axis.line.y = element\_line(colour = "black"),  
 panel.background = element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = ggplot2::element\_blank(),  
 panel.grid.minor = ggplot2::element\_blank())+  
 ggplot2::ylab("Hazard ratio")+  
 ggplot2::labs(tag="B")  
  
 # create plot for RMST  
 dRMST <- eval(parse(text=paste0("dRMST.", outcome)))  
 RMST.plot <- ggplot2::ggplot(data=RMST\_df\_quantile, ggplot2::aes(x=risk.group, y=estimate))+  
 ggplot2::geom\_hline(yintercept=0)+  
 ggplot2::geom\_hline(yintercept=dRMST, linetype="dashed", linewidth=1.2)+  
 ggplot2::geom\_errorbar(ggplot2::aes(ymin=lower, ymax=upper),  
 position=ggplot2::position\_dodge(0.9), width=0)+  
 ggplot2::geom\_point(size=5, shape=18, color="#EC7F12")+  
 ggplot2::scale\_x\_discrete(limits=c("1"="IPI score 0-1", "2"="IPI score 2", "3"="IPI score 3", "4"="IPI score 4-5"))+  
 ggplot2::scale\_y\_continuous(limits=c(min(RMST\_df\_quantile$lower, 0),   
 max(RMST\_df\_quantile$upper)),  
 labels=scales::number\_format(accuracy=0.01))+  
 ggplot2::theme(text=ggplot2::element\_text(size=12),  
 axis.line.y = element\_line(colour = "black"),  
 panel.background = element\_rect(fill='transparent'), #transparent panel bg  
 plot.background = element\_rect(fill='transparent', color=NA), #transparent plot bg  
 panel.grid.major = ggplot2::element\_blank(),  
 panel.grid.minor = ggplot2::element\_blank())+  
 ggplot2::xlab("IPI score")+  
 ggplot2::ylab("\u0394RMST")+  
 ggplot2::labs(tag="D")  
   
 # save plot  
 ggsave(filename=paste0(file.path, "Results/", outcome, ".grouped.TE.", weights.name, ".png"),  
 plot=ggpubr::ggarrange(KM.plots, table.KM.plots,  
 hazard.plot / diff.plot / RMST.plot,   
 nrow=3, ncol=1, heights=c(5, 2, 10)),  
 width=8, height=8, dpi=300)  
 }  
}

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Î”RMST' in 'mbcsToSbcs': dot substituted for <94>