ANALYSIS OF THE TCGA-PANCANCER DATASET

#R version 3.6.1 (2019-07-05) – "Action of the Toes" #Copyright (C) 2019 The R Foundation for Statistical Computing #Platform: x86_64-w64-mingw32/x64 (64-bit)

PART 0: PREPARE THE ENVIRONMENT ————————————————

#1. Load (or install) all necessary packages.

```
library("readxl")
library("writexl")
library("readr")
library("ComplexHeatmap")
library("circlize")
library("ggplot2")
library("gridExtra")
library("gplots")
library("mclust")
library("GSEABase")
library("methods")
library("edgeR")
library("geneplotter")
library("genefilter")
library("BiocGenerics")
library("Biobase")
library("graph")
library("XML")
library("lattice")
library("limma")
library("shinythemes")
library("shiny")
library("RColorBrewer")
library("parallel")
library("cluster")
library("Matrix")
library("locfit")
library("snow")
library("GSVA")
library("dplyr")
library("ggplot2")
library("data.table")
library ("remotes")
library("plyr")
library("magrittr")
library("OIsurv")
```

```
library("survival")
library("KMsurv")
library("splines")
library("survminer")
library("ggpubr")
library("survutils")
library("scales")
library("ggpubr")
library("tidyverse")
library("corrr")
library("igraph")
library("ggraph")
library("tidygraph")
library("CoxBoost")
library("glmnet")
library("randomForest")
library("class")
library("dml")
library("MASS")
library("readr")
library("Rtsne")
library("stats")
library("ggridges")
library("gdata")
library("ggrepel")
library("corrplot")
library("ggExtra")
library("gridExtra")
library("rstatix")
library("ggpubr")
library("viridis")
library("corrplot")
library("xlsx")
library("plotly")
library("plot3D")
library("tidyr")
```

PART 1: DATA IMPORTATION

#1. Import the TGFB and ALTEJ genelists.

```
TGFBgeneset <- read.table("Input/TGFB_list.txt", quote="\"", comment.char="", stringsAsFactors=FALSE)
ALTEJgeneset <- read.table("Input/ALTEJ_list.txt", quote="\"", comment.char="", stringsAsFactors=FALSE)
TGFBlist <- as.list(TGFBgeneset[,1:1]) #55 genes (50 + synonyms)
ALTEJlist <- as.list(ALTEJgeneset[,1:1]) #45 genes (36 + synonyms)
Bothgenelists <- list(TGFBUPgeneset=TGFBlist, ALTEJgeneset=ALTEJlist)
```

#2. Import the file with normalized gene expression.

```
Exp = fread("Input/ZscoresofLogTMMvaluesofallgenes.txt", data.table=FALSE)
Exp[1:10,1:100]
```

```
class(Exp) #[1] "data.frame"
rownames(Exp) <- Exp$V1
Exp$V1 <- NULL
dim(Exp) #9.572 samples x 16.335 genes. There are 4 genes missing from the TGFB signature: LAMC2, FAP,
class(Exp$A1BG) #numeric
##This file contatins normalized gene expression of the TCGA-pancancer primary solid tumor samples.
##Normalization (as described in Methods): TMM normalization -> Log2 transformation -> gene-centered Z-
##The original unnormalized file was downloaded by Mao from the GDC portal (https://gdc.cancer.gov/abou
```

#3. Import the file with the clinical information.

```
Clin <- read.delim("Input/Clinicalinfoprimarytumorsunduplicated.txt") #11.248 patients.
##This file contatins clinical information from TCGA-pancancer patients with primary solid tumors.
##It was sent by a collaborator (Mao), who generated it from the file "TCGA-CDR-SupplementalTableS1.xls
```

#4. Import the files with genes' weights.

```
TGFBimportance <- read_excel("Input/Relative importance of BAlt genes.xlsx", sheet = "TGFB")
ALTEJimportance <- read_excel("Input/Relative importance of BAlt genes.xlsx", sheet = "ALTEJ")
##These files are the ones generated based on the results from the "inhouse HNSC dataset", and can also
```

PART 2.A: SCORES CALCULATION - ORIGINAL BALT

#1. Turn the dataframe with gene expression into a matrix.

```
Exp2 <- as.matrix(Exp)
class(Exp2) #matrix
Exp2[1:10, 1:50] #Genes are columns and samples are rows.
```

#2. Calculate the ssGSEA scores of the TGFB and ALTEJ signatures in each sample.

```
Exp2<-t(Exp2) #Put genes as rows.
Exp2[1:20, 1:10]
ssgsea <- gsva(Exp2, Bothgenelists, method=c("ssgsea"))
```

#3. Traspose the matrix with the ssGSEA results and turn it into a dataframe.

```
ssgsea <- t(ssgsea)
ssgsea <- as.data.frame(ssgsea)
names(ssgsea)
ssgsea$TGFBUPgeneset_original_ssgsea <- ssgsea$TGFBUPgeneset
ssgsea$ALTEJgeneset_original_ssgsea <- ssgsea$ALTEJgeneset
cor.test(ssgsea$TGFBUPgeneset_original_ssgsea,
         ssgsea$ALTEJgeneset_original_ssgsea,
         method = "pearson") #Test if TGFB and ALTEJ are anticorrelated: PCC= -0.126.
```

#4. Create a new variable that is the original Balt score.

```
ssgsea$balt <- sqrt((max(ssgsea$ALTEJgeneset_original_ssgsea)-ssgsea$ALTEJgeneset_original_ssgsea)^2+
                      (min(ssgsea$TGFBUPgeneset_original_ssgsea)-ssgsea$TGFBUPgeneset_original_ssgsea)^2)-
                sqrt((min(ssgsea$ALTEJgeneset_original_ssgsea)-ssgsea$ALTEJgeneset_original_ssgsea)^2+
                      (max(ssgsea$TGFBUPgeneset_original_ssgsea)-ssgsea$TGFBUPgeneset_original_ssgsea)^2)
ssgsea$balt <- ssgsea$balt * -1
ssgsea$balt_original_ssgsea <- ssgsea$balt
```

PART 2.B: SCORES CALCULATION - WEIGHTED BALT

#1. Create a matrix with only the expression of the TGFB and ALTEJ genes.

```
Exp2[1:20, 1:10]
genes <- Exp2[which(rownames(Exp2) %in% TGFBlist | rownames(Exp2) %in% ALTEJlist), ]
dim(genes) #9,572 samples x 82 genes because 4 are missing.
```

#2. Edit gene names so that they match the ones from the genes' weight dataframe.

```
genes <- as.data.frame(genes)
genes$Gene <- rownames(genes)
genes$Gene <- ifelse(genes$Gene=="APEX2", "APE2",
                     ifelse(genes$Gene=="HRAS", "HRAS1",
                     ifelse(genes$Gene=="PRPF19", "PRP19",
                     ifelse(genes$Gene=="RAD51L3", "RAD51D",
                     ifelse(genes$Gene=="KAT5", "TIP60",
                     ifelse(genes$Gene=="ARHGAP32", "RICS",
                     ifelse(genes$Gene=="C19orf40", "FAAP24",
                     ifelse(genes$Gene=="CYTH1", "PSCD1",
                     ifelse(genes$Gene=="NUDT1", "MTH1",
                     ifelse(genes$Gene=="OBFC2B", "NABP2",
                     ifelse(genes$Gene=="STAG1", "TMEPAI", genes$Gene)))))))))))
rownames(genes) <- genes$Gene
genes[1:10, 1:10]
```

#3. Merge the genes and their weights in one dataframe.

```
genesimportance <- rbind(TGFBimportance[,c("Gene", "mean weight")], ALTEJimportance[,c("Gene", "mean wei
genes2 <- merge(genesimportance, genes,
                by.x="Gene", by.y="Gene",
                all.x=TRUE, all.y=TRUE) #86 genes, but 4 missing information.
genes2[1:86,1:10]
```

#4. Calculate the TGFB and ALTEJ weighted scores in each sample, by multiplying each gene by its factor.

```
##Factor = 1+weight if weight>0; or 1 if weight<0.
genes2$factor <- ifelse(genes2$`mean weight`< 0, 1, genes2$`mean weight`)
genes2$factor <- ifelse(genes2$factor==1, 1, 1+genes2$factor)
rownames(genes2) <- genes2$Gene
scores <- sapply(genes2[,-which(colnames(genes2) %in% c("Gene", "mean weight", "factor"))], '*', (genes2
rownames(scores) <- rownames(genes2)
```

```
scores <- as.data.frame(t(scores))
scores$TGFBUPgeneset <- rowSums(scores[ , which(colnames(scores) %in% TGFBlist)], na.rm=TRUE)
scores$ALTEJgeneset <- rowSums(scores[ , which(colnames(scores) %in% ALTEJlist)], na.rm=TRUE)
scores <- scores[,c("TGFBUPgeneset", "ALTEJgeneset")]
cor.test(scores$TGFBUPgeneset,
         scores$ALTEJgeneset,
         method = "pearson") #Test if TGFB and ALTEJ are anticorrelated: PCC= -0.04.
```

#5. Create a new variable that is the weighted Balt score.

```
scores$balt <- sqrt((max(scores$ALTEJgeneset)-scores$ALTEJgeneset)^2+
                      (min(scores$TGFBUPgeneset)-scores$TGFBUPgeneset)^2) -
               sqrt((min(scores$ALTEJgeneset)-scores$ALTEJgeneset)^2+
                      (max(scores$TGFBUPgeneset)-scores$TGFBUPgeneset)^2)
scores$balt <- scores$balt * -1
scores$balt_weighted <- scores$balt
scores$TGFBUPgeneset_weighted <- scores$TGFBUPgeneset
scores$ALTEJgeneset_weighted <- scores$ALTEJgeneset
scores_weighted <- scores
```

#6. Put all the scores (original and weighted) in the same dataframe.

```
scores_weighted$SampleID <- rownames(scores_weighted)
ssgsea$SampleID <- rownames(ssgsea)
scores <- merge(scores_weighted[,c("SampleID", "TGFBUPgeneset_weighted", "ALTEJgeneset_weighted", "balt
                ssgsea[,c("SampleID", "TGFBUPgeneset_original_ssgsea", "ALTEJgeneset_original_ssgsea",
                by.x="SampleID", by.y="SampleID",
                all.x=TRUE, all.y=TRUE) #9.572p +9.572p-->9.572patients. Of those, all are in common.
```

#7. Export the dataset with the original and weighted scores.

```
write.xlsx(scores, file="Output/balt_scores_TCGApancancer.xlsx", col.names = TRUE, row.names = TRUE)
```

PART 3: DATA WRANGLING

#1. Remove recurrent and normal tissue samples.

```
scores$Sampletype <- scores$SampleID
scores$Sampletype <- substr(scores$Sampletype, 13, 16) #Keep only sample type numbers.
table(scores$Sampletype) #All 9.572 samples are from primary tumors.
```

```
##
## -01A
## 9572
```

#2. Add the clinical information to the scores dataframe.

```
scores$bcr_patient_barcode <- substring(scores$SampleID,1,12)
ALL <- merge(scores, Clin,
             by.x = "bcr_patient_barcode", by.y = "bcr_patient_barcode",
             all.x=TRUE, all.y=FALSE) #9.572 + 11.248 --> 9.572 patients, of which 9.568 have clinical
```

#3. Remove samples that are from hematologic tumors.

```
table(ALL$type)
ALL <- ALL[-which(ALL$type == "DLBC" | ALL$type == "LAML"),] #9.572->9.525 patients.
```

#4. Check if some patients have more than one sample.

```
Duplicated <- ALL[duplicated(ALL$bcr_patient_barcode), ] #There are only 2 patients that have 2 samples
```

#5. Remove the PAAD samples that are mislabelled according to the article (doi: 10.1016/j.ccell.2017.07.007).

```
List_of_TCGA_PAAD_150_correct_tumors <- read_excel("Input/List_of_TCGA-PAAD_150_correct_tumors.xlsx")
PAADlist <- as.list(List_of_TCGA_PAAD_150_correct_tumors$`Table S1, related to Figure 1`)
ALL$sample <- substring(ALL$SampleID,1,16) #keep only characters 1-16
ALL <- ALL[-which(ALL$type=="PAAD" & !ALL$sample %in% PAADlist),] #9.525-> 9.497 patients.
table(ALL$type)
```

#6. Remove the GBM neural samples, since it is now thought that these samples consist mostly of normal brain tissue.

```
Subtype <- read.delim("Input/TCGA.GBM.sampleMap_GBM_clinicalMatrix.txt") #This file was downloaded from
Neural <- Subtype[which(Subtype$GeneExp_Subtype=="Neural"),]
Neural$sampleID <- as.character(Neural$sampleID)
Neurallist <- as.list(Neural$sampleID)
ALL$sample <- substring(ALL$SampleID,1,15) #keep only characters 1-15
ALL <- ALL[-which(ALL$type=="GBM" & ALL$sample %in% Neurallist),] #9.497-> 9.472 patients.
table(ALL$type)
```

#7. Prepare the survival variables so that they are in the appropriate format.

```
table(ALL$OS)
class(ALL$OS.time) #[1] "numeric"
ALL$OS.months <- ALL$OS.time/30.5
table(ALL$PFI)
class(ALL$PFI.time) #[1] "numeric"
ALL$PFI.months <- ALL$PFI.time/30.5
```

#8. Group the tumor stages into fewer groups.

```
table(ALL$ajcc_pathologic_tumor_stage)
ALL$stage <- ifelse(ALL$ajcc_pathologic_tumor_stage=="IS", "I-II",
                    ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage I", "I-II",
                    ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IA", "I-II",
                    ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IB", "I-II",
                    ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage II", "I-II",
```

```
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIA", "I-II",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIB", "I-II",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIC", "I-II",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage III", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIIA", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIIB", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IIIC", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IV", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IVA", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IVB", "III-IV",
                  ifelse(ALL$ajcc_pathologic_tumor_stage=="Stage IVC", "III-IV", NA)))))))))))))))
table(ALL$stage)
```

#9. Prepare other variables.

```
ALL$age <- ALL$age_at_initial_pathologic_diagnosis
```

PART 4: SELECT PATIENTS WHO RECEIVED GENOTOXIC TREATMENT

#1. Explore what information is available.

```
names(ALL)
table(ALL$type)
table(ALL$ajcc_pathologic_tumor_stage)
table(ALL$clinical_stage)
table(ALL$Radiation.Therapy)
```

#2. Create a dataframe of patients treated with RT.

```
ALL_RT <- ALL[which(ALL$Radiation.Therapy == "Yes"),] #9.472-->2.416 patients.
```

#3. Create a dataframe of patients who, based on their cancer type and stage, their standard of care treatment includes RT and/or genotoxic ChT.

```
ALL_RTChT <- ALL[which(ALL$Radiation.Therapy == "Yes" |
                         (ALL$type == "BLCA" & ALL$ajcc_pathologic_tumor_stage == "Stage IV") |
                         (ALL$type == "CHOL" & ALL$ajcc_pathologic_tumor_stage == "Stage III" & ALL$ajc
                         (ALL$type == "COAD" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         (ALL$type == "ESCA" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         ALL$type == "GBM" |
                         (ALL$type == "HNSC" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         (ALL$type == "LUAD" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         (ALL$type == "LUSC" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         ALL$type == "MESO" |
                         (ALL$type == "OV" & !ALL$clinical_stage == "Stage IC") |
                         ALL$type == "PAAD" |
                         (ALL$type == "READ" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         (ALL$type == "STAD" & !ALL$ajcc_pathologic_tumor_stage == "Stage I" & !ALL$ajc
                         (ALL$type == "TGCT" & !ALL$ajcc_pathologic_tumor_stage == "IS" & !ALL$ajcc_pat
),] #9.472-->4.597 patients.
```

```
##Reasoning behind the inclusion criteria:
  ##BLCA: Stage 4 usually includes genotoxic ChT with platin agents and earlier stages are frequently t
  ##CHOL: Early stages can be treated with surgery alone (+/- RT +/- ChT). Locally advanced or metastat
  ##COAD: Stage I may be treated with surgery alone. In other stages ChT usually includes genotoxic Pla
  ##ESCA: Stages T1-2N0M0 (stages <IIB) may be treated with surgery alone. Otherwise RT, ChT or both ar
  ##GBM: Standard treatment is usually surgery + RT + ChT. ChT usually consists of genotoxic Temozolami
  ##HNSC: Stage I may be treated with surgery alone. Otherwise treatment usually includes RT, ChT or bo
  ##LUAD and LUSC: Stage I may be treated with surgery alone. Otherwise treatment usually includes RT,
  ##MESO: Treatment usually includes genotoxic platinum ChT +/- RT +/- surgery.
  ##OV: Stage I may be treated with surgery alone and stage>I usually receive ChT. ChT usually includes
  ##PAAD: Rarely treated with surgery alone. RT and/or ChT are frequently added. ChT most usually inclu
  ##READ: Stage I may be treated with surgery alone. Otherwise RT is given unless the tumor is metastat
  ##STAD: Stage IA may be treated with surgery alone. Otherwise RT, ChT or both are usually used. ChT u
  ##TGCT: Stage 1 can be treated with surgery only (+ RT in seminomas). In stage >1, treatment usually
  ##BRCA: Usually treated with surgery + RT +/- HT +/- Trastuzumab +/- ChT. ChT frequently consists of
  ##CESC: Mainly treated with surgery + RT depending on the stage + ChT dependinng on the stage. ChT ca
  ##KIRC: RT is rarely indicated. Many times treated with surgery alone and the systemic treatment, whe
  ##LIHC: Many treated with surgery alone. Systemic treatment rarely includes genotoxic drugs, as the m
  ##PRAD: Usually treated with HT combined with surgery, RT or both. ChT is given only in some stage IV
  ##SKCM: Many treated with surgery alone. Systemic treatment rarely includes genotoxic drugs.
  ##THCA: Most of them are treated with surgery alone. Few exceptions are treated with both RT and ChT,
  ##UCEC: Stage I may be treated with surgery alone. Otherwise RT is usually given. ChT is used mainly
```

PART 5.A: SURVIVAL CURVES - ORIGINAL BALT

#1. Select the dataset with patients treated with RT and/or genotoxic ChT.

```
X <- ALL_RTChT
```

#2. Create a new variable that is the original balt tertiles.

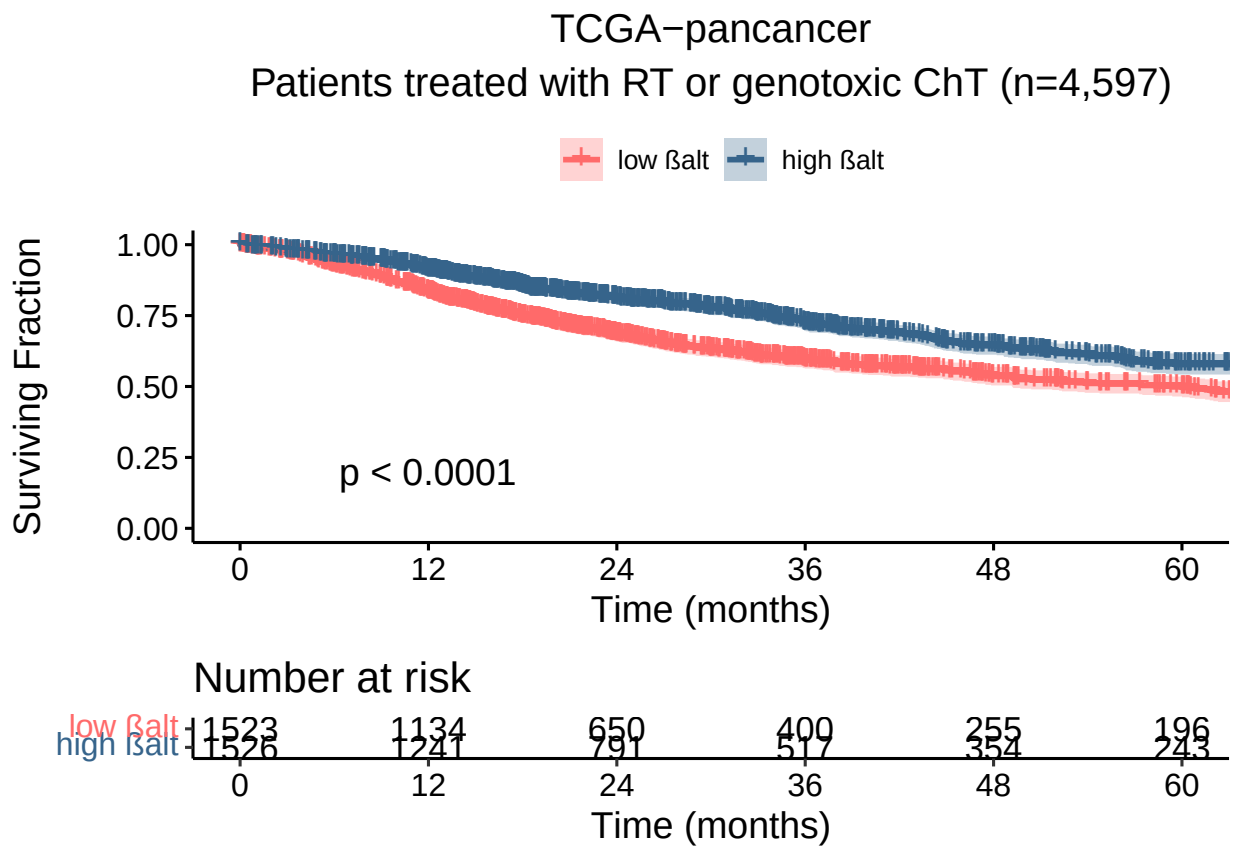
```
X$tertile <- cut(X$balt_original_ssgsea, quantile(X$balt_original_ssgsea, c(0, 1/3, 2/3, 1), na.rm=TRUE)
                 na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile <- ifelse(X$tertile == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile == "High", "low TGF
X$Tertile <- as.character(X$Tertile)
```

#3. Plot and compare the overall survival curves between the original Balt score top and bottom tertiles.

```
##Without the intermediate tertile.
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
my.fit<-survfit(my.surv.object~X$Tertile)
my.fit
```

```
## Call: survfit(formula = my.surv.object ~ X$Tertile)
##
##    1548 observations deleted due to missingness
##                                      n events median 0.95LCL 0.95UCL
## X$Tertile=high TGFß and low ALTEJ 1523    585   59.4    49.2    67.2
## X$Tertile=low TGFß and high ALTEJ 1526    442   88.3    79.0   103.2
```

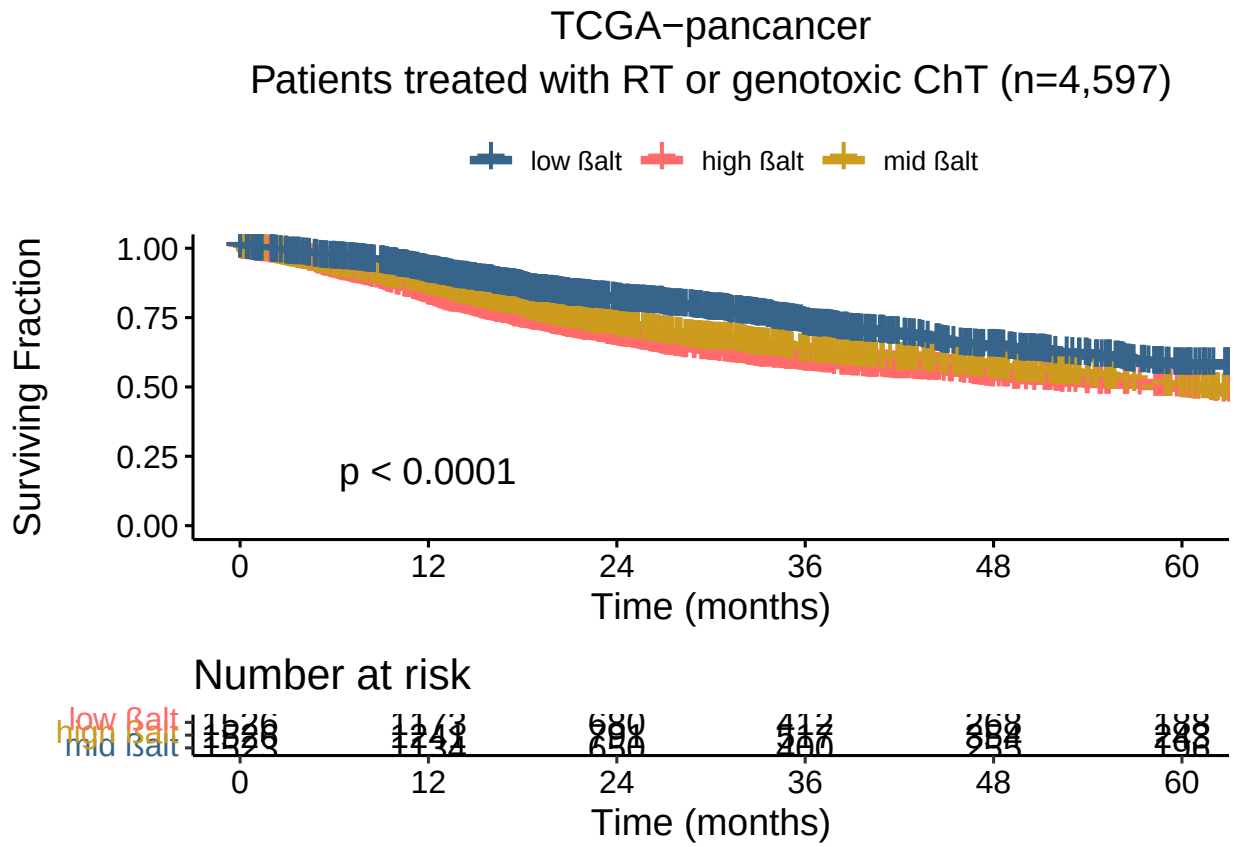
```
p<- ggsurvplot(my.fit, data=X, pval = TRUE, conf.int = TRUE, break.time.by = 12,
               xlab="Time (months)", ylab="Surviving Fraction", xlim=c(0, 60),
               legend.labs=c("low  alt", "high  alt"), legend.title=" ",
               font.main=15, palette=c("indianred1", "steelblue4"),
               #surv.median.line = "hv",
               risk.table = TRUE)
p$plot <- p$plot + labs(title="TCGA-pancancer", subtitle = "Patients treated with RT or genotoxic ChT (
  theme(plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5))
p1<-p;p1 #PDF 8x7
```




```
##With the intermediate tertile.
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
my.fit<-survfit(my.surv.object~X$tertile)
my.fit
```

```
## Call: survfit(formula = my.surv.object ~ X$tertile)
##
##    22 observations deleted due to missingness
##                      n events median 0.95LCL 0.95UCL
## X$tertile=Low     1523    585   59.4    49.2    67.2
## X$tertile=Middle 1526    543   59.4    54.8    66.5
## X$tertile=High    1526    442   88.3    79.0   103.2
```

```
p<-ggsurvplot(my.fit, data=X, pval = TRUE, size = 1.5, censor.size=7, conf.int = FALSE, break.time.by =
              xlab="Time (months)", ylab="Surviving Fraction", , xlim=c(0, 60),
              legend.labs=c("low  alt", "high  alt", "mid  alt"), legend.title=" ",
              font.main=15, palette=c("steelblue4", "indianred1", "goldenrod3"),
              risk.table = TRUE)
p$plot <- p$plot + labs(title="TCGA-pancancer", subtitle = "Patients treated with RT or genotoxic ChT (
  theme(plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5))
p1<-p;p1 #PDF 8x7
```



#4. Calculate the overall survival hazard ratio between the original Balt score top and bottom tertiles.

```
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile)
##
##   n= 3049, number of events= 1027
##    (1548 observations deleted due to missingness)
##
##                                    coef exp(coef) se(coef)      z Pr(>|z|)
## X$Tertilelow TGFß and high ALTEJ -0.43970   0.64423  0.06316 -6.962 3.36e-12
##
## X$Tertilelow TGFß and high ALTEJ ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                   exp(coef) exp(-coef) lower .95 upper .95
## X$Tertilelow TGFß and high ALTEJ    0.6442      1.552    0.5692    0.7291
##
## Concordance= 0.567  (se = 0.008 )
## Likelihood ratio test= 49.08  on 1 df,   p=2e-12
## Wald test            = 48.47  on 1 df,   p=3e-12
## Score (logrank) test = 49.24  on 1 df,   p=2e-12
```

PART 5.B: SURVIVAL CURVES - WEIGHTED BALT

#1. Select the dataset with patients treated with RT and/or genotoxic ChT.

```
X <- ALL_RTChT
```

#2. Create a new variable that is the weighted balt tertiles.

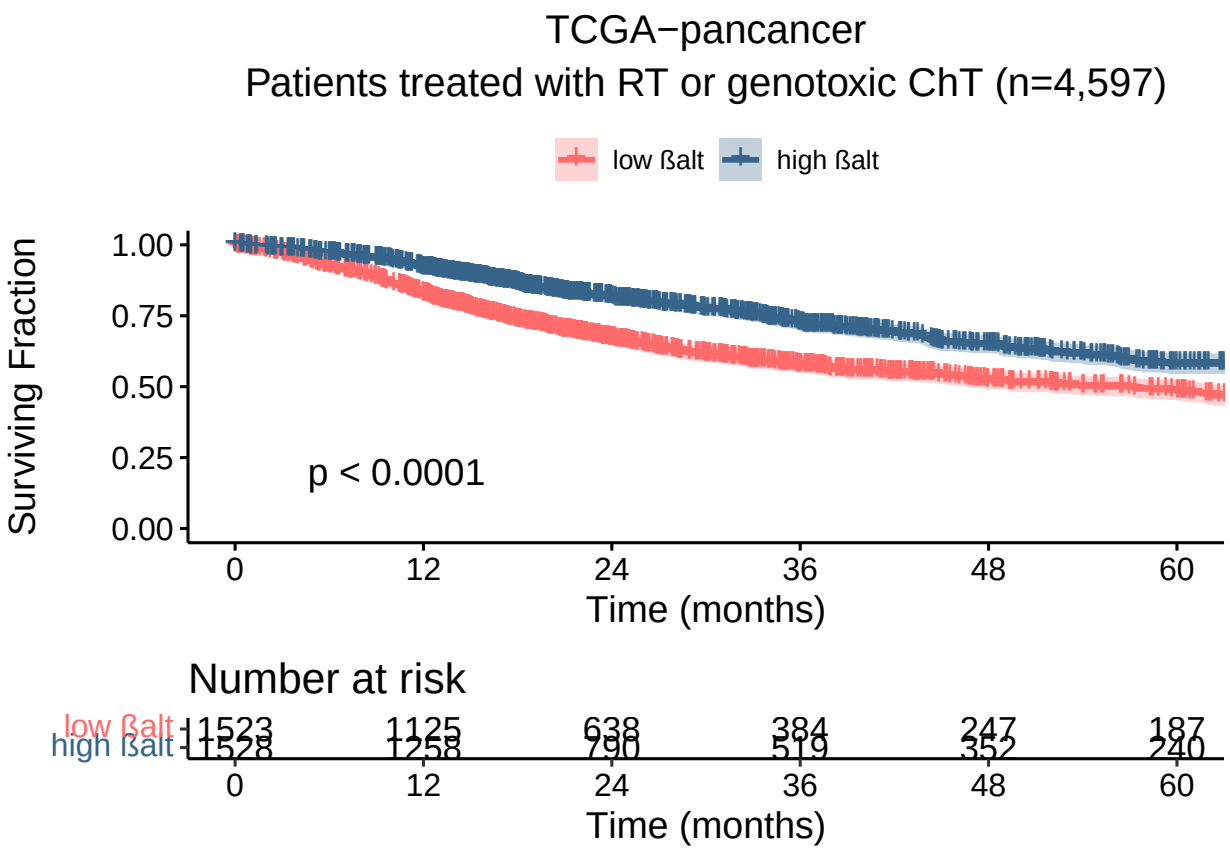
```
X$tertile <- cut(X$balt_weighted, quantile(X$balt_weighted, c(0, 1/3, 2/3, 1), na.rm=TRUE),
                 na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile <- ifelse(X$tertile == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile == "High", "low TGF
X$Tertile <- as.character(X$Tertile)
```

#3. Plot and compare the overall survival curves between the weighted Balt score top and bottom tertiles.

```
##Without the intermediate tertile.
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
my.fit<-survfit(my.surv.object~X$Tertile)
my.fit
```

```
## Call: survfit(formula = my.surv.object ~ X$Tertile)
##
##    1546 observations deleted due to missingness
##                                      n events median 0.95LCL 0.95UCL
## X$Tertile=high TGFß and low ALTEJ 1523    603   54.8    46.6    65.9
## X$Tertile=low TGFß and high ALTEJ 1528    442   88.3    79.8   103.2
```

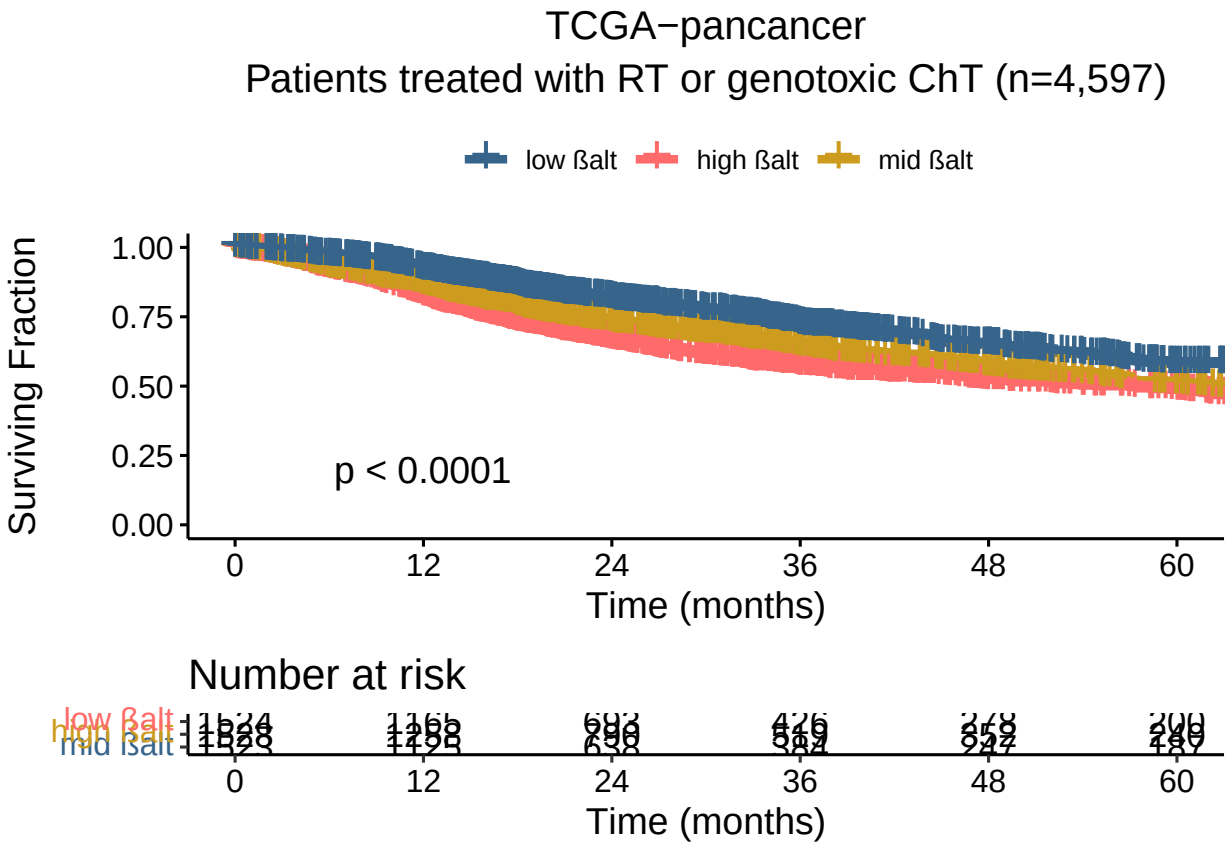
```
p<- ggsurvplot(my.fit, data=X, pval = TRUE, conf.int = TRUE, break.time.by = 12,
               xlab="Time (months)", ylab="Surviving Fraction", xlim=c(0, 60),
               legend.labs=c("low  alt", "high  alt"), legend.title=" ",
               font.main=15, palette=c("indianred1", "steelblue4"),
               #surv.median.line = "hv",
               risk.table = TRUE)
p$plot <- p$plot + labs(title="TCGA-pancancer", subtitle = "Patients treated with RT or genotoxic ChT (
  theme(plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5))
p1<-p;p1 #PDF 8x7
```




```
##With the intermediate tertile.
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
my.fit<-survfit(my.surv.object~X$tertile)
my.fit
```

```
## Call: survfit(formula = my.surv.object ~ X$tertile)
##
##    22 observations deleted due to missingness
##                      n events median 0.95LCL 0.95UCL
## X$tertile=Low     1523    603   54.8    46.6    65.9
## X$tertile=Middle 1524    525   61.7    55.3    71.5
## X$tertile=High   1528    442   88.3    79.8   103.2
```

```
p<-ggsurvplot(my.fit, data=X, pval = TRUE, size = 1.5, censor.size=7, conf.int = FALSE, break.time.by =
              xlab="Time (months)", ylab="Surviving Fraction", , xlim=c(0, 60),
              legend.labs=c("low  alt", "high  alt", "mid  alt"), legend.title=" ",
              font.main=15, palette=c("steelblue4", "indianred1", "goldenrod3"),
              risk.table = TRUE)
p$plot <- p$plot + labs(title="TCGA-pancancer", subtitle = "Patients treated with RT or genotoxic ChT (
  theme(plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5))
p1<-p;p1 #PDF 8x7
```



#4. Calculate the overall survival hazard ratio between the weighted Balt score top and bottom tertiles.

```
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile)
##
##   n= 3051, number of events= 1045
##    (1546 observations deleted due to missingness)
##
##                                       coef exp(coef) se(coef)      z Pr(>|z|)
## X$Tertilelow TGFß and high ALTEJ -0.50120   0.60580  0.06284 -7.976 1.51e-15
##
## X$Tertilelow TGFß and high ALTEJ ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                   exp(coef) exp(-coef) lower .95 upper .95
## X$Tertilelow TGFß and high ALTEJ    0.6058      1.651    0.5356    0.6852
##
## Concordance= 0.579  (se = 0.008 )
## Likelihood ratio test= 64.63  on 1 df,   p=9e-16
## Wald test            = 63.62  on 1 df,   p=2e-15
## Score (logrank) test = 64.94  on 1 df,   p=8e-16
```

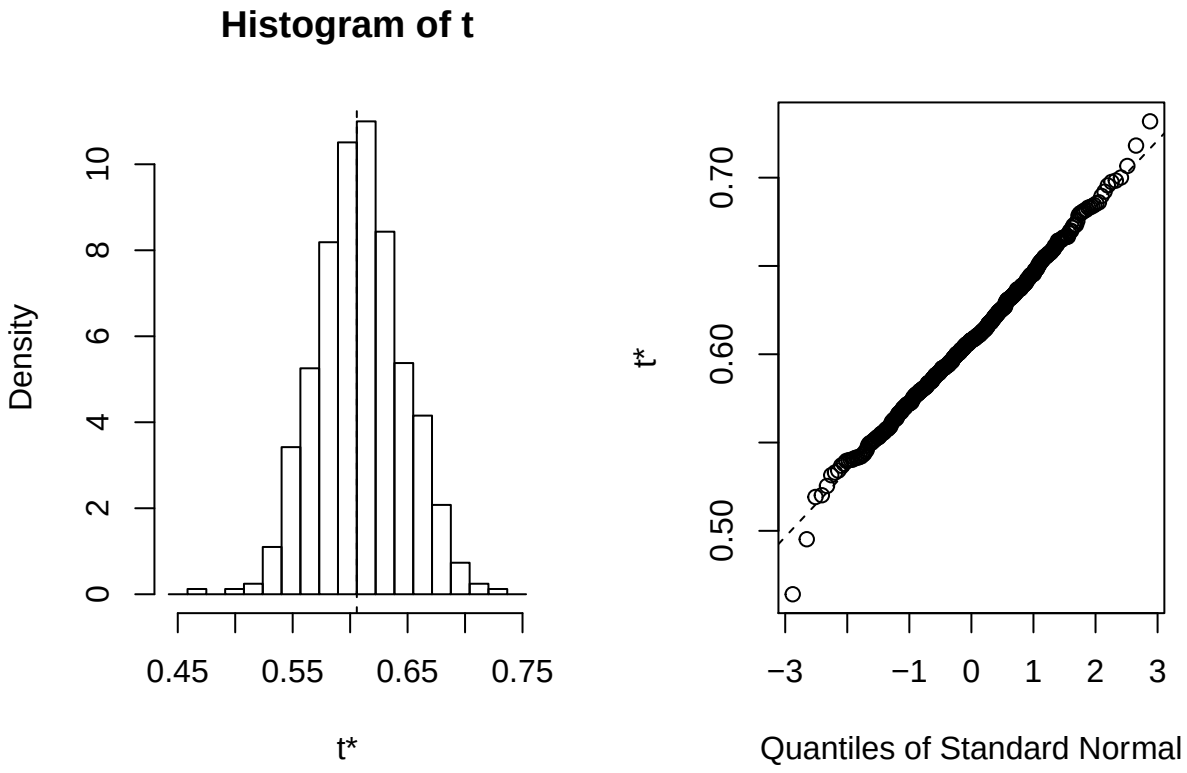
PART 6: ORIGINAL VS WEIGHTED BALT COMPARISION WITH BOOTSTRAPING

#1. Create a function to calculate the overall survival hazard ratio (HR) between tertile 1 and tertile 3.

```
HR <-  function(dataset, i){
  dataset2 <- dataset[i,]
  OSmonths <- dataset2$OS.months
  OSstatus <- dataset2$OS
  my.surv.object <- Surv(time=OSmonths, event=OSstatus)
  cox<-coxph(my.surv.object ~  Tertile, data=dataset2, na.action=na.omit)
  return(exp(cbind(coef(cox),confint(cox))))
  #return(coefficients(cox))
}
HR(X)
```

#2. Test the performance of the weighted Balt score with bootstrapping.

```
##Create a new variable that is the weighted balt tertiles.
X <- ALL_RTChT
X$tertile <- cut(X$balt_weighted, quantile(X$balt_weighted, c(0, 1/3, 2/3, 1), na.rm=TRUE),
                 na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile <- ifelse(X$tertile == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile == "High", "low TGF
X$Tertile <- as.character(X$Tertile)
##Check that the HR function works well.
HR(X)
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  Tertile, data=X, na.action=na.omit)
summary(cox)
##Create 500 different datasets with bootstraping and calculate the tertile 1 vs 3 HR in each of them.
library("boot")
set.seed(0)
results <- boot(data=X, statistic=HR, R=500)
plot(results, index=1)
```

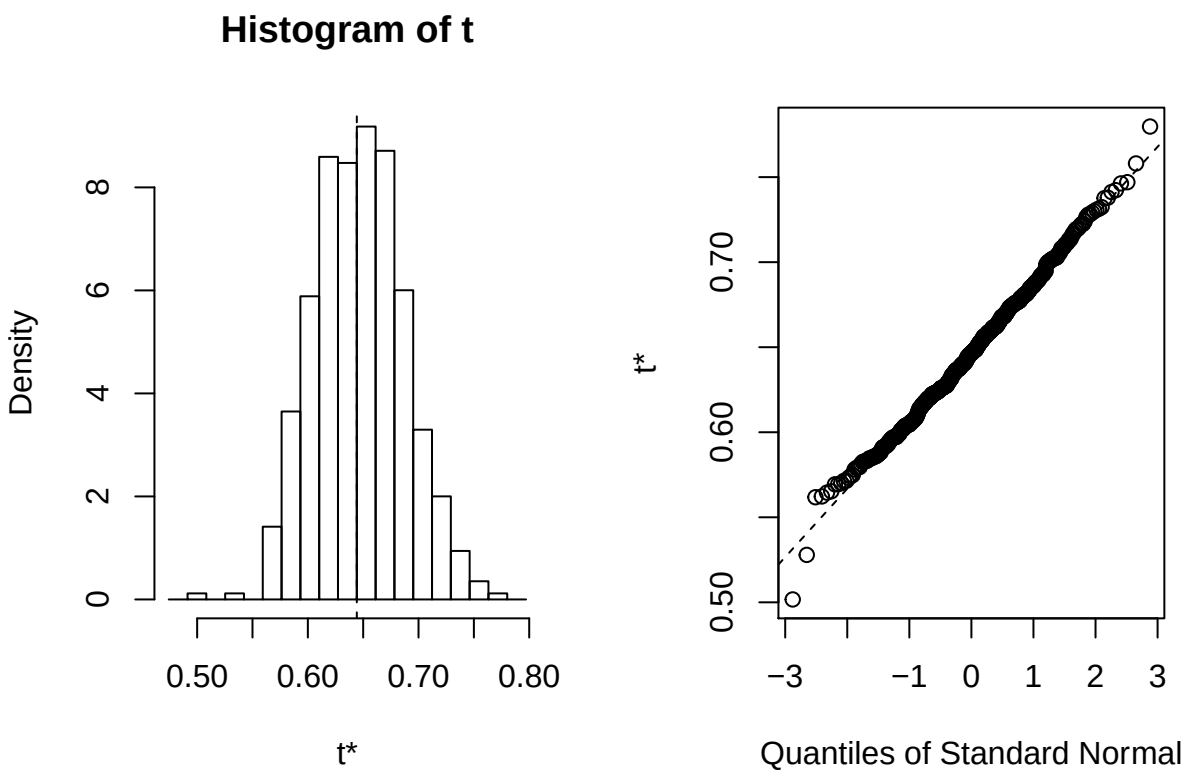



```
##Create a dataset with the results.
summary(results)
results$t0
results$t
hazardratios <- data.frame(HR=results$t)
colnames(hazardratios) <- unlist(as.list(c("HR", "Low.CI", "High.CI")))
hazardratios$set <- as.numeric(as.character(rownames(hazardratios)))
hazardratios_balt_weighted <- hazardratios
```

#3. Test the performance of the original Balt score with bootstrapping.

```
##Create a new variable that is the original balt tertiles.
X <- ALL_RTChT
X$tertile <- cut(X$balt_original_ssgsea, quantile(X$balt_original_ssgsea, c(0, 1/3, 2/3, 1), na.rm=TRUE)
                 na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile <- ifelse(X$tertile == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile == "High", "low TGF
X$Tertile <- as.character(X$Tertile)
##Check that the HR function works well.
HR(X)
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  Tertile, data=X, na.action=na.omit)
summary(cox)
##Create 500 different datasets with bootstraping and calculate the tertile 1 vs 3 HR in each of them.
library("boot")
set.seed(0)
```

```
results <- boot(data=X, statistic=HR, R=500)
plot(results, index=1)
```

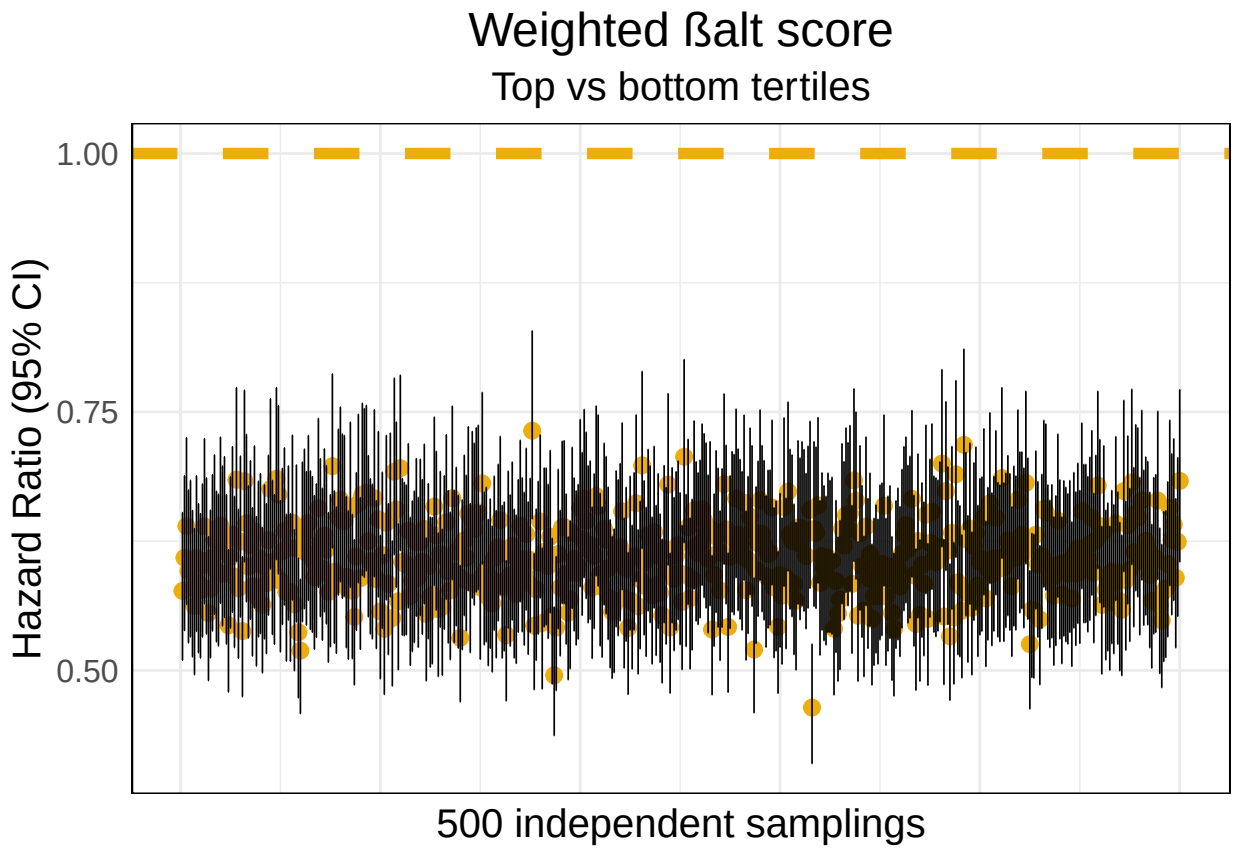



```
##Create a dataset with the results.
summary(results)
results$t0
results$t
hazardratios <- data.frame(HR=results$t)
colnames(hazardratios) <- unlist(as.list(c("HR", "Low.CI", "High.CI")))
hazardratios$set <- as.numeric(as.character(rownames(hazardratios)))
hazardratios_balt_original_ssgsea <- hazardratios
```

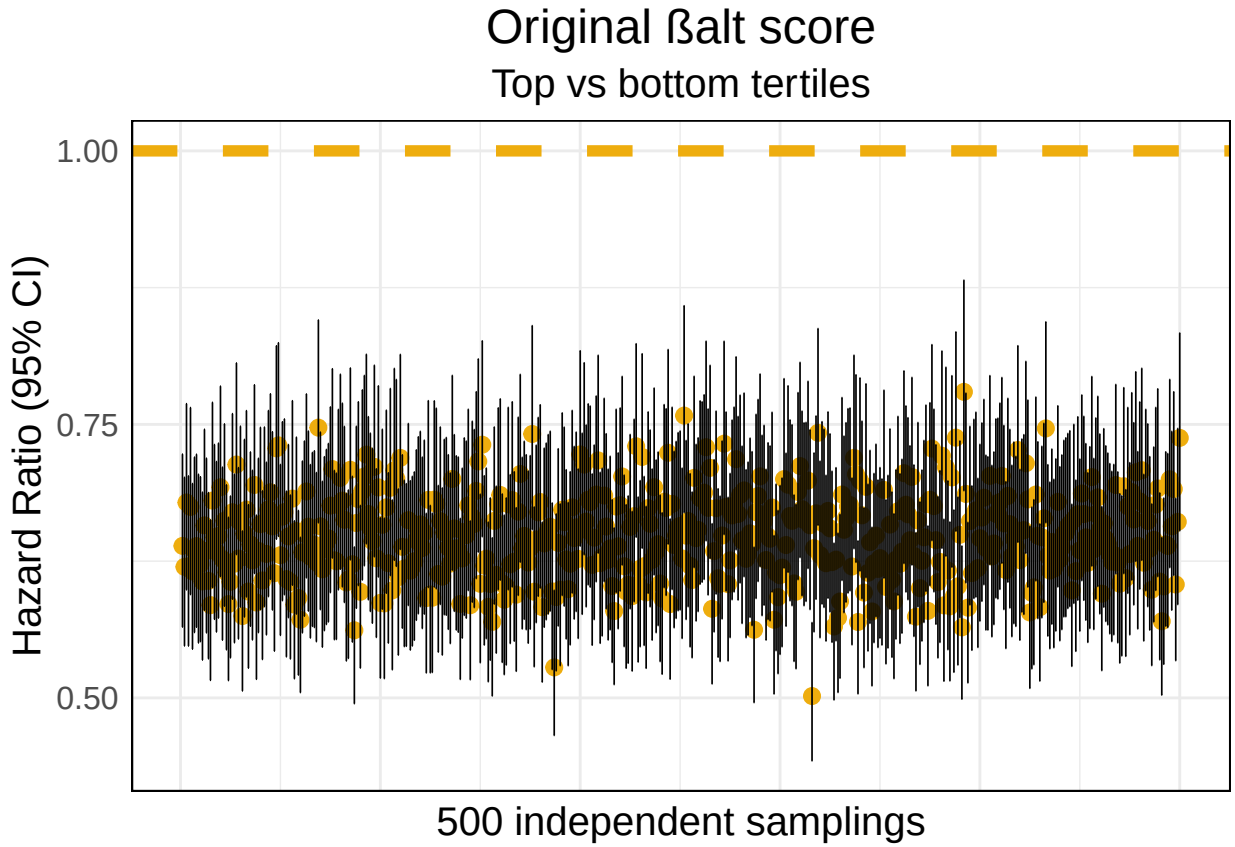
#4.Create forest plots showing the hazard ratios of the scores in the 500 bootstrapped datasets.

```
##Weighted Balt.
ggplot(hazardratios_balt_weighted, aes(x=set, y=HR, ymin=Low.CI, ymax=High.CI)) +
  geom_point(size=2.5, colour="darkgoldenrod2", stroke = 0.5,position=position_dodge(width = 0.7)) +
  geom_errorbar(aes(ymin=Low.CI, ymax=High.CI),width=0.5,cex=0.3) +
  geom_hline(yintercept=1, lty=2, col="darkgoldenrod2", cex=2) +
  xlab("500 independent samplings")+ ylab("Hazard Ratio (95% CI)") +
  scale_y_continuous(breaks = seq(0, 100, by = 0.25)) +
  #scale_x_continuous(breaks = seq(-5, 105)) +
  labs(title="Weighted  alt score", subtitle = "Top vs bottom tertiles") +
  theme_minimal() +
  theme(text = element_text(size=15), axis.text.x=element_blank(),
```

```
      plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5),
      panel.border = element_rect(colour = "black", size=0.5, fill=NA)) #PDF 6x11.
```




```
##Original Balt.
ggplot(hazardratios_balt_original_ssgsea, aes(x=set, y=HR, ymin=Low.CI, ymax=High.CI)) +
  geom_point(size=2.5, colour="darkgoldenrod2", stroke = 0.5,position=position_dodge(width = 0.7)) +
  geom_errorbar(aes(ymin=Low.CI, ymax=High.CI),width=0.5,cex=0.3) +
  geom_hline(yintercept=1, lty=2, col="darkgoldenrod2", cex=2) +
  xlab("500 independent samplings")+ ylab("Hazard Ratio (95% CI)") +
  scale_y_continuous(breaks = seq(0, 100, by = 0.25)) +
  #scale_x_continuous(breaks = seq(-5, 105)) +
  labs(title="Original  alt score", subtitle = "Top vs bottom tertiles") +
  theme_minimal() +
  theme(text = element_text(size=15), axis.text.x=element_blank(),
      plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5),
      panel.border = element_rect(colour = "black", size=0.5, fill=NA)) #PDF 6x11.
```



#5. Create a dataset with the hazard ratios of the original and the weighted Balt scores.

```
hazardratios_balt_original_ssgsea$set <- " alt original"
hazardratios_balt_weighted$set <- " alt weighted"
hazardratios <- rbind(hazardratios_balt_original_ssgsea, hazardratios_balt_weighted)
```

#6. Test for statistical significance the differences on the hazard ratios of the original and the wegihted Balt scores.

```
wilcox.test(HR ~ set, data=hazardratios) #Mann-Whitney test.
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  HR by set
## W = 189437, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(HR ~ set, data=hazardratios, paired = TRUE) #Paired T-test.
```

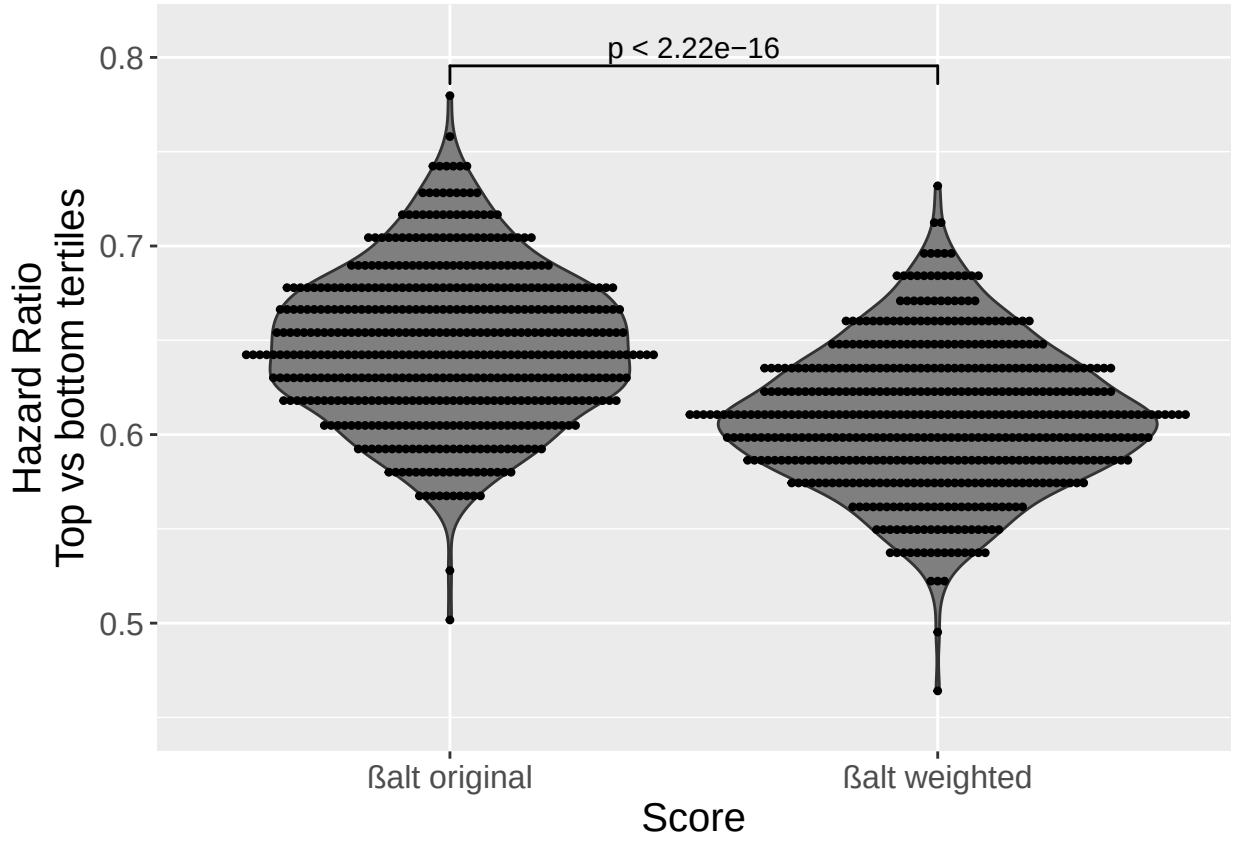
```
##
##  Paired t-test
##
## data:  HR by set
```

```
## t = 46.269, df = 499, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03675839 0.04001855
## sample estimates:
## mean of the differences
##              0.03838847
```

#7. Create violinplots comparing the hazard ratios of the original and the wegihted Balt scores.

```
p<-ggplot(data=hazardratios, aes(x=set, y=HR, fill=set)) +
  geom_violin() +
  scale_fill_manual(values=c("grey50", "grey50")) +
  labs(x = "Score", y = "Hazard Ratio\nTop vs bottom tertiles") +
  geom_dotplot(binaxis='y', stackdir='center', dotsize=0.3, fill="black") +
  theme(text = element_text(size=15)) + theme(legend.title=element_text(size=14)) +
  theme(axis.text.x = element_text(angle = 0, size = 12)) +
  rremove("legend"); p
```

```
##Add p-values to the plot.
library("ggpubr")
my_comparisons <- list(c(" alt original", " alt weighted"))
p + stat_compare_means(method = "wilcox.test", comparisons = my_comparisons,
                       label = "p.format",  #p.signif or p.format
                       bracket.size=0.5) + ylim(0.45, 0.81) #PDF 5x6.5.
```



#8. Calculate the mean hazard ratio of the original and the wegihted Balt scores.

```
tapply(hazardratios$HR, hazardratios$set, mean)
```

```
## ßalt original ßalt weighted
##     0.6472127     0.6088242
```

PART 7: CHECK THE PROPORTIONAL HAZARDS ASSUMPTION

#1.Create new variables that are the original and the weighted balt tertiles.

```
X <- ALL_RTChT
X$tertile_original <- cut(X$balt_original_ssgsea, quantile(X$balt_original_ssgsea, c(0, 1/3, 2/3, 1), na
                  na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile_original <- ifelse(X$tertile_original == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile_or
X$Tertile_original <- as.character(X$Tertile_original)
X$tertile_weighted <- cut(X$balt_weighted, quantile(X$balt_weighted, c(0, 1/3, 2/3, 1), na.rm=TRUE),
                          na.rm=TRUE, include.lowest=TRUE, labels = c("Low", "Middle", "High"))
X$Tertile_weighted <- ifelse(X$tertile_weighted == "Low", "high TGF  and low ALTEJ", ifelse(X$tertile_we
X$Tertile_weighted <- as.character(X$Tertile_weighted)
```

#2. Check the proportional hazards assumption for univariate Cox regressions of the weighted Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile_weighted)
summary(cox)
test.ph <- cox.zph(cox)
test.ph
ggcoxzph(test.ph)
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_weighted)
summary(cox)
test.ph <- cox.zph(cox)
test.ph
ggcoxzph(test.ph)
```

#3. Check the proportional hazards assumption for multivariate Cox regressions of the weighted Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
cox<-coxph(my.surv.object ~  X$Tertile_weighted + X$age + X$stage)
summary(cox)
```

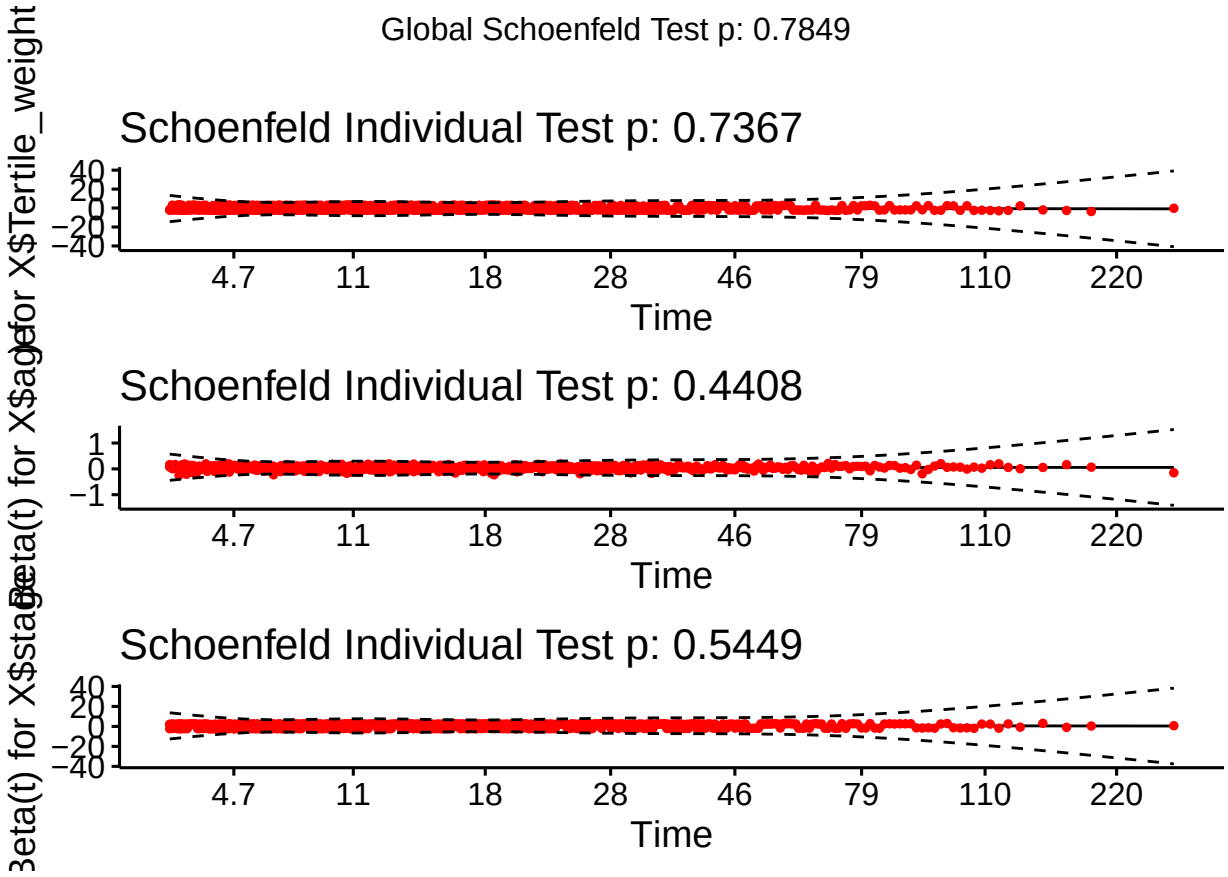
```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile_weighted + X$age +
##     X$stage)
##
```

```
##   n= 2121, number of events= 709
##    (2476 observations deleted due to missingness)
##
##                                              coef exp(coef)  se(coef)      z
## X$Tertile_weightedlow TGFß and high ALTEJ -0.543249  0.580858  0.081855 -6.637
## X$age                                      0.038005  1.038737  0.003004 12.651
## X$stageIII-IV                              0.577804  1.782120  0.077648  7.441
##                                             Pr(>|z|)
## X$Tertile_weightedlow TGFß and high ALTEJ 3.21e-11 ***
## X$age                                      < 2e-16 ***
## X$stageIII-IV                             9.97e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                           exp(coef) exp(-coef) lower .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.5809     1.7216    0.4948
## X$age                                        1.0387     0.9627    1.0326
## X$stageIII-IV                                1.7821     0.5611    1.5305
##                                           upper .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.6819
## X$age                                        1.0449
## X$stageIII-IV                                2.0751
##
## Concordance= 0.669  (se = 0.011 )
## Likelihood ratio test= 291.1  on 3 df,   p=<2e-16
## Wald test            = 261  on 3 df,   p=<2e-16
## Score (logrank) test = 269.4  on 3 df,   p=<2e-16
```

```
test.ph <- cox.zph(cox)
test.ph
```

```
##                     chisq df    p
## X$Tertile_weighted 0.113  1 0.74
## X$age              0.594  1 0.44
## X$stage            0.367  1 0.54
## GLOBAL             1.068  3 0.78
```

```
ggcoxzph(test.ph)
```



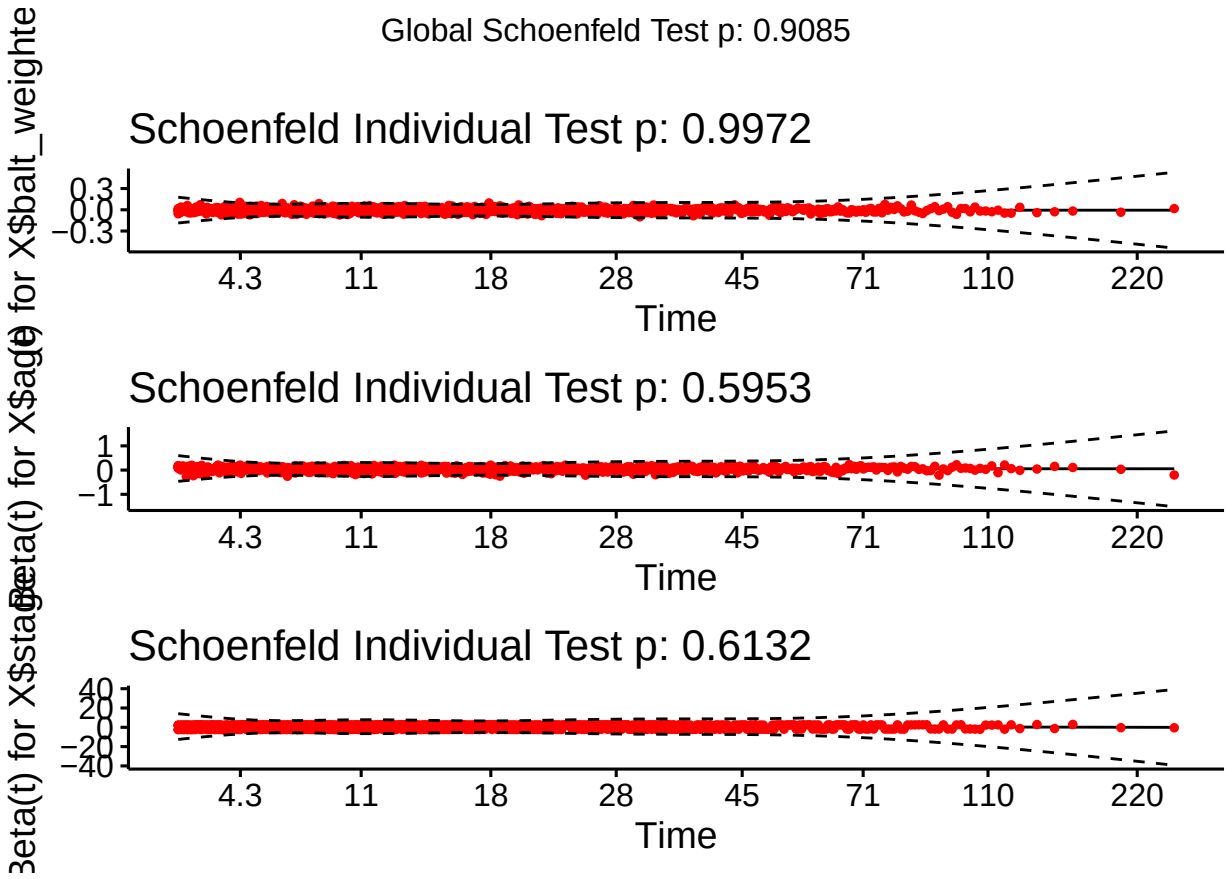

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_weighted + X$age + X$stage)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$balt_weighted + X$age + X$stage)
##
##   n= 3144, number of events= 1032
##    (1453 observations deleted due to missingness)
##
##                        coef  exp(coef)   se(coef)      z Pr(>|z|)
## X$balt_weighted -0.0063648  0.9936554  0.0008865 -7.179    7e-13 ***
## X$age            0.0387044  1.0394631  0.0025923 14.930   <2e-16 ***
## X$stageIII-IV    0.6773486  1.9686511  0.0650529 10.412   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
## X$balt_weighted    0.9937      1.006    0.9919    0.9954
## X$age              1.0395      0.962    1.0342    1.0448
## X$stageIII-IV      1.9687      0.508    1.7330    2.2364
##
## Concordance= 0.672  (se = 0.009 )
## Likelihood ratio test= 426  on 3 df,   p=<2e-16
## Wald test            = 381.1  on 3 df,   p=<2e-16
## Score (logrank) test = 386.5  on 3 df,   p=<2e-16
```

```
test.ph <- cox.zph(cox)
test.ph
```

```
##                     chisq df    p
## X$balt_weighted 1.26e-05  1 1.00
## X$age           2.82e-01  1 0.60
## X$stage         2.56e-01  1 0.61
## GLOBAL          5.47e-01  3 0.91
```

```
ggcoxzph(test.ph)
```



#4. Check the proportional hazards assumption for univariate Cox regressions of the original Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile_original)
summary(cox)
test.ph <- cox.zph(cox)
test.ph
ggcoxzph(test.ph)
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_original_ssgsea)
summary(cox)
```

```
test.ph <- cox.zph(cox)
test.ph
ggcoxzph(test.ph)
```

#5. Check the proportional hazards assumption for multivariate Cox regressions of the original Balt score.

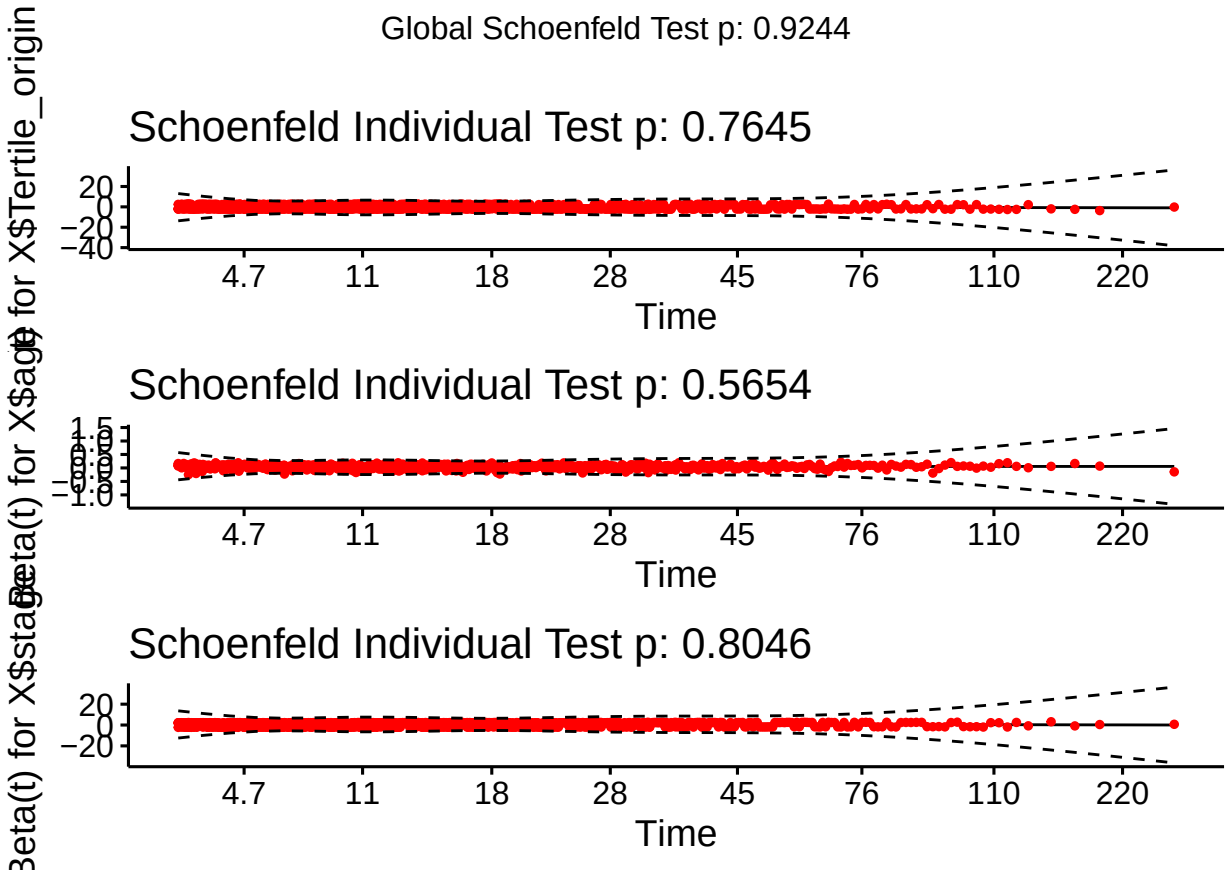
```
##As a categoric variable (tertiles 1 vs 3).
cox<-coxph(my.surv.object ~  X$Tertile_original + X$age + X$stage)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile_original + X$age +
##     X$stage)
##
##   n= 2127, number of events= 697
##    (2470 observations deleted due to missingness)
##
##                                              coef exp(coef)  se(coef)      z
## X$Tertile_originallow TGFß and high ALTEJ -0.476765  0.620788  0.079711 -5.981
## X$age                                      0.039738  1.040538  0.003013 13.189
## X$stageIII-IV                              0.582881  1.791191  0.077843  7.488
##                                             Pr(>|z|)
## X$Tertile_originallow TGFß and high ALTEJ 2.22e-09 ***
## X$age                                      < 2e-16 ***
## X$stageIII-IV                             7.00e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                           exp(coef) exp(-coef) lower .95
## X$Tertile_originallow TGFß and high ALTEJ    0.6208     1.6109     0.531
## X$age                                        1.0405     0.9610     1.034
## X$stageIII-IV                                1.7912     0.5583     1.538
##                                           upper .95
## X$Tertile_originallow TGFß and high ALTEJ    0.7258
## X$age                                        1.0467
## X$stageIII-IV                                2.0864
##
## Concordance= 0.673  (se = 0.011 )
## Likelihood ratio test= 297.8  on 3 df,   p=<2e-16
## Wald test            = 265.3  on 3 df,   p=<2e-16
## Score (logrank) test = 274.9  on 3 df,   p=<2e-16
```

```
test.ph <- cox.zph(cox)
test.ph
```

```
##                      chisq df    p
## X$Tertile_original 0.0898  1 0.76
## X$age              0.3305  1 0.57
## X$stage            0.0612  1 0.80
## GLOBAL             0.4746  3 0.92
```

```
ggcoxzph(test.ph)
```




```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_original_ssgsea + X$age + X$stage)
summary(cox)
```

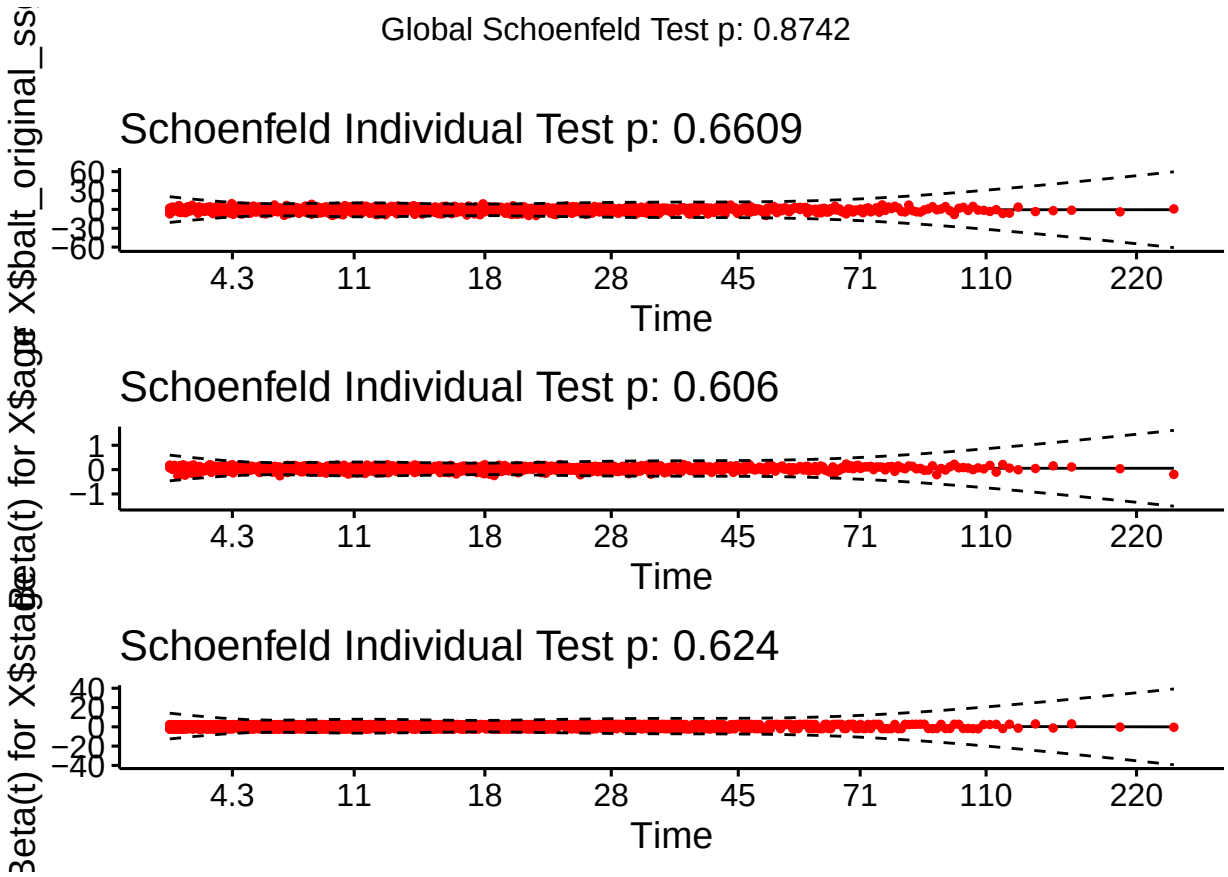
```
## Call:
## coxph(formula = my.surv.object ~ X$balt_original_ssgsea + X$age +
##     X$stage)
##
##   n= 3144, number of events= 1032
##    (1453 observations deleted due to missingness)
##
##                              coef exp(coef)  se(coef)      z Pr(>|z|)
## X$balt_original_ssgsea -0.654523  0.519690  0.099886 -6.553 5.65e-11 ***
## X$age                   0.038636  1.039392  0.002582 14.962  < 2e-16 ***
## X$stageIII-IV           0.689768  1.993254  0.065081 10.599  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## X$balt_original_ssgsea    0.5197     1.9242    0.4273    0.6321
## X$age                     1.0394     0.9621    1.0341    1.0447
## X$stageIII-IV             1.9933     0.5017    1.7546    2.2644
##
```

```
## Concordance= 0.671  (se = 0.009 )
## Likelihood ratio test= 416.1  on 3 df,   p=<2e-16
## Wald test            = 375.5  on 3 df,   p=<2e-16
## Score (logrank) test = 382.5  on 3 df,   p=<2e-16
```

```
test.ph <- cox.zph(cox)
test.ph
```

```
##                          chisq df    p
## X$balt_original_ssgsea 0.192  1 0.66
## X$age                  0.266  1 0.61
## X$stage                0.240  1 0.62
## GLOBAL                 0.696  3 0.87
```

```
ggcoxzph(test.ph)
```



PART 8: COX REGRESSIONS

#1. Calculate the univariate Cox regressions of the weighted Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile_weighted)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile_weighted)
##
##   n= 3051, number of events= 1045
##    (1546 observations deleted due to missingness)
##
##                                            coef exp(coef) se(coef)      z
## X$Tertile_weightedlow TGFß and high ALTEJ -0.50120   0.60580  0.06284 -7.976
##                                           Pr(>|z|)
## X$Tertile_weightedlow TGFß and high ALTEJ 1.51e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                           exp(coef) exp(-coef) lower .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.6058      1.651    0.5356
##                                           upper .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.6852
##
## Concordance= 0.579  (se = 0.008 )
## Likelihood ratio test= 64.63  on 1 df,   p=9e-16
## Wald test            = 63.62  on 1 df,   p=2e-15
## Score (logrank) test = 64.94  on 1 df,   p=8e-16
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_weighted)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$balt_weighted)
##
##   n= 4575, number of events= 1570
##    (22 observations deleted due to missingness)
##
##                        coef  exp(coef)   se(coef)      z Pr(>|z|)
## X$balt_weighted -0.0057660  0.9942506  0.0006702 -8.604   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
## X$balt_weighted    0.9943      1.006    0.9929    0.9956
##
## Concordance= 0.576  (se = 0.008 )
## Likelihood ratio test= 74.97  on 1 df,   p=<2e-16
## Wald test            = 74.02  on 1 df,   p=<2e-16
## Score (logrank) test = 73.99  on 1 df,   p=<2e-16
```

#2. Calculate the multivariate Cox regressions of the weighted Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
cox<-coxph(my.surv.object ~  X$Tertile_weighted + X$age + X$stage)
summary(cox)
```

```
## Call:
```

```
## coxph(formula = my.surv.object ~ X$Tertile_weighted + X$age +
##     X$stage)
##
##   n= 2121, number of events= 709
##    (2476 observations deleted due to missingness)
##
##                                              coef exp(coef)  se(coef)      z
## X$Tertile_weightedlow TGFß and high ALTEJ -0.543249  0.580858  0.081855 -6.637
## X$age                                      0.038005  1.038737  0.003004 12.651
## X$stageIII-IV                              0.577804  1.782120  0.077648  7.441
##                                             Pr(>|z|)
## X$Tertile_weightedlow TGFß and high ALTEJ 3.21e-11 ***
## X$age                                      < 2e-16 ***
## X$stageIII-IV                             9.97e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                           exp(coef) exp(-coef) lower .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.5809     1.7216    0.4948
## X$age                                        1.0387     0.9627    1.0326
## X$stageIII-IV                                1.7821     0.5611    1.5305
##                                           upper .95
## X$Tertile_weightedlow TGFß and high ALTEJ    0.6819
## X$age                                        1.0449
## X$stageIII-IV                                2.0751
##
## Concordance= 0.669  (se = 0.011 )
## Likelihood ratio test= 291.1  on 3 df,   p=<2e-16
## Wald test            = 261  on 3 df,   p=<2e-16
## Score (logrank) test = 269.4  on 3 df,   p=<2e-16
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_weighted + X$age + X$stage)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$balt_weighted + X$age + X$stage)
##
##   n= 3144, number of events= 1032
##    (1453 observations deleted due to missingness)
##
##                        coef  exp(coef)   se(coef)      z Pr(>|z|)
## X$balt_weighted -0.0063648  0.9936554  0.0008865 -7.179    7e-13 ***
## X$age            0.0387044  1.0394631  0.0025923 14.930   <2e-16 ***
## X$stageIII-IV    0.6773486  1.9686511  0.0650529 10.412   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
## X$balt_weighted    0.9937      1.006    0.9919    0.9954
## X$age              1.0395      0.962    1.0342    1.0448
## X$stageIII-IV      1.9687      0.508    1.7330    2.2364
##
## Concordance= 0.672  (se = 0.009 )
```

```
## Likelihood ratio test= 426  on 3 df,   p=<2e-16
## Wald test            = 381.1  on 3 df,   p=<2e-16
## Score (logrank) test = 386.5  on 3 df,   p=<2e-16
```

#3. Calculate the univariate Cox regressions of the original Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
my.surv.object <- Surv(time=X$OS.months, event=X$OS)
cox<-coxph(my.surv.object ~  X$Tertile_original)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile_original)
##
##   n= 3049, number of events= 1027
##    (1548 observations deleted due to missingness)
##
##                                              coef exp(coef) se(coef)      z
## X$Tertile_originallow TGFß and high ALTEJ -0.43970   0.64423  0.06316 -6.962
##                                             Pr(>|z|)
## X$Tertile_originallow TGFß and high ALTEJ 3.36e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                          exp(coef) exp(-coef) lower .95
## X$Tertile_originallow TGFß and high ALTEJ    0.6442      1.552    0.5692
##                                          upper .95
## X$Tertile_originallow TGFß and high ALTEJ    0.7291
##
## Concordance= 0.567  (se = 0.008 )
## Likelihood ratio test= 49.08  on 1 df,   p=2e-12
## Wald test            = 48.47  on 1 df,   p=3e-12
## Score (logrank) test = 49.24  on 1 df,   p=2e-12
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_original_ssgsea)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$balt_original_ssgsea)
##
##   n= 4575, number of events= 1570
##    (22 observations deleted due to missingness)
##
##                             coef exp(coef) se(coef)      z Pr(>|z|)
## X$balt_original_ssgsea -0.61215   0.54219  0.07947 -7.703 1.33e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                         exp(coef) exp(-coef) lower .95 upper .95
## X$balt_original_ssgsea    0.5422      1.844     0.464    0.6336
##
```

```
## Concordance= 0.564  (se = 0.008 )
## Likelihood ratio test= 58.5  on 1 df,   p=2e-14
## Wald test            = 59.34  on 1 df,   p=1e-14
## Score (logrank) test = 59.43  on 1 df,   p=1e-14
```

#4. Calculate the multivariate Cox regressions of the original Balt score.

```
##As a categoric variable (tertiles 1 vs 3).
cox<-coxph(my.surv.object ~  X$Tertile_original + X$age + X$stage)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$Tertile_original + X$age +
##     X$stage)
##
##   n= 2127, number of events= 697
##    (2470 observations deleted due to missingness)
##
##                                              coef exp(coef)  se(coef)      z
## X$Tertile_originallow TGFß and high ALTEJ -0.476765  0.620788  0.079711 -5.981
## X$age                                      0.039738  1.040538  0.003013 13.189
## X$stageIII-IV                              0.582881  1.791191  0.077843  7.488
##                                             Pr(>|z|)
## X$Tertile_originallow TGFß and high ALTEJ 2.22e-09 ***
## X$age                                      < 2e-16 ***
## X$stageIII-IV                             7.00e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                           exp(coef) exp(-coef) lower .95
## X$Tertile_originallow TGFß and high ALTEJ    0.6208     1.6109     0.531
## X$age                                        1.0405     0.9610     1.034
## X$stageIII-IV                                1.7912     0.5583     1.538
##                                           upper .95
## X$Tertile_originallow TGFß and high ALTEJ    0.7258
## X$age                                        1.0467
## X$stageIII-IV                                2.0864
##
## Concordance= 0.673  (se = 0.011 )
## Likelihood ratio test= 297.8  on 3 df,   p=<2e-16
## Wald test            = 265.3  on 3 df,   p=<2e-16
## Score (logrank) test = 274.9  on 3 df,   p=<2e-16
```

```
##As a continuous variable.
cox<-coxph(my.surv.object ~  X$balt_original_ssgsea + X$age + X$stage)
summary(cox)
```

```
## Call:
## coxph(formula = my.surv.object ~ X$balt_original_ssgsea + X$age +
##     X$stage)
##
##   n= 3144, number of events= 1032
```

```
##    (1453 observations deleted due to missingness)
##
##                             coef exp(coef)  se(coef)      z Pr(>|z|)
## X$balt_original_ssgsea -0.654523  0.519690  0.099886 -6.553 5.65e-11 ***
## X$age                   0.038636  1.039392  0.002582 14.962  < 2e-16 ***
## X$stageIII-IV           0.689768  1.993254  0.065081 10.599  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                         exp(coef) exp(-coef) lower .95 upper .95
## X$balt_original_ssgsea    0.5197     1.9242    0.4273    0.6321
## X$age                     1.0394     0.9621    1.0341    1.0447
## X$stageIII-IV             1.9933     0.5017    1.7546    2.2644
##
## Concordance= 0.671  (se = 0.009 )
## Likelihood ratio test= 416.1  on 3 df,   p=<2e-16
## Wald test            = 375.5  on 3 df,   p=<2e-16
## Score (logrank) test = 382.5  on 3 df,   p=<2e-16
```

PART 9: FOREST PLOTS BY CANCER TYPE

#1. Scale the original and weighted Balt scores to the same range.

```
X <- ALL_RTChT
X$type <- as.character(X$type)
library("scales")
X$balt_weighted <- rescale(X$balt_weighted, to = c(0, 1))
X$balt_original_ssgsea <- rescale(X$balt_original_ssgsea, to = c(0, 1))
```

#2. Remove cancer types that don't have enough events.

```
table(X$OS, X$type)
X <- X[-which(X$type=="KIRC" | X$type=="KIRP" | X$type=="LIHC" | X$type=="PCPG" | X$type=="ACC" |
              X$type=="PRAD" | X$type=="SKCM" | X$type=="TGCT" | X$type=="THCA" | X$type=="THYM" |
              X$type=="UCS"),] #Remove cancer types with <=10 events. 4.597->4.112 patients.
```

#3. Calculate the Cox regression coefficients of the original and weighted Balt scores with survival in each cancer type.

```
##Weighted Balt.
coxdf <- X %>% split(.$type)
coxdf <- coxdf %>%
         purrr::map(~ survival::coxph(Surv(OS.time, OS) ~ balt_weighted, data = .) %>%
         broom::tidy(exponentiate = TRUE, conf.int = TRUE)) #Do a Cox regression for each cancer type s
class(coxdf) #list of dataframes
Unicox <- do.call(rbind, coxdf) #Extract elements from the nested list.
Unicox$cancer <- rownames(Unicox)
Unicox$estimate <- log(Unicox$estimate) #Calculate ln of estimate to have the Cox regression coefficien
Unicox$conf.low <- log(Unicox$conf.low)
Unicox$conf.high <- log(Unicox$conf.high)
```

```
Unicox$OSassociation <- ifelse(Unicox$estimate < 0, "Better OS", "Worse OS")
Unicox_weighted <- Unicox
```

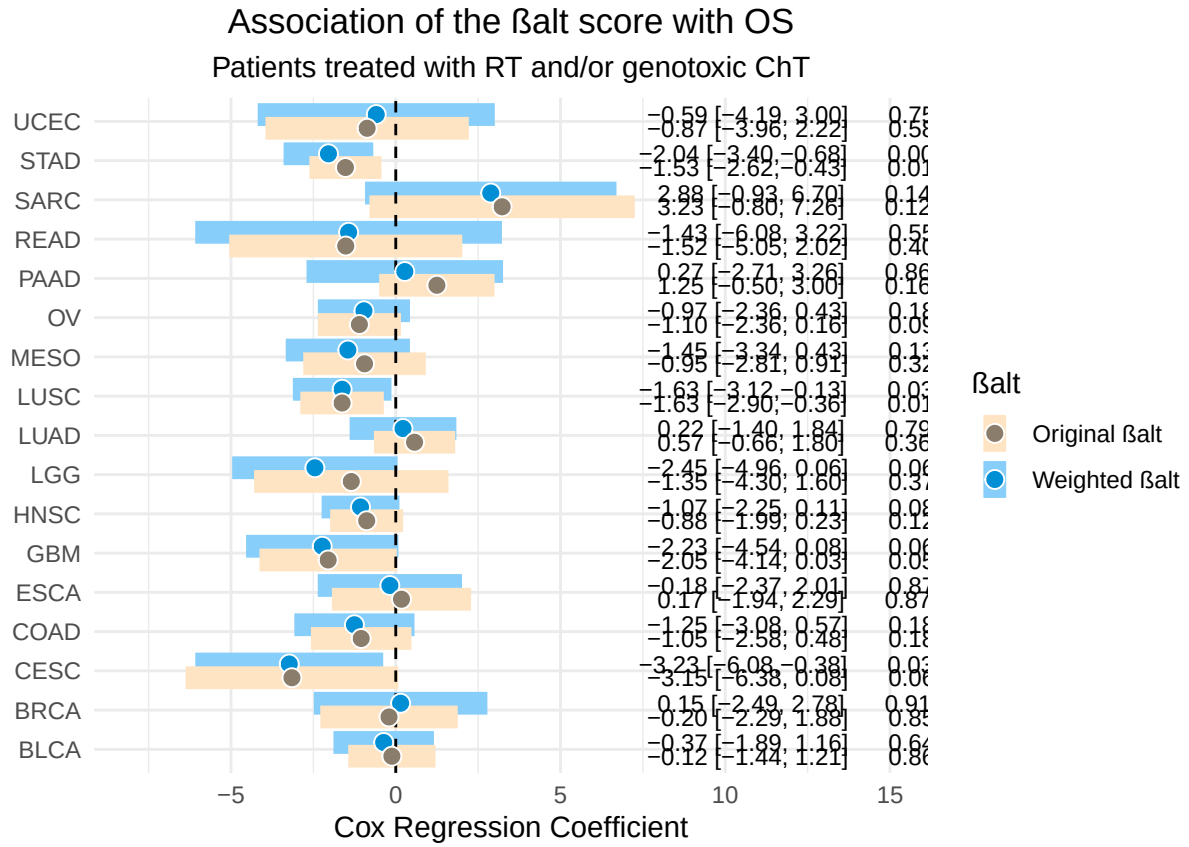
```
##Original Balt.
coxdf <- X %>% split(.$type)
coxdf <- coxdf %>%
         purrr::map(~ survival::coxph(Surv(OS.time, OS) ~ balt_original_ssgsea, data = .) %>%
         broom::tidy(exponentiate = TRUE, conf.int = TRUE)) #Do a Cox regression for each cancer type s
class(coxdf) #list of dataframes
Unicox <- do.call(rbind, coxdf) #Extract elements from the nested list.
Unicox$cancer <- rownames(Unicox)
Unicox$estimate <- log(Unicox$estimate) #Calculate ln of estimate to have the Cox regression coefficien
Unicox$conf.low <- log(Unicox$conf.low)
Unicox$conf.high <- log(Unicox$conf.high)
Unicox$OSassociation <- ifelse(Unicox$estimate < 0, "Better OS", "Worse OS")
Unicox_original_ssgsea <- Unicox
```

#4. Put the results of both scores in the same dataframe.

```
Unicox <- rbind(Unicox_weighted, Unicox_original_ssgsea)
Unicox$ alt <- ifelse(Unicox$term=="balt_original_ssgsea", "Original  alt",
                      ifelse(Unicox$term=="balt_weighted", "Weighted  alt", NA))
Unicox$estimate2 <- format(round(Unicox$estimate, 2), nsmall = 2)
Unicox$conf.low2 <- format(round(Unicox$conf.low, 2), nsmall = 2)
Unicox$conf.high2 <- format(round(Unicox$conf.high, 2), nsmall = 2)
Unicox$conf <- paste(Unicox$conf.low2, Unicox$conf.high2, sep=",")
Unicox$conf <- paste("[", Unicox$conf, sep="")
Unicox$conf <- paste(Unicox$conf, "]", sep="")
Unicox$text <- paste(Unicox$estimate2, Unicox$conf, sep=" ")
Unicox$p.value2 <- format(round(Unicox$p.value, 2), nsmall = 2)
Unicox$p.value2 <- ifelse(Unicox$p.value<0.05, paste(Unicox$p.value2, "*", sep=""), Unicox$p.value2)
Unicox$text <- paste(Unicox$text, Unicox$p.value2, sep="      ")
```

#5. Create a forest plot of association of the original and the weighted BAlt scores with survival in each cancer type.

```
ggplot(Unicox, aes(x=cancer, y=estimate, ymin=conf.low, ymax=conf.high,col= alt, fill= alt)) +
  geom_linerange(size=4,position=position_dodge(width = 0.7)) +
  geom_hline(yintercept=0, lty=2) +
  geom_point(size=3, shape=21, colour="white", stroke = 0.5,position=position_dodge(width = 0.7)) +
  scale_fill_manual(values=c("bisque4", "#008fd5")) + #Dot colors.
  scale_color_manual(values=c("bisque1", "lightskyblue")) + #Bar colors.
  scale_x_discrete(name=" ") +
  scale_y_continuous(name="Cox Regression Coefficient", limits = c(-8, 15)) +
  coord_flip() +
  theme_minimal() +
  labs(title="Association of the  alt score with OS", subtitle = "Patients treated with RT and/or genotc
  theme(plot.title=element_text(hjust = 0.5), plot.subtitle=element_text(hjust = 0.5)) +
  geom_text(aes(y=12, label=text),size=3, position=position_dodge(width=0.7), color="black") #+rremove(
```


Association of the ßalt score with OS
Patients treated with RT and/or genotoxic ChT
UCEC
STAD
SARC
READ
PAAD
OV
MESO
LUSC
LUAD
LGG
HNSC
GBM
ESCA
COAD
CESC
BRCA
BLCA
−0.59 [−4.19, 3.00]
−0.87 [−3.96, 2.22]
−2.04 [−3.40, −0.68]
−1.53 [−2.62, −0.43]
2.88 [−0.93, 6.70]
3.23 [−0.80, 7.26]
−1.43 [−6.08, 3.22]
−1.52 [−5.05, 2.02]
0.27 [−2.71, 3.26]
1.25 [−0.50, 3.00]
−0.97 [−2.36, 0.43]
−1.10 [−2.36, 0.16]
−1.45 [−3.34, 0.43]
−0.95 [−2.81, 0.91]
−1.63 [−3.12, −0.13]
−1.63 [−2.90, −0.36]
0.22 [−1.40, 1.84]
0.57 [−0.66, 1.80]
−2.45 [−4.96, 0.06]
−1.35 [−4.30, 1.60]
−1.07 [−2.25, 0.11]
−0.88 [−1.99, 0.23]
−2.23 [−4.54, 0.08]
−2.05 [−4.14, 0.03]
−0.18 [−2.37, 2.01]
0.17 [−1.94, 2.29]
−1.25 [−3.08, 0.57]
−1.05 [−2.58, 0.48]
−3.23 [−6.08, −0.38]
−3.15 [−6.38, 0.08]
0.15 [−2.49, 2.78]
−0.20 [−2.29, 1.88]
−0.37 [−1.89, 1.16]
−0.12 [−1.44, 1.21]
−5
0
5
10
15
Cox Regression Coefficient
ßalt
Original ßalt
Weighted ßalt

```
#PDF10x9.
```