TCGA BRCA Data Analysis

Asad

5/5/2023

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.4.2     v purrr   1.0.1
## v tibble  3.2.1     v stringr 1.5.0
## v tidyr   1.3.0     v forcats 0.5.1
## v readr   2.1.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggpubr)
library(RColorBrewer)
library(wesanderson)
library(survival)
library(survminer)
```

```
##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

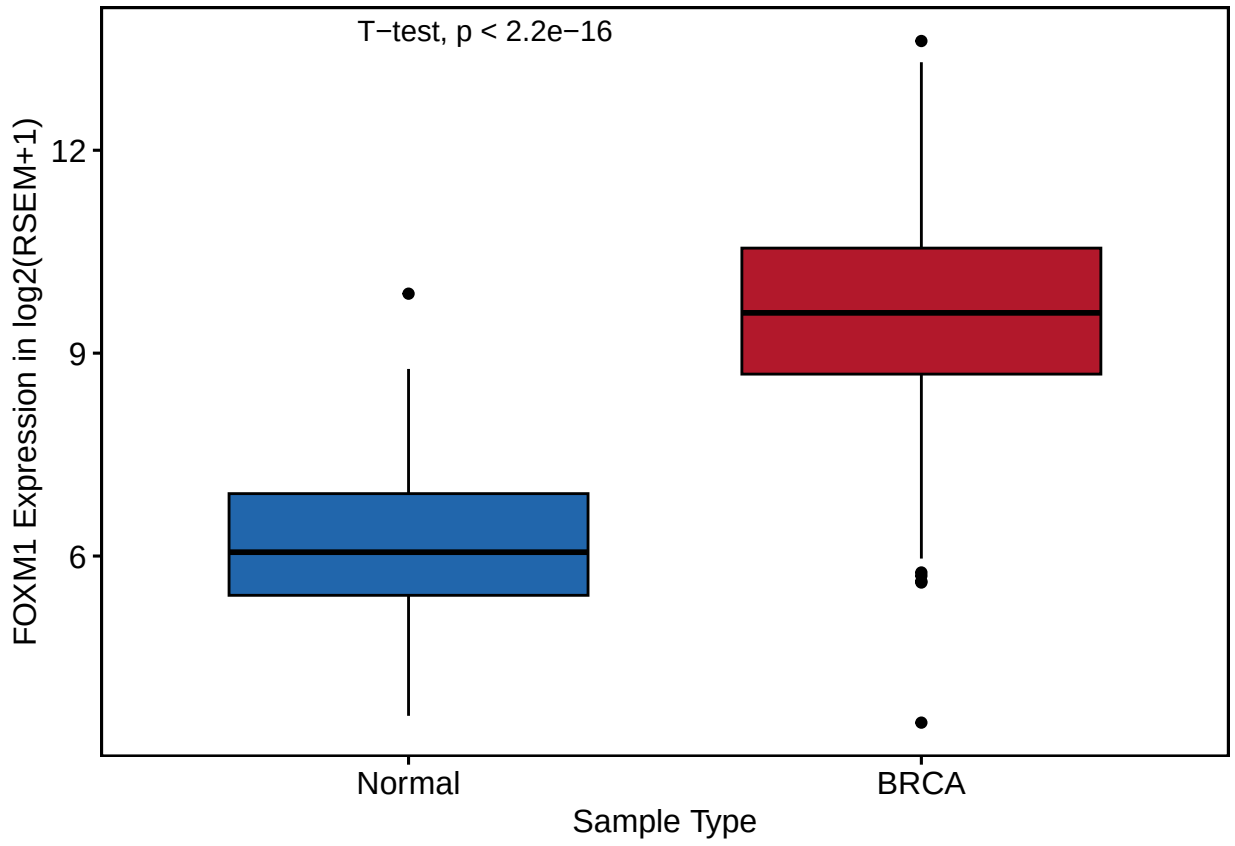
```
#Set directory
setwd('D:/CancerData/TCGA BRCA cBioPortal/BRCA')

#Read RNA-seq data file and make data frame
cancer_df<- read.table('data_mrna_seq_v2_rsem.txt', header = T, sep = '\t')
normal_df<- read.table('data_mrna_seq_v2_rsem_normal_samples.txt', header = T, sep = '\t')

cancer_df<- data.frame(cancer_df)
normal_df<- data.frame(normal_df)

#For example, we'll be working with FOXM1 and MCM2 genes. So, let's filter those
cancer_df<- filter(cancer_df, Hugo_Symbol=="FOXM1" | Hugo_Symbol== "MCM2")
normal_df<- filter(normal_df, Hugo_Symbol=="FOXM1" | Hugo_Symbol== "MCM2")

#Getting col data as row data
cancer_df<- data.frame(t(cancer_df))
normal_df<- data.frame(t(normal_df))

#Save
#write.csv(cancer_df, "cancer_rsem.csv")
#write.csv(normal_df, "normal_rsem.csv")

#Let's prepare normalized and log2 groups
cancer_rsem<- read.csv("cancer_rsem.csv", row.names = 1)
normal_rsem<- read.csv('normal_rsem.csv', row.names = 1)
log2_cancer<- log2(cancer_rsem+1)
log2_normal<- log2(normal_rsem+1)

#Save again
#write.csv(log2_cancer, "log2_cancer.csv")
#write.csv(log2_normal, "log2_normal.csv")

#Read log2 cancer
log2_cancer<- read.csv("log2_cancer.csv")
log2_cancer$PATIENT_ID<- gsub(".01","", as.character(log2_cancer$PATIENT_ID))

#Getting clinical data and manipulating according to expression data
clin_data<- data.frame(read.table('data_clinical_patient.txt', header = T, sep = '\t'))
clin_data$PATIENT_ID<- gsub("-",".", as.character(clin_data$PATIENT_ID))

#Merge two datasets
```

```
Merged_cancer<- left_join(log2_cancer, clin_data, by="PATIENT_ID")
#Merged_log2<- left_join(log2_expression, clin_data, by="PATIENT_ID")


#Save
#write.csv(Merged_cancer, "Merged_cancer.csv")
Merged_cancer<- read.csv('Merged_cancer.csv')
```

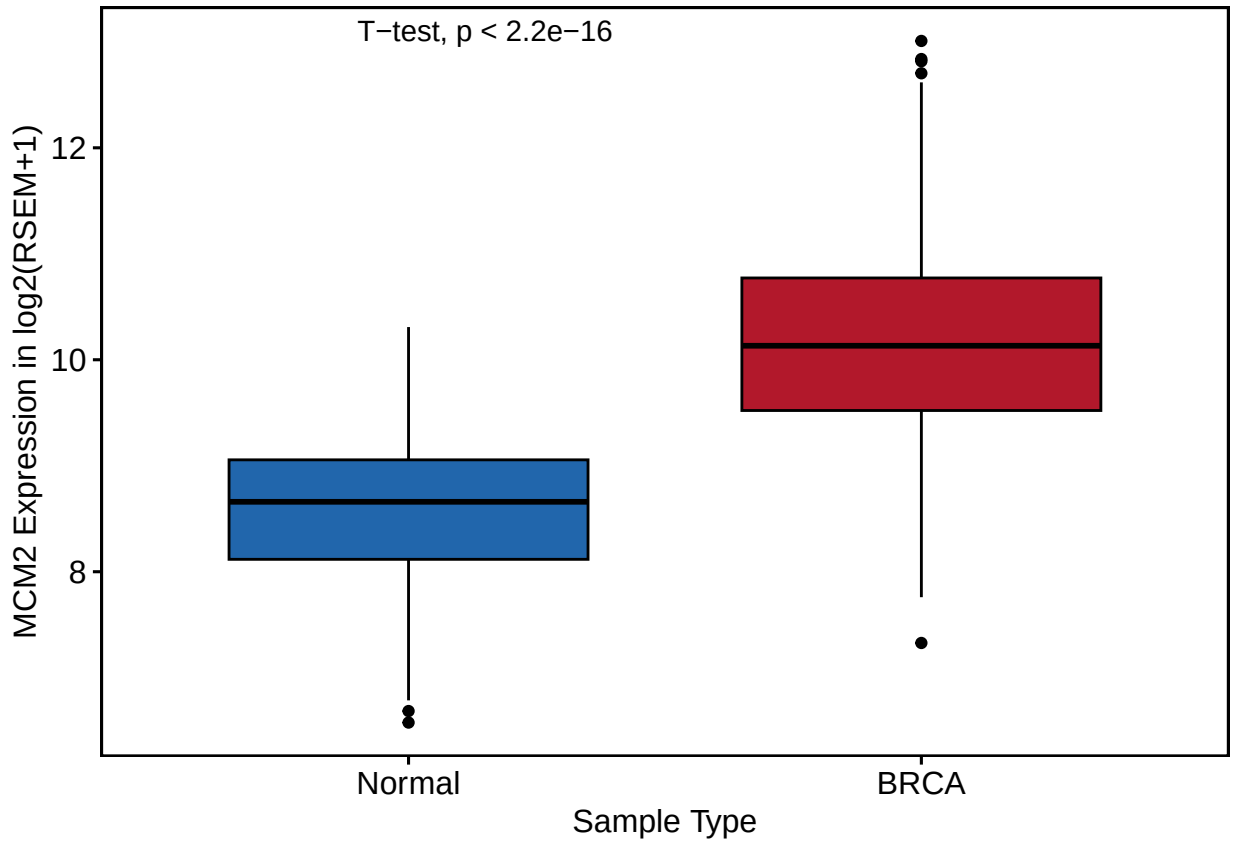
```
#Check expression pattern in accordance with cancer type
cancer_type<- Merged_cancer[,c(2,3,5)]
cancer_type<- na.omit(cancer_type)
colnames(cancer_type)
```

```
## [1] "FOXM1"                "MCM2"                 "CANCER_TYPE_ACRONYM"
```

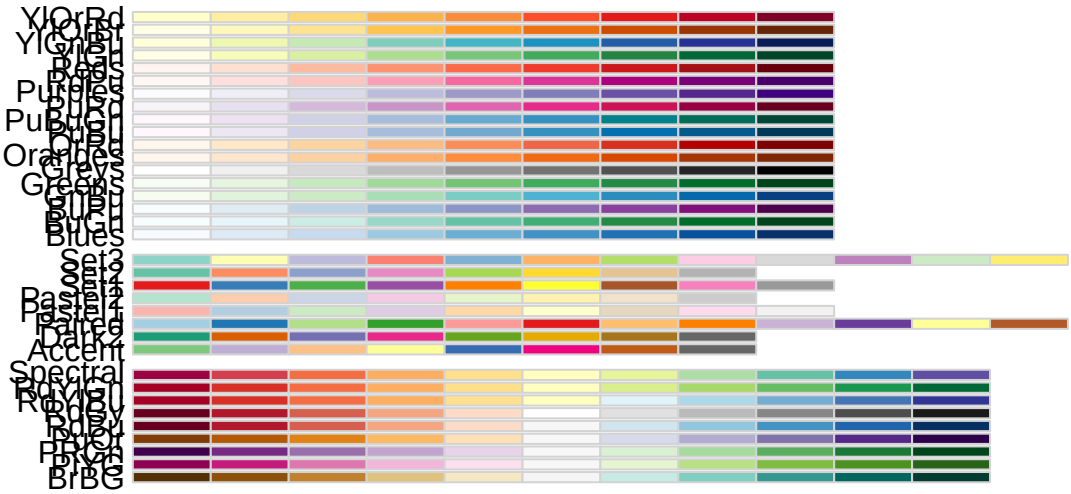
```
FOXM1_Type <- ggboxplot(cancer_type, x = "CANCER_TYPE_ACRONYM", y = "FOXM1",
                          fill = "CANCER_TYPE_ACRONYM", , palette = c("#2166AC","#B2182B"),
                          order = c("Normal", "BRCA"), xlab = "Sample Type",
                           ylab = "FOXM1 Expression in log2(RSEM+1)" ) +
                         stat_compare_means(method = 't.test', paired = F) + border()

FOXM1_Type<- ggpar(FOXM1_Type, legend = 'none')
FOXM1_Type
```



```
MCM2_Type <- ggboxplot(cancer_type, x = "CANCER_TYPE_ACRONYM", y = "MCM2",
                       fill = "CANCER_TYPE_ACRONYM", , palette = c("#2166AC","#B2182B"),
                       order = c("Normal", "BRCA"), xlab = "Sample Type",
                       ylab = "MCM2 Expression in log2(RSEM+1)" ) +
                       stat_compare_means(method = 't.test', paired = F) + border()

MCM2_Type<- ggpar(MCM2_Type, legend = 'none')
MCM2_Type
```




```
#Check expression pattern in accordance with cancer subtype
subtype<- na.omit(Merged_cancer[,2:4])
subtype[subtype==""]<- NA #Filling blank rows with NA values
subtype<- na.omit(subtype) #Removing NA values again
subtype<- subtype[!subtype$SUBTYPE=='BRCA_Normal',]
subtype$SUBTYPE<- gsub("BRCA_","", as.character(subtype$SUBTYPE))


#Change color
display.brewer.all()
```

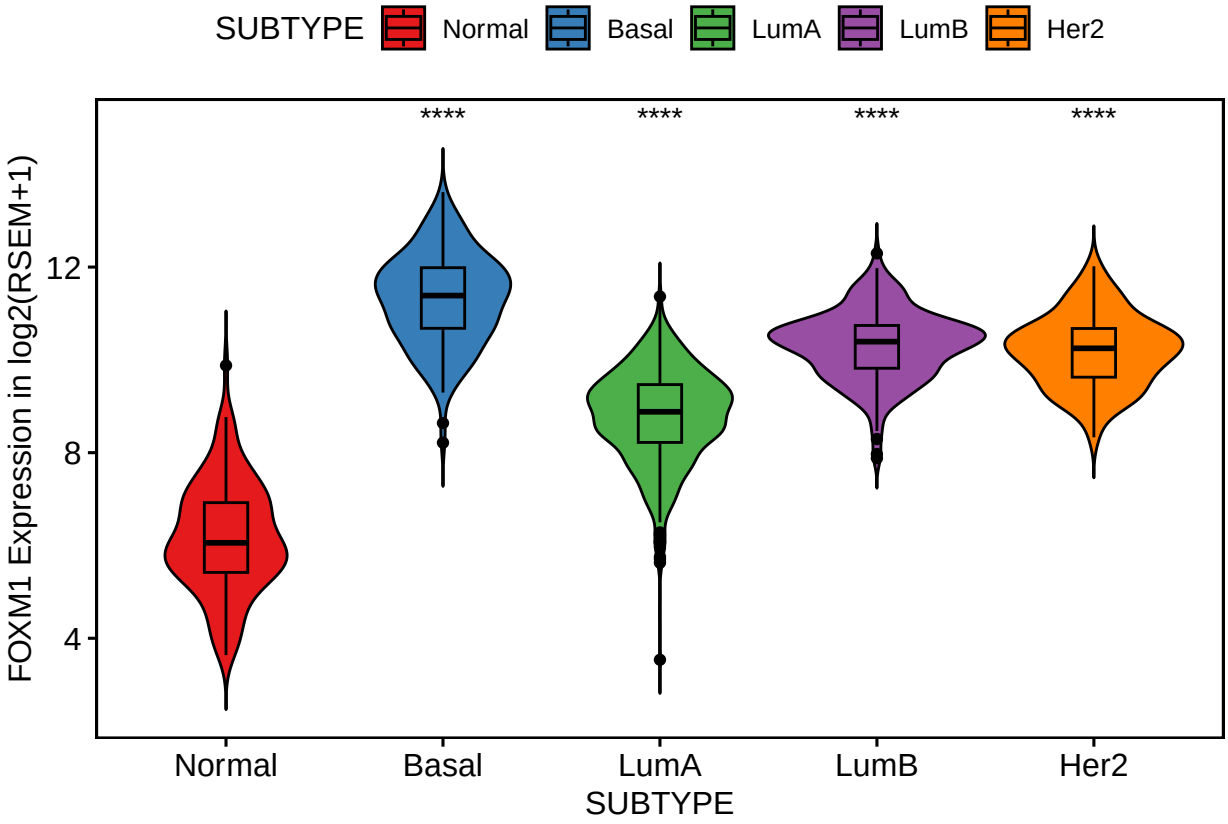



```
display.brewer.pal(n=10,name='Set1')
```

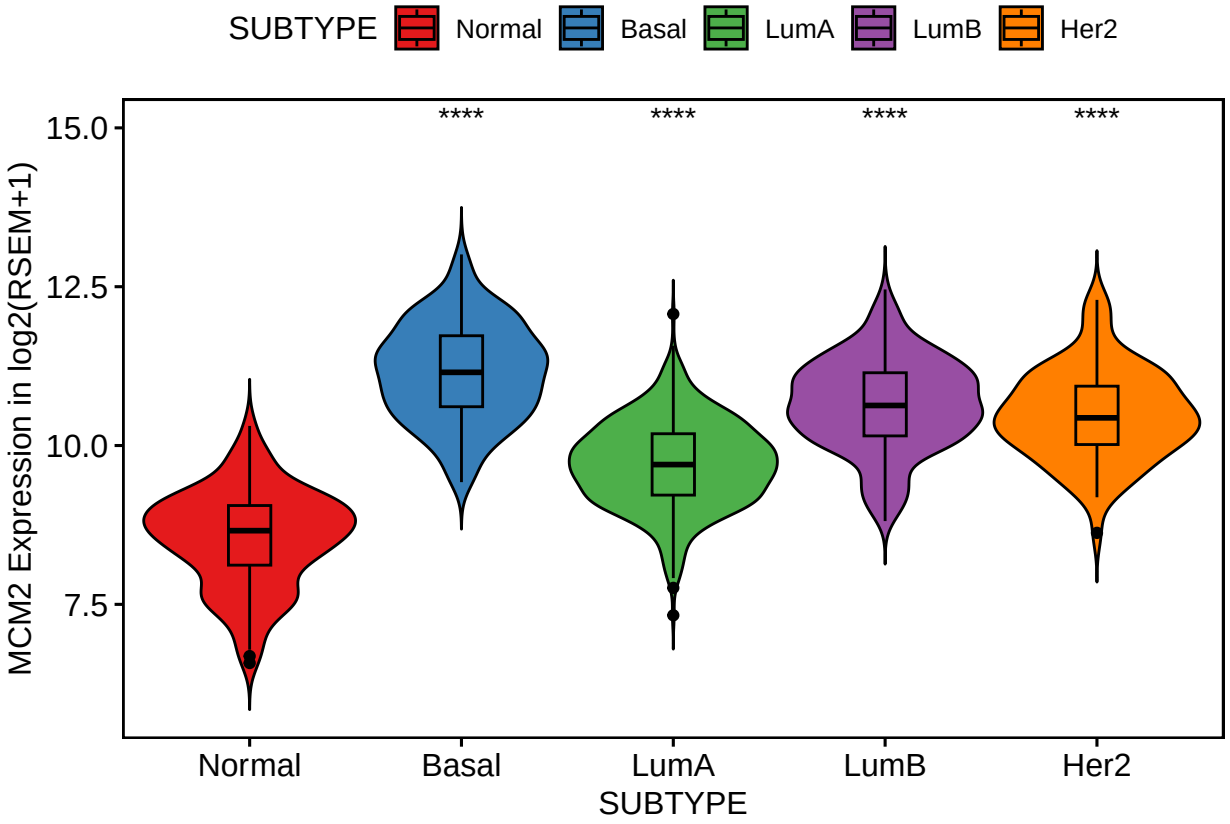
```
## Warning in display.brewer.pal(n = 10, name = "Set1"): n too large, allowed maximum for palette Set1 i
## Displaying the palette you asked for with that many colors
```

Set1 (qualitative)

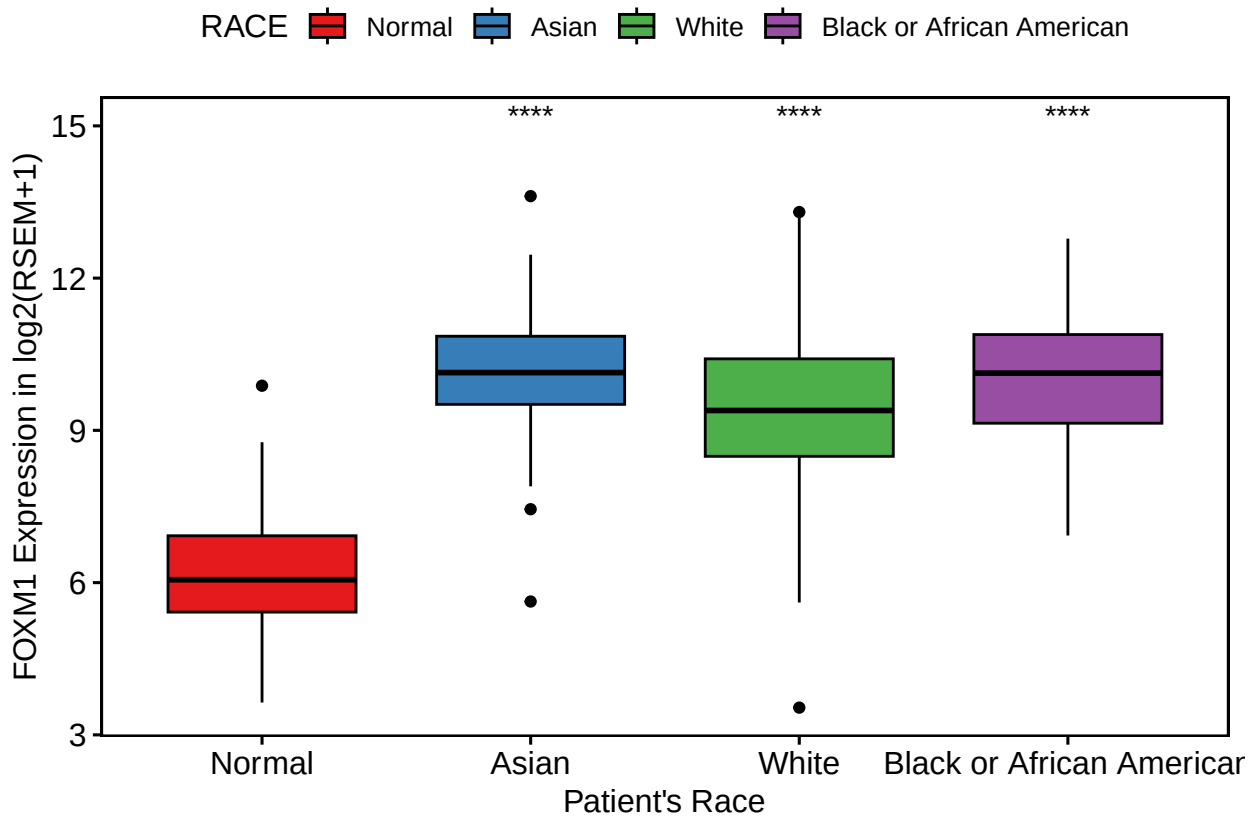
```
comp<- list(c('Normal','Basal'), c('Normal','LumA'), c('Normal','LumB'), c('Normal','Her2'))

FOXM1_Subtype<- ggviolin(subtype, x = "SUBTYPE", y = "FOXM1",
                         fill = "SUBTYPE",  palette = brewer.pal(n=9,name='Set1'),
                         order = c('Normal','Basal', 'LumA', 'LumB', 'Her2'), add = 'boxplot',
                         xlab = "SUBTYPE", ylab = "FOXM1 Expression in log2(RSEM+1)" ) +
                         stat_compare_means(method = 't.test', label = 'p.signif', ref.group = 'Normal'
                                            label.y = 15) + border()
FOXM1_Subtype
```



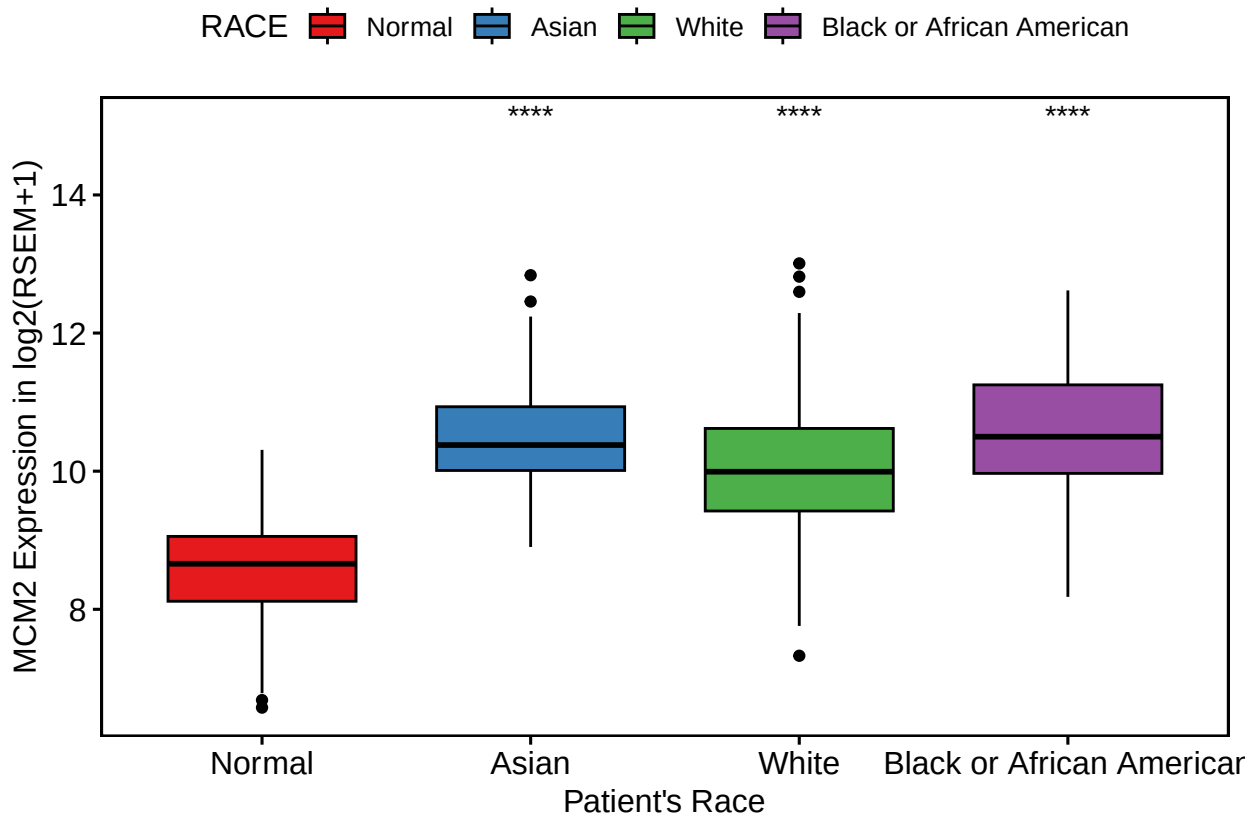

```
MCM2_Subtype<- ggviolin(subtype, x = "SUBTYPE", y = "MCM2",
                        fill = "SUBTYPE",  palette = brewer.pal(n=9,name='Set1'),
                        order = c('Normal','Basal', 'LumA', 'LumB', 'Her2'), add = 'boxplot',
                        xlab = "SUBTYPE", ylab = "MCM2 Expression in log2(RSEM+1)" ) +
                        stat_compare_means(method = 't.test', label = 'p.signif', ref.group = 'Normal'
                                           label.y = 15) + border()
MCM2_Subtype
```




```
#Check expression pattern in accordance with race
race<- Merged_cancer[,c(2,3,28)]
race[race==""]<- NA #Filling blank rows with NA values
race<- na.omit(race)
race<- race[!race$RACE=='American Indian or Alaska Native',]

FOXM1_race<- ggboxplot(race, x = "RACE", y = "FOXM1",
                       fill = "RACE",  palette = brewer.pal(n=9,name='Set1'),
                       order = c('Normal','Asian', 'White', 'Black or African American'),
                       xlab = "Patient's Race", ylab = "FOXM1 Expression in log2(RSEM+1)" ) +
                      stat_compare_means(method = 't.test', ref.group = 'Normal', label = 'p.signif',
                       label.y = 15) + border()
FOXM1_race
```



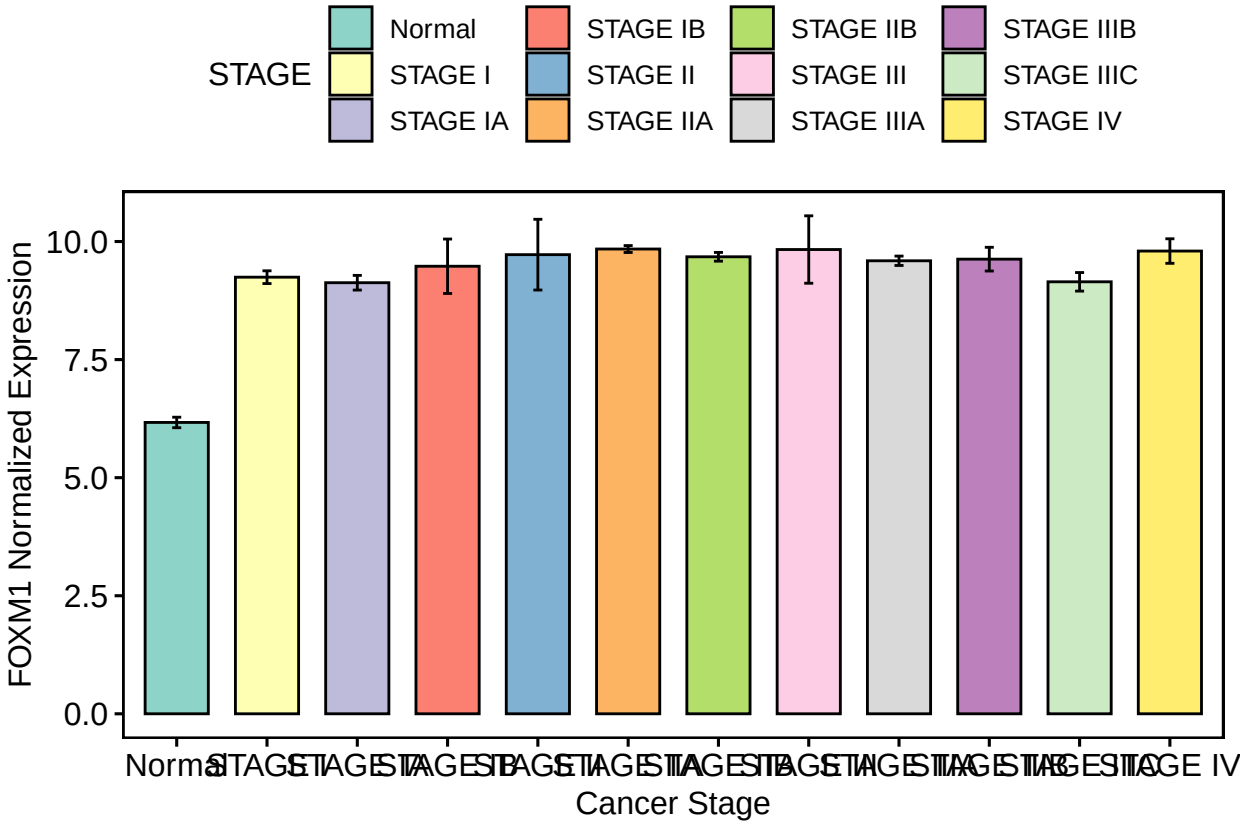

```
MCM2_race<- ggboxplot(race, x = "RACE", y = "MCM2",
                      fill = "RACE",  palette = brewer.pal(n=9,name='Set1'),
                      order = c('Normal','Asian', 'White', 'Black or African American'),
                      xlab = "Patient's Race", ylab = "MCM2 Expression in log2(RSEM+1)" ) +
                     stat_compare_means(method = 't.test', ref.group = 'Normal', label = 'p.signif',
                      label.y = 15) + border()
MCM2_race
```




```
#Cancer Stage
names(Merged_cancer)[5] ="Cancer Type"
stage<- Merged_cancer[, c(2,3,5,9)]
stage[stage==""]<- NA #Filling blank rows with NA values
stage<- na.omit(stage) #Removing NA values again
names(stage)[4] ="STAGE" #Rename column name
stage<- stage[!stage$STAGE=="STAGE X",]

order<- c('Normal', 'STAGE I', "STAGE IA", "STAGE IB", "STAGE II", "STAGE IIA", "STAGE IIB",
          "STAGE III", "STAGE IIIA", "STAGE IIIB", "STAGE IIIC", "STAGE IV")

FOXM1_stage<- ggbarplot(stage, x="STAGE", y='FOXM1', add = 'mean_se', fill ='STAGE',
                      palette = brewer.pal(n = 13, name = "Set3"), position = position_dodge(0.7),
                      xlab = "Cancer Stage", ylab = "FOXM1 Normalized Expression", order = order)+borde
```

```
## Warning in brewer.pal(n = 13, name = "Set3"): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```

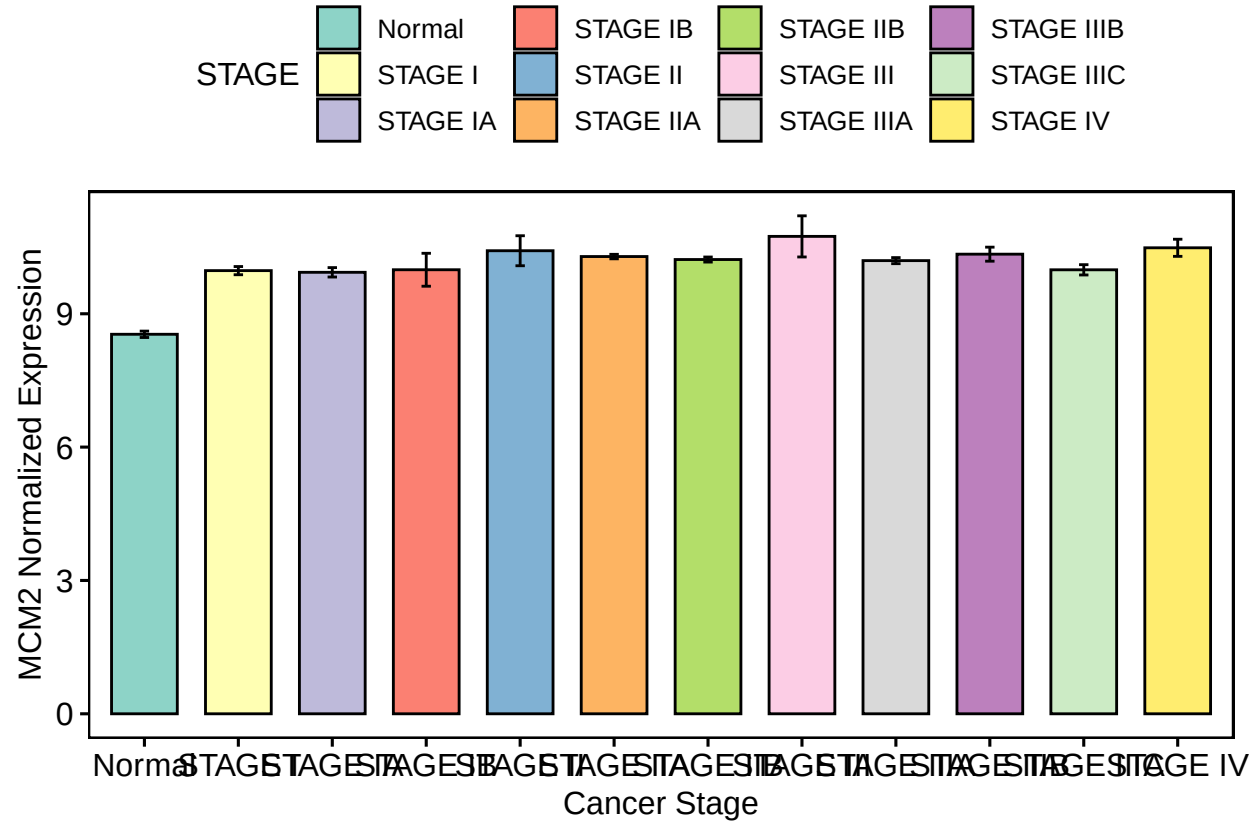
```
FOXM1_stage
```




```
MCM2_stage<- ggbarplot(stage, x="STAGE", y='MCM2', add = 'mean_se', fill ='STAGE',
                       palette = brewer.pal(n = 13, name = "Set3"), position = position_dodge(0.7),
                       xlab = "Cancer Stage", ylab = "MCM2 Normalized Expression", order = order)+borde
```

```
## Warning in brewer.pal(n = 13, name = "Set3"): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```

```
MCM2_stage
```




```
#Performing survival analysis
surv_data<- Merged_cancer[, c(2,3,11,32,33)]
write.csv(surv_data, 'surv_data.csv')
surv_data<- read.csv("surv_data.csv")



#Preparing the dataset
#FOXM1
surv_data$strata<- ifelse(surv_data$FOXM1>=6.056712, "High", "Low")
surv_data$censored<- ifelse(surv_data$OS_STATUS=="0:LIVING", FALSE, TRUE)
colnames(surv_data)
```

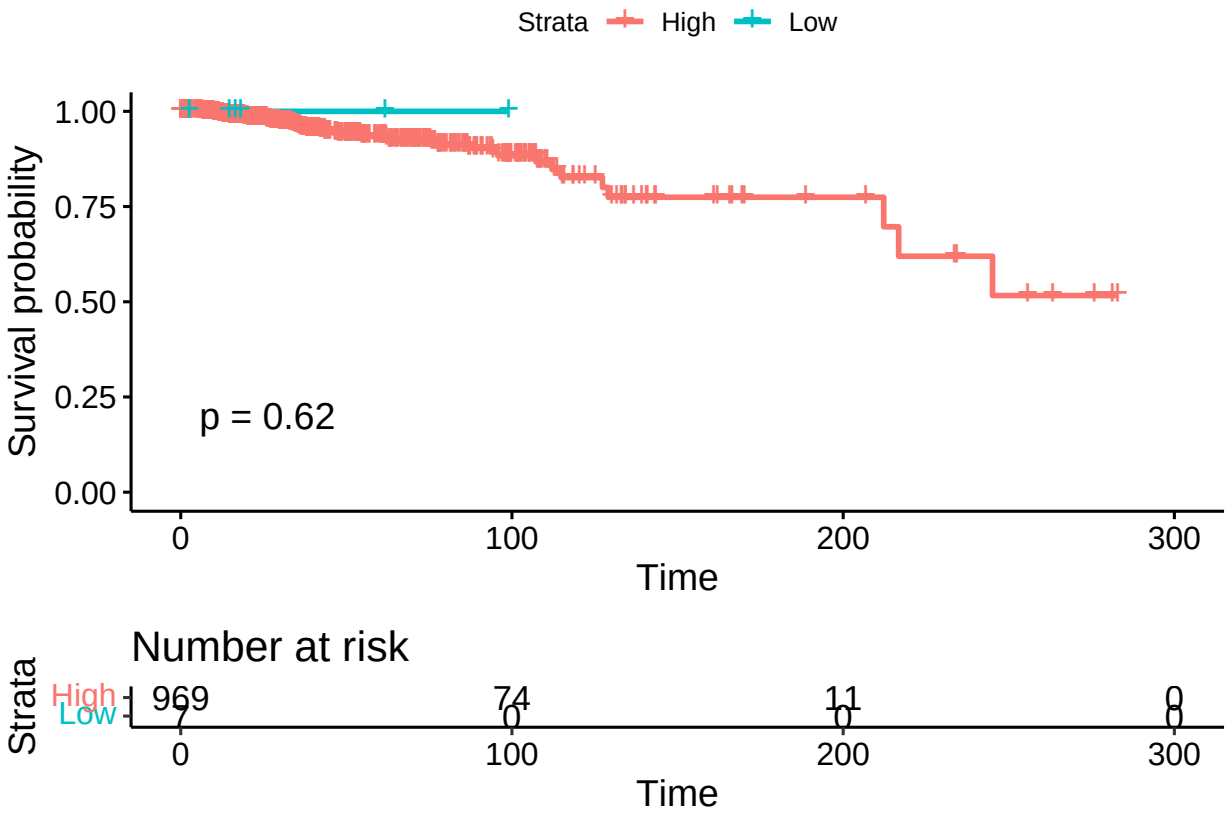
```
## [1] "X"                  "FOXM1"              "MCM2"
## [4] "DAYS_LAST_FOLLOWUP" "OS_STATUS"          "OS_MONTHS"
## [7] "strata"             "censored"
```

```
#Cleaning the dataset
surv_data[surv_data==""]<- NA
surv_data<- na.omit(surv_data)

#Fitting model for survival analysis
fit1 <- survfit(Surv(OS_MONTHS, censored) ~ strata, data = surv_data)
fit1
```

```
## Call: survfit(formula = Surv(OS_MONTHS, censored) ~ strata, data = surv_data)
```

```
##
##                n events median 0.95LCL 0.95UCL
## strata=High 969     47     NA     217      NA
## strata=Low    7      0     NA      NA      NA
```

```
ggsurvplot(fit1,
           data = surv_data,
           pval = T,
           risk.table = T)
```



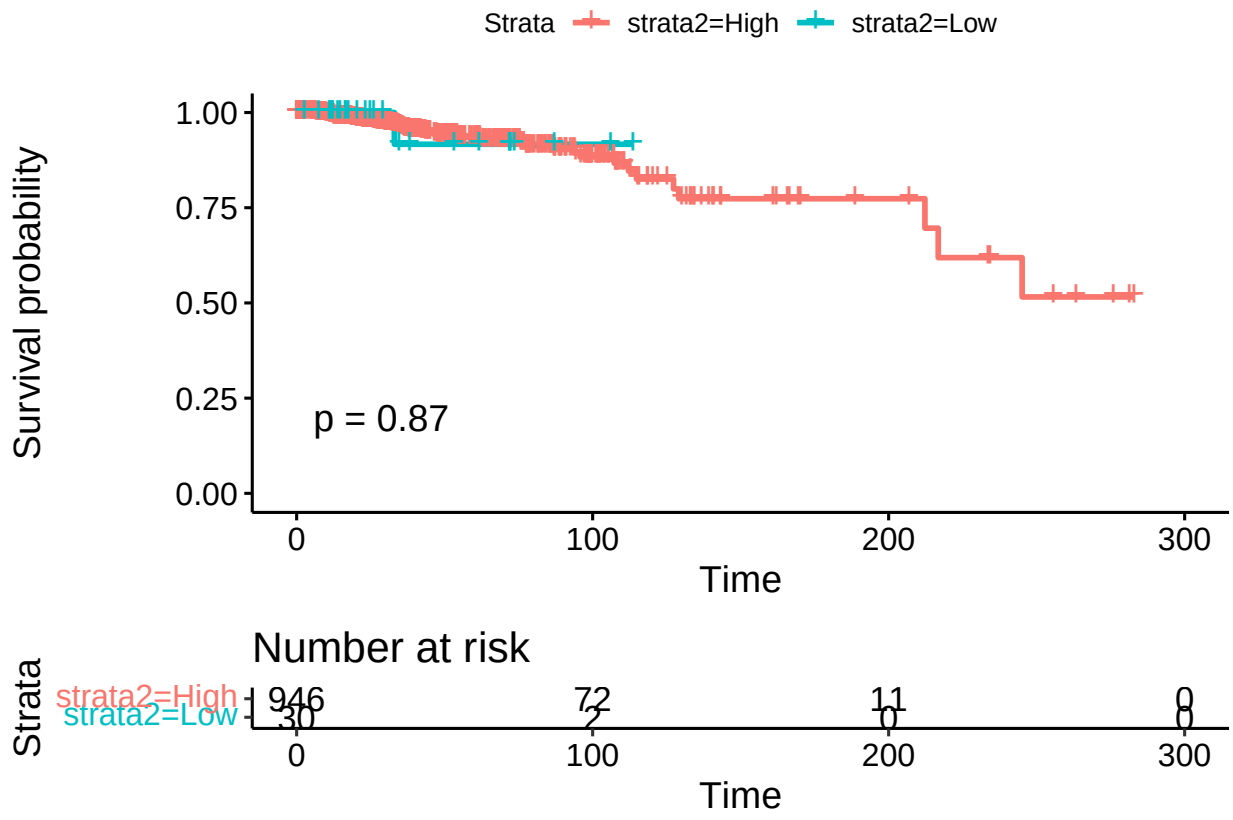

```
#MCM2
surv_data$strata2<- ifelse(surv_data$MCM2>=8.659733, "High", "Low")
surv_data$censored<- ifelse(surv_data$OS_STATUS=="0:LIVING", FALSE, TRUE)
colnames(surv_data)
```

```
## [1] "X"                  "FOXM1"              "MCM2"
## [4] "DAYS_LAST_FOLLOWUP" "OS_STATUS"          "OS_MONTHS"
## [7] "strata"             "censored"           "strata2"
```

```
#Cleaning the dataset
surv_data[surv_data==""]<- NA
surv_data<- na.omit(surv_data)

#Fitting model for survival analysis
fit3 <- survfit(Surv(OS_MONTHS, censored) ~ strata2, data = surv_data)
fit3
```

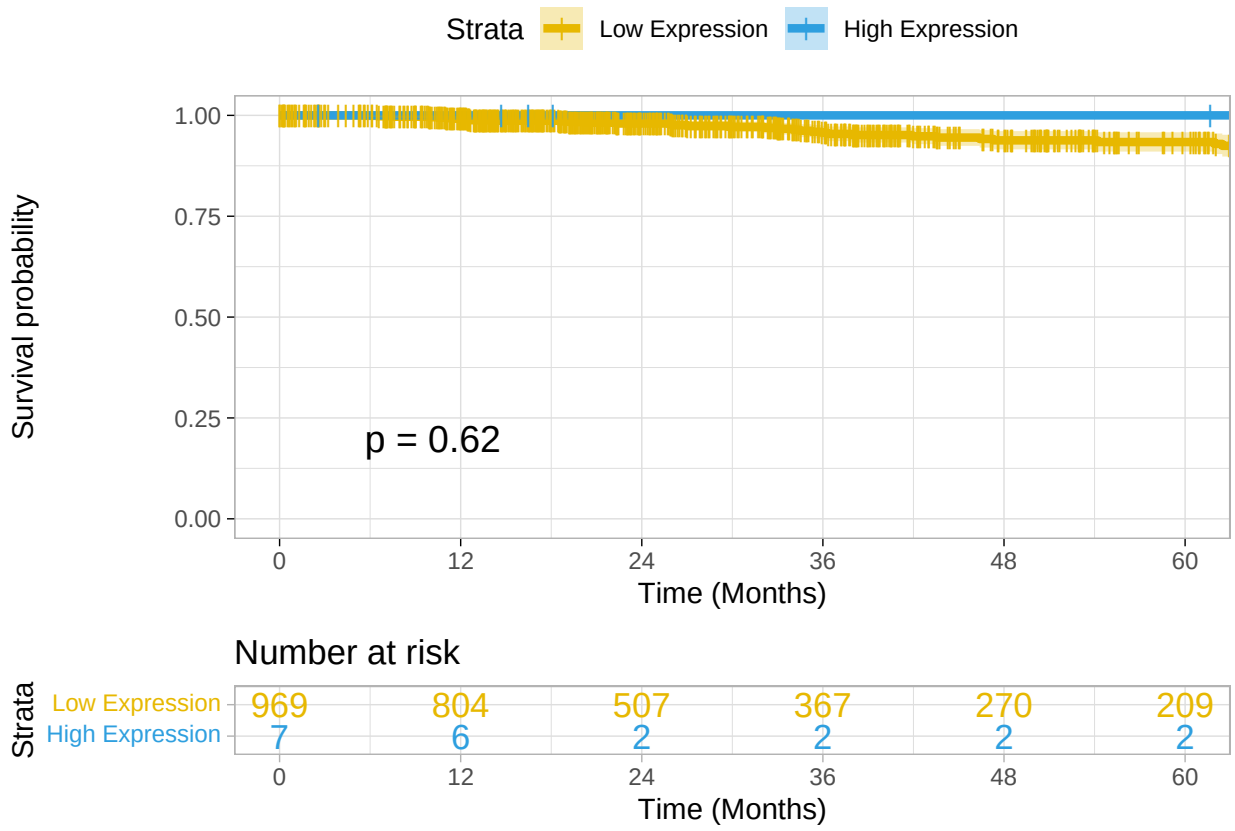
```
## Call: survfit(formula = Surv(OS_MONTHS, censored) ~ strata2, data = surv_data)
##
##                 n events median 0.95LCL 0.95UCL
## strata2=High 946     46     NA     217      NA
## strata2=Low   30      1     NA      NA      NA
```

```
ggsurvplot(fit3,
           data = surv_data,
           pval = T,
           risk.table = T)
```

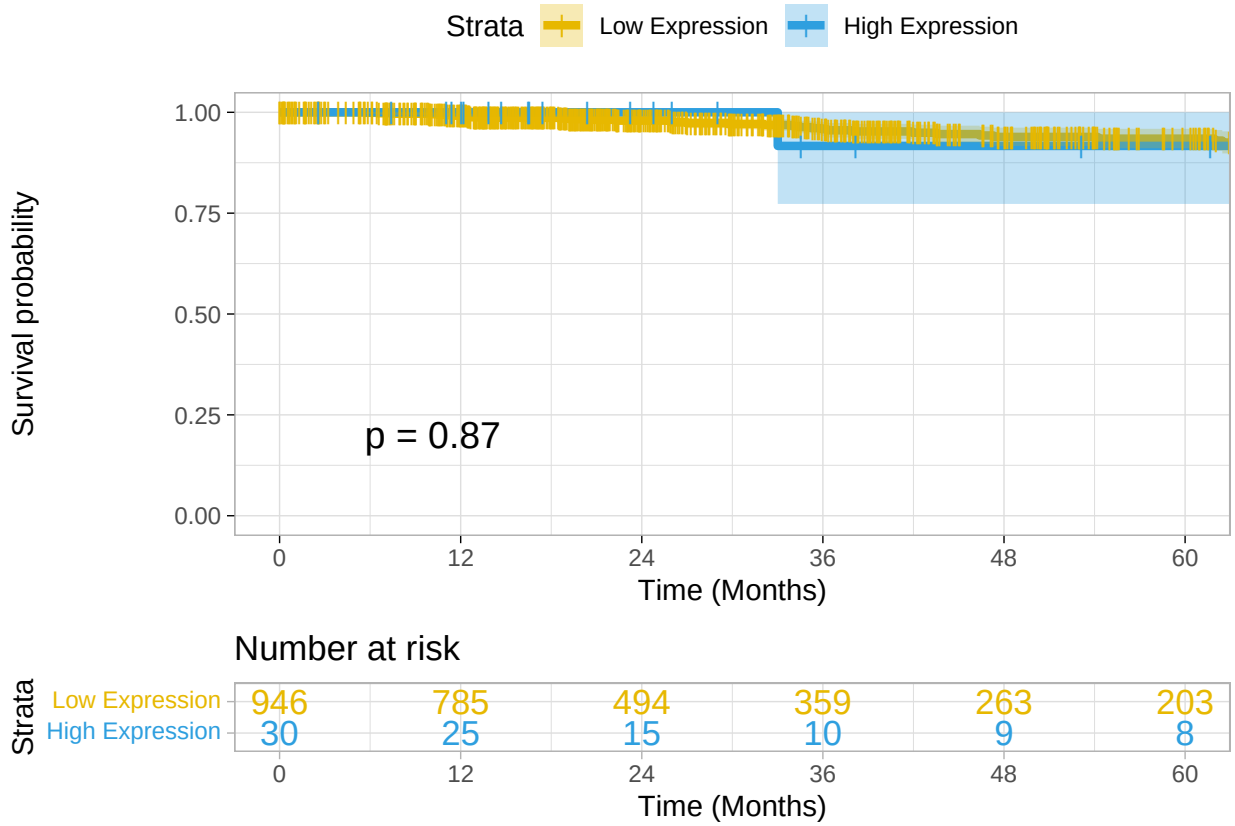



```
fit2 <- survdiff(Surv(OS_MONTHS, censored) ~ strata, data = surv_data)

#Customize KM plot
ggsurvplot(
  fit1,
  data = surv_data,
  size = 1.5,  xlim=c(0, 60),break.x.by = 12,xlab="Time (Months)",
  censor.shape="|", censor.size = 3, # change line size
  palette =
    c("#E7B800", "#2E9FDF"),# custom color palettes
  conf.int = TRUE,          # Add confidence interval
  pval = TRUE,              # Add p-value which will perform log-rank T test
  risk.table = TRUE,        # Add risk table
  risk.table.col = "strata",# Risk table color by groups
```

```
  legend.labs =
    c("Low Expression", "High Expression"),     # Change legend labels
  risk.table.height = 0.25, # Useful to change when you have multiple groups
  ggtheme = theme_light()      # Change ggplot2 theme
)
```

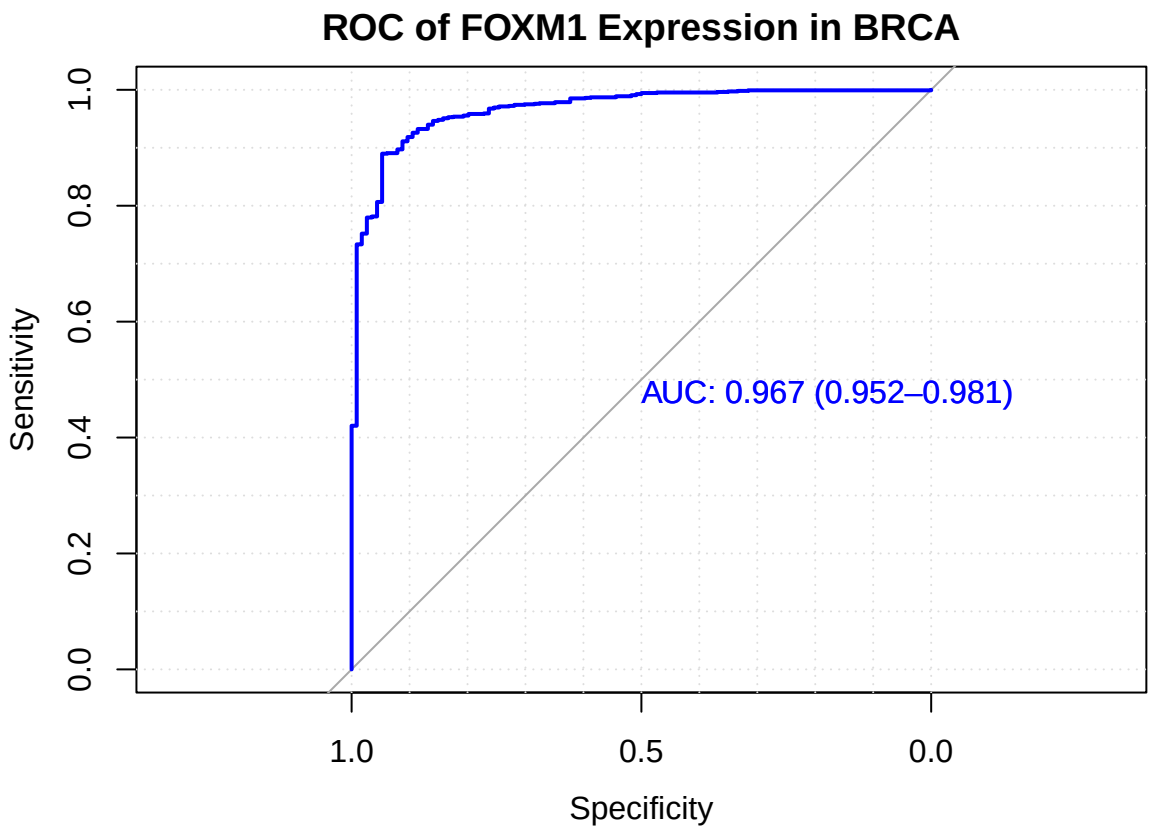



```
ggsurvplot(
  fit3,
  data = surv_data,
  size = 1.5,  xlim=c(0, 60),break.x.by = 12,xlab="Time (Months)",
  censor.shape="|", censor.size = 3, # change line size
  palette =
    c("#E7B800", "#2E9FDF"),# custom color palettes
  conf.int = TRUE,          # Add confidence interval
  pval = TRUE,              # Add p-value which will perform log-rank T test
  risk.table = TRUE,        # Add risk table
  risk.table.col = "strata",# Risk table color by groups
  legend.labs =
    c("Low Expression", "High Expression"),     # Change legend labels
  risk.table.height = 0.25, # Useful to change when you have multiple groups
  ggtheme = theme_light()      # Change ggplot2 theme
)
```



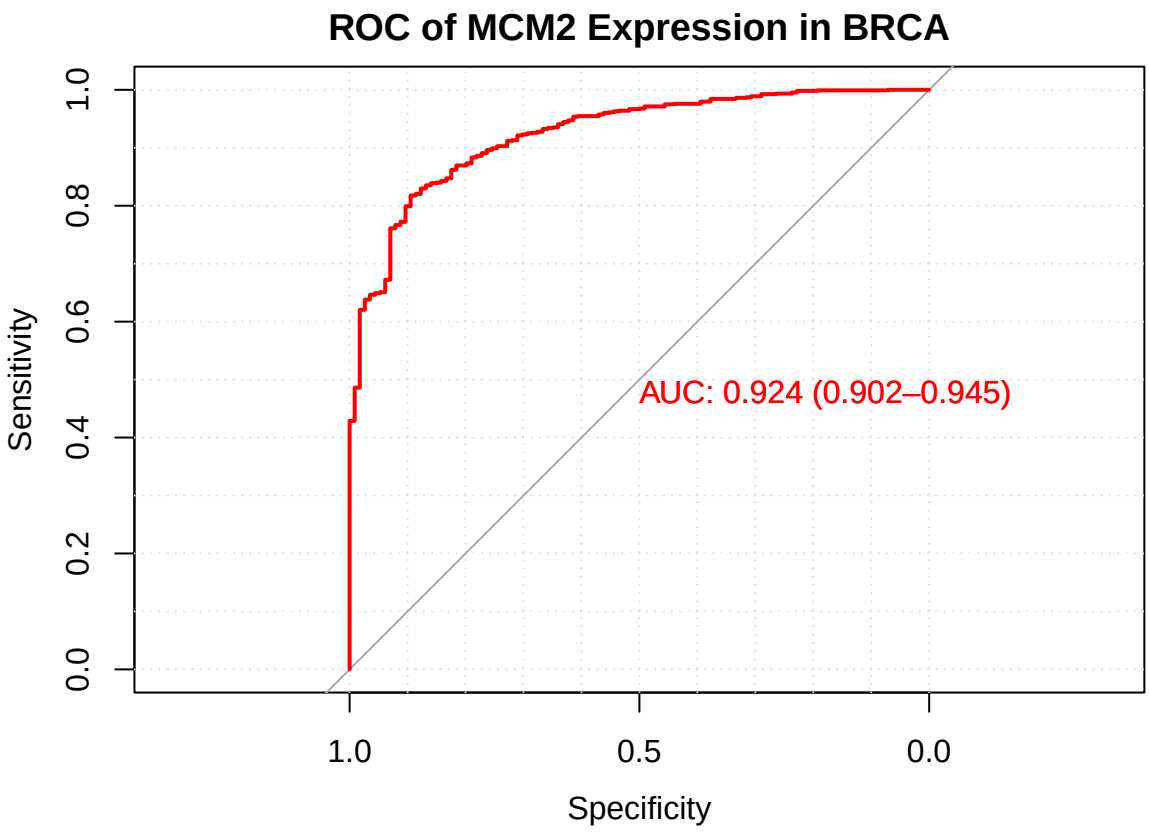

```
#ROC Curve for FOXM1 and MCM2
roc<- read.csv('roc.csv')
rocobj1 <- plot.roc(roc$Sample, roc$FOXM1, levels=c("Normal", "BRCA"),alpha=0.05,
                    ci=T, print.auc=T, percent=F, grid=T, main="ROC of FOXM1 Expression in BRCA",
                    col= 'blue')
```

```
## Setting direction: controls < cases
```




```
rocobj2 <- plot.roc(roc$Sample, roc$MCM2, levels=c("Normal", "BRCA"),alpha=0.05,
                    ci=T, print.auc=T, percent=F, grid=T, main="ROC of MCM2 Expression in BRCA",
                    col= 'red')
```

```
## Setting direction: controls < cases
```


ROC of MCM2 Expression in BRCA
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
AUC: 0.924 (0.902–0.945)
1.0
0.5
0.0
Specificity