Mutation Analysis

2023-11-21

```
clinical <- read.csv(file="patient_data_shared.csv")
mutation <- read.csv(file="mutation_data_shared.csv")
```

```
RNAseq <- read.csv(file="rnaseq_data_shared.csv")
```

```
library(readxl)
library(readr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

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

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

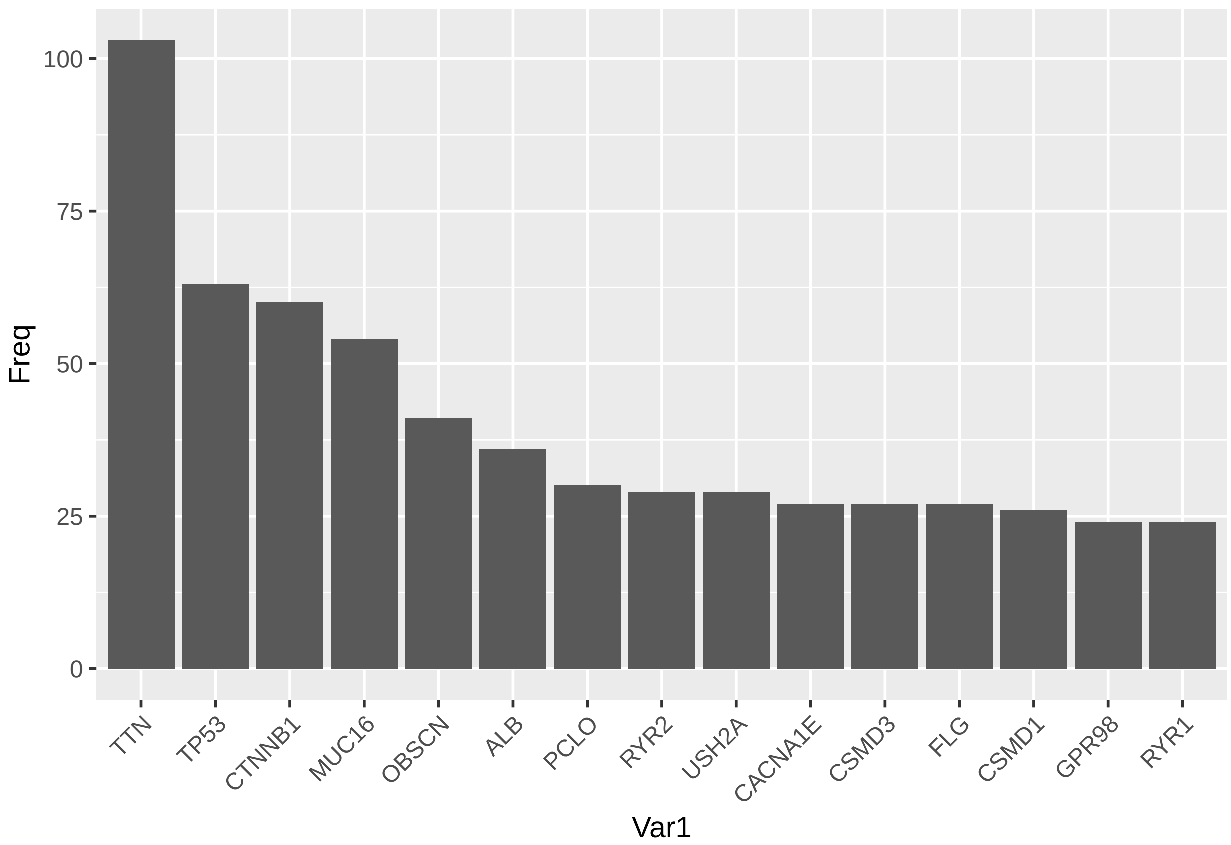
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

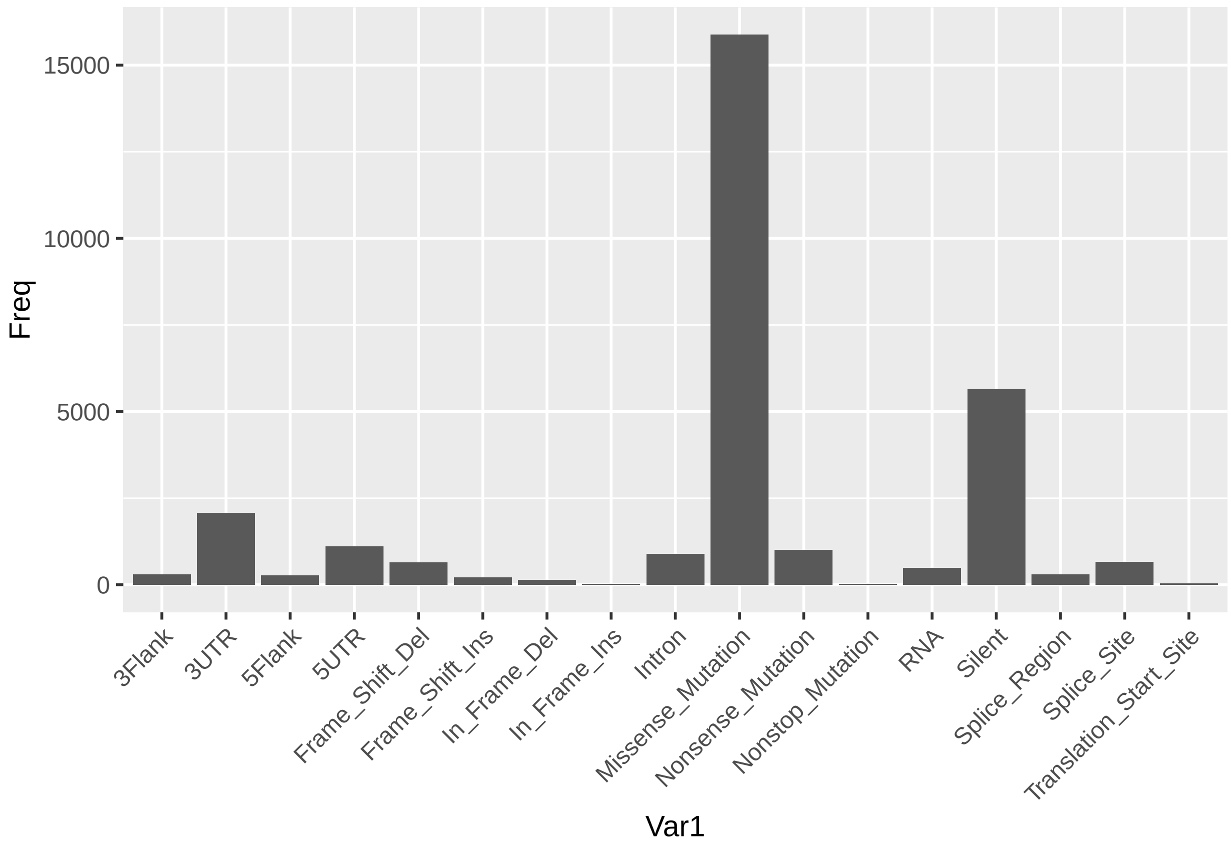
```
#create oncomat
cnv_events = unique(mutation$Variant_Classification)
oncomat = reshape2::dcast(
  data = mutation,
  formula = Hugo_Symbol ~ Tumor_Sample_Barcode,
  fun.aggregate = function(x, cnv = cnv_events) {
    x = as.character(x) # >= 2 same/distinct variant classification = Multi_Hit
    xad = x[x %in% cnv]
    xvc = x[!x %in% cnv]

    if (length(xvc) > 0) {
      xvc = ifelse(test = length(xvc) > 1,
                   yes = 'Multi_Hit',
                   no = xvc)
    }
    x = ifelse(
      test = length(xad) > 0,
      yes = paste(xad, xvc, sep = ';'),
      no = xvc
    )
    x = gsub(pattern = ';$',
             replacement = '',
             x = x)
    x = gsub(pattern = '^;',
             replacement = '',
             x = x)
    return(x)
  },
  value.var = 'Variant_Classification',
  fill = '',
  drop = FALSE
)
```

```
#mutated gene frequency
hugo <- as.data.frame(table(mutation$Hugo_Symbol))
hugo.ordered <- hugo[order(-hugo$Freq),]

ggplot(data=hugo.ordered[1:15,], aes(x=Var1, y=Freq))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 45,hjust=1))+
  scale_x_discrete(limits = hugo.ordered[1:15,]$Var1)
```



```r
var.class <- as.data.frame(table(mutation$Variant_Classification),
                           stringsAsFactors=FALSE)

#renaming some classifications to general Flank and UTR
for (i in grep("3Flank", var.class[,1])){
  var.class[i,1] <- "3Flank"
}

for (i in grep("5Flank", var.class[,1])){
  var.class[i,1] <- "5Flank"
}

for (i in grep("3UTR", var.class[,1])){
  var.class[i,1] <- "3UTR"
}

for (i in grep("5UTR", var.class[,1])){
  var.class[i,1] <- "5UTR"
}

#aggregating frequencies of each variant classification type
var.class <- aggregate(.~Var1, var.class, sum)
```

```r
#variant classifications
ggplot(data=var.class, aes(x=Var1, y=Freq))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 45,hjust=1))
```

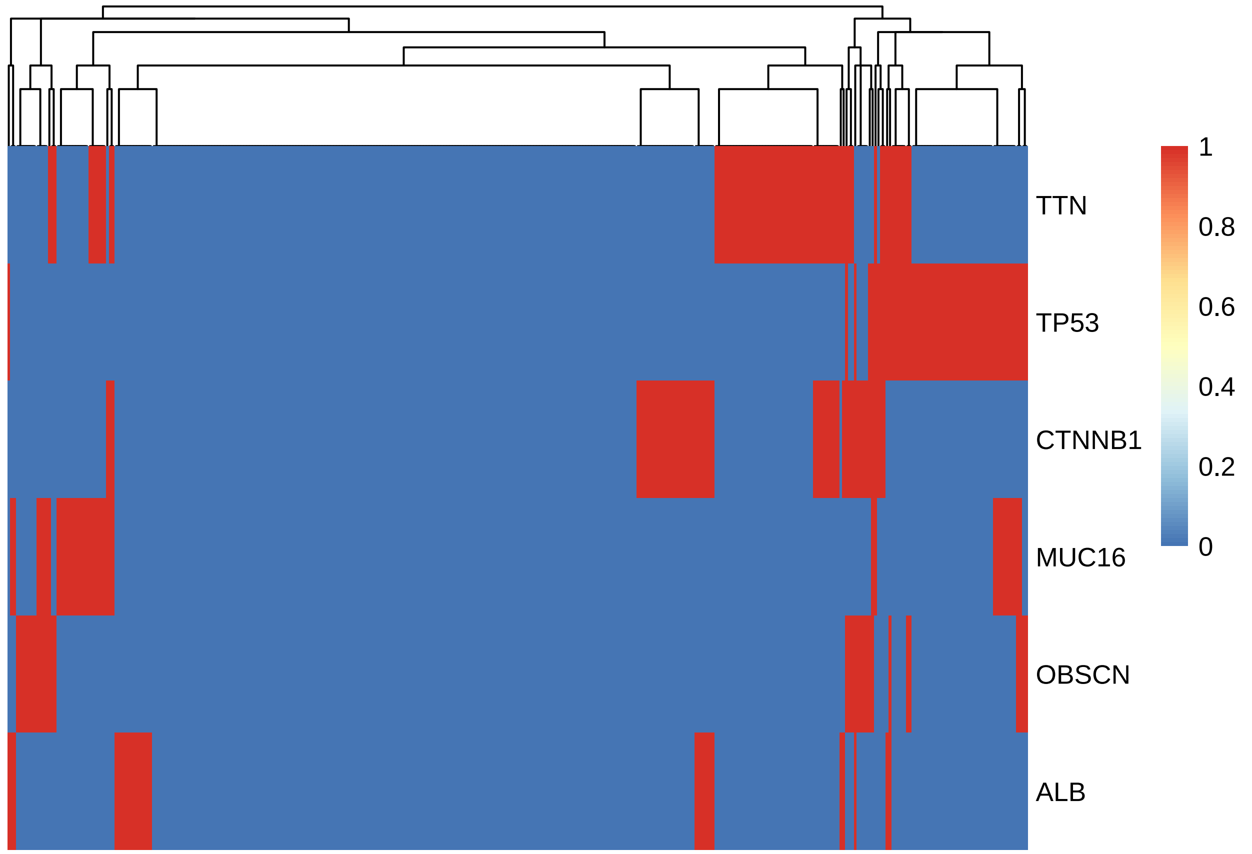


```
#Modify the row label to reflect the Hugo symbol (gene name)
rownames(oncomat) = oncomat$Hugo_Symbol
oncomat <- oncomat[,-1]

#Reorder the rows according to the occurrence of heavily mutated genes
oncomat.ordered <- oncomat[order(-hugo$Freq),]
```

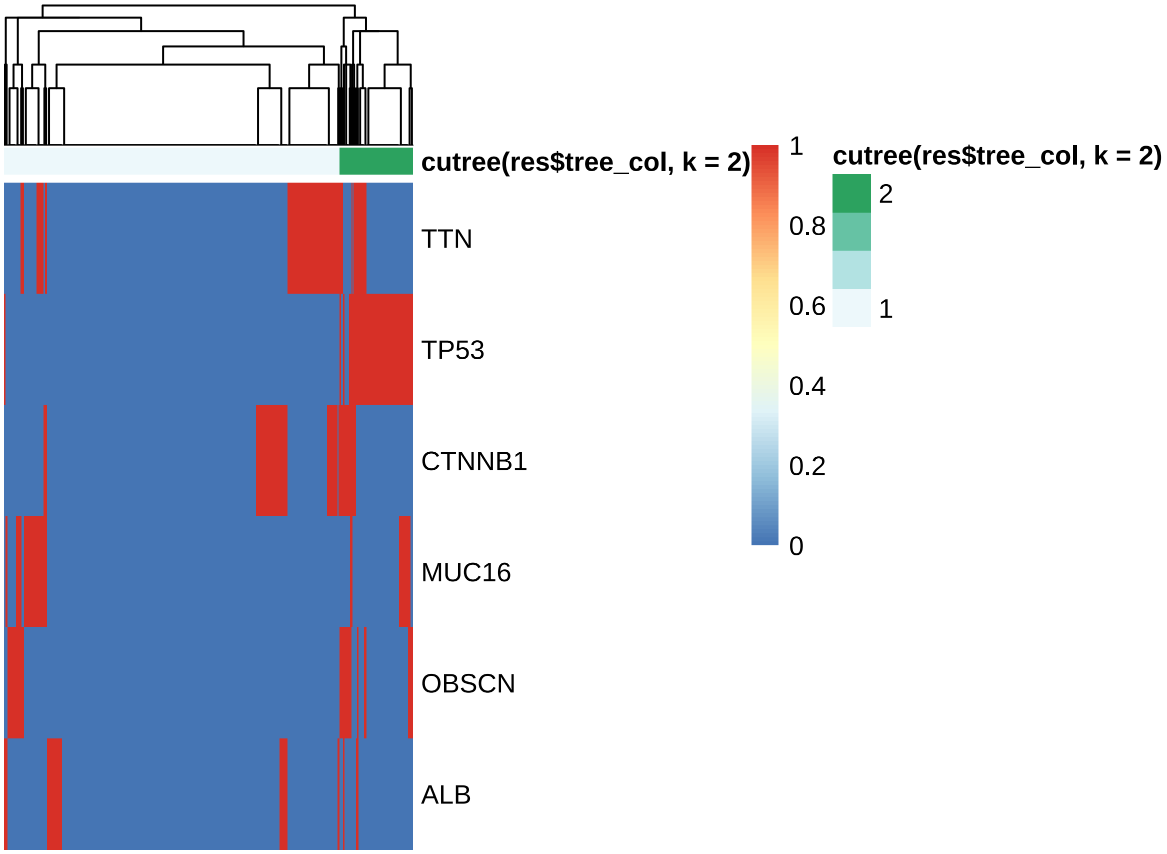
```
#transform the matrix into a binary matrix
mat <- oncomat.ordered
mat[mat=="Silent"]=0 #remove silent
mat[mat!=""]=1 #remaining mutations
mat[mat==""]=0

mat <- apply(mat, 2 ,as.numeric)
mat <- as.matrix(mat)
rownames(mat)  <-  row.names(oncomat.ordered)

library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.3.2
```

```r
#at least 10% of patients have the mutated genes
cutoff <- length(clinical$PATIENT_ID) * 0.1

#include mutations common in at >10% patients
reduce.mat <- mat[1:sum(hugo.ordered[,2]>cutoff),]
res <- pheatmap(reduce.mat,
         cluster_rows = F,
         show_colnames=FALSE)
```



```r
#with cluster label
res <- pheatmap(reduce.mat,
         cluster_rows = F,
         show_colnames=FALSE,
         annotation_col = as.data.frame(cutree(res$tree_col, k = 2)))
```



```r
#two clusters, heatmap shows either 0 or 1
cluster <-  as.data.frame(cutree(res$tree_col, k = 2))
```

Survival Analysis

```r
library("TCGAbiolinks")
library("survival")
library("survminer")
```

```
## Loading required package: ggpubr
```

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

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

```r
library("SummarizedExperiment")
```

```
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
```

```
## Warning: package 'matrixStats' was built under R version 4.3.2
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
##
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from ’package:base’:
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: ’S4Vectors’
```

```
## The following objects are masked from ’package:dplyr’:
##
##     first, rename
```

```
## The following object is masked from ’package:utils’:
##
##     findMatches
```

```
## The following objects are masked from ’package:base’:
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: ’IRanges’
```

```
## The following objects are masked from ’package:dplyr’:
##
##     collapse, desc, slice
```

```
## The following object is masked from ’package:grDevices’:
##
##     windows
```

```
## Loading required package: GenomeInfoDb
```

```
## Warning: package ’GenomeInfoDb’ was built under R version 4.3.2
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     ’browseVignettes()’. To cite Bioconductor, see
##     ’citation("Biobase")’, and for packages ’citation("pkgname")’.
```

```
##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```r
# #looking at some clinical features

# table(clinical$OS_STATUS)
# table(clinical$AJCC_PATHOLOGIC_TUMOR_STAGE) #mostly stage I
# table(clinical$DSS_STATUS) #most dead/alive tumor free
# table(clinical$RACE) #mostly asian/white
# table(clinical$SEX) #double male samples
```

```r
#convert to months (match DSS/OS scale)
clinical$DAYS_LAST_FOLLOWUP <- (clinical$DAYS_LAST_FOLLOWUP / 365) * 12
```

```r
clinical$subgroup <- cluster[,1]

#looking at all patients
clin_df = clinical[,
                    c("PATIENT_ID",
                      "OS_STATUS",
                      "OS_MONTHS",
                      "DSS_STATUS",
                      "DSS_MONTHS",
                      "DAYS_LAST_FOLLOWUP",
                      "SEX",
                      "subgroup")]
```

```r
# create a new boolean variable that has TRUE for dead patients (with tumor)
# and FALSE for patients without tumor
clin_df$deceased_DSS = clin_df$DSS_STATUS == "1:DEAD WITH TUMOR"

# create an "overall survival" variable that is equal to days_to_death
# for dead patients, and to days_to_last_follow_up for patients who
# are still alive
clin_df$DSS_survival = ifelse(clin_df$deceased_DSS,
                                   clin_df$DSS_MONTHS,
                                   clin_df$DAYS_LAST_FOLLOWUP)

#overall survival - can compare with DSS
clin_df$deceased_OS = clin_df$OS_STATUS == "1:DECEASED"

clin_df$OS_survival = ifelse(clin_df$deceased_OS,
                                   clin_df$OS_MONTHS,
                                   clin_df$DAYS_LAST_FOLLOWUP)
```

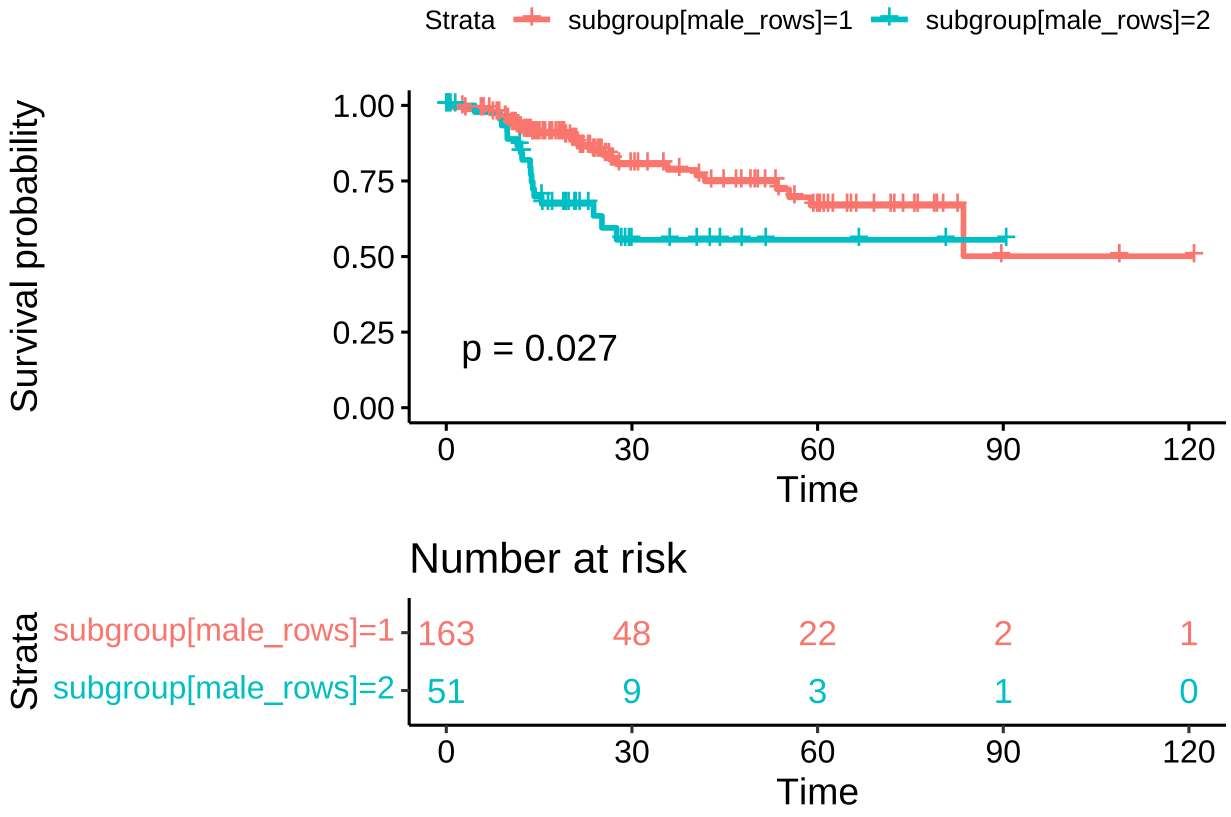
```r
#try male group1 vs male group 2
#one sex subgroup vs other subgroup opp sex
sub1_rows <- which(clin_df$subgroup==1)
female_sub1_rows <- sub1_rows[clin_df$SEX[sub1_rows]=="Female"]
male_sub1_rows <- sub1_rows[clin_df$SEX[sub1_rows]=="Male"]

sub2_rows <- which(clin_df$subgroup==2)
female_sub2_rows <- sub2_rows[clin_df$SEX[sub2_rows]=="Female"]
male_sub2_rows <- sub2_rows[clin_df$SEX[sub2_rows]=="Male"]

male_rows <- sort(c(male_sub1_rows, male_sub2_rows))
female_rows <- sort(c(female_sub1_rows, female_sub2_rows))
fem1_male2 <- sort(c(female_sub1_rows, male_sub2_rows))
fem2_male1 <- sort(c(female_sub2_rows, male_sub1_rows))
```

Kaplan-Meier Curves

```r
#DSS
#male 1 vs male 2
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup[male_rows]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup[male_rows]
```

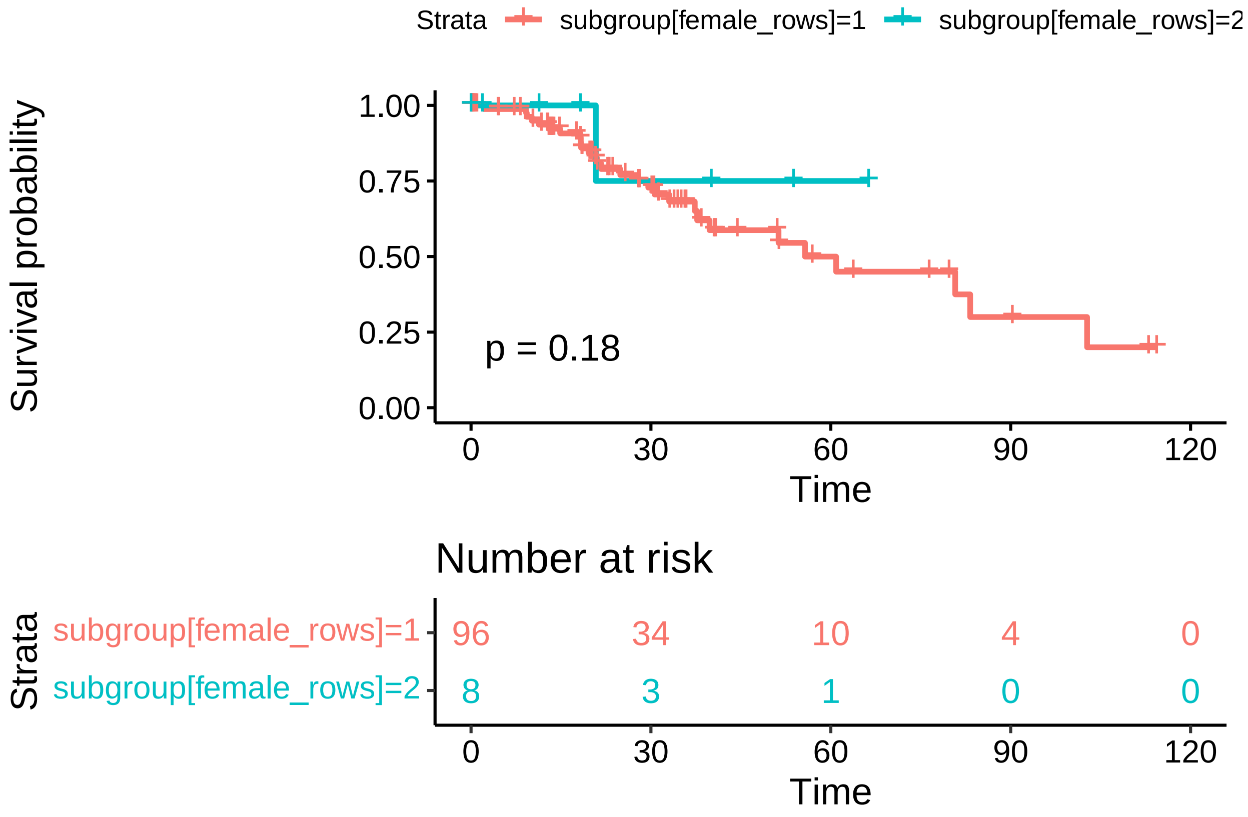
```r
fit = survfit(Surv(DSS_survival[male_rows], deceased_DSS[male_rows]) ~ subgroup[male_rows], data=clin_d

ggsurvplot(fit, data=clin_df[male_rows,], pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```r
#female 1 vs female 2
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup[female_rows]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup[female_rows]
```

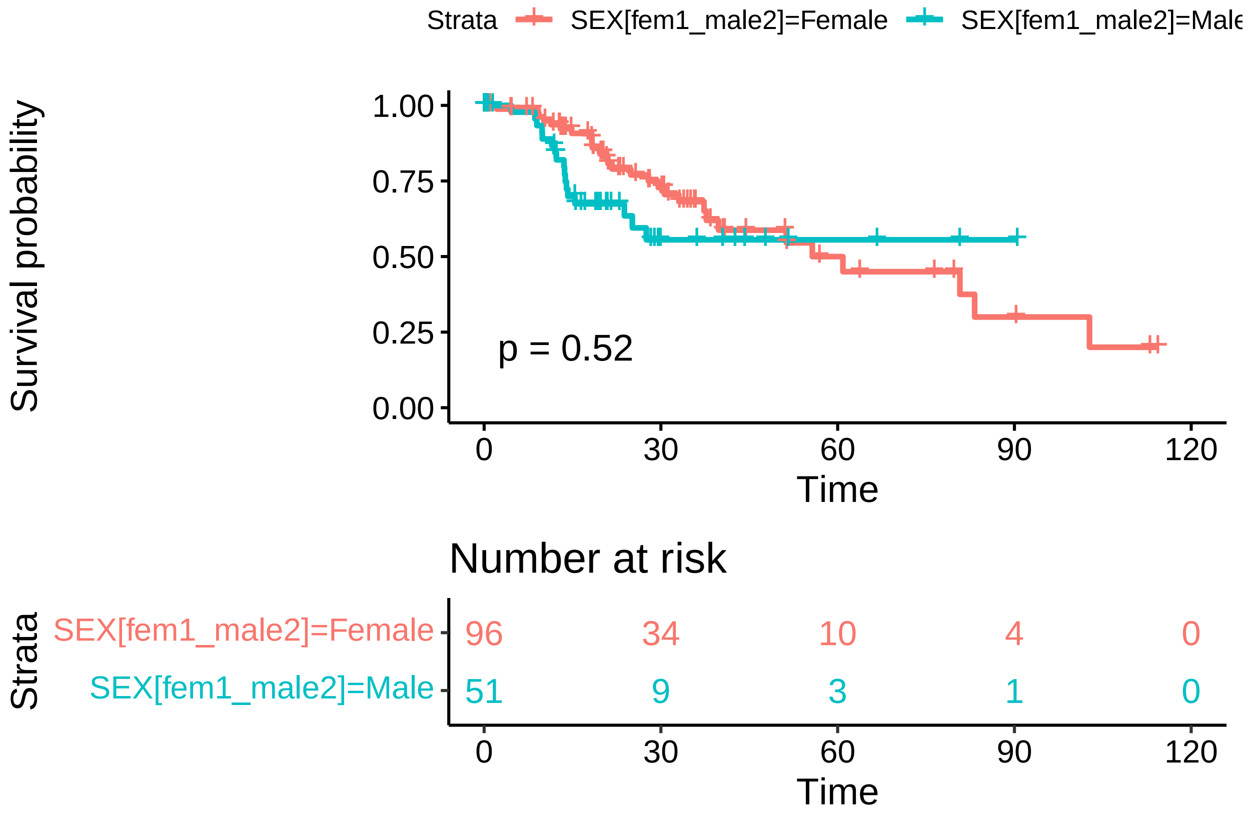
```r
fit = survfit(Surv(DSS_survival[female_rows], deceased_DSS[female_rows]) ~ subgroup[female_rows], data=

ggsurvplot(fit, data=clin_df[female_rows,], pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```
#female 1 vs male 2
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[fem1_male2]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[fem1_male2]
```

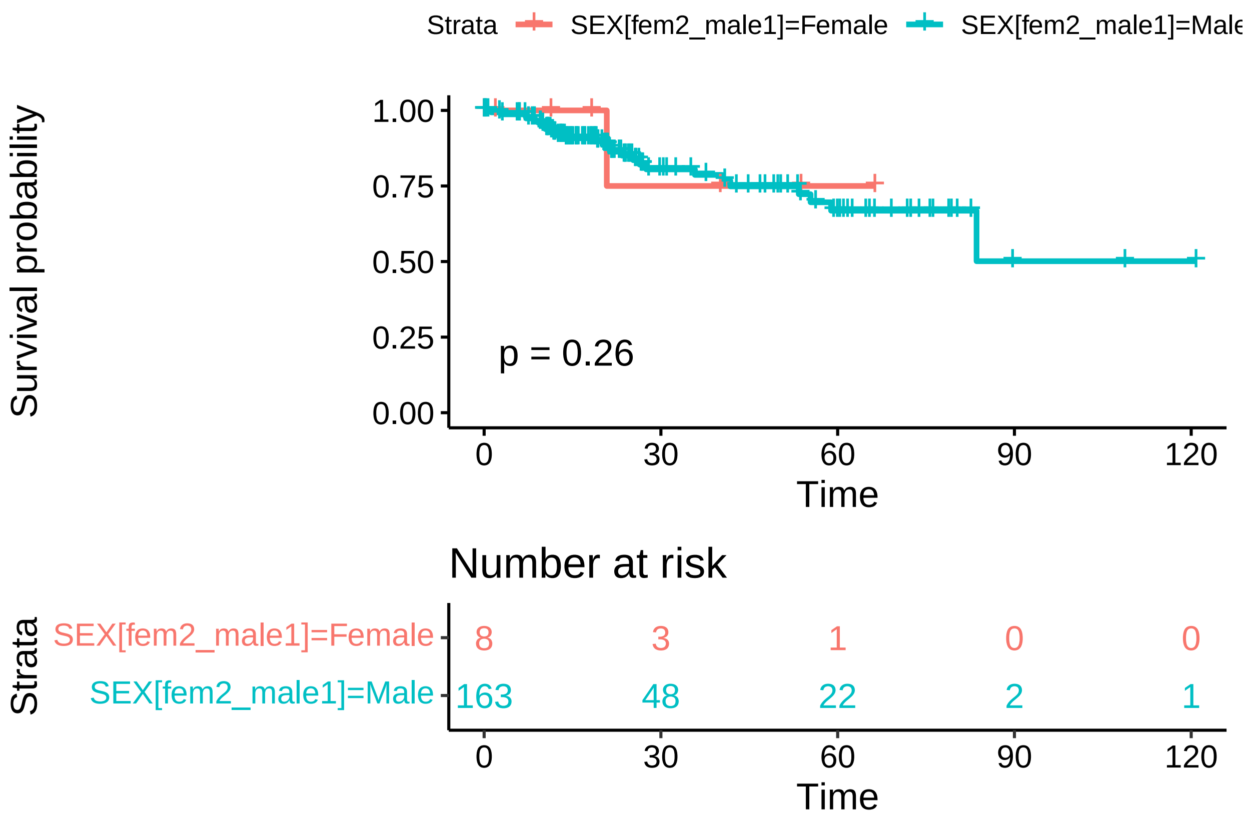
```
fit = survfit(Surv(DSS_survival[fem1_male2], deceased_DSS[fem1_male2]) ~ SEX[fem1_male2], data=clin_df)

ggsurvplot(fit, data=clin_df[fem1_male2,], pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```
#female 2 vs male 1
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[fem2_male1]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[fem2_male1]
```

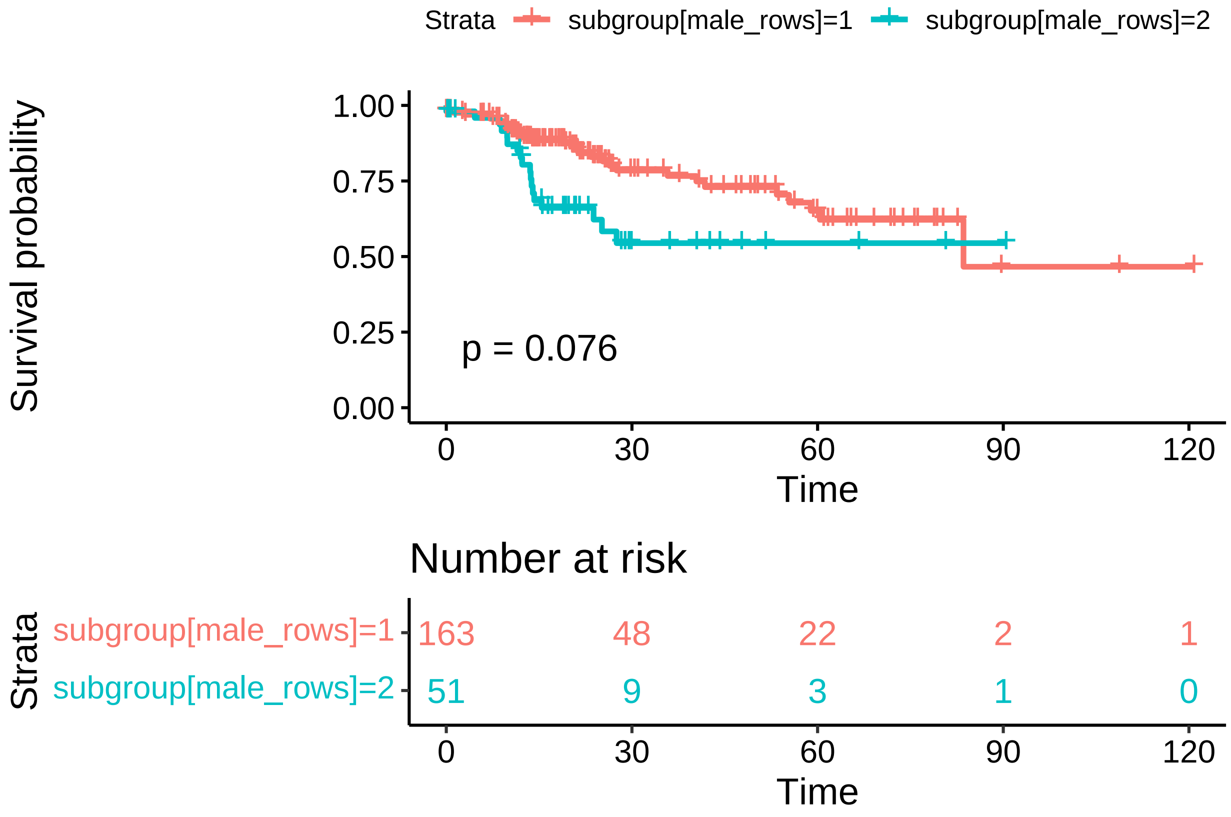
```
fit = survfit(Surv(DSS_survival[fem2_male1], deceased_DSS[fem2_male1]) ~ SEX[fem2_male1], data=clin_df)

ggsurvplot(fit, data=clin_df[fem2_male1,], pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```
#OS -> just to see if large differences from DSS
#male 1 vs male 2
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup[male_rows]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup[male_rows]
```

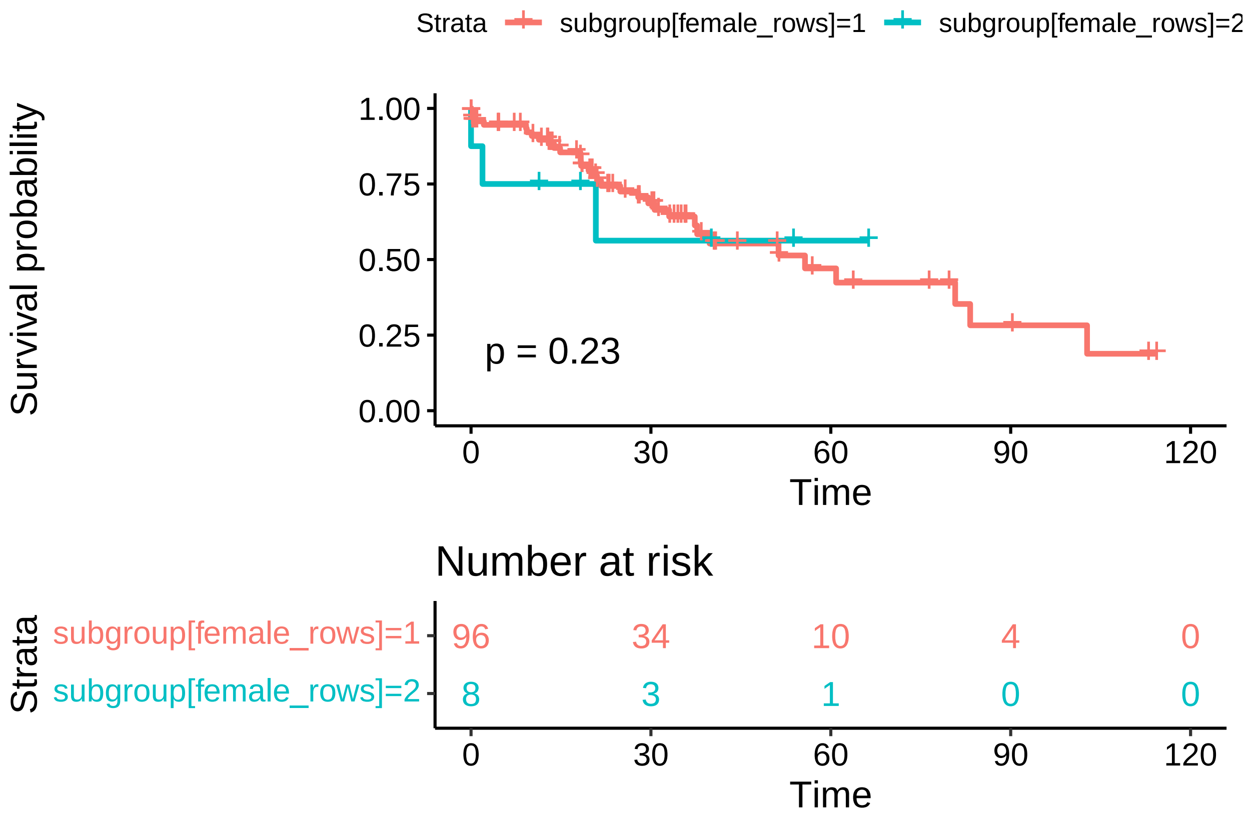
```
fit = survfit(Surv(DSS_survival[male_rows],
                   deceased_OS[male_rows]) ~ subgroup[male_rows], data=clin_df)

ggsurvplot(fit, data=clin_df[male_rows,], pval=T,
           risk.table=T, risk.table.col="strata", risk.table.height=0.35)
```



```
#female 1 vs female 2
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup[female_rows]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup[female_rows]
```

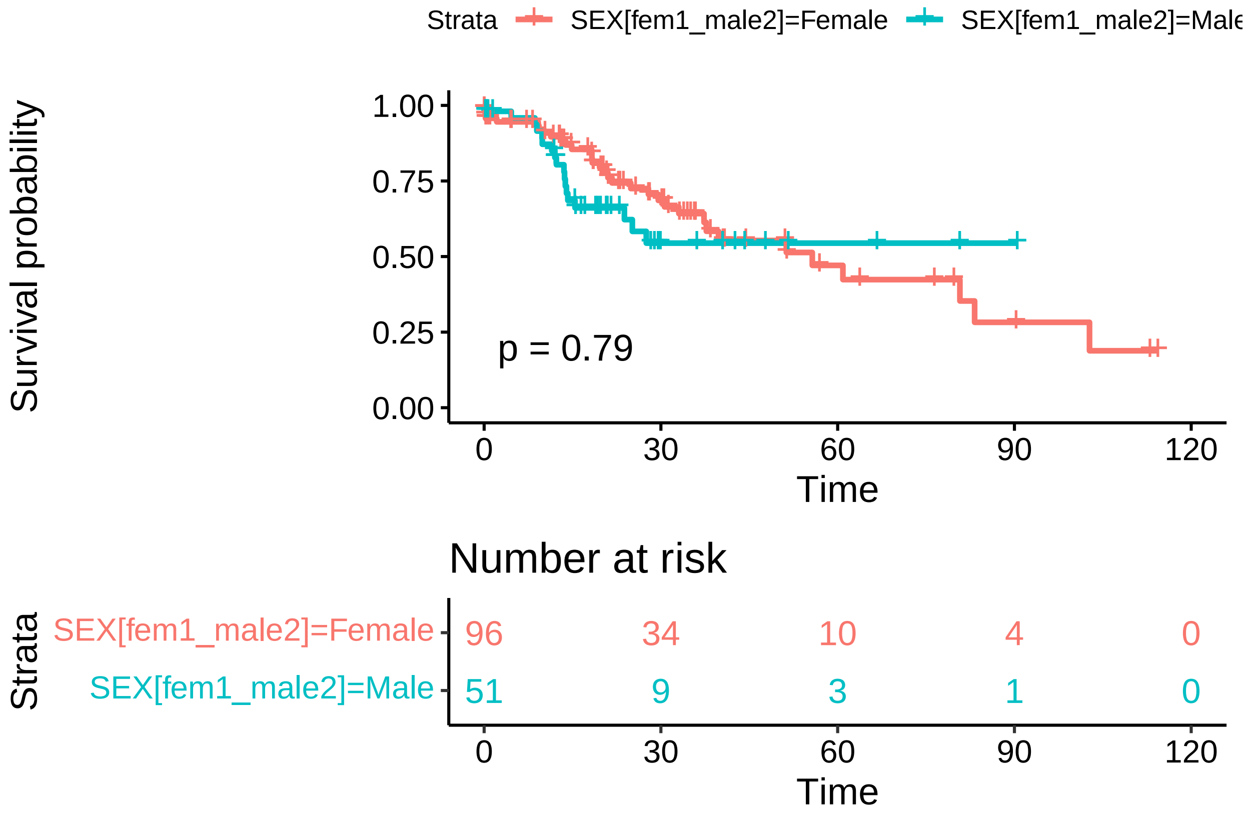
```
fit = survfit(Surv(DSS_survival[female_rows],
                   deceased_OS[female_rows]) ~ subgroup[female_rows], data=clin_df)

ggsurvplot(fit, data=clin_df[female_rows,], pval=T,
           risk.table=T, risk.table.col="strata", risk.table.height=0.35)
```



```
#female 1 vs male 2
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[fem1_male2]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[fem1_male2]
```

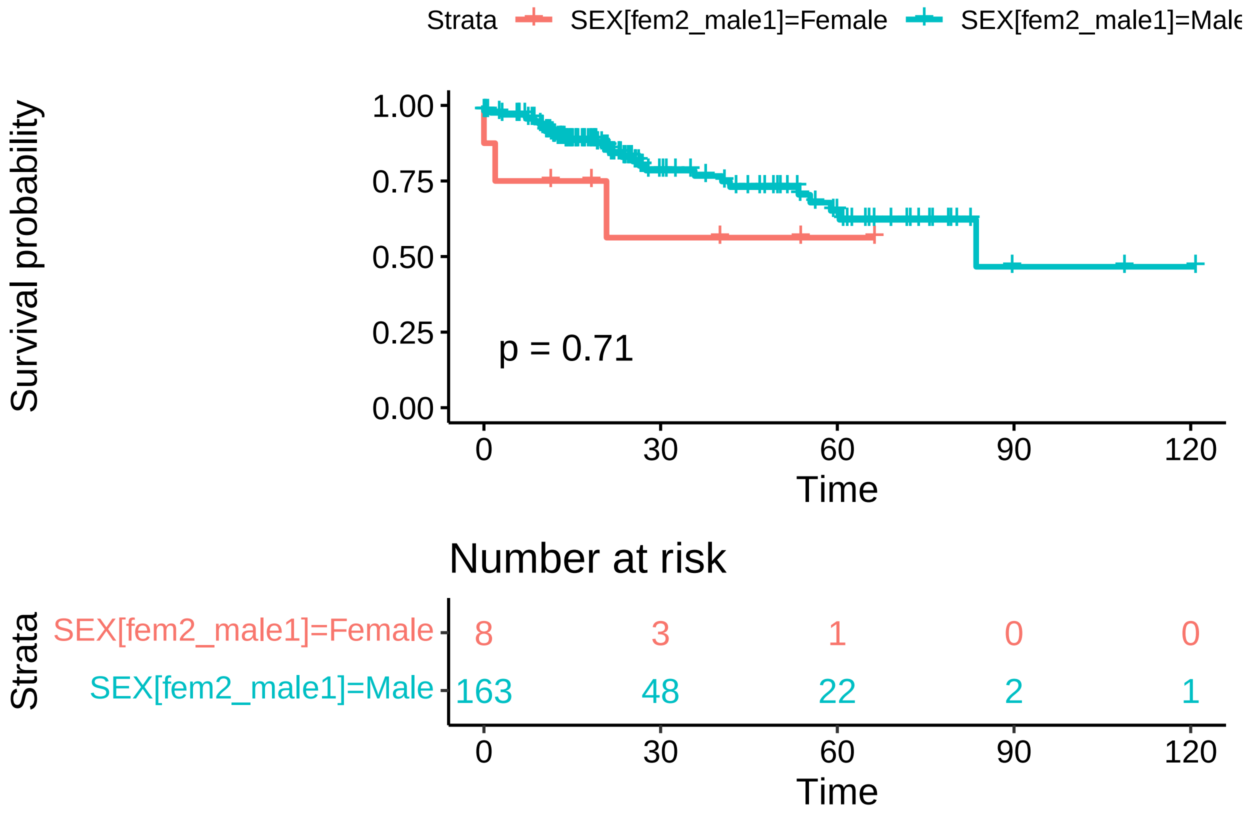
```
fit = survfit(Surv(DSS_survival[fem1_male2],
                   deceased_OS[fem1_male2]) ~ SEX[fem1_male2], data=clin_df)

ggsurvplot(fit, data=clin_df[fem1_male2,], pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```r
#female 2 vs male 1
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[fem2_male1]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[fem2_male1]
```

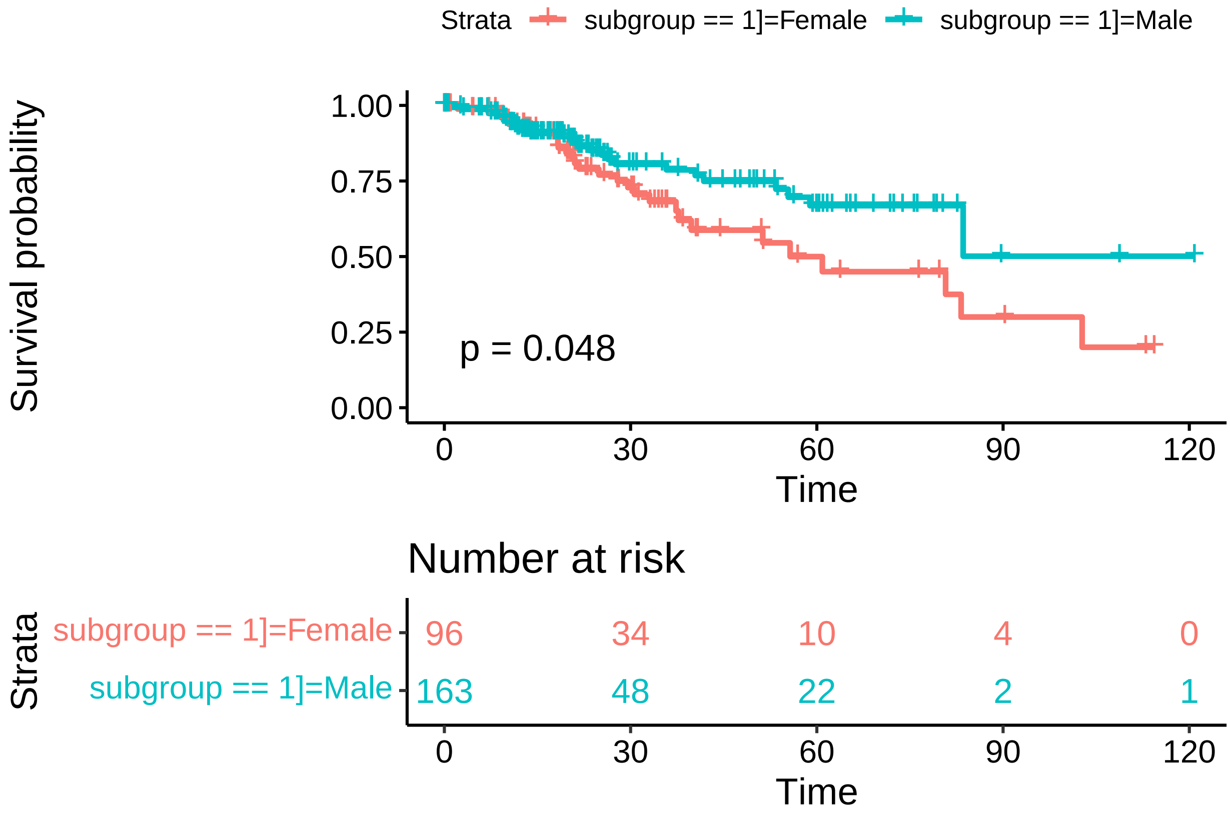
```r
fit = survfit(Surv(DSS_survival[fem2_male1],
                   deceased_OS[fem2_male1]) ~ SEX[fem2_male1], data=clin_df)

ggsurvplot(fit, data=clin_df[fem2_male1,], pval=T,
           risk.table=T, risk.table.col="strata", risk.table.height=0.35)
```



```
#DSS
#subgroup 1 by sex
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[clin_df$subgroup==1]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[clin_df$subgroup ==
##     1]
```

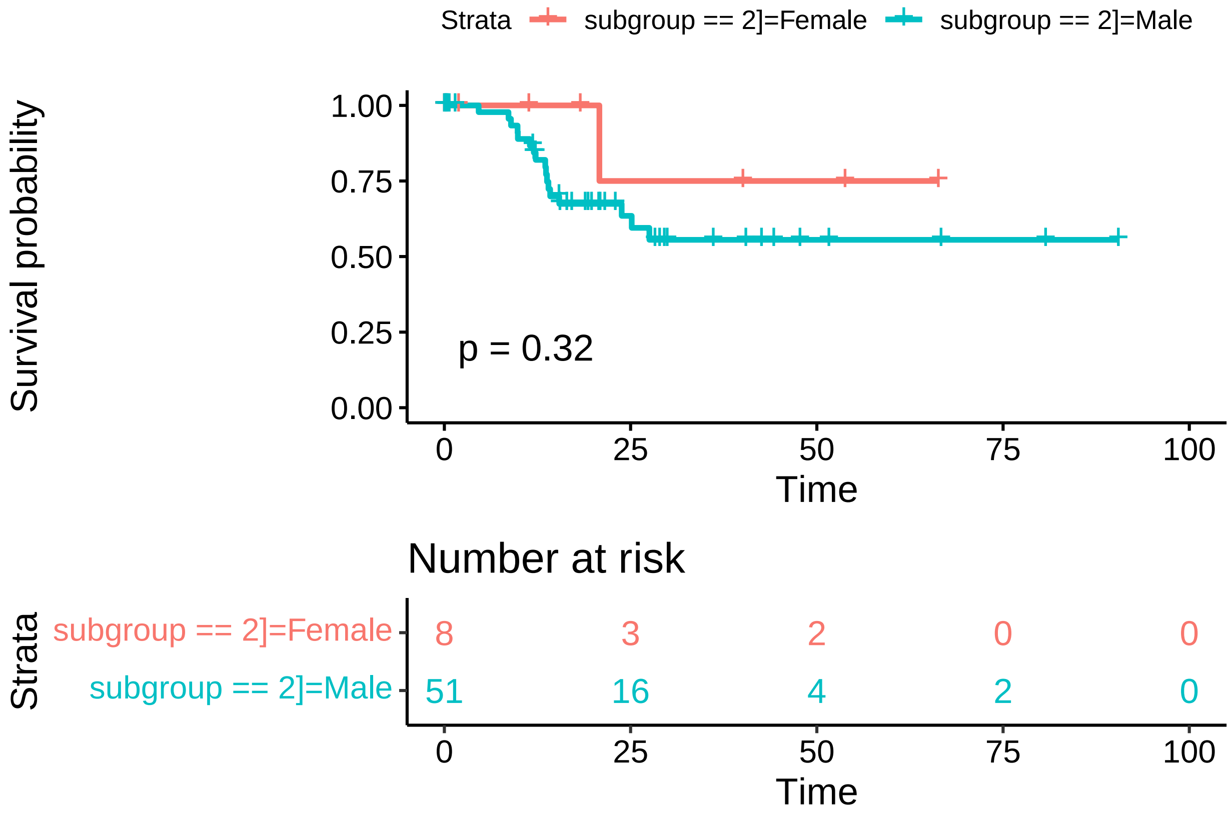
```
fit = survfit(Surv(DSS_survival[clin_df$subgroup==1],
                   deceased_DSS[clin_df$subgroup==1]) ~ SEX[clin_df$subgroup==1], data=clin_df)

ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```
#subgroup 2 by sex
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[clin_df$subgroup==2]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$SEX[clin_df$subgroup ==
##     2]
```

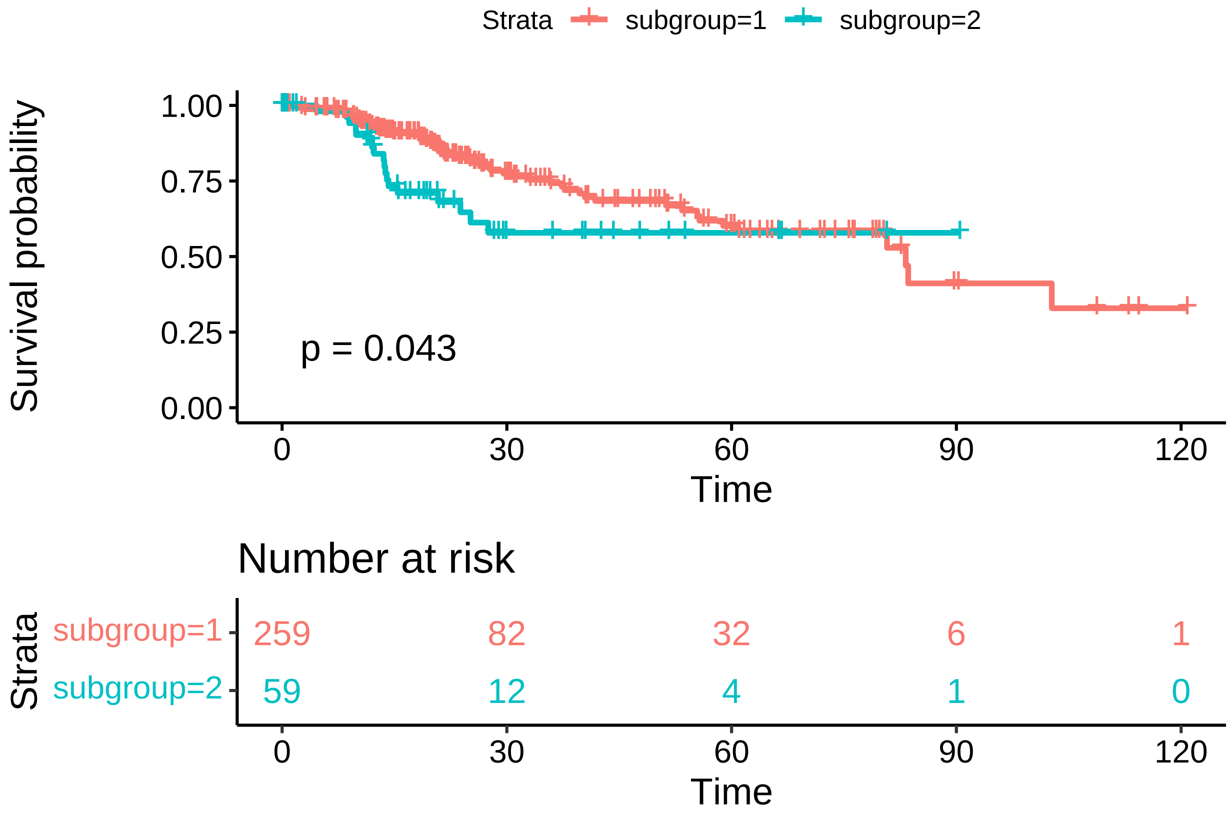
```
fit = survfit(Surv(DSS_survival[clin_df$subgroup==2],
                   deceased_DSS[clin_df$subgroup==2]) ~ SEX[clin_df$subgroup==2], data=clin_df)

ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```r
#between overall subgroups
Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_DSS) ~ clin_df$subgroup
```

```r
fit = survfit(Surv(DSS_survival, deceased_DSS) ~ subgroup, data=clin_df)

# subtype vs survival
ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```

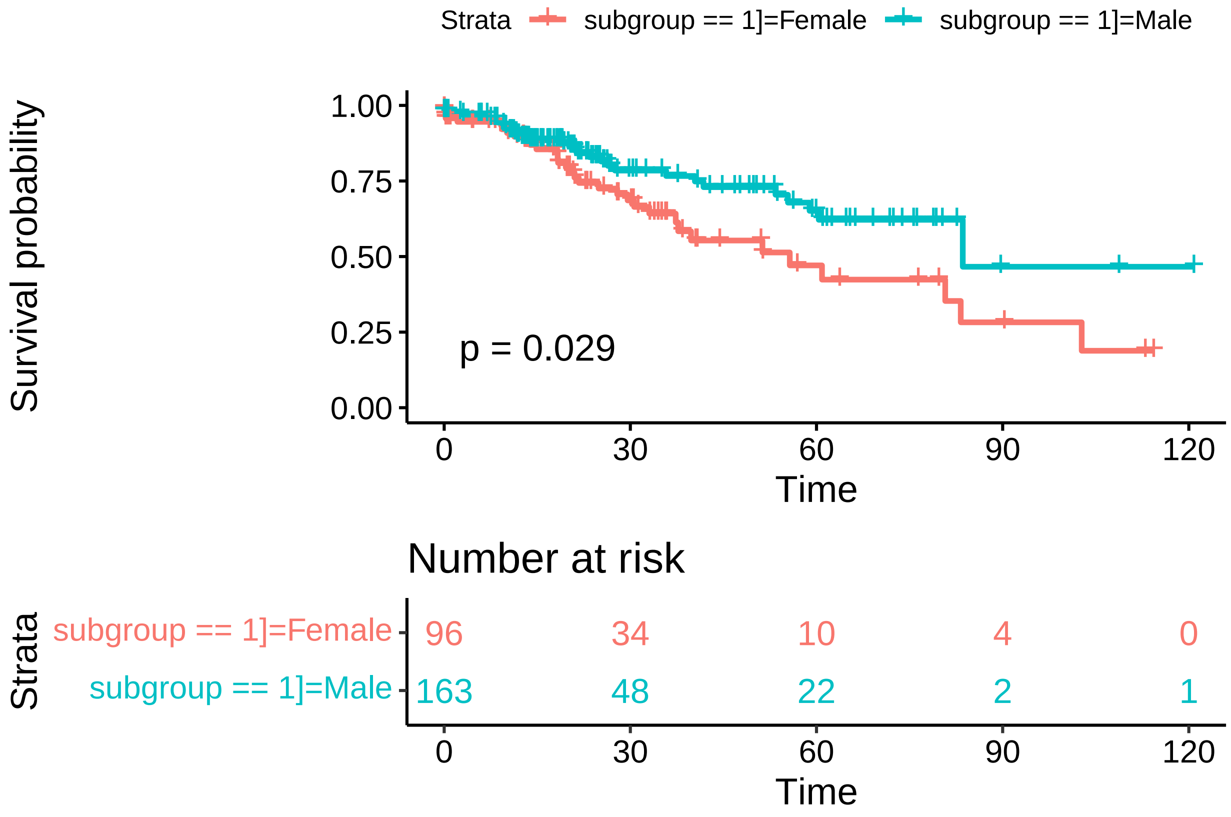


```
group1 <- rownames(cluster)[cluster[,1]==1] #list of patients in group1
group2 <- rownames(cluster)[cluster[,1]==2] #list of patients in group2
```

```
#OS

#subgroup 1 by sex
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[clin_df$subgroup==1]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[clin_df$subgroup ==
##     1]
```

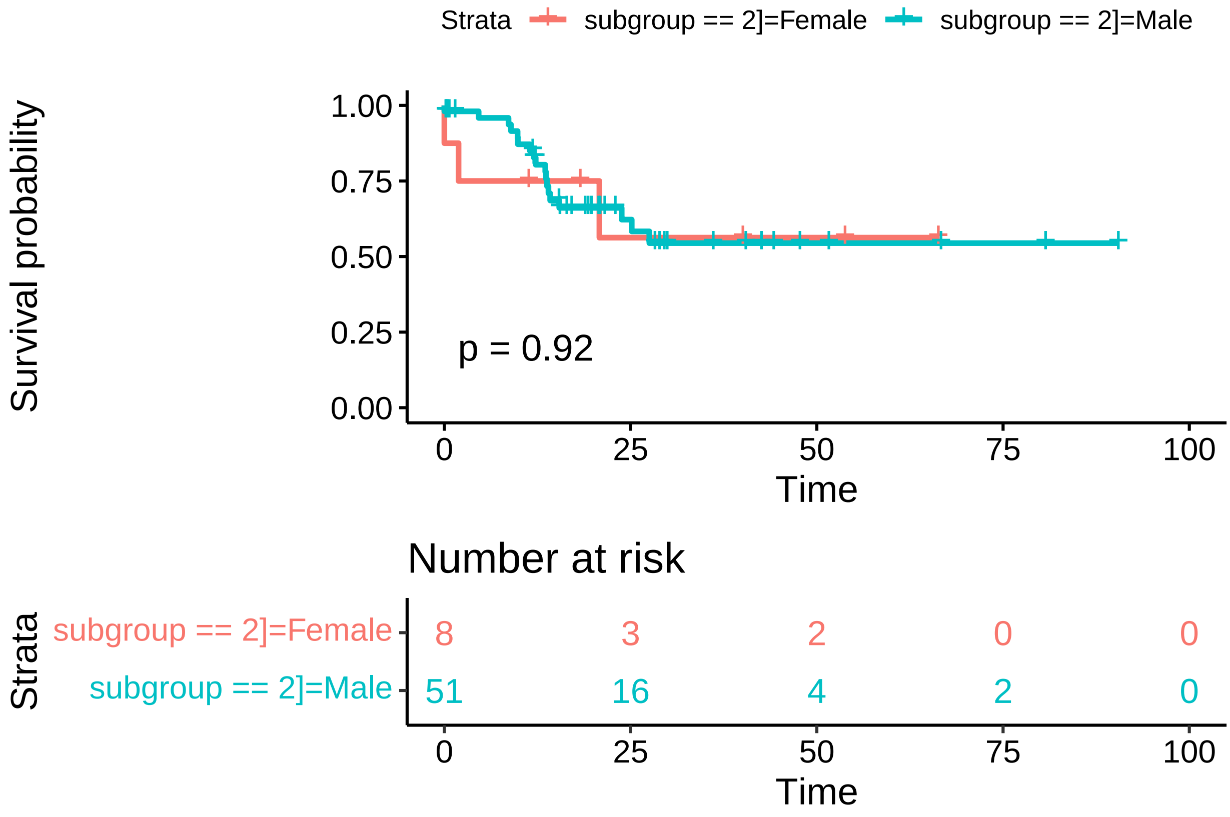
```
fit = survfit(Surv(DSS_survival[clin_df$subgroup==1],
                   deceased_OS[clin_df$subgroup==1]) ~ SEX[clin_df$subgroup==1], data=clin_df)

ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```
#subgroup 2 by sex
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[clin_df$subgroup==2]
```

```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$SEX[clin_df$subgroup ==
##     2]
```

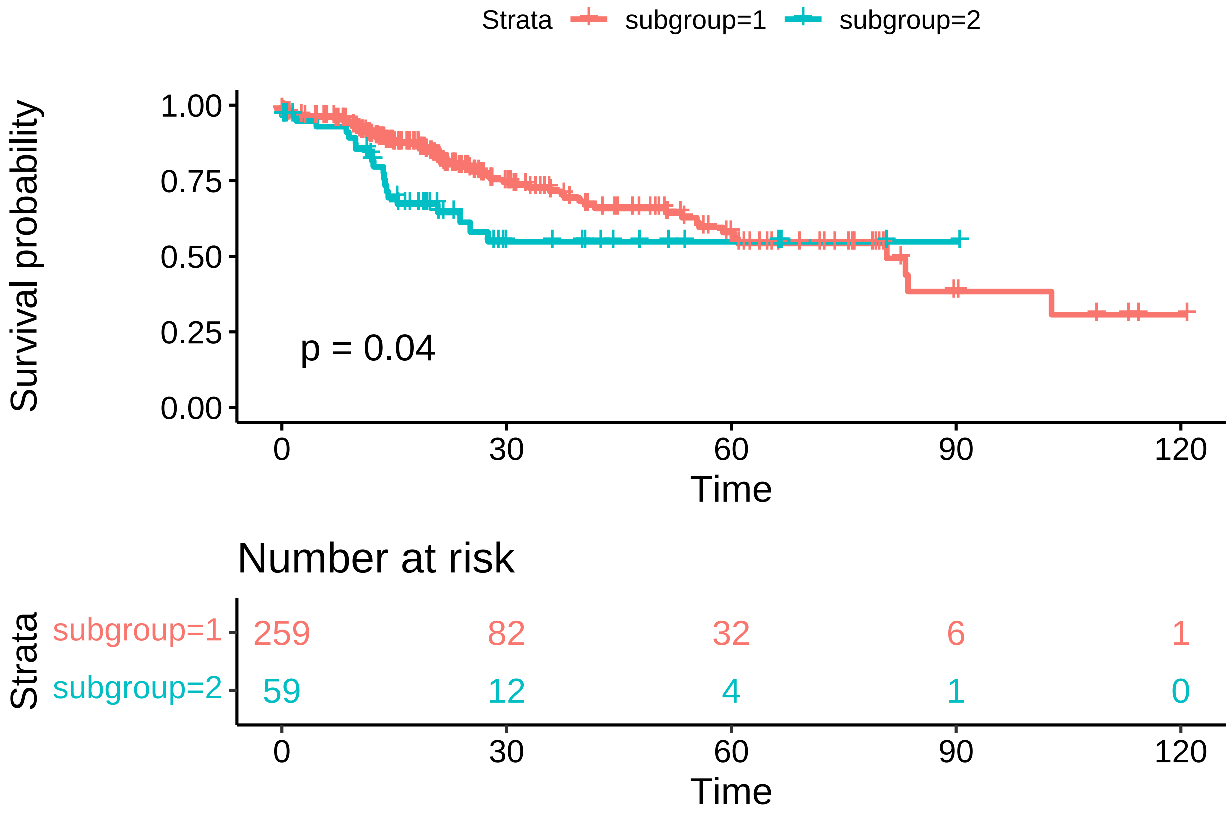
```
fit = survfit(Surv(DSS_survival[clin_df$subgroup==2],
                   deceased_OS[clin_df$subgroup==2]) ~ SEX[clin_df$subgroup==2], data=clin_df)

ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



```r
#between overall subgroups
Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup
```

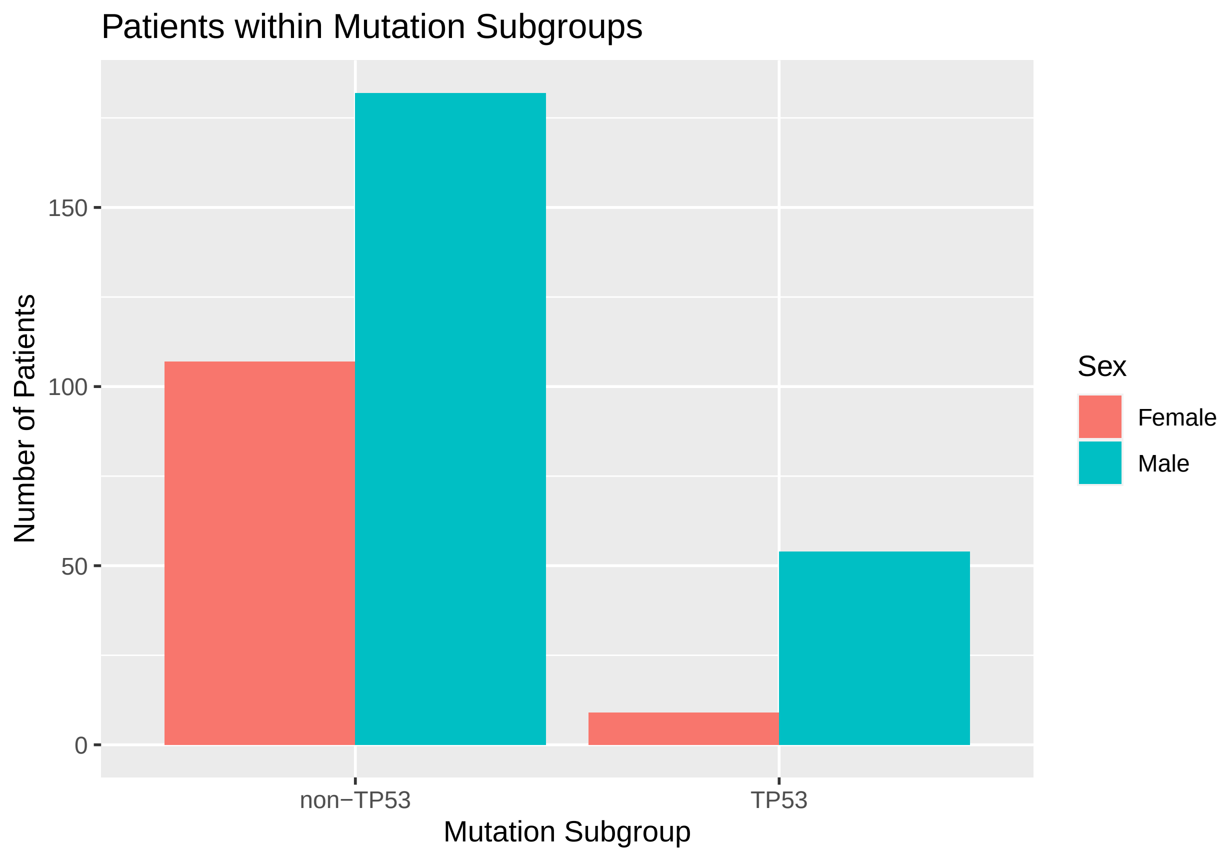
```
## Surv(clin_df$DSS_survival, clin_df$deceased_OS) ~ clin_df$subgroup
```

```r
fit = survfit(Surv(DSS_survival, deceased_OS) ~ subgroup, data=clin_df)

# subtype vs survival
ggsurvplot(fit, data=clin_df, pval=T, risk.table=T,
           risk.table.col="strata", risk.table.height=0.35)
```



#Male/Female within clusters

```
fem1 <- c(length(female_sub1_rows), "non-TP53", "Female")
fem2 <- c(length(female_sub2_rows), "TP53", "Female")
male1 <- c(length(male_sub1_rows), "non-TP53", "Male")
male2 <- c(length(male_sub2_rows), "TP53", "Male")

sex.data <- rbind(fem1, fem2, male1, male2)
colnames(sex.data) <- c("count", "subgroup", "sex")
# Use position=position_dodge()
ggplot(as.data.frame(sex.data),
       aes(x=subgroup, y = as.numeric(count), fill = sex)) +
      geom_bar(stat = "identity", position = "dodge") +
      labs(title="Patients within Mutation Subgroups",
           x="Mutation Subgroup", y="Number of Patients", fill = "Sex")
```

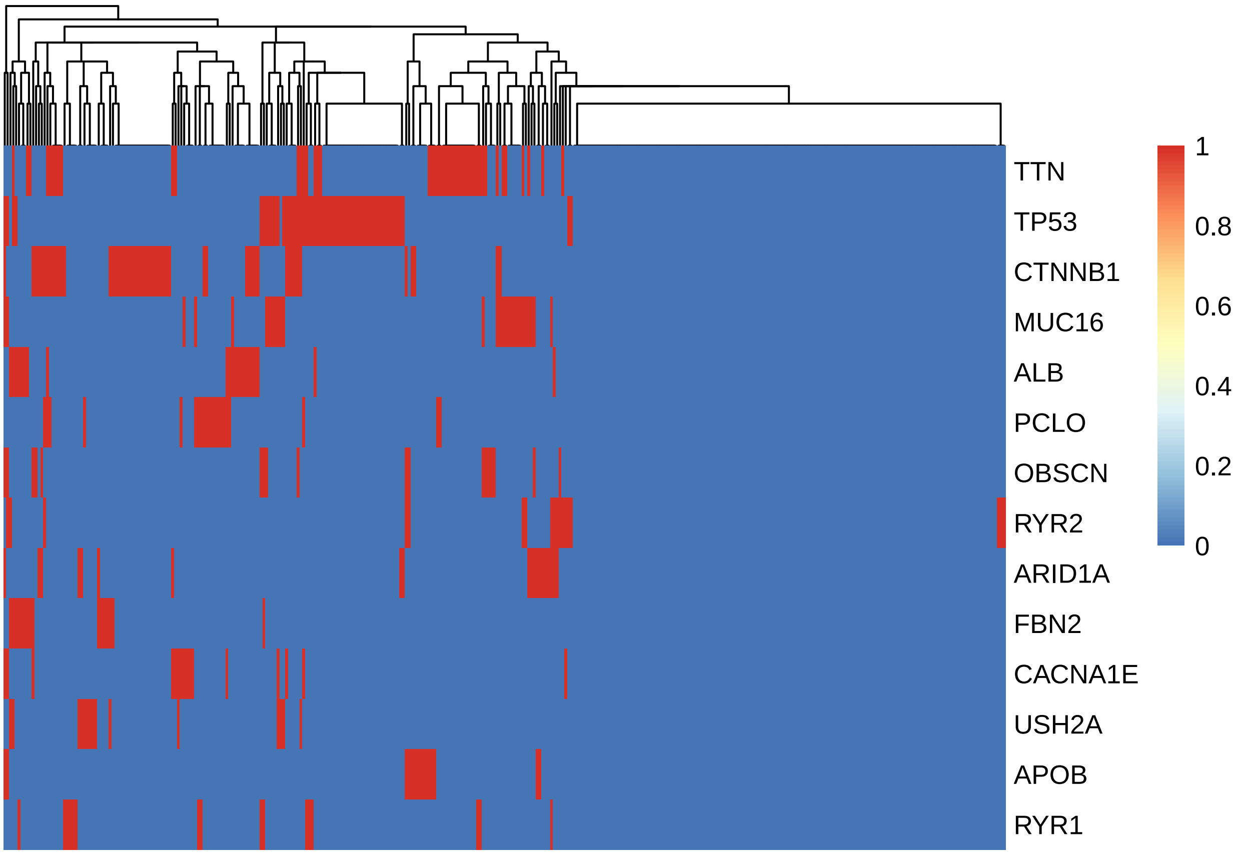


Heatmap filtered to only high/moderate consequence mutations

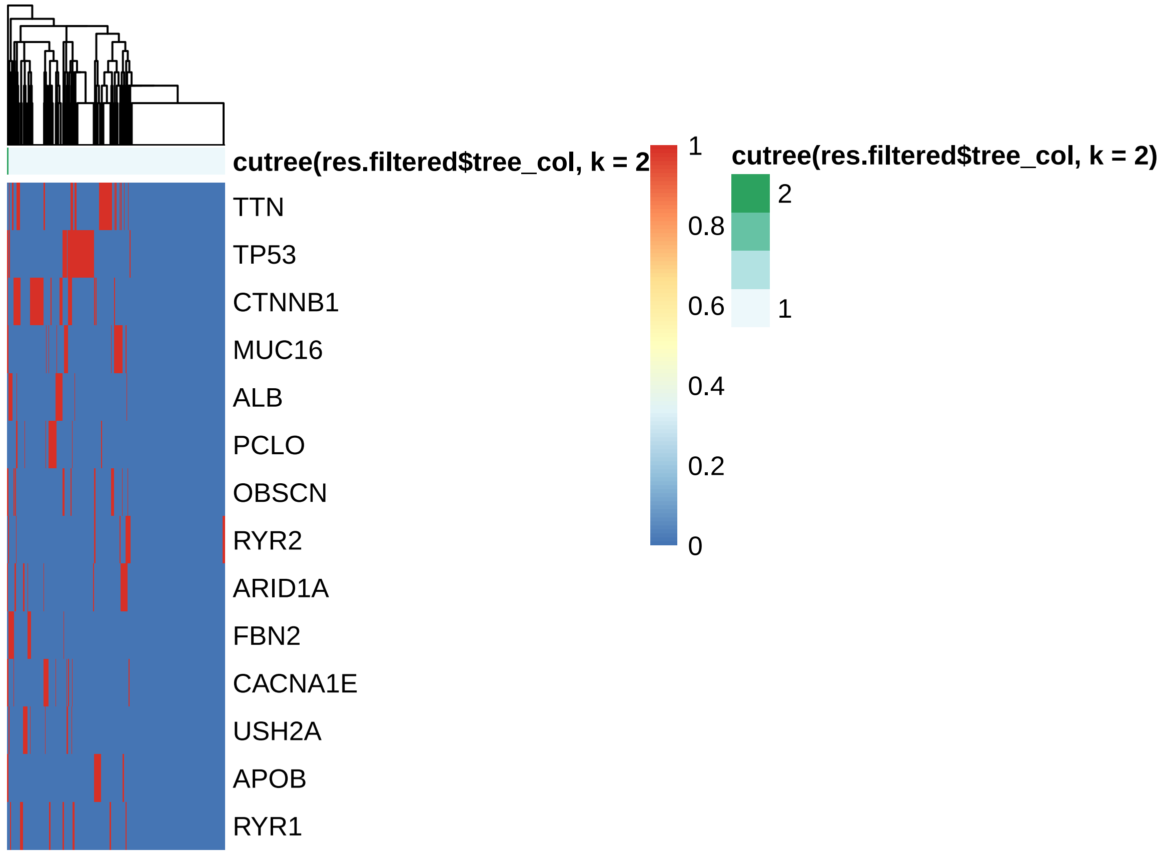
```
#create oncomat with variant consequence
cnv_events = unique(mutation$Consequence)
oncomat = reshape2::dcast(
  data = mutation,
  formula = Hugo_Symbol ~ Tumor_Sample_Barcode,
  fun.aggregate = function(x, cnv = cnv_events) {
    x = as.character(x) # >= 2 same/distinct variant classification = Multi_Hit
    xad = x[x %in% cnv]
    xvc = x[!x %in% cnv]

    if (length(xvc) > 0) {
      xvc = ifelse(test = length(xvc) > 1,
                   yes = 'Multi_Hit',
                   no = xvc)
    }

    x = ifelse(
      test = length(xad) > 0,
      yes = paste(xad, xvc, sep = ';'),
      no = xvc
    )
    x = gsub(pattern = ';$',
             replacement = '',
             x = x)
    x = gsub(pattern = '^;',
             replacement = '',
             x = x)
    return(x)
  },
  value.var = 'Consequence',
  fill = '',
  drop = FALSE
)

#high/moderate consequence based on snpeff documentation
high_moderate <- "chromosome_number_variation|exon_loss|frameshift_variant|rare_amino_acid_variant|splic

library(dplyr)
# Function to replace values that match the pattern in a column
replace_if_pattern_exists <- function(column) {
  ifelse(grepl(high_moderate, column),
         1, #replacement value
         column)
}

# Apply the function to each column using mutate_at() from dplyr
oncomat <- oncomat %>%
  mutate_at(vars(everything()), ~replace_if_pattern_exists(.))


mat.filtered <- oncomat[,-1]

mat.filtered[mat.filtered!=1]=0

mat.filtered <- apply(mat.filtered, 2 ,as.numeric)
mat.filtered <- as.matrix(mat.filtered)
rownames(mat.filtered)  <-  oncomat[,1]
```

```
library(pheatmap)
cutoff <- length(clinical$PATIENT_ID) * 0.05 #5% of total patients

#filter mutation data for only high/moderate impact,
#can then make filtered hugo freq list to see
#highest occuring high/mod impact variants

mutation.filtered <- mutation %>% filter(grepl(high_moderate, Consequence))
hugo.filtered <- as.data.frame(table(mutation.filtered$Hugo_Symbol))
hugo.ordered.filtered <- hugo.filtered[order(-hugo.filtered$Freq),]

#names of top mutation rows
sigRows <- as.character(hugo.ordered.filtered
                        [1:sum(hugo.ordered.filtered[,2]>cutoff),1])

reduce.mat.filtered <- mat.filtered[sigRows,]
```

```
res.filtered <- pheatmap(reduce.mat.filtered,
          cluster_rows = F,
          show_colnames=FALSE)
```



```
#annotate groups
res.filtered <- pheatmap(reduce.mat.filtered,
          cluster_rows = F,
          show_colnames=FALSE,
          annotation_col = as.data.frame(cutree(res.filtered$tree_col, k = 2)))
```



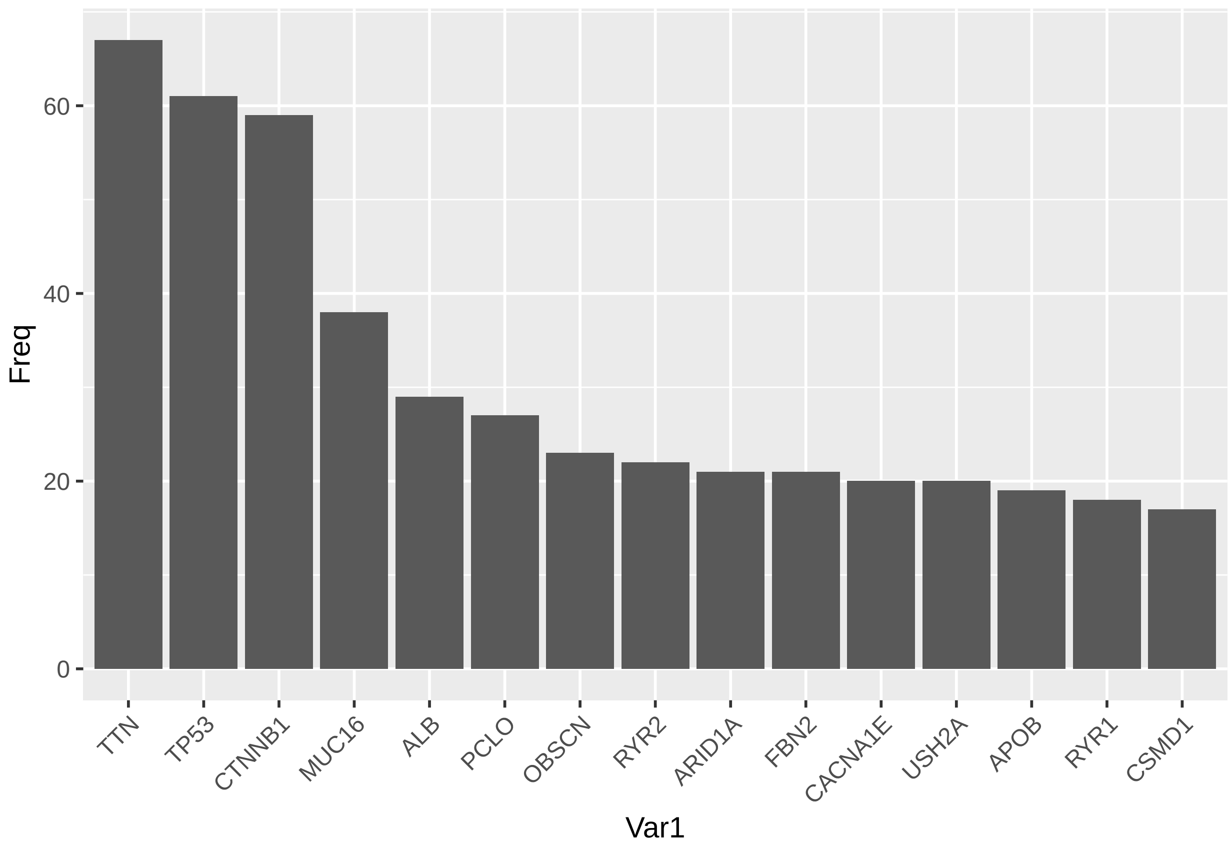
```
#two clusters
cluster <-  as.data.frame(cutree(res.filtered$tree_col, k = 2))
sum(cluster==1)
```

```
## [1] 350
```

```
sum(cluster==2)
```

```
## [1] 2
```

```
ggplot(data=hugo.ordered.filtered[1:15,], aes(x=Var1, y=Freq))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 45,hjust=1))+
  scale_x_discrete(limits = hugo.ordered.filtered[1:15,]$Var1)
```


Freq
0
20
40
60
Var1
TTN
TP53
CTNNB1
MUC16
ALB
PCLO
OBSCN
RYR2
ARID1A
FBN2
CACNA1E
USH2A
APOB
RYR1
CSMD1