```
raw.clinical.patients <- read.table("data_clinical_patient.txt", sep = "\t",
                                    header = TRUE)
raw.data.mutations <- read.table("data_mutations.txt", sep = "\t",
                                 header = TRUE)
raw.data.RNAseq <- read.csv("RNAseq_BRCA.csv", row.names=1)

#Filter data where you only have 0 or 1 read count across all samples.
raw.data.RNAseq <- raw.data.RNAseq[rowSums(raw.data.RNAseq)>1,]
```

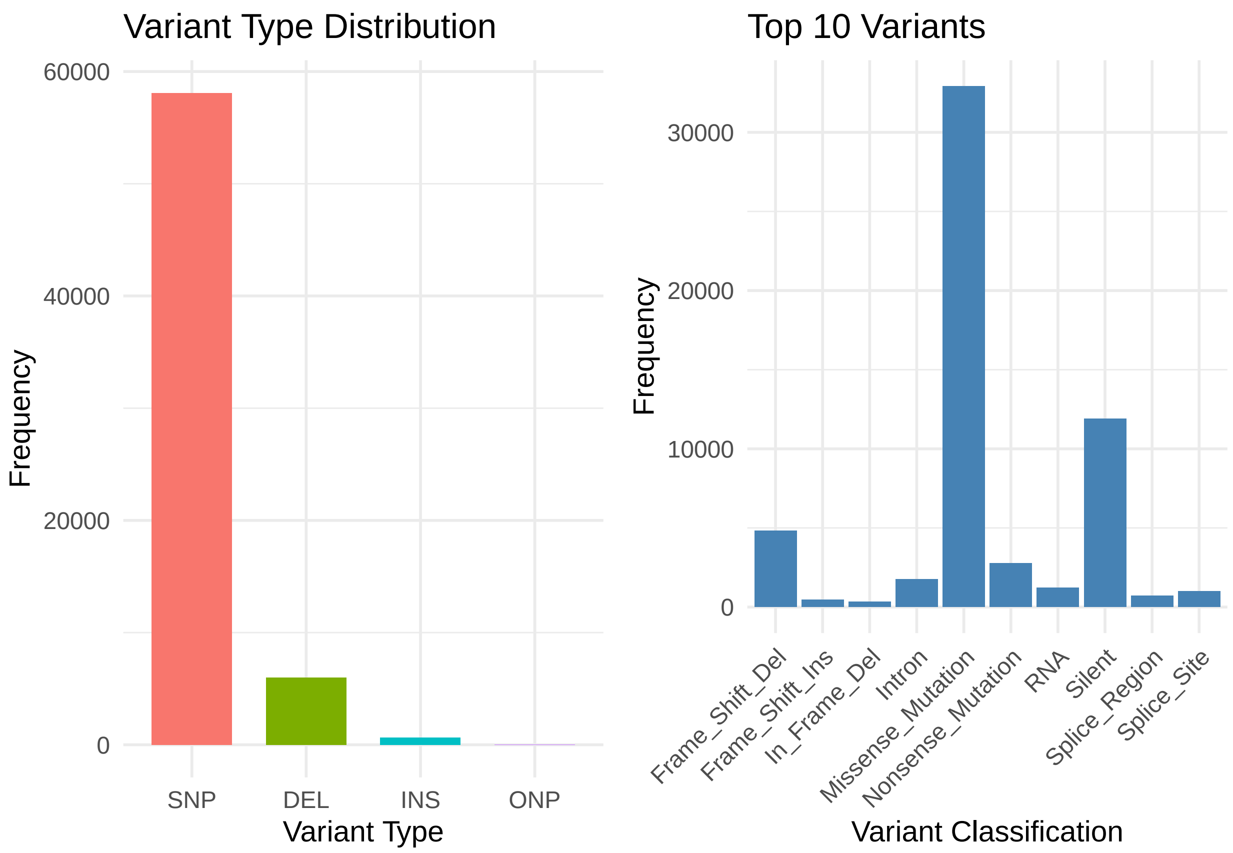
```
colnames(raw.data.RNAseq) <- make.unique(sapply(colnames(raw.data.RNAseq), function(name) {
  segments <- strsplit(name, "\\.")[[1]]
  paste(segments[1:3], collapse = "-")
}))
```

```
#Unique Patients in each data set
unique.clinical <- as.data.frame(unique(raw.clinical.patients$PATIENT_ID))
unique.mutations <- as.data.frame(unique
                                  (raw.data.mutations$Tumor_Sample_Barcode))
unique.RNA <- as.data.frame(colnames(raw.data.RNAseq[,1:length(raw.data.RNAseq)]))

#Addition patient ID's to Mutation data
mutation.patients <- as.data.frame(raw.data.mutations$Tumor_Sample_Barcode)
colnames(mutation.patients) <- "Patient_ID"
mutation.patients$Patient_ID <- substr(mutation.patients$Patient_ID, 1, 12)
raw.data.mutations <- cbind(mutation.patients, raw.data.mutations)

colnames(unique.clinical) <- "Patient_ID"
colnames(unique.mutations) <- "Patient_ID"
colnames(unique.RNA) <- "Patient_ID"
unique.mutations$Patient_ID <- substr(unique.mutations$Patient_ID, 1, 12)

#Finding common patients
common_patient_ids <- Reduce(intersect, list(
  unique.clinical$Patient_ID,
  unique.mutations$Patient_ID,
  unique.RNA$Patient_ID
))

#3 data sets with all 975 common patients
clinical.data <- raw.clinical.patients[raw.clinical.patients$PATIENT_ID
                                        %in% common_patient_ids, ]
mutation.data <- raw.data.mutations[raw.data.mutations$Patient_ID
                                    %in% common_patient_ids, ]
seq.data <- raw.data.RNAseq[,names(raw.data.RNAseq)
                            %in% clinical.data$PATIENT_ID]
```

```
library(ggplot2)
library(gridExtra)

data_counts_mutation <- data.frame(table(mutation.data$Hugo_Symbol))
colnames(data_counts_mutation) <- c("Gene", "Count")
data_counts_mutation$Percentage <- round((data_counts_mutation$Count /
                                            sum(data_counts_mutation$Count)) * 100, 4)
```

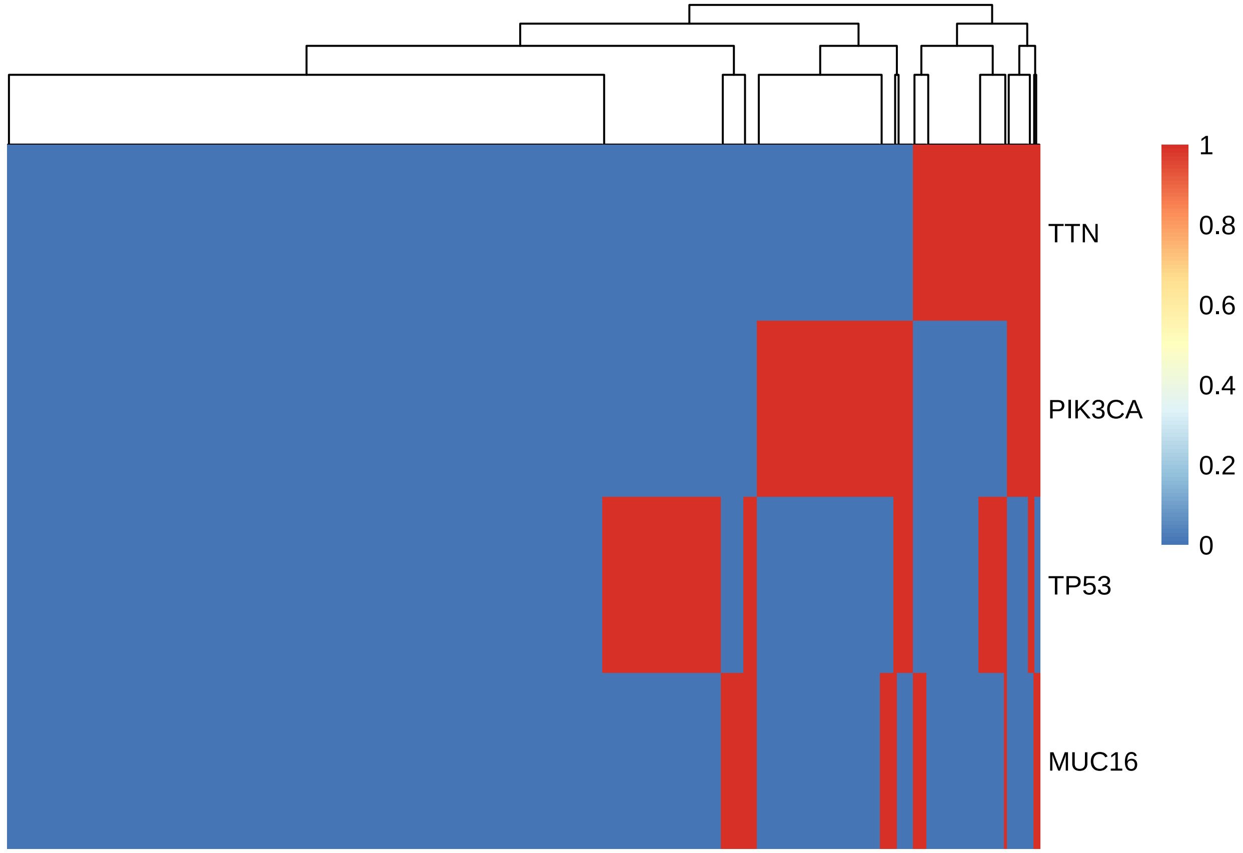
```r
top_50_mutations <- data_counts_mutation[order(data_counts_mutation$Count,
                                       decreasing = TRUE), ][1:30, ]



# Reorder the gene factor for the top 50
top_50_mutations$Gene <- factor(top_50_mutations$Gene,
                                    levels = top_50_mutations$Gene
                                    [order(top_50_mutations$Count,
                                           decreasing = TRUE)])

top_Mutations_plt <- ggplot(top_50_mutations, aes(x = Gene, y=Count)) +
  geom_bar(stat="identity", fill = "steelblue") +
  labs(title = "Top 30 Gene Variants", x = "Gene", y = "Count") +
  theme_minimal()+
  theme(plot.title = element_text(size = 20)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),)

ggsave("top_30_mutation.png", top_Mutations_plt, width = 8, height =6)
```

```r
# Create data frame for Variant Type
var.type <- as.data.frame(table(mutation.data$Variant_Type))
var.type <- var.type[order(var.type$Freq, decreasing = TRUE), ]

# Convert Var1 to a factor with levels in the desired order
var.type$Var1 <- factor(var.type$Var1, levels = var.type$Var1)

# Plot 1: Variant Type Distribution
plt1 <- ggplot(data = var.type, aes(x = Var1, y = Freq)) +
  geom_col(aes(fill = Var1), width = 0.7) +
  labs(title = "Variant Type Distribution", x = "Variant Type", y = "Frequency") +
  theme_minimal()+
  theme(legend.position = "none")

# Create data frame for Variant Classification
var.class <- as.data.frame(table(mutation.data$Variant_Classification))

# Plot 2: Top 10 Variants
plt2 <- ggplot(data = var.class[var.class$Freq > 100, ], aes(x = Var1, y = Freq)) +
  geom_col(fill = "steelblue") +
  labs(title = "Top 10 Variants", x = "Variant Classification", y = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


plt_var_class <- grid.arrange(plt1, plt2, nrow = 1)
```



```r
# Save Combined Plot
ggsave("plt_var_class.png", plot = plt_var_class, width = 6, height = 4)
```

```r
library(pheatmap)
cnv_events = unique(mutation.data$Variant_Classification)
oncomat = reshape2::dcast(
  data = mutation.data,
  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 = ';$',
             replacement = '',
             x = x)
    x = gsub(pattern = '^;',
             replacement = '',
             x = x)
    return(x)
  },
  value.var = 'Variant_Classification',
  fill = '',
  drop = FALSE
)
rownames(oncomat) = oncomat$Hugo_Symbol
oncomat <- oncomat[,-1]
hugo <- as.data.frame(table(mutation.data$Hugo_Symbol))
oncomat.ordered <- oncomat[order(-hugo$Freq),]
mat <- oncomat.ordered
mat[mat!=""]=1
mat[mat==""]=0
mat <- apply(mat, 2 ,as.numeric)
mat <- as.matrix(mat)
rownames(mat)  <-  row.names(oncomat.ordered)
reduce.mat <- mat[1:4,]

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



```r
ggsave("Mutation_Heatmap.png", res$gtable, width = 7, height = 4)

cluster <- as.data.frame(cutree(res$tree_col,k = 2))
```

```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
```

```
##
## 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 object is masked from 'package:gridExtra':
##
##     combine

## 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, table,
##     tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## 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

## Loading required package: GenomeInfoDb

## 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
```

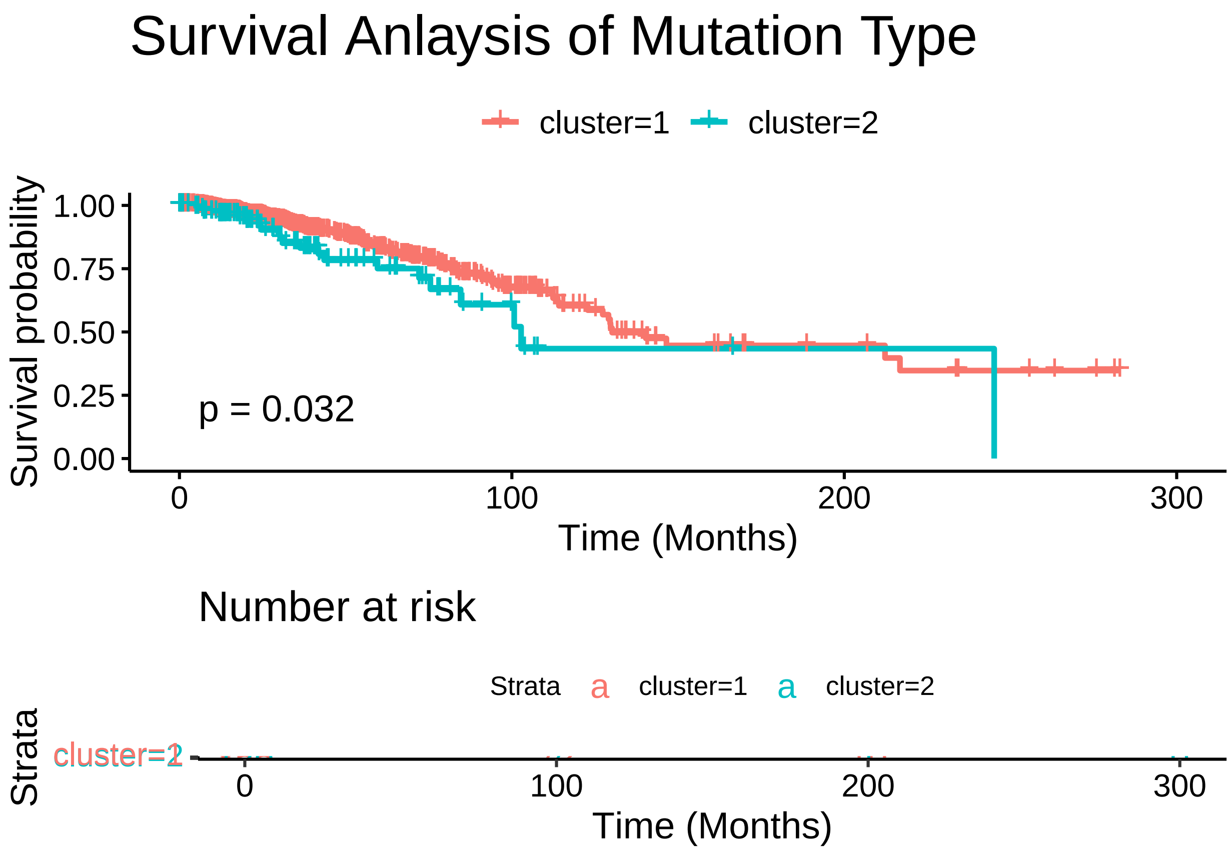
```
rownames(cluster) <- substr(rownames(cluster), 1, 12)
clinical.data$deceased = clinical.data$OS_STATUS == "1:DECEASED"
clinical_info <- clinical.data[, c("PATIENT_ID", "OS_MONTHS", "deceased")]
clinical_info$cluster <- cluster$`cutree(res$tree_col, k = 2)`

Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ cluster
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ cluster
```

```
fit = survfit(Surv(OS_MONTHS, deceased) ~ cluster, data=clinical_info)

mut_surve <- ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.tal

mut_surve$plot <- mut_surve$plot +
  theme(
    plot.title = element_text(size = 21),  # Title size
    legend.text = element_text(size = 12),              # Legend text size
    legend.title = element_text(size = 12)              # Legend title size
  ) +
  labs(color = NULL, fill = NULL, linetype = NULL)

  mut_surve$table <- mut_surve$table
  theme(legend.position = "none")
```

```
## List of 1
##  $ legend.position: chr "none"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

```
combined_plot_mut <- grid.arrange(mut_surve$plot, mut_surve$table, ncol = 1, heights = c(2, 1))
```



```
ggsave("Cluster_Heatmap.png", combined_plot_mut, width = 8, height = 6)

summary_fit <- summary(fit, times =60)

# Display survival probabilities for each cluster
print(summary_fit$surv)
```

```
## [1] 0.8301303 0.7505613
```

```
library(dplyr)
```

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

## The following object is masked from 'package:Biobase':
##
##     combine

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

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

## The following object is masked from 'package:matrixStats':
##
##     count

## The following object is masked from 'package:gridExtra':
##
##     combine

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

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

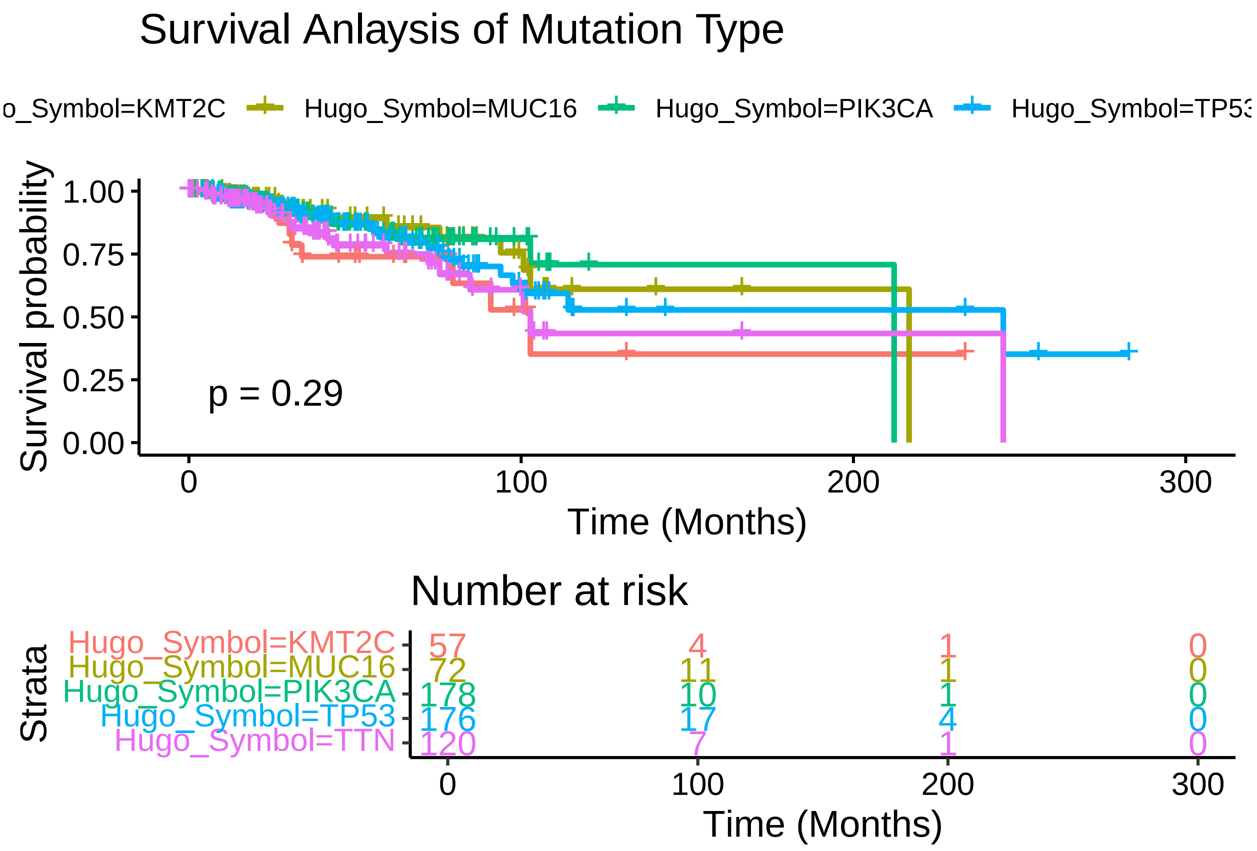
```r
clinical.data$deceased = clinical.data$OS_STATUS == "1:DECEASED"

filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[1:5], ]


clinical_info <- clinical.data[, c("PATIENT_ID", "OS_MONTHS", "deceased")]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

unique_merged_data <- merged_data %>%
  distinct(PATIENT_ID, Hugo_Symbol, .keep_all = TRUE)

table(unique_merged_data$Hugo_Symbol)
```

```
##
##  KMT2C  MUC16 PIK3CA   TP53    TTN
##     57     72    178    176    120
```

```r
Surv(unique_merged_data$OS_MONTHS, unique_merged_data$deceased) ~ unique_merged_data$Hugo_Symbol
```

```
## Surv(unique_merged_data$OS_MONTHS, unique_merged_data$deceased) ~
##     unique_merged_data$Hugo_Symbol
```

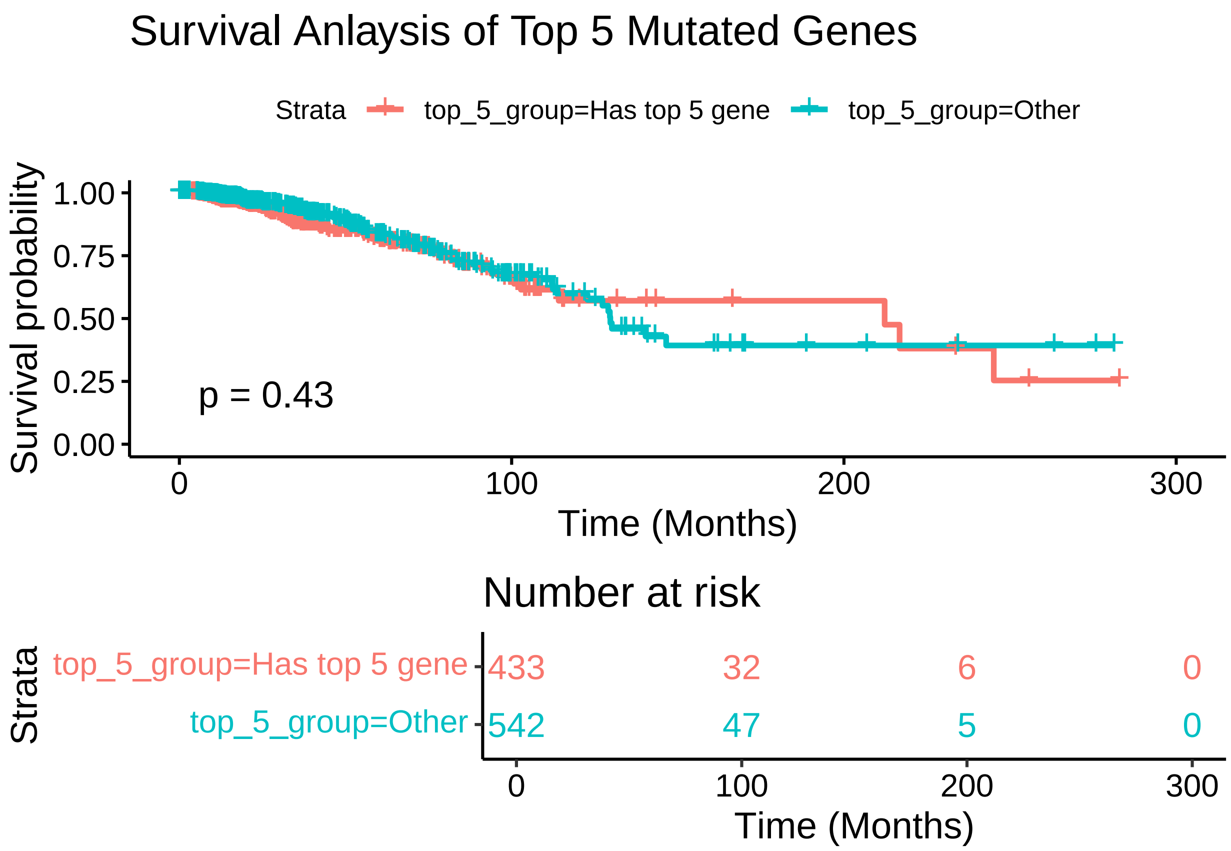
```
fit = survfit(Surv(OS_MONTHS, deceased) ~ Hugo_Symbol, data=unique_merged_data)

ggsurvplot(fit, data=unique_merged_data, pval=T, risk.table=T, risk.table.col="strata", risk.table.heig
```



```
filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[1:5], ]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

clinical_info$top_5_group <- ifelse(clinical_info$PATIENT_ID %in%
                                merged_data$PATIENT_ID, "Has top 5 gene", "Other")

Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ top_5_group
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ top_5_group
```

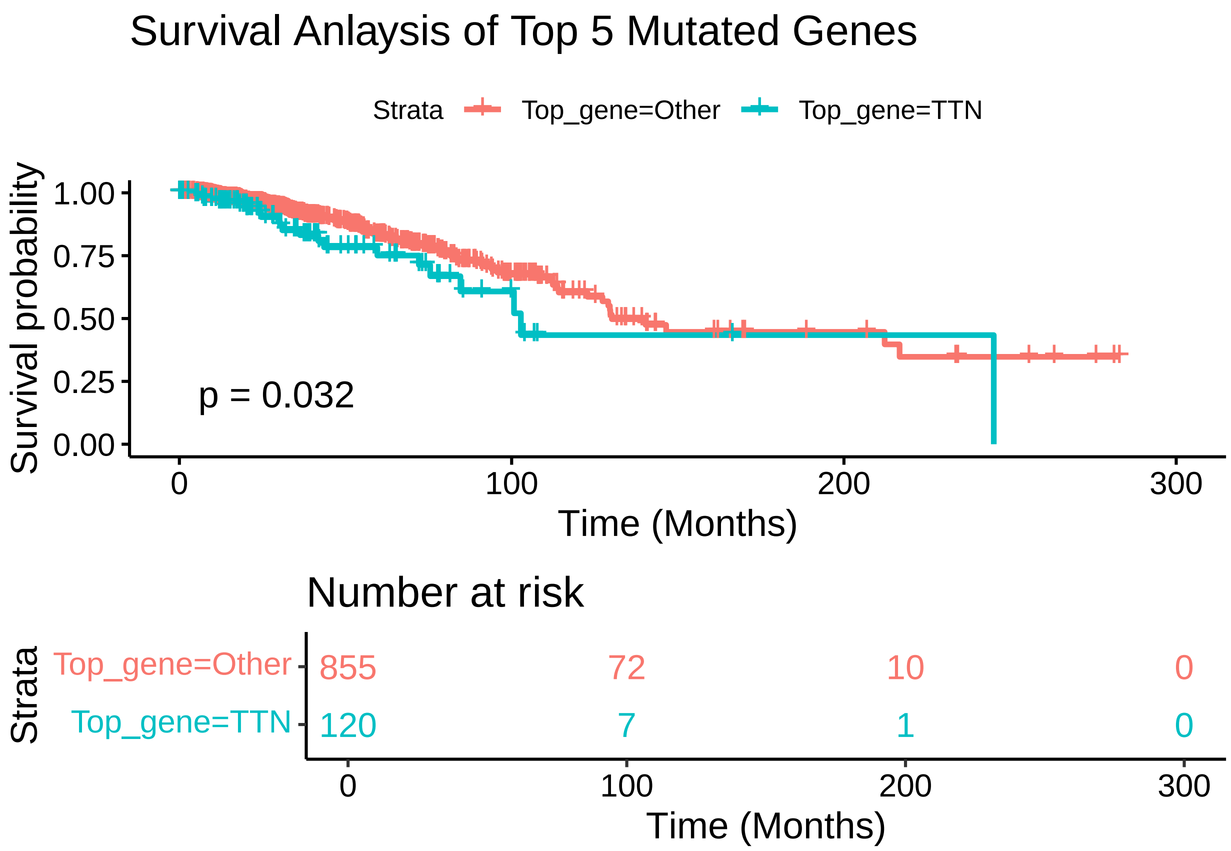
```
fit = survfit(Surv(OS_MONTHS, deceased) ~ top_5_group, data=clinical_info)

ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.table.height=0.
```



```
filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[1], ]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

clinical_info$Top_gene <- ifelse(clinical_info$PATIENT_ID %in%
                                 merged_data$PATIENT_ID, merged_data$Hugo_Symbol, "Other")


Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Top_gene
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Top_gene
```

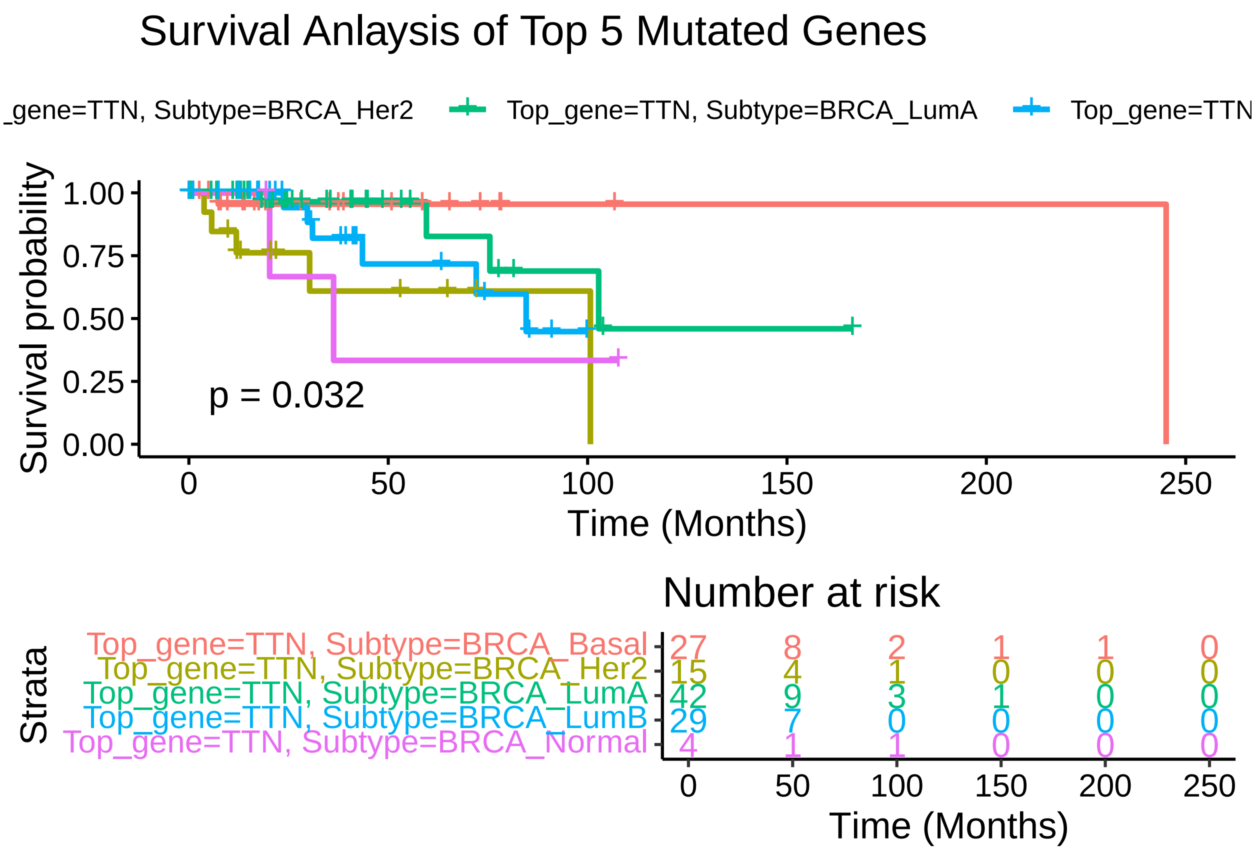
```
fit = survfit(Surv(OS_MONTHS, deceased) ~ Top_gene, data=clinical_info)

ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.table.height=0.
```



```
clinical_info$Subtype <- clinical.data$SUBTYPE

subgroup <- clinical_info[clinical_info$Top_gene == "TTN" & clinical_info$Subtype != ""
                          & !is.na(clinical_info$Subtype),]

Surv(subgroup$OS_MONTHS, subgroup$deceased) ~ Top_gene + Subtype
```

```
## Surv(subgroup$OS_MONTHS, subgroup$deceased) ~ Top_gene + Subtype
```

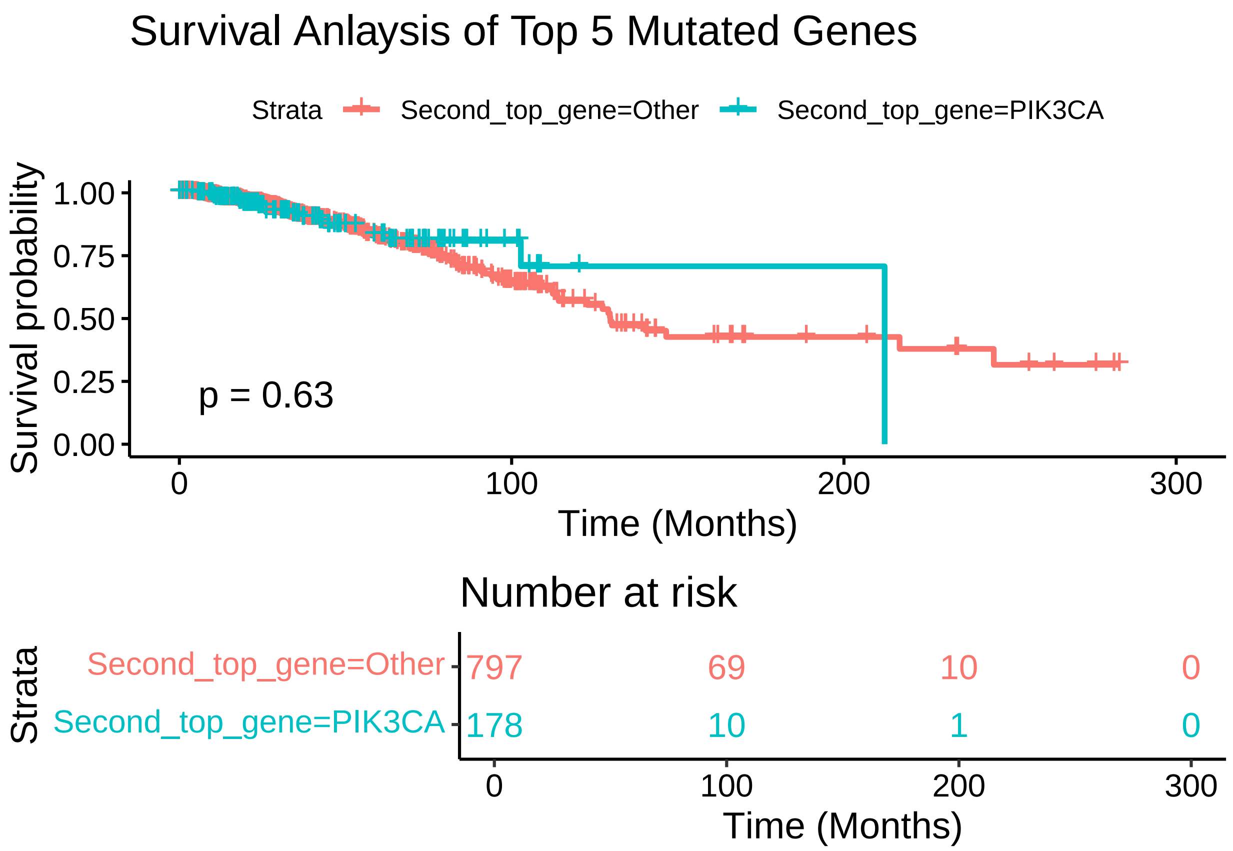
```
fit = survfit(Surv(OS_MONTHS, deceased) ~ Top_gene+ Subtype, data=subgroup)

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



```r
filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[2], ]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

clinical_info$Second_top_gene <- ifelse(clinical_info$PATIENT_ID %in%
                                 merged_data$PATIENT_ID, merged_data$Hugo_Symbol, "Other")

Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Second_top_gene
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Second_top_gene
```

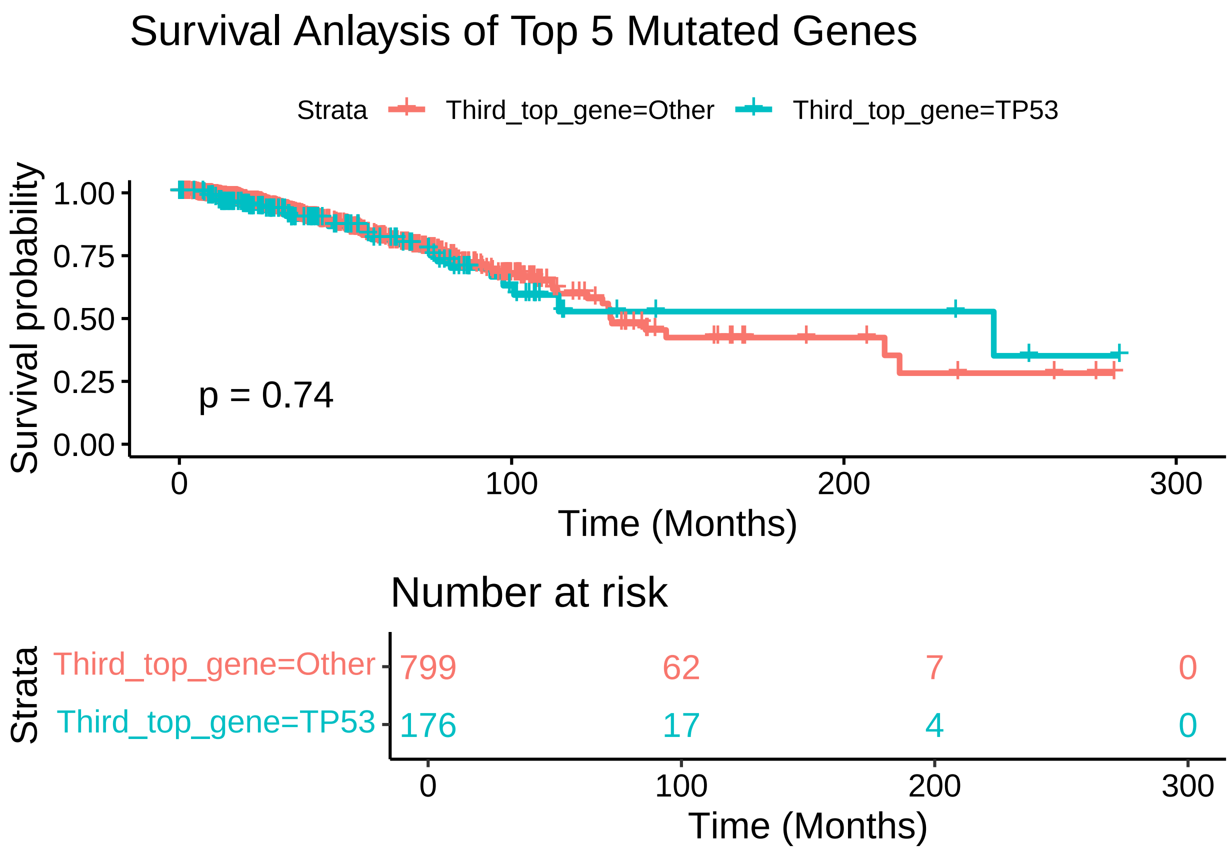
```r
fit = survfit(Surv(OS_MONTHS, deceased) ~ Second_top_gene, data=clinical_info)

ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.table.height=0.
```



```
filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[3], ]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

clinical_info$Third_top_gene <- ifelse(clinical_info$PATIENT_ID %in%
                                merged_data$PATIENT_ID, merged_data$Hugo_Symbol, "Other")

Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Third_top_gene
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Third_top_gene
```

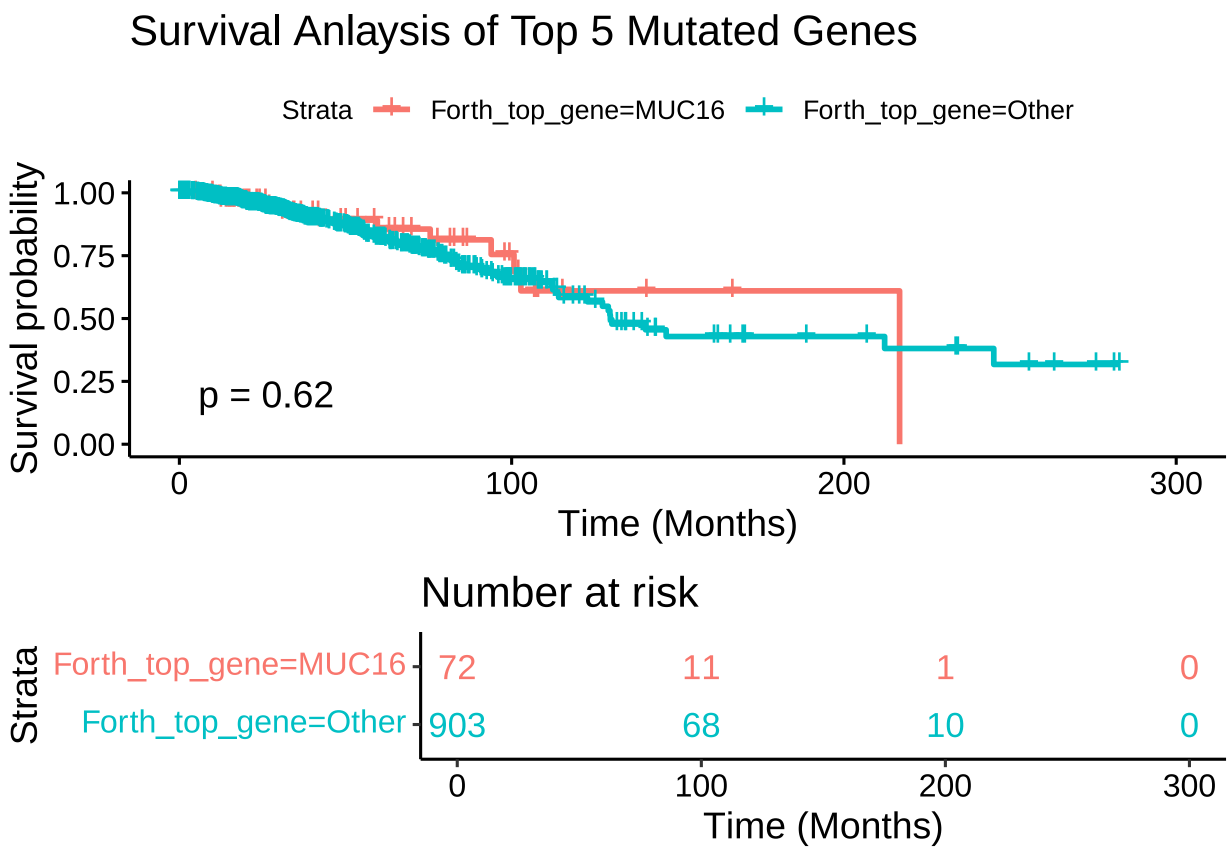
```
fit = survfit(Surv(OS_MONTHS, deceased) ~ Third_top_gene , data=clinical_info)

ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.table.height=0.
```



```
filtered_mutations <- mutation.data[mutation.data$Hugo_Symbol %in% top_50_mutations$Gene[4], ]

colnames(filtered_mutations)[colnames(filtered_mutations) == "Patient_ID"] <- "PATIENT_ID"

# Merge the filtered mutations with the clinical data based on Patient_ID
merged_data <- merge(filtered_mutations, clinical_info, by = "PATIENT_ID", all.x = TRUE)

clinical_info$Forth_top_gene <- ifelse(clinical_info$PATIENT_ID %in%
                                 merged_data$PATIENT_ID, merged_data$Hugo_Symbol, "Other")

Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Forth_top_gene
```

```
## Surv(clinical_info$OS_MONTHS, clinical_info$deceased) ~ Forth_top_gene
```

```
fit = survfit(Surv(OS_MONTHS, deceased) ~ Forth_top_gene , data=clinical_info)

ggsurvplot(fit, data=clinical_info, pval=T, risk.table=T, risk.table.col="strata", risk.table.height=0.
```


Survival Anlaysis of Top 5 Mutated Genes
Strata
Forth_top_gene=MUC16
Forth_top_gene=Other
Survival probability
1.00
0.75
0.50
0.25
0.00
p = 0.62
0
100
200
300
Time (Months)
Number at risk
Strata
Forth_top_gene=MUC16
Forth_top_gene=Other
72
11
1
0
903
68
10
0
0
100
200
300
Time (Months)

#TTN