LUSC Data Analysis - BMEG 310 Final Project Report

2024-11-21

Clinical Data

```
# Load the dataset
clinical_data <- read.delim("data_clinical_patient.txt", header = TRUE, sep = "\t",
                            stringsAsFactors = FALSE, skip = 4)

# glimpse(clinical_data)
```

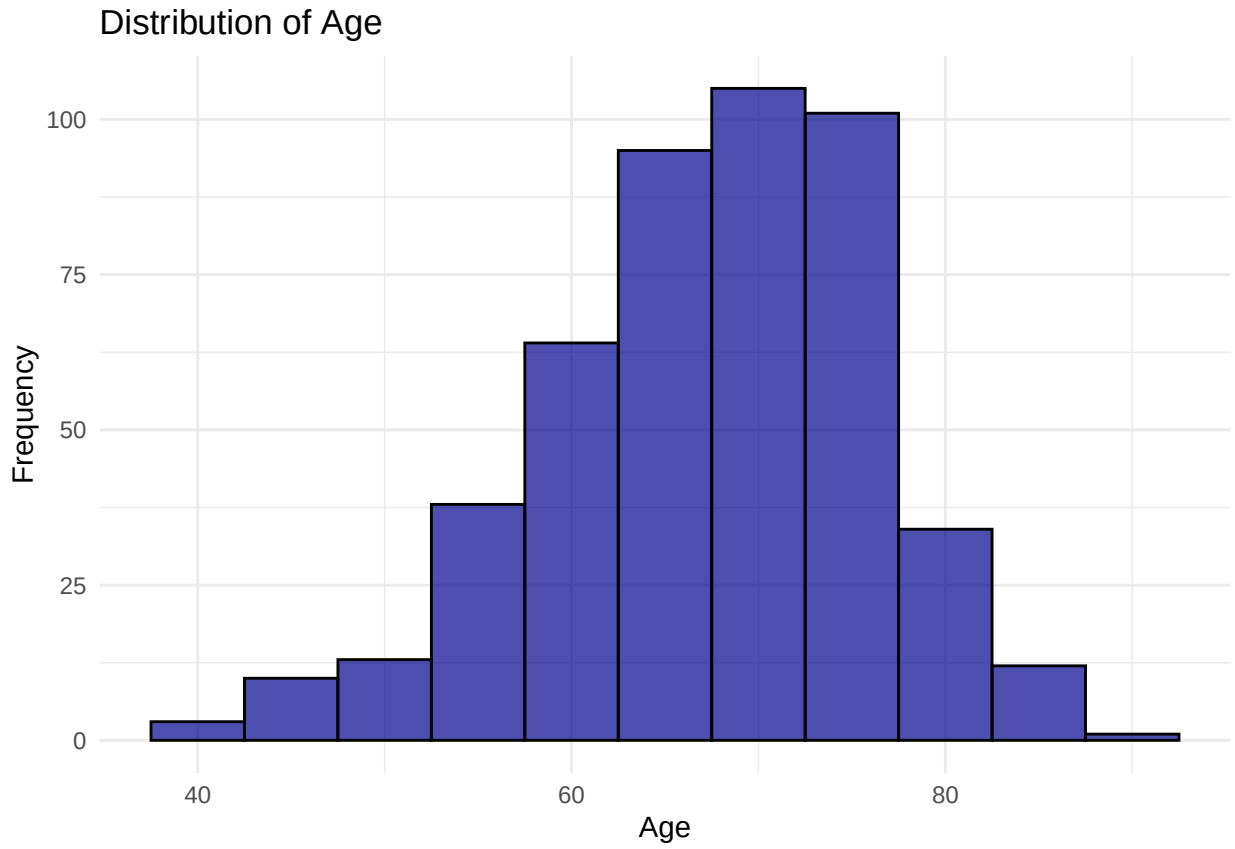
Data Cleaning and Preparation

```
# Clean up the dataset for exploration
clinical_data_clean <- clinical_data %>%
  mutate(
    AGE = as.numeric(AGE),
    AJCC_PATHOLOGIC_TUMOR_STAGE = as.factor(AJCC_PATHOLOGIC_TUMOR_STAGE),
    SEX = as.factor(SEX),
    OS_MONTHS = as.numeric(OS_MONTHS),
    PFS_MONTHS = as.numeric(PFS_MONTHS)
  )

# Additional cleaning for survival analysis
clinical_data <- clinical_data_clean %>%
  mutate(
    deceased = ifelse(OS_STATUS == "1:DECEASED", 1, 0),
    overall_survival = OS_MONTHS,
    gender = ifelse(SEX == "Male", "Male", "Female"),
    tumor_stage = case_when(
      AJCC_PATHOLOGIC_TUMOR_STAGE %in% c("STAGE IA", "STAGE IB") ~ "Stage I",
      AJCC_PATHOLOGIC_TUMOR_STAGE %in% c("STAGE IIA", "STAGE IIB") ~ "Stage II",
      AJCC_PATHOLOGIC_TUMOR_STAGE %in% c("STAGE IIIA", "STAGE IIIB",
                                         "STAGE IIIC") ~ "Stage III",
      AJCC_PATHOLOGIC_TUMOR_STAGE %in% c("STAGE IVA", "STAGE IVB") ~ "Stage IV",
      TRUE ~ "Unknown"
    )
  ) %>%
  filter(!is.na(overall_survival), tumor_stage != "Unknown")
```

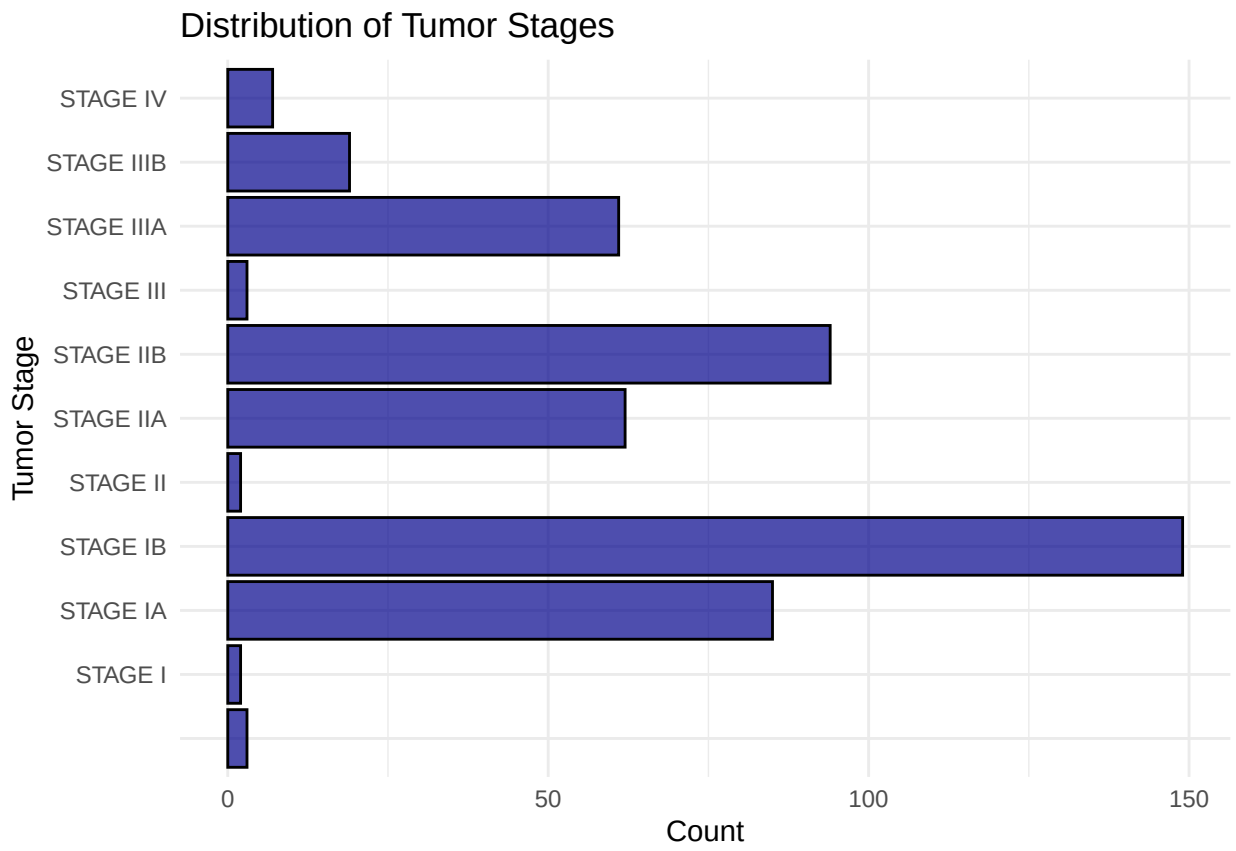
Data Visualization

```
# 1. Distribution of Age
ggplot(clinical_data_clean, aes(x = AGE)) +
  geom_histogram(binwidth = 5, color = "black", fill = "darkblue", alpha = 0.7) +
  labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
  theme_minimal()
```

```
## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

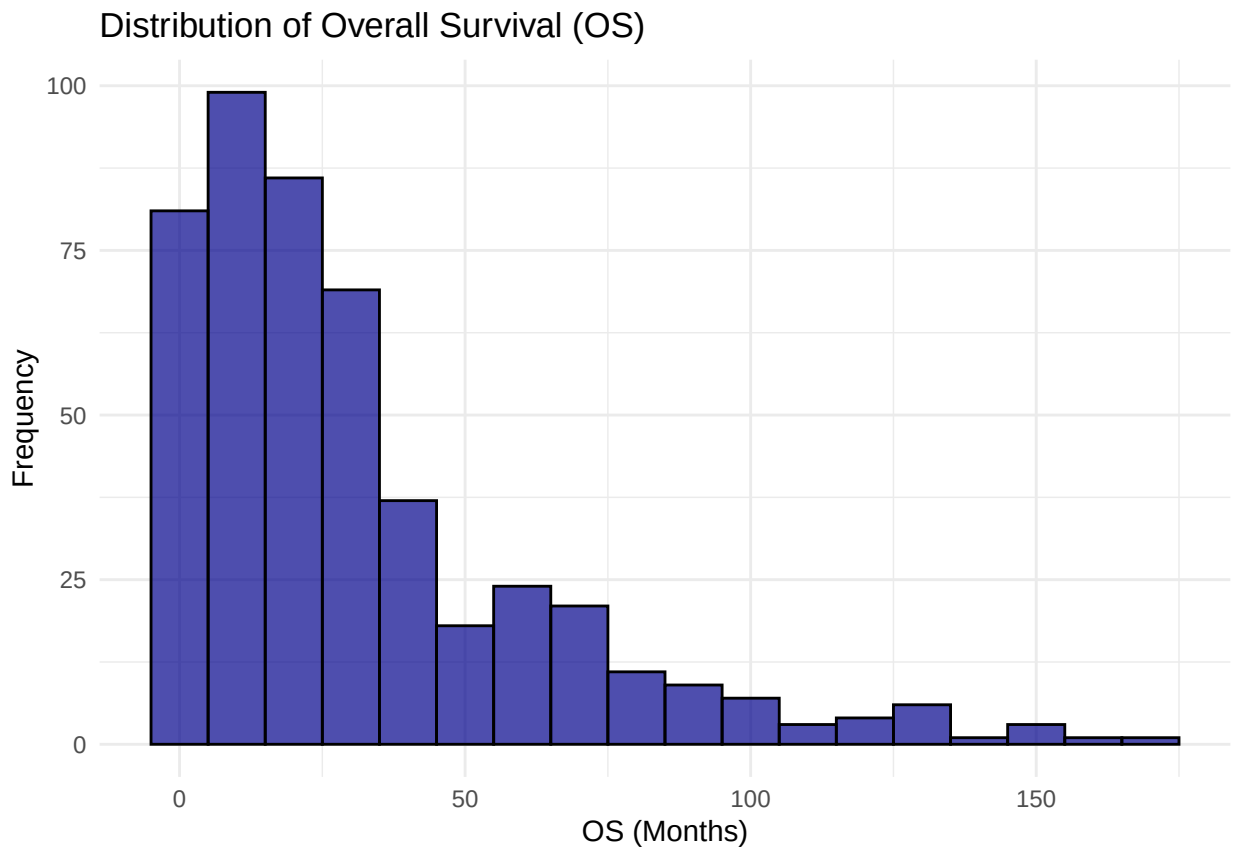



```r
# 2. Tumor Stage Distribution
ggplot(clinical_data_clean, aes(x = AJCC_PATHOLOGIC_TUMOR_STAGE)) +
  geom_bar(color = "black", fill = "darkblue", alpha = 0.7) +
  labs(title = "Distribution of Tumor Stages", x = "Tumor Stage", y = "Count") +
  theme_minimal() +
  coord_flip()
```



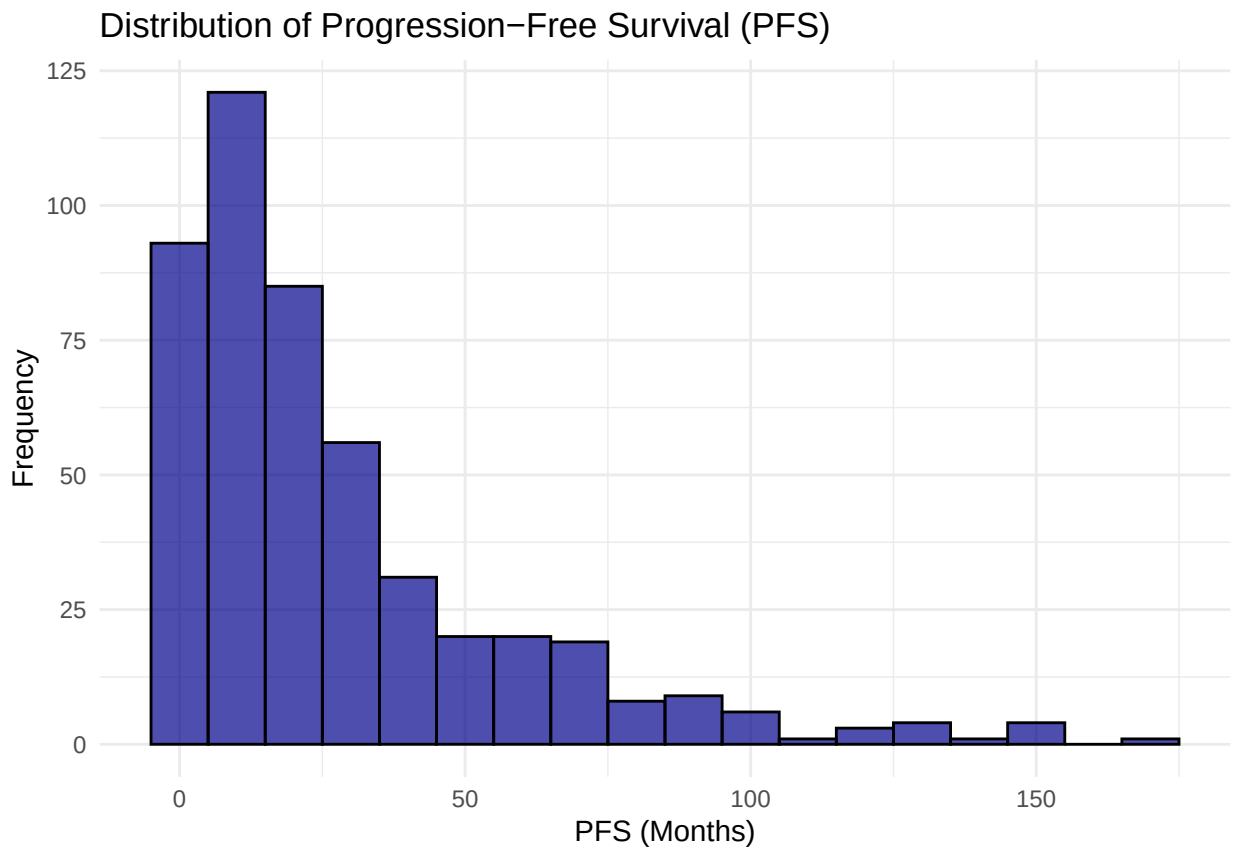

```r
# 3. Overall Survival (OS) Distribution
ggplot(clinical_data_clean, aes(x = OS_MONTHS)) +
  geom_histogram(binwidth = 10, color = "black", fill = "darkblue", alpha = 0.7) +
  labs(title = "Distribution of Overall Survival (OS)", x = "OS (Months)",
       y = "Frequency") +
  theme_minimal()
```

```
## Warning: Removed 6 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



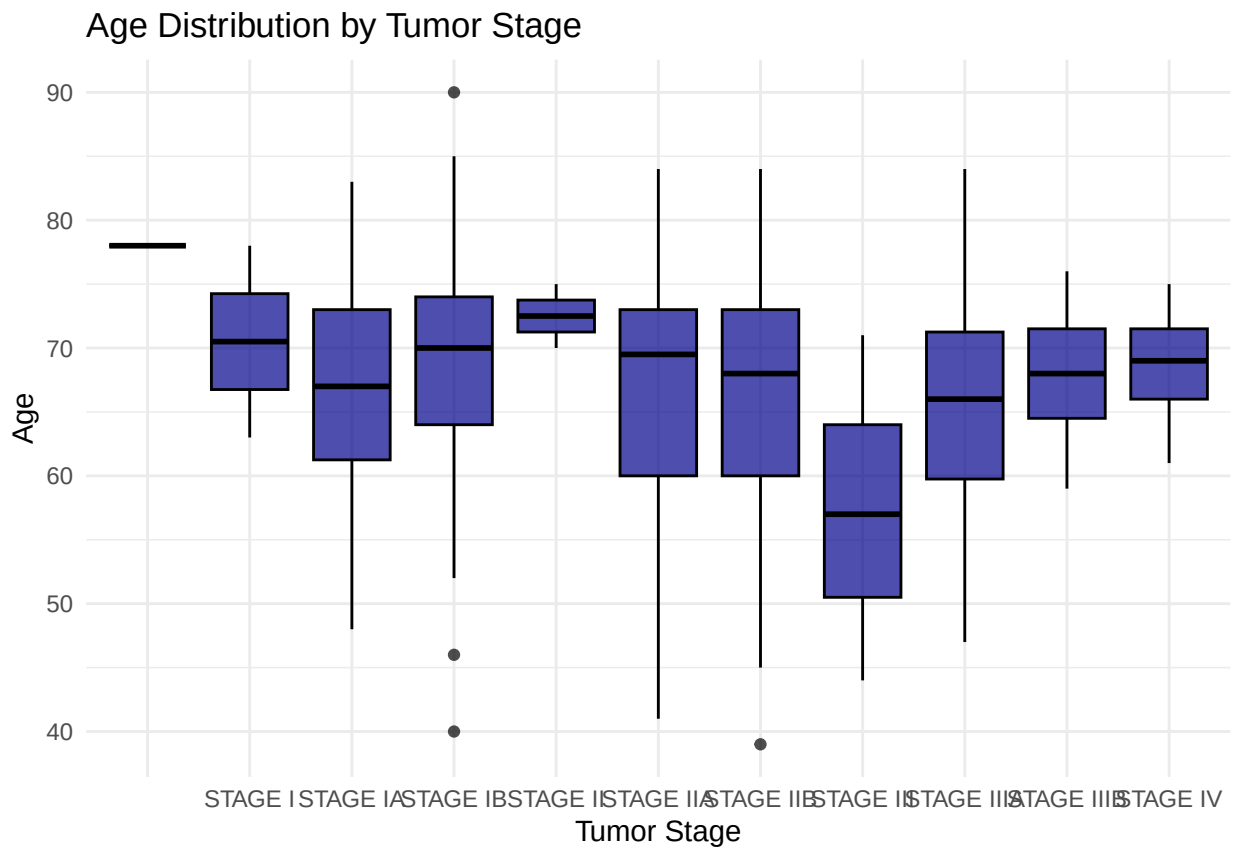

```r
# 4. Progression-Free Survival (PFS) Distribution
ggplot(clinical_data_clean, aes(x = PFS_MONTHS)) +
  geom_histogram(binwidth = 10, color = "black", fill = "darkblue", alpha = 0.7) +
  labs(title = "Distribution of Progression-Free Survival (PFS)", x = "PFS (Months)",
       y = "Frequency") +
  theme_minimal()
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



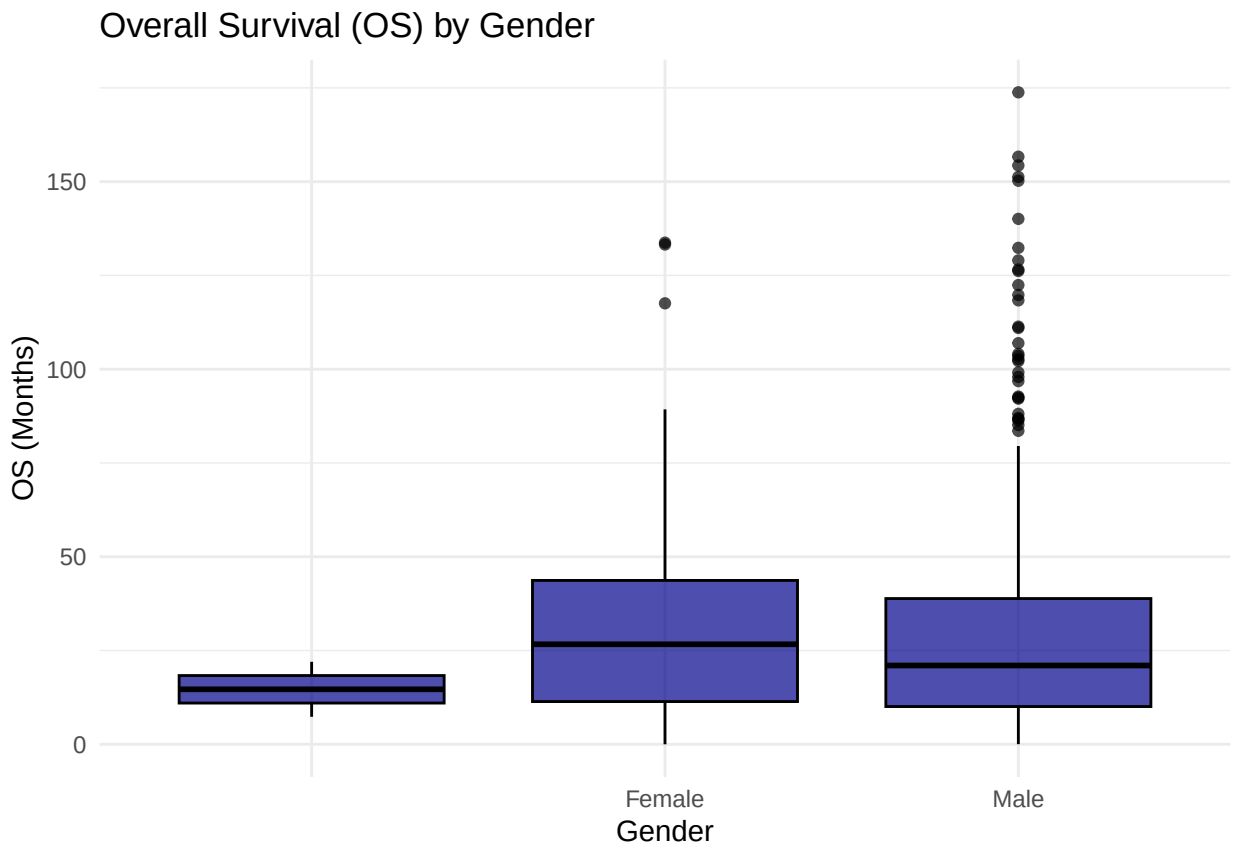

```r
# 5. Age Distribution by Tumor Stage
ggplot(clinical_data_clean, aes(x = AJCC_PATHOLOGIC_TUMOR_STAGE, y = AGE,
                                fill = AJCC_PATHOLOGIC_TUMOR_STAGE)) +
  geom_boxplot(alpha = 0.7, color = "black", fill = "darkblue") +
  labs(title = "Age Distribution by Tumor Stage", x = "Tumor Stage", y = "Age") +
  theme_minimal() +
  theme(legend.position = "none")
```

```
## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



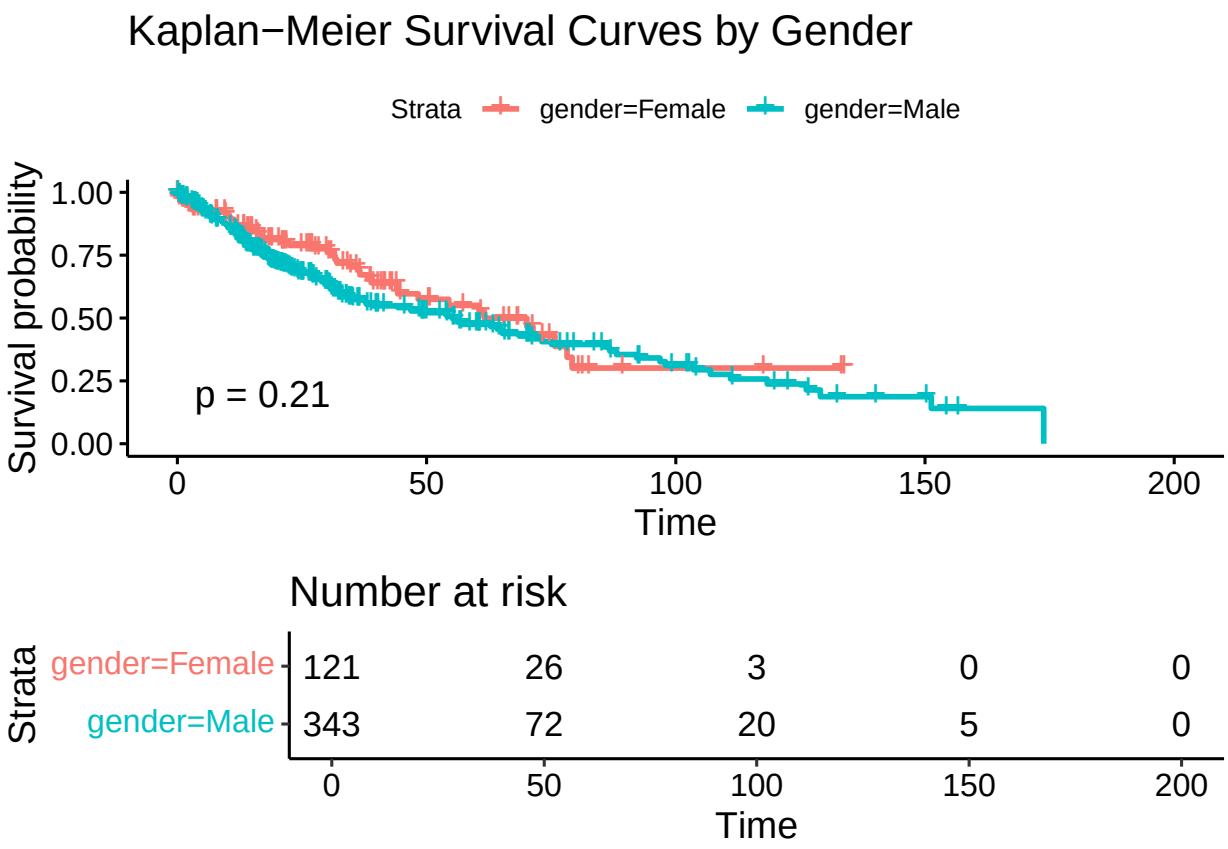

```
# 6. Overall Survival (OS) by Gender
ggplot(clinical_data_clean, aes(x = SEX, y = OS_MONTHS, fill = SEX)) +
  geom_boxplot(alpha = 0.7, color = "black", fill = "darkblue") +
  labs(title = "Overall Survival (OS) by Gender", x = "Gender", y = "OS (Months)") +
  theme_minimal() +
  theme(legend.position = "none")
```

```
## Warning: Removed 6 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

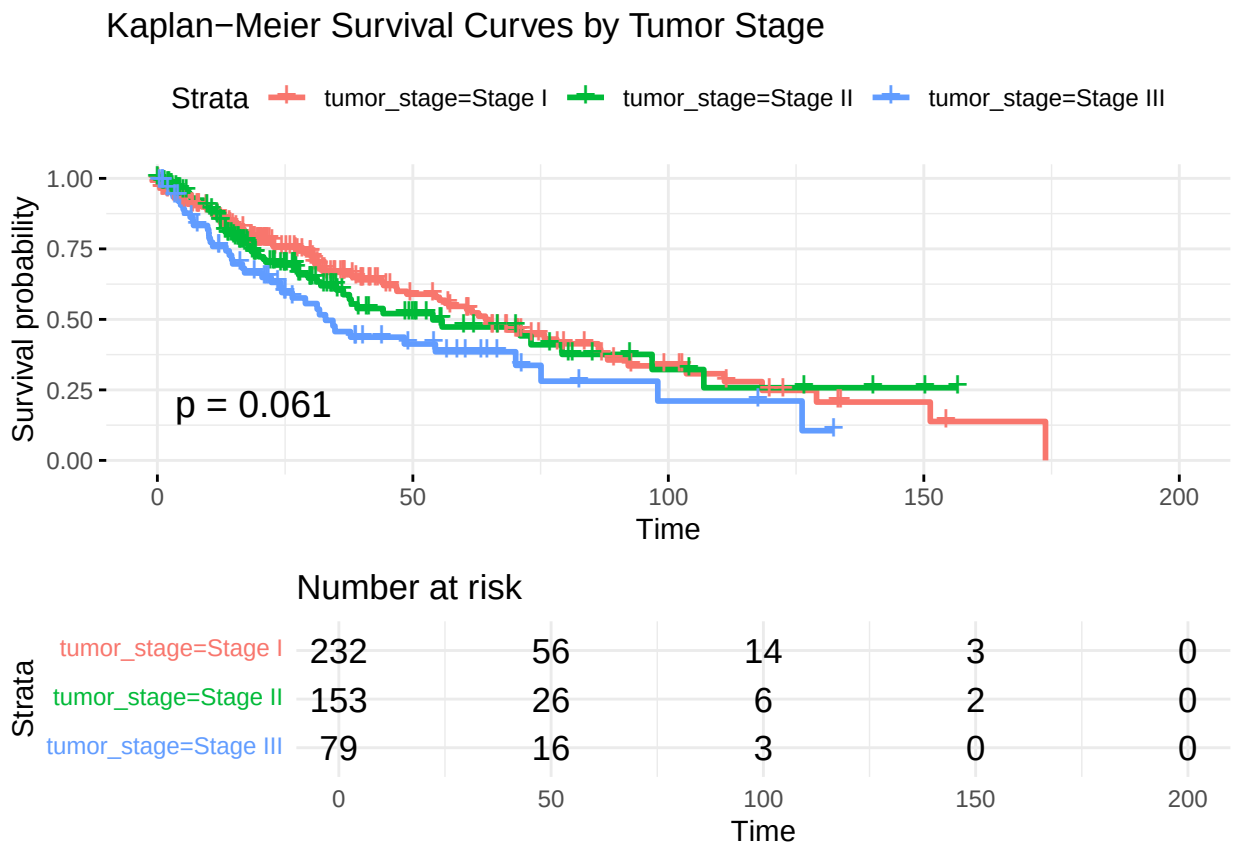


Survival Analysis

```
# Create survival object
surv_object <- Surv(clinical_data$overall_survival, clinical_data$deceased)

# 1. Survival Analysis by Gender
fit_gender <- survfit(surv_object ~ gender, data = clinical_data)
ggsurvplot(fit_gender, data = clinical_data, pval = TRUE,
           risk.table = TRUE, risk.table.height = 0.35,
           title = "Kaplan-Meier Survival Curves by Gender")
```




```r
# 2. Survival Analysis by Tumor Stage
fit_stage <- survfit(surv_object ~ tumor_stage, data = clinical_data)
ggsurvplot(fit_stage, data = clinical_data, pval = TRUE,
           risk.table = TRUE, risk.table.height = 0.35,
           ggtheme = theme_minimal(),
           title = "Kaplan-Meier Survival Curves by Tumor Stage")
```



Mutation Data

```
# Read in mutation data
mutation_data <- read.delim("data_mutations.txt", header = TRUE, sep = "\t",
                            stringsAsFactors = FALSE)

# glimpse(mutation_data)
```

```
# Create the mutation matrix
mutation_matrix <- mutation_data %>%
  group_by(Tumor_Sample_Barcode, Hugo_Symbol) %>%
  summarize(
    mutation_status = ifelse(
      any(Variant_Classification %in% c(
        "Missense_Mutation", "Nonsense_Mutation", "Frame_Shift",
        "Splice_Site", "In_Frame"), na.rm = TRUE),
      1, 0
    ),
    .groups = "drop"
  ) %>%
  pivot_wider(names_from = Hugo_Symbol, values_from = mutation_status, values_fill = 0)

# Check the mutation matrix
print(dim(mutation_matrix))
```

```
## [1]   469 18905
```

```r
print(head(mutation_matrix))
```

```
## # A tibble: 6 x 18,905
##   Tumor_Sample_Barcode ABCA2 ABCC1  ACP2 ADAM20 ADAMTS12 ADAMTS19 ADCY4 ADCY5
##   <chr>                <dbl> <dbl> <dbl>  <dbl>    <dbl>    <dbl> <dbl> <dbl>
## 1 TCGA-18-3406-01          1     1     0      1        0        1     1     0
## 2 TCGA-18-3407-01          0     0     0      0        0        0     0     0
## 3 TCGA-18-3408-01          0     0     0      0        0        0     0     0
## 4 TCGA-18-3410-01          0     0     0      0        0        0     0     0
## 5 TCGA-18-3411-01          0     0     0      0        0        0     0     0
## 6 TCGA-18-3412-01          0     0     0      0        0        0     1     0
## # i 18,896 more variables: AGRN <dbl>, AKAP9 <dbl>, AMER1 <dbl>, AMIGO1 <dbl>,
## #   ANKMY2 <dbl>, ANPEP <dbl>, AQP1 <dbl>, ARHGEF12 <dbl>, ARID1B <dbl>,
## #   ASH1L <dbl>, ATF6B <dbl>, ATN1 <dbl>, ATP5G2 <dbl>, ATP8B1 <dbl>,
## #   ATRX <dbl>, BAI1 <dbl>, BAI2 <dbl>, BAIAP3 <dbl>, BDP1 <dbl>, BHMT <dbl>,
## #   BIRC6 <dbl>, BRSK1 <dbl>, BRWD3 <dbl>, C10orf88 <dbl>, C12orf5 <dbl>,
## #   C12orf76 <dbl>, C18orf54 <dbl>, C4BPB <dbl>, C7 <dbl>, CA7 <dbl>,
## #   CARD6 <dbl>, CASP9 <dbl>, CCAR1 <dbl>, CCDC141 <dbl>, CCDC155 <dbl>, ...
```

```r
mutation_matrix_clust <- mutation_matrix %>%
  column_to_rownames("Tumor_Sample_Barcode") %>%  # Set Tumor_Sample_Barcode as row names
  as.matrix()

# Check dimensions
print(dim(mutation_matrix_clust))
```

```
## [1]   469 18904
```

```r
# Identify the top 20 most mutated genes
gene_frequencies <- colSums(mutation_matrix_clust)
print(gene_frequencies)
```

```
##                ABCA2                ABCC1                 ACP2
##                   11                   10                    2
##               ADAM20              ADAMTS12              ADAMTS19
##                    6                   78                   24
##               ADCY4                ADCY5                 AGRN
##                   11                   13                   10
##               AKAP9                AMER1               AMIGO1
##                   31                   16                    8
##              ANKMY2                ANPEP                 AQP1
##                    7                   11                    2
##             ARHGEF12               ARID1B                ASH1L
##                   13                   26                   22
##               ATF6B                 ATN1               ATP5G2
##                    4                   12                    5
##              ATP8B1                 ATRX                 BAI1
##                    6                   20                   21
##               BAI2                BAIAP3                 BDP1
##                   17                   11                   19
##               BHMT                BIRC6                BRSK1
##                    8                   48                    9
```

```
##                   REEP3
##                       0
```

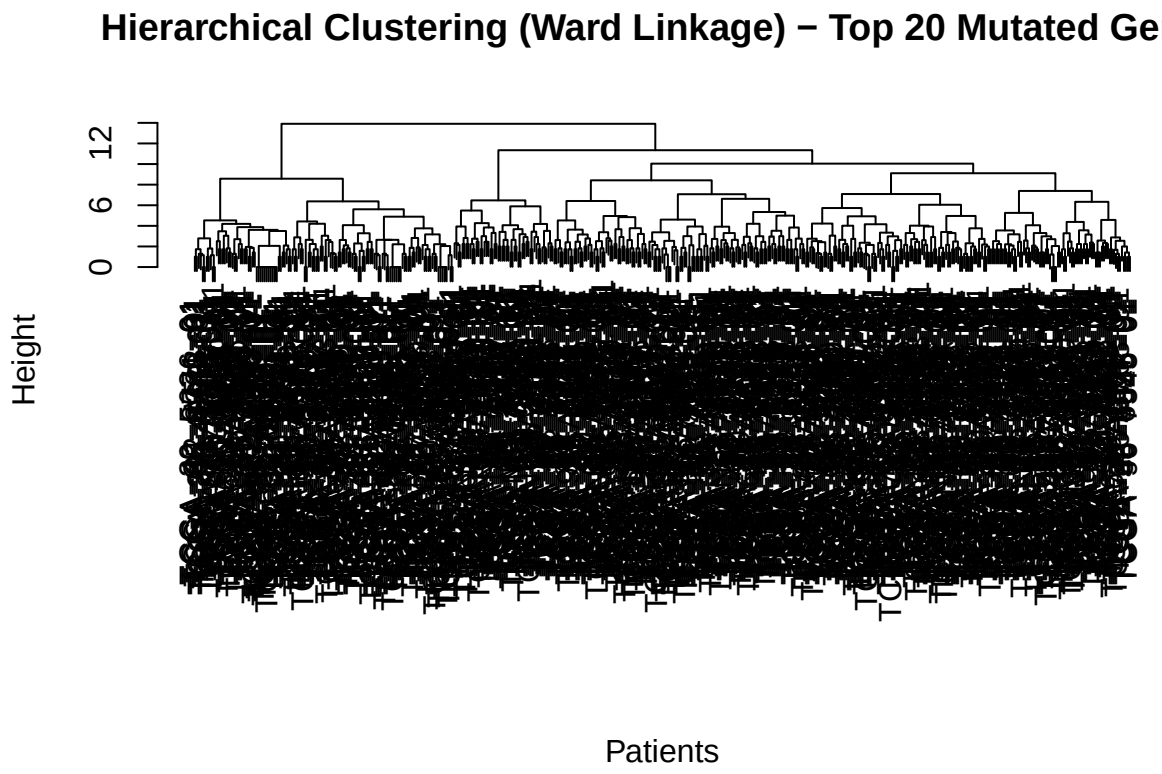
```
top_genes <- names(sort(gene_frequencies, decreasing = TRUE)[1:20])
top_genes <- intersect(top_genes, colnames(mutation_matrix_clust))
print(top_genes)
```

```
##  [1] "TTN"     "TP53"    "CSMD3"   "RYR2"    "MUC16"   "LRP1B"   "USH2A"
##  [8] "SYNE1"   "ZFHX4"   "FAM135B" "NAV3"    "SPTA1"   "CDH10"   "XIRP2"
## [15] "PCDH15"  "RYR3"    "KMT2D"   "DNAH5"   "PKHD1L1" "PAPPA2"
```

```
# Subset the mutation matrix
mutation_matrix_top_20 <- mutation_matrix_clust[, top_genes, drop = FALSE]
print(dim(mutation_matrix_top_20))
```

```
## [1] 469  20
```

```
# Perform hierarchical clustering
hc_ward <- hclust(dist(mutation_matrix_top_20), method = "ward.D2")
plot(hc_ward, main = "Hierarchical Clustering (Ward Linkage) - Top 20 Mutated Genes", xlab = "Patients"
```

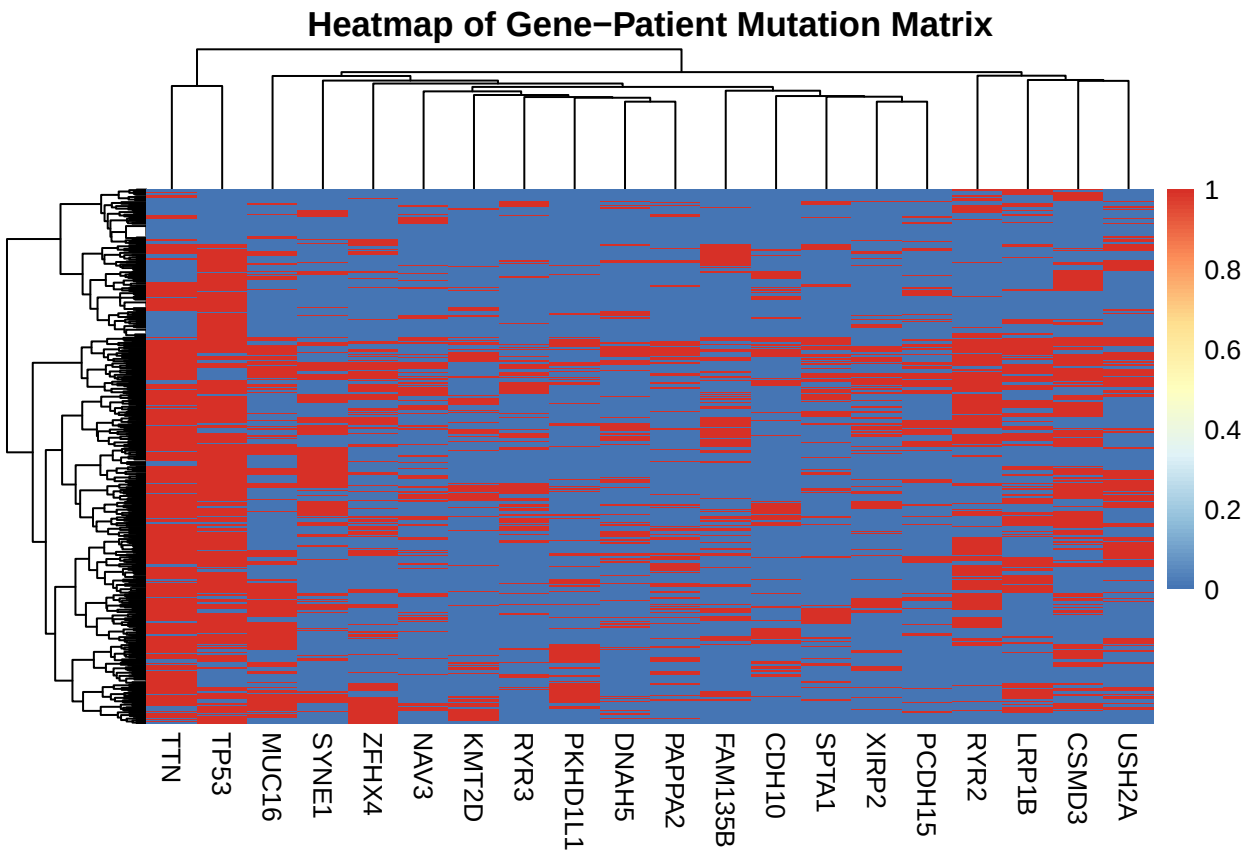



```
# Save cluster assignments
num_clusters <- 3
clusters <- cutree(hc_ward, k = num_clusters)
```

```r
# Create a dataframe with patient clusters
if (nrow(mutation_matrix_top_20) != length(clusters)) {
  stop("Mismatch between number of patients and cluster assignments.")
}

patient_clusters <- data.frame(
  Tumor_Sample_Barcode = rownames(mutation_matrix_top_20),
  Cluster = clusters
)

# Check and save patient clusters
print(head(patient_clusters))
```

```
##                 Tumor_Sample_Barcode Cluster
## TCGA-18-3406-01      TCGA-18-3406-01       1
## TCGA-18-3407-01      TCGA-18-3407-01       1
## TCGA-18-3408-01      TCGA-18-3408-01       2
## TCGA-18-3410-01      TCGA-18-3410-01       1
## TCGA-18-3411-01      TCGA-18-3411-01       1
## TCGA-18-3412-01      TCGA-18-3412-01       2
```

```r
write.csv(patient_clusters, "patient_clusters.csv", row.names = FALSE)


pheatmap(mutation_matrix_top_20, cluster_rows = hc_ward, cluster_cols = TRUE,
main = "Heatmap of Gene-Patient Mutation Matrix",
show_rownames = FALSE, show_colnames = TRUE, scale = "none")
```




```
head(patient_clusters)
```

```
##                  Tumor_Sample_Barcode Cluster
## TCGA-18-3406-01      TCGA-18-3406-01       1
## TCGA-18-3407-01      TCGA-18-3407-01       1
## TCGA-18-3408-01      TCGA-18-3408-01       2
## TCGA-18-3410-01      TCGA-18-3410-01       1
## TCGA-18-3411-01      TCGA-18-3411-01       1
## TCGA-18-3412-01      TCGA-18-3412-01       2
```

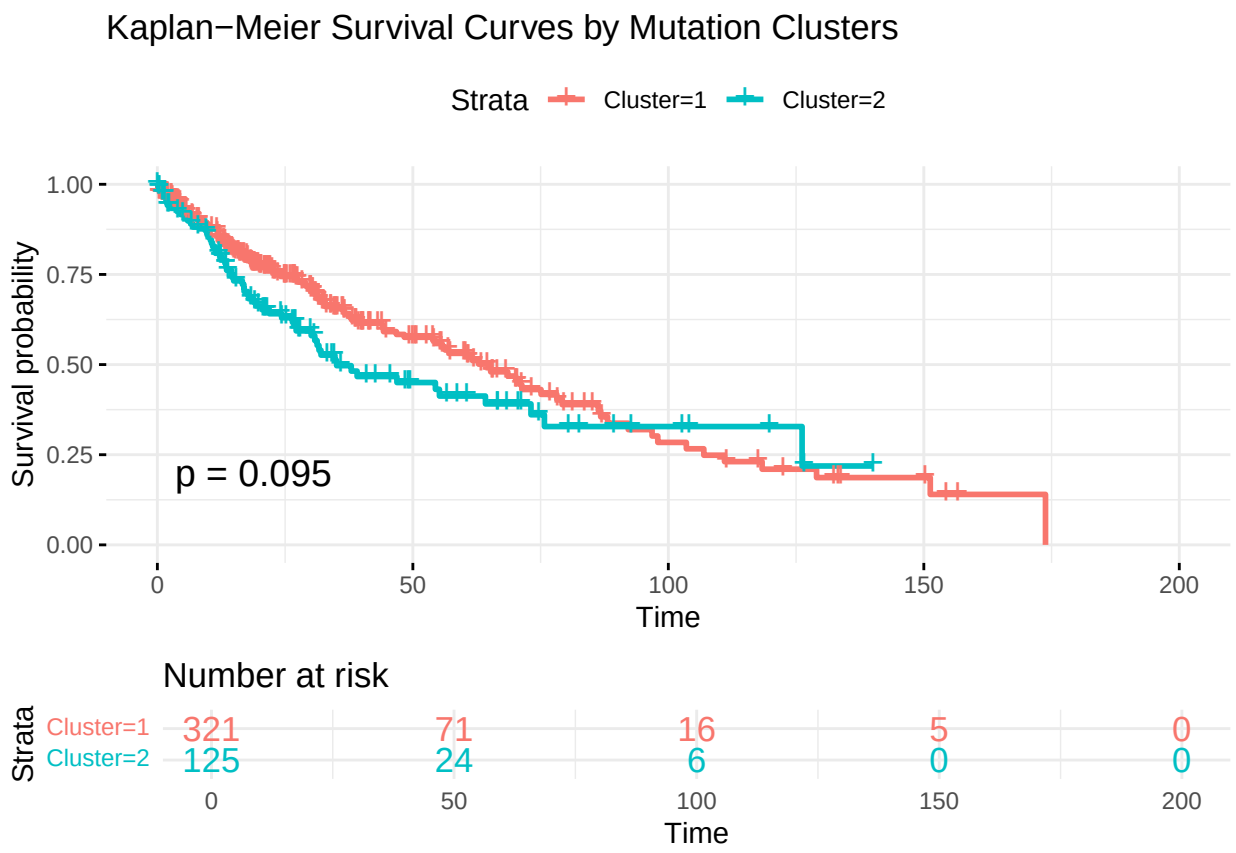
Survival Analysis on Mutation Data

```
# Assign clusters to patients based on the dendrogram
cluster_assignments <- cutree(hc_ward, k = 2)  # Cut tree into 2 clusters

# Create a data frame for cluster assignments
cluster_data <- data.frame(
  Tumor_Sample_Barcode = row.names(mutation_matrix_top_20),
  Cluster = as.factor(cluster_assignments)
)

# Remove "-01" suffix from Tumor_Sample_Barcode in cluster_data
cluster_data <- cluster_data %>%
  mutate(Tumor_Sample_Barcode = sub("-\\d+$", "", Tumor_Sample_Barcode))
```

```r
# Join the clinical_data with the cluster_data
clinical_data_with_clusters <- clinical_data %>%
  inner_join(cluster_data, by = c("PATIENT_ID" = "Tumor_Sample_Barcode"))

# Create a survival object
surv_object <- Surv(clinical_data_with_clusters$overall_survival,
                    clinical_data_with_clusters$deceased)

# Fit a survival model by cluster
fit_clusters <- survfit(surv_object ~ Cluster, data = clinical_data_with_clusters)

# Summary of survival model
print(fit_clusters)
```

```
## Call: survfit(formula = surv_object ~ Cluster, data = clinical_data_with_clusters)
##
##             n events median 0.95LCL 0.95UCL
## Cluster=1 321    126   64.9    54.4    79.2
## Cluster=2 125     59   35.1    30.1    75.7
```

```r
# Kaplan-Meier plot for survival by clusters
ggsurvplot(
  fit_clusters,
  data = clinical_data_with_clusters,
  pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  risk.table.height = 0.25,
  ggtheme = theme_minimal(),
  title = "Kaplan-Meier Survival Curves by Mutation Clusters"
)
```



Differential Expression Analysis

```
# Load expression data (rows: genes, columns: patients)
countData <- read.csv("RNAseq_LUSC.csv", row.names = 1)
```

```
# Process countData column names
colnames(countData) <- sapply(colnames(countData), function(x) {
  # Truncate to the first 15 characters
  truncated <- substr(x, 1, 15)
  # Replace '.' with '-' to match TCGA ID format
  gsub("\\.", "-", truncated)
})

# Filter sample IDs in clusters to retain only those present in countData
sample_ids_in_clusters <- patient_clusters$Tumor_Sample_Barcode
sample_ids_in_clusters <- sample_ids_in_clusters[sample_ids_in_clusters %in% colnames(countData)]

# Subset countData columns to match the filtered sample IDs
countData <- countData[, sample_ids_in_clusters, drop = FALSE]

# Prepare colData
colData <- DataFrame(Cluster = factor(patient_clusters$Cluster))  # Ensure Cluster is a factor
rownames(colData) <- patient_clusters$Tumor_Sample_Barcode

# Filter colData to retain only rows with IDs in sample_ids_in_clusters
colData <- colData[sample_ids_in_clusters, , drop = FALSE]
```

```r
# Ensure countData and colData are aligned
if (!identical(rownames(colData), colnames(countData))) {
  stop("colData row names and countData column names are not aligned!")
}

# Ensure countData is a numeric matrix
countData <- as.matrix(countData)
mode(countData) <- "numeric"
```

```r
# Create DESeq2 dataset
dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design = ~ Cluster)
```

```
## converting counts to integer mode
```

```r
dds = DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 8846 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

```r
dds
```

```
## class: DESeqDataSet
## dim: 60660 466
## metadata(1): version
## assays(6): counts mu ... replaceCounts replaceCooks
## rownames(60660): ENSG00000000003.15 ENSG00000000005.6 ...
##   ENSG00000288674.1 ENSG00000288675.1
## rowData names(27): baseMean baseVar ... maxCooks replace
## colnames(466): TCGA-18-3406-01 TCGA-18-3407-01 ... TCGA-O2-A5IB-01
##   TCGA-XC-AA0X-01
## colData names(3): Cluster sizeFactor replaceable
```

```
res <- results(dds)
res <- results(dds, contrast = c("Cluster", "3", "2"))
mcols(res, use.names = TRUE)
```

```
## DataFrame with 6 rows and 2 columns
##                         type            description
##                  <character>            <character>
## baseMean       intermediate mean of normalized c..
## log2FoldChange      results log2 fold change (ML..
## lfcSE               results standard error: Clus..
## stat                results Wald statistic: Clus..
## pvalue              results Wald test p-value: C..
## padj                results   BH adjusted p-values
```

```
summary(res)
```

```
##
## out of 57820 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1547, 2.7%
## LFC < 0 (down)     : 2327, 4%
## outliers [1]       : 0, 0%
## low counts [2]     : 21250, 37%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
res.05 <- results(dds, alpha = 0.05)
table(res.05$padj < 0.05)
```

```
##
## FALSE  TRUE
## 34759  1817
```

```
resLFC1 <- results(dds, lfcThreshold=1)
table(resLFC1$padj < 0.1)
```

```
##
## FALSE  TRUE
## 42147    19
```

```
res.order <- res[order(res$pvalue),]
summary(res.order)
```

```
##
## out of 57820 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1547, 2.7%
## LFC < 0 (down)     : 2327, 4%
## outliers [1]       : 0, 0%
```

```
## low counts [2]     : 21250, 37%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
# How many adjusted p-values were less than 0.1?
sum(res.order$padj < 0.1, na.rm=TRUE)
```

```
## [1] 3874
```

```
# Multiple testing
sum(res$pvalue < 0.05, na.rm=TRUE)
```

```
## [1] 8386
```

```
sum(!is.na(res$pvalue))
```

```
## [1] 57826
```

```
sum(res$padj < 0.06, na.rm=TRUE)
```

```
## [1] 2472
```

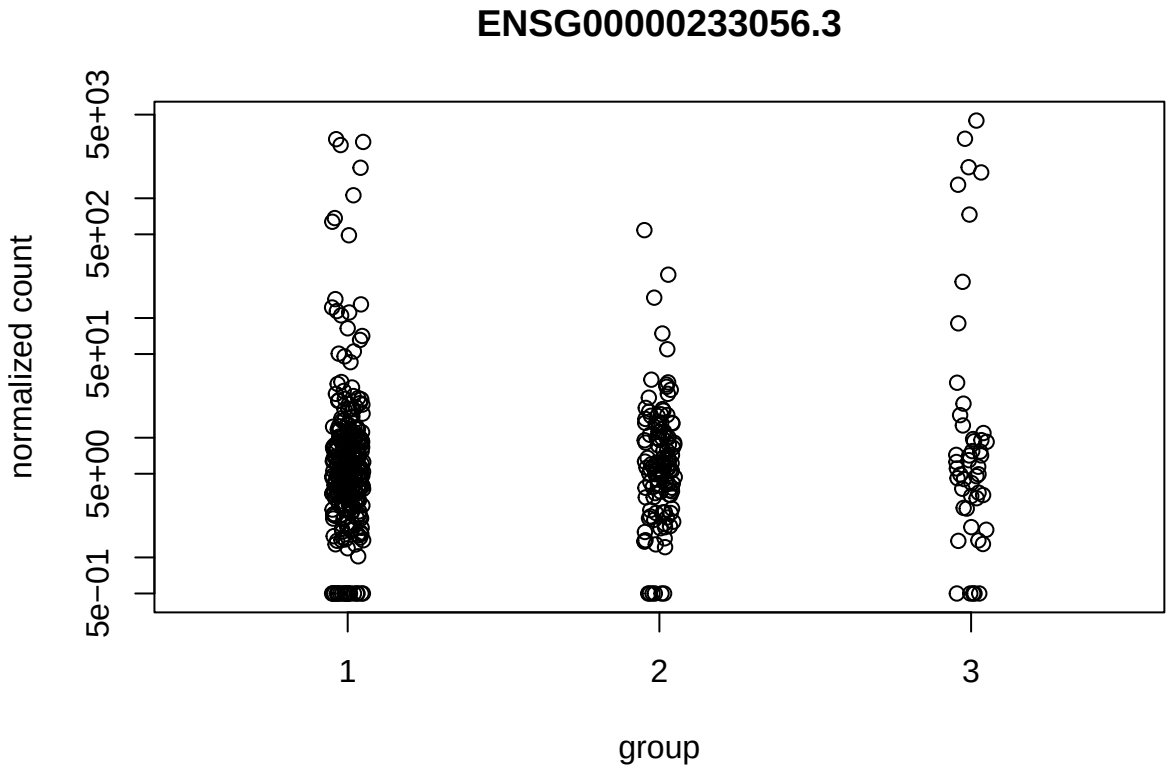
```
resSig <- subset(res, padj < 0.06)
head(resSig[ order( resSig$log2FoldChange ), ])
```

```
## log2 fold change (MLE): Cluster 3 vs 2
## Wald test p-value: Cluster 3 vs 2
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000171209.3   28.13101       -5.46788  0.970129  -5.63624 1.73805e-08
## ENSG00000126545.14   4.84703       -4.79355  0.956149  -5.01339 5.34793e-07
## ENSG00000260073.2    5.65027       -4.53925  0.924305  -4.91099 9.06187e-07
## ENSG00000261166.1    3.48789       -4.06345  1.094384  -3.71301 2.04812e-04
## ENSG00000075388.4    5.46653       -3.96459  1.000707  -3.96179 7.43891e-05
## ENSG00000181195.11 450.49067       -3.84298  0.464456  -8.27416 1.29367e-16
##                          padj
##                     <numeric>
## ENSG00000171209.3  1.38198e-05
## ENSG00000126545.14 1.98885e-04
## ENSG00000260073.2  2.87914e-04
## ENSG00000261166.1  1.09042e-02
## ENSG00000075388.4  6.05981e-03
## ENSG00000181195.11 9.46342e-13
```

```
head(resSig[ order( resSig$log2FoldChange, decreasing=TRUE), ])
```

```
## log2 fold change (MLE): Cluster 3 vs 2
## Wald test p-value: Cluster 3 vs 2
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000007350.17 110.02126        4.60647  0.509914   9.03381 1.65797e-19
## ENSG00000257883.1    3.92968        4.24474  0.628500   6.75376 1.44057e-11
## ENSG00000187581.3    8.48493        4.16422  0.453764   9.17706 4.43026e-20
## ENSG00000233056.3   38.42814        4.13128  0.393399  10.50151 8.50125e-26
## ENSG00000184697.7   71.26858        4.10950  0.416704   9.86193 6.08672e-23
## ENSG00000173237.4    5.20095        3.49598  0.542737   6.44139 1.18385e-10
##                          padj
##                     <numeric>
## ENSG00000007350.17 1.51605e-15
## ENSG00000257883.1  4.05311e-08
## ENSG00000187581.3  5.40138e-16
## ENSG00000233056.3  3.10942e-21
## ENSG00000184697.7  1.11314e-18
## ENSG00000173237.4  2.45237e-07
```

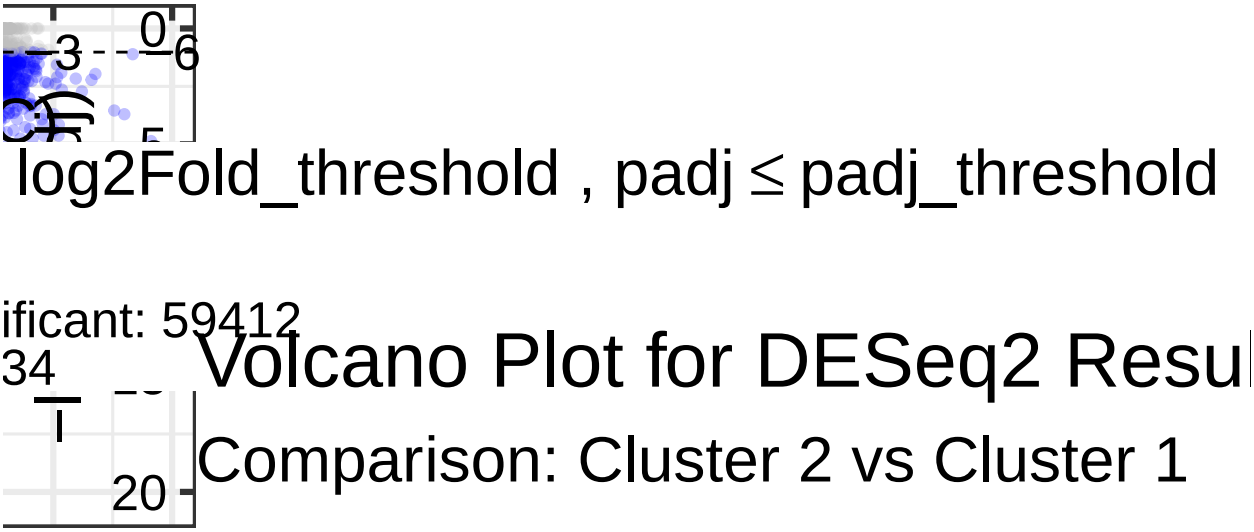
```
plotCounts(dds, gene=which.min(res$padj), intgroup="Cluster")
```



Volcano Plot!!!

```r
library(ggplot2)
library(ggrepel)

# Function to prepare DESeq2 results for plotting
prepare_res_for_plot <- function(res) {
  # Convert DESeqResults object to data.frame
  res_df <- as.data.frame(res)

  # Add rownames as a gene column (optional, adjust based on dataset)
  res_df$hgnc_symbol <- rownames(res_df)

  return(res_df)
}

# Prepare the results for the plot
res_df <- prepare_res_for_plot(res)

# Function to create the volcano plot
volcplot <- function(data, padj_threshold = 0.05, log2Fold_threshold = 1, plot_title = 'Volcano Plot', p
  # Set the fold-change thresholds
  neg_log2fc <- -log2Fold_threshold
  pos_log2fc <- log2Fold_threshold

  # Replace NA values
  data$padj[is.na(data$padj)] <- 1
  data$log2FoldChange[is.na(data$log2FoldChange)] <- 0

  # Add log2fc_threshold column
  data$log2fc_threshold <- ifelse(
    data$log2FoldChange >= pos_log2fc & data$padj <= padj_threshold, 'up',
    ifelse(data$log2FoldChange <= neg_log2fc & data$padj <= padj_threshold, 'down', 'ns')
  )

  # Count up, down, and unchanged genes
  up_genes <- sum(data$log2fc_threshold == 'up')
  down_genes <- sum(data$log2fc_threshold == 'down')
  unchanged_genes <- sum(data$log2fc_threshold == 'ns')

  # Generate legend labels
  legend_labels <- c(
    paste0('Up: ', up_genes),
    paste0('Not significant: ', unchanged_genes),
    paste0('Down: ', down_genes)
  )

  # Calculate x-axis limits
  x_axis_limits <- ceiling(max(abs(data$log2FoldChange)))

  # Define plot colors
  plot_colors <- c(
    'up' = 'red',
    'ns' = 'gray',
    'down' = 'blue'
```

```r
  )

  # Create the plot
  plot <- ggplot(data) +
    geom_point(
      aes(x = log2FoldChange, y = -log10(padj), color = log2fc_threshold),
      alpha = 0.25,
      size = 1.5
    ) +
    geom_vline(xintercept = c(neg_log2fc, pos_log2fc), linetype = 'dashed') +
    geom_hline(yintercept = -log10(padj_threshold), linetype = 'dashed') +
    scale_x_continuous(
      'log2(FC)',
      limits = c(-x_axis_limits, x_axis_limits)
    ) +
    scale_color_manual(
      values = plot_colors,
      labels = legend_labels
    ) +
    labs(
      color = expression(paste("log2Fold: ", log2Fold_threshold, " , padj" <= padj_threshold)),
      title = plot_title,
      subtitle = plot_subtitle
    ) +
    theme_bw(base_size = 24) +
    theme(
      aspect.ratio = 1,
      axis.text = element_text(color = 'black'),
      legend.margin = margin(0, 0, 0, 0),
      legend.box.margin = margin(0, 0, 0, 0),
      legend.spacing.x = unit(0.2, 'cm')
    )

  return(plot)
}

# Generate the volcano plot
volcano_plot <- volcplot(
  data = res_df,
  padj_threshold = 0.095,
  log2Fold_threshold = 1,
  plot_title = "Volcano Plot for DESeq2 Results",
  plot_subtitle = "Comparison: Cluster 2 vs Cluster 1"
)

# Display the plot
print(volcano_plot)
```



Pathway Analysis

```
# Load the required libraries
library(clusterProfiler)
```

```
##

## clusterProfiler v4.12.6 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## S Xu, E Hu, Y Cai, Z Xie, X Luo, L Zhan, W Tang, Q Wang, B Liu, R Wang,
## W Xie, T Wu, L Xie, G Yu. Using clusterProfiler to characterize
## multiomics data. Nature Protocols. 2024, doi:10.1038/s41596-024-01020-z

##
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:IRanges':
##
##     slice

## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```
## The following object is masked from 'package:purrr':
##
##     simplify

## The following object is masked from 'package:lattice':
##
##     dotplot

## The following object is masked from 'package:stats':
##
##     filter
```

```
library(org.Hs.eg.db)
```

```
## Loading required package: AnnotationDbi

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##     select

## The following object is masked from 'package:dplyr':
##
##     select

##
```

```
library(pathview)
```

```
## ##############################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ##############################################################################
```

```
library(enrichplot)
```

```
##
## Attaching package: 'enrichplot'

## The following object is masked from 'package:ggpubr':
##
##     color_palette
```

```
## The following object is masked from 'package:lattice':
##
##     dotplot
```

```
library(gage)
library(gageData)
```

```
# Filter significant genes
res_sig <- subset(res, padj < 0.05)

# Extract gene identifiers (ENSEMBL or ENTREZ)
sig_genes <- rownames(res_sig)
clean_sig_genes <- sub("\\.\\d+$", "", sig_genes)  # Clean version numbers from ENSEMBL IDs

# Map ENSEMBL IDs to ENTREZ IDs
entrez_genes <- mapIds(org.Hs.eg.db,
                       keys = clean_sig_genes,
                       keytype = "ENSEMBL",
                       column = "ENTREZID")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
entrez_genes <- na.omit(entrez_genes)  # Remove NA values
```

```
# Create a named vector of log2FoldChange values
fold_changes <- res_sig$log2FoldChange
names(fold_changes) <- entrez_genes
```

```
# Perform KEGG pathway enrichment analysis
enrich_kegg <- enrichKEGG(
  gene = entrez_genes,
  organism = "hsa",
  pvalueCutoff = 0.05
)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/hsa/pathway"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/hsa"...
```

```
# Load gage data for KEGG pathways
data(kegg.sets.hs)
data(sigmet.idx.hs)

# Focus on signaling and metabolic pathways
kegg.sets.hs <- kegg.sets.hs[sigmet.idx.hs]

# Run gage pathway analysis
keggres <- gage(fold_changes, gsets = kegg.sets.hs, same.dir = TRUE)
```

```r
# Top upregulated pathways
top_up <- head(keggres$greater, 5)

# Top downregulated pathways
top_down <- head(keggres$less, 5)

# Extract pathway IDs
upregulated_ids <- substr(rownames(top_up), start = 1, stop = 8)
downregulated_ids <- substr(rownames(top_down), start = 1, stop = 8)

upregulated_ids
```

```
## [1] "hsa04972" "hsa03010" "hsa04062" "hsa00230" "hsa04020"
```

```r
downregulated_ids
```

```
## [1] "hsa04810" "hsa04010" "hsa04510" "hsa04020" "hsa00230"
```

```r
# Visualize the first upregulated pathway
pathview(gene.data = fold_changes,
         pathway.id = upregulated_ids[1],
         species = "hsa")
```

```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa04972.pathview.png
```

```r
# Plot all upregulated pathways
for (path_id in upregulated_ids) {
  pathview(gene.data = fold_changes,
           pathway.id = path_id,
           species = "hsa")
}
```

```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa04972.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa03010.pathview.png

## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa04062.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa00230.pathview.png

## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)
## Warning in cbind(blk.ind, j): number of rows of result is not a multiple of
## vector length (arg 2)

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/chantellexu/Documents/RStudio/BMEG 310/Final Project

## Info: Writing image file hsa04020.pathview.png
```

```r
# Extract genes in the top pathway
pathway_genes <- enrich_kegg@result$geneID[1]
pathway_genes <- unlist(strsplit(pathway_genes, "/"))

# Filter fold changes for these genes
pathway_fc <- fold_changes[names(fold_changes) %in% pathway_genes]

# Create a heatmap
pheatmap(as.matrix(pathway_fc),
         cluster_rows = FALSE,
         cluster_cols = FALSE,
         color = colorRampPalette(c("blue", "white", "red"))(50),
         main = paste("Gene Expression in Pathway:", enrich_kegg@result$Description[1]))
```

Clinical Comparison of Mutation Clusters

```
# Preprocess to align IDs if necessary (example assumes PATIENT_ID requires suffix removal)
clinical_data <- clinical_data %>%
  mutate(Tumor_Sample_Barcode = gsub("-..$", "", PATIENT_ID))

patient_clusters <- patient_clusters %>%
  mutate(PATIENT_ID = gsub("-..$", "", Tumor_Sample_Barcode))

# Merge datasets
merged_data <- patient_clusters %>%
  inner_join(clinical_data, by = "PATIENT_ID")

merged_data
```

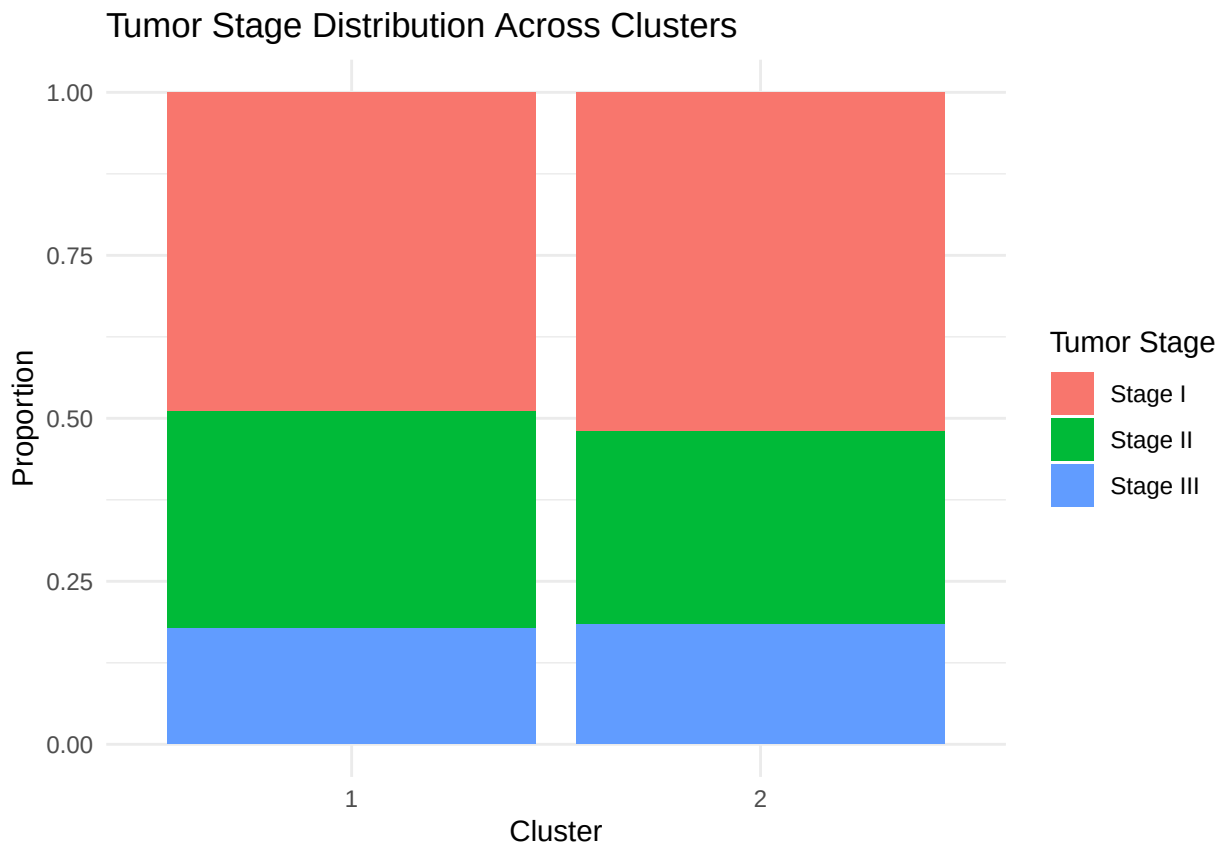
```
##      Tumor_Sample_Barcode.x Cluster   PATIENT_ID SUBTYPE CANCER_TYPE_ACRONYM
## 1           TCGA-18-3406-01       1 TCGA-18-3406    LUSC                LUSC
## 2           TCGA-18-3407-01       1 TCGA-18-3407    LUSC                LUSC
## 3           TCGA-18-3408-01       2 TCGA-18-3408    LUSC                LUSC
## 4           TCGA-18-3410-01       1 TCGA-18-3410    LUSC                LUSC
## 5           TCGA-18-3411-01       1 TCGA-18-3411    LUSC                LUSC
## 6           TCGA-18-3412-01       2 TCGA-18-3412    LUSC                LUSC
## 7           TCGA-18-3415-01       1 TCGA-18-3415    LUSC                LUSC
## 8           TCGA-18-3416-01       1 TCGA-18-3416    LUSC                LUSC
## 9           TCGA-18-3421-01       1 TCGA-18-3421    LUSC                LUSC
## 10          TCGA-18-4083-01       1 TCGA-18-4083    LUSC                LUSC
```

```r
# Subset the data for Cluster 1 and Cluster 2
merged_data <- merged_data[merged_data$Cluster %in% c(1, 2), ]
```

```r
# Create a contingency table for tumor stage and cluster
table_tumor_stage <- table(merged_data$tumor_stage, merged_data$Cluster)

# Perform Chi-squared test
chisq_test <- chisq.test(table_tumor_stage)

# View results
chisq_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  table_tumor_stage
## X-squared = 0.55715, df = 2, p-value = 0.7569
```

```r
# Create stacked bar plot for tumor stage by cluster
ggplot(merged_data, aes(x = as.factor(Cluster), fill = tumor_stage)) +
  geom_bar(position = "fill") +  # Proportional stacking
  theme_minimal() +
  labs(title = "Tumor Stage Distribution Across Clusters",
       x = "Cluster",
       y = "Proportion",
       fill = "Tumor Stage")
```

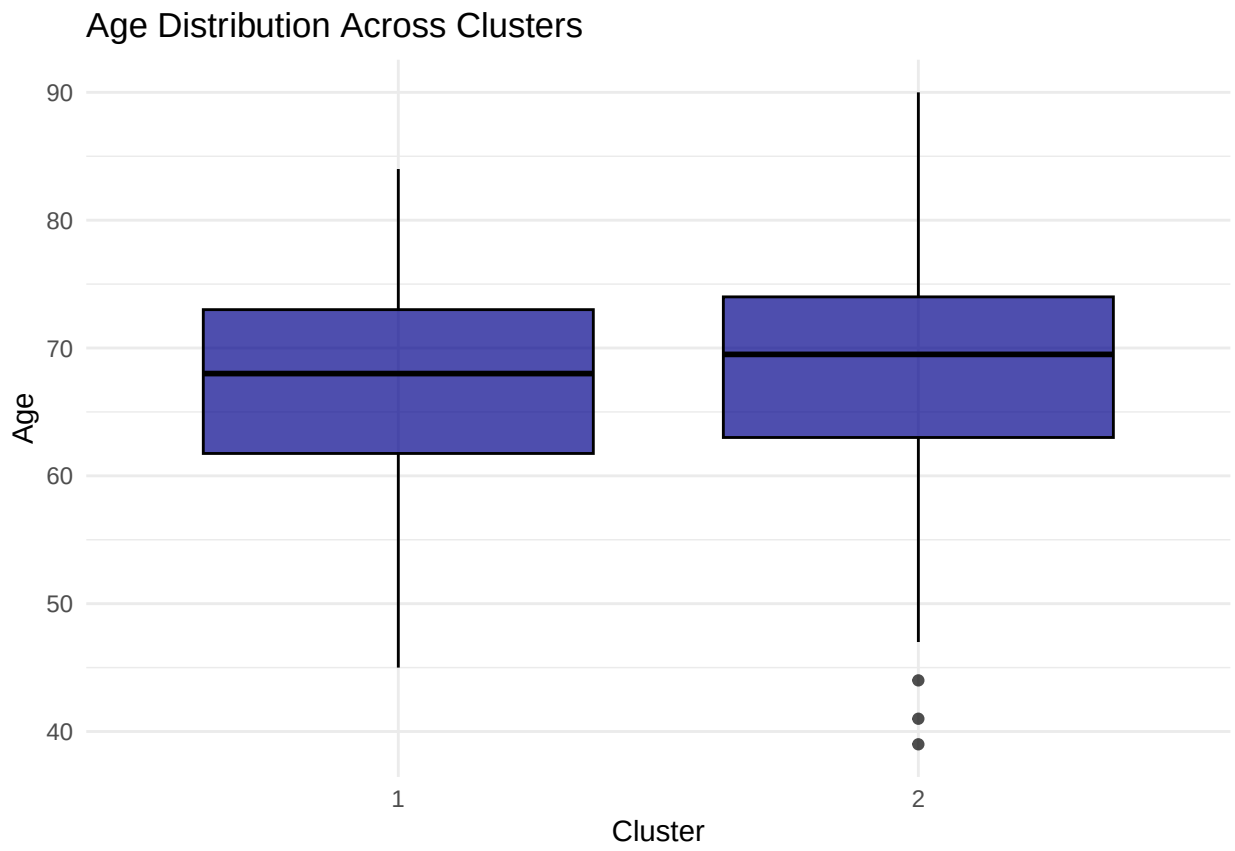



```r
# Perform ANOVA to test age differences across clusters
anova_age <- aov(AGE ~ Cluster, data = merged_data)

# View ANOVA results
summary(anova_age)
```

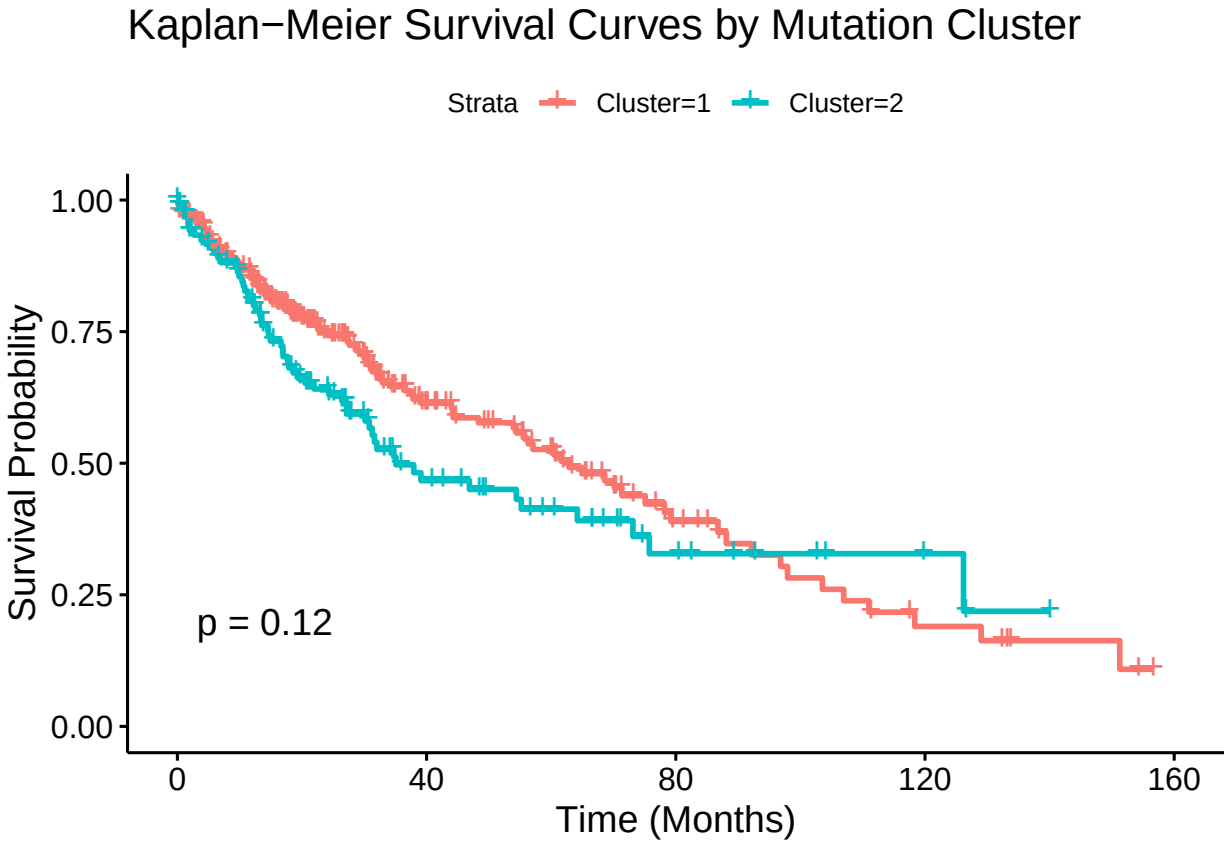
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Cluster       1    160  160.44    2.23  0.136
## Residuals   394  28345   71.94
## 5 observations deleted due to missingness
```

```r
# Create boxplot for age by cluster
ggplot(merged_data, aes(x = as.factor(Cluster), y = AGE)) +
  geom_boxplot(color = "black", fill = "darkblue", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Age Distribution Across Clusters",
       x = "Cluster",
       y = "Age")
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```r
# Create a survival object
surv_object <- Surv(merged_data$OS_MONTHS, merged_data$OS_STATUS == "1:DECEASED")

# Fit a Kaplan-Meier survival model by mutation cluster
km_fit <- survfit(surv_object ~ Cluster, data = merged_data)

# Plot the Kaplan-Meier survival curves
ggsurvplot(km_fit, data = merged_data, pval = TRUE,
           title = "Kaplan-Meier Survival Curves by Mutation Cluster",
           xlab = "Time (Months)", ylab = "Survival Probability")
```




```r
# Perform log-rank test to compare survival between clusters
surv_diff <- survdiff(surv_object ~ Cluster, data = merged_data)

# View results
surv_diff
```

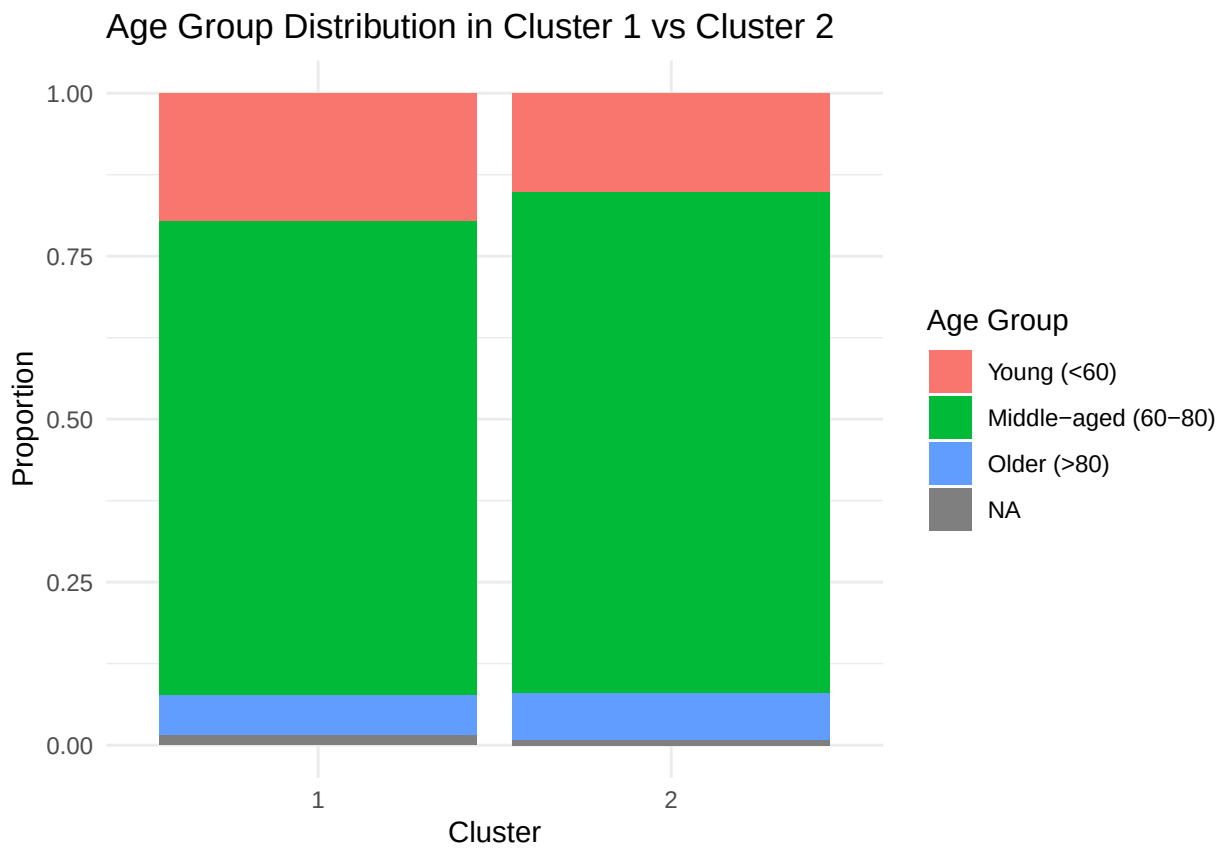
```
## Call:
## survdiff(formula = surv_object ~ Cluster, data = merged_data)
##
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## Cluster=1 276      110    119.2     0.703      2.39
## Cluster=2 125       59     49.8     1.680      2.39
##
##  Chisq= 2.4  on 1 degrees of freedom, p= 0.1
```

```r
# Create age categories
merged_data$age_group <- cut(merged_data$AGE,
                             breaks = c(0, 60, 80, Inf),
                             labels = c("Young (<60)", "Middle-aged (60-80)", "Older (>80)"),
                             right = FALSE)

# Chi-squared test for age group by cluster
table_age_group <- table(merged_data$age_group, merged_data$Cluster)
chisq_age_group <- chisq.test(table_age_group)
```

```r
# View result
chisq_age_group
```

```
##
##  Pearson's Chi-squared test
##
## data:  table_age_group
## X-squared = 1.221, df = 2, p-value = 0.5431
```

```r
# Bar plot to visualize the distribution of age groups across clusters
ggplot(merged_data, aes(x = as.factor(Cluster), fill = age_group)) +
  geom_bar(position = "fill") +
  theme_minimal() +
  labs(title = "Age Group Distribution in Cluster 1 vs Cluster 2",
       x = "Cluster", y = "Proportion", fill = "Age Group")
```




```r
# Mann-Whitney U test for age comparison
wilcox_test_age <- wilcox.test(AGE ~ Cluster, data = merged_data)

# View result
wilcox_test_age
```

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  AGE by Cluster
## W = 15060, p-value = 0.08743
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Violin plot to visualize the distribution of age across clusters
ggplot(merged_data, aes(x = as.factor(Cluster), y = AGE, fill = as.factor(Cluster))) +
  geom_violin() +
  theme_minimal() +
  labs(title = "Age Distribution Across Clusters",
       x = "Cluster", y = "Age")
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (`stat_ydensity()`).
```