Report on Survival analysis for Invasive Breast Cancer

Dr. Nishat Fatima Mohammad

06/05/2024

Table of Contents

## 1. Introduction.

Invasive breast cancer poses a significant global health concern and is a type of cancer that has spread from the original ducts or lobules to surrounding breast tissue. The factors influencing survival rates in patients with invasive breast cancer is crucial to transform treatment strategies and patient outcomes.

This Report will go into Survival analysis of Invasive Breast cancer using RNA-Seq and clinical data obtained from cbio Portal.

Survival analysis plays a pivotal role in biomedical research, particularly in oncology, where understanding the factors influencing patient outcomes is of paramount importance.

In this report, the application of survival analysis techniques is delved into to investigate the impact of gene expression patterns on patient survival in cancer cohorts.

The primary objective was to explore how variations in gene expression levels may correlate with differences in patient survival times.

To achieve this, a combination of Cox Proportional Hazards (CoxPH) modeling and Kaplan-Meier (KM) survival analysis were employed, coupled with clustering techniques. By examining these relationships, potential biomarkers and prognostic indicators that could aid in patient stratification and clinical decision-making can be uncovered.

The analysis will be carried out in R and on HPCC and published in an RMD notebook rendered to a word document.

This report outlines methodology, presents findings, and discusses the implications of results for future research. This analysis will endeavor to contribute to the growing body of knowledge aimed at improving patient outcomes in oncology.

## 2. Load packages.

The packages in R that will be used are loaded below.

library(readr)  
library(dplyr)  
library(tidyr)  
library(tibble)  
library(survival)  
library(factoextra)  
library(cluster)  
library(NbClust)  
library(gridExtra)  
library(ggplot2)  
library(Rtsne)  
library(umap)

## 3. Load the data.

# Path to RNA Seq expression file  
file\_path\_expr <- "../data/data\_mrna\_seq\_v2\_rsem.txt"  
  
# Path to clinical data  
file\_path\_clin <- "../data/data\_clinical\_patient.txt"  
  
# Load expression data  
data\_expression <- read\_delim(file\_path\_expr, delim = "\t")  
# View expression data  
#head(data\_expression)  
  
# Load clinical data  
data\_clinical <- read\_delim(file\_path\_clin, delim = "\t")  
# View clinical data  
#head(data\_clinical)

The RNA Seq expression data .txt file and corresponding clinical data .txt file were downloaded from cbio portal and loaded into the respective variables. Let us move on to explore the data.

## 4. Data Exploration.

### 4.1. RNA Seq Data exploration.

# Look at the first 10 rows and columns  
data\_expression[1:10,1:10]

## # A tibble: 10 × 10  
## Hugo\_Symbol Entrez\_Gene\_Id `TCGA-3C-AAAU-01` `TCGA-3C-AALI-01`  
## <chr> <dbl> <dbl> <dbl>  
## 1 <NA> 100130426 0 0   
## 2 <NA> 100133144 16.4 9.27   
## 3 UBE2Q2P2 100134869 12.9 17.4   
## 4 HMGB1P1 10357 52.2 69.8   
## 5 <NA> 10431 408. 564.   
## 6 <NA> 136542 0 0   
## 7 <NA> 155060 1187. 516.   
## 8 RNU12-2P 26823 0 1.09   
## 9 SSX9P 280660 0 0.544  
## 10 <NA> 317712 0 0   
## # ℹ 6 more variables: `TCGA-3C-AALJ-01` <dbl>, `TCGA-3C-AALK-01` <dbl>,  
## # `TCGA-4H-AAAK-01` <dbl>, `TCGA-5L-AAT0-01` <dbl>, `TCGA-5T-A9QA-01` <dbl>,  
## # `TCGA-A1-A0SB-01` <dbl>

dimensions\_data\_expression <- dim(data\_expression)  
  
# Missing values exploration  
any(is.na(data\_expression))

## [1] TRUE

# Missing values in other columns apart from first  
any(is.na(data\_expression[,-1]))

## [1] FALSE

sum(is.na(data\_expression$Hugo\_Symbol))

## [1] 13

# Zero expression values  
num\_rows\_with\_zeros <- sum(apply(data\_expression[,3:ncol(data\_expression)], 1, function(row) any(row == 0)))  
  
# Get the rows with constant values for all samples  
constant\_rows <- apply(data\_expression[,3:ncol(data\_expression)], 1, function(x) length(unique(x)) == 1)  
num\_constant\_rows <- sum(constant\_rows==TRUE)

The RNA Seq Data has dimensions 20531, 1084, the first column is Hugo\_Symbol and the second is Entrez\_Gene\_Id, the rest are sample Ids. The Hugo\_symbol is missing 13 gene symbols. I may consider removing these rows during data cleaning.  
There are a total of 7760 rows that have zero expression values.These may or may not be missing values.  
To solve that dilemma I checked the rows with constant expression values. There are a total of 282 rows that have all expression values constant which is biologically impossible and will be considered as artifacts sourced at some point during the experiment and will be removed these rows for ease of analysis.

### 4.2. Clinical Data Exploration.

# First 10 rows and columns  
data\_clinical[1:10.1:10]

## Warning in 1:10.1:10: numerical expression has 10 elements: only the first used

## # A tibble: 1,088 × 10  
## `#Patient Identifier` Subtype TCGA PanCanAtlas Can…¹ `Other Patient ID`  
## <chr> <chr> <chr> <chr>   
## 1 #Identifier to uniquely sp… Subtype Text field to hold ca… Legacy DMP patien…  
## 2 #STRING STRING STRING STRING   
## 3 #1 1 1 1   
## 4 PATIENT\_ID SUBTYPE CANCER\_TYPE\_ACRONYM OTHER\_PATIENT\_ID   
## 5 TCGA-3C-AAAU BRCA\_L… BRCA 6E7D5EC6-A469-467…  
## 6 TCGA-3C-AALI BRCA\_H… BRCA 55262FCB-1B01-448…  
## 7 TCGA-3C-AALJ BRCA\_L… BRCA 427D0648-3F77-4FF…  
## 8 TCGA-3C-AALK BRCA\_L… BRCA C31900A4-5DCD-402…  
## 9 TCGA-4H-AAAK BRCA\_L… BRCA 6623FC5E-00BE-447…  
## 10 TCGA-5L-AAT0 BRCA\_L… BRCA 86C6F993-327F-452…  
## # ℹ 1,078 more rows  
## # ℹ abbreviated name: ¹​`TCGA PanCanAtlas Cancer Type Acronym`  
## # ℹ 6 more variables: `Diagnosis Age` <chr>, Sex <chr>,  
## # `Neoplasm Disease Stage American Joint Committee on Cancer Code` <chr>,  
## # `American Joint Committee on Cancer Publication Version Type` <chr>,  
## # `Last Communication Contact from Initial Pathologic Diagnosis Date` <chr>,  
## # `Birth from Initial Pathologic Diagnosis Date` <chr>

# Take of the first 4 rows of the data  
data\_clinical2 <- data\_clinical[5:nrow(data\_clinical),]  
  
# explore the data  
str(data\_clinical2)

## tibble [1,084 × 38] (S3: tbl\_df/tbl/data.frame)  
## $ #Patient Identifier : chr [1:1084] "TCGA-3C-AAAU" "TCGA-3C-AALI" "TCGA-3C-AALJ" "TCGA-3C-AALK" ...  
## $ Subtype : chr [1:1084] "BRCA\_LumA" "BRCA\_Her2" "BRCA\_LumB" "BRCA\_LumA" ...  
## $ TCGA PanCanAtlas Cancer Type Acronym : chr [1:1084] "BRCA" "BRCA" "BRCA" "BRCA" ...  
## $ Other Patient ID : chr [1:1084] "6E7D5EC6-A469-467C-B748-237353C23416" "55262FCB-1B01-4480-B322-36570430C917" "427D0648-3F77-4FFC-B52C-89855426D647" "C31900A4-5DCD-4022-97AC-638E86E889E4" ...  
## $ Diagnosis Age : chr [1:1084] "55" "50" "62" "52" ...  
## $ Sex : chr [1:1084] "Female" "Female" "Female" "Female" ...  
## $ Neoplasm Disease Stage American Joint Committee on Cancer Code : chr [1:1084] "STAGE X" "STAGE IIB" "STAGE IIB" "STAGE IA" ...  
## $ American Joint Committee on Cancer Publication Version Type : chr [1:1084] "6TH" "6TH" "7TH" "7TH" ...  
## $ Last Communication Contact from Initial Pathologic Diagnosis Date : chr [1:1084] "4047" "4005" "1474" "1448" ...  
## $ Birth from Initial Pathologic Diagnosis Date : chr [1:1084] "-20211" "-18538" "-22848" "-19074" ...  
## $ Last Alive Less Initial Pathologic Diagnosis Date Calculated Day Value : chr [1:1084] "0" "0" "0" "0" ...  
## $ Ethnicity Category : chr [1:1084] "Not Hispanic Or Latino" "Not Hispanic Or Latino" "Not Hispanic Or Latino" "Not Hispanic Or Latino" ...  
## $ Form completion date : chr [1:1084] "1/13/14" "7/28/14" "7/28/14" "7/28/14" ...  
## $ Neoadjuvant Therapy Type Administered Prior To Resection Text : chr [1:1084] "No" "No" "No" "No" ...  
## $ ICD-10 Classification : chr [1:1084] "C50.9" "C50.9" "C50.9" "C50.9" ...  
## $ International Classification of Diseases for Oncology, Third Edition ICD-O-3 Histology Code: chr [1:1084] "8520/3" "8500/3" "8500/3" "8500/3" ...  
## $ International Classification of Diseases for Oncology, Third Edition ICD-O-3 Site Code : chr [1:1084] "C50.9" "C50.9" "C50.9" "C50.9" ...  
## $ Informed consent verified : chr [1:1084] "Yes" "Yes" "Yes" "Yes" ...  
## $ New Neoplasm Event Post Initial Therapy Indicator : chr [1:1084] "No" "No" "No" "No" ...  
## $ American Joint Committee on Cancer Metastasis Stage Code : chr [1:1084] "MX" "M0" "M0" "M0" ...  
## $ Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code : chr [1:1084] "NX" "N1A" "N1A" "N0 (I+)" ...  
## $ American Joint Committee on Cancer Tumor Stage Code : chr [1:1084] "TX" "T2" "T2" "T1C" ...  
## $ Person Neoplasm Cancer Status : chr [1:1084] "With Tumor" "Tumor Free" "Tumor Free" "Tumor Free" ...  
## $ Primary Lymph Node Presentation Assessment : chr [1:1084] "Yes" "Yes" "Yes" "Yes" ...  
## $ Prior Diagnosis : chr [1:1084] "No" "No" "No" "No" ...  
## $ Race Category : chr [1:1084] "White" "Black or African American" "Black or African American" "Black or African American" ...  
## $ Radiation Therapy : chr [1:1084] "No" "Yes" "No" "No" ...  
## $ Patient Weight : chr [1:1084] NA NA NA NA ...  
## $ In PanCan Pathway Analysis : chr [1:1084] "Yes" "Yes" "Yes" "Yes" ...  
## $ Overall Survival Status : chr [1:1084] "0:LIVING" "0:LIVING" "0:LIVING" "0:LIVING" ...  
## $ Overall Survival (Months) : chr [1:1084] "133.0505967" "131.6697899" "48.45974291" "47.60495775" ...  
## $ Disease-specific Survival status : chr [1:1084] "0:ALIVE OR DEAD TUMOR FREE" "0:ALIVE OR DEAD TUMOR FREE" "0:ALIVE OR DEAD TUMOR FREE" "0:ALIVE OR DEAD TUMOR FREE" ...  
## $ Months of disease-specific survival : chr [1:1084] "133.0505967" "131.6697899" "48.45974291" "47.60495775" ...  
## $ Disease Free Status : chr [1:1084] "1:Recurred/Progressed" "0:DiseaseFree" "0:DiseaseFree" NA ...  
## $ Disease Free (Months) : chr [1:1084] "59.44044449" "131.6697899" "48.45974291" NA ...  
## $ Progression Free Status : chr [1:1084] "1:PROGRESSION" "0:CENSORED" "0:CENSORED" "0:CENSORED" ...  
## $ Progress Free Survival (Months) : chr [1:1084] "59.44044449" "131.6697899" "48.45974291" "47.60495775" ...  
## $ Genetic Ancestry Label : chr [1:1084] "EUR" "AFR" "AFR\_ADMIX" "AFR" ...

glimpse(data\_clinical2)

## Rows: 1,084  
## Columns: 38  
## $ `#Patient Identifier` <chr> …  
## $ Subtype <chr> …  
## $ `TCGA PanCanAtlas Cancer Type Acronym` <chr> …  
## $ `Other Patient ID` <chr> …  
## $ `Diagnosis Age` <chr> …  
## $ Sex <chr> …  
## $ `Neoplasm Disease Stage American Joint Committee on Cancer Code` <chr> …  
## $ `American Joint Committee on Cancer Publication Version Type` <chr> …  
## $ `Last Communication Contact from Initial Pathologic Diagnosis Date` <chr> …  
## $ `Birth from Initial Pathologic Diagnosis Date` <chr> …  
## $ `Last Alive Less Initial Pathologic Diagnosis Date Calculated Day Value` <chr> …  
## $ `Ethnicity Category` <chr> …  
## $ `Form completion date` <chr> …  
## $ `Neoadjuvant Therapy Type Administered Prior To Resection Text` <chr> …  
## $ `ICD-10 Classification` <chr> …  
## $ `International Classification of Diseases for Oncology, Third Edition ICD-O-3 Histology Code` <chr> …  
## $ `International Classification of Diseases for Oncology, Third Edition ICD-O-3 Site Code` <chr> …  
## $ `Informed consent verified` <chr> …  
## $ `New Neoplasm Event Post Initial Therapy Indicator` <chr> …  
## $ `American Joint Committee on Cancer Metastasis Stage Code` <chr> …  
## $ `Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code` <chr> …  
## $ `American Joint Committee on Cancer Tumor Stage Code` <chr> …  
## $ `Person Neoplasm Cancer Status` <chr> …  
## $ `Primary Lymph Node Presentation Assessment` <chr> …  
## $ `Prior Diagnosis` <chr> …  
## $ `Race Category` <chr> …  
## $ `Radiation Therapy` <chr> …  
## $ `Patient Weight` <chr> …  
## $ `In PanCan Pathway Analysis` <chr> …  
## $ `Overall Survival Status` <chr> …  
## $ `Overall Survival (Months)` <chr> …  
## $ `Disease-specific Survival status` <chr> …  
## $ `Months of disease-specific survival` <chr> …  
## $ `Disease Free Status` <chr> …  
## $ `Disease Free (Months)` <chr> …  
## $ `Progression Free Status` <chr> …  
## $ `Progress Free Survival (Months)` <chr> …  
## $ `Genetic Ancestry Label` <chr> …

dimensions\_data\_clinical2 <- dim(data\_clinical2)  
#summary(data\_clinical2)  
  
# Missing values  
any(is.na(data\_clinical2))

## [1] TRUE

data\_clinical\_na <- sum(is.na(data\_clinical2))  
data\_clinical\_na\_cols <- colSums(is.na(data\_clinical2))  
  
# Survival and Event features  
unique(data\_clinical2$`Overall Survival Status`)

## [1] "0:LIVING" "1:DECEASED"

The Clinical data has dimensions of 1084, 38, the first 4 rows are short notes and the actual data begins in row 5, the 4th row has the appropriate column names, and will be used for that purpose in data cleaning section. There are a total of 2809 which can be found in the following columns:  
0, 103, 0, 0, 0, 0, 5, 140, 104, 15, 0, 169, 0, 1, 0, 0, 0, 0, 199, 0, 0, 0, 123, 364, 1, 90, 101, 1084, 0, 0, 0, 20, 2, 142, 143, 1, 2, 0.

For this analysis, the survival data is in the last 9 features and will be extracted from the clinical data

## 5. Data Preprocessing.

### 5.1. Clinical Data Preprocessing.

# Fix Column names  
colnames(data\_clinical2) <- data\_clinical[4,]  
colnames(data\_clinical2)

## [1] "PATIENT\_ID"   
## [2] "SUBTYPE"   
## [3] "CANCER\_TYPE\_ACRONYM"   
## [4] "OTHER\_PATIENT\_ID"   
## [5] "AGE"   
## [6] "SEX"   
## [7] "AJCC\_PATHOLOGIC\_TUMOR\_STAGE"   
## [8] "AJCC\_STAGING\_EDITION"   
## [9] "DAYS\_LAST\_FOLLOWUP"   
## [10] "DAYS\_TO\_BIRTH"   
## [11] "DAYS\_TO\_INITIAL\_PATHOLOGIC\_DIAGNOSIS"   
## [12] "ETHNICITY"   
## [13] "FORM\_COMPLETION\_DATE"   
## [14] "HISTORY\_NEOADJUVANT\_TRTYN"   
## [15] "ICD\_10"   
## [16] "ICD\_O\_3\_HISTOLOGY"   
## [17] "ICD\_O\_3\_SITE"   
## [18] "INFORMED\_CONSENT\_VERIFIED"   
## [19] "NEW\_TUMOR\_EVENT\_AFTER\_INITIAL\_TREATMENT"   
## [20] "PATH\_M\_STAGE"   
## [21] "PATH\_N\_STAGE"   
## [22] "PATH\_T\_STAGE"   
## [23] "PERSON\_NEOPLASM\_CANCER\_STATUS"   
## [24] "PRIMARY\_LYMPH\_NODE\_PRESENTATION\_ASSESSMENT"  
## [25] "PRIOR\_DX"   
## [26] "RACE"   
## [27] "RADIATION\_THERAPY"   
## [28] "WEIGHT"   
## [29] "IN\_PANCANPATHWAYS\_FREEZE"   
## [30] "OS\_STATUS"   
## [31] "OS\_MONTHS"   
## [32] "DSS\_STATUS"   
## [33] "DSS\_MONTHS"   
## [34] "DFS\_STATUS"   
## [35] "DFS\_MONTHS"   
## [36] "PFS\_STATUS"   
## [37] "PFS\_MONTHS"   
## [38] "GENETIC\_ANCESTRY\_LABEL"

# Get Survival data  
data\_survival <- data.frame(data\_clinical2[,1], data\_clinical2[,5:6], data\_clinical2[, (ncol(data\_clinical2)-8):(ncol(data\_clinical2)-1)])  
  
# Categorical features  
cat\_fx <- data.frame(data\_survival$SEX,data\_survival$OS\_STATUS, data\_survival$DSS\_STATUS, data\_survival$DFS\_STATUS, data\_survival$PFS\_STATUS)  
  
# Look at the unique values for each column  
get\_uniq\_vals <- function(data) {  
 uniq\_vals <- lapply(data, unique)  
 return(uniq\_vals)  
}  
unique\_survival\_values <- get\_uniq\_vals(cat\_fx)  
  
# Handle Missing values for categorical features  
data\_survival\_cleaned <- data\_survival %>%  
 filter(!is.na(DFS\_STATUS), !is.na(DSS\_STATUS))  
colSums(is.na(data\_survival\_cleaned))

## PATIENT\_ID AGE SEX OS\_STATUS OS\_MONTHS DSS\_STATUS DSS\_MONTHS   
## 0 0 0 0 0 0 1   
## DFS\_STATUS DFS\_MONTHS PFS\_STATUS PFS\_MONTHS   
## 0 1 0 1

# Binary encode categorical features  
data\_survival\_cleaned$SEX <- as.factor(ifelse(data\_survival\_cleaned$SEX == "Female", 1, 0))  
data\_survival\_cleaned$OS\_STATUS <- as.factor(ifelse(data\_survival\_cleaned$OS\_STATUS == "1:DECEASED", 1, 0))  
data\_survival\_cleaned$DSS\_STATUS <- as.factor(ifelse(data\_survival\_cleaned$DSS\_STATUS == "1:DEAD WITH TUMOR", 1, 0))  
data\_survival\_cleaned$DFS\_STATUS <- as.factor(ifelse(data\_survival\_cleaned$DFS\_STATUS == "1:Recurred/Progressed", 1, 0))  
data\_survival\_cleaned$PFS\_STATUS <- as.factor(ifelse(data\_survival\_cleaned$PFS\_STATUS == "1:PROGRESSION", 1, 0))  
  
# Change factored categorical variables to numeric  
data\_survival\_cleaned$SEX <-as.numeric(data\_survival\_cleaned$SEX)  
data\_survival\_cleaned$OS\_STATUS <- as.numeric(data\_survival\_cleaned$OS\_STATUS)  
data\_survival\_cleaned$DSS\_STATUS <- as.numeric(data\_survival\_cleaned$DSS\_STATUS)  
data\_survival\_cleaned$DFS\_STATUS <- as.numeric(data\_survival\_cleaned$DFS\_STATUS)  
data\_survival\_cleaned$PFS\_STATUS <- as.numeric(data\_survival\_cleaned$PFS\_STATUS)  
  
# Change continuous variables to numeric class  
cont\_vars <- c("AGE", "OS\_MONTHS", "DSS\_MONTHS", "DFS\_MONTHS", "PFS\_MONTHS")  
data\_survival\_cleaned[cont\_vars] <- lapply(data\_survival\_cleaned[cont\_vars], as.numeric)  
  
# Impute mean for NA in continuous variables   
cont\_var\_na <- c("DSS\_MONTHS", "PFS\_MONTHS", "DFS\_MONTHS")  
# Get Mean for each cont column  
means <- sapply(data\_survival\_cleaned[cont\_var\_na], function(x) mean(as.numeric(x), na.rm = TRUE))  
# Impute here  
for(var in cont\_var\_na) {  
 data\_survival\_cleaned[[var]][is.na(data\_survival\_cleaned[[var]])] <- means[var]  
}  
  
# Explore after Clean up  
glimpse(data\_survival\_cleaned)

## Rows: 929  
## Columns: 11  
## $ PATIENT\_ID <chr> "TCGA-3C-AAAU", "TCGA-3C-AALI", "TCGA-3C-AALJ", "TCGA-4H-AA…  
## $ AGE <dbl> 55, 50, 62, 50, 70, 59, 56, 54, 61, 39, 52, 39, 77, 50, 67,…  
## $ SEX <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2,…  
## $ OS\_STATUS <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ OS\_MONTHS <dbl> 133.050597, 131.669790, 48.459743, 11.440971, 8.514975, 14.…  
## $ DSS\_STATUS <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ DSS\_MONTHS <dbl> 133.050597, 131.669790, 48.459743, 11.440971, 8.514975, 14.…  
## $ DFS\_STATUS <dbl> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ DFS\_MONTHS <dbl> 59.440444, 131.669790, 48.459743, 11.440971, 8.514975, 14.3…  
## $ PFS\_STATUS <dbl> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
## $ PFS\_MONTHS <dbl> 59.440444, 131.669790, 48.459743, 11.440971, 8.514975, 14.3…

summary(data\_survival\_cleaned)

## PATIENT\_ID AGE SEX OS\_STATUS   
## Length:929 Min. :26.00 Min. :1.000 Min. :1.00   
## Class :character 1st Qu.:49.00 1st Qu.:2.000 1st Qu.:1.00   
## Mode :character Median :58.00 Median :2.000 Median :1.00   
## Mean :58.32 Mean :1.988 Mean :1.08   
## 3rd Qu.:67.00 3rd Qu.:2.000 3rd Qu.:1.00   
## Max. :90.00 Max. :2.000 Max. :2.00   
## OS\_MONTHS DSS\_STATUS DSS\_MONTHS DFS\_STATUS   
## Min. : 0.00 Min. :1.000 Min. : 0.00 Min. :1.000   
## 1st Qu.: 14.73 1st Qu.:1.000 1st Qu.: 14.73 1st Qu.:1.000   
## Median : 25.94 Median :1.000 Median : 25.94 Median :1.000   
## Mean : 40.37 Mean :1.043 Mean : 40.38 Mean :1.086   
## 3rd Qu.: 53.62 3rd Qu.:1.000 3rd Qu.: 53.62 3rd Qu.:1.000   
## Max. :282.90 Max. :2.000 Max. :282.90 Max. :2.000   
## DFS\_MONTHS PFS\_STATUS PFS\_MONTHS   
## Min. : 0.00 Min. :1.000 Min. : 0.00   
## 1st Qu.: 14.24 1st Qu.:1.000 1st Qu.: 14.24   
## Median : 24.79 Median :1.000 Median : 24.79   
## Mean : 37.74 Mean :1.096 Mean : 37.51   
## 3rd Qu.: 50.79 3rd Qu.:1.000 3rd Qu.: 50.73   
## Max. :281.29 Max. :2.000 Max. :281.29

str(data\_survival\_cleaned)

## 'data.frame': 929 obs. of 11 variables:  
## $ PATIENT\_ID: chr "TCGA-3C-AAAU" "TCGA-3C-AALI" "TCGA-3C-AALJ" "TCGA-4H-AAAK" ...  
## $ AGE : num 55 50 62 50 70 59 56 54 61 39 ...  
## $ SEX : num 2 2 2 2 2 2 2 2 2 2 ...  
## $ OS\_STATUS : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ OS\_MONTHS : num 133.05 131.67 48.46 11.44 8.51 ...  
## $ DSS\_STATUS: num 1 1 1 1 1 1 1 1 1 1 ...  
## $ DSS\_MONTHS: num 133.05 131.67 48.46 11.44 8.51 ...  
## $ DFS\_STATUS: num 2 1 1 1 1 1 1 1 1 1 ...  
## $ DFS\_MONTHS: num 59.44 131.67 48.46 11.44 8.51 ...  
## $ PFS\_STATUS: num 2 1 1 1 1 1 1 1 1 1 ...  
## $ PFS\_MONTHS: num 59.44 131.67 48.46 11.44 8.51 ...

any(is.na(data\_survival\_cleaned))

## [1] FALSE

In the above chunk of code, the column names were tidied. Missing values removed form categorical variables, and imputed with mean for the continuous variables. Binary encoding was carried out for the categorical variables. The variables were converted to numeric type. The cleaned data has been explored above.

### 5.2. Expression Data Presprocessing.

# Remove rows with NA  
data\_expression\_filtered <- data\_expression %>% drop\_na()  
  
# Transpose columns to rows  
data\_expression\_t <- as.data.frame(t(data\_expression\_filtered))  
  
# Fix column names  
colnames(data\_expression\_t) <- data\_expression\_t[1,]  
data\_expression\_t <- data\_expression\_t[-1,]  
  
# Fix row names to omit -01 ending  
row.names(data\_expression\_t) <- gsub("-01$", "", row.names(data\_expression\_t))  
  
# Get the patient Ids  
patient\_ids <- data\_survival\_cleaned$PATIENT\_ID  
# Get all expression data for Pt ids  
data\_expression\_cleaned <- data\_expression\_t[rownames(data\_expression\_t) %in% patient\_ids, ]  
  
# Change class to numeric  
data\_expression\_cleaned\_num <- as.data.frame(lapply(data\_expression\_cleaned, as.numeric))  
  
# Remove columns with constant values for all patients as missing values  
constant\_cols <- apply(data\_expression\_cleaned\_num, 2, function(x) length(unique(x)) == 1)  
sum(constant\_cols==TRUE)

## [1] 290

data\_expression\_cleaned\_num <- data\_expression\_cleaned\_num[, !constant\_cols]  
  
# Log transformation  
data\_expression\_log <- log1p(data\_expression\_cleaned\_num)  
  
# Get PATIENT\_ID column  
data\_expression\_cleaned\_log <- data.frame(PATIENT\_ID = rownames(data\_expression\_cleaned), data\_expression\_log, row.names = NULL)  
  
# Explore after clean up  
dim(data\_expression\_cleaned\_log)

## [1] 927 20229

data\_expression\_cleaned\_log[1:10,1:10]

## PATIENT\_ID UBE2Q2P2 HMGB1P1 RNU12.2P SSX9P EZHIP EFCAB8  
## 1 TCGA-3C-AAAU 2.634160 3.973124 0.0000000 0.0000000 1.0018444 0.2961709  
## 2 TCGA-3C-AALI 2.911209 4.259227 0.7359672 0.4342469 4.9774232 1.7739349  
## 3 TCGA-3C-AALJ 2.325266 5.045339 0.0000000 0.0000000 0.0000000 1.3136968  
## 4 TCGA-4H-AAAK 2.736301 4.445152 0.3545226 0.0000000 0.3545226 1.3809551  
## 5 TCGA-A1-A0SB 2.733945 4.765476 0.3719083 0.0000000 0.6423801 0.6423801  
## 6 TCGA-A1-A0SD 2.511557 4.115176 0.2857803 0.0000000 0.5077810 0.2857803  
## 7 TCGA-A1-A0SE 1.694257 5.037894 0.0000000 0.0000000 0.0000000 0.6867266  
## 8 TCGA-A1-A0SF 2.463010 4.957185 0.0000000 0.0000000 0.0000000 1.3859943  
## 9 TCGA-A1-A0SG 1.682242 4.242539 0.3074847 0.0000000 0.0000000 1.1505720  
## 10 TCGA-A1-A0SH 1.387494 4.391981 0.0000000 0.0000000 0.0000000 0.3083667  
## SRP14P1 TRIM75P SPATA31B1P  
## 1 1.636255 0.5243142 0  
## 2 0.967478 0.4342469 0  
## 3 1.313697 0.0000000 0  
## 4 1.380955 0.0000000 0  
## 5 1.030333 0.3719083 0  
## 6 1.728376 0.2857803 0  
## 7 1.886221 0.2845020 0  
## 8 2.344197 0.0000000 0  
## 9 2.265931 0.0000000 0  
## 10 1.603963 0.0000000 0

any(is.na(data\_expression\_cleaned\_log))

## [1] FALSE

sum((apply(data\_expression\_cleaned\_log, 2, function(x) length(unique(x)) == 1))==TRUE)

## [1] 0

The RNA Seq expression data was cleaned in the above chunk of code. The rows with missing values were taken off. The data was transposed. Tidying of column and row names was done. Patient Ids were matched with those from expression and clinical data. The features with constant values for every patient were considered as missing values as this is not biological compatible and were thus removed. The type was changed to numeric for the expression values after which they were log transformed. A new column was created for the patient IDs.

## 6. Feature Selection.

# Merge expression data with survival data  
data\_merged <- merge(data\_survival\_cleaned, data\_expression\_cleaned\_log, by = "PATIENT\_ID")  
  
# Get expression data  
data\_expression2 <- data\_merged %>%   
 select(-PATIENT\_ID, -AGE, -SEX, -OS\_STATUS, -DSS\_STATUS, -DFS\_STATUS, -PFS\_STATUS, -OS\_MONTHS, -DSS\_MONTHS, -DFS\_MONTHS,-PFS\_MONTHS)  
data\_expression2[1:10,1:10]

## UBE2Q2P2 HMGB1P1 RNU12.2P SSX9P EZHIP EFCAB8 SRP14P1 TRIM75P  
## 1 2.634160 3.973124 0.0000000 0.0000000 1.0018444 0.2961709 1.636255 0.5243142  
## 2 2.911209 4.259227 0.7359672 0.4342469 4.9774232 1.7739349 0.967478 0.4342469  
## 3 2.325266 5.045339 0.0000000 0.0000000 0.0000000 1.3136968 1.313697 0.0000000  
## 4 2.736301 4.445152 0.3545226 0.0000000 0.3545226 1.3809551 1.380955 0.0000000  
## 5 2.733945 4.765476 0.3719083 0.0000000 0.6423801 0.6423801 1.030333 0.3719083  
## 6 2.511557 4.115176 0.2857803 0.0000000 0.5077810 0.2857803 1.728376 0.2857803  
## 7 1.694257 5.037894 0.0000000 0.0000000 0.0000000 0.6867266 1.886221 0.2845020  
## 8 2.463010 4.957185 0.0000000 0.0000000 0.0000000 1.3859943 2.344197 0.0000000  
## 9 1.682242 4.242539 0.3074847 0.0000000 0.0000000 1.1505720 2.265931 0.0000000  
## 10 1.387494 4.391981 0.0000000 0.0000000 0.0000000 0.3083667 1.603963 0.0000000  
## SPATA31B1P REXO1L6P  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0  
## 7 0 0  
## 8 0 0  
## 9 0 0  
## 10 0 0

# Get OS data for Surv object  
data\_surv\_OS <- data\_merged %>% select(PATIENT\_ID, OS\_MONTHS, OS\_STATUS)  
# Get Surv Object  
surv\_obj\_OS <- Surv(time = data\_surv\_OS$OS\_MONTHS, event = data\_surv\_OS$OS\_STATUS)  
head(surv\_obj\_OS)

## [1] 133.050597+ 131.669790+ 48.459743+ 11.440971+ 8.514975+ 14.366966+

any(is.na(surv\_obj\_OS))

## [1] FALSE

### 6.1. Top 100 Genes by Fitting Cox Proportional Hazard Model.

# Get p values  
p\_vals <- numeric(ncol(data\_expression2))  
  
# Fit a coxph model  
for (i in 1:ncol(data\_expression2)) {  
 gene <- data\_expression2[, i]  
 cox\_model <- coxph(surv\_obj\_OS ~ gene)  
 p\_vals[i] <- summary(cox\_model)$coefficients[5]  
}  
  
# Genes with p-values in a data frame  
gene\_p\_vals <- data.frame(Gene = colnames(data\_expression2), P\_Value = p\_vals)  
# sort by p values  
gene\_p\_vals <- gene\_p\_vals[order(gene\_p\_vals$P\_Value), ]  
# Get top 100 genes  
top\_genes <- gene\_p\_vals[1:100, ]  
knitr::kable(top\_genes)

|  | Gene | P\_Value |
| --- | --- | --- |
| 13054 | PCNAP1 | 0.0000000 |
| 9521 | LHX5 | 0.0000003 |
| 2491 | C3P1 | 0.0000008 |
| 12363 | OR13C5 | 0.0000117 |
| 6821 | GALP | 0.0000203 |
| 10436 | MAFA | 0.0000347 |
| 14659 | RBBP8 | 0.0000483 |
| 12618 | OR5T3 | 0.0000542 |
| 13050 | PCMT1 | 0.0000553 |
| 8659 | JAK2 | 0.0000763 |
| 7731 | HES3 | 0.0000771 |
| 2284 | C20orf191 | 0.0000788 |
| 9415 | LCE3C | 0.0000998 |
| 13236 | PFKL | 0.0000999 |
| 9158 | KLRB1 | 0.0001126 |
| 15466 | SAV1 | 0.0001174 |
| 19093 | VN1R4 | 0.0001773 |
| 14829 | RFX3 | 0.0002043 |
| 476 | AK7 | 0.0002074 |
| 19141 | VSIG8 | 0.0002176 |
| 4714 | DDAH1 | 0.0002653 |
| 2664 | C7orf33 | 0.0002690 |
| 9656 | GALNT9.AS1 | 0.0002715 |
| 3455 | CDK14 | 0.0002738 |
| 19201 | WDR17 | 0.0003025 |
| 17650 | TFPI2 | 0.0003115 |
| 927 | ARHGAP26 | 0.0003167 |
| 11804 | NGB | 0.0003417 |
| 2913 | CAMK4 | 0.0003641 |
| 19318 | WNT3A | 0.0003678 |
| 2164 | C1orf173 | 0.0004010 |
| 5216 | DTHD1 | 0.0004199 |
| 13454 | PKHD1L1 | 0.0004456 |
| 8763 | KCNIP3 | 0.0004515 |
| 17230 | SURF4 | 0.0005153 |
| 2803 | C9orf79 | 0.0005449 |
| 12943 | PAX7 | 0.0005567 |
| 12670 | OR8B4 | 0.0005708 |
| 829 | APOA2 | 0.0006205 |
| 11782 | NFIA | 0.0006278 |
| 10696 | MDFIC | 0.0006402 |
| 7243 | GPM6A | 0.0006460 |
| 1618 | BTBD10 | 0.0006594 |
| 13374 | PIGR | 0.0006638 |
| 19826 | ZNF385B | 0.0006688 |
| 137 | ACADM | 0.0006965 |
| 8436 | IMPACT | 0.0006972 |
| 7965 | HOOK1 | 0.0006984 |
| 12628 | OR6C4 | 0.0007236 |
| 19656 | ZNF101 | 0.0007399 |
| 13679 | POFUT2 | 0.0007584 |
| 1740 | C11orf46 | 0.0007905 |
| 12412 | OR2AG1 | 0.0008190 |
| 9008 | KIAA2026 | 0.0008635 |
| 16877 | SPINT1 | 0.0008670 |
| 19822 | ZNF382 | 0.0008839 |
| 16484 | SMR3A | 0.0009012 |
| 8195 | ICOSLG | 0.0009313 |
| 13048 | PCLO | 0.0009446 |
| 3231 | CCNA1 | 0.0009648 |
| 13263 | PGK1 | 0.0010253 |
| 14708 | RBMXL1 | 0.0010333 |
| 2719 | C8orf55 | 0.0010395 |
| 9416 | LCE3D | 0.0010488 |
| 19105 | VPS13C | 0.0010546 |
| 10256 | LRRC30 | 0.0010569 |
| 10900 | MIER1 | 0.0010766 |
| 3875 | CLIC6 | 0.0010841 |
| 18011 | TMEM31 | 0.0011329 |
| 9427 | LCN1 | 0.0011971 |
| 15805 | SF3B5 | 0.0012865 |
| 17126 | STK39 | 0.0013133 |
| 4545 | CYP27A1 | 0.0013149 |
| 1487 | BFSP1 | 0.0013230 |
| 8304 | IGJ | 0.0013360 |
| 11187 | MS4A1 | 0.0013517 |
| 1338 | B4GALT3 | 0.0013586 |
| 2344 | C21orf90 | 0.0013821 |
| 16988 | SS18 | 0.0014050 |
| 13762 | POT1 | 0.0014096 |
| 407 | AGBL2 | 0.0014248 |
| 15315 | RSPH4A | 0.0014502 |
| 10133 | LINC00917 | 0.0014586 |
| 4275 | CRYZ | 0.0014600 |
| 18039 | TMEM56 | 0.0014649 |
| 15905 | SH3GL1 | 0.0014955 |
| 17551 | TCP1 | 0.0015130 |
| 16089 | SLC19A1 | 0.0015391 |
| 9642 | MAPT.AS1 | 0.0015863 |
| 10432 | MAEA | 0.0015980 |
| 9562 | LIN7C | 0.0016073 |
| 16851 | SPG20 | 0.0016121 |
| 13968 | PRDM12 | 0.0016234 |
| 12258 | ODF3L2 | 0.0016634 |
| 3016 | CBLL1 | 0.0016921 |
| 8548 | IRF2 | 0.0016949 |
| 19145 | VSTM2B | 0.0017222 |
| 6058 | FAM46C | 0.0017857 |
| 1831 | C13orf28 | 0.0018145 |
| 2797 | C9orf69 | 0.0018640 |

The Top 100 Genes are listed above in ascending order by p-values.

## 7. K-Means Clustering.

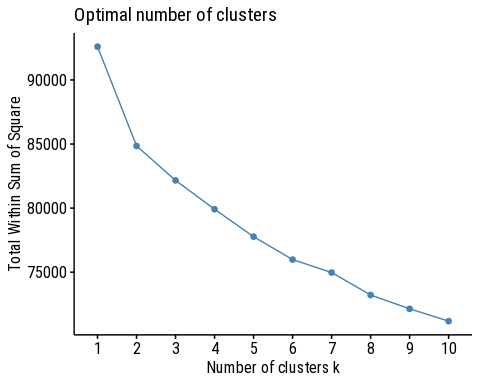
# Scale expression data using z-score method  
data\_expression\_scaled <- scale(data\_expression2)  
  
# Add row names  
rownames(data\_expression\_scaled) <- data\_merged$PATIENT\_ID  
  
# Get the top 100 gene expressions  
data\_expression\_top100 <- data\_expression\_scaled[, top\_genes$Gene]  
data\_expression\_top100[1:10,1:10]

## PCNAP1 LHX5 C3P1 OR13C5 GALP MAFA  
## TCGA-3C-AAAU -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 -0.56848760  
## TCGA-3C-AALI -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 0.98342964  
## TCGA-3C-AALJ -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 0.79228727  
## TCGA-4H-AAAK -0.1070592 -0.3099526 -0.1190506 2.4996550 -0.209855 -0.56848760  
## TCGA-A1-A0SB -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 -0.56848760  
## TCGA-A1-A0SD -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 -0.56848760  
## TCGA-A1-A0SE -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 0.03143534  
## TCGA-A1-A0SF -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 0.18337168  
## TCGA-A1-A0SG -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 5.51700429  
## TCGA-A1-A0SH -0.1070592 -0.3099526 -0.1190506 -0.2703569 -0.209855 -0.56848760  
## RBBP8 OR5T3 PCMT1 JAK2  
## TCGA-3C-AAAU 1.09419010 -0.08523252 -2.67632248 -0.02202686  
## TCGA-3C-AALI -0.92321034 -0.08523252 -0.54166904 -0.86473414  
## TCGA-3C-AALJ -0.09630321 -0.08523252 0.24217440 -1.23967175  
## TCGA-4H-AAAK -0.60578106 -0.08523252 -0.09297389 0.33193680  
## TCGA-A1-A0SB 0.45373520 -0.08523252 -0.82164467 0.50452272  
## TCGA-A1-A0SD -0.71977343 -0.08523252 0.43808424 0.13284856  
## TCGA-A1-A0SE 0.42285264 -0.08523252 -1.55340045 0.63849944  
## TCGA-A1-A0SF -0.62853694 -0.08523252 -0.21549565 0.26476024  
## TCGA-A1-A0SG 0.30286094 -0.08523252 -0.61458466 0.99802573  
## TCGA-A1-A0SH -0.57540307 -0.08523252 -0.04668021 0.24220227

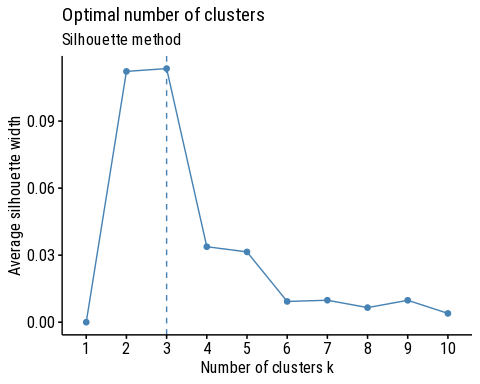
Here the expression data for the top 100 genes was extracted.

### 7.1. Number of Clusters.

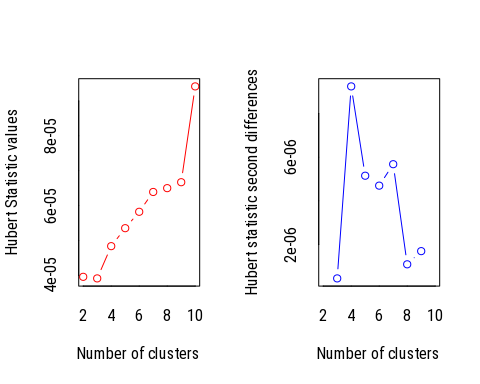
# Range for clusters  
k\_range <- 2:10  
  
# Within-cluster sum of squares (WCSS) for each k value  
wcss <- vector("numeric", length(k\_range))  
  
# k-means for each value of k and get WCSS  
for (i in seq\_along(k\_range)) {  
 k <- k\_range[i]  
 kmeans\_res <- kmeans(data\_expression\_top100, centers = k, nstart = 25)  
 wcss[i] <- kmeans\_res$tot.withinss  
}  
  
# Elbow plot  
fviz\_nbclust(data\_expression\_top100, kmeans, method = "wss", k.max = 10, nstart = 25)



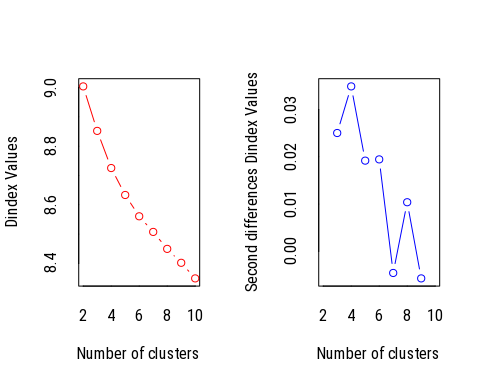
# silhouette method  
fviz\_nbclust(data\_expression\_top100, kmeans, method = "silhouette")+  
 labs(subtitle = "Silhouette method")



# Gap\_Stat is very slow  
#set.seed(786)  
#fviz\_nbclust(data\_expression\_top100, kmeans, nstart = 25, method = "gap\_stat", nboot = 500, verbose=FALSE)+  
# labs(subtitle = "Gap statistic method")  
  
# Using NbClust()  
set.seed(786) # For reproducibility  
nbclust\_res <- NbClust(data = data\_expression\_top100,   
 distance = "euclidean",   
 min.nc = 2,   
 max.nc = 10,   
 method = "kmeans")

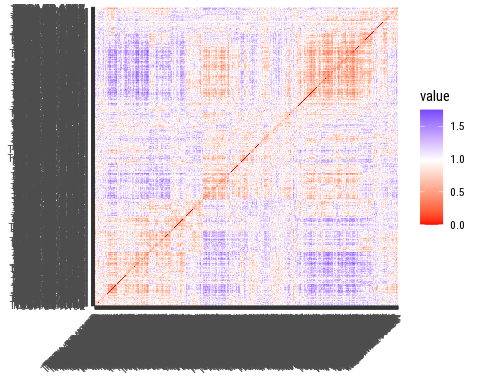


## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##

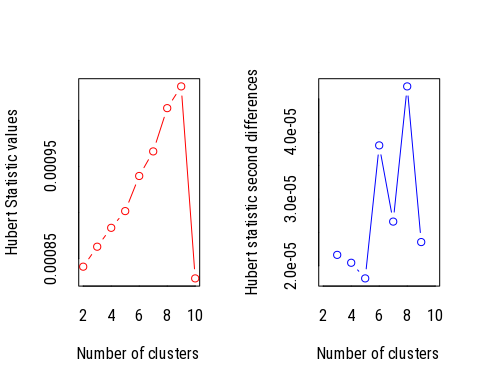


## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 11 proposed 2 as the best number of clusters   
## \* 4 proposed 3 as the best number of clusters   
## \* 2 proposed 4 as the best number of clusters   
## \* 1 proposed 6 as the best number of clusters   
## \* 1 proposed 9 as the best number of clusters   
## \* 5 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

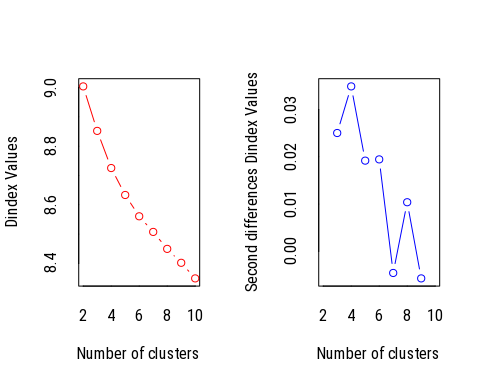
# Correlation matrix for the distance  
dist.cor <- get\_dist(data\_expression\_top100, method = "spearman")  
fviz\_dist(dist.cor)



nbclust\_spearman <- NbClust(data = data\_expression\_top100,  
 diss = dist.cor,  
 distance = NULL,  
 min.nc = 2,  
 max.nc = 10,  
 method = "kmeans")



## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.   
##



## \*\*\* : The D index is a graphical method of determining the number of clusters.   
## In the plot of D index, we seek a significant knee (the significant peak in Dindex  
## second differences plot) that corresponds to a significant increase of the value of  
## the measure.   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   
## \* Among all indices:   
## \* 11 proposed 2 as the best number of clusters   
## \* 5 proposed 3 as the best number of clusters   
## \* 2 proposed 4 as the best number of clusters   
## \* 2 proposed 6 as the best number of clusters   
## \* 4 proposed 10 as the best number of clusters   
##   
## \*\*\*\*\* Conclusion \*\*\*\*\*   
##   
## \* According to the majority rule, the best number of clusters is 2   
##   
##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

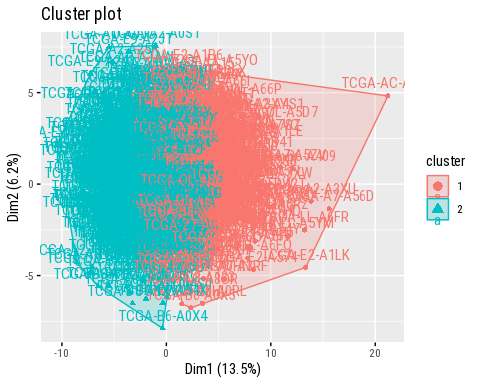
The elbow and Silhouette Plots were not conclusive. Using NbClust(), the Spearman correlation was applied to the diss option with NULL added to the distance option. Finally, 2 as the value of k number of clusters based on the Hubert Statistics Plots.

### 7.2 Clustering with k=2.

set.seed(786)  
kmeans\_res <- kmeans(data\_expression\_top100, centers = 2, nstart = 25)  
str(kmeans\_res)

## List of 9  
## $ cluster : Named int [1:927] 2 1 1 2 2 2 2 2 2 2 ...  
## ..- attr(\*, "names")= chr [1:927] "TCGA-3C-AAAU" "TCGA-3C-AALI" "TCGA-3C-AALJ" "TCGA-4H-AAAK" ...  
## $ centers : num [1:2, 1:100] 0.0204 -0.0121 0.257 -0.153 0.0483 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "1" "2"  
## .. ..$ : chr [1:100] "PCNAP1" "LHX5" "C3P1" "OR13C5" ...  
## $ totss : num 92600  
## $ withinss : num [1:2] 40750 44102  
## $ tot.withinss: num 84852  
## $ betweenss : num 7748  
## $ size : int [1:2] 346 581  
## $ iter : int 1  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

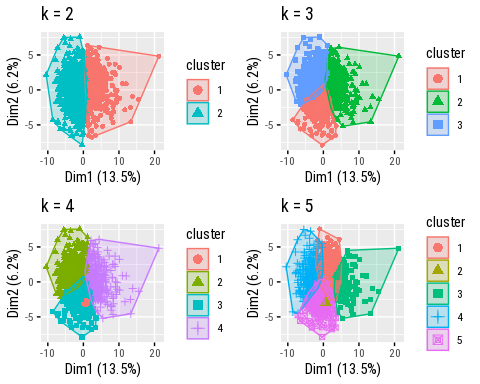
# Visualize clusters  
fviz\_cluster(kmeans\_res, data = data\_expression\_top100)



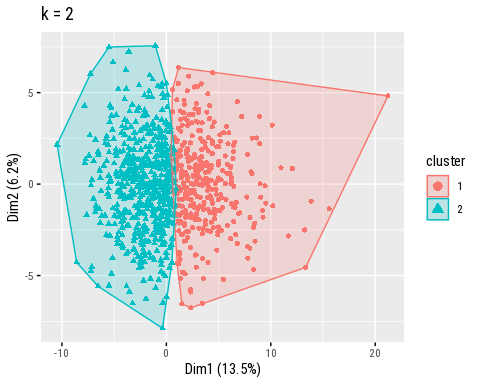
Note the two clusters are overlapping, higher values of K will be tried out.

### 7.3. K-means clustering with higher values of k.

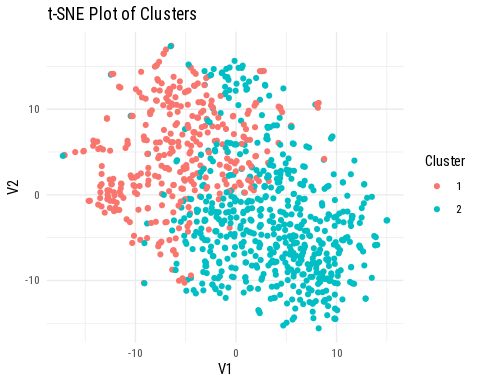
k3 <- kmeans(data\_expression\_top100, centers = 3, nstart = 25)  
k4 <- kmeans(data\_expression\_top100, centers = 4, nstart = 25)  
k5 <- kmeans(data\_expression\_top100, centers = 5, nstart = 25)  
  
# plots to compare  
p1 <- fviz\_cluster(kmeans\_res, geom = "point", data = data\_expression\_top100) + ggtitle("k = 2")  
p2 <- fviz\_cluster(k3, geom = "point", data = data\_expression\_top100) + ggtitle("k = 3")  
p3 <- fviz\_cluster(k4, geom = "point", data = data\_expression\_top100) + ggtitle("k = 4")  
p4 <- fviz\_cluster(k5, geom = "point", data = data\_expression\_top100) + ggtitle("k = 5")  
  
grid.arrange(p1, p2, p3, p4, nrow = 2)



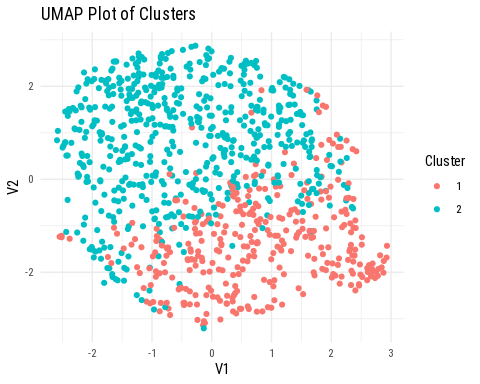
# Final Kmeans with k = 2  
final\_kmeans <- kmeans\_res  
fviz\_cluster(final\_kmeans, geom = "point", data = data\_expression\_top100) + ggtitle("k = 2")



# Perform t-SNE  
tsne\_res <- Rtsne(as.matrix(data\_expression\_top100), dims = 2, perplexity = 30)  
# t-SNE result data frame  
tsne\_df <- as.data.frame(tsne\_res$Y)  
tsne\_df$Cluster <- as.factor(final\_kmeans$cluster)  
# Visualize t-SNE result  
ggplot(tsne\_df, aes(x = V1, y = V2, color = Cluster)) +  
 geom\_point() +  
 ggtitle("t-SNE Plot of Clusters") +  
 theme\_minimal()



# Perform UMAP  
umap\_res <- umap(data\_expression\_top100)  
  
# UMAP result in a data frame  
umap\_df <- as.data.frame(umap\_res$layout)  
umap\_df$Cluster <- as.factor(final\_kmeans$cluster)  
  
# Visualize UMAP result  
ggplot(umap\_df, aes(x = V1, y = V2, color = Cluster)) +  
 geom\_point() +  
 ggtitle("UMAP Plot of Clusters") +  
 theme\_minimal()



The directive from business specifies the use of kmeans for clustering and thus has to be followed through but other methods of clustering may be used to look for better outlined clusters. For now, we move forward with kmeans using 2 clusters. The clusters can be seen using tSNE and UMAP.

### 7.4. Assign clusters to data.

data\_expression\_clustered <- data.frame(data\_expression\_top100,final\_kmeans$cluster)  
# Samples in each cluster  
print(table(data\_expression\_clustered$final\_kmeans.cluster))

##   
## 1 2   
## 346 581

Above are the number of patients in cluster 1 and 2 being 346 and 581 respectively.

## 8. Survival Analysis in Clusters.

# Choose a random gene  
set.seed(786)  
random\_gene <- sample(top\_genes$Gene, 1)  
random\_gene

## [1] "IRF2"

The random gene is IRF2.

### 8.1. Cox Proportional Hazards Model.

#### 8.1.1. Effect of IRF2 within Cluster 1.

# Effect of random gene on cluster 1  
data\_expression\_clustered$PATIENT\_ID <- rownames(data\_expression\_clustered)  
data\_cluster1 <- data\_expression\_clustered %>%  
 filter(final\_kmeans.cluster == 1)  
  
# Merge Survival OS data and the clusterd expression data  
data\_cluster1OS <- merge(data\_cluster1, data\_surv\_OS, by ="PATIENT\_ID")  
  
# Survival object for cluster 1  
surv\_obj\_cluster1 <- Surv(time = data\_cluster1OS$OS\_MONTHS, event = data\_cluster1OS$OS\_STATUS)  
head(surv\_obj\_cluster1)

## [1] 131.66979+ 48.45974+ 73.84029+ 87.25384+ 101.98244+ 136.73275+

dim(surv\_obj\_cluster1)

## [1] 346 2

dim(data\_cluster1)

## [1] 346 102

# Fit the CoxPH model  
cox\_model\_cluster1 <- coxph(surv\_obj\_cluster1 ~ data\_cluster1[[random\_gene]], data = data\_cluster1)  
  
# Display results  
coxph\_cluster1 <- summary(cox\_model\_cluster1)  
coxph\_cluster1

## Call:  
## coxph(formula = surv\_obj\_cluster1 ~ data\_cluster1[[random\_gene]],   
## data = data\_cluster1)  
##   
## n= 346, number of events= 46   
##   
## coef exp(coef) se(coef) z Pr(>|z|)  
## data\_cluster1[[random\_gene]] -0.1959 0.8221 0.1298 -1.509 0.131  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## data\_cluster1[[random\_gene]] 0.8221 1.216 0.6374 1.06  
##   
## Concordance= 0.561 (se = 0.048 )  
## Likelihood ratio test= 2.16 on 1 df, p=0.1  
## Wald test = 2.28 on 1 df, p=0.1  
## Score (logrank) test = 2.28 on 1 df, p=0.1

The summary of the Cox Proportional Hazards Model (CoxPH) for cluster 1 indicates the following:

1. Coefficient: This is associated with the expression of the randomly chosen gene (IRF2) within cluster 1 is approximately -0.1959.
2. Hazard Ratio: The hazard ratio (the change in the hazard for a one-unit increase in the expression of random\_gene, is approximately 0.8221). The hazard of experiencing the event (death) decreases by approximately 17.79% (1 - 0.8221 = 0.1779 or 17.79%). Higher expression levels of random\_gene are associated with a decreased risk of death occurring within cluster 1 indicating a potential protective effect.
3. Standard Error: The standard error associated with the coefficient estimate is 0.1298.
4. Z-value (z): The z-value (the coefficient divided by its standard error) is approximately -1.509.
5. p-value: The p-value associated with the z-value is 0.131, which indicates that there is no significant association between the expression of random\_gene and survival within cluster 1A p-value less than the conventional significance level of 0.05 is typically considered statistically significant. Here, the p-value is greater than 0.05 (p = 0.131), indicating that there is insufficient evidence to reject the null hypothesis of no association between random\_gene expression and survival within cluster 1.
6. 95% Confidence Interval (lower .95, upper .95): The 95% confidence interval for the hazard ratio ranges from approximately 0.6374 to 1.06.
7. Concordance: The concordance index, a measure of predictive accuracy, is approximately 0.561. This suggests that the model’s predictive accuracy is slightly better than random chance (0.5). It is not high enough, indicating that there is room for improvement in the model’s ability to rank individuals according to their risk of experiencing the event (survival time).
8. Likelihood Ratio Test: The likelihood ratio test statistic is 2.16 on 1 degree of freedom, with a corresponding p-value of 0.1.
9. Wald Test: The Wald test statistic is 2.28 on 1 degree of freedom, with a corresponding p-value of 0.1.
10. Score (logrank) Test: The score test statistic, which is also known as the logrank test, is 2.28 on 1 degree of freedom, with a corresponding p-value of 0.1.

There is a trend suggesting an association between the expression of IRF2 and survival within cluster 1, the results are not statistically significant at the conventional significance level of 0.05. Further investigation or larger sample sizes may be warranted to confirm any potential associations. Other clustering methods may also be explored in further analysis.

#### 8.1.2. Effect of IRF2 within Cluster 2.

# Effect of random gene on cluster 2  
data\_cluster2 <- data\_expression\_clustered %>%  
 filter(final\_kmeans.cluster == 2)  
  
# Merge Survival OS data and the clusterd expression data  
data\_cluster2OS <- merge(data\_cluster2, data\_surv\_OS, by ="PATIENT\_ID")  
  
# Survival object for cluster 1  
surv\_obj\_cluster2 <- Surv(time = data\_cluster2OS$OS\_MONTHS, event = data\_cluster2OS$OS\_STATUS)  
head(surv\_obj\_cluster2)

## [1] 133.050597+ 11.440971+ 8.514975+ 14.366966+ 43.429661+ 48.098103+

dim(surv\_obj\_cluster2)

## [1] 581 2

dim(data\_cluster2)

## [1] 581 102

# Fit the CoxPH model  
cox\_model\_cluster2 <- coxph(surv\_obj\_cluster2 ~ data\_cluster2[[random\_gene]], data = data\_cluster2)  
  
# Display results  
summary(cox\_model\_cluster2)

## Call:  
## coxph(formula = surv\_obj\_cluster2 ~ data\_cluster2[[random\_gene]],   
## data = data\_cluster2)  
##   
## n= 581, number of events= 28   
##   
## coef exp(coef) se(coef) z Pr(>|z|)  
## data\_cluster2[[random\_gene]] -0.07365 0.92900 0.22789 -0.323 0.747  
##   
## exp(coef) exp(-coef) lower .95 upper .95  
## data\_cluster2[[random\_gene]] 0.929 1.076 0.5943 1.452  
##   
## Concordance= 0.495 (se = 0.07 )  
## Likelihood ratio test= 0.1 on 1 df, p=0.7  
## Wald test = 0.1 on 1 df, p=0.7  
## Score (logrank) test = 0.1 on 1 df, p=0.7

From the results for cluster 2: 1. Hazard Ratio (HR): The coefficient associated with the random gene (IRF2) is -0.07365. The hazard ratio (HR) is calculated as exp(coef), which is 0.929. For each unit increase in the expression of IRF2, the risk or hazard of an event decreases by approximately 7.1% (1 - HR \* 100).

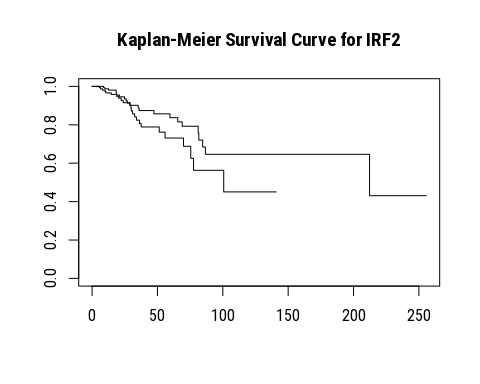
1. P-value: The p-value associated with the coefficient is 0.747, this show not week significance. The p-value is high, suggesting that the association may not be statistically significant.
2. Concordance Index (C-index): The C-index is reported as 0.495 with a standard error of 0.07. The C-index measures the predictive accuracy of the model. A value closer to 1 indicates better predictive accuracy. Here, a value of 0.495 suggests poor predictive accuracy within Cluster 2.
3. Likelihood Ratio Test (LR Test), Wald Test, and Score (Log-rank) Test: The p-values for these tests are approximately 0.7. They assess the significance of the coefficient associated with IRF2. The high p-values suggest that there is no significant evidence to reject the null hypothesis, indicating that the association between IRF2 expression and survival within Cluster 2 may not be statistically significant.

To Summarize the effect of IRF2 on cluster 1 and 2 through CoxPH, the model shows that for both clusters the hazard ratios indicate the direction and magnitude of the association between IRF2 and survival. The p-values and C-index values suggest that the association may not be statistically significant and the predictive accuracy of the models may be limited within these clusters. Further investigation and refinement of the models may be necessary to identify significant predictors of survival within each cluster.

### 8.2. Kaplan-Meier Survival Analysis.

#### 8.2.1. Effect of High and Low expression of IRF2 on Cluster 1.

# Function for KM analysis  
analyze\_survival\_km <- function(data, gene\_expression\_column, median\_expression) {  
 # Column for gene expression above or below the median  
 data$group <- ifelse(data[[gene\_expression\_column]] >= median\_expression, "High", "Low")  
   
 # Survival object  
 surv\_obj <- Surv(time = data$OS\_MONTHS, event = data$OS\_STATUS)  
   
 # Fit Kaplan-Meier survival model  
 surv\_fit <- survfit(surv\_obj ~ group, data = data)  
   
 # Visualize Kaplan-Meier survival curves  
 plot(surv\_fit, main = paste("Kaplan-Meier Survival Curve for", gene\_expression\_column))  
   
 # Log-rank test  
 surv\_diff <- survdiff(surv\_obj ~ group, data = data)  
 print(surv\_diff)  
}  
  
# Perform survival analysis for cluster 1  
analyze\_survival\_km(data\_cluster1OS, "IRF2", median\_expression = median(data\_cluster1OS$IRF2))



## Call:  
## survdiff(formula = surv\_obj ~ group, data = data)  
##   
## N Observed Expected (O-E)^2/E (O-E)^2/V  
## group=High 173 22 27.5 1.09 2.8  
## group=Low 173 24 18.5 1.61 2.8  
##   
## Chisq= 2.8 on 1 degrees of freedom, p= 0.09

From the above curves and statistical tests for Cluster 1:  
The KM curve shows two lines representing the survival probability over time for the “High” and “Low” IRF2 expression groups. The x-axis represents the time in months, while the y-axis represents the survival probability. The separation between the two lines indicates the difference in survival probability between the two groups.

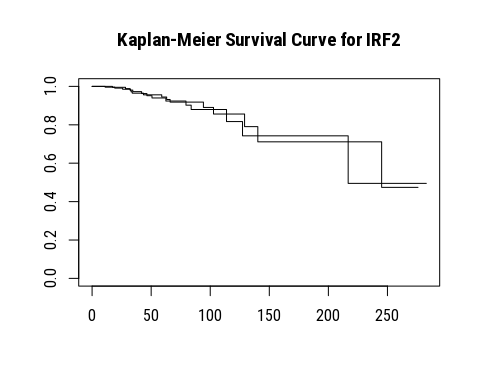
For the Log-Rank Test:

1. The chi-squared statistic is 2.8 with 1 degree of freedom.
2. The p-value associated with this test statistic is 0.09.

Since the p-value (0.09) is greater than the significance level of 0.05, we fail to reject the null hypothesis. There is not enough statistical evidence to conclude that there is a significant difference in survival between the “High” and “Low” expression groups for the gene IRF2 in Cluster 1.

#### 8.2.2. Effect of High and Low expression of IRF2 on Cluster 1.

# Perform survival analysis for cluster 2  
analyze\_survival\_km(data\_cluster2OS, "IRF2", median\_expression = median(data\_cluster2OS$IRF2))



## Call:  
## survdiff(formula = surv\_obj ~ group, data = data)  
##   
## N Observed Expected (O-E)^2/E (O-E)^2/V  
## group=High 291 15 15.2 0.00287 0.0063  
## group=Low 290 13 12.8 0.00341 0.0063  
##   
## Chisq= 0 on 1 degrees of freedom, p= 0.9

For Cluster 2 KM analysis: Similar to Cluster 1, the KM curve for Cluster 2 show the survival probabilities over time for the “High” and “Low” IRF2 expression groups. the KM curves can provide a visual confirmation of the lack of difference in survival probabilities between the groups with several points of overlap and crosing over. The chi-squared statistic is 0, and the p-value is 0.9. Since the p-value is significantly higher than 0.05, we conclude that there is no statistically significant difference in survival between the “High” and “Low” expression groups for the gene IRF2 in Cluster 2. The expression level of IRF2 does not have a significant impact on the survival of patients in Cluster 2.

## 9. Results and Interpretation.

### 9.1. Summary of Results.

#### Cox Proportional Hazards Model (CoxPH) Analysis

##### Cluster 1:

Gene: IRF2

Coefficient (coef): -0.1959

Hazard Ratio (exp(coef)): 0.8221

P-value: 0.131

Comments: The hazard ratio of 0.8221 indicates that higher expression of IRF2 is associated with a 17.79% decrease in the hazard (risk of death). However, this result is not statistically significant (p > 0.05).

##### Cluster 2:

Gene: IRF2

Coefficient (coef): -0.07365

Hazard Ratio (exp(coef)): 0.9290

P-value: 0.747

Comments: The hazard ratio of 0.9290 suggests a slight decrease in the hazard with higher expression of IRF2, but this result is not statistically significant (p > 0.05).

#### Kaplan-Meier (KM) Survival Analysis.

##### Cluster 1:

Chi-squared Statistic (Chisq): 2.8

P-value: 0.09  
Comment: There is no significant difference in survival between the “High” and “Low” IRF2 expression groups (p > 0.05).

#### Cluster 2:

Chi-squared Statistic (Chisq): 0  
P-value: 0.9  
Comment: There is no significant difference in survival between the “High” and “Low” IRF2 expression groups (p > 0.05)

### 9.2. Final Conclusion.

The survival analysis using both Cox Proportional Hazards Models and Kaplan-Meier analysis for the gene IRF2 in Clusters 1 and 2 did not show any statistically significant impact on patient survival. In both clusters, neither the CoxPH nor the KM analyses indicated significant differences in survival based on IRF2 expression levels.

### 9.3. Suggestions for Improvement.

To enhance the accuracy and reliability of clustering and survival analysis, different clustering algorithms such as hierarchical clustering and DBSCAN can be used while evaluating their performance. Advanced methods like consensus clustering can identify more robust clusters.

Including additional covariates in the CoxPH model can account for potential confounding factors, and performing multivariate analysis may improve the model’s predictive power. Covariates like age and gender are a great place to start.

Data pre-processing techniques like feature selection and PCA can reduce noise and improve clustering quality, while ensuring proper normalization and scaling.

Validation of findings using independent data sets and cross-validation techniques will ensure robustness and generalizing. In R, the clValid package provides nice options.

With additional time and discussion on business needs, these improvements can be systematically incorporated into the current analysis. Implementing these enhancements will lead to more robust, accurate, and insightful results in both clustering and survival analyses. This will facilitate more meaningful interpretations and conclusions to advance understanding of the data. While this study focused on the event of death in the given time, the DSS, DFS and PFS status can be looked into to understand how disease progression can affect the chances of survival.

Thank you for your audience!!!