TCIA Data Exploration

Grant Carr

2025-06-13

R Setup

Load packages that are necessary for running all code below.

```
#' example code for installing packages for first time:
#' install.packages("readxl")


library(readxl)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(survival)
library(ggfortify)
library(devtools)
```

```
## Loading required package: usethis
```

```
#' install_github("jokergoo/ComplexHeatmap")
library(ComplexHeatmap)
```

```
## Loading required package: grid
## ========================================
## ComplexHeatmap version 2.20.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
```

```
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
```

Reading Clinical Data

```
clinData <- read_xlsx(
  "/Users/grantcarr/Documents/Michigan/BDSI2025/Data/Clinical_and_Other_Features.xlsx",
  sheet = 1, skip = 1
)
```

```
## New names:
## * `` -> `...33`
## * `` -> `...34`
## * `` -> `...40`
## * `` -> `...41`
## * `` -> `...43`
## * `` -> `...44`
## * `` -> `...45`
## * `` -> `...46`
## * `` -> `...47`
## * `` -> `...48`
## * `` -> `...68`
## * `Shape` -> `Shape...71`
## * `Margin` -> `Margin...72`
## * `Tumor Size (cm)` -> `Tumor Size (cm)...76`
## * `Shape` -> `Shape...77`
## * `Margin` -> `Margin...78`
## * `Tumor Size (cm)` -> `Tumor Size (cm)...79`
```

```
clinData <- clinData[-1,]
```

The '<-' is the assignment operator in R. clinData <- read_xlsx(...) assigns the output of read_xlsx() into a variable/object called clinData. Keyboard shortcut on a Mac is option, -. You can also use '='. '<-' is used to differentiate setting function arguments versus assigning data to a variable.

If you open the dataset, you will see that the first row is variable group headings. We want to skip that row when we read the data with the function argument "skip = 1".

The second row is treated as the column names and the third row is treated as data. In excel, the third row is variable descriptions, not data. So we remove the first row with clinData[-1,]. When you are working with 2-dimensional datasets/matrix objects, you access rows/columns with clinData[rows, columns]. You can pass a vector of numbers to select (or negative numbers to select everything except) specific rows/columns of the data.

Changing Column Names

In the output above, you can see that R changed blank column names to "...[column number]", and duplicated column names to "[column name]...[column number]". Below I manually changed the names of those blank/duplicated column names into descriptive column names based on the variable descriptions in the excel file.

```
clinData <- clinData %>% rename(
  TumorGradeT = `Tumor Grade`,
  TumorGradeN = ...33,
  TumorGradeM = ...34,
  DiffBilateralReceptors = ...40,
  AnnotatedBilateralSide = ...41,
  OtherBilateralSide = `For Other Side If Bilateral`,
  OtherBilateralOncotype = ...43,
  OtherBilateralGrade = ...44,
  OtherBilateralER = ...45,
  OtherBilateralPR = ...46,
  OtherBilateralHER2 = ...47,
  OtherBilateralSubtype = ...48,
  Shape_mammo = Shape...71,
  Margin_mammo = Margin...72,
  TumorSize_mammo = `Tumor Size (cm)...76`,

  Shape_us = Shape...77,
  Margin_us = Margin...78,
  TumorSize_us = `Tumor Size (cm)...79`
) %>% select(-...68)
```

The %>% operator (command, shift, m shortcut on Mac) is from the dplyr package and means "use the output on the left side as the first argument of the function on the right". It is useful for sequentially performing multiple functions in a single line of code, especially for data formatting. "select(-...68)" again means that we select every column except for the one named "...68".

```
table(clinData$AnnotatedBilateralSide, useNA = "if")
```

```
##
##    L   NC   NP    R
##   16  271  623   12
```

```
table(clinData$Shape_us, useNA = "if")
```

```
##
##    [1]       0   0 (1)       1 1   [1]   1 (1)   1 [1]       2      NA      NC
##      4       9       2      82       1       1       1      10     161     651
```

```
table(clinData$`Reconstruction Diameter`, useNA = "if")
```

```
##
##    0    1   10   11   12   13   14   15   16   17   18   19    2    3    4    6
##    1    1  112  135   23   44    6   18    1    1    1    1    8    8   29   12
##    7    8    9 <NA>
##   55   23  149  294
```

Tabulating the possible values of the example variables above, we see the labels "NA", "NC", and "NP". Note that "NA" is the letters "NA", not R's value for NA. This means that R is treating it as data instead of missing.

From the excel file, NA means not applicable, NC means not collected, and NP means not pertinent. All of these indicate that we don't have the relevant data for that patient.

Depending on the variable, we may or may not wish to replace "NC" or "NP" with a missing value, but this is something to keep in mind.

Reading Image Features

```
imFeatures <- read_xlsx(
  "/Users/grantcarr/Documents/Michigan/BDSI2025/Data/Imaging_Features.xlsx",
  sheet = 1
)
```

In general, the featureCitations.docx file describes the features and provides citations. The features are grouped into the categories listed below.

Breast and fibroglandular tissue (FGT) volume features: volume and density of breast area and fibroglandular area.

Tumor size and morphology: regularity/roundness vs irregularity of tumor shape and size.

FGT enhancement: measure how much FGT is enhanced when we add contrast. This is referred to as BPE, background parenchymal enhancement, which may confound tumor enhancement/identification.

Tumor enhancement: measure how much tumor is enhanced when we add contrast

Combining tumor and FGT enhancement: measure how tumor and FGT are enhanced when adding contrast

FGT enhancement texture: describe the enhancement due to contrast. Looking at local patterns, does enhancement look gritty or smooth?

Tumor enhancement texture: same as FGT but for tumor area.

Tumor Enhancement Spatial Heterogeneity: measure similarity between tumor subregions of the tumor.

FGT enhancement variation: global variation of contrast enhancement of FGT

Tumor enhancement variation: global variation of contrast enhancement of tumor

Task 1: What is in the data?

How many patients are in the dataset?

```
nrow(clinData)
```

```
## [1] 922
```

How many missing values are there among image features?

```
sum(is.na(imFeatures))
```

```
## [1] 0
```

How many missing values are there among clinical features?

```
sum(is.na(clinData))
```

```
## [1] 2056
```

```
missingness <- apply(
  clinData, 2, function(x){
    sum(is.na(x))
  }
)
sum(missingness == 0) # there are 86 variables with no missingness
```

```
## [1] 86
```

```
sum(missingness != 0) # the remaining 11 variables have some missing values
```

```
## [1] 11
```

```
missingness[missingness != 0]
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##                                                                                                    Days to las
##
## Age at last contact in EMR f/u(days)(from the date of diagnosis) ,last time patient known to be alive
##
```

Some clinical variables have minimal missing values, such as "Contrast Agent" (5). Others have so many missing values that the variable is statistically useless, such as "Contrast Bolus Volume (mL)" (653).

Task 2: Visualizing Data

Tabulate some discrete clinical variables and plot some continuous clinical variables. Take note of any abnormalities such as missingness, low data in a category, or significant skew in continuous variables.

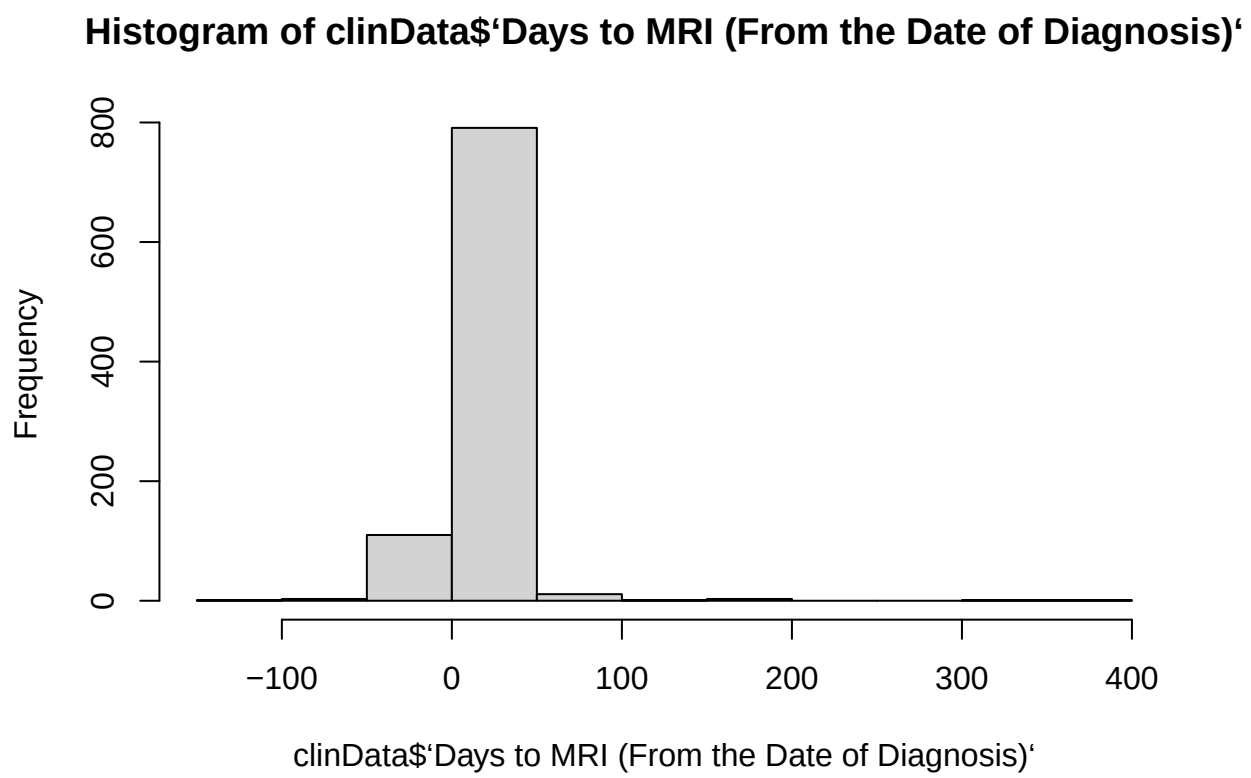
```r
apply(clinData, 2, function(x){
  sum(is.na(x)) + sum(x %in% c("NA", "NC", "NP"))
}) %>% table()
```

```
## .
##   0   2   5   6   7   8   9  15  21  24  25  29  30  43  47  49  51  54 271 276
##  33   1   1   1   2   1   1   1   1   1   2   2   2   1   1   1   1   1   3   1
## 284 294 610 652 653 657 661 765 786 789 808 812 825 846 851 858 860 879 886 894
##   1   1   6   1   2   1   1   1   1   1   1   1   1   1   2   1   1   1   1   3
## 895 902 905 906 908 920
##   1   1   1   1   1   5
```

```r
sum(is.na(clinData$`Days to MRI (From the Date of Diagnosis)`))
```

```
## [1] 0
```

```r
hist(clinData$`Days to MRI (From the Date of Diagnosis)`)
```




```r
sum(clinData$`Days to MRI (From the Date of Diagnosis)` > 0) # 808
```

```
## [1] 808
```

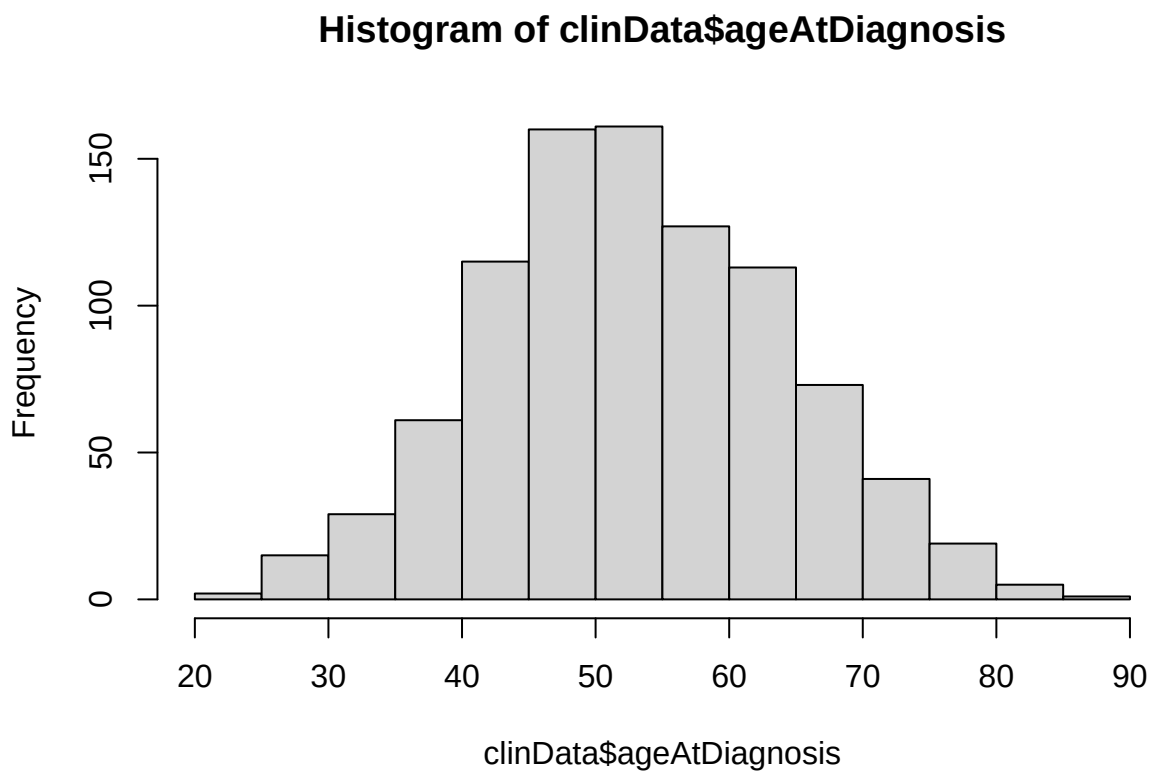
```r
sum(clinData$`Days to MRI (From the Date of Diagnosis)` == 0) # 9
```

```
## [1] 9
```

```r
sum(clinData$`Days to MRI (From the Date of Diagnosis)` < 0) # 105
```

```
## [1] 105
```

```r
#' everyone gets MRI at different times. knowledge of when other clinical
#' data is collected may be important in understanding whether
#' image features can reflect relevant information

clinData <- clinData %>% mutate(
  ageAtDiagnosis = -1*as.numeric(`Date of Birth (Days)`)/365
)
hist(clinData$ageAtDiagnosis)
```




```r
summary(clinData[,"ageAtDiagnosis"])
```

```
##  ageAtDiagnosis
##  Min.   :21.75
##  1st Qu.:45.43
##  Median :52.25
```

```
##  Mean   :52.98
##  3rd Qu.:60.83
##  Max.   :89.49
```

```
clinData <- clinData %>% mutate(
  raceEth = case_when(
    `Race and Ethnicity` == 0 ~ NA,
    `Race and Ethnicity` == 1 ~ "White",
    `Race and Ethnicity` == 2 ~ "Black",
    `Race and Ethnicity` == 3 ~ "Asian",
    `Race and Ethnicity` == 4 ~ "Native",
    `Race and Ethnicity` == 5 ~ "Hispanic",
    `Race and Ethnicity` == 6 ~ "Multi",
    `Race and Ethnicity` == 7 ~ "Hawaiian",
    `Race and Ethnicity` == 8 ~ "American Indian",
    .default = "?"
  )
)
table(clinData$raceEth)
```

```
##
## American Indian           Asian           Black        Hawaiian        Hispanic
##               4              14             203               1              18
##           Multi          Native           White
##               9               3             651
```

```
clinData <- clinData %>%
  mutate(
    Subtype = factor(
      case_when(
        `Mol Subtype` == 0 ~ "Luminal",
        `Mol Subtype` == 1 ~ "ER/PR+ and HER2+",
        `Mol Subtype` == 2 ~ "HER2+",
        .default = "Triple Negative"
      ), levels = c("Luminal", "ER/PR+ and HER2+", "HER2+", "Triple Negative")
    )
  )

unique(clinData[clinData$Subtype == "Luminal", c("ER", "PR", "HER2")])
```

```
## # A tibble: 3 x 3
##   ER    PR    HER2
##   <chr> <chr> <chr>
## 1 1     1     0
## 2 1     0     0
## 3 0     1     0
```

```
unique(clinData[clinData$Subtype == "ER/PR+ and HER2+", c("ER", "PR", "HER2")])
```

```
## # A tibble: 3 x 3
##   ER    PR    HER2
##   <chr> <chr> <chr>
```

```
## 1 1     0     1
## 2 1     1     1
## 3 0     1     1
```

```
unique(clinData[clinData$Subtype == "HER2+", c("ER", "PR", "HER2")])
```

```
## # A tibble: 1 x 3
##   ER    PR    HER2
##   <chr> <chr> <chr>
## 1 0     0     1
```

```
unique(clinData[clinData$Subtype == "Triple Negative", c("ER", "PR", "HER2")])
```

```
## # A tibble: 1 x 3
##   ER    PR    HER2
##   <chr> <chr> <chr>
## 1 0     0     0
```

```
#' Luminal: at least one ER+/PR+ but not HER2+
#' ER/PR and HER2: HER2+ and at least one ER+/PR+
#' HER2+: HER2+ only
#' triple negative: ER-/PR-/HER2-
table(clinData["Subtype"])
```

```
## Subtype
##          Luminal ER/PR+ and HER2+            HER2+  Triple Negative
##              595              104               59              164
```

```
# dominated by luminal subtype, some sparsity in others

table(clinData["Staging(Tumor Size)# [T]"])
```

```
## Staging(Tumor Size)# [T]
##   1   2   3   4  NA
## 409 395  90  22   6
```

```
# dominated by stage 1/2, sparsity in stages 3/4
table(clinData["Staging(Nodes)#(Nx replaced by -1)[N]"])
```

```
## Staging(Nodes)#(Nx replaced by -1)[N]
##   0   1   2   3  NA
## 529 265  61  43  24
```

```
# dominated by stage 0/1, sparsity in stages 2/3
table(clinData["Staging(Metastasis)#(Mx -replaced by -1)[M]"])
```

```
## Staging(Metastasis)#(Mx -replaced by -1)[M]
##  -1   0   1
## 203 689  30
```

```r
# dominated by 0 (no mets) and -1 (can't be evaluated)

#' in general, early stage disease

table(clinData["TumorGradeT"], useNA = "if")
```

```
## TumorGradeT
##   1   2   3  NA
##  68 144 695  15
```

```r
table(clinData["TumorGradeN"], useNA = "if")
```

```
## TumorGradeN
##   1   2   3  NA
##  54 401 460   7
```

```r
table(clinData["TumorGradeM"], useNA = "if")
```

```
## TumorGradeM
##   1   2   3  NA
## 546 201 154  21
```

```r
#' very little missingness for tumor grade
#' high Tubule/Nuclear grade, low Mitotic grade

table(clinData["Nottingham grade"], useNA = "if")
```

```
## Nottingham grade
##    1    2    3   NA <NA>
##  113  318  207   13  271
```

```r
table(clinData["Histologic type"], useNA = "if")
```

```
## Histologic type
##    0    1    2    3    5    9   NA <NA>
##    1  575   63    1    2    4    5  271
```

```r
#' high degree of missingness in these variables

table(clinData[,"Bilateral Information"])
```

```
## Bilateral Information
##   0   1  NC
## 623  28 271
```

```r
# lots of missingness, and very few with bilateral disease

table(clinData["Surgery"], useNA = "if")
```

```
## Surgery
##   0   1  NA
##  36 879   7
```

```
# nearly all patients had surgery
clinData <- clinData %>% mutate(
  surgeryDays = as.numeric(`Days to Surgery (from the date of diagnosis)`)
)
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘surgeryDays = as.numeric(‘Days to Surgery (from the date of
##   diagnosis)‘)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```
sum(is.na(clinData["surgeryDays"])) # 47 missing values
```

```
## [1] 47
```

```
sum(complete.cases(clinData[ , c("surgeryDays", "Surgery")]))
```

```
## [1] 875
```

```
# 875 observations for time to surgery
table(clinData["Definitive Surgery Type"], useNA = "if")
```

```
## Definitive Surgery Type
##   0   1  NA  NP
## 463 416   7  36
```

```
# 50/50 BCS vs mastectomy

sum(
  clinData["Neoadjuvant Radiation Therapy"] == 1, na.rm = T
) # 22 patients neoadjuvant radiation therapy (given before surgery)
```

```
## [1] 22
```

```
sum(
  clinData["Adjuvant Radiation Therapy"] == 1, na.rm = T
) # 614 patients adjuvant radiation therapy (given after surgery)
```

```
## [1] 614
```

```
clinData$dead <- ifelse(
  clinData$`Days to death (from the date of diagnosis)` == "NP",
  0, 1
) # if days to death is not pertinent, then they did not die
table(clinData["dead"]) # 62 deaths
```

```
## dead
##   0   1
## 860  62
```

```
clinData$survDays <- ifelse(
  clinData$dead == 1, as.numeric(clinData$`Days to death (from the date of diagnosis)`),
  pmax(
    as.numeric(clinData$`Days to last distant recurrence free assemssment(from the date of diagnosis)`)
    as.numeric(clinData$`Days to last local recurrence free assessment (from the date of diagnosis)`),
    as.numeric(clinData$`Age at last contact in EMR f/u(days)(from the date of diagnosis) ,last time pa
  )

)
```

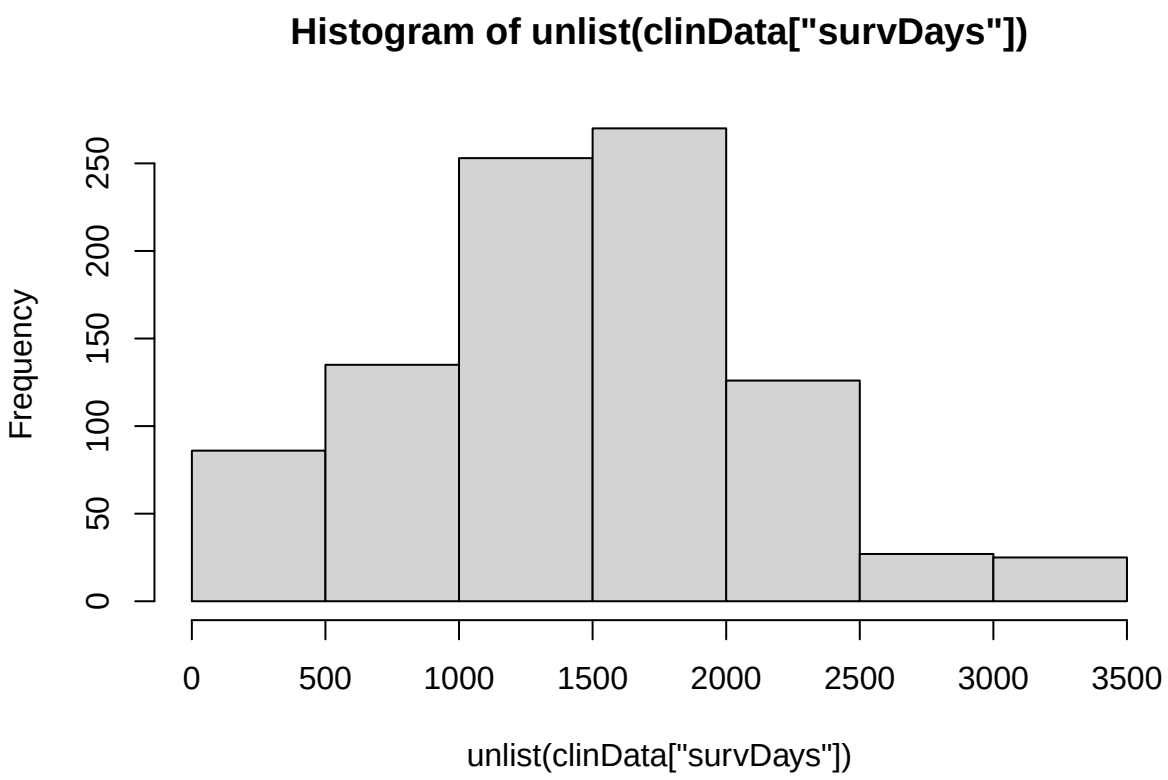
```
## Warning in ifelse(clinData$dead == 1, as.numeric(clinData$‘Days to death (from
## the date of diagnosis)‘), : NAs introduced by coercion
```

```
## Warning in pmax(as.numeric(clinData$‘Days to last distant recurrence free
## assemssment(from the date of diagnosis)‘), : NAs introduced by coercion
## Warning in pmax(as.numeric(clinData$‘Days to last distant recurrence free
## assemssment(from the date of diagnosis)‘), : NAs introduced by coercion
```

```
#' if dead, then use days to death
#' if they did not die, then use the last time we know any information
#' about them being alive. pmax = parallel maximum of vectors, pmax of
#' last local recurrence free assessment, last distant recurrence free
#' assessment, and last contact in electronic medical record
summary(clinData["survDays"]/365) # IQR 2.8-5.2, median 4 years
```

```
##     survDays
##  Min.   :0.000
##  1st Qu.:2.822
##  Median :4.074
##  Mean   :4.034
##  3rd Qu.:5.218
##  Max.   :9.512
```

```
hist(unlist(clinData["survDays"]))
```




```
sum(complete.cases(clinData[ , c("survDays", "dead")]))
```

```
## [1] 922
```

```
# 922 observations for time to death

clinData$recurrence <- as.numeric(clinData$`Recurrence event(s)`)
#' make recurrence a numeric indicator
table(clinData["recurrence"], useNA = "if") # 87 recurrence, 2 missing
```

```
## recurrence
##    0    1 <NA>
##  833   87    2
```

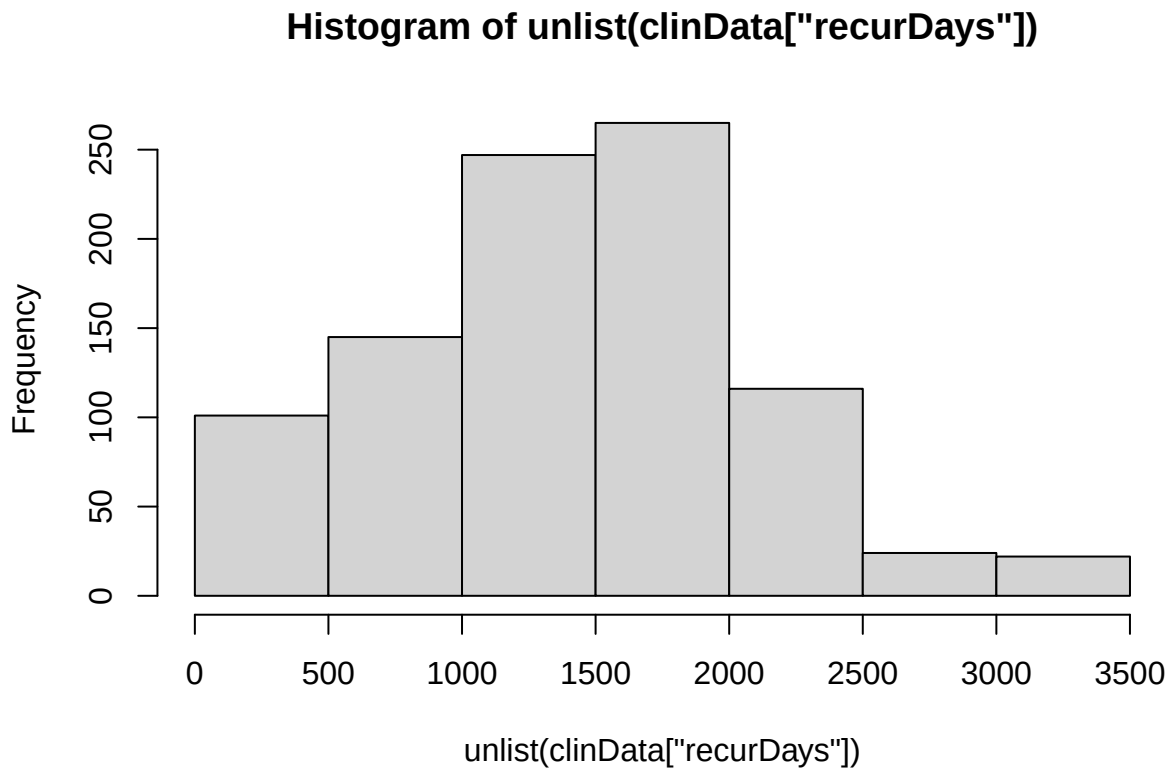
```
clinData$recurDays <- ifelse(
  clinData$recurrence == 0, as.numeric(clinData$survDays),
  ifelse(
    clinData$recurrence == 1,
    pmin(
    as.numeric(clinData$`Days to distant recurrence(from the date of diagnosis)`),
    as.numeric(clinData$`Days to local recurrence (from the date of diagnosis)`),
    na.rm = T
  ),
    NA
```

```
  )

)
```

```
## Warning in pmin(as.numeric(clinData$`Days to distant recurrence(from the date
## of diagnosis)`), : NAs introduced by coercion
```

```
## Warning in pmin(as.numeric(clinData$`Days to distant recurrence(from the date
## of diagnosis)`), : NAs introduced by coercion
```

```
#' if no recurrence, then days to recurrence is just survival days
#' if recurrence, then days to recurrence is minimum of days to local
#' recurrence and days to distant recurrence
#' if recurrence is missing, then days to recurrence is missing
hist(unlist(clinData["recurDays"]))
```




```
sum(complete.cases(clinData[ , c("recurDays", "recurrence")]))
```

```
## [1] 920
```

```
# 920 observations for time to recurrence

table(clinData[ , "Neoadjuvant Chemotherapy"])
```

```
## Neoadjuvant Chemotherapy
##   0   1  NA
## 601 292  29
```

```
table(clinData[ , "Adjuvant Chemotherapy"])
```

```
## Adjuvant Chemotherapy
##   0   1  NA
## 537 336  49
```

```
# many with neoadjuvant or adjuvant chemotherapy

table(clinData[ , "Neoadjuvant Anti-Her2 Neu Therapy"], useNA = "if")
```

```
## Neoadjuvant Anti-Her2 Neu Therapy
##   0   1  NA
## 809  83  30
```

```
table(clinData[ , "Adjuvant Anti-Her2 Neu Therapy"], useNA = "if")
```

```
## Adjuvant Anti-Her2 Neu Therapy
##   0   1  NA
## 733 138  51
```

```
# few with neoadjuvant or adjuvant anti-Her2

table(clinData[,"Received Neoadjuvant Therapy or Not"], useNA = "if")
```

```
## Received Neoadjuvant Therapy or Not
##   1   2  NA
## 312 581  29
```

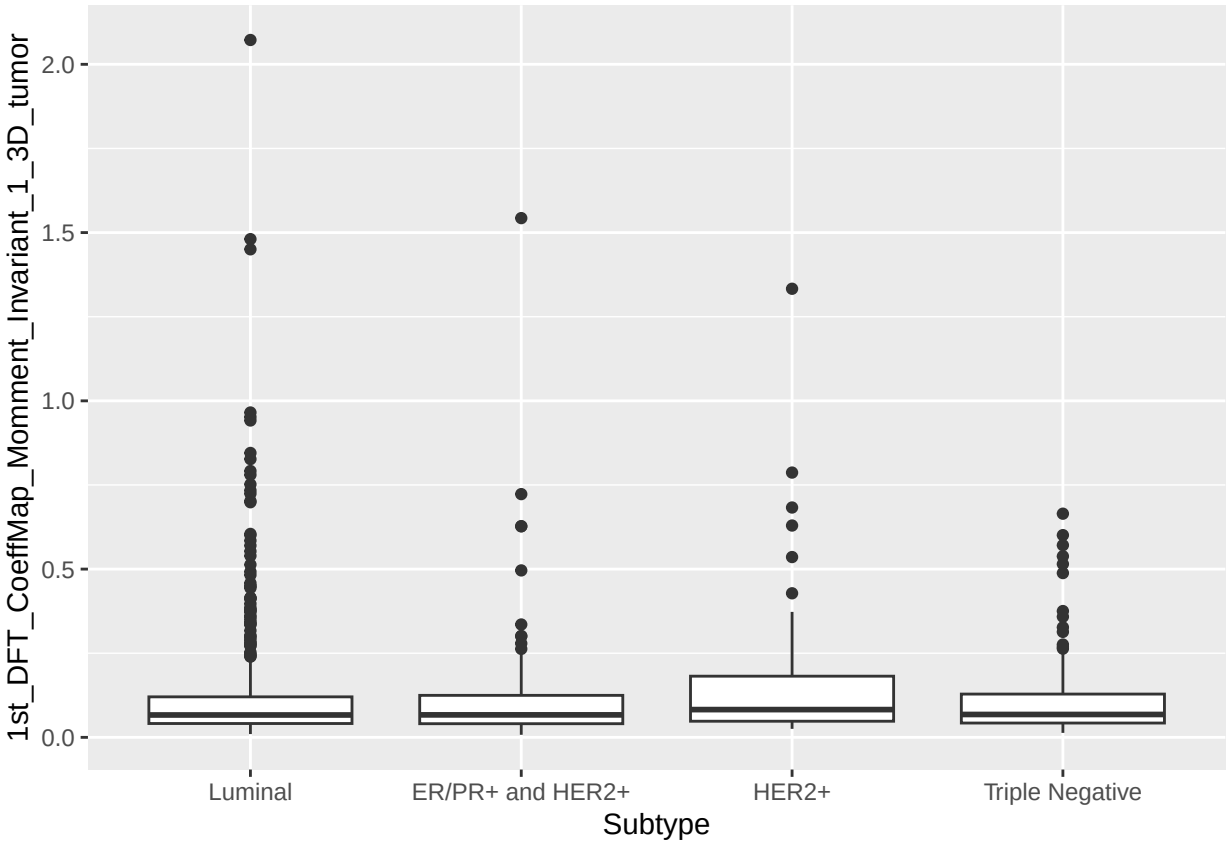
```
# overall, most patients did not receive any type of neoadjuvant therapy
```

Pick a group of image features as they are grouped in the featureDocumentation.docx file. Explore whether there is an association between the image feature(s) and any clinical variable(s) that you are interested in. You can use scatterplots or any visualization tool you see fit.

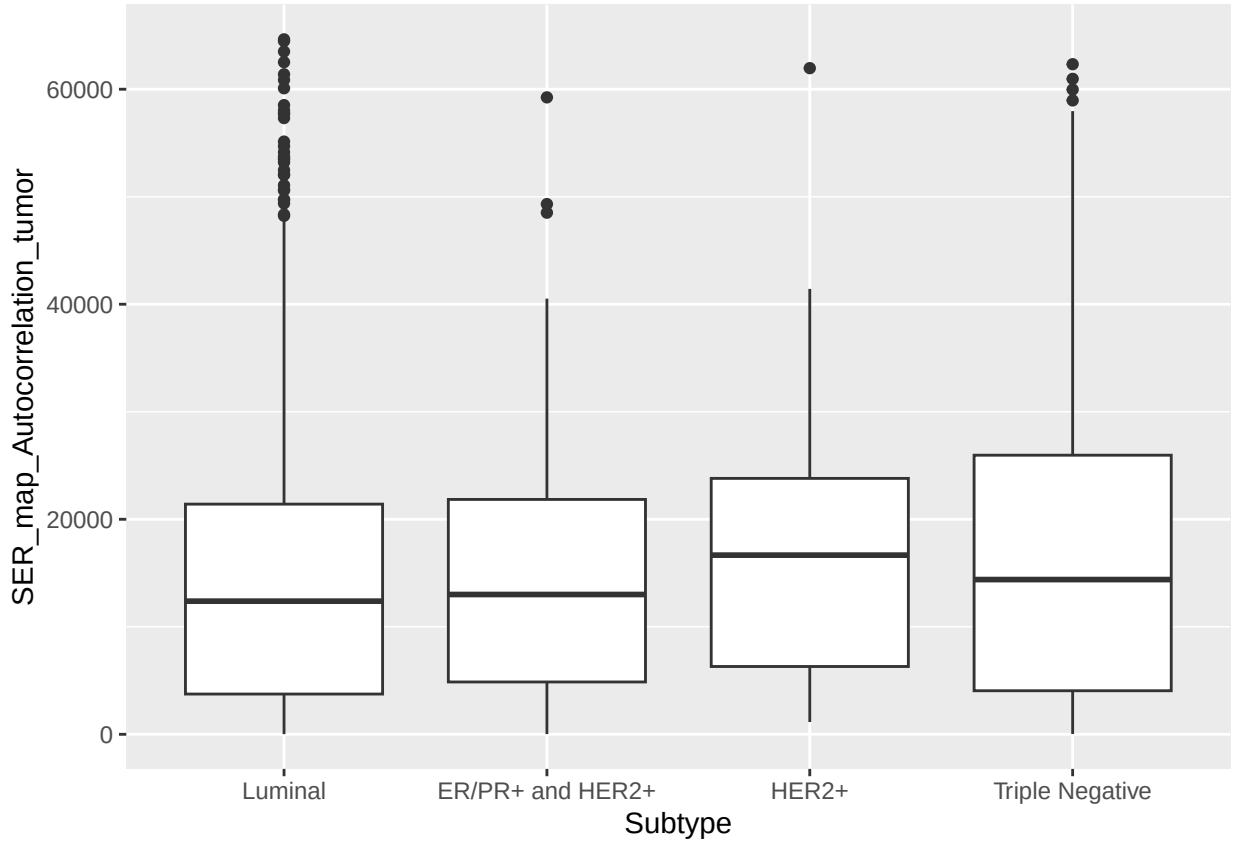
```
#' I am interested in tumor enhancement texture and its relationship
#' to tumor subtype
tumorEnhanceTexture_vars <- colnames(imFeatures)[
  c(189:232, 279:281, 287:308, 313:334, 339:360, 27:48)
]
# manually identify all variables in the tumor enhancement texture group,
# 135 of them

fullData <- left_join(
  clinData, imFeatures
)
```

```
## Joining with `by = join_by(`Patient ID`)`
```

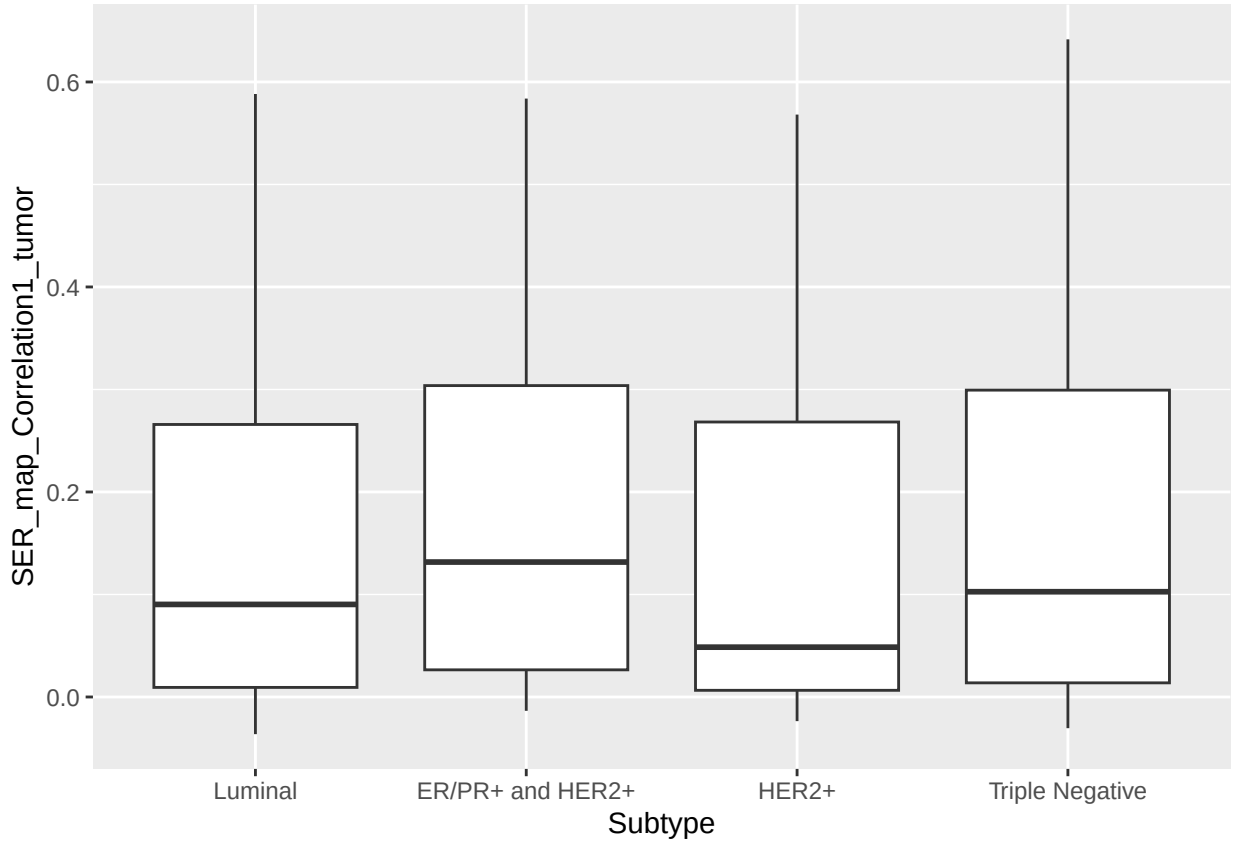
```
subData <- fullData %>%
  select(Subtype, all_of(tumorEnhanceTexture_vars))
ggplot(
  subData,
  aes(x = Subtype, y = `1st_DFT_CoeffMap_Momment_Invariant_1_3D_tumor`)
) +
  geom_boxplot()
```




```
ggplot(
  subData,
  aes(x = Subtype, y = SER_map_Autocorrelation_tumor)
) +
  geom_boxplot()
```



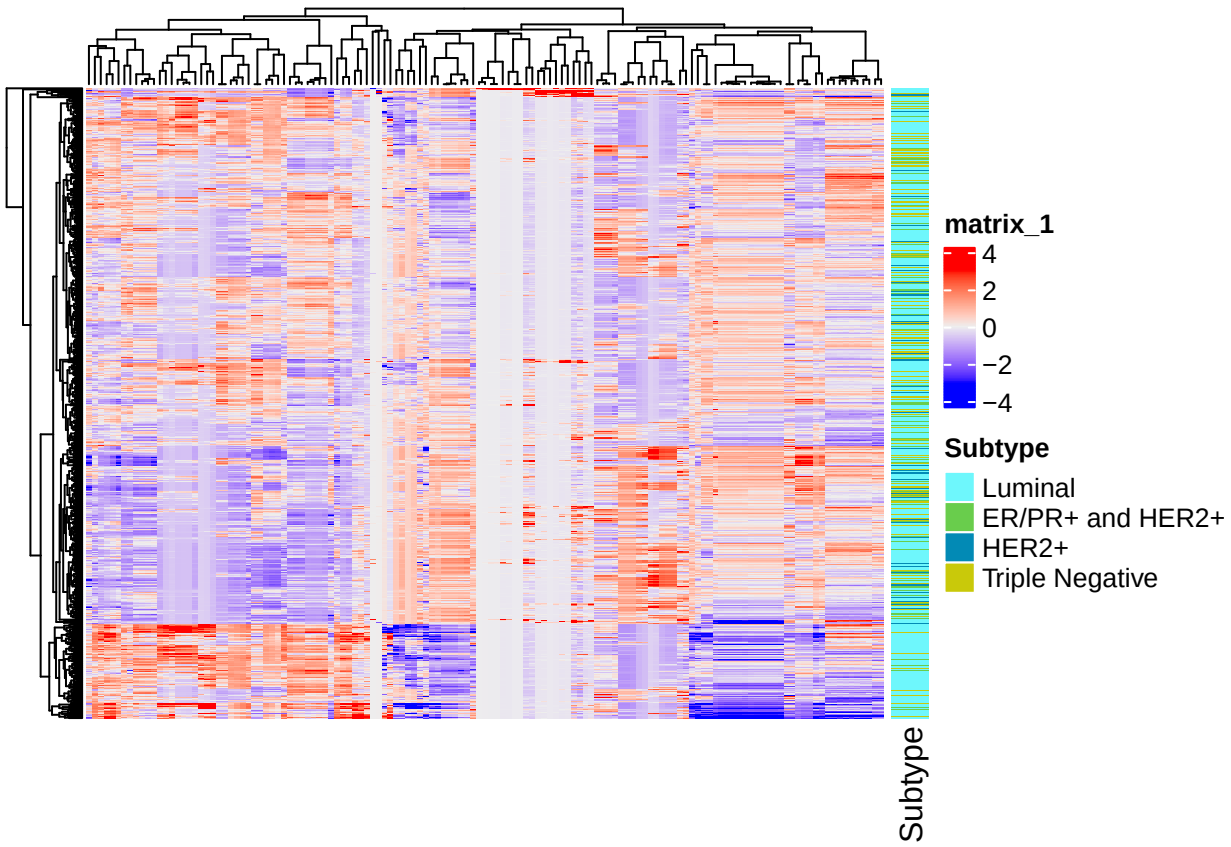

```
ggplot(
  subData,
  aes(x = Subtype, y = SER_map_Correlation1_tumor)
) +
  geom_boxplot()
```



Next, find a function from an R package that can create a "heat map" and see if the features cluster together. Overlay your clustering with a clinical feature. Does the clustering of image features seem to correlate with the clinical feature?

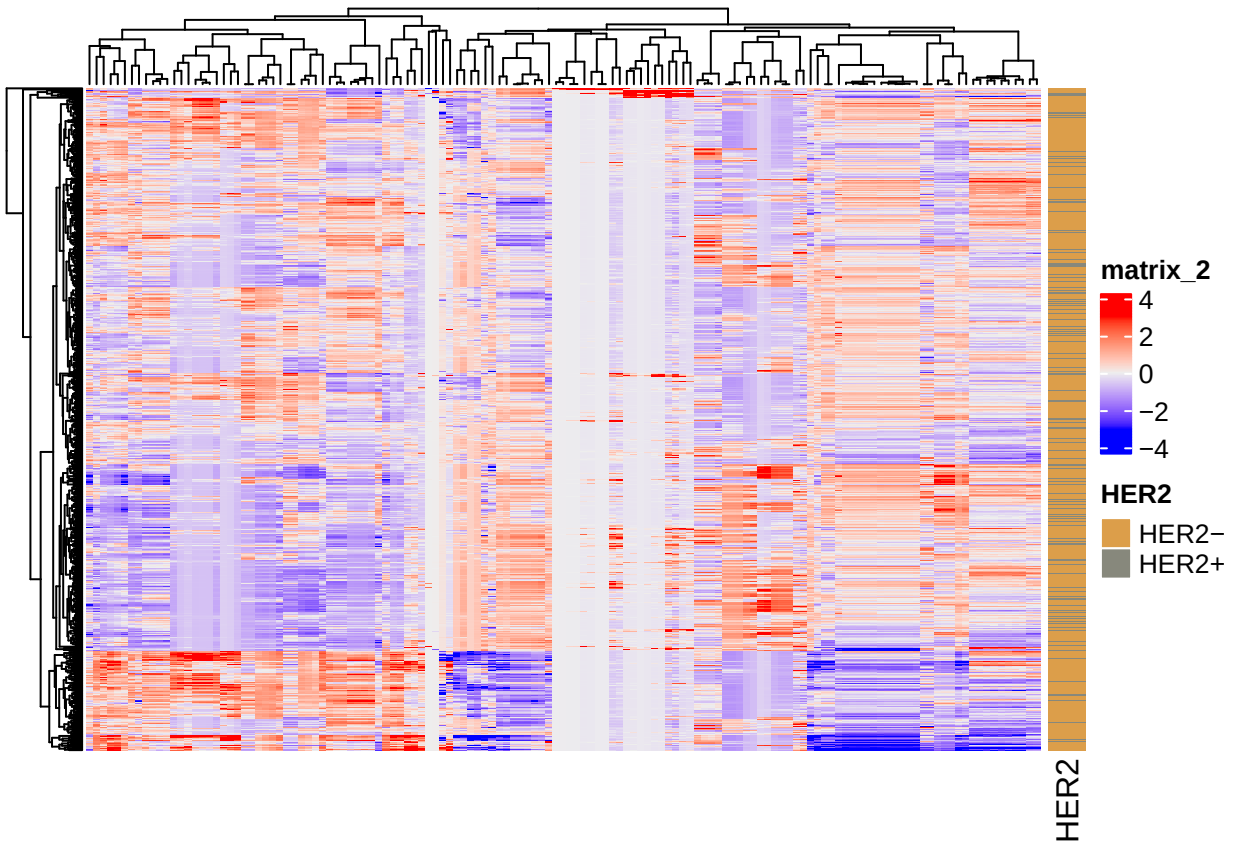
```
heatMatrix <- scale(subData[,-1])

row_ha <- rowAnnotation(Subtype = subData$Subtype)
Heatmap(
  heatMatrix, column_labels = rep("", ncol(heatMatrix)),
  right_annotation = row_ha
)
```

```
## The automatically generated colors map from the minus and plus 99^th of
## the absolute values in the matrix. There are outliers in the matrix
## whose patterns might be hidden by this color mapping. You can manually
## set the color to ‘col‘ argument.
##
## Use ‘suppressMessages()‘ to turn off this message.
```



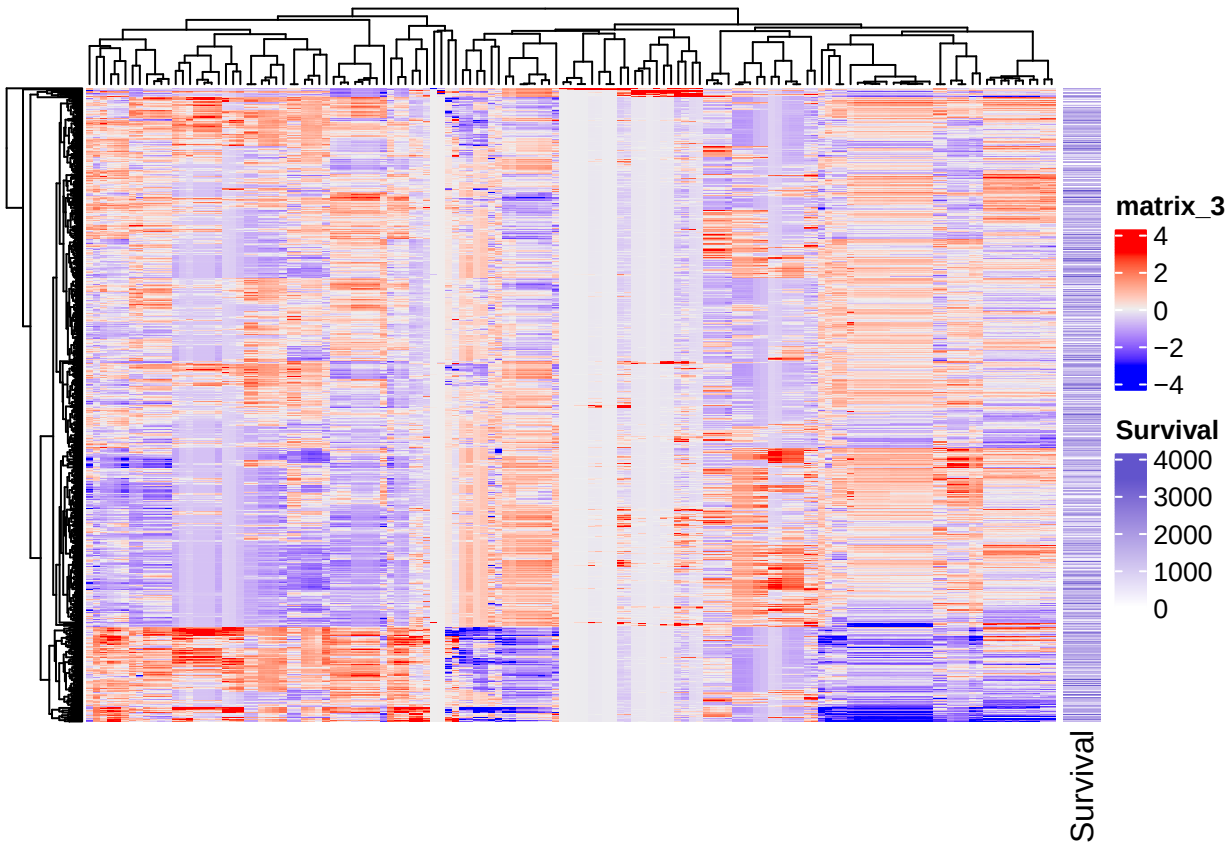

```
row_ha_her2 <- rowAnnotation(HER2 = ifelse(fullData$HER2 == 1, "HER2+", "HER2-"))
Heatmap(
  heatMatrix, column_labels = rep("", ncol(heatMatrix)),
  right_annotation = row_ha_her2
)
```

```
## The automatically generated colors map from the minus and plus 99^th of
## the absolute values in the matrix. There are outliers in the matrix
## whose patterns might be hidden by this color mapping. You can manually
## set the color to ‘col‘ argument.
##
## Use ‘suppressMessages()‘ to turn off this message.
```




```
row_ha_surv <- rowAnnotation(Survival = fullData$survDays)
Heatmap(
  heatMatrix, column_labels = rep("", ncol(heatMatrix)),
  right_annotation = row_ha_surv
)
```

```
## The automatically generated colors map from the minus and plus 99^th of
## the absolute values in the matrix. There are outliers in the matrix
## whose patterns might be hidden by this color mapping. You can manually
## set the color to ‘col‘ argument.
##
## Use ‘suppressMessages()‘ to turn off this message.
```

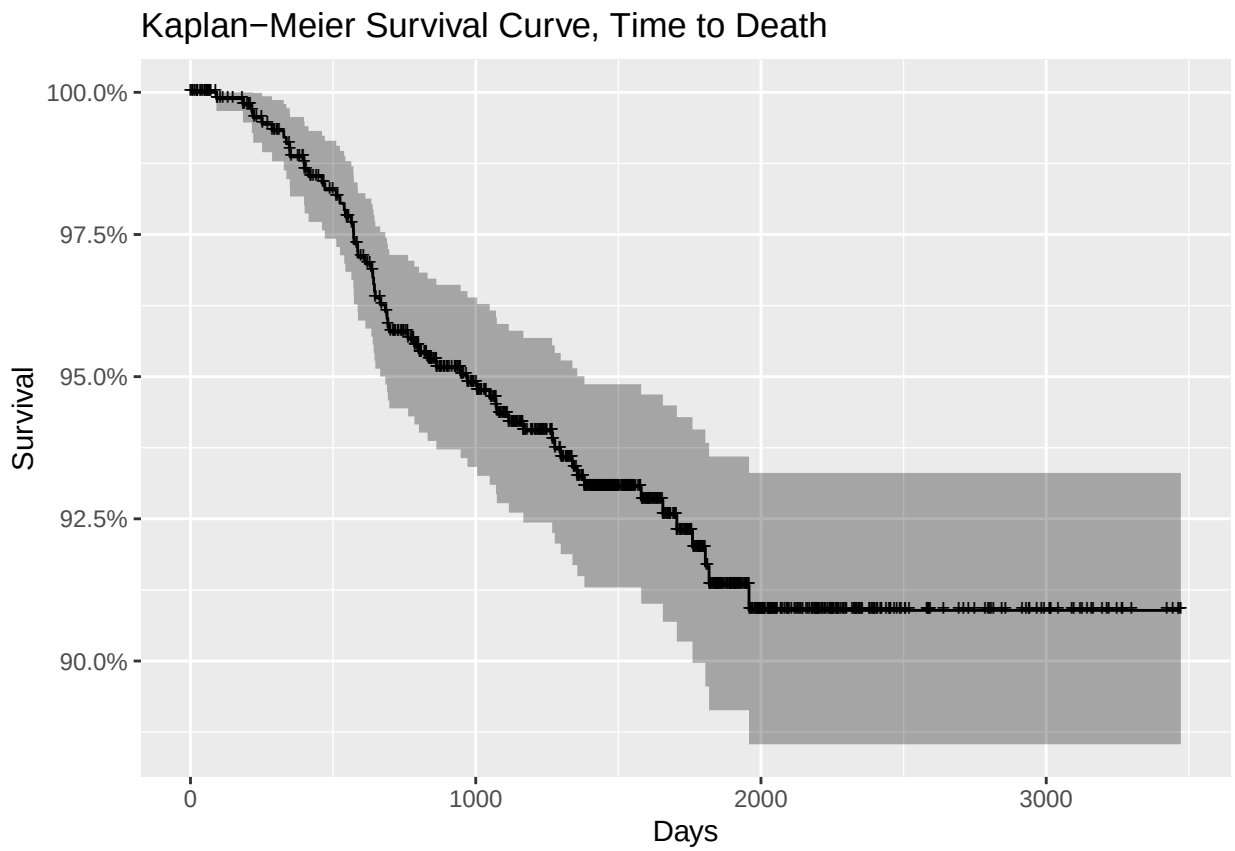


Heat maps can be a useful tool for finding broad patterns in data. In the above example, there may be some clustering of textural features, but the clustering does not correspond to any clinical outcomes. Clustering does not provide rigorous statistical evidence of associations, but it can be a useful exploratory step.

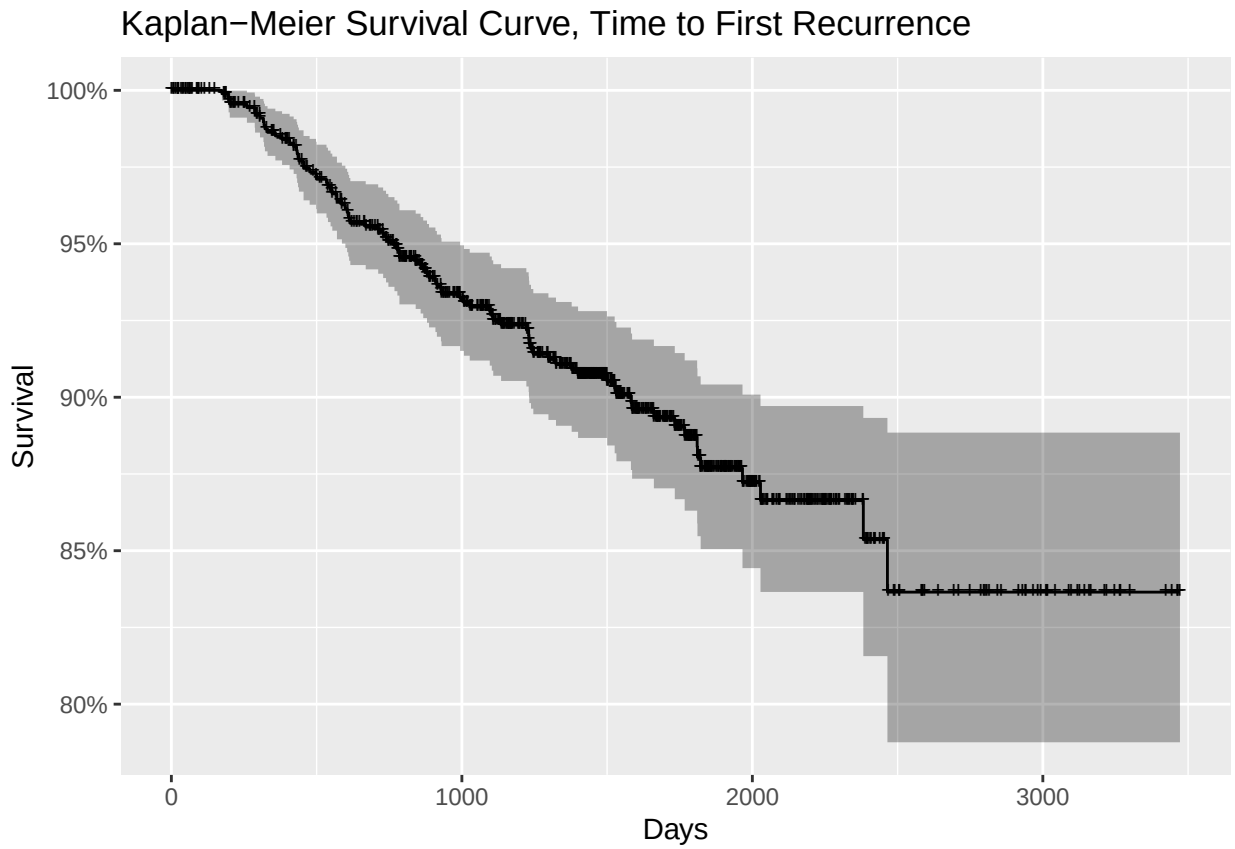
Task 3: Survival Information

Fit a Kaplan-Meier survival curve to the time to death and time to recurrence data.

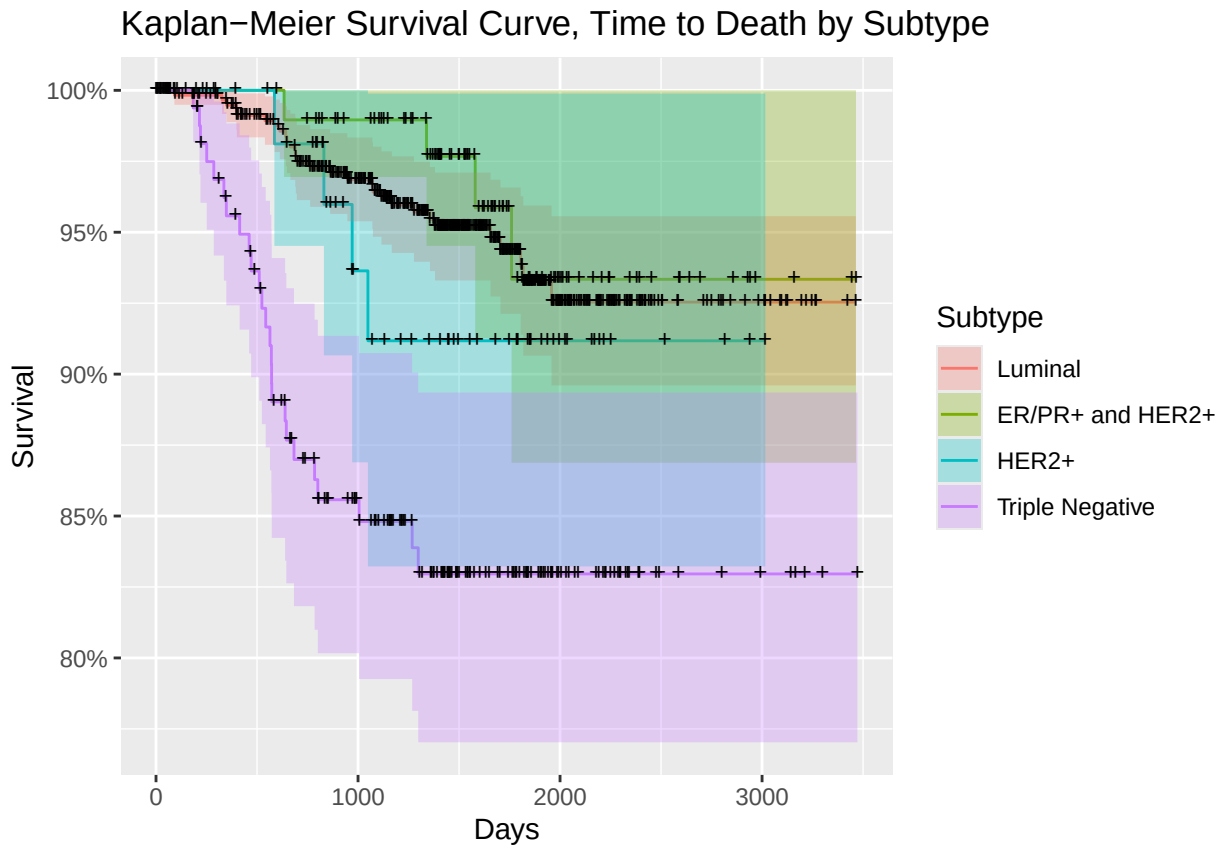
```
deathFit <- survfit(Surv(survDays, dead) ~ 1, data = clinData)
autoplot(deathFit) +
  labs(title = "Kaplan-Meier Survival Curve, Time to Death",
       x = "Days", y = "Survival")
```



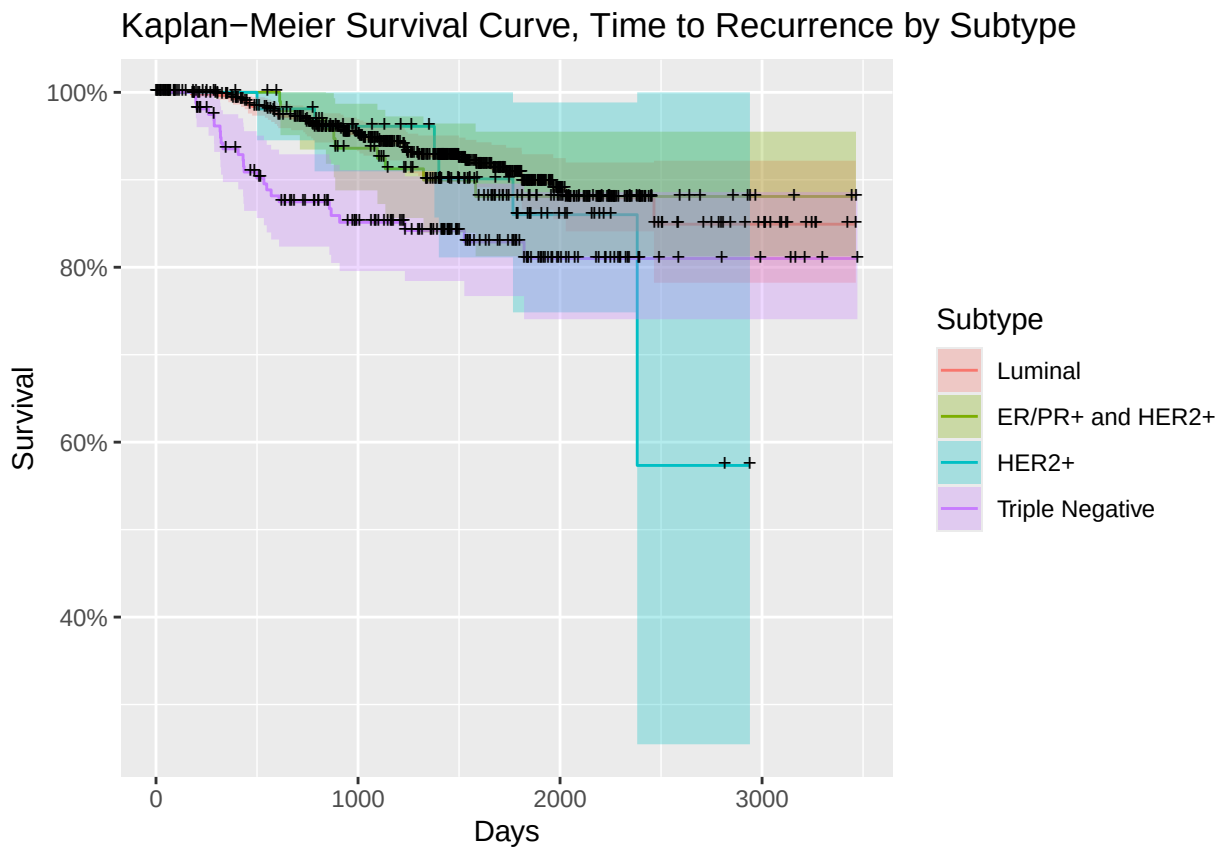

```
recurFit <- survfit(Surv(recurDays, recurrence) ~ 1, data = clinData)
autoplot(recurFit) +
  labs(title = "Kaplan-Meier Survival Curve, Time to First Recurrence",
       x = "Days", y = "Survival")
```




```
clinData <- clinData %>%
  mutate(
    Subtype = factor(
      case_when(
        `Mol Subtype` == 0 ~ "Luminal",
        `Mol Subtype` == 1 ~ "ER/PR+ and HER2+",
        `Mol Subtype` == 2 ~ "HER2+",
        .default = "Triple Negative"
      ), levels = c("Luminal", "ER/PR+ and HER2+", "HER2+", "Triple Negative")
    )
  )

autoplot(update(deathFit, .~.+Subtype)) +
  labs(title = "Kaplan-Meier Survival Curve, Time to Death by Subtype",
       x = "Days", y = "Survival", col = "Subtype", fill = "Subtype")
```




```
autoplot(update(recurFit, .~.+Subtype)) +
  labs(title = "Kaplan-Meier Survival Curve, Time to Recurrence by Subtype",
       x = "Days", y = "Survival", col = "Subtype", fill = "Subtype")
```


Kaplan−Meier Survival Curve, Time to Recurrence by Subtype
100%
80%
60%
40%
Survival
0
1000
2000
3000
Days
Subtype
Luminal
ER/PR+ and HER2+
HER2+
Triple Negative