SYDE780 ASSIGNMENT 4 - R Module on Machine Learning Techniques

Benjamin Jingwen Luo (b33luo; 20890448)

2025 March 12

Load in the required packages and dataset, organize data

```
fileToImport = "~/Downloads/4B/Code/SYDE 780/dataset.csv"
Data.RSA = read.csv(fileToImport, header=TRUE)
```

For this assignment, we are going to look at 1-year MTPM migration and the change in migration from 1-2 years ("continuous migration"), so we will no longer have longitudinal data (i.e. now only 1 row per subject). Note that the change in migration from 1-2 years is generally used to define if the migration has stabilized (i.e. low migration over this period is stable, high is unstable.)

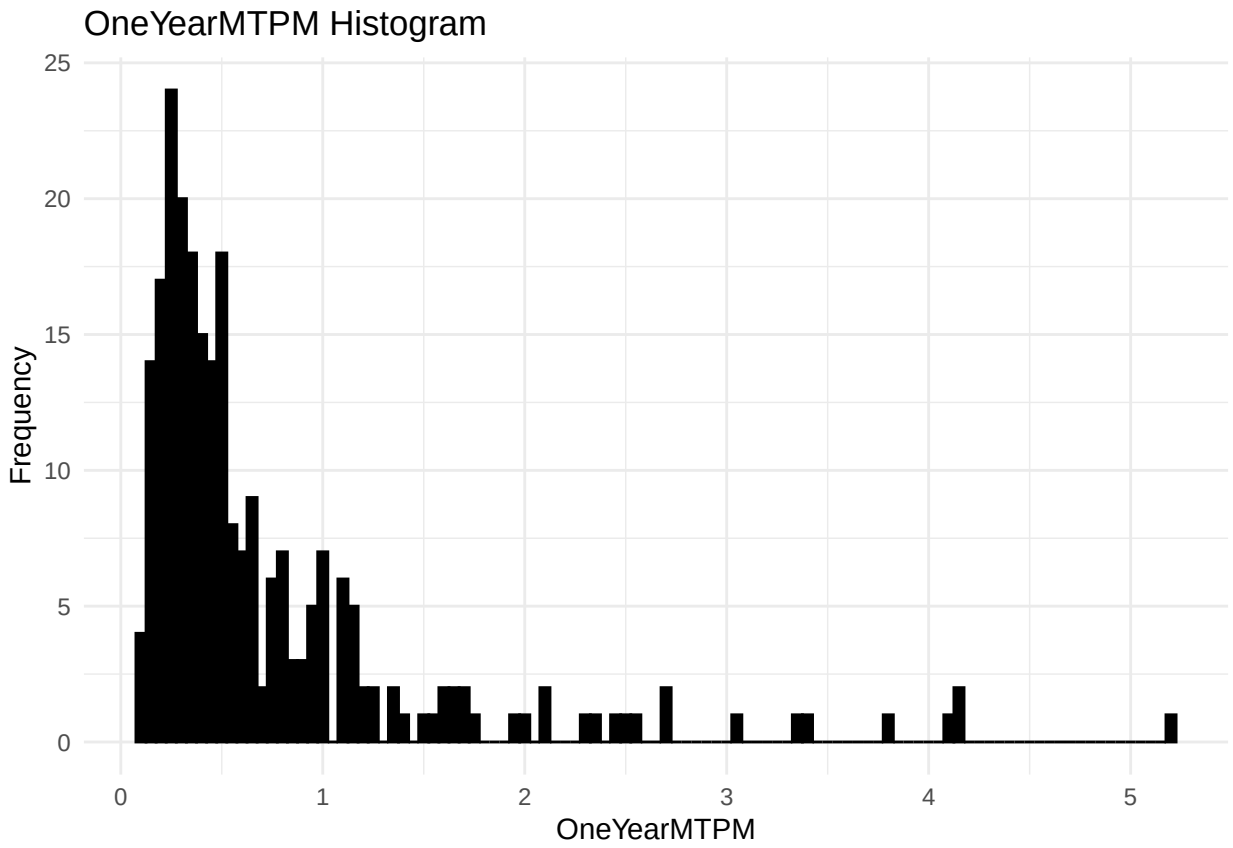
The first step is to calculate the change in migration from one to two years

```
#Make a dataframe with demographics and just one-year migration; rename MTPM as "OneYearMTPM"
Data.RSA.12Mon<-Data.RSA[which(Data.RSA$Months==12), ]
names(Data.RSA.12Mon)[names(Data.RSA.12Mon)=="MTPM"]<-"OneYearMTPM"

#Make a dataframe  just two-year migration; rename MTPM as "TwoYearMTPM".  Keep "Subject" for merging
Data.RSA.24Mon<-Data.RSA[which(Data.RSA$Months==24), ]
names(Data.RSA.24Mon)[names(Data.RSA.24Mon)=="MTPM"]<-"TwoYearMTPM"
Data.RSA.24Mon.MTPM<-Data.RSA.24Mon[ , (names(Data.RSA.24Mon) %in% c("Subject","TwoYearMTPM"))]

#Merge dataframes by Subject.  Drops any missing OneYearMTPM OR TwoYearMTPM (i.e. only keeps complete c
Data.RSA.OneTwo <- merge(Data.RSA.12Mon, Data.RSA.24Mon.MTPM, by = "Subject", all = FALSE)

#Calculate change in migration from 1 to 2 years
Data.RSA.OneTwo$OneToTwoMTPM <- Data.RSA.OneTwo$TwoYearMTPM - Data.RSA.OneTwo$OneYearMTPM

#This is the dataset we are going to use in further analysis
Data <- Data.RSA.OneTwo
```

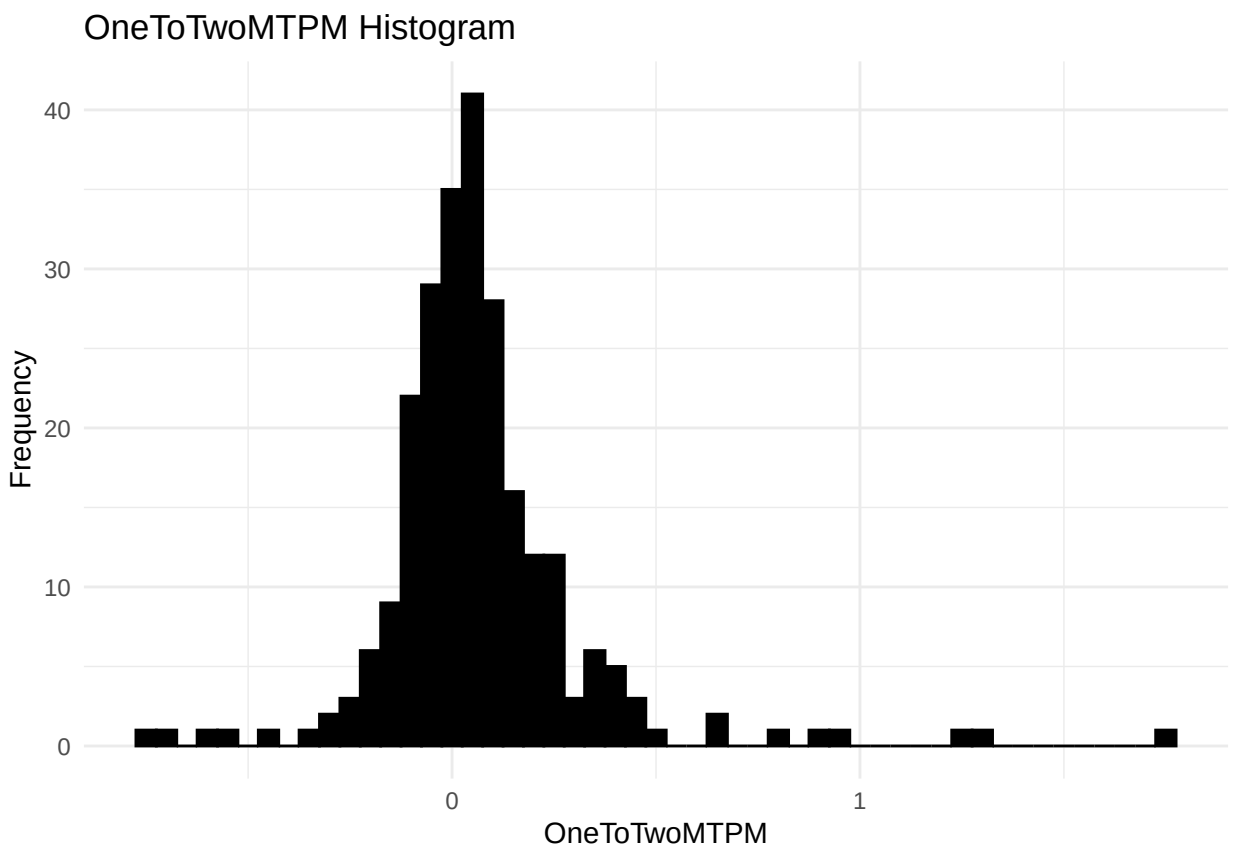
Assignment Question

PART 1 - Cluster Analysis

```
Data.Cluster = na.omit(Data[, c("Subject","Age","BMI","OneYearMTPM","OneToTwoMTPM","SexDesc","Fixation"]

# Histogram: OneYearMTPM
ggplot(Data, aes(x = OneYearMTPM)) +
  geom_histogram(binwidth = 0.05, fill = "black", color = "black") +
  labs(x = "OneYearMTPM", y = "Frequency", title = "OneYearMTPM Histogram") +
  theme_minimal()
```

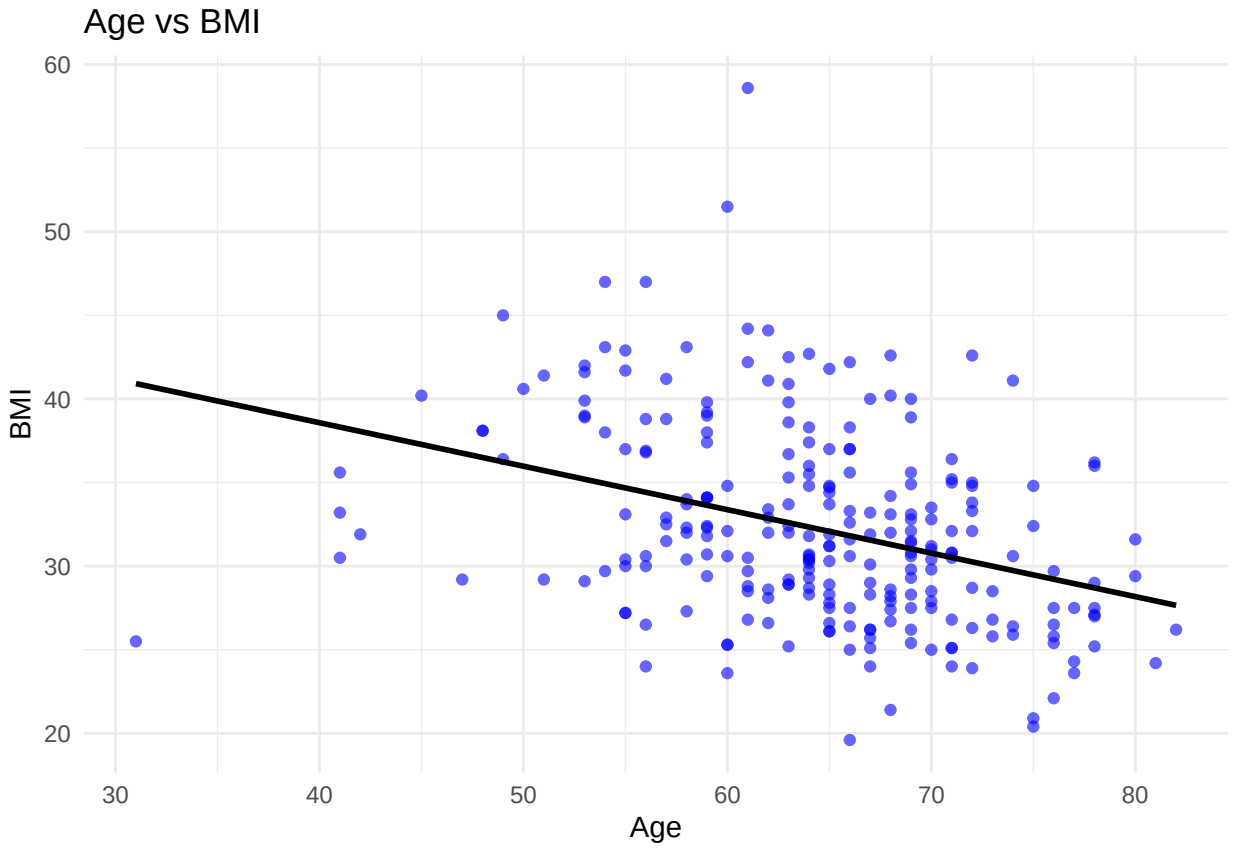



```
# Histogram: OneToTwoMTPM
ggplot(Data, aes(x = OneToTwoMTPM)) +
  geom_histogram(binwidth = 0.05, fill = "black", color = "black") +
  labs(x = "OneToTwoMTPM", y = "Frequency", title = "OneToTwoMTPM Histogram") +
  theme_minimal()
```




```
# Scatter plot: Age v BMI
ggplot(Data, aes(x = Age, y = BMI)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", color = "black", se = FALSE) +
  labs(x = "Age", y = "BMI", title = "Age vs BMI") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Q1. Using code from previous assignments, plot the raw data, at a minimum including a histogram of OneYearMTPM, a histogram of OneToTwoMTPM, and an examination of correlation between Age and BMI. Provide your assessments of the data. [5 marks]

- **OneYearMTPM**: The histogram shows a left-skewed normal distribution with mean around 0.3 mm
- **OneToTwoMTPM**: The histogram shows a normal distribution with mean around 0.05 mm. In general, MTPM increases from year 1 to 2, but MTPM may also decrease.
- **Age v BMI**: Age and BMI are negatively correlated but with high variability. A linear model appears to be appropriate, but further testing is required to confirm this

We will first try algorithms for mixed-type data

Code to implement Modha-Spangler weighting method on PAM

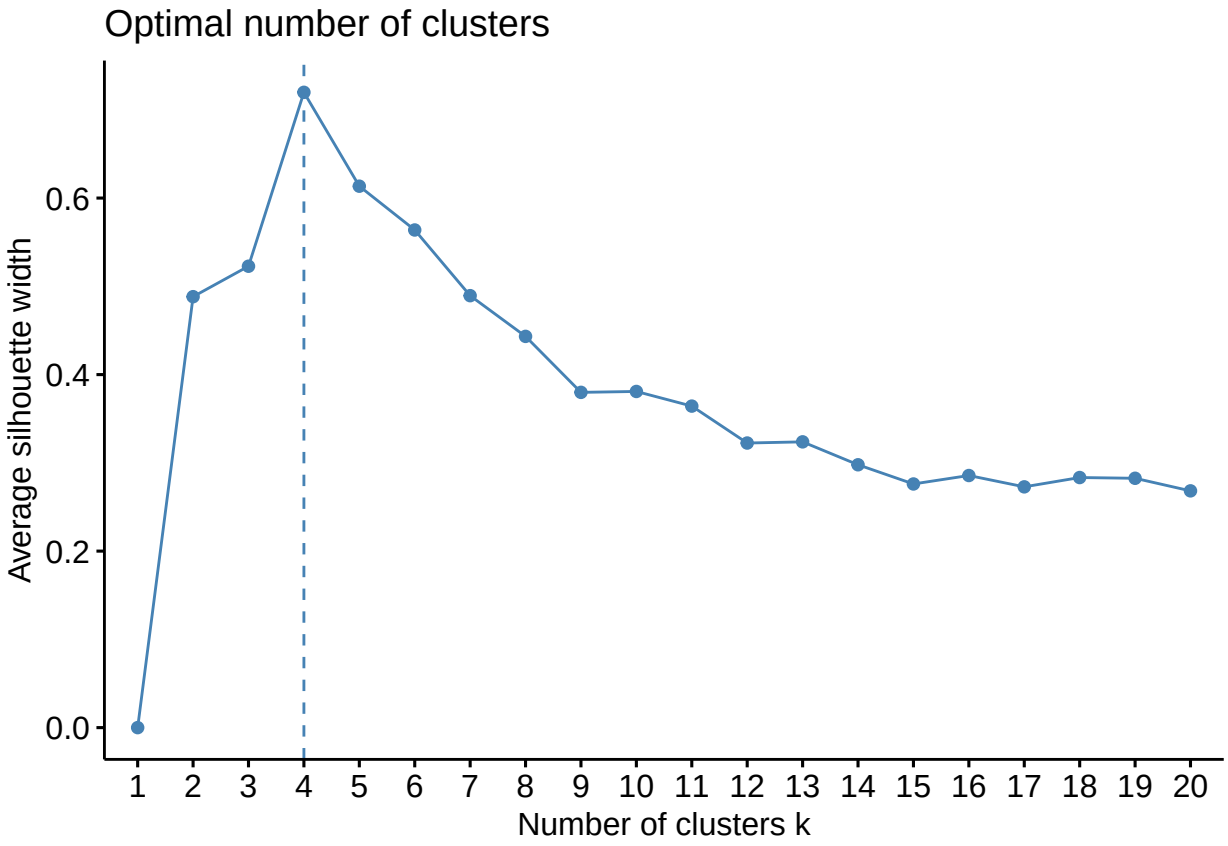
Compute Gower's Distance (PAM)

```
Data.Cluster[sapply(Data.Cluster, is.character)] = lapply(Data.Cluster[sapply(Data.Cluster, is.characte

#Function to Standardize numerical variables to range [0,1]
rangeStandardize <- function(x) {
(x - min(x)) / diff(range(x))
}
```

```r
#Separate Variables into continuous (standardized) and categorical
catVars = Data.Cluster[ ,sapply(Data.Cluster,is.factor)]
conVars = as.data.frame(lapply(Data.Cluster[ ,sapply(Data.Cluster,is.numeric) & colnames(Data.Cluster)



#CODE TO IMPLEMENT MODHA-SPANGLER WEIGHTING ON PAM

#Write functions implementing the L1 distance for continuous variables and matching distance for catego
L1Dist <- function(v1, v2) {
  sum(abs(v1 - v2))
  }
matchingDist <- function(v1, v2) {
  sum(as.integer(v1) != as.integer(v2))
}

#Wrapper for the pam function from cluster package, using the daisy function to implement Gower's dista
#This function will be an input argument to the gmsClust function. It must return a list containing at
pammix <- function(conData, catData, conWeight, nclust, ...) {
  conData <- as.data.frame(conData)
  catData <- as.data.frame(catData)

  distMat <- daisy(x = cbind(conData, catData), metric = "gower",
  weights = rep(c(conWeight, 1 - conWeight),
  times = c(ncol(conData), ncol(catData))))

  clustRes <- pam(x = distMat, k = nclust, diss = TRUE, ...)

  return(list(cluster = clustRes$clustering,
  conCenters = conData[clustRes$id.med, , drop = FALSE],
  catCenters = catData[clustRes$id.med, , drop = FALSE]))
}
###

### CALCULATE THE DISSIMILARITY MATRIX FOR CLASSIC PAM

gower_dist = daisy(cbind(conVars,catVars), metric = "gower", stand = FALSE)
gower_mat = as.matrix(gower_dist)
```

##Compute optimal number of (k) clusters for classic PAM and MS

```r
fviz_nbclust(gower_mat,cluster::pam,method = "silhouette", k.max = 20) #try the silhouette method
```




```
PredictionThreshold = 0.8 #This is the threshold for prediction strength to determine the number of clu

prediction.strength(gower_mat,2,10,M=50,clustermethod=claraCBI, cutoff = PredictionThreshold, classifica
```

```
## Prediction strength
## Clustering method:  clara/pam
## Maximum number of clusters:  10
## Resampled data sets:  50
## Mean pred.str. for numbers of clusters:  1 0.8272176 0.736873 1 0.6767925 0.5633824 0.4814099 0.43295
## Cutoff value:  0.8
## Largest number of clusters better than cutoff:  4
```

Assignment Question

Q2. Based on these results, what are the optimal number of clusters for PAM and PAM-MS?

- **Optimal Clusters**: 4
- **Sihoulette Method**: The optimal cluster count corresponds to the maxiumum silhouette width (n=0.7)
- **Prediction Strength**: The optimal cluster count is the first count above the defined threshold (n=0.8)
- Both the sihoulette method and prediction strength agree that 4 is the optimal cluster number

```
## ASSIGN THE NUMBER OF CLUSTERS
numclust = 4 #assign 'numclust' to the appropriate number of clusters as determined by prediction stren

numberOfClusters = 2:10 #Determine the range of (k) number of clusters to test for KAMILA
```

Run the 3 different Cluster Algorithms (PAM, Modha-Spangler PAM, KAMILA)

```
## Run the algorithms
pam_fit = pam(gower_dist, k = numclust, diss = TRUE) #Classic PAM

ms_fit = gmsClust(conData = conVars, catData = catVars, nclust = numclust,
clustFun = pammix, conDist = L1Dist, catDist = matchingDist) #Modha-Spangler PAM
```

```
## Warning in gmsClust(conData = conVars, catData = catVars, nclust = numclust, : At least one entry of
##     is nclust >= the number of categorical variable
##     level combinations?
```

```
kam_fit = kamila(conVars, catVars, numClust = numberOfClusters,numInit = 50, maxIter = 50, calcNumClust

#Add cluster designation to the data
Data$PAMcluster = pam_fit$clustering
Data$MScluster = ms_fit$results$cluster
Data$KAMcluster = kam_fit$finalMemb

# Calculate the RAND INDEX between clustering methods
#RAND_PAM_MS = cluster_similarity(Data$PAMcluster,Data$MScluster, similarity = "rand")
library(mclust)
```

```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
RAND_PAM_MS = adjustedRandIndex(Data$PAMcluster,Data$MScluster)
RAND_PAM_KAM = adjustedRandIndex(Data$PAMcluster,Data$KAMcluster)
RAND_KAM_MS = adjustedRandIndex(Data$MScluster,Data$KAMcluster)
as.data.frame(cbind(RAND_PAM_MS,RAND_PAM_KAM,RAND_KAM_MS))
```

```
##   RAND_PAM_MS RAND_PAM_KAM RAND_KAM_MS
## 1           1    0.9636273   0.9636273
```

Summarize each Cluster into Tables

```
library(tableone)
library(pander)
variable_list <- c("Fixation", "SexDesc", "Age", "BMI", "OneYearMTPM", "OneToTwoMTPM")
factor_variables <- c("SexDesc","Fixation")


tab0 <- CreateTableOne(vars = variable_list,, data = Data, factorVars = factor_variables, test = FALSE)
print(tab0,  formatOptions = list(big.mark = ","))
```

```
                     Overall
```

n 247

Fixation = Uncemented (%) 121 (49.0) SexDesc = Male (%) 97 (39.3) Age (mean (SD)) 64.30 (8.03) BMI (mean (SD)) 32.26 (5.89) OneYearMTPM (mean (SD)) 0.73 (0.79) OneToTwoMTPM (mean (SD)) 0.07 (0.26)

```
tab1 <- CreateTableOne(vars = variable_list, strata = "PAMcluster" , data = Data, factorVars = factor_va
print(tab1,  formatOptions = list(big.mark = ","))
```

```
                       Stratified by PAMcluster
                        1            2             3
```

n 84 42 66
Fixation = Uncemented (%) 0 (0.0) 0 (0.0) 66 (100.0) SexDesc = Male (%) 0 (0.0) 42 (100.0) 0 (0.0) Age (mean (SD)) 62.18 (8.64) 66.24 (8.35) 65.11 (7.07)
BMI (mean (SD)) 33.76 (6.75) 31.05 (5.34) 32.32 (5.78)
OneYearMTPM (mean (SD)) 0.40 (0.30) 0.50 (0.34) 1.16 (1.12)
OneToTwoMTPM (mean (SD)) 0.07 (0.24) 0.05 (0.18) 0.07 (0.30)
Stratified by PAMcluster 4
n 55
Fixation = Uncemented (%) 55 (100.0) SexDesc = Male (%) 55 (100.0) Age (mean (SD)) 65.09 (7.38)
BMI (mean (SD)) 30.83 (4.40)
OneYearMTPM (mean (SD)) 0.90 (0.79)
OneToTwoMTPM (mean (SD)) 0.09 (0.28)

```
tab2 <- CreateTableOne(vars = variable_list, strata = "MScluster" , data = Data, factorVars = factor_var
print(tab2,  formatOptions = list(big.mark = ","))
```

```
                       Stratified by MScluster
                        1            2             3
```

n 84 42 66
Fixation = Uncemented (%) 0 (0.0) 0 (0.0) 66 (100.0) SexDesc = Male (%) 0 (0.0) 42 (100.0) 0 (0.0) Age (mean (SD)) 62.18 (8.64) 66.24 (8.35) 65.11 (7.07)
BMI (mean (SD)) 33.76 (6.75) 31.05 (5.34) 32.32 (5.78)
OneYearMTPM (mean (SD)) 0.40 (0.30) 0.50 (0.34) 1.16 (1.12)
OneToTwoMTPM (mean (SD)) 0.07 (0.24) 0.05 (0.18) 0.07 (0.30)
Stratified by MScluster 4
n 55
Fixation = Uncemented (%) 55 (100.0) SexDesc = Male (%) 55 (100.0) Age (mean (SD)) 65.09 (7.38)
BMI (mean (SD)) 30.83 (4.40)
OneYearMTPM (mean (SD)) 0.90 (0.79)
OneToTwoMTPM (mean (SD)) 0.09 (0.28)

```
tab3 <- CreateTableOne(vars = variable_list, strata = "KAMcluster" , data = Data, factorVars = factor_va
print(tab3,  formatOptions = list(big.mark = ","))
```

```
                       Stratified by KAMcluster
                        1             2            3
```

n 7 84 42
Fixation = Uncemented (%) 7 (100.0) 0 (0.0) 0 (0.0) SexDesc = Male (%) 2 (28.6) 0 (0.0) 42 (100.0) Age (mean (SD)) 68.86 (7.06) 62.18 (8.64) 66.24 (8.35)
BMI (mean (SD)) 30.50 (5.52) 33.76 (6.75) 31.05 (5.34)
OneYearMTPM (mean (SD)) 4.02 (0.61) 0.40 (0.30) 0.50 (0.34)
OneToTwoMTPM (mean (SD)) 0.50 (0.60) 0.07 (0.24) 0.05 (0.18)
Stratified by KAMcluster 4 5
n 61 53
Fixation = Uncemented (%) 61 (100.0) 53 (100.0) SexDesc = Male (%) 0 (0.0) 53 (100.0) Age (mean (SD)) 64.74 (6.92) 65.02 (7.49)

BMI (mean (SD)) 32.60 (5.77) 30.68 (4.36)
OneYearMTPM (mean (SD)) 0.91 (0.71) 0.80 (0.59)
OneToTwoMTPM (mean (SD)) 0.03 (0.25) 0.07 (0.23)

Assignment Questions

Q3a. Discuss the clusters for PAM and MS-PAM including the characteristics of the clusters and how the clusters compare between the two methods. [5 marks]

- **Rand Index**: "RAND_PAM_MS" has a value of 1, which indicates an exact match between the produced clusters. This is confirmed by analyzing the tabular outputs as both clusters have the same summary values.
- **Cluster Discussion**: The main differentiators between the clusters appears to be fixation type and sex. The other metrics (Age, BMI, OneYearMTPM, OneToTwoMTPM) are relatively close between clusters, and with high standard deviations. Given this, the clustering can be conceptualized as a decision tree like so:

```
if (SexDesc == female):
    if (Fixation == cemented):
        cluster 1
    else if (Fixation == uncemented):
        cluster 2
else if (SexDesc == male):
    if (Fixation == cemented):
        cluster 1
    else if (Fixation == uncemented):
        cluster 2
```

Q3b. Discuss the clusters from KAMILA, including a discussion of what Cluster 1 represents and how the remaining clusters compare to those identified in PAM and MS-PAM.[3 marks]

- **KAMILA**: The first cluster is the smallest (n=7) and its distinguishing feature is that it has notably high mean OneYearMTPM (n=4.02) compared to the other clusters which have MTPM ranging between [0.40, 0.91]. The fixture type is purely uncemented, but the sex is heterogeneous unlike any of the other clusters. I believe it would be reasonable for a 6th cluster to be created which contains cemented fixtures with high OneYearMTPM due to the relatively high standard deviations in the cemented fixture clusters (#4,5).
- **Comparison**: (1) There are 5 clusters instead of 4, (2) the average values of the numeric data is mostly the same, aside from the OneYearMTPM of cluster 1, (3) PAM/MS-PAM methods have fully homogeneous clusters wrt categorical variables, whereas KAMILA cluster #1 is an exception as it is heterogeneous for sex.

Q3c. What do you notice about the categorical variables for these results? [1 mark]

- **Aggregation**: The categorical variables are reported as totals and proportions instead of means and standard deviations. The clusters are most easily distinguished by their categorical variables due to their homogeneity (i.e. proportions are mostly either 0% or 100%)

Let's look at disaggregating the data and re-running the cluster analyses on subgroups (4 categories: Female Cemented (FC), Female Uncemented (FU), Male Cemented (MC), Male Uncemented (MU).

```r
# Separate into 4 different groups based on
Data.Female.Cemented = Data[ which(Data$SexDesc == "Female" & Data$Fixation =="Cemented"), ]
Data.Female.Uncemented = Data[ which(Data$SexDesc == "Female" & Data$Fixation =="Uncemented"), ]
Data.Male.Cemented = Data[ which(Data$SexDesc == "Male" & Data$Fixation =="Cemented"), ]
Data.Male.Uncemented = Data[ which(Data$SexDesc == "Male" & Data$Fixation =="Uncemented"), ]

#Create dataframe for each group containing only the variables to be clustered
var = c("Subject","Age","BMI","OneYearMTPM","OneToTwoMTPM")

Data.ClusterFC = na.omit(Data.Female.Cemented[, var])
Data.ClusterFU = na.omit(Data.Female.Uncemented[, var])
Data.ClusterMC = na.omit(Data.Male.Cemented[, var])
Data.ClusterMU = na.omit(Data.Male.Uncemented[, var])

round(colMeans(Data.ClusterFC), 2)
```

```
##      Subject          Age          BMI  OneYearMTPM OneToTwoMTPM
##      2840.93        62.18        33.76         0.40         0.07
```

```r
round(colMeans(Data.ClusterFU), 2)
```

```
##      Subject          Age          BMI  OneYearMTPM OneToTwoMTPM
##      3929.32        65.11        32.32         1.16         0.07
```

```r
round(colMeans(Data.ClusterMC), 2)
```

```
##      Subject          Age          BMI  OneYearMTPM OneToTwoMTPM
##      2768.05        66.24        31.05         0.50         0.05
```

```r
round(colMeans(Data.ClusterMU), 2)
```

```
##      Subject          Age          BMI  OneYearMTPM OneToTwoMTPM
##      3603.00        65.09        30.83         0.90         0.09
```

Assignment Question

Q4. What do you notice about the disaggregated groups and the clusters you defined above? [2 marks]

- **Comparison**: The disaggregated groups appear to be identical to the PAM/PAM-MS clusters (see below) according to the mean values of the numeric variables. This is reasonable because the clusters are cleanly split based on sex and fixture type, as discussed in Q3a

Age	62.18 (8.6)	66.24 (8.35)	65.11 (7.07)	65.09 (7.38)
BMI	33.76 (6.7)	31.05 (5.34)	32.32 (5.78)	30.83 (4.40)
OneYearMTPM	0.40 (0.3)	0.50 (0.34)	1.16 (1.12)	0.90 (0.79)
OneToTwoMTPM	0.07 (0.2)	0.05 (0.18)	0.07 (0.30)	0.09 (0.28)

Outliers

Since cluster analysis is sensitive to outliers, we are going to proactively identify them and look at them separately, while performing a cluster analysis on the remaining data to see if we can identify any more subtle patterns in the data.

```r
#Identify outliers function
IsOutlier <- function(x) {
  outlier = NA
  for (i in 1:length(x)) {
    outlier[i] = ( (x[i] < quantile(x, 0.25, names=FALSE) - 3 * IQR(x)) | (x[i] > quantile(x, 0.75, nam
    #Using Tukey's method (1977): outliers are values below "fence" (Quartile 1)-(1.5*IQR) and above "f
    #outlier of 1.5 is "outlier"
    #outlier of 3 is "far out"
  }
  outlier = as.numeric(outlier)
  outlier
}

###

# Identify Outliers in each group
for (i in colnames(Data.ClusterFC[ ,-1])) {
    Data.ClusterFC[[paste0("Outlier_",i)]] = IsOutlier(Data.ClusterFC[ , i])
  }
  #Create column identifying outliers
  Data.ClusterFC$Exclude = rowSums(Data.ClusterFC[ , grepl("Outlier", names(Data.ClusterFC))])
  Data.Female.Cemented$Exclude = rowSums(Data.ClusterFC[ , grepl("Outlier", names(Data.ClusterFC))])
  Data.ClusterFC = Data.ClusterFC[ ,-grep("Outlier", names(Data.ClusterFC))]

for (i in colnames(Data.ClusterFU[ ,-1])) {
  Data.ClusterFU[[paste0("Outlier_",i)]] = IsOutlier(Data.ClusterFU[ , i])
}
  Data.ClusterFU$Exclude = rowSums(Data.ClusterFU[ , grepl("Outlier", names(Data.ClusterFU))])
  Data.Female.Uncemented$Exclude = rowSums(Data.ClusterFU[ , grepl("Outlier", names(Data.ClusterFU))])
  Data.ClusterFU = Data.ClusterFU[ ,-grep("Outlier", names(Data.ClusterFU))]


for (i in colnames(Data.ClusterMC[ ,-1])) {
  Data.ClusterMC[[paste0("Outlier_",i)]] = IsOutlier(Data.ClusterMC[ , i])
}
  Data.ClusterMC$Exclude = rowSums(Data.ClusterMC[ , grepl("Outlier", names(Data.ClusterMC))])
  Data.Male.Cemented$Exclude = rowSums(Data.ClusterMC[ , grepl("Outlier", names(Data.ClusterMC))])
  Data.ClusterMC = Data.ClusterMC[ ,-grep("Outlier", names(Data.ClusterMC))]


for (i in colnames(Data.ClusterMU[ ,-1])) {
  Data.ClusterMU[[paste0("Outlier_",i)]] = IsOutlier(Data.ClusterMU[ , i])
}
  Data.ClusterMU$Exclude = rowSums(Data.ClusterMU[ , grepl("Outlier", names(Data.ClusterMU))])
  Data.Male.Uncemented$Exclude = rowSums(Data.ClusterMU[ , grepl("Outlier", names(Data.ClusterMU))])
  Data.ClusterMU = Data.ClusterMU[ ,-grep("Outlier", names(Data.ClusterMU))]

###
```

```r
# Create dataframes of outliers
FC_Outliers = Data.Female.Cemented[ which(Data.Female.Cemented$Exclude != 0), ]
Data.Female.Cemented = Data.Female.Cemented[ -which(Data.Female.Cemented$Exclude != 0), ]
FU_Outliers = Data.Female.Uncemented[ which(Data.Female.Uncemented$Exclude != 0), ]
Data.Female.Uncemented = Data.Female.Uncemented[ -which(Data.Female.Uncemented$Exclude != 0), ]
```

```
MC_Outliers = Data.Male.Cemented[ which(Data.Male.Cemented$Exclude != 0), ]
Data.Male.Cemented = Data.Male.Cemented[ -which(Data.Male.Cemented$Exclude != 0), ]
MU_Outliers = Data.Male.Uncemented[ which(Data.Male.Uncemented$Exclude != 0), ]
Data.Male.Uncemented = Data.Male.Uncemented[ -which(Data.Male.Uncemented$Exclude != 0), ]


#Remove outliers from data to be clustered
Data.ClusterMU = Data.ClusterMU[ which(Data.ClusterMU$Exclude == 0) ,-grep("Exclude", names(Data.Cluste
Data.ClusterMC = Data.ClusterMC[ which(Data.ClusterMC$Exclude == 0) ,-grep("Exclude", names(Data.Cluste
Data.ClusterFU = Data.ClusterFU[ which(Data.ClusterFU$Exclude == 0) ,-grep("Exclude", names(Data.Cluste
Data.ClusterFC = Data.ClusterFC[ which(Data.ClusterFC$Exclude == 0) ,-grep("Exclude", names(Data.Cluste
```

In order to choose what clustering approach and how many clusters to use, we are going to compare methods. We are now only including numerical values, so we'll look at the classical clustering approaches and will also compare distance methods

Test all possible combinations of methods for best silhouette score

```
[1] CF
[1] "euclidean"
                         2       3       4       5       6
kmeans Silhouette   0.3389  0.3088  0.2877  0.2772  0.2666
pam    Silhouette   0.3161  0.2767  0.2385  0.2346  0.2600
agnes  Silhouette   0.3681  0.2869  0.2303  0.2246  0.1802
diana  Silhouette   0.3533  0.3373  0.2609  0.2446  0.2411

[1] "manhattan"
                         2       3       4       5       6
kmeans Silhouette   0.3454  0.3040  0.2523  0.2561  0.2758
pam    Silhouette   0.2864  0.2982  0.2493  0.2332  0.2292
agnes  Silhouette   0.3454  0.2918  0.2226  0.2309  0.2319
diana  Silhouette   0.3454  0.3228  0.2979  0.2716  0.2329


[1] UF
[1] "euclidean"
                         2       3       4       5       6
kmeans Silhouette   0.4535  0.3829  0.3863  0.3742  0.2517
pam    Silhouette   0.4029  0.2883  0.2148  0.2329  0.2191
agnes  Silhouette   0.5177  0.3808  0.3756  0.3779  0.3336
diana  Silhouette   0.4741  0.4295  0.3940  0.3062  0.3049

[1] "manhattan"
                         2       3       4       5       6
kmeans Silhouette   0.4169  0.3669  0.3331  0.3091  0.2988
pam    Silhouette   0.3971  0.3533  0.2076  0.2146  0.1889
agnes  Silhouette   0.4514  0.3902  0.3606  0.3470  0.3392
diana  Silhouette   0.5109  0.4155  0.3431  0.3325  0.3202


[1] CM
[1] "euclidean"
                         2       3       4       5       6
```

```
kmeans Silhouette     0.3264  0.2943  0.2594  0.3115  0.2717
pam    Silhouette     0.3094  0.2558  0.2990  0.2524  0.2222
agnes  Silhouette     0.4948  0.2105  0.2417  0.2069  0.2240
diana  Silhouette     0.3451  0.2692  0.2869  0.3115  0.2903

[1] "manhattan"
                           2       3       4       5       6
kmeans Silhouette     0.3211  0.2724  0.2448  0.3222  0.3122
pam    Silhouette     0.2930  0.2420  0.2665  0.2538  0.2652
agnes  Silhouette     0.3627  0.2519  0.2685  0.3195  0.3104
diana  Silhouette     0.3117  0.2764  0.2465  0.2992  0.3002

[1] UM
[1] "euclidean"
                           2       3       4       5       6
kmeans Silhouette     0.3422  0.3114  0.3072  0.2992  0.2671
pam    Silhouette     0.2908  0.2985  0.2436  0.2463  0.2661
agnes  Silhouette     0.4753  0.4013  0.3041  0.2901  0.2928
diana  Silhouette     0.4753  0.4013  0.3171  0.3026  0.3021

[1] "manhattan"
                           2       3       4       5       6
kmeans Silhouette     0.3755  0.3308  0.2770  0.2364  0.2345
pam    Silhouette     0.3286  0.1988  0.2059  0.2089  0.2457
agnes  Silhouette     0.3595  0.2905  0.2790  0.2635  0.2362
diana  Silhouette     0.4156  0.3553  0.2476  0.2486  0.2416
```

Assignment Questions

5a. Using a Silhouette value as your primary criteria (closest to 1), for each of the 4 categories (UF, CF, UM, CM), which distance metric, method and how many clusters (k) would you choose to proceed with? [4 marks]

Optimal strategy by silhouette score: Go through each fixture-sex group, then search within the 2 distance matrices to find the largest silhouette value.

- **UF**: Euclidean, Agnes, k=2 (silhouette=0.3681)
- **CF**: Euclidean, Agnes, k=2 (silhouette=0.5177)
- **UM**: Euclidean, Agnes or Diana, k=2 (silhouette=0.4753)
- **CM**: Euclidean, Agnes, k=2 (silhouette=0.4948)

5b. Using the same approach for clustering for all 4 categories will make it easier to compare between categories. Which method (& k#) would you select to analyze all categories and why? [2 marks]

- **Optimal strategy**: I would choose Euclidean distance with Agnes clustering and K=2 as this combination consistently performed the best across all fixture-sex groupings.

Compute optimal distance matrices

```
library(NbClust)
distFC = daisy(Data.ClusterFC[ , -1], metric = "euclidean", stand = TRUE)
distmatFC = as.matrix(distFC)

distFU = daisy(Data.ClusterFU[ , -1], metric = "euclidean", stand = TRUE)
distmatFU = as.matrix(distFU)

distMC = daisy(Data.ClusterMC[ , -1], metric = "euclidean", stand = TRUE)
distmatMC = as.matrix(distMC)

distMU = daisy(Data.ClusterMU[ , -1], metric = "euclidean", stand = TRUE)
distmatMU = as.matrix(distMU)
```

Perform cluster analysis

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.19.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

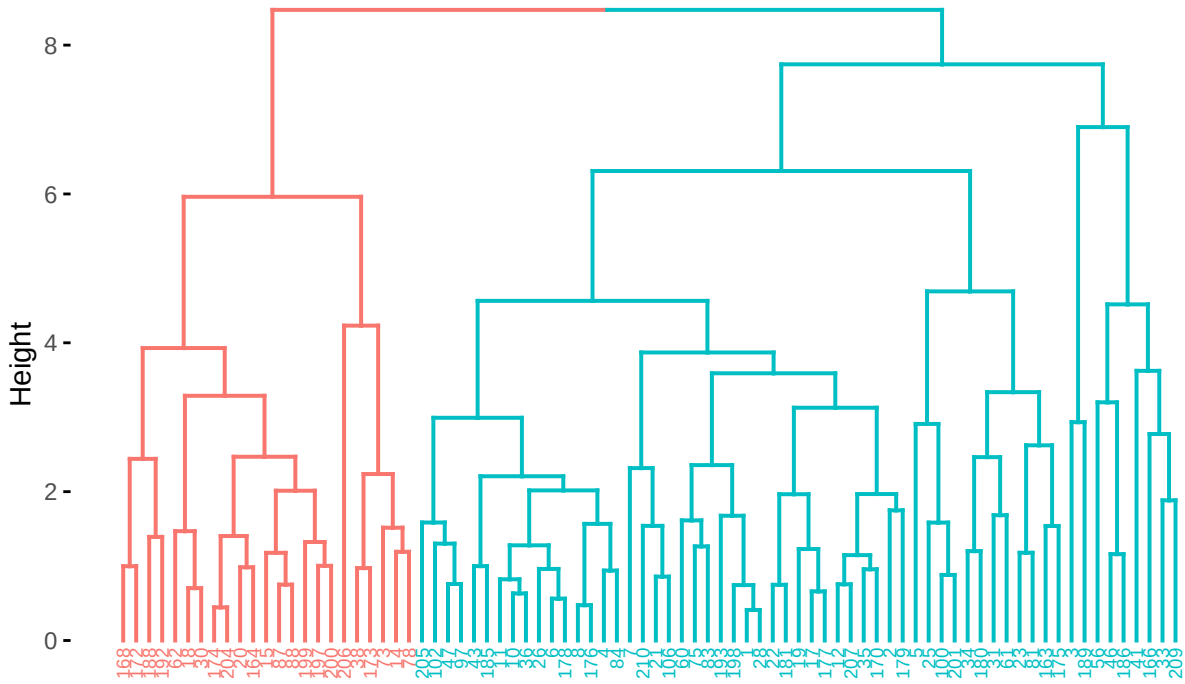
```
##
## Attaching package: 'dendextend'
```

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

```
library(mclust)
FC_fit = hclust(distFC, method = "complete")
Data.Female.Cemented$HCluster = cutree(FC_fit, k =2)
Data.ClusterFC$HCluster = cutree(FC_fit, k =2)
fviz_dend(FC_fit, k =2, color_labels_by_k = TRUE, cex = 0.5, main = "Female Cemented Dendrogram")
```

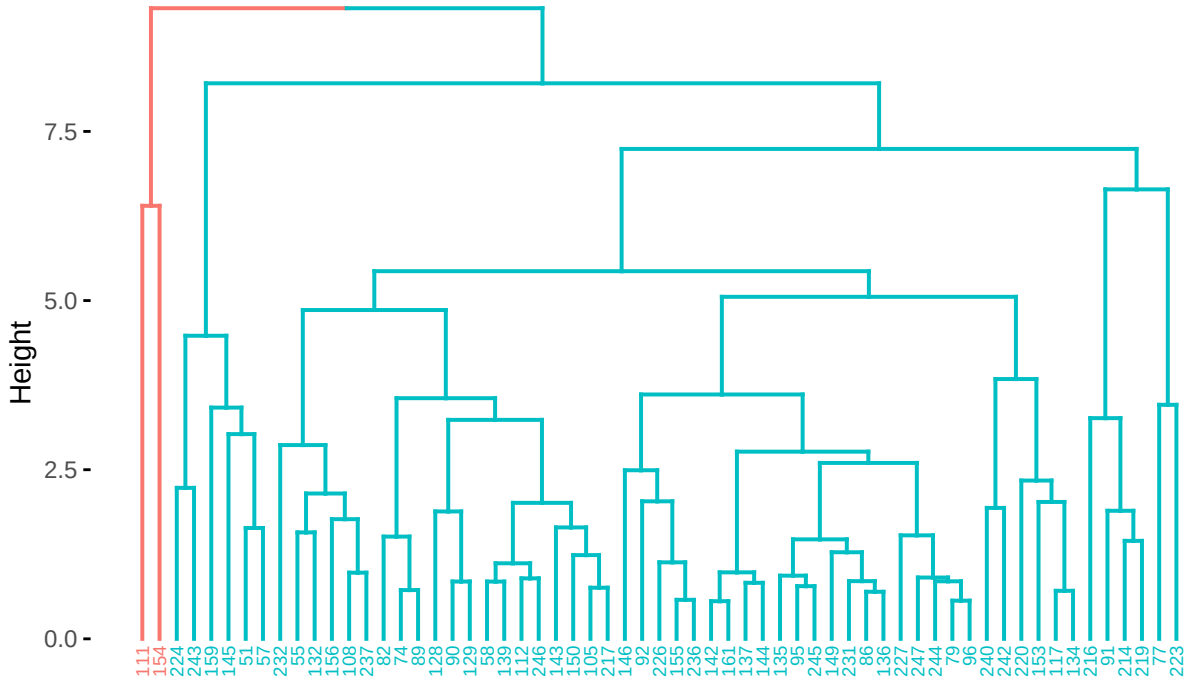
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Female Cemented Dendrogram

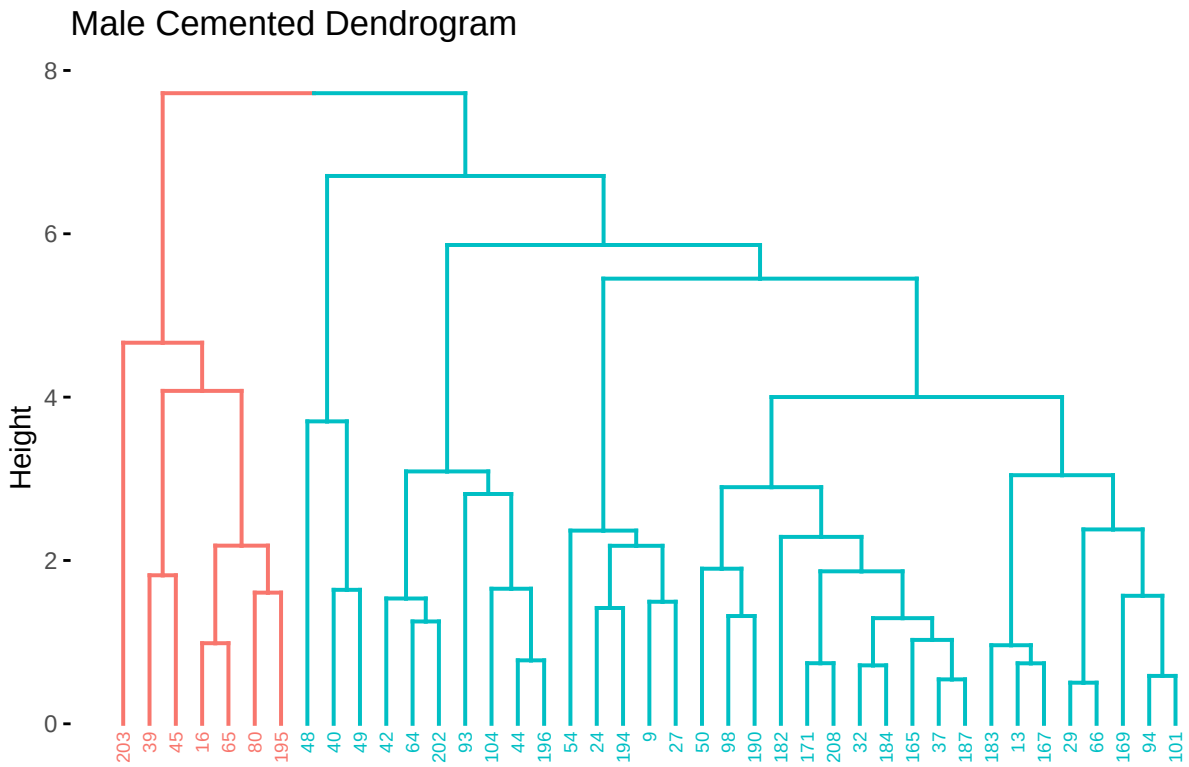



```
FU_fit = hclust(distFU, method = "complete")
Data.Female.Uncemented$HCluster = cutree(FU_fit, k =2)
Data.ClusterFU$HCluster = cutree(FU_fit, k =2)
fviz_dend(FU_fit, k =2, color_labels_by_k = TRUE, cex = 0.5, main = "Female Uncemented Dendrogram")
```

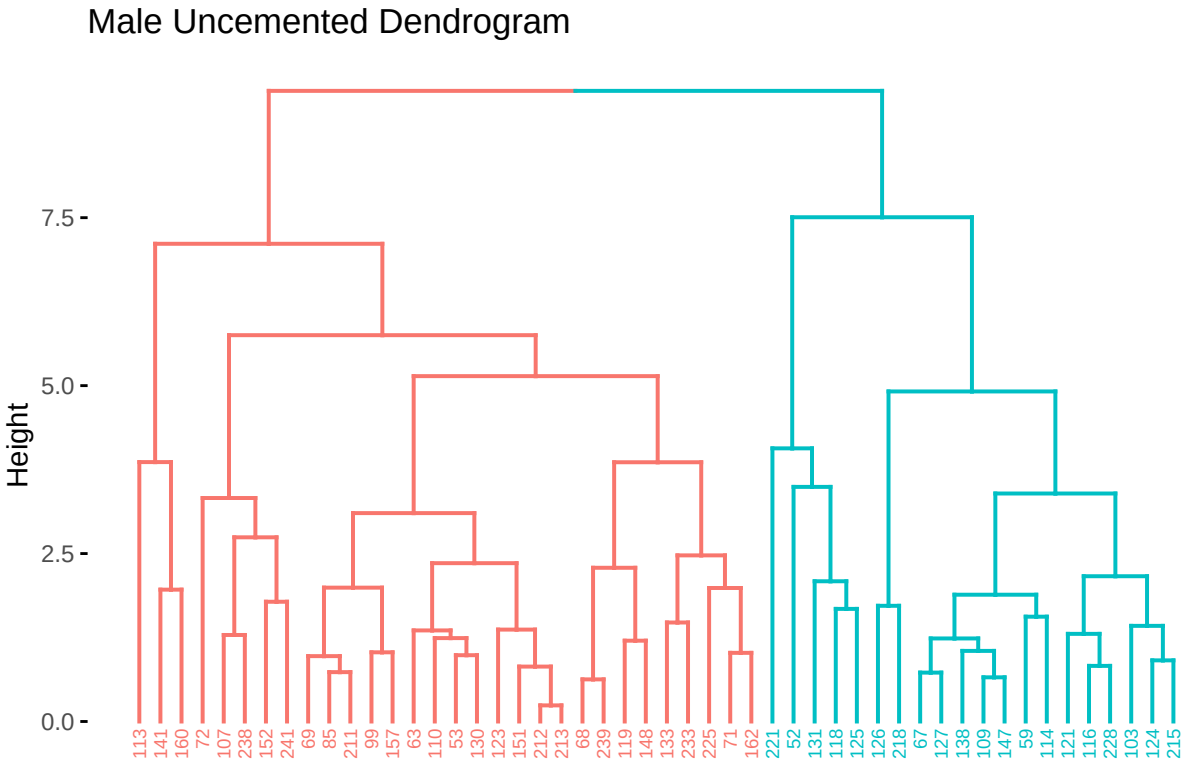
Female Uncemented Dendrogram



```
MC_fit = hclust(distMC, method = "complete")
Data.Male.Cemented$HCluster = cutree(MC_fit, k =2)
Data.ClusterMC$HCluster = cutree(MC_fit, k =2)
fviz_dend(MC_fit, k =2, color_labels_by_k = TRUE, cex = 0.5, main = "Male Cemented Dendrogram")
```




```
MU_fit = hclust(distMU, method = "complete")
Data.Male.Uncemented$HCluster = cutree(MU_fit, k =2)
Data.ClusterMU$HCluster = cutree(MU_fit, k =2)
fviz_dend(MU_fit, k =2, color_labels_by_k = TRUE, cex = 0.5, main = "Male Uncemented Dendrogram")
```



Subset test of stability

```
print("Female Cemented")
```

```
## [1] "Female Cemented"
```

```
cboot_FC
```

```
## * Cluster stability assessment *
## Cluster method:  hclust/cutree
## Full clustering results are given as parameter result
## of the clusterboot object, which also provides further statistics
## of the resampling results.
## Number of resampling runs:  200
##
## Number of clusters found in data:  2
##
##  Clusterwise Jaccard subsetting mean:
## [1] 0.9351137 0.6634466
## dissolved:
## [1]  0 59
## recovered:
## [1] 199  74
```

```
print("Female Uncemented")
```

```
## [1] "Female Uncemented"
```

```
cboot_FU
```

```
## * Cluster stability assessment *
```

```
## Cluster method:  hclust/cutree
## Full clustering results are given as parameter result
## of the clusterboot object, which also provides further statistics
## of the resampling results.
## Number of resampling runs:  200
##
## Number of clusters found in data:  2
##
##  Clusterwise Jaccard subsetting mean:
## [1] 0.9448232 0.6708648
## dissolved:
## [1]  0 82
## recovered:
## [1] 198  96
```

```
print("Male Cemented")
```

```
## [1] "Male Cemented"
```

```
cboot_MC
```

```
## * Cluster stability assessment *
## Cluster method:  hclust/cutree
## Full clustering results are given as parameter result
## of the clusterboot object, which also provides further statistics
## of the resampling results.
## Number of resampling runs:  200
##
## Number of clusters found in data:  2
##
##  Clusterwise Jaccard subsetting mean:
## [1] 0.9229645 0.5480595
## dissolved:
## [1]   0 121
## recovered:
## [1] 183  49
```

```
print("Male Uncemented")
```

```
## [1] "Male Uncemented"
```

```
cboot_MU
```

```
## * Cluster stability assessment *
## Cluster method:  hclust/cutree
## Full clustering results are given as parameter result
## of the clusterboot object, which also provides further statistics
## of the resampling results.
## Number of resampling runs:  200
##
## Number of clusters found in data:  2
##
##  Clusterwise Jaccard subsetting mean:
## [1] 0.9655039 0.6763777
## dissolved:
## [1]  0 88
## recovered:
```

```
## [1] 197  85
```

Assignment Question

6. Using the following guidelines, classify the stability of the 2 clusters in each of the 4 categories [4 marks]

There are 3 metrics from the stability assessment, which are reported in pairs that correspond to each cluster for every class. For classification I'll only consider the Jaccard similarity coefficient to keep things simple

- **Jaccard**: Measures similarity between original and final clusters from 0 (no matches; unstable) to 1 (identical; stable)
- **Dissolved**: Counts the number of reassigned subjects (higher = more unstable)
- **Recovered**: Counts the number of retained subjects (higher = more stable)

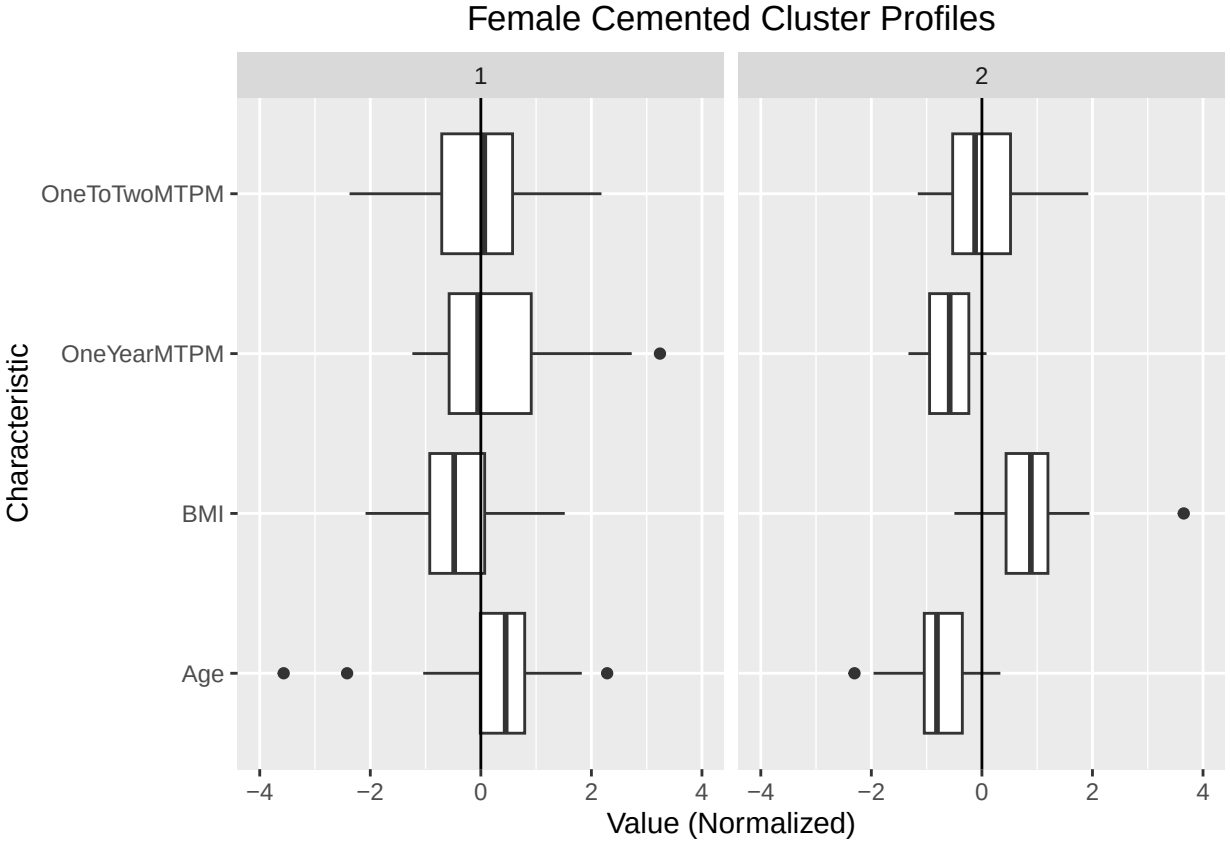
- **CF**: The first cluster is stable ($n_1 \approx 0.94$) but the second is unstable ($n_2 \approx 0.66$)
- **UF**: The first cluster is stable ($n_1 \approx 0.94$) but the second is unstable ($n_2 \approx 0.67$)
- **CM**: The first cluster is stable ($n_1 \approx 0.92$) but the second is unstable ($n_2 \approx 0.55$)
- **UM**: The first cluster is stable ($n_1 \approx 0.97$) but the second is unstable ($n_2 \approx 0.68$)

Clusterwise Jaccard subsetting mean: < 0.6 unstable 0.6 - 0.75 low stability 0.75 - 0.85 moderate stablity $>$ 0.85 stable

Summary of cluster results

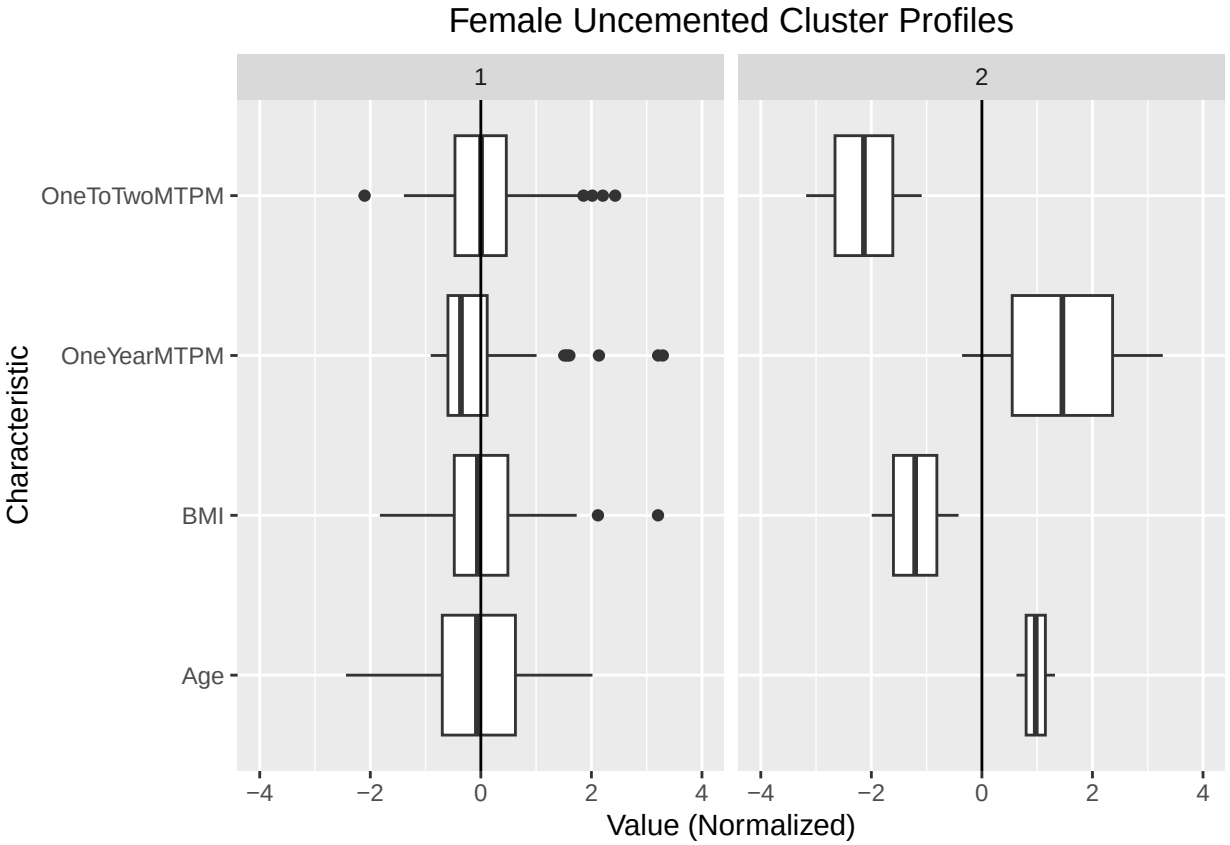
```
##
## Attaching package: 'dplyr'

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

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

## Warning in summary_table.grouped_df(dplyr::group_by(Data.Female.Cemented, : grouped_df detected. Set
##   c('HCluster')
```

	Data.Female.Cemented (N = 82)	1 (N = 59)	2 (N = 23)
Age			
	Age (Mean ± SD)	**62.09 ± 8.72**	**64.86 ± 8.13**	**54.96 ± 5.60**
BMI			
	BMI (Mean ± SD)	**33.78 ± 6.80**	**31.36 ± 5.52**	**39.97 ± 5.83**
OneYearMTPM			
	1 Year MTPM (Mean ± SD)	**0.38 ± 0.23**	**0.43 ± 0.24**	**0.24 ± 0.10**
OneToTwoMTPM			
	Change in MTPM (1yr to 2yr) (Mean ± SD)	**0.05 ± 0.15**	**0.05 ± 0.16**	**0.05 ± 0.13**
Smoke			
	Smoker	**8 (9.76%)**	**6 (10.17%)**	**2 (8.70%)**
TibiaArea			
	Tibia Area (Mean ± SD)	**24.64 ± 2.48**	**24.51 ± 2.51**	**24.99 ± 2.43**



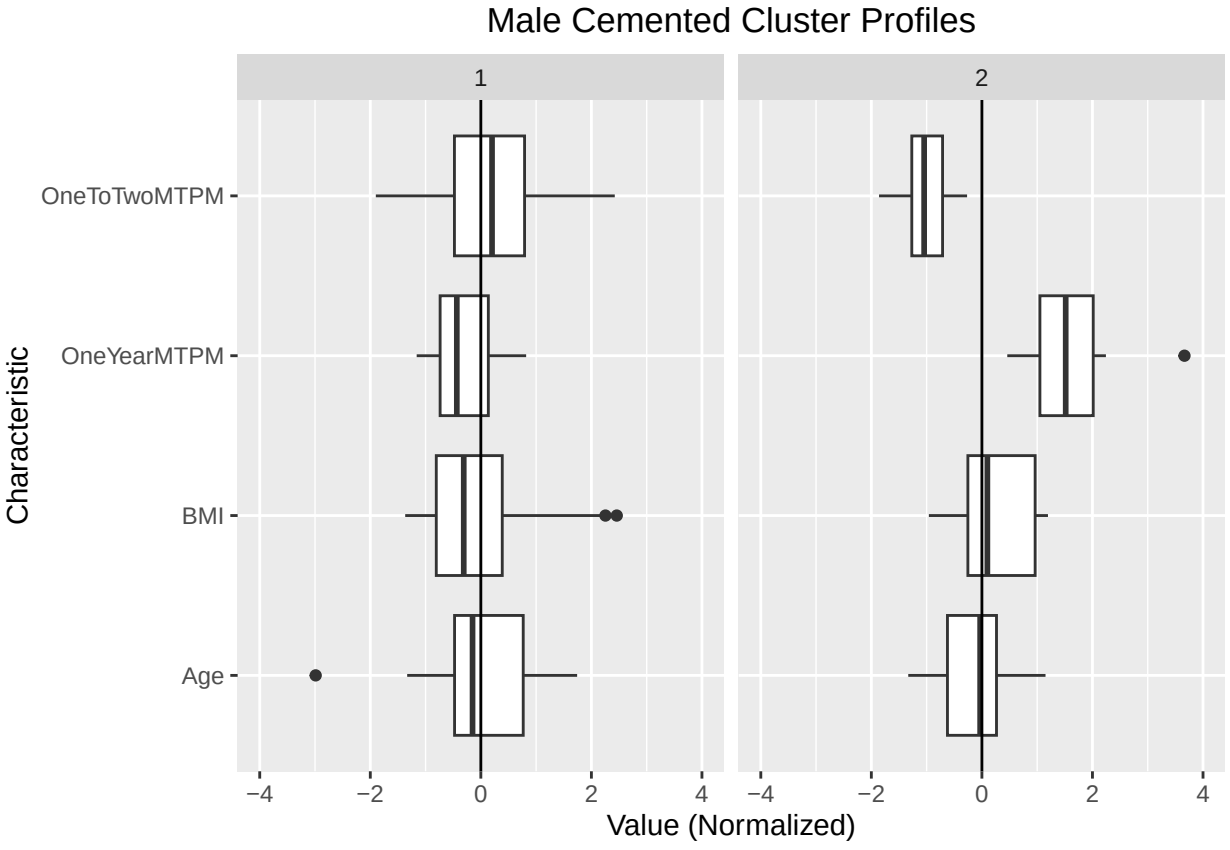

```
## Warning in summary_table.grouped_df(dplyr::group_by(Data.Female.Uncemented, : grouped_df detected. Se
##   c('HCluster')
```

	Data.Female.Uncemented (N = 61)	1 (N = 59)	2 (N = 2)
Age | | | | | Age (Mean ± SD) |65.51 ± 7.18 |65.27 ± 7.17 |72.50 ± 3.54 | |BMI | | | | | BMI (Mean ± SD) |32.32 ± 5.98 |32.57 ± 5.87 |25.10 ± 6.65 | |OneYearMTPM | | | | | 1 Year MTPM (Mean ± SD) |1.01 ± 0.96 |0.96 ± 0.88 |2.41 ± 2.46 | |OneToTwoMTPM | | | | | Change in MTPM (1yr to 2yr) (Mean ± SD) |0.05 ± 0.19 |0.06 ± 0.17 |-0.35 ± 0.28 | |Smoke | | | | | Smoker |5 (8.20%) |5 (8.47%) |0 (0.00%) | |TibiaArea | | | | | Tibia Area (Mean ± SD) |25.10 ± 2.76 |25.12 ± 2.62 |24.60 ± 7.64 |			



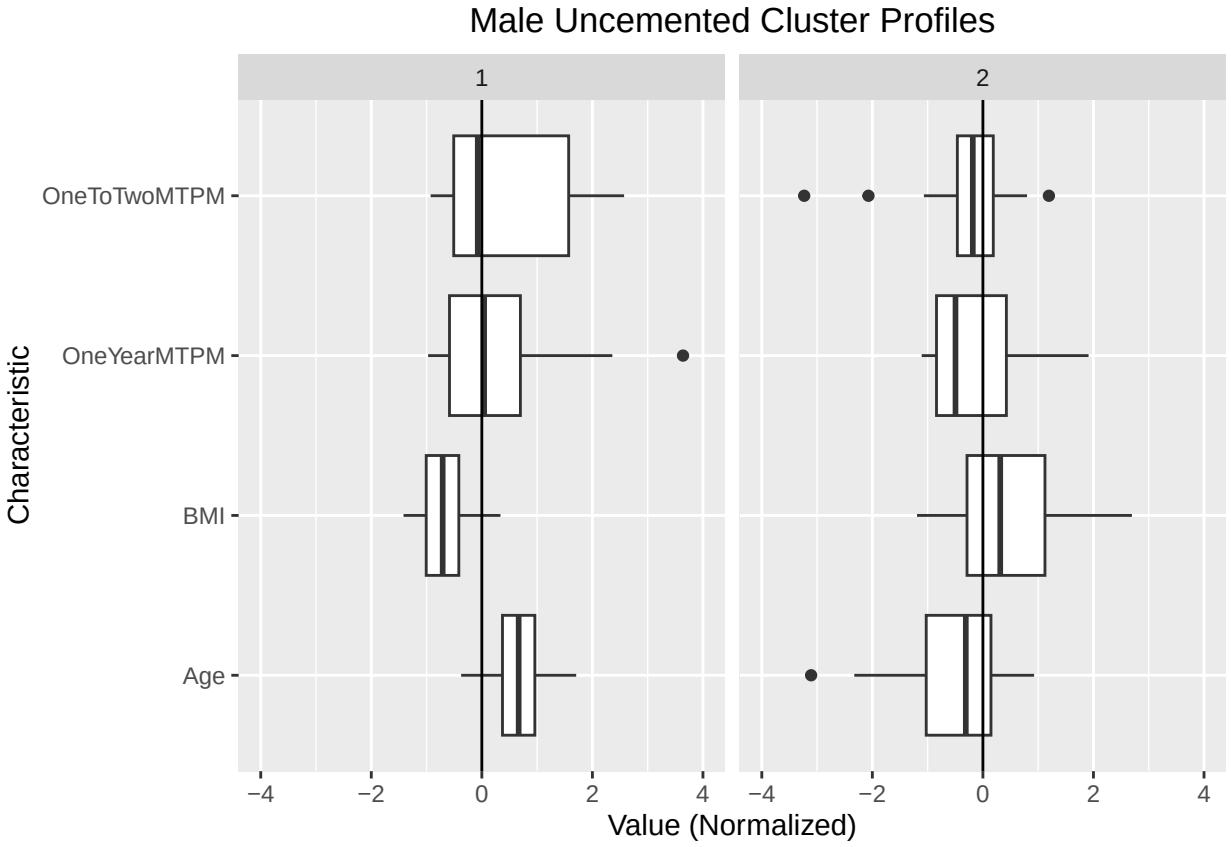

```
## Warning in summary_table.grouped_df(dplyr::group_by(Data.Male.Cemented, : grouped_df detected. Settin
##   c('HCluster')
```

	Data.Male.Cemented (N = 41)	1 (N = 34)	2 (N = 7)
Age | | | | | Age (Mean ± SD) |66.27 ± 8.46 |66.50 ± 8.81 |65.14 ± 6.91 | |BMI | | | | | BMI (Mean ± SD) |30.97 ± 5.38 |30.69 ± 5.58 |32.31 ± 4.41 | |OneYearMTPM | | | | | 1 Year MTPM (Mean ± SD) |0.47 ± 0.29 |0.37 ± 0.16 |0.96 ± 0.31 | |OneToTwoMTPM | | | | | Change in MTPM (1yr to 2yr) (Mean ± SD) |0.03 ± 0.13 |0.05 ± 0.12 |-0.10 ± 0.07 | |Smoke | | | | | Smoker |4 (9.76%) |4 (11.76%) |0 (0.00%) | |TibiaArea | | | | | Tibia Area (Mean ± SD) |30.88 ± 3.13 |31.05 ± 3.24 |30.06 ± 2.53 |			




```
## Warning in summary_table.grouped_df(dplyr::group_by(Data.Male.Uncemented, : grouped_df detected. Set
##   c('HCluster')
```

	Data.Male.Uncemented (N = 50)	1 (N = 20)	2 (N = 30)
Age | | | | | Age (Mean ± SD) |64.88 ± 7.68 |70.10 ± 4.67 |61.40 ± 7.36 | |BMI | | | | | BMI (Mean ± SD) |30.60 ± 4.45 |27.63 ± 2.17 |32.59 ± 4.50 | |OneYearMTPM | | | | | 1 Year MTPM (Mean ± SD) |0.75 ± 0.53 |0.90 ± 0.63 |0.66 ± 0.44 | |OneToTwoMTPM | | | | | Change in MTPM (1yr to 2yr) (Mean ± SD) |0.05 ± 0.16 |0.11 ± 0.18 |0.01 ± 0.13 | |Smoke | | | | | Smoker |7 (14.00%) |1 (5.00%) |6 (20.00%) | |TibiaArea | | | | | Tibia Area (Mean ± SD) |31.10 ± 3.02 |31.50 ± 2.97 |30.84 ± 3.07 |			



Assignment Questions

7a. For each of the 4 categories (UF, CF, UM, CM), provide a description of the clusters. [8 marks]

It feels unusual comparing the clusters because the values are normalized and there's a lot of overlap between the distributions. I'm ideally looking for differences in the dependent variables (OneToTwoMTPM, OneYearMTPM) which can be explained by different explanatory variables (BMI, Age). I could add additional explanatory variables (Smoke, TibiaArea) for more granular analysis but the clusters were already defined and I do not want to impact previous clustering.

- **CF**: Clusters are largely differentiated by age and BMI. Cluster 1 generally has older patients with lower BMIs and subsequently higher mean MTPM metrics with wider distributions which indicates higher variability.
- **UF**: Cluster 1 appears to model the 'average' patient as all boxplots intersect with the $x = 0$ separator. On the other hand, cluster 2 boxplots are noticeably displaced left and right of the $x = 0$ separator. It contains older patients with lower BMI, large OneYearMTPM, and negative OneToTwoMTPM.
- **CM**: There doesn't seem to be significant differences between age and BMI this time as the boxplots largely overlap. The main differentiators seem to be MTPM which indicates either the clustering is suboptimal or that age and BMI are insufficient to explain the differences in MTPM. Cluster 2 has more extreme MTPM values with OneToTwoMTPM being lower and OneYearMTPM being higher whereas Cluster 1 has more moderate values that are close to $x = 0$.
- **UM**: There aren't notable differences between the MTPM aside from the OneToTwoMTPM which has a wide right-skewed distribution for cluster 1 and a sharper distribution for cluster 2. Cluster 1 has older patients with lower BMIs. The clustering for the UM group focused too much on age and BMI instead of MTPM metrics which may signal suboptimal clustering.

7b. For each cluster within each category, provide 2-4 keywords that highlight that cluster and help differentiate it (these might be things like old/young, high/low migration, stable/unstable etc.) [4 marks]

Responses were discussed in the previous section in greater detail. Ideally we should see clustering done based on high/low migration with appreciable differences in the explanatory variables (Age, BMI) and sharper distributions that indicate finer separations. Most of the clustering is done only on old/young which may indicate sub-optimal clustering. Age and BMI were also found to be negatively correlated previously so this may also add bias to the clustering. The negative correlation is evident in the boxplots as age and BMI distributions appear symmetric across the $x = 0$ line

- **CF**: old/young
- **UF**: old/young, high/low migration
- **CM**: high/low migration
- **UM**: old/young, stable/unstable

7c. Discuss the similarities/differences between categories, based on their clusters. [4 marks]

I don't believe it's fair to compare between categories (CF, UF, CM, UM) as their clustering prioritized different parameters. This could be a result of some categories having significantly fewer participants, larger ranges, or more outliers which increases instability when clustering. I feel only the UF clustering was done well as all 4 metrics follow a noticeably different distribution between the 2 clusters. All the other ones either only differentiated on old/young or on high/low migration.

- **CF and UM** follow similar trends with clustering done primarily on age and BMI
- **UF and CM** both have moderate cluster #1 values and more extreme mean values for cluster #2

```r
dataframes <- list(
  Data.ClusterFC = Data.ClusterFC,
  Data.ClusterFU = Data.ClusterFU,
  Data.ClusterMU = Data.ClusterMU,
  Data.ClusterMC = Data.ClusterMC
)

variables <- c("BMI", "Age", "OneYearMTPM", "OneToTwoMTPM")

results <- lapply(names(dataframes), function(df_name) {
  cat("Dataframe:", df_name, "\n")
  df <- dataframes[[df_name]]

  lapply(variables, function(var) {
    cat("Variable:", var, "\n")
    if (var == "Age" || var == "BMI") {
      aov_result <- aov(get(var) ~ HCluster, data = df)
      summary_result <- summary(aov_result)
      cat("F-statistic:", summary_result[[1]]$F[1], "\n")
      cat("p-value:", summary_result[[1]]$`Pr(>F)`[1], "\n")
    } else {
      kw_result <- wilcox.test(get(var) ~ HCluster, data = df)
      cat("W-statistic:", kw_result$statistic, "\n")
      cat("p-value:", kw_result$p.value, "\n")
    }
    cat("\n")
  })
  cat("-------------\n")
})
```

```
## Dataframe: Data.ClusterFC
## Variable: BMI
## F-statistic: 39.0049
## p-value: 1.914243e-08
##
## Variable: Age
## F-statistic: 28.70277
## p-value: 7.91533e-07
##
## Variable: OneYearMTPM
## W-statistic: 1041
## p-value: 0.0001865208
##
## Variable: OneToTwoMTPM
## W-statistic: 710.5
## p-value: 0.7450714
##
## -------------
## Dataframe: Data.ClusterFU
## Variable: BMI
## F-statistic: 3.121318
## p-value: 0.08244663
##
## Variable: Age
```

```
## F-statistic: 1.994544
## p-value: 0.1631196
##
## Variable: OneYearMTPM
## W-statistic: 30.5
## p-value: 0.2567708
##
## Variable: OneToTwoMTPM
## W-statistic: 113
## p-value: 0.03025003
##
## -------------
## Dataframe: Data.ClusterMU
## Variable: BMI
## F-statistic: 20.94044
## p-value: 3.35835e-05
##
## Variable: Age
## F-statistic: 21.96345
## p-value: 2.32386e-05
##
## Variable: OneYearMTPM
## W-statistic: 374
## p-value: 0.1466931
##
## Variable: OneToTwoMTPM
## W-statistic: 350
## p-value: 0.3301635
##
## -------------
## Dataframe: Data.ClusterMC
## Variable: BMI
## F-statistic: 0.5234014
## p-value: 0.4737115
##
## Variable: Age
## F-statistic: 0.1463396
## p-value: 0.7041351
##
## Variable: OneYearMTPM
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
## W-statistic: 2
## p-value: 5.417659e-05
##
## Variable: OneToTwoMTPM
## W-statistic: 210
## p-value: 0.0007615891
##
## -------------
```

Assignment Questions

8. Using statistical methods from your previous assignments, within each of the 4 categories (UF, CF, UM, CM), determine if the clusters are statistically significantly different for any of the variables included in the cluster analysis (age, BMI, etc). This will require 16 tests. Use the actual values, not normalized values. [16 marks]

I wanted to find a clean way to compare all 16 relations so I searched online for the code. I use the Mann-Whitney-Wilcoxon Test on non-parametric distributions and the t-test (MTPM) on normally distributed data (age, BMI). I should have determined the distributions on a case-by-case basis based on the above histograms, but I instead generalized that age and BMI are normally distributed and the MTPM distributions were generally too skewed for the t-test.

- **CF**: BMI, Age, and OneYearMTPM are significantly different between clusters with $p < 0.001$
- **UF**: Only OneToTwoMTPM is significantly different with $p \approx 0.03$
- **CM**: OneYearMTPM and OneToTwoMTPM are significantly different with $p < 0.001$
- **UM**: Age and BMI are significantly different with $p < 0.001$

Final results are mostly consistent with the empirical comparisons done in question 7b

Characterize Outliers

Note coding:

Outliers: 1 is the non-outlier group Outliers: 2 is the outlier group

```
##
## Attaching package: 'kableExtra'

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

Outlier	Age	BMI	OneYearMTPM	OneToTwoMTPM
1	62.09 $\pm$ 8.72	33.78 $\pm$ 6.80	0.38 $\pm$ 0.23	0.05 $\pm$ 0.15
2	66.00 $\pm$ 2.83	32.80 $\pm$ 6.51	1.56 $\pm$ 0.76	0.88 $\pm$ 1.24

Outlier	Age	BMI	OneYearMTPM	OneToTwoMTPM
1	65.51 $\pm$ 7.18	32.32 $\pm$ 5.98	1.01 $\pm$ 0.96	0.05 $\pm$ 0.19
2	60.20 $\pm$ 2.68	32.24 $\pm$ 2.37	3.05 $\pm$ 1.29	0.24 $\pm$ 0.92

Outlier	Age	BMI	OneYearMTPM	OneToTwoMTPM
1	66.27 $\pm$ 8.46	30.97 $\pm$ 5.38	0.47 $\pm$ 0.29	0.03 $\pm$ 0.13
2	65.00 $\pm$ NA	34.40 $\pm$ NA	1.66 $\pm$ NA	0.89 $\pm$ NA

Outlier	Age	BMI	OneYearMTPM	OneToTwoMTPM
1	64.88 $\pm$ 7.68	30.60 $\pm$ 4.45	0.75 $\pm$ 0.53	0.05 $\pm$ 0.16

2	67.20 $\pm$ 2.59	33.08 $\pm$ 3.40	2.41 $\pm$ 1.35	0.45 $\pm$ 0.77

```
features <- c("Age", "BMI", "OneYearMTPM", "OneToTwoMTPM")
rbind(FC_summary[, features], FC_Outliers[, features])
```

```
## # A tibble: 4 x 4
##   Age                 BMI                 OneYearMTPM        OneToTwoMTPM
## * <chr>               <chr>               <chr>              <chr>
## 1 "62.09 $\\pm$ 8.72" "33.78 $\\pm$ 6.80" "0.38 $\\pm$ 0.23" "0.05 $\\pm$ 0.15"
## 2 "66.00 $\\pm$ 2.83" "32.80 $\\pm$ 6.51" "1.56 $\\pm$ 0.76" "0.88 $\\pm$ 1.24"
## 3 "64"                "37.4"              "1.024"            "1.758"
## 4 "68"                "28.2"              "2.092"            "0.011000000000000~
```

```
rbind(FU_summary[, features], FU_Outliers[, features])
```

```
## # A tibble: 7 x 4
##   Age                 BMI                 OneYearMTPM        OneToTwoMTPM
## * <chr>               <chr>               <chr>              <chr>
## 1 "65.51 $\\pm$ 7.18" "32.32 $\\pm$ 5.98" "1.01 $\\pm$ 0.96" "0.05 $\\pm$ 0.19"
## 2 "60.20 $\\pm$ 2.68" "32.24 $\\pm$ 2.37" "3.05 $\\pm$ 1.29" "0.24 $\\pm$ 0.92"
## 3 "65"                "28.3"              "3.362"            "1.251"
## 4 "59"                "32.3"              "2.304"            "-0.721"
## 5 "59"                "34.1"              "5.189"            "0.635"
## 6 "59"                "34.1"              "2.085"            "-0.756"
## 7 "59"                "32.4"              "2.334"            "0.781"
```

```
rbind(MC_summary[, features], MC_Outliers[, features])
```

```
## # A tibble: 3 x 4
##   Age                 BMI                 OneYearMTPM        OneToTwoMTPM
## * <chr>               <chr>               <chr>              <chr>
## 1 "66.27 $\\pm$ 8.46" "30.97 $\\pm$ 5.38" "0.47 $\\pm$ 0.29" "0.03 $\\pm$ 0.13"
## 2 "65.00 $\\pm$  NA"  "34.40 $\\pm$  NA"  "1.66 $\\pm$  NA"  "0.89 $\\pm$  NA"
## 3 "65"                "34.4"              "1.662"            "0.888"
```

```
rbind(MU_summary[, features], MU_Outliers[, features])
```

```
## # A tibble: 7 x 4
##   Age                 BMI                 OneYearMTPM        OneToTwoMTPM
## * <chr>               <chr>               <chr>              <chr>
## 1 "64.88 $\\pm$ 7.68" "30.60 $\\pm$ 4.45" "0.75 $\\pm$ 0.53" "0.05 $\\pm$ 0.16"
## 2 "67.20 $\\pm$ 2.59" "33.08 $\\pm$ 3.40" "2.41 $\\pm$ 1.35" "0.45 $\\pm$ 0.77"
## 3 "70"                "31"                "3.799"            "-0.05099999999999~
## 4 "69"                "29.3"              "0.511"            "0.939"
## 5 "68"                "33.1"              "2.721"            "-0.616"
## 6 "64"                "38.3"              "3.406"            "1.302"
## 7 "65"                "33.7"              "1.598"            "0.661"
```

Assignment Questions

9a. For each of the 4 categories (UF, CF, UM, CM) separately, summarize the outlier groups, considering what does and/or doesn't set them apart from the non-outlier groups in their category. [8 marks]

Boxplots would be the most visual way to identify which metrics are responsible for the outlier classification but I'll manually compare the outlier values against the non-outlier range because there are only few outliers

- **CF**: OneYearMTPM is too high (>0.70) for both outliers. OneToTwoMTPM is too high for one outlier ($1.76 > 0.31$)
- **UF**: OneYearMTPM is too high (>2.28) for 4/5 outliers. OneToTwoMTPM is either too low or too high for all outliers ($x < -0.23, x > 0.37$)
- **CM**: Both OneYearMTPM and OneToTwoMTPM are too high for the only outlier
- **UM**: BMI is too high for 1/5 outliers (38.3>35.23). OneYearMTPM is too high for 3/5 outliers (>1.69). OneToTwoMTPM is either too low or too high for 4/5 outliers ($x < -0.19, x > 0.37$)

9b. Make a statement about the 4 outlier groups overall. i.e. are they each an outlier group for similar or different reasons? [2 marks]

In general, the OneYearMTPM and OneToTwoMTPM values are what ultimately cause the outlier classification. Age and BMI are typically within the acceptable range.

PART 2 - Logistic Regression

To perform a logistic regression we use the generalized linear model function (glm) but specify that the outcome variable is binary by including "family = bionomial".

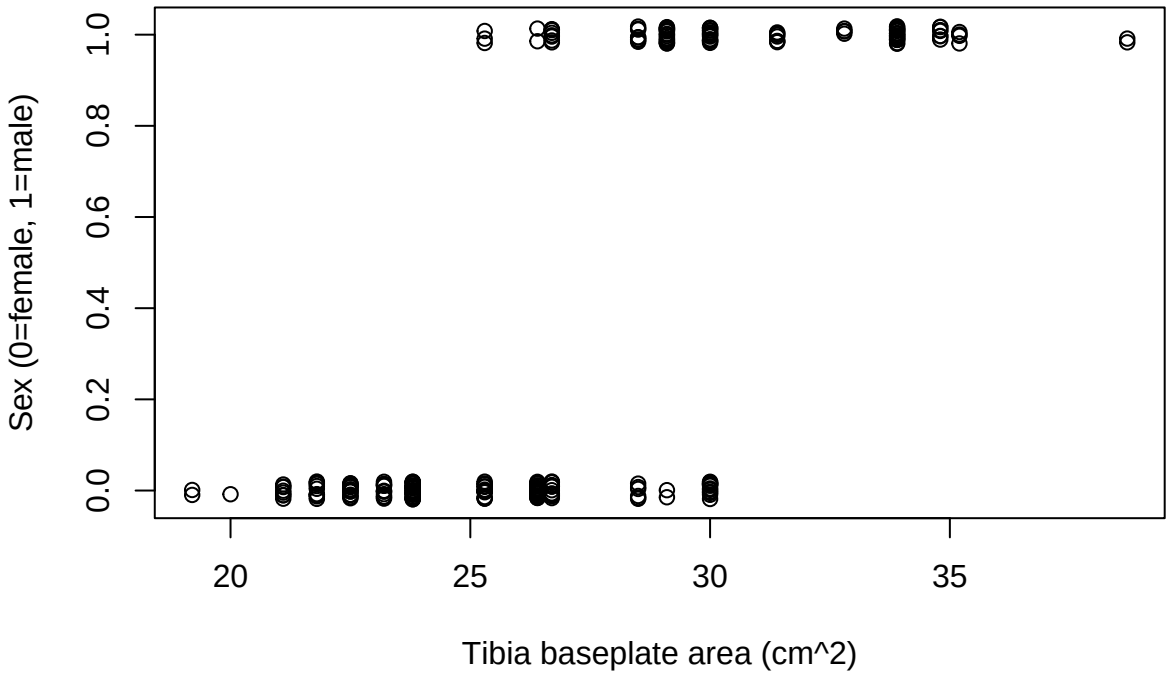
Examine the relationship between TibiaArea (i.e. implant size) and patient sex

```
Data$SexNumeric<-ifelse(Data$Sex == 1, 1, 0) #Recode with dummy variables for plot

#run logistic regression and show output
glm.TibSizeBySex <- glm(factor(SexNumeric) ~ TibiaArea, data=Data, family=binomial)
summary(glm.TibSizeBySex)
```

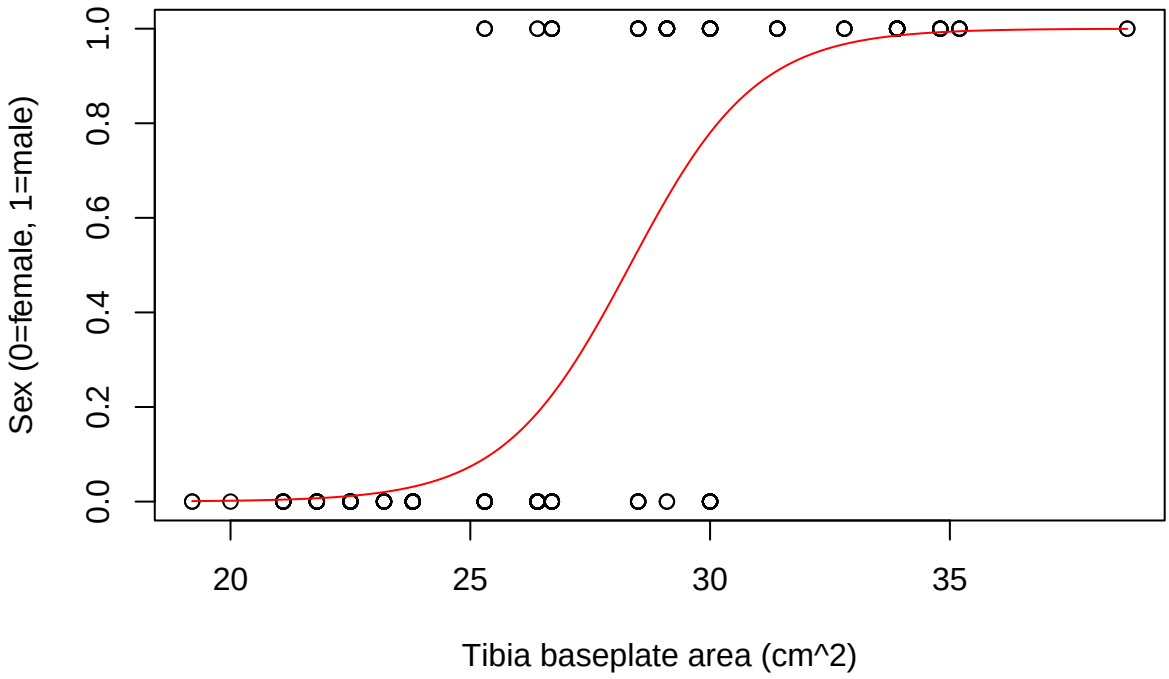
```
##
## Call:
## glm(formula = factor(SexNumeric) ~ TibiaArea, family = binomial,
##     data = Data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -21.47649    2.72515  -7.881 3.25e-15 ***
## TibiaArea     0.75805    0.09713   7.804 5.99e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 330.95  on 246  degrees of freedom
## Residual deviance: 154.74  on 245  degrees of freedom
## AIC: 158.74
```

```
##
## Number of Fisher Scoring iterations: 6
```

```
#For visualization, generate a vector of all possible values of TibiaArea
xv<-seq(min(Data$TibiaArea), max(Data$TibiaArea), 0.1)

#Use the predict function to calculate the probability of being female or male based on TibiaArea
yv<-predict(glm.TibSizeBySex, list(TibiaArea = xv), type = "response")

#Plot the predicted response
plot(Data$TibiaArea, jitter(Data$SexNumeric, amount=0), xlab="Tibia baseplate area (cm^2)", ylab= "Sex
```




```
plot(Data$TibiaArea, Data$SexNumeric, xlab="Tibia baseplate area (cm^2)", ylab= "Sex (0=female, 1=male)
lines(xv,yv, col= "red")
```



For a single Tibia baseplate size, predict the probability of being female or male

```
predict(glm.TibSizeBySex, list(TibiaArea = 25), type = "response")
```

```
##          1
## 0.07410197
```

```
predict(glm.TibSizeBySex, list(TibiaArea = 28), type = "response")
```

```
##         1
## 0.4375327
```

```
predict(glm.TibSizeBySex, list(TibiaArea = 30), type = "response")
```

```
##         1
## 0.7798706
```

Assignment Questions

10a. Is TibiaArea a good predictor of patient sex? How do you know? [2 marks]

No, there's too much overlap between male and female datapoints in the TibiaArea $\in [25, 30]$ range. Visually the line is very smooth in the middle which indicates a region of uncertainty whereas it should be almost like a zig-zag for more confident classification. There's certainly a trend between Sex and TibiaArea but the metric alone is insufficient, especially for TibiaArea values between [25, 30].

10b. Try 2 additional sizes for TibiaArea (in addition to 40 cm2 as in the example) and discuss the probabilities of the patient being female or male for all 3 sizes [3 marks]

- $25\mathbf{cm}^2$: 7% chance to be male
- $28\mathbf{cm}^2$: 44% chance to be male
- $30\mathbf{cm}^2$: 78% chance to be male

I've prompted the model with datapoints in the center region ([25, 30]) where there's substantial overlap between male and female datapoints. The moderate probabilities ($\approx 50\%$) demonstrate the model is uncertain in this region, which is concerning because a majority of the datapoints appear to fall within this region.

Literature indicates that individuals with greater that 0.2 mm of MTPM migration from 1-2 years have "continuous migration" and are at risk for later failure

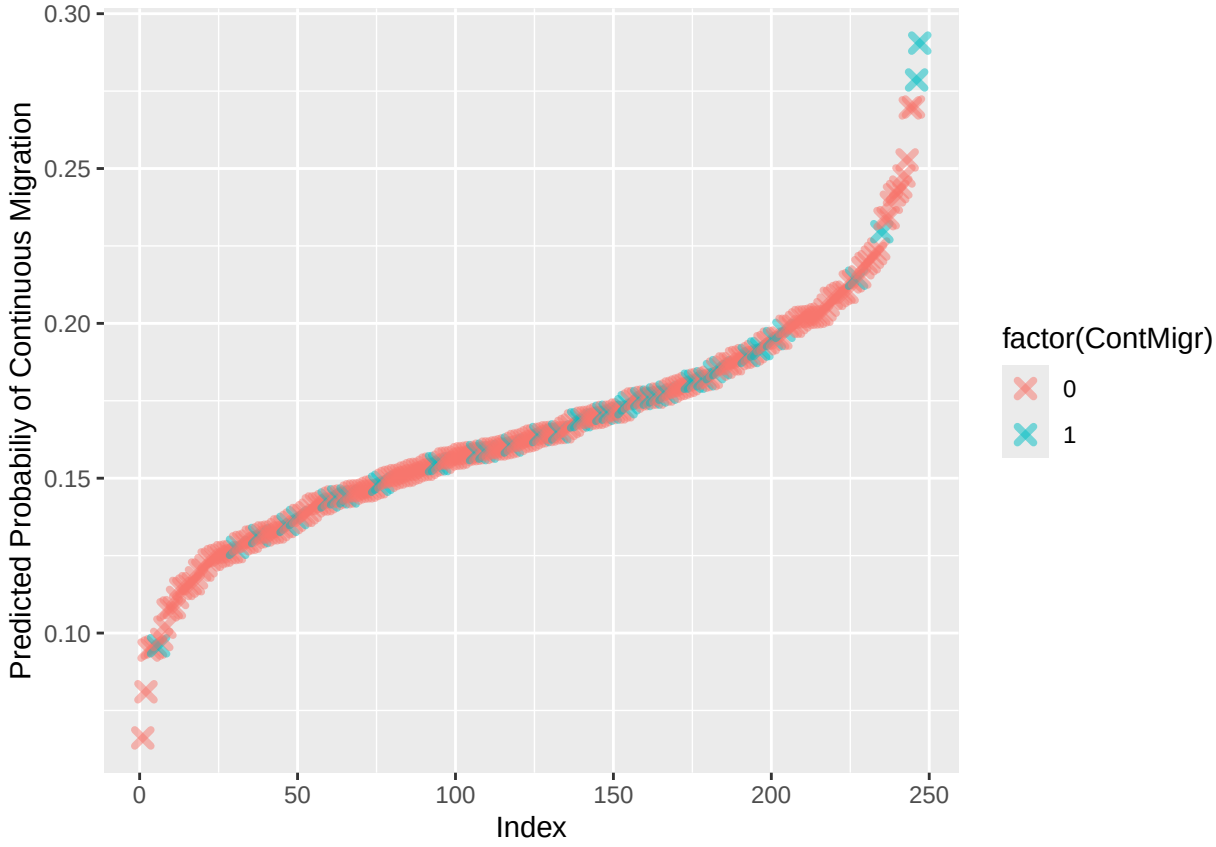
Do the patient demographics and implant fixation predict who will have continuous migration?

```
#Code a ContMigr variabale based on 1-2 year MTPM
Data$ContMigr[Data$OneToTwoMTPM<= 0.2] <- 0
Data$ContMigr[Data$OneToTwoMTPM>0.2] <- 1
Data$ContMigr<-as.numeric(Data$ContMigr)

#Fit logistic regression model
glm.contmigr.1 <- glm(factor(ContMigr) ~ factor(Fix)+factor(Sex) + Age+ BMI, data=Data, family=binomial
summary(glm.contmigr.1)
```

```
##
## Call:
## glm(formula = factor(ContMigr) ~ factor(Fix) + factor(Sex) +
##     Age + BMI, family = binomial, data = Data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.46566    2.18992  -2.039   0.0414 *
## factor(Fix)2  0.13598    0.34792   0.391   0.6959
## factor(Sex)2  0.26560    0.36857   0.721   0.4711
## Age           0.02541    0.02443   1.040   0.2982
## BMI           0.02999    0.03087   0.971   0.3314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 222.04  on 246  degrees of freedom
## Residual deviance: 219.79  on 242  degrees of freedom
## AIC: 229.79
```

```
##
## Number of Fisher Scoring iterations: 4
```

```
#Plot the response curve
predicted.data<-data.frame(probability.of.ContMigr=glm.contmigr.1$fitted.values, ContMigr=Data$ContMigr
predicted.data<-predicted.data[order(predicted.data$probability.of.ContMigr, decreasing = FALSE),]
predicted.data$rank<-1:nrow(predicted.data)

ggplot(data = predicted.data, aes(x = rank, y = probability.of.ContMigr)) +
  geom_point(aes(colour = factor(ContMigr)), alpha=.5, shape = 4, stroke = 2) +
  xlab("Index")+
  ylab("Predicted Probabiliy of Continuous Migration")
```



Let's also calculate the confidence intervals for our estimates

```
## Waiting for profiling to be done...
```

```
##              Estimate glm.contmigr.1 Std..Error    z.value   Pr...z..
## (Intercept)              -4.46566127 2.18991849 -2.0391906 0.04143101
## factor(Fix)2              0.13597863 0.34791780  0.3908355 0.69591884
## factor(Sex)2              0.26560368 0.36857327  0.7206265 0.47113932
## Age                       0.02540864 0.02442654  1.0402063 0.29824407
## BMI                       0.02998532 0.03087017  0.9713364 0.33138079
##                    X2.5..     X97.5..
## (Intercept)  -8.92365348 -0.31914645
## factor(Fix)2 -0.54749878  0.82375655
## factor(Sex)2 -0.44453802  1.00978579
## Age          -0.02086155  0.07506072
## BMI          -0.03160781  0.09029257
```

Assignment Question

11. Do you think this model is a good predictor of continuous migration? Why or why not? [2 marks]

I do not agree because there is substantial overlap between the logistic regression curves whereas there should be a distinctive y-shift upward for the ContMigr=1 (OneToTwoMTPM>0.2) group to indicate a higher probability of experiencing continuous migration. In terms of model significance, the fixture type and demographics are not statistically significant as all the p values exceed 0.05. This indicates that none of the explanatory variables have a significant effect on continuous migration so the model is poorly constructed.