Wine Customer Segment Analysis

Ani Laliashvili

9/22/2020

# Load Packages  
library(tidyverse)

## -- Attaching packages ---------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.1 v dplyr 1.0.0  
## v tidyr 1.1.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(cluster)  
library(haven)  
library(ggdendro)  
library(NbClust)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(klaR)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

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

library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

library(rlang)

##   
## Attaching package: 'rlang'

## The following object is masked from 'package:data.table':  
##   
## :=

## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

library(dplyr)  
library(NbClust)  
library(ggpubr)  
theme\_set(theme\_pubr())  
library(corrplot)

## corrplot 0.84 loaded

library(ggplot2)  
library(ggpubr)

# Load Data  
#path <- file.path("../input", "ClusterAnonFaculty.sav", fsep="/") #Set path  
#mydataraw <- read\_sav(path) # Read .sav file from the path  
  
mydataraw <- read\_csv("C:/Users/anila/OneDrive/Desktop/Wine.csv")

## Parsed with column specification:  
## cols(  
## Alcohol = col\_double(),  
## Malic\_Acid = col\_double(),  
## Ash = col\_double(),  
## Ash\_Alcanity = col\_double(),  
## Magnesium = col\_double(),  
## Total\_Phenols = col\_double(),  
## Flavanoids = col\_double(),  
## Nonflavanoid\_Phenols = col\_double(),  
## Proanthocyanins = col\_double(),  
## Color\_Intensity = col\_double(),  
## Hue = col\_double(),  
## OD280 = col\_double(),  
## Proline = col\_double(),  
## Customer\_Segment = col\_double()  
## )

head(mydataraw)

## # A tibble: 6 x 14  
## Alcohol Malic\_Acid Ash Ash\_Alcanity Magnesium Total\_Phenols Flavanoids  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 14.2 1.71 2.43 15.6 127 2.8 3.06  
## 2 13.2 1.78 2.14 11.2 100 2.65 2.76  
## 3 13.2 2.36 2.67 18.6 101 2.8 3.24  
## 4 14.4 1.95 2.5 16.8 113 3.85 3.49  
## 5 13.2 2.59 2.87 21 118 2.8 2.69  
## 6 14.2 1.76 2.45 15.2 112 3.27 3.39  
## # ... with 7 more variables: Nonflavanoid\_Phenols <dbl>, Proanthocyanins <dbl>,  
## # Color\_Intensity <dbl>, Hue <dbl>, OD280 <dbl>, Proline <dbl>,  
## # Customer\_Segment <dbl>

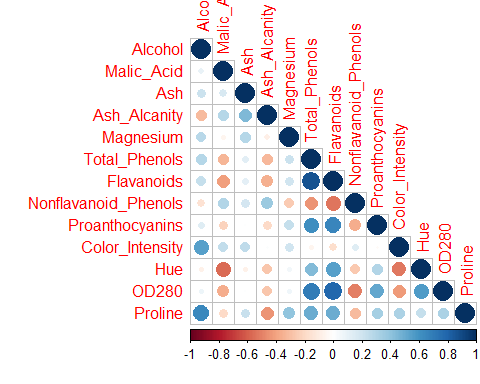
head(mydataraw)

## # A tibble: 6 x 14  
## Alcohol Malic\_Acid Ash Ash\_Alcanity Magnesium Total\_Phenols Flavanoids  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 14.2 1.71 2.43 15.6 127 2.8 3.06  
## 2 13.2 1.78 2.14 11.2 100 2.65 2.76  
## 3 13.2 2.36 2.67 18.6 101 2.8 3.24  
## 4 14.4 1.95 2.5 16.8 113 3.85 3.49  
## 5 13.2 2.59 2.87 21 118 2.8 2.69  
## 6 14.2 1.76 2.45 15.2 112 3.27 3.39  
## # ... with 7 more variables: Nonflavanoid\_Phenols <dbl>, Proanthocyanins <dbl>,  
## # Color\_Intensity <dbl>, Hue <dbl>, OD280 <dbl>, Proline <dbl>,  
## # Customer\_Segment <dbl>

# Prepare Data  
  
  
# Extract variable names from dataset  
d <- dim(mydataraw)[2]  
for(i in 1:d) {  
 var <- names(mydataraw)[i]  
 nam <- paste("var", i, sep = "")  
 assign(nam, var)  
   
}  
  
# Remove variables to be excluded from analysis  
#mydata <- mydataraw[, setdiff(names(mydataraw), c(var1, var8, var7)), with = FALSE] ###CHANGE  
mydata <- mydataraw[-14]  
  
# Capture the new variable names  
d <- dim(mydata)[2]  
for(i in 1:d) {  
 var <- names(mydata)[i]  
 nam <- paste("var", i, sep = "")  
 assign(nam, var)  
   
}  
  
# Listwise deletion of missing values  
mydata <- na.omit(mydata)   
  
head(mydata)

## # A tibble: 6 x 13  
## Alcohol Malic\_Acid Ash Ash\_Alcanity Magnesium Total\_Phenols Flavanoids  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 14.2 1.71 2.43 15.6 127 2.8 3.06  
## 2 13.2 1.78 2.14 11.2 100 2.65 2.76  
## 3 13.2 2.36 2.67 18.6 101 2.8 3.24  
## 4 14.4 1.95 2.5 16.8 113 3.85 3.49  
## 5 13.2 2.59 2.87 21 118 2.8 2.69  
## 6 14.2 1.76 2.45 15.2 112 3.27 3.39  
## # ... with 6 more variables: Nonflavanoid\_Phenols <dbl>, Proanthocyanins <dbl>,  
## # Color\_Intensity <dbl>, Hue <dbl>, OD280 <dbl>, Proline <dbl>

# Transform categorial variables into numerical flag values  
mydata <- data.frame(model.matrix(~ .- 1, data = mydata))  
# Correlations between variables  
mydata %>% mutate\_if(is.character,as.numeric) %>% cor %>% corrplot(type = "lower") #all



# Store variable names  
var <- data.frame("")  
d <- dim(mydataraw)[2]  
for(i in 1:d) {  
 var <- cbind(var,names(mydataraw)[i])   
}

# Rename columns for easier use  
names(mydata)[1:ncol(mydata)]<- paste0(rep("var", each=10), "\_",1:d)

## Warning in names(mydata)[1:ncol(mydata)] <- paste0(rep("var", each = 10), :  
## number of items to replace is not a multiple of replacement length

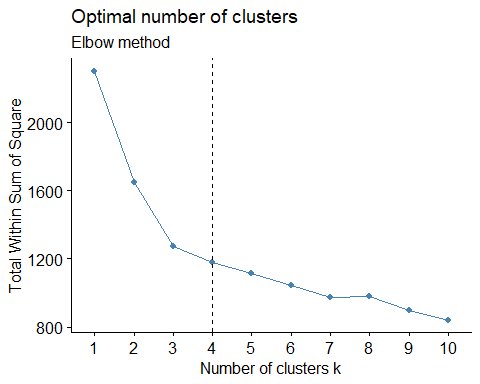
head(mydata)

## var\_1 var\_2 var\_3 var\_4 var\_5 var\_6 var\_7 var\_8 var\_9 var\_10 var\_11 var\_12  
## 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92  
## 2 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40  
## 3 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17  
## 4 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45  
## 5 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93  
## 6 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85  
## var\_13  
## 1 1065  
## 2 1050  
## 3 1185  
## 4 1480  
## 5 735  
## 6 1450

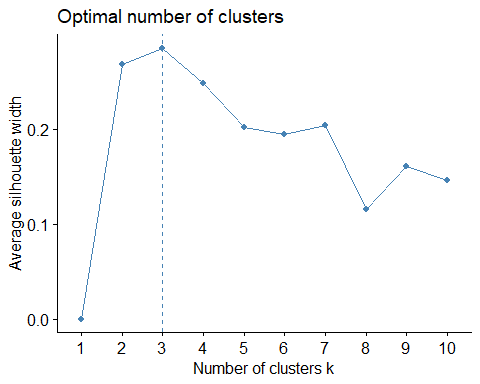
# Set variable paths  
var1path <- mydata$var\_1  
var2path <- mydata$var\_2  
var3path <- mydata$var\_3  
var4path <- mydata$var\_4  
var5path <- mydata$var\_5  
var6path <- mydata$var\_6  
var7path <- mydata$var\_7  
var8path <- mydata$var\_8  
var9path <- mydata$var\_9  
var10path <- mydata$var\_10  
var11path <- mydata$var\_11  
var12path <- mydata$var\_12  
var13path <- mydata$var\_13  
var14path <- mydata$var\_14  
var15path <- mydata$var\_15  
  
# Standardize data  
mydatastandardized <- scale(mydata) # Transform categorial variables into numerical flag values and standardize data  
  
head(mydatastandardized)

## var\_1 var\_2 var\_3 var\_4 var\_5 var\_6 var\_7  
## 1 1.5143408 -0.56066822 0.2313998 -1.1663032 1.90852151 0.8067217 1.0319081  
## 2 0.2455968 -0.49800856 -0.8256672 -2.4838405 0.01809398 0.5670481 0.7315653  
## 3 0.1963252 0.02117152 1.1062139 -0.2679823 0.08810981 0.8067217 1.2121137  
## 4 1.6867914 -0.34583508 0.4865539 -0.8069748 0.92829983 2.4844372 1.4623994  
## 5 0.2948684 0.22705328 1.8352256 0.4506745 1.27837900 0.8067217 0.6614853  
## 6 1.4773871 -0.51591132 0.3043010 -1.2860793 0.85828399 1.5576991 1.3622851  
## var\_8 var\_9 var\_10 var\_11 var\_12 var\_13  
## 1 -0.6577078 1.2214385 0.2510088 0.3611585 1.8427215 1.01015939  
## 2 -0.8184106 -0.5431887 -0.2924962 0.4049085 1.1103172 0.96252635  
## 3 -0.4970050 2.1299594 0.2682629 0.3174085 0.7863692 1.39122370  
## 4 -0.9791134 1.0292513 1.1827317 -0.4263410 1.1807407 2.32800680  
## 5 0.2261576 0.4002753 -0.3183774 0.3611585 0.4483365 -0.03776747  
## 6 -0.1755994 0.6623487 0.7298108 0.4049085 0.3356589 2.23274072

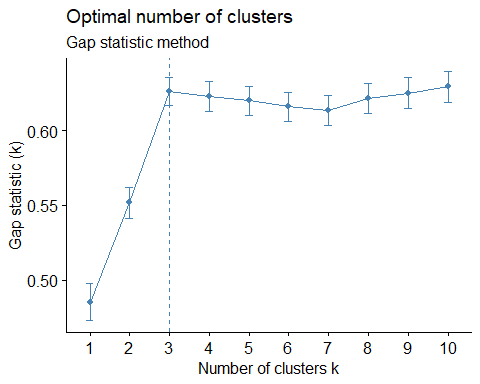
### Hierarchical Clustering   
# Determine the number of clusters  
fviz\_nbclust(mydatastandardized, kmeans, method = "wss") +  
 geom\_vline(xintercept = 4, linetype = 2)+  
 labs(subtitle = "Elbow method")



fviz\_nbclust(mydatastandardized, kmeans, method = c("silhouette", "wss", "gap\_stat"))



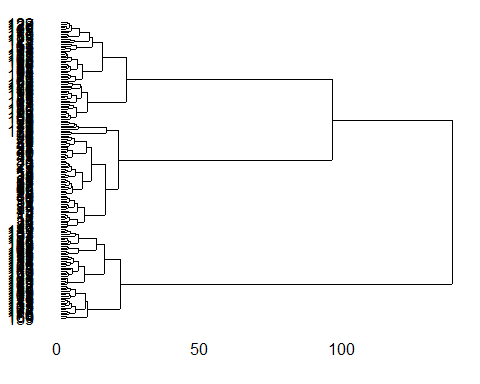
set.seed(123)  
fviz\_nbclust(mydatastandardized, kmeans, nstart = 25, method = "gap\_stat", nboot = 50)+  
 labs(subtitle = "Gap statistic method")



# Ask for user input - determine the number of cluster solutions to test (must between 2 and the number of rows in the database)  
#choose.level <- function(){readline("How many clustering solutions to test (< row numbers)? ")}   
#m <- as.integer(choose.level())

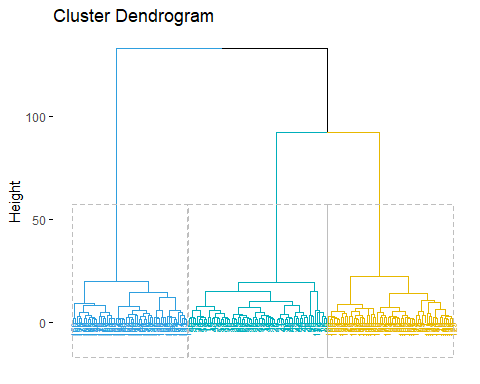
# Set cluster number  
 m <- 3 ### CHANGE  
  
# Set cluster names  
clust1name <- "Top" ### CHANGE  
clust2name <- "Mid" ### CHANGE  
clust3name <- "Lower" ### CHANGE

# Compute  
cluster <- mydatastandardized %>%   
 dist(method = "euclidean") %>%   
 hclust(method="ward.D")  
  
ggdendrogram(cluster,rotate = TRUE)



# Visualize  
fviz\_dend(cluster, k = m, # Cut in m groups  
 cex = 0.5, # label size  
 k\_colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),  
 color\_labels\_by\_k = TRUE, # color labels by groups  
 rect = TRUE # Add rectangle around groups  
)

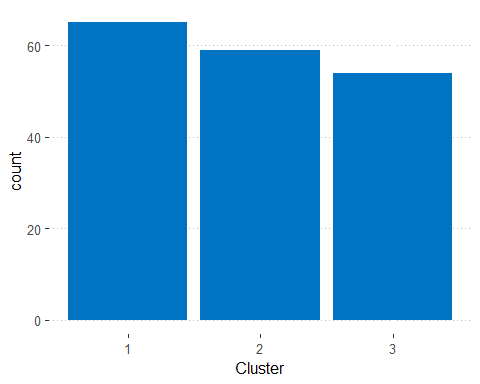
## Warning in get\_col(col, k): Length of color vector was longer than the number of  
## clusters - first k elements are used



# Bind data with respective clusters  
groups <- cutree(cluster, k=m) # cut tree into 2 clusters  
  
databind <- cbind(mydata, Cluster = groups) # Bind data with respective clusters  
  
Customer\_Segment <- mydataraw$Customer\_Segment  
databind1 <- cbind(databind, Customer\_Segment)  
head(databind1)

## var\_1 var\_2 var\_3 var\_4 var\_5 var\_6 var\_7 var\_8 var\_9 var\_10 var\_11 var\_12  
## 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92  
## 2 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40  
## 3 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17  
## 4 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45  
## 5 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93  
## 6 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85  
## var\_13 Cluster Customer\_Segment  
## 1 1065 1 1  
## 2 1050 1 1  
## 3 1185 1 1  
## 4 1480 1 1  
## 5 735 1 1  
## 6 1450 1 1

# Histogram  
theme\_set(theme\_pubr())  
ggplot(databind, aes(Cluster)) +  
 geom\_bar(fill = "#0073C2FF") +  
 theme\_pubclean()

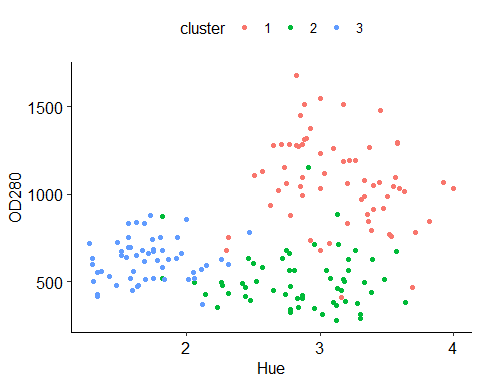
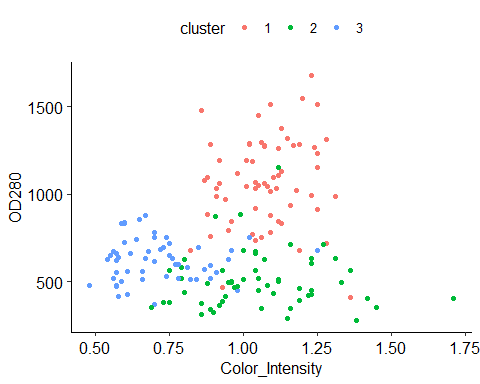
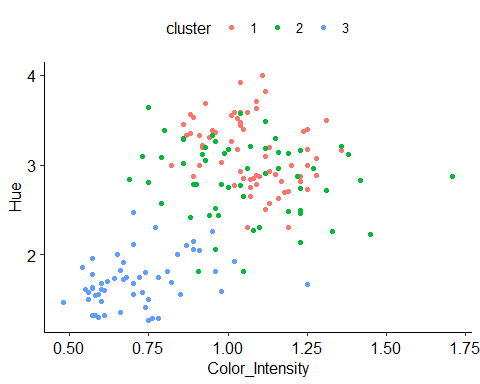
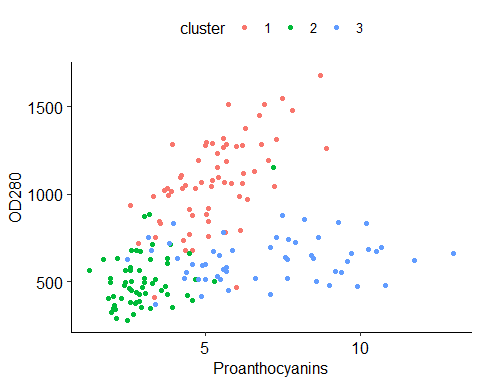
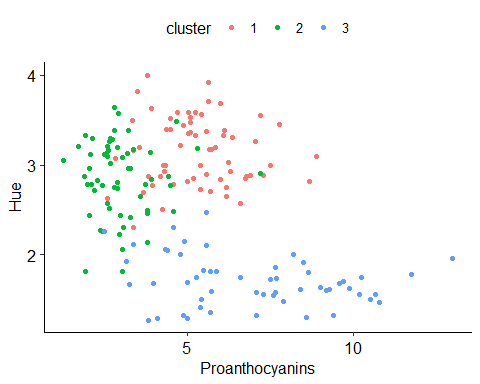
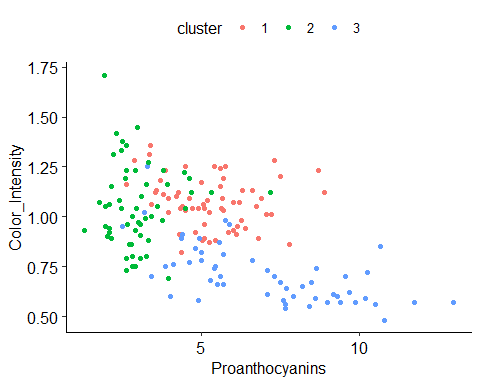
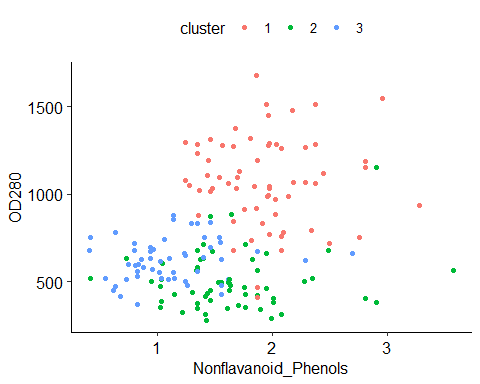
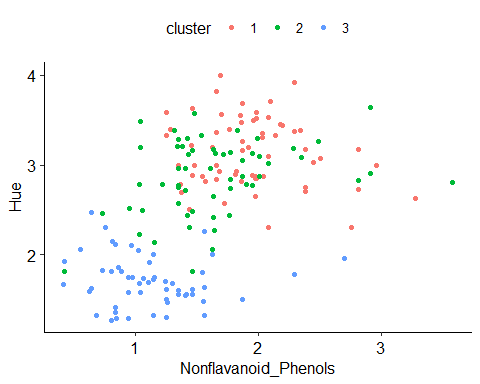
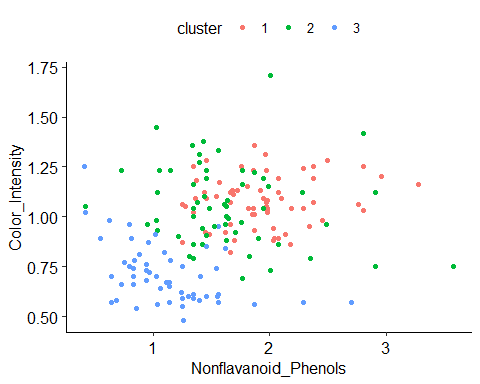
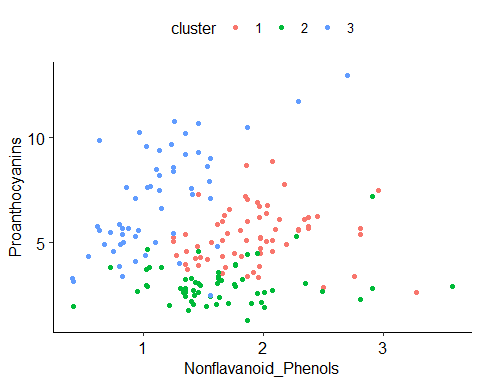
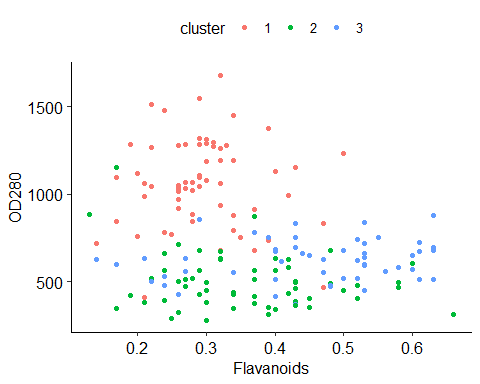
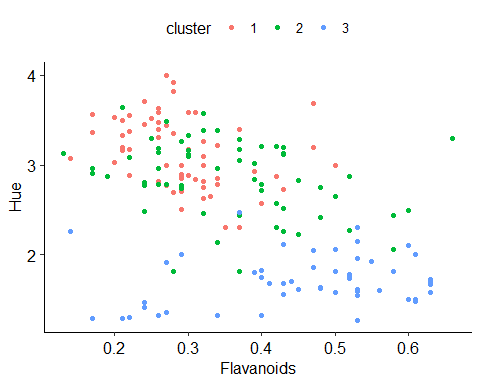
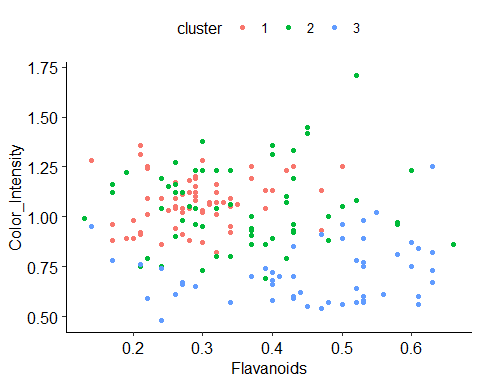
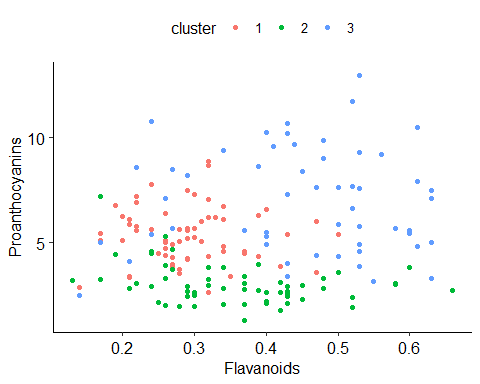
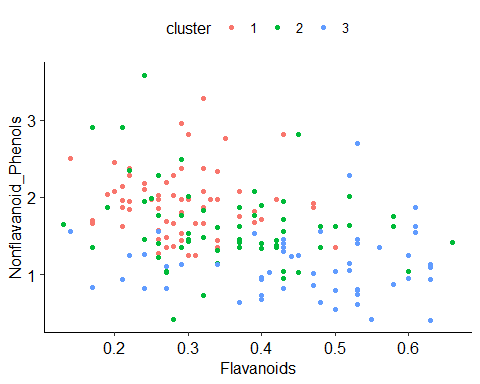
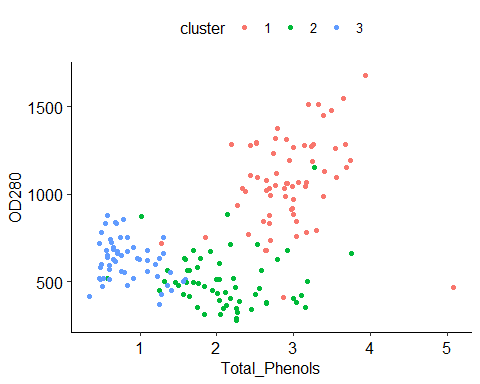
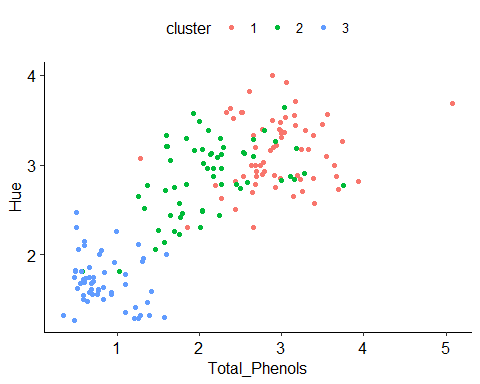
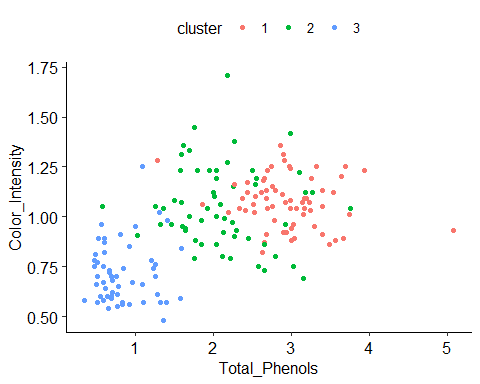
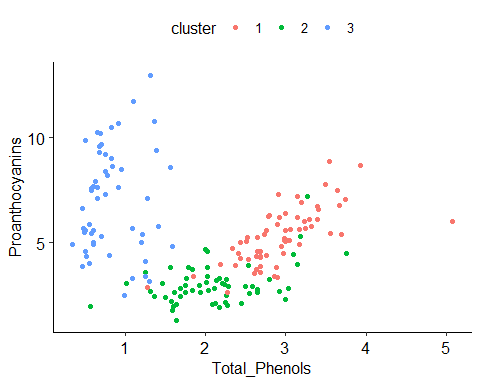
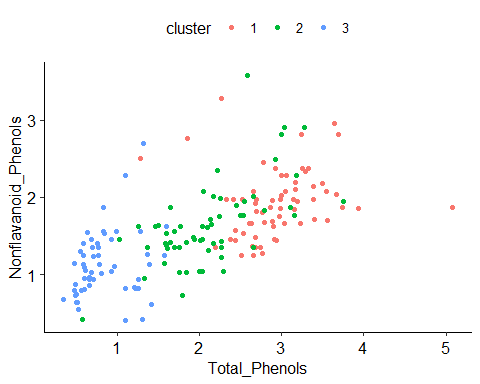
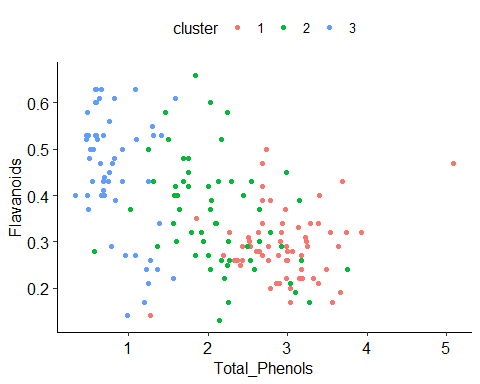
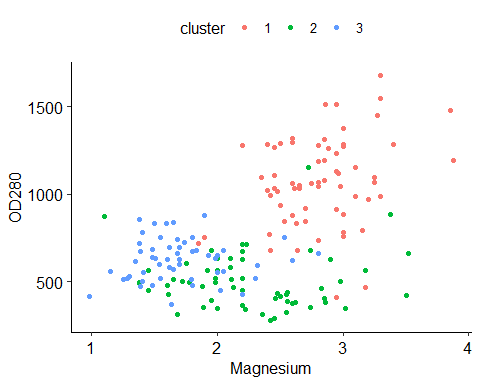
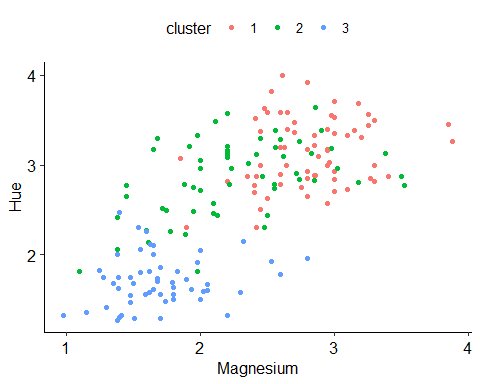
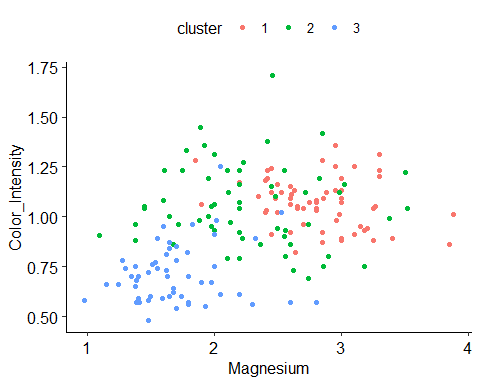
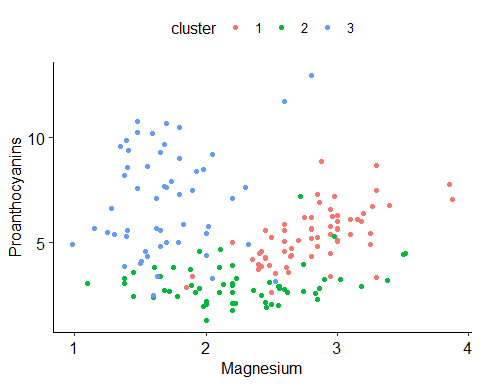
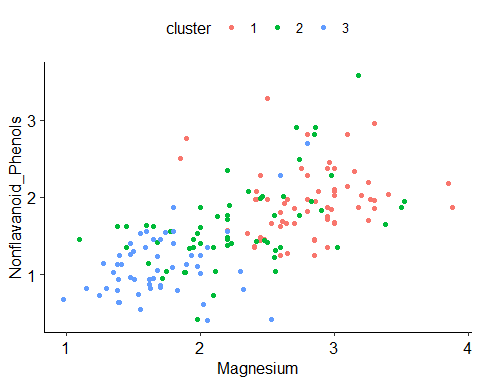
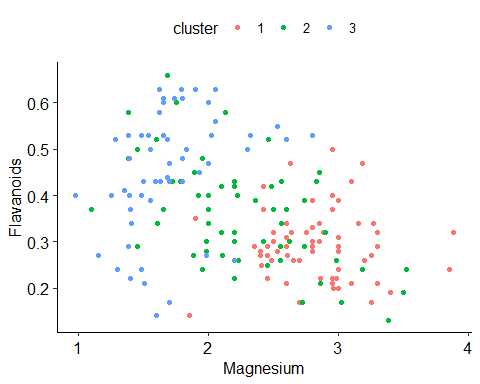
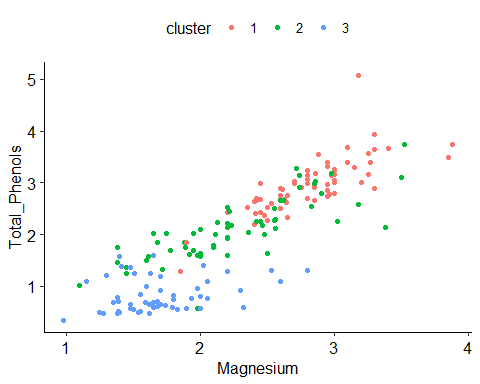
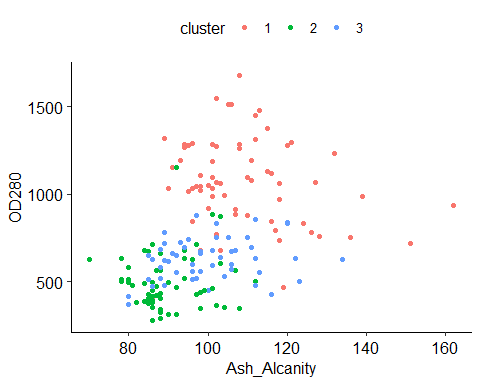
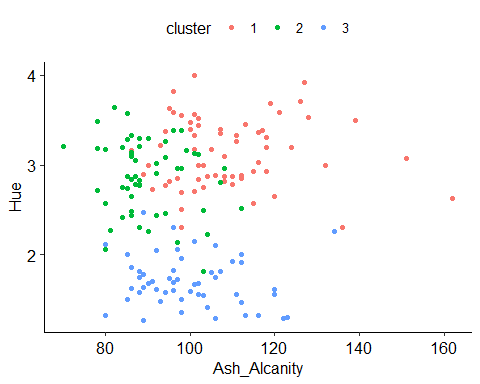
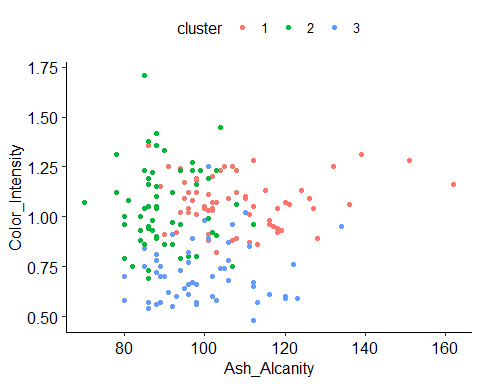
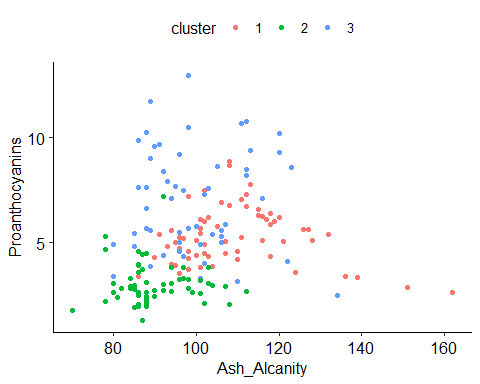
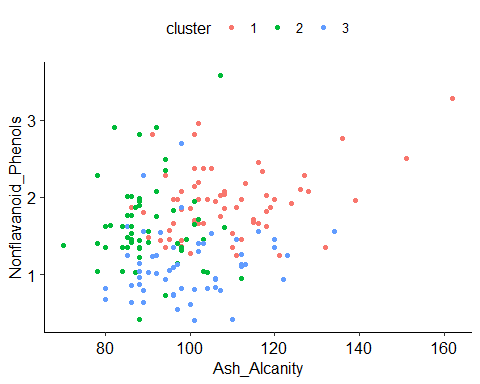
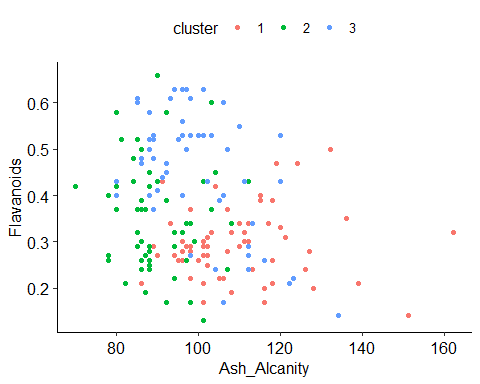
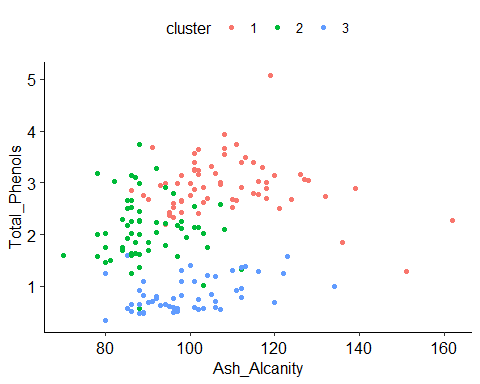
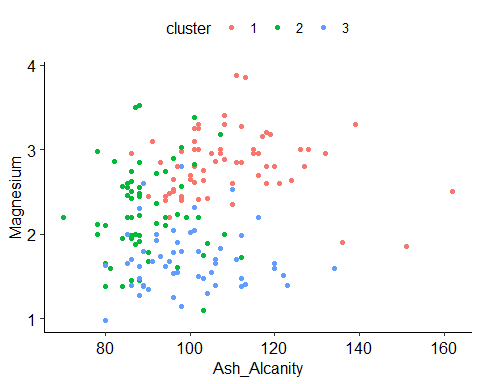
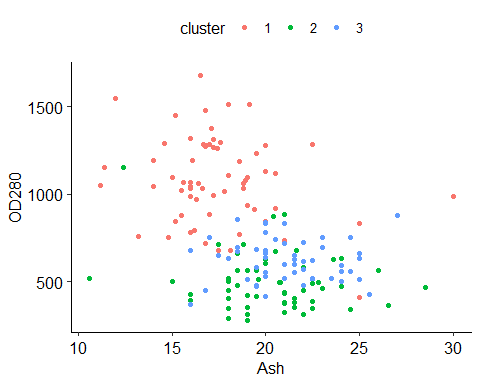
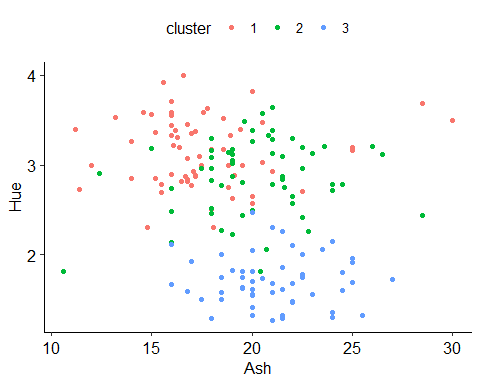
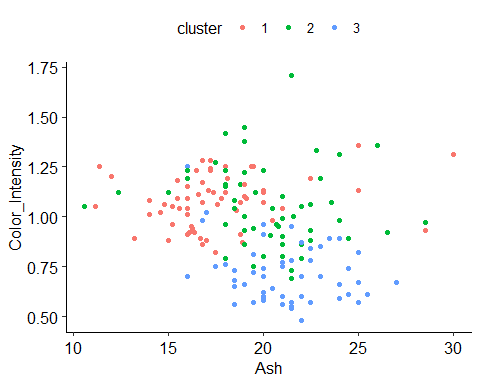
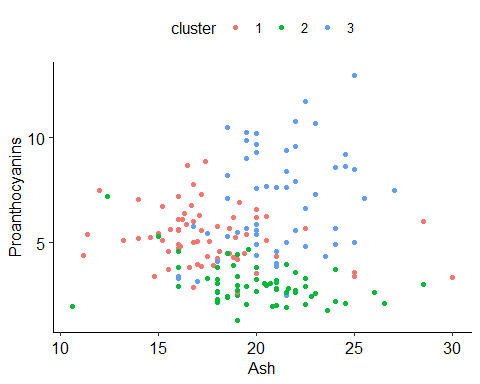
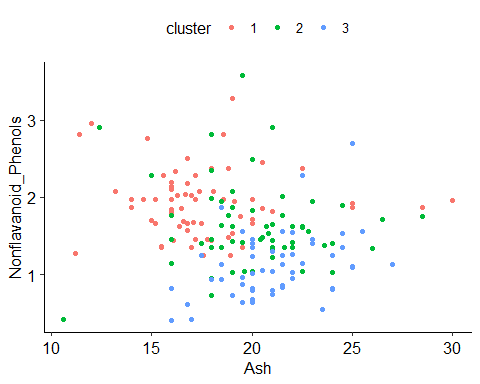
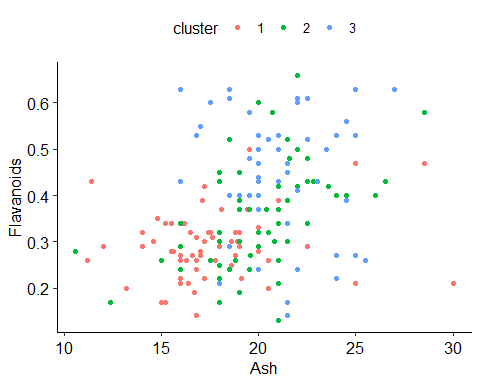
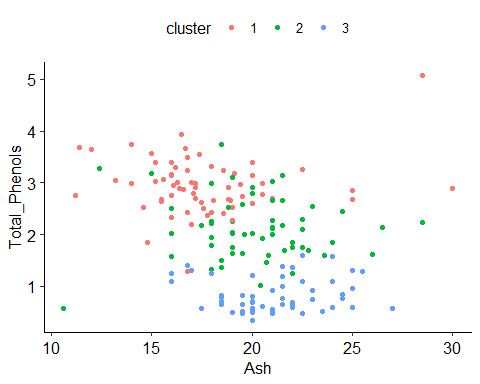
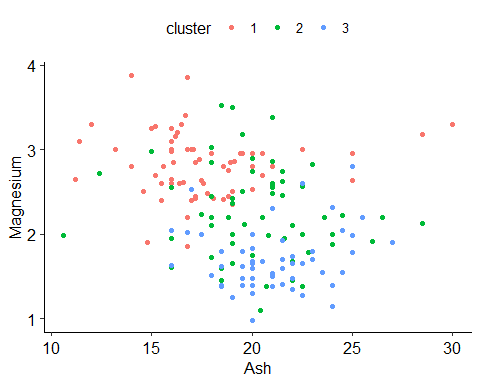
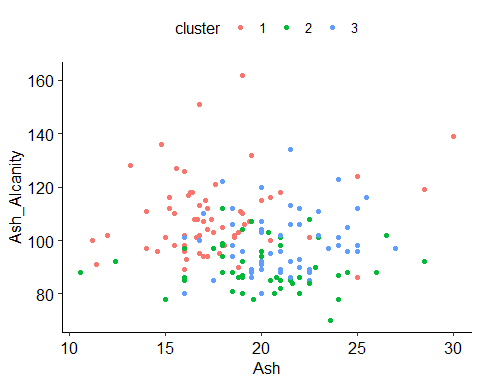
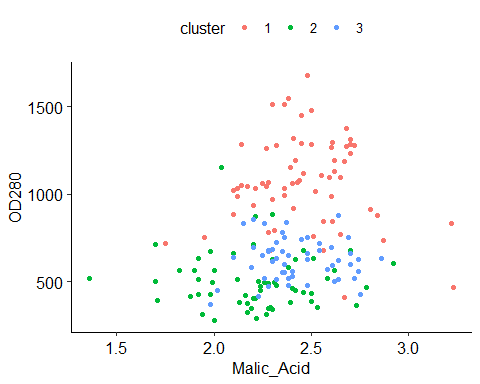
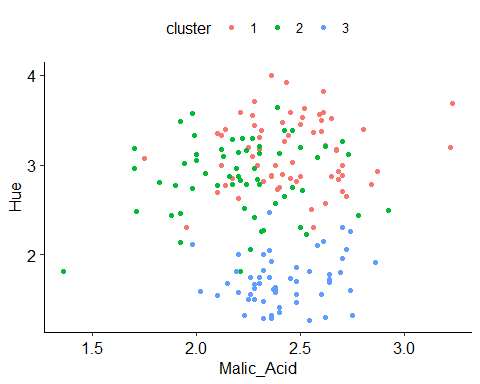
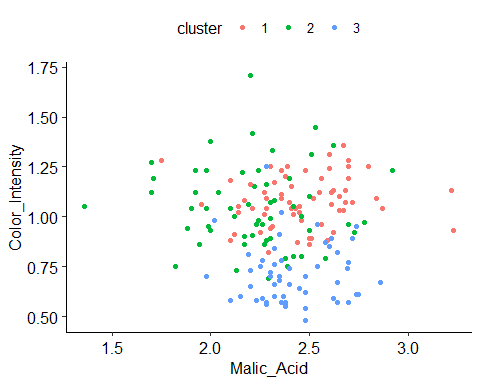
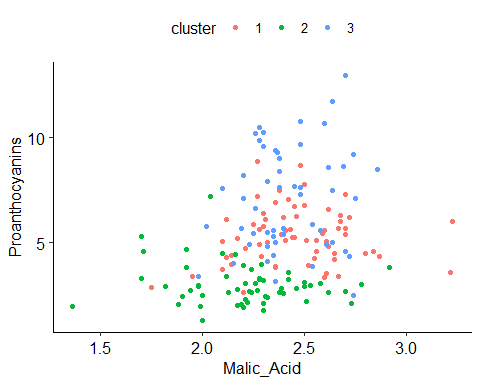
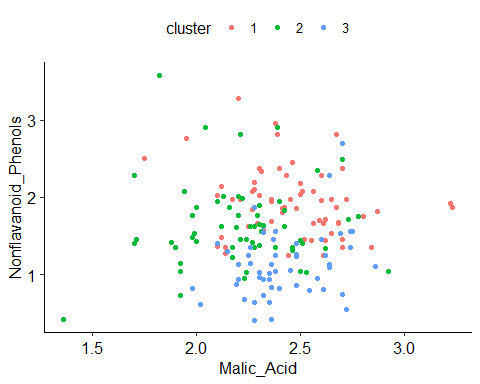
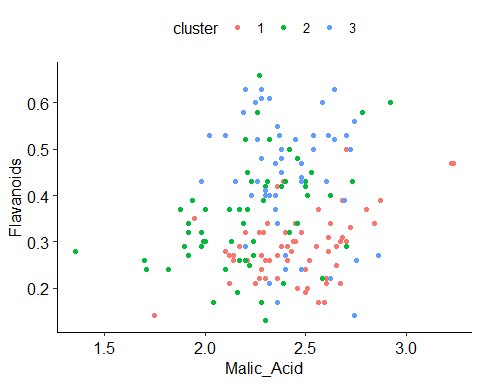
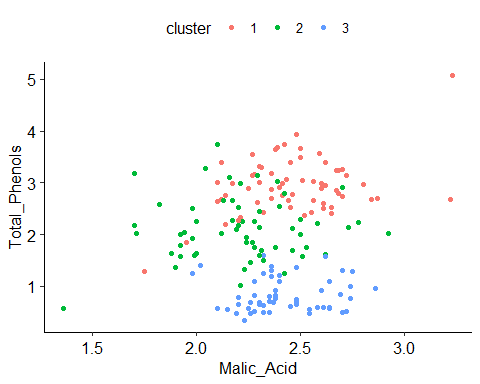
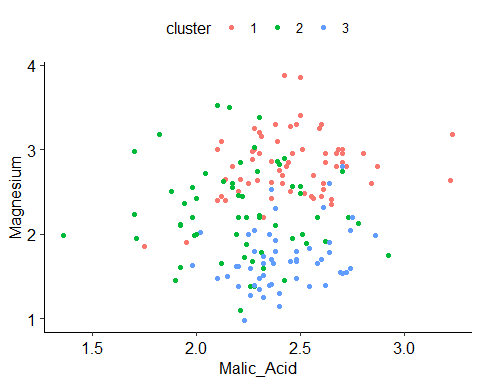
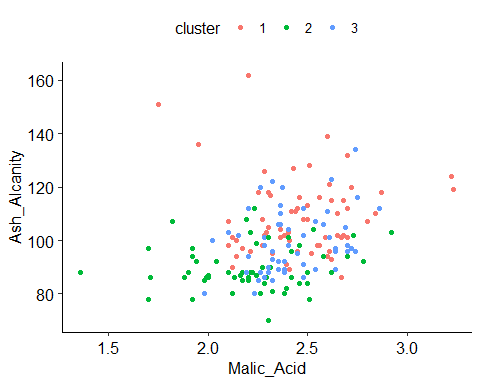
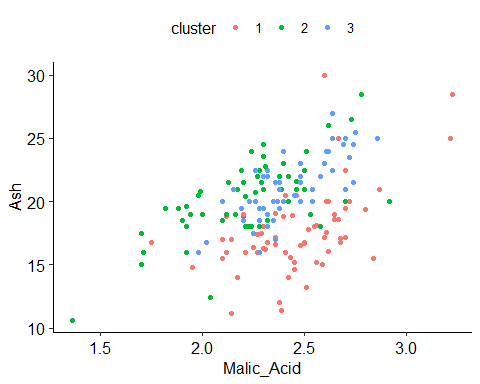
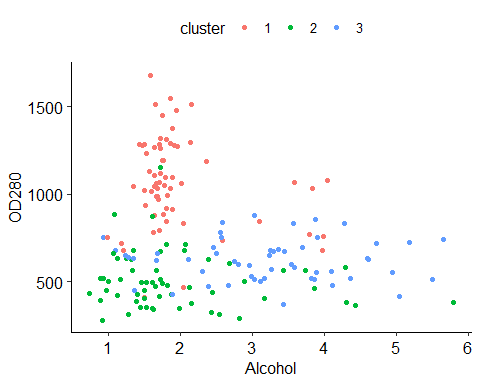
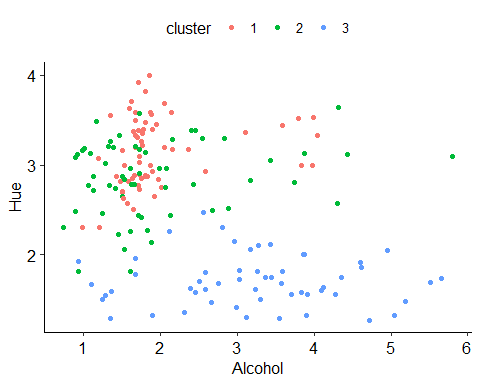
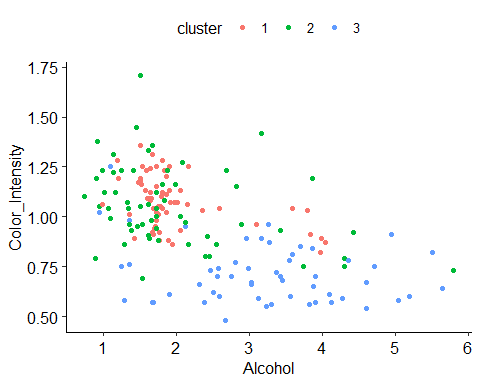
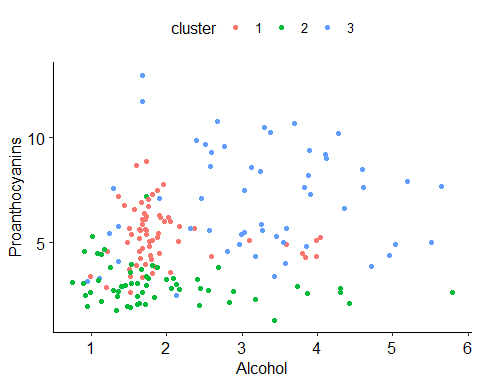
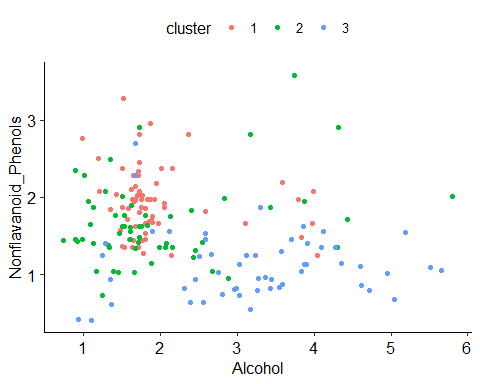
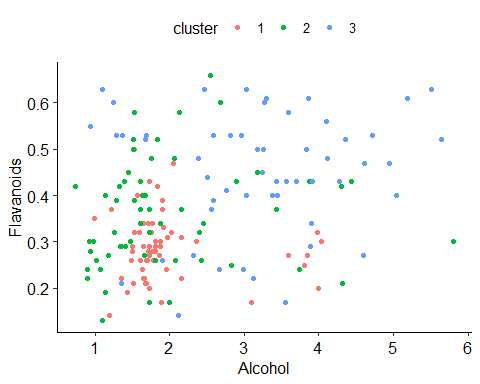
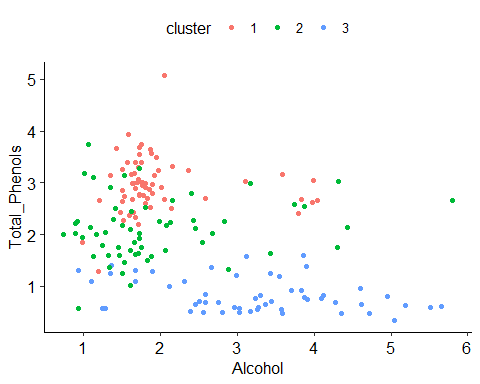
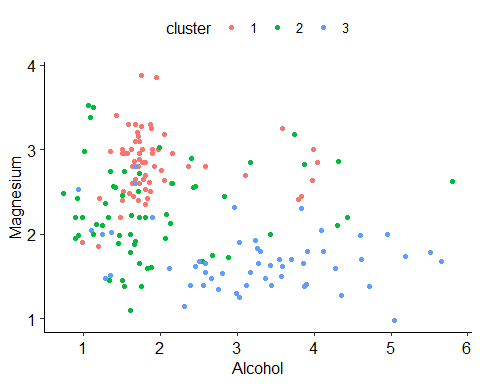
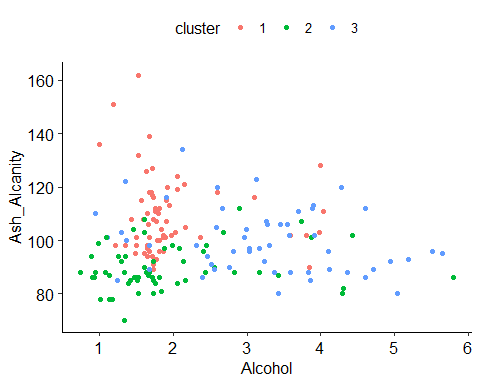
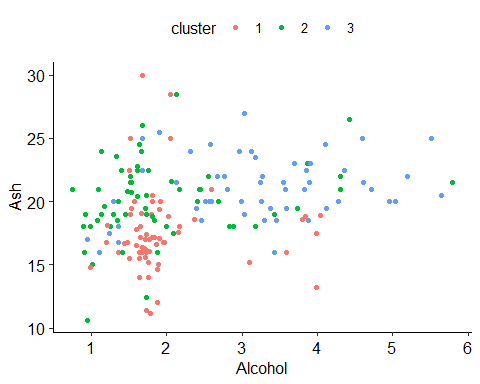
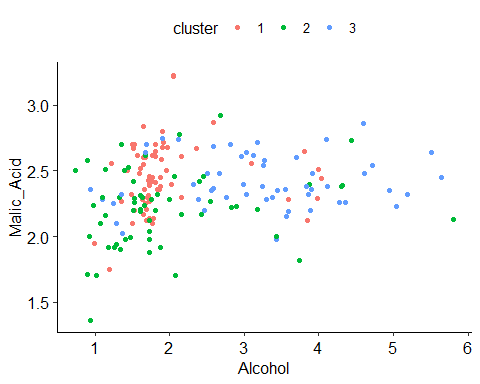
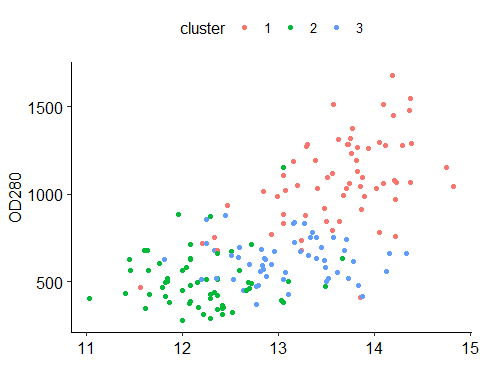
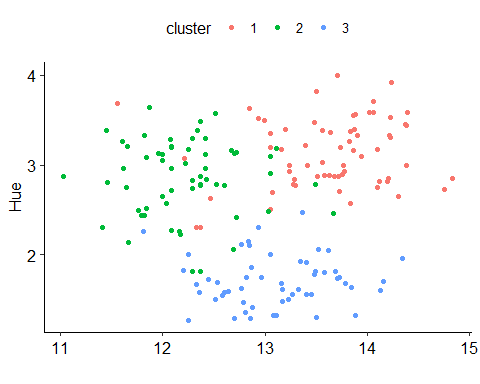
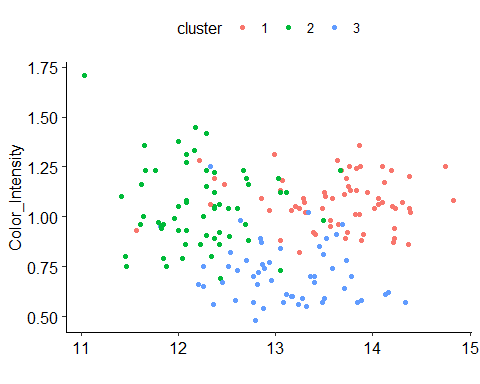
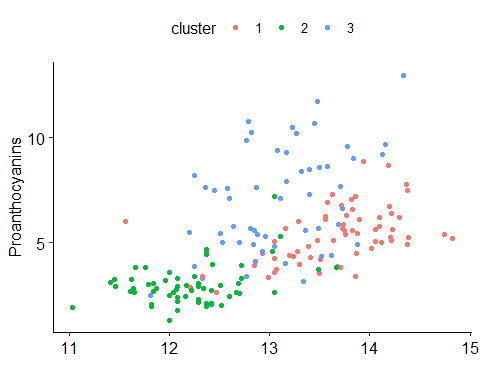
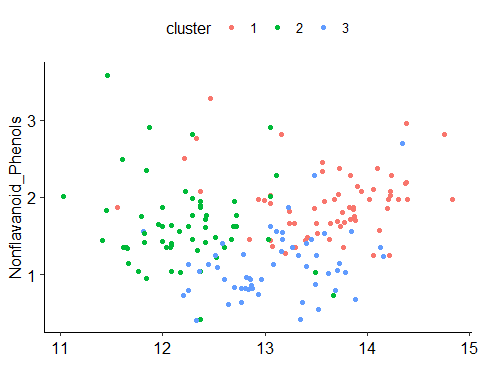
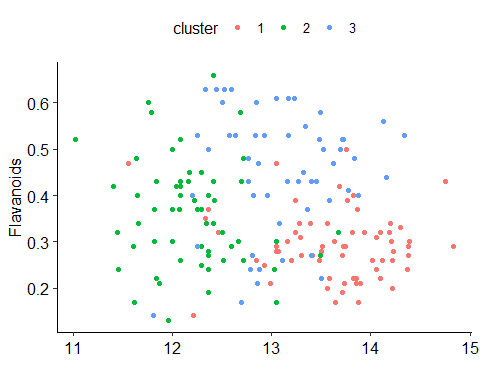
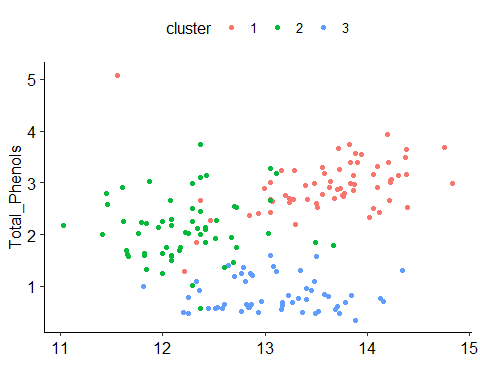
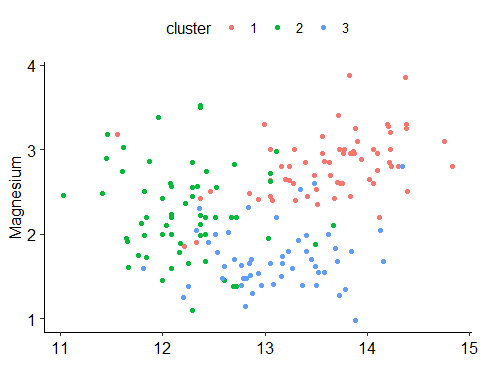
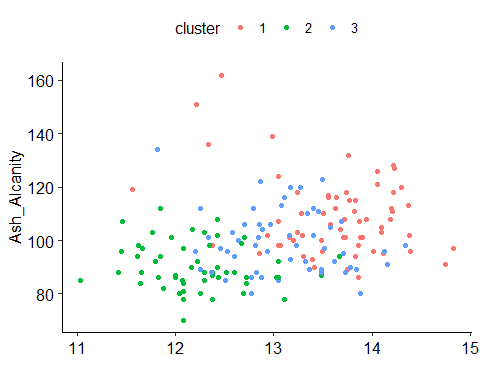
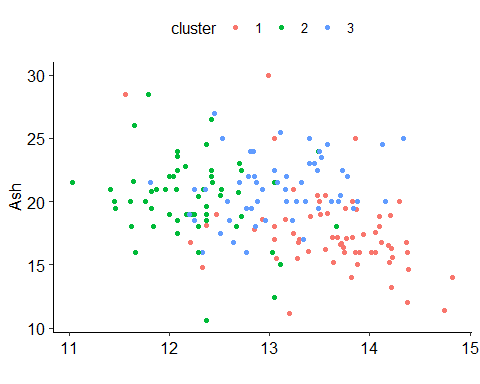
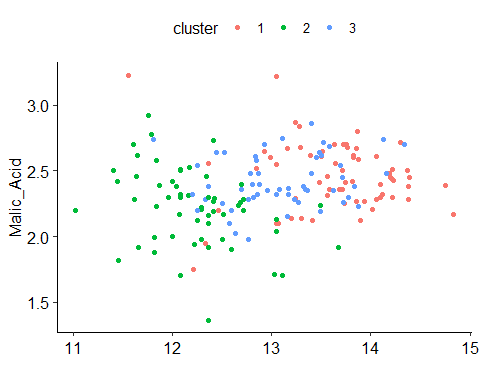
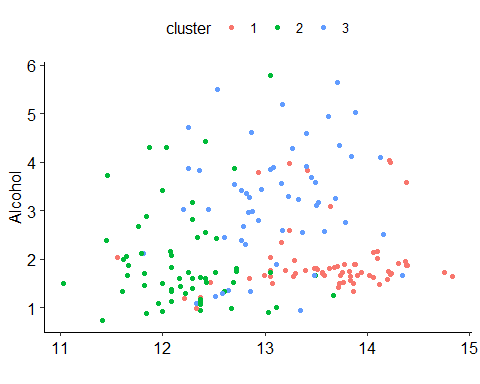


# Visualization of variables and clusters  
library(ggplot2)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

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

colcount <- ncol(mydata)  
xy <- data.frame(cmdscale(dist(mydata)), factor(groups))  
names(xy) <- c("x", "y", "cluster")  
xy$model <- rownames(xy)  
  
for(k in 2:colcount){  
 for(l in k: colcount){  
 plot <- ggplot(xy, aes(mydata[[k-1]], mydata[[l]])) + geom\_point(aes(colour=cluster)) + labs(x = var[[k-1]], y = var[[l]])   
 print(plot)  
 }  
}



## Creating a summary table for clusters  
# Calculating mean, Standard Deviation of Var1 by cluster  
Meanvar1 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_1))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar1 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_1))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var2 by cluster  
Meanvar2 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_2))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar2 <- databind %>%   
group\_by(Cluster) %>%   
 summarise(average = sd(var\_2))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var3 by cluster  
Meanvar3 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_3))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar3 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_3))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var4 by cluster  
Meanvar4 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_4))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar4 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_4))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var5 by cluster  
Meanvar5 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_5))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar5 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_5))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var6 by cluster  
  
Meanvar6 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_6))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar6 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_6))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var7 by cluster  
Meanvar7 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_7))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar7 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_7))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var8 by cluster  
Meanvar8 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_8))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar8 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_8))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var9 by cluster  
Meanvar9 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_9))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar9 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_9))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var10 by cluster  
Meanvar10 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_10))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar10 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_10))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var11 by cluster  
Meanvar11 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_11))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar11 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_11))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var12 by cluster  
Meanvar12 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_12))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar12 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_12))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var13 by cluster  
Meanvar13 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_13))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar13 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_13))

## `summarise()` ungrouping output (override with `.groups` argument)

#clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), stringsAsFactors=FALSE))  
   
if (m == 1){clustercount = t(data.frame(sum(databind$Cluster == 1), stringsAsFactors=FALSE))}  
  
if (m == 1){clustercount = t(data.frame(sum(databind$Cluster == 1), stringsAsFactors=FALSE))}  
  
if (m == 2) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), stringsAsFactors=FALSE))}  
  
if (m == 3) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), stringsAsFactors=FALSE))}  
  
if (m == 4) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), stringsAsFactors=FALSE))}  
  
if (m == 5) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), stringsAsFactors=FALSE))}  
  
if (m == 6) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), stringsAsFactors=FALSE))}  
  
if (m == 7) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), stringsAsFactors=FALSE))}  
  
if (m == 8) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), stringsAsFactors=FALSE))}  
  
if (m == 9) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), sum(databind$Cluster == 9), stringsAsFactors=FALSE))}  
  
if (m == 10) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), sum(databind$Cluster == 9), sum(databind$Cluster == 10), stringsAsFactors=FALSE))}  
  
  
# Combining data frames  
var1table <- merge(Meanvar1, SDvar1, by = "Cluster", sort = TRUE)  
colnames(var1table) <- c("Cluster","Mean1", "SD1")  
  
var2table <- merge(Meanvar2, SDvar2, by = "Cluster", sort = TRUE)  
colnames(var2table) <- c("Cluster","Mean2", "SD2")  
  
var3table <- merge(Meanvar3, SDvar3, by = "Cluster", sort = TRUE)  
colnames(var3table) <- c("Cluster","Mean3", "SD3")  
  
var4table <- merge(Meanvar4, SDvar4, by = "Cluster", sort = TRUE)  
colnames(var4table) <- c("Cluster","Mean4", "SD4")  
  
var5table <- merge(Meanvar5, SDvar5, by = "Cluster", sort = TRUE)  
colnames(var5table) <- c("Cluster","Mean5", "SD5")  
  
var6table <- merge(Meanvar6, SDvar6, by = "Cluster", sort = TRUE)  
colnames(var6table) <- c("Cluster","Mean6", "SD6")   
   
var7table <- merge(Meanvar7, SDvar7, by = "Cluster", sort = TRUE)  
colnames(var7table) <- c("Cluster","Mean7", "SD7")  
   
var8table <- merge(Meanvar8, SDvar8, by = "Cluster", sort = TRUE)  
colnames(var8table) <- c("Cluster","Mean8", "SD8")   
   
var9table <- merge(Meanvar9, SDvar9, by = "Cluster", sort = TRUE)  
colnames(var9table) <- c("Cluster","Mean9", "SD9")  
   
var10table <- merge(Meanvar10, SDvar10, by = "Cluster", sort = TRUE)  
colnames(var10table) <- c("Cluster","Mean10", "SD10")  
   
var11table <- merge(Meanvar11, SDvar11, by = "Cluster", sort = TRUE)  
colnames(var11table) <- c("Cluster","Mean11", "SD11")  
   
var12table <- merge(Meanvar12, SDvar12, by = "Cluster", sort = TRUE)  
colnames(var12table) <- c("Cluster","Mean12", "SD12")   
   
var13table <- merge(Meanvar13, SDvar13, by = "Cluster", sort = TRUE)  
colnames(var13table) <- c("Cluster","Mean13", "SD13")   
   
   
table <- var1table %>% merge(var2table, by = "Cluster", sort = TRUE) %>% merge(var3table, by = "Cluster", sort = TRUE) %>%  
 merge(var4table, by = "Cluster", sort = TRUE) %>% merge(var5table, by = "Cluster", sort = TRUE) %>% merge(var6table, by = "Cluster", sort = TRUE) %>%  
 merge(var7table, by = "Cluster", sort = TRUE) %>% merge(var8table, by = "Cluster", sort = TRUE) %>% merge(var9table, by = "Cluster", sort = TRUE) %>%  
 merge(var10table, by = "Cluster", sort = TRUE) %>% merge(var11table, by = "Cluster", sort = TRUE)%>% merge(var12table, by = "Cluster", sort = TRUE) %>%  
 merge(var13table, by = "Cluster", sort = TRUE)  
  
table <- cbind(table, "Elements in Cluster" =clustercount[, 1])  
  
rownames(table) <- c(clust1name,clust2name,clust3name)  
  
table <- table[c(0,1, 28, 2, 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27)] #Change the order of columns  
  
labels <- data.frame( "", "", var1, var1, var2, var2, var3, var3, var4, var4, var5, var5, var6,var6, var7, var7, var8, var8, var9,var9, var10,var10, var11,  
 var11, var12, var12, var13, var13)  
  
colnames(labels) <- c("Cluster","Elements in Cluster", "Mean1", "SD1", "Mean2", "SD2", "Mean3", "SD3", "Mean4", "SD4", "Mean5", "SD5", "Mean6", "SD6", "Mean7", "SD7", "Mean8", "SD8", "Mean9", "SD9", "Mean10", "SD10", "Mean11", "SD11", "Mean12", "SD12", "Mean13", "SD13")  
table <- rbind(table, labels)  
rownames(table) <- c(clust1name,clust2name,clust3name, "")  
table

## Cluster Elements in Cluster Mean1 SD1  
## Top 1 65 13.6258461538462 0.615081021867476  
## Mid 2 59 12.2222033898305 0.504705676596084  
## Lower 3 54 13.0985185185185 0.549278797846596  
## Alcohol Alcohol  
## Mean2 SD2 Mean3 SD3  
## Top 1.95446153846154 0.686692777851818 2.45846153846154 0.259688700473945  
## Mid 1.96610169491525 1.03206524945616 2.21627118644068 0.289576970323387  
## Lower 3.20055555555556 1.15777179728615 2.42 0.199357458416823  
## Malic\_Acid Malic\_Acid Ash Ash  
## Mean4 SD4 Mean5 SD5  
## Top 17.6153846153846 3.3206376820401 108.815384615385 14.5085021492704  
## Mid 20.1237288135593 3.07230897013119 89.8983050847458 8.40298574862087  
## Lower 21.0703703703704 2.50987706454032 99.5740740740741 11.8713256112363  
## Ash\_Alcanity Ash\_Alcanity Magnesium Magnesium  
## Mean6 SD6 Mean7 SD7  
## Top 2.81461538461538 0.377131761304475 2.94723076923077 0.533603632426225  
## Mid 2.26305084745763 0.53925636612324 2.11016949152542 0.59813644944664  
## Lower 1.70481481481481 0.361880522036224 0.835925925925926 0.324022585507458  
## Total\_Phenols Total\_Phenols Flavanoids Flavanoids  
## Mean8 SD8 Mean9  
## Top 0.292461538461538 0.0737824244237485 1.93030769230769  
## Mid 0.356440677966102 0.113786895673969 1.64203389830508  
## Lower 0.451296296296296 0.129978899540065 1.12648148148148  
## Nonflavanoid\_Phenols Nonflavanoid\_Phenols Proanthocyanins  
## SD9 Mean10 SD10 Mean11  
## Top 0.429614541008744 5.31 1.33349751593319 1.07430769230769  
## Mid 0.5446187520073 3.00474576271186 0.970000632655683 1.05230508474576  
## Lower 0.427353229389298 6.99833331481481 2.48289295735297 0.713148148148148  
## Proanthocyanins Color\_Intensity Color\_Intensity Hue  
## SD11 Mean12 SD12 Mean13  
## Top 0.125933628734162 3.13938461538462 0.384321142255556 1071.07692307692  
## Mid 0.203631788575672 2.85474576271186 0.420013776178844 503.389830508475  
## Lower 0.149588417396359 1.71092592592593 0.279441954134054 622.722222222222  
## Hue OD280 OD280 Proline  
## SD13  
## Top 255.240714062989  
## Mid 159.31177597329  
## Lower 118.868790303326  
## Proline

## ANOVA table  
# Degrees of freedom  
countobs <- count(mydata)  
betw.grpsdf <- m - 1  
within.grpsdf <- countobs - m  
totdf <- countobs-1

## Sum of Squares between groups, var1  
clusterpath <- databind$Cluster #set cluster path  
  
varpath <- var1path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar1[1,2])   
MeanTwo <-as.integer(Meanvar1[2,2])   
MeanThree <- as.integer(Meanvar1[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var1ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var1ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var1ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var2  
varpath <- var2path # set variable path  
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar2[1,2])   
MeanTwo <-as.integer(Meanvar2[2,2])   
MeanThree <- as.integer(Meanvar2[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var2ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var2ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var2ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var3  
varpath <- var3path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar3[1,2])   
MeanTwo <-as.integer(Meanvar3[2,2])   
MeanThree <- as.integer(Meanvar3[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var3ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var3ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var3ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var4  
varpath <- var4path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar4[1,2])   
MeanTwo <-as.integer(Meanvar4[2,2])   
MeanThree <- as.integer(Meanvar4[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var4ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var4ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var4ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var5path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar5[1,2])   
MeanTwo <-as.integer(Meanvar5[2,2])   
MeanThree <- as.integer(Meanvar5[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var5ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var5ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var5ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var6path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar6[1,2])   
MeanTwo <-as.integer(Meanvar6[2,2])   
MeanThree <- as.integer(Meanvar6[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var6ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var6ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var6ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var7path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar7[1,2])   
MeanTwo <-as.integer(Meanvar7[2,2])   
MeanThree <- as.integer(Meanvar7[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var7ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var7ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var7ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var8path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar8[1,2])   
MeanTwo <-as.integer(Meanvar8[2,2])   
MeanThree <- as.integer(Meanvar8[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var8ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var8ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var8ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var9path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar9[1,2])   
MeanTwo <-as.integer(Meanvar9[2,2])   
MeanThree <- as.integer(Meanvar9[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var9ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var9ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var9ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var10  
varpath <- var10path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar10[1,2])   
MeanTwo <-as.integer(Meanvar10[2,2])   
MeanThree <- as.integer(Meanvar10[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var10ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var10ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var10ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var11  
varpath <- var11path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar11[1,2])   
MeanTwo <-as.integer(Meanvar11[2,2])   
MeanThree <- as.integer(Meanvar11[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var11ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var11ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var11ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var12  
varpath <- var12path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar12[1,2])   
MeanTwo <-as.integer(Meanvar12[2,2])   
MeanThree <- as.integer(Meanvar12[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var12ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var12ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var12ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var13path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar13[1,2])   
MeanTwo <-as.integer(Meanvar13[2,2])   
MeanThree <- as.integer(Meanvar13[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var13ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var13ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var13ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

labels <- t(data.frame(var1, "", "", var2, "", "", var3, "", "", var4, "", "", var5, "", "", var6, "", "", var7, "", "", var8, "", "", var9, "", "", var10, "", "", var11, "", "", var12, "", "", var13, "", ""))  
ANOVAtable <- rbind(var1ANOVA, var2ANOVA, var3ANOVA, var4ANOVA, var5ANOVA, var6ANOVA,var7ANOVA, var8ANOVA, var9ANOVA, var10ANOVA, var11ANOVA, var12ANOVA, var13ANOVA)  
cbind(ANOVAtable, labels)

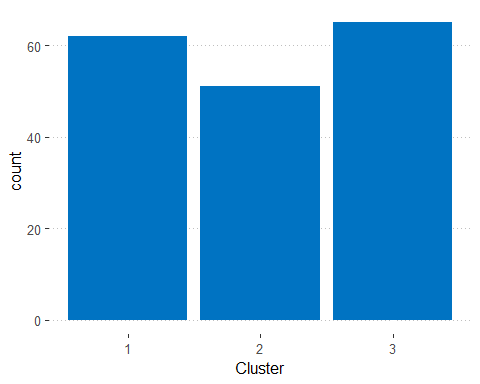
## Sum of Squares DF Mean Square F-Statistic  
## Between Values 1.006424 2 0.503212 1.049932   
## Within Values 83.8741 175 0.4792806 ""   
## Total 84.88052 177 "" ""   
## Between Values 1.029774 2 0.5148871 0.322431   
## Within Values 279.4559 175 1.596891 ""   
## Total 280.4857 177 "" ""   
## Between Values 0.03388803 2 0.01694402 0.07963825   
## Within Values 37.2334 175 0.2127623 ""   
## Total 37.26729 177 "" ""   
## Between Values 6.410082 2 3.205041 0.3477627   
## Within Values 1612.83 175 9.216171 ""   
## Total 1619.24 177 "" ""   
## Between Values 179.252 2 89.62602 0.6237643   
## Within Values 25145 175 143.6857 ""   
## Total 25324.25 177 "" ""   
## Between Values 0.6193625 2 0.3096813 0.5067187   
## Within Values 106.9513 175 0.6111503 ""   
## Total 107.5707 177 "" ""   
## Between Values 2.273267 2 1.136633 1.407633   
## Within Values 141.3087 175 0.8074783 ""   
## Total 143.582 177 "" ""   
## Between Values 0.01284454 2 0.006422272 0.04314635   
## Within Values 26.0485 175 0.1488486 ""   
## Total 26.06134 177 "" ""   
## Between Values 0.3334967 2 0.1667483 0.2429014   
## Within Values 120.135 175 0.6864857 ""   
## Total 120.4685 177 "" ""   
## Between Values 8.044225 2 4.022113 1.267827   
## Within Values 555.1782 175 3.172447 ""   
## Total 563.2224 177 "" ""   
## Between Values 0.08233657 2 0.04116828 0.2210657   
## Within Values 32.58964 175 0.1862265 ""   
## Total 32.67197 177 "" ""   
## Between Values 1.148912 2 0.5744562 1.052849   
## Within Values 95.4836 175 0.5456206 ""   
## Total 96.63251 177 "" ""   
## Between Values 179807.4 2 89903.71 2.461986   
## Within Values 6390431 175 36516.75 ""   
## Total 6570238 177 "" ""   
##   
## Between Values "Alcohol"   
## Within Values ""   
## Total ""   
## Between Values "Malic\_Acid"   
## Within Values ""   
## Total ""   
## Between Values "Ash"   
## Within Values ""   
## Total ""   
## Between Values "Ash\_Alcanity"   
## Within Values ""   
## Total ""   
## Between Values "Magnesium"   
## Within Values ""   
## Total ""   
## Between Values "Total\_Phenols"   
## Within Values ""   
## Total ""   
## Between Values "Flavanoids"   
## Within Values ""   
## Total ""   
## Between Values "Nonflavanoid\_Phenols"  
## Within Values ""   
## Total ""   
## Between Values "Proanthocyanins"   
## Within Values ""   
## Total ""   
## Between Values "Color\_Intensity"   
## Within Values ""   
## Total ""   
## Between Values "Hue"   
## Within Values ""   
## Total ""   
## Between Values "OD280"   
## Within Values ""   
## Total ""   
## Between Values "Proline"   
## Within Values ""   
## Total ""

### K-means Clustering  
result.kmean = kmeans(mydatastandardized, m, nstart = 50, iter.max = 10)

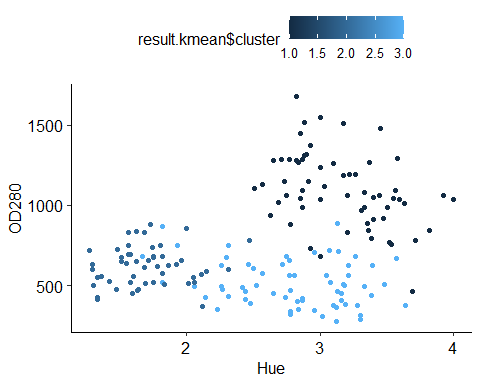
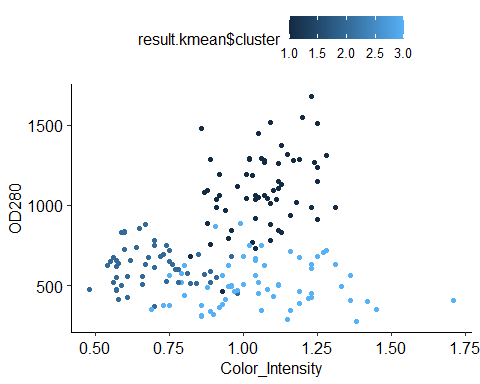
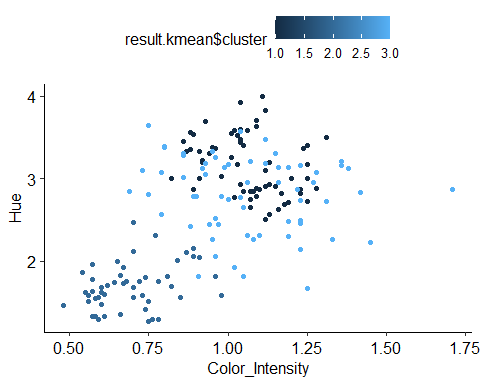
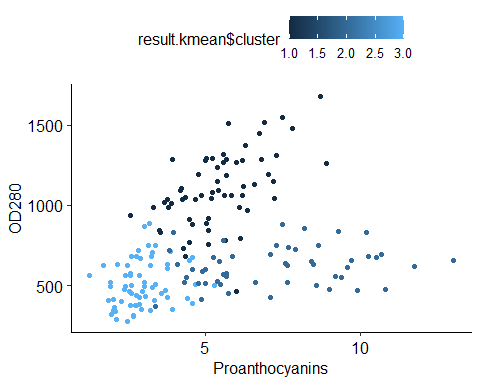
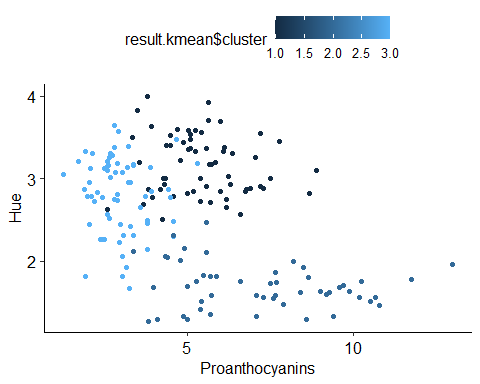
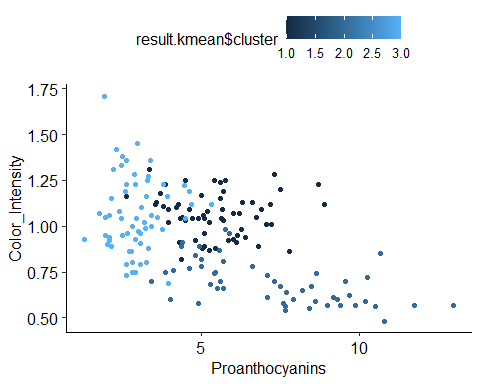
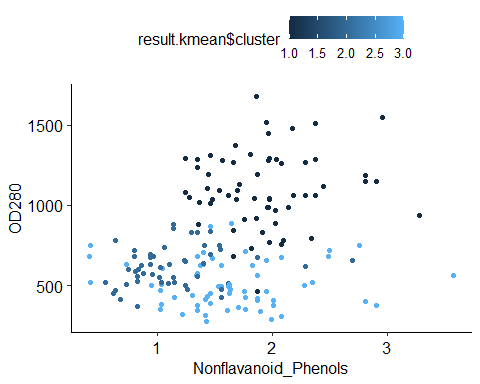
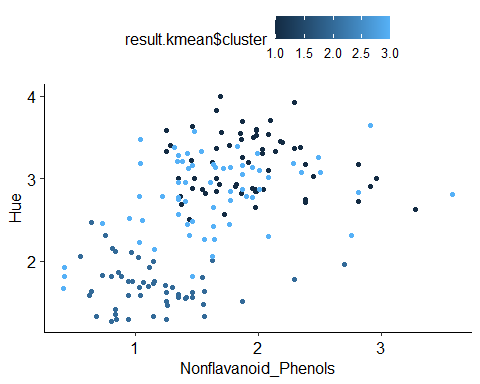
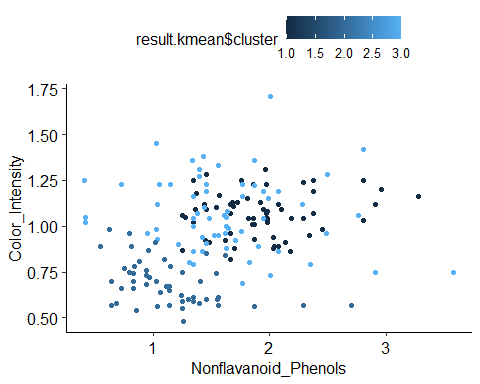
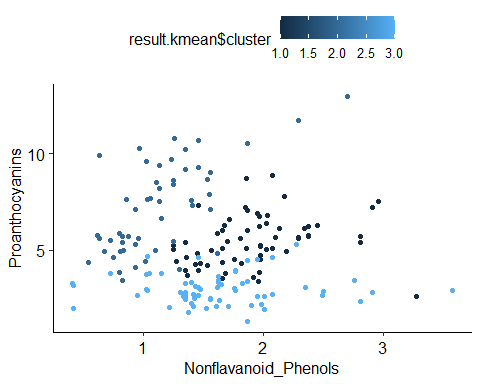
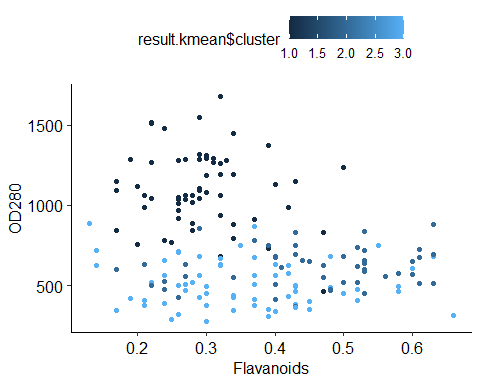
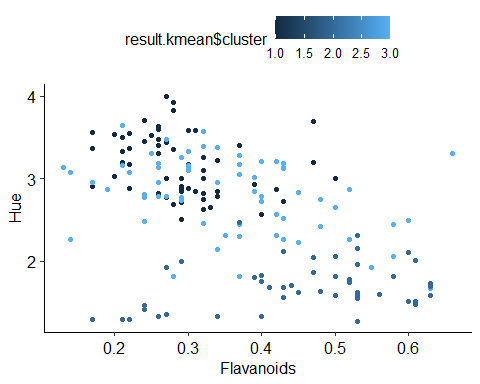
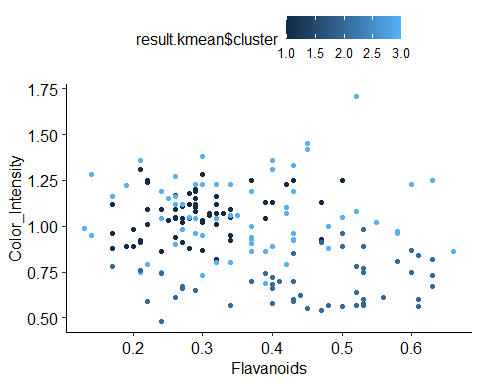
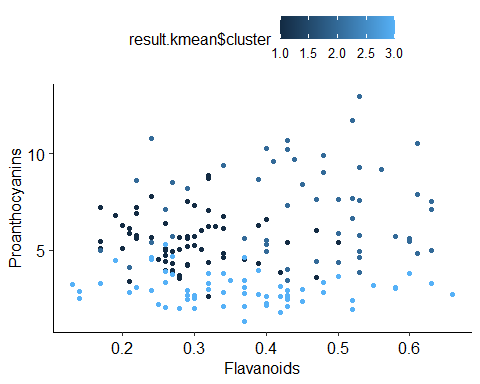
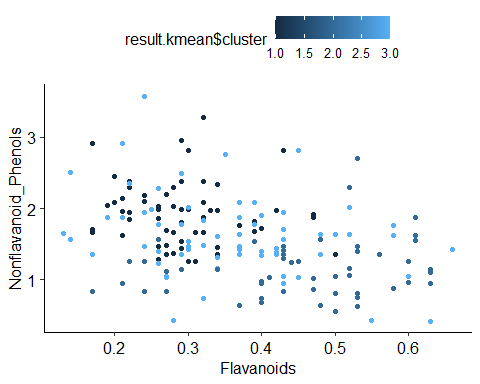
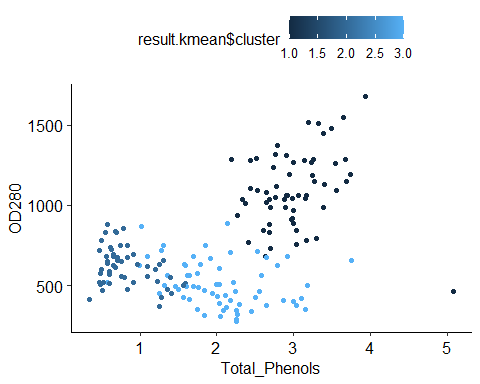
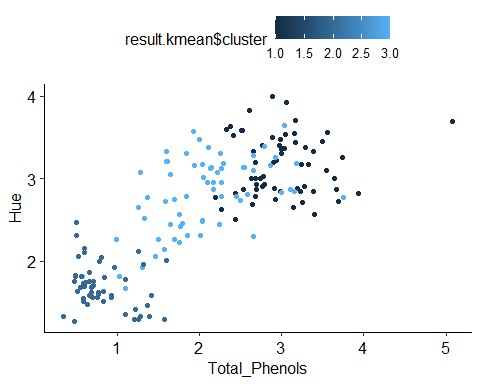
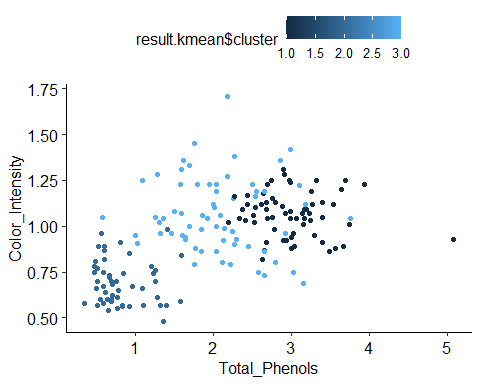
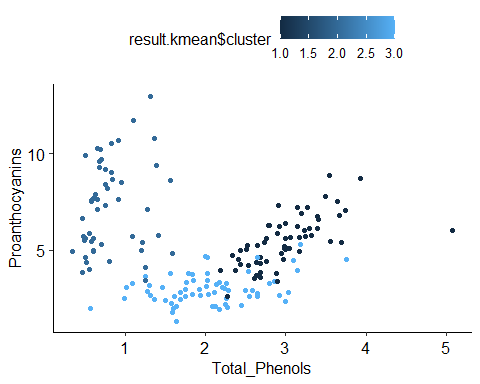
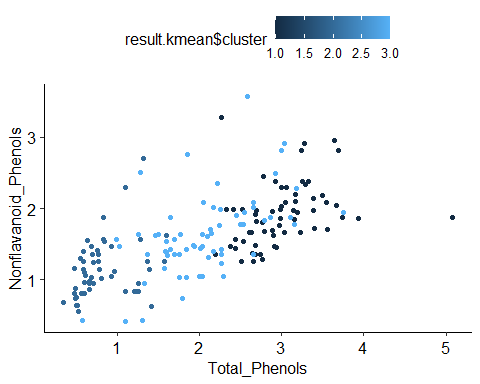
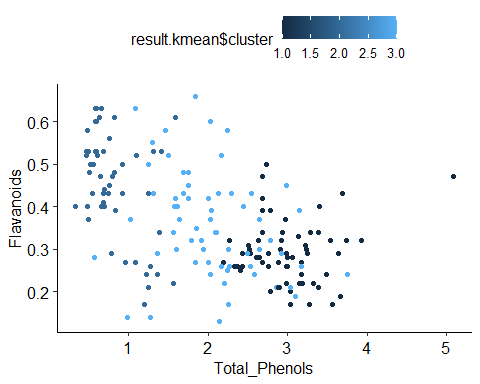
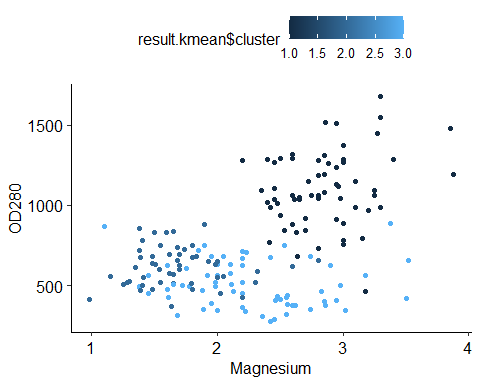
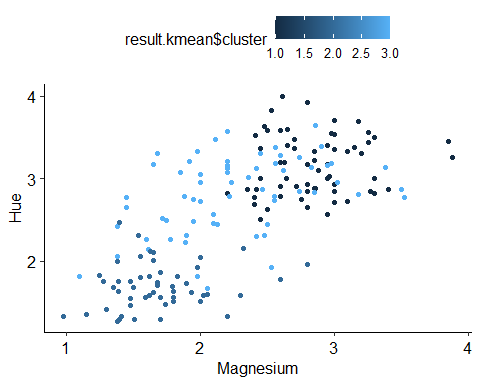
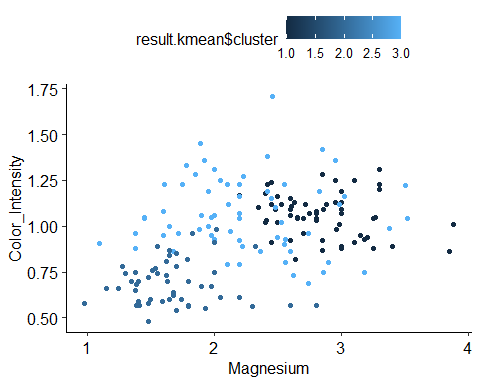
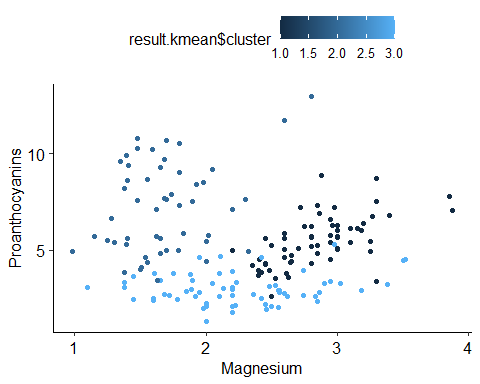
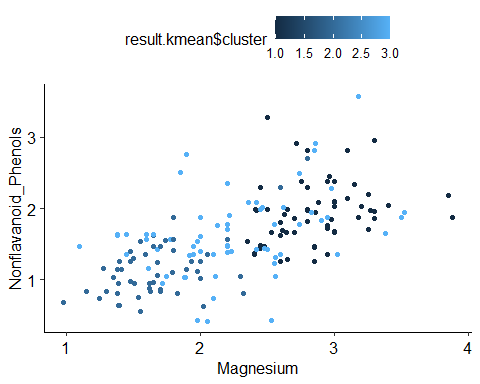
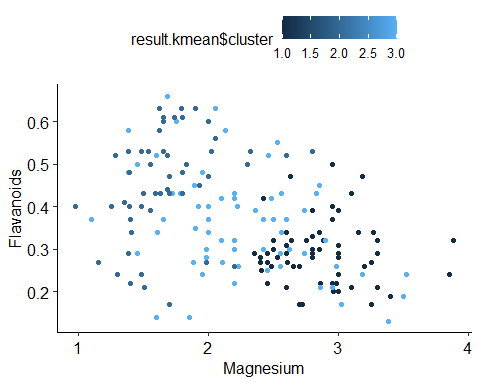
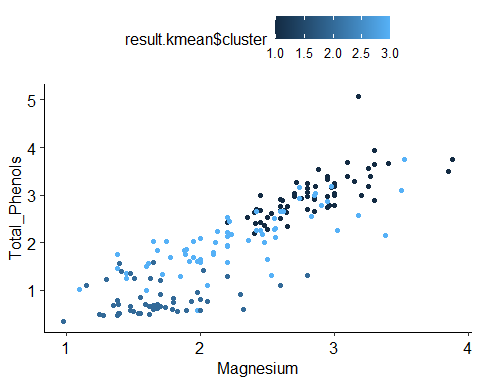
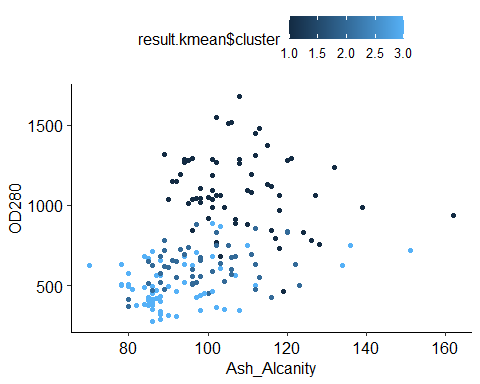
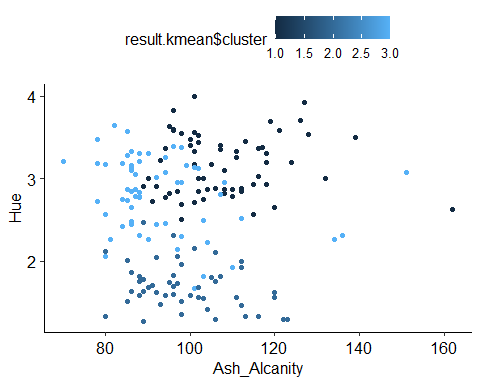
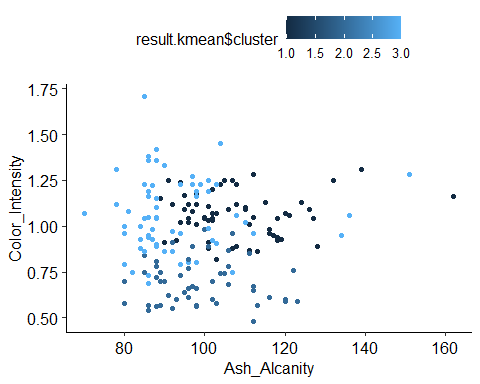
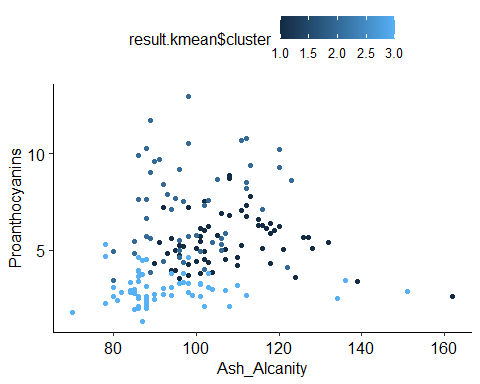
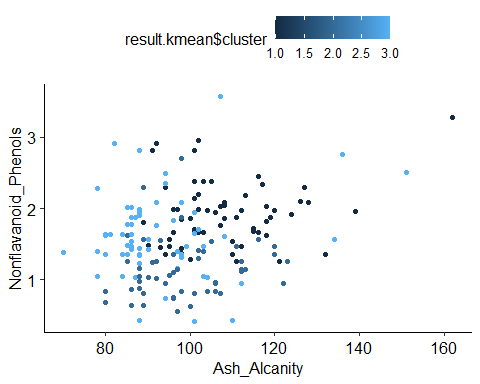
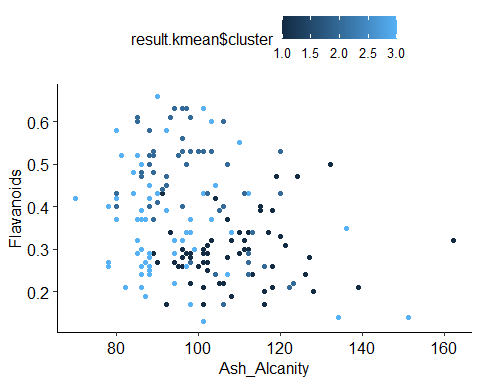
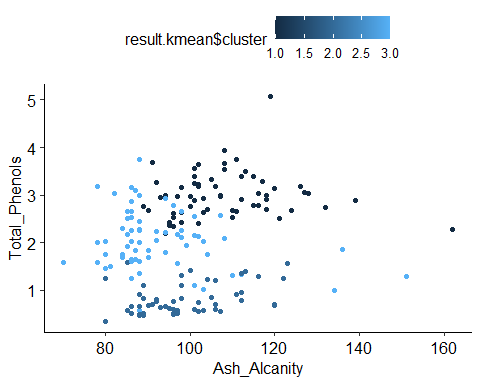
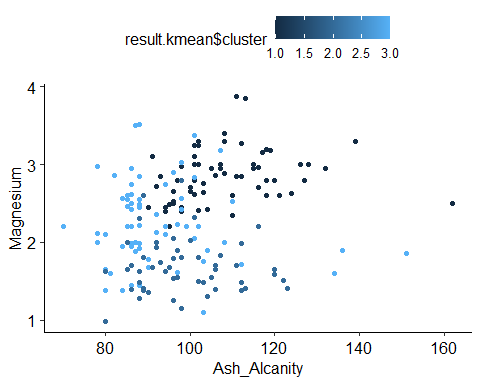
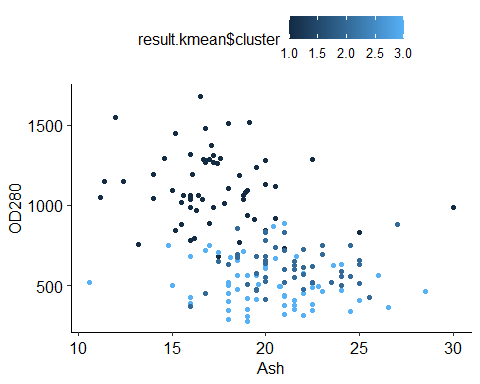
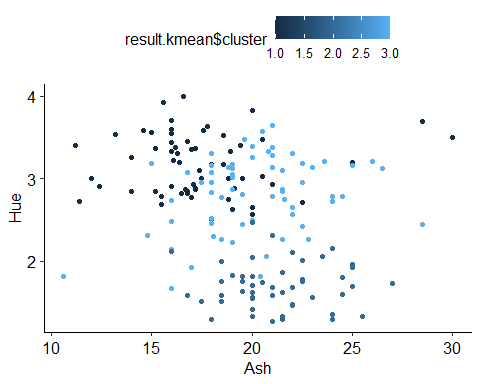
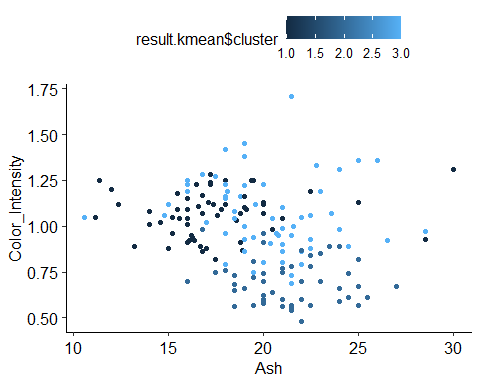
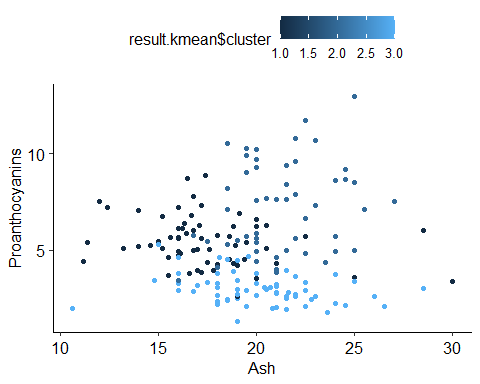
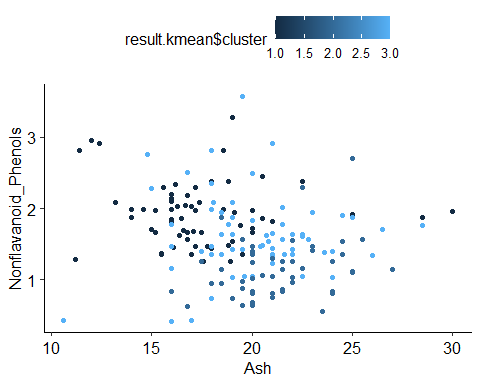
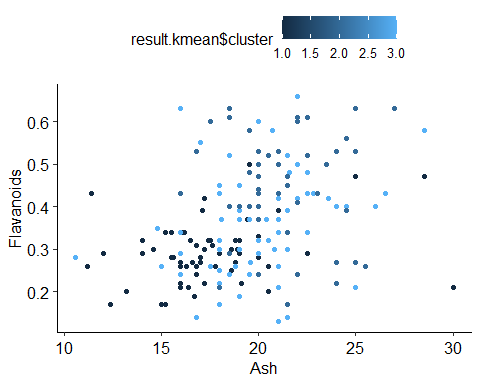
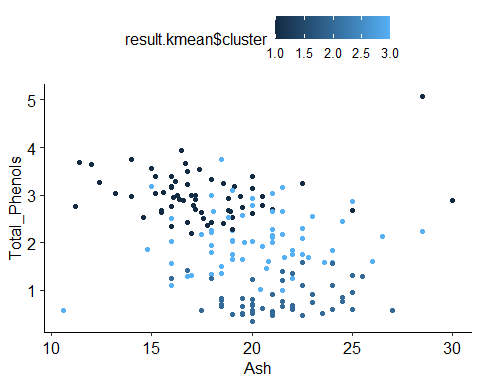
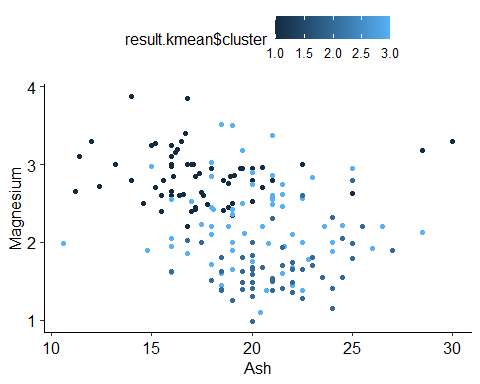
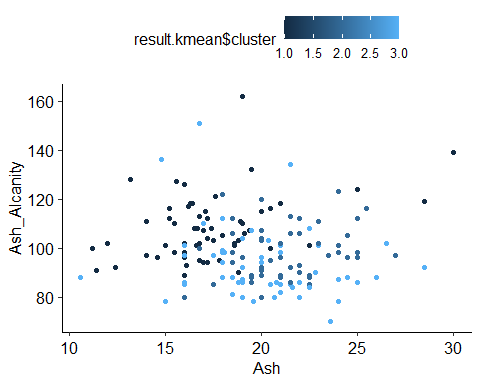
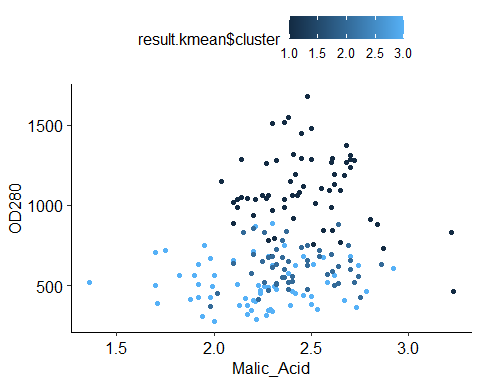
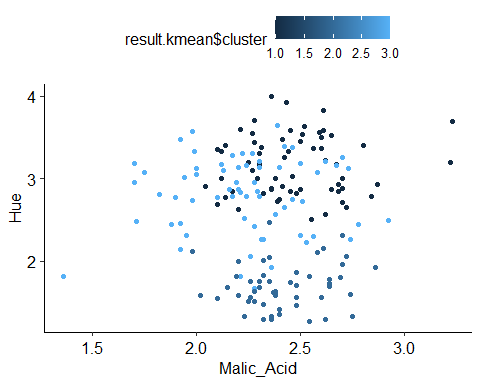
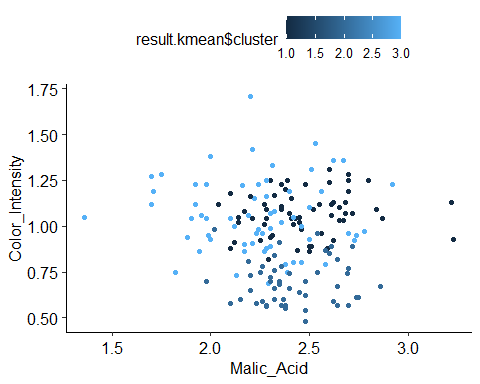
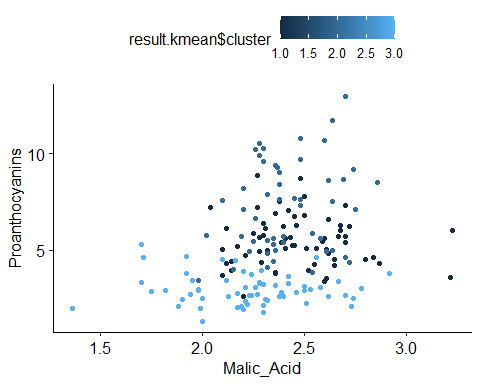
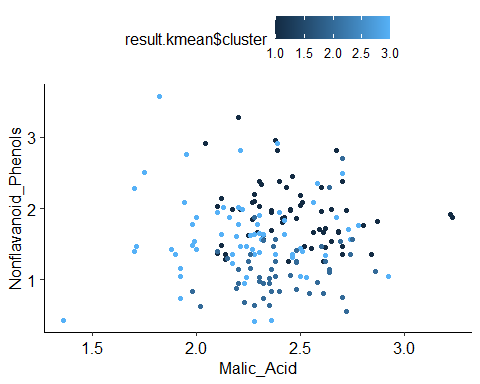
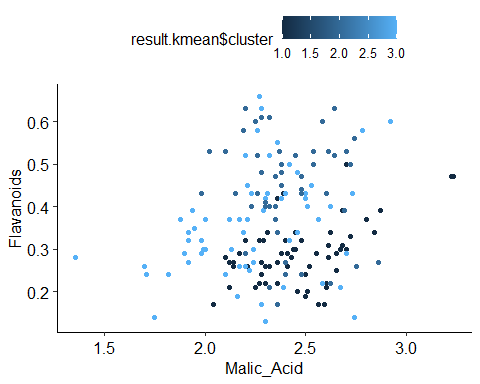
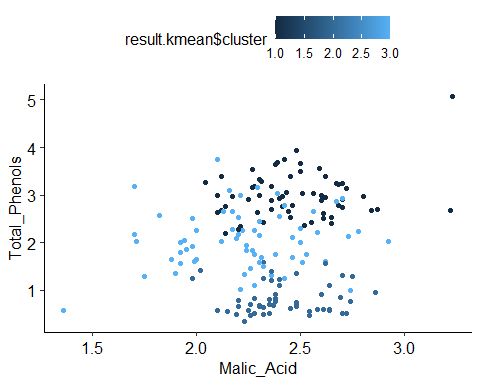
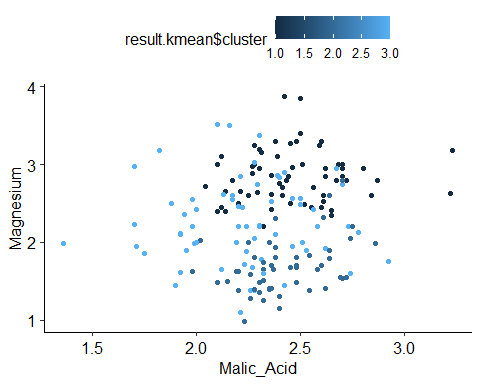
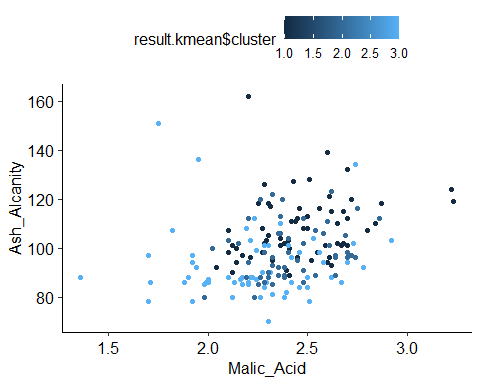
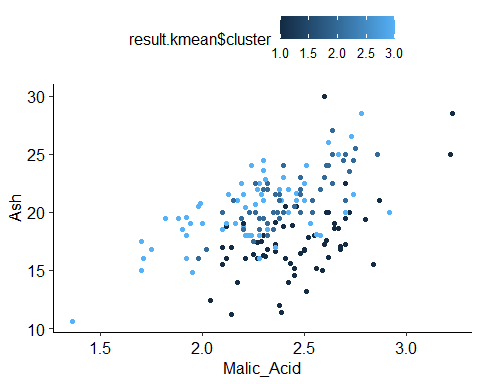
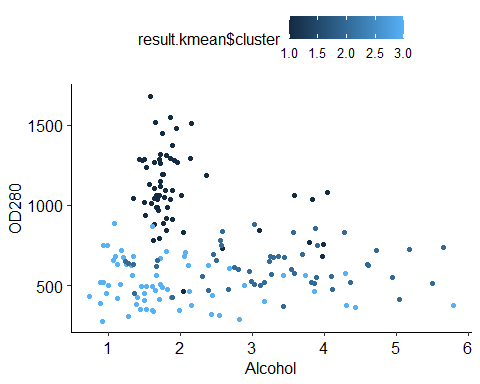
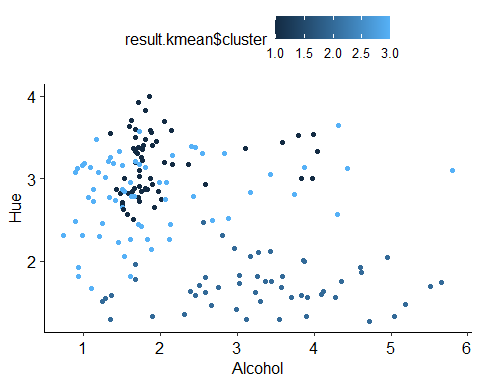
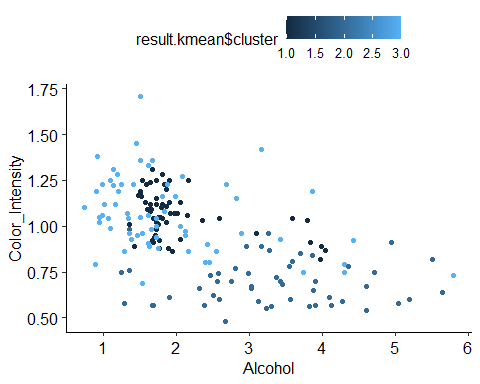
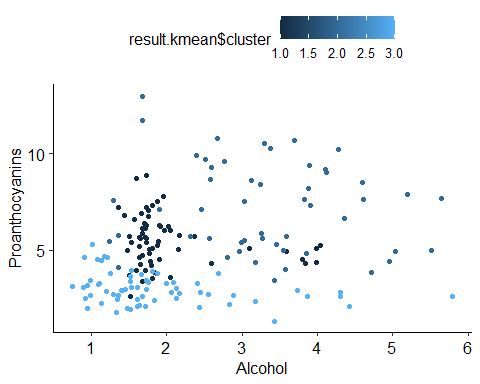
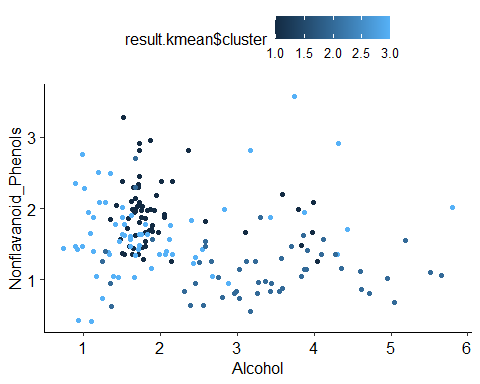
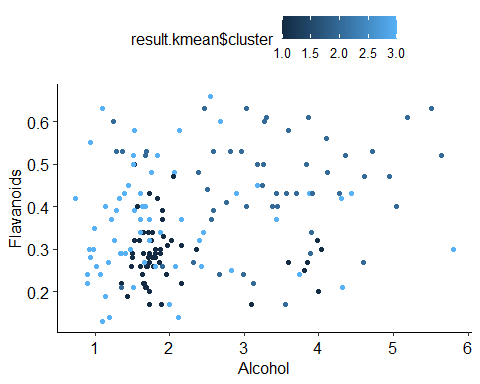
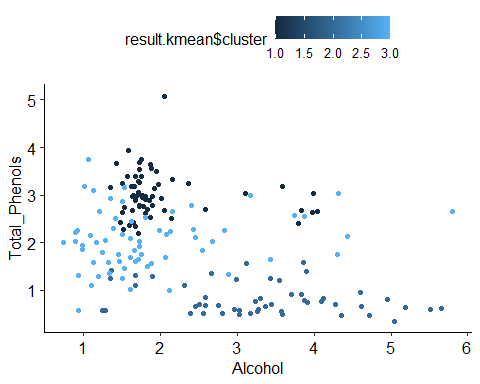
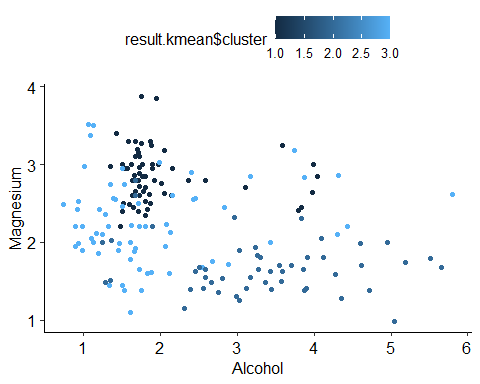
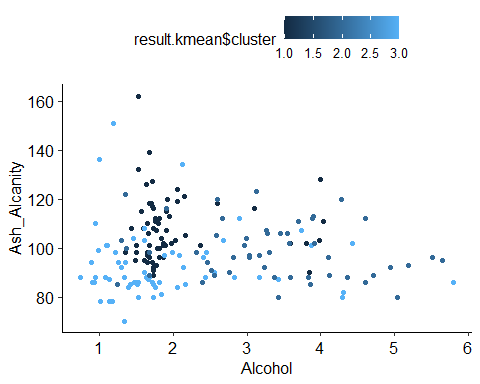
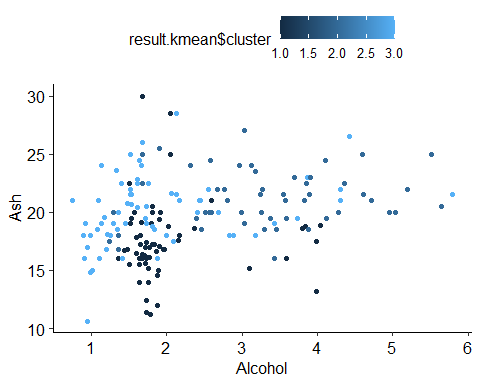
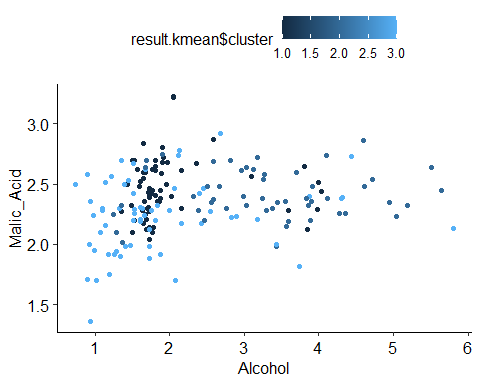
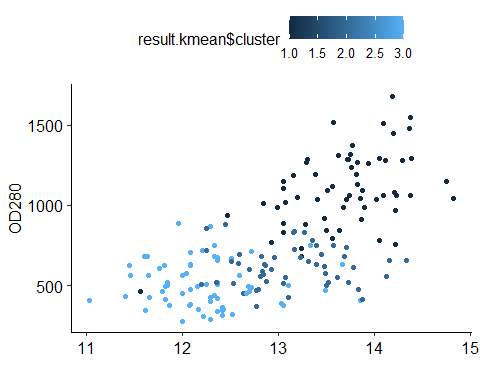
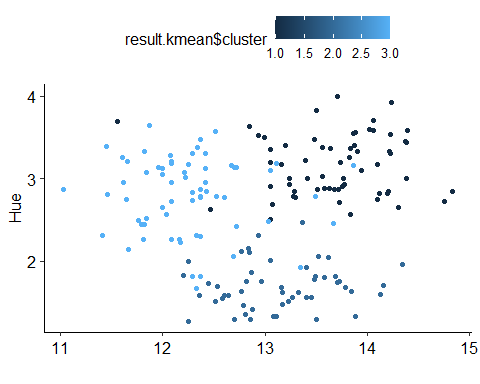
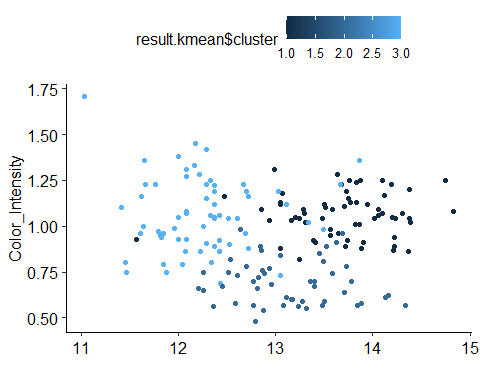
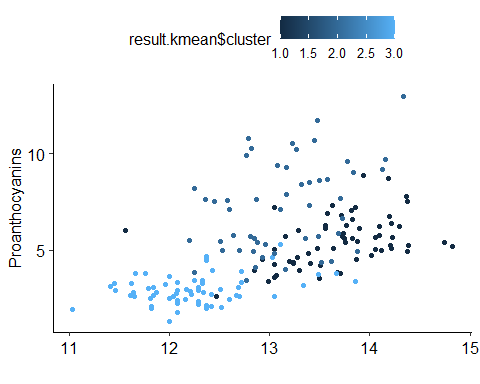
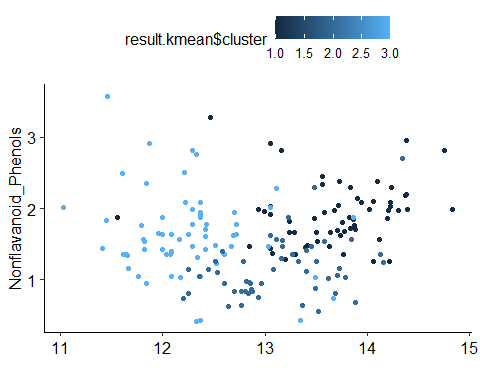
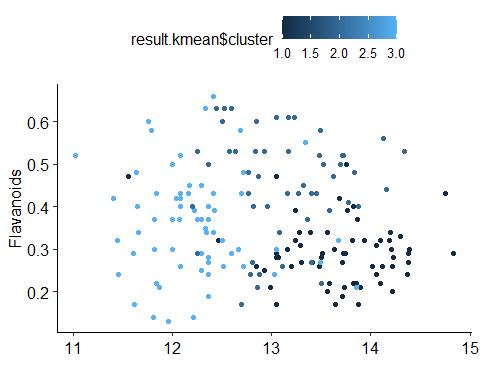
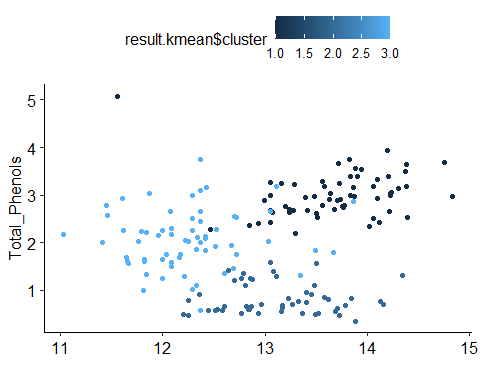
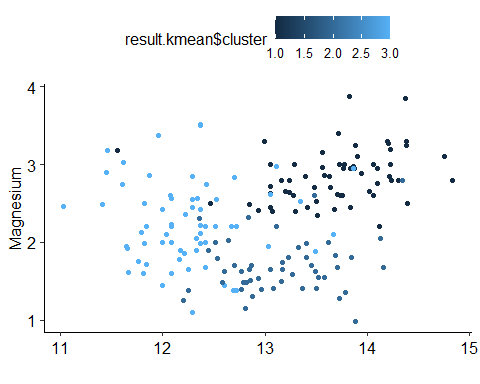
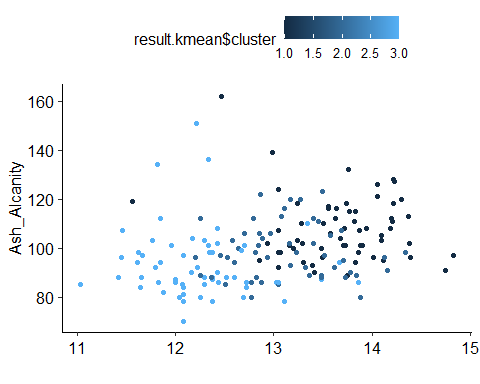
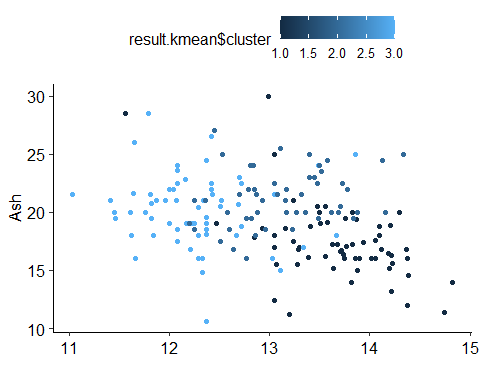
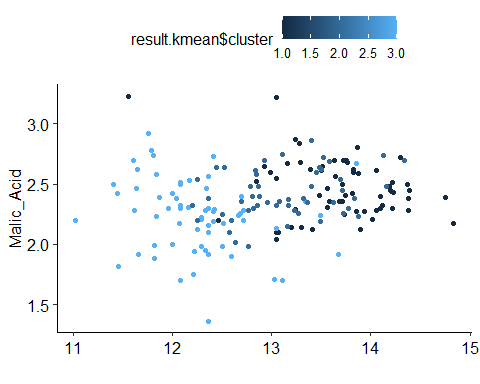
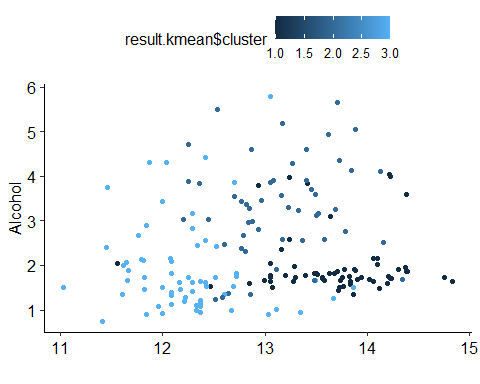
# Bind data with respective clusters  
databind <- cbind(mydata, Cluster = result.kmean$cluster)  
head(databind)

## var\_1 var\_2 var\_3 var\_4 var\_5 var\_6 var\_7 var\_8 var\_9 var\_10 var\_11 var\_12  
## 1 14.23 1.71 2.43 15.6 127 2.80 3.06 0.28 2.29 5.64 1.04 3.92  
## 2 13.20 1.78 2.14 11.2 100 2.65 2.76 0.26 1.28 4.38 1.05 3.40  
## 3 13.16 2.36 2.67 18.6 101 2.80 3.24 0.30 2.81 5.68 1.03 3.17  
## 4 14.37 1.95 2.50 16.8 113 3.85 3.49 0.24 2.18 7.80 0.86 3.45  
## 5 13.24 2.59 2.87 21.0 118 2.80 2.69 0.39 1.82 4.32 1.04 2.93  
## 6 14.20 1.76 2.45 15.2 112 3.27 3.39 0.34 1.97 6.75 1.05 2.85  
## var\_13 Cluster  
## 1 1065 1  
## 2 1050 1  
## 3 1185 1  
## 4 1480 1  
## 5 735 1  
## 6 1450 1

# Histogram  
library(ggplot2)  
library(ggpubr)  
theme\_set(theme\_pubr())  
ggplot(databind, aes(Cluster)) +  
 geom\_bar(fill = "#0073C2FF") +  
 theme\_pubclean()



for(k in 2:colcount){  
 for(l in k: colcount){  
   
 plot <- ggplot(xy, aes(mydata[[k-1]], mydata[[l]])) + geom\_point(aes(colour=result.kmean$cluster)) + labs(x = var[[k-1]], y = var[[l]])   
 print(plot)  
 }  
}



## Creating a summary table for clusters  
# Calculating mean, Standard Deviation of Var1 by cluster  
Meanvar1 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_1))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar1 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_1))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var2 by cluster  
Meanvar2 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_2))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar2 <- databind %>%   
group\_by(Cluster) %>%   
 summarise(average = sd(var\_2))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var3 by cluster  
Meanvar3 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_3))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar3 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_3))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var4 by cluster  
Meanvar4 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_4))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar4 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_4))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var5 by cluster  
Meanvar5 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_5))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar5 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_5))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var6 by cluster  
  
Meanvar6 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_6))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar6 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_6))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var7 by cluster  
Meanvar7 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_7))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar7 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_7))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var8 by cluster  
Meanvar8 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_8))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar8 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_8))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var9 by cluster  
Meanvar9 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_9))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar9 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_9))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var10 by cluster  
Meanvar10 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_10))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar10 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_10))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var11 by cluster  
Meanvar11 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_11))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar11 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_11))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var12 by cluster  
Meanvar12 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_12))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar12 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_12))

## `summarise()` ungrouping output (override with `.groups` argument)

# Calculating mean, Standard Deviation of Var13 by cluster  
Meanvar13 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = mean(var\_13))

## `summarise()` ungrouping output (override with `.groups` argument)

SDvar13 <- databind %>%   
 group\_by(Cluster) %>%   
 summarise(average = sd(var\_13))

## `summarise()` ungrouping output (override with `.groups` argument)

#clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), stringsAsFactors=FALSE))  
   
if (m == 1){clustercount = t(data.frame(sum(databind$Cluster == 1), stringsAsFactors=FALSE))}  
  
if (m == 1){clustercount = t(data.frame(sum(databind$Cluster == 1), stringsAsFactors=FALSE))}  
  
if (m == 2) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), stringsAsFactors=FALSE))}  
  
if (m == 3) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), stringsAsFactors=FALSE))}  
  
if (m == 4) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), stringsAsFactors=FALSE))}  
  
if (m == 5) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), stringsAsFactors=FALSE))}  
  
if (m == 6) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), stringsAsFactors=FALSE))}  
  
if (m == 7) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), stringsAsFactors=FALSE))}  
  
if (m == 8) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), stringsAsFactors=FALSE))}  
  
if (m == 9) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), sum(databind$Cluster == 9), stringsAsFactors=FALSE))}  
  
if (m == 10) {clustercount <- t(data.frame(sum(databind$Cluster == 1), sum(databind$Cluster == 2), sum(databind$Cluster == 3), sum(databind$Cluster == 4), sum(databind$Cluster == 5), sum(databind$Cluster == 6), sum(databind$Cluster == 7), sum(databind$Cluster == 8), sum(databind$Cluster == 9), sum(databind$Cluster == 10), stringsAsFactors=FALSE))}  
  
  
# Combining data frames  
var1table <- merge(Meanvar1, SDvar1, by = "Cluster", sort = TRUE)  
colnames(var1table) <- c("Cluster","Mean1", "SD1")  
  
var2table <- merge(Meanvar2, SDvar2, by = "Cluster", sort = TRUE)  
colnames(var2table) <- c("Cluster","Mean2", "SD2")  
  
var3table <- merge(Meanvar3, SDvar3, by = "Cluster", sort = TRUE)  
colnames(var3table) <- c("Cluster","Mean3", "SD3")  
  
var4table <- merge(Meanvar4, SDvar4, by = "Cluster", sort = TRUE)  
colnames(var4table) <- c("Cluster","Mean4", "SD4")  
  
var5table <- merge(Meanvar5, SDvar5, by = "Cluster", sort = TRUE)  
colnames(var5table) <- c("Cluster","Mean5", "SD5")  
  
var6table <- merge(Meanvar6, SDvar6, by = "Cluster", sort = TRUE)  
colnames(var6table) <- c("Cluster","Mean6", "SD6")   
   
var7table <- merge(Meanvar7, SDvar7, by = "Cluster", sort = TRUE)  
colnames(var7table) <- c("Cluster","Mean7", "SD7")  
   
var8table <- merge(Meanvar8, SDvar8, by = "Cluster", sort = TRUE)  
colnames(var8table) <- c("Cluster","Mean8", "SD8")   
   
var9table <- merge(Meanvar9, SDvar9, by = "Cluster", sort = TRUE)  
colnames(var9table) <- c("Cluster","Mean9", "SD9")  
   
var10table <- merge(Meanvar10, SDvar10, by = "Cluster", sort = TRUE)  
colnames(var10table) <- c("Cluster","Mean10", "SD10")  
   
var11table <- merge(Meanvar11, SDvar11, by = "Cluster", sort = TRUE)  
colnames(var11table) <- c("Cluster","Mean11", "SD11")  
   
var12table <- merge(Meanvar12, SDvar12, by = "Cluster", sort = TRUE)  
colnames(var12table) <- c("Cluster","Mean12", "SD12")   
   
var13table <- merge(Meanvar13, SDvar13, by = "Cluster", sort = TRUE)  
colnames(var13table) <- c("Cluster","Mean13", "SD13")   
   
   
table <- var1table %>% merge(var2table, by = "Cluster", sort = TRUE) %>% merge(var3table, by = "Cluster", sort = TRUE) %>%  
 merge(var4table, by = "Cluster", sort = TRUE) %>% merge(var5table, by = "Cluster", sort = TRUE) %>% merge(var6table, by = "Cluster", sort = TRUE) %>%  
 merge(var7table, by = "Cluster", sort = TRUE) %>% merge(var8table, by = "Cluster", sort = TRUE) %>% merge(var9table, by = "Cluster", sort = TRUE) %>%  
 merge(var10table, by = "Cluster", sort = TRUE) %>% merge(var11table, by = "Cluster", sort = TRUE)%>% merge(var12table, by = "Cluster", sort = TRUE) %>%  
 merge(var13table, by = "Cluster", sort = TRUE)  
  
table <- cbind(table, "Elements in Cluster" =clustercount[, 1])  
  
rownames(table) <- c(clust1name,clust2name,clust3name)  
  
table <- table[c(0,1, 28, 2, 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27)] #Change the order of columns  
  
labels <- data.frame( "", "", var1, var1, var2, var2, var3, var3, var4, var4, var5, var5, var6,var6, var7, var7, var8, var8, var9,var9, var10,var10, var11,  
 var11, var12, var12, var13, var13)  
  
colnames(labels) <- c("Cluster","Elements in Cluster", "Mean1", "SD1", "Mean2", "SD2", "Mean3", "SD3", "Mean4", "SD4", "Mean5", "SD5", "Mean6", "SD6", "Mean7", "SD7", "Mean8", "SD8", "Mean9", "SD9", "Mean10", "SD10", "Mean11", "SD11", "Mean12", "SD12", "Mean13", "SD13")  
table <- rbind(table, labels)  
rownames(table) <- c(clust1name,clust2name,clust3name, "")  
table

## Cluster Elements in Cluster Mean1 SD1  
## Top 1 62 13.6767741935484 0.55909403940382  
## Mid 2 51 13.1341176470588 0.521855062141159  
## Lower 3 65 12.2509230769231 0.5330095070591  
## Alcohol Alcohol  
## Mean2 SD2 Mean3 SD3  
## Top 1.99790322580645 0.675642104058287 2.46629032258065 0.24534508230859  
## Mid 3.30725490196078 1.09336559015066 2.41764705882353 0.199023498464821  
## Lower 1.89738461538462 1.01225348376018 2.23123076923077 0.299199492878015  
## Malic\_Acid Malic\_Acid Ash Ash  
## Mean4 SD4 Mean5 SD5  
## Top 17.4629032258065 3.30946154161055 107.967741935484 13.1535294047325  
## Mid 21.2411764705882 2.40800136798867 98.6666666666667 11.0736925488595  
## Lower 20.0630769230769 3.01944579097148 92.7384615384615 13.6808224842717  
## Ash\_Alcanity Ash\_Alcanity Magnesium Magnesium  
## Mean6 SD6 Mean7 SD7  
## Top 2.84758064516129 0.34124019464143 3.00322580645161 0.47997873652885  
## Mid 1.68392156862745 0.349497229925346 0.818823529411765 0.323911389480664  
## Lower 2.24769230769231 0.53029110292947 2.05 0.608948889480882  
## Total\_Phenols Total\_Phenols Flavanoids Flavanoids  
## Mean8 SD8 Mean9  
## Top 0.292096774193548 0.0728829534764911 1.92209677419355  
## Mid 0.451960784313725 0.123028770746409 1.14588235294118  
## Lower 0.357692307692308 0.12143374261775 1.62415384615385  
## Nonflavanoid\_Phenols Nonflavanoid\_Phenols Proanthocyanins  
## SD9 Mean10 SD10 Mean11  
## Top 0.438496451550307 5.45354838709677 1.29419649960411 1.06548387096774  
## Mid 0.411335271867552 7.2347058627451 2.34559615133402 0.691960784313725  
## Lower 0.570397044576441 2.97307692307692 0.791805537752411 1.06270769230769  
## Proanthocyanins Color\_Intensity Color\_Intensity Hue  
## SD11 Mean12 SD12 Mean13  
## Top 0.119749429915544 3.16338709677419 0.363773229435219 1100.22580645161  
## Mid 0.120383048770882 1.69666666666667 0.27485026226414 619.058823529412  
## Lower 0.201925862251852 2.80338461538462 0.457811631989201 510.169230769231  
## Hue OD280 OD280 Proline  
## SD13  
## Top 232.721772907918  
## Mid 120.723553917983  
## Lower 142.732438634186  
## Proline

## ANOVA table  
# Degrees of freedom  
countobs <- count(mydata)  
betw.grpsdf <- m - 1  
within.grpsdf <- countobs - m  
totdf <- countobs-1

## Sum of Squares between groups, var1  
clusterpath <- databind$Cluster #set cluster path  
varpath <- var1path # set variable path ### CHANGE  
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar1[1,2])   
MeanTwo <-as.integer(Meanvar1[2,2])   
MeanThree <- as.integer(Meanvar1[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var1ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var1ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var1ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var2  
varpath <- var2path # set variable path ### CHANGE  
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar2[1,2])   
MeanTwo <-as.integer(Meanvar2[2,2])   
MeanThree <- as.integer(Meanvar2[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var2ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var2ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var2ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var3  
varpath <- var3path # set variable path  
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar3[1,2])   
MeanTwo <-as.integer(Meanvar3[2,2])   
MeanThree <- as.integer(Meanvar3[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var3ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var3ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var3ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var4  
varpath <- var4path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar4[1,2])   
MeanTwo <-as.integer(Meanvar4[2,2])   
MeanThree <- as.integer(Meanvar4[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var4ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var4ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var4ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var5  
varpath <- var5path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar5[1,2])   
MeanTwo <-as.integer(Meanvar5[2,2])   
MeanThree <- as.integer(Meanvar5[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var5ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var5ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var5ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var6path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar6[1,2])   
MeanTwo <-as.integer(Meanvar6[2,2])   
MeanThree <- as.integer(Meanvar6[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var6ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var6ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var6ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var7  
varpath <- var6path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar7[1,2])   
MeanTwo <-as.integer(Meanvar7[2,2])   
MeanThree <- as.integer(Meanvar7[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var7ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var7ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var7ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var8  
varpath <- var8path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar8[1,2])   
MeanTwo <-as.integer(Meanvar8[2,2])   
MeanThree <- as.integer(Meanvar8[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var8ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var8ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var8ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var9  
varpath <- var9path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar9[1,2])   
MeanTwo <-as.integer(Meanvar9[2,2])   
MeanThree <- as.integer(Meanvar9[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var9ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var9ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var9ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var10path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar10[1,2])   
MeanTwo <-as.integer(Meanvar10[2,2])   
MeanThree <- as.integer(Meanvar10[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var10ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var10ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var10ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var11path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar11[1,2])   
MeanTwo <-as.integer(Meanvar11[2,2])   
MeanThree <- as.integer(Meanvar11[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var11ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var11ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var11ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var12path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar12[1,2])   
MeanTwo <-as.integer(Meanvar12[2,2])   
MeanThree <- as.integer(Meanvar12[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var12ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var12ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var12ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

## Sum of Squares between groups, var6  
varpath <- var13path # set variable path   
  
y\_j <- tapply(varpath , clusterpath, mean) # Calculate group means  
y\_t <- mean(varpath) # Calculate the grand mean  
ssbetween <- sum((y\_j - y\_t)^2) # Calculate the sum of squares  
  
  
## Sum of Squares within salary group  
# sum((x - mean(x) )^2 )  
# Create a separate vector of IQ gains for each training group  
y\_i1 <- subset(varpath, clusterpath == "1")  
y\_i2 <- subset(varpath, clusterpath == "2")  
y\_i3 <- subset(varpath, clusterpath == "3")  
  
  
# Subtract group means from the individual values  
MeanOne <- as.integer(Meanvar13[1,2])   
MeanTwo <-as.integer(Meanvar13[2,2])   
MeanThree <- as.integer(Meanvar13[3,2])   
  
s\_1 <- y\_i1 - MeanOne  
s\_2 <- y\_i2 - MeanTwo  
s\_3 <- y\_i3 - MeanThree  
  
# Put everything back together into one vector  
s\_t <- c(s\_1, s\_2, s\_3)  
  
# Calculate the sum of squares using s\_t  
sswithin <- data.frame(sum(s\_t^2))  
sswithin <- sswithin[1,1]  
  
# Mean Square Salary Between Groups  
msbetween <- ssbetween/betw.grpsdf   
#msSalarybetween <- msSalarybetween[1,1]  
  
# Mean Square Salary Within Groups  
mswithin <- sswithin/within.grpsdf   
mswithin <- mswithin[1,1]  
  
# F-stat  
Fstat <- msbetween/mswithin  
# Fstat <- Fstat[1,1]  
# Total Sum of Squares  
totss <- sswithin + ssbetween  
#totssSalary <- totssSalary[1,1]  
  
  
var13ANOVA <- matrix(c(ssbetween,betw.grpsdf,msbetween,Fstat,sswithin,within.grpsdf,mswithin, "", totss, totdf, "", ""),ncol=4,nrow = 3, byrow=TRUE)  
rownames(var13ANOVA) <- c("Between Values","Within Values","Total")  
colnames(var13ANOVA) <- c("Sum of Squares","DF","Mean Square", "F-Statistic")

labels <- t(data.frame(var1, "", "", var2, "", "", var3, "", "", var4, "", "", var5, "", "", var6, "", "", var7, "", "", var8, "", "", var9, "", "", var10, "", "", var11, "", "", var12, "", "", var13, "", ""))  
ANOVAtable <- rbind(var1ANOVA, var2ANOVA, var3ANOVA, var4ANOVA, var5ANOVA, var6ANOVA,var7ANOVA, var8ANOVA, var9ANOVA, var10ANOVA, var11ANOVA, var12ANOVA, var13ANOVA)  
cbind(ANOVAtable, labels)

## Sum of Squares DF Mean Square F-Statistic  
## Between Values 1.037052 2 0.5185259 1.076749   
## Within Values 84.2741 175 0.4815663 ""   
## Total 85.31115 177 "" ""   
## Between Values 1.249894 2 0.6249469 0.4019381   
## Within Values 272.0959 175 1.554834 ""   
## Total 273.3458 177 "" ""   
## Between Values 0.03087137 2 0.01543568 0.07254897   
## Within Values 37.2334 175 0.2127623 ""   
## Total 37.26427 177 "" ""   
## Between Values 7.501293 2 3.750646 0.4212776   
## Within Values 1558.03 175 8.903029 ""   
## Total 1565.531 177 "" ""   
## Between Values 117.8688 2 58.93442 0.3583573   
## Within Values 28780 175 164.4571 ""   
## Total 28897.87 177 "" ""   
## Between Values 0.6810241 2 0.340512 0.5752375   
## Within Values 103.5913 175 0.5919503 ""   
## Total 104.2723 177 "" ""   
## Between Values 0.6810241 2 0.340512 0.3287679   
## Within Values 181.2513 175 1.035722 ""   
## Total 181.9323 177 "" ""   
## Between Values 0.01300262 2 0.006501312 0.04367736   
## Within Values 26.0485 175 0.1488486 ""   
## Total 26.0615 177 "" ""   
## Between Values 0.3088376 2 0.1544188 0.2249411   
## Within Values 120.135 175 0.6864857 ""   
## Total 120.4438 177 "" ""   
## Between Values 9.241324 2 4.620662 1.635225   
## Within Values 494.4982 175 2.825704 ""   
## Total 503.7395 177 "" ""   
## Between Values 0.09323496 2 0.04661748 0.2798683   
## Within Values 29.14964 175 0.1665693 ""   
## Total 29.24287 177 "" ""   
## Between Values 1.178383 2 0.5891913 1.101309   
## Within Values 93.6236 175 0.534992 ""   
## Total 94.80198 177 "" ""   
## Between Values 197223.8 2 98611.9 3.233915   
## Within Values 5336282 175 30493.04 ""   
## Total 5533506 177 "" ""   
##   
## Between Values "Alcohol"   
## Within Values ""   
## Total ""   
## Between Values "Malic\_Acid"   
## Within Values ""   
## Total ""   
## Between Values "Ash"   
## Within Values ""   
## Total ""   
## Between Values "Ash\_Alcanity"   
## Within Values ""   
## Total ""   
## Between Values "Magnesium"   
## Within Values ""   
## Total ""   
## Between Values "Total\_Phenols"   
## Within Values ""   
## Total ""   
## Between Values "Flavanoids"   
## Within Values ""   
## Total ""   
## Between Values "Nonflavanoid\_Phenols"  
## Within Values ""   
## Total ""   
## Between Values "Proanthocyanins"   
## Within Values ""   
## Total ""   
## Between Values "Color\_Intensity"   
## Within Values ""   
## Total ""   
## Between Values "Hue"   
## Within Values ""   
## Total ""   
## Between Values "OD280"   
## Within Values ""   
## Total ""   
## Between Values "Proline"   
## Within Values ""   
## Total ""