CSCI946 Assignment

Yao Xiao
SID 2019180015

October 26, 2020

1 Task 1: Hypothesis Testing

Null hypothesis:

The approach1 and approach2 do not effectively improve student learning performance.

Alternative hypothesis:

The two new learning approaches do effectively improve student learning performance.

First, we should divide the datasets into 3 parts

```

> dataframe <- read.csv("A1_performance_test.csv")
> app1 <- dataframe[dataframe$approach=="approach1",]$performance
> app2 <- dataframe[dataframe$approach=="approach2",]$performance
> appNo <- dataframe[dataframe$approach=="no_approach",]$performance
> summary(app1)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  -1.073   54.815   74.100   77.345   95.648  155.282
> summary(app2)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   14.97    63.08    82.48    83.30   102.14   161.37
> summary(appNo)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  -23.39    19.03    38.88    40.94    62.90   119.99

> t.test(app1,appNo,var.equal = TRUE)

#        Two Sample t-test

# data:  app1 and appNo
# t = 11.93, df = 379, p-value < 2.2e-16
# alternative hypothesis: true difference in means is not equal to 0
# 95 percent confidence interval:
#  30.40739 42.40905
# sample estimates:
```

```
# mean of x mean of y
# 77.34459  40.93637
> qt(p=0.05/2, df=379, lower.tail= FALSE)
# [1] 1.966243
```

We can see that $|1-11.93| > 1.97$, so the original hypthsis will be denied.

```
> t.test(app2,appNo,var.equal = TRUE)

#        Two Sample t-test

# data:  app2 and appNo
# t = 14.021, df = 401, p-value < 2.2e-16
# alternative hypothesis: true difference in means is not equal to
  0
# 95 percent confidence interval:
#  36.42716 48.30779
# sample estimates:
# mean of x mean of y
#  83.30384  40.93637

> qt(p=0.05/2, df=401, lower.tail= FALSE)
# [1] 1.965897
```

We can see that $|1-14.021| > 1.966$, so the original hypthsis will be denied.

For the approach 1&2:

```
> t.test(app1,app2,var.equal = TRUE)

#        Two Sample t-test

# data:  app1 and app2
# t = -1.9988, df = 414, p-value = 0.04629
# alternative hypothesis: true difference in means is not equal to
  0
# 95 percent confidence interval:
#  -11.81998428  -0.09851884
# sample estimates:
# mean of x mean of y
#  77.34459  83.30384

> qt(p=0.05/2, df=414, lower.tail= FALSE)
# [1] 1.965711
```

We can see that $|1-1.9988| = 0.9988 < 1.9657$, the original hypothesis will not be denied.

Conclusion:

The two new learning approaches do effectively improve student learning performance. In terms of improving students' learning performance, there is no significant difference between approach1 and approach2.

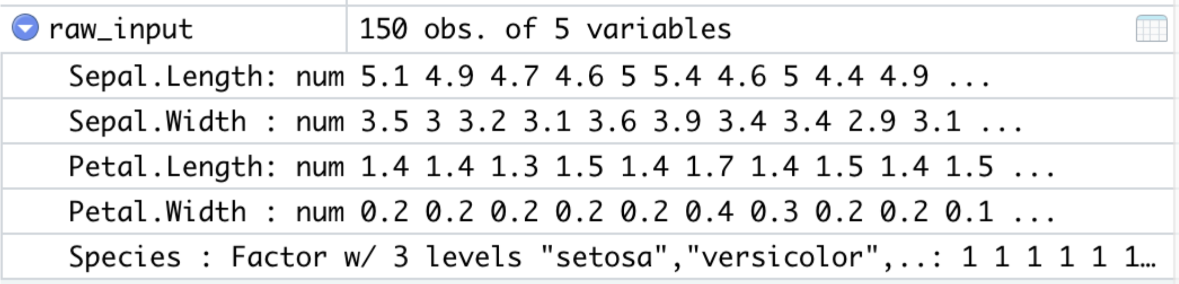
Souce Code:

```
dataframe <- read.csv(”A1_performance_test.csv”)
app1 <- dataframe[dataframe$approach==”approach1”,]$performance
app2 <- dataframe[dataframe$approach==”approach2”,]$performance
appNo <- dataframe[dataframe$approach==”no_approach”,]$performance

summary(app1)
summary(app2)
summary(appNo)

t.test(app1,appNo,var.equal = TRUE)
qt(p=0.05/2, df=379, lower.tail= FALSE)

t.test(app2,appNo,var.equal = TRUE)
qt(p=0.05/2, df=401, lower.tail= FALSE)

t.test(app1,app2,var.equal = TRUE)
qt(p=0.05/2, df=414, lower.tail= FALSE)
```

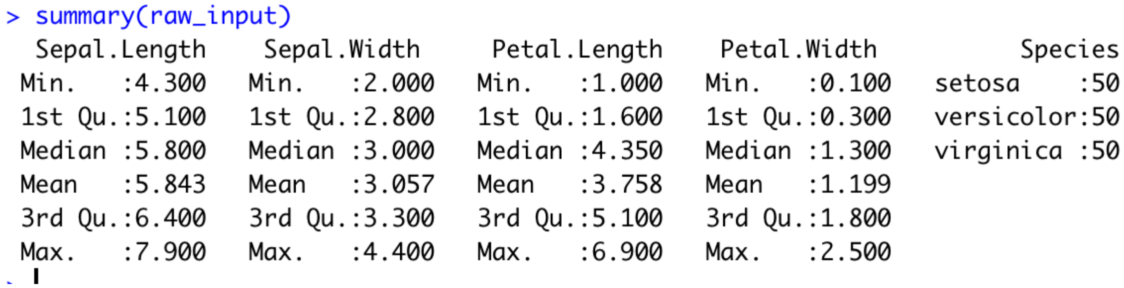
2 Task 2: Clustering

2.1 Answer 1

The Iris dataset has 5 attributes: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Species. And the dataset has 150 datas and each data belongs to one of 3 species(setosa, versicolor, virginica), for the species, it uses length and width of flower’s sepal and petal to describe.



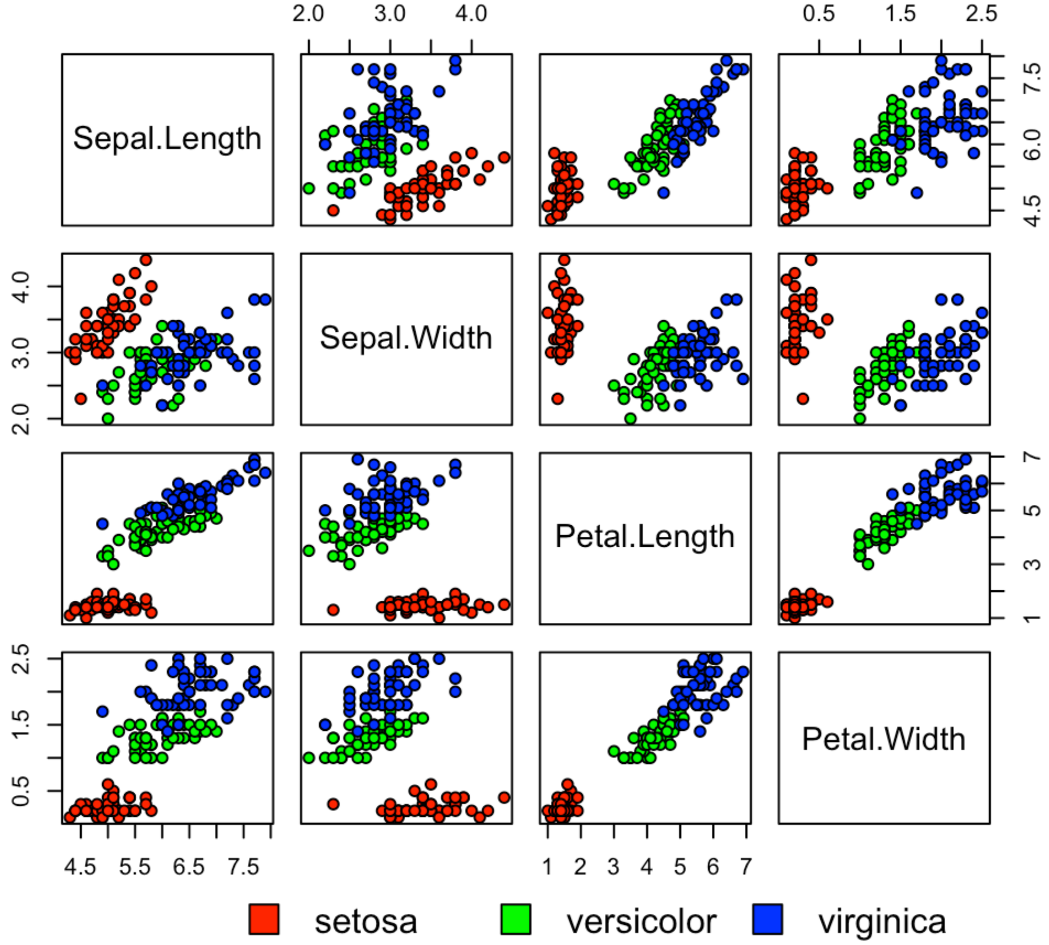
```
raw_input      150 obs. of 5 variables
   Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
   Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
   Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
   Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
   Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1…
```



```
> summary(raw_input)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                  
```

2.2 Answer 2

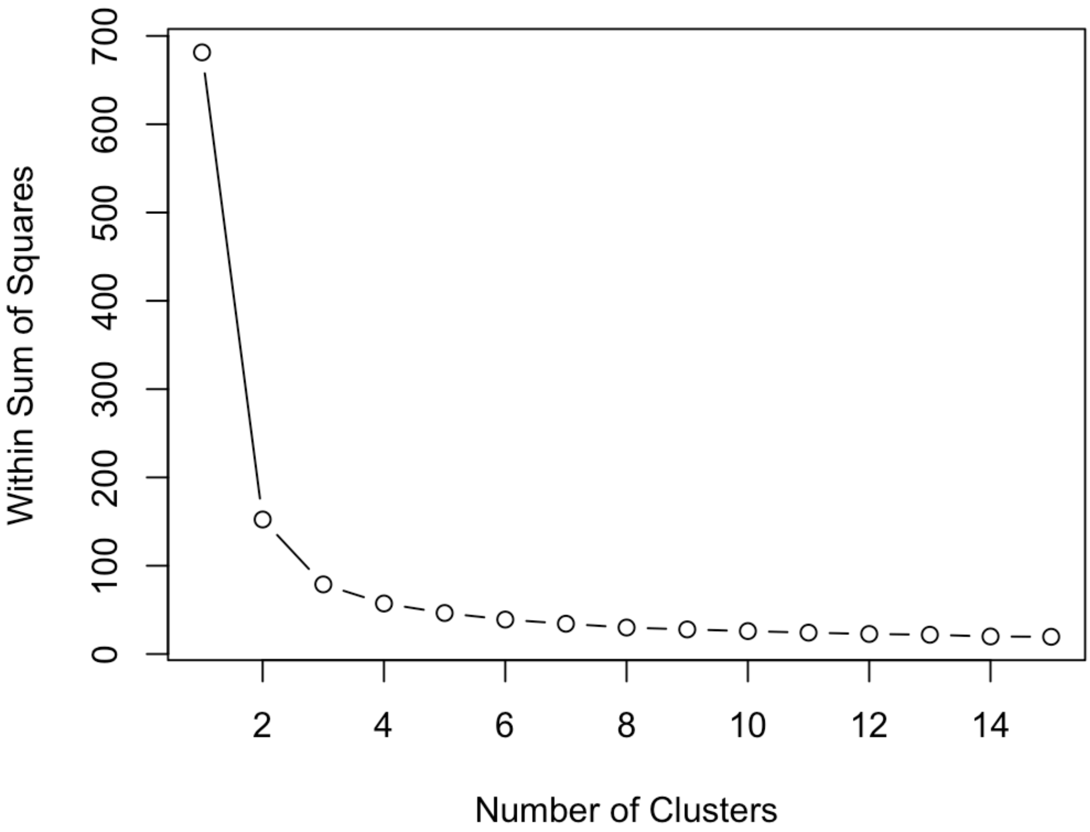
Figure 1: Iris Dataset



2.3 Answer 3

We should first remove the species attribute, and use the k-means algorithm to calculate the clustering k from 1 to 15, and calculate wss for each k value.

Figure 2: WSS of k value



We can find when k > 3 from the figure 2, the trend of wss becomes linear. Therefore, the k-means analysis should select k = 3.

```
> km = kmeans(kmdata,3)
> km
# K-means clustering with 3 clusters of sizes 62, 50, 38

# Cluster means:
#   Sepal.Length Sepal.Width Petal.Length Petal.Width
# 1     5.901613    2.748387     4.393548    1.433871
# 2     5.006000    3.428000     1.462000    0.246000
# 3     6.850000    3.073684     5.742105    2.071053

# Clustering vector:
#   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
  2 2 2 2 2 2 2 2
#  [39] 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   1 1 1 1 1 1 1 1
#  [77] 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3
   1 3 3 3 3 3 3 1
# [115] 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3
   3 3 1 3 3 1

# Within cluster sum of squares by cluster:
# [1] 39.82097 15.15100 23.87947
#  (between_SS / total_SS =  88.4 %)

# Available components:

# [1] "cluster"      "centers"      "totss"        "withinss"     "
  tot.withinss"
# [6] "betweenss"    "size"         "iter"         "ifault"

> str(km)
# List of 9
#  $ cluster     : int [1:150] 2 2 2 2 2 2 2 2 2 2 ...
#  $ centers     : num [1:3, 1:4] 5.9 5.01 6.85 2.75 3.43 ...
#   ..- attr(*, "dimnames")=List of 2
#   .. ..$ : chr [1:3] "1" "2" "3"
#  .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "
  Petal.Width"
#  $ totss       : num 681
#  $ withinss    : num [1:3] 39.8 15.2 23.9
#  $ tot.withinss: num 78.9
#  $ betweenss   : num 603
#  $ size        : int [1:3] 62 50 38
#  $ iter        : int 2
#  $ ifault      : int 0
#  - attr(*, "class")= chr "kmeans"

> table(iris$Species,km$cluster)

#               1  2  3
#   setosa      0 50  0
#   versicolor 48  0  2
#   virginica  14  0 36
```

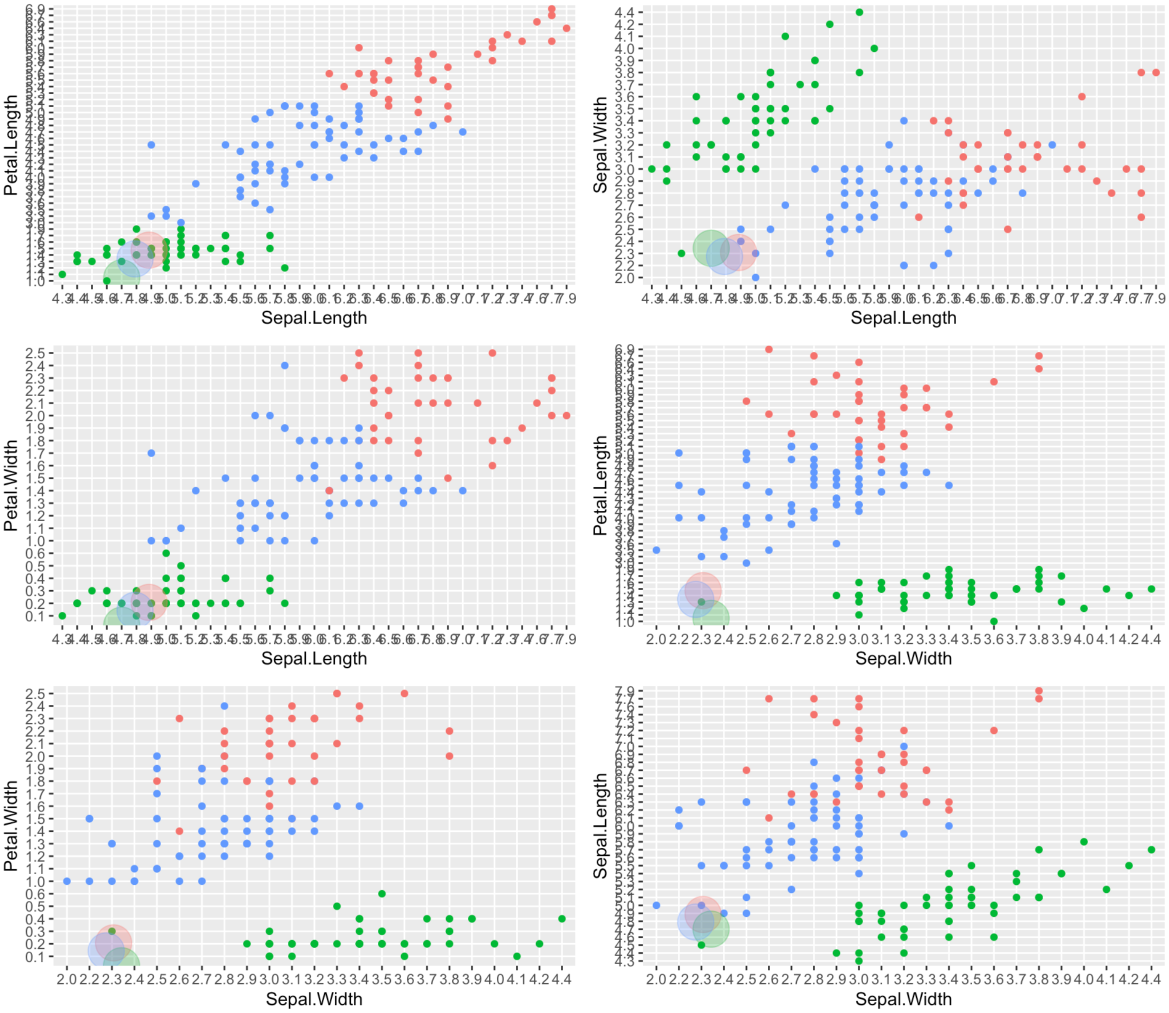
2.4 Answer 4

From the figure 3, we can conclude:

1. Most points in different clusters are well separted from each other, however there are still few points appear in other cluster.
2. No clusters have a few points.

3. Actually the centroids are too close to each other.

Figure 3: Visualization of results



2.5 Answer 5

Here we extract 50 data vs 100 data randomly and perform hierarchical agglomerative clustering

Figure 4: 50 data cluster dendrogram

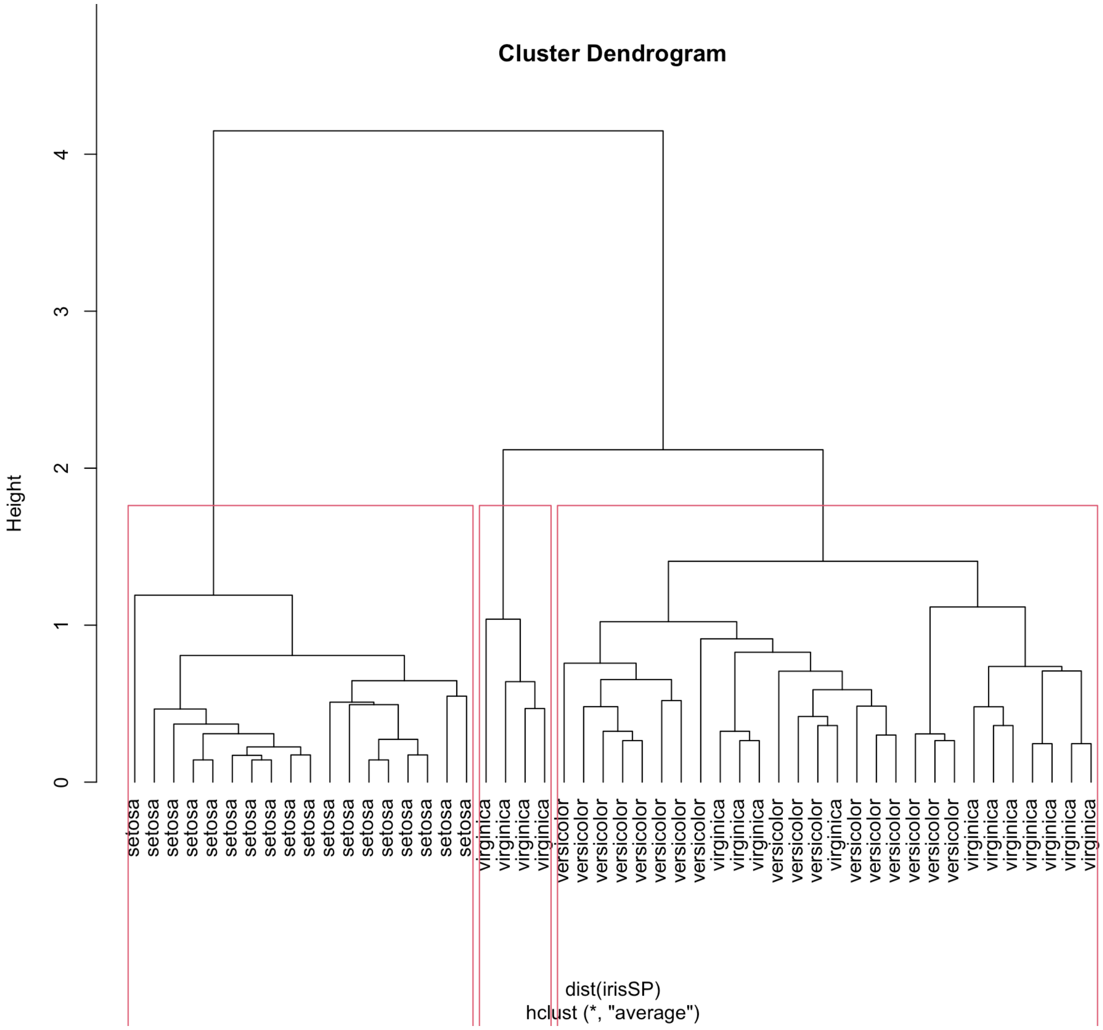
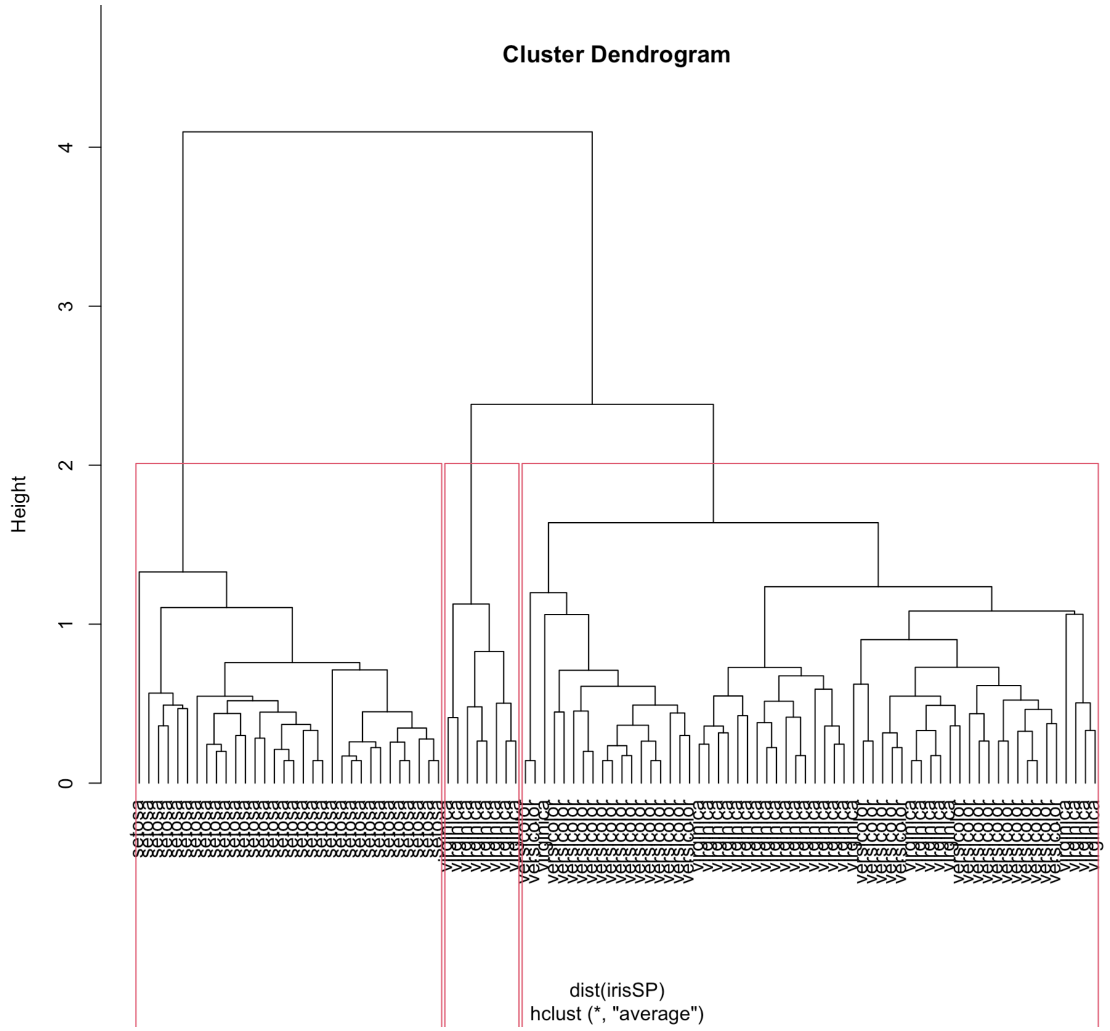


Figure 5: 100 data cluster dendrogram



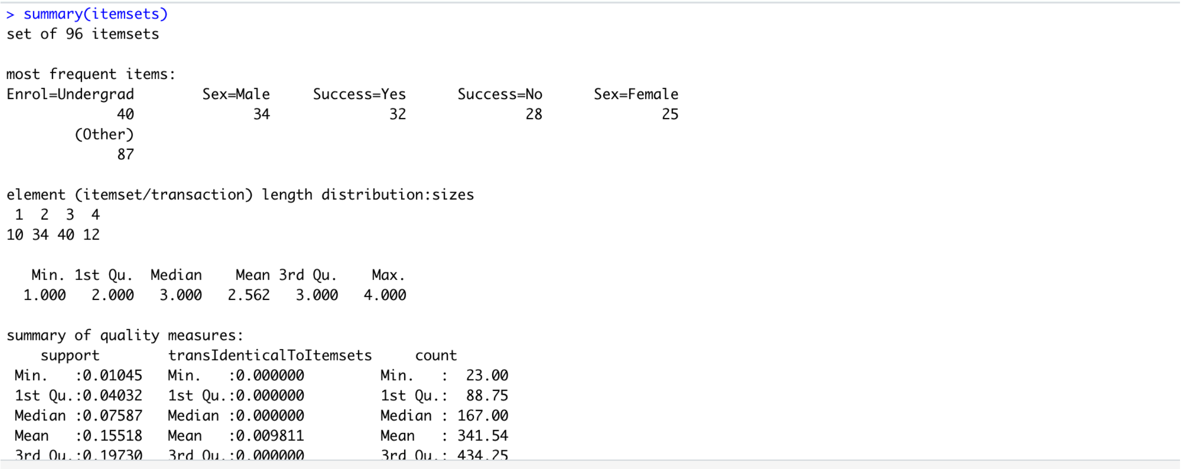
2.6 Source Code

```
library(plyr)
#install.packages('ggplot2')
#install.packages('colorspace')
library(ggplot2)
library(cluster)
library(lattice)
library(graphics)
library(grid)
library(gridExtra)

raw_input = as.data.frame(iris)
kmdata_org = as.matrix(raw_input[,c("Sepal.Length","Sepal.Width","Petal.Length","Petal.Width","Species")])
```

```
summary(raw_input)

colors <-c(”red”,”green”,”blue”)
pairs(iris[1:4],pch=21,bg=colors[unclass(iris$Species)])

par(xpd=TRUE)
legend(0.2,0.02,horiz=TRUE,as.vector(unique(iris$Species)),fill=colors,bty=”n”)

kmdata <- kmdata_org[,1:4]

wss <- numeric(15)
for (k in 1:15)
  wss[k] <- sum(kmeans(kmdata, centers=k,nstart=25)$withinss)
plot(1:15, wss, type=”b”, xlab=”Number␣of␣Clusters”, ylab=”Within␣Sum␣of␣Squares”)

km = kmeans(kmdata, 3)
km
str(km)
table(iris$Species,km$cluster)

df = as.data.frame(kmdata[,1:4])
df$cluster = factor(km$cluster)
centers = as.data.frame(km$centers)

fig1 = ggplot(data=df, aes(x=Sepal.Length, y=Petal.Length, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Length,y=Petal.Length, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend = FALSE)
fig2 = ggplot(data=df, aes(x=Sepal.Length, y=Sepal.Width, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Length, y=Sepal.Width, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend = FALSE)
fig3 = ggplot(data=df, aes(x=Sepal.Length, y=Petal.Width, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Length,y=Petal.Width, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend = FALSE)
fig4 = ggplot(data=df, aes(x=Sepal.Width, y=Petal.Length, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Width,y=Petal.Length, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend = FALSE)
fig5 = ggplot(data=df, aes(x=Sepal.Width, y=Petal.Width, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Width,y=Petal.Width, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend =  FALSE)
fig6 = ggplot(data=df, aes(x=Sepal.Width, y=Sepal.Length, color=cluster ))+geom_point() + geom_point(data=centers,aes(x=Sepal.Width,y=Sepal.Length, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend = FALSE)
grid.arrange(arrangeGrob(fig1 + theme(legend.position = ”none”),
                                                fig2 + theme(legend.position = ”none”),
                                                fig3 + theme(legend.position = ”none”),
                                                fig4 + theme(legend.position = ”none”),
                                                fig5 + theme(legend.position = ”none”),
                                                fig6 + theme(legend.position = ”none”),
                                                ncol = 2))

idx <− sample(1:dim(iris)[1],100)
irisSP <− iris[idx,]
irisSP$Species <− NULL

hc <− hclust(dist(irisSP),method = ”ave”)
plot(hc, hang= −1, labels = iris$Species[idx])

rect.hclust(hc, k=3)
groups <− cutree(hc, k=3)
```

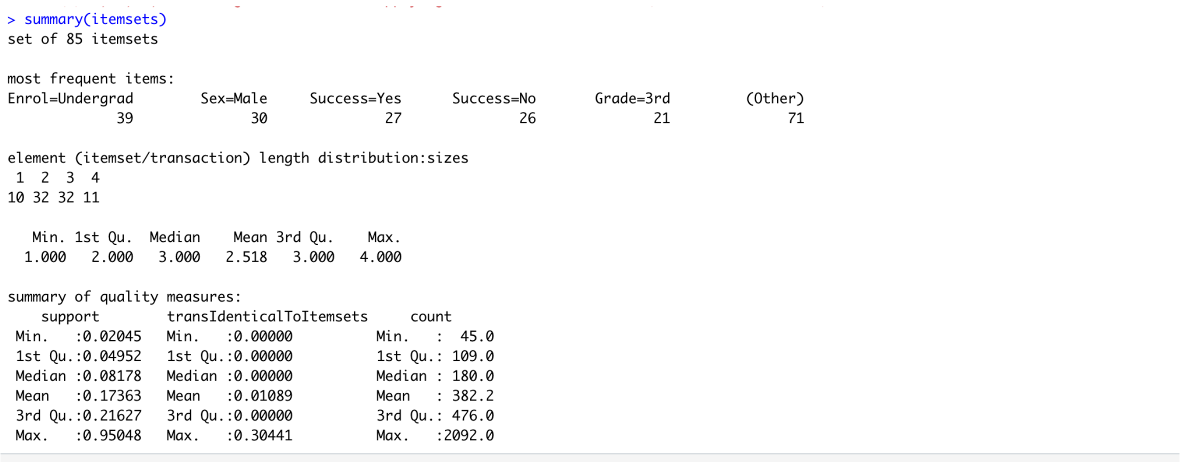
3 Task 3: Association Rule

Use support 0.01:



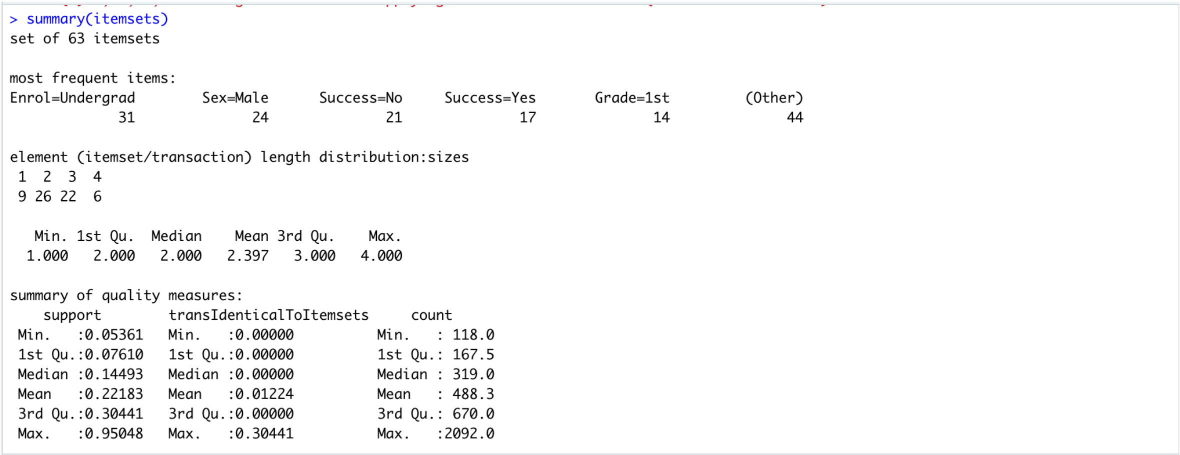
```
> summary(itemsets)
set of 96 itemsets

most frequent items:
Enrol=Undergrad        Sex=Male     Success=Yes      Success=No      Sex=Female 
             40              34              32              28              25 
        (Other) 
             87 

element (itemset/transaction) length distribution:sizes
 1  2  3  4 
10 34 40 12 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   3.000   2.562   3.000   4.000 

summary of quality measures:
    support        transIdenticalToItemsets     count       
 Min.   :0.01045   Min.   :0.000000         Min.   :  23.00  
 1st Qu.:0.04032   1st Qu.:0.000000         1st Qu.:  88.75  
 Median :0.07587   Median :0.000000         Median : 167.00  
 Mean   :0.15518   Mean   :0.009811         Mean   : 341.54  
 3rd Qu.:0.19730   3rd Qu.:0.000000         3rd Qu.: 434.25  
```

Use support 0.02:



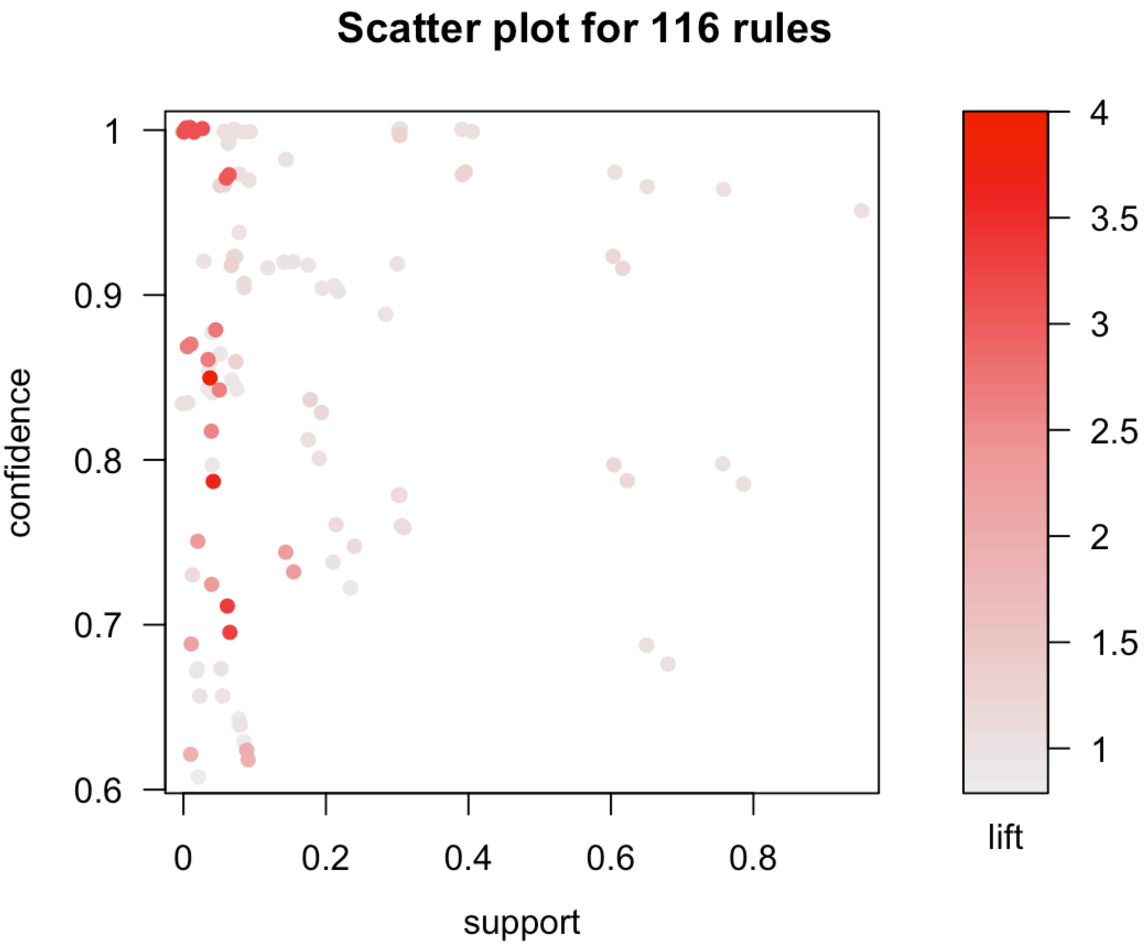
```
> summary(itemsets)
set of 85 itemsets

most frequent items:
Enrol=Undergrad        Sex=Male     Success=Yes      Success=No       Grade=3rd         (Other) 
             39              30              27              26              21              71 

element (itemset/transaction) length distribution:sizes
 1  2  3  4 
10 32 32 11 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   3.000   2.518   3.000   4.000 

summary of quality measures:
    support        transIdenticalToItemsets     count       
 Min.   :0.02045   Min.   :0.00000          Min.   :  45.0  
 1st Qu.:0.04952   1st Qu.:0.00000          1st Qu.: 109.0  
 Median :0.08178   Median :0.00000          Median : 180.0  
 Mean   :0.17363   Mean   :0.01089          Mean   : 382.2  
 3rd Qu.:0.21627   3rd Qu.:0.00000          3rd Qu.: 476.0  
 Max.   :0.95048   Max.   :0.30441          Max.   :2092.0  
```

Use support 0.05:

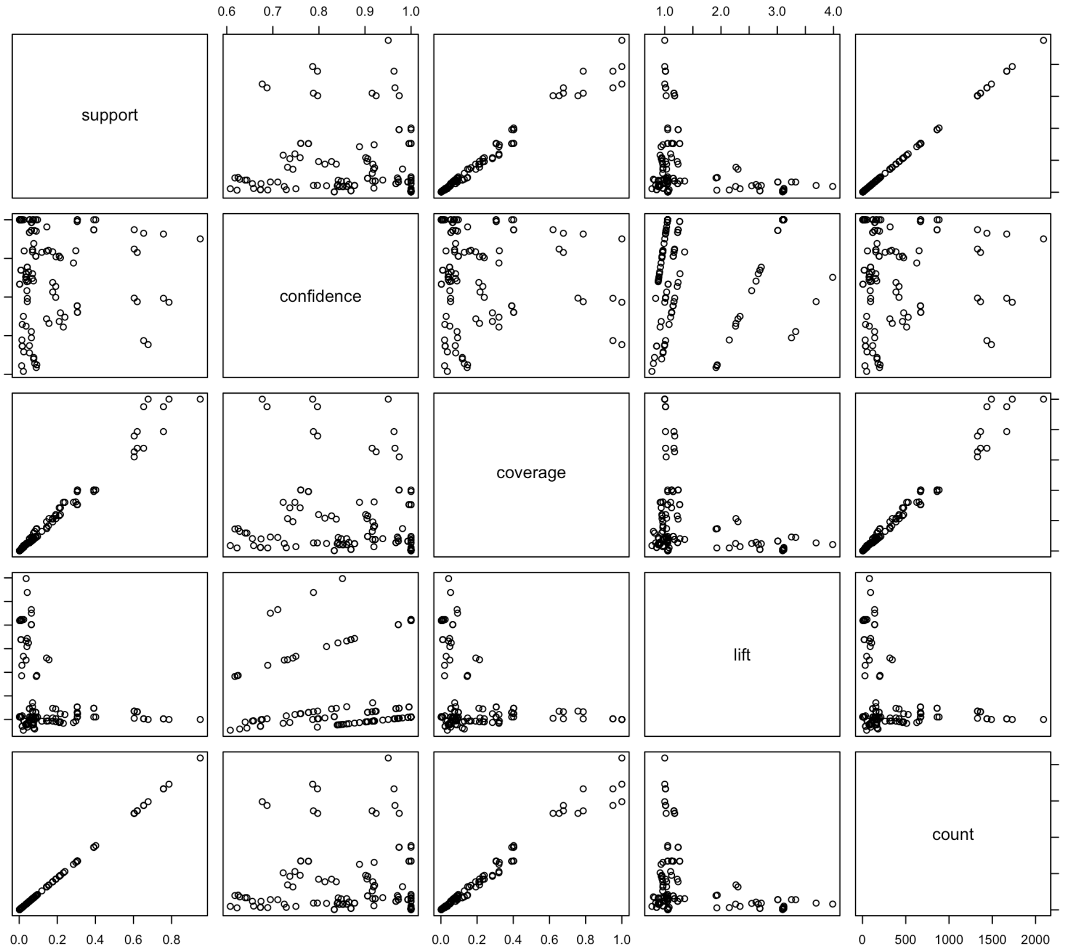


```
> summary(itemsets)
set of 63 itemsets

most frequent items:
Enrol=Undergrad        Sex=Male      Success=No     Success=Yes       Grade=1st         (Other) 
             31              24              21              17              14              44 

element (itemset/transaction) length distribution:sizes
 1  2  3  4 
 9 26 22  6 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   2.000   2.397   3.000   4.000 

summary of quality measures:
    support        transIdenticalToItemsets     count       
 Min.   :0.05361   Min.   :0.00000          Min.   : 118.0  
 1st Qu.:0.07610   1st Qu.:0.00000          1st Qu.: 167.5  
 Median :0.14493   Median :0.00000          Median : 319.0  
 Mean   :0.22183   Mean   :0.01224          Mean   : 488.3  
 3rd Qu.:0.30441   3rd Qu.:0.00000          3rd Qu.: 670.0  
 Max.   :0.95048   Max.   :0.30441          Max.   :2092.0  
```

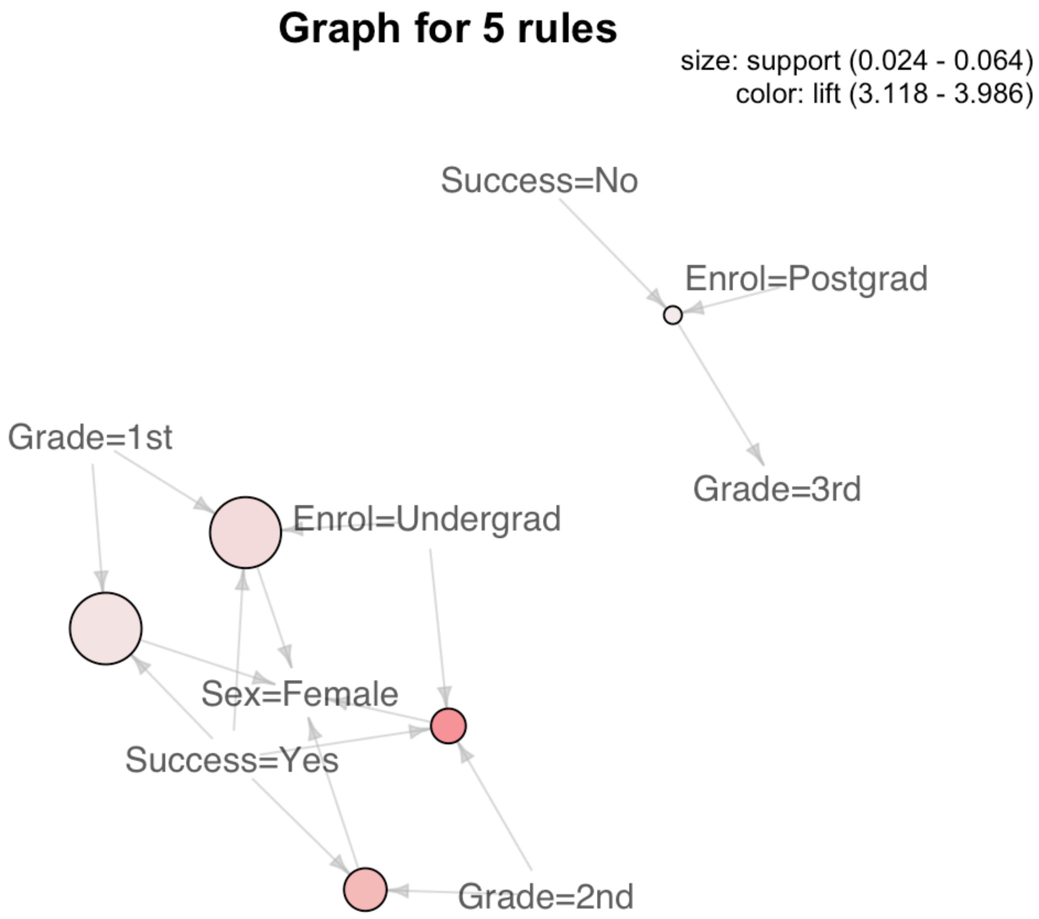
Visualization of rules:



Relationship among support, confidence and lift:



Visualization of graphs:



Source Code:

```
library(arules)
library(arulesViz)

df <- read.csv(”A1_success_data.csv”)

itemsets<- apriori(df, parameter=list(minlen=1, maxlen=10, support=0.01, target=”frequent␣itemsets”))

summary(itemsets)

itemsets<- apriori(df, parameter=list(minlen=1, maxlen=10, support=0.02, target=”frequent␣itemsets”))

summary(itemsets)

```

```
itemsets<- apriori(df, parameter=list(minlen=1, maxlen=10, support=0.05, target="frequent itemsets"))

summary(itemsets)


rules<- apriori(df, parameter=list(support=0.001,confidence=0.6, target = "rules"))
plot(rules)
plot(rules@quality)

slope<- sort(round(rules@quality$lift / rules@quality$confidence, 2))
unlist(lapply(split(slope,f=slope),length))
inspect(head(sort(rules, by="lift"), 10))
inspect(head(sort(rules, by="confidence"), 10))
inspect(head(sort(rules, by="support"), 10))

confidentRules<- rules[quality(rules)$confidence > 0.9]
plot(confidentRules, method="matrix", measure=c("lift", "confidence"))

highLiftRules <- head(sort(rules, by="lift"), 5)
plot(highLiftRules, method="graph", control=list(type="items"))

test<-inspect(sort(rules, by="lift"))
test[test$rhs=="{Success=Yes}",]
test[test$rhs=="{Success=No}",]
```