Homework 5

Kenigbolo Meya Stephen

March 9, 2016

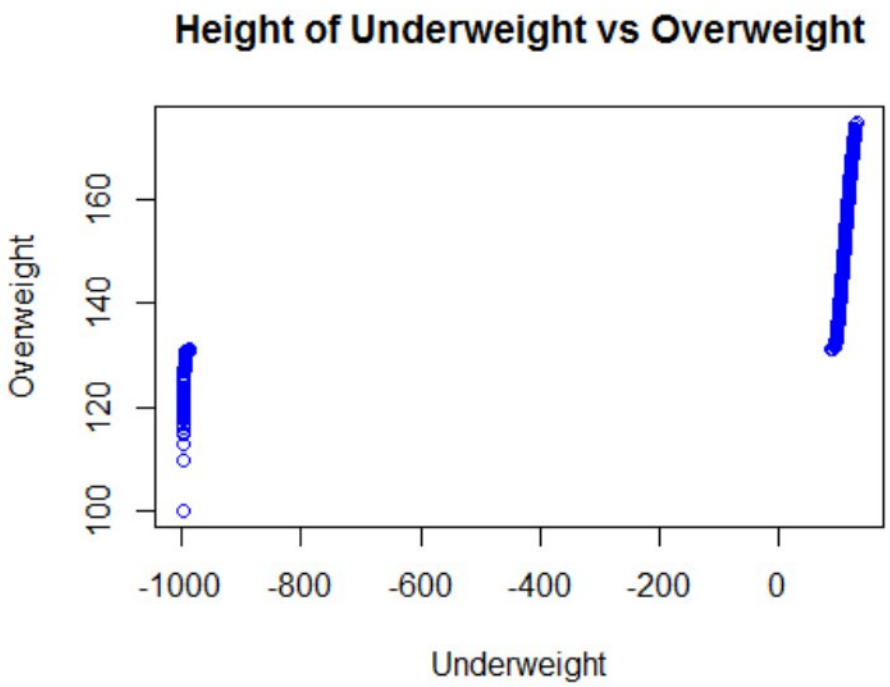
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When plotting some plot, please provide your interpretation - what is your conclusion based on that plot.

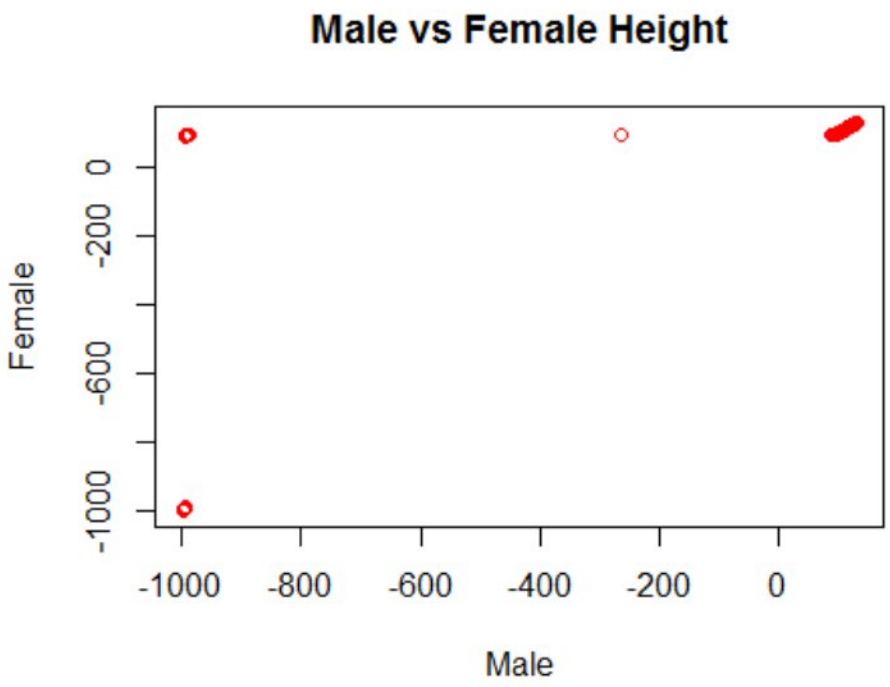
1. Use the data of child height/weight and study them using qq-plots in a specific age and gender at a time.

Compare heights of underweight and very overweight children Compare one of the attributes (height, weight, BMI) between boys and girls (select either younger or older age group) to each other.

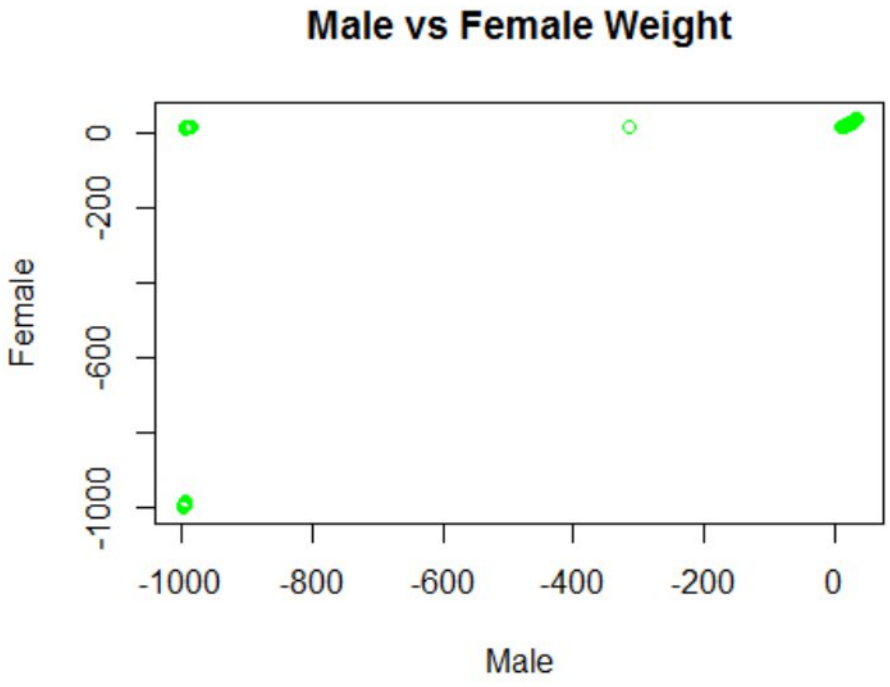
```
uk_childData <- read.csv('C:/Users/Kenigbolo PC/Desktop/Data
Mining/ncmp_1415_final_non_disclosive.csv')
childData <- uk_childData[,c(2, 3, 5, 8, 11)]

underweight <- subset(childData, bmi < 18.5 & ageinmonths <= 65.60)
overweight <- subset(childData, ageinmonths > 65.60 & bmi > 25)

qqplot(underweight$height, overweight$height, plot.it = TRUE, main="Height of
Underweight vs Overweight", col="blue", xlab =
deparse(substitute(Underweight)), ylab = deparse(substitute(Overweight)))
```



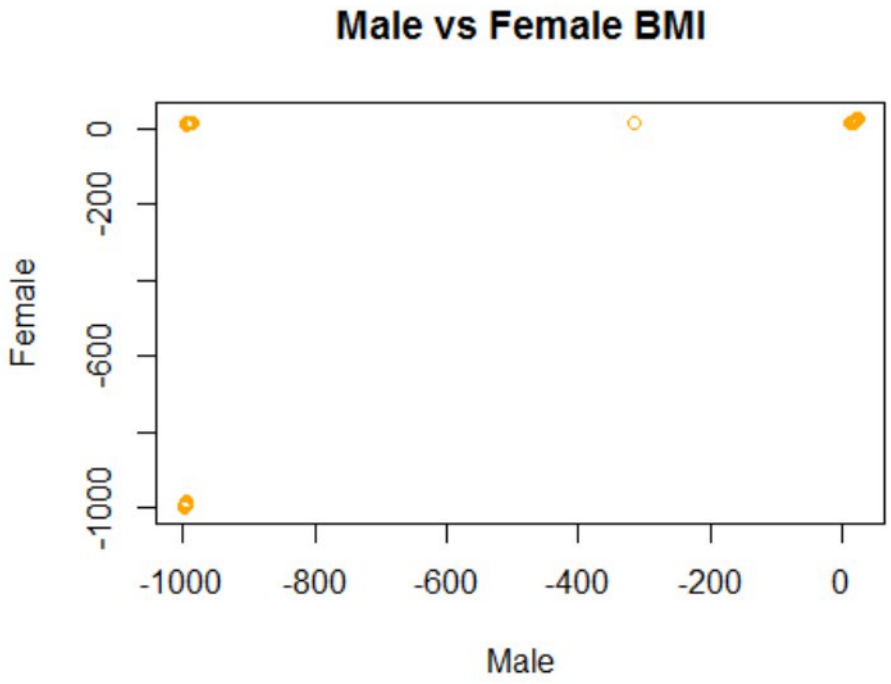

```
male <- subset(childData, genderdescription == "Male" & ageinmonths <= 65.60)
female <- subset(childData, genderdescription == "Female" & ageinmonths <=
65.60)

qqplot(male$height, female$height, main="Male vs Female Height", col="red",
xlab = deparse(substitute(Male)), ylab = deparse(substitute(Female)))
```




```
qqplot(male$weight, female$weight, main="Male vs Female Weight", col="green",
xlab = deparse(substitute(Male)), ylab = deparse(substitute(Female)))
```



```
qqplot(male$bmi, female$bmi, main="Male vs Female BMI", col="orange", xlab =
deparse(substitute(Male)), ylab = deparse(substitute(Female)))
```



It is important to note the categorization of underweight and overweight was taken from the CDC's website at: http://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/ Hence children with a BMI of less than 18.5 (< 18.5) were classified as underweight while those with thier BMI greater than 25 (>25) were categorized as overweight.

My sample age group for comparison is based on the fact that 143.30 is the maximum age and 65.60 is the median age(as can be seen below) hence I categoized the younger age group to be all ages from the minimum age (48.40) and up to/including the median age (65.60) while the older age group are those who fall into the category of being older than the median age (> 65.60)

```
print(summary(childData$ageinmonths))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   48.40   59.10   65.60   92.46  129.90  143.30
```

2. Fix the age (look only at one specific age - of exactly the same nr of months). Compare the distribution of height vs BMI using two qq-plots - one for boys, one for girls. (Hopefully different students will pick different age for this)

```
print(which(table(childData$ageinmonths) ==
max(table(childData$ageinmonths))))
```

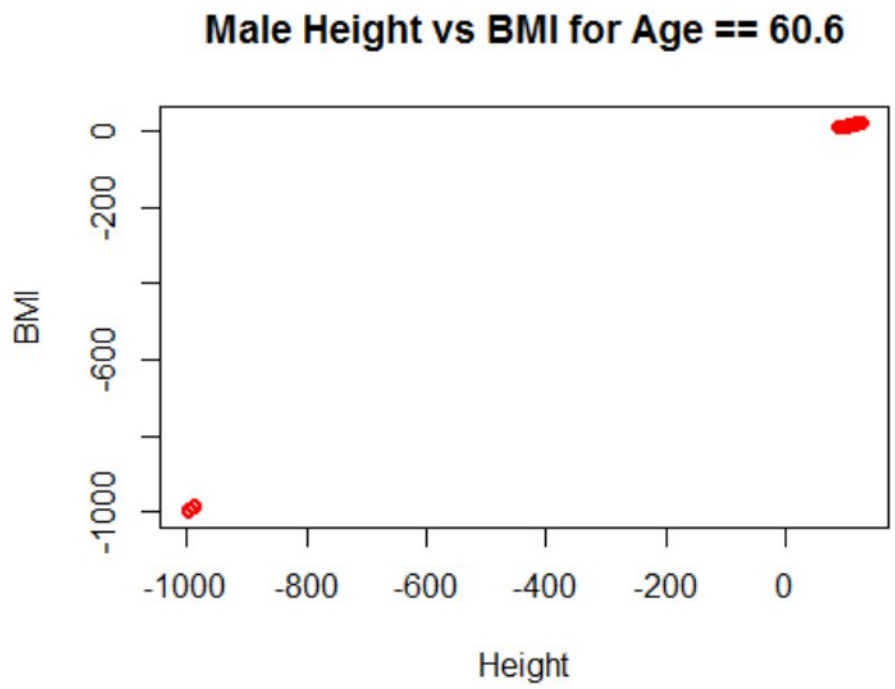
```
## 60.6
##  123
```

```
print(sort(table(childData$ageinmonths)[table(childData$ageinmonths) >
6000]))
```

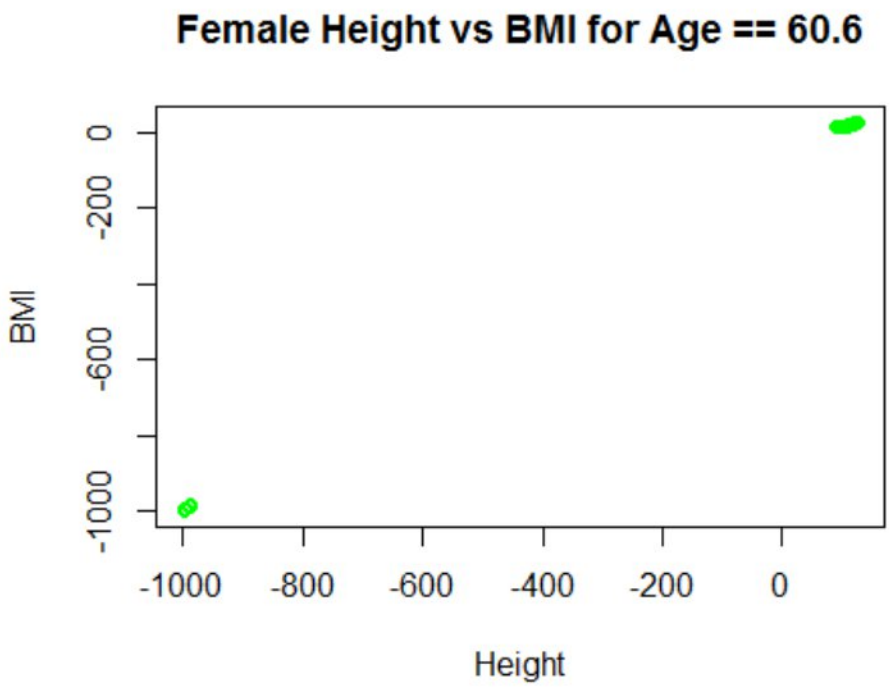
```
##
## 59.3 59.8 58.3 60.6
## 6119 6157 7194 7380
```

As can be seen from the statistics above, 60.6 happens to be the most frequent age distribution in the entire data set hence I will adopt this as the age to use for the comparison.

```
malebyage <- subset(childData, genderdescription == "Male" & ageinmonths <=
60.6)
femalebyage <- subset(childData, genderdescription == "Female" & ageinmonths
<= 60.6)


qqplot(malebyage$height, malebyage$bmi, main="Male Height vs BMI for Age ==
60.6", col="red", xlab = deparse(substitute(Height)), ylab =
deparse(substitute(BMI)))
```

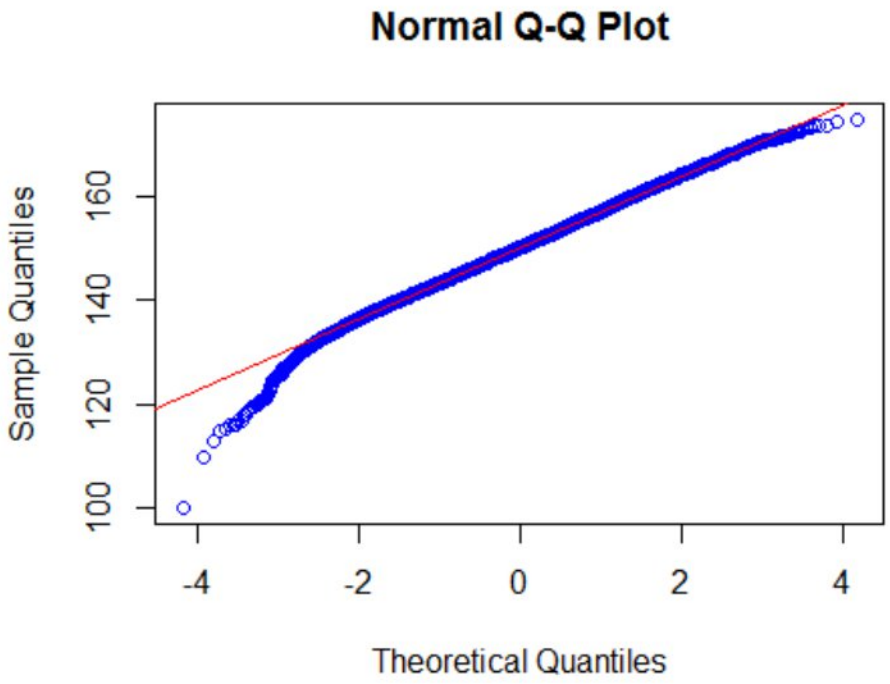



```
qqplot(femalebyage$height, femalebyage$bmi, main="Female Height vs BMI for
Age == 60.6", col="green", xlab = deparse(substitute(Height)), ylab =
deparse(substitute(BMI)))
```



3. Compare the height of overweight children against theoretical normal distribution. You can limit to certain age group and gender.

```
qqnorm(overweight$height, col="blue")
qqline(overweight$height, col="red")
```



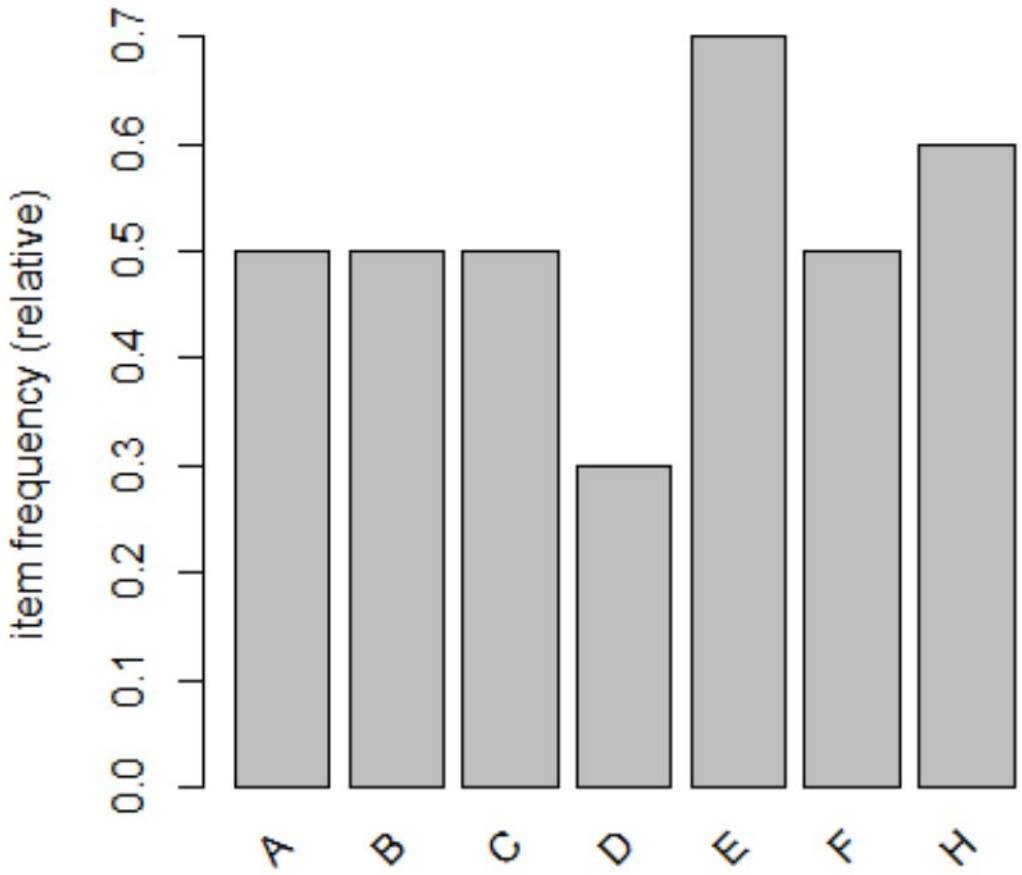
For this task I decided to take the older age grade for the over weight kids (ageinmonths > 65.60) because for the previous task (task two) I had used the younger age group

4. Follow the apriori algorithm principle and enumerate all itemsets that have support of 0.3 or higher, provide support. (probably best to solve using pen and paper or simple text editor and Unix command line tools)

```
library(arules)

## Warning: package 'arules' was built under R version 3.2.4

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##     %in%, abbreviate, write

abcset <- read.transactions("C:/Users/Kenigbolo PC/Desktop/Data
Mining/abcset.csv", rm.duplicates= FALSE, format="basket", sep=",")
inspect(abcset)

##    items
## 1  {A,B,C,F,H}
## 2  {C,E,F,H}
## 3  {B,D,E}
```

```
## 4  {A,C,F,H}
## 5  {A,E,F}
## 6  {B,D,H}
## 7  {B,C,D,E,F}
## 8  {A,C,E,H}
## 9  {A,E,G}
## 10 {B,E,H}
```

```
itemFrequencyPloti(itemFrequencyPlot (abcset, support = 0.3), support = 0.3)
```



5. Calculate the support and confidence for every possible association rule from the above example where there is exactly one item on the left and one item on the right (e.g. A->E). Make two 8x8 tables (A..H) x (A..H), one for support and the other for confidence. Be clever, create some simple script for calculating this. Color these as heatmap (e.g. in Excel)

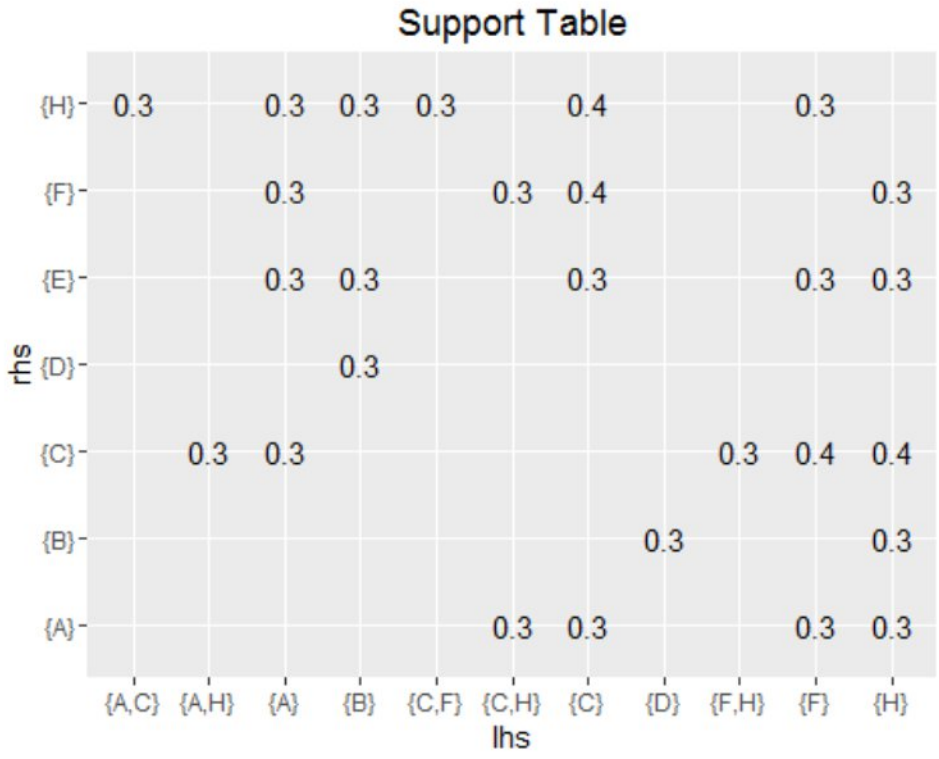
Which rules are "most interesting" from 5 based on those data?

```
library(arulesViz)
```

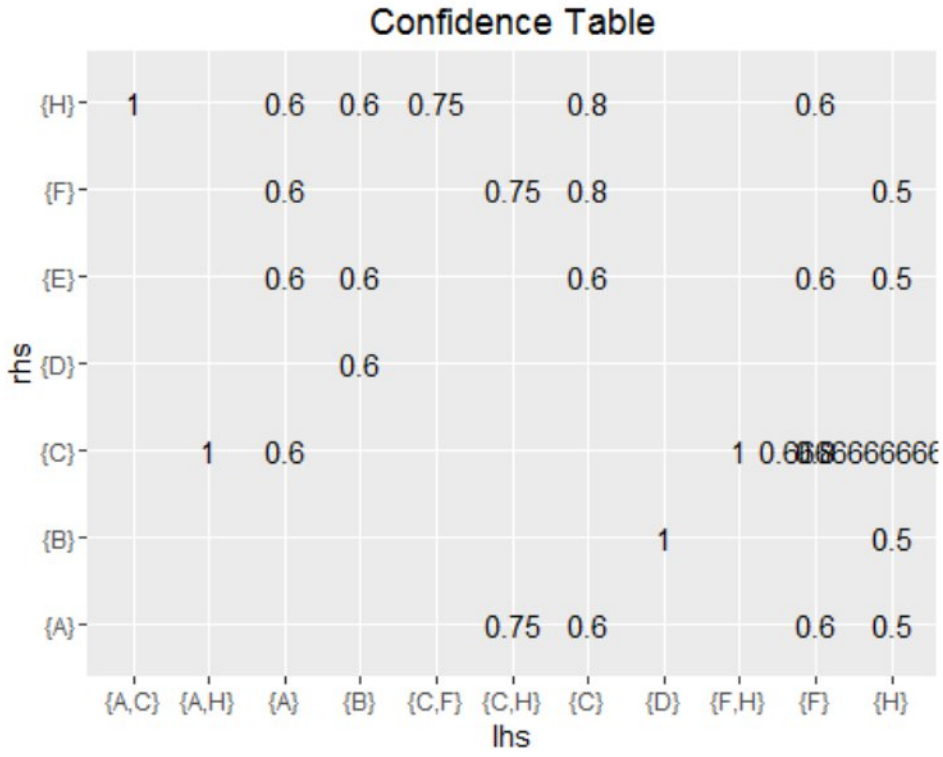
```
## Warning: package 'arulesViz' was built under R version 3.2.4

## Loading required package: grid

library(qualityTools)

## Warning: package 'qualityTools' was built under R version 3.2.4

## Loading required package: Rsolnp

## Loading required package: MASS

library(ggplot2)
abcset <-read.transactions("C:/Users/Kenigbolo PC/Desktop/Data
Mining/abcset.csv",rm.duplicates= FALSE, format="basket", sep=",")
abcsetsupport <- apriori(abcset, parameter= list(supp=0.3, conf=0.5))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##         0.5    0.1    1 none FALSE            TRUE     0.3      1     10
##  target   ext
##   rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 3
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[8 item(s), 10 transaction(s)] done [0.00s].
## sorting and recoding items ... [7 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [33 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

abcsetdata <- inspect(abcsetsupport)

##    lhs      rhs support confidence lift
## 1  {}    => {B} 0.5     0.5000000  1.0000000
## 2  {}    => {A} 0.5     0.5000000  1.0000000
## 3  {}    => {F} 0.5     0.5000000  1.0000000
## 4  {}    => {C} 0.5     0.5000000  1.0000000
## 5  {}    => {H} 0.6     0.6000000  1.0000000
## 6  {}    => {E} 0.7     0.7000000  1.0000000
## 7  {D}   => {B} 0.3     1.0000000  2.0000000
## 8  {B}   => {D} 0.3     0.6000000  2.0000000
## 9  {B}   => {H} 0.3     0.6000000  1.0000000
## 10 {H}   => {B} 0.3     0.5000000  1.0000000
## 11 {B}   => {E} 0.3     0.6000000  0.8571429
```

```
## 12 {A}   => {F} 0.3     0.6000000  1.2000000
## 13 {F}   => {A} 0.3     0.6000000  1.2000000
## 14 {A}   => {C} 0.3     0.6000000  1.2000000
## 15 {C}   => {A} 0.3     0.6000000  1.2000000
## 16 {A}   => {H} 0.3     0.6000000  1.0000000
## 17 {H}   => {A} 0.3     0.5000000  1.0000000
## 18 {A}   => {E} 0.3     0.6000000  0.8571429
## 19 {F}   => {C} 0.4     0.8000000  1.6000000
## 20 {C}   => {F} 0.4     0.8000000  1.6000000
## 21 {F}   => {H} 0.3     0.6000000  1.0000000
## 22 {H}   => {F} 0.3     0.5000000  1.0000000
## 23 {F}   => {E} 0.3     0.6000000  0.8571429
## 24 {C}   => {H} 0.4     0.8000000  1.3333333
## 25 {H}   => {C} 0.4     0.6666667  1.3333333
## 26 {C}   => {E} 0.3     0.6000000  0.8571429
## 27 {H}   => {E} 0.3     0.5000000  0.7142857
## 28 {A,C} => {H} 0.3     1.0000000  1.6666667
## 29 {A,H} => {C} 0.3     1.0000000  2.0000000
## 30 {C,H} => {A} 0.3     0.7500000  1.5000000
## 31 {C,F} => {H} 0.3     0.7500000  1.2500000
## 32 {F,H} => {C} 0.3     1.0000000  2.0000000
## 33 {C,H} => {F} 0.3     0.7500000  1.5000000
```

```
abcsetdata <- abcsetdata[-c(1,2,3,4,5,6), ]
ggplot(aes(lhs, rhs),
data=abcsetdata)+geom_text(aes(label=abcsetdata[,4]))+labs(title="Support
Table")
```

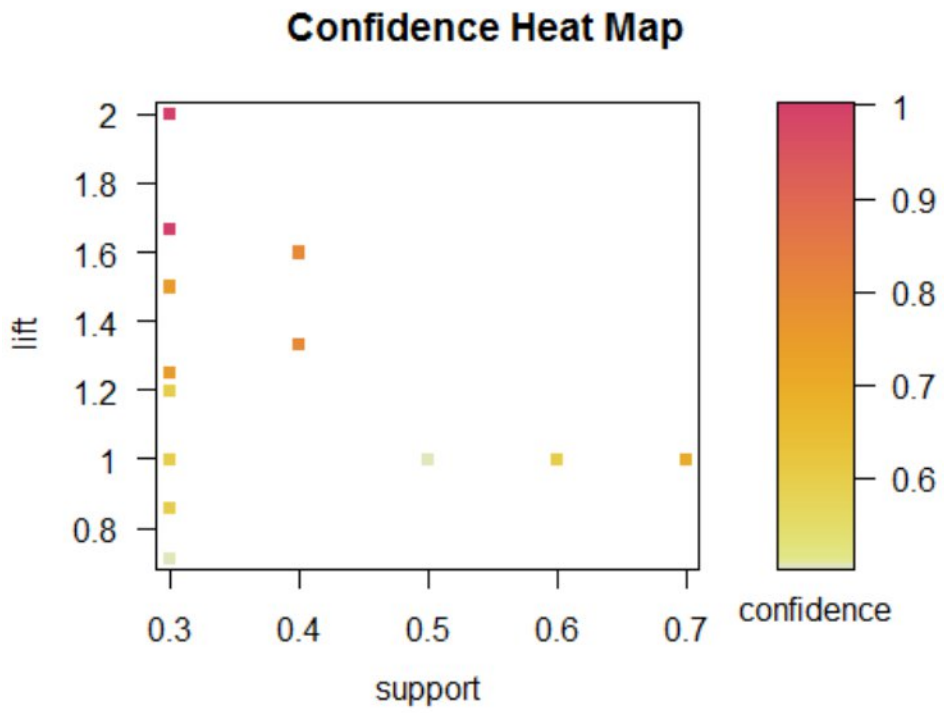


```
ggplot(aes(lhs, rhs),
data=abcsetdata)+geom_text(aes(label=abcsetdata[,5]))+labs(title="Confidence
Table")
```

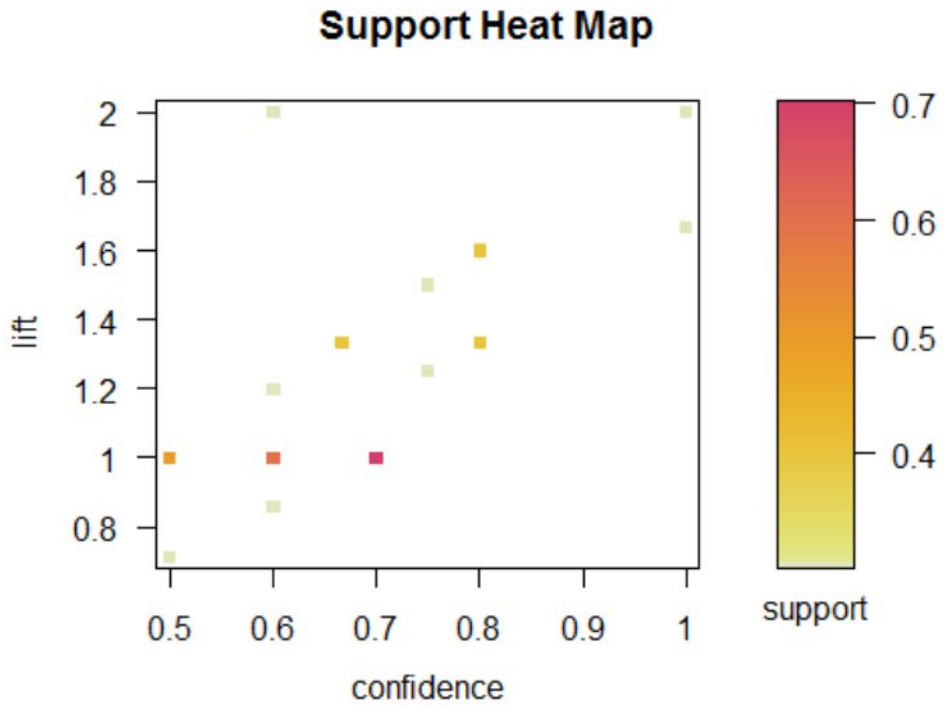


In the above association rules and tables, I was unable to make a specific 8x8 table rule after trying multiple confidence numbers mostly because of the abscence of "G" however the above tables represent both support and confidence.

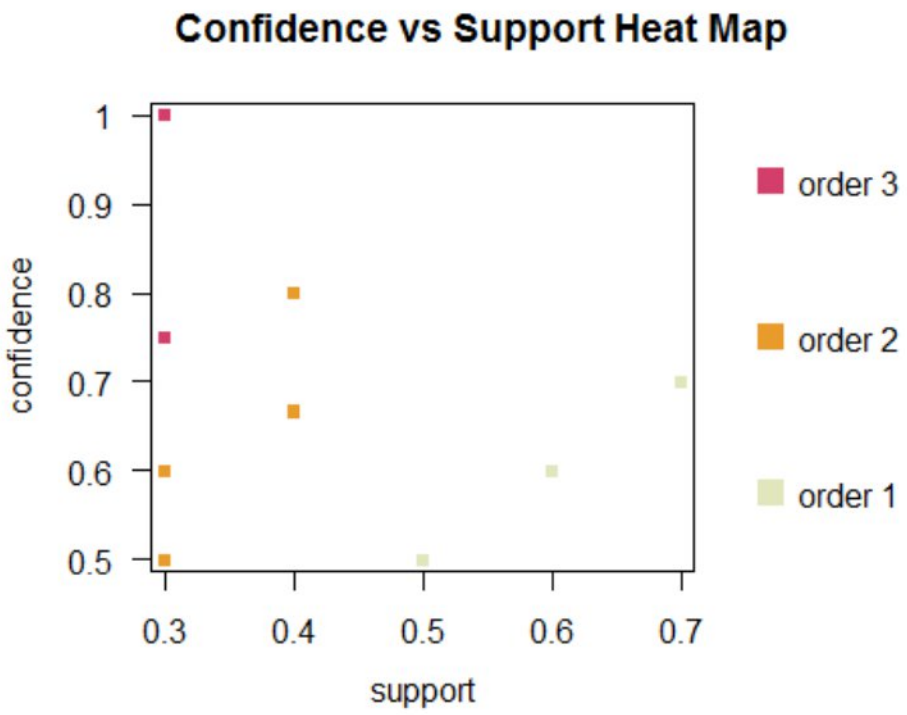
```
library(arulesViz)
plot(abcsetsupport, measure=c("support", "lift"), shading="confidence",
control=list(main = "Confidence Heat Map"))
```




```
plot(abcsetsupport, measure=c("confidence", "lift"), shading="support",
control=list(main = "Support Heat Map"))
```



```
plot(abcsetsupport, shading="order", control=list(main = "Confidence vs
Support Heat Map"))
```



The rules that were the most interesting for me were the following

		Support	Confidence	
## 19	{F} => {C}	0.4	0.8000000	1.6000000
## 20	{C} => {F}	0.4	0.8000000	1.6000000
## 7	{D} => {B}	0.3	1.0000000	2.0000000
## 24	{C} => {H}	0.4	0.8000000	1.3333333
## 28	{A,C} => {H}	0.3	1.0000000	1.6666667
## 29	{A,H} => {C}	0.3	1.0000000	2.0000000
## 32	{F,H} => {C}	0.3	1.0000000	2.0000000

It seems that the lower the support the higher the confidence