ASSOCIATION ANALYSIS

Saloni Mishra

11/2/2020

Loading Libraries

arules library has been used in this association analysis.

Data Processing

After reading the data, I got shape of the data i.e., 3333 instances with 21 variables. After initial exploration, I subset the data and got the four variables i.e., VMail Plan, Intl Plan, CustServ Calls, and Churn (as per the assignment requirement). VMail Plan, Intl Plan and Churn are categorical variables. Although CustServ Calls are ordinal.

```
setwd("C:\\Users\\salon\\Desktop\\DATA SCIENCE\\Predictive Analytics\\week7")
df=read.csv("data.csv")
#View(df1)
df1<-df[, c(5,6,20,21)]
str(df1)
```

```
## 'data.frame':    3333 obs. of  4 variables:
##  $ Int.l.Plan    : chr  "no" "no" "no" "yes" ...
##  $ VMail.Plan    : chr  "yes" "yes" "no" "no" ...
##  $ CustServ.Calls: int  1 1 0 2 3 0 3 0 1 0 ...
##  $ Churn.        : chr  "False." "False." "False." "False." ...
```

```
df2<-df[, c(5,6,20,21)]
colSums(is.na(df1)) ## detecting null values
```

```
##     Int.l.Plan     VMail.Plan CustServ.Calls         Churn.
##              0              0              0              0
```

Descriptive Statistics of All Variables

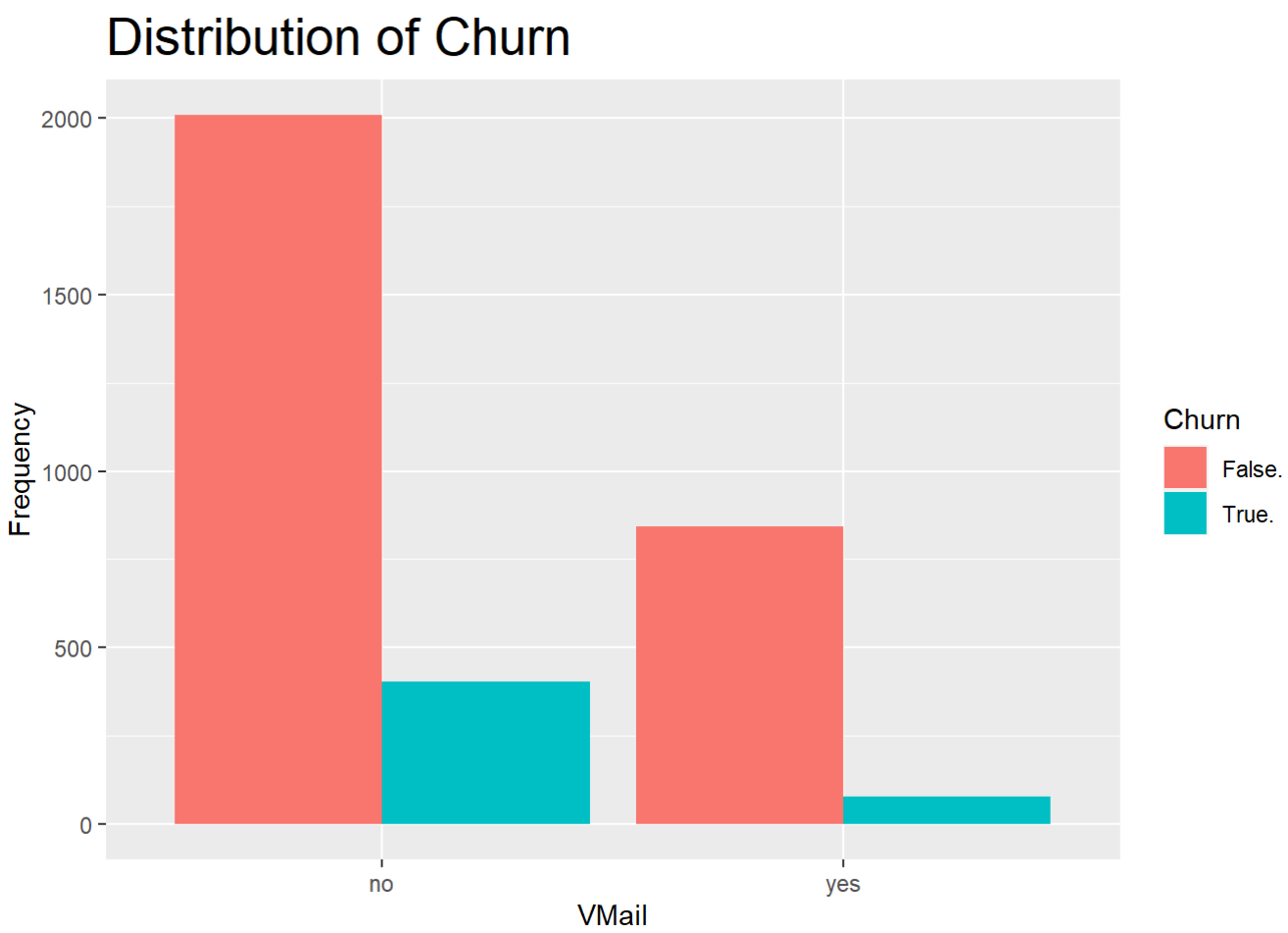
```
summary(df1) ## descriptive Statistics of All variable
```

```
##   Int.l.Plan         VMail.Plan        CustServ.Calls     Churn.
##  Length:3333        Length:3333        Min.   :0.000   Length:3333
##  Class :character   Class :character   1st Qu.:1.000   Class :character
##  Mode  :character   Mode  :character   Median :1.000   Mode  :character
##                                        Mean   :1.563
##                                        3rd Qu.:2.000
##                                        Max.   :9.000
```

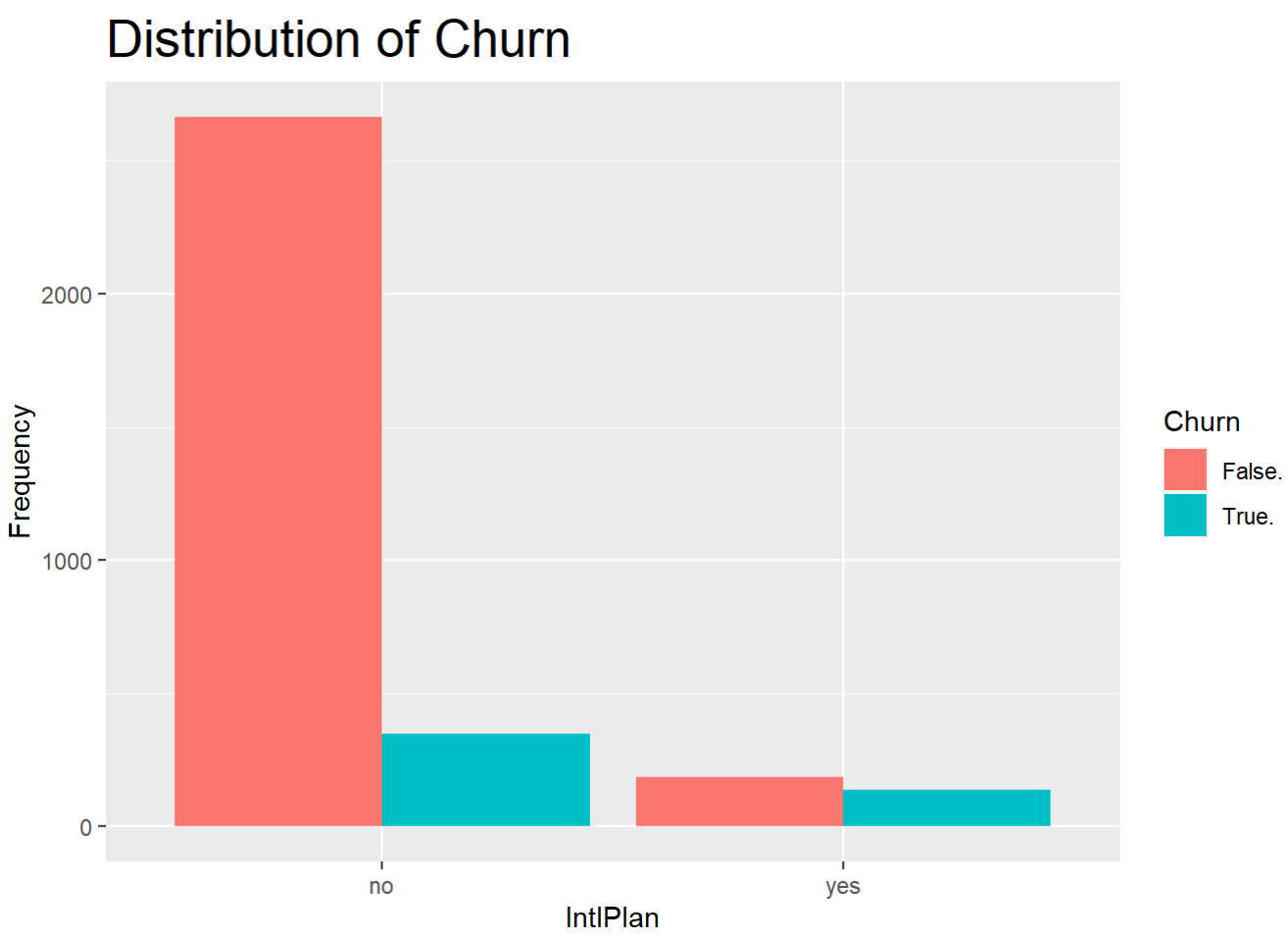
Exploratory Data Analysis

In EDA, I used doughnut plot to explore how Churn variable is distributed in between yes(Churned) and no(not Churned). I found that in this data 85.5% are not churned where as 14.5% are churned. To understand the distribution of the variables with respect to Churn is explored as below. I found that the very less number of people with internation plan has churned. Moreover, people having the VMail plan doesn't have a big affect in churning. People who called customer service atleast once have most churned customers indicating bad customer service. Although most people called atleast once.

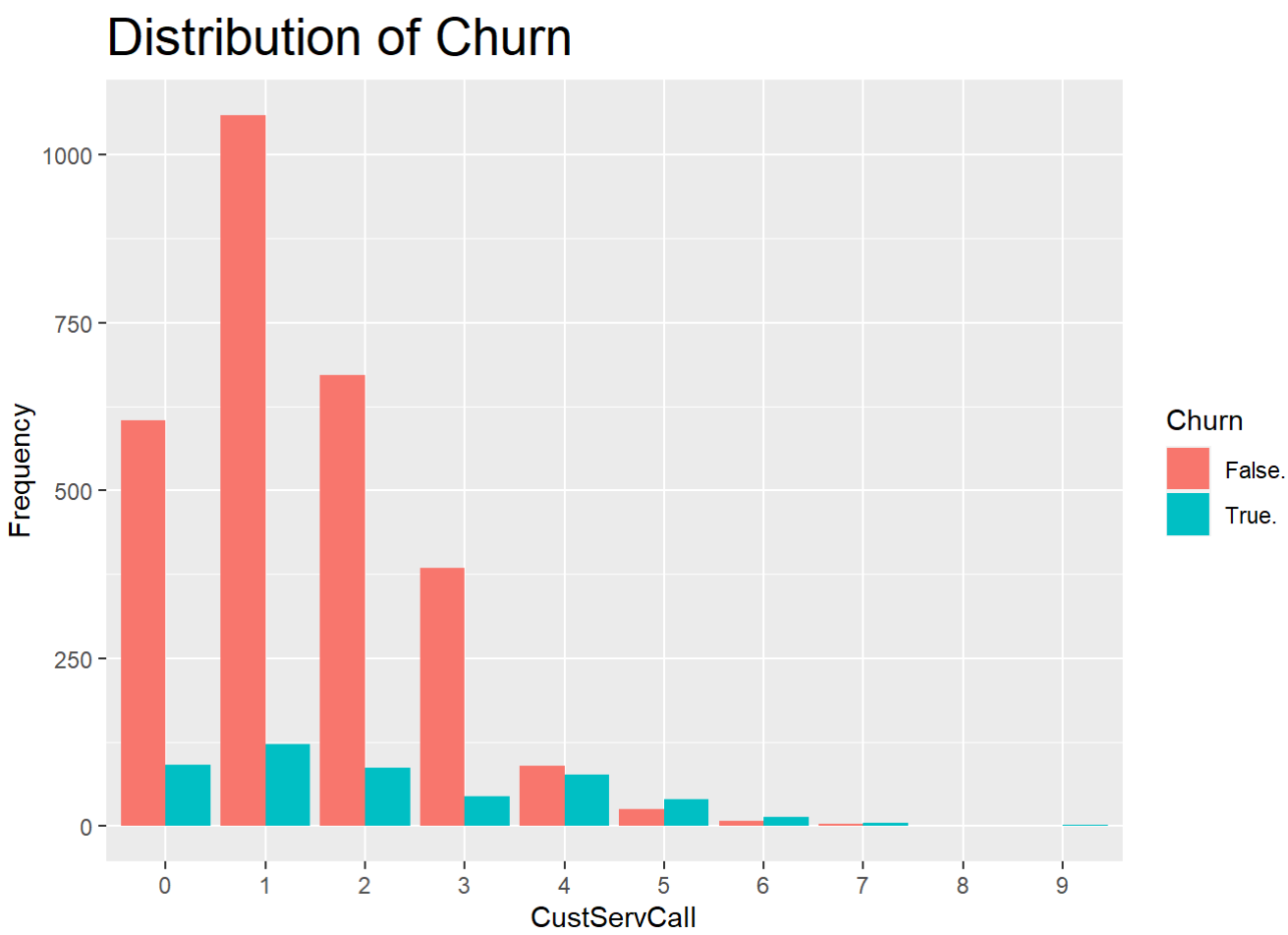
```
plot1<-data.frame(table(df1$'VMail.Plan', df1$'Churn.'))
names(plot1)<-c("VMail", "Churn", "Frequency")
ggplot(data=plot1,aes(x=VMail, y=Frequency, fill=Churn))+
  geom_bar(stat="identity", position="dodge")+
  ggtitle("Distribution of Churn")+
  theme(plot.title = element_text(size = 20))
```




```
plot2<-data.frame(table(df1$'Int.l.Plan', df1$'Churn.'))
names(plot2)<-c("IntlPlan", "Churn", "Frequency")
ggplot(data=plot2,aes(x=IntlPlan, y=Frequency, fill=Churn))+
  geom_bar(stat="identity", position="dodge")+
  ggtitle("Distribution of Churn")+
  theme(plot.title = element_text(size = 20))
```



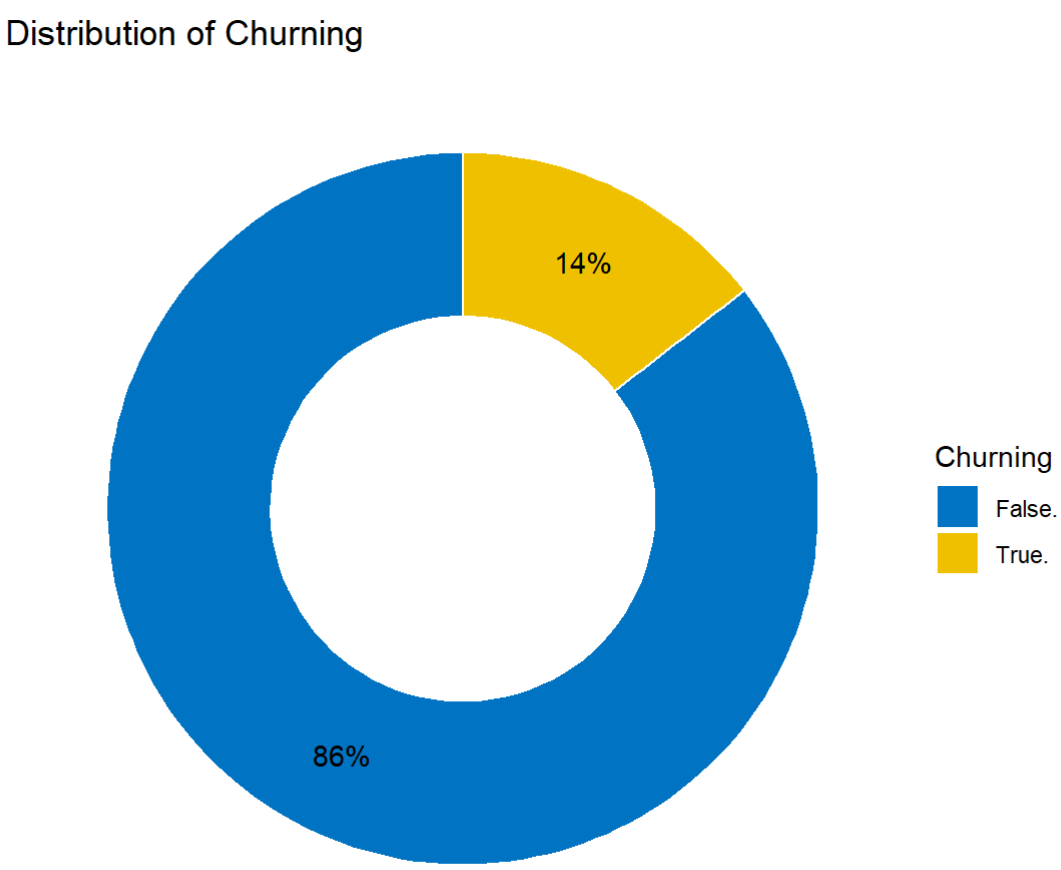

```
plot3<-data.frame(table(df1$'CustServ.Calls', df1$'Churn.'))
names(plot3)<-c("CustServCall", "Churn", "Frequency")
ggplot(data=plot3,aes(x=CustServCall, y=Frequency, fill=Churn))+
  geom_bar(stat="identity", position="dodge")+
  ggtitle("Distribution of Churn")+
  theme(plot.title = element_text(size = 20))
```




```
plot7<-data.frame(table(df1$'Churn.'))
names(plot7)<-c("Churning", "Frequency")
plot7$prop<-plot7$Frequency/sum(plot7$Frequency)
head(plot7)
```

```
##   Churning Frequency      prop
## 1   False.      2850 0.8550855
## 2    True.       483 0.1449145
```

```
mycols <- c("#0073C2FF", "#EFC000FF")
ggplot(plot7, aes(x = 2, y = prop, fill = Churning)) +
  geom_bar(stat = "identity", color = "white") +
  coord_polar(theta = "y", start = 0)+
  geom_text(aes(label=paste0(round(prop*100), "%")), position=position_stack(vjust=0.5))+
  scale_fill_manual(values = mycols) +
  theme_void()+
  xlim(0.5, 2.5)+
  labs(x=NULL, y=NULL, title="Distribution of Churning")
```



Association Analysis

Association rules analysis is a technique to unveil how one item is associated with other in a large data set. Apriori is an algorithm used to explore association analysis. We use below measures to understand the association.

* Support is a measure of how frequently the itemset appears in the dataset.

* Confidence is a measure of how often the rule has been found to be true. For example, for a rule A=>B the percentage in which B is bought with A is Confidence.

* Lift is the ratio of the observed support to that expected if the two rules were independent

A high support means that the rule occurs very frequently, while a high confidence indicates that the rule has a high predictive power. Association rule is expected to show a strong relationship between the items.

As per the assignment requirement, the minimum antecedent support is set to 1%, the minimum rule confidence is set to 5%, and the maximum number of antecedents to 2(max_length) as giving 1 was not taking antecedents.

From the analysis, I found highest lift is 4.18 for rule CustomerServ Calls=5 then Churn =True with support 1% and confidence 60%. Although, I was not able to remove the Churn variable from antecedent.I got 10 rules with highest lift as below.

In this library, we have appearance with 4 different options: lhs,rhs,none,items. From this we can select which variable we want to take on which side or don't want to take but removing one variable from lhs.

```
df1$Churn.<-as.factor(df1$Churn.)
df1[sapply(df1, is.character)] <- lapply(df1[sapply(df1, is.character)],
                                         to_factor)
df1$CustServ.Calls=to_factor(df1$CustServ.Calls)
str(df1)
```

```
## 'data.frame':    3333 obs. of  4 variables:
##  $ Int.l.Plan    : Factor w/ 2 levels "no","yes": 1 1 1 2 2 2 1 2 1 2 ...
##  $ VMail.Plan    : Factor w/ 2 levels "no","yes": 2 2 1 1 1 1 2 1 1 2 ...
##  $ CustServ.Calls: Factor w/ 10 levels "0","1","2","3",..: 2 2 1 3 4 1 4 1 2 1 ...
##  $ Churn.        : Factor w/ 2 levels "False.","True.": 1 1 1 1 1 1 1 1 1 1 ...
```

```
df1<-data.frame(df1)
ap1<-apriori(df1, parameter = list(support = 0.01, confidence = 0.05, maxlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##       2  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 33
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3333 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [93 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(ap1, by = "lift"), 10))
```

```
##      lhs                    rhs                 support    confidence coverage
## [1]  {CustServ.Calls=5} => {Churn.=True.}      0.01200120 0.60606061 0.01980198
## [2]  {Churn.=True.}     => {CustServ.Calls=5} 0.01200120 0.08281573 0.14491449
## [3]  {CustServ.Calls=4} => {Churn.=True.}      0.02280228 0.45783133 0.04980498
## [4]  {Churn.=True.}     => {CustServ.Calls=4} 0.02280228 0.15734990 0.14491449
## [5]  {Int.l.Plan=yes}   => {Churn.=True.}      0.04110411 0.42414861 0.09690969
## [6]  {Churn.=True.}     => {Int.l.Plan=yes}   0.04110411 0.28364389 0.14491449
## [7]  {Int.l.Plan=yes}   => {CustServ.Calls=0} 0.02490249 0.25696594 0.09690969
## [8]  {CustServ.Calls=0} => {Int.l.Plan=yes}   0.02490249 0.11908178 0.20912091
## [9]  {VMail.Plan=no}    => {Churn.=True.}      0.12091209 0.16715056 0.72337234
## [10] {Churn.=True.}     => {VMail.Plan=no}    0.12091209 0.83436853 0.14491449
##      lift     count
## [1]  4.182195  40
## [2]  4.182195  40
## [3]  3.159321  76
## [4]  3.159321  76
## [5]  2.926889 137
## [6]  2.926889 137
## [7]  1.228791  83
## [8]  1.228791  83
## [9]  1.153443 403
## [10] 1.153443 403
```

In second model, I selected all variables except Churn to be on lhs. From this, I got the same result as above but Churn variable is not in lhs. Also other variables are not in rhs by default.

```
ap<-apriori(df1, parameter = list(support = 0.01, confidence = 0.05, maxlen=2),
            appearance = list(default="rhs",lhs=c("VMail.Plan=yes","VMail.Plan=no",                          "Int.l.Plan=y
es", "Int.l.Plan=no", "CustServ.Calls=0",
                          "CustServ.Calls=1","CustServ.Calls=2", "CustServ.Calls=3",
                          "CustServ.Calls=4","CustServ.Calls=5")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##       2  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 33
##
## set item appearances ...[10 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3333 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2
```

```
## Warning in apriori(df1, parameter = list(support = 0.01, confidence = 0.05, :
## Mining stopped (maxlen reached). Only patterns up to a length of 2 returned!
```

```
##  done [0.00s].
## writing ... [21 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(ap, by = "lift"), 10))
```

```
##      lhs                    rhs             support    confidence coverage
## [1]  {CustServ.Calls=5} => {Churn.=True.}  0.01200120 0.6060606  0.01980198
## [2]  {CustServ.Calls=4} => {Churn.=True.}  0.02280228 0.4578313  0.04980498
## [3]  {Int.l.Plan=yes}   => {Churn.=True.}  0.04110411 0.4241486  0.09690969
## [4]  {VMail.Plan=no}    => {Churn.=True.}  0.12091209 0.1671506  0.72337234
## [5]  {VMail.Plan=yes}   => {Churn.=False.} 0.25262526 0.9132321  0.27662766
## [6]  {CustServ.Calls=3} => {Churn.=False.} 0.11551155 0.8974359  0.12871287
## [7]  {CustServ.Calls=1} => {Churn.=False.} 0.31773177 0.8966977  0.35433543
## [8]  {CustServ.Calls=2} => {Churn.=False.} 0.20162016 0.8853755  0.22772277
## [9]  {Int.l.Plan=no}    => {Churn.=False.} 0.79927993 0.8850498  0.90309031
## [10] {CustServ.Calls=0} => {Churn.=False.} 0.18151815 0.8680057  0.20912091
##      lift     count
## [1]  4.182195   40
## [2]  3.159321   76
## [3]  2.926889  137
## [4]  1.153443  403
## [5]  1.068001  842
## [6]  1.049528  385
## [7]  1.048664 1059
## [8]  1.035423  672
## [9]  1.035042 2664
## [10] 1.015110  605
```

Model 3, with rhs Churn with value True only as below.On decreasing order of lift, I got rules as; CustServ Calls=5, CustServ Calls=4, Int'l Plan=yes, VMail Plan=no for Churn =true. Other rules are also given below.

```
ap2<-apriori(df1, parameter = list(support = 0.01, confidence = 0.05, maxlen=2),
            appearance = list(default="lhs",rhs=c("Churn.=True.")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##       2  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 33
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[16 item(s), 3333 transaction(s)] done [0.00s].
## sorting and recoding items ... [12 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2
```

```
## Warning in apriori(df1, parameter = list(support = 0.01, confidence = 0.05, :
## Mining stopped (maxlen reached). Only patterns up to a length of 2 returned!
```

```
##  done [0.00s].
## writing ... [11 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(ap2, by = "lift"), 10))
```

```
##      lhs                   rhs            support    confidence coverage
## [1]  {CustServ.Calls=5} => {Churn.=True.} 0.01200120 0.6060606  0.01980198
## [2]  {CustServ.Calls=4} => {Churn.=True.} 0.02280228 0.4578313  0.04980498
## [3]  {Int.l.Plan=yes}   => {Churn.=True.} 0.04110411 0.4241486  0.09690969
## [4]  {VMail.Plan=no}    => {Churn.=True.} 0.12091209 0.1671506  0.72337234
## [5]  {}                 => {Churn.=True.} 0.14491449 0.1449145  1.00000000
## [6]  {CustServ.Calls=0} => {Churn.=True.} 0.02760276 0.1319943  0.20912091
## [7]  {Int.l.Plan=no}    => {Churn.=True.} 0.10381038 0.1149502  0.90309031
## [8]  {CustServ.Calls=2} => {Churn.=True.} 0.02610261 0.1146245  0.22772277
## [9]  {CustServ.Calls=1} => {Churn.=True.} 0.03660366 0.1033023  0.35433543
## [10] {CustServ.Calls=3} => {Churn.=True.} 0.01320132 0.1025641  0.12871287
##      lift      count
## [1]  4.1821946  40
## [2]  3.1593205  76
## [3]  2.9268888 137
## [4]  1.1534427 403
## [5]  1.0000000 483
## [6]  0.9108424  92
## [7]  0.7932275 346
## [8]  0.7909803  87
## [9]  0.7128499 122
## [10] 0.7077560  44
```

Fourth model, as I took CustServ Call as ordinal but in this I categorized as below. In the rhs, I used churn true/false and lhs as default.Highest lift is 2.92 with 4% support and 42% confidence with rule Int'l Plan=yes then Churn=True.As, for customer service calls variable I took median as medium category above median low and below high. For rule 2 CustServ Calls=High then churn=True with support 5% and confidence 26% with lift 1.8.

```
df2[[ "CustServ.Calls"]] <- ordered(cut(df2[[ "CustServ.Calls"]],
        c(-Inf,0,median(df2[[ "CustServ.Calls"]][df2[["CustServ.Calls"]]>0]),Inf)),
                                           labels = c("low", "medium", "High"))
ap3<-apriori(df2, parameter = list(support = 0.01, confidence = 0.05, maxlen=2),
             appearance = list(default="lhs",rhs=c("Churn.=True.", "Churn.=False.")))
```

```
## Warning: Column(s) 1, 2, 4 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.05    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##       2  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 33
##
## set item appearances ...[2 item(s)] done [0.00s].
## set transactions ...[9 item(s), 3333 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2
```

```
## Warning in apriori(df2, parameter = list(support = 0.01, confidence = 0.05, :
## Mining stopped (maxlen reached). Only patterns up to a length of 2 returned!
```

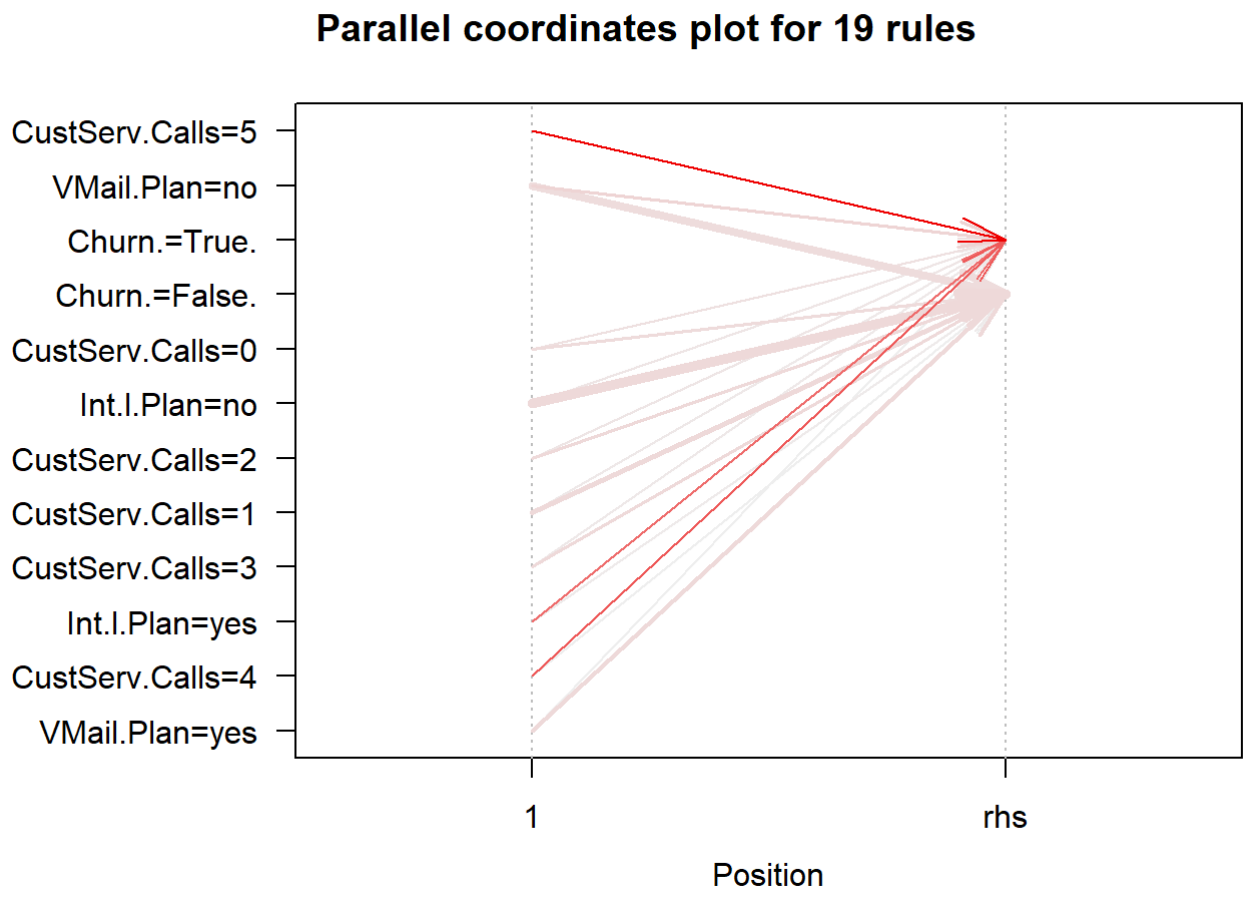
```
##  done [0.00s].
## writing ... [16 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(ap3, by = "lift"), 10))
```

```
##      lhs                          rhs             support    confidence
## [1]  {Int.l.Plan=yes}        => {Churn.=True.}  0.04110411 0.4241486
## [2]  {CustServ.Calls=High}   => {Churn.=True.}  0.05460546 0.2614943
## [3]  {VMail.Plan=no}         => {Churn.=True.}  0.12091209 0.1671506
## [4]  {VMail.Plan=yes}        => {Churn.=False.} 0.25262526 0.9132321
## [5]  {CustServ.Calls=medium} => {Churn.=False.} 0.51935194 0.8922680
## [6]  {Int.l.Plan=no}         => {Churn.=False.} 0.79927993 0.8850498
## [7]  {CustServ.Calls=low}    => {Churn.=False.} 0.18151815 0.8680057
## [8]  {}                      => {Churn.=True.}  0.14491449 0.1449145
## [9]  {}                      => {Churn.=False.} 0.85508551 0.8550855
## [10] {VMail.Plan=no}         => {Churn.=False.} 0.60246025 0.8328494
##      coverage   lift      count
## [1]  0.09690969 2.9268888  137
## [2]  0.20882088 1.8044728  182
## [3]  0.72337234 1.1534427  403
## [4]  0.27662766 1.0680009  842
## [5]  0.58205821 1.0434840 1731
## [6]  0.90309031 1.0350425 2664
## [7]  0.20912091 1.0151099  605
## [8]  1.00000000 1.0000000  483
## [9]  1.00000000 1.0000000 2850
## [10] 0.72337234 0.9739955 2008
```

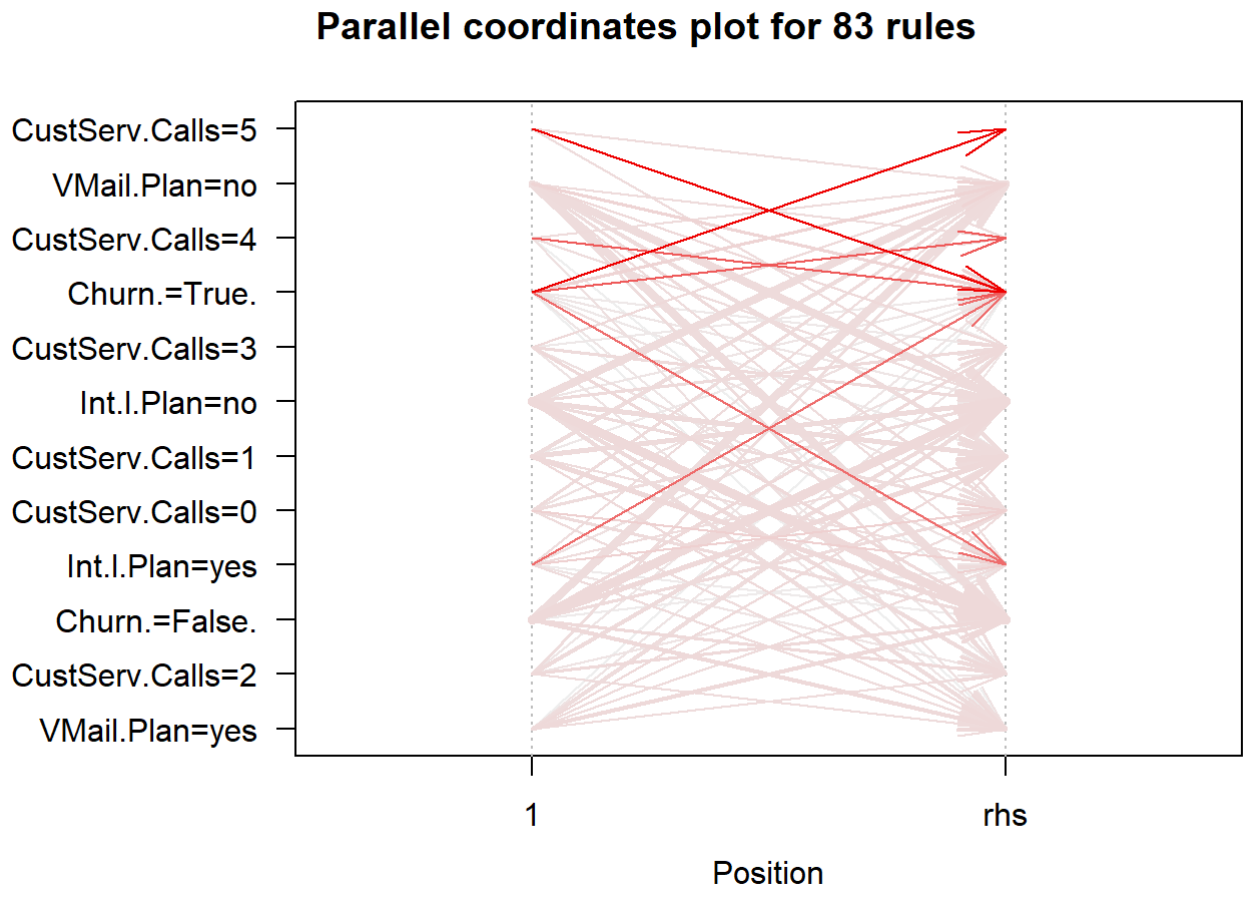
From the first model, I plotted parallel coordinates which is suggesting for Churn=yes, Custserv Calls=4, 5 and Intl Plan is yes whereas for churn=false VMail plan and Intl Plan is no.

```
plot(ap, method="paracoord", control=list(reorder=TRUE))
```



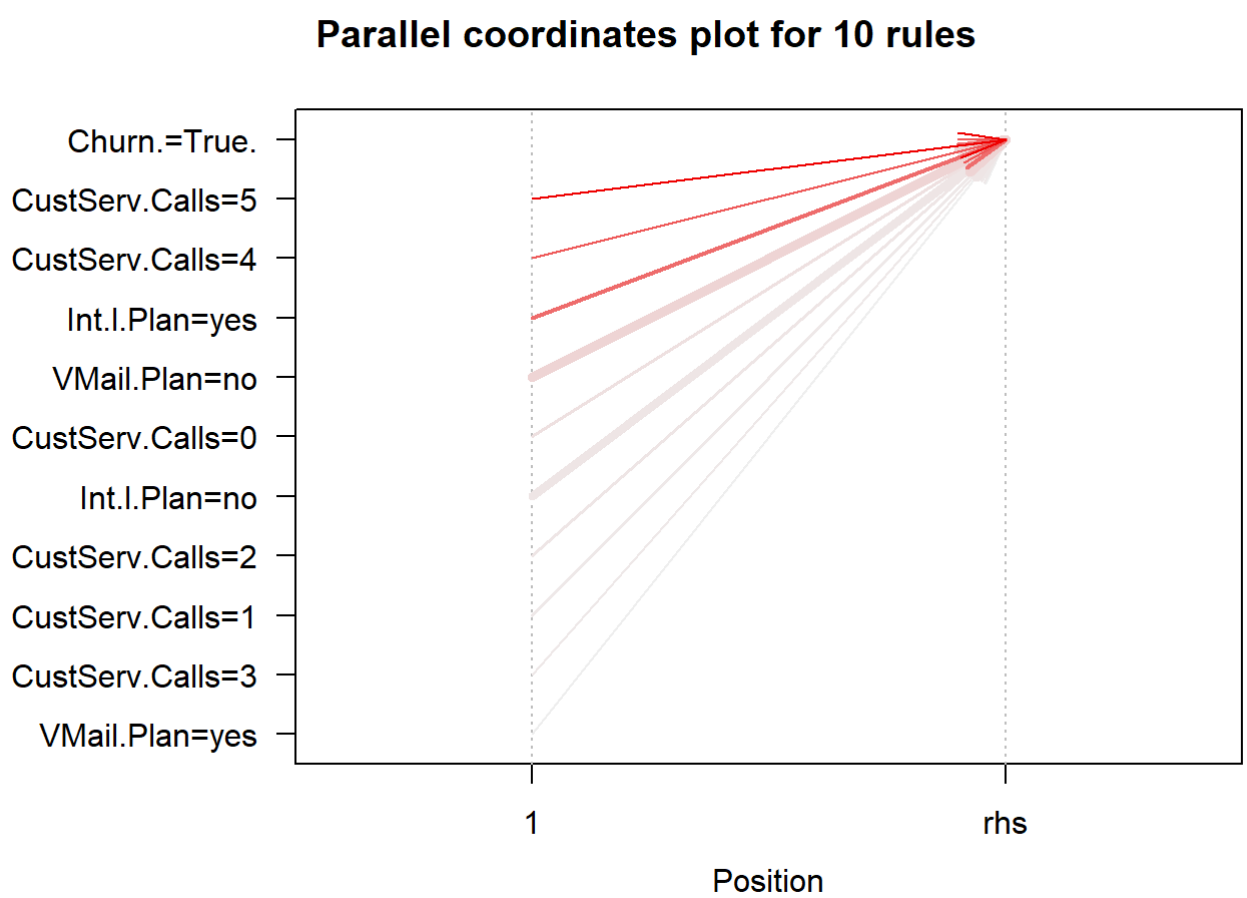
From model 2, if we take churn=true in antecedent it’s showing consequent will be CustServ Call 4, 5 and Intl Plan is yes.

```
plot(ap1, method="paracoord", control=list(reorder=TRUE))
```



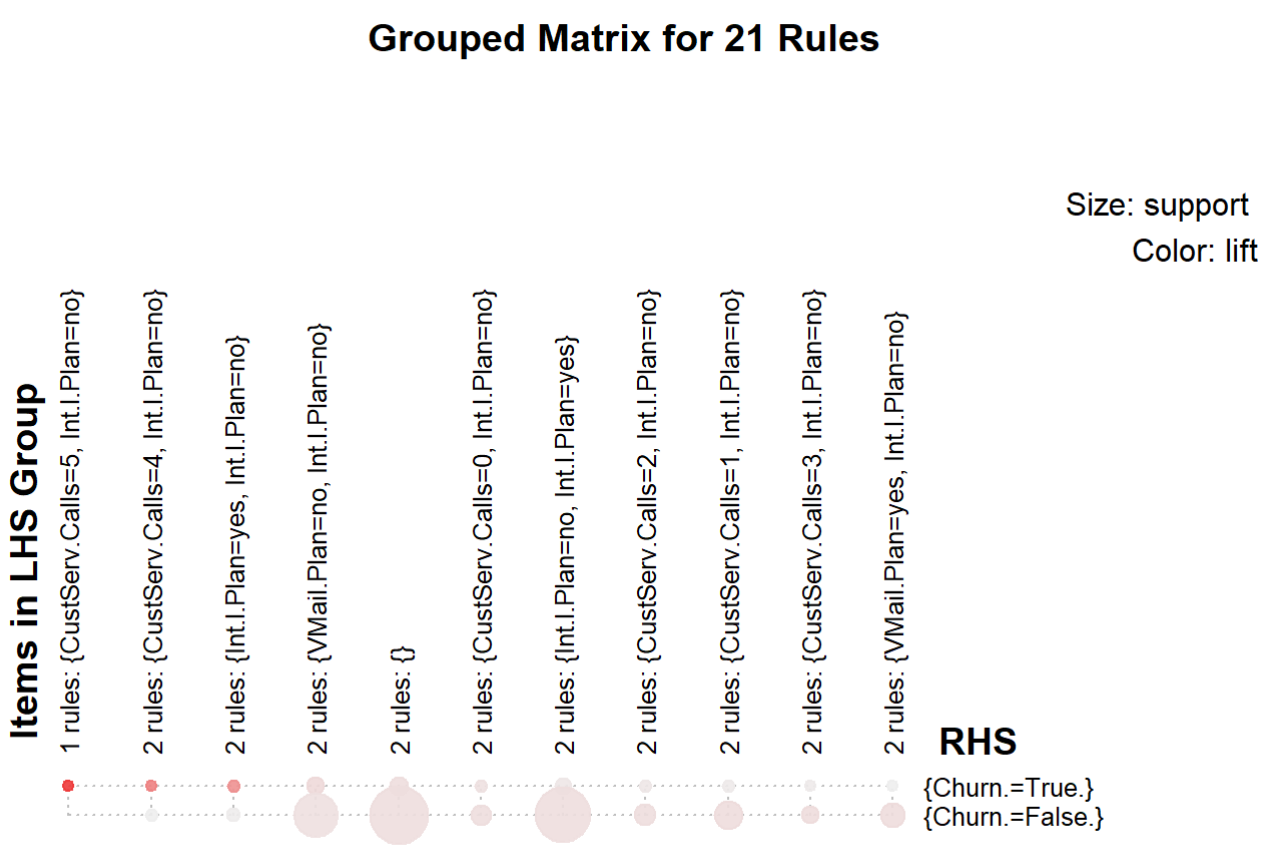
In model 3, when in consequent I only took churn=True, it's representing the same antecedents as above.

```
plot(ap2, method="paracoord", control=list(reorder=TRUE))
```



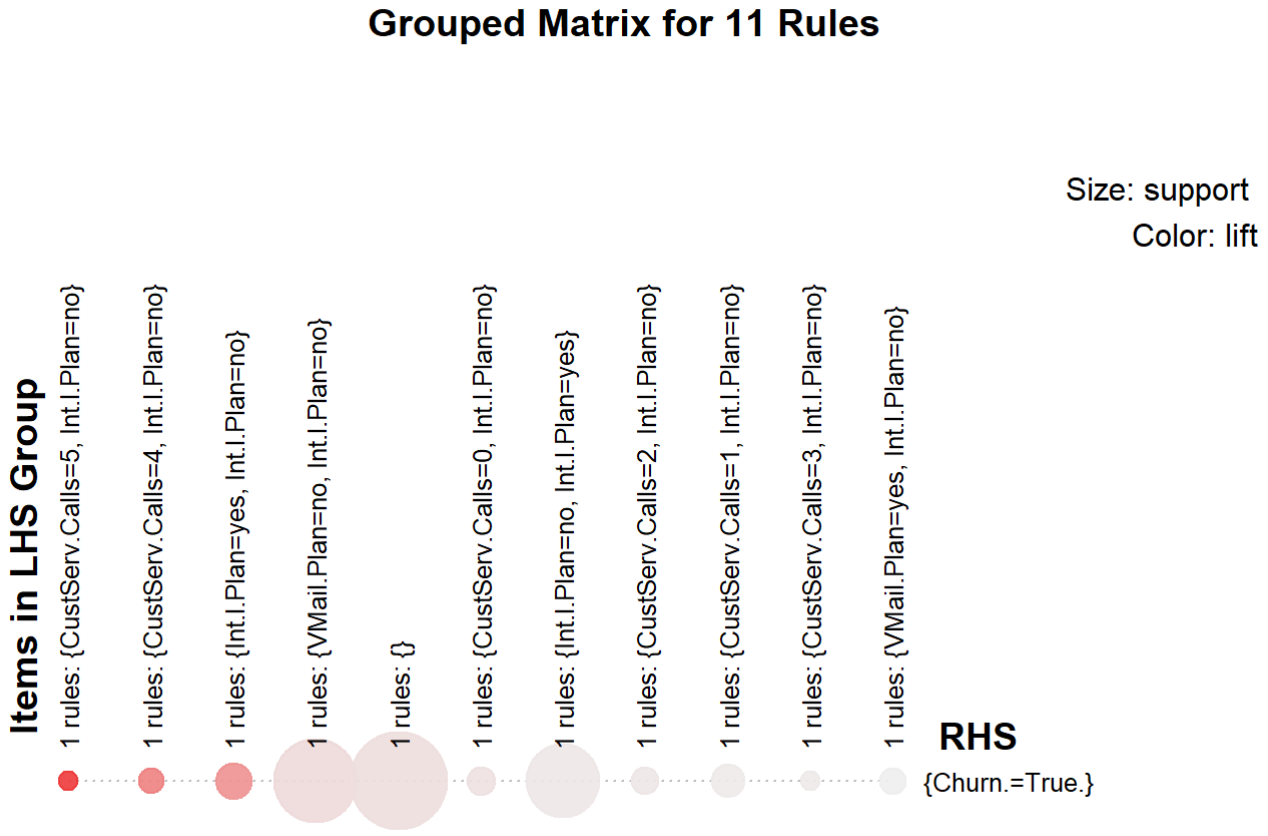
From the ap model, I plotted this bubble chart size is showing support where as colou is lift, all antecedents are in y axis with Churn true/false as consequent in x axis. For high support, VMail plan and Intl Plan is no for consequent churn=False.

```
plot(ap, method="grouped")
```



From below plot(model ap2), antecedent CustServ Calls=5 and Intl plan is No for consequent Churn=Truewhere as for high support VMail plan and Intl plan is no for consequent churn=true.

```
plot(ap2, method="grouped")
```



Conclusion

For the association analysis, I found highest lift is 4.18 for rule antecedent is CustomerServ Calls=5 then consequent is Churn =True with support 1% and confidence 60%. Different other models are also developed with different values in appearance option and different results were found(as above). From the models different plots like bubble plot and parallel coordinates were plotted to better understand the rules.