APPLIED PREDICTIVE ANALYTICS

Group 12

Airline company satisfaction

Wellington Daniel | Devereesh KM | Namratha Velivelli | Mohammed Rizwan

Replacing missing value by regression for arrival time

```
#loading data
rm(list=ls())
resolvrdata <- read_excel("resolvrdata.xlsx")
datarm<-na.omit(resolvrdata)

##93% correlation
l<-lm(datarm$`Arrival Delay in Minutes`~datarm$`Departure Delay in Minutes`)
summary(l)
```

```
##
## Call:
## lm(formula = datarm$`Arrival Delay in Minutes` ~ datarm$`Departure Delay in Minutes`)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.509  -1.975  -0.757  -0.467 236.438
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.7573788  0.0299329    25.3   <2e-16 ***
## datarm$`Departure Delay in Minutes` 0.9788417  0.0007361  1329.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.05 on 129445 degrees of freedom
## Multiple R-squared:  0.9318, Adjusted R-squared:  0.9318
## F-statistic: 1.768e+06 on 1 and 129445 DF,  p-value: < 2.2e-16
```

```
#predicting new values
datapred<-data.frame(Customer_ID=resolvrdata$`Customer ID`[is.na(resolvrdata$`Arrival Delay i
n Minutes`)])
datapred$`Departure Delay in Minutes`<-resolvrdata$`Departure Delay in Minutes`[is.na(resolvr
data$`Arrival Delay in Minutes`)]
datapred$`Arrival Delay in Minutes`<-cbind(rep(1,nrow(datapred)),datapred$`Departure Delay in
Minutes`)%*%l$coefficients

#Updating the old values with the new ones
resolvrdatacheck<-data.frame(resolvrdata)
rownames(resolvrdatacheck) = resolvrdatacheck$Customer.ID
rownames(datapred) = datapred$Customer_ID
resolvrdatacheck[rownames(datapred),c(25)] = datapred$`Arrival Delay in Minutes`

#data is the final data
sum(is.na(resolvrdatacheck))
```

```
## [1] 48
```

```
data<-na.omit(resolvrdatacheck)
rm(resolvrdata,datapred,datarm,l,resolvrdatacheck)
```

Outlier removal

```
#Replacing Arrival time
Arrivaltemp<-data$Arrival.Delay.in.Minutes
Arrivaltemp[Arrivaltemp>0]<-log(Arrivaltemp[Arrivaltemp>0])
m<-mean(Arrivaltemp[Arrivaltemp>0])+3*sqrt(var(Arrivaltemp[Arrivaltemp>0]))
sum(Arrivaltemp>m)
```

```
## [1] 20
```

```
datatemp<-data
datatemp$Arrival.Delay.in.Minutes[log(datatemp$Arrival.Delay.in.Minutes)>6.460957]<-NA
sum(is.na(datatemp))
```

```
## [1] 20
```

```
datatemp<-na.omit(datatemp)
rm(Arrivaltemp,m)


#Replacing Departure time
mean(log(data$Departure.Delay.in.Minutes)[log(data$Departure.Delay.in.Minutes)>0])+3*sqrt(var
(log(data$Departure.Delay.in.Minutes)[log(data$Departure.Delay.in.Minutes)>0]))
```

```
## [1] 6.490634
```

```
sum(log(data$Departure.Delay.in.Minutes)>6.490634)
```

```
## [1] 20
```

```
datatemp$Departure.Delay.in.Minutes[log(datatemp$Departure.Delay.in.Minutes)>6.490634]<-NA
sum(is.na(datatemp))
```

```
## [1] 0
```

```
datatemp<-na.omit(datatemp)

#Replacing Departure time
mean(log(data$Flight.Distance)[log(data$Flight.Distance)>0])-3*sqrt(var(log(data$Flight.Dista
nce)[log(data$Flight.Distance)>0]))
```

```
## [1] 3.95244
```

```
sum(log(data$Flight.Distance)<3.95244)
```

```
## [1] 11
```

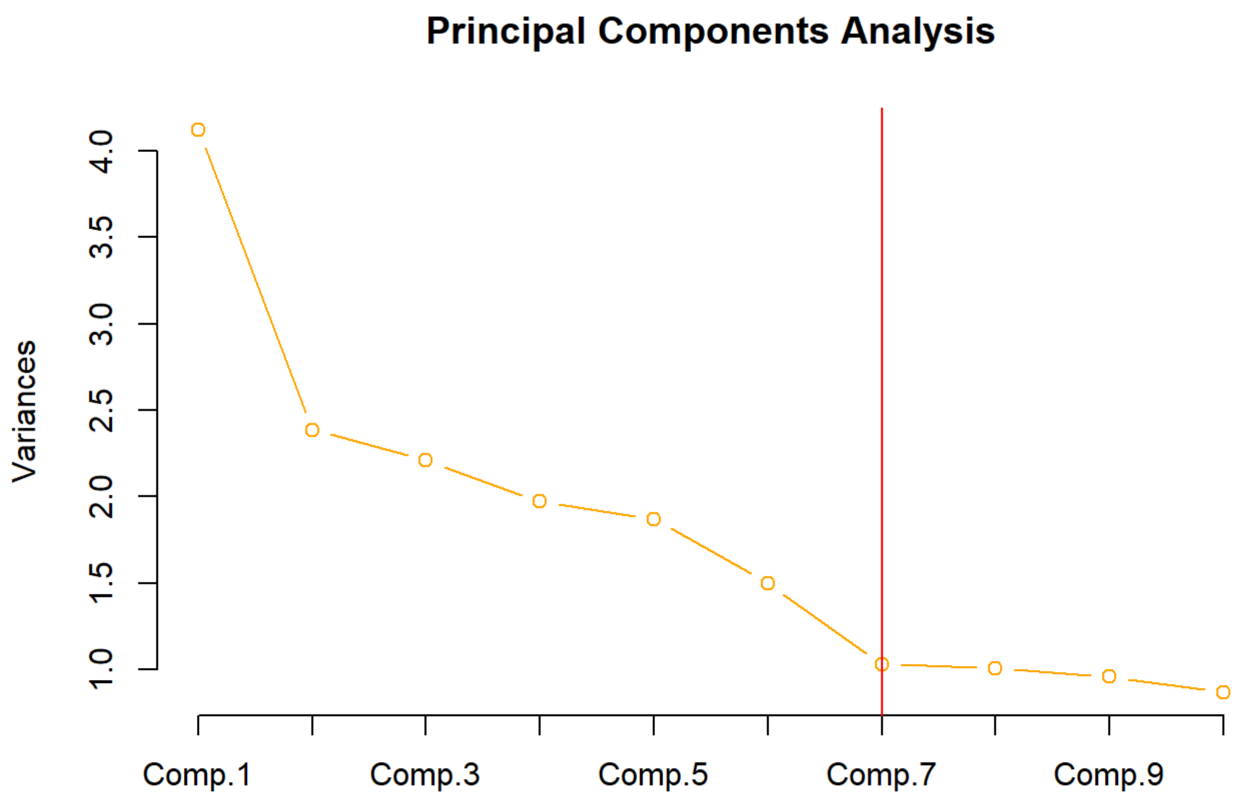
```
datatemp$Flight.Distance[log(datatemp$Flight.Distance)<3.95244]<-NA
sum(is.na(datatemp))
```

```
## [1] 11
```

```
datatemp<-na.omit(datatemp)

##storage
write.csv(datatemp,"DataFinal.csv")
Data<-datatemp[,c(-1,-6,-26)]
rm(data)
```

Principal Component Analysis

```
#PCA (to choose number of factors)
prin<-princomp(scale(Data))
plot(prin,type="l",col="orange",main="Principal Components Analysis")
abline(v=7,col="red")
```




```
summary(prin)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     2.0292779 1.5450845 1.48678279 1.40412311 1.36778088
## Proportion of Variance 0.1790435 0.1037958 0.09611044 0.08572073 0.08134082
## Cumulative Proportion  0.1790435 0.2828393 0.37894979 0.46467052 0.54601135
##                            Comp.6    Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     1.22482312 1.0158173 1.00322908 0.98026745 0.93125482
## Proportion of Variance 0.06522623 0.0448649 0.04375984 0.04177964 0.03770618
## Cumulative Proportion  0.61123757 0.6561025 0.69986232 0.74164196 0.77934814
##                           Comp.11    Comp.12    Comp.13    Comp.14    Comp.15
## Standard deviation     0.90065146 0.83037901 0.70878883 0.69327201 0.67378043
## Proportion of Variance 0.03526867 0.02997977 0.02184285 0.02089695 0.01973842
## Cumulative Proportion  0.81461681 0.84459657 0.86643942 0.88733637 0.90707478
##                           Comp.16    Comp.17    Comp.18    Comp.19    Comp.20
## Standard deviation     0.64945530 0.60511230 0.56550522 0.54161174 0.52942270
## Proportion of Variance 0.01833893 0.01592016 0.01390429 0.01275415 0.01218655
## Cumulative Proportion  0.92541372 0.94133388 0.95523816 0.96799232 0.98017886
##                           Comp.21     Comp.22     Comp.23
## Standard deviation     0.49283059 0.418802248 0.193920834
## Proportion of Variance 0.01056017 0.007625942 0.001635025
## Cumulative Proportion  0.99073903 0.998364975 1.000000000
```

```
#Factor Analysis (6 Factors are chosen)
```

Factor Analysis (6 Factors are chosen)

```
fact<-factanal(scale(Data),6,rotation="varimax")
ftemp<-fact$loadings
ftemp<-matrix(fact$loadings,ncol=6)
ftemp<-data.frame(ftemp)
fact
```

```
##
## Call:
## factanal(x = scale(Data), factors = 6, rotation = "varimax")
##
## Uniquenesses:
##                      GenderMale1                   LoyalCustomer1
##                            0.998                            0.458
##                              Age       Travel_Personal1Business0
##                            0.865                            0.261
##                   Class_Ecoplus1                  Class_Business2
##                            0.942                            0.307
##                  Flight.Distance           Inflight.wifi.service
##                            0.671                            0.360
## Departure.Arrival.time.convenient          Ease.of.Online.booking
##                            0.620                            0.142
##                    Gate.location                   Food.and.drink
##                            0.751                            0.394
##                  Online.boarding                    Seat.comfort
##                            0.564                            0.362
##           Inflight.entertainment                On.board.service
##                            0.190                            0.496
##                 Leg.room.service                Baggage.handling
##                            0.736                            0.410
##                  Checkin.service                Inflight.service
##                            0.893                            0.338
##                      Cleanliness       Departure.Delay.in.Minutes
##                            0.263                            0.069
##         Arrival.Delay.in.Minutes
##                            0.005
##
## Loadings:
##                                   Factor1 Factor2 Factor3 Factor4 Factor5
## GenderMale1
## LoyalCustomer1
## Age                                                                0.165
## Travel_Personal1Business0         -0.100                          -0.753
## Class_Ecoplus1                                                    -0.228
## Class_Business2                            0.169                   0.799
## Flight.Distance                                                    0.493
## Inflight.wifi.service              0.151   0.109   0.775
## Departure.Arrival.time.convenient          0.108   0.507          -0.217
## Ease.of.Online.booking                             0.918           0.126
## Gate.location                                      0.497
## Food.and.drink                     0.778
## Online.boarding                    0.327           0.398           0.350
## Seat.comfort                       0.757                           0.172
## Inflight.entertainment             0.783   0.431
## On.board.service                   0.113   0.689                   0.122
## Leg.room.service                           0.465                   0.182
## Baggage.handling                           0.759
## Checkin.service                    0.110   0.268
## Inflight.service                           0.804
## Cleanliness                        0.853
## Departure.Delay.in.Minutes                                 0.965
## Arrival.Delay.in.Minutes                                   0.997
##                                   Factor6
## GenderMale1
```

```
## LoyalCustomer1                    0.728
## Age                               0.320
## Travel_Personal1Business0         0.396
## Class_Ecoplus1
## Class_Business2                   0.133
## Flight.Distance                   0.277
## Inflight.wifi.service
## Departure.Arrival.time.convenient  0.249
## Ease.of.Online.booking
## Gate.location
## Food.and.drink
## Online.boarding                   0.213
## Seat.comfort                      0.182
## Inflight.entertainment
## On.board.service
## Leg.room.service
## Baggage.handling
## Checkin.service                   0.126
## Inflight.service
## Cleanliness
## Departure.Delay.in.Minutes
## Arrival.Delay.in.Minutes
##
##                 Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
## SS loadings       2.725   2.249   2.121   1.929   1.818   1.065
## Proportion Var   0.118   0.098   0.092   0.084   0.079   0.046
## Cumulative Var   0.118   0.216   0.308   0.392   0.471   0.518
##
## Test of the hypothesis that 6 factors are sufficient.
## The chi square statistic is 80068.75 on 130 degrees of freedom.
## The p-value is 0
```

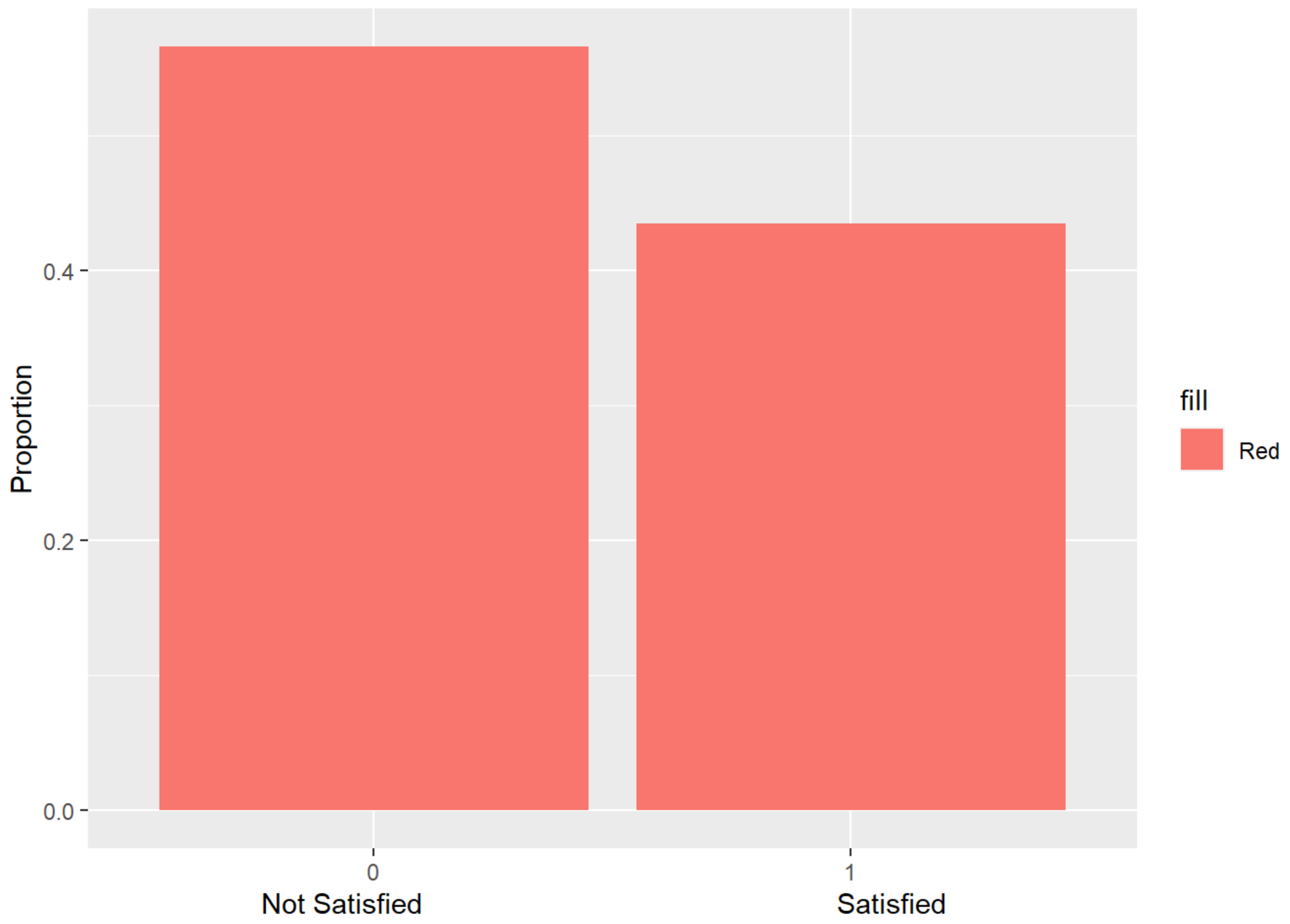
```
rm(fact,ftemp,prin,Data)
```

Exploratory Data Analysis

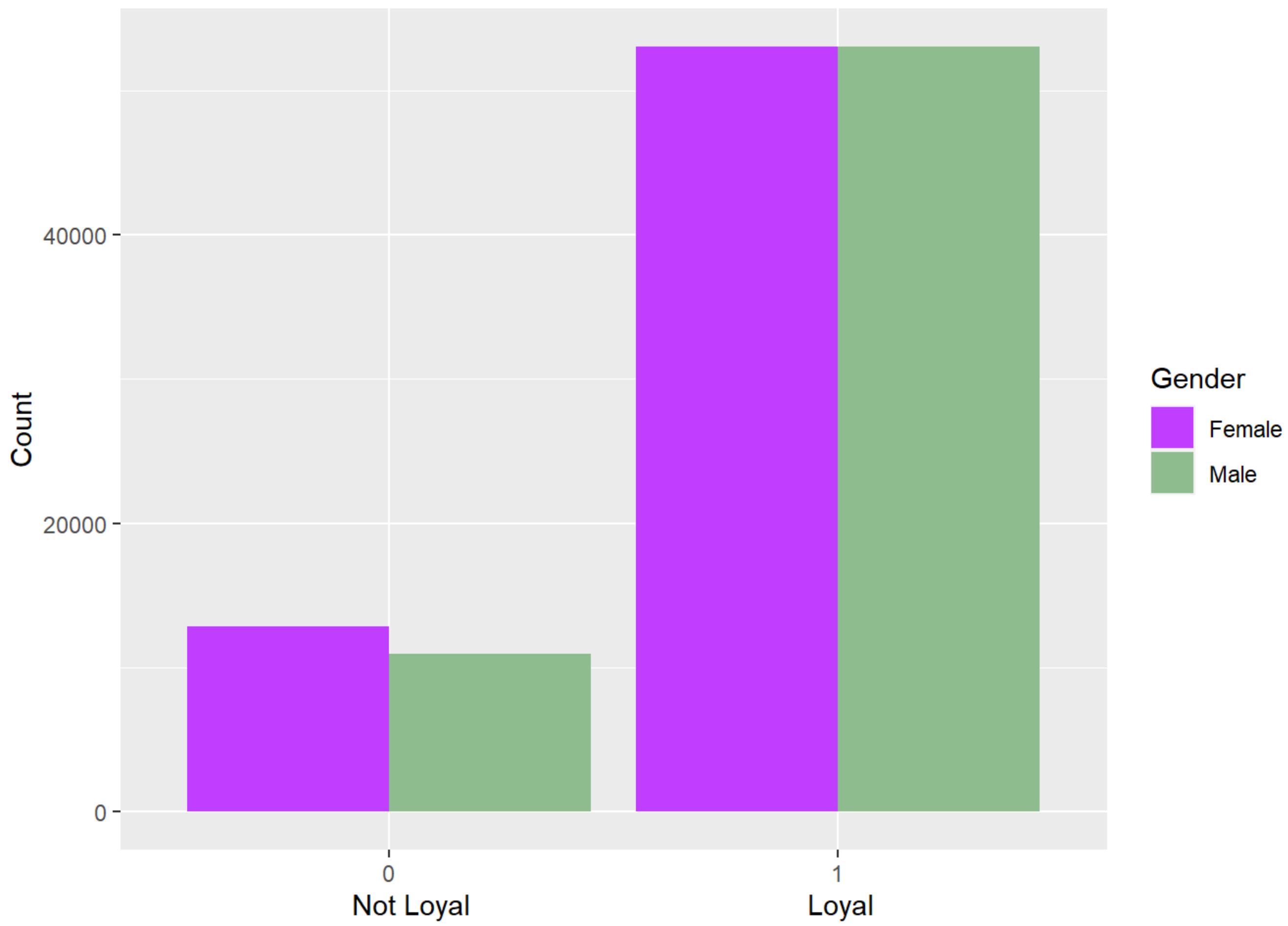
```
Finaldata<-datatemp
str(Finaldata)
```

```
## 'data.frame':    129809 obs. of  26 variables:
##  $ Customer.ID                      : num  70172 5047 110028 24026 119299 ...
##  $ GenderMale1                      : num  1 1 0 0 1 0 0 0 1 0 ...
##  $ LoyalCustomer1                   : num  1 0 1 1 1 1 1 1 0 0 ...
##  $ Age                              : num  13 25 26 25 61 26 52 41 20 24 ...
##  $ Travel_Personal1Business0        : num  1 0 0 0 0 1 0 0 0 0 ...
##  $ Class_Eco0                       : num  0 0 0 0 0 1 0 0 1 1 ...
##  $ Class_Ecoplus1                   : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ Class_Business2                  : num  0 1 1 1 1 0 1 1 0 0 ...
##  $ Flight.Distance                  : num  460 235 1142 562 214 ...
##  $ Inflight.wifi.service            : num  3 3 2 2 3 3 4 1 3 4 ...
##  $ Departure.Arrival.time.convenient: num  4 2 2 5 3 4 3 2 3 5 ...
##  $ Ease.of.Online.booking           : num  3 3 2 5 3 2 4 2 3 5 ...
##  $ Gate.location                    : num  1 3 2 5 3 1 4 2 4 4 ...
##  $ Food.and.drink                   : num  5 1 5 2 4 1 5 4 2 2 ...
##  $ Online.boarding                  : num  3 3 5 2 5 2 5 3 3 5 ...
##  $ Seat.comfort                     : num  5 1 5 2 5 1 5 3 3 2 ...
##  $ Inflight.entertainment           : num  5 1 5 2 3 1 5 1 2 2 ...
##  $ On.board.service                 : num  4 1 4 2 3 3 5 1 2 3 ...
##  $ Leg.room.service                 : num  3 5 3 5 4 4 5 2 3 3 ...
##  $ Baggage.handling                 : num  4 3 4 3 4 4 5 1 4 5 ...
##  $ Checkin.service                  : num  4 1 4 1 3 4 4 4 4 3 ...
##  $ Inflight.service                 : num  5 4 4 4 3 4 5 1 3 5 ...
##  $ Cleanliness                      : num  5 1 5 2 3 1 4 2 2 2 ...
##  $ Departure.Delay.in.Minutes       : num  25 1 0 11 0 0 4 0 0 0 ...
##  $ Arrival.Delay.in.Minutes         : num  18 6 0 9 0 0 0 0 0 0 ...
##  $ satisfaction                     : num  0 0 1 0 1 0 1 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:11] 8270 15884 24767 69038 73311 87897 88499
90478 106442 107760 ...
##   ..- attr(*, "names")= chr [1:11] "29863" "29824" "30125" "30130" ...
```

```
#Proportion of satisfied customers
ggplot(Finaldata, aes(x = as.factor(satisfaction))) +
  geom_bar(mapping = aes(x = as.factor(satisfaction), y = stat(prop), group = 1,fill="Red"))
 +
  ylab("Proportion")+
  xlab(("         Not Satisfied                                                    Satisfied
"))
```

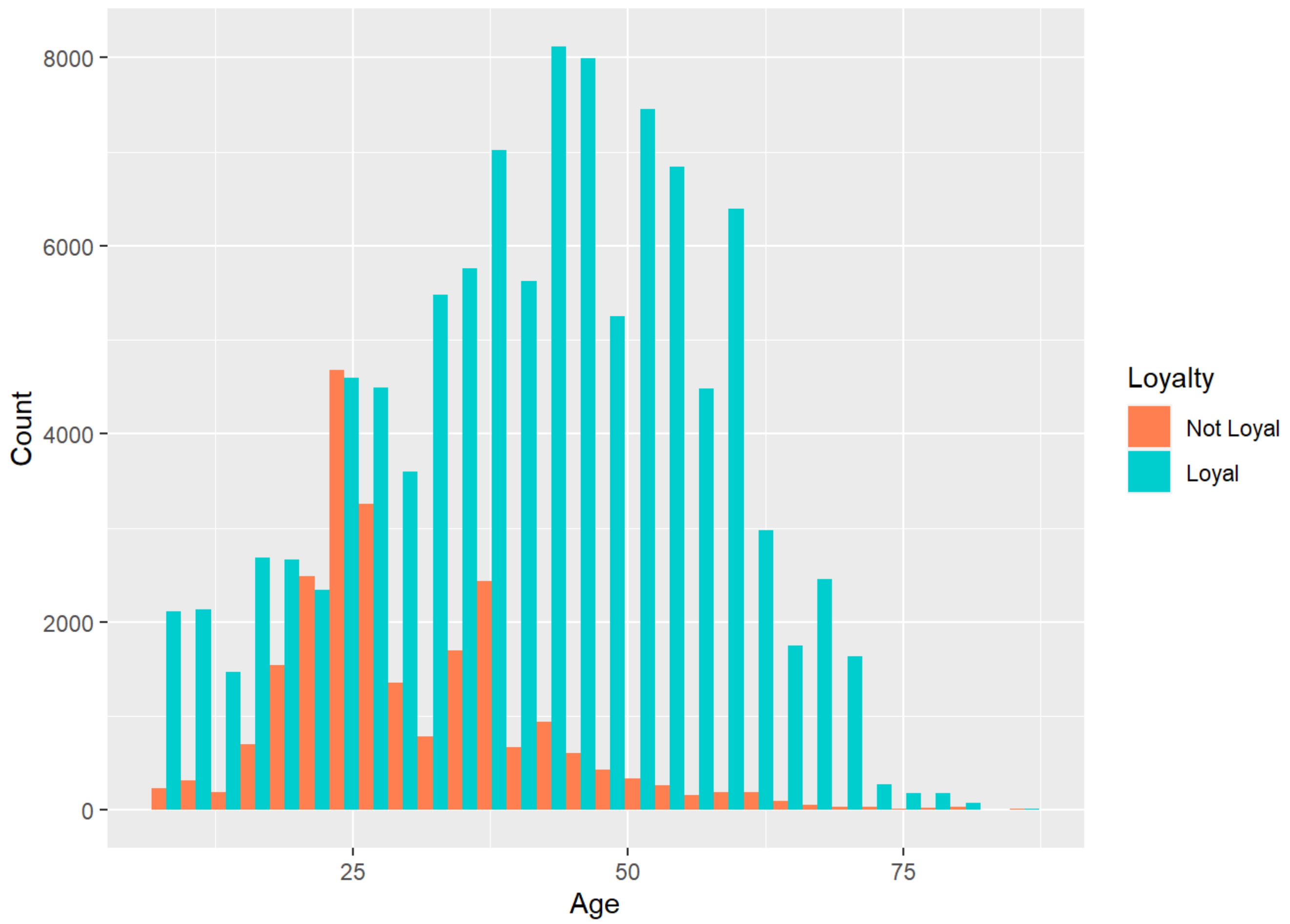



```
#Loyalty and Gender (Not significant)
ggplot(Finaldata, aes(x = as.factor(Finaldata$LoyalCustomer1), fill = as.factor(as.factor(Fin
aldata$GenderMale1)))) +
  geom_bar(position = "dodge")+
  ylab("Count")+
  xlab("Not Loyal                                            Loyal")+
  scale_fill_manual(name="Gender", values=c("darkorchid1","darkseagreen"), labels=c("Female",
"Male"))
```



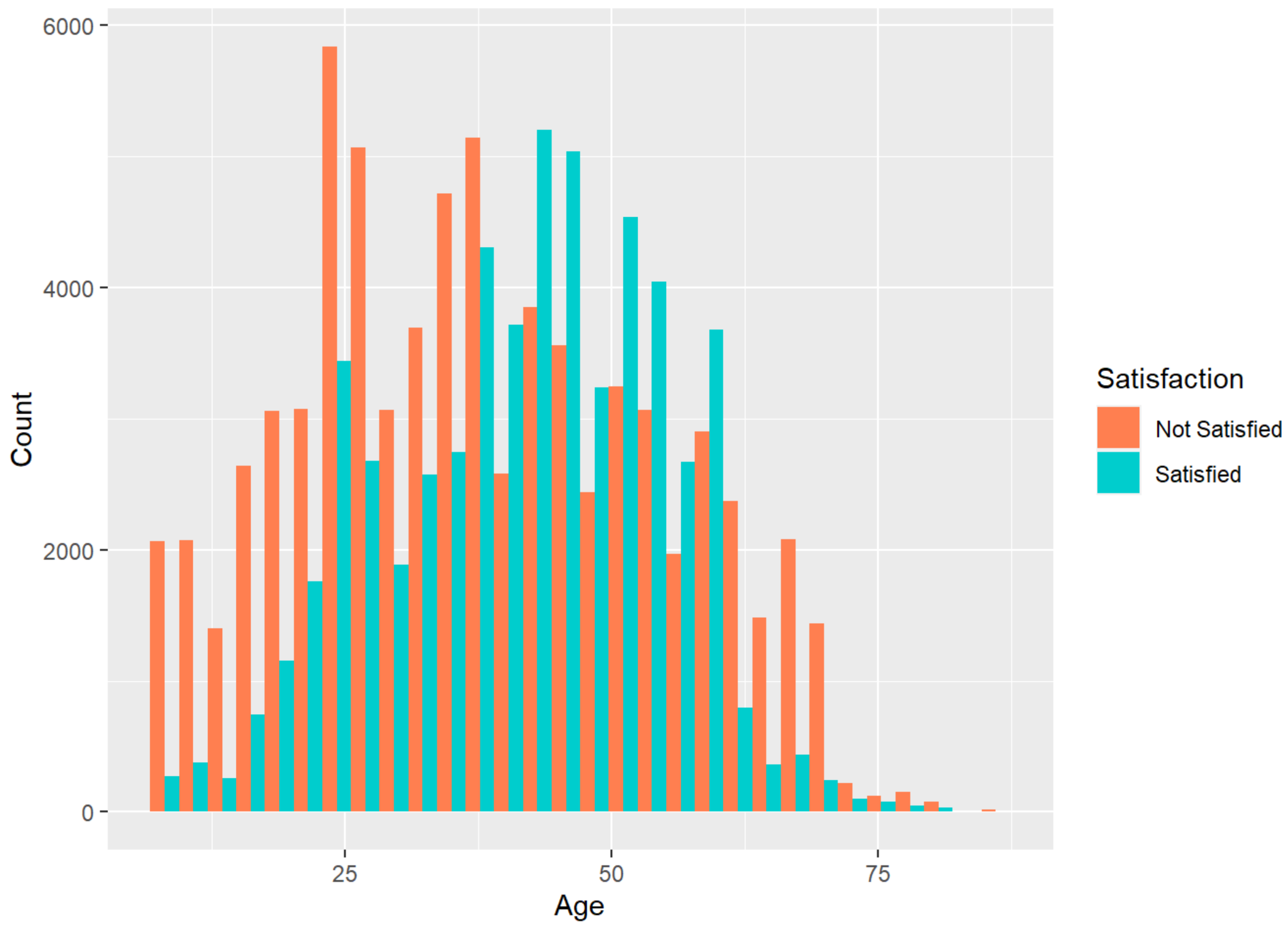

```
#Age effect on loyalty
ggplot(data = Finaldata, aes(x =Age,fill = as.factor(Finaldata$LoyalCustomer1))) +
  geom_histogram(position = "dodge")+
  ylab("Count")+
  scale_fill_manual(name="Loyalty", values=c("coral","cyan3"), labels=c("Not Loyal", "Loyal"
))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

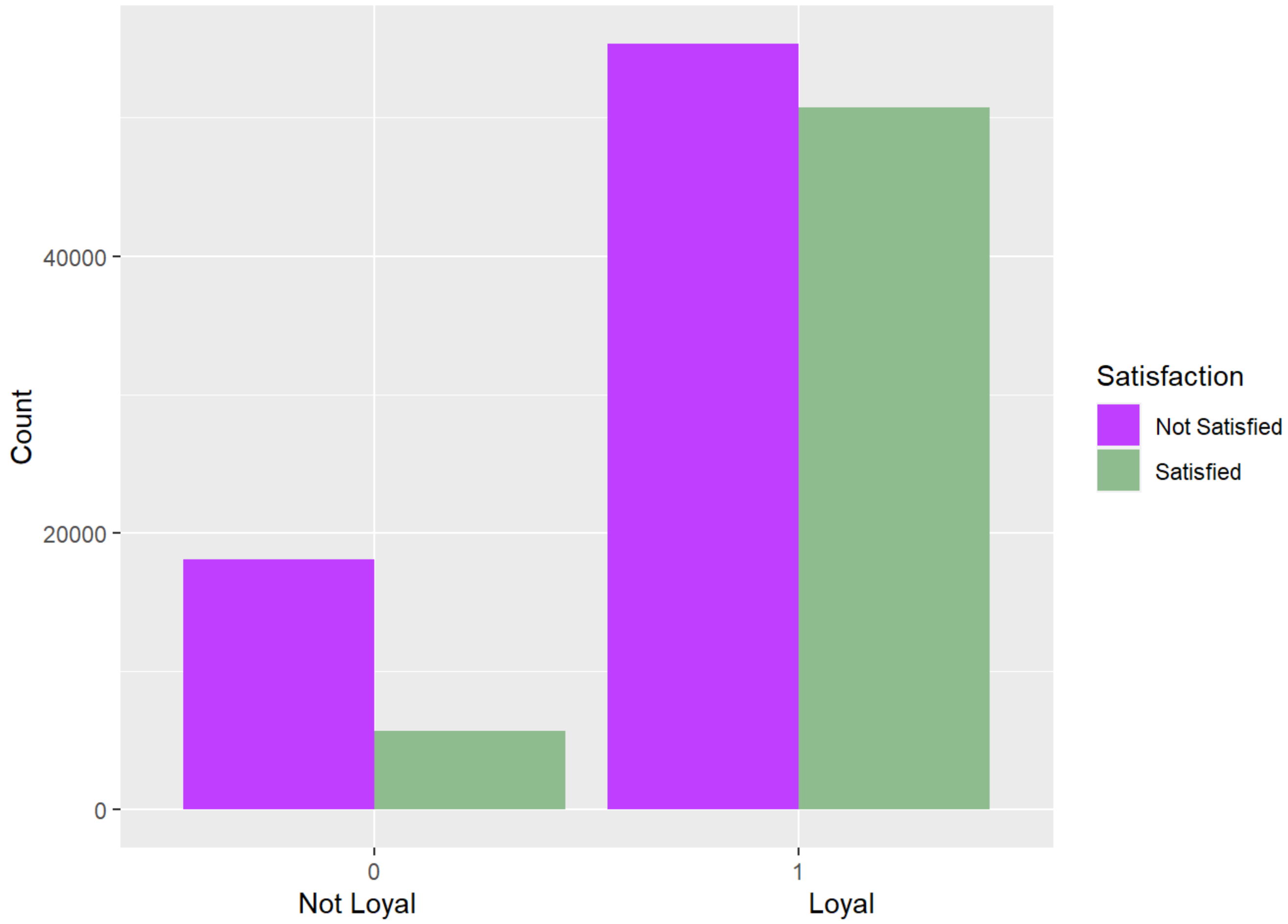



```
#Age effect on Satisfaction
ggplot(data = Finaldata, aes(x =Age,fill = as.factor(Finaldata$satisfaction))) +
  geom_histogram(position = "dodge")+
  ylab("Count")+
  scale_fill_manual(name="Satisfaction", values=c("coral","cyan3"), labels=c("Not Satisfied",
"Satisfied"))
```

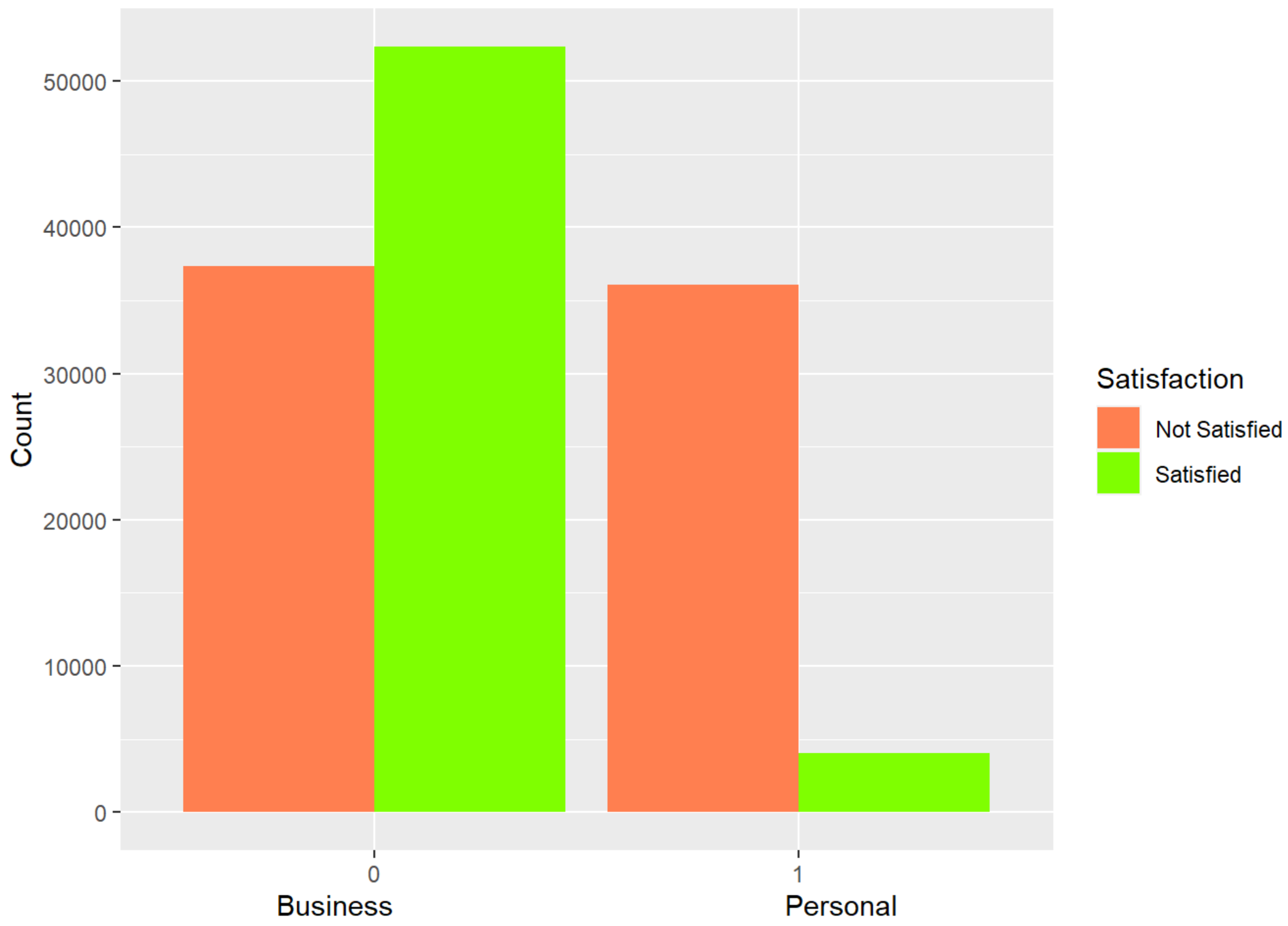
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Loyalty and Satisfaction
ggplot(Finaldata, aes(x = as.factor(Finaldata$LoyalCustomer1), fill = as.factor(as.factor(Fin
aldata$satisfaction)))) +
  geom_bar(position = "dodge")+
  ylab("Count")+
  xlab("Not Loyal                                                  Loyal")+
  scale_fill_manual(name="Satisfaction", values=c("darkorchid1","darkseagreen"), labels=c("No
t Satisfied", "Satisfied"))
```

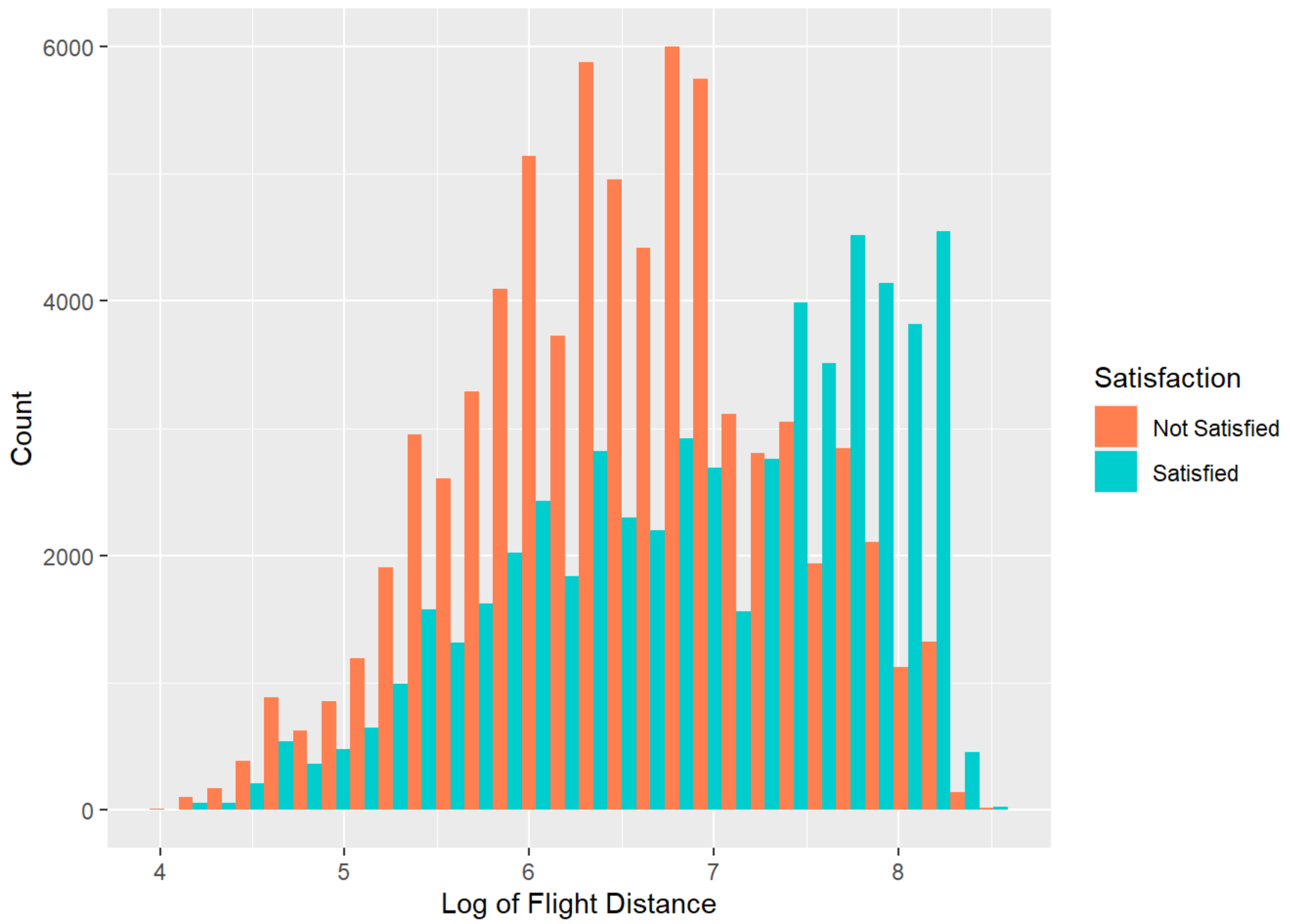



```
#Type of travelers and satisfaction
ggplot(Finaldata, aes(x = as.factor(Finaldata$Travel_Personal1Business0), fill = as.factor(a
s.factor(Finaldata$satisfaction)))) +
  geom_bar(position = "dodge")+
  ylab("Count")+
  xlab("Business                                                  Personal")+
  scale_fill_manual(name="Satisfaction", values=c("coral","chartreuse"), labels=c("Not Satisf
ied", "Satisfied"))
```




```
#Effect of Flight Distance on satisfaction
ggplot(data = Finaldata, aes(x =log(Finaldata$Flight.Distance),fill = as.factor(Finaldata$sat
isfaction))) +
  geom_histogram(position = "dodge")+
  ylab("Count")+
  xlab("Log of Flight Distance")+
  scale_fill_manual(name="Satisfaction", values=c("coral","cyan3"), labels=c("Not Satisfied",
"Satisfied"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



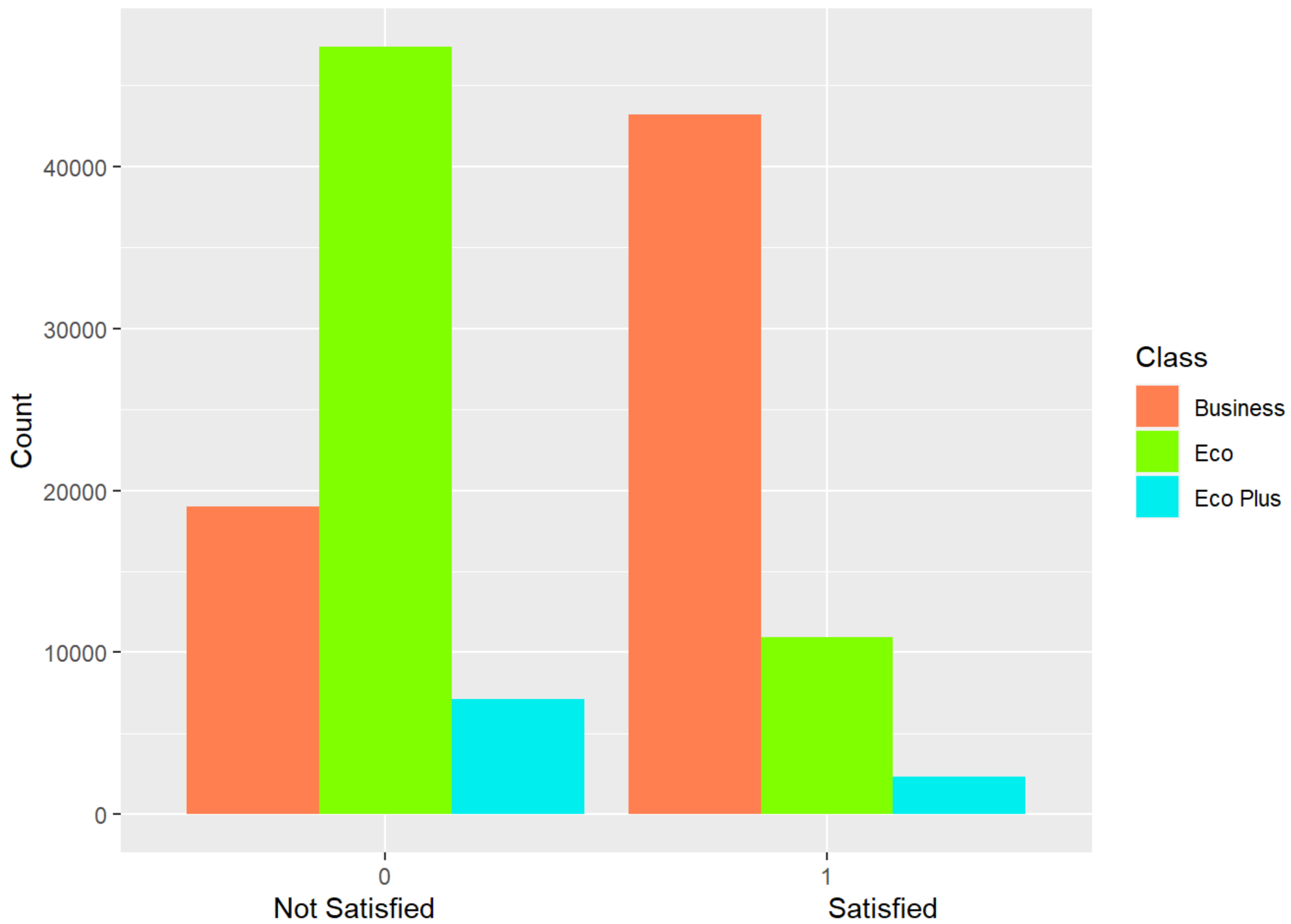

```
varnew <- read_excel("varnew.xlsx")
```

```
## New names:
## * `Type of Travel` -> `Type of Travel...3`
## * `Type of Travel` -> `Type of Travel...7`
```

```
#Effect of class on satisfaction
ggplot(varnew, aes(x = as.factor(varnew$satisfaction), fill = varnew$Class)) +
  geom_bar(position = "dodge")+
  ylab("Count")+
  xlab("Not Satisfied                                                  Satisfied")+
  scale_fill_manual(name="Class", values=c("coral","chartreuse","cyan2"))
```

```
## Warning: Use of `varnew$satisfaction` is discouraged. Use `satisfaction`
## instead.
```

```
## Warning: Use of `varnew$Class` is discouraged. Use `Class` instead.
```



HCPC (Francois Husson, Julie Josse, Jerome Pages 2010)

```
varnew[,c(6:9)]<-scale(varnew[,c(6:9)])
head(varnew)
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree
```

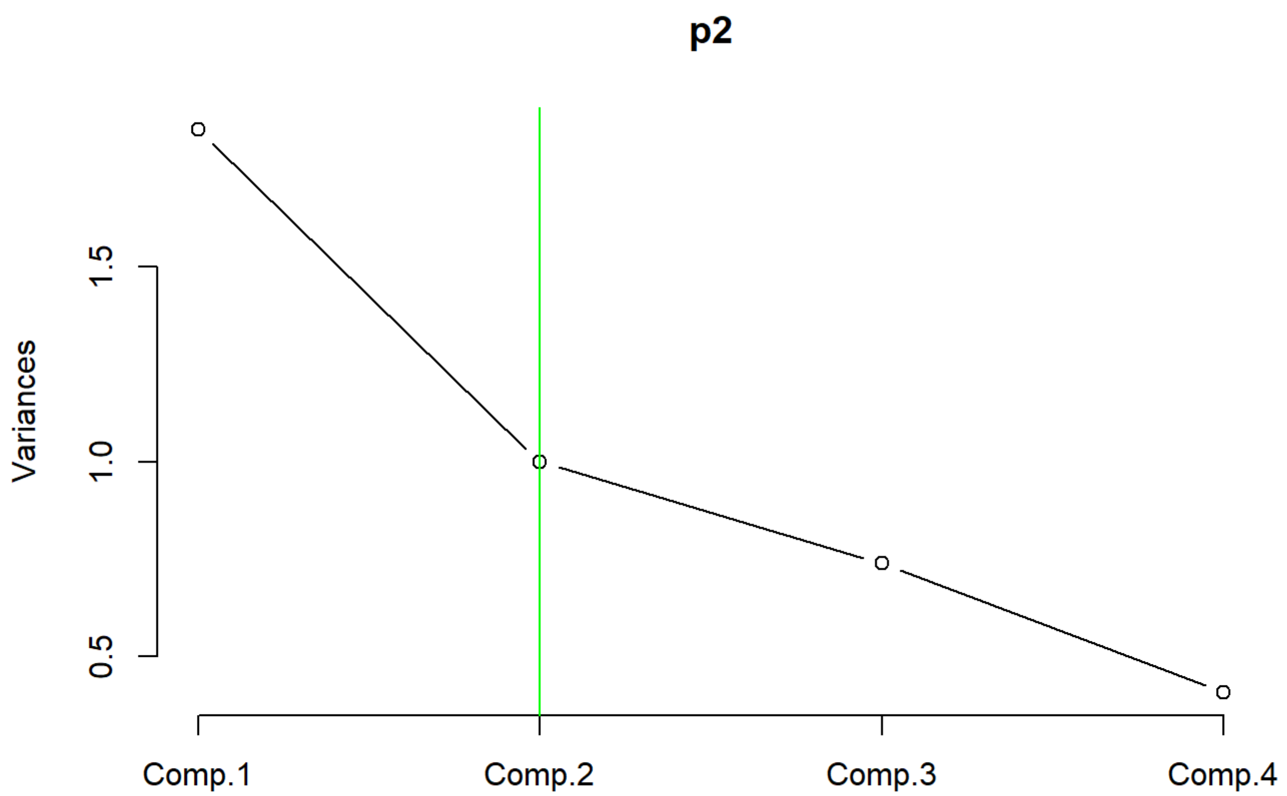
```
## # A tibble: 6 x 9
##   `Customer ID` Class `Type of Travel~ `Customer Type` satisfaction Class_No
##           <dbl> <chr> <chr>            <chr>                  <dbl>    <dbl>
## 1         70172 Eco ~ Personal Travel  Loyal Customer             0  -0.0308
## 2          5047 Busi~ Business travel  disloyal Custo~            0   1.01
## 3        110028 Busi~ Business travel  Loyal Customer             1   1.01
## 4         24026 Busi~ Business travel  Loyal Customer             0   1.01
## 5        119299 Busi~ Business travel  Loyal Customer             1   1.01
## 6        111157 Eco   Personal Travel  Loyal Customer             0  -1.07
## # ... with 3 more variables: `Type of Travel...7` <dbl>, `Flight
## #   Distance` <dbl>, Age <dbl>
```

PCA (choose eigenvalues>1)

```
#PCA ( choose eigenvalues>1)
p2<-princomp(na.omit(varnew[,c(6:9)]))
p2$loadings
```

```
##
## Loadings:
##                    Comp.1 Comp.2 Comp.3 Comp.4
## Class_No            0.639                0.763
## Type of Travel...7  0.571         0.606 -0.554
## Flight Distance     0.516        -0.789 -0.332
## Age                       -1.000
##
##                Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

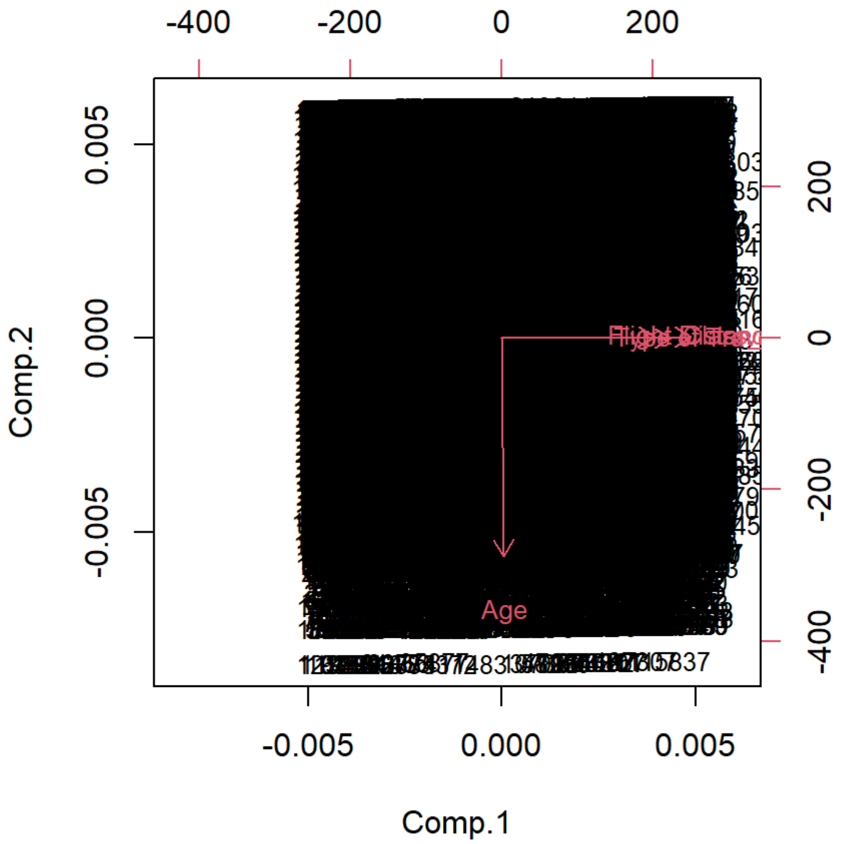
```
plot(p2,type="l")
abline(v=2,col="green")
```




```
summary(p2)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4
## Standard deviation     1.3608900 0.9999806 0.8601185 0.6389001
## Proportion of Variance 0.4630077 0.2499916 0.1849519 0.1020488
## Cumulative Proportion  0.4630077 0.7129993 0.8979512 1.0000000
```

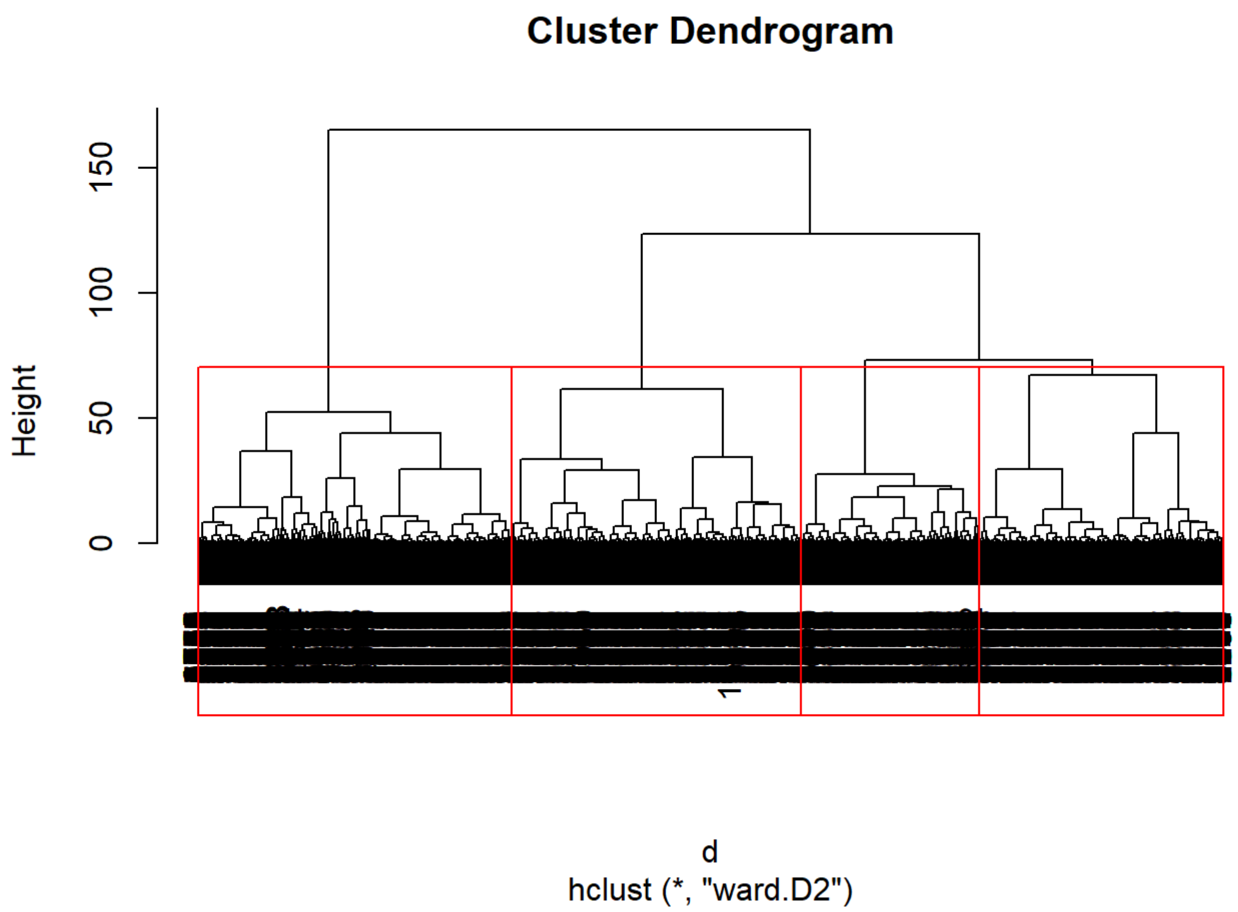
```
biplot(p2,cex=0.8)
```




```
p3<-data.frame(p2$scores)
rname<-na.omit(varnew)
rname<-rname$`Customer ID`
```

Hierarchical Clustering

```
d<-dist(p3[1:10000,])
fitH<-hclust(d,"ward.D2")
plot(fitH)
rect.hclust(fitH,k=4,border = "red")
```

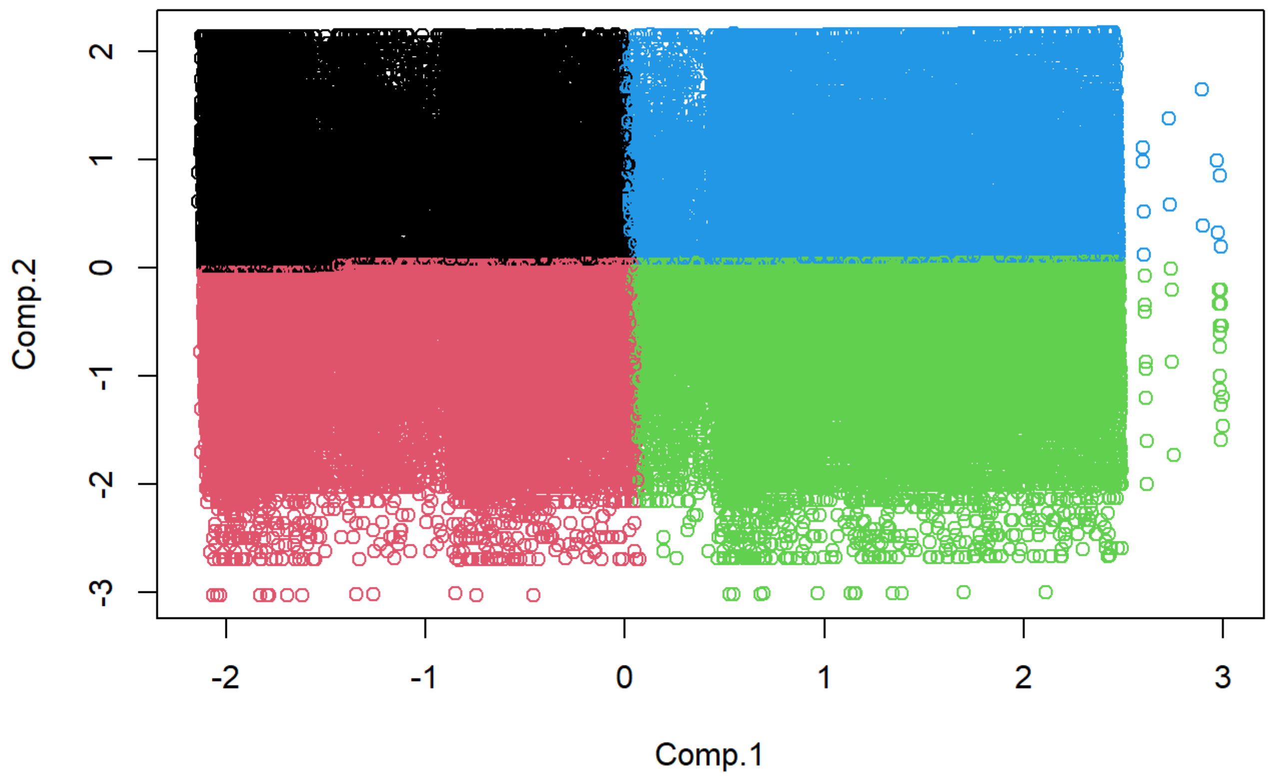


K-means

```
p4<-data.frame(matrix(p2$loadings,nrow=4))
k<-kmeans(p3[,c(1:2)],4)
summary(k)
```

```
##              Length Class  Mode
## cluster      129829 -none- numeric
## centers           8 -none- numeric
## totss             1 -none- numeric
## withinss          4 -none- numeric
## tot.withinss      1 -none- numeric
## betweenss         1 -none- numeric
## size              4 -none- numeric
## iter              1 -none- numeric
## ifault            1 -none- numeric
```

```
p5<-data.frame(matrix(k$centers,nrow=4))
plo<-data.frame(cbind(p3[,c(1:2)],k$cluster))
plot(plo[,1:2],col=plo$k.cluster)
```

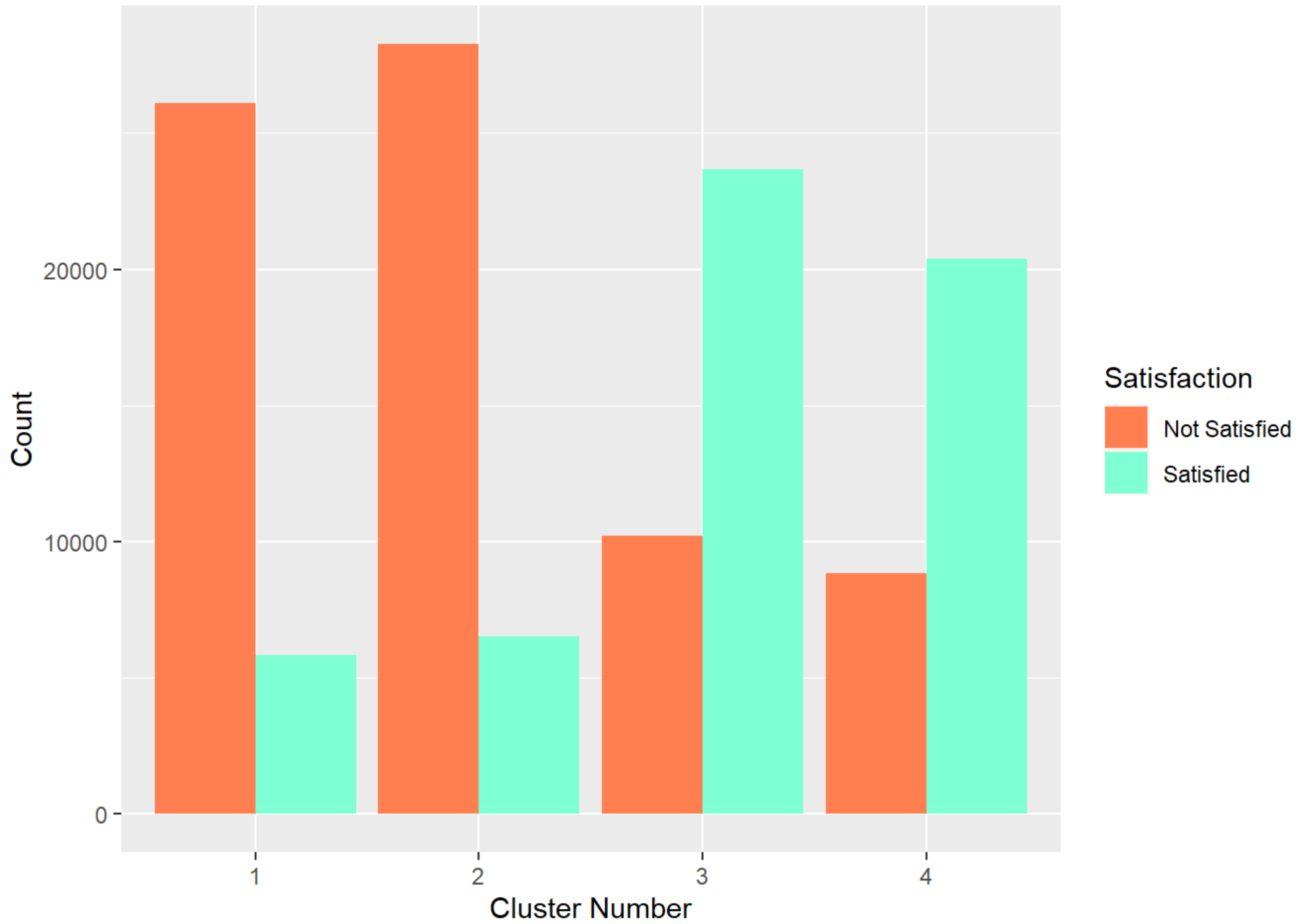



```
finalinsightdata<-data.frame(Customer.ID=rname,k$cluster)

grap<-data.frame(cbind(na.omit(varnew)$satisfaction,k$cluster))
head(grap)
```

```
##   X1 X2
## 1  0  1
## 2  0  4
## 3  1  4
## 4  0  4
## 5  1  3
## 6  0  1
```

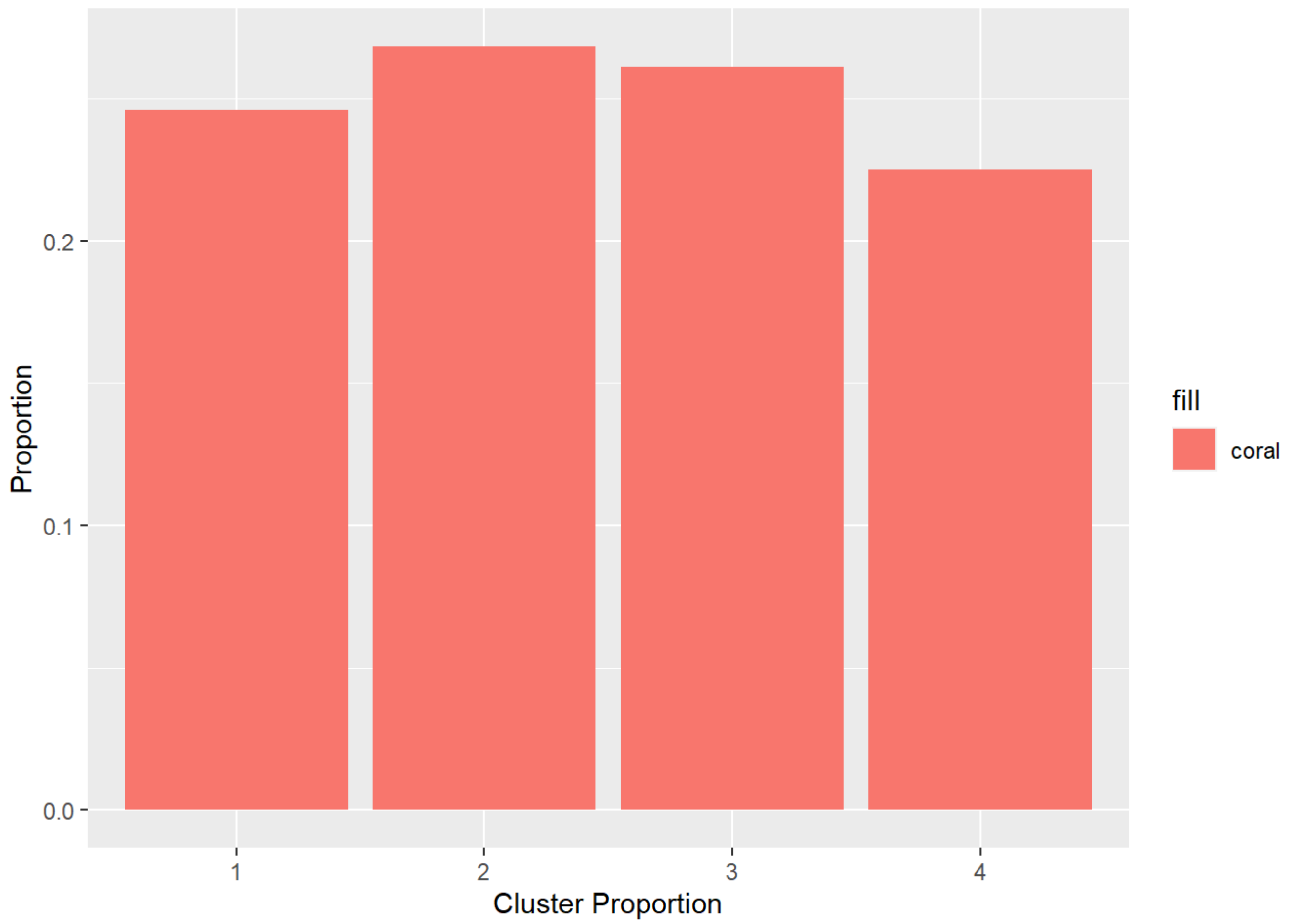
Satisfaction by cluster

```
ggplot(grap, aes(x = as.factor(X2), fill = as.factor(X1)))+
  geom_bar(position = "dodge")+
  ylab("Count")+
  xlab("Cluster Number")+
  scale_fill_manual(name="Satisfaction", values=c("coral","aquamarine"), labels=c("Not Satisf
ied", "Satisfied"))
```



Total fliers by cluster

```
ggplot(grap, aes(x = as.factor(X2))) +
  geom_bar(mapping = aes(x = as.factor(X2), y = stat(prop), group = 1,fill="coral")) +
  ylab("Proportion")+
  xlab(("Cluster Proportion"))
```



Model Fitting

```
rm(grap,k,p2,p3,p4,p5,plo,varnew,datatemp,d,fitH)
```

Boruta Variable selection

```
boruta_output <- Boruta(as.factor(satisfaction) ~ ., data=Finaldata[1:2000,], doTrace=2)
```

```
##  1. run of importance source...
```

```
##  2. run of importance source...
```

```
##  3. run of importance source...
```

```
##  4. run of importance source...
```

```
##  5. run of importance source...
```

```
##  6. run of importance source...
```

```
##  7. run of importance source...
```

```
##  8. run of importance source...
```

```
##  9. run of importance source...
```

```
##  10. run of importance source...
```

```
##  11. run of importance source...
```

```
##  12. run of importance source...
```

```
## After 12 iterations, +10 secs:
```

```
##  confirmed 22 attributes: Age, Baggage.handling, Checkin.service, Class_Business2, Class_E
co0 and 17 more;
```

```
##  still have 3 attributes left.
```

```
##  13. run of importance source...
```

```
##  14. run of importance source...
```

```
##  15. run of importance source...
```

```
##  16. run of importance source...
```

```
## After 16 iterations, +14 secs:
```

```
##  confirmed 2 attributes: Arrival.Delay.in.Minutes, Departure.Delay.in.Minutes;
```

```
##  still have 1 attribute left.
```

```
##  17. run of importance source...
```

```
##  18. run of importance source...
```

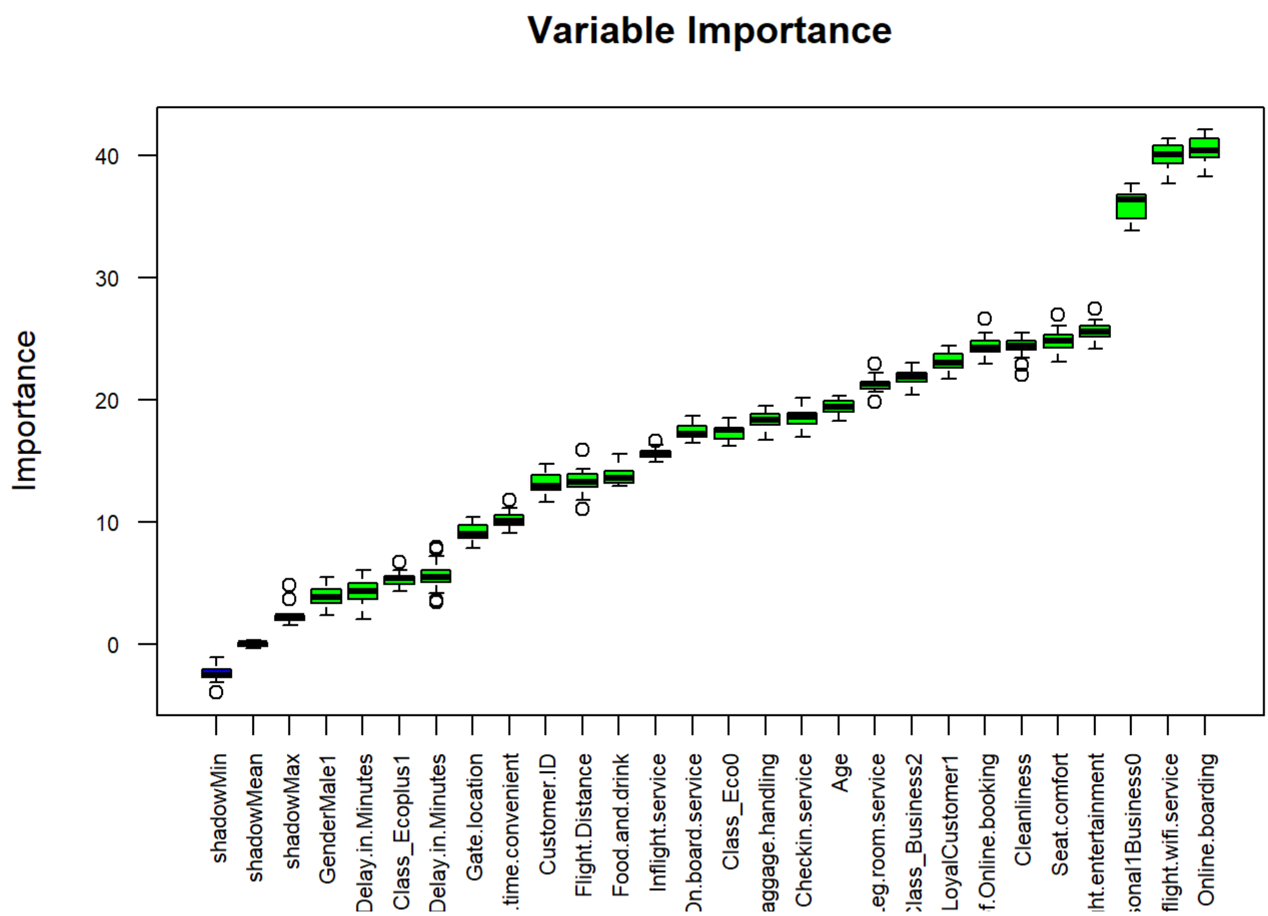
```
##  19. run of importance source...
```

```
## After 19 iterations, +18 secs:
```

```
##  confirmed 1 attribute: GenderMale1;
```

```
##  no more attributes left.
```

```
boruta_signif <- names(boruta_output$finalDecision[boruta_output$finalDecision %in% c("Confir
med", "Tentative")])
plot(boruta_output, cex.axis=0.7, las=2, xlab="", main="Variable Importance")
```



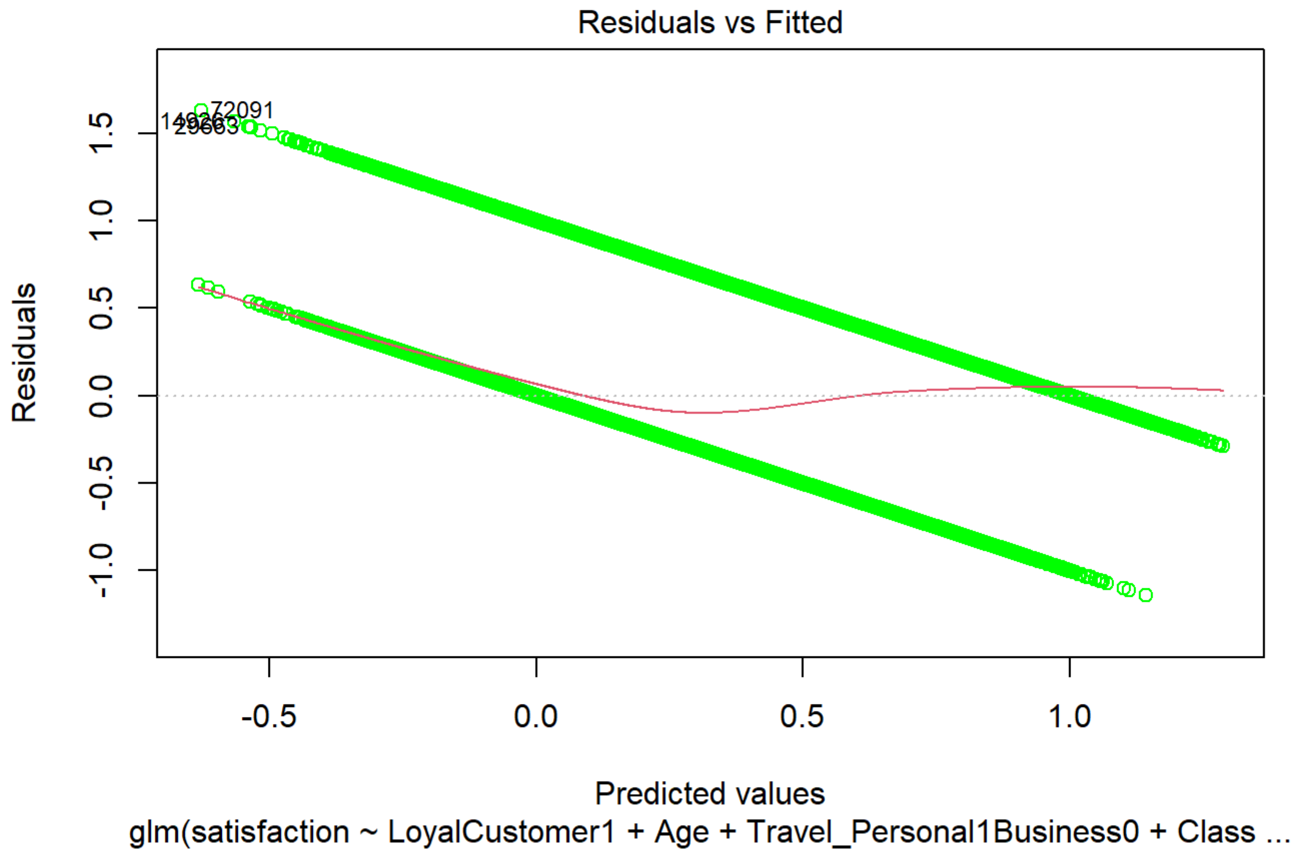

```
rm(boruta_output,boruta_signif)
```

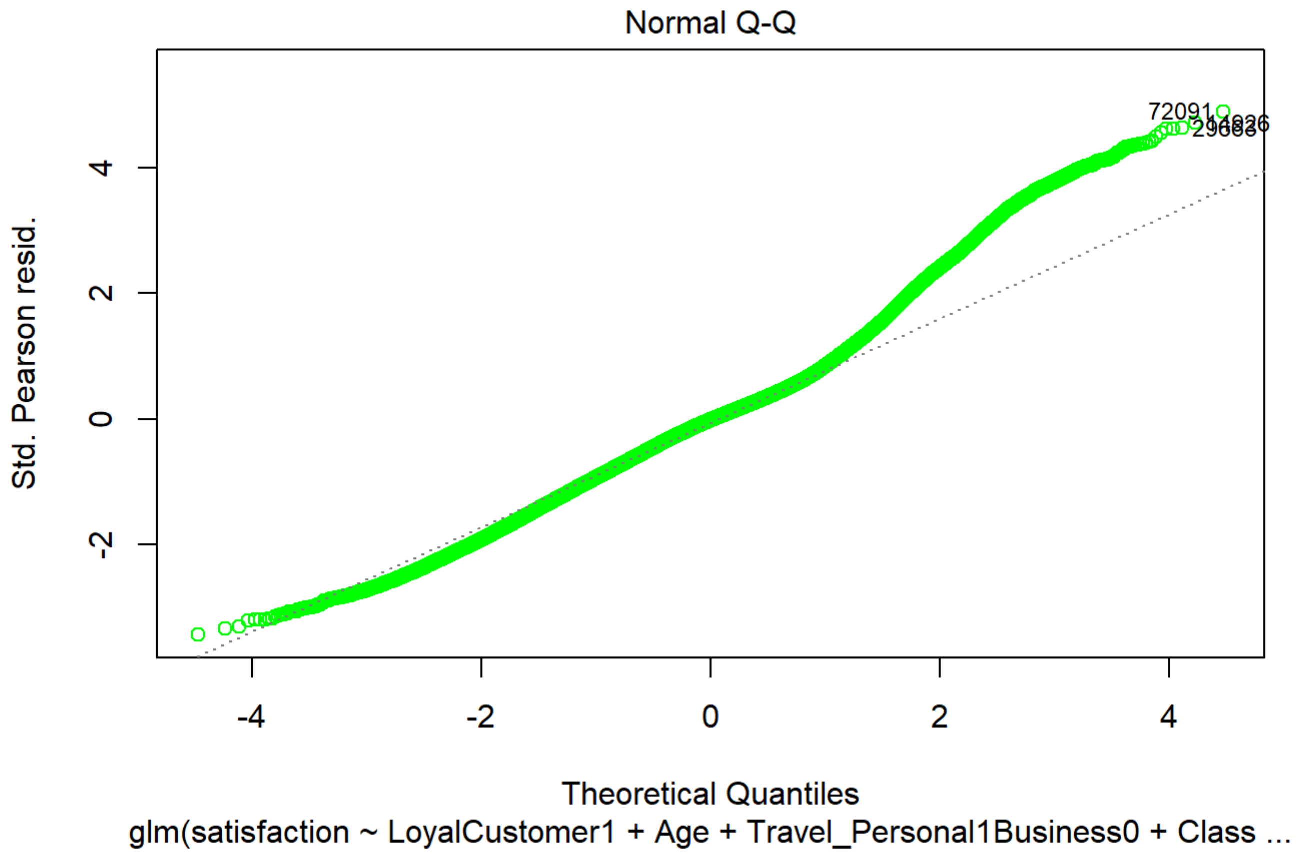
Logistic Regression Model

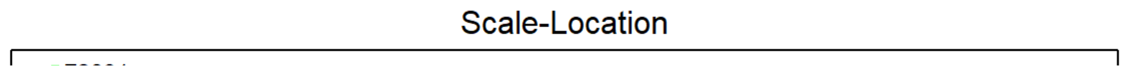
```
temp<-data.frame(Finaldata[,2:26])
temp[,c(1:7,9:22,24)]<-scale(temp[,c(1:7,9:22,24)])
temp[,8]<-scale(log(temp[,8]))
temp[,23][temp[,23]>0]<-scale(log(temp[,23][temp[,23]>0]))
temp[,24][temp[,24]>0]<-scale(log(temp[,24][temp[,24]>0]))

l<-glm(satisfaction~LoyalCustomer1+Age+Travel_Personal1Business0+Class_Eco0+Class_Ecoplus1+
         Flight.Distance+Inflight.wifi.service+Departure.Arrival.time.convenient+Ease.of.Onli
ne.booking+
         Online.boarding+Seat.comfort+Inflight.entertainment+On.board.service+Leg.room.servic
e+Baggage.handling+
         Checkin.service+Inflight.service+Cleanliness+Departure.Delay.in.Minutes,data=temp)
summary(l)
```

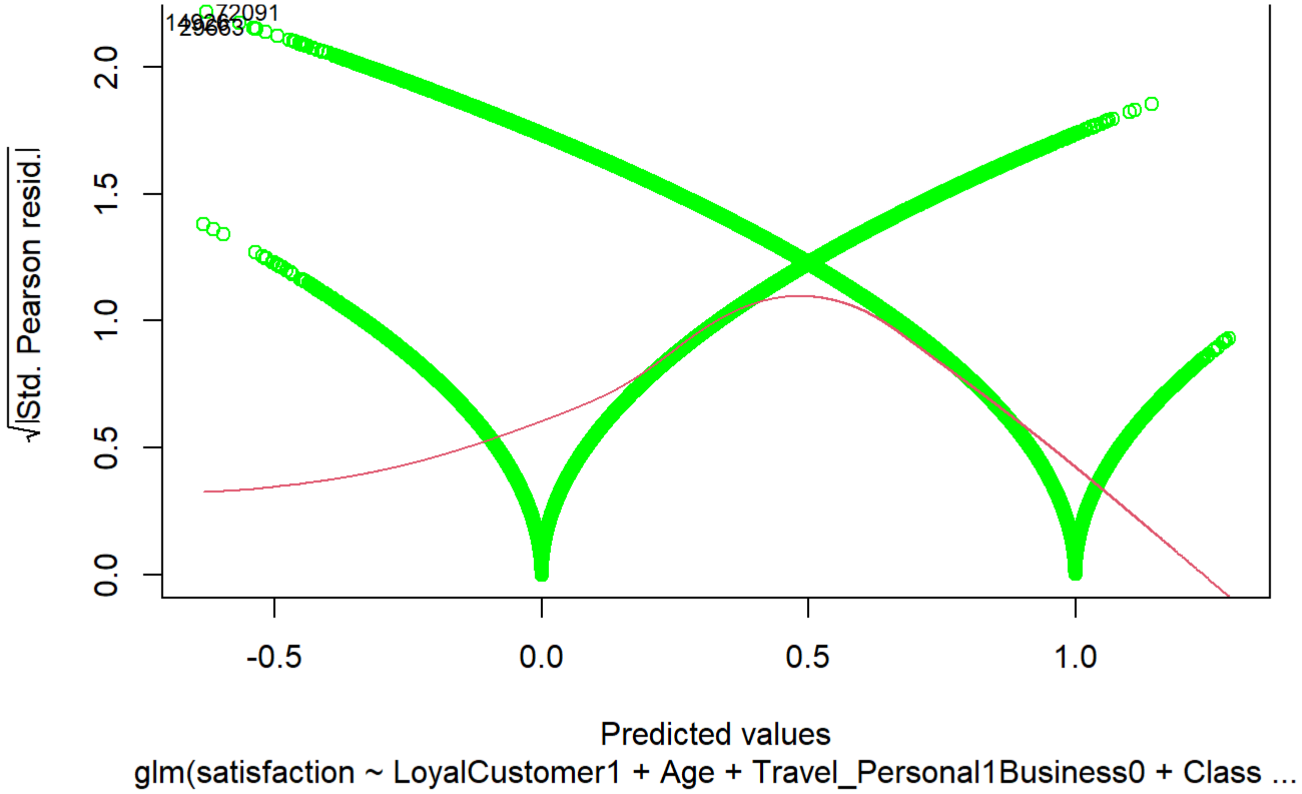
```
##
## Call:
## glm(formula = satisfaction ~ LoyalCustomer1 + Age + Travel_Personal1Business0 +
##     Class_Eco0 + Class_Ecoplus1 + Flight.Distance + Inflight.wifi.service +
##     Departure.Arrival.time.convenient + Ease.of.Online.booking +
##     Online.boarding + Seat.comfort + Inflight.entertainment +
##     On.board.service + Leg.room.service + Baggage.handling +
##     Checkin.service + Inflight.service + Cleanliness + Departure.Delay.in.Minutes,
##     data = temp)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.14299  -0.20726  -0.00556   0.16534   1.62852
##
## Coefficients:
##                                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                       0.4345307  0.0009223  471.159  < 2e-16 ***
## LoyalCustomer1                    0.1210130  0.0011459  105.605  < 2e-16 ***
## Age                              -0.0144105  0.0009931  -14.511  < 2e-16 ***
## Travel_Personal1Business0        -0.1778718  0.0013225 -134.493  < 2e-16 ***
## Class_Eco0                       -0.0639504  0.0013351  -47.900  < 2e-16 ***
## Class_Ecoplus1                   -0.0370856  0.0010469  -35.425  < 2e-16 ***
## Flight.Distance                  -0.0032500  0.0010334   -3.145  0.00166 **
## Inflight.wifi.service             0.0910466  0.0014412   63.174  < 2e-16 ***
## Departure.Arrival.time.convenient -0.0249938  0.0011366  -21.989  < 2e-16 ***
## Ease.of.Online.booking           -0.0556642  0.0014620  -38.074  < 2e-16 ***
## Online.boarding                   0.1057406  0.0012722   83.117  < 2e-16 ***
## Seat.comfort                      0.0096062  0.0014102    6.812 9.68e-12 ***
## Inflight.entertainment            0.0177250  0.0017034   10.406  < 2e-16 ***
## On.board.service                  0.0452373  0.0012211   37.045  < 2e-16 ***
## Leg.room.service                  0.0429300  0.0010538   40.739  < 2e-16 ***
## Baggage.handling                  0.0195647  0.0012712   15.391  < 2e-16 ***
## Checkin.service                   0.0472334  0.0010205   46.282  < 2e-16 ***
## Inflight.service                  0.0187174  0.0013231   14.146  < 2e-16 ***
## Cleanliness                       0.0327078  0.0015189   21.533  < 2e-16 ***
## Departure.Delay.in.Minutes       -0.0215424  0.0014010  -15.376  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1104108)
##
##     Null deviance: 31896  on 129808  degrees of freedom
## Residual deviance: 14330  on 129789  degrees of freedom
## AIC: 82364
##
## Number of Fisher Scoring iterations: 2
```

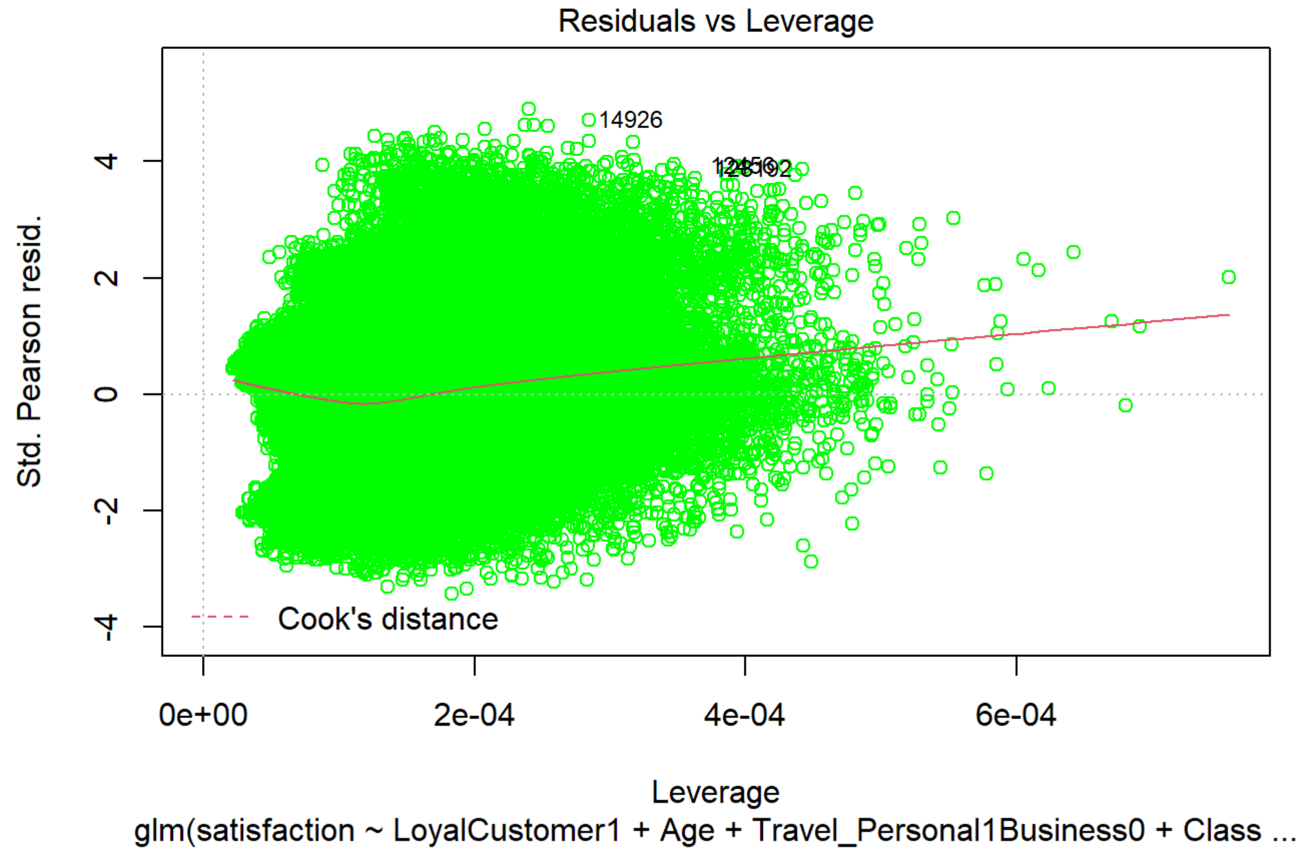
```
plot(l,col="green")
```


Residuals vs Fitted
Residuals
Predicted values
glm(satisfaction ~ LoyalCustomer1 + Age + Travel_Personal1Business0 + Class ...
72091
1.5
1.0
0.5
0.0
-0.5
-1.0
-0.5
0.0
0.5
1.0


Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(satisfaction ~ LoyalCustomer1 + Age + Travel_Personal1Business0 + Class ...
72091
4
2
0
-2
-4
-2
0
2
4


Scale-Location






```
#confusion matrix
check<-data.frame(l$fitted.values,temp$satisfaction)
a<-matrix(c(sum(check[,1]>0.5&check[,2]==1),sum(check[,1]>0.5&check[,2]==0),sum(check[,1]<0.5
&check[,2]==1),sum(check[,1]<0.5&check[,2]==0)))
matrix(a/(sum(c(sum(check[,1]>0.5&check[,2]==1),sum(check[,1]<0.5&check[,2]==0),sum(check[,1]
>0.5&check[,2]==0),sum(check[,1]<0.5&check[,2]==1)))),ncol=2,byrow=TRUE)
```

```
##             [,1]       [,2]
## [1,] 0.36284079 0.05784653
## [2,] 0.07168994 0.50762274
```

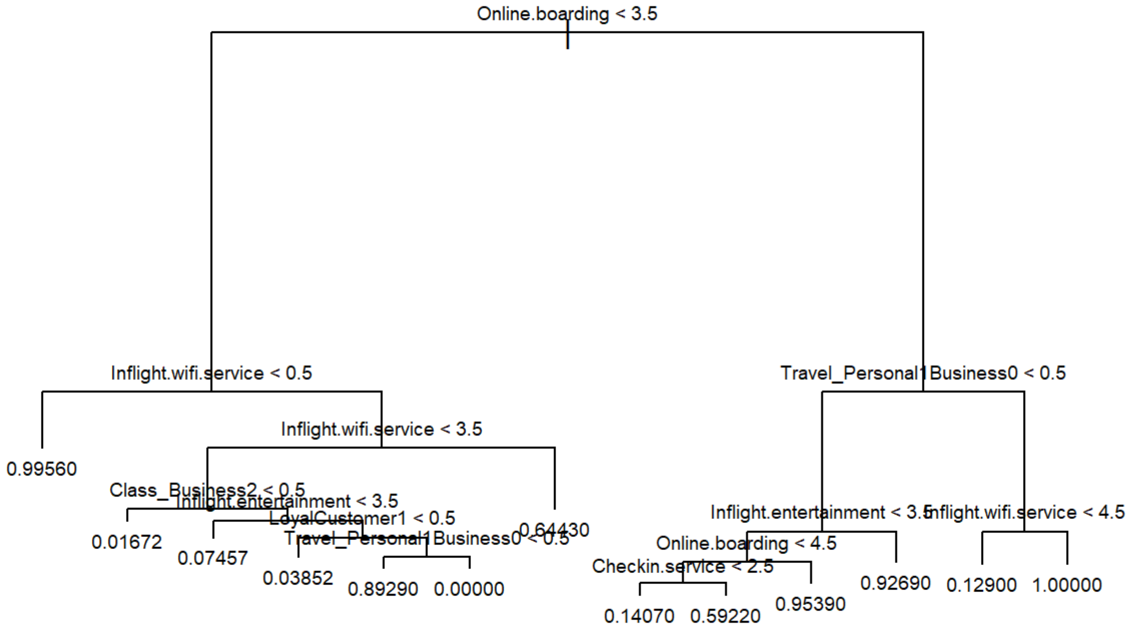
```
rm(l,a,check)
```

Decision Tree

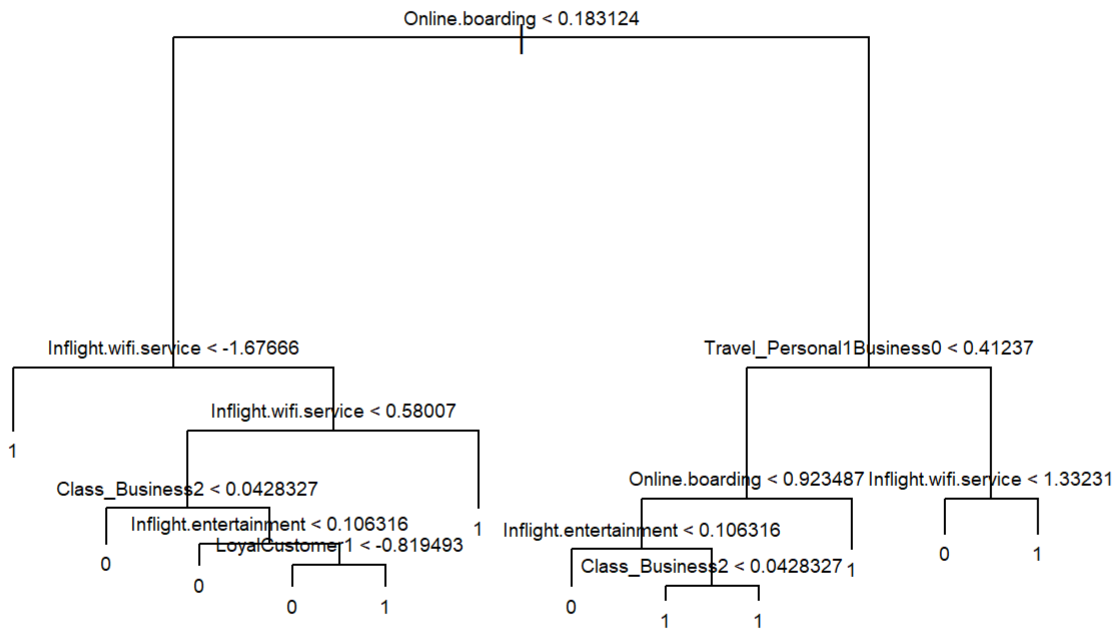
```
tree.satisf = tree(satisfaction~ . , data = Finaldata[,-1])
summary(tree.satisf)
```

```
##
## Regression tree:
## tree(formula = satisfaction ~ ., data = Finaldata[, -1])
## Variables actually used in tree construction:
## [1] "Online.boarding"           "Inflight.wifi.service"
## [3] "Class_Business2"           "Inflight.entertainment"
## [5] "LoyalCustomer1"            "Travel_Personal1Business0"
## [7] "Checkin.service"
## Number of terminal nodes:  13
## Residual mean deviance:  0.06986 = 9067 / 129800
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.99560 -0.03852 -0.01672  0.00000  0.07305  0.98330
```

```
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.6)
```




```
#Validation
set.seed(1011)
train = sample(1:nrow(temp), 50000)
tree.satisf = tree(as.factor(satisfaction) ~ . , temp, subset = train)
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.6)
```



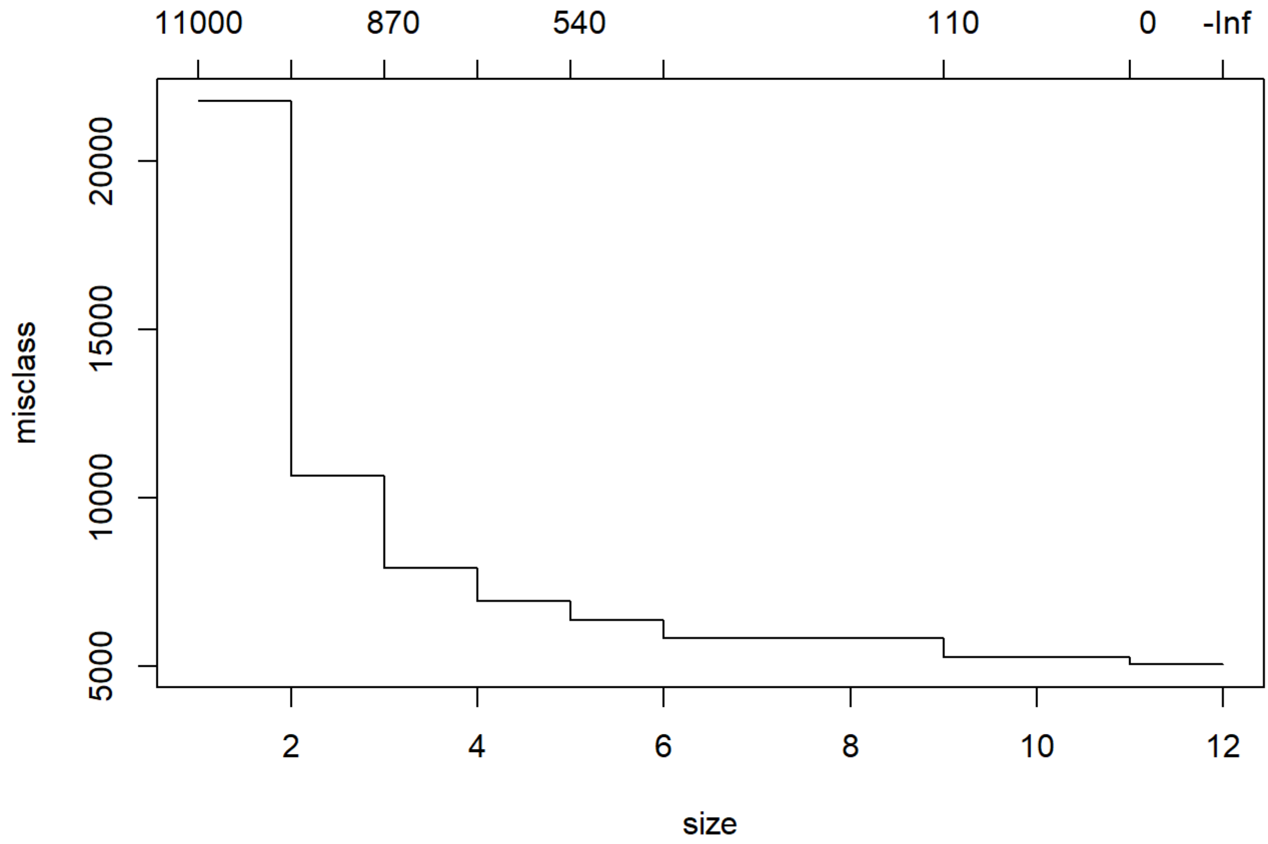

```
tree.pred = predict(tree.satisf, temp[-train, ], type = "class")
with(temp[-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 41508  4418
##         1  3669 30214
```

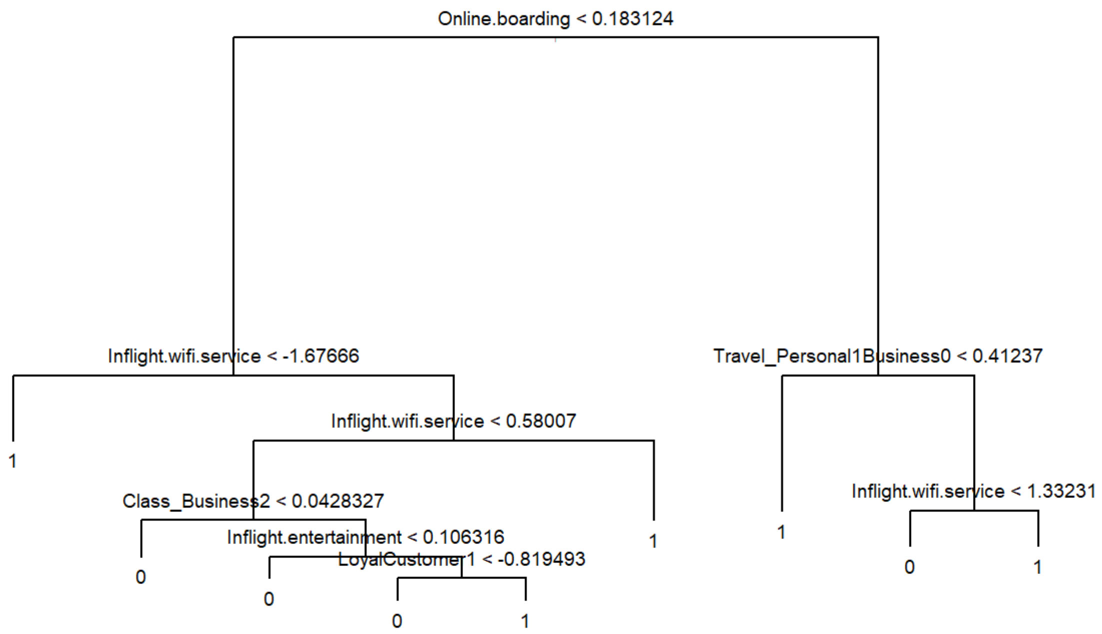
```
#Pruning for simplicity
cv.satisf = cv.tree(tree.satisf, FUN = prune.misclass)
cv.satisf
```

```
## $size
## [1] 12 11  9  6  5  4  3  2  1
##
## $dev
## [1]  5042  5042  5254  5820  6365  6927  7898 10642 21774
##
## $k
## [1]       -Inf     0.0000   111.5000   189.6667   545.0000   666.0000   867.0000
## [8]  2744.0000 11132.0000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.satisf)
```




```
prune.satisf = prune.misclass(tree.satisf, best = 8)
plot(prune.satisf,cex=0.2)
text(prune.satisf,pretty=0,cex=0.6)
```




```
tree.pred = predict(prune.satisf, temp[-train, ], type = "class")
with(temp[-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 38721  1900
##         1  6456 32732
```

```
rm(tree.satisf,prune.satisf,cv.satisf,tree.pred,train)
```

Random Forest

```
temp<-Finaldata[,c(-1,-25)]
temp$satisfaction<-as.factor(Finaldata$satisfaction)
set.seed(101)
train = sample(1:nrow(Finaldata),10000)
rf.satisf = randomForest(satisfaction ~ ., data = temp, subset = train,ntree=1000)
rf.satisf
```

```
## 
## Call:
##  randomForest(formula = satisfaction ~ ., data = temp, ntree = 1000,      subset = train) 
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
## 
##         OOB estimate of  error rate: 5.48%
## Confusion matrix:
##      0    1 class.error
## 0 5400  200  0.03571429
## 1  348 4052  0.07909091
```

Gradient Boosting Machine

```
boost.satisf = gbm(as.factor(satisfaction) ~ ., data = Finaldata[,-1][train, ],distribution = 
"multinomial",
                   n.trees = 1000, shrinkage = 0.01, interaction.depth = 9)
```

```
## Warning: Setting `distribution = "multinomial"` is ill-advised as it is
## currently broken. It exists only for backwards compatibility. Use at your own
## risk.
```

```
predmat = predict(boost.satisf, newdata = Finaldata[,-1][-train, ], n.trees = 1000,type="resp
onse")
labels = colnames(predmat)[apply(predmat, 1, which.max)]
result = data.frame(Finaldata[,-1][-train, ]$satisfaction, labels)
cm = confusionMatrix(as.factor(Finaldata[,-1][-train, ]$satisfaction), as.factor(labels))
print(cm)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 65540  2263
##          1  3669 48337
##
##                Accuracy : 0.9505
##                  95% CI : (0.9492, 0.9517)
##     No Information Rate : 0.5777
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8989
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9470
##             Specificity : 0.9553
##          Pos Pred Value : 0.9666
##          Neg Pred Value : 0.9295
##              Prevalence : 0.5777
##          Detection Rate : 0.5470
##    Detection Prevalence : 0.5659
##       Balanced Accuracy : 0.9511
##
##        'Positive' Class : 0
##
```

```
rm(labels,result,cm,predmat,train,boost.satisf)
```

Insights from Data (Decision tree is used for Interpretablity)

```
require(plyr)
```

```
## Loading required package: plyr
```

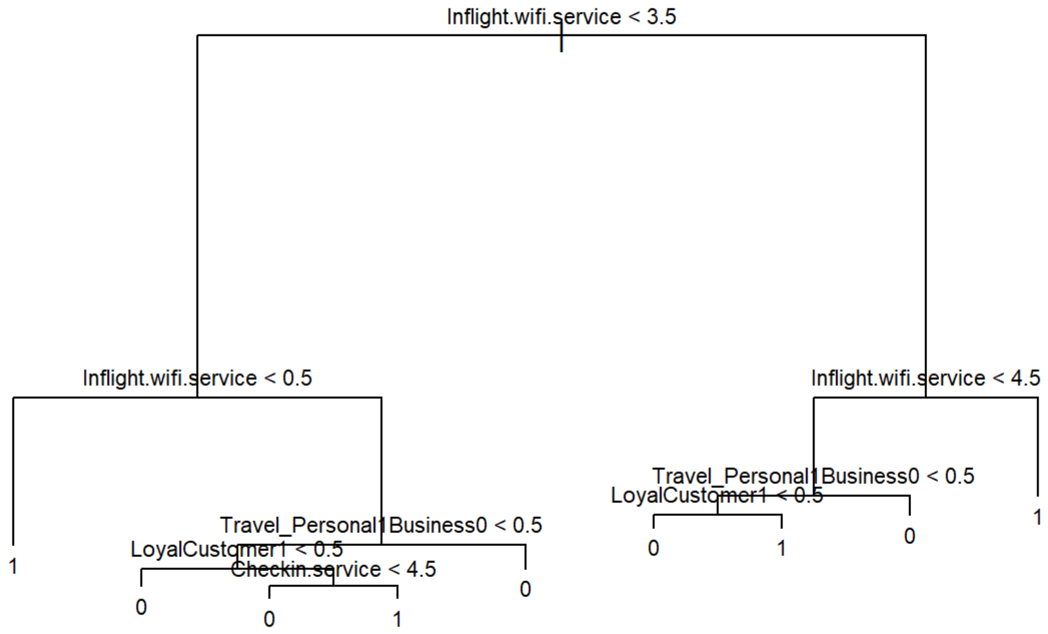
```
## Warning: package 'plyr' was built under R version 4.0.2
```

```
FData<-join(finalinsightdata,Finaldata,type="inner")
```

```
## Joining by: Customer.ID
```

Cluster 1

```
FDataclust1<-subset(FData,FData$k.cluster==1)


#Decision Tree

set.seed(1011)
train = sample(1:nrow(FDataclust1[,c(-1,-2)]), 5000)
tree.satisf = tree(as.factor(satisfaction) ~ . , FDataclust1[,c(-1,-2)], subset = train)
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.7)
```



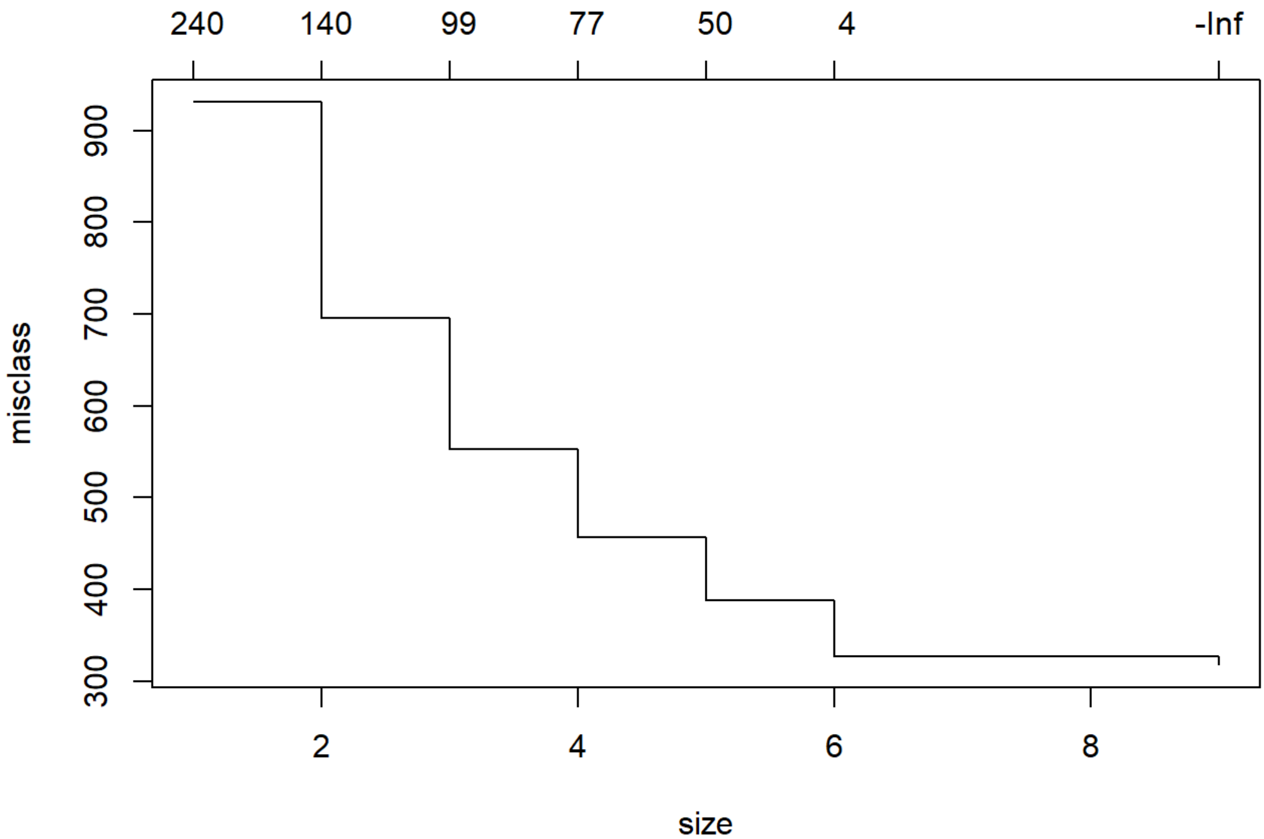

```
tree.pred = predict(tree.satisf, FDataclust1[,c(-1,-2)][-train, ], type = "class")
with(FDataclust1[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 21611  1240
##         1   398  3669
```

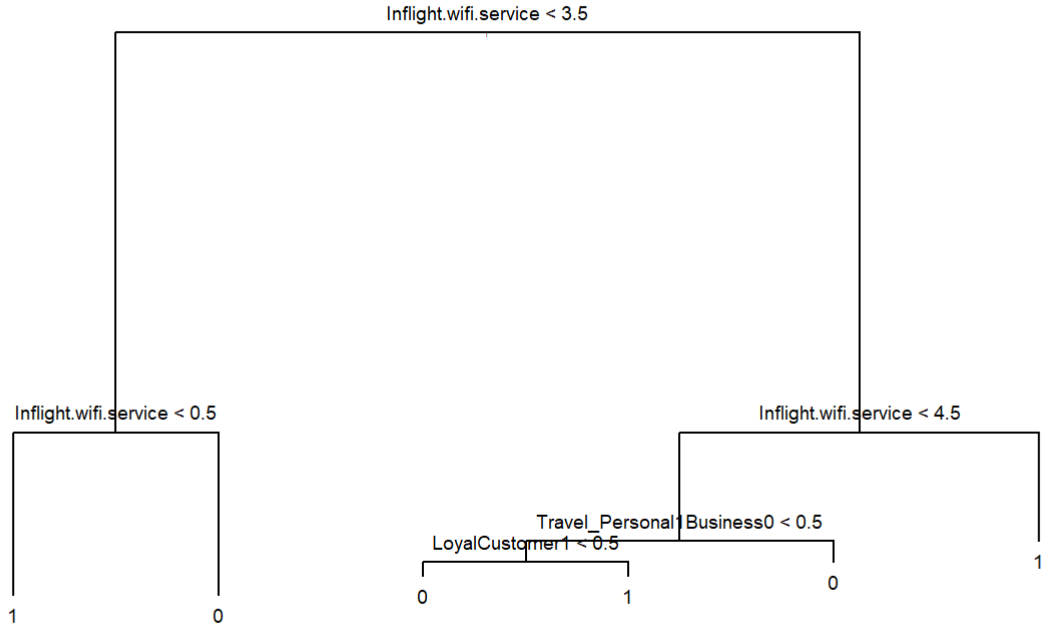
```
#Pruning for simplicity
cv.satisf = cv.tree(tree.satisf, FUN = prune.misclass)
cv.satisf
```

```
## $size
## [1] 9 6 5 4 3 2 1
##
## $dev
## [1] 318 327 388 457 553 696 931
##
## $k
## [1] -Inf    4   50   77   99  141  237
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.satisf)
```




```
prune.satisf = prune.misclass(tree.satisf, best = 6)
plot(prune.satisf,cex=0.2)
text(prune.satisf,pretty=0,cex=0.6)
```



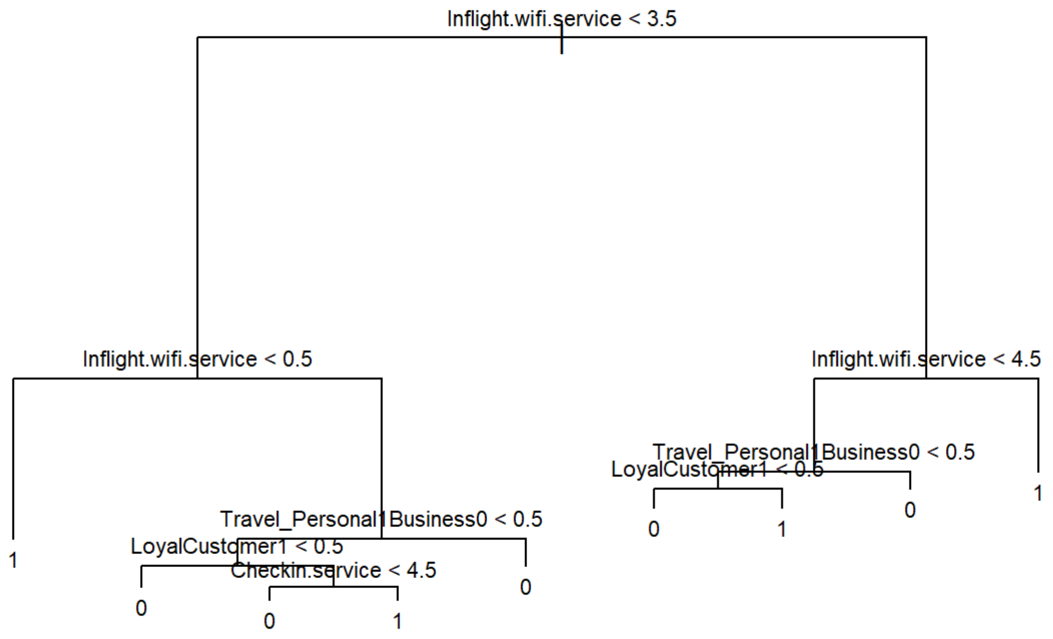

```
tree.pred = predict(prune.satisf, FDataclust1[,c(-1,-2)][-train, ], type = "class")
with(FDataclust1[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 21612  1308
##         1   397  3601
```

```
rm(tree.satisf,prune.satisf,cv.satisf,tree.pred,train)
```

Cluster 2

```
FDataclust2<-subset(FData,FData$k.cluster==2)


#Decision Tree

set.seed(1011)
train = sample(1:nrow(FDataclust2[,c(-1,-2)]), 5000)
tree.satisf = tree(as.factor(satisfaction) ~ . , FDataclust2[,c(-1,-2)], subset = train)
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.7)
```



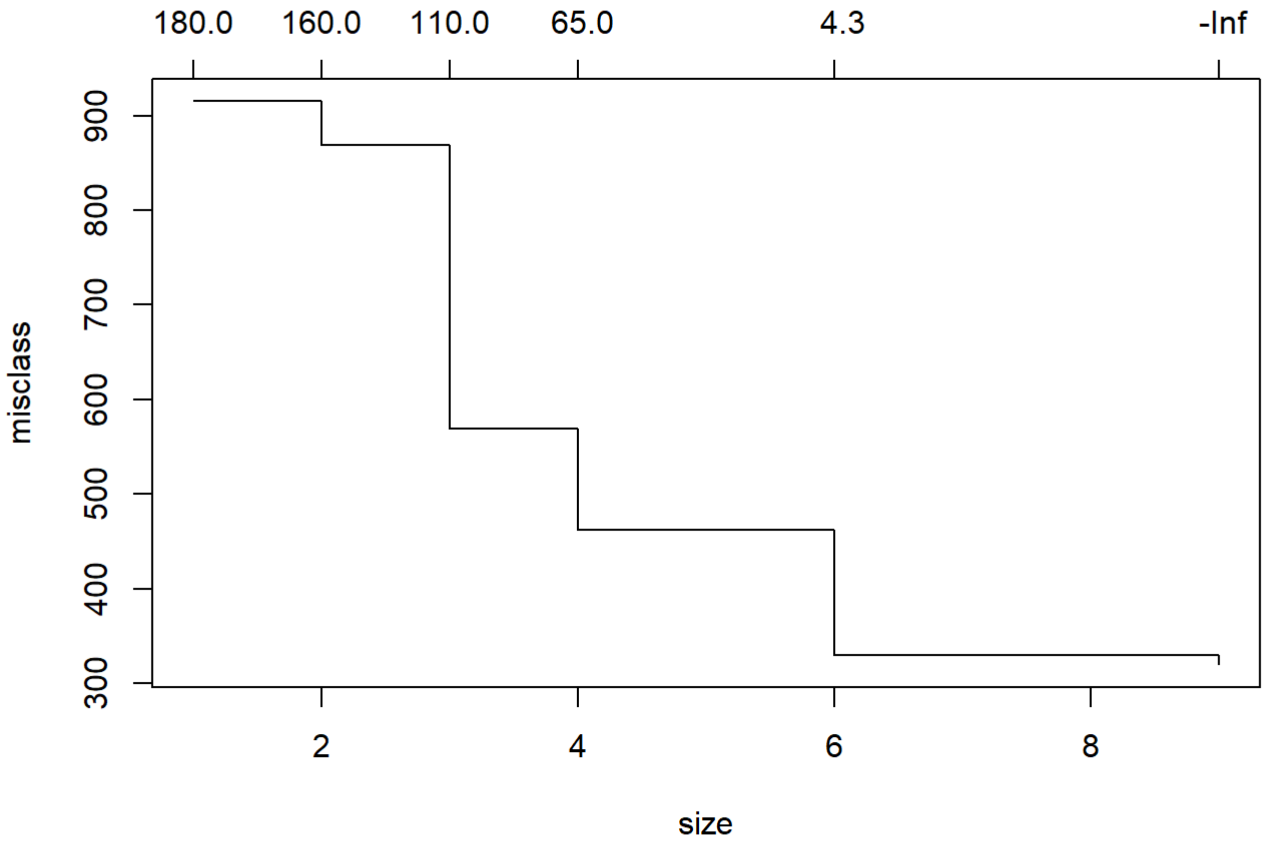

```
tree.pred = predict(tree.satisf, FDataclust2[,c(-1,-2)][-train, ], type = "class")
with(FDataclust2[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 23674  1405
##         1   477  4225
```

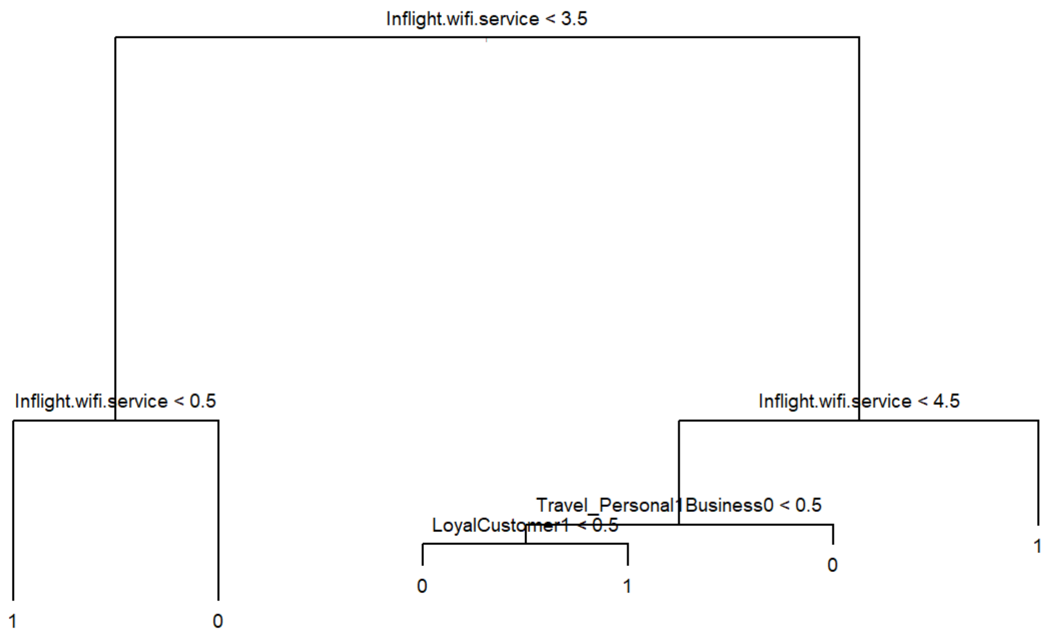
```
#Pruning for simplicity
cv.satisf = cv.tree(tree.satisf, FUN = prune.misclass)
cv.satisf
```

```
## $size
## [1] 9 6 4 3 2 1
##
## $dev
## [1] 320 330 462 569 869 915
##
## $k
## [1]       -Inf   4.333333  65.000000 107.000000 160.000000 179.000000
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.satisf)
```




```
prune.satisf = prune.misclass(tree.satisf, best = 6)
plot(prune.satisf,cex=0.2)
text(prune.satisf,pretty=0,cex=0.6)
```



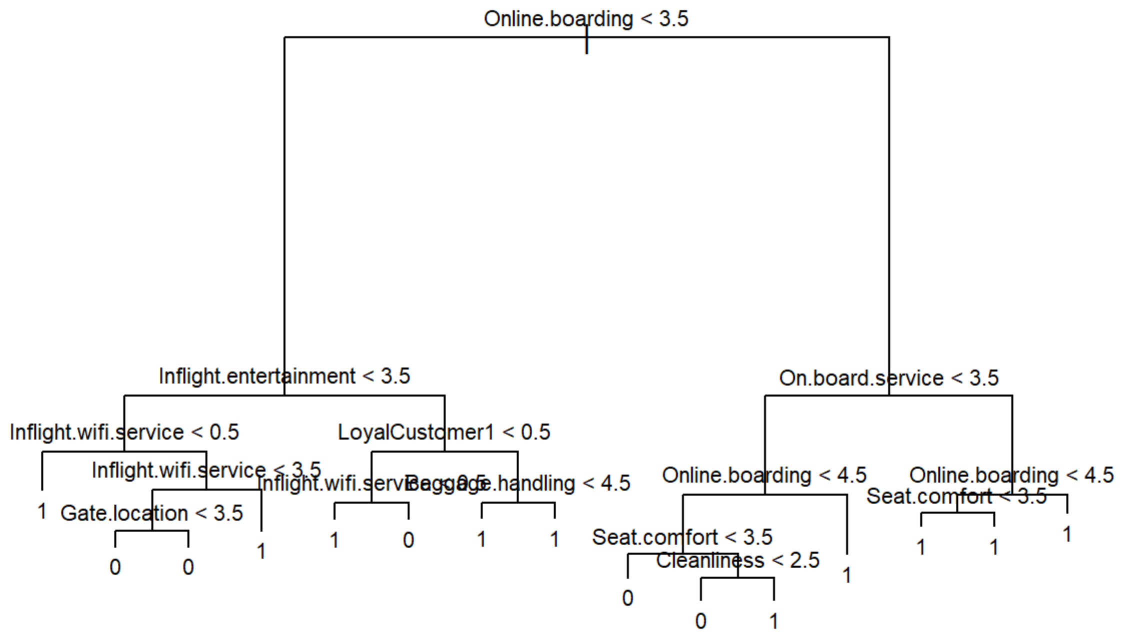

```
tree.pred = predict(prune.satisf, FDataclust2[,c(-1,-2)][-train, ], type = "class")
with(FDataclust2[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0 23674  1483
##         1   477  4147
```

```
rm(tree.satisf,prune.satisf,cv.satisf,tree.pred,train)
```

Cluster 3

```
FDataclust3<-subset(FData,FData$k.cluster==3)


#Decision Tree

set.seed(1011)
train = sample(1:nrow(FDataclust3[,c(-1,-2)]), 5000)
tree.satisf = tree(as.factor(satisfaction) ~ . , FDataclust3[,c(-1,-2)], subset = train)
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.7)
```



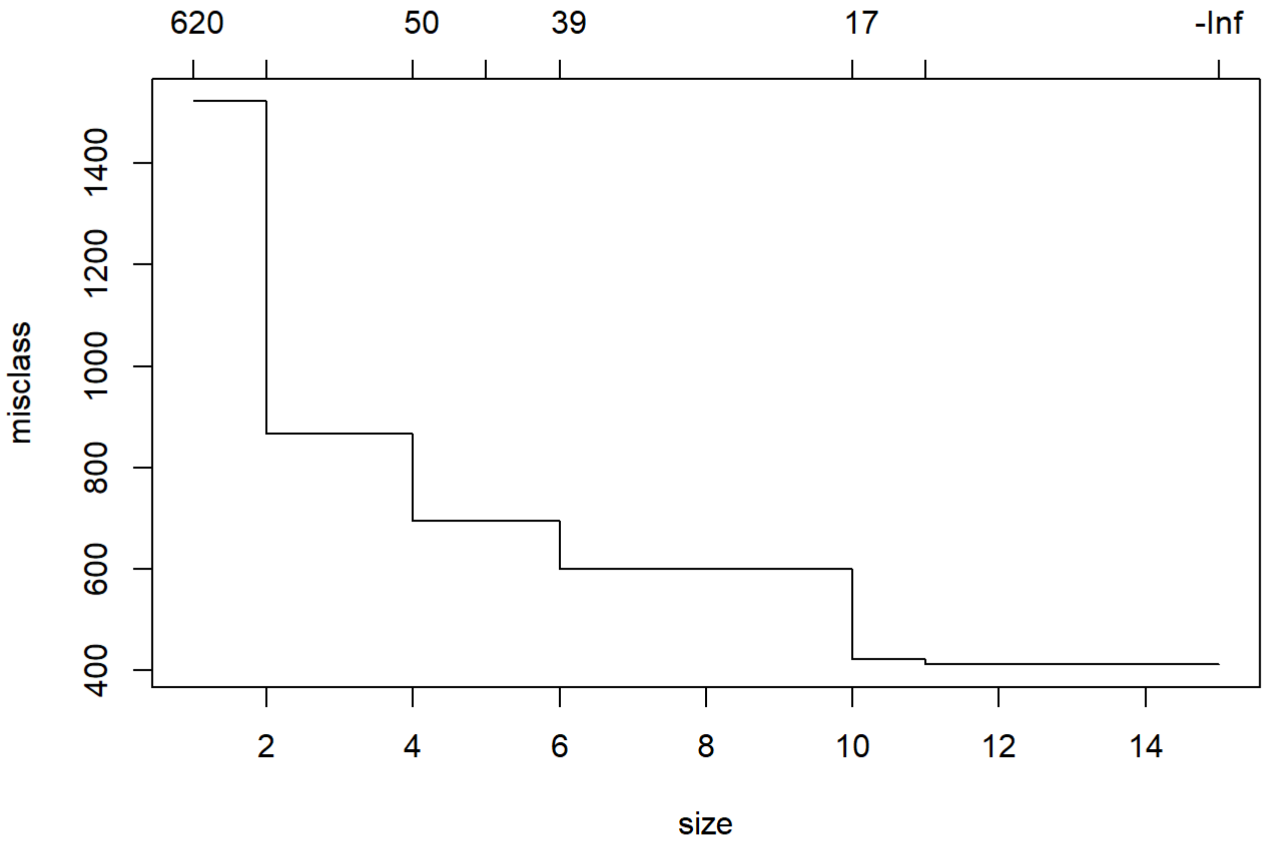

```
tree.pred = predict(tree.satisf, FDataclust3[,c(-1,-2)][-train, ], type = "class")
with(FDataclust3[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0  7017   766
##         1  1680 19403
```

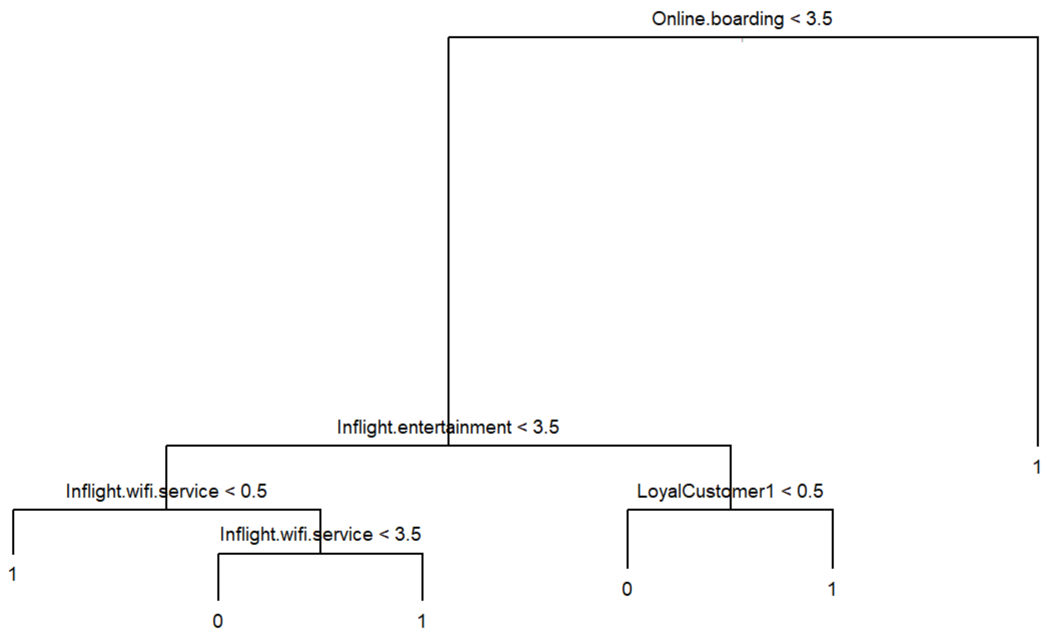
```
#Pruning for simplicity
cv.satisf = cv.tree(tree.satisf, FUN = prune.misclass)
cv.satisf
```

```
## $size
## [1] 15 11 10  6  5  4  2  1
##
## $dev
## [1]  412  412  423  601  695  695  867 1522
##
## $k
## [1]   -Inf   0.00  17.00  38.75  47.00  50.00 115.50 619.00
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.satisf)
```




```
prune.satisf = prune.misclass(tree.satisf, best = 6)
plot(prune.satisf,cex=0.2)
text(prune.satisf,pretty=0,cex=0.6)
```



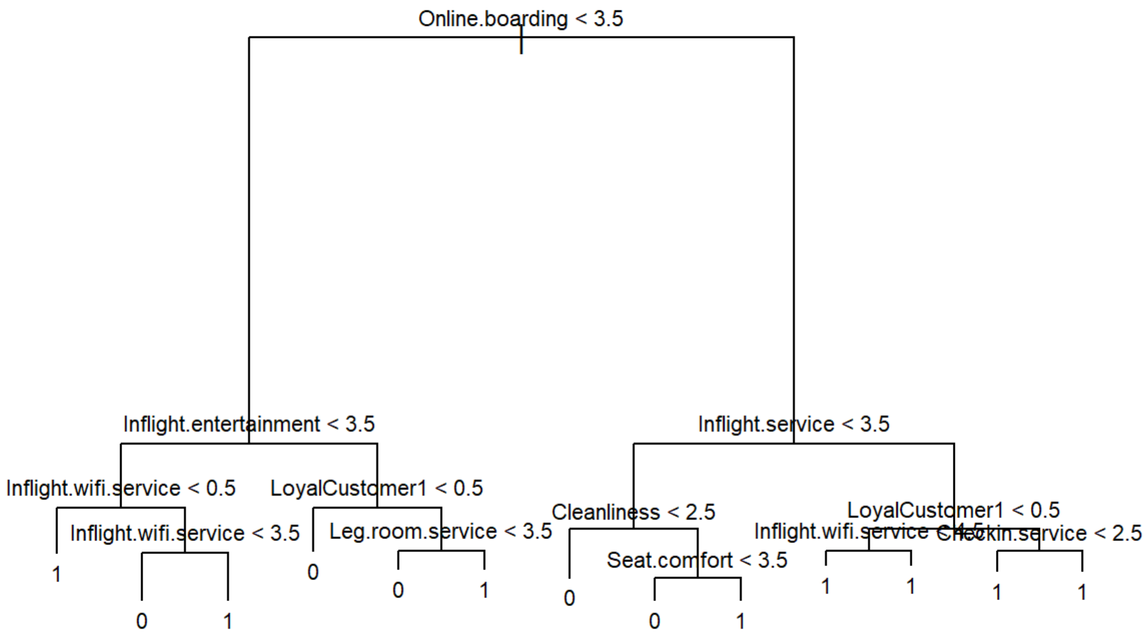

```
tree.pred = predict(prune.satisf, FDataclust3[,c(-1,-2)][-train, ], type = "class")
with(FDataclust3[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0  6105   581
##         1  2592 19588
```

```
rm(tree.satisf,prune.satisf,cv.satisf,tree.pred,train)
```

Cluster 4

```
FDataclust4<-subset(FData,FData$k.cluster==4)


#Decision Tree

set.seed(1011)
train = sample(1:nrow(FDataclust4[,c(-1,-2)]), 5000)
tree.satisf = tree(as.factor(satisfaction) ~ . , FDataclust4[,c(-1,-2)], subset = train)
plot(tree.satisf)
text(tree.satisf, pretty = 0,cex=0.7)
```



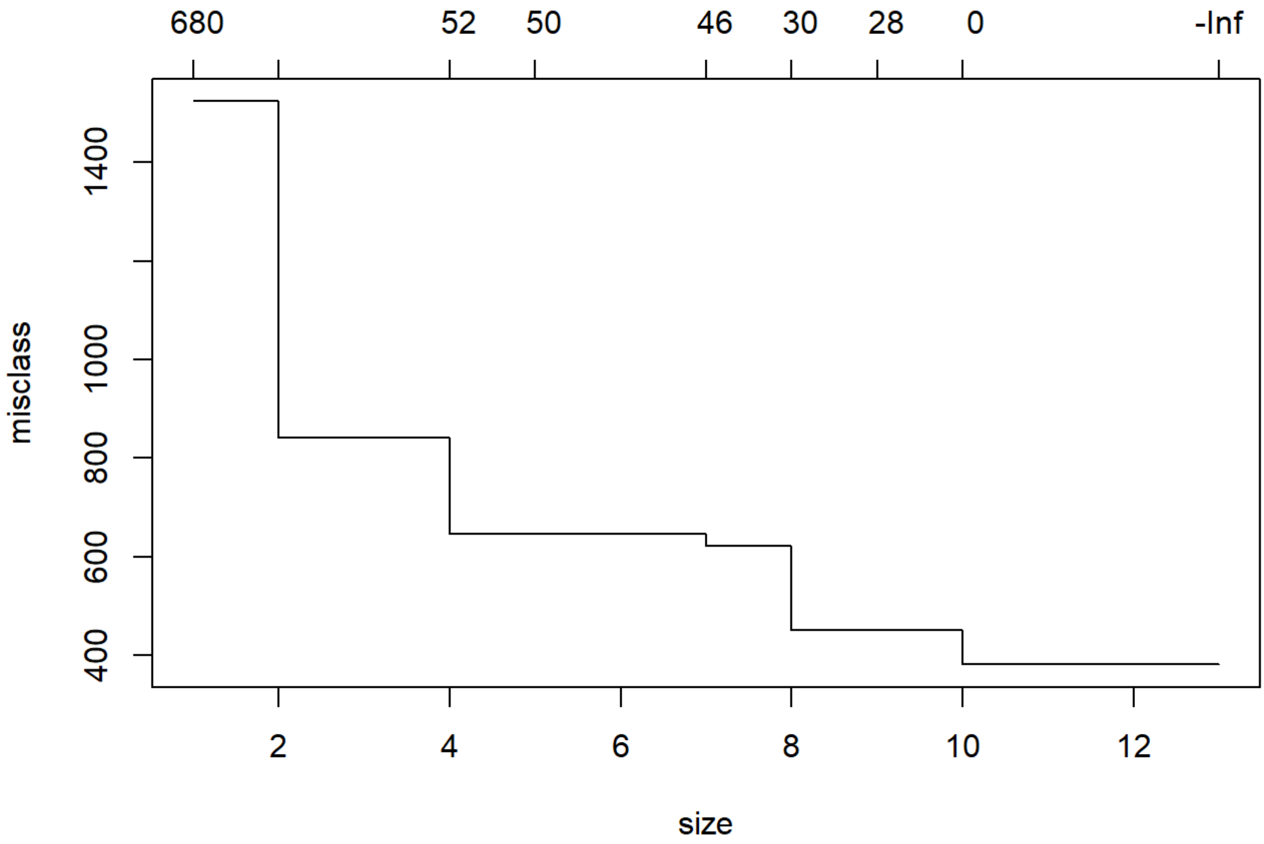

```
tree.pred = predict(tree.satisf, FDataclust4[,c(-1,-2)][-train, ], type = "class")
with(FDataclust4[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0  5980   748
##         1  1330 16135
```

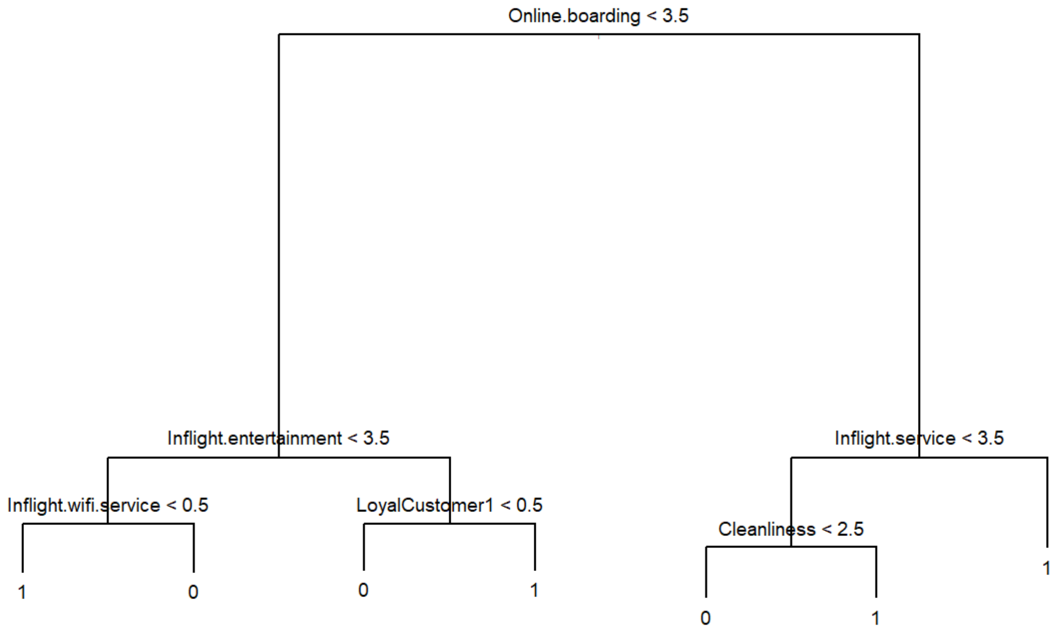
```
#Pruning for simplicity
cv.satisf = cv.tree(tree.satisf, FUN = prune.misclass)
cv.satisf
```

```
## $size
## [1] 13 10  9  8  7  5  4  2  1
##
## $dev
## [1]  381  381  450  450  621  646  646  842 1524
##
## $k
## [1] -Inf    0   28   30   46   50   52  100  682
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv.satisf)
```




```
prune.satisf = prune.misclass(tree.satisf, best = 6 )
plot(prune.satisf,cex=0.2)
text(prune.satisf,pretty=0,cex=0.6)
```




```
tree.pred = predict(prune.satisf, FDataclust4[,c(-1,-2)][-train, ], type = "class")
with(FDataclust4[,c(-1,-2)][-train, ], table(tree.pred, satisfaction))
```

```
##          satisfaction
## tree.pred     0     1
##         0  5620   858
##         1  1690 16025
```

```
rm(tree.satisf,prune.satisf,cv.satisf,tree.pred,train)
```

Comparison between clusters

```
#All clusters included

mean(FData$Inflight.wifi.service)
```

```
## [1] 2.728911
```

```
mean(FData$Online.boarding)
```

```
## [1] 3.252678
```

```
mean(FData$Inflight.entertainment)
```

```
## [1] 3.358267
```

```
sqrt(var(FData$Inflight.wifi.service))
```

```
## [1] 1.329365
```

```
sqrt(var(FData$Online.boarding))
```

```
## [1] 1.35068
```

```
sqrt(var(FData$Inflight.entertainment))
```

```
## [1] 1.334072
```

```
#Inflight WiFi
mean(FDataclust1$Inflight.wifi.service)
```

```
## [1] 2.667617
```

```
mean(FDataclust2$Inflight.wifi.service)
```

```
## [1] 2.67051
```

```
mean(FDataclust3$Inflight.wifi.service)
```

```
## [1] 2.803845
```

```
mean(FDataclust4$Inflight.wifi.service)
```

```
## [1] 2.778577
```

```
sqrt(var(FDataclust1$Inflight.wifi.service))
```

```
## [1] 1.21894
```

```
sqrt(var(FDataclust2$Inflight.wifi.service))
```

```
## [1] 1.224135
```

```
sqrt(var(FDataclust3$Inflight.wifi.service))
```

```
## [1] 1.431645
```

```
sqrt(var(FDataclust4$Inflight.wifi.service))
```

```
## [1] 1.431939
```

```
#Online Boarding
mean(FDataclust1$Online.boarding)
```

```
## [1] 2.83185
```

```
mean(FDataclust2$Online.boarding)
```

```
## [1] 2.83399
```

```
mean(FDataclust3$Online.boarding)
```

```
## [1] 3.695063
```

```
mean(FDataclust4$Online.boarding)
```

```
## [1] 3.698421
```

```
sqrt(var(FDataclust1$Online.boarding))
```

```
## [1] 1.328309
```

```
sqrt(var(FDataclust2$Online.boarding))
```

```
## [1] 1.324445
```

```
sqrt(var(FDataclust3$Online.boarding))
```

```
## [1] 1.227976
```

```
sqrt(var(FDataclust4$Online.boarding))
```

```
## [1] 1.229837
```

```
#In flight Entertainment

mean(FDataclust1$Inflight.entertainment)
```

```
## [1] 3.08973
```

```
mean(FDataclust2$Inflight.entertainment)
```

```
## [1] 3.105287
```

```
mean(FDataclust3$Inflight.entertainment)
```

```
## [1] 3.633497
```

```
mean(FDataclust4$Inflight.entertainment)
```

```
## [1] 3.633988
```

```
sqrt(var(FDataclust1$Inflight.entertainment))
```

```
## [1] 1.369794
```

```
sqrt(var(FDataclust2$Inflight.entertainment))
```

```
## [1] 1.366426
```

```
sqrt(var(FDataclust3$Inflight.entertainment))
```

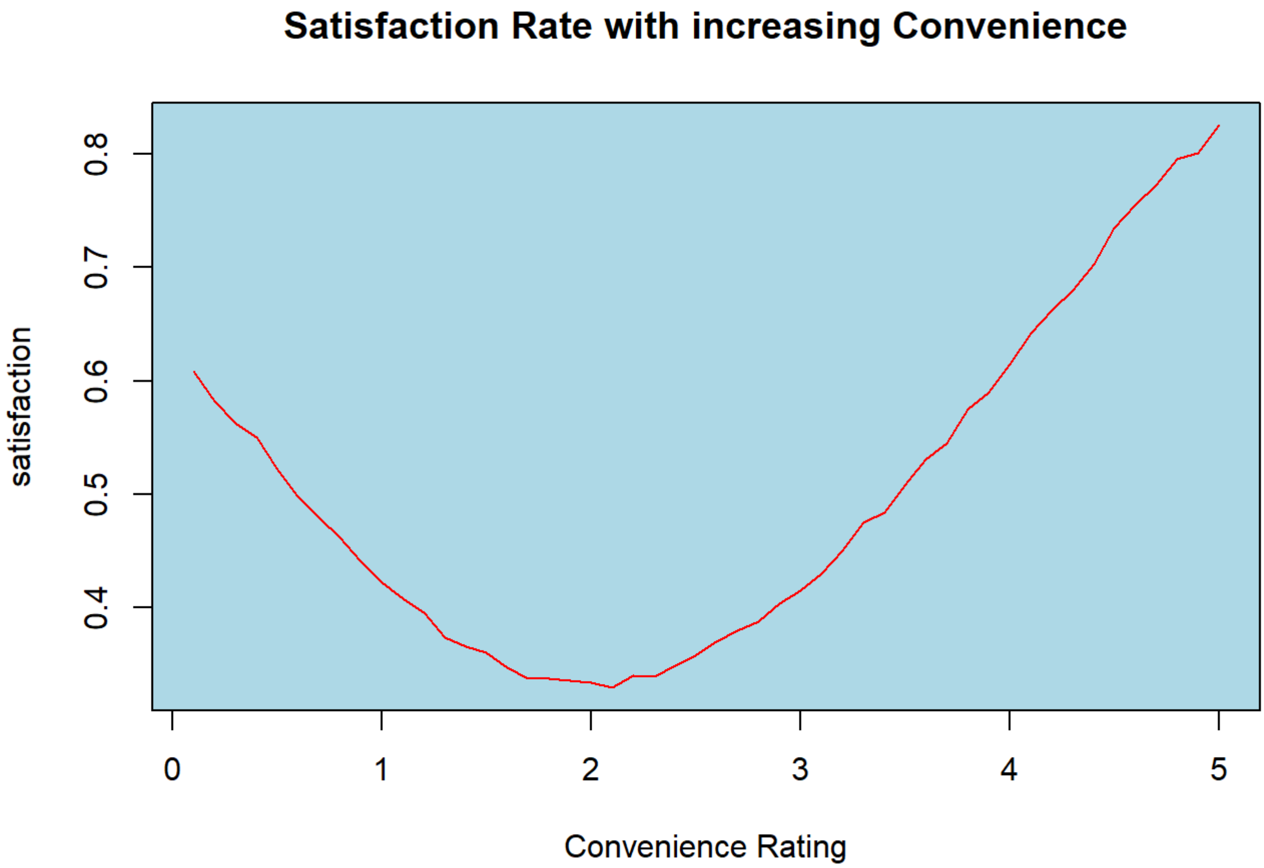
```
## [1] 1.239144
```

```
sqrt(var(FDataclust4$Inflight.entertainment))
```

```
## [1] 1.238791
```

Recommendation

```
Recdn<-FData[1:10000,c(-1,-2)]

satisfaction<-c(1:50)
for (i in 1:50)
{
Recdn$Inflight.wifi.service<-rnorm(nrow(Recdn),i/10,sqrt(var(FData$Inflight.wifi.service)))
Recdn$Online.boarding<-rnorm(nrow(Recdn),i/10,sqrt(var(FData$Online.boarding)))
Recdn$Inflight.entertainment<-rnorm(nrow(Recdn),i/10,sqrt(var(FData$Inflight.entertainment)))
predtree<-predict(rf.satisf,Recdn)
s<-summary(predtree)
satisfaction[i]<-s[2]/sum(s)
}
plot(seq(0.1,5,by=0.1),satisfaction,type="l",col="red",main="Satisfaction Rate with increasin
g Convenience",xlab = "Convenience Rating")
rect(par("usr")[1],par("usr")[3],par("usr")[2],par("usr")[4],col = "light blue")
points(seq(0.1,5,by=0.1),satisfaction,type="l",col="red")
```



Satisfaction Rate with increasing Convenience
satisfaction
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
Convenience Rating

The End