0.Load data and libraries

```
#Please install the following libraries if not
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(mde)
```

```
## Welcome to mde. This is mde version 0.3.2.
##  Please file issues and feedback at https://www.github.com/Nelson-Gon/mde/issues
## Turn this message off using 'suppressPackageStartupMessages(library(mde))'
##  Happy Exploration :)
```

```
library(hrbrthemes)
```

```
## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.
##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and
##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```
library(tidyverse)
library(ggridges)
library(ggthemes)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:ggthemes':
##
##     theme_map
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
source("DataAnalyticsFunctions.R")
source("PerformanceCurves.R")
df <- read.csv('Invistico_Airline.csv') # load data
```

1. Business Understanding

Airline Passenger Satisfaction Analysis

Business Problem:

Before delving into building the model, it is important to understand why it is essential for airline companies to improve customers' satisfaction level. The level of customer satisfaction is a crucial determinant of whether or not a consumer will book a flight with a certain airline. Therefore, understanding what influences customer happiness and what makes consumers more satisfied is essential for airliners. ***More importantly, in this project, we want to find out that how we can change satisfaction level to maximize airline companies' profit.***

Variable Description:

The dependent variable of our data set is satisfaction, which is binary variable [satisfied, not satisfied]. We have some demographic variables about airline passengers as well as some subjective question regarding their satisfaction level of some feature out of scale 1-5 (0 as not applicable). The details of each variable description is at below

Gender: Gender of the passengers (Female, Male)

Customer Type: The customer type (Loyal customer, disloyal customer)

Age: The actual age of the passengers

Type of Travel: Purpose of the flight of the passengers (Personal Travel, Business Travel)

Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Flight distance: The flight distance of this journey

Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not Applicable;1-5)

Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient

Ease of Online booking: Satisfaction level of online booking

Gate location: Satisfaction level of Gate location

Food and drink: Satisfaction level of Food and drink

Online boarding: Satisfaction level of online boarding

Seat comfort: Satisfaction level of Seat comfort

Inflight entertainment: Satisfaction level of inflight entertainment

On-board service: Satisfaction level of On-board service

Leg room service: Satisfaction level of Leg room service

Baggage handling: Satisfaction level of baggage handling

Check-in service: Satisfaction level of Check-in service

Inflight service: Satisfaction level of inflight service

Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure

Arrival Delay in Minutes: Minutes delayed when Arrival

Satisfaction: Airline satisfaction level(Satisfaction, neutral or dissatisfaction)

2. Data Understanding (insights)

2.1 Depedent variable

```
#data cleaning
#drop Nulls
df <- df[!is.na(df$Arrival.Delay.in.Minutes),]
#dropped all the subjective satisfaction x variables which contain 0
df <- df[which(rowSums(df[,c('Seat.comfort','Departure.Arrival.time.convenient','Food.and.drink', "Gate


piecustomer<-c(sum(df$satisfaction=='satisfied'),sum(df$satisfaction!='satisfied'))
label <- c('satisfied','unsatisfied')
label<- paste(label, piecustomer)
pie(piecustomer,label,main = 'Number of Satisfied Customers VS Unsatisfied')
```


```
piecustomer
```

```
## [1] 64569 54686
```

Inference: We can see that number of satisfied and unsatisfied customers are similar with slightly more satisfied customers in the data set. Good news is that the label is not unbalanced, which is great for later logistic etc model building.

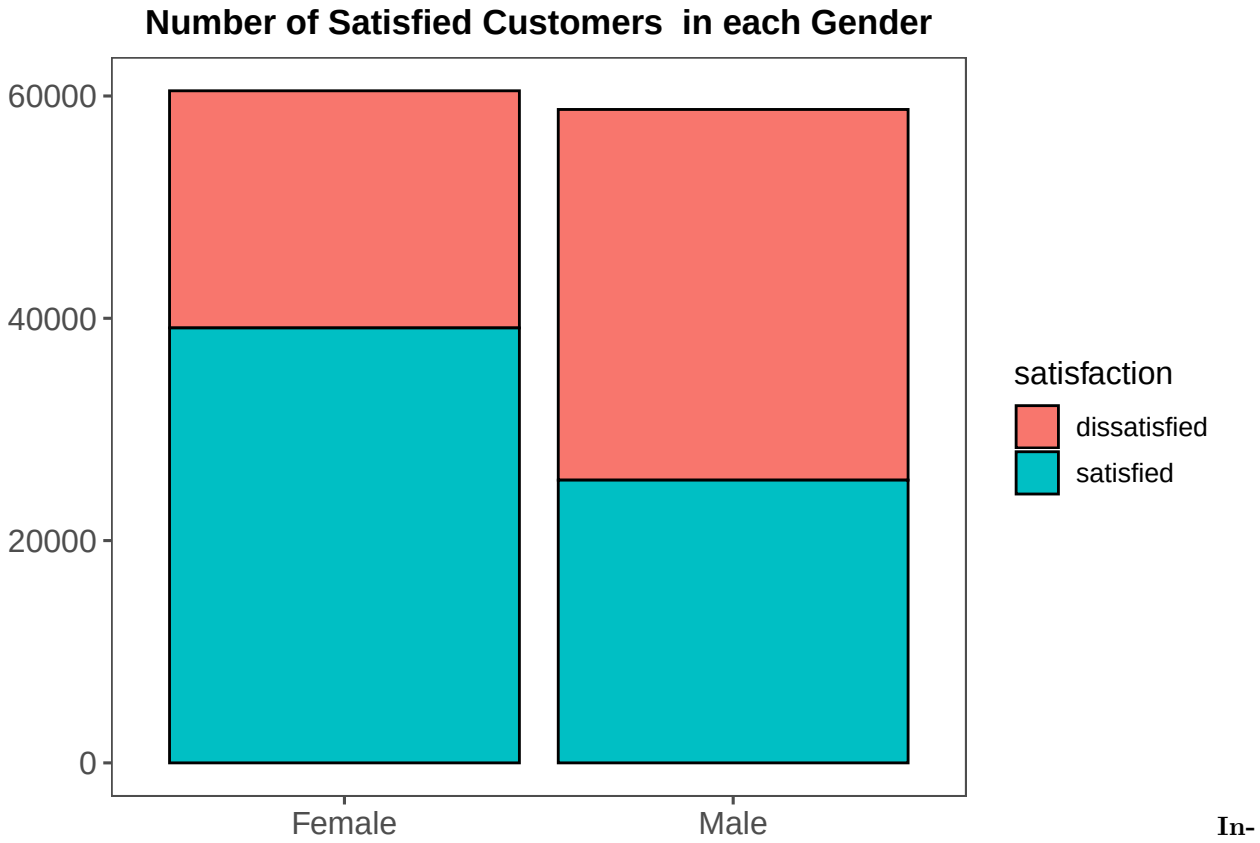
2.2 Overview of customers' demographic with satisfaction

Demographic Variables: age, gender

2.2.1 Gender

```
tema = theme(plot.title = element_text(size=13, hjust=.5,face='bold'),
             axis.text.x = element_text(size=12),
             axis.text.y = element_text(size=12))

ggplot(df,aes(x=Gender,fill=satisfaction))+geom_bar(color='black',state='identity')+labs(x=NULL,y=NULL,
```

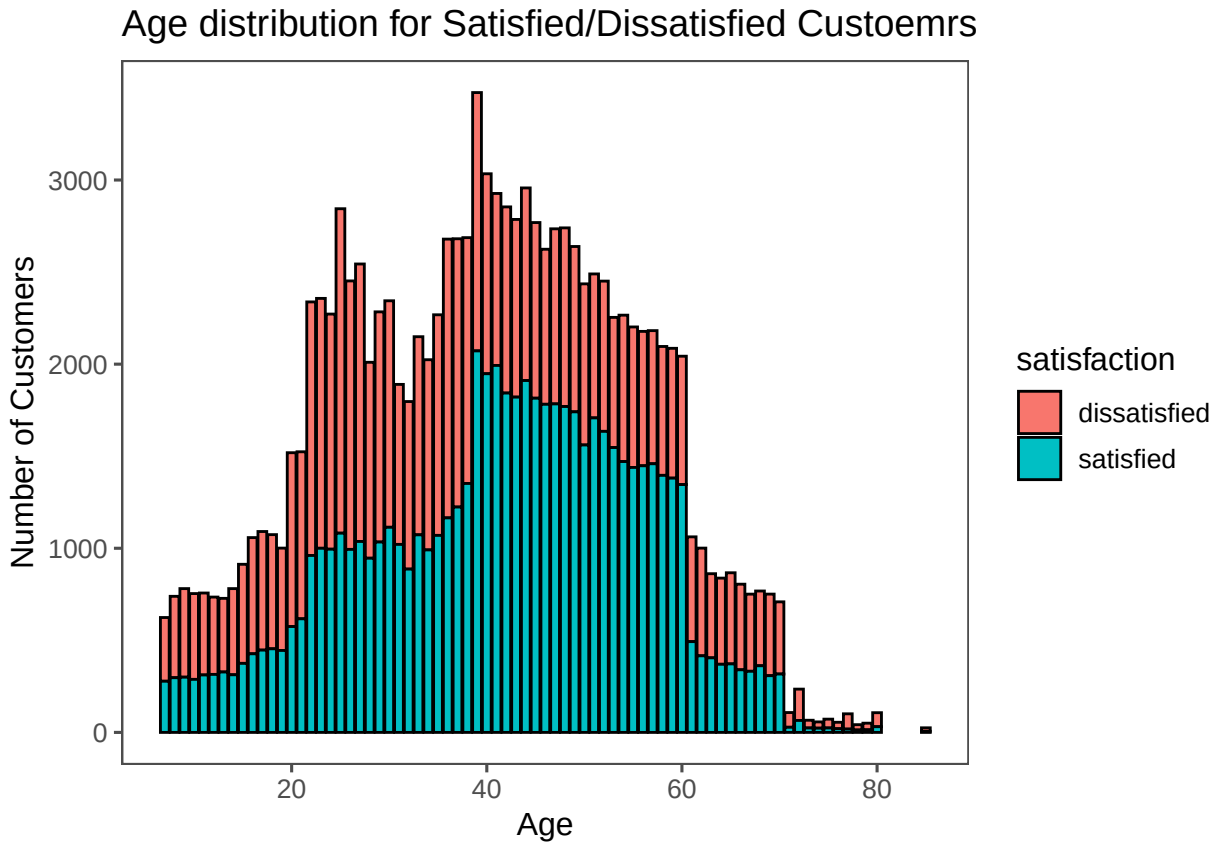
```
## Warning: Ignoring unknown parameters: state
```



Inference: We can see that each gender is well-representative and overall, female are more satisfied with the overall service of Airlines.

2.2.2 Age

```
tema = theme(plot.title = element_text(size=13, hjust=.5,face='bold'),
             axis.text.x = element_text(size=12),
             axis.text.y = element_text(size=12)) + theme_few()

ggplot(df,aes(x=Age,fill=satisfaction))+geom_bar(color='black') +labs(x='Age',y='Number of Customers',
```



Inference: We can see that people who are 25 to 60 are most representative in this data set. Customers who are older than 60 and younger than 20 are less representative in this data set. This difference could also be that the young and old groups don't travel much often compared to 20-60 age group and thus there are less satisfaction survey filled out.

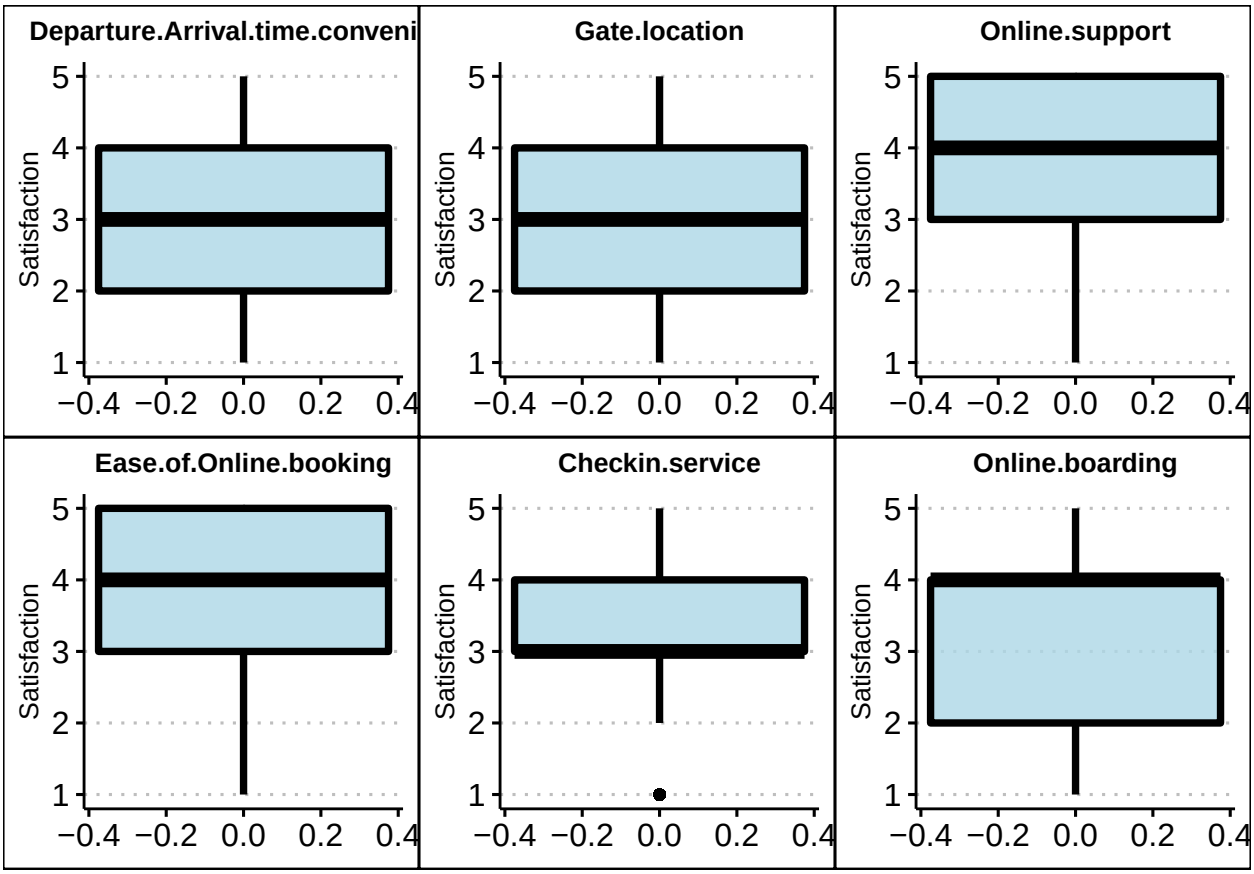
2.3 Overview of pre-flight with satisfaction

Pre-flight:

```
#Pick pre-flight subjective variables
pre.flight.subjective <- df %>% select(c('Departure.Arrival.time.convenient','Gate.location','Online.su
```

```
#Visualization
tema = theme(plot.title = element_text(size=10, hjust=.5),
             axis.text.x = element_text(size=12),
             axis.text.y = element_text(size=12))

theme_set(theme_clean())

Departure.Arrival.time.convenient <- ggplot(data = df, mapping = aes(x = Departure.Arrival.time.conveni
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Departure.Arrival.time.convenient',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Gate.location <- ggplot(data = df, mapping = aes(x = Gate.location)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Gate.location',x='Satisfaction',y=NULL) +
```

```
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Online.support <- ggplot(data = df, mapping = aes(x = Online.support)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Online.support',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Ease.of.Online.booking <- ggplot(data = df, mapping = aes(x = Ease.of.Online.booking)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Ease.of.Online.booking',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Checkin.service <- ggplot(data = df, mapping = aes(x = Checkin.service)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Checkin.service',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Online.boarding <- ggplot(data = df, mapping = aes(x = Online.boarding)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Online.boarding',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

plot_grid(Departure.Arrival.time.convenient,Gate.location,Online.support,Ease.of.Online.booking,Checkin
```



2.4 Overview of mid-flight with satisfaction

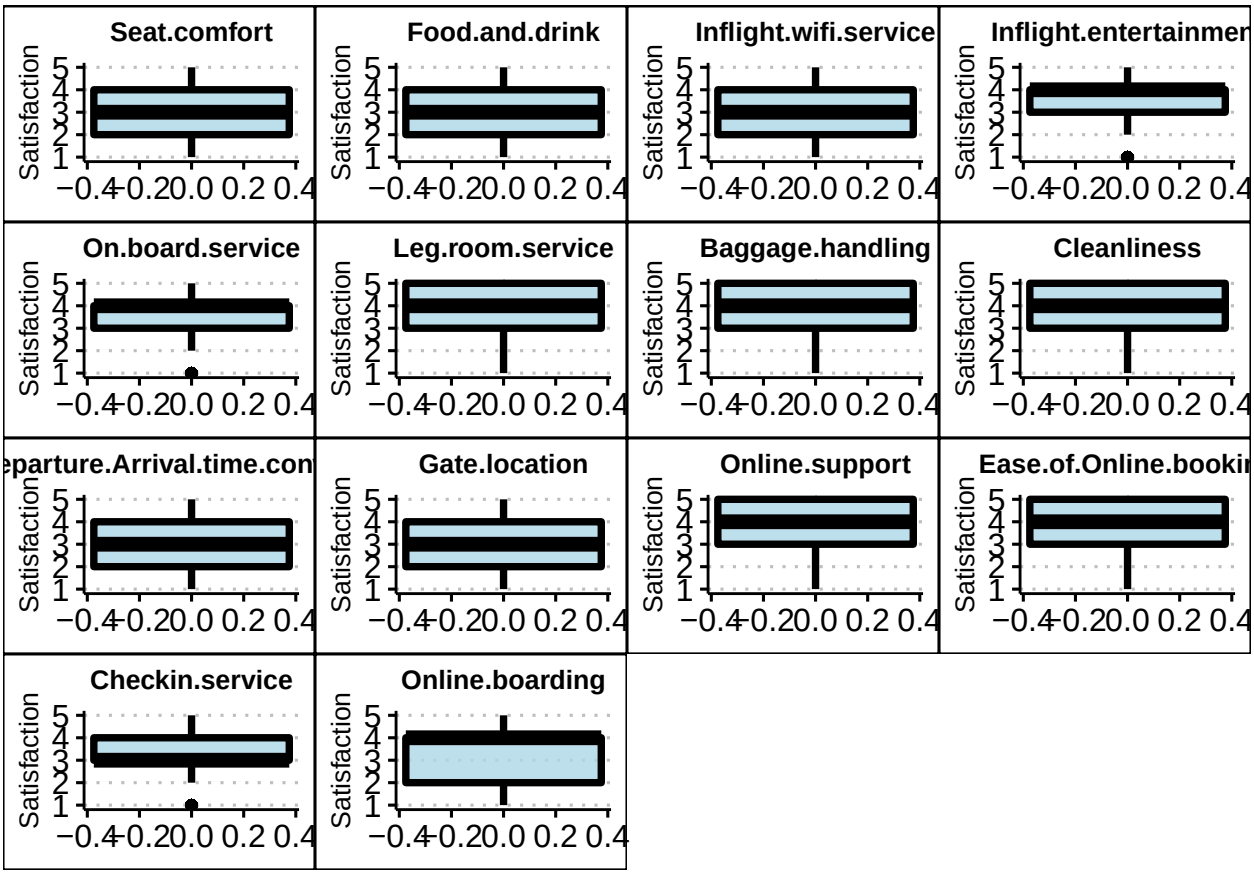
```
in.flight.subjective <- df[,!(colnames(df)  %in%  colnames(pre.flight.subjective))] %>% select(-c('satis

names(in.flight.subjective)
```

```
## [1] "Seat.comfort"           "Food.and.drink"         "Inflight.wifi.service"
## [4] "Inflight.entertainment" "On.board.service"       "Leg.room.service"
## [7] "Baggage.handling"       "Cleanliness"
```

```
#Visualization
tema = theme(plot.title = element_text(size=10, hjust=.5),
             axis.text.x = element_text(size=12),
             axis.text.y = element_text(size=12))

theme_set(theme_clean())

Seat.comfort <- ggplot(data = df, mapping = aes(x = Seat.comfort)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Seat.comfort',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Food.and.drink <- ggplot(data = df, mapping = aes(x = Food.and.drink)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Food.and.drink',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Inflight.wifi.service <- ggplot(data = df, mapping = aes(x = Inflight.wifi.service)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Inflight.wifi.service',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Inflight.entertainment <- ggplot(data = df, mapping = aes(x = Inflight.entertainment)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Inflight.entertainment',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

On.board.service <- ggplot(data = df, mapping = aes(x = On.board.service)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='On.board.service',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Leg.room.service <- ggplot(data = df, mapping = aes(x = Leg.room.service)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Leg.room.service',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()

Baggage.handling <- ggplot(data = df, mapping = aes(x = Baggage.handling)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Baggage.handling',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()
```

```
Cleanliness <- ggplot(data = df, mapping = aes(x = Cleanliness)) +
    geom_boxplot(fill = "light blue", color = "black", size = 1.3, alpha = .8) +
    labs (title='Cleanliness',x='Satisfaction',y=NULL) +
    theme(plot.title = element_text(hjust=.5,size=25)) + tema + coord_flip()
```

```
plot_grid(Seat.comfort,Food.and.drink,Inflight.wifi.service,Inflight.entertainment, On.board.service,Leg
```

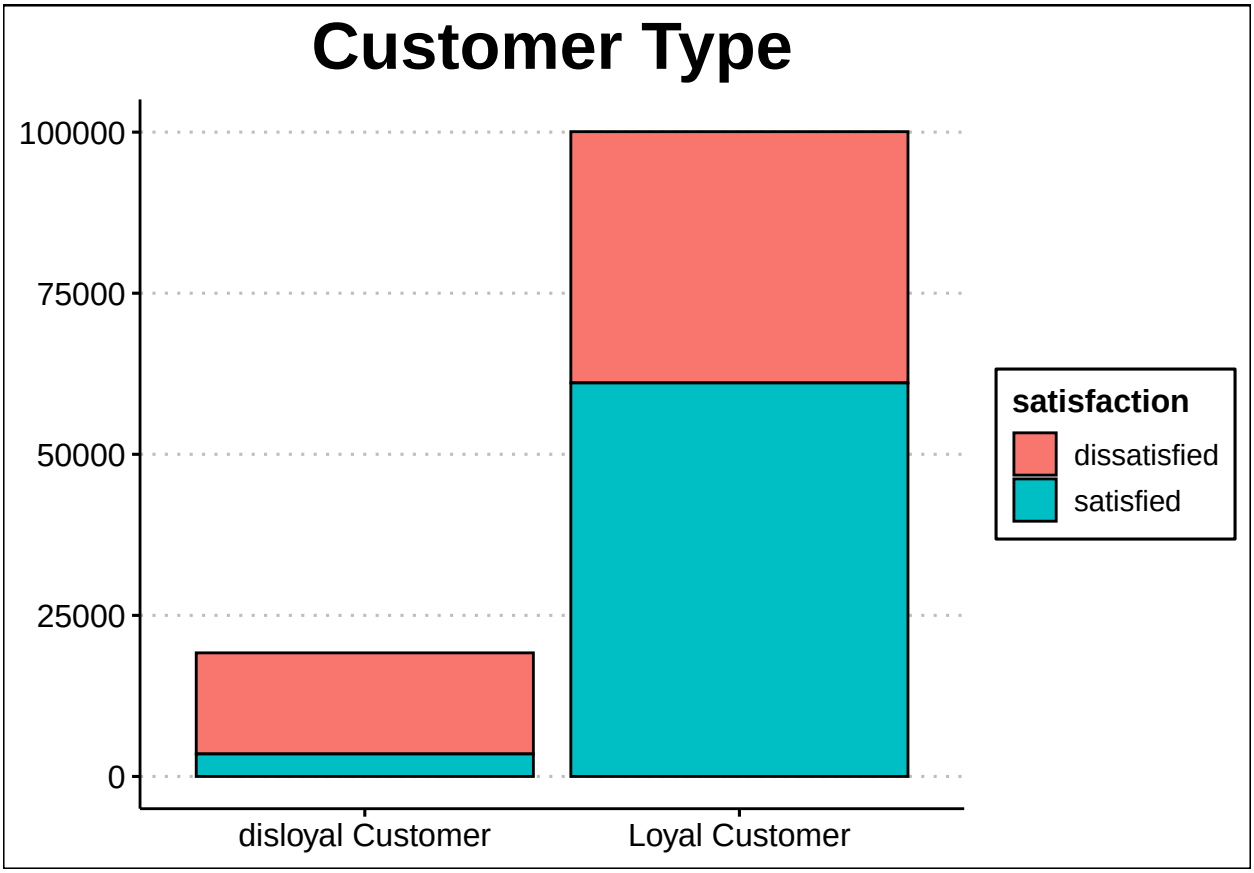


2.5 Other variables

```
others_1 <- df[,!(colnames(df)  %in% colnames(pre.flight.subjective))]
others <- others_1[,!(colnames(others_1)  %in% colnames(in.flight.subjective))]
```

2.5.1 Customer Types

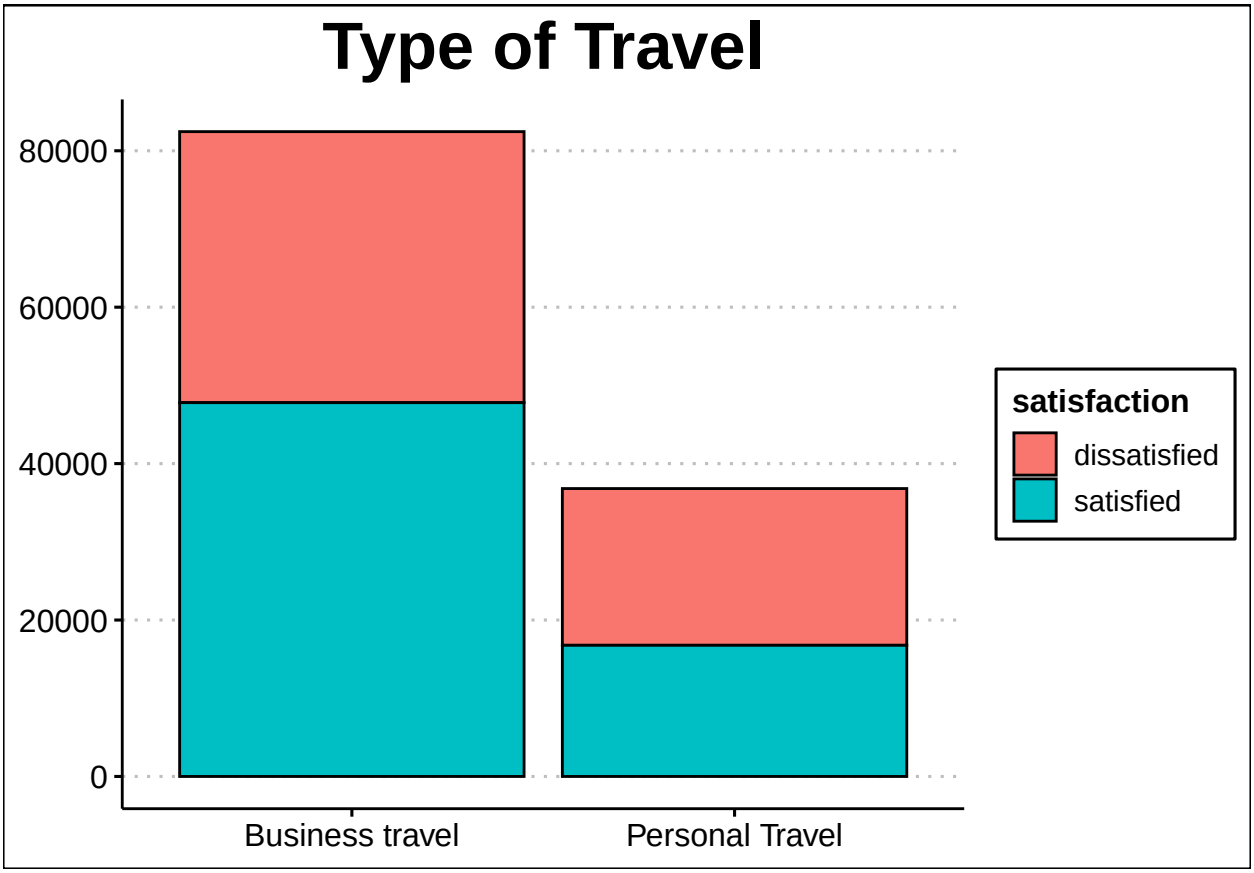
```
#Visualization
ggplot(others, aes(Customer.Type)) + geom_bar(aes(fill=satisfaction), color='black') + tema + labs (titl
    theme(plot.title = element_text(hjust=.5,size=25))
```



Inference: Even though there are more loyal customers than disloyal customers, we can still see that loyal customer tend to be more satisfied than disloyal customer. This graph might indicate that airline companies should try to convert more customers into loyalty program so as to increase the total satisfaction level of customers. Though there are many factors impacting the satisfaction that we need to consider, loyal customer can bring more revenue and are easier to retain compared to disloyal.

2.5.2 Type.of.Travel

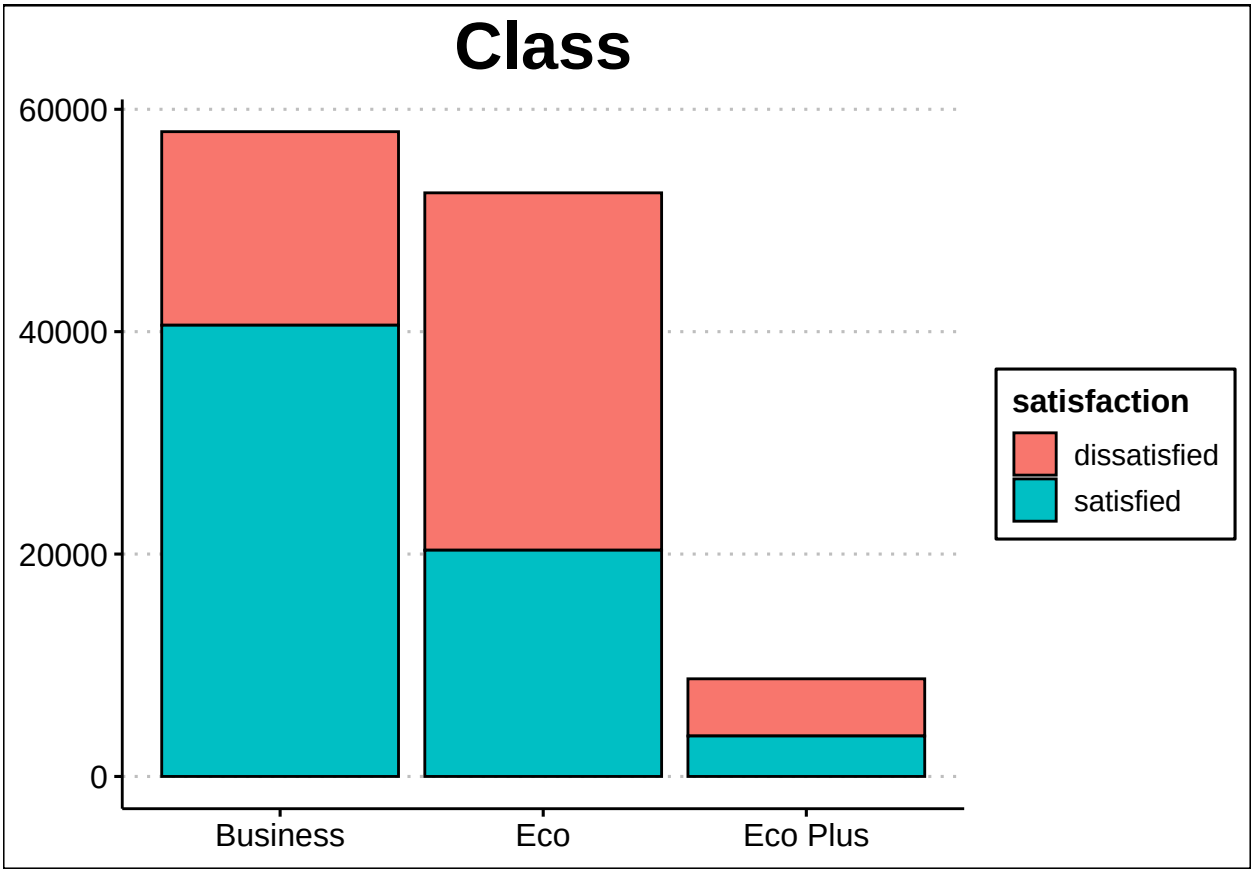
```
ggplot(others, aes(Type.of.Travel)) + geom_bar(aes(fill=satisfaction), color='black') + tema + labs (ti
    theme(plot.title = element_text(hjust=.5,size=25))
```



Inference: We can see that there are more business travelers than personal travels, which might explain why there are more loyal customers than disloyal customers. From the graph, it is hard to see that either business or personal travel has more satisfied passengers or not.

2.5.3 Class

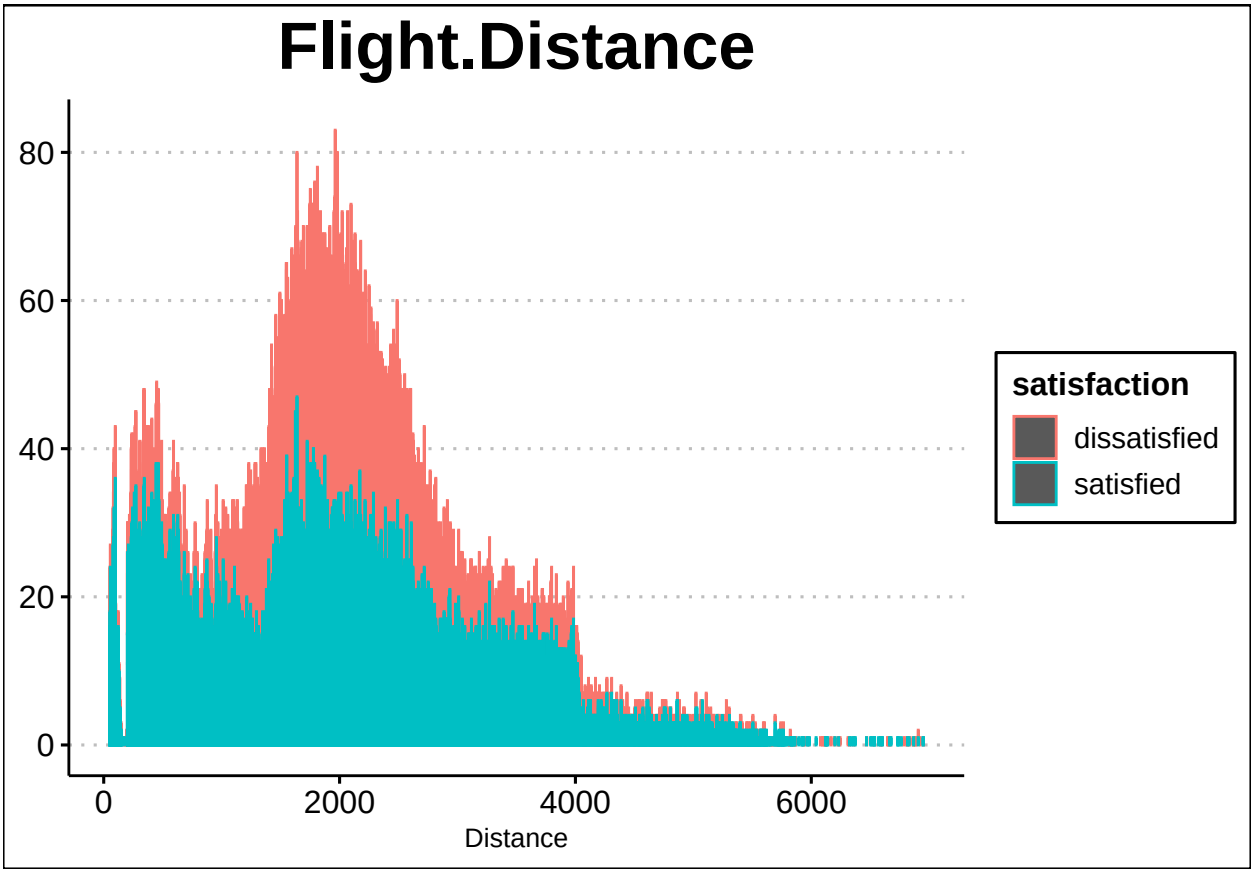
```
ggplot(others, aes(Class)) + geom_bar(aes(fill=satisfaction), color='black') + tema + labs (title='Class
    theme(plot.title = element_text(hjust=.5,size=25))
```



Inference: We have more travelers from business class [indicated by the previous graph where there are more business travelers]. we can see that in general, passengers from business class are more satisfied than Eco and Eco plus.

2.5.4 Flight.Distance

```
ggplot(others, aes(Flight.Distance, color=satisfaction)) + geom_bar() + tema + labs(title='Flight.Distar
```



Inference: We can see that there are more passengers who travel around 2000 miles distance.

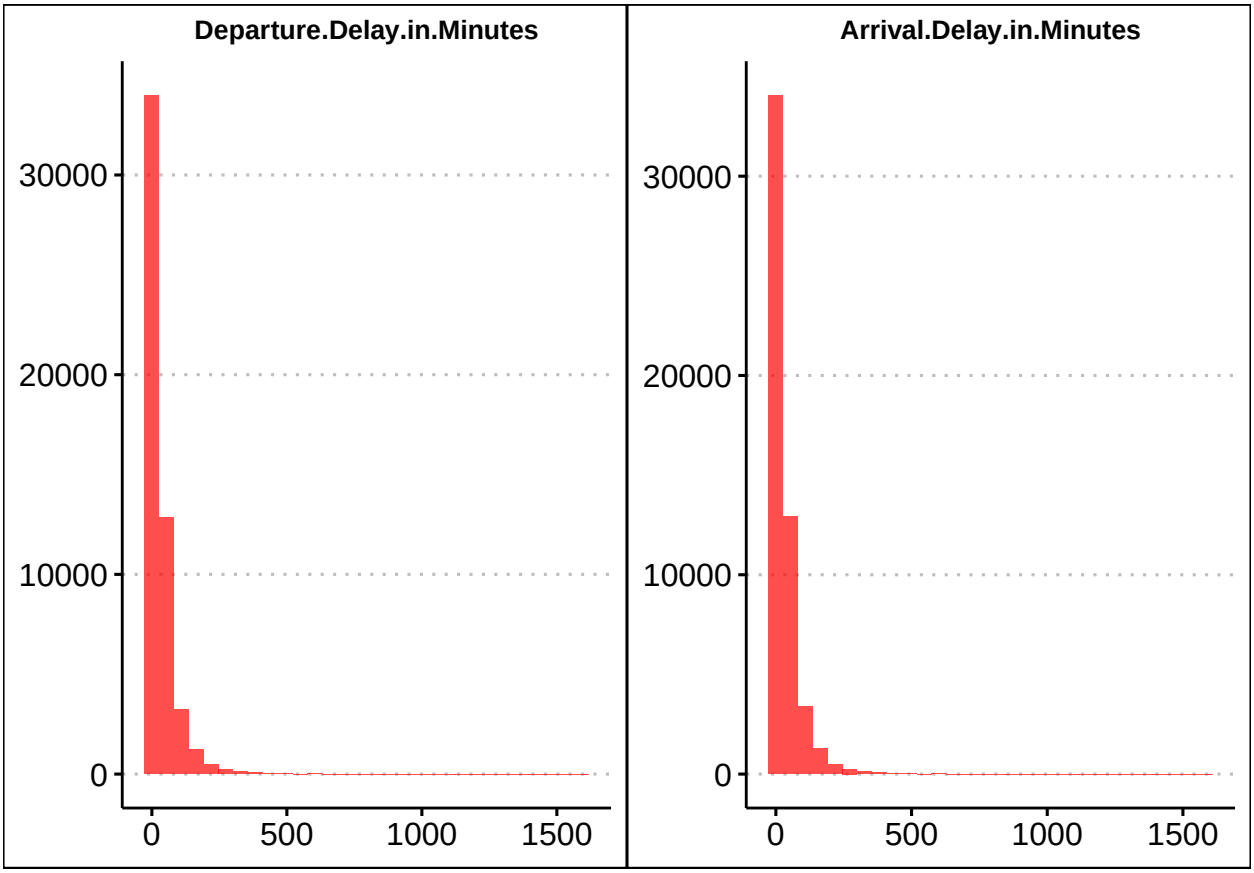
2.5.5

```
names(others)
```

```
## [1] "satisfaction"                "Gender"
## [3] "Customer.Type"               "Age"
## [5] "Type.of.Travel"              "Class"
## [7] "Flight.Distance"             "Departure.Delay.in.Minutes"
## [9] "Arrival.Delay.in.Minutes"
```

```
Departure.Delay.in.Minutes.hist <- df %>% filter(Departure.Delay.in.Minutes > 0) %>% ggplot(mapping = ae
    geom_histogram(fill = "red", size = 1.6, alpha = .7) +
    labs (title='Departure.Delay.in.Minutes',y=NULL,x=NULL) +
    tema

Arrival.Delay.in.Minutes.hist <- df %>% filter(Arrival.Delay.in.Minutes > 0) %>% ggplot(mapping = aes(x
    geom_histogram(fill = "red", size = 1.6, alpha = .7) +
    labs (title='Arrival.Delay.in.Minutes',y=NULL,x=NULL) +
    tema

plot_grid(Departure.Delay.in.Minutes.hist,Arrival.Delay.in.Minutes.hist)
```

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



Inference: The similarities between these two graphs are evident. This makes sense since if an airline's departure is delayed, it is quite likely that its arrival will also be delayed.

2.6 additional data exploration - kmeans

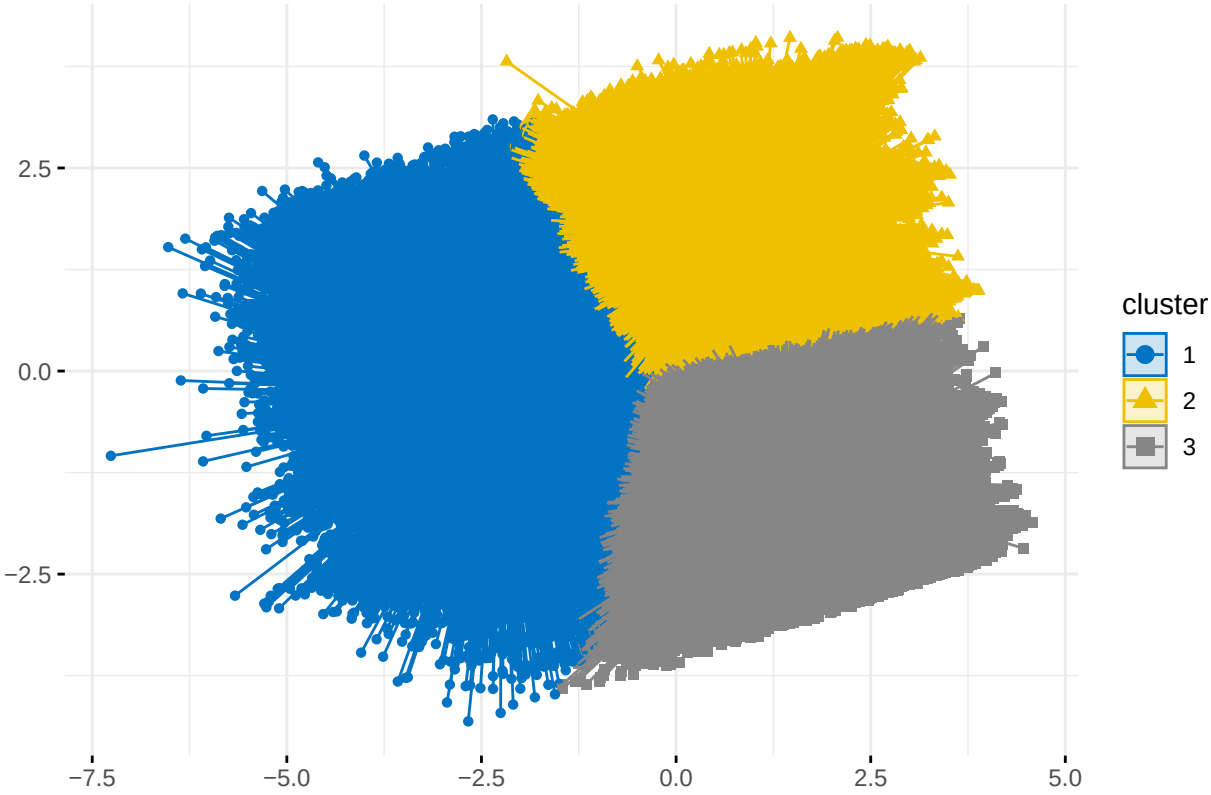
2.6.1 K-mean visualization

```
### k-means
library(factoextra)
```

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

```
set.seed(42)
airline_kmean <- model.matrix(satisfaction ~.  ,data=df)[,-1]
airline_kmean <- scale(airline_kmean)

ThrCenters <- kmeans(airline_kmean,3,nstart=25)
fviz_cluster(object=ThrCenters,airline_kmean,
             ellipse.type = "euclid",star.plot=T,repel=T,
             geom = "point",palette='jco',main="",
```

```
            ggtheme=theme_minimal())+
  theme(axis.title = element_blank())
```




```
#divide into 3 clusters
ThrCenters$centers[1,]
```

```
##                          GenderMale       Customer.TypeLoyal Customer
##                           0.18126723                       -0.28814218
##                                  Age     Type.of.TravelPersonal Travel
##                          -0.14853633                        0.01300681
##                            ClassEco                     ClassEco Plus
##                           0.26176298                        0.06113482
##                       Flight.Distance                      Seat.comfort
##                           0.05791269                       -0.38789420
## Departure.Arrival.time.convenient                     Food.and.drink
##                          -0.04891749                       -0.14914142
##                        Gate.location             Inflight.wifi.service
##                          -0.01576213                       -0.69550763
##              Inflight.entertainment                    Online.support
##                          -0.66348085                       -0.83937002
##              Ease.of.Online.booking                  On.board.service
##                          -1.00983981                       -0.60706284
##                     Leg.room.service                  Baggage.handling
##                          -0.50859836                       -0.53390801
##                      Checkin.service                       Cleanliness
##                          -0.41042710                       -0.54217468
##                     Online.boarding         Departure.Delay.in.Minutes
```

```
##                       -0.84293362                       0.11790534
##          Arrival.Delay.in.Minutes
##                       0.12394095
```

```
ThrCenters$centers[2,]
```

```
##                         GenderMale       Customer.TypeLoyal Customer
##                         -0.03568036                        0.12750734
##                                 Age     Type.of.TravelPersonal Travel
##                          0.11230770                       -0.16875344
##                            ClassEco                      ClassEco Plus
##                         -0.28900735                       -0.07424884
##                     Flight.Distance                      Seat.comfort
##                          0.02734304                       -0.56969960
## Departure.Arrival.time.convenient                     Food.and.drink
##                         -0.74913760                       -0.73039167
##                       Gate.location             Inflight.wifi.service
##                         -0.75814620                        0.42459579
##             Inflight.entertainment                     Online.support
##                          0.22548714                        0.52467448
##             Ease.of.Online.booking                   On.board.service
##                          0.62626445                        0.37437989
##                    Leg.room.service                   Baggage.handling
##                          0.31856723                        0.32942684
##                     Checkin.service                       Cleanliness
##                          0.25228399                        0.33182649
##                     Online.boarding        Departure.Delay.in.Minutes
##                          0.52318652                       -0.05456096
##          Arrival.Delay.in.Minutes
##                         -0.05885271
```

```
ThrCenters$centers[3,]
```

```
##                         GenderMale       Customer.TypeLoyal Customer
##                        -0.159837079                        0.187477310
##                                 Age     Type.of.TravelPersonal Travel
##                          0.052822998                       0.144886397
##                            ClassEco                      ClassEco Plus
##                         -0.007391584                       0.004625158
##                     Flight.Distance                      Seat.comfort
##                         -0.087515152                       0.949845161
## Departure.Arrival.time.convenient                     Food.and.drink
##                          0.756985964                       0.846284869
##                       Gate.location             Inflight.wifi.service
##                          0.730085767                       0.342618550
##             Inflight.entertainment                     Online.support
##                          0.495769692                       0.401959745
##             Ease.of.Online.booking                   On.board.service
##                          0.488268228                       0.295494657
##                    Leg.room.service                   Baggage.handling
##                          0.242945605                       0.259733275
##                     Checkin.service                       Cleanliness
##                          0.200560182                       0.266295896
```

```
##                   Online.boarding        Departure.Delay.in.Minutes
##                       0.407161853                      -0.074469333
##          Arrival.Delay.in.Minutes
##                      -0.076871409
```

###2.6.2 Inference

```
### Sizes of clusters
size <- ThrCenters$size
size
```

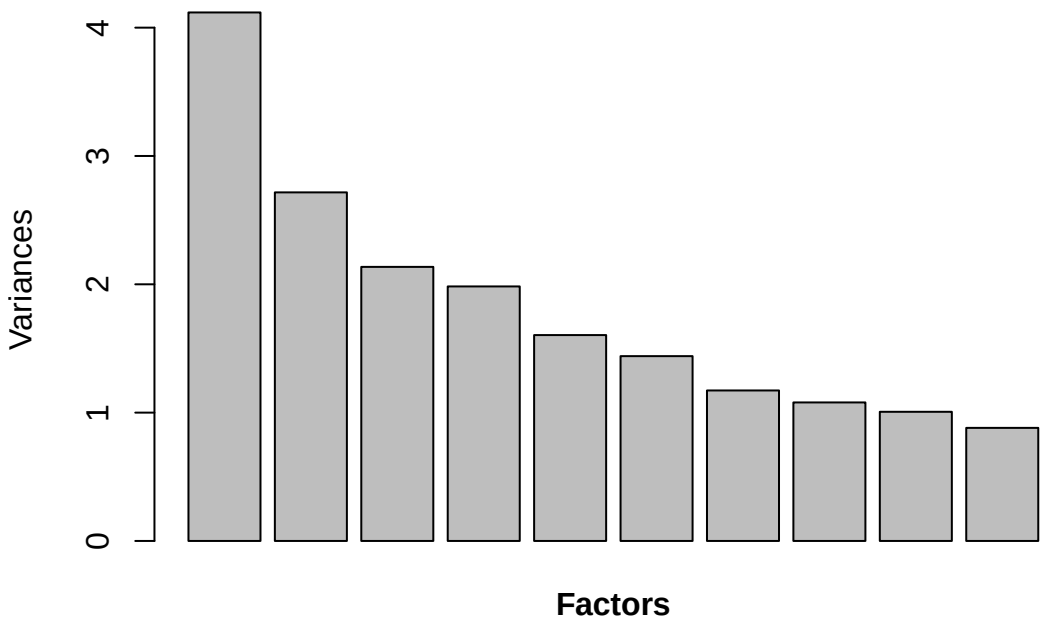
```
## [1] 42304 37302 39649
```

```
a<-aggregate( df$satisfaction=='dissatisfied' ~ ThrCenters$cluster, FUN=mean)
b<-aggregate( df$Seat.comfort ~ ThrCenters$cluster, FUN = mean )
c<-aggregate( df$Departure.Arrival.time.convenient ~ ThrCenters$cluster, FUN = mean )
d<-aggregate( df$Food.and.drink ~ ThrCenters$cluster, FUN = mean )
e<-aggregate( df$Gate.location ~ ThrCenters$cluster, FUN = mean )
f<-aggregate( df$Inflight.wifi.service ~ ThrCenters$cluster, FUN = mean )
g<-aggregate( df$Inflight.entertainment ~ ThrCenters$cluster, FUN = mean )
h<-aggregate( df$Online.support ~ ThrCenters$cluster, FUN = mean )
i<-aggregate( df$Ease.of.Online.booking ~ ThrCenters$cluster, FUN = mean )
j<-aggregate( df$On.board.service ~ ThrCenters$cluster, FUN = mean )
k<-aggregate( df$Leg.room.service~ ThrCenters$cluster, FUN = mean )
l<-aggregate( df$Baggage.handling~ ThrCenters$cluster, FUN = mean )
m<-aggregate( df$Checkin.service~ ThrCenters$cluster, FUN = mean )
n<-aggregate( df$Cleanliness~ ThrCenters$cluster, FUN = mean )
o<-aggregate( df$Online.boarding~ ThrCenters$cluster, FUN = mean )


kmeans_output <- cbind(a,size,b[,2],c[,2],d[,2],e[,2],f[,2],g[,2],h[,2],i[,2],j[,2],k[,2],l[,2],m[,2],n
colnames(kmeans_output) <-c('cluster','dissatisfied','size','Seat comfort', 'Departure/Arrival time con
view(kmeans_output)
```

2.7 additional data exploration - PCA

###2.7.1 PCA visualization

```
### Lets compute the (Full) PCA
airline_x <- model.matrix(satisfaction ~.,data=df)[,-1]
airline_pca <- prcomp(airline_x, scale=TRUE)
### Lets plot the variance that each component explains
par(mar=c(4,4,4,4)+0.3)
plot(airline_pca,main="PCA: Variance Explained by Factors")
mtext(side=1, "Factors",  line=1, font=2)
```



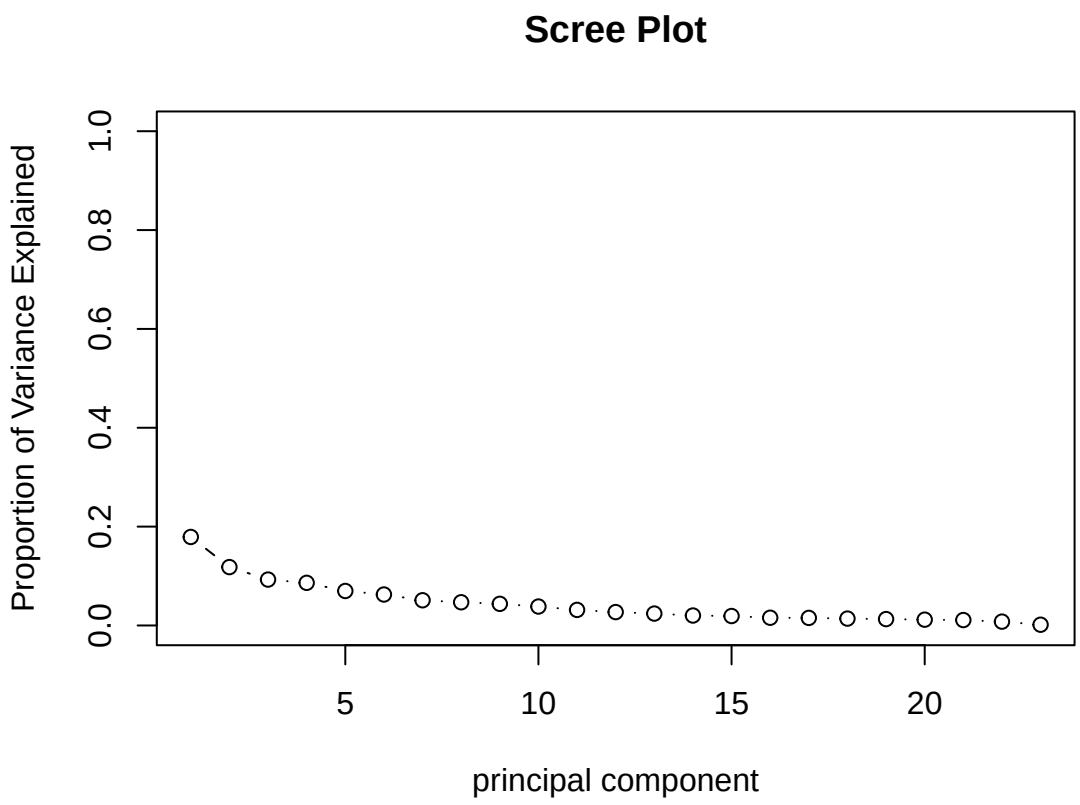

```
# Compute variance
airline_pca.var <- airline_pca$sdev ^ 2
airline_pca.var
```

```
##  [1] 4.11856176 2.71635777 2.13581355 1.98339766 1.60442931 1.44025300
##  [7] 1.17265878 1.07940841 1.00645842 0.88110408 0.72405656 0.62195184
## [13] 0.55299001 0.46302245 0.43829993 0.35899251 0.35191748 0.32013549
## [19] 0.29599780 0.26952081 0.25175829 0.17799828 0.03491581
```

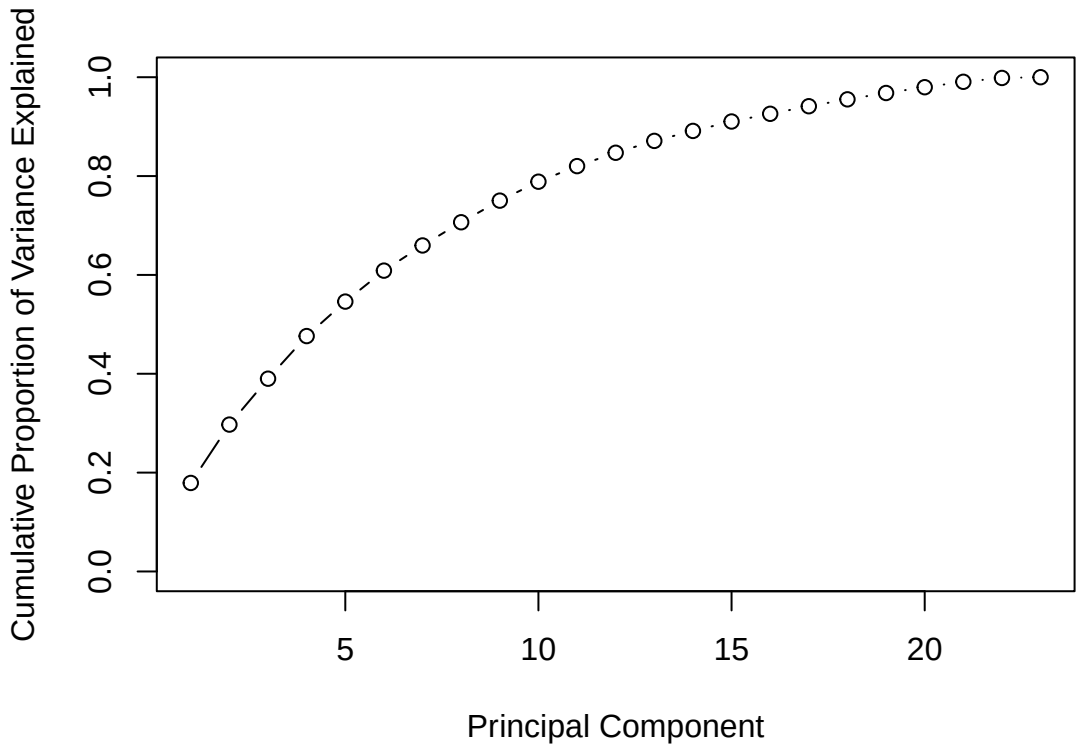
```
# Proportion of variance for a scree plot
propve <- airline_pca.var / sum(airline_pca.var)
propve
```

```
##  [1] 0.179067903 0.118102512 0.092861459 0.086234681 0.069757796 0.062619695
##  [7] 0.050985164 0.046930800 0.043759062 0.038308873 0.031480720 0.027041384
## [13] 0.024043044 0.020131411 0.019056519 0.015608370 0.015300760 0.013918934
## [19] 0.012869470 0.011718296 0.010946013 0.007739056 0.001518079
```

```
# Plot variance explained for each principal component
plot(propve, xlab = "principal component",
             ylab = "Proportion of Variance Explained",
             ylim = c(0, 1), type = "b",
             main = "Scree Plot")
```




```r
# Plot the cumulative proportion of variance explained
plot(cumsum(propve),
    xlab = "Principal Component",
    ylab = "Cumulative Proportion of Variance Explained",
    ylim = c(0, 1), type = "b")
```



2.7.2 PCA inference

```
####Loading 1
loadings <- airline_pca$rotation[,1:3]
v<-loadings[order(abs(loadings[,1]), decreasing=TRUE)[1:ncol(airline_x)],1]
loadingfit <- lapply(1:ncol(airline_x), function(k) ( t(v[1:k])%*%v[1:k] - 3/4 )^2)
v[1:which.min(loadingfit)]
```

```
## Ease.of.Online.booking         Online.support        Online.boarding
##              0.4085906              0.3364708             0.3295607
## Inflight.entertainment       On.board.service            Cleanliness
##              0.3022009              0.2895912             0.2701654
##  Inflight.wifi.service       Baggage.handling
##              0.2693186              0.2674208
```

```
#### Looking at which are large positive and large negative
#### First factor is Online service

#### Loading 2
v<-loadings[order(abs(loadings[,2]), decreasing=TRUE)[1:ncol(airline_x)],2]
loadingfit <- lapply(1:ncol(airline_x), function(k) ( t(v[1:k])%*%v[1:k] - 3/4 )^2)
v[1:which.min(loadingfit)]
```

```
##                    Food.and.drink Departure.Arrival.time.convenient
##                        -0.5089377                        -0.4892752
##                     Gate.location
##                        -0.4745705
```

```
#### Second factor is food and drink, convenient
```

3. Data Preparation

3.1 Null values

```
na_summary(df) #this function is used to give the null summary of the dataset
```

```
##                              variable missing complete percent_complete
## 1                                 Age       0   119255              100
## 2           Arrival.Delay.in.Minutes       0   119255              100
## 3                   Baggage.handling       0   119255              100
## 4                    Checkin.service       0   119255              100
## 5                              Class       0   119255              100
## 6                        Cleanliness       0   119255              100
## 7                      Customer.Type       0   119255              100
## 8  Departure.Arrival.time.convenient       0   119255              100
## 9         Departure.Delay.in.Minutes       0   119255              100
## 10            Ease.of.Online.booking       0   119255              100
## 11                   Flight.Distance       0   119255              100
## 12                    Food.and.drink       0   119255              100
## 13                     Gate.location       0   119255              100
## 14                            Gender       0   119255              100
## 15            Inflight.entertainment       0   119255              100
## 16             Inflight.wifi.service       0   119255              100
## 17                  Leg.room.service       0   119255              100
## 18                  On.board.service       0   119255              100
## 19                   Online.boarding       0   119255              100
## 20                    Online.support       0   119255              100
## 21                      satisfaction       0   119255              100
## 22                      Seat.comfort       0   119255              100
## 23                     Type.of.Travel       0   119255              100
##    percent_missing
## 1                0
## 2                0
## 3                0
## 4                0
## 5                0
## 6                0
## 7                0
## 8                0
## 9                0
## 10               0
## 11               0
## 12               0
## 13               0
## 14               0
## 15               0
## 16               0
## 17               0
```

```
## 18              0
## 19              0
## 20              0
## 21              0
## 22              0
## 23              0
```

We can see that the null data (Arrival.Delay) 393 is nothing compared to the overall data set size. So, we can safely drop them without worrying too much about its impact on the integrity of the whole data set.

```
df <- df[!is.na(df$Arrival.Delay.in.Minutes),] #drop Nulls
```

3.2 Feature engineering

Question: Because we want to see how the change of satisfaction rate can impact passengers' profit. So, there are two important variables that we need to create and estimate for our study. * The ticket price of each customer * The cost associated with the change of customers' satisfaction level

A. Estimate the tick price for each passenger

```
df <- df %>% mutate(Class_level = if_else(Class == 'Eco',1,if_else(Class=='Business',3,2))) #change cus
df <- df %>% mutate(ticket_price = ((1/5) * df$Flight.Distance * df$Class_level) + 100)
# we estimate an equation to compute the tick price
```

B. Satisfaction and cost model

```
#We dropped all the subjective satisfaction x variables which contain 0 [0 indicates no applicable: use

df <- df[which(rowSums(df[,c('Seat.comfort','Departure.Arrival.time.convenient','Food.and.drink', "Gate

#Merge all satisfactions into one overall satisfaction for simple calculation
df = df %>% mutate(satisfaction.level.overall =0.3 *df$Food.and.drink + 0.2*df$Inflight.entertainment +
#We use the satisfaction moedel and obtained the overall satisfaction level for each customer

#Finally, cost and satisfaction function - average cost of satisfaction improvement on each class passe
cost_function <- function(satisfaction, class){
  cost = 100 * satisfaction
  return(cost)
}
```

4. Modeling

4.1 Data transformation

```
#convert all strings into factors
df <- df %>% select(-c('Class_level','Seat.comfort','Departure.Arrival.time.convenient','Food.and.drink
df[sapply(df,is.character)] <- lapply(df[sapply(df,is.character)],as.factor)

## train test split
library(caTools)
set.seed(42)
sample <- sample.split(df$satisfaction, SplitRatio = 0.7)
#70% for training and 30% for validation
train <- df %>% filter(sample == T)
test <- df %>% filter(sample == F)
```

4.2 The model that airline businesses can employ to predict passenger satisfaction

```
### Need to estimate satisfaction
### Compare different models
### m.lr : logistic regression
### m.lr.l : logistic regression with lasso
### m.lr.pl : logistic regression with post lasso
### m.lr.tree : classification tree

library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-4
```

```
#### Lets run Lasso
#### First lets set up the data for it
#### the features need to be a matrix ([,-1] removes the first column which is the intercept)

Mx<- model.matrix(satisfaction ~ ., data=train)[,-1]
My<- train$satisfaction == "satisfied"
lasso <- glmnet(Mx,My, family="binomial")
lassoCV <- cv.glmnet(Mx,My, family="binomial")

#for post lasso parameters setup
num.features <- ncol(Mx)
num.n <- nrow(Mx)
num.sat <- sum(My)
w <- (num.sat/num.n)*(1-(num.sat/num.n))
lambda.theory <- sqrt(w*log(num.features/0.05)/num.n)
lassoTheory <- glmnet(Mx,My, family="binomial",lambda = lambda.theory)
summary(lassoTheory)
```

```
##            Length Class     Mode
## a0          1     -none-    numeric
## beta       11     dgCMatrix S4
## df          1     -none-    numeric
## dim         2     -none-    numeric
## lambda      1     -none-    numeric
## dev.ratio   1     -none-    numeric
## nulldev     1     -none-    numeric
## npasses     1     -none-    numeric
## jerr        1     -none-    numeric
## offset      1     -none-    logical
## classnames  2     -none-    character
## call        5     -none-    call
## nobs        1     -none-    numeric
```

```
support(lassoTheory$beta)
```

```
## [1]  1  2  4  5  6  7  9 10 11
```

```
features.min <- support(lasso$beta[,which.min(lassoCV$cvm)])
features.min <- support(lassoTheory$beta)
  length(features.min)
```

```
## [1] 9
```

```
data.min <- data.frame(Mx[,features.min],My)
```

4.3 Performance metrics and k-fold set up

```
### prediction is a probability score
### we convert to 1 or 0 via prediction > threshold
PerformanceMeasure <- function(actual, prediction, threshold=.5) {
  # 1-mean( abs( (prediction>threshold) - actual ) )
  R2(y=actual, pred=prediction, family="binomial")
  # 1-mean( abs( (prediction - actual) ) )
}

#kfold setup
n <- nrow(train)
nfold <- 10
OOS <- data.frame(m.lr=rep(NA,nfold), m.lr.l=rep(NA,nfold), m.lr.pl=rep(NA,nfold), m.tree=rep(NA,nfold)
#names(OOS)<- c("Logistic Regression", "Lasso on LR ", "Post Lasso on LR ", "Classification Tree", "Ran
foldid <- rep(1:nfold,each=ceiling(n/nfold))[sample(1:n)]
```

4.4 Model running

```
#kfold
library(tree)
library(partykit)
```

```
## Loading required package: grid

## Loading required package: libcoin

## Loading required package: mvtnorm
```

```
library(glmnet)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked _by_ '.GlobalEnv':
##
##     R2

## The following object is masked from 'package:purrr':
##
##     lift
```

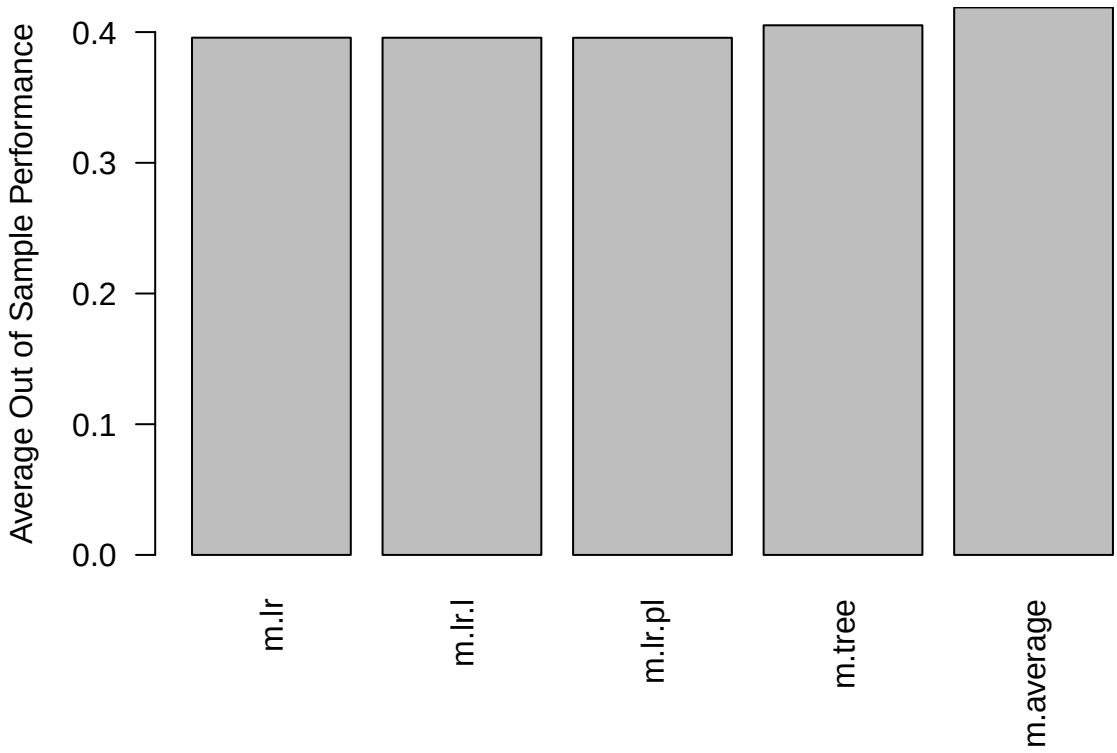
```
for(k in 1:nfold){
  train_k <- which(foldid!=k) # train on all but fold `k'

  ### Logistic regression
  m.lr <-glm(satisfaction~., data=train,subset=train_k,family="binomial")
  pred.lr <- predict(m.lr, newdata=train[-train_k,], type="response")
  OOS$m.lr[k] <- PerformanceMeasure(actual=My[-train_k], pred=pred.lr)

  ### the Post Lasso Estimates
  m.lr.pl <- glm(My~., data=data.min, subset=train_k, family="binomial")
  pred.lr.pl <- predict(m.lr.pl, newdata=data.min[-train_k,], type="response")
  OOS$m.lr.pl[k] <- PerformanceMeasure(actual=My[-train_k], prediction=pred.lr.pl)

  ### the Lasso estimates
  m.lr.l  <- glmnet(Mx[train_k,],My[train_k], family="binomial",lambda = lassoCV$lambda.min)
  pred.lr.l <- predict(m.lr.l, newx=Mx[-train_k,], type="response")
  OOS$m.lr.l[k] <- PerformanceMeasure(actual=My[-train_k], prediction=pred.lr.l)

  ### the classification tree
  m.tree <- tree(satisfaction~ ., data=train, subset=train_k)
  pred.tree <- predict(m.tree, newdata=train[-train_k,], type="vector")
  pred.tree <- pred.tree[,2]
  OOS$m.tree[k] <- PerformanceMeasure(actual=My[-train_k], prediction=pred.tree)

  ###Average
  pred.m.average <- rowMeans(cbind(pred.tree, pred.lr.l, pred.lr.pl, pred.lr))
  OOS$m.average[k] <- PerformanceMeasure(actual=My[-train_k], prediction=pred.m.average)

  print(paste("Iteration",k,"of",nfold,"completed"))

}
```

```
## [1] "Iteration 1 of 10 completed"
## [1] "Iteration 2 of 10 completed"
## [1] "Iteration 3 of 10 completed"
## [1] "Iteration 4 of 10 completed"
## [1] "Iteration 5 of 10 completed"
## [1] "Iteration 6 of 10 completed"
## [1] "Iteration 7 of 10 completed"
## [1] "Iteration 8 of 10 completed"
## [1] "Iteration 9 of 10 completed"
## [1] "Iteration 10 of 10 completed"
```

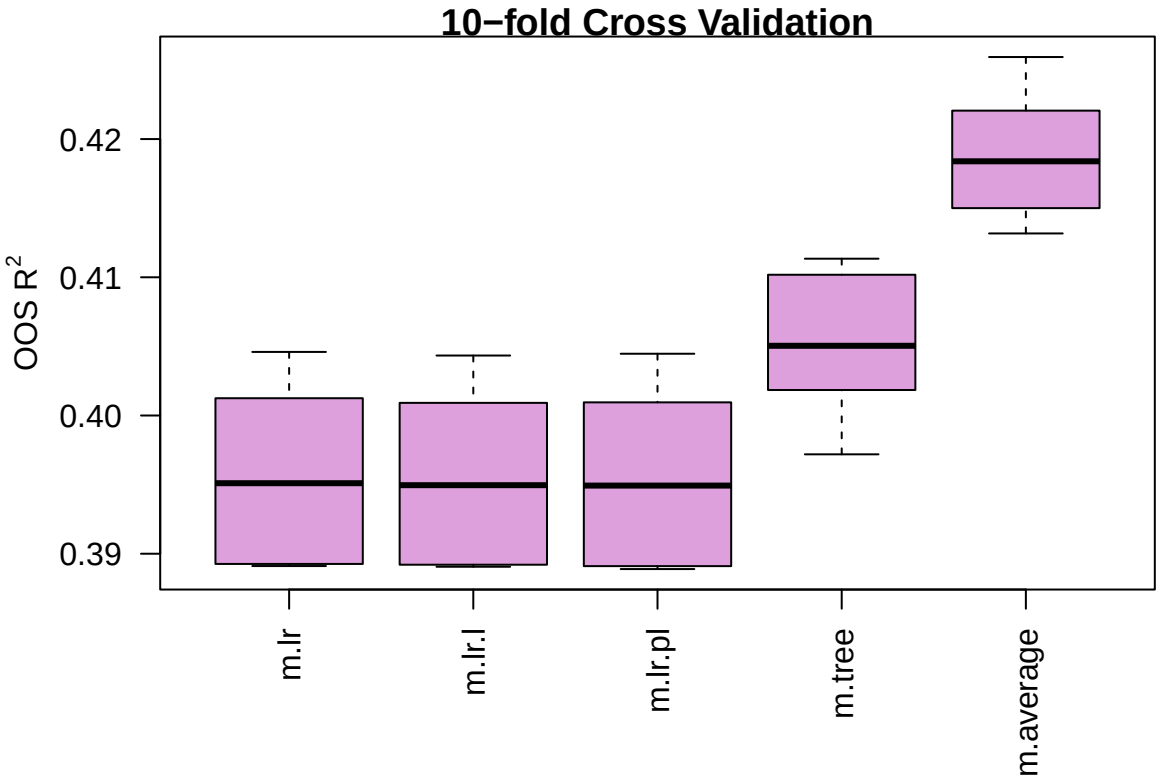
5. Evaluation

5.1 OOS performance and model selection

```
par(mar=c(7,5,.5,1)+0.3)
bar.plot <- barplot(colMeans(OOS), las=2,xpd=FALSE , xlab="",  ylab = bquote( "Average Out of Sample Pe
```



```
box.plot <- boxplot(OOS, col="plum", las = 2, ylab=expression(paste("OOS ",R^2)), xlab="", main="10-fold
```




```
round(colMeans(OOS),4)
```

```
##      m.lr    m.lr.l   m.lr.pl    m.tree m.average
##    0.3958    0.3957    0.3957    0.4052    0.4189
```

```
### We use the logistic regression Estimates to predict models as it has the highest OOS R^2
### and we build the method with the whole training set
m.lr <- glm(satisfaction~., data=train, family="binomial")
pred.lr <- predict(m.lr, newdata=test, type="response")
```

5.2 Confusion matrix

```
#confusion matrix
### We can make predictions using the rule
### if hat prob >= threshold, we set hat Y= 1
### otherwise we set hat Y= 0
### threshold = 0.5 [We will use threshold 0.5 as our prediction benchmark]
My <- test$satisfaction == 'satisfied'
PL.performance <- FPR_TPR(pred.lr>=0.5 , My)
PL.performance
```

```
##         FPR       TPR      ACC    TP   FP   FN    TN
## 1 0.2450933 0.8363017 0.798977 16200 4021 3171 12385
```

```
confusion.matrix <- c( sum(pred.lr>=0.5) *PL.performance$TP,  sum(pred.lr>=0.5) * PL.performance$FP,  su

confusion.matrix <- c( sum( (pred.lr>=0.5) * My ),  sum( (pred.lr>=0.5) * !My ) , sum( (pred.lr<0.5) * !
confusion.matrix
```

```
## [1] 16200  4021  3171 12385
```

5.3 Cost benefit

```
#cost benefit matrix
cost.benefit.matrix <- c( 150,-100,130,-80 )
#we estimate the cost benefit matrix based on the mean of tick price and our intuition
```

5.4 Expected profit

```
### Expected profit
t(cost.benefit.matrix) %*% confusion.matrix
```

```
##         [,1]
## [1,] 1449330
```

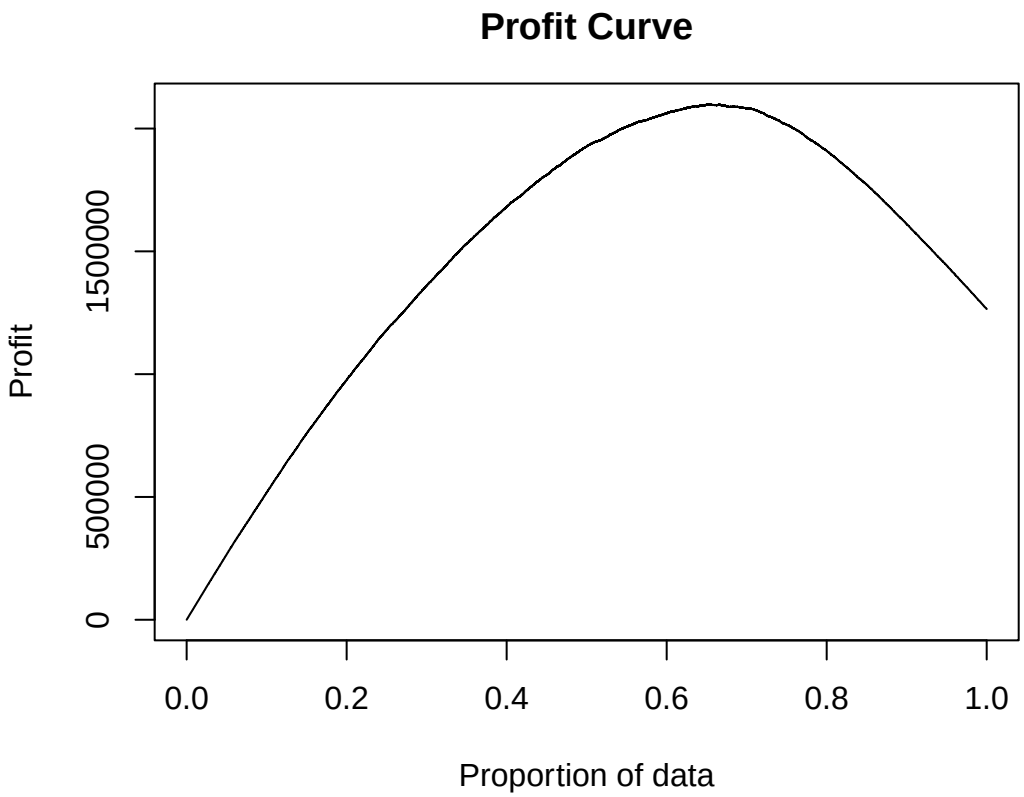
```
### Baseline of majority rule (nobody unsatisfied prediction)
cost.benefit.matrix %*% c( sum(My), sum(!My),0, 0 )
```

```
##         [,1]
## [1,] 1265050
```

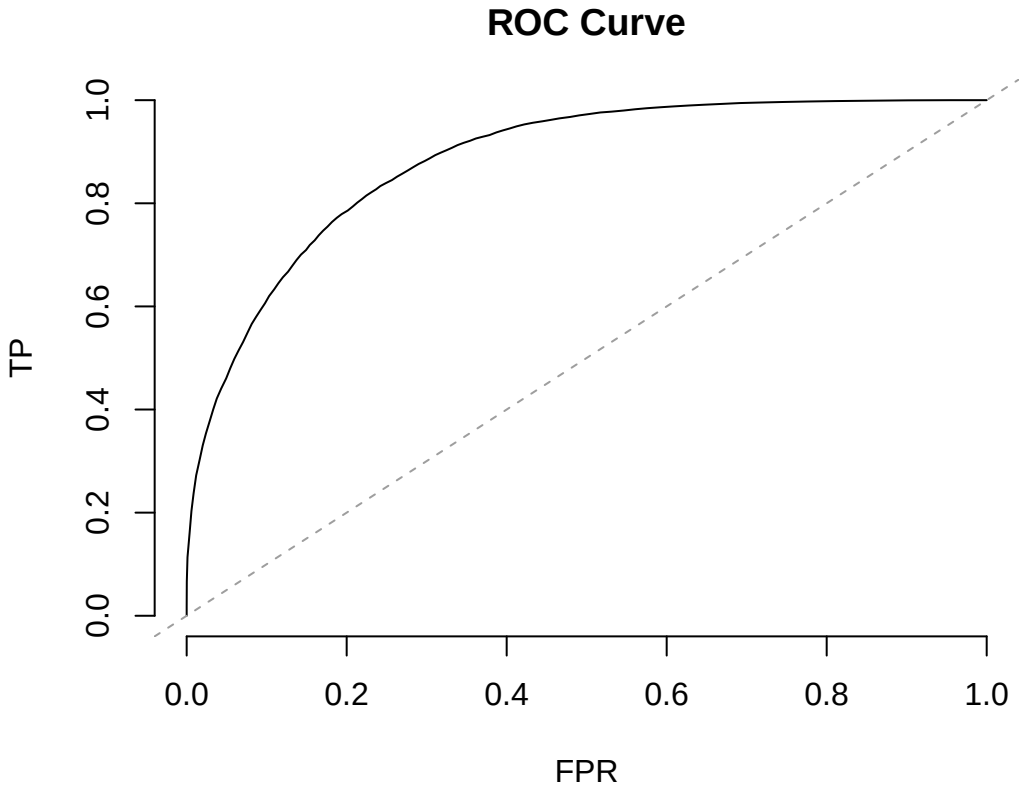
Inference: * expected profit of logistic regression: 187663 * expected profit of majority rule: 126505

5.5 Curves

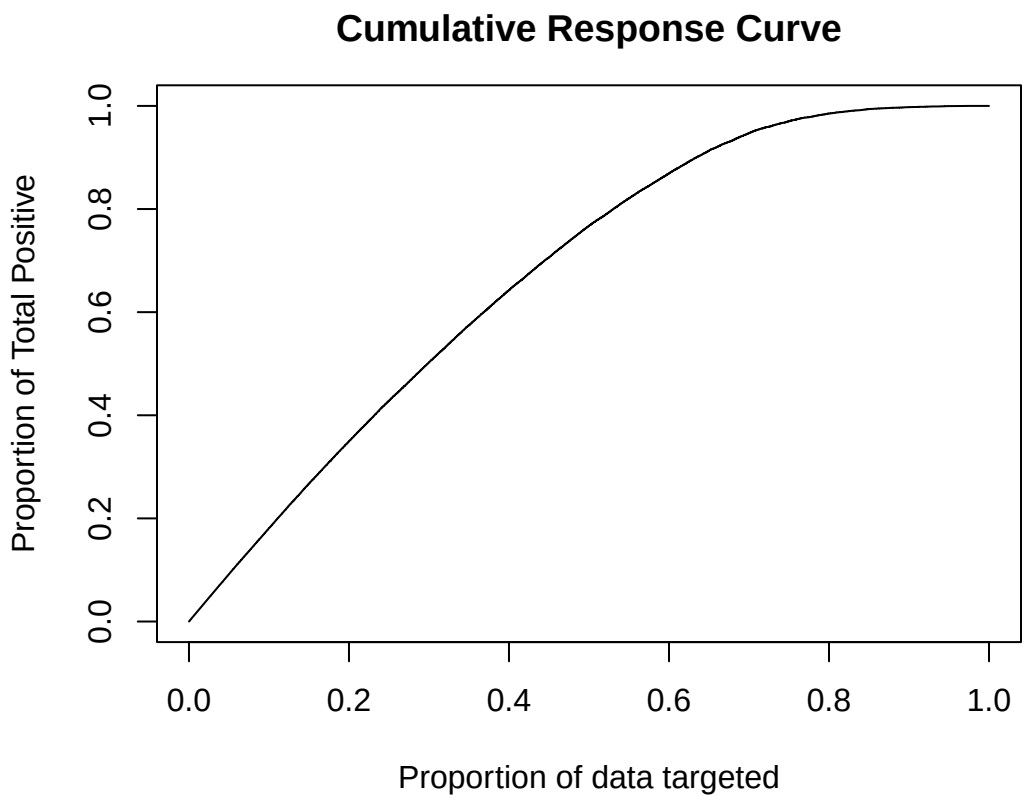
```
### for logistic predictions, we will use functions in PerformanceCurves.R files (load earlier)
par(mar=c(5,5,3,5))
profit <- profitcurve(p=pred.lr,y=My,cost.benefit.m=cost.benefit.matrix)
```



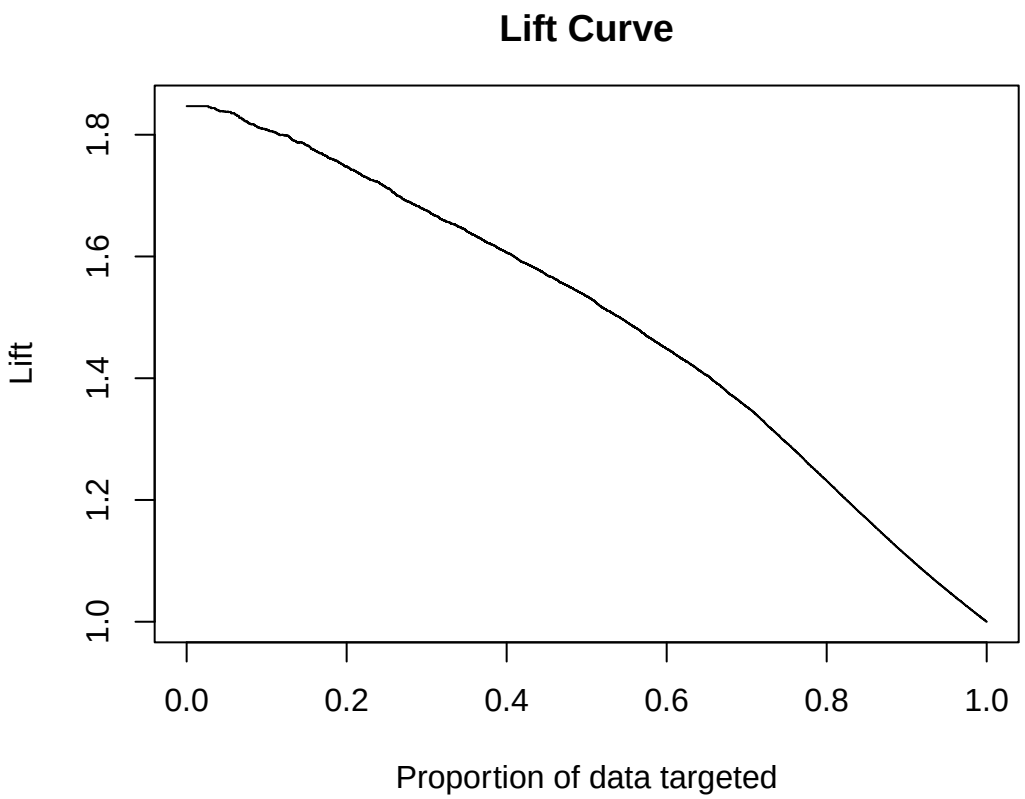

```
roccurve <-  roc(p=pred.lr, y=My, bty="n")
```



```
cumulative <- cumulativecurve(p=pred.lr,y=My)
```




```
lift <- liftcurve(p=pred.lr,y=My)
```

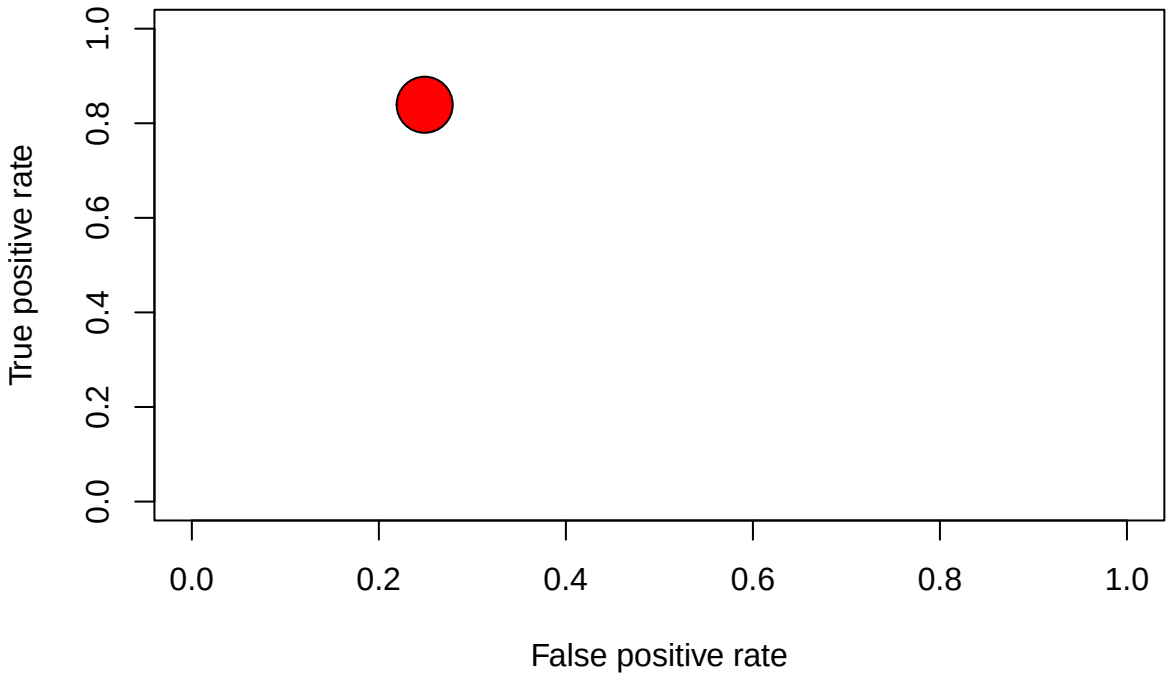


Inference: *

`Profit Curve`: Order the customer based on their profit, we should target around 60% of customers to make the maximized amount of profit.

ROC in details [0.5 is the best]

```
##ROC Curve in details##
#### This is the code for the "red dot plots" ####

index <- c(50)
radius <- 0.03 *rep(1,length(index))
color <- c("red")
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=0.5 , My)
```

```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

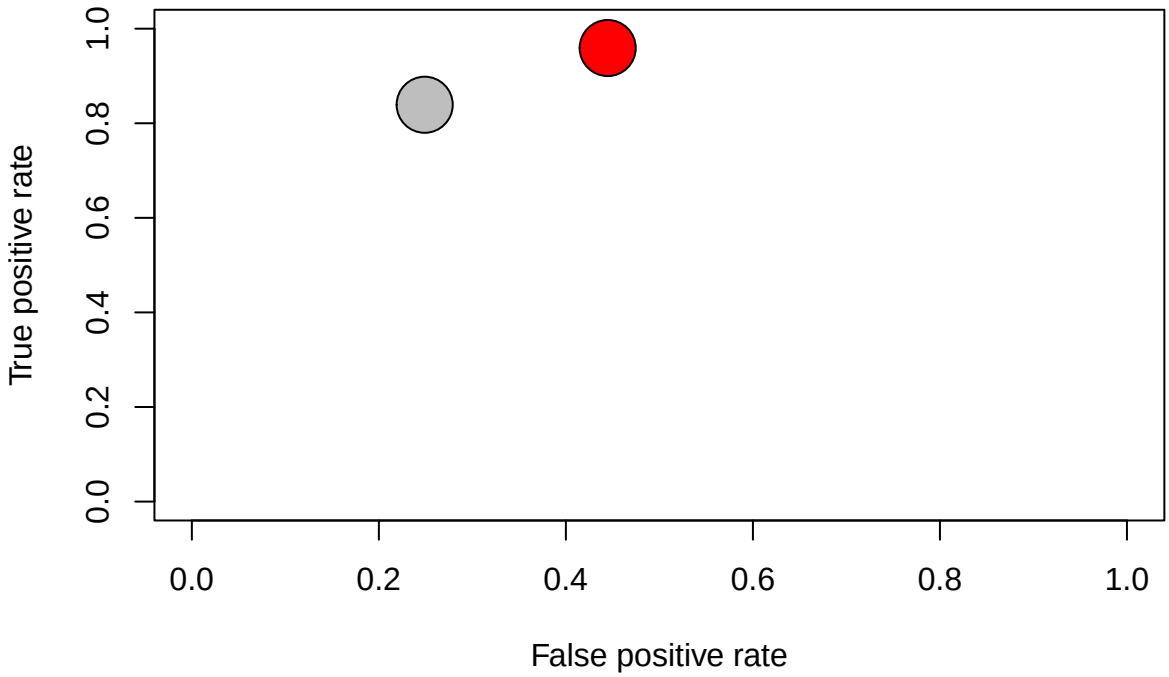
```
## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
##         FPR       TPR       ACC    TP   FP   FN   TN
## 1 0.5137145 0.6021888 0.5490399 11665 8428 7706 7978
```

```
index <- c(25,50)
radius <- 0.03 *rep(1,length(index))
color <- c("red","grey")
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=0.25 , My)
```

```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

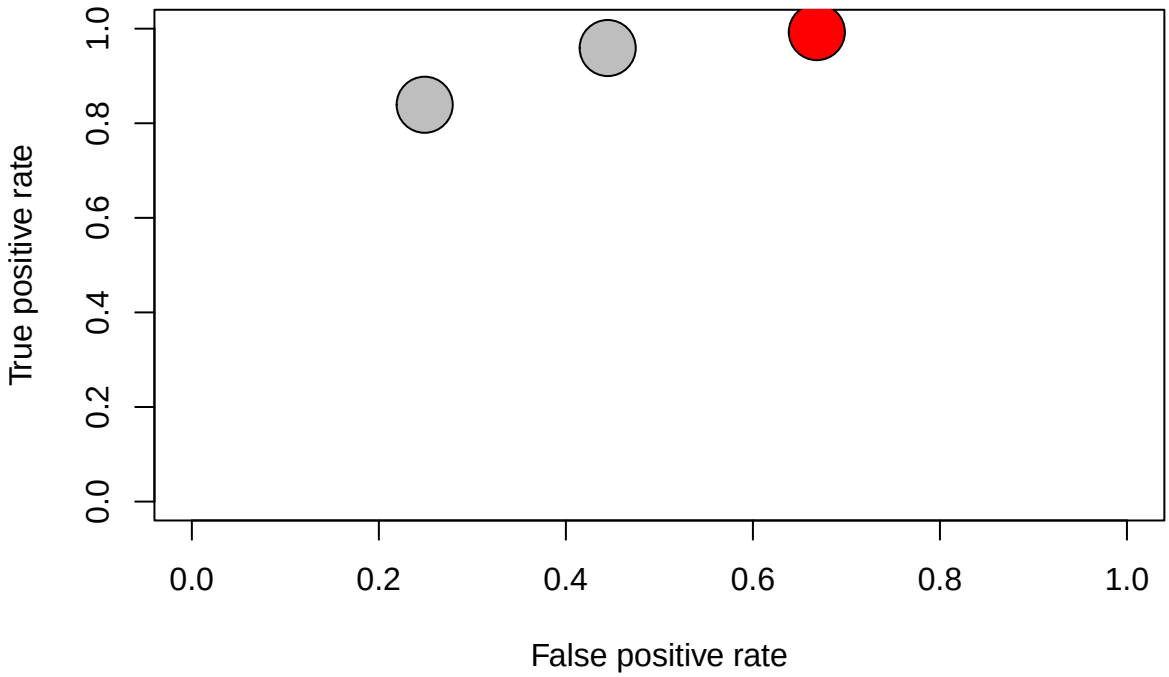
```
## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
##        FPR       TPR      ACC    TP    FP   FN   TN
## 1 0.665732 0.7466316 0.557537 14463 10922 4908 5484
```

```
index <- c(10,25,50)
color <- c("red","grey","grey")
radius <- 0.03 *rep(1,length(index))
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=0.1 , My)
```

```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

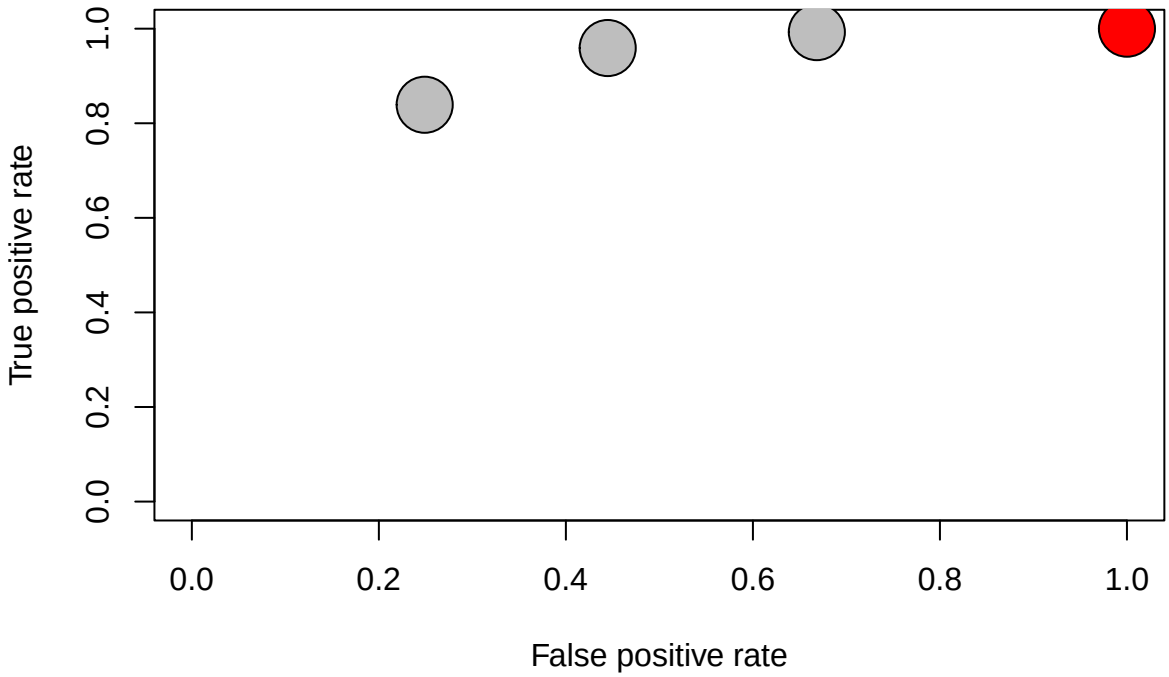
```
## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
##         FPR       TPR       ACC    TP    FP   FN   TN
## 1 0.7962331 0.8552992 0.5565307 16568 13063 2803 3343
```

```
index <- c(1, 10,25,50)
color <- c("red","grey","grey","grey")
radius <- 0.03 *rep(1,length(index))
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=0 , My)
```

```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

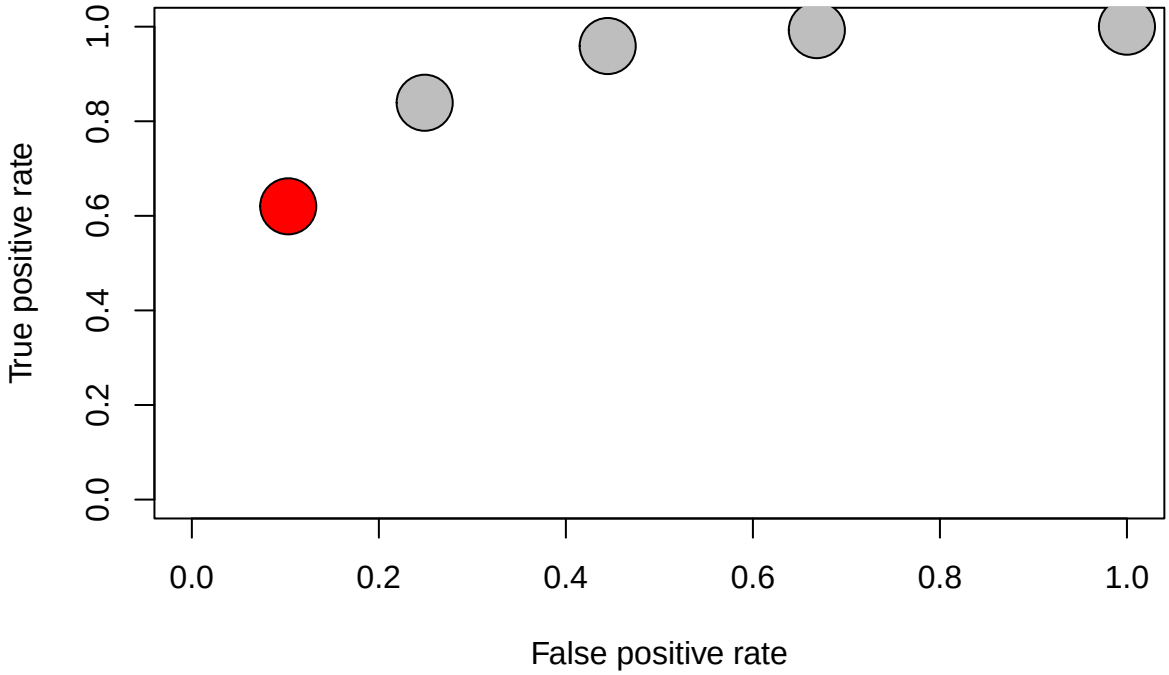
```
## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
##   FPR TPR       ACC    TP    FP FN TN
## 1   1   1 0.5414372 19371 16406  0  0
```

```
index <- c(75, 1, 10,25,50)
color <- c("red","grey","grey","grey","grey")
radius <- 0.03 *rep(1,length(index))
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=0.75 , My)
```

```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

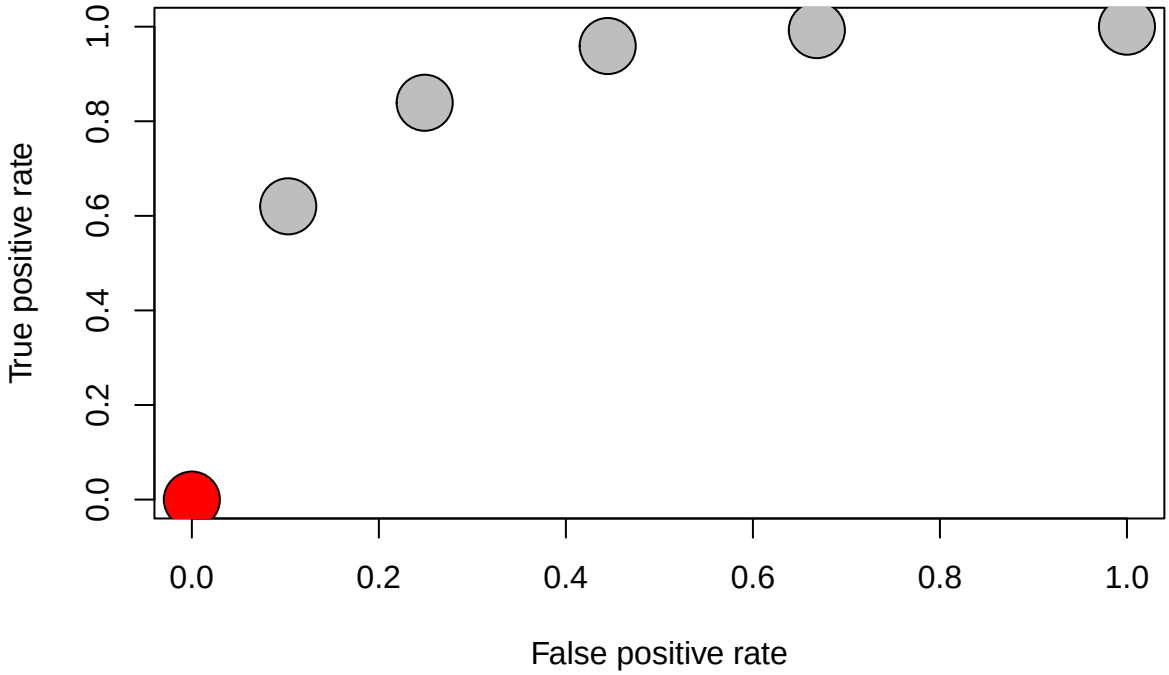
```
## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length
```

```
## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length
```

```
##         FPR       TPR       ACC   TP   FP    FN    TN
## 1 0.3282945 0.4135563 0.5319339 8011 5386 11360 11020
```

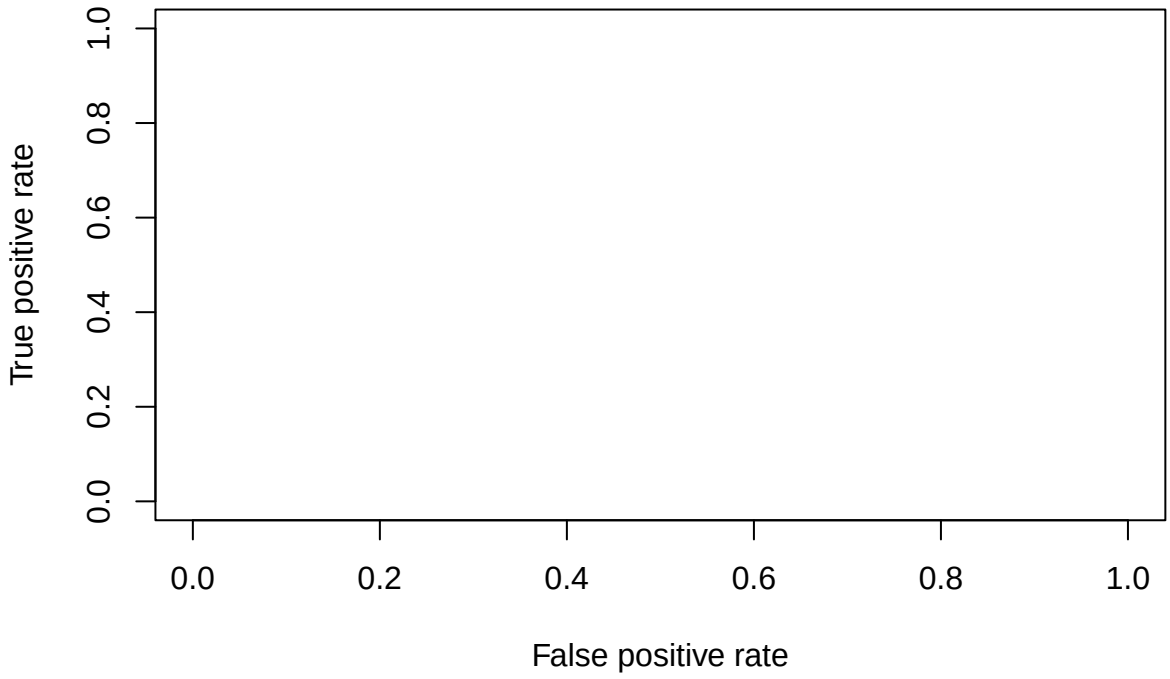
```
index <- c(100, 75, 1, 10,25,50)
color <- c("red","grey","grey","grey","grey","grey")
radius <- 0.03 *rep(1,length(index))
symbols(roccurve[index ,], circles=radius, inches = FALSE,ylim=c(0,1), xlim=c(0,1), ylab="True positive
```




```
FPR_TPR(pred.lr.pl>=1 , My)
```

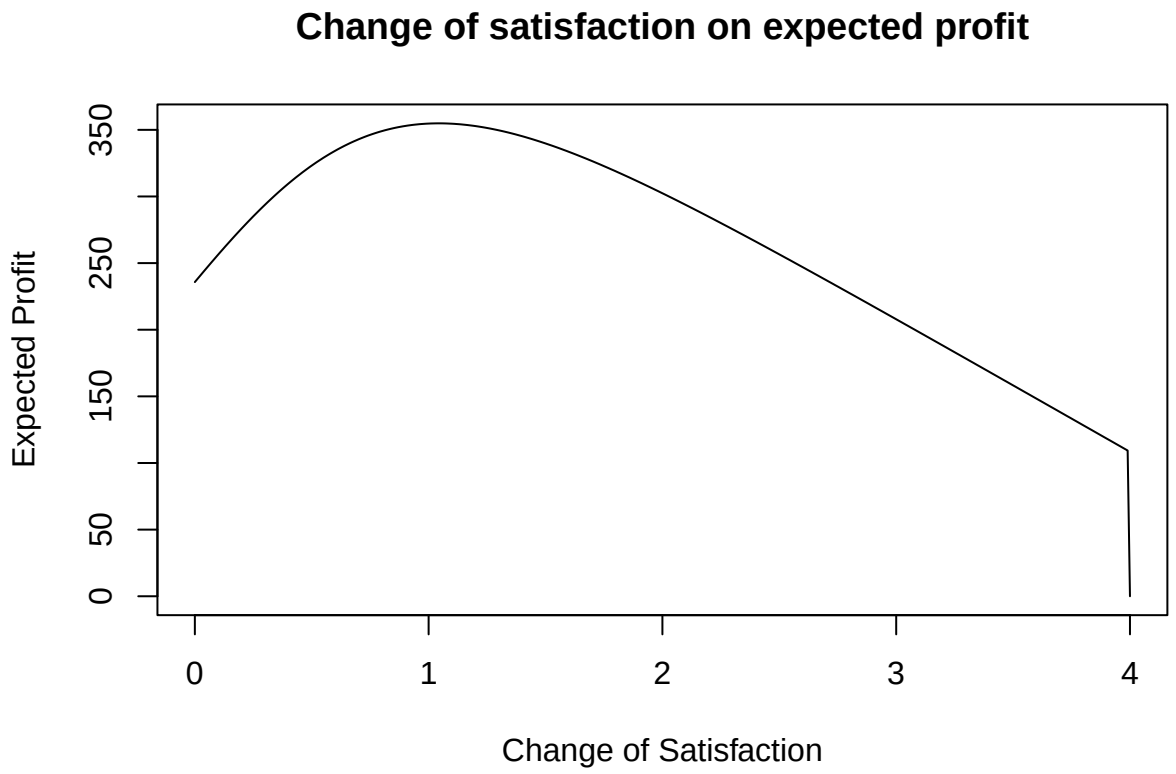
```
## Warning in (prediction) * (actual): longer object length is not a multiple of
## shorter object length

## Warning in (prediction) * (!actual): longer object length is not a multiple of
## shorter object length

## Warning in (!prediction) * (actual): longer object length is not a multiple of
## shorter object length

## Warning in (!prediction) * (!actual): longer object length is not a multiple of
## shorter object length

##   FPR TPR       ACC TP FP    FN    TN
## 1   0   0 0.4585628  0  0 19371 16406
```

```
plot(roccurve[50,],  ylim=c(0,1), xlim=c(0,1), ylab="True positive rate", xlab="False positive rate",ty
```

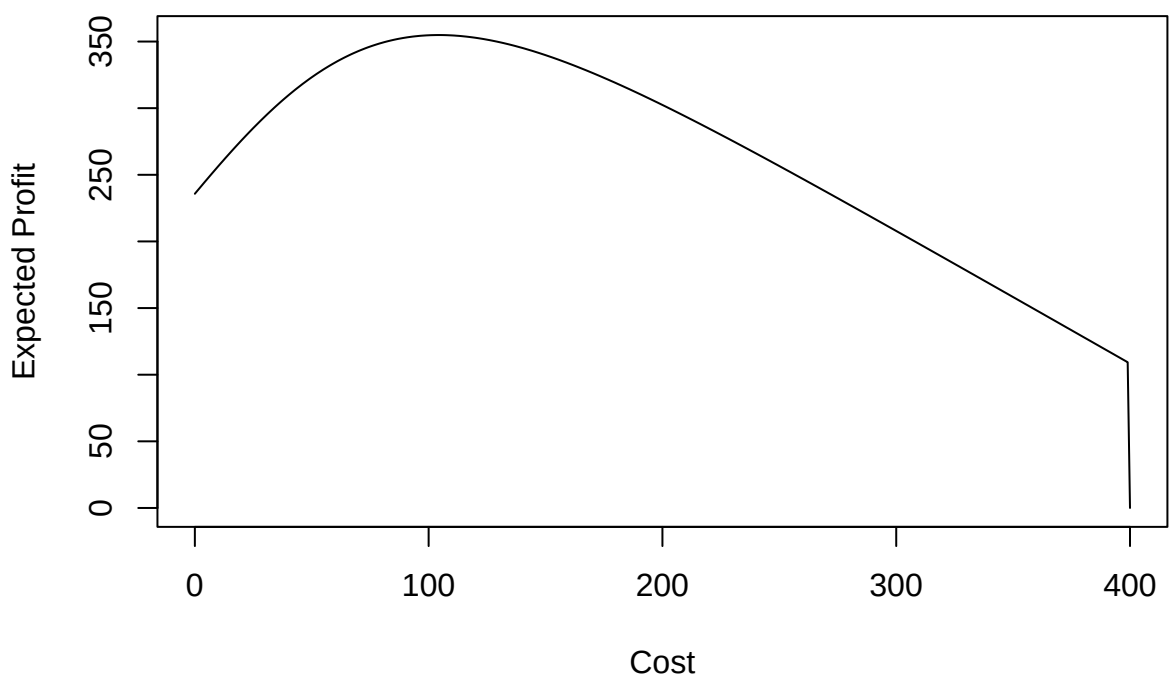


6. Deployment

```
#### Cost curves to maximize profit
n.new <- nrow(test)
i<- 120 ### randomly selected one passenger
x.new <- test[i,]

maxCost <- as.numeric(floor(x.new$ticket_price))
maxSat <- seq(0,400,by=1) #intend improve satisfaction. We timed 100 for calculation purpose
profit <- rep(0,length(maxSat))

 for ( s in maxSat) {
  ### prob of dissatisfy with increasing satisfaction [IS]
  x.new['clv'] <- test[i,'ticket_price'] - 100*s*0.01
  x.new['satisfaction.level.overall'] <- test[i,'satisfaction.level.overall'] + s*0.01
  prob.sat.x.new <- predict(m.lr, newdata=x.new, type="response")
  ### CLV with IS given satisfy
  CLV <- x.new[1,'clv']
  profit[s]<- (prob.sat.x.new)*CLV
}

#Sat curve

plotsat <- seq(0,4,by=0.01)
sat.curve <- plot(plotsat, profit,xlab="Change of Satisfaction", ylab="Expected Profit", type="l",main =
```




```
#cost curve
cost.curve <- plot(cost_function(maxSat*0.01), profit,  xlab="Cost", ylab="Expected Profit", type="l",ma
```



Thank you!